A2_DSC424

Alex Teboul

October 6, 2019

##DSC 324/424 ##Assignment 2 (DUE SUNDAY, October 6th by Midnight)

Problem 1

Problem 1: (10 Points) Post to the final project forum with the following:

Subject Area Housing Market

Source of Data https://www.kaggle.com/c/house-prices-advanced-regression-techniques/ (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/)

Dataset Description * Number of Metric Variables: 33 * Number of Categorical Variables: 47 * Number of Samples: 1459 * Number of Tables: 1 * Note that the attribute counts are raw numbers. The actual number being used in the analysis will be reduced (namely the categorical ones).

Group Members Alex Teboul, Gianna LoVerde, Joshua Smith, Sara Elkasevic, Yvonne Renard

Planned Technology Usage We plan to use R to analyze this data.

Problem 2

Problem 2: (10 points) Answer each of the following questions:

a) What are the advantages and disadvantages of using ridge regression and lasso regression? How are these regressions different?

- When you have relatively small sample sizes, then Ridge Regression can be used to improve predictions made on new data by making such predictions less sensitive to the training data (variance is reduced). This is accomplished by adding a penalty to the minimized parameter (ex. sum of squared residuals + lambda x Slope^2). The lambda is generally selected using cross-validation or in R picks the one that minimizes MSE. When we don't have enough data to find Least Squares estimates, Ridge Regression is a handy tool, as it can still find a solution using cross-validation and that penalty. Ridge sacrifices some bias to decrease beta standard deviations. Has significance computation.
- Ridge regression modifies the regression formula.
- Lasso Regression is pretty similar to Ridge Regression, but instead of doing (sum of squared residuals + lambda x Slope^2) it's (sum of squared residuals + lambda x AbsoluteValue_Slope). So we add a bit of bias, and get less variance as a result (which is helpful in situations where you don't have many samples).
- The difference between Lasso and Ridge is that in Lasso that penalty of lambda +|slope| means that certain parameters can end up going to zero in your equation, whereas in Ridge they only approach zero. So the advantage of Lasso is that it can exclude those useless variables from your regression equation and is therefore slightly better at reduction model variance when there are many such useless variables. Lasso minimized residual error while keeping sum abs betas low. No significance computation
- If however, you know that or most of you variables are necessary to the model because of some domain expertise, then Ridge is a better bet because all variables will stay in the model.

- Lasso is an optimization technique that has the benefit of also performing variable selection.

b) What are some causes of overfitting? How do we diagnose and treat overfitting in regression models?

- Overfitting is when your model is too closely fit to a certain dataset. This is commonly diagnosed when a model performs well on training data but poorly on testing data. Specifically, a high accuracy or other metric on the training data, but a significanrly lower value for the testing. Also can be diagnosed by large hard to explain betas in a regression.
- Some causes of overfitting include: a small number of samples, too many parameters relative to the number of samples, too complex of a model, etc.
- Some ways to address overfitting include: selecting a less complex model or a different type of model all together. Regularization is also effective in many cases. With Ridge and Lasso regression the added penalty with reduce model complexity and likely address overfitting in certain circumstances.

c) What is multicollinearity? How do we diagnose and treat multicollinearity in regression models?

- Multicollinearity is caused by high correlations between variables/parameters in your model which should be independent of each other. When multicollinearity exists, it will cause problems with your model fit and also interpretting the regression equation.
- Two ways to diagnose multicollinearity include computing a correlation matrix to identify which variables have correlations of 0.7 and above and to use the VIF which identifies multicollinearity for VIF>=10. Also signs like betas close to zero or with wrong sign can indicate multicollinearity in some cases.
- To treat this we can remove those variables that are highly correlated directly or use a variable selection technique (forward, backward, stepwise), transform the data, remove those with VIF >=10, or use a technique like Lasso Regression which performs the variable selection at the same time. *PCA also eliminates most multicollinearity by rotating the data.

Problem 3

Problem 3: (Paper review 1) (10 Points) An academic paper from a conference or Journal will be posted to the Homework 2 content section of D2L. Review the paper and evaluate their usage of Factor Analysis. In particular address the following: (See article on What pulls ancestral tourist home An Analysis of ancestral tourist motivations)

• How are they applying Factoring Analysis?

- They are applying factor analysis to survey data collected on 282 ancentral tourists for the purpose of exploring how one’s heritage influences tourist motivations. The premise was that identifying factors contributing ancestral heritage tourism (practice of travelling to a place which your forebearers came from) could inform marketing and services geared towards this market segment.

• What kind of factor rotation do they use?

- They analyzed the data “using principal axis factoring with a promax rotation in SPSS 22.0,” (Murdy et al, 2017, p.15).
- Rotating components helps to re-distribute variance among components in order to simplifiy the loading matrix.
- The promax rotation means that they rotated the components in an oblique manner where components were dependent on eachother. Though in class we discussed this method as computationally efficient and therefore a good choice for large datasets, whereas this dataset is quite small.

• How many factors do they concentrate on in their analysis? How did they arrive at these number of factors?

- 3 factors: ancestral tourist motivation; heritage tourist motivations; and mass tourist motivation - and cross-loading attribute Scottish identity for exploration
- They also used k-means to cluster the ancestral tourists into 4 segments from the factor analysis: ancestral tourism motivation; mass tourism motivation; and heritage tourism motivation.
- They used the principal axis factoring with promax rotation to arrive at those 3 factors, SPSS did the calculations. Though from Table 4 on page 16, we can see the results of the exploratory factor analysis.

• Explain the breakdown of the factors and the significance of their names.

- Ancentral Tourism motivation is influenced more by attributes like family tree and pursing ancentral doc evidence, while mass tourism falls in line with general tourist attractions and drinks, and heritage tourism more on the side of culture, country, and history.

Table 4
Exploratory factor analysis.

Attributes	Factors		
	Ancestral tourism motivation	Mass tourism motivation	Heritage tourism motivation
Family tree	0.871		
Obtain documentary evidence	0.820		
Know where they lived	0.707		
How they lived	0.699		
Entertainment		0.742	
Shop for Scottish products		0.710	
Whisky		0.607	
Tourist attractions		0.575	
Local food		0.527	
Scottish country/ wildlife			0.769
Culture and heritage			0.748
Explore Scottish history			0.721
Scottish identity			0.430
α	0.87	0.78	0.78

A2_problem3_pic

• How do they evaluate the stability of the components (i.e. factorability)?

- They evaluate factorability using all three methods we discussed in class.
- Examination of KMO, KMO=0.79 and 67.16% of variance explained.
- Bartlett's Test of Sphericity, p=0.000
- Cronbach's alphas above the 0.70 cut-off

• Do they use these factors in later analysis, such as regression? If so, what do they discover?

- They do not do regression analysis after this, but they do find clusters with k-means, use chi-squared analysis on the cluster differences, and ANOVA to indicate contribution of the factors to the differences between clusters. Basically, they find that gender doesn't change much, general interest/heritage focus clusters were more likely to cite ancentral reasons for their travel, and that those with very specific purposes for visiting as ancestral enthusiasts or full heritage immersion were likely to contact such organizations for their travels.

• What overall conclusions does Factor Analysis allow them to draw?

- The factor analysis helped them to identify three factors related to ancestral tourism that help provide greater understanding of tourist motivations (which are not well understood in this context). This also led to their further segmentation of the data to find those clusters. As it pertains to the Scottish tourism industry, they conclude that their scale will be valuable to develop promotional materials, make sure that ancestral/cultural offerings are more effectively marketed, and ancestral services are better delivered.

Problem 4

Problem 4: (Paper review 2) (10 Points) An academic paper from a conference or Journal will be posted to the Homework 2 content section of D2L. Review the paper and evaluate their usage of PCA. In particular address the following: (See article on How to Make a Successful Movie Factor Analysis from both Financial and Critical Perspectives)

• How are they applying Principal Component Analysis?

- First, the researchers are studying what makes a successful movie in terms of financial ROI and movie ratings as obtained through the IMDB dataset. For movies released between 1996 and 2016, their dataset included 6981 movies. Failures were labelled as ROI less than 0 and user rating below global average, and successes the opposite - leading to 2076 success and 1960 failures in the dataset.
- Now, as to how they applied PCA: they use PCA to reduce noise and convert observations from the set of potentially correlated variable to linearly uncorrelated ones. The resulting reduced represenation of the dataset helped reveal feature relationships and improve computational efficiency. They found 5 components to explain 66% of the variance in the success label.

• What kind of rotation do they use?

- They don’t specify a rotation method, but they may have used promax because when discussing the results they discuss in parts how components are dependent on each other and it would also have led to fast computation as well.

• How many components do they concentrate on in their analysis? How did they arrive at these number of components?

- They concentrate on 5 components in their analysis, which they arrive at using PCA (the associated eigenvalue greater than 1). They are able to explain 66.62% of the variance in the success label.

• Explain the breakdown of the components and the significance of their names.

Table 2. PCA component & SVM weight matrix

Feature		PCA components				
		PC-1	PC-2	PC-3	PC-4	PC-5
Basic features	genre (Drama)			0.9314		
	genre (Comedy)			0.5852		
	genre (Western)					0.8259
	rating (USA:TV-MA)				0.4208	
	rating (USA:PG)				0.3463	
Advanced features	tAPt	0.3262				
	aAPa	0.3905				
	tAGt	0.3794				
	tAGa	0.3344				
	tpAPt	0.3165				
	tpAPa	0.3097				
	aDCP	0.4477	0.3429			
	aDRa		0.3736			
	aARa		0.3467			
	aDGa		0.4099			
Topic modeling features	topic_5 (family)					0.5750
	topic_11 (marriage and family)					0.4356
Eigenvalue		6.7552	3.2879	2.8152	1.1912	1.0574
Explained variance		29.78%	14.49%	12.41%	5.25%	4.63%
SVM model weight		0.1704	1.2955	0.2243	−0.0262	−0.1447
Total explained variance		66.62%				

A2_problem4_pic

- No real significance to the names of the components as they're just listed in order of explained variance. PC1 was based on financial indicators, PC2 has more to do with actor-drector relationships and acting careers, PC3 describes more comedy/drama impacts, PC4 is influenced by certification, and PC5 is mainly about family being an important topic to movie goers.

• How do they evaluate the stability of the components (i.e. factorability)?

- They do not appear to evaluate the stability of components using any of the 3 methods discussed in class. The jump right into inputing to an SVM and then using evaluation metrics on that. They do use Hamming loss and Matthews coefficient but that's for the SVM model using the principal components.

• Do they use these components in later analysis, such as regression? If so, what do they discover?

- Yes they use the components in their support vector machine model for which they achieve .79 accuracy (79%) with similar precision/recall and F1 score. They discover that the can predict relatively well the success of a movie from those limited principal components.

• What overall conclusions does Principal Component Analysis allow them to draw?

- Their overall conclusions were that positive actor/director relationships and successful histories plus a film in the drama or comedy genre that speak to family are more likely to be succesful. They also conclude that considering these factors those who invest in movies will be able to produce better grossing and more highly acclaimed films (more often).

Problem 5

Problem 5: (Principal Component Analysis - 20 points): The data given in the file 'nutrition.csv' is the 8,618 rows with 45 different variables from the USDA National Nutrient Database. Techniques such as Principal Component Analysis (PCA) can be used to determine different nutritional groupings.

```
#Libraries
library(Hmisc) #Describe Function
```

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

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

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

```
## Loading required package: ggplot2
```

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

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

```
library(psych) #Multiple Functions for Statistics and Multivariate Analysis
```

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

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

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

```
library(GGally) #ggpairs Function
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggplot2) #ggplot2 Functions
library(vioplot) #Violin Plot Function
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.6: type help(sm) for summary information
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(corrplot) #Plot Correlations
```

```
## corrplot 0.84 loaded
```

```
library(REdaS) #Bartlett's Test of Sphericity
```

```
## Loading required package: grid
```

```
library(psych) #PCA/FA functions
library(factoextra) #PCA Visualizations
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library("FactoMineR") #PCA functions
library(ade4) #PCA Visualizations
```

```
##
## Attaching package: 'ade4'
```

```
## The following object is masked from 'package:FactoMineR':
##
##     reconst
```

```
library(readxl)
#################################################################################################
```

```
#Get File
setwd("C:/Users/ateboul/Desktop")

#Read in Datasets
nutrition_data <- read_excel("Nutrition_v2.xlsx")

#Check Sample Size and Number of Variables
dim(nutrition_data)
```

```
## [1] 8618   30
```

```
#Missing values
sum(is.na(nutrition_data))
```

```
## [1] 22499
```

