R Primer

Jay Hooper

May 27, 2015

The intent of this document is for statistics students in an introductory statistics course to be able to use R. When first seeing R, it can seem intimidating. This document is intended to help serve as a bridge to enable students (and teachers) to use R. The exact "code" that generated each output is provided. This code can be copied and pasted and then modified. Data can be changed, titles can be changed, etc. The ability to communicate statistics and graphics electronically is an essential skill in today's world. R is used by companies such as Google, Facebook, and the New York Times. It is also used by research scientists around the world as an actual research and publication tool. How amazing would it be for our students to leave middle or high school with a working knowledge of a statistical program like R!? And it is free...

1 Basic Calculator

You can use R like a basic calculator, even defining variables. Look at the examples below. You should be able to do all calculations within R.

```
2+2
```

```
> 2+2
[1] 4
```

```
sqrt(2^2+3^2)/12
```

```
> sqrt(2^2+3^2)/12
[1] 0.3004626
```

Store a variable - the "arrow" formed by the < and - assign that value to the variable. The arrow should point toward the variable name. You can then do calculations with that variable.

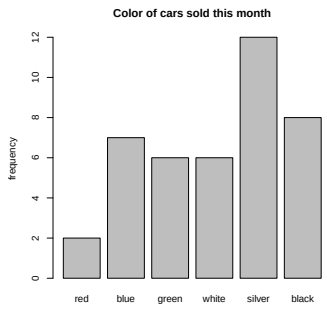
```
n <- sqrt(2^2+3^2)/12
n
12*n
```

```
> n <- sqrt(2^2+3^2)/12
> n
[1] 0.3004626
> 12*n
[1] 3.605551
```

2 Qualitative (Categorical) Data

2.1 Pareto Chart (Bar Chart)

```
counts <- c(2,7,6,6,12,8)
names(counts) <- c("red", "blue", "green", "white","silver","black")
barplot(counts, ylab = "frequency", main="Color of cars sold this month")
```

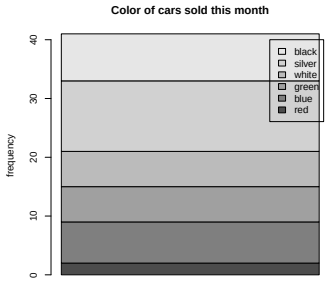


2.2 Segmented Bar Chart (Better Alternative to Pie Chart)

This one is trickier to make, but it is a great chart! It is the linear equivalent to a pie chart. The human eye is good at judging linear distances but is less good at judging areas (like in a pie chart). This chart is preferred over the pie chart.

The line that says par(xpd=T, mar=par()$mar+c(0,0,0,6)) extends the right margin to make room for a legend. The line that says "legend(locator(1)..." has the user click where the legend is to go. That is what the "locator" function does. It will wait (and not do anything else) until you click on the chart where you want the legend to go. Your legend placement will look better than the one below!

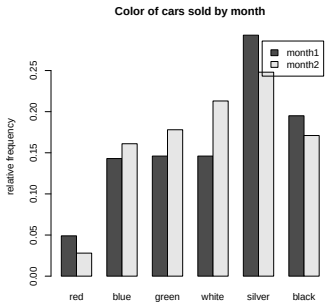
```
data <- matrix(c(2, 7, 6, 6, 12, 8), ncol=1, byrow=TRUE)
rownames(data) <- c("red", "blue", "green", "white", "silver", "black")
# Change the margins to put the legend at the right
par(xpd=T, mar=par()$mar+c(0,0,0,6))
barplot(data, ylab = "frequency", main="Color of cars sold this month",
col=grey.colors(length(data)))
legend(locator(1), title="Legend", rev(rownames(data)),
fill=rev(grey.colors(length(data))))
# Restore default margins
par(mar=c(5, 4, 4, 2) + 0.1)
```



2.3 Double Bar Graph - Comparing Two Distributions

Note that the counts (the heights of the bars) are expressed as a relative frequency. They represent percentages rather than actual counts. This chart could be made with either relative frequencies (percentages) or actual counts.

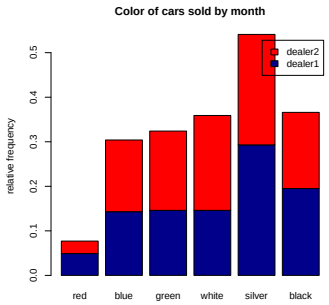
```
counts1<- c(0.049,0.143,0.146,0.146,0.293,0.195)
counts2<- c(0.028,0.161,0.178,0.213,0.248,0.171)
counts<-rbind(counts1,counts2)
colnames(counts) <- c("red", "blue", "green", "white","silver","black")
barplot(counts, ylab = "relative frequency", main="Color of cars sold by month",
legend = c("month1", "month2"), beside=TRUE)
```



2.4 Stacked Bar Graph - Comparing Two Distributions

Note that the counts (the heights of the bars) are expressed as a relative frequency again. This chart is useful for comparing two distributions expressed as relative frequencies. As mentioned before, the human eye is a good judge at linear distance and can judge how the two distributions share the relative frequencies across various categories.

```
dealer1 <- c(0.049,0.143,0.146,0.146,0.293,0.195)
dealer2 <- c(0.028,0.161,0.178,0.213,0.248,0.171)
carsales<-rbind(dealer1,dealer2)
colnames(carsales) <- c("red", "blue", "green", "white","silver","black")
barplot(carsales, ylab = "relative frequency", main="Color of cars sold by
month", col=c("darkblue","red"), legend = c("dealer1", "dealer2"))
```



3 Quantitative Data

3.1 Number Crunching (1-Var-Stats - mean, median, standard deviation, 5 number summary)

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
mean(data)
median(data)
sd(data)
quantile(data, type=2)
```

```
[1] 88.77778

[1] 93

[1] 10.9628

  0%  25%  50%  75% 100%
  59   82   93   96  100
```

3.2 Stem and Leaf Plot

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
stem(data)
```

```
The decimal point is 1 digit(s) to the right of the |

 6 | 1
 7 | 3
 8 | 01235
 9 | 2333356
10 | 0000
```

3.3 Stem and Leaf Plot - Split Stems

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
stem(data, scale=2)
```

```
The decimal point is 1 digit(s) to the right of the |

 6 | 1
 6 |
 7 | 3
 7 |
 8 | 0123
 8 | 5
 9 | 23333
 9 | 56
10 | 0000
```

3.4 Back-to-back Stemplots

Back-to-back stemplots is not a feature that comes with the "base" functions in R. It requries an additional package. You'll need to install the package "aplpack" in order to run this code. The first line of code activates this package. You will get an error if the package has not been installed.

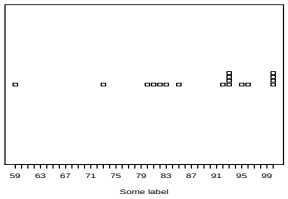
You can run this line of code to install the package: install.packages(c("aplpack"))

```
library(aplpack)
data1 <- c(394.4, 395, 394.4, 392.9, 393.8, 393.5, 393.9, 392.8, 393.4, 394.7, 394, 394.2)
data2<- c(390.7, 391.9, 393.3, 390.2, 393.1, 391.7, 390.3, 394.3, 392, 391.9, 391.6, 391.8)
stem.leaf.backback(data1,data2,m=1,depths=FALSE)
```

