Non-Parametric Methods: Sections 29 - 38

How to Test a Hypothesis about a Proportion or Comparing Two Proportions

How to Do a One Sample Mann-Whitney Wilcoxon Test

How to Run a Mann-Whitney-Wilcoxon Test for Two Independent Samples

How to Run a Repeated Measures Mann-Whitney-Wilcoxon Test

How to Run a Kruskal-Wallis Test

How to Run Friedman's ANOVA

How to Create a Contingency Table and Run a Chi-Square Test

How to Test for Normality: Beyond Graphical Methods

How to Check Pairs of Values for Correlation Non-parametrically

How to Run a Binary Logistic Regression

Section 29: How to Test a Hypothesis about a Proportion or Comparing Two Proportions

(Uses no data files)

Sometimes you need to test a hypothesis about proportions. This can be a test concerning the proportion of one population that has a certain property, or it can be a test to compare the proportions of two populations that have a certain property. For the single sample case, use:

```
> binom.test (number in sample with property, sample size, hypothesized proportion, hypothesis type)
```

OR

```
> prop.test (number in sample with property, sample size, hypothesized proportion, hypothesis type)
```

In the two proportion case, there are some additional commands that must be used first to set up the test. There is also a slight modification to the command itself in that you do not specify any hypothesized proportion. (See EXAMPLE B below.)

EXAMPLE A (Single proportion): Suppose you want to test the hypothesis that a particular biased coin will come up with heads more than 65% of the time. Set up your hypotheses. Assume that you have flipped the coin 50 times and gotten 35 heads in your sample.

H_0: prop ≤ .65
H_1: prop > .65.
Set level of significance; this example uses alpha (α) = .05.

You have a choice to make. You can use either the binomial test or the proportion test. The binomial test is exact, but often not included in textbooks because the computations are more involved than those in the approximate proportion test. With software, this is not an issue. Here is how to do it both ways.

```
> binom.test (35, 50, 0.65, alternative = ”greater”)
```

OR

```
> prop.test (35, 50, 0.65, alternative = ”greater”)
```

The output from the binomial test is as shown below, with comments on the right.

```
Exact binomial test

data:  35 and 50
number of successes = 35, number of trials = 50, p-value = 0.2801      ← p-value is greater than α.
alternative hypothesis: true probability of success is greater than 0.65   ← Indicates upper tail test.
95 percent confidence interval:                ← You can be 95% confident that the actual proportion of
 0.576267 1.000000                             heads produced by this coin is at least 57.6%.
sample estimates:
probability of success                         ← Calculated sample percentage of heads is 70%.
                0.7
```

Note: For the other method, the (approximate) proportion test, most textbooks use a normal approximation with a correction factor. This is because the binomial is a discrete distribution and the normal is a continuous distribution. R uses a modification of the latter, in which it squares the test statistic that is used with the normal approximation. The square of a normal random variable has a chi-square distribution with one degree of freedom. While you really don't need to know all this to run the test, it explains why there is a "df" value in the output and the test statistic is a square.

The resulting output from the proportion test is on the left as follows; explanatory comments are on the right.

1-sample proportions test with continuity correction	
data: 35 out of 50, null probability 0.65	← "Null probability" is the 65% as in H_0: prop ≤ .65.
X-squared = 0.3516, df = 1, p-value = 0.2766	← X-squared is the test statistic. p-value is greater than α.
alternative hypothesis: true p is greater than 0.65	← Indicates upper tail test.
95 percent confidence interval: percentage of 0.5750075 1.0000000	← You can be 95% confident that the actual proportion of heads produced by this coin is at least 57.5%.
sample estimates: p 0.7	← Calculated sample percentage of heads is 70%.

INTERPRETATION: Whichever, method you choose, the p-value is greater than α. That means there is not enough evidence to reject the null hypothesis. That is, there is insufficient evidence to conclude that the coin comes up heads more than 65% of the time.

EXAMPLE B (Compare two proportions): Suppose you have two coins, and you believe the first coin is more biased than the second one. You want to test hypotheses to compare their proportions of heads. Set up your hypotheses.

H_0: proportion (coin 1) <= proportion (coin 2)
H_1: proportion (coin1) > proportion (coin2)
Pre-set your level of significance; this example uses alpha (α) = .05.

For the sample, you flip each coin 40 times. The first coin comes up with heads 28 times. The second coin comes up with heads 26 times.

First, some preliminaries. You have to create two vectors, one for the numbers of heads and one for the sample sizes. In R, the "c" command puts the values you supply into a single vector. Be sure to keep the entries in the same order: c (first coin, second coin).

```
> heads = c (28, 26)
> samplesize = c (40, 40)
```

Here you use a proportion test. You use the vectors instead of single values in the command where the numbers of items with the property of interest (in this case, heads) and the sample size (here, number of flips) are specified. You do not have to specify the hypothesized difference as zero, since that is the default. You do have to indicate the form of the alternative hypothesis, however, so that R runs an upper tail test. So you use:

```
> prop.test (heads, samplesize, alternative = ”greater”)
```

The output for the proportion test is shown below.

```
2-sample test for equality of proportions with continuity correction

data:  heads out of samplesize

X-squared = 0.057, df = 1, p-value = 0.4057   ← X-squared is the test statistic;  p-value is greater than α

alternative hypothesis: greater               ← Indicates upper tail test.

95 percent confidence interval:               ← You can be 95% sure that the proportions of heads
-0.1470229     1.0000000                        for the two coins differ by between -0.15 and 1
                                                (not a very useful confidence interval in this example).

sample estimates:
prop 1     prop 2                             ← Calculated sample proportions.
 0.70       0.65
```

INTERPRETATION: The p-value is greater than α, so you fail to reject the null hypothesis. There is insufficient evidence to indicate that coin 1 is more biased than coin 2.

