Week 8 Quiz Material

When copy and pasting from a code block, or from your local `R` session, be sure to include all available digits for any numeric answer. It would be best to copy and paste values that were returned using printing methods that do not round results. (Notably the direct output from calling `summary()`.) Also, do not modify the default digits option in the code blocks or your local `R` session.

Practice

Exercise 1

```
# starter
```

Consider the model

$$Y = 5 - 2x + \epsilon$$

where

$$\epsilon \sim N(0, \sigma^2 = \frac{|x|}{4}).$$

That is

$$\text{Var}[Y \mid X = x] = \frac{|x|}{4}.$$

Calculate

$$P[Y > 1 \mid X = 3].$$

```
# solution
x = 3
mu_x = 5 - 2 * x
sigma_x = sqrt(abs(x) / 4)
pnorm(1, mean = mu_x, sd = sigma_x, lower.tail = FALSE)
```

```
## [1] 0.01046067
```

- Hint: Both the mean and variance are conditioned on x.

Exercise 2

```
# preamble
gen_data = function(sample_size = 20, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = exp(2 + 3 * x + 0.35 * x ^ 2 + rnorm(n = sample_size, sd = 3))
  data.frame(x = x, y = y)
}

quiz_data = gen_data()
```

```
# starter
quiz_data
```

The above code block has access to a data frame named `quiz_data` with two variables `y` and `x`. Here, we use `y` as the response.

Fit a simple linear regression model to this data. What is the Cook's distance for the observation with the largest leverage?

```
# solution
fit = lm(y ~ x, data = quiz_data)
unname(cooks.distance(fit)[which.max(hatvalues(fit))])
```

```
## [1] 1.966891
```

- Hint: The `which.max()` function may be very useful.

Exercise 3

```
# preamble
gen_data = function(sample_size = 20, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = exp(2 + 3 * x + 0.35 * x ^ 2 + rnorm(n = sample_size, sd = 3))
  data.frame(x = x, y = y)
}

quiz_data = gen_data()
```

```
# starter
quiz_data
```

The above code block has access to a data frame named `quiz_data` with two variables `y` and `x`. Here, we use `y` as the response.

Fit a simple linear regression model to this data. Calculate the p-value of the Shapiro-Wilk test for the normality assumption.

```
# solution
fit = lm(y ~ x, data = quiz_data)
shapiro.test(resid(fit))$p.value
```

```
## [1] 0.004583584
```

- Hint: You may need to first obtain the residuals of the model.

Exercise 4

```
# preamble
gen_data = function(sample_size = 20, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = exp(2 + 3 * x + 0.35 * x ^ 2 + rnorm(n = sample_size, sd = 3))
  data.frame(x = x, y = y)
}

quiz_data = gen_data()
```

```
# starter
quiz_data
```

The above code block has access to a data frame named `quiz_data` with two variables `y` and `x`. Here, we use `y` as the response.

Fit the model

$$\log(y) = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon.$$

Use the Shapiro-Wilk test to asses the normality assumption for this model. Use $\alpha = 0.05$.

```
# solution
fit = lm(log(y) ~ x + I(x ^ 2), data = quiz_data)
shapiro.test(resid(fit))$p.value
```

```
## [1] 0.4021823
```

Select the correct decision and interpretation:

- Fail to Reject H_0. Normality assumption is suspect.
- **Fail to Reject H_0. Normality assumption is *not* suspect.**
- Reject H_0. Normality assumption is suspect.
- Reject H_0. Normality assumption is *not* suspect.

- Hint: The null hypothesis of the test assumes normality.

Exercise 5

```
# preamble
gen_data = function(sample_size = 20, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = exp(2 + 3 * x + 0.35 * x ^ 2 + rnorm(n = sample_size, sd = 3))
  data.frame(x = x, y = y)
}

quiz_data = gen_data()
```

```
# starter
quiz_data
```

The above code block has access to a data frame named `quiz_data` with two variables `y` and `x`. Here, we use `y` as the response.