- While we do have 22499 missing values in the dataset, they're really mostly in the name columns which we won't use anyways.

```
#Show for first 6 rows of data
head(nutrition_data)
```

```
## # A tibble: 6 x 30
##      ID FoodGroup ShortDescrip Descrip CommonName MfgName ScientificName
##   <dbl> <chr>     <chr>        <chr>   <chr>      <chr>   <chr>
## 1  1001 Dairy an~ BUTTER,WITH~ Butter~ <NA>       <NA>    <NA>
## 2  1002 Dairy an~ BUTTER,WHIP~ Butter~ <NA>       <NA>    <NA>
## 3  1003 Dairy an~ BUTTER OIL,~ Butter~ <NA>       <NA>    <NA>
## 4  1004 Dairy an~ CHEESE,BLUE  Cheese~ <NA>       <NA>    <NA>
## 5  1005 Dairy an~ CHEESE,BRICK Cheese~ <NA>       <NA>    <NA>
## 6  1006 Dairy an~ CHEESE,BRIE  Cheese~ <NA>       <NA>    <NA>
## # ... with 23 more variables: Energy_kcal <dbl>, Protein_g <dbl>,
## #   Fat_g <dbl>, Carb_g <dbl>, Sugar_g <dbl>, Fiber_g <dbl>,
## #   VitA_mcg <dbl>, VitB6_mg <dbl>, VitB12_mcg <dbl>, VitC_mg <dbl>,
## #   VitE_mg <dbl>, Folate_mcg <dbl>, Niacin_mg <dbl>, Riboflavin_mg <dbl>,
## #   Thiamin_mg <dbl>, Calcium_mg <dbl>, Copper_mcg <dbl>, Iron_mg <dbl>,
## #   Magnesium_mg <dbl>, Manganese_mg <dbl>, Phosphorus_mg <dbl>,
## #   Selenium_mcg <dbl>, Zinc_mg <dbl>
```

```
#Column Names
names(nutrition_data)
```

```
##  [1] "ID"             "FoodGroup"      "ShortDescrip"   "Descrip"
##  [5] "CommonName"     "MfgName"        "ScientificName" "Energy_kcal"
##  [9] "Protein_g"      "Fat_g"          "Carb_g"         "Sugar_g"
## [13] "Fiber_g"        "VitA_mcg"       "VitB6_mg"       "VitB12_mcg"
## [17] "VitC_mg"        "VitE_mg"        "Folate_mcg"     "Niacin_mg"
## [21] "Riboflavin_mg"  "Thiamin_mg"     "Calcium_mg"     "Copper_mcg"
## [25] "Iron_mg"        "Magnesium_mg"   "Manganese_mg"   "Phosphorus_mg"
## [29] "Selenium_mcg"   "Zinc_mg"
```

```
#how many food groups are in the datatset?
unique(nutrition_data$FoodGroup)
```

```
##  [1] "Dairy and Egg Products"
##  [2] "Spices and Herbs"
##  [3] "Baby Foods"
##  [4] "Fats and Oils"
##  [5] "Poultry Products"
##  [6] "Soups, Sauces, and Gravies"
##  [7] "Sausages and Luncheon Meats"
##  [8] "Breakfast Cereals"
##  [9] "Snacks"
## [10] "Fruits and Fruit Juices"
## [11] "Pork Products"
## [12] "Vegetables and Vegetable Products"
## [13] "Nut and Seed Products"
## [14] "Beef Products"
## [15] "Beverages"
## [16] "Finfish and Shellfish Products"
## [17] "Legumes and Legume Products"
## [18] "Lamb, Veal, and Game Products"
## [19] "Baked Products"
## [20] "Sweets"
## [21] "Cereal Grains and Pasta"
## [22] "Fast Foods"
## [23] "Meals, Entrees, and Side Dishes"
## [24] "American Indian/Alaska Native Foods"
## [25] "Restaurant Foods"
```

```
#Get a dataframe with just numeric variables for use in PCA.

#Show Structure of Dataset
str(nutrition_data, list.len=ncol(nutrition_data))
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    8618 obs. of  30 variables:
##  $ ID            : num  1001 1002 1003 1004 1005 ...
##  $ FoodGroup     : chr  "Dairy and Egg Products" "Dairy and Egg Products" "Dairy and Egg Prod
ucts" "Dairy and Egg Products" ...
##  $ ShortDescrip  : chr  "BUTTER,WITH SALT" "BUTTER,WHIPPED,WITH SALT" "BUTTER OIL,ANHYDROUS"
"CHEESE,BLUE" ...
##  $ Descrip       : chr  "Butter, salted" "Butter, whipped, with salt" "Butter oil, anhydrous"
"Cheese, blue" ...
##  $ CommonName    : chr  NA NA NA NA ...
##  $ MfgName       : chr  NA NA NA NA ...
##  $ ScientificName: chr  NA NA NA NA ...
##  $ Energy_kcal   : num  717 717 876 353 371 334 300 376 406 387 ...
##  $ Protein_g     : num  0.85 0.85 0.28 21.4 23.24 ...
##  $ Fat_g         : num  81.1 81.1 99.5 28.7 29.7 ...
##  $ Carb_g        : num  0.06 0.06 0 2.34 2.79 0.45 0.46 3.06 1.33 4.78 ...
##  $ Sugar_g       : num  0.06 0.06 0 0.5 0.51 0.45 0.46 0 0.28 0 ...
##  $ Fiber_g       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VitA_mcg      : num  684 684 840 198 292 174 241 271 263 233 ...
##  $ VitB6_mg      : num  0.003 0.003 0.001 0.166 0.065 0.235 0.227 0.074 0.049 0.074 ...
##  $ VitB12_mcg    : num  0.17 0.13 0.01 1.22 1.26 1.65 1.3 0.27 0.88 0.83 ...
##  $ VitC_mg       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VitE_mg       : num  2.32 2.32 2.8 0.25 0.26 0.24 0.21 0 0.78 0 ...
##  $ Folate_mcg    : num  3 3 0 36 20 65 62 18 26 18 ...
##  $ Niacin_mg     : num  0.042 0.042 0.003 1.016 0.118 ...
##  $ Riboflavin_mg : num  0.034 0.034 0.005 0.382 0.351 0.52 0.488 0.45 0.434 0.293 ...
##  $ Thiamin_mg    : num  0.005 0.005 0.001 0.029 0.014 0.07 0.028 0.031 0.027 0.046 ...
##  $ Calcium_mg    : num  24 24 4 528 674 184 388 673 675 643 ...
##  $ Copper_mcg    : num  0 0.016 0.001 0.04 0.024 0.019 0.021 0.024 0.056 0.042 ...
##  $ Iron_mg       : num  0.02 0.16 0 0.31 0.43 0.5 0.33 0.64 0.16 0.21 ...
##  $ Magnesium_mg  : num  2 2 0 23 24 20 20 22 27 21 ...
##  $ Manganese_mg  : num  0 0.004 0 0.009 0.012 0.034 0.038 0.021 0.033 0.012 ...
##  $ Phosphorus_mg : num  24 23 3 387 451 188 347 490 473 464 ...
##  $ Selenium_mcg  : num  1 1 0 14.5 14.5 14.5 14.5 14.5 28.3 14.5 ...
##  $ Zinc_mg       : num  0.09 0.05 0.01 2.66 2.6 2.38 2.38 2.94 3.43 2.79 ...
```

```
#Capture numeric variables
nutrition_data2 <- nutrition_data[,c(8:30)]
head(nutrition_data2)
```

```
## # A tibble: 6 x 23
##   Energy_kcal Protein_g Fat_g Carb_g Sugar_g Fiber_g VitA_mcg VitB6_mg
##         <dbl>     <dbl> <dbl>  <dbl>   <dbl>   <dbl>    <dbl>    <dbl>
## 1         717      0.85  81.1   0.06    0.06       0      684    0.003
## 2         717      0.85  81.1   0.06    0.06       0      684    0.003
## 3         876      0.28  99.5   0       0          0      840    0.001
## 4         353     21.4   28.7   2.34    0.5        0      198    0.166
## 5         371     23.2   29.7   2.79    0.51       0      292    0.065
## 6         334     20.8   27.7   0.45    0.45       0      174    0.235
## # ... with 15 more variables: VitB12_mcg <dbl>, VitC_mg <dbl>,
## #   VitE_mg <dbl>, Folate_mcg <dbl>, Niacin_mg <dbl>, Riboflavin_mg <dbl>,
## #   Thiamin_mg <dbl>, Calcium_mg <dbl>, Copper_mcg <dbl>, Iron_mg <dbl>,
## #   Magnesium_mg <dbl>, Manganese_mg <dbl>, Phosphorus_mg <dbl>,
## #   Selenium_mcg <dbl>, Zinc_mg <dbl>
```

```
#Check new data has no missing data
sum(is.na(nutrition_data2))
```

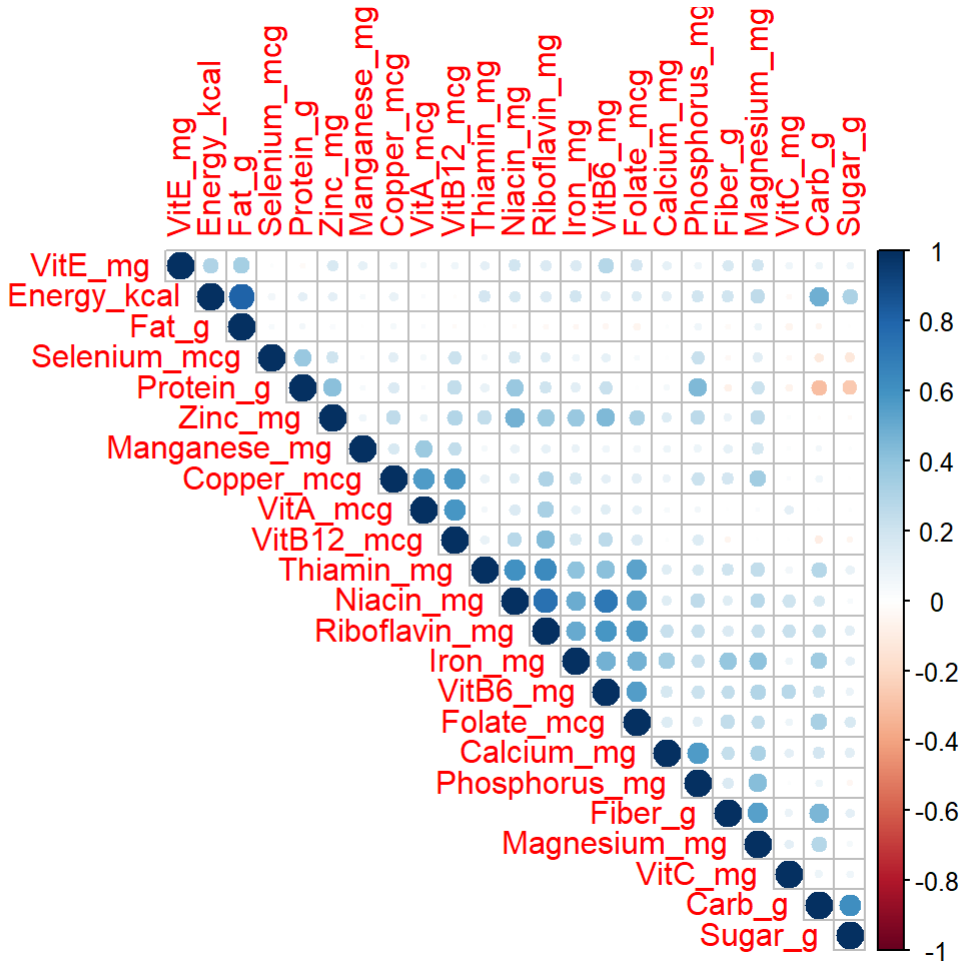
```
## [1] 0
```

- Because the variables are not in the same units, they must be scaled. Divide by stdev with scale = T later on.
- Also have confirmed that there are no missing data points, so we can proceed.

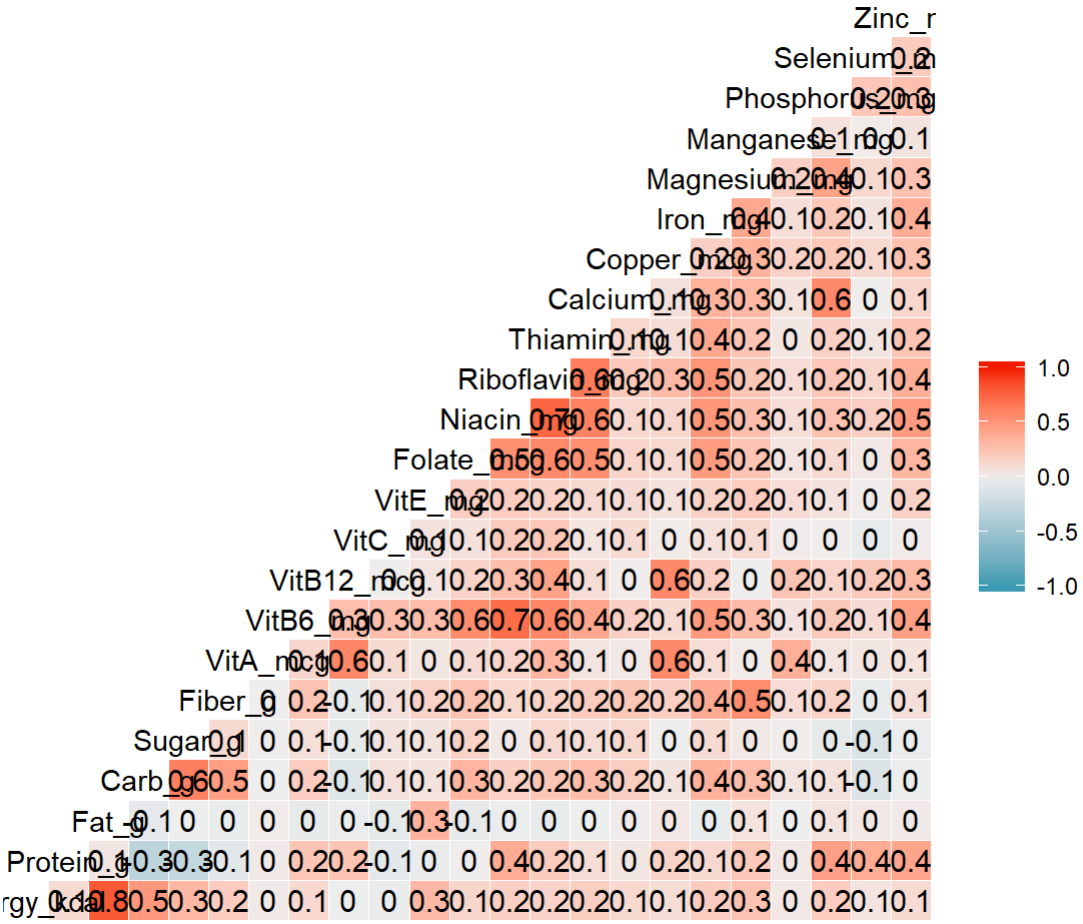
```
#Describe the data
describe(nutrition_data2)
```

```
##               vars    n   mean     sd median trimmed    mad min      max
## Energy_kcal      1 8618 226.44 169.39 191.00  207.70 167.53   0   902.00
## Protein_g        2 8618  11.52  10.55   8.29   10.44  10.44   0    88.32
## Fat_g            3 8618  10.65  15.87   5.24    7.40   7.23   0   100.00
## Carb_g           4 8618  21.82  27.24   8.95   17.31  13.26   0   100.00
## Sugar_g          5 8618   6.56  13.60   0.37    2.94   0.55   0    99.80
## Fiber_g          6 8618   2.02   4.31   0.30    1.07   0.44   0    79.00
## VitA_mcg         7 8618  93.97 779.36   1.50   12.24   2.22   0 30000.00
## VitB6_mg         8 8618   0.26   0.48   0.12    0.18   0.18   0    12.00
## VitB12_mcg       9 8618   1.23   4.32   0.08    0.54   0.12   0    98.89
## VitC_mg         10 8618   7.93  57.58   0.00    1.78   0.00   0  2400.00
## VitE_mg         11 8618   0.87   3.85   0.11    0.23   0.16   0   149.40
## Folate_mcg      12 8618  50.31 186.56   7.00   14.98  10.38   0  5881.00
## Niacin_mg       13 8618   3.41   4.83   2.10    2.62   2.94   0   127.50
## Riboflavin_mg   14 8618   0.24   0.45   0.15    0.16   0.16   0    17.50
## Thiamin_mg      15 8618   0.21   0.52   0.08    0.12   0.09   0    23.38
## Calcium_mg      16 8618  73.41 201.36  19.00   35.90  20.76   0  7364.00
## Copper_mcg      17 8618   0.17   0.55   0.08    0.09   0.08   0    15.05
## Iron_mg         18 8618   2.70   5.73   1.33    1.57   1.41   0   123.60
## Magnesium_mg    19 8618  32.75  56.07  20.00   21.19  14.83   0   781.00
## Manganese_mg    20 8618   0.50   6.38   0.02    0.11   0.03   0   328.00
## Phosphorus_mg   21 8618 155.99 203.09 133.00  132.04 131.95   0  9918.00
## Selenium_mcg    22 8618  12.61  28.29   3.90    9.25   5.78   0  1917.00
## Zinc_mg         23 8618   1.97   3.36   0.84    1.39   1.18   0    90.95
##                  range  skew kurtosis   se
## Energy_kcal     902.00  1.17     1.79 1.82
## Protein_g        88.32  1.17     2.74 0.11
## Fat_g           100.00  3.31    13.72 0.17
## Carb_g          100.00  1.16    -0.09 0.29
## Sugar_g          99.80  2.95     9.66 0.15
## Fiber_g          79.00  5.79    56.36 0.05
## VitA_mcg      30000.00 24.08   716.76 8.40
## VitB6_mg         12.00  7.55   101.60 0.01
## VitB12_mcg       98.89 13.03   221.53 0.05
## VitC_mg        2400.00 30.90  1114.39 0.62
## VitE_mg         149.40 13.84   337.83 0.04
## Folate_mcg     5881.00 10.32   176.63 2.01
## Niacin_mg       127.50  6.43    96.39 0.05
## Riboflavin_mg    17.50 11.31   292.70 0.00
## Thiamin_mg       23.38 17.50   576.45 0.01
## Calcium_mg     7364.00 13.42   339.07 2.17
## Copper_mcg       15.05 15.94   344.97 0.01
## Iron_mg         123.60  6.84    70.17 0.06
## Magnesium_mg    781.00  5.56    43.99 0.60
## Manganese_mg    328.00 39.73  1784.81 0.07
## Phosphorus_mg  9918.00 18.55   763.67 2.19
## Selenium_mcg   1917.00 38.19  2427.75 0.30
## Zinc_mg          90.95  9.63   185.80 0.04
```

```
#Check Correlations
nut_cor_mat<-cor(nutrition_data2)
#nut_cor_mat
corrplot(nut_cor_mat, type = "upper", order = "hclust")
```




```
ggcorr(nutrition_data2, label=TRUE)
```



