Homework 2. PCA. (60 Points)

Puja Kumari

2023-10-08

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

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

Dividing it by φ_{i1}:

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

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

$$\hat{\beta_0} + r_i = \frac{z_{i1}}{\varphi_{i1}}$$

$$\frac{\varphi_{i2}}{\varphi_{i1}} = -\hat{\beta_1}$$

Answer: *(just yes or no)*

No.

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

here should be the answer. help yourself with a plot

The difference between linear regression coefficients optimization and the first PCA calculation lies in their objectives and methods:

Objective:

Linear Regression: The objective of linear regression is to find the coefficients (β_0 and β_1 in the simple linear regression case) that minimizes the sum of squared residuals (i.e., the vertical distances between data points and the regression line).

PCA: The objective of PCA is to find the principal components (eigenvectors) that maximize the variance of the data when projected onto these components. PCA is not focused on modeling relationships between specific input features and output but rather on finding a new orthogonal basis that captures the most significant directions of variance in the data.

Method:

Linear Regression: Linear regression estimates coefficients through techniques like least squares optimization, which minimizes the sum of squared residuals. It directly models the relationship between input features and the target variable (output).

PCA: PCA calculates the principal components by performing eigenvalue decomposition or singular value decomposition on the data's covariance or correlation matrix. It does not involve estimating coefficients that describe relationships between specific features and the target variable.

In summary, while there may be some resemblance in the equations presented, linear regression and PCA serve fundamentally different purposes and operate through different methods. Linear regression aims to model and predict the target variable based on input features, while PCA seeks to capture the underlying structure or variance in the data without specific reference to target variables or coefficients.

```r
# Sample data
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 5, 4, 5)

# Linear Regression Line
lm_model <- lm(y ~ x)
linear_regression_line <- predict(lm_model)

# PCA First Principal Component
pca_component <- prcomp(cbind(x, y))
first_principal_component <- pca_component$x[, 1]

# Create a new window for the plot
dev.new()

# Plot the data points
plot(x, y, pch = 19, col = "blue", xlab = "x", ylab = "y")

# Add Linear Regression Line
lines(x, linear_regression_line, col = "red", lwd = 2, type = "l", legend.text = "Linear Regression")
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "legend.text" is not a
## graphical parameter
```

```r
# Add PCA First Principal Component
lines(x, first_principal_component, col = "green", lwd = 2, type = "l", legend.text = "PCA 1st PC")
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "legend.text" is not a
## graphical parameter
```

```r
# Add legend
legend("topright", legend = c("Linear Regression", "PCA 1st PC"), col = c("red", "green"), lwd = 2)

# Add title
title("Linear Regression vs. 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 (https://www.stem.org.uk/resources/elibrary/resource/28452/large-datasets-stats4schools).

Obtained from https://www.openintro.org/data/index.php?data=smoking (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(tidyverse)
library(caret)
```

```
# Load data
data <- source("smoking.R")$value
```

Take a look into data

```
head(data)
```

gen...	a..	marital_status	highest_qualification	nationality	ethnicity	gross_income	regio
<fct>	<int>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
Male	38	Divorced	No Qualification	British	White	2,600 to 5,200	The N
Female	42	Single	No Qualification	British	White	Under 2,600	The N
Male	40	Married	Degree	English	White	28,600 to 36,400	The N
Female	40	Married	Degree	English	White	10,400 to 15,600	The N
Female	39	Married	GCSE/O Level	British	White	2,600 to 5,200	The N
Female	37	Married	GCSE/O Level	British	White	15,600 to 20,800	The N

6 rows | 1-8 of 12 columns

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 a new data frame with selected columns
selected_data <- data[, c("smoke", "gender", "age", "marital_status", "highest_qualification",
"gross_income")]

# View the first few rows of the new data frame to check
head(selected_data)
```

smoke <fct>	gender <fct>	age <int>	marital_status <fct>	highest_qualification <fct>	gross_income <fct>
No	Male	38	Divorced	No Qualification	2,600 to 5,200
Yes	Female	42	Single	No Qualification	Under 2,600
No	Male	40	Married	Degree	28,600 to 36,400
No	Female	40	Married	Degree	10,400 to 15,600
No	Female	39	Married	GCSE/O Level	2,600 to 5,200
No	Female	37	Married	GCSE/O Level	15,600 to 20,800

6 rows

2.2 Omit all incomplete records.(3 points)

```
# Omit rows with incomplete records
cleaned_data <- na.omit(selected_data)

