

Test 1 Memo

Dr. Niladri Chakraborty

2024-04-15

Q1. Create two numeric vectors v1 and v2 with 50 random samples from the Normal distribution with mean 0 and variance 1. Compute their dot product (sum of the products of their corresponding elements).

[10]

Ans.

```
set.seed(123)
v1 <- rnorm(50)
v2 <- rnorm(50)

dot_product <- t(v1)%*%v2

dot_product

##           [,1]
## [1,] -1.221612
```

Q2. Draw a set of random samples of size 25 from exponential distribution with mean 1.

- (i) Create a **square matrix** M with elements with appropriately chosen number of rows and columns.
- (ii) Calculate its transpose, then compute the product of M and its transpose.

[5+5 = 10]

```
set.seed(123)
rand_sample <- rexp(25)

M <- matrix(rand_sample, nrow = 5)

M1 = t(M)

M1%*%M

##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 2.8144432 0.7289356 1.520402 2.500226 3.350357
## [2,] 0.7289356 7.6532294 1.543350 2.558545 4.495277
## [3,] 1.5204025 1.5433500 1.496927 2.722812 2.456817
```

```
## [4,] 2.5002257 2.5585451 2.722812 20.073933 8.456087
## [5,] 3.3503569 4.4952770 2.456817 8.456087 7.032537
```

Q3. Call the 'mpg' dataset.

- (i) Print the structure of the dataset and find how many variables and observations are there in the 'mpg' dataset?
- (ii) How many of the variables in the dataset are character strings and how many are integer valued variables?
- (iii) Now create a histogram of the 'displ' variable which is the engine displacement in liters. Add labels for the X and Y axis as 'Displacement' and 'Frequency', respectively. Also add a title for the histogram plot as 'Histogram of the engine displacement in liters'. The histogram bars should be filled with 'blue' color and the borders should be in 'black'. Use a 'classic' theme for the background.

[1+1+13 = 15]

Ans.

```
library(tidyverse)

## — Attaching core tidyverse packages — tidyverse
## 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2    3.4.4      ✓ tibble     3.2.1
## ✓ lubridate  1.9.3      ✓ tidyr      1.3.1
## ✓ purrr      1.0.2
## — Conflicts —
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors

library(hrbrthemes)

## Warning: package 'hrbrthemes' was built under R version 4.3.3

data <- mpg

# Structure of the data
str(data)

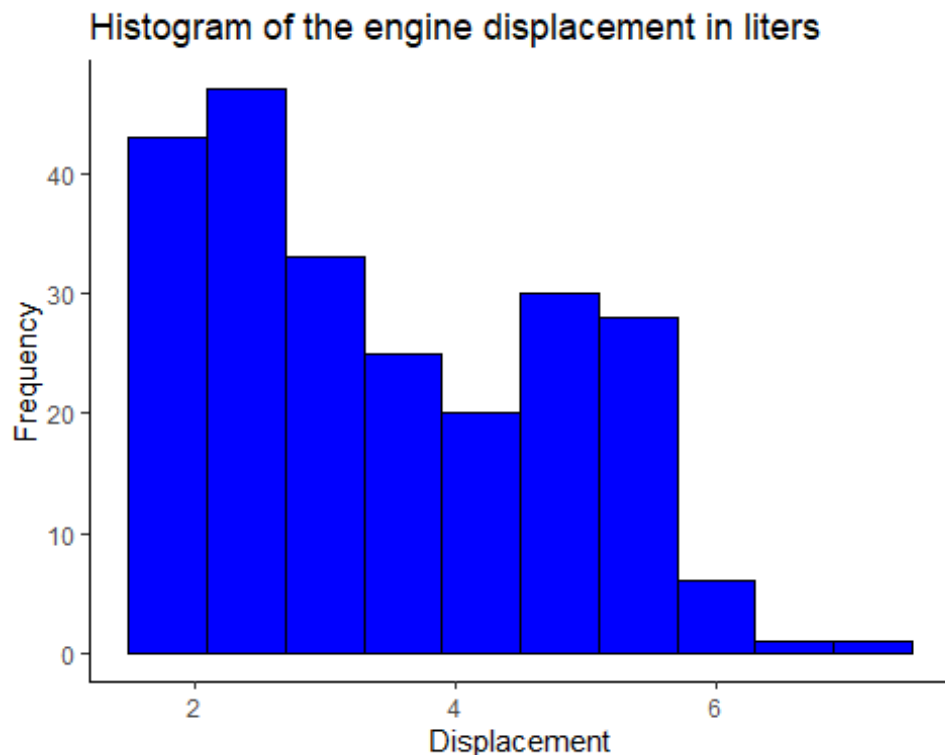
## tibble [234 × 11] (S3: tbl_df/tbl/data.frame)
## $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
## $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
## $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
```

```
## $ year      : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999
2008 ...
## $ cyl       : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
## $ trans     : chr [1:234] "auto(15)" "manual(m5)" "manual(m6)"
"auto(av)" ...
## $ drv       : chr [1:234] "f" "f" "f" "f" ...
## $ cty       : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
## $ hwy       : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
## $ fl        : chr [1:234] "p" "p" "p" "p" ...
## $ class     : chr [1:234] "compact" "compact" "compact" "compact" ...
```

