Homework 2: PCA (60 Points)

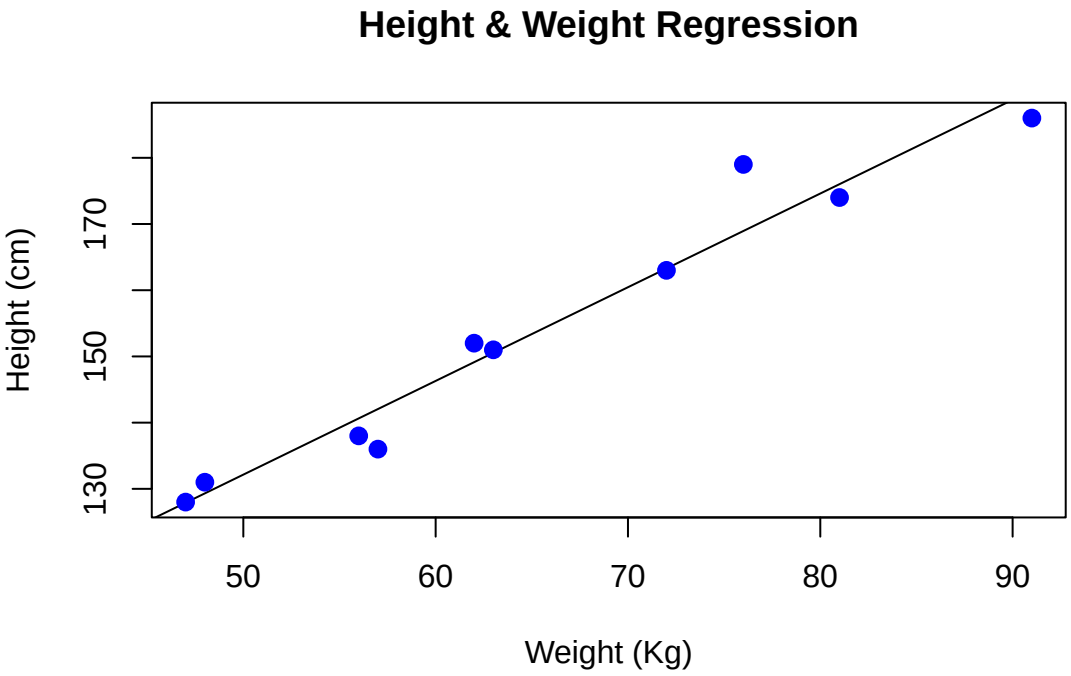
Sushrut Gaikwad (50604159)

2025-04-03

```
# load libraries
library(tidyverse)
library(gridExtra)
library(ggbiplot)
knitr::opts_chunk$set(fig.width=6, fig.height=4)
```

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

Bob and Mary study height and weight dataset.



Mary used PCA with height and weight treated as features and Bob used linear regression (LR) with height treated as an outcome and weight as an explanatory variable. They argued which method is most appropriate here. A third student overheard it and said that LR and 1st principal component (PC) in PCA would give the same answer as they both do linear fit. The centering and scale of 1st-PC will be accounted in shift of LR (β_0) and relationship between β_1 and ϕ's can be easily obtained if needed.

Is the third student right?

Answer: No.

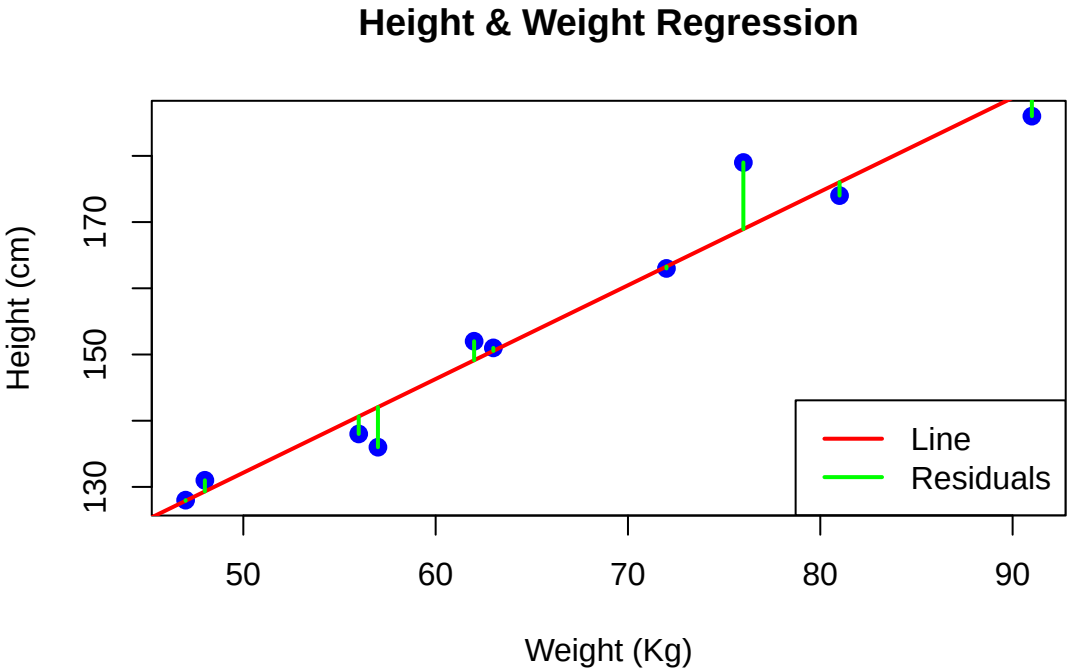
What is the difference between optimization for linear regression coefficients and 1st-PC calculations?

Answer:

Optimization for linear regression is finding the coefficients β_i by minimizing the sum of squared errors, i.e.,

$$\min_{\beta_j} \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_i \right) \right)^2$$

where y_i is the true output, and $\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_i\right)$ is the predicted output. Here, I am considering that there are p predictors in the data, and n data points. This amounts to finding a best fitting line such that the *vertical distance*, also known as *residuals*, between the true output y_i and the predicted output is the least. This is illustrated in the following plot.



