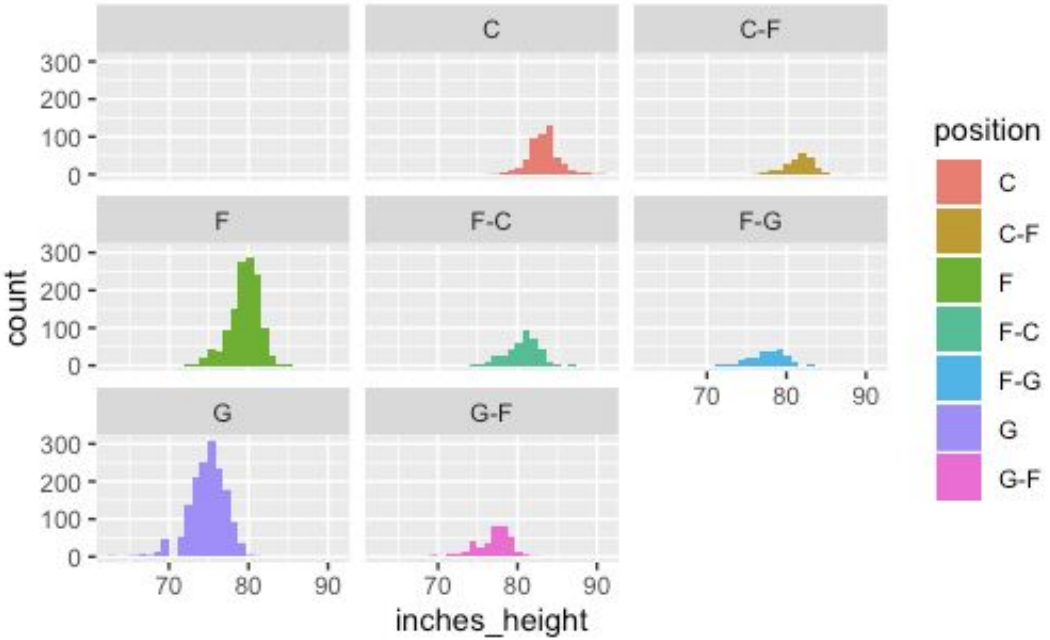
Unit5: For Live Session

DS6306
Garrity

PART 1 - BBALL Heights

Convert to inches

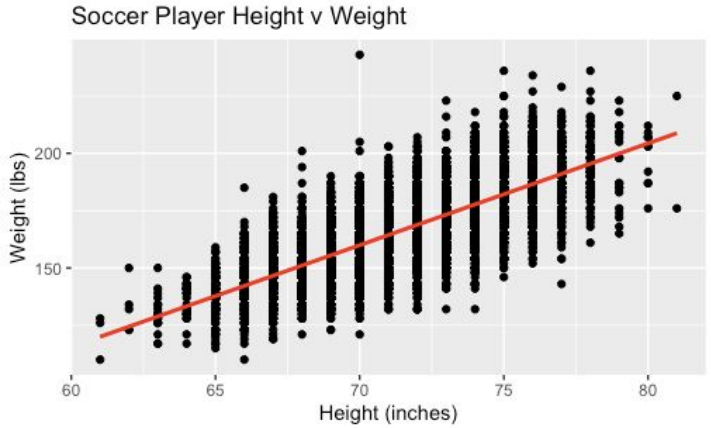
```
temp_var <-
str_split_fixed(playerbb$height, n = 2,
pattern = "-")

playerbb$inches_height =
(as.numeric(temp_var[,1])*12)+as.numeric(te
mp_var[,2])

ggplot(playerbb,aes(x=inches_height, fill =
position))+
    geom_histogram() +
  facet_wrap(~position)
```