```
________________________
  1 | 2: represents 1.2, leaf unit: 0.1
    data1       data2
________________________
         | 390 |237
         | 391 |67899
      98 | 392 |0
    9854 | 393 |13
   74420 | 394 |3
       0 | 395 |
________________________
n:     12       12
________________________
```

3.5 Dotplot (Stripchart)

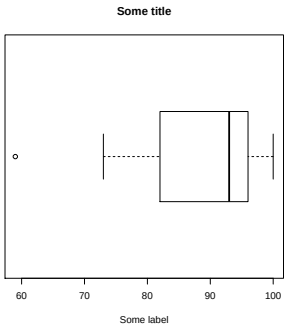
```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
stripchart(data, method="stack", offset=0.5, xlab="Some label", xaxt="n")
axis(1,seq(min(data),max(data)))
```



3.6 Box-And-Whisker Plot (with 1.5 IQR rule applied by default)

By default, when the function "boxplot" is called, R will use the 1.5 IQR rule to identify outliers. Outliers will be identified on the plot as an open circle (dot).

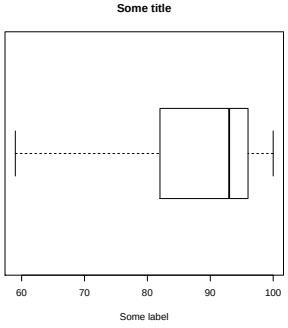
```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
boxplot(data, horizontal=TRUE, xlab="Some label", main="Some title")
```



3.7 Box-And-Whisker Plot (with no 1.5 IQR rule applied)

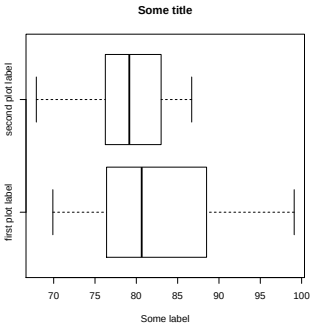
The 1.5IQR rule for identifying outliers can be "turned off" using the "range" argument. Setting "range=0" rather than the default of 1.5 causes R to extend the whiskers to the far extremes of the data.

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
boxplot(data, range=0, horizontal=TRUE, xlab="Some label", main="Some title")
```



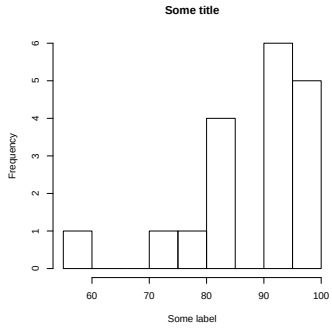
3.8 Parallel Boxplots

```
data1 <- c(77.0, 74.0, 92.3, 76.4, 98.9, 77.4, 79.5, 79.4, 88.5, 69.9, 81.8, 83.5, 73.8, 86.8, 89.4, 70.2, 99.1, 85.5)
data2 <- c(77.5, 82.5, 78.8, 75.3, 79.5, 85.3, 83.5, 85.3, 81.7, 77.2, 67.9, 74.5, 71.7, 86.7, 81.9, 77.9)
boxplot(data1, data2, horizontal=TRUE, xlab="Some label", names=c("first plot label", "second plot label"), main="Some title")
```



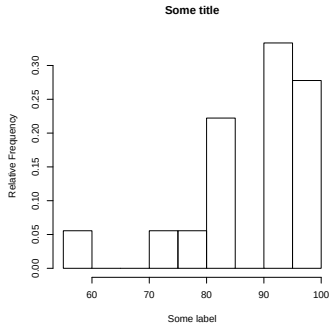
3.9 Histogram

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
hist(data, xlab="Some label", main="Some title")
```



3.10 Relative Frequency Histogram

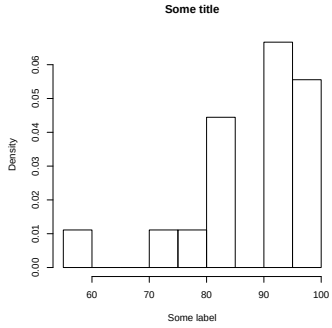
```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
thisHist <- hist(data, plot=FALSE)
thisHist$counts <- thisHist$counts/length(data)
plot(thisHist, xlab="Some label", ylab="Relative Frequency", main="Some title")
```



3.11 Density Histogram

A density histogram is a histogram whose vertical axis has been rescaled to represent "density" rather than a simple frequency or count. It is scaled in such a way that if you took each bar as a rectangle and added up the areas, they would add up to 1. This is accomplished by adding the argument "freq=FALSE" to tell R not to plot frequencies but densities instead.

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
hist(data, xlab="Some label", main="Some title", freq=FALSE)
```

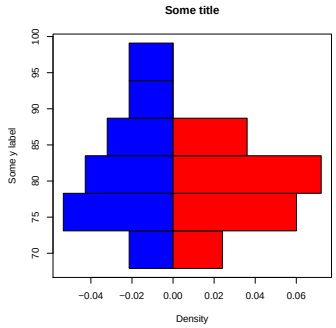


3.12 Back-To-Back Density Histograms

It is actually quite difficult to make back-to-back histograms. There are add-on packages that attempt to do this, but the results seem to be quirky. Here is some code below that will manually draw the rectangles. Everything should be "automated" in that you'll only need to change the data and the labels. Study the code if you want to see how the rectangles were drawn.

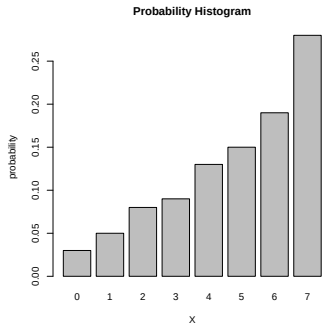
Also, it would be possible to make back-to-back histograms with actual counts (frequencies), but this would only make sense to do if the lengths of the two data sets were similar. If you want to do this, simply change every "$density" in the code to say "$counts". It makes more sense to compare density histograms (which have been re-scaled to have a total area of all the bars that adds to 1).

```
data1 <- c(77.0, 74.0, 92.3, 76.4, 98.9, 77.4, 79.5, 79.4, 88.5, 69.9, 81.8,
83.5, 73.8, 86.8, 89.4, 70.2, 99.1, 85.5)
data2 <- c(77.5, 82.5, 78.8, 75.3, 79.5, 85.3, 83.5, 85.3, 81.7, 77.2, 67.9,
74.5, 71.7, 86.7, 81.9, 77.9)
alldata <- c(data1,data2)
numbreaks <- round(1+log(length(alldata))/log(2),digits=0)+1
bins <- seq(min(alldata),max(alldata),length.out=numbreaks)
hist1 <- hist(data1,breaks=bins, plot=FALSE)
hist2 <- hist(data2,breaks=bins, plot=FALSE)
plot(c(min(-1*hist1$density), max(hist2$density)), c(min(hist1$breaks),
max(hist2$breaks)), type = "n", xlab="Density", ylab="Some y label", main
= "Some title")
rect(-1*hist1$density, hist1$breaks[1:numbreaks-1], 0, hist1$breaks[2:numbreaks],
col="blue")
rect(hist2$density, hist2$breaks[1:numbreaks-1], 0, hist2$breaks[2:numbreaks],
col="red")
```



3.13 Discrete Probability Histogram and Calculations

```
xvals <- c(0,1,2,3,4,5,6,7)
px <- c(0.03,0.05,0.08,0.09,0.13,0.15,0.19,0.28)
names(px) <- xvals
barplot(px, ylab = "probability", xlab="X", main="Probability Histogram")
```



