Linear Regression and Using R Markdown

Cathy Poliak cpoliak@central.uh.edu

Lab 2 - Spring 2023

R Markdown

- Introduction
- An R Markdown file is a plain text file that has the extension `.Rmd`.
- When this is opened in RStudio the file becomes a notebook interface for R.
- We will use Rmarkdown for today's lab. For more information on how it works see: R Markdown
- Here is a cheat sheet for R Markdown: Cheat Sheet

Set Up for Lab

- Open up `RStudio`.
- Click `File` → `New File` → `R Markdown`.
- If you do not find `R Markdown` in you can install R Markdown by typing in the Console:

```
install.packages("rmarkdown")
```

- 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 `R Markdown` file.
- The questions that I want you to respond to will be in red. Type that answer in your `R Markdown` file.
- After finishing you can Knit the `R Markdown` file to get a finished file with text and `R` output.
- *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)
```

```
## Warning: package 'ISLR2' was built under R version 4.2.2
```

```
head(Boston)
```

- 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?

medv = 28.7 rm = 6.430

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?

It is a Positive Linear Model

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?

Observation = 365, rm = 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?

Observation 366, rm = 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.

Y = -34.671 + 9.102x

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

Percent of variation = 0.4835

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

No, since it counts for below 50% of the standard variation

Task 5

- In a code chunk type:

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

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

95% confidence interval is 8.278855, 9.925363

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?

The predicted mean value is 10.83992

- 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 $12,45.

- 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?

Yes, there are extreme values

- 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))
```

- The `which.max()` function identifies the index (row) of the largest element of a vector.

- Question 11: Which row has the largest leverage?

Row 366

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

```
Boston[366,]
```

- Question 12: How many average number of rooms per dwelling and what is the median value of the homes in this suburb?

rm = 3.561, medv = 27.5

Completing

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