Fit the model

$$\log(y) = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon.$$

Calculate the residual sum of squares (RSS) in the original units of y. That is, calculate

$$\sum (\hat{y}_i - y_i)^2.$$

Report your answer in billions.

```
# solution
fit = lm(log(y) ~ x + I(x ^ 2), data = quiz_data)
sum((exp(fitted(fit)) - quiz_data$y) ^ 2) / 1000000000
```

```
## [1] 42.27957
```

- Hint: $\hat{y}_i$ are the fitted values after undoing the log transformation.
- Hint: Divide the RSS you obtain by `1000000000`

Graded

Exercise 1

```r
# preamble
gen_data_1 = function(sample_size = 25, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = 2 + 3 * x + rnorm(n = sample_size)
  data.frame(x = x, y = y)
}

gen_data_2 = function(sample_size = 25, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = 2 + 3 * x + rt(n = sample_size, df = 2)
  data.frame(x = x, y = y)
}

data_1 = gen_data_1()
data_2 = gen_data_2()
```

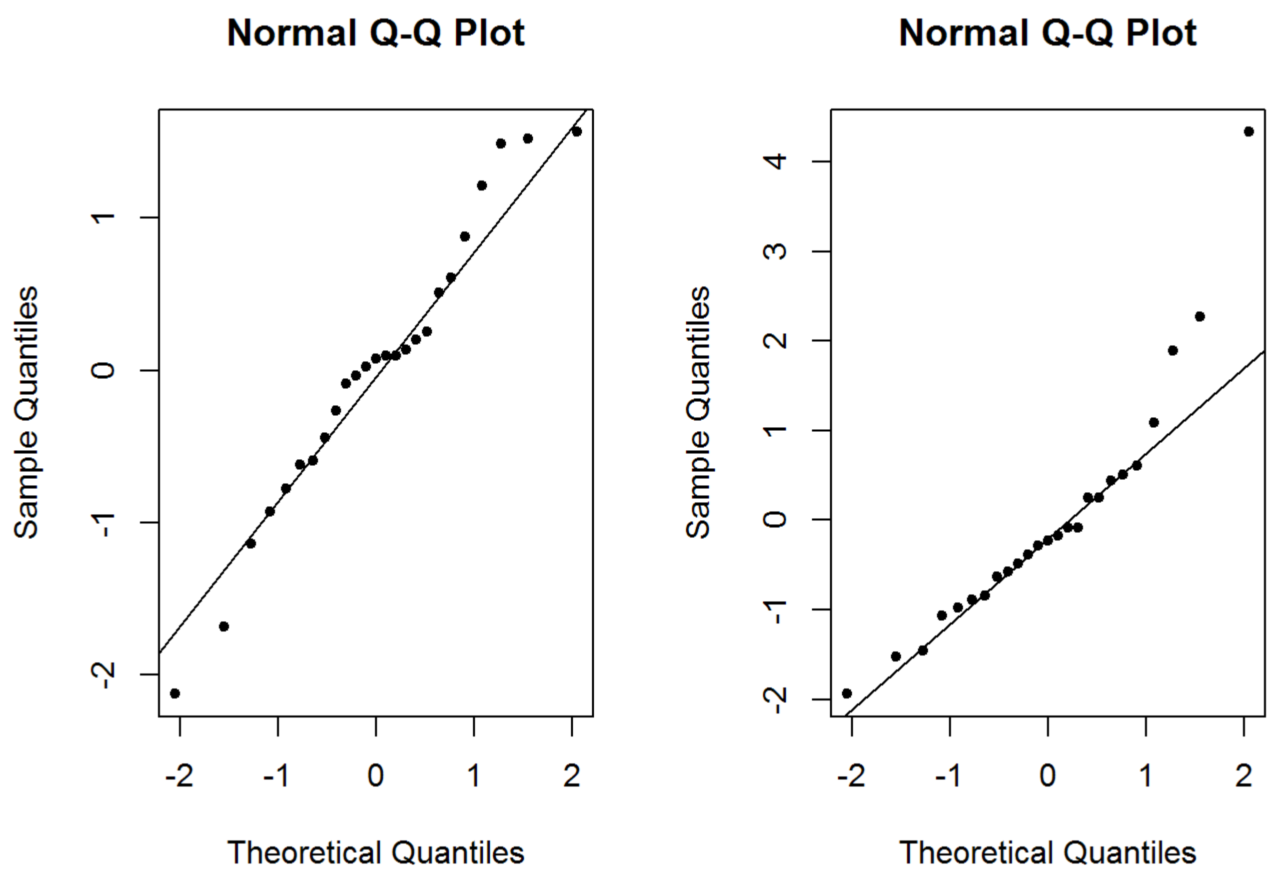
```r
# starter
data_1
data_2
```

The above code block has access to two data frames named `data_1` and `data_2`, both with variables variables `y` and `x`. Here, we use `y` as the response.

Fit a simple linear regression to both datasets. For both fitted regressions, create a Normal Q-Q Plot.

```r
# solution, plot code block
fit_1 = lm(y ~ x, data = data_1)
fit_2 = lm(y ~ x, data = data_2)

par(mfrow = c(1, 2))
qqnorm(resid(fit_1), pch = 20)
qqline(resid(fit_1))
qqnorm(resid(fit_2), pch = 20)
qqline(resid(fit_2))
```



Based on the plots:

- The normality assumption is more suspect for the model fit to `data_1` .
- **The normality assumption is more suspect for the model fit to `data_2` .**

Exercise 2

```
# preamble
gen_data_2 = function(sample_size = 100, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = 2 + 3 * x + rnorm(n = sample_size)
  data.frame(x = x, y = y)
}

gen_data_1 = function(sample_size = 100, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = -3, max = 0)
  y = 2 + 3 * x + sqrt(abs(x * rnorm(n = sample_size)))
  data.frame(x = x, y = y)
}

data_1 = gen_data_1()
data_2 = gen_data_2()
```

