ST314: Introduction to Statistics for Engineers

R Commands

Written by Katie Jager

This subset of notes contains only R commands.

Binomial Distribution in R

Function	Function Values	What does it do?
dbinom(x,n,p)	Where x is the value of interest, n = sample size, and p is the probability of success	This is the probability mass function. Finds $P(X = x)$ for binomial distribution
pbinom(x,n,p)	Where x is the value of interest, n = sample size, and p is the probability of success	This is the cumulative distribution function. Finds $P(X \leq x)$ for binomial distribution

```
# Example R CODE
# How likely is it that the student will get 10 questions correct?
P(X=10)
dbinom(10, 20, 0.25)

# How likely is it that the student will get less than 20% of the
# questions correct? P(X<=3)
pbinom(3, 20, 0.25)

# How likely is it that the student will get more than 10
# of the questions correct? P(X>10) = PX(X>=11) = 1-P(X<=10)
1-pbinom(10,20,0.25)
```

Binomial Distribution in R Video Tutorial

Poisson Distribution in R

Function	Function Values	What does it do?
dpois(x,lambda)	Where x is the value of interest, lambda is the rate parameter λ	This is the probability mass function. Finds $\mathbf{P(X = x)}$ for Poisson distribution
ppois(x,lambda)	Where x is the value of interest, lambda is the rate parameter λ	This is the **cumulative** distribution function. Finds $\mathbf{P(X \leq x)}$ for Poisson distribution

```
# Example R CODE
# How likely is it the individual will get exactly 10 texts in an
hour? P(X=10)
dpois(10, 8)

# How likely is it they will get 3 or fewer texts in an hour? P(X<=3)
ppois(3, 8)

# How likely is it the individual will get more than 1 text in an
hour?
# P(X>1) = 1 – P(X<=1)= 1-[P(X=0)+P(X=1)]
1-ppois(1,8)
```

Poisson Distribution in R Video Tutorial

Uniform Distribution in R

Function	Function Values	What does it do?
punif(x,a,b)	Where x is the value of interest, a is the lower bound and b is the upper bound of the uniform dist.	This is the cumulative density function. Finds $P(X \leq x)$ for Uniform distribution
qunif(p,a,b)	Where p is percentage that falls below x, a is the lower bound and b is the upper bound of the uniform dist.	This is the inverse of cumulative density function. Finds percentiles for Uniform distribution. That is, x_p for expression $P(X \leq x_{\mathrm{p}}) = p$

```
# Example R code for a uniform random variable when a = 0 and b= 4

# P(X<1) Area to the left of x
punif(1,0,4)

# P(X>1) Area to the right of x
1-punif(1,0,4)

# Find the value of the 20th percentile
qunif(0.20,a,b)
```

Uniform Distribution in R Video Tutorial

Gamma Distribution in R

Function	Function Values	What does it do?
pgamma(x,shape, rate)	Where x is the value of interest, shape = α and rate = $\frac{1}{\beta}$	Cumulative density function, finds $P(X \leq x)$ for gamma distribution
qgamma(p,shape, rate)	Where p is percentage that falls below x, shape = α and rate = $\frac{1}{\beta}$	Inverse of cumulative density function. Finds percentiles for gamma distribution. That is, x_p for expression $P(X \leq x_{\mathrm{p}}) = p$

```
# Example R code for alpha = 3 and beta = 12.4 and x = 50
# P(X<50) Area to the left of x
pgamma(50,3,rate = 1/12.4)

# P(X>50) Area to the right of x
1-pgamma(50,3,rate = 1/12.4)

# 50th percentile Median
qgamma(0.5,3,rate = 1/12.4)
```

Gamma Distribution in R Video Tutorial

Exponential Distribution in R

Function	Function Values	What does it do?
pexp(x,lambda)	Where x is the value of interest, λ is the rate parameter	Cumulative density function, finds $P(X \leq x)$ for exponential distribution
qexp(p,lambda)	Where p is percentage that falls below x, λ is the rate parameter	Inverse of cumulative density function. Finds percentiles for exponential distribution. That is, x_p for expression $P\left(X \leq x_{\mathrm{p}}\right) = p$

```
# Example R code for an exponential random variable when  lambda = 1/2.725

# P(X<5) Area to the left of x
pexp(5, 1/2.725)

# P(X>5) Area to the right of x
1-pexp(5, 1/2.725)

# 50th percentile
qexp(0.5, 1/2.725)
```

