Datathon 2018 : The Telenor Case

The Changemakers

9/29/2018

Reading CSV file for Telenor data and loading required libraries :

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

```
## Warning: package 'dplyr' was built under R version 3.5.1
```

```
##
## 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(bit64)
```

```
## Loading required package: bit
```

```
## Attaching package bit
```

```
## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL-2)
```

```
## creators: bit bitwhich
```

```
## coercion: as.logical as.integer as.bit as.bitwhich which
```

```
## operator: ! & | xor != ==
```

```
## querying: print length any all min max range sum summary
```

```
## bit access: length<- [ [<- [[ [[<-
```

```
## for more help type ?bit
```

```
##
## Attaching package: 'bit'
```

```
## The following object is masked from 'package:data.table':
##
##     setattr
```

```
## The following object is masked from 'package:base':
##
##     xor
## Attaching package bit64
```

```
## package:bit64 (c) 2011-2012 Jens Oehlschlaegel
```

```
## creators: integer64 seq :
```

```
## coercion: as.integer64 as.vector as.logical as.integer as.double as.character as.bin
```

```
## logical operator: ! & | xor != == < <= >= >
```

```
## arithmetic operator: + - * / %/% %% ^
```

```
## math: sign abs sqrt log log2 log10
```

```
## math: floor ceiling trunc round
```

```
## querying: is.integer64 is.vector [is.atomic} [length] format print str
```

```
## values: is.na is.nan is.finite is.infinite
```

```
## aggregation: any all min max range sum prod
```

```
## cumulation: diff cummin cummax cumsum cumprod
```

```
## access: length<- [ [<- [[ [[<-
```

```
## combine: c rep cbind rbind as.data.frame
```

```
## WARNING don't use as subscripts
```

```
## WARNING semantics differ from integer
```

```
## for more help type ?bit64
```

```
##
## Attaching package: 'bit64'
```

```
## The following object is masked from 'package:bit':
##
##     still.identical
```

```
## The following objects are masked from 'package:base':
##
##     %in%, :, is.double, match, order, rank
```

```r
library(tidyr)
library(ggplot2)
library(forecast)
library(tseries)
data = fread(input = "/Users/prafuldev/Desktop/data.csv")
head(data,10)
```

```
##             DATETIME                       RAVEN_NAME FAMILY_NAME
##  1: 2018-07-06 10:15  Razzle Dazzle Rose raven Shadow   Targerian
##  2: 2018-07-06 19:30         Tiny Insolent raven Lulu   Targerian
##  3: 2018-07-06 18:30    Blithe And Beautiful raven Mo   Targerian
##  4: 2018-07-06 00:45 Illuminating Emerald raven Daffy   Lannister
##  5: 2018-07-06 12:00    Blithe And Beautiful raven Mo     Greyjoy
##  6: 2018-07-06 16:15    Blithe And Beautiful raven Mo     Greyjoy
##  7: 2018-07-06 03:45                 Goofy raven Cleo     Baelish
##  8: 2018-07-06 04:45       Shocking Pink raven Wilbur   Lannister
##  9: 2018-07-06 10:30       Shocking Pink raven Wilbur   Targerian
## 10: 2018-07-06 12:15       Shocking Pink raven Wilbur     Greyjoy
##       MEMBER_NAME NETWORK FIRST_GET_RESPONSE_SUCCESS_D PAGE_BROWSING_DELAY
##  1: Maester Aemon      2g                          644                2832
##  2: Maester Aemon      4g                          258               51752
##  3:       Rheagar      3g                         1094               19562
##  4:         Kevan      2g                          432                1488
##  5:         Theon      4g                         1094               25132
##  6:         Aeron      2g                        44856               51494
##  7: Petyr Baelish      4g                          434                3018
##  8:        Cersei      2g                        17138               24412
##  9: Maester Aemon      2g                            0                1760
## 10:         Balon      2g                         1286                3834
##     TCP_SETUP_TOTAL_DELAY PAGE_CONTENT_DOWNLOAD_TOTAL_D
##  1:                  2100                           730
##  2:                   260                         51334
##  3:                   470                         17130
##  4:                   558                           638
##  5:                   576                         13054
##  6:                   436                         50576
##  7:                   458                          2178
##  8:                  1818                         21714
##  9:                   448                          1290
## 10:                  1692                          1628
##     FIRST_DNS_RESPONSE_SUCCESS_D DNS_RESPONSE_SUCCESS_DELAY
##  1:                            2                          2
##  2:                            0                          0
##  3:                            4                          4
##  4:                          272                        272
##  5:                            4                          4
##  6:                            2                          2
##  7:                          366                        366
##  8:                          382                        382
##  9:                            2                          2
## 10:                            2                          2
##     FIRST_TCP_RESPONSE_SUCCESS_D PAGE_SR_DELAYS SYN_SYN_DELAY
##  1:                         2100           2746             4
##  2:                          260            676             8
##  3:                          470           3526             6
##  4:                          558           1282           430
##  5:                          576          13172           462
##  6:                          436          45774            14
##  7:                          458           1274           366
##  8:                         1818          19836          1288
##  9:                          448            470           384
## 10:                         1692           3492            16
##     TCP_CONNECT_DELAY PAGE_BROWSING_DELAYS
##  1:              2100                    0
##  2:               260                    0
##  3:               470                    0
##  4:               558                    0
##  5:               576                    0
##  6:               436                    0
##  7:               458                    0
##  8:              1818                    0
##  9:               448                    0
## 10:              1692                    0
```

```
str(data)
```

```
## Classes 'data.table' and 'data.frame':   30091754 obs. of  16 variables:
##  $ DATETIME                     : chr  "2018-07-06 10:15" "2018-07-06 19:30" "2018-07-06 18:30" "2018-07
-06 00:45" ...
##  $ RAVEN_NAME                   : chr  "Razzle Dazzle Rose raven Shadow" "Tiny Insolent raven Lulu" "Bli
the And Beautiful raven Mo" "Illuminating Emerald raven Daffy" ...
##  $ FAMILY_NAME                  : chr  "Targerian" "Targerian" "Targerian" "Lannister" ...
##  $ MEMBER_NAME                  : chr  "Maester Aemon" "Maester Aemon" "Rheagar" "Kevan" ...
##  $ NETWORK                      : chr  "2g" "4g" "3g" "2g" ...
##  $ FIRST_GET_RESPONSE_SUCCESS_D : int  644 258 1094 432 1094 44856 434 17138 0 1286 ...
##  $ PAGE_BROWSING_DELAY          : int  2832 51752 19562 1488 25132 51494 3018 24412 1760 3834 ...
##  $ TCP_SETUP_TOTAL_DELAY        : int  2100 260 470 558 576 436 458 1818 448 1692 ...
##  $ PAGE_CONTENT_DOWNLOAD_TOTAL_D: int  730 51334 17130 638 13054 50576 2178 21714 1290 1628 ...
##  $ FIRST_DNS_RESPONSE_SUCCESS_D : int  2 0 4 272 4 2 366 382 2 2 ...
##  $ DNS_RESPONSE_SUCCESS_DELAY   : int  2 0 4 272 4 2 366 382 2 2 ...
##  $ FIRST_TCP_RESPONSE_SUCCESS_D : int  2100 260 470 558 576 436 458 1818 448 1692 ...
##  $ PAGE_SR_DELAYS               : int  2746 676 3526 1282 13172 45774 1274 19836 470 3492 ...
##  $ SYN_SYN_DELAY                : int  4 8 6 430 462 14 366 1288 384 16 ...
##  $ TCP_CONNECT_DELAY            : int  2100 260 470 558 576 436 458 1818 448 1692 ...
##  $ PAGE_BROWSING_DELAYS         : int  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
summary(data)
```

```
##    DATETIME          RAVEN_NAME        FAMILY_NAME
##  Length:30091754    Length:30091754    Length:30091754
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  MEMBER_NAME          NETWORK          FIRST_GET_RESPONSE_SUCCESS_D
##  Length:30091754    Length:30091754    Min.   :       0
##  Class :character   Class :character   1st Qu.:      56
##  Mode  :character   Mode  :character   Median :     423
##                                        Mean   :    2607
##                                        3rd Qu.:    1831
##                                        Max.   :28168107
##  PAGE_BROWSING_DELAY TCP_SETUP_TOTAL_DELAY PAGE_CONTENT_DOWNLOAD_TOTAL_D
##  Min.   :       0    Min.   :      0       Min.   :       0
##  1st Qu.:    1723    1st Qu.:    105       1st Qu.:    1220
##  Median :    8211    Median :    345       Median :    6986
##  Mean   :   38310    Mean   :   1580       Mean   :   35227
##  3rd Qu.:   32969    3rd Qu.:   1160       3rd Qu.:   29833
##  Max.   :40286029    Max.   :3469391       Max.   :37853749
##  FIRST_DNS_RESPONSE_SUCCESS_D DNS_RESPONSE_SUCCESS_DELAY
##  Min.   :     0               Min.   :     0.0
##  1st Qu.:     0               1st Qu.:     0.0
##  Median :     2               Median :     1.0
##  Mean   :    56               Mean   :    50.8
##  3rd Qu.:    10               3rd Qu.:     9.0
##  Max.   :376446               Max.   :367035.0
##  FIRST_TCP_RESPONSE_SUCCESS_D PAGE_SR_DELAYS     SYN_SYN_DELAY
##  Min.   :      0              Min.   :       0   Min.   :       0
##  1st Qu.:    105              1st Qu.:     375   1st Qu.:      17
##  Median :    345              Median :    1254   Median :     102
##  Mean   :   1580              Mean   :    5690   Mean   :     580
##  3rd Qu.:   1160              3rd Qu.:    4362   3rd Qu.:     350
##  Max.   :3469391              Max.   :30600387   Max.   :13566475
##  TCP_CONNECT_DELAY  PAGE_BROWSING_DELAYS
##  Min.   :       0   Min.   : 0.000
##  1st Qu.:     134   1st Qu.: 0.000
##  Median :     400   Median : 0.000
##  Mean   :    1816   Mean   : 0.109
##  3rd Qu.:    1316   3rd Qu.: 0.000
##  Max.   :14469128   Max.   :77.000
```

