PSTAT100 Lab6: Regression (Solution)

Introduction

This lab covers the nuts and bolts of fitting linear models. The linear model expresses a response variable, y, as a linear function of $p-1$ explanatory variables $x_1, \ldots, x_{p-1}$ and a random error ϵ. Its general form is:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_{p-1} x_{p-1} + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

Usually, the response and explanatory variables and error term are indexed by observation $i = 1, \ldots, n$, so that the model describes a dataset comprising n values of each variable:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i,p-1} + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

Matrix Form Representation

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

where:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \ldots & x_{1,p-1} \\ 1 & x_{21} & \ldots & x_{2,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{n,p-1} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Fitting a model of this form means **estimating the parameters** $\beta_0, \ldots, \beta_{p-1}$ and σ^2 from a set of data.

Estimation using Ordinary Least Squares (OLS)

- The OLS estimate of β is given by:

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

- The error variance σ^2 is estimated as:

$$\widehat{\sigma}^2 = \frac{1}{n-p-1}(\mathbf{y} - \mathbf{X}\widehat{\beta})'(\mathbf{y} - \mathbf{X}\widehat{\beta})$$

When fitting a linear model, it is also of interest to quantify uncertainty by estimating the variability of $\widehat{\beta}$ and measure overall quality of fit. This lab illustrates that process and the computations involved.

Objectives

In this lab, you'll learn how to:

- compute OLS estimates;
- calculate fitted values and residuals;
- compute the error variance estimate;
- compute the variance-covariance matrix of $\widehat{\beta}$, which quantifies the variability of model estimates;
- compute standard errors for each model estimate;
- compute the proportion of variation captured by a linear model.

Throughout you'll use simple visualizations to help make the connection between fitted models and the aspects of a dataset that model features describe.

Data: Fertility Rates

By way of data, you'll work with country indicators, total fertility rates, and gender indicators for a selection of countries in 2018, and explore the decline in fertility rates associated with developed nations.

The data are stored in separate `.csv` files and imported below:

```
# Load necessary library
library(dplyr)
library(tidyr)
library(ggplot2)

# Read the data
fertility <- read.csv("data/fertility.csv")
country <- read.csv("data/country-indicators.csv")
gender <- read.csv("data/gender-data.csv")
```

The variables you'll work with in this portion are the following:

Dataset	Name	Variable	Units
fertility	`fertility_total`	National fertility rate	Average number of children per woman
country	`hdi`	Human development index	Index between 0 and 1 (0 is lowest, 1 is highest)
gender	`educ_expected_yrs_f`	Expected years of education for adult women	Years

Because the variables of interest are stored in three separate dataframes, you'll first need to **extract** them and **merge by country**.

```
# Select variables of interest
fertility_sub <- fertility %>% select(Country, fertility_total)
gender_sub <- gender %>% select(Country, educ_expected_yrs_f)
country_sub <- country %>% select(Country, hdi)

# Merge datasets
reg_data <- fertility_sub %>%
  inner_join(gender_sub, by = "Country") %>%
  left_join(country_sub, by = "Country") %>%
  drop_na()
```

```r
# Preview data
head(reg_data, 4)
```

```
      Country fertility_total educ_expected_yrs_f   hdi
1 Afghanistan           4.473            6.795722 0.509
2     Albania           1.617           13.201755 0.792
3     Algeria           3.023           12.108990 0.746
4      Angola           5.519            6.973901 0.582
```

We'll treat the fertility rates as our variable of interest.

Exploratory analysis

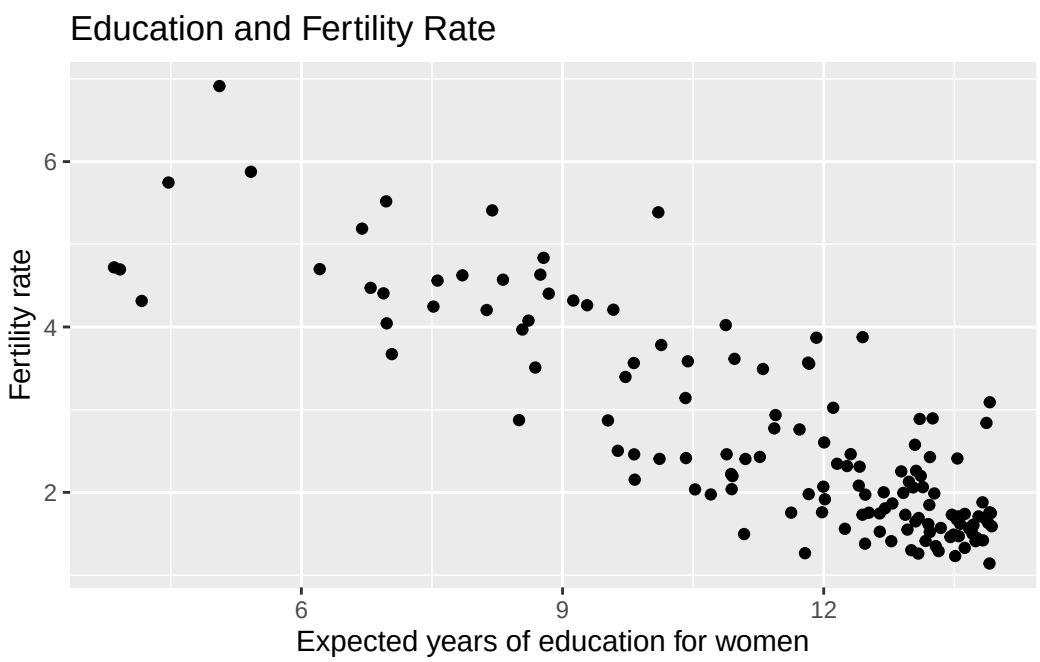
A preliminary step in regression analysis is typically data exploration through scatterplots. The objective of exploratory analysis in this context is to identify **an approximately linear relationship** to model.

Question 1: Education and fertility rate

Construct a **scatterplot** of total **fertility** against **expected years of education for women**. Label the axes 'Fertility rate' and 'Expected years of education for women'. Store this plot as `scatter_educ` and display the graphic.