Exponential Distribution in R Video Tutorial

Normal Distribution in R

Function	Function Values	What does it do?
pnorm(x,mu, sigma)	Where x is the value of interest, mu is the mean and sigma the standard deviation	Cumulative density function, finds $P(X \leq x)$ for norm distribution
qnorm(p,mu, sigma)	Where p is percentage that falls below x, mu is the mean and sigma the standard deviation	Inverse of cumulative density function. Finds percentiles for norm distribution. That is, x_p for expression $P\left(X \leq x_{\mathrm{p}}\right) = p$

```
# Example R code for a Normal Random Variable when mu = 2 and sigma = 0.6

# P(X<x) Area to the left of x
pnorm(1.5, 2,0.6)

# P(X>x) Area to the righ t of x
1-pnorm(1.5, 2,0.6)

# 90th percentile
qnorm(0.9,2,0.6)
```

Normal Distribution in R Video Tutorial

Getting Data into R

Function	Function Values	What does it do?
c($x_1, x_2, \ldots, x_n$)	Where x_1 *to* x_n are variable values.	Makes a vector of values x_1 *to* x_n.
data.frame(v1,v2,v3…)	v1, v2… etc are vector names.	Combine vectors into a dataframe.
read.csv(filename.csv, header = TRUE)	Where filename.csv is either the data set name or file path. header = TRUE, means the first row is variable names. Can use command file.choose() for search window.	Loads .csv data set in R.

```
# c() makes a vector called exam score
examscore = c(79,88,92,87,68)
examscore

# reads in a csv file using a search window and calls data set “data”
data = read.csv(file.choose(), header = TRUE)

# Look at names and first rows of data
head(data)

# $ calls variable from data set
data$state
data$percent
```

R Video Tutorial!

Making a Histogram or Stemplot in R

Function	Function Values	What does it do?
hist(v)	Where v is the quantitative variable of interest. Can obtain from a data set with datasetname$v.	Creates histogram for quantitative variable.
stem(v)	Where v is the quantitative variable of interest. Can obtain from a data set with datasetname$v.	Creates stemplotfor quantitative variable.

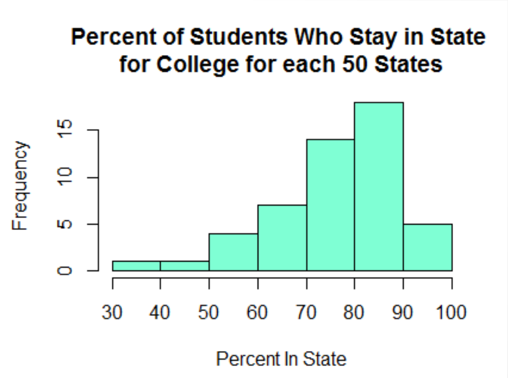
```
# Enter data for percent of students who stay in state for first year of college.
percent = c(83,75,82,70,92,77,69,60,84,83,81,62,68,83,80,80,83,86,63,81,74,91,79,
80,85,73,83,55,81,55,92,82,91,81,85,80,76,73,44,79,68,84,92,76,76,36,84,81,59,63)

# Verify length or sample size.
length(percent)

# Create a histogram
hist(percent)

# create a steplot
stem(percent)

# Create a histogram with some pizzazz!
aka a title, labels and some color.
hist(percent,
     main = "Percent of Students Who Stay in State
for College for each 50 States",
     col = "aquamarine",
     xlab="Percent In State")
```



R Video Tutorial!

Calculating the Mean and Median in R

Function	Function Values	What does it do?
mean(v)	Where v is the quantitative variable of interest. Can obtain from a data set with datasetname$v.	Calculates average for quantitative variable.
median(v)	Where v is the quantitative variable of interest. Can obtain from a data set with datasetname$v.	Calculates median for quantitative variable.

```
# Calculate the Mean
mean(percent)

# Calculate the median
median(percent)
```

R Video Tutorial!

