Homework 1. PCA. (60 Points)

Drashti Thummar

2023-02-23

Part 1. PCA vs Linear Regression (6 points).

Let's say we have two 'features': let one be x and another y. Recall that in linear regression, we are looking to get a model like:

$$y_i = \beta_0 + \beta_1 * x_i + \varepsilon_i$$

after the fitting, for each data point we would have:

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 * x_i + r_i$$

where r_i is residual. It can be rewritten as:

$$\hat{\beta}_0 + r_i = y_i - \hat{\beta}_1 * x_i \quad (1)$$

The first principal component z_1 calculated on (x, y) is

$$z_{i1} = \phi_{i1} y_i + \phi_{i1} x_i$$

Dividing it by ϕ_{i1}:

$$\frac{z_{i1}}{\phi_{i1}} = y_i + \frac{\phi_{i1}}{\phi_{i1}} x_i \quad (2)$$

There is a functional resemblance between equations (1) and (2) (described linear relationship between y and x). Is the following true:

$$\hat{\beta}_0 + r_i = \frac{z_{i1}}{\phi_{i1}}$$

$$\frac{\phi_{i1}}{\phi_{i1}} = -\hat{\beta}_1$$

Answer: *No*

What is the difference between linear regression coefficients optimization and first PCA calculations?

Answer: *(here should be the answer. help yourself with a plot)*

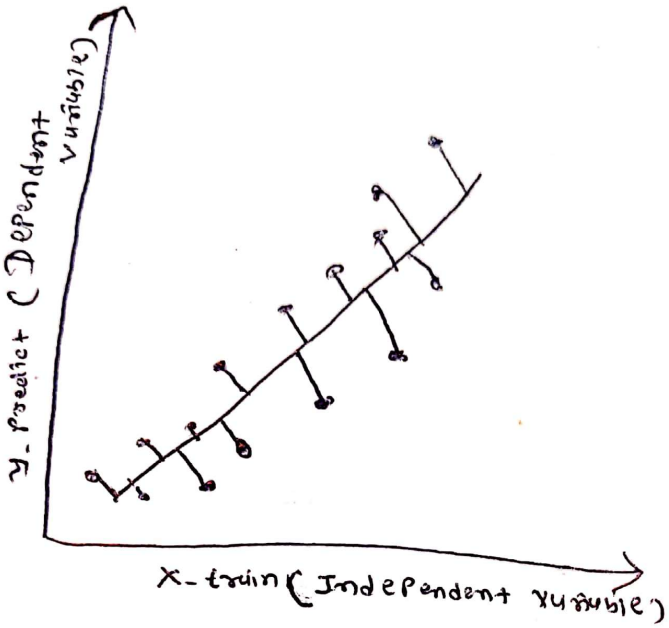
Linear regression is used to find the best-fit line that describes the relationship between the variables. it is calculated by the values of the regression coefficients, which represent the slope and intercept of the line. The coefficients are calculated using the method of least squares, which minimizes the sum of the squared differences between the observed and predicted values.

While PCA is a technique used to reduce the dimensionality of a dataset by identifying the principal components that explain the most variance in the data. The goal of PCA is to transform the original data into a new set of variables that are uncorrelated and have fewer dimensions.

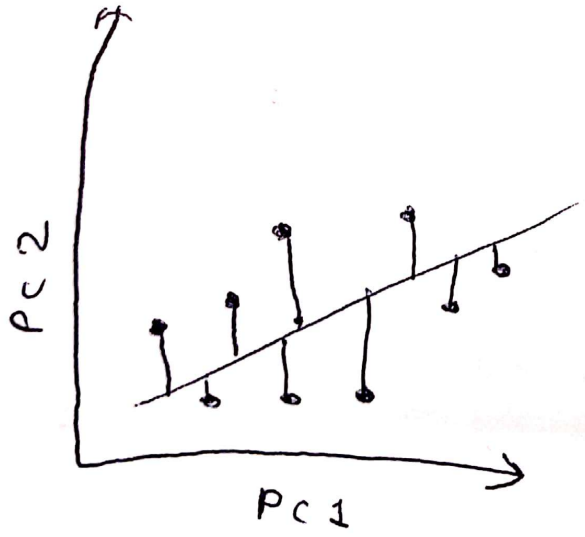
In linear regression, the coefficients are calculated using the method of least squares and mostly used to minimize sum of squared error between actual output y and predicted y values. In PCA, the principal components are calculated using eigenvectors and eigenvalues of the covariance matrix of the original data.it focuses on minimizing the dimensionality of data, but still maintaining real important features of data. Below are graphs, showing uses these two different terms:

```
knitr::include_graphics("pca_lin.jpg")
```

Linear regression



PCA



Part 2. PCA Exercise (27 points).

In this exercise we will study UK Smoking Data (`smoking.R`, `smoking.rda` or `smoking.csv`):

Description

Survey data on smoking habits from the UK. The data set can be used for analyzing the demographic characteristics of smokers and types of tobacco consumed.

Format

A data frame with 1691 observations on the following 12 variables.

`gender` - Gender with levels Female and Male.

`age` - Age.

`marital_status` - Marital status with levels Divorced, Married, Separated, Single and Widowed.

