MTH 4230 Lab 1

Due Wed., Feb. 5

1 Part A: Simple Linear Regression

1.1 Murder Rates Data Set

The following are the **murder rates** (per 100,000 people) for $n = 10$ U.S. counties:

15.1, 11.3, 7.8, 10.1, 10.3, 6.8, 3.1, 6.2, 10.7, 13.9

Here are the **illiteracy rates** for the same counties:

2.1, 1.5, 1.8, 1.9, 1.1, 0.7, 1.1, 0.9, 1.3, 2.0

1. Use `c()` to create two vectors called `murder` and `illit` containing the above data.
2. Use `plot()` to make a scatterplot of the **murder rates** (y-axis) versus **illiteracy rates** (x-axis) by typing:

```
plot(x = illit, y = murder, pch = 19,
     xlab = "Illiteracy Rate", ylab = "Murder Rate",
     main = "Plot of Murder vs Illiteracy for Ten U.S. Counties")
```

3. We'll fit the ***linear regression model***

$$Y = \beta_0 + \beta_1 X + \epsilon$$

to the data, with **murder rate** as the response variable and **illiteracy rate** as the predictor, and carry out a t test of the hypotheses

$$H_0 : \beta_1 = 0$$
$$H_a : \beta_1 \neq 0$$

The function `lm()` will carry out the ***linear regression analysis***. Among its arguments are:

`formula`	A formula specifying the regression model

Use `lm()` to fit the regression model and carry out the hypothesis test, saving the results in an object called, say, `my.reg`, for example by typing:

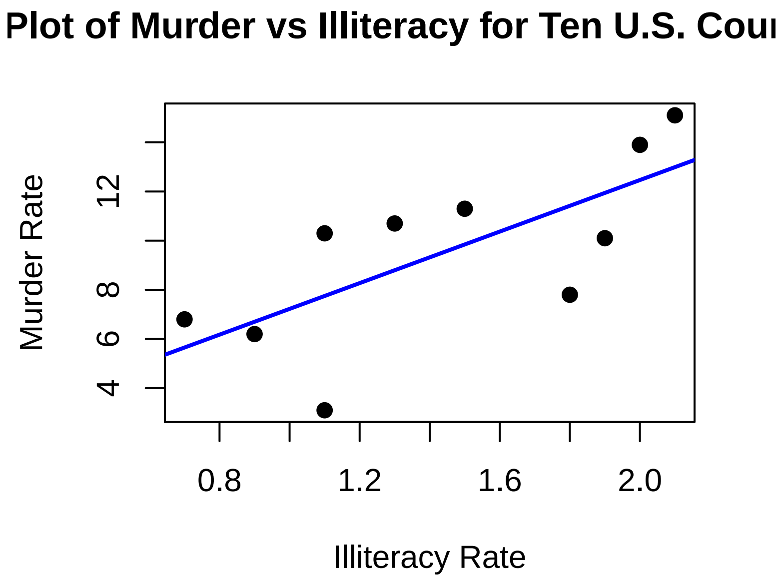
```
my.reg <- lm(murder ~ illit)
```

(Above, the *formula* is `murder ~ illit`.)

4. Recreate the scatterplot of Step 2 again and then **add** the **fitted regression line** to the **plot** by typing:

```
abline(my.reg)
```

Your plot should look something like this:



5. Now use `summary()` to obtain the estimates of the coefficients β_0 and β_1 and to look at the results of the t test for the slope:

```
summary(my.reg)
```

6. The object `my.reg` is a *list* (type `is.list(my.reg)`). Typing:

```
names(my.reg)
```

will show the names of the objects stored in `my.reg`, and the `$` operator can be used to extract specific objects from `my.reg`.

Check the **normality assumption** for the error term ϵ in the regression model by using `hist()` to make a histogram of the residuals (`my.reg$residuals`):

```
hist(my.reg$residuals, col = "blue")
```

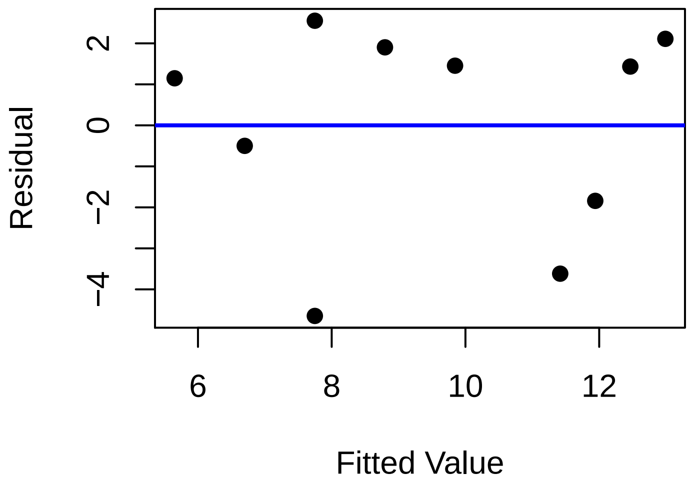
7. Now check the **constant standard deviation assumption** for ϵ by making a plot (use `plot()`) of the residuals (y-axis) versus the fitted values (`my.reg$fitted.values`, x-axis:

```
plot(x = my.reg$fitted.values, y = my.reg$residuals, pch = 19)
```

Add a horizontal line to the plot at $y = 0$ by typing:

```
abline(h = 0)
```

You should end up with something similar to this:



8. Look at the ***regression ANOVA table*** by typing:

```
anova(my.reg)
```

2 Part B: Intercept-Only Regression

2.1 Murder Rates Data Set (Continued)

1. We can fit the ***intercept-only regression model***

$$Y = \beta_0 + \epsilon$$

to the data, with **murder rate** as the response variable (and **no predictor**).