Calculating Spread in R

Function	Function Values	What does it do?
sd(v)	Where v is the quantitative variable of interest. Can obtain from a data set with datasetname$v.	Calculates the sample standard deviation.
quantile(v, p/100)	Where v is the quantitative variable of interest and p is the desired percentile. Can find quartiles with p = 0.25 and p = 0.75.	Calculates the percentile value.

```
# Calculate sample standard deviation
sd(percent)


# Calculate Quartiles with the quantile command.
Q1 = quantile(percent, 0.25)
Q3 = quantile(percent, 0.75)
Q1
Q3

# Calculate IQR using Q1 and Q3 or with the command
IQR = Q3-Q1
IQR

iqr(percent)
```

R Video Tutorial!

Creating a Boxplot in R

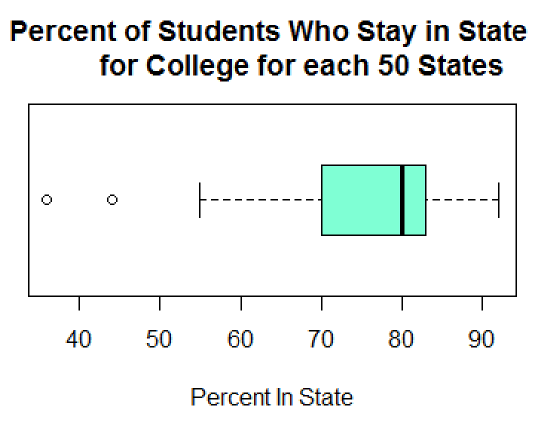
Function	Function Values	What does it do?
summary(v)	Where v is the quantitative variable of interest. Can obtain from a data set with datasetname$v.	Calculates values of the five number summary and mean.
boxplot(v)	Where v is the quantitative variable of interest. Can obtain from a data set with datasetname$v.	Creates boxplot for quantitative variable.

```
# Get all of the summary statistics with summary()
summary(percent)

# Well almost all of them... still need sd()
summary(percent);sd(percent)

# Create a boxplot
boxplot(percent)

# Jazz it up a bit... boxplot with title, color and labels.
# Also now horizontal.
boxplot(percent, main = "Percent of Students Who Stay in State
        for College for each 50 States",
        col = "aquamarine", xlab="Percent In State",
        horizontal = TRUE)
```



R Video Tutorial!

Finding *t* Critical Values

Recall for a $(1-\alpha)100\%\ CI$ the critical value is the $\left(1-\frac{\alpha}{2}\right)^{th}$ percentile.

For 95% CI ***t*** critical value we need to find the 97.5th percentile!

Function	Function Values	What does it do?
qt(percentile,df)	Percentile is area below value. df is degrees of freedom.	Finds a percentile from t_{df} distribution.

```
# Get 95% CI Critical Value for 9 degrees of Freedom
# Must find the 97.5th percentile or 2.5th percentile.
qt(97.5, 9)
```

Steps to Find the critical value with a t table:

1. Find df Row
2. Match df row value with confidence level column

$$t_{9,0.975} = 2.262$$

***t* Table**

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	[illegible]	2.353	3.182	4.541	5.841	10.215	12.924
8	0.000	0.706	0.889	1.108	1.397	[illegible]	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

Finding p-values for *t* test

For a t test the p-value is the area under the curve of a t distribution with a certain degrees of freedom. The area will reflect the alternative hypothesis.

Function	Function Values	What does it do?
pt(t stat,df)	t stat is test statistic and df defines degrees of freedom.	Finds area to the left of t stat from t_{df} distribution.

```
# Get lower one-sided p-value for t stat -2.81 and 9 degrees of Freedom
pt(-2.81, 9)
# Get upper one sided p-value for t stat -2.81 and 9 degrees of Freedom
1-pt(-2.81, 9)
# Get two sided p-value for t stat -2.81 and 9 degrees of Freedom
pt(-2.81, 9)+(1-pt(2.81, 9))
```

Click here to see more on how to find p-values using a t table.

Steps to p-value with a t table:

1. Calculate test statistic and df.
2. Find df row on t table
3. Search for the absolute value of your test stat. Match the columns the t stat is between with one or two tailed p-values at **top**. Your p-value is between these two areas.