`highest_qualification` - Highest education level with levels A Levels, Degree, GCSE/CSE, GCSE/O Level, Higher/Sub Degree, No Qualification, ONC/BTEC and Other/Sub Degree

`nationality` - Nationality with levels British, English, Irish, Scottish, Welsh, Other, Refused and Unknown.

`ethnicity` - Ethnicity with levels Asian, Black, Chinese, Mixed, White and Refused Unknown.

`gross_income` - Gross income with levels Under 2,600, 2,600 to 5,200, 5,200 to 10,400, 10,400 to 15,600, 15,600 to 20,800, 20,800 to 28,600, 28,600 to 36,400, Above 36,400, Refused and Unknown.

`region` - Region with levels London, Midlands & East Anglia, Scotland, South East, South West, The North and Wales

`smoke` - Smoking status with levels No and Yes

`amt_weekends` - Number of cigarettes smoked per day on weekends.

`amt_weekdays` - Number of cigarettes smoked per day on weekdays.

`type` - Type of cigarettes smoked with levels Packets, Hand-Rolled, Both/Mainly Packets and Both/Mainly Hand-Rolled

Source National STEM Centre, Large Datasets from stats4schools, https://www.stem.org.uk/resources/elibrary/resource/28452/large-datasets-stats4schools.

Obtained from https://www.openintro.org/data/index.php?data=smoking

Read and Clean the Data

2.1 Read the data from smoking.R or smoking.rda (3 points) > hint: take a look at source or load functions > there is also smoking.csv file for a refference

```
# load libraries
if (!require("tibble")) install.packages("tibble")
if (!require("readr")) install.packages("readr")
if (!require("broom")) install.packages("broom")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("ISLR2")) install.packages("ISLR2")
if (!require("ggbiplot")) {
  if (!require("devtools")) install.packages("devtools")
  devtools::install_github("vqv/ggbiplot")
}
```

```
library(tibble)
library(readr)
library(dplyr)

library(broom)
library(cowplot)

library(ggplot2)
library(ggbiplot)
library(fastDummies)
library(tidyverse)
library(plotly)
```

```
# Load data
data <- load("smoking.rda")
```

Take a look into data

```
# place holder
glimpse(smoking)
```

```
## Rows: 1,691
## Columns: 12
## $ gender                <fct> Male, Female, Male, Female, Female, Female, Male~
## $ age                   <int> 38, 42, 40, 40, 39, 37, 53, 44, 40, 41, 72, 49, ~
## $ marital_status        <fct> Divorced, Single, Married, Married, Married, Mar~
## $ highest_qualification <fct> No Qualification, No Qualification, Degree, Degr~
## $ nationality           <fct> British, British, English, English, British, Bri~
## $ ethnicity             <fct> White, White, White, White, White, White, White,~
## $ gross_income          <fct> "2,600 to 5,200", "Under 2,600", "28,600 to 36,4~
## $ region                <fct> The North, The North, The North, The North, The ~
## $ smoke                 <fct> No, Yes, No, No, No, No, Yes, No, Yes, Yes, No, ~
## $ amt_weekends          <int> NA, 12, NA, NA, NA, NA, 6, NA, 8, 15, NA, NA, NA~
## $ amt_weekdays          <int> NA, 12, NA, NA, NA, NA, 6, NA, 8, 12, NA, NA, NA~
## $ type                  <fct> , Packets, , , , , Packets, , Hand-Rolled, Packe~
```

There are many fields there so for this exercise lets only concentrate on smoke, gender, age, marital_status, highest_qualification and gross_income.

Create new data.frame with only these columns.

```
# place holder
df = smoking %>%  dplyr::select(smoke, gender, age, marital_status,
                                highest_qualification, gross_income)
glimpse(df)
```

```
## Rows: 1,691
## Columns: 6
## $ smoke                 <fct> No, Yes, No, No, No, No, Yes, No, Yes, Yes, No, ~
## $ gender                <fct> Male, Female, Male, Female, Female, Female, Male~
## $ age                   <int> 38, 42, 40, 40, 39, 37, 53, 44, 40, 41, 72, 49, ~
## $ marital_status        <fct> Divorced, Single, Married, Married, Married, Mar~
## $ highest_qualification <fct> No Qualification, No Qualification, Degree, Degr~
## $ gross_income          <fct> "2,600 to 5,200", "Under 2,600", "28,600 to 36,4~
```

2.2 Omit all incomplete records.(3 points)

```
# place holder
df$gross_income[df$gross_income %in% c("Unknown", "Refused")] <- NA

# Remove rows with NA values in dataframe
df <- na.omit(df)
df
```

```
## # A tibble: 1,565 x 6
##    smoke gender   age marital_status highest_qualification gross_income
##    <fct> <fct>  <int> <fct>          <fct>                 <fct>
##  1 No    Male      38 Divorced       No Qualification      2,600 to 5,200
##  2 Yes   Female    42 Single         No Qualification      Under 2,600
##  3 No    Male      40 Married        Degree                28,600 to 36,400
##  4 No    Female    40 Married        Degree                10,400 to 15,600
##  5 No    Female    39 Married        GCSE/O Level          2,600 to 5,200
##  6 No    Female    37 Married        GCSE/O Level          15,600 to 20,800
##  7 Yes   Male      53 Married        Degree                Above 36,400
##  8 No    Male      44 Single         Degree                10,400 to 15,600
##  9 Yes   Male      40 Single         GCSE/CSE              2,600 to 5,200
## 10 Yes   Female    41 Married        No Qualification      5,200 to 10,400
## # ... with 1,555 more rows
```