We call the 'mpg' dataset from the 'tidyverse' package. In this dataset, there are 234 observations for 11 variables. Out of these 11 variables, 6 variables are 'character' strings and 4 variables are 'integer' valued variables.

#Next we create a histogram as suggested in the question for the engine displacement values in liters.

```
disp <- data$displ
# Plot histogram
ggplot(data, aes(x = disp)) +
  geom_histogram(bins = 10, fill = "blue", color = "black") +
  labs(title = "Histogram of the engine displacement in liters", x =
"Displacement", y = "Frequency")+ theme_classic()
```



Q4. Use the 'dplyr' package for the following analysis.

- (i) From the mpg dataset, create a subset of the dataset with the car models of corolla, altima, jetta, maxima, pathfinder 4wd, grandprix. Use the 'filter' function with suitable operator.
- (ii) Use this subset data to create a barplot of the models across the manufacturers. This will help you to visualize the distribution of car models within their respective manufacturers. Give a title for the barplot as 'Count of car models by manufacturer'. Also provide labels for the X and Y axis as 'Manufacturer' and 'Count', respectively. Use a 'minimal' theme for the plot.
- (iii) Which manufacturer has more variations in car models that you find from the barplot?
- (iv) Which manufacturer produces 'grand prix' cars?

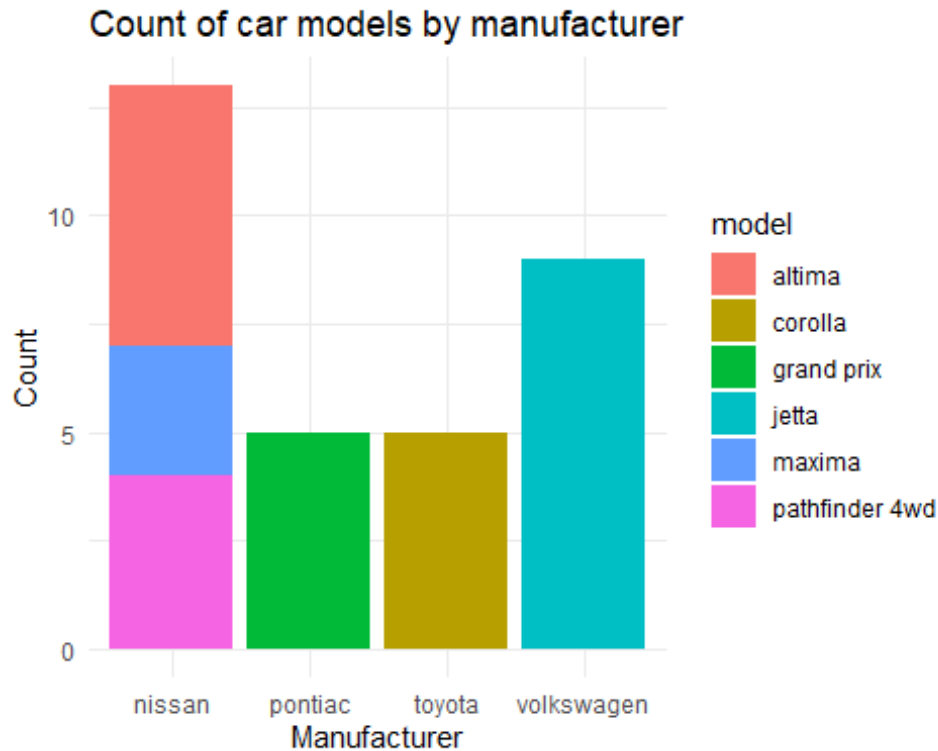
[5+13+1+1 = 20]