# View the first few rows of the cleaned data to check
head(cleaned_data)
```

smoke <fct>	gender <fct>	age <int>	marital_status <fct>	highest_qualification <fct>	gross_income <fct>
No	Male	38	Divorced	No Qualification	2,600 to 5,200
Yes	Female	42	Single	No Qualification	Under 2,600
No	Male	40	Married	Degree	28,600 to 36,400
No	Female	40	Married	Degree	10,400 to 15,600
No	Female	39	Married	GCSE/O Level	2,600 to 5,200
No	Female	37	Married	GCSE/O Level	15,600 to 20,800

6 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 (e.g. is_married, is_divorced). 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)

```
# Convert binary variables to numeric (0 and 1)
cleaned_data$gender <- as.numeric(cleaned_data$gender == "Male")
cleaned_data$smoke <- as.numeric(cleaned_data$smoke == "Yes")

# Convert marital_status to binary variables (e.g., is_married, is_divorced, etc.)
cleaned_data$is_married <- as.numeric(cleaned_data$marital_status == "Married")
cleaned_data$is_divorced <- as.numeric(cleaned_data$marital_status == "Divorced")
cleaned_data$is_single <- as.numeric(cleaned_data$marital_status == "Single")
cleaned_data$is_separated <- as.numeric(cleaned_data$marital_status == "Separated")
cleaned_data$is_widowed <- as.numeric(cleaned_data$marital_status == "Widowed")

# Convert ordinal variables to numeric levels
cleaned_data$gross_income <- as.numeric(factor(cleaned_data$gross_income, ordered = TRUE, 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", "Unknow
n"
)))

cleaned_data$highest_qualification <- as.numeric(factor(cleaned_data$highest_qualification, orde
red = TRUE, levels = c(
  "No Qualification", "ONC/BTEC", "GCSE/CSE", "GCSE/O Level", "A Levels", "Higher/Sub Degree",
"Degree", "Other/Sub Degree"
)))

cleaned_data_copy <- cleaned_data

# Exclude the "marital_status" column from the data frame
cleaned_data <- cleaned_data[, !colnames(cleaned_data) %in% c("marital_status")]

# View the first few rows of the cleaned data with numeric features
head(cleaned_data)
```

sm...	gen...	a..	highest_qualification	gross_income	is_married	is_divorced	is_single	is_sepa
<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
0	1	38	1	2	0	1	0	
1	0	42	1	1	0	0	1	
0	1	40	7	7	1	0	0	
0	0	40	7	4	1	0	0	
0	0	39	4	2	1	0	0	
0	0	37	4	5	1	0	0	

6 rows | 1-9 of 10 columns

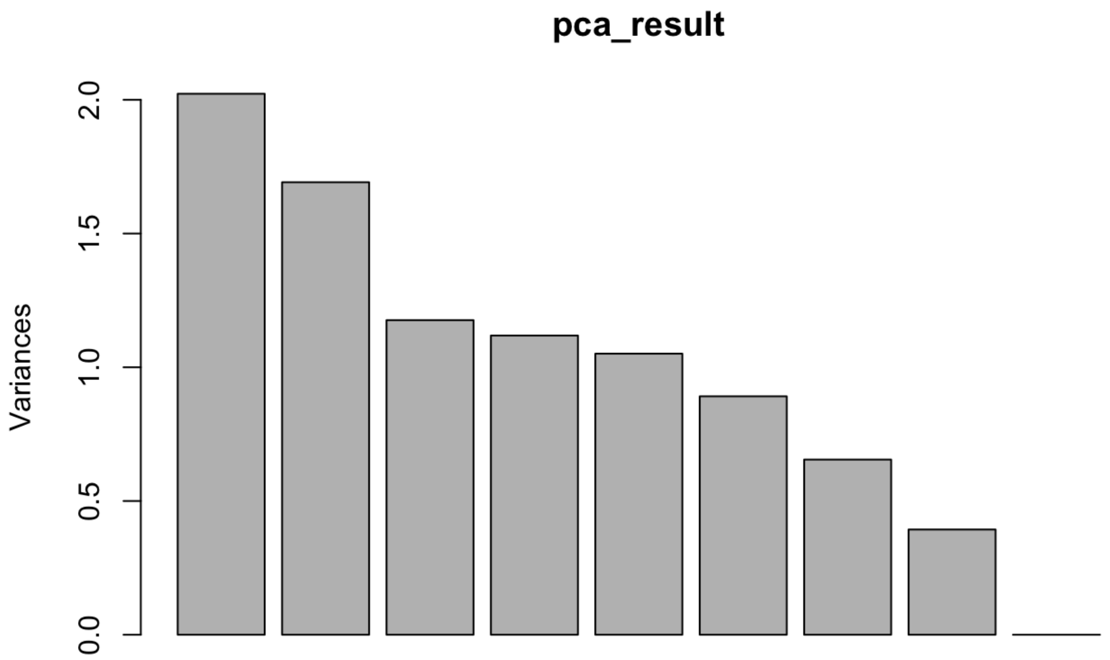
2.4. Do PCA on all columns except smoking status. (3 points)

