ECO 530 - Exercise 1

Fall 2023 - Devan Arnold

Q1

Consider the function:

$$y = f(x) = 8 \cdot x - 3$$

a. Evaluate $f(x)$ at $x = 3$

To evaluate the above expression, I will code the function as an expression and store the value of the expression in that variable y. Setting $x = 3$ yields:

```
y <- expression(8*x-3)
x <- 3
eval(y)
```

```
## [1] 21
```

Therefore, $f(x = 3) = 21$

b. What is the slope of the function?

Utilizing the above expression $y = 8x - 3$ we will apply the R derivative function D() to determine the slope of the function

```
y_prime <- D(y,'x')
eval(y_prime)
```

```
## [1] 8
```

As shown, this yields a constant value of 8

c. How does the slope of the function at $x = 3$ compare to the slope of the function at $x = 6$

Since the slope of $y = 8x - 3$ is a constant value of 8, the value of x will not influence the slope of y. To prove this, we will run the below code:

```
x <- 3
eval(y)
```

```
## [1] 21
```

```
eval(y_prime)
```

```
## [1] 8
```

```
x <- 6
eval(y)
```

```
## [1] 45
```

```
eval(y_prime)
```

```
## [1] 8
```

As we can see, y' is not affected by the value of x, as implied by the claim of constant slope.

Q2

Consider the function:

$$y = f(x) = 3 - x + 2 \cdot x^2$$

a. Evaluate $f(x)$ at $x = 3$

   Similar to Question 1 part (a), first we will define y as an expression in terms of x

```
y <- expression(3 - x + (2*(x^2)))
```

   Then, we will set $x = 3$ and then evaulate $y(x)$

```
x <- 3

eval(y)
```

```
## [1] 18
```

   Resulting in $y(x = 3) = 18$

b. What is the slope of the function?

To determine the slope of the function $y(x) = f(x)$ we will use the D() function in R to set the value of the expression y', which we will use the variable `y_prime` to represent:

```
y_prime <- D(y,'x')
print(y_prime)
```

```
## 2 * (2 * x) - 1
```

Unlike in Question 1, part (b) the slope of $y = 3 - x + 2 \cdot x^2$ is NOT a constant value, and is thus dependent on the value of x pursuant to the equation $y' = -1 + 4x$.

c. How does the slope of the function at $x = 3$ compare to the slope of the function at $x = 6$

Since $y'(x)$ is not a constant slope, the value of $y'(x = 3) = / = y'(x = 6)$. We can demonstrate this using the code below:

```
x <- 3
eval(y_prime)
```

```
## [1] 11
```

```
x <- 6
eval(y_prime)
```

```
## [1] 23
```

As this code shows, $y'(3) = 11$ and $y'(6) = 23$. This is consistant with the y' function derived in part (a).

d. Does this function have a maximum or a minimum? What is it?

To determine if the function y has a minimum or maximum we will determine if there exists a root for y' such that $y'(x) = 0$. If this value does exist, then we will move forward to determine the nature of this point.

```
y_prime_parameters <- c(-1,4)
y_roots <- polyroot(y_prime_parameters)
print(y_roots)
```

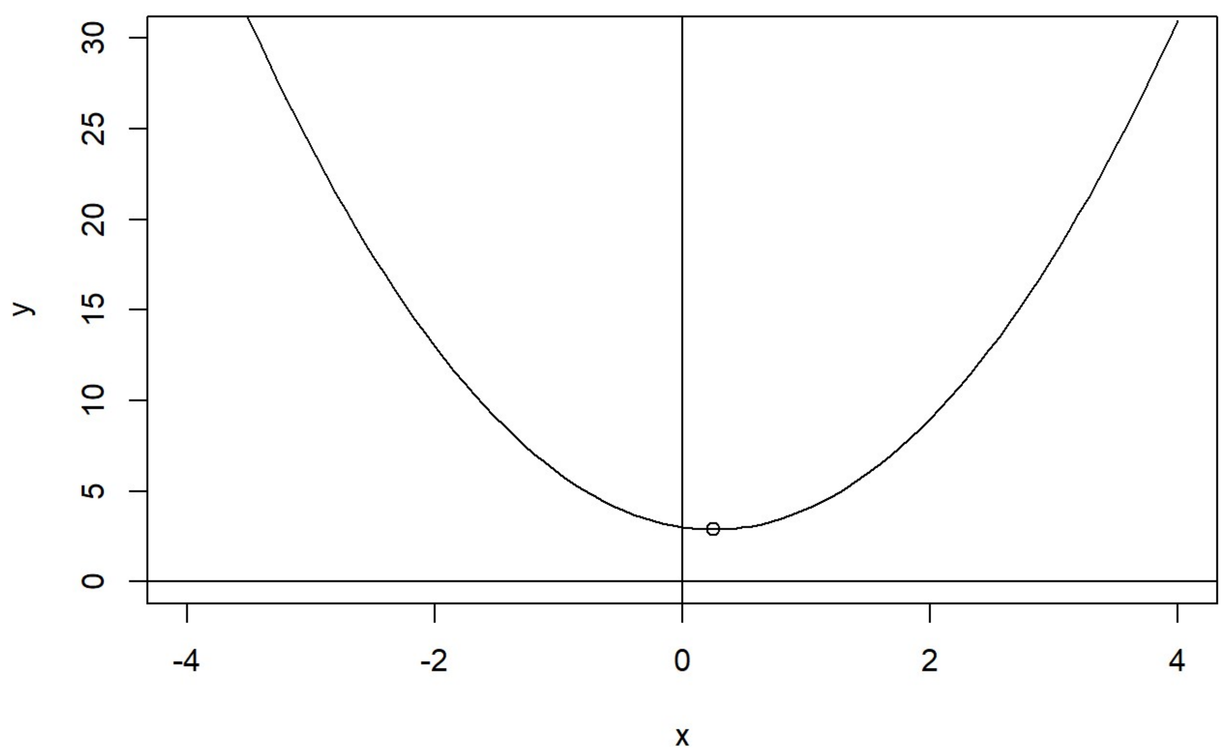
```
## [1] 0.25+0i
```

So from the above code, we can say that there does exist a root for y' at $x = 0.25$. Next, we need to determine if this value represents a minimum or maximum of the function. to do this, we will determine the value of y' at x values greater than and less than the root value of $x = 0.25$.

```
x <- 0.24
y_prime_1 <- eval(y_prime)

x <- 0.26
y_prime_2 <- eval(y_prime)

if (y_prime_1 > y_prime_2){
  inflection <- "Maxima"
} else if (y_prime_2 > y_prime_1){
  inflection <- "Minima"
}

results <- "The point x=0.25 represents a"
results <- paste(results, inflection)

print(results)
```

```
## [1] "The point x=0.25 represents a Minima"
```

As we can see, the extrema that occurs at $x = 0.25$ represents a minimum of the function $y = 3 - x + 2 \cdot x^2$. We can visulize this with the following plot:



Q3

Consider the function:

$$y = f(x, z) = 100 + 3 \cdot x^2 + 2 \cdot z - 5 \cdot x \cdot z$$

a. Define and derive the two elements below:

$$\frac{\partial y}{\partial x} \qquad \frac{\partial y}{\partial z}$$

The first element is the partial derivative of y with respect to x. We can evaluate this by treating the variable z as a constant and performing derivation on y with respect to x. This evaluates as follows:

```
y <- expression(100+3*(x^2)+2*(z)-5*(x*z))
partial_yx <- D(y,'x')
print(partial_yx)
```

```
## 3 * (2 * x) - 5 * z
```

The second element is the partial derivative of y with respect to z. As above, this represents the derivative of the y function with respect to z while holding the other variable, x, constant. This results in:

```
partial_yz <- D(y,'z')
print(partial_yz)
```

```
## 2 - 5 * x
```

Thus, $\frac{\partial y}{\partial x} = 6x - 5z$ and $\frac{\partial y}{\partial z} = 2 - 5x$

b. Define and derive the two elements below:

$$\frac{\partial^2 y}{\partial x^2} \qquad \frac{\partial^2 y}{\partial z \partial x}$$

The element $\frac{\partial^2 y}{\partial x^2}$ represents the second order partial derivative of y with respect to x. We can determine this expression by taking the partial derivative of $\frac{\partial y}{\partial x}$ with respect to x. This evaluates as:

```
partial_yxx <- D(partial_yx,'x')
print(partial_yxx)
```

```
## 3 * 2
```

Thus $\frac{\partial^2 y}{\partial x^2} = 6$, since we again hold z values constant and treat them as such in our derivation.

The second element, $\frac{\partial^2 y}{\partial z \partial x}$ represents the second order partial derivative of y, but this time with respect to z then x. In order to evaluate this, we can take our result from $\frac{\partial y}{\partial z}$ in part (a) and then perform partial derivation again but with respect to x. This evaluates as:

```
partial_yzx <- D(partial_yz,'x')
print(partial_yzx)
```

```
## -5
```

Thus yielding $\frac{\partial^2 y}{\partial z \partial x} = -5$ for similar reasons as above.