PART 2A - FIFA Height v Weight

Linear Regression

```
fifa_sh <- fifa %>% separate(Height,into = c("Feet",
"Inches"), sep = "'")
fifa$InchesHeight =
(as.numeric(fifa_sh$Feet)*12+as.numeric(fifa_sh$Inches))

temp_weight <- str_split(fifa$Weight, "lbs")

weightHolder = c()

for(i in 1 : length(temp_weight)) #for each important word
in the headline
{
  weightHolder[i] <- temp_weight[[i]][1]
}

fifa$LbsWeight <- as.numeric(weightHolder)

fifa %>% ggplot(aes(x=InchesHeight, y=LbsWeight)) +
  geom_point() +
  labs(title="Soccer Player Height v Weight", x = "Height
(inches)", y = "Weight (lbs)")

fit_HvW <- lm(fifa$LbsWeight ~ fifa$InchesHeight)
summary(fit_HvW)
```

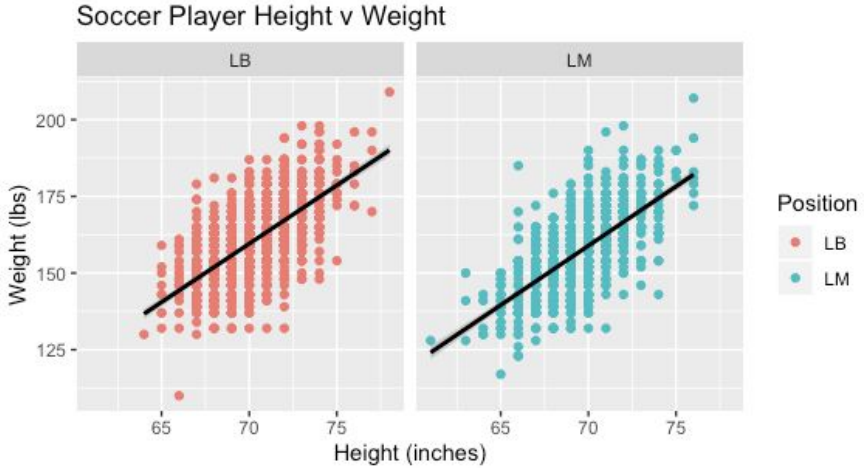



```
lm(formula = fifa$LbsWeight ~ fifa$InchesHeight)

Residuals:
    Min      1Q  Median      3Q     Max
-48.023  -5.933  -0.609   7.184  83.067

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -150.95778    2.04623  -73.77   <2e-16 ***
fifa$InchesHeight    4.44130    0.02865  155.00   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.23 on 18157 degrees of freedom
  (48 observations deleted due to missingness)
Multiple R-squared:  0.5695,  Adjusted R-squared:  0.5695
F-statistic: 2.402e+04 on 1 and 18157 DF,  p-value: < 2.2e-16
```

PART 2B - FIFA Height v Weight for LB and LM Positions

Linear Regression

```
fifa %>% filter(Position == "LB" | Position == "LM") %>%
  ggplot(aes(x=InchesHeight, y=LbsWeight, col = Position)) +
  geom_point() +
  facet_wrap(~Position) +
  geom_smooth(method = "lm", col = "black") +
  labs(title="Soccer Player Height v Weight", x = "Height
(inches)", y = "Weight (lbs)")
```




```
############################################################
> fit_HvW_LB <- lm(fifa$LbsWeight[fifa$Position == "LB"] ~
fifa$InchesHeight[fifa$Position == "LB"])
> summary(fit_HvW_LB)

Residuals:
    Min      1Q  Median      3Q     Max
-35.176  -5.986  -0.367   6.633  30.871

Coefficients:
                                          Estimate Std. Error
t value Pr(>|t|)
(Intercept)                               -107.100      9.146
-11.71   <2e-16 ***
fifa$InchesHeight[fifa$Position == "LB"]     3.809      0.130
29.30   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’
1

Residual standard error: 9.72 on 1320 degrees of freedom
Multiple R-squared:  0.3941,  Adjusted R-squared:  0.3937
F-statistic: 858.7 on 1 and 1320 DF,  p-value: < 2.2e-16
```

```
> fit_HvW_LM <- lm(fifa$LbsWeight[fifa$Position == "LM"] ~
fifa$InchesHeight[fifa$Position == "LM"])
> summary(fit_HvW_LM)

Residuals:
    Min      1Q  Median      3Q     Max
-31.370  -5.633  -0.423   5.972  41.577

Coefficients:
                                          Estimate Std. Error
t value Pr(>|t|)
(Intercept)                               -111.8895     9.0557
-12.36   <2e-16 ***
fifa$InchesHeight[fifa$Position == "LM"]     3.8684     0.1302
29.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.381 on 1093 degrees of freedom
Multiple R-squared:  0.4469,  Adjusted R-squared:  0.4464
F-statistic: 883.2 on 1 and 1093 DF,  p-value: < 2.2e-16
```

