STOCKHOLM UNIVERSITY
Department of Statistics
Ellinor Fackle Fornius

EXAM IN R PROGRAMMING
November 25, 2022

Time: 8.00-13.00

Results will be announced no later than December 16.

At the start of the exam, go to the web page http://tenta.stat.su.se/tenta.html and select the option 'Download SEB-config File for R programming'. Run the configuration file to start the Safe Examination Browser (SEB). At the homepage in SEB you find the option to download your supporting file. Fill out your name and your anonymous code to get access to your file.

The exam should be written as an R Markdown report. You should submit the R markdown file (.Rmd) as well as an output file in .html format. You are free to use either R Base packages or Tidyverse packages to solve the tasks, unless stated otherwise in the instructions.

Notify the computer exam staff when you are ready to hand in your exam. Before you submit your exam, check that you have NOT written your name anywhere in your files (which would break the anonymity of the exam). Upload your exam files at the homepage in SEB.

Note: Don't forget to save often! In the event of R crashing, any unsaved changes could be lost.

Task 1. Load the `mtcars` data and solve the following tasks.

a) Which is the heaviest car in the data set?

b) Change the class of the variables `am`, `cyl` and `vs` to integer and store the new dataset as `newcars`.

c) Round the `newcars` data to one digit.

d) Subset the rows of cars that get less than 16 miles per gallon (`mpg`) and have more than 100 horsepower (`hp`). How many such cars are there?

e) Convert all the column names of `newcars` to upper case, e.g convert `mpg` to `MPG`.

Task 2. A party is attended by 33 people. This task aims to answer the question: How likely is it that at least two of these 33 individuals share a birthday? Use `set.seed(321)` for random number generation in this task.

a) Assume there are no leap days, that all years are 365 days, and that births are uniformly distributed over the year. Take a sample of birthdays for the 33 individuals and check if any of these share birthdays.

b) Repeat the simulation in a) 10 000 times and use it to estimate the probability that at least two individuals at the party share the same birthday.

