# Midterm Exam 2

Matt Krasnow (the goat)

Due Friday, November 22, 2024 at 1:00pm

**NAME Matthew Krasnow HUID 81653380**

- This exam consists of 4 problems and 5 pages. The exam is worth 100 points (each question is worth 25 points) with the opportunity to earn 2 additional bonus points.
- Before submitting the exam, read and sign the statement below confirming that you have not cheated on this exam. A digital signature is fine.
- Some parts of a question may require the answer to an earlier part. If you cannot solve the earlier part, you can still receive partial credit for the later parts; make up a reasonable answer for the earlier part to use in subsequent parts of the problem.
- Show your work and explain your reasoning; the final answer is not as important as the process by which you arrived at that answer. We can more easily give partial credit if you have written out your steps clearly.
- This exam is open book and open notes, and you may use the internet. But there can be no discussing the exam with other people, including people who are not in this class.

*By signing below, I confirm that I have not cheated while taking this exam: I have not used any unauthorized resources nor copied another student's responses.*

Signature: Matthew Krasnow

| Problem Scoring | | |
|---|---|---|
| Problem | Point Value | Points Scored |
| 1 | 25 | |
| 2 | 25 | |
| 3 | 25 | |
| 4 | 25 | |
| Total | 100 | |

**Problem 1: Warm-up** [25 points]

Your collaborator is investigating the relationship between educational attainment and cognitive ability. She has collected data on a random sample of adults in Massachusetts in the dataset `p1.csv`. The variable `COG` encodes a continuous measure of cognitive ability designed to have a mean around 50 (higher scores are associated with higher cognitive ability). Educational attainment refers to the highest level of education an individual has completed, and is encoded in the `EDUC` variable as follows:

`0`: Primary School (grade 5)

`1`: Junior High School (grade 8)

`2`: High School (grade 12)

`3`: College degree or higher

Your collaborator asks you to fit a linear regression to predict cognitive ability from educational attainment.

**(a)** Write the formal statistical model you intend to fit.

I will use a linear regression model where $EDUC_i$ is treated as a categorical variable (ordinal) with four levels. I will define indicator variables to represent the education levels. \

$$

D1_i = 1  if  EDUC_i = 1, 0 otherwise (middle school) \

D2_i = 1  if  EDUC_i = 2, 0 otherwise (high schoole) \

D3_i = 1  if  EDUC_i = 2, 0 otherwise (college or higher ) \

$$

Thus the formal statistical model is:

$$COG_i = \beta_0 + \beta_1 * D1 + \beta_2 * D2 + \beta * D3 + \epsilon_i$$

Define

$\beta_0$ = mean cognative ability for people with primary school edu \

$\beta_1$ = mean cognative ability for people with middle school edu\

$\beta_2$ = mean cognative ability for people with high school edu \

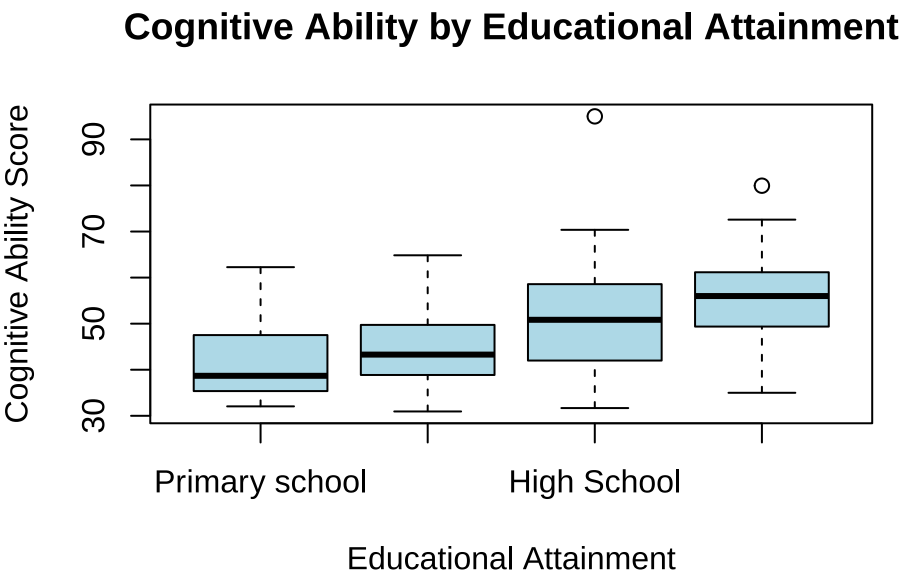
$\beta_3$ = mean cognative ability for people with college or higher education \

$\epsilon$ is the independent error terms. I would like to say that these are $\sim N(0, \sigma^2)$, but we have not properly evaluated our assumptions yet.

For this model to be valid, we need to evaluate our ELI H assumptions, but we have not done that yet

**(b)** Create an appropriate plot to visualize the association between educational attainment and cognitive ability.

```r
set.seed(139)

data <- read.csv("data/p1.csv")

# factor with labels
data$EDUC <- factor(data$EDUC, levels = 0:3, labels = c("Primary school", "Junior High School"
                                                        "High School", "College edu or higher"

# boxplot
boxplot(COG ~ EDUC, data = data, xlab = "Educational Attainment",
        ylab = "Cognitive Ability Score", main = "Cognitive Ability by Educational Attainment"
```



I chose a boxplot for this question because it demonstrates the spread of the data where the data is separated into categories effectively.

**(c)** Fit the linear regression that you specified in part (a) and interpret all the mean parameter estimates.

```r
model <- lm(COG ~ EDUC, data = data)

summary(model)
```

```
##
## Call:
## lm(formula = COG ~ EDUC, data = data)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -21.06  -5.82  -0.78   5.77  43.91
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 42.15       2.65   15.88  < 2e-16 ***
## EDUCJunior High School       2.10       2.91    0.72   0.4709    
## EDUCHigh School              8.94       2.92    3.07   0.0025 ** 
## EDUCCollege edu or higher   13.90       2.87    4.84  2.6e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.19 on 196 degrees of freedom
## Multiple R-squared:  0.244,  Adjusted R-squared:  0.232
## F-statistic: 21.1 on 3 and 196 DF,  p-value: 7.08e-12
```

Interpreation: $\beta_0 = 42.148$ on average, individuals with only primary school education have a COG of 42.148

$\beta_1 = 2.100$ On average, individuals with only junior high school education have a COG that is 2.1 points higher than those with only a primary school education (44.248)

$\beta_2 = 8.937$ On average, individuals with only high school education have a COG that is 8.937 points higher than those with only a primary school education (42.148+8.937)

$\beta_3 = 13.902$ On average, individuals with college education or higher have a COG that is 13.902 points higher than those with only a primary school education (42.148+13.902)