Q4

Evaluate the following expressions. Show your work where necessary.

a. $\sum_{y=1}^{10} y$

A summation function from $y = 1$ to $y = 10$ of y. This evaluates to:

```
y <- function(lower,upper){
  output <- 0
  for(i in lower:upper){
    output <- output + i
  }
  return(output)
}

y(1,10)
```

```
## [1] 55
```

Thus, $\sum_{y=1}^{10} y = 55$

b. $\sum_{i=1}^{10} 5$ (or, more generally, any constant c)

A summation function from $i = 1$ to $i = 10$ of 5. This evaluates to:

```
con <- function(const,lower,upper){
  output <- 0
  for(i in lower:upper){
    output <- output + const
  }
  return(output)
}

con(5,1,10)
```

```
## [1] 50
```

This implies that $\sum_{i=1}^{10} 5 = 50$. More broadly, any constant value that has the summation operation performed over it is equal to that constant times the number of occurrences in the summation. Another way to say that is the range over which the constant is summed times the constant is the result of this operation.

```
con(10,9,10)
```

```
## [1] 20
```

```
con(3,5,10)
```

```
## [1] 18
```

```
con(2,1,2)
```

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

c. $\frac{df}{dx}$ where $f(x) = (12x+3)(6x^2+8x-x^3)$

As seen in Questions 1 & 2, derivation of functions can be performed by R, resulting in:

```
f <- expression((12*x+3)*(6*(x^2)+8*(x)-(x^3)))
D(f,'x')
```

```
## 12 * (6 * (x^2) + 8 * (x) - (x^3)) + (12 * x + 3) * (6 * (2 *
##     x) + 8 - 3 * x^2)
```

Thus, $\frac{df}{dx} = 12(6x^2+8x-x^3) + (12x+3)(12x+8-3x^2)$. Simplifying yields $\frac{df}{dx} = -48x^3 + 207x^2 + 228x + 24$

d. $\frac{df}{dx}$ where $f(x) = \frac{12x+3}{(6x^2+8x-x^3)}$

In order for us to evaluate this derivative we must employ the quotient rule, which is:
$\frac{d}{dx}(\frac{f(x)}{g(x)}) = \frac{f(x)'g(x)-f(x)g(x)'}{g(x)^2}$
Where $f(x) = 12x + 3$, $f(x)' = 12$, $g(x) = 6x^2 + 8x - x^3$, and $g(x)' = 12x + 8 - 3x^2$.
Plugging these values into the quotient rule yields:
$\frac{df}{dx} = \frac{12(6x^2+8x-x^3)-(12x+3)(12x+8-3x^2)}{(6x^2+8x-x^3)^2}$
Which simplifies to:
$\frac{df}{dx} = \frac{24x^3-117x^2-36x-24}{(6x^2+8x-x^3)^2}$
As with prior parts, R can also perform this evaluation. Doing so yields:

```
y <- expression((12*x+3)/(6*(x^2)+8*x-(x^3)))
y_prime <- D(y,'x')
print(y_prime)
```

```
## 12/(6 * (x^2) + 8 * x - (x^3)) - (12 * x + 3) * (6 * (2 * x) +
##     8 - 3 * x^2)/(6 * (x^2) + 8 * x - (x^3))^2
```

e. $\frac{df}{dx}$ where $f(x) = e^{-6x+2}$

For the equation $y = e^{f(x)}$, we employ a method similar to the product rule used in part (c). Since $e^{f(x)}$ follows the pattern $\frac{dy}{dx} = \frac{df(x)}{dx}e^{f(x)}$ in its derivation, we can apply this form to the equation in part (e)
$f(x) = -6x + 2$
$f'(x) = -6$
Thus, $\frac{df(x)}{dx} = -6e^{-6x+2}$

And once again, in R

```
y <- expression(e^((-6*x)+2))
y_prime <- D(y,'x')
print(y_prime)
```

```
## -(e^((-6 * x) + 2) * (log(e) * 6))
```

Since the above log operation would be in base e (ln), the function then is just multipled by the constant 6, yielding the same result as my derivation.

Q5

Indicate whether the following expressions are correct or incorrect. If incorrect, briefly explain why.

a. $log(x^{\beta}) = \beta \cdot log(x)$

CORRECT. This follows the power rule of logarithms

b. $log(0) = 1$

INCORRECT. $log(1) = 0$, but $log(0) = undefined$ as no base can be raised to a power to achieve a value of 0.

c. $log(6x) = 6 \cdot log(x)$

INCORRECT. The product rule of logarithms states that $log(A \cdot B) = log(A) + log(B)$.

d. $log(xyz) = log(x) + log(y) + log(z)$

CORRECT. This follows the product rule of logarithms.

e. $log(x^2) = log(x)log(x)$

INCORRECT. Per the power rule of logarithms, $log(A^B) = B \cdot log(A)$. The above represents $(log(x))^2$.

f. $\sum_{i=1}^{100} (X_i + Y_i + Z_i) = \sum_{i=1}^{100} X_i + \sum_{i=1}^{100} Y_i + \sum_{i=1}^{100} Z_i$

CORRECT. The summation of addition is the same as the addition of summation.

g. $\sum_{i=1}^{100} (X_i \cdot Y_i + Z_i) = \sum_{i=1}^{100} X_i \cdot \sum_{i=1}^{100} Y_i + \sum_{i=1}^{100} Z_i$

INCORRECT. The summation of a product is not equal to the product of summation. For example:

```
a <- 1:15
b <- 16:30

for(i in 1:length(a)){
  sum_product <- a[i]*b[i]
}
sum_product
```

```
## [1] 450
```

```
product_sum <- sum(a)*sum(b)
product_sum
```

```
## [1] 41400
```

h. $\lim_{n\to\infty} \frac{6}{n} = 0$

CORRECT. Dividing a constant by a large number approaching infinity approaches 0.

i. $\lim_{n\to\infty} \frac{6n^2+n}{n^2} = 0$

INCORRECT. Since the numerator and denominator are both growing at the same exponential rate of n^2 and the numerator has an additional n term, the limit as this approaches infinity would be greater than 0.

j. $\prod_{i=1}^{5} y_i = y_1 \cdot y_2 \cdot y_3 \cdot y_4 \cdot y_5$

CORRECT. This demonstrates the correct usage of the product operation.

k. $\prod_{i=1}^{5} e^{y_i} = e^{(y_1+y_2+y_3+y_4+y_5)}$

CORRECT. Multiplication of like bases allows you to instead add the exponents and then use the resulting sum on the base. In other words, $e^{y_1} + e^{y_2} + e^{y_3} + e^{y_4} + e^{y_5} = e^{(y_1+y_2+y_3+y_4+y_5)}$. This is the proper usage of the product function.

For the Questions that Follow:

Assume that the X,Y, and Z are independent random variables with expected values μ_X, μ_Y, and μ_Z and variances σ_X^2, σ_Y^2, and σ_Z^2 respectively.

Q6

Let W be a new random variable defined as:

$$W = 6X + Y - Z$$

a. What is $E[W]$?

Since $E[X] = \mu_x$, and by the properties of the expected value function $E[A + B] = E[A] + E[B]$ and $E[aA] = a \cdot E[A]$ we can determine that:
$E[W] = E[6X + Y - Z]$
$E[W] = E[6x] + E[Y] + E[-Z]$
$E[W] = 6E[X] + E[Y] - E[Z]$
$E[W] = 6\mu_x + \mu_y - \mu_z$

b. What is σ_W^2?

Variance has the following properties that I will utilize to evaluate σ_w^2. First, the scalar translations of random variables affect the variance by a squared factor of the scalar, such that $\sigma_{a \cdot A}^2 = a^2 \cdot \sigma_A^2$. Secondly, for independent random variables A and B, the variance of their sum or difference is equal to the sum of their variances, such that $\sigma_{A \pm B}^2 = \sigma_A^2 + \sigma_B^2$. With these properties in mind, we can now evaluate σ_W^2.
$\sigma_W^2 = \sigma_{6X+Y-Z}^2$
$\sigma_W^2 = \sigma_{6X}^2 + \sigma_Y^2 + \sigma_Z^2$
$\sigma_W^2 = 6^2 \cdot \sigma_X^2 + \sigma_Y^2 + \sigma_Z^2$

