WEEK 9: CHAPTER 13 & CHAPTER 14, PAIRED TWO SAMPLE PROBLEMS

TABLE OF CONTENTS

Page:

List of Figures

List of Tables

List of m.files

Assignment

Required reading

! Larsen, R. J. and M. L. Marx. 2006. An introduction to mathematical statistics and its applications, 4th edition. Prentice Hall, Upper Saddle River, NJ. 920 pp.
- " Read Chapter 13.3
- " Read Chapter 14.3

Understanding by Design Templates

Understanding By Design Stage I — Desired Results Week 9

LM Chapter 13.3, 14.3 (Paired Two-sample tests)

G Established Goals

- Using parametric and nonparametric paired tests

U Understand

- Both the paired t test and signed rank tests assume the distributions are symmetric
- Differences on the log scale are equivalent to ratios of geometric means (medians) on the natural scale.

Q Essential Questions

- What are the assumptions that matter for paired tests and how can you assess them?
- What is the relationship between the binomial probability distribution and the sign test?
- If their assumptions are met, is the paired t test more powerful than the signed rank test?

K *Students will know how to define (in words or equations)*

- **Asymptotic relative power efficiency**, one-sample and paired t test, one sample and Wilcoxon signed rank test, sign test

S *Students will be able to*

- Plot histograms to evaluate the assumptions of the paired tests
- Apply Matlab's one sample paired *t* test (ttest), sign test (signtest) and Wilcoxon signed rank (signrank) test to environmental data.

Understanding by Design Stage II — Assessment Evidence Week 9 (7/26-8/1 M)

Chapter 13.3, 14 (Paired) Two-sample tests

- **Post in the discussion section by 8/3 W 10 PM**
  - Under what conditions would a sign test be used in preference to a paired *t* or a signed rank test?
  - Under what conditions would a signed rank test be preferred over a paired t test?
- **HW 6 Problems due Wednesday 8/3/11 W 10 PM**
  - **Basic problems (4 problems 10 points)**
    - Problem 13.3.2 Depth perception data. Perform a paired *t* test.
    - Problem 14.3.1 Perform both a paired t test and Wilcoxon signed rank test. Hint Use Case study14.3.3 as a model. Report the actual p values
    - Problem 14.3.6 using the case study 13.3.1 data
    - Problem 14.3.8
  - **Advanced problems (2.5 points each)**
    - **Problem** Plot the appropriately scaled histogram of differences with normal pdf for Problem 13.3.2
    - **Problem** Reanalyze Problem 14.3.1 with a log transform, plot the histogram of differences and compare with the untransformed scale. Back transform your results and confidence limits to the natural scale. Hint the difference of means on the log scale is equivalent to the ratio of geometric means in the natural scale.
  - **Master problems (1 only, 5 points)**
    - skeweffects.m is a program designed to assess the effects of positive skew on the paired t test, sign test, and Wilcoxon signed rank test. It generates Pearson random numbers with a given mean (0), std (1), skew(0) and kurtosis (3) (the standard normal values are in parentheses). Evaluate the program and make a report on which test is affected most by positive skew.

Paired t-tests & the two-sample Wilcoxon signed rank test

The two-sample equal variance Student *t* test, the Welch test and the Wilcoxon rank sum test assume that the observations in the two samples are independent. In many experiments, the lack of independence between the two samples can be used to produce a much more powerful test of the null hypothesis. The paired *t*-test is a subset of a general class of tests called 'repeated measures tests.' By measuring the **same** subject under two different experimental conditions, one can reduce the effects of subject-to-subject variance on the outcome, producing a much more powerful test for treatment effects.

Imagine that you are designing an experimental protocol to test the effectiveness of a new drug, say a new cholesterol-lowering drug. One could select 40 subjects from the population of interest, and then randomly assign individuals to receive the drug or a placebo. After a suitable period, one could test for differences in serum cholesterol and test for differences using the independent samples Student's t test. However, there is a far better design available which would remove a major source of variability in the data. Using a paired t-test design, one could randomly assign half the patients to receive the placebo first, and the other half the drug first. After a suitable period, the serum cholesterol could be measured and both groups removed from the drugs or placebos. After a sufficient length of time to allow for the effects of the drug to be dissipate, one could give each patient the other treatment. Those receiving the placebo would receive the cholesterol-lowering drug, and those receiving the drug would receive the placebo. The response variable would be the difference between placebo and drug for each patient. By allowing each patient to serve as a control, one can remove most of the subject-to-subject variability in baseline cholesterol level.

While the paired Student's t test earns a prominent place in every statistics book, it is merely the one-sample *t* test (**Larsen & Marx, 2006, Section 7.4 p 489 ff**). Under the null hypothesis of no treatment effect, the expected difference between treatment and control is 0. So, the differences can be entered in stud1sample.m and compared to an expected mean of 0.

Fisher's sign test, which tests the number of + and - signs in the differences in response variables among paired subjects, is another way of analyzing paired data. Fisher's sign test is identical to a binomial test. The Wilcoxon signed rank statistic is the major non-parametric equivalent to the paired sample t test.

Paired designs can arise without having repeated measures of the same subject, but pairing observations by subject is by far the most common. If there is prior information, one can set the expected difference at values different from 0, as in the 1 sample t test.

If there is a strong correlation among subjects, the improvement in the power of the test can be dramatic. Drug trials with a paired design can often assess treatment effects with a small fraction of the number of subject required to assess a similar treatment effect with the independent samples t test.
Nonparametric 1-sample tests

Fisher's Sign Test

Fisher invented the deceptively simple, but very powerful sign test to test one-sample problems. It is available on several of the toolboxes, but it is easy enough to calculate using principles from early in the course. I'll use the data from example 7.4.3 to describe the test, which is described in the text in Section 14.2. If the true proportion is 62% then subtract this expected proportion from the observed proportion. The expected number of differences with positive signs should equal the expected differences with negative signs. This can be tested exactly with the binomial probability density function. Of the 12 banks surveyed, only 3 of the 12 had approval rates greater than 62%. What is the probability of observing that result, OR ONE MORE EXTREME, by chance. Using bernoull.m, that is easy to calculate. One has to calculate the binomial pdf for 0, 1, 2, and 3, given the null hypothesis that states that the probability of negative and positive deviations from the expected values should be 0.5 (equal likelihood of positive and negative deviations):

```
k=0:3;p1tailed=sum(bernoull(k,12,0.5))
p1tailed =    0.0730
```

In this case, the 2-tailed p value would by **0.146.** The sign test offers no evidence to reject the null hypothesis that the true proportion is 62%.

The sign test does not assume any underlying parametric distribution for the data, such as normally distributed errors. The relative efficiency of this test is 95% for n=6, declining to 63% for large n, IF THE ASSUMPTIONS OF THE T TEST ARE MET. The asymptotic relative efficiency of a statistic is a difficult concept, but there is a nice graphical display in Figure 6.4.5 in Larsen & Marx (2001, p. 385), shown as Figure 1. If the assumptions of the parametric test are not met, the power efficiency of the nonparametric test can be MUCH higher than the parametric test.

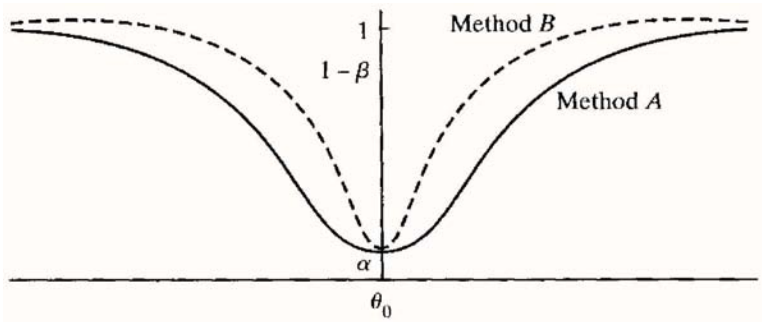

Figure 1. Power curves for two statistical tests, A & B. Method B has the higher asymptotic relative efficiency. The probability of Type II error (β) is lower for B at most values for the alternate hypothesis, and the power (1-β) is higher for Method B. The asymptotic relative efficiency reflects the reduction in sample size for the more powerful method to achieve the same power as the less powerful method. A relative power efficiency of 0.5 of test A relative to B, would mean that Method A would require twice as many samples to achieve the same power as method B.

Wilcoxon signed rank test

The sign test doesn't take into account the magnitude of the deviations from the expected value. There is a nonparametric test which DOES take into account the magnitude of the differences from the expected value. In the case of Example 7.5.3, the expected value is 62. The 12 banks had observed proportions shown below:

X=

59 65 69 53 60 53 58 64 46 67 51 59

The Wilcoxon signed rank statistic subtracts the expected value from each observed value. It then ranks the observations from smallest to largest, without respect to sign, and assigns ranks to these values.

