Homework 1. PCA. (60 Points)

Zeming Zhang

2023-02-03

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:

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

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

$$z_{i1} = \phi_{i1} y_i + \phi_{i2} x_i$$

Dividing it by ϕ_{i1}:

$$\frac{z_{i1}}{\phi_{i1}} = y_i + \frac{\phi_{i2}}{\phi_{i1}} x_i \qquad (2)$$

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

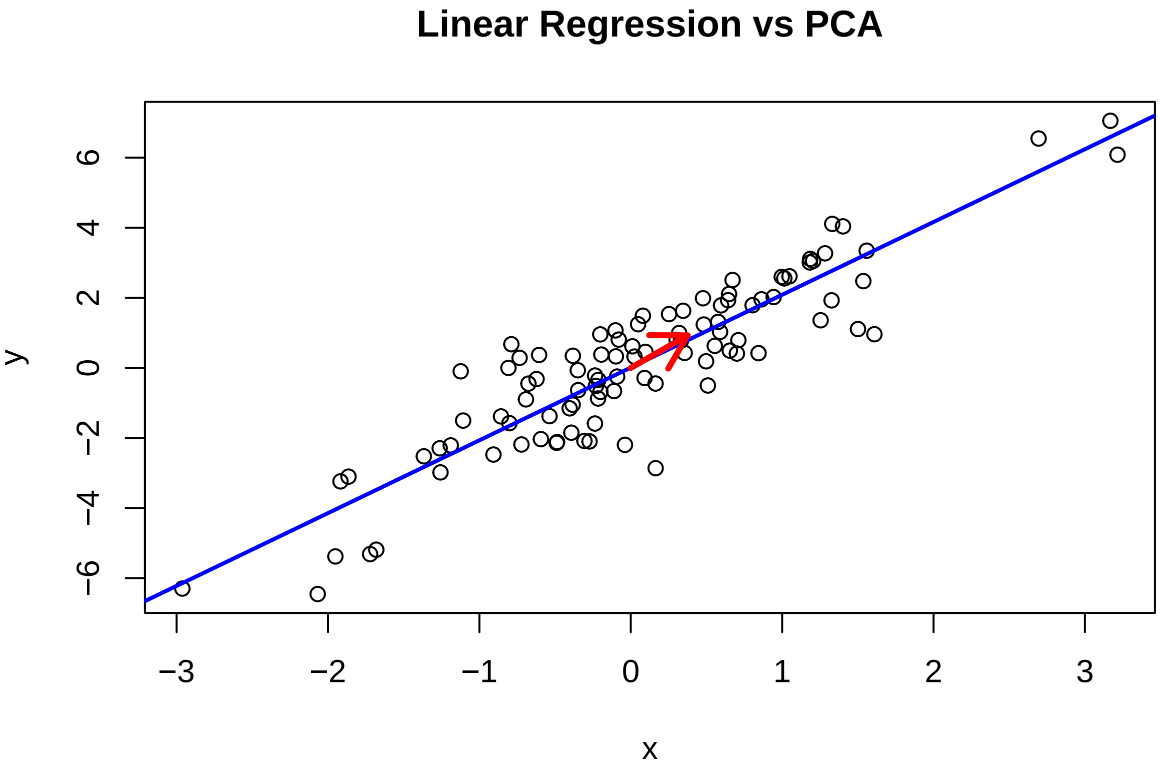
$$\hat{\beta}_0 + r_i = \frac{z_{i1}}{\phi_{i1}}$$

$$\frac{\phi_{i2}}{\phi_{i1}} = -\hat{\beta}_1$$

Answer: *(just yes or no)* No

What is the difference between linear regression coefficients optimization and first PCA calculations? **Answer**: Linear regression is used to find the best fit line or plane that explains the relationship between two or more variables, while PCA is used to reduce the dimensions of the data while retaining the maximum amount of information. Both techniques involve linear transformations of data, but they serve different purposes. Linear regression is used to predict the value of a dependent variable based on the values of one or more independent variables, while PCA is used to identify patterns or relationships between variables. *(here should be the answer. help yourself with a plot)*

```r
library(stats)

# generate some sample data
set.seed(0629)
x <- rnorm(100)
y <- 2*x + rnorm(100)

# perform linear regression
fit <- lm(y ~ x)

# perform PCA
data <- data.frame(x, y)
pca <- princomp(data)

# create a scatter plot of the data
plot(x, y, xlab = "x", ylab = "y", main = "Linear Regression vs PCA")
abline(fit, col = "blue", lwd = 2)
arrows(0, 0, pca$loadings[1,1], pca$loadings[1,2],
       col = "red", length = 0.2, lwd = 3)
```



The blue line shows the linear regression best fit line and the red arrows represent the principal components from PCA. Linear regression predicts y based on x, while PCA finds patterns and relationships in the underlying structure of the data.