**(d)** Based on your plot, you might be tempted to use a linear term to model the association between educational attainment and cognitive ability instead, retaining the original coding scheme (that is, treating `EDUC` as a continuous variable). Statistically test whether this would be a better model. Comment on whether you think this is a better approach.

```r
# reload to make sure none of the other analysis affects this
data <- read.csv("data/p1.csv")

data$EDUC_num <- data$EDUC

data$EDUC <- factor(data$EDUC_num,
                    levels = 0:3,
                    labels = c("Primary School", "Junior High School",
                               "High School", "College edu or higher"))

# treat EDUC as continuous
model2 <- lm(COG ~ EDUC_num, data = data)

summary(model2)
```

```
##
## Call:
## lm(formula = COG ~ EDUC_num, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21.05  -6.44  -1.04   5.70  44.38
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    39.77       1.48   26.83  < 2e-16 ***
## EDUC_num        5.42       0.69    7.86  2.4e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.18 on 198 degrees of freedom
## Multiple R-squared:  0.238,  Adjusted R-squared:  0.234
## F-statistic: 61.8 on 1 and 198 DF,  p-value: 2.42e-13
```

```
#compare models using an F-test
anova(model2, model)  # model is categorical
```

```
## Analysis of Variance Table
##
## Model 1: COG ~ EDUC_num
## Model 2: COG ~ EDUC
##   Res.Df   RSS Df Sum of Sq    F Pr(>F)
## 1    198 16700
## 2    196 16565  2       134 0.79   0.45
```

The F test demonstrates that the data does not imply a statistically signficant difference between the categorical model and the continuous model. This indicates that the null hypothesis that the two models are the same should be retained (not accepted) until further data may show that one is better than the other.

Intuitively, I do not think it makes sense to treat education as a continuous scale. The scale does not accurately reflect the actual learning changes that occur between stages–I do not think that partial education is a linear scale – going to half of college or high school is not equally as rewarding as half of total college or high school. This should be treated as categorical.

**(e)** Suppose instead that you retained the original coding scheme (treating `EDUC` as a continuous variable) but fit a third degree polynomial (with all lower order terms included as well). Would this improve the fit of the model in part c)? What if you fit an even higher order polynomial?

```
model_poly3 <- lm(COG ~ EDUC_num + I(EDUC_num^2) + I(EDUC_num^3), data = data)

summary(model_poly3)
```

```
##
## Call:
## lm(formula = COG ~ EDUC_num + I(EDUC_num^2) + I(EDUC_num^3),
##     data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21.06  -5.82  -0.78   5.77  43.91
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      42.148      2.654   15.88   <2e-16 ***
## EDUC_num         -2.470      6.306   -0.39     0.70
## I(EDUC_num^2)     5.672      4.688    1.21     0.23
## I(EDUC_num^3)    -1.101      0.972   -1.13     0.26
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.19 on 196 degrees of freedom
## Multiple R-squared:  0.244,  Adjusted R-squared:  0.232
## F-statistic: 21.1 on 3 and 196 DF,  p-value: 7.08e-12
```

```
anova(model, model_poly3)
```

```
## Analysis of Variance Table
##
## Model 1: COG ~ EDUC
## Model 2: COG ~ EDUC_num + I(EDUC_num^2) + I(EDUC_num^3)
##   Res.Df   RSS Df Sum of Sq F Pr(>F)
## 1    196 16565
## 2    196 16565  0 -3.27e-11
```

Based on this, - the polynomial model does not improve uplon the categorical model from part c - the categorical model is better because it 1. makes more sense in the situation 2. there is no signficant evidence that the poly is better – we woul drather use a simpler model - same RSS and degrees of freedom, they explain the variance in COG equally well - adding new polynomials would not increase the fit because we have already observed polynomial terms which are not signficant–adding more will not help

**BONUS (2 points)**: Which of the assumptions of linear regression would you have to test for the model you fit in part c)? Test them and comment on the validity of that model. Is there anything else you should make your collaborator aware of?

We need to test our ELI H assumptions:

### Existence

- we standardly assume that the mean exists. This implies that there is finite variance of the residuals. Otherwise, our model would not converge.

### Linearity

Singe EDUC is categorical, the linearity assumption pertains to peroperly specifiying the model to include all categories. We have included all the categories in EDUC, so we are good (assumptino satisfied)

### Independence of Errors

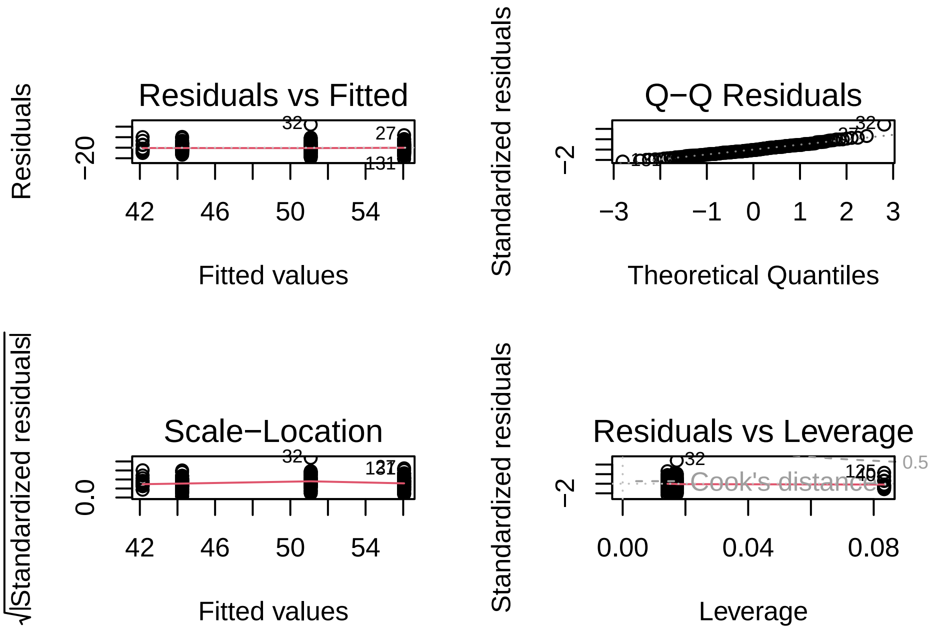
- this depends on the study design, so we don't really know based on this information
- to meet this assumption, we would need to see an SRS and no clustering or repeated measures

- if there is any reason to assume that the values of datapoints affects other datapoints, we cannot meet this assumption

### Homoskedasticity

- use the following test:

```
par(mfrow=c(2,2))
plot(model)
```



```
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 7.1, df = 3, p-value = 0.07
```

