Lab 8

PUBLISHED
Invalid Date

Due: Friday, April 12th at 8:30am

Goals of this lab

- In this lab, you will work with observational data to assess the relationship between an exposure and the incidence of a disease.
- You will practice bootstrap to calculate a 95%CI.
- You will create an interactive web application with `shiny`.

Exercise 1:

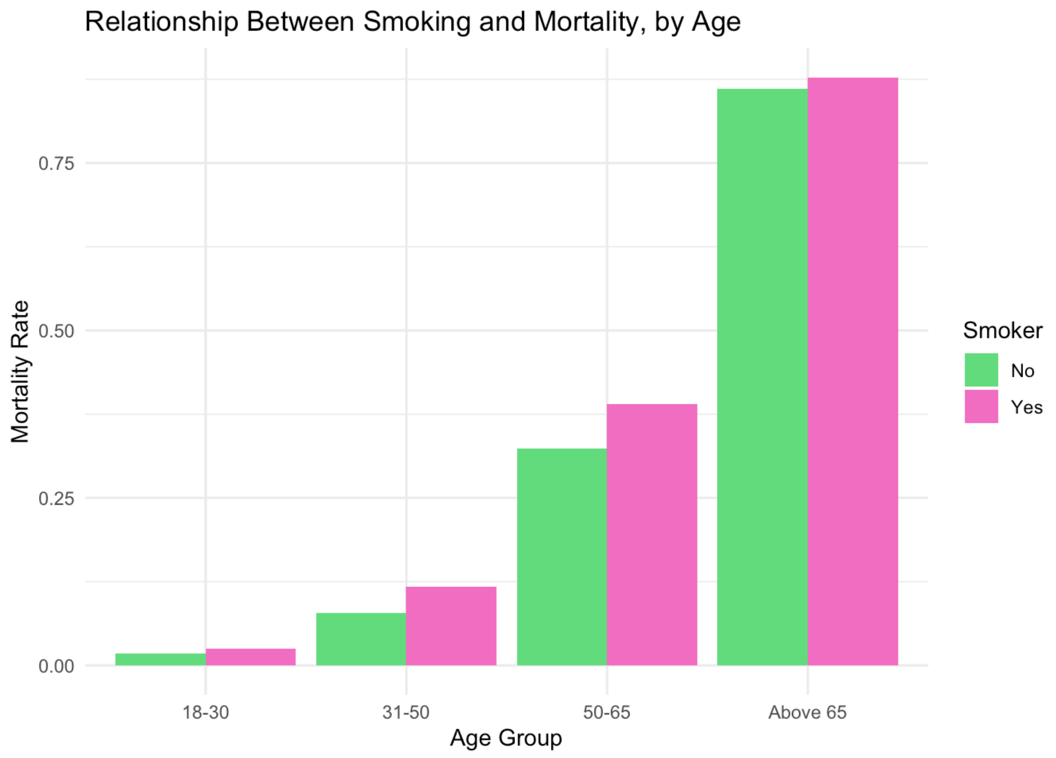
The `Whickham` data set in the `mosaicData` package includes data on age, smoking, and mortality from a one-in-six survey of the electoral roll in Whickham, a mixed urban and rural district near Newcastle upon Tyne, in the United Kingdom. The survey was conducted in 1972–1974 to study heart disease and thyroid disease. A follow-up on those in the survey was conducted 20 years later. Describe the association between smoking status and mortality in this study. Be sure to consider the role of age as a possible confounding factor.

```r
library(mosaicData)
data("Whickham")

Whickham <- Whickham %>%
  mutate(
    age_group = case_when(
      age >= 18 & age <= 30 ~ "18-30",
      age > 30 & age <= 50 ~ "31-50",
      age > 50 & age <= 65 ~ "50-65",
      age > 65 ~ "Above 65",
      TRUE ~ "Other"
    )
  )

mortality_rates <- Whickham %>%
  group_by(smoker, age_group) %>%
  summarise(
    mortality_rate = sum(outcome == "Dead") / n(),
  )

ggplot(mortality_rates, aes(x = age_group, y = mortality_rate,
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Age Group", y = "Mortality Rate", fill = "Smoker")
  ggtitle("Relationship Between Smoking and Mortality, by Age")
    scale_fill_manual(values = c("Yes" = "#F270C4", "No" = "#65
  theme_minimal()
```



As evident in my bar graph, smoking is generally associated with greater mortality, across age groups. The relationship weakens as age groups get older likely due to generally increasing mortality with age.

Exercise 2:

a. Calculate the mean age of birthing people from the `Gestation` data set from the `mosaicData` package.

```
data("Gestation")

mean_age <- mean(Gestation$age, na.rm = TRUE)
mean_age
```

```
[1] 27.25527
```

b. Use the bootstrap to generate and interpret a 95% confidence interval for the mean age of birthing people.

```
bootstrap_means <- Gestation %>%
  rep_sample_n(size = nrow(Gestation), replace = TRUE, reps = 1
  group_by(replicate) %>%
  summarize(x_bar = mean(age, na.rm=T))

bootstrap_means %>%
  slice_head(n = 10)
```

```
# A tibble: 10 × 2
   replicate x_bar
       <int> <dbl>
 1         1  27.2
 2         2  27.0
 3         3  27.6
 4         4  27.2
 5         5  27.2
 6         6  26.9
 7         7  27.2
 8         8  27.2
 9         9  27.3
10        10  27.2
```

```
q <- bootstrap_means %>%
  summarize(
    lower_bound = quantile(x_bar, probs = c(.025)),
    upper_bound = quantile(x_bar, probs = c(.975))
  )
q
```

```
# A tibble: 1 × 2
  lower_bound upper_bound
        <dbl>       <dbl>
1        26.9        27.6
```

