Analysis of Montreal breakins data 2015-16

Quan Nguyen

May 18, 2016

Get data

Data source:

http://donnees.ville.montreal.qc.ca/dataset/actes-criminels

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

```
## Warning: package 'dplyr' was built under R version 3.2.4
```

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

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

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

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

```
library(dygraphs)
```

```
## Warning: package 'dygraphs' was built under R version 3.2.5
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
DT = fread(input = "donneesouvertes-citoyens.csv", header = T, sep=",", stringsAsFactors = T)
DT
```

```
##           CATEGORIE     DATE QUART PDQ        X       Y      LAT      LONG
##     1: Introduction 20150101  nuit   8 289215.1 5036423 45.46756 -73.69931
##     2: Introduction 20150101  nuit  48 302729.3 5050946 45.59841 -73.52654
##     3: Introduction 20150101  nuit  38 298080.3 5042832 45.52538 -73.58602
##     4: Introduction 20150101  jour  23 302375.2 5046522 45.55861 -73.53106
##     5: Introduction 20150101  jour  27 291594.6 5045993 45.55372 -73.66913
```

```
##    ---
## 10854: Introduction 20160331  jour  21      0.0       0  1.00000   1.00000
## 10855: Introduction 20160331  soir  26 295797.5 5040826 45.50729 -73.61521
## 10856: Introduction 20160331  soir  38 297165.9 5042638 45.52362 -73.59773
## 10857: Introduction 20160331  soir  16 299214.3 5035799 45.46210 -73.57143
## 10858: Introduction 20151118  jour  26 295080.9 5041034 45.50916 -73.62438
```

```
DT %>% summary()
```

```
##         CATEGORIE          DATE           QUART           PDQ
##  Introduction:10858   Min.   :20150101   jour:4542   Min.   : 3.00
##                       1st Qu.:20150428   nuit:1978   1st Qu.:20.00
##                       Median :20150823   soir:4338   Median :27.00
##                       Mean   :20152420               Mean   :28.31
##                       3rd Qu.:20151204               3rd Qu.:38.00
##                       Max.   :20160331               Max.   :50.00
##        X                Y               LAT             LONG
##  Min.   :     0   Min.   :      0   Min.   : 1.00   Min.   :-73.94
##  1st Qu.:294670   1st Qu.:5040089   1st Qu.:45.50   1st Qu.:-73.63
##  Median :297511   Median :5043516   Median :45.53   Median :-73.59
##  Mean   :289988   Mean   :4926634   Mean   :44.50   Mean   :-71.87
##  3rd Qu.:299544   3rd Qu.:5046490   3rd Qu.:45.56   3rd Qu.:-73.56
##  Max.   :306256   Max.   :5062126   Max.   :45.70   Max.   :  1.00
```

```
str(DT)
```

```
## Classes 'data.table' and 'data.frame':   10858 obs. of  8 variables:
##  $ CATEGORIE: Factor w/ 1 level "Introduction": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DATE     : int  20150101 20150101 20150101 20150101 20150101 20150101 20150101 20150101 20150101 1
##  $ QUART    : Factor w/ 3 levels "jour","nuit",..: 2 2 2 1 1 1 1 1 1 1 ...
##  $ PDQ      : int  8 48 38 23 27 23 49 38 16 42 ...
##  $ X        : num  289215 302729 298080 302375 291595 ...
##  $ Y        : num  5036423 5050946 5042832 5046522 5045993 ...
##  $ LAT      : num  45.5 45.6 45.5 45.6 45.6 ...
##  $ LONG     : num  -73.7 -73.5 -73.6 -73.5 -73.7 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Plot over time

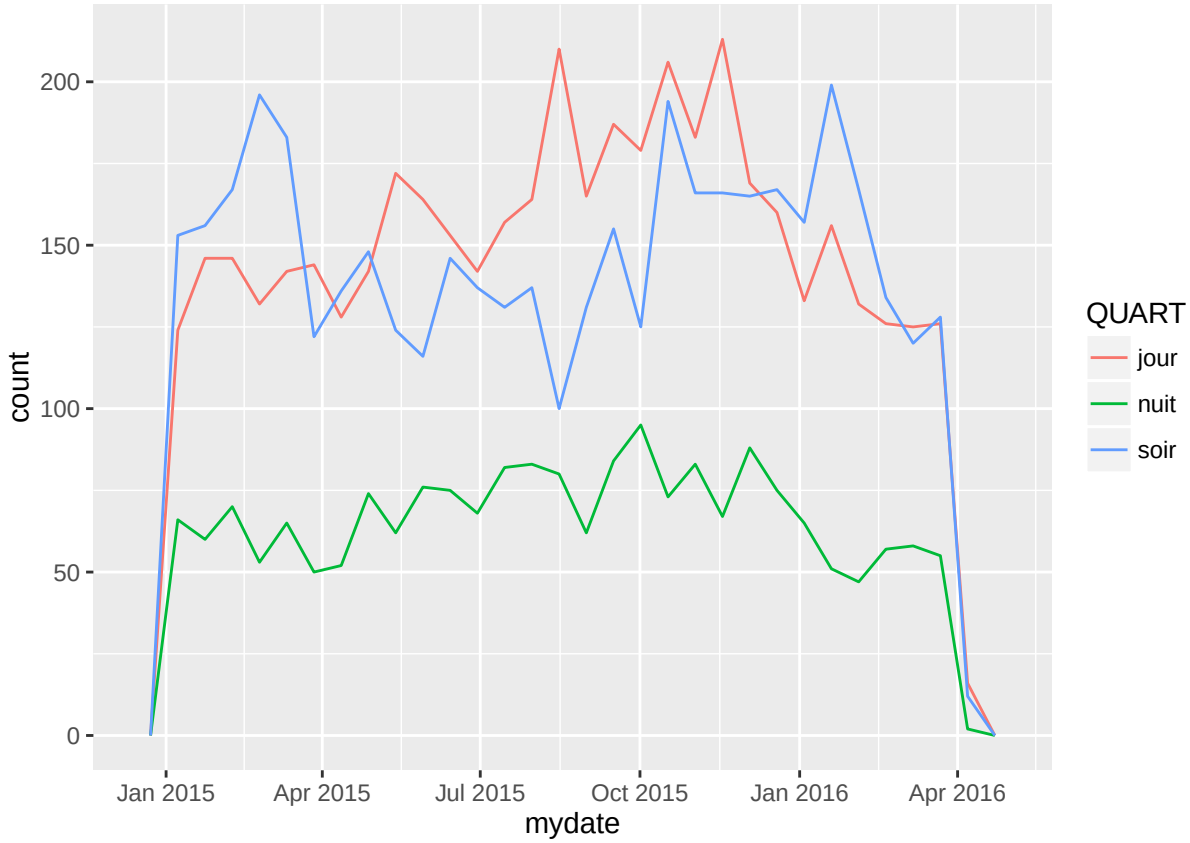
```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