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
library(dplyr)
library(tidyr)
library(ggplot2)
library(forcats)
```

```
# Load data
smoking_data <- data.frame(source("smoking.R"))
```

Take a look into data

```
#Show the first few rows of the data.
unique(smoking_data$value.smoke)
```

```
## [1] No  Yes
## Levels: No Yes
```

```
head(smoking_data)
```

```
##   value.gender value.age value.marital_status value.highest_qualification
## 1         Male        38             Divorced            No Qualification
## 2       Female        42               Single            No Qualification
## 3         Male        40              Married                      Degree
## 4       Female        40              Married                      Degree
## 5       Female        39              Married                GCSE/O Level
## 6       Female        37              Married                GCSE/O Level
##   value.nationality value.ethnicity value.gross_income value.region value.smoke
## 1           British           White     2,600 to 5,200    The North          No
## 2           British           White        Under 2,600    The North         Yes
## 3           English           White   28,600 to 36,400    The North          No
## 4           English           White   10,400 to 15,600    The North          No
## 5           British           White     2,600 to 5,200    The North          No
## 6           British           White   15,600 to 20,800    The North          No
##   value.amt_weekends value.amt_weekdays value.type visible
## 1                 NA                 NA              FALSE
## 2                 12                 12    Packets   FALSE
## 3                 NA                 NA              FALSE
## 4                 NA                 NA              FALSE
## 5                 NA                 NA              FALSE
## 6                 NA                 NA              FALSE
```

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.

```
# Create new data.frame with only mentioned columns
new_data = data.frame(smoking_data$value.smoke, smoking_data$value.gender,
                      smoking_data$value.age, smoking_data$value.marital_status,
                      smoking_data$value.highest_qualification,
                      smoking_data$value.gross_income)

head(new_data)
```

```
##   smoking_data.value.smoke smoking_data.value.gender smoking_data.value.age
## 1                       No                      Male                     38
## 2                      Yes                    Female                     42
## 3                       No                      Male                     40
## 4                       No                    Female                     40
## 5                       No                    Female                     39
## 6                       No                    Female                     37
##   smoking_data.value.marital_status smoking_data.value.highest_qualification
## 1                          Divorced                         No Qualification
## 2                            Single                         No Qualification
## 3                           Married                                   Degree
## 4                           Married                                   Degree
## 5                           Married                             GCSE/O Level
## 6                           Married                             GCSE/O Level
##   smoking_data.value.gross_income
## 1                  2,600 to 5,200
## 2                     Under 2,600
## 3                28,600 to 36,400
## 4                10,400 to 15,600
## 5                  2,600 to 5,200
## 6                15,600 to 20,800
```

2.2 Omit all incomplete records.(3 points)

```
# Omit all NA in records
new_data <- na.omit(new_data)
head(new_data)
```

```
##   smoking_data.value.smoke smoking_data.value.gender smoking_data.value.age
## 1                       No                      Male                     38
## 2                      Yes                    Female                     42
## 3                       No                      Male                     40
## 4                       No                    Female                     40
## 5                       No                    Female                     39
## 6                       No                    Female                     37
##   smoking_data.value.marital_status smoking_data.value.highest_qualification
## 1                          Divorced                         No Qualification
## 2                            Single                         No Qualification
## 3                           Married                                   Degree
## 4                           Married                                   Degree
## 5                           Married                             GCSE/O Level
## 6                           Married                             GCSE/O Level
##   smoking_data.value.gross_income
## 1                    2,600 to 5,200
## 2                       Under 2,600
## 3                  28,600 to 36,400
## 4                  10,400 to 15,600
## 5                    2,600 to 5,200
## 6                  15,600 to 20,800
```

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 kind of sensible level (note that levels in factors are not in order). (3 points)

```
colnames(new_data) <- gsub(".*\\.", "", colnames(new_data))

for (i in seq_along(unique(new_data$gross_income))) {
  cat(sprintf("Index %d: %s\n", i, unique(new_data$gross_income)[i]))
}
```

```
## Index 1: 2,600 to 5,200
## Index 2: Under 2,600
## Index 3: 28,600 to 36,400
## Index 4: 10,400 to 15,600
## Index 5: 15,600 to 20,800
## Index 6: Above 36,400
## Index 7: 5,200 to 10,400
## Index 8: Refused
## Index 9: 20,800 to 28,600
## Index 10: Unknown
```

```
# Convert smoke and gender to numeric binary
new_data$gender <- ifelse(new_data$gender == "Male", 1, 0)
new_data$smoke <- ifelse(new_data$smoke == "Yes", 1, 0)

# Replace "Degree" with 2, "no qualifications" with 0, and all other with 1
new_data$highest_qualification <- ifelse(
  new_data$highest_qualification == "Degree", 2,
  ifelse(new_data$highest_qualification == "No Qualification", 0, 1))

# Define a vector of income ranges in the desired order
income_ranges <- 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", "Refused", "Unknown")
# Define a vector of corresponding numerical levels
income_levels <- c(1, 2, 3, 4, 5, 6, 7, 8, -1, -1)

# Map income ranges to numerical levels in new_data$gross_income
new_data$gross_income <- match(new_data$gross_income, income_ranges)
new_data$gross_income <- ifelse(is.na(new_data$gross_income), -1,
                                income_levels[new_data$gross_income])

