# Xinyi & Daiyan 632 Project

```
library(tidyverse)
library(stringr)
library(randomForest)
library(vip)
```

## Data Cleaning

Load the data and inspect:

```
df_raw <- read.csv("data.csv")
head(df_raw)
```

```
##            Id              Channel       Subscribers
## 1 FozCkl1xj-w            JRE Clips 6.28M subscribers
## 2 RN8yoi-e2yc     Mythical Kitchen  1.9M subscribers
## 3 IugcIAAZJ2M             Munchies 4.59M subscribers
## 4 JiEO6F8i0eU Parks and Recreation  282K subscribers
## 5 1T4XMNN4bNM               Vsauce 17.4M subscribers
## 6 0ZWGeidvrJw           Doctor Who 1.59M subscribers
##                                                                        Title CC
## 1                       Former CIA Agent Breaks Down Jeffrey Epstein Case  0
## 2 $420 Pizza Hut Stuffed Crust Pizza | Fancy Fast Food | Mythical Kitchen  1
## 3                    The Iconic $1 Pizza Slice of NYC | Street Food Icons  0
## 4                  Ron Swanson: The Papa of Pawnee | Parks and Recreation  0
## 5                              What's The Most Dangerous Place on Earth?  1
## 6     The Doctor Defeats the Abzorbaloff | Love and Monsters | Doctor Who  1
##                                          URL    Released      Views
## 1 https://www.youtube.com/watch?v=FozCkl1xj-w 2 years ago 7.9M views
## 2 https://www.youtube.com/watch?v=RN8yoi-e2yc             2.7M views
## 3 https://www.youtube.com/watch?v=IugcIAAZJ2M 2 years ago  11M views
## 4 https://www.youtube.com/watch?v=JiEO6F8i0eU 3 years ago 2.3M views
## 5 https://www.youtube.com/watch?v=1T4XMNN4bNM 9 years ago  21M views
## 6 https://www.youtube.com/watch?v=0ZWGeidvrJw 7 years ago 8.5M views
##               Category
## 1                 Blog
## 2                 Food
## 3                 Food
## 4 Entertainment,Comedy
## 5              Science
## 6        Entertainment
##
## 1
## 2 - Oh, that's dirty.\n- Wow! - Whoa.\n- You're a dirty girl. (upbeat music) - Hey man. - What'd you
## 3
```

```
## 4
## 5
## 6
##   Length
## 1  13:32
## 2  24:26
## 3   7:51
## 4  10:06
## 5   9:29
## 6   4:20
```

We notice several problems, like:

1. Views: The Views variable is in string format and the units are different, like "10K views", "10M views". We prefer it to be in number and in the same unit in order to conduct statistical analysis.
2. Subscribers: The same problems as Views. The Subscribers variable is like "10K subscribers", "10M subscribers".
3. Length: The video length is in string format, like "12:00", "1:12:00". We need it to be in number and in the same unit.
4. Released: The Released variable is in string format, like "2 years age", "10 month ago". We need it to be in number and in the same unit.

