Assignment 1

Exercise A: "Coin flipping"

The most common introductory exercises for the demonstration of the Bernoulli distribution are the coin flipping exercises, where usually the task is to determine the probability of k heads (or tails) in an n long sequence produced by an unbiased or biased coin . These tasks are usually very easy to solve. However one can construct some non-trivial exercises too with the coins and sometimes even it is easier to simulate the task to determine the probabilities then to find an analytical formula.

Let's consider a 50 long sequence produced by an unbiased coin (the probability of either the head or the tail is 50%). The task: **Determine the probability that a 6 long run of the same coin sides (heads or tails) appears in a 50 long sequence with simulation.**

In the `homeworkA.R` file you can find an almost ready code, which simulates the above task. It generates an n long binary (0,1) sequence 1000 times. (We can assign the heads to the 1s.) For each random sequence it iterates on the sequence's elements, and counts with two counters (`counterH` for heads and `counterT` for tails) the lengths of head and tail runs. If one of the counters reach the value of 6, the code changes the corresponding element of the `hit_v` vector from 0 to 1 and quits from the actual `for` loop (the inner `for` loop) with the `break` command. After the outer `for` loop finishes the iteration the `hit_v` vector will contain 0s for the sequences where there were no 6 long (or even longer) runs, and it will contain 1s otherwise. We can estimate the probability of the 6 long runs in 50 long sequences as the proportion of the simulated sequences which contains at least one 6 long (or longer) run. The estimated probability is stored in the list `p`. (You can find more info about the code in the comments.)

Your task:

1. Understand the code.
2. Rewrite the value assignment for the list `p` (line 26) because it stores now the sum of the elements in `hit_v` vector, not the estimated probability.
3. Determine the estimated probability for a biased coin too ($\mathbb{P}(HEAD) = 0.7$ (prob. of head is 70%)) and store the estimated probability in an other element of the list `p` called "ub".

   Use this:

   ```
   for ( k in c(0.5, 0.7) ){…
   ```
   #for $\mathbb{P}(HEAD) = 0.5$ and $\mathbb{P}(HEAD) = 0.7$

   and this:

   ```
   if …
      p["ub"] = …
   else
      p["b"] = …
   ```
4. Print out the results with the print() and toString() functions:

   ```
   print( paste('p_ub: ',     ......   ) , quote = F)
   ```

```
print( paste('p_b: ',     ......   ) , quote = F)
```

hint: put `p[['ub']]` and `p[['b']]` as a string at the dotted lines

5. There is a bag with 10 coins: 4 out of 10 coins are biased the rest are unbiased. Someone pulled out a coin randomly from the bag and flipped the coin 50 times. There was a 6 long (or longer) run in the sequence. Calculate the probability using the Bayes rule that the given coin was biased. Print out the evidence $\mathbb{P}(6\ long\ seq)$ (let's call it: `prob_6seq`) and the posterior: $\mathbb{P}(biased \mid 6\ long\ seq)$ (: `prob_biasG6seq`) **in percent**. UNCOMMENT the lines started with `q` and complete the unfinished lines.

The output of the new code should look like:

```
[1] p_ub:  0.xxx
[1] p_b:  0.xxx
[1] prob_6seq:  xx.xx %
[1] prob_biasG6seq:  xx.xx %
```

Exercise B: "Beta function and plotting"

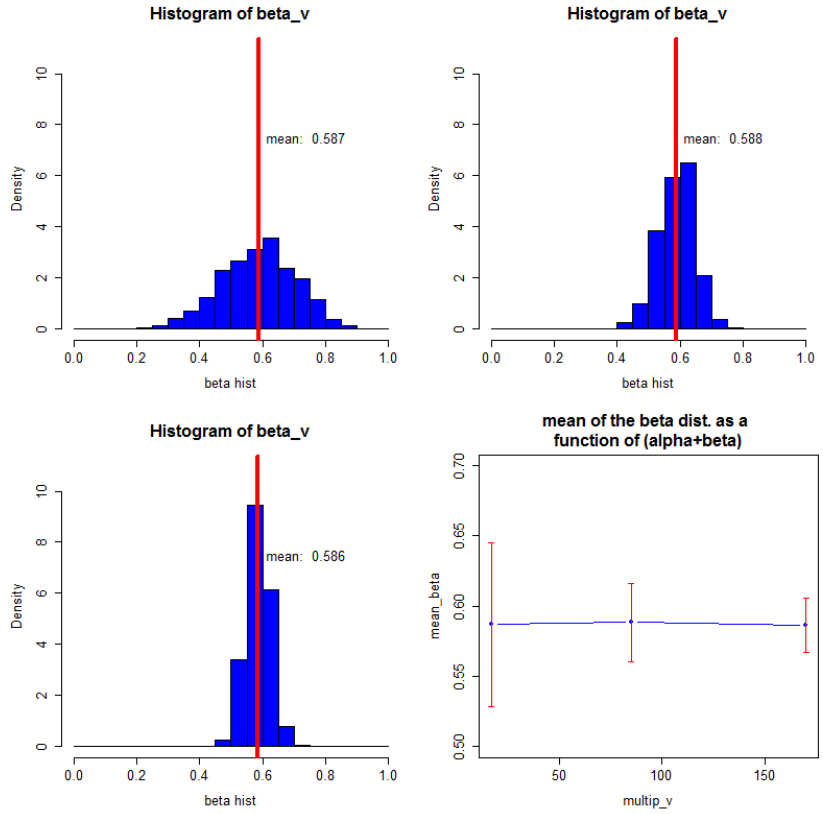
In this part you can learn something from the beta function and can practice the visualization of the results.

