hall_ass_final

Chloe Hall

2022-11-08

Contents

Sentiment Analysis

1 Introduction

I will be performing a Principle Component Analysis on the Goodreads Book dataset to try and make a more accurate logistic regression that is able to predict if a book will have a good or bad rating on Goodreads from a sentiment analysis of its book description. I chose rating as my outcome variable and the sentiments from the NRC sentiment dictionary as my predictors, which incluces anger, anticipation, disgust, fear, joy, negative, sadness, surprise, and trust. In my Assignment 6, I conducted a similar project using the NRC sentiment dictionary as my predictor variables for assignement 6, so I have changed the parameters of the good and bad ratings and have added in a PCA to see if these will increase my model's accuracy and improve its prediction ability on the dataset.

1.1 Research Question

Can the sentiment counts for words in Goodreads book description related to anger, anticipation, disgust, fear, joy, sadness, surprise, and trust derived from the NRC sentiment dictionary be reduced using a Principle Component Analysis into predictor variables that predict if book reviews will be good or bad?

1.2 Data Discussion

The data used in this analysis is the Goodreads Book Datasets With User Rating 2M data set. The data set contains information about approximately 10,000,000 books from the site Goodreads archive including Book Title, Rating Distribution for 1-5 stars, Pages Number, Total Rating Distribution, Publish Month, Publish Day, Publisher, Count of Reviews, Publish Year, Language, Authors, Rating, ISBN, Count of Text Reviews, and a Book Description. There are several files within the dataset so for the sake of scale, I will be using the csv file book2000k-3000k and operating my analysis on only the first 100k books in that dataset.

I will be using the words within the Book Description text in order to run a NLP sentiment analysis on what words within a books description are associated with positively reviewed books and what are associated with negatively reviewed books.

Then I will be using the AFINN sentiment dictionary from tidytext to to quantify the sentiments related to anger, anticipation, disgust, fear, joy, sadness, surprise, and trust to see which elements in a book description are correlated with book rating.

The data used in this assignment is available at : https://www.kaggle.com/datasets/bahramjannesarr/goodreads-book-datasets-10m

2 Data Wrangling

In order to organize the data successfully the following steps were completed. 1. Called in the data frame 2. Filtered to the English Language so the results would be interpretable for me. 3. Filtered to only books with ratings so the results would not be skewed by zeros . 4. Removed all NA values 5. Unnesting the book descriptions to be able to analyze each word 6. Loading in the NRC sentiment dataset and then merging the datasets by word 7. Merging the book words with the NRC dictionary 8. Getting sentiment counts and normalizing based on number of words in the description. 9. Creating equal class sizes and a train and test dataset to run my logisitic regression on.

2.1 Read and wrangle data.

```
#Loading the necessary libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(stringr)
library(tidytext)
library(textdata)
library(dplyr)

#importing the data set & Filtering to relevant variables and filtering out NAs
goodreads_corpus <- read_csv("archive/book2000k-3000k.csv") %>%
  filter(Language == "eng") %>%
  filter(Rating!=0)%>%
  na_if("") %>% #convert empty cells to NA
  na.omit()
```

```
## Rows: 395957 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (12): Name, Authors, ISBN, Publisher, RatingDist5, RatingDist4, RatingDi...
## dbl  (7): Id, Rating, PublishYear, PublishMonth, PublishDay, CountsOfReview,...
##
## 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.
```

```
#Creating an index column
goodreads_corpus$id <- 1:nrow(goodreads_corpus)
```

2.1.1 Unnesting the words

```
#Sorting the books into good and bad reviews
goodreads_corpus<- goodreads_corpus %>%
  mutate(bookrating = case_when(Rating > 3 ~ "goodrat",
                                Rating <= 3 ~ "badrat"))

#get the word frequency for each text type (pos and neg sentiment)
goodreads_corpus_words <- goodreads_corpus %>%
  unnest_tokens(word, Description)

goodreads_corpus_words
```

```
## # A tibble: 2,726,892 x 21
##         Id Name     Authors ISBN  Rating Publi~1 Publi~2 Publi~3 Publi~4 Ratin~5
##      <dbl> <chr>    <chr>   <chr>  <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  2 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  3 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  4 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  5 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  6 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  7 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  8 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
##  9 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
## 10 2000045 Strateg~ Steve ~ 0750~      4    2005       1       1 Butter~ 5:2
## # ... with 2,726,882 more rows, 11 more variables: RatingDist4 <chr>,
## #   RatingDist3 <chr>, RatingDist2 <chr>, RatingDist1 <chr>,
## #   RatingDistTotal <chr>, CountsOfReview <dbl>, Language <chr>,
## #   PagesNumber <dbl>, id <int>, bookrating <chr>, word <chr>, and abbreviated
## #   variable names 1: PublishYear, 2: PublishMonth, 3: PublishDay,
## #   4: Publisher, 5: RatingDist5
```

Here I changed my parameter to above or below 3 in the hopes of creating a larger distinction between the datasets instead of using the mean.

2.1.2 Loading in the NRC sentiment

```
get_sentiments("nrc")
```

```
## # A tibble: 13,875 x 2
##    word        sentiment
##    <chr>       <chr>
```

```
##  1 abacus      trust
##  2 abandon     fear
##  3 abandon     negative
##  4 abandon     sadness
##  5 abandoned   anger
##  6 abandoned   fear
##  7 abandoned   negative
##  8 abandoned   sadness
##  9 abandonment anger
## 10 abandonment fear
## # ... with 13,865 more rows
```

```
nrc_data <- get_sentiments("nrc")

nrc_data
```

```
## # A tibble: 13,875 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 abacus      trust
##  2 abandon     fear
##  3 abandon     negative
##  4 abandon     sadness
##  5 abandoned   anger
##  6 abandoned   fear
##  7 abandoned   negative
##  8 abandoned   sadness
##  9 abandonment anger
## 10 abandonment fear
## # ... with 13,865 more rows
```