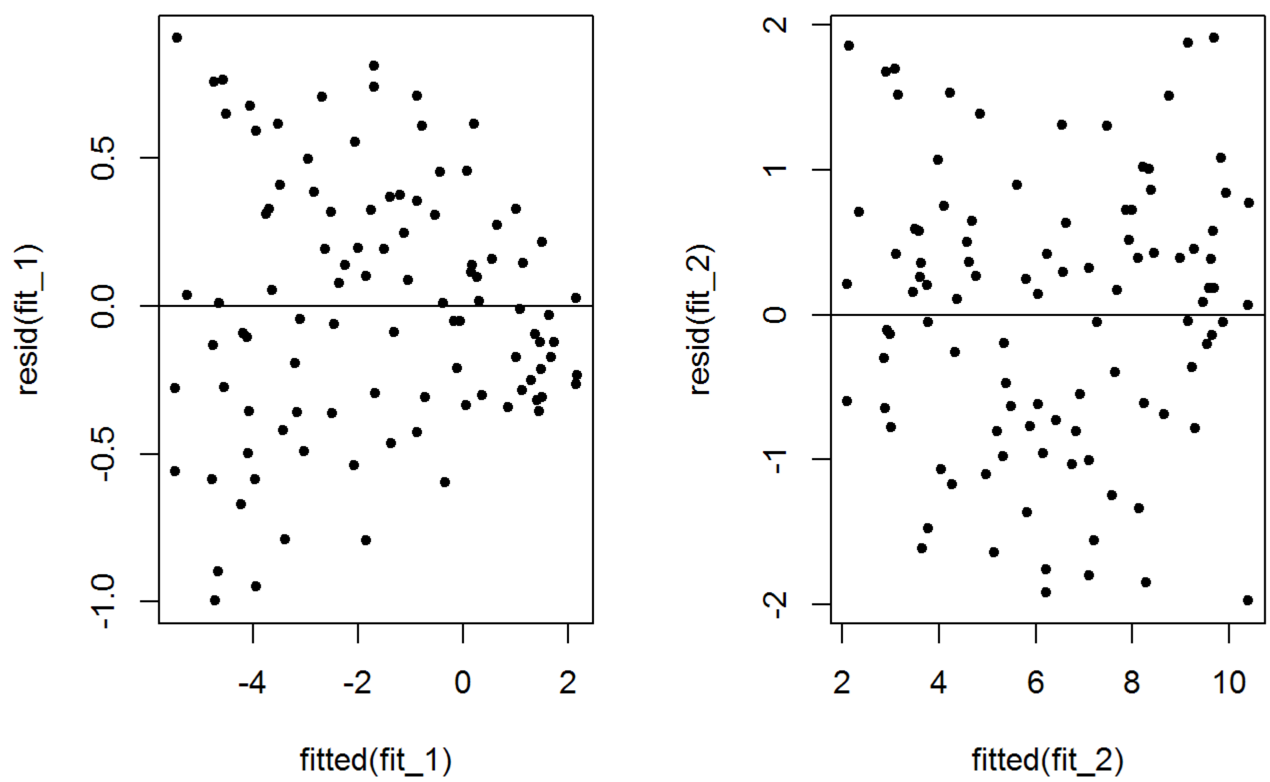
```
# starter
data_1
data_2
```

The above code block has access to two data frames named `data_1` and `data_2` , both with variables variables `y` and `x` . Here, we use `y` as the response.

Fit a simple linear regression to both datasets. For both fitted regressions, create Fitted versus Residuals plot.

```
# solution, plot code block
fit_1 = lm(y ~ x, data = data_1)
fit_2 = lm(y ~ x, data = data_2)

par(mfrow = c(1, 2))
plot(fitted(fit_1), resid(fit_1), pch = 20)
abline(h = 0)
plot(fitted(fit_2), resid(fit_2), pch = 20)
abline(h = 0)
```



Based on the plots:

- **The equal variance assumption is more suspect for the model fit to `data_1` .**
- The equal variance assumption is more suspect for the model fit to `data_2` .

Exercise 3

```
# starter
```

Consider the model

$$Y = 2 + 4x + \epsilon$$

where

$$\epsilon \sim N(0, \sigma^2 = x^2).$$

That is

$$\text{Var}[Y \mid X = x] = x^2.$$

Calculate

$$P[Y < -12 \mid X = -3].$$

```
# solution
x = -3
mu_x = 2 + 4 * x
sigma_x = sqrt(x ^ 2)
pnorm(-12, mean = mu_x, sd = sigma_x)
```

```
## [1] 0.2524925
```

Exercise 4

```
# starter
```

For exercises 4 - 9, use the `LifeCycleSavings` dataset which is built into `R`. Fit a multiple linear regression model with `sr` as the response and the remaining variables as predictors. What proportion of observations have a standardized residual less than 2 in magnitude?

```
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
mean(abs(rstandard(mod)) < 2)
