

CEU-R-Tools-Project

This is my new project

```
install.packages('data.table') install.packages('nycflights13') install.packages('dplyr') install.packages("lubridate")
install.packages("tidyr") install.packages("pander")
```

CLEAR MEMORY

```
rm(list = ls())
```

```
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 3.3.2
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.3.2
```

```
library(dplyr)
```

```
## -----
```

```
## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!
```

```
## -----
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##   between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library(scales)
```

```
## Warning: package 'scales' was built under R version 3.3.2
```

```
library(plyr)
```

```
## -----  
  
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)  
  
## -----  
  
##  
## Attaching package: 'plyr'  
  
## The following objects are masked from 'package:dplyr':  
##  
##   arrange, count, desc, failwith, id, mutate, rename, summarise,  
##   summarize
```

```
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
  
## The following object is masked from 'package:plyr':  
##  
##   here  
  
## The following objects are masked from 'package:data.table':  
##  
##   hour, isoweek, mday, minute, month, quarter, second, wday,  
##   week, yday, year  
  
## The following object is masked from 'package:base':  
##  
##   date
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.3.2
```

```
library(class)
library(pander)
```

Read the data table.

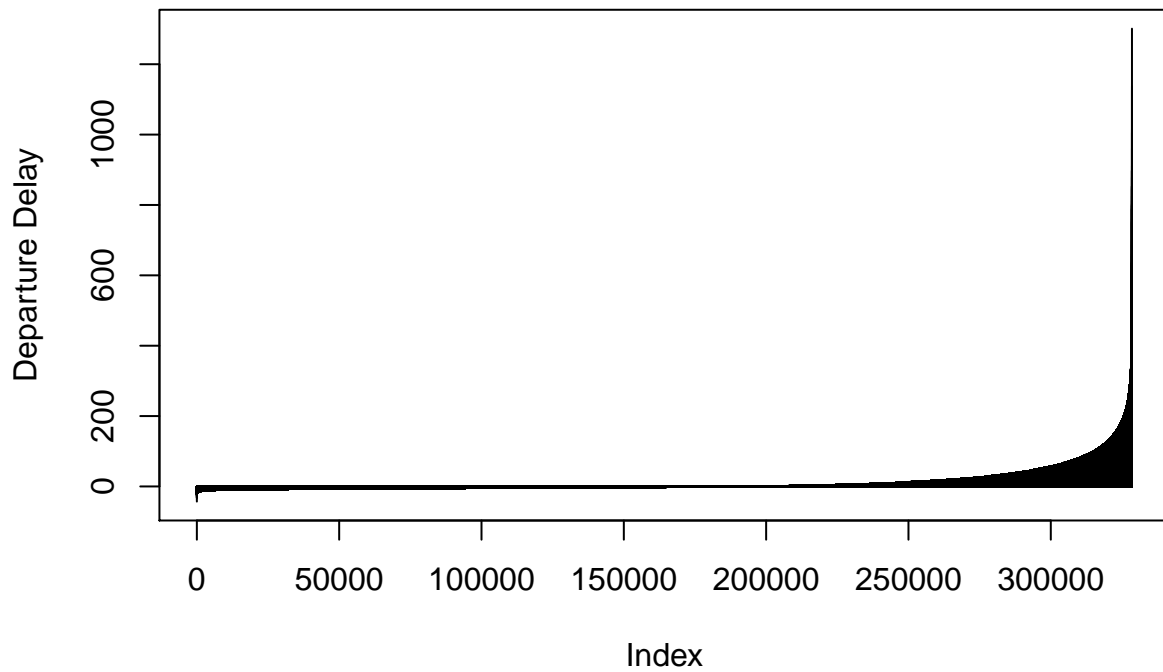
```
dtflights <- data.table(flights)
dtairports <- data.table(airports)
dtairlines <- data.table(airlines)
dtplanes <- data.table(planes)
dtweather <- data.table(weather)
str(flights)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':  336776 obs. of  19 variables:
## $ year      : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
## $ month     : int   1  1  1  1  1  1  1  1  1  1 ...
## $ day       : int   1  1  1  1  1  1  1  1  1  1 ...
## $ dep_time  : int  517 533 542 544 554 554 555 557 557 558 ...
## $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
## $ dep_delay : num   2  4  2 -1 -6 -4 -5 -3 -3 -2 ...
## $ arr_time  : int  830 850 923 1004 812 740 913 709 838 753 ...
## $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
## $ arr_delay : num   11 20 33 -18 -25 12 19 -14 -8 8 ...
## $ carrier   : chr  "UA" "UA" "AA" "B6" ...
## $ flight    : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
## $ tailnum   : chr  "N14228" "N24211" "N619AA" "N804JB" ...
## $ origin    : chr  "EWR" "LGA" "JFK" "JFK" ...
## $ dest      : chr  "IAH" "IAH" "MIA" "BQN" ...
## $ air_time  : num  227 227 160 183 116 150 158 53 140 138 ...
## $ distance  : num  1400 1416 1089 1576 762 ...
## $ hour      : num   5  5  5  5  6  5  6  6  6  6 ...
## $ minute    : num   15 29 40 45  0 58  0  0  0  0 ...
## $ time_hour : POSIXct, format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

```
?flights
```

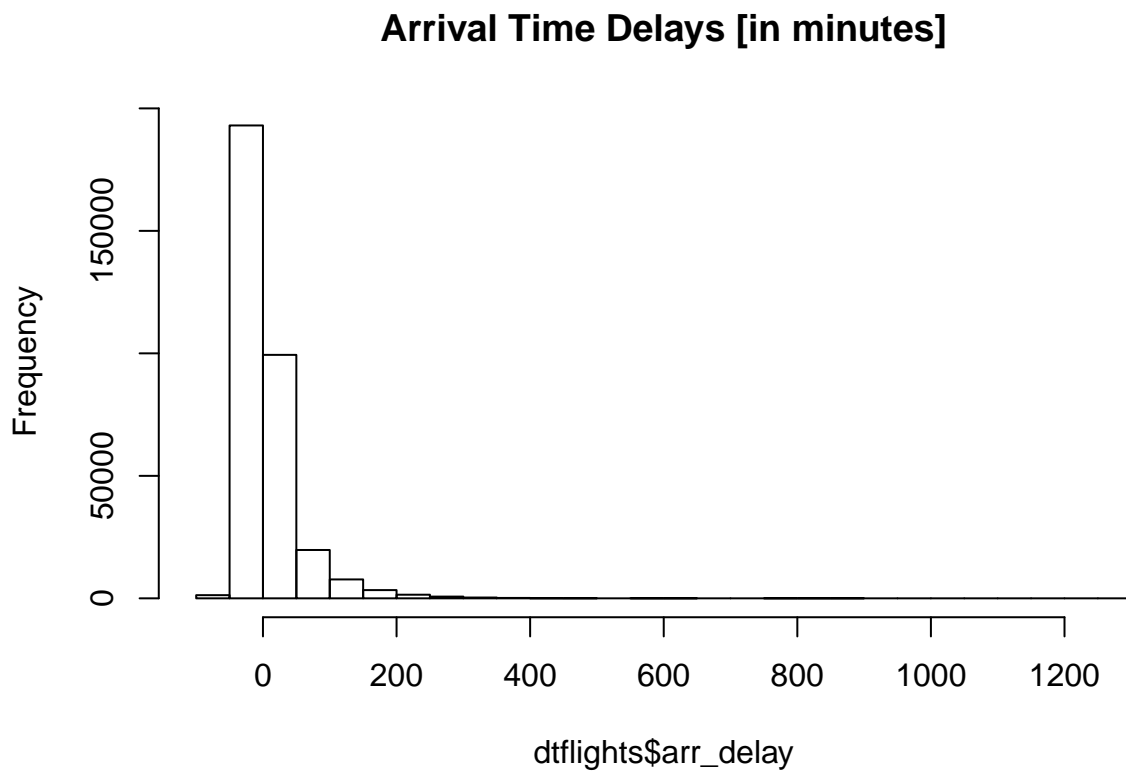
Flights departure delay

```
plot(sort(flights$dep_delay), type="h", ylab="Departure Delay")
```



Arrival Time delays in minute

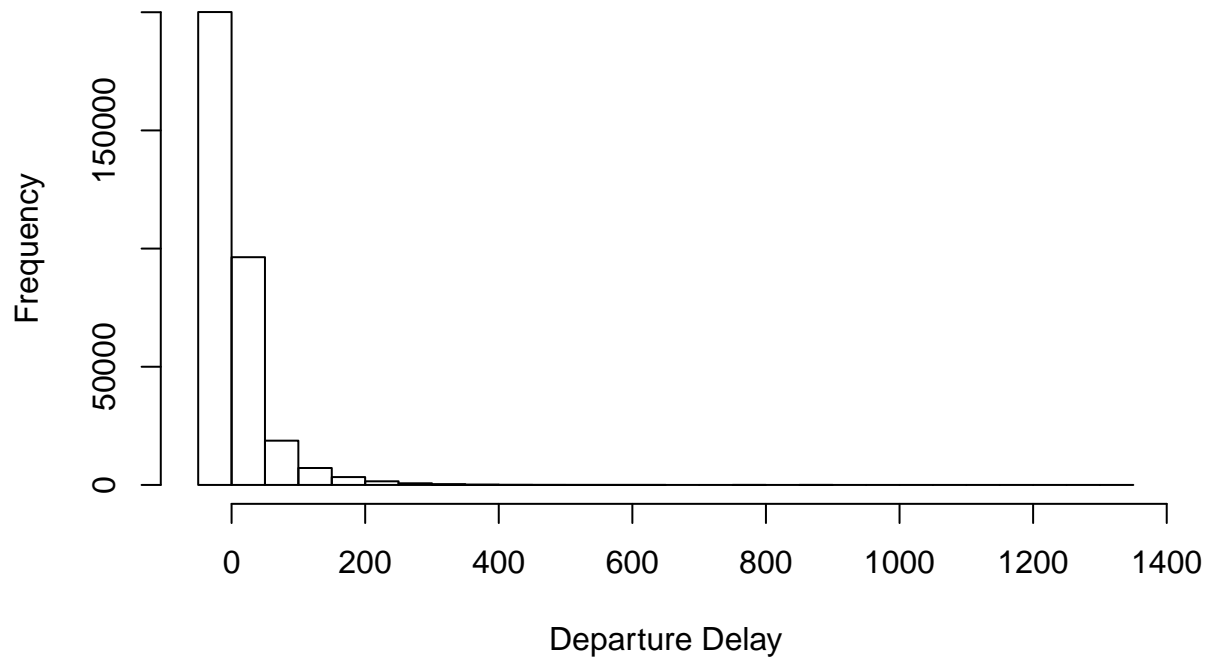
```
hist(dtflights$arr_delay, main = "Arrival Time Delays [in minutes]")
```



Flights departure delay with histogram and density plot

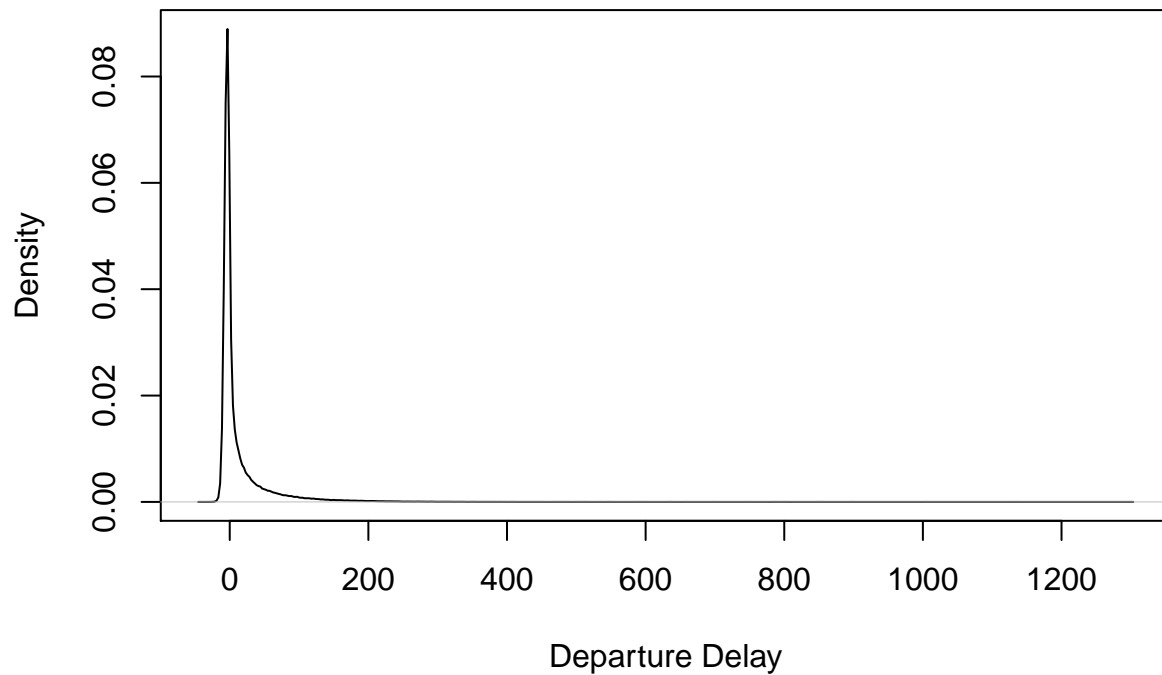
```
hist(flights$dep_delay, xlab="Departure Delay")
```

Histogram of flights\$dep_delay



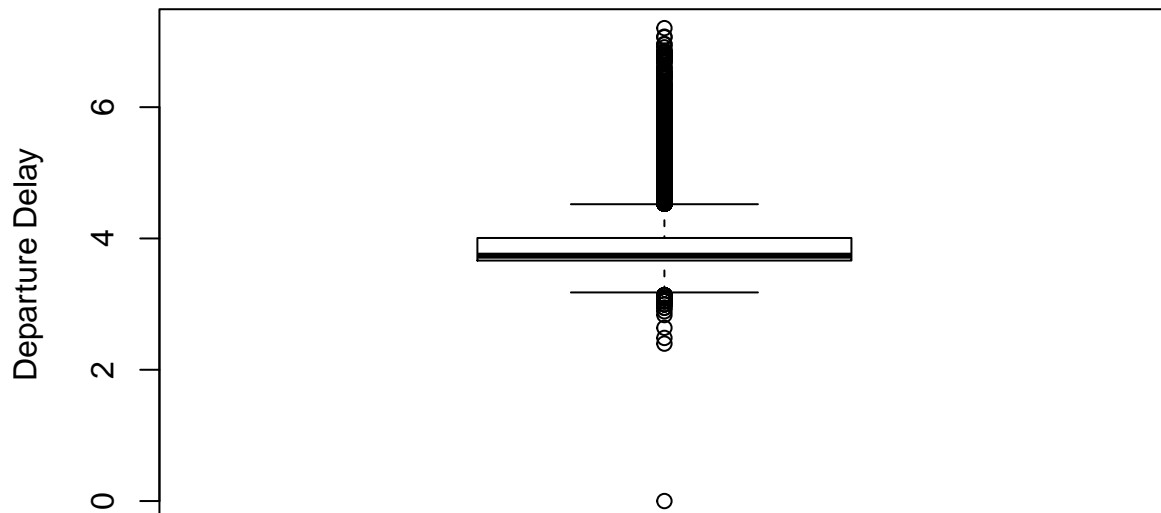
```
plot(density(flights$dep_delay, na.rm=TRUE), xlab="Departure Delay")
```