# This is for 2.9
new_data_copy <- data.frame(new_data)

# Create a new column for married status
new_data$is_married <- ifelse(new_data$marital_status == "Married", 1, 0)

# Create a new column for divorced status
new_data$is_divorced <- ifelse(new_data$marital_status == "Divorced", 1, 0)

# Create a new column for widowed status
new_data$is_widowed <- ifelse(new_data$marital_status == "Widowed", 1, 0)

# Create a new column for separated status
new_data$is_separated <- ifelse(new_data$marital_status == "Separated", 1, 0)

# Create a new column for single status
#(any status other than married, divorced, widowed, or separated)
new_data$is_single <- ifelse(!(new_data$is_married|new_data$is_divorced
                               |new_data$is_widowed|new_data$is_separated),1,0)

# Drop the original columns
new_data <- select(new_data, -marital_status)

head(new_data)
```

```
##   smoke gender age highest_qualification gross_income is_married is_divorced
## 1     0      1  38                     0            2          0           1
## 2     1      0  42                     0            1          0           0
## 3     0      1  40                     2            7          1           0
## 4     0      0  40                     2            4          1           0
## 5     0      0  39                     1            2          1           0
## 6     0      0  37                     1            5          1           0
##   is_widowed is_separated is_single
## 1          0            0         0
## 2          0            0         1
## 3          0            0         0
## 4          0            0         0
## 5          0            0         0
## 6          0            0         0
```

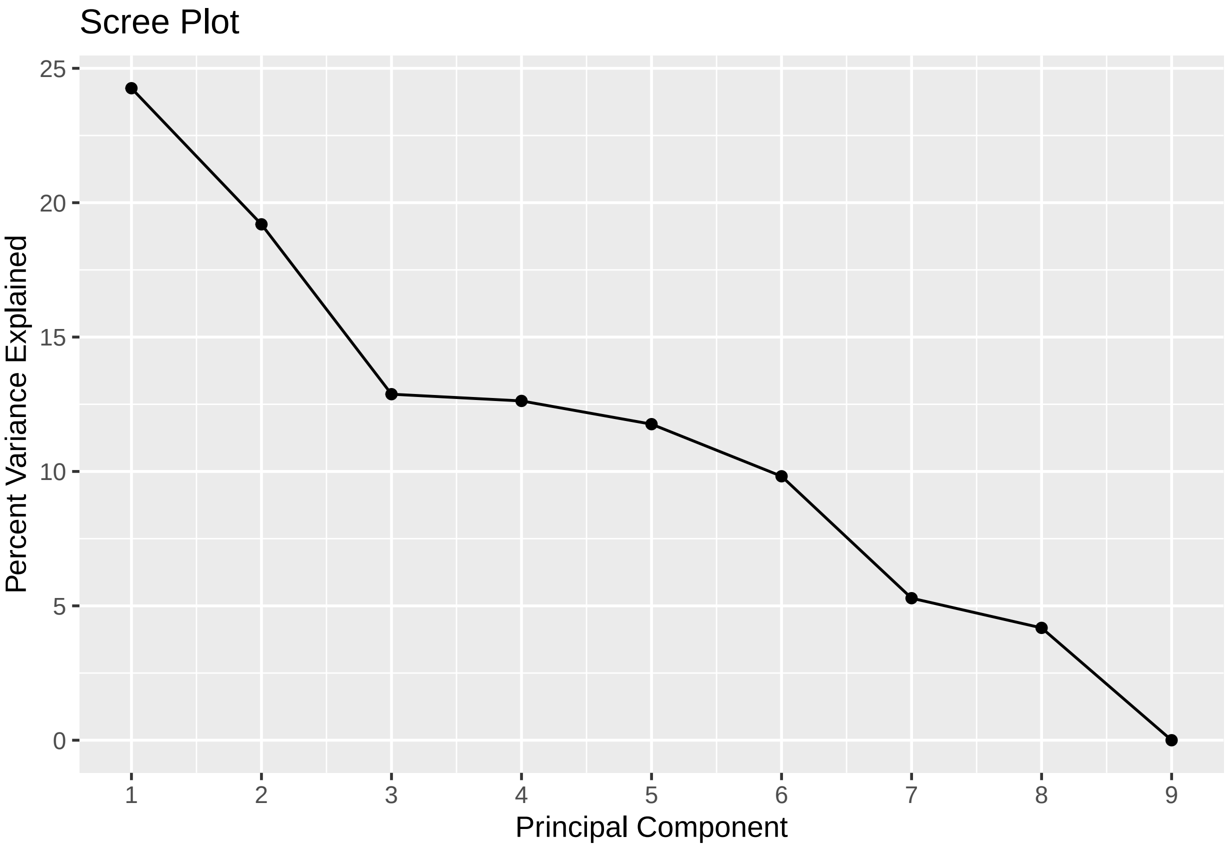
2.4. Do PCA on all columns except smoking status. (3 points)

```r
# Exclude the "smoking" column from the dataset
data_for_pca <- new_data[, !names(new_data) %in% "smoke"]

