Linear Regression and Using Quarto in Rstudio

Lab 2 - MATH 4322

Quarto

- Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

- When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
#| echo: fenced

1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
#| echo: true

2 * 2
```

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

- To use Quarto, you need to install this as a package `install.packages("quarto")`

Set Up for Lab

- Open up `RStudio`.
- Click `File` → `New File` → `Quarto Document`.
- Select which output format you want to use.
- You will be instructed to give the output of `R` and answer questions for this lab.
- Make sure you save this file.
- The questions that I want you to respond to will be in red. Type that answer in your `.qmd` file.
- After finishing you can **Render** the file to get a finished file with text and `R` output.
- If you use this `.qmd` file you need to delete in the code blocks `#| results: hide`, `#| fig-show: hide` and `#| eval: false`.
- *Note*: If you want to use PDF output you have to install the package `tinytex` in `RStudio` then you can use LaTex syntax.

Code Chunks

- The R Markdown is a text file with code chunks.
- You type in any text and if you want to use `R`.
- To insert a code chunk you type in ```` ```{r} ```` at the beginning and ```` ``` ```` at the end.
- To get the headings type in a double hashtag, #.

Task 1

- We will use the `Boston` data set from the `ISLR2` library.
- You will need to install that package and call it. In your console type in.

```
install.packages("ISLR2")
```

- In your R Markdown file type in as a chunk

```
library(ISLR2)
head(Boston)
```

```
     crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

- We are wanting to find a linear model with `medv` (median house value per $1000) as the response (output) and `rm` (average number of rooms per dwelling) as the predictor (input).

- Question 1: For the 6th suburb of Boston what is the median house value and the average number of rooms per dwelling?

Answer

```
Boston$medv[6]
```

```
[1] 28.7
```

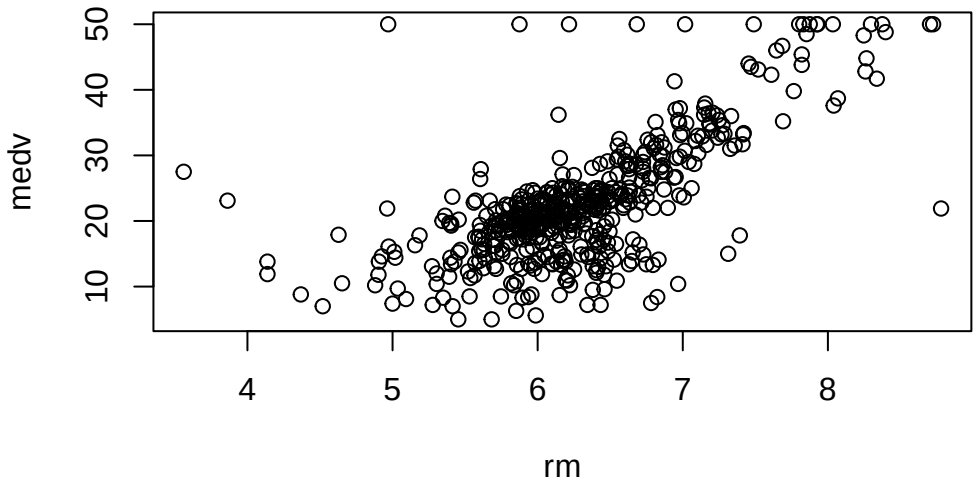
```
Boston$rm[6]
```

```
[1] 6.43
```

Task 2

- We can add plots to the rendered file by in the code chunk type in

```
plot(Boston$rm,Boston$medv,xlab = "rm",ylab = "medv")
```



- Question 2: According to the plot what is the relationship between median value of homes and average number of rooms per dwelling?

Answer This is positive, somewhat linear, and somewhat strong

Task 3

- We can use the functions `wich.max()`, `which.min()`, and `wich()` to see certain observations. Type and run the following in `R`.

```
which.max(Boston$rm)
```

```
[1] 365
```

```
which.min(Boston$rm)
```

```
[1] 366
```

- Question 3: Which observation has the largest average number of rooms per dwelling? What is the largest average number of rooms per dwelling?

```
Boston$rm[365]
```

```
[1] 8.78
```

- Question 4: Which observation has the smallest average number of rooms per dwelling? What is the smallest average number of rooms per dwelling?

```
Boston$rm[366]
```

```
[1] 3.561
```

Task 4

- In a code chunk type:

```
lm.fit <- lm(medv ~ rm, data = Boston)
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ rm, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-23.346  -2.547   0.090   2.986  39.433

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,    Adjusted R-squared:  0.4825
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

- Question 5: Give the linear model equation.

Answer

$$\hat{medv} = -34.671 + 9.102\ rm$$

- Question 6: What is the percent of variation of `medv` that can be explained by this model?

Answer

```
summary(lm.fit)$r.squared
```

```
[1] 0.4835255
```

- Question 7: Is `rm` a good predictor for `medv`? Justify your answer.

Answer Not really a good fit. This says that about 48% of the variation in the median value can be accounted for by this linear equation.

Task 5

- In a code chunk type:

```
confint(lm.fit)
```

```
                2.5 %     97.5 %
(Intercept) -39.876641 -29.464601
rm            8.278855   9.925363
```

- Question 8: What is the 95% confidence interval for the slope β_1 of this model?

Answer (8.279, 9.9254)

Task 6

- The `predict()` function can be used to produce predictions, confidence interval and prediction intervals for the prediction of `medv` for a given value of `rm`.
- The **confidence interval** is used to determine the *average* predicted value for the response variable.
- The **prediction interval** is used to determine the prediction for *one* observation of the response variable.
- Suppose we want to determine a predicted value of `medv` based on the average number of rooms per dwelling at 5, 6, and 7. We can type the following in a code chunk

```
predict(lm.fit, data.frame(rm = c(5, 6, 7)))
```

```
       1        2        3
10.83992 19.94203 29.04414
```

```
predict(lm.fit, data.frame(rm = c(5, 6, 7)), interval = "confidence")
```

```
       fit       lwr      upr
1 10.83992  9.634769 12.04508
2 19.94203 19.318469 20.56560
3 29.04414 28.219061 29.86922
```

```
predict(lm.fit, data.frame(rm = c(5, 6, 7)), interval = "prediction")
```

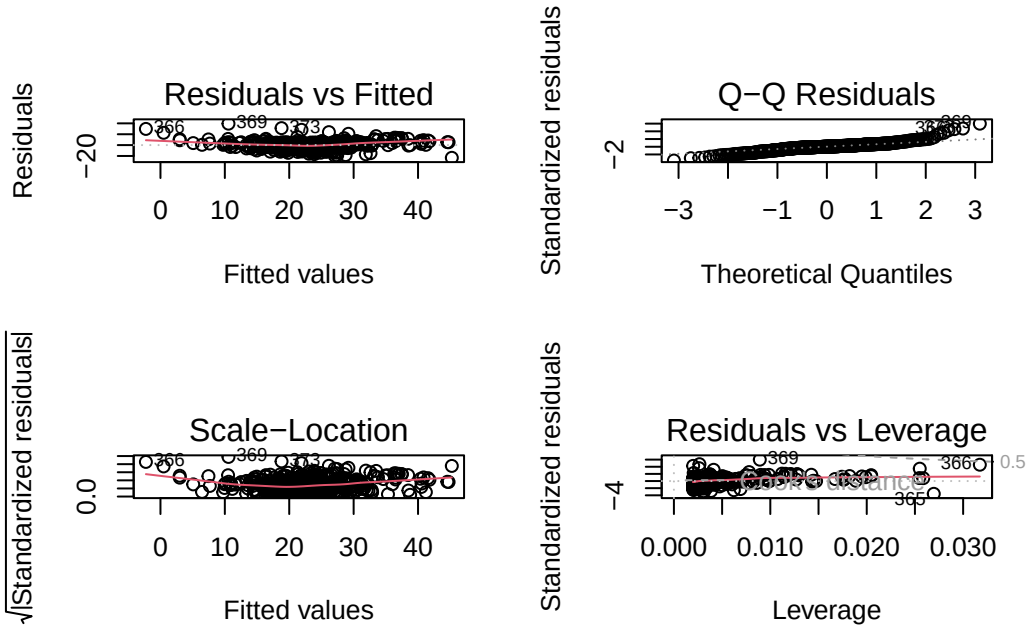
```
       fit       lwr      upr
1 10.83992 -2.214474 23.89432
2 19.94203  6.928435 32.95563
3 29.04414 16.019333 42.06895
```

- Question 9: What is the predicted median value of homes where the average number of rooms per dwelling is 5?
  Answer $10,840
- Notice that the **confidence interval** for 5 is [9.634, 12.045]. The interpretation is: on *average* the median value of the homes in all of the suburbs with average of 5 rooms is between $9,634 and $12,045.
- Notice that the **prediction interval** for 5 is [-2.214, 23.894]. The interpretation is: if we look at *one* suburb, the predicted median home value for that suburb will be between -$2,214 and $23,894.

Task 7

- We can check assumptions through the plots of the model.
- Using the code chunk type:

```
par(mfrow = c(2,2))
plot(lm.fit)
```



- Question 10: Do there appear to be extreme values?
  Answer Observation number 366
- We can use the leverage statistics to determine extreme values. The function to find the leverage statistics `hatvalues()`.
- Using the code chunk type:

```
which.max(hatvalues(lm.fit))
```

```
366
366
```

- The `which.max()` function identifies the index (row) of the largest element of a vector.
- Question 11: Which row has the largest leverage?
  Answer 366

- Using the code chunk type: `Boston[number of largest leverage,]`.

```
Boston[366,]
```

```
      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
366 4.55587  0  18.1    0 0.718 3.561 87.9 1.6132  24 666    20.2  7.12 27.5
```

- Question 12: How many average number of rooms per dwelling and what is the median value of the homes in this suburb?
  Answer rm = 3.5611, medv = 27.5

Completing

- Make sure you save this file.
- You can click on the `Render` icon in the tool bar.
- Upload the rendered file (PDF) to Canvas under Lab 2 in today's lecture.