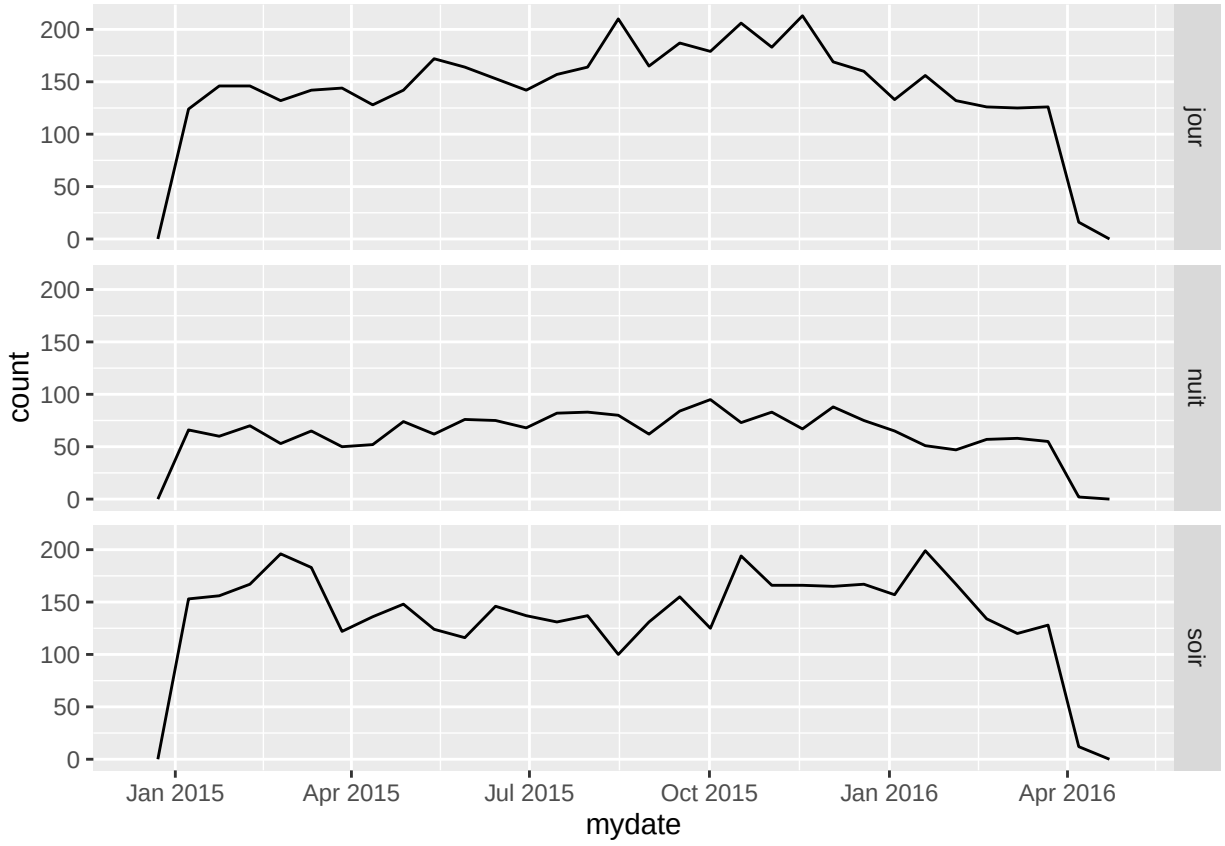
```
# Add a few date attributes for timeseries processing
DT$mydate = as.Date(as.character(DT$DATE), tz="EST", format="%Y%m%d")
DT$year = as.Date(cut(DT$mydate, breaks="year"))
DT$month = as.Date(cut(DT$mydate, breaks="month"))
DT$week = as.Date(cut(DT$mydate, breaks="week"))
DT$weekday = as.POSIXlt(DT$mydate)$wday

# plot of daily breakins split by period (day, evening, night)
ggplot(DT, aes(x=mydate, color=QUART)) + geom_freqpoly()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

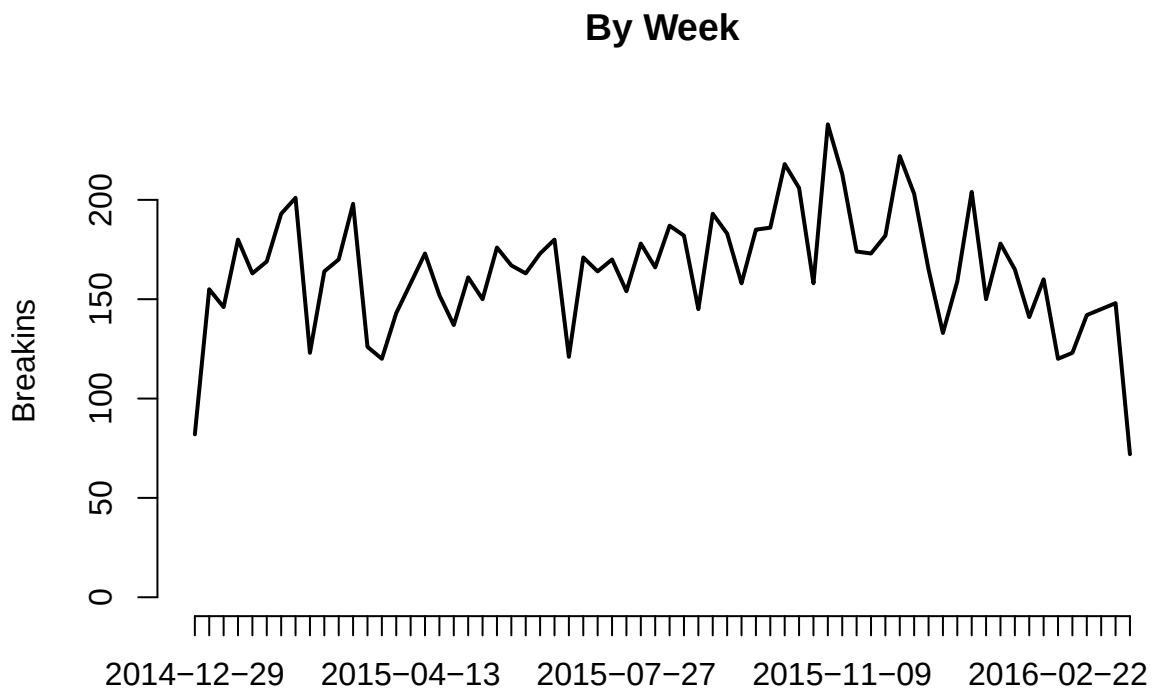



