RDojo 01/18/2018 - Introduction to R

0: Some Preliminaries

0a: Working Directories

```
getwd() #see wd
rm(list = ls()) #clears EVERYTHING
setwd("~/Desktop/Dropbox/blah") #set wd
```

0b: Installing packages

If a package has not already been installed (through R) on the computer you are using, you will need to use install.packages("nameinquotes").

```
install.packages("name")
```

0c: Using installed packages

library(namenotinquotes) must be used every time you open a new session in RStudio; equivalently, you can use the "packages" section of the bottom right box to check the box next to that package once it has been installed.

```
library(name)
```

0d: Datasets included with base R

R includes a number of datasets useful for playing around and learning how R works. We will be using a number of these today. Read these datasets in using the data(X) function.

```
data(iris)
data(mtcars)
```

0e: Function information

If you don't know how to use a function, or just want to know more about how it works, you can use ?functionname ***without*** parentheses to pull up a help box in the bottom right toolpane.

```
?View
```

0f: Function code

If you want to see the actual code that R uses when it runs a function, type the function alone, ***without*** parentheses or a question mark.

```
View
```

```
## function (x, title)
## {
##     check <- Sys.getenv("_R_CHECK_SCREEN_DEVICE_", "")
##     msg <- "View() should not be used in examples etc"
##     if (identical(check, "stop"))
##         stop(msg, domain = NA)
##     else if (identical(check, "warn"))
##         warning(msg, immediate. = TRUE, noBreaks. = TRUE, domain = NA)
##     if (missing(title))
##         title <- paste("Data:", deparse(substitute(x))[1])
##     as.num.or.char <- function(x) {
##         if (is.character(x))
##             x
##         else if (is.numeric(x)) {
##             storage.mode(x) <- "double"
##             x
##         }
##         else as.character(x)
##     }
##     x0 <- as.data.frame(x)
##     x <- as.list(format.data.frame(x0))
##     rn <- row.names(x0)
##     if (any(rn != seq_along(rn)))
##         x <- c(list(row.names = rn), x)
##     if (!is.list(x) || !length(x) || !all(sapply(x, is.atomic)) ||
##         !max(lengths(x)))
##         stop("invalid 'x' argument")
##     if (grepl("darwin", R.version$os))
##         check_for_XQuartz()
##     invisible(.External2(C_dataviewer, x, title))
## }
## <bytecode: 0x7fdb7b05fe28>
## <environment: namespace:utils>
```

0g: Non-code comments

Use # to comment out parts of code so that they don't run

```
data(iris) #this is a comment
```

1: Importing data

1a: read.table()/read.csv() - read in general data files

read.table() and read.csv() read in general data files.

```
data1 <- read.table(file.choose())
```

-file.choose() pulls up a drop-down menu to choose the file manually, instead of using the file path.
-The argument "sep =" tells the function what symbol is used to denote a new cell. For csv's ("comma-separated files"), this is a comma, so read.csv(file.choose()) and read.table(file.choose() , sep = ",") are equivalent in that sense.
-The argument "header = TRUE/FALSE" tells the function to either register the first row of the dataset as a

row of column names (TRUE) or not (FALSE).
-The agument "na.rm = TRUE/FALSE" tells the function remove (TRUE) or keep (FALSE) rows that contain ***any*** missing values.

1b: read.spss - read in SPSS files

read.spss() reads in SPSS files. this function requires the "foreign" package.

```
install.packages("foreign")
library(foreign)
data1 <- read.spss(file.choose())
```

2: Functions for basic info about data

2a: View()

View() shows the chosen data or object in a separate tab

```
View(mtcars)
```

2b: head()

head() shows the first 6 observations of an object

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

2c: str()

str() gives a summary of the structure of an object

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

2d: summary()

summary() gives various descriptives about all of the variables in an object

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

3: dplyr package intro

3.0: Install/load dplyr

```
install.packages("dplyr")
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

3a: filter()

filter() selects rows meeting certain conditions
The following function creates a dataset with all observations from mtcars with 6 cylinders and 4 gears.

```
cyl6.gear4 <- filter(mtcars, cyl == 6, gear == 4)
```

The following function creates a dataset with all observations from mtcars with 6 or more cylinders and less than 4 gears.

