

Util Functions

<code>getwd()</code>	gets the working directory
<code>setwd("C:/file/path")</code>	sets the working directory
<code>data = read.csv(file.choose())</code>	opens file explorer to get data
<code>ls()</code>	lists the variables
<code>str(var)</code>	structure of the variable
<code>rm(var)</code>	removes the variable
<code>help.start()</code>	opens help
<code>install.packages("package name")</code>	installs the package
<code>library("package name")</code>	makes the contents available to use
<code>detach("package name")</code>	detaches the package

Strings

<code>toString(x)</code>	helper function to produce a single character string
<code>toupper()/tolower()</code>	converts text to upper/lower case
<code>substring(chr,n,n)</code>	retrieve or replaces the substring of the character
<code>paste(..., sep = " ", collapse = NULL)</code>	Concatenate vectors after converting to character

Arrays and Matrix

<code>1D = array(1:24)</code>	1 dimensional array
<code>2D = array(1:24, dim = c(6,4))</code>	2 dimensional array
<code>3D = array(1:24, dim = c(4,3,2))</code>	3 dimensional array
<code>mat = matrix(1:12, nrow=4, ncol=3)</code>	matrix
<code>cbind(mat1,mat2)</code>	column bind
<code>rbind(mat1,mat2)</code>	row bind

Vector

<code>num = c(1,2,3,4,5,6)</code>	numeric vector
<code>chr = c("aaa","bbb")</code>	character vector
<code>log = c(TRUE,TRUE,FALSE)</code>	logical vector
<code>mean(vec)</code>	mean
<code>sd(vec)</code>	standard deviation
<code>var(vec)</code>	variance
<code>range(vec)</code>	range
<code>which.min(vec)/which.max(vec)</code>	position of the min/max value
<code>rep(1:5,times=3)</code>	replicate elements of vector

DataFrame

<code>df = data.frame(subjectID=1:5,gender=c("M","F","M","M","F"),score=c(8,3,6,5,5))</code>	dataframe
<code>view(df)</code>	opens editor
<code>head(df)/tail(df)</code>	displays top/bottom n rows
<code>summary(df)</code>	returns descriptive statistics of data

Descriptive Statistics

<code>rowMeans(data[])</code>	row mean
<code>rowSums(data[])</code>	row sum
<code>colMeans(data[])</code>	column mean
<code>colSums(data[])</code>	column sum

Loops

<code>for (variable in sequence){ Do something }</code>	for loop
<code>while (condition){ Do something }</code>	while loop
<code>if (condition){ Do something } else { Do something different }</code>	ifelse statement

Hypothesis

<code>t.test(data)</code>	1 sample t test
<code>t.test(data1,data2)</code>	2 sample t test
<code>t.test(pre,post,paired=TRUE)</code>	paired sample t test
<code>wilcox.test(data)</code>	Wilcox test
<code>cor.test(data1,data2)</code>	correlation test
<code>chisq.test(data)</code>	Chi square test
<code>shapiro.test(data)</code>	Shapiro test
<code>aov()</code>	ANOVA

Visualization

<code>qplot(data, line=TRUE,...)</code>	produces quantile-quantile plot
<code>ggplot(data = NULL, mapping = aes(), ...)</code>	initializes a ggplot object
<code>geom_bar()</code>	bar graph
<code>coord_flip()</code>	flip x and y coordinates
<code>facet_grid()</code>	lay out panels in a grid
<code>geom_density</code>	density plot
<code>geom_hist</code>	histogram
<code>geom_point</code>	scatter plots

Probability

<code>rbinom(n, size, prob)</code>	Binomial distribution
<code>rpois(n,size)</code>	Poisson distribution
<code>runif(n, min = 0, max = 1)</code>	Uniform distribution
<code>rnorm(n,mean,sd)</code>	Normal distribution
<code>rexp(n)</code>	Exponential distribution

Statistics

<code>summary(lm(y ~ x1 + x2 + x3, data=mydata))</code>	multiple regression
<code>summary(glm(y ~ x1 + x2 + x3, family="", data=mydata))</code>	classification
<code>cluster = kmeans(data)</code>	kmeans cluster analysis