# Perform PCA on the remaining columns
pca_result <- prcomp(data_for_pca, scale. = TRUE)

# Print the result
pca_result
```

```
## Standard deviations (1, .., p=9):
## [1] 1.477607e+00 1.314356e+00 1.076449e+00 1.065939e+00 1.028782e+00
## [6] 9.401742e-01 6.896654e-01 6.133669e-01 1.160989e-15
## 
## Rotation (n x k) = (9 x 9):
##                               PC1         PC2         PC3         PC4
## gender                -0.19573193  0.23015895 -0.02768541 -0.57376753
## age                    0.54440642  0.18335325 -0.06775764 -0.22090189
## highest_qualification -0.46040792  0.11114273 -0.13740344 -0.06901914
## gross_income          -0.31340497  0.30243394 -0.34823441 -0.45542112
## is_married            -0.02184860  0.69055745  0.25908403  0.29062992
## is_divorced            0.05845678 -0.14469228 -0.77162136  0.26163778
## is_widowed             0.46301226 -0.17309258  0.05151860 -0.48789934
## is_separated          -0.01155953 -0.08194482 -0.29732346  0.01268427
## is_single             -0.36973505 -0.52450739  0.31767287 -0.13672746
##                                PC5          PC6         PC7         PC8
## gender                 0.148132625  0.632890826  0.37300554  0.13081664
## age                    0.042629258 -0.004615786  0.09170864 -0.77869472
## highest_qualification -0.026912948 -0.572873995  0.62242843 -0.18895083
## gross_income          -0.016975322 -0.234843442 -0.64964767 -0.06444947
## is_married             0.016931598  0.017083700 -0.02125042  0.04065764
## is_divorced            0.401621075  0.130343482  0.07408600  0.02965848
## is_widowed            -0.004467317 -0.385133493  0.10699775  0.43151280
## is_separated          -0.894373852  0.200386036  0.07346725 -0.04660084
## is_single              0.117116575  0.101627723 -0.14216493 -0.38178238
##                                 PC9
## gender                 6.295478e-16
## age                   -4.436479e-17
## highest_qualification -3.253586e-16
## gross_income           5.805936e-17
## is_married            -6.069428e-01
## is_divorced           -3.565611e-01
## is_widowed            -4.110463e-01
## is_separated          -2.386652e-01
## is_single             -5.277920e-01
```

2.5 Make a scree plot (3 points)

```r
# Extract the variance explained by each principal component
pca_var <- pca_result$sdev^2
pca_var_percent <- pca_var / sum(pca_var) * 100

# Create a data frame for the scree plot
scree_data <- data.frame(
  PC = 1:length(pca_var),
  Variance = pca_var_percent
)

# Create the scree plot
ggplot(scree_data, aes(x = PC, y = Variance)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:length(pca_var)) +
  labs(x = "Principal Component", y = "Percent Variance Explained",
       title = "Scree Plot")
```

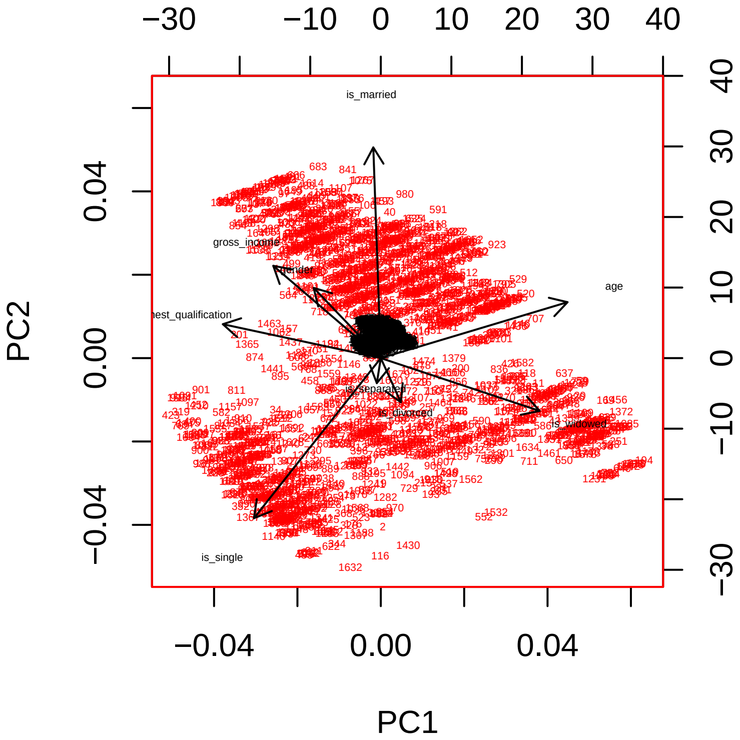


Comment on the shape, if you need to reduce dimensions home many would you choose

The elbow in the scree plot indicates the optimal number of components to retain. This point is where the slope of the curve changes dramatically. As a general rule, principal components with eigenvalues greater than 1 should be retained. So, based on the scree plot, we can determine that PC1 to PC3 should be retain. However since the combination of the variance is around 54% we need to obtain PC1 to PC6 this would increase the sum of variance percentage over 85%, thus making PC6 our true elbow.

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

```
# Create a biplot
biplot(pca_result, col = ifelse(new_data$smoke == 1, "black", "red"),
       cex = 0.3)