Q7

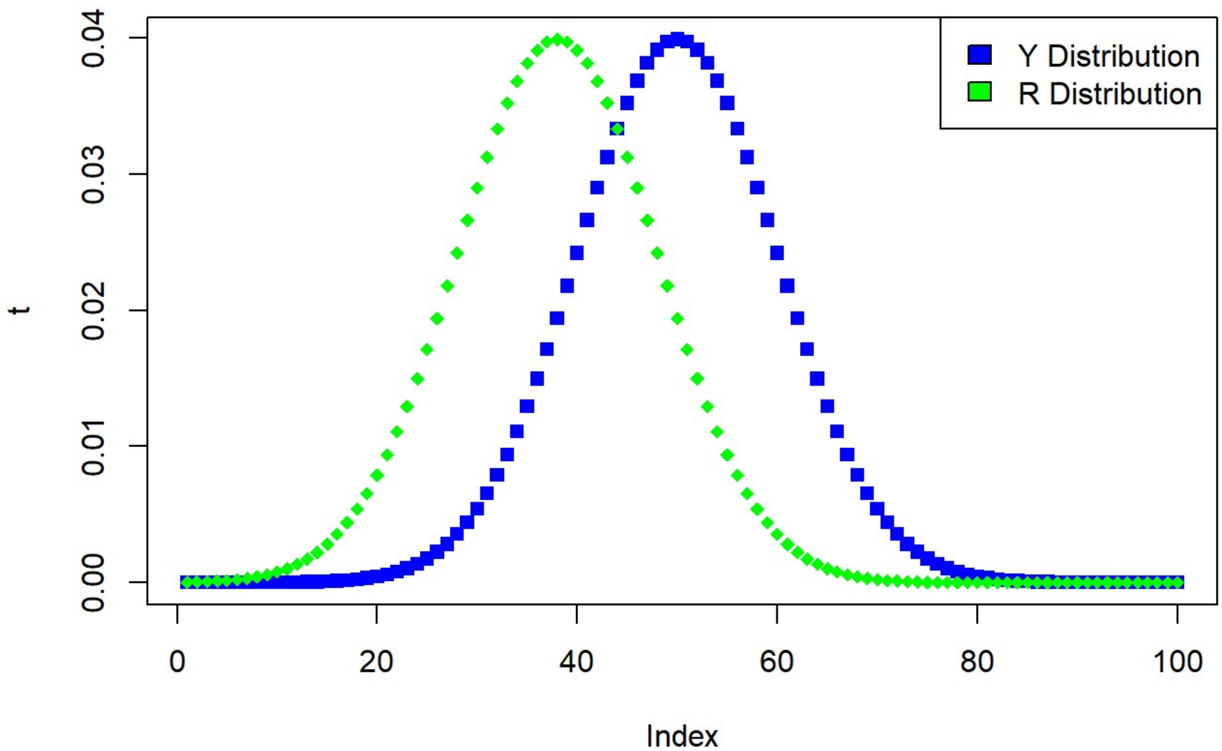
Let R be a new random variable defined as:

$$R = Y - 12$$

a. What is $E[R]$?
Since the expected value of a constant is a constant and a linear function, we know that $E[A + a] = E[A] + a$. Thus, the expected value of R can be determined as follows:
$E[R] = E[Y - 12]$
$E[R] = E[Y] + E[-12]$
From question 6 above, we know that $E[Y] = \mu_y$, thus
$E[R] = \mu_y - 12$

b. What is σ_R^2?
As an additional property of variance, we know that a random variable that is subjected to addition or subtraction does NOT experience changes to its variance. This is because the variation of data from the population mean is not affected by shifting the function up or down the number line by a constant - the spread of the data around the mean remains the same. Thus, $\sigma_{A+a}^2 = \sigma_A^2$, allowing us to solve for σ_R^2 as follows:
$\sigma_R^2 = \sigma_{Y-12}^2$
$\sigma_R^2 = \sigma_Y^2$

c. Assume that Y was normally distributed. Sketch the probability density functions for Y and R.



Q8

Let Q be a new random variable defined as:

$$Q = \frac{Z - \mu_Z}{\sigma_Z}$$

a. What is $E[Q]$?

Since μ_Z and σ_Z are both constant, as in they don't vary as Z varies, they can be treated as some rational constant. Thus:

$E[Q] = E[\frac{Z-\mu_z}{\sigma_Z}]$
$E[Q] = \frac{1}{\sigma_z}E[Z - \mu_z]$
$E[Q] = \frac{1}{\sigma_z}(E[Z] - \mu_Z)$
$E[Q] = \frac{1}{\sigma_Z}(\mu_z - \mu_z)$
$E[Q] = \frac{1}{\sigma_Z}(0)$
$E[Q] = 0$

b. What is σ_Q^2?

Using the previously derived properties of variance:

$\sigma_Q^2 = \sigma^2_{\frac{Z-\mu_Z}{\sigma_Z}}$

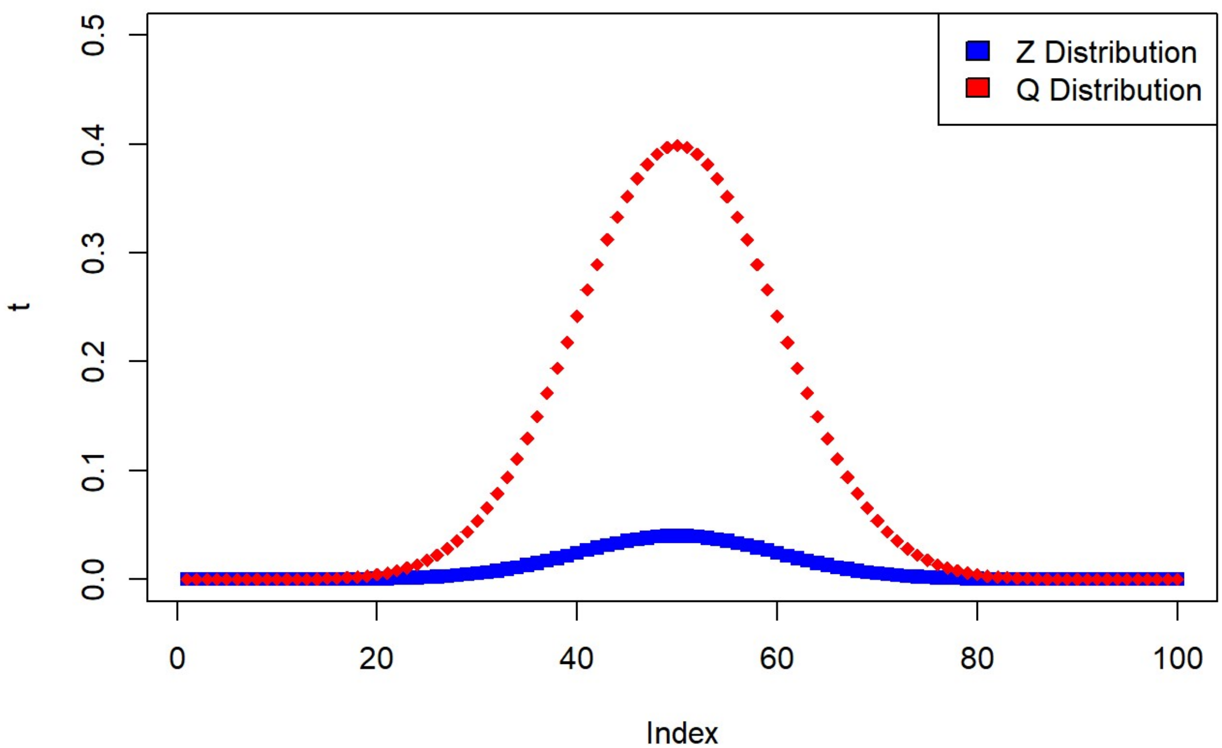
$\sigma_Q^2 = \sigma^2_{\frac{Z}{\sigma_Z} - \frac{\mu_Z}{\sigma_Z}}$

$\sigma_Q^2 = \sigma^2_{\frac{1}{\sigma_Z} Z - \frac{\mu_Z}{\sigma_Z}}$

$\sigma_Q^2 = \sigma_Z^2 \cdot \frac{1}{\sigma_z}^2$

$\sigma_Q^2 = 1$

c. Assume that Y was normally distributed. Sketch the probability density function for Z and Q.



R Exercises

Write a script that allows you to answer the questions below. Submit both a text version of your answers and the (heavily commented) script that you used to generate your answers.

R1

a. Load the tidyverse and vtable libraries.

```
#(a) Install/Load packages
#install.packages("tidyverse")
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.1
```

```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

```
## Warning: package 'tibble' was built under R version 4.3.1
```

```
## Warning: package 'tidyr' was built under R version 4.3.1
```

```
## Warning: package 'readr' was built under R version 4.3.1
```

```
## Warning: package 'purrr' was built under R version 4.3.1
```

```
## Warning: package 'dplyr' was built under R version 4.3.1
```

```
## Warning: package 'stringr' was built under R version 4.3.1
```

```
## Warning: package 'forcats' was built under R version 4.3.1
```

```
## Warning: package 'lubridate' was built under R version 4.3.1
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
#install.packages("vtable")
library(vtable)
```

```
## Warning: package 'vtable' was built under R version 4.3.1
```

```
## Loading required package: kableExtra
```

```
## Warning: package 'kableExtra' was built under R version 4.3.1
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(ggplot2)
# Above code only requires that the install.packages() functions be removed from
# comment status to be run for the first time (user should delete the leading '#')
```

b. Set pathways to the `data'',` scripts'', and ``tables and figures'' folders associated with Exercise 1.

```
#(b) Sets the file paths for data, scripts, and tables and figures
datapath <- "F:/Users/Devan/Documents/Education/ECO530/Assignments/Assignment 1/data"
scriptpath <- "F:/Users/Devan/Documents/Education/ECO530/Assignments/Assignment 1/script
s"
tablesfigurespath <- "F:/Users/Devan/Documents/Education/ECO530/Assignments/Assignment 1
/tables and figures"
```

c. Change the directory to the `data'' folder and read in the` cars.csv'' data file.

