Analyzing Medical Readmissions with PCA

Introduction:

The goal of this project is to determine if it is possible to identify the factors most responsible for patients being readmitted to the hospital. This will be done by using principal component analysis on a medical dataset. To achive this it is vital to understand what PCA is. According to (Vidhya, 2020), "In simple words, PCA is a method of obtaining important variables (in form of components) from a large set of variables available in a data set. It extracts low dimensional set of features by taking a projection of irrelevant dimensions from a high dimensional data set with a motive to capture as much information as possible. With fewer variables obtained while minimising the loss of information, visualization also becomes much more meaningful."

According to (O'Rourke, 2020) One of the main assumptions of PCA is Linearity. "The relationship between all observed variables should be linear."

Research Question:

Can PCA be used to identify the factors most responsible for hospital readmission?

Step One: Install the necesarry packages

In [1]:

```
library(ggplot2)
library(tidyverse)
```

```
Warning message:
"package 'ggplot2' was built under R version 3.6.3"
Warning message:
"package 'tidyverse' was built under R version 3.6.3"
-- Attaching packages -----------------------------------------------------------------
------------- tidyverse 1.3.0 --

v tibble  3.0.4     v dplyr   1.0.2
v tidyr   1.1.2     v stringr 1.4.0
v readr   1.4.0     v forcats 0.5.0
v purrr   0.3.4

Warning message:
"package 'tibble' was built under R version 3.6.3"
Warning message:
"package 'tidyr' was built under R version 3.6.3"
Warning message:
"package 'readr' was built under R version 3.6.3"
```

```
Warning message:
"package 'purrr' was built under R version 3.6.3"
Warning message:
"package 'dplyr' was built under R version 3.6.3"
Warning message:
"package 'stringr' was built under R version 3.6.3"
Warning message:
"package 'forcats' was built under R version 3.6.3"
-- Conflicts ------------------------------------------------------------------
------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

In [2]:

```
install.packages("ggfortify")
```

```
package 'ggfortify' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\ContactTracer\AppData\Local\Temp\RtmpM3m6dv\downloaded_packages
```

In [3]:

```
library(ggfortify)
```

```
Warning message:
"package 'ggfortify' was built under R version 3.6.3"
```

Step Two: Import the data

In [4]:

```
data1 <- read_csv("C:\\Users\\ContactTracer\\Desktop\\D212 Data Mining II Task 2\\medic
```

```
-- Column specification --------------------------------------------------------
--------------------------------
cols(
  .default = col_character(),
  CaseOrder = col_double(),
  Zip = col_double(),
  Lat = col_double(),
  Lng = col_double(),
  Population = col_double(),
  Children = col_double(),
  Age = col_double(),
  Income = col_double(),
  VitD_levels = col_double(),
  Doc_visits = col_double(),
  Full_meals_eaten = col_double(),
  vitD_supp = col_double(),
  Initial_days = col_double(),
  TotalCharge = col_double(),
  Additional_charges = col_double(),
  Item1 = col_double(),
  Item2 = col_double(),
  Item3 = col_double(),
  Item4 = col_double(),
  Item5 = col_double()
  # ... with 3 more columns
)
i Use `spec()` for the full column specifications.
```

Many of the factors included in the dataset will either have no effect on hospital readmission, or will tell nothing of interest about the patients. All unneeded factors will be removed, and then the data will be scaled. The reason for scaling the data is that accoriding to (Vadapalli, 2020) "All variables should be accessed on the same ratio level of measurement." The reason for this is if the scale for one variable was greater then the other variables, that variable would have an undue affect on the PCAs.

In [5]:
```
data2 <- data1[ -c(1:14, 18:19, 23, 25:39, 43:50) ]
```

In [6]:
```
data2$ReAdmis <- as.numeric(as.factor(data2$ReAdmis))
```

In [7]:
```
addmission_data <- data2$ReAdmis
```

In [8]:
```
data3 <- data2[-c(4)]
```

In [9]:
```
data_scaled <-scale(data3)
```

In [45]:
```
write.csv(data_scaled,"C:\\Users\\ContactTracer\\Desktop\\data_scaled.csv", row.names =
```

Now that the data is scaled a covariance matrix can be created

In [34]:
```
S <- cov(data_scaled)
```

In [35]:
```
S
```

A matrix: 9 × 9 of type dbl

	Children	Age	Income	VitD_levels	Doc_visits	vitD_supp
Children	1.000000000	0.009835607	0.007176498	0.009487013	-0.002292199	-0.004319286
Age	0.009835607	1.000000000	-0.012228139	0.010315387	0.006897840	0.010014004
Income	0.007176498	-0.012228139	1.000000000	-0.013115002	0.013463739	0.001253438
VitD_levels	0.009487013	0.010315387	-0.013115002	1.000000000	0.010210475	-0.007203220
Doc_visits	-0.002292199	0.006897840	0.013463739	0.010210475	1.000000000	0.005680941
vitD_supp	-0.004319286	0.010014004	0.001253438	-0.007203220	0.005680941	1.000000000
Initial_days	0.022466620	0.016264290	-0.012464817	-0.003641791	-0.006754303	0.015974224
TotalCharge	0.024100265	0.016875738	-0.014345109	-0.001403277	-0.005043103	0.016924046
Additional_charges	0.013548457	0.716853618	-0.009824976	0.008289993	0.008071608	0.010327340

The values of the covariance matrix are just the eigen values for the factors. The sum of the diagonals must equal the number of factors. There are 9 factors and so the sum of the diagonals must be equal to 9. (Variance and Eigenvalues : AnnMaria’s Blog, 2015)

In [11]:

```
sum(diag(S))
```

```
9
```

In [12]:

```
s.eigen <- eigen(S)
```

In [13]:

```
s.eigen
```

```
eigen() decomposition
$values
[1] 1.99400120 1.71397311 1.01722679 1.01334167 1.00416540 0.99185499 0.97004965
[8] 0.28366996 0.01171723

$vectors
              [,1]        [,2]         [,3]          [,4]         [,5]
 [1,] -0.034637405 -0.01714061  0.170642795 -0.4872777216 -0.541146755
 [2,] -0.087304472 -0.70097831 -0.007768503  0.0146013398 -0.011350334
 [3,]  0.020640127  0.01820389 -0.610809775 -0.3717083867 -0.388134005
 [4,]  0.001570476 -0.01954319  0.539897154 -0.4996783050  0.376656471
 [5,]  0.007234201 -0.01617596 -0.330906145 -0.5913827095  0.483676654
 [6,] -0.024779756 -0.01570502 -0.443355512  0.1569494183  0.424598769
 [7,] -0.700272823  0.09361444 -0.005980279 -0.0005454392  0.003972517
 [8,] -0.701449826  0.08303869 -0.004618832 -0.0023307447  0.005513263
 [9,] -0.087575357 -0.70103844 -0.011094321  0.0113723732 -0.015487393
             [,6]         [,7]          [,8]          [,9]
 [1,]  0.56131217  0.352167530  0.0037294642 -0.0009302585
 [2,] -0.02064614 -0.010685067  0.7066633769  0.0262943903
 [3,] -0.15886455 -0.558654086  0.0022016225  0.0013088884
 [4,]  0.11082004 -0.551625787 -0.0019284637 -0.0015317998
 [5,] -0.25488816  0.491467008  0.0012237570 -0.0011057201
 [6,]  0.76161154 -0.132685333  0.0003295861 -0.0005752266
 [7,] -0.02906538 -0.011945265  0.0316056194 -0.7062691177
 [8,] -0.02776358 -0.011013256 -0.0314695299  0.7064935359
 [9,] -0.01864509 -0.008322533 -0.7061257282 -0.0367823818
```

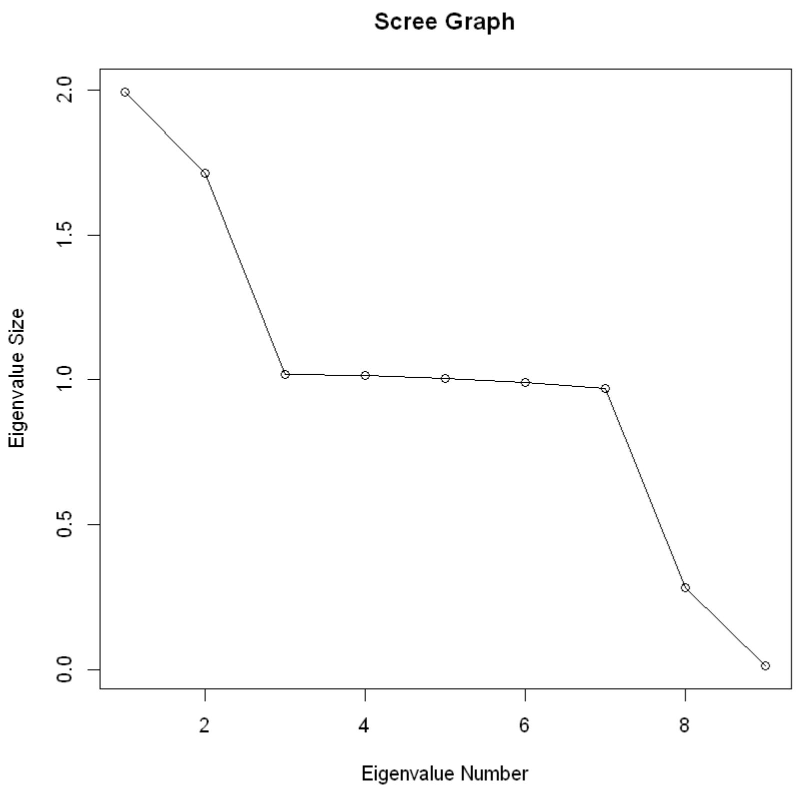
In [14]:

```
for (s in s.eigen$values) {
  print(s / sum(s.eigen$values))
}
```

```
[1] 0.2215557
[1] 0.1904415
[1] 0.1130252
[1] 0.1125935
[1] 0.1115739
[1] 0.1102061
[1] 0.1077833
[1] 0.03151888
[1] 0.001301915
```

From looking at the values of the normalized eginvalues it is clear that only the first 7 PCA's contribute in a meaningful way.

In [16]:
```
plot(s.eigen$values, xlab = 'Eigenvalue Number', ylab = 'Eigenvalue Size', main = 'Scre
lines(s.eigen$values)
```



From the Scree plot it is clear that the egin value drops below 1 once the number of PCA's goes above 7.

In [18]:
```
data.pca <- prcomp(data_scaled)
```

In [19]:
```
data.pca
```

```
Standard deviations (1, .., p=9):
[1] 1.4120911 1.3091880 1.0085766 1.0066487 1.0020805 0.9959192 0.9849110
[8] 0.5326068 0.1082462

Rotation (n x k) = (9 x 9):
                            PC1         PC2          PC3           PC4
Children            0.034637405 -0.01714061  0.170642795  0.4872777216
Age                 0.087304472 -0.70097831 -0.007768503 -0.0146013398
Income             -0.020640127  0.01820389 -0.610809775  0.3717083867
VitD_levels        -0.001570476 -0.01954319  0.539897154  0.4996783050
Doc_visits         -0.007234201 -0.01617596 -0.330906145  0.5913827095
vitD_supp           0.024779756 -0.01570502 -0.443355512 -0.1569494183
Initial_days        0.700272823  0.09361444 -0.005980279  0.0005454392
TotalCharge         0.701449826  0.08303869 -0.004618832  0.0023307447
Additional_charges  0.087575357 -0.70103844 -0.011094321 -0.0113723732
                            PC5         PC6          PC7           PC8
Children            0.541146755 -0.56131217  0.352167530 -0.0037294642
Age                 0.011350334  0.02064614 -0.010685067 -0.7066633769
Income              0.388134005  0.15886455 -0.558654086 -0.0022016225
VitD_levels        -0.376656471 -0.11082004 -0.551625787  0.0019284637
Doc_visits         -0.483676654  0.25488816  0.491467008 -0.0012237570
vitD_supp          -0.424598769 -0.76161154 -0.132685333 -0.0003295861
```

```
Initial_days       -0.003972517  0.02906538 -0.011945265 -0.0316056194
TotalCharge        -0.005513263  0.02776358 -0.011013256  0.0314695299
Additional_charges  0.015487393  0.01864509 -0.008322533  0.7061257282
                             PC9
Children            0.0009302585
Age                -0.0262943903
Income             -0.0013088884
VitD_levels         0.0015317998
Doc_visits          0.0011057201
vitD_supp           0.0005752266
Initial_days        0.7062691177
TotalCharge        -0.7064935359
Additional_charges  0.0367823818
```

In [43]:

```
summary(data.pca)
```

```
Importance of components:
                          PC1    PC2   PC3    PC4    PC5    PC6    PC7     PC8
Standard deviation     1.4121 1.3092 1.009 1.0066 1.0021 0.9959 0.9849 0.53261
Proportion of Variance 0.2216 0.1904 0.113 0.1126 0.1116 0.1102 0.1078 0.03152
Cumulative Proportion  0.2216 0.4120 0.525 0.6376 0.7492 0.8594 0.9672 0.99870
                          PC9
Standard deviation     0.1082
Proportion of Variance 0.0013
Cumulative Proportion  1.0000
```

The first seven components accout for 97% of the total variation.

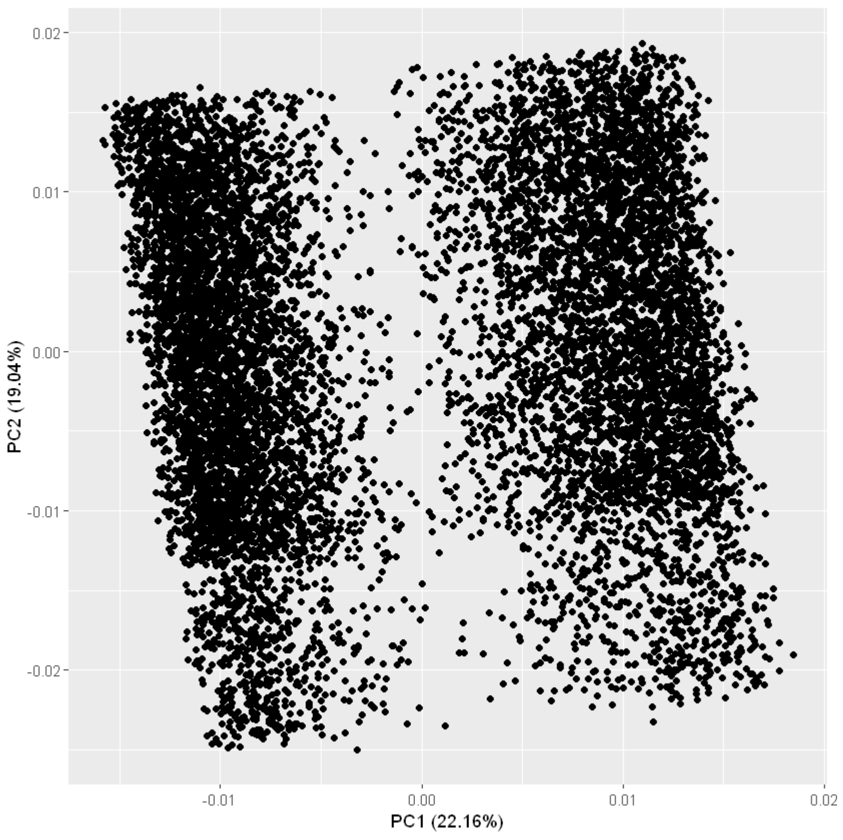
In [20]:

```
pca.plot <- autoplot(data.pca, data = data_scaled)
```

```
Warning message:
"`select_()` is deprecated as of dplyr 0.7.0.
Please use `select()` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_warnings()` to see where this warning was generated."
```

In [21]:

```
pca.plot
```



Next calculate the mean of the components

In [26]:
```
data.pca$center
```

Children: 3.58948981649121e-17 **Age:** 1.17302695334942e-16 **Income:** 5.46927954314658e-17 **VitD_levels:** -8.04464741349653e-16 **Doc_visits:** -2.37310171513627e-19 **vitD_supp:** 1.0502709812954e-17 **Initial_days:** 9.83477188576387e-17 **TotalCharge:** 9.49747225309494e-17 **Additional_charges:** -2.25928650787743e-17

Next calculate the standard deviation of the components

In [28]:
```
std_dev <- data.pca$sdev
```

Now that the standard deviation has been found it is possible to calculate the variance of each component

In [29]:
```
data_var <- std_dev^2
```

In [30]:
```
data_var
```

1.99400120064886 · 1.71397310505695 · 1.01722678823847 · 1.01334166547608 · 1.00416539916782 · 0.99185498989761 · 0.970049654335241 · 0.283669962891095 · 0.0117172342878669

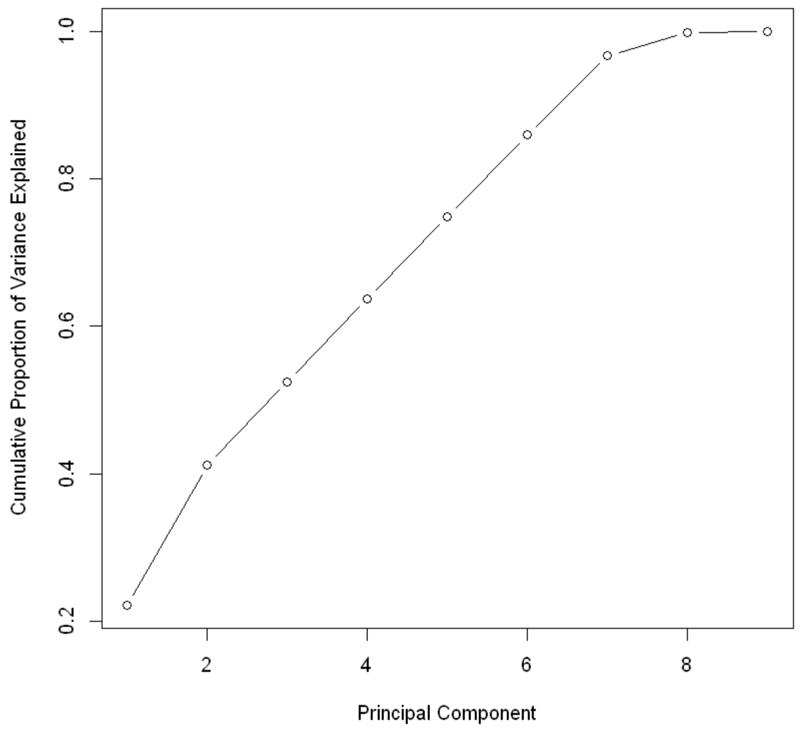
To determine how many components must be retained it is important to look at the proportion of variance explained by each component

In [31]:
```
prop_var <- data_var/sum(data_var)
```

In [32]:
```
prop_var
```

```
0.221555688960984 · 0.190441456117439 · 0.113025198693164 · 0.112593518386232 ·
0.111573933240869 · 0.110206109988623 · 0.107783294926138 · 0.0315188847656772 ·
0.0013019149208741
```

In [33]:
```
plot(cumsum(prop_var), xlab = "Principal Component",
              ylab = "Cumulative Proportion of Variance Explained",
              type = "b")
```



Conclusion:

The first two components are the most important. The most important factors in the first two PCA's are Initial Days, Total Charge, and Age. Age is negative which shows that surprisingly older patients are less likely to be readmitted. One reason for this could be the older patients tend to be have worse insurance which is less likely to approve additional hospital visits. Initial days is the number of days that the patient was hospitalized during their initial

visit. Patients with long initial stays are far more likely to be dealing with a serious medical condition, and hence need to be readmitted. Patients with a high total charge are also more likely to be suffering from a serious ailment and therefore need to be readmitted.

References

Vidhya, A. (2020, February 6). PCA: A Practical Guide to Principal Component Analysis in R & Python. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/

Vadapalli, P. (2020, November 12). PCA in Machine Learning: Assumptions, Steps to Apply & Applications. UpGrad Blog. https://www.upgrad.com/blog/pca-in-machine-learning/#Assumptions_in_PCA

Variance and Eigenvalues : AnnMaria’s Blog. (2015, September 6). Thejuliagroup. https://www.thejuliagroup.com/blog/factor-analysis-and-eigenvalues/

O’Rourke, N. (2020). A Step-by-Step Approach to Using SASÂ® for Univariate & Multivariate Statistics, Second Edition. O’Reilly Online Learning. https://www.oreilly.com/library/view/a-step-by-step-approach/9781590474174/9781590474174_ch15lev1sec7.html