SmartFly: Exploratory Analysis For Scheduled Flight Data

Cindy Lamm

11:33, Thursday 15th January, 2015

Assuming that scheduled flight data and historic flight data have the same variables, I load these variable names and types of historic data (prepared in an additional csv file):

```
nameTypeDataFile  <- "resources/raw_variables.csv"
variableNames <- read.csv(nameTypeDataFile, header=TRUE, stringsAsFactors=FALSE)
variableNames
```

```
##                        name      type
## 1                        id character
## 2                      year    factor
## 3                     month    factor
## 4               day_of_month    factor
## 5                day_of_week    factor
## 6  scheduled_departure_time    factor
## 7    scheduled_arrival_time    factor
## 8                   airline    factor
## 9             flight_number    factor
## 10              tail_number    factor
## 11              plane_model    factor
## 12       seat_configuration    factor
## 13          departure_delay   numeric
## 14           origin_airport    factor
## 15      destination_airport    factor
## 16       distance_travelled   numeric
## 17             taxi_time_in   numeric
## 18            taxi_time_out   numeric
## 19                cancelled   integer
## 20        cancellation_code    factor
```

```
factor_idx <- which(variableNames$type=="factor")
```

Then load scheduled data into R. As I did for the historic data I set empty strings to `NA` (here because of variable `tail_number`).

```
scheduledDataFile <- "../../data/smartfly_scheduled.csv"
predictDataTyped <- read.csv(scheduledDataFile, header=FALSE, stringsAsFactors=FALSE,
                          col.names=variableNames$name, colClasses=variableNames$type,
                          na.strings=c("NA",""))
```

Checkout data content:

```
str(predictDataTyped)
```

```
## 'data.frame': 566376 obs. of  20 variables:
##  $ id                     : chr  "4972683369271453960" "4755622236989466036" "1092083446069765248" "
##  $ year                   : Factor w/ 1 level "2015": 1 1 1 1 1 1 1 1 1 1 ...
##  $ month                  : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ day_of_month           : Factor w/ 31 levels "1","10","11",..: 4 5 6 7 8 9 11 13 14 15 ...
##  $ day_of_week            : Factor w/ 7 levels "1","2","3","4",..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ scheduled_departure_time: Factor w/ 1086 levels "0","10","100",..: 877 877 877 877 877 877 877 87
##  $ scheduled_arrival_time  : Factor w/ 1250 levels "1","10","100",..: 1206 1206 1206 1206 1206 1206 1
##  $ airline                : Factor w/ 19 levels "AA","AS","B6",..: 16 16 16 16 16 16 16 16 16 16 ...
##  $ flight_number          : Factor w/ 7321 levels "1","10","100",..: 3913 3913 3913 3913 3913 3913 3
##  $ tail_number            : Factor w/ 4687 levels "0","N050AA","N051AA",..: 3904 4092 1887 3998 4013
##  $ plane_model            : Factor w/ 6 levels "737","747","757",..: 2 2 1 3 5 6 2 3 3 2 ...
##  $ seat_configuration     : Factor w/ 6 levels "Standard","Three Class",..: 6 2 4 4 2 4 4 4 6 4 ...
##  $ departure_delay        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ origin_airport         : Factor w/ 274 levels "ABE","ABI","ABQ",..: 196 196 196 196 196 196 196 1
##  $ destination_airport    : Factor w/ 274 levels "ABE","ABI","ABQ",..: 60 60 60 60 60 60 60 60 60 60
##  $ distance_travelled     : num  599 599 599 599 599 599 599 599 599 599 ...
##  $ taxi_time_in           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ taxi_time_out          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ cancelled              : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ cancellation_code      : Factor w/ 0 levels: NA NA NA NA NA NA NA NA NA NA ...
```

Checkout factor levels for the variables.

Note: If I train a model on levels that don't exist in the prediction data the prediction phase might fail.

As I did for the historic data the variables `scheduled_departure_time` and `scheduled_arrival_time` are first reformatted and then truncated to the hour.

```
predictDataTyped$scheduled_departure_time <- as.factor(
  sprintf("%04s", as.character(predictDataTyped$scheduled_departure_time)))
predictDataTyped$scheduled_arrival_time <- as.factor(
  sprintf("%04s", as.character(predictDataTyped$scheduled_arrival_time)))
```

```
predictDataTyped$scheduled_departure_time <- as.factor(
  substr(as.character(predictDataTyped$scheduled_departure_time),1,2))
predictDataTyped$scheduled_arrival_time <- as.factor(
  substr(as.character(predictDataTyped$scheduled_arrival_time),1,2))
```

I also again reformat the variables `day_of_month` and `month`:

```
predictDataTyped$month <- as.factor(
  sprintf("%02s", as.character(predictDataTyped$month)))
predictDataTyped$day_of_month <- as.factor(
  sprintf("%02s", as.character(predictDataTyped$day_of_month)))
```

See summary of descriptive statistics of the scheduled data:

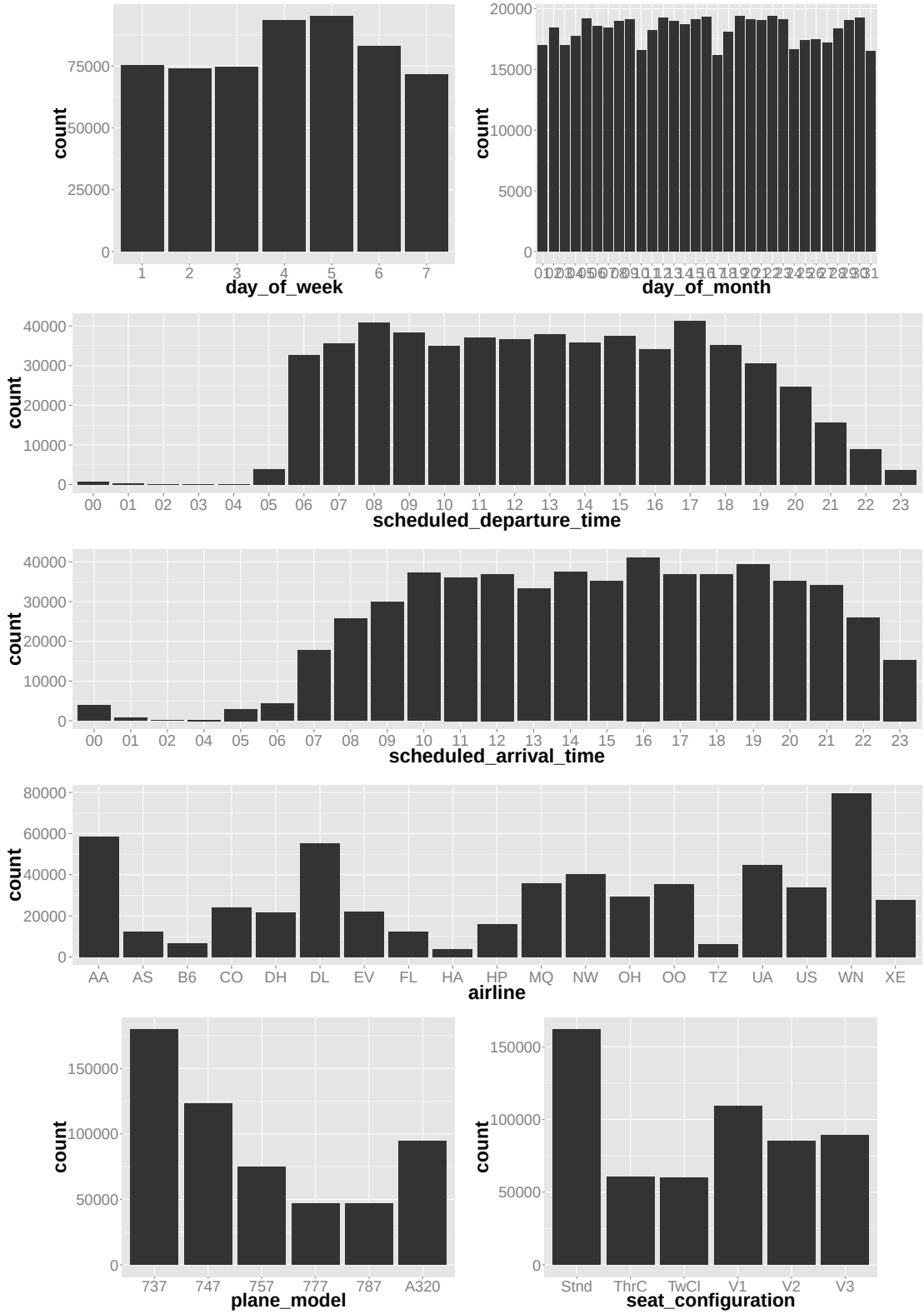
```
summary(predictDataTyped)
```

```
##       id              year        month         day_of_month    day_of_week
##  Length:566376      2015:566376   01:566376   22     : 19395   1:75237
##  Class :character                             19     : 19347   2:73819
##  Mode  :character                             16     : 19286   3:74482
##                                               30     : 19268   4:93432
##                                               12     : 19210   5:95177
##                                               05     : 19206   6:82832
##                                               (Other):450664   7:71397
##  scheduled_departure_time scheduled_arrival_time    airline       flight_number
##  17     : 41179           16     : 41124          WN     : 79417   524    :   440
##  08     : 40947           19     : 39394          AA     : 58593   186    :   439
##  09     : 38285           14     : 37482          DL     : 55480   238    :   439
##  13     : 37904           10     : 37170          UA     : 44792   273    :   437
##  15     : 37474           12     : 36938          NW     : 40149   417    :   428
##  11     : 37030           18     : 36871          MQ     : 35795   217    :   416
##  (Other):333557           (Other):337397          (Other):252150   (Other):563777
##   tail_number     plane_model     seat_configuration departure_delay  origin_airport
##  N478HA :   339   737 :179931   Standard   :162109   Min.   : NA      ATL    : 33615
##  N481HA :   339   747 :123049   Three Class: 60695   1st Qu.: NA      ORD    : 30168
##  N484HA :   334   757 : 75092   Two Class  : 60174   Median : NA      DFW    : 28801
##  N183UW :   314   777 : 46719   V1         :109484   Mean   :NaN      LAX    : 18899
##  N487HA :   310   787 : 46837   V2         : 84879   3rd Qu.: NA      CVG    : 16747
##  N95    :   309   A320: 94748   V3         : 89035   Max.   : NA      IAH    : 16169
##  (Other):564431                                      NA's   :566376   (Other):421977
##  destination_airport distance_travelled  taxi_time_in    taxi_time_out
##  ATL    : 33533      Min.   :  11.0     Min.   : NA      Min.   : NA
##  ORD    : 30063      1st Qu.: 305.0     1st Qu.: NA      1st Qu.: NA
##  DFW    : 28743      Median : 547.0     Median : NA      Median : NA
##  LAX    : 18889      Mean   : 712.9     Mean   :NaN      Mean   :NaN
##  CVG    : 16583      3rd Qu.: 944.0     3rd Qu.: NA      3rd Qu.: NA
##  IAH    : 16148      Max.   :4962.0     Max.   : NA      Max.   : NA
##  (Other):422417                         NA's   :566376   NA's   :566376
##    cancelled      cancellation_code
##  Min.   : NA      NA's:566376
##  1st Qu.: NA
##  Median : NA
##  Mean   :NaN
##  3rd Qu.: NA
##  Max.   : NA
##  NA's   :566376
```