- There are some weak but interesting correlations between the variables. For example energy and fat, vitamin B12 and vitamin A, thiamin niacin riboflavin iron vitB6 and folate, calcium and phosphorous, fiber and magnesium, and carbs and sugar. Let's see if PCA turns these into components that will allow us to reduce the data.

```
#Test KMO Sampling Adequacy
#library(psych)
KMO(nutrition_data2)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = nutrition_data2)
## Overall MSA =  0.64
## MSA for each item =
##   Energy_kcal     Protein_g         Fat_g        Carb_g       Sugar_g
##          0.34          0.30          0.22          0.35          0.86
##       Fiber_g      VitA_mcg      VitB6_mg    VitB12_mcg       VitC_mg
##          0.66          0.70          0.86          0.76          0.61
##       VitE_mg    Folate_mcg     Niacin_mg Riboflavin_mg    Thiamin_mg
##          0.90          0.92          0.86          0.86          0.87
##    Calcium_mg    Copper_mcg       Iron_mg  Magnesium_mg  Manganese_mg
##          0.63          0.68          0.91          0.77          0.59
## Phosphorus_mg  Selenium_mcg       Zinc_mg
##          0.71          0.92          0.91
```

```
#Overall MSA =  0.64
#This is >=0.5 or 0.6 - good

#Test Bartlett's Test of Sphericity
#library(REdaS)
bart_spher(nutrition_data2)
```

```
##  Bartlett's Test of Sphericity
##
## Call: bart_spher(x = nutrition_data2)
##
##      X2 = 120430.831
##      df = 253
## p-value < 2.22e-16
```

```
#p-value < 2.22e-16 (Very Small Number)
#This is significant

#Test for Reliability Analysis using Cronbach's Alpha
#library(psych)
alpha(nutrition_data2, check.keys=TRUE)
```

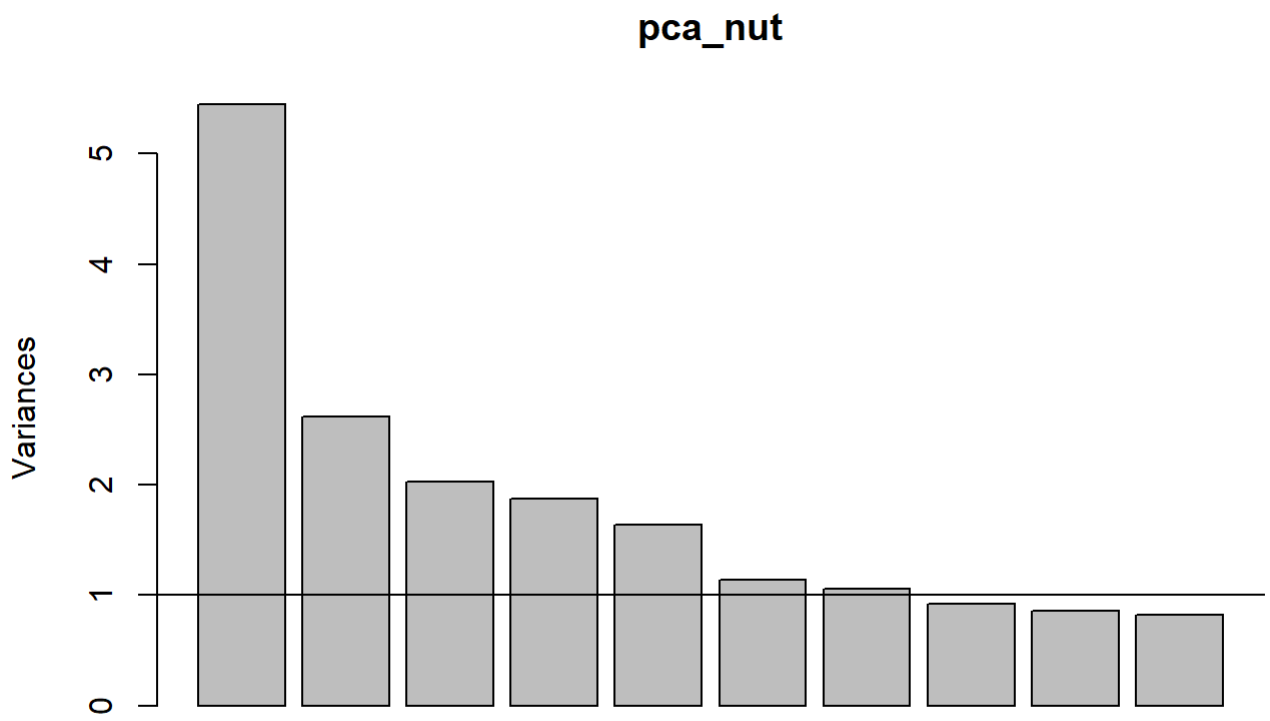
```
##
## Reliability analysis
## Call: alpha(x = nutrition_data2, check.keys = TRUE)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean sd median_r
##       0.27      0.83    0.91      0.17 4.7 0.0083   31 44     0.13
##
##  lower alpha upper     95% confidence boundaries
## 0.25 0.27 0.28
##
##  Reliability if an item is dropped:
##               raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r
## Energy_kcal        0.23      0.82    0.88      0.17 4.5   0.0080 0.032
## Protein_g          0.27      0.82    0.90      0.18 4.7   0.0083 0.031
## Fat_g              0.26      0.83    0.89      0.18 4.9   0.0083 0.030
## Carb_g             0.26      0.82    0.89      0.17 4.6   0.0083 0.030
## Sugar_g            0.27      0.83    0.91      0.18 4.8   0.0083 0.031
## Fiber_g            0.27      0.82    0.90      0.17 4.5   0.0083 0.032
## VitA_mcg           0.55      0.82    0.90      0.17 4.6   0.0053 0.032
## VitB6_mg           0.27      0.81    0.90      0.16 4.2   0.0083 0.030
## VitB12_mcg         0.26      0.82    0.90      0.17 4.5   0.0083 0.031
## VitC_mg            0.26      0.83    0.91      0.18 4.8   0.0084 0.033
## VitE_mg            0.27      0.82    0.91      0.17 4.6   0.0083 0.034
## Folate_mcg         0.21      0.81    0.90      0.16 4.3   0.0083 0.031
## Niacin_mg          0.26      0.81    0.90      0.16 4.2   0.0083 0.029
## Riboflavin_mg      0.27      0.80    0.89      0.16 4.1   0.0083 0.029
## Thiamin_mg         0.27      0.81    0.90      0.17 4.4   0.0083 0.031
## Calcium_mg         0.20      0.82    0.90      0.17 4.6   0.0078 0.033
## Copper_mcg         0.27      0.82    0.90      0.17 4.5   0.0083 0.033
## Iron_mg            0.26      0.81    0.90      0.16 4.2   0.0083 0.031
## Magnesium_mg       0.24      0.81    0.90      0.16 4.3   0.0083 0.033
## Manganese_mg       0.26      0.83    0.91      0.18 4.8   0.0083 0.033
## Phosphorus_mg      0.19      0.82    0.90      0.17 4.5   0.0079 0.033
## Selenium_mcg       0.26      0.83    0.91      0.18 4.8   0.0083 0.033
## Zinc_mg            0.27      0.81    0.90      0.17 4.4   0.0083 0.032
##               med.r
## Energy_kcal    0.13
## Protein_g      0.14
## Fat_g          0.15
## Carb_g         0.13
## Sugar_g        0.15
## Fiber_g        0.13
## VitA_mcg       0.14
## VitB6_mg       0.12
## VitB12_mcg     0.13
## VitC_mg        0.15
## VitE_mg        0.13
## Folate_mcg     0.13
## Niacin_mg      0.12
## Riboflavin_mg  0.12
## Thiamin_mg     0.13
## Calcium_mg     0.13
## Copper_mcg     0.13
```

```
## Iron_mg        0.12
## Magnesium_mg   0.12
## Manganese_mg   0.14
## Phosphorus_mg  0.13
## Selenium_mcg   0.14
## Zinc_mg        0.13
##
##  Item statistics
##                   n raw.r std.r r.cor r.drop   mean     sd
## Energy_kcal    8618  0.33  0.48  0.51   0.17 226.44 169.39
## Protein_g      8618  0.16  0.33  0.34   0.15  11.52  10.55
## Fat_g          8618  0.18  0.20  0.21   0.16  10.65  15.87
## Carb_g         8618  0.25  0.42  0.44   0.22  21.82  27.24
## Sugar_g        8618  0.14  0.24  0.19   0.12   6.56  13.60
## Fiber_g        8618  0.22  0.43  0.40   0.21   2.02   4.31
## VitA_mcg       8618  0.83  0.37  0.34   0.11  93.97 779.36
## VitB6_mg       8618  0.36  0.67  0.66   0.36   0.26   0.48
## VitB12_mcg     8618  0.52  0.44  0.42   0.52   1.23   4.32
## VitC_mg        8618  0.18  0.23  0.16   0.13   7.93  57.58
## VitE_mg        8618  0.19  0.37  0.32   0.18   0.87   3.85
## Folate_mcg     8618  0.39  0.58  0.56   0.21  50.31 186.56
## Niacin_mg      8618  0.38  0.70  0.71   0.37   3.41   4.83
## Riboflavin_mg 8618  0.52  0.73  0.74   0.52   0.24   0.45
## Thiamin_mg     8618  0.27  0.55  0.53   0.27   0.21   0.52
## Calcium_mg     8618  0.43  0.42  0.38   0.24  73.41 201.36
## Copper_mcg     8618  0.57  0.47  0.45   0.57   0.17   0.55
## Iron_mg        8618  0.37  0.64  0.62   0.36   2.70   5.73
## Magnesium_mg   8618  0.34  0.58  0.57   0.29  32.75  56.07
## Manganese_mg   8618  0.36  0.29  0.23   0.35   0.50   6.38
## Phosphorus_mg 8618  0.46  0.49  0.48   0.28 155.99 203.09
## Selenium_mcg   8618  0.14  0.26  0.19   0.11  12.61  28.29
## Zinc_mg        8618  0.25  0.53  0.50   0.25   1.97   3.36
```

```
#raw_alpha = 0.27
#This should be > 0.7 but it is low. It's only exploratory so maybe okay, also scales were not a
djusted.
```

A) How many components are need to explain 100% of total variation for this data? How many components are determined from the scree plot? What number of components would you use in the model?

```
#Create PCA
pca_nut = prcomp(nutrition_data2, center=T, scale=T)

#Check Scree Plot
plot(pca_nut)
abline(1, 0)
```




```
#Check PCA Summary Information
summary(pca_nut)
```

```
## Importance of components:
##                           PC1    PC2     PC3    PC4     PC5     PC6
## Standard deviation     2.3344 1.6182 1.42545 1.3708 1.27893 1.06788
## Proportion of Variance 0.2369 0.1139 0.08834 0.0817 0.07112 0.04958
## Cumulative Proportion  0.2369 0.3508 0.43911 0.5208 0.59193 0.64151
##                           PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     1.02998 0.96242 0.92845 0.90812 0.85512 0.77259
## Proportion of Variance 0.04612 0.04027 0.03748 0.03586 0.03179 0.02595
## Cumulative Proportion  0.68764 0.72791 0.76539 0.80125 0.83304 0.85899
##                           PC13    PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.71320 0.68517 0.63874 0.58132 0.57408 0.56678
## Proportion of Variance 0.02212 0.02041 0.01774 0.01469 0.01433 0.01397
## Cumulative Proportion  0.88111 0.90152 0.91926 0.93395 0.94828 0.96224
##                           PC19    PC20    PC21    PC22    PC23
## Standard deviation     0.50616 0.48800 0.45952 0.39887 0.06161
## Proportion of Variance 0.01114 0.01035 0.00918 0.00692 0.00017
## Cumulative Proportion  0.97338 0.98374 0.99292 0.99983 1.00000
```

- In order to explain 100% of the variation in the data we would need all 23 variables, but that defeats the whole purpose of doing PCA in the first place. So, based on the Scree Plot, we have some options for picking a components number. With eigenvalue >= 1 or the “knee” method we would keep between 5 and 7

components, and then depending on some sort of expert guidance make a better decision. Choosing 5 components gives 59% of variance explained and choosing 7 components gives about 69% variance explained. Whether that additional 10% matters for this particular dataset is unclear.

- I will go with **6 components** for 64% of the variance which is at least above 60%, and it is also the number of interesting groupings I had in my corrplot.