```
# Exclude the "smoke" column from the data frame
data_for_pca <- cleaned_data[, !colnames(cleaned_data) %in% c("smoke")]

# Perform PCA
pca_result <- prcomp(data_for_pca, center = TRUE, scale. = TRUE)  # Center and scale the data

# Assign feature names as col names to the loading matrix
#colnames(pca_result$rotation) <- colnames(data_for_pca)

# View a summary of the PCA results
summary(pca_result)
```

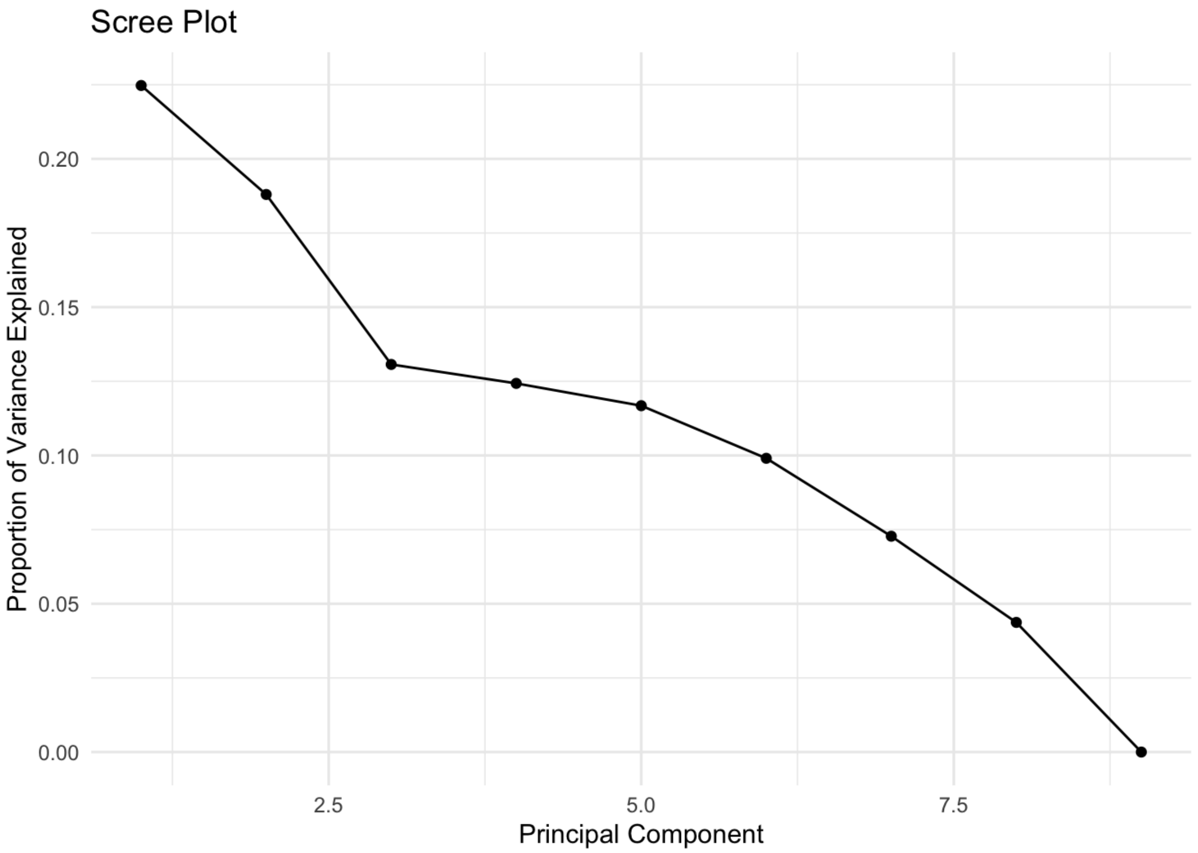
```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.4222 1.3007 1.0845 1.0577 1.0252 0.94413 0.80925
## Proportion of Variance 0.2247 0.1880 0.1307 0.1243 0.1168 0.09904 0.07276
## Cumulative Proportion  0.2247 0.4127 0.5434 0.6677 0.7845 0.88352 0.95629
##                            PC8      PC9
## Standard deviation     0.62724 2.16e-15
## Proportion of Variance 0.04371 0.00e+00
## Cumulative Proportion  1.00000 1.00e+00
```

2.5 Make a scree plot (3 points)

