

# Practice Session 11

## Part 1 Review of Inference for (Simple) Linear Regression

As we saw last week, hypothesis testing can be done for more than sample means and proportions. We can also test hypotheses relating to regression parameters, like  $\beta_1$ , the slope, and  $\beta_0$ , the y-intercept. The hypotheses typically test whether the parameter is greater than zero, less than zero, or not equal to zero.

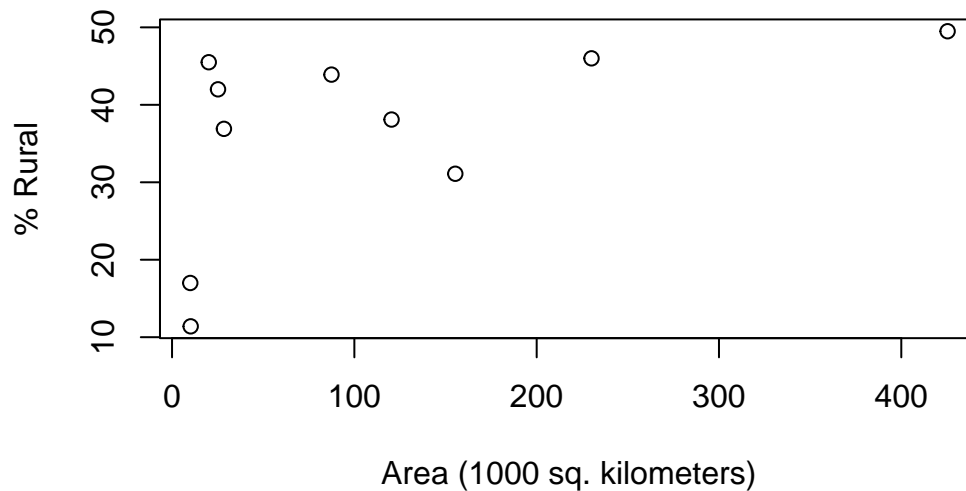
### Question 1

Do larger countries have more rural land? Investigate this question using the `TenCountries` data set from the `Lock5Data` library. Fit a regression model, and run a hypothesis test to see if countries with more area tend to have a higher percentage of rural land. In other words, run a test to see if the slope coefficient is greater than zero. **Assume that the percent of rural land is the outcome for the model, and the area of the country is the predictor.**

a) First, create a scatterplot to visualize the relationship between `Area` and `PctRural`

```
library(Lock5Data)
plot(TenCountries$Area, TenCountries$PctRural,
     xlab = "Area (1000 sq. kilometers)", ylab = "% Rural",
     main = "Rural Land % by Area for Ten Countries")
```

## Rural Land % by Area for Ten Countries



b) State the null and alternative hypotheses using symbols

- $H_0 : \beta_1 = 0$
- $H_A : \beta_1 > 0$

c) Fit the regression model with `PctRural` as the outcome, and `Area` as the predictor. Extract the slope coefficient by using the `coef()` function, and selecting the second value (e.g., `coef(my_model)[2]`).

```
country_model = lm(data = TenCountries, PctRural ~ Area)
obs_country_slope = coef(country_model)[2]
obs_country_slope
```