Converting non-numeric columns into factors

```
data$RAVEN_NAME <- as.factor(data$RAVEN_NAME)
data$FAMILY_NAME <- as.factor(data$FAMILY_NAME)
data$MEMBER_NAME <- as.factor(data$MEMBER_NAME)
data$NETWORK <- as.factor(data$NETWORK)
```

```
date <- separate(data,col = DATETIME,into = c('date','time'),sep = 11)
str(date)
```

```
## Classes 'data.table' and 'data.frame':   30091754 obs. of  17 variables:
##  $ date                         : chr  "2018-07-06 " "2018-07-06 " "2018-07-06 " "2018-07-06 " ...
##  $ time                         : chr  "10:15" "19:30" "18:30" "00:45" ...
##  $ RAVEN_NAME                   : Factor w/ 7848 levels "Abnormally White raven Abu",..: 5470 7055 714 3
284 714 714 2609 6139 6139 6139 ...
##  $ FAMILY_NAME                  : Factor w/ 5 levels "Baelish","Greyjoy",..: 5 5 5 3 2 2 1 3 5 2 ...
##  $ MEMBER_NAME                  : Factor w/ 28 levels "Aegon","Aeron",..: 16 16 21 14 24 2 20 8 16 5 ...
##  $ NETWORK                      : Factor w/ 3 levels "2g","3g","4g": 1 3 2 1 3 1 3 1 1 1 ...
##  $ FIRST_GET_RESPONSE_SUCCESS_D : int  644 258 1094 432 1094 44856 434 17138 0 1286 ...
##  $ PAGE_BROWSING_DELAY          : int  2832 51752 19562 1488 25132 51494 3018 24412 1760 3834 ...
##  $ TCP_SETUP_TOTAL_DELAY        : int  2100 260 470 558 576 436 458 1818 448 1692 ...
##  $ PAGE_CONTENT_DOWNLOAD_TOTAL_D: int  730 51334 17130 638 13054 50576 2178 21714 1290 1628 ...
##  $ FIRST_DNS_RESPONSE_SUCCESS_D : int  2 0 4 272 4 2 366 382 2 2 ...
##  $ DNS_RESPONSE_SUCCESS_DELAY   : int  2 0 4 272 4 2 366 382 2 2 ...
##  $ FIRST_TCP_RESPONSE_SUCCESS_D : int  2100 260 470 558 576 436 458 1818 448 1692 ...
##  $ PAGE_SR_DELAYS               : int  2746 676 3526 1282 13172 45774 1274 19836 470 3492 ...
##  $ SYN_SYN_DELAY                : int  4 8 6 430 462 14 366 1288 384 16 ...
##  $ TCP_CONNECT_DELAY            : int  2100 260 470 558 576 436 458 1818 448 1692 ...
##  $ PAGE_BROWSING_DELAYS         : int  0 0 0 0 0 0 0 0 0 0 ...
```