```
density.default(x = flights$dep_delay, na.rm = TRUE)
```



Flights departure delay with boxplot

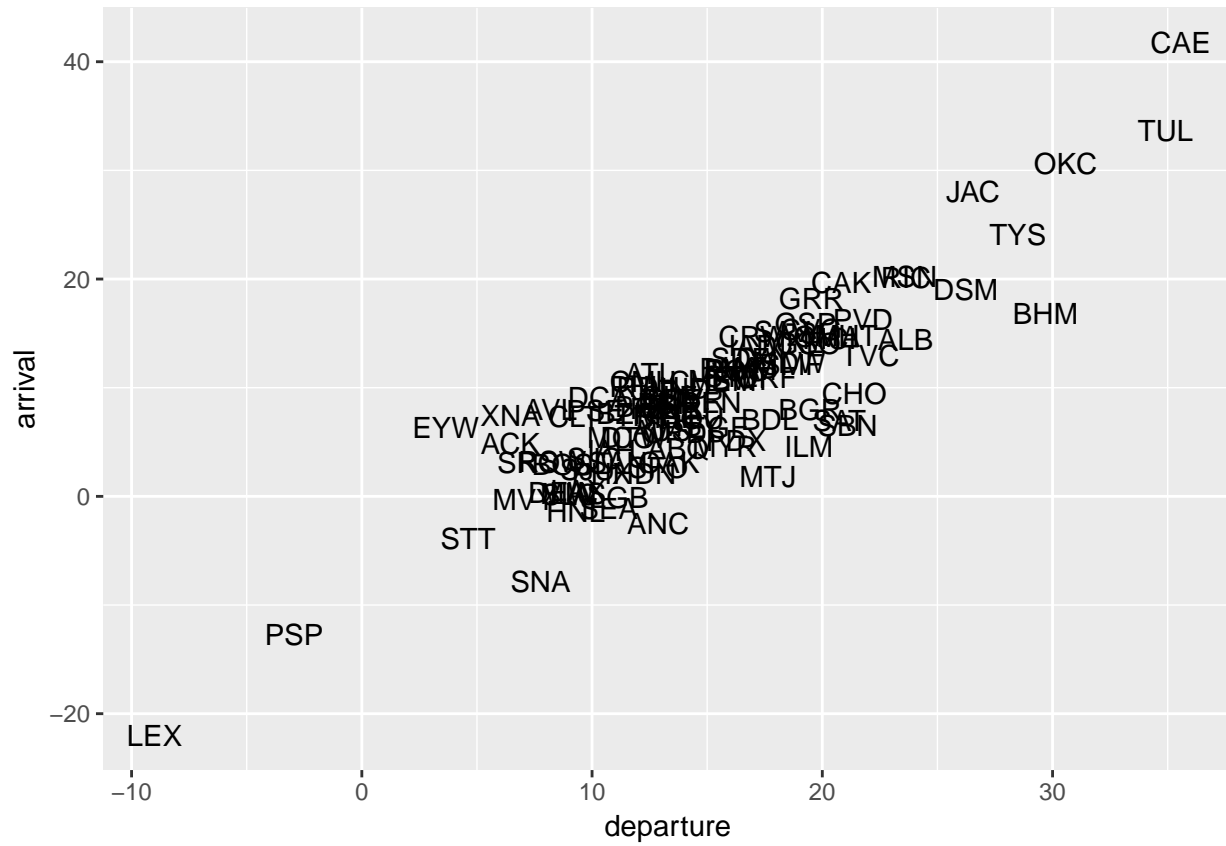
```
boxplot(log(flights$dep_delay -  
  min(flights$dep_delay, na.rm=TRUE)  
  +1), ylab="Departure Delay")
```



The average departure and arrival delays per destination

```
dta <- dtflights[, .(departure = mean(dep_delay, na.rm = TRUE),  
  arrival = mean(arr_delay, na.rm = TRUE)), by = dest]  
ggplot(dta, aes(departure, arrival, label = dest)) + geom_text()
```

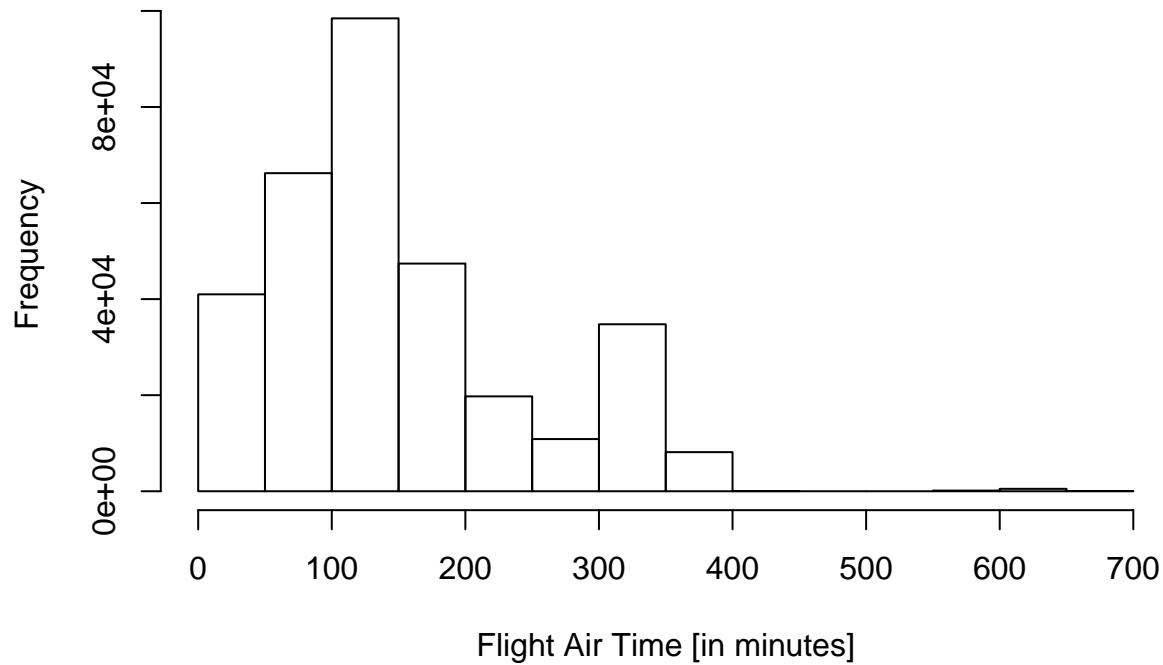
```
## Warning: Removed 1 rows containing missing values (geom_text).
```



Histogram of Flight Air time

```
hist(dtflights$air_time, xlab = "Flight Air Time [in minutes]", main = "Histogram of Flight Air Time")
```

Histogram of Flight Air Time



Which destination has the lowest average delay from 'EWR'?

```
dta <- dtflights[origin == 'EWR', .(delay = mean(arr_delay, na.rm = TRUE)), by = dest]
setorder(dta, delay)
head(dta)
```

```
##   dest    delay
## 1:  LGA      NaN
## 2:  SNA -7.868227
## 3:  SBN -5.500000
## 4:  EGE -5.349057
## 5:  ANC -2.500000
## 6:  RSW -2.259129
```

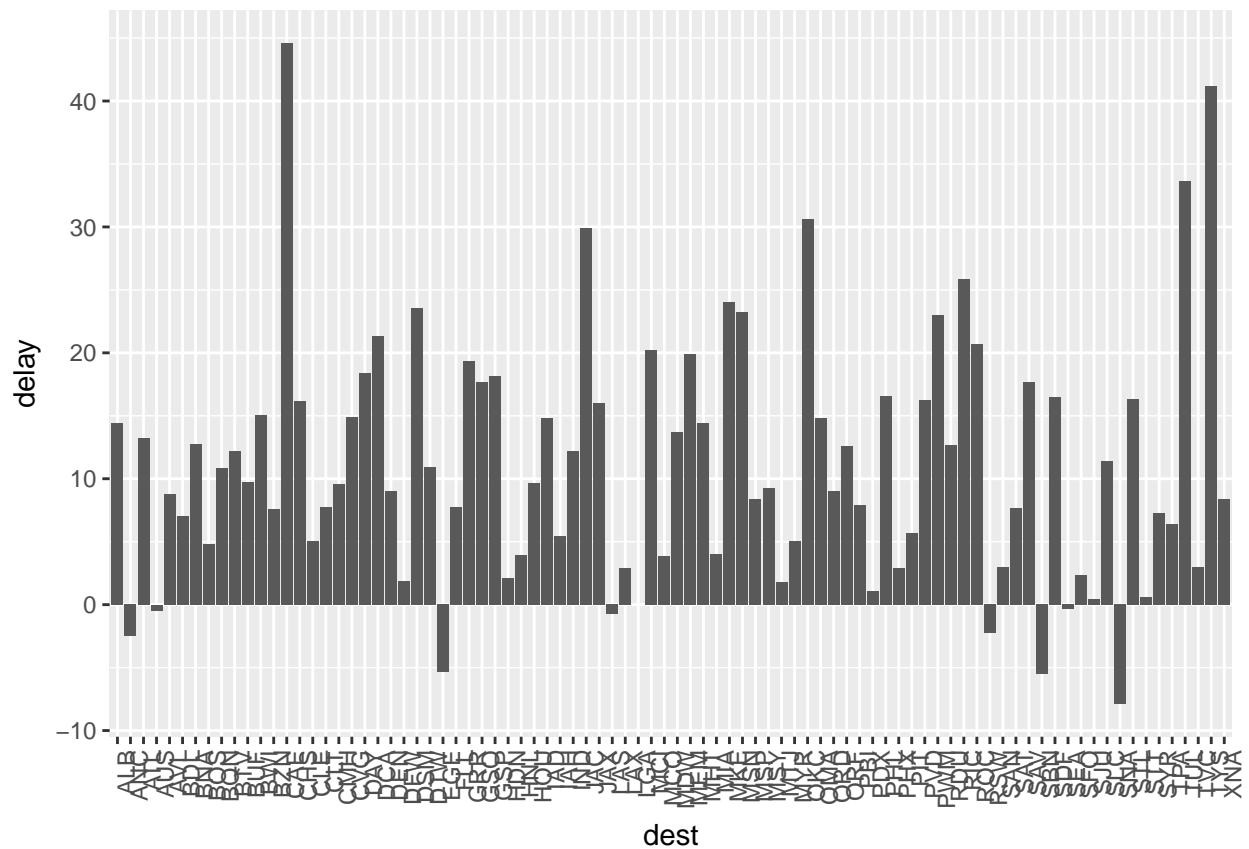
```
dta[1]
```