***t* Table**

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
8	0.000	0.706	0.889	1.108	1.397	1.860	[illegible]	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
100	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

t Distributions and Procedures in R

Function	Function Values	What does it do?
pt(t,df)	Where t is the test statistic, and df are the degrees of freedom for the t distribution	This is the cumulative density function. Calculates p-value from t tests.
qt(p/100,df)	Where p is percentage that falls below t and df are the degrees of freedom for the t distribution.	Inverse cdf. Calculate for confidence intervals when p/100 is $\frac{\alpha}{2}$ or $1-\frac{\alpha}{2}$
t.test(data, mu = μ_0 , alternative = alt, conf.level = $1-\alpha$)	Where data is sampled data, μ_0 is the claim, alt is either “two.sided”, “less”, “greater”, conf.level is $1-\alpha$.	Gives results for one sample t test and confidence interval for specified claim, alternative and α level.

```
pt(-2.81, 9) # lower one sided p-value
qt(0.975, 9) # 95% t critical value for df = 9

mileage = c(11601,8987,12166,9657,10143,8230,3111,13009,7891,10392)
t.test(mileage, mu = 12000, alternative = "two.sided", conf.level = 0.95)
#See help(t.test) for more options.
```

T test R Video Tutorial Click Here!

Two Sample t Test in R

Function	Function Values	What does it do?
t.test(data1, data2, mu = δ_0 , alternative = alt, conf.level = $1-\alpha$)	Where data1 and data2 are sampled data, δ_0 is the claim, alt is either "two.sided", "less", "greater", conf.level is $1-\alpha$.	Gives results for two sample t test and confidence interval for specified claim, alternative and α level.

```
# Read in the 30 observations for each server
ServerA=c(6.66,6.11,6.83,7.06,5.84,6.65,6.43,7.40,6.74,6.92,6.08,6.31,6.23,6.92,
6.62,6.96,5.69,6.22,6.12,6.59,6.65,7.35,7.62,6.21,6.29,6.27,6.92,6.20,5.52,6.29)

ServerB=c(6.23,6.77,6.94,8.15,8.67,8.71,6.08,6.25,7.70,5.33,7.26,5.49,6.83,7.32,
9.64,7.66,7.99,8.37,8.26,6.60,8.67,5.55,7.74,7.91,8.15,5.31,7.94,4.90,7.52,10.25)

# t.test() calculates the test but also a confidence interval with $conf.int
t.test(ServerA, ServerB, conf.level =0.95)$conf.int

# t.test() calculates a lower one sided two sample t test.
t.test(ServerA, ServerB, mu = 0, alternative = "less", conf.level =0.95)
```

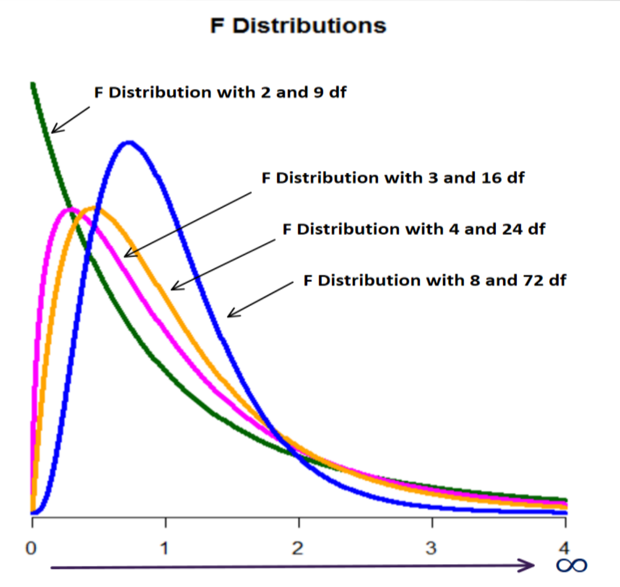
Two Sample t test

R Video Tutorial

F Distributions

F Distribution

- Positively skewed distribution ranging from 0 to ∞.
- Shape is defined by numerator and denominator degrees of freedom, $v_1, and\ v_2$ denoted as F_{v_1,v_2}
- As degrees of freedom increase distribution becomes more symmetric.
- A ratio of chi-square distributions with corresponding degrees of freedom.



For an Single Factor ANOVA F test the p-value is the area under to the **right** of the F test statistic in the distribution $F_{I-1,I(J-1)}$.