YOUR ANSWER:

```r
library(ggplot2)
# Scatterplot of fertility rate vs. expected years of education for women
scatter_educ <- ggplot(reg_data, aes(x = educ_expected_yrs_f,
                                     y = fertility_total)) +
  geom_point() +
  labs(x = "Expected years of education for women", y = "Fertility rate",
       title = "Education and Fertility Rate")
scatter_educ
```



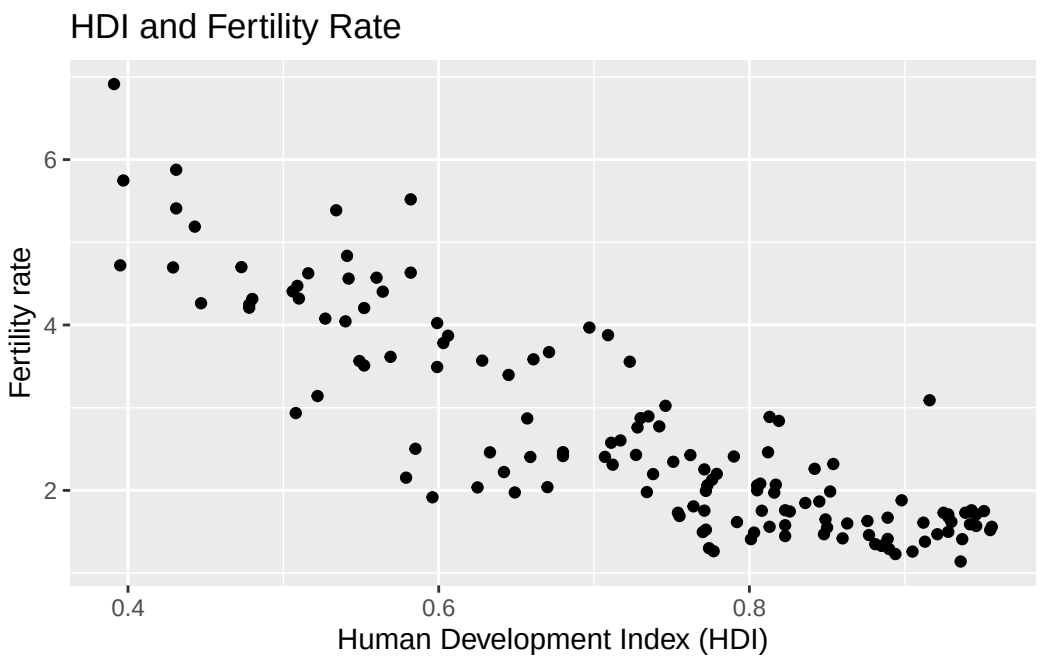
This figure shows a clear **negative association** between fertility rate and women's educational attainment, and that the relationship is **roughly linear**. Next, check whether **HDI** seems to be related to fertility rate.

Question 2: HDI and fertility rate

Now construct a **scatterplot** comparing **fertility rate** with **HDI**. Make sure you choose appropriate labels for your axes and plot. Store this plot as `scatter_hdi` and display the graphic.

YOUR ANSWER:

```
# Scatterplot of fertility rate vs. HDI
scatter_hdi <- ggplot(reg_data, aes(x = hdi, y = fertility_total)) +
  geom_point() +
  labs(x = "Human Development Index (HDI)", y = "Fertility rate",
       title = "HDI and Fertility Rate")
scatter_hdi
```



This figure shows **a negative relationship** between fertility rate and HDI; it may **not be exactly linear**, but a line should provide a decent approximation. So, the plots suggest that a **linear regression model** in one or both explanatory variables is reasonable.

Simple linear regression

To start you'll fit a **simple linear model** regressing **fertility on education**.

First we'll need to store the quantities – the response and explanatory variables – needed for model fitting in the proper format. Recall that the linear model in matrix form is:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

where:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Notice that the explanatory variable matrix $\mathbf{X}$ includes a column of ones for the intercept. So, the quantities needed are:

- $\mathbf{y}$, a one-dimensional array of the total fertility rates for each country.
- $\mathbf{X}$, a two-dimensional array with a column of ones (intercept) and a column of the expected years of education for women (explanatory variable).

The cell below constructs these arrays in R:

```
# Retrieve response variable
y <- reg_data$fertility_total

# Construct explanatory variable (matrix)
x <- reg_data$educ_expected_yrs_f
x_with_leading1 <- model.matrix(~ x)

# Print first few rows of X
head(x_with_leading1)
```

```
  (Intercept)         x
1           1  6.795722
2           1 13.201755
3           1 12.108990
4           1  6.973901
5           1 12.914441
6           1 13.061253
```

Estimation

Fitting a model refers to computing estimates; the `lm()` function in R will fit a linear regression model based on the response vector and explanatory variable matrix. The model structure follows:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

The following code fits the simple linear model:

```
# Fit simple linear model
lm_fit <- lm(y ~ x, data = reg_data)

# Display summary of results
summary(lm_fit)
```

```
Call:
lm(formula = y ~ x, data = reg_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4155 -0.4242 -0.0576  0.2805  2.1930

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.51142    0.26236   28.63   <2e-16 ***
x           -0.42747    0.02256  -18.95   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6617 on 137 degrees of freedom
Multiple R-squared:  0.7238,    Adjusted R-squared:  0.7218
F-statistic:   359 on 1 and 137 DF,  p-value: < 2.2e-16
```

Extracting Estimates

- The **coefficient estimates** $\hat{\beta}_0$ and $\hat{\beta}_1$ are obtained using:

```r
# Coefficients
coef(lm_fit)
```

```
(Intercept)           x
  7.5114231  -0.4274721
```

- The **error variance estimate** $\hat{\sigma}^2$ can be retrieved as:

```r
# Variance estimate
sigma_hat2 <- summary(lm_fit)$sigma^2
sigma_hat2
```

```
[1] 0.4378592
```

- The **variance-covariance matrix** of the **estimated coefficients** is:

$$\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$$

  which can be retrieved in R using:

```r
# Variance-covariance matrix of coefficients
vcov(lm_fit)
```

```
             (Intercept)             x
(Intercept)  0.068833597 -0.0057817958
x           -0.005781796  0.0005089428
```

Model Interpretation

A standard metric often reported with linear models is the R^2 **score**, which quantifies the **proportion of variation in the response explained by the model**:

```r
# Compute R-squared
summary(lm_fit)$r.squared
```

```
[1] 0.7238143
```

So, the expected years of education for women in a country explains 72.38% of variability in fertility rates, and furthermore, according to the fitted model:

- For a country in which women are **entirely uneducated**, the estimated mean fertility rate is 7.5 children on average by the end of a woman's reproductive period.
- Each additional year of education for women is associated with a **decrease** in a country's fertility rate by an estimated 0.43.
- After accounting for women's education levels, fertility rates vary by a standard deviation of $0.66 = \sqrt{0.438}$ across countries.
- This model provides an initial assessment of the relationship, but further **diagnostics** are necessary to validate assumptions.

