Report Data Analysis with R
2nd assignment

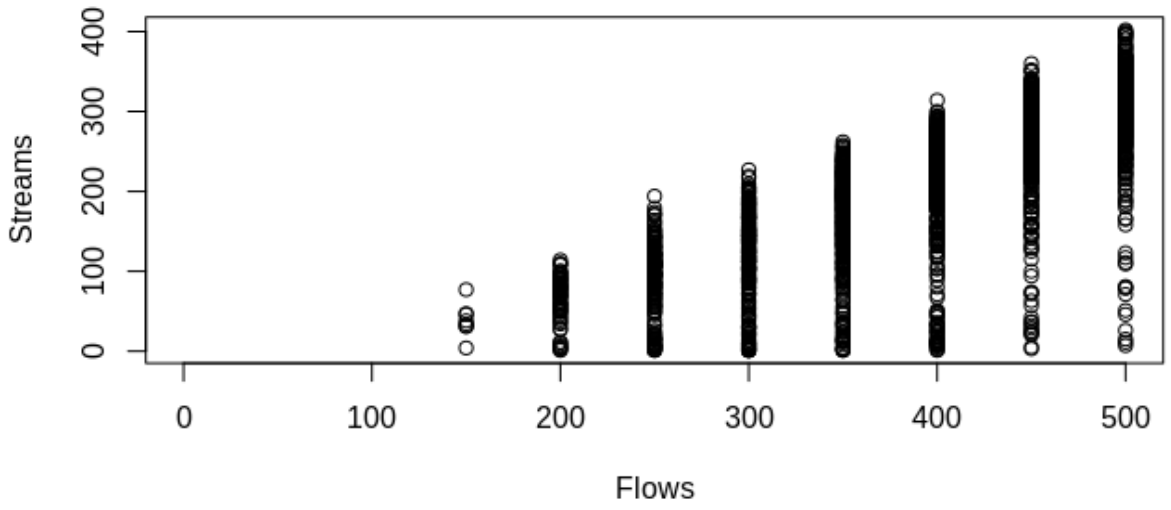
People in charge: Mr. Mai Long, Dr.Navet
Student: Pedro Gomes, 017066611B

Question 1.

See code, #Question1

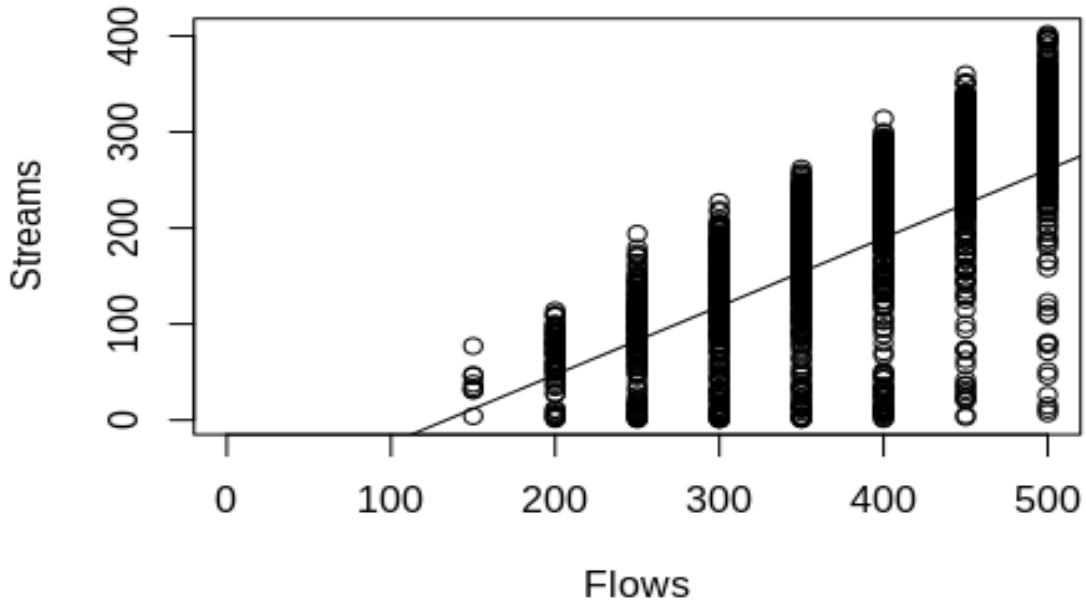
Question 2.

a) I observe that when the number of flows increases so does the number of streams. Furthermore, one is also able to observe that there are no flows with a value under 150 which do not respect their constraints.