Section 30: How to Run a One-Sample Mann-Whitney-Wilcoxon Test To Test a Single Population Median

(Uses data file: MO Life Expectancy.txt)

This example will test the hypothesis that the median overall U.S. life expectancy is 76 years. You will use the Missouri life expectancy by county as your sample. This may not be random enough to provide a representative sample for the whole country, but proceed as if it is for this example.

First read in the data, attach it and display it if you wish.

```
> Data = read.table ("E:/Data Files/MO Life Expectancy.txt", header = TRUE)
> attach (Data)
> Data
```

Here is a portion of the data set.

	County	Total	Male	Female
1	Adair	77.7	75.4	79.9
2	Andrew	77.8	75.1	80.3
3	Atchison	78.3	75.5	81.2
:				
114	Worth	78.7	75.3	81.8
115	Wright	75.5	72.7	78.2

For purposes of this example, the hypothesis are:

H_0: Median is 76 years.
H_1: Median is not 76 years.
Pre-select your level of significance: alpha (α) = .05

Run a Mann-Whitney-Wilcoxon test. The syntax for this in R is "wilcox.test" – the other two names are dropped. Specify the variable name and the hypothesized median. The variable you need from this data set is the column called Total and the hypothesized value is 76. The last item in the command, where it says "correct=FALSE," tells R not to bother with a continuity correction factor because the data (life span totals combining men and women) is already continuous data.

```
> wilcox.test (Total, mu = 76, correct = FALSE)
```

The resulting output is as shown at left. Explanations are on the right.

```
Wilcoxon signed rank test

data:  Total

V = 4586, p-value = 3.638e-05        ← Small p-value in scientific notation, meaning .00003638

alternative hypothesis: true location is not equal to 76
```

INTERPRETATION: Based on the p-value, you would reject the null hypothesis. The data suggests that the median is not 76 years.

Section 31: How to Run a Mann-Whitney-Wilcoxon Test for Two Independent Samples

Compare the Medians of Two Independent Populations – Unpaired Data

(Uses data file: Life Exp by Gender - nonpaired.txt)

Sometimes you may want to compare the medians of two populations, using two unpaired samples. The samples may or may not be the same size. This is similar to the t-test to compare the means of two populations using two unpaired samples, although it requires fewer assumptions. Since it is similar, the same data set is used here in the example.

```
>Data = read.table ("E:/Data Files/Life Exp by Gender - nonpaired.txt", header = TRUE)
>attach (Data)
> Data
```

A portion of the data set is shown below.

```
  Gender  Years

1      M    75.4
2      M    75.1
3      M    75.5
:
38     F    81.3
39     F    80.3
40     F    79.6
```

While it is not part of the test, you may want to split the data into two separate lists. In the full data set, the first 20 entries are for males (M) and the remaining entries are for females (F). You can split the data into two lists, and then display both lists, as follows.

```
> M.Years = Years [1:20]
> F.Years = Years [21:40]
>  M.Years; F.Years
```

This gives you the following two lists, the first for males and the second for females.

```
 75.4 75.1 75.5 73.3 73.1 75.5 74.4 73.4 73.3 77.5 73.6 71.1 73.0 74.6 75.8 75.9 73.6 69.5 75.9 72.6

 79.6 81.4 78.5 80.7 78.9 80.5 79.4 78.4 78.8 79.6 78.8 78.2 78.3 80.4 74.9 79.3 79.0 81.3 80.3 79.6
```

You can now find their sample medians as shown below.

```
> median (M.Years)
```

R returns the following as the sample median for males.

```
74
```

```
> median (F.Years)
```

R returns the following as the sample median for females.

```
79.35
```

There is about a five-year difference between the median male and female life expectancies in the sample. To test whether or not there is a difference in the medians of male and female populations, proceed as follows. Assume you want a two-tailed test with level of significance = .05

H_0: Median male life expectancy = Median female life expectancy.
H_1: Median male and female life expectancies are different.
Set alpha (α) = .05

```
> wilcox.test (Years ~ Gender,mu=0,alternative="two.sided",paired=FALSE,correct=FALSE,conf.int=TRUE)
```

Note that you tell R to test mu=0; that is because, if the medians are equal, then the hypothesized difference should be zero. Also, since the data is unpaired, you specify that "paired=FALSE". You also specify not to use the correction factor ("correct = FALSE") because the variable Years is continuous. Finally, "conf.int=TRUE" tells R to find a confidence interval for the difference in medians.

The following output results. Actual output is on the left; explanations are on the right.

```
Wilcoxon rank sum test

data:  Years by Gender

W = 392, p-value = 2.038e-07          ← Small p-value in scientific notation; means .0000002038.
                                        This p-value is less than α.

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

 4.100038  6.000015   ← Thus you can be 95% confident the actual difference in population medians is
                         between about 4.1 and 6.0.

sample estimates:
difference in location
            5.200001   ← This is a calculated difference that, in theory, should match the difference in
                         sample medians (79.35-74 = 5.35 from above). It does not quite match
                         because the test works with a quantity called the "pseudomedian." If the
                         distribution is symmetric, it will match; otherwise, it may not.

Warning messages:
1: In wilcox.test.default(x = c(79.6, 81.4, 78.5, 80.7, 78.9, 80.5,  :   ← There were ties in the data, that is,
  cannot compute exact p-value with ties                                    at least one of the data values
2: In wilcox.test.default(x = c(79.6, 81.4, 78.5, 80.7, 78.9, 80.5,  :     occurred more than once.
cannot compute exact confidence intervals with ties                         So an approximation was used.
```

Note: If your data had originally been in two columns of equal length, with headers “M” and “F”, you could have run the test using the following syntax.

```
> wilcox.test (M, F, alternative = "two.sided", paired = FALSE, correct = FALSE, conf.int=TRUE)
```

However, this would not have worked if you had more data in one column than the other, because the original “read.table” command would have failed.