# Add labels for each point
for(i in 1:nrow(new_data)) {
  text(pca_result$x[i,1], pca_result$x[i,2], labels = row.names(new_data)[i],
       pos = 3, cex = 0.3)
}
```



```
# Add axis labels and title
xlab <- paste("PC1 (", round(pca_result$sdev[1] / sum(pca_result$sdev) * 100),
              "%)", sep = "")
ylab <- paste("PC2 (", round(pca_result$sdev[2] / sum(pca_result$sdev) * 100),
              "%)", sep = "")
ggtitle <- "PCA Biplot of Health Survey Data"
ggplot2::labs(x = xlab, y = ylab, title = ggtitle)
```

```
## $x
## [1] "PC1 (18%)"
##
## $y
## [1] "PC2 (16%)"
##
## $title
## [1] "PCA Biplot of Health Survey Data"
##
## attr(,"class")
## [1] "labels"
```

```
# Increase plot size
options(repr.plot.width = 25, repr.plot.height = 25)
```

Comment on observed biplot.

The biplot is a representation of the relationship between the variables (smoking and non-smoking) and the principal components (PC1 and PC2) obtained through PCA step. The plot shows that the smoking and non-smoking groups are well separated and don't overlap, indicating that PC1 and PC2 do provide a clear separation between the two groups. Additionally, there are some clustering of the smoking group in the lower right quadrant of the plot, which also suggests that there are association between smoking and the PCs.

Can we use first two PC to discriminate smoking?

Based on the observed biplot, it seems that the first two principal components (PC1 and PC2) does provide a separation among smoking and non-smoking groups. Hence, we can use first two PC to discriminate smoking.

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

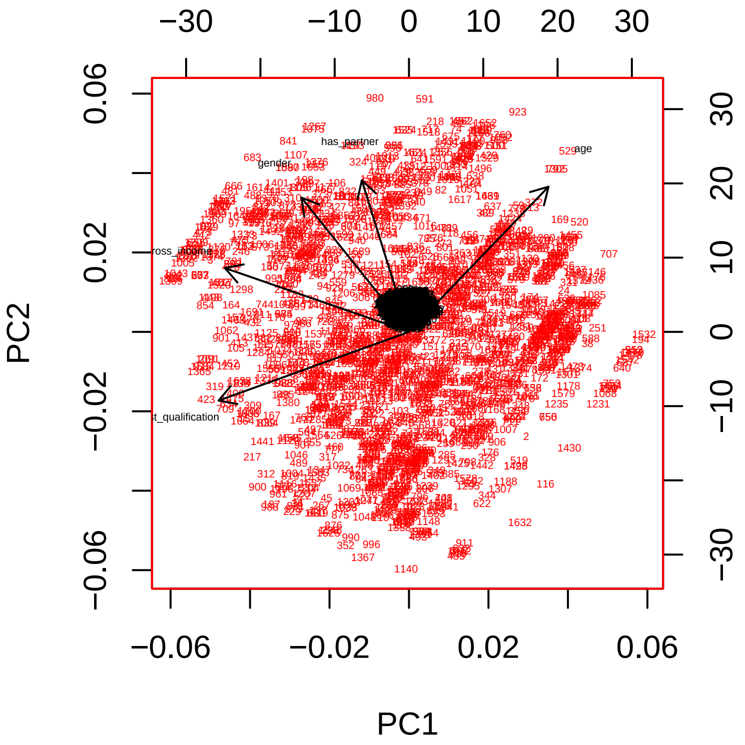
Based on the loading vector, PC1 is mainly driven by the variables related to marriage status, while PC2 is mainly driven by the variables related to age. Therefore, PC1 can be named as the "marriage status" component, and PC2 can be named as the "age consumption" component.

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)

One approach would be to explore alternative methods for encoding categorical variables, such as grouping the marriage status. The marriage status column would be one binary column that combine all the non married people this could would be called "has_partner".

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

```r
# Create a new column for married status
new_data_copy$has_partner <- ifelse(new_data_copy$marital_status == "Married",
                                    1, 0)

new_data_copy <- select(new_data_copy, -marital_status)

# Exclude the "smoking" column from the dataset
data_for_pca <- new_data_copy[, !names(new_data_copy) %in% "smoke"]

# Perform PCA on the remaining columns
pca_result <- prcomp(data_for_pca, scale. = TRUE)

# Create a biplot
biplot(pca_result, col = ifelse(new_data_copy$smoke == 1, "black", "red"),
       cex = 0.3)

# Add labels for each point
for(i in 1:nrow(new_data_copy)) {
  text(pca_result$x[i,1], pca_result$x[i,2],
       labels = row.names(new_data_copy)[i], pos = 3, cex = 0.3)
}
```



```
# Add axis labels and title
xlab <- paste("PC1 (", round(pca_result$sdev[1] / sum(pca_result$sdev) * 100),
              "%)", sep = "")
ylab <- paste("PC2 (", round(pca_result$sdev[2] / sum(pca_result$sdev) * 100),
              "%)", sep = "")
ggtitle <- "PCA Biplot of Health Survey Data"
ggplot2::labs(x = xlab, y = ylab, title = ggtitle)
```