2.1.3 Inner Joining the Data

```
#match the words with NRC data
nrc_words <- goodreads_corpus_words %>%
  inner_join(nrc_data)
```

```
## Joining, by = "word"
```

```
nrc_words
```

```
## # A tibble: 769,389 x 22
##         Id Name     Authors ISBN  Rating Publi~1 Publi~2 Publi~3 Publi~4 Ratin~5
##      <dbl> <chr>    <chr>   <chr>  <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  2 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  3 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  4 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  5 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  6 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  7 2000294 Bastard~ Kazush~ 1421~   3.86    2008       1       8 VIZ Me~ 5:13
##  8 2000294 Bastard~ Kazush~ 1421~   3.86    2008       1       8 VIZ Me~ 5:13
##  9 2000294 Bastard~ Kazush~ 1421~   3.86    2008       1       8 VIZ Me~ 5:13
## 10 2000294 Bastard~ Kazush~ 1421~   3.86    2008       1       8 VIZ Me~ 5:13
## # ... with 769,379 more rows, 12 more variables: RatingDist4 <chr>,
## #   RatingDist3 <chr>, RatingDist2 <chr>, RatingDist1 <chr>,
```

```
## #   RatingDistTotal <chr>, CountsOfReview <dbl>, Language <chr>,
## #   PagesNumber <dbl>, id <int>, bookrating <chr>, word <chr>, sentiment <chr>,
## #   and abbreviated variable names 1: PublishYear, 2: PublishMonth,
## #   3: PublishDay, 4: Publisher, 5: RatingDist5
```

```
#create a count by text for each group of words
count_data <- nrc_words %>%
  group_by(id, sentiment) %>%
  count

count_data
```

```
## # A tibble: 171,135 x 3
## # Groups:   id, sentiment [171,135]
##       id sentiment        n
##    <int> <chr>        <int>
##  1     1 positive         6
##  2     2 anger            6
##  3     2 anticipation     6
##  4     2 disgust          4
##  5     2 fear             6
##  6     2 joy              1
##  7     2 negative         9
##  8     2 positive         8
##  9     2 sadness          3
## 10     2 surprise         1
## # ... with 171,125 more rows
```

```
#widen out the tibble
count_data_wide <- count_data %>%
  pivot_wider(names_from = sentiment, values_from = n)

count_data_wide
```

```
## # A tibble: 22,069 x 11
## # Groups:   id [22,069]
##       id posit~1 anger antic~2 disgust  fear   joy negat~3 sadness surpr~4 trust
##    <int>   <int> <int>   <int>   <int> <int> <int>   <int>   <int>   <int> <int>
##  1     1       6    NA      NA      NA    NA    NA      NA      NA      NA    NA
##  2     2       8     6       6       4     6     1       9       3       1     8
##  3     3      14    NA       9       2     2     3       7       6       4     8
##  4     4       1    NA       1       1     2    NA       1      NA       1     1
##  5     5       5     1       1      NA    NA     3       1      NA      NA    NA
##  6     6       1    NA       1       1     2    NA       1      NA       1     1
##  7     7       1     4       1       8     9     1       8       4       1     4
##  8     8      14     1       7       1     4     5       5       3       3     3
##  9     9       2    NA       2      NA    NA     1       1      NA      NA     4
## 10    10      10    11       6       9    13     2      16       3       3     6
## # ... with 22,059 more rows, and abbreviated variable names 1: positive,
## #   2: anticipation, 3: negative, 4: surprise
```

```
#get word count for texts
nw_texts <- goodreads_corpus %>%
  mutate(nw = str_count(Description, "\\W+")) #this will create new variable called nw using stringr fu

nw_texts
```

```
## # A tibble: 22,477 x 22
##         Id Name     Authors ISBN  Rating Publi~1 Publi~2 Publi~3 Publi~4 Ratin~5
##      <dbl> <chr>    <chr>   <chr>  <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  2 2000294 Bastard~ Kazush~ 1421~   3.86    2008       1       8 VIZ Me~ 5:13
##  3 2000356 Visible~ Jack N~ 1570~   3.85    2007       4      10 Sasqua~ 5:13
##  4 2000413 Mouryou~ Tamayo~ 1595~   2.83    2005       6       7 Tokyop~ 5:3
##  5 2000444 Mouryou~ Tamayo~ 1595~   2.95    2004      12       7 Tokyop~ 5:9
##  6 2000446 Mouryou~ Tamayo~ 1595~   3       2005       3       8 Tokyop~ 5:4
##  7 2000458 Parasyt~ Hitosh~ 0345~   4.29    2007      10      30 Del Rey 5:1024
##  8 2000471 Jewels   Victor~ 0340~   4.11    2007       2      23 Not Av~ 5:476
##  9 2000483 Hebrews~ Stuart~ 1880~   3.83    1995       6       1 Messia~ 5:8
## 10 2000512 The Edg~ P.T. D~ 0312~   4       1994       4       1 St. Ma~ 5:84
## # ... with 22,467 more rows, 12 more variables: RatingDist4 <chr>,
## #   RatingDist3 <chr>, RatingDist2 <chr>, RatingDist1 <chr>,
## #   RatingDistTotal <chr>, CountsOfReview <dbl>, Language <chr>,
## #   PagesNumber <dbl>, Description <chr>, id <int>, bookrating <chr>, nw <int>,
## #   and abbreviated variable names 1: PublishYear, 2: PublishMonth,
## #   3: PublishDay, 4: Publisher, 5: RatingDist5
```

2.2 Combining the sentiment and word databases

```
#join tibbles (nw and sentiment) together and get normed counts for sentiment/nw by text
final_nrc_review_tib <- nw_texts %>%
  inner_join(count_data_wide, by = "id") %>% #join tibbles together
  mutate_at(vars(anger:trust), list(normed = ~./nw)) %>%  #this creates new variables that are normed s
  mutate(bookrating = case_when(Rating > 3 ~ "goodrat",
                                Rating <= 3 ~ "badrat")) #Create a new variable called sentiment. Assi
```

2.2.1 Manipulating the dataframe

```
#now we have a tibble!
final_nrc_review_tib
```

```
## # A tibble: 22,069 x 41
##         Id Name     Authors ISBN  Rating Publi~1 Publi~2 Publi~3 Publi~4 Ratin~5
##      <dbl> <chr>    <chr>   <chr>  <dbl>   <dbl>   <dbl>   <dbl> <chr>   <chr>
##  1 2000045 Strateg~ Steve ~ 0750~   4       2005       1       1 Butter~ 5:2
##  2 2000294 Bastard~ Kazush~ 1421~   3.86    2008       1       8 VIZ Me~ 5:13
##  3 2000356 Visible~ Jack N~ 1570~   3.85    2007       4      10 Sasqua~ 5:13
##  4 2000413 Mouryou~ Tamayo~ 1595~   2.83    2005       6       7 Tokyop~ 5:3
##  5 2000444 Mouryou~ Tamayo~ 1595~   2.95    2004      12       7 Tokyop~ 5:9
##  6 2000446 Mouryou~ Tamayo~ 1595~   3       2005       3       8 Tokyop~ 5:4
##  7 2000458 Parasyt~ Hitosh~ 0345~   4.29    2007      10      30 Del Rey 5:1024
##  8 2000471 Jewels   Victor~ 0340~   4.11    2007       2      23 Not Av~ 5:476
##  9 2000483 Hebrews~ Stuart~ 1880~   3.83    1995       6       1 Messia~ 5:8
## 10 2000512 The Edg~ P.T. D~ 0312~   4       1994       4       1 St. Ma~ 5:84
## # ... with 22,059 more rows, 31 more variables: RatingDist4 <chr>,
## #   RatingDist3 <chr>, RatingDist2 <chr>, RatingDist1 <chr>,
## #   RatingDistTotal <chr>, CountsOfReview <dbl>, Language <chr>,
## #   PagesNumber <dbl>, Description <chr>, id <int>, bookrating <chr>, nw <int>,
## #   positive <int>, anger <int>, anticipation <int>, disgust <int>, fear <int>,
## #   joy <int>, negative <int>, sadness <int>, surprise <int>, trust <int>,
## #   anger_normed <dbl>, anticipation_normed <dbl>, disgust_normed <dbl>, ...
```

```
#Making a dataframe
final_nrc_review_df <- as.data.frame(final_nrc_review_tib)

#Making all NA values into zeros
final_nrc_review_df[is.na(final_nrc_review_df)] = 0

#Checking the df
#final_nrc_review_df

#Making a new sentiment column at the end of the dataset
final_nrc_review_df$sentiment <- final_nrc_review_df$bookrating
```

2.2.2 Removing NA values

```
final_nrc_review_df<-final_nrc_review_df %>%
  filter(sentiment=="goodrat"|sentiment=="badrat")
```

2.2.3 Stratifying my dataframe

```
#There are 1191 bad ratings and 20878 good ones so let's make them equal!
final_nrc_review_df %>%
  count(bookrating)
```

```
##   bookrating     n
## 1     badrat  1191
## 2    goodrat 20878
```

```
#Creating seperate datasets to stratify our sample from
goodrat<-final_nrc_review_df %>%
  filter(bookrating=="goodrat")

badrat<-final_nrc_review_df %>%
  filter(bookrating=="badrat")

#Creating equal samples of each rating
goodrat_sample<-goodrat[sample(nrow(goodrat), 1000), ]
badrat_sample<-badrat[sample(nrow(badrat), 1000), ]

#Combining the two samples
goodreads_nrc_sample<-rbind(goodrat_sample, badrat_sample)

#Double checking the dist
goodreads_nrc_sample %>%
  count(bookrating) #Perfect!
```