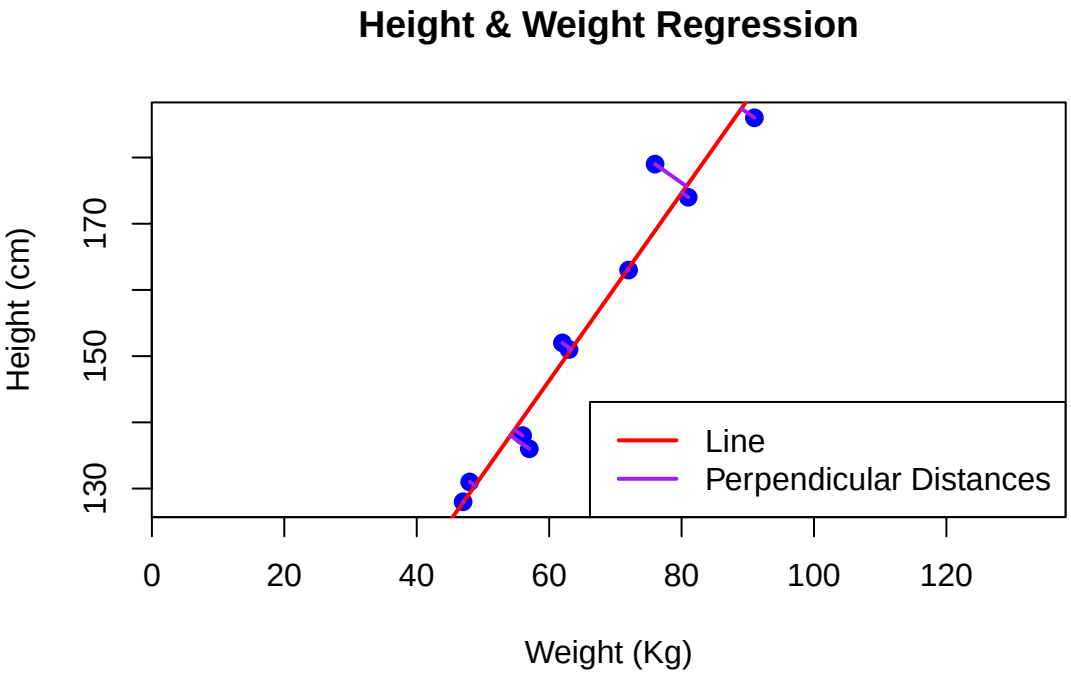
On the other hand, the 1st-PC $\mathbf{z}_1$ of a set of predictors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ is found by taking the linear combination of these features, i.e.,

$$\mathbf{z}_1 = \phi_{11}\mathbf{x_1} + \phi_{21}\mathbf{x_2} + \cdots + \phi_{p1}\mathbf{x_p}$$

that have the largest variance subject to the following constraint:

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

We aim to find these parameters ϕ_{j1}, known as *loadings*, to find the first principal component $\mathbf{z}_1$. This amounts to finding a line whose *perpendicular distance* from the points is the least. This is illustrated in the following plot.



Minimizing the vertical distance of the points from the line (i.e., linear regression) and minimizing the perpendicular distance of the points from the line (i.e., finding the first PC) would generally result in two different lines. Hence, the third student is wrong.

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 **1,691 observations** on the following **12 variables**.

Variable	Description
`gender`	Gender with levels **Female** and **Male**.
`age`	Age of the individual.
`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**.

Variable	Description
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. Obtained from OpenIntro.

Read and Clean the Data

2.1: Read the data from "smoking.R" or "smoking.rda". (3 points)

Hint: Take a look at the `source` or `load` functions. There is also "smoking.csv" file for a reference.

```
# Load data
load("../data/smoking.rda")
```

Take a look into data

```
head(smoking)
```

```
## # A tibble: 6 x 12
##   gender   age marital_status highest_qualification nationality ethnicity
##   <fct>  <int> <fct>          <fct>                  <fct>       <fct>
## 1 Male      38 Divorced       No Qualification       British     White
## 2 Female    42 Single         No Qualification       British     White
## 3 Male      40 Married        Degree                 English     White
## 4 Female    40 Married        Degree                 English     White
## 5 Female    39 Married        GCSE/O Level           British     White
## 6 Female    37 Married        GCSE/O Level           British     White
## # i 6 more variables: gross_income <fct>, region <fct>, smoke <fct>,
## #   amt_weekends <int>, amt_weekdays <int>, type <fct>
```

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.

```
smoking_subset <- smoking[
  , c("smoke", "gender", "age", "marital_status", "highest_qualification", "gross_income")
]
```

2.2: Omit all incomplete records. (3 points)

```
smoking_subset <- na.omit(smoking_subset)
```

2.3: For PCA, features 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 (e.g. `is_married`, `is_devorced`). Several features in the data set are ordinal (e.g., `gross_income` and `highest_qualification`). Convert them to some king of sensible level (note that levels in factors are not in order). (3 points)

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

# One-hot encoding for marital status
smoking_subset$is_married <- as.integer(smoking_subset$marital_status == "Married")
smoking_subset$is_divorced <- as.integer(smoking_subset$marital_status == "Divorced")
smoking_subset$is_separated <- as.integer(smoking_subset$marital_status == "Separated")
smoking_subset$is_single <- as.integer(smoking_subset$marital_status == "Single")
smoking_subset$is_widowed <- as.integer(smoking_subset$marital_status == "Widowed")

# Drop original marital_status column
smoking_subset$marital_status <- NULL

# Assign numeric values to ordinal variables

## Encoding highest qualification (ordered from lowest to highest education)
education_levels <- c("No Qualification", "GCSE/CSE", "GCSE/O Level", "ONC/BTEC",
                      "A Levels", "Other/Sub Degree", "Higher/Sub Degree", "Degree")

smoking_subset$highest_qualification <- as.numeric(
  factor(smoking_subset$highest_qualification,
  levels = education_levels,
  ordered = TRUE)
)