```
## $x
## [1] "PC1 (27%)"
##
## $y
## [1] "PC2 (22%)"
##
## $title
## [1] "PCA Biplot of Health Survey Data"
##
## attr(,"class")
## [1] "labels"
```

```
# Increase plot size
options(repr.plot.width = 25, repr.plot.height = 25)
```

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)

```
# Import Libs
library(caret)
```

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

```
# Read CSV into DataFrame from URl:
# https://raw.githubusercontent.com/zemingzhang1/CSE574LECC-
# Intro-Machine-Learning/main/Assignment%200/penguins.csv
df <- read.csv(
'https://raw.githubusercontent.com/zemingzhang1/CSE574LECC-Intro-Machine-Learning/main/Assignment%200/pe

head(df)
```

```
##   species    island bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## 1  Adelie Torgersen           39.1          18.7               181        3750
## 2  Adelie Torgersen           39.5          17.4               186        3800
## 3  Adelie Torgersen           40.3          18.0               195        3250
## 4  Adelie Torgersen             NA            NA                NA          NA
## 5  Adelie Torgersen           36.7          19.3               193        3450
## 6  Adelie Torgersen           39.3          20.6               190        3650
##      sex year
## 1   male 2007
## 2 female 2007
## 3 female 2007
## 4   <NA> 2007
## 5 female 2007
## 6   male 2007
```

```
# Create new data.frame with only necessary columns
new_data = data.frame(df$species, df$island, df$bill_length_mm,df$bill_depth_mm,
                      df$flipper_length_mm, df$body_mass_g,df$sex)
head(new_data)
```

```
##   df.species df.island df.bill_length_mm df.bill_depth_mm df.flipper_length_mm
## 1     Adelie Torgersen              39.1             18.7                  181
## 2     Adelie Torgersen              39.5             17.4                  186
## 3     Adelie Torgersen              40.3             18.0                  195
## 4     Adelie Torgersen                NA               NA                   NA
## 5     Adelie Torgersen              36.7             19.3                  193
## 6     Adelie Torgersen              39.3             20.6                  190
##   df.body_mass_g df.sex
## 1           3750   male
## 2           3800 female
## 3           3250 female
## 4             NA   <NA>
## 5           3450 female
## 6           3650   male
```

```
# Omit all NA in records
new_data <- na.omit(new_data)
head(new_data)
```

```
##   df.species df.island df.bill_length_mm df.bill_depth_mm df.flipper_length_mm
## 1     Adelie Torgersen              39.1             18.7                  181
## 2     Adelie Torgersen              39.5             17.4                  186
## 3     Adelie Torgersen              40.3             18.0                  195
## 5     Adelie Torgersen              36.7             19.3                  193
## 6     Adelie Torgersen              39.3             20.6                  190
## 7     Adelie Torgersen              38.9             17.8                  181
##   df.body_mass_g df.sex
## 1           3750   male
## 2           3800 female
## 3           3250 female
## 5           3450 female
## 6           3650   male
## 7           3625 female
```

```
# convert categorical variables to numerical variables using one-hot encoding,
# excluding the 'sex' column
df_encoded <- predict(dummyVars(" ~ . - df.sex", data = new_data),
                      newdata = new_data)