```
#(c) Sets the working directory to the data folder
setwd(datapath)
# Reads the csv file name 'cars' in the datapath folder and stores the data into a
# new data frame name cars.data
cars.data <- read.csv("cars.csv",header=TRUE)
```

d. Use the summary table command () to report the contents of the data.

```
#(d) Creates a summary table of the cars.data data frame
st(cars.data)
```

Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
price	74	6165	2949	3291	4220	6332	15906
mpg	74	21	5.8	12	18	25	41
rep78	69	3.4	0.99	1	3	4	5
headroom	74	3	0.85	1.5	2.5	3.5	5
trunk	74	14	4.3	5	10	17	23
weight	74	3019	777	1760	2250	3600	4840
length	74	188	22	142	170	204	233
turn	74	40	4.4	31	36	43	51
displacement	74	197	92	79	119	245	425
gear_ratio	74	3	0.46	2.2	2.7	3.4	3.9
foreign	74						
… Domestic	52	70%					
… Foreign	22	30%					

R2

Starting with the ``cars'' data frame:

a. Use `group_by()` and summarize() to create a data frame called ``summary'' containing the average price, mpg, and weight of Foreign/Domestic cars in the data.

```
#(a) Creates a summary of the average mpg, weight, and price of cars grouped by
# their foreign or domestic status. Stores the data in the summary.data data frame
summary.data <- cars.data %>%
  group_by(foreign) %>%
  summarize("Avg. MPG" = round(mean(mpg),digits=1),
            "Avg. Weight" = round(mean(weight),digits=0),
            "Avg. Price" = round(mean(price),digits=2))
summary.data
```

```
## # A tibble: 2 × 4
##   foreign  `Avg. MPG` `Avg. Weight` `Avg. Price`
##   <chr>         <dbl>         <dbl>        <dbl>
## 1 Domestic       19.8          3317        6072.
## 2 Foreign        24.8          2316        6385.
```

b. Use `kable()` to make a nicely formatted version of your ``summary()'' data frame.

```
#(b) Creates a formatted table from the summary.data data frame
kable(summary.data)
```

foreign	Avg. MPG	Avg. Weight	Avg. Price
Domestic	19.8	3317	6072.42
Foreign	24.8	2316	6384.68

c. Make a new data frame containing only Domestic Cars called "domestic.cars"

```
#(c) Creates a new data frame named domestic.cars with only information from cars.data
# on domestically manufactured cars.
domestic.cars <- cars.data[cars.data$foreign=="Domestic",]
domestic.cars
```

```
##                  make price mpg rep78 headroom trunk weight length turn
## 1         AMC Concord  4099  22     3      2.5    11   2930    186   40
## 2           AMC Pacer  4749  17     3      3.0    11   3350    173   40
## 3          AMC Spirit  3799  22    NA      3.0    12   2640    168   35
## 4       Buick Century  4816  20     3      4.5    16   3250    196   40
## 5       Buick Electra  7827  15     4      4.0    20   4080    222   43
## 6       Buick LeSabre  5788  18     3      4.0    21   3670    218   43
## 7          Buick Opel  4453  26    NA      3.0    10   2230    170   34
## 8         Buick Regal  5189  20     3      2.0    16   3280    200   42
## 9       Buick Riviera 10372  16     3      3.5    17   3880    207   43
## 10      Buick Skylark  4082  19     3      3.5    13   3400    200   42
## 11       Cad. Deville 11385  14     3      4.0    20   4330    221   44
## 12      Cad. Eldorado 14500  14     2      3.5    16   3900    204   43
## 13       Cad. Seville 15906  21     3      3.0    13   4290    204   45
## 14     Chev. Chevette  3299  29     3      2.5     9   2110    163   34
## 15       Chev. Impala  5705  16     4      4.0    20   3690    212   43
## 16       Chev. Malibu  4504  22     3      3.5    17   3180    193   31
## 17 Chev. Monte Carlo  5104  22     2      2.0    16   3220    200   41
## 18        Chev. Monza  3667  24     2      2.0     7   2750    179   40
## 19         Chev. Nova  3955  19     3      3.5    13   3430    197   43
## 20         Dodge Colt  3984  30     5      2.0     8   2120    163   35
## 21     Dodge Diplomat  4010  18     2      4.0    17   3600    206   46
## 22       Dodge Magnum  5886  16     2      4.0    17   3600    206   46
## 23     Dodge St. Regis 6342  17     2      4.5    21   3740    220   46
## 24        Ford Fiesta  4389  28     4      1.5     9   1800    147   33
## 25       Ford Mustang  4187  21     3      2.0    10   2650    179   43
## 26 Linc. Continental 11497  12     3      3.5    22   4840    233   51
## 27       Linc. Mark V 13594  12     3      2.5    18   4720    230   48
## 28  Linc. Versailles 13466  14     3      3.5    15   3830    201   41
## 29       Merc. Bobcat  3829  22     4      3.0     9   2580    169   39
## 30       Merc. Cougar  5379  14     4      3.5    16   4060    221   48
## 31      Merc. Marquis  6165  15     3      3.5    23   3720    212   44
## 32      Merc. Monarch  4516  18     3      3.0    15   3370    198   41
## 33         Merc. XR-7  6303  14     4      3.0    16   4130    217   45
## 34       Merc. Zephyr  3291  20     3      3.5    17   2830    195   43
## 35            Olds 98  8814  21     4      4.0    20   4060    220   43
## 36     Olds Cutl Supr  5172  19     3      2.0    16   3310    198   42
## 37       Olds Cutlass  4733  19     3      4.5    16   3300    198   42
## 38      Olds Delta 88  4890  18     4      4.0    20   3690    218   42
## 39         Olds Omega  4181  19     3      4.5    14   3370    200   43
## 40      Olds Starfire  4195  24     1      2.0    10   2730    180   40
## 41      Olds Toronado 10371  16     3      3.5    17   4030    206   43
## 42        Plym. Arrow  4647  28     3      2.0    11   3260    170   37
## 43        Plym. Champ  4425  34     5      2.5    11   1800    157   37
## 44      Plym. Horizon  4482  25     3      4.0    17   2200    165   36
## 45      Plym. Sapporo  6486  26    NA      1.5     8   2520    182   38
## 46       Plym. Volare  4060  18     2      5.0    16   3330    201   44
## 47     Pont. Catalina  5798  18     4      4.0    20   3700    214   42
## 48     Pont. Firebird  4934  18     1      1.5     7   3470    198   42
## 49  Pont. Grand Prix  5222  19     3      2.0    16   3210    201   45
## 50      Pont. Le Mans  4723  19     3      3.5    17   3200    199   40
```

```
## 51     Pont. Phoenix  4424  19    NA       3.5     13   3420    203   43
## 52     Pont. Sunbird  4172  24     2       2.0      7   2690    179   41
##    displacement gear_ratio  foreign
## 1           121       3.58 Domestic
## 2           258       2.53 Domestic
## 3           121       3.08 Domestic
## 4           196       2.93 Domestic
## 5           350       2.41 Domestic
## 6           231       2.73 Domestic
## 7           304       2.87 Domestic
## 8           196       2.93 Domestic
## 9           231       2.93 Domestic
## 10          231       3.08 Domestic
## 11          425       2.28 Domestic
## 12          350       2.19 Domestic
## 13          350       2.24 Domestic
## 14          231       2.93 Domestic
## 15          250       2.56 Domestic
## 16          200       2.73 Domestic
## 17          200       2.73 Domestic
## 18          151       2.73 Domestic
## 19          250       2.56 Domestic
## 20           98       3.54 Domestic
## 21          318       2.47 Domestic
## 22          318       2.47 Domestic
## 23          225       2.94 Domestic
## 24           98       3.15 Domestic
## 25          140       3.08 Domestic
## 26          400       2.47 Domestic
## 27          400       2.47 Domestic
## 28          302       2.47 Domestic
## 29          140       2.73 Domestic
## 30          302       2.75 Domestic
## 31          302       2.26 Domestic
## 32          250       2.43 Domestic
## 33          302       2.75 Domestic
## 34          140       3.08 Domestic
## 35          350       2.41 Domestic
## 36          231       2.93 Domestic
## 37          231       2.93 Domestic
## 38          231       2.73 Domestic
## 39          231       3.08 Domestic
## 40          151       2.73 Domestic
## 41          350       2.41 Domestic
## 42          156       3.05 Domestic
## 43           86       2.97 Domestic
## 44          105       3.37 Domestic
## 45          119       3.54 Domestic
## 46          225       3.23 Domestic
## 47          231       2.73 Domestic
## 48          231       3.08 Domestic
```

```
## 49           231       2.93 Domestic
## 50           231       2.93 Domestic
## 51           231       3.08 Domestic
## 52           151       2.73 Domestic
```

d. Add a variable (using the mutate() function) to the domestic.cars data frame that is equal to `price'' divided by` mpg".

