Lab 05: Regression

PSTAT 100: Spring 2024 (Instructor: Ethan P. Marzban)

MEMBER 1 (NetID 1)   MEMBER 2 (NetID 2)
MEMBER 3 (NetID 3)

May 12, 2024

Required Packages

```
library(ottr)        # for checking test cases (i.e. autograding)
library(pander)      # for nicer-looking formatting of dataframe outputs
library(tidyverse)   # for graphs, data wrangling, etc.
library(gridExtra)   # for multipanel graphs
```

Logistical Details

Logistical Details

- This lab is due by **11:59pm on Wednesday, May 15, 2024**.
- Collaboration is allowed, and encouraged!
  - If you work in groups, list ALL of your group members' names and NetIDs (not Perm Numbers) in the appropriate spaces in the YAML header above.
  - Please delete any "MEMBER X" lines in the YAML header that are not needed.
  - No more than 3 people in a group, please.
- Ensure your Lab properly renders to a `.pdf`; non-`.pdf` submissions will not be graded and will receive a score of 0.
- Ensure all test cases pass (test cases that have passed will display a message stating `"All tests passed!"`)

Lab Overview and Objectives

In this lab, we will discuss:

- Regression using a categorical predictors
- Multiple regression
- Regression diagnostics

Multiple Regression and Modeling

Given data $\mathcal{D} := \{\vec{x}_i, y_i\}_{i=1}^{n}$ consisting of observations y_i of a **response variable** `y` and observations $\vec{x} := (x_{i1}, \cdots, x_{ip})$ on p **explanatory** or **predictor variables** $\texttt{x}_1$ through $\texttt{x}_\texttt{p}$, a **statistical model** assumes the relationship between y_i and $\vec{x}_i$ to be

$$y_i = f(\vec{x}) + \varepsilon_i$$

for some **noise** term ε_i.

A **linear regression** model assumes:

1) A linear signal function; i.e. $f(\vec{x}_i) = \beta_0 + \sum_{j=1}^{p} x_{ij}$
2) Numerical response values (i.e. `y` is assumed to be a numerical variable as opposed to a categorical one).

The matrix representation of a linear regression model is:

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{:=\vec{y}} = \underbrace{\begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}}_{:=\mathbf{X}} \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}}_{:=\vec{\beta}} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}}_{:=\vec{\varepsilon}}$$

It is typical to assume i.i.d. Gaussian errors:

$$\varepsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2) \iff \vec{\varepsilon} \sim \mathcal{N}_n\left(\vec{0}\ ,\ \sigma^2\mathbf{I}\right)$$

The **ordinary least squares** (OLS) fit to the data seeks to find estimators $\widehat{\beta}$ that solve the following minimization problem:

$$\widehat{\beta} = \arg\ \min_{\vec{b}} \left\{ \left\| \vec{y} - \mathbf{X}\vec{b} \right\|^2 \right\}$$

which, under certain conditions, admits the following solution:

$$\widehat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \vec{y}$$

Let's start out by comparing what the `lm()` function does against what we might do "by hand", according to the theory above.

Consider the following toy dataset, consisting of observations on one response variable `y` and two explanatory variables `x1` and `x2`.

```r
y <- c(1, 1, 2, 3, 3, 4, 5, 6, 7, 8)
x1 <- c(1, 2, 1, 4, 5, 7, 6, 7, 9, 10)
x2 <- c(1, 1, 1, 2, 3, 3, 4, 3, 1, 2)
```

! Question 1

Run the above code chunk to create variables `y`, `x1`, and `x2` with the appropriate values. Then, construct the data matrix, and store this in a variable called `X`. **Important:** Recall that we (unless otherwise specified) *always* include an intercept in our model. What does this mean about the first column of `X`?

Solution:

```r
## replace this line with your code
X <- matrix(c(rep(1, length(x1)), x1, x2),
            byrow = F,
            ncol = 3)
```

Answer Check:

```r
# DO NOT EDIT THIS LINE
invisible({check("tests/q1.R")})
```

```
All tests passed!
```

! Question 2

Compute the OLS estimates using **ONLY** the following quantities/functions/operators: `solve()`, `t()`, `%*%`, and `X` [where `X` is the variable you created in Question 1 above]. Store your result in a vector called `ols_hand`.

Solution:

```r
## replace this line with your code
ols_hand <- solve(t(X) %*% X) %*% t(X) %*% y
```

Answer Check:

```r
# DO NOT EDIT THIS LINE
invisible({check("tests/q2.R")})
```

```
All tests passed!
```

Question 3

Now, obtain the OLS estimates using the `lm()` function. Display both these estimates and your `ols_hand` values, and compare.

