Practice Exam

Colin Yip, Student No. 3953629

2023-12-14

Exercise 1

Q1

```
car_df <- read.csv("cartype.csv")

head(car_df)
```

```
##    Entity Code Year battery_electric_number petrol_number diesel_gas_number
## 1 Austria  AUT 2001                       0        100754            192734
## 2 Austria  AUT 2002                       1         84920            194555
## 3 Austria  AUT 2003                       0         85889            214222
## 4 Austria  AUT 2004                       0         96388            214771
## 5 Austria  AUT 2005                       0        119632            187813
## 6 Austria  AUT 2006                       0        116237            191773
##   hybrid_number
## 1            39
## 2            17
## 3             8
## 4           133
## 5           460
## 6           585
```

The data is in a long format, as it is year over year per row.

Q2

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
target_countries <- c("France", "Germany", "Italy", "Spain", "United Kingdom")
filtered_car_df <- dplyr::filter(car_df, Entity %in% target_countries)

head(filtered_car_df)
```

```
##   Entity Code Year battery_electric_number petrol_number diesel_gas_number
## 1 France  FRA 2001                     407        986491           1267750
## 2 France  FRA 2002                     233        793425           1351362
## 3 France  FRA 2003                     113        655678           1353419
## 4 France  FRA 2004                     460        619675           1392905
## 5 France  FRA 2005                       6        641022           1423906
## 6 France  FRA 2006                      13        565984           1428153
##   hybrid_number
## 1            84
## 2            51
## 3            36
## 4           669
## 5          2855
## 6          6413
```

Q3

```r
filtered_car_df$total <- filtered_car_df$battery_electric_number +
  filtered_car_df$petrol_number +
  filtered_car_df$diesel_gas_number +
  filtered_car_df$hybrid_number
total_per_year <- filtered_car_df %>%
  group_by(Year) %>%
  summarize(all_numbers = sum(total))
total_per_type_per_year <- filtered_car_df %>%
  group_by(Year) %>%
  summarize(
    batt = sum(battery_electric_number),
    petrol = sum(petrol_number),
    diesel = sum(diesel_gas_number),
    hybrid = sum(hybrid_number)
  )
batt_prop <- cbind(
  total_per_year$Year,
  total_per_type_per_year$batt / total_per_year$all_numbers
)
petrol_prop <- cbind(
  total_per_year$Year,
  total_per_type_per_year$petrol / total_per_year$all_numbers
)
diesel_prop <- cbind(
  total_per_year$Year,
  total_per_type_per_year$diesel / total_per_year$all_numbers
)
hybrid_prop <- cbind(
  total_per_year$Year,
  total_per_type_per_year$hybrid / total_per_year$all_numbers
)
```

Q4

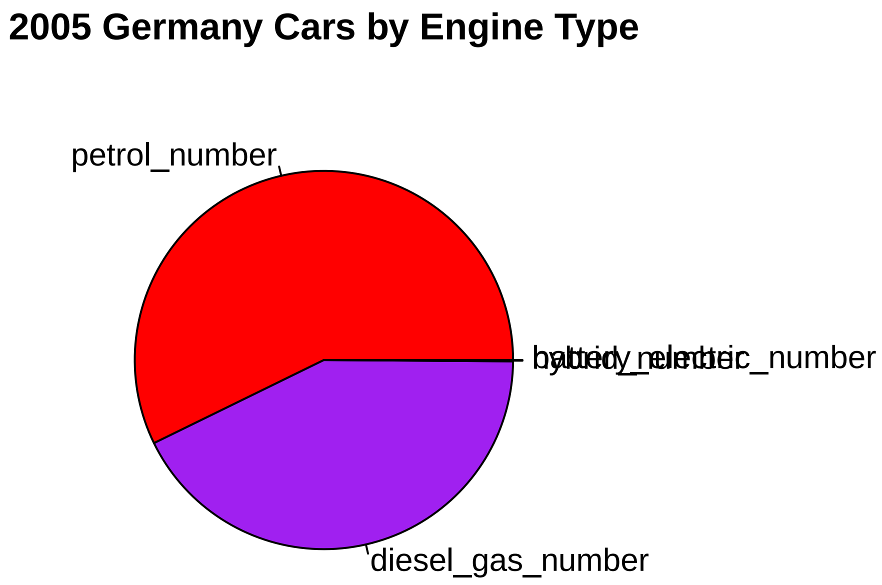
```r
filtered_cars_2008 <- filter(filtered_car_df, Year == 2008)
max_diesel_idx <- which.max(filtered_cars_2008$diesel_gas_number)
max_diesel_2008 <- filtered_cars_2008[max_diesel_idx, "Entity"]