Use `lm()` to fit the model, saving the results as `my.reg`, for example by typing:

```
my.reg <- lm(murder ~ 1)
```

(Above, the `1` in the *formula* `murder ~ 1` represents the intercept β_0.)

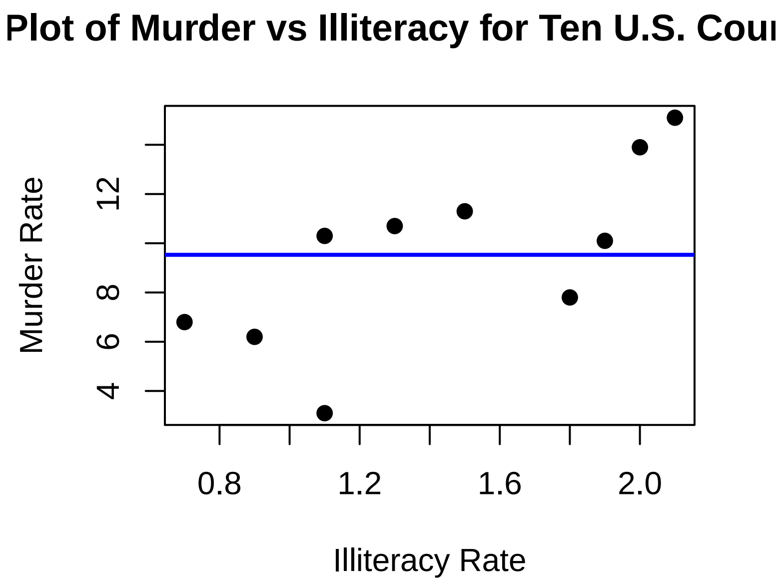
2. Now use `summary()` to obtain the estimate of the coefficient β_0.

```
summary(my.reg)
```

3. Recreate the scatterplot of Step 2 in Part A and then **add** the **fitted regression line** to the **plot** by typing:

```
abline(my.reg)
```

Your plot should look something like this:



4. Compute the **sample mean** $\bar{Y}$ of the **murder rates**:

```
mean(murder)
```

and compare its value to b_0 in the fitted intercept-only model.

5. Carry out a **one-sample *t* test** of the hypotheses:

$$\begin{aligned} H_0 : \mu &= 0 \\ H_a : \mu &\neq 0 \end{aligned}$$

where μ is the **population mean murder rate** (for the population of U.S. counties) by typing:

```
t.test(x = murder, mu = 0)
```

and compare the results (t value and p-value) to those for the t test of

$$\begin{aligned} H_0 : \beta_0 &= 0 \\ H_a : \beta_0 &\neq 0 \end{aligned}$$

in the intercept-only model.

3 Part C: Confidence Interval for the Slope, Confidence Interval for a Mean Response, and Prediction Interval

3.1 Murder Rates Data Set (Continued)

1. Fit the linear regression model again with **murder rate** as the response and **illiteracy rate** as the predictor:

```
my.reg <- lm(murder ~ illit)
```

2. Now we'll compute a ***95% confidence interval for β_1***,

$$b_1 \pm t(0.025, n-2)s\{b_1\}.$$

Use `confint()` to compute the confidence interval:

```
confint(my.reg)
```

3. Now we want to obtain the ***estimate***

$$\hat{Y}_h = b_0 + b_1 X_h$$

of the mean response

$$E(Y_h) = \beta_0 + \beta_1 X_h$$

(also the ***predicted value*** of Y) when $X_h = 1.6$. Use `predict()` to obtain $\hat{Y}_h$:

```
illit.new <- data.frame(illit = 1.6)        # Create a data frame with
                                            # the new illiteracy rate
predict(my.reg, newdata = illit.new)
```

4. We can obtain a ***95% confidence interval for $E(Y_h)$***.

$$\hat{Y}_h \pm t(0.025, n-2)s\{\hat{Y}_h\}.$$

   Use `predict()` as in Step 3 but with the optional argument `interval = "confidence"` to obtain the confidence interval.

5. We can also obtain a ***95% prediction interval*** for a **new response $Y_{h(new)}$**,

$$\hat{Y}_h \pm t(0.025, n-2)s\{\text{pred}\}.$$

   Use `predict()`, as in Step 3 but this time with the optional argument `interval = "prediction"`, to obtain the prediction interval.