```
#(d) Adds a column to the domestic.cars data frame that contains the value of
# that car's price divided by its mpg rating. Rounds to 1 digit after the decimal.
domestic.cars<-  domestic.cars %>% mutate(pricepermpg = round(price/mpg,digits=1))
domestic.cars
```

```
##                  make price mpg rep78 headroom trunk weight length turn
## 1        AMC Concord  4099  22     3      2.5    11   2930    186   40
## 2          AMC Pacer  4749  17     3      3.0    11   3350    173   40
## 3         AMC Spirit  3799  22    NA      3.0    12   2640    168   35
## 4      Buick Century  4816  20     3      4.5    16   3250    196   40
## 5      Buick Electra  7827  15     4      4.0    20   4080    222   43
## 6      Buick LeSabre  5788  18     3      4.0    21   3670    218   43
## 7         Buick Opel  4453  26    NA      3.0    10   2230    170   34
## 8        Buick Regal  5189  20     3      2.0    16   3280    200   42
## 9      Buick Riviera 10372  16     3      3.5    17   3880    207   43
## 10     Buick Skylark  4082  19     3      3.5    13   3400    200   42
## 11      Cad. Deville 11385  14     3      4.0    20   4330    221   44
## 12     Cad. Eldorado 14500  14     2      3.5    16   3900    204   43
## 13      Cad. Seville 15906  21     3      3.0    13   4290    204   45
## 14    Chev. Chevette  3299  29     3      2.5     9   2110    163   34
## 15      Chev. Impala  5705  16     4      4.0    20   3690    212   43
## 16      Chev. Malibu  4504  22     3      3.5    17   3180    193   31
## 17 Chev. Monte Carlo  5104  22     2      2.0    16   3220    200   41
## 18       Chev. Monza  3667  24     2      2.0     7   2750    179   40
## 19        Chev. Nova  3955  19     3      3.5    13   3430    197   43
## 20        Dodge Colt  3984  30     5      2.0     8   2120    163   35
## 21    Dodge Diplomat  4010  18     2      4.0    17   3600    206   46
## 22      Dodge Magnum  5886  16     2      4.0    17   3600    206   46
## 23   Dodge St. Regis  6342  17     2      4.5    21   3740    220   46
## 24       Ford Fiesta  4389  28     4      1.5     9   1800    147   33
## 25      Ford Mustang  4187  21     3      2.0    10   2650    179   43
## 26 Linc. Continental 11497  12     3      3.5    22   4840    233   51
## 27      Linc. Mark V 13594  12     3      2.5    18   4720    230   48
## 28  Linc. Versailles 13466  14     3      3.5    15   3830    201   41
## 29      Merc. Bobcat  3829  22     4      3.0     9   2580    169   39
## 30      Merc. Cougar  5379  14     4      3.5    16   4060    221   48
## 31     Merc. Marquis  6165  15     3      3.5    23   3720    212   44
## 32     Merc. Monarch  4516  18     3      3.0    15   3370    198   41
## 33        Merc. XR-7  6303  14     4      3.0    16   4130    217   45
## 34      Merc. Zephyr  3291  20     3      3.5    17   2830    195   43
## 35           Olds 98  8814  21     4      4.0    20   4060    220   43
## 36    Olds Cutl Supr  5172  19     3      2.0    16   3310    198   42
## 37      Olds Cutlass  4733  19     3      4.5    16   3300    198   42
## 38     Olds Delta 88  4890  18     4      4.0    20   3690    218   42
## 39        Olds Omega  4181  19     3      4.5    14   3370    200   43
## 40     Olds Starfire  4195  24     1      2.0    10   2730    180   40
## 41     Olds Toronado 10371  16     3      3.5    17   4030    206   43
## 42       Plym. Arrow  4647  28     3      2.0    11   3260    170   37
## 43       Plym. Champ  4425  34     5      2.5    11   1800    157   37
## 44     Plym. Horizon  4482  25     3      4.0    17   2200    165   36
## 45     Plym. Sapporo  6486  26    NA      1.5     8   2520    182   38
## 46      Plym. Volare  4060  18     2      5.0    16   3330    201   44
## 47    Pont. Catalina  5798  18     4      4.0    20   3700    214   42
## 48    Pont. Firebird  4934  18     1      1.5     7   3470    198   42
## 49  Pont. Grand Prix  5222  19     3      2.0    16   3210    201   45
## 50     Pont. Le Mans  4723  19     3      3.5    17   3200    199   40
```

```
## 51     Pont. Phoenix  4424  19    NA      3.5    13   3420    203   43
## 52     Pont. Sunbird  4172  24     2      2.0     7   2690    179   41
##    displacement gear_ratio  foreign pricepermpg
## 1           121       3.58 Domestic       186.3
## 2           258       2.53 Domestic       279.4
## 3           121       3.08 Domestic       172.7
## 4           196       2.93 Domestic       240.8
## 5           350       2.41 Domestic       521.8
## 6           231       2.73 Domestic       321.6
## 7           304       2.87 Domestic       171.3
## 8           196       2.93 Domestic       259.4
## 9           231       2.93 Domestic       648.2
## 10          231       3.08 Domestic       214.8
## 11          425       2.28 Domestic       813.2
## 12          350       2.19 Domestic      1035.7
## 13          350       2.24 Domestic       757.4
## 14          231       2.93 Domestic       113.8
## 15          250       2.56 Domestic       356.6
## 16          200       2.73 Domestic       204.7
## 17          200       2.73 Domestic       232.0
## 18          151       2.73 Domestic       152.8
## 19          250       2.56 Domestic       208.2
## 20           98       3.54 Domestic       132.8
## 21          318       2.47 Domestic       222.8
## 22          318       2.47 Domestic       367.9
## 23          225       2.94 Domestic       373.1
## 24           98       3.15 Domestic       156.8
## 25          140       3.08 Domestic       199.4
## 26          400       2.47 Domestic       958.1
## 27          400       2.47 Domestic      1132.8
## 28          302       2.47 Domestic       961.9
## 29          140       2.73 Domestic       174.0
## 30          302       2.75 Domestic       384.2
## 31          302       2.26 Domestic       411.0
## 32          250       2.43 Domestic       250.9
## 33          302       2.75 Domestic       450.2
## 34          140       3.08 Domestic       164.6
## 35          350       2.41 Domestic       419.7
## 36          231       2.93 Domestic       272.2
## 37          231       2.93 Domestic       249.1
## 38          231       2.73 Domestic       271.7
## 39          231       3.08 Domestic       220.1
## 40          151       2.73 Domestic       174.8
## 41          350       2.41 Domestic       648.2
## 42          156       3.05 Domestic       166.0
## 43           86       2.97 Domestic       130.1
## 44          105       3.37 Domestic       179.3
## 45          119       3.54 Domestic       249.5
## 46          225       3.23 Domestic       225.6
## 47          231       2.73 Domestic       322.1
## 48          231       3.08 Domestic       274.1
```

```
## 49          231       2.93 Domestic       274.8
## 50          231       2.93 Domestic       248.6
## 51          231       3.08 Domestic       232.8
## 52          151       2.73 Domestic       173.8
```

R3

Generate vectors containing 250 draws from each of the following normal distributions:

- $var1 \sim N(3, 1)$
- $var2 \sim N(-1, 2)$
- $var3 \sim N(2, 3)$

Place all three variables in a data frame together called ``random.draws''. Include in your data frame a variable called “id” that indicates an observations row number.

```r
# Creates var1, var2, and var3. Performs 250 draws from a normal distribution
# based on the mean and standard deviation provided
var1 <- rnorm(250,mean=3,sd=sqrt(1))
var2 <- rnorm(250,mean=-1,sd=sqrt(2))
var3 <- rnorm(250,mean=2,sd=sqrt(3))

# Creates the id variable for later use
id <- 1:250

# Creates the 'random.draws' data frame by combining the vectors var1, var2, and var3
random.draws <- data.frame(id,var1,var2,var3)
random.draws
```

```
##      id      var1        var2         var3
## 1     1 2.3406391  0.46490414  5.374726871
## 2     2 4.1584851 -2.76329634  1.366386524
## 3     3 3.7083028  1.86820118  5.260092569
## 4     4 1.6986142 -1.29788536  2.849975597
## 5     5 4.2638219  0.39146543  3.850547383
## 6     6 2.4595408  1.00276420 -2.793867550
## 7     7 2.3945001 -0.83954071  1.814266826
## 8     8 4.2481664 -0.18249298  5.259665086
## 9     9 3.1326033  0.05280666 -1.215262315
## 10   10 3.7339385 -1.77505830  2.410117185
## 11   11 2.2416654 -1.74207398  2.636635646
## 12   12 3.5446571 -1.05874223  5.827492428
## 13   13 3.3160953 -1.63036969  1.888556329
## 14   14 3.3749880 -1.21667793  0.195070517
## 15   15 4.2724221 -0.49169971  3.805155101
## 16   16 2.4106619 -4.04304343  7.686624526
## 17   17 4.1521901  1.31464423 -0.817553050
## 18   18 2.3213758 -0.31989191  1.040014893
## 19   19 3.0093241 -0.10972609 -0.237746339
## 20   20 2.8010698 -0.93145613  2.674389929
## 21   21 3.3040054 -1.63125729  2.560896191
## 22   22 3.2493362 -2.38904336  2.339524663
## 23   23 3.4819438 -3.82023192 -0.227130776
## 24   24 2.0719775  0.97485942  1.772051755
## 25   25 2.2253335 -0.26596263  1.645701059
## 26   26 3.5118037 -1.44701092  1.871523219
## 27   27 2.6464869  1.54476957  4.940668392
## 28   28 2.2715900  1.23778816  1.075669197
## 29   29 1.9458871 -0.48711852  2.196643642
## 30   30 3.6406332 -0.51716699  4.047465335
## 31   31 3.4318606 -0.17090211 -0.165327295
## 32   32 2.2345026 -2.67656309 -0.137488964
## 33   33 2.4099141 -2.32698940  1.315235112
## 34   34 2.3112589 -2.62379895  2.401467568
```

