Assignment 7

1st Task: Reproduce Figure 6.20

1.1 Setup and Data

```
setwd("~/Project_Solution")
library(pls)
Credit <- read.csv("../Project_Solution/Credit.csv")

set.seed(1) # Set seed for reproducibility of PCR
```

1.2 Run Principal Component Regression

It combines your correlated variables into a smaller set of new uncorrelated "super-variables" called Principal Components.

```
pcr_model <- pcr(
  # standardize the data to a z-Distribution (scale = TRUE)
  Balance ~ ., data = Credit, scale = TRUE, validation = "CV")
```

What is done is the following:

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j, \qquad \text{with } \sum_{j=1}^{p} \phi_{jm}^2 = 1.$$

PCR Regression Model

$$Y_i = \theta_0 + \sum_{m=1}^{M} \theta_m Z_{im} + \varepsilon_i.$$

Relationship between original coefficients and PCR coefficients

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

1.3 Extract Coefficients

The raw coefficients are a 3D array: `[Variables, Balance, PCs]`. We extract "Balance" because it's our dependent variable. For each category (e.g. Income), we take the coefficient that explains the relationship between that variable and Balance.

```
coef_matrix <- pcr_model$coefficients[, "Balance", ]
```

1.4 Data Preparation for Smoothing

In Principal Component Analysis (PCA), we cannot create more components than we have input variables. Component 1 explains the most variance, while Component 11 explains the least. If we use all 11 components, our PCR model becomes exactly the same as a standard Ordinary Least Squares (OLS).

```
n_components <- 11
stretch_factor <- 10
# turns 11 data points into 110 so the plot is more beautiful (and like the textbook)
```

Repeat the columns to match the plotting grid. Transpose (`t`) is used so that the plot reads components as x and variables as y.

```
stretched_coefs <- t(coef_matrix[, rep(1:n_components, each = stretch_factor)])
```

Create the x-axis grid.

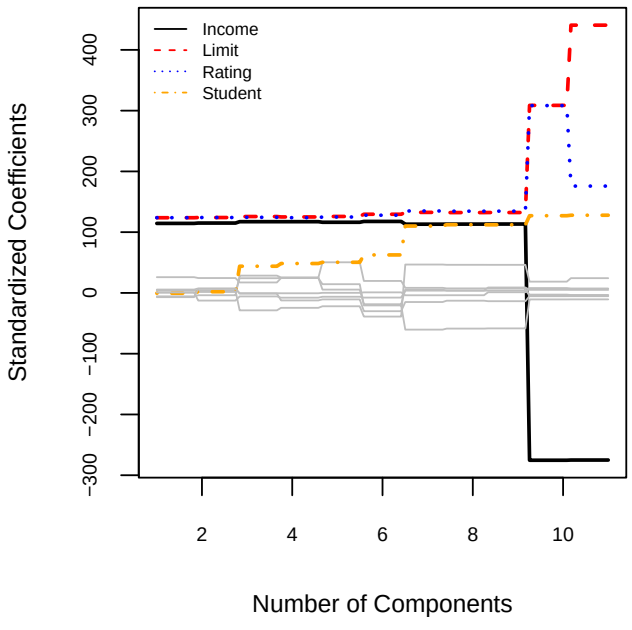
```
x_grid <- seq(1, n_components, length = n_components * stretch_factor)
```