Question 3: center the explanatory variable

Note that no countries report an expected zero years of education for women, so the meaning of the intercept is artificial. As we saw in lecture, **centering** the explanatory variable can improve **interpretability** of the intercept. **Center** the expected years of education for women and **refit** the model by following the steps outlined below. Display the **coefficient estimates** and **standard errors**.

YOUR ANSWER:

```r
# Center the education column by subtracting its mean from each value
educ_ctr <- x - mean(x)

# Fit new model
lm_ctr <- lm(y ~ educ_ctr)

# Extract results
summary(lm_ctr)
```

```
Call:
lm(formula = y ~ educ_ctr)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4155 -0.4242 -0.0576  0.2805  2.1930

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.65517    0.05613   47.31   <2e-16 ***
educ_ctr    -0.42747    0.02256  -18.95   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6617 on 137 degrees of freedom
Multiple R-squared:  0.7238,    Adjusted R-squared:  0.7218
F-statistic:   359 on 1 and 137 DF,  p-value: < 2.2e-16
```

```r
# Arrange estimates and standard errors in a dataframe and display
coef_tbl <- data.frame(
  Estimate = coef(lm_ctr),
  `Standard Error` = sqrt(diag(vcov(lm_ctr)))
)

print(coef_tbl)
```

```
             Estimate Standard.Error
(Intercept)  2.6551676     0.05612545
educ_ctr    -0.4274721     0.02255976
```

Fitted values and residuals

Fitted values

The **fitted value** for y_i is the value along the line specified by the model that corresponds to the matching explanatory variable x_i. In other words:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

These can be obtained directly from the fitted model in R:

```
# Fitted values
fitted_values <- fitted(lm_fit)

# Display first few fitted values
head(fitted_values)
```

```
       1        2        3        4        5        6
4.606441 1.868041 2.335168 4.530275 1.990860 1.928102
```

The result is an array with **length** matching the number of observations X used to fit the model. The fitted values correspond to the **predicted response** for each explanatory variable.

Residuals

Recall that **model residuals** are the **difference** between observed and fitted values:

$$e_i = y_i - \hat{y}_i$$

Residuals can be retrieved similarly as an attribute of the regression results:

```
# Obtain residuals
residuals <- residuals(lm_fit)

# Display first few residuals
head(residuals)
```

```
           1            2            3            4            5            6
-0.133441258 -0.251041225  0.687832282  0.988725204  0.003140285  0.332898098
```

Again, `residuals` is an array with **length** matching the number of observations X used to fit the model. And these residuals are returned in the same order as the original observations.

Question 4: calculations 'by hand'

Calculate the **fitted values** and **residuals** *manually*. Store the results as arrays `fitted_manual` and `resid_manual`, respectively.

Hint: Use **matrix-vector multiplication**.

YOUR ANSWER:

```
X <- x_with_leading1

# Compute fitted values manually
fitted_manual <- X %*% coef(lm_fit)

# Compute residuals manually
resid_manual <- y - fitted_manual

# Display first few values
head(fitted_manual)
```

```
      [,1]
1 4.606441
2 1.868041
3 2.335168
4 4.530275
5 1.990860
6 1.928102
```

```
head(resid_manual)
```

```
           [,1]
1 -0.133441258
2 -0.251041225
3  0.687832282
4  0.988725204
5  0.003140285
6  0.332898098
```

It is often convenient to add the **fitted values** and **residuals** as new columns in `reg_data`.

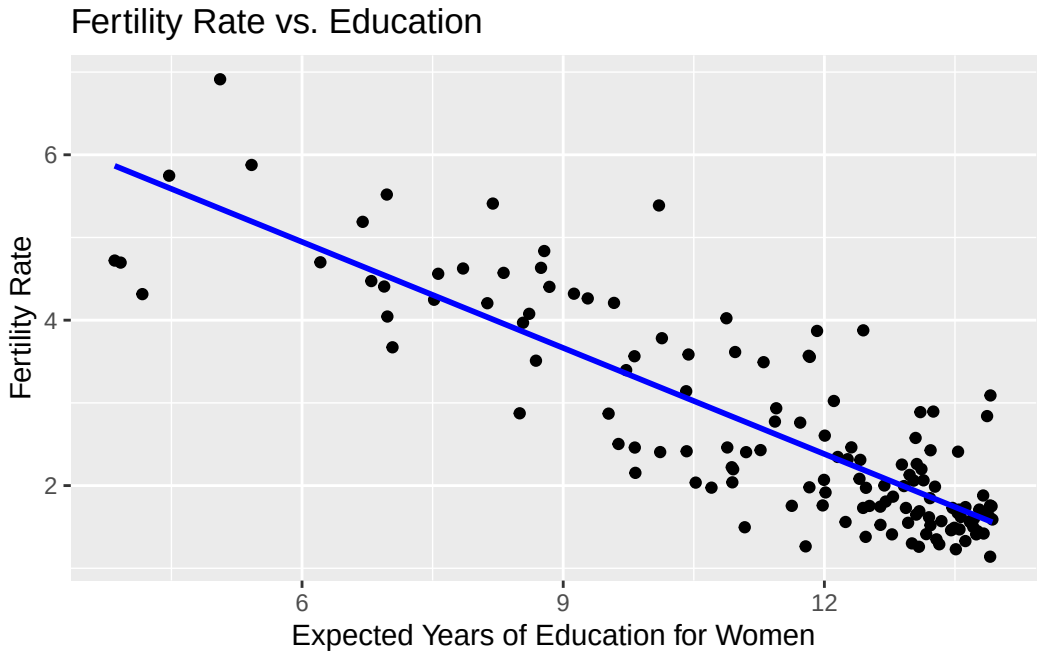
```r
# Append fitted values and residuals
reg_data$fitted_slr <- fitted(lm_fit)
reg_data$resid_slr <- residuals(lm_fit)

# Display first few rows
head(reg_data, 3)
```

```
      Country fertility_total educ_expected_yrs_f   hdi fitted_slr  resid_slr
1 Afghanistan           4.473            6.795722 0.509   4.606441 -0.1334413
2     Albania           1.617           13.201755 0.792   1.868041 -0.2510412
3     Algeria           3.023           12.108990 0.746   2.335168  0.6878323
```