```
ggplot(DT, aes(x=mydate)) + facet_grid(QUART ~ .) + geom_freqpoly()
```

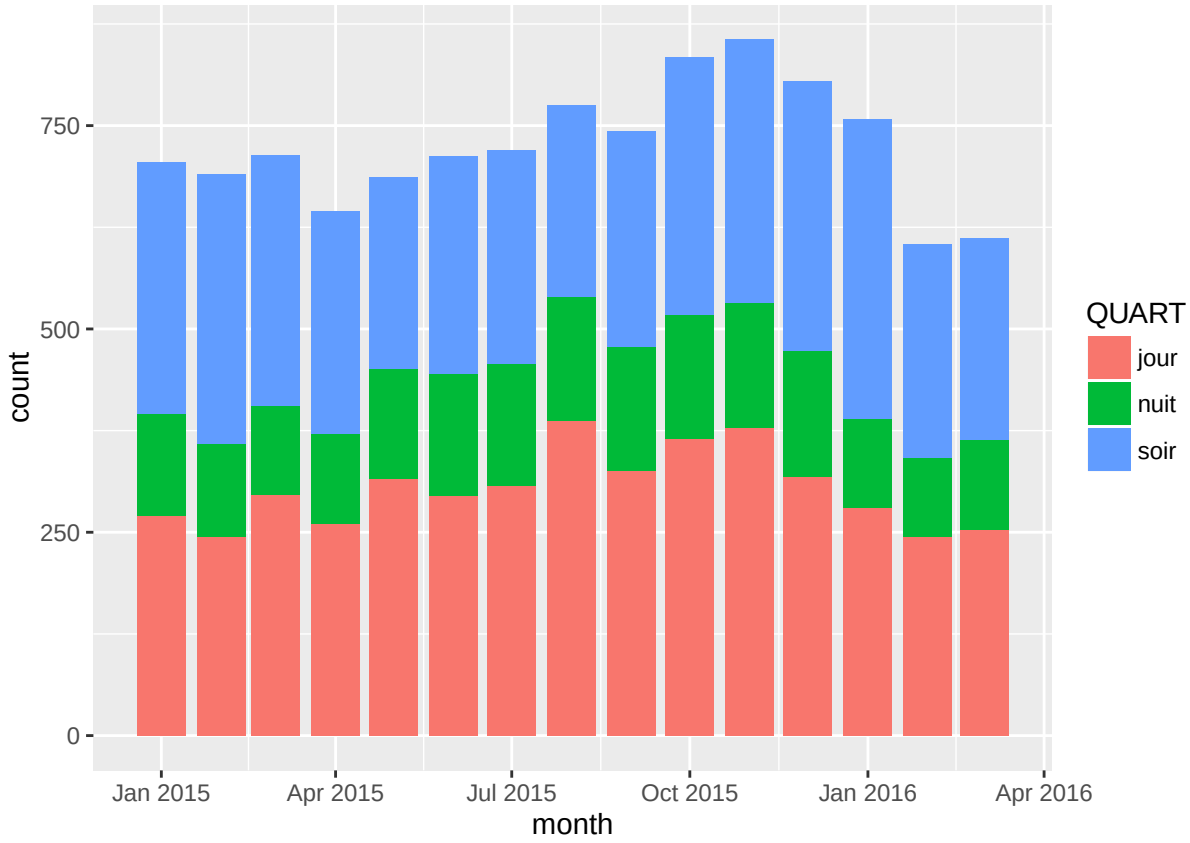
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# plot number of breakins per week
plot(table(DT$week), main="By Week", ylab="Breakins", type='l')
```