Solution:

```
## replace this line with your code
lm(y ~ x1 + x2) %>% coef()
```

```
(Intercept)          x1          x2 
  0.4018692   0.7523364  -0.1495327 
```

```
ols_hand
```

```
           [,1]
[1,]  0.4018692
[2,]  0.7523364
[3,] -0.1495327
```

Answer Check:
There is no autograder for this question.

Interpretation is key. In a SLR setting, the interpretation of the $\hat{\beta}_1$ coefficient is relatively straightforward: a one-unit increase in the explanatory variable corresponds to a predicted $\hat{\beta}_1$-unit increase in the response variable. In a multiple linear regression (MLR) setting, note that

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip} + \varepsilon_i$$

Hence, we can interpret the value of $\hat{\beta}_i$, for any i, as: a one-unit change in the i^{th} predictor corresponds to a predicted $\hat{\beta}_i$-unit change in the response, *ceterus paribus* (holding all else constant). Keep this in mind for the next part of this lab.

Regression Diagnostics

Recall that simply fitting and interpreting a regression model is not enough - we must perform some **model diagnostics** as well. As with many aspects of data science and statistical modeling, there isn't a single procedural approach to performing model diagnostics - rather, with practice you learn which tools to use in which situation.

For the purposes of this class, there are two main tools we can use for diagnostics:

1) QQ-plots (to assess the normality assumption of the errors)
2) Residuals plots (to check for poorly fitting models, outliers, and heteroskedasticity).

In this portion of the lab, we'll deal with a mock dataset consisting of three variables:

- `scores`: the midterm scores of students in a particular PSTAT course (maximum number of points was 35)
- `slp_hrs`: the amount of sleep (in hours) students got the night before the exam
- `stdy_hrs`: the amount of time (in hours) students studied for the exam

Our goal is to regress `scores` on `slp_hrs` and `stdy_hrs` (i.e. we'll treat `scores` as the response variable).

! Question 4

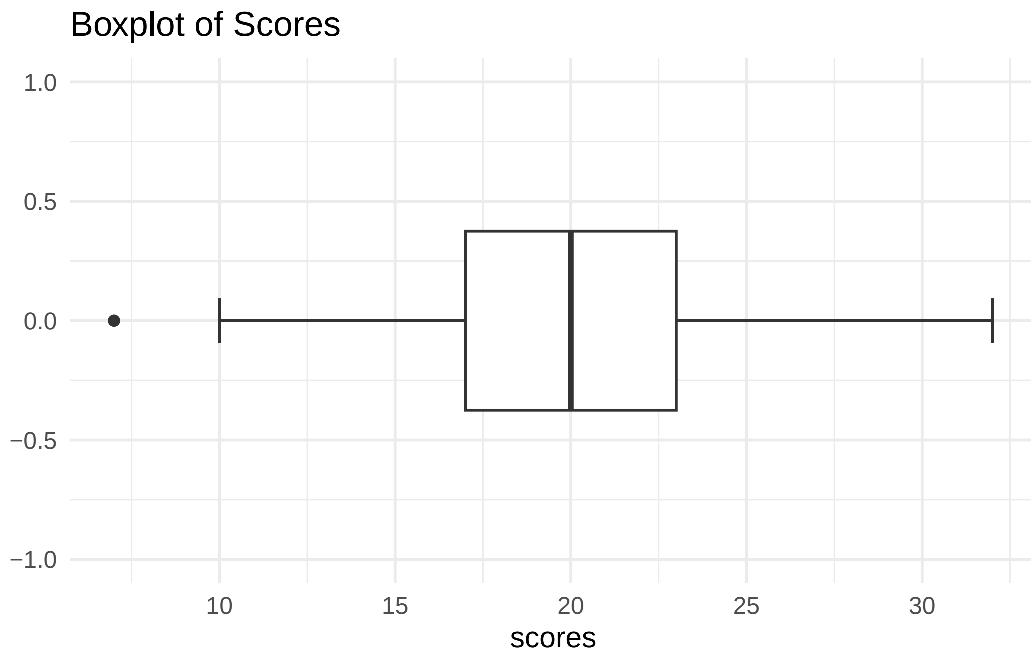
Import the file called `scores.csv`, located in the `data` subfolder. As a quick proxy for EDA, generate the following:

- a plot to visualize the distribution of scores on the exam
- a plot to visualize the distribution of the amount of sleep students got the night before the exam
- a plot to visualize the distribution of the amount of time students spent studying for the exam