Visualizing the Model

We can use this augmented dataframe to visualize the deterministic part of the model:

```r
# Construct scatterplot with fitted line
ggplot(reg_data, aes(x = educ_expected_yrs_f, y = fertility_total)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Fertility Rate vs. Education",
       x = "Expected Years of Education for Women",
       y = "Fertility Rate")
```



Uncertainty Bands

To obtain **uncertainty bands** about the **estimated mean**, we'll compute predictions at each observed value using confidence intervals.

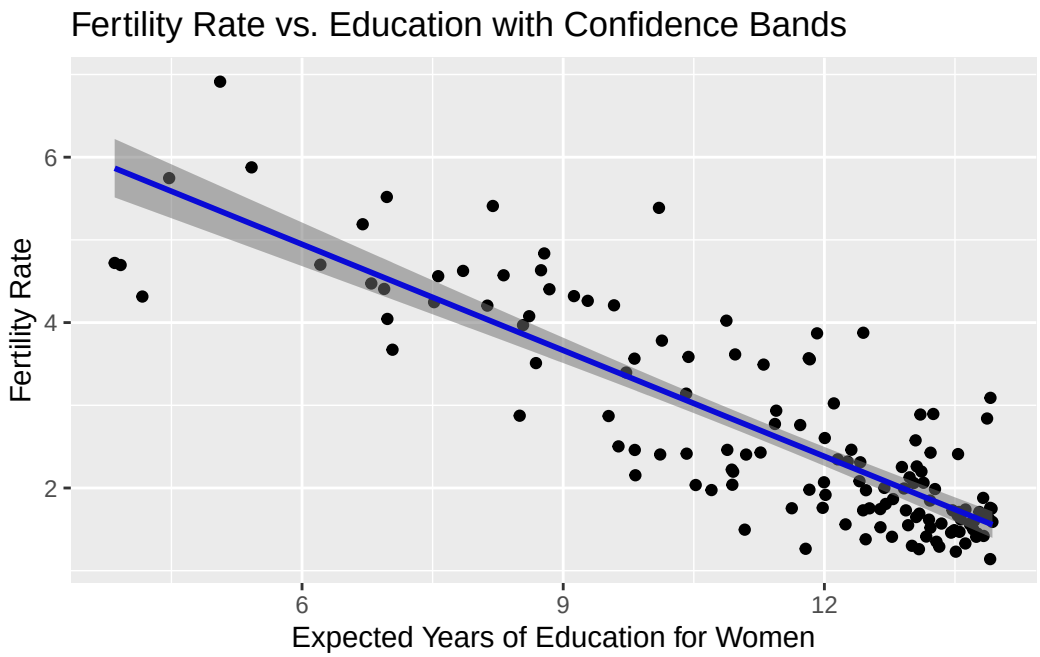
```
# Compute confidence intervals for estimated mean
conf_int <- predict(lm_fit, interval = "confidence")

# Append lower and upper bounds to the data
reg_data$lwr_mean <- conf_int[, "lwr"]
reg_data$upr_mean <- conf_int[, "upr"]

# Display first few rows
head(reg_data)
```

```
              Country fertility_total educ_expected_yrs_f   hdi fitted_slr
1         Afghanistan           4.473            6.795722 0.509   4.606441
2             Albania           1.617           13.201755 0.792   1.868041
3             Algeria           3.023           12.108990 0.746   2.335168
4              Angola           5.519            6.973901 0.582   4.530275
5 Antigua and Barbuda           1.994           12.914441 0.772   1.990860
6           Argentina           2.261           13.061253 0.842   1.928102
```

```
     resid_slr lwr_mean upr_mean
1 -0.133441258 4.374528 4.838354
2 -0.251041225 1.729965 2.006117
3  0.687832282 2.219268 2.451067
4  0.988725204 4.305309 4.755240
5  0.003140285 1.860002 2.121717
6  0.332898098 1.793660 2.062544
```

Now, we can visualize the uncertainty bands:

```
# Construct plot with uncertainty bands
ggplot(reg_data, aes(x = educ_expected_yrs_f, y = fertility_total)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  geom_ribbon(aes(ymin = lwr_mean, ymax = upr_mean), alpha = 0.2) +
  labs(title = "Fertility Rate vs. Education with Confidence Bands",
       x = "Expected Years of Education for Women",
       y = "Fertility Rate")
```



As discussed in lecture, we can also compute and display uncertainty bounds for **predicted observations** (rather than the mean).

```r
head(predict(lm_fit, interval = "prediction"))
```

```
       fit       lwr      upr
1 4.606441 3.2775637 5.935319
2 1.868041 0.5522916 3.183791
3 2.335168 1.0215602 3.648775
4 4.530275 3.2025920 5.857958
5 1.990860 0.6758481 3.305871
6 1.928102 0.6127287 3.243475
```

These will be wider, because there is more uncertainty associated with predicting observations compared with estimating the mean.

Question 5: Prediction Intervals

The **standard error** for **predictions** is stored with the output of `predict()` as part of the confidence interval calculation. The prediction standard error captures **variability** when predicting new observations rather than estimating the mean.

Use this method to compute **95% uncertainty bounds for the predicted observations**. Add the lower and upper bounds as new columns in `reg_data`, named `lwr_obs` and `upr_obs`, respectively. Construct **a plot** showing data scatter, the **model predictions**, and **prediction uncertainty bands**.

