Homework 1

Kitu Komya

October 7, 2016

1. Perform Basic Calculations in R

1.1 Use `R` as a calculator and show how to get answers for:

1.

```
pt_1 <- 3 * (-2)^2 + 5 * (-2) - 2
pt_1
```

```
## [1] 0
```

The answer to this calculation is 0.

2.

```
pt_2 <- sqrt(12) * (1 - (1/(3*3)) + (1/(5*(3^2))) - (1/(7*(3^3))))
pt_2
```

```
## [1] 3.137853
```

The answer to this calculation is 3.1378529.

1.2 Vectorized operations

```
vec <- c((-50:50)^2)
vec
```

```
##   [1] 2500 2401 2304 2209 2116 2025 1936 1849 1764 1681 1600 1521 1444 1369
##  [15] 1296 1225 1156 1089 1024  961  900  841  784  729  676  625  576  529
##  [29]  484  441  400  361  324  289  256  225  196  169  144  121  100   81
##  [43]   64   49   36   25   16    9    4    1    0    1    4    9   16   25
##  [57]   36   49   64   81  100  121  144  169  196  225  256  289  324  361
##  [71]  400  441  484  529  576  625  676  729  784  841  900  961 1024 1089
##  [85] 1156 1225 1296 1369 1444 1521 1600 1681 1764 1849 1936 2025 2116 2209
##  [99] 2304 2401 2500
```

I have outputted the squared values of the vectors ranging from -50 to 50 above.

1.3 Sample vs. population standard deviation

1.

```
values <- c(27, 36, 50, -24, 9, -38) # I've created and stored a vector into values
```

2.

```r
sd(values) # This function will compute and output the standard deviation of my vector, values.
```

```
## [1] 34.71599
```

The standard deviation of the vector, values, is 34.7159906.

3.

```r
population_sd <- sqrt( (1/length(values)) * sum((values - mean(values))^2)) # This is the formula for t
population_sd # This will output the calculation.
```

```
## [1] 31.69122
```

The population standard deviation is 31.6912186.

```r
sample_sd <- sqrt( (1/(length(values)-1)) * sum((values - mean(values))^2)) # This is the formula for t
sample_sd # This will output the calculation.
```

```
## [1] 34.71599
```

The sample standard deviation is 34.7159906.

4.
The function sd() calculates the sample standard deviaton since they both match. Let's confirm this.

```r
sd(values) == sample_sd # This will compare the numerical value of the two outputs to see if they match
```

```
## [1] TRUE
```

Since the output is `TRUE`, our first intuition (or rather, eyesight) was correct! Instead of carrying out these tedious calculations, I could have simply Googled this question and used the answers from stackoverflow.com. :)

1.4 Vector Classes

1.

```r
numbers <- c(-4, 5, 35, 12) # I've created a vector of the class integers/numbers.
strings <- c("James", "is", "awesome!") # I've created a vector of the class strings.
booleans <- c(F, T, F, T) # I've created a vector of the class booleans.
```

2.

```r
nr <- typeof(c(numbers, strings)) # This vector stores the class of the newly created vector.
nr # This will output the class.
```

```
## [1] "character"
```

```r
nb <- typeof(c(numbers, booleans)) # This vector stores the class of the newly created vector.
nb # This will output the class.
```

```
## [1] "double"
```

```r
sb <- typeof(c(strings, booleans)) # This vector stores the class of the newly created vector.
sb # This will output the class.
```

```
## [1] "character"
```

```r
nsb <- typeof(c(numbers, strings, booleans)) # This vector stores the class of the newly created vector
nsb # This will output the class.
```

```
## [1] "character"
```

1. The class of a vector of numbers and strings is character.
2. The class of a vector of numbers and booleans is double.
3. The class of a vector of strings and booleans is character.
4. The class of a vector of numbers, strings, and booleans is character.

3.
R probably does this so that all of the objects within a vector are of the same class so that they can all be manipulated based on the class' functionality. It makes sense for vectors that contain strings and other classes to become characters to conserve its data, while making numbers and booleans (booleans are 0 and 1) doubles, indicative of a numerical vector.