## Encoding gross income (ordered from lowest to highest)
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", "Refused", "Unknown")

smoking_subset$gross_income <- as.numeric(factor(smoking_subset$gross_income,
                                                 levels = income_levels,
                                                 ordered = TRUE))
```

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

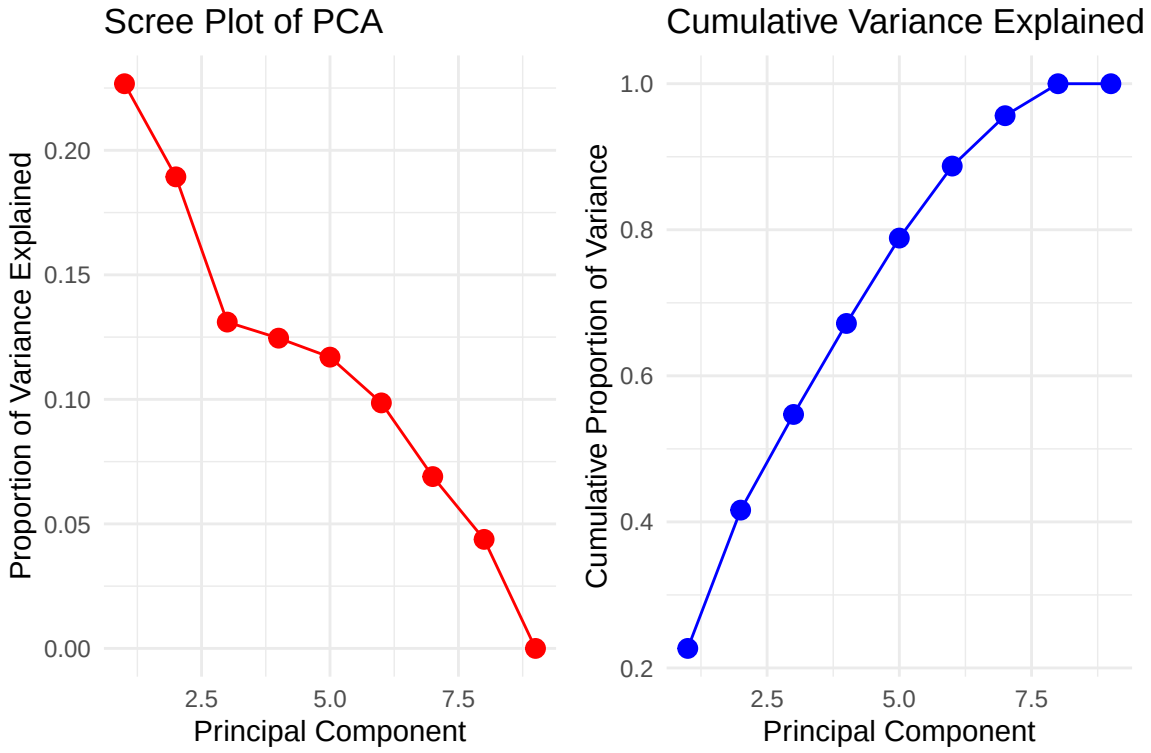
```r
# Remove the `smoke` column
pca_data <- smoking_subset[, !(names(smoking_subset) %in% c("smoke"))]

# Standardize the data
pca_data_scaled <- scale(pca_data)

# Perform PCA
pca_result <- prcomp(pca_data_scaled, center = TRUE, scale. = TRUE)

# Add the PCA-transformed data back to the original dataset
pca_transformed <- data.frame(pca_result$x, smoke = smoking_subset$smoke)

# Summary of PCA
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6    PC7
## Standard deviation     1.4285 1.3054 1.0860 1.0588 1.0260 0.94191 0.7880
## Proportion of Variance 0.2267 0.1893 0.1310 0.1246 0.1169 0.09858 0.0690
## Cumulative Proportion  0.2267 0.4161 0.5471 0.6717 0.7886 0.88722 0.9562
##                           PC8       PC9
## Standard deviation     0.62771 1.176e-15
## Proportion of Variance 0.04378 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00
```

2.5: Make a scree plot. (3 points)

```r
# Compute explained variance
explained_variance <- pca_result$sdev^2 / sum(pca_result$sdev^2)

# Compute cumulative variance
cumulative_variance <- cumsum(explained_variance)

# Scree Plot: Proportion of variance explained
scree_plot <- ggplot(
  data.frame(PC = 1:length(explained_variance), Variance = explained_variance),
  aes(x = PC, y = Variance)
) +
  geom_line(aes(group = 1), color = "red") +
  geom_point(color = "red", size = 3) +
  labs(
    title = "Scree Plot of PCA",
    x = "Principal Component",
    y = "Proportion of Variance Explained"
  ) +
  theme_minimal()

# Cumulative Variance Plot
cumulative_plot <- ggplot(
  data.frame(
    PC = 1:length(cumulative_variance), CumulativeVariance = cumulative_variance
  ),
```

```
  aes(x = PC, y = CumulativeVariance)
) +
  geom_line(aes(group = 1), color = "blue") +
  geom_point(color = "blue", size = 3) +
  labs(
    title = "Cumulative Variance Explained",
    x = "Principal Component",
    y = "Cumulative Proportion of Variance"
  ) +
  theme_minimal()

