

HANDBOOK FOR TEACHING ASSISTANTS

DEPARTMENT OF STATISTICS

UNIVERSITY OF CALIFORNIA, DAVIS

**A HANDBOOK FOR TEACHING ASSISTANTS
IN THE DEPARTMENT OF STATISTICS AT
THE UNIVERSITY OF CALIFORNIA, DAVIS**

Revised June, 2007

The original work for this handbook was supported through a UIIP grant from the Teaching Resources Center at UC Davis. Revisions made in 1997, 1999, and 2000 were supported through Minigrants from the TRC's TA Training Grant Program. The handbook has benefited from contributions from Professors Alan Fenech, Frank Samaniego, Jessica Utts and Jane-Ling Wang, and from graduate students Scott Bartell, Sam Chong, Sandra Mohammed, Gene Pennello, Justin Smith and Eric Sues.

Note: This handbook is written as the Department of Statistics' supplement to the TRC's TA Handbook and the Graduate Student Handbook published by the Office of Graduate Studies.

TABLE OF CONTENTS

Part I: General Guidelines for the Teaching Assistant	4
1. Introduction.....	4
2. Conducting a Discussion Section	5
Positive Attitude.....	5
Preparation	5
Practical Advice for Conducting a Discussion Section.....	6
How to Encourage Discussion	7
Other Tips for Involving the Class.....	8
Common Criticisms of Teaching Assistants' Performance.....	8
3. Giving a Lecture	9
4. Conducting Projects in Discussion Sections	9
5. Grading	12
Grading a Problem.....	12
Grading Projects	15
Handling Student Questions about Grading	16
Indications of Academic Dishonesty	16
6. Monitoring Examinations	16
7. Office Hours	18
8. Solution Sets and Handouts	19
Homework Solutions	19
Handouts	21
9. Teaching Your Own Class	22
Part II: Technology	24
1. Creating a Class Webpage	24
2. Scanning Documents.....	28
3. Introduction to Computing Packages.....	30
MINITAB	31
SAS.....	36
R	42
LaTeX	55
Part III: Resources	60
1. Department and Campus Contacts.....	60
2. Teaching References	62
Appendices (from Student Judicial Affairs)	63
◆ UC Davis Code of Academic Conduct	http://sja.ucdavis.edu/files/cac.pdf
◆ Reporting Academic Misconduct	http://sja.ucdavis.edu/files/reporting.pdf
◆ Disruption in Class	http://sja.ucdavis.edu/files/disrupt.pdf
◆ Tips to Prevent Cheating	http://sja.ucdavis.edu/files/tips.pdf
◆ Confronting In-Progress Cheating	http://sja.ucdavis.edu/files/confront.pdf

Part I: General Guidelines for the Teaching Assistant

1. Introduction

The suggestions contained in this document are designed to help graduate students in Statistics perform effectively as teaching assistants (TA's) in Statistics right from the start. It should be particularly useful for first-time teaching assistants. In addition to this handbook, experienced TA's can be especially useful as sources of information about handling TA responsibilities; you should not hesitate to seek out their advice on carrying out particular responsibilities.

As a teaching assistant, your job is to assist the teacher in conducting a course. Before the beginning of each quarter you will find out for whom you will be a TA. See these individuals right away to learn exactly what your TA duties will be for that quarter. Duties will vary from instructor to instructor and from course to course. Promptly find out which days during the quarter you need to reserve for TA jobs (e.g., grading exams), and compare these dates with those that you need for other major jobs (e.g., taking your own exams). Within the first week of the quarter, work out any conflicts. Make a habit of checking in with the instructors you work for at least once a week throughout the quarter, just to keep in touch. *Some instructors may require that you attend the lectures for the class. Whether or not it is required, it is strongly recommended that you do so if you have not been a TA for this instructor and course before. It is the only way you can know what the students in the class are being taught.*

Statistics TA's often are responsible for the following teaching jobs: conducting discussion sessions, giving lectures, monitoring and grading exams, holding office hours constructing solution sets, web pages and computer handouts and corresponding with students by email. In the pages that follow, we will discuss each of the TA's jobs.

These jobs will challenge not only your technical skill in statistics, but also your skills as a thinker, writer, speaker, and conversationalist. It will require planning, care, commitment, and time. You need to spend a reasonable amount of all of these on your TA duties, while making sure you do not hinder your progress in your own schoolwork.

There are many important benefits connected with doing a good job as a TA in Statistics. We discuss three of those benefits here. First, you will develop your communications skills. If you are like most people, your future occupation will require you to be able to think on your feet, to write and speak well, and to listen. Carrying out TA duties exercises these skills, and TA evaluations should provide you with some evaluation of your skills and ideas for improving them. Do not underestimate the importance of developing communication skills while you are a graduate student. Second, as a TA, you will work directly with faculty members and get to know them better than you would otherwise. This should result in a better education for you as a graduate student. Simultaneously, the faculty will know you better than otherwise, as your TA work necessarily will leave an impression on the faculty. This is a double-edged sword, as the impression may be either good or bad. It is important that this impression be a good one. For example, letters of reference will be easier to obtain and of much better quality if you leave a good impression on the faculty than if you do not. Third, you will improve your understanding of course material. This is a direct consequence of writing quality solution sets and preparing quality lectures and discussions, since you need to be accurate when preparing these.

2. Conducting a Discussion Section

Discussion sections are part of many statistics courses. These sections provide students with the opportunity to express their ideas about course material in an open forum of fellow students and the TA, through the active process of asking and answering questions. They provide the chance for students to discuss specific points. This should help students learn course material, become actively engaged in the course, and systematically exercise their powers of expression and rational thought. Sometimes, a discussion will be used to help the students learn how to use a statistical computing package that complements the course. Sometimes it will be used for hands-on activities. The instructor for the course usually will provide you with instructions for what to do in the discussion sections. If you are not sure, do not hesitate to ask!

Here are some suggestions that should help you conduct an effective discussion section.

Positive Attitude

In discussions, it is important that you project a positive attitude toward the students and show interest in and commitment to the subject. As discussion leader, you are responsible for the tone of the discussion. Aim to express to the students that you are willing and able to help them in the course and that you want and expect them to learn. In the case that the students do not understand the material, encourage them to keep trying. Keep a positive attitude toward your students and encourage them to ask questions and volunteer answers. Encourage them to use your office hours or send you email if they come up with questions outside of the discussion section. Always treat students with proper respect. Note, though, that it is a mistake to be too familiar with students. If you are the instructor or a TA, you should not become personally involved with a student taking that class. If a student comes to regard a TA as a personal friend, the student may come to expect more help and time from the TA than is reasonable. Also, if you favor particular students, other students may feel that they are at an unfair disadvantage. Avoid any conflict of interest.

If you plan to use email, give out your address during the first discussion session. You should also state how often you check your mail, so that the students will have an idea how long it will take for their questions to be answered.

Preparation

Before you lead a discussion section, you should know what you want to achieve during that hour. You should be completely familiar with the main points and related examples that you plan to cover. Most of the time, the instructor will tell you what topics to cover. Sometimes, however, an instructor will leave it to you to decide how to use the discussion section. If the instructor lets you decide what to cover, you may aim to explain difficult or unclear material from the lectures or problem sets. Figure out in advance what points you want to make and how long it takes to make them. If you think you will have extra time, prepare to take questions from the students. Make sure that you know what kinds of questions the instructor does and does not allow you to answer, though. If you are allowed to answer questions about the homework, be ready to answer any of the assigned problems. In case there are no questions, prepare some additional material.

It is a good idea to write out completely what you intend to present in a section. Prepare the material twenty-four hours ahead of time, if possible. This puts the material clearly into your own mind and

also gives you a chance to discover errors in your reasoning or in your planning. Then you can correct your errors before the discussion section begins. It is ineffective and embarrassing to discover your errors on the board during a section. (By the way, it is a big mistake to try to bluff your way through an error. Students can easily recognize that you are faking; this damages your credibility and your reputation as a teacher. If you make an error, simply correct it and proceed from there.)

It is also important to plan what you want to write on the board; it is not feasible to write your entire presentation on the board. What you do write should be something useful for later reference: a student, reviewing notes for an exam, should be able to look at the notes and remember something about what was discussed on that day. Make examples interesting; you need not take all of your examples from the book. There are a variety of textbooks used in the Department. If you need new examples, textbooks other than the one used for your class can be a good resource.

If you plan to discuss computing, prepare handouts so that the students can follow along with your discussion. There may already be some handouts prepared by previous TA's. If there are, you may adapt them for use with the current class, making necessary changes.

One purpose of your section presentations may be to clarify points. Thus, it is usually a mistake to use notation different from that presented during the lecture. Be familiar with what material has been covered in the lectures and readings and with the notation used therein. In most classes, you can do this efficiently by attending lectures occasionally, by reviewing current problem sets and reading assignments, and by reviewing class notes. If the instructor wants to meet with you weekly to discuss class material, you can use those meetings to discuss what notation you will use. Otherwise, consult with the instructor if you have any questions concerning notation.

Practical Advice for Conducting a Discussion Section:

The following suggestions give practical advice for conducting a discussion section.

1. When explaining a problem, proceed in a systematic step-by-step fashion, interacting with the students along the way. Do not just read your notes! Tips for encouraging interaction are given in the next section.
2. Handle questions of value to the course in the discussion; politely defer other questions to an after-class meeting.
3. Remember that it is very easy to go too quickly in explaining a problem, especially in a beginning course like Statistics 13. Aim for simplicity and clarity in your explanations.
4. Learn good blackboard technique:
 - a. write legibly
 - b. write down enough so that the resulting student notes are useful
 - c. work systematically from left to right
 - d. don't erase material before students have had time to write it down

5. If your discussion of an important point or problem does not work, promptly prepare and make available a handout on the point. Distribute the handout as soon as possible.
6. If you don't know how to answer a question, don't try to bluff. Be honest, and tell the students you will find out and let them know by e-mail or at the next discussion.
7. After the discussion section, take note of the questions that you need to answer and the points that you need to clarify. Be sure to take care of them before the start of the next discussion section.

How to Encourage Discussion¹

It is more interesting for you and for the students if you encourage them to get involved in the discussion. Here are some time-tested methods for encouraging discussion:

1. Rather than telling the students everything, involve them in discussion. Ask questions. After you ask a question, resist the temptation to answer it immediately yourself. Although it will seem like an eternity at first, you should wait at least 30 seconds before jumping in to help out, especially for thought-provoking questions.
2. Ask open-ended rather than yes-or-no kinds of questions. It's an old journalism trick. If you ask for a yes-or-no answer, that's all you'll get. For instance, rather than asking, "Would you reject the null hypothesis in this situation?" ask "Who can tell me what decision would be made in this situation, and why?"
3. If no one volunteers an answer after a decent wait, call on someone who seems to be interested in answering. Many students are shy about volunteering, but are quite willing to participate when asked. You can tell who is interested because they don't look away when it appears that you are about to call on them.
4. If the class is truly stuck, say something like "I guess I didn't word that very well. Let me ask it differently."
5. Ask questions for which there is more than one correct answer. Remember that you are trying to stimulate discussion, not give an oral exam.
6. Don't allow a few students to dominate all of the discussions. Call on others, or say "How about someone who hasn't spoken recently?"
7. If someone gives an answer that is completely wrong, don't put him or her down, even with your body language. If you think you can steer that person to the right answer quickly, try to do so. Otherwise, let them off the hook by saying something like "Let's try again ... anyone else want to give it a try?"

¹ Material in this section is from *Teaching 'Seeing Through Statistics': An Instructor's Resource Manual*, Jessica M. Utts, ©Duxbury Press, 1999

8. If it seems like no one is getting the right answer, ask the class what question they *think* you have just asked. Sometimes you will discover that what you thought you were asking is not what was understood by the class.

9. Try to learn the students' names. In a small class you can do so fairly easily. In a larger class, ask students to state their name when answering a question, until you know them. Some instructors require each student to see them during office hours at least once so they can get to know them. Once you know their names you can easily choose students to call on, and they will be more willing to cooperate if they don't feel anonymous.

Other tips for involving the class

Students often learn best from each other. If your class is the right size you can have them break into small groups to discuss questions you provide. Then reconvene the class and have each group report on one of the questions.

Another idea is to have pairs of students explain things to each other, since the best way to understand something is to try to teach it. Have students choose a partner (or two if necessary). Tell them you are going to explain something to the class, after which one partner then has to explain in his or her own words to the other one. After they have done this, call on someone to explain the idea to the class. Then explain another idea, and have them switch roles. The pressure of possibly being called upon to explain will encourage them to listen carefully and to formulate their own understanding of what you have said. It will also provide excellent feedback to you to learn whether or not you are getting your ideas across.

Common Criticisms of Teaching Assistants' Performance

The following negative comments are frequent on TA evaluations:

- Sloppy handwriting with poor blackboard organization.
- Ineffective/inaudible speaking style.
- Blackboard erased too rapidly to let students copy the material down.
- Inadequate interaction with the class.
- Focus is entirely on formula manipulation.
- Distracting personal habits.

Notice that most of these are easily corrected with a bit of practice. You can ask the Teaching Resources Center, or a friend, to observe your class and give you feedback on whether these issues are problematic for you.

3. Giving a Lecture

As a TA it is likely that you will be called upon to give lectures from time to time. You might substitute for the instructor, for example, or be asked by the instructor to give a lecture in a discussion section. Although some of the principles for giving discussion sections can be applied to lectures, the purpose of a lecture is different from that of a discussion section. Usually, the purpose of a lecture is to communicate new material to an audience, which is different from the clarifying, reviewing and open discussion of already-introduced material that goes on during most discussion sections. A lecture is a more active presentation of the topics. Also, whereas a discussion section tends to cover several different points, a lecture should present the material in a more continuous flow.

Generally, the audience at a lecture will be relatively quiet, since they will be uninformed, uncertain and perhaps apathetic about the material you are discussing. For example, if you are introducing the Binomial distribution to a class of Statistics 13 students, you can be sure that most of the students in class have only the vaguest idea what you are really talking about. You shouldn't be disappointed if some students don't follow your lecture, since it is just not possible for many students to understand new technical material as it is being presented. You should be concerned, though, if students are inattentive and are not even trying to follow you.

So, if students don't follow you during a lecture, what is the point of a lecture? There are two: First, to generate curiosity and enthusiasm about the material you are introducing and a desire to study it. Second, to present a planned amount of new material to the observing students with the object of leaving the students with a record that is good enough to enable the student to learn the material through later study of the record. The record amounts to: the student's notes on your board work (definitions, general results, worked examples, and so on) and recollection of what you said, together with any of your handouts pertaining to the lecture. You are responsible for producing a good and interesting record; it is the individual student's responsibility to learn the material using (among other things) that record. With these two points in mind, the specific suggestions on preparing for a discussion section should help you prepare a quality lecture (positive attitude, preparation, etc.).

4. Conducting Projects in Discussion Sections

Some instructors may ask you to use the discussion section to conduct an in-class project. These can be fun for you and for the students, or they can be pandemonium! It is even more important to be prepared if you are conducting a project. Make sure you have all the supplies you will need and that you pace the class so that you don't run out of time.

It is also important, when conducting projects, to make sure the class understands the purpose of the project. Giving them a "menu" to follow without discussing what they can learn from it is a waste of their time. Make sure you spend a few minutes at the beginning of the class explaining the goals of the project. Ideally, you should explain that the project is a simplistic example of what statistical methods might be used for solving problems or answering questions in the real world.

On the next two pages are: an example project, the learning goals for the project, and the "team recording sheet" that would be used for students to record their work.

Here is an example of a project, taken from the *Instructor's Resource Manual for 'Mind on Statistics'*, Jessica M. Utts and Robert F. Heckard, ©Duxbury Press, 2004:

Project 2.2: Team constructed questions about relationships

Supplies:

- N index cards or slips of paper of each of T colors [N = number of students in class, T = number of teams.]
- $2T$ overhead transparencies and pens

Students should work in teams of 4 to 6.

Step 1: Each team formulates a question about two variables for which they want to know if there is relationship, such as whether someone is a firstborn (or only) child and whether they prefer indoor or outdoor activities. (Recent research suggests that firstborns prefer indoor activities and later births prefer outdoor activities.) They can be two categorical variables, or one categorical and one quantitative variable.

Step 2: Each team is assigned a color, from the T colors of index cards. For instance Team 1 might be blue.

Step 3: Each team has a chance to ask the whole class its two questions. The team writes the questions on an overhead transparency and displays them. Students write their answers to them on the index card corresponding to that team's color. For instance, all students in the class write their answers to Team 1's questions on the blue index card.

Step 4: After each team has asked its questions and students have written their answers, the cards are collected and given to the appropriate team. For instance, Team 1 receives all the blue cards.

Step 5: Each team tallies, summarizes and prepares a graphical display of the data for their questions. The results are written on an overhead transparency.

Step 6: Each team presents the results to the class.