1.Top 10 ravens with fails

```
new_date <- subset(date, FIRST_GET_RESPONSE_SUCCESS_D  != 0 & PAGE_BROWSING_DELAY != 0 & TCP_SETUP_TOTAL_DEL
AY != 0 &
                         PAGE_CONTENT_DOWNLOAD_TOTAL_D != 0 & FIRST_DNS_RESPONSE_SUCCESS_D != 0 & DNS_RESPO
NSE_SUCCESS_DELAY != 0
                         & FIRST_TCP_RESPONSE_SUCCESS_D != 0 & PAGE_SR_DELAYS != 0 & SYN_SYN_DELAY != 0 & T
CP_CONNECT_DELAY !=0 &
                         PAGE_BROWSING_DELAYS != 0)
dim(date)
```

```
## [1] 30091754       17
```

```
dim(new_date)
```

```
## [1] 1577147      17
```

```
str(new_date)
```

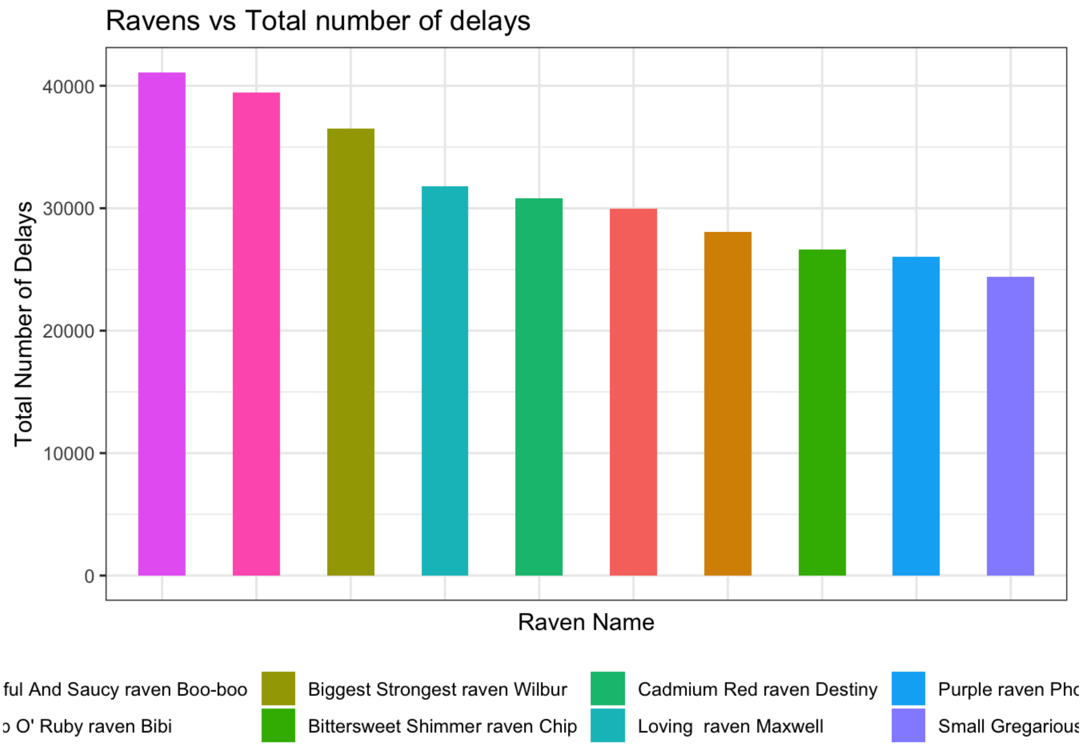
```
## Classes 'data.table' and 'data.frame':   1577147 obs. of  17 variables:
##  $ date                         : chr  "2018-07-06 " "2018-07-06 " "2018-07-06 " "2018-07-06 " ...
##  $ time                         : chr  "21:15" "11:00" "09:15" "11:00" ...
##  $ RAVEN_NAME                   : Factor w/ 7848 levels "Abnormally White raven Abu",..: 7495 1184 5211
5211 5211 5211 6518 1184 1184 1184 ...
##  $ FAMILY_NAME                  : Factor w/ 5 levels "Baelish","Greyjoy",..: 5 4 5 5 2 1 5 4 2 5 ...
##  $ MEMBER_NAME                  : Factor w/ 28 levels "Aegon","Aeron",..: 16 4 1 3 24 20 1 4 2 27 ...
##  $ NETWORK                      : Factor w/ 3 levels "2g","3g","4g": 1 1 3 1 1 3 1 3 2 3 ...
##  $ FIRST_GET_RESPONSE_SUCCESS_D : int  82584 68890 8034 22446 28152 31642 8238 19204 148 16800 ...
##  $ PAGE_BROWSING_DELAY          : int  965374 716276 183722 767086 370708 957970 202612 471890 187586 22
6944 ...
##  $ TCP_SETUP_TOTAL_DELAY        : int  34406 23286 474 51674 8206 6546 6698 5780 3518 7076 ...
##  $ PAGE_CONTENT_DOWNLOAD_TOTAL_D: int  913760 680540 181706 696948 353770 935986 194390 465764 181704 21
9588 ...
##  $ FIRST_DNS_RESPONSE_SUCCESS_D : int  648 54 14 4 16 108 8 22 188 26 ...
##  $ DNS_RESPONSE_SUCCESS_DELAY   : int  646 54 14 4 16 108 8 20 188 26 ...
##  $ FIRST_TCP_RESPONSE_SUCCESS_D : int  34406 23286 474 51674 8206 6546 6698 5780 3518 7076 ...
##  $ PAGE_SR_DELAYS               : int  134198 104626 10050 92584 45090 53626 16460 25330 6030 24156 ...
##  $ SYN_SYN_DELAY                : int  12894 18636 64 1312 3816 3612 580 5316 132 5998 ...
##  $ TCP_CONNECT_DELAY            : int  34806 23286 474 52492 19180 8360 6698 5900 3518 7076 ...
##  $ PAGE_BROWSING_DELAYS         : int  2 2 2 8 2 6 2 4 2 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
raven <- new_date %>%
  group_by(RAVEN_NAME) %>%
  summarise(TotalDelays = n()) %>%
  arrange(-TotalDelays) %>%
  head(10)

raven
```

```
## # A tibble: 10 x 2
##    RAVEN_NAME                        TotalDelays
##    <fct>                                   <int>
##  1 Sole Musical raven Azul                 41114
##  2 Withered raven Mo                       39447
##  3 Biggest Strongest raven Wilbur          36498
##  4 Loving  raven Maxwell                   31787
##  5 Cadmium Red raven Destiny               30820
##  6 Beautiful And Saucy raven Boo-boo       29989
##  7 Big Dip O' Ruby raven Bibi              28054
##  8 Bittersweet Shimmer raven Chip          26605
##  9 Purple raven Phoenix                    26024
## 10 Small Gregarious raven Paco             24380
```

```
ggplot(raven,aes(x = reorder(RAVEN_NAME,-TotalDelays), y = TotalDelays)) +
  geom_bar(stat = "Identity", aes(fill = RAVEN_NAME), width = 0.5) +
  theme_bw() +
  theme(axis.text.x = element_blank(),axis.ticks.x = element_blank(),legend.position = "bottom") +
  labs(title = "Ravens vs Total number of delays",
       x = "Raven Name" , y = "Total Number of Delays")
```



