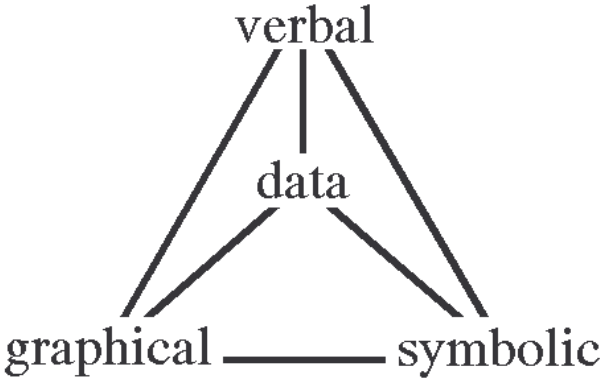
# Laboratories in Quantitative Biology

D. C. Schneider
Memorial University of Newfoundland
St. John's

September 2004



Material in this manual has been copied under license from CANCOPY.
Resale or further copying of this material is strictly prohibited.

Material in this manual has been copied under license from CANCOPY.
Resale or further copying of this material is strictly prohibited.

List of data sets:

| | | |
|---|---|---|
| srbx9_5.dat | Guide | *Daphnia* ages |
| srbx1311.dat | Guide, Lab 8 | Guinea pig litter size |
| srbx14_1.dat | Lab 6 | Tribolium weight loss |
| srbx14_4.dat | Lab 6 | Tribolium survival |
| Garrod.dat | Lab 6 | Cod mortality |
| srtab8_1.dat | Lab 7 | Fly wing lengths |
| fishmov.dat | Lab 7 | Cod movements |
| srbx11_7.dat | Lab 8 | Frog lactic acid production |
| srex1412.dat | Lab 8 | Duck growth |
| wworm1.dat | Lab 9 | Wire worm density 1 |
| wworm2.dat | Lab 9 | Wire worm density 2 |
| leprosy.dat | Lab 9 | Leprosy bacilli |

These data sets can be found on the website for this course.

www.mun.ca/biology/schneider/b4605/data

# Laboratory #1. Inference ('guess the process').

We use inference every day to make general statements about specific events. We might, for example, notice that relatively small snowflakes have begun to fall, and from this expect a big snowfall. Of course not all snow storms that start with small flakes end up dropping a large amount of snow, but the association is frequent enough to be captured in a rule of thumb ("Big snow, little snow; Little snow, big snow") and so serve as a rough guide to how much snow we might expect from a storm.

Inductive methods, where we proceed from the specific to the general, are used to generate knowledge about our world, including the world of living things. Any inductive method can lead us astray, however, because there is no necessary reason that some generalization has to be correct. We might, for example, generalize that all mammals bear live young, then discover an exception (such as a platypus from Australia) that forces us to discard the generalization. This process of guessing and discarding ("conjectures and refutations") is a very effective way of learning about the natural world, especially if we use some well established rules that make the whole process more efficient, and we are conscious of how easily we become attached to our own conjectures.

The purpose of this lab is to provide you with experience in the systematic application of inductive methods.

Specific goals of this lab are:

1. Compare experimental with observational methods of testing and developing generalizations.

2. Compare formal decision trees with more casually designed experiments to test and develop generalizations.

The use of formal decision trees in scientific research was advocated most strongly by Platt (1964), who argues that other methods are slow and ineffective. Here is the first part of Platt's article, which describes the explicit use of decision trees in making inferences.

---

Scientists these days tend to keep up a polite fiction that all science is equal. Except for the work of the misguided opponent whose arguments we happen to be refuting at the time, we speak as though every scientist's field and methods of study are as good as every other scientist's, and perhaps a little better. This keep us all cordial when it comes to recommending each other for government grants.

But I think anyone who looks at the matter closely will agree that some fields of science are moving forward very much faster than others, perhaps by an order of magnitude, if numbers could be put on such estimates. The discoveries leap from the headlines--and they are real advances in complex and difficult subjects, like molecular biology and high-energy physics....

Why should there be such rapid advances in some fields and not in others? I think the usual explanations that we tend to think of--such as the tractability of the subject, or the quality or education of the men drawn into it, or the size of research contracts--are important but inadequate. I have begun to believe that the primary factor in scientific advance is an intellectual one. These rapidly moving fields are fields where a particular method of doing scientific research is systematically used and taught, an accumulative method of inductive inference that is so effective that I think it should be given the name of "strong inference." I believe it is important to examine this method, its use and history and rationale, and to see whether other groups and individuals might learn to adopt it profitably in their own scientific and intellectual work.

In its separate elements, strong inference is just the simple and old-fashioned method of inductive inference that goes back to Francis Bacon. The steps are familiar to every college student and are practised, off and on, by every scientist. The difference comes in their systematic application. Strong inference consists of applying the following steps to every problem in science, formally and explicitly and regularly:

**1) Devising alternative hypotheses;**
**2) Devising a crucial experiment (or several of them), with alternative possible outcomes, each of which will, as nearly as possible, exclude one or more of these hypotheses;**
**3) Carrying out the experiment so as to get a clean result;**
**1') Recycling the procedure, making subhypotheses or sequential hypotheses to refine the possibilities that remain; and so on.**

It is like climbing a tree. At the first fork, we choose--or in this case, "nature" or the experimental outcome chooses--to go to the right branch or the left; at the next fork, to go left or right; and so on. There are similar branch points in a "conditional computer program," where the next move depends on the result of the last calculations.

*continued... (over)*

Material in this manual has been copied under license from CANCOPY.
Resale or further copying of this material is strictly prohibited.

*...Continued from previous page*

And there is a "conditional inductive tree" or "logical tree" of this kind written out in detail in many first-year chemistry books, in the table of steps for qualitative analysis of an unknown sample, where the student is led through a real problem of consecutive inference: Add reagent A; if you get a red precipitate it is subgroup alpha and filter and add reagent B; if not, you add the other reagent B'; and so on.

On any new problem, of course, inductive inference is not as simple and certain as deduction, because it involves reaching out into the unknown. Steps 1 and 2 require intellectual inventions, which must be cleverly chosen so that hypotheses, experiment, outcome, and exclusion will be related in a rigorous syllogism; and the question of how to generate such inventions is one which has been intensively discussed elsewhere (2,3). What the formal schema reminds us to do is to try to make the inventions, to take the next step, to proceed to the next fork, without dawdling or getting tied up in irrelevancies.

It is clear why this makes for rapid and powerful progress. For exploring the unknown, there is no faster method; this is the minimum sequence of steps.

Reprinted from:
J.R. Platt (1964). Strong Inference. *Science* **146**: 347-353. © AAAS, Washington, D.C.

Material in this manual has been copied under license from CANCOPY.
Resale or further copying of this material is strictly prohibited.

Does Platt's method of strong inference work better than other, less formal methods? To find out, we will play 3 versions of a game called inferential cards. The third version uses Platt's method of strong inference. If Platt's method is better we'll be able to measure the difference by comparing the 3 versions.

The rules for inferential cards are simple.

--One person, ("Nature") develops a rule for distributing cards into two or more piles.
--This person then shuffles a deck of cards, takes cards one at a time off the top of the deck, and places them face up in piles according to the rule.
--The other players ("the scientists") observe how cards are placed.
--As soon as possible, the scientists guess the rule for placement.
--The person placing the cards replies "yes" or "no" to each guess or "conjecture."
--The game continues until the scientists correctly guess the rule, or until the deck is exhausted.

To start, form up into groups of 3 (groups of 4 or 2 will also work, but 5 is too many!). The dealer ("Nature") thinks up a rule, then places cards according to the rule. The scientists work together to guess what the rule might be, then present that rule to "Nature." In true decision-theoretic fashion "Nature" can **only answer YES or NO** to each conjecture posed by the scientists. When the rule is guessed (or the deck exhausted) write the rule on the form that has been drawn up for this purpose (at the end of this lab). Fill out the names of the people who have seen the rule, and record the number of cards that were placed before the rule was correctly guessed. Record this number of cards placed under "random" cards. Fold the paper so that names are on the outside and rules on the inside, out of sight. Hand this in to the lab instructor--we will be using these again later.

After one game, switch roles so that the dealer (who made up the rule) is now a scientist. The new dealer makes up a new rule. When this has been guessed, write the rule and the score on a new piece of paper, folded with rule on the inside, names on the outside. Pass this also to the lab instructor.

The next step will be to compare this method, based on observations from cards in random order, to a more experimental method, based on selected cards. We will find out whether the scientists can guess the correct rule more efficiently by selecting cards for "Nature" to classify.

To make the comparison, we will use rules that have already been guessed by the observational method. One person ("Nature") should obtain one of these rules from the lab instructor, then read it silently. The scientists, who now hold the deck of cards, decide which cards to show to "Nature" for placement in piles. As soon as possible, the scientists make a conjecture as to the rule. As before, "Nature" can only reply "NO" or "YES." When the rule is guessed, add everyone's name to the list of people who have seen the rule. Record, as "selected cards," the number of cards to guess the rule. Did the scientists do better (fewer cards) than with the randomly presented cards?

Everyone should have a chance to use the experimental method, so appoint another person to be "Nature" and repeat this with another rule. "Nature" should go find the lab instructor in order to exchange the rule just guessed for a new rule that your group has not seen. The scientists again select cards for "Nature" to place according to the rule. Conjectures continue until the rule is guessed. The number of cards to guess the rule is recorded as "selected cards." Now that you have seen the rule, add your names to the sheet for this rule.

The last step will be to see if Platt's method reduces the number of cards to guess the rule. By now we have arrived at a fairly sophisticated style of "conjecture and refutation" where we deliberately choose a sequence of cards in order to infer the correct rule as efficiently as possible. We have remained casual about how we develop this sequence. We have been using an informal decision tree, rather than working from a formal decision tree. Platt (1964) states that a formal decision tree (strong inference) works more efficiently. We'll test this.

Delegate someone to be "Nature." While "Nature" is off exchanging an old rule for a new rule, the scientists re-read Platt's 3-step program, devise alternative hypotheses (step 1), and devise a crucial experiment to distinguish these hypotheses (step 2). Plat recommends "multiple working hypotheses" but this makes step 2) difficult. Try using mutually exclusive hypotheses for step 1) rather than the "one or more" hypotheses of Platt. Before showing any cards to "Nature" write down step 1) and step 2). Here is an example of mutually exclusive hypotheses, with test.

1) H1: Red and black cards in separate piles.
   H2: Red and black mixed in same pile.
2) Test: Red 10 and black 10 presented to "Nature"
3) Result: Cards in separate piles so H1 accepted.

Repeat the cycle by devising a new pair of mutually exclusive hypotheses. Write down hypotheses and test, then present the test cards to nature. Keep a record of your sequence of hypotheses, treatments, and outcomes (a sort of lab notebook) as you will need this later.

When the rule is guessed, add the names in your group to the list for this rule. Then write down the number of "test" (either/or) cards required for mutually exclusive hypotheses. Was there an improvement over the two previous methods used to guess this rule ?

**Table 1.1**. Number of cards until rule is guessed correctly.

| | random | selected | test either/or | crucial |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| etc... | | | | |

At the end of the lab we will tabulate results.

**Optional**. Platt recommends multiple working hypotheses at each step, rather than sequence of mutually exclusive alternatives at each step. Platt does not mention mutually exclusive hypotheses, nor does he give reasons for multiple working hypotheses. More recent studies have shown that people tend to use "confirmation bias" when inferring rules for patterns. That is, they stick with a conjecture and try to show that it is correct, rather than trying to discredit it. They become attached to their idea.
One way to correct this is to use multiple working hypotheses, so as not to become attached to one idea. If your group is interested, try using the method of multiple working hypotheses, rather than a sequence of mutually exclusive pairs. To do this, you will need to keep a record of multiple hypotheses for step 1). At step 2) you will need to select one or more "crucial cards" to present for placement in piles by "Nature."

If you are interested in finding out whether multiple working hypotheses work better than tests of mutually exclusive alternatives, then send "Nature" to obtain a rule from the lab instructor. Make sure the rule has already been tested by random cards, selected cards, and test (either/or) cards. Keep a record of all of your sequence of hypotheses, treatments, and outcomes in your lab notebook. Record the number of cards required on the rule sheet, as "crucial cards."

**Write-up.**

Hand in a copy of Table 1.1, first 3 columns only.
(An effort will be made to send this to everyone via e-mail)
Hand in one example of a decision tree.
Provide a brief written comparison of the 3 methods in Table 1.1.
Discuss the following questions.

1. Did a selected sequence of cards improve the efficiency of inference, as compared to a random sequence?
2. Did a formal decision tree improve the efficiency of inference, as measured by number of cards, relative to an informal tree?
3. You might also want to comment on Platt's (1964) use of "men" instead of "people" in the third paragraph of the article. (This part will not be marked).

Inferential Cards.

The rule should be kept out of sight until guessed. NO clues.

"Nature" is allowed to say "yes" or "no" in good decision theoretic form, as with statistical analysis.

Write out the rule here. When your group is done, use the reverse side to write names of people (either as a guesser or as Nature) who have seen this rule.

Before placing cards according to the rule, make sure no one has seen the rule (names on reverse).

RULE:

Number of random cards ____

Number of selected cards ____

Number of test (either/or) cards _____ (Number of experiments _____)

Number of crucial cards ____ (multiple working hypotheses)

Number of _____ cards
(selected/test/crucial)

## Inferential Cards

Names of people who have seen this rule.

Names _____ _____

Names _____ _____

Names _____ _____

Names _____ _____

Names _____ _____

Names _____ _____

Names _____ _____

Names _____ _____

Names _____ _____

Names _____ _____

# Laboratory #2. Using Equations.

The purpose of this lab is to give you practice in translating and using formal models of biological phenomena, expressed in the form of equations.

Quantitative descriptions of biological phenomena permit one to calculate outcomes of interest, whether these be the clearance rate of a drug from the blood, or the production of wheat in a dry year. Calculations follow a recipe or algorithm that often is written in the compact form of a symbolic expression, or equation. The ability to carry out quantitative work in biology, as in the other natural sciences, depends on facility in translating and using symbolic expressions such as equations.

A second motivation, apart from the practical ability to carry out calculations, is that important ideas in some areas of biology are expressed in symbolic form, typically as equations or diagrams. The ability to understand these ideas depends upon the ability to translate abstract equations into concrete terms. For example, Kleiber's Law states that metabolic rate varies with body mass according to the following relation:

$$\dot{E} = \alpha M^{\beta}$$

where $\dot{E}$ is metabolic rate at rest, M is mass of an animal, $\alpha$ is a parameter with units that scale metabolic rate to body mass, and $\beta$ is a unitless parameter slightly greater than a theoretical value of 2/3 based on the ratio of surface area ($Length^2$) to volume ($Length^3$). Remember that parameters hold constant in a situation, they are conventionally expressed with Greek symbols, and they are often estimated from data by statistical methods. In contrast, the variable quantities $\dot{E}$ and M can take on any of several values. With this information we can now state Kleiber's Law in more concrete terms:

> "*The metabolic rate $\dot{E}$ of an animal at rest is directly proportional to its biomass M, raised to a power slightly greater than 2/3.*" (2/3 is the surface to volume ratio).

For passerine (perching) birds, Lasiewski and Dawson (1967, *Condor* 69:13) estimated that $\alpha$ = 129 kcal $day^{-1}$ $kg^{-0.724}$, and that $\beta$ = 0.724, a pure number with no units or dimensions.

Now re-write the equation with these parameter values _____

With this equation, you should now be able to calculate the minimum food energy requirement (in kilocalories per day) of a 20 gram (0.02 kg) canary.

$\dot{E}$(0.02 kg) = _____ kcal/day  Try stating this result in words.

The conventional way of reading $\dot{E}(0.02\ kg)$ is "metabolic rate at two hundredths of a kilogram."

Now pick several reasonable values of the explanatory quantity (avian mass in kg), and make calculations of minimum (resting) metabolic rates.

$\dot{E}$(____kg) = _____ kcal/day  $\dot{E}$(____kg) = _____ kcal/day

$\dot{E}$(____kg) = _____ kcal/day  $\dot{E}$(____kg) = _____ kcal/day

Everyone should do at least two of these on their own calculator, to become familiar with the mechanics. It also helps to state one of these calculations in verbal form, as a complete sentence. For example:

"The expected metabolic rate of a ___ gram bird, based on Laseiwski and Dawson's equation, is _____ kcal per day."

The parameter values in this example are by no means unique. Other values are possible. For example Laseiwski and Dawson (1967) also estimated $\alpha$ and $\beta$ for non-passerine birds. Try substituting their estimates ($\alpha = 78.3$ kcal $day^{-1}$ $kg^{-0.723}$, $\beta = 0.723$) to obtain an equation to calculate resting metabolic rate of a non-passerine bird from body mass, according to Kleiber's Law.

$\dot{E}$ = _____

Compare this equation (or model) to that for passerines. Which will have the greater metabolic rate, a passerine bird, or a non-passerine of the same mass?

_____

Now check your guess by calculating the resting metabolic rate of a 20 gram non-passerine, and comparing it to the calculated rate for a passerine of the same mass.

$\dot{E}(0.02\ kg)$ = _____ kcal/day  Try stating this result in words.

The parameter $\alpha$ in Kleiber's Law has units, as do many of the parameters used in biology. Parameters should be stated with their units, because otherwise it is all too easy to write equations leading to erroneous calculations. For example, if a physiologist reports an estimate of $\alpha = 619$ for passerine birds, and fails to report the units, then it is natural to assume that this refers to kilocalories, which is incorrect. If we assume the physiologist meant kilocalories (for failure to report units) then our calculation will be far too high.

Try writing Kleiber's Law for $\alpha = 619$, then calculate the metabolic rate for the 20 gram canary.

$\dot{E}(0.02\ kg) =$ _____ (kcal/day?)

This calculation is _____times higher than the rate for a resting passerine.

Failure to report units ($\alpha = 619$ kilojoule $day^{-1}\ kg^{-0.724}$) led to this error.

Try calculating the metabolic rate (in kilojoules per day) of a 20 gram canary and a 500 gram crow.

$\dot{E}(0.02\ kg) =$ _____ kJ/day $\dot{E}(0.5\ kg) =$ _____ kJ/day

Based on this example, here is a generic recipe for translating and using equations:

Table 2.1. Calculations from equations expressing biological ideas.

1. Write the equation.
2. Write each symbol, with units.
3. State in words the idea expressed by the equation.
4. Obtain parameter estimates and write an equation for these values. Sometimes these values are obtained from theory but more often they are obtained by estimation procedures such as least squares regression.
5. Make sure units on left side equal those on the right.
6. Substitute specific values of variable quantities to calculate the quantity of interest from the equation.

For this lab, you will be given 3 examples from publications that use formal models (equations) to express biological concepts. In each example (tamarin behaviour, *Notonecta* allometry, helminth infection of snails) one equation has been circled. For each equation, provide a complete translation and calculation, following the 6 steps listed in the Table above.

You are encouraged to work in groups, discussing your answers.

After you have completed these 3 examples (all 6 steps), find another example in the library, state the journal name with volume and page number, and provide a complete translation and calculation.

---

**Example 1** Tamarin behaviour.

In a study of maternal behaviour of 12 red-bellied tamarins, estradiol was the most abundant urinary estrogen during late pregnancy. Infant survival was used to divide this group into good and poor mothers as follows (p. 721 of article).

"In the group of females who were subsequently defined as good mothers, five out of six had two surviving infants after one week, including two mothers without sibling infant care experience; a third good mother without experience had one surviving infant. In the group defined as poor mothers, two mothers with sibling infant care experience had one surviving infant, and all other mothers had no surviving infants at the end of the first week; 10 out of 13 infants of poor mothers died at day 0."

Figure 1 from the article shows the mean concentrations of urinary estradiol during the last 35 days of gestation in good and poor mothers.

Calculate the expected estradiol level in poor mothers 33 days before birth, using the linear model shown in Figure 1B.

Is this model any good? (Hint: consider whether the calculated value would be consistently too high or too low, depending on the number of days before birth).

This material has been copied under license from CanCopy.
Resale or further copying of this material is strictly prohibited.

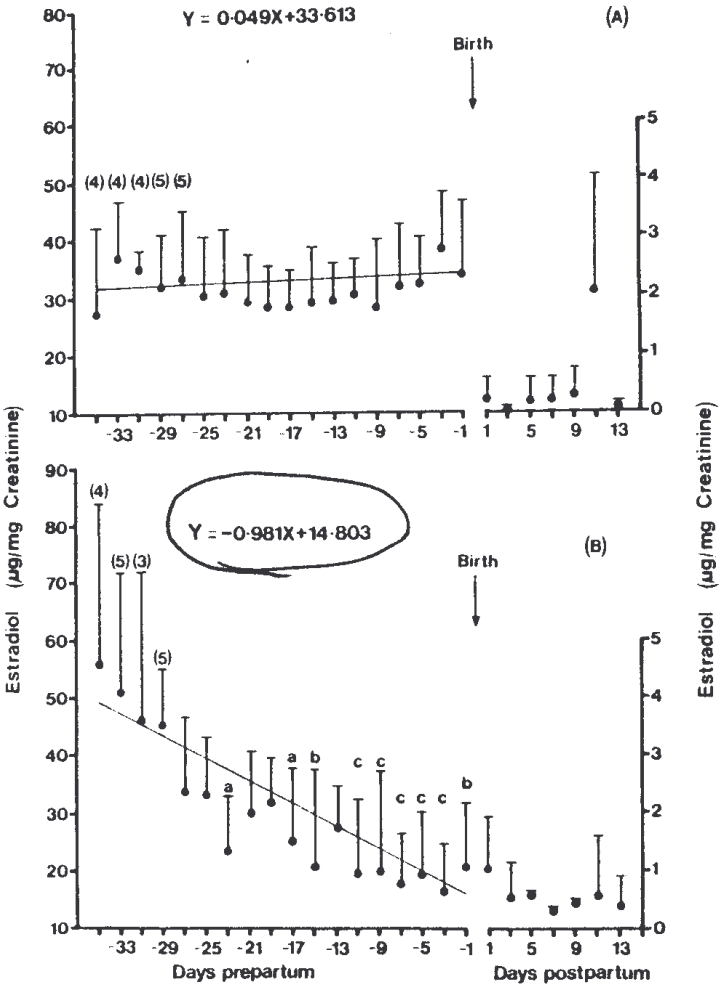



FIG. 1. Changes in the urinary concentration (mean + confidence limits) of total estradiol during late pregnancy in red-bellied tamarins defined as (A) good and (B) poor mothers. Values are expressed as the antilog of the mean of transformed data and, for clarity, only the upper 95% confidence limit is shown. Means were calculated from all samples collected from each type of mother within each two-day period. Unless otherwise stated, the prepartum data in both (A) and (B) are for 6 females; the postpartum data are for 3 females. The relationship between urinary estradiol (Y) and time prior to birth (X) in each type of mother has been estimated by linear least-means square regression; the solid line is for the regression equation [good mothers: $F(1,91)=0.20$, $p>0.65$; poor mothers: $F(1,96)=39.35$, $p<0.0001$ (23)]. a=$p<0.05$ versus day −35. b=$p<0.05$ versus day −35, −33. c=$p<0.05$ versus day −35, −33, −31, −29 (Duncan's Multiple Range Test).

Reprinted from C.R. Pryce, D.H. Abbott, J.K. Hodges, and R.D. Martin (1988) *Physiology and Behaviour* **44**: 717-726. © Pergamon Press.

**Example 2**. *Notonecta* allometry.

In example 104 from Simpson et al. (1960) the exponent α is called the coefficient of allometry (Greek *allo* = "other" and *metron* = "measure"). These coefficients describe the degree to which a part of the body remains in direct proportion to the total length as length changes. Synthlipsis to vertex is a measure of head length in the water bug *Notonecta*.

