M3 Homework Assignment

Joshua Conte

October 1, 2017

1 M3 Homework Assignment

R studio was configured with the following parameters before beginning the project:

```
# clears the console in RStudio
cat("\014")
```

```
# clears environment
rm(list = ls())

# Load required packages
require(ggplot2)
```

1.1 Load U.S. Bureau of Labor Statistics (BLS) Data.

I opened the data using read.csv2, I set stringsAsFactors = FALSE so that the strings in a data frame will be treated as plain strings and not as factor variables, row names is set at null since there are no row names, headers is true since the data has headers, and I added NA strings in case there is missing data. Below is my R code:

```
# Some csv files are really big and take a while to open.  This command checks to
# see if it is already opened, if it is, it does not open it again.
# I also omitted the first column

if (!exists("dfBLS")) {
  dfBLS<-
    read.csv2(
     "2014.annual.singlefile.csv",
      sep = ",",
      stringsAsFactors = FALSE,
      row.names = NULL,
      header = TRUE,
      na.strings=c("","NA")
    )
}

# This is another way of opening the file, but it takes R a really long time to
# complete these tasks, so I have it commented out as a reference:

# if (!exists("dfBLS")) {
#   data_url <-
#     'http://www.bls.gov/cew/data/files/2014/csv/2014_annual_singlefile.zip'
#   temp <- tempfile()
#   download.file(data_url, temp)
#   dfBLS <-
#     read.csv2(
#       unz(temp, "2014.annual.singlefile.csv"),
#       sep = ",",
#       stringsAsFactors = FALSE,
#       row.names = NULL,
#       header = TRUE,
#       na.strings=c("","NA")
#     )
#   unlink(temp)
```

```
# }
```

1.2 Run Principal Components Analysis

I ran str() to display the internal structure of the data frame below. According to the results, there is a mixture of text, integers, and numeric data. However, for principal components analysis (PCA), R allows only numeric data to be used.

