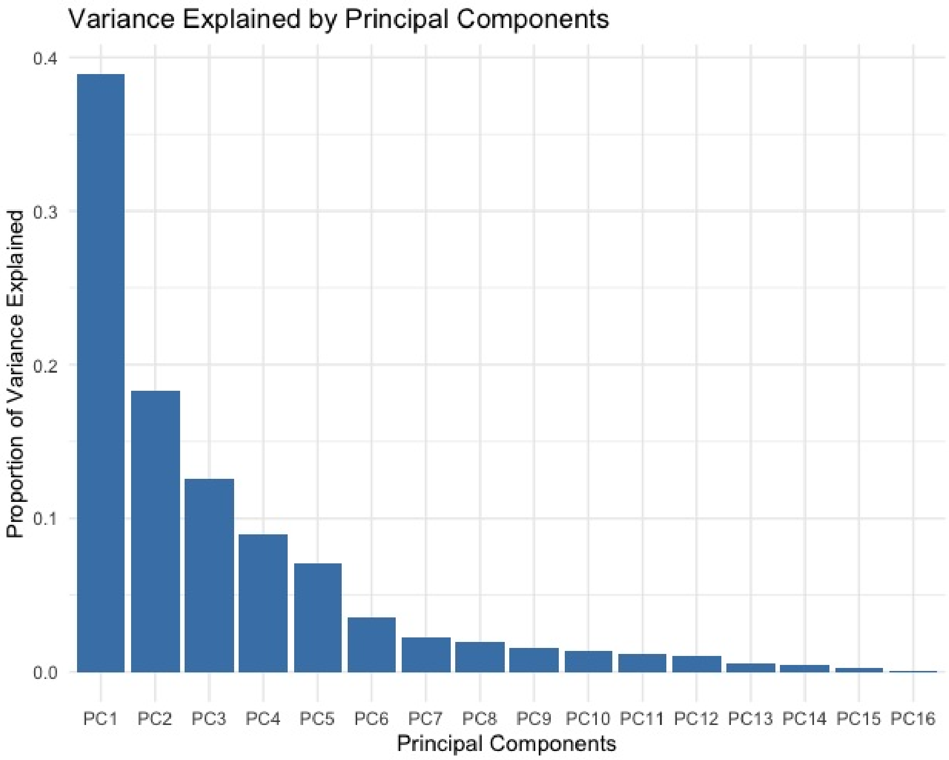
Question 9

To analyze this question, we will first perform PCA on the US Crime dataset to reduce the dimensionality of our data. After PCA, we will build our regression model with the features that explain most of the variance in the dataset. Once we have our regression model built, we will spend some time analyzing the results and comparing them to our results from running the linear regression with every feature on the dataset.

PCA on US Crime Dataset

To perform PCA on the US crime dataset, we can use the R method prcomp. We can visualize the results of the PCA algorithm like the following:



The results of this PCA tells us that the first 5 principal components tell us about 90% of the variance in the data. However, since PCs are linear combinations of the original variables, this graph does not tell us about the original predictors at all, in the next section we will add a regression component to our model and transform the model back into the original predictors.

Build and run regression model

For this, we can use the same regression model we used on Q8.2, except with the top n principal components. To do this we first use our code from the previous section and combine it with our regression code from Q8.2. Here are the results from running this code:

```
Residuals:
    Min      1Q  Median      3Q     Max 
-420.79 -185.01   12.21  146.24  447.86 
```

The distribution suggests some large residuals, which may indicate potential outliers or areas where the model fits less well.

