Lab 3

Nadzeya_Boyeva

2024-11-14

Task 1

```
library(ggplot2)
```

```
df = read.table("data9_lab3.txt")
```

Step 1. Load data

```
X <- as.matrix(df)
N <- dim(X)[1]
K <- dim(X)[2]
```

Step 2. Normalize data

$$x'_{ij} = \frac{x_{ij} - \overline{X}_j}{\sigma(X_j)}$$

```
X_prime <- apply(X, 2, function(col)(col - mean(col))/sd(col))
```

Step 3. Build covariance matrix

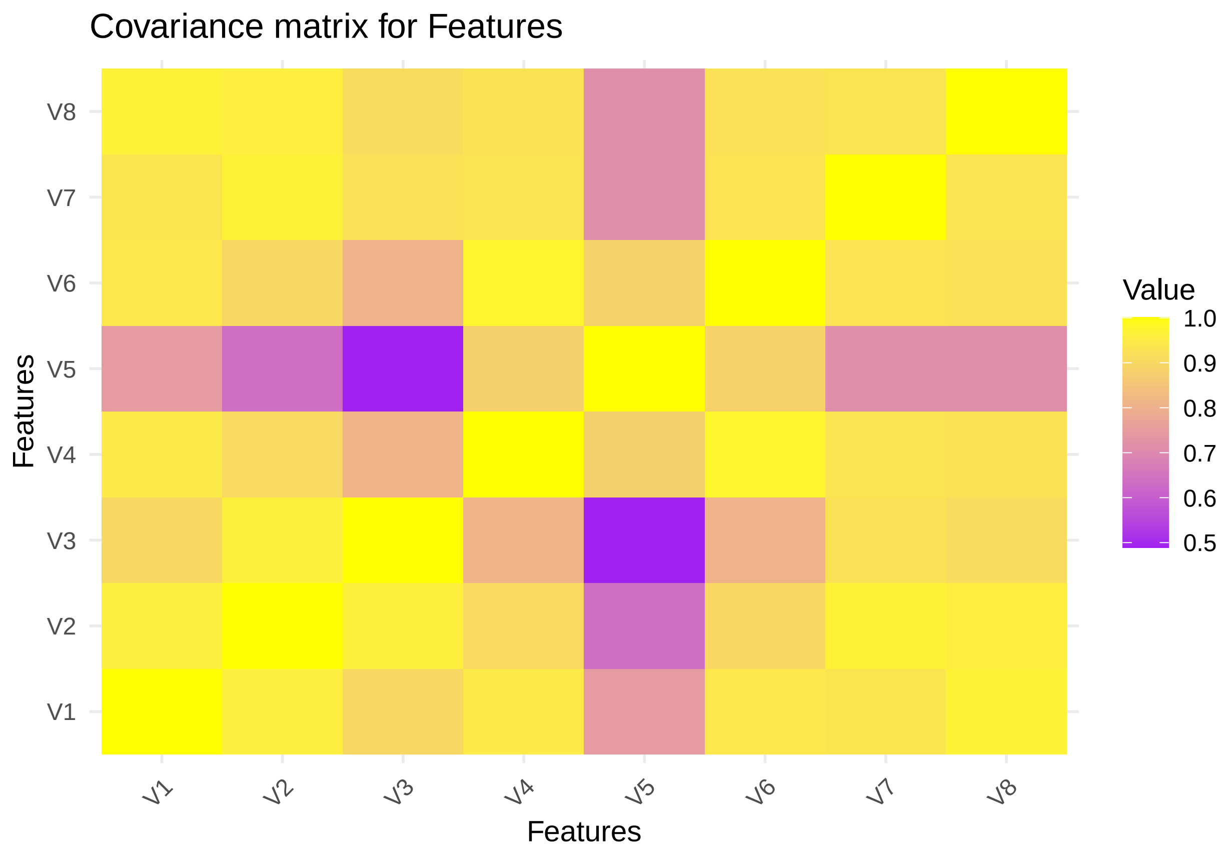
$$\mathbf{Cov} = \mathbf{R} = \frac{\mathbf{X}'^T\mathbf{X}'}{N-1}$$

```
R <- (t(X_prime) %*% X_prime)/(N - 1)
print(R)
```

```
##           V1        V2        V3        V4        V5        V6        V7
## V1 1.0000000 0.9582873 0.8976661 0.9451660 0.7457571 0.9399074 0.9341541
## V2 0.9582873 1.0000000 0.9615689 0.9028945 0.6354360 0.8984859 0.9673927
## V3 0.8976661 0.9615689 1.0000000 0.8092452 0.4892329 0.8063334 0.9238153
## V4 0.9451660 0.9028945 0.8092452 1.0000000 0.8763460 0.9778561 0.9317092
## V5 0.7457571 0.6354360 0.4892329 0.8763460 1.0000000 0.8857045 0.7151661
## V6 0.9399074 0.8984859 0.8063334 0.9778561 0.8857045 1.0000000 0.9284302
## V7 0.9341541 0.9673927 0.9238153 0.9317092 0.7151661 0.9284302 1.0000000
## V8 0.9674998 0.9565865 0.9086007 0.9255137 0.7154778 0.9238084 0.9299144
##           V8
## V1 0.9674998
## V2 0.9565865
## V3 0.9086007
## V4 0.9255137
## V5 0.7154778
## V6 0.9238084
## V7 0.9299144
## V8 1.0000000
```

```r
R_df <- as.data.frame(as.table(R))
colnames(R_df) <- c("Row", "Column", "Value")

ggplot(R_df, aes(x = Column, y = Row, fill = Value)) +
  geom_tile() +
  scale_fill_gradient(low = "purple", high = "yellow") +
  theme_minimal() +
  labs(title = "Covariance matrix for Features",
       x = "Features",
       y = "Features") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



$$d = N \sum_{i=1}^{K} \sum_{j=i+1}^{K} r_{ij}^2$$