Ans.

```
library(dplyr)

# Create a subset of the dataset for specified models
selected_models <- c("corolla", "altima", "jetta", "maxima", "pathfinder
4wd", "grand prix")
subset_data <- mpg %>%
  filter(model %in% selected_models)

# Plot boxplot
ggplot(subset_data) +
  geom_bar(aes(x = manufacturer, fill = model)) +
  labs(title = "Count of car models by manufacturer", x = "Manufacturer", y =
"Count") +
  theme_minimal()
```



*##From the plot, we find 'nissan' has the highest variety of car models.
The 'grand prix' cars are produced by 'pontiac'.*

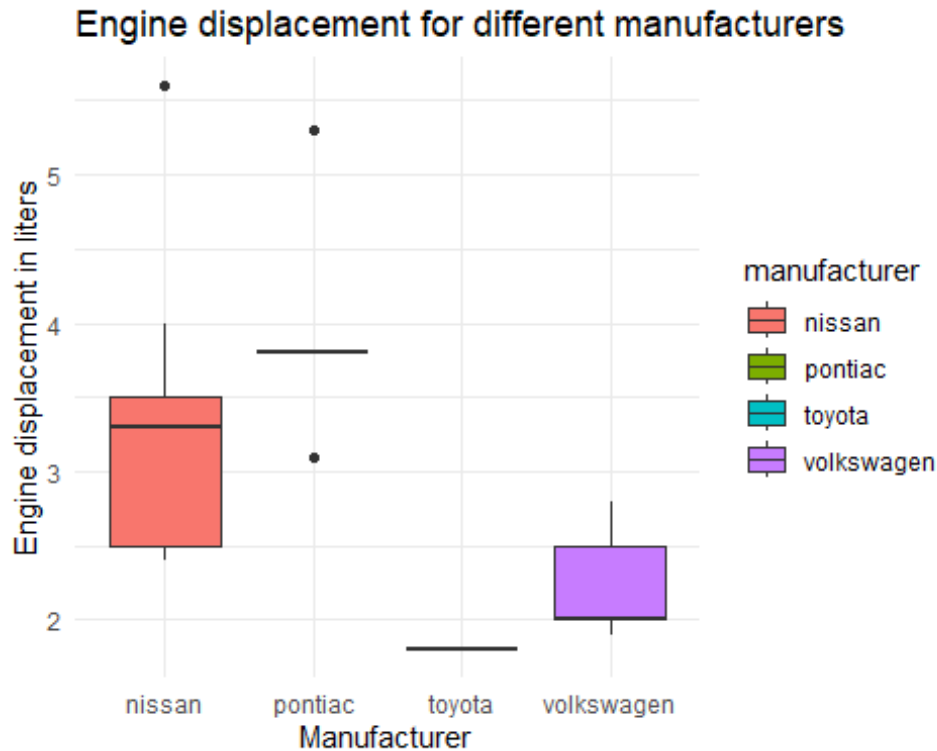
Q5. Now use the same subset of the 'mpg' data,

Then create a boxplot for the 'displ' variable (Engine displacement in liters) for different manufacturers. All the boxplots should be in the same plot. Use different colors for different manufacturers. Provide a title of the boxplot as 'Engine displacement for different manufacturers'. Provide X and Y axis labels as 'Manufacturer' and 'Engine displacement in liters', respectively. Use the minimal theme for the plot.

For which manufacturer, the engine displacement values are negatively skewed?

[14+1 = 15]

```
# Plot boxplot
ggplot(subset_data, aes(x = manufacturer, y = displ, fill = manufacturer)) +
  geom_boxplot() +
  labs(title = "Engine displacement for different manufacturers", x =
"Manufacturer", y = "Engine displacement in liters") +
  theme_minimal()
```



For 'nissan', the engine displacement values are negatively skewed.