This material has been copied under license from CanCopy.
Resale or further copying of this material is strictly prohibited.

EXAMPLE **104**. Constants of the allometric growth formula $Y = bx^{\alpha}$ for the average values of 37 females and 35 males of *Notonecta undulata*. In each case, $X$ is total body length and $Y$ is the measurement listed in the table. (Data from Clark and Hersh, 1939)

| | FEMALES | | MALES | |
|---|---|---|---|---|
| | $b$ | $\alpha$ | $b$ | $\alpha$ |
| Body width | 0.616 | 0.810 | 0.656 | 0.765 |
| Head width | 0.445 | 0.790 | 0.482 | 0.742 |
| Synthlipsis to vertex | 1.039 | −0.271 | 1.020 | −0.262 |
| First leg | 0.422 | 1.074 | 0.470 | 1.013 |
| Second leg | 0.549 | 1.069 | 0.605 | 1.012 |
| Third leg | 1.100 | 0.948 | 1.178 | 0.908 |
| Femur 1 | 0.153 | 1.066 | 0.188 | 0.971 |
| Femur 2 | 0.190 | 1.114 | 0.209 | 1.060 |
| Femur 3 | 0.268 | 1.141 | 0.286 | 1.104 |
| Tibia 1 | 0.140 | 1.144 | 0.152 | 1.101 |
| Tibia 2 | 0.209 | 1.059 | 0.223 | 1.013 |
| Tibia 3 | 0.348 | 0.964 | 0.373 | 0.924 |
| Tarsus 1 | 0.119 | 1.028 | 0.138 | 0.945 |
| Tarsus 2 | 0.143 | 1.056 | 0.150 | 1.010 |
| Tarsus 3 | 0.517 | 0.742 | 0.503 | 0.743 |

Reprinted from G.G. Simpson, A. Roe, R.C. Lewontin. (1960) *Quantitative Zoology*. © Harcourt Brace & World, New York.

Which parts of *Notonecta* have allometry coefficients close to 1? (These parts of the body remain in proportion as the animal grows)

Which parts of *Notonecta* have positive coefficients far from 1? (These parts change in proportion to body length, such that the relative size of the part becomes smaller in larger specimens).

This material has been copied under license from CanCopy.
Resale or further copying of this material is strictly prohibited.

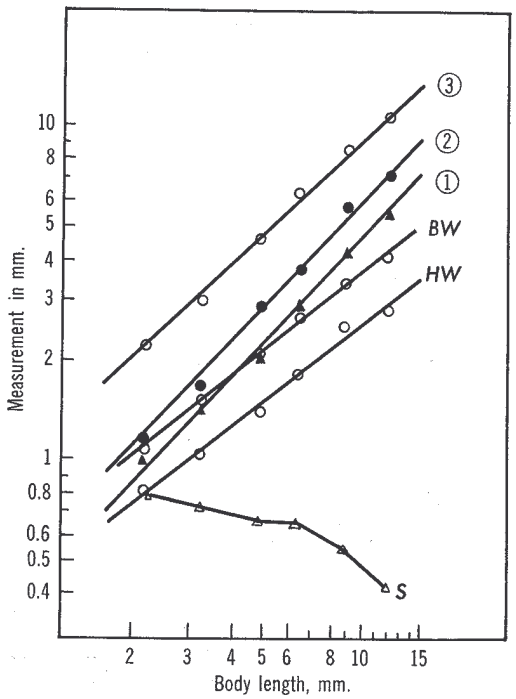



FIGURE **56.** Relative growth curves of several measurements in *Notonecta undulata*, plotted logarithmically against the logarithm of body length. *S.* head length; *HW:* head width; *BW:* body width; *1*, *2*, and *3:* lengths of first, second, and third legs. (Data of Example 105, from Clark and Hersh, 1939)

Reprinted from G.G. Simpson, A. Roe, R.C. Lewontin. 1960. Quantitative Zoology. © Harcourt Brace & World, New York.

Which parts of *Notonecta* have negative coefficients of allometry? (The absolute size of this part becomes smaller in larger specimens)

_____

Write an equation for body width of male *Notonecta undulata*.

Calculate the expected body width of a 4 mm and an 8 mm long male, and express this as the ratio of the expected ratio of body width to body length.

Width(4 mm) = _____ Width(4mm) / 4mm = _____

Width(8 mm) = _____ Width(8mm) / 8mm = _____

This material has been copied under license from CanCopy.
Resale or further copying of this material is strictly prohibited.

**Example 3** Helminth infection of snails.

This model predicts that in areas where schistosomiasis is endemic, in the sense that the average human worm burden is greater than or of the order of unity, the equilibrium fraction of snails observed to be releasing cercariae, $y^*$, will be

$$y^* = f / [ f + (1-f)(\mu''/\mu')]. \quad (1a)$$

Here f is defined for convenience as

$$f = e^{-\mu'\tau} \quad (1b)$$

The data discussed in the preceding sections suggests the death rate of infected snails during the latent period of infection ($\mu'$) is approximately equal to the death rate of uninfected snails ($\mu$). If this is assumed to be true, equation (1) reduces to

$$y^* = f / [ f + (1-f)(\mu''/\mu)], \quad (2a)$$

with

$$f = e^{-\mu\tau} \quad (2b)$$

Empirical estimates of the parameters $\mu$ (death rate of uninfected snails and infected snails not releasing cercariae), $\mu''$ (death rate of snails shedding cercariae) and $\tau$ (pre-patent period of infection) are presented in Tables 4, 5, and 6. For example, *S. mansoni* has a pre-patent period ($\tau$) of approximately 5 weeks at 25 °C (Fig. 4a) and, in St Lucia, Sturrock & Webbe (1971) estimated the mortality rates $\mu$ and $\mu''$ to be roughly 0.15 and 0.61/snail/week, respectively. The insertion of the parameter values into equation (2) yields the prediction that the equilibrium fraction of snails shedding cercariae (prevalence of infection) is approximately 18%. This estimate is slightly higher than the observed prevalence figures listed in Table 1, but it is a considerable improvement over the predictions of conventional Macdonald-Nasell-Hirsch models.

Reprinted from R.M. Anderson and R.M. May. 1979. Parasitology 79:63-94. © Cambridge University Press.

First, check the calculation from Anderson and May (1979).

Next, examine Table 4 from Anderson and May (1979) to determine what happens to latent period ($\tau$) with change in temperature.

This material has been copied under license from CanCopy. Resale or further copying of this material is strictly prohibited.

Table 4. *Latent period ($\tau$) from point of infection to release of cercariae:* Schistosoma mansoni

| Snail species | Temperature (°C) | Latent period ($\tau$) (days) | Author(s) |
|---|---|---|---|
| *Biomphalaria glabrata* | 21–24 | 42 | Pan (1965) |
| *Biomphalaria glabrata* | 23–25 | 30–37 | Stirewalt (1954) |
| *Biomphalaria glabrata* | 26–28 | 22–23 | Stirewalt (1954) |
| *Biomphalaria glabrata* | 31–33 | 18 | Stirewalt (1954) |
| *Biomphalaria glabrata* | 23–24 | 31–38 | Evans & Stirewalt (1951) |
| *Biomphalaria glabrata* | 25–28 | 25–45 | Standen (1949) |
| *Biomphalaria glabrata* | 28–30 | 28–35 | Schreiber & Schubert (1949) |
| *Biomphalaria glabrata* | 18 | 55 | Sturrock & Webbe (1971) |
| *Biomphalaria glabrata* | 32 | 17 | Sturrock & Webbe (1971) |
| *Biomphalaria glabrata* | 22–28 | 21–35 | Sturrock & Sturrock (1970) |

Reprinted from R.M. Anderson and R.M. May. 1979. Parasitology 79:63-94. © Cambridge University Press.

What happens to latent period $\tau$ as temperature decreases? _____

What happens to the factor f as latent period changes due to decrease in temperature?

_____

At a guess, will this increase or decrease the equilibrium fraction $y^*$? _____

Calculate a prediction for equilibrium fraction of snails shedding cercariae at 18 °C, based on information in Table 4.

_____

Did decrease in temperature increase or decrease the predicted prevalence of infection?

_____

**Write-up.**

Each individual is responsible for turning in a lab report, which consists of 3 group presentations (steps 1-6) and one individual presentation (steps 1-6).

Name_____

# Laboratory #3.
# Probability Values from Distribution Functions

Statistical analysis begins with frequency distributions. In order to evaluate any particular outcome we need to compare it to a distribution of outcomes. An example of a particular outcome is the mean value we calculate from data. How reliable is this estimate ? To work this out, we need a distribution of mean values. Another example is a statistic used to declare a decision (accept or reject a null hypothesis). To do this, we need a distribution of outcomes when the null hypothesis is true. We declare our decision by comparing an observed outcome (from a sample) to the distribution of outcomes (for the population) when the null hypothesis is true. This is the frequentist approach, developed in first half of the $20^{th}$ century and widely used in biology. An alternative approach is to modify our belief about some hypothesis, based on the distribution of the data. This is the Bayesian approach, which goes back to Bernouilli and Gauss in the $18^{th}$ century. Bayesian analysis is used today in some areas of medical and environmental biology, but will not be covered in this course. Either way, frequentist or Bayesian, we use frequency distributions.

A distribution can be obtained by sampling data repeatedly. In the widely used frequentist approach, we make the null hypothesis true for the data by sampling the data in random order, then generate a distribution of outcomes by sampling the data repeatedly. This is the basis of a randomization test (see Laboratory 4).

A distribution can also be obtained via a mathematical expression, the probability density function or pdf. In the frequentist approach this function is used to calculate the expected relative frequency distribution of outcomes. Using a function is quicker and easier than generating our own distribution by randomizing the data. But there is a catch. The theoretical distribution must be appropriate. For example, if the residuals are not normal in a regression, then a theoretical distribution (the F-distribution) cannot be used to calculate p-values concerning the slope of the regression line.

☞ Randomization tests make no assumptions, but are a lot of work.

☞ Theoretical distributions are less work, but require assumptions that have to be defended.

The commonly used theoretical distributions for data are binomial, Poisson, and normal. There are many others (uniform, gamma, negative binomial, etc). The commonly used theoretical distributions for statistics are the normal distribution, the t distribution (for means based on small samples), the $X^2$ (chi-square) distribution (for variances based on small samples), and the F-distribution (for ratios of variances based on small samples).

The goal of this laboratory is to demonstrate the calculation of probability values from theoretical frequency distributions.

We begin with the binomial distribution. Name_____

1. After a week of training, a planarian worm correctly guesses whether to turn left or right to obtain food in a T-maze on 6 successive trials. What is the probability of doing this by chance alone?
   n = 6 trials
   X = 6 successes in 6 trials
   p = 50% That is, we assuming an equal probability of turning left or right.

We can use our calculators to work out the answer, step by step.
The probability of 1 success in 1 trial 0.5
The probability of 2 success in 2 trials (0.5)(0.5) = 0.25
The probability of 3 success in 3 trials _____ = _____
The probability of 6 successes in 6 trials _____ = _____

We can use a computer to calculate probabilities, either with a spreadsheet or with a statistical package. To do this we need a functional expression to guide us.

We are interested in P{X= x} the probability (relative frequency) of x successes in n trials.
The symbol P{X= x} is read: 'the probability that successes X will take on a certain value x.'
The probability distribution is a table drawn from numbers, or a graph, or a functional expression, or some other device that relates possible values x to probabilities P{X= x}.
The functional expression for a run of x successes in n trials $f(x) = p^x$
In the example above $f(6) = (0.5)^6 = 0.0156$

The calculation according to the functional expression can be carried out on a calculator with statistical functions, on any spreadsheet, or with any statistical package. Here are three examples. One is for a spreadsheet (Excel). The second is for Minitab, using pull down menus. The third is for Minitab again, this time with the line commands rather than from the menu.

Excel Spreadsheet. Select a cell
Excel menu
Function
Statistical functions
Binomdist
number_s = 6
trials = 6
probability_s = 0.5
cumulative = false (we want the probability, pdf)

Name_____

Minitab Menu
Calculate
Probability Distributions
Binomial
Probability (pdf)
trials = n = 6
success rate = p = 0.5
Input constant = x = 6

```
MTB> pdf 6;
SUBC> binomial 6 0.5.
```

Even though these 3 boxes look really different, they all use the same functional expression to make the same calculation

$f(x) = p^x$

$(6) = 0.5^6 = 0.0156$

For the remainder of this lab, you can use any method you like: calculator with statistical functions, spreadsheet, or statistical package. Help with computational details will be provided from time to time, using Excel and Minitab menus. It should be possible for you to work out the computational details for the menu-driven approaches. The command line approach is harder to use, so it will be shown more often.

At this point write out the method you will be using in this lab _____

Now calculate the probability of
5 correct choices in a row, in 5 trials
assuming a success rate of 50% on each trial _____

7 correct choices in 7 trials, assuming a
success rate of 80% at each trial _____

Now that we can calculate a single probability, we move to calculating the entire probability distribution.

Name_____

2. If we carry out the planaria experiment repeatedly, and no learning occurs (p = 50%), what is the probability distribution of outcomes p{X = 0 successes, 1 success, 2 successes, 3 successes, etc.} in 6 trials ?

   To compute this, we use the functional expression for the binomial distribution.

$$f(x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

   In this expression x varies from 0 to n.
   n! means the factorial of the number n. 6! = 6·5·4·3·2·1 0! = 1
   (any number)$^0$ = 1
   From this, we calculate the probability of 6 successes.

$$f(6) = \frac{6!}{6!(6-6)!} 0.5^6 (1-0.5)^{6-6} = [\ \_\_\_\_\_\ ]$$ Be sure to apply algebra to simplify the expression.

$f(0)$ = _____ = [ _____]

Write out the expression for $f(0)$ then do the calculation by using x = 0 instead of x = 6.

We could continue in this fashion, for $f(1), f(2)$, *etc*. This is laborious so we will calculate a column of probabilities $f(x)$ from another column x. The procedure is nearly the same in any package with a spreadsheet, including statistical packages that use spreadsheet input. Because the procedure is so similar among packages, it is first shown as pseudocode–a list of procedures to be carried out in any package. The pseudocode is then translated into the specific procedures of a spreadsheet (Excel), a menu-based statistical package (Minitab), and a command line procedure (Minitab again).

Generic recipe to calculate a probability density function $f(x)$

Pseudocode (applicable to almost any package).
- Select a column and name it x.
- Place the values k = 0,1,...6 into this column.
- Select an adjacent column and name it $f(x)$.
- Select the first cell in column $f(x)$.
- Apply the binomial function to calculate $f(0)$ from the cell x = 0.
- Apply the function to the rest of the column $f(x)$.

***Calculate a probability density function f(x)***

Once you have looked at the pseudocode, carry out the procedure in any package you like.

Name_____

Excel Spreadsheet. Select a top cell in $f(x)$
Excel menu
Function
Statistical functions
Binomdist
number_s = adjacent cell with value of x = 0
trials = 6
probability_s = 0.5
cumulative = false (we want the probability, pdf)
Select cell with $f(0)$, copy, then paste in rest of column $f(x)$.
Make a plot of $f(x)$ versus x.

Minitab worksheet. Column 1, name it x. Put 0,1,2, ... 6 into column
Column 2, name it $f(x)$
Minitab menu
Calculate
Probability Distributions
Binomial
probability
trials = n = 6
probability of success = p = 0.5
Input column = c1
Storage = c2
Graph
Plot $f(x)$ on Y-axis, x on x axis

```
MTB > set c1
DATA> 0 1 2 3 4 5 6
DATA> end
MTB > pdf c1 c2;
SUBC> binomial 6 .5.
MTB > name c1 'x' c2 'f(x)'
MTB > print c1 c2
MTB > plot c2 c1
```

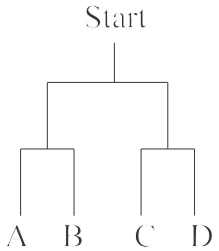
Print your plot and tape or paste it here.

Name_____

Next, we move to cumulative frequency distributions.

3. Can a planarian worm learn to avoid making two successive right turns ? In a maze with 2 tiers (see right) a randomly moving worm will arrive at each of the four endpoints (A, B, C, or D) with equal probability. What proportion of the time will a randomly moving worm arrive at endpoint D?

p = _____




A worm trained to avoid endpoint D arrives at D only once in 6 trials.
What is the probability of doing this by chance alone ?
Place your guess (this will be marked as present, not right or wrong) $P\{X \leq 1\}$ = _____

To calculate the answer, we need to accumulate probabilities.
We need to add the probability of no arrivals and the probability of arriving once at D.

What is the chance of never arriving at D in 6 trials? $f(0)$ = _____
What is the chance of arriving at D exactly once in 6 trials ? $f(1)$ = _____

Now add the probabilities to find out the chance
of arriving one or fewer times at endpoint D. $P\{X \leq 1\} = F(1)$ = _____

Adding up the probabilities in the tail of the distribution gets tedious.
Our spreadsheet or statistical package will do this for us by using the cumulative distribution function F(x).

The functional expression is $F(x) = \Sigma f(x)$
In the example above $F(1) = f(0) + f(1) = 0.17798 + 0.35596 = 0.53394$

Here is the generic recipe for calculating the cumulative distribution function F(x)

Pseudocode (applicable to almost any package).
- Select a column and name it x.
- Place the values k = 0,1,...6 into this column.
- Select an adjacent column and name it F(x).
- Select the first cell in column F(x).
- Apply the binomial function to calculate F(0) from the cell x = 0.
- Apply the function to the rest of the column F(x).

***Calculate a cumulative distribution function F(x)***

Specifics for a spreadsheet (Excel) and a statistical package (Minitab) appear on the next page.

Name_____

Excel Spreadsheet. Select a top cell in F(x)
Excel menu
Function
Statistical functions
Binomdist
number_s = adjacent cell with value of x = 0
trials = 6
probability_s = 0.25
cumulative = true (we want the cdf)
Select cell with F(0), copy, then paste in rest of column F(x).

Minitab worksheet. Column 3, name it F(x)
Minitab menu
Calculate
Probability Distributions
Binomial
Cumulative probability, *etc*

```
MTB > set c1
DATA> 0 1 2 3 4 5 6
DATA> end
MTB > pdf c1 c3;
SUBC> binomial 6 .25.
MTB > name c3 'F(x)'
```

Make sure your calculations using the cumulative distribution function match the calculations by summing the probability density function.

Tape, paste, or fill in by hand your table showing

| x | pdf *f*(x) | cdf F(x) |
|---|---|---|
| 0 | | |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |

Name_____

In statistical work, we use cumulative distributions all of the time.
In the last example we looked at $P\{X \leq x\}$ the probability of x or less successes in n trials.
More often, we are interested in $P\{X > x\}$ the probability of exceeding a certain value x. To obtain this probability we compute the cumulative probability $P\{X \leq x\}$, then subtract it from 1, to obtain $P\{X > x\}$.

4. A planarian worm is trained to make two right turns to arrive at endpoint A in the two tier maze shown above. What is the probability of arriving at endpoint A by chance 5 or more times in 6 trials ?

We can compute this by accumulating probabilities in the upper tail, using the pdf.

The functional expression is $P\{X > x\} = 1 - F(x) = 1 - \Sigma f(x)$
In the example above $P\{X > 4\} = 1 - F(4) = 1 - \Sigma f(x) = f(5) + f(6)$

$f(5) =$ _____
$f(6) =$ _____
$P\{X > 4\} =$ _____

We can also compute this by using the cumulative distribution function.

The functional expression is $P\{X > x\} = 1 - F(x) = 1 - \Sigma f(x)$
In the example above $P\{X > 4\} = 1 - F(4) = 1 -$ _____ = _____

5. Another example. Binomial distribution.
Draw a 3 tier maze with 8 possible endpoints. Label them A through H
Success is defined as arrival at point H.

What is the expected success rate on each run through the maze? p = _____

What is the chance of 4 correct endpoints in 4 trials? $P\{X = 4\}$ = _____

3 or more correct endpoints in 4 trials? $P\{X > 2\}$ = _____

6 or more correct outcomes in 9 trials? $P\{X > 5\}$ = _____

Name_____

Next we move to theoretical frequency distributions for commonly encountered statistics. That is, the Chi-square, t, and F-distributions.

6. A botanist analysing the sex ratios of Holly trees *Ilex* in a field obtains a G-statistic of 3.84 in a single degree of freedom test of goodness of fit to a one-to-one sex ratio. What is the chance of obtaining this outcome (G = 3.84 or more) by chance alone?

To evaluate this outcome, we use the chi-square $X^2(n)$ with n = 1 degree of freedom.
We want the probability in the upper tail $P\{X^2(1) > 3.84\}$.
This is the probability of obtaining a value of $X^2(1) = 3.84$ or more.
Some packages report the upper tail $P\{X^2(n) > 3.84\}$, which is what we want.
Other packages report the cumulative distribution $F(x) = P\{X^2(n) \leq x\}$.
So when we plug in the value x = 3.84 and df = 1 we obtain $F(3.84) = P\{X^2(1) \leq 3.84\}$
To obtain the upper tail, we subtract: $P\{X^2(1) > 3.84\} = 1 - F(3.84)$

Here is the generic recipe for calculating a p-value for a Chisquare statistic on a spreadsheet.

***Calculate a p-value for a Chisquare distribution***

```
Pseudocode (applicable to almost any package).
    Select a column and name it Gstat.
    Place the value Gstat = 3.84  in the first cell.
    Select an adjacent column and name it P(Gstat).
    Select the first cell in column P(Gstat).
    Apply the Chisquare function to calculate P(3.84) from
        cell Gstat = 3.84.
    Compute the upper tail if the package gives the cdf, F(x).
```

Now try using the pseudocode to carry out the calculations in the package you are using.
If you have trouble, the specifics are shown for a spreadsheet that calculates the tail probability.
The specifics are then shown for a statistical package that calculates the cumulative distribution function, hence returns p rather than 1 – p

```
Excel menu
    Function
        Statistical functions
            Chidist
                X = cell 1 in adjacent column (3.84)
                Deg_freedom = 1
```

Name_____

Minitab menu
Calculate
Probability Distributions
Chisquare
Cumulative probability
Degrees of freedom = 1
Input = column 1
Storage = column 2

```
MTB > cdf 3.84 k1;
SUBC> chisquare  1.
MTB > let k2 = 1 - k1
MTB > print k2
```

The probability of obtaining a G-statistic of 3.84 or more, by chance is _____
(not 0.95)

Now try evaluating the probability of obtaining the following statistic as an outcome of an analysis, assuming that the Gstatistic follows the chisquare theoretical distribution:

Gstat = 5.67 df = 3 p = _____
(not 0.8712)

Gstat = 10.23 df = 5 p = _____

Gstat = 41.4 df = 23 p = _____

7 Graphical analysis is useful in visualizing the flow of computations.
Let's compare the probability density function pdf and the cumulative distribution function cdf of the chisquare distribution having 1 degree of freedom.

| chisq=x | pdf=f(x) | cdf = F(x) | 1-cdf |
|---|---|---|---|
| 0.5 | 0.4394 | 0.5205 | 0.4795 |
| 1 | 0.2420 | 0.6827 | 0.3173 |
| 2 | 0.1038 | 0.8427 | 0.1573 |
| 4 | 0.0270 | 0.9545 | 0.0455 |
| 8 | 0.0026 | 0.9953 | 0.0047 |

Use your package to plot the pdf versus chisquare values on the x axis.

Sketch pdf = $f(x)$

Name_____

In the box, make a sketch of the pdf plot you just created.
Then try to draw the cdf. (this will be marked as present or absent, not on whether it is correct).