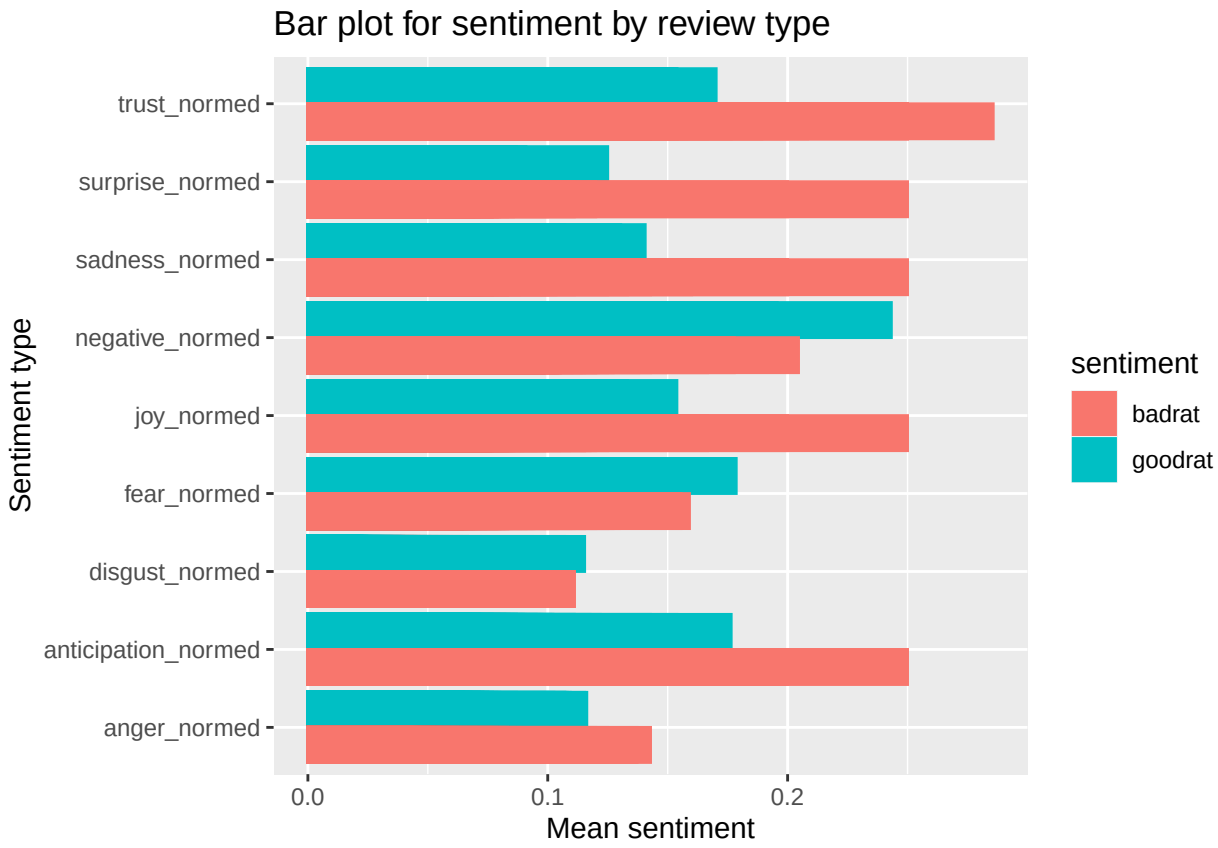
```
##   bookrating    n
## 1     badrat 1000
## 2    goodrat 1000
```

2.3 Visualizing the data

```
bar_plot_nrc <- goodreads_nrc_sample%>%
  select(anger_normed: sentiment) %>%
```

```
  pivot_longer(!sentiment, names_to = "sentiment_type", values_to = "count") %>%
  ggplot(aes(x = factor(sentiment_type), y = count, fill = sentiment, colour = sentiment)) +
  geom_bar(stat = "identity", position = "dodge") + #dodge places bars side by side
  xlab("Sentiment type") + #label stuff
  ylab("Mean sentiment") +
  ggtitle("Bar plot for sentiment by review type") +
  coord_flip()

bar_plot_nrc
```



3 PCA

3.1 Looking for Multicollinearity

```
#Selecting relevant variables
goodreads_nrc_sample<-goodreads_nrc_sample %>%
  dplyr::select(c(5,20,33:41))

cor_df<-goodreads_nrc_sample[, c(3:11)]
cor(cor_df)
```

```
##                     anger_normed anticipation_normed disgust_normed fear_normed
## anger_normed          1.000000000          0.06568045     0.68148602  0.69569230
## anticipation_normed   0.065680454          1.00000000     0.07454853  0.09793713
## disgust_normed        0.681486020          0.07454853     1.00000000  0.59972174
## fear_normed           0.695692299          0.09793713     0.59972174  1.00000000
```

```
## joy_normed          -0.005757848          0.56880590     0.04838701  0.03093264
## negative_normed      0.734680124          0.05690969     0.63822832  0.76086313
## sadness_normed       0.538612088          0.20443824     0.50969157  0.55445701
## surprise_normed      0.181031480          0.48607800     0.23696992  0.22687524
## trust_normed        -0.054342105          0.31981076    -0.02228397 -0.02234893
##                        joy_normed negative_normed sadness_normed
## anger_normed        -0.0057578483    0.7346801238     0.53861209
## anticipation_normed  0.5688058988    0.0569096940     0.20443824
## disgust_normed       0.0483870077    0.6382283236     0.50969157
## fear_normed          0.0309326355    0.7608631349     0.55445701
## joy_normed           1.0000000000   -0.0005964795     0.18924467
## negative_normed     -0.0005964795    1.0000000000     0.60289434
## sadness_normed       0.1892446657    0.6028943375     1.00000000
## surprise_normed      0.4537961785    0.1860503014     0.38858609
## trust_normed         0.3896639694   -0.0681095998     0.03373578
##                     surprise_normed trust_normed
## anger_normed              0.1810315  -0.05434210
## anticipation_normed       0.4860780   0.31981076
## disgust_normed            0.2369699  -0.02228397
## fear_normed               0.2268752  -0.02234893
## joy_normed                0.4537962   0.38966397
## negative_normed           0.1860503  -0.06810960
## sadness_normed            0.3885861   0.03373578
## surprise_normed           1.0000000   0.23998123
## trust_normed              0.2399812   1.00000000
```

No variables have a correlation above .8 so there is no cause for concern about multicollinearity or a need to remove variables before we run the regression.

3.2 Scaling all variables

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
data_pca<-goodreads_nrc_sample[, c(3:11)]

str(data_pca)
```

```
## 'data.frame':    2000 obs. of  9 variables:
##  $ anger_normed       : num  0 0.00971 0 0.01795 0.05426 ...
##  $ anticipation_normed: num  0.0508 0.0388 0.0588 0.0308 0.0388 ...
##  $ disgust_normed     : num  0 0.00971 0 0.00769 0.03101 ...
##  $ fear_normed        : num  0 0.00971 0.02941 0.02308 0.03876 ...
##  $ joy_normed         : num  0.0339 0.01942 0 0.01282 0.00775 ...
##  $ negative_normed    : num  0.0169 0.0388 0.0294 0.0308 0.0698 ...
##  $ sadness_normed     : num  0 0.02913 0.02941 0.00769 0.02326 ...
##  $ surprise_normed    : num  0.0169 0.0194 0 0.0128 0.031 ...
##  $ trust_normed       : num  0.0508 0.0194 0 0.041 0.0233 ...
```