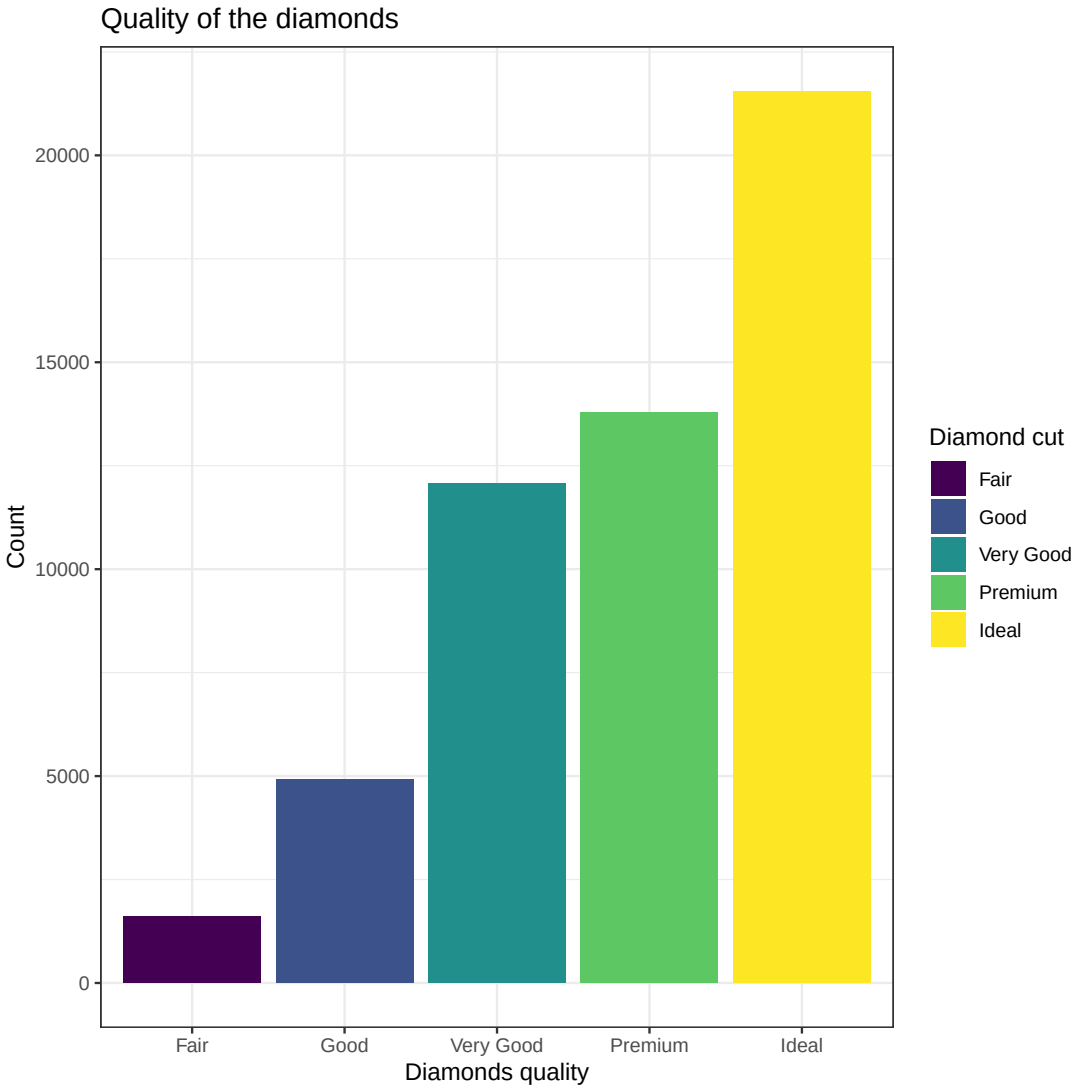
Task 3. Take the English alphabet (a, b, c, ..., x, y, z) and create a `for` loop that prints the output as given below. For every iteration, print the number of vowels (a, e, i, o, u, y) among the first `i` letters, where `i` is the loop index, as well as the first `i` letters. The first 10 iterations of the loop are shown here.

```
[1] "Number of vowels: 1 , a"
[1] "Number of vowels: 1 , a b"
[1] "Number of vowels: 1 , a b c"
[1] "Number of vowels: 1 , a b c d"
[1] "Number of vowels: 2 , a b c d e"
[1] "Number of vowels: 2 , a b c d e f"
[1] "Number of vowels: 2 , a b c d e f g"
[1] "Number of vowels: 2 , a b c d e f g h"
[1] "Number of vowels: 3 , a b c d e f g h i"
[1] "Number of vowels: 3 , a b c d e f g h i j"
```