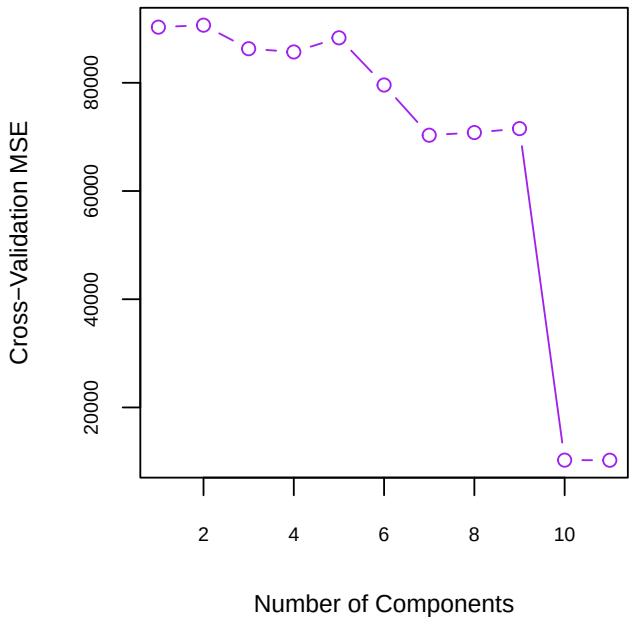
1.5 Plotting

```
par(mfrow = c(1, 2))
par(cex.axis = 0.7)
head(stretched_coefs,0)
```

```
##      Income Limit Rating Cards Age Education OwnYes StudentYes MarriedYes
##      RegionSouth RegionWest
```

```
line_cols <- c("black", "red", "blue", "grey", "grey", "grey", "grey",
               "orange","grey", "grey", "grey")
line_types <- c(1, 2, 3, 1, 1, 1, 1, 4, 1, 1, 1)
line_widths <- c(2, 2, 2, 1, 1, 1, 1, 2, 1, 1, 1)
# --- Plot Left: Standardized Coefficients ---
matplot(x_grid, stretched_coefs,
        type = 'l',
        col = line_cols,
        lty = line_types,
        lwd = line_widths,
        xlab = "Number of Components",
        ylab = "Standardized Coefficients",
        cex.lab = 0.8)
legend("topleft", names(Credit)[c(1, 2, 3, 8)], col = c("black", "red", "blue", "orange"),
       lty = 1:4, lwd = 1, bty = "n", cex = 0.6)
# --- Plot Right: Cross-Validation MSE ---
# Using 'MSEP' function to extract Mean Squared Error of Prediction
cv_mse <- MSEP(pcr_model)$val[1, 1, -1]
# $val$ (1-> Cross Validation Standard error, 1 -> Balance, -1 -> Remove Intercept )
plot(cv_mse,
     col = "purple", type = "b", ylab = "Cross-Validation MSE", xlab = "Number of Components",
     cex.lab = 0.8)
```





2nd Task: Ridge Regression & Shrinkage

2.1 Setup

Problem - ISLR (p. 239): As λ increases, the ℓ_2-norm of $\hat{\beta}_\lambda$ decreases, and so does $\frac{\|\hat{\beta}_\lambda^R\|_2}{\|\hat{\beta}^{\text{OLS}}\|_2}$."

Goal:

1. Provide a clean mathematical explanation using the Singular Value Decomposition of X
2. Prove monotone shrinkage of $\hat{\beta}_\lambda^R\|2$, explain why the shrinkage ratio tends to decrease from 1 to 0.

Setup and Definitions:

Data:

$$y \in \mathbb{R}^n, \qquad X \in \mathbb{R}^{n \times p} \quad \text{(standardized columns, centered y)}$$

L2-norm:

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2} \qquad (Formula \quad 2.1)$$

Shrinkage ratio (ISLR x-axis):

$$\text{Shrinkage}(\lambda) = \frac{\|\hat{\beta}_\lambda^R\|_2}{\|\hat{\beta}^{\text{OLS}}\|_2} \qquad (Formula \quad 2.2)$$

2.2. Explanation of Ridge Regression Norms

Definition of Lambda (λ): λ is the tuning parameter defined in shrinkage regression (Ridge), which serves to artificially minimize (shrink) the coefficients (β) Ridge Regression Formula, the general Ridge regression objective is:

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad (Formula \quad 2.3)$$

$$RSS = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 \qquad (Formula \quad 2.4)$$

2.3 ISLR Simplification (Special Case)

In the special case presented in ISLR (Equation 6.14), we assume no intercept (centered data), $N = p$, and the design matrix X is a diagonal matrix with 1s on the diagonal. This effectively means we have orthogonal predictors with unit length. Meaning in this simplified example, we assume the variables are completely separate and don't interfere with each other and they are all scaled to have the exact same strength, which basically means we can calculate each answer individually. The expanded least squares term is:

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \qquad (Formula \quad 2.5)$$

y: The target vector (response variable) containing the observed values.

x_j: The vector representing the j-th predictor (feature column)

β_j: The regression coefficient (weight) for the j-th predictor

λ: The tuning parameter (shrinkage constant) that controls the penalty strength

j, k: Indices used as counters to iterate through the predictors in the summation

$\top$: The transpose symbol, indicating the vector is flipped from column to row for multiplication.