Find the mean, variance, and standard deviation

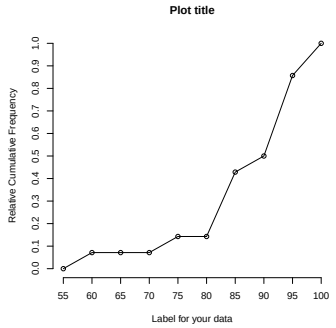
```
xvals <- c(0,1,2,3,4,5,6,7)
px <- c(0.03,0.05,0.08,0.09,0.13,0.15,0.19,0.28)
mu <- sum(xvals*px)
variance <- sum((xvals-mu)^2*px)
sigma <- sqrt(variance)
data.frame(mu,variance,sigma)
```

```
> xvals <- c(0,1,2,3,4,5,6,7)
> px <- c(0.03,0.05,0.08,0.09,0.13,0.15,0.19,0.28)
> mu <- sum(xvals*px)
> variance <- sum((xvals-mu)^2*px)
> sigma <- sqrt(variance)
> data.frame(mu,variance,sigma)

    mu variance   sigma
1 4.85   4.0475 2.01184
```

3.14 Relative Cumulative Frequency (Ogive)

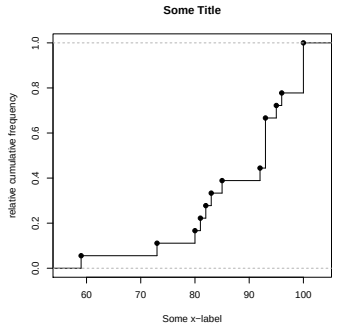
Here is the code for a traditional looking ogive.

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
breaks <- hist(data,plot=FALSE)$breaks
cut.data=cut(data, breaks, right=FALSE)
frequency=table(cut.data)
cummul.freq=cumsum(frequency)
relative.frequency = frequency/sum(frequency)
cf=as.data.frame(cummul.freq)
cummul.freq=cf[,1]
cummul.percentile=cummul.freq/max(cummul.freq)
graph.cummul.perc =c(0, cummul.percentile)
plot(breaks, graph.cummul.perc, xlab ="Label for your data", main = "Plot
title", ylab="Relative Cumulative Frequency", axes=FALSE)
lines(breaks, graph.cummul.perc)
axis(2,at=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1))
axis(1,at=breaks)
```



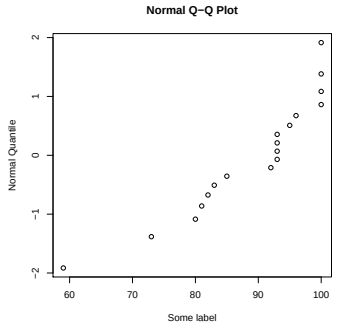
In R, a “relative cumulative frequency” is calculated by the function “ecdf” which stands for “empirical cdf” or “empirical cumulative distribution function”. It is plotted as a step-function rather than connecting the line segments as is sometimes done. This can make a nice looking relative cumulative frequency chart with less coding.

```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
plot.ecdf(data,verticals=TRUE, xlab="Some x-label", ylab="relative cumulative frequency", main="Some Title")
```



3.15 Normal Quantile Plot (Normal Probability Plot)

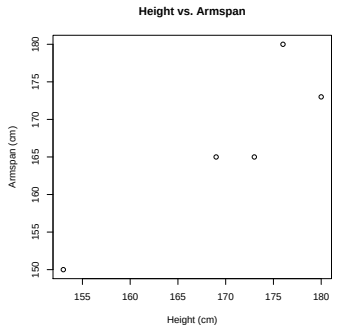
```
data <- c(59,81,100,100,92,96,85,83,95,100,93,93,82,93,100,73,80,93)
qqnorm(data, datax=TRUE, ylab="Some label", xlab="Normal Quantile")
```



4 Paired Data (Scatterplots and LSRL)

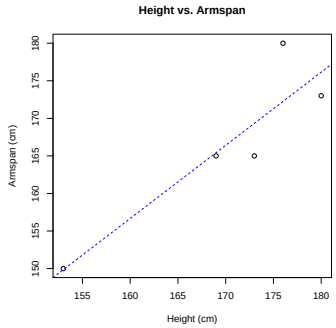
4.1 Basic Scatterplot

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
plot(xdata, ydata, main="Height vs. Armspan", xlab="Height (cm)", ylab="Armspan (cm)")
```



4.2 Scatterplot With Best-Fit Line

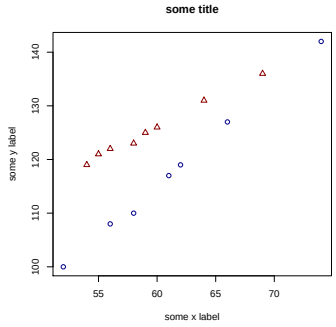
```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
plot(xdata, ydata, main="Height vs. Armspan", xlab="Height (cm)",
ylab="Armspan (cm)")
abline(result, lty=2, col="blue")
```



4.3 Scatterplot displaying two data sets

There are several ways to display two data sets on one scatterplot. A very simple way is to simply create two separate plots with a line of code in between that says "par(new=TRUE)". This will begin the new plot on top of the existing plot. The first plot's labels have been cleared, and the labeling is done on the second plot.

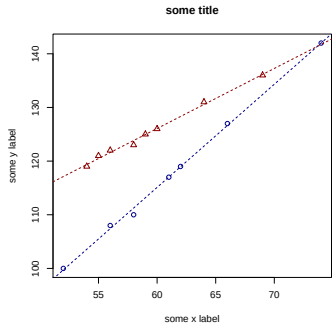
```
xdata1 <- c(74, 66, 61, 56, 58, 52, 62)
ydata1 <- c(142, 127, 117, 108, 110, 100, 119)
xdata2 <- c(69, 59, 55, 60, 58, 64, 54, 56)
ydata2 <- c(136, 125, 121, 126, 123, 131, 119, 122)
length1 <- length(xdata1)
length2 <- length(xdata2)
mydata <- data.frame(c(xdata1,xdata2),c(ydata1,ydata2))
plot(mydata, pch=c(rep(1,length=length1),rep(2,length=length2)),
col=c(rep("darkblue",length=length1),rep("darkred",length=length2)),
xlab="some x label", ylab="some y label", main="some title")
```



4.4 Scatterplot displaying two data sets *and* the lines of best fit

There are several ways to display two data sets on one scatterplot. A very simple way is to simply create two separate plots with a line of code in between that says "par(new=TRUE)". This will begin the new plot on top of the existing plot. The first plot's labels have been cleared, and the labeling is done on the second plot.

```
xdata1 <- c(74, 66, 61, 56, 58, 52, 62)
ydata1 <- c(142, 127, 117, 108, 110, 100, 119)
result1 <- lm(ydata1~xdata1)
xdata2 <- c(69, 59, 55, 60, 58, 64, 54, 56)
ydata2 <- c(136, 125, 121, 126, 123, 131, 119, 122)
result2 <- lm(ydata2~xdata2)
length1 <- length(xdata1)
length2 <- length(xdata2)
mydata <- data.frame(c(xdata1,xdata2),c(ydata1,ydata2))
plot(mydata, pch=c(rep(1,length=length1),rep(2,length=length2)),
col=c(rep("darkblue",length=length1),rep("darkred",length=length2)),
xlab="some x label", ylab="some y label", main="some title")
abline(result1, lty=2, col="darkblue")
abline(result2, lty=2, col="darkred")
```



4.5 Correlation Coefficient (r)

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
cor(xdata,ydata)
```

```
[1] 0.9069859
```

4.6 Linear Regression Model

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
result
```

```
Call:
lm(formula = ydata ~ xdata)

Coefficients:
(Intercept)        xdata
     0.8625       0.9738
```

4.7 Making Predictions: Plugging values into the regression equation

R has a built in function called "predict" that can be complicated to use, especially when you simply want to plug a number into your regression equation. It is easier to simply ask R what the slope and y-intercept are and use them in $y = mx + b$. The following code accomplishes this task.