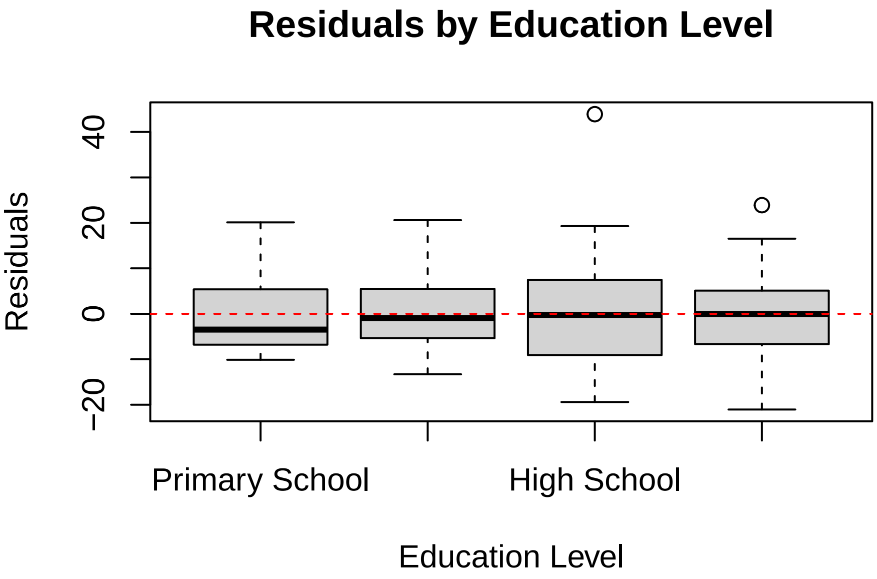
- at the = 0.05 value, this is NOT. a statistically significant finding. Thus we do not have fail to reject the null hypothesis
- we do not have data to state that the data is heteroskedastic

**Normality**

```r
residuals <- resid(model)
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.97, p-value = 0.001
```

```r
boxplot(residuals ~ data$EDUC,
        main="Residuals by Education Level",
        ylab="Residuals",
        xlab="Education Level")
# Add horizontal line at y=0
abline(h=0, col="red", lty=2)
```



- At the 0.05 level, we would reject the null hypothesis. This indicates that the residuals are NOT normally distributed.

Since our E.L. assumptions are met, if we assume linearity, we can use our linear model to estimate the mean parameters. however, we cannot do standard inference with this data because the residuals are not distributed normally.

I would want a collaborator to know that we did not formally check for independence and that this should be done before applying this data to a real world scenario.

**Problem 2: Open-ended application** [25 points]

You have another collaborator who has asked you to help him develop a model to estimate the associations between various predictors and systolic blood pressure (SBP) in a random sample of American adults aged at least 50. For this purpose, he has curated a dataset (p2.csv) which contains the following variables:

SBP: Systolic blood pressure, in mmHg.

AGE: Age in years.

GENDER: Gender indicator; 0 = Male, 1 = Female.

EDUC: Highest level of education attained. 0 = Primary School, 1 = Junior High, 2 = High School, 3 = College degree or higher.

DM: Diabetes mellitus indicator; 0 = No, 1 = Yes.

SMOKE: Smoking status indicator; 0 = No, 1 = Yes.

BMI: Body mass index (kg/m$^2$).

STATIN: Statin use indicator; 0 = No, 1 = Yes.

**(a)** Your collaborator does not have a model in mind, but he would like you to do some exploratory analyses, as he believes many of the variables he has collected are important. Used principled model selection procedure(s) to arrive at what you believe to be an appropriate model. For simplicity, do not consider any polynomial terms in this problem. Start by making sure the categorical variables are coded as factors.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3

## corrplot 0.95 loaded
```

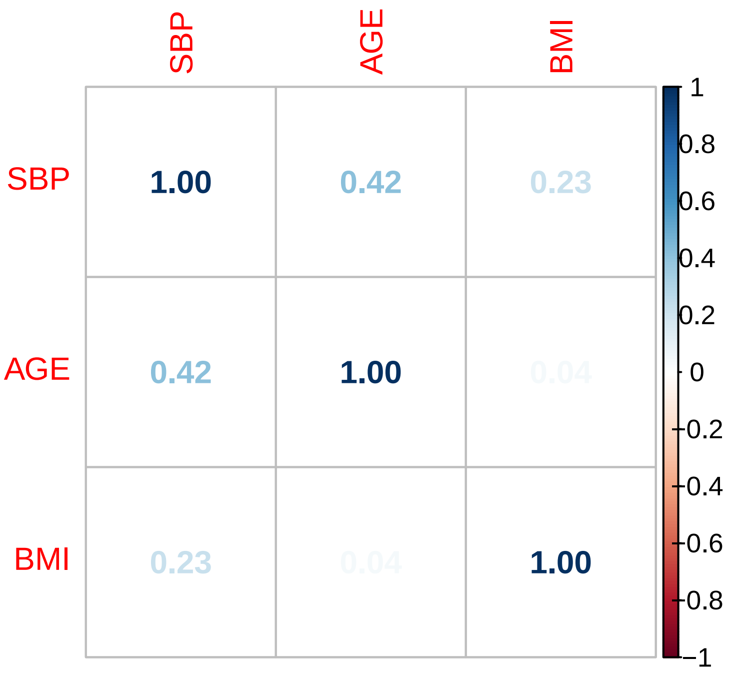
```
data <- read.csv("data/p2.csv")

# categorical variables => factors with appropriate labels
data$GENDER <- factor(data$GENDER, levels = c(0, 1), labels = c("Male", "Female"))
data$EDUC <- factor(data$EDUC, levels = c(0, 1, 2, 3),
                    labels = c("Primary School", "Junior High", "High School", "College or hig
data$DM <- factor(data$DM, levels = c(0, 1), labels = c("No", "Yes"))
data$SMOKE <- factor(data$SMOKE, levels = c(0, 1), labels = c("No", "Yes"))
data$STATIN <- factor(data$STATIN, levels = c(0, 1), labels = c("No", "Yes"))

str(data)
```

```
## 'data.frame':    314 obs. of  8 variables:
##  $ AGE   : int  55 65 68 70 53 64 70 75 66 64 ...
##  $ GENDER: Factor w/ 2 levels "Male","Female": 2 2 2 1 1 1 1 1 1 2 ...
##  $ EDUC  : Factor w/ 4 levels "Primary School",..: 3 2 3 3 1 2 4 2 1 2 ...
##  $ DM    : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ SMOKE : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
##  $ BMI   : num  39.7 29 22.3 32.4 29.2 ...
##  $ SBP   : num  108 124 116 166 159 ...
##  $ STATIN: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 2 1 ...
```

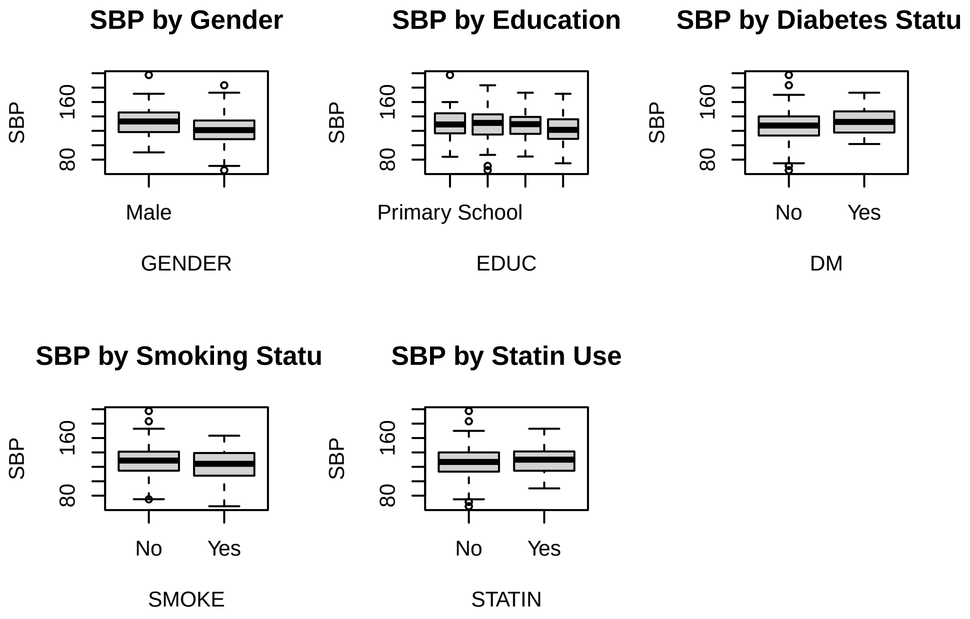
```
summary(data)
```