# Display both plots side by side
grid.arrange(scree_plot, cumulative_plot, ncol = 2)
```

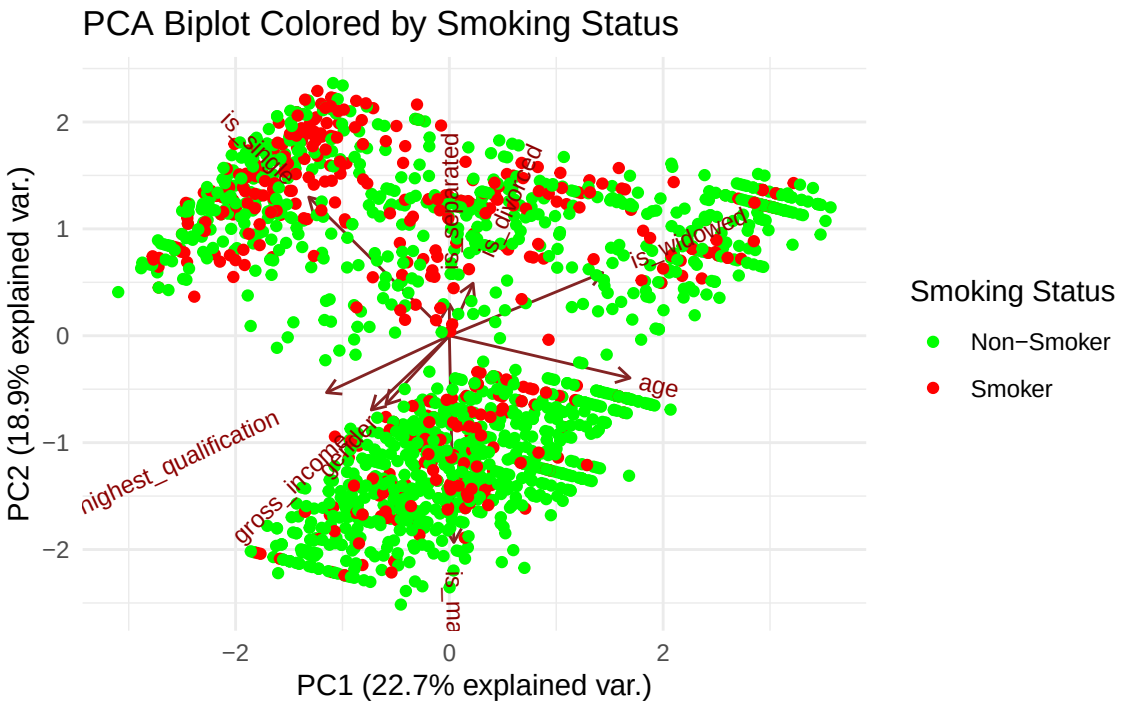


Comment on the shape. If you need to reduce dimensions home many would you choose? The scree plot shows a steep drop in the proportion of variance explained till the first three principal components. After this, the drop is gradual. This means that most of the information is captured by the first three principal components. The cumulative variance plot shows a sharp rise initially and the cumulative proportion of variance explained reaches nearly 90% just around 6 principal components. After this, additional principal components contribute little new information.

For choosing the principal components, we can use the elbow method by looking at the scree plot. This plot shows that the elbow occurs around at least the third principal component. Hence, we can keep at least three principal components. If we want to use the cumulative variance plot to select the number of principal components, we can choose that we want nearly 90% of the cumulative variance explained. In this case, we would want to choose at least six principal components.

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

```r
# Convert `smoke` to a factor for coloring
pca_transformed$smoke <- factor(
  pca_transformed$smoke, levels = c(0, 1), labels = c("Non-Smoker", "Smoker")
)

# Create the PCA biplot
ggbiplot(
  pca_result,
  obs.scale = 1,
  var.scale = 1,
  groups = pca_transformed$smoke,
) +
  scale_color_manual(values = c("green", "red")) +
  labs(
    title = "PCA Biplot Colored by Smoking Status",
    color = "Smoking Status"
  ) +
  theme_minimal()
```



Comment on observed biplot. The observed plot shows the distribution of data points based on the first two principal components, i.e., PC1 and PC2, and the colors indicate smoking status. The arrows are the loadings that indicate how much each variable contributes to these two PCs. The distribution of smokers (red) and non-smokers (green) appears to be mixed, i.e., there is no clear separation between the two.

Can we use first two PCs to discriminate smoking? As the distribution of smokers and non-smokers appears to be mixed, there is no clear boundary between the two. Hence, we cannot use the first two PCs to discriminate smoking.

2.7: Based on the loading vectors can we name the PCs with some descriptive name? (3 points)

Let us first have a look at the loading vectors.

```
print(pca_result$rotation)
```

```
##                                  PC1        PC2         PC3         PC4
## gender                 -0.1991400189 -0.2359277 -0.49051589 -0.17566803
## age                     0.5663582074 -0.1449435 -0.22687021 -0.09404125
## highest_qualification  -0.3852326771 -0.1949863 -0.08607121 -0.15162194
## gross_income           -0.2448078261 -0.2546756 -0.47131737 -0.40425559
## is_married              0.0140457697 -0.7090911  0.23376239  0.23874408
## is_divorced             0.0742352995  0.1802873  0.33815430 -0.80039797
## is_separated            0.0001401743  0.1090227  0.18022923 -0.08359789
## is_single              -0.4408398418  0.4758893 -0.19956310  0.25579994
## is_widowed              0.4808311528  0.2162866 -0.48690303  0.06186538
##                                  PC5          PC6         PC7         PC8
## gender                 -0.011509545  0.660976262  0.42411275  0.12775971
## age                    -0.003399743 -0.005606802  0.12258318 -0.76344370
## highest_qualification  -0.068455028 -0.648611672  0.58889258 -0.10488565
## gross_income           -0.139169384 -0.194351674 -0.65599286 -0.04668537
## is_married              0.094135120  0.032737266 -0.07705951  0.03313639
## is_divorced             0.260376099  0.077482520  0.07234496  0.02883543
## is_separated           -0.934709410  0.113586852  0.04805337 -0.05281107
## is_single               0.157665473  0.077107814 -0.06225898 -0.40085653
## is_widowed             -0.024587551 -0.280510987  0.10306975  0.47142993
##                                  PC9
## gender                 -1.067273e-15
## age                     9.971424e-17
## highest_qualification  -2.962745e-16
## gross_income           -9.796532e-18
## is_married              6.069428e-01
## is_divorced             3.565611e-01
## is_separated            2.386652e-01
## is_single               5.277920e-01
## is_widowed              4.110463e-01
```

Based on these loading vectors, the variables most affecting the PCs are summarized in the following table.

PC	Descriptive Name	Main Positive Influences	Main Negative Influences
PC1	Socioeconomic & Marital Status	`age`, `is_widowed`	`highest_qualification`, `gross_income`, `is_single`
PC2	Married vs. Unmarried	`is_single`, `is_widowed`, `is_divorced`	`is_married`
PC3	Gender & Financial Independence	`is_divorced`, `is_married`	`gender`, `gross_income`
PC4	Divorced vs. Financial Stability	`is_divorced`, `gross_income`	`is_married`, `is_single`
PC5	Separated vs. Others	`is_divorced`	`is_separated`
PC6	Education & Gender	`gender`	`highest_qualification`
PC7	Income vs. Education Trade-off	`highest_qualification`	`gross_income`
PC8	Age & Income Stability	`is_widowed`	`age`, `is_single`
PC9	Marital Flexibility	`is_married`, `is_single`, `is_widowed`	*None dominant*

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)

I previously did ordered encoding on the columns `highest_qualification` and `gross_income`. These columns have a lot of categories. I would like to ordinally encode them differently in the following way.

- `highest_qualification`:
  - I previously encoded this column in the order `No Qualification`, `GCSE/CSE`, `GCSE/O Level`, `ONC/BTEC`, `A Levels`, `Other/Sub Degree`, `Higher/Sub Degree`, and `Degree`. However, I would like to group some of these into the same level and reduce the number of categories after encoding.
- `gross_income`:
  - My previous encoding order was `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`. Firstly I will convert the `Refused` and `Unknown` categories into `NA` and then drop the `NA` values. Next, I will again group some of these categories into the same level, reducing the number of categories after encoding.

2.9: Follow your suggestion in 2.8 and redo PCA and biplot. (3 points)

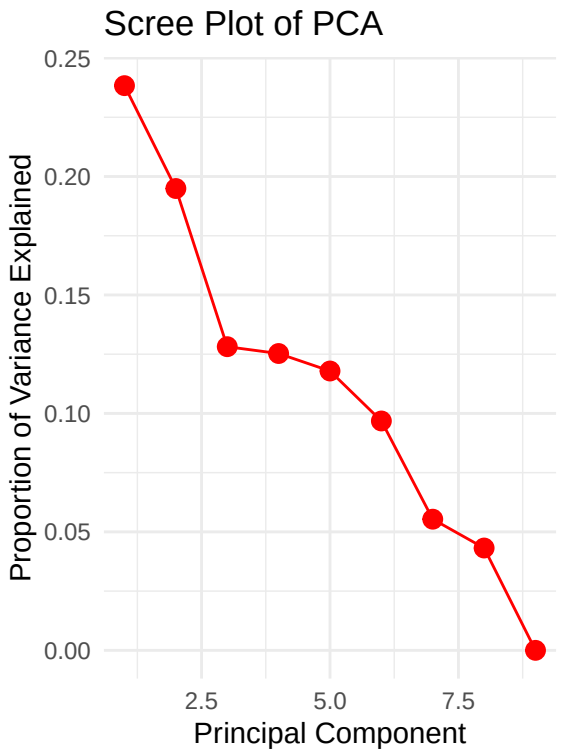
```
load("../data/smoking.rda")
smoking_subset <- smoking[
  ,
  c("smoke", "gender", "age", "marital_status", "highest_qualification", "gross_income")
]
smoking_subset <- na.omit(smoking_subset)