Baby Names - Question 1

Data Munging

```
baby = read.csv2("/Users/stevengarrity/SMU_MSDS/DS6306_DoingDataScience/DDS_Git/Unit 5/yob2016.txt", header=FALSE) # read in
name data

df <- data.frame(Name = baby[,1], Sex = baby[,2], Count = baby[,3])

> head(df)
      Name Sex Count
1     Emma   F 19414
2   Olivia   F 19246
3      Ava   F 16237
4   Sophia   F 16070
5 Isabella   F 14722
6      Mia   F 14366

> writeLines(df$Name[str_detect(df$Name,"yyy")])
Fionayyy


> y2016 <- df[-c(212),]
> dim(df)
[1] 32869     3
> dim(y2016)
[1] 32868     3
```

Baby Names - Question 2

Data Munging

```
> tail(y2015,10)
        Name Sex Count
33054   Ziyu   M     5
33055   Zoel   M     5
33056  Zohar   M     5
33057 Zolton   M     5
33058   Zyah   M     5
33059 Zykell   M     5
33060 Zyking   M     5
33061  Zykir   M     5
33062  Zyrus   M     5
33063   Zyus   M     5
```

These are some rather unique names. Odd that the count is "5" for all. Hmm….

```
> final <- merge(y2016, y2015, by = c("Name" = "Name", "Sex" = "Sex"))
> head(final)

       Name Sex Count.x Count.y
1     Aaban   M       9      15
2     Aabha   F       7       7
3 Aabriella   F      11       5
4     Aadam   M      18      22
5   Aadarsh   M      11      15
6     Aaden   M     194     297
```

Baby Names - Question 3

Data Munging

```
final <- final %>% mutate(total = Count.x + Count.y)
head(final, 2)
   Name Sex Count.x Count.y total
1 Aaban   M       9      15    24
2 Aabha   F       7       7    14
```

Top 10 most popular baby names:

```
> final %>% arrange(desc(total))
           Name Sex Count.x Count.y total
1          Emma   F   19414   20415 39829
2        Olivia   F   19246   19638 38884
3          Noah   M   19015   19594 38609
4          Liam   M   18138   18330 36468
5        Sophia   F   16070   17381 33451
6           Ava   F   16237   16340 32577
7         Mason   M   15192   16591 31783
8       William   M   15668   15863 31531
9         Jacob   M   14416   15914 30330
10     Isabella   F   14722   15574 30296
```

Top 10 most popular female names:

```
> final %>% filter(Sex == "F") %>% arrange(desc(total))
         Name Sex Count.x Count.y total
1        Emma   F   19414   20415 39829
2      Olivia   F   19246   19638 38884
3      Sophia   F   16070   17381 33451
4         Ava   F   16237   16340 32577
5    Isabella   F   14722   15574 30296
6         Mia   F   14366   14871 29237
7   Charlotte   F   13030   11381 24411
8     Abigail   F   11699   12371 24070
9       Emily   F   10926   11766 22692
10     Harper   F   10733   10283 21016
```