Therefore, we need to do data cleaning first.

```
# Unify units and convert string to number, like: 10K views -> 10, 10M views -> 10000
cleanViews <- function(str) {
  str <- str_remove(str, " views")
  last <- str_sub(str, -1)
  views <- str %>% str_remove(last) %>% as.numeric()
  if (last == "M") return(1000*views)
  else return(views)
}

# Unify units and convert string to number, like: 10K subscribers -> 10, 10M subscribers -> 10000
cleanSubscribers <- function(str) {
  str <- str_remove(str, " subscribers")
  last <- str_sub(str, -1)
  views <- str %>% str_remove(last) %>% as.numeric()
  if (last == "M") return(1000*views)
  else return(views)
}

# Convert time in string format to number of minutes, like: 12:00 -> 12, 1:12:00 -> 72
cleanLength <- function(str) {
  list <- str_split(str, ":")
  len <- length(list[[1]])
  if (len == 3) {
     h <- as.numeric(list[[1]][1])
     m <- as.numeric(list[[1]][2])
     return((m + 1) + 60*m)
  } else {
```

```
    m <- as.numeric(list[[1]][1])
    return(m+1)
  }
}

# Convert time to number of months ago, like: 1 years ago -> 12, 10 months ago to 10
cleanReleased <- function(str) {
  str <- str_remove(str, "Streamed ")
  list <- str_split(str, " ")
  if (list[[1]][2] == "years") return(as.numeric(list[[1]][1])*12)
  else return(as.numeric(list[[1]][1]))
}

# Remove NAs
df <- df_raw %>%
  filter(
    !is.na(Released) & Released != ""
  )

# Clean the data
df <- df %>% mutate(
  Views = map_dbl(Views, cleanViews),
  Subscribers = map_dbl(Subscribers, cleanSubscribers),
  Length = map_dbl(Length, cleanLength),
  Released = map_dbl(Released, cleanReleased)
)

head(df)

# Save for future use
write_csv(df, "cleaned_data.csv")
```

After cleaning, Views, Subscribers, Released and Length are numbers, while Views, Subscribers are in K and Released and Length are in minute.

## Data Discovery

```
df <- read_csv("cleaned_data.csv")
```

```
## Rows: 2098 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Id, Channel, Title, URL, Category, Transcript
## dbl (5): Subscribers, CC, Released, Views, Length
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

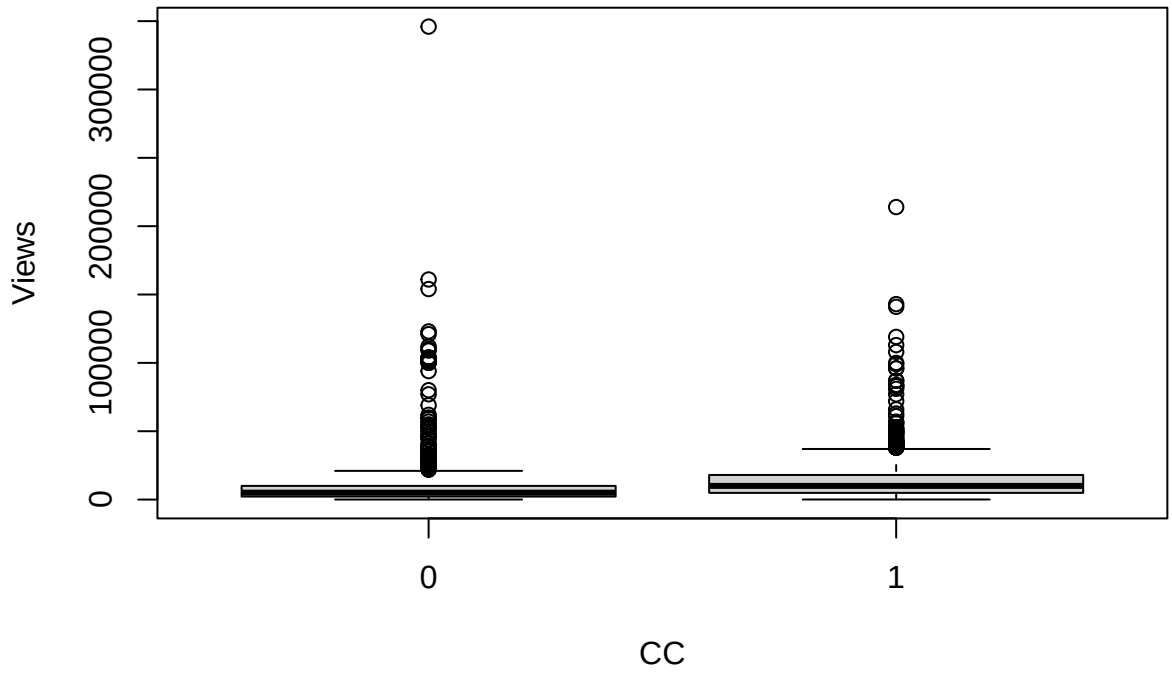
```
head(df)
```