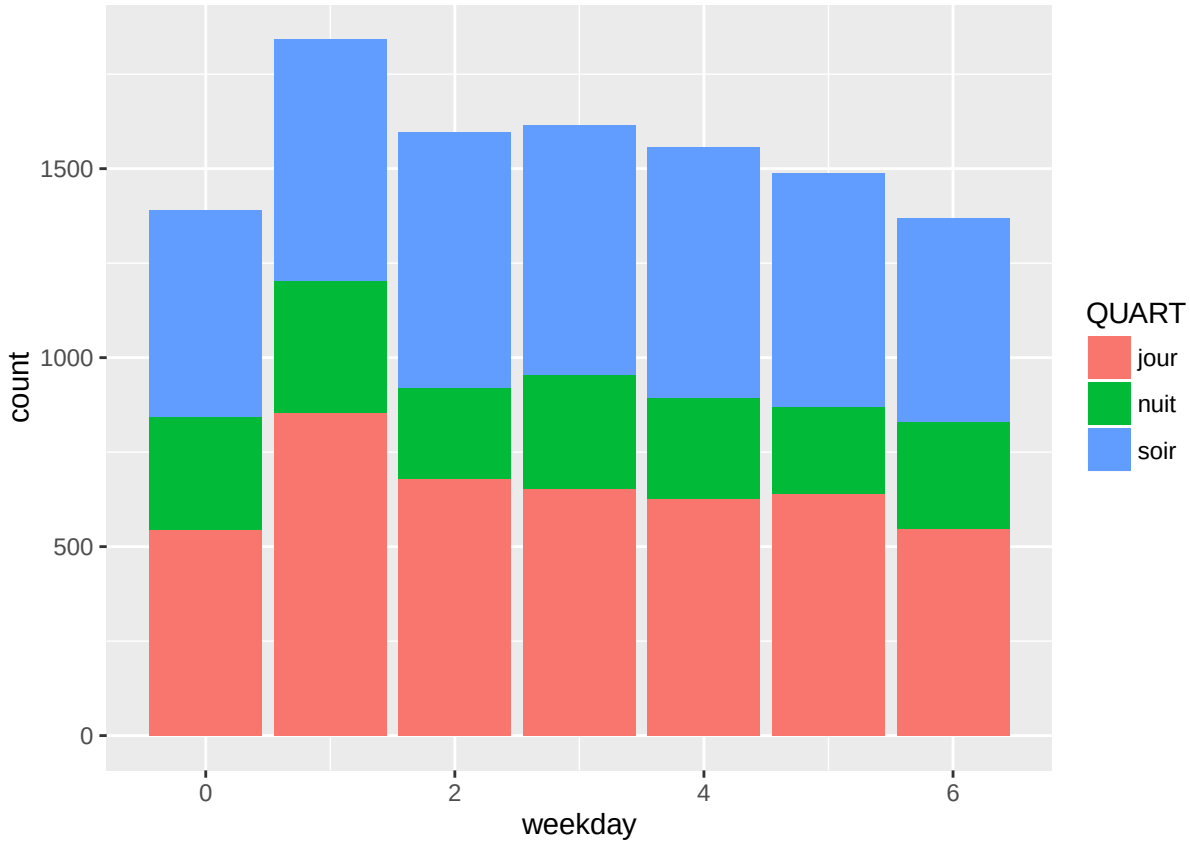

```
# Plot nunber of breakins per month split by period
ggplot(DT, aes(x=month, fill=QUART)) + geom_bar()
```




```
# By weekday (0: Sunday)
table(DT$weekday);
```

```
##
##    0    1    2    3    4    5    6
## 1390 1842 1597 1614 1557 1488 1370
```

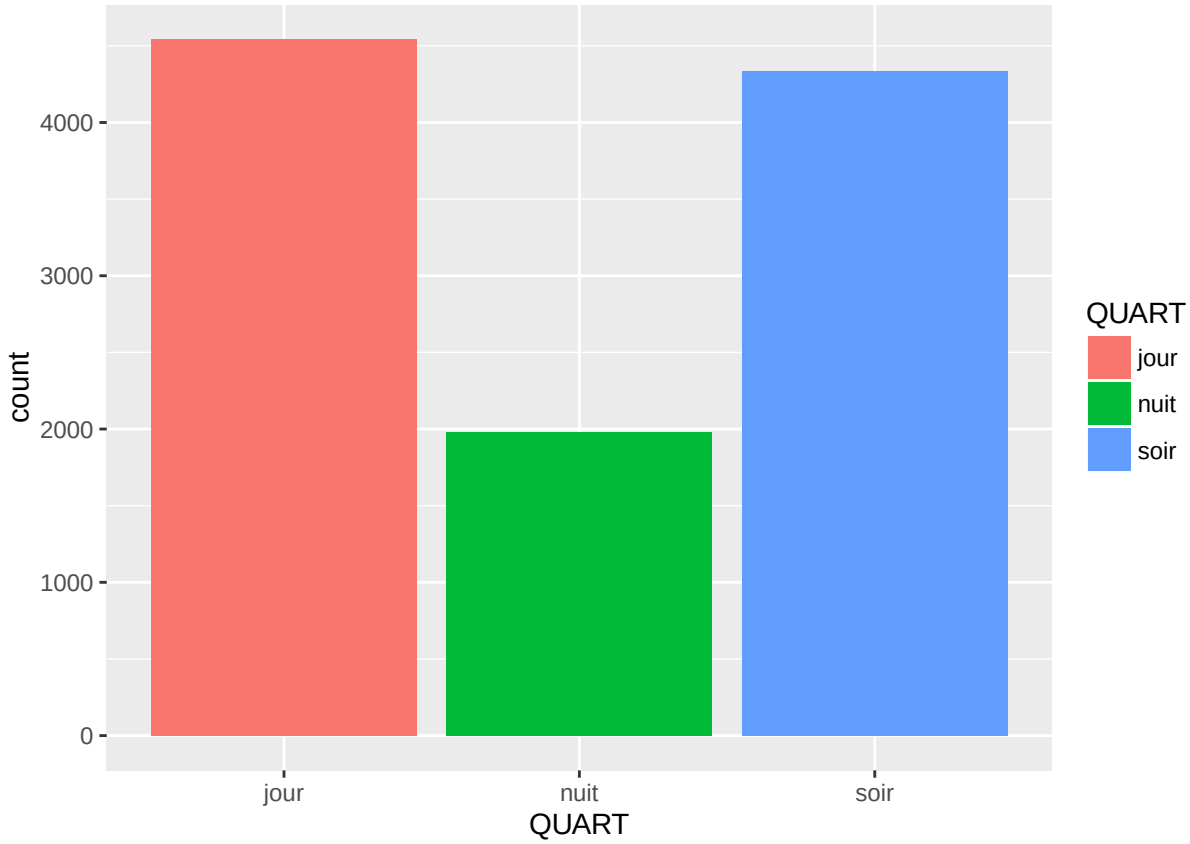
```
ggplot(DT, aes(x=weekday, fill=QUART)) + geom_bar()
```




```
# By Quart
table(DT$QUART);
```

```
##
## jour nuit soir
## 4542 1978 4338
```

```
ggplot(DT, aes(x=QUART, fill=QUART)) + geom_bar()
```

