Project 2: Data Mining, Classification, Prediction

SDS322E

Mining, Classification, Prediction

Sarah Zimmerle, syz234

Introduction

I am using the Bechdel dataset from the 'fivethirtyeight' library. My categorical variable is the Bechdel test (test). My numeric variables are the budget for the movies in dollars (budget), gross profits from the movie domestically (domgross), gross profits from the movie internationally (intgross), and net return after taking costs and budget out for international profits and domestic profits (int_aftercosts and dom_aftercosts). My binary variable is if the movie passed or failed the Bechdel test (final_result). I researched what the Bechdel test was and it turns out that it's a test to measure Hollywood's gender bias originally promoted by Alison Bechdel. A movie needs to meet all three criteria in order to pass the Bechdel test. The first criteria is that the movie needs to have at least two women. The second criteria is that the women talk to each other. The third criteria is that the women talk to each other about something other than men. I am interested in seeing if movies that I've watched display gender bias according to the Bechdel test, and if there is appropriate female representation in movies that I've watched. I am also intrigued to see if the net profit made after costs differs between movies that passed the Bechdel test and failed the Bechdel test. There are 1794 observations (9 columns) in the dataset. In the categorical variable 'test', there were 141 movies with no women (nowomen), 514 movies with at least 2 women but never talked to each other (notalk), 194 movies with at least 2 women who talked to each other but about a man (men), 142 movies that met all three criteria but were considered 'on-the-line' and got neither a pass or fail (dubious), and 803 movies that met all three criteria and passed (ok). For the binary variable 'final_result', 991 failed and 803 passed the Bechdel test.

```
library(tidyverse)
library(dplyr)
library(fivethirtyeight)
data(bechdel)

bechdel <- bechdel %>% select(-2, -4, -10, -11, -12, -13, -14,
    -15)
bechdel <- bechdel %>% rename(final_result = binary)
bechdel <- bechdel %>% rename(test = clean_test)

bechdel <- bechdel %>% mutate(net_int_aftercosts = (intgross -
    budget), net_dom_aftercosts = (domgross - budget))

glimpse(bechdel)
```

```
## Rows: 1,794
## Columns: 9
## $ year               <int> 2013, 2012, 2013, 2013, 2013, 2013, 2013, 2013, 201~
## $ title              <chr> "21 & Over", "Dredd 3D", "12 Years a Slave", "2 Gun~
## $ test               <ord> notalk, ok, notalk, notalk, men, men, notalk, ok, o~
```

```
## $ final_result       <chr> "FAIL", "PASS", "FAIL", "FAIL", "FAIL", "FAIL", "FA~
## $ budget             <int> 13000000, 45000000, 20000000, 61000000, 40000000, 2~
## $ domgross           <dbl> 25682380, 13414714, 53107035, 75612460, 95020213, 3~
## $ intgross           <dbl> 42195766, 40868994, 158607035, 132493015, 95020213,~
## $ net_int_aftercosts <dbl> 29195766, -4131006, 138607035, 71493015, 55020213, ~
## $ net_dom_aftercosts <dbl> 12682380, -31585286, 33107035, 14612460, 55020213, ~
```

```
bechdel %>% summarise_all(n_distinct)
```

```
## # A tibble: 1 x 9
##    year title  test final_result budget domgross intgross net_int_aftercosts
##   <int> <int> <int>        <int>  <int>    <int>    <int>              <int>
## 1    44  1768     5            2    272     1751     1757               1755
## # ... with 1 more variable: net_dom_aftercosts <int>
```

```
bechdel %>% count(test)
```