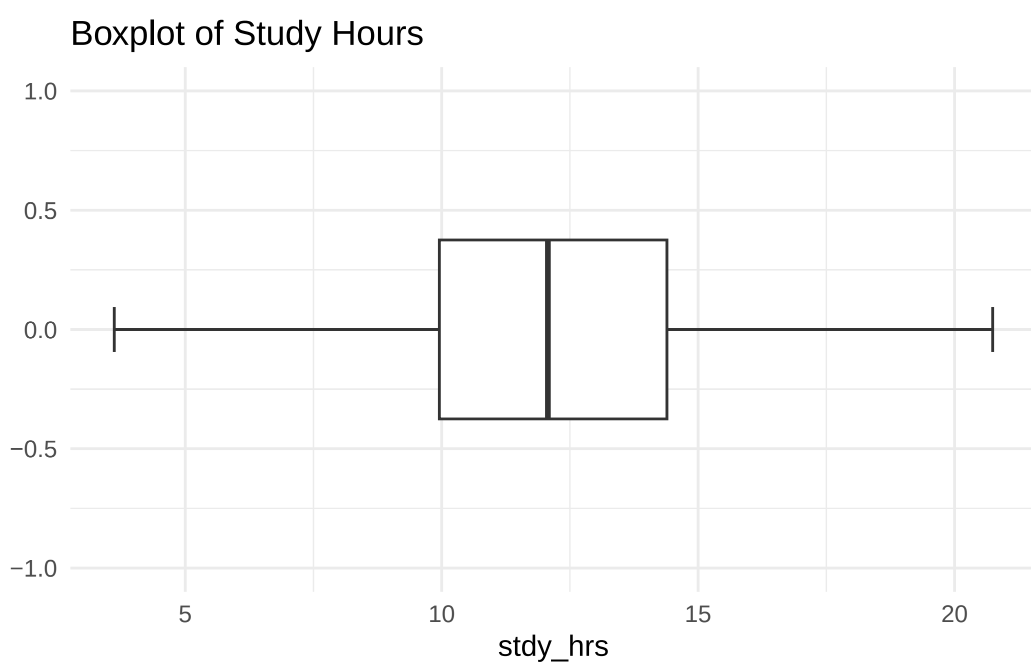
It's up to you to figure out which plot is best-suited for each of these; if there are potentially multiple plots that could be produced, just pick one.

Solution:

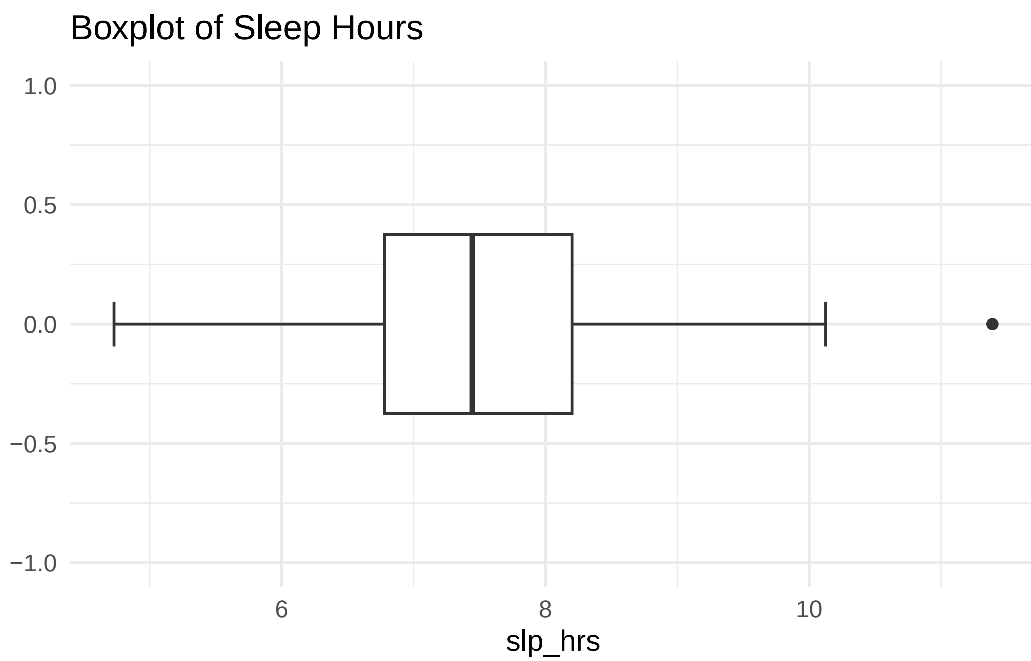
```r
## replace this line with your code
scores_data <- read.csv("data/scores.csv")

scores_data %>%
  ggplot(aes(x = scores)) +
  geom_boxplot(staplewidth = 0.25) +
  theme_minimal() +
  ggtitle("Boxplot of Scores")  +
  ylim(c(-1, 1)) # not strictly necessary; just for aesthetic purposes
```




```r
scores_data %>%
  ggplot(aes(x = stdy_hrs)) +
  geom_boxplot(staplewidth = 0.25) +
  theme_minimal() +
  ggtitle("Boxplot of Study Hours") +
  ylim(c(-1, 1)) # not strictly necessary; just for aesthetic purposes
```



```r
scores_data %>%
  ggplot(aes(x = slp_hrs)) +
  geom_boxplot(staplewidth = 0.25) +
  theme_minimal() +
  ggtitle("Boxplot of Sleep Hours") +
  ylim(c(-1, 1))  # not strictly necessary; just for aesthetic purposes
```