```
updiag_elements_sum <- 0

for (i in 1:(K-1)) {
  for (j in (i+1):K) {
    updiag_elements_sum <- updiag_elements_sum + R[i,j]^2
  }
}

d <- N * updiag_elements_sum
d
```

```
## [1] 2172.775
```

```
degrees_of_freedom <- K*(K - 1)/2
alpha <- 0.05
critical_value <- qchisq(1 - alpha,
                         degrees_of_freedom)
critical_value
```

```
## [1] 41.33714
```

Since d significantly exceeds critical_value threshold, usage of PCA is appropriate in this case.

Step 4. Calculate A and L matrices

Now we compute eigenvalues and eigenvectors of features covariance matrix.

$$\mathbf{RA} = \mathbf{AL}$$

Matrix L – diagonal matrix of eigenvalues.

Matrix A – eigenvectors.

```
eig <- eigen(R)
A <- eig$vectors
L <- diag(eig$values)
```

```
RA <- as.matrix(R %*% A)
AL <- as.matrix(A %*% L)
attr(RA, "dimnames") <- NULL
attr(AL, "dimnames") <- NULL

all.equal(RA, AL)
```

```
## [1] TRUE
```

Step 5. Calculate objects projections on PCs

$$\mathbf{Z} = \mathbf{X}^{'}\mathbf{A}$$

```r
Z <- X_prime %*% A
dim(Z)
```

```
## [1] 100   8
```

Task 2

Create a function to run PCA algorithm on the dataframe given.

```r
check_cov <- function(K, N, R, alpha=0.05) {

  updiag_elements_sum <- 0
  for (i in 1:(K-1)) {
    for (j in (i+1):K) {
      updiag_elements_sum <- updiag_elements_sum + R[i,j]^2
    }
  }
  d <- N * updiag_elements_sum

  degrees_of_freedom <- K*(K - 1)/2
  critical_value <- qchisq(1 - alpha,degrees_of_freedom)
  critical_value

  if (d <= critical_value) {
    return(1)
  } else {
    return(0)
  }
}


pca_alg <- function(df) {
  X <- as.matrix(df)
  N <- dim(X)[1]
  K <- dim(X)[2]
  X_prime <- apply(X, 2, function(col)(col - mean(col))/sd(col))
  R <- (t(X_prime) %*% X_prime)/(N - 1)