The learning goals for this project are:

1. Identify types of data and appropriate summaries for them
2. Use data wisely to answer questions, including numerical and graphical summaries.
3. Understand that natural variability exists in any set of measurements
4. Understand that variability in measurements can be a combination of natural variability and measurement error. (You can discuss this concept even if no teams use variables recorded with measurement error.)
5. Understand that different quantitative measurements have different shapes.

PROJECT TEAM FORM

Project #1: Team constructed questions about relationships.

TEAM MEMBERS:

- | | |
|----------|----------|
| 1. _____ | 4. _____ |
| 2. _____ | 5. _____ |
| 3. _____ | 6. _____ |

TEAM INSTRUCTIONS:

1. Choose two variables for which you think there might be an interesting relationship. At least one of them should be a categorical variable; the other can be categorical or quantitative. Categorical variables should have only two categories, so that you can finish on time.

Examples:

"(1) Would you rather party or study? (2)What is your GPA?"

"(1) Are you male or female? (2)In choosing a romantic partner are looks or personality more important?"

2. You will be assigned a "team color." Write your team color and your two questions on an overhead transparency. When your team is instructed to do so, display your questions and team color to the class. Class members will record their answers to your questions on the index card they have been given with your team color. Collect the cards.

List the two questions you chose below:

Question 1:

Question 2:

3. Create appropriate numerical and graphical summaries to display on your team's overhead transparency. Write a brief summary of your findings below. Turn in this sheet.

5. Grading

One responsibility often given to teaching assistants is the grading of problems on midterm and final examinations. This important job can be time-consuming, difficult, and thankless. In general, students at Davis are very concerned about their grades, and in a Statistics course their grades are likely to depend heavily on midterm and final exam scores. Indeed, grades are important. It is therefore fair for students to expect grading to be thoughtful, thorough and consistent from one student's paper to the next.

Generally, you should strive to be fair, consistent and reasonable in your grading. The objective is to give a student proper amount of credit for their work: you should give most of the possible points for an answer that shows an understanding of the concepts but has some calculation errors, whereas you should give little or no credit for an answer that shows no understanding of the material. At the same time, you want the students to be able to see that their questions have been graded in an attentive, reasonable, and consistent manner.

Of course, it is not your responsibility to produce letter grades for an examination. That is the instructor's job. In what follows, we consider in some detail the two specific jobs that TA's do in grading: first, the grading of individual problems from an examination or course project and second, the handling of student questions concerning the grading of their work.

Grading a Problem

Most examinations in Statistics will be made up of several questions, with each question worth a certain specified number of points. As a grader, your job will be to grade individual problems. Specifically, suppose that you are assigned to grade some of the problems on, say, 250 Statistics 13 midterms. Here is a problem taken from a Statistics 13 examination.

1. (25 pts) In a study to investigate whether lack of sleep hurts short-term memory, 100 adults were each tested twice. Both times they studied a list of 30 words for two minutes, and were then asked to recall as many as possible, with a different but comparable list used each time. For one of the two tests they had been without sleep for 24 hours, while for the other they had a normal amount of sleep. (The order was randomly assigned.) The difference in number of words recalled was recorded (with sleep – without sleep) for each person. The average difference was 1.5 words, with a standard deviation of 6.1 words. Carry out the five steps of the hypothesis test for this situation, filling in the details in the space provided. Use $\alpha = 0.05$. [Space to write answers has been eliminated here.]

Step 1: Determine the null and alternative hypotheses. (Be sure you define the parameter you use.)

Step 2: Verify necessary data conditions, and if met, summarize the data into an appropriate test statistic.

Step 3: Assuming the null hypothesis is true, find the p -value.

Step 4: Decide whether or not the result is statistically significant.

Step 5: Report the conclusion in the context of the situation.

This problem is worth 25 points, but notice that the instructor has not designated how to divide the points among the 5 steps. Here are some suggestions that should enable you to grade such an examination question well.

1. Discuss the grading policy with the instructor. The instructor may have some preferences about what to look for when grading and may want you to mark the examination in a specific way.
2. Carefully study the problem yourself, producing your own detailed written solution. The instructor may also provide you with a written solution, but you should still write your own, preferably before you see the one provided by the instructor. When you go over both solutions, you will become more aware of the approaches to the problem and notation used in class and in solution sets. You will also ensure that you have a correct solution. Make sure that you confirm the solution with the instructor. For the problem above, here is a detailed solution that might result.

SOLUTION (The solution is in *italics*, the original statement of the problem is not.)

Step 1: Determine the null and alternative hypotheses. (Be sure you define the parameter you use.)

μ_d = mean difference in number of words people in the population would recall after normal sleep and lack of sleep, if everyone were to take the test.

$$H_0: \mu_d = 0$$

$$H_a: \mu_d > 0$$

Step 2: Verify necessary data conditions, and if met, summarize the data into an appropriate test statistic.

Data conditions met because $n = 100$ and is thus large.

$$\text{Test statistic is } t = \frac{\bar{d} - 0}{\frac{s_d}{\sqrt{n}}} = \frac{1.5 - 0}{\frac{6.1}{\sqrt{100}}} = 2.46$$

Step 3: Assuming the null hypothesis is true, find the p -value.

p -value = area above 2.46 for a t -distribution with $df = 100 - 1 = 99$.

Using Table A.3, notice that 2.46 falls between 2.33 and 2.58, so the p -value is between 0.011 and 0.006 (using either $df = 90$ or $df = 100$). This is usually presented as $0.006 < p\text{-value} < 0.011$.

Step 4: Decide whether or not the result is statistically significant.

The result is statistically significant because the p -value is less than 0.05, which was the stated level of significance to use.

Step 5: Report the conclusion in the context of the situation.

Reject the null hypothesis and accept the alternative hypothesis that the population mean difference in the number of words recalled before and after adequate sleep would be greater than 0. In other words, lack of sleep does seem to hurt short-term memory.

3. With the detailed solution to the problem in hand, you should now devise a key, by partitioning your solution into (ideally) natural parts, together with a reasonable assignment of points to each of these parts. You should read a few students' solutions without grading them, to get an idea of the general character of the students' solutions and the types of mistakes that they make. That will help you decide how many points to assign for each part of the solution, and how to treat common errors, creating a useful key. If possible, go over this key with the course instructor. Also, find out if what the instructor wants you to do about:

- Mistakes carried over from one part to the next. For instance, if a two-sided test is used by mistake, should points be taken off at both steps 1 and 3, or just once?
- Arithmetic mistakes, such as subtracting or dividing incorrectly.

Here is a key that might be produced for our example problem (*points in bold italics*).

Step 1: Determine the null and alternative hypotheses. (Be sure you define the parameter you use.)

5 points, with 3 for defining the parameter and 2 for the hypotheses.

Common mistakes and points off: *-1 for not mentioning population, -1 for \neq or $>$ in H_a*

μ_d = mean difference in number of words people in the population would recall after normal sleep and lack of sleep, if everyone were to take the test.

$H_0: \mu_d = 0$

$H_a: \mu_d > 0$

Step 2: Verify necessary data conditions, and if met, summarize the data into an appropriate test statistic.

8 points, with 2 for data conditions, 3 for test statistic formula, 3 for filling it in correctly.

Common mistakes and points off: *-4 for using two-sample t formula*

Data conditions met because $n = 100$ and is thus large.

$$\text{Test statistic is } t = \frac{\bar{d} - 0}{\frac{s_d}{\sqrt{n}}} = \frac{1.5 - 0}{\frac{6.1}{\sqrt{100}}} = 2.46$$

Step 3: Assuming the null hypothesis is true, find the p -value. **4 points total.**

Common mistakes and points off: *-2 for using the wrong df , -2 if not consistent with H_a in Step 1 (i.e. must use two-sided p -value if they stated a two-sided H_a)*

p -value = area above 2.46 for a t -distribution with $df = 100 - 1 = 99$.

Using Table A.3, notice that 2.46 falls between 2.33 and 2.58, so the p -value is between 0.011 and 0.006 (using either $df = 90$ or $df = 100$). This is usually presented as $0.006 < p\text{-value} < 0.011$.

Step 4: Decide whether or not the result is statistically significant. **4 points total**

The result is statistically significant because the p -value is less than 0.05, which was the stated level of significance to use.

Step 5: Report the conclusion in the context of the situation. **4 points total**

Common mistakes and points off: *-2 if "population" is not stated explicitly.*

Reject the null hypothesis and accept the alternative hypothesis that the population mean difference in the number of words recalled before and after adequate sleep would be greater than 0. In other words, lack of sleep does seem to hurt short-term memory.

4. With the key in hand, you are ready to grade a problem. To promote consistency in the grading of a problem, grade every student's solution to one problem before starting to grade another problem; in other words, grade problems sequentially and not in parallel. To ensure further consistency in your grading, you should keep a list of the most common (and most uncommon) errors along with number of points subtracted for each mistake. Alternatively, you may set aside some exams with errors so that you can remind yourself how many points you deduct for each type of error. Most people find that grading is not easy. You need to focus your attention on (say) 250 individual solutions to the problem, asking yourself to look at each solution in a fresh and attentive way. Doing this tries your patience and concentration, so it is a good idea to take some breaks.

5. It is very important that you are neat and clear with the marks that you make on student papers. (These marks might include comments, corrections, indicators of errors, points awarded, and so on.) Also have your own markings for no work, such as slash through the space on the paper, and a mark for a wrong answer, such as an X. That way, if a student comes to you claiming to have written a solution, but you had a slash through the page, you will know the student added the solution later. (In that case, take the exam and discuss the situation with the instructor, who may want to turn it over to Student Judicial Affairs; see the Appendices for how to do this.) A student needs to be able to read what you have written on his or her paper. And, fair or not, many students take scratchy or careless markings on their papers as an indication that the grading itself was done without care; you don't want to leave students with this false impression.

6. If the exam is designed such that the students enter their answers in the space provided, check the reverse sides in case a student showed additional work on the back of a page. If the students give their solutions in a blue book, check all pages before deciding that one response is missing; a student may have answered questions in a different order than the order they were given.

7. When the grading is done, the scores should be added and recorded. To protect privacy, *do not indicate the score at the top of the first page of an exam or on the outside of the cover of a blue book.* Descriptive statistics, along with a histogram of scores, should be prepared for the professor of the course. This step may take a long time, depending on the size of the class and method used for entering grades.

If you follow the approach just outlined, you should do a very good job grading examinations. One more important suggestion: Find out right at the start of the quarter what exam grading you will be doing that quarter. Then block out time on your calendar for this work right away. As already mentioned, good grading is time-consuming and energy-consuming. You need to plan to be able to set all of your other work aside when your grading is to be done.

Finally, if any questions arise or you feel uncomfortable about any part of the grading process, do not hesitate to consult the instructor for whom you are doing the grading.

Grading Projects

Grading a project will be different from grading an examination. Whereas you can often scan through an answer on an exam to check that it is correct, you will have to read a project thoroughly. Consult the instructor of the course to see how you should grade a project.

Handling Student Questions about Grading

After an examination has been returned, some students will have questions about the grading of their examination. In responding to these questions, aim to be open, helpful and informative. After leaving your office, the student ideally should be informed about their questions and believe that they have been properly addressed. Be sure to have an examination key. If possible, have a histogram of the exam scores and the table of letter grade equivalents for exam scores.

When a student has a question about the grading of a particular problem, the student often has two questions implicitly in mind. First, are there really mistakes here? Second, if there really are mistakes, where are they and were the proper number of points deducted? Start by handling the first question, which should be relatively easy. Go over the key's solution with the student, and then go over the student's solution. If there are mistakes in the student's work, you should be able to point them out. Often, the student will be satisfied at this point, because when he/she arrived at your office, he/she had little understanding of the problem and/or sincerely thought his/her exam solution was completely correct. If the question of correct awarding of points comes up, the student should be referred to the individual who graded the problem. Suppose you are the one who graded the problem. You should be able to go over the student's solution with the key in hand, and see if points have been awarded consistently with the key. If you find that you have made a mistake, you should, of course, correct it. Regrading while a student is watching you can easily lead to mistakes in judgment. A better procedure here is to agree to regrade the problem by the next day. This provides you with the opportunity to do the regrading alone, after some careful thought. If you have graded the problem consistently with the key and the student is still not happy with the grading, refer the student to the course instructor. At all times, try to be reasonable, helpful, and patient.

Indications of Academic Dishonesty

Some signs of academic dishonesty on an examination are: having the correct response to a question on the wrong version of the exam, having answers with no justification, and scribbling answers to a question on a different page from the page where the question is given. If you come across any of these or other signs of dishonesty, report them to the instructor of the course, who may want to make a copy of the exam, keep the original, and return the copy to the student. Also, if you suspect that a student changed his or her answer before turning an exam in for regrading, notify the instructor and make a note to photocopy that student's future exams before returning them to the student.

6. Monitoring Examinations

As a TA you often will administer and monitor an examination. Large classes, for example, are often split into two groups for examinations, with the groups taking the exam in separate rooms. You may be responsible for administering the exam to one group, while the instructor does the same for the other group.

This job is really relatively easy, provided you make sure of a few things. First, make sure that you know how the examination is to be conducted! Go over the examination instructions with the instructor to make sure that you understand them. Find out from the instructor about any examination conventions not mentioned in the instructions. Are calculators allowed? Are class notes allowed? Is a page of summary notes allowed? Second, make sure the students understand the examination conventions. Get to your exam room early and distribute the exams to the students,

so that the distribution is done close to the time that the exam is scheduled to start. Have the students sit in alternate seats, if there are enough seats in the room. Then take a minute or two and go over the examination with the students. In particular, go over the exam instructions, and any exam conventions that are supposed to be followed, indicating finally a stopping time for the exam. ("It is 8:03 a.m. now, and we will work for 50 minutes, until 8:53."). Then start the exam. Third, during the examination it's your job to maintain conditions that are good for exam taking. Monitor the students. Maintain quiet. Respond to students' questions for clarification on exam questions. Here it is essential that you have gone over the exam earlier with the instructor, so that you understand the exam questions very well yourself. Be as unobtrusive as possible as you do these things. Make sure you have a watch so you can end the exam at the proper time.

Finally, when the time limit is reached, collect the students' examinations. Know whether or not students are allowed to keep exam copies. You should have a careful count of the number of students in the room, and of the number of exams turned in. For the sake of security, you should have a box or large envelope in which to put the exams. As soon as possible, deliver these exams to the instructor (or other TAs if instructed to do that instead).

Suspected in-progress cheating requires action on your part. The ideas in the following paragraphs are taken from Student Judicial Affairs documents and the TA Handbook. (See the Appendices in this Handbook as well.) According to our Code of Academic Conduct, you are not expected to police students, but you do share responsibility for encouraging academic honesty. The less responsibility you take during the administration of an exam, the greater the pressure on the students. In general, it is advisable to circulate in the classroom while examinations are being given and to stop any blatant in-progress cheating. However, it is important not to disturb the atmosphere, which must be conducive to taking an exam.

In the case of students using inadmissible materials during an exam, it is strongly recommended that you simply confiscate them. If, when you grade the exam, you suspect that impermissible materials were used and you report the academic misconduct, the materials will be necessary in order to prove the charge, should the student deny cheating.

While actively monitoring examinations, care should always be taken not to disturb the atmosphere in which the examination is being administered. After reminding the students that they are under the Code of Academic Conduct, you can assume they are following it and need intervene only if there is evidence to the contrary. If you observe the students doing anything that has been prohibited, you can ask them to stop doing it. You cannot, however, stop a student from completing an exam, simply because you believe some academic misconduct is taking place. If, after an exam has been administered, you have reason to believe that a student has cheated, you should follow the established procedures for reporting academic misconduct.

Perhaps the best way to handle the problem of cheating is to set up examination conditions that make cheating very difficult.

Note: A copy of the Code of Academic Conduct and a copy of Rights and responsibilities of Instructors when confronting Student Misconduct in Class--Confronting In-Progress cheating have

been included in the appendix for your information. These documents are provided by Student Judicial Affairs. Please study these documents carefully before the first examination you monitor. It is also a good idea to review the guidelines before each exam. For more information, refer to the "Policies and Procedures" section of the TA handbook.

7. Office Hours

A teaching assistant will almost always have to hold office hours. Many undergraduate students need help with their statistics courses, and these students correctly regard teaching assistant office hours as a primary way of getting this help. As a teaching assistant, realize that students depend on you for many kinds of help during office hours and be prepared to provide this help. It is very important to keep your office hours faithfully, and to be on time. (No one likes to be stood up or to be kept waiting). Try to arrive with the intention of wholeheartedly and generously spending your office hour on your students. When your office hour ends, however, remember that you have your own work to do. At a reasonable stopping point, dismiss the students and see them out of your office. Don't leave students to study in a TA office; TA offices are not undergraduate study halls. It may be wise to study and work in locations other than your office, in order to avoid interruptions and unwarranted requests for your time.