```
##       AGE           GENDER                    EDUC        DM        SMOKE
##  Min.   :50.0   Male  :175   Primary School   : 47   No :284   No :242
##  1st Qu.:54.0   Female:139   Junior High      :120   Yes: 30   Yes: 72
##  Median :61.0                High School      : 66
##  Mean   :61.8                College or higher: 81
##  3rd Qu.:69.0
##  Max.   :81.0
##       BMI             SBP         STATIN
##  Min.   :18.4   Min.   : 65.3   No :214
##  1st Qu.:24.4   1st Qu.:113.9   Yes:100
##  Median :26.6   Median :128.3
##  Mean   :27.4   Mean   :127.7
##  3rd Qu.:29.8   3rd Qu.:140.3
##  Max.   :61.0   Max.   :197.5
```

```
#EDA

# correlation matrix for continuous variables
continuous_vars <- data[c("SBP", "AGE", "BMI")]
cor_matrix <- cor(continuous_vars)
corrplot(cor_matrix, method = "number")
```



```
# boxplots for categorical variables
par(mfrow = c(2, 3))
boxplot(SBP ~ GENDER, data = data, main = "SBP by Gender")
boxplot(SBP ~ EDUC, data = data, main = "SBP by Education")
boxplot(SBP ~ DM, data = data, main = "SBP by Diabetes Status")
boxplot(SBP ~ SMOKE, data = data, main = "SBP by Smoking Status")
boxplot(SBP ~ STATIN, data = data, main = "SBP by Statin Use")

par(mfrow = c(1, 1))
```



```
# model selection
```

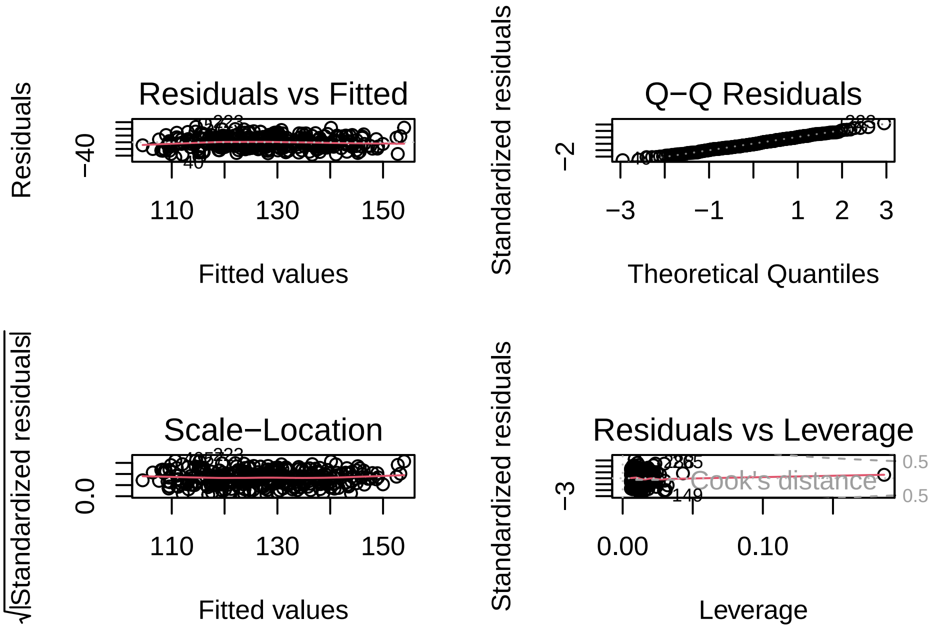
```r
# fit the full model
full_model <- lm(SBP ~ AGE + GENDER + EDUC + DM + SMOKE + BMI + STATIN, data = data)

# stepwise selection aic
step_model_aic <- step(full_model, direction = "both", trace = FALSE)

# stepwise selection bic
n <- nrow(data)
step_model_bic <- step(full_model, direction = "both", trace = FALSE,
                       k = log(n))  # Using BIC penalty

summary(step_model_aic)
```

```
##
## Call:
## lm(formula = SBP ~ AGE + GENDER + BMI, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45.38 -11.95  -1.35  12.79  56.64
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    48.403      9.700    4.99  1.0e-06 ***
## AGE             0.907      0.122    7.46  8.7e-13 ***
## GENDERFemale   -9.382      2.065   -4.54  8.0e-06 ***
## BMI             0.996      0.225    4.44  1.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.9 on 310 degrees of freedom
## Multiple R-squared:  0.267,  Adjusted R-squared:  0.26
## F-statistic: 37.7 on 3 and 310 DF,  p-value: <2e-16
```

```r
summary(step_model_bic)
```

```
##
## Call:
## lm(formula = SBP ~ AGE + GENDER + BMI, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45.38 -11.95  -1.35  12.79  56.64
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    48.403      9.700    4.99  1.0e-06 ***
## AGE             0.907      0.122    7.46  8.7e-13 ***
## GENDERFemale   -9.382      2.065   -4.54  8.0e-06 ***
## BMI             0.996      0.225    4.44  1.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.9 on 310 degrees of freedom
## Multiple R-squared:  0.267,  Adjusted R-squared:  0.26
## F-statistic: 37.7 on 3 and 310 DF,  p-value: <2e-16
```