2.Top 10 ravens without fail

```
without_fail <- subset(date, FIRST_GET_RESPONSE_SUCCESS_D  == 0 & PAGE_BROWSING_DELAY == 0 & TCP_SETUP_TOTAL
_DELAY == 0 &
                             PAGE_CONTENT_DOWNLOAD_TOTAL_D == 0 & FIRST_DNS_RESPONSE_SUCCESS_D == 0 & DNS_R
ESPONSE_SUCCESS_DELAY  == 0                            & FIRST_TCP_RESPONSE_SUCCESS_D == 0 & PAGE_SR_DELAY
S == 0 & SYN_SYN_DELAY == 0 & TCP_CONNECT_DELAY == 0
                             & PAGE_BROWSING_DELAYS == 0)
dim(without_fail)
```

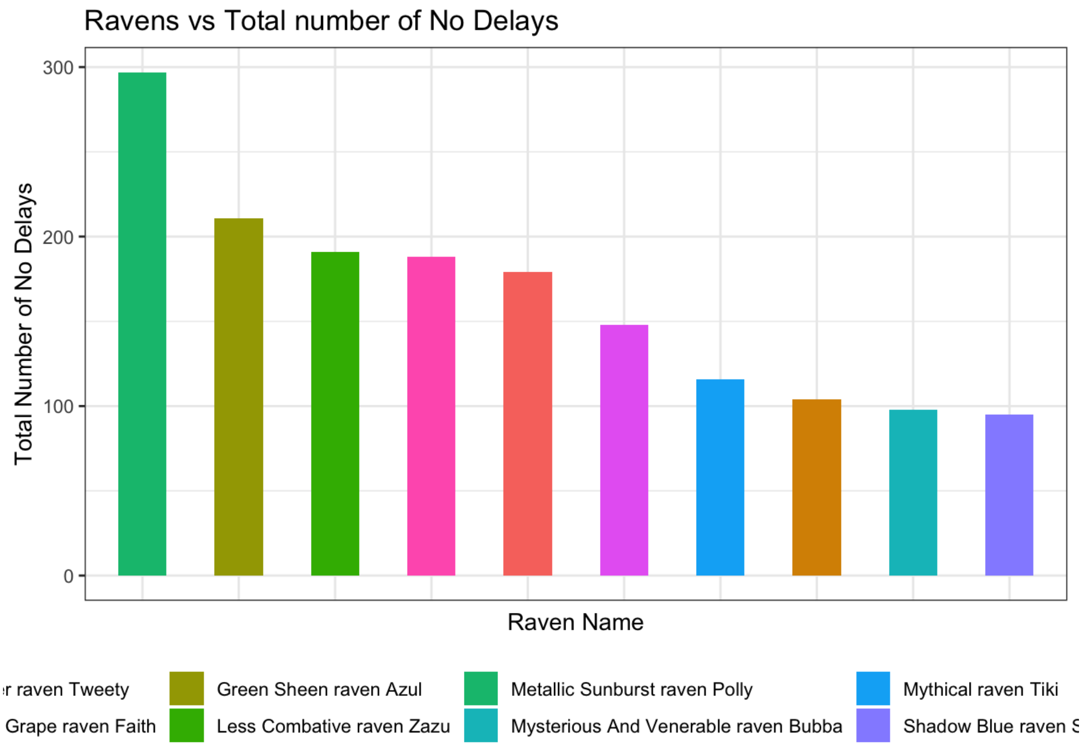
```
## [1] 10811    17
```

```
raven_withoutfail <- without_fail %>%
  group_by(RAVEN_NAME) %>%
  summarise(TotalNoDelays = n()) %>%
  arrange(-TotalNoDelays) %>%
  head(10)

raven_withoutfail
```

```
## # A tibble: 10 x 2
##   RAVEN_NAME                            TotalNoDelays
##   <fct>                                         <int>
##  1 Metallic Sunburst raven Polly                  297
##  2 Green Sheen raven Azul                         211
##  3 Less Combative raven Zazu                      191
##  4 Weak  raven Buddy                              188
##  5 Copper raven Tweety                            179
##  6 Spectral Yellow raven Zazu                     148
##  7 Mythical raven Tiki                            116
##  8 Cyber Grape raven Faith                        104
##  9 Mysterious And Venerable raven Bubba            98
## 10 Shadow Blue raven Sammy                         95
```

```
ggplot(raven_withoutfail,aes(x = reorder(RAVEN_NAME,-TotalNoDelays), y = TotalNoDelays)) +
  geom_bar(stat = "Identity", aes(fill = RAVEN_NAME), width = 0.5) +
  theme_bw() +
  theme(axis.text.x = element_blank(),axis.ticks.x = element_blank(),legend.position = "bottom")+
  labs(title = "Ravens vs Total number of No Delays",
       x = "Raven Name" , y = "Total Number of No Delays")
```