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
b=as.numeric(result$coef[1])
m=as.numeric(result$coef[2])
# Type your desired x value in the next line. The last line will plug it into the regression equation.
x <- 170
m*x+b
```

```
[1] 166.4052
```

Here is how to use the prediction option built into R.

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
xpredict = data.frame(xdata=170)
predict(result, xpredict)
```

```
       1
166.4052
```

And finally, since most things that you can calculate have a level of uncertainty, R can give you a confidence interval for the prediction.

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
xpredict = data.frame(xdata=170)
predict(result, xpredict, interval="prediction", level=0.90)
```

```
       fit     lwr      upr
1 166.4052 152.371 180.4395
```

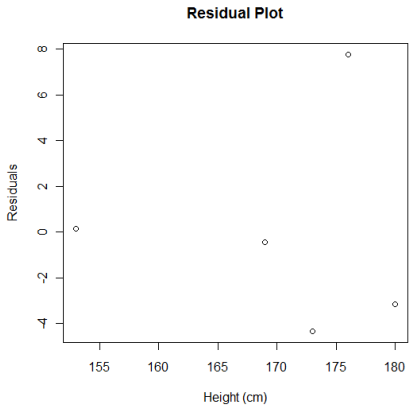
4.8 Linear Regression Model - Full Analysis

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
summary(result)
```

```
Call:
lm(formula = ydata ~ xdata)

Residuals:
      1       2       3       4       5
-4.3266  7.7521 -0.4315 -3.1431  0.1490

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.8625    44.4999   0.019   0.9858
xdata         0.9738     0.2611   3.730   0.0336 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.444 on 3 degrees of freedom
Multiple R-squared:  0.8226,        Adjusted R-squared:  0.7635
F-statistic: 13.91 on 1 and 3 DF,  p-value: 0.03357
```

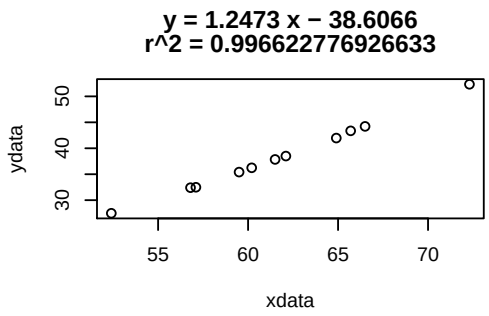
4.9 Residual Plot

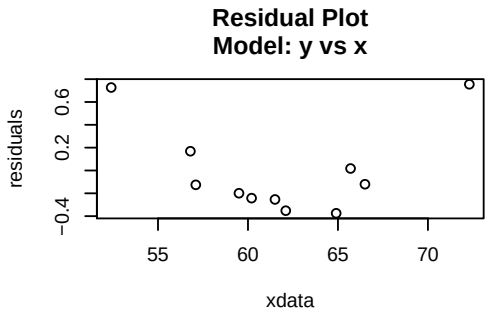
```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
residuals <- result$residuals
plot(xdata, residuals, main="Residual Plot", xlab="Height (cm)",
ylab="Residuals")
```

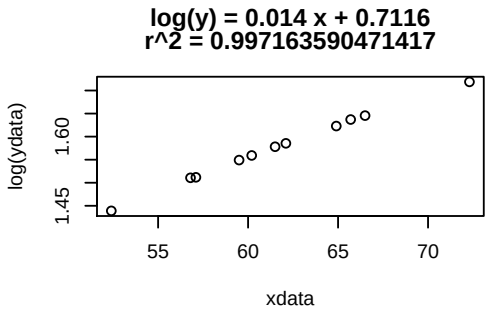


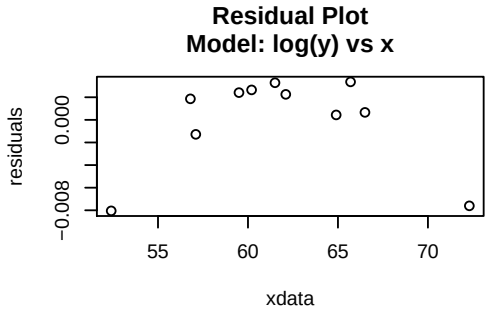
4.10 Nonlinear Analysis (using logarithms)

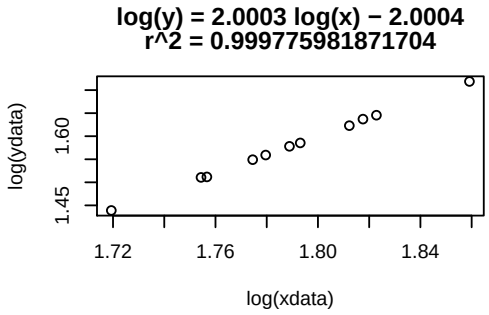
```
xdata <- c(59.5, 65.7, 62.1, 64.9, 52.4, 57.1, 60.2, 61.5, 72.3, 66.5, 56.8)
ydata <- c(35.41, 43.36, 38.50, 41.97, 27.48, 32.49, 36.24, 37.85, 52.33, 44.22, 32.41)
# Set up a 3x2 graphics output
par(mfrow=c(3,2))
#Generate a linear model
result <- lm(ydata~xdata)
modelCoef <- round(result$coefficients,digits=4)
interceptSign <- if(modelCoef[1]<0) "-" else "+"
thisEqn <- paste("y = ",modelCoef[2]," x ",interceptSign," ",abs(modelCoef[1]),sep="")
thisRsq <- paste("r^2 = ",summary(result)$r.squared,sep="")
thisTitle <- paste(c(thisEqn,thisRsq),sep="\n")
plot(xdata,ydata, main=thisTitle)
plot(xdata,result$residuals, ylab="residuals", main="Residual Plot\nModel: y vs x")
#Generate a linear model of log(y) vs x (representing an exponential model)
logresult <- lm(log(ydata,base=10)~xdata)
modelCoef <- round(logresult$coefficients,digits=4)
interceptSign <- if(modelCoef[1]<0) "-" else "+"
thisEqn <- paste("log(y) = ",modelCoef[2]," x ",interceptSign," ",abs(modelCoef[1]),sep="")
thisRsq <- paste("r^2 = ",summary(logresult)$r.squared,sep="")
thisTitle <- paste(c(thisEqn,thisRsq),sep="\n")
plot(xdata,log(ydata,base=10), ylab="log(ydata)", main=thisTitle)
plot(xdata,logresult$residuals, ylab="residuals", main="Residual Plot\nModel: log(y) vs x")
#Generate a linear model of log(y) vs log(x) (representing a power model)
loglogresult <- lm(log(ydata,base=10)~log(xdata,base=10))
modelCoef <- round(loglogresult$coefficients,digits=4)
interceptSign <- if(modelCoef[1]<0) "-" else "+"
thisEqn <- paste("log(y) = ",modelCoef[2]," log(x) ",interceptSign," ",abs(modelCoef[1]),sep="")
thisRsq <- paste("r^2 = ",summary(loglogresult)$r.squared,sep="")
thisTitle <- paste(c(thisEqn,thisRsq),sep="\n")
plot(log(xdata,base=10),log(ydata,base=10),xlab="log(xdata)", ylab="log(ydata)", main=thisTitle)
plot(log(xdata,base=10),loglogresult$residuals, xlab="log(xdata)", ylab="residuals", main="Residual Plot\nModel: log(y) vs log(x)")
```