In order to get a better understanding of how the data should be organized, I referenced the government website (https://data.bls.gov/cew/doc/access/csv_data_slices.htm) that has a description of the data. According to the website, the Annual Average Data Slice Layout has 10 text types and 28 numeric types. Since PCA requires numeric data, I removed all of the text types and converted everything else to numeric. The process is shown below:

```
# Check data.
str(dfBLS)
```

```
## 'data.frame':    3569127 obs. of  38 variables:
##  $ area_fips                       : chr  "01000" "01000" "01000" "01000" ...
##  $ own_code                        : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ industry_code                   : chr  "10" "10" "102" "1021" ...
##  $ agglvl_code                     : int  50 51 52 53 53 53 53 53 53 53 ...
##  $ size_code                       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ year                            : int  2014 2014 2014 2014 2014 2014 2014 2014 2014 2014 ...
##  $ qtr                             : chr  "A" "A" "A" "A" ...
##  $ disclosure_code                 : chr  NA NA NA NA ...
##  $ annual_avg_estabs               : int  117452 1186 1186 587 2 17 43 34 26 3 ...
##  $ annual_avg_emplvl               : int  1863561 53491 53491 11462 13 147 1712 6239 810 500 ...
##  $ total_annual_wages              : num  8.07e+10 4.15e+09 4.15e+09 7.20e+08 4.31e+05 ...
##  $ taxable_annual_wages            : num  1.39e+10 0.00 0.00 0.00 0.00 ...
##  $ annual_contributions            : num  3.16e+08 0.00 0.00 0.00 0.00 ...
##  $ annual_avg_wkly_wage            : int  832 1491 1491 1207 649 1527 1616 1283 383 2085 ...
##  $ avg_annual_pay                  : int  43287 77550 77550 62776 33771 79389 84056 66729 19904 1084
##  $ lq_disclosure_code              : chr  NA NA NA NA ...
##  $ lq_annual_avg_estabs            : chr  "1.00" "1.55" "1.55" "1.55" ...
##  $ lq_annual_avg_emplvl            : chr  "1.00" "1.44" "1.46" "1.26" ...
##  $ lq_total_annual_wages           : chr  "1.00" "1.74" "1.77" "1.60" ...
##  $ lq_taxable_annual_wages         : chr  "1.00" "0.00" "0.00" "0.00" ...
##  $ lq_annual_contributions         : chr  "1.00" "0.00" "0.00" "0.00" ...
##  $ lq_annual_avg_wkly_wage         : chr  "1.00" "1.21" "1.22" "1.27" ...
##  $ lq_avg_annual_pay               : chr  "1.00" "1.21" "1.21" "1.27" ...
##  $ oty_disclosure_code             : chr  NA NA NA NA ...
##  $ oty_annual_avg_estabs_chg       : int  1394 -10 -10 0 0 0 -3 2 0 0 ...
##  $ oty_annual_avg_estabs_pct_chg   : chr  "1.2" "-0.8" "-0.8" "0.0" ...
##  $ oty_annual_avg_emplvl_chg       : int  18475 -1097 -1097 -75 0 -7 -48 -6 -52 -66 ...
##  $ oty_annual_avg_emplvl_pct_chg   : chr  "1.0" "-2.0" "-2.0" "-0.7" ...
##  $ oty_total_annual_wages_chg      : num  2.67e+09 9.79e+07 9.79e+07 2.48e+07 -7.29e+03 ...
##  $ oty_total_annual_wages_pct_chg  : chr  "3.4" "2.4" "2.4" "3.6" ...
##  $ oty_taxable_annual_wages_chg    : num  3.11e+08 0.00 0.00 0.00 0.00 ...
##  $ oty_taxable_annual_wages_pct_chg: chr  "2.3" "0.0" "0.0" "0.0" ...
##  $ oty_annual_contributions_chg    : num  -70421983 0 0 0 0 ...
##  $ oty_annual_contributions_pct_chg: chr  "-18.2" "0.0" "0.0" "0.0" ...
##  $ oty_annual_avg_wkly_wage_chg    : int  19 64 64 49 -25 43 38 31 5 398 ...
##  $ oty_annual_avg_wkly_wage_pct_chg: chr  "2.3" "4.5" "4.5" "4.2" ...
```

```
##  $ oty_avg_annual_pay_chg          : int  1011 3353 3353 2552 -1258 2195 1984 1635 256 20696 ...
##  $ oty_avg_annual_pay_pct_chg      : chr  "2.4" "4.5" "4.5" "4.2" ...
```

```
# First, keep only data needed, this process removes only text information as
# outlined by the website, https://data.bls.gov/cew/doc/access/csv_data_slices.htm
# In this case it is easier to drop than keep:
drops <-
  c(
    "area_fips",
    "own_code",
    "industry_code",
    "agglvl_code",
    "size_code",
    "year",
    "qtr",
    "disclosure_code",
    "lq_disclosure_code",
    "oty_disclosure_code"
  )

# This is a new data frame with the test information removed.  I did this in case I
# need to access the original information again.
dfBLS2 <- dfBLS[,!(names(dfBLS) %in% drops)]

# change to numeric
dfBLS2[1:28] <- sapply(dfBLS2[1:28], as.numeric)

# Check data again
str(dfBLS2)
```

```
## 'data.frame':    3569127 obs. of  28 variables:
##  $ annual_avg_estabs               : num  117452 1186 1186 587 2 ...
##  $ annual_avg_emplvl               : num  1863561 53491 53491 11462 13 ...
##  $ total_annual_wages              : num  8.07e+10 4.15e+09 4.15e+09 7.20e+08 4.31e+05 ...
##  $ taxable_annual_wages            : num  1.39e+10 0.00 0.00 0.00 0.00 ...
##  $ annual_contributions            : num  3.16e+08 0.00 0.00 0.00 0.00 ...
##  $ annual_avg_wkly_wage            : num  832 1491 1491 1207 649 ...
##  $ avg_annual_pay                  : num  43287 77550 77550 62776 33771 ...
##  $ lq_annual_avg_estabs            : num  1 1.55 1.55 1.55 1.24 1.7 2.26 2.19 1.31 1.45 ...
##  $ lq_annual_avg_emplvl            : num  1 1.44 1.46 1.26 0.14 0.79 1.92 1.21 1.13 7.92 ...
##  $ lq_total_annual_wages           : num  1 1.74 1.77 1.6 0.08 ...
##  $ lq_taxable_annual_wages         : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ lq_annual_contributions         : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ lq_annual_avg_wkly_wage         : num  1 1.21 1.22 1.27 0.55 0.88 1.2 1.1 0.55 2.47 ...
##  $ lq_avg_annual_pay               : num  1 1.21 1.21 1.27 0.55 0.88 1.2 1.1 0.55 2.47 ...
##  $ oty_annual_avg_estabs_chg       : num  1394 -10 -10 0 0 ...
##  $ oty_annual_avg_estabs_pct_chg   : num  1.2 -0.8 -0.8 0 0 0 -6.5 6.2 0 0 ...
##  $ oty_annual_avg_emplvl_chg       : num  18475 -1097 -1097 -75 0 ...
##  $ oty_annual_avg_emplvl_pct_chg   : num  1 -2 -2 -0.7 0 -4.5 -2.7 -0.1 -6 -11.7 ...
##  $ oty_total_annual_wages_chg      : num  2.67e+09 9.79e+07 9.79e+07 2.48e+07 -7.29e+03 ...
##  $ oty_total_annual_wages_pct_chg  : num  3.4 2.4 2.4 3.6 -1.7 -2.2 -0.4 2.4 -4.7 9.2 ...
##  $ oty_taxable_annual_wages_chg    : num  3.11e+08 0.00 0.00 0.00 0.00 ...
##  $ oty_taxable_annual_wages_pct_chg: num  2.3 0 0 0 0 0 0 0 0 0 ...
##  $ oty_annual_contributions_chg    : num  -70421983 0 0 0 0 ...
##  $ oty_annual_contributions_pct_chg: num  -18.2 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ oty_annual_avg_wkly_wage_chg    : num  19 64 64 49 -25 43 38 31 5 398 ...
##  $ oty_annual_avg_wkly_wage_pct_chg: num  2.3 4.5 4.5 4.2 -3.7 2.9 2.4 2.5 1.3 23.6 ...
##  $ oty_avg_annual_pay_chg          : num  1011 3353 3353 2552 -1258 ...
##  $ oty_avg_annual_pay_pct_chg      : num  2.4 4.5 4.5 4.2 -3.6 2.8 2.4 2.5 1.3 23.6 ...
```