2.3 For PCA feature should be numeric. Some of fields are binary (`gender` and `smoke`) and can easily be converted to numeric type (with one and zero). Other fields like `marital_status` has more than two categories, convert them to binary (i.e. is_married, is_devorced). Several features in the data set are ordinal (`gross_income` and `highest_qualification`), convert them to some king of sensible level (note that levels in factors are not in order). (3 points)

```
# place holder
df$smoke <- as.numeric(df$smoke == "No")
df$gender <- ifelse(df$gender == "Male", 1, 0)

# create new column with 1s and 0s
df$is_married <- ifelse(df$marital_status == "Married", 1, 0)
df$is_Divorced <- ifelse(df$marital_status == "Divorced", 1, 0)
df$is_Single <- ifelse(df$marital_status == "Single", 1, 0)
df$is_Widowed <- ifelse(df$marital_status == "Widowed", 1, 0)

# remove original column
df <- df[, -which(names(df) == "marital_status")]

category_map <- c(`No Qualification`=0, `GCSE/O Level`=1, `GCSE/CSE`=1,
                  `Other/Sub Degree`=1, `ONC/BTEC`=1, `A Levels`=1,
                  `Higher/Sub Degree` =1, Degree=2)

# Use mutate function from dplyr to add new column with mapped numeric values
df <- df %>%
 dplyr:: mutate(highest_qualification = category_map[highest_qualification])

df$gross_income <- factor(df$gross_income, levels = c('Under 2,600', '2,600 to 5,200',
                                                      '5,200 to 10,400', '10,400 to 15,600',
```

```
                                                    '15,600 to 20,800', '20,800 to 28,600',
                                                    '28,600 to 36,400', 'Above 36,400'))

# Map the categorical values to sequential values using as.numeric()
df$gross_income <- as.numeric(df$gross_income)

# Print the modified data frame
df
```

```
## # A tibble: 1,565 x 9
##    smoke gender   age highest_qualific~1 gross~2 is_ma~3 is_Di~4 is_Si~5 is_Wi~6
##    <dbl>  <dbl> <int>              <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1     1      1    38                  1       2       0       1       0       0
##  2     0      0    42                  1       1       0       0       1       0
##  3     1      1    40                  1       7       1       0       0       0
##  4     1      0    40                  1       4       1       0       0       0
##  5     1      0    39                  1       2       1       0       0       0
##  6     1      0    37                  1       5       1       0       0       0
##  7     0      1    53                  1       8       1       0       0       0
##  8     1      1    44                  1       4       0       0       1       0
##  9     0      1    40                  1       2       0       0       1       0
## 10     0      0    41                  1       3       1       0       0       0
## # ... with 1,555 more rows, and abbreviated variable names
## #   1: highest_qualification, 2: gross_income, 3: is_married, 4: is_Divorced,
## #   5: is_Single, 6: is_Widowed
```

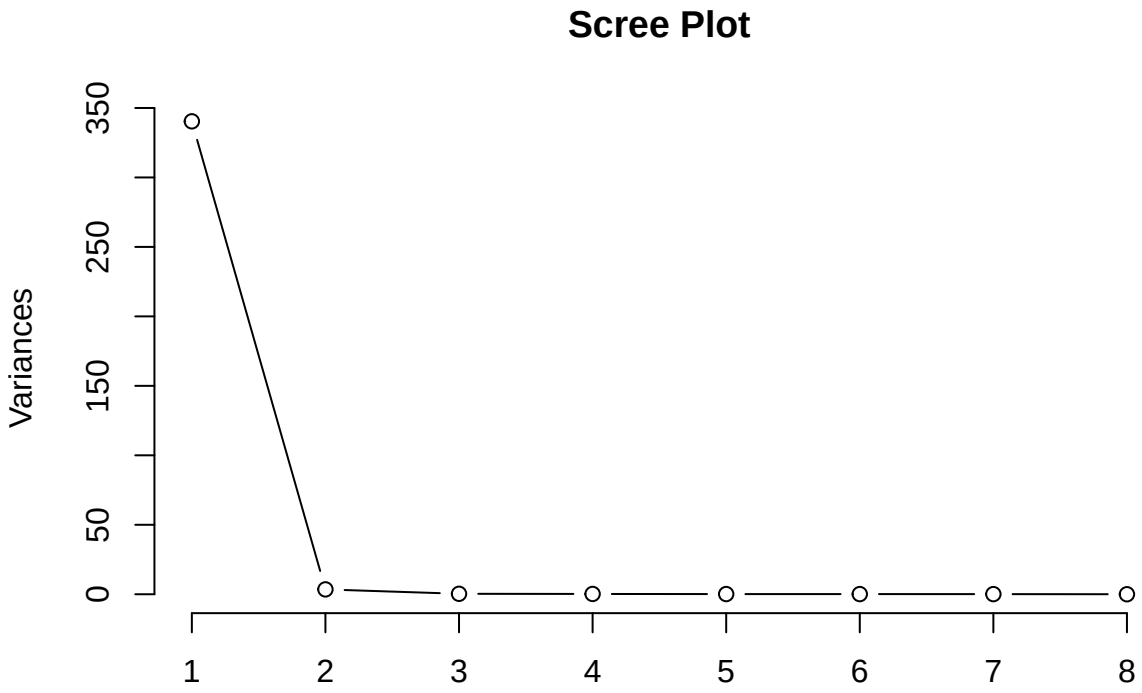
2.4. Do PCA on all columns except smoking status. (3 points)

```
# place holder
rectified_df <- df[ , -1]
pca <- prcomp(rectified_df)
summary(pca)
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     18.4491 1.87039 0.57343 0.46478 0.35984 0.34359 0.30570
## Proportion of Variance  0.9873 0.01015 0.00095 0.00063 0.00038 0.00034 0.00027
## Cumulative Proportion   0.9873 0.99740 0.99836 0.99898 0.99936 0.99970 0.99997
##                            PC8
## Standard deviation     0.09894
## Proportion of Variance 0.00003
## Cumulative Proportion  1.00000
```