Ideally, a teaching assistant should arrive at an office hour knowing course structure, general information about current and future course work, and having a good operational understanding of currently assigned readings and homework. In particular, all the administrative details that go with a course (homework assignments, due dates, exam and exam review session dates, grading policies, and so on) should be familiar to the TA. Often students arrive at office hours looking only for this kind of information, and providing it immediately is very helpful to them. Keep a record of all the assignments made in the course, and a copy of all handouts, for reference. Take the initiative and keep in close touch with instructors so as to have completely up-to-date information about the courses for which you are the TA.

Another major function of a TA during office hours is to help students understand class notes, readings and current and prior homework assignments. Be sure to find out from the instructor what the policy is on helping students with current homework; in some classes, the instructor may not want the students to get too much help on the homework. Prepare by reviewing the assignments and readings beforehand and if possible by actually doing the problems. You should expect to be stumped by a question now and then. When this happens, admit this directly and assure the student that you will find out the answer to the question and have it for the student by a specified future time.

You may find that explaining problems to students is not easy to do well. You need to listen carefully, to learn precisely what about a problem is not understood by the student. You need then to be able to think of examples or observations which may clarify the point of misunderstanding. And then you need to be able to communicate these examples and observations to the student.

Expect your office hours to be very busy just before examinations. At these times, students will often ask general questions about what material will be covered and how to study for exams. You may offer your best advice here, based on your knowledge of the instructor and of the course itself. You can prepare for this question by asking the instructor yourself to specify what students should study to prepare for the exam. But include a disclaimer with your advice, and encourage students to

see the course instructor for answers to these kinds of questions.

In interacting with students, you want to be friendly, forthright, and encouraging. But keep in mind that your job is to help students in their efforts to study and to master course material. Do not do their work for them. Also avoid giving a student inside information that gives an unfair advantage to this student over other students in the class.

Sometimes a student approaches a TA for help that the TA should not give. For example, a student may ask for special information about an upcoming midterm. At the minimum, such a request should be politely but firmly turned down. Or a student may approach you with legitimate questions about course material to start with, and then move to seeking your advice about and help with academic stress problems or other personal problems. If the situation permits it, be sympathetic with such a student and listen, but be absolutely sure to refer the student to an appropriate counseling center whose personnel are prepared to deal with these kinds of problems.

8. Solution Sets and Handouts

Homework solutions are provided in many statistics classes and are often written by the TA. Many statistics classes also require the students to do homework via computer, and the TA's usually are responsible for producing the needed computer handouts. Useful guidelines for writing each kind of document appear in this section. In both cases, the objective is of course to write a document that is useful to the students.

Many textbooks have a "solutions manual" that gives solutions to all of the problems. The manuals are provided to the instructor. Ask your instructor if there is a solutions manual for the textbook used in the class. If the instructor doesn't know, then the office staff may be able to find out for you by contacting the publisher. Some instructors will allow you to "cut and paste" from the solutions manual when preparing solutions for students.

Homework Solutions

From the student's perspective, homework solutions should fill in the gaps that hinder them from solving problems. So the TA should attempt to provide solutions that have the following qualities:

1. The solutions are organized, clear, and concise. These qualities enable the student to scan the solutions quickly for the information they need.
2. The solutions are complete. The TA should not skip major steps in a solution.
3. The solutions are neat.
4. The notation of the solutions is consistent with the book and the instructor's lectures.
5. When appropriate and when time permits, the TA could additionally describe alternative solutions or provide insight to conceptual questions associated with a problem.

Taking the time to follow all the guidelines above will result in excellent solutions, but in all likelihood will leave the TA no time for his/her own studies. Be realistic about producing these documents. Nevertheless, aim to follow these guidelines.

It is helpful for a TA to view writing solution sets as an opportunity to expand and organize his/her

knowledge of the subject. Relearning of ideas not only gives one personal satisfaction, but has practical advantages as well. It prepares the TA for discussion sections and office hours; for certain subjects, it prepares the TA for the required first year examination.

A few examples may clarify how one implements the suggestions above:

Problem (adapted from Exercise 8.70 of *Mind on Statistics, 3rd edition* by Utts and Heckard):

Suppose the length of time a person takes to use an ATM machine is normally distributed with mean $\mu = 100$ seconds and standard deviation $\sigma = 10$ seconds. There are $n = 4$ people ahead of Jackson in a line of people waiting to use the machine. He is concerned about $T =$ total time the four people will take to use the machine, and hopes it will be less than or equal to 360 seconds (6 minutes). Assuming the times taken by the individuals are all independent, find $P(T \leq 360)$, the probability that the total waiting time is less than 360 seconds.

Solution:

First, the mean and standard deviation of T must be found. The mean for the total of four individuals is the sum of the means for individuals. This is $100 + 100 + 100 + 100 = 400$ seconds. The variance for an individual is $(10)^2 = 100$, so the variance for the total of four independent individuals $= 100 + 100 + 100 + 100 = 400$. (The problem states that the times are independent, so the variances can be added.) The standard deviation for the total is $\sqrt{400} = 20$ seconds. For $T = 360$, $z = \frac{360 - 400}{20} = -2$. Thus, $P(T \leq 360) = P(Z \leq -2) = .0228$.

Problem (part of Exercise 4.9 of *Mind on Statistics, 3rd edition* by Utts and Heckard):

For each of the following research scenarios, explain whether a randomized experiment could be used.

- a. To study the relationship between long-term practice of meditation and blood pressure.
- b. To determine whether a special training program improves scores on a standard college admissions test.

Solution:

a. A randomized experiment is probably not possible here. Although it would be possible to randomly assign some people to long-term meditation, it would be hard to be sure that participants complied and it would take a long time to wait for them to be “long-term” meditators. Also, some assigned to the “no meditation” group may really want to meditate and would do so anyway. It would be much more practical to do an observational study in this situation.

b. Yes, a randomized experiment could be done. People could be randomized to two groups - one that takes the special training program and one that does not. Afterward, the standardized test scores of the two groups could be compared.

Note: In general, if a group of volunteers can be recruited who are willing to be assigned to either treatment, and if doing so would be ethical, then a randomized experiment can be done.

These solutions are clear and concise. They explain the reasoning, and in the case of the first problem, the solution is explained rather than simply written with formulas. The notation in the

solution is consistent with the notation in the problem. The note in the second solution exemplifies guideline five; it enhances the solution but is not essential to good solution writing.

The TA may find suitable solutions in solution manuals, but he/she should at least attempt to solve the problems first before looking at these manuals. Solving a problem promotes thinking about it, and thinking about it prepares the TA for discussion sections and office hours. It alerts the TA to alternate solutions to the problem, which the TA could encounter from a student. Also, it is not uncommon for solution manuals to contain mistakes.

Usually, the instructor expects the solutions to be written following particular guidelines. The TA should communicate with the instructor regularly, so that he or she will write useful solutions the first time. Also, ask the instructor when the solutions will be needed, and *make sure you have them ready by that time*. Probably the biggest single complain instructors have had about TA performance in the past is that the TA did not prepare solutions and make them available to students by the time the instructor told the class that they would be ready. But, make sure you don't provide the solutions to students before the date given by the instructor; that might mean students have them before the homework is to be turned in!

The instructor may have a preferred method of making solution sets and handouts available to the students. You may be asked to take copies of the solution sets and handouts to the copy shop, fax them into the copy shop, or make them available through e-mail or the web.

Handouts

In many statistics classes the students are required to learn to use a statistical package. If this is done on a networked computer, many students will have to learn their way around the computer and network, in addition to learning the statistical package. Most instructors give the TA the responsibility of writing computer handouts that explain both topics. Although many students come in with experience working on a computer, the TA should realize that at least some students have not used computers very much before. When you are constructing a computer handout, then, do not assume that the students have prerequisite computer knowledge of essential commands. Also, test commands and examples on the computer, making sure that you obtain the desired results, before issuing instructions on a handout.

Ideally handouts concerning the campus computers should cover:

- how to log on and log off
- how to get help (from the computer as well as consultants)
- what is a file and what is its syntax
- how to create, edit, delete, and purge files
- how to print files and how to operate the printer
- how to abort a command's execution
- how to interrupt and restart a command's execution

Ideally handouts concerning a statistical package should cover:

- how to enter and exit the package
- how to save output from the package

- how to save input (the data set being analyzed)
- how to use the statistical commands relevant to the course

After looking at all these items, the reader might envision a lengthy handout, produced through many, many hours of toil by the TA. In fact, covering most of these items simply involves looking up a particular command. Often you will work from an already-written handout, obtained from an instructor or another TA.

Again it should be emphasized that the TA should write statistics solutions and computer handouts so that the product is an effective source of information for the student. Look upon these writing tasks as challenges to express the subject matter as clearly as possible and as opportunities to improve your own understanding of the material covered.

You can make copies of your handouts through the department. If you have a small amount of copies to be made, you can use the copy machine in the Statistics department office. Your four-digit code is your office number. Don't forget to logout when you are finished.

If you have many copies to make, for example, 250 tests, it is simpler for you and less expensive for the department if you use of the campus copy service. You can give the original to the front desk in the department office at least two days before you need the copies and they will help you fill out the necessary form and send it off for you. When the copies come in, you will receive an email as to where they can be picked up in the front office.

9. Teaching Your Own Class

Sometime in your graduate career, you may have the opportunity to be an instructor (AI) for an undergraduate Statistics class. This means that you are the lead instructor of a class and would most likely have TAs working for you. It is your responsibility to choose a textbook, create and give lectures, create and give homework and tests, and deal with students on a more administrative level. Below are some administrative details that should be known to any AI.

Adding/Dropping

When you are an AI, it is your responsibility to deal with students who want to add or drop your class. If your class is full and there is no room for the student, students can put themselves on a waiting list and will be automatically enrolled if someone drops. If the class is full, but there are still seats available in the classroom and you allow the student to add, they can get a PTA (Permission to Add) number from Min in the department office and can use this number to enroll in the class online.

It is important to note that the last day to drop for certain designated classes, including STA 13, is on the 10th day of instruction, while the last day to add is on the 12th day of instruction.

Some students may want to add your class who are enrolled through University Extension. If this is the case, they will need your signature in order for them to officially add and only if there is room in the class.

Incompletes

You may get a request from a student for an incomplete, rather than a grade. This is only allowed if the student's grade is of passing quality at the point the incomplete is asked for or for good reason, e.g. death in the family, illness, etc. You would need to fill out a Request for Incomplete Grade form which stays in the department and explains what the student needs to do to complete the class. An incomplete must be completed by the end of the third succeeding quarter of the student's academic residence or it automatically becomes an F..

Grade Changes

It is highly recommended to make grades available to students about 24 hours before you actually submit them so you can make any changes needed before the final submission. If you do need to change a grade after they have been submitted, forms are available through Min. The completed form is then assessed by a campus committee which determines if there is sufficient evidence and good reason as to why the grade should be changed. It is common that grade change requests are denied. Thus, it best to work things out with your students before final grades are submitted.

Part II: Technology

1. Creating a Class Webpage

Using a Current Layout

The easiest way to make a webpage is to “borrow” the layout from a current one. To do this, find a website that you would like to use. Save the page as a webpage (e.g. file.html) file. You can now open this file in Microsoft Word or FrontPage and change things appropriately to fit your needs.

You can start a webpage from scratch in Microsoft Word or FrontPage. Everything is similar to typing a regular document except that in Word, you want to save the file as a webpage (e.g. file.html).

You will most likely want links to other pages or files (e.g. .doc or .html). To create a hyperlink in Word or FrontPage, highlight the word(s) you want the link to be, right click and chose ‘Hyperlink.’ From here, you need to type in the URL address of where the link is supposed to end up. How to find this URL address will be discussed later. To later edit hyperlinks, right-click on the hyperlink and choose Hyperlink Properties.

Using HTML

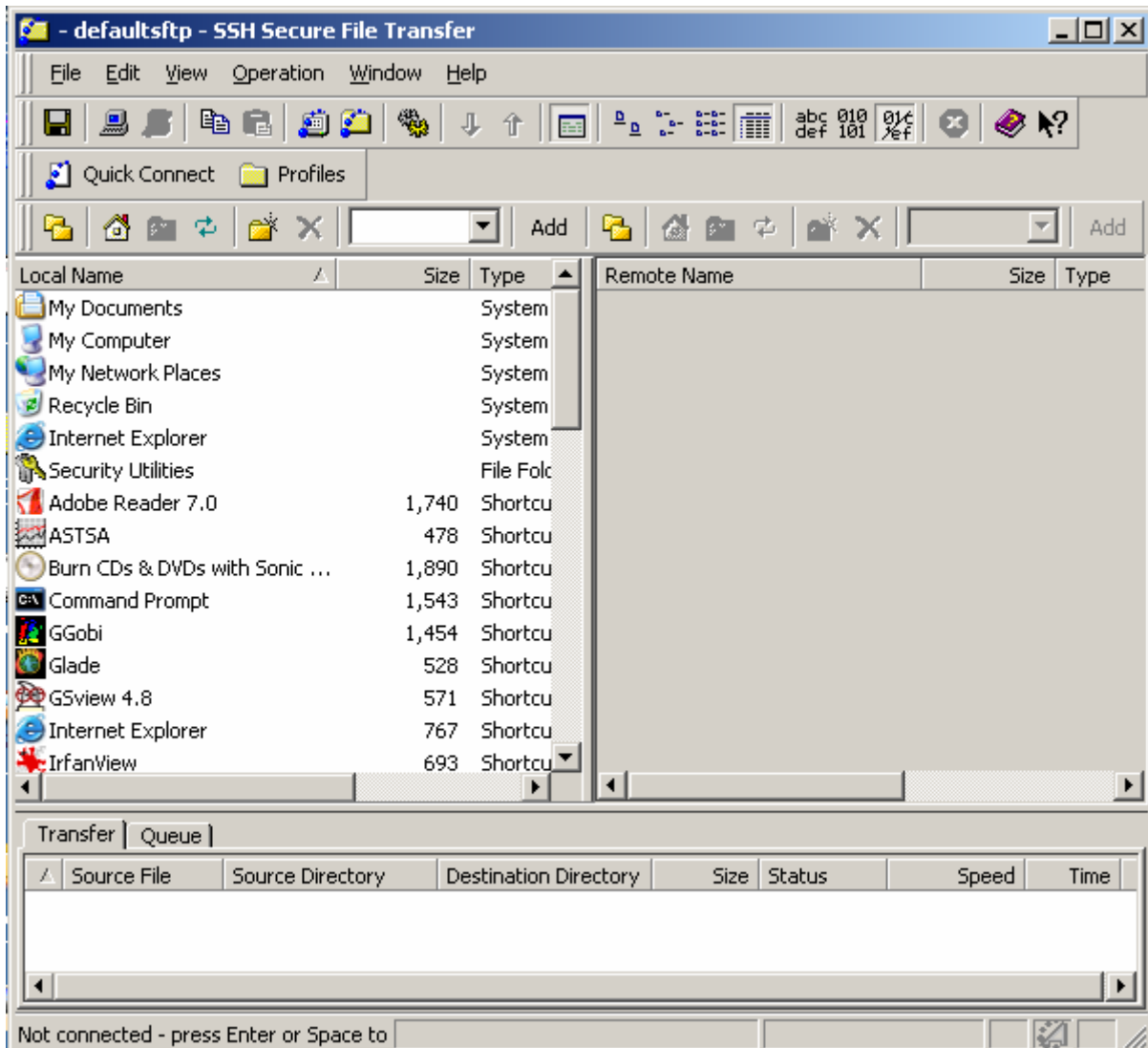
Another way to create a webpage is to write HTML code. Once again, you can ‘borrow’ the html code (aka source code) from a current website as a starting point for your own. You can do this by right-clicking on a page you like and choose ‘View Page Source.’ This will open up an HTML code file that you can save or copy and use in Notepad, Word or FrontPage. If you use Notepad or Word, you will just work with the code itself. FrontPage actually allows you to view both the actual layout and the source code at the same time. Sometimes it is easier to change things in the code than in the document itself, especially when a change to a link needs to be made. Also, there are many resources online that will help you with HTML code. One useful site is <http://w3schools.com/html>.