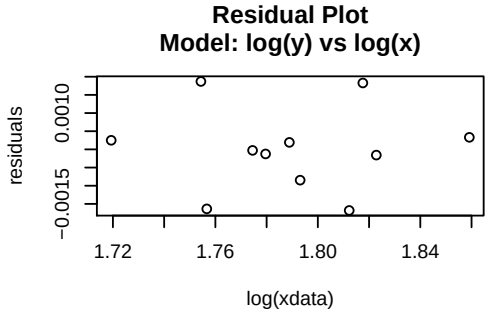








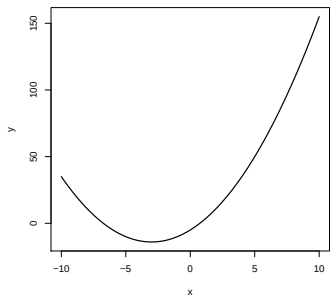




5 Plotting functions

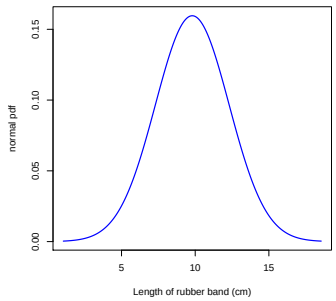
5.1 Simple function: $f(x) = x^2 + 6x - 5$

```
x=seq(-10,10,length=500)
y=x^2+6*x-5
plot(x, y, type="l", lwd=2)
```



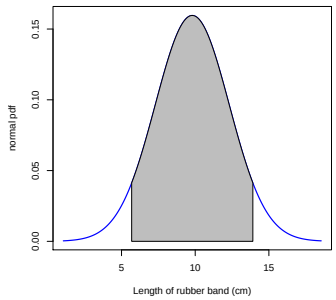
5.2 Normal Distribution

```
mu=9.8
sigma=2.5
x=seq(mu-3.5*sigma,mu+3.5*sigma,length=500)
y=dnorm(x,mean=mu,sd=sigma)
plot(x, y, type="l", lwd=2, col="blue", xlab = "Length of rubber band (cm)", ylab
= "normal pdf")
```



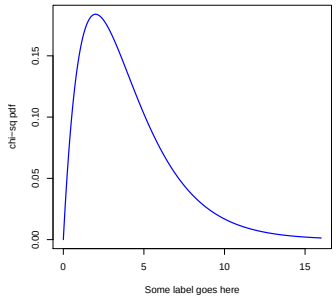
5.3 Normal Distribution With Region Shaded - (90% confidence interval)

```
mu=9.8
sigma=2.5
zc=1.645
x=seq(mu-3.5*sigma,mu+3.5*sigma,length=500)
y=dnorm(x,mean=mu,sd=sigma)
plot(x, y, type="l", lwd=2, col="blue", xlab = "Length of rubber band (cm)", ylab
= "normal pdf")
x=seq(mu-zc*sigma,mu+zc*sigma,length=500)
y=dnorm(x,mean=mu,sd=sigma)
polygon(c(mu-zc*sigma,x,mu+zc*sigma),c(0,y,0),col="gray")
```



5.4 Chi-Square Distribution

```
df = 4
x=seq(0, 4*df, length=500)
y=dchisq(x, df)
plot(x, y, type="l", lwd=2, col="blue", xlab = "Some label goes here", ylab =
"chi-sq pdf")
```



6 Statistical Distributions

For every distribution, R has 4 functions. One starts with a "d" and stands for the distribution function (pdf). One starts with a "p" and stands for the cumulative distribution function (cdf). One starts with a "q" and stands for quantile. It gives the inverse of the cdf function. The other starts with an "r" and will generate random numbers using the chosen distribution.

6.1 Binomial Distribution

6.1.1 binompdf: *dbinom*(X, n, p)

If you have 15 trials with a probability of success of $p = 0.24$, you can calculate the probability of observing 5 successes:

```
dbinom(5,15,0.24)
```

```
> dbinom(5,15,0.24)
[1] 0.153726
```

6.1.2 binomcdf: *pbinom*(X, n, p)

If you have 15 trials with a probability of success of $p = 0.24$, you can calculate the probability of observing 5 successes or less:

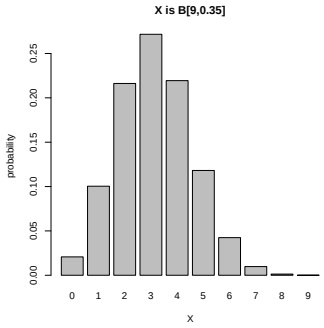
```
pbinom(5,15,0.24)
```

```
> pbinom(5,15,0.24)
[1] 0.87276
```

6.2 Histogram of a Discrete Probability Distribution

We can use R to construct a *probability model* for a discrete random variable (that follows something like the binomial or geometric distribution) and then make a *probability histogram* displaying the distribution.

```
xvals <- 0:9
px <- dbinom(xvals,9,0.35)
names(px) <- xvals
barplot(px, ylab = "probability", xlab="X", main="X is B[9,0.35]")
```



6.3 Normal Distribution

6.3.1 normalcdf: *pnorm*(X, μ, σ)

If you have a z-score of $z = 1.31$, you can find the probability that z is less than 1.31:

```
pnorm(1.31)
```

```
> pnorm(1.31)
[1] 0.9049021
```

If you have a population with a mean of $\mu = 72.4$ and a standard deviation of $\sigma = 2.7$, you can find the probability of obtaining a random observation that is less than 70 by first calculating a z-score (using R) and then using pnorm.

```
z <- (70-72.4)/2.7
pnorm(z)
```

```
> z <- (70-72.4)/2.7
> pnorm(z)
[1] 0.1870314
```

You could also just directly enter the mean and standard deviation to calculate the same probability.

```
pnorm(70,72.4,2.7)
```

```
> pnorm(70,72.4,2.7)
[1] 0.1870314
```

6.3.2 Inverse Norm: *qnorm*$(area, \mu, \sigma)$

You can find the z-score that has an area of 0.85 to the left:

```
qnorm(0.85)
```

```
> qnorm(0.85)
[1] 1.036433
```

If you have a population with a mean of $\mu = 72.4$ and a standard deviation of $\sigma = 2.7$, you can find value for which 85% of the observations will be less (the 85th percentile):

```
qnorm(0.85,72.4,2.7)
```

```
> qnorm(0.85,72.4,2.7)
[1] 75.19837
```

6.4 t-Distribution

6.4.1 tcdf: *pt*(X, df)

If you have a t-score of $t = 1.31$ with 8 degrees of freedom, you can find the probability that t is less than 1.31:

```
pt(1.31,8)
```

```
> pt(1.31,8)
[1] 0.8867207
```

6.4.2 Inverse t: $qt(area, df)$

If you have a t distribution with 8 degrees of freedom, you can find the t-score for which 85% of the observations are less (the 85th percentile):

```
qt(0.85,8)
```

```
> qt(0.85,8)
[1] 1.108145
```

6.5 χ^2 Distribution

6.5.1 χ^2cdf: $pchisq(X, df)$

If you have a χ^2 value of $\chi^2 = 10.3$ with 8 degrees of freedom, you can find the probability that χ^2 is less than 10.3:

```
pchisq(10.3,8)
```

```
> pchisq(10.3,8)
[1] 0.755402
```

6.5.2 Inverse χ^2: $qchisq(area, df)$

If you have a χ^2 distribution with 8 degrees of freedom, you can find the χ^2 value for which 85% of the observations are less (the 85th percentile):

```
qchisq(0.85,8)
```

```
> qchisq(0.85,8)
[1] 12.02707
```

7 Inference Procedures (Confidence Intervals and Hypothesis Tests)

7.1 1 Prop Z Interval

You would like to run for city council, and you would like to have an informed stance on city zoning related to a new commercial development. You obtain a random sample of residents, and 96 out of 431 residents are in favor of the new development. Construct a 90% confidence interval for the proportion of all residents in the city who are in favor of the new development.

```
prop.test(96, 431, conf.level=0.90, correct=FALSE)
```

```
        1-sample proportions test without continuity correction

data:  96 out of 431, null probability 0.5
X-squared = 132.5313, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.5
90 percent confidence interval:
 0.1915587 0.2573762
sample estimates:
        p
0.2227378
```

Note: R actually runs the whole hypothesis test and reports the confidence interval as part of the results.