1.2.1 princomp()

I will use princomp to perform PCA on this data. princomp() performs a principal components analysis on the given numeric data matrix and returns the results as an object of class princomp.

```
# Save the fit to a fit object to run additional analysis later.
dfBLS2.fit.A <-
  princomp(
    formula = ~ .,
    data = dfBLS2,
    cor = TRUE,
    na.action = na.exclude
  )
```

1.2.2 Extracting Components

To understand how many components should be extracted from the dfBLS2.fit, I need to run some tests such as basic analysis of the data, screeplots, and biplots.

1.2.2.1 Analysis

Summary will give me a good understanding of what the data entails.

```
summary(dfBLS2.fit.A)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     3.0871244 1.9539345 1.6826951 1.55623688 1.33638121
## Proportion of Variance 0.3403692 0.1363521 0.1011237 0.08649547 0.06378267
## Cumulative Proportion  0.3403692 0.4767213 0.5778450 0.66434046 0.72812313
##                           Comp.6     Comp.7     Comp.8     Comp.9
## Standard deviation     1.18300674 1.16297817 1.04012465 0.98557329
## Proportion of Variance 0.04998232 0.04830422 0.03863783 0.03469124
## Cumulative Proportion  0.77810545 0.82640967 0.86504750 0.89973874
##                           Comp.10    Comp.11    Comp.12     Comp.13
## Standard deviation     0.93421215 0.79887577 0.64953541 0.500375063
## Proportion of Variance 0.03116973 0.02279295 0.01506772 0.008941972
## Cumulative Proportion  0.93090847 0.95370142 0.96876914 0.977711111
##                           Comp.14     Comp.15     Comp.16     Comp.17
## Standard deviation     0.40175783 0.387520708 0.363070080 0.304467983
## Proportion of Variance 0.00576462 0.005363296 0.004707853 0.003310741
## Cumulative Proportion  0.98347573 0.988839027 0.993546880 0.996857621
##                           Comp.18      Comp.19      Comp.20      Comp.21
## Standard deviation     0.175743873 0.1341101339 0.1268132207 0.1142661371
## Proportion of Variance 0.001103068 0.0006423403 0.0005743426 0.0004663125
## Cumulative Proportion  0.997960689 0.9986030297 0.9991773723 0.9996436848
##                             Comp.22      Comp.23      Comp.24      Comp.25
## Standard deviation     0.0838781775 4.800663e-02 2.502386e-02 2.313752e-03
## Proportion of Variance 0.0002512696 8.230846e-05 2.236405e-05 1.911946e-07
```