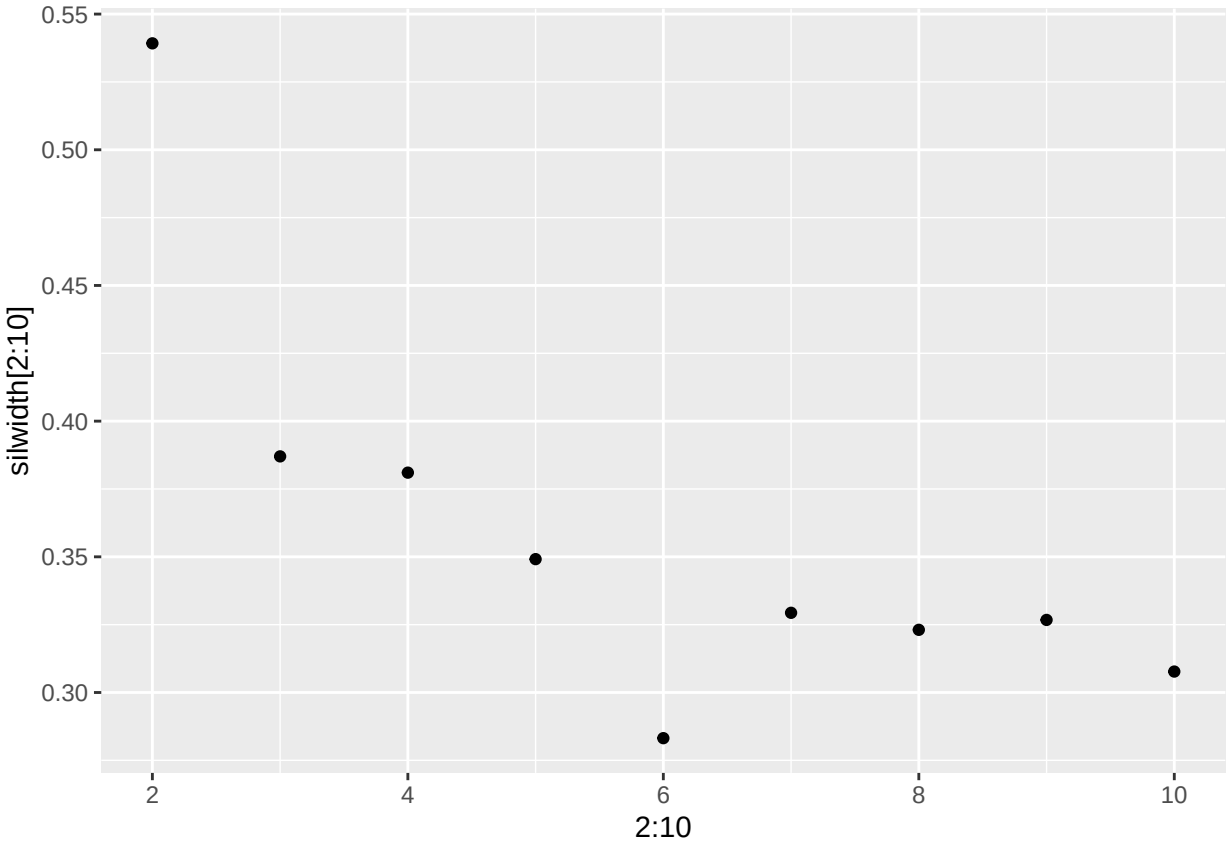
```
## # A tibble: 5 x 2
##   test        n
##   <ord>   <int>
## 1 nowomen   141
## 2 notalk    514
## 3 men       194
## 4 dubious   142
## 5 ok        803
```

```
bechdel %>% count(final_result)
```

```
## # A tibble: 2 x 2
##   final_result     n
##   <chr>        <int>
## 1 FAIL           991
## 2 PASS           803
```