filtered_cars_2018 <- filter(filtered_car_df, Year == 2018)
max_diesel_idx <- which.max(filtered_cars_2018$diesel_gas_number)
max_diesel_2018 <- filtered_cars_2018[max_diesel_idx, "Entity"]
```

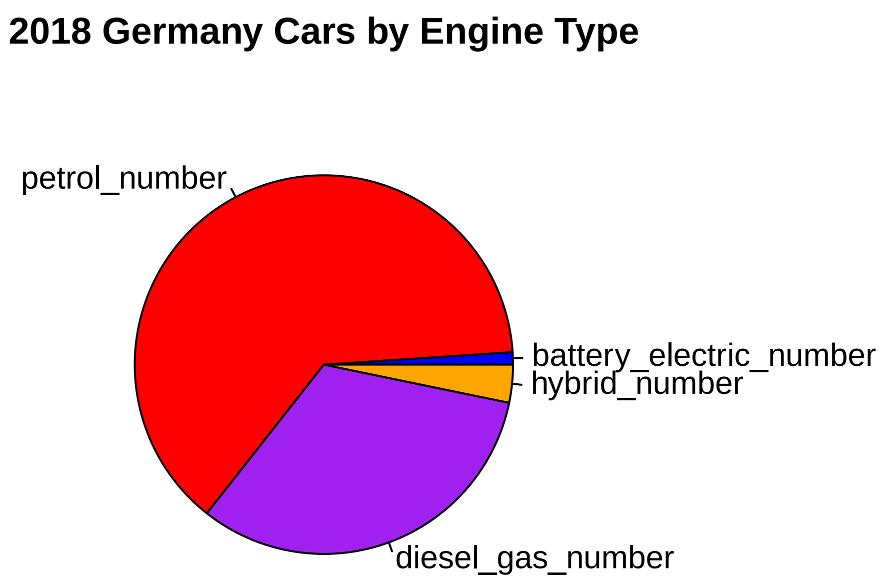
Maximum diesel car country in 2008 was France and maximum diesel car country in 2018 was Germany.

Q5

```r
filtered_cars_2005 <- filter(filtered_car_df, Year == 2005, Entity == "Germany")
engine_type_names <- c(
  "battery_electric_number", "petrol_number",
  "diesel_gas_number", "hybrid_number"
)
pie(as.numeric(filtered_cars_2005[, engine_type_names]),
  col = c("blue", "red", "purple", "orange"),
  main = "2005 Germany Cars by Engine Type",
  labels = engine_type_names
)
```