Function	Function Values	What does it do?
pf(f,v1, v2)	Where f is test statistic, v1 is numerator degrees and v2 is denominator degrees of freedom for an f distribution	Gives values for cumulative density function. For p-values subtract from 1.

```
# Get the p-value for an F stat of 12.097 with v1 = 3 and and v2 = 16 degrees of Freedom
1-pt(12.097,3,16)
```

Using R for ANOVA

Function	Function Values	What does it do?
mod = aov(response~treatment) summary(mod)	Where response is the quantitative response and treatment is the levels of the single factor.	Provides an ANOVA table and results from a single factor ANOVA F test.

```
# Data must be in two columns or vectors, a quantitative vector and a corresponding
categorical vector that defines the groups.
carbon = c(7.75, 7.00, 7.1, 7.50, 7.9,  6.90, 7.60, 7.20, 7.00, 7.70,
           6.70, 6.90, 7.10, 6.30, 6.60,  6.40, 6.20 ,6.00 ,6.50, 6.60)
# Use the rep() command to repeat a value multiple times.
temp.factor = c(rep("300",5), rep("400",5), rep("500", 5), rep("600", 5))
# Use aov() to perform ANOVA call it something, I use "mod" take the summary() to
get ANOVA table.
mod = aov(carbon~temp.factor)
summary(mod)
```

```
              Df  Sum Sq  Mean Sq  F value    Pr(>F)
temp.factor    3   3.919    1.306     12.1  0.000219 ***
Residuals     16   1.728    0.108
```

ANOVA R Tutorial Video

Using R for Multiple Comparisons

Function	Function Values	What does it do?
TukeyHSD(mod, conf.level =0.95)	Mod is the result from aov(response~treatment), conf.level = (0.90, 0.95, 0.99).	Provides simultaneous comparison for means. Gives Difference, CI and adjusted p-value for the desire family wise error rate

```
# Data must be in two columns or vectors, a quantitative vector and a corresponding categorical vector that defines the groups.
carbon = c(7.75, 7.00, 7.1, 7.50, 7.9,  6.90, 7.60, 7.20, 7.00, 7.70,
           6.70, 6.90, 7.10, 6.30, 6.60,  6.40, 6.20 ,6.00 ,6.50, 6.60)
temp.factor = c(rep("300",5), rep("400",5), rep("500", 5), rep("600", 5))
mod = aov(carbon~temp.factor)
summary(mod)
# Perform Multiple Comparisons Procedure at 0.05 family-wise significance
TukeyHSD(mod, conf.level = 0.95)
# Get a plot of the confidence intervals for each comparison
plot(TukeyHSD(mod, conf.level = 0.95))
```

ANOVA R Tutorial Video

Scatterplot and Correlation in R

Function	Function Values	What does it do?
plot(x,y)	Where x is the explanatory variable and y is the response variable. Add ons like main = “”, xlab =“”, and ylab = “” are for titles. pch = , changes the point character, col = “” changes the color of the points.	Plots pairs of quantitative data. Makes a scatterplot.
cor(x,y)	Where x and y are two quantitative variables.	Calculates correlation coefficient r.

```
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
#Basic Scatterplot
plot(HSGPA, CollegeGPA)

#Add titles, labels and color
plot(HSGPA, CollegeGPA, main = "Graduating HighSchool GPA vs Freshman Year
College GPA", col = "darkgreen", pch = 16, xlab = "High School GPA“, ylab
= "College GPA")

# Calculate Correlation Coefficient
cor(HSGPA, CollegeGPA)
```

R Tutorial Video

Calculating the LSRL in R

Function	Function Values	What does it do?
mod = lm(y~x) summary(mod)	Where x is the explanatory variable and y is the response variable. Function lm() is for linear model. Expression called "mod" take summary of "mod".	lm() computes the linear model. Summary outputs it into a nice format. Gives the LSRL estimates, standard errors and performs a t test on the slope. Provides R^2 and adjusted R^2value and results from an model utility F test.

```
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)

# Use the lm() command to get the linear model.
# Take the summary() of lm() to get more information.
mod = lm(CollegeGPA~HSGPA)
summary(mod)
```

R Tutorial Video

Calculating and Plotting Residuals in R

Function	Function Values	What does it do?
mod = lm(y~x) summary(mod)	Where x is the explanatory variable and y is the response variable. Function lm() is for linear model. Expression called "mod" take summary of "mod".	lm() computes the linear model. Summary outputs it into a nice format. Gives the LSRL estimates, standard errors and performs a t test on the slope. Provides R^2 and adjusted R^2value and results from an model utility F test.
mod$residuals	"mod" is the name of your linear model, $residuals extracts residual values from linear model.	Calculates residuals = $y - \hat{y}$