B) For the number of components in part A, give the formula for each component and a brief interpretation after rotating the components. What names might you give for each of the components?

```
#Using the psych package for next part as instructed - with 6 components
pca_nut2 = psych::principal(nutrition_data2, rotate="varimax", nfactors=6, scores=TRUE)
pca_nut2
```

```
## Principal Components Analysis
## Call: psych::principal(r = nutrition_data2, nfactors = 6, rotate = "varimax",
##     scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                 RC1   RC5   RC4   RC2   RC3   RC6   h2    u2 com
## Energy_kcal    0.12  0.16  0.01  0.38  0.84  0.21 0.93 0.069 1.7
## Protein_g      0.27  0.20  0.06 -0.51  0.09  0.57 0.71 0.286 2.8
## Fat_g         -0.10 -0.04  0.01 -0.01  0.93  0.11 0.89 0.111 1.1
## Carb_g         0.22  0.29 -0.02  0.83  0.08 -0.08 0.83 0.166 1.4
## Sugar_g        0.10  0.01  0.00  0.76  0.08 -0.01 0.60 0.403 1.1
## Fiber_g        0.14  0.63  0.04  0.30  0.02 -0.23 0.56 0.439 1.9
## VitA_mcg       0.09 -0.07  0.87  0.03  0.01 -0.06 0.78 0.225 1.0
## VitB6_mg       0.80  0.17  0.07 -0.09  0.07 -0.16 0.71 0.294 1.2
## VitB12_mcg     0.28 -0.11  0.77 -0.12 -0.03  0.22 0.75 0.248 1.5
## VitC_mg        0.23  0.13  0.07 -0.17  0.00 -0.58 0.44 0.556 1.6
## VitE_mg        0.22  0.12  0.06 -0.12  0.60 -0.35 0.57 0.429 2.2
## Folate_mcg     0.74  0.08  0.03  0.23 -0.01 -0.07 0.61 0.391 1.3
## Niacin_mg      0.88  0.09  0.08 -0.10  0.07  0.05 0.81 0.190 1.1
## Riboflavin_mg  0.82  0.06  0.30  0.08  0.00  0.03 0.77 0.230 1.3
## Thiamin_mg     0.72  0.07 -0.03  0.20  0.02  0.06 0.57 0.430 1.2
## Calcium_mg     0.10  0.69  0.01  0.05 -0.01  0.01 0.49 0.512 1.1
## Copper_mcg     0.08  0.21  0.77  0.03  0.03  0.16 0.68 0.320 1.3
## Iron_mg        0.59  0.41  0.08  0.18 -0.01  0.00 0.55 0.448 2.0
## Magnesium_mg   0.20  0.77  0.11  0.03  0.14 -0.01 0.66 0.337 1.3
## Manganese_mg  -0.01  0.16  0.52 -0.02  0.05 -0.17 0.33 0.675 1.5
## Phosphorus_mg  0.14  0.69  0.08 -0.18  0.07  0.37 0.67 0.326 1.9
## Selenium_mcg   0.16  0.09  0.09 -0.25  0.05  0.51 0.37 0.632 1.9
## Zinc_mg        0.54  0.18  0.12 -0.23  0.09  0.27 0.47 0.529 2.4
##
##                        RC1  RC5  RC4  RC2  RC3  RC6
## SS loadings           4.24 2.47 2.38 2.13 2.00 1.52
## Proportion Var        0.18 0.11 0.10 0.09 0.09 0.07
## Cumulative Var        0.18 0.29 0.40 0.49 0.58 0.64
## Proportion Explained  0.29 0.17 0.16 0.14 0.14 0.10
## Cumulative Proportion 0.29 0.46 0.62 0.76 0.90 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 6 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.06
##  with the empirical chi square  16728.75  with prob <  0
##
## Fit based upon off diagonal values = 0.94
```

```
print(pca_nut2$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##               RC1    RC5    RC4    RC2    RC3    RC6
## VitB6_mg       0.795
## Folate_mcg     0.736
## Niacin_mg      0.883
## Riboflavin_mg  0.819
## Thiamin_mg     0.722
## Iron_mg        0.589  0.407
## Zinc_mg        0.537
## Fiber_g               0.631
## Calcium_mg            0.689
## Magnesium_mg          0.769
## Phosphorus_mg         0.686
## VitA_mg                      0.870
## VitB12_mg                    0.773
## Copper_mg                    0.775
## Manganese_mg                 0.516
## Carb_g                              0.829
## Sugar_g                             0.762
## Energy_kcal                                0.839
## Fat_g                                      0.931
## VitE_mg                                    0.604
## Protein_g                          -0.515         0.570
## VitC_mg                                           -0.584
## Selenium_mcg                                       0.510
##
##                  RC1   RC5   RC4   RC2   RC3   RC6
## SS loadings    4.243 2.474 2.385 2.127 2.005 1.521
## Proportion Var 0.184 0.108 0.104 0.092 0.087 0.066
## Cumulative Var 0.184 0.292 0.396 0.488 0.575 0.642
```

- While I do not know much about the nutrient compositions of different food groups:
- RC1 - Nutrient diverse (beans, grains, and veggies) explains some general nutrients you need
- RC2 - Fibrous and trace minerals (nuts and legumes) explains some of those trace minerals you need and some healthy fiber sources
- RC3 - Red Meats or vitaminA,B12,copper, magnesium (meats) explains some of the nutrients from maybe meat I’m not sure
- RC4 - Sugars (sugary foods) lots of carbs and sugar interestingly negative on the protein which makes sense, you rarely get both together
- RC5 - Fats (Fatty High Energy foods) high fat and high energy
- RC6 - More assorted meats and fruits (probably too many components in this model)

```
#PCAs Other Available Information
#ls(pca_nut2)

#pca_nut2$values
#pca_nut2$communality
#pca_nut2$rot.mat
```

C) What subjects have the highest and lowest values for each principal component (only include the number of components specified in part A. For each of those subjects, give the principal component scores (again only for the number of components specified in part A).

```
#Calculating scores
scores <- pca_nut2$scores
scores_1 <- scores[,1]
scores_2 <- scores[,2]
scores_3 <- scores[,3]
scores_4 <- scores[,4]
scores_5 <- scores[,5]
scores_6 <- scores[,6]

min_score1 <- min(scores_1)
min_score1
```

```
## [1] -5.904182
```

```
max_score1 <- max(scores_1)
max_score1
```

```
## [1] 29.89839
```

```
#Calculating scores
min_score2 <- min(scores_2)
min_score2
```

```
## [1] -10.28396
```

```
max_score2 <- max(scores_2)
max_score2
```

```
## [1] 27.41177
```

```
#Calculating scores
min_score3 <- min(scores_3)
min_score3
```

```
## [1] -4.07404
```

```
max_score3 <- max(scores_3)
max_score3
```

```
## [1] 35.40282
```

```
#Calculating scores
min_score4 <- min(scores_4)
min_score4
```

```
## [1] -10.24372
```

```
max_score4 <- max(scores_4)
max_score4
```

```
## [1] 6.214694
```

```
#Calculating scores
min_score5 <- min(scores_5)
min_score5
```

```
## [1] -4.232759
```

```
max_score5 <- max(scores_5)
max_score5
```

```
## [1] 17.53254
```

```
#Calculating scores
min_score6 <- min(scores_6)
min_score6
```

```
## [1] -23.07518
```

```
max_score6 <- max(scores_6)
max_score6
```

```
## [1] 21.70471
```

D) Finally, run a common factor analysis on the same data. What difference, if any, do you find? Does the factor analysis change your ability to interpret the results practically?

```
#Factor Analysis
fit = factanal(nutrition_data2, 6)
print(fit$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##               Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
## VitB6_mg       0.730
## Folate_mcg     0.664
## Niacin_mg      0.845
## Riboflavin_mg  0.794
## Thiamin_mg     0.669
## Iron_mg        0.536
## VitA_mcg               0.760
## VitB12_mcg             0.751
## Copper_mcg             0.756
## Energy_kcal                    0.839                   0.485
## Fat_g                          0.989
## Fiber_g                                0.666
## Magnesium_mg                           0.782
## Protein_g                                      0.977
## Carb_g                                                 0.905
## Sugar_g                                                0.613
## VitC_mg
## VitE_mg
## Calcium_mg
## Manganese_mg
## Phosphorus_mg                                  0.431
## Selenium_mcg
## Zinc_mg
##
##                Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
## SS loadings      3.561   1.994   1.858   1.844   1.785   1.609
## Proportion Var   0.155   0.087   0.081   0.080   0.078   0.070
## Cumulative Var   0.155   0.242   0.322   0.402   0.480   0.550
```

```
summary(fit)
```

```
##              Length Class    Mode
## converged      1    -none-   logical
## loadings     138    loadings numeric
## uniquenesses  23    -none-   numeric
## correlation  529    -none-   numeric
## criteria       3    -none-   numeric
## factors        1    -none-   numeric
## dof            1    -none-   numeric
## method         1    -none-   character
## rotmat        36    -none-   numeric
## STATISTIC      1    -none-   numeric
## PVAL           1    -none-   numeric
## n.obs          1    -none-   numeric
## call           3    -none-   call
```

- The factor analysis is slightly different. An important difference is that here Factor6 exposes how high-carb high-sugar food is related to also more energy just like fatty foods. So the foods with all the vitB6 and folate are probably not super high energy as well which is an interesting distinction. Also vitamin C, E, Calcium, magnesium, and zinc are empty given the cutoff. Maybe fewer factors would capture the data better here. Practically, an important concept could be that you should not combine highly sugary and highly fatty foods because both give lots of energy, which if you don't use could probably lead to greater fat accumulation in the body.

Problem 6

Problem 6: (Principal Component Analysis - 20 points) Begin with the "census2.csv" datafile, which contains census data on various tracts in a district. The fields in the data are: * Total Population (thousands) * Professional degree (percent) * Employed age over 16 (percent) * Government employed (percent) * Median home value (dollars)

```
#Read in Datasets
census_data <- read.csv("Census2.csv")

#Check Sample Size and Number of Variables
dim(census_data)
```

```
## [1] 61  5
```

```
#Missing values
sum(is.na(census_data))
```

```
## [1] 0
```

```
#Show for first 6 rows of data
head(census_data)
```

```
##   Population Professional Employed Government MedianHomeVal
## 1       2.67         5.71    69.02       30.3        148000
## 2       2.25         4.37    72.98       43.3        144000
## 3       3.12        10.27    64.94       32.0        211000
## 4       5.14         7.44    71.29       24.5        185000
## 5       5.54         9.25    74.94       31.0        223000
## 6       5.04         4.84    53.61       48.2        160000
```

```
#Column Names
names(census_data)
```

```
## [1] "Population"    "Professional"  "Employed"      "Government"
## [5] "MedianHomeVal"
```

```
#Describe the data
describe(census_data)
```

```
##               vars  n      mean       sd   median   trimmed      mad
## Population       1 61      4.47     1.84 4.72e+00      4.41     2.16
## Professional     2 61      3.96     3.11 3.38e+00      3.42     2.30
## Employed         3 61     71.42     7.46 7.13e+01     71.64     7.50
## Government       4 61     26.91     9.44 2.44e+01     25.56     6.38
## MedianHomeVal    5 61 163557.38 56446.88 1.49e+05 154653.06 38547.60
##                    min       max     range  skew kurtosis      se
## Population        1.36      9.21      7.85  0.24    -0.82    0.24
## Professional      0.72     16.70     15.98  1.94     4.56    0.40
## Employed         49.50     86.54     37.04 -0.38     0.19    0.95
## Government       16.30     68.50     52.20  1.85     4.72    1.21
## MedianHomeVal 93000.00 364000.00 271000.00  1.56     2.34 7227.28
```

```
#Check Correlations
census_cor_mat<-cor(census_data)
#corrplot(census_cor_mat, type = "lower")

#ggcorr(census_data, label=TRUE)
```

```
#Test KMO Sampling Adequacy
#library(psych)
KMO(census_data)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = census_data)
## Overall MSA =  0.47
## MSA for each item =
##    Population  Professional      Employed    Government MedianHomeVal
##          0.37          0.47          0.46          0.51          0.47
```

```
#Overall MSA =  0.47
#This is NOT >=0.5 or 0.6 - NOT good

#Test Bartlett's Test of Sphericity
#library(REdaS)
bart_spher(census_data)
```

```
##  Bartlett's Test of Sphericity
##
## Call: bart_spher(x = census_data)
##
##      X2 = 68.473
##      df = 10
## p-value < 2.22e-16
```

```
#p-value < 2.22e-16 (Very Small Number)
#This is significant

#Test for Reliability Analysis using Cronbach's Alpha
#library(psych)
alpha(census_data, check.keys=TRUE)
```

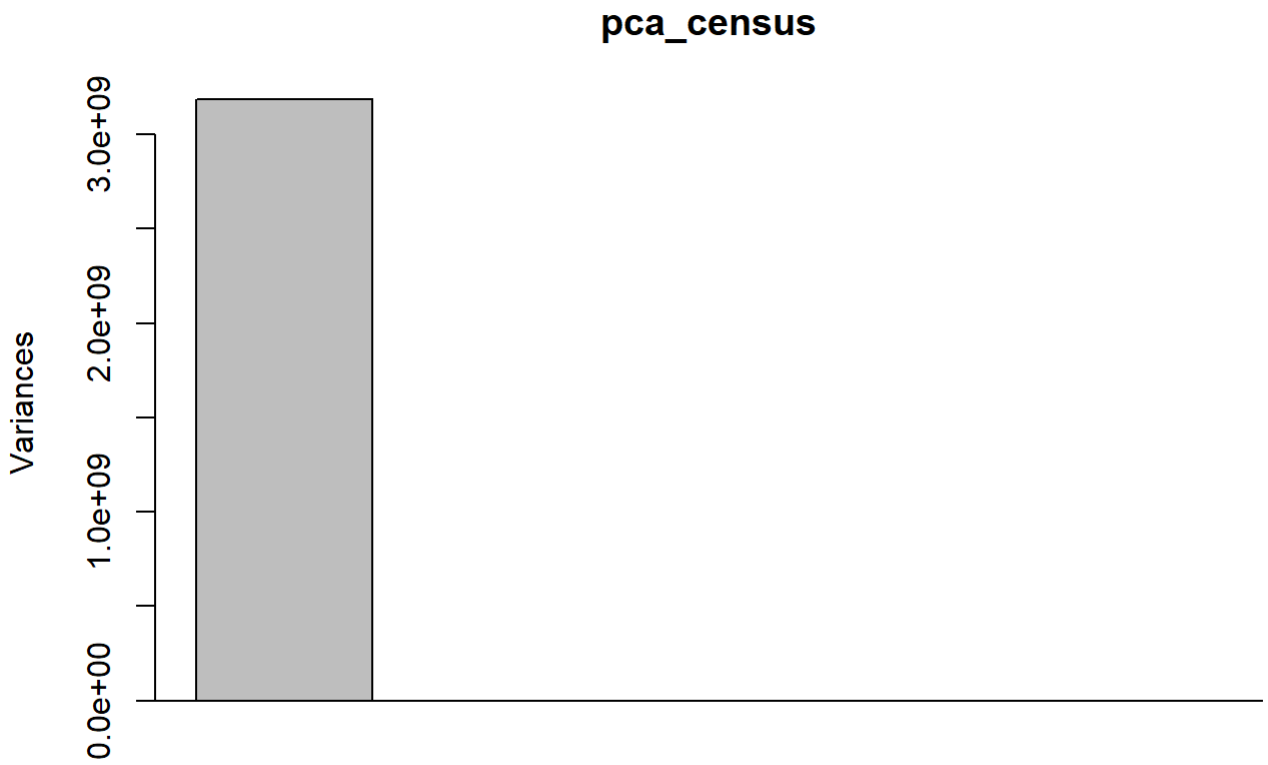
```
## Warning in alpha(census_data, check.keys = TRUE): Some items were negatively correlated with
total scale and were automatically reversed.
##  This is indicated by a negative sign for the variable name.
```

```
##
## Reliability analysis
## Call: alpha(x = census_data, check.keys = TRUE)
##
##   raw_alpha std.alpha G6(smc) average_r S/N     ase   mean    sd median_r
##    0.00017       0.6    0.67      0.23 1.5 8.8e-05 178303 11290     0.19
##
##  lower alpha upper     95% confidence boundaries
## 0 0 0
##
##  Reliability if an item is dropped:
##               raw_alpha std.alpha G6(smc) average_r  S/N alpha se var.r
## Population-     1.8e-04      0.62    0.66      0.29 1.61  9.0e-05 0.064
## Professional    8.2e-05      0.45    0.43      0.17 0.81  8.9e-05 0.029
## Employed-       1.8e-04      0.58    0.62      0.25 1.36  6.5e-05 0.061
## Government      1.0e-04      0.51    0.58      0.21 1.04  5.3e-05 0.071
## MedianHomeVal   5.1e-01      0.57    0.55      0.25 1.30  7.2e-02 0.020
##               med.r
## Population-    0.28
## Professional   0.15
## Employed-      0.19
## Government     0.13
## MedianHomeVal  0.25
##
##  Item statistics
##                n  raw.r std.r r.cor r.drop   mean      sd
## Population-   61 -0.026  0.51  0.31 -0.026 363996     1.8
## Professional  61  0.685  0.75  0.74  0.685      4     3.1
## Employed-     61  0.011  0.58  0.42  0.010 363929     7.5
## Government    61  0.180  0.67  0.55  0.180     27     9.4
## MedianHomeVal 61  1.000  0.60  0.53  0.241 163557 56446.9
```

```
#raw_alpha = 0
#This should be > 0.7 but it is low. Medhomevalue is messing it up.
```