# Convert binary categorical variables to numeric
smoking_subset$gender <- ifelse(smoking_subset$gender == "Male", 1, 0)
smoking_subset$smoke <- ifelse(smoking_subset$smoke == "Yes", 1, 0)

# One-hot encoding for marital status
smoking_subset$is_married <- as.integer(smoking_subset$marital_status == "Married")
smoking_subset$is_divorced <- as.integer(smoking_subset$marital_status == "Divorced")
smoking_subset$is_separated <- as.integer(smoking_subset$marital_status == "Separated")
smoking_subset$is_single <- as.integer(smoking_subset$marital_status == "Single")
smoking_subset$is_widowed <- as.integer(smoking_subset$marital_status == "Widowed")

# Drop original marital_status column
smoking_subset$marital_status <- NULL

# Assign numeric values to ordinal variables

## Encoding highest qualification (ordered from lowest to highest education)
education_levels <- c(
  "No Qualification",          # Level 1 (Lowest)
  "GCSE/CSE", "GCSE/O Level", # Level 2 (Secondary school)
  "ONC/BTEC", "A Levels",      # Level 3 (Vocational vs. Academic)
  "Other/Sub Degree", "Higher/Sub Degree", # Level 4/5 (Sub-degree qualifications)
  "Degree"                     # Level 6 (Highest - Bachelor's degree)
)

smoking_subset$highest_qualification <- as.numeric(factor(
  smoking_subset$highest_qualification,
  levels = education_levels,
  labels = c(1, 2, 2, 3, 3, 4, 4, 5),
```

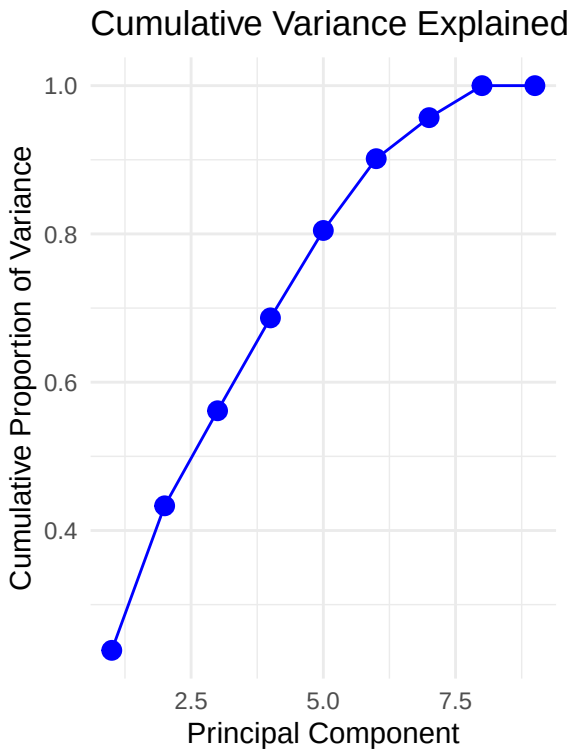
```
  ordered = TRUE
))

## Encoding gross income (ordered from lowest to highest)
income_levels <- c(
  "Under 2,600", "2,600 to 5,200", "5,200 to 10,400",  # Low Income (1)
  "10,400 to 15,600", "15,600 to 20,800", "20,800 to 28,600",  # Middle Income (2)
  "28,600 to 36,400", "Above 36,400"  # High Income (3)
)

smoking_subset <- smoking_subset[
  !(smoking_subset$gross_income %in% c("Refused", "Unknown")),
]

smoking_subset$gross_income <- as.numeric(factor(
  smoking_subset$gross_income,
  levels = income_levels,
  labels = c(1, 1, 1, 2, 2, 2, 3, 3),
  ordered = TRUE
))