```
# Plot the PCA results
plot(pca_result)
```



```
# Create a data frame with explained variances
data_explained_var <- data.frame(
  Principal_Component = 1:length(pca_result$sdev),  # Use pca_result$sdev for explained variance
s
  Variance_Explained = (pca_result$sdev^2) / sum(pca_result$sdev^2)  # Calculate explained varia
nces
)

# Create a scree plot using ggplot2
library(ggplot2)

ggplot(data_explained_var, aes(x = Principal_Component, y = Variance_Explained)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(title = "Scree Plot", x = "Principal Component", y = "Proportion of Variance Explained")
```



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

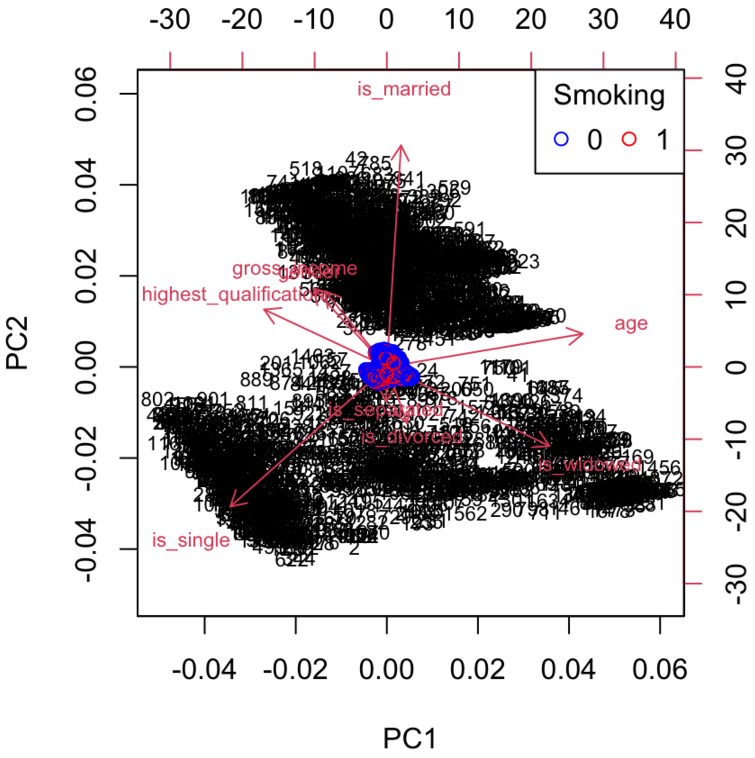
At the beginning of the plot, there is a steep decline in eigenvalues. This corresponds to the first few principal components, which capture most of the variance in the data. These components are the most important. After the initial steep drop, the eigenvalues start to level off. The point where this leveling-off occured is often referred to as the “elbow” of the scree plot.Beyond the elbow point, the eigenvalues continue to decrease gradually but at a slower rate. These components capture less and less variance. So, from the scree plot we can say that PC7, PC8,PC9 is not capturing variance at all.So, we can discard last three. We can keep 6 dimensions and reduce it’s dimension by 3. Till 6 only cummlative propotion exceeding 80% which implies more that 80 % of the dependent variable.

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

```
# Ensure "smoke" is treated as a factor
cleaned_data$smoke <- factor(cleaned_data$smoke, levels = c("0", "1"))

# Create a biplot
biplot(pca_result, cex = 0.7)

point_colors <- ifelse(cleaned_data$smoke == "0", "blue", "red")

points(pca_result$x[,1], pca_result$x[,2], col = point_colors)

# Color the points by the "smoking" field
points(pca_result$x[,1], pca_result$x[,2], col = point_colors)
print(levels(cleaned_data$smoke))
```

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

```
# Add a legend for the colors
legend("topright", legend = levels(cleaned_data$smoke),  col = c("blue", "red"), pch = 1, title
= "Smoking",horiz = TRUE, x.intersp = 1)
```



Comment on observed biplot.

Longer vectors indicate variables that contribute more to the variance explained by the principal components. In our biplot it's is_married, is_single and age play an important role in contributing variance.The gross income and gender are in the same direction so, they are positively correlated. The age and is_single vectors are in opposite direction. So, its negatively correlated. So, from the biplot we can say that is_seperated is not capturing variance at all.So, we can discard this. We can keep 6 dimensions and reduce by 3.There is no outlier as such. Highest_qualification, is_married and is_single they are perpendicular to each other so they are not correlated. The data points are more cluster is in the direction of is_married. So, therefore is_married is able to explain most of the variance.is_seperated is the least PCA explaining very less variance followed by is_divorced.

Can we use first two PC to discriminate smoking?

Biplot is showing overlapping points for the two smoking categories (0 and 1) in the first two principal components (PCs), it suggests that these PCs may not be effective at discriminating between the smoking categories based on the current features and data distribution. Overlapping points indicate that there is no clear separation in the PC space for the smoking status.

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