```
## # A tibble: 6 x 11
##   Id    Channel Subscribers Title    CC URL   Released Views Category Transcript
##   <chr> <chr>         <dbl> <chr> <dbl> <chr>    <dbl> <dbl> <chr>    <chr>
## 1 FozC~ JRE Cl~        6280 Form~     0 http~       24  7900 Blog     "the Joe ~
## 2 Iugc~ Munchi~        4590 The ~     0 http~       24 11000 Food     "if you w~
## 3 JiEO~ Parks ~         282 Ron ~     0 http~       36  2300 Enterta~ "April wh~
## 4 1T4X~ Vsauce        17400 What~     1 http~      108 21000 Science  "Hey, Vsa~
## 5 OZWG~ Doctor~        1590 The ~     1 http~       84  8500 Enterta~ "Oh, what~
## 6 YiEj~ A&E            7930 Live~     1 http~       24 14000 News     "[music p~
## # ... with 1 more variable: Length <dbl>
```

```
table(df$CC)
```

```
##
##    0    1
## 1094 1004
```

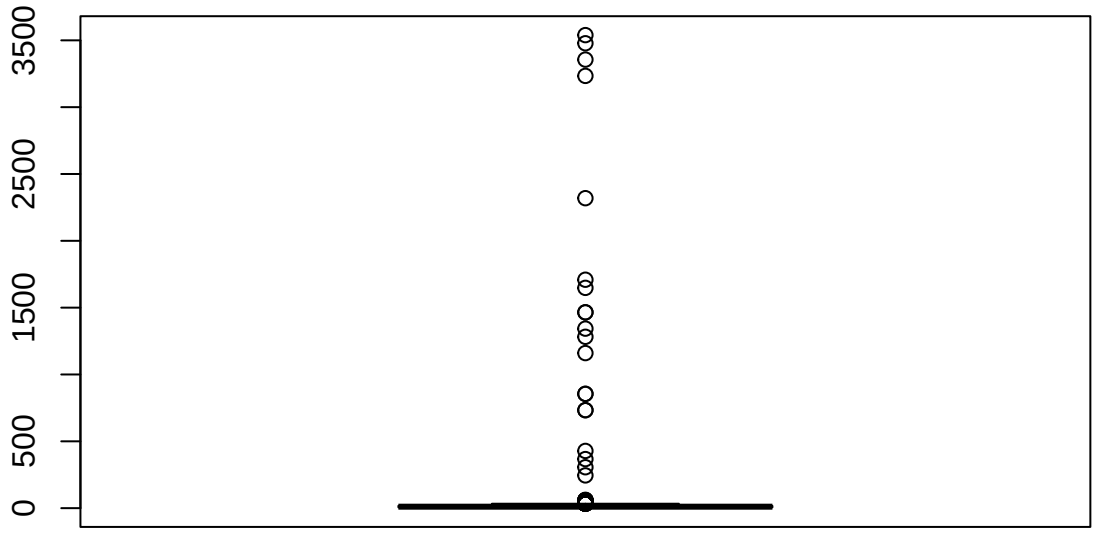
```
boxplot(Views ~ CC, data = df)
```