```
cyl6p.gearLT4 <- filter(mtcars, cyl >= 6 , gear < 4)
```

3b: arrange()

arrange() arranges dataset by values of particular columns

```
arrange(mtcars, cyl, disp) #arranges and prints by cyl, and then by disp within each level of cyl,
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1  33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## 2  30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## 3  32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## 4  27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## 5  30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## 6  22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 7  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 8  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 9  21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## 10 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## 11 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## 12 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## 13 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 14 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 15 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 16 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## 17 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## 18 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 19 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## 20 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## 21 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## 22 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## 23 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## 24 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## 25 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## 26 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 27 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 28 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## 29 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## 30 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## 31 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## 32 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
```

```
#each in ascending order
arrange(mtcars, cyl, desc(disp)) #same as above, except disp arrangement within cyl levels
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1  24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## 2  22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## 3  21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## 4  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 5  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 6  22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
```

```
## 7  30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## 8  27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## 9  32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## 10 30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## 11 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## 12 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 13 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## 14 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 15 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## 16 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 17 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 18 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## 19 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## 20 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## 21 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## 22 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## 23 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 24 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## 25 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 26 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## 27 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## 28 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## 29 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## 30 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## 31 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## 32 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
```

```
#is in descending order
cyldisp.ord <- arrange(mtcars, cyl, disp) #creates new arranged dataset "cyldisp.ord"
```

3c: select()

select() allows you to select particular columns of dataset without the others, while also renaming the selected columns (if desired).

```
select(iris , Petal.Length) #selects and prints only the variable "Petal.Length" from the dataset,
```

```
##     Petal.Length
## 1            1.4
## 2            1.4
## 3            1.3
## 4            1.5
## 5            1.4
## 6            1.7
## 7            1.4
## 8            1.5
## 9            1.4
## 10           1.5
## 11           1.5
## 12           1.6
## 13           1.4
## 14           1.1
## 15           1.2
## 16           1.5
```

```
## 17          1.3
## 18          1.4
## 19          1.7
## 20          1.5
## 21          1.7
## 22          1.5
## 23          1.0
## 24          1.7
## 25          1.9
## 26          1.6
## 27          1.6
## 28          1.5
## 29          1.4
## 30          1.6
## 31          1.6
## 32          1.5
## 33          1.5
## 34          1.4
## 35          1.5
## 36          1.2
## 37          1.3
## 38          1.4
## 39          1.3
## 40          1.5
## 41          1.3
## 42          1.3
## 43          1.3
## 44          1.6
## 45          1.9
## 46          1.4
## 47          1.6
## 48          1.4
## 49          1.5
## 50          1.4
## 51          4.7
## 52          4.5
## 53          4.9
## 54          4.0
## 55          4.6
## 56          4.5
## 57          4.7
## 58          3.3
## 59          4.6
## 60          3.9
## 61          3.5
## 62          4.2
## 63          4.0
## 64          4.7
## 65          3.6
## 66          4.4
## 67          4.5
## 68          4.1
## 69          4.5
## 70          3.9
```

```
## 71          4.8
## 72          4.0
## 73          4.9
## 74          4.7
## 75          4.3
## 76          4.4
## 77          4.8
## 78          5.0
## 79          4.5
## 80          3.5
## 81          3.8
## 82          3.7
## 83          3.9
## 84          5.1
## 85          4.5
## 86          4.5
## 87          4.7
## 88          4.4
## 89          4.1
## 90          4.0
## 91          4.4
## 92          4.6
## 93          4.0
## 94          3.3
## 95          4.2
## 96          4.2
## 97          4.2
## 98          4.3
## 99          3.0
## 100         4.1
## 101         6.0
## 102         5.1
## 103         5.9
## 104         5.6
## 105         5.8
## 106         6.6
## 107         4.5
## 108         6.3
## 109         5.8
## 110         6.1
## 111         5.1
## 112         5.3
## 113         5.5
## 114         5.0
## 115         5.1
## 116         5.3
## 117         5.5
## 118         6.7
## 119         6.9
## 120         5.0
## 121         5.7
## 122         4.9
## 123         6.7
## 124         4.9
```

```
## 125          5.7
## 126          6.0
## 127          4.8
## 128          4.9
## 129          5.6
## 130          5.8
## 131          6.1
## 132          6.4
## 133          5.6
## 134          5.1
## 135          5.6
## 136          6.1
## 137          5.6
## 138          5.5
## 139          4.8
## 140          5.4
## 141          5.6
## 142          5.1
## 143          5.1
## 144          5.9
## 145          5.7
## 146          5.2
## 147          5.0
## 148          5.2
## 149          5.4
## 150          5.1
```