```
# Check model assumptions for the final chosen model
par(mfrow = c(2, 2))
plot(step_model_aic)
```



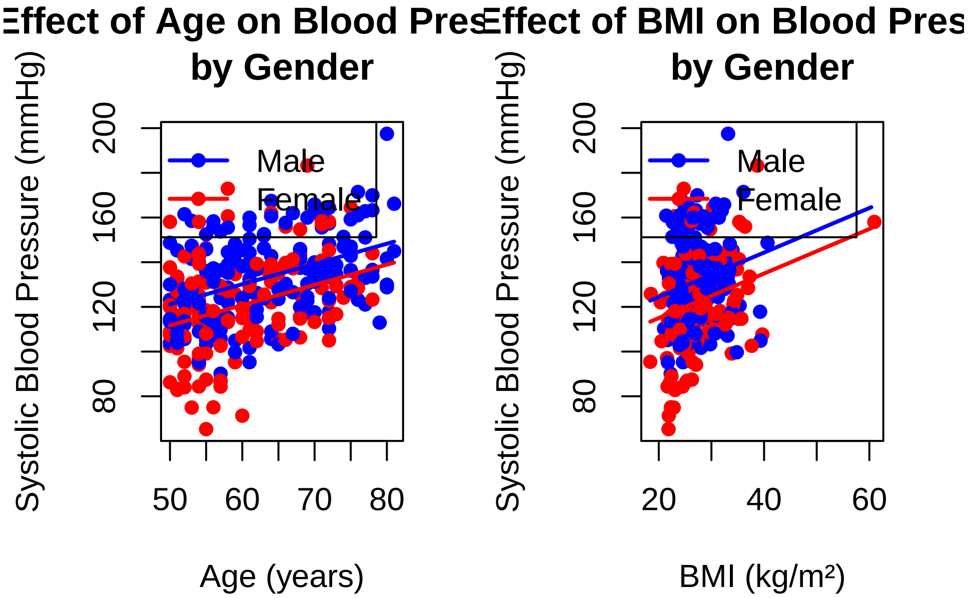
INTERPRETATION: - final model uses: age, gender, BMI to predict systolic BP. - R^2 => 26% of the variation in SBP is explained by this model - females have lower SBP - higher age => higher SBP - higher BMI => higher SBP - other variabeles are not statistically significant in this dataset

ASSUMPTIONS:

Existence: Assumed to be met so that the model can converge Linearity: The residuals vs fitted plot demonstratess a random scatter of points around the 0 line, indicating approximate linearity. Independenc: Since we do not hav eaccess to how to the data was collected, we cannot say for certain if the data points are independent or not. This is something to be cautious of. We would like to see an SRS and no dependencies. Since the problem states that it was a random sample of american adults, we shall say that this assumption was met. Homoskedasticity: We see a relatively constant spread of ressiduals across fitted values in the scale-location plot. This means that our homoskedasticity assumption is met. Normality of residuals: While we are not doing inference in this question, this is not necessary for this question. However, since the QQ plot closely follows the diagonal line, we can say that the residuals are approximately normally distributed.

**(b)** Explain your final model to your collaborator. Remember, your collaborator is not a statistician, and some of the terms in your model might be difficult to explain. It might be helpful to explain some of the relationships visually, and/or in particular subgroups. Every parameter should in some way be included in your interpretation.

```r
# For Age effect
age_seq <- seq(min(data$AGE), max(data$AGE), by = 1)
pred_age_male <- data.frame(
  AGE = age_seq,
  GENDER = factor("Male", levels = c("Male", "Female")),
  BMI = mean(data$BMI)
)
pred_age_female <- data.frame(
  AGE = age_seq,
  GENDER = factor("Female", levels = c("Male", "Female")),
  BMI = mean(data$BMI)
)

# For BMI effect
bmi_seq <- seq(min(data$BMI), max(data$BMI), by = 1)
pred_bmi_male <- data.frame(
  AGE = mean(data$AGE),
  GENDER = factor("Male", levels = c("Male", "Female")),
  BMI = bmi_seq
)
pred_bmi_female <- data.frame(
  AGE = mean(data$AGE),
  GENDER = factor("Female", levels = c("Male", "Female")),
  BMI = bmi_seq
)

# predictions
pred_age_male$SBP <- predict(step_model_aic, pred_age_male)
pred_age_female$SBP <- predict(step_model_aic, pred_age_female)
pred_bmi_male$SBP <- predict(step_model_aic, pred_bmi_male)
pred_bmi_female$SBP <- predict(step_model_aic, pred_bmi_female)

par(mfrow = c(1, 2))

# Age effect plot
plot(data$AGE, data$SBP, pch = 16, col = ifelse(data$GENDER == "Male", "blue", "red"),
     xlab = "Age (years)", ylab = "Systolic Blood Pressure (mmHg)",
     main = "Effect of Age on Blood Pressure\nby Gender")
lines(pred_age_male$AGE, pred_age_male$SBP, col = "blue", lwd = 2)
lines(pred_age_female$AGE, pred_age_female$SBP, col = "red", lwd = 2)
legend("topleft", c("Male", "Female"), col = c("blue", "red"),
       pch = 16, lty = 1, lwd = 2)

# BMI effect plot
plot(data$BMI, data$SBP, pch = 16, col = ifelse(data$GENDER == "Male", "blue", "red"),
     xlab = "BMI (kg/m²)", ylab = "Systolic Blood Pressure (mmHg)",
     main = "Effect of BMI on Blood Pressure\nby Gender")
lines(pred_bmi_male$BMI, pred_bmi_male$SBP, col = "blue", lwd = 2)
lines(pred_bmi_female$BMI, pred_bmi_female$SBP, col = "red", lwd = 2)
legend("topleft", c("Male", "Female"), col = c("blue", "red"),
       pch = 16, lty = 1, lwd = 2)
```



Explanation for collaborator:

There are three factors in our dataset that are related to SBP: sex, age, and BMI. We find that: - for every year increase in age, we expect BP to increase by about 1 mmHg if we hold other variables constant - women typically have lower SBP than men (9 mmHg lower) - for every 1 point increase in BMI BP increases by about 1 mmHG with other variables held constant - the other variables like education level, diabetes status, smoking, and statin use didn't show strong enough relationships with BP to be included in our final model

We used a linear model here. Linear models are particularly useful. Our model explains about 26% of the variation in the data. We know that our model is not complete – it doesn't have all the factors that may be affecting SBP. But these three variables explain about a quarter of the variation of SBP in the dataset.

Based on our findings, we should more closely monitor "at risk" individuals, specifically people who fall into the higher predicted SBP groups.

**Problem 3: The mechanics of prediction** [25 points]

A year later, your collaborator returns to you, but he is no longer interested in estimating the associations between predictor variables and `SBP`. Instead, he is hoping you can help him to build a best predictive model. He has expanded his target population to include all adults, and has collected several more variables. The data (`p3.csv`) now consists of the following variables:

`SBP`: Systolic blood pressure, in mmHg.

`AGE`: Age in years.

`GENDER`: Gender indicator; 0 = Male, 1 = Female.

`EDUC`: Highest level of education attained. 0 = Primary School, 1 = Junior High, 2 = High School, 3 = College degree or higher.

`DM`: Diabetes mellitus indicator; 0 = No, 1 = Yes.

`SMOKE`: Smoking status indicator; 0 = No, 1 = Yes.

`BMI`: Boprobdy mass index (kg/m$^2$).

`STATIN`: Statin use indicator; 0 = No, 1 = Yes.

`CVD`: History of cardiovascular event indicator; 0 = No, 1 = Yes.

`HYPERTENSION`: Indicator of hypertension; 0 = No, 1 = Yes.

`CHOL`: Total cholesterol, in mmol/L.

`HDL`: HDL cholesterol, in mmol/L.

`eGFR`: Estimated glomerular filtration rate, a measure of kidney function in mL/min (lower values indicate possible kidney damage).

**(a)** Start by making sure your categorical variables are coded as factors.