```
##   dest delay
## 1:  LGA   NaN
```

The average delay to all destinations from 'EWR'

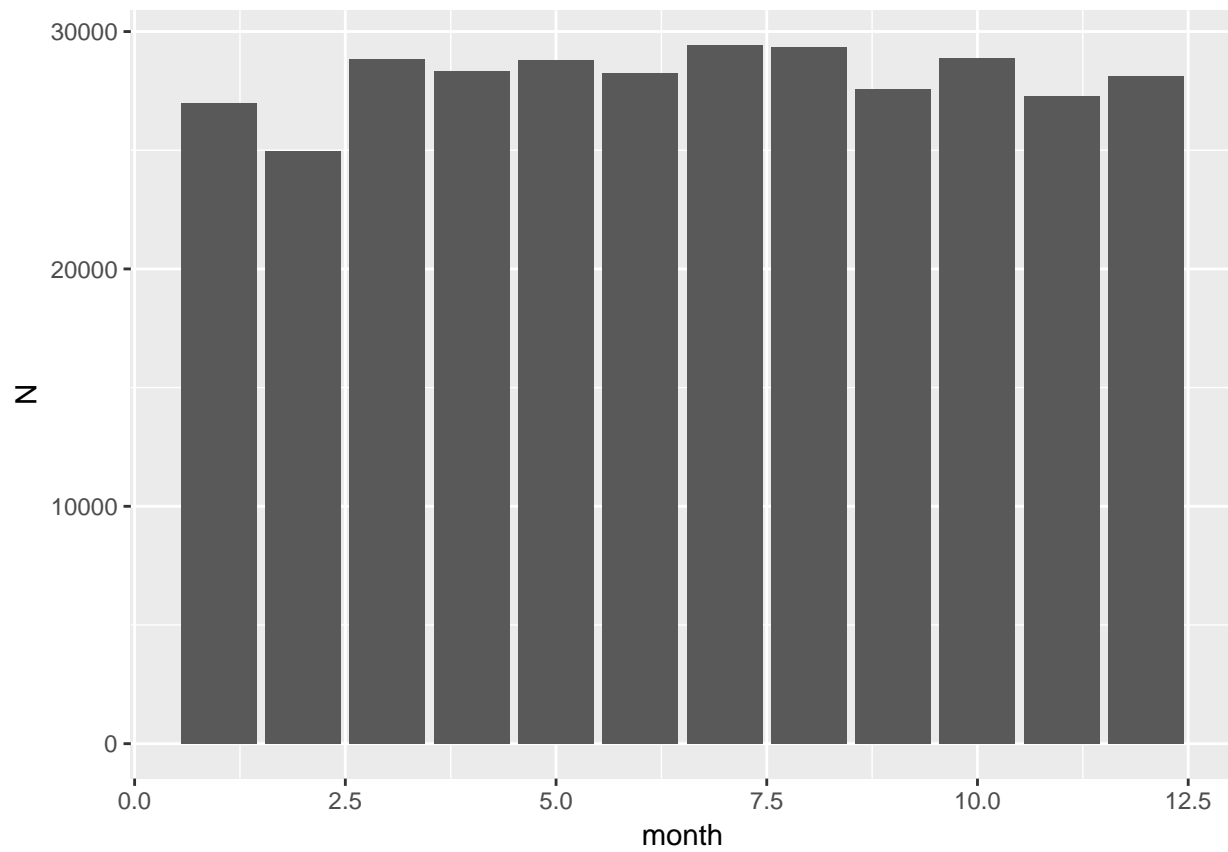
```
ggplot(dta, aes(dest, delay)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(angle = 90))
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```

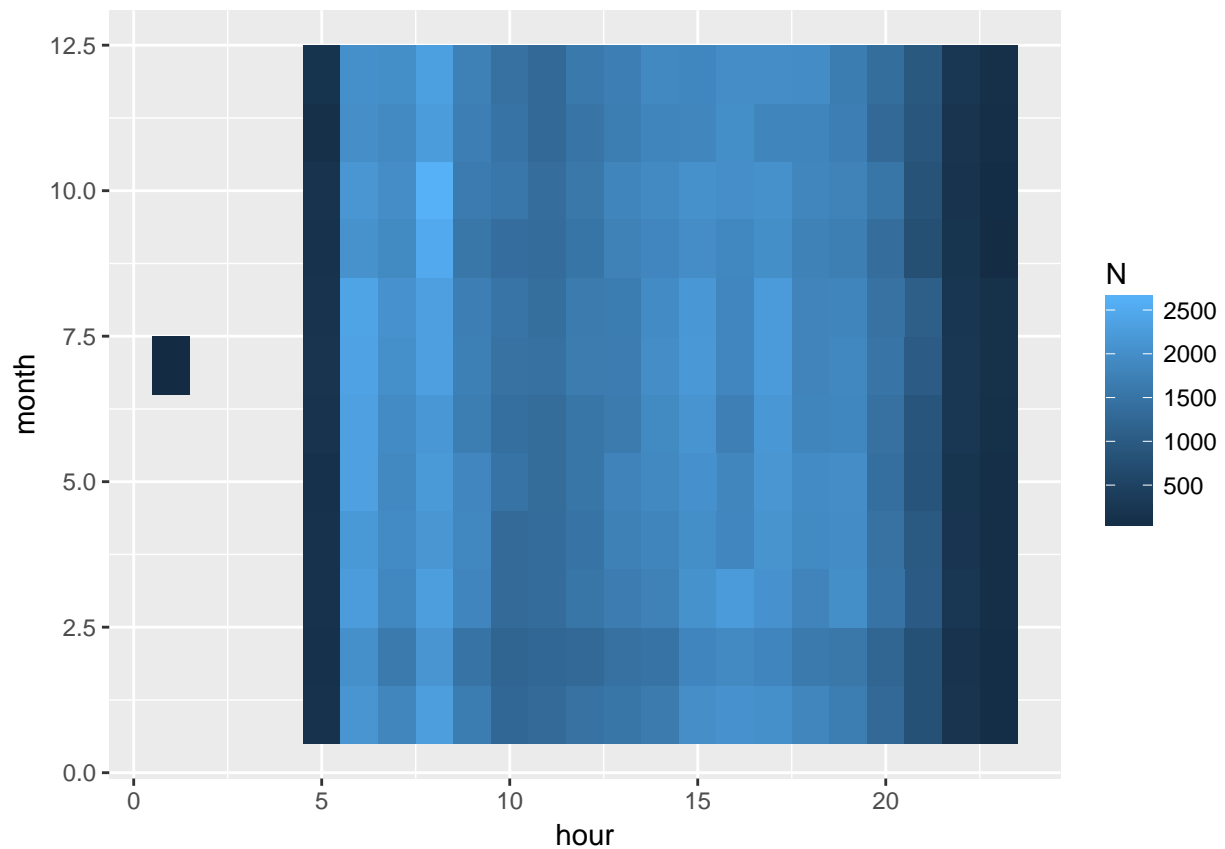
The number of flights per month

```
ggplot(dtflights[, .N, by = month], aes(month, N)) + geom_bar(stat = 'identity')
```



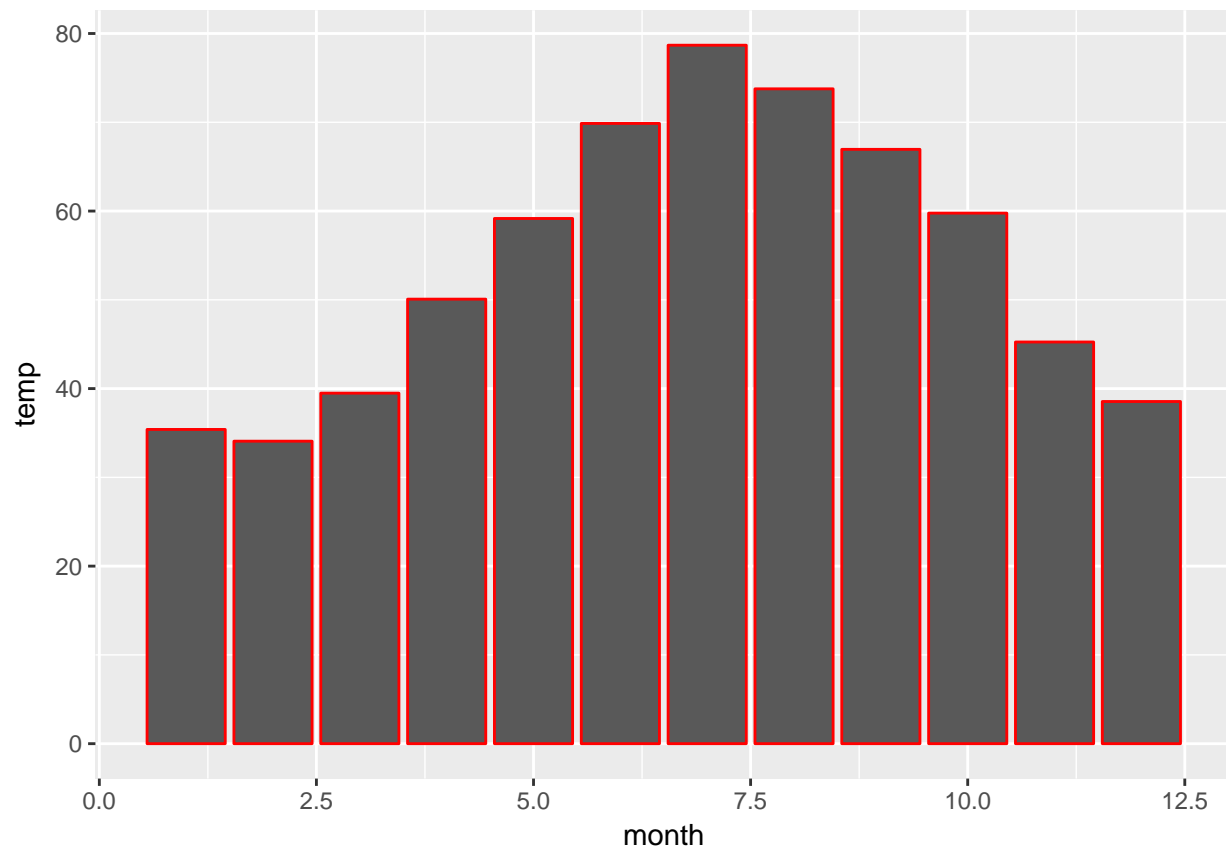
Heatmap on the number of flights per month and hour of the day

```
ggplot(dtflights[, .N, by = .(month, hour)], aes(hour, month, fill = N)) + geom_tile()
```