# Remove the `smoke` column
pca_data <- smoking_subset[, !(names(smoking_subset) %in% c("smoke"))]

# Standardize the data
pca_data_scaled <- scale(pca_data)

# Perform PCA
pca_result <- prcomp(pca_data_scaled, center = TRUE, scale. = TRUE)

# Add the PCA-transformed data back to the original dataset
pca_transformed <- data.frame(pca_result$x, smoke = smoking_subset$smoke)

# Summary of PCA
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.4647 1.3245 1.0740 1.0619 1.0301 0.93344 0.70584
## Proportion of Variance 0.2384 0.1949 0.1282 0.1253 0.1179 0.09681 0.05536
## Cumulative Proportion  0.2384 0.4333 0.5615 0.6867 0.8046 0.90146 0.95682
##                           PC8       PC9
## Standard deviation     0.62340 8.916e-16
## Proportion of Variance 0.04318 0.000e+00
## Cumulative Proportion  1.00000 1.000e+00
```

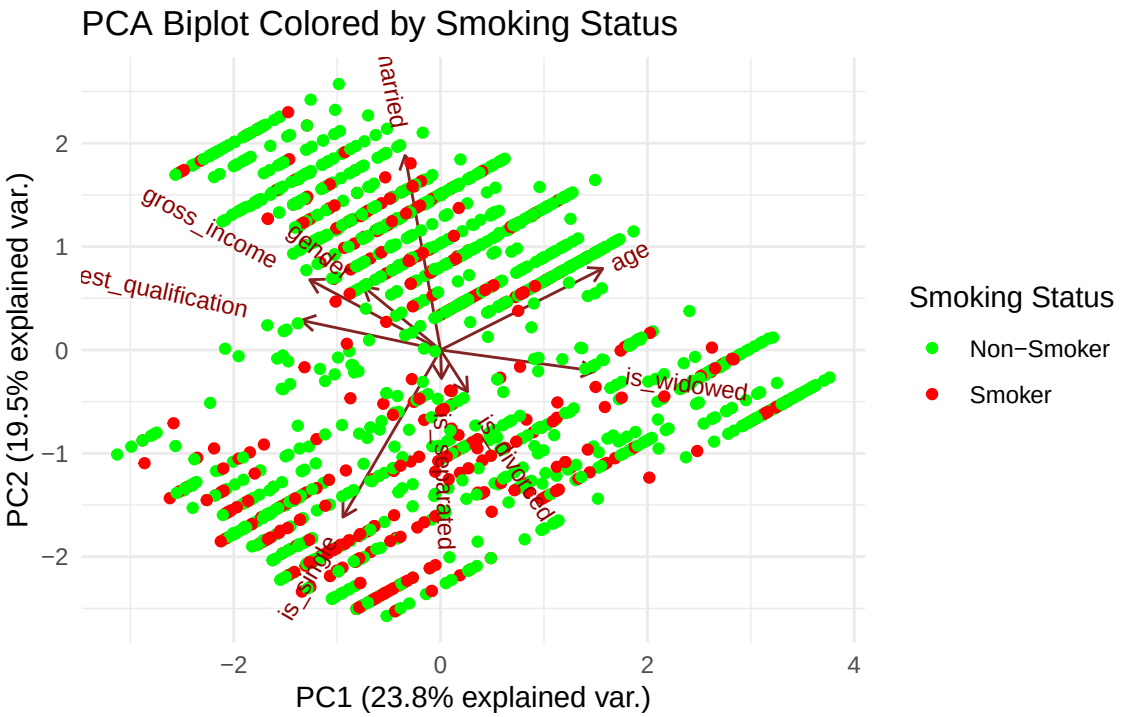
```
# Compute explained variance
explained_variance <- pca_result$sdev^2 / sum(pca_result$sdev^2)

# Compute cumulative variance
cumulative_variance <- cumsum(explained_variance)

# Scree Plot: Proportion of variance explained
scree_plot <- ggplot(
```

```
  data.frame(PC = 1:length(explained_variance), Variance = explained_variance),
  aes(x = PC, y = Variance)
) +
  geom_line(aes(group = 1), color = "red") +
  geom_point(color = "red", size = 3) +
  labs(
    title = "Scree Plot of PCA",
    x = "Principal Component",
    y = "Proportion of Variance Explained"
  ) +
  theme_minimal()

# Cumulative Variance Plot
cumulative_plot <- ggplot(
  data.frame(
    PC = 1:length(cumulative_variance), CumulativeVariance = cumulative_variance
  ),
  aes(x = PC, y = CumulativeVariance)
) +
  geom_line(aes(group = 1), color = "blue") +
  geom_point(color = "blue", size = 3) +
  labs(
    title = "Cumulative Variance Explained",
    x = "Principal Component",
    y = "Cumulative Proportion of Variance"
  ) +
  theme_minimal()

# Display both plots side by side
grid.arrange(scree_plot, cumulative_plot, ncol = 2)
```






```r
# Convert `smoke` to a factor for coloring
pca_transformed$smoke <- factor(
  pca_transformed$smoke, levels = c(0, 1), labels = c("Non-Smoker", "Smoker")
)

# Create the PCA biplot
ggbiplot(
  pca_result,
  obs.scale = 1,
  var.scale = 1,
  groups = pca_transformed$smoke,
) +
  scale_color_manual(values = c("green", "red")) +
  labs(
    title = "PCA Biplot Colored by Smoking Status",
    color = "Smoking Status"
  ) +
  theme_minimal()
```



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 for the in class PCA examples (iris, mpg, diamonds, and so on).

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

Solution

I will be using the heart failure dataset with the following features.

Variable	Description
`Age`	Age of the patient [years].
`Sex`	Sex of the patient [`M`: Male, `F`: Female].
`ChestPainType`	Chest pain type [`TA`: Typical Angina, `ATA`: Atypical Angina, `NAP`: Non-Anginal Pain, `ASY`: Asymptomatic].
`RestingBP`	Resting blood pressure [mm Hg].
`Cholesterol`	Serum cholesterol [mm/dl].
`FastingBS`	Fasting blood sugar [`1`: if `FastingBS` > 120 mg/dl, `0`: otherwise].
`RestingECG`	Resting electrocardiogram results [`Normal`: Normal, `ST`: having ST-T wave abnormality, `LVH`: showing probable or definite left ventricular hypertrophy by Estes' criteria].
`MaxHR`	Maximum heart rate achieved [Numeric value between 60 and 202].
`ExerciseAngina`	Exercise-induced angina [`Y`: Yes, `N`: No].
`Oldpeak`	Oldpeak = ST [Numeric value measured in depression].
`ST_Slope`	The slope of the peak exercise ST segment [`Up`: up sloping, `Flat`: flat, `Down`: down sloping].

Variable	Description
`HeartDisease`	Output class [1: heart disease, 0: normal].

