Public Figures

Nkosi Sampson

2024-08-15

```r
# Packages
library(tidyverse)
library(tidymodels)
library(tidyverse)
library(tidymodels)
library(recipes)
library(broom)
library(tidyclust)
library(mclust)
library(dplyr)
library(probably)
library(pROC)
library(ModelMetrics)
library(MASS) # LDA
# Data

# Regression Problem
public_figures <- readr::read_csv("public_figures.csv")
```

The public_figures.csv file contains information about 226 20th and 21st-century public figures. Please read the `public_figures_dictionary` file on on GitHub for a more complete description of the variables. The objective of this project is to build a model to predict the likability rating of a public figure, based primarily on their personality.

Part a

Immediately split data into a training set (75% of the rows) and test set (remaining 25%).

```r
set.seed(1)
pf_split <- initial_split(public_figures, prop = 0.75)
pf_train <- training(pf_split)
pf_test <- testing(pf_split)
```

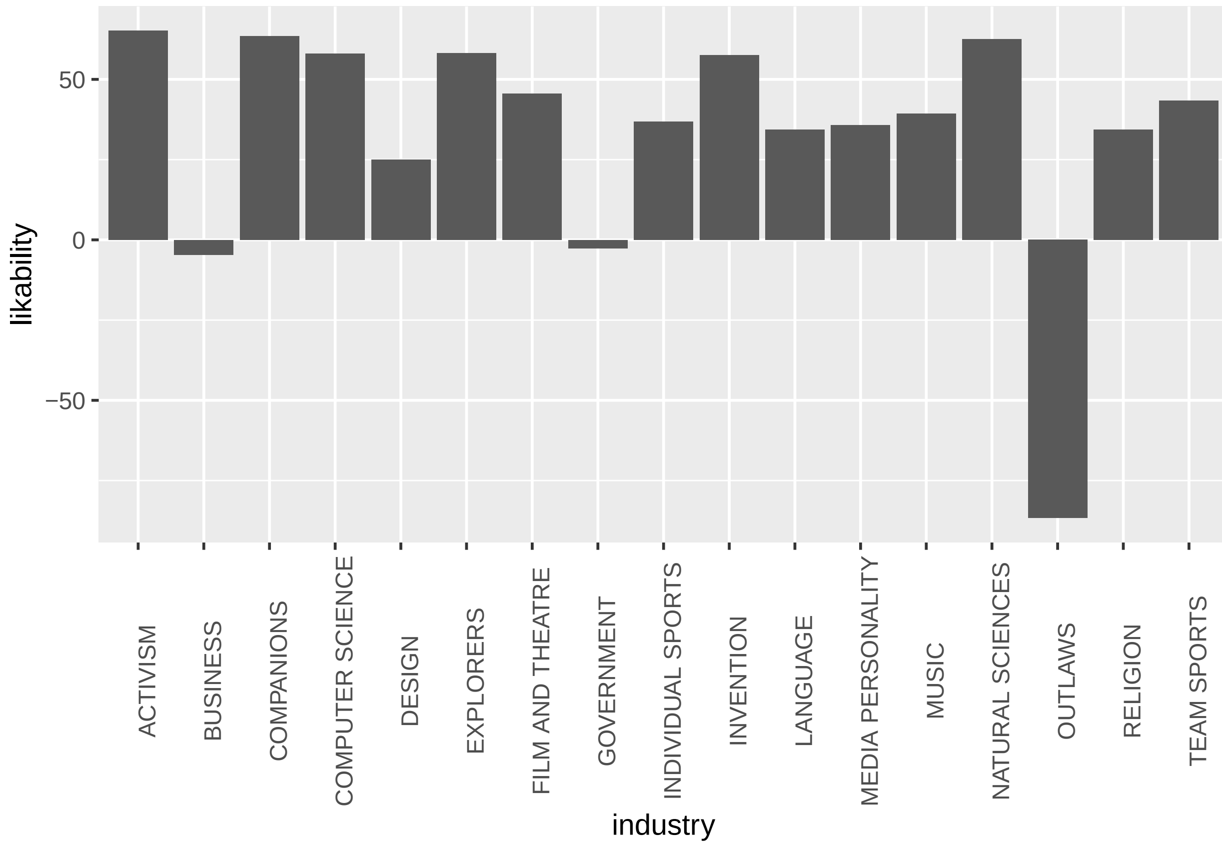
Ask questions about this dataset

1. *What industry is the most liked on average?*

```r
ggplot(aes(x = industry, y = likability), data = public_figures) + stat_summary(fun.y = "mean", geom =
```

```
## Warning: The `fun.y` argument of `stat_summary()` is deprecated as of ggplot2 3.3.0.
```

```
## i Please use the `fun` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

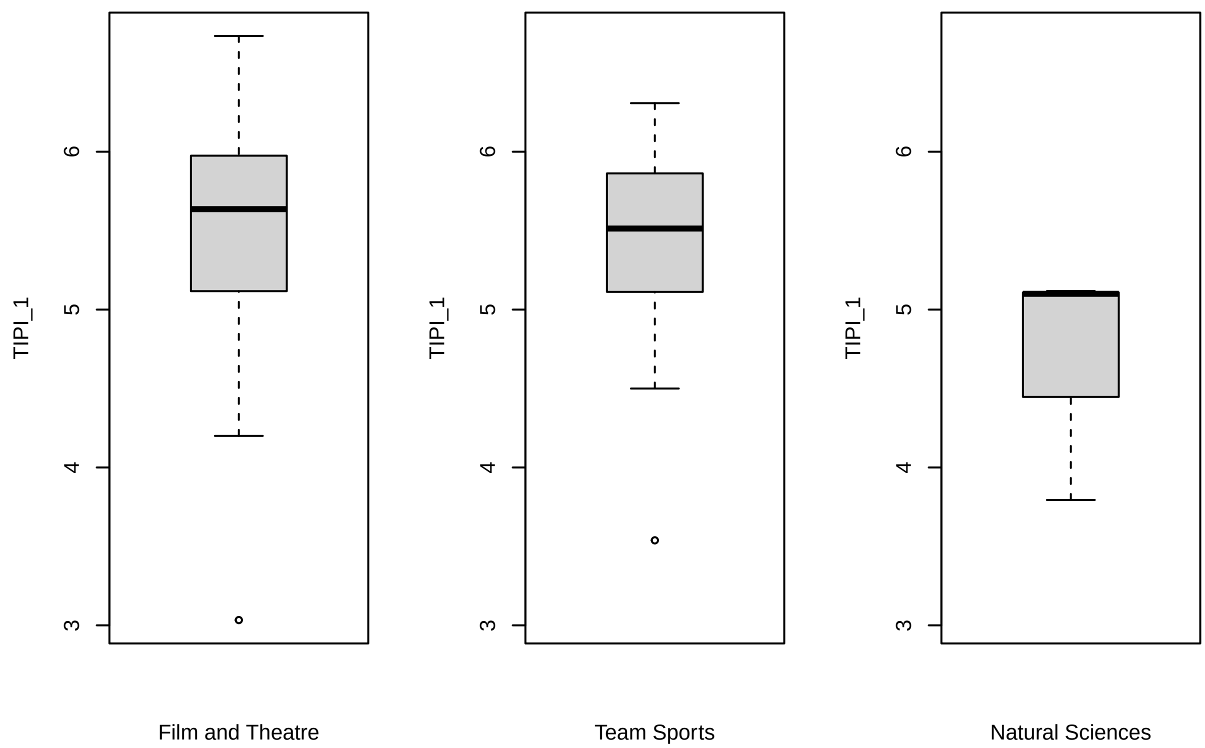


Surprisingly, Natural Sciences is at the top. I would expect "Team Sports" or "Film and Theater" to be at the top, because those are the things people are most engrossed in out of all these, in my experience. Maybe it's because while "Team Sports" and "Film and Theater" have the highest ratings, they also have some of the lowest ratings because there are some athletes and film and theater people who have some very undesirable attributes that are revealed because they are in the spotlight, whereas for people in the natural sciences, their personalities aren't always under the microscope, so people can't say many bad things about them.

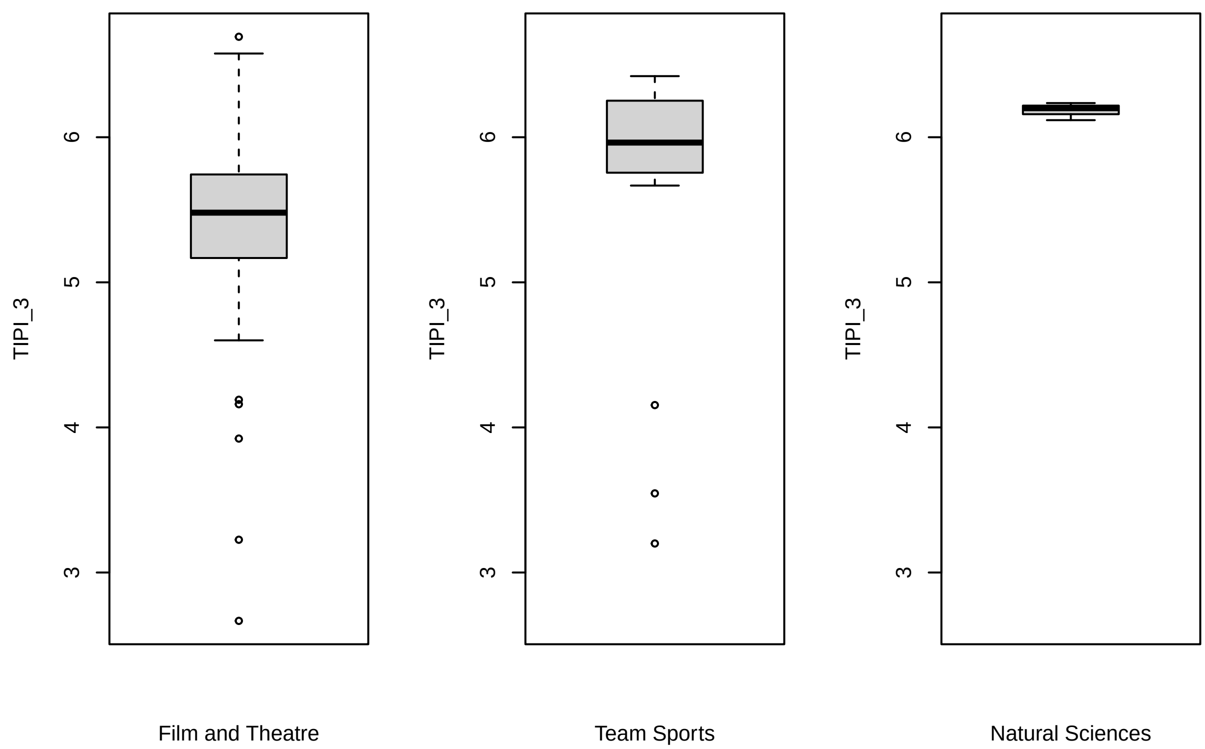
2. Of the favorable attributes, who has higher scores out of those in the "Team Sports", "Film and Theater", and "Natural Sciences" industries? *For the favorable attributes, I will use the ones that are clearly favorable : `TIPI_1`:"Extroverted, enthusiastic", `TIPI_3`: "Dependable, self-disciplined", `TIPI_7`: "Sympathetic, warm", `TIPI_9`: "Calm, emotionally stable"