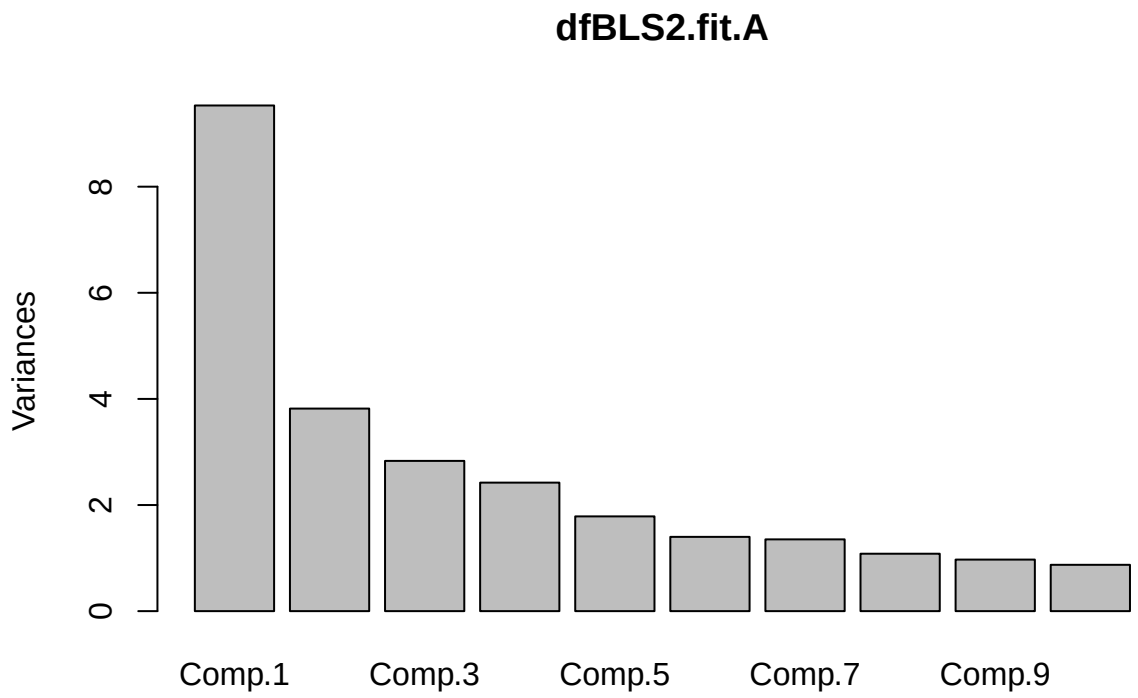
```
## Cumulative Proportion  0.9998949544 9.999773e-01 9.999996e-01 9.999998e-01
##                            Comp.26      Comp.27      Comp.28
## Standard deviation     2.017401e-03 9.981984e-04 1.649247e-04
## Proportion of Variance 1.453539e-07 3.558571e-08 9.714336e-10
## Cumulative Proportion  1.000000e+00 1.000000e+00 1.000000e+00
```

The proportion of variance is telling me that 34.0% of the variance can be captured by taking the first component, 13.6% by taking the second component,... etc. The cumulative proportion is a running total of what variance can be captured, for example 34.0% with component 1, 47.6% with comp 1 and 2,... etc. To get at least 75% of the variance, the first 6 componets are required.