YOUR ANSWER:

```r
# Compute prediction intervals
pred_int <- predict(lm_fit, interval = "prediction")

# Store lower and upper bounds in the dataset
reg_data$lwr_obs <- pred_int[, "lwr"]
reg_data$upr_obs <- pred_int[, "upr"]

# Display first few rows
head(reg_data)
```

```
              Country fertility_total educ_expected_yrs_f   hdi fitted_slr
1         Afghanistan           4.473             6.795722 0.509   4.606441
2             Albania           1.617            13.201755 0.792   1.868041
3             Algeria           3.023            12.108990 0.746   2.335168
4              Angola           5.519             6.973901 0.582   4.530275
```

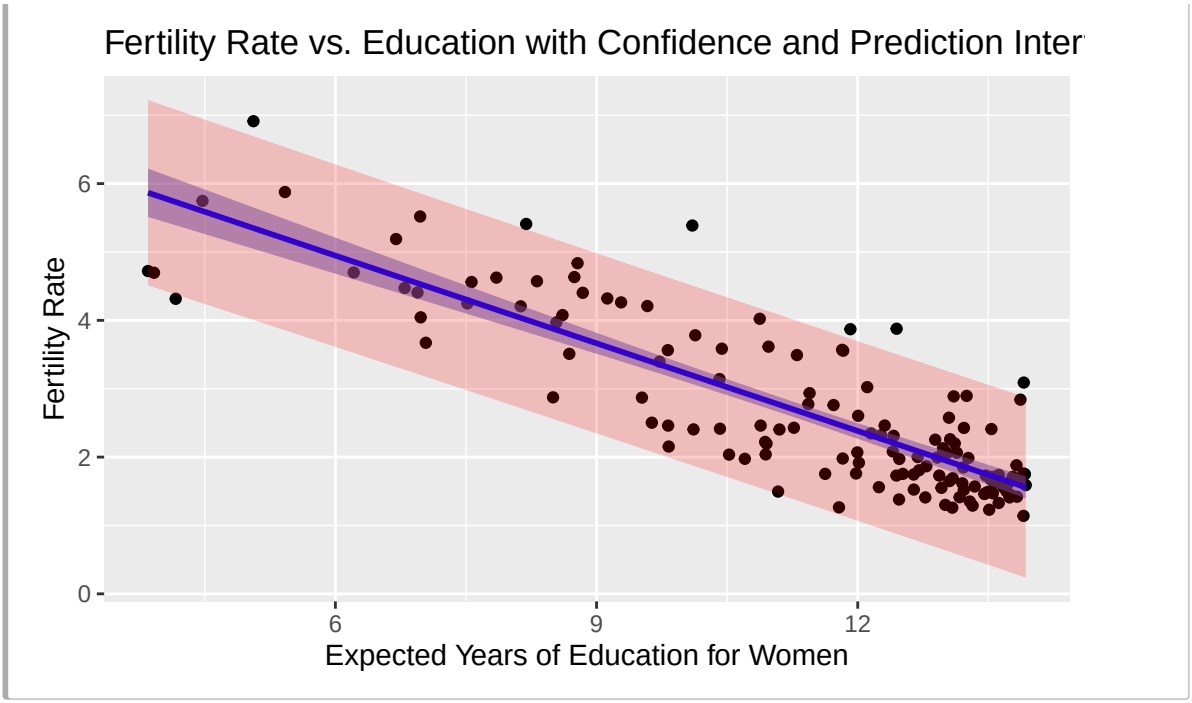
```
5 Antigua and Barbuda           1.994           12.914441 0.772   1.990860
6           Argentina           2.261           13.061253 0.842   1.928102
     resid_slr lwr_mean upr_mean   lwr_obs  upr_obs
1 -0.133441258 4.374528 4.838354 3.2775637 5.935319
2 -0.251041225 1.729965 2.006117 0.5522916 3.183791
3  0.687832282 2.219268 2.451067 1.0215602 3.648775
4  0.988725204 4.305309 4.755240 3.2025920 5.857958
5  0.003140285 1.860002 2.121717 0.6758481 3.305871
6  0.332898098 1.793660 2.062544 0.6127287 3.243475
```

Visualization of Prediction Intervals:

Now, we can create a plot displaying both confidence intervals (for the mean) and prediction intervals (for new observations):

YOUR ANSWER:

```r
# Construct plot showing prediction uncertainty
ggplot(reg_data, aes(x = educ_expected_yrs_f, y = fertility_total)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  geom_ribbon(aes(ymin = lwr_mean, ymax = upr_mean),
              fill = "blue", alpha = 0.2) +  # Confidence interval
  geom_ribbon(aes(ymin = lwr_obs, ymax = upr_obs),
              fill = "red", alpha = 0.2) +    # Prediction interval
  labs(title = "Fertility Rate vs. Education with Confidence and Prediction Intervals",
       x = "Expected Years of Education for Women",
       y = "Fertility Rate")
```



Interpretation

- The **confidence interval** (shaded in blue) represents uncertainty in estimating the mean response.
- The **prediction interval** (shaded in red) is *wider* because it accounts for additional variability when predicting new observations.
- The **prediction band** is interpreted as follows: 95% of the time, the true observed value will fall within this range.

Question 6: coverage

What proportion of observed values are within the prediction bands? Compute and store this value as `coverage_prop`.

YOUR ANSWER:

```r
# Compute the proportion of observed values within prediction bands
coverage_prop <- mean(reg_data$fertility_total >= reg_data$lwr_obs &
                        reg_data$fertility_total <= reg_data$upr_obs)

# Display the computed proportion
coverage_prop
```