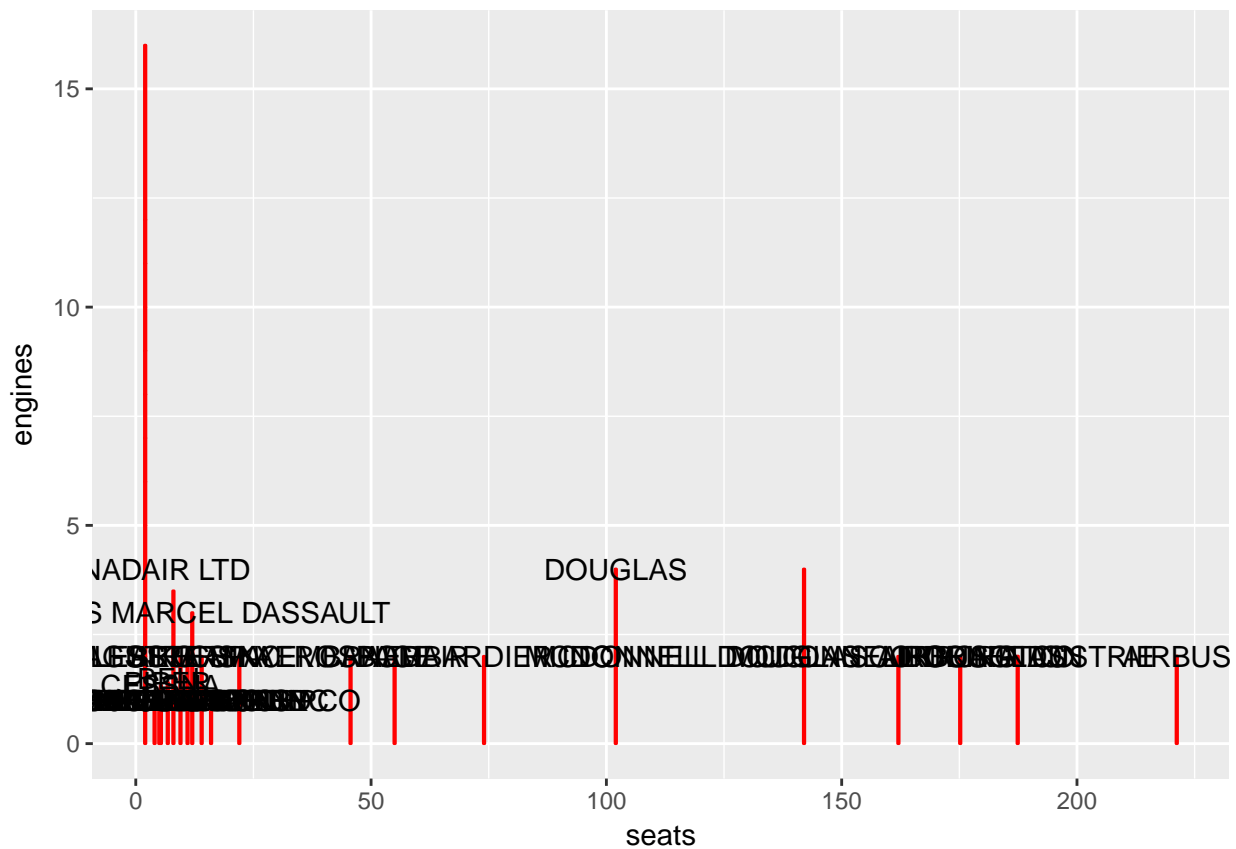
#The average temperature at noon in JFK for each month based on the `weather` dataset

```
dt <- data.table(weather)
ggplot(dt[origin == 'JFK', .(temp = mean(temp, na.rm = TRUE)), by = month], aes(month, temp)) +
  geom_bar(stat = 'identity', color = 'red')
```



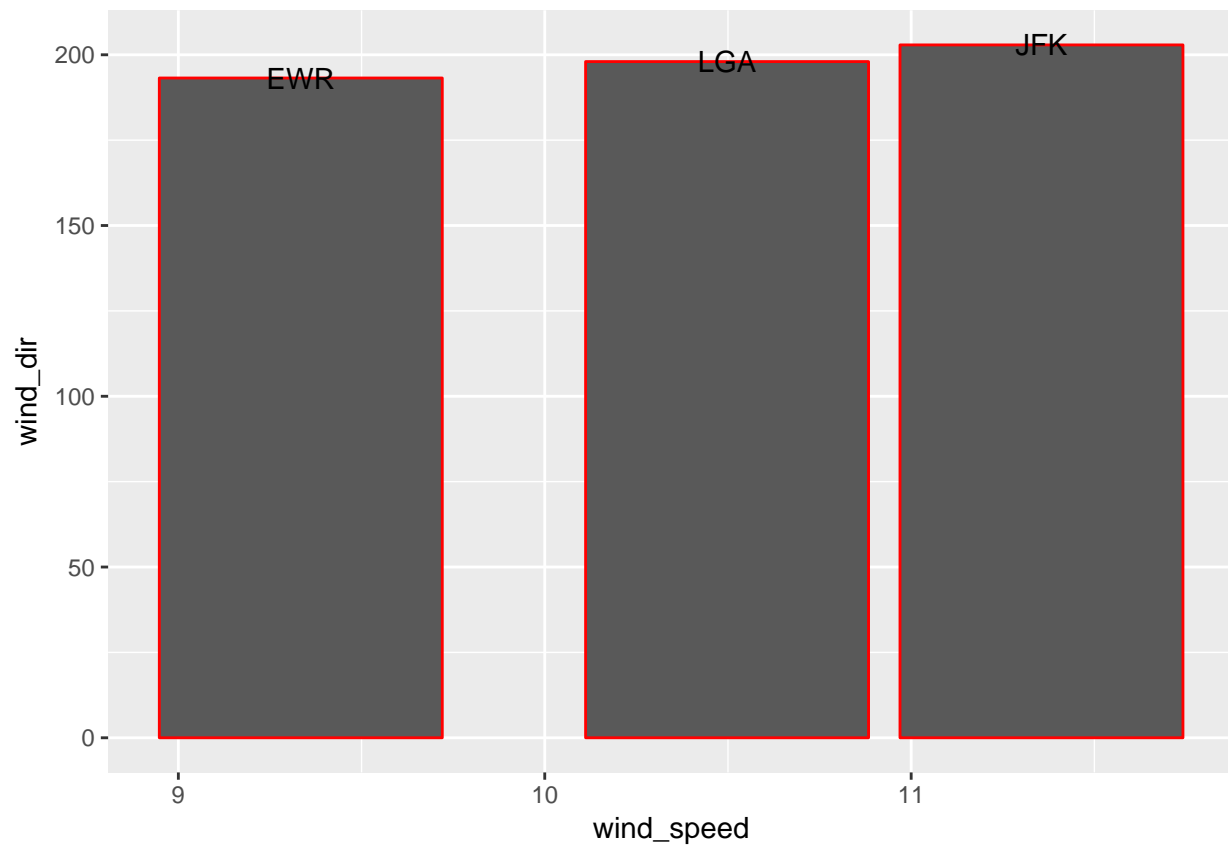
The average seats and engines per manufacturer

```
dta <- dtplanes[, .(seats = mean(seats, na.rm = TRUE),  
                    engines = mean(engines, na.rm = TRUE)), by = manufacturer]  
ggplot(dta, aes(seats, engines, label = manufacturer)) + geom_bar(stat = 'identity', color = 'red') + g
```



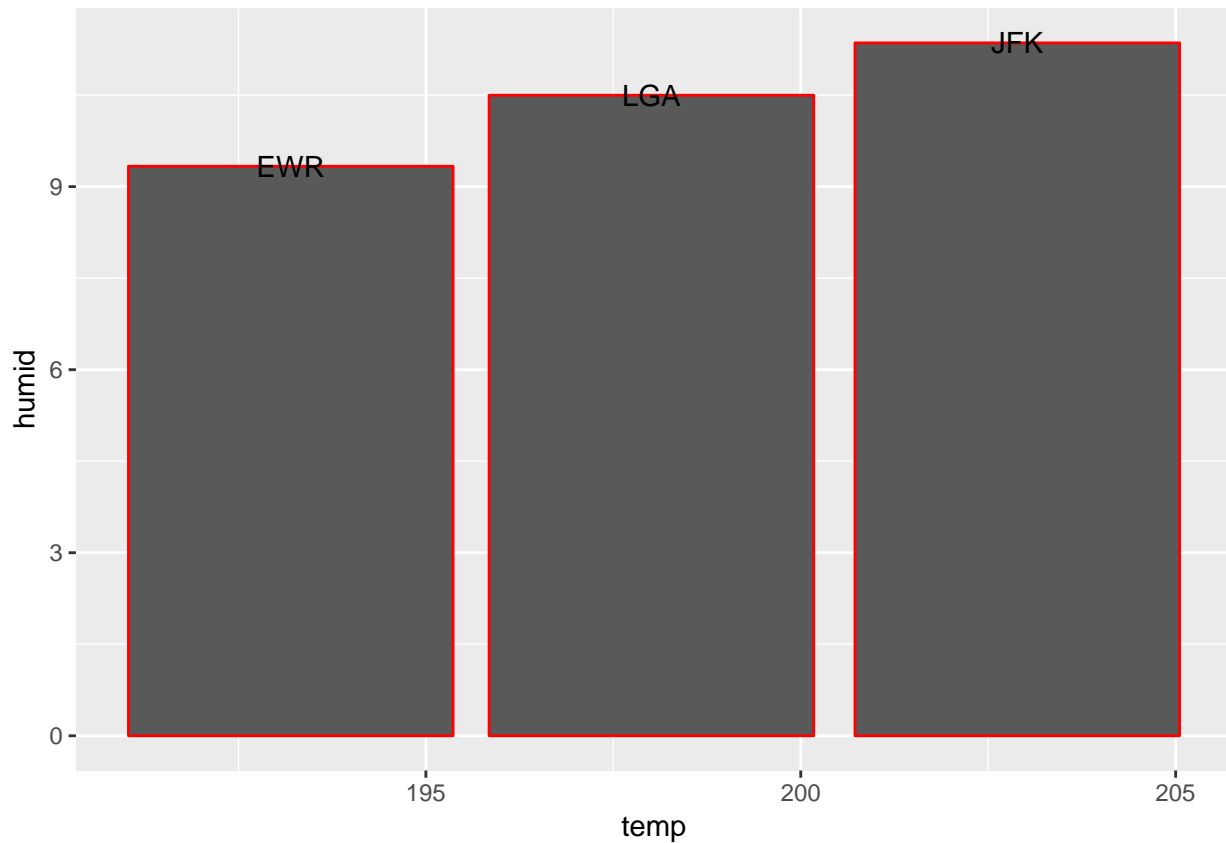
The average windspeed and wind direction per origin

```
dta <- dtweather[, .(wind_dir = mean(wind_dir, na.rm = TRUE),
                                wind_speed = mean(wind_speed, na.rm = TRUE)), by = origin]
ggplot(dta, aes(wind_speed, wind_dir, label = origin)) + geom_bar(stat = 'identity', color = 'red') + g
```



The average temperature and humidity per origin

```
dta <- dtweather[, .(temp = mean(wind_dir, na.rm = TRUE),  
                      humid = mean(wind_speed, na.rm = TRUE)), by = origin]  
ggplot(dta, aes(temp, humid, label = origin)) + geom_bar(stat = 'identity', color = 'red') + geom_text()
```