In matrix notation, expanding the square gives:

$$y^\top y - 2\sum_j \beta_j x_j^\top y + \sum_j \sum_k \beta_j \beta_k x_j^\top x_k + \lambda \sum_j \beta_j^2 \qquad (Formula \quad 2.6)$$

Implications of Orthogonality and Unit Length: Because X is orthogonal ($x_j^\top x_k = 0$ for $j \neq k$) and has unit length ($x_j^\top x_j = 1$), the cross-terms cancel out

$$x_j^\top x_k = \begin{cases} 0 & \text{if } j \neq k \\ 1 & \text{if } j = k \end{cases}$$

This simplifies the interaction term in Formula 2.6:

$$\sum_{j,k} \beta_j \beta_k x_j^\top x_k = \sum_{j=1}^{p} \beta_j^2 \qquad (Formula \quad 2.7)$$

This then yields, when Substituting the interaction term by the simplified version:

$$y^\top y - 2\sum_j \beta_j x_j^\top y + \sum_{j=1}^{p} \beta_j^2 + \lambda \sum_j \beta_j^2 \qquad (Formula \quad 2.8)$$

Consequently, the optimization problem separates into p independent univariate problems. The Ridge estimator becomes:

$$\hat{\beta}_{j,\lambda}^R = \frac{y_j}{1+\lambda} \qquad (Formula \quad 2.9)$$

2.4 Minimization Derivation

To find the minimum, we differentiate the simplified objective function with respect to β_j. Starting from the expanded matrix form assuming orthogonal unit vectors where $x_j^\top y = y_j$ and $y^\top y = \sum y_j^2$:

$$\begin{aligned}
\mathcal{L} &= y^\top y - 2\sum_{j=1}^{p} \beta_j x_j^\top y + \sum_{j=1}^{p} \beta_j^2 + \lambda \sum_{j=1}^{p} \beta_j^2 && (Formula \quad 2.10) \\
&= \sum_{j=1}^{p} y_j^2 - 2\sum_{j=1}^{p} \beta_j y_j + \sum_{j=1}^{p} \beta_j^2 + \lambda \sum_{j=1}^{p} \beta_j^2 && \text{(Substitute scalar equivalents)} \\
&= \sum_{j=1}^{p} \left(y_j^2 - 2\beta_j y_j + \beta_j^2 \right) + \lambda \sum_{j=1}^{p} \beta_j^2 && \text{(Group terms by index } j\text{)} \\
&= \sum_{j=1}^{p} (y_j - \beta_j)^2 + \sum_{j=1}^{p} \lambda \beta_j^2 && \text{(Complete the square)} \\
&= \sum_{j=1}^{p} \left[(y_j - \beta_j)^2 + \lambda \beta_j^2 \right] && \text{(Final Sum of Independent Problems)}
\end{aligned}$$

To find the minimum, we differentiate the term inside the brackets with respect to β_j:

$$\frac{\partial \mathcal{L}}{\partial \beta_j}\left((y_j - \beta_j)^2 + \lambda\beta_j^2\right) = -2(y_j - \beta_j) + 2\lambda\beta_j = 0$$

$$\frac{\partial \mathcal{L}}{\partial \beta_j} = -2(y_j - \beta_j) + 2\lambda\beta_j = 0$$

$$-2y_j + 2(1+\lambda)\beta_j = 0$$

Rearranging for β_j yields the Ridge estimator for this special case:

$$\hat{\beta}^R_{j,\lambda} = \frac{y_j}{1+\lambda} \qquad (Formula \quad 2.11)$$

```r
par(mfrow = c(1, 2))
par(cex.axis = 0.7)

### Build standardized design matrix and response
X_raw <- model.matrix(Balance ~ ., data = Credit)[, -1]
X     <- scale(X_raw)
y     <- scale(Credit$Balance, center = TRUE, scale = TRUE)[, 1]

p   <- ncol(X)
XtX <- t(X) %*% X
Xty <- t(X) %*% y

### Ridge coefficients over lambda grid
lambdas <- exp(seq(-2, 10, length = 100))
betas   <- sapply(lambdas, function(lambda)
  solve(XtX + diag(lambda, p), Xty)
)

### Shrinkage ratio ||beta_r(lambda)|| / ||beta_OLS||
beta_ols <- solve(XtX, Xty)
norm_ols <- sqrt(sum(beta_ols^2))
ratio    <- apply(betas, 2, function(b) sqrt(sum(b^2)) / norm_ols)

### Highlight the four key variables
vars <- colnames(X)
hilite_names <- c("Income", "Limit", "Rating",
                  if ("StudentYes" %in% vars) "StudentYes" else "Student")
idx  <- match(hilite_names, vars)

line_cols   <- rep("grey", p)
line_types  <- rep(1, p)
line_widths <- rep(1, p)

cols <- c("black", "red", "blue", "orange")
lts  <- c(1, 2, 3, 4)
lwds <- c(2, 2, 2, 2)

line_cols[idx]   <- cols
line_types[idx]  <- lts
```