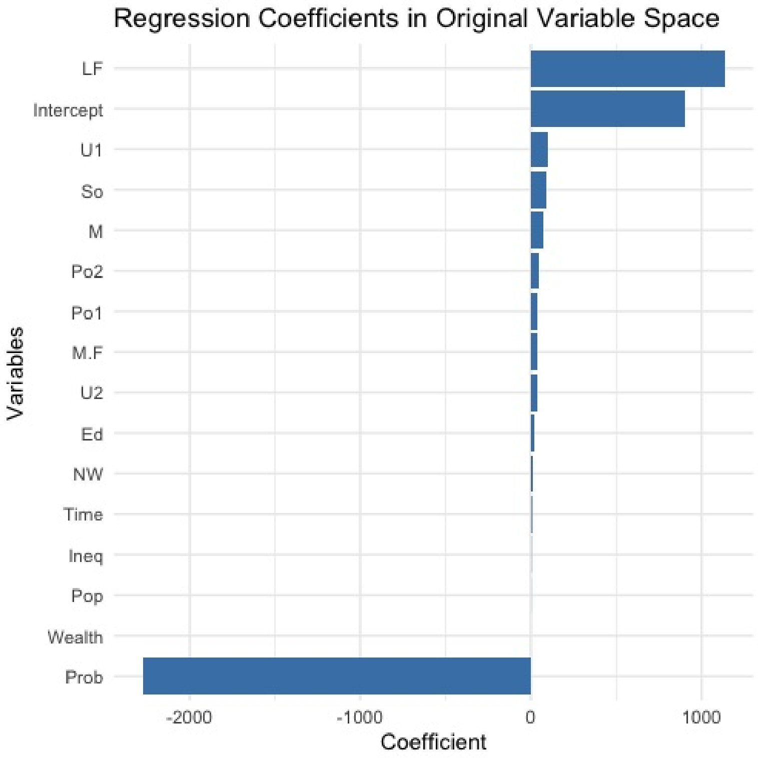
```
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   905.09      35.59  25.428  < 2e-16 ***
PCsPC1         65.22      14.67   4.447 6.51e-05 ***
PCsPC2        -70.08      21.49  -3.261  0.00224 ** 
PCsPC3         25.19      25.41   0.992  0.32725    
PCsPC4         69.45      33.37   2.081  0.04374 *  
PCsPC5       -229.04      36.75  -6.232 2.02e-07 ***
```

We can interpret each principal component from the last column above:

- PC1: Significant positive effect on the crime rate.
- PC2: Significant negative effect.
- PC3: Not statistically significant.
- PC4: Marginally significant positive effect.
- PC5: Significant negative effect.

```
Residual standard error: 244 on 41 degrees of freedom
Multiple R-squared:  0.6452,	Adjusted R-squared:  0.6019 
F-statistic: 14.91 on 5 and 41 DF,  p-value: 2.446e-08
```

Lastly, we will take a look at transforming the model back to original predictors, allowing us to interpret the model in terms of these predictors. We can visualize the results:



This plot shows that LF(Labor force) and intercept variables have a positive effect to the crime rate, while the Prob(Probability of imprisonment) variable has a dramatically negative on the crime rate.

Compare against previous model on new data

To redefine the performance of the previous model, we will assume that we used all 15 predictors, the predicted value for the same city, in this case was 155. This was not a very accurate result, given that it was less than half of the minimum value in the US Crimes dataset.

From the analysis last time, we saw that by reducing the model predictors by their p-value, which explained the more plausible results when we only included only 5 of the predictors with low p-values.

In this exercise, we applied PCA to the same crime dataset to address the limitations of the previous models. By transforming the original 15 predictors into a set of uncorrelated principal components, we reduced dimensionality while retaining the most significant information. We then built a regression model using the first few principal components that explained a substantial portion of the variance in the data.

We see that the prediction on the new dataset is close to the expected target value.

Just like the previous model, looking at R^2 value, we can see some overfitting to the data. This is not surprising since we are using the same dataset as last time which has very little rows, albeit less predictors.

Conclusion

In this analysis, we leveraged Principal Component Analysis (PCA) to enhance our understanding and modeling of the US Crime dataset. By transforming the original 15 correlated predictors into a smaller set of uncorrelated principal components, we effectively reduced the dimensionality of the data while retaining the most critical information. Specifically, the first five principal components accounted for approximately 90% of the variance in the dataset, making them strong candidates for our regression model.

Building a regression model using these five principal components allowed us to capture the underlying patterns in the data more efficiently. Upon transforming the model back into the original predictor space, we were able to interpret the influence of each original variable on the crime rate. The visualization of the regression coefficients revealed that variables such as LF (labor force participation) positively impact the crime rate, while Prob (probability of imprisonment) has a significantly negative effect.

The model that we ended up building performed similarly to the regression model with less predictors.