2. Reading in different data file types

2.1 Childhood Respiratory Disease

1.

```
crd <- read.table(file = "http://www.statsci.org/data/general/fev.txt", header = TRUE) # The file has b
```

2.

```
names(crd) # This outputs the variables in the data.
```

```
## [1] "ID"     "Age"    "FEV"    "Height" "Sex"    "Smoker"
```

The variables in the data are ID, Age, FEV, Height, Sex, Smoker.
3.

```
names(crd) <- c("id", "age", "lung_cap", "height", "sex", "smoker") # I have renamed the variables.
```

4.

```
names(crd) # I'm re-printing the variables with the new names.
```

```
## [1] "id"       "age"      "lung_cap" "height"   "sex"      "smoker"
```

The re-named variables in the data are id, age, lung_cap, height, sex, smoker.
5.

```
table(crd$sex, crd$smoker) # This table outputs the number of each sex who smokes and who don't smoke.
```

```
##
##          Current Non
##   Female      39 279
##   Male        26 310
```

In order to answer the question, we will do some math.

```
f <- 39/(39+279) # I divided those females who smoke over the total female sample population to find th
f # It will output the proportion.
```

```
## [1] 0.1226415
```

The proportion of females who smoke is 0.1226415.

```
m <- 26/(26+310) # I divided those males who smoke over the total male sample population to find the pr
m # It will output the proportion.
```

```
## [1] 0.07738095
```

The proportion of males who smoke is 0.077381. It seems that the proportion of female smokers is higher than male smokers. Let's confirm:

```
f > m # This boolean will validate our claim that females' proportion is greater than males'
```

```
## [1] TRUE
```

Since the boolean reads `TRUE`, we were correct in assessing that there is a higher proportion of female smokers than male smokers in our sample population.

2.2 Reading 'Stata', 'SAS', and 'SPSS' files

1.

```
library(haven) # I am loading the package haven into R.
```

2.

```
spss <- read_sav(file = "http://www.sjsu.edu/people/carlos.e.garcia/courses/soci104/Course-Assignments/

stata <- read_dta(file = "http://qcpages.qc.cuny.edu/~rvesselinov/statadata/WAGEPAN.DTA") # I have load

sas <- read_sas("http://biostat3.net/download/sas/colon.sas7bdat") # I have loaded the .sas7bdat data s
```

3.

1.

```
names(spss) # This will print out the variables of the spss data set.
```

```
##  [1] "id"       "wrkstat"  "marital"  "age"      "educ"     "sex"
##  [7] "race"     "partyid"  "polviews" "hlth4"    "hlth5"    "relig"
## [13] "owngun"   "cappun"   "gunlaw"   "grass"    "empathy1" "empathy2"
## [19] "empathy3" "empathy4" "empathy5" "empathy6" "empathy7"
```

As you can see, there are 23 variables in the spss data set!

2.

```
observations <- ncol(stata)*nrow(stata) # I am multiplying the number of rows by the number of columns
observations # This will print out the number of observations.
```

```
## [1] 44000
```

```
names(stata) # This will print out the variables of the stata data set.
```

```
##  [1] "nr"       "year"     "agric"    "black"    "bus"      "construc"
##  [7] "ent"      "exper"    "fin"      "hisp"     "poorhlth" "hours"
## [13] "manuf"    "married"  "min"      "nrthcen"  "nrtheast" "occ1"
## [19] "occ2"     "occ3"     "occ4"     "occ5"     "occ6"     "occ7"
## [25] "occ8"     "occ9"     "per"      "pro"      "pub"      "rur"
## [31] "south"    "educ"     "tra"      "trad"     "union"    "lwage"
## [37] "d81"      "d82"      "d83"      "d84"      "d85"      "d86"
## [43] "d87"      "expersq"
```

As you can see, there are 44 variables in the stata data set, and 44000 observations!