```
summary(df$Length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    6.00   11.00   26.94   17.00 3539.00
```

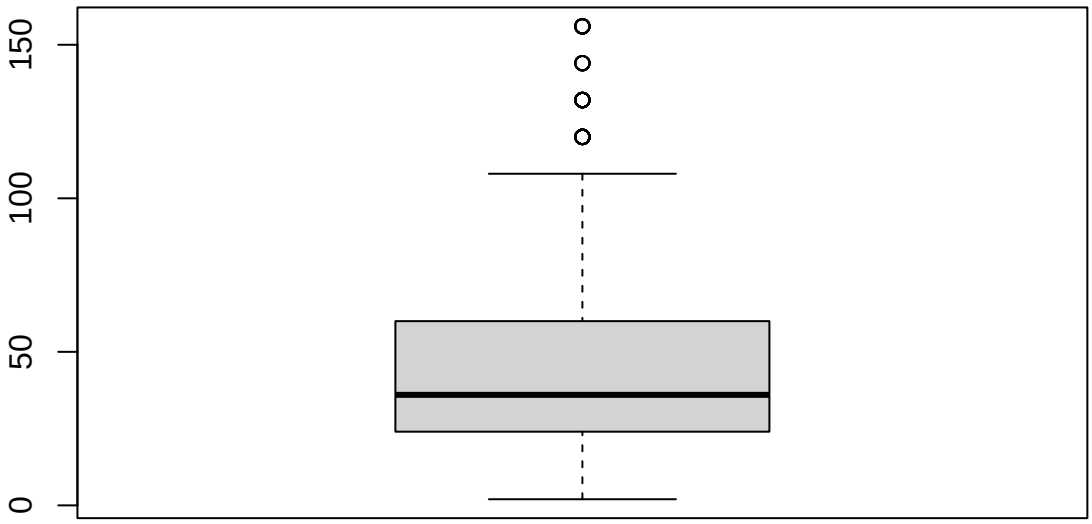
```
boxplot(df$Length)
```



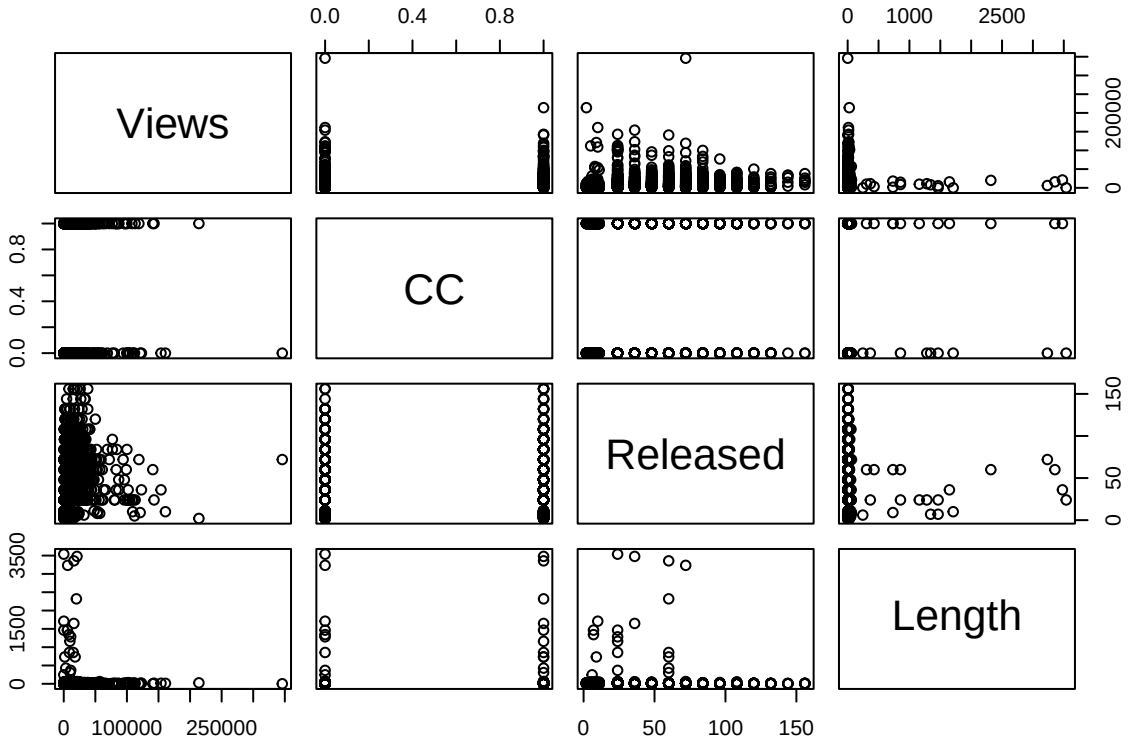