Observed differences:

-3 3 7 -9 -2 -9 -4 2 -16 5 -11 -3

Ranked by absolute value, there are 3 sets of tied ranks, and these are assigned average values.

4 4 8 9.5 1.5 9.5 6.0 1.5 12 7 11 4

Note, that there were 2 values tied with abs(obs-exp)=2, assigned an average rank of 1.5, 3 absolute differences of 3, assigned an average rank of 4 (=mean ([3 4 5]), and two differences of -9, assigned an average rank of 9.5.

The Wilcoxon signed rank statistic is the sum of positive ranks (could also be the sum of negative ranks too). I've programmed the test and it is called by setting up a vector of expected values of 62 in this case

>> [2tailedpvalue,W]=Wilcoxsignrank(X,repmat(62,size(X)))
2tailedpvalue = 0.1458
W = 20.5 % The sum of positive ranks (4+8+1.5+7)

There is an exact test available for the Wilcoxon signed rank statistic (not to be confused with the Wilcoxon rank sum statistic to be discussed in the 2-sample tests section). The approximate test, is based on the normal distribution and produces a 2tailedpvalue of roughly 0.15.

Note that the p values for the Student's 1 sample test was p=0.125, the sign test was p=0.146 and the Wilcoxon signed rank test was p=0.146 too. This is not surprising as the power efficiency of the nonparametric tests is often quite high compared to their parametric equivalents.

Annotated outline (with Matlab scripts) for Larsen & Marx Chapter 8-9, 14

13 Randomized Block Designs

13.1 INTRODUCTION

13.2 THE F TEST FOR A RANDOMIZED BLOCK DESIGN

Theorem 13.2.1 & 13.2.2

Theorem 13.2.1. Suppose that k treatment levels are measured over a set of b blocks. Then

a. SSTOT = SSTR + SSB + SSE
b. SSTR, SSB, and SSE are independent random variables.

Proof. The independence of the three terms that combine to give SSTOT can be established using the same approach that was taken in Chapter 12. The details will be omitted.

Theorem 13.2.2. Suppose that k treatment levels, with means $\mu_1, \mu_2, \ldots, \mu_k$, are measured over a set of b blocks, where the block effects are $\beta_1, \beta_2, \ldots, \beta_k$. Then

a. When $H_0: \mu_1 = \mu_2 = \ldots = \mu_k$ is true, $SSTR/\sigma^2$ has a chi square distribution with k – 1 degrees of freedom.
b. When $H_0: \beta_1 = \beta_2 = \ldots = \beta_k$ is true, SSB/σ^2 has a chi square distribution with b – 1 degrees of freedom.
c. Regardless of whether the μ_j's and/or the β_i's are equal, SSE/σ^2 has a chi square distribution with (b-1)(k-1) degrees of freedom.

Theorem 13.2.3

Theorem 13.2.3. Suppose that k treatment levels with means $\mu_1, \mu_2 \ldots \mu_k$ are measured over a set of b blocks. Then

a. If $H_0: \mu_1 = \mu_2 = \ldots = \mu_k$ is true.

$$F = \frac{SSTR/(k-1)}{SSE/(b-1)(k-1)}$$

has an F distribution with k -1 and (b-1)(k-1) degrees of freedom.

b. At the α level of significance, $H_0: \mu_1 = \mu_2 = \ldots = \mu_k$ should be rejected if $F \geq F_{1-\alpha,k-1,(b-1)(k-1)}$

Theorem 13.2.4

Case Study 13.2.1 Acrophobia

```
%LMcs130201_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics 4th edition
% Case Study 13.2.1 Page 778-779
% Written by Eugene.Gallagher@umb.edu 12/3/2010; revised 2/15/2011
% Calls Trujillo-Ortiz et al. adTukeyAOV2.m from Matlab file central
% Tukey's test for additivity.
```

```
DATA=[8 2 -2
11 1 0
9 12 6
16 11 2
24 19 11];
boxplot(DATA,'sym','r*','labels',...
{'Contact Desensitization','Demonstration','Live Modeling'});
figure(gcf);pause
plot(DATA');
ax1=gca;
set(ax1,'xtick',[1 2 3]);legend('A','B','C','D','E');
xlabel('Therapy');ylabel('HAT Score');
figure(gcf);pause
pause % Needed so can see the box plots
sumDATA=sum(DATA);
meanDATA=mean(DATA);
% Since the design is balanced, either anova2 or anovan can be used.
[p,table,stats]=anova2(DATA,1)
% This will produce the ANOVA table as a figure. The results are
% printed out in table. stats could be sued for multcompare
pause
% The data can also be analyzed using anovan, producing identical results.
% anovan allows labeling of the ANOVA table.
y=DATA(:);
% convert the data into columns; drop the NaN elements, if any
g1=repmat(['A';'B';'C';'D';'E'],3,1);
g2=[repmat('CD',5,1);repmat('DP',5,1);repmat('LM',5,1)];
i=~isnan(y);y=y(i);g1=g1(i,:);g2=g2(i,:);
[p,table,stats] = anovan(y,{g1 g2},'model','linear',...
   'varnames',{'Blocks';'Therapies'});
% This is Example 13.2.1, comparing treatments
[c,m,h] = multcompare(stats,'ctype','tukey-kramer','display','on',...
   'dimension',2)
fprintf('Pairwise Difference \tLower 95\tEstimate\tUpper 95\n');
fprintf('\t\t%1.0f - %1.0f\t\t\t   %4.1f\t\t%4.1f\t\t%4.1f\n',c')
figure(h);
title(' '),xlabel(' ');xlabel('HAT score');title('Case Study 13.2.1'),
figure(gcf)
pause;
% Since the treatments were not replicated within blocks, a test
% for additivity should be run. If replicates were available a formal
% block by interaction test could have been run.
r1=repmat([1:5]',3,1);
r2=[repmat(1,5,1);repmat(2,5,1);repmat(3,5,1)];
X=[y r1 r2];
```

```
adTukeyAOV2(X,2,0.05)
```

Case Study 13.2.2 Rat poison

```
%LMcs130202_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics 4th edition
% Case Study 13.2.1 Page 780-781
% Written by Eugene.Gallagher@umb.edu 12/3/2010; revised 12/7/2010
% Calls other files
DATA=[13.8 11.7 14.0 12.6
12.9 16.7 15.5 13.8
25.9 29.8 27.8 25.0
18.0 23.1 23.0 16.9
15.2 20.2 19.0 13.7];
boxplot(DATA,'sym','r*','labels',...
{'Plain','Butter Vanilla','Roast Beef','Bread'});
figure(gcf);pause
plot(DATA');
ax1=gca;
set(ax1,'xtick',[1 2 3 4]);legend('1','2','3','4','5');
xlabel('Therapy');ylabel('HAT Score');
figure(gcf);pause
pause % Needed so can see the box plots
sumDATA=sum(DATA);
meanDATA=mean(DATA);
y=DATA(:);
% convert the data into columns;
g1=repmat(['S1';'S2';'S3';'S4';'S5'],4,1);
g2=[repmat('Pl',5,1);repmat('Bu',5,1);repmat('RB',5,1);repmat('Br',5,1)];
% find and delete any NaN elements, if any
i=~isnan(y);y=y(i);g1=g1(i,:);g2=g2(i,:);
[p,table,stats] = anovan(y,{g1 g2},'model','linear',...
   'varnames',{'Survey';'Flavor'});
[c,m,h] = multcompare(stats,'display','on','dimension',2);
figure(h)
title(' '),xlabel(' ');figure(gcf)
pause;
% Since the treatments were not replicated within blocks, Tukey's test
% for additivity should be run:
r1=repmat([1:5]',4,1);
r2=[repmat(1,5,1);repmat(2,5,1);repmat(3,5,1);repmat(4,5,1)];
X=[y r1 r2];
adTukeyAOV2(X,2,0.05)
```