2.5 Make a scree plot (3 points)

```
# place holder
plot(pca, type = "l", main = "Scree Plot")
```

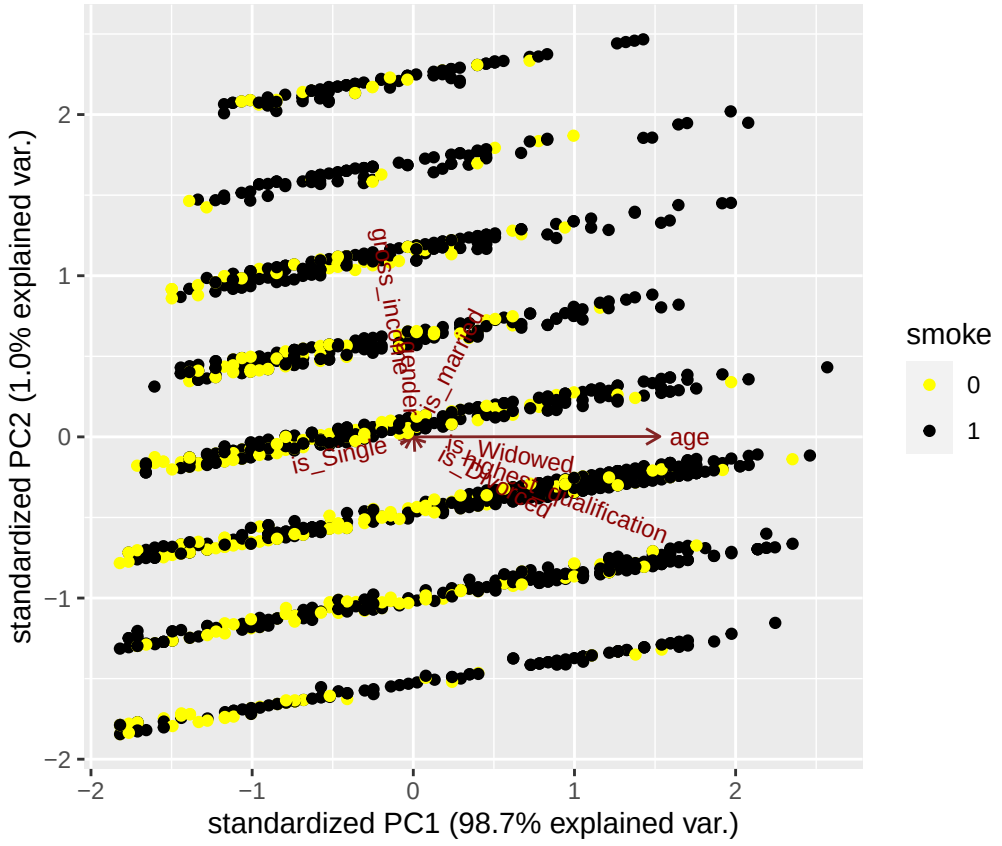


Comment on the shape, if you need to reduce dimensions home many would you choose

From the given scree plot, it can be deduced that I need to reduce dimension by 8 features. From the cumulative proportion, it can be seen that 98.67% data is covered by first PC only. So,it would be palusible to abandon the rest features.

2.6 Make a biplot color points by smoking field. (3 points)

```
# place holder
ggbiplot(pca, groups = as.factor(df$smoke),
         var.axes = TRUE) +

  # Map the 'smoke' column values to color aesthetic
  aes(color = smoke) +

  # Define the color scale
  scale_color_manual(values = c("yellow", "black"))
```



Comment on observed biplot.

--> It can be seen from the biplot that data points are divided into 8 different size of clusters very separate to eachother. variables like groass_income,gender, is_married are relying on PC1,while highest_qualification, is_widowed, is_divorced etc are highly correlated to eachother because angle between them are really small, their value lie on PC2 axis. age attribute is making major impacts on classifying smoke data. I have arranged smoke attribute value to yellow color so it can be easily notify.