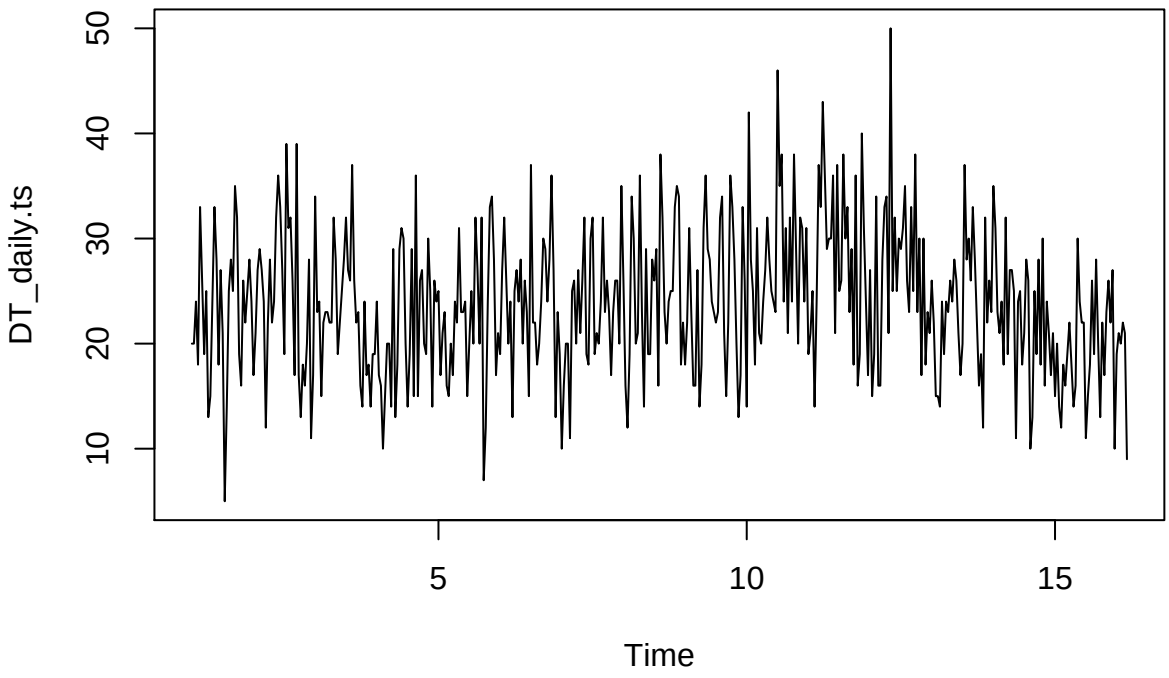


Create daily aggregate

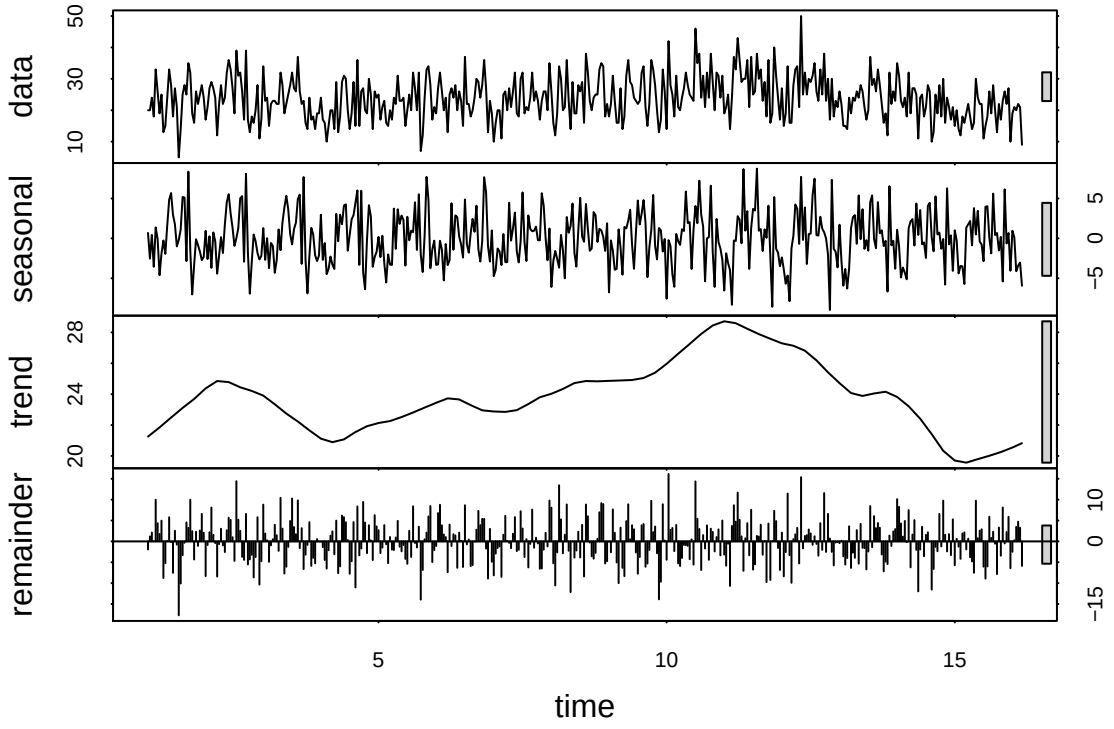
```
DT_daily = count(DT, mydate)
```

Convert to ts() series

```
# frequency=365 will give proper x-axis tic values
# however will not be accepted by stl() because of
# series is not periodic or has less than two periods
DT_daily.ts = ts(
  DT_daily$n,
  frequency=30 #, start=c(2015,0.1)
)
plot(DT_daily.ts)
```




```
# timeseries decomposition
DT_daily.stl = stl(DT_daily.ts, s.window=7)
plot(DT_daily.stl)
```




```
# Convert to zoo series

DT_daily.z = zoo(
  DT_daily$n,
  order.by=DT_daily$mydate,
  frequency=7
)
```