4.

```
library(knitr) # I am loading the package knitr into R.
```

```
## Warning: package 'knitr' was built under R version 3.4.2
```

```
kable(head(sas, nrows = 6)) # I am creating a table of the first six rows of the sas data set.
```

sex	age	stage	mmdx	yydx	surv_mm	surv_yy	status	subsite	year8594	agegrp	dx	exit
2	77	3	9	1977	16.5	1.5	1	2	0	3	1977-09-07	1979-01-22
2	78	1	10	1978	82.5	6.5	2	1	0	3	1978-10-07	1985-08-22
1	78	3	12	1978	1.5	0.5	1	3	0	3	1978-12-07	1979-01-22
1	76	3	10	1976	1.5	0.5	1	3	0	3	1976-10-07	1976-11-22
1	80	1	4	1980	8.5	0.5	1	3	0	3	1980-04-07	1980-12-22
2	75	1	11	1975	23.5	1.5	1	1	0	3	1975-11-07	1977-10-22

Wow, the kable function sure made our table look very pretty!

3. Create and interpret plots

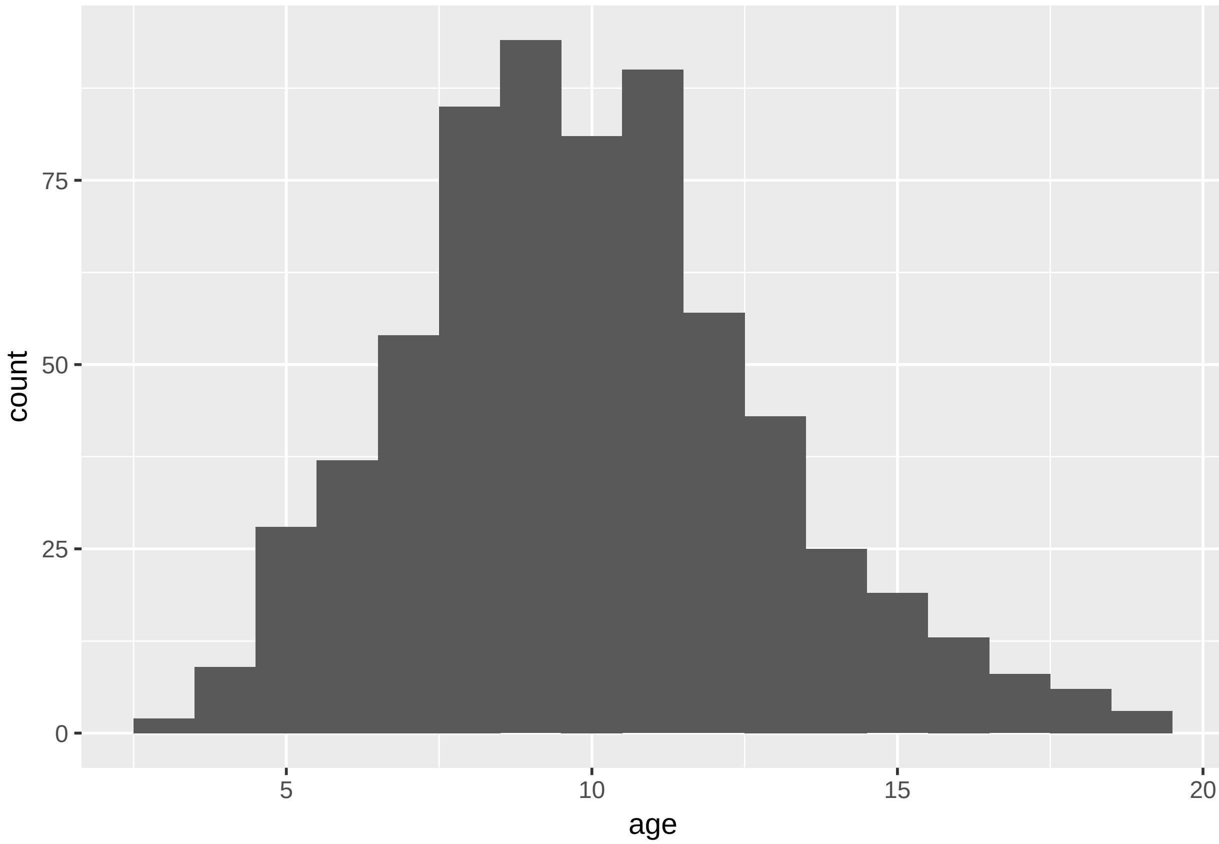
1.

```
library(ggplot2) # I am loading the package ggplot2 into R.
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

2.

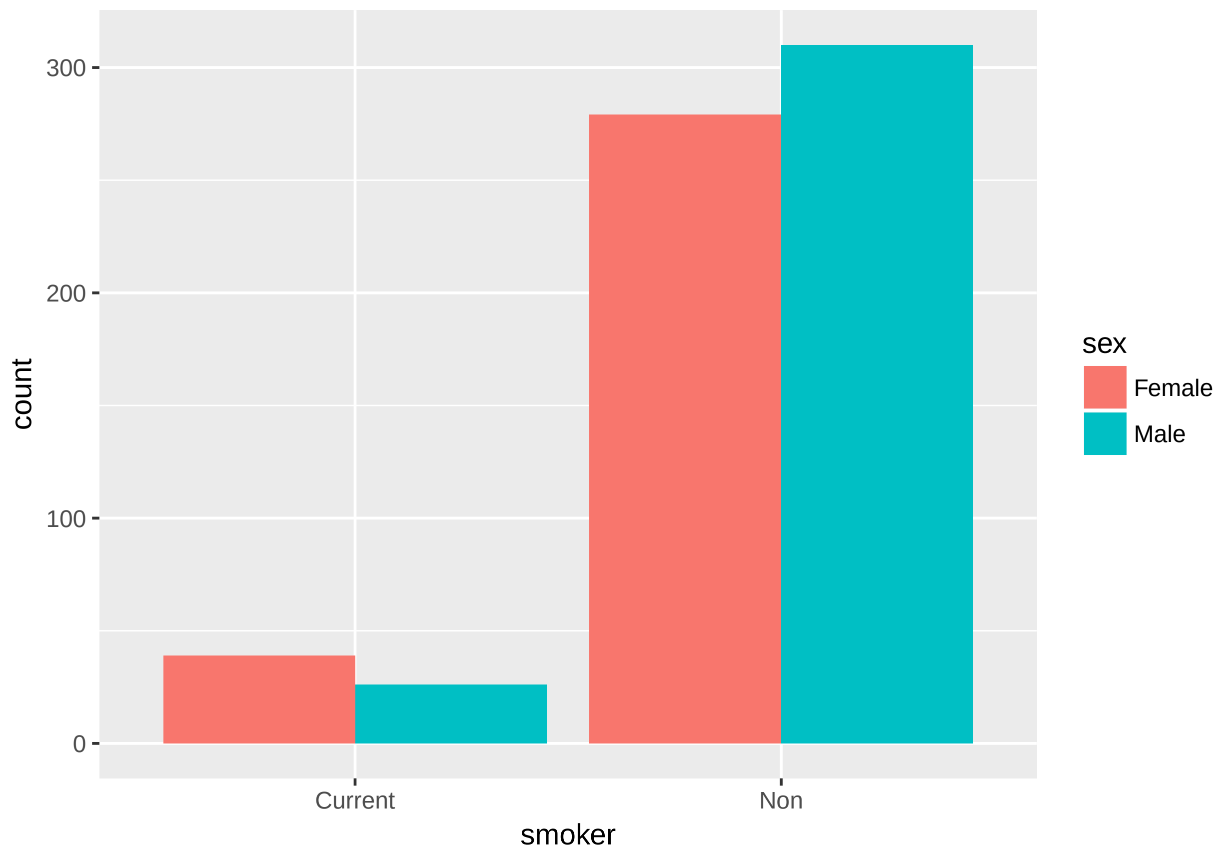
```
p <- ggplot(data = crd) # This is the first layer of ggplot2, where we read in the data
p + geom_histogram(aes(x = age), binwidth = 1) # The additional layer creates the histogram.
```



We can see that the histogram of the ages of the people in the CRD data is almost normally distributed with a mean centered around 10 years old.

3.

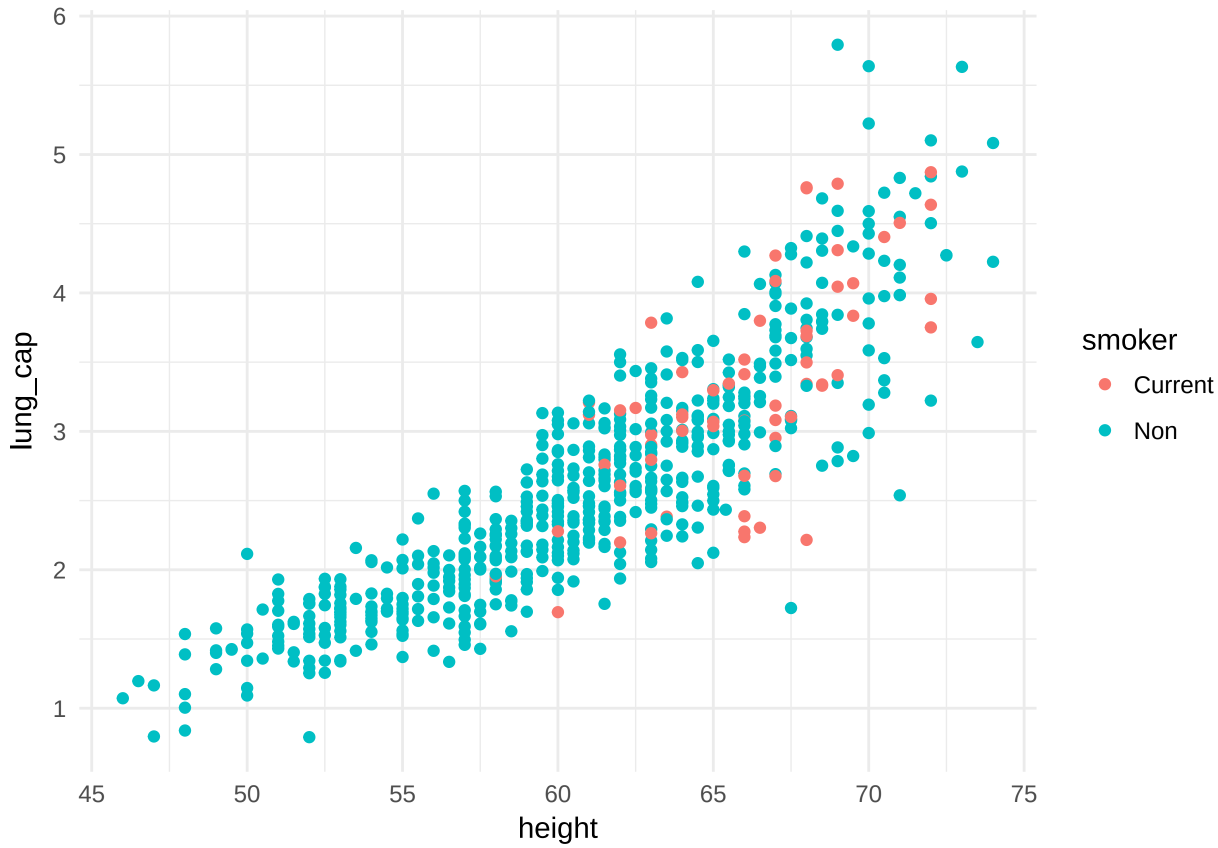
```
p + geom_bar(aes(x = smoker, fill = sex), position = "dodge") # I have created a bar graph to compare s
```



It is interesting to note that from those who smoke, there is a higher count of women, and from those who do not smoke, there is a lower count of women, leading us to believe that women smoke more often than men, as extrapolated from this sample.

4.

```
q <- ggplot(data = crd, aes(x = height, y = lung_cap)) # This is the first layer of ggplot2, where we r
q + geom_point(aes(color = smoker)) + theme_minimal() # The additional layer colors the data points bas
```



This is a very interesting graph! It shows a positive, linear, moderately strong association between height and lung capacity, with the variable of smoking not affecting lung capacity to the extent predicted. What a cool visualization!