7.2 1 Prop Z Test

A recent poll claims that 29% of American adults get less than 7 hours of sleep each night. You ask a random sample of people 160 people from your town, and you find that 51 of them get less than 7 hours of sleep each night. Is there evidence, at the 5% significance level, that the proportion of people who get less than 7 hours of sleep each night is in your town is different than the recent poll?

```
prop.test(51, 160, p=.29, alt="two.sided", correct=FALSE)
```

```
        1-sample proportions test without continuity correction

data:  51 out of 160, null probability 0.29
X-squared = 0.6423, df = 1, p-value = 0.4229
alternative hypothesis: true p is not equal to 0.29
95 percent confidence interval:
 0.2515198 0.3944794
sample estimates:
      p
0.31875
```

Note: R actually runs a Chi-square test with 1 degree of freedom. The results will be identical to what you would have calculated with a z-test. If you would like to use a different alternate hypothesis, choose "less", "greater", or "two.sided".

7.3 T Interval

You are researching a new car, and you are interested in finding the mean sticker price for a specific make and model. You do some research and come across the following random sample from car dealerships in your state (units are thousands of dollars).

17.1, 17.0, 16.4, 17.3, 17.9, 17.5, 16.9, 17.3, 16.7, 16.6, 17.4, 17.4

Estimate the mean sticker price for this model of car in your state at a 85% confidence level.

```
data <- c(17.1, 17.0, 16.4, 17.3, 17.9, 17.5, 16.9, 17.3, 16.7, 16.6, 17.4, 17.4)
t.test(data, conf.level=0.85)
```

```
        One Sample t-test

data:  data
t = 139.0373, df = 11, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
85 percent confidence interval:
 16.93439 17.31561
sample estimates:
mean of x
   17.125
```

7.4 T Test

You have a friend who claims that he can consistently run a sub-5 minute mile, on average. You have gathered his last 9 times (in minutes) for running a mile:

5.10, 4.92, 5.18, 4.94, 4.87, 5.04, 4.99, 5.09, 5.08

Is there evidence, at the 5% significance level, for your friend's claim, that the mean running time for a mile is less than 5 minutes?

```
data <- c(5.10, 4.92, 5.18, 4.94, 4.87, 5.04, 4.99, 5.09, 5.08)
t.test(data, conf.level=0.95, mu = 5, alt="less")
```

```
        One Sample t-test

data:  data
t = 0.6974, df = 8, p-value = 0.7473
alternative hypothesis: true mean is less than 5
95 percent confidence interval:
    -Inf 5.08555
sample estimates:
mean of x
 5.023333
```

7.5 Chi-squared Variance (Standard Deviation) Confidence Interval

You have a friend who claims that he can consistently run a sub-5 minute mile, on average. You have gathered his last 9 times (in minutes) for running a mile:

5.10, 4.92, 5.18, 4.94, 4.87, 5.04, 4.99, 5.09, 5.08

Is there evidence, at the 5% significance level, for your friend's claim, that the overall standard deviation for running time for a mile is less than 0.2 minutes?

Note: This is not a built-in function of R, so here is code to accomplish this. The code for the confidence interval below is two sided.

```
data <- c(5.10, 4.92, 5.18, 4.94, 4.87, 5.04, 4.99, 5.09, 5.08)
df <- length(data)-1
conf.level <- 0.90
upper.chisq <- qchisq((1-conf.level)/2,df)
lower.chisq <- qchisq((1-conf.level)/2+conf.level,df)
conf.int <- sqrt(c(df*var(data)/lower.chisq, df*var(data)/upper.chisq))
names(conf.int) <- c("lower","upper")
conf.int
```

```
     lower      upper
0.07209402 0.17174202
```

7.6 Chi-squared Variance (Standard Deviation) Hypothesis Test

You have a friend who claims that he can consistently run a sub-5 minute mile, on average. You have gathered his last 9 times (in minutes) for running a mile:

5.10, 4.92, 5.18, 4.94, 4.87, 5.04, 4.99, 5.09, 5.08

Is there evidence, at the 5% significance level, for your friend's claim, that the overall standard deviation for running time for a mile is less than 0.2 minutes?

Note: This is not a built-in function of R, so here is code to accomplish this. The code below is for a left-tailed test. To make it a right-tailed test, the p-value would need to be 1-pchisq. It would need to be doubled for a two-tailed test.

```
data <- c(5.10, 4.92, 5.18, 4.94, 4.87, 5.04, 4.99, 5.09, 5.08)
df <- length(data)-1
sigma <- 0.2
test.statistic <- df*sd(data)^2/sigma^2
names(test.statistic) <- "chi-squared test statistic"
p.value <- pchisq(test.statistic,df)
names(p.value) <- "p-value"
test.statistic
p.value
```

```
chi-squared test statistic
                     2.015

   p-value
0.01945146
```

7.7 2 Prop Z Interval

You own an ice cream company that has stores in two large cities. You are interested in your customer's favorite flavor and decide to gather data. You gather random samples of customer's opinions from each city. In one city, out of 392 surveys, 203 chose "Chocolate Tornado" as their favorite flavor. In the other city, out of 428 surveys, 252 chose "Chocolate Tornado" as

their favorite flavor. Estimate the difference proportion of customers who choose "Chocolate Tornado" as their favorite flavor between the two cities. Use a 95% confidence level.

```
prop.test(c(203,252), c(392,428), conf.level=0.95, correct=FALSE)
```

```
        2-sample test for equality of proportions without continuity
        correction

data:  c(203, 252) out of c(392, 428)
X-squared = 4.1675, df = 1, p-value = 0.04121
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.138897580 -0.002958227
sample estimates:
   prop 1    prop 2
0.5178571 0.5887850
```

7.8 2 Prop Z Test

You own an ice cream company that has stores in two large cities. You are interested in your customer's favorite flavor and decide to gather data. You gather random samples of customer's opinions from each city. In one city, out of 392 surveys, 203 chose "Chocolate Tornado" as their favorite flavor. In the other city, out of 428 surveys, 252 chose "Chocolate Tornado" as their favorite flavor. Is there evidence that the proportion of customers who choose "Chocolate Tornado" as their favorite flavor is different between the two cities? Use a 10% significance level.

```
prop.test(c(203,252), c(392,428), conf.level=0.90, correct=FALSE)
```

```
        2-sample test for equality of proportions without continuity
        correction

data:  c(203, 252) out of c(392, 428)
X-squared = 4.1675, df = 1, p-value = 0.04121
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.138897580 -0.002958227
sample estimates:
   prop 1    prop 2
0.5178571 0.5887850
```

7.9 2 Sample T Interval

Two students are comparing designs of paper airplanes. They each make a paper airplane and record the flight distance for various throws. The data is displayed below. Units are feet.