Appendix

```
###### HW6 CODE #########
library(ggplot2)
read_dataset <- function(data_path){
  data <- read.table(data_path, stringsAsFactors = FALSE, header = TRUE)
  return(data)
}

df <- read_dataset("HW5/uscrime.txt")


apply_pca <- function(df) {
  pca_result <- prcomp(df, scale. = TRUE)
  print(summary(pca_result))

  variance_explained <- pca_result$sdev^2 / sum(pca_result$sdev^2)

  # Prepare data frame for plotting
  variance_df <- data.frame(
    PC = factor(paste0("PC", 1:length(variance_explained)), levels = paste0("PC", 1:length(variance_explained))),
    Variance_Explained = variance_explained
  )

  # Create bar chart
  ggplot(variance_df, aes(x = PC, y = Variance_Explained)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    theme_minimal() +
    labs(
      title = "Variance Explained by Principal Components",
      x = "Principal Components",
      y = "Proportion of Variance Explained"
    )
}

apply_pca(df)

apply_pca_regression <- function(df, num_pcs) {
  # Separate predictors and response variable
  X <- df[, -ncol(df)]  # Assuming the response variable is the last column
  y <- df[, ncol(df)]
  X_scaled <- scale(X)
  scaling_params <- list(
    means = attr(X_scaled, "scaled:center"),
    sds = attr(X_scaled, "scaled:scale")
  )
  pca_result <- prcomp(X_scaled, scale. = FALSE)
  PCs <- pca_result$x[, 1:num_pcs]
  pca_reg_model <- lm(y ~ PCs)
  print(summary(pca_reg_model))
  beta_pca <- coef(pca_reg_model)[-1]
  loadings <- pca_result$rotation[, 1:num_pcs]
  beta_original_scaled <- loadings %*% beta_pca
  beta_original <- beta_original_scaled / scaling_params$sds
  intercept <- coef(pca_reg_model)[1] - sum(scaling_params$means * beta_original_scaled / scaling_params$sds)
  coef_df <- data.frame(
    Variable = c("(Intercept)", colnames(X)),
    Coefficient = c(intercept, beta_original)
  )
  ggplot(coef_df, aes(x = reorder(Variable, Coefficient), y = Coefficient)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    coord_flip() +
    theme_minimal() +
    labs(
      title = "Regression Coefficients in Original Variable Space",
      x = "Variables",
      y = "Coefficient"
    )
  return(list(
    scaling_params = scaling_params,
    pca_model = pca_result,
    regression_model = pca_reg_model,
    beta_original = beta_original,
    intercept = intercept
  ))
}

df <- read_dataset("HW5/uscrime.txt")
model_objects <- apply_pca_regression(df, num_pcs = 5)
predict_new_data <- function(new_data, model_objects) {
  scaling_params <- model_objects$scaling_params
  pca_model <- model_objects$pca_model
  regression_model <- model_objects$regression_model
  beta_original <- model_objects$beta_original
  intercept <- model_objects$intercept

  num_pcs <- length(coef(regression_model)) - 1
  X_new <- new_data[, names(scaling_params$means)]
  X_new_scaled <- sweep(X_new, 2, scaling_params$means, FUN = "-")
  X_new_scaled <- sweep(X_new_scaled, 2, scaling_params$sds, FUN = "/")

  PCs_new <- as.matrix(X_new_scaled) %*% pca_model$rotation[, 1:num_pcs]

  y_pred <- predict(regression_model, newdata = data.frame(PCs = PCs_new))
  y_pred_original <- intercept + as.matrix(X_new) %*% beta_original

  return(y_pred_original)
}
```

```
new_data <- data.frame(
  M = 14.0,
  So = 0,
  Ed = 10.0,
  Po1 = 12.0,
  Po2 = 15.5,
  LF = 0.640,
  M.F = 94.0,
  Pop = 150,
  NW = 1.1,
  U1 = 0.120,
  U2 = 3.6,
  Wealth = 3200,
  Ineq = 20.1,
  Prob = 0.04,
  Time = 39.0
)
predicted_crime_rate <- predict_new_data(new_data, model_objects)
print(predicted_crime_rate)
```