```r
FilmTheatre <- public_figures %>%
  filter(industry == "FILM AND THEATRE")
TeamSports <- public_figures %>%
  filter(industry == "TEAM SPORTS")
NaturalSciences <- public_figures %>%
  filter(industry == "NATURAL SCIENCES")


# Determine the overall y-axis limits
```

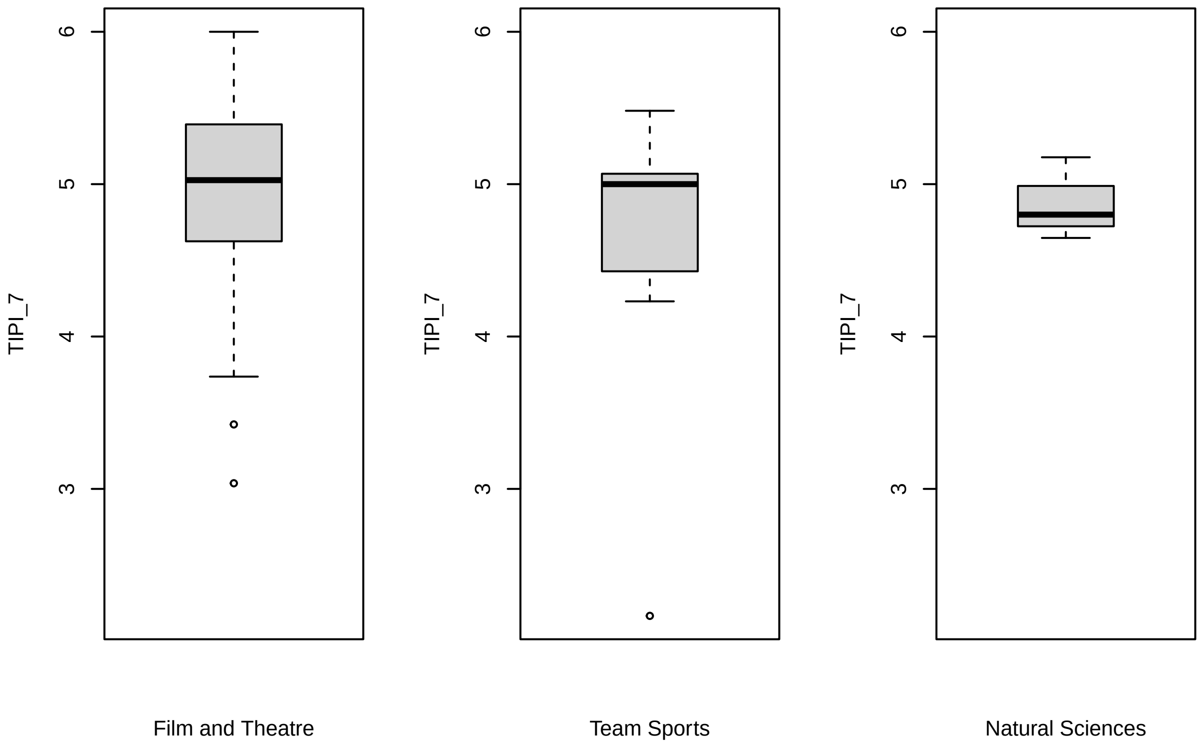
```r
y_limits_TIPI_1 <- range(c(FilmTheatre$TIPI_1, TeamSports$TIPI_1, NaturalSciences$TIPI_1), na.rm = TRUE)
y_limits_TIPI_3 <- range(c(FilmTheatre$TIPI_3, TeamSports$TIPI_3, NaturalSciences$TIPI_3), na.rm = TRUE)
y_limits_TIPI_7 <- range(c(FilmTheatre$TIPI_7, TeamSports$TIPI_7, NaturalSciences$TIPI_7), na.rm = TRUE)
y_limits_TIPI_9 <- range(c(FilmTheatre$TIPI_9, TeamSports$TIPI_9, NaturalSciences$TIPI_9), na.rm = TRUE)

# TIPI_1
par(mfrow = c(1,3))
boxplot(FilmTheatre$TIPI_1, xlab = "Film and Theatre", ylab = "TIPI_1", ylim = y_limits_TIPI_1)
boxplot(TeamSports$TIPI_1, xlab = "Team Sports", ylab = "TIPI_1", ylim = y_limits_TIPI_1)
boxplot(NaturalSciences$TIPI_1, xlab = "Natural Sciences", ylab = "TIPI_1", ylim = y_limits_TIPI_1)
```



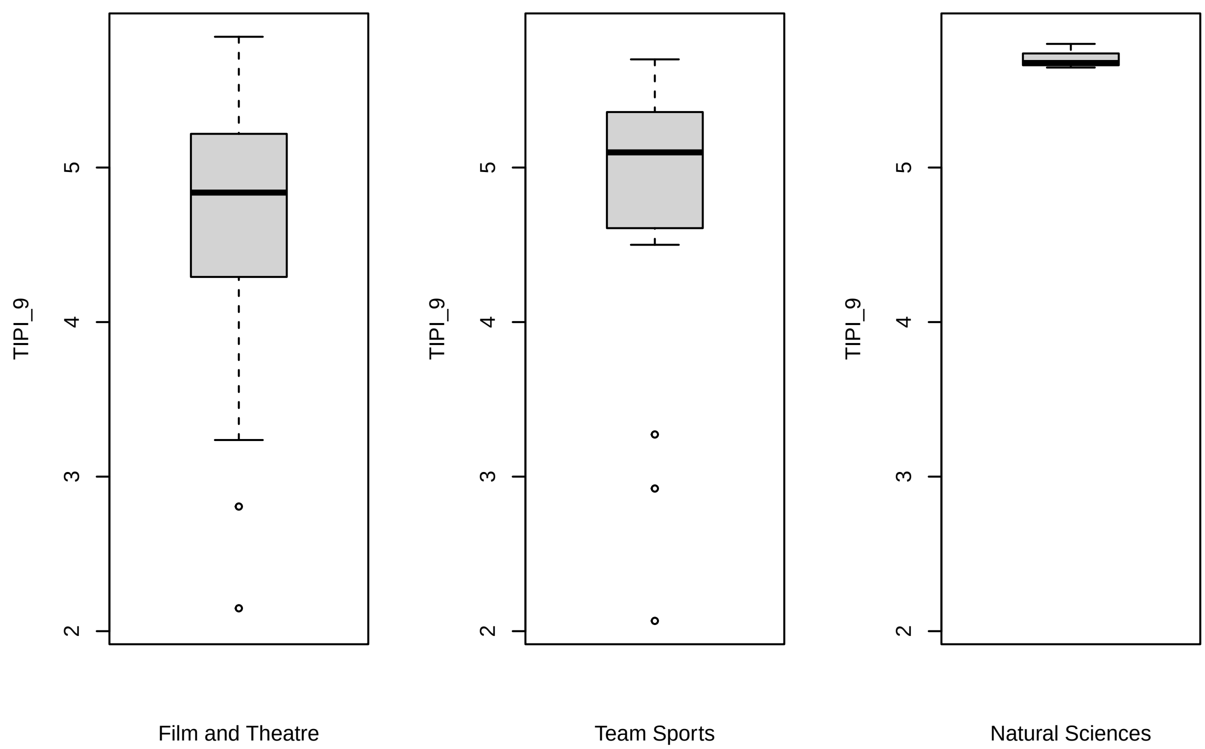

```r
# TIPI_3
par(mfrow = c(1,3))
boxplot(FilmTheatre$TIPI_3, xlab = "Film and Theatre", ylab = "TIPI_3", ylim = y_limits_TIPI_3)
boxplot(TeamSports$TIPI_3, xlab = "Team Sports", ylab = "TIPI_3", ylim = y_limits_TIPI_3)
boxplot(NaturalSciences$TIPI_3, xlab = "Natural Sciences", ylab = "TIPI_3", ylim = y_limits_TIPI_3)
```