Section 32: How to Run a Repeated Measures Mann-Whitney-Wilcoxon Test Comparing Two Population Medians Using Paired Data

(Uses data file: Life Exp by Gender - paired.txt)

This is an example of a Mann-Whitney-Wilcoxon test to compare the medians of paired data. In this particular example, the data consists of Missouri life expectancies for men and women, paired by county. A sample of n=24 counties is used.

The null hypothesis is that the median life expectancy is the same for both genders; the alternative is that it is not the same. Set up your test.

H_0: The median life expectancies of males and females are equal.
H_1: The median life expectancies are not equal.
Preset your level of significance; this example uses alpha (α) = .05.

```
> Data = read.table ("E:/Data Files/Life Exp by Gender - paired.txt", header = TRUE)
> attach (Data)
> Data
```

A portion of the data set is shown below. The "Difference" column is not relevant for this example and should be ignored.

	County	Male	Female	Difference
1	Adair	75.4	79.9	4.5
2	Andrew	75.1	80.3	5.2
3	Cole	76.1	80.5	4.4
:				
23	Worth	75.3	81.8	6.4
24	Wright	72.7	78.2	5.5

Now run a "paired Wilcoxon" test as follows. Note that you set "paired=TRUE" because the data is paired by county. Also, since ages are continuous data, you do not need a correction factor. Therefore you set "correct = FALSE").

```
wilcox.test (Male, Female, alternative = "two.sided", paired = TRUE, correct = FALSE)
```

The resulting output is shown below. Output is on the left; explanatory comments are on the right.

```
Wilcoxon signed rank test

data:  Male and Female

V = 0, p-value = 1.804e-05          ← Small p-value in scientific notation; meaning .00001804

alternative hypothesis: true location shift is not equal to 0
```

Since the p-value is less than α, reject the hypothesis of equal medians.

Sometimes there is additional output when you run this test. In this example, your output also shows the following message at the end.

```
Warning message:
In wilcox.test.default (Male, Female, alternative = "two.sided",  :
cannot compute exact p-value with ties
```

This is no cause for alarm. R is telling you that it used an approximation because there were ties, that is, there were some duplicate values in the data set.

Section 33: How to Run a Kruskal-Wallis Test
Compare Medians of Multiple Populations
(Uses data file: Anxiety.txt)

You use the Kruskal-Wallis test when you want to compare the medians of more than two populations and the populations are defined by a single characteristic. It is the non-parametric analogue of one-way ANOVA, and R calls it "kruskal.test."

This example uses a fictional data set of "anxiety scores" of 156 college students in four types of institutions (coded as LPR = large private university, SPR = small private university, STA = state university and COM = community college). The goal is to determine whether or not the median score is the same or different by type of institution. The null hypothesis is that all four types have the same median anxiety score; the alternative is that at least one median is different. Use level of significance alpha (α) =.05.

Set up your test formally.

H_0: The median scores for all types of institutions are equal.
H_1: At least one type has a different median score.
Preset your level of significance; this example uses alpha (α) =.05.

First read in data table giving the scores, attach it and display it if you wish.

```
> Data = read.table ("E:/Data Files/Anxiety.txt", header = TRUE)
> attach (Data)
> Data
```

A partial display of the data is as follows.

```
     Type  AnxScore

1     LPR        25
2     SPR        11
3     STA         8
4     COM        17
       :
153   SPR        33
154   SPR        30
155   COM        21
156   COM        19
```

A single line of code is sufficient to run the test, using the AnxScore as a function of Type.

```
> kruskal.test (AnxScore ~ Type)
```

The resulting output is as shown below.

Kruskal-Wallis rank sum test

data: AnxScore by Type
Kruskal-Wallis chi-squared = 2.5423, df = 3, p-value = 0.4677

The p-value= 0.4677 > α, so the null hypothesis is accepted based on this sample. This means that the sample data does not indicate a difference in population median anxiety scores for the four types of institutions.

Section 34: How to Run Friedman's ANOVA
Comparing Multiple Medians in the Case of Repeated Measures
(Uses data file: AnxietyRepeat.txt)

The example uses the data set AnxietyRepeat.txt. This data set contains scores on an Anxiety Test, repeated three times on thirty-six (fictional) students. The test is originally given during their first term in college (labelled Fall1), repeated during their second term (labelled Spr1) and then again in their third term (labelled Fall2). The supposed goal of the testing is to see whether or not, on average, students' median anxiety level is generally consistent or changes over time as they adjust to college.

First read in data table giving the anxiety scores and attach it. Display it if you wish.

```
> Data = read.table ("E:/Data Files/AnxietyRepeat.txt", header = TRUE)
> attach (Data)
> Data
```

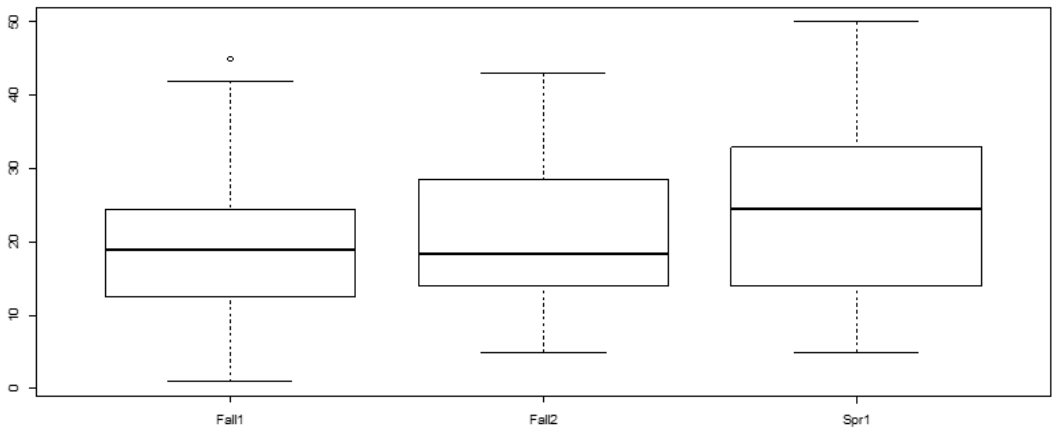
A partial display of the output is as follows.

```
     ID  Test.Session  Anx.Score
1    S1      Fall1         25
2    S1      Spr1          22
3    S1      Fall2         33
:
106  S36     Fall1         24
107  S36     Spr1          30
108  S36     Fall2         37
```