a) Conduct a principal component analysis using the covariance matrix (the default for prcomp and many routines in other software), and interpret the results. How much of the variance is accounted for in the first component and why is this?

```
#Create PCA
pca_census = prcomp(census_data)

#Check Scree Plot
plot(pca_census)
abline(1, 0)
```




```
#Check PCA Summary Information
summary(pca_census)
```

```
## Importance of components:
##                           PC1   PC2   PC3   PC4  PC5
## Standard deviation     56447 10.21 6.219 2.247 1.56
## Proportion of Variance     1  0.00 0.000 0.000 0.00
## Cumulative Proportion      1  1.00 1.000 1.000 1.00
```

- Without scaling or fixing medianhomevalue the 1st principal component supposedly accounts for 100% of the variance in the data. This is wrong. We need to scale first. The standard deviation for median home value is about 56000 whereas the other variables are between 1 and 9. This must be addressed prior to pca or as part of it.

```
#Using the psych package for next part as instructed - with 5
pca_census_a2 = psych::principal(census_data, rotate="varimax", nfactors=5, scores=TRUE)
pca_census_a2
```

```
## Principal Components Analysis
## Call: psych::principal(r = census_data, nfactors = 5, rotate = "varimax",
##     scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                RC5   RC4   RC3   RC2   RC1 h2      u2 com
## Population    0.02 -0.04  0.98  0.15 -0.09  1 2.2e-16 1.1
## Professional  0.41  0.20 -0.12  0.00  0.88  1 2.2e-15 1.6
## Employed      0.00 -0.21  0.16  0.96  0.00  1 8.9e-16 1.2
## Government    0.07  0.96 -0.04 -0.21  0.16  1 1.2e-15 1.2
## MedianHomeVal 0.94  0.06  0.04  0.00  0.33  1 1.7e-15 1.3
##
##                        RC5  RC4  RC3  RC2  RC1
## SS loadings           1.06 1.01 1.01 1.00 0.92
## Proportion Var        0.21 0.20 0.20 0.20 0.18
## Cumulative Var        0.21 0.41 0.62 0.82 1.00
## Proportion Explained  0.21 0.20 0.20 0.20 0.18
## Cumulative Proportion 0.21 0.41 0.62 0.82 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 5 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
```

```
print(pca_census_a2$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##               RC5    RC4    RC3    RC2    RC1
## MedianHomeVal  0.942
## Government            0.960
## Population                  0.983
## Employed                           0.965
## Professional   0.413                       0.880
##
##                  RC5   RC4   RC3   RC2   RC1
## SS loadings    1.063 1.012 1.010 1.000 0.915
## Proportion Var 0.213 0.202 0.202 0.200 0.183
## Cumulative Var 0.213 0.415 0.617 0.817 1.000
```

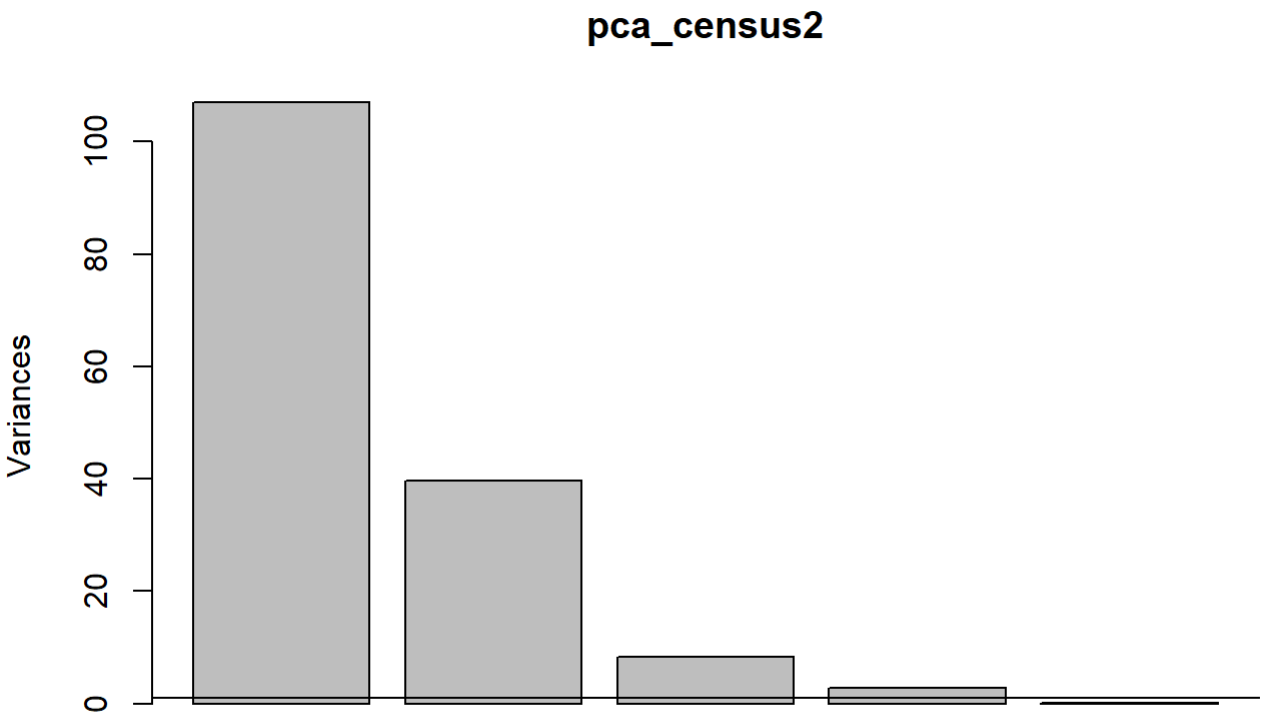
- Doesn't make much sense, as expected.

b) Try dividing the MedianHomeValue field by 100,000 so that the median home value in the dataset is measured in $100,000's rather than in dollars. How does this change the analysis?

```
#new dataframe
census_data_fixed <- census_data
census_data_fixed$MedianHomeVal <-  census_data_fixed$MedianHomeVal/100000
head(census_data_fixed)
```

```
##   Population Professional Employed Government MedianHomeVal
## 1       2.67         5.71    69.02       30.3          1.48
## 2       2.25         4.37    72.98       43.3          1.44
## 3       3.12        10.27    64.94       32.0          2.11
## 4       5.14         7.44    71.29       24.5          1.85
## 5       5.54         9.25    74.94       31.0          2.23
## 6       5.04         4.84    53.61       48.2          1.60
```

```
#Create PCA
pca_census2 = prcomp(census_data_fixed)

#Check Scree Plot
plot(pca_census2)
abline(1, 0)
```

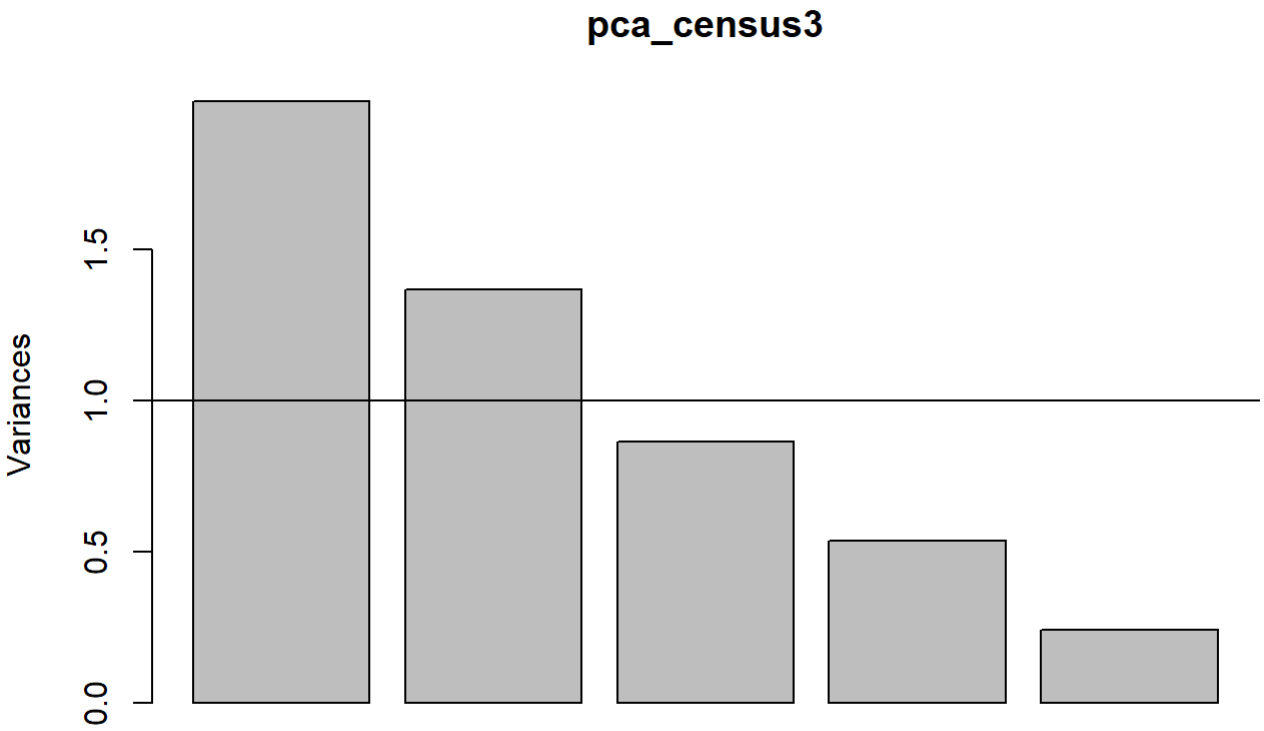



```
#Check PCA Summary Information
summary(pca_census2)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     10.345 6.2986 2.89324 1.69348 0.39331
## Proportion of Variance  0.677 0.2510 0.05295 0.01814 0.00098
## Cumulative Proportion   0.677 0.9279 0.98088 0.99902 1.00000
```

- So this is better than before for sure, but it's still not as good as just scaling in prcomp so everything is divided by standard deviation. Note here how the variance scale has shrunk making the contribution of the different components to be exposed, and so that a single component does not account for all the variance. Going off this method I'd pick 2 principal components to account for about 93% of the variance in the data. Below I check what it would be with scaling with the divide by standard deviation method.

```
#Create PCA
pca_census3 = prcomp(census_data, center=T, scale=T)

#Check Scree Plot
plot(pca_census3)
abline(1, 0)
```




```
#Check PCA Summary Information
summary(pca_census3)
```

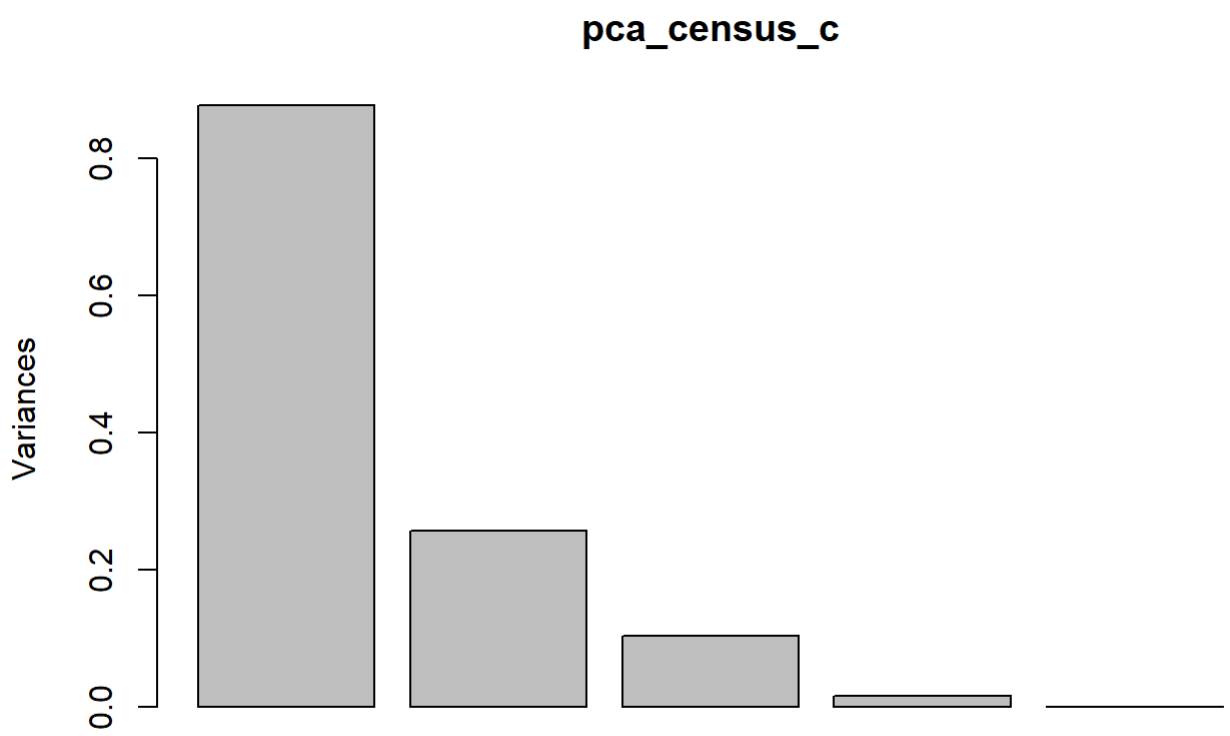
```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5
## Standard deviation     1.4114 1.1694 0.9296 0.7315 0.49126
## Proportion of Variance 0.3984 0.2735 0.1728 0.1070 0.04827
## Cumulative Proportion  0.3984 0.6719 0.8447 0.9517 1.00000
```

- I like this better, and it confirms the 2 principal components is probably good enough and will capture more realistically 67% of the variance.

c) Compute the PCA with the correlation matrix instead. How does this change the result and how does your answer compare (if you did it) with your answer in b)?