```r
# TIPI_7
par(mfrow = c(1,3))
boxplot(FilmTheatre$TIPI_7, xlab = "Film and Theatre", ylab = "TIPI_7", ylim = y_limits_TIPI_7)
boxplot(TeamSports$TIPI_7, xlab = "Team Sports", ylab = "TIPI_7", ylim = y_limits_TIPI_7)
boxplot(NaturalSciences$TIPI_7, xlab = "Natural Sciences", ylab = "TIPI_7", ylim = y_limits_TIPI_7)
```



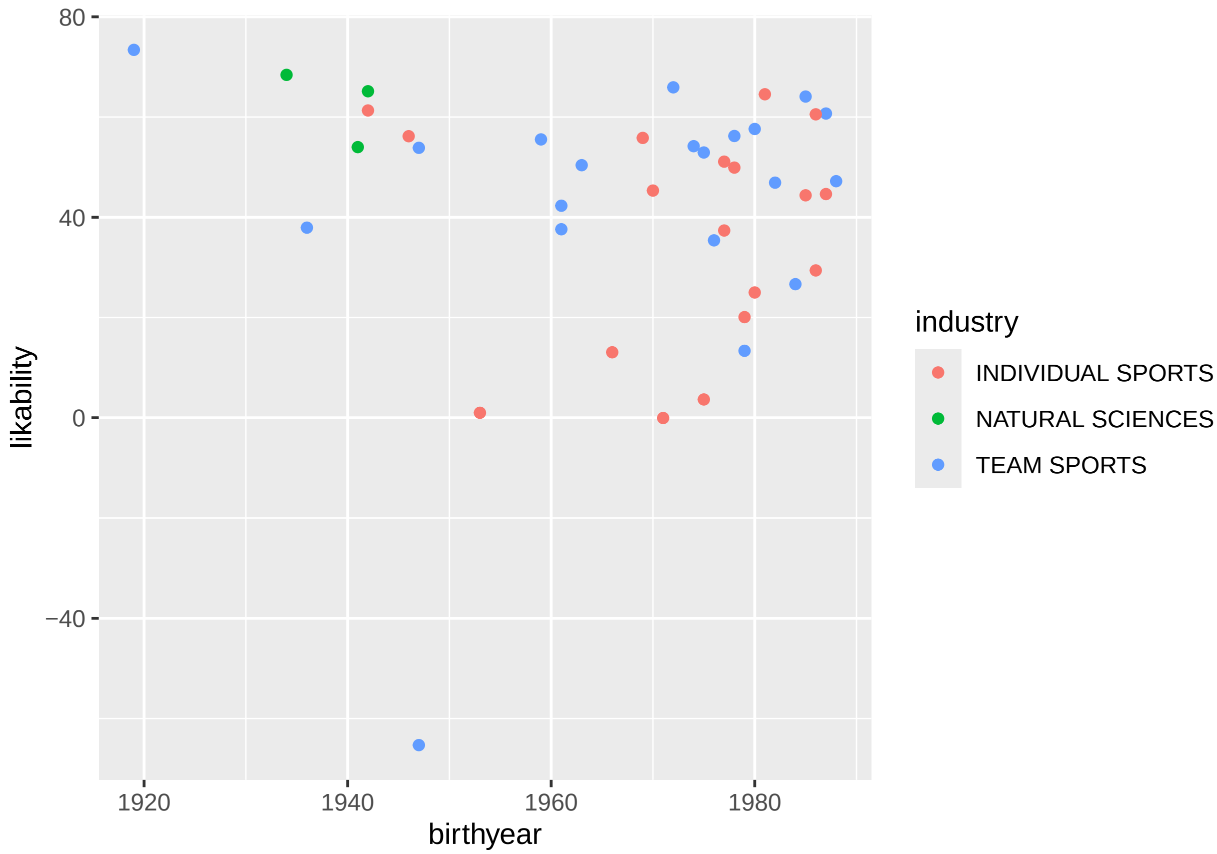

```r
# TIPI_9
par(mfrow = c(1,3))
boxplot(FilmTheatre$TIPI_9, xlab = "Film and Theatre", ylab = "TIPI_9", ylim = y_limits_TIPI_9)
boxplot(TeamSports$TIPI_9, xlab = "Team Sports", ylab = "TIPI_9", ylim = y_limits_TIPI_9)
boxplot(NaturalSciences$TIPI_9, xlab = "Natural Sciences", ylab = "TIPI_9", ylim = y_limits_TIPI_9)
```



These results make sense. For `TIPI_1`: "Extroverted, enthusiastic", I would expect athletes and actors/actresses to be seen as having this quality more than those in the natural sciences. For `TIPI_3`: "Dependable, self-disciplined", I would expect those in the natural sciences and team sports to be seen as more dependable and self disciplined than those in acting, because self-discipline is crucial for maintaining peak physical condition, and dependability is needed to become well-known int he natural sciences. For `TIPI_7`: "Sympathetic, warm", I don;t picture people in the natural sciences as warm and sympathetic, becasue we generally don;t see that side of people who are famous for that profession. I would expect to see those in the film and theater occupation as the clear leader for this category, but here the results are pretty similar. For `TIPI_9`: "Calm, emotionally stable", we see that natural sciences far surpasses the other two categories, because even if these people are not calm and emotionally stable, we generally don't see that type of behavior publicized.

3. I'll bet that as age increases for those in the "TEAM SPORTS" or "INDIVIDUAL SPORTS" categories, likability increases faster than in the "NATURAL SCIENCES" category, because I feel like people like retired athletes much more than active athletes, because retired athletes can't threaten your team's playoffs hopes and aren't always in the headlines for doing bad stuff on the field/court.

```
pfq4 <- public_figures %>%
  filter(industry == "TEAM SPORTS" |
           industry == "INDIVIDUAL SPORTS" |
           industry == "NATURAL SCIENCES") %>%
  dplyr::select(industry, likability, birthyear)

ggplot(pfq4, aes(x = birthyear, y = likability, color = industry)) + geom_point()
```



There is a problem with a lack of observations for the `NATURAL SCIENCES` industry, which explains why their median rating for `TIPI_1` is so high. As it relates to my hypothesis from this question, the likability for those in Team Sports does seem to increase with age slightly, mostly due to a couple of points near the top left corner, and the likability of those in Natural Sciences also seems to increase with age, but again, there's only 3 samples, so it's hard to make any judgments on this. Also, there's a clear outlier in the bottom of the graph. I wonder what athlete is disliked that much? Mike Tyson for biting Evander Holyfield's ear?

```r
filter(public_figures,
       likability < -50 & industry == "TEAM SPORTS")
```

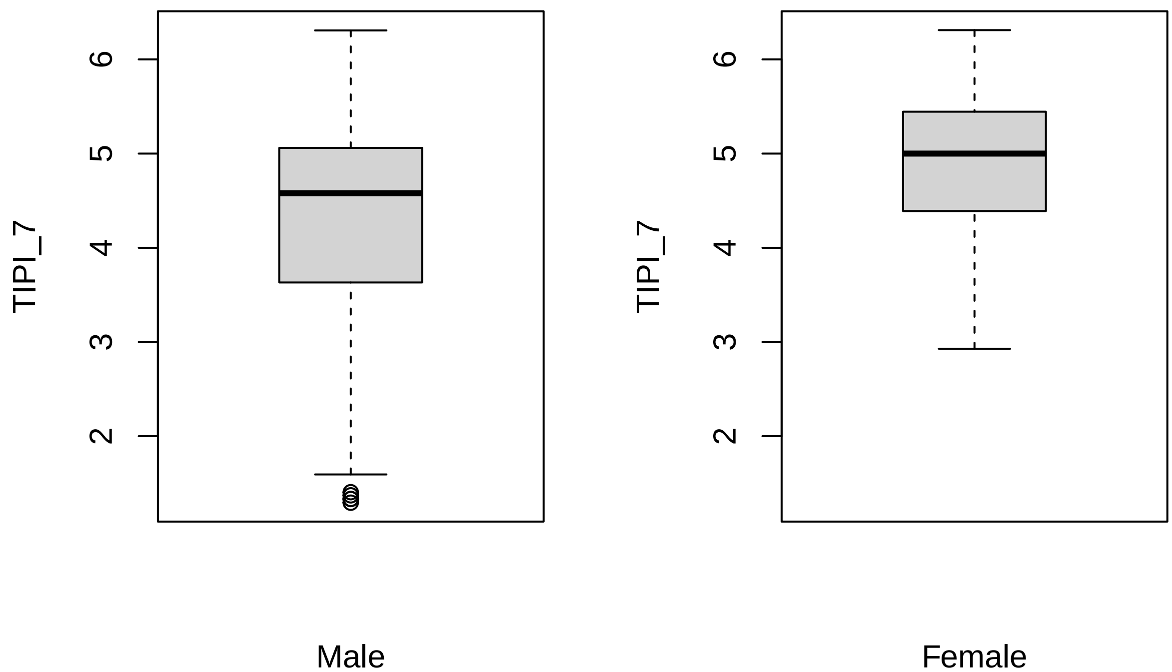
```
## # A tibble: 1 x 18
##   name        gender birthyear n_raters occupation industry TIPI_1 TIPI_2 TIPI_3
##   <chr>       <chr>      <dbl>    <dbl> <chr>      <chr>     <dbl>  <dbl>  <dbl>
## 1 O. J. Simp~ Male        1947       30 AMERICAN ~ TEAM SP~   5.13   5.77    3.2
## # i 9 more variables: TIPI_4 <dbl>, TIPI_5 <dbl>, TIPI_6 <dbl>, TIPI_7 <dbl>,
## #   TIPI_8 <dbl>, TIPI_9 <dbl>, TIPI_10 <dbl>, likability <dbl>,
## #   pred_likability <dbl>
```

That makes sense. I guess a lot of people think he was guilty of the crime he was charged with.

4. *I think that girls will have higher ratings for "Sympathetic, warm" than guys, because girls usually act that way more than guys. I will check overall male/female comparison.

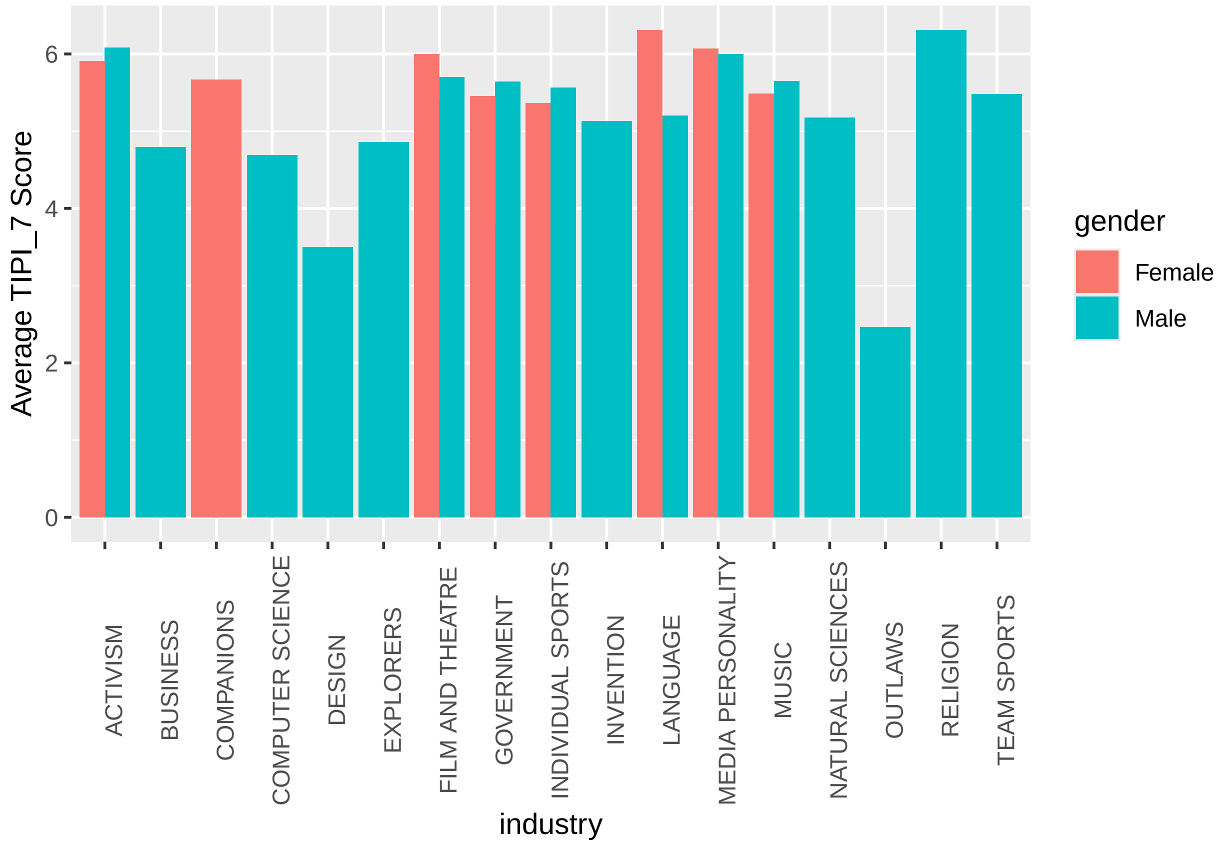
```r
Male <- public_figures %>%
  filter(gender == "Male")
```

