R Commands Cheat Sheet for an Introduction to Statistics for Engineers, Oregon State University, by Katie Jager

Probability Distributions 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)`	"	This is the cumulative distribution function. Finds P(X<=x) for binomial distribution
`dpois(x,lambda)`	Where x is the value of interest, average lambda is the rate of success per unit	This is the probability mass function. Finds P(X=x) for poisson distribution
`ppois(x,lambda)`	"	This is the cumulative distribution function. Finds P(X<=x) for poisson distribution
`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≤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≤x_p)=p
`pgamma(x,shape, rate)`	Where x is the value of interest, shape = α and rate = 1/β	Cumulative density function, finds P(X≤x) for gamma distribution
`qgamma(p,shape, rate)`	Where p is percentage that falls below x, shape = α and rate = 1/β	Inverse of cumulative density function.
`pexp(x,lambda)`	Where x is the value of interest, λ is the rate parameter	Cumulative density function, finds P(X≤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.
`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≤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(X≤x_p)=p

Entering Data Function	Function Values	What does it do?
`c(x_1,x_2,…,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 data frame.
`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.

Basic Visualization and Summary commands 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)`	""	Creates stemplot for quantitative variable.
`mean(v)`	""	Calculates average for quantitative variable.
`median(v)`	""	Calculates median for quantitative variable.
`sd(v)`	""	Calculates the sample standard deviation.
`summary(v)`	""	Calculates values of the five number summary and mean.
`boxplot(v)`	""	Creates boxplot for quantitative variable.
`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.

t Distribution and Commands 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 α/2 or 1-α/2
`t.test(data, mu = μ_0 , alternative = alt, conf.level = 1-α)`	Where data is sampled data, μ_0 is the claim, alt is either “two.sided”, “less”, “greater”, conf.level is 1-α.	Gives results for one sample t test and confidence interval for specified claim, alternative and α level.
`t.test(data1, data2, mu = δ_0 , alternative = alt, conf.level = 1-α)`	Where data1 and data2 are sampled data, δ_0is the claim, alt is either “two.sided”, “less”, “greater”, conf.level is 1-α.	Gives results for two sample t test and confidence interval for specified claim, alternative and α level.

f Distribution and ANOVA 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.
`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.
`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

Bivariate Regression and Visualization		
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.
`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}$
`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.
`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.

Multivariate Regression and Visualization		
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.
`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.
`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.

Control Charts Function	Must install qcc package 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.
`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.