# Assignment 02

Simulating from the Regression Model

This assignment is worth 16 points.

## Simulation 1: Modeling Heteroskedasticity

**1. Create the fixed $X$ values you will use in each trial of the simulation. To do this, draw $n = 200$ $X$-values from a uniform distribution with a minimum of $-2$ and a maximum of $+2$. Prior to drawing these values, set your starting seed to 678910. Report the syntax you used, and the first six $X$ values.**

```
set.seed(678910)
x = runif(200, min = -2, max = 2)
head(x)
```

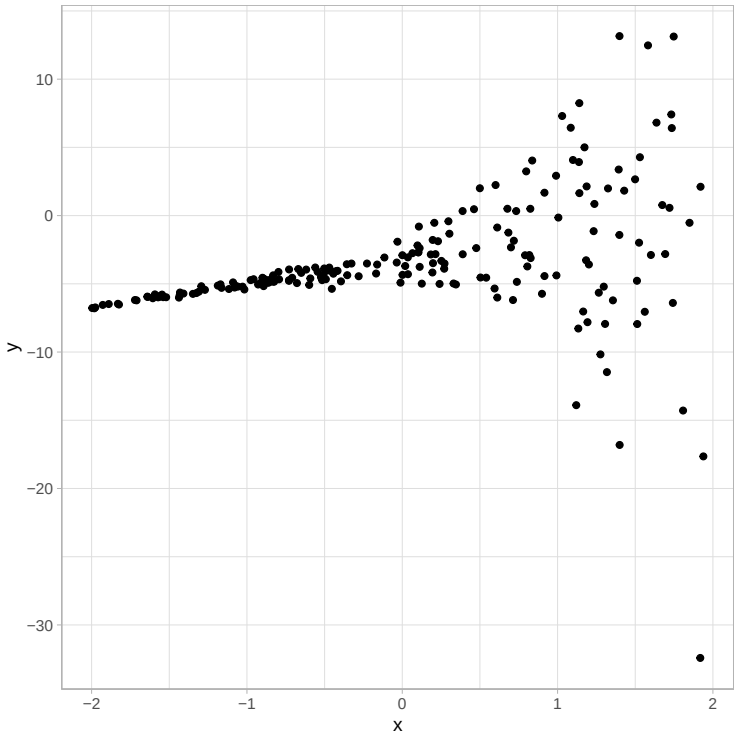
```
[1]  0.19430183  1.26529340  0.03661028 -1.55748375 -0.46987453 -1.63929187
```

**2. Create a the $Y$-values for the first trial of the simulation by using the model: $y_i = -3.2 + 1.75(x_i) + \epsilon_i$ where $\epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma)$ and $\sigma = e^{\gamma(x_i)}$ where $\gamma = 1.5$. Report the syntax you used, and the first six $Y$ values.**

```
y = -3.2 + 1.75*x + rnorm(200, mean = 0, sd = exp(1.5*x))
head(y)
```

```
[1] -4.169444 -5.651138 -4.295246 -5.917156 -3.809642 -5.960746
```

**3. Create and report the scatterplot of the $Y$-values versus the $X$-values for this first trial of the simulation.**




**4. Describe the pattern of heteroscedasticity.**

The variation in **y** is smaller at lower **x**-values and increases at higher **x**-values.

**5. Does the pattern of heteroscedasticity you described in Question 4 make sense given how the error term was created. Explain.**

Yes. The variation in the error term was computed as $e^{1.5x}$, so this value is larger at larger values of **x** and smaller for smaller values of **x**.

*Carry out 1000 trials of the simulation. (Reminder: Be sure to use these same X values in each trial of the simulation; they are fixed.) For each trial, collect: (a) the estimate of the intercept, (b) the estimate of the slope, and (c) the estimate of the residual standard error. Compute and report the mean value for the residual standard error.*

```
set.seed(678910)
x = runif(200, min = -2, max = 2)

# Set parameters
beta_0 = -3.2
beta_1 = 1.75
my_sigma = exp(1.5*x)
n = 200
```

```r
trials = 1000

# Set up empty list to store simulation results
my_estimates = vector(mode = "list", length = trials)

# Repeat the following steps in the simulation
for(i in 1:trials){
  # 1. Simulate the y values
  y = beta_0 + beta_1 * x + rnorm(n, mean = 0, sd = my_sigma)

  # 2. Fit regression model to simulated y values and X
  fitted_model = lm(y ~ 1 + x)

  # 3. Store estimated values in my_estimates matrix (in the first and second element of the ith list s
  my_estimates[[i]][1] = coef(fitted_model)[[1]] #Extract intercept estimate
  my_estimates[[i]][2] = coef(fitted_model)[[2]] #Extract slope estimate
  my_estimates[[i]][3] = summary(fitted_model)$sigma #Extract residual standard error
}

# Convert the list to a data frame for easier computing
hetero = data.frame(do.call(rbind, my_estimates))
names(hetero) = c("b_0", "b_1", "rse") # name columns
```

**6. Compute and report the mean value for the residual standard error.**

- The mean value of the 1000 residual standard error estimates is 5.027.
- The standard deviation value of the 1000 residual standard error estimates is 0.603.

