sample.R

grapefroot

Sun Oct 18 14:39:22 2015

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

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

```
library(forecast)
```

```
## Loading required package: timeDate
## This is forecast 6.1
```

```
library(ggplot2)
library(gridExtra)

test = fread("./test.csv")
train = fread("./train.csv")
store = fread("./store.csv")

## Take a look on the data
str(train)
```

```
## Classes 'data.table' and 'data.frame':   1017209 obs. of  9 variables:
##  $ Store        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ DayOfWeek    : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Date         : chr  "2015-07-31" "2015-07-31" "2015-07-31" "2015-07-31" ...
##  $ Sales        : int  5263 6064 8314 13995 4822 5651 15344 8492 8565 7185 ...
##  $ Customers    : int  555 625 821 1498 559 589 1414 833 687 681 ...
##  $ Open         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Promo        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ StateHoliday : chr  "0" "0" "0" "0" ...
##  $ SchoolHoliday: chr  "1" "1" "1" "1" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(test)
```

```
## Classes 'data.table' and 'data.frame':   41088 obs. of  8 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Store        : int  1 3 7 8 9 10 11 12 13 14 ...
##  $ DayOfWeek    : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Date         : chr  "2015-09-17" "2015-09-17" "2015-09-17" "2015-09-17" ...
##  $ Open         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Promo        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ StateHoliday : chr  "0" "0" "0" "0" ...
##  $ SchoolHoliday: chr  "0" "0" "0" "0" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(store)
```

```
## Classes 'data.table' and 'data.frame':   1115 obs. of  10 variables:
##  $ Store                    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ StoreType                : chr  "c" "a" "a" "c" ...
##  $ Assortment               : chr  "a" "a" "a" "c" ...
##  $ CompetitionDistance      : int  1270 570 14130 620 29910 310 24000 7520 2030 3160 ...
##  $ CompetitionOpenSinceMonth: int  9 11 12 9 4 12 4 10 8 9 ...
##  $ CompetitionOpenSinceYear : int  2008 2007 2006 2009 2015 2013 2013 2014 2000 2009 ...
##  $ Promo2                   : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ Promo2SinceWeek          : int  NA 13 14 NA NA NA NA NA NA NA ...
##  $ Promo2SinceYear          : int  NA 2010 2011 NA NA NA NA NA NA NA ...
##  $ PromoInterval            : chr  "" "Jan,Apr,Jul,Oct" "Jan,Apr,Jul,Oct" "" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
##transform data
train[, Date := as.Date(Date)]
test[, Date := as.Date(Date)]

#order by date
train = train[order(Date)]
test = test[order(Date)]

test[is.na(test)] = 1

train[, lapply(.SD, function(x) length(unique(x)))]
```

```
##    Store DayOfWeek Date Sales Customers Open Promo StateHoliday
## 1:  1115         7  942 21734      4086    2     2            4
##    SchoolHoliday
## 1:             2
```

```
test[, lapply(.SD, function(x) length(unique(x)))]
```

```
##       Id Store DayOfWeek Date Open Promo StateHoliday SchoolHoliday
## 1: 41088   856         7   48    2     2            2             2
```

```
#All test stores are in the train
sum(unique(test$Store) %in% unique(train$Store))
```

```
## [1] 856
```

```
#259 from train are not into the train
sum(!(unique(train$Store) %in% unique(test$Store)))
```

```
## [1] 259
```

```
#percentage of open stores in train and test
table(train$Open)/nrow(train)
```

```
##
##         0         1
## 0.1698933 0.8301067
```

```
table(test$Open)/nrow(test)
```

```
##
##         0         1
## 0.1456386 0.8543614
```

```
#percentage of promotions in train and test
table(train$Promo)/nrow(train)
```

```
##
##         0         1
## 0.6184855 0.3815145
```

```
table(test$Promo)/nrow(test)
```

```
##
##         0         1
## 0.6041667 0.3958333
```

```
#percentage of school holidays in train and test.
table(train$SchoolHoliday)/nrow(train)
```

```
##
##         0         1
## 0.8213533 0.1786467
```

```
table(test$SchoolHoliday)/nrow(test)
```

```
##
##         0         1
## 0.5565129 0.4434871
```