```r
Female <- public_figures %>%
  filter(gender == "Female")


# Determine the overall y-axis limits for TIPI_7
y_limits_TIPI_7 <- range(c(Male$TIPI_7, Female$TIPI_7), na.rm = TRUE)

# Create the boxplots with the same scale
par(mfrow = c(1,2))  # Adjust to c(1,2) since there are two plots
boxplot(Male$TIPI_7, xlab = "Male", ylab = "TIPI_7", ylim = y_limits_TIPI_7)
boxplot(Female$TIPI_7, xlab = "Female", ylab = "TIPI_7", ylim = y_limits_TIPI_7)
```



As expected, the females tend to have higher rating for `TIPI_7` *than do makes. We will check among each industry, though, because the disparities in some industries could make the difference in rating seem more pronounced than it actually is.*

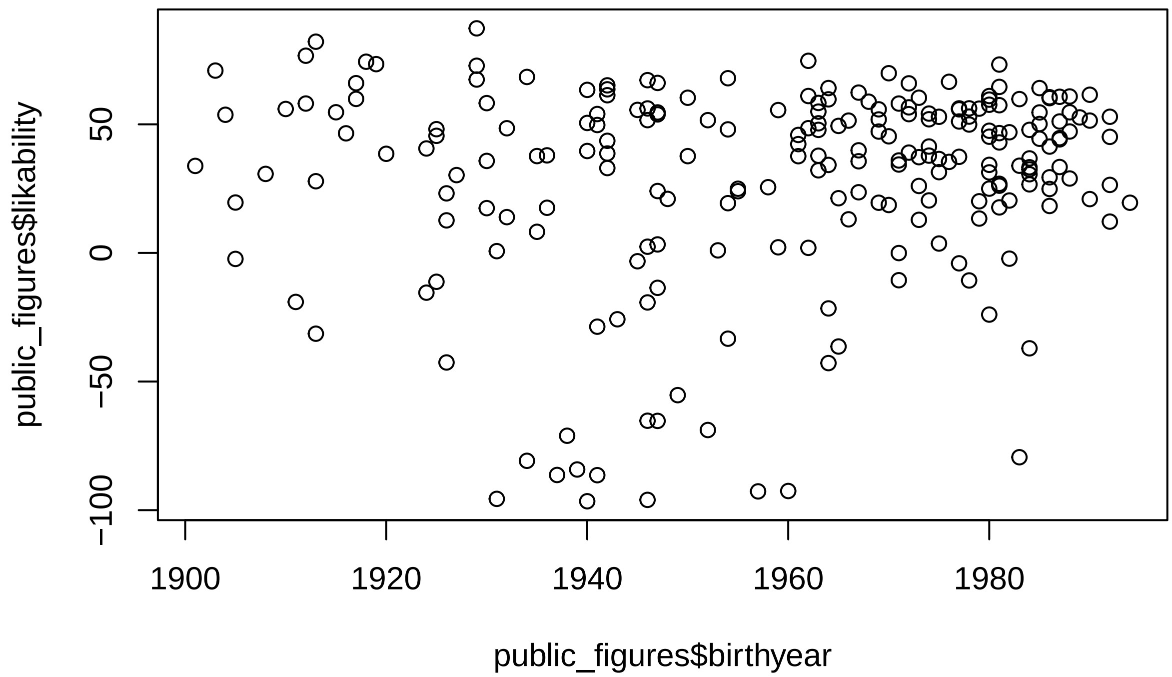
```r
ggplot(public_figures, aes(x = industry, y = TIPI_7, fill = gender)) +
  geom_bar(position = "dodge", stat = "identity") +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(y = "Average TIPI_7 Score")
```



For 3/7 categories that have males and females, the average rating for "`TIPI_7`:"Sympathetic, warm" was higher for females than it was for males. In the "LANGUAGE" category, the `TIPI_7` rating is considerable higher for females compared to males, and in the other industries where there is a difference, the difference is very small. Therefore, it's likely that this industry is the main contributor to the overall difference we see in the `TIPI_7` rating between males and females. Surprisingly, there were no females in the "NATURAL SCIENCES" and "DESIGN" category, because I know there are a lot of important women in those 2 categories.

5. In general, how does the variability of likability progress with age? I feel like on average, younger people tend to be observed more than older people in the media, so i feel like the older people in the dataset will have less variability on average than younger people in the data set.

```
plot(public_figures$birthyear, public_figures$likability)
```




```
min(public_figures$birthyear)
```

```
## [1] 1901
```

```
public_figures_1 <- filter(public_figures, birthyear >= 1900 & birthyear <= 1920)

public_figures_2 <- filter(public_figures, birthyear > 1920 & birthyear <= 1940)

public_figures_3 <- filter(public_figures, birthyear > 1940 & birthyear <= 1960)

public_figures_4 <- filter(public_figures, birthyear > 1960 & birthyear <= 1980)

public_figures_1 <- filter(public_figures, birthyear > 1980 & birthyear <= 2000)

var(public_figures_1$likability)
```

```
## [1] 677.6304
```

```
var(public_figures_2$likability)
```

```
## [1] 2961.017
```

```
var(public_figures_3$likability)
```

```
## [1] 2579.618
```

```
var(public_figures_4$likability)
```

```
## [1] 612.1762
```

So the variance of likability is much higher for those born from 1920-1960 than for those born from 1900-1920 and 1980-2000. Of course, that lends to the question of why were those people from 1900-1920 that were selected selected. They probably wouldn't have been be selected if they were just popular for a short amount of time, like I'm sure a lot of the people who are younger are. For example, no one is gonna forget an Adolf Hitler for a long time, but i bet it won't be long before people forget who Megan Fox is.

Part b

Continue exploratory data analysis by performing a principal component analysis on all 10 `TIPI` variables.

```
pfigures <- public_figures %>%
  dplyr::select(name, TIPI_1, TIPI_2, TIPI_3, TIPI_4, TIPI_5, TIPI_6, TIPI_7, TIPI_8, TIPI_9, TIPI_10)
```

```
pca_recipe <- recipe(
  ~ ., data = pfigures
) |>
## ~ . indicates to use all variables in the dataset as predictors
   update_role(name, new_role = "id") |>
  step_normalize(all_numeric_predictors()) |>
  step_pca(all_predictors(), num_comp = 10)
```