Can we use first two PC to discriminate smoking?

--> From the biplot, It can be stated that smoking status can not be discriminated by first two PC, because data points are not linearly separable or easily classsified.

2.7 Based on the loading vector can we name PC with some descriptive name? (3 points)

--> In this case, it can be visualized from the graph that marriage status related datapoints are heavily dependent upon PC1. so we can describe PC1 as marital_status at some level.

2.8 May be some of splits between categories or mapping to numerics should be revisited, if so what will you do differently? (3 points)

--> If I need to change some categorical values, then I will different values to all highest_qualification criteria to be more specific, rather than assigning same values to group of values.

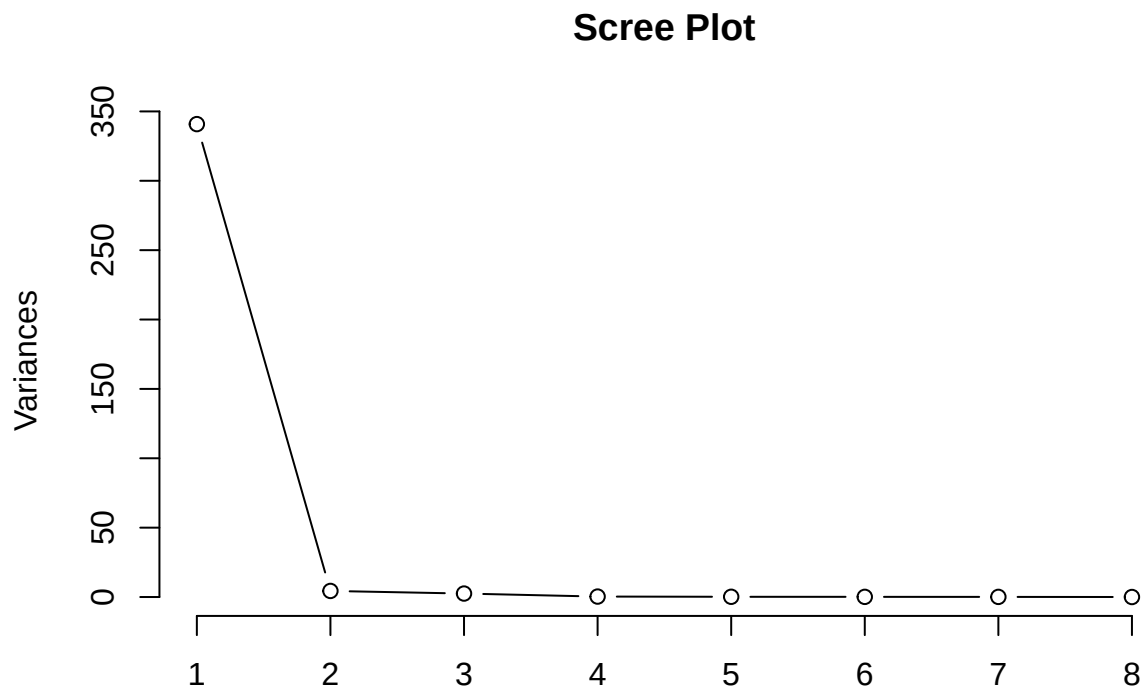
2.9 Follow your suggestion in 2.10 and redo PCA and biplot (3 points)

```
# place holder
new_df = smoking %>%  dplyr::select(smoke, gender, age, marital_status,
                                highest_qualification, gross_income)
new_df$gross_income[new_df$gross_income %in% c("Unknown", "Refused")] <- NA

# Remove rows with NA values in dataframe
new_df <- na.omit(new_df)
new_df$smoke <- as.numeric(new_df$smoke == "No")
new_df$gender <- ifelse(new_df$gender == "Male", 1, 0)

# create new column with 1s and 0s
new_df$is_married <- ifelse(new_df$marital_status == "Married", 1, 0)
new_df$is_Divorced <- ifelse(new_df$marital_status == "Divorced", 1, 0)
new_df$is_Single <- ifelse(new_df$marital_status == "Single", 1, 0)
new_df$is_Widowed <- ifelse(new_df$marital_status == "Widowed", 1, 0)

# remove original column
new_df <- new_df[, -which(names(new_df) == "marital_status")]

category_map <- c(`No Qualification`=0, `GCSE/O Level`=1, `GCSE/CSE`=2,
                  `Other/Sub Degree`=3, `ONC/BTEC`=4, `A Levels`=5,
                  `Higher/Sub Degree` =6, Degree=7)

# Use mutate function from dplyr to add new column with mapped numeric values
new_df <- new_df %>%
 dplyr:: mutate(highest_qualification = category_map[highest_qualification])

new_df$gross_income <- factor(new_df$gross_income, levels = c('Under 2,600', '2,600 to 5,200',
                                                              '5,200 to 10,400', '10,400 to 15,600',
                                                              '15,600 to 20,800', '20,800 to 28,600',
                                                              '28,600 to 36,400', 'Above 36,400'))

# Map the categorical values to sequential values using as.numeric()
new_df$gross_income <- as.numeric(new_df$gross_income)
```

