Programming and Quantitative Skills 2025 Exam Solutions

Short-Answer Questions (4 points)

Question 1 (1 point)

Use R to calculate the following:

$$\frac{2^3 + \log_3(10)}{\sqrt{12}}$$

Provide both the R command and numeric value in your answer.

Answer:

```r
(2^3 + log(10, 3)) / sqrt(12)
```

```
[1] 2.914436
```

Question 2 (1 point)

Write an R command in the box below that repeats the sequence $(10, 9, 8, \ldots, 3, 2, 1)$ five times.

The resulting sequence should have the form:

$$(10, 9, 8, \ldots, 3, 2, 1, 10, 9, 8, \ldots, 3, 2, 1, \ldots)$$

and should have 50 elements in total.

Your answer must use a sequence-generating function. It is not permitted to write out the full vector manually.

Answer:

```r
rep(seq(from = 10, to = 1, by = -1), times = 5)
```

```
 [1] 10  9  8  7  6  5  4  3  2  1 10  9  8  7  6  5  4  3  2  1 10  9  8  7  6
[26]  5  4  3  2  1 10  9  8  7  6  5  4  3  2  1 10  9  8  7  6  5  4  3  2  1
```

Another valid approach would be:

```
rep(10:1, times = 5)
```

```
 [1] 10  9  8  7  6  5  4  3  2  1 10  9  8  7  6  5  4  3  2  1 10  9  8  7  6
[26]  5  4  3  2  1 10  9  8  7  6  5  4  3  2  1 10  9  8  7  6  5  4  3  2  1
```

Question 3 (1 points)

Write an R command using the `%in%` operator that returns the elements of `x` that are fruits. That is, the elements of `x` that are contained somewhere in the vector `fruits`.

```
x <- c("apple", "banana", "carrot", "dill", "egg", "fig")
fruits <- c("apple", "banana", "fig", "grape", "lemon", "mango")
```

The code you provide in your answer does not need to include these above 2 lines .

The output of your command should be `[1] "apple"  "banana" "fig"`

Answer:

```
x[x %in% fruits]
```

```
[1] "apple"  "banana" "fig"
```

Question 4 (1 point)

The variable `x` is a numeric vector. Write an R command using the `ifelse()` function that returns a new vector with 1s whenever `x` is positive and 0s whenever `x` is not positive (0 or negative).

Defined mathematically, for the vector $x = (x_1, x_2, \dots, x_n)$, for each element x_i the output of your command should be:

$$\begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, 2, \dots, n$.

For example, if `x` is the vector `c(-2, 1, 0, -1, 2)`, then your command should return the vector $(0, 1, 0, 0, 1)$.

Answer:

`ifelse(x > 0, 1, 0)`

We can test this using the example provided:

```r
x <- c(-2, 1, 0, -1, 2)
ifelse(x > 0, 1, 0)
```

```
[1] 0 1 0 0 1
```

Data Analysis (7 points)

Download the dataset sales-data-nov-2025.csv. The dataset contains information on the total sales of a firm for different products in different regions where it has stores in November 2025. The variable descriptions are:

- `product_id`: The product ID (`A`, `B`, `C` and `D`).
- `region`: The province (`Noord-Brabant`, `Gelderland` or `Limburg`).
- `price`: The selling price of the product.
- `sales`: The total number of units sold of that product in that region in November 2025.

When reading the dataset into R, assign it to `df`.

Question 5 (1 point)

What is the median of the variable `sales`?

Provide both the numerical answer and the R command required to obtain the answer (if the dataframe is assigned to `df`).

Answer:

```r
df <- read.csv("sales-data-nov-2025.csv")
median(df$sales)
```

```
[1] 1026.5
```

Question 6 (2 points)

Part (a): Write an R command in the box below that creates a new variable in `df` called `revenue`, which is price multiplied by sales.

Answer:

```r
df$revenue <- df$price * df$sales
```

Part (b): Write an R command in the box below that returns the total revenue of the firm in Limburg (i.e. the sum of revenue for the 4 products in Limburg).