- Not sure if this means to compute with the dataset from B (medhomeval/100000) or A (original). So I did it for both and found it doesn't matter which makes sense. The correlations don't change when scaling a single column like that.

```
#using census_cor_mat from the original dataset

#census_cor_mat<-cor(census_data)
#nut_cor_mat
#corrplot(census_cor_mat, type = "lower")
#ggcorr(census_cor_mat, label=TRUE)

#Create PCA
pca_census_c = prcomp(census_cor_mat)

#Check Scree Plot
plot(pca_census_c)
abline(1, 0)
```



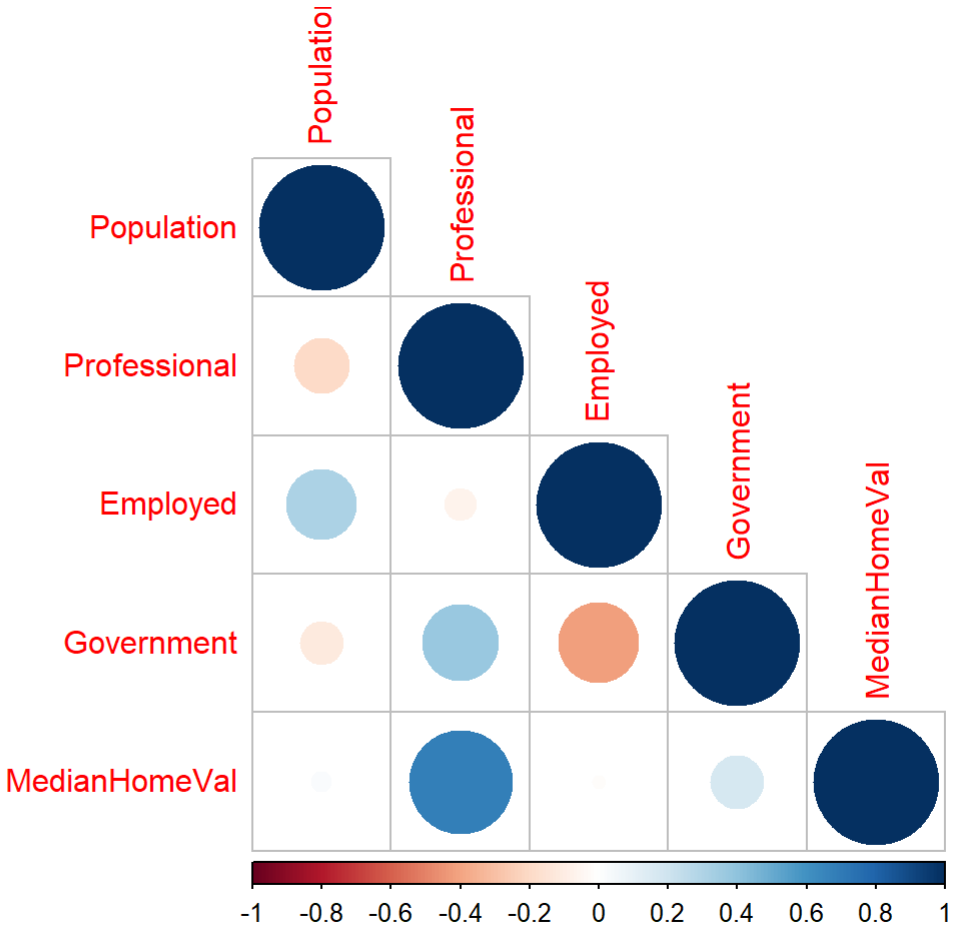

```
#Check PCA Summary Information
summary(pca_census_c)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4       PC5
## Standard deviation     0.9371 0.5068 0.32236 0.12312 3.073e-17
## Proportion of Variance 0.7002 0.2048 0.08287 0.01209 0.000e+00
## Cumulative Proportion  0.7002 0.9051 0.98791 1.00000 1.000e+00
```

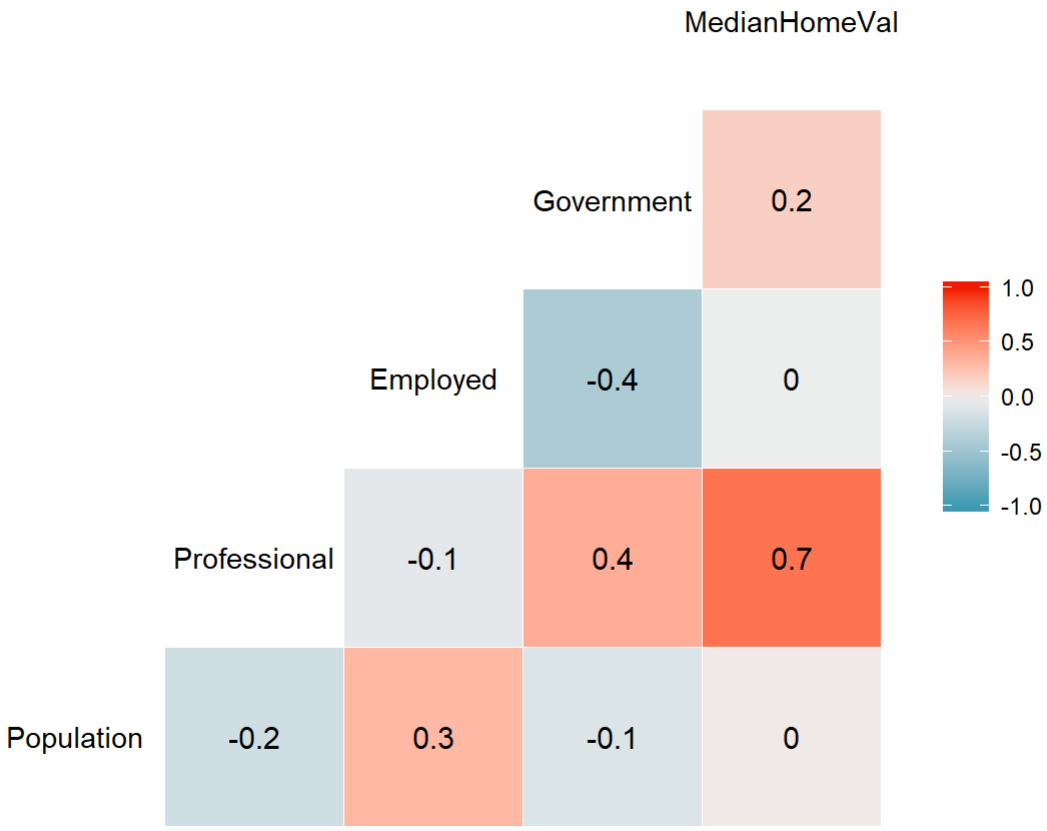
- There is a difference compared to be. The scree plot variances are all below 1 now, but the general shape is the same. In terms of how much cumulative variance is explained by the different principal components - PC1 and PC2 in this case add up to 90% while in part B PC1 and PC2 added up to 93%. So they are quite similar, despite the scree plot changing.

d) Analyze the correlation matrix for this dataset for significance, and also look for variables that are extremely correlated or uncorrelated. Discuss the effect of this on the analysis.

```
#Check Correlations
#census_cor_mat<-cor(census_data)
corrplot(census_cor_mat, type = "lower")
```




```
ggcorr(census_data, label=TRUE)
```




```
census_cor_mat
```

```
##                Population Professional    Employed Government
## Population      1.00000000   -0.1922736  0.31321982 -0.1194831
## Professional   -0.19227360    1.0000000 -0.06523680  0.3731722
## Employed        0.31321982   -0.0652368  1.00000000 -0.4111161
## Government     -0.11948307    0.3731722 -0.41111605  1.0000000
## MedianHomeVal   0.02614869    0.6852879 -0.01034666  0.1797010
##                MedianHomeVal
## Population        0.02614869
## Professional      0.68528795
## Employed         -0.01034666
## Government        0.17970100
## MedianHomeVal     1.00000000
```

- Significant Correlations: MedianHomeVal and Professional - This makes sense that it is a strong positive correlation between professional and the median home price found. Assuming Professional is the percentage of economy that is in professional/industry setting, or length of time spent as a professional, years of education, or even some sort of job count in an area, all should lead to higher home prices.
- Other Noteable Correlations: Professional/Government(0.4), Employed/Government(-0.4), Population/Employed(0.3) So we could argue that more professional in an area leads more government, but maybe less overall employment. The lower correlation between population and employed could be that in higher population density areas, maybe in urban settings, there are often more jobs.

- Employed/MedianHomeVal and Population/MedianHomeVal have no correlation which is interesting but makes sense for population but is strange for employed. I would have thought that more employement would lead to higher home values, but maybe not.

Problem 7

Problem 7: (Principal Component Analysis - 20 Points) Download the “glass.csv” dataset and perform a principal component analysis on the data. The data provides different elements used to determine the type of window glass. There are 214 samples and 10 variables, including the type of glass variable (do not include in the PCA).

Attribute Information:

*** Id number: 1 to 214**

*** RI: refractive index**

*** Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)**

*** Mg: Magnesium**

*** Al: Aluminum**

*** Si: Silicon**

*** K: Potassium**

*** Ca: Calcium**

*** Ba: Barium**

*** Fe: Iron**

*** Type of glass: (class attribute) – 1 building_windows_float_processed – * 2 building_windows_non_float_processed – 3 vehicle_windows_float_processed – * 4 vehicle_windows_non_float_processed (none in this database) – 5 containers – * 6 tableware – 7 headlamps**

Choose your PCA method carefully and give a reason for your choice. Try different ways of formulating the analysis until you get a small set of components that are easy to interpret.

```
#Read in Datasets
glass_data <- read.csv("glass.csv")

#Check Sample Size and Number of Variables
dim(glass_data)
```

```
## [1] 214  10
```

```
#Missing values
sum(is.na(glass_data))
```

```
## [1] 0
```

```
#Show for first 6 rows of data
head(glass_data)
```

```
##        RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
## 2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
## 3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
## 4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
## 5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
## 6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
```

```
#Column Names
names(glass_data)
```

```
##  [1] "RI"   "Na"   "Mg"   "Al"   "Si"   "K"    "Ca"   "Ba"   "Fe"   "Type"
```

```
#structure
str(glass_data)
```

```
## 'data.frame':    214 obs. of  10 variables:
##  $ RI  : num  1.52 1.52 1.52 1.52 1.52 ...
##  $ Na  : num  13.6 13.9 13.5 13.2 13.3 ...
##  $ Mg  : num  4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...
##  $ Al  : num  1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...
##  $ Si  : num  71.8 72.7 73 72.6 73.1 ...
##  $ K   : num  0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...
##  $ Ca  : num  8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...
##  $ Ba  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Fe  : num  0 0 0 0 0 0.26 0 0 0 0.11 ...
##  $ Type: int  1 1 1 1 1 1 1 1 1 1 ...
```

```
#describe
describe(glass_data)
```

```
##      vars   n  mean   sd median trimmed  mad   min   max range  skew
## RI      1 214  1.52 0.00   1.52    1.52 0.00  1.51  1.53  0.02  1.60
## Na      2 214 13.41 0.82  13.30   13.38 0.64 10.73 17.38  6.65  0.45
## Mg      3 214  2.68 1.44   3.48    2.87 0.30  0.00  4.49  4.49 -1.14
## Al      4 214  1.44 0.50   1.36    1.41 0.31  0.29  3.50  3.21  0.89
## Si      5 214 72.65 0.77  72.79   72.71 0.57 69.81 75.41  5.60 -0.72
## K       6 214  0.50 0.65   0.56    0.43 0.17  0.00  6.21  6.21  6.46
## Ca      7 214  8.96 1.42   8.60    8.74 0.66  5.43 16.19 10.76  2.02
## Ba      8 214  0.18 0.50   0.00    0.03 0.00  0.00  3.15  3.15  3.37
## Fe      9 214  0.06 0.10   0.00    0.04 0.00  0.00  0.51  0.51  1.73
## Type   10 214  2.78 2.10   2.00    2.48 1.48  1.00  7.00  6.00  1.10
##      kurtosis   se
## RI       4.72 0.00
## Na       2.90 0.06
## Mg      -0.45 0.10
## Al       1.94 0.03
## Si       2.82 0.05
## K       52.87 0.04
## Ca       6.41 0.10
## Ba      12.08 0.03
## Fe       2.52 0.01
## Type    -0.33 0.14
```

- Drop the type column because it's not really numeric.

```
#Capture numeric variables
glass_data2 <- glass_data[,c(1:9)]
head(glass_data2)
```

```
##        RI    Na   Mg   Al    Si    K   Ca Ba   Fe
## 1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00
## 2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00
## 3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00
## 4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00
## 5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00
## 6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26
```

```
#Test KMO Sampling Adequacy
#library(psych)
KMO(glass_data2)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = glass_data2)
## Overall MSA =  0.13
## MSA for each item =
##   RI   Na   Mg   Al   Si    K   Ca   Ba   Fe
## 0.83 0.07 0.12 0.14 0.07 0.07 0.17 0.10 0.14
```

```
#Overall MSA =  0.13
#This is NOT >=0.5 or 0.6 - NOT good - VERY low

#Test Bartlett's Test of Sphericity
#library(REdaS)
bart_spher(glass_data2)
```

```
##  Bartlett's Test of Sphericity
##
## Call: bart_spher(x = glass_data2)
##
##      X2 = 1837.456
##      df = 36
## p-value < 2.22e-16
```