```r
data <- read.csv("data/p3.csv")

# paranoia
str(data)
```

```
## 'data.frame':    4065 obs. of  13 variables:
##  $ AGE         : int  35 35 35 35 35 35 35 35 35 35 ...
##  $ GENDER      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CVD         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ DM          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HYPERTENSION: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ BMI         : num  24.6 21.2 29.2 29.2 29.6 ...
##  $ SBP         : num  116 118 132 130 118 ...
##  $ CHOL        : num  5.5 3.65 6.93 3.95 4.58 5.64 5.35 6.16 4.68 5.09 ...
##  $ HDL         : num  0.94 0.87 1.14 0.98 0.92 1.1 0.9 1.19 0.87 1.31 ...
##  $ STATIN      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ eGFR        : num  68.2 104.6 98.5 113.1 90.6 ...
##  $ EDUC        : int  3 3 2 2 3 1 2 0 2 2 ...
##  $ SMOKE       : int  1 0 1 0 0 0 1 0 0 1 ...
```

```r
summary(data)
```

```
##       AGE          GENDER            CVD              DM        
##  Min.   :35.0   Min.   :0.000   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:45.0   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0000  
##  Median :54.0   Median :0.000   Median :0.000   Median :0.0000  
##  Mean   :54.6   Mean   :0.476   Mean   :0.075   Mean   :0.0615  
##  3rd Qu.:63.0   3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.:0.0000  
##  Max.   :82.0   Max.   :1.000   Max.   :1.000   Max.   :1.0000  
##   HYPERTENSION       BMI             SBP              CHOL            HDL      
##  Min.   :0.000   Min.   :16.8   Min.   : 77.5   Min.   : 1.58   Min.   :0.48  
##  1st Qu.:0.000   1st Qu.:23.7   1st Qu.:112.5   1st Qu.: 4.64   1st Qu.:1.14  
##  Median :0.000   Median :26.1   Median :123.0   Median : 5.31   Median :1.37  
##  Mean   :0.339   Mean   :26.8   Mean   :125.9   Mean   : 5.36   Mean   :1.41  
##  3rd Qu.:1.000   3rd Qu.:29.2   3rd Qu.:136.5   3rd Qu.: 6.05   3rd Qu.:1.63  
##  Max.   :1.000   Max.   :61.0   Max.   :218.5   Max.   :10.02   Max.   :3.44  
##      STATIN           eGFR             EDUC          SMOKE      
##  Min.   :0.000   Min.   : 20.9   Min.   :0.00   Min.   :0.000  
##  1st Qu.:0.000   1st Qu.: 69.6   1st Qu.:1.00   1st Qu.:0.000  
##  Median :0.000   Median : 77.9   Median :2.00   Median :0.000  
##  Mean   :0.221   Mean   : 78.6   Mean   :1.85   Mean   :0.239  
##  3rd Qu.:0.000   3rd Qu.: 87.7   3rd Qu.:3.00   3rd Qu.:0.000  
##  Max.   :1.000   Max.   :149.5   Max.   :3.00   Max.   :1.000  
```

```r
# binary variables to factors
binary_vars <- c("GENDER", "DM", "SMOKE", "STATIN", "CVD", "HYPERTENSION")
data[binary_vars] <- lapply(data[binary_vars], factor)

# Convert EDUC to an ordered factor with appropriate levels
data$EDUC <- factor(data$EDUC,
                    levels = 0:3,
                    labels = c("Primary School", "Junior High",
                             "High School", "College degree or higher"),
                    ordered = TRUE)

str(data)
```

```
## 'data.frame':    4065 obs. of  13 variables:
##  $ AGE         : int  35 35 35 35 35 35 35 35 35 35 ...
##  $ GENDER      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CVD         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DM          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HYPERTENSION: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ BMI         : num  24.6 21.2 29.2 29.2 29.6 ...
##  $ SBP         : num  116 118 132 130 118 ...
##  $ CHOL        : num  5.5 3.65 6.93 3.95 4.58 5.64 5.35 6.16 4.68 5.09 ...
##  $ HDL         : num  0.94 0.87 1.14 0.98 0.92 1.1 0.9 1.19 0.87 1.31 ...
##  $ STATIN      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ eGFR        : num  68.2 104.6 98.5 113.1 90.6 ...
##  $ EDUC        : Ord.factor w/ 4 levels "Primary School"<..: 4 4 3 3 4 2 3 1 3 3 ...
##  $ SMOKE       : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 1 1 2 ...
```

```r
#lets go
```

**(b)** Break the dataset into an 80-20 train-test split.

```r
set.seed(139)


train_size <- floor(0.8 * nrow(data))


train_indices <- sample(seq_len(nrow(data)), size = train_size)


train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]


cat("Training size:", nrow(train_data), "observations\n")
```

```
## Training size: 3252 observations
```

```r
cat("Test size:", nrow(test_data), "observations\n")
```

```
## Test size: 813 observations
```

```r
cat("Split proportion (train):", nrow(train_data)/nrow(data), "\n")
```

```
## Split proportion (train): 0.8
```

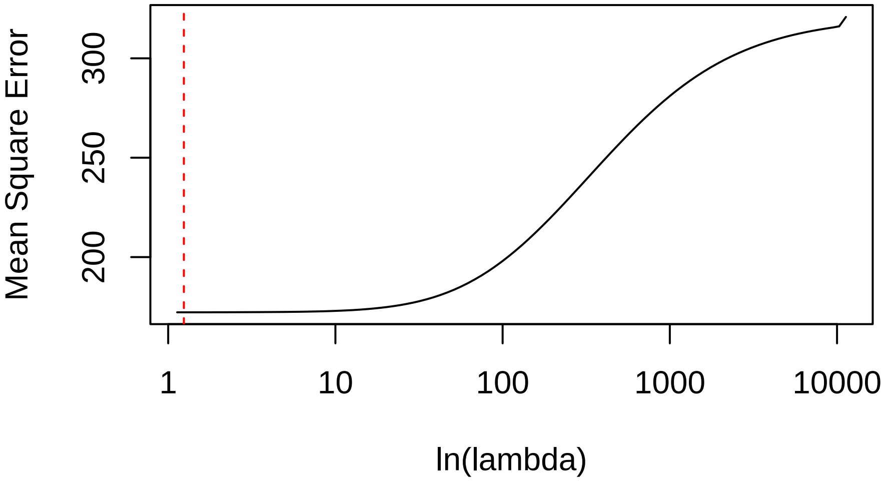
**(c)** Using your training data and a $k$ of 5, perform $k$-fold cross validation with ridge regression to identify the value of $\lambda$ that produces the lowest mean RMSE for a full model that includes all main effects and 2-way interaction terms. Plot the MSE vs. $\ln(\lambda)$. Report the optimal $\lambda$ and the RMSE on your test data.