```r
# Assuming pca_result contains PCA results
loadings <- pca_result$rotation

# Define descriptive names for the PCs based on the top-loading features
pc_names <- character(length = ncol(loadings))

for (i in 1:ncol(loadings)) {
  top_loading_feature <- rownames(loadings)[which.max(abs(loadings[, i]))]
  pc_names[i] <- paste("PC", i, sep = "")
}

# Assign the descriptive names to the principal components
rownames(loadings) <- pc_names

# View the loading matrix with descriptive PC names
print(loadings)
```

```
##               PC1        PC2        PC3         PC4           PC5         PC6
## PC1 -0.183079538  0.2448819 -0.5154379 -0.24935358  0.0746410222 -0.55113873
## PC2  0.579718430  0.1071864 -0.1939163 -0.11358528  0.0014442241  0.03392484
## PC3 -0.363141523  0.1859614  0.0144142 -0.14313583  0.0108278096  0.71868413
## PC4 -0.217051029  0.2542110 -0.4347801 -0.44903621  0.1712480870  0.21398517
## PC5  0.042016783  0.7170317  0.1836246  0.24946200 -0.0991187595 -0.05536302
## PC6  0.060165645 -0.1816610  0.4210856 -0.76803915 -0.2254262680 -0.11820587
## PC7 -0.463038550 -0.4517977 -0.2392688  0.22353747 -0.1579217392 -0.08592558
## PC8 -0.001658562 -0.1121402  0.2009810 -0.01694426  0.9377647055 -0.05236001
## PC9  0.481282397 -0.2559440 -0.4458752  0.02069461  0.0001842107  0.32501725
##              PC7         PC8           PC9
## PC1  0.50291590 -0.12810813  3.989266e-16
## PC2  0.13516469  0.76322342  6.206608e-16
## PC3  0.53393447  0.10555018 -3.086599e-16
## PC4 -0.64750853  0.05670584 -5.058197e-17
## PC5 -0.07624650 -0.03268211 -6.069428e-01
## PC6  0.05691271 -0.03190025 -3.565611e-01
## PC7 -0.05157323  0.40088515 -5.277920e-01
## PC8  0.06876035  0.05392688 -2.386652e-01
## PC9  0.08951215 -0.47012676 -4.110463e-01
```

From the loading matrix we can say that is_married, is_single and age play an important role in contributing variance.

PC1 (Age and Marital Status): This component appears to be primarily related to age and marital status, with a strong negative loading for age and loadings indicating relationships with various marital statuses.

PC2 (Marital Status and Gender): PC2 seems to capture variations related to marital status and gender, with positive loadings for marital status variables and gender.

PC5 (Age and Smoking Habits): This component might be related to age and smoking habits, with a strong positive loading for age and a negative loading for smoking-related variables.

PC6 (Age and Highest Qualification): PC6 could be associated with age and highest qualification, with positive loadings for age and a notable loading for the highest qualification variable.

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)

From the loading matrix and the interpretation of the principal components, it appears that the mapping of certain categorical variables to numeric values may need to be revisited to better reflect the relationships between these variables and the principal components. Here are some potential improvements:

Marital Status: In PC1 and PC2, marital status variables (is_married, is_single, is_divorced) play a significant role. It might be beneficial to revisit how these categories are mapped to numeric values. Consider using a one-hot encoding approach, where each category is represented by a binary indicator variable (0 or 1). This would avoid the assumption of ordinality and capture the categorical nature of marital status more accurately.

Gender: PC3 seems to be related to gender. The current mapping of gender to numeric values may be fine, as it is a binary variable (0 or 1). However, it's essential to ensure that the encoding accurately reflects the meaning of gender in your dataset.

Smoking Habits: PC5 appears to be related to age and smoking habits. If the current mapping of smoking-related variables to 0 and 1 represents non-smoker and smoker, respectively, this seems reasonable. However, ensure that the encoding aligns with the definitions in your dataset.

Highest Qualification: In PC6, highest qualification variables are involved. Depending on the original categories, you might consider revisiting the numeric encoding to ensure that higher qualifications receive higher numeric values, reflecting the ordinal nature of education levels.

Gross Income: Gross income variables are involved in PC7. Similar to highest qualification, ensure that the mapping of income categories to numeric values respects the ordinality of income levels.

Overall, the choice of how to encode categorical variables into numeric values should align with the underlying meaning of the categories. Using one-hot encoding for non-ordinal categorical variables and maintaining ordinality for ordinal variables is a common practice to accurately represent the relationships between variables and principal components

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

Certainly! Based on the suggestions provided in 2.8, I will redo the PCA by revisiting the encoding of categorical variables and then create a biplot. Let's proceed step by step:

Step 1: Revisit the encoding of categorical variables, especially marital status, to ensure it accurately reflects the relationships between categories.

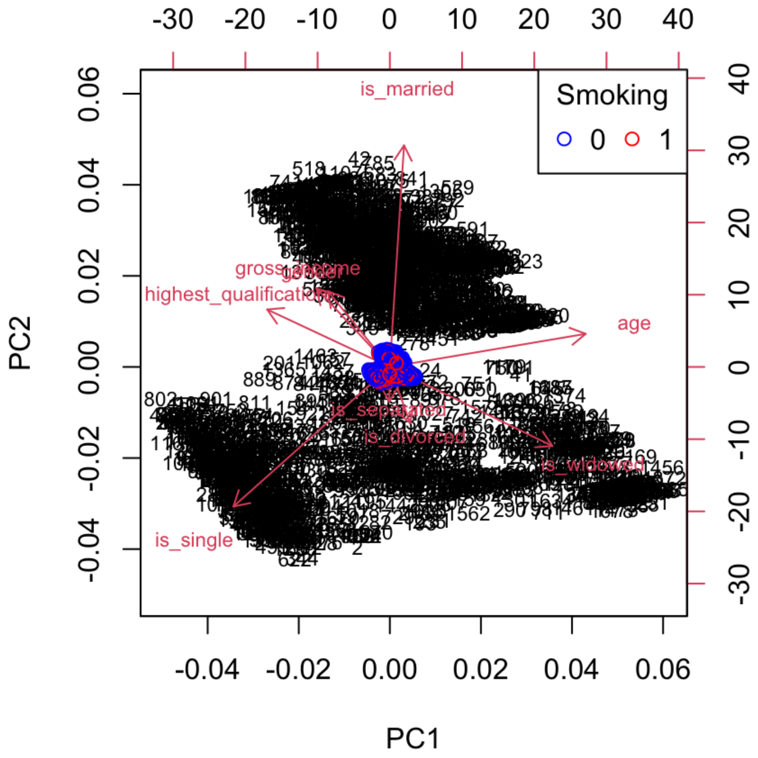
Step 2: Perform PCA on the revised dataset.

Step 3: Create a biplot to visualize the principal components.