Cluster Analysis

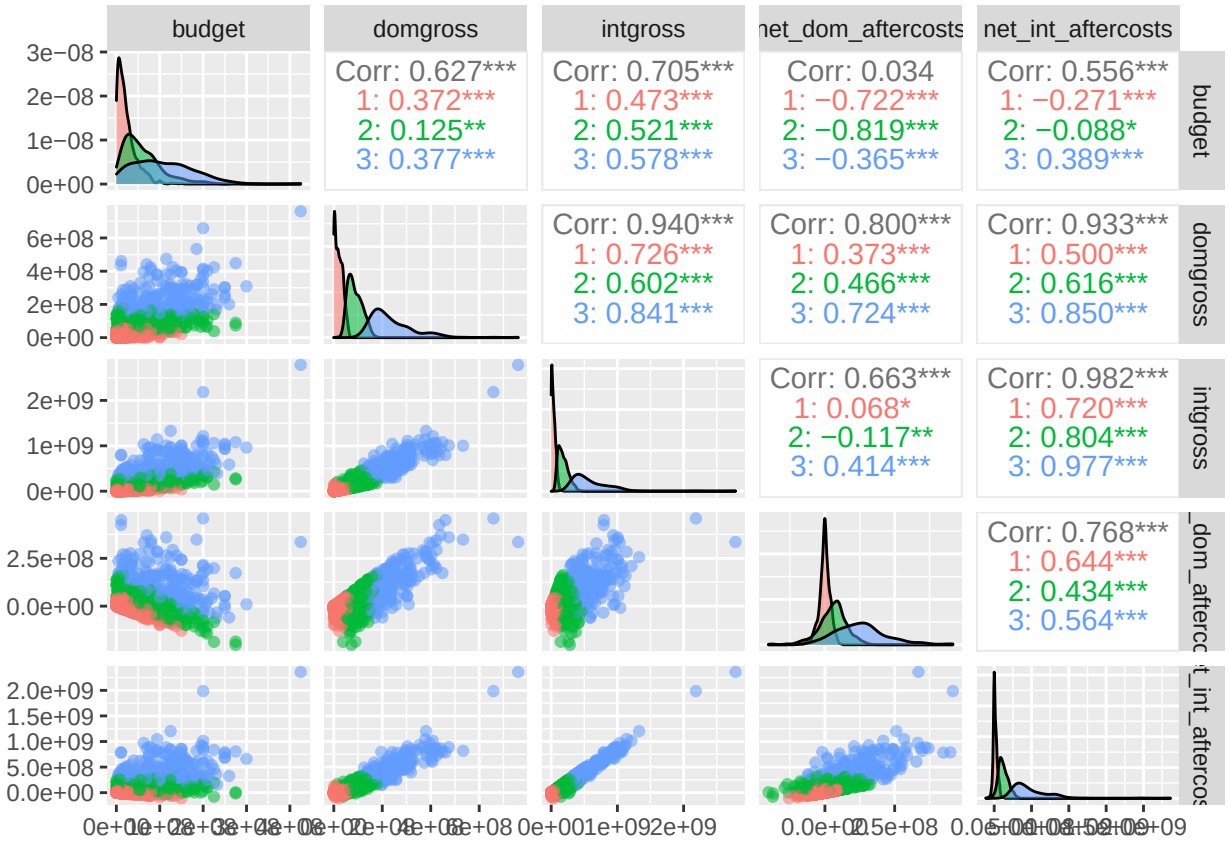
```
library(cluster)
# clustering code here
bechdel1 <- na.omit(bechdel)

silwidth <- vector()
for (i in 2:10) {
    money <- bechdel1 %>% select("budget", "domgross", "intgross",
        "net_dom_aftercosts", "net_int_aftercosts") %>% scale %>%
        pam(k = i)
    silwidth[i] <- money$silinfo$avg.width
}
ggplot() + geom_point(aes(x = 2:10, y = silwidth[2:10]))
```




```
pam1 <- bechdel1 %>% select("budget", "domgross", "intgross",
    "net_dom_aftercosts", "net_int_aftercosts") %>% scale %>%
    pam(k = 3)
pam1
```

```
## Medoids:
##        ID     budget   domgross    intgross net_dom_aftercosts
## [1,]  531 -0.5520998 -0.6306146 -0.56128298         -0.3834584
## [2,] 1297  0.1002528  0.1542178  0.05118315          0.1205716
## [3,]  512  0.9286370  1.7407156  1.65095846          1.5173761
##      net_int_aftercosts
## [1,]        -0.50893150
## [2,]         0.03300715
## [3,]         1.68369093
## Clustering vector:
##   [1] 1 1 2 2 2 2 2 1 1 2 2 1 1 1 2 1 2 3 1 2 2 2 1 1 2 3 3 2 2 1 3 2 2 1 2 2 3
##  [38] 2 1 2 2 1 2 3 1 2 2 2 3 2 1 1 1 2 2 1 2 1 1 1 2 2 1 3 1 1 1 1 3 1 3 1 1 2
##  [75] 2 2 3 1 3 1 1 1 2 1 2 2 1 1 2 3 1 2 3 1 2 2 2 2 3 1
##  [ reached getOption("max.print") -- omitted 1677 entries ]
## Objective function:
##    build     swap
## 1.025792 0.979272
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
plot(pam1, which = 3)

library(GGally)
bechdel1 %>% mutate(cluster = as.factor(pam1$clustering)) %>%
    ggpairs(columns = c("budget", "domgross", "intgross", "net_dom_aftercosts",
        "net_int_aftercosts"), aes(color = cluster, alpha = 0.5))
```




```
pamclust1 <- bechdel1 %>% mutate(cluster = as.factor(pam1$clustering))
pamclust1 %>% ggplot(aes(budget, net_dom_aftercosts, color = cluster)) +
    geom_point()
```


```
pamclust1 %>% group_by(cluster) %>% summarize_if(is.numeric,
    mean, na.rm = T)
```

```
## # A tibble: 3 x 7
##   cluster  year     budget   domgross intgross net_int_afterco~ net_dom_afterco~
##   <fct>   <dbl>      <dbl>      <dbl>    <dbl>            <dbl>            <dbl>
## 1 1       2002.  22753224.  20833173.   3.63e7        13511396.        -1920051.
## 2 2       2003.  59825013.  87498646.   1.81e8       121280078.        27673633.
## 3 3       2003. 108019103. 238838231.   5.88e8       479832477.       130819128.
```

```
bechdel1 %>% slice(pam1$id.med)
```

```
## # A tibble: 3 x 9
##    year title      test   final_result budget domgross intgross net_int_afterco~
##   <int> <chr>      <ord>  <chr>         <int>    <dbl>    <dbl>            <dbl>
## 1  2009 The Inven~ men    FAIL         1.85e7   1.85e7   3.27e7         14179264
## 2  1999 Payback    notalk FAIL         5   e7   8.15e7   1.62e8        111626121
## 3  2009 Sherlock ~ notalk FAIL         9   e7   2.09e8   4.98e8        408438212
## # ... with 1 more variable: net_dom_aftercosts <dbl>
```