```r
filtered_cars_2018 <- filter(filtered_car_df, Year == 2018, Entity == "Germany")
engine_type_names <- c(
  "battery_electric_number", "petrol_number",
  "diesel_gas_number", "hybrid_number"
)
pie(as.numeric(filtered_cars_2018[, engine_type_names]),
  col = c("blue", "red", "purple", "orange"),
  main = "2018 Germany Cars by Engine Type",
  labels = engine_type_names
)
```



The proportion of diesel and petrol cars has reduced, with the proportion of hybrid and electric cars increasing.

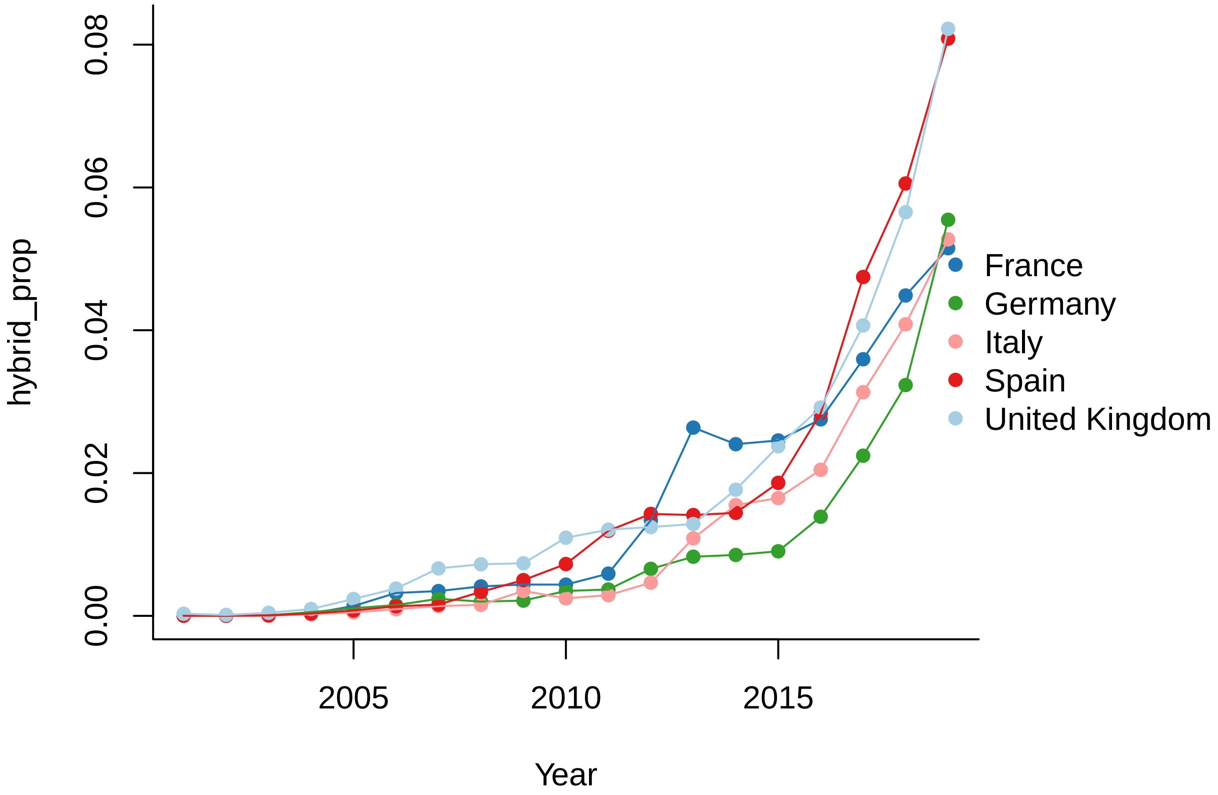
Q6

```
# Group car numbers by country
hybrid_car_by_country <- filtered_car_df %>%
  group_by(Year, Entity) %>%
  summarize(
    hybrid_number = hybrid_number,
    total_cars = sum(
      battery_electric_number,
      petrol_number,
      diesel_gas_number,
      hybrid_number
    )
  )
```

```
## ‘summarise()‘ has grouped output by ’Year’. You can override using the
## ‘.groups‘ argument.
```

```
# Get proportion by country, year
hybrid_car_by_country$hybrid_prop <- hybrid_car_by_country$hybrid_number /
  hybrid_car_by_country$total_cars