```r
#Creating a set of predictors from the training dataset
scaled_data_pca <- data_pca  %>%
  mutate_at(c(1:9), ~(scale(.) %>% as.vector))

str(scaled_data_pca)
```

```
## 'data.frame':    2000 obs. of  9 variables:
##  $ anger_normed       : num  -0.777 -0.287 -0.777 0.13 1.965 ...
##  $ anticipation_normed: num  0.965 0.456 1.303 0.114 0.453 ...
##  $ disgust_normed     : num  -0.64789 -0.00991 -0.64789 -0.14241 1.38968 ...
##  $ fear_normed        : num  -0.92088 -0.5302 0.26265 0.00773 0.63881 ...
##  $ joy_normed         : num  0.39 -0.221 -1.041 -0.5 -0.714 ...
##  $ negative_normed    : num  -0.5169 0.2196 -0.0975 -0.0518 1.2604 ...
##  $ sadness_normed     : num  -0.778 0.595 0.608 -0.415 0.318 ...
##  $ surprise_normed    : num  0.1845 0.3268 -0.7926 -0.0535 0.995 ...
##  $ trust_normed       : num  0.3544 -0.7017 -1.3542 0.0244 -0.5727 ...
```

3.2.1 Describing the variables

```r
psych::describe(scaled_data_pca)
```

```
##                     vars    n mean sd median trimmed  mad   min   max range
## anger_normed           1 2000    0  1  -0.31   -0.19 0.69 -0.78  6.44  7.22
## anticipation_normed    2 2000    0  1  -0.16   -0.11 0.86 -1.19  9.40 10.59
## disgust_normed         3 2000    0  1  -0.65   -0.21 0.00 -0.65  6.93  7.58
## fear_normed            4 2000    0  1  -0.25   -0.16 0.99 -0.92  6.26  7.19
## joy_normed             5 2000    0  1  -0.17   -0.14 0.90 -1.04  9.51 10.55
## negative_normed        6 2000    0  1  -0.20   -0.13 1.00 -1.09  7.10  8.18
## sadness_normed         7 2000    0  1  -0.26   -0.18 0.77 -0.78 11.00 11.78
## surprise_normed        8 2000    0  1  -0.26   -0.17 0.79 -0.79 13.62 14.41
## trust_normed           9 2000    0  1  -0.12   -0.10 0.79 -1.35  8.25  9.60
##                     skew kurtosis   se
## anger_normed        1.96     5.28 0.02
## anticipation_normed 1.67     7.33 0.02
## disgust_normed      2.58     9.56 0.02
## fear_normed         1.52     3.21 0.02
## joy_normed          1.84     7.64 0.02
## negative_normed     1.27     2.86 0.02
## sadness_normed      2.86    16.63 0.02
## surprise_normed     3.31    26.75 0.02
## trust_normed        1.82     7.81 0.02
```

This tells me the distribution of the normed variables, which are all relatively similar since they have already been normed so are relatively low values.

3.2.2 Visualizing the PCA

```r
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
#line below runs a simple PCA with a component for each variable.
#the most variance will be explained in component 1 and 2

viz_pca <- prcomp(scaled_data_pca, center = TRUE,scale. = TRUE)
```

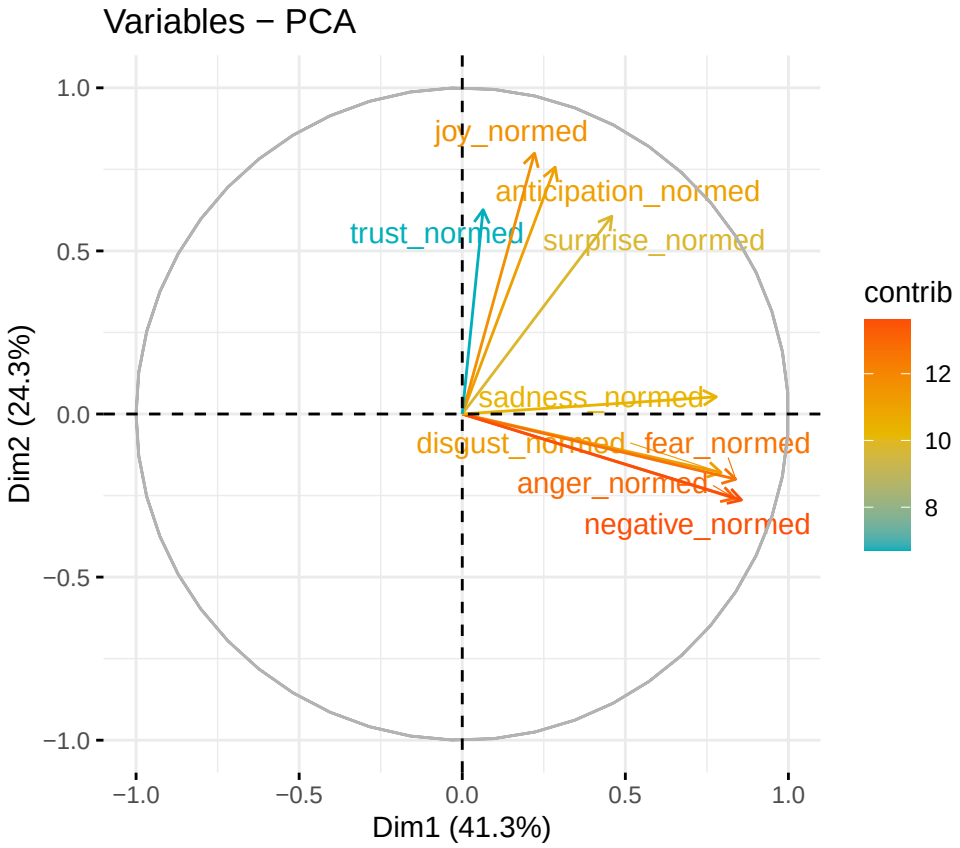
```
summary(viz_pca) #show the proportion of variance explained by all possible components along with cumul
```

```
## Importance of components:
##                            PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.9277 1.4777 0.87262 0.73845 0.68568 0.66510 0.61753
## Proportion of Variance 0.4129 0.2426 0.08461 0.06059 0.05224 0.04915 0.04237
## Cumulative Proportion  0.4129 0.6555 0.74014 0.80073 0.85297 0.90212 0.94449
##                            PC8     PC9
## Standard deviation     0.52511 0.47314
## Proportion of Variance 0.03064 0.02487
## Cumulative Proportion  0.97513 1.00000
```

```
viz_pca$rotation #show the loadings for each component by variable
```

```
##                             PC1         PC2         PC3        PC4         PC5
## anger_normed        0.43351379 -0.17491736  0.11807425 -0.1869052  0.17047922
## anticipation_normed 0.14774158  0.51214716 -0.21681028 -0.5059368  0.06226174
## disgust_normed      0.41162371 -0.12154462  0.09257829 -0.0648109  0.53888020
## fear_normed         0.43507496 -0.13523207  0.10213991 -0.1214146 -0.10758994
## joy_normed          0.11488392  0.54114642 -0.05730522 -0.3217581 -0.25571025
## negative_normed     0.44410974 -0.17843131  0.07742305 -0.1002806 -0.17334695
## sadness_normed      0.40417880  0.03562199 -0.16626755  0.4210498 -0.62524143
## surprise_normed     0.23795462  0.41038342 -0.38147797  0.5816769  0.42258723
## trust_normed        0.03315477  0.42389590  0.85883637  0.2468540  0.01221219
##                             PC6         PC7          PC8          PC9
## anger_normed         0.001273979  0.01856149 -0.808626768  0.222054159
## anticipation_normed  0.471376981  0.42573552  0.072858067 -0.016344048
## disgust_normed      -0.477770240  0.35580271  0.397194193  0.002562400
## fear_normed          0.320033300 -0.47021843  0.412961798  0.511476554
## joy_normed          -0.615524714 -0.37450881 -0.050645475  0.008357172
## negative_normed      0.148601884 -0.19765914  0.046984674 -0.813307992
## sadness_normed      -0.112069803  0.45268542  0.006499081  0.148290000
## surprise_normed      0.140031871 -0.28379474 -0.087418567 -0.068696386
## trust_normed         0.118298957  0.07690427 -0.007550541 -0.023688422
```