```
heart_sample <- read_csv("../data/heart.csv")
```

```
## Rows: 918 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Sex, ChestPainType, RestingECG, ExerciseAngina, ST_Slope
## dbl (7): Age, RestingBP, Cholesterol, FastingBS, MaxHR, Oldpeak, HeartDisease
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
spec(heart_sample)
```

```
## cols(
##   Age = col_double(),
##   Sex = col_character(),
##   ChestPainType = col_character(),
##   RestingBP = col_double(),
##   Cholesterol = col_double(),
##   FastingBS = col_double(),
##   RestingECG = col_character(),
##   MaxHR = col_double(),
##   ExerciseAngina = col_character(),
##   Oldpeak = col_double(),
##   ST_Slope = col_character(),
##   HeartDisease = col_double()
## )
```

```
heart_data <- read_csv("../data/heart.csv", col_types = cols(
  Age = col_double(),
  Sex = col_character(),
  ChestPainType = col_character(),
  RestingBP = col_double(),
  Cholesterol = col_double(),
  FastingBS = col_double(),
  RestingECG = col_character(),
  MaxHR = col_double(),
  ExerciseAngina = col_character(),
  Oldpeak = col_double(),
  ST_Slope = col_character(),
  HeartDisease = col_double()
))

cols_to_factor <- c("Sex", "ChestPainType", "RestingECG", "ExerciseAngina", "ST_Slope")
heart_data[cols_to_factor] <- lapply(heart_data[cols_to_factor], as.factor)
```

Convert columns to proper format. (9 points)

```
# Binary Encoding: Convert `Sex` and `ExerciseAngina` into 0/1
heart_data$Sex <- ifelse(heart_data$Sex == "M", 1, 0)
```

```r
heart_data$ExerciseAngina <- ifelse(heart_data$ExerciseAngina == "Y", 1, 0)

# Ordinal Encoding: Convert categorical variables into ordered factors
heart_data$ChestPainType <- as.numeric(factor(heart_data$ChestPainType,
                                              levels = c("TA", "ATA", "NAP", "ASY"),
                                              ordered = TRUE))

heart_data$RestingECG <- as.numeric(factor(heart_data$RestingECG,
                                           levels = c("Normal", "ST", "LVH"),
                                           ordered = TRUE))

heart_data$ST_Slope <- as.numeric(factor(heart_data$ST_Slope,
                                         levels = c("Down", "Flat", "Up"),
                                         ordered = TRUE))

# Ensure `HeartDisease` is numeric for classification
heart_data$HeartDisease <- as.numeric(heart_data$HeartDisease)
```

Perform PCA. (3 points)

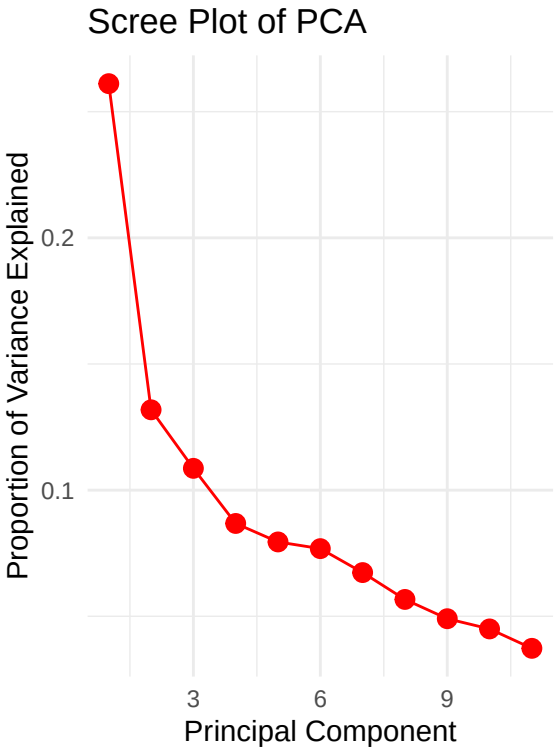
```r
# Remove the `HeartDisease` column
pca_data <- heart_data[, !(names(heart_data) %in% c("HeartDisease"))]

# Standardize the data
pca_data_scaled <- scale(pca_data)

# Perform PCA
pca_result <- prcomp(pca_data_scaled, center = TRUE, scale. = TRUE)

# Add the PCA-transformed data back to the original dataset
pca_transformed <- data.frame(pca_result$x, HeartDisease = heart_data$HeartDisease)

# Summary of PCA
summary(pca_result)
```

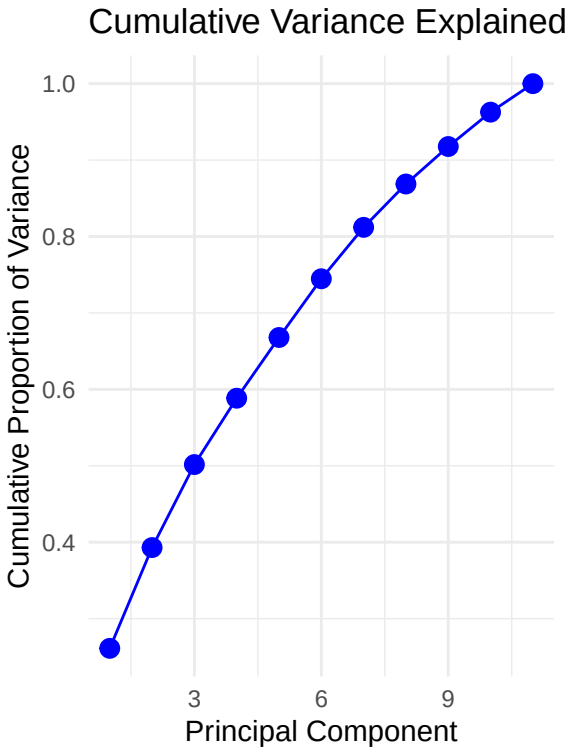
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.6948 1.2042 1.0932 0.97710 0.93494 0.91924 0.86058
## Proportion of Variance 0.2611 0.1318 0.1086 0.08679 0.07946 0.07682 0.06733
## Cumulative Proportion  0.2611 0.3930 0.5016 0.58838 0.66785 0.74467 0.81199
##                            PC8     PC9    PC10   PC11
## Standard deviation     0.78958 0.73448 0.70350 0.6405
## Proportion of Variance 0.05668 0.04904 0.04499 0.0373
## Cumulative Proportion  0.86867 0.91771 0.96270 1.0000
```