How to affect humidity or temperature on delays

Featuring

```
dtflights$hour <- ifelse(dtflights$hour == 24, 0, flights$hour)
flights_weather <- left_join(dtflights, dtweather)

## Joining, by = c("year", "month", "day", "origin", "hour", "time_hour")

flights_weather$arr_delay <- ifelse(flights_weather$arr_delay >= 0,
                                   flights_weather$arr_delay, 0)
flights_weather$dep_delay <- ifelse(flights_weather$dep_delay >= 0,
                                   flights_weather$dep_delay, 0)
flights_weather$total_delay <- flights_weather$arr_delay + flights_weather$dep_delay

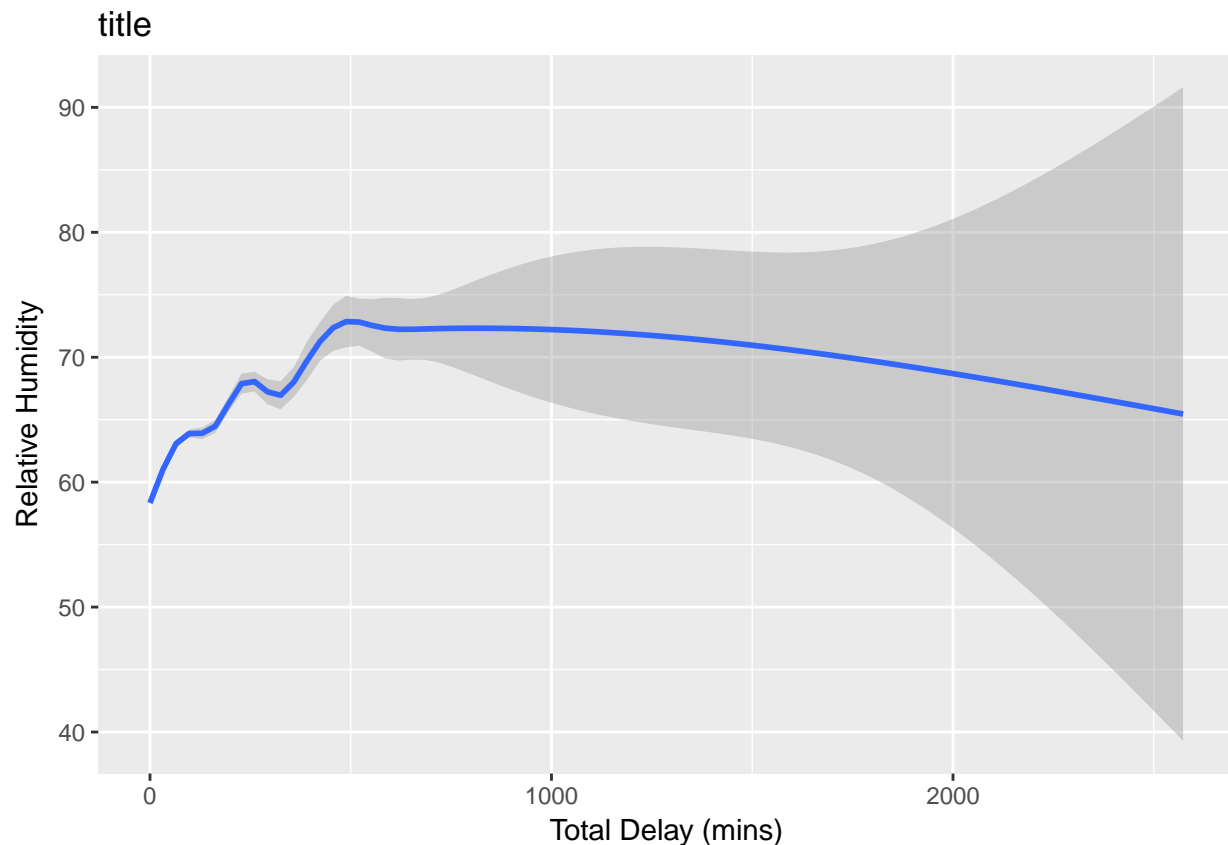
data <- select(flights_weather, total_delay, temp, dewp, humid,
               wind_dir, wind_speed, wind_gust, precip, pressure, visib)
```

Effect of the humidity on delays

```
g <- ggplot(data, aes(y = humid, x = total_delay,
                      title = "Total Delay / Humidity"))
g + geom_smooth() + ylab("Relative Humidity") +
  xlab("Total Delay (mins)")
```

```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Removed 202037 rows containing non-finite values (stat_smooth).
```

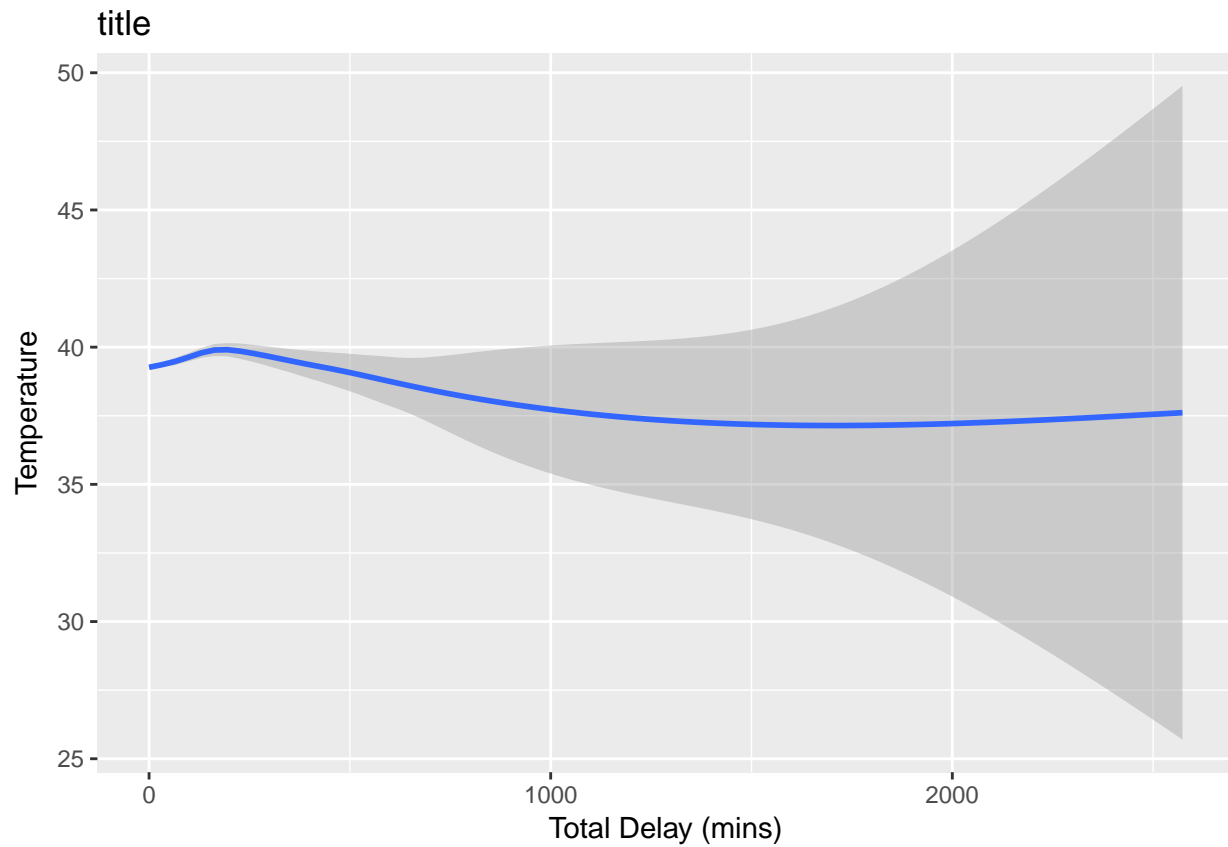


Effect of the temperature on delays

```
g <- ggplot(data, aes(y = temp, x = total_delay,
                      title = "Total Delay / Temperature"))
g + geom_smooth() + ylab("Temperature") +
  xlab("Total Delay (mins)")
```

```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Removed 202037 rows containing non-finite values (stat_smooth).
```

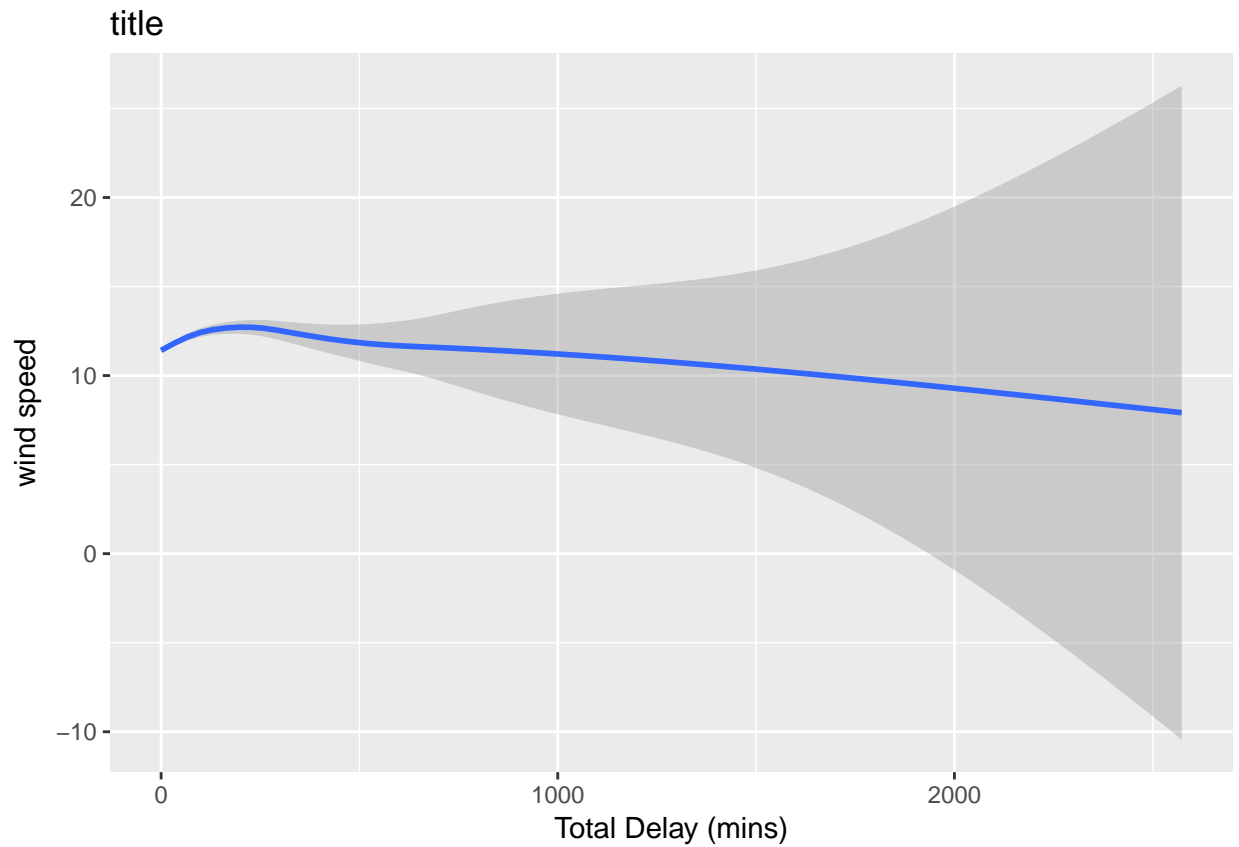



Effect of the wind speed on delays

```
g <- ggplot(data, aes(y = wind_speed, x = total_delay,
                      title = "Total Delay / wind speed"))
g + geom_smooth() + ylab("wind speed") +
  xlab("Total Delay (mins)")
```

```
## `geom_smooth()` using method = 'gam'
```