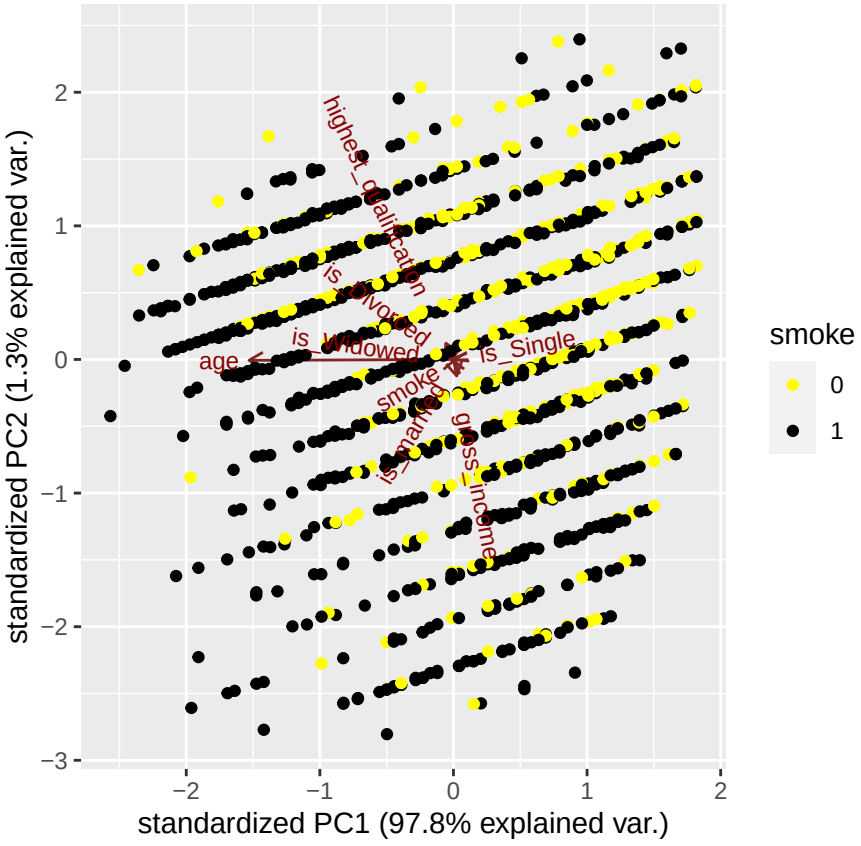
```
df_rec <- new_df[ , -2]
pca3 <- prcomp(df_rec)
summary(pca3)
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4    PC5     PC6     PC7
## Standard deviation     18.4618 2.09225 1.58231 0.57797 0.4154 0.35055 0.30700
## Proportion of Variance  0.9781 0.01256 0.00719 0.00096 0.0005 0.00035 0.00027
## Cumulative Proportion   0.9781 0.99071 0.99789 0.99885 0.9993 0.99970 0.99997
##                            PC8
## Standard deviation     0.09899
## Proportion of Variance 0.00003
## Cumulative Proportion  1.00000
```

```r
plot(pca3, type = "l", main = "Scree Plot")
```




```r
ggbiplot(pca3, groups = as.factor(df$smoke),
        var.axes = TRUE) +
        aes(color = smoke) +
        scale_color_manual(values = c("yellow", "black"))
```



Part 3. Freestyle. (27 points).

Get the data set from your final project (or find something suitable). The data set should have at least four variables and it shouldn't be used in class PCA examples: iris, mpg, diamonds and so on).