Posting Your Webpage

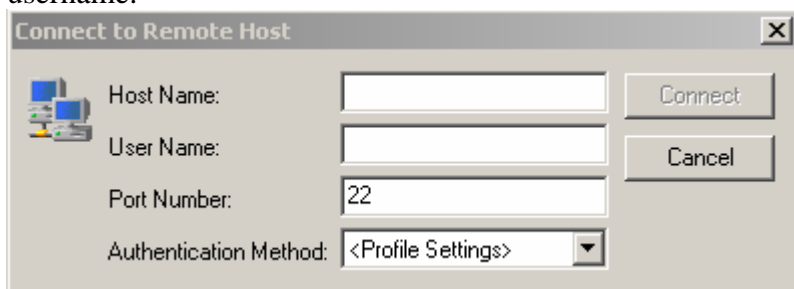
Once you have finished your webpage, you will want to post it. The easiest way to do this is to transfer your file onto the Statistics webpage server, anson. This is done by logging on to anson.ucdavis.edu using a file transfer program, either on a campus computer or on your home computer. If you are using a computer in one of the Statistics labs, you can open SSH Secure



File Transfer Client which can be found on the desktop with When you open the file transfer program (e.g. SFTP (you can get a free FTP from ucdavis.edu)), the screen should be similar to

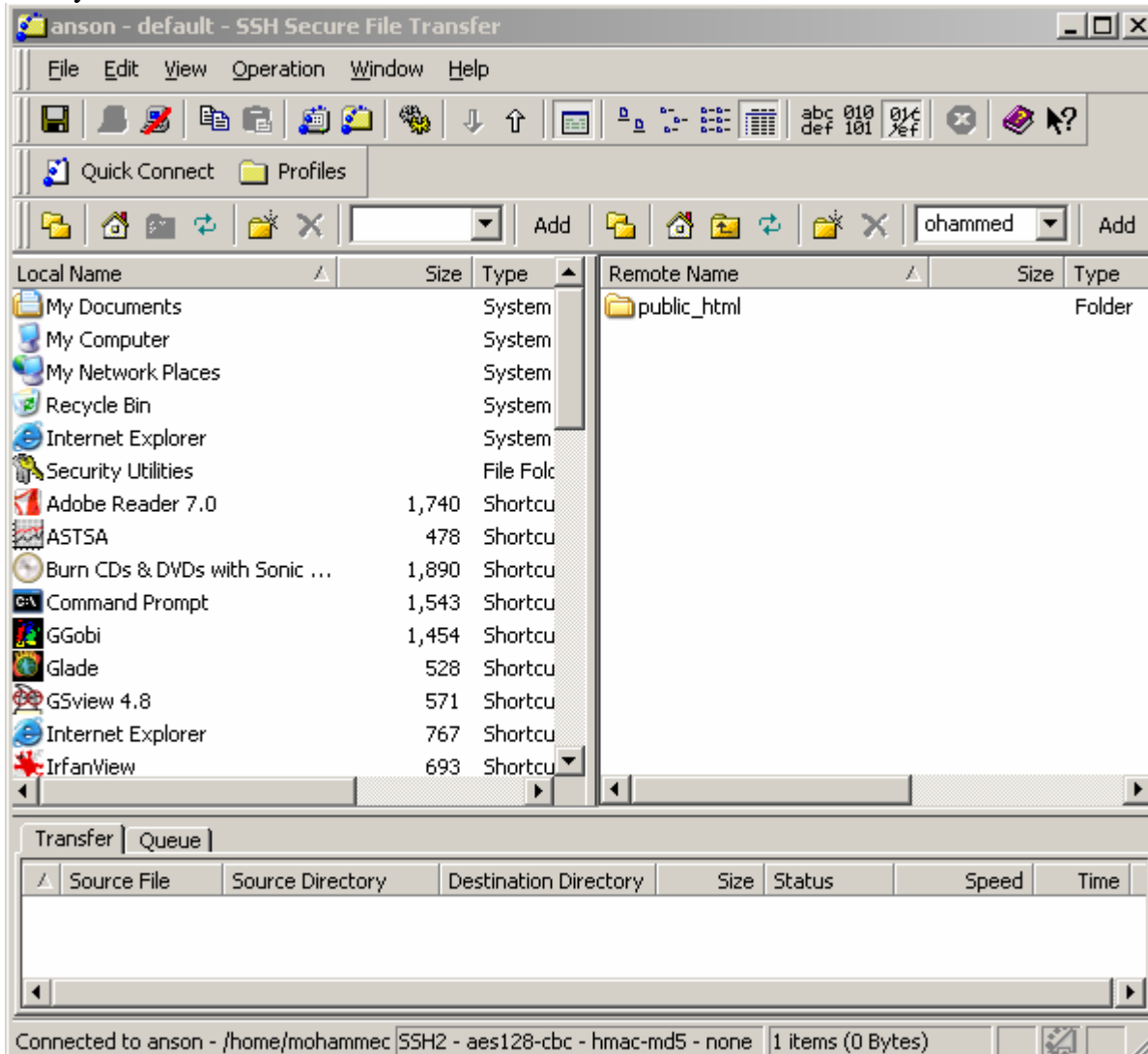


Click on 'Quick Connect' on the top left and you will come to a screen that asks for the host and username.



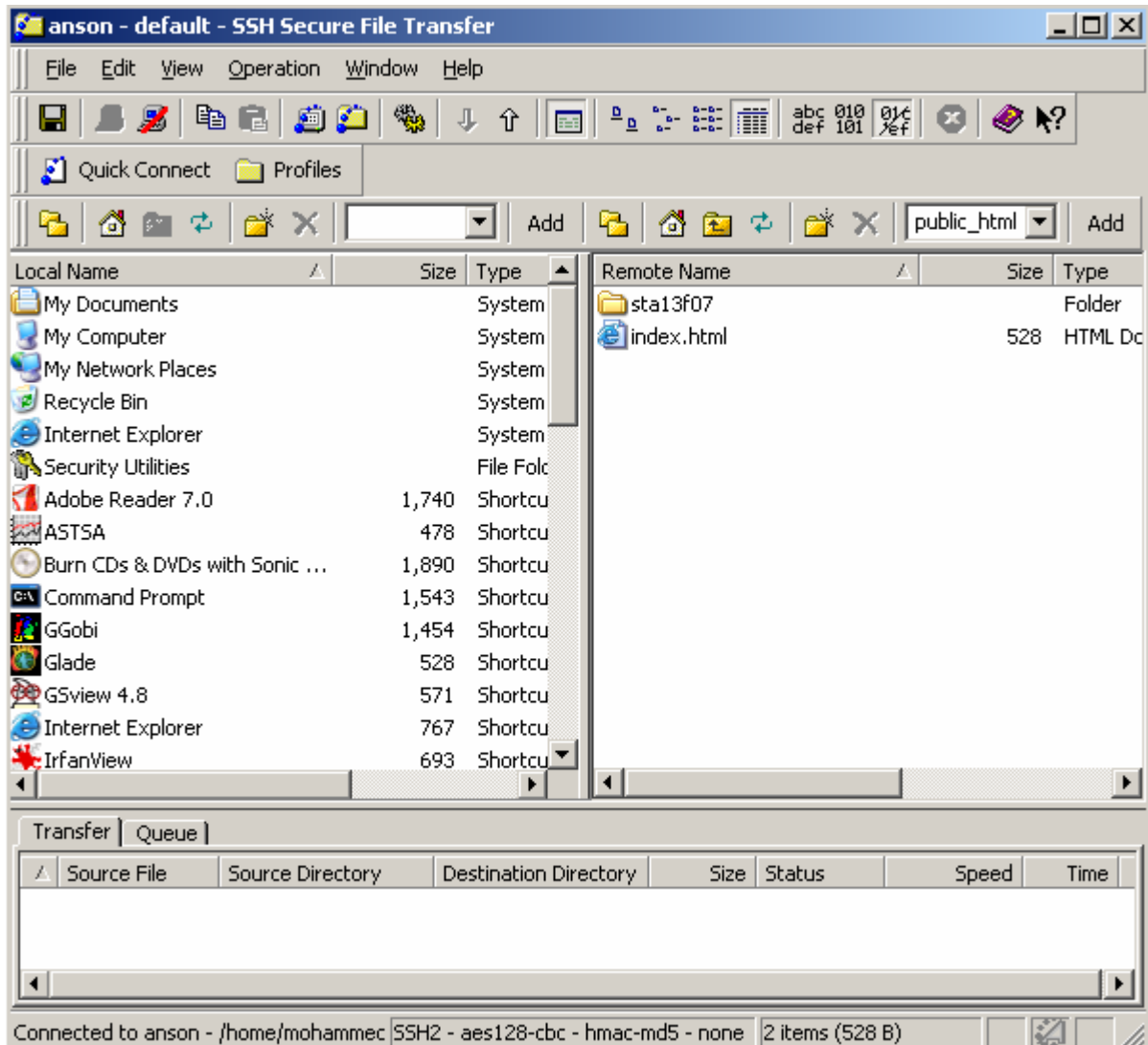
The host name is anson (Though if you are trying to login through your home computer, the host name is anson.ucdavis.edu). Your username is the same one used for your department email. (e.g. If my email address was smith@wald.ucdavis.edu , my username is smith.). After you click Connect, another box should pop up asking for you password. This is the password you use to check your department email.

Now you should have a screen with two sides as shown below.



The left shows all the files on the computer you are currently on while the right side shows the files under your name on the anson server. There should be a folder on the right named public_html. Open this folder. Notice that there is a file labeled index.html. This is your own homepage to use if you like, with the URL <http://www.stat.ucdavis.edu/~username>. To do this, save your created webpage as index.html. You can then find this file on the left side of the screen and transfer it to the right, overwriting the original index.html.

If you are making a webpage for a class you TA for, you need to create a new folder on the anson server (the right hand side of the screen) for each specific class, usually named with the class number, quarter and year (e.g. sta13f07).



After you create a new folder, open it. Within this folder, you can now drag and drop the files corresponding to the webpage from the left side into this folder on the right. Make sure to name the main page 'index.html.' Add any other files or pages that are included in this webpage, e.g. other linked pages you created and Word/PDF files. Make sure that the hyperlinks match with the names of the files you are linking to. Each file in this folder has a URL address. You can find the URL for a specific file by opening the file from within the anson folder. Use this URL when creating your hyperlinks on your webpage. After this, your webpage will be up and ready to use. Try typing in the URL to the main page just to make sure. The URL should be something like <http://www.stat.ucdavis.edu/~username/sta13f07> and make sure all your links are working. Also, one thing that is often overlooked is that webpages look different depending on what browser is being used. It is highly recommended to at least open your page using Internet Explorer and Mozilla Firefox to make sure there aren't formatting problems in either.

After you have made sure all things are the way they are supposed to be, you need to email the webmaster the URL and which class you are a TA for so that the link can be put up on the

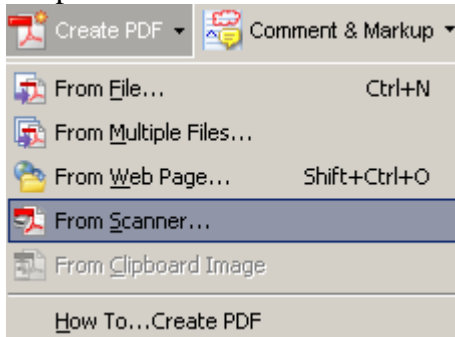
Statistics department course page, <http://www.stat.ucdavis.edu/courses>.

Be careful what you put on your web page. Most graphics and photographs on the web are copyrighted and cannot be put on your page without permission from the owners. Pornographic or other patently offensive material is not allowed on your department web page. If you post inappropriate material you will be directed to remove it and you may lose your account privileges. You should also consider your family's safety when deciding what material to post. Some safety experts recommend that you not post children's pictures or names, your home phone number, or your home address. Remember that anybody in the world can potentially access your web page.

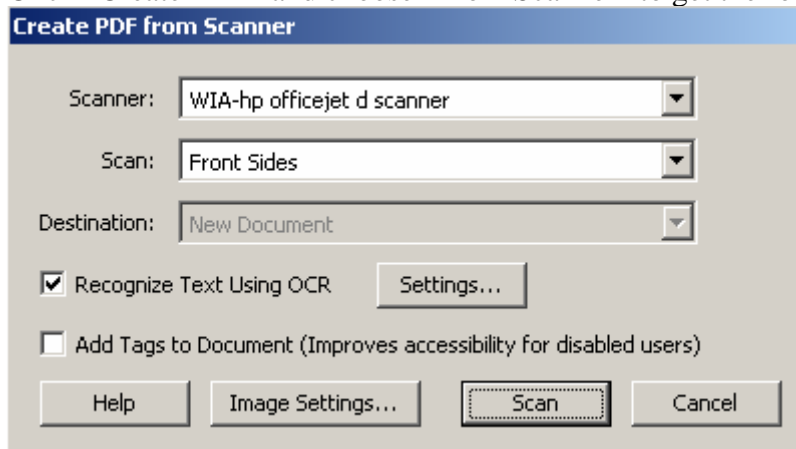
2. Scanning Documents

You may be asked to scan certain documents to put on the class website. You will most likely want to scan a document into a PDF. There are a few scanners available for use in the Statistics department. There is one in the department office and there are two in the TA office. Each scanner is slightly different, but the general concept explained below is the same.

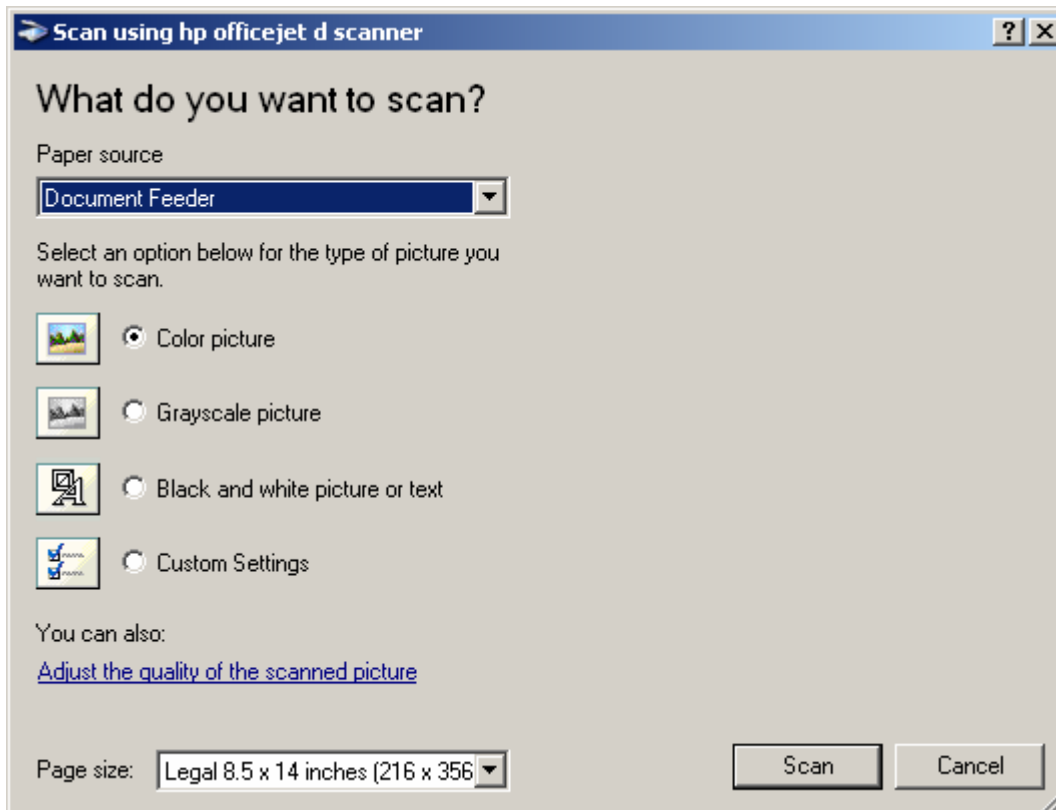
First, make sure the scanner is on and place your first page on the scanner, appropriately aligned. Note that the correct alignment of the page varies from scanner to scanner. You will now need to open Adobe Acrobat Professional (not Adobe Acrobat Reader).



Click "Create PDF" and choose "From Scanner" to get the following pop-up box:



Make sure the appropriate scanner is chosen, and then click "Scan" to get the next pop-up box.



Choose your paper source, if asked for, and what you are scanning. Black and white documents tend to turn out clearer if “Color Picture” is chosen instead of “Black and White.” Now, click “Scan” again and the scanner will scan your page. If you are scanning more than one page, continue the last couple of steps until you are finished. When you have finished scanning for your current document, click “Cancel” and/or “Done.” (This step varies from scanner to scanner.) Now your scanned pages will show up on the screen and you can save this file for later use.

3. Introduction to Computing Packages

A TA already familiar with computers and statistical packages should have little trouble. For a TA without this knowledge, he or she may have to spend extra time learning specifics. Because most companies will only employ statisticians that have experience with computers and statistical packages, the time spent learning computing will be helpful. Graduate students in Statistics should aim to be proficient at statistical computing and a variety of statistical software before they leave graduate school.

This part contains introductory sections for some of the statistical computing packages commonly used in classes offered by the Department of Statistics. MINITAB is most commonly used in undergraduate classes. SAS, which may be used in Statistics 106 or 108, is usually used in more advanced courses such as Statistics 138. R may be used in some undergraduate courses such as Statistics 32, 106, 108 and 135.

When you are conducting lab sessions for the students, keep in mind that many of the people there may not be familiar with the particular operating system used. While you may be familiar with the operating system and know some of the shortcuts, remember that someone else may be seeing the operating system for the first time. Thus, it is a good idea to go slowly and make sure that each person is following what you are doing, so that you do not leave students behind.

The sections here do not give a full guide to using the software packages described, but they are a starting point for learning each package. Subsequent handouts will depend on the class for which the software package is being used. Make sure that you and the students know where to go for help.