```
## 35   35 2.6687803  0.39045519  4.811161885
## 36   36 4.8379192 -0.09741336 -2.130957680
## 37   37 0.8864581 -1.25340039  4.284323822
## 38   38 4.8410500 -2.85106560  1.861851111
## 39   39 2.5601531  2.03881130 -0.257969151
## 40   40 3.6374408 -2.75647136  1.791679786
## 41   41 3.0331648 -0.53351911 -0.165620966
## 42   42 2.6394468 -0.12896756  0.824371820
## 43   43 2.0444426 -4.11330587  6.106475652
## 44   44 2.4620564 -0.48740760  0.535228897
## 45   45 0.6386191 -1.03497458  3.408719754
## 46   46 3.7177482 -0.03659012  2.110166647
## 47   47 4.6164864 -2.18522180  2.972442185
## 48   48 1.5609729  0.66663094  3.475679352
## 49   49 2.1736694  0.54083761  2.477908662
## 50   50 5.0282428 -1.95577589  1.071786410
## 51   51 3.0389713 -1.50885324  2.685779307
## 52   52 3.4467969 -1.10456063 -0.548361588
## 53   53 2.2550954 -2.51459146  2.539800170
## 54   54 3.4844987 -1.56972932 -0.185380928
## 55   55 5.6436888 -1.48254450  2.542833573
## 56   56 2.4318607  0.09367543 -1.510131667
## 57   57 3.1783977 -1.35403983  2.990127201
## 58   58 3.5862842 -1.01875650  2.719176245
## 59   59 2.1446753 -3.00860785  2.253043059
## 60   60 1.2930866  0.27947737  2.381447661
## 61   61 2.5391142  1.31062503 -0.440006017
## 62   62 2.7496083  0.74267218  1.990085915
## 63   63 4.1857750  1.20884517 -0.278432143
## 64   64 2.1101243 -1.34502470  4.488931872
## 65   65 2.6100962 -0.06737674  0.405276527
## 66   66 1.6108610 -3.83841590  0.771036702
## 67   67 2.3238506 -0.92469994  1.935558004
## 68   68 3.4181948 -1.71584599  3.232155650
## 69   69 1.4478418  1.42661622  4.025367279
## 70   70 3.6967678  0.42923920  4.298649334
## 71   71 1.1317960 -2.60396317  0.757428076
## 72   72 3.7676487 -1.27480658  0.774765624
## 73   73 1.7286794 -1.15029607 -0.325372085
## 74   74 1.4906844 -0.79699175 -0.234405010
## 75   75 1.4884850 -3.39665459  1.102274604
## 76   76 3.3551154 -2.92340644 -0.003664721
## 77   77 3.8019597 -0.60674000  1.511886905
## 78   78 4.2366532 -0.95211590  2.489506873
## 79   79 2.6679904 -2.30465240  0.530604387
## 80   80 2.8985913 -0.68627507  1.208328091
## 81   81 1.7277361 -2.72209718  3.177820794
## 82   82 3.8573044 -0.71404849  1.964605008
## 83   83 3.4623827 -2.19703092 -0.058679394
## 84   84 2.1329039 -2.37377062  2.582100931
## 85   85 3.3049417  0.66224023  4.087393791
```

```
## 86   86 3.1530188 -1.97656061  1.286697985
## 87   87 3.6645094 -0.27730771  2.465737182
## 88   88 4.3400146 -1.30728094  1.731987921
## 89   89 4.3454108 -2.18738039  3.804669733
## 90   90 3.8303533 -1.58398479  4.556461678
## 91   91 3.6553977 -0.94643861  3.201299655
## 92   92 4.3586748 -1.13518071  0.151602048
## 93   93 1.3213564  0.82178796  1.508099252
## 94   94 3.8968970 -0.57968415  0.852734056
## 95   95 1.1289800 -1.29138954  2.367415121
## 96   96 3.7202125 -0.42998146  2.825527429
## 97   97 2.4654164 -1.46773207  1.161161244
## 98   98 3.0180332 -0.15637301  2.329866090
## 99   99 4.3615217 -2.85987796  1.171197614
## 100 100 3.5097910 -0.89820263 -0.293086966
## 101 101 2.9680694 -0.67814023  2.447573124
## 102 102 5.6996832 -0.91132462  0.958024719
## 103 103 3.9240192 -1.52608013  1.829397715
## 104 104 2.1703895 -3.48259096  1.505236100
## 105 105 2.2974114  0.18910121  1.381193351
## 106 106 4.3706061 -0.93986360  1.186533375
## 107 107 3.7084236 -0.65711119 -0.515883828
## 108 108 3.6433099 -2.07185823  1.822868934
## 109 109 3.6368532 -0.76109549  6.803451065
## 110 110 3.9615594 -2.72091080  1.810803008
## 111 111 4.5413570  0.19184959  3.693557269
## 112 112 2.6842351 -0.59477486  0.053029262
## 113 113 3.0973255 -1.50307831 -2.910722142
## 114 114 2.7808387 -1.68586421  4.726437408
## 115 115 4.4655897 -0.62256662  2.369982632
## 116 116 4.9248422 -1.34883170  4.127635109
## 117 117 2.3874604  3.04942290 -2.171243222
## 118 118 2.2193305 -1.00028089  4.387297098
## 119 119 3.7079846 -3.30269834 -2.045259915
## 120 120 2.0900897 -1.61027168  2.567763666
## 121 121 2.5355019 -0.89580652  2.752734064
## 122 122 3.2546358 -1.85939260  0.869048721
## 123 123 1.5681212 -1.54511061  1.212414191
## 124 124 5.1613758  0.77755437 -1.025864624
## 125 125 3.8514311  1.38461483  2.330707840
## 126 126 3.1452648 -0.96007648  3.522715464
## 127 127 4.1176548 -0.43409244  1.864874173
## 128 128 3.4896020  0.09977402  2.972072566
## 129 129 2.3811066 -1.93498846  2.889943902
## 130 130 3.4339098  0.14871784  1.801483393
## 131 131 3.3625692 -0.54089277  1.101415858
## 132 132 2.0967136 -1.69466895  2.388696438
## 133 133 4.7855621 -0.19491332  2.365336328
## 134 134 1.7465504  0.03863181  0.151587694
## 135 135 4.9328789 -0.40575920 -0.127639385
## 136 136 2.2199372 -1.31256632  4.521242592
```

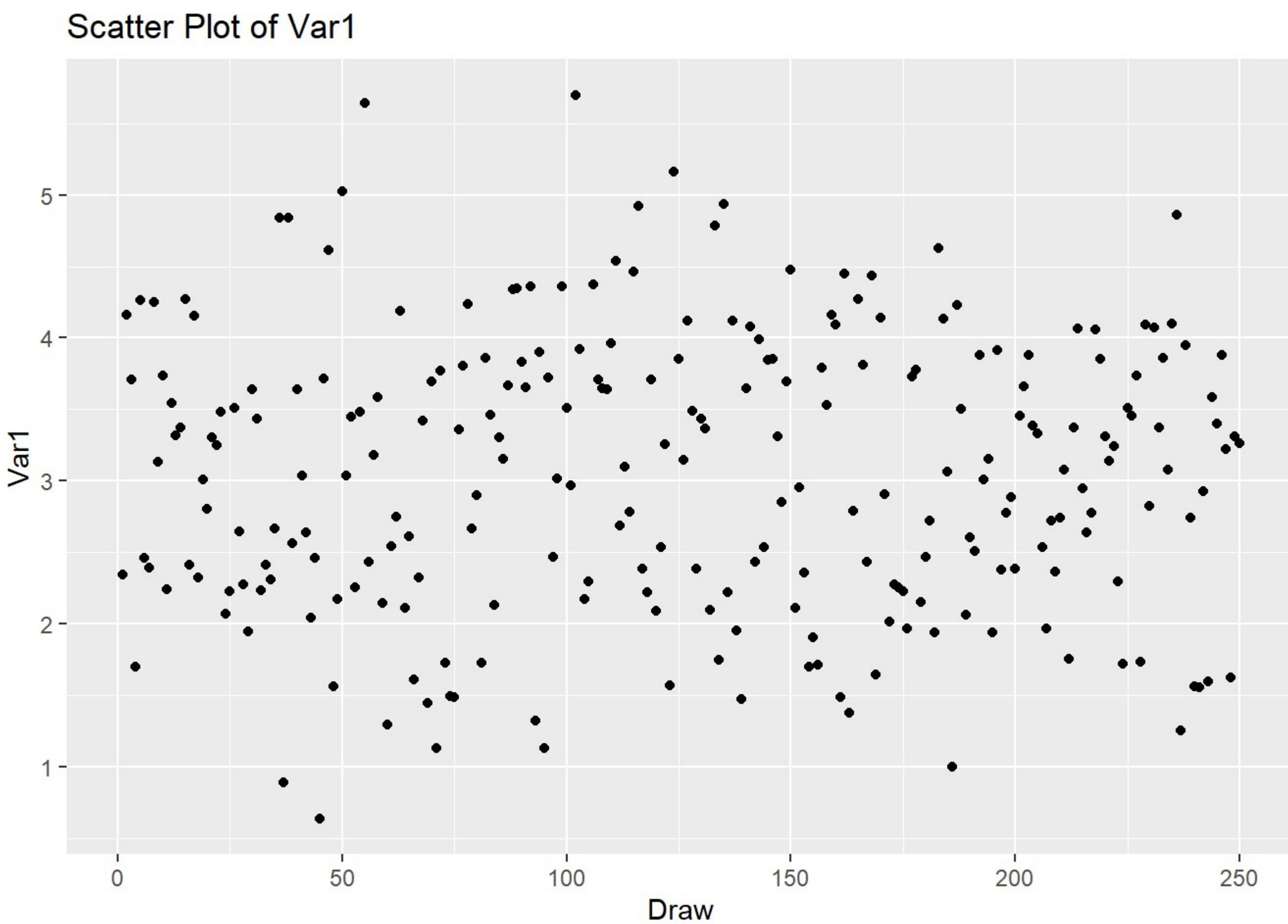
```
## 137 137 4.1225183 -1.19452514  1.799620151
## 138 138 1.9501096 -0.36472410  2.969817668
## 139 139 1.4689719 -2.37837906  4.049884649
## 140 140 3.6495796 -0.27536693 -0.135757521
## 141 141 4.0798462 -2.03851204  1.939080739
## 142 142 2.4340098 -0.50482837  1.624537344
## 143 143 3.9870481 -1.90524007  2.069365945
## 144 144 2.5378565  2.12838971  1.193046740
## 145 145 3.8421679  1.19831685  4.004269619
## 146 146 3.8502573 -0.64929068 -0.130401330
## 147 147 3.3100238 -1.80733499  1.522225738
## 148 148 2.8508458 -0.83639811  2.460753590
## 149 149 3.6937165 -1.78118966  1.644147096
## 150 150 4.4733310 -3.01628393  0.957947572
## 151 151 2.1090241 -1.43525699  0.634645683
## 152 152 2.9513976 -2.32596408  1.582600853
## 153 153 2.3567002  0.56532699  1.450871477
## 154 154 1.6969808  0.43392182  0.279469635
## 155 155 1.9011182  0.05760114  3.563472238
## 156 156 1.7092503  0.99038992  4.166562817
## 157 157 3.7871191 -0.42557313  2.539972336
## 158 158 3.5316692 -0.87111700 -0.119288849
## 159 159 4.1610813 -0.13031935 -1.479050240
## 160 160 4.0904419  0.89673430  2.308109258
## 161 161 1.4839443 -3.44726179  2.569418033
## 162 162 4.4469075 -2.33294184  2.141536949
## 163 163 1.3728467 -0.29009770  4.798740108
## 164 164 2.7864967 -0.12385917  3.018555426
## 165 165 4.2702699 -2.10795015  2.140614207
## 166 166 3.8081393 -3.38682977  1.976733226
## 167 167 2.4348547  0.30359204  1.472670434
## 168 168 4.4339205 -2.68510226  2.095346975
## 169 169 1.6458827 -1.89406090  3.452224139
## 170 170 4.1382490 -1.53696469  3.992446752
## 171 171 2.9028764 -1.72572655  1.477790470
## 172 172 2.0141719 -2.48036908  4.098746719
## 173 173 2.2776278 -1.64375422  4.716069983
## 174 174 2.2510162 -0.11116854  0.848922328
## 175 175 2.2255744  2.15548351  4.455701362
## 176 176 1.9665040 -1.48090601  2.266940937
## 177 177 3.7273687  1.76220661  3.491240015
## 178 178 3.7749585 -2.43709963  2.468743303
## 179 179 2.1536034 -1.93941145  3.593149238
## 180 180 2.4693451  0.23440483  1.189995109
## 181 181 2.7199348 -1.21243288  6.119654648
## 182 182 1.9401715 -1.46601045  2.562581732
## 183 183 4.6249937 -1.00506956 -0.288881638
## 184 184 4.1312076  2.74739269  3.436588967
## 185 185 3.0645149 -4.45105115  2.043436934
## 186 186 0.9979898 -0.65191503  2.832342285
## 187 187 4.2263074 -0.35694618  3.032392294
```