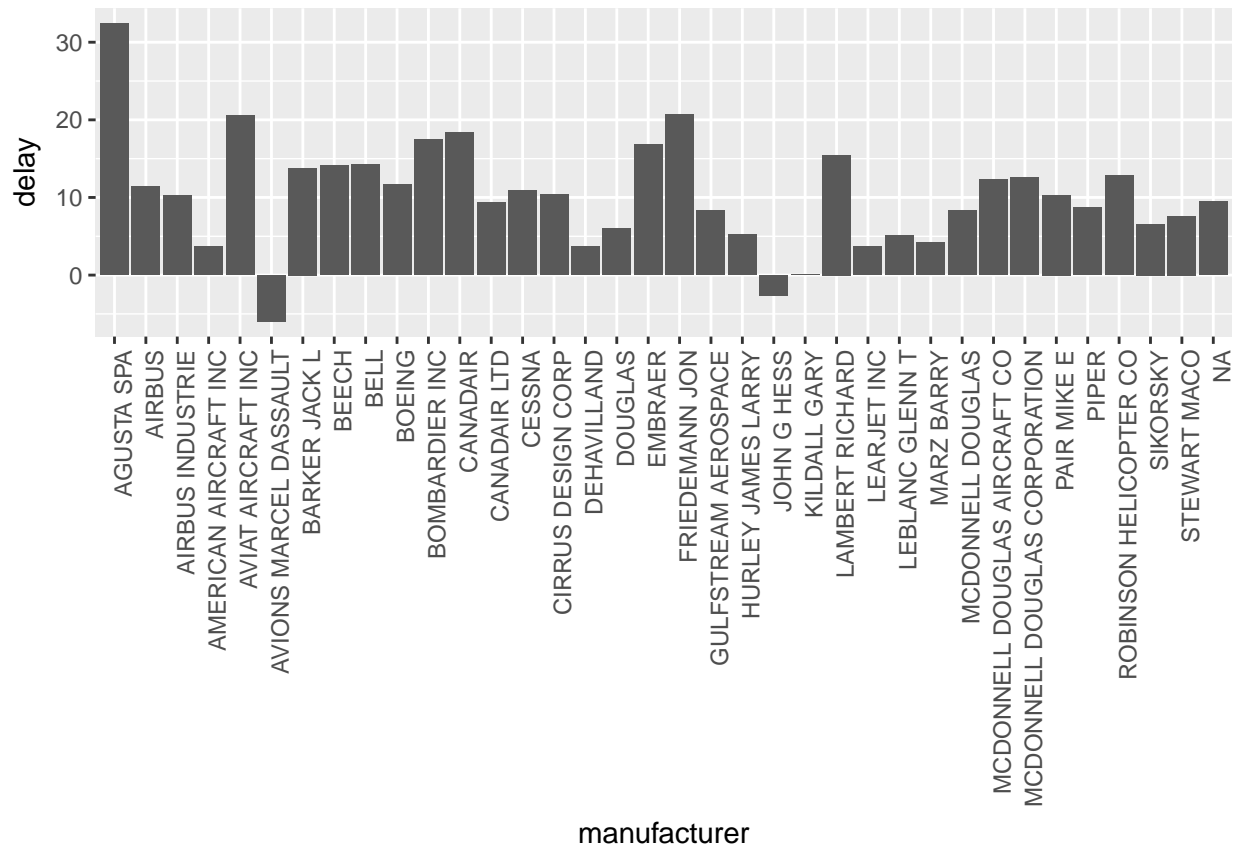
```
## Warning: Removed 202053 rows containing non-finite values (stat_smooth).
```



Average departure delay by manufacturer

```
flights_planes <- left_join(dtflights, dtplanes, by = 'tailnum')
dta <- ddply(flights_planes, ~manufacturer, summarise, delay = mean(dep_delay, na.rm = TRUE))
setorder(dta, delay)

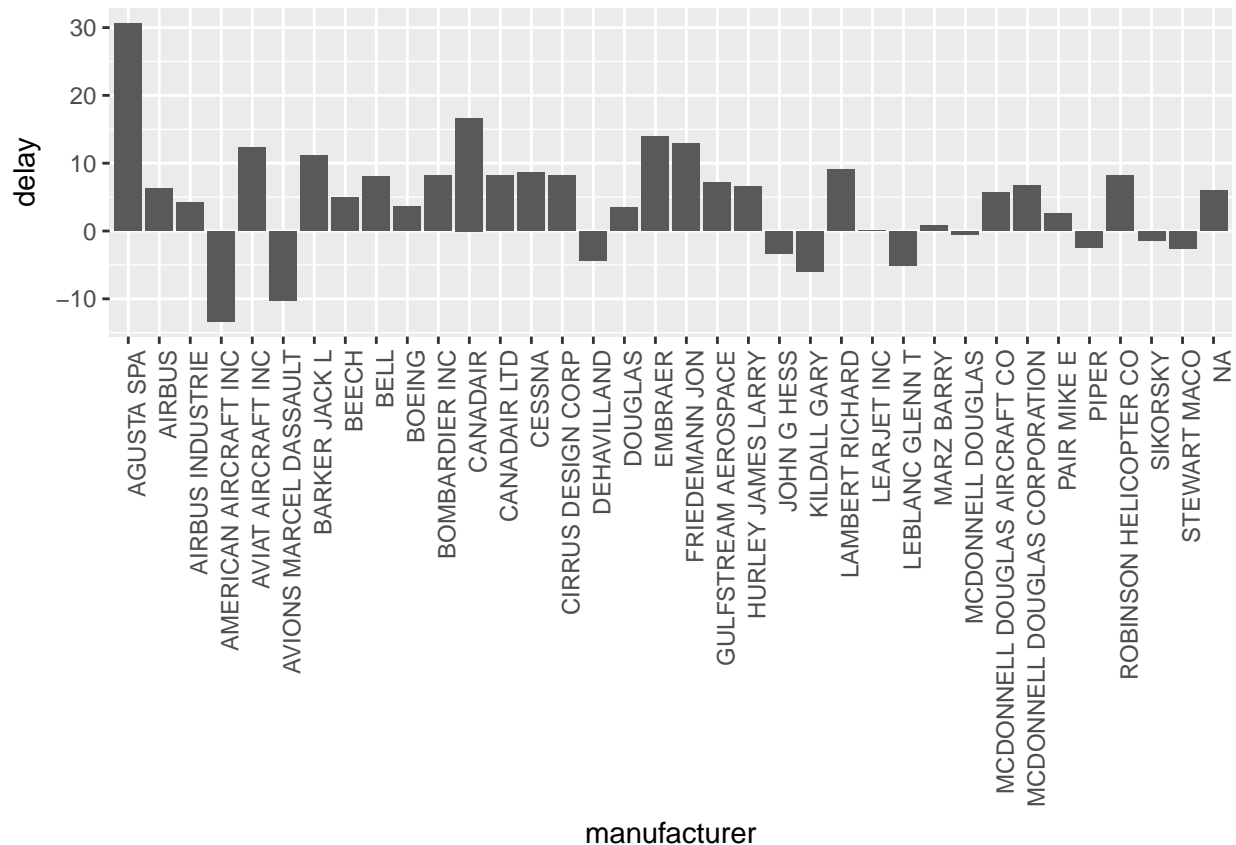
ggplot(dta, aes(manufacturer, delay)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(
```



Average arrival delay by manufacturer

```
dta <- ddply(flights_planes, ~manufacturer, summarise, delay=mean(arr_delay, na.rm=TRUE))
setorder(dta, delay)

ggplot(dta, aes(manufacturer, delay)) + geom_bar(stat = 'identity') + theme(axis.text.x = element_text(
```



More feature engineering

Data table without NA's, make binary variable which shows if an airplane delay more than 15 minutes. I decreased the observations number to 15.0000.

```
dtflights <- subset (flights, !is.na(flights$dep_time) &
  !is.na(flights$dep_delay) &
  !is.na(flights$arr_delay) &
  !is.na(flights$dep_time) &
  !is.na(flights$arr_time))
dtflights$tailnumfac <- as.factor(dtflights$tailnum)
dtflights$carrierfac <- as.factor(dtflights$carrier)
dtflights$originfac <- as.factor(dtflights$origin)
dtflights$destfac <- as.factor(dtflights$dest)

dtflights <- dtflights[sample(1:nrow(dtflights), 15000, replace=FALSE),]

dtflights$tailnumnum <- as.numeric(dtflights$tailnumfac)
dtflights$carriernum <- as.numeric(dtflights$carrierfac)
dtflights$originnum <- as.numeric(dtflights$originfac)
dtflights$destnum <- as.numeric(dtflights$destfac)

dtflights$year <- NULL
dtflights$tail_num <- NULL
dtflights$tailnum <- NULL
dtflights$carrier <- NULL
```

```

dtflights$dest <- NULL
dtflights$origin <- NULL
dtflights$tailnum <- NULL
dtflights$tailnumfac <- NULL
dtflights$carrierfac <- NULL
dtflights$destfac <- NULL
dtflights$originfac <- NULL
dtflights$tailnumfac <- NULL

str(dtflights)

```

```

## Classes 'tbl_df', 'tbl' and 'data.frame': 15000 obs. of 18 variables:
## $ month      : int  7 2 1 7 1 12 8 10 11 6 ...
## $ day        : int  29 7 29 18 11 5 19 15 30 24 ...
## $ dep_time   : int  1005 1954 1900 751 954 2001 2121 1458 655 1941 ...
## $ sched_dep_time: int  1008 2000 1905 800 1000 1900 2130 1455 700 1735 ...
## $ dep_delay  : num  -3 -6 -5 -9 -6 61 -9 3 -5 126 ...
## $ arr_time   : int  1213 2117 2036 859 1236 2128 6 1615 1019 2228 ...
## $ sched_arr_time: int  1215 2139 2040 913 1321 2043 14 1613 1015 2030 ...
## $ arr_delay  : num  -2 -22 -4 -14 -45 45 -8 2 4 118 ...
## $ flight     : int  2319 1094 981 2165 499 654 1371 1164 23 1901 ...
## $ air_time   : num  144 62 132 48 322 119 144 35 351 182 ...
## $ distance   : num  1020 404 738 214 2402 ...
## $ hour       : num  10 20 19 8 10 19 21 14 7 17 ...
## $ minute     : num  8 0 5 0 0 0 30 55 0 35 ...
## $ time_hour   : POSIXct, format: "2013-07-29 10:00:00" "2013-02-07 20:00:00" ...
## $ tailnumnum  : num  1031 123 3098 2961 1582 ...
## $ carriernum  : num  5 12 15 13 12 12 4 5 4 2 ...
## $ originnum   : num  3 1 3 3 1 1 3 2 2 2 ...
## $ destnum     : num  61 23 59 29 89 69 36 12 50 44 ...

```

```

dtflights$arrdelay15 <- ifelse(dtflights$arr_delay > 15,1,0)
dtflights$year <- NULL
time_format <- "%Y-%m-%d %H:%M:%S"
dtflights$weekday <- as.factor(format(strptime(dtflights$time_hour, format=time_format),"%A"))
dtflights$time_hour <- NULL

```

2-Nearest Neighbors algorithm

```

dtflights$rnd <- runif(dim(dtflights[1]))
dtflights <- dtflights[order(dtflights$rnd),]

```

```
dtflights
```

```

## # A tibble: 15,000 × 20
##   month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int>   <int>         <int>         <dbl>   <int>         <int>
## 1     7    29     1005           1008          -3     1213           1215
## 2     1    29     1900           1905          -5     2036           2040

```

```
## 3      1    11      954          1000          -6      1236          1321
## 4      8    19     2121          2130          -9        6           14
## 5     11    30      655           700          -5     1019          1015
## 6      8    21     1227          1230          -3     1539          1558
## 7     11     7      625           632          -7      854           854
## 8      8    21     1553          1559          -6     1821          1843
## 9      8    21      722           615          67     1027           910
## 10     4     4      639           649         -10      840           920
## # ... with 14,990 more rows, and 13 more variables: arr_delay <dbl>,
## #   flight <int>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, tailnumnum <dbl>, carriernum <dbl>, originnum <dbl>,
## #   destnum <dbl>, arrdelay15 <dbl>, weekday <fctr>, rnd <dbl>
```

```
train <- dtflights[0:round((dim(dtflights)[1])*0.7),]
test <- dtflights[(round((dim(dtflights)[1])*0.7)+1):(dim(dtflights)[1]),]
dtflights$rnd <- NULL

fit <- knn(train[,1:15], test[,1:15], train$arrdelay15, k = 2)

pander(table(test$arrdelay15,fit))
```

	0	1
0	3106	353
1	579	462

5-Nearest Neighbors algorithm

```
fit2 <- knn(train[,1:15], test[,1:15], train$arrdelay15, k = 5)
pander(table(test$arrdelay15,fit2))
```

	0	1
0	3330	129
1	738	303

```
total <- dim(test)[1]
```

All in all the 2-NN model has provided 78 % result, the 5- NN model has provided 80 % good result.