```

```
## [1] 0.96
```

Exercise 5

```
# starter
```

Continue using the model fit in Exercise 4. Note that each observation is about a particular country. Which country (observation) has the standardized residual with the largest magnitude?

```
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
names(which.max(abs(rstandard(mod))))
```

```
## [1] "Zambia"
```

- Acceptable solution inputs: “Zambia”, Zambia, zambia

Exercise 6

```
# starter
```

Continue using the model fit in Exercise 4. How many observations have “high” leverage? Use twice the average leverage as the cutoff for “high.”

```
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
sum(hatvalues(mod) > 2 * mean(hatvalues(mod)))
```

```
## [1] 4
```

Exercise 7

```
# starter
```

Continue using the model fit in Exercise 4. Which country (observation) has the largest leverage?

```
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
names(which.max(hatvalues(mod)))
```

```
## [1] "Libya"
```

- Acceptable solution inputs: “Libya”, Libya, libya

Exercise 8

```
# starter
```

Continue using the model fit in Exercise 4. Report the largest Cook’s Distance for observations in this dataset.

```
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
max(cooks.distance(mod))
```

```
## [1] 0.2680704
```

Exercise 9

```
# starter
```

Continue using the model fit in Exercise 4. Find the observations that are influential. Use $\frac{4}{n}$ as the cutoff for labeling an observation influential.