## Simulation 2: Homoskedastic Model

**7. Carry out 1000 trials of the simulation for the appropriate homoskedastic model. (Reminder: Be sure to use these same $X$ values in this simulation as in the previous simulation.) For each trial, collect: (a) the estimate of the intercept, (b) the estimate of the slope, and (c) the estimate of the residual standard error. Report your syntax.**

```r
set.seed(678910)
x = runif(200, min = -2, max = 2)

# Set parameters
beta_0 = -3.2
beta_1 = 1.75
my_sigma2 = mean(hetero$rse)
n = 200
trials = 1000

# Set up empty list to store simulation results
my_estimates2 = vector(mode = "list", length = trials)

# Repeat the following steps in the simulation
for(i in 1:trials){
  # 1. Simulate the y values
```

```r
  y = beta_0 + beta_1 * x + rnorm(n, mean = 0, sd = my_sigma2)

  # 2. Fit regression model to simulated y values and X
  fitted_model = lm(y ~ 1 + x)

  # 3. Store estimated values in my_estimates matrix (in the first and second element of the ith list s
  my_estimates2[[i]][1] = coef(fitted_model)[[1]] #Extract intercept estimate
  my_estimates2[[i]][2] = coef(fitted_model)[[2]] #Extract slope estimate
  my_estimates2[[i]][3] = summary(fitted_model)$sigma #Extract residual standard error
}

# Convert the list to a data frame for easier computing
homo = data.frame(do.call(rbind, my_estimates2))
names(homo) = c("b_0", "b_1", "rse") # name columns
```

## Comparing Results from the Two Simulations: Evaluating the Effects of Hetroskedasticity

**8. Create a density plot of the distribution of intercept estimates. Show the density curve for both models on the same plot, differentiating the curves using color, linetype, or both. Also add a vertical line to this plot at the population value of the intercept.**

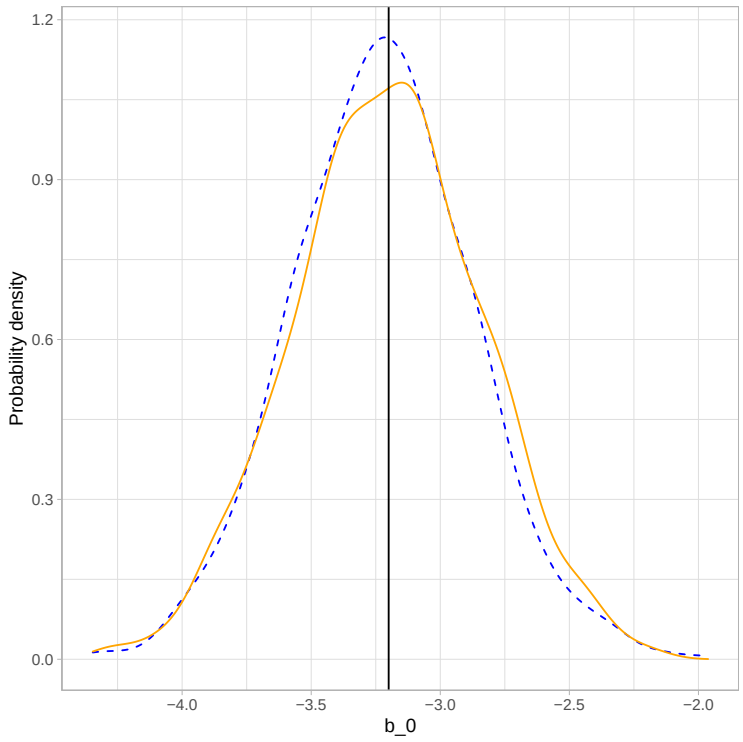



Figure 1: Density plots for the intercept estimates simulated from the heteroskedastic (blue, dashed line) and the homoskedastic (orange, solid line) models. A vertical line is displayed at the population value of -3.2.

**9. Based on your responses to Question 8, does the intercept estimate seem to be biased under heteroskedasticity? Explain.**

The mean value in the heteroskedastic distribution looks a little biased (smaller than the population intercept value of $-3.2$). However, it is fairly similar, so the bias is not great.

**10. Based on your responses to Question 8, does the intercept estimate seem to be less efficient under heteroskedasticity? Explain.**

The heteroskedastic distribution seems to have a similar standard deviation to the homoskedastic distribution. Thus it does not seem less efficient.