```r
# Load required library
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
set.seed(139)

predictors <- names(train_data)[names(train_data) != "SBP"]
formula <- as.formula(paste("SBP ~",
                          paste(paste0("(",
                                paste(predictors, collapse = "+"),
                                ")^2"))))

# model matrices for both training and test data
x_train <- model.matrix(formula, data = train_data)[,-1]  # Remove intercept
y_train <- train_data$SBP

x_test <- model.matrix(formula, data = test_data)[,-1]
y_test <- test_data$SBP

#  k-fold
cv_ridge <- cv.glmnet(x_train, y_train,
                      alpha = 0, # ridge!!
                      nfolds = 5,
                      standardize = TRUE)
plot(cv_ridge$lambda, cv_ridge$cvm,
     type = "l",
     xlab = "ln(lambda)",
     ylab = "Mean Square Error",
     log = "x")
abline(v = cv_ridge$lambda.min, col = "red", lty = "dashed")
```



```
optimal_lambda <- cv_ridge$lambda.min

# Fit final model with optimal lambda
final_model <- glmnet(x_train, y_train,
                      alpha = 0,
                      lambda = optimal_lambda,
                      standardize = TRUE)

# Make predictions
predictions <- predict(final_model, newx = x_test)

# RMSE
test_rmse <- sqrt(mean((y_test - predictions)^2))
```

```
cat("Optimal lambda:", optimal_lambda, "\n")
```

```
## Optimal lambda: 1.241
```

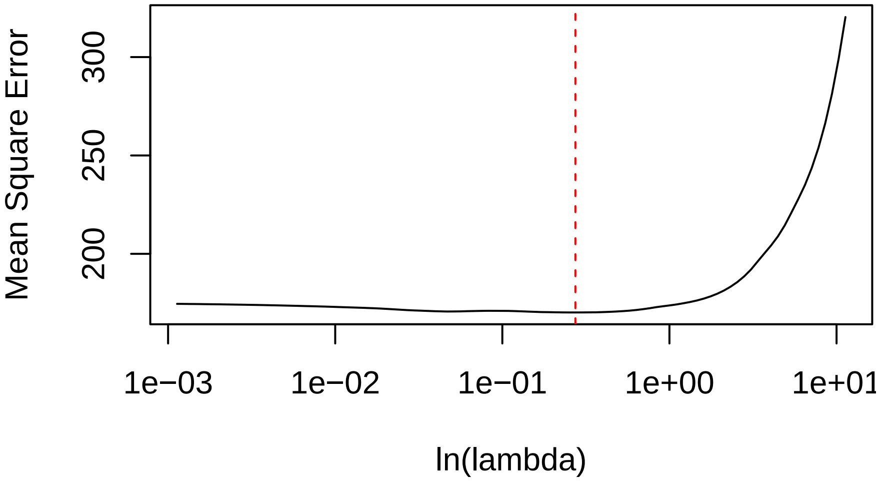
```
cat("Test set RMSE:", test_rmse, "\n")
```

```
## Test set RMSE: 13.71
```

The optimal lambda is 1.241 The test set RMSE is 13.71

**(d)** Repeat part c) using LASSO.

```
set.seed(139)
cv_lasso <- cv.glmnet(x_train, y_train,
                      alpha = 1, # LASSO regression
                      nfolds = 5,
                      standardize = TRUE)

plot(cv_lasso$lambda, cv_lasso$cvm,
     type = "l",
     xlab = "ln(lambda)",
     ylab = "Mean Square Error",
     log = "x")
abline(v = cv_lasso$lambda.min, col = "red", lty = "dashed")
```



```
optimal_lambda_lasso <- cv_lasso$lambda.min

final_model_lasso <- glmnet(x_train, y_train,
                            alpha = 1,
                            lambda = optimal_lambda_lasso,
                            standardize = TRUE)

predictions_lasso <- predict(final_model_lasso, newx = x_test)

test_rmse_lasso <- sqrt(mean((y_test - predictions_lasso)^2))
```

```r
# Print results
cat("Optimal lambda:", optimal_lambda_lasso, "\n")
```

```
## Optimal lambda: 0.2736
```

```r
cat("Test set RMSE:", test_rmse_lasso, "\n")
```

```
## Test set RMSE: 13.82
```

Optimal lambda: 0.2736 Test set RMSE: 13.82

**Problem 4: A little theory and simulation** [25 points]

In this problem, you will explore the concept of shrinkage (or regularization) through an example called "Stein's paradox" (parts a-c) and then draw a connection with ridge regression in part d.

Suppose $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ for $i = 1, \dots, n$. You observe $Y_1, \dots Y_n$ and want to estimate $\mu_1, \dots, \mu_n$.

**(a)** The most obvious estimator of $\mu_i$ is $\hat{\mu}_i = Y_i$. What is the bias of this estimator?

Bias calculation:

$$Bias(\hat{\mu}_i) = E[\hat{\mu}_i] - \mu_i = \mu_i - \mu_i = 0$$

This is an unbiased estimator

**(b)** A *shrinkage* estimator of $\mu_i$ is $\hat{\mu}_i(c) = cY_i$ where $c \in [0, 1]$ is a user-specified parameter that controls the amount of shrinkage. Write R code to do the following:

- Set $n = 1000$. Randomly generate $\mu_i \sim \text{Uniform}(0, 10)$ for $i = 1, \dots, n$.
- Randomly generate $Y_i \sim \mathcal{N}(\mu_i, \sigma^2)$ for $i = 1, \dots, n$ where $\sigma = 4$.
- For each $c \in \{0.0, 0.01, 0.02, \dots, 0.99, 1.0\}$ compute MSE(c) where:

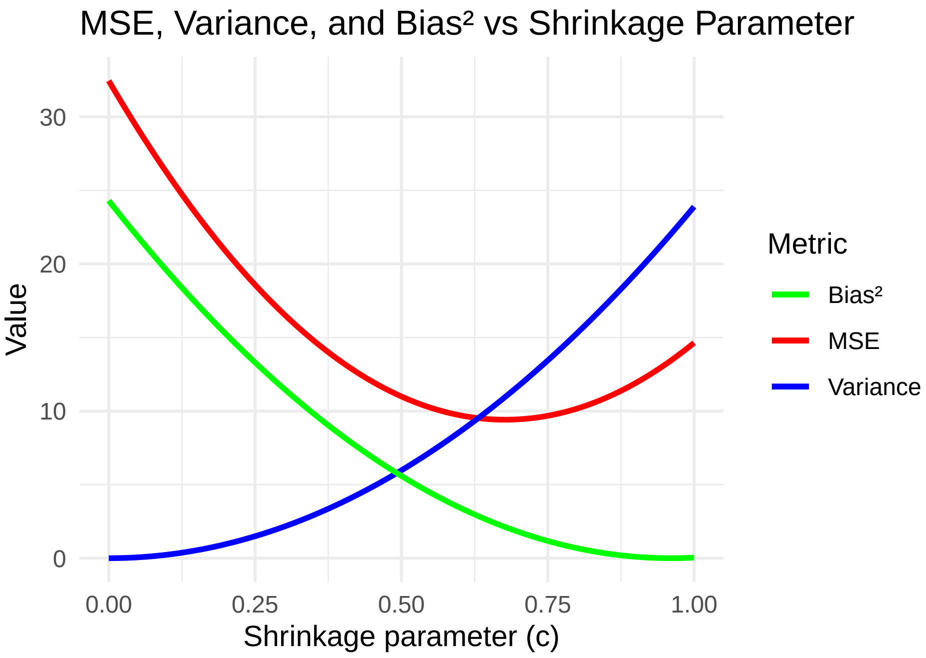
$$\text{MSE(c)} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}_i(c) - \mu_i)^2$$

Plot MSE(c) versus $c$, along with the variance and the square of the bias. For your dataset, what value of $c$ minimizes MSE(c)? So, does shrinkage help?