# And convert to data frame
df_encoded <- data.frame(df_encoded)

# view the resulting data frame
head(df_encoded)
```

```
##   df.speciesAdelie df.speciesChinstrap df.speciesGentoo df.islandBiscoe
## 1                1                   0                0               0
## 2                1                   0                0               0
## 3                1                   0                0               0
## 5                1                   0                0               0
## 6                1                   0                0               0
## 7                1                   0                0               0
##   df.islandDream df.islandTorgersen df.bill_length_mm df.bill_depth_mm
## 1              0                  1              39.1             18.7
## 2              0                  1              39.5             17.4
## 3              0                  1              40.3             18.0
## 5              0                  1              36.7             19.3
## 6              0                  1              39.3             20.6
## 7              0                  1              38.9             17.8
##   df.flipper_length_mm df.body_mass_g
## 1                  181           3750
## 2                  186           3800
## 3                  195           3250
## 5                  193           3450
## 6                  190           3650
## 7                  181           3625
```

- Perform PCA

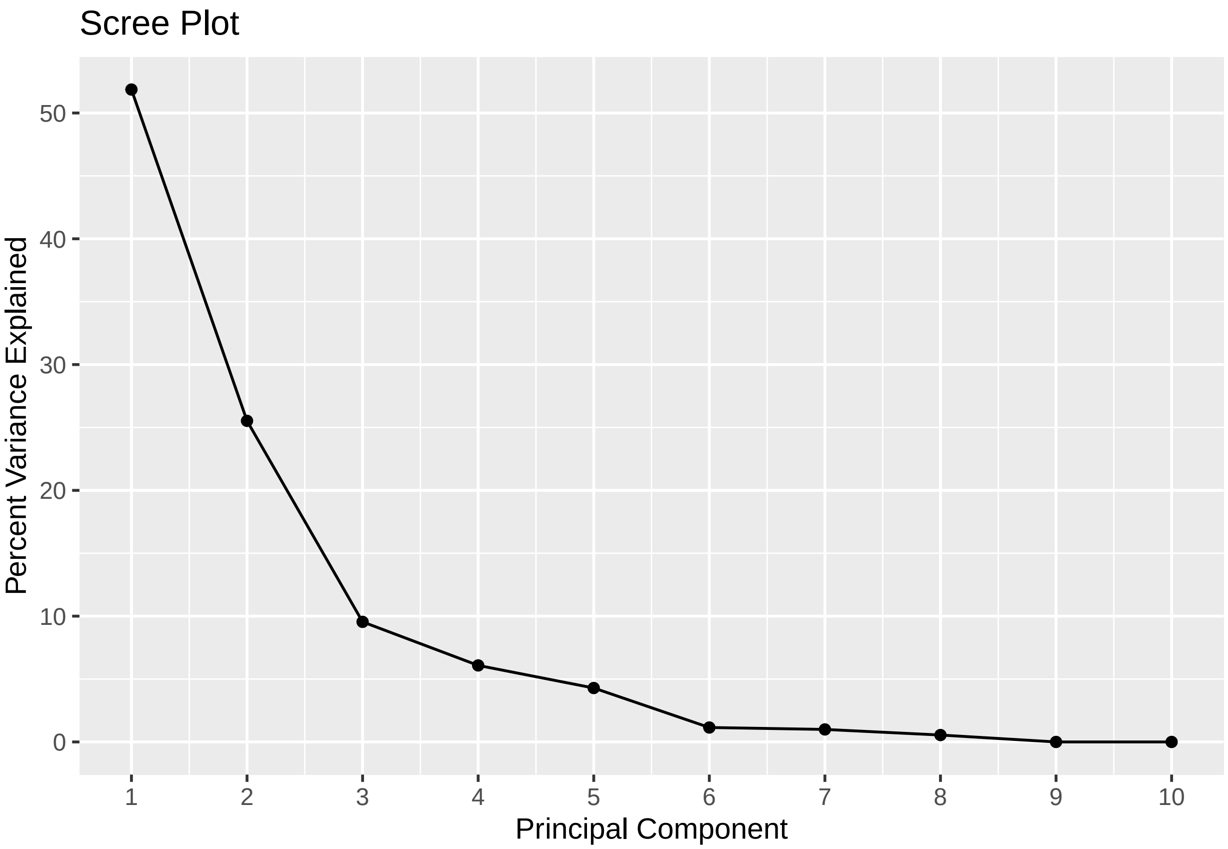
```
# Perform PCA on the remaining columns
pca_result <- prcomp(df_encoded, scale. = TRUE)

# Print the result
pca_result
```

```
## Standard deviations (1, .., p=10):
##  [1] 2.277326e+00 1.597678e+00 9.770532e-01 7.800328e-01 6.548117e-01
##  [6] 3.387949e-01 3.155327e-01 2.345380e-01 5.937132e-15 4.032288e-15
##
```