```
# Assuming 'marital_status' is a categorical variable, use one-hot encoding
cleaned_data_copy <- cleaned_data_copy %>%
  mutate(is_married = as.numeric(marital_status == "Married"),
         is_divorced = as.numeric(marital_status == "Divorced"),
         is_single = as.numeric(marital_status == "Single"),
         is_separated = as.numeric(marital_status == "Separated"),
         is_widowed = as.numeric(marital_status == "Widowed"))

# Convert 'smoke' column to numeric
cleaned_data_copy$smoke <- as.numeric(cleaned_data_copy$smoke)

# Exclude 'marital_status' and 'smoke' columns from the data frame
data_for_pca <- cleaned_data_copy[, !colnames(cleaned_data_copy) %in% c("marital_status", "smok
e")]

# Perform PCA
pca_result <- prcomp(data_for_pca, center = TRUE, scale. = TRUE)  # Center and scale the data

summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.4222 1.3007 1.0845 1.0577 1.0252 0.94413 0.80925
## Proportion of Variance 0.2247 0.1880 0.1307 0.1243 0.1168 0.09904 0.07276
## Cumulative Proportion  0.2247 0.4127 0.5434 0.6677 0.7845 0.88352 0.95629
##                            PC8      PC9
## Standard deviation     0.62724 2.16e-15
## Proportion of Variance 0.04371 0.00e+00
## Cumulative Proportion  1.00000 1.00e+00
```

```
# Create a biplot
biplot(pca_result, cex = 0.7)

# Color the points by the 'smoke' field
point_colors <- ifelse(cleaned_data_copy$smoke == 0, "blue", "red")

points(pca_result$x[,1], pca_result$x[,2], col = point_colors)