As a first look, you may want to obtain the boxplots for each test session and see how the medians compare. One line of code will get you this.

```
>  boxplot (Anx.Score ~ Test.Session)
```

The following graph results.



The boxplot indicates that the median of the Spr1 sample is different but you don't know if the same statement applies to the population medians. An appropriate hypothesis test for this purpose is Friedman's (non-parametric) ANOVA.

First set up your hypotheses; assume your chosen level of significance is alpha (α) = .05.

H_0: The medians of the anxiety scores are the same for all test sessions.
H_1: The median anxiety score for at least one test session is different.

One line of code will do this. It applies the Friedman test to the anxiety scores, treating them as a function of test session with each individual serving as his/her own "control" from one session to the next. (That is the meaning of the vertical bar followed by "ID" at the end of the line, right before the close-parenthesis.)

```
> friedman.test (Anx.Score ~ Test.Session | ID)
```

You get the following output.

```
Friedman rank sum test

data:  Anx.Score and Test.Session and ID
Friedman chi-squared = 2.3803,   df = 2,   p-value = 0.3042
```

As indicated by the p-value being greater than α, the evidence in the sample does not support rejecting the null hypothesis.

Section 35: How to Create a Contingency Table and Run a Chi-Square Test on It
(Uses no data files)

The null hypothesis in this example is that the scores on a test are independent of gender. The alternative is that there is a difference in the distribution of test results that is dependent on gender.

The results are presented in a contingency table, with the first row being men's results and the second row being women's results. The result counts are in the following order: Pass, Fail, Retest.

The data to be entered are:

	Pass	Fail	Retest
Men	42	29	10
Women	50	23	17

To create a contingency table: Enter it as a matrix, in order row by row, and specify dimensions (number of rows, not counting the labels). The example here has two rows of values, so specify that "nrow = 2." The "byrow = TRUE" subcommand will cause R to split the list of values into the number of rows you specified. That means, in this example, you will end up with your six data values in two rows and three columns, as you wanted them. The name you are giving the matrix is ExamResult.

```
> ExamResult = matrix (c (42,29,10,50,23,17), nrow = 2, byrow = TRUE)
> ExamResult
```

The resulting output is at left; explanatory comments are added at right.

```
     [,1]  [,2]  [,3]
[1,]   42    29   10
[2,]   50    23   17
```

← R-default labels are column and row numbers. As presented above, the rows are men and women. Columns are Pass, Fail, Retest.

Now set up your hypothesis test.

H_0: Test results are independent of gender.
H_1: Test results and gender are dependent.
Preselect your level of significance; this example uses alpha (α) = .05.

You are now ready to run the test. It will be a two-tail test by default.

```
> chisq.test (ExamResult)
```

Output is as follows:

```
Pearson's Chi-squared test

data:  ExamResult

X-squared = 2.7367, df = 2, p-value = 0.2545
```

If you want, you can also have R obtain the .025 and .975 quantiles of the chi-square distribution with degrees of freedom = (rows-1)*(columns-1) = (2-1)*(3-1) = 2. This process is discussed in more detail in Section 13 of this Manual.

To get the lower critical value, enter the following.

```
> qchisq (.025, 2)
```

R returns the lower critical value.

```
0.05063562
```

Now for the upper critical value.

```
> qchisq (.975, 2)
```

R returns the upper critical value.

```
7.377759
```

The p-value is larger than α, so the null hypothesis is accepted. There is insufficient evidence to believe that test results are dependent upon gender. The test statistic 2.7367 is within the interval between the critical values, and confirms your "no difference by gender" conclusion.

There are a couple of related items that may be of interest. First, you may want to know the expected values for each cell of the contingency table if the null hypothesis is true. You can calculate these by hand without much trouble, or there is a single-word command in R that will do it for you. You just have to add it to the command that you used to run the test.

Modify the command you used before to say the following; the new piece is highlighted for you.

```
> chisq.test (ExamResult)$expected
```

Now the output is not the test result, but the expected values in each cell as follows:

```
      [,1]     [,2]     [,3]
[1,] 43.57895 24.63158 12.78947  ← R-default labels are column and row numbers
[2,] 48.42105 27.36842 14.21053    Rows are men/women.
                                   Columns are Pass, Fail, Retest.
```

Also, to get a measure of strength of association between variables (gender and exam outcome), you may want to use Cramer's V. You can have R calculate this. If necessary, install the package: DescTools. Then load it. If you need a reference on how to install and load packages, see Section 3.

Then type the command below.

```
> CramerV (ExamResult)
```

For this example, R produces a value V = 0.1265066, or for practical purposes. 0.127. This is small, since Cramer's V always falls in the range between zero and one, with zero meaning "no association" and one meaning "perfect association." Therefore, in this example, there is little association between gender and exam outcome. This is what you should expect, since the hypothesis test indicated that there was no significant difference by gender.