```
summary(df$Released)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   24.00   36.00   46.46   60.00  156.00
```

```
boxplot(df$Released)
```




```
pairs(Views ~ CC + Released + Length, data=df)
```




### Sentiment Analysis

TODO by Xinyi

### Linear Regression

TODO by Xinyi

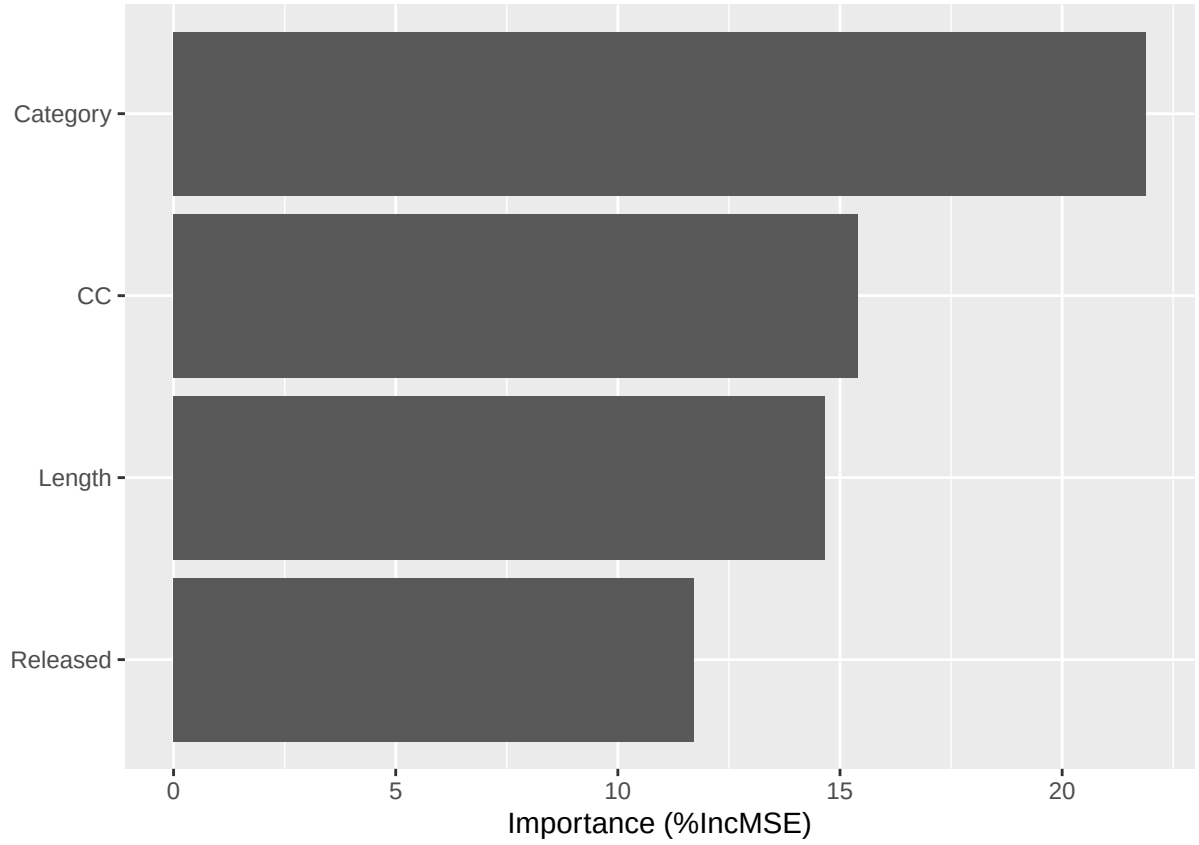
### Random Forest

```
set.seed(652)
rf1 <- randomForest(Views ~ CC + Released + Category + Length, importance = TRUE, data = df)
rf1
```