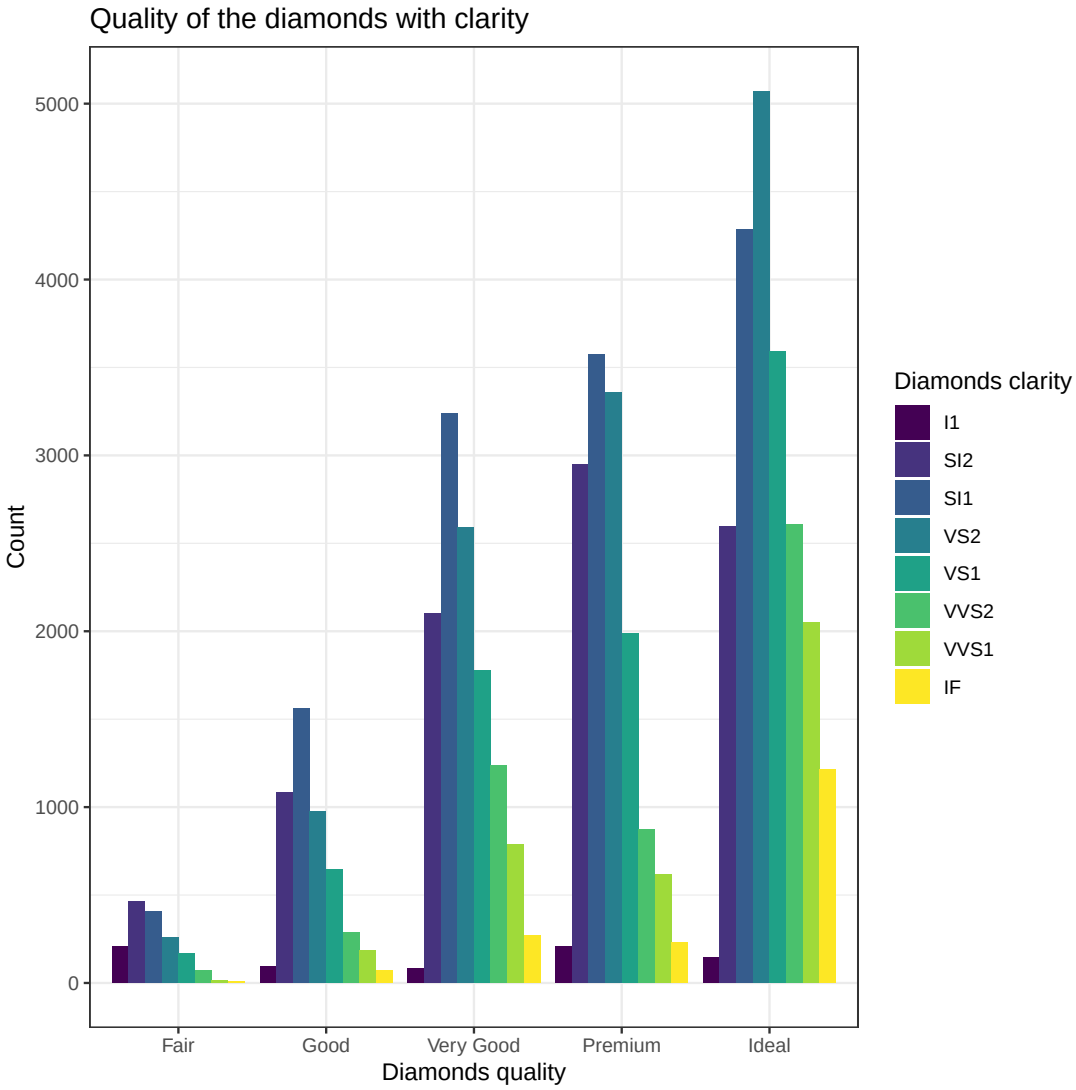
Task 4. Load the `diamonds` data set that comes with the `ggplot2` package. Reproduce the output/graph in the following sub-tasks. The displayed outputs are created with Tidyverse packages, but you may also use base R functions to produce similar corresponding outputs.

a)
```
# A tibble: 5 x 4
  cut       max_price min_price median_price
  <ord>         <int>     <int>        <dbl>
1 Fair          18574       337        3282
2 Good          18788       327        3050.
3 Very Good     18818       336        2648
4 Premium       18823       326        3185
5 Ideal         18806       326        1810
```