library(ptmixed)
make.spaghetti(Year, hybrid_prop,
  group = Entity, id = Entity,
  data = data.frame(hybrid_car_by_country)
)
```



Spain and UK have fastest diffusion of hybrids.

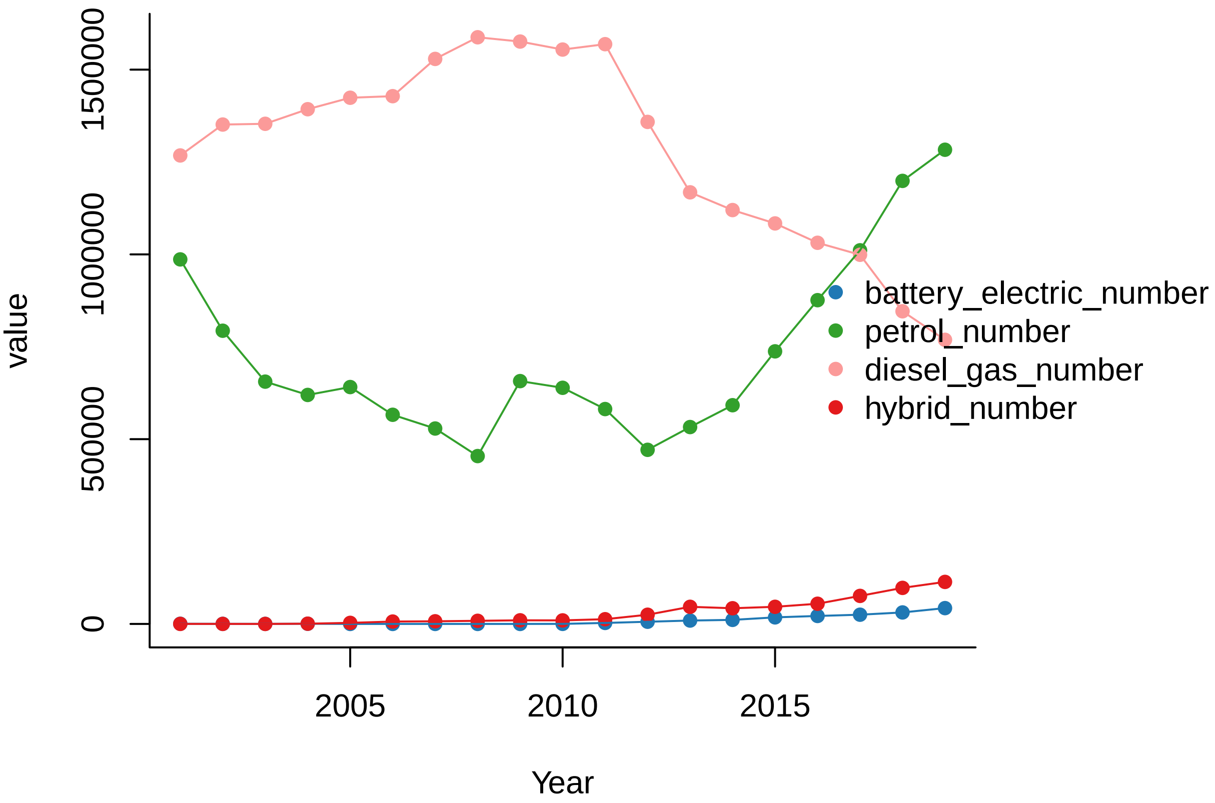
Q7

```
library(reshape2)
france_data <- filter(filtered_car_df, Entity == "France")
france_car_types <- france_data %>%
  group_by(Entity, Year) %>%
  summarise(
    battery_electric_number = battery_electric_number,
    petrol_number = petrol_number,
    diesel_gas_number = diesel_gas_number,
    hybrid_number = hybrid_number
  )
```

```
## ‘summarise()‘ has grouped output by ’Entity’. You can override using the
## ‘.groups‘ argument.
```

```
melted_france_car_types <- melt(france_car_types,
  id.vars = c("Entity", "Year")
)

make.spaghetti(Year, value,
  group = variable, id = variable,
  data = melted_france_car_types
)
```



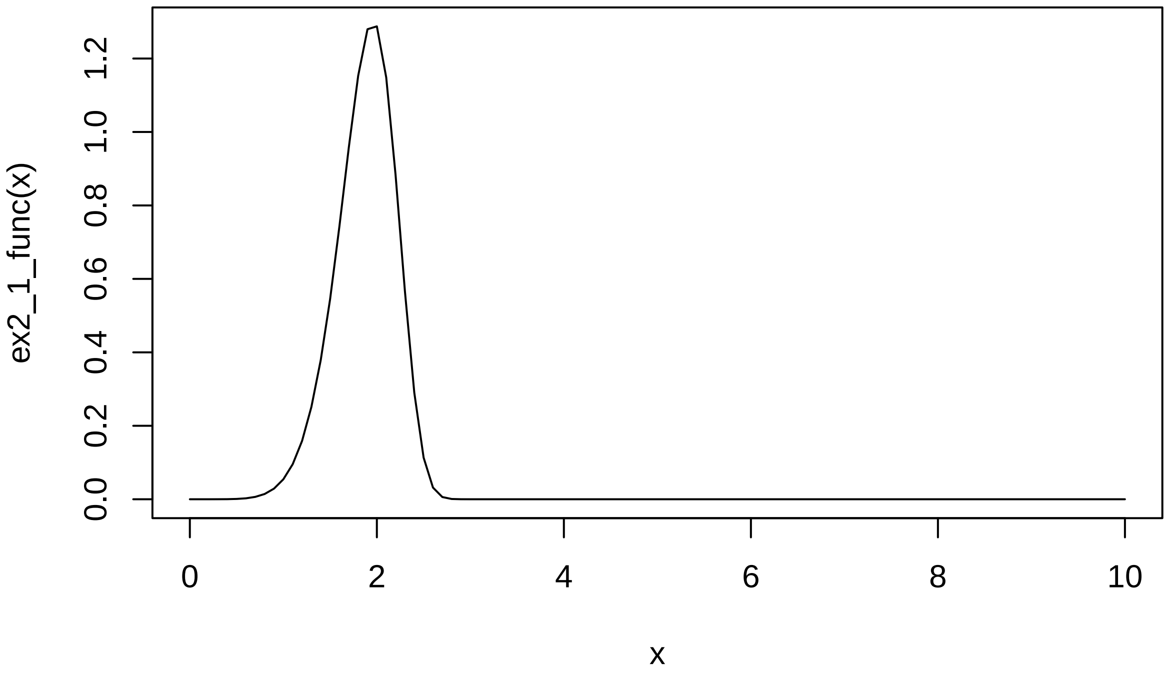
Exercise 2

Q1