F(x) = cdf

Now use your computer package to graph the cdf versus the same chisquare values on the X axis. Make sketch of this graph, in the box to the left. (This will also be marked on whether it is roughly correct, not on whether it is correct in detail).

How does the plot compare to your sketch? _____

8. Use your package to compute p-values for the F-distribution. Here is the line code only.

```
MTB > cdf 4.56;
SUBC> F 8 23.
```

$F_{obs}$ = 4.56 numerator df = 8, denominator df = 23 p = _____ (not 0.998)

$F_{obs}$ = 2.28 df numerator = 8, df denominator = 23 p = _____ (not 0.9416)

$F_{obs}$ = 1.23 df numerator = 8, df denominator = 23 p = _____ (not 0.6738)

9. Use your package to compute p-values for a t-distribution. $t_{obs}$ = 3.46 df = 23 p = _____

$t_{obs}$ = −2.15 df = 12 p = _____

10. The Z statistic is normally distributed.
Compute p-values from the normal distribution.

Z = 1.96 mu = 0 sigma = 1 p = _____

Hint:

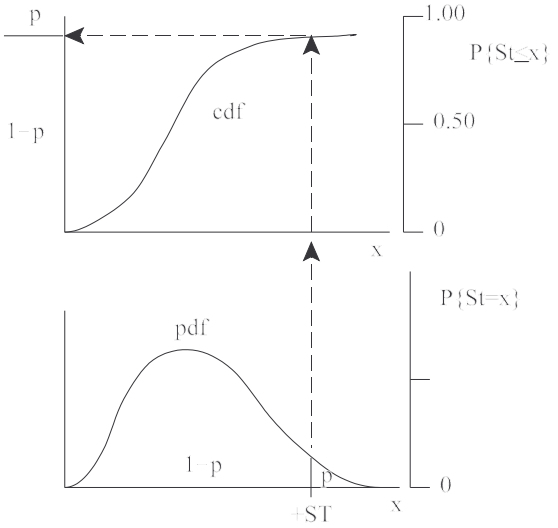
```
MTB > cdf 1.96;
SUBC> normal 0 1.
```

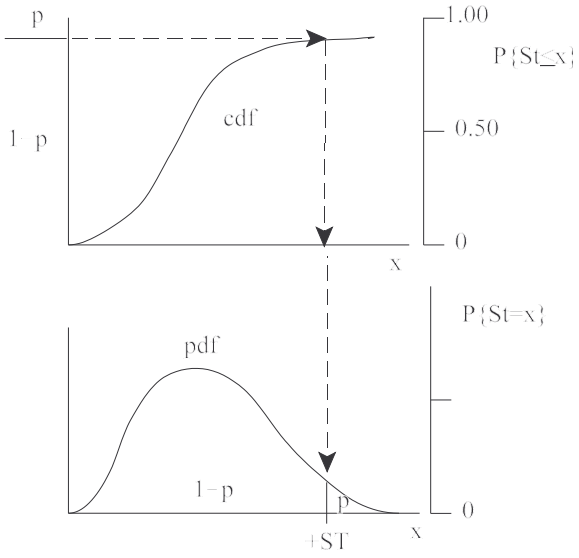
Name_____

The p-values from statistical distributions are the
probabilities from the right tail of the probability density function pdf.
To compute these we use the cumulative distribution function cdf.
As we have seen, we can compute the exact probability of any statistic we like.
Before the days of computers, this was a time consuming chore.
To relieve the chore, tables of critical values were constructed.
These tables work backward from the critical p-value (1%, 5%, *etc*.) to the critical value of the statistic (F, t, Chisquare, *etc*.). The result is an antique style of statistical practice which compares a statistic to a critical value, rather than reporting the exact p-value. The next example demonstrates the construction of these tables.

To work back from a critical p-value (*e.g.* 5%) to a statistic, you need to know how your package behaves. Does it return the probability in the right tail $P\{X > St\}$ for a statistic St? (as Excel does). If it does, use the critical p-value to obtain the critical value of the statistic. Does your package return the cumulative probability $P\{X \leq St\}$ for the statistic St? (as Minitab does). If so, then you will need to use 1−p rather than p to obtain the critical value of the statistic.

Below and to the left is a diagram showing the flow of computations for computing a p-value from a cumulative distribution function, cdf.




On the right is a diagram showing the reverse flow of computations, going from a p-value to a statistic. Find the inverse form of the chisquare distribution for your package.

Write its name _____

Now use the chisquare (df = 1) distribution to calculate the critical value of the $X^2$ statistic, using either p = 5% or 1 – p = 95%, as appropriate for your package.

Did it return the critical value of $X^2 = 3.84$ ? If not, use the appropriate probability so that the package returns a value of 3.84 corresponding to p = 5%.

11. An extremely conservative morphologist wants to work with Type I error set at $\alpha$ = 0.0001, but does not have tables that supply critical values at this extreme probability level. Use the inverse CDF to obtain critical values of $X^2$ on a single degree of freedom test at $\alpha$ = 0.0001.

Use your statistical package to compute the critical chisquare value. Here is the minitab line code only.

```
MTB > invcdf 0.9999;
SUBC> chisquare 1.
```

For $\alpha$ = 0.0001, the critical chisquare value = _____

Why did we use 0.9999 in Minitab to obtain the critical value corresponding to $\alpha$ = 0.0001?

Name_____

12. Now make your own statistical table, for some unusual alpha levels of 0.002 and 0.0002, place critical values of t into the following t-table.

Table 3.1. Critical t-values

| | alpha | |
|---|---|---|
| d.f. | .002 | .0002 |
| 1 | 159.156 | 1590 |
| 5 | _____ | _____ |
| 10 | _____ | _____ |

These tables became unnecessary in the 1970s, when calculators with statistical functions become available. Astoundingly, the tables of critical values are still with us. Even more astoundingly, the backward practice of comparing critical values rather than reporting exact the exact p-value is still with us. If there is electricity in the house, we no longer see hand cranked wringers for clothes, or a block of ice sitting at the top of a cabinet to keep food cool. Remarkably, we still see people with computers using statistical tables.

Now that you know how to compute p-values, here is a poll.
(The next two questions will be marked present or absent only)

Have you ever used statistical tables to work out whether a statistic is significant ? _____

Does it take more time to compute a p-value than
to look up a critical value in a table ? More time to look up the critical value _____
About the same time to do either _____
More time to compute the exact p-value _____

Ronald Smith obtains a t-statistic of $t_{df=5} = 2.60$ ($p = 0.049$)
Ronald Fisher uses a better design, which results in $t_{df=5} = 3.94$ ($p = 0.011$)

If both investigators report only that the test was significant at 5%, without reporting the p-value, which investigator has weakened the reporting of their analysis more ? _____

Why ?

**Write-up for this lab**. Fill in blank spaces as requested, on all the pages of this lab.

# Laboratory #4.
# Probability Values by Randomization.

In the quantitative analysis of biological data we are frequently interested in the uncertainty surrounding our result. Within the frequentist approach, the question of uncertainty translates to a more specific question : How often could the observed result have been obtained by chance alone? To answer this question we compare the observed outcome with the distribution of outcomes in the absence of any pattern or relation (i.e. when the null hypothesis is true).

In some cases we can make this decision relative to a p-value computed from a theoretical distribution of outcomes under the null hypothesis, that there is no pattern or relation. The theoretical distributions that are commonly used for this purpose are the normal, F, t, and $X^2$ (pronounced *khai-square*) distribution.

We cannot always assume that one of these distributions will describe the distribution of our statistic under the null hypothesis. The assumptions for these distributions (normal and homogeneous errors) may not hold for our data. Or, we may need to use a statistic that has no known theoretical distribution of outcomes.

In these cases we can always evaluate our statistic by constructing a frequency distribution of outcomes based on repeated sampling of outcomes when the null hypothesis has been made true by randomization of the data.

This approach to hypothesis testing is called a **randomization test**. It requires no assumptions about the deviations of the data from the model. It will work for any statistic you might devise, or any set of data you might encounter.

**Table 4.1.** Generic recipe for randomization test

1. Compute a statistic (observed outcome)
2. Make the null hypothesis true by randomizing the data
3. Re-compute the statistic to obtain an outcome when the null hypothesis is true
4. Repeat this many times
5. Construct a frequency distribution of outcomes when the null hypothesis is true
6. Compare the observed outcome to the distribution of outcomes, in order to calculate a probability level

The goal of this laboratory is to show how to use widely used statistical packages to obtain probability estimates on an outcome using a randomization methods. The examples will be fairly simple: tests of whether two means differ to a statistically significant degree.

Once you have completed the lab and write-up, you should have

- a better understanding of randomization methods for computing p-values

- a working knowledge of the mechanics of computing p-values by randomization

To begin, look again at Table 4.1 and then draw a diagram showing how to do a randomization test.

The first example is from Box 9.5 of Sokal and Rohlf (1995). Does the average age at beginning of reproduction in one strain of *Daphnia longispina* differ from that of another strain? We begin by computing the difference in average age at reproduction between the two strains.

Generic recipe for bringing data into a spreadsheet or spreadsheet interface.

| Pseudocode (applies to any spreadsheet) | ***Define Data from file*** |
|---|---|
| Open file that has data, usually on the desktop.<br>Use mouse to highlight the data<br>Copy the highlighted data<br>Paste onto the spreadsheet<br>Name the columns | |

For code specific to Minitab, see the ***Define data*** command in the computational guide in this manual.

Generic recipe for calculating difference of means of two variables (columns)

| Pseudocode (applies to any spreadsheet) | ***Calculate statistic AvDiff(X1,X2)*** |
|---|---|
| Select a location to place *AvDiff(X1 –X2)*<br>Define the function *AvDiff(X1 –X2) = mean(X2) – mean(X2)*.<br>Calculate *AvDiff(X1 –X2) = mean(X2) – mean(X2)*. | |

Now that you have looked at the pseudocode, compute the difference in mean age between the two strains of *Daphnia*, using the data from Srbx9_5.dat.

If you are using Minitab, be sure you can see the commands in your output. To do this:

Minitab menu.
Editor
Enable commands

Now use the calculator.

Minitab menu.
Calculate
Calculator
Store result in k1
Expression = mean(c1)
Calculator
Store result in k2
Expression = mean(c2)
Calculator
Store result in k3
Expression = mean(c1 - mean(c2)

```
MTB > let k1 = mean(c1)
MTB > let k2 = mean(c2)
MTB > let k3 = k1 - k2
MTB > print k1 k2 k3
K1      7.51429
K2      7.55714
K3      -0.0428576
```

The difference in time to first reproduction is 0.043 days. This is our statistic St = _____

What is this, in hours ? _____hr

Could this difference be merely a matter of chance? To find out we randomize the data, to compute the difference in time to reproduction due to chance sampling of the data.

Generic recipe for calculating random difference of means

Pseudocode (applies to any package)
Stack variables *X1* and *X2* into variable *X3*
Sample 7 values from variable *X3* to random variable *Xr1*
Sample 7 values from variable *X3* to random variable *Xr2*
Select a location to place *AvDiff(Xr1 −Xr2)*
Define the function *AvDiff(Xr1 −Xr2) = mean(Xr2) − mean(Xr2)*.
Calculate *AvDiff(X1 −X2) = mean(X2) − mean(X2)*.

***Sample to Xr1, Xr2***

***Calculate statistic AvDiff(X1,X2)***

Now that you have looked at the pseudocode, compute a random difference in mean age between the two strains of *Daphnia*, using the data from Srbx9_5.dat.

```
Minitab menu.
        Manipulate
                Stack
                        Stack columns                                   Sample to Xr1
                                Stack the following c1 c2
                                Column c3
        Calculate
                Random data
                        Sample from columns
                                Sample 7 rows                           Sample to Xr2
                                From c3
                                Store samples in c4
                                With replacement
        Calculate
                Random data                                             Calculate statistic
                        Sample from columns                             AvDiff(X1,X2)
                                Sample 7 rows
                                From c3
                                Store samples in c5
                                With replacement
        Calculate
                Calculator
                        Store result in k3
                        Expression = mean(c4) - mean(c5)
        Manipulate
                Stack
                        Stack columns
                                Stack the following c6 k3
                                Column c6
```

***Sample to Xr1, Xr2***

***Calculate AvDiff(Xr1, Xr2)***

```
MTB > stack c1 c2 c3
MTB > sample 7 c3 c4;
SUBC> replace.
MTB > sample 7 c3 c5;
SUBC> replace.
MTB > let k1 = mean(c4)
MTB > let k2 = mean(c5)
MTB > let k3 = mean(c4) - mean(c5)
MTB > print k1 k2 k3
  K1      7.48571  <---your values may differ
  K2      7.5857   <---
  K3      -0.1     <---
MTB > stack k3 c6 c3
```

The difference in time to reproduction, based on a single randomization, was 0.1 day, or about 2 hours earlier out of 7.5 days. Your value will differ because it is a different sample of the data. This value is the difference in time to reproduction that can arise due to chance sampling of the data.

To assign a probability to the observed difference, *Diff(X1,X2)* = −0.04 days or 1 hour, we will need to accumulate many chance differences. To accumulate chance differences, we define a sequence of commands that we can execute repeatedly.

Generic recipe for calculating random difference of means repeatedly.

Pseudocode (applies to any package)
- Define a sequence of commands
- Place into a file (sometimes called a macro)
- Execute the file

***Define macro***

***Execute macro***

Now that you have looked at the pseudocode, define and execute a macro to calculate the difference in mean age between the two strains of *Daphnia*, using the data from Srbx9_5.dat.

Minitab session.
- Highlight (select) the sequence of commands that produced the random difference

Minitab menu
- Edit
  - Command line I/O (commands should appear in box)
- Submit

***Define macro***

***Execute macro***

```
MTB > store 'srbx9_5.ctl'
STOR> sample 7 c3 c4
STOR> sample 7 c3 c5
STOR> let k1 = mean(c4)
STOR> let k2 = mean(c5)
STOR> let k3 = mean(c4) - mean(c5)
STOR> stack c6 k3 c6
STOR> end
MTB > execute 'srbx9_5.ctl'
```

***Define macro***

***Execute macro***

You should be able to see the additional random difference accumulate in the appropriate column. Now keep repeating this until you have accumulated at least 100 random differences.
Depending on the package you are using and its capabilities, you can do this by
repeating the ***Define macro, Execute macro*** sequence 100 times
running a loop with 100 repetitions
storing your macro in a file that is run repeatedly

Here is a command that will run the routine in the control file 100 times.

```
MTB > execute 'srbx9_5.ctl' 100
```

Now that we have a distribution of outcomes (when the hypothesis of no difference is true), let's look at this distribution. All statistical packages have a readily available routine for computing and displaying a histogram.

Generic recipe for creating and displaying a histogram

```
Pseudocode (applies to any package)
    Name the variable
    Apply histogram routine.
```

***Display histogram***

```
Minitab spreadsheet
        Select name box of variable, type in a name
Minitab menu.
        Calculate
                Descriptive
                        Histogram
```

***Define macro***

***Execute macro***

```
MTB > name c6 'randiff'
MTB > histogram c6
```

***Display histogram***

Now that we have a distribution, we can compare it to our outcome (St = `-0.0428576`) to obtain a probability of that outcome. The probability P(|X| < St) is the proportion of the distribution that is more extreme than the observed value St = `-0.0428576`. We count the number of values in both tails, the left tail N(X < St) and the right tail N(X > St). We add the tails and then express this as a proportion of the number of observations in the distribution.

Generic recipe for calculating a probability P(|X| < St) from histogram

| Pseudocode (applies to any package) | |
|---|---|
| Sort values | ***Sort to new column*** |
| Find closest value to St in left tail (small values) | |
| Count the number of values less than St | ***N(X < St)*** |
| Find closest value to St in right tail (large values) | |
| Count the number of values greater than St | ***N(X > St)*** |
| Add the counts, divide by the number of values in the histogram | ***P(X > St)*** |

You should be able to do this now, by finding the sort command, and putting the sorted values in c3 into a new column. You don't have to count values from the top or the bottom. Each row in the spreadsheet is numbered, so you can note the row number of the value nearest St, to obtain the count in both tails.

Your Name _____

These Minitab calculations above provide enough information to compute the following:

$N$ = Number of randomized differences = _____

$n_{neg}$ = Number of randomized differences more negative than −0.043 = _____

$n_{pos}$ = Number of randomized differences more positive than +0.043 = _____

Now compute the p-value for the observed outcome, a difference of −0.043 hours.

% of the outcomes less than the observed outcome of −0.043 = $\frac{n_{neg}}{N}$ = _____

% of the outcomes greater than + 0.043 = $\frac{n_{pos}}{N}$ = _____

% of the outcomes that were <u>either</u> greater than +0.043

<u>or</u> less than −0.043 = $\frac{(n_{neg} + n_{pos})}{N}$ = _____

This is your estimate of the p-value under the null hypothesis that the true difference is zero (i.e., a two-tailed test). The reason for computing a two-tailed test is that at the outset we had no idea whether strain I would reproduce earlier or later than strain II. Hence the need to calculate the probability of obtaining ***either*** a negative ***or*** a positive difference in timing. Your p-value should be close to the theoretical value (p = 0.908) calculated from an F-distribution with 1 and 12 degrees of freedom ($F_{1,12}$ = 0.014). This suggests that we could have used the Minitab command MTB>cdf to compute the p-value, as in the previous lab. The computation of this F-ratio ($F_{1,12}$=0.014) will be explained later in the course, as will the assumption for using the cdf command.

Place your frequency distribution of randomized differences for the *Daphnia* data in the space below. Label both axes of the frequency distribution.

## Write-up for lab #4.

Name _____

1. Complete previous page.

2. Carry out a randomization test for data in Box 13.11 of Sokal and Rohlf (1995), which shows mean litter size of two strains of Guinea pig, compared over 9 years.

   Assign a symbol to your statistic (difference of means) _____

   State the observed value of your statistic: _____ = _____

   Report your p-value _____

   Place your frequency distribution of at least 300 randomized outcomes in the space below. Be sure to label both axes of your graph.

Laboratory #4. Randomization tests

# Laboratory #5.
# Evaluating Graphs and Tables

Graphs are an effective way of presenting data and of communicating quantitative relations. Tables can also be used to present data effectively, although their main use is for archiving data for computations. Not all graphs and tables are as effective as they could be, and some graphical and tabular presentations can mislead more than inform.

The purpose of this lab is to give you practice in:

1) looking critically at graphs and tables;

2) making editorial improvements in graphs and tables.

3) developing your own general rules for good graphics

The approach will be to form up into groups of 3 or 4 people to discuss displays from a "Gallery" of graphs and tables.

**Table 5.1.** Here are some ideas for working effectively in groups.

| | |
|---|---|
| 1. | Agree on definition of task. |
| 2. | Everyone must contribute, so that the most voluble person does not dominate the group to the detriment of the shy. |
| 3. | Relevance: stay on the topic. |
| 4. | Listen and try to understand others. |
| 5. | Respond to others' comments. "I like that" "I partly agree and partly disagree." |
| 6. | Sum up. |

The 6 figures in group 1 of the gallery show several common problems with graphical presentations of data. The even numbered figures show several ways of improving graphical presentation. As you compare Figure 2 with Figure 1, try stating why Figure 2 is more effective at communicating results than Figure 1. Also, see if you can identify a problem that still remains in Figure 2 (hint: look at the improvements in Figure 6).

To carry out this lab, choose a display from the "Gallery," discuss its design and execution, and make a quick list of good and bad points.

Here is a checklist to start your discussion, but try not to limit yourself to this list.

**Table 5.2.** Checklist for evaluating graphs and tables.

| |
|---|
| **Graphs** |
| Are the axis labels and titles (or captions) adequate? |
| Are units stated for x and y axes? |
| Have appropriate symbols or lines been used? |
| Does the graph have freak characteristics, such as uninformative decoration? |
| Does the graph convey a story (bring out relations between variables)? |
| Does the graph mislead the reader in any way? |
| Can the data in the graph be pulled off accurately as numbers? |
| Are there variables that could be added to help interpret the trends shown? |
| **Tables** |
| Are row labels, column labels, and captions adequate? |
| Can rows and columns be regrouped or rearranged to facilitate comparison? |
| Would the addition of statistics (sums, deviations, etc) help? |

In looking at each graph or table ask yourself:

-- Is the relation of one variable to another quickly and easily grasped?

-- Can the information in the graph be translated into numbers? (could you pull the numbers off the graph and redraw it yourself?)

As you look at each display, think of one or two changes that would make the most improvement. Formulate a specific rule for improving that display. For example, a rule for improvement might be:

Put units on axes (display 3)

EACH PERSON in a group should write down the rule (or rules) followed by the display number.

As you accumulate specific rules, try to group them under general rules. For example:

A. Arrange rows and columns of tables in a way that facilitates comparison.
   1. Place comparable columns next to one another (display 47).
   2. Arrange rows into groups to bring out relationships (display 51).

After your group has discussed 5 to 10 displays, you should trade places with someone in another group. This is important because it increases the movement of ideas about criticizing and improving graphs and tables throughout the lab, exposing you to a greater range of experience and ideas about graphical and tabular displays. Keeping the same discussion group will tend to isolate the spread of good rules for improving displays. Moving to new groups will increase what you can learn from this lab, increase your list of good rules, and improve your write-up of this laboratory. During the course of the lab try to participate in at least 3 groups and try to evaluate at least 35 displays.

In general you will find that it is more productive to cover many displays, rather than concentrating on an exhaustive analysis of a few displays. If your group finds a display that only 1 or 2 people have seen, then let the people who have not seen the display discuss the graph for a minute or so before showing them what another group decided. They might find something else to improve in that display! But don't spend too much time on any one display. The idea is to gain as wide an experience as possible in as short a time as possible. Try to gain as much experience as you can by using both discovery (ideas generated by seeing a new display) and exchange (ideas generated from previous examinations by other groups).

## Write-up for this lab

1. Group effort.

Develop a list of at least 4 general rules for improving graphs, and 2 rules for improving tables. Record the display number (or numbers) after each rule. Arrange these rules in a hierarchical order by grouping specific rules under general rules. State specific rules as concisely as possible. Separate rules for graphs and tables:

I Graphs

A. General rule . . . . .
    1. Specific rule . . . . . (Display ___ )
    2. Specific rule . . . . . (Display ___ )
B. General rule . . . . .
    etc

II Tables

A. General rule . . . . .
    1. Specific rule . . . . . (Display ___ )
    2. Specific rule . . . . . (Display ___ )
B. General rule . . . . .
    etc

2. Individual effort.

Comment on which problems are serious, and which are less so.
State reasons or criteria for judging that some problems are serious, some are not.

# Laboratory #6.
# The General Linear Model: Regression

The purpose of this laboratory is to give you practice in using the General Linear Model in the analysis of data. The General Linear Model includes many well-known tests as special cases (ANOVAs, t-tests, regressions, and analysis of covariance ANCOVA).

The General Linear Model can be used in either an exploratory analysis (What is the best model?), or in a confirmatory analysis (What can we conclude, given a model?). The examination of residuals is an important part of the execution of the General Linear Model, in either an exploratory or confirmatory analysis.

In exploratory analysis, we can examine residuals in order to diagnose whether our description of pattern in the data (the formal model) is adequate. If the residual versus fit plot shows a uniform horizontal band, with no bowls or arches, then we have arrived at an adequate description of pattern in the data. If the residuals do show pattern, then we can use observed patterns in the residuals to construct a better guess about pattern in the response variable relative to explanatory variables.

In confirmatory analysis, we can examine residuals in order to diagnose whether our data meet the assumptions for computing p-values using a theoretical distribution such as the F-distribution. p-values calculated from F, t, or chi-square distributions cannot be trusted if the ***residuals*** are correlated, heterogeneous, or non-normal. Some people think the data have to be "normal" before undertaking analysis, but this is incorrect.