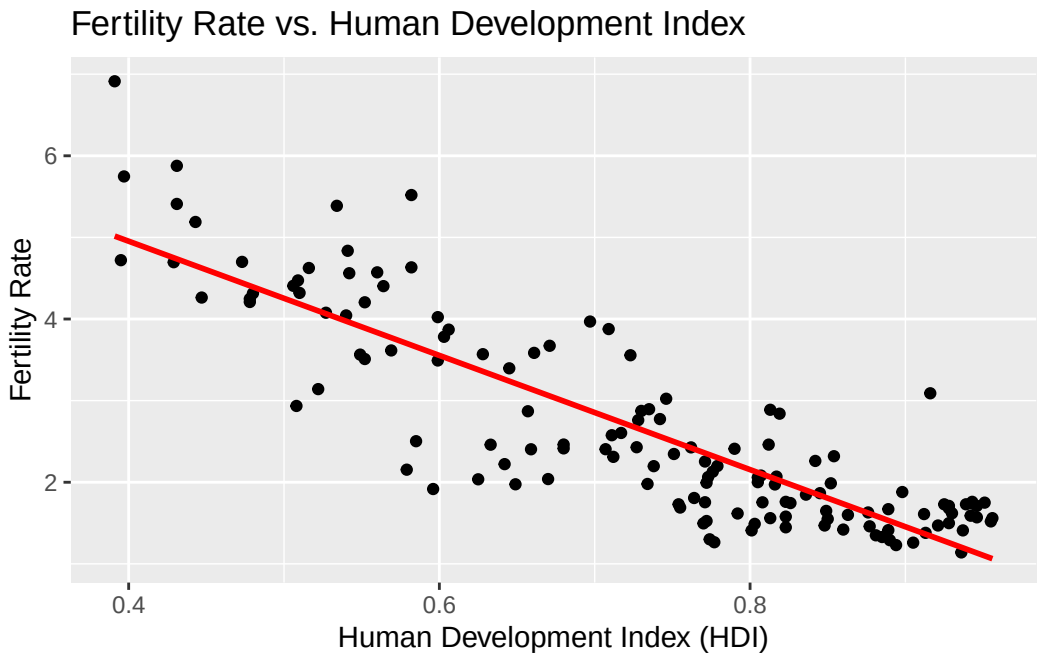
```
[1] 0.9496403
```

Multiple Linear Regression

Now let's consider adding the **human development factor** to the model. First, let's investigate the *univariate* relationship between **HDI** (Human Development Index) and **fertility rate**.

A scatterplot is shown below with a regression line overlaid. The relationship may not be perfectly linear, but a line should provide a decent approximation.

```r
# Scatterplot of HDI vs Fertility Rate with Regression Line
ggplot(reg_data, aes(x = hdi, y = fertility_total)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Fertility Rate vs. Human Development Index",
       x = "Human Development Index (HDI)",
       y = "Fertility Rate")
```



Question 7: Fit a Model with HDI Only

Fit the model plotted above. Display the **coefficient estimates**, **standard errors**, and R^2 **statistic**.

YOUR ANSWER:

```
# Fit simple linear regression with HDI only
lm_hdi <- lm(fertility_total ~ hdi, data = reg_data)

# Display summary of results
summary(lm_hdi)
```

```
Call:
lm(formula = fertility_total ~ hdi, data = reg_data)

Residuals:
     Min       1Q   Median       3Q      Max
-1.66491 -0.34778 -0.05478  0.42486  1.89684

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.7517     0.2596   29.86   <2e-16 ***
hdi          -6.9964     0.3486  -20.07   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6344 on 137 degrees of freedom
Multiple R-squared:  0.7462,    Adjusted R-squared:  0.7443
F-statistic: 402.7 on 1 and 137 DF,  p-value: < 2.2e-16
```

```r
# Coefficients
paste("Coefficient estimates (beta0): ", coef(lm_hdi)[1])
```

```
[1] "Coefficient estimates (beta0):  7.75173493789121"
```

```r
paste("Coefficient estimates (beta1): ", coef(lm_hdi)[2])
```

```
[1] "Coefficient estimates (beta1):  -6.99635435299517"
```

```r
# Variance estimate
paste("Error variance estimate is: ", summary(lm_hdi)$sigma^2)
```

```
[1] "Error variance estimate is:  0.402410034109962"
```

```r
# Variance-covariance matrix
vcov <- vcov(lm_hdi)
paste("Standard errors of estimated beta0 are: ", sqrt(diag(vcov))[1])
```

```
[1] "Standard errors of estimated beta0 are:  0.259597080622218"
```

```r
paste("Standard errors of estimated beta1 are: ", sqrt(diag(vcov))[2])
```

```
[1] "Standard errors of estimated beta1 are:  0.348625462907735"
```

```r
# Compute R-squared
paste("R^2 statistic is: ", summary(lm_hdi)$r.squared)
```

```
[1] "R^2 statistic is:  0.746174354912843"
```

You should have observed that this model also explains about **70% of variance in fertility rates**. This suggests that **HDI** is an **equally good predictor of fertility rates**.

However, HDI is **highly correlated** with women's education. Let's compute their **correlation**:

```
# Compute correlation between HDI and education
cor(reg_data$hdi, reg_data$educ_expected_yrs_f)
```

```
[1] 0.8794943
```

So what do you think will happen if we fit a model with both explanatory variables?

- Will fertility rate have a stronger association with one or the other?
- Will the coefficient estimates also be highly correlated?

Take a moment to consider this and come up with a hypothesis.

Multiple Linear Regression: HDI and Education

The model is fit **exactly** the same way as the **SLR models**—the only difference is that instead of using a **single predictor**, we now use **two predictors (HDI and Education).**

```
# Construct explanatory variable matrix with both predictors
mlr_fit <- lm(fertility_total ~ hdi + educ_expected_yrs_f, data = reg_data)

# Store results
summary(mlr_fit)
```

```
Call:
lm(formula = fertility_total ~ hdi + educ_expected_yrs_f, data = reg_data)

Residuals:
     Min       1Q   Median       3Q      Max
-1.42815 -0.33534 -0.01116  0.35723  1.72439

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         7.96037    0.24494  32.499  < 2e-16 ***
hdi                -4.13262    0.68016  -6.076 1.16e-08 ***
```

```
educ_expected_yrs_f -0.20200    0.04219  -4.787 4.35e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.589 on 136 degrees of freedom
Multiple R-squared:  0.7828,    Adjusted R-squared:  0.7796
F-statistic:   245 on 2 and 136 DF,  p-value: < 2.2e-16
```

Extracting Estimates

```r
# Coefficients
coef(mlr_fit)
```

```
        (Intercept)                 hdi educ_expected_yrs_f
          7.9603708          -4.1326227           -0.2019956
```

```r
# Standard errors
sqrt(diag(vcov(mlr_fit)))
```

```
        (Intercept)                 hdi educ_expected_yrs_f
          0.2449394           0.6801555            0.0421940
```

```r
# Variance estimate
sigma_hat2_mlr <- summary(mlr_fit)$sigma^2
sigma_hat2_mlr
```

```
[1] 0.3469089
```

Coefficient Interpretation

- The association with HDI is **weaker in the multiple linear model** (around -4.13) compared to the simple linear model (-7.00 when education is not included).
- Similarly, the association with education is **also weaker** (around -0.20) compared to the simple model (-0.43 when HDI is not included).