Discussion of clustering here

Dimensionality Reduction with PCA

```
# PCA code here
```

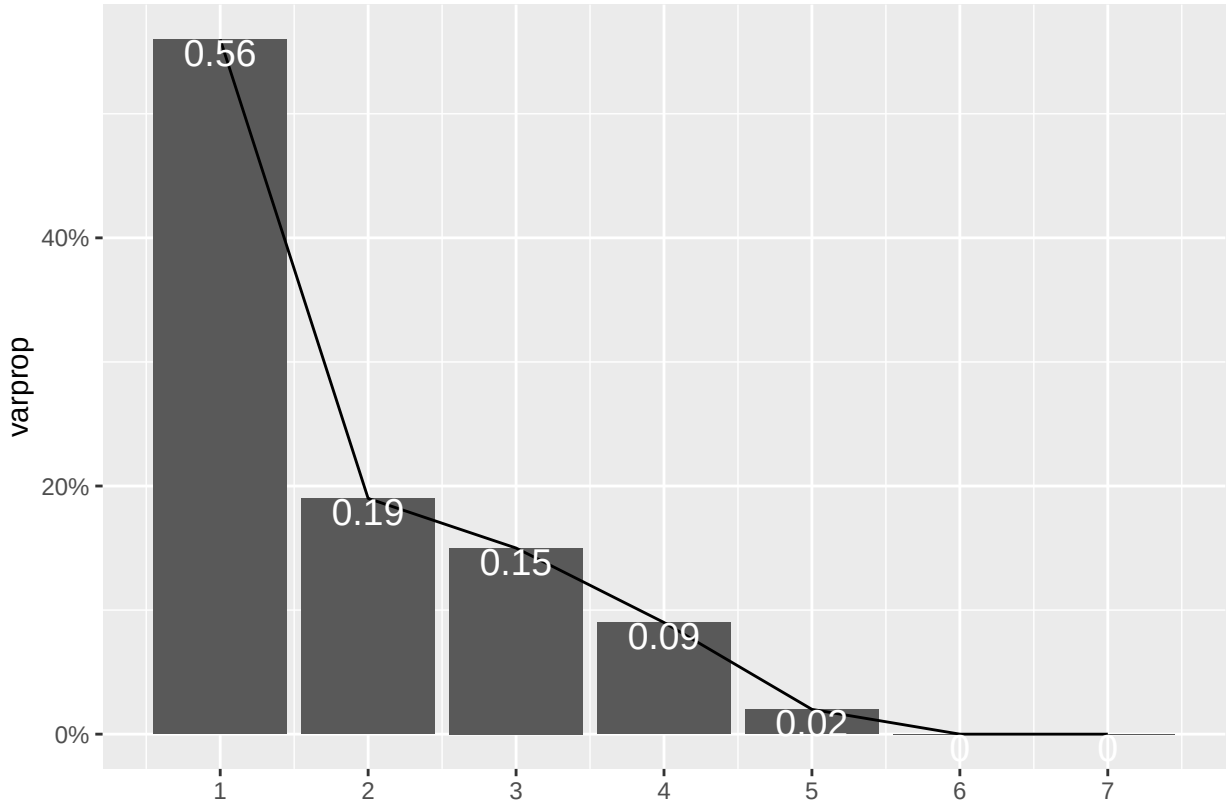
```r
library(tidyverse)

trying1 <- bechdel1 %>% mutate(final_result = ifelse(final_result ==
    "FAIL", 0, 1))

bechdel_nums <- trying1 %>% select_if(is.numeric) %>% scale

bechdel_pca <- princomp(bechdel_nums)
names(bechdel_pca)
```

```
## [1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"   "call"
```