MINITAB

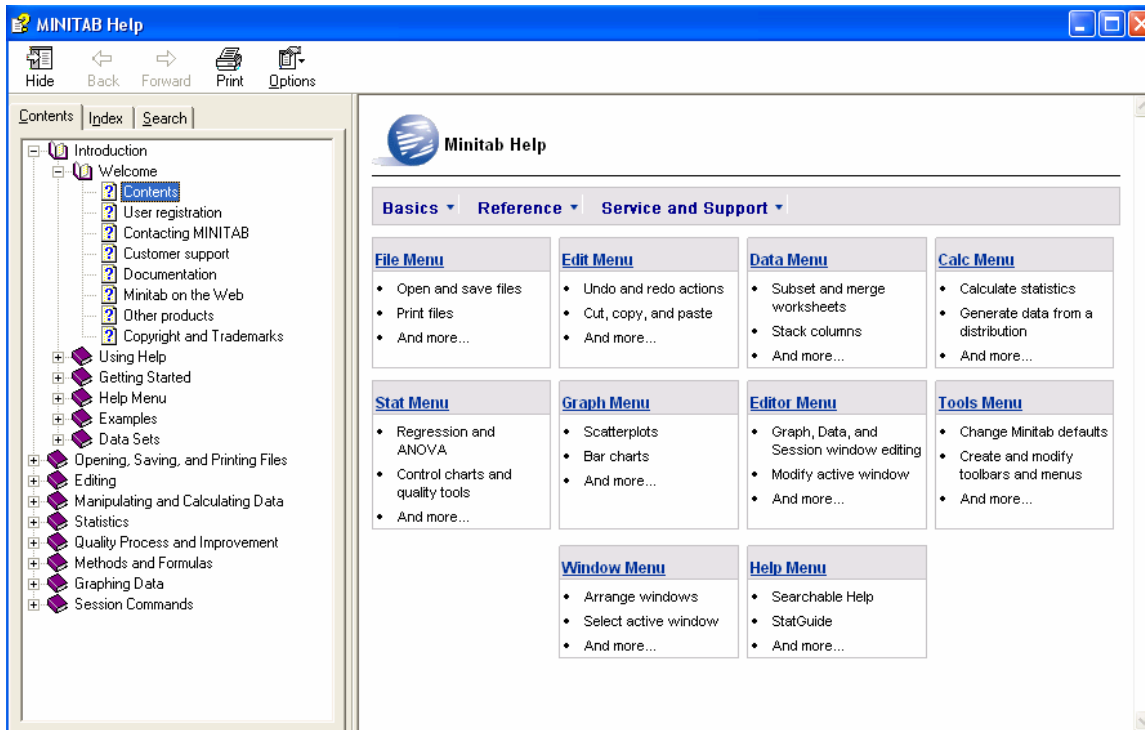
What is MINITAB?

MINITAB is a statistical computing package available on the Macintosh, Windows, DOS, and VMS. It is popular because it is easy to learn and simple to use for basic statistical analysis.

MINITAB is currently installed on window machines in the Statistics computing lab and TA offices of the Statistics Department. It's also available on most labs on campus. For the windows version, to get started with MINITAB, you can either click on the MINITAB short cut icon or open it from the start menu -> All Program -> MINITAB. The current updated version is MINITAB 14, though MINTAB 12 may be on the computers in the Statistics lab. There are a few differences between versions 12 and 14. Some of these changes can be found at <http://www.gtpcc.org/gtpcc/minitab14.htm>. MINITAB can also be used interactively using command lines; menu commands are optional on the Macintosh, Windows, and DOS versions. However, the easy way for beginners to learn MINITAB is the point-and-click way. This is the way we will discuss.

Getting Help

To get an overview on MINITAB, you can click on the help menu bar as the following picture shows:

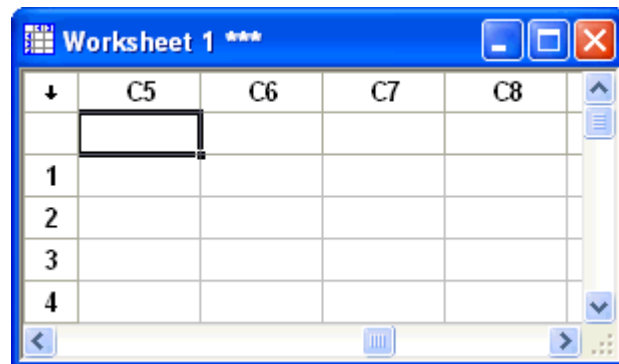


It will give you a list of several topics. Follow the instructions there for more information.

Entering Data and Saving Worksheets and Projects

With Version 14, MINITAB can allow you to enter up to 10000 cells; whereas, it's only 5000 cells in Version 13 or older ones. Following is the picture demonstrating where you can enter your data. Then save your worksheet as a .MTW file. By default, a MINITAB file has such an extension. NOTE: The extension MTW is for MINITAB worksheets, and the extension MPJ is for "MINITAB Project." If you click on the file menu while in an active session, it gives you the option to save the current worksheet, or to save the project. If you save the project, the entire session, including the worksheet and all output generated so far are all saved. You can then open the project later and continue where you left off. You can also save individual graphs by choosing File -> "Save graph as" and use them later in a report, such as homework solutions for a class webpage.

Data window works like a spreadsheet
Minitab starts with the spreadsheet format and features you already know how to use, then adds the statistical power other spreadsheet applications just can't offer.



If you already have a data set (perhaps saved in a different extension, e.g. .dat, .txt, .xls), you can import your data by using MINITAB wizard by doing the following:

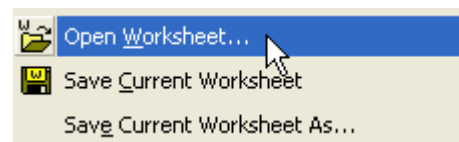
File -> open worksheet

Easily

Powerful import and export features make it easy to transfer data between Minitab and almost any type of file.

Import

Data



Most tool bars (File, Edit, Data, Calc, Stat, Graph, Editor) in MINITAB are quite useful in manipulating and managing data for an analysis.

Minitab includes the full range of basic statistical procedures used for simple estimation and hypothesis testing with one or two samples. These often form the foundation for more advanced investigation. Among them:

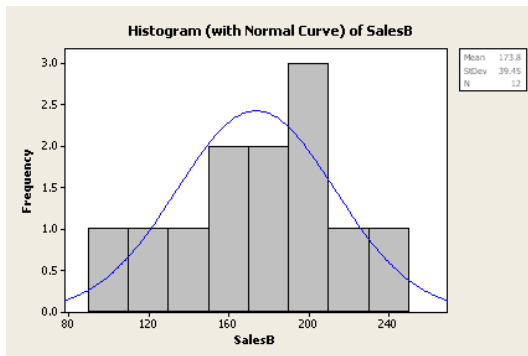
- Calculating or storing descriptive statistics

- Hypothesis tests and confidence intervals of the mean or difference in means
- Hypothesis tests and confidence intervals for a proportion or the difference in proportions
- Hypothesis test for equality of variance
- Measuring association
- Testing for normality of a distribution

The followings are some examples. To get the histogram use Stat tool

Stat -> Basic Statistics -> Display Descriptive Statistics.

At the current window dialog box, you can click Graph option and mark histogram plot. The following is the what the output will look like.



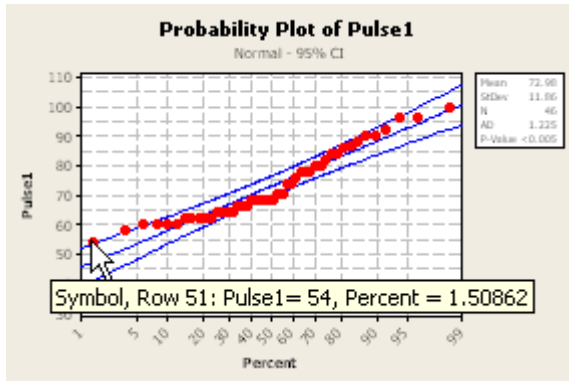
You can use

Stat -> Basic Statistics -> other options (e.g. 1-sample, 2-sample and paired t- tests, 1 proportion, 2 proportions tests, equal variance tests, correlation, and Normality tests.)

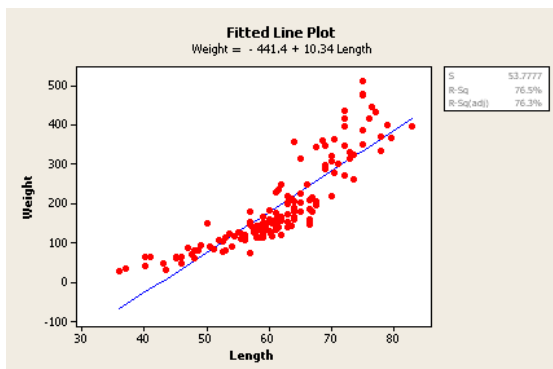
If you want to **perform a regression analysis**, you can use Stat tool

Stat -> Regression -> Regression.

At the current window dialog box, you will see two sub windows. In the Response sub window, enter the column containing the response (Y) variable. (A list of eligible variables will appear on the left. If you position the mouse in a sub window on the right, then go over to the list of variables and double click the one you want, it will be entered in the sub window.) In the Predictors sub window, enter the columns containing the predictor (X) variables. If you like, use one or more of the dialog box options (e.g. Graphs, Results, Options, Storage) to store results for further residual analysis—modeling diagnostic. Then click OK button. The output will be the regression equation (including parameter estimates and T-statistics, p-values), R-square value and ANOVA table. The following picture is obtained from the diagnostic.



Stat -> Regression -> Fitted line Plot. At the current window dialog box, enter the response and predictor(s) to the sub dialog boxes to get the following picture.



Also, use Stat -> regression -> other options (e.g. Stepwise, Best subsets, Binary Logistic Regression.)

To perform an Analysis of Variance (ANOVA)

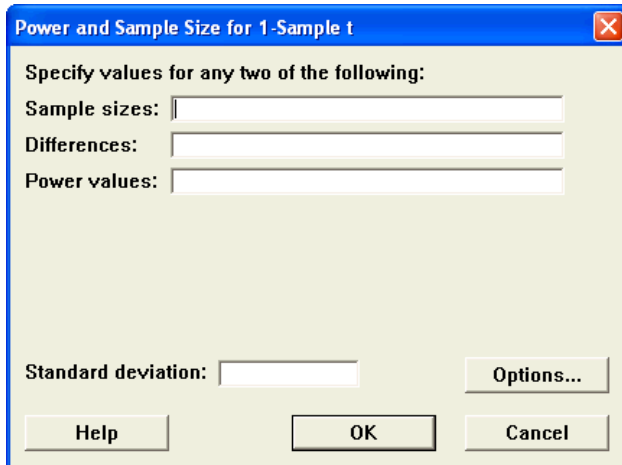
Use Stat -> ANOVA -> options (One-way, Two-way).

At the current window dialog box, enter Response (Y) and Factor variables in the sub dialog boxes. You can also use the other options in the current dialog box (e.g. confidence level, store residuals, store fits) or Graphs, Comparison to do your own specific analysis. The output will be ANOVA table mainly.

Stat -> ANOVA can also lead you to other options such as perform Interval Plot, Main Effect plot, Interaction plot.

Besides such above common analyses, Stat tool can also help you to perform other analyses such as Chi-square test, Time-series analysis, Non-parametric tests (Wilcoxon, Mann-Whitney, Kruskal-Wallis, etc.) and calculate Power and Sample size as well.

Stat -> Power and Sample Size -> 1-sample t will give you the following picture for this calculation.



Calc tool is the second most useful tool (after Stat tool) in MINITAB. With this tool, you get a calculator that includes (in the sub window dialog box) several practical functions (e.g. Natural log, mean, median, Normal score, etc.)

Use Calc -> Probability Distributions -> options (e.g. Normal, t, F, Uniform, poisson, binomial, Weibull, etc.) to help you to calculate cumulative probability at a particular value and vice versa.

You can also use Calc tool to perform Column/Row statistics, Standardization, Matrices calculation (e.g. Transpose, Inverse, Eigen values, etc.)

Graph tool is the third most useful tool in MINITAB. With this tool, you can perform scatter plot, Matrix plot, Probability plot, Stem-and-leaf, dot plot, box plot, pie chart, bar chart, etc.

Data tool is also a useful tool in MINITAB. To create dummy variables/indicator variable which is often used in stat 106 (Analysis of Variance) course, you can use this tool as following

Data -> Code -> Numeric to Numeric.

At the current window dialog box, enter the value of old variable and new variable (dummy variable) into sub dialog boxes. Then click OK.

There is also a direct tool for creating dummy variables:

Calc -> Make Indicator Variables

Note: Minitab 14 also provides Survival Analysis tool which is not included in the Student version, yet it's in trial version. You can download this version (only works for 30 days) at <http://www.minitab.com/products/minitab/14/demo/>

There are also time series analyses, logistic regression, and some other analysis methods.

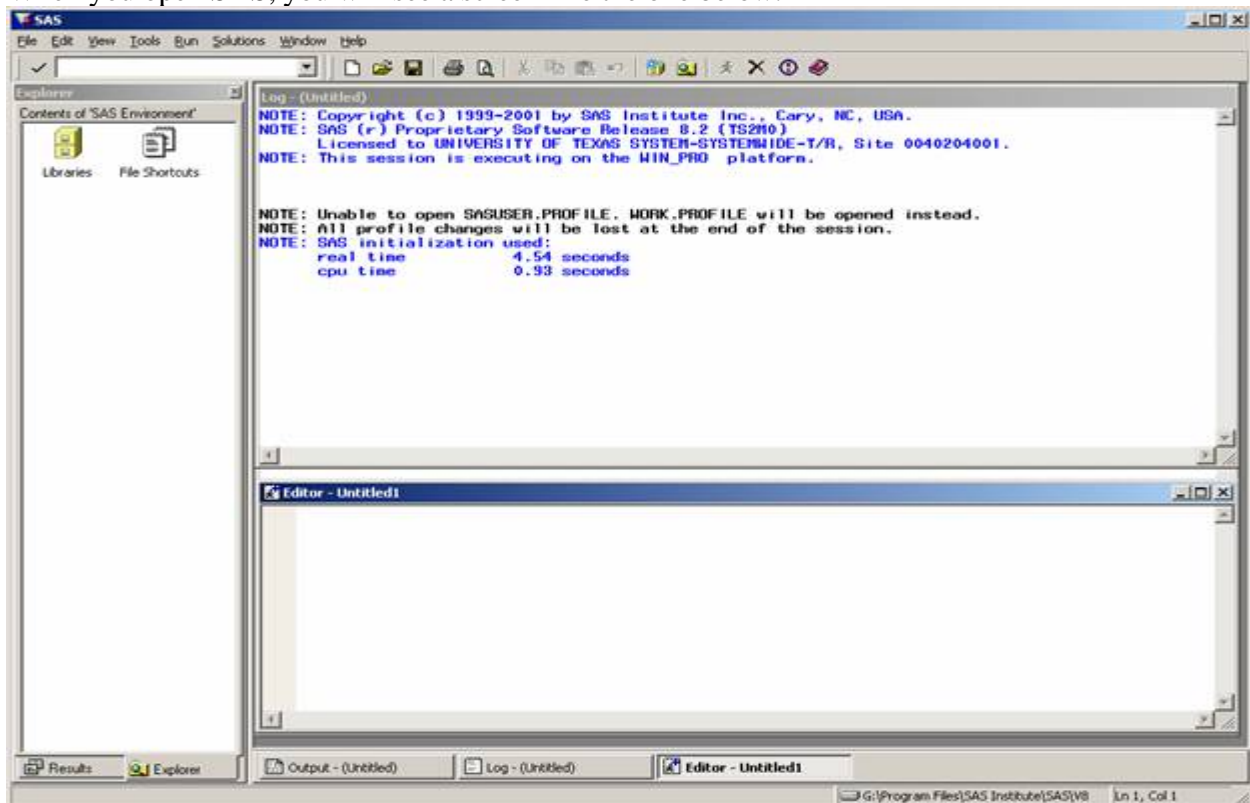
SAS

Introduction

SAS, a package for statistical data analysis, is available on two of the desk tops (window version) in the TA lab and on some labs for all UCD students, e.g. 73 and 75 Hutchison.

The types of files most commonly used with SAS are called data files, program files, log files, and list (or output) files.

When you open SAS, you will see a screen like the one below.



Results Window

This is a summary of what can be found in the output window after running a program. You can select specific output to edit or delete before printing or saving the results.

Explorer Window

This is used to browse SAS libraries and files.

Editor Window

In this window, you create, edit and run SAS program files. This is where you write your code or programs. When people refer to a "SAS program," they will usually be referring to this file.

Log Window

This window contains the details of the progress of running the program file. If there are errors that prevent SAS from running it successfully, you can look in the log window to see what errors might have occurred. Successful procedures are in blue, warnings are in green and errors are in red.