3. Family with most fails

```
family_mfails <- new_date %>%
  group_by(FAMILY_NAME) %>%
  summarise(MostDelays = n()) %>%
  arrange(-MostDelays) %>%
  head(1)
family_mfails
```

```
## # A tibble: 1 x 2
##   FAMILY_NAME MostDelays
##   <fct>            <int>
## 1 Targerian       487966
```

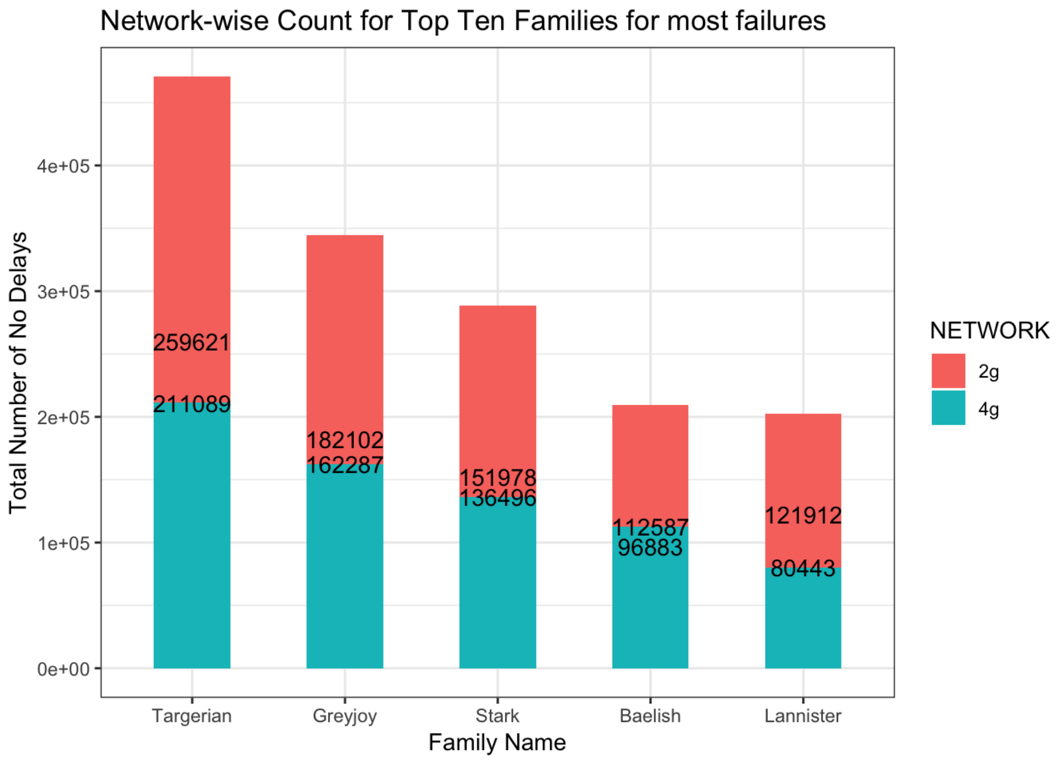
Family and network-wise fails

```
family_and_network_mfails <- new_date %>%
  group_by(FAMILY_NAME,NETWORK) %>%
  summarise(MostDelays = n()) %>%
  arrange(-MostDelays) %>%
  head(10)

family_and_network_mfails
```

```
## # A tibble: 10 x 3
## # Groups:   FAMILY_NAME [5]
##    FAMILY_NAME NETWORK MostDelays
##    <fct>       <fct>        <int>
##  1 Targerian   2g          259621
##  2 Targerian   4g          211089
##  3 Greyjoy     2g          182102
##  4 Greyjoy     4g          162287
##  5 Stark       2g          151978
##  6 Stark       4g          136496
##  7 Lannister   2g          121912
##  8 Baelish     4g          112587
##  9 Baelish     2g           96883
## 10 Lannister   4g           80443
```

```
ggplot(family_and_network_mfails,aes(x = reorder(FAMILY_NAME,-MostDelays), y =MostDelays)) +
  geom_bar(stat = "Identity", aes(fill = NETWORK), width = 0.5) +
  theme_bw() +
  geom_text(data = family_and_network_mfails, aes(x = FAMILY_NAME, y = MostDelays, label = MostDelays), vjus
t = 0.5) +
  labs(title = "Network-wise Count for Top Ten Families for most failures",
       x = "Family Name" , y = "Total Number of No Delays")
```



4. Family with least fails

```
family_lfails <- new_date %>%
  group_by(FAMILY_NAME) %>%
  summarise(LeastDelays = n()) %>%
  arrange(-LeastDelays) %>%
  tail(1)

family_lfails
```

```
## # A tibble: 1 x 2
##   FAMILY_NAME LeastDelays
##   <fct>             <int>
## 1 Lannister        212823
```