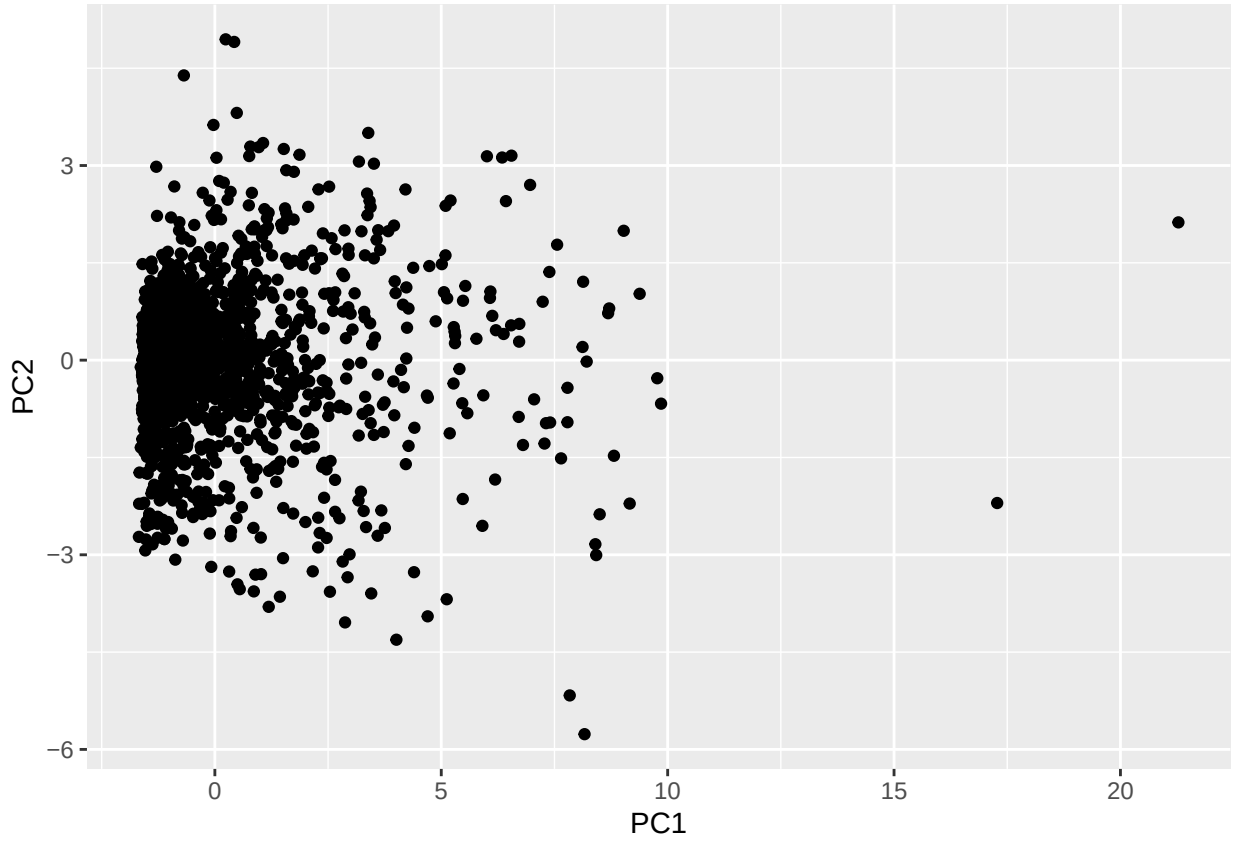
```r
summary(bechdel_pca, loadings = T)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     1.9760866 1.1439469 1.0196778 0.79268512 0.33828334
## Proportion of Variance 0.5581595 0.1870502 0.1486183 0.08981478 0.01635715
## Cumulative Proportion  0.5581595 0.7452097 0.8938281 0.98364285 1.00000000
##                           Comp.6 Comp.7
## Standard deviation             0      0
## Proportion of Variance         0      0
## Cumulative Proportion          1      1
##
## Loadings:
##                    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## year                        0.713  0.242  0.656
## final_result                       0.939 -0.339
## budget              0.332  0.515 -0.177 -0.531  0.328        -0.445
## domgross            0.498                       0.488  0.167  0.695
## intgross            0.498                      -0.420 -0.741  0.121
## net_int_aftercosts  0.494                      -0.580  0.633 -0.104
## net_dom_aftercosts  0.383 -0.465  0.155  0.404  0.373 -0.130 -0.542
```