The beginning of the code is in the `homeworkB.R` file. You have to modify this file. The plot.R file could help you which contains the code of the last picture in the R_introduction slides.

Your task:

1. Change the distribution of the random vector from standard normal to beta with parameters $\alpha = 10,\ \beta = 7$, and set the vector length to 1000.
2. In the `hist()` function set the beginning of the first bin to 0, the end of the last bin to 1, and the bin width to 0.05. Use density instead of frequency (see freq option). Set the y axis limits from 0 to 11 (see `xlim, ylim`). Fill the bars with blue color. Name the x axis to "beta hist".
3. Calculate the mean of the random vector instead of median, and store the mean in the first element of the `mean_beta` vector. Calculate also the standard deviation of the `beta_v` vector, and store the result in the first element of the `sd_beta` vector. Draw a vertical red line to where the mean of the vector is (use the `abline()` function!). Set the red line width to 4.
4. Put a text next to the red line: "mean: xxxx" rounded to 3 digits, (instead of the text which contains the median).

5. Copy your code to the end of the script and prepare the same plot but with parameters: $\alpha = 50$, $\beta = 35$. Store the mean and the sd to the second element of the corresponding vectors.
6. Copy your code to the end of the script and prepare the same plot but with beta distribution: $\alpha = 100$, $\beta = 70$. Store the mean and the sd to the third element of the corresponding vectors.
7. Create an error bar plot with the function `errbar()`, where you plot the `mean_beta` in the function of $(\alpha + \beta)$, and add the standard deviation as error. For this, you need to install the Hmisc package with `install.packages("Hmisc")` command, (maybe you have to put a ✓ in the checkbox next to the Hmis package at the packages menu point in the multifunctional panel after the instillataion). (For more info about the `errbar()` function type `?errbar` to the command line after the installation of Hmisc.) Write your code under your already existing script.
8. Connect the markers in the plot, and set the color of the plot to blue. Set the color of the error bars to red. Set the limits of y axis to 0.5 and 0.7. Add a title above the graph (use the external `title()` command to do that, because the title option inside the errbar() function doesn't work).
9. Create a 2 by 2 picture from the graphs that looks like this (just uncomment the 4th line in the code):



10. Save your image as an 800 x 800 png picture:

```
dev.print(file="YOUR_PATH\\homework.png", png, height = 800,
width=800)
dev.off()
```

Exercise C: “Coin flipping again”

There are two bags filled with coins:

- The first bag contains two types of coins: unbiased ones (P(Head=0.5)), and biased ones (P(Head=0.7)). There are equal numbers of the two types of coins in the first bag.
- In the second bag there are nine types of coins. The following vector contains the probability of heads for the different coins: `seq(0.1,0.9,0.1)`. There are only one coin from each type in the second bag.

Someone grabbed one of the bags, pulled out a coin and tossed it 8 times. He wrote down the output of the tosses on a paper. Then he put back to coin and pulled out again a coin from the same bag. He tossed again this 8 times. Here you can see the two output sequences:

1st: 0,1,0,0,1,0,1,0

2nd: 0,1,1,1,0,1,1,1

Let's model the two bags first, then determine which model fits better the data. Write a code in R that can calculate the Bayes factor (relative evidence for two models). Please write a few sentence long explanation about the result.

Please submit the following:

- all your R codes with short comments in the codes
- the saved image in exercise B
- your explanation of the result in exercise C