Modeling

CLEAR MEMORY

```
rm(list = ls())
```

```
library(h2o)
```

```
## Warning: package 'h2o' was built under R version 3.3.2

##
## -----
##
## Your next step is to start H2O:
##   > h2o.init()
##
## For H2O package documentation, ask for help:
##   > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit http://docs.h2o.ai
##
## -----

##
## Attaching package: 'h2o'

## The following objects are masked from 'package:lubridate':
##
##   day, hour, month, week, year

## The following objects are masked from 'package:data.table':
##
##   hour, month, week, year

## The following objects are masked from 'package:stats':
##
##   cor, sd, var

## The following objects are masked from 'package:base':
##
##   &&, %*%, %in%, ||, apply, as.factor, as.numeric, colnames,
##   colnames<-, ifelse, is.character, is.factor, is.numeric, log,
##   log10, log1p, log2, round, signif, trunc
```

```
h2o.init()
```

```
##
## H2O is not running yet, starting it now...
##
## Note: In case of errors look at the following log files:
##   /var/folders/_2/ny9pbkp90zb9ks3c034xd0j80000gn/T//RtmpbYXB8d/h2o_Attila_started_from_r.out
##   /var/folders/_2/ny9pbkp90zb9ks3c034xd0j80000gn/T//RtmpbYXB8d/h2o_Attila_started_from_r.err
##
##
## Starting H2O JVM and connecting: ..... Connection successful!
##
```

```
## R is connected to the H2O cluster:
##   H2O cluster uptime:      4 seconds 478 milliseconds
##   H2O cluster version:    3.10.3.3
##   H2O cluster version age: 23 days
##   H2O cluster name:       H2O_started_from_R_Attila_nzi117
##   H2O cluster total nodes: 1
##   H2O cluster total memory: 0.12 GB
##   H2O cluster total cores: 4
##   H2O cluster allowed cores: 2
##   H2O cluster healthy:    TRUE
##   H2O Connection ip:      localhost
##   H2O Connection port:    54321
##   H2O Connection proxy:   NA
##   R Version:              R version 3.3.1 (2016-06-21)
##
## Note:  As started, H2O is limited to the CRAN default of 2 CPUs.
##        Shut down and restart H2O as shown below to use all your CPUs.
##        > h2o.shutdown()
##        > h2o.init(nthreads = -1)
```

write demo data to disk

```
library(nycflights13)
write.csv(flights, 'flights.csv', row.names = FALSE)
flights.hex <- h2o.uploadFile('flights.csv', destination_frame = 'flights')
```

```
##
|
|
|
|=====| 100%
```

```
str(flights.hex)
```

```
## Class 'H2OFrame' <environment: 0x7fe14a748f18>
## - attr(*, "op")= chr "Parse"
## - attr(*, "id")= chr "flights"
## - attr(*, "eval")= logi FALSE
## - attr(*, "nrow")= int 336776
## - attr(*, "ncol")= int 19
## - attr(*, "types")=List of 19
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "enum"
```



```
## ..$ : chr "int"
## ..$ : chr "enum"
## ..$ : chr "enum"
## ..$ : chr "enum"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "int"
## ..$ : chr "time"
## - attr(*, "data")='data.frame': 10 obs. of 19 variables:
## ..$ year : num 2013 2013 2013 2013 2013 ...
## ..$ month : num 1 1 1 1 1 1 1 1 1 1
## ..$ day : num 1 1 1 1 1 1 1 1 1 1
## ..$ dep_time : num 517 533 542 544 554 554 555 557 557 558
## ..$ sched_dep_time: num 515 529 540 545 600 558 600 600 600 600
## ..$ dep_delay : num 2 4 2 -1 -6 -4 -5 -3 -3 -2
## ..$ arr_time : num 830 850 923 1004 812 ...
## ..$ sched_arr_time: num 819 830 850 1022 837 ...
## ..$ arr_delay : num 11 20 33 -18 -25 12 19 -14 -8 8
## ..$ carrier : Factor w/ 16 levels "9E","AA","AS",...: 12 12 2 4 5 12 4 6 4 2
## ..$ flight : num 1545 1714 1141 725 461 ...
## ..$ tailnum : Factor w/ 4044 levels "D942DN","NOEGMQ",...: 180 524 2401 3204 2661 1142 1829 3
## ..$ origin : Factor w/ 3 levels "EWR","JFK","LGA": 1 3 2 2 3 1 1 3 2 3
## ..$ dest : Factor w/ 105 levels "ABQ","ACK","ALB",...: 44 44 59 13 5 70 36 43 55 70
## ..$ air_time : num 227 227 160 183 116 150 158 53 140 138
## ..$ distance : num 1400 1416 1089 1576 762 ...
## ..$ hour : num 5 5 5 5 6 5 6 6 6 6
## ..$ minute : num 15 29 40 45 0 58 0 0 0 0
## ..$ time_hour : num 1.36e+12 1.36e+12 1.36e+12 1.36e+12 1.36e+12 ...
```

```
head(flights.hex)
```

```
## year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1 2013 1 1 517 515 2 830 819
## 2 2013 1 1 533 529 4 850 830
## 3 2013 1 1 542 540 2 923 850
## 4 2013 1 1 544 545 -1 1004 1022
## 5 2013 1 1 554 600 -6 812 837
## 6 2013 1 1 554 558 -4 740 728
## arr_delay carrier flight tailnum origin dest air_time distance hour
## 1 11 UA 1545 N14228 EWR IAH 227 1400 5
## 2 20 UA 1714 N24211 LGA IAH 227 1416 5
## 3 33 AA 1141 N619AA JFK MIA 160 1089 5
## 4 -18 B6 725 N804JB JFK BQN 183 1576 5
## 5 -25 DL 461 N668DN LGA ATL 116 762 6
## 6 12 UA 1696 N39463 EWR ORD 150 719 5
## minute time_hour
## 1 15 1.357013e+12
## 2 29 1.357013e+12
## 3 40 1.357013e+12
## 4 45 1.357013e+12
## 5 0 1.357016e+12
## 6 58 1.357013e+12
```

```
head(flights.hex, 3)
```

```
##   year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
## 1 2013     1   1      517           515         2      830           819
## 2 2013     1   1      533           529         4      850           830
## 3 2013     1   1      542           540         2      923           850
##   arr_delay carrier flight tailnum origin dest air_time distance hour
## 1         11      UA   1545 N14228  EWR  IAH      227      1400    5
## 2         20      UA   1714 N24211  LGA  IAH      227      1416    5
## 3         33      AA   1141 N619AA   JFK  MIA      160      1089    5
##   minute    time_hour
## 1      15 1.357013e+12
## 2      29 1.357013e+12
## 3      40 1.357013e+12
```

```
summary(flights.hex)
```

```
## Warning in summary.H2OFrame(flights.hex): Approximated quantiles
## computed! If you are interested in exact quantiles, please pass the
## `exact_quantiles=TRUE` parameter.
```

```
##   year          month          day          dep_time
## Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   : 1.0
## 1st Qu.: NaN    1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 905.8
## Median : NaN    Median : 7.000   Median :16.00   Median :1400.2
## Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349.1
## 3rd Qu.: NaN    3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1743.4
## Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400.0
##                                     NA's   :8255
##   sched_dep_time  dep_delay      arr_time      sched_arr_time
## Min.   : 106.0    Min.   : -43.00   Min.   : 1      Min.   : 1
## 1st Qu.: 903.9    1st Qu.: -5.34   1st Qu.:1103    1st Qu.:1124
## Median :1357.0    Median : -2.65   Median :1535    Median :1556
## Mean   :1344.3    Mean   : 12.64   Mean   :1502    Mean   :1536
## 3rd Qu.:1728.9    3rd Qu.: 10.80   3rd Qu.:1938    3rd Qu.:1945
## Max.   :2359.0    Max.   :1301.00   Max.   :2400    Max.   :2359
##                                     NA's   :8255   NA's   :8713
##   arr_delay      carrier  flight      tailnum      origin
## Min.   : -86.000   UA:58665   Min.   : 1      N725MQ: 575   EWR:120835
## 1st Qu.: -18.050   B6:54635   1st Qu.: 545    N722MQ: 513   JFK:111279
## Median : -5.819   EV:54173   Median :1488    N723MQ: 507   LGA:104662
## Mean   :  6.895   DL:48110   Mean   :1972    N711MQ: 486
## 3rd Qu.: 13.207   AA:32729   3rd Qu.:3460    N713MQ: 483
## Max.   :1272.000   MQ:26397   Max.   :8500    N258JB: 427
## NA's   :9430                                     NA   :2512
##   dest      air_time      distance      hour
## ORD:17283   Min.   : 20.0    Min.   : 17.0   Min.   : 1.00
## ATL:17215   1st Qu.: 82.0    1st Qu.: 498.8   1st Qu.: 9.00
## LAX:16174   Median :129.0    Median : 871.3   Median :13.00
## BOS:15508   Mean   :150.7    Mean   :1039.9   Mean   :13.18
## MCO:14082   3rd Qu.:192.0    3rd Qu.:1387.9   3rd Qu.:17.00
## CLT:14064   Max.   :695.0    Max.   :4983.0   Max.   :23.00
```

```
##           NA's      :9430
## minute      time_hour
## Min.       : 0.00
## 1st Qu.: 8.00
## Median :29.00
## Mean      :26.23
## 3rd Qu.:44.00
## Max.      :59.00
##
```

convert numeric to factor/enum

```
flights.hex[, 'flight'] <- as.factor(flights.hex[, 'flight'])
summary(flights.hex)
```

```
## Warning in summary.H2OFrame(flights.hex): Approximated quantiles
## computed! If you are interested in exact quantiles, please pass the
## `exact_quantiles=TRUE` parameter.
```

```
## year      month      day      dep_time
## Min.      :2013      Min.      : 1.000      Min.      : 1.00      Min.      : 1.0
## 1st Qu.: NaN      1st Qu.: 4.000      1st Qu.: 8.00      1st Qu.: 905.8
## Median : NaN      Median : 7.000      Median :16.00      Median :1400.2
## Mean      :2013      Mean      : 6.549      Mean      :15.71      Mean      :1349.1
## 3rd Qu.: NaN      3rd Qu.:10.000      3rd Qu.:23.00      3rd Qu.:1743.4
## Max.      :2013      Max.      :12.000      Max.      :31.00      Max.      :2400.0
##                                     NA's      :8255
## sched_dep_time  dep_delay      arr_time      sched_arr_time
## Min.      : 106.0      Min.      : -43.00      Min.      : 1      Min.      : 1
## 1st Qu.: 903.9      1st Qu.: -5.34      1st Qu.:1103      1st Qu.:1124
## Median :1357.0      Median : -2.65      Median :1535      Median :1556
## Mean      :1344.3      Mean      : 12.64      Mean      :1502      Mean      :1536
## 3rd Qu.:1728.9      3rd Qu.: 10.80      3rd Qu.:1938      3rd Qu.:1945
## Max.      :2359.0      Max.      :1301.00      Max.      :2400      Max.      :2359
##                                     NA's      :8255      NA's      :8713
## arr_delay      carrier  flight  tailnum      origin      dest
## Min.      : -86.000      UA:58665  15 :968  N725MQ: 575      EWR:120835  ORD:17283
## 1st Qu.: -18.050      B6:54635  27 :898  N722MQ: 513      JFK:111279  ATL:17215
## Median : -5.819      EV:54173  181:882  N723MQ: 507      LGA:104662  LAX:16174
## Mean      : 6.895      DL:48110  301:871  N711MQ: 486                        BOS:15508
## 3rd Qu.: 13.207      AA:32729  161:786  N713MQ: 483                        MCO:14082
## Max.      :1272.000      MQ:26397  695:782  N258JB: 427                        CLT:14064
## NA's      :9430                        NA      :2512
## air_time      distance      hour      minute
## Min.      : 20.0      Min.      : 17.0      Min.      : 1.00      Min.      : 0.00
## 1st Qu.: 82.0      1st Qu.: 498.8      1st Qu.: 9.00      1st Qu.: 8.00
## Median :129.0      Median : 871.3      Median :13.00      Median :29.00
## Mean      :150.7      Mean      :1039.9      Mean      :13.18      Mean      :26.23
## 3rd Qu.:192.0      3rd Qu.:1387.9      3rd Qu.:17.00      3rd Qu.:44.00
## Max.      :695.0      Max.      :4983.0      Max.      :23.00      Max.      :59.00
## NA's      :9430
```

```
## time_hour
##
##
##
##
##
##
##
```

```
flights.hex$flight <- as.factor(flights.hex$flight)
for (v in c('month', 'day', 'dep_delay', 'arr_delay')) {
  flights.hex[, v] <- as.factor(flights.hex[, v])
}
summary(flights.hex)
```

```
## Warning in summary.H2OFrame(flights.hex): Approximated quantiles
## computed! If you are interested in exact quantiles, please pass the
## `exact_quantiles=TRUE` parameter.
```

```
## year      month      day      dep_time      sched_dep_time
## Min.      :2013      7 :29425  18:11399  Min.      : 1.0  Min.      : 106.0
## 1st Qu.: NaN      8 :29327  11:11359  1st Qu.: 905.8  1st Qu.: 903.9
## Median : NaN     10:28889  22:11345  Median :1400.2  Median :1357.0
## Mean      :2013      3 :28834  15:11317  Mean      :1349.1  Mean      :1344.3
## 3rd Qu.: NaN      5 :28796   8 :11271  3rd Qu.:1743.4  3rd Qu.:1728.9
## Max.      :2013      4 :28330  10:11227  Max.      :2400.0  Max.      :2359.0
##
##                      NA's      :8255
## dep_delay arr_time      sched_arr_time arr_delay carrier  flight
## -5:24821  Min.      : 1  Min.      : 1  -13:7177  UA:58665  15 :968
## -4:24619  1st Qu.:1103  1st Qu.:1124  -10:7088  B6:54635  27 :898
## -3:24218  Median :1535  Median :1556  -12:7046  EV:54173  181:882
## -2:21516  Mean      :1502  Mean      :1536  -14:6975  DL:48110  301:871
## -6:20701  3rd Qu.:1938  3rd Qu.:1945  -11:6863  AA:32729  161:786
## -1:18813  Max.      :2400  Max.      :2359  -9 :6815  MQ:26397  695:782
## NA: 8255  NA's      :8713                      NA :9430
## tailnum    origin      dest      air_time      distance
## N725MQ: 575  EWR:120835  ORD:17283  Min.      : 20.0  Min.      : 17.0
## N722MQ: 513  JFK:111279  ATL:17215  1st Qu.: 82.0  1st Qu.: 498.8
## N723MQ: 507  LGA:104662  LAX:16174  Median :129.0  Median : 871.3
## N711MQ: 486                      BOS:15508  Mean      :150.7  Mean      :1039.9
## N713MQ: 483                      MCO:14082  3rd Qu.:192.0  3rd Qu.:1387.9
## N258JB: 427                      CLT:14064  Max.      :695.0  Max.      :4983.0
## NA      :2512                      NA's      :9430
## hour      minute      time_hour
## Min.      : 1.00  Min.      : 0.00
## 1st Qu.: 9.00  1st Qu.: 8.00
## Median :13.00  Median :29.00
## Mean      :13.18  Mean      :26.23
## 3rd Qu.:17.00  3rd Qu.:44.00
## Max.      :23.00  Max.      :59.00
##
```