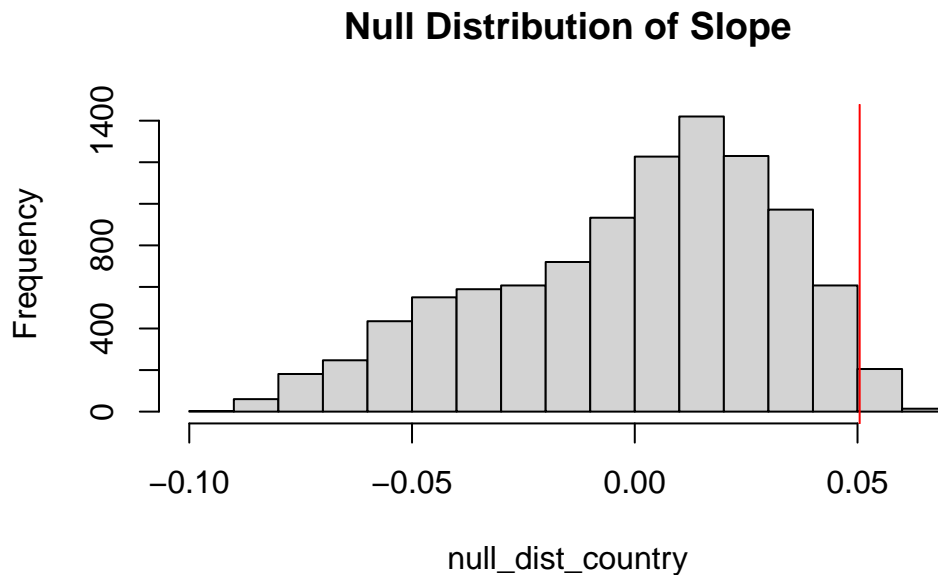
```
Area
0.05048028
```

d) Create a null distribution by using the `do_it` function. The approach you will want to take is to fit a regression model inside the `do_it` call (maybe call it `curr_model`), and you will use `PctRural` as the outcome, and a shuffled `Area` as the predictor. You can shuffle the `Area` variable using the `shuffle()` function. Extract the slope coefficient after fitting each model.

```
library(SDS1000)
null_dist_country = do_it(10000) * {
  curr_mod = lm(data = TenCountries, PctRural ~ shuffle(Area))
  coef(curr_mod)[2]
}
```

- e) Plot a histogram of your null distribution, and add a red vertical line at the observed slope

```
hist(null_dist_country, main = "Null Distribution of Slope")
abline(v = obs_country_slope, col = "red")
```



- f) Calculate the p-value by seeing the proportion of null values that are more extreme than the one you observed.

```
pnull(obs_country_slope, null_dist_country, lower.tail = F)
```

```
[1] 0.0207
```

- g) State your conclusion.

Since our p-value is less than 0.05, we will reject the null hypothesis. We therefore have evidence that countries with more area tend to have a higher percentage of rural land.

- h) Compare your final result to the output from using the `summary()` function on your regression model. What do you notice?

```
summary(country_model)
```

Call:

```
lm(formula = PctRural ~ Area, data = TenCountries)
```

Residuals:

Min	1Q	Median	3Q	Max
-19.639	-6.073	2.681	7.957	13.961

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	30.52265	4.86425	6.275	0.000239 ***
Area	0.05048	0.02895	1.743	0.119410

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.52 on 8 degrees of freedom

Multiple R-squared: 0.2753, Adjusted R-squared: 0.1848

F-statistic: 3.04 on 1 and 8 DF, p-value: 0.1194

- i) A collaborator raises concerns about this analysis. She notes that there is a disagreement between your permutation results, and the output from the `summary()` function. What could be causing this issue? Could there be an issue with the data set that you used?

The sample size here is very small ( $n = 10$ ), which makes fitting a regression model very difficult. The results for this analysis are not very trustworthy, which is why we see a disagreement between the two results. We should recommend that we obtain a larger sample size before running this analysis again.

## Question 2

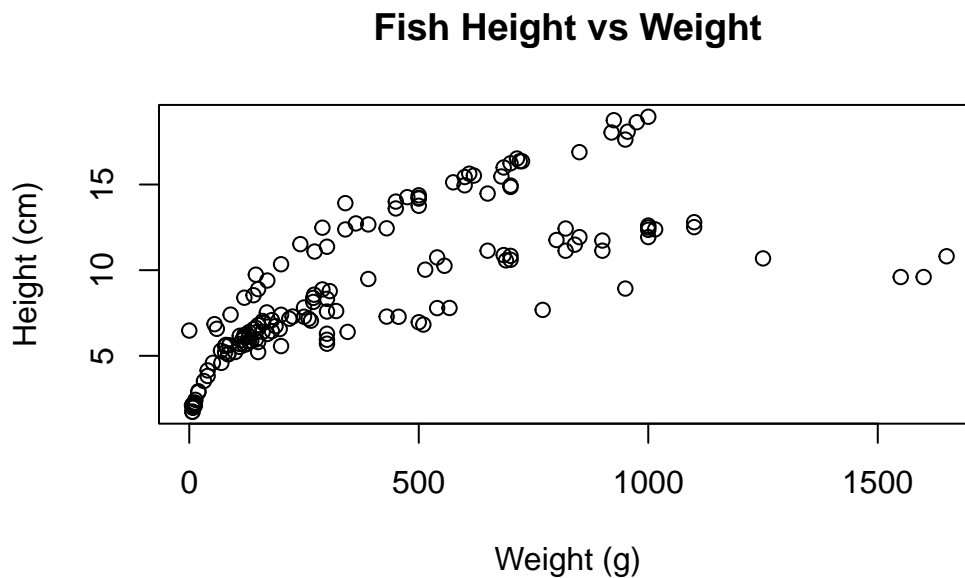
Are heavier fish longer? Data collected from fish markets provides measurements on the weight and height of different species of fish. Fit a regression model and run a hypothesis test to see