Arima forecast

```
library(forecast)
```

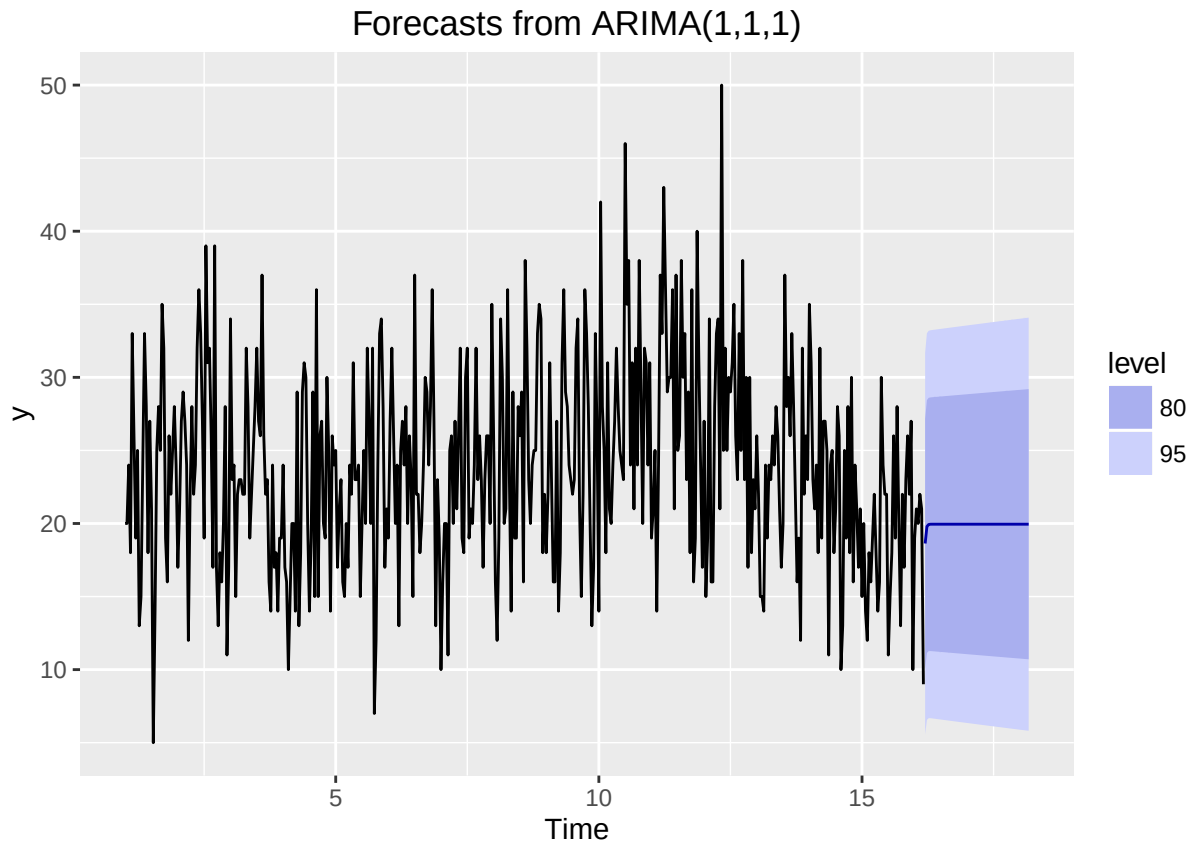
```
## Warning: package 'forecast' was built under R version 3.2.5
```

```
## Loading required package: timeDate
```

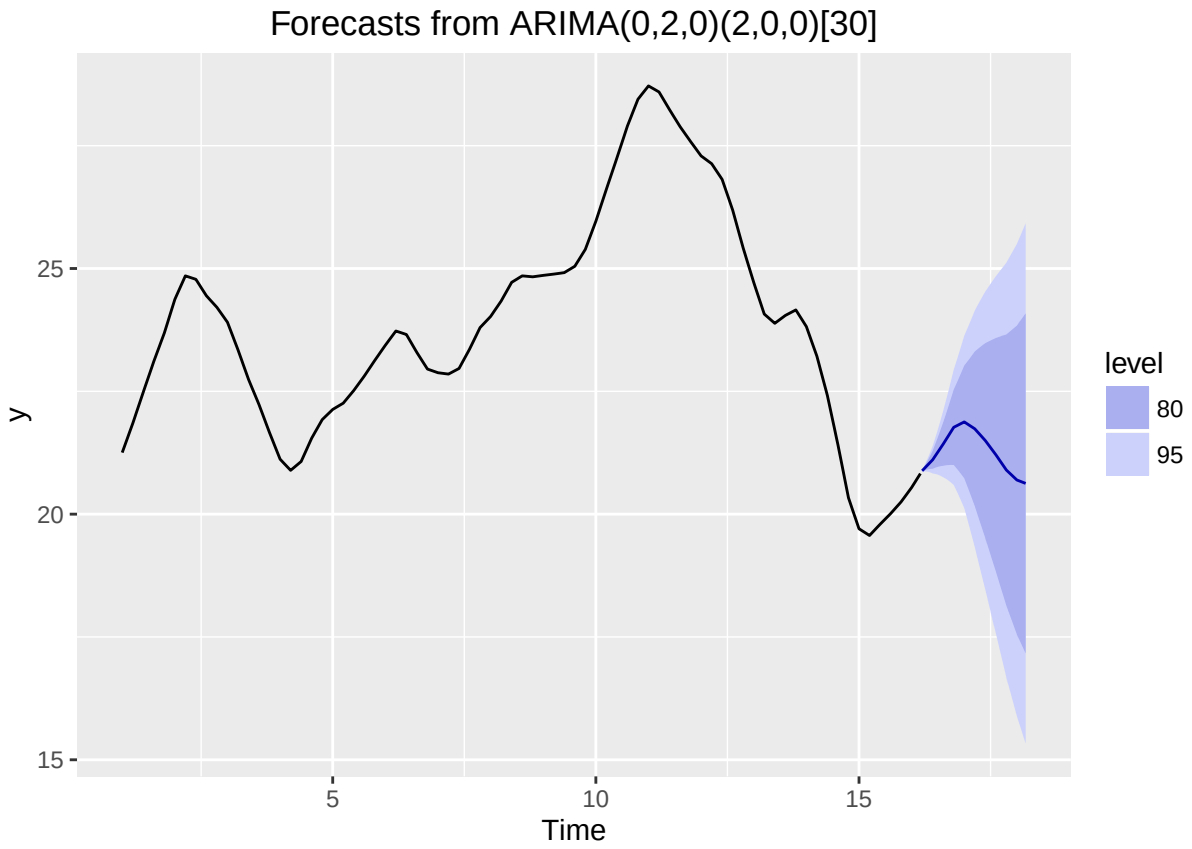
```
## This is forecast 7.1
```

```
library(ggplot2)
DT_daily.z.forecast =
  forecast.Arima(
```

```
    auto.arima(
      DT_daily.ts
    ),
    h=60
  )
autoplot(DT_daily.z.forecast)
```




```
# prediction of the trendline

DT_daily.ts.forecast =
  forecast.Arima(
    auto.arima(
      DT_daily.stl$time.series[,2]
    ),
    h=60
  )
autoplot(DT_daily.ts.forecast)
```