This is due to **multicollinearity**, where HDI and education are **highly correlated**. Let's recall the correlation between them:

```r
# Compute correlation between HDI and education
cor(reg_data$hdi, reg_data$educ_expected_yrs_f)
```

```
[1] 0.8794943
```

Assessing Multicollinearity

```r
# Compute variance-covariance matrix
vcov_mlr <- vcov(mlr_fit)

# Compute correlation between coefficient estimates
stderr_mlr <- sqrt(diag(vcov_mlr))
corr_mx <- diag(1/stderr_mlr) %*% vcov_mlr %*% diag(1/stderr_mlr)

# Display correlation between coefficient estimates
corr_mx[1,2]  # Correlation between HDI and Education coefficient estimates
```

```
[1] -0.3016613
```

Model Fit and R^2 Statistic

The multiple linear regression model captures a little bit more variance than either simple linear regression model individually:

```r
# Compute R-squared
summary(mlr_fit)$r.squared
```

```
[1] 0.7827796
```

Discussion

- The MLR model doesn't add much value in terms of fit, so if that is our only concern we might prefer one of the SLR models.

- However, the presence of additional predictors changes the parameter interpretation – in the MLR model, the coefficients give the estimated changes in mean fertility rate associated with changes in each explanatory variable after accounting for the other explanatory variable. This is one way of understanding why the estimates change so much in the presence of additional explanatory variables – the association between, e.g., HDI and fertility, is different than the association between HDI and fertility after adjusting for women's expected education.
- More broadly, these data are definitely not a representative sample of any particular population of nations – the countries (observational units) are conveniently chosen based on which countries reported data. So there is no scope of inference here, for any of the models we've fit.
- Although we can't claim that, for example, 'the mean fertility rate decreases with education at a rate of 0.2 children per woman per expected year of education after accounting for development status', we can say 'among the countries reporting data, the mean fertility rate decreases with education at a rate of 0.2 children per woman per expected year of education after accounting for development status'. This is a nice example of how a model might be used in a descriptive capacity.

Bootstrap for Estimating Sampling Distribution

The bootstrap method is a **resampling** technique that allows us to estimate the **sampling distribution of a statistic** (such as the **mean**) *without relying on theoretical assumptions*. It is especially useful when the underlying distribution of the data is unknown or difficult to model analytically.

Bootstrap procedure

The **bootstrap procedure** follows these steps:

1. **Resample with replacement** from the observed data, creating a new sample of the same size.
2. **Compute the statistic of interest** (e.g., sample mean) for each resampled dataset.
3. **Repeat the process** many times (e.g., 1000 iterations) to generate an empirical distribution of the statistic.
4. **Analyze the results**, including estimating confidence intervals.

Bootstrap Sampling of the Mean Fertility Rate

We will apply the bootstrap method to estimate the **sampling distribution of the mean fertility rate**.

Step 1: Bootstrap Resampling

We generate **1000 bootstrap samples**, each obtained by randomly resampling (with replacement) from the original dataset.

```
# Load necessary libraries
library(tibble)
library(rsample)
library(ggplot2)
library(purrr)

# Set seed for reproducibility
set.seed(123)

# Create a tibble with fertility rate
bootstrap_data <- tibble(fertility = reg_data$fertility_total) |>
  bootstraps(times = 1000) |>
  mutate(bootstrap_mean = map_dbl(splits, ~ mean(as_tibble(.)$fertility)))

# Display first few bootstrap sample means
head(bootstrap_data$bootstrap_mean)
```

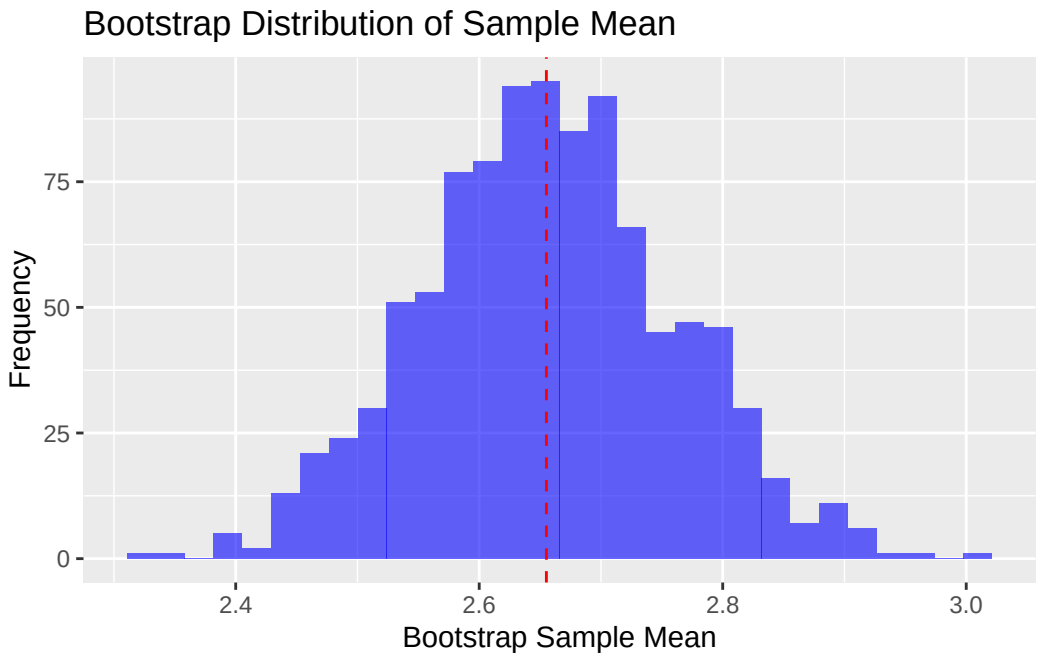
```
[1] 2.820365 2.670422 2.420689 2.699350 2.670547 2.529610
```

