MATH 254 - Statistical Modeling and Applications - Lab 3

STEMI Hospital data

Tural Sadigov

9/16/22

Table of contents

0.1 NAME: ________________ 1

0.2 Who are you working with? ________________ 1

1 Understanding the data at hand 2

2 Linear Regression (1st machine learning algorithm) 3

3 Bootstrap methods 4

0.1 NAME: ________________

0.2 Who are you working with? ________________

In 2002-2003, a study was conducted on in-hospital deaths from myocardial infarction with ST elevation (**STEMI**). STEMI is a very serious type of heart attack that blocks one of the major arteries that carries oxygen to heart. Study was conducted in 9 hospitals in Providence Health system in Oregon. There were 913 STEMI patients that were treated in these hospitals, and 105 of them died. Each of these patients were assigned a **Thrombolysis in Myocardial Infarction Risk Score** that ranged between 0-14. Higher the risk score, higher the risk of death. Data contains 13 cases (different risk groups), the risk score of each category, number of patients in each risk group, observed deaths within each risk group and national mortality rate of death within each group.

In this mini-project, we will investigate the relationship between *observed deaths* and *expected deaths* within each category using *linear regression.* We will estimate the coefficients in the linear regression and using *bootstrap methods*, we will find *standard errors of these estimates* and *Bootstrap Percentile Confidence Intervals*.

Edit/write your code whenever you see three dots (…) in the code chunk.

Load libraries and data.

```
# libraries
library(tidyverse)
library(infer)


# Import Hospital data from GitHub
hospital = read_csv(...)

# view the  the hospital data
hospital

# data wrangling
hospital_df <-
  hospital %>%
  mutate(expected_deaths = Patients * NRMI_Mortality_in_percents/100) %>%
  select(Deaths, expected_deaths) %>%
  rename(observed_deaths = Deaths) %>%
  relocate(observed_deaths, .after = expected_deaths)

# view the data we will be using
hospital_df
```

1 Understanding the data at hand

1. Create a scatterplot of observed deaths (y) vs expected deaths (x). Makes sure your plot contains correct labels, title, etc.

   ```
   # add your code below
   hospital_df %>%
     ggplot(aes(x = ...,
                y = ...)) +
     geom_point(size = 5)
   ```

2. Comment on the scatterplot.

- Do you think there is a positive or negative trend?
- Do you think the TRUE relationship between observed deaths and expected deaths is linear or nonlinear?
- How strong is the relationship?

2 Linear Regression (1st machine learning algorithm)

In linear regression, we assume that there is a TRUE linear relationship between two variables with some normal error, z, that has $\mu_z = 0$ and $\sigma_z = \sigma$.

$$y = \beta_0 + \beta_1 x + z$$

This linear model has three UNKNOWN (true) parameters: β_0, β_1, σ. Here β_0 is the true y-intercept, β_1 is the true slope and σ is the true standard deviation of the errors. We will estimate these model parameters using the data at hand, and fit a model. Our estimators will be $\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}$, and our fitted model will be

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

In the second part of the course, we will investigate regression (as part of machine learning algorithms) in details. In turns out that, for the least squares regresson line, the true slope, β_1, of the model can be approximated by

$$\beta_1 \approx \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

3. Using the formula above, find $\hat{\beta}_1$ in the R chunk below. What is your estimate for the true slope?

```r
# add your code below
x <-  hospital_df$expected_deaths
y <-  hospital_df$observed_deaths
x_bar <-  mean(x)
y_bar <-  ...
beta_1_hat <-  sum(...)/sum(...)
beta_1_hat
```

It also turns out that, pair $(\bar{x}, \bar{y})$, the center of the data cloud, always lies right on the least squares regression line. Using this, one can estimate

$$\beta_0 \approx \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

4. Using the formula above, find $\hat{\beta}_0$ in the R chunk below. What is your estimate for true y-intercept?

```
# add your code below
beta_0_hat <-  y_bar - ...
beta_0_hat
```

5. R has its own command of fitting the least squares regression line to bivariate data. Use the code below to find fitted/estimated y-intercept,$\hat{\beta}_0$, (under Intercept) and fitted/estimated slope, $\hat{\beta}_1$ (under x, expected deaths). Do these estimates match with your estimates in previous parts of the mini-project?

```
lm(observed_deaths ~ expected_deaths,
   data = hospital_df)
```

6. Impose the fitted regression line to the scatterplot you created (part (1)) and have the point $(\bar{x}, \bar{y})$ to show up explicitly on the plot. Does your regression line indeed contain $(\bar{x}, \bar{y})$?

```
# create scatterplot again
hospital_df %>%
  ggplot(aes(x = ...,
             y = ...)) +
  geom_...(size = 3) +
  geom_smooth(method = 'lm', lwd = 1) +
  geom_point(aes(x_bar, y_bar),
             color = "red", size = 6)
```

3 Bootstrap methods

7. Explain each line the code below by commenting directly in the R chunk that creates Bootstrap distributions of $\hat{\beta}_1$ and $\hat{\beta}_0$ (with smoothed histograms imposed).

```r
set.seed(2022)
boot_estimates <-
  hospital_df %>%
  specify(observed_deaths ~ expected_deaths) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 100, type = "bootstrap") %>%
  fit()


boot_estimates %>%
  ggplot(aes(x = estimate, fill = term)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~term, scales = "free", ncol = 1)
```

8. What are the standard errors of both estimates, $\hat{\beta}_1$ and $\hat{\beta}_0$? Interpret the standard errors.

```r
boot_estimates %>%
  pivot_wider(names_from = term,
              values_from = estimate) %>%
  ungroup() %>%
  summarise(s_intercept = ...,
            s_x = ...)
```

9. Create 99% Bootstrap Percentile Confidence Intervals for true slope, β_1, and true y-intercept, β_0. Interpret these intervals. Do you think it is still plausible that the slope is 0?

```r
observed_fit <-
  hospital_df %>%
  specify(observed_deaths~expected_deaths) %>%
  fit()

boot_estimates %>%
  get_...(level = ...,
          type = 'se',
          point_estimate = observed_fit)
```

10. Using built in R routine, *summary(model)*, one can obtain many statistics (such as fitted/estimated intercept and slope) and their standard errors using classical methods (not bootstrap). What are the standard errors of estimated y-intercept and estimated

slope from the output of '*summary(model)*? Do they agree with your standard errors from Bootstrap method above?

```r
model <- lm(observed_deaths~expected_deaths,
            data = hospital_df)

summary(model)
```