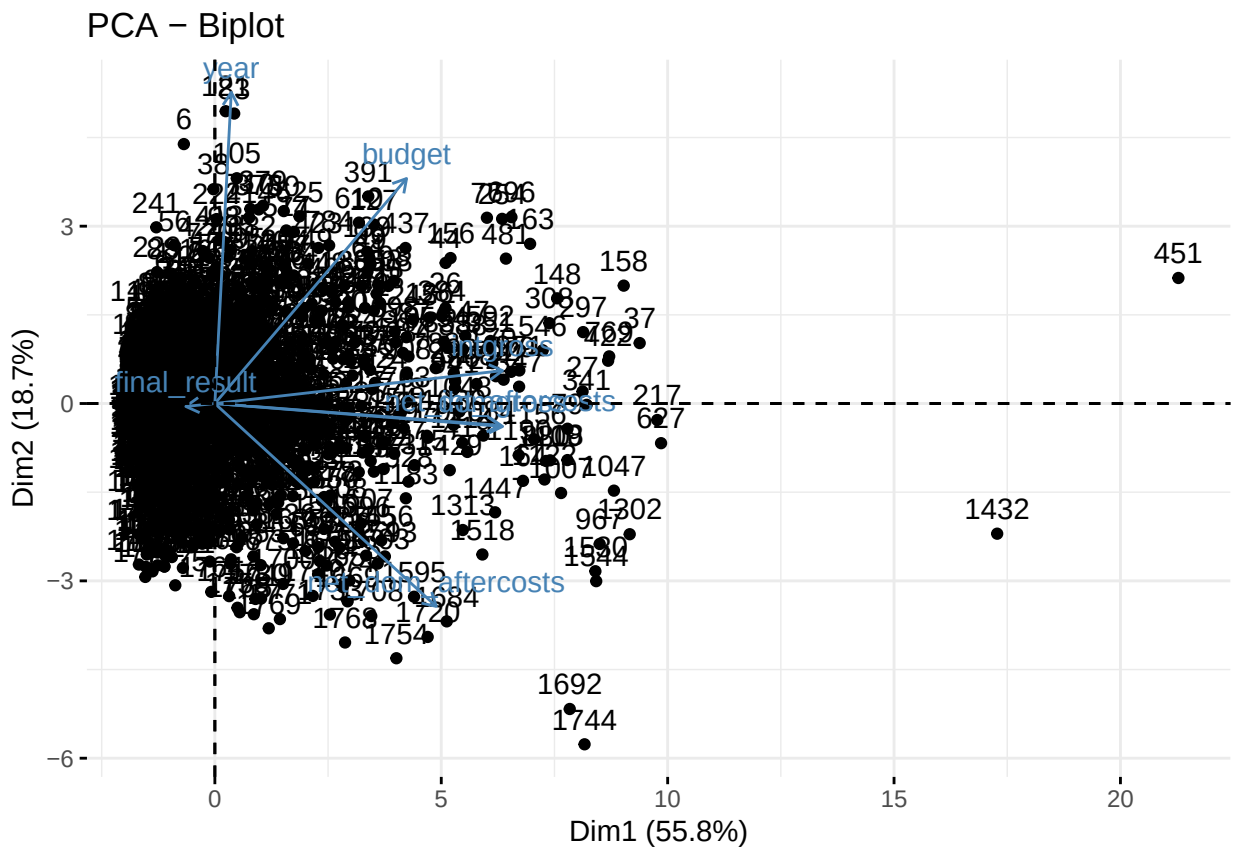
```r
# determining how many PCs to keep
eigval <- bechdel_pca$sdev^2
varprop = round(eigval/sum(eigval), 2)

ggplot() + geom_bar(aes(y = varprop, x = 1:7), stat = "identity") +
    xlab("") + geom_path(aes(y = varprop, x = 1:7)) + geom_text(aes(x = 1:7,
    y = varprop, label = round(varprop, 2)), vjust = 1, col = "white",
    size = 5) + scale_y_continuous(breaks = seq(0, 0.6, 0.2),
    labels = scales::percent) + scale_x_continuous(breaks = 1:10)
```




```
# Keeping PC1 and PC2 & plotting them

bech_df <- data.frame(PC1 = bechdel_pca$scores[, 1], PC2 = bechdel_pca$scores[,
    2])

ggplot(bech_df, aes(PC1, PC2)) + geom_point()
```




```
# plotting biplot
library(factoextra)
fviz_pca_biplot(bechdel_pca)
```