```
pca_prep <- pca_recipe |>
  prep()
pca_prep
```

```
##

## -- Recipe ----------------------------------------------------------------------

##

## -- Inputs

## Number of variables by role

## predictor: 10
## id:         1

##
```

```
## -- Training information

## Training data contained 226 data points and no incomplete rows.

##

## -- Operations

## * Centering and scaling for: TIPI_1, TIPI_2, TIPI_3, TIPI_4, ... | Trained

## * PCA extraction with: TIPI_1, TIPI_2, TIPI_3, TIPI_4, TIPI_5, ... | Trained
```

```
pca_baked <- pca_prep |>
  bake(new_data = NULL)
pca_baked
```

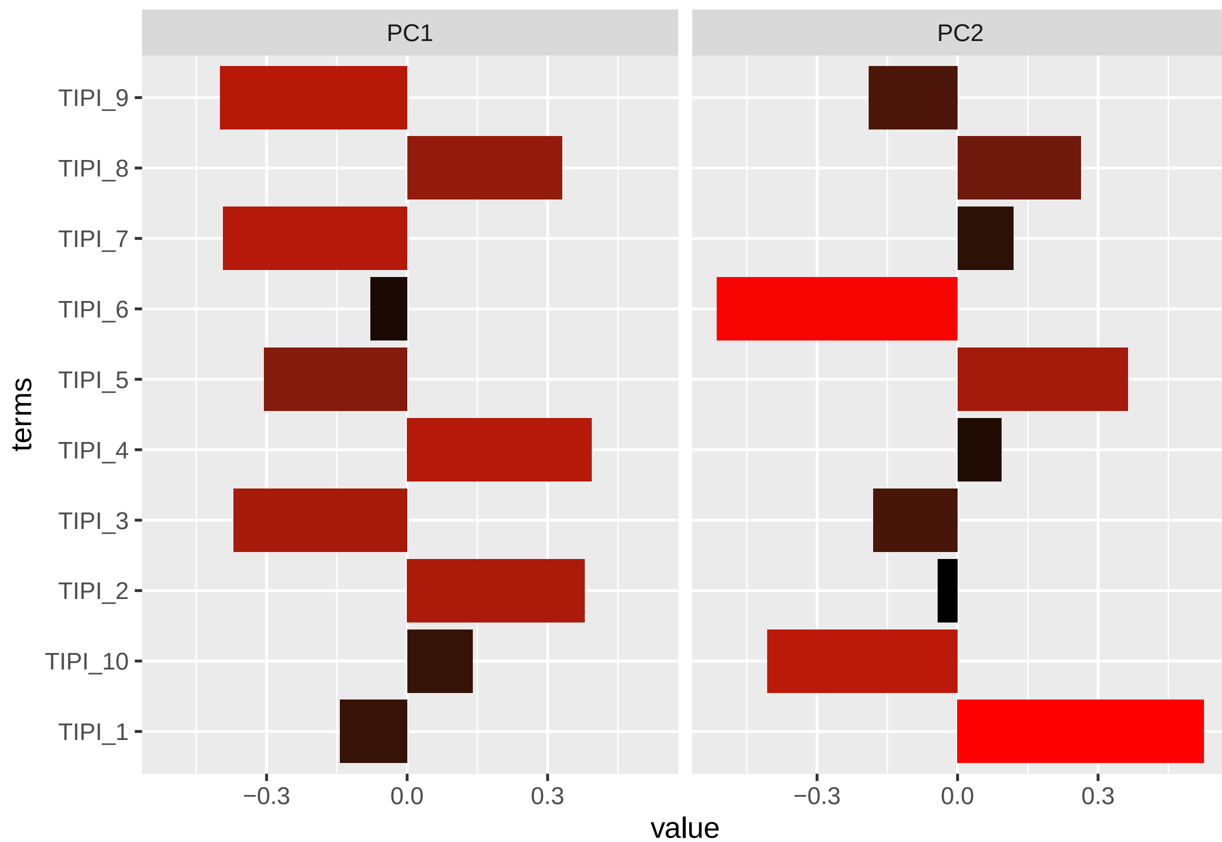
```
## # A tibble: 226 x 11
##    name       PC01    PC02   PC03    PC04    PC05    PC06   PC07     PC08    PC09
##    <fct>     <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>  <dbl>    <dbl>   <dbl>
##  1 Barack~ -2.62  -0.456   1.12   0.0717  0.0910 -0.0640  0.134  0.406    0.0681
##  2 Kim Ka~  2.93   1.79    0.913 -0.328  -0.136  -0.291   0.624  0.0469  -0.0827
##  3 Mark Z~  2.15  -3.86   -1.16   0.915   0.838  -0.146   0.456  0.00854  0.0722
##  4 Cristi~ -0.798  0.0106  0.849  0.162  -0.574  -0.392  -0.411 -0.230    0.462
##  5 John F~ -1.96  -0.0339  0.374 -0.316   0.345  -0.0208 -0.162  0.154    0.261
##  6 Nelson~ -3.23  -1.07   -0.176  0.311   0.256   0.135  -0.249  0.672   -0.246
##  7 Michae~ -1.33  -0.682   0.814  0.771  -0.0565 -0.0774 -0.234 -0.344    0.148
##  8 Lionel~ -1.48  -1.05    0.571  0.148   0.263   0.347  -0.270 -0.0558   0.506
##  9 Bill G~ -1.58  -1.98   -1.22   1.27    0.415   0.132   0.123 -0.0476  -0.0676
## 10 George~  2.74  -2.26    0.723 -2.04    0.780  -0.248  -0.384  0.205    0.0773
## # i 216 more rows
## # i 1 more variable: PC10 <dbl>
```

```
pca_tidy <- tidy(pca_prep, 2, type = "coef") # tidy step 3 - the PCA step
head(pca_tidy, 20)
```

```
## # A tibble: 20 x 4
##    terms     value component id
##    <chr>     <dbl> <chr>     <chr>
##  1 TIPI_1  -0.144  PC1       pca_EKC19
##  2 TIPI_2   0.379  PC1       pca_EKC19
##  3 TIPI_3  -0.371  PC1       pca_EKC19
##  4 TIPI_4   0.394  PC1       pca_EKC19
##  5 TIPI_5  -0.306  PC1       pca_EKC19
##  6 TIPI_6  -0.0780 PC1       pca_EKC19
##  7 TIPI_7  -0.393  PC1       pca_EKC19
##  8 TIPI_8   0.331  PC1       pca_EKC19
##  9 TIPI_9  -0.399  PC1       pca_EKC19
## 10 TIPI_10  0.140  PC1       pca_EKC19
## 11 TIPI_1   0.526  PC2       pca_EKC19
## 12 TIPI_2  -0.0425 PC2       pca_EKC19
## 13 TIPI_3  -0.180  PC2       pca_EKC19
```

```
## 14 TIPI_4   0.0937 PC2       pca_EKC19
## 15 TIPI_5   0.363  PC2       pca_EKC19
## 16 TIPI_6  -0.514  PC2       pca_EKC19
## 17 TIPI_7   0.120  PC2       pca_EKC19
## 18 TIPI_8   0.263  PC2       pca_EKC19
## 19 TIPI_9  -0.190  PC2       pca_EKC19
## 20 TIPI_10 -0.406  PC2       pca_EKC19
```

```
pca_tidy |>
  filter(component %in% c("PC1", "PC2")) |>
  ggplot(aes(x = value, y = terms, fill = abs(value))) +
  geom_col() +
  theme(legend.position = "none") +
  scale_fill_gradient(low = "black", high = "red") +
  facet_wrap(vars(component))
```




```
pca_loadings <- pca_tidy |>
  pivot_wider(names_from = "component",
              values_from = "value") |>
  dplyr::select(!id)
arrange(pca_loadings, desc(abs(PC1)))
```

```
## # A tibble: 10 x 11
##    terms       PC1     PC2     PC3     PC4     PC5     PC6      PC7     PC8
##    <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>    <dbl>   <dbl>
```

```
##  1 TIPI_9  -0.399  -0.190   0.142  -0.0944  0.262  -0.199   0.0126   0.384
##  2 TIPI_4   0.394   0.0937 -0.231   0.0724 -0.715  -0.213   0.117    0.118
##  3 TIPI_7  -0.393   0.120  -0.0887 -0.353  -0.381  -0.101  -0.377    0.514
##  4 TIPI_2   0.379  -0.0425  0.177   0.474   0.222  -0.466  -0.197    0.431
##  5 TIPI_3  -0.371  -0.180   0.158   0.405  -0.248  -0.337  -0.424   -0.475
##  6 TIPI_8   0.331   0.263  -0.153  -0.505   0.310  -0.366  -0.455   -0.251
##  7 TIPI_5  -0.306   0.363  -0.311   0.0753  0.132  -0.540   0.553   -0.0523
##  8 TIPI_1  -0.144   0.526   0.403  -0.0508 -0.147  -0.0429  0.00181 -0.225
##  9 TIPI_10  0.140  -0.406   0.561  -0.440  -0.169  -0.341   0.337   -0.0862
## 10 TIPI_6  -0.0780 -0.514  -0.514  -0.125  -0.0124 -0.176  -0.0123  -0.198
## # i 2 more variables: PC9 <dbl>, PC10 <dbl>
```