**11. Create a density plot of the distribution of slope estimates. Show the density curve for both models on the same plot, differentiating the curves using color, linetype, or both. Also add a vertical line to this plot at the population value of the slope.**

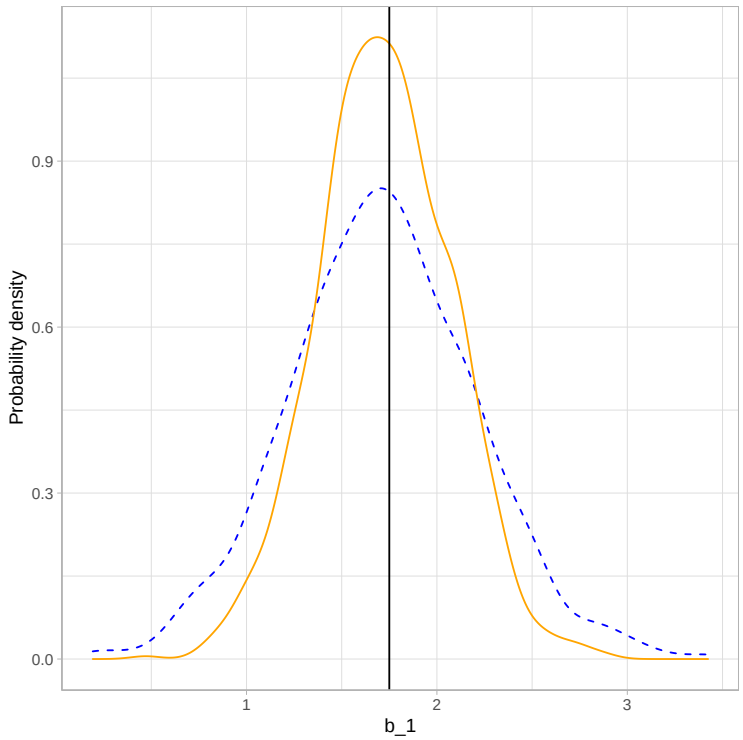



Figure 2: Density plots for the slope estimates simulated from the heteroskedastic (blue, dashed line) and the homoskedastic (orange, solid line) models. A vertical line is displayed at the population value of -3.2.

**12. Based on your responses to Question 11, does the slope estimate seem to be biased under heteroskedasticity? Explain.**

The mean of the heteroskedatic distribution seems close to the population slope value of 1.75. As such, the estimate does not seem biased.

**13. Based on your responses to Question 11, does the slope estimate seem to be less efficient under heteroskedasticity? Explain.**

The heteroskedastic distribution seems to have a higher standard deviation than the homoskedastic distribution. Thus the estimates under heteroskeaticity seem less efficient than those assuming homoskedasticity.

**14. Create a density plot of the distribution of residual standard error estimates. Show the density curve for both models on the same plot, differentiating the curves using color, linetype, or both. Also add a vertical line to this plot at the population value of the residual standard error.**

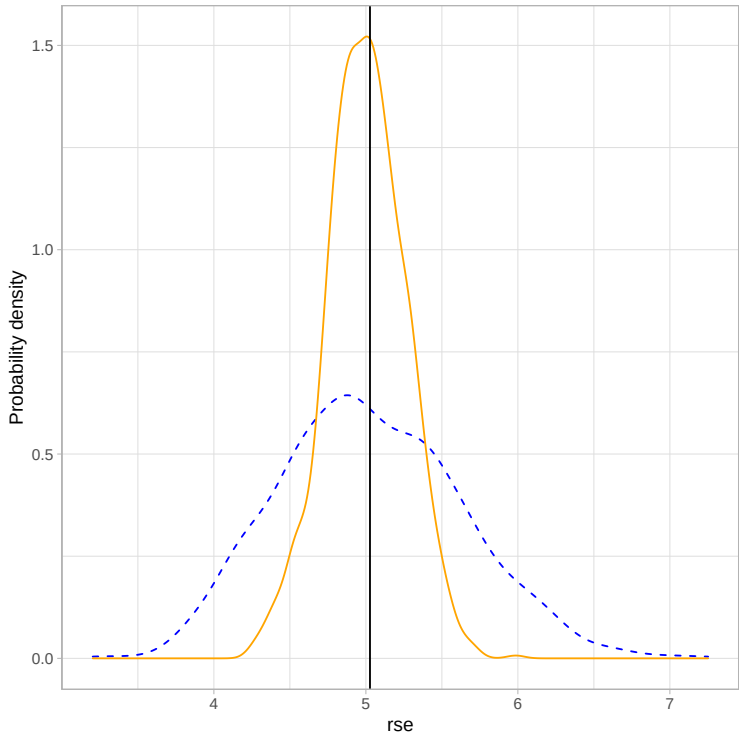



Figure 3: Density plots for the residual standard error estimates simulated from the heteroskedastic (blue, dashed line) and the homoskedastic (orange, solid line) models. A vertical line is displayed at the population value of -3.2.

**15. Based on your responses to Question 14, does the residual standard error estimate seem to be biased under heteroskedasticity? Explain.**

The mean of the heteroskedatic distribution seems smaller than the population value of 5.0269356. As such, the estimate seems negatively biased.

**16. Based on your responses to Question 14, does the residual standard error estimate seem to be less efficient under heteroskedasticity? Explain.**

The heteroskedastic distribution seems to have a higher standard deviation than the homoskedastic distribution. Thus the estimates under heteroskeaticity seem less efficient than those assuming homoskedasticity.