1.2.2.2 Screeplot

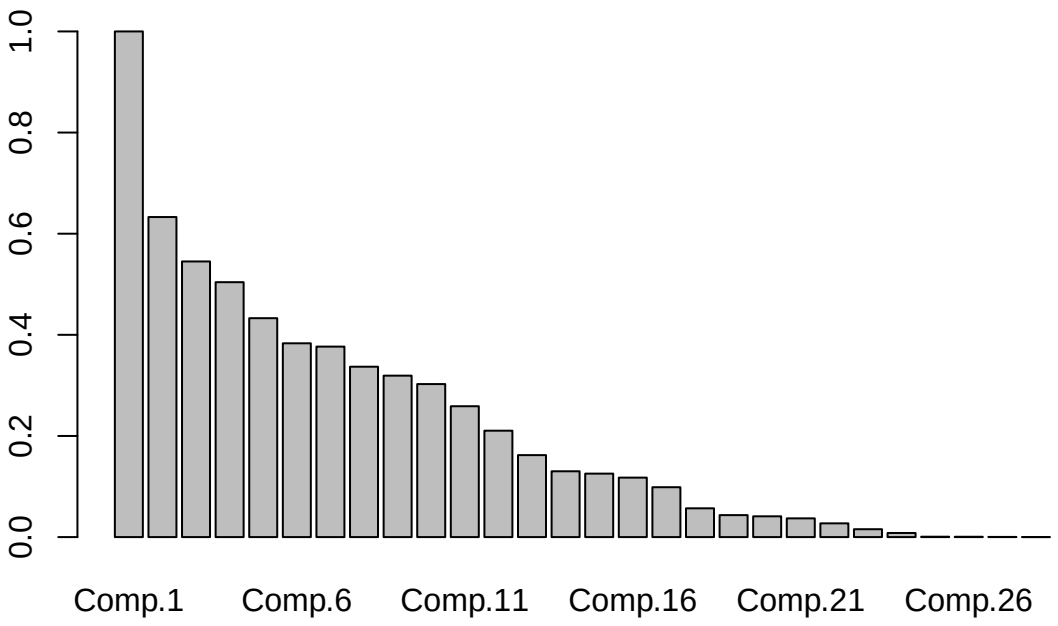
Screeplots plots variances against the number of the principal component.

```
screeplot(dfBLS2.fit.A)
```



Since the first principal component is large and there are 28 components, I scaled the principal component's by the first principal component.

```
barplot(dfBLS2.fit.A$sdev/dfBLS2.fit.A$sdev[1])
```



1.2.2.3 Biplots

Evidence of clustering in the data can be visualized by looking scores of the first principal component against the scores of the second, scores of the second principal component against the scores of the third, etc. These are called biplots of the principal component's. Since this R function takes a really long time to load, I commented out the function and inserted a .jpg of the results, refer to Figure 1 for the plot:

```
#biplot(dfBLS2.fit.A)
```

It was really hard to read that plot so I zoomed in a little with this command (see Figure 2 for the plot)

```
#biplot(dfBLS2.fit.A, expand=500, xlim=c(-0.20, 0.1), ylim=c(-0.2, 0.1))
```

1.3 Questions

1 .What proportion of the total variation in the data is explained by each of the principal components?

34.0% of the total variance is captured by taking the first component, the second component has 13.6% of the total variance so comp 1 and 2 has then 47.6% of the total variance. Each component has less and less variance, 95% of the total variance is contained in the first 11 of 28 components. That means the last 17 components have less than 5% of the total variance.

2. What happens when you plot a screeplot?

The scree plot visually shows the variance information for each component. It shows that most of the variance is contained in the first set of plots and it gets low and lower for each additional component.