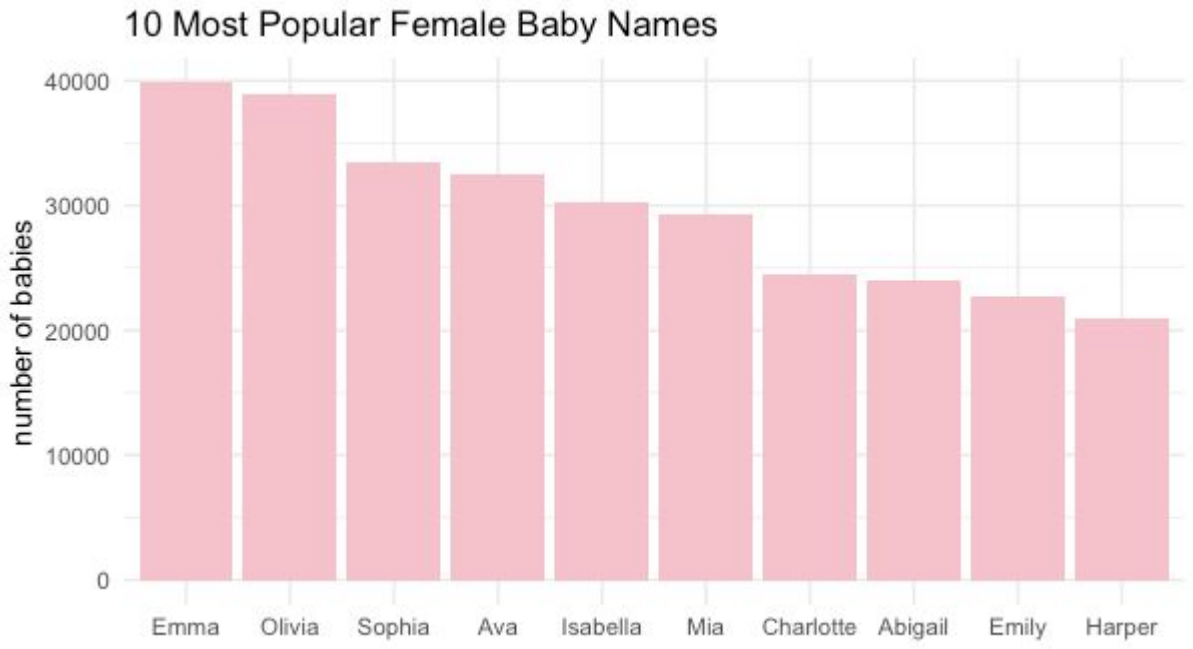
Write data to file:

```
temp <- final %>% filter(Sex == "F") %>% arrange(desc(total))
PopularFemaleNames <- data.frame(Name = temp$Name, Count = temp$total)
write.csv(PopularFemaleNames, file = "PopularFemaleNames.csv")
```