Once you have completed the lab write-up, you should have

- capacity to undertake regression analysis using either the GLM or regression routine in a statisical package.

- a working knowledge of the mechanics of residual analysis in a statistical package

- capacity to decide when to use a randomization to compute a p-value

- a working knowledge of the mechanics of computing a randomized p-value for a general linear model.

At this point make sure that you have three data sets:.

| | |
|---|---|
| *Tribolium* weights | Srbx14_1.dat |
| *Tribolium* survival | Srbx14_4.dat |
| Cod mortality rate | Garrod.dat |

They can be found on the Biology Department server.

www.mun.ca/biology/schneider/b4605/data

**Analysis #1. *Tribolium* weight in relation to humidity. Regression routine.**

Most statistical packages contain separate routines for regression and for the general linear model. Analysis #1 demonstrates the general linear model and residual analysis with a regression routine. The example comes from a Box 14.1 in the text by Sokal and Rohlf (1995).. The research questions are:
Does weight loss in the flour beetle *Tribolium* depend on humidity ?
If it does, what is the functional relation of weight loss to humidity ?

When you define the data from the file Srbx14_1.dat into the package, name the first column 'wloss' and the second 'humidity'. If you need to recall the code for this generic command check the Guide to Computer Use, elsewhere in this manual.

***Define Data from file***

Generic recipe for regression of Y-variable against X variable

***Run regression with Residual diagnostics***

Pseudocode (applies to any statistical package)
- Define the response variable, Y.
- Define the explanatory variable X.
- Save residuals and fitted values.
- Plot residuals vs fitted values to check linear assumption.
- Revise model if linear assumption not met.
- Check p-value assumptions of homogeneous normal residuals.

Now that you have looked at the pseudocode, find the regression command in your statistical package and run it on the *Tribolium* data.

***Run regression with Residual diagnostics***

Minitab menu.
- Statistics
  - Regression
    - Regression
      - Response ‘wloss’
      - Predictor ‘humidity’
      - Graphs
        - Histogram
        - Normal plot
        - Residuals vs Fits
        - Residuals vs order

***Run regression***

```
MTB > plot 'wloss' * 'humidity'
MTB > regress 'wloss' 1 'humidity';
SUBC> fits c4 ;
SUBC> residuals c5.
MTB > name c4 ‘fits’ c5 'res'
```

The line code for residual diagnostics will be shown later.

**Analysis #1 (continued).**

When we use the general linear model (as in regression) we make assumptions about the structural part of the model (Y hat) and about the error distribution.

We will begin with the assumptions about the structural part of the model $Y = \hat{Y} + error$ model (group A), then move to the assumptions about the error distribution (group B).

Assumption A1. A straight line model is appropriate.
To check this assumption we look at the residual vs fit plot.
Most statistical packages now produce this plot automatically, as an option for regression output.

```
MTB > plot 'res' * 'fits'
```

***Model linear?***

There are no obvious bowls or arches (Assumption A1 met).
We accept the linear model. Write the regression equation here, immediately below the model.

$\hat{Y} = \alpha + \beta_x X$

$Wloss = \_\_\_ + \_\_\_ Humidity$

Statistical packages report results in this familiar slope intercept form, as above.
Compare this to the statistical model, which use $\beta_o$ instead of the Y-intercept α.

$$Y = \hat{Y} + error$$

$$Y = \beta_o + \beta_x (X - \overline{X}) + error$$

The Y-intercept α is computed from $\beta_o$ by the statistical package. Try this calculation yourself.

$\overline{Y} = \beta_o = \_\_\_\_\_$

$\overline{X} = \_\_\_\_\_$

$\alpha = \beta_o - \beta_x \overline{X} = \_\_\_\_\_$

Next we move to the error component of the statistical model. The p-values reported for the two parameters α and $\beta_x$ depend on four assumptions about the errors.

Assumption B1: The errors sum to zero. Statistical packages estimate parameters in a way that makes this true, so we don't need to check this assumption.

The graphics interface version of many packages (including Minitab) will produce diagnostic plots on residuals as options. Only the line code version of the residual diagnostics will be shown below.

**Analysis #1 (continued)**

Assumption B2: The errors are independent of one another. One way to check this assumption is to plot the errors in the order in which the data were collected. Another way is to plot each residual against a neighboring value. If your package makes these computations, check this assumption now.

```
MTB > let c20 = lag('res')
MTB > plot c20 * 'res'
```

***Errors independent?***

The regression residuals from the analysis of *Tribolium* data show no obvious trends, so residuals are taken to be independent.

Assumption B3: The errors are homogeneous. This is the most important assumption. Violations of this assumption will have the greatest biassing effect on the p-value. This assumption is checked by going back to the residual vs. fit plot we examined before. This time we are looking for cones expanding either to the right (which is common) or left (rare). We are looking for really strong patterns, not just slight tendencies toward cones.

```
MTB > plot 'res' * 'fits'
```

***Errors homogeneous?***

Do you see any strong cones in the residual vs fit plot for the *Tribolium* data ? _____

Assumption B4: The residuals are normal. This assumption tends to attract more attention than the homogeneity assumption, when in fact violations of the normality assumption have less of a biassing effect on the p-value. The two commonest ways to check this assumption are with a histogram and with normal scores.

Assumption B4 (normality) checked with a histogram.
If the assumption is met the residuals will cluster around their mean (zero). If the assumption is not met the histogram will be strongly skewed. Evaluation of the histogram will be difficult when there are few residuals, as in the *Tribolium* data. Many packages produce this diagnostic as an option. Here is the line code version.

```
MTB > histogram 'res'
```

***Errors normal?***

The visual impression from a histogram can depend very much on the number of classes used to construct the histogram. If you package will allow, replot the histogram with fewer classes.

```
MTB > histogram 'res' ;
SUBC> ninterval 4.
```

***Errors normal?***

Now the histogram looks slightly leptokurtotic, with heavy tails due to a few rather large errors.

**Analysis #1 (continued)**

Assumption B4 (normality) checked with normal scores.
We can check the normality assumption by computing normal scores for each residual, and then plotting the normal score against the residual. The result is a straight line rising diagonally, if the residuals are normal. The plot becomes twisted and S-shaped if the residuals deviate from normality. Many statistical packages produce this plot as an option. Here is the line code.

```
MTB > nscores  'res'  c21
MTB > plot c21 * 'res'
```

***Errors normal?***

The plot is not a straight line for the *Tribolium* data. The residuals deviate from normal.

The p-value calculated by Minitab, in the ANOVA table assumes that errors are independent, homogeneous, and normal. The assumptions were not met, in the analysis of the *Tribolium* data. We have already learned the remedy: recompute the p-value by randomization. This produces a p-value we can trust, without the assumptions required for the F-distribution. We also know it is a lot of work. So we ask at this point whether the extra work is required. If the p-value is close to the criterion for significance ($\alpha = 5\%$) then an incorrect p-value can lead to an erroneous decision. But if the p-value is far from the criterion (say by a factor of 5 or a factor of 1/5) then the extra work is not worth the effort. No matter how badly the assumptions are violated, p-values from the F-distribution do not deviate from the randomization p-value by factors as large as 5 or 1/5. For the *Tribolium* data the F-statistic is huge ($F = 267$) and hence the p-value is minuscule ($p < 0.001$). If we recompute the p-value via randomization we get a more accurate p-value, we put a lot of time into the effort, but the decision (that the slope of the regression is not zero) will not change. So for the *Tribolium* data we are not going to put time into an effort that won't change our decision. We will reject the null hypothesis in favour of the alternative hypothesis, that weight loss depends on humidity. We will also accept the regression equations with the parameters produced by the statistical package.

**Analysis #2. *Tribolium* weight in relation to humidity. GLM routine.**

Any linear regression can be carried out using the GLM routine instead of the regression routine. To acquaint yourself with the GLM routine in your package you should go back to the beginning of the analysis of the *Tribolium* data and repeat it with the GLM routine. This should not take more than about 5 minutes, as it is very similar in structure and output.

***Define Data from file***

Here are the specifics for the GLM routine in Minitab.

***Run regression with Residual diagnostics***

```
Minitab menu.
    Statistics
        ANOVA
            General Linear Model
                Response 'wloss'
                Model  'humidity'
                Covariates 'humidity'
                Graphs
                    Histogram
                    Normal plot
                    Residuals vs Fits
```

***Run regression***

```
MTB > plot 'wloss' * 'humidity'
MTB > glm 'wloss' = 'humidity';
SUBC> covariate 'humidity';
SUBC> fits c4 ;
SUBC> residuals c5.
MTB > name c4 'fits' c5 'res'
```

The line code version of GLM stores residuals in the same was as the line code version of Regression. Residual diagnostics are thus executed with the same commands after the GLM as after the regression routine.

After you have run the GLM routine make the following comparisons.

_____ Are the estimates of the Y-intercept $\alpha$ and the slope $\beta_x$ the same ?

_____ Is the ANOVA table the same ?

_____ Is the residual vs fit plot the same ?

_____ Is the normal score plot the same ?

**Analysis #3. *Tribolium* survival in relation to egg density. GLM routine.**

The next example demonstrates the analysis of more than one Y-value for each X-value. The data set will be another text example, taken from Box 14.4 in Sokal and Rohlf (1995). The research question is, does *Tribolium* survival depend on density?

At this point it is a good idea to close your previous session, saving any work you need, then start a new session. Begin by opening the data file Srbx14_4.dat, examining its structure. and then defining the data from this file to the statistical package. Read in both survival and arcsin(survival), as well as density.

***Define Data from file***

Generic recipe for regression of Y-variable against X variable

***Run regression with Residual diagnostics***

Pseudocode (applies to any statistical package)

Define the response variable, Y = % survival
Define the explanatory variable X = density
Save residuals and fitted values.
Plot residuals vs fitted values to check linear assumption.
Revise model if linear assumption not met.
Check p-value assumptions of homogeneous normal residuals.

Use the GLM command to carry out the analysis.

***Run GLM regression with Residual diagnostics***

Minitab menu.
Statistics
ANOVA
General Linear Model
Response 'survival'
Model 'density'
*Etc.*

***Run GLM regression***

```
MTB > glm 'survival'  =  'density';
SUBC> covariate 'density';
SUBC> fits c4;
SUBC> residuals c5.
MTB > name c4 'fits'  c5 'res'
```

_____Is the straight line assumption valid ? (any bowls or arches in residual vs fit plot?)

_____ Are the residuals homogeneous ? (cones in residual vs fit plot?)

_____ Are the residuals normal ? (use histogram and normal scores)

**Analysis #3. (Continued).**

The response variable is a percentage, and hence we might expect the residuals to be non-normal. Instead, we found that the residuals were normal but not homogeneous.
The arcsin transformation is usually recommended for response variables that are percentages. Redo the analysis, using the column of arcsin transformed data (you should now get the same ANOVA table as in Box 14.4 of the Sokal and Rohlf 1995 text).

For the analysis of arcsine transformed survival:

_____Is the straight line assumption valid ? (any bowls or arches in residual vs fit plot?)

_____ Are the residuals homogeneous ? (cones in residual vs fit plot?)

_____ Are the residuals normal ? (use histogram and normal scores)

_____ Did the arcsin transform do anything to meet the assumptions ?

What has the analysis of residuals told you about automatically using the arcsin transform with response variables expressed as a percentage ? _____

The assumptions were not met, and hence the p-value (with or without arcsin transform) from the F-distribution may not be correct. As before, we now ask whether it is worth taking the time to carry out a randomization test.

_____ Is the p-value close to $\alpha = 5\%$ ?

_____ Is the decision based on this p-value likely to change if we obtain a more accurate p-value by randomization ?

Extra. ANOVA tables and F-ratios for regression with several Y-values for each X-value.

The recommended procedure with several Y-values is to form the F-ratio based on all of the data, rather than based on just the means for each group (Freund 1971). Now that you have done this, try carrying out the regression analysis with just the group means. You will need to include group size = n in the the analysis, as weights. If you use the means for arcsin transformed data, you should be able to reproduce the ANOVA table in Box 14.4 of Sokal and Rohlf 1995.

**Analysis #4. Cod mortality in relation to fishing effort. GLM routine.**

The next example shows the application of a randomization test to regression. The data are cod mortality rates in relation to fishing effort, reported by D.J. Garrod in the 1960s. Begin by opening the data file Garrod.dat, examining its structure. and then defining the data from this file to the statistical package.

| |
|---|
| |

***Define Data from file***

Generic recipe for regression of Y-variable against X variable

| |
|---|
| Pseudocode (applies to any statistical package)<br>Define the response variable, Y = mortality<br>Define the explanatory variable X = effort<br>*Etc.* |

***Run GLM regression with Residual diagnostics***

_____Is the straight line assumption valid ? (any bowls or arches in residual vs fit plot?)

_____ Are the residuals homogeneous ? (cones in residual vs fit plot?)

_____ Are the residuals normal ? (use histogram and normal scores)

_____ Is the p-value close to $\alpha = 5\%$ ?

_____ Is the decision based on this p-value likely to change if we obtain a more accurate p-value by randomization ?

To compute a better p-value, we keep track of how many randomized F-ratios exceed the observed F-ratio ($F_{observed} = 4.91$). We will do this by hand, to illustrate the procedure.

**Table 6.2.** Randomization tests, tallied by hand.

1. Write down the observed F-ratio, from the ANOVA table.
2. Randomize the data.
3. Compute a randomized F-ratio.
4. Keep a tally of the number of F-values less than or equal to $F_{observed}$, and the number greater than $F_{observed}$.

**Analysis #4 (continued)**

Here is a generic recipe for carrying a randomization test, using the GLM.

Generic recipe for randomization tests with GLM

Pseudocode (applies to any statistical package)
- Sample from Y variable into a new column = 'Yrandom'
- Regress 'Yrandom' against the explanatory variable X
- Record the F-ratio.
- Repeat, and tally the number of F-ratios greater the $F_{observed}$
- Divide by the number of F-ratios

***GLM p-values by randomization***

Here are the specifics for obtaining a single randomized F-ratio in Minitab.

Minitab menu.
- Calculate
  - Random data
    - Sample from columns
      - Sample 13 rows
      - From column (mortality)
      - Store in c7

Name c7 'RanMort'
- Statistics
  - ANOVA
    - General Linear Model
      - Response 'Ranmort'
      - Model 'effort'
      - *Etc.*

```
MTB > sample 13 'mort'  [into] 'Ranmort'
MTB > glm c7 = 'effort';
MTB > covariate 'effort'.
```

Be sure to tally this randomized F-ratio. Here is a Table to use.

| # ≤ 4.91 | # > 4.91 |
|---|---|
| | |

Record tallies only, you do not need to record the value of each F-ratio.

**Analysis #4 (continued)**

Next, we want to generate many of these randomized F-ratios. Here are the specifics.

Minitab session. Highlight (select) code producing random F-ratio
Minitab menu
    Edit
        Command Line (selected code should appear)
        Submit

```
MTB > store 'Garrod.ctl'
STOR> sample 13 'mort'  [into] 'Ranmort'
STOR> glm c7 = 'effort';
STOR> covariate 'effort'
STOR> end
MTB > execute 'Garrod.ctl'
```

Be sure to tally this randomized F-ratio in the table provided above.

Now run the batch file repeatedly and fill in the following tally sheet for 200 randomized F-ratios:

After 100 repeats, you will have a tally of the random F-ratios, above and below $F_{observed}$

Record the number (out of 100) that exceeded $F_{observed}$ _____

Record the <u>percentage</u> (out of 100) that exceeded $F_{observed}$ _____

This percentage is a rough estimate of the p-value for the randomization test.

If none of your randomized F-ratios were greater than 4.91, then your p value is less than 1 in 200 ($p < 1/200 = 0.005$). **It is not zero.**

To obtain a really good estimate of the p-value, more randomized F-ratios are needed. This will be a lot of work, unless we can use Minitab to accumulate randomized F-ratios for us. This is easier with line code than with a graphics interface. If you decide to use randomization tests you will need some code that can be run to accumulate randomization results automatically.

Here is a set of commands that will work in Minitab, in case you wish to use them after taking this course. The F-ratio is formed by calculating the mean squared error (MSE) for both the residuals and the fitted values. C22 is the randomized response variable. C21 is the regression (X) variable. Randomized F-ratios accumulate in c26.

```
MTB > glm c22 = c21;
SUBC> covariate c21;
SUBC> fits c24;
SUBC> residuals c25.
MTB > let c24 = c24 - mean(c22)
MTB > let k1 = ssq(c24)/1                 MSE for model
MTB > let k2 = ssq(c25)/(N(c22)-2)        MSE error
MTB > let k3 = k1/k2                      F-ratio
MTB > stack k3 c26 c26
```

## Write-up for this laboratory.

Please do not hand unlabelled computer output! Instead, cut sections of output to a document and label each section in the document. If you paste tabular input into your document, use a non-scalable font (`such as Courier 10`) to display this material. Otherwise the material will be distorted and hard to read.

Analysis #1

Use the generic recipe for hypothesis testing with the general linear model (see Handouts) to write up the analysis of the *Tribolium* data (srbx14_1.dat). Because these are exercises, you may not have enough information to identify the population, step 1. Include a plot of residual against fitted values with comments on whether the model is suitable for analysis. Include appropriate plots with comments on whether straight line is appropriate (assumption A1) and whether residuals are homogeneous and normal (assumptions B3 and B4).

Analysis #2

Write the regression equation based on parameter estimates from the GLM routine.
Show the ANOVA table from the GLM analysis of the *Tribolium* data (srbx14_1.dat).

Analysis #3

Present labelled residual plots for both survival and arcsin(survival) in relation to density. (Srbx14_4.dat).

Analysis #4

Use the generic recipe for hypothesis testing with the general linear model (see Handouts) to write up the analysis of the cod mortality data (Garrod.dat). Include appropriate plots with comments on whether straight line is appropriate (assumption A1) and whether residuals are homogeneous and normal (assumptions B3 and B4).

# Laboratory #7.
# The General Linear Model: Analysis of Variance

The purpose of this laboratory is to increase your familiarity with ANOVA (Analysis of Variance). ANOVA is a special case of the General Linear Model: the response variables are factors (on a nominal scale) with 2 or more classes within a factor. ANOVA compares mean values among these classes within a factor.

Because we are using the General Linear Model, much of what you learned in the previous lab (regression) carries over to ANOVA. We will use the same steps to analyze the data and then carry out an analysis of residuals.

Once you have completed the lab and write-up, you should have

- the capacity to re-organize data from tabular to model format

- greater facility in the diagnosis of residuals

- a working knowledge of ANOVA in the model format

At this point make sure that you have two data sets:

| | |
|---|---|
| srtab8_1.dat | (fly wing lengths, Table 8.1 in Sokal and Rohlf, 1995) |
| fishmov.dat | (cod movements, by hour of day) |

Both are available on the server in Biology: www.mun.ca/biology/Schneider/B4605/data

**Analysis #1 ANOVA in tabular format.**

The data used in ANOVA are usually represented in a tabular fashion. For a single factor, the data for each class are typically listed in separate columns. Early versions of statistical packages read data according to this tabular format, then carried out the analysis assuming that a column of data was a class within a factor. Routines based on tabular input are still used. The first analysis demonstrates this approach to ANOVA.

The data come from Table 8.1 in the text by Sokal and Rohlf (1995), thus tying the output from the statistical package to the explanatory material in the text. The research question is:
Does fly wing lengths differ among cages ?

**Analysis #1 ANOVA in tabular format**

Begin by opening the data file SrTab8_1.dat, examining its structure. and then defining the data from this file to the statistical package. Bring in all 7 columns, representing 7 cages. You do not need to label each column.

If you need to recall the code for this generic command check the Guide to Computer Use, elsewhere in this manual.

| | |
|---|---|
| | ***Define Data from file*** |

Generic recipe for oneway ANOVA in tabular form.

| | |
|---|---|
| Pseudocode (applies to any statistical package)<br>Place data for each class in a separate column.<br>Find the command for oneway ANOVA in tabular format.<br>Carry out analysis on columns (1 for each class within a factor) | ***Run tabular ANOVA*** |

If you can't find a tabular format ANOVA in your statistical package, skip to **Analysis #2**.

Here are the specifics for Minitab

| | |
|---|---|
| Minitab menu<br>Statistics<br>ANOVA<br>One-way (Unstacked)<br>c1-c7 | ***Run tabular ANOVA*** |

| | |
|---|---|
| `MTB > aovoneway c1-c7` | ***Run tabular ANOVA*** |

**Analysis #2. ANOVA in model format. Fly winglengths.**

ANOVA in model format allows use several factors easily, rather than just 1 or 2 factors as in tabular format. The model format allows far greater flexibility in the design, including choice over interaction terms when there is more than one factor. The model format was rare 20 years ago, but is widely used today.

To use the model format, we will need to re-organize tabular data. We are going to stack all of the values of the response variable into a single column, then place labels next to each value, in the next column. The second column, consisting of labels, will be the explanatory variable.

**Analysis #2 (continued)**

Here is a generic recipe (pseudocode) for re-organizing data from tabular to model format.

```
Read tabular data into the packages spreadsheet, one column per class
Insert a new column into the spreadsheet.  Assign it a name (Y)
Insert a new column adjacent to Y.  Assign it a name (factor1)
Copy the first tabular column, paste it to Y
For each value in Y, place the numeral 1 in the adjacent row of
factor1
Copy the second tabular column, paste to Y.
For each new value of Y, place the numeral 2 in the adjacent row.
Continue for all of the tabular format columns.
```

***Reorganize from tabular to model format***

You should be able to carry this out in any statistical package with a spreadsheet for data.

Here is the line code in Minitab.

```
MTB > stack c1-c7 c8.
SUBC> subscripts c9
MTB > name c8 'wlength'c9 'groups'
MTB > print 'wlength' 'groups'.
MTB > print c1-c9.
```

***Reorganize from tabular to model format***

Now the analysis.  Here is the generic command for model based analysis.

```
Pseudocode (applies to any statistical package)
    Define the response variable, Y = wlength
    Define the explanatory variable X = groups
    Save residuals and fitted values.
    Check p-value assumptions of homogeneous normal residuals.
```

***Run GLM ANOVA with Residual diagnostics***

Here are the specifics for ANOVA routine in Minitab

```
Minitab menu
    Statistics
        ANOVA
```

***Run GLM ANOVA with Residual diagnostics***

**Analysis #2 (continued).**

Here are the specifics for ANOVA routine in Minitab line code.

```
MTB > anova 'wlength' = 'groups';
SUBC> fits c10;
SUBC> residuals c11.
MTB > name c10 'fits'  c11 'res'
```

***Run GLM ANOVA***

Here are the specifics for the GLM routine in Minitab

Minitab menu
- Statistics
  - ANOVA
    - General linear model
      - Response = Y variable
      - Model = X variable (can be several)
      - Graphs
        - Histogram
        - Normal plot
        - Residual vs fit

***Run GLM ANOVA with Residual diagnostics***

```
MTB > glm 'wlength' = 'groups';
SUBC> fits....;
SUBC> res .....
```

***Run GLM ANOVA***

**Analysis #2 (continued)**

The model based format uses data equations, which are easy to construct for oneway ANOVA.

Here is the generic command for constructing the data equations, oneway ANOVA.

Pseudocode (applies to any statistical package)
Place all values of the response variable in a single column Y
Calculate the mean for each cell (class within a factor)
Adjacent to each value place the mean for its cell
Label this column 'fits'
In a third column, calculate Y - 'fits'
Label this column 'resids'

***Compute Data equations***

With a graphics interface, you can use the cut and past functions to accomplish this.