```
arrange(pca_loadings, desc(abs(PC2)))
```

```
## # A tibble: 10 x 11
##    terms      PC1     PC2     PC3     PC4     PC5     PC6      PC7     PC8
##    <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>    <dbl>   <dbl>
##  1 TIPI_1  -0.144   0.526   0.403  -0.0508 -0.147  -0.0429  0.00181 -0.225
##  2 TIPI_6  -0.0780 -0.514  -0.514  -0.125  -0.0124 -0.176  -0.0123  -0.198
##  3 TIPI_10  0.140  -0.406   0.561  -0.440  -0.169  -0.341   0.337   -0.0862
##  4 TIPI_5  -0.306   0.363  -0.311   0.0753  0.132  -0.540   0.553   -0.0523
##  5 TIPI_8   0.331   0.263  -0.153  -0.505   0.310  -0.366  -0.455   -0.251
##  6 TIPI_9  -0.399  -0.190   0.142  -0.0944  0.262  -0.199   0.0126   0.384
##  7 TIPI_3  -0.371  -0.180   0.158   0.405  -0.248  -0.337  -0.424   -0.475
##  8 TIPI_7  -0.393   0.120  -0.0887 -0.353  -0.381  -0.101  -0.377    0.514
##  9 TIPI_4   0.394   0.0937 -0.231   0.0724 -0.715  -0.213   0.117    0.118
## 10 TIPI_2   0.379  -0.0425  0.177   0.474   0.222  -0.466  -0.197    0.431
## # i 2 more variables: PC9 <dbl>, PC10 <dbl>
```

Based on the descriptions of the variables in the data dictionary, the first principal component represents public figures who are viewed as either having non-favorable personality traits or favorable, as there are strong positive coefficients for unfavorable value like `Anxious, easily upset` and `Critical, quarrelsome` and strong negative coefficients for `Calm, emotionally stable` and `Sympathetic, warm`. High positive scores for PC1 correspond to non-favorable attributes, and high negative scores on PC1 correspond to favorable attributes.

Principal Component 2 tells shows us the public figures who are either very extroverted or very introverted, because these are the two qualities that have massive coefficients. High positive scores for PC2 correspond to extrovert qualities, as `TIPI_1` is "Extroverted, enthusiastic" and `TIPI_5` is "Open to new experiences, complex". High negative scores correspond to introverted qualities, as `TIPI_6` is "Reserved, quiet" and `TIPI_10` is "Conventional, uncreative".

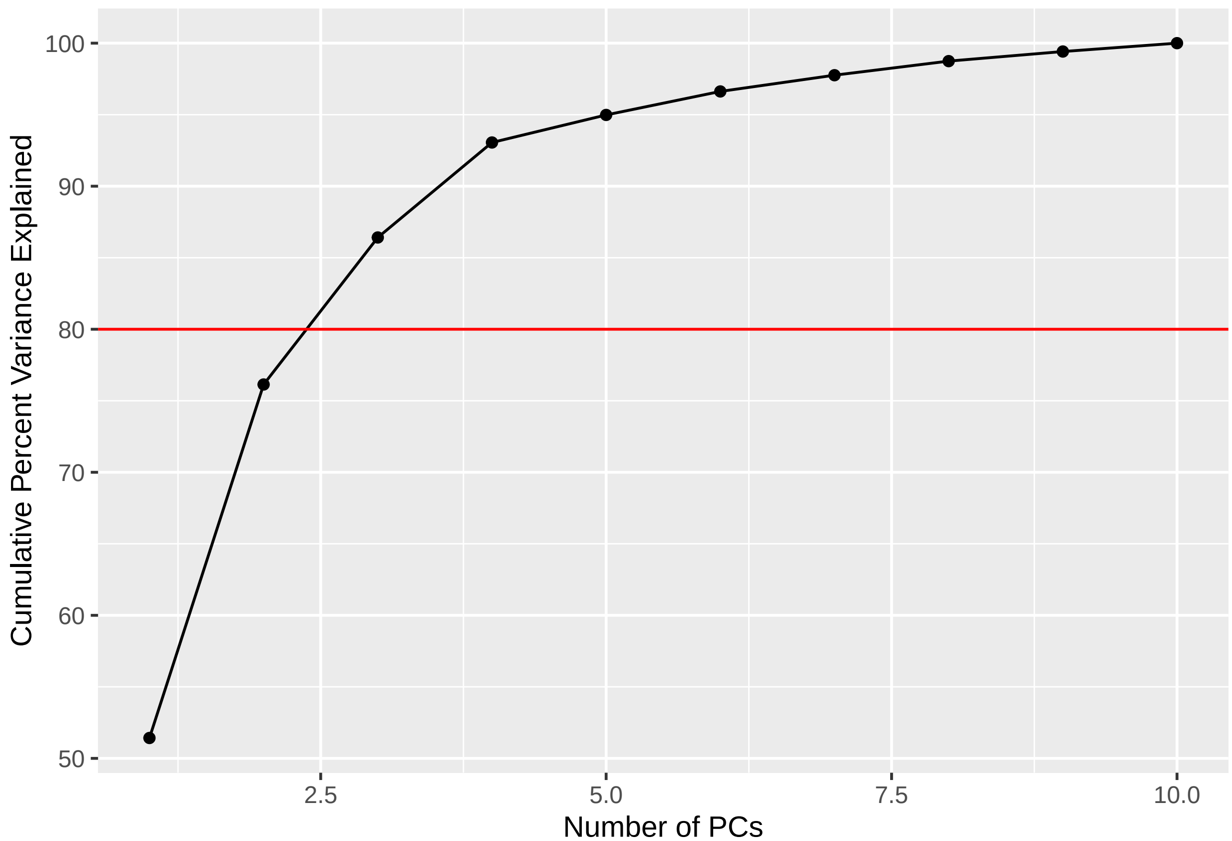
- If I want to reduce the 10 TIPI variables, how many principal components should I choose?

```
pca_pve <- tidy(pca_prep, type = "variance", number = 2) # Step 3 - PCA step

filter(pca_pve, (component == "1" |
           component == "2" |
           component == "3") &
           terms == "percent variance")
```

```
## # A tibble: 3 x 4
```

```
##   terms              value component id
##   <chr>              <dbl>     <int> <chr>
## 1 percent variance  51.4         1 pca_EKCl9
## 2 percent variance  24.7         2 pca_EKCl9
## 3 percent variance  10.3         3 pca_EKCl9
```

```
ggplot(pca_pve |> filter(terms == "cumulative percent variance"),
       aes(x = component, y = value)) +
  geom_point() +
  geom_line() +
  geom_hline(yintercept = 80, color = "red") +
  labs(x = "Number of PCs",
       y = "Cumulative Percent Variance Explained")
```



3 principal components is appropriate to interpret the data, because that is the minimum amount of principal components such that the cumulative PVE is >= 80%.

Part c

Continue exploratory data analysis by performing a cluster analysis using the `TIPI` variables.

```
kmeans_recipe <- recipe(~ TIPI_6 + TIPI_7 + TIPI_8 + TIPI_9 + TIPI_10,
                        data = pf_train) |>
  step_YeoJohnson(all_numeric_predictors()) |> # deal with skew issues
  step_zv(all_predictors())
```

```r
kmeans_model <- k_means(num_clusters = tune()) |>
  set_args(nstart = 20)
```

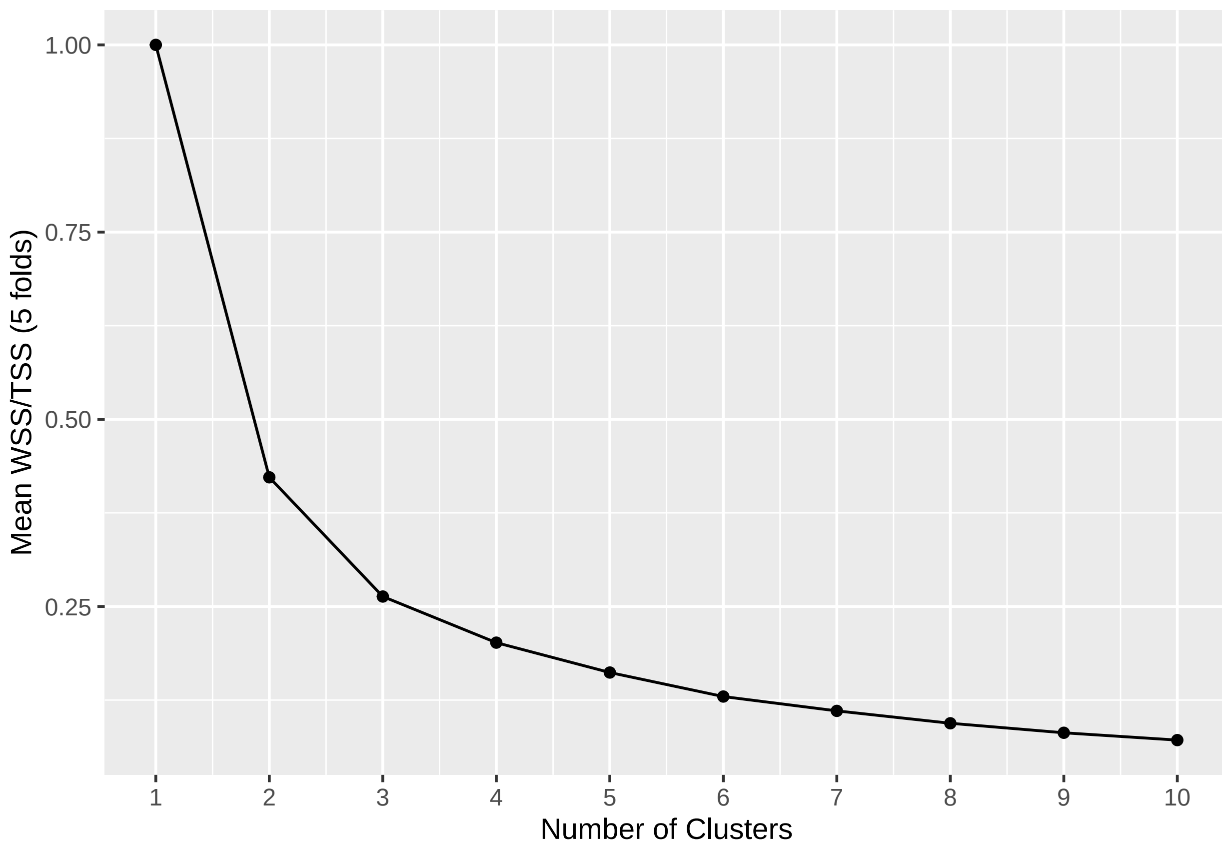
```r
kmeans_wflow <- workflow() |>
  add_model(kmeans_model) |>
  add_recipe(kmeans_recipe)
```

```r
set.seed(1002)
kfold_tidy <- vfold_cv(pf_train, v = 5, repeats = 1)
# grid is now expected to be a tibble or data frame instead of a list of named parameters
nclusters_grid <- data.frame(num_clusters = seq(1, 10))

kmeans_tuned <- tune_cluster(kmeans_wflow,
                             resamples = kfold_tidy,
                             metrics = cluster_metric_set(sse_total,
                                                          sse_within_total, sse_ratio),
                             grid = nclusters_grid)

tuned_metrics <- collect_metrics(kmeans_tuned)

tuned_metrics |>
  arrange(desc(.metric), num_clusters) |>
  dplyr::select(num_clusters, .metric, mean, everything())
```

```
## # A tibble: 30 x 7
##    num_clusters .metric            mean .estimator     n std_err .config
##           <int> <chr>             <dbl> <chr>      <int>   <dbl> <chr>
##  1            1 sse_within_total 78438. standard       5  10466. Preprocessor1_~
##  2            2 sse_within_total 33258. standard       5   4880. Preprocessor1_~
##  3            3 sse_within_total 20731. standard       5   3025. Preprocessor1_~
##  4            4 sse_within_total 15905. standard       5   2364. Preprocessor1_~
##  5            5 sse_within_total 12756. standard       5   1876. Preprocessor1_~
##  6            6 sse_within_total 10160. standard       5   1334. Preprocessor1_~
##  7            7 sse_within_total  8655. standard       5   1131. Preprocessor1_~
##  8            8 sse_within_total  7348. standard       5    925. Preprocessor1_~
##  9            9 sse_within_total  6363. standard       5    809. Preprocessor1_~
## 10           10 sse_within_total  5610. standard       5    743. Preprocessor1_~
## # i 20 more rows
```

Choosing the Number of Clusters