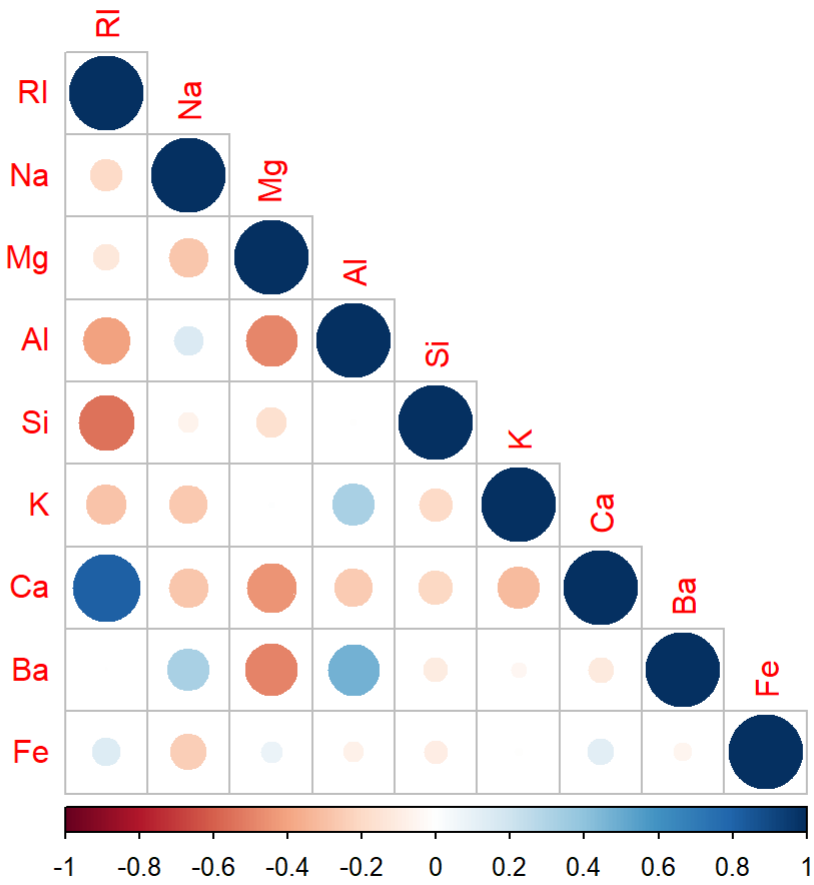
```
#p-value < 2.22e-16 (Very Small Number)
#This is significant

#Test for Reliability Analysis using Cronbach's Alpha
#library(psych)
alpha(glass_data2, check.keys=TRUE)
```

```
## Warning in alpha(glass_data2, check.keys = TRUE): Some items were negatively correlated with
total scale and were automatically reversed.
##  This is indicated by a negative sign for the variable name.
```

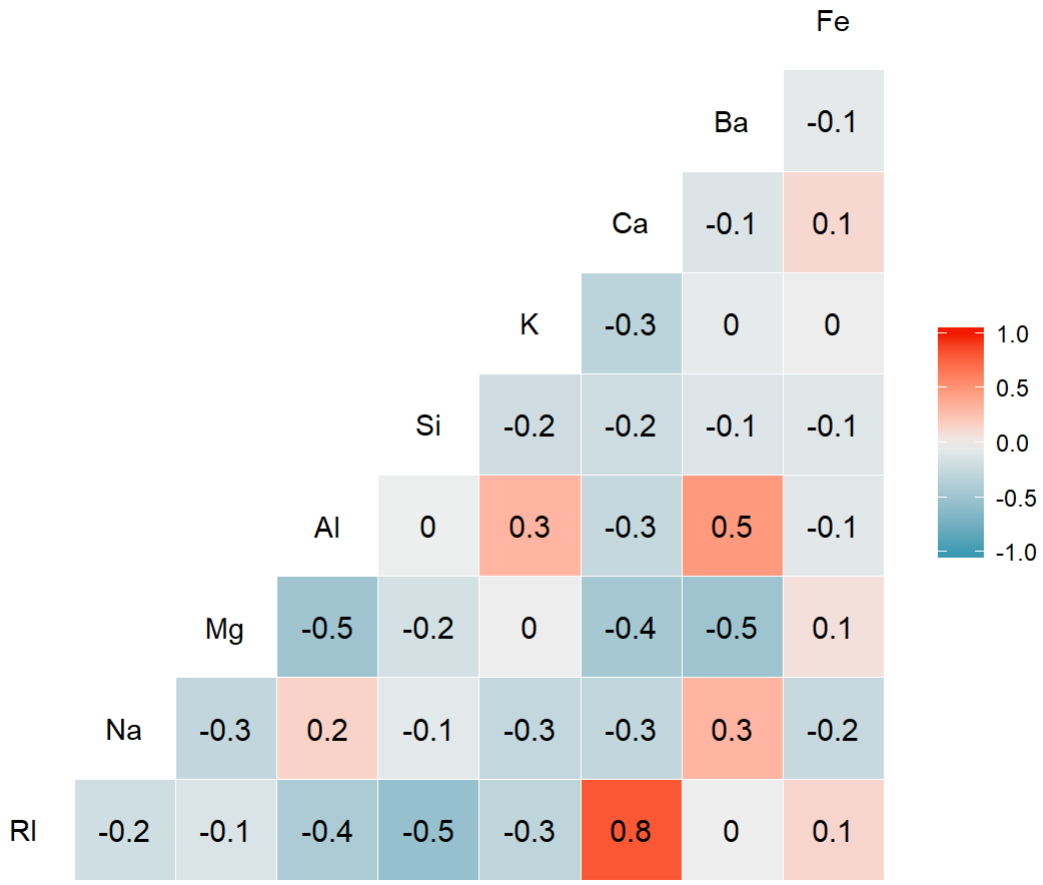
```
##
## Reliability analysis
## Call: alpha(x = glass_data2, check.keys = TRUE)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##       0.37      0.63    0.95      0.16 1.7 0.065   42 0.34     0.15
##
##  lower alpha upper     95% confidence boundaries
## 0.24 0.37 0.5
##
##  Reliability if an item is dropped:
##     raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## RI-      0.38      0.53    0.93      0.12 1.1    0.066 0.050  0.12
## Na       0.27      0.61    0.83      0.16 1.6    0.078 0.068  0.12
## Mg-      0.42      0.62    0.79      0.17 1.6    0.046 0.053  0.15
## Al       0.22      0.53    0.88      0.13 1.1    0.081 0.065  0.12
## Si       0.36      0.64    0.86      0.18 1.8    0.068 0.063  0.17
## K        0.37      0.65    0.86      0.19 1.9    0.067 0.061  0.16
## Ca-      0.44      0.57    0.82      0.14 1.3    0.044 0.048  0.12
## Ba       0.27      0.60    0.89      0.16 1.5    0.077 0.065  0.16
## Fe-      0.37      0.63    0.99      0.17 1.7    0.067 0.077  0.20
##
##  Item statistics
##       n raw.r std.r r.cor r.drop  mean    sd
## RI- 214  0.64  0.72  0.72  0.639 73.89 0.003
## Na  214  0.53  0.47  0.48  0.303 13.41 0.817
## Mg- 214  0.54  0.43  0.44  0.081 72.73 1.442
## Al  214  0.70  0.70  0.71  0.601  1.44 0.499
## Si  214  0.35  0.36  0.37  0.108 72.65 0.775
## K   214  0.28  0.32  0.32  0.075  0.50 0.652
## Ca- 214  0.51  0.59  0.61  0.059 66.45 1.423
## Ba  214  0.58  0.51  0.52  0.454  0.18 0.497
## Fe- 214  0.24  0.40  0.23  0.210 75.35 0.097
```

```
#raw_alpha = 0.37
#This should be > 0.7 but it is low.
```

```
#Check Correlations
glass_cor_mat<-cor(glass_data2)
corrplot(glass_cor_mat, type = "lower")
```



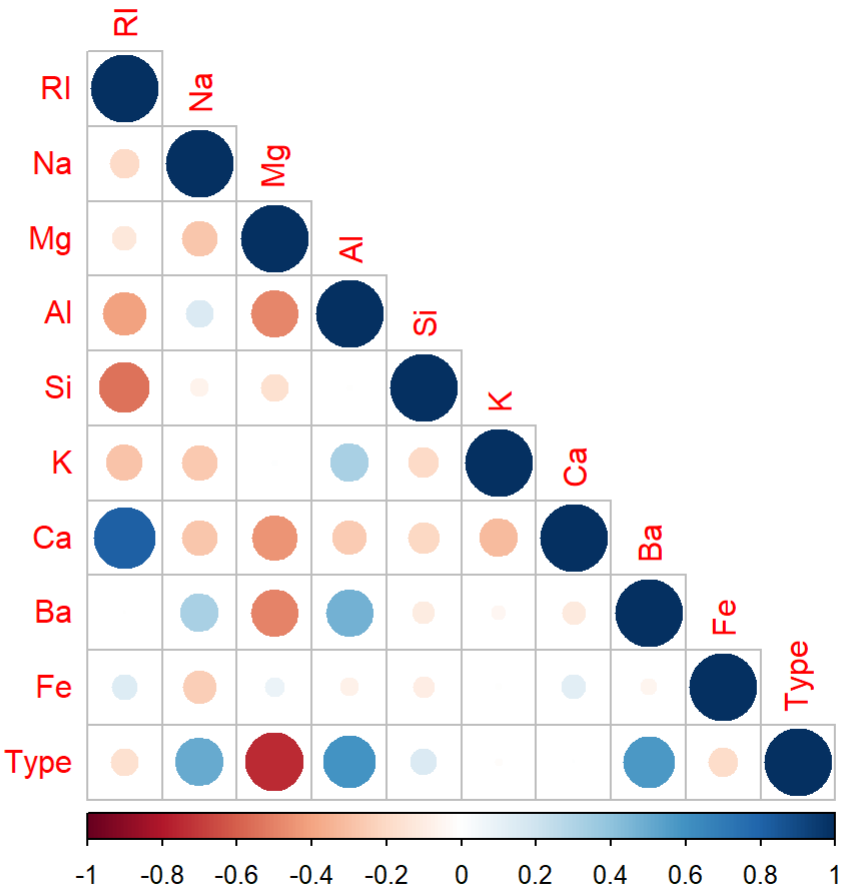

```
ggcorr(glass_data2, label=TRUE)
```



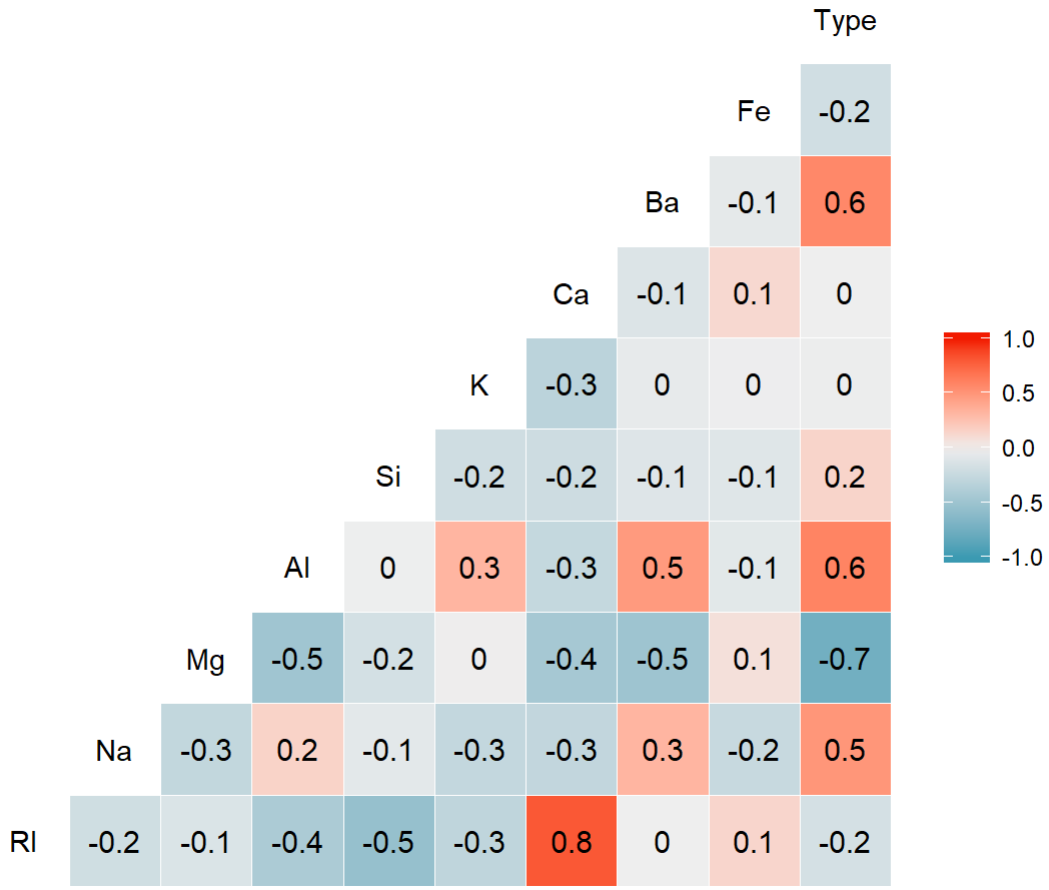

```
glass_cor_mat
```

```
##               RI          Na           Mg          Al          Si
## RI  1.0000000000 -0.19188538 -0.122274039 -0.40732603 -0.54205220
## Na -0.1918853790  1.00000000 -0.273731961  0.15679367 -0.06980881
## Mg -0.1222740393 -0.27373196  1.000000000 -0.48179851 -0.16592672
## Al -0.4073260341  0.15679367 -0.481798509  1.00000000 -0.00552372
## Si -0.5420521997 -0.06980881 -0.165926723 -0.00552372  1.00000000
## K  -0.2898327111 -0.26608650  0.005395667  0.32595845 -0.19333085
## Ca  0.8104026963 -0.27544249 -0.443750026 -0.25959201 -0.20873215
## Ba -0.0003860189  0.32660288 -0.492262118  0.47940390 -0.10215131
## Fe  0.1430096093 -0.24134641  0.083059529 -0.07440215 -0.09420073
##                K          Ca            Ba           Fe
## RI -0.289832711  0.8104027 -0.0003860189  0.143009609
## Na -0.266086504 -0.2754425  0.3266028795 -0.241346411
## Mg  0.005395667 -0.4437500 -0.4922621178  0.083059529
## Al  0.325958446 -0.2595920  0.4794039017 -0.074402151
## Si -0.193330854 -0.2087322 -0.1021513105 -0.094200731
## K   1.000000000 -0.3178362 -0.0426180594 -0.007719049
## Ca -0.317836155  1.0000000 -0.1128409671  0.124968219
## Ba -0.042618059 -0.1128410  1.0000000000 -0.058691755
## Fe -0.007719049  0.1249682 -0.0586917554  1.000000000
```

```
#what about with Type
glass_cor_mat_withtype<-cor(glass_data)
corrplot(glass_cor_mat_withtype, type = "lower")
```




```
ggcorr(glass_data, label=TRUE)
```