Here is the line code to print the three columns showing data equations.

```
MTB > glm 'wlength' = 'groups';
SUBC> fits 'fits';
SUBC> res 'res'.
MTB > print 'wlength' 'fits' 'res'.
```

Pick a winglength observation and write a data equation.

Data = Model + Residual

_____ = _____

One of the advantages of the model-based style of analysis is that it allows the residuals to be examined. The assumptions concerning the residuals are the same for any GLM, so we will use exactly the same procedure as we did with regression.

Assumption B1. Errors sum to zero. Statistical packages estimate parameters in a way that makes this true, so we don't need to check this assumption.

Assumption B2: The errors are independent of one another. One way to check this assumption is to plot the errors in the order in which the data were collected. Another way is to plot each residual against a neighboring value. If your package makes these computations, check this assumption now.

```
MTB > let c20 = lag('res')
MTB > plot c20 'res'
```

***Errors independent?***

The residuals show no obvious trends, so residuals are taken to be independent.

**Analysis #2 (continued)**
Assumption B3: The errors are homogeneous. You will recall that this is the most important assumption. Violations of this assumption will have the greatest biassing effect on the p-value.

```
MTB > plot 'res'  'fits'
```

***Errors homogeneous?***

Do you see any strong cones in the residual vs fit plot ? _____

Assumption B4 (normality) checked with a histogram.
If the assumption is met the residuals will cluster around their mean (zero). If the assumption is not met the histogram will be strongly skewed.

```
MTB > histogram 'res'
```

***Errors normal?***

Some packages will compare the fit of the histogram to a normal distribution.

```
MTB > rootogram 'res'
```

***Errors normal?***

Assumption B4 (normality) checked with a normal scores.
Normal residuals will fall along a straight line rising diagonally.
The plot becomes twisted and S-shaped if the residuals deviate from normality.

```
MTB > nscores  'res'  c21
MTB > plot c21 * 'res'
```

***Errors normal?***

If the residuals are independent, homogeneous, and normal then the p-value calculated for this analysis (from an F-distribution) can be trusted. A decision can be declared, based on this p-value.

This represents a thorough analysis of the data on housefly wing lengths, including checks an appropriateness of the use of the theoretical F-distribution, which assumes that residuals are homogeneous and normal.

Table 7.1. Diagnosing residuals for normality.

| |
|---|
| The histogram of the residuals should look like a normal distribution. |
| The fit of the histogram to a normal distribution should be acceptable. |
| The normal scores (nscores) of the residuals will fall along a straight line, when plotted against normally distributed residuals. |

**Analysis #3. ANOVA in model format. Cod movements.**

The next data set, in file 'fishmov.dat' was collected by Don Clark, a graduate student at Memorial University. The data are used with his permission. The research question was whether movements of juvenile cod *Gadus morhua* depended on time of day. The data set illustrates typical problems encountered by students when they analyze their own data. The data are too voluminous to analyze by hand, and as you will see, there are substantial problems with the residuals. Hence p-values displayed by the statistical package (via the F-distribution) may not be correct. You will need to decide whether randomization is warranted, to correct the problem.

At this point you should close the session where you analyzed the fly winglength data, saving any material that you need later.

Open a new session or worksheet in your package to analyze the cod movement data.
Open the data file fishmov.dat, and examine its structure. Information about the structure is at the end (bottom) of the file. The information you need to identify Response and Explanary variables also occurs at the end of this file.

| | |
|---|---|
| | ***Define Data from file*** |
| Pseudocode (applies to any statistical package)<br>Define the response variable, Y = distance<br>Define the explanatory variable X = time of day<br>Save residuals and fitted values.<br>Check p-value assumptions of homogeneous normal residuals. | ***Run GLM ANOVA with Residual diagnostics*** |

## Write-up for this laboratory:

Please do not hand in entire outfiles! Instead, cut sections out of your outfiles to show your results. Use a non-scalable font (`Courier 10 or Courier 12`) to display results. Otherwise, the graphs produced by Minitab will be distorted and hard to read. If you prepare your lab reports by hand, then edit your outfile to what you need, print it out, cut it with scissors, and paste or tape these pieces onto your write-up to show **your** results.

The write-up consists of an Analysis of Variance for 2 data sets: **srtab8_1.dat, fishmov.dat**

The write-up for each of the 2 analyses should follow the generic recipe for hypothesis testing for the general linear model (see Handouts). Because these are exercises, you may not have enough information to identify the population (Step 1 of the generic recipe).

Be sure to plot residuals for both analyses. Step 8 (check assumptions) should include a histogram with comments on whether residuals are independent, homogeneous, and normal.

Name _____

# Laboratory #8.
# Applying the General Linear Model

The purpose of this lab is to increase your skill and facility in applying the general linear model to what the statistician John Tukey called "data situations." A data situation includes all of the accessory information that is typically associated with a set of measurements. For example, the designer of an experiment may already know that a convenient blocking factor, such as separate greenhouses, will have effects that are independent of treatments and controls. In this situation, the experimenter will carry out a randomized blocks design, without the replication needed to estimate the interaction term (treatments x blocks). Or to take another example, a physiologist may know that the effects of a drug on a rat can carry over several days, and so not use any single rat on successive days. The term "data situation" is not in wide use, but is valuable in distinguishing "data" (= a set of scaled numbers) from situational knowledge that accompanies any set of data.

In this lab you will be presented with 3 "data situations." In the first analysis you will shown how to construct a simple model based on the research question, which concerns guinea pig litter sizes (Box 13.11, Sokal and Rohlf 1995). You will then be shown how to complete the analysis based on the model. You will then step step through the analysis of a second data situation, which concerns lactic acid production in frog embryos(Box 11.7 in Sokal and Rohlf 1995). There is a missing value in the second situation, which leads to an unbalanced data that many routines in statistical packages cannot handle. The GLM command can handle unbalanced data. Based on what you learned from the first two situations, you will be asked to construct a model to analyze a third data situation, concerning growth rates of fish on experimental diets (exercise 14.12 from Sokal and Rohlf 1995).

Once you have completed the lab write-up, you should have

- ability to translate a data situation into a model-based analytic format

- ability to use a statistical package to execute any model you write

- greater facility in applying the general linear model to data situations with more than one explanatory variable.

At this point make sure that you have three data sets:

srbx11_7.dat
srbx1311.dat
srex1412.dat

Both are available on the server in Biology: www.mun.ca/biology/schneider/b4605/data

**Data situation #1**

The first example of the use of the General Linear Model command will be a re-analysis of the Guinea pig data from Box 13.11 in Sokal and Rohlf (1995).

| StrainB | Strain13 | Year |
|---|---|---|
| 2.68 | 2.36 | 1916 |
| 2.60 | 2.41 | 1917 |
| 2.43 | 2.39 | 1918 |
| 2.90 | 2.85 | 1919 |
| 2.94 | 2.82 | 1920 |
| 2.70 | 2.73 | 1921 |
| 2.68 | 2.58 | 1922 |
| 2.98 | 2.89 | 1923 |
| 2.85 | 2.78 | 1924 |

Assign a symbol to each variable:

The response variable is litter size = ____

The first explanatory variable is strain = ____

The second explanatory variable is year = ____

Write a general linear model relating the response variable to the two explanatory variables. Your model should have 1 term on the left, and 4 terms on the right in addition to $\beta_o$.

_____ = $\beta_o$ + _____

df: _____ = _____ + _____ + _____ + _____

Fill in the degrees of freedom, moving from left to right.
Do you have enough degrees of freedom to
estimate the interaction term in your model? _____

Re-write your model as a randomized blocks design, with interaction assumed to be zero.

_____ = $\beta_o$ + _____

df: _____ = _____ + _____ + _____

Fill in the degrees of freedom for your model.
The GLM command to partition the Sums of Squares $SS_{total}$ is:

MTB> glm 'lsize' = 'year' 'strain'

SS: _____ = _____ + _____ + _____

The commands on the next page will show you how to obtain the sum of squares partitioned according to your model.

**Data situation #1 (continued)** Name _____

Now open the Guinea pig data file (Srbx 13_11.dat) and examine its structure.
You will need to bring the data into the statistical package, and then reorganize it from tabular to model format. When you are done, you should have three columns variables: lsize, yr, strain.
Each column variable should have 18 rows.

Here is the generic recipe (pseudocode) for reorganizing data to model format.

***Define Data from file***

***Reorganize from tabular to model format***

***Run GLM ANOVA with Residual diagnostics***

```
Read tabular data into the package spreadsheet, one column per class
Insert a new column into the spreadsheet.  Assign it a name (Y)
Insert a new column adjacent to Y.  Assign it a name (factor X1)
Paste in the appropriate values of factor X1
Insert a new column adjacent to Y.  Assign it a name (factor X2)
Paste in appropriate values of factor X2
```

```
Pseudocode (applies to any statistical package)
    Define the response variable, Y = lsize
    Define the explanatory variable X1 = age     X2 = strain
    Save residuals and fitted values.
    Check p-value assumptions of homogeneous normal residuals.
```

You should be able to display the specific sequence of actions for the graphics interface in Minitab or the package you are using.
Here is the command line, as it appears in Minitab.

```
MTB > glm 'lsize' = 'yr' 'strain'
SUBC>     ;
SUBC>     .
```

If you don't see the command line in the output, find out how to force the package you are using to display the command line.

To review the concept behind this analysis, fill in both the df and SS below each term in the model expressed by the command line.

MTB> glm 'lsize' = 'year' 'strain'

df: _____ = _____ + _____ + _____

SS: _____ = _____ + _____ + _____

**Data situation #1 (continued)** Name _____

Tape a copy of the command line here, as displayed by your package.

Tape your ANOVA table here. If you decide to attach your ANOVA table to the end of the lab report, be sure it is adequately labelled (name of data file, name of the response variable).

Comment on whether the residuals are homogeneous and normal
If you attach your comments at end of report, state the name of input data file and response variable.

State whether you would trust the p-value computed by Minitab from an F-distribution, and why.
If you attach your comments at end of report, state the name of input data file and response variable.

**Data situation #2** Name _____

Next, we use the model based approach to analyse another randomized block experiment. The analysis will be the same as the previous one, except that this time a missing value creates unbalanced data that cannot be handled by many ANOVA routines. The GLM command will handle the unbalanced data.

| I | II | III | IV |
|---|---|---|---|
| 21.4 | * | 7.0 | 9.5 |
| 14.3 | 13.5 | 5.4 | 6.6 |
| 23.4 | 14.1 | 5.9 | 7.1 |
| 29.1 | 8.2 | 4.2 | 3.2 |
| 26.6 | 13.5 | 4.9 | 6.0 |
| 21.7 | 5.2 | 6.6 | 5.9 |

Data from Box 11.7 Sokal and Rohlf 1995

Lactic acid production in frog embryos from 4 clutches (4 columns) at 6 stages (6 rows) after 1st cleavage (0 min = 1st row, 360 = 2nd, 720 = 3rd, 1200 = 4th, 1600 = 5th, 2000 = 6th).

* = missing value.

The research question is whether lactic acid production depends on time after cleavage.

Symbol

The response variable is _____ = _____

The explanatory variable of research interest is _____ = _____

The remaining or "control" explanatory variable is _____ = _____

Write a general linear model that relates the response variable to both explanatory variables. As before, you should have 5 terms on the right side of the model.

_____ = $\beta_o$ + _____

As before, the interaction term in your model cannot be estimated.
Why not?

**Data situation #2 (continued)** Name _____

Now write another model, this time assuming there is no interaction term.
Compute the degrees of freedom for each term, then write the model format command

| | | | |
|---|---|---|---|
| | _____ | = $\beta_o$ + | _____ |
| df: | _____ | = | _____ + _____ + _____ |
| `MTB > anova` | _____ | = | _____ |
| `MTB > glm` | _____ | = | _____ |

This model examines the effect of whether lactic acid production depends on stage, controlled for differences in clutches. on the response variable, controlling for the second variable.

Next, set up the data for model-based analysis

Refer to Data situation #1 for the generic recipe (pseudocode) or specific sequence of actions

| | |
|---|---|
| Read tabular data into the package spreadsheet, one column per class<br>*Etc.* | ***Define Data from file*** |

Statistical packages handle missing values in different ways. Some packages treat * as missing. Other packages treat a blank as missing. Minitab treats blank as missing, so you will need to convert * to a blank.

| | |
|---|---|
| | ***Reorganize from tabular to model format*** |

| | |
|---|---|
| Pseudocode (applies to any statistical package)<br>Define the response variable, Y = Lap<br>Define the explanatory variable X1 = clutch X2 = stage<br>Save residuals and fitted values.<br>Check p-value assumptions of homogeneous normal residuals. | ***Run GLM ANOVA***<br><br>***with Residual diagnostics*** |

If your package has a two way ANOVA command try using it.

What happens?_____

**Data Situation #2 (continued)** Name _____

Now try the GLM command (be sure to compute fits and residuals)

Tape your ANOVA table here. If you decide to attach your ANOVA table to the end of the lab report, be sure it is adequately labelled (name of data file, name of the response variable).
Comment on whether the residuals are acceptable (independent, homogeneous, and normal).
If you attach your comments at end of report, state the name of input data file and response variable.

State whether you would trust the p-value computed by Minitab from an F-distribution, providing reasons. If you attach your comments at end of report, state the name of input data file and response variable.

Is your glm ANOVA table the same as that obtained by Sokal and Rohlf (1995) in Box 11.7 ?

Report the F-ratio from Sokal and Rohlf in Box 11.7.

$MS_{clutches}/MS_{error}$ = _____ / _____ = _____

**Data Situation #3** Name _____

Analyze the data in exercise 14.12 from Sokal and Rohlf (1995).
Follow the steps in the generic recipe for the execution of the general linear model.

The response variable is: _____

The explanatory variable (categorical) is : _____

The explanatory variable (regression variable) is: _____

Note: All three variables (and an interaction term) appear in the GLM command.
You must also define whether your explanatory variables are categorical (ANOVA type) or regression variables.
Some packages assume all variables are categorical (*e.g.* Minitab).
For these packages you must declare the regression variables ( MTB > covariate X1)
Some packages assume all variables are regression variables (*e.g.,* SAS)
For these packages you must declare the ANOVA variables (SAS> Class X2)

Note:
You will find a substantial bowl in the residual vs fit plot.
The text book recommends a square root transformation to correct the problem.
Transform the data by taking the square root of the response variable, repeat the analysis and plot the residuals vs fits.
Did this correct the problem of a bowl ?
Next try taking the logarithm of both weight and age, then repeat the analysis.
Did this correct the problem of a bowl ?

To obtain parameter estimates, you will have to use several commands.
- a command that gives the overall mean, which is an estimate of $\beta_o$
- a command that gives the mean in each group (parameters of the categorical variable)
- a command that gives the overall slope (parameter of the regression variable)
- a series of commands to obtain the slope in each group
(these are the parameters of the interaction term)

## Write-up for this lab.

1. Fill in the blank spaces, as requested, throughout Laboratory 8.
(Data situation #1 and #2)

2. Data situation #3. Analyze the data in exercise 14.12 from Sokal and Rohlf (1995).
Use the generic recipe for the execution of the general linear model.
Show residual versus fit plots for all three models:
wt = f(age, meal)
sqrt(wt) = f(age, meal)
ln(wt) = f( ln(age), meal)
Skip step 4 ($H_A$ $H_o$) until you have a valid model (no bowls in residual vs fit plot).

# Laboratory #9.
# Problem-Solving with the GLM. I. Setting up the Analysis.

The purpose of this lab is to gain practice in setting up an analysis, often the most difficult step. In this lab you will apply the model based approach you learned in Lab 8 to three data sets. The first example is a sophisticated experimental design. The second is an extension of the first. The third is a data set for which there are multiple models. Having completed Lab 8, you will have no difficulty in completing Lab 9 and the subsequent write up in Lab 10. Upon completing Labs 9 and 10 you will have a greater capacity to undertake statistical analysis of biological data than if you had spent an entire year learning a battery of tests and their names.

The 3 data sets are available on the website for this course
www.mun.ca/biology/schneider/b4605/data
They are printed for you, later in this lab, with descriptive statistics.

wworm.dat
wworm2.dat
leprosy.dat

You are encouraged to work in groups to decide how to analyze each data set.
Lab 9 focuses on problem set-up, while Lab 10 focuses on execution.

The three examples use data from:

Snedecor, G.W. and Cochran, W.G. (1980) Statistical Methods 7th ed. Iowa State University Press: Ames, Iowa

Here is a recipe for model based analysis of biological data.
The recipe is an extension of the generic recipe for data analysis with the general linear model.

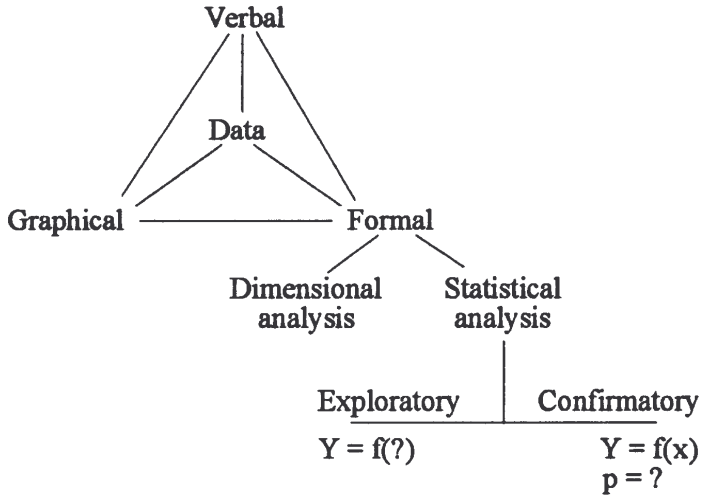
**Table 9.1.** Model based statistical analysis of biological data.

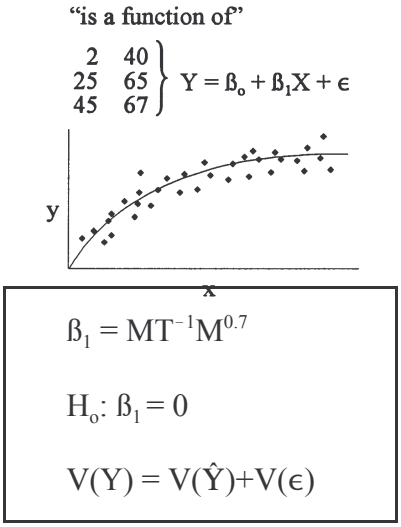
1. Define variables with symbols, scale of measurement, and units if possible.
2. State sample and population if possible, and whether sample is representative.
3. Construct model.
   Distinguish response from explanatory variables.
   If distinction not clear, undertake exploratory analysis.
   If clear, write out a brief verbal model about the response variable.
   Sketch how variables are thought to be related.
   Write out statistical model.
   Partition df according to model, complete Source and df columns of ANOVA table.
   Place data in model format, code model statement in statistical package.
4. If regression line is used, examine plot of residuals against fitted values.
   If bowl or arch is evident, revise the form of the model (back to step 3).
5. State $H_o/H_A$ (some analyses may require several pairs).
   State test statistic, its distribution (pdf), and tolerance of Type I error.
6. Partition df and SS = df*var(Response) according to model.
   Table Source, df, SS, MS, F (by computer usually).
7. Calculate Type I error (the p-value) from density function (F, t, or $X^2$ distribution).
8. Check assumptions for use of p-value from density function.
   residuals independent ? (plot residuals versus residuals at lag 1)
   residuals homogeneous ? (residual versus fit plot)
   residuals normal ? (histogram of residuals, quantile or normal score plot)
9. If assumptions are met then step 10. If not, decide whether to recompute p-value.
   Recompute better p-value by randomization if sample small ($n < 30$), and p near $\alpha$
10. Declare decision about model terms: If $p < \alpha$ then reject $H_o$ and accept $H_A$
    If $p \geq \alpha$ then accept $H_o$ and reject $H_A$
    Report conclusion with evidence: F-ratio, df1,df2, and p-value (not $\alpha$) for each term.
11. Examine parameters of interest. Report conclusions with parameter estimates
    (means, slopes) and one measure of uncertainty (st. error, st. dev., or conf. intervals)

The write-up for this lab (steps 1 through 3 in Table 9.1) and the write up for the next lab (steps 4-11 in Table 9.1) are both due together a week after Lab 10.

The next 4 pages expand on and describe in more detail the model based statistical approach outlined in Table 9.1. The three data situations in this lab have one response variable and several explanatory variables. The measurements are considered to be representative (samples of all possible measurements that could have been taken, given the experimental protocol). Consequently we will be using the confirmatory analysis and the machinery of formal hypothesis testing.

At this point skip directly to the data displays later in the lab (wworm1, wworm2, leprosy) and complete steps 1-3 in Table 9.1 for each of the data situations.



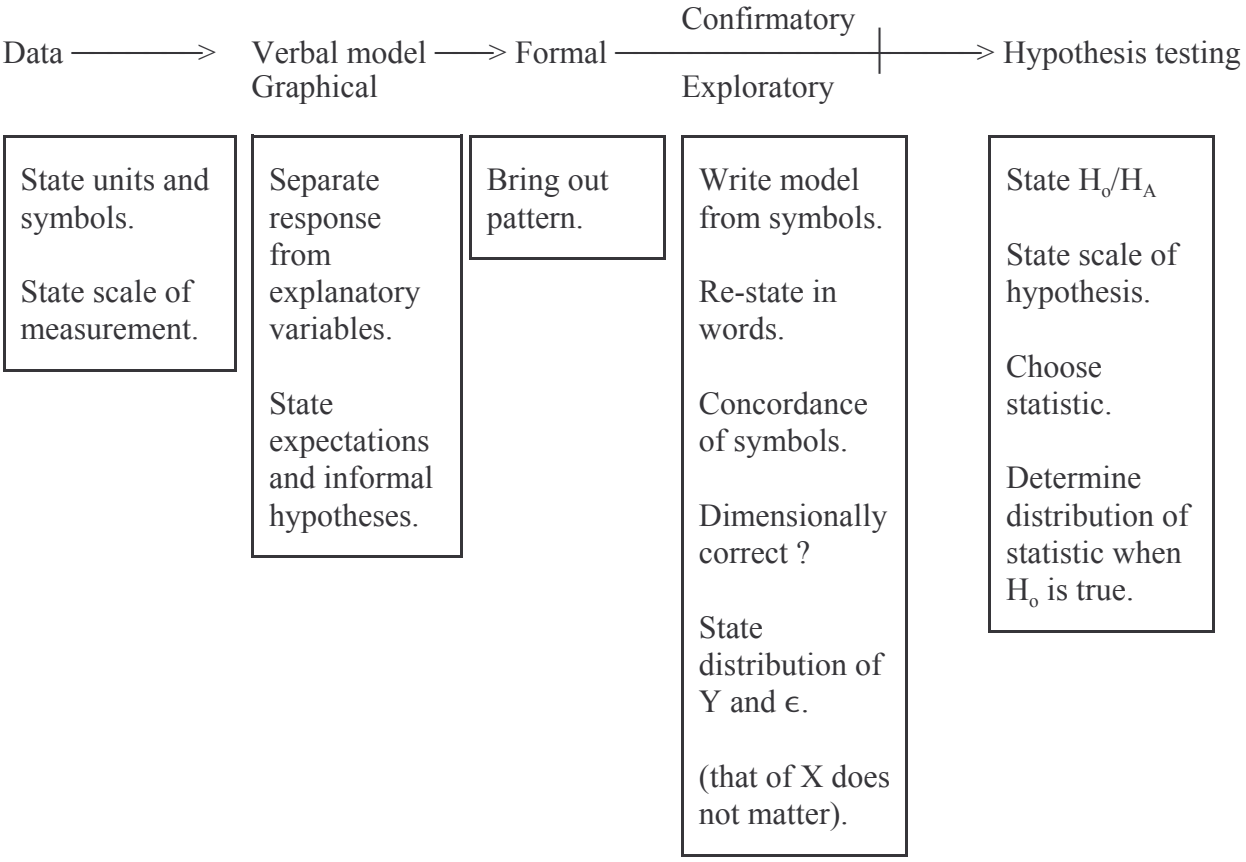




These diagrams combine, as mnemonic elements, the concepts presented thus far in the course. At the upper left is the familiar triangle, relating data to three forms of summarization. The first is verbal: one quantity is said to be a function of another quantity. For example, oxygen consumption is a function of cell size. The second form of summarization is graphical. For example, a series of measurements of oxygen consumption are plotted relative to cell size. A line drawn through the resulting cloud of points will represent the relation graphically. Finally, the relation can be expressed formally, as an equation. In the example at hand, oxygen consumption is the response variable, expressed as a function of cell size, the explanatory variable.