Output Window

If SAS is able to run the program successfully, you can find the output of the program in this window. You can view, copy, print and save the SAS output.

Writing a SAS program

There are several types of instructions you can give in your SAS program file. Each of the instructions, or statements, will end with a semicolon. The semi-colon tells SAS that the current command has ended. Without it, SAS would think the next line is part of the command as well. In the file, you can name your program and tell SAS to set your options, to read and transform your data, to analyze your data, and to print your results.

Commenting Your Code

There are two ways to comment a SAS program. One is to start with a * and end with a semi-colon as in

***This is a comment ;**

Problems occur with this type of commenting when you need to include a ; in your comment. To do this, you can comment starting with /* and ending with */ as in

/*This comment can include a ; and span over multiple lines*/

The title

The title step will give your program a name, which will appear on each page of the output.

Setting options

The options used most frequently are the options that tell SAS how to format its output. You can tell SAS how many columns you want in a line, how many lines you want in a page, or to suppress its printing of the date or the page number. The command

options nodate nonumber linesize = 80;

tells SAS to keep the output at 80 or fewer columns and to suppress the printing of the date and page number, which it would normally print on each page. For more help on options, look in the "SAS System Options" chapter in the *SAS Language* manual.

DATA and PROC Steps

After the title and options are set (which are not required for the program), there are two basis steps in creating a SAS program. These are the data step and the procedure step. The data step creates and/or edits a data set to be used in following data analysis procedures and almost always starts with **DATA**. The procedure step carries out analyses on a data file created from the data step and almost always starts with **PROC**.

Data files

Data files will be used to store data for use with SAS programs. There are a few ways to create or import a data file in SAS. You can either read data from a file or put the data directly into the program. The advantage of reading data in from a file is that the same data file can be used by several different programs. The advantage of putting the data directly into the program file is that the program will become slightly easier to manage. With a long data set, as data sets that commonly occur in practice, we will usually want to read data from a file.

To create a new data file, you can use the **datalines** statement.

```
DATA new;  
    INPUT name gender $ age;  
DATALINES;  
bob    M 25  
jane   F 23  
charlie M 27  
mary   F 26  
;
```

The **INPUT** statement names and indicates the type of each of the variables. The \$ after gender indicates that gender is a categorical, rather than numerical, variable. The **DATALINES** statement tells SAS to read the following as data with the above variable names.

One way to import a data set into SAS is to use the **INFILE** statement. Suppose we had the above data set in a file called new.txt. We could then run the following:

```
DATA new;  
    INFILE 'c:\new.txt';  
    INPUT name gender $ age;
```

Another way to import a data set into SAS is to use the SAS Import Wizard. In the menu, go to File, then Import Data. From here you can choose the options that correspond with the file you want to read in.

Once you are finished with this, you can call this data set in future procedure steps with the **DATA = new** statement.

Naming the data

SAS allows the user to use more than one data set in the same program. To enable the management of multiple data sets, SAS has the user give a name to each data set. A valid SAS name has no more than eight characters, begins with either a letter or the underscore character (_), and contains only letters, numbers, or underscore characters. This restriction will apply to variable names as well as to names of data sets.

Transforming data

If your data require transformations, you can do it in this step.

While you are in the data set, you can transform your data. For example, we might read in data with one variable that follows a lognormal distribution and two variables that follow a normal

distributions.

```
data norm;
  infile 'abtable.dat';
  input lognorm norm1 norm2;
  norm3 = log(lognorm);
  norm4 = norm1 + norm2;
```

Then each observation the variable norm3 will be the natural log of the corresponding observation of the variable lognorm, and each observation of the variable norm4 will be the sum of the corresponding observation of the variables norm1 and norm2. We may use the variables norm3 and norm4 later in the program for data analysis.


The PROC step

In this step, you can print your data, get descriptive statistics, analyze the data, and print graphs. Each step has different options but each ends with the statement

RUN;

Many procedures are available in the SAS environment. Frequently used are the **UNIVARIATE** procedure for summary statistics, stem-and-leaf plots, and normal probability plots, the **PRINT** procedure for seeing the data, and the **PLOT** procedure for X versus Y plots. In addition to the procedures already mentioned, basic procedures for statistical analysis, such as the **FREQ** procedure for analyzing tabulated data and the **MEANS** procedure to test for equal means, are available in what is known as base SAS. To get help on any of these procedures, look in the *SAS Procedures Guide* (This can be found online). More advanced procedures, such as the **ANOVA** procedure for analysis of variance, the **REG** procedure for regression analysis, and the **GLM** procedure, which can handle analysis of covariance as well as analysis of variance and regression, are part of the package known as SAS/STAT. See one of the two *SAS/STAT User Guide* volumes for details on how to use each of these procedures.

Running the SAS program

You can run your program at any time during the process of writing your code, though it may not always run smoothly if the code is not finished. Make sure all your procedures end with **RUN;** To run your program, click on the  button at the top of the SAS window. From here you can view the log and output files.

Examples

Data file

The file below (part of it is attached), apples.dat, contains one variable for the yield and two variables for types of treatments. Each line of the file represents one observation.

```
287 1 1
271 1 2
234 1 3
189 1 4
```

.....

Program file

The following program file, apples.sas, analyzes the above data set using analysis of variance.

```

title 'Randomized Block Design: Apple Yields';
options nodate linesize = 80;
data apples;
  infile 'apples.dat';
  input yield block cover;
proc anova;
  class block cover;
  model yield = block cover;
  means block cover;
run;

```

Log file

The log file contains any messages given by SAS while running the program. We will not show the log file here.

Output file

This file contains the output of the SAS program, given that the program runs successfully.

Below is the (edited) output found in apples.lst

```

Randomized Block Design: Apple Yields          1
  Analysis of Variance Procedure
    Class Level Information

```

```

Class  Levels  Values
BLOCK    4    1 2 3 4
COVER    6    1 2 3 4 5 6

```

Number of observations in data set = 24

```

Randomized Block Design: Apple Yields          2          Analysis of Variance Procedure

```

Dependent Variable: YIELD

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	48602.333333	6075.291667	3.89	0.0113
Error	15	23432.166667	1562.144444		
Corrected Total	23	72034.500000			

```

R-Square      C.V.      Root MSE      YIELD Mean
0.674709      14.33326  39.523973     275.75000

```

Source	DF	Anova SS	Mean Square	F Value	Pr > F
BLOCK	3	47852.833333	15950.944444	10.21	0.0006
COVER	5	749.500000	149.900000	0.10	0.9915

Randomized Block Design: Apple Yields
Analysis of Variance Procedure

3

Level of	-----YIELD-----		
BLOCK	N	Mean	SD
1	6	235.500000	37.1739156
2	6	248.500000	45.0987805
3	6	268.666667	28.6333139
4	6	350.333333	24.5085019

Level of	-----YIELD-----		
COVER	N	Mean	SD
1	4	284.500000	22.1585198
2	4	267.750000	59.2023367
3	4	275.250000	71.2805490
4	4	270.250000	88.6467710
5	4	277.250000	54.7197405
6	4	279.500000	61.9058425

2. Proc reg example

```

/* option clb for 95% CI of beta*/
/* option xpx for X'X, X'y, y'x, y'y partition matrices*/
/* option i for inv(X'X), beta.hat, beta.hat, sse partition matrices*/
/* option covb for covariance-variance matrix of beta.hat */
proc reg data = demo.example1;
  model y = x1 x2 / clb xpx i covb;
  output out=A1 p = pred stdp= stdpred r= res;
  test1: test x1=0, x2=0; /*test beta.1= beta.2 = 0 */
  test2: test x1-x2=0; /*test beta.1 = beta.2 */
  test3: test x1 -5*x2 = 0; /*test beta.1 = 5beta.2 */
run;

```

R

Introduction

S-PLUS is a statistical tool based on the S language originally developed by AT&T Bell Labs. Recently a team of statisticians has developed a freeware package called R which is essentially compatible with S-PLUS. Both S-PLUS and R are available on the department Unix servers. Windows versions of each are also available and are loaded onto most of the Windows machines in the department computer lab and in the student offices. Although the two languages are nearly identical, there are some slight differences in implementation. One difference is that S-PLUS creates a permanent file for every variable or function you create while programming. You should remove these files when you are done with them. In R the variables and functions are only created temporarily in RAM; they will not be stored for later use unless you choose to save your workspace when exiting the program.

More detailed information on the differences between the two languages and files for installing R can be found at <http://www.r-project.org/>.

Only R will be discussed here.

Getting into R

There are a few ways to get into R. One way is to open R in a Windows environment, by opening the R program. Another is to open R on a UNIX system. You can do this by typing R at the command prompt on the system you are currently on.

R will give you some opening messages and a prompt. Your prompt, which will probably be > on a line by itself, notifies you that R is ready to receive a command. Note: if you see a + as the prompt, R wants you to complete a command that you started on a previous line.

Getting out of R

1. At the R prompt, enter

```
> q()
```

Note that the parentheses are important, even when there is nothing inside them.

2. After R quits, you will see a prompt like neyman:5> (if you are on a UNIX server). If you have other things to do in neyman, you can do them now; enter cd (which is short for “change directory”). if you need to return to your main directory. Otherwise, enter logout at the prompt.

Getting Help

To get help on any command, enter **help(command)** or **?command** at the R prompt, where command is the function on which you want help. For example, **help(mean)** or **?mean** would give you help on the mean function in R. If you don't know the name of the function you want, you can use **help.search(“topic”)** to get a list of functions about that topic.

Another invaluable resource is the web, especially Google. Just type “R + topic” and you will get immediate get helpful results on any topic in R.

Making comments

R allows you to make comments, words that you type in but do not affect the action of the program. The # symbol tells R to ignore the rest of the line.

```
> # This is a comment.
```

Defining a variable

There are a couple of ways to define variables in R. The simplest way is to use one of the assignment operators, <- or =

```
> x <- 5
```

```
> x = 5
```

These will assign the value 5 to the variable x.

Listing variables

To see what variables you have created so far, use the ls command, with empty parentheses:

```
> ls()
```

```
[1] "x"
```

Removing data

If you have data that you no longer need in R, you can use the rm function to delete it. Suppose that you do not need the variable x any longer. Since you do not want to clutter your directory with useless variables, you would enter

```
> rm(x)
```

to delete the variable x from your current workspace.

Note: If you want to remove all objects to clear up the space so you can start running a new job, you would enter

```
> rm(list=ls())
```

Entering data

As shown in the example above, the simplest way to enter data is to use an assignment operator. You do not have to declare variables in R as you have to do in other programming languages; the program will create the variables as they are needed.

There are many types of data in R, including vector, matrix, list, and frame data. Frames are versatile but complicated, so we will not discuss them here. The variable that we created above is a vector variable with length 1. There are no real scalars in R. All scalars are considered a vector of length 1. We will discuss the creation of other types of variables.

Creating vectors

```
> x1 <- c(2, 5, 1, 7)
```

creates a vector of length 4, with elements 2, 5, 1, and 7.

```
> x2 <- 1:5
```

creates a vector of length 5, with elements 1, 2, 3, 4, and 5.

```
> x3 <- rep(3, 7)
```

creates a vector of length 7, with all elements having value 3.

```
> x4 <- seq(2, 7, 0.5)
```

creates a vector with elements 2, 2.5, 3, ..., 7.

If you have a longer data set, you might want to use the scan function to enter your data. Below is an example of the scan function.

```
> x5 <- scan()
```

```
1: 1 4 2 8 5 7
```

```
7: 7 6 9 2 3 0
```

```
13:
```

Enter a blank line at the prompt to signal that you are done entering data for the variable. The variable x5 will have 12 elements: 1, 4, 2, 8, 5, 7, 7, 6, 9, 2, 3, and 0.

Creating matrices

You will generally use the matrix command to create a matrix. Suppose that you want to create a 3-by-4 matrix using the data from the variable x5. Then you would use the command

```
> x6 <- matrix(x5, nrow=3, ncol=4)
```

Note that R fills matrices by columns by default. If you wanted to fill in the matrix by rows, you would use the option `byrow=T`:

```
> x7 <- matrix(x5, nrow=3, ncol=4, byrow=T)
```

Creating lists

Lists are a very useful tool in R. They let you combine data from different classes into a single variable. To create a simple list, use list function:

```
> list1 <- list(name="Bob", age=24, body=list(height=70, weight = 140))
```

The command above will create a list with three elements: the first is a character vector of length 1 containing the word "Bob", the second is a numeric vector of length 1 containing the number 24 and the last is a list containing two elements of height and weight.

Loading data from a file

You may want to load data from a file with a .txt or .dat extension. Usually this type of data is in a table format. You can use the command `read.table()` to bring the data into R. For example, if you want to read in a text file where the first row is the header (i.e. variable names) and the file is named "bob.txt", you would want to use the following command:

```
> bob <- read.table(file = "bob.txt", header = TRUE)
```

The above would create a new variable called "bob" which would be a data frame of the data in "bob.txt".

Checking your data

To see your data, enter the name of the variable by itself. Then R will return the contents of the variable to you. Try looking at the contents of x1.

If you find out that you made a mistake in entering your data, you have several options. If you have a short data set and do not mind entering the whole thing again, you may do that. If you have a long data set and just a few changes to make, you would probably want to change just a few elements individually. To do that, you would have to know how to access individual elements of a vector,

matrix, or list.

Accessing individual elements of a vector

The square brackets enable you to access individual elements of a vector.

```
> x1[1]
```

```
[1] 2
```

This returns the first element of the vector x1.

You can also access multiple elements at once.

```
> x1[c(2,4)]
```

```
[1] 5 7
```

This returns the 2nd and 4th element of the vector x1.

You can use logical vectors to access elements as well.

```
> x1[x1 < 3]
```

```
[1] 2 1
```

This returns the elements of the vector x1 that are less than 3

The symbol ! represents “not.”

```
> x1[!(x1 < 3)]
```

```
[1] 5 7
```

This returns the elements of the vector x1 that are not less than 3.

The symbol | represents “or”.

```
> x1[x1 > 6 | x1 < 3]
```

```
[1] 2 1 7
```

This returns the elements of the vector x1 that are greater than 6 or less than 3.

Suppose that you wanted the third element of x1 to be a 3 instead of a 1. Then you would give the command

```
> x1[3] <- 3
```

If you check the contents of x1 again, you will see the change.

Accessing elements of a matrix

Interestingly, R allows users to access elements of a matrix as if the matrix were a vector with the columns stacked, one on top of the next. Thus, the eighth element of the 3-by-4 matrix x7 would be the second element in the third column. Usually, you would want to use the row and column references, so you would use x7[2,3] to access that element. To access a whole second row of that matrix, you would omit the column index:

```
> x7[2,]
```

Similarly, you would omit the row index to access an entire column of a matrix.

Accessing elements of a list

There are two main ways to access the list that we created. First, we could use the name of the list item. For example, list1\$age would get the age of the person described by the variable list1. The

other way is to use double brackets; `list1[[2]]` would also get the age, since age is the second item of the list.

Making changes through an editor

If you have many changes to make, you can use the `fix` command:

```
> fix(x4)
```

will allow you to edit the contents of `x4`.

If you are using R on a Windows platform, you will be able to click and change what you want on a new window that opens. If you are on a UNIX system, you can edit the contents of the variable using `vi` editor commands. If you do not like the `vi` editor, you can enter the command

```
> options(editor="pico")
```

before using the `fix` command, and you would use the `pico` editor to edit the contents of `x4`.

Labels in data

If you store variables and then leave them alone for a long time, you may not remember what the variables mean. R allows you to put labels in your data so that you can keep track of what the variables represent. To put labels on vector data, you would use the `names` command:

```
> names(x1) <- c("Joe", "Bob", "Phil", "Eric")
```

If you look at the contents of `x1` again, you will see a difference.

To label matrices, you would use the `dimnames` command:

```
> dimnames(x6) <- list(NULL, c("Joe", "Bob", "Phil", "Eric"))
```

would give no names to the rows but would give names to the four columns. For more information on `dimnames`, please read the help file.

Getting descriptive statistics

Some of the descriptive statistics available in R are **mean**, **median**, minimum (**min**), maximum (**max**), **sum**, and variance (**var**). The **summary** command gives several descriptive statistics at once. Suppose that you want to know the maximum value of the vector `x5`. You would give the command

```
> max(x5)
```

or

```
> summary(x5)
```

Or you can use the corresponding built-in functions, e.g. **colSums**, **rowSums**, **colMeans**, **rowMeans**.

Operations in R

First, we will discuss operations on scalars or vectors. Then we will discuss operations involving matrices, which are more complicated. Finally, we will discuss the relationship between vectors and matrices in R.