```
## 188 188 3.5055028 -2.23293688  2.663297506
## 189 189 2.0637296 -1.22486581  2.931972356
## 190 190 2.6044970 -1.23911855 -1.077195564
## 191 191 2.5076496  0.96859164 -0.533603510
## 192 192 3.8796115 -0.85087284 -0.520982795
## 193 193 3.0072800 -2.42693514  1.575751686
## 194 194 3.1558667 -0.69451796  0.784070782
## 195 195 1.9359587 -1.41586764  1.273827896
## 196 196 3.9105010 -2.62911990 -0.110310231
## 197 197 2.3778319  0.43119416  2.003090731
## 198 198 2.7779552 -0.16295723  1.896514050
## 199 199 2.8854024 -0.36299792  2.653760223
## 200 200 2.3825735 -2.46223723  3.339754202
## 201 201 3.4546565 -0.28073167  1.024046385
## 202 202 3.6616661 -0.59742253  3.772969574
## 203 203 3.8793264  1.59810842  2.249145936
## 204 204 3.3888127  2.37363855  1.134187877
## 205 205 3.3282333 -0.54687923  2.220393516
## 206 206 2.5363598 -0.14997864  0.634405401
## 207 207 1.9636876 -2.46711379  0.275894633
## 208 208 2.7180401  0.11249630  3.002500724
## 209 209 2.3614748 -0.90060219  0.773625853
## 210 210 2.7402490  1.17252139  2.477254409
## 211 211 3.0779090 -2.84435663  0.442263101
## 212 212 1.7534337 -0.63242008 -1.078124072
## 213 213 3.3752313 -0.72280655  1.698194831
## 214 214 4.0668094  2.21833315  1.714531146
## 215 215 2.9489013  3.03326084  2.929171866
## 216 216 2.6389585 -0.20163116  3.451378299
## 217 217 2.7746594 -1.90470212  1.116961065
## 218 218 4.0595564 -2.35916083  0.741508187
## 219 219 3.8522095 -0.46339348  2.079610156
## 220 220 3.3120428  0.37836084  2.100491310
## 221 221 3.1411880 -1.55727111  0.264677096
## 222 222 3.2406915 -0.33000100 -1.407198162
## 223 223 2.2938487 -1.05084804  4.502615375
## 224 224 1.7200654  1.55867696  2.962379966
## 225 225 3.5117919 -2.74775890  4.326418039
## 226 226 3.4519592 -1.08356357 -1.592525945
## 227 227 3.7345215 -2.15975535  0.229149390
## 228 228 1.7317686 -2.11604288  3.449765355
## 229 229 4.0937645 -0.70319347 -2.888873391
## 230 230 2.8241075 -2.37201145  3.117612773
## 231 231 4.0713683 -0.75327998  0.046244283
## 232 232 3.3689491 -0.93608619  1.949876928
## 233 233 3.8606570  0.22158248  1.207296497
## 234 234 3.0775890 -0.96701826 -0.415168354
## 235 235 4.0965670 -1.23852383  0.896419323
## 236 236 4.8590189 -2.05353440  1.647317872
## 237 237 1.2520256  1.60468891  4.921587356
## 238 238 3.9498457 -0.13472283  2.584324963
```