Discussions of PCA here.

Linear Classifier

```
# linear classifier code here

trying2 <- bechdel1 %>% mutate(final_result = ifelse(final_result ==
    "FAIL", 0, 1))

newdata <- trying2[, c("final_result", "net_int_aftercosts",
    "net_dom_aftercosts", "domgross", "intgross", "budget")]


fit <- glm(final_result ~ ., family = binomial, data = newdata)

summary(fit)
```

```
##
## Call:
## glm(formula = final_result ~ ., family = binomial, data = newdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2866  -1.1142  -0.9091   1.2092   1.9154
##
```

```
## Coefficients: (2 not defined because of singularities)
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          9.876e-02  7.371e-02   1.340   0.1803
## net_int_aftercosts   1.721e-09  7.704e-10   2.234   0.0255 *
## net_dom_aftercosts   5.465e-09  1.367e-09   3.999 6.36e-05 ***
## domgross            -9.155e-09  1.889e-09  -4.847 1.26e-06 ***
## intgross                    NA         NA      NA       NA
## budget                      NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2443.3  on 1776  degrees of freedom
## Residual deviance: 2407.7  on 1773  degrees of freedom
## AIC: 2415.7
##
## Number of Fisher Scoring iterations: 4
```

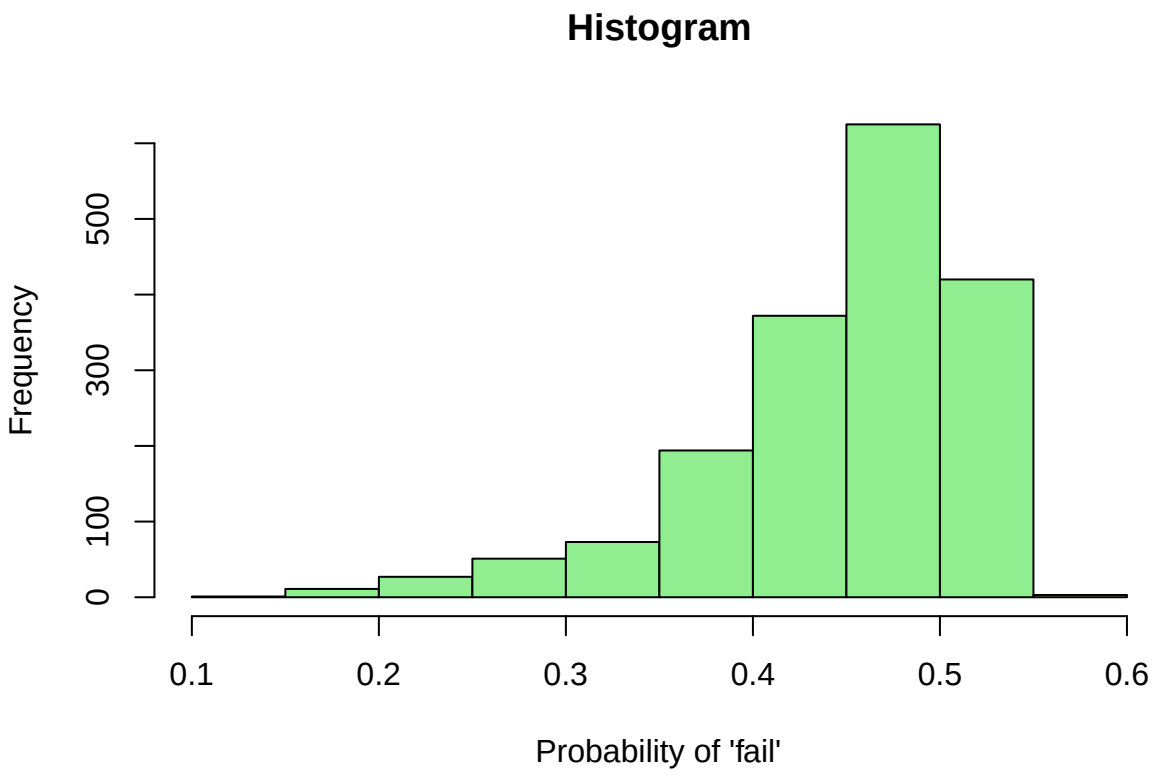
```
score <- predict(fit, type = "response")
score %>% round(3)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 0.496 0.449 0.508 0.404 0.407 0.196 0.429 0.527 0.492 0.346 0.424 0.467 0.441
##    14    15    16    17    18    19    20    21    22    23    24    25    26
## 0.491 0.422 0.475 0.393 0.466 0.504 0.351 0.318 0.365 0.429 0.421 0.486 0.360
##    27    28    29    30    31    32    33    34    35    36    37    38    39
## 0.332 0.328 0.420 0.469 0.383 0.367 0.481 0.475 0.414 0.508 0.318 0.231 0.474
##    40    41    42    43    44    45    46    47    48    49    50    51    52
## 0.464 0.429 0.518 0.495 0.197 0.495 0.433 0.355 0.375 0.204 0.282 