```
#without renaming it
select(iris, petal_length = Petal.Length) #selects and prints original "Petal.Length" as "petal_length"
```

```
##     petal_length
## 1            1.4
## 2            1.4
## 3            1.3
## 4            1.5
## 5            1.4
## 6            1.7
## 7            1.4
## 8            1.5
## 9            1.4
## 10           1.5
## 11           1.5
## 12           1.6
## 13           1.4
## 14           1.1
## 15           1.2
## 16           1.5
## 17           1.3
## 18           1.4
## 19           1.7
## 20           1.5
## 21           1.7
## 22           1.5
## 23           1.0
## 24           1.7
```

```
## 25          1.9
## 26          1.6
## 27          1.6
## 28          1.5
## 29          1.4
## 30          1.6
## 31          1.6
## 32          1.5
## 33          1.5
## 34          1.4
## 35          1.5
## 36          1.2
## 37          1.3
## 38          1.4
## 39          1.3
## 40          1.5
## 41          1.3
## 42          1.3
## 43          1.3
## 44          1.6
## 45          1.9
## 46          1.4
## 47          1.6
## 48          1.4
## 49          1.5
## 50          1.4
## 51          4.7
## 52          4.5
## 53          4.9
## 54          4.0
## 55          4.6
## 56          4.5
## 57          4.7
## 58          3.3
## 59          4.6
## 60          3.9
## 61          3.5
## 62          4.2
## 63          4.0
## 64          4.7
## 65          3.6
## 66          4.4
## 67          4.5
## 68          4.1
## 69          4.5
## 70          3.9
## 71          4.8
## 72          4.0
## 73          4.9
## 74          4.7
## 75          4.3
## 76          4.4
## 77          4.8
## 78          5.0
```

```
## 79          4.5
## 80          3.5
## 81          3.8
## 82          3.7
## 83          3.9
## 84          5.1
## 85          4.5
## 86          4.5
## 87          4.7
## 88          4.4
## 89          4.1
## 90          4.0
## 91          4.4
## 92          4.6
## 93          4.0
## 94          3.3
## 95          4.2
## 96          4.2
## 97          4.2
## 98          4.3
## 99          3.0
## 100         4.1
## 101         6.0
## 102         5.1
## 103         5.9
## 104         5.6
## 105         5.8
## 106         6.6
## 107         4.5
## 108         6.3
## 109         5.8
## 110         6.1
## 111         5.1
## 112         5.3
## 113         5.5
## 114         5.0
## 115         5.1
## 116         5.3
## 117         5.5
## 118         6.7
## 119         6.9
## 120         5.0
## 121         5.7
## 122         4.9
## 123         6.7
## 124         4.9
## 125         5.7
## 126         6.0
## 127         4.8
## 128         4.9
## 129         5.6
## 130         5.8
## 131         6.1
## 132         6.4
```

```
## 133          5.6
## 134          5.1
## 135          5.6
## 136          6.1
## 137          5.6
## 138          5.5
## 139          4.8
## 140          5.4
## 141          5.6
## 142          5.1
## 143          5.1
## 144          5.9
## 145          5.7
## 146          5.2
## 147          5.0
## 148          5.2
## 149          5.4
## 150          5.1
```

```
iris.PLength <- select(iris, petal_length = Petal.Length) #saves "Petal.Length" as "petal_length"
#in new object "iris.PLength"
```

3d: rename()

rename() keeps all variables of a given dataset, bwhile renaming specified variables

```
rename(iris, petal_length = Petal.Length) #renames "Petal.Length" to "petal_length"
```

```
##     Sepal.Length Sepal.Width petal_length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 5            5.0         3.6          1.4         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 25           4.8         3.4          1.9         0.2     setosa
```