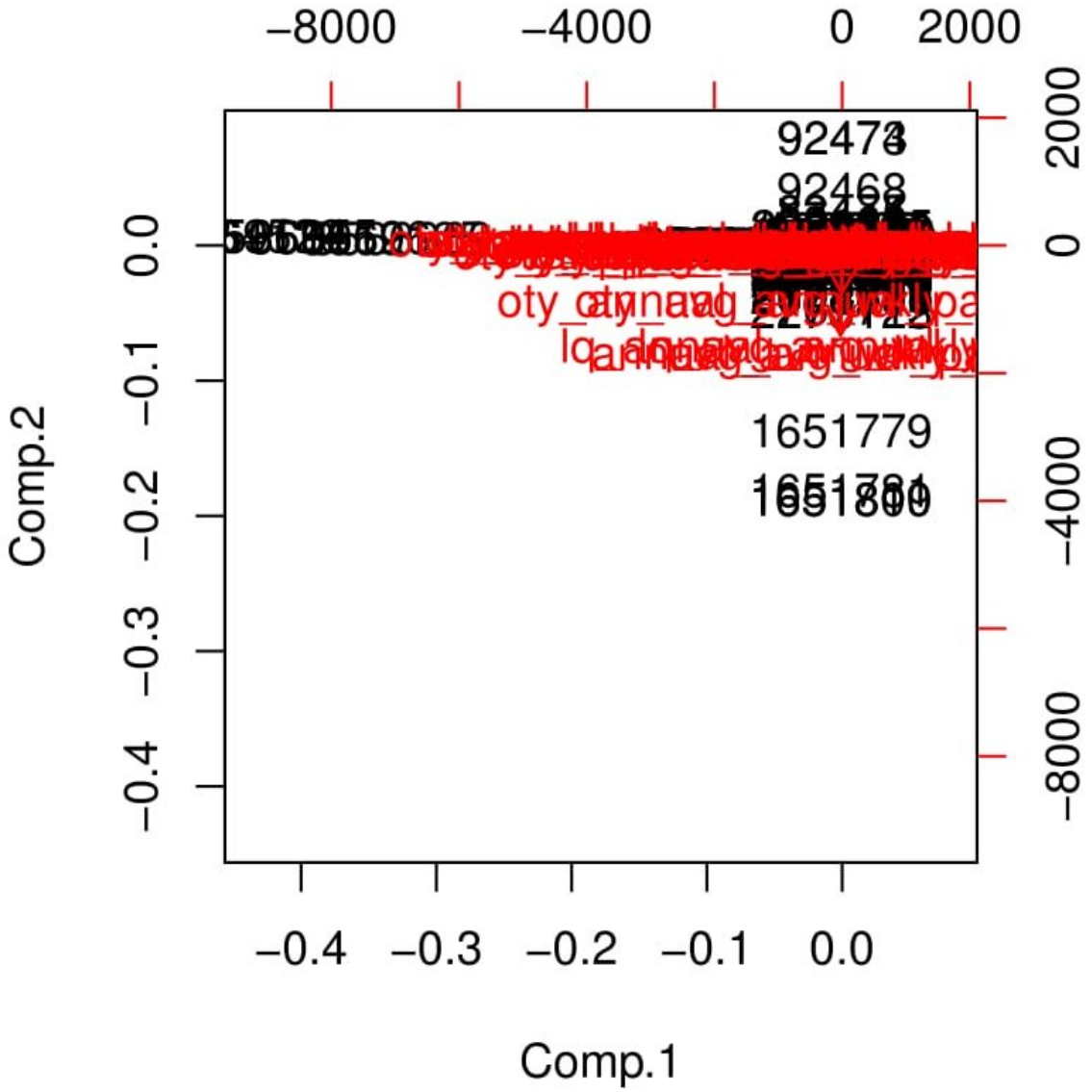


Figure 1: Biplot Image

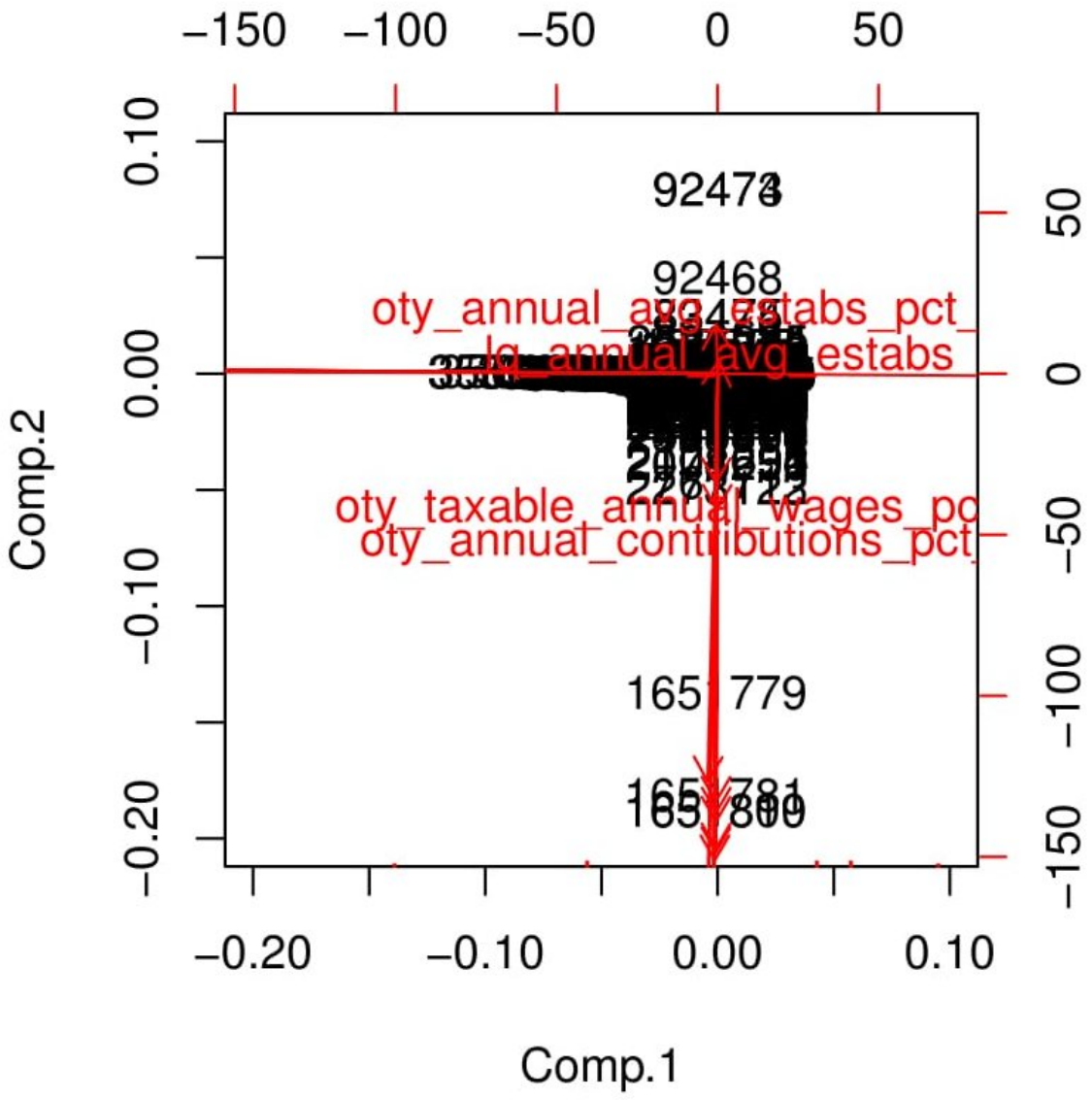


Figure 2: Zoomed Biplot Image

3. Based on the variation explained for each of these components, which, if any, components would you use? Why?

I would use the first 11 of 28 components, the first 11 contain 95% of the variance, anything after that is so small that I do not think it would have an effect on the analysis.

4. Is there evidence of clustering in the data by creating biplots of the each of the components plotted against one another? Why or why not?

The biplot looks like everything is very close together, with an exception to a couple one off points. The points look to be in between -50 and 50, and -0.05 and 0.05, I think it would be nicer if they were more spread out. For the vector points, there are too many to make sense of it all. If I was going to do a detailed analysis, I would have to figure out a way to spread the variables out a little bit more. Overall, I would call these data "clustered", since there are so many points in a small area.

5. Do any of the biplots reveal any interesting structure?

No, it looks like a blob of points in the middle of a graph.

6. How many pcs are required to explain 75% of the variance in the data?

Six components are required, they would explain 77% of the variance, refer to section 1.2.2.1 Analysis, for the detailed information.