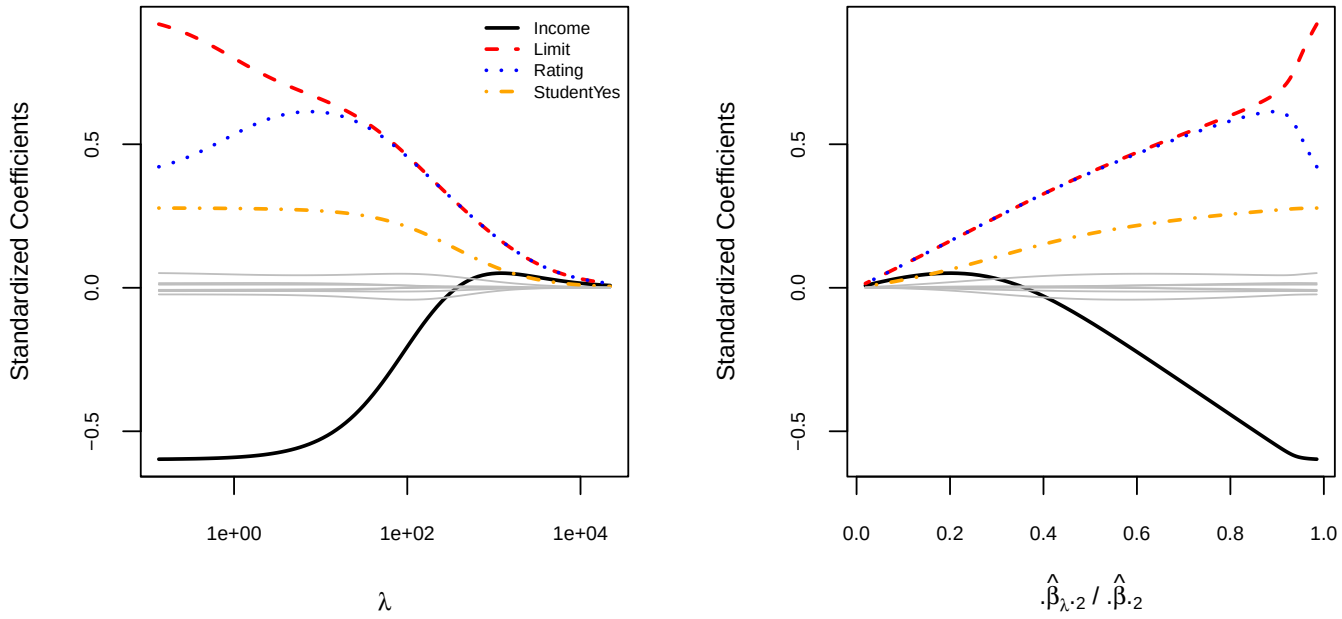
```r
line_widths[idx] <- lwds

### LEFT: coefficients vs log(lambda)
matplot(log(lambdas), t(betas),
        type = "l",
        col  = line_cols,
        lty  = line_types,
        lwd  = line_widths,
        xlab = expression(lambda),
        ylab = "Standardized Coefficients",
        ylim = range(betas),
        cex.lab = 0.8,
        xaxt = "n")

axis(1,
     at = log(c(1e-02, 1e+00, 1e+02, 1e+04)),
     labels = c("1e-02", "1e+00", "1e+02", "1e+04"))

legend("topright", hilite_names,
       col = cols, lty = lts, lwd = lwds, bty = "n", cex = 0.6)

### RIGHT: coefficients vs shrinkage ratio
matplot(ratio, t(betas),
        type = "l",
        col  = line_cols,
        lty  = line_types,
        lwd  = line_widths,
        xlab = expression(
  paste("‖", hat(beta)[lambda], "‖"[2], " / ",
        "‖", hat(beta),         "‖"[2])
),
        ylab = "Standardized Coefficients",
        cex.lab = 0.8)
```



2.5 Results

Norm Relationship From the formula above, we can derive the relationship between the norms

Vector Form: Stacking the coefficients into a vector:

$$\hat{\beta}_{\lambda}^{R} = \frac{1}{1+\lambda} * y$$

Squared l_2 Norm: Taking the squared l_2 norm of both sides:

$$||\hat{\beta}_{\lambda}^{R}||_2^2 = \frac{1}{(1+\lambda)^2} \sum_{j=1}^{p} y_j^2$$

$$||\hat{\beta}_{\lambda}^{R}||_2^2 = \frac{1}{(1+\lambda)^2} ||y||_2^2$$

Final l_2 Norm: Taking the square root uses the standard identity $||cv||_2 = |c| \cdot ||v||_2$:

$$||\hat{\beta}_{\lambda}^{R}||_2 = \frac{1}{1+\lambda} ||y||_2$$

This mathematical derivation confirms the statement: as λ increases, the denominator $(1+\lambda)$ increases, causing the norm $||\hat{\beta}_{\lambda}^{R}||_2$ to decrease.