```
fviz_pca_var(viz_pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE #Avoid overlapping text if possible
             )
```



3.2.3 Bartlett's Test

```
cortest.bartlett(scaled_data_pca, 2000) #2000 equals sample size
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 8189.447
##
## $p.value
## [1] 0
##
## $df
## [1] 36
```

If significant, the R-matrix is not an identity matrix

There are some relationships between the variables in the analysis i.e., you can conduct a PCA if not significant, the variables are not related

3.2.4 KMO

```
KMO(scaled_data_pca)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = scaled_data_pca)
## Overall MSA =  0.84
## MSA for each item =
```

```
##        anger_normed anticipation_normed      disgust_normed         fear_normed
##                0.87                0.74                0.90                0.87
##          joy_normed     negative_normed      sadness_normed     surprise_normed
##                0.72                0.84                0.90                0.81
##        trust_normed
##                0.80
```

The test measures sampling adequacy for each variable in the model and for the complete model. All my values are above .7 which means they are good for being in the model.

3.3 Baseline PCA

```
pca_base <- principal(scaled_data_pca, nfactors = 9, rotate = "none")

pca_base #results
```

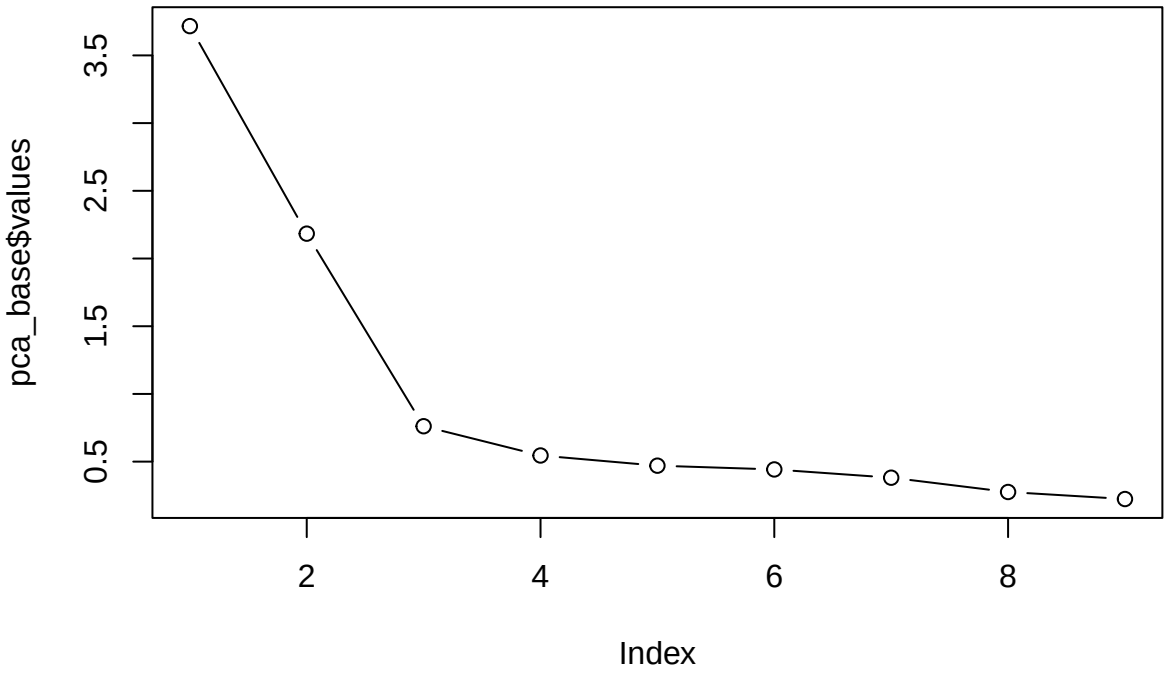
```
## Principal Components Analysis
## Call: principal(r = scaled_data_pca, nfactors = 9, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                       PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9 h2
## anger_normed         0.84 -0.26  0.10  0.14  0.12  0.00  0.01  0.42 -0.11  1
## anticipation_normed 0.28  0.76 -0.19  0.37  0.04 -0.31  0.26 -0.04  0.01  1
## disgust_normed       0.79 -0.18  0.08  0.05  0.37  0.32  0.22 -0.21  0.00  1
## fear_normed          0.84 -0.20  0.09  0.09 -0.07 -0.21 -0.29 -0.22 -0.24  1
## joy_normed           0.22  0.80 -0.05  0.24 -0.18  0.41 -0.23  0.03  0.00  1
## negative_normed      0.86 -0.26  0.07  0.07 -0.12 -0.10 -0.12 -0.02  0.38  1
## sadness_normed       0.78  0.05 -0.15 -0.31 -0.43  0.07  0.28  0.00 -0.07  1
## surprise_normed      0.46  0.61 -0.33 -0.43  0.29 -0.09 -0.18  0.05  0.03  1
## trust_normed         0.06  0.63  0.75 -0.18  0.01 -0.08  0.05  0.00  0.01  1
##                           u2 com
## anger_normed          2.2e-16 1.9
## anticipation_normed  5.6e-16 2.7
## disgust_normed        8.9e-16 2.3
## fear_normed           5.6e-16 2.0
## joy_normed            3.3e-16 2.2
## negative_normed       1.0e-15 1.8
## sadness_normed        4.4e-16 2.4
## surprise_normed      -2.2e-16 4.3
## trust_normed          3.3e-16 2.1
##
##                        PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
## SS loadings           3.72 2.18 0.76 0.55 0.47 0.44 0.38 0.28 0.22
## Proportion Var        0.41 0.24 0.08 0.06 0.05 0.05 0.04 0.03 0.02
## Cumulative Var        0.41 0.66 0.74 0.80 0.85 0.90 0.94 0.98 1.00
## Proportion Explained  0.41 0.24 0.08 0.06 0.05 0.05 0.04 0.03 0.02
## Cumulative Proportion 0.41 0.66 0.74 0.80 0.85 0.90 0.94 0.98 1.00
##
## Mean item complexity =  2.4
## Test of the hypothesis that 9 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

The only factors that are informative enough to include in the regression will have an SS loading above 1, which means it will only be PC1 and PC2 since PC3 has an SS loading value of .77

3.4 Scree Plot

```
#How many components to extract? The number of SS loadings greater than 1 (Kaiser's criterion).

#scree plot using eigen values stored in pca_1$values
plot(pca_base$values, type = "b")
```



My point of inflection looks like 3 but I know the third factor has an SS loading under 1 so I will go forward with two factors!