if heavier fish tend to also be longer. Assume that fish height is the outcome for the model, and fish weight is the predictor.

The data is available in the `Fish.csv` data set. Use `read.csv()` to load the data.

a) First, create a scatterplot to visualize the relationship between `Height` and `Weight`.

```
fish = read.csv("Fish.csv")
plot(fish$Weight, fish$Height, xlab = "Weight (g)", ylab = "Height (cm)",
     main = "Fish Height vs Weight")
```



b) State the null and alternative hypotheses using symbols

- $H_0 : \beta_1 = 0$
- $H_A : \beta_1 > 0$

c) Fit the regression model with `Height` as the outcome, and `Weight` as the predictor. Extract the slope coefficient by using the `coef()` function, and selecting the second value (e.g., `coef(my_model)[2]`).

```
fish_model = lm(data = fish, Height ~ Weight)
obs_fish_slope = coef(fish_model)[2]
obs_fish_slope
```

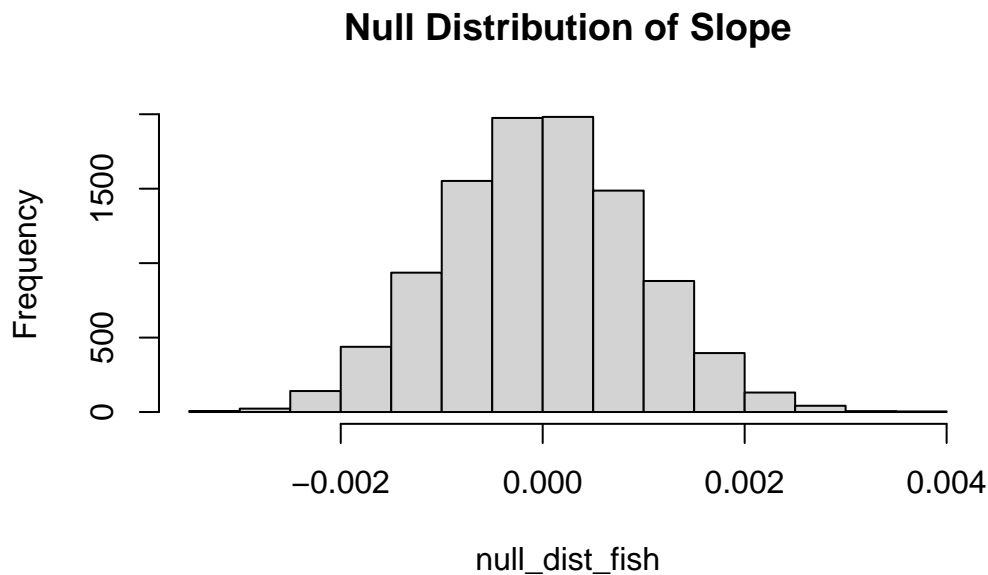
Weight  
0.008672856

- d) Create a null distribution by using the `do_it` function. The approach you will want to take is to fit a regression model inside the `do_it` call (maybe call it `curr_model`), and you will use `Height` as the outcome, and a shuffled `Weight` as the predictor. You can shuffle the `Weight` variable using the `shuffle()` function. Extract the slope coefficient after fitting each model.

```
null_dist_fish = do_it(10000) * {  
  curr_mod = lm(data = fish, Height ~ shuffle(Weight))  
  coef(curr_mod)[2]  
}
```