```r
tuned_metrics |>
  filter(.metric == "sse_ratio") |>
  ggplot(aes(x = num_clusters, y = mean)) +
  geom_point() +
  geom_line() +
  labs(x = "Number of Clusters", y = "Mean WSS/TSS (5 folds)") +
  scale_x_continuous(breaks = seq(1, 10))
```




```
pfigures_clust <- public_figures |>
  dplyr::select(TIPI_6, TIPI_7, TIPI_8, TIPI_9, TIPI_10)

x.matrix <- model.matrix(~ TIPI_6 + TIPI_7 + TIPI_8 + TIPI_9 + TIPI_10, data = public_figures)[,-1]
```

```
kmeans_wflow <- workflow() |>
  add_model(kmeans_model) |>
  add_recipe(kmeans_recipe)
```

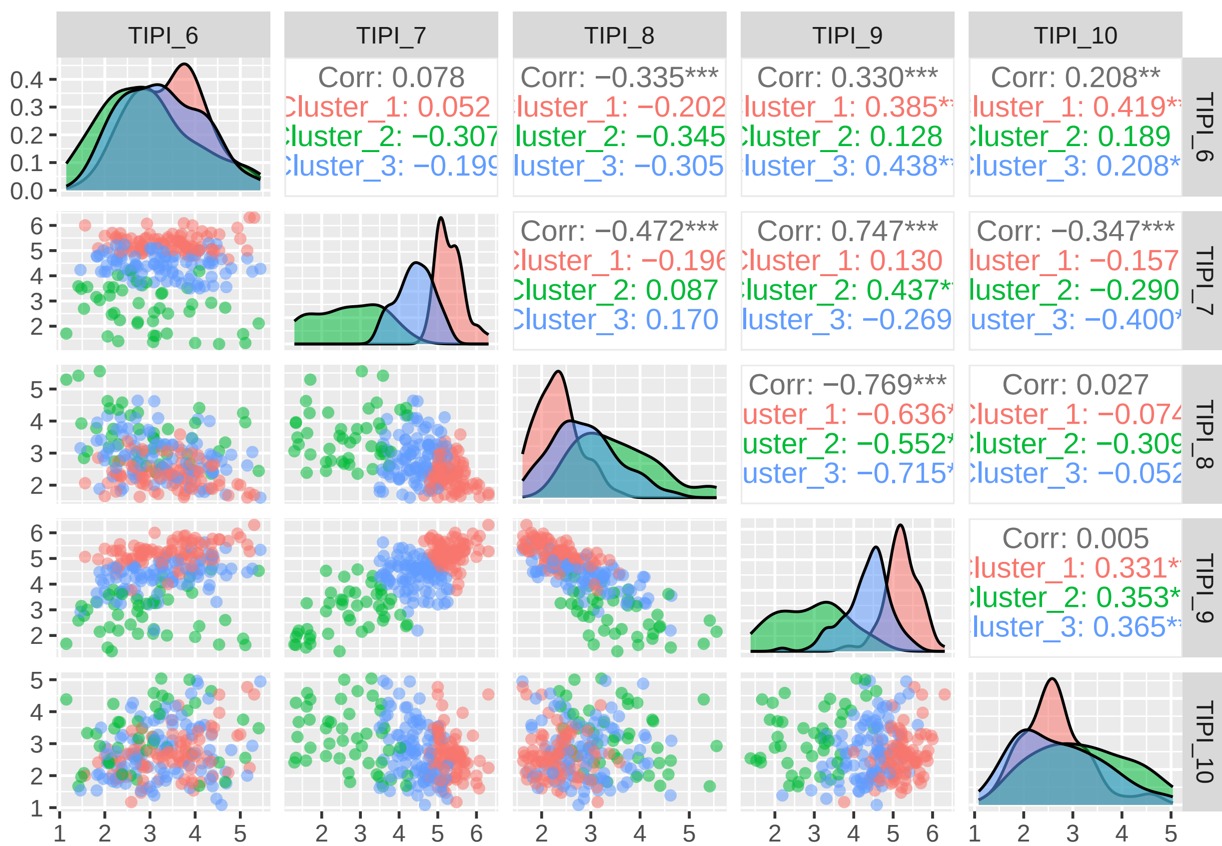
```
kmeans_4clusters <- kmeans_wflow |>
  finalize_workflow_tidyclust(parameters = list(num_clusters = 3))
```

```
set.seed(56685)
# always reset the seed before you re-fit, just in case something weird happens

kmeans_fit4 <- kmeans_4clusters |>
  fit(data = public_figures)
```

```
assignments4 <- bind_cols(
  public_figures,
  kmeans_fit4 |> extract_cluster_assignment())

assignments4 |>
  dplyr::select(name, .cluster, everything())
```

```
## # A tibble: 226 x 19
##    name      .cluster gender birthyear n_raters occupation industry TIPI_1 TIPI_2
##    <chr>     <fct>    <chr>      <dbl>    <dbl> <chr>      <chr>     <dbl>  <dbl>
##  1 Barack ~ Cluster~ Male        1961       29 POLITICIAN GOVERNM~   5.76   3.34
##  2 Kim Kar~ Cluster~ Female      1980       32 CELEBRITY  MEDIA P~   6.09   4.97
##  3 Mark Zu~ Cluster~ Male        1984       27 BUSINESSP~ BUSINESS   2.81   5.37
##  4 Cristia~ Cluster~ Male        1985       11 SOCCER PL~ TEAM SP~   6.09   4
##  5 John F.~ Cluster~ Male        1917       33 POLITICIAN GOVERNM~   5.70   3.36
##  6 Nelson ~ Cluster~ Male        1918       24 SOCIAL AC~ ACTIVISM   4.75   3.17
##  7 Michael~ Cluster~ Male        1963       38 BASKETBAL~ TEAM SP~   5.58   3.89
##  8 Lionel ~ Cluster~ Male        1987       10 SOCCER PL~ TEAM SP~   5.3    3.8
##  9 Bill Ga~ Cluster~ Male        1955       34 BUSINESSP~ BUSINESS   4.06   3.71
## 10 George ~ Cluster~ Male        1946       32 POLITICIAN GOVERNM~   4.16   4.78
## # i 216 more rows
## # i 10 more variables: TIPI_3 <dbl>, TIPI_4 <dbl>, TIPI_5 <dbl>, TIPI_6 <dbl>,
## #   TIPI_7 <dbl>, TIPI_8 <dbl>, TIPI_9 <dbl>, TIPI_10 <dbl>, likability <dbl>,
## #   pred_likability <dbl>
```

```r
library(GGally)
ggpairs(assignments4, columns = c("TIPI_6", "TIPI_7", "TIPI_8", "TIPI_9", "TIPI_10"),
        aes(color = .cluster, alpha = 0.3))
```



How many clusters best grouped the people in the training set?

I chose to use 3 clusters because the total mean WSS/TSS is at 0.25 at 3 clusters, and doesn't significantly decrease as the number of clusters increases from there.

In the green cluster, it includes people who are rated as not being `Sympathetic, warm`*, not being* `calm, emotionally stable`*, and being* `disorganized, careless`

Part d

Fit a least absolute shrinkage and selection operator (LASSO) model

#LASSO

```
lasso_model <- linear_reg(mode = "regression", engine = "glmnet",
                          penalty = tune(), # let's tune the lambda penalty term
                          mixture = 1) # mixture = 1 specifies pure LASSO

lasso_wflow <- workflow() |>
  add_model(lasso_model)
```

```
lasso_recipe <- recipe(
  likability ~   gender + birthyear + n_raters + TIPI_1 + TIPI_2 + TIPI_3 + TIPI_4 + TIPI_5 + TIPI_6 + T
  data = pf_train
) |>
  step_normalize(all_numeric_predictors()) |> # don't scale the response
  step_dummy(all_nominal_predictors())

lasso_wflow <- lasso_wflow |>
  add_recipe(lasso_recipe)
```

```
set.seed(2)
pf_cv <- vfold_cv(pf_train, v = 10)


lasso_tune1 <- tune_grid(lasso_model,
                      lasso_recipe,
                      resamples = pf_cv)
```

```
# Check results
results1 <- collect_metrics(lasso_tune1)
print(results1)
```

```
## # A tibble: 20 x 7
##     penalty .metric .estimator   mean     n std_err .config
##       <dbl> <chr>   <chr>       <dbl> <int>   <dbl> <chr>
##   1 2.49e-10 rmse    standard   11.2      10  0.523  Preprocessor1_Model01
##   2 2.49e-10 rsq     standard    0.903    10  0.0230 Preprocessor1_Model01
##   3 4.10e- 9 rmse    standard   11.2      10  0.523  Preprocessor1_Model02
##   4 4.10e- 9 rsq     standard    0.903    10  0.0230 Preprocessor1_Model02
##   5 2.62e- 8 rmse    standard   11.2      10  0.523  Preprocessor1_Model03
##   6 2.62e- 8 rsq     standard    0.903    10  0.0230 Preprocessor1_Model03
##   7 6.80e- 7 rmse    standard   11.2      10  0.523  Preprocessor1_Model04
##   8 6.80e- 7 rsq     standard    0.903    10  0.0230 Preprocessor1_Model04
##   9 2.30e- 6 rmse    standard   11.2      10  0.523  Preprocessor1_Model05
##  10 2.30e- 6 rsq     standard    0.903    10  0.0230 Preprocessor1_Model05
##  11 1.81e- 5 rmse    standard   11.2      10  0.523  Preprocessor1_Model06
```

```
## 12 1.81e- 5 rsq     standard    0.903    10   0.0230 Preprocessor1_Model06
## 13 5.34e- 4 rmse    standard   11.2      10   0.523  Preprocessor1_Model07
## 14 5.34e- 4 rsq     standard    0.903    10   0.0230 Preprocessor1_Model07
## 15 1.43e- 3 rmse    standard   11.2      10   0.523  Preprocessor1_Model08
## 16 1.43e- 3 rsq     standard    0.903    10   0.0230 Preprocessor1_Model08
## 17 1.41e- 2 rmse    standard   11.2      10   0.523  Preprocessor1_Model09
## 18 1.41e- 2 rsq     standard    0.903    10   0.0230 Preprocessor1_Model09
## 19 6.54e- 1 rmse    standard   11.2      10   0.766  Preprocessor1_Model10
## 20 6.54e- 1 rsq     standard    0.904    10   0.0224 Preprocessor1_Model10
```