3.4.1 Check that resids are normally distributed

```
pca_resid <- principal(scaled_data_pca, nfactors = 2, rotate = "none")
pca_resid #results. 3 looks good
```

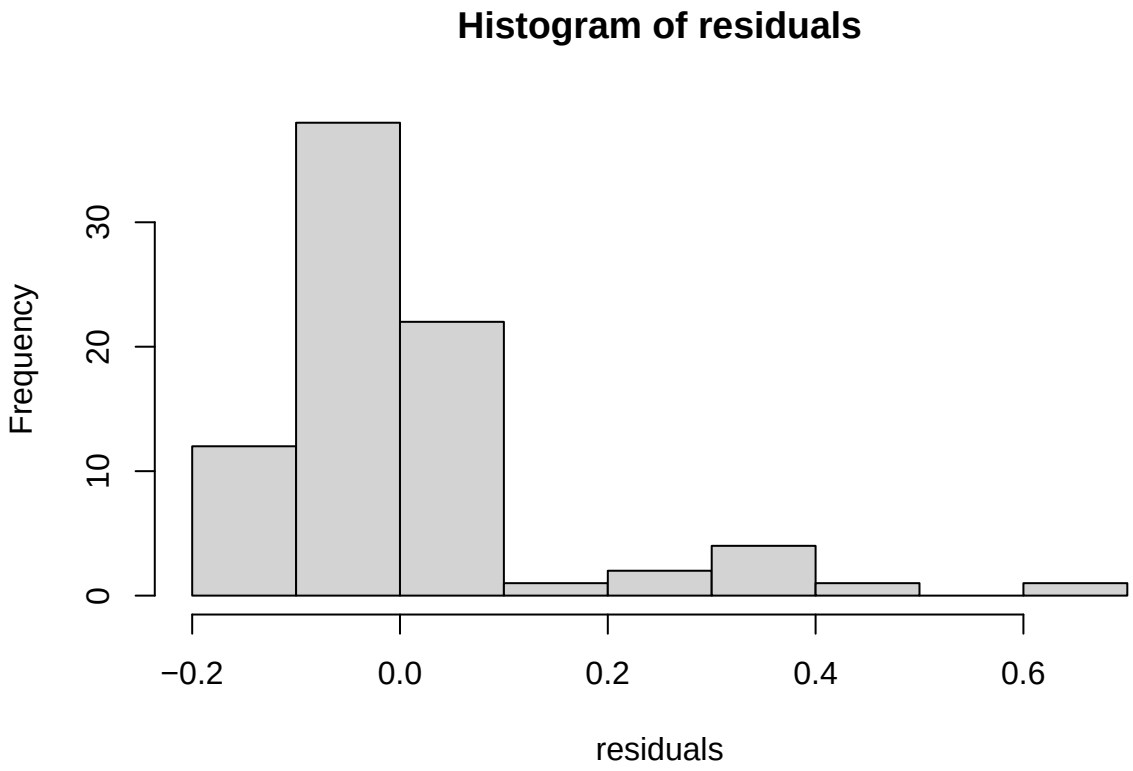
```
## Principal Components Analysis
## Call: principal(r = scaled_data_pca, nfactors = 2, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                       PC1   PC2   h2   u2 com
## anger_normed         0.84 -0.26 0.77 0.23 1.2
## anticipation_normed 0.28  0.76 0.65 0.35 1.3
## disgust_normed       0.79 -0.18 0.66 0.34 1.1
## fear_normed          0.84 -0.20 0.74 0.26 1.1
## joy_normed           0.22  0.80 0.69 0.31 1.2
## negative_normed      0.86 -0.26 0.80 0.20 1.2
## sadness_normed       0.78  0.05 0.61 0.39 1.0
## surprise_normed      0.46  0.61 0.58 0.42 1.9
## trust_normed         0.06  0.63 0.40 0.60 1.0
##
##                        PC1  PC2
```

```
## SS loadings           3.72 2.18
## Proportion Var        0.41 0.24
## Cumulative Var        0.41 0.66
## Proportion Explained  0.63 0.37
## Cumulative Proportion 0.63 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  759.79  with prob <  7.5e-149
##
## Fit based upon off diagonal values = 0.97
```

```
#residuals
#require correlation matrix for final data
corMatrix<-cor(scaled_data_pca)
#corMatrix

#next,create an object from the correlation matrix and the pca loading. Call it residuals. It will cont
residuals<-factor.residuals(corMatrix, pca_resid$loadings)

#call a histogram to check residuals
hist(residuals) #are the residuals normally distributed? They look okay. That is good
```



The residuals look realtively normally distributed so it is okay to move forward.

3.4.2 Informed PCA with specific number of components

```
#rotation. Since factors should be related, use oblique technique (promax), if unrelated, use varimax
pca_final <- principal(scaled_data_pca, nfactors = 2, rotate = "promax")
```

```
pca_final #results.
```

```
## Principal Components Analysis
## Call: principal(r = scaled_data_pca, nfactors = 2, rotate = "promax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                        RC1   RC2   h2   u2 com
## anger_normed          0.88 -0.07 0.77 0.23 1.0
## anticipation_normed -0.02  0.81 0.65 0.35 1.0
## disgust_normed        0.81 -0.01 0.66 0.34 1.0
## fear_normed           0.86 -0.02 0.74 0.26 1.0
## joy_normed           -0.09  0.84 0.69 0.31 1.0
## negative_normed       0.91 -0.08 0.80 0.20 1.0
## sadness_normed        0.71  0.22 0.61 0.39 1.2
## surprise_normed       0.20  0.70 0.58 0.42 1.2
## trust_normed         -0.18  0.63 0.40 0.60 1.2
##
##                          RC1  RC2
## SS loadings             3.59 2.31
## Proportion Var          0.40 0.26
## Cumulative Var          0.40 0.66
## Proportion Explained    0.61 0.39
## Cumulative Proportion   0.61 1.00
##
##  With component correlations of
##      RC1  RC2
## RC1 1.00 0.16
## RC2 0.16 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  759.79  with prob <  7.5e-149
##
## Fit based upon off diagonal values = 0.97
```

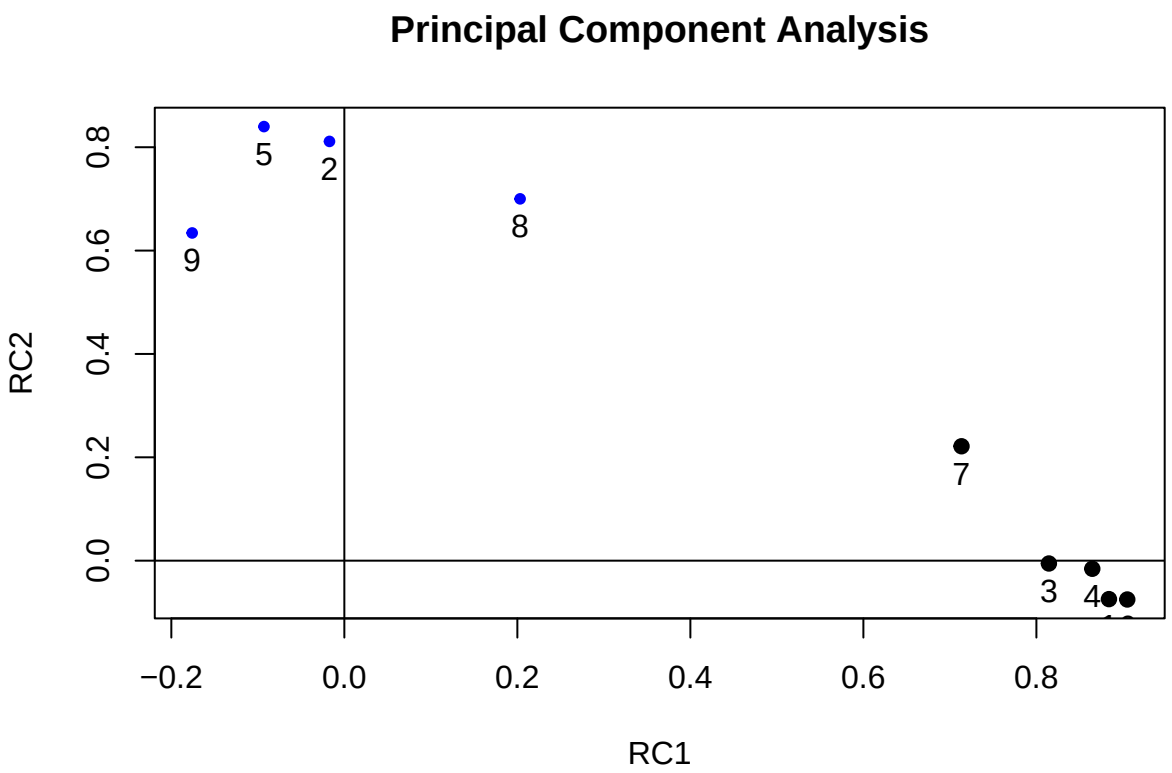
This looks good that they cumulatively explain a large proportion of the variance.

```
#let's make the results easier to read. Include loadings over 3 and sort them
print.psych(pca_final, cut = 0.3, sort = TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = scaled_data_pca, nfactors = 2, rotate = "promax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                       item   RC1   RC2   h2   u2 com
## negative_normed          6  0.91       0.80 0.20 1.0
## anger_normed             1  0.88       0.77 0.23 1.0
## fear_normed              4  0.86       0.74 0.26 1.0
## disgust_normed           3  0.81       0.66 0.34 1.0
## sadness_normed           7  0.71       0.61 0.39 1.2
## joy_normed               5        0.84 0.69 0.31 1.0
## anticipation_normed      2        0.81 0.65 0.35 1.0
## surprise_normed          8        0.70 0.58 0.42 1.2
## trust_normed             9        0.63 0.40 0.60 1.2
##
```