# Add a legend for the colors
legend("topright", legend = c("0", "1"),  col = c("blue", "red"), pch = 1, title = "Smoking",hor
iz = TRUE, x.intersp = 1)
```



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)

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
# Read the "us-states.csv" dataset
diabetes_data <- read.csv("diabetes.csv")
head(diabetes_data)
```

	id	chol	stab.glu	hdl	ratio	glyhb	location	age	gender	▸
	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<chr>	<int>	<chr>	
1	1000	203	82	56	3.6	4.31	Buckingham	46	female	
2	1001	165	97	24	6.9	4.44	Buckingham	29	female	
3	1002	228	92	37	6.2	4.64	Buckingham	58	female	
4	1003	78	93	12	6.5	4.63	Buckingham	67	male	
5	1005	249	90	28	8.9	7.72	Buckingham	64	male	
6	1008	248	94	69	3.6	4.81	Buckingham	34	male	

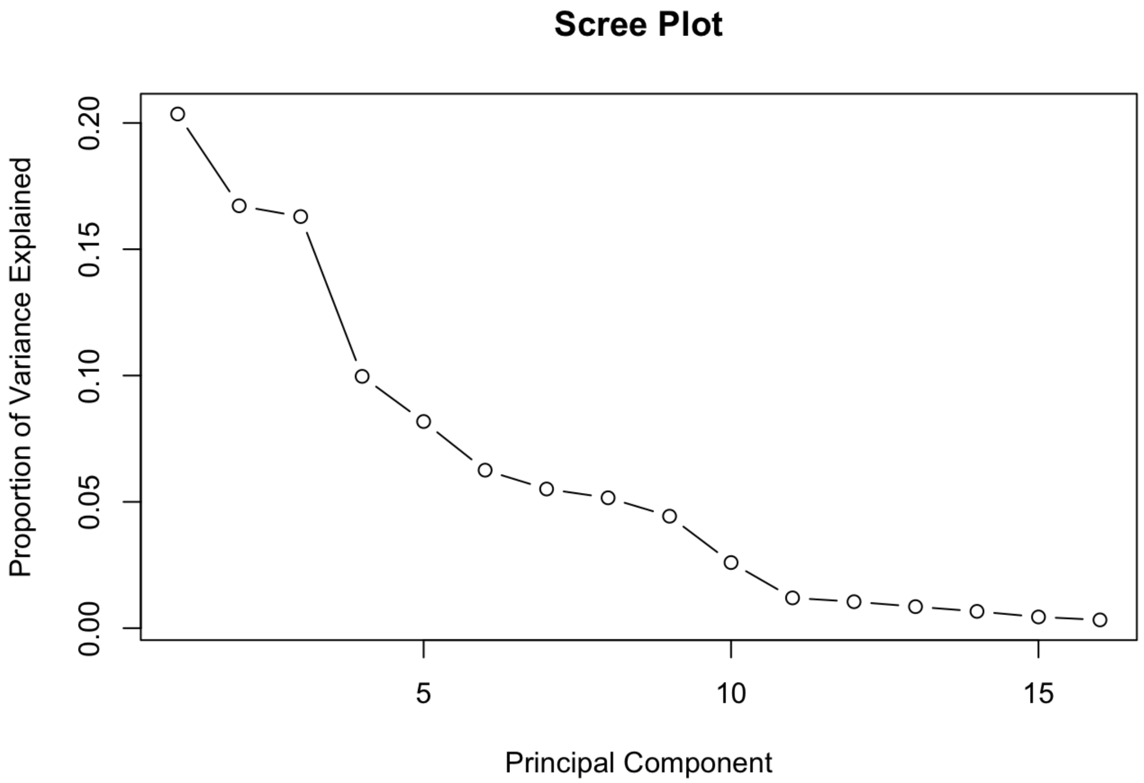
6 rows | 1-10 of 20 columns

```
#Convert a columns to proper format
# Exclude non-numeric columns and columns with missing values
diabetes_data <- diabetes_data %>% select_if(is.numeric) %>% na.omit()
```