13.2.1 **Tukey Comparisons for Randomized Block Data**

Theorem 13.2.5

Example 13.2.1

Tukey tests already incorporated in previous m.files for the case studies

13.2.2 **Contrasts for randomized block designs**

Case Study 13.2.3

```
%LMcs130203_4th.m
%LMcs130203_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics 4th edition
% Case Study 13.2.3 Page 778-779, The Transylvannia5 effect
% An example of linear contrasts for Randomized Block data.
% Written by Eugene.Gallagher@umb.edu 2/15/11
% Calls Trujillo-Ortiz et al. adTukeyAOV2.m from Matlab file central
% Tukey's test for additivity.
fprintf('\nAnalysis of Case Study 13.2.3: The Transylvannia Effect\n')
DATA=[6.4 5.0 5.8
   7.1 13.0 9.2
   6.5 14.0 7.9
   8.6 12 7.7
   8.1 6 11
   10.4 9 12.9
   11.5 13.0 13.5
   13.8 16.0 13.1
   15.4 25.0 15.8
   15.7 13.0 13.3
   11.7 14.0 12.8
   15.8 20 14.5];
[R,C]=size(DATA);
boxplot(DATA,'sym','r*','labels',...
{'Before Full Moon','During Full Moon','After Full Moon'});
figure(gcf);pause
plot(DATA');
ax1=gca;
set(ax1,'xtick',[1 2 3]);
set(ax1,'XtickLabel',...
  {'Before Full Moon','During Full Moon','After Full Moon'},'FontSize',9);
legend('Au','Se','Oc','Nv','De','Ja','Fe','Mr','Ap','My','Jn','Jl');
xlabel('Moon Phase');ylabel('Hospital Admission Rates');
figure(gcf);pause
pause % Needed so can see the box plots
sumDATA=sum(DATA);
meanDATA=mean(DATA);
% Since the design is balanced, either anova2 or anovan can be used.
[p,table,stats]=anova2(DATA,1)
% This will produce the ANOVA table as a figure. The results are
% printed out in table. stats could be sued for multcompare
pause
% The data can also be analyzed using anovan, producing identical results.
% anovan allows labeling of the ANOVA table.
```

```
y=DATA(:);
% convert the data into columns; drop the NaN elements
g1=repmat(['Au';'Se';'Oc';'Nv';'De';'Ja';'Fe';'Mr';'Ap';'My';'Jn';'Jl'],3,1);
g2=[repmat('BFM',12,1);repmat('DFM',12,1);repmat('AFM',12,1)];
group=[repmat(1,12,1) repmat(2,12,1) repmat(3,12,1)];
% find and delete any NaN elements, if any
i=~isnan(y);y=y(i);g1=g1(i,:);g2=g2(i,:);group=group(i);
% Use Trujillo-Ortiz's Levenestest
levenetest([y group],0.05);
[p,table,stats] = anovan(y,{g1 g2},'model','linear',...
    'varnames',{'Months';'Lunar Cycles'})
% This is Example 13.2.1, comparing treatments
[c,m,h] = multcompare(stats,'ctype','tukey-kramer','display','on',...
    'dimension',2);
fprintf('Pairwise Difference \tLower 95\tEstimate\tUpper 95\n');
fprintf('\t\t%1.0f - %1.0f\t\t\t   %4.1f\t\t%4.1f\t\t%4.1f\n',c')
figure(h);
title(' '),xlabel(' ');xlabel('Hospital Admission Rates');
title('Case Study 13.2.3'),
figure(gcf)
pause;

% Since the treatments were not replicated within blocks, a test
% for additivity should be run. If replicates were available a formal
% block by interaction test could have been run.
r1=repmat([1:R]',C,1);
r2=[repmat(1,R,1);repmat(2,R,1);repmat(3,R,1)];
X=[y r1 r2];
adTukeyAOV2(X,2,0.05)
% Note that there is evidence (p=0.046) to reject the additivity assumption

fprintf('\nCheck the additivity assumption with just 2 groups:\n')
% Reanalyze the data pooling 2 non-full moon periods.
D=mean(DATA(:,[1 3])')';D=[D DATA(:,2)];
plot(D');
ax1=gca;
set(ax1,'xtick',[1 2]);
set(ax1,'XtickLabel',...
   {'Not Full Moon','Full Moon'},'FontSize',9);
legend('Au','Se','Oc','Nv','De','Ja','Fe','Mr','Ap','My','Jn','Jl');
xlabel('Moon Phase');ylabel('Hospital Admission Rates');
figure(gcf);pause
pause % Needed so can see the box plots
[r,c]=size(D);
r1=repmat([1:r]',c,1);
```

```
r2=[repmat(1,r,1);repmat(2,r,1)];
X=[D(:) r1 r2];
adTukeyAOV2(X,2,0.05);
% p=0.0367; so still a strong interaction evident.
[p2,table2,stats2] = anovan(D(:),[r1 r2],'model','linear',...
   'varnames',{'Months';'Lunar Cycles'})
% Not covered in Larsen & Marx, but now it is possible to test formally
% for the interaction term.
Y=DATA(:);
G1=repmat(['Au';'Se';'Oc';'Nv';'De';'Ja';'Fe';'Mr';'Ap';'My';'Jn';'Jl'],3,1);
% set two groups: Not full moon and During Full moon
G2=[repmat('NFM',12,1);repmat('DFM',12,1);repmat('NFM',12,1)];
Group=[repmat(1,12,1) repmat(2,12,1) repmat(1,12,1)];
% find and delete any NaN elements, if any
i=~isnan(Y);Y=Y(i);G1=G1(i,:);G2=G2(i,:);Group=Group(i);
% Use Trujillo-Ortiz's Levenestest
levenetest([Y Group],0.05);
[p,table,stats] = anovan(Y,{G1 G2},'model',2,...
   'varnames',{'Months';'Lunar Cycles'})
% There should be no formal analysis of main effects of the main effects if
% I was taught in my graduate statistics class that if there is a
% significant interaction, show the interactions in an effects plot,discuss
% them and end the analysis.

% If there were no interactions, this would be a valid post hoc analysis:
% The following analysis uses the concept of linear contrast presented on
% page 751-758 in Larsen & Marx. The linear contrast between the full moon
% period and the other two phases was set a priori, so it can be tested and
% reported with an alpha level of 0.05.
LMatrix=[-1/2 1 -1/2];
planned=0;
anovalc(LMatrix, y, group,stats,planned)
```

---

function anovalc(LMatrix, y, group, stats, planned)

```
% format anovaLC(LMatrix, y, group, stats,planned)
% Input LMatrix
% Each row of the LMatrix should contain a linear contrast
%   LMatrix = [-1 1 0 0;-0.5 0.5 0 0] will return identical contrasts
%   y=data in a column vector
%   group is the column vector indicating group membership
%   stats is output from anova1, anova2 or anovan
%   planned =1 if the contrast was planned a priori
%   planned =0 if the contrast was not planned, in which case Scheffe
%             multipliers will be used.
% Written by Eugene D. Gallagher 12/7/2010
if nargin<5;planned=1;end
```

```
[R,C]=size(LMatrix);
% Create placeholder vectors for the output of the data
G=unique(group);   % Contains indices indicating treatment membership
n=zeros(1,C);
meanDATA=zeros(1,C);
sumDATA=zeros(1,C);
SSC=zeros(R,1);
F=zeros(R,1);
Fprob=zeros(R,1);
g=zeros(R,1);
seg=zeros(R,1);
tdf=tinv(0.975,stats.dfe);
for j=1:C
   i=find(group==G(j));
   n(j)=length(i);
   sumDATA(j)=sum(y(i));
   meanDATA(j)=mean(y(i));
end
for i=1:R % do each linear contrast
      sumLM=sum(LMatrix(i,:));
      sumabsLM=sum(abs(LMatrix(i,:)));
      fprintf('\nContrast Result Number %1.0f:\n',i)
      format rat
       disp(LMatrix(i,:));
       format
      if abs(sumLM)>=3*eps
         error('Linear contrasts must sum to 0');
      elseif abs((sumabsLM-2))>eps
         % This corrects an issue that is found in PASW, in which
         % ANOVA doesn't allow fractional linear contrasts and the
         % effects size and standard error are wrong if a contrast
         % such as [-1 -1 2 0] is used, in which case the sum of
         % the absolute value of the contrasts is 4, not 2 and the
         % estimated effect size and standard are 2x too large.
         LMatrix(i,:)=1/(sumabsLM/2)*LMatrix(i,:);
         fprintf(...
            'Linear Contrast %1.0f converted to equivalent form:\n',i)
         format rat
            disp(LMatrix(i,:))
         format
      end
      SSC(i)=sum(LMatrix(i,:).*sumDATA./n)^2/sum(LMatrix(i,:).^2./n);
      % Calculate the value of the linear contrast g (from Sleuth)
      g(i)=sum(LMatrix(i,:).*meanDATA);
      % The equation for the standard error of the linear contrast
```

```
        % can be found in Statistical Sleuth Chapter 6
        seg(i)=sqrt(stats.mse).*sqrt(sum(LMatrix(i,:).^2./n));
        F(i)=SSC(i)/stats.mse;
        Fprob(i)=1-fcdf(F(i),1,stats.dfe);
        if planned==1
            fprintf('The difference in means is %5.2f +/- %5.2f\n',...
                g(i),seg(i)*tdf)
        else
            Scheffe=sqrt((C-1)*finv(1-0.05,C-1,stats.dfe));
            fprintf(...
      'The difference in means is %5.2f +/- %5.2f (Scheffe Interval)\n',...
                g(i),seg(i)*Scheffe)
        end
        fprintf('\n Source      SS     df    MS    F   Prob\n')
        fprintf(...
          'Contrast    %4.1f     1   %4.1f %4.1f %5.3g\n',SSC(i),SSC(i),...
           F(i),Fprob(i))
  fprintf('   Error   %4.1f    %2.0f %5.2g\n',stats.mse*stats.dfe,...
      stats.dfe,stats.mse)
end
```