  # Check if PCA usage is appropriate
  if (check_cov(K, N, R) == 1) {
    stop("PCA usage is not appropriate: covariance matrix looks similar to identity matrix")
  }

  eig <- eigen(R)
  A <- eig$vectors
  L <- diag(eig$values)

  RA <- as.matrix(R %*% A)
  AL <- as.matrix(A %*% L)

  Z <- X_prime %*% A

  return(Z)
}
```

```
Z_via_func <- pca_alg(df)

all.equal(Z, Z_via_func)
```

```
## [1] TRUE
```

Task 3

Step 1

Check the equality of the sums of the sample variances of the original features and the sample variances of the projections of objects onto the principal components.

```
sum(apply(Z, 2, var))
```

```
## [1] 8
```

```
sum(apply(X_prime, 2, var))
```

```
## [1] 8
```

Step 2

Determine the relative proportion of the variance that falls on the principal components. Construct a covariance matrix for projections of objects onto the principal components.

$$\alpha_j = \frac{\sigma^2(Z_j)}{\sum_{i=1}^{K} \sigma^2(Z_i)}$$

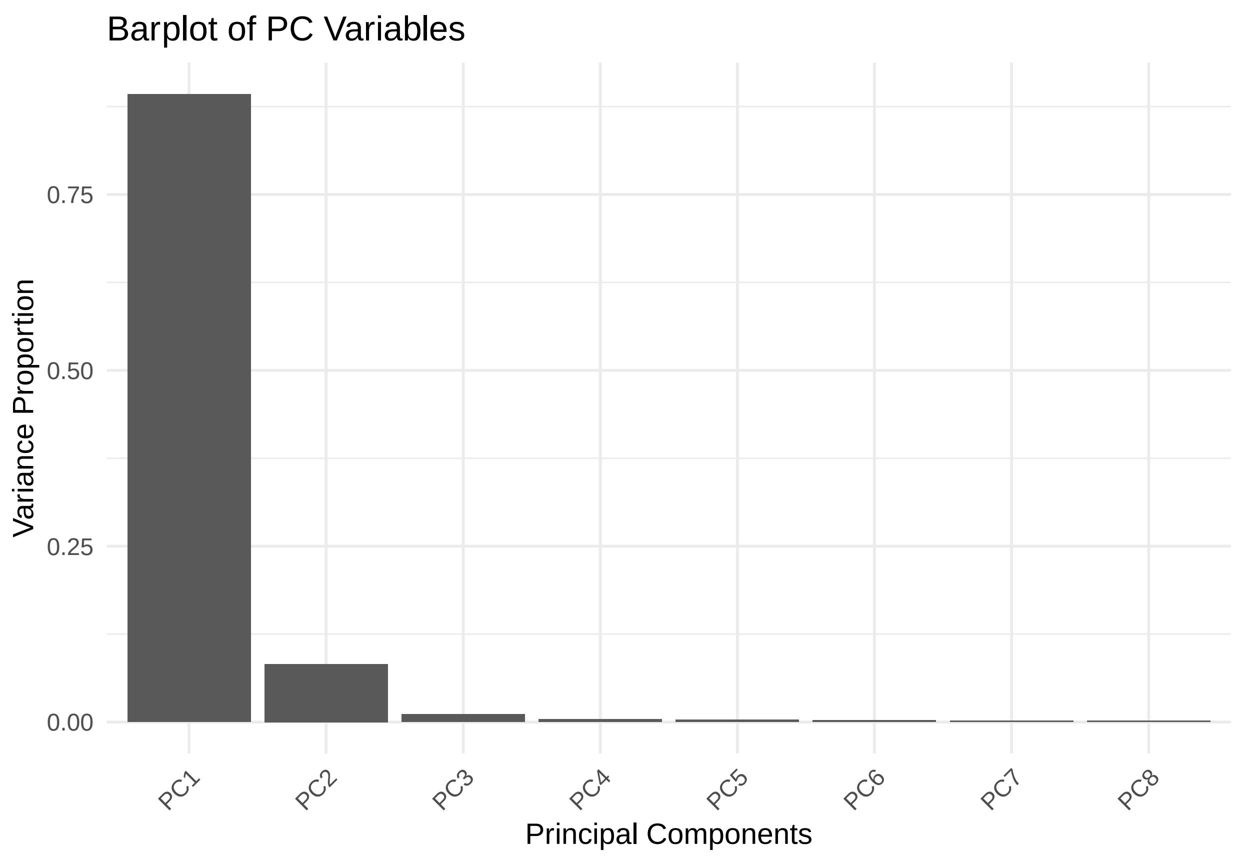
```
PC_names <- c()
PC_vars <- c()