Section 36: How to Test for Normality beyond Graphical Methods

(Uses data file: Mortgage Rates2.txt)

You can check your data set for normality using graphical methods in two ways: (1) by creating a histogram and making a judgment about whether or not it indicates that the data could reasonably come from a normal distribution, or (2) by creating a "normal Q-Q plot." These were discussed in Section 10: "How to Check for Normality with a Normal Probability Plot." This section will revisit them and then extend your options to include analytic methods.

The following example uses mortgage rate data by month for the years 2003-2017. (Data came from http://www.freddiemac.com/pmms/pmms30.html.)

First, read the data into R and attach it. Display it if you choose.

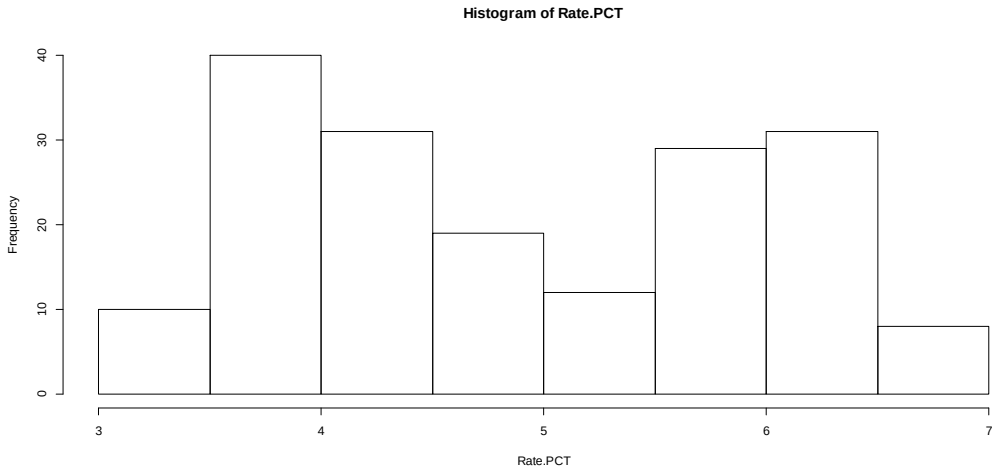
```
> Data = read.table ("E:/Data Files/Mortgage Rates2.txt", header = TRUE)
> attach (Data)
> Data
```

The beginning of data set is shown below. The example will use the column called "Rate.PCT."

```
     Time      Rate.PCT
1    Jan17     4.15
2    Feb17     4.17
3    Mar17     4.20
4    Apr17     4.05
:
:        (etc)
```

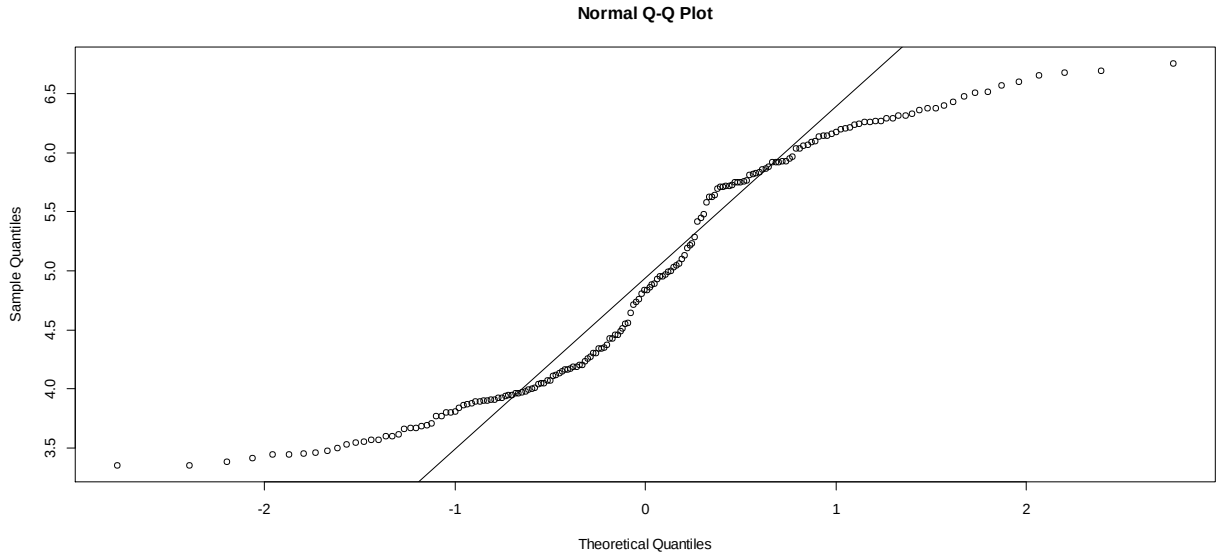
First, examine the data graphically. Make R create the histogram of the column called "Rate.PCT."

```
> hist(Rate.PCT)
```



Also, make R create the normal probability plot of the column called "Rate.PCT."

```
> qqnorm (Rate.PCT)
> qqline (Rate.PCT)
```



Both graphs indicate non-normality. The histogram appears to have two local maximum values, which would not occur in a normal distribution. The normal probability plot shows long "tails" on both ends that move farther and farther away from the line that corresponds to normally distributed data. Thus the data does not appear to come from a normally distributed population.

For a more analytical approach, you also have the option of running a hypothesis test for normality. Here is the set-up.

H_0: The sample is compatible with a normally-distributed population.
H_1: The sample is not compatible with a normally-distributed population.
Preset your level of significance; this example uses alpha (α) = .05.