```
## 239 239 2.7439308 -0.90514650  3.990705856
## 240 240 1.5629543 -2.17645785  4.187936136
## 241 241 1.5512238 -0.85110292  1.434321618
## 242 242 2.9252098  3.83897151  1.972211830
## 243 243 1.5981426  1.50193651  2.967081344
## 244 244 3.5842115  0.17170505  4.389265809
## 245 245 3.3972767  1.14311186  2.820916055
## 246 246 3.8761750  0.98199998  2.390900227
## 247 247 3.2238741  0.18368541 -0.953079464
## 248 248 1.6197673 -0.63969394  2.667149146
## 249 249 3.3109228 -1.49408181  3.384539733
## 250 250 3.2641871 -0.37092738  3.368560976
```

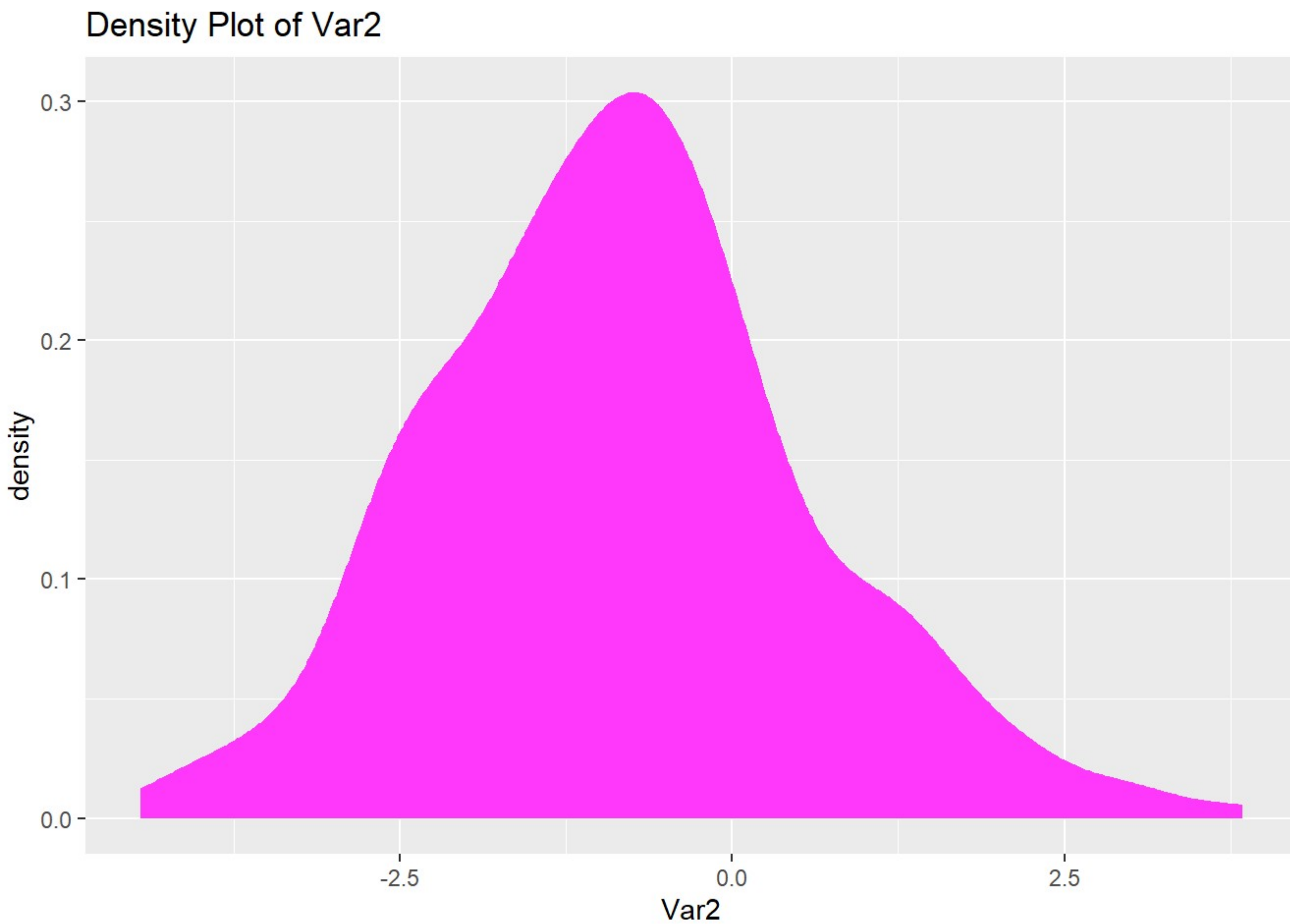
R4

Generate a scatter plot of `var1` . Make sure that your scatter plot has a title and informative labels on the axes.



R5

Generate a density plot of `var2`. Choose a fill color different than ggplot()'s default.



R6

Create a new variable in your data frame called `var4` that is equal to the sum of the other three.

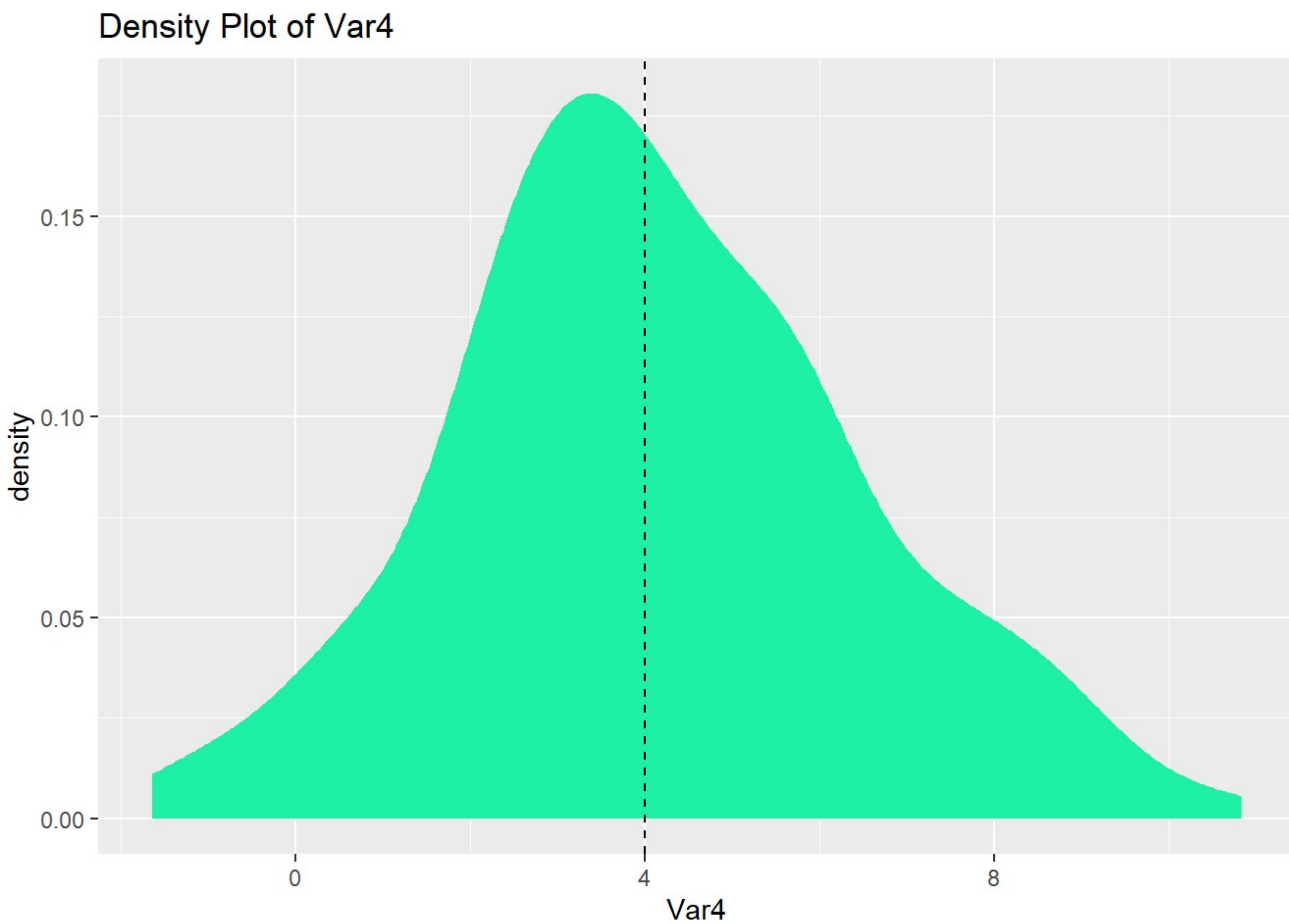
```
```r
# Creates var4 as a summation of vars 1, 2, & 3 and assigns it to the random.draws data frame
random.draws <- random.draws %>% mutate(var4 = var1 + var2 + var3)
```
```

a. Where do you expect it's density to be centered when you plot it?
Since var4 is a combination of vars 1-3, I suspect that the expected value of var4 should be equal to 4. This is because $E[var1] = \mu_{var1}$, and since $var4 = var1 + var2 + var3$ ->

$$E[var4] = E[var1 + var2 + var3]$$

$E[var4] = E[var1] + E[var2] + E[var3]$
$E[var4] = \mu_{var1} + \mu_{var2} + \mu_{var3}$
$E[var4] = 3 + (-1) + 2$
$E[var4] = 4$

b. Create a density plot of your new variable, placing a dashed, vertical line at it's expected value.



R7

Create a new variable in your data frame called `var5` by subtracting one from each element in `var3` and dividing each element by 2.

a. Where do you expect the new variable's density to be centered?
I expect that the new density plot will be centered around x=0.5. This represents the expected value of var5, which is $E[var5] = E[(var3 - 1)/2]$

$E[var5] = \frac{1}{2}E[var3 - 1]$
$E[var5] = \frac{1}{2}(E[var3] - 1)$
$E[var5] = \frac{1}{2}(2 - 1)$
$E[var5] = \frac{1}{2}(1) = 0.5$

b. How else do you expect it to change?

Since the coefficient of $\frac{1}{2}$ is part of the var5 transformation, I expect to see tighter grouping of values around the mean. In other words, $\sigma^2_{var5} < \sigma^2_{var3}$. This is drawn from the variance properties utilized earlier in the excerise.

c. Create a density plot of your new variable placing a dashed, vertical line at it's expected value.