```
##
## Call:
##  randomForest(formula = Views ~ CC + Released + Category + Length,      data = df, importance = TRUE
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 239842215
##                     % Var explained: 26.06
```

Use the `vip()` function to make a variable importance plot. Which variables are most important?

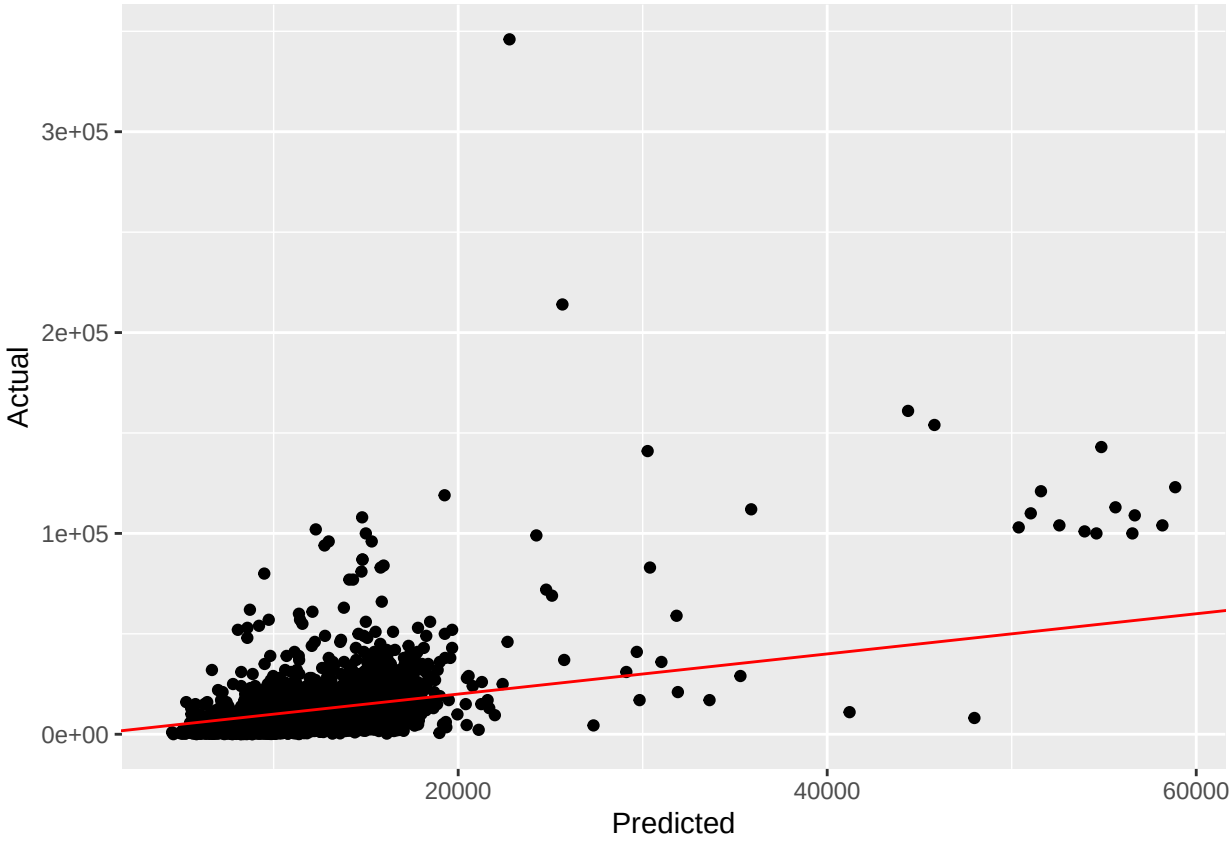
```
vip(rf1, num_features = 14,  include_type = TRUE)
```




Make a plot of the predicted versus actual values using the out-of-bag data. Add the 1-1 line to the plot.

```
pred_df <- data.frame(
  Actual = df$Views,
  Predicted = predict(rf1) # makes predictions using OOB data
)
```

```
ggplot(pred_df, aes(x = Predicted, y = Actual)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red")
```




## Conclusion

TODO