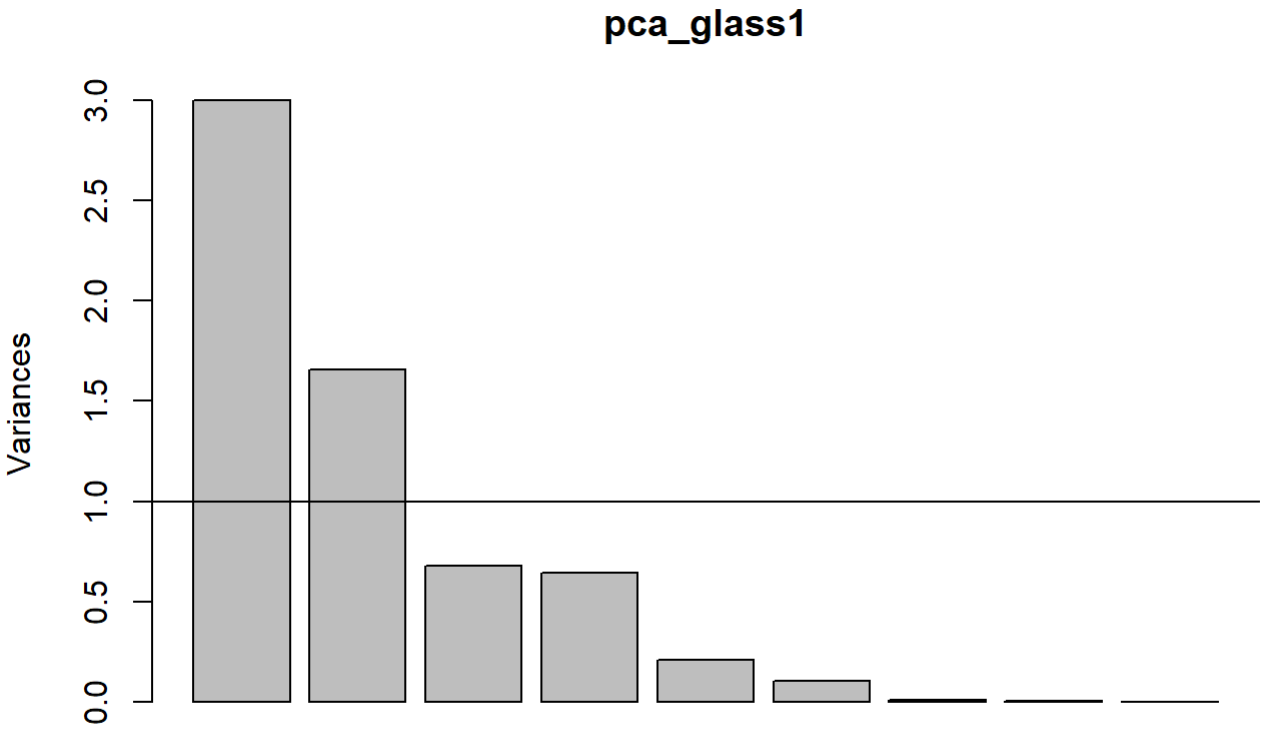



```
glass_cor_mat_withtype
```

```
##                RI          Na           Mg          Al          Si
## RI    1.0000000000 -0.19188538 -0.122274039 -0.40732603 -0.54205220
## Na   -0.1918853790  1.00000000 -0.273731961  0.15679367 -0.06980881
## Mg   -0.1222740393 -0.27373196  1.000000000 -0.48179851 -0.16592672
## Al   -0.4073260341  0.15679367 -0.481798509  1.00000000 -0.00552372
## Si   -0.5420521997 -0.06980881 -0.165926723 -0.00552372  1.00000000
## K    -0.2898327111 -0.26608650  0.005395667  0.32595845 -0.19333085
## Ca    0.8104026963 -0.27544249 -0.443750026 -0.25959201 -0.20873215
## Ba   -0.0003860189  0.32660288 -0.492262118  0.47940390 -0.10215131
## Fe    0.1430096093 -0.24134641  0.083059529 -0.07440215 -0.09420073
## Type -0.1642372146  0.50289804 -0.744992888  0.59882921  0.15156526
##                 K            Ca            Ba           Fe          Type
## RI   -0.289832711  0.8104026963 -0.0003860189  0.143009609 -0.1642372146
## Na   -0.266086504 -0.2754424856  0.3266028795 -0.241346411  0.5028980423
## Mg    0.005395667 -0.4437500264 -0.4922621178  0.083059529 -0.7449928875
## Al    0.325958446 -0.2595920102  0.4794039017 -0.074402151  0.5988292084
## Si   -0.193330854 -0.2087321537 -0.1021513105 -0.094200731  0.1515652579
## K     1.000000000 -0.3178361547 -0.0426180594 -0.007719049 -0.0100544638
## Ca   -0.317836155  1.0000000000 -0.1128409671  0.124968219  0.0009522246
## Ba   -0.042618059 -0.1128409671  1.0000000000 -0.058691755  0.5751614590
## Fe   -0.007719049  0.1249682190 -0.0586917554  1.000000000 -0.1882775640
## Type -0.010054464  0.0009522246  0.5751614590 -0.188277564  1.0000000000
```

- Before jumping into the PCA just interesting to note the high correlation between RI and Ca. So if you want a greater refractive index in you glass you'd want to add Calcium apparently. Now on to PCA.
- When including Type we see some of the relationships that exist between the elements and the Type but it's very hard to interpret. So we continue as planned with Type not included.

```
#Create PCA
pca_glass1 = prcomp(glass_data2)

#Check Scree Plot
plot(pca_glass1)
abline(1, 0)
```




```
#Check PCA Summary Information
summary(pca_glass1)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.7326 1.2881 0.8244 0.8020 0.45657 0.31806 0.09486
## Proportion of Variance 0.4762 0.2632 0.1078 0.1020 0.03307 0.01605 0.00143
## Cumulative Proportion  0.4762 0.7394 0.8472 0.9492 0.98229 0.99834 0.99977
##                           PC8      PC9
## Standard deviation     0.03844 0.000985
## Proportion of Variance 0.00023 0.000000
## Cumulative Proportion  1.00000 1.000000
```

- So if we wanted to account for 95% of the variance in the data we'd pick 4 components. That said, from the "knee" in the Scree Plot, we can see that 2 components will probably more accurately account for the variance. So I will choose 2 components for this PCA which will have a Cumulative Proportion of Variance of 74% which is decent for this exploratory analysis.

```
#Using the psych package for next part as instructed - with 2 components
library(GPArotation) #used if rotated with "oblimin". The result was similar so I stayed with va
rimax.
pca_glass2 = psych::principal(glass_data2, rotate="varimax", nfactors=2, scores=TRUE)
pca_glass2
```

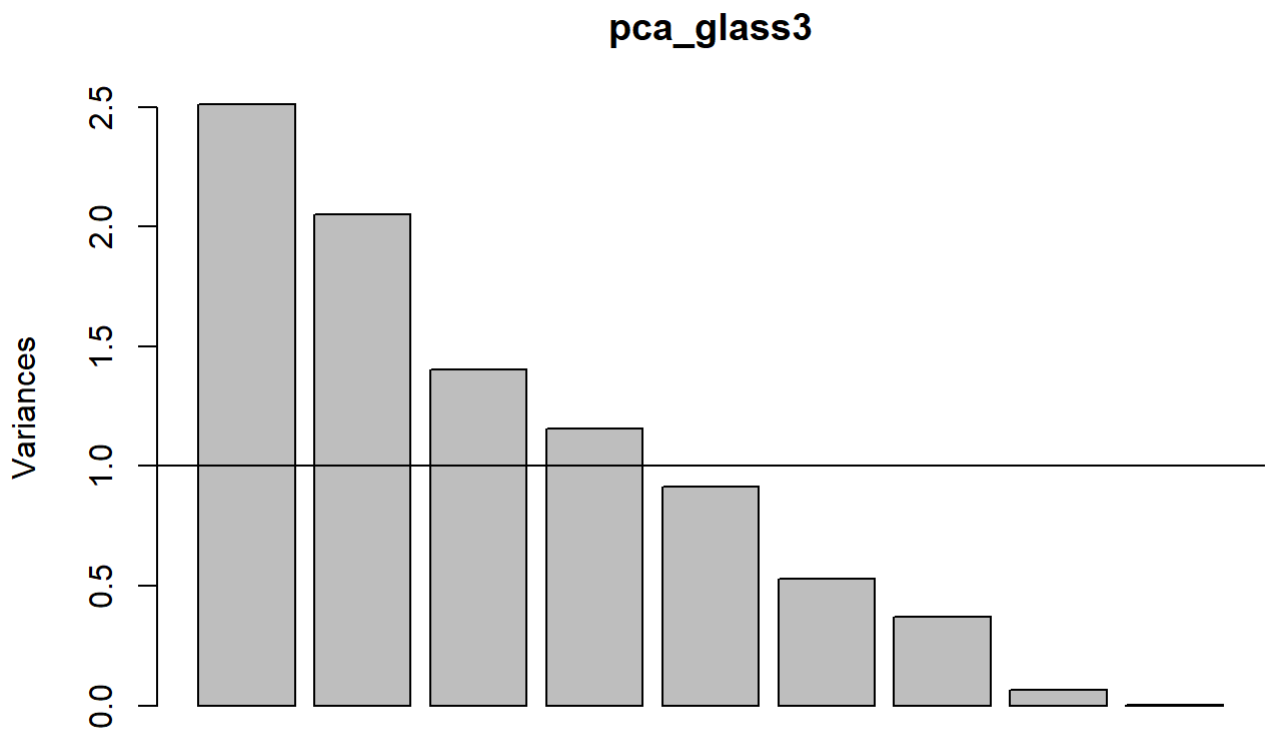
```
## Principal Components Analysis
## Call: psych::principal(r = glass_data2, nfactors = 2, rotate = "varimax",
##     scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      RC1   RC2    h2    u2 com
## RI  0.94 -0.15 0.914 0.086 1.0
## Na -0.12  0.55 0.317 0.683 1.1
## Mg -0.33 -0.80 0.753 0.247 1.3
## Al -0.32  0.73 0.640 0.360 1.4
## Si -0.42  0.02 0.181 0.819 1.0
## K  -0.41  0.01 0.169 0.831 1.0
## Ca  0.92 -0.03 0.853 0.147 1.0
## Ba  0.06  0.80 0.639 0.361 1.0
## Fe  0.19 -0.24 0.095 0.905 1.9
##
##                        RC1  RC2
## SS loadings           2.37 2.20
## Proportion Var        0.26 0.24
## Cumulative Var        0.26 0.51
## Proportion Explained  0.52 0.48
## Cumulative Proportion 0.52 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  0.15
##  with the empirical chi square  336.33  with prob <  6.8e-60
##
## Fit based upon off diagonal values = 0.74
```

```
print(pca_glass2$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##    RC1    RC2
## RI  0.944
## Ca  0.923
## Na         0.550
## Mg        -0.801
## Al         0.732
## Ba         0.797
## Si -0.425
## K  -0.411
## Fe
##
##                  RC1   RC2
## SS loadings    2.365 2.196
## Proportion Var 0.263 0.244
## Cumulative Var 0.263 0.507
```

- Thinking about our Components with this 2 Component model.
- RC1 connects Refractive index with Ca, Si, and K. But the relationships vary. More Ca and higher RI, More Si or K yields a lower RI.
- RC2 connects Na, Mg, Al, Ba together as a property of glass. More Na, Al, Ba in glass is connected to lower Mg levels.
- Fe doesn't show up with our cutoff, but from the correlation plot we can see that it isn't very linearly related to the other variables so it our reduced 2 component model it is understandable that it wouldn't show up.
- Just to Check what it would look like if scaled:

```
#Create PCA but scaled because
pca_glass3 = prcomp(glass_data2, center=T, scale=T)

#Check Scree Plot
plot(pca_glass3)
abline(1, 0)
```




```
#Check PCA Summary Information
summary(pca_glass3)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6    PC7
## Standard deviation     1.585 1.4318 1.1853 1.0760 0.9560 0.72639 0.6074
## Proportion of Variance 0.279 0.2278 0.1561 0.1286 0.1016 0.05863 0.0410
## Cumulative Proportion  0.279 0.5068 0.6629 0.7915 0.8931 0.95173 0.9927
##                           PC8     PC9
## Standard deviation     0.25269 0.04011
## Proportion of Variance 0.00709 0.00018
## Cumulative Proportion  0.99982 1.00000
```

- Scaled give 4 components, but accounts for roughly the same cumulative proportion of variance as the two components above. But also scaling might not be a good idea assuming the units for all the element ‘ingredients’ in the glass dataset are the same. So the 0.00 for more Ba and low Fe, etc are probably relevant and should not be obscured by scaling unnecessarily.
- I choose the 2 components.

Finally, run a common factor analysis on the same data. What difference, if any, do you find? Does the factor analysis change your ability to interpret the results practically?

```
#Factor Analysis
fit_glass = factanal(glass_data2, 2)
print(fit_glass$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##    Factor1 Factor2
## RI  0.822
## Ca  0.997
## Mg -0.407  -0.911
## Al          0.657
## Ba          0.598
## Na          0.427
## Si
## K
## Fe
##
##                Factor1 Factor2
## SS loadings      2.190   1.975
## Proportion Var   0.243   0.219
## Cumulative Var   0.243   0.463
```

```
summary(fit_glass)
```

```
##              Length Class    Mode
## converged     1     -none-   logical
## loadings     18     loadings numeric
## uniquenesses  9     -none-   numeric
## correlation  81     -none-   numeric
## criteria      3     -none-   numeric
## factors       1     -none-   numeric
## dof           1     -none-   numeric
## method        1     -none-   character
## rotmat        4     -none-   numeric
## STATISTIC     1     -none-   numeric
## PVAL          1     -none-   numeric
## n.obs         1     -none-   numeric
## call          3     -none-   call
```

- There are some differences, namely Factor1 has RI, Ca, Mg together wheras PC1 had RI, Ca with Si and K. That said Mg has a negative sign in both Factor1 and Factor two so perhaps when the other elements in the Factor are present, there's less Mg. Problematically, Si, Ki, and Fe are not included in Factor1 or Factor2, but they Si and K are parts of the Principal Components (above the cutoff 0.4). Overall this is slightly harder to interpret - at least with 2 factors and 2 components for the PCA. When I switched to 3 for PCA and 3 for Factor Analysis the the results changed, and may provide more practically relevant variable groupings and relationships. Without any material science background I have a hard time understanding how the factors make sense for glass production or design. I think the PCA is more understandable at least for this 2 factor/component case.