0.467 0.466
##    53    54    55    56    57    58    59    60    61    62    63    64    65
## 0.493 0.388 0.435 0.305 0.386 0.458 0.480 0.450 0.449 0.429 0.467 0.213 0.474
##    66    67    68    69    70    71    72    73    74    75    76    77    78
## 0.462 0.498 0.479 0.499 0.494 0.360 0.459 0.478 0.232 0.394 0.401 0.262 0.436
##    79    80    81    82    83    84    85    86    87    88    89    90    91
## 0.284 0.457 0.420 0.511 0.147 0.421 0.499 0.412 0.520 0.507 0.397 0.378 0.485
##    92    93    94    95    96    97    98    99   100
## 0.428 0.312 0.504 0.334 0.460 0.436 0.290 0.253 0.419
##  [ reached getOption("max.print") -- omitted 1677 entries ]
```

```
summary(fit$fitted.values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1469  0.4142  0.4652  0.4468  0.4987  0.5629
```

```
hist(fit$fitted.values, main = " Histogram ", xlab = "Probability of 'fail'",
    col = "light green")
```




```
newdata$Predict <- ifelse(fit$fitted.values > 0.5, "fail", "pass")

fit$aic
```

```
## [1] 2415.688
```

```
mytable <- table(newdata$final_result, newdata$Predict)
rownames(mytable) <- c("Obs. pass", "Obs. fail")
colnames(mytable) <- c("Pred. pass", "Pred. fail")
mytable
```

```
##
##             Pred. pass Pred. fail
##   Obs. pass        208        775
##   Obs. fail        215        579
```

```
efficiency <- sum(diag(mytable))/sum(mytable)
efficiency
```

```
## [1] 0.4428813
```

```
# cross-validation of linear classifier here

set.seed(322)
k = 10
```

Discussion here

Non-Parametric Classifier

```
library(caret)
# non-parametric classifier code here
```

```
# cross-validation of np classifier here
```

Discussion

Regression/Numeric Prediction

```
# regression model code here
```

```
# cross-validation of regression model here
```

Discussion

Python

```
library(reticulate)
```

```
# python code here
```

Discussion

Concluding Remarks

Include concluding remarks here, if any