There are two tests that are commonly used. The first is the Shapiro Test, which only requires you to enter the name of the variable.

```
> shapiro.test (Rate.PCT)
```

The output is equally brief.

```
Shapiro-Wilk normality test

data:  Rate.PCT
W = 0.91644, p-value = 1.313e-08          ← p-value is less than α; reject normality hypothesis.
```

Alternatively, you can use the Kolmogorov-Smirnov Test, which is called "ks.test" in R. If you use this test, you also have to enter (in order):

1. The variable name
2. The name of the hypothesized distribution (here it is pnorm; the test can be used for other types of distributions as well)
3. The parameters of the hypothesized distribution (to illustrate here, hypothesize that the mean is about 4.9 and the standard deviation is about 1.2).

Therefore the appropriate R command is:

```
> ks.test (Rate.PCT, pnorm, 4.9, 1.2)
```

You will get the following output, shown on the left. Comments are on the right.

```
    One-sample Kolmogorov-Smirnov test

data:  Rate.PCT
D = 0.10306, p-value = 0.04368          ← p-value is less than α; reject the normality hypothesis.
alternative hypothesis: two-sided
```

A further comment about the Kolmogorov-Smirnov test: if you make a minor change in the mean and standard deviation, the test may result in accepting the normality hypothesis. See the following.

```
> ks.test (Rate.PCT, pnorm, 4.91, 1.21)
```

The output is:

```
 One-sample Kolmogorov-Smirnov test

data:  Rate.PCT
D = 0.098655, p-value = 0.06017          ← p-value is greater than α; accept the normality hypothesis.
alternative hypothesis: two-sided
```

Given that all of the previous evidence supports a conclusion of non-normality, this last result suggests that the Kolmogorov-Smirnov Test is not a good choice for use on this data. The output, both times, also showed a warning message.

```
Warning message:
In ks.test,   ties should not be present for the Kolmogorov-Smirnov test.
```

If you go back and sort the values of Rate.PCT, you will find several values that are repeated. These are "ties" and may very well be the reason why the Kolmogorov-Smirnov test yields poor results here. Therefore, it is wise to run either the graphical methods or the Shapiro test as well, and not rely solely on the results of the Kolmogorov-Smirnov test.

Section 37: How to Check Pairs of Data Values for Correlation Non-Parametrically Spearman's rho and Kendall's tau

(Uses data files: HeightWeight.txt, Head Circumference.txt)

Here is an example, using the heights (in inches) and weights (in pounds) of twenty-five fictional army recruits. The data set is the same one used in Section 25 dealing with Pearson's r. The question here is whether or not Height and Weight are correlated in some fashion, but not necessarily linearly as with Pearson's r.

EXAMPLE A The first example uses a data set where a linear relationship is actually appropriate.

First read in the table of data, attach it and then display it if you wish.

```
> HWTable = read.table ("E:/Data Files/HeightWeight.txt", header = TRUE)
> attach (HWTable)
 > HWTable
```

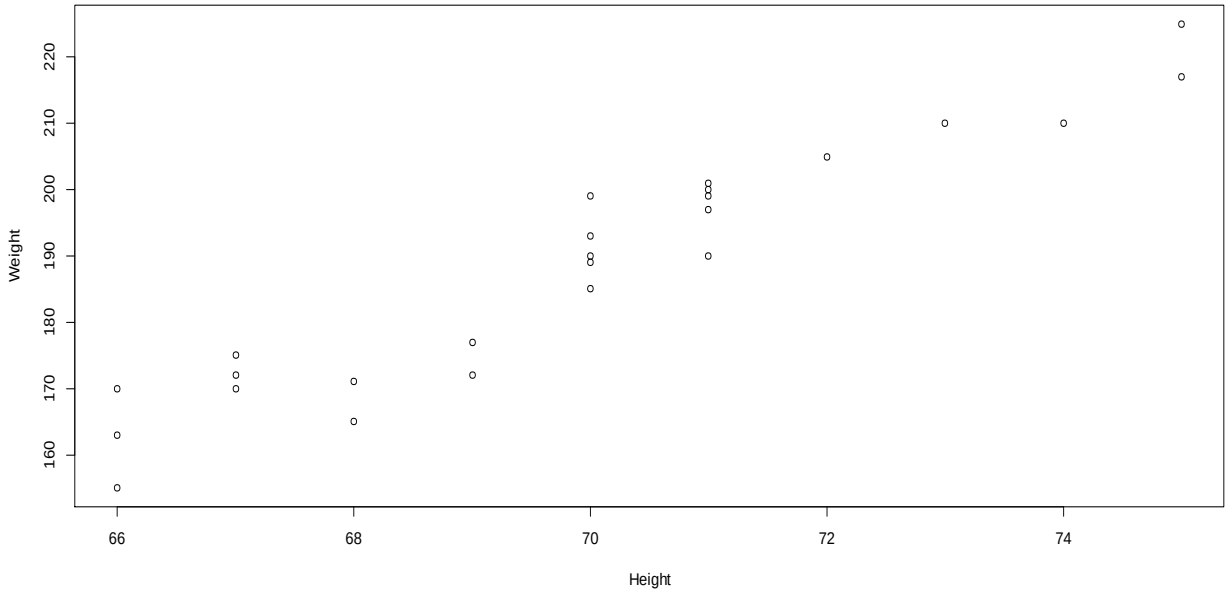
Resulting data output is the following.

```
   Height Weight
1      67    175
2      73    210
3      70    189
4      75    225
5      70    193
6      71    197
7      71    200
8      75    217
9      68    165
10     74    210
11     67    170
12     71    201
13     66    163
14     68    171
15     71    190
16     66    170
17     67    172
18     69    172
19     70    199
20     69    177
21     71    199
22     66    155
23     72    205
24     70    190
25     70    185
```

Now create the scatterplot. This is a good idea to do first, as it will help you visualize your data. The first variable you list will go on the horizontal axis; the second one will go on the vertical axis.

```
> plot (Height, Weight)
```

The resulting graph is as shown below.