R Tutorial Video

```
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)

mod = lm(CollegeGPA~HSGPA)
summary(mod)

plot(HSGPA, mod$residuals) # plots residuals against x
abline(h=0, lty =2) # adds a reference line at 0.
```

t confidence interval for β_1 in R

Function	Function Values	What does it do?
mod = lm(y~x) summary(mod)	Where x is the explanatory variable and y is the response variable. Function lm() is for linear model. Expression called “mod” take summary of “mod”.	lm() computes the linear model. Summary outputs it into a nice format. Gives the LSRL estimates, standard errors and performs a t test on the slope. Provides R^2 and adjusted R^2value and results from an model utility F test.
confint(mod, conf.level = 0.95)	“mod” is the name of the linear model. conf.level defines the level of confidence	Calculates the confidence intervals for the regression coefficients for the linear model.

R Tutorial Video

```
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
mod = lm(CollegeGPA~HSGPA)
summary(mod)

# Use the outcome from the model lm() to calculate a 95%CI for B0 and b1
confint(mod, conf.level = 0.95)
```

t Test for the Slope β_1 in R

Function	Function Values	What does it do?
mod = lm(y~x) summary(mod)	Where x is the explanatory variable and y is the response variable. Function lm() is for linear model. Expression called "mod" take summary of "mod".	lm() computes the linear model. Summary outputs it into a nice format. Gives the LSRL estimates, standard errors and **performs a t test** on the slope. Provides R^2 and adjusted R^2value and results from an model utility F test.

```
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)

# the lm() command provides the t test statistic and p-value for the
two sided test on the slope.
mod = lm(CollegeGPA~HSGPA)
summary(mod)
```

R Tutorial Video

Prediction for the Response in R

Function	Function Values	What does it do?
mod = lm(y~x) summary(mod)	Where x is the explanatory variable and y is the response variable. Function lm() is for linear model. Expression called “mod” take summary of “mod”.	lm() computes the linear model.
predict(mod, data.frame(x = c(xnew), conf.level = 0.95, interval ="confidence")	“mod” linear model name, conf.level defines level of confidence. Define "confidence" or "Prediction" for type of interval, must put new value in as data.frame(), where xnew is new value, x is explanatory variable name.	Calculates the confidence or prediction interval of response for a value for the explanatory variable.

```
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
mod = lm(CollegeGPA~HSGPA)
summary(mod)
#Conf. Interval for average college GPA of all students with a GPA of 3.6
predict(mod, data.frame(HSGPA= c(3.6)), conf.level = 0.95,
interval="confidence")
#Pred. Interval for college GPA of a single student with a GPA of 3.6
predict(mod, data.frame(HSGPA= c(3.6)), conf.level = 0.95,
interval="prediction")
```

R Tutorial Video

Creating a Scatterplot Matrix in R

Function	Function Values	What does it do?
pairs(y~x_1+x_2+…+x_k)	Y, x_1, x_2, x_k are names for the response variable and explanatory variables.	Creates a scatterplot matrix of the variables.

```
# Enter the data into R.
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
SAT = c(1810, 1700, 2100, 1900, 2190, 2310, 1790, 1900, 2100, 2060)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)

# Data must be in a dataframe. To create scatterplot matrix with pairs()
pairs(CollegeGPA~HSGPA+SAT)
```

Multivariate Data Visualization R Tutorial

Multivariate Regression Equation in R

Function	Function Values	What does it do?
mod = lm(y~x_1+x_2+…+x_k) summary(mod)	Where x's are explanatory variables and y is the response variable. lm() is for linear model. Expression called "mod". Take summary of "mod".	Summary provides LSRE estimates, standard errors and performs a t test on the slope, R^2, adj R^2 and results from an model utility F test.