drop columns

```
dt <- data.table(flights)
dt$delay15 <- ifelse(dt$arr_delay > 15,1,0)
str(dt)
```

```
## Classes 'data.table' and 'data.frame':  336776 obs. of  20 variables:
## $ year      : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
## $ month     : int   1  1  1  1  1  1  1  1  1  1 ...
## $ day       : int   1  1  1  1  1  1  1  1  1  1 ...
## $ dep_time  : int  517 533 542 544 554 554 555 557 557 558 ...
## $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
## $ dep_delay : num   2  4  2 -1 -6 -4 -5 -3 -3 -2 ...
## $ arr_time  : int  830 850 923 1004 812 740 913 709 838 753 ...
## $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
## $ arr_delay : num  11 20 33 -18 -25 12 19 -14 -8 8 ...
## $ carrier   : chr  "UA" "UA" "AA" "B6" ...
## $ flight    : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
## $ tailnum   : chr  "N14228" "N24211" "N619AA" "N804JB" ...
## $ origin    : chr  "EWR" "LGA" "JFK" "JFK" ...
## $ dest      : chr  "IAH" "IAH" "MIA" "BQN" ...
## $ air_time  : num  227 227 160 183 116 150 158 53 140 138 ...
## $ distance  : num  1400 1416 1089 1576 762 ...
## $ hour      : num   5  5  5  5  6  5  6  6  6  6 ...
## $ minute    : num  15 29 40 45  0 58  0  0  0  0 ...
## $ time_hour : POSIXct, format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
## $ delay15   : num   0  1  1  0  0  0  1  0  0  0 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

```
dt <- dt[, .(month, day, dest, origin,
             carrier, flight, tailnum, distance, delay15)]
```

transform to factor

```
for (v in c('month', 'day', 'flight', 'carrier')) {
  set(dt, j = v, value = as.factor(dt[, get(v)]))
}
str(dt)
```

```
## Classes 'data.table' and 'data.frame':  336776 obs. of  9 variables:
## $ month     : Factor w/ 12 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ day       : Factor w/ 31 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ dest      : chr  "IAH" "IAH" "MIA" "BQN" ...
## $ origin    : chr  "EWR" "LGA" "JFK" "JFK" ...
## $ carrier   : Factor w/ 16 levels "9E","AA","AS",...: 12 12 2 4 5 12 4 6 4 2 ...
## $ flight    : Factor w/ 3844 levels "1","2","3","4",...: 1382 1545 1042 677 425 1527 469 3700 69 266 .
## $ tailnum   : chr  "N14228" "N24211" "N619AA" "N804JB" ...
## $ distance  : num  1400 1416 1089 1576 762 ...
## $ delay15   : num   0  1  1  0  0  0  1  0  0  0 ...
## - attr(*, ".internal.selfref")=<externalptr>
```

re-upload to H2O

```
h2o.ls()
```

```
##           key
## 1 RTMP_sid_a861_7
## 2         flights
```

```
h2o.rm('flights')
as.h2o(dt, 'flights')
```

```
##
|
|                                     | 0%
|
|=====| 100%
```

```
##  month day dest origin carrier flight tailnum distance delay15
## 1     1   1  IAH   EWR      UA   1545  N14228     1400      0
## 2     1   1  IAH   LGA      UA   1714  N24211     1416      1
## 3     1   1  MIA   JFK      AA   1141  N619AA     1089      1
## 4     1   1  BQN   JFK      B6    725  N804JB     1576      0
## 5     1   1  ATL   LGA      DL    461  N668DN      762      0
## 6     1   1  ORD   EWR      UA   1696  N39463      719      0
##
## [336776 rows x 9 columns]
```

split the data

```
flights.hex <- h2o.getFrame('flights')
h2o.splitFrame(data = flights.hex , ratios = 0.75, destination_frames = c('train', 'test'))
```

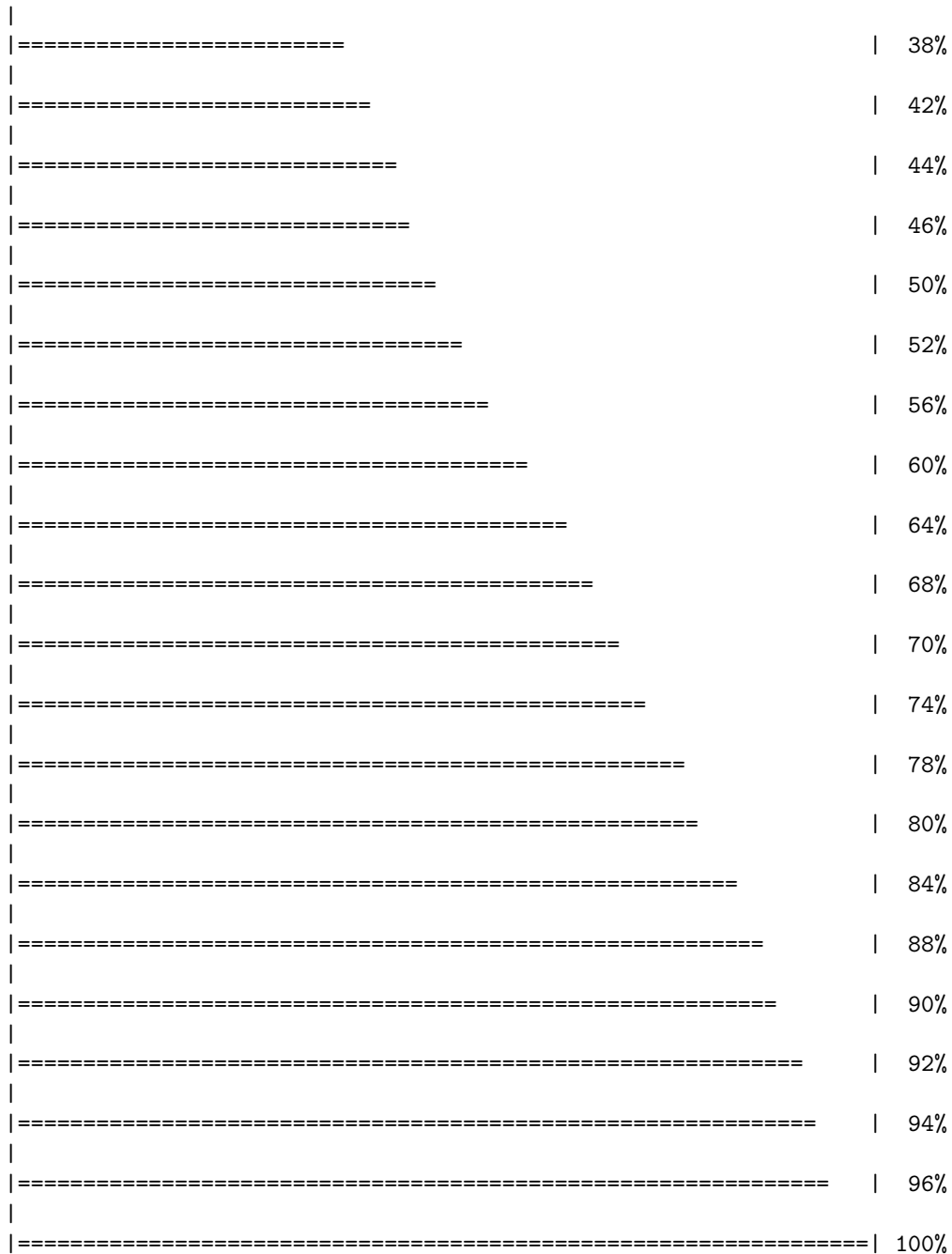
```
## [[1]]
##  month day dest origin carrier flight tailnum distance delay15
## 1     1   1  IAH   LGA      UA   1714  N24211     1416      1
## 2     1   1  MIA   JFK      AA   1141  N619AA     1089      1
## 3     1   1  BQN   JFK      B6    725  N804JB     1576      0
## 4     1   1  ATL   LGA      DL    461  N668DN      762      0
## 5     1   1  ORD   EWR      UA   1696  N39463      719      0
## 6     1   1  FLL   EWR      B6    507  N516JB     1065      1
##
## [252774 rows x 9 columns]
##
## [[2]]
##  month day dest origin carrier flight tailnum distance delay15
## 1     1   1  IAH   EWR      UA   1545  N14228     1400      0
## 2     1   1  TPA   JFK      B6     71  N657JB     1005      0
## 3     1   1  SFO   EWR      UA   1124  N53441     2565      0
## 4     1   1  BOS   JFK      B6   1806  N708JB      187      0
```

h2o.ls()

build the first model

```
## Warning in .h2o.startModelJob(algo, params, h2oRestApiVersion): Dropping constant columns: [origin, ...]
```

##		
		0%
=		2%
===		4%
====		6%
=====		10%
=====		14%
=====		18%
=====		22%
=====		24%
=====		26%
=====		32%
=====		34%



flights.rf

```
## Model Details:
## =====
##
## H2ORegressionModel: drf
## Model ID: DRF_model_R_1488049732289_1
## Model Summary:
##   number_of_trees number_of_internal_trees model_size_in_bytes min_depth
## 1           50           50           9252386           20
##   max_depth mean_depth min_leaves max_leaves mean_leaves
```



```

## 1      20    20.00000      3124      10503  7067.20000
##
##
## H2ORegressionMetrics: drf
## ** Reported on training data. **
## ** Metrics reported on Out-Of-Bag training samples **
##
## MSE:  0.1575037
## RMSE:  0.3968674
## MAE:  0.3221067
## RMSLE:  0.2772203
## Mean Residual Deviance :  0.1575037
##
##
## H2ORegressionMetrics: drf
## ** Reported on validation data. **
##
## MSE:  0.1555002
## RMSE:  0.3943351
## MAE:  0.3213388
## RMSLE:  0.2753863
## Mean Residual Deviance :  0.1555002

```