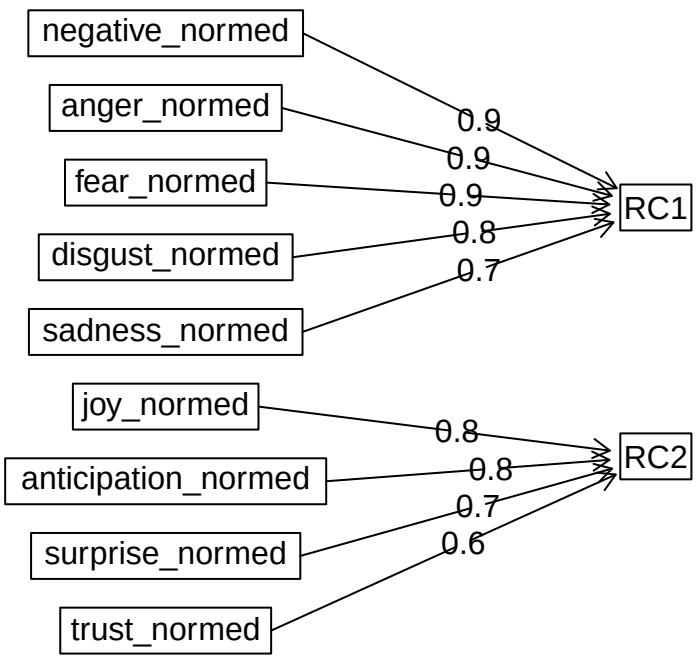
```
##                         RC1  RC2
## SS loadings            3.59 2.31
## Proportion Var         0.40 0.26
## Cumulative Var         0.40 0.66
## Proportion Explained   0.61 0.39
## Cumulative Proportion  0.61 1.00
##
##  With component correlations of
##      RC1  RC2
## RC1 1.00 0.16
## RC2 0.16 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  759.79  with prob <  7.5e-149
##
## Fit based upon off diagonal values = 0.97
```

```
plot(pca_final)
```



This shows a good distinction between the variables within each factor.

```
fa.diagram(pca_final)
```



3.4.3 Collect Factor Scores

```
#we need the pca scores
pca_final_scores <- as.data.frame(pca_final$scores) #scores for each text on each factor. You can use t
```

3.4.4 Rename PCA variables

```
pca_final_scores<-rename(pca_final_scores, "Negativity" = "RC1")
pca_final_scores<-rename(pca_final_scores, "Security" = "RC2")

str(pca_final_scores)
```

```
## 'data.frame':    2000 obs. of  2 variables:
##  $ Negativity: num  -0.855 0.0301 -0.1431 -0.0887 1.43 ...
##  $ Security  : num  0.58717 0.04954 -0.44869 -0.19459 -0.00337 ...
```

3.4.5 Merge my predictor with PCA outcome variables

```
predictor_pca<-cbind(goodreads_nrc_sample, pca_final_scores)

predictor_pca<-predictor_pca %>%
  mutate(bookrating = case_when(Rating > 3 ~ "1",
                                Rating <= 3 ~ "0"))

predictor_pca<-predictor_pca[, c(12:14)]
```

4 PCA Cross Validation

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
set.seed(1234)
train.control <- trainControl(method = "cv", number = 10)
#the GLM
lm_cv10_step <- train(bookrating ~ .,
                      data = predictor_pca,
                           method = "glm", #stepwise selection
                           family=binomial(), #using 1-7 predictor that we have
                           trControl = train.control)
#the model
summary(lm_cv10_step)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.894  -1.146  -0.168   1.184   1.311
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.001343   0.044987   0.030   0.9762
## Negativity  0.185197   0.046854   3.953 7.73e-05 ***
## Security    0.093003   0.047396   1.962   0.0497 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2772.6  on 1999  degrees of freedom
## Residual deviance: 2749.3  on 1997  degrees of freedom
## AIC: 2755.3
##
## Number of Fisher Scoring iterations: 4
```

Since only the negativity variable is statistically significant, that is the only factor I will use in my final regression.

4.1 My Final Logisitic Regression

```
str(predictor_pca)
```

```
## 'data.frame':    2000 obs. of  3 variables:
```

```
##  $ Negativity: num  -0.855 0.0301 -0.1431 -0.0887 1.43 ...
##  $ Security  : num  0.58717 0.04954 -0.44869 -0.19459 -0.00337 ...
##  $ bookrating: chr  "1" "1" "1" "1" ...
```

```
predictor_pca$bookrating<-as.numeric(predictor_pca$bookrating)

finalModel<-glm(bookrating ~ Negativity, data = predictor_pca, family = binomial)

summary(finalModel)
```

```
##
## Call:
## glm(formula = bookrating ~ Negativity, family = binomial, data = predictor_pca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7257  -1.1450  -0.1716   1.1863   1.2848
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.001093   0.044942   0.024    0.981
## Negativity  0.199989   0.046330   4.317 1.58e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2772.6  on 1999  degrees of freedom
## Residual deviance: 2753.3  on 1998  degrees of freedom
## AIC: 2757.3
##
## Number of Fisher Scoring iterations: 4
```

4.2 Final Regression Statistics

```
exp(finalModel$coefficients) #odds ratio as the exponential of the b coefficient for the predictor vari
```

```
## (Intercept)  Negativity
##    1.001093    1.221389
```

This shows that more of the negativity factor sentiments are correlated with a positive review.

4.2.1 Rerun final regression model using lrm function to get chi square, p value, R2 scores, and C index.