Operations on scalars and vectors

R operates on scalars as you would expect, with `+`, `-`, `*`, and `/` having their usual meanings and `^` for exponentiation, `/%/%` for integer division, and `%%` for the modulus operator.

```
> 3 + 4
```

```
[1] 7
```

If you request an operation between a scalar and a vector, the program will apply the operation between the scalar and each element of the vector:

```
> x <- c(1, 3, 4)
> x - 1
[1] 0 2 3
```

For two vectors of the same length, R applies the requested operation element by element:

```
> y <- c(1, 2, 3)
> x - y
[1] 0 1 1
```

To obtain the inner product of two equal-length vectors x and y, we would use `sum(x * y)`:

```
> sum(x * y)
[1] 19
```

If the two vectors have different lengths, the program will repeat the elements of the shorter vector as many times as needed:

```
> z <- c(2, 3, 5, 7, 11, 13)
> z - x
[1] 1 0 1 6 8 9
```

Operations on matrices

R will also perform the above operations element by element on two matrices, provided that the matrices have the same dimensions. It will also perform those operations between vectors and matrices, element by element. Also available is the matrix multiplication operator, `%*%`. To use this with two matrices, we must be sure that the number of columns of the first matrix is equal to the number of rows in the second matrix. If we use the matrix multiplication operator between a vector and a matrix, R will attempt to convert the vector into a suitable matrix and then multiply:

```
> M <- matrix(z, nrow=2, ncol=3)
> M %*% y
  [,1]
[1,] 45
[2,] 56
> y %*% M
Error in y %*% M : non-conformable arguments
```

In the first case, we can multiply a 2x3 matrix by a 3x1 matrix, so R converts the vector y into a 3x1 matrix and takes the product of the two matrices. In the second case, R cannot multiply either a 3x1 matrix or a 1x3 matrix by a 2x3 matrix, so it returns an error message.

Vectors and matrices

Since R considers an n-element vector as distinct from either a 1xn matrix or an nx1 matrix, and a scalar as distinct from a 1x1 matrix, we need some operations to convert between vectors and matrices:

```
> x1 <- as.matrix(x)
```

```
> y1 <- as.matrix(y)
```

Converts the n-element vectors into nx1 matrices.

```
> xy1 <- cbind(x1, y1)
```

Binds the columns of two or more matrices. The two matrices should have equal numbers of rows, so that their columns can be put together into a single matrix.

```
> xy1
```

```
  [,1] [,2]
[1,]  1  1
[2,]  3  2
[3,]  4  3
```

```
> xy2 <- rbind(x1, y1)
```

Binds the rows of two or more matrices. The matrices should have equal numbers of columns.

```
> xy2
```

```
  [,1]
[1,]  1
[2,]  3
[3,]  4
[4,]  1
[5,]  2
[6,]  3
```

```
> as.vector(xy2)
```

```
[1] 1 3 4 1 2 3
```

Converts a matrix into a vector.

```
> M
```

```
  [,1] [,2] [,3]
[1,]  2  5 11
[2,]  3  7 13
```

```
> m <- as.vector(M)
```

```
> m
```

```
[1] 2 3 5 7 11 13
```

If the matrix has more than one column, the matrix elements will be extracted by columns.

```
> t(M)
```

```
  [,1] [,2]
[1,]  2  3
[2,]  5  7
[3,] 11 13
```

Takes the transpose of the matrix. Since t is a function in R, you should avoid creating a variable with that name.

```
> length(M)
```

```
[1] 6
```

```
> length(m)
```

```
[1] 6
```

Since M and m each have six elements, R returns 6 as the length of each.

```
> dim(M)
[1] 2 3
> dim(m)
NULL
```

For the dimension of a matrix, R return what we might expect. For vectors, though, we get NULL for the dimension, since they do not have any specific orientation.

```
> is.matrix(M)
[1] T
> is.vector(M)
[1] F
> class(M)
[1] "matrix"
```

If you see a variable in your directory and you do not remember whether it is a matrix or a vector (or something else), you can use the commands above to find out. Also available are `is.list`, `is.numeric`, `storage.mode`, `attributes`, and several other functions.

Vectors and lists

The `as.list` function converts a vector to a list. If you want to convert a list to a vector, use the `unlist` function. The `use.names` option tells R whether to use the names of the list items.

```
> unlist(list1)
  name   age body.height body.weight
"Bob"  "24"  "70"    "140"
> unlist(list1, use.names=F)
[1] "Bob" "24" "70" "140"
```

Functions

One reason that R is popular is that it has built-in functions for statistical analysis, as in a statistical package, and you can also write your own functions, as in a programming languages. Some of the available statistical functions are the linear regression function (`lm`), ANOVA (`aov`), and generalized linear models (`glm`). We will cover this topic only briefly, since there are many options for writing functions. Suppose you want to write a function to find the dot product of two vectors. We might write the simple function below.

```
> dot <- function(x, y)
+ { temp <- sum(x * y)
+ return(temp)
+ }
```

This function takes two arguments and will find the dot product of the given vectors. The braces indicate the content of the function. The variable `temp`, used within the function, will be discarded when the function ends.

```
> dot
function(x, y)
{
  temp <- sum(x * y)
  return(temp)
}
```

If you give R the function name by itself, the contents of the function will be returned. This is useful for learning how the built in functions work or for viewing your own functions.

```
> dot(x, y)
[1] 19
```

Just to check, we find the dot product of two vectors that we created today. Indeed, R return the dot product of the vectors. If you told the function to return more than one object, then a list would result. As is standard, you would use the dollar sign to access the list elements.

Some people have already written add-in functions and made them available for others to use. One example is the bootstrap function. Many of the functions are available through StatLib, which has an address <http://lib.stat.cmu.edu/>

A library has been set up on [neyman](#) so that people in this Department can share functions with one another.

Running UNIX commands from R

To run any UNIX command from R running on a UNIX platform, start the command with an exclamation mark. For example:

```
> !date
```

would tell your operating system to give you the date and time.

Sending output to a file

You have several options for sending output to a file.

1. If you want to record your whole R session into a file, you can enter the command:

```
script
```

from your UNIX prompt before you start R. After you exit the program enter the UNIX command:

```
exit
```

to stop recording. Then the session will be recorded into a file called typescript. We could give the script file a different name through the command **script filename**.

2. You can construct a file containing the commands that you want to give R. If your file is called infile.s, you can enter:

```
R < infile.s > outfile.out
```

at the UNIX prompt, and the program will run the commands listed in infile.s and send the output to outfile.out

3. If you are already in R, you can give the command:

```
> sink("outfile.out")
```

and any output will go to the named file instead of to the screen. When you enter:

```
> sink()
```

R will resume sending output to the screen. Another way to use a file containing your commands is the command:

```
> source("infile.s")
```

The file containing commands is sometimes called a source file.

There are some things to note when you send output to a file instead of the screen. First, if you use option 2 or 3, any error messages that you get will be sent to the screen and not to the file. If you use option 3, then output from UNIX commands will also be sent to the screen. One more difference under option 2 or 3 is that nothing will happen if you just give the name of a variable; you need to use the print function to see the contents of the variable.

When you use the sink function, the output file by default does not contain the commands that you give R. When you give the command

```
> options(echo=T)
```

each command you give will be returned as part of the output, until you give the command

```
> options(echo=F)
```

The apply function

Suppose you have a matrix and you want to calculate the sum of each column. One way is to use the sum function for each column, but that is an inefficient way to carry out the task. The apply function allows you to use the same command for all columns of the matrix. For example, suppose we give the following commands:

```
> z <- matrix(1:9, nrow=3, ncol=3, byrow=T)
```

```
> colsum <- apply(z, 2, sum)
```

```
> colsum
```

```
[1] 12 15 18
```

The user gives the name of the matrix and tells whether to apply the command to the rows or to the columns(1 for rows, 2 for columns). Finally, the user gives the name of the function to apply. In the example above, the first element of colsum was 12, since the first column of z had 1, 4, and 7. A 1 as the second argument to apply would tell R to apply the named function to the rows:

```
> rowsum <- apply(z, 1, sum)
```

```
> rowsum
```

```
[1] 6 15 24
```

Computational efficiency

Although it is convenient, the apply function is not always the most efficient method of obtaining summary statistics over rows or columns. In the case of sums, we can use matrix multiplication, a relatively fast operation in R, to reduce computing time:

```
> ones <- rep(1, 3) # repeats the number 1 three times
> colsum <- as.vector(ones %*% z)
> rowsum <- as.vector(z %*% ones)
```

The results will be identical to those above, but with less computing time involved.

Regression

Graphics

Enter `x11()` at the R prompt to get a graphics window on the UNIX platform. No command to open a graphics window is necessary on the Windows platform. Not all UNIX platforms will bring up graphics as the commands are given. To save the graphic as a PDF file, you first need to open a new graphics device.

```
> pdf("foo.pdf")
```

This opens a new PDF file named `foo`. Now you can plot as shown below. When you are finished with the details of your plot, you need to close the device.

```
> dev.off()
```

This closes the PDF file and makes it accessible in the current directory on the UNIX system you are on. If you are on a UNIX system on a Windows platform, it is easiest to get these graphics files from a File Transfer Program.

Available plots include the histogram, the scatterplot, the pie chart, and more. Suppose we want a histogram of the `x5` vector. Then we would enter

```
> hist(x5)
```

after returning to the window (use Alt-Tab, with the left-hand Alt button, to switch quickly between windows). To see the histogram, we would go back to the graphics window. If we wanted to label the histogram, we might give the command

```
> hist(x5, ylab="Count", xlab="Years in Davis", main="Students")
```

to get more information on the chart. The `main` option is available on most types of plots and gives the title of the plot, but the other options vary depending on the type of plot. To allow more than one plot on the same page, we might use the command

```
> par(mfrow=c(2,3))
```

to get a page that allows up to two rows and up to three columns of plots. The `mfc` option would also allow you the same number of plots, but it would fill the page by columns instead of by rows. As usual, the help files will give you more information about the particular plots that you want to use.

Multiple plots on the same set of axes

For this example, we will use the states data set, one of the data sets that come with R. We will plot two of the variables from the set, along with a linear fit and a local regression fit.

```
> x <- state.x77["Murder"]
> y <- state.x77["Life Exp"]
> plot(x, y, xlab="Murder", ylab="Life Exp")
```

This tells the program to draw an original plot.

```
> life <- lm(y ~ x)
> abline(coefficients(life))
```

The lines above tell R to fit a regression line to the data and add it to the graph.

```
> life2 <- loess(y ~ x)
> local <- fitted.values(life2)
```

The lines above tell R to fit a local regression model to the data. In order to plot a good curve we need to sort the x values. Then we can tell R to connect the fitted points of the local regression model.

```
> xorder <- order(x)
```

If we just wanted to sort the x data, we could use the `sort` function. Since we want to do more, we use the `order` function.

```
> lines(x[xorder], local[xorder], lty=2)
```

After telling R the order in which to connect the points, we tell it to plot using a dashed line. Through the `lty`, or line type, option, we can tell the program what kind of line to plot. Also available is the `points` function, which tells the program to put additional points on the graph. To distinguish between the new points and the original points, we could specify a plotting character, for example the `+` sign, through the `pch` option.

Note: If you want to create another graph without erasing the current one, use `X11()`.

Some other useful functions In R

save()	save R objects for use in later sessions
load()	load R objects or datasets written with the function save
write()	write data to a file
write.table() , write.csv	write a data frame to a text or CSV file
read.table() , read.csv	read a text or CSV file into a table format
head()	show the top few rows of a table
names()	get or set names of an object
class()	get class of an object
legend()	add a legend to a current plot
history()	display the commands history
sapply()	apply a function to each element of a vector or list
nrow(), ncol()	number of rows/columns in a vector, matrix or data frame
plot()	plot an x-y scatterplot
lm()	simple linear regression
anova()	compute analysis of variance table
aov()	fit an analysis of variance model
confint()	create confidence intervals for model parameters
var() , cov() , cor()	variance, covariance and correlation matrices
predict()	predictions for a model
qqplot()	normal probability plot (Q-Q plot)
pairs()	create a matrix of scatterplots
step()	stepwise regression
AIC()	model selection using AIC

Useful R Tips

R commands are case sensitive.

Pressing up on the keyboard will bring up previously entered commands. This is very helpful if you just need to change one thing in your previous command but do not want to type the entire command again.

If you want to read data from a file (.dat, .txt, .csv, etc.) and you don't want to write the address of the file, you can write **file.choose()** instead.

```
> read.table(file.choose( ))
```

This allows you to find the file on your computer as in a regular Windows software program. Note that this only works when using R on a Windows platform.

LaTeX

Introduction

(adapted from <http://www.colorado.edu/ITS/docs/latex/Unix/A0.0.html>)

LaTeX is a typesetting system which produces documents of the highest typographical quality, especially those involving mathematical and scientific notation. **LaTeX** differs from word processors (*Word*, *WordPerfect*, *PageMaker*, etc.) in several ways. It encourages authors to write well structured texts because this is how **LaTeX** works -- by specifying structure. The author composes a document by typing files of plain text, inserting occasional special **LaTeX** commands that control the logical structure of the document: markers and titles for each new section or subsection, lists, and markup of words that should appear in bold or italic style. The **LaTeX** software interprets the text files to produce a formatted document, automatically handling details of layout: line breaks, hyphenation and justification, placement of footnotes, numbering of pages and headings, references, bibliography.

By discouraging the author from spending time on the details of layout, fonts, or numbering the headings and equations, **LaTeX** can help the author produce a document excelling in structure and readability. The layout is professionally crafted, and the typesetting of mathematical formulae simply cannot be beat. Because the formatting requirements for theses and dissertations are so strict anyway (e.g., variations of font not allowed) such documents are particularly appropriate for using **LaTeX** instead of a word processor.

The real work of creating a **LaTeX** document is composing the text. This can be done with any editor at all. This text is considered the “source” of the document, and resides in one or more plain ASCII files named with the suffix `.tex`. Paragraphs are separated by a blank line. Figures and tables can be created using special **LaTeX** macros. Also, figures and tables in separate PostScript or PDF files can be inserted where needed.

There are two main ways to write using LaTeX, in a Windows environment or in a UNIX environment.

Latex window version

You can use latex (window version) through WinEdt program. This tool is installed to all window machines in the computer labs and TA offices of the Statistics Department. You can open WinEdt tool by clicking on its short cut icon or from the start menu. Here you will write your source code and save it as `.tex` file. Every `.tex` file starts with specifying the class of the document in one of the following ways:

for slide presentation

```
\documentclass[pdf]{prosper}
```

for writing an article (e.g. exam, report, thesis)

```
\documentclass[10pt]{article}
```

The most common classification of a document is as an article.

LaTeX automatically indents your document. You can set certain other properties, like this, before you start writing your document. For example, we can type.

```
\setlength{\parindent}{0pt}
```

This command will create your document without any indentation. You would put this command anytime before the next step. Please find more document set-up commands in the references found in this section.

Each document also starts with

```
\begin{document}
```

and ends with

```
\end{document}
```

with your text in between these two statements. It is recommended that you type these two statements when you first start your document, leaving space in between them for your text.

From here, you can your text type as in a regular word processing file. No special commands or notation are needed to type plain text.

If your document involves Greek letters such as μ , the LaTeX code will be `\mu`. Most commands start with a backslash, followed by the name of the letter, symbol, or character. If you don't know the command for a certain Greek letters, math symbol, or any other special character, you can use the toolbar in WinEdt instead. You can click on the character you want and WinEdt will write the LaTeX code for that character for you. This is a good way to learn all the commands you will regularly use.

Two backslashes, `\\`, results in starting a new line without a blank line in between. If you want a blank line, you can add another two backslashes. Another way to create a blank line or more than one is to use the command

```
\vspace{20pt}
```

Where you can adjust the amount of space you want to create by changing the 20pt. This command would create blank space that as tall as 20pt font. This command is useful when you are typing a test and need to leave large blank spaces for the students to write their answers.

If your document involves mathematical notation, such as an equation, you can use either

```
$...$
```

or

```
\begin{eqnarray*}
```

```
...
```

```
\end{eqnarray*}
```

To add a subscript and/or superscript to a character, use `_` and/or `^`, respectively.

The following is an example of two ways to type an equation using subscripts and superscripts.

```
$y_{i}=\beta_{0}+\beta_{1}x_{i}+\beta_{2}x_{i}^{2}+\epsilon_{i}, i=1,\dots,n$
```

or

```
\begin{eqnarray*}
```

```
$y_{i}=\beta_{0}+\beta_{1}x_{i}+\beta_{2}x_{i}^{2}+\epsilon_{i}, \quad i=1,\dots,n$
```

```
\end{eqnarray*}
```

Both of the above result in $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i, i = 1, \dots, s$. If the * after eqnarray were

removed, the equation would have an equation number to the right of it.

If your document involves matrices, you can use

```
\begin{array}{..}  
...  
\end{array}
```

The following is an example of this.

```
\Omega_k=\left(\begin{array}{cc}1&a_k\\b_k&2\end{array}\right)
```

The above will result in $\Omega_k = \begin{pmatrix} 1 & a_k \\ b_k & 2 \end{pmatrix}$. The parentheses are created by `\left` and `\right`. The

`{cc}` after `\begin{array}` tells the command to create a 2 by 2 matrix. If we changed it to `{ccc}`, it would create a 2 by 3 matrix. The data is entered by rows rather than by columns.