Step 2: Visualizing the Bootstrap Distribution

A **histogram** of the **bootstrap sample means** allows us to approximate the sampling distribution.

```
# Plot the bootstrap distribution of sample means
bootstrap_data |>
  ggplot() +
  geom_histogram(aes(x = bootstrap_mean), bins = 30,
                 fill = "blue", alpha = 0.6) +
  geom_vline(aes(xintercept = mean(reg_data$fertility_total)),
             col = "red", linetype = "dashed") +
```

```
  labs(title = "Bootstrap Distribution of Sample Mean",
       x = "Bootstrap Sample Mean",
       y = "Frequency")
```



Interpretation:

- The histogram represents the **empirical distribution** of the **sample mean**.
- The red dashed line represents the **original sample mean**.
- The bootstrap method provides an **approximation of the sampling distribution**, helping us quantify uncertainty in the sample mean.

Bootstrap Confidence Intervals

A key application of bootstrap methods is **constructing confidence intervals** for an estimator. We can estimate a 95% confidence interval for **the mean fertility rate** using the **percentile method**.

Step 3: Computing the 95% Confidence Interval

```
# Compute 95% confidence interval from bootstrap distribution
ci_boot <- quantile(bootstrap_data$bootstrap_mean, probs = c(0.025, 0.975))
ci_boot
```

```
    2.5%    97.5%
2.462500 2.859132
```

Interpretation:

- The confidence interval provides a **plausible** range for the **population mean**.
- Unlike theoretical methods, bootstrap confidence intervals **do not require normality** assumptions.

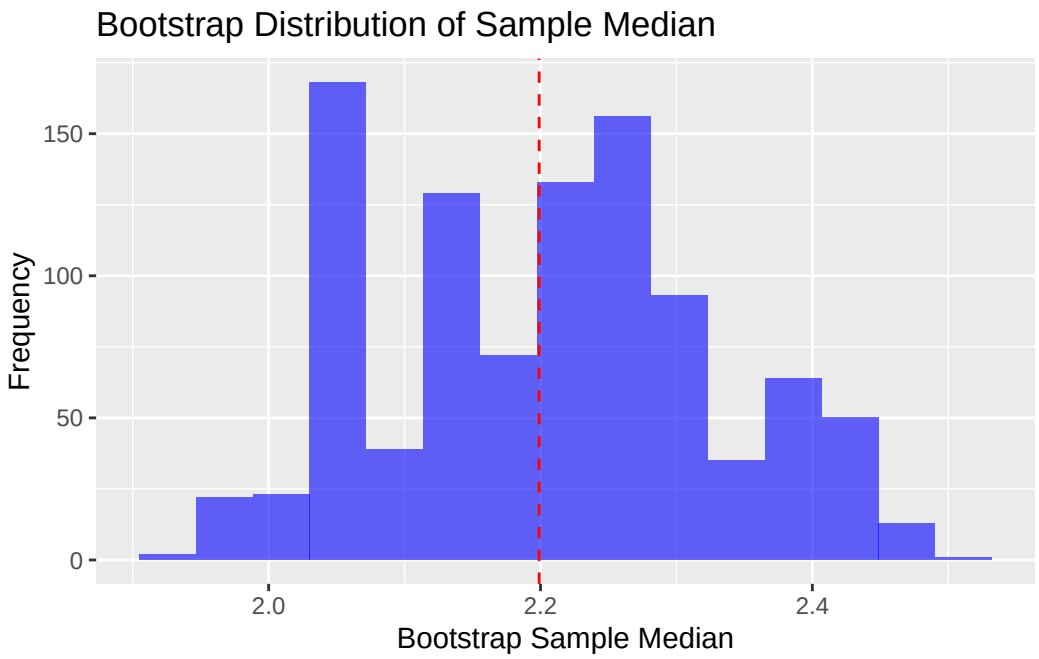
Question 8: Bootstrap Sampling of the Median Fertility Rate

Instead of the mean,

- Estimate the **sampling distribution** of the **median fertility rate** using **1000 bootstrap resamples**.
- Visualize the bootstrap distribution using a **histogram**.
- **Compare** the bootstrap sample medians with the original sample median.

YOUR ANSWER:

```r
# Set seed for reproducibility
set.seed(123)

# Create bootstrap resamples and compute median for each
bootstrap_data_median <- tibble(fertility = reg_data$fertility_total) |>
  bootstraps(times = 1000) |>
  mutate(bootstrap_median = map_dbl(splits, ~ median(as_tibble(.)$fertility)))

# Plot the bootstrap distribution of sample medians
bootstrap_data_median |>
  ggplot() +
  geom_histogram(aes(x = bootstrap_median), bins = 15,
                 fill = "blue", alpha = 0.6) +
  geom_vline(aes(xintercept = median(reg_data$fertility_total)),
             col = "red", linetype = "dashed") +
  labs(title = "Bootstrap Distribution of Sample Median",
       x = "Bootstrap Sample Median",
       y = "Frequency")
```



Question 9: Bootstrap Confidence Interval for HDI Mean

Now, compute a **95% confidence interval** for the **median fertility rate** using the **1000 bootstrap samples** we have drawn.

YOUR ANSWER:

```
# Compute 95% confidence interval for the median
ci_boot_median <- quantile(bootstrap_data_median$bootstrap_median,
                           probs = c(0.025, 0.975))
ci_boot_median
```

```
   2.5%   97.5%
1.99400 2.42705
```

Summary

- Bootstrap resampling allows us to estimate the **sampling distribution of a statistic**.
- The bootstrap confidence interval provides an **empirical way** to quantify estimation uncertainty.
- This method is particularly useful when **theoretical assumptions** about the data, e.g., properties of the underlying distribution, are **uncertain**.

Submission

1. Rename and save the notebook.
2. Restart the kernel and run all cells. (**CAUTION**: if your notebook is not saved, you will lose your work.)
3. Carefully look through your notebook and verify that all computations execute correctly. You should see **no errors**; if there are any errors, make sure to correct them before you submit the notebook.
4. Download the notebook as an `.qmd` file. This is your backup copy.
5. Export the notebook as PDF and upload to Canvas.