This graph certainly looks as if a rising line would fit fairly well through the data set. You actually got prior confirmation for this in Section 25 where you found the linear correlation coefficient Pearson's r. That result is repeated here for reference.

```
> cor (Height, Weight)
```

R returned the following value for Pearson's r:

```
.9540717
```

This value is positive and very close to one. It supports the belief that a rising line is a good way to represent the relationship between height and weight of army recruits.

However, suppose you were only interested in whether or not height and weight generally increased together, or whether they increased according to some non-specified (and not necessarily linear) pattern. Then you might choose to use Spearman's rho or Kendall's tau.

Here is the "cor" command three ways, adapted to specify the method and the output in each case. The first one is equivalent to the one immediately above, because Pearson's r is the default when no method is specified. Thus the output will be the value you already got immediately above.

```
> cor (Height, Weight, method = "pearson")
```

Here is the command adapted for Spearman's rho, with its output.

```
> cor (Height, Weight, method = "spearman")
```

The value returned for Spearman's rho is:

```
0.9547427
```

And here it is adapted for Kendall's tau.

```
> cor (Height, Weight, method = "kendall")
```

The value returned for Kendall's tau is:

```
0.8700405
```

As you can see, all indicate strong positive relationships.

EXAMPLE B Now suppose you try it on a data set where the two variables change together, but not linearly. An example uses the data set "Head Circumference.txt", which give the 50th percentile head circumferences for male infants from birth to three years of age (0 to 36 months). The data is excerpted from: https://www.cdc.gov/growthcharts/html_charts/hcageinf.htm#males

As usual, first read in, attach and examine the data set.

```
> Data = read.table ("C:/Users/enewton/Desktop/Head Circumference.txt", header = TRUE)
> attach (Data)
> Data
```

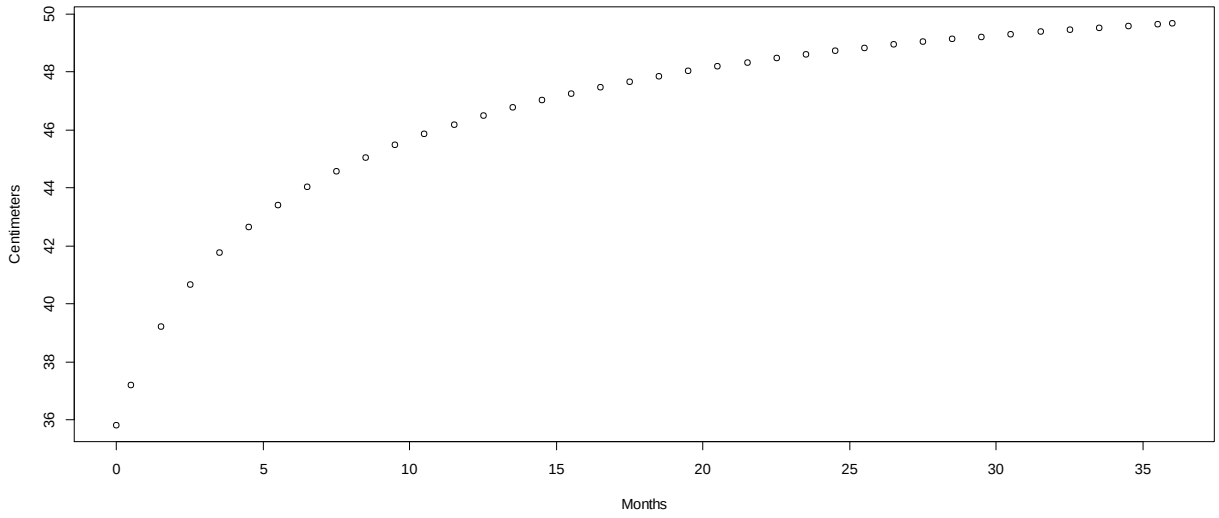
A portion of the data set is shown below.

```
   Months   Centimeters
1    0.0      35.814
2    0.5      37.194
3    1.5      39.207
4    2.5      40.652
5    3.5      41.765
:
35   33.5     49.526
36   34.5     49.592
37   35.5     49.654
38   36.0     49.684
```

You should check the scatter plot.

```
> plot (Months, Centimeters)
```

The graph is shown below; you can see that the variables increase together but not linearly.



Since the relationship is not linear, you would not choose to use Pearson's r, but the following shows all three methods for comparison purposes.

```
> cor (Months, Centimeters, method = "pearson")     ← R returns 0.8849858
```

```
> cor (Months, Centimeters, method = "spearman")  ← R returns 1
```

```
> cor (Months, Centimeters, method = "kendall")      ← R returns 1
```

This example is a little too "perfect" because the data did not come from a random sample but a known growth rate relationship. However you can see clearly that, since the increasing relationship of the two variables is "perfect" , Spearman's rho and Kendall's tau both result in a value of 1. But since the relationship is not linear, Pearson's r indicates a strong relationship but not a "perfect" linear one.

Section 38: How to Run a Binary Logistic Regression

(Uses data file: AdultFluData.txt)

This example uses a data set called AdultFluData.txt. It is a modification of the file used in the section on multiple regression. It contains information on various patients, age 20 or older, who have influenza. For convenience, it has been sorted into those without severe cough and those with severe cough. Within each group, it has also been sorted by smoking status and then by age. (The sorting was not necessary for the analysis to work.)

The variables and coding information are:

Age (in years)
Gender (0 = Female, 1 = Male)
Vaccine (0 = No flu shot, 1= Had flu shot)
Treatment (0 = Not treated, 1 = Treated)
Smoking (0 = Non-smoker, 1 = Smoker)
Temperature (body temperature in Farenheit)
Severe.Cough (0 = Cough absent or not severe)
1 = Severe cough present)

The goal is to find a linear equation that describes the odds of having a severe cough as a function of the variables that make a difference, and to omit those that do not.