Anomaly detection

```
library(AnomalyDetection)
myts = as.data.frame(
  cbind(
    as.POSIXct(index(DT_daily.z)),
    coredata(DT_daily.z)
  )
)
data_anomaly = AnomalyDetectionTs(myts, max_anoms=0.01, direction="pos", plot=F, e_value = T, na.rm = T
```

```
# No anomaly detected as NULL result returned
data_anomaly
```

```
## $anoms
## data frame with 0 columns and 0 rows
##
## $plot
## NULL
```

```
data_anomaly$plot
```

```
## NULL
```

EDA

```
# Breakins by PDQ (Police de quartier) sorted
DT %>% select(PDQ) %>% table %>% sort(decreasing=T)
```

```
## .
##   38   26   23   44   39   48   27   35   15    7   31   16   42    8   22
## 1099  733  689  588  555  551  542  531  480  429  395  389  372  365  364
##   30   21   20   11   13   45   10   33   46   49    3   12   24   50
##  321  312  286  245  215  208  205  199  192  192  168  123  107    3
```

```
DT %>% group_by(PDQ) %>% summarise(n = n()) %>% mutate(freq = n / sum(n)) %>% arrange(desc(freq)) %>% s
```

```
## Source: local data table [29 x 2]
##
##      PDQ       freq
##    (int)      (dbl)
## 1     38 0.10121569
## 2     26 0.06750783
## 3     23 0.06345552
## 4     44 0.05415362
## 5     39 0.05111439
## 6     48 0.05074599
## 7     27 0.04991711
## 8     35 0.04890403
## 9     15 0.04420704
## 10     7 0.03951004
## ..   ...        ...
```

```
DT %>% select(PDQ, QUART) %>% table
```

```
##     QUART
## PDQ  jour nuit soir
##   3    66   28   74
##   7   188   90  151
##   8   183   59  123
##   10   80   37   88
##   11  112   46   87
##   12   47   19   57
##   13   99   45   71
##   15  211   95  174
##   16  141   60  188
##   20  143   59   84
##   21  150   67   95
##   22  162   62  140
##   23  260  115  314
##   24   42   18   47
##   26  259  101  373
##   27  221   76  245
##   30  138   73  110
##   31  146   66  183
##   33   85   39   75
##   35  226   87  218
##   38  429  205  465
##   39  256  124  175
##   42  161   85  126
##   44  232  112  244
##   45   78   43   87
##   46   88   36   68
##   48  248   89  214
##   49   91   40   61
##   50    0    2    1
```

```r
# columns without/with zero values
colSums(DT == 0)
```

```
## CATEGORIE      DATE     QUART       PDQ         X         Y       LAT
##         0         0         0         0       252       252         0
##      LONG    mydate      year     month      week   weekday
##         0         0         0         0         0      1390
```

```r
colSums(DT != 0)
```

```
## CATEGORIE      DATE     QUART       PDQ         X         Y       LAT
##     10858     10858     10858     10858     10606     10606     10858
##      LONG    mydate      year     month      week   weekday
##     10858     10858     10858     10858     10858      9468
```

```r
DT_goodXY = DT %>% filter(X != 0); DT_goodXY
```

```
##           CATEGORIE     DATE QUART PDQ        X       Y      LAT      LONG
##     1: Introduction 20150101  nuit   8 289215.1 5036423 45.46756 -73.69931
##     2: Introduction 20150101  nuit  48 302729.3 5050946 45.59841 -73.52654
##     3: Introduction 20150101  nuit  38 298080.3 5042832 45.52538 -73.58602
##     4: Introduction 20150101  jour  23 302375.2 5046522 45.55861 -73.53106
##     5: Introduction 20150101  jour  27 291594.6 5045993 45.55372 -73.66913
##    ---
## 10602: Introduction 20160331  jour  48 301601.0 5049969 45.58962 -73.54100
## 10603: Introduction 20160331  soir  26 295797.5 5040826 45.50729 -73.61521
## 10604: Introduction 20160331  soir  38 297165.9 5042638 45.52362 -73.59773
## 10605: Introduction 20160331  soir  16 299214.3 5035799 45.46210 -73.57143
## 10606: Introduction 20151118  jour  26 295080.9 5041034 45.50916 -73.62438
##            mydate       year      month       week weekday
##     1: 2015-01-01 2015-01-01 2015-01-01 2014-12-29       4
##     2: 2015-01-01 2015-01-01 2015-01-01 2014-12-29       4
##     3: 2015-01-01 2015-01-01 2015-01-01 2014-12-29       4
##     4: 2015-01-01 2015-01-01 2015-01-01 2014-12-29       4
##     5: 2015-01-01 2015-01-01 2015-01-01 2014-12-29       4
##    ---
## 10602: 2016-03-31 2016-01-01 2016-03-01 2016-03-28       4
## 10603: 2016-03-31 2016-01-01 2016-03-01 2016-03-28       4
## 10604: 2016-03-31 2016-01-01 2016-03-01 2016-03-28       4
## 10605: 2016-03-31 2016-01-01 2016-03-01 2016-03-28       4
## 10606: 2015-11-18 2015-01-01 2015-11-01 2015-11-16       3
```

```r
library(ggplot2)
# ggplot(DT_goodXY, aes(y=LAT, x=LONG, col=DT_goodXY$QUART)) + geom_point()
```