Make a scree plot. (3 points)

```r
# Compute explained variance
explained_variance <- pca_result$sdev^2 / sum(pca_result$sdev^2)

# Compute cumulative variance
cumulative_variance <- cumsum(explained_variance)
```

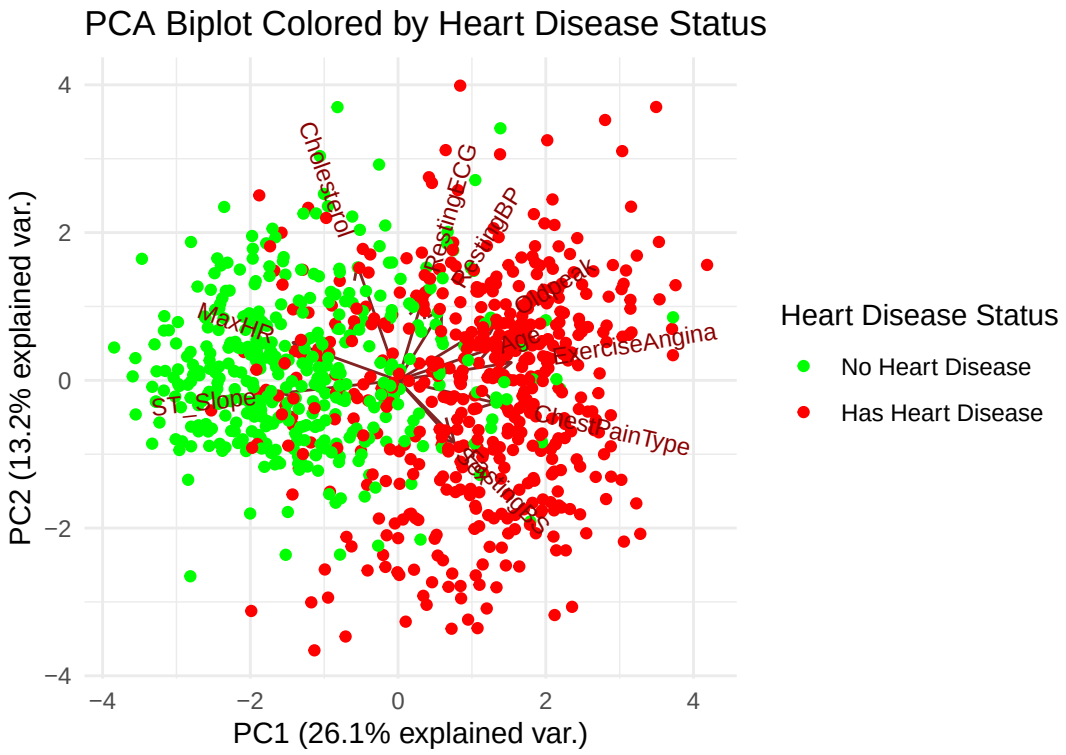
```r
# Scree Plot: Proportion of variance explained
scree_plot <- ggplot(
  data.frame(PC = 1:length(explained_variance), Variance = explained_variance),
  aes(x = PC, y = Variance)
) +
  geom_line(aes(group = 1), color = "red") +
  geom_point(color = "red", size = 3) +
  labs(
    title = "Scree Plot of PCA",
    x = "Principal Component",
    y = "Proportion of Variance Explained"
  ) +
  theme_minimal()

# Cumulative Variance Plot
cumulative_plot <- ggplot(
  data.frame(
    PC = 1:length(cumulative_variance), CumulativeVariance = cumulative_variance
  ),
  aes(x = PC, y = CumulativeVariance)
) +
  geom_line(aes(group = 1), color = "blue") +
  geom_point(color = "blue", size = 3) +
  labs(
    title = "Cumulative Variance Explained",
    x = "Principal Component",
    y = "Cumulative Proportion of Variance"
  ) +
  theme_minimal()

# Display both plots side by side
grid.arrange(scree_plot, cumulative_plot, ncol = 2)
```





Make a biplot. (3 points)

```
# Convert `HeartDisease` to a factor for coloring
pca_transformed$HeartDisease <- factor(
  pca_transformed$HeartDisease,
  levels = c(0, 1),
  labels = c("No Heart Disease", "Has Heart Disease")
)

# Create the PCA biplot
ggbiplot(
  pca_result,
  obs.scale = 1,
  var.scale = 1,
  groups = pca_transformed$HeartDisease,
) +
  scale_color_manual(values = c("green", "red")) +
  labs(
    title = "PCA Biplot Colored by Heart Disease Status",
    color = "Heart Disease Status"
  ) +
  theme_minimal()
```



Discuss your observations. (9 points)

Scree plot

- The scree plot shows a steep drop in the proportion of variance explained till the first four PCs. After this, the drop is gradual. Hence, using the elbow method, most of the information is captured by the first four PCs.
- The cumulative proportion of variance explained is above 80% taking the first seven PCs.
- So, if dimensionality reduction is the goal, then considering the first 4 to 7 PCs might be an optimal trade-off.

Biplot

- The red points (Has Heart Disease) and the green points (No Heart Disease) are somewhat separated, especially along the first PC. This means that PC1 carries significant information distinguishing heart disease status.
- We can see that the variables `Age`, `ExerciseAngina`, `ChestPainType`, `MaxHR`, and `ST_Slope` are in the same direction as PC1. And as PC1 is somewhat able to separate people with heart disease from people who do not, these variables may play a significant role in this separation. Further, `Age`, `ExerciseAngina`, and `ChestPainType` are associated positively with PC1 (increasing them may contribute towards having heart disease) whereas `MaxHR`, and `ST_Slope` are negatively associated (increasing them may contribute towards not having heart disease).