If we repeated our sampling procedure a large number of times, we expect about 95% of the resulting confidence intervals to capture the value of the mean age of birthing people

Exercise 3:

Using data from the `palmerpenguins` package, create a Shiny app that displays measurements from the penguins dataframe. Allow the user to select a species or a gender, and to choose between various attributes on a scatterplot. (Hint: examples of similar apps can be found at the Shiny gallery).

```
library(palmerpenguins)
data("penguins")

library(shiny)

# User interface
ui <- fluidPage(
  titlePanel("Let's Measure Some Penguins!"),
  sidebarLayout(
    sidebarPanel(
      # Text input widget for penguin species
      selectizeInput(inputId = "species",
                     label = "Enter Penguin Species Here",
                     choices = NULL,
                     multiple = TRUE),

      p("Put single space between the species."),

      # Button input widget for penguin sex
      radioButtons(inputId = "sex",
                   label = "Sex",
                   choices = c("male", "female"),
                   selected = "female"),

      # Button input widgets for measurement attributes
      #x variable
      radioButtons("x_var",
                   label = "Select X Variable:",
                   choiceNames = list("Bill Length", "Bill Dept
                   choiceValues = list("bill_length_mm", "bill_
                   selected = c("bill_length_mm"),
                   inline = TRUE),
      #y variable
      radioButtons("y_var",
                   label = "Select Y Variable:",
                   choiceNames = list("Bill Length", "Bill Dept
                   choiceValues = list("bill_length_mm", "bill_
                   selected = c("bill_depth_mm"),
                   inline = TRUE),

      submitButton("Update Results!")
    ),

    mainPanel(
      plotOutput(outputId = "graph")
    )
  )
)

server <- function(input, output, session){

  updateSelectizeInput(session, 'species',
                       choices = unique(penguins$species),
                       server = TRUE)

  dat_penguins <- reactive({
    penguins %>%
      filter(species %in% c(unlist(str_split(input$species, " '
             sex == input$sex)
  })

   output$graph <- renderPlot({

    ggplot(data = dat_penguins(),
         mapping = aes(x = input$x_var, y = input$y_var, color
      geom_point()
  })

}

# Creates app
shinyApp(ui = ui, server = server)
```

Let's Measure Some Penguins!

Enter Penguin Species Here

Put single space between the species.

Sex

○ male [✓ female]

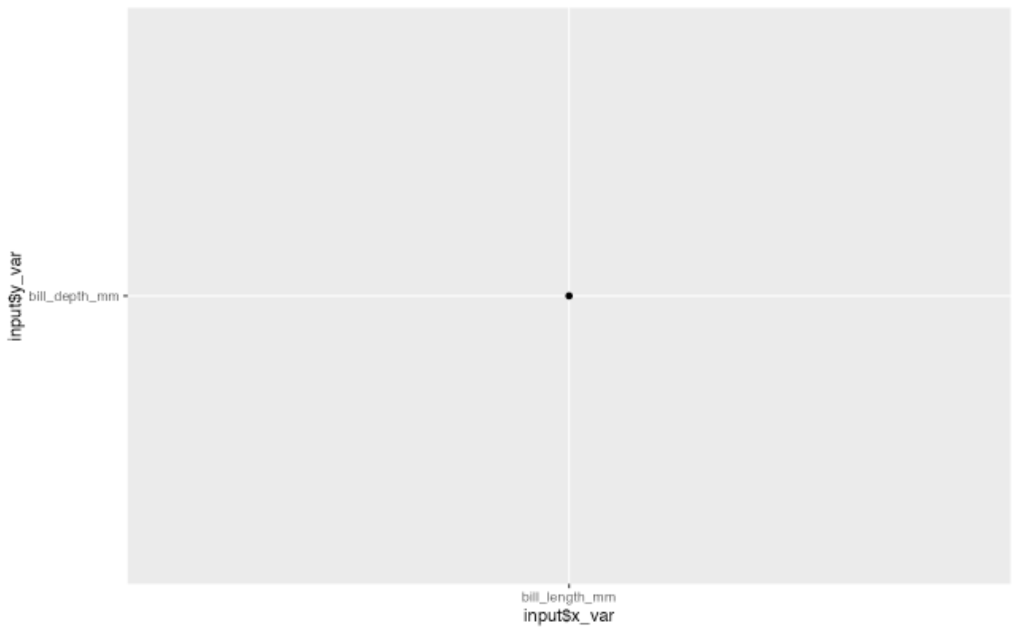
Select X Variable:

[✓ Bill Length] ○ Bill Depth ○ Flipper Length ○ Body Mass

Select Y Variable:

○ Bill Length [✓ Bill Depth] ○ Flipper Length ○ Body Mass

Update Results!


input$y_var
bill_depth_mm
bill_length_mm
input$x_var