Create a subset of the original data that excludes these influential observations and refit the same model to this new data. Report the sum of the estimated regression cofficients.

```
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
keep = cooks.distance(mod) < 4 / length(resid(mod))
new_mod = lm(sr ~ ., data = LifeCycleSavings, subset = keep)
sum(coef(new_mod))
```

```
## [1] 19.63769
```

Exercise 10

```
# starter
airquality = na.omit(airquality)
```

For exercises 10 - 15, use the `airquality` dataset which is built into `R`. For simplicity, we will remove any observations with missing data. We will use `Ozone` as the response and `Temp` as a single predictor.

Fit the model

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

Test for the significance of the quadratic term. Report the p-value of this test.

```
# solution
fit_quad = lm(Ozone ~ Temp + I(Temp ^ 2), data = airquality)
summary(fit_quad)$coefficients[3, "Pr(>|t|)"]
```

```
## [1] 0.0004941148
```

Exercise 11

```
# starter
airquality = na.omit(airquality)
```

Fit the model

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \epsilon$$

Test to compare this model to the model fit in Exercise 10. Report the p-value of this test.

```
# solution
fit_quad = lm(Ozone ~ Temp + I(Temp ^ 2), data = airquality)
fit_quar = lm(Ozone ~ Temp + I(Temp ^ 2) + I(Temp ^ 3) + I(Temp ^ 4), data = airquality)
anova(fit_quad, fit_quar)[2, "Pr(>F)"]
```

```
## [1] 0.02436082
```

Exercise 12

```
# starter
airquality = na.omit(airquality)
```

Use the Shapiro-Wilk test to asses the normality assumption for the model in Exercise 11. Use $\alpha = 0.01$.

```
# solution
fit_quar = lm(Ozone ~ Temp + I(Temp ^ 2) + I(Temp ^ 3) + I(Temp ^ 4), data = airquality)
shapiro.test(resid(fit_quar))$p.value
```

```
## [1] 5.00861e-11
```

Select the correct decision and interpretation:

- Fail to Reject H_0. Normality assumption is suspect.
- Fail to Reject H_0. Normality assumption is *not* suspect.
- **Reject H_0. Normality assumption is suspect.**
- Reject H_0. Normality assumption is *not* suspect.

Exercise 13

```
# starter
airquality = na.omit(airquality)
```

Fit the model

$$\log(y) = \beta_0 + \beta_1 x + \epsilon.$$

Use the Shapiro-Wilk test to asses the normality assumption for this model. Use $\alpha = 0.01$.

```
# solution
fit_log = lm(log(Ozone) ~ Temp, data = airquality)
shapiro.test(resid(fit_log))$p.value
```

```
## [1] 0.04867205
```

Select the correct decision and interpretation:

- Fail to Reject H_0. Normality assumption is suspect.
- **Fail to Reject H_0. Normality assumption is *not* suspect.**
- Reject H_0. Normality assumption is suspect.
- Reject H_0. Normality assumption is *not* suspect.

Exercise 14

```
# starter
airquality = na.omit(airquality)
```

Use the model from Exercise 13 to create a 90% prediction interval for `Ozone` when the temperate is 84 degree Fahrenheit. Report the upper bound of this interval

```
# solution
fit_log = lm(log(Ozone) ~ Temp, data = airquality)
exp(predict(fit_log, newdata = data.frame(Temp = 84), interval = "prediction", level = 0.90)[,
"upr"])
```

```
## [1] 122.1224
```

Exercise 15

```
# starter
airquality = na.omit(airquality)
```

Using the model from Exercise 13, calculate the ratio of:

- The sample variance of residuals for obersations with a fitted value less than 3.5
- The sample variance of residuals for obersations with a fitted value greater than 3.5

(While not a formal test for the equal variance assumption, we would hope that this value is close to 1.)

```
# solution
fit_log = lm(log(Ozone) ~ Temp, data = airquality)
var(resid(fit_log)[fitted(fit_log) < 3.5]) / var(resid(fit_log)[fitted(fit_log) > 3.5])
```

```
## [1] 1.353182
```