Questions 784-788

13.3 **THE PAIRED *t* TEST**

Theorem 13.3.1

Case Study 13.3.1

```
% LMcs130301_4th.m
% Case Study 13.3.1 p 790-791 in
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% Written by Eugene.Gallagher@umb.edu 11/14/10, revised 1/21/11, 2/28/11
% Revised: 1/21/11
% X(:,1) Hemoglobin before 60 km walk and X(:,2) after 60-km walk
X=[14.6 13.8
   17.3 15.4
   10.9 11.3
   12.8 11.6
   16.6 16.4
   12.2 12.6
   11.2 11.8
   15.4 15.0
   14.8 14.4
   16.2 15.0];
D=(X(:,1)-X(:,2));
[H,P,CI,STATS] = TTEST(X(:,1),X(:,2),0.05,'both');
fprintf(...
 'The mean "After-Before" difference in hemoglobin was %5.3f g/dl \n',...
 mean(D));
```

```
fprintf(...
 'The sd for the difference in hemoglobin was %5.3f g/dl \n',...
 STATS.sd);
fprintf(...
   'The CI for the difference in hemoglobin (g/dl) was [%5.3f %5.3f]\n',...
 CI);
fprintf('The paired t statistic was %6.4f with %2.0f df\n',...
    STATS.tstat,STATS.df)
fprintf('The paired t test 2-tailed p=%6.4f\n',P);

% Plot histogram and check for symmetry
binsize=0.25;  % Needed in order to properly scale the normal pdf
edges=-3:binsize:3;

% hist(D);
[N,BIN] = histc(D,edges);
bar(edges,N,'histc')
axis([-3.25 3.25 0 2.1])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('"After-Before" Difference in Hemoglobin (g/dl)','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:3])
figure(gcf);pause

% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
n=length(D);
mu_j=mean(D);
sigma_j=std(D); % sigmaj is the standard deviation; = 1 after Z transform
y_j=-3.25:0.1:3.25;
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;  % will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
h1=line(y_j,fy_j ,'Color','r','Parent',ax1,'Linewidth',2);
set(h1,'linestyle','--','color','r','linewidth',2)
s=sprintf('Case Study 13.3.1, %1.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
```

Case study 13.3.2

```
% LMcs130302_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 791. A case study solved by the paired t test
% Written by Eugene.Gallagher@umb.edu 11/14/10 Revised 11/16/10, 3/1/11
% X(:,1) Alamo rent-a-car, X(:,2)Avis rent-a-car
X=[48.99 51.99
   49.99 55.99
   42.99 47
   34.99 42.99
   42.99 44.95
   33.99 38.99
   59 69
   42.89 50.99
   47.99 49.99
   47.99 53.99
   35.99 42.99
   44.99 44.99]
D=(X(:,1)-X(:,2));
[H,P,CI,STATS] = TTEST(X(:,1),X(:,2),0.05,'both');
fprintf(...
 'The mean "Avis-Alamo" difference in price was $%5.2f.\n',...
 mean(D));
fprintf(...
 'The sd for the difference in price was $%5.2f.\n',...
 STATS.sd);
fprintf(...
   'The CI for the difference in price was [$%5.2f $%5.2f]\n',...
 CI);
fprintf('The paired t statistic was %6.4f with %2.0f df\n',...
   STATS.tstat,STATS.df)
fprintf('The paired t test 2-tailed p=%6.4f\n',P);
[p,h,stats] = signtest(D,0,0.05,'method','exact');
fprintf('The sign test exact p=%6.4f\n',p);
[p,h,stats] = signtest(D,0,'method','approximate');
fprintf('The sign test approximate p=%6.4f\n',p);
[P,H,STATS] = signrank(X(:,1),X(:,2),'alpha',0.05,'method','exact');
fprintf('The signed rank statistic is %4.1f\n',STATS.signedrank);
fprintf('The sign rank test exact p=%6.4f\n',P);
[P,H,STATS] = signrank(X(:,1),X(:,2),'alpha',0.05,'method','approximate');
fprintf('The signed rank z statistic is %6.4f\n',STATS.zval);
fprintf('The sign rank test approximate p=%6.4f\n',P);
% Plot histogram and check for symmetry
binsize=2;  % Needed in order to properly scale the normal pdf
edges=-12:binsize:4;
```

```
% hist(D);
[N,BIN] = histc(D,edges);
bar(edges,N,'histc')
axis([-12.1 4.1 0 4.1])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('"Avis-Alamo" Difference in Price','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:4],'Xtick',[-12:2:4],'XTickLabel',...
   {'$12','$10','$8','$6','$4','$2','$0','$2','$4'})
figure(gcf);pause
% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
n=length(D);
mu_j=mean(D);
sigma_j=std(D); % sigmaj is the standard deviation; = 1 after Z transform
y_j=-12.1:0.1:4.1;
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;  % will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
h1=line(y_j,fy_j ,'Color','r','Parent',ax1,'Linewidth',2);
set(h1,'linestyle','--','color','r','linewidth',2)
s=sprintf('Case Study 13.3.2, %1.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
```

13.3.1 Criteria for Pairing

13.3.2 **The equivalence of the paired t test and the randomized block ANOVA when k = 2**

Questions 795-796

13.4 **Taking a second look at statistics (choosing between a two-sample t test and a paired t test)**

Example 13.4.1 Comparing two weight loss plans

Example 13.4.2 Comparing two eye surgery techniques

Appendix 13.A.1 Minitab applications

14 Nonparametric statistics

14.1 Introduction

14.2 The Sign Test

Theorem 14.2.1

Case Study 14.2.1

```
% LMcs140201_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 804. A case study solved by the sign test
% Written by Eugene.Gallagher@umb.edu 11/16/10 Revised 11/16/10, 3/1/11
%
D=[7.02 7.35 7.32 7.33 7.15 7.26 7.25 7.35 7.38 7.20 7.31 7.24 7.34 ...
   7.32 7.34 7.14 7.20 7.41 7.77 7.12 7.45 7.28 7.34 7.22 7.32 7.4 ...
   6.99 7.1 7.3 7.21 7.33 7.28 7.35 7.24 7.36 7.09 7.32 6.95 7.35 ...
   7.36 6.6 7.29 7.31];
[p,h,stats] = signtest(D,7.39,0.05,'method','exact');
fprintf('\nThe sign test exact p=%6.4g\n',p);
[p,h,stats] = signtest(D,7.39,'method','approximate');
fprintf('The sign test approximate p=%6.4g;z=%6.4f\n',p,stats.zval);
[H,P,CI,STATS] = ttest(D,7.39);
fprintf('The one-sample t test 2-tailed p=%6.4g\n',P);
fprintf('The mean pH = %4.2f with 95%% CI: [%4.2f %4.2f]\n',mean(D),...
   CI(1),CI(2));
[P,H,STATS] = signrank(D,7.39,'alpha',0.05,'method','exact');
fprintf('The sign rank test exact p=%6.4g\n',P);
[P,H,STATS] = signrank(D,7.39,'alpha',0.05,'method','approximate');
fprintf('The sign rank test approximate p=%6.4g\n',P);
% Plot histogram and check for symmetry
binsize=.1;  % Needed in order to properly scale the normal pdf
edges=6.4:binsize:8;
% hist(D);
[N,BIN] = histc(D,edges);
bar(edges,N,'histc')
axis([6.35 8.05 0 21])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('pH','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:5:25],'Xtick',[6.4:0.2:8])
figure(gcf);pause
% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
n=length(D);
mu_j=mean(D);
sigma_j=std(D); % sigmaj is the standard deviation; = 1 after Z transform
y_j=6.35:0.01:8.05;
```

```
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;
cutoff=7.39;
fy_cutoff=n*binsize/(sqrt(2*pi)*sigma_j)*exp(-1/2*((cutoff-mu_j)./sigma_j).^2);
% will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
v=axis;
h1=line(y_j,fy_j ,'Color','r','Parent',ax1);
set(h1,'linestyle','--','color','r','linewidth',2)
h1=line([cutoff cutoff]',[0 fy_cutoff]','Color','b','Parent',ax1);
set(h1,'linestyle','-.','color','b','linewidth',3)
h2=line([cutoff cutoff]',[fy_cutoff v(4)]','Color','b','Parent',ax1);
set(h2,'linestyle','-.','color','b','linewidth',3)
s=sprintf('Case Study 14.2.1, %2.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
```