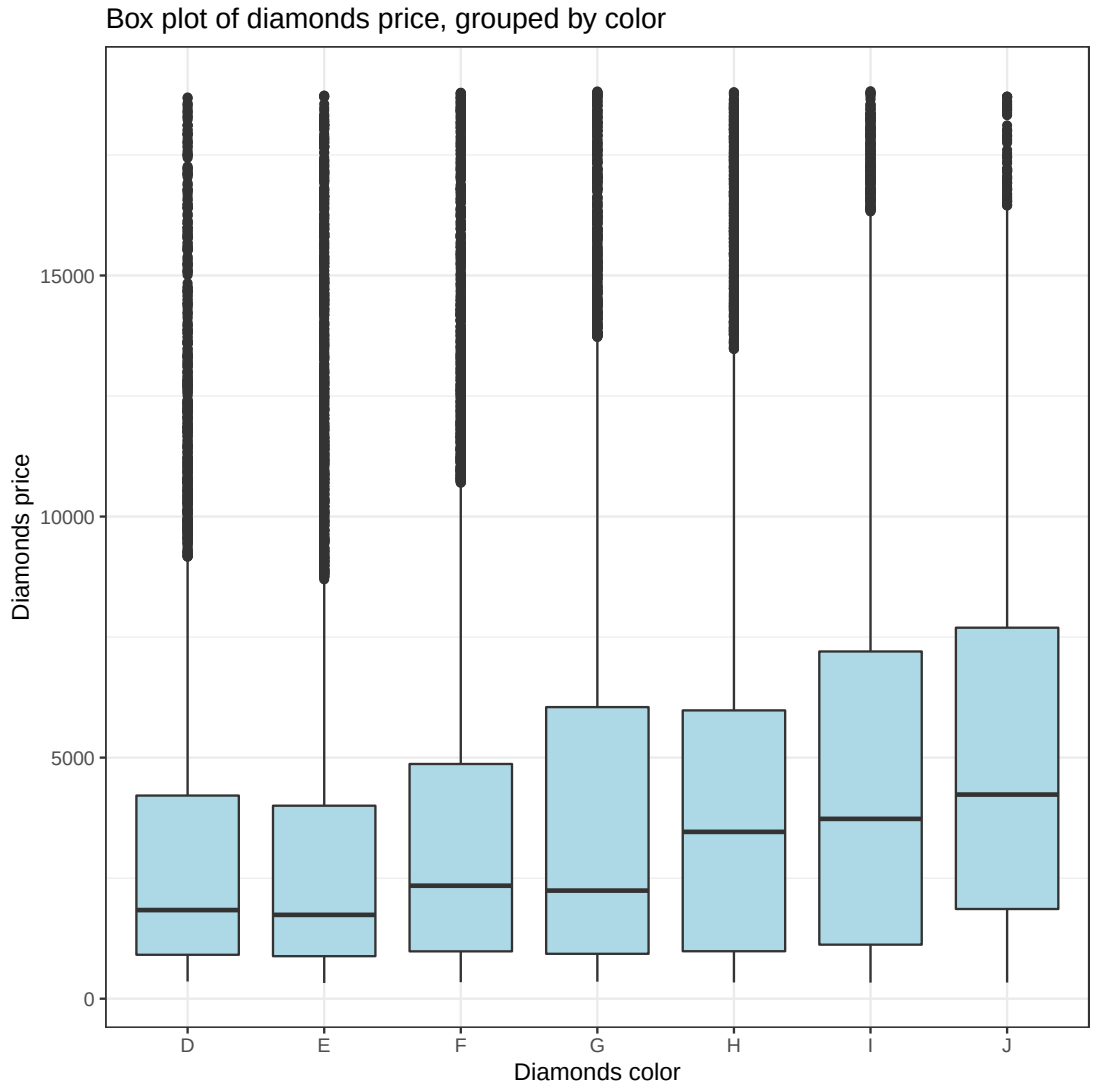
b)


Quality of the diamonds
Count
20000
15000
10000
5000
0
Fair
Good
Very Good
Premium
Ideal
Diamonds quality
Diamond cut
Fair
Good
Very Good
Premium
Ideal

c)


Quality of the diamonds with clarity
Count
5000
4000
3000
2000
1000
0
Fair
Good
Very Good
Premium
Ideal
Diamonds quality
Diamonds clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

d)



Task 5. Bootstrapping is a technique that can be used for estimation of the sampling distribution of many types of statistics, such as the sample variance. It uses random sampling with replacement to mimic the sampling process. Use `set.seed(321)` for random number generation in this task.

a) Create a function called `myvar` that computes the sample variance of a numeric vector according to the formula below. Do not use any built-in variance, standard deviation or mean functions.

$$\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

b) Use your function to calculate the variance for the following duration data.

```
durations <-
c(35.8, 33.4, 34.9, 17.9, 35.6, 10.3, 14.9, 28.3, 39.2, 25.4, 23.4, 7.1, 38.9, 9.2, 8.1)
```

c) Take one sample with replacement from the `durations` data having the same number of observations as `durations`. Here is one realization of such a sample

```
 [1] 33.4 25.4 14.9 23.4  9.2 35.6 35.6  7.1 33.4 28.3 23.4 33.4 38.9 33.4 39.2
```

d) Repeat this sampling with replacement 1000 times and calculate the variance in the sample at each iteration. These are your 1000 bootstrapped samples and variances.

e) We can get a 95% bootstrap percentile confidence interval from the 2.5% and 97.5% percentiles of the 1000 bootstrapped variances. Calculate the 95 % bootstrap percentile confidence interval.

Task 6. Describe the following pairs of functions (in your own words) and comment on the similarities/differences between them.

- `lapply()` and `do.call()`
- `mclapply()` and `parLapply()`

Good Luck!