```r
set.seed(139)

n <- 1000
sigma <- 4
c_values <- seq(0, 1, by=0.01)


mu <- runif(n, min=0, max=10)


Y <- rnorm(n, mean=mu, sd=sigma)


results <- sapply(c_values, function(c) {
    # Shrinkage estimates
    mu_hat <- c * Y

    # Calculate MSE
    mse <- mean((mu_hat - mu)^2)
    #  Bias
    bias <- mean(mu_hat - mu)

    #  Variance
    var <- mean((mu_hat - mean(mu_hat))^2)

    return(c(mse=mse, bias=bias, var=var))
})


results_df <- data.frame(
    c = c_values,
    MSE = results[1,],
    Bias = results[2,],
    Variance = results[3,]
)




p <- ggplot(results_df) +
    geom_line(aes(x=c, y=MSE, color="MSE"), size=1) +
    geom_line(aes(x=c, y=Variance, color="Variance"), size=1) +
    geom_line(aes(x=c, y=Bias^2, color="Bias²"), size=1) +
    labs(x="Shrinkage parameter (c)",
         y="Value",
         title="MSE, Variance, and Bias² vs Shrinkage Parameter",
         color="Metric") +
    theme_minimal() +
    scale_color_manual(values=c("MSE"="red", "Variance"="blue", "Bias²"="green"))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
print(p)
```



```
opt_c <- c_values[which.min(results_df$MSE)]
cat("Optimal c value:", opt_c, "\n")
```

```
## Optimal c value: 0.68
```

```
cat("Minimum MSE:", min(results_df$MSE), "\n")
```

```
## Minimum MSE: 9.413
```

Optimal c value: 0.68 Minimum MSE: 9.413

YES shrinkagedoes help because the c value is not 1. If the optimal shrinkage was 1 or close to 1, it would not help becuase it wouldn't change hte $Y_i$ much, but it is .68

**(c)** Explain your findings from (b) in terms of the bias-variance tradeoff. What happens to the bias and the variance when $c \to 0$?

bias vairance tradeoff - variance increases quadratically as C increases - bias^2 decreases as C increase

SO when C goes to 0, - variance approaches 0 - bias approaches the maximum of around 24 - this happens because as we are shrinking hteestimates, this means that all of the estimates get closer together (lower variance) and farther away from the true means (bias increase)

This is the whole point of shrinkage – we want to optimize bias variance tradeoff

**(d)** Suppose we defined our ridge regression estimates as:

$$\hat{\beta}^R_\lambda = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\vec{\mathbf{y}}$$

where $\vec{\mathbf{y}} \in \mathbb{R}^{n\times 1}$. Note this is a slightly different definition than was used in class. But suppose you want to frame the shrinkage problem in parts a) - c) as a ridge regression problem. How should you define $\mathbf{X}$ and $\mathbf{I}$? Make sure to include dimensions.

Model definition $\beta_i$: $Y_i = \beta_i + \varepsilon_i$

In matrix notation this is: $\vec{\mathbf{y}} = \mathbf{X}\beta + \varepsilon$

To match the shrinkage estimator, define: $\mathbf{X}$: An $n \times n$ identity matrix. This means each $Y_i$ is directly associated with its own $\beta_i$.

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

$\mathbf{X}$ is the identity matrix. Since $\mathbf{X}$ is the identity matrix:

$$\mathbf{X}^\top \mathbf{X} = \mathbf{I}$$

Thus our estimator simplifies to be:

$$\hat{\beta}_\lambda^R = (\mathbf{I} + \lambda \mathbf{I})^{-1} \vec{\mathbf{y}} = \frac{1}{1+\lambda} \vec{\mathbf{y}}$$

Thus when we set $c = \frac{1}{1+\lambda}$, this matches our shrinkage estimator $\hat{\mu}(c) = cY_i$

Dimensions:

- $\mathbf{X}$ is an $n \times n$ identity matrix
- $\mathbf{I}$ is also the $n \times n$ identity matrix

When we make $\mathbf{X}$ an identity matrix, the shrinkage estimator becomes a ridge regression estimator. This shows that the shrinkage in this context is the same as applying RIDGE with a $\lambda$.

**(e)** Using the design matrix you defined in part d), show that $\hat{\beta}_\lambda^R = c\vec{\mathbf{y}}$ for some $c \in \mathbb{R}$. Give the actual mathematical expression for $c$ as a function of $\lambda$.

The ridge regression estimator is:

$$\hat{\beta}_\lambda^R = \left(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^\top \vec{\mathbf{y}}$$

Since $\mathbf{X}$ is the identity matrix, we have:

$$\mathbf{X}^\top \mathbf{X} = \mathbf{I}^\top \mathbf{I} = \mathbf{I}\mathbf{I} = \mathbf{I}$$

Therefore, the term $\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}$ simplifies to:

$$\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I} = \mathbf{I} + \lambda \mathbf{I} = (1+\lambda)\mathbf{I}$$

The inverse of this term is:

$$\left(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}\right)^{-1} = \left((1+\lambda)\mathbf{I}\right)^{-1} = \frac{1}{1+\lambda}\mathbf{I}$$

Next, since $\mathbf{X}^\top = \mathbf{I}^\top = \mathbf{I}$, we have:

$$\mathbf{X}^\top \vec{\mathbf{y}} = \mathbf{I}\vec{\mathbf{y}} = \vec{\mathbf{y}}$$

Substituting back into the ridge regression estimator:

$$\begin{aligned}\hat{\beta}_\lambda^R &= \left(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}\right)^{-1} \mathbf{X}^\top \vec{\mathbf{y}} \\ &= \left(\frac{1}{1+\lambda}\mathbf{I}\right)\vec{\mathbf{y}} \\ &= \frac{1}{1+\lambda}\vec{\mathbf{y}}\end{aligned}$$

Thus, we have shown that:

$$\hat{\beta}_\lambda^R = c\vec{\mathbf{y}}$$

Using the identity matrix for both $\mathbf{X}$ and $\mathbf{I}$, the ridge regression estimator simplifies to scaling the vector $\mathbf{y}$ by a constant factor $c$ which depends on $\lambda$.