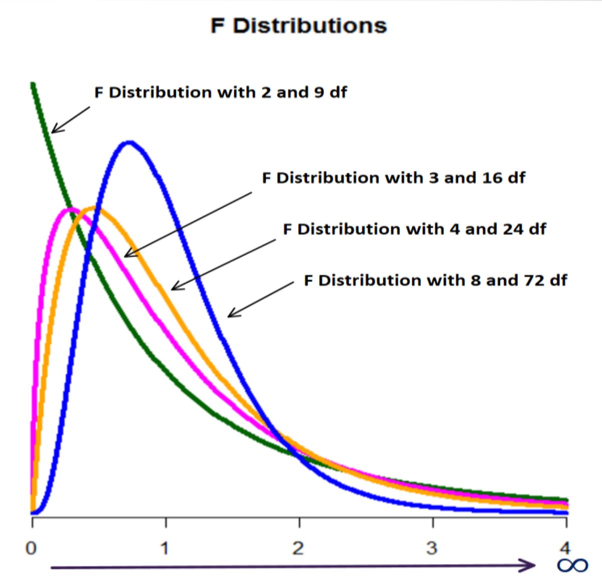
```
# Enter in the data in vectors like the GPA example her or from a file.
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
SAT = c(1810, 1700, 2100, 1900, 2190, 2310, 1790, 1900, 2100, 2060)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
# Create the linear model.
# This also provides the R^2 value.
mod = lm(CollegeGPA ~ HSGPA+SAT)
summary(mod)
```

MLR in R Video Tutorial

F Distributions and F test P-values

F Distribution

- Positively skewed distribution ranging from 0 to ∞.
- Shape is defined by numerator and denominator degrees of freedom, $v_1, and\ v_2$ denoted as F_{v_1,v_2}
- As degrees of freedom increase distribution becomes more symmetric.
- A ratio of chi-square distributions with corresponding degrees of freedom.



For an Model Utility F test the p-value is the area under to the **right** of the F test statistic in the distribution $F_{k,n-(k+1)}$.

Function	Function Values	What does it do?
pf(f,v1, v2)	Where f is test statistic, v1 is numerator degrees and v2 is denominator degrees of freedom for an f distribution	Gives values for cumulative density function. For p-values subtract from 1.

```
# Get the p-value for an F stat of 140.4 with v1 = 3 and and v2 = 137 degrees of Freedom
1-pf(140.4,3,137)
```

Model Utility F Test in R

Function	Function Values	What does it do?
mod = lm(y~x_1+x_2+…+x_k) summary(mod)	Where x's are explanatory variables and y is the response variable. lm() is for linear model. Expression called "mod". Take summary of "mod".	Summary provides LSRE estimates, standard errors and performs a t test on the slope, R^2, adj R^2 and results from an model utility F test.

```
# Enter in the data in vectors like the GPA example her or from a file.
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
SAT = c(1810, 1700, 2100, 1900, 2190, 2310, 1790, 1900, 2100, 2060)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
# Create the linear model.
# Provides F test
mod = lm(CollegeGPA ~ HSGPA+SAT)
summary(mod)
```

MLR in R Video Tutorial

Individual *t* Tests for β_j in R

Function	Function Values	What does it do?
mod = lm(y~x_1+x_2+…+x_k) summary(mod)	Where x's are explanatory variables and y is the response variable. lm() is for linear model. Expression called "mod". Take summary of "mod".	Summary provides LSRE estimates, standard errors and performs a t test on the slope, R^2, adj R^2 and results from an model utility F test.

```
# Enter in the data in vectors like the GPA example her or from a file.
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
SAT = c(1810, 1700, 2100, 1900, 2190, 2310, 1790, 1900, 2100, 2060)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
# Create the linear model.
# Provides F test
mod = lm(CollegeGPA ~ HSGPA+SAT)
summary(mod)
```

MLR in R Video Tutorial

Residuals in R

Function	Function Values	What does it do?
mod = lm(y~x_1+x_2+…+x_k) summary(mod)	Where x's are explanatory variables and y is the response variable. lm() is for linear model. Expression called "mod". Take summary of "mod".	Summary provides LSRE estimates, standard errors and performs a t test on the slope, R^2, adj R^2 and results from an model utility F test.
mod$residuals	"mod" is the name of your linear model, $residuals extracts residual values from linear model.	Calculates residuals = $y - \hat{y}$

```
# Enter in the data in vectors like the GPA example her or from a file.
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
SAT = c(1810, 1700, 2100, 1900, 2190, 2310, 1790, 1900, 2100, 2060)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
# Create the linear model.
mod = lm(CollegeGPA ~ HSGPA+SAT)
summary(mod)
# plot the residuals and fitted values. Put a line at 0.
plot(mod$fitted, mod$resdiuals)
abline(h = 0, lty = 2)
```