Answer:

```r
sum(df$revenue[df$region == "Limburg"])
```

```
[1] 1710910
```

Question 7 (2 points)

Create a scatter plot with `price` on the horizontal axis and `sales` on the vertical axis. Make the color of the points vary with the sales region.

Based on your plot, answer the following 2 questions.

Part (a): Choose the correct option from the following.

- Products with a higher price on average sell *fewer* units compared to products with a lower price.
- Products with a higher price on average sell *more* units compared to products with a lower price.
- Products with a higher price on average sell *roughly the same number* of units compared to products with a lower price.
- There is no clear relationship between price and sales in these data.

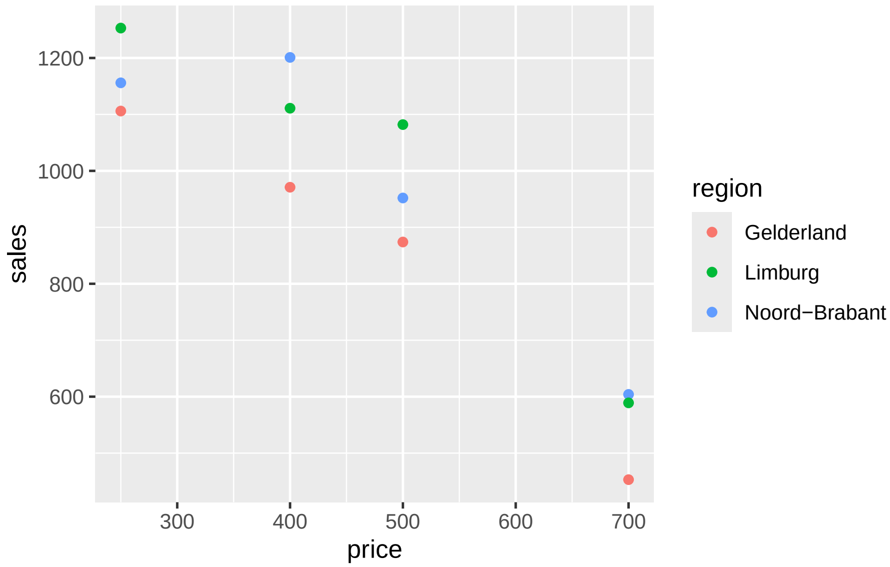
Part (b): Choose the correct option from the following.

Based on the colors of the points:

- For each product, *Gelderland* typically has lower sales than the other two regions.
- For each product, *Limburg* typically has lower sales than the other two regions.
- For each product, *Noord-Brabant* typically has lower sales than the other two regions.

Answer: We first make the plot:

```r
library(ggplot2)
ggplot(df, aes(price, sales, color = region)) + geom_point()
```



We can see that when price is high, the sales is lower. Therefore the correct option for the first question is: "Products with a higher price on average sell *fewer* units compared to products with a lower price."

Note that part (a) could be answered without making the color vary by region.

Looking at the colors of the points, we can see that the red dots corresponding to Gelderland is always lower down compared to the other two regions. Therefore the correct option for part (b) is "For each product, *Gelderland* typically has lower sales than the other two regions."

Question 8 (1 point)

Write an R command in the box below using the `aggregate()` function that returns the total number of units sold for each product.

Answer:

```
aggregate(sales ~ product_id, data = df, FUN = sum)
```

```
  product_id sales
1          A  2908
2          B  3283
3          C  3515
4          D  1646
```

Question 9 (1 point)

Using an appropriate function from the `reshape2` package, write an R command in the box below to reshape the data such that there are 4 rows, one for each product, and the columns are:

- The product ID.
- The sales for each product in Gelderland.
- The sales for each product in Limburg.
- The sales for each product in Noord-Brabant.

The output should be the following:

```
  product_id Gelderland Limburg Noord-Brabant
1          A        874    1082           952
2          B        971    1111          1201
3          C       1106    1253          1156
4          D        453     589           604
```

Hint: Load the `reshape2` package using the command `library(reshape2)`. You do not need to include loading this package in your answer.