```
## 26          5.0         3.0          1.6         0.2     setosa
## 27          5.0         3.4          1.6         0.4     setosa
## 28          5.2         3.5          1.5         0.2     setosa
## 29          5.2         3.4          1.4         0.2     setosa
## 30          4.7         3.2          1.6         0.2     setosa
## 31          4.8         3.1          1.6         0.2     setosa
## 32          5.4         3.4          1.5         0.4     setosa
## 33          5.2         4.1          1.5         0.1     setosa
## 34          5.5         4.2          1.4         0.2     setosa
## 35          4.9         3.1          1.5         0.2     setosa
## 36          5.0         3.2          1.2         0.2     setosa
## 37          5.5         3.5          1.3         0.2     setosa
## 38          4.9         3.6          1.4         0.1     setosa
## 39          4.4         3.0          1.3         0.2     setosa
## 40          5.1         3.4          1.5         0.2     setosa
## 41          5.0         3.5          1.3         0.3     setosa
## 42          4.5         2.3          1.3         0.3     setosa
## 43          4.4         3.2          1.3         0.2     setosa
## 44          5.0         3.5          1.6         0.6     setosa
## 45          5.1         3.8          1.9         0.4     setosa
## 46          4.8         3.0          1.4         0.3     setosa
## 47          5.1         3.8          1.6         0.2     setosa
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
```

```
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85          5.4         3.0          4.5         1.5 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 90          5.5         2.5          4.0         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 95          5.6         2.7          4.2         1.3 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 100         5.7         2.8          4.1         1.3 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
## 102         5.8         2.7          5.1         1.9  virginica
## 103         7.1         3.0          5.9         2.1  virginica
## 104         6.3         2.9          5.6         1.8  virginica
## 105         6.5         3.0          5.8         2.2  virginica
## 106         7.6         3.0          6.6         2.1  virginica
## 107         4.9         2.5          4.5         1.7  virginica
## 108         7.3         2.9          6.3         1.8  virginica
## 109         6.7         2.5          5.8         1.8  virginica
## 110         7.2         3.6          6.1         2.5  virginica
## 111         6.5         3.2          5.1         2.0  virginica
## 112         6.4         2.7          5.3         1.9  virginica
## 113         6.8         3.0          5.5         2.1  virginica
## 114         5.7         2.5          5.0         2.0  virginica
## 115         5.8         2.8          5.1         2.4  virginica
## 116         6.4         3.2          5.3         2.3  virginica
## 117         6.5         3.0          5.5         1.8  virginica
## 118         7.7         3.8          6.7         2.2  virginica
## 119         7.7         2.6          6.9         2.3  virginica
## 120         6.0         2.2          5.0         1.5  virginica
## 121         6.9         3.2          5.7         2.3  virginica
## 122         5.6         2.8          4.9         2.0  virginica
## 123         7.7         2.8          6.7         2.0  virginica
## 124         6.3         2.7          4.9         1.8  virginica
## 125         6.7         3.3          5.7         2.1  virginica
## 126         7.2         3.2          6.0         1.8  virginica
## 127         6.2         2.8          4.8         1.8  virginica
## 128         6.1         3.0          4.9         1.8  virginica
## 129         6.4         2.8          5.6         2.1  virginica
## 130         7.2         3.0          5.8         1.6  virginica
## 131         7.4         2.8          6.1         1.9  virginica
## 132         7.9         3.8          6.4         2.0  virginica
## 133         6.4         2.8          5.6         2.2  virginica
```

```
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

```
#IN ORIGINAL OBJECT, while keeping remaining variables the same
iris.rename <- rename(iris, petal_length = Petal.Length) #creates new object "iris.rename" with
#"Petal.Length" renamed to "petal_length"
```

4 - standard analyses

4a: ANOVA

You can create ANOVA models using the function aov and running a summary on the the created model object.

```
mpgONcyl <- aov(mtcars$mpg ~ mtcars$cyl) #creates an object with results from the ANOVA,
#with cyl as the group variable and mpg as the dependent variable
summary(mpgONcyl) #prints the ANOVA table summary from the created object
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## mtcars$cyl   1  817.7   817.7   79.56 6.11e-10 ***
## Residuals   30  308.3    10.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Alternatively, you can create an OLS model and then use the anova() command to run an ANOVA on the created OLS model.

```
cyl.fit <- lm(mtcars$mpg ~ mtcars$cyl) #creates an OLS model with mpg as the DV and cyl as the IV
anova(cyl.fit) #runs an ANOVA on the OLS model created - notice that these results are
```

```
## Analysis of Variance Table
##
## Response: mtcars$mpg
##            Df Sum Sq Mean Sq F value    Pr(>F)
## mtcars$cyl  1 817.71  817.71  79.561 6.113e-10 ***
## Residuals  30 308.33   10.28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#the same as the results from summary(mpgONcyl)
```

4b: linear model (OLS)

The lm() function creates an OLS model, with your dependent variable on the left of the tilde, and your predictor variable or variables on the right of the tilde.