b) *See code, #Question2#b*

c) There is a linear relationship between the two quantities. As we can see we have a linear line that increases in Y(streams), as the number of flows increases. The r-square is 70%, which means the line should pass through 70% of all points, this seems however hard to see for me:



d) Feasible: 2018 -- NOT feasible: 1982

Question 3.

a) *See code, #Question3#a*
b) See code, #Question3#b
c) See code, #Question3#c
d) See code, #Question3#d
e) The optimal K varies depending on the shuffled data I have. Every time the data is shuffled the k varies. If I do not shuffle the data AT ALL, this is the K I obtain: k = 3 with 0.998 accuracy. In order to check for false negatives and false positives, I did a cross table(again, the cross table varied depending in how the data was shuffled.). For a k = 27 with 0.964% accuracy I got the following table:

Total Observations in Table: 1000

	best_k_result		
test_labels	Feasible	NOT Feasible	Row Total
Feasible	484	14	498
	0.972	0.028	0.498
	0.953	0.028	
	0.484	0.014	
NOT Feasible	24	478	502
	0.048	0.952	0.502
	0.047	0.972	
	0.024	0.478	
Column Total	508	492	1000
	0.508	0.492	

You can find the shuffled data for this cross table attached, and you can load it with: *load("<file_name>").*

False Positive: 14 – 2%
False Negative: 24 – 4%

f) (In order to better understand my answer, please have a look at my code #Question3#f)

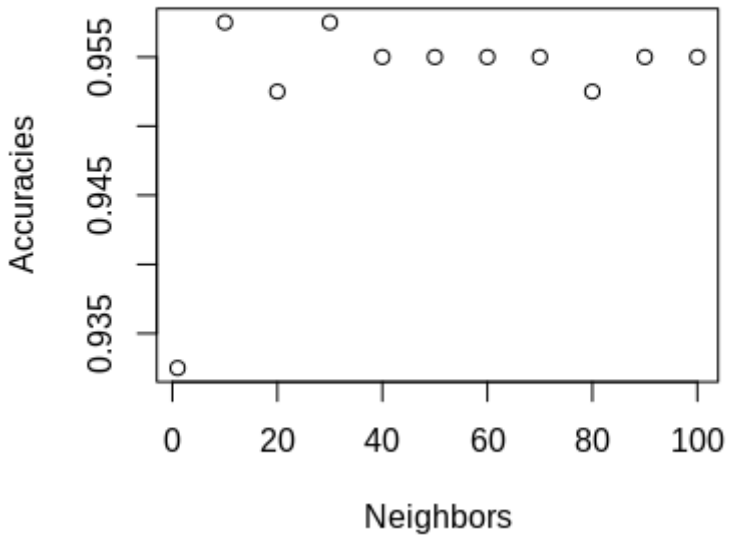
I started by making a for loop in 2:16, where 16 is the maximum number of features we can have. Within this loop, I made another loop for the range of K, 1 to 100. For the shuffled data that is attached(same from previous question: e)), I got the following results:

```
[1] 52.000  0.915  2.000
[1] 20.000  0.924  3.000
[1] 52.000  0.928  4.000
[1] 28.000  0.966  5.000
[1] 28.000  0.959  6.000
[1] 13.00  0.96  7.00
[1] 6.000 0.957 8.000
[1] 14.000  0.955  9.000
[1] 16.000  0.952 10.000
[1] 13.000  0.951 11.000
[1] 15.00  0.95 12.00
[1] 42.000  0.948 13.000
[1]  8.000  0.947 14.000
[1] 34.000  0.948 15.000
[1]  4.000  0.964 16.000
```

As you can see in my code, in the 1st column we have the optimal k(s), in the 2nd column we have the accuracy with the optimal k, and finally in the 3th column we have the number of features. This means that in this case, we have the best accuracy with 5 features, where the k = 28, and the accuracy = 96% . However it is worth noticing that with 16 features, with a k = 4, we have a very much close result(0.964 vs 0.966).

Question 4.

a) With the data shuffled like in the attached file, I got the following plot:



The numbers of neighbors that maximize the accuracy is 10 and 40. Please note that if you run the calculations of the Question4#a several times without cleaning the existing variables with the broom logo in the environment tab you will obtain different results. In order to obtain exactly what I have here, load the file attached and run the script up until the line: *training_labels <- labels[start_traning:end_training],* after that skip immediately to question 4a. Also, when you load the data file, you can uncomment the line: *#load(file="shuffled_data_k50.Rda")* start from here, and ignore the code that comes before this line.

b) It does not seem zscore normalization improves the accuracy of results, except for two cases: 40 and 50 neighbors.