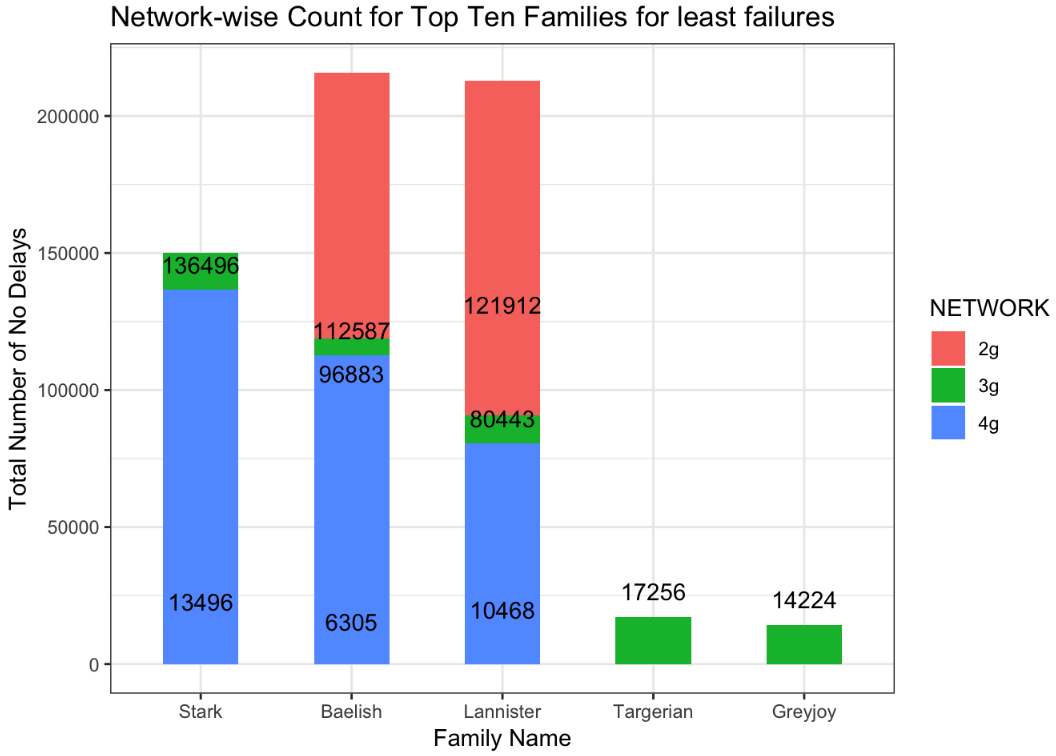
Family and Network-wise least fails

```
family_and_network_lfails <- new_date %>%
  group_by(FAMILY_NAME,NETWORK) %>%
  summarise(LeastDelays = n()) %>%
  arrange(-LeastDelays) %>%
  tail(10)

family_and_network_lfails
```

```
## # A tibble: 10 x 3
## # Groups:   FAMILY_NAME [5]
##    FAMILY_NAME NETWORK LeastDelays
##    <fct>       <fct>         <int>
##  1 Stark       4g           136496
##  2 Lannister   2g           121912
##  3 Baelish     4g           112587
##  4 Baelish     2g            96883
##  5 Lannister   4g            80443
##  6 Targerian   3g            17256
##  7 Greyjoy     3g            14224
##  8 Stark       3g            13496
##  9 Lannister   3g            10468
## 10 Baelish     3g             6305
```

```
ggplot(family_and_network_lfails,aes(x = reorder(FAMILY_NAME,-LeastDelays), y =LeastDelays)) +
  geom_bar(stat = "Identity", aes(fill = NETWORK), width = 0.5) +
  theme_bw() +
  geom_text(data = family_and_network_lfails, aes(x = FAMILY_NAME, y = LeastDelays, label = LeastDelays), vj
ust = -1.00) +
  labs(title = "Network-wise Count for Top Ten Families for least failures",
       x = "Family Name" , y = "Total Number of No Delays")
```



5. Family member with most fail

```
member_most <- new_date %>%
  group_by(MEMBER_NAME) %>%
  summarise(MostDelays = n()) %>%
  arrange(-MostDelays) %>%
  head(10)

member_most
```

```
## # A tibble: 10 x 2
##    MEMBER_NAME    MostDelays
##    <fct>               <int>
##  1 Petyr Baelish      215775
##  2 Deanerys           192597
##  3 Theon              146974
##  4 Maester Aemon      136599
##  5 Eddard              78912
##  6 Maester Kerwin      72942
##  7 Aeron               65503
##  8 Robb                59121
##  9 Rheagar             52103
## 10 Viserys             50084
```

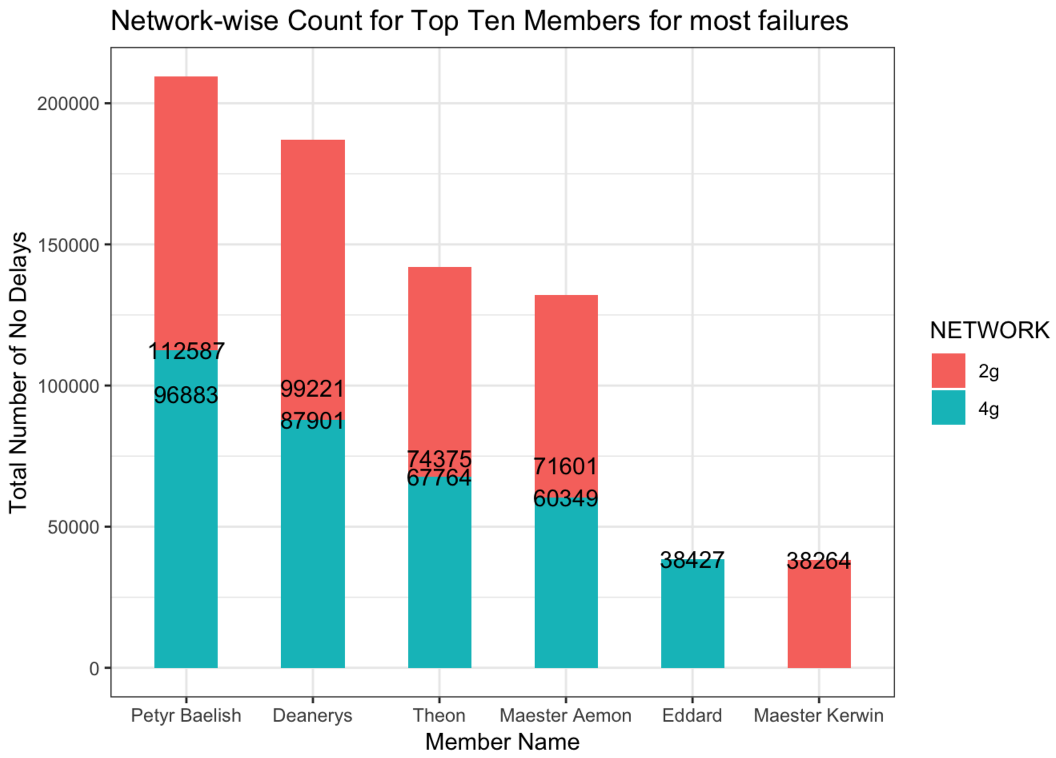
Family member and Network-wise most fails

```
member_and_network_most <- new_date %>%
  group_by(MEMBER_NAME,NETWORK) %>%
  summarise(MostDelays = n()) %>%
  arrange(-MostDelays) %>%
  head(10)

member_and_network_most
```

```
## # A tibble: 10 x 3
## # Groups:   MEMBER_NAME [6]
##    MEMBER_NAME    NETWORK MostDelays
##    <fct>          <fct>        <int>
##  1 Petyr Baelish  4g          112587
##  2 Deanerys       2g           99221
##  3 Petyr Baelish  2g           96883
##  4 Deanerys       4g           87901
##  5 Theon          2g           74375
##  6 Maester Aemon  2g           71601
##  7 Theon          4g           67764
##  8 Maester Aemon  4g           60349
##  9 Eddard         4g           38427
## 10 Maester Kerwin 2g           38264
```

```
ggplot(member_and_network_most,aes(x = reorder(MEMBER_NAME,-MostDelays), y =MostDelays)) +
  geom_bar(stat = "Identity", aes(fill = NETWORK), width = 0.5) +
  theme_bw() +
  geom_text(data = member_and_network_most, aes(x = MEMBER_NAME, y = MostDelays, label = MostDelays), vjust
= 0.5) +
  labs(title = "Network-wise Count for Top Ten Members for most failures",
       x = "Member Name" , y = "Total Number of No Delays")
```