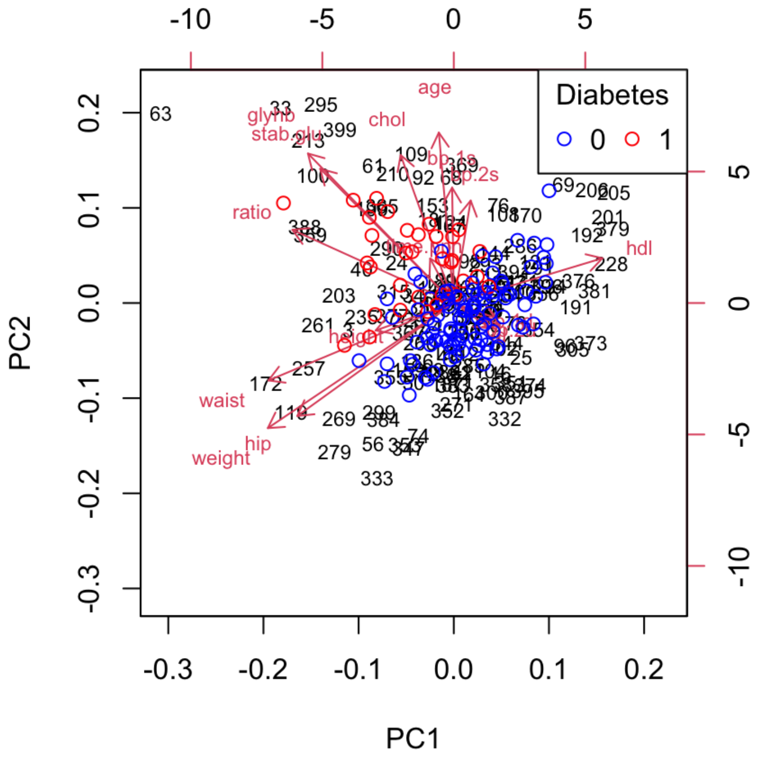
```
# Perform PCA
pca_result <- prcomp(diabetes_data, center = TRUE, scale. = TRUE)

summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4    PC5     PC6     PC7
## Standard deviation     1.8046 1.6357 1.6146 1.26284 1.1440 1.00040 0.93882
## Proportion of Variance 0.2035 0.1672 0.1629 0.09967 0.0818 0.06255 0.05509
## Cumulative Proportion  0.2035 0.3707 0.5337 0.63336 0.7152 0.77771 0.83279
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.9087 0.84210 0.64464 0.43776 0.40923 0.36914 0.32598
## Proportion of Variance 0.0516 0.04432 0.02597 0.01198 0.01047 0.00852 0.00664
## Cumulative Proportion  0.8844 0.92872 0.95469 0.96667 0.97714 0.98565 0.99229
##                           PC15    PC16
## Standard deviation     0.26643 0.22875
## Proportion of Variance 0.00444 0.00327
## Cumulative Proportion  0.99673 1.00000
```

```
# Create a scree plot (3 points)
# Calculate explained variances
explained_var <- (pca_result$sdev^2) / sum(pca_result$sdev^2)

# Scree plot
plot(explained_var, type = "b", xlab = "Principal Component", ylab = "Proportion of Variance Exp
lained", main = "Scree Plot")
```



```
# Make a biplot (3 points)

# Convert "diabetes" column to factor with appropriate levels
diabetes_data$diabetes <- factor(ifelse(diabetes_data$glyhb >= 6.5, "1", "0"), levels = c("0",
"1"))

# Perform PCA on selected variables (assuming you have already prepared your dataset)

# Create a biplot
biplot(pca_result, cex = 0.7)

# Color the points by the "diabetes" field
point_colors <- ifelse(diabetes_data$diabetes == "0", "blue", "red")

# Plot the points with the selected colors
points(pca_result$x[, 1], pca_result$x[, 2], col = point_colors)

# Add a legend for the colors
legend("topright", legend = levels(diabetes_data$diabetes), col = c("blue", "red"), pch = 1, tit
le = "Diabetes", horiz = TRUE, x.intersp = 1)
```



Discuss your observations (9 points)

Dataset Description:

Several hundred rural African-American patients were included. The diabetes.csv file contains the raw data of all patients, including those with missing data

If their hemoglobin A1 c was 6.5 or greater they were labelled with diabetes = yes [column = “glyhb”].

The “diabetes” column is created based on the condition that if the hemoglobin A1c (glyhb) is greater than or equal to 6.5, it is labeled as “1” (indicating diabetes), and otherwise, it is labeled as “0” (indicating no diabetes).

The points are colored based on the “diabetes” field, with “blue” for no diabetes and “red” for diabetes.

Biplot is showing overlapping points for the two diabetes categories (0 and 1) in the first two principal components (PCs), it suggests that these PCs may not be effective at discriminating between the diabetes categories based on the current features and data distribution. Overlapping points indicate that there is no clear separation in the PC space for the diabetes status.

From the biplot we can say that glynb contributing the maximum variance, folllowed by stab glucose then age. stab glucose and glynb are in the same direction it means that they are positively correlated.

Waist, hip and weight are in the same direction, so they are positively correlated.