```
k <- 7
lambda <- 2

ex2_1_func <- function(x) {
  return((k / lambda) * ((x / lambda)^(k - 1)) * (exp(-((x / lambda)^k))))
}

x_range <- seq(0, 10, 0.1)
curve(ex2_1_func, from = 0, to = 10)
```



Q2

```
k <- 3
lambda <- 4
```

Exercise 3

Q1

```
getSummaries <- function(x) {
  n <- length(x)
  mean_vec <- mean(x)
  median_vec <- median(x)
  variance_vec <- var(x)

  values <- c(n, mean_vec, median_vec, variance_vec)

  statistic_names <- c(
    "Sample size",
    "Mean",
    "Median",
    "Variance"
  )

  output_df <- data.frame(cbind(statistic_names, values))
  names(output_df) <- c("Statistic", "Value")
  knitr::kable(output_df)
}

some_vec <- 1:10
```

```r
getSummaries(some_vec)
```

Statistic	Value
Sample size	10
Mean	5.5
Median	5.5
Variance	9.16666666666667

Q2

```r
getSummaries <- function(x, digits) {
  n <- length(x)
  mean_vec <- round(mean(x), digits)
  median_vec <- round(median(x), digits)
  variance_vec <- round(var(x), digits)

  values <- c(n, mean_vec, median_vec, variance_vec)

  statistic_names <- c(
    "Sample size",
    "Mean",
    "Median",
    "Variance"
  )

  output_df <- data.frame(cbind(statistic_names, values))
  names(output_df) <- c("Statistic", "Value")
  knitr::kable(output_df)
}

getSummaries(some_vec, 2)
```

Statistic	Value
Sample size	10
Mean	5.5
Median	5.5
Variance	9.17

Q3

```r
getSummaries <- function(x, digits, ignoreNAs) {
  n <- length(x)
  if (NA %in% x & !ignoreNAs) {
    warning("There's NAs in x. Returning only n.")
    output_df <- cbind(c("Sample size"), c(n))
  } else if (NA %in% x & ignoreNAs) {
    mean_vec <- round(mean(x, na.rm = T), digits)
    median_vec <- round(median(x, na.rm = T), digits)
    variance_vec <- round(var(x, na.rm = T), digits)

    values <- c(n, mean_vec, median_vec, variance_vec)

    statistic_names <- c(
      "Sample size",
      "Mean",
      "Median",
      "Variance"
    )

    output_df <- data.frame(cbind(statistic_names, values))
  } else {
    mean_vec <- round(mean(x), digits)
    median_vec <- round(median(x), digits)
    variance_vec <- round(var(x), digits)

    values <- c(n, mean_vec, median_vec, variance_vec)

    statistic_names <- c(
      "Sample size",
      "Mean",
      "Median",
      "Variance"
    )

    output_df <- data.frame(cbind(statistic_names, values))
  }
  names(output_df) <- c("Statistic", "Value")
  knitr::kable(output_df)
}

vec_with_na <- c(1:5, NA, 7:10)
getSummaries(vec_with_na, 2, F)
```

```
## Warning in getSummaries(vec_with_na, 2, F): There's NAs in x. Returning only n.
```

Sample size	10

Q4

```
getSummariesPerVec <- function(x, digits, ignoreNAs) {
  n <- length(x)
  if (NA %in% x & !ignoreNAs) {
    warning("There's NAs in x. Returning only n.")
    output_df <- cbind(c("Sample size"), c(n))
  } else if (NA %in% x & ignoreNAs) {
    mean_vec <- round(mean(x, na.rm = T), digits)
    median_vec <- round(median(x, na.rm = T), digits)
    variance_vec <- round(var(x, na.rm = T), digits)

    values <- c(n, mean_vec, median_vec, variance_vec)

    statistic_names <- c(
      "Sample size",
      "Mean",
      "Median",
      "Variance"
    )

    output_df <- data.frame(cbind(statistic_names, values))
  } else {
    mean_vec <- round(mean(x), digits)
    median_vec <- round(median(x), digits)
    variance_vec <- round(var(x), digits)
```