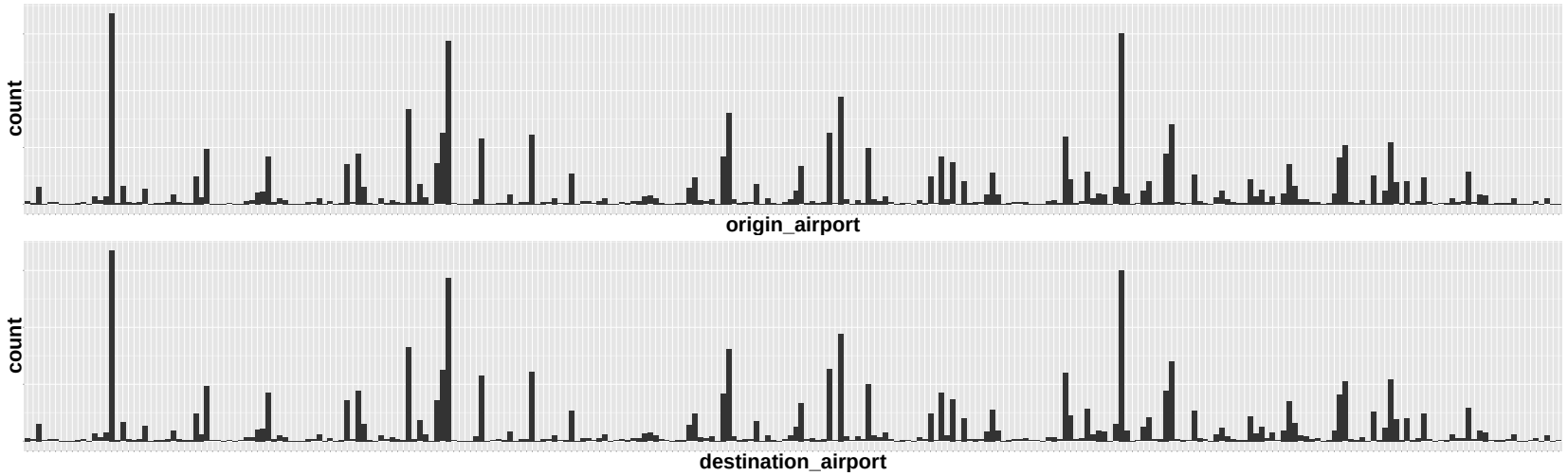
Save data frame for next step:

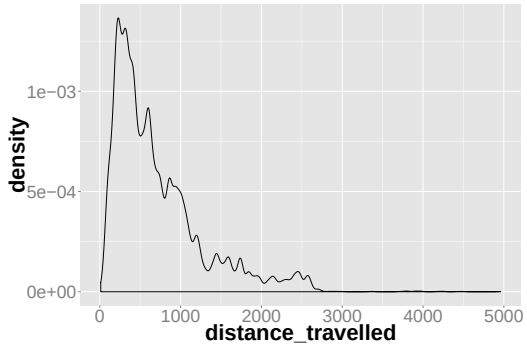
```
save(predictDataTyped, file="predictDataTyped.Rdata")
```

Plot the data:
All scheduled flight data is in January 2015 - no need for a plot there.



The variables `flight_number` and `tail_number` don't produce any valuable plots due to their large number in levels.


count
origin_airport
count
destination_airport


density
1e−03
5e−04
0e+00
0
1000
2000
3000
4000
5000
distance_travelled