for (i in 1:K) {
  print(paste0("Proportion of variance that falls on PC",
               i, ": ", round(diag(L)[i]/sum(diag(L)), 4)))
  PC_names <- c(PC_names, paste0("PC", i))
  PC_vars <- c(PC_vars, diag(L)[i]/sum(diag(L)))
}
```

```
## [1] "Proportion of variance that falls on PC1: 0.8928"
## [1] "Proportion of variance that falls on PC2: 0.0825"
## [1] "Proportion of variance that falls on PC3: 0.0108"
## [1] "Proportion of variance that falls on PC4: 0.0042"
## [1] "Proportion of variance that falls on PC5: 0.0034"
## [1] "Proportion of variance that falls on PC6: 0.0026"
## [1] "Proportion of variance that falls on PC7: 0.0021"
## [1] "Proportion of variance that falls on PC8: 0.0016"
```

```
PC_vars_df <- data.frame(PC=PC_names, Var=PC_vars)
```

```
ggplot(PC_vars_df, aes(x = PC, y = Var)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Barplot of PC Variables",
       x = "Principal Components",
       y = "Variance Proportion") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Now let's construct a covariance matrix for projections of objects onto the principal components.

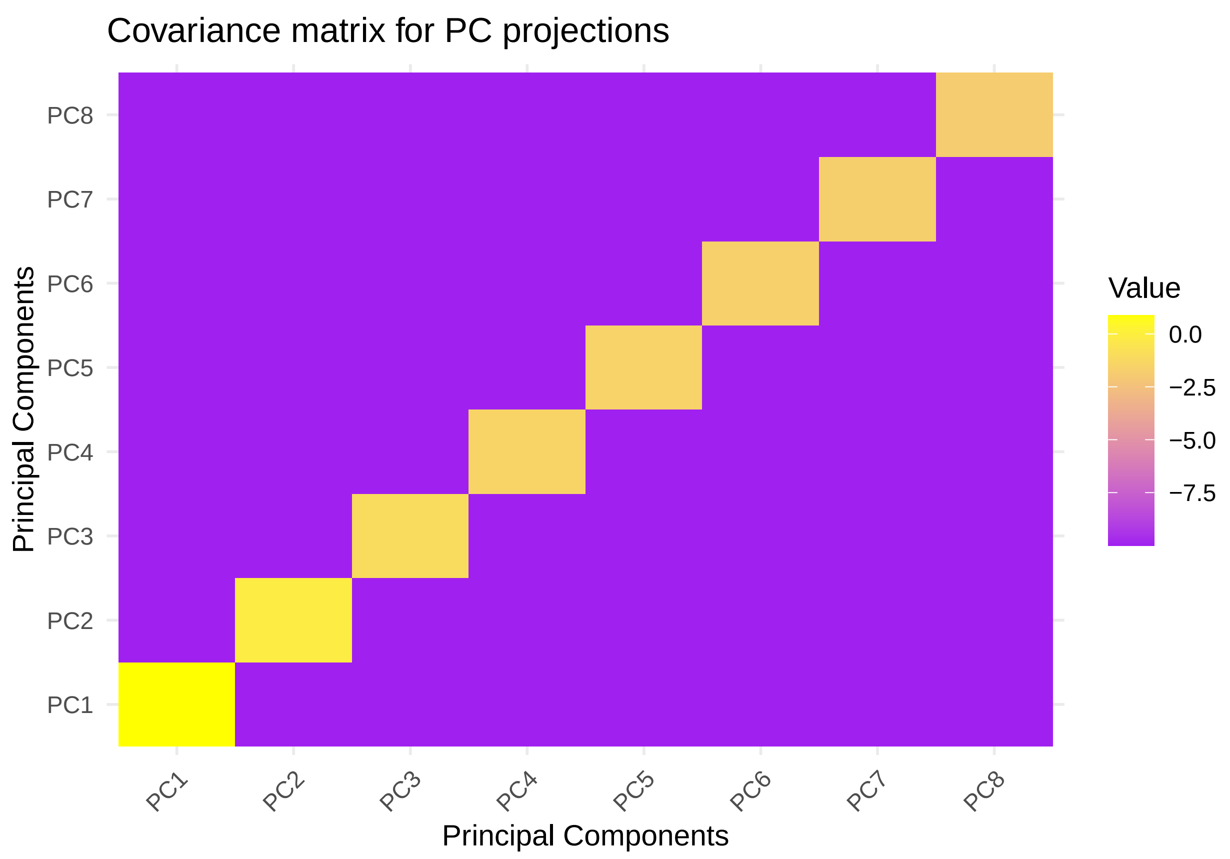
```
Z_cov <- (t(Z) %*% Z)/(N - 1)
Z_cov
```

```