14.2.1 A Small-Sample Sign Test, Use the exact binomial

Case Study 14.2.2

```
% LMcs140202_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 806. A case study solved by the sign test
% Written by Eugene.Gallagher@umb.edu 11/16/10 Revised 11/16/10, 3/1/11
%
D=[4.8 4.0 3.8 4.3 3.9 4.6 3.1 3.7];
expected=3.55;
[H,P,CI,STATS] = ttest(D,expected,0.05,'both');
fprintf('\nThe one-sample t test 2-tailed p=%6.4g\n',P);
fprintf(...
'The mean caffeine = %4.2f (g/100g residue) with 95%% CI: [%4.2f %4.2f]\n',...
   mean(D), CI(1),CI(2));
[p,h,stats] = signtest(D,expected,0.05,'method','exact');
fprintf('The sign test statistic is %4.1f with exact 2-tailed p=%6.4g\n',stats.sign,p);
[p,h,stats] = signtest(D,expected,'method','approximate');
fprintf('The sign test z=%5.3f with approximate 2-tailed p=%6.4g\n',stats.zval,p);
[P,H,STATS] = signrank(D,expected,'alpha',0.05,'method','exact');
fprintf(...
'The sign rank test statistic is %4.1f with exact 2-tailed p=%6.4g\n',...
 STATS.signedrank,P);
[P,H,STATS] = signrank(D,expected,'alpha',0.05,'method','approximate');
fprintf('The sign rank test approximate 2-tailed p=%6.4g for z=%6.4f\n',P,STATS.zval);

% Plot histogram and check for symmetry
binsize=.2;  % Needed in order to properly scale the normal pdf
```

```
edges=3:binsize:5;
% hist(D);
[N,BIN] = histc(D,edges);
bar(edges,N,'histc')
axis([2.9 5.1 0 2.1])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('Caffeine Residue (g/100 g dry weight)','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:2],'Xtick',[3:0.2:5])
figure(gcf);pause
% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
n=length(D);
mu_j=mean(D);
sigma_j=std(D); % sigmaj is the standard deviation; = 1 after Z transform
y_j=2.9:0.01:5.1;
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;
cutoff=expected;
fy_cutoff=n*binsize/(sqrt(2*pi)*sigma_j)*exp(-1/2*((cutoff-mu_j)./sigma_j).^2);
% will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
v=axis;
h1=line(y_j,fy_j ,'Color','r','Parent',ax1);
set(h1,'linestyle','--','color','r','linewidth',2)
h1=line([cutoff cutoff]',[0 fy_cutoff]','Color','b','Parent',ax1);
set(h1,'linestyle','-.','color','b','linewidth',3)
h2=line([cutoff cutoff]',[fy_cutoff v(4)]','Color','b','Parent',ax1);
set(h2,'linestyle','-.','color','b','linewidth',3)
s=sprintf('Case Study 14.2.2, %2.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
```

14.2.2 Using the Sign Test for Paired Data (p. 807)

Case Study 14.2.3

```
% LMcs140203_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 807. A case study solved by the sign test
```

```
% Written by Eugene.Gallagher@umb.edu 11/16/10 Revised 11/16/10, 3/1/11
%
D=[15 13;12 8;12 12.5;14 12;13 12;13 12.5;13 12.5;12 14;12.5 12;12 11;
   12.5 10];
[p,h,stats] = signtest(D(:,2),D(:,1),0.05,'method','exact');
fprintf('\nThe sign test statistic is %4.1f with exact 1-tailed p=%6.4g\n',stats.sign,p/2);
[p,h,stats] = signtest(D(:,2),D(:,1),'method','approximate');
fprintf('The sign test z=%5.3f with approximate 1-tailed p=%6.4g\n',stats.zval,p/2);
% It is a 1-tailed p test to the left since the expectation is that mean
% circulation time is reduced by 4 months of cyclandelate.
[H,P,CI,STATS] = ttest(D(:,2),D(:,1),0.05,'left');
fprintf('The t statistic was %5.3f with %2.0f df.\n',STATS.tstat,STATS.df)
fprintf('The paired t test 1-tailed p=%6.4g\n',P);
fprintf(...
'The mean circulation time = %4.2f secs with 95%% CI: [%4.2f %4.2f]\n',...
   mean(D(:,2)-D(:,1)), CI(1),CI(2));
[P,H,STATS] = signrank(D(:,2),D(:,1),'alpha',0.05,'method','exact');
fprintf(...
'The sign rank test statistic is %4.1f with exact 1-tailed p=%6.4g\n',...
 STATS.signedrank,P/2);
[P,H,STATS] = signrank(D(:,2),D(:,1),'alpha',0.05,'method','approximate');
fprintf('The sign rank test approximate 1-tailed p=%6.4g for z=%6.4f\n',P/2,STATS.zval);

% Plot histogram and check for symmetry
binsize=.5;  % Needed in order to properly scale the normal pdf
edges=-4:binsize:2.5;
% hist(D);
[N,BIN] = histc(D(:,2)-D(:,1),edges);
bar(edges,N,'histc')
axis([-4.1 2.6 0 3.3])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('"After-Before" Mean Circulation Time (secs)','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:3],'Xtick',[-4:0.5:2.5])
figure(gcf);pause
% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
[n,c]=size(D);
mu_j=mean(D(:,2)-D(:,1));
sigma_j=std(D(:,2)-D(:,1)); % sigmaj is the standard deviation; = 1 after Z transform
y_j=-4.1:0.01:2.6;
```

```
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;
cutoff=0;
fy_cutoff=n*binsize/(sqrt(2*pi)*sigma_j)*exp(-1/2*((cutoff-mu_j)./sigma_j).^2);
% will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
v=axis;
h1=line(y_j,fy_j ,'Color','r','Parent',ax1);
set(h1,'linestyle','--','color','r','linewidth',2)
h1=line([cutoff cutoff]',[0 fy_cutoff]','Color','b','Parent',ax1);
set(h1,'linestyle','-.','color','b','linewidth',3)
h2=line([cutoff cutoff]',[fy_cutoff v(4)]','Color','b','Parent',ax1);
set(h2,'linestyle','-.','color','b','linewidth',3)
s=sprintf('Case Study 14.2.2, %2.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
```

Questions p 809-810

14.3 WILCOXON TESTS

14.3.1 **Testing** H_o: $\mu=\mu_o$

Theorem 14.3.1

14.3.2 Calculating $p_w(w)$

14.3.3 Tables of the cdf, $F_W(w)$

Case Study 14.3.1 Swell sharks

```
% LMcs140301_4th.m
% Case Study 14.3.1 from
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 815. A case study using Wilcoxon signed rank test
% Written by Eugene.Gallagher@umb.edu 11/16/10 Revised 11/16/10, 3/1/11
%
D=[13.32 13.06 14.02 11.86 13.58 13.77 13.51 14.42 14.44 15.43];
expected=14.6;
[P,H,STATS] = signrank(D,expected,'alpha',0.05,'method','exact');
fprintf(...
'\nThe signed rank test statistic is %4.1f with exact 2-tailed p=%6.4g\n',...
 STATS.signedrank,P);
[P,H,STATS] = signrank(D,expected,'alpha',0.05,'method','approximate');
fprintf('The signed rank test approximate 2-tailed p=%6.4g for z=%6.4f\n',P,STATS.zval);
[H,P,CI,STATS] = ttest(D,expected,0.05,'both');
fprintf('The one-sample t test 2-tailed p=%6.4g\n',P);
fprintf(...
'The mean TL/HDl = %4.2f with 95%% CI: [%4.2f %4.2f]\n',...
   mean(D), CI(1),CI(2));
```

```
[p,h,stats] = signtest(D,expected,0.05,'method','exact');
fprintf('The sign test statistic is %4.1f with exact 2-tailed p=%6.4g\n',stats.sign,p);
[p,h,stats] = signtest(D,expected,'method','approximate');
fprintf('The sign test z=%5.3f with approximate 2-tailed p=%6.4g\n',stats.zval,p);

% Plot histogram and check for symmetry
binsize=0.25;  % Needed in order to properly scale the normal pdf
edges=11:binsize:16;
% hist(D);
[N,BIN] = histc(D,edges);
bar(edges,N,'histc')
axis([10.9 16.1 0 max(N)+0.2])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('TL/HDl','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:max(N)],'Xtick',edges)
figure(gcf);pause
% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
n=length(D);
mu_j=mean(D);
sigma_j=std(D); % sigmaj is the standard deviation; = 1 after Z transform
y_j=10.9:0.01:16.1;
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;
cutoff=expected;
fy_cutoff=n*binsize/(sqrt(2*pi)*sigma_j)*exp(-1/2*((cutoff-mu_j)./sigma_j).^2);
% will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
v=axis;
h1=line(y_j,fy_j ,'Color','r','Parent',ax1);
set(h1,'linestyle','--','color','r','linewidth',2)
h1=line([cutoff cutoff]',[0 fy_cutoff]','Color','b','Parent',ax1);
set(h1,'linestyle','-.','color','b','linewidth',3)
h2=line([cutoff cutoff]',[fy_cutoff v(4)]','Color','b','Parent',ax1);
set(h2,'linestyle','-.','color','b','linewidth',3)
s=sprintf('Case Study 14.3.1, %2.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
```