6. Family member with least fail

```
member_least <- new_date %>%
  group_by(MEMBER_NAME) %>%
  summarise(LeastDelays = n()) %>%
  arrange(-LeastDelays) %>%
  tail(10)

member_least
```

```
## # A tibble: 10 x 2
##    MEMBER_NAME LeastDelays
##    <fct>             <int>
##  1 Jamie             28804
##  2 Aegon             27414
##  3 Cersei            27323
##  4 Yara/Asha         24405
##  5 Tywin             24331
##  6 Benjen            20657
##  7 Lancel            19265
##  8 Joanna            18292
##  9 Sansa             16912
## 10 Euron             15823
```

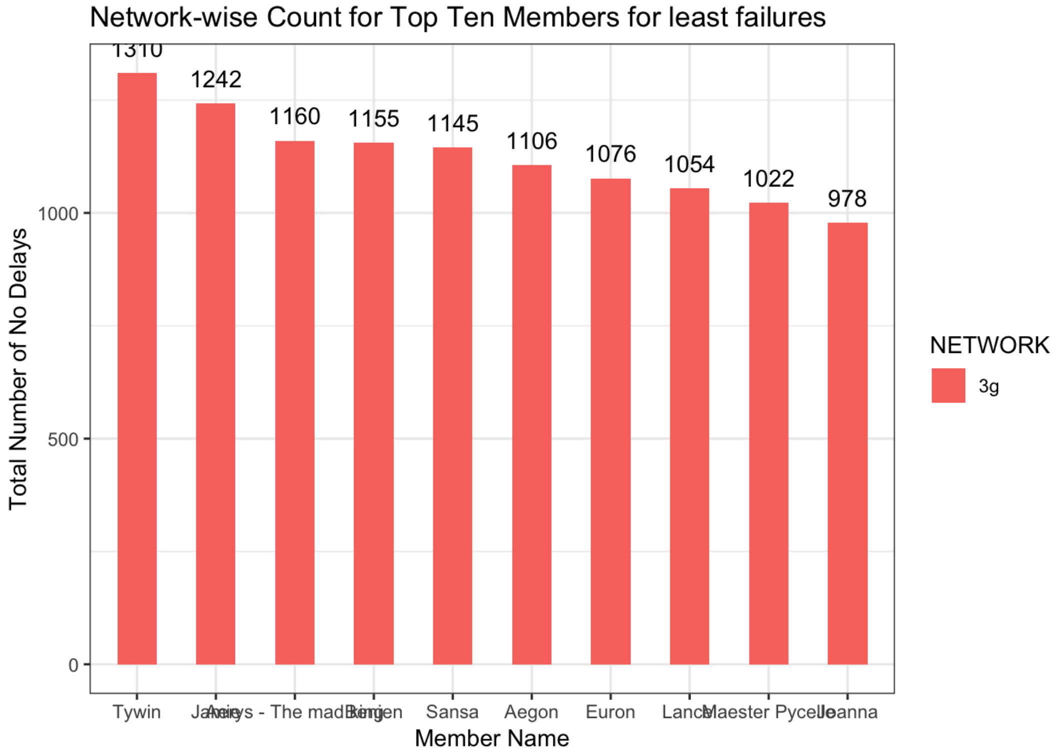
Family member and Network-wise least fails

```
member_and_network_least <- new_date %>%
  group_by(MEMBER_NAME,NETWORK) %>%
  summarise(LeastDelays = n()) %>%
  arrange(-LeastDelays) %>%
  tail(10)

member_and_network_least
```

```
## # A tibble: 10 x 3
## # Groups:   MEMBER_NAME [10]
##    MEMBER_NAME          NETWORK LeastDelays
##    <fct>                <fct>         <int>
##  1 Tywin                3g             1310
##  2 Jamie                3g             1242
##  3 Aerys - The mad king 3g             1160
##  4 Benjen               3g             1155
##  5 Sansa                3g             1145
##  6 Aegon                3g             1106
##  7 Euron                3g             1076
##  8 Lancel               3g             1054
##  9 Maester Pycelle      3g             1022
## 10 Joanna               3g              978
```