Answer Check:
There is no autograder for this question.

! Question 5

Use `lm()` to regress `scores` onto `slp_hrs` and `stdy_hrs`. Provide verbal interpretations of the coefficients, and note whether any coefficients are deemed statisically insignificant (**hint:** regression table, as was shown during one of the lecture demos).

Solution:

```r
## replace this line with your code
linmod1 <- lm(scores ~ slp_hrs + stdy_hrs,
              data = scores_data)

linmod1 %>% summary()
```

```
Call:
lm(formula = scores ~ slp_hrs + stdy_hrs, data = scores_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5207 -1.2688  0.0372  1.3398  4.8695 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.25762    1.01522  -1.239    0.217    
slp_hrs      1.00831    0.12034   8.379 1.07e-14 ***
stdy_hrs     1.14259    0.04006  28.524  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.901 on 194 degrees of freedom
Multiple R-squared:  0.8238,	Adjusted R-squared:  0.822 
F-statistic: 453.5 on 2 and 194 DF,  p-value: < 2.2e-16
```

- Holding all else constant, sleeping one hour more appears to be associated with a 1.008-point increase in midterm score.
- Holding all else constant, studying one hour more appears to be associated with a 1.143-point increase in midterm score.
- Holding all else constant, and assuming the model extends to the origin, studying for 0 hours and sleeping 0 hours the night before the midterm is associated with scoring a -1.258 on the exam. (Remember, though, that interpreting the slope is tricky and is avoided in some cases.)

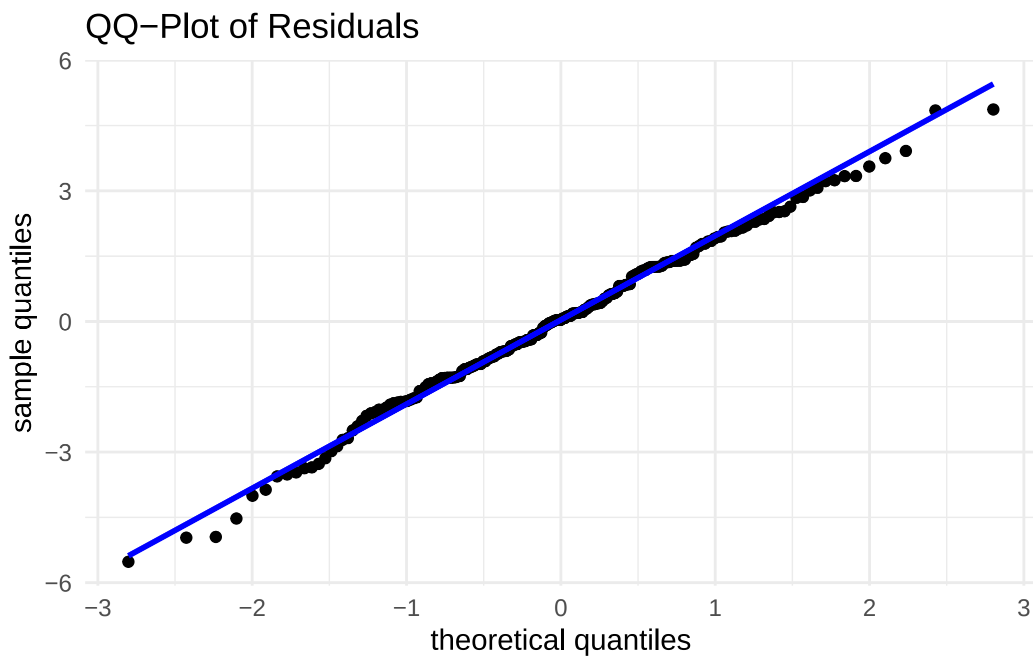
Answer Check:
There is no autograder for this question.