```
## Rotation (n x k) = (10 x 10):
##                             PC1          PC2         PC3         PC4
## df.speciesAdelie     -0.3132925  0.396491714  0.09638073  0.30635873
## df.speciesChinstrap  -0.1204261 -0.560941054 -0.13086419 -0.21352436
## df.speciesGentoo      0.4256915  0.061325435  0.01028779 -0.13759441
## df.islandBiscoe       0.3558355  0.214115707  0.30149196  0.13927634
## df.islandDream       -0.2471130 -0.443106032  0.23915588  0.07095865
## df.islandTorgersen   -0.1683482  0.306817822 -0.76439577 -0.29833332
## df.bill_length_mm     0.2659753 -0.418633454 -0.30695005  0.15527961
## df.bill_depth_mm     -0.3364212 -0.094172178 -0.23922205  0.69226432
## df.flipper_length_mm  0.3997574 -0.062613427 -0.21303040  0.17344983
## df.body_mass_g        0.3825952  0.003188805 -0.21037624  0.44131300
##                              PC5          PC6          PC7         PC8
## df.speciesAdelie     -0.10609976  0.235036111 -0.218496738  0.37625842
## df.speciesChinstrap   0.36806793 -0.293507212  0.226406532  0.28404448
## df.speciesGentoo     -0.19975925  0.003535728  0.035783863 -0.62852276
## df.islandBiscoe       0.54457490 -0.054792094 -0.009284758  0.06242433
## df.islandDream       -0.53774977  0.048932717 -0.020941203 -0.05222814
## df.islandTorgersen   -0.03645068  0.010837891  0.042359807 -0.01722789
## df.bill_length_mm     0.18072948  0.684769561 -0.365280819  0.02589365
## df.bill_depth_mm      0.24122167 -0.246255420 -0.017452095 -0.47155179
## df.flipper_length_mm -0.26016440 -0.563232028 -0.549755992  0.27140247
## df.body_mass_g       -0.26031373  0.079395190  0.679288875  0.28123874
##                                PC9          PC10
## df.speciesAdelie      9.209131e-03  6.209686e-01
## df.speciesChinstrap   7.481675e-03  5.044868e-01
## df.speciesGentoo      8.894095e-03  5.997259e-01
## df.islandBiscoe      -6.431192e-01  9.537629e-03
## df.islandDream       -6.209196e-01  9.208404e-03
## df.islandTorgersen   -4.479248e-01  6.642845e-03
## df.bill_length_mm     5.358504e-16 -5.201394e-16
## df.bill_depth_mm     -2.790674e-15  1.476517e-16
## df.flipper_length_mm  3.196128e-16  1.393801e-15
## df.body_mass_g        1.122801e-15 -6.485000e-16
```

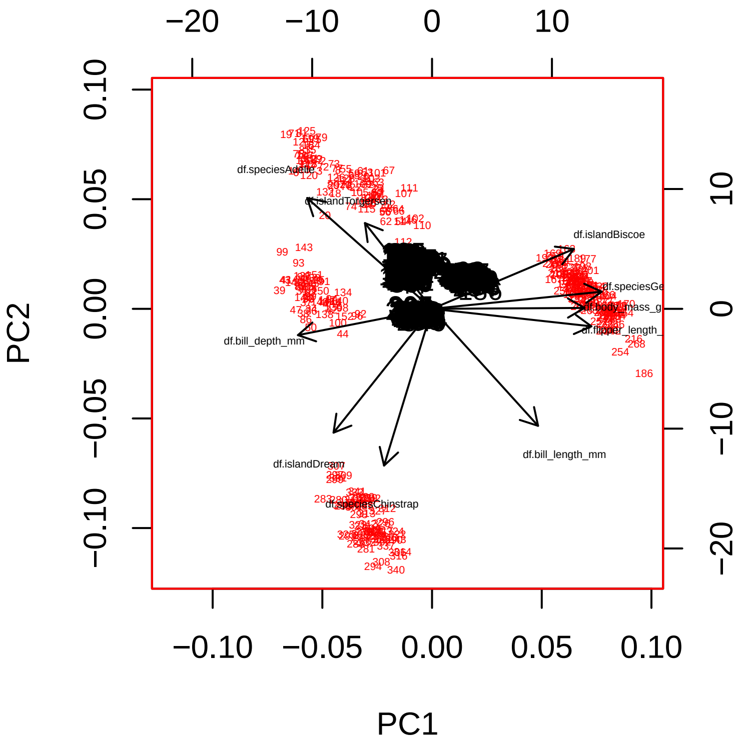
- Make a skree plot

```r
# Extract the variance explained by each principal component
pca_var <- pca_result$sdev^2
pca_var_percent <- pca_var / sum(pca_var) * 100

# Create a data frame for the scree plot
scree_data <- data.frame(
  PC = 1:length(pca_var),
  Variance = pca_var_percent)

# Create the scree plot
ggplot(scree_data, aes(x = PC, y = Variance)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:length(pca_var)) +
  labs(x = "Principal Component", y = "Percent Variance Explained",
       title = "Scree Plot")
```



- Make a biplot

```
# Create a biplot
biplot(pca_result, col = ifelse(new_data$df.sex == "female", "black", "red"),
       cex = 0.3)

# Add labels for each point
for(i in 1:nrow(df_encoded)) {
  text(pca_result$x[i,1], pca_result$x[i,2], labels = row.names(df_encoded)[i],
       pos = 3, cex = 0.8)
}
```



```r
# Add axis labels and title
xlab <- paste("PC1 (", round(pca_result$sdev[1] / sum(pca_result$sdev) * 100),
              "%)", sep = "")
ylab <- paste("PC2 (", round(pca_result$sdev[2] / sum(pca_result$sdev) * 100),
              "%)", sep = "")
ggtitle <- "PCA Biplot of Health Survey Data"
ggplot2::labs(x = xlab, y = ylab, title = ggtitle)
```

```
## $x
## [1] "PC1 (32%)"
##
## $y
## [1] "PC2 (22%)"
##
## $title
## [1] "PCA Biplot of Health Survey Data"
##
## attr(,"class")
## [1] "labels"
```

```r
# Increase plot size
options(repr.plot.width = 25, repr.plot.height = 25)
```

Discuss your observations (9 points)

Comment on the shape of scree plot, if you need to reduce dimensions home many would you choose?

Based on the scree plot, we can determine that PC1 to PC4 should be retain. However since the combination of the sum of variance percentage is over 85%, thus making PC3 our true elbow.

Comment on observed biplot.

The biplot is a representation of the relationship between the variables PC1 to PC3 obtained through PCA step. The plot shows that the species groups are well separated and don't overlap, indicating that the PC3 do provide a clear separation between the species groups.

Can we use first two PC to discriminate species?

Based on the observed biplot, it seems that the first two principal components (PC1 and PC2) does provide a separation among species groups. Hence, we can use first two PC to discriminate species.

Based on the loading vector can we name PC with some descriptive name?

Based on the loading vector, PC1 is mainly driven by the variables related to body mass, while PC2 is mainly driven by the variables related to island location and PC3 is driven by bill depth. Therefore, PC1 can be named as the "body size" component,PC2 can be named as the "island location" component and PC3 can be named as "bill depth"component.

May be some of splits between categories or mapping to numeric should be revisited, if so what will you do differently?

One approach would be to explore alternative methods such as taking out the species column.