```r
    values <- c(n, mean_vec, median_vec, variance_vec)

    statistic_names <- c(
      "Sample size",
      "Mean",
      "Median",
      "Variance"
    )

    output_df <- data.frame(cbind(statistic_names, values))
  }
  names(output_df) <- c("Statistic", "Value")
  knitr::kable(output_df)
}

getSummaries <- function(x, digits, ignoreNAs) {
  if (is.matrix(x)) {
    x_ncol <- ncol(x)
    x_ncol_range <- 1:x_ncol
    apply(x, 2, getSummariesPerVec, digits, ignoreNAs)
  } else if (is.vector(x)) {
    getSummariesPerVec(x, digits, ignoreNAs)
  }
}

some_matrix <- matrix(1:4, nrow = 2, ncol = 2)
getSummaries(some_matrix, 2, T)
```

```
##      [,1]                   [,2]
## [1,] "|Statistic   |Value |" "|Statistic   |Value |"
## [2,] "|:-----------|:-----|" "|:-----------|:-----|"
## [3,] "|Sample size |2     |" "|Sample size |2     |"
## [4,] "|Mean        |1.5   |" "|Mean        |3.5   |"
## [5,] "|Median      |1.5   |" "|Median      |3.5   |"
## [6,] "|Variance    |0.5   |" "|Variance    |0.5   |"
```

Q5

```r
getSummaries <- function(x, digits, ignoreNAs) {
  if (is.matrix(x)) {
    x_ncol <- ncol(x)
    x_ncol_range <- 1:x_ncol
    apply(x, 2, getSummariesPerVec, digits, ignoreNAs)
  } else if (is.vector(x) & is.numeric(x)) {
    getSummariesPerVec(x, digits, ignoreNAs)
  } else {
    stop("Neither matrix nor vector passed.")
  }
}
```

Q6

```r
set.seed(3078)
x1 <- rhyper(300, 5, 20, 10)
x2 <- c(rep(cars$speed, 3), rep(NA, 50), rep(cars$dist, 2))
x3 <- cbind(x1, x2)

sapply(list(x1, x2, x3), getSummaries, digits = 2, ignoreNAs = T)
```

```
## [[1]]
##
##
## |Statistic   |Value |
## |:-----------|:-----|
## |Sample size |300   |
## |Mean        |1.96  |
## |Median      |2     |
## |Variance    |1.01  |
##
## [[2]]
##
##
## |Statistic   |Value  |
## |:-----------|:------|
## |Sample size |300    |
## |Mean        |26.43  |
## |Median      |19     |
## |Variance    |461.15 |
##
## [[3]]
##      x1                        x2
## [1,] "|Statistic   |Value |" "|Statistic   |Value  |"
## [2,] "|:-----------|:-----|" "|:-----------|:------|"
## [3,] "|Sample size |300   |" "|Sample size |300    |"
## [4,] "|Mean        |1.96  |" "|Mean        |26.43  |"
## [5,] "|Median      |2     |" "|Median      |19     |"
## [6,] "|Variance    |1.01  |" "|Variance    |461.15 |"
```

Q7

```r
sapply(list(x1, x2, x3), getSummaries, digits = 4, ignoreNAs = F)
```

```
## Warning in getSummariesPerVec(x, digits, ignoreNAs): There's NAs in x.
## Returning only n.
```

```
## Warning in FUN(newX[, i], ...): There's NAs in x. Returning only n.
```

```
## [[1]]
##
##
## |Statistic   |Value  |
## |:-----------|:------|
## |Sample size |300    |
## |Mean        |1.9633 |
## |Median      |2      |
## |Variance    |1.0053 |
##
## [[2]]
##
##
## |            |    |
## |:-----------|:---|
## |Sample size |300 |
##
## [[3]]
## [[3]]$x1
##
##
## |Statistic   |Value  |
## |:-----------|:------|
## |Sample size |300    |
## |Mean        |1.9633 |
## |Median      |2      |
## |Variance    |1.0053 |
##
## [[3]]$x2
##
##
## |            |    |
## |:-----------|:---|
## |Sample size |300 |
```