Put it on the map

```
library(maptools)
```

```
## Warning: package 'maptools' was built under R version 3.2.4
```

```
## Loading required package: sp
```

```
## Warning: package 'sp' was built under R version 3.2.5
```

```
## Checking rgeos availability: TRUE
```

```
# needs all the files shipped in addition to .shp files
mtl_admin_shp = readShapeSpatial("data/LIMADMIN.shp")
mtl_admin_poly = readShapePoly("data/LIMADMIN")

mtl_admin_points = fortify(mtl_admin_shp)
```

```
## Regions defined for each Polygons
```

```
centroids.df <- as.data.frame(coordinates(mtl_admin_poly))
names(centroids.df) <- c("long", "lat")
centroids.df$id = as.character(mtl_admin_poly$NOM)

# in ggplot over ggmap
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.2.4
```

```
montreal12 <- get_map(location = "montreal", zoom=12)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=montreal&zoom=12&size=640x640&sca
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=montreal&sensor=false
```

```
gc3 = geocode("Pont Viau, Quebec", source="google")
```

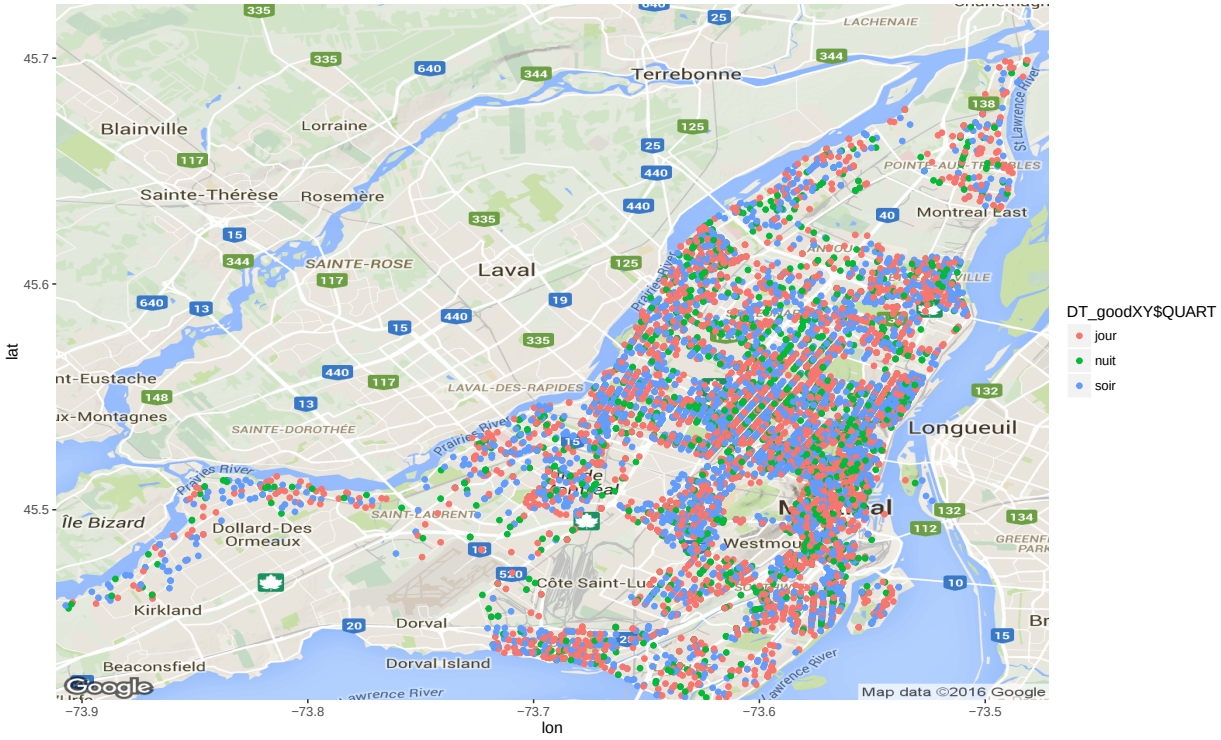
```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Pont%20Viau,%20Quebec
```

```
center <- as.numeric(gc3)
montreal12 <- get_googlemap(location = "montreal", zoom=11, center=center)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=45.570336,-73.692033&zoom=11&size
```

```
ggmap(montreal12) +
  coord_equal() +
  geom_point(data=DT_goodXY, aes(y=LAT, x=LONG, col=DT_goodXY$QUART)
  )
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```



Conclusion

- Breakins occur motly in daytime or evening, much less during the night
- Breaskins are highest in Oct-Dec
- Breaksins are highest on Monday
- The data is just for 15 months so it's not enough to get a feel for yearly trend
- Police de quartier stations 38, 26, 23 see the most number of breakins (10%, 6.7% and 6.3% respectively)
- Note that the map only shows areas that are part of the city of Montreal only. There doesn't seem any part that has higher infractions