To make a table, use

```
\begin{tabular}{...}  
...  
\end{tabular}
```

Here is an example.

```
\begin{tabular}{|c|cccc|}\hline  
x&1&2&3&4&5\\ \hline  
y&6&7&8&9&10\\ \hline  
\end{tabular}
```

This will result in the following table:

x	1	2	3	4	5
y	6	7	8	9	10

The code `{|c|cccc|}` creates a 2 by 6 table with horizontal lines on the left, right and in between the first and second columns. The command `\hline` creates a vertical line where placed.

To center anything, type between

```
\begin{center}
```

and

```
\end{center}
```

If you want to input your data or anything else not involving commands, in latex, the following option would be useful

```
\begin{verbatim}  
...  
\end{verbatim}
```

Whatever is in between these two commands will show up as typed in your final document.

Here is an example.

```
\begin{verbatim}
1 2 3
a    b
\end{verbatim}
```

This will result in









```
1 2 3
a    b
```

exactly as is.

If you want make your presentation written in latex, you need to use several blocks starting with

```
\begin{slide}
...
\end{slide}
```

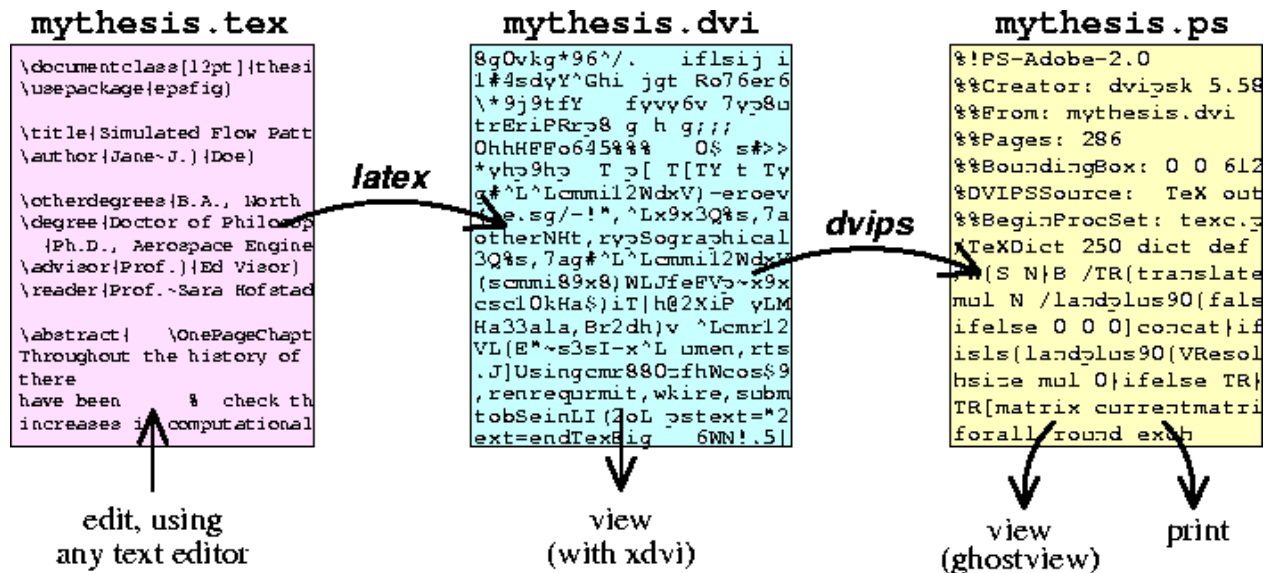
nested inside `\begin{document} ... \end{document}`

To execute your code and finally get .pdf or .ps files, you need to use buttons— , , , or . First, click . This executes your code and creates a DVI file. To view the file before converting it to a PDF or PS file, click . You can do this anytime while typing your document. It is highly recommended to execute the code and view the DVI file every once in awhile to make sure your commands are correct and/or your document looks how you want it to. From here you can create your PDF or PS file by clicking  or , respectively.

There are many resources, books and article for learning more about writing in LaTeX. One great article is “The Not So Short Introduction to LaTeX” by Tobias Oetiker, Hubert Partl, Irene Hyna and Elisabeth Schlegl. This, and many other free articles, can be found on scholar.google.com.

LaTeX Unix Version

The easy part of creating the **LaTeX** document is using the executable program `latex` to read the .tex files and create a fully formatted document in the form of a .dvi (**d**evice **i**ndependent) file. This file may be viewed by using the program [xdvi](#). To print the document requires one additional step, converting it to the form suited to your printer -- usually PostScript or PDF, using the program [dvips](#).



Three files are involved in the **LaTeX** process. The second and third files are relatively easy to create and view/print using the commands below.

```

unix> latex mythesis.tex (produce DVI file)
unix> xdvi mythesis.dvi (preview DVI file)
unix> dvips -o mythesis.ps mythesis.dvi (produce PostScript file)
unix> ghostview mythesis.ps (preview PostScript file)
unix> lp -d printername mythesis.ps (print PostScript file)

```

One advantage of using **LaTeX** is that the initial “source” file(s) can be composed on *any* computer keyboard. Any simple text editor can be used; no special keys or mouse/button controls are needed. The name of the text files must have the suffix `.tex`. Any text editor or word processor can be used to edit the `.tex` files. They must be saved as plain text, however, to be usable by **LaTeX**.

Before running `mydoc.tex` through LaTeX, you should take advantage of the spell-checking capabilities of the computer. Since the Unix filter “`spell`” by itself will consider LaTeX macros as misspellings, first filter each `.tex` file through “`detex -lnw`” and pipe the results through `spell`, e.g.,

```
unix> detex -lnw mydoc.tex | spell
```

Use the results to spot and correct spelling errors in the original `.tex` files, then compile the document to produce the DVI file `mydoc.dvi`:

```
unix> latex mydoc.tex
```

Please see the following reference for more details about LaTeX.
<http://www.stat.berkeley.edu/users/spector/latex2e.pdf>

Part III: Resources

1. Department and Campus Contacts

Department Office - MSB 4118 – 752-2361

Pat Aguilera – Management Service Officer, MSB 4115, aguilera@wald.ucdavis.edu
Contact for summer TA

Linda Brhely – Business Office Manager, MSB 4117, lbrhely@wald.ucdavis.edu

Ruby Castillo – Academic Personnel Supervisor, MSB 4113, rcastillo@wald.ucdavis.edu
Contact for academic personnel:leaves(leaves of absence, sabbatical leaves, short-term leaves), summer compensation, staff personnel, visa issues, payroll issues

Ann Maeda – Accounting Assistant, MSB 4118, atmaeda@wald.ucdavis.edu
Contact for expense reimbursement, purchasing, supplies, room scheduling, reservations, faxes, copies, office keys

Pete Scully – Graduate Program Coordinator, MSB 4118E, pscully@wald.ucdavis.edu
Contact for graduate student employment (GSR or TA), admissions, fellowships, letters of recommendation, student records, student exams (Written/Oral Ph.D., M.A. Arrangements, Basic Exam), room reservations (student events)

Min Yu – Undergraduate Program Coordinator, MSB 4118B, minyu@wald.ucdavis.edu
Contact for reader assignments, classroom or schedule changes, grade changes, textbook orders, evaluations coordination

Larry Tai – System Manager and Programmer, MSB 4206, larry@wald.ucdavis.edu
Contact for computer purchases, computer support, activation of ID Card for labs and building, check out laptops for discussion sections

Urban Cabbage – Computer Resource Specialist, MSB 4202, urban@wald.ucdavis.edu
Contact for computer support

General, support@wald.ucdavis.edu or help@wald.ucdavis.edu

Campus

Class Mailing List, <http://email.ucdavis.edu/eml>

Contact to create and manage class mailing lists

Classrooms, Hotline 752-3333, <http://registrar.ucdavis.edu/schedule>

Contact for on-call support for using the classroom media equipment, keys to media cabinet or other supplies or special equipment, view actual pictures of the inside of classrooms

Graduate Student Association (GSA), <http://gsa.ucdavis.edu>

Contact for financial assistance, legal advice, grievance assistance, awards and other funding and events including Bagel-Donut Coffee day every Friday.

Graduate Studies, <http://gradstudies.ucdavis.edu>

Contact for graduate forms and general graduate student advice

IT Express, <http://itexpress.ucdavis.edu>

Contact for campus computing accounts, loginID's and passwords

Student Judicial Affairs – 3200 Dutton Hall, 752-1128, sja@ucdavis.edu, <http://sja.ucdavis.edu>

Contact for dealing with issues of plagiarism, cheating and academic misconduct

TA Consultants, 17 Wellman Hall, 752-6050, tac@ucdavis.edu, <http://trc.ucdavis.edu>

Contact to make an appointment to use scantron machines for grading scantrons, general TA information, class videotapings, mid-quarter evaluations

The Teaching Resources Center and the campus Information and Educational Technology center can assist you with your computing responsibilities. Because information changes often, it's best to consult their websites rather than to print information in this Handbook. Here are some useful links:

Faculty Technology Guide

<http://ftg.ucdavis.edu/>

Information and Educational Technology Home Page

<http://iet.ucdavis.edu/index.cfm>

Teaching Resources Center Guide to Instructional Media and Technology:

<http://trc.ucdavis.edu/TP/media.html>

See the TA Handbook, "The Beginning TA's Gude to Instruction at UC Davis," by the Teaching Resources Center for even more contacts and resources.

2. Teaching References

“First Day to Final Grade: A Graduate Student’s Guide to Teaching”

by Anne Curzan and Lisa Damour

- ISBN-10: 0472031880
- ISBN-13: 978-0472031887

“McKeachie’s TEACHING TIPS”

by Wilbert J. McKeachie and Marilla Svinicki

- ISBN-10: 0618515569
- ISBN-13: 978-0618515561

“The Joy of Teaching: A Practical Guide for New College Instructors”

By Peter Filene

- ISBN-10: 0807829420
- ISBN-13: 978-0807829424

“What the Best College Teachers Do”

By Ken Bain

- ISBN-10: 0674013255
- ISBN-13: 978-0674013254

“Successful Beginnings for College Teaching”

By Angela Provitera McGlynn

- ISBN-10: 1891859382
- ISBN-13: 978-1891859380

“Teaching American Students: A Guide for International Faculty and Teaching Assistants in Colleges and Universities”

By Ellen Sarkisian

- ISBN-10: 067402141X
- ISBN-13: 978-0674021419

“Teaching Statistics: A Bag of Tricks”

By Andrew Gelman and Deborah Nolan

- ISBN-10: 0198572255
- ISBN-13: 978-0198572251

GOOD LUCK!!!

APPENDICES

From Student Judicial Affairs

<http://sja.ucdavis.edu>

UC Davis Code of Academic Conduct
<http://sja.ucdavis.edu/files/cac.pdf>

Reporting Academic Misconduct
<http://sja.ucdavis.edu/files/reporting.pdf>

Disruption in Class
<http://sja.ucdavis.edu/files/disrupt.pdf>

Tips to Prevent Cheating
<http://sja.ucdavis.edu/files/tips.pdf>

Confronting In-Progress Cheating
<http://sja.ucdavis.edu/files/confront.pdf>

INDEX

adding a class	22
AI	22
adding/dropping	22
incompletes/grade changes	23
cheating	17, 63
class webpages	24–28
computing packages.....	30
LaTeX	55–59
MINITAB	31–35
R.....	42–54
SAS	36–41
contacts	60
copies	22
department office	60
discussion.....	5–11, 15
projects.....	9–11
dropping a class	22
grade changes.....	23
grading	12–16, 18
handouts.....	6, 9, 18, 21–22
homework solutions	21–22
incompletes	23
LaTeX	55–59
lecture	9
MINITAB	31–35
office hours	8, 18–19
PDF	28–29
projects.....	9–11
R.....	42–54
references	62
resources	60
department contacts	60
on campus	61
teaching references	62
SAS	36–41
scanning	28–29
solutions.....	13–15
homework	21–22
test.....	13–15
teaching referencecs.....	62
tests	
grading	12–16
monitoring.....	16–18
webpages.....	24–28

Responsibility of the Administration

The following recommendations are made for the administration:

1. Provide physical settings for examinations that minimize opportunities for cheating.
2. Provide technical assistance for instructors, through the Teaching Resources Center and other sources, in preparing and administering examinations to minimize opportunities for academic misconduct.
3. Appoint adjudication committees to deal promptly and justly with cases of alleged academic dishonesty so the procedures gain and hold the confidence of students and instructors. Basic procedural fairness is to be insured in all disciplinary procedures.³

Reporting Violations and Judicial Procedures

All alleged violations should be reported by instructors to the Office of Student Judicial Affairs.⁴ Centralization of authority, responsibility and record keeping is essential for fair and impartial administration of the Code. Students should report cases to the instructor or to Student Judicial Affairs. Student Judicial Affairs shall inform the instructor of alleged violations that have been reported by students.

Where cheating is discovered or suspected, the instructor should inform the student(s) involved and undertake an investigation of the matter. If the instructor is then convinced that a violation did occur, the following procedure should be used:

If the student admits a violation, the instructor may assign a grade of "0" or "F" to the examination or assignment on which cheating, plagiarism or other academic dishonesty occurred. If the student admits a violation of this Code not involving dishonesty (e.g. talking during an exam, but not sharing answers), the faculty member may assign an appropriate grade penalty. In either case, the name of the student and a brief description of the incident should be reported to Student Judicial Affairs for determination of the appropriate disciplinary penalty. Reporting is essential to assure that repeat offenders are penalized accordingly.

If the student does not admit a violation, the instructor should refer the case to the Office of Student Judicial Affairs, which will attempt to resolve the matter informally and, if necessary, arrange for adjudication by a faculty-student committee.

Any contested accusation of cheating, plagiarism or other academic misconduct in an examination or assignment submitted by a student to satisfy course requirements will be adjudicated by a faculty-student committee appointed by the Vice Chancellor for Student Affairs in consultation with the Committee on Committees of the Davis Division. In this case, the instructor should withhold the student's grade until after adjudication.

If, after adjudication, the student is found in violation, the grade should be assigned in accordance with paragraph 7 under "Responsibility of Faculty." If there is no violation, the instructor shall assign a grade based on the quality of the work completed. A student who is dissatisfied with the decision in his or her hearing may appeal the decision (see UC Davis *Administration of Student Discipline* on the Student Judicial Affairs website <http://sja.ucdavis.edu/>).

Grades are an evaluation of academic performance, and their assignment is a responsibility resting solely with the faculty consistent with the authority granted to the Academic Senate by the Regents.⁵ The assignment of a grade of "0" or "F" to work or tests in which academic dishonesty or other academic misconduct has occurred is based on the premise that the examination or assignment (1) cannot be regarded as the student's own work and

thus is invalidated and/or (2) the student's violation is unfair and/or disruptive and thus harmful to the instructor and others in the class.

The instructor shall not apply a more severe grade adjustment than recording a "0" or "F" on the specific assignment or test on which misconduct occurred. Where an instructor has referred a case for adjudication but has not received a report confirming a violation at the time that the course grades are due, the words "evaluation incomplete" should be written in the "Memorandum" column of the Course Report.

Specific procedures to be followed in the adjudication of cases of academic misconduct have been established by the campus (see the UC Davis *Administration of Student Discipline*). Procedural fairness is basic to the proper enforcement of University policies and campus regulations.⁶ Procedures must ensure fairness not only for students accused of academic dishonesty but also for victims of dishonest acts, e.g., other students. The confidence of students and instructors in the procedures must be maintained.

1. See Section 102.00 of *University of California Policies Applying to Campus Activities, Organizations, and Students (Part A)* (hereafter, *UC Policies*) for the *University Standards of Conduct* (available at the Office of Student Judicial Affairs).
2. *The Code of Academic Conduct was originally approved and issued by the Office of the Chancellor, with the support of the Davis Division of the Academic Senate, November 12, 1976. The Chancellor approved a revised Code of Academic Conduct in 1985, and again in 1987 and 1993, after the proposed revisions had been reviewed and endorsed by the Davis Division of the Academic Senate. The Code was updated and revised again in 1995, following review and endorsement of the revisions by the Davis Division and approval by the Chancellor. Under University policy, the Chancellor has the authority and responsibility for the establishment of student conduct regulations and the administration of student discipline. See Sections 103.00 through 104.80 of the UC Policies. The document implementing the system of student discipline on the Davis campus is the Administration of Student Discipline (available at the Office of Student Judicial Affairs).*
3. See Section 103.10 of the UC Policies.
4. Under the general direction of the Vice Chancellor for Student Affairs, the Office of Student Judicial Affairs is directly responsible for the administration of student discipline. See current *UCD Administration of Student Discipline, approved by the Chancellor Fall, 2006.*
5. Page 57(a), *Regents Standing Order, 105, Academic Senate, The Manual of the Academic Senate.*
6. See Sections 103.10 through 103.11 of the UC Policies.