MLR in R Video Tutorial

t Confidence Intervals for slopes in R

Function	Function Values	What does it do?
mod = lm(y~x_1+x_2+…+x_k) summary(mod)	Where x's are explanatory variables and y is the response variable. lm() is for linear model. Expression called "mod". Take summary of "mod".	Summary provides LSRE estimates, standard errors and performs a t test on the slope, R^2, adj R^2 and results from an model utility F test.
conf.int(mod,conf.level = 0.95)	"mod" is the name of the linear model. conf.level defines the level of confidence	Calculates the confidence intervals for the regression coefficients for the linear model.

```
# Enter in the data in vectors like the GPA example her or from a file.
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
SAT = c(1810, 1700, 2100, 1900, 2190, 2310, 1790, 1900, 2100, 2060)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
# Create the linear model.
mod = lm(CollegeGPA ~ HSGPA+SAT)
summary(mod)
# Calcuates confidence intervals for B0, B1… BK
conf.int(mod, conf.level = 0.95)
```

MLR in R Video Tutorial

Prediction for the Response in R

Function	Function Values	What does it do?
mod = lm(y~x_1+x_2+…+x_k) summary(mod)	Where x's are explanatory variables and y is the response variable. lm() is for linear model. Expression called "mod". Take summary of "mod".	Summary provides LSRE estimates, standard errors and performs a t test on the slope, R^2, adj R^2 and results from an model utility F test.
predict(mod,data.frame (x1 = x1*, x2 = x2*), conf.level = 0.95, interval ="confidence")	mod is name of the linear model, conf.level defines confidence level, define "confidence" or "prediction" for type of interval, must put new values in as data.frame()	Calculates the confidence or prediction interval of a response for a new set of the explanatory variable values.

```
# Enter in the data in vectors like the GPA example her or from a file.
HSGPA = c(3.4, 3.1, 3.7,3.3, 3.5, 4, 3.2, 3.8, 3.9, 3.6)
SAT = c(1810, 1700, 2100, 1900, 2190, 2310, 1790, 1900, 2100, 2060)
CollegeGPA = c(3.1,3,3.6,3.6,3.4,3.9,3.2,3.7,3.6, 3.3)
# Create the linear model.
mod = lm(CollegeGPA ~ HSGPA+SAT)
summary(mod)
# Use predict command. Change interval to "prediction" for PI
predict(mod, data.frame(HSGPA = 3.6, SAT = 1900), interval = "confidence")
```

MLR in R Video Tutorial

Using Software

Function	Function Values	What does it do?
qcc(data, type ="xbar", std.dev = "UWAVE-SD")	Where data is your dataset, “xbar” is the type of control chart can also be “S” or “R”. If it is an xbar chart specify "UWAVE-SD" or "UWAVE-R" represents how to calculate the variation either with standard deviation or ranges.	Creates an xbar-s chart or an xbar – R chart, an S chart or an R chart .

```
#Install the package qcc
install.packages("qcc")
#Open the commands from the “qcc” library.
library("qcc")

#Attach dataset pistonrings use help(pistonrings) to learn more
data(pistonrings)
attach(pistonrings) #Attach the dataset
diameter <- qcc.groups(diameter, sample) #Format the dataset

qcc(diameter, type="xbar", std.dev = "UWAVE-SD") # Create x bar s chart
qcc(diameter, type="xbar", std.dev = "UWAVE-R") # Create x bar R chart
qcc(diameter, type="S") # Create S chart
qcc(diameter, type="R") # Create R chart
```

Control Chart R Tutorial Video

Using Software

Function	Function Values	What does it do?
qcc(data, type = "p", size = n)	Where data is your dataset, "p" is the type of control chart and n is your subgroup sample size (must be the same for all k).	Creates a p chart.
qcc(data, type = "c")	Where data is your dataset, "c" is the type of control chart	Creates a c chart.

```
#Install the package qcc and call the library qcc.
install.packages("qcc")
library("qcc")

# p chart
p.data = c(20,10,15,18,19, 20, 17, 16, 14,29,16,17)
n = 100
qcc(p.data, type="p", size = 100)

#c chart
c.data = c(18,12,14,13,19,22,25,24,29,27,24,28)
qcc(c.data, type="c")
```

Control Chart R Tutorial Video