! Question 6

Produce a QQ-plot of the residuals. Does the normality assumption appear to be violated?

Solution:

```r
## replace this line with your code
data.frame(x = linmod1$residuals) %>%
  ggplot(aes(sample = x)) +
  geom_qq() +
  geom_qq_line(linewidth = 1,
               col = "blue") +
  theme_minimal() +
  xlab("theoretical quantiles") +
  ylab("sample quantiles") +
  ggtitle("QQ-Plot of Residuals")
```



There does *not* appear to be much deviation from linearity (even in the tails), leading us to conclude that the residuals are fairly normally distributed.

Answer Check:
There is no autograder for this question.

! Question 7

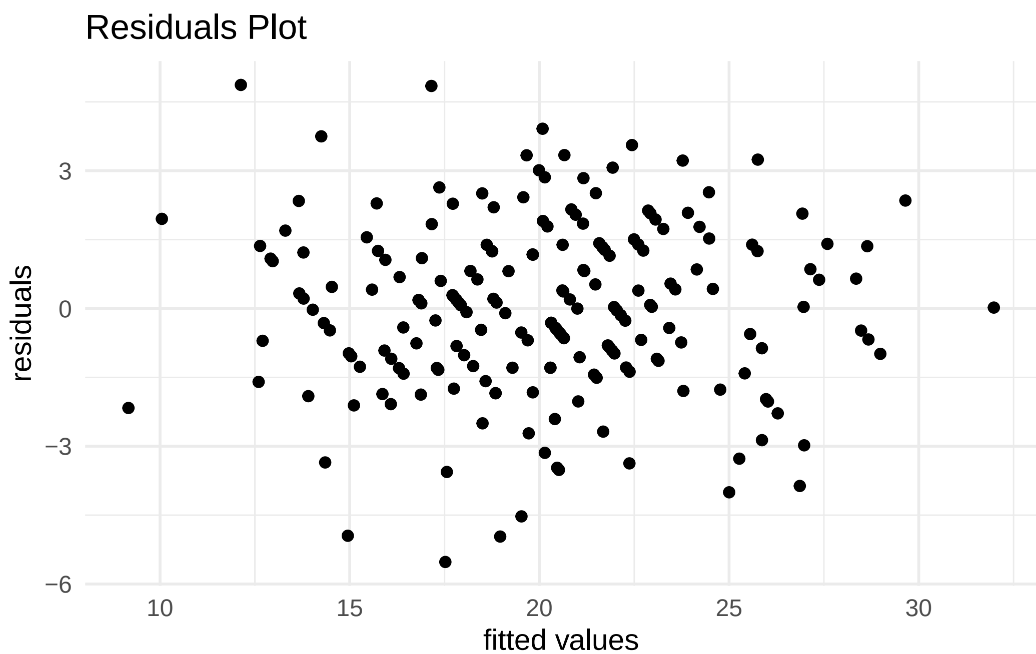
Produce a residuals plot. As a hint: save your call to `lm()` from Question 7 as a variable so that you can use `$residuals` and `$fitted.values` to access the residuals and fitted values.

Comment on the plot. Specifically:

- Are there any outliers? If so, are they influential points, points of high leverage, or both? How can you tell?
- Is there any heteroskedasticity apparent? If so, what is the nature of the heteroskedasticity (e.g. is the variance increasing *linearly* with the mean? quadratically? etc.)

Solution:

```
## replace this line with your code

data.frame(x = linmod1$fitted.values,
           y = linmod1$residuals) %>%
  ggplot(aes(x = x, y = y)) +
  geom_point() +
  theme_minimal() +
  xlab("fitted values") +
  ylab("residuals") +
  ggtitle("Residuals Plot")
```



There do not appear to be any outliers, and there does not appear to be any marked heteroskedasticity.

Answer Check:
There is no autograder for this question.

Submission Details

1) Check that all of your tables, plots, and code outputs are rendering correctly in your final `.pdf`.
2) Check that you passed all of the test cases (on questions that have autograders). You'll know that you passed all tests for a particular problem when you get the message "All tests passed!".
3) Submit **ONLY** your `.pdf` to Gradescope. Make sure to match pages to your questions - we'll be lenient on the first few labs, but after a while failure to match pages will result in point penalties.