##                  [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]  7.142075e+00  5.831474e-16 -1.238067e-15  1.656363e-15 -1.799907e-16
## [2,]  5.831474e-16  6.603384e-01  5.410935e-16  4.161234e-16  5.272158e-16
## [3,] -1.238067e-15  5.410935e-16  8.650084e-02  2.846348e-16 -1.396190e-16
## [4,]  1.656363e-15  4.161234e-16  2.846348e-16  3.350162e-02  5.512128e-16
## [5,] -1.799907e-16  5.272158e-16 -1.396190e-16  5.512128e-16  2.747898e-02
## [6,] -1.292457e-16  5.809046e-16  4.347673e-16  2.543823e-17 -3.734737e-16
## [7,] -2.287732e-16 -6.414622e-16 -7.665025e-16  1.461373e-16 -5.838483e-17
## [8,] -6.571623e-16 -8.813096e-16  1.854016e-15 -2.502207e-16 -9.875658e-17
##               [,6]          [,7]          [,8]
## [1,] -1.292457e-16 -2.287732e-16 -6.571623e-16
## [2,]  5.809046e-16 -6.414622e-16 -8.813096e-16
## [3,]  4.347673e-16 -7.665025e-16  1.854016e-15
## [4,]  2.543823e-17  1.461373e-16 -2.502207e-16
## [5,] -3.734737e-16 -5.838483e-17 -9.875658e-17
## [6,]  2.058287e-02 -1.415464e-16  4.777674e-16
## [7,] -1.415464e-16  1.706080e-02  5.506258e-16
## [8,]  4.777674e-16  5.506258e-16  1.246177e-02
```

```r
Z_cov_df <- as.data.frame(as.table(Z_cov))
colnames(Z_cov_df) <- c("Row", "Column", "Value")