Plane #1: 18.5, 19.6, 20.3, 23.4, 26.5, 24.9, 23.4, 21.7, 14.6, 22.9, 17.3, 20.1, 21.4, 23.0, 19.4

Plane #2: 26.3, 28.3, 28.2, 30.6, 29.7, 17.4, 28.5, 29.1, 25.3, 24.7, 30.2, 24.2, 26.1, 27.0, 27.1

Estimate the difference in mean flight distances at an 80% confidence level.

```
data1 <- c(18.5, 19.6, 20.3, 23.4, 26.5, 24.9, 23.4, 21.7, 14.6, 22.9, 17.3, 20.1, 21.4, 23.0, 19.4)
data2 <- c(26.3, 28.3, 28.2, 30.6, 29.7, 17.4, 28.5, 29.1, 25.3, 24.7, 30.2, 24.2, 26.1, 27.0, 27.1)
t.test(data1, data2, conf.level=0.80)
```

```
        Welch Two Sample t-test

data:  data1 and data2
t = -4.9381, df = 27.889, p-value = 3.317e-05
alternative hypothesis: true difference in means is not equal to 0
80 percent confidence interval:
 -7.232058 -4.194609
sample estimates:
mean of x mean of y
 21.13333  26.84667
```

7.10 2 Sample T Test

Same as above. You might choose to identify an alternate hypothesis.

```
data1 <- c(18.5, 19.6, 20.3, 23.4, 26.5, 24.9, 23.4, 21.7, 14.6, 22.9, 17.3, 20.1, 21.4, 23.0, 19.4)
data2 <- c(26.3, 28.3, 28.2, 30.6, 29.7, 17.4, 28.5, 29.1, 25.3, 24.7, 30.2, 24.2, 26.1, 27.0, 27.1)
t.test(data1, data2, conf.level=0.80, alt="two.sided")
```

```
        Welch Two Sample t-test

data:  data1 and data2
t = -4.9381, df = 27.889, p-value = 3.317e-05
alternative hypothesis: true difference in means is not equal to 0
80 percent confidence interval:
 -7.232058 -4.194609
sample estimates:
mean of x mean of y
 21.13333  26.84667
```

7.11 2 Sample F Test

```
data1 <- c(18.5, 19.6, 20.3, 23.4, 26.5, 24.9, 23.4, 21.7, 14.6, 22.9, 17.3, 20.1, 21.4, 23.0, 19.4)
data2 <- c(26.3, 28.3, 28.2, 30.6, 29.7, 17.4, 28.5, 29.1, 25.3, 24.7, 30.2, 24.2, 26.1, 27.0, 27.1)
var.test(data1, data2, conf.level=0.80, alt="two.sided")
```

```
        F test to compare two variances

data:  data1 and data2
F = 0.8814, num df = 14, denom df = 14, p-value = 0.8166
alternative hypothesis: true ratio of variances is not equal to 1
80 percent confidence interval:
 0.4358013 1.7825312
sample estimates:
ratio of variances
         0.8813792
```

7.12 χ^2 Goodness of Fit Test

The mayor of your city has hired you to review the public perception of a major construction project. When it had been proposed, a large amount of data was collected. The proportions of people with a "positive perception" of the project was reported to be (by city district):

District 1	District 2	District 3	District 4	District 5	District 6
0.84	0.92	0.57	0.79	0.90	0.88

The mayor wants to know, as the project nears completion, if the public's perception has changed. You collect a sample of 50 people from each district. You report, in the table below, the number of people who have a "positive perception" of the project.

District 1	District 2	District 3	District 4	District 5	District 6
44	47	43	35	48	43

Note: For this test, R makes calculations a bit different. The expected counts will be forced to add to the same total as the observed counts. We do not want to impose this condition - we simply want to calculate what the expected counts "should" be if you sampled 50 people from each district. The code below will conduct the test "by hand" and report the results. You would need to modify the observed, sampsizes, and props vectors (the first three lines).

```
observed <- c(44, 47, 43, 35, 48, 43)
sampsizes <- c(50, 50, 50, 50, 50, 50)
props <- c(0.84, 0.92, 0.57, 0.79, 0.90, 0.88)
expected <- props*sampsizes
chisq <- sum((observed-expected)^2/expected)
pvalue <- pchisq(chisq,length(observed)-1,lower.tail=FALSE)
data.frame(chisq,pvalue)
```

```
    chisq    pvalue
1 8.229556 0.1440299
```

Note: Depending on your data, you might find it more convenient to simply enter observed and expected counts like this:

```
observed <- c(44, 47, 43, 35, 48, 43)
expected <- c(42, 46, 28.5, 39.5, 45, 44)
chisq <- sum((observed-expected)^2/expected)
pvalue <- pchisq(chisq,length(observed)-1,lower.tail=FALSE)
data.frame(chisq,pvalue)
```

```
    chisq    pvalue
1 8.229556 0.1440299
```

7.13 χ^2 Independence or Homogeneity Test

You are on student council, and you are interested in finding out if there is any preference among the student body about this year's homecoming theme. Specifically, you want to see if there is a difference among each of the grade-levels in the proportions of students who favor each suggested homecoming theme. You sample 25 students from each class and collect the following data, shown in the table below.

	Fresh	Soph	Junior	Senior
Theme 1	13	11	7	5
Theme 2	6	8	13	8
Theme 3	6	6	5	12

Is there evidence, at a 10% significance level, that there is a difference, across the grade levels, in the proportions of preference for the homecoming themes?

```
row1 <- c(13,11,7,5)
row2 <- c(6,8,13,8)
row3 <- c(6,6,5,12)
counts <- t(as.matrix(data.frame(row1,row2,row3)))
chisq.test(counts, correct=FALSE)
```

```
        Pearson's Chi-squared test

data:  counts
X-squared = 11.743, df = 6, p-value = 0.06795
```

7.14 LinReg T-Test

The number of students late to school first hour and the outdoor temperature were recorded for randomly chosen days in April and May. The following data was collected:

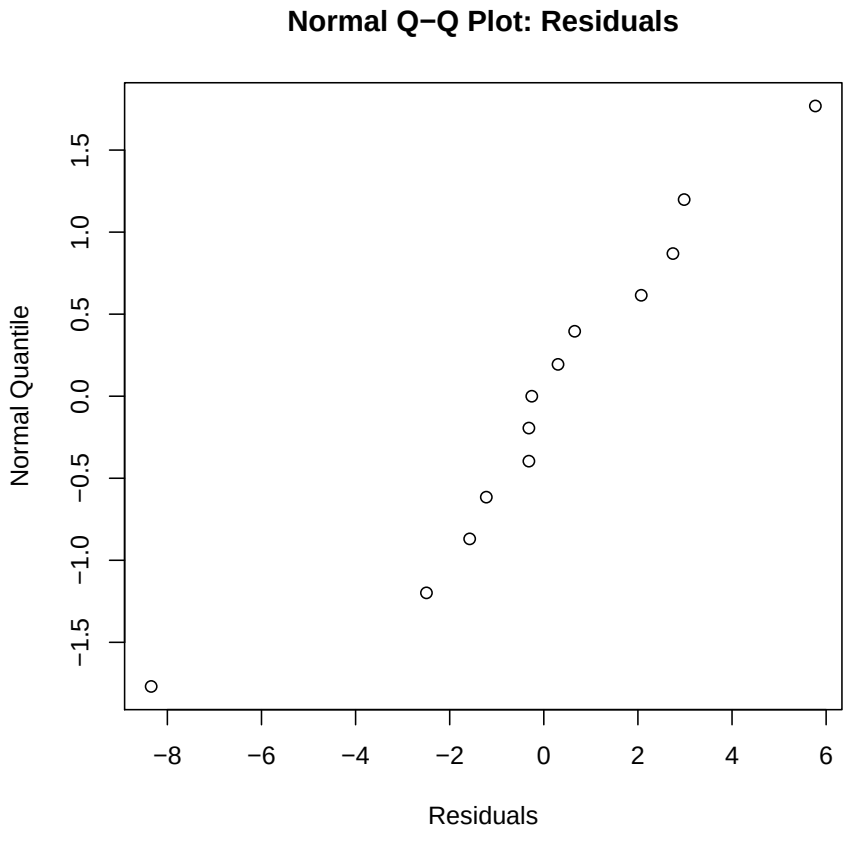
Temperature	44	48	52	53	67	67	68	68	72	78	79	79	81
Tardies	10	10	14	15	16	13	5	14	14	20	14	14	12

Is there significant evidence that the outdoor temperature and the number of tardies to first hour are related? (This could also be worded, "Is there significant evidence that there is a correlation between the outdoor temperature and the number of tardies?")