Answer:

```r
library(reshape2)
dcast(df, product_id ~ region, value.var = "sales")
```

```
  product_id Gelderland Limburg Noord-Brabant
1          A        874    1082           952
2          B        971    1111          1201
3          C       1106    1253          1156
4          D        453     589           604
```

Data Cleaning (7 points)

Download the following two datasets:

- ice-cream-sales-2025.csv
- temperature-2025.csv

The first dataset contains the total number of ice cream scoops sold by an ice cream salesman in Tilburg in the summer of 2025 (May 15 to September 20). This salesman takes Mondays off every week, so it says `"Day off"` for the `sales` variable on Mondays.

The second dataset contains the daily high temperature in Tilburg on each date in Tilburg from May-September 2025.

When reading the datasets into R, assign `ice-cream-sales-2025.csv` to `df1` and `temperature-2025.csv` to `df2`.

Question 10 (6 points)

Perform the following cleaning steps:

Part (a): Write an R command in the box below using the `as.Date()` function that will correctly format the `date` variable in `df1` to an R date.

Answer:

```
df1 <- read.csv("ice-cream-sales-2025.csv")
df1$date <- as.Date(df1$date, format = "%d-%m-%y")
```

Part (b): Write an R command in the box below that will drop all Mondays from `df1`. After this command `df1` should have 111 rows.

Answer:

```
df1 <- df1[df1$day_of_week != "Mon", ]
```

Part (c): Write an R command in the box below that will convert the `sales` variable in `df1` to numeric.

Answer:

```
df1$sales <- as.numeric(df1$sales)
```

Part (d): Write an R command in the box below that will create a variable called `weekend` that is `TRUE` if the day of the week is Saturday or Sunday and `FALSE` otherwise.

Answer:

```
df1$weekend <- df1$day_of_week %in% c("Sat", "Sun")
```

Part (e): Write an R command in the box below that will convert the `date` variable in `df2` to an R date.

Answer:

```
df2 <- read.csv("temperature-2025.csv")
df2$date <- as.Date(df2$date, format = "%Y-%m-%d")
```

Part (f): Write an R command in the box below that will merge `df1` and `df2` by the `date` variable. Assign the output of this command to `df`.

Answer:

```
df <- merge(df1, df2, by = "date")
```

Question 11 (1 point)

If you performed the data cleaning steps from the previous exercise correctly, your final dataset should match the following file: temperature-sales-2025.csv

Using your cleaned dataset (or the linked file above), calculate the average sales on days that were *either* the weekend *or* had a temperature of at least 30 degrees.

Answer:

```
mean(df$sales[df$weekend | df$temp >= 30])
```

```
[1] 500.3488
```

Optimization (2 points)

The following 2 questions will involve working with the following mathematical function defined over all real numbers x:

$$f(x) = 10 - 4x + 2x^2$$

Question 12 (1 point)

Plot the function between the x values -3 and $+5$. Add the correct options in the boxes below that best describe the plot.

Part (a): The shape of the function is *a straight line / flat / U-shaped / inverse U-shaped*.

Part (b): When $f(x) = 40$, the corresponding values of x are _____ and _____.

Part (c): At $x = 3$, the function is *downward-sloping / flat / upward-sloping*.

Answer:

```
f <- function(x) {
  y <- 10 - 4*x + 2 * x^2
  return(y)
}
library(ggplot2)
x <- seq(-3, 5, length.out = 2000)
y <- f(x)
df <- data.frame(x, y)
ggplot(df, aes(x, y)) + geom_line()
```


```
# Part (a) The shape of the function is U-shaped.
# Part (b) The values of x when f(x)=40 are -3 and +5.
# Part (c) At x=3, the function is upward-sloping.
```

Question 13 (1 point)

Use R to find the value of x at an extreme point of this function.

Part (a) Type this value of x in the box below.

Answer:

```
f_max <- optimize(f, c(-100, 100), maximum = FALSE)
f_max$minimum
```

```
[1] 1
```

Part (b): What value does the function take at the extreme point?

Answer:

```
f_max$objective
```

```
[1] 8
```

```r
# or alternatively:
f(f_max$minimum)
```

```
[1] 8
```