```
#Rerun final regression model using lrm function to get chi square, p value, R2 scores, and C index.
library(rms)
```

```
## Loading required package: Hmisc
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
```

```
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:psych':
##
##     describe
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
final_reg_stat <-lrm(bookrating ~ Negativity, data = predictor_pca)

final_reg_stat
```

```
## Logistic Regression Model
##
##  lrm(formula = bookrating ~ Negativity, data = predictor_pca)
##
##                        Model Likelihood     Discrimination    Rank Discrim.
##                              Ratio Test            Indexes          Indexes
##  Obs          2000    LR chi2      19.28     R2       0.013    C       0.568
##   0           1000    d.f.             1    R2(1,2000)0.009    Dxy     0.136
##   1           1000    Pr(> chi2) <0.0001    R2(1,1500)0.012    gamma   0.136
##  max |deriv| 1e-08                          Brier    0.248    tau-a   0.068
##
##             Coef   S.E.   Wald Z Pr(>|Z|)
##  Intercept  0.0011 0.0449 0.02   0.9806
##  Negativity 0.2000 0.0463 4.32   <0.0001
##
```

The chi square is 29.45, the p value is <0.0001, the R2 score is 0.019 and the C index is 0.588.

4.2.2 Using the test dataset

```
predicted<-predict(lm_cv10_step, newdata = predictor_pca)
pred_label <- as.factor(ifelse(predicted ==1, 0, 1))
actual<-predictor_pca$bookrating
actual<-as.factor(actual)
```

4.2.3 Prediction Tool

```
# Matrix predictions for testing set
table(predicted, actual)
```

```
##          actual
## predicted   0   1
##         0 659 533
##         1 341 467
```

```
        actual
         0   1
```

predicted 0 686 509 1 314 491

4.2.4 Creating a confusion matrix

```
# create confusion matrix using CARET
confusionMatrix(actual, predicted,
                mode = "everything", #what you want to report in stats
                positive="1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 659 341
##          1 533 467
##
##                Accuracy : 0.563
##                  95% CI : (0.5409, 0.5849)
##     No Information Rate : 0.596
##     P-Value [Acc > NIR] : 0.9987
##
##                   Kappa : 0.126
##
##  Mcnemar's Test P-Value : 1.042e-10
##
##             Sensitivity : 0.5780
##             Specificity : 0.5529
##          Pos Pred Value : 0.4670
##          Neg Pred Value : 0.6590
##               Precision : 0.4670
##                  Recall : 0.5780
##                      F1 : 0.5166
##              Prevalence : 0.4040
##          Detection Rate : 0.2335
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.5654
##
##        'Positive' Class : 1
##
```

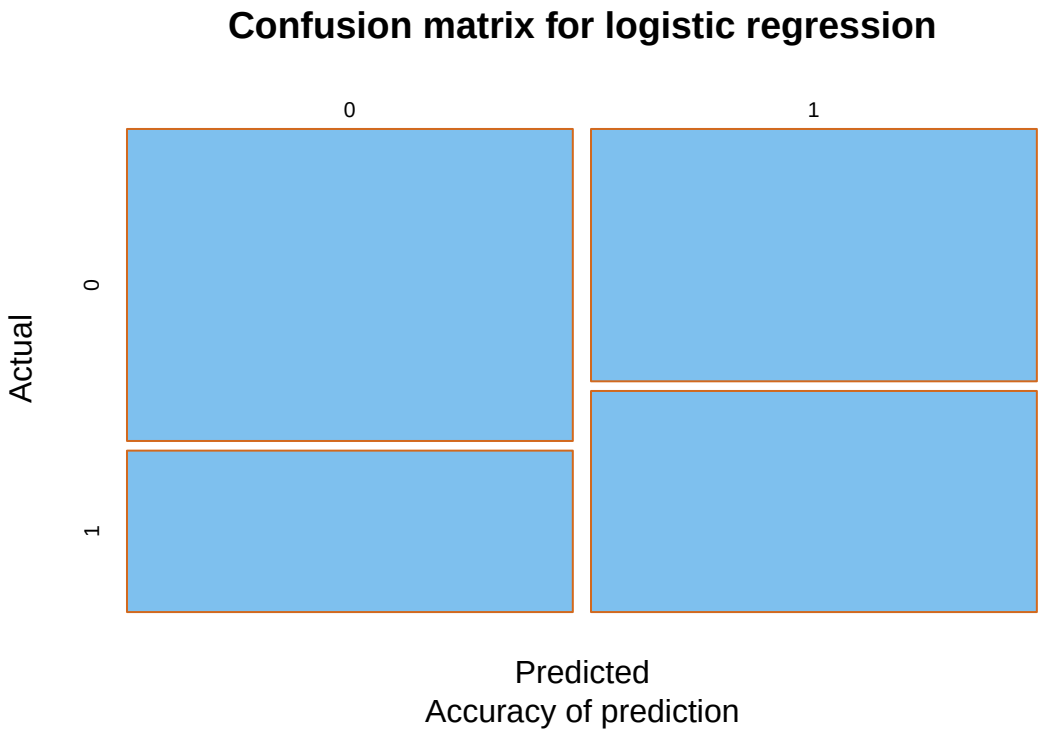
The Accuracy is 0.5885 and the Kappa value is 0.177

4.2.5 Visualize out with mosaic plot.

```
#put the actual and predicted values into a table
mosaic_table <- table(actual, predicted)
mosaic_table #check on that table
```

```
##       predicted
## actual   0   1
##      0 659 341
##      1 533 467
```

```
#simple mosaic plot
mosaicplot(mosaic_table,
           main = "Confusion matrix for logistic regression",
           sub = "Accuracy of prediction",
           xlab = "Predicted",
           ylab = "Actual",
           color = "skyblue2",
           border = "chocolate")
```



5 Discussion

I used the same data set as I had used for assignment 6, but this time my definition of good and bad ratings was not based around the mean. So, I made bad ratings anything below or equal to 3 and good ratings anything above 3, which meant that my sample size of bad ratings significantly decreased from 10000 to 1000. This, in turn, meant my stratified samples could only have a combined sample size of 2000 instead of 20000. To counteract the change in sample size, I changed my code to a 10-fold cross glm validation instead of a train and test set.

My final logistic regression looked like: bookrating = 1.290841 * Negativity

With a predicted 1 being a positive rating meaning 4 or a 5 stars and prediction close to 0 being a negative

rating meaning 1 to 3 stars. With the dimension reduction, the only statistically relevant group was the Negativity group, which included negative_normed, anger_normed, fear_normed, disgust_normed, and sadness_normed.

The actual predictive value of this model is incredibly low. With an accuracy of 0.5885 this model is slightly better at predicting whether a book will have good or bad reviews than just guessing 50-50 on each one without doing a sentiment analysis of the book description. However, my last model had an accuracy of .5143, so there is significant improvement whether from the PCA or the change in my definition of good and bad ratings. This is also shown in the kappa value being 0.177, which is better than my previous model's kappa of 0.0286 which is also incredibly low. If it was below .01 this would indicate that there is no relationship between the model predictions and the true results. At .068 there is slight agreement between the model and the true results.

The mosaic table of the confusion matrix shows how there is not a really strong predictive power on either class and the sensitivity at 0.6099 and the specificity at 0.5741 support this. These are both increases compared to my last model's predictability. There is a slightly higher positive predictive value at 0.4910 than the negative predictive value at 0.6860 but both are close to each other. This shows that the model performs significantly better on predicting bad reviews than good reviews, but this could be because in my model at large there is so many less negative reviews than positive ones since people on average rate books highly, so it is easier to distinguish the sentiments of the ones people really did not like.

For the final model, which did not experience any difficulties from suppression or multicollinearity, The chi square is 29.45, the p value is <0.0001, the R2 score is 0.019 and the C index is 0.588. The chi squared showed how the deviance (unexplained variation) of the current model differ from a model without any predictors which the p value says this it does. The R2 is variance explained which is low which is likely contributing to the low predictability and accuracy of the model. The C index is the trade-offs between sensitivity (recall or probability of true prediction given positive outcome) and specificity (probability of false prediction given negative outcome). If it was below .5 this would be a bad model or at exactly .5 the model is no better than predicting an outcome than random chance. In this case, again, the model is better than chance but not by much.

Overall, I would say that is is not super surprising that the sentiment of the book descriptions are not strong predictors of their rating since that is already not intuitively tied to the rating of a book anyways. Therefore, the sentiment of the words chosen would also not be an easy model to create. This would likely be improved if I was working off of the sentiments of the text reviews on the site but I had to work within the limitations of my data set, which is why I chose to do the more expansive sentiment dictionary in hopes of having a more interesting relationship to discover. By creating a more distinct difference between the good and the bad ratings and using a PCA, I was able to successfully increase the model accuracy, but not by much. Just like its not easy to judge a book by its cover, it is apparently even more difficult to judge a book by the sentiment of its description!