Once data have been summarized in a formal manner, this relation can be analysed further. Dimensional analysis uses reasoning about quantities based on the principle of similarity. This principle is used extensively in physiology, and to an increasing degree in ecology and environmental biology.

The second form, statistical analysis, is far more prevalent in biology. It is used in either a confirmatory or in an exploratory fashion. In an exploratory analysis, the emphasis is in identifying pattern. The goal is discovery of functional relationships. A convenient mnemonic symbol is Y = f(?), as in the diagram. In a confirmatory analysis the emphasis is on determining whether a relation (of known form) is present or not. In confirmatory analysis, the emphasis is on hypothesis testing, using a null/alternative pair. The $H_o/H_A$ pair can be on a nominal, ordinal, or interval scale. Hypothesis testing often relies upon comparison of the variance due to the model $V(\hat{Y})$ with the error variance $V(\epsilon)$.

The most common analytic route is from data to a verbal model, then to a graphical model, and then to a verbal model. This then becomes the basis for statistical analysis in either a confirmatory or exploratory fashion. The following diagram shows this route. The diagram shows one of many possible routes through the diagram on the previous page.



Many of the statistical tests used in biology are special cases of the general linear model. This family of models includes the most commonly encountered statistical tests in biology: regressions, ANOVAs, t-tests, ANCOVAs, etc. The general linear model is, in turn, a special case of the general**ized** linear model. Special cases of the generalized linear model include G-statistics, logistic regression, probit analysis, and of course the general linear model. The general**ized** linear model GzLM, like the general linear model GLM, relates a response variable to one or more explanatory variables.

The test that is used in any particular situation depends on the nature of the response variable, it depends on the number of explanatory variables, and it depends on the whether the explanatory variables are nominal or interval scale.

Here is a logical framework for deciding on which test to use. It has been set up in the form of a key, much like the keys used to identify an organism to family, genus, and species.

**Table 9.2.** Deciding which generalized linear model GzLM to use. See Table 9 in handouts

1*. Is the response variable Y a count ?
- If Y is Poisson----> GzLM with poisson error (G-statistics, log-linear models)
- If Y is binomial----> GzLM with binomial error (logistic regression).

2. Is the response variable Y continuous rather than consisting of discrete counts ?
- If explanatory variable X is nominal ----> ANOVA
  - If one X ----> one way ANOVA
  - If several X----> multiway ANOVA (2-way, 3 way etc)
    - Nested design-----> no interaction term
    - Crossed design--> interaction term (if enough df)
- If explanatory variable X is interval or ratio scale ------> Regression
  - If one X-----> simple regression
  - If several X-----> multiple regression
- If both nominal X and interval (or ratio) X------> ANCOVA

* Note that count variables are sometimes analyzed using normal errors because of the flexibility in defining the structural model with GLM routines (which assume normal errors). If you do analyze count data with the General Linear Model (normal errors) rather than the General*ized* Linear Model (which include binomial and Poisson errors), then you must check the residuals for homogeneity. Often the residuals will be heterogeneous--they will show cones when plotted against fitted values.

Generic recipe for setting up an analysis. Expands somewhat on Table 9.1.

**Table 9.3.** Generic recipe for setting up an analysis.

1. State symbols for each variable, state units, state scale of measurement of each variable (nominal, ordinal, interval, ratio).
2. Separate response from explanatory variables.
3. State whether measurements of the response variable are independent of one another, and whether these measurements are a representative sample from a larger population. If not known, state "not known". If a true sample has been taken, state the population.
4. State the purpose of the analysis, whether exploratory or confirmatory.
   State the level of analytic outcome desired, whether nominal (yes or no decision), ordinal (ranking of several decisions), or ratio (estimating parameters).
5. State one or more verbal models about the response variable(s).
6. Convert your verbal model into a sketch of response variable(s) as a function of explanatory variable(s),
7. If formal hypothesis testing is warranted, invoke the machinery of formal hypothesis testing:
   state $H_o/H_A$ (+ the General Linear Model if appropriate)
   state statistic St
   state tolerance of type I error ($\alpha$)
   It helps to state the name of the test, if any.
   Write a formal model using symbols from step (1), combined with parameters as needed.
   Prepare a concordance of symbols, if this is needed.
   For example, there are several forms of notation for regression.
   Determine whether your equation is dimensionally correct by writing out the dimensions of each symbol, then checking to make sure that all terms have the same dimensions.
8. Execute analysis, examine residuals. In a confirmatory setting, declare decision if residuals are acceptable. If not acceptable, take appropriate action:
   - better model of relation of variables,
   - better model of error:
     - generate distribution (randomization test)
     - non-nomal error (generalized linear model)

```
4 3     0 2     2 5     3 1     1 4
1 6     3 0     0 6     2 4     4 4
0 4     1 9     3 1     4 6     2 5
2 17    4 8     1 8     0 9     3 0
3 4     2 4     4 2     1 4     0 8

T  N    T N     T N     T N     T N
```

**WWORM1.dat**

Snedecor and Cochran (1980) p 289

N = number of wire worms per 9 x 9 x 5 inch plot
T = treatment: control (0) and 4 soil fumigants.
Plots are arranged in a 5 x 5 array.
Each treatment occurs in each row and in each column. This is called a Latin Square

```
MTB > read 'wworm1.dat' c1-c10;
SUBC> nobs = 5.
MTB > stack c1 c3 c5 c7 c9 [into] c11
MTB > stack c2 c4 c6 c8 c10 [into] c12
MTB > name c11 'trtmnt'  c12 'count'
MTB > set [into] c13
DATA> (1 2 3 4 5)5
DATA> end
MTB > set [into] c14
DATA> 5(1 2 3 4 5)
DATA> end
MTB > name c13 'col'  c14 'rows'
MTB > print c11-c14
  ROW  trtmnt  count   col   rows
    1       4      3     1      1
    2       1      6     1      2
    3       0      4     1      3
    4       2     17     1      4
    5       3      4     1      5
    6       0      2     2      1
    7       3      0     2      2
    8       1      9     2      3
    9       4      8     2      4
   10       2      4     2      5
   11       2      5     3      1
   12       0      6     3      2
   13       3      1     3      3
   14       1      8     3      4
   15       4      2     3      5
   16       3      1     4      1
   17       2      4     4      2
   18       4      6     4      3
   19       0      9     4      4
   20       1      4     4      5
   21       1      4     5      1
   22       4      4     5      2
   23       2      5     5      3
   24       3      0     5      4
   25       0      8     5      5
```

*continued... (over)*

```
MTB > plot 'count' 'trtmnt'                                          WWORM1.dat
     18.0+
        -                              *
 count  -
        -
        -
     12.0+
        -
        -
        -   2            2                                 *
        -
      6.0+   *            *                                 *
        -                              2
        -   *            2             2            *       *
        -   *                                                2
        -                                          2
      0.0+                                         2
        -
          --+---------+---------+---------+---------+---------+----trtmnt
         0.00      0.80      1.60      2.40      3.20      4.00

 MTB > describe 'count';
 SUBC> by 'rows'.
              rows        N     MEAN   MEDIAN   TRMEAN    STDEV   SEMEAN
 count           1        5    3.000    3.000    3.000    1.581    0.707
                 2        5     4.00     4.00     4.00     2.45     1.10
                 3        5     5.00     5.00     5.00     2.92     1.30
                 4        5     8.40     8.00     8.40     6.02     2.69
                 5        5    4.400    4.000    4.400    2.191    0.980

 MTB > describe 'count';
 SUBC> by 'col'.
               col        N     MEAN   MEDIAN   TRMEAN    STDEV   SEMEAN
 count           1        5     6.80     4.00     6.80     5.81     2.60
                 2        5     4.60     4.00     4.60     3.85     1.72
                 3        5     4.40     5.00     4.40     2.88     1.29
                 4        5     4.80     4.00     4.80     2.95     1.32
                 5        5     4.20     4.00     4.20     2.86     1.28
```

```
MTB > anova 'count' = 'rows' 'col';
SUBC> residuals c15.

Analysis of Variance for count

Source      DF         SS         MS       F      P
rows         4      84.56     21.140    1.61  0.220
col          4      22.16      5.540    0.42  0.791
Error       16     210.24     13.140
Total       24     316.96     13.207

MTB > anova 'count' = 'rows' 'col' 'trtmnt'
* ERROR * Unequal cell counts        <---this will not work; try GLM command
                                          (Be sure to examine residuals)
```

```
4  6    0  3     2 29     3  8     1 17
1  8    3 13     0 18     2 12     4 16
0 15    1 13     3  7     4 10     2 28
2 14    4 11     1 13     0 22     3  7
3  7    2 26     4 24     1 14     0 20

T  N    T  N     T  N     T  N     T  N
```

**WWORM2.dat**

Snedecor and Cochran (1980)
p 273

T = treatment: 0 = contrl 1,2,3,4 = 4 soil fumigants
N = number of wire worms per 9 x 9 x 5 inch plot
Each treatment occurs in each row, and in each column of the 5 x 5 array.
This is called a Latin Square layout.
This is a repeat (1 year later) of the experiment reported as 'WWORM1.dat'

**WWORM2.dat**

```
MTB > read 'wworm2.dat' c1-c10;
SUBC> nobs = 5.
MTB > stack c1 c3 c5 c7 c9 into c11
MTB > stack c2 c4 c6 c8 c10 into c12
MTB > name c11 'trtmnt'  c12 'count'
MTB > set into c13
DATA> (1 2 3 4 5)5
DATA> end
MTB > set into c14
DATA> 5(1 2 3 4 5)
DATA> end
MTB > name c13 'col'  c14 'rows'
MTB > plot 'count' 'trtmnt'
          -
        30+                               *
          -                               *
 count    -                               *
          -                                                   *
          -   *
        20+   *
          -   *             *
          -   *                                             *
          -                 3             *             *
          -                               *                   *
        10+                                                   *
          -                 *                       4
          -                                                   *
          -   *
          -
         0+ --+---------+---------+---------+---------+---------+----trtmnt
           0.00      0.80      1.60      2.40      3.20      4.00
```

*continued... (over)*

WWORM2.dat (continued)

```
MTB > describe 'count';
SUBC> by 'rows'.
              rows        N     MEAN   MEDIAN   TRMEAN    STDEV   SEMEAN
count            1        5    12.60     8.00    12.60    10.55     4.72
                 2        5    13.40    13.00    13.40     3.85     1.72
                 3        5    14.60    13.00    14.60     8.08     3.61
                 4        5    13.40    13.00    13.40     5.50     2.46
                 5        5    18.20    20.00    18.20     7.76     3.47

MTB > describe 'count';
SUBC> by 'col'.

              col        N     MEAN   MEDIAN   TRMEAN    STDEV   SEMEAN
count            1        5    10.00     8.00    10.00     4.18     1.87
                 2        5    13.20    13.00    13.20     8.26     3.69
                 3        5    18.20    18.00    18.20     8.70     3.89
                 4        5    13.20    12.00    13.20     5.40     2.42
                 5        5    17.60    17.00    17.60     7.57     3.39
```

```
11   6    6   0    16  13
 8   0    6   2    13  10
 5   2    7   3    11  18
14   8    8   1     9   5
19  11   18  18    21  23
 6   4    8   4    16  12
10  13   19  14    12   5
 6   1    8   9    12  16
11   8    5   1     7   1
 3   0   15   9    12  20

 B   A    B   A     B   A
 TRI      TRII   Control
```

**leprosy.dat**

Snedecor and Cochran (1980) p 368

Scores for Leprosy bacilli before (B) and after (A) treatment with two types of antibiotics (TRI and TRII), with one control (C)

```
MTB > read 'leprosy.dat' c1-c6;
SUBC> nobs = 10.
MTB > stack c1 c3 c5 [into] c7
MTB > stack c2 c4 c6 [into] c8
MTB > name c7 'bef'  c8 'aft'
MTB > set [into] c9
DATA> (1 2 3)10
DATA> end
MTB > name c9 'trtmnt'
MTB > describe c1-c6
                    N      MEAN    MEDIAN    TRMEAN
STDEV   SEMEAN
 bTRI              10      9.30      9.00      8.88
4.76     1.51
 aTRI              10      5.30      5.00      5.00
4.64     1.47
 bTRII             10     10.00      8.00      9.50
5.25     1.66
 aTRII             10      6.10      3.50      5.37
6.15     1.95
 bCONT             10     12.90     12.00     12.62
3.96     1.25
 aCONT             10     12.30     12.50     12.37
7.15     2.26

 MTB > plot c2 c1
     15.0+
         -
 aTRI    -                              *
         -
         -
              *
     10.0+
         -
         -                                       *         *
         -
         -                                       *
      5.0+
         -              *
         -
         -           *
         -              *
      0.0+   *                  *

--+---------+---------+---------+---------+-------
--+----bTRI
            3.0       6.0       9.0      12.0
15.0      18.0
```

*continued... (over)*

leprosy.dat (continued)

```
 MTB > plot c4 c3

    18.0+                                                         *
       -
aTRII  -
       -                                                              *
       -
    12.0+
       -
       -
       -              *                          *
       -
     6.0+
       -
       -              *
       -     *   *
       - *           *
     0.0+     *
       -
         +---------+---------+---------+---------+---------+------bTRII
        5.0       7.5       10.0      12.5      15.0      17.5


MTB > plot c6 c5

    24.0+
       -                                                              *
aCONT  -                       *
       -
       -                  *
    16.0+                      *
       -
       -                                      2
       -
       -                           *
     8.0+
       -
       -          *           *
       -
       - *
     0.0+
       -
         --+---------+---------+---------+---------+---------+----bCONT
           7.5       10.0      12.5      15.0      17.5      20.0
```

# Laboratory #10.
# Problem-Solving with the GLM. II. Executing the Analysis.

Data analysis typically consists of an iterative process of setting up an analysis, execution of the analysis, and then subsequent analyses as required. In a confirmatory setting the follow-up consists of examination (and perhaps formal analysis) of the residuals. If the residuals are not acceptable, appropriate action is taken, either by using a more appropriate model, or by generating an empirical frequency distribution to calculate a p-value to declare a decision.

Typical sequence of steps in confirmatory data analysis.

1. Initial setup of problem for analysis, according to steps 1 - 3 in Lab # 9 (Table 9.1).
   Data ⟶ Verbal ⟶ Graphical ⟶ Formal ⟶ Hypotheses
2. Execution of the analysis with calculation of residuals (step 4-11 in lab #9, Table 9.1).

In an exploratory setting there is a more extended sequence of set-up, execution, and analysis of residuals. The sequence is repeated until no more pattern can be extracted no more pattern can be extracted from the residual plots, or the pattern has been shown to be stable under a variety of techniques.

The purpose of labs 9 and 10 is to give you practice in setting up (Lab 9) and executing (Lab 10) statistical analyses of biological data in a confirmatory setting.

**Write up for Lab 10.**

Complete steps 1-11 in Table 9.1 for the analysis of the three data sets displayed in Lab #9.

Honours students with a data set of their own can substitute their own data set for one of the three sets in Lab 9 analyses. If you decide to use your own data set, be sure to describe (in a short paragraph) how the data were collected.

Be sure to use computer outputs as exhibits, using only relevant parts of the computer output. Each exhibit should be labelled and accompanied by verbal statement of the result next to the exhibit.

Name _____

# Laboratory #11.
# Bootstrap Estimates

Randomization methods so far have been used to compute p-values for hypothesis testing. Randomization methods can also be used to place confidence limits around estimates. Randomization methods can be used to place a confidence limit around any statistic, not just those that have known statistical distributions. As can be seen by referring to the key for choosing a frequency distribution (see Handouts), randomization methods using the "bootstrap" method are especially useful when errors around an estimate are not normally distributed, or when the statistical distribution for a statistic is unknown.

The bootstrap is a technique for estimating a statistic and its distribution. It uses resampling. The resampling for randomization tests (Lab 4) was carried out WITHOUT replacement. The values of the response variable (in Minitab column c1, for example) were sampled without replacement into a new arrangement (in Minitab column c2, for example). Each value in the first arrangement (c1) occurs only once in the second (c2). The advantage of repeated sampling and calculation, of course, was that a statistical analysis could be undertaken without making assumptions about the distribution of outcomes.

Bootstrap methods use sampling WITH replacement. That is, any individual observation in one column can be sampled repeatedly. Any single observation in column c1 can appear repeatedly in c2. Or it may not appear at all in c2.

The bootstrap was invented by Bradley Efron (*Annals of Statistics* 7:1-26). It is described in Manly (1991: *Randomization and Monte Carlo Methods in Biology* Chapman Hall).
The purpose of this lab is:

→ to describe the bootstrap

→ to show how to compute a bootstrap estimate using Minitab

→ to show how to devise an accurate computational procedure in Minitab.

To complete this lab, you will need a package that returns a random sample of values from a column of data. Most statistical packages will do this. The basic version of commercially available spreadsheets do not have functions to do this directly.

You will also need a statistical package or spreadsheet that will execute a command repeatedly in order to accumulate the values of a statistic from repeated runs into a single column. This can be done readily with line code commands. It is hard to execute a batch file from a graphics interface.

**The Bootstrap.**

The idea behind the bootstrap is simple. A sample of n values of a variable quantity $Q = Q_1, Q_2, ..., Q_n$ is taken from a population and used to estimate some parameter, such as the mean of the population, or the skewness of the population. The true value of the parameter can never be known exactly unless the entire population is sampled, but of course the parameter can be estimated from the n observations. The sampling variation in this estimate is assessed by taking random samples of size n. This collection of random samples of size n can be as large as we like. We are going to regard this observed distribution as the best approximation of the true distribution of all possible samples from the population.

The samples that make up this approximate distribution of $Q$ are called bootstrap samples. Each sample is used to make a bootstrap estimate of the true parameter (mean, slope, variance, diversity index, etc). The distribution of these bootstrap estimates, which we require for hypothesis testing and stating confidence limits, can be obtained by repeated sampling and then constructing an observed frequency distribution.

Example: Mandible lengths of male golden jackals (Manly 1991)

$Q$ = [ 120 107 110 116 114 111 113 117 114 112 ] · mm

The observed mean value is mean($Q$) = 114.3 mm. This is an estimate of a parameter, the true mean.

To demonstrate how bootstrap estimates work we will to obtain a bootstrap estimate of the mean. This will be done by taking 10 samples (WITH replacement) from the collection of 10 observations, computing the mean from each sample, and then collecting 500 of these estimates.

To do this, we will use Minitab to sample WITH replacement.

The next page shows a series of steps for determining how to accomplish this in Minitab.

Name _____

**Table 11.1.** Generic recipe for devising computational procedure in any statistical package.

1. State the computation, in words.
2. Find a set of commands in your package to execute the procedure
3. Check the commands with a sample set of made-up data.
4. If it is not correct, look for another set of commands.
5. Repeat until a correct procedure is found.

Step 1. State the computation. sample with replacement from c1 into c2

Step 2. Use the help file in your package to find the commands.

```
MTB > Help Commands
```

You may need to choose a collection of commands:

```
MTB > Help commands 15
                                                    Σ
```

Which command are you going to use to sample with replacement from c1 into c2?

_____

Now display the help file for this command.

Step 3. Next, see if this command works on a small data set.
Put some data into a column for a test run.

```
MTB > set into c1
DATA>   3 4 5 6 7 8
DATA>  end
```

If you sample from c1 into c2 with replacement, what do expect to see in c2? (this will be marked as present absent, not on whether it is correct)

Now execute the command to sample from c1 into c2. Then display c1 c2.

You should see some of the numbers in c1 several times in c2. You should also see that some numbers from c1 are missing from c2. If this is not happening, the computational procedure is probably not correct.

Step 4. If it is not correct, look for another set of commands.

Step 5. Repeat until an accurate procedure is found.

* * *

**Bootstrap estimates. Mean mandible lengths of golden jackals.**

Now that we have a computational procedure for sampling with replacement, we can move on to making bootstrap estimates of a statistic.

The first statistic will be the mean for mandible lengths of male golden jackals.

Example: Mandible lengths of male golden jackals (Manly 1991)

$Q$ = [ 120 107 110 116 114 111 113 117 114 112 ] · mm

Set these 10 observations into column 1 of your statistical spreadsheet

***Define Data from keyboard***

Next, place a random sample of 10 observations from c1 into c2.

Name _____

Next, calculate the mean value of the random sample in c2 and store this in k1.
You may need to use the help facility to do this.
The mean in c1 and c2 should differ slightly.

Next, store your bootstrap estimate of the difference in two means at the top of column 3
You may need the help facility.

Once you have several values (random means) in column c3, write a batch file to add a bootstrap estimate to column 3 every time you execute the file.

```
MTB > Store 'Jackal.ctl'
STOR>
```

***Create control file***

Make sure the batch file is adding a new and slight different estimate to c3 each time the batch file is executed.

Once the batch file is working correctly, then Tape, paste, or write out by hand the batch file:

**Bootstrap estimates. Confidence intervals.** **Name**_____

Now accumulate 500 bootstrap estimates of the mean in c3.

To make sure you have 500 observations

```
MTB > n c3
```

Compute the mean value of the bootstrap estimates. _____

How does this compare to the estimate of 113.4? _____

A bootstrap estimate of a mean is hardly necessary because simply taking the sum and dividing by the number of observations will produce a perfectly good estimate. But what about the confidence limits around the mean? A theoretical distribution may not be available for obtaining a confidence limit, in which case a bootstrap estimate of the confidence limits will be useful.

You will recall that the confidence limit is a range around an estimate that will include the true value of the parameter 95% of the time (or 99% of the time, or whatever).

To obtain a rough idea of the 95% confidence limits around the bootstrap estimate of the mean, construct a histogram of the bootstrap estimates in c3.

Tape or paste your histogram here

Name _____

Based on this histogram, make a rough guess and write out the upper and lower limits that include 95% of the bootstrap estimates (answer does not have to be exact).

Upper limit = _____ mm

Lower limit = _____ mm

Next, try to think of a way to determine the 95% limits exactly, from the 500 bootstrap estimates in c3 (this will be marked as present/absent not on whether it is correct).

There are several ways of computing the 95% limits exactly for the 500 bootstrap estimates in c3. One is to use the percentile function found in many packages, including spreadsheets.
Another is to use the sort function, found in almost all packages.
With the sort function you will need to count downward in the sorted column, to reach the value for which 2.5% of the values are smaller. . You will then need to count upward in the column to reach the value for which 2.5% of the values are larger.

Try the percentile method. If you package does not make the computation, try the sort method.

Compute the upper and lower limits that include 95% of the bootstrap estimates.