Q6. Use the 'airquality' dataset. Create a dataframe with the variables 'Ozone', 'Solar.R', 'Temp' from the dataset. Now create a correlation plot for these variables using the GGally package. Interpret the correlation plot.

[8+2 = 10]

```
library(GGally)

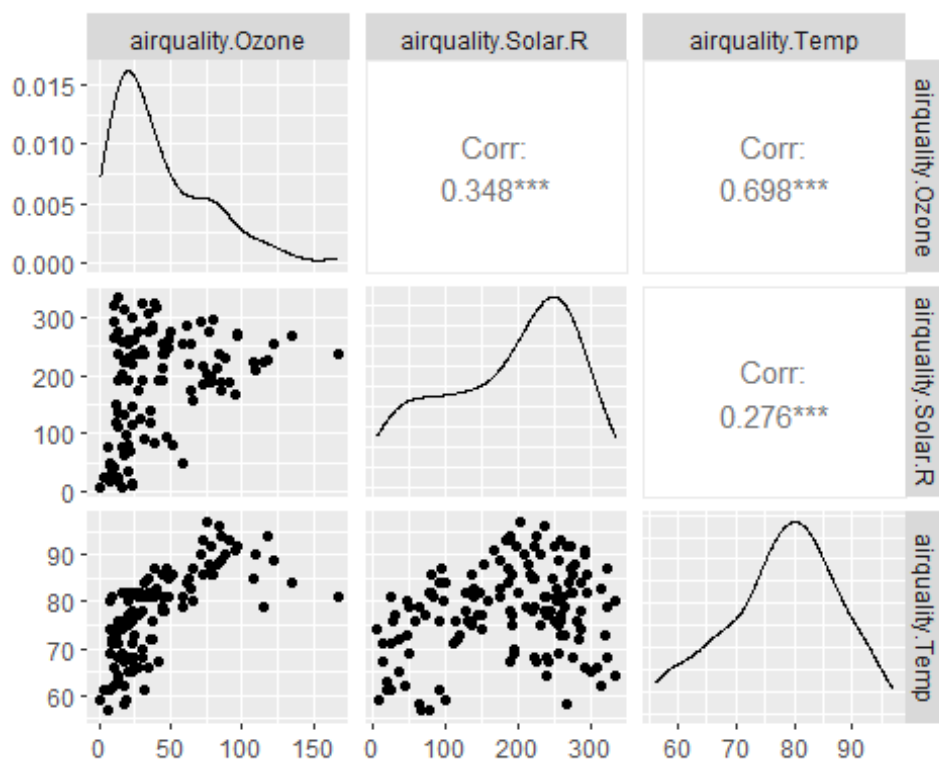
## Warning: package 'GGally' was built under R version 4.3.3

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg      ggplot2

data("airquality")
data = data.frame(airquality$Ozone,airquality$Solar.R,airquality$Temp)
ggpairs(data)

## Warning: Removed 37 rows containing non-finite values (`stat_density()`).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm,
## :
## Removed 42 rows containing missing values
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm,
:
## Removed 37 rows containing missing values
## Warning: Removed 42 rows containing missing values (`geom_point()`).
## Warning: Removed 7 rows containing non-finite values (`stat_density()`).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm,
:
## Removed 7 rows containing missing values
## Warning: Removed 37 rows containing missing values (`geom_point()`).
## Warning: Removed 7 rows containing missing values (`geom_point()`).
```



Strong positive correlations are evident among all the variables, suggesting that an increase in one variable tends to be associated with an increase in the others. The distribution of the Ozone levels shows a positive skew, while the distributions of the other two variables exhibit negative skews.

Q7.

Write a code using a 'for' loop to compute the sum of all even numbers from 1 to 100.

```

# Initialize the sum variable
sum_evens <- 0

# For Loop to sum even numbers from 1 to 100
for (i in 1:100) {
  if (i %% 2 == 0) { # Check if the number is even
    sum_evens <- sum_evens + i
  }
}

sum_evens

## [1] 2550

```

Q8.

Write a code using a 'while' loop to calculate the smallest multiple of 13 that is greater than 100.

[10]

```

# Initialize the starting number
number <- 101

# While loop to find the smallest multiple of 13 greater than 100
while (number %% 13 != 0) { # Check if the current number is not a multiple
of 13
  number <- number + 1
}
number

## [1] 104

```