- Convert a columns to proper format (9 points)
- Perform PCA (3 points)
- Make a skree plot (3 points)
- Make a biplot (3 points)
- Discuss your observations (9 points)

```
data = read.csv('stroke_data.csv')
glimpse(data)
```

```
## Rows: 5,110
## Columns: 12
## $ id                <int> 9046, 51676, 31112, 60182, 1665, 56669, 53882, 10434~
## $ gender            <chr> "Male", "Female", "Male", "Female", "Female", "Male"~
## $ age               <dbl> 67, 61, 80, 49, 79, 81, 74, 69, 59, 78, 81, 61, 54, ~
## $ hypertension      <int> 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1~
## $ heart_disease     <int> 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0~
## $ ever_married      <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "No~
## $ work_type         <chr> "Private", "Self-employed", "Private", "Private", "S~
```

```
## $ Residence_type    <chr> "Urban", "Rural", "Rural", "Urban", "Rural", "Urban"~
## $ avg_glucose_level <dbl> 228.69, 202.21, 105.92, 171.23, 174.12, 186.21, 70.0~
## $ bmi               <chr> "36.6", "N/A", "32.5", "34.4", "24", "29", "27.4", "~
## $ smoking_status    <chr> "formerly smoked", "never smoked", "never smoked", "~
## $ stroke            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
```

```
dframe = data %>%  dplyr::select(gender, age, hypertension, heart_disease,
                                 ever_married, work_type, Residence_type,
                                 avg_glucose_level, bmi, smoking_status, stroke)
glimpse(dframe)
```

```
## Rows: 5,110
## Columns: 11
## $ gender            <chr> "Male", "Female", "Male", "Female", "Female", "Male"~
## $ age               <dbl> 67, 61, 80, 49, 79, 81, 74, 69, 59, 78, 81, 61, 54, ~
## $ hypertension      <int> 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1~
## $ heart_disease     <int> 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0~
## $ ever_married      <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "No~
## $ work_type         <chr> "Private", "Self-employed", "Private", "Private", "S~
## $ Residence_type    <chr> "Urban", "Rural", "Rural", "Urban", "Rural", "Urban"~
## $ avg_glucose_level <dbl> 228.69, 202.21, 105.92, 171.23, 174.12, 186.21, 70.0~
## $ bmi               <chr> "36.6", "N/A", "32.5", "34.4", "24", "29", "27.4", "~
## $ smoking_status    <chr> "formerly smoked", "never smoked", "never smoked", "~
## $ stroke            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
```

```
#omit null value
dframe <- dframe[- grep("N/A", dframe$bmi),]
```

```
#Convert a columns to proper format

dframe$gender <- ifelse(dframe$gender == "Male", 1, 0)

# create new column with 1s and 0s
dframe$ever_married <- ifelse(dframe$ever_married == "Yes", 1, 0)

category_map <- c("Private"=0, "Self-employed" = 1, "Govt_job"= 2,
                  "children" = 3, "Never_worked" = 4)
# Use mutate function from dplyr to add new column with mapped numeric values
dframe <- dframe %>% dplyr:: mutate(work_type = category_map[work_type])

dframe$is_rural <- ifelse(dframe$Residence_type == "Rural", 1, 0)
# remove original column
dframe <- dframe[, -which(names(dframe) == "Residence_type")]

dframe <- transform(dframe, bmi = as.numeric(bmi))

smoke_map <- c(`never smoked`=0, `formerly smoked`=1, `smokes`=2, `Unknown`=3)
# Use mutate function from dplyr to add new column with mapped numeric values
dframe <- dframe %>% dplyr:: mutate(smoking_status = smoke_map[smoking_status])
glimpse(dframe)
```

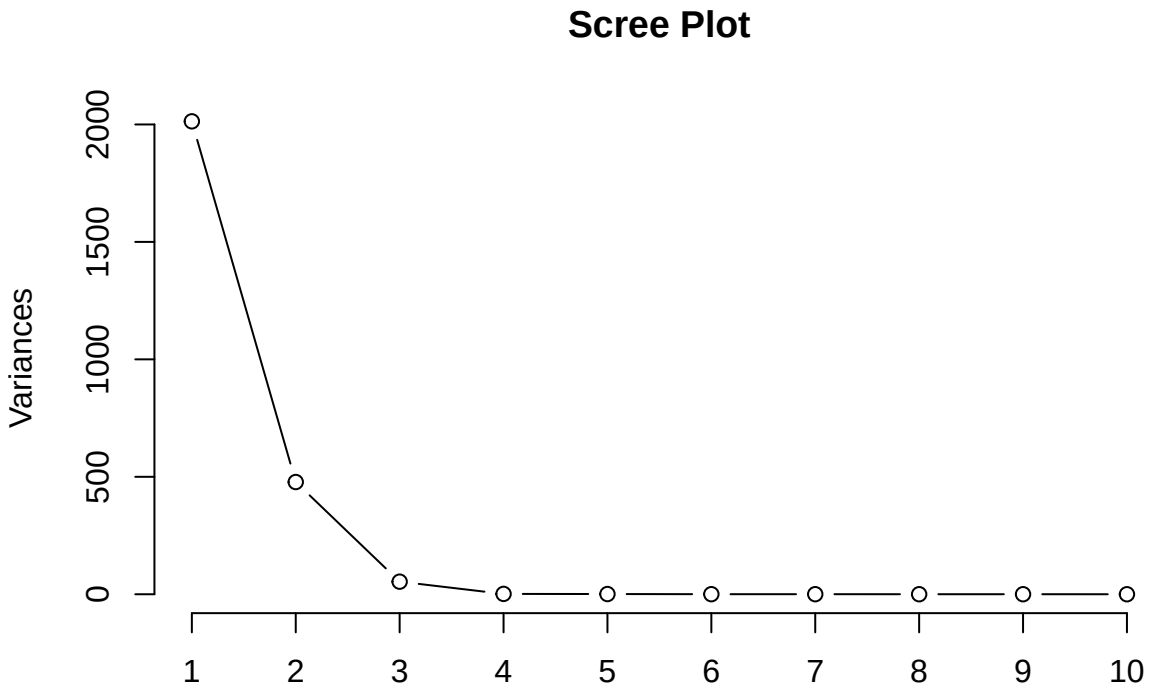
```
## Rows: 4,909
```

```
## Columns: 11
## $ gender            <dbl> 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0~
## $ age               <dbl> 67, 80, 49, 79, 81, 74, 69, 78, 81, 61, 54, 79, 50, ~
## $ hypertension      <int> 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1~
## $ heart_disease     <int> 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0~
## $ ever_married      <dbl> 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1~
## $ work_type         <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 2, 1~
## $ avg_glucose_level <dbl> 228.69, 105.92, 171.23, 174.12, 186.21, 70.09, 94.39~
## $ bmi               <dbl> 36.6, 32.5, 34.4, 24.0, 29.0, 27.4, 22.8, 24.2, 29.7~
## $ smoking_status    <dbl> 1, 0, 2, 0, 1, 0, 0, 3, 0, 2, 2, 0, 0, 2, 2, 0, 2, 0~
## $ stroke            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ is_rural          <dbl> 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0~
```