Upper limit = _____ mm

Lower limit = _____ mm

Now compute then compare these bootstrap estimates of the 95% confidence limits to the theoretical limits using the t-distribution. Here are the computations:

```
MTB > describe c1
      N     MEAN   MEDIAN   TRMEAN    STDEV   SEMEAN
C1   10   113.40   113.50   113.37     3.72     1.18

MTB > invcdf .95;
SUBC> t 9.
     0.9500    1.8331
```

**What is wrong with this invcdf command wrong?** _____

Name _____

```
MTB > invcdf .975;
SUBC> t 9.
     0.9750    2.2622
```

Now compute the theoretically based upper and lower limits using: the mean, the standard deviation, sample number n, and the t-statistic:

```
MTB > let k1 = 113.4 - 2.2622*stdev(c1)/sqrt(10)
MTB > let k2 = 113.4 + 2.2622*stdev(c1)/sqrt(10)
MTB > print k1 k2

K1       110.740
K2       116.060
```

## Write up for this laboratory:

1. Work through the bootstrap example shown above, filling in all of the blanks.
2. Use bootstrap methods to set 95% confidence limits on the proportion of ants in the diet of short-horned lizards *Phrynosoma douglassi brevirostre*, from near Bow Island, Alberta during the month of June (data from Manly 1991 p 247). Use 1000 bootstrap estimates to obtain a good estimate of the upper and lower limits.

   p = [ 33 100 43 100 14 64 40 0.0 47 ]·%

   Upper limit = _____ mm

   Lower limit = _____ mm

   Compute the upper and lower limit 95% confidence limits using a t-distribution.

   Upper limit = _____ mm

   Lower limit = _____ mm

   Should you report the first or the second pair of confidence intervals in a thesis or other document? Why?

# Guide to Computing -- Quantitative Biology

The cheap computing power of the personal computer has completely changed the way that statistics are now done in biology. As the cost of calculation has dropped, more and better techniques have become widely available. An example is multivariate analysis, which became far easier with a personal computer. Other examples of good practice made possible by cheap computing power include diagnostics of residuals, randomization tests, and iterative estimation of parameters. Commonly used procedures in statistics are packaged together as a set of routines. Most statistical packages (for example, Minitab, SPlus, Systat) have easy to use graphical (menu-driven) interfaces. To complete the lab assignments and problem sets in this course you will need a statistical package. You can use a package that you already know, provided it meets the requirement in the table below.

Students at Memorial University have access to a recent version of Minitab with a graphics interface and pull-down menus. The SAS statistical package (line code only) is available to everyone on the University's PLATO operating system (Unix). SPSS is available in several computer labs at the University, but its general linear model procedure has limitations. The following table compares several packages with respect to procedures that are either convenient or necessary for this course.

| | Spreadsheet | Statistical packages | | | | | |
|---|---|---|---|---|---|---|---|
| | Excel | Minitab | SAS | SPSS | Splus | Systat | Other |
| Spreadsheet visible | ✓ | ✓ | No | ✓ | No | ✓ | |
| Pull down menus | ✓ | ✓ | No | ✓ | No | ✓ | |
| *Basic statistical functions | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| *Return a column of data in random order. | No | ✓ | ✓ | ✓ | ✓ | ✓ | |
| *General Linear Model | No | ✓ | ✓ | L | ✓ | ✓ | |
| *Residual analysis | ✓ | ✓ | ✓ | L | ✓ | ✓ | |
| *Logistic regression | No | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Generalized Linear Model | No | No | ✓ | No | ✓ | ✓ | |

L indicates limited capacity or problems with output.
* required for this course

How to use this guide.

This guide steps you through the basics you will need to use computers in this course. The guide contains tables and reference material that you can use as a reference later on, when you are completing the computer laboratories and problem sets. In the computer labs you will be given generic computational commands that will be explained and linked to specific examples. These generic commands will appear in boldface italics.

***Define Data***
***from file***

To see some examples, look for the boxes in Laboratory 3. We will work through an example later in this guide.

These are called generic commands because they describe a sequence of tasks, regardless of the details of the package you are using. Upon first occurrence these generic commands will have explanations and a sequence of actions (called pseudocode) to be executed regardless of the package. The pseudocode will be followed by examples of execution with a menu driven package (Minitab), with line code (Minitab), and in some cases in a spreadsheet (Excel). After several appearances the actions specific to a package will disappear. When the generic command or pseudocode appears again, you will have to decide how to carry it out in your statistical package.

Review:

How have computers changed the practice of statistics in Biology ?

What is a generic command ?

What is pseudocode ?

## The Basics

Here is a checklist of basic tasks you need to know how to do, before starting the computer labs, or doing problem sets on the computer. As you work through this guide, check off each skill as you learn it.

If you are already familiar with some or all of these basics, you should still go through the list, check off the items you know how to accomplish, perform the rest, and check them off as you complete them.

_____ start the computer and log in if necessary

_____ list the names of files on your computer, or in your account if you have one

_____ create a data file on your computer, or in your account if you have one

_____ copy a data file from a server to your computer

_____ start up the statistical package

_____ read data into a package

_____ use the package to compute a mean

_____ send output from package to a file

_____ extract and print key pieces of this output file

_____ log off of your account

If you are using a package on your own personal computer or in a lab at the University, work through Section I. If you are using SAS on the University computer (Plato) work through section II.

Section I. Statistical packages on PCs.
- Part A. Meet the system
- Part B. Browsers, Email, Text editors.
- Part C. Meet the statistical package
  - Input
  - Descriptive statistics
  - Output
  - Saving your work

Section II. SAS on time-shared Unix system
- Part A. Meet the system
- Part B. Text editor (PICO) and email (Pine)
- Part C. Meet SAS

# Section I. Statistical packages on PCs.
## Part A. Meet the system.

The operating system takes care of house keeping on any computer you use. Examples of operating systems include:

Unix, Windows, Apple OS, Linux, MS-DOS

Operating systems perform tasks such as manipulating files, running programs, viewing file contents, and viewing lists of files (directories). Some system level commands are used frequently, while others are rarely used. In graphically oriented systems (Apple, Windows) the commonly used commands occur near the top of pull down menus.

The computers available in the labs at Memorial require you to have an account with userid and password. You can use this to log on to any computer in both teaching and open access labs across campus.

At this point, make sure you can

start the computer and log in if necessary;
list the names of files on your computer, or in your account if you have one;
create a data file on your computer, or in your account if you have one.

# Section I. Statistical packages on PCs.
# Part B. Browsers, Email, and Text editors.

**Browsers** are one of the most widely used programs on PCs. In this course we will use the browser to move data files from a central location (a server in Biology) out to the computer you are using. Here is the address.

www.mun.ca/biology/schneider/b4605/data

Using your web browser or ftp utility go to this server and find the following data file, and move it to your desktop.

Srbx9_5.dat

**Email** is another widely used program on most computers. There are many email programs.

Pegasus
Eudora
Access (Microsoft)
Pine (on Unix machines)
Communicator (in the Netscape Browser)

You can move files from one account to another with the email utility. In this course you can use email to ask questions about material in the lecture, about completing the labs, or about completing the problem sets. For this to happen, a class mailing list is needed. The most accurate way to assemble the list is for you to send a message to the prof:

A84DCS@Mun.Ca

so that you are on the mailing list for the course.

# Section I. Statistical packages on PCs.
# Part B. Browsers, Email, and Text editors.

**Text editors** complete the trio of most widely used programs on computers. In this course you will need a text editor to open data files and move data into the statistical package. You will also find a text editor useful in preparing reports with graphical and tabular display of statistical results. Text editors can be used to create data files, or edit them. You won't need to use a text editor to create or edit data files if your statistical package displays a spreadsheet. If you are using a statistical package that does not display a spreadsheet (such as SAS on Plato) you will find the text editor useful for creating and editing data sets.

All personal computers come with text editors, sometimes several. These vary in their complexity and sophistication. It doesn't matter which editor you use, but you do need to know how to save data as ASCII (unformatted) text. Some text editors automatically save files in ASCII format. Here are some examples:

pico, vi and edt (UNIX)
Notepad Wordpad(Windows)

Many word processors (WordPerfect, MS Word, etc.) save files in their own non-ASCII format. To save a file as ASCII, use the 'Save As' option under the file menu. One of the options under 'Save As' will be a standard (ASCII) text file.

The advantage of an ASCII file is that it can be read by any package on any system. ASCII was around before any of the currently used text editors. It will be around long after currently used text editors are gone. Many a researcher has lost data because they saved it only in a format used by a single package. If they had saved it as a text (ASCII) file they would still have their data.

> ASCII (American Standard Code for Information Interchange) a nearly universal format for representing characters on a computer. It consists of 256 characters (essentially all those appearing on your keyboard, plus some extras that you won't need to use).

Now open the data file Srbx9_5.dat using a text editor. You can click directly on the file, which should open with a default text editor. Or you can open your preferred word processor, then open the data file from inside the program. Save the file in ASCII (text) format using the appropriate choice under the 'Save As' option in the word processor.

# Section I. PCs.
# Part C. Meet the statistical package. Input .

The previous sections covered the basic computer skills you will need for this course. These were:

A. At the system level:
   find and move files,
   find commands and obtain help in using them,
   print a file.
B. Use the browser to move data files
   use the email utility to extract, save, and send files.
   use a text editor to open or create and save a data file.

At this point you should go back to the list of items to do (Basics) and make sure you are comfortable with the first 4 items in the list. With these skills, it is time to become acquainted with the statistical package.

Statistical packages with a graphics interface display at least two and sometimes three windows. One is for data and the others are for commands and output. The window for data looks and operates like a spreadsheet. If you already know how to use a spreadsheet (Excel, QuattroPro, Lotus, *etc*.), then you can use what you know to work with data in statistical packages with a spreadsheet interface (Minitab, SPSS, Systat, *etc.*).

There are several ways to bring data into a spreadsheet or spreadsheet interface. One is to type values in one at at time into the cells of the spreadsheet. For less than 10 values, it is often quicker to type the data than to bring it in from a file. For more than 10 values, it is better to move data with the cut and paste functions on the mouse. We'll call the process of bringing data into the statistical package 'data definition.' Here is a generic recipe (pseudocode) with generic command.

Generic recipe for bringing data into a spreadsheet or spreadsheet interface.

***Define Data from file***

Pseudocode (applies to any spreadsheet)
- Open file that has data, usually on the desktop.
- Use mouse to highlight the data
- Copy the highlighted data
- Paste onto the spreadsheet
- Name the columns

Once you have looked at the pseudocode, carry out the procedure in any package you like. Use the data from Srbx9_5.dat, which shows age of *Daphnia* for genetic strains I and II.

## Section I. PCs.
## Part C. Meet the statistical package. Input .

Excel Spreadsheet.
Paste to cell (data will appear below and to left of cell)
If several columns of data appear in one spreadsheet column, then go back to the original data file and insert tabs after every number in every row. Copy and paste again.
In the spreadsheet, insert a row above the data.
In the first row of column 1 type *age(I)*
In the first row of column 2 type *age(II)*

Minitab worksheet.
Paste to cell (data will appear below and to left of cell).
Click on top of column 1, type in variable name *age(I)*
Click on top of column2 , type in variable name *age(II)*

To see or use the Minitab command lines, you may need to switch on the command line display. Once you have, you can use both command lines and the graphics interface. In fact all the graphics interface does is write the command lines which are then executed. All statistical packages with a graphics interface work this way.

Minitab Menu
Editor
Enable commands

```
MTB > read 'srbx9_5.dat' c1 c2;
SUBC> nobs=7.
MTB > name c1 'age(I)'  c2 'age(II)'
```

This is called ***Data definition*** because we are assigning a name as well as values to the variable.

# Section I. PCs.
# Part C. Meet the statistical package. Descriptive statistics.

Next, we perform simple operations on the variables within the spreadsheet. These operations are performed by functions in the statistical package. One simple function is taking the mean of variable X, for which the functional expression is *mean(X)*. The mean value will be placed at a location in the spreadsheet, which we will call *f(x)*.

Here is the generic recipe (Pseudocode) with generic command.

Generic recipe for calculating means of variables (columns) in a spreadsheet

***Calculate statistic mean(X)***

Pseudocode (applies to any spreadsheet)
- Select a location to place *f(X) = mean(X)*.
- Find the *mean(X)* function
- Apply the function to the variable (entire column).
- Make the calculation.

Now that you have looked at the pseudocode, compute the mean age for each strain of *Daphnia*, using the data from Srbx9_5.dat that you pasted into your spreadsheet.

Excel spreadsheet
- Click cell at bottom of column 1
  - This will be the location of *f(X) = mean(X)*.
- Function
  - Statistical
    - Average
      - A2:A8 (7 values in column A)

Minitab spreadsheet
- Calculate
  - Column statistics
    - Mean
    - Input variable Age(I)

Highlight the result, copy and then paste onto the spreadsheet.

```
MTB > let k1 = mean(c1)
MTB > let k2 = mean(c2)
MTB > print k1 k2
MTB > desc c1-c2
```

## Section I. PCs.
## Part C. Meet the statistical package. Descriptive statistics.

Now that we have results, we need to communicate it to others in a format that is readily understood. For statistical results, this means explanations tied to the numbers and graphs. One easy way to accomplish this is to use cut and paste functions to move selected portions of the statistical results to a document in a word processor. Here is a generic command with recipe (pseudocode).

Generic recipe for output of statistical results to a document for printing.

***Output to document***

Pseudocode (applicable to almost any package).
- Open document in a new window.
- Return to spreadsheet or statistical package output
- Highlight output to be reported, click copy
- Return to document and paste at appropriate place.
- Repeat process for each selected portion of output.
- Add explanation.
- Print the document.

Now that you have looked at the pseudocode, try moving the *Daphnia* data and the means to a document in your preferred word processor.

As you move results from the package to your document consider how the document will look. Be sure to move only the results you need. Leave out the extraneous material. Be sure to check the format of the results after you paste them into the document. If the results were in columns, and they are no longer aligned, highlight the portion you transferred, then apply a format with uniform spacing (such as courier). This should bring mis-aligned columns back into alignment.

Here is an example. First with a scalable font that destroys alignment.

| SOURCE | DF | SS | MS | F | p |
|---|---|---|---|---|---|
| Regression | 1 | 0.096644 | 0.096644 | 23.64 | 0.005 |
| Error | 5 | 0.020442 | 0.004088 | | |
| Total | 6 | 0.117086 | | | |

Now the same text displayed with a non-scalable font (courier) to restore the intended alignment.

```
SOURCE        DF          SS          MS          F         p
Regression     1    0.096644    0.096644     23.64    0.005
Error          5    0.020442    0.004088
Total          6    0.117086
```

## Section I. PCs.
## Part C. Meet the statistical package. Saving your work.

Once you have moved the results from the package to a document, you can either save the worksheet you have created or you can discard it. If you save the worksheet, be sure to use a name that is informative with the correct extension ( .mtw .xls *etc.*). Examples are

| | |
|---|---|
| Srbx9_5.mtw | Minitab worksheet |
| Srbx9_5.xls | Excel spreadsheet |

In general it is a good idea to use the 'save as' command within a statistical package or spreadsheet to save the worksheet. That way, you choose the name and format for saving your work. At this point, click the 'save as' option under 'file' in the package you are using. Then look through the list of formats into which you can save your worksheet. You will see that each format has its own distinctive extension.

Table G2. Recommended extension for file names.

| Command | extension | Type of file | Complementary commands |
|---|---|---|---|
| MTB > write | .dat | data file | MTB > read, > set, > insert |
| MTB > save | .mpj | worksheet | MTB > retrieve |
| MTB > store | .ctl | commands | MTB > execute |
| MTB > outfile | .out | listing | plato> lpr |

Using extensions in a consistent way will make statistical analysis on the computer easier for you.

| | |
|---|---|
| **write** the data as | 'clam.dat' |
| **save** the worksheet as | 'clam.mpj' |
| **store** repeated commands as | 'clam.ctl' |
| **outfile** the results as | 'clam.out' or 'clam_out.txt' |

If you read data into a spreadsheet, find an error, and correct it in the spreadsheet, then you should correct the error in the source data file. Often this can be accomplished by highlighting the data columns, then saving these in ASCII (text) format back to the original file.

# Section II. SAS on the Unix system.

## Part A. Meet the system.

If you are using SAS you will need to know how to carry out basic tasks on the UNIX operating system. To use SAS you will need to use only a few system level commands, to accomplish the following tasks.

Obtaining a listing of files in a directory
Obtaining help
Deleting a file
Moving a file from one directory to another
Creating a new directory
Sending a file to a printer

If you are a new user of the UNIX system, start by logging onto your UNIX account, entering your username and your password. If you have logged on successfully, you will see some announcements and eventually you will have a symbol at the bottom left of the screen resembling 'plato>'. This is known as the 'prompt'; it is where system commands are entered to perform tasks.

# Section I. SAS on the UNIX system.

## Part A. Meet the system.

If you are used to operating systems with a graphical interface that uses a mouse, then the line oriented (typed) commands in UNIX will take some getting used to. Commands are not listed in a menu, so they have to be actively recalled, rather than recognized from a list. Commands are abbreviated (to save typing longer words frequently), which makes the commands harder to remember.

Table G2. System Commands

| Command | Mnemonic | Equivalent Commands* | Explanation |
|---|---|---|---|
| `cd` | **c**hange **d**irectory | _____ | change from present directory up to a named subdirectory or down to the next directory (`cd ..`) |
| `cp` | **co**p**y** file | _____ | copy a file from one location to another |
| `dir` | **dir**ectory list (vertical) | _____ | list contents of a directory vertically |
| `exit` | **exit** | _____ | exit from your account |
| `ls` | directory **ls**t (horizontal) | _____ | list contents of a directory horizontally (‘`ls -al`’ will list vertically with all file characteristics) |
| `man` | help **man**ual | _____ | obtain help about system commands and topics (i.e. ‘`man mkdir`’) |
| `mkdir` | **m**a**k**e **dir**ectory | _____ | create a new directory |
| `more` | see **more** of a file | _____ | read an ASCII text file (i.e. data file) |
| `mv` | **m**o**v**e file | _____ | move a file from one location to another |
| `rm` | **r**e**m**ove file | _____ | delete a file (warning... this is permanent!) |
| `lpr` | to **l**ine **pr**inter | _____ | send file to printer |

* If your operating system is not UNIX.

~~Now try using your system command to list the files in your account.~~

[Don't forget to check off this item on your list]

Then use a system command to find out how to create a new subdirectory.

## Section II. SAS on the Unix system.
## Part B. Meet the text editor (PICO).

If you are using SAS on the Unix computer, you will need to move data files from their storage place (the server in Biology) to your account on Plato, the Unix computer at MUN. You can do this with the ftp command from your account on Plato. Alternatively, you can move the file to your desktop, then email it to your account on Plato, then extract it to a file on Plato.

Once you have moved a file to your Plato account, you will need to use a text editor on Plato. The following narrative explains how to use the PICO text editor on UNIX. It explains how to start up the text editor, create a new file, and edit a pre-existing file.

At the system prompt type the following:

```
plato> pico
```

***NOTE:*** *UNIX is a case-sensitive operating system, which means that a file named 'Sample.txt' is considered a completely different file than one named 'sample.txt'. The importance of this point will become more apparent when you are introduced to Minitab, the statistical package.*

You are now in the PICO text editor, editing a new file that has yet to be named. The top of the screen will read 'new buffer', meaning this file is new and unnamed. You are now free to type any text you wish in the editing window. When you are finished typing your text, you can leave the editing program and save the file by holding down the <CTRL> key and pressing X. This is written in shorthand from now on as ^X, and appears as such in the commands at the bottom of the text editor. NOTE: Because PICO is user-friendly, many useful commands are displayed at the bottom of the screen, and you are encouraged to refer to those before asking questions, so you can try and answer your questions for yourselves. The text editor will ask you if you wish to save the changes made. Press Y if you do, and then type the name which you wish to call this new file. For consistency, we will call this file 'sample.txt'. Then press <ENTER> . This same procedure can be used to create a data file while at the system prompt (however you'd preferably use sample.dat as the extension, to let yourself know that it is a data file).

We will now use that same file to edit a file that already exists. Type the same command at the prompt as before, however this time put a space after 'PICO' and type the name of the file we just created.

```
plato> pico sample.txt
```

You will notice this time that instead of ‘new buffer’ appearing at the top of the screen, the file name will appear. You can edit the file contents however you like, then save it in the same manner as before, except this time you won’t have to type the file name... it will automatically appear as, in this case, ‘sample.txt’. This same procedure can be used to edit any existing data file while at the system prompt. [save another for your list]

## Section II. SAS on the Unix system.
## Part B. Meet the email utility (Pine).

Another program that has commands independent of the operating system is the email utility. This section shows to use Pine to compose an e-mail message, mail a file, and extract a file. If you already know how to use these commands in E-mail, then skip ahead to the next part, Meet the statistical package.

Here is how to start up the Pine e-mail utility on UNIX, then use it to send a message, to extract a data file sent to you, and to send a data file to a friend.

To start the program from the UNIX system prompt:

```
plato> pine                                                    ↵
```

You will now see the main menu in front of you.

Let's start by composing a message.

*The first step is to press 'C' to compose (as on the screen). Now you are presented with several fields, the first being 'To', where you type the e-mail address of the person receiving the mail (**a84dcs@plato.ucs.mun.ca** is your prof's e-mail address). You can skip the next two fields using the down arrow, and the 'Re' field is optional... it only provides a small phrase describing the content of the mail message. Using the arrow key again, you can move the cursor to the message body. This is where you type the message, or the message you are going to send. In this case, you need only type in your name.*

*To send this message, use the ^X command, by holding down the control key and pressing X. The ^X send command is listed on the bottom of the screen. Then confirm that you want to send the message by pressing 'Y'. Now to exit, press 'Q'.*

Congratulations, you've successfully sent an e-mail message with Pine!

[Another check on your list]

## Section II. SAS on the Unix system
## Part B. Meet the email utility (Pine).

A shortcut for sending an e-mail message is to type the Email addresss of the recipient of the message after the command at the prompt:

```
plato> pine a84dcs@plato.ucs.mun.ca
```

If you use email rather than ftp to move a data file from the server to your Plato account you will need to extract the file and save it using the e-mail utility. To extract a file, start up the E-mail program, as you did in the previous example.

*Instead of hitting 'C' for compose, hit 'I' for 'INBOX', the folder that contains new mail messages. You will see a list of e-mail messages that you have received. Using the up and/or down arrow keys on the keyboard, highlight this message. If there is more than one message, the rightmost field in this list (brief message contents) should contain an identifiable phrase concerning this data file (srbx9_5.dat). When you have highlighted the proper message, press <ENTER>.*

*You will see that the message is composed of two parts, or 'attachments'; the second one is the attached data file you wish to extract into your own directory on the system. To do so, press 'V' to view attachments, and then, as in the INBOX, use the up/down arrow keys to highlight the attachment containing the data file. Then press <ENTER> to see the file and ensure it is the proper one. Now press 'S' to save this attachment. The file name should automatically appear at the bottom of the screen when it asks you what you'd like to call the new file, so just press <ENTER>. Many e-mail systems make it easy to use the same file name that the sender used. In this course, this feature ensures that everyone has the same name on identical data files, saving everyone a lot of time. Now exit as you did earlier, by pressing 'Q' and acknowledging the confirmation.*

To ensure that the file was correctly extracted, use your system command to view the file.

```
plato> more srbx9_5.dat
```

[Score another check mark on your list]

## Section II. SAS on the Unix system
## Part B. Meet the email utility (Pine).

If extracted correctly, the file should appear on your screen looking like this:

```
 7.2  8.8
 7.1  7.5
 9.1  7.7
 7.2  7.6
 7.3  7.4
 7.2  6.7
 7.5  7.2

Data from Box 9.5 Sokal and Rohlf 1995.
Age (days) at first reproduction in two
 genetic groups of Daphnia longispina.
MTB > read 'srbx9_5.dat' c1 c2;
SUBC> nobs=7.
MTB > stack c1 c2     c3
MTB > set  c4
DATA> (0 1)7
DATA> end
MTB > name c3 'age'  c4 'group'
```

To e-mail a data file in your directory to a friend, start by entering the e-mail utility.