First read in the data table, and attach it so that R can work with it. You probably do not want to display all of it because there are 382 lines of data.

```
> Data = read.table ("E:/Data Files/AdultFluData.txt", header = TRUE)
> attach (Data)
```

A portion of the data set is as shown. The line in the middle has been added here to help you to spot where the “no severe cough” data ends and the “severe cough” data starts; it is not part of the actual data file.

```
     Age  Gender  Vaccine  Treatment  Smoking  Temperature  Severe.Cough
1    20       0        0          1        0        101.8             0
2    22       1        1          0        0        103.3             0
3    24       0        0          0        0        101.7             0
:
60   79       1        1          1        1        100.5             0
61   91       0        1          0        0         99.9             0
     ---------------------------------------------------------------------------
62   20       0        0          0        0        102.8             1
63   20       1        0          1        0        103.7             1
64   20       0        0          0        0        103.6             1
:
381  76       0        1          1        1        101.3             1
382  81       1        1          1        1        100.7             1
```

If you have no idea which of the variables play a role in determining whether a severe cough occurs, you could create a binary logistic regression model for response Severe.Cough with all five other variables

(excluding temperature, which is more likely another response than a predictor) as predictors initially. The "summary" command displays details about the resulting model. Note the following:

1. The ~ symbol means that the variable on its left is being considered as a function of the variables on its right,
2. the code uses "glm" for "generalized linear model" (instead of the "lm" for "linear model") that is used in simple and multiple regression, and
3. since the cough is either not severe (0) or severe (1), the binomial distribution must be specified at the end (where the command says "family=binomial").

```
> CoughModel = glm (Severe.Cough ~ Age+Gender+Vaccine+Treatment+Smoking, family = binomial)
> summary (CoughModel)
```

Here is the resulting output, with explanatory comments added on the right.

```
glm(formula = Severe.Cough ~ Age + Gender + Vaccine + Treatment +  Smoking, family = binomial)

Deviance Residuals:
   Min       1Q       Median      3Q     Max
-1.8256   -1.2829   0.7657   0.9164   1.3714

Coefficients:
                 Estimate       Std. Error       z value       Pr(>|z|)
(Intercept)      2.048118       0.565917          3.619        0.000296 ***
Age             -0.034563       0.013842         -2.497        0.012527 *   ← Significant if α = .05
Gender          -0.434944       0.335940         -1.295        0.195421
Vaccine          0.312389       0.467650          0.668        0.504136
Treatment       -0.004883       0.354819         -0.014        0.989019
Smoking          0.966185       0.494432          1.954        0.050686 .   ← Significant if α = .10
                                                                              and very close at .05.
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

 Null deviance: 232.15  on 181  degrees of freedom      ← Deviance and AIC will be discussed
                                                          briefly at the end of this section.
Residual deviance: 222.65  on 176  degrees of freedom
AIC: 234.65

Number of Fisher Scoring iterations: 4
```

In the results above, the intercept also shows up as significant, but your purpose here is to determine which predictor variables are significant. As noted in the comments on the right, only Age and Smoking show up as being significant predictor variables. Therefore, you might choose to run another model, using only Age and Smoking as input variables.

```
> CoughModel2 = glm (Severe.Cough ~ Age+Smoking, family = binomial)
> summary (CoughModel2)
```

The resulting output is as follows.

```
glm(formula = Severe.Cough ~ Age + Smoking, family = binomial)

Deviance Residuals:
   Min       1Q       Median      3Q      Max
-1.7251   -1.3634     0.7816    0.9187   1.4290

Coefficients:
             Estimate     Std. Error     z value     Pr(>|z|)
(Intercept)   1.92695      0.54996       3.504       0.000459 ***
Age          -0.03474      0.01371      -2.534       0.011275 *
Smoking       0.94958      0.48707       1.950       0.051228 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 232.15  on 181  degrees of freedom
Residual deviance: 224.62  on 179  degrees of freedom
AIC: 230.62

Number of Fisher Scoring iterations: 4
```

You can see that both Age and Smoking are still significant, at approximately the same levels as before. The deviances are virtually unchanged. The AIC, which stands for Aikeke's Information Criterion, has not changed much, and in fact, has decreased slightly. Generally speaking, the smaller the AIC is, the better. The same holds for the deviances. Finally, the second model is considerably simpler. Therefore the second model is preferable to the first.

Now a word or two about interpreting and using the result. The equation that comes from the second model is:

$$\ln\left(\frac{p}{1-p}\right) = 1.93 - (0.035 \text{ Age}) + (0.950 \text{ Smoking})$$, where p = probability severe cough is present.

The right-hand side was obtained from the output, using the estimates of the intercept and coefficients of Age and Smoking. The left side is not just the response itself (Severe.Cough) as it would be in a simple or multiple regression. In binary logistic regression, the left-hand side is the natural logarithm of the odds of the response. If you want the value of p, you need to solve the logarithmic equation for p.

Here is one example. Suppose you have a new patient who is age 45 and a non-smoker. Then the equation, applied to the data for this specific new patient, becomes:

$$\ln\left(\frac{p}{1-p}\right) = 1.93 - (0.035\ \text{Age}) + (0.950\ \text{Smoking})$$
$$= 1.93 - (0.035 * 45) + (0.950 * 0)$$
$$= 0.355$$

Using the fact that the exponential function (base *e*) and the natural logarithm function are inverses, you obtain:

$$\frac{p}{1-p} = e^{0.355} = 1.426$$

Solve this algebraically for *p*: $p = 1.426 - 1.426 * p$, or equivalently $p = 0.589$. Therefore, the prediction is that there is about a 59% chance that this person will develop a severe cough.