**INSTALLING R AND RSTUDIO & START AN R-SCRIPT FILE**

Download and install R from: <https://cran.r-project.org/bin/windows/base/>

Download and install RStudio from: [www.rstudio.com/products/rstudio/download](http://www.rstudio.com/products/rstudio/download)

Install packages:

“ggplot2”: run “install.packages(“ggplot2”)” at the command line in console pane

How to load a package after installation: use command “library(“package name”)”. For example, library(“ggplot2”). Noted that R will unload all of its packages when you close RStudio!

How to install multiple packages: install.packages(c("ggplot2", "reshape2", "dplyr"))

How to check installed packages: library()

**Start RStudio**

Open RStudio and change working environment to “C:\Users\<User Name>\Documents” (using menu “Session”)

Start a new R script file by: File => New File => R script and then save as “My First R session.R”. The script can be run in the console pane by “Ctrl + Enter”

Example: write in the script pane: x <- c(1,2,3), press Ctrl+Enter (or Ctrl+R) depending on your computer) and see what happen in the console pane (type “x” in the console pane to see the result)

**BASIC STATISTICS**

1. **Find the sum of numbers in a vector**

Command: > sum(x)

1. **Find the average of numbers in a vector**

Command: > mean(x)

1. **Find the median of numbers in a vector**

Command: > median(x)

1. **Find the mode value of numbers in a vector**

There is no built-in function in base R to help find the mode!

First, let install the library “modeest”: run “install.packages(“modeest”)” at the command line in console pane and then load it using “library(modeest)” (or on the Packages tab, select Install, and then in the Install dialog box, type modeest in the Packages box and click Install. Then check its check box when it appears on the Packages tab to load it).

Command: > mfv(x) (mfv: most frequent value)

1. **Find the z-scores**

*Direct calculation*:

Example:

Command: > x = sample(1:50, 100, replace = T)

> m = mean(x)

> s = sd(x)

> zs = (x-m)/s

(> summary(x)): see what happens!

(> summary(zs)): see what happens!

Noted that for population: s = sd(x)\*sqrt((length(x)-1)/length(x))

*Using scale*() *command*:

Example:

Command: > scale(x)

Result:

[,1]

[1,] -1.39404395

[2,] 0.47233827

[3,] -1.60939574

…

[98,] -1.10690822

[99,] 0.97482579

[100,] 0.04163468

attr(,"scaled:center")

[1] 23.42 This value is mean(x)

attr(,"scaled:scale")

[1] 13.93069 This value is sd(x)

To show the results as a row vector:

> sc\_zs = scale(x)[,1]

(> summary(sc\_zs)): see what happens!

Noted that for population: each element in the vector scale(x) should be divided by sqrt((length(x)-1)/length(x))

> scale(x)/sqrt((length(x)-1)/length(x))

Result:

[,1]

[1,] -1.40106688

[2,] 0.47471782

[3,] -1.61750358

…

[97,] 0.69115452

[98,] -1.11248462

[99,] 0.97973678

[100,] 0.04184443

attr(,"scaled:center")

[1] 23.42

attr(,"scaled:scale")

[1] 13.93069

1. **Find the percentiles and quartiles**

There are 9 methods to determine quantile, the default is Type 7!

Example 1:

Command: > x <- c(1,4,2,3,5,6)

> quantile(x, c(0.25, 0.35, 0.55))

Result: 25% 35% 55%

2.25 2.75 3.75

Example 2:

Command: > x <- c(1,4,2,3,5,6)

> quantile(x)

Result: 0% 25% 50% 75% 100%

1.00 2.25 3.50 4.75 6.00

Example 3:

Command: > x <- c(1,4,2,3,5,6)

> quantile(x, probs = seq(0,1,0.1))

Result: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0

Noted that > seq(0,1,0.1) will give: 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

1. **Find the range**

Command: > max(x) – min(x)

1. **Find the variance of sample**

Command: > var(x)

1. **Find the variance of population**

Command: > n=length(x); var(x)\*(n-1)/n (n can be replaced by length(x) directly in the formula)

or > mean((x-mean(x))^2)

Note: If n=1 the first formula yield 0 while the second formula yields N/A

1. **Find the standard deviation of sample**

Command: > sd(x)

1. **Find the standard deviation of population**

Command: > sd(x)\*sqrt((length(x)-1)/length(x))