- e) Plot a histogram of your null distribution, and add a red vertical line at the observed slope

```
hist(null_dist_fish, main = "Null Distribution of Slope")  
abline(v = obs_fish_slope, col = "red")
```



- f) Calculate the p-value by seeing the proportion of null values that are more extreme than the one you observed.

```
pnull(obs_fish_slope, null_dist_fish, lower.tail = F)
```

```
[1] 0
```

g) State your conclusion.

**Since our p-value is less than 0.05, we will reject the null hypothesis. We therefore have evidence that heavier fish tend to be longer.**

h) Compare your final result to the output from using the `summary()` function on your regression model. What do you notice?

```
summary(fish_model)
```

Call:

```
lm(formula = Height ~ Weight, data = fish)
```

Residuals:

Min	1Q	Median	3Q	Max
-9.793	-1.708	-0.533	2.240	5.448

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	5.5163660	0.3523298	15.66	<2e-16 ***
Weight	0.0086729	0.0006588	13.16	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.964 on 157 degrees of freedom

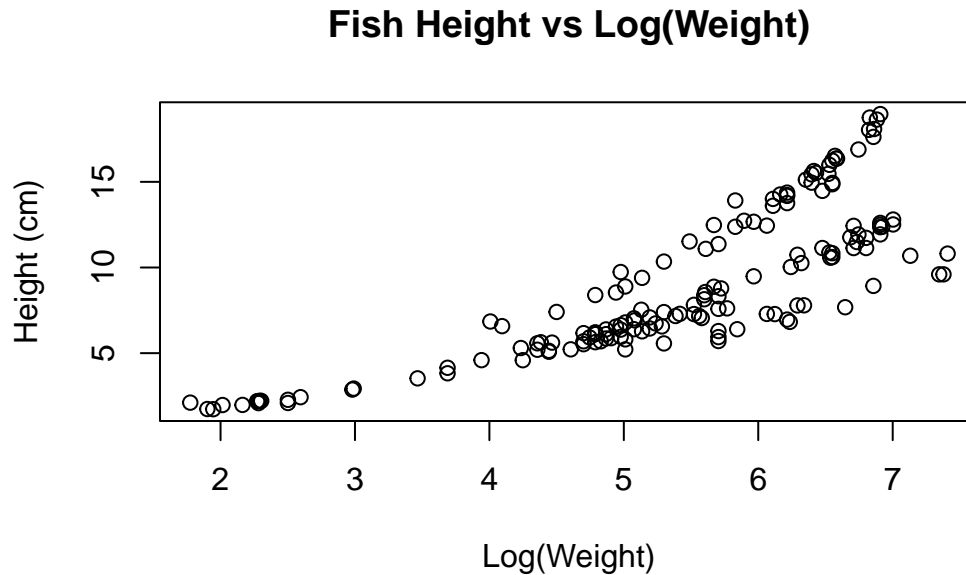
Multiple R-squared: 0.5247, Adjusted R-squared: 0.5216

F-statistic: 173.3 on 1 and 157 DF, p-value: < 2.2e-16

i) The same collaborator raises concerns about the fish analysis. She notes that the scatterplot doesn't appear to be linear. Are your regression results still valid? Can you think of any solutions for fixing the data so that we can run a linear regression?

**No, the regression results are not valid. Linearity is a requirement for running linear regression. One solution is to transform the data by taking the logarithm of Weight (the predictor). This will make the plot appear more linear, and help meet the linearity assumption.**

```
plot(log(fish$Weight), fish$Height, xlab = "Log(Weight)", ylab = "Height (cm)",  
     main = "Fish Height vs Log(Weight)")
```



## Part 2 Inference for (Multiple) Linear Regression

So far, we have fit regression models where we had a single predictor or variable relating to our outcome variable. However, we often are interested in understanding the relationship among the outcome and two or more other variables. Regression with more than one predictor is known as multiple linear regression.

### Question 3

### Question 4