Questions p 816-817

14.3.4 **A large sample Wilcoxon signed rank test**

Theorem 14.3.2

Theorem 14.3.3

Case Study 14.3.2 Heroine addiction

```
% LMcs140302_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 819. A case study using Wilcoxon signed rank test
% Written by Eugene.Gallagher@umb.edu 11/16/10 Revised 12/12/10, 3/1/11
%
D=[51 53 43 36 55 55 39 43 45 27 21 26 22 43];
expected=28;
[P,H,STATS] = signrank(D,expected,'alpha',0.05,'method','exact');
fprintf(...
'\nThe signed rank test statistic is %4.1f with exact 1-tailed p=%6.4g\n',...
 STATS.signedrank,P/2);
[P,H,STATS] = signrank(D,expected,'alpha',0.05,'method','approximate');
fprintf('The signed rank test approximate 1-tailed p=%6.4g for z=%6.4f\n',P/2,STATS.zval);
[H,P,CI,STATS] = ttest(D,expected,0.05,'right');
fprintf('The one-sample t test 1-tailed p=%6.4g\n',P);
fprintf(...
'The mean Q Score = %4.2f with 95%% CI: [%4.2f %4.2f]\n',...
   mean(D), CI(1),CI(2));
[p,h,stats] = signtest(D,expected,0.05,'method','exact');
fprintf('The sign test statistic is %4.1f with exact 1-tailed p=%6.4g\n',stats.sign,p/2);
[p,h,stats] = signtest(D,expected,'method','approximate');
fprintf('The sign test z=%5.3f with approximate 1-tailed p=%6.4g\n',stats.zval,p/2);

% Plot histogram and check for symmetry
binsize=2;  % Needed in order to properly scale the normal pdf
edges=20:binsize:56;
% hist(D);
[N,BIN] = histc(D,edges);
bar(edges,N,'histc')
axis([19.5 56.5 0 max(N)+0.2])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('Q Score','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:max(N)],'Xtick',edges)
figure(gcf);pause
% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
```

```
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
n=length(D);
mu_j=mean(D);
sigma_j=std(D); % sigmaj is the standard deviation; = 1 after Z transform
y_j=19.5:0.01:56.5;
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;
cutoff=expected;
fy_cutoff=n*binsize/(sqrt(2*pi)*sigma_j)*exp(-1/2*((cutoff-mu_j)./sigma_j).^2);
% will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
v=axis;
h1=line(y_j,fy_j ,'Color','r','Parent',ax1);
set(h1,'linestyle','--','color','r','linewidth',2)
h1=line([cutoff cutoff]',[0 fy_cutoff]','Color','b','Parent',ax1);
set(h1,'linestyle','-.','color','b','linewidth',3)
h2=line([cutoff cutoff]',[fy_cutoff v(4)]','Color','b','Parent',ax1);
set(h2,'linestyle','-.','color','b','linewidth',3)
s=sprintf('Case Study 14.3.2, %2.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 819. A case study using Wilcoxon signed rank test
% Written by Eugene.Gallagher@umb.edu 11/16/10 Revised 12/12/10
%
D=[51 53 43 36 55 55 39 43 45 27 21 26 22 43];
hist(D-28);figure(gcf);pause
hist(log(D)-log(28));figure(gcf);pause
M=28;
[H,P,CI,STATS] = ttest(D,M,0.05,'right');
fprintf('\nThe paired t test 1-tailed p=%6.4g\n',P);
fprintf('The mean Q score = %4.2f with 95%% CI: [%4.2f %4.2f]\n',...
   mean(D), CI(1),CI(2));
[H,P,CI,STATS] = ttest(log(D),log(M),0.05,'right');
fprintf('\nThe paired t test of log transform 1-tailed p=%6.4g\n',P);
[p,h,stats] = signtest(D,M,0.05,'method','exact');
fprintf('The sign test exact 1-tailed p=%6.4g\n',p/2);
[p,h,stats] = signtest(D,M,'method','approximate');
fprintf('The sign test approximate 1-tailed p=%6.4g\n',p/2);
[P,H,STATS] = signrank(D,M,'alpha',0.05,'method','exact');
fprintf('The sign rank test exact 1-tailed p=%6.4g\n',P/2);
[P,H,STATS] = signrank(D,M,'alpha',0.05,'method','approximate');
fprintf('The sign rank test approximate 1-tailed p=%6.4g\n',P/2);
```

14.3.5 **Testing** H_o: $\mu_D = 0$ (Paired data)

Case Study 14.3.3

```
% LMcs140303_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% page 821. A case study solved by the sign and Wilcoxon signed rank
% test
% Written by Eugene.Gallagher@umb.edu 11/16/10 Revised 3/1/11
%
D=[4.67 4.36;3.5 3.64;3.5 4;3.88 3.26;3.94 4.06;4.88 4.58;4 3.52
   4.4 3.66;4.41 4.43;4.11 4.28;3.45 4.25;4.29 4;4.25 5;4.18 3.85
   4.65 4.18];
expected=0;
[P,H,STATS] = signrank(D(:,1),D(:,2),'alpha',0.05,'method','exact');
fprintf(...
'\nThe signed rank test statistic is %4.1f with exact 2-tailed p=%6.4g\n',...
 STATS.signedrank,P);
[P,H,STATS] = signrank(D(:,1),D(:,2),'alpha',0.05,'method','approximate');
fprintf('The signed rank test approximate 2-tailed p=%6.4g for z=%6.4f\n',P,STATS.zval);
[H,P,CI,STATS] = ttest(D(:,1),D(:,2),0.05,'both');
fprintf('The paired t test 2-tailed p=%6.4g\n',P);
fprintf(...
'The mean difference in ratings (In-Class - Online) = %4.2f with 95%% CI: [%4.2f %4.2f]\n',...
   mean(D(:,1)-D(:,2)), CI(1),CI(2));
[p,h,stats] = signtest(D(:,1),D(:,2),0.05,'method','exact');
fprintf('The sign test statistic is %4.1f with exact 2-tailed p=%6.4g\n',stats.sign,p);
[p,h,stats] = signtest(D(:,1),D(:,2),'method','approximate');
fprintf('The sign test z=%5.3f with approximate 2-tailed p=%6.4g\n',stats.zval,p);

% Plot histogram and check for symmetry
binsize=.2;  % Needed in order to properly scale the normal pdf
edges=-.8:binsize:.8;
% hist(D);
[N,BIN] = histc(D(:,1)-D(:,2),edges);
bar(edges,N,'histc')
axis([-.82 .82 0 max(N)+0.2])
set(get(gca,'Children'),'FaceColor',[.8 .8 1]);
xlabel('Difference in evaluations, In-Class - Online','FontSize',20);
ylabel('Number of Cases','FontSize',20);
ax1=gca;
set(ax1,'Ytick',[0:max(N)],'Xtick',edges)
figure(gcf);pause
% Superimpose the normal probability pdf on a histogram of differences.
% The normal probability equation is provided on p. 293
% This is for mean 0, and unit standard
```

```
% deviation. The more general equation (Legendre & Legendre, 1998 p. 147) is:
% f(y_j)=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)/sigma_j)^2);
n=length(D(:,1)-D(:,2));
mu_j=mean(D(:,1)-D(:,2));
sigma_j=std(D(:,1)-D(:,2)); % sigmaj is the standard deviation; = 1 after Z transform
y_j=-.82:0.01:.82;
fy_j=1/(sqrt(2*pi)*sigma_j)*exp(-1/2*((y_j-mu_j)./sigma_j).^2);
fy_j=n*binsize*fy_j;
cutoff=expected;
fy_cutoff=n*binsize/(sqrt(2*pi)*sigma_j)*exp(-1/2*((cutoff-mu_j)./sigma_j).^2);
% will properly scale the height of the pdf
% fyj=1/(sqrt(2*pi)*sigmaj)*exp(-1/2*((y-muj)/sigmaj).^2);
% Plot using ax1 handle, saved above,to save this graph
% on top of the previous graph.
v=axis;
h1=line(y_j,fy_j ,'Color','r','Parent',ax1);
set(h1,'linestyle','--','color','r','linewidth',2)
h1=line([cutoff cutoff]',[0 fy_cutoff]','Color','b','Parent',ax1);
set(h1,'linestyle','-.','color','b','linewidth',3)
h2=line([cutoff cutoff]',[fy_cutoff v(4)]','Color','b','Parent',ax1);
set(h2,'linestyle','-.','color','b','linewidth',3)
s=sprintf('Case Study 14.3.3, %2.0f samples untransformed',n);
title(s,'FontSize',22)
figure(gcf);pause
```