This code will construct the linear model and give you the full report needed for inference. The code is also provided to analyze whether the residuals appear to be approximately normally distributed (a normal probability plot of residuals is given).

```
xdata <- c(44, 48, 52, 53, 67, 67, 68, 68, 72, 78, 79, 79, 81)
ydata <- c(10, 10, 14, 15, 16, 13, 5, 14, 14, 20, 14, 14, 12)
result <- lm(ydata~xdata)
resid <- result$residuals
qqnorm(resid, datax=TRUE, main="Normal Q-Q Plot: Residuals", ylab="Residuals", xlab="Normal Quantile")
summary(result)
```

```
Call:
lm(formula = ydata ~ xdata)

Residuals:
    Min      1Q  Median      3Q     Max
-8.3443 -1.2218 -0.2559  2.0707  5.7713

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.33043    5.33835   1.373    0.197
xdata        0.08844    0.07972   1.109    0.291

Residual standard error: 3.498 on 11 degrees of freedom
Multiple R-squared:  0.1006,        Adjusted R-squared:  0.01886
F-statistic: 1.231 on 1 and 11 DF,  p-value: 0.2909
```



Note: The p-value is found in the column "Pr(>|t|)" column and the row "xdata". In this case, it is 0.291. If this was a one-sided test, this value would need to be divided in two.

7.15 LinReg T Interval

If we want to construct a confidence interval for the slope or intercept of the regression line, the following code is used.

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
confint(result, level=0.90)
```

```
                    5 %       95 %
(Intercept) -103.8618724 105.586803
xdata          0.3594009   1.588161
```

7.16 LinReg Prediction and Confidence Intervals

If we want to predict the *mean* response for the number of tardies at a given number of absences and construct a 90% confidence interval, we'd use the following code.

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
xpredict = data.frame(xdata=170)
predict(result, xpredict, interval="confidence", level=0.90)
```

```
       fit      lwr      upr
1 166.4052 160.6747 172.1358
```

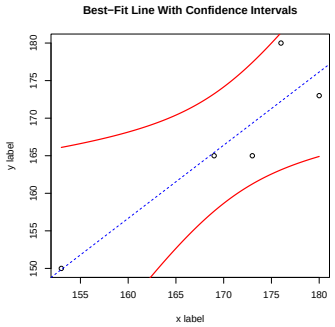
If we want to construct a confidence interval for an *individual's* predicted response (rather than the mean predicted response), we'd use the following code.

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
xpredict = data.frame(xdata=170)
predict(result, xpredict, interval="predict", level=0.90)
```

```
       fit     lwr      upr
1 166.4052 152.371 180.4395
```

A powerful plot to construct for confidence (or prediction) intervals can be constructed that displays the intervals directly on the scatterplot.

```
xdata <- c(173,176,169,180,153)
ydata <- c(165,180,165,173,150)
result <- lm(ydata~xdata)
residuals <- result$residuals
plot(xdata, ydata, main="Best-Fit Line With Confidence Intervals", xlab="x label",
ylab="y label")
abline(result, lty=2, col="blue")
newx <- seq(min(xdata), max(xdata), length=100)
nd <- data.frame(xdata=newx)
conf <- predict(result,nd,interval="confidence")
points(newx,conf[,"lwr"], type="l", lwd=2, col="red")
points(newx,conf[,"upr"], type="l", lwd=2, col="red")
```



7.17 F-Test (One Way ANOVA): Testing for differences in multiple means

Suppose you are growing plants under three different lighting conditions. A UV bulb was used with one group during normal daylight hours. A new different brand of UV bulb was used with another group. The third group was exposed to natural sunlight. All other factors such as temperature and water were the same across the groups. Each group had 5 plants. After 14 days, the height of each plant (in centimeters) was measured.

UV1: 9, 8, 10, 15, 16 (Mean: 11.6, Variance: 13.3)
UV2: 11, 9, 15, 11, 8 (Mean: 10.8, Variance: 7.2)
Sun: 18, 14, 15, 16, 21 (Mean: 16.8, Variance: 7.7)

Note: R forces you to think of each sample as a different treatment. We join all three samples into one list and create another list with the named treatments (in this case "UV1", "UV2", and "sun".

```
data1 <- c(9, 8, 10, 15, 16)
names(data1) <- rep("UV1",length(data1))
data2 <- c(11, 9, 15, 11, 8)
names(data2) <- rep("UV2",length(data2))
data3 <- c(18, 14, 15, 16, 21)
names(data3) <- rep("sun",length(data3))
values <- c(data1,data2,data3)
treatments <- names(values)
result <- aov(values~treatments)
summary(result)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
treatments   2  106.1   53.07   5.645 0.0187 *
Residuals   12  112.8    9.40
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note: The p-value is found in the column "Pr(>F)"

7.18 Sign Test for the Median

A group of students has been keeping track of the wind speed (miles per hour) at 10am for an SRS of 12 days in the spring. The following data was collected:

19.76, 4.53, 31.36, 4.9, 4.14, 7.22, 15.74, 9.68, 8.37, 11.91, 7.34, 23.29

Is there evidence, at a 10% significance level, that the median wind speed is less than 16 miles per hour?

Note: This data set is small and skewed to the right. A t-test would not be appropriate. The non-parametric sign-test would be more appropriate. The code below subtracts the hypothesized median from each value and observes the sign. The number of positive signs is observed - this value should be 50% if the hypothesized median is correct. This allows the p-value to be calculated with the binomial distribution. We will run the binom.test and note the p-value.

```
hypmedian <- 16
data <- c(19.76, 4.53, 31.36, 4.9, 4.14, 7.22, 15.74, 9.68, 8.37, 11.91, 7.34, 23.29)
signs <- sign(data - hypmedian)
compare <- c(sum(signs == 1), sum(signs == -1))
binom.test(compare, alt="less")
```

```
        Exact binomial test

data:  compare
number of successes = 3, number of trials = 12, p-value = 0.073
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.5273266
sample estimates:
probability of success
                  0.25
```

Note: Even though the binom.test is the correct procedure to run, the only meaningful result is the p-value. We are not interested in proportions of "success" - only how likely it is to see 3 out of 12 values below the hypothesized median. (It should have been half.)

7.19 Wilcoxon Signed Rank Test for the Median

A group of students has been keeping track of the wind speed (miles per hour) at 10am for an SRS of 12 days in the spring. The following data was collected:

19.76, 4.53, 31.36, 4.9, 4.14, 7.22, 15.74, 9.68, 8.37, 11.91, 7.34, 23.29

Is there evidence, at a 10% significance level, that the median wind speed is less than 16 miles per hour?

Note: This data set is small and skewed to the right. A t-test would not be appropriate. A non-parametric test (like this one) would be more appropriate.

```
data <- c(19.76, 4.53, 31.36, 4.9, 4.14, 7.22, 15.74, 9.68, 8.37, 11.91, 7.34, 23.29)
wilcox.test(data, alt="less", mu=16)
```

```
        Wilcoxon signed rank test

data:  data
V = 19, p-value = 0.0647
alternative hypothesis: true location is less than 16
```