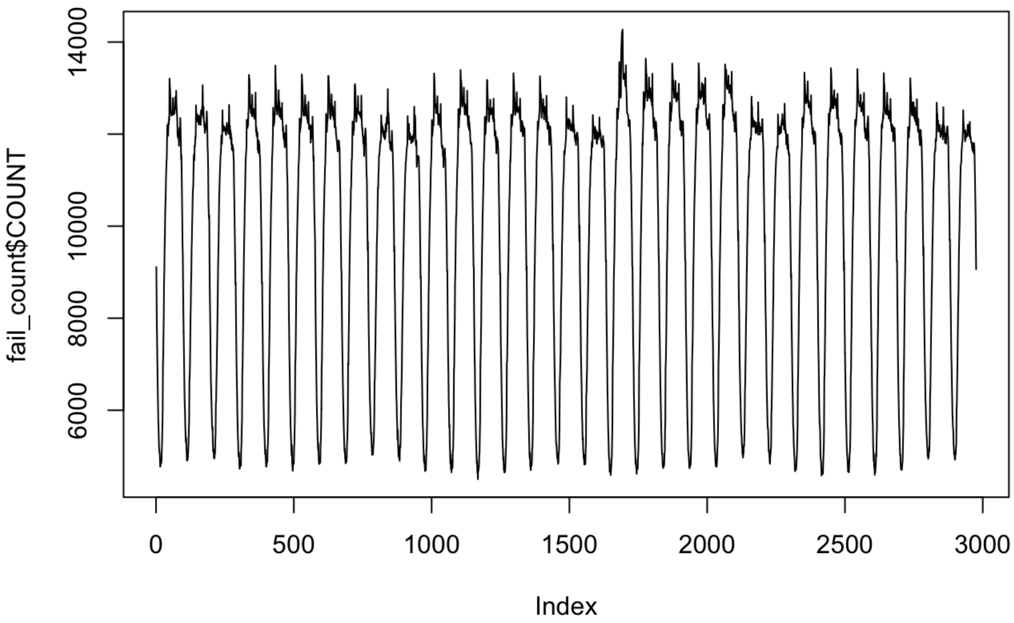
```
ggplot(member_and_network_least,aes(x = reorder(MEMBER_NAME,-LeastDelays), y =LeastDelays)) +
  geom_bar(stat = "Identity", aes(fill = NETWORK), width = 0.5) +
  theme_bw() +
  geom_text(data = member_and_network_least, aes(x = MEMBER_NAME, y = LeastDelays, label = LeastDelays), vju
st = -1.00) +
  labs(title = "Network-wise Count for Top Ten Members for least failures",
       x = "Member Name" , y = "Total Number of No Delays")
```



Model forcasting

```
fail_count <- data %>%
  group_by(DATETIME) %>%
  summarise(COUNT = n()) %>%
  arrange(DATETIME)

fail_count <- as.data.frame(fail_count)
plot(fail_count$COUNT,type = "l")
```



```
adf.test(fail_count$COUNT)
```

```
## Warning in adf.test(fail_count$COUNT): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  fail_count$COUNT
## Dickey-Fuller = -11.101, Lag order = 14, p-value = 0.01
## alternative hypothesis: stationary
```

```
arima(fail_count$COUNT, c(0,0,2))
```

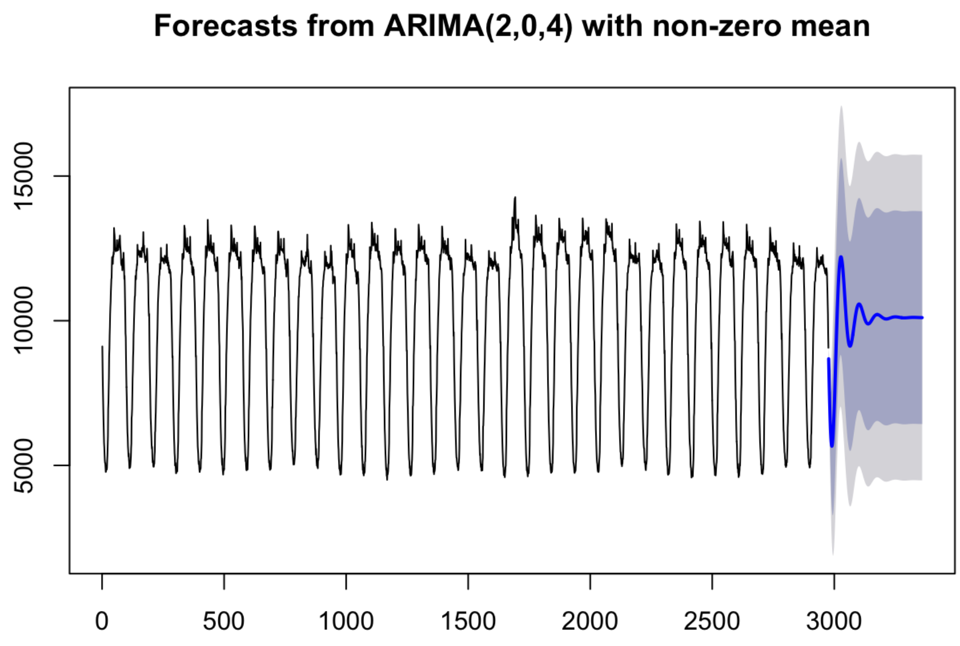
```
##
## Call:
## arima(x = fail_count$COUNT, order = c(0, 0, 2))
##
## Coefficients:
##          ma1     ma2   intercept
##       1.4993  0.9198  10106.8198
## s.e.  0.0069  0.0063     56.5433
##
## sigma^2 estimated as 814428:  log likelihood = -24477.14,  aic = 48962.28
```

```
arima(fail_count$COUNT, c(0,1,2))
```

```
##
## Call:
## arima(x = fail_count$COUNT, order = c(0, 1, 2))
##
## Coefficients:
##          ma1     ma2
##       0.3322  0.2877
## s.e.  0.0188  0.0145
##
## sigma^2 estimated as 71186:  log likelihood = -20841.37,  aic = 41688.74
```

aic value is minimum for model c(0,1,2) Data is given for 15 minutes interval, so h = (4 * 24 * 4)

```
model_arima <- auto.arima(fail_count$COUNT)
plot(forecast(model_arima,h=384))
```



Model is giving better prediction for 1 day.

```
length(unique(data$RAVEN_NAME))
```

```
## [1] 7848
```

There are 7848 unique ravens, we can build arima model for each of them so that

failure information can be forcasted for each of them.