14.3.6 **Testing** H_o: $\mu_X = \mu_Y$ (The Wilcoxon Rank Sum Test)

Theorem 14.3.4

Case Study 14.3.4

```
% LMcs140304_4th.m
% Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th edition
% Written by Eugene.Gallagher@umb.edu; written 11/16/10; revised 11/23/10
% Calls Matlab's ranksum.m and Gallagher's Wilcoxranksum.m

AL=[177 177 165 172 172 179 163 175 166 182 177 168 179 177]';
NL=[166 154 159 168 174 174 177 167 165 161 164 161]';
boxplot([AL;NL],[ones(length(AL),1);zeros(length(NL),1)]);figure(gcf)
[P,H,STATS] = ranksum(AL,NL,'alpha',0.05,'method','exact');
fprintf(...
   '\n\nUsing Matlab''s ranksum, exact p=%6.4f, Rank sum = %4.1f\n',P,...
   STATS.ranksum)
if H==1
   fprintf('Reject Ho\n\n')
else
   fprintf('Fail to reject Ho\n\n')
end
```

```
[pvalue,W,U]=Wilcoxranksum(AL,NL,1);
fprintf('Using Gallagher''s Wilcoxranksum, exact p=%6.4f;\n', P)
fprintf('Wilcoxon''s W = %4.1f; Mann-Whitney U=%4.1f;\n',W,U)
[P,H,STATS] = ranksum(AL,NL,'alpha',0.05,'method','approximate');
fprintf('\nUsing Matlab''s ranksum, large sample p=%6.4f;\n',P)
fprintf('Rank sum = %4.1f; z-value=%5.2f\n',STATS.ranksum,STATS.zval)
if H==1
   fprintf('Reject Ho\n\n')
else
   fprintf('Fail to reject Ho\n\n')
end
[pvalue,W,U,Wstar]=Wilcoxranksum(AL,NL,0);
fprintf('Using Gallagher''s Wilcoxranksum, large sample p=%6.4f;\n',P)
fprintf('Wilcoxon''s W = %4.1f; Mann-Whitney U=%4.1f; z-value=%5.2f\n',...
   W,U,Wstar)
```

```
function [pvalue,W,U,Wstar]=Wilcoxranksum(X,Y,Ex)
% Wilcoxon rank-sum test
% [pvalue,W,U,Wstar]=Wilcoxranksum(X,Y,Ex)
% Tests the null hypothesis that X & Y have the same pdf.
% Input: X,Y two samples,Ex~=0 indicates do an exact test.
% Output: pvalue: pvalue, 2-sided p value for large sample approximation N(0,1) distribution
%        W=Wilcoxon rank sum statistic
%        U=Mann-Whitney U statistic
%        Wstar=z value for asymptotic large sample approximation
% Calls Wilcoxrsexact
% Written by Eugene.Gallagher@umb.edu
% Revised 11/14/10

X=X(:);Y=Y(:);
n=length(X);
m=length(Y);
% Rank the X&Y values from smallest to largest, assigning average ranks to ties.
[T,R,ind]=ties([X;Y]);T=T';  % calls Gallagher's ties.m
% Find sum of ranks of the smaller sample;
if n<m;
   W=sum(R(1:n));
else
   W=sum(R(n+1:n+m));
   n=m;  % Expected value & variance equastions assume n is the size of the smaller group.
   m=length(X);
end
U=W-n*(n+1)/2; % Mann-Whitney U statistic
largesample=logical(1);
if nargin>2
   if Ex~=0
```

```
        largesample=logical(0);
    end
end
if nargin>2 & ~largesample
  ncomb=nchoosek(n+m,n);
  if ncomb>1e6
      t=sprintf(...
          '%d combinations, T=%d min (1e6 combs take 1 min on p4)\n',...
          ncomb,round(ncomb/1e6));
      toomany=menu(t,'Stop','Continue');
      if toomany==1
         largesample=logical(1);fprintf('Large sample approximation for 2-tailed p\n');
      end
  end
  if ~largesample
      pexuptail=wilcoxrsexact(n,m,W,R);
      if pexuptail<=0.5
          pvalue=2*pexuptail;
      else
          pvalue=2*(1-pexuptail);
      end
  end
end
if largesample
    % Large sample approximation;% Hollander & Wolfe p. 108
    EoW=(n*(m+n+1))/2;
    % Calculate the variance of W, without ties and with ties.
    if isempty(T)  % Size of tied groups from ties.m
        VaroW=(m*n*(m+n+1))/12;
    else
        VaroW=(m*n)/12*(m+n+1-(sum((T-1).*T.*(T+1)))/((m+n)*(m+n-1)));
    end
    Wstar=(W-(n*(m+n+1)/2))/sqrt(VaroW);  % Without ties, tends to an asymptotic N(0,1)
distribution.
    % Find the 2-tailedprobability of Wstar from the standard normal distributioin
    pvalue=erfc(abs(Wstar)/sqrt(2));
    % Note that the exact p values are tabulated, and an exact test, even in the presence of ties
    % can be performed, see pp. 113-116 in Hollander & Wolfe.
end
```

```
function pexuptail=Wilcoxrsexact(n,m,W,ranks);
% Exact upper tail p values for Wilcoxon Rank Sum statistic
% function pexuptail=Wilcoxrsexact(n,m,W,ranks);
% Borrows shamelessly from Strauss's combvals.m
% Note that Matlab's nchoosek will also generate the list
% of combinations. This program doesn't generate the full
```

```
% matrix of combinations, but calculates the test stat only.
% Input: n size of smaller group
%        m size of larger group
%        W Wilcoxon signed rank statistic
%        ranks, actual ranks of n+m items if there are ties present.
% Written by E. Gallagher, Eugene.Gallagher@umb.edu
% Help file for Strauss' combvals:
% COMBVALS: Generates the combinations of n integers taken r at a time.  The
%           number of such combinations is given by function nc=combin().
%     Usage: c = combvals(n,r)
%           n = number of integers (1:n) to be combined.
%           r = number to be taken at a time (0 < r <= n).
%           -------------------------------------------------------
%           c = [nc x r] matrix of combinations.

% Based on ACM Algorithm 94, J. Kurtzberg, Comm. ACM, June 1962.
% RE Strauss, 12/18/98

% An exact conditional distribution with ties follows Hollander & Wolfe p. 115
if nargin<4
   ranks=1:n+m;
   notiedr=logical(1);
else
   if length(ranks)<n+m
      error(...
         sprintf(...
         'Number of ranks (%d) doesn''t match n+m (%d)\n',...
         length(ranks),n+m));
   end
   ranks=sort(ranks);
   notiedr=logical(0);  % could do a check to see if there really are ties with ties.m
end
ranks=ranks(:);
fudranks=flipud(ranks);
N=n+m;
r = n;
ncomb = nchoosek(N,r);  % Matlab's built-in combination function.
if W>=n*(n+m+1)-W;
   uppertail=logical(1);
else
   W=n*(n+m+1)-W;
   uppertail=logical(0);
end
if W>sum(fudranks(1:n))
   if uppertail
```

```
        error('W impossibly large')
     else
        error('W impossibly small')
     end
elseif W==sum(fudranks(1:n)) & notiedr
     if uppertail
        pexuptail=1/ncomb;
     else
        pexuptail=(ncomb-1)/ncomb;
     end
     return
end
  % Strauss's combval lists combinations in c in lexicographic
  % order, thus the critical values for sum(C) are larger than
  % observed W. We can speed up the process by using
  % Wstar=min(W,n*(m+n+1)-W) and exiting loop when Wstar fails
  % to be less than critical value
  if ncomb>1e6
     t=sprintf(...
        '%d combinations, T=%d min (1e6 combs take 1 min on p4)\n',...
        ncomb,round(ncomb/1e6));
     toomany=menu(t,'Stop','Continue');
     if toomany==1
        return
     end
  end
  % c = zeros(ncomb,r);     % Don't need to store values.
  Tally=0;
  j = zeros(1,r);

  for i = 1:ncomb
    b = 1;
    endflag = 0;
    while(~endflag)
      if (j(b)>=b)
        a = j(b)-b-1;
        for l = 1:b
          j(l) = l+a;
        end;
        endflag = 1;
      else
        if (b==r)
          for b = 1:r
            j(b) = N-r-1+b;
          end;
```

```
          endflag = 1;
        end;
        b = b+1;
      end;
    end;
    % c(i,:) = N-j(r:-1:1);
    c=N-j(r:-1:1);
    if sum(ranks(c))>=W
        Tally=Tally+1;
    end
  end;
  pexuptail=Tally/ncomb;
  if ~uppertail
      pexuptail=1-pexuptail;
  end
```

```
function [T,R,ind]=ties(A)
% format: [T,R,ind]=ties(A)
% a function to return a row vector of tied groups, T,
% Ranks R (including average ranks) and indices of tied elements
% needed to calculate variance of S using Kendall's
% variance formula & Spearman's r.
% input: A is a row or column vector
% T: a row vector containing number of members of tied groups
% T=0 if there are no tied groups
% sum(T) is equal to the number of tied elements.
% each element of T equals the number in each tied group
% tied groups are sorted in ascending order.
% Examples: A=[1 2 3];[T,R,ind]=ties(A)=>   T=0,R=[1 2 3],ind=[]
%          A=[1 2 3 1];          T=2,R=[1.5 3 4 1.5],ind=[1 4]
%          A=[2 1 2 3 1 2];      T=[2 3],R=[4 1.5 4 6 1.5 4],
%                                ind=[5 2 3 1 6]
%          A=[2 1 2 3 3 1 2];    T=[2 3 2],R=[4 1.5 4 6.5 6.5 1.5 4]
%                                ind=[6 2 3 1 7 4 5]
% R (Row vec)=numerical rankings of A with ave. ranks for ties
% ind: indices of tied elements, sorted by rank; sorted tied elements=A(ind);
% ties.m is used in Kendall.m as T=ties(A), and Spear.m
% written by E. Gallagher, Environmental Sciences Program
% UMASS/Boston, Email: Eugene.Gallagher@umb.edu
% written: 6/16/93, revised 6/17/93
[r,c]=size(A);
if r>c
    A=A';                   % change to row vector
end
[Asort,k]=sort(A);
iota=1:length(A);iota=iota';
```

```
R(k)=iota;
index=[k' iota];
ind=[];
CDA=[~diff(Asort) 0];
min1=min(find(CDA==1));
if isempty(min1)
   T=0;
   return
end
i=0;
[rw,cl]=size(CDA);
T=zeros(size(rw,cl));
while ~isempty(min1)
   min0=min(find(CDA==0));
   if min0<min1
      CDA(min0:min1-1)=[];
      index(min0:min1-1,:)=[];
   else
    i=i+1;
    T(i)=min0-min1+1;
    CDA(min1:min0)=[];
    ind=[ind index(min1:min0,1)'];
    R(1,index(min1:min0))=ones(1,T(i))*sum(index(min1:min0,2))/T(i);
    index(min1:min0,:)=[];
   end
   min1=min(find(CDA==1));
end
T(find(T==0))=[];
```

