**STK 353**

**Practical 2: Scripts**

Name: SJP Eloff

Student number: 10237161

Submission: 30 July 2018, 17:30

Answer Sheet

|  |  |
| --- | --- |
| 1 | 1. There are 10,200 flights that had an arrival delay of two or more hours, as seen from the following (partial) output:   # A tibble: 10,200 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 1 1 811 630 101 1047  2 2013 1 1 848 1835 853 1001  …   1. 9,313 Flights had Houston (IAH or HOU) as their destination:   # A tibble: 9,313 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 1 1 517 515 2 830  2 2013 1 1 533 529 4 850  …   1. 139,504 Flights were under control of United, American and Delta airlines:   # A tibble: 139,504 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 1 1 517 515 2 830  2 2013 1 1 533 529 4 850  …   1. 86,326 Flights departed during summer:   # A tibble: 86,326 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 7 1 1 2029 212 236  2 2013 7 1 2 2359 3 344  …   1. 34,583 Flights departed on time (or early) and still arrived more than 2 hours late:   # A tibble: 34,583 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 1 1 554 558 -4 740  2 2013 1 1 555 600 -5 913  …   1. 83,728 Flights departed at least 1 hour late, but made up for at least 30 minutes lost time in flight:   # A tibble: 83,728 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 1 1 601 600 1 844  2 2013 1 1 623 610 13 920  …   1. 9,344 Flights departed between 00:00 and 06:00 (inclusively):   # A tibble: 9,344 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 1 1 517 515 2 830  2 2013 1 1 533 529 4 850  …  NOTE to g): It seems that flights that departed exactly at 24:00 should be included by nature of the question, however, this does not correspond to question 2’s answers so I assume the lecturer does not want me to include those flights – I therefore only coded for the filter filter(dep\_time <= 600) in stead of filter(dep\_time <= 600|dep\_time == 2400) which delivers the result of 9,373 flights. |
| 2 | 1. The between() function analyses whether the values of a numerical input vector falls in a specific range. 2. Using between() in question 1g) delivers the same result:   # A tibble: 9,344 x 19  year month day dep\_time sched\_dep\_time dep\_delay arr\_time  *<int>* *<int>* *<int>* *<int>* *<int>* *<dbl>* *<int>*  1 2013 1 1 517 515 2 830  2 2013 1 1 533 529 4 850  … |
| 3 | 1. 8255 Flights’ departure times are missing:   [1] 8255   1. The output below shows the columns that contain (at least one) missing value(s):   [1] "dep\_time" "dep\_delay" "arr\_time" "arr\_delay" "tailnum"  [6] "air\_time"   1. There are many possible causes to consider, e.g. (1) flights that might have been cancelled (for whatever reason, be it not having enough passengers, systems failure of the aircraft, pilot not showing up, etc.), (2) failure to capture the data (be it faulty technology or human error), or (3) flights might have been diverted and never reached their final destination. Note these are only some of the possible causes of the missing data. |
| 4 | 1. NA ^ 0 is equal to 1 and not to NA as I expected, since NA is treated as a placeholder for *some possible number*, and from the laws of mathematics, (any possible number)^0 = 1. Thus NA ^ 0 is not missing. 2. NA | TRUE is not missing since, again, NA is simply a placeholder. In this case, as a logical boolean object, i.e. it can only be either TRUE or FALSE. Therefore, NA | TRUE will always produce the output TRUE. 3. FALSE & NA Is not missing for the same reason as in b) and will always produce the output FALSE. |
| 5 | The following code produces the desired result:  flights %>% arrange(desc(is.na(dep\_time)), desc(is.na(dep\_delay)),  desc(is.na(arr\_time)), desc(is.na(arr\_delay)),  desc(is.na(tailnum)),desc(is.na(air\_time))) |
| 6 | The code below first sorts flights according to their real departure time in descending order from the most delayed flight to the flight that left earliest, and then in ascending order from the flight that left earliest to the flight that was most delayed:  arrange(flights, desc(dep\_delay))  arrange(flights, dep\_delay) |
| 7 | The code below sorts flights in ascending order according to their flight duration:  arrange(flights, air\_time) |
| 8 | The code below first sorts flights according to their distance travelled in descending order and thereafter in ascending order:  arrange(flights, desc(distance))  arrange(flights, distance) |
| 9 | The code below converts dep\_time and sched\_dep\_time to number of minutes since midnight:  flights %>% mutate(dep\_time = (dep\_time %/% 100) \* 60 +  (dep\_time %% 100),  sched\_dep\_time = (sched\_dep\_time %/% 100) \* 60 +  (sched\_dep\_time %% 100)) |
| 10 | I would expect to find that the average value of (arr\_time – dep\_time) - air\_time is close to zero.  Firstly, arr\_time needs to be mutated to the minutes since midnight format as done with dep\_time and sched\_dep\_time in question 9.  Furthermore, notice that arr\_time – dep\_time sometimes result in large negative numbers; this occurs when a flight departs before midnight and arrives after it. We can, however, just manipulate the variables algebraically once again to solve this problem. |
| 11 | I would expect that (dep\_time – sched\_dep\_time)- dep\_delay averages close to zero. |
| 12 | If there were ties, I might consider (depending on domain knowledge) to use row\_number() to deal with them. |
| 13 | 1:3 + 1:10 outputs the following:  [1] 2 4 6 5 7 9 8 10 12 11  Warning message:  In 1:3 + 1:10 :  longer object length is not a multiple of shorter object length  i.e. the problem is that we cannot repeat a vector of size 3 to size 10 since 10 is not a multiple of 3. |
| 14 | From the ?Trig function, R can compute the cosine, sine, tangent, arc-cosine, arc-sine, arc-tangent, and the two-argument arc-tangent. |
| 15 | Below is a graph depicting the relationship between distance and average arrival delay for each of the locations: |
| 16 | Below follows the depicted relationships of carrier vs. mean delay, month vs. mean delay, arr\_time vs. mean delay and dep\_time vs. mean delay respectively:  From the four graphs above, it looks to be the case that dep\_time might be the best predictor for delay time. Furthermore, comparing (firstly) the total number of delayed flights on different days of the month also does not seem very useful and (secondly) neither does the comparison between average delay time and day of the month: |
| 17 | The average delay indeed seems highly related to the number of cancelled flights per day (cncld/k), as depicted below: |

**Total Marks (out of 10):**