```
lasso_best <- lasso_tune1 |>
  select_by_one_std_err(
    metric = "rmse",
    desc(penalty) # order penalty from largest (highest bias = simplest model) to smallest
)
lasso_best
```

```
## # A tibble: 1 x 2
##   penalty .config
##     <dbl> <chr>
## 1   0.654 Preprocessor1_Model10
```

```
lasso_wflow_final <- lasso_wflow |>
  finalize_workflow(parameters = lasso_best)
```

#Random Forests Model

```
rfR_model <- rand_forest(mode = "regression", engine = "ranger") |>
  set_args(seed = 395,
           importance = "permutation",
           mtry = tune()
  )

rfR_recipe <- recipe(
  likability ~  gender + birthyear + n_raters + TIPI_1 + TIPI_2 + TIPI_3 + TIPI_4 + TIPI_5 + TIPI_6 + TI
  data = pf_train
)


rfR_wflow <- workflow() |>
  add_model(rfR_model) |>
  add_recipe(rfR_recipe)
```

```
pf_kfold <- vfold_cv(pf_train, v = 5, repeats = 3)

n_predictors <- sum(rfR_recipe$var_info$role == "predictor")
manual_grid <- expand.grid(mtry = seq(2, n_predictors))

rfR_tune1 <- tune_grid(rfR_model,
                      rfR_recipe,
                      resamples = pf_kfold,
                      grid = manual_grid)
```

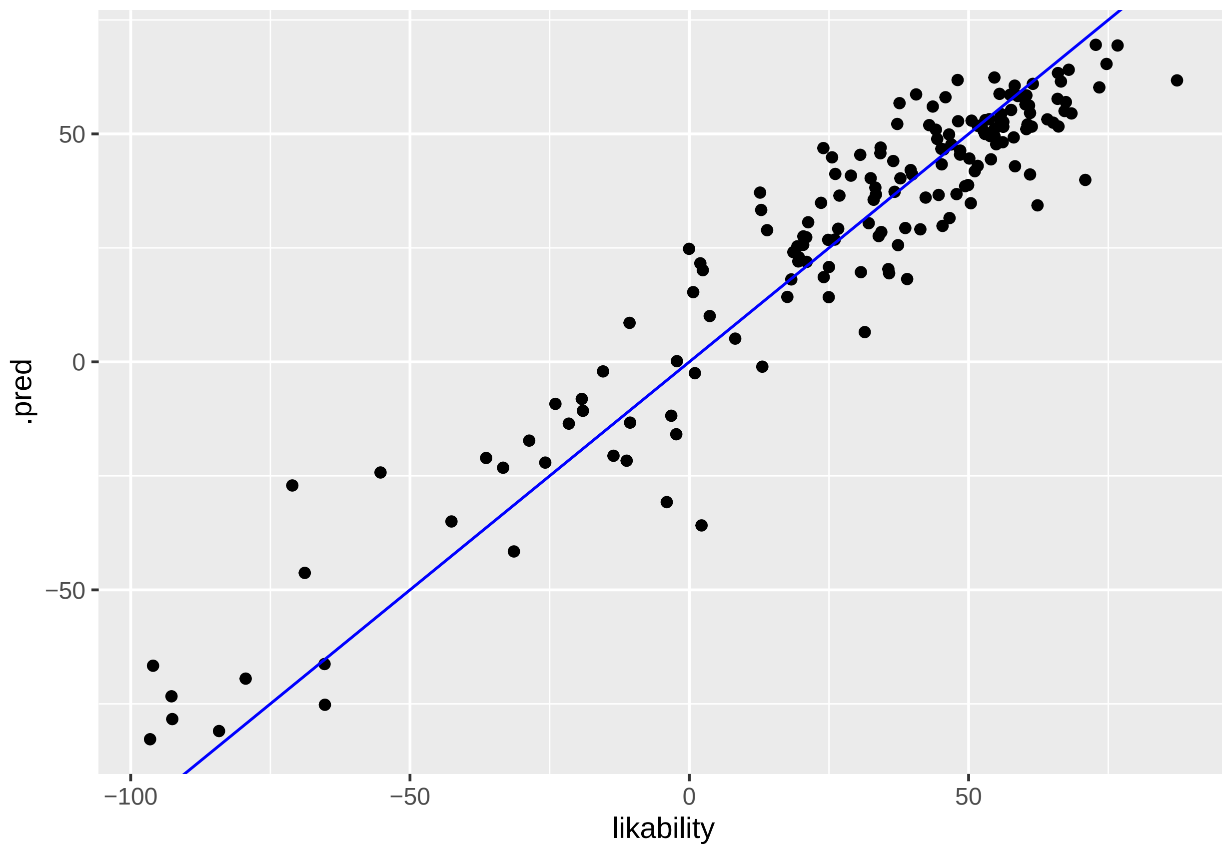
```r
rfR_best <- select_by_one_std_err(
  rfR_tune1,
  metric = "rmse",
  mtry
)
rfR_best
```

```
## # A tibble: 1 x 2
##    mtry .config
##   <int> <chr>
## 1     5 Preprocessor1_Model04
```

```r
rfR_wflow_final <- finalize_workflow(rfR_wflow, parameters = rfR_best)
rfR_fit <- fit(rfR_wflow_final, data = pf_train)


rfR_engine <- rfR_fit |>
  extract_fit_engine()
rfR_engine |> pluck("prediction.error")
```

```
## [1] 151.2057
```

```r
rfR_pred_check <- tibble(
  likability = pf_train$likability,
  .pred = rfR_engine |> pluck("predictions")
)

ggplot(rfR_pred_check, aes(x = likability, y = .pred)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "blue")
```




```
pf_predictions <- broom::augment(rfR_fit,
                                  new_data = pf_train)
mse(pf_predictions$likability, pf_predictions$.pred)
```

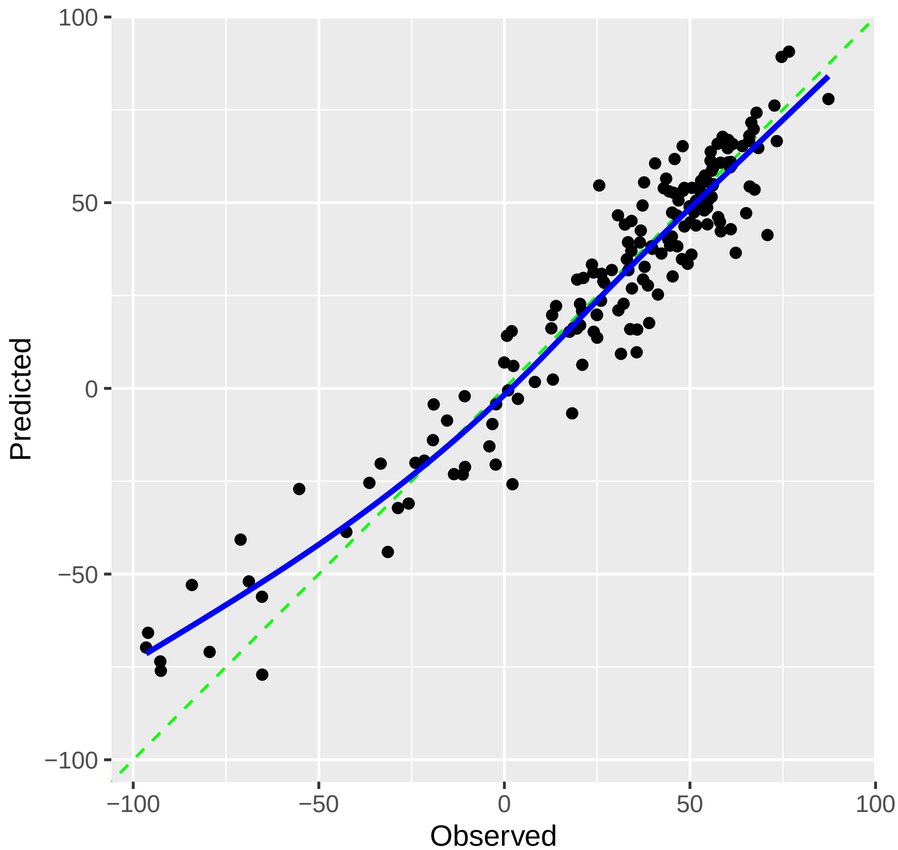
```
## [1] 26.16108
```

Part e

Fit LASSO model on the (full) training set and make predictions on the holdout set. Evaluate the quality of the holdout set predictions

```
lasso_wflow_final <- lasso_wflow |>
  finalize_workflow(parameters = lasso_best)

lasso_pred_check <- lasso_wflow_final |>
  fit_resamples(
    resamples = pf_cv,
    # save the cross-validated predictions
    control = control_resamples(save_pred = TRUE)
) |>
  collect_predictions()

# using built-in defaults from probably
cal_plot_regression(
```

```
  lasso_pred_check,
  truth = likability,
  estimate = .pred
)
```



As seen by the calibration plot, it's predictions very closely align with the actual values for most of the observations. However, for those predicted as less likable (-75 to -25), these predictions are not as accurate. They are not horribly off, but the model predicts these people to be rated as more likable than they actually are rated.

```
lasso_fit <- lasso_wflow_final |>
  fit(data = pf_train)
```

```
predictions_lasso <- lasso_fit |>
  broom::augment(new_data = pf_test)
predictions_lasso |>
  dplyr::select(
    likability,
    .pred
)
```

```
## # A tibble: 57 x 2
##    likability .pred
##         <dbl> <dbl>
##  1      -37.1 -37.2
```

```
##  2       64.1  52.8
##  3       59.9  58.9
##  4       74.3  82.3
##  5       60.7  52.5
##  6       51.1  57.8
##  7       23.2  10.4
##  8      -80.8 -65.8
##  9       17.4  17.9
## 10       33.8  46.3
## # i 47 more rows
```

```r
mse(predictions_lasso$likability, predictions_lasso$.pred)
```

```
## [1] 218.1957
```