Baby Names - Question 4

Data Viz

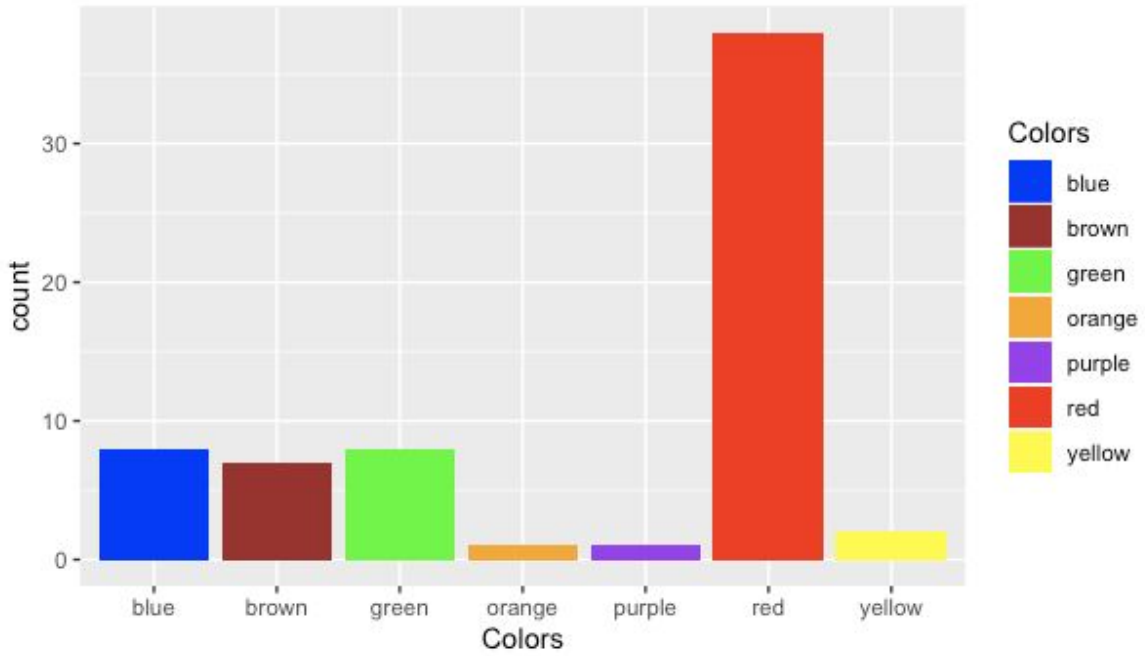
```
final_viz <- final %>% filter(Sex == "F") %>% arrange(desc(total))
final_viz = final_viz[1:10,]

final_viz %>% ggplot(aes(x = reorder(Name, -total), y = total)) +
  geom_bar(stat="identity", fill = "pink") +
  labs(title="10 Most Popular Female Baby Names", x="", y="number of babies") +
  theme_minimal()
```



Assignment from Async Videos

Run the code below. 61 sentences are selected in the has_color object. Inspect sentence 41 and 43, and explain what the problem is and why it is occurring. For live session, fix the problem, and reproduce the plot.

```
colors =
c("orange","blue","yellow","green","purple",
"brown","red")
color_expression = str_c(colors, collapse =
"|")
color_expression
has_color =
str_subset(sentences,color_expression) #
filtering based on the variable
color_expression
has_color
has_color <- has_color[-c(41,43)]
matches =
str_extract(has_color,color_expression)
matches
matches_all =
str_extract_all(has_color,color_expression,
simplify = TRUE)
matches_all
class(matches_all)
matches_all =
unlist(str_extract_all(has_color,color_expre
ssion))
matches_all
matchDF = data.frame(Colors = matches_all)
matchDF %>% ggplot(aes(x = Colors, fill =
Colors)) + geom_bar()+
scale_fill_manual(values=colors[order(colors
)])
```



Takeaways & Questions

Maybe I missed it in the async material, but I could not figure out how to get the index using "`str_detect`". I had to hunt through the entire logical TRUE/FALSE matrix and manually identify the index (row) and then use "`variable_name[-c(index),]`" to drop the row. Not very efficient! I could use "`str_replace`" to fill with NA or "`[!str_detect]`", but it would still leave the data in the other columns. I was hoping to eliminate the entire row containing the duplicate name just to keep things as tidy as possible. I'm sure there is a better way....

Droplevels is kicking my ass. I tried the following for the FIFA dataset (similar situation with the BBall dataset), but it made a mess of the data frame. The upshot is that my graphs continue to have empty levels plotted. What am I doing wrong?

```
levels(fifa$Height) = droplevels(fifa$Height,"")
```

The homework was relatively straightforward and the async material was clear but my grasp of regular expressions is pretty shaky. Going to need some more practice!