Z_cov_df$Value <- log10(abs(Z_cov_df$Value) + 1e-10)

Z_cov_df$Column <- factor(Z_cov_df$Column,
                          labels = paste0("PC",
                                          1:length(unique(Z_cov_df$Column))))
Z_cov_df$Row <- factor(Z_cov_df$Row,
                       labels = paste0("PC",
                                       1:length(unique(Z_cov_df$Row))))

ggplot(Z_cov_df, aes(x = Column, y = Row, fill = Value)) +
  geom_tile() +
  scale_fill_gradient(low = "purple", high = "yellow") +
  theme_minimal() +
  labs(title = "Covariance matrix for PC projections",
       x = "Principal Components",
       y = "Principal Components") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

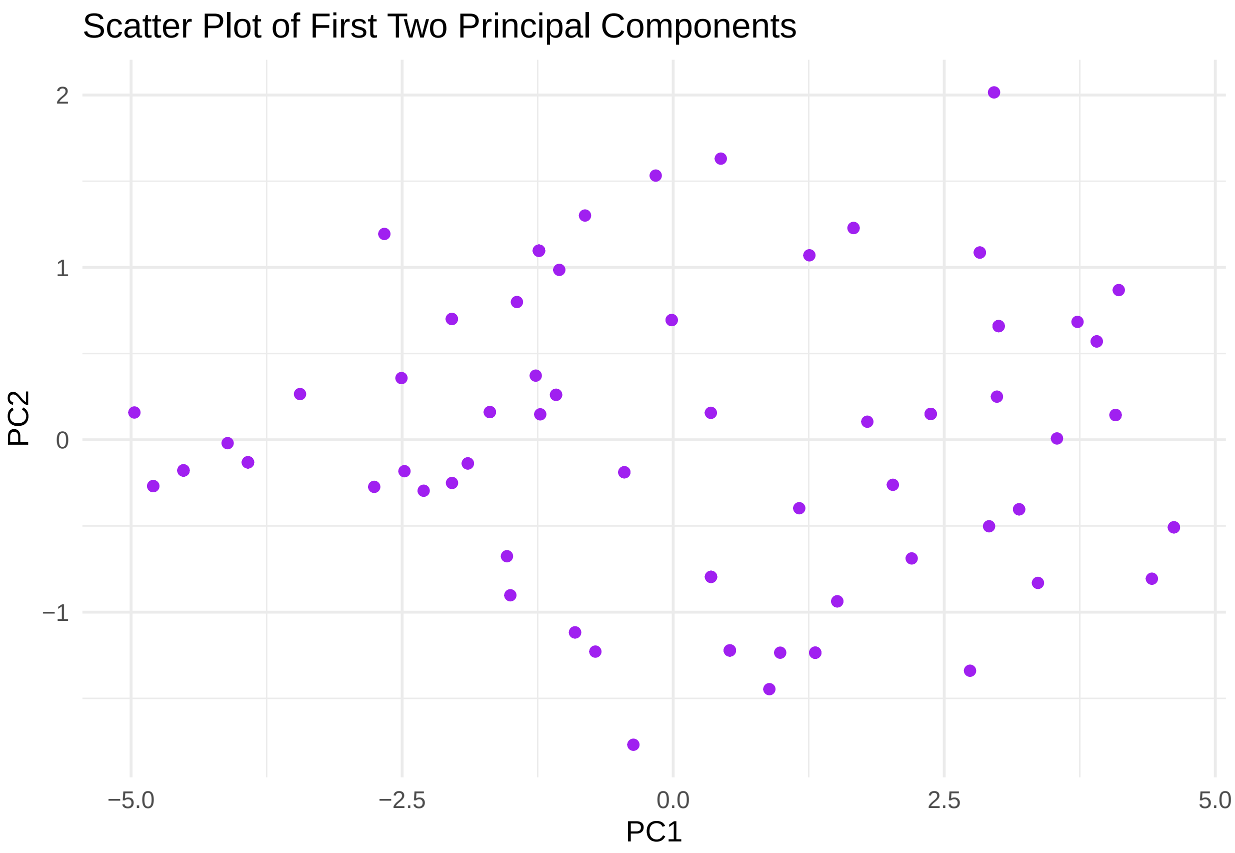


Step 3

Based on the first M = 2 principal components, construct a scatter plot.

```
PC_data <- data.frame(PC1 = Z[, 1], PC2 = Z[, 2])


ggplot(PC_data, aes(x = PC1, y = PC2)) +
  geom_point(color = "purple") +
  theme_minimal() +
  labs(title = "Scatter Plot of First Two Principal Components",
       x = "PC1",
       y = "PC2")
```



We can see that along PC1 axis data is distributed more uniformly than along PC2 axis. The reason for this is that PC1 has more variance, than PC2.