*Now begin composing a Pine E-mail message. However in the 'Attachment' field toward the top of the screen, type the name of the data file (located in the directory from which you entered the mail program). If the file is in another directory in your account, press ^J (control key + J), then ^T to obtain a listing of file and directory names. Then using the arrow keys, manually choose the file name. Now send the message as you would a regular e-mail message.*

# Section II. SAS on the Unix System.
# Part C. Meet SAS.

The statistical package SAS has a number of advantages. It can handle larger data sets, it can read complex data sets consisting of several types of records, and it contains a far larger repertoire of statistical procedures. SAS allows one to reorganize or generate new data sets. It can carry out the generalized linear model, one of the most important advances in statistics in the last 3 decades. This greater capacity and flexibility comes at a cost. SAS is harder to use, as one might expect of any complex instrument.

SAS on the Unix system (Plato) does not have a graphics interface. A SAS session typically works with several data sets each with its own columns (variables) and rows (cases). Input to the SAS package consists of data definition statements and procedures; these are organized into a series of commands that are typically submitted as a batch, rather than being typed in one line at a time from the terminal. The output is automatically directed to two different files. One is a list file with results. The other is a log file that reports on the comings and goings of data files, the success or failure of each command, and any errors made in executing the batch file.

Routines are built out of commands, these routines are modified to suite the situation and goals of computation. Assignments in this course can be accomplished by modifying one of the following routines, depending on the situation or goals of computation.

Sections of the routines have been assigned names, which are used in other parts of the lab manual. These generic commands occur in ***boldface italics*** to the right of the routine box.

To see how SAS works, we will read a data file into a structured SAS data set, print the structured data set, and compute descriptive statistics for variables in that data set. This is the same sequence used in the first section of this guide, for PCs. We will use the same data file, srbx9_5.dat, as in the PC section.

# Section II. SAS on the Unix system
# Part C. Meet SAS.

Here is a SAS command file that reads data into a a structured data set, lists data to an output file, and computes a variance. This set of commands is collected together into a file called srbx9_5.sas, which you will create with a text editor.

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A;  infile 'srbx9_5.dat' obs = 7;
  Input SeriesI  SeriesII;

Proc print;

Proc means;
   Var SeriesI SeriesII;
```

***Define Data from file***

***Output to log file***

***Compute*** a variance

When you are done creating the file, check it carefully against the example in the box. Make sure that all of the semicolons are entered as shown in the example. Failure to place semicolons correctly is one the most frequent reasons for the failure of a SAS program to run.

The SAS commands are executed by turning the file over to the SAS package for processing.

```
plato> sas srbx9_5.sas
```

:

## Section II. SAS on the Unix system
## Part C. Meet SAS..

Before executing this set of commands, let's look at what they do. First, the commands that define the session and the SAS dataset.

```
Title 'Daphnia longispina data.';
```
*This command puts labelling information into the output or list file.*
```
Options linesize=72;
```
*This command controls the width of the list file to 72 characters. The default is 132 characters, too wide to see easily on a standard computer screen.*
```
Data A;  infile 'srbx9_5.dat' obs = 7;
  Input SeriesI  SeriesII;
```
*This data definition command defines a data set called A. The Infile subcommand tells the SAS package that the data is to be read from an external file called srbx9_5.dat; only 7 lines are to be read from this file. The Input statement defines two variables, SeriesI and SeriesII.*

***Define data from file***

Next, we'll look at the commands that carry out computations.

```
Proc print;
```
*This command prints out the last data set, A, and sends it to the list or output file.*
```
Proc means;
  Var SeriesI  SeriesII;
```
*This procedural command computes descriptive statistics on two variables, SeriesI and SeriesII, from data set A. These results are sent automatically to the list or output file.*

***Output to log file***

***Compute*** variances

When the command file srbx9_5.sas is turned over to SAS, a log file called srbx9_5.log is automatically created. This has a record of the session, including a listing of any errors encountered in the command file. To view the log file, use a system command or a text editor.

Procedural commands send output to a list file, srbx9_5.lis, unless errors are encountered. If no list file is created, or if something is missing from the file, the log file must be checked to find the errors; the errors are then corrected in the control file (the file with the .sas extension). The corrected control file is turned over to SAS.

To see if a list file was created:

```
plato> ls *.lis
```

# Section II. SAS on the Unix system.
# Part C. Meet SAS..

To look at the results, view the file at the system prompt, i.e., with a system command. Or you can view it with the text editor, or send it to a printer. When you look at this list file, you will see that it contains a printout of the structured (SAS) dataset A, including variable names and a unique observation number for each case (row) in this data set. It also contains the formatted and labelled results from the Proc Means command. The SAS list file does not record the history of the session, only the results of computations.

If you are using SAS on a remote system rather than your own computer, you can move the list file from the system to computer using email or a file transfer program such as ftp. Once the SAS list file is on your computer, you can send it to your local printer or pull it into your local editor, in order to mark and extract pieces of it to display in reporting the results of your analysis with SAS. If you are going to print sections of a SAS list file, be sure to format these sections in a non-scalable font such as Courier 10. If you use a scalable font the results in the list file will be distorted and nearly unreadable. In this manual, a scalable font (Times Roman) is used outside the boxes, while a non-scalable font (Courier) is used inside boxes, to retain spacing and legibility.

* * *

The data definition commands in SAS permit extraordinary flexibility in reading and generating new data sets. To demonstrate this flexibility, let's re-organize the *Daphnia longispina* data into a new data set consisting of two variables: age at beginning of reproduction (days) and series (either I or II). This is the data structure that would be used to analyse this data with a General Linear Model, demonstrated later in the course.

Here are the SAS commands to read and re-organize the data. They are stored in a command file (either a revised version of srbx9_5.sas, or a new command file). The file is created with a text editor, either your own, or the one that some versions of SAS supply.

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A;  infile 'srbx9_5.dat' obs = 7;
  Input SeriesI SeriesII;

Data B; Set A;
  Age = SeriesI; Series = 'I'; output;
  Age = SeriesII; Series = 'II'; output;

Proc print;

Proc means; By Series;
   Var Age;
```

***Define data from a file***

***Reorganize***

***Output to a log file***

***Compute a variance***

## Section II. SAS on the Unix system.
## Part C. Meet SAS.

As before, let's look at the commands before executing them. First, the data statement used to reorganize the data.

```
Data B; Set A;
  Age = SeriesI; Series = 'I'; output;
  Age = SeriesII; Series = 'II'; output;
```

***Reorganize***

This creates a new data set labelled B. The Set command reads the variables from data set A into B, one line at a time. All subsequent commands occur within a loop that is executed for each line of the input data set. When the first line is read, the first value of SeriesI is assigned to the variable Age in the first line of data set B. The variable Series is assigned a value of I in this first line. The output statement forces output to the reorganized dataset B. The next line assigns the first value of SeriesII to the variable Age in the second line of data set B. The variable Series is assigned a value of II in this second line. SAS has now reached the end of the Data statement, so it goes back and executes the statement again. It reads the second line of dataset A. The first value is assigned to SeriesI then output to the third line of data set B. SAS continues in this fashion, looping repeatedly through the Data statement, until all of the data is read from data set A, or until SAS is otherwise told to stop.

These data definition statements create a new data set that stacks SeriesI and SeriesII into one column, then creates a new variable the identifies the source of the *Daphnia* age measurements.

Next, the Procedural statements that print the re-organized data set and compute summary statistics on it.

```
Proc print;
       This prints the contents of the last data set, B.
Proc means; By Series;
   Var Age;
       This computes summary statistics on the variable Age for each of
       the two source groups.
```

***Output to log file***

***Compute*** a variance

The command file is created with the editor. This is a new version, so it will be given a slightly different name (srbx9_5b.sas), by adding the letter b to the file name. This command file is turned in for execution as before.

```
plato> sas srbx9_5b.sas
```

SAS automatically creates a log file called srbx9_5b.log . SAS creates a list file called srbx9_5b.lis, if there are no fatal errors in the command file.

# Section II. SAS on the Unix system.
# Part C. Meet SAS.

.

This reorganized data file in SAS can be saved as a system file, rather than being generated again and again. Here is a command file that places the reorganized data into a file called srbx9_5b.dat, created by SAS. Note the consistent use of the extensions .dat and .sas to distinguish data file from command files concerning the same data set.

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A;  infile 'srbx9_5.dat' obs = 7;
  Input SeriesI  SeriesII;

Data B; Set A;
  Age = SeriesI; Series = 'I'; output;
  Age = SeriesII; Series = 'II'; output;

Data C(Keep = Age Series);
  Set B;
  Filename srOUT 'srbx9_5b.dat';
  File srOUT;
  Put @6 Series  @2 Age;

Proc Print;
```

***Define data from a file***

***Reorganize***

***Define data to a file***

The only new code here is the third data statement, which creates dataset C. This new set contains only two variables (Age and Series) in contrast to dataset B, which contains four variables (SeriesI, SeriesII, Age, and Series). Both Age and Series have been written to a file. The Put statement is much like an input statement in form, except that it writes rather than reads data. This Put statement writes the variable Age starting at card column 6, then writes the variable Series starting at card column 2. This shows the flexibility of SAS in reading, reorganizing and writing data.

A longer list of SAS commands, which carry out computations needed for labs and problem sets, can be found elsewhere in the lab manual.

# Guide to Computing
# Minitab commands used in labs (mtbcode.out)

A full listing of Minitab commands can be found by invoking the HELP command while running Minitab. A reference card, with listing of available commands, can be purchased in the University Bookstore.

Use the HELP command (or reference card) much as you would a dictionary or thesaurus, to find out how a command works or find a command to accomplish the calculation at hand.

The commands listed here are for common routines used in this course:

- Reading Data into Minitab
- Writing Data and Output Files from Minitab
- Summarizing data
- Re-organizing Data for Analysis by the General Linear Model
- Executing the General Linear Model
- Calculating Residuals
- Using Residuals to Check Assumptions
- Randomization Tests with Minitab

These routines are built up out of commands, just as sentences are built out of words. Most of the assignments in this course can be accomplished by modifying one of the following routines, depending on the situation or goals of computation.

Many routines have been assigned a name, which is used in other parts of the lab manual. The name of the routine occurs in ***boldface italics*** to the right of the routine box.

**Reading Data into Minitab.**

Data are read from a file (in quotes) into columns. For the data files in this course, you will need to state the number of lines of data, because information about the data is listed below the data in the same file.

```
MTB > read 'srbx9_5.dat' c1 c2;
SUBC> nobs = 7.
MTB > name c1 'age(I)' c2 'age(II)'
```

***Define Data from file***

Data are also typed directly from the keyboard.

```
MTB > set into c1
DATA> 2.68 2.60 2.43 2.90 2.94 2.70 2.68 2.98 2.85
DATA> end
MTB > set c2
DATA> 2.36 2.41 2.39 2.85 2.82 2.73 2.58 2.89 2.78
DATA> end
MTB > name c1 'N(B)' c2 'N(13)'
```

***Define Data from keyboard***

**Writing Data and Output Files from Minitab**

Data in columns can be saved by writing to a named system file, as in the box above.

```
MTB > write [to] 'srbx1311.dat' c1 c2
```

***Define Data to file***

This command prints output directly to the screen.

```
MTB > print c1 c2
```

***Output to screen***

Most commands send output to the screen automatically.

```
MTB > describe c1 c2
MTB > histogram c1 c2
```

Results of computations need to be printed to the screen

```
MTB > let k2 = std(c1)
MTB > let k2 = k2*k2
MTB > print k2
```

***Compute a variance***
***Output to screen***

Output that appears on the screen can be written to a named system file, this file can then be printed out, or moved to another computer.

```
MTB > outfile 'srbx9_5.out'
MTB > print c1 c2
MTB > describe c1 c2
MTB > nooutfile
```

***Outfile open***

***Outfile closed***

**Summarizing Data**

The following 4 commands calculate and display descriptive statistics on data in column 1

```
MTB > describe c1
MTB > histogram c1
MTB > dotplot c1
MTB > mean c1
```

This next routine calculates a cumulative relative relative frequency
CRF(C1 < 2) = 0.09
from the following data:

```
1  2  3  3  4  4  4  5  5  6  7
```

***CRF(C1 < 2)***

```
MTB > histogram c1;
MTB > start 2.
MTB > let k1 = 1                    <--from screen display
MTB > let k2 = 11                   <--from screen display
MTB > let k3 = k1/k2
MTB > print k3
```

**Re-organizing Data for Analysis by the General Linear Model**

The following commands reorganize data from tabular format (7 columns) to model format (2 columns, response and explanatory variable).

***Reorganize***

```
MTB > read 'srtab8_1.dat' c1-c7;
SUBC> nobs = 5.
MTB > stack c1-c7 c8
MTB > name c8 'wlength'
MTB > set c9
DATA> (1 2 3 4 5 6 7)5
DATA> end
MTB > name c9 'groups'
MTB > print c8 c9
```

## Executing the General Linear Model, Calculating Residuals

ANOVA designs

```
MTB > anova 'wlength' = 'groups';
SUBC> fits c10;
SUBC> residuals c11.
MTB > name c10 'fits' c11 'res'
```

***Run GLM ANOVA***

```
MTB > glm 'wlength' = 'groups';
SUBC> fits c10;
SUBC> residuals c11.
MTB > name c10 'fits' c11 'res'
```

***Run GLM ANOVA***

Regression designs

```
MTB > read 'ryder.dat' c1 c2;
SUBC> nobs = 23.
MTB > name c1 'area'  c2 'yield'
MTB > regress 'yield' 1 predictor 'area';
SUBC> residuals c10.
MTB > name c10 'res'
MTB > let k1 = 837382                    <---from screen display
MTB > let k2 = 1.45                      <---from screen display
MTB > let c11 = k1 + k2*'area'
MTB > name c11 'fits'
```

***Run GLM Regression***

```
MTB > read 'ryder.dat' c1 c2;
SUBC> nobs = 23.
MTB > name c1 'area'  c2 'yield'
MTB > glm 'yield' = 'area';
SUBC> covariate 'area';
SUBC> fits c9;
SUBC> residuals c10.
MTB > name c9 'fits' c10 'res'
```

***Run GLM Regression***

## Using Residuals to Check Assumptions

A. linear relation of response to explanatory ? (bowls and arches)

```
MTB > plot 'res' vs 'fits'
```

***GLM linear?***

B1. Do errors sum to zero ?

Note that this assumption is listed for completeness. There is no need to check it if fitted values are calculated via least squares, as in statistical packages such as Minitab.

```
MTB > let k1 = sum('res')
MTB > print k1
```

B2. Are errors independent ?

```
MTB > let c20 = lag('res')
MTB > plot c20 'res'                          <--graphical analysis

MTB > runs 'res'                              <--runs test (optional)

MTB > corr c20 'res'                  <--autocorrelation, lag 1(optional)

MTB > acf 'res'                     <--autocorrelation, many lags (optional)
```

***Errors independent?***

B3. Are the errors homogeneous ?
(no cones facing left or right)

```
MTB > plot 'res' vs 'fits'
```

***Errors homogeneous?***

B4 Are the errors normally distributed ?

```
MTB > hist 'res'                              <---graphical analysis
MTB > nscores 'res' c30

MTB > plot c30 'res'                      <--- 2nd graphical analysis

MTB > rootogram 'res'                     <---3rd graphical analysis,
                                                shows confidence limits
```

***Errors normal ?***

## Randomization Tests with Minitab

Statistic is mean (compared to zero)

***Calculate statistic***

```
MTB > let k1 = mean(c1)
```

***Set up***

```
MTB > nooutfile
MTB > set into c2
DATA> -1 1
DATA> end
MTB > let k2 = 0
```

***Create control file***

```
MTB > store 'ran.ctl'
STOR> let k2 = k2 + 1
STOR> sample 12 c2 c3;
STOR> replace.
STOR> let c4 = c1*c3
STOR> let k3 = mean(c4) - k1
STOR> let c5(k2) = k3
STOR> end
```

***Check file***

***Run file***

```
MTB > execute 'ran.ctl'
MTB > execute 'ran.ctl'  1000
```

***CRF(c5 > k1)***

```
MTB > outfile
MTB > histogram c5;
SUBC> start k1.
```

p-value is % of distribution greater than statistic in k1

## Randomization Tests with Minitab

Statistic is difference between two means.

***Calculate statistic***

```
MTB > let k10 = mean(c6) - mean(c7)
MTB > print k10
MTB > let c25(1) = k10
MTB > name c25 'F(st)'
```

***set up***

```
MTB > nooutfile
```

***Create control file***

```
MTB > store into 'srbx9_5.ctl'
STOR> stack c1 c2 into c3;
STOR> subscripts c4.
STOR> sample 14 times from c3 into c5
STOR> unstack c5 c6 c7;
STOR> subscripts c4.
STOR> let k3 = mean(c6) - mean(c7)
STOR> stack c25 k3 into c25
STOR> end
```

***Check file***

***Run file***

```
MTB > execute 'srbx9_5.ctl'
MTB > print 'F(st)'
MTB > execute 'srbx9_5.ctl' 10 times
MTB > print 'F(st)'
MTB > execute 'srbx9_5.ctl' 500
```

***CRF(c25 > k10)***

```
MTB > outfile
MTB > hist 'F(st)'
MTB > hist 'F(st)';
SUBC> start k10.
```

p-value is cumulative relative frequency (%) of outcomes in c25 that are **larger** than value in k10  CRF(c25 > k10) = %

## Randomization Tests with Minitab

Statistic is mean squared error (MSE)

***Calculate statistic***

```
MTB > regress 'wloss' on 1 predictor 'humidity';
SUBC> residuals c5.
MTB > name c5 'res'
MTB > let k10 = std('res')**2
MTB > print k10
```

***Set up***

```
MTB > let c25(1) = k10
MTB > name c25 'F(st)'
MTB > nooutfile
```

***Create control file***

```
MTB > store into 'srbx14_1.ctl'
STOR> sample 9 times from 'wloss' into c8
STOR> regress c8 1 'humidity';
SUBC> residuals c5.
STOR> let k3 = std('res')**2
STOR> stack c25 k3 into c25
STOR> end
```

***Check file***

```
MTB > execute 'srbx14_1.ctl'
MTB > print 'F(st)'
MTB > execute 'srbx14_1.ctl' 10 times
MTB > print 'F(st)'
MTB > execute 'srbx14_1.ctl' 500
```

***Run file***

***CRF(c25 < k10)***

```
MTB > outfile
MTB > hist 'F(stat)'
MTB > hist 'F(stat)';
SUBC> start k10.
```

p-value is cumulative relative frequency (%) of outcomes in c25 that are **smaller** than value in k10. CRF(c25 < k10) = %

# Computing Guide
# SAS commands used in labs (SasCode.out)

A full listing of SAS commands can be found in the manuals for this package, at the computer centre in the Henrietta Harvey building. SAS also provides on-line help for commands.

SAS routines are built out of commands, these routines are modified to suite the situation and goals of computation. Assignments in this course can be accomplished by modifying one of the following routines, depending on the situation or goals of computation.

SAS routines are saved in control files (with extension .sas) which are turned over to the package for processing.

Sections of the routines have been assigned names, which are used in other parts of the lab manual. These names occur in ***boldface italics*** to the right of the routine box.

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A;  infile 'srbx9_5.dat' obs = 7;
  Input SeriesI  SeriesII;
Proc print;
Proc means;
   Var SeriesI SeriesII;
```

***Define Data from file***

***Output to log file***

***Compute*** a variance

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A;  infile 'srbx9_5.dat' obs = 7;
  Input SeriesI SeriesII;
Data B; Set A;
  Age = SeriesI; Series = 'I'; output;
  Age = SeriesII; Series = 'II'; output;
Proc print;
Proc means; By Series;
   Var Age;
```

***Define data from a file***

***Reorganize***

***Output to log file***

***Compute*** a variance

## SAS commands used in labs

```
Title 'Daphnia longispina data.';
Options linesize=72;
Data A;  infile 'srbx9_5.dat' obs = 7;
  Input SeriesI  SeriesII;
Data B; Set A;
  Age = SeriesI; Series = 'I'; output;
  Age = SeriesII; Series = 'II'; output;
Data C(Keep = Age Series);
  Set B;
  Filename srOUT 'srbx9_5b.dat';
  File srOUT;
  Put @6 Series  @2 Age;
Proc Print;
```

***Define data from a file***

***Reorganize***

***Define data to a file***

```
proc glm data=a outstat=z0;
      class d;
      model r=d / ss3;
      output out=out1 r=res p=pred;
      proc plot data=out1; plot res*pred/vref=0;
      proc univariate data=out1 normal; var res;
```

***Run GLM*** (Anova)

Note that the glm command in SAS assumes that all explanatory variables are regression variables. ANOVA (classification variables are defined by the 'class' statement.

```
proc glm data=a outstat=z0;
      model r=d / ss3;                          Etc..
```

***Run GLM*** (Regression)

```
proc glm data=a outstat=z0;
      class da;
      model r=da db da*db / ss3;                Etc...
```

***Run GLM*** (Ancova)

On the next page is a program written by Tammo Bult, to compute p-values by randomization, for the general linear model, when the assumptions concerning residuals are not met.

```
options nocenter nodate linesize=75 pagesize=35;

/* READ DATA */
   data a;         infile datxd2 missover;
         input d 2 r 1;

/* SPLIT UP DATASET IN 2 SEP. DATASETS ON EXPL.VAR.(1) AND RESP.VAR.(2) */
   data ar; set a; keep r; /* response variable */
   data ae; set a; drop r; /* explanatory variables */

/* PERFORM INITIAL ANALYSIS */
   proc glm data=a outstat=z0;        class d;
    model r=d / ss3;
    output out=out1 r=res p=pred;
   proc plot data=out1; plot res*pred/vref=0;
   proc univariate data=out1 normal; var res;

/* SET NUMBER OF RANDOMIZATIONS AND NUMBER OF OBSERVATIONS */
   %let nrand=434;
   %let nobs1=20;

/* MISCELLANEOUS */
   data y1; input _SOURCE_ $ F; cards;
                              .
                              ;

/* MACRO THAT PERFORMS RANDOMIZATION SCHEME */
   %macro loop;
   %do l=1 %to &nrand;
    data b; retain seed &l; do i=1 to &nobs1;
         v1=ranuni(seed); output; end; keep v1;
    data c; merge b ar;
    proc sort data=c; by v1;
    data c; set c; keep r;
    data d; merge c ae;
   /* analysis on randomized data */
    proc glm data=d outstat=z1 noprint;
         class d;
         model r=d / ss3;
   /* organize F values calculated via this analysis */
    data z1; set z1;
         keep _SOURCE_ F;
         if F=. then delete;
    proc append base=y1 data=z1 force;
   %end;
   %mend loop;
   %loop;

/* CALCULATE P VALUES */
   data z0; set z0;          F0=F;
                  if F=. then delete;
                  keep _SOURCE_ F0;
   data y1; set y1;
    if F=. then delete;
   proc sort data=z0; by _SOURCE_;
   proc sort data=y1; by _SOURCE_;
   data p0; merge y1 z0; by _SOURCE_;
   data p0; set p0;          if F >F0 then do; nh=1; end;
                  if F<=F0 then do; nh=0; end;
   proc means data=p0 noprint; var nh; by _SOURCE_;
    output out=p1 n=nobs sum=nh;
   data p1; set p1; pc=nh/nobs;
   proc print data=p1; var _SOURCE_ pc;
endsas;
```