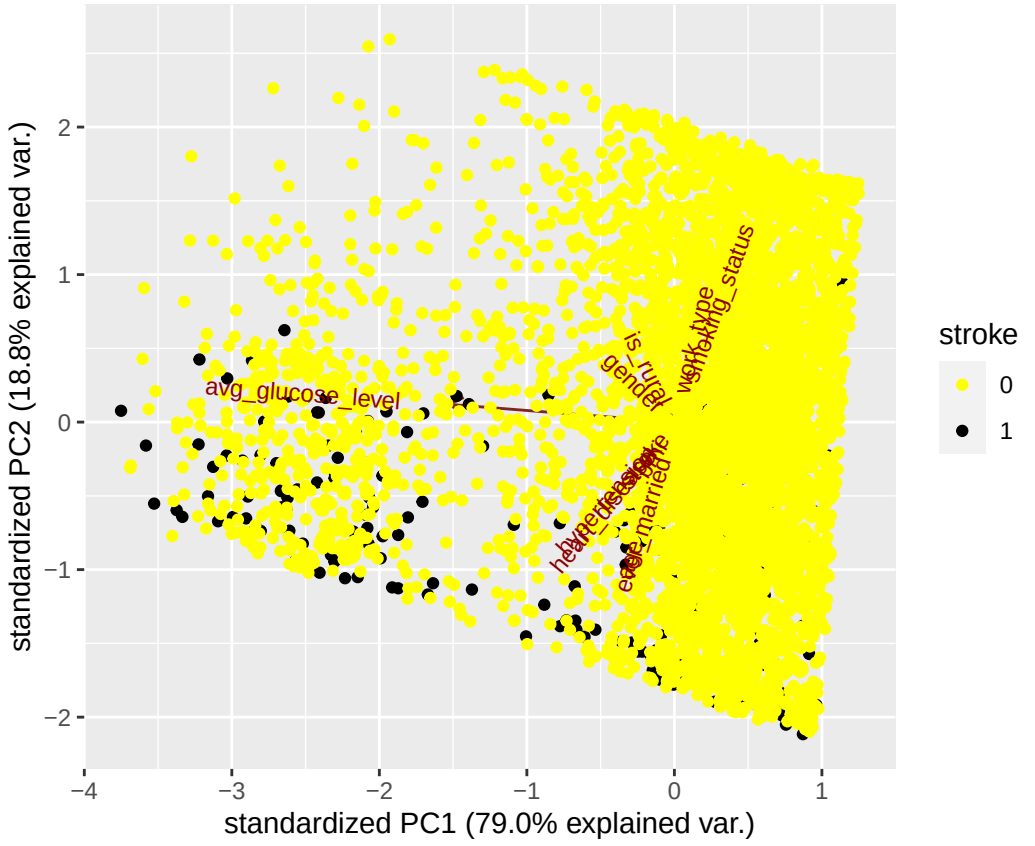
```
#Perform PCA
pca1 <- prcomp(dframe)
summary(pca1)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5    PC6     PC7
## Standard deviation     44.8691 21.8566 7.31991 1.22847 0.94411 0.5000 0.48854
## Proportion of Variance  0.7902  0.1875 0.02103 0.00059 0.00035 0.0001 0.00009
## Cumulative Proportion   0.7902  0.9777 0.99876 0.99935 0.99970 0.9998 0.99989
##                           PC8     PC9    PC10    PC11
## Standard deviation     0.34292 0.27418 0.20800 0.19119
## Proportion of Variance 0.00005 0.00003 0.00002 0.00001
## Cumulative Proportion  0.99994 0.99997 0.99999 1.00000
```

```
#Make a skree plot
plot(pca1, type = "l", main = "Scree Plot")
```




```
#Make a biplot
ggbiplot(pca1, groups = as.factor(dframe$stroke),
         var.axes = TRUE) +

  # Map the 'stroke' column values to color aesthetic
  aes(color = stroke) +

  # Define the color scale
  scale_color_manual(values = c("yellow", "black"))
```



#Discuss your observations
--> It can be deduced from the cumulative proportion that the elbow point appears to be at the third component. This suggests that there should retain the first three components, as they capture the majority of the variation in the data. The remaining components do not contribute significantly to the overall variability of the data and can be discarded.Its covering 99.87% principle components.

--> From biplot data, it can be observed that variables like work_type, smoking_status are highly correlated, while While, ever_married, hypertension, heart_disease are also correlated and are explicitly opposite of these. avg_glucose_level has significant impact upon making decision for prediction of stroke.

--> stroke datapoints can be easily classify into 2 categories so they are linearly separable.