Questions p 825-826

14.4 The **KRUSKAL-WALLIS TEST**

Theorem 14.4.1

Case Study 14.4.1 Draft lottery

```
% LMcs140401_4th.m
% Case Study 14.4.1
% 1969 draft lottery
% From Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th ed
% Written by Eugene.Gallagher@umb.edu 12/7/2010
% Are the data random?
DATA=[1 305 086 108 032 330 249 093 111 225 359 019 129
2 159 144 029 271 298 228 350 045 161 125 034 328
3 251 297 267 083 040 301 115 261 049 244 348 157
4 215 210 275 081 276 020 279 145 232 202 266 165
5 101 214 293 269 364 028 188 054 082 024 310 056
6 224 347 139 253 155 110 327 114 006 087 076 010
```

```
7 306 091 122 147 035 085 050 168 008 234 051 012
8 199 181 213 312 321 366 013 048 184 283 097 105
9 194 338 317 219 197 335 277 106 263 342 080 043
10 325 216 323 218 065 206 284 021 071 220 282 041
11 329 150 136 014 037 134 248 324 158 237 046 039
12 221 068 300 346 133 272 015 142 242 072 066 314
13 318 152 259 124 295 069 042 307 175 138 126 163
14 238 004 354 231 178 356 331 198 001 294 127 026
15 017 089 169 273 130 180 322 102 113 171 131 320
16 121 212 166 148 055 274 120 044 207 254 107 096
17 235 189 033 260 112 073 098 154 255 288 143 304
18 140 292 332 090 278 341 190 141 246 005 146 128
19 058 025 200 336 075 104 227 311 177 241 203 240
20 280 302 239 345 183 360 187 344 063 192 185 135
21 186 363 334 062 250 060 027 291 204 243 156 070
22 337 290 265 316 326 247 153 339 160 117 009 053
23 118 057 256 252 319 109 172 116 119 201 182 162
24 059 236 258 002 031 358 023 036 195 196 230 095
25 052 179 343 351 361 137 067 286 149 176 132 084
26 092 365 170 340 357 022 303 245 018 007 309 173
27 355 205 268 074 296 064 289 352 233 264 047 078
28 077 299 223 262 308 222 088 167 257 094 281 123
29 349 285 362 191 226 353 270 061 151 229 099 016
30 164 NaN 217 208 103 209 287 333 315 038 174 003
31 211 NaN 030 NaN 313 NaN 193 011 NaN 079 NaN 100];
DATA=DATA(:,2:13);
y=DATA(:); % convert the data into columns; drop the NaN elements
group=repmat(1:12,31,1);group=group(:);i=~isnan(y);y=y(i);group=group(i);
[p,table,stats] = kruskalwallis(y,group)
multcompare(stats)
% As described on page 829, test the 1st vs. 2nd 6 months.
g=group;g(group<=6)=1;g(group>6)=2;
[p2,table2,stats2] = kruskalwallis(y,g)
```

Questions p 830-832

14.5 THE FRIEDMAN TEST

Theorem 14.5.1

Case Study 14.5.1

```
% LMcs140501_4th.m
% Case Study 14.5.1
% Base running example from Hollander & Wolfe
% From Larsen & Marx (2006) Introduction to Mathematical Statistics, 4th ed
% Written by Eugene.Gallagher@umb.edu 12/7/2010
%
DATA=[5.5 5.55
   5.7 5.75
```

```
   5.6 5.5
   5.5 5.4
   5.85 5.7
   5.55 5.6
   5.4 5.35
   5.5 5.35
   5.15 5
   5.8 5.7
   5.2 5.1
   5.55 5.45
   5.35 5.45
   5 4.95
   5.5 5.4
   5.55 5.5
   5.55 5.35
   5.5 5.55
   5.45 5.25
   5.6 5.4
   5.65 5.55
   6.3 6.25];
plot(DATA');
ax1=gca;
set(ax1,'Xtick',[1 2])
set(ax1,'XtickLabel',{'Narrow-Angle','Wide-Angle'})

figure(gcf);pause
[P,TABLE,STATS]=friedman(DATA);
```

14.6 **TESTING FOR RANDOMNESS**

Case Study 14.6.1

```
% LMcs140601_4th.m
% Uses the resampling toolbox function runs.m
DATA=...
[61 53 58 51 52 34 45 52 46 52 37 39 50 38 55 59 57 64 73 46 48 47 40 35 40]';
n=length(DATA);
[H,P,STATS]=runstest(diff(DATA)>0); % This is not the same runs test a
                                     % Larsen and Marx. Matlab's runs test
                                     % considers the number of positive and
                                     % negative runs, but L&M's test just
                                     % considers the total N (25) in
                                     % calculating its test statistic. Thus,
                                     % L&M's test assumes no trend.
% Theorem 14.6.1:
EW=(2*n-1)/3;
VarW=(16*n-29)/90;
```

```
Z=(STATS.nruns-EW)/sqrt(VarW)
if Z>0
   p=1-normcdf(Z);
else
   p=normcdf(Z);
end
fprintf(...
 'With Matlab''s runs test, P(%2.0f runs with %2.0f cases) is %5.3f\n',...
 STATS.nruns,n,P)
fprintf(...
 'With Larsen & Marx''s runs test P(%2.0f runs with %2.0f cases) = %5.3f\n',...
 STATS.nruns,n,p)

% Although undocumented, Matlab is probably using the Wald-Wolfowitz runs
% test; When I can get access to my stats books with the exact version
% of the test, I'll check.
```

Questions p. 838-841

14.7 Taking a second look at statistics (comparing parametric and nonparametric procedures

Appendix 14.A.1 Minitab applications

References

Larsen, R. J. and M. L. Marx. 2006. An introduction to mathematical statistics and its applications, 4th edition. Prentice Hall, Upper Saddle River, NJ. 920 pp. {?}

Index