Live Session Unit 05

Carson Drake

10/1/2018

Data Munging (30 points)

Utilize yob2016.txt for this question. This file is a series of ***popular children's names*** born in the year 2016 in the United States. It consists of three columns with a first name, a gender, and the amount of children given that name. However, the data is raw and will need cleaning to make it tidy and usable.

a. First, import the .txt file into R so you can process it. Keep in mind this is not a CSV file. You might have to open the file to see what you're dealing with. Assign the resulting data frame to an object, df, that consists of three columns with human- readable column names for each.

```
## Load in yob2016 data
file_y2016 <- file("../data/yob2016.txt");
df <- read.csv2(file=file_y2016, header = F, col.names = c("name", "gender", "amount of children"));

## Reformat name to characters
df$name <- as.character(df$name);
```

b. Display the summary and structure of df

```
## Summary of 2016 children names.
knitr::kable(summary(df), caption = "Summary of 2016 Names");
```

Table 1: Summary of 2016 Names

name	gender	amount.of.children
Length:32869	F:18758	Min. : 5.0
Class :character	M:14111	1st Qu.: 7.0
Mode :character	NA	Median : 12.0
NA	NA	Mean : 110.7
NA	NA	3rd Qu.: 30.0
NA	NA	Max. :19414.0

```
## Structure of Data
print("Structure of df");
```

```
## [1] "Structure of df"
```

```
str(df)
```

```
## 'data.frame':    32869 obs. of  3 variables:
##  $ name              : chr  "Emma" "Olivia" "Ava" "Sophia" ...
##  $ gender            : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
##  $ amount.of.children: int  19414 19246 16237 16070 14722 14366 13030 11699 10926 10733 ...
```

c. Your client tells you that there is a problem with the raw file. One name was entered twice and misspelled. The client cannot remember which name it is; there are thousands he saw! But he did mention he accidentally put three y's at the end of the name. Write an R command to figure out which name it is and display it.

```
## Find match for names endding in "yyy" using a regex.
mispelled <- df[str_detect(df$name,"\\w*yyy\\b"),];
mispelled;
```

	name	gender	amount.of.children
212	Fionayyy	F	1547

d. Upon finding the misspelled name, please remove this particular observation, as the client says it's redundant. Save the remaining dataset as an object: y2016

```
## Filter out the mispelled name.
y2016 <- df %>%
  filter(name != mispelled$name);

## show last five entries
head(y2016,5);
```

name	gender	amount.of.children
Emma	F	19414
Olivia	F	19246
Ava	F	16237
Sophia	F	16070
Isabella	F	14722

Data Merging (30 points)

Utilize yob2015.txt for this question. This file is similar to yob2016, but contains names, gender, and total children given that name for the year 2015.

a. Like 1a, please import the .txt file into R. Look at the file before you do. You might have to change some options to import it properly. Again, please give the dataframe human-readable column names. Assign the dataframe to y2015.

```
## Load in the yob2015 data.
y2015 <- read.csv(file("../data/yob2015.txt"),
                  header = F,
                  col.names = c("name", "gender", "amount of children"));

## Reformat name to character
y2015$name <- as.character(y2015$name);
```

b. Display the last ten rows in the dataframe. Describe something you find interesting about these 10 rows.

```
## Tail the last 10 rows in the set.
tail(y2015,10);
```

	name	gender	amount.of.children
33054	Ziyu	M	5
33055	Zoel	M	5
33056	Zohar	M	5
33057	Zolton	M	5

	name	gender	amount.of.children
33058	Zyah	M	5
33059	Zykell	M	5
33060	Zyking	M	5
33061	Zykir	M	5
33062	Zyrus	M	5
33063	Zyus	M	5

An interesting observation is that all of the names are male, and that they are all 5 for their usage count.

c. Merge y2016 and y2015 by your Name column; assign it to final. The client only cares about names that have data for both 2016 and 2015; there should be no NA values in either of your amount of children rows after merging.

```
#Inner join the two data sets so that only the rows in both remain
final <- inner_join(y2015,y2016, by=c("name", "gender")) %>%
  filter(!is.na(amount.of.children.x)|!is.na(amount.of.children.y)) %>%
  rename(count2015 = amount.of.children.x, count2016 = amount.of.children.y)
```

Data Summary (30 points)

a. Create a new column called "Total" in final that adds the amount of children in 2015 and 2016 together. In those two years combined, how many people were given popular names?

```
## Calcualte the total population between the two years.
final$Total <- final$count2015 + final$count2016;
```

b. Sort the data by Total. What are the top 10 most popular names?

```
## Reorder the names by the most popular.
final <- final %>%
  arrange(desc(Total));

## Print out the 10 most popular names.
head(final,10);
```

name	gender	count2015	count2016	Total
Emma	F	20415	19414	39829
Olivia	F	19638	19246	38884
Noah	M	19594	19015	38609
Liam	M	18330	18138	36468
Sophia	F	17381	16070	33451
Ava	F	16340	16237	32577
Mason	M	16591	15192	31783
William	M	15863	15668	31531
Jacob	M	15914	14416	30330
Isabella	F	15574	14722	30296

c. The client is expecting a girl! Omit boys and give the top 10 most popular girl's names.

```
## Filter out the male names and then list the 10 most popular names.
top10.f <- final %>%
```

```
  filter(gender == "F") %>%
  head(n=10);
top10.f
```

name	gender	count2015	count2016	Total
Emma	F	20415	19414	39829
Olivia	F	19638	19246	38884
Sophia	F	17381	16070	33451
Ava	F	16340	16237	32577
Isabella	F	15574	14722	30296
Mia	F	14871	14366	29237
Charlotte	F	11381	13030	24411
Abigail	F	12371	11699	24070
Emily	F	11766	10926	22692
Harper	F	10283	10733	21016

d. Write these top 10 girl names and their Totals to a CSV file. Leave out the other columns entirely.

```
## Take the top 10 list from above step,
## strip out unecesary columns, and save to
## data directory in CSV format.
top10.f %>%
  select(name,Total) %>%
  write_csv(path = "../data/top10_female_names.csv");
```