```
fit1 <- lm(mtcars$mpg ~ mtcars$wt) #runs a regression of mpg on wt
summary(fit1) #gives results of regression model
```

```
##
## Call:
## lm(formula = mtcars$mpg ~ mtcars$wt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## mtcars$wt    -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

```
fit1.b <- lm(mpg ~ wt , data = mtcars) #same as above, except with the "data = X" argument
#instead of "data$X"
```

You can also include interaction terms. Using var1*var2 includes both the interaction and the lower-order (main-effect) terms, while var1:var2 includes ***only*** the interaction term.

```
fit2 <- lm(mtcars$mpg ~ mtcars$wt*mtcars$cyl) #regression of mpg on interaction term between
#wt and cyl plus singular mpg and cyl terms.
summary(fit2)
```

```
##
## Call:
## lm(formula = mtcars$mpg ~ mtcars$wt * mtcars$cyl)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2288 -1.3495 -0.5042  1.4647  5.2344
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           54.3068     6.1275   8.863 1.29e-09 ***
## mtcars$wt             -8.6556     2.3201  -3.731 0.000861 ***
## mtcars$cyl            -3.8032     1.0050  -3.784 0.000747 ***
## mtcars$wt:mtcars$cyl   0.8084     0.3273   2.470 0.019882 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.368 on 28 degrees of freedom
## Multiple R-squared:  0.8606, Adjusted R-squared:  0.8457
## F-statistic: 57.62 on 3 and 28 DF,  p-value: 4.231e-12
```

```
fit2.b <- lm(mtcars$mpg ~ mtcars$wt:mtcars$cyl) #regression of mpg on JUST the interaction term
summary(fit2.b)
```

```
##
## Call:
## lm(formula = mtcars$mpg ~ mtcars$wt:mtcars$cyl)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0415 -2.2230 -0.7857  1.4079  7.1506
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           30.26785    1.10332   27.43  < 2e-16 ***
## mtcars$wt:mtcars$cyl -0.47935    0.04619  -10.38 1.92e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.86 on 30 degrees of freedom
## Multiple R-squared:  0.7821, Adjusted R-squared:  0.7749
## F-statistic: 107.7 on 1 and 30 DF,  p-value: 1.916e-11
```

```
fit2.c <- lm(mtcars$mpg ~ mtcars$wt + mtcars$cyl + mtcars$wt:mtcars$cyl) #equivalent to "fit2"
summary(fit2.c)
```

```
##
## Call:
## lm(formula = mtcars$mpg ~ mtcars$wt + mtcars$cyl + mtcars$wt:mtcars$cyl)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2288 -1.3495 -0.5042  1.4647  5.2344
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           54.3068     6.1275   8.863 1.29e-09 ***
## mtcars$wt             -8.6556     2.3201  -3.731 0.000861 ***
## mtcars$cyl            -3.8032     1.0050  -3.784 0.000747 ***
## mtcars$wt:mtcars$cyl   0.8084     0.3273   2.470 0.019882 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.368 on 28 degrees of freedom
## Multiple R-squared:  0.8606, Adjusted R-squared:  0.8457
## F-statistic: 57.62 on 3 and 28 DF,  p-value: 4.231e-12
```

4c: generalized linear models - RCT example taken from Dobson (1990), page 93

The glm(function) allows you to run more flexible generalized linear models, using more a variety of built-in and self-created distributional family forms. Here, we create an example of count data, and run a glm using a

Poisson family distribution appropriate for count data.

```
counts <- c(18,17,15,20,10,20,25,13,12) #creates a vector of counts of events
outcome <- gl(3,1,9) #generates a vector of factor levels, with 3 levels, 1 replication each,
#for a total of 9 observations (1-3, 3 times, i.e. 1,2,3,1,2,3,1,2,3)
treatment <- gl(3,3) #generates a vector of factor levels, with 3 levels, 3 replications each
#(i.e. 1,1,1,2,2,2,3,3,3)
treatment.b <- gl(3,3,9) #equivalent to "treatment"
d.AD <- data.frame(treatment, outcome, counts) #creates object "d.AD" combining three created vectors
glm.D93 <- glm(d.AD$counts ~ d.AD$outcome + d.AD$treatment, family = poisson()) #creates a GLM
#with created dataset, using a Poisson distribution.
```

```
anova(glm.D93) #ANOVA results of glm fit
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: d.AD$counts
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev
## NULL                               8    10.5814
## d.AD$outcome    2   5.4523         6     5.1291
## d.AD$treatment  2   0.0000         4     5.1291
```

```
summary(glm.D93) #full summary of GLM model
```

```
##
## Call:
## glm(formula = d.AD$counts ~ d.AD$outcome + d.AD$treatment, family = poisson())
##
## Deviance Residuals:
##        1         2         3         4         5         6         7
## -0.67125   0.96272  -0.16965  -0.21999  -0.95552   1.04939   0.84715
##        8         9
## -0.09167  -0.96656
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.045e+00  1.709e-01  17.815   <2e-16 ***
## d.AD$outcome2   -4.543e-01  2.022e-01  -2.247   0.0246 *
## d.AD$outcome3   -2.930e-01  1.927e-01  -1.520   0.1285
## d.AD$treatment2  1.338e-15  2.000e-01   0.000   1.0000
## d.AD$treatment3  1.421e-15  2.000e-01   0.000   1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 10.5814  on 8  degrees of freedom
## Residual deviance:  5.1291  on 4  degrees of freedom
## AIC: 56.761
##
```

```
## Number of Fisher Scoring iterations: 4
```

Extras

Extra 1 - $ for variable indication

You can use data$variable to select the *named* column "variable" in the object "data". You can also select rows (or columns) of a dataset meeting certain conditions; we will using row/column indices and brackets in a bit. The following commands are equivalent to the commands in **Section 3a**.

```
cyl6.gear4 <- mtcars[mtcars$cyl == 6 & mtcars$gear == 4 , ]
cyl6p.gearLT4 <- mtcars[mtcars$cyl >= 6 & mtcars$gear < 4 , ]
```

And here is a generic version that you can fill in with your own data, variables, and values. You can use "==" for equals, ">=" or "<=" for greater/less than or equal to, and ">" or "<" for greater than or less than.

```
dat2 <- dat1[dat1$var1 == X & dat1$var2 == Y , ]
```

Extra 2 - index/bracket notation

You can use brackets to indicate particular portions of an object.
If you are using a 1-dimensional object (a vector), only one number (or set of numbers) will be in the bracket. This tells you which observation (or set of observations) within the vector you are referring to.
If you are using a 2-dimensional object with both rows and columns (such as a dataset), there will be two numbers in the brackets separated by a comma. The first number refers to the row number or numbers; the second refers to the column number or numbers.

```
vec1[A] #prints Ath observation of object "vec1"
dat1[R , C] #prints observation in Rth row and Cth column of object "dat1"
```

You can use ":" to indicate a range selection.

```
vec1[A:D] #prints Ath-Dth observations of object "vec1"
dat1[R1:R4 , C] #Prints R1th through R4th rows of Cth column of "dat1"
dat1[R , C1:C4] #Prints observations in Rth row and C1th through C4th column of "dat1"
```

With 2-dimensional objects, you can also select all of the rows, or all of the columns, by leaving one or the other side of the comma blank.

```
dat1[ , C] #prints all rows of Cth column of "dat1"
dat1[R , ] #prints all columns of Rth row of "dat1"
```

You can use c() to indicate a non-continuous group of observations

```
vec1[c(A , B , E , Z)] #prints out observations A, B, E, and Z of "vec1"
```

You can use "-" to indicate all but an observation (or set of observations)

```
vec1[-A] #prints all but Ath observation of object "vec1"
vec1[-c(A , B , E , Z)] #prints all but observations A, B, E, and Z of "vec1"
```

The above functions only print out the result of you reference; you can also create a new object using this reference.

```
vec2 <- vec1[-A] #creates a new 1-dimensional object, "vec2", which contains all but
#the Ath observation of "vec1"
```

Extra 3: Practice codes for Index/Bracket Notation

What do each of these functions do?

```
vec1[c(I,J,L,X)]
```

```
vec3 <- vec1[c(I , J , L , X)]
```

```
dat1[R , A:D]
```

```
dat3 <- dat1[R , C1:C4]
```

```
dat4 <- dat1[c(R1 , R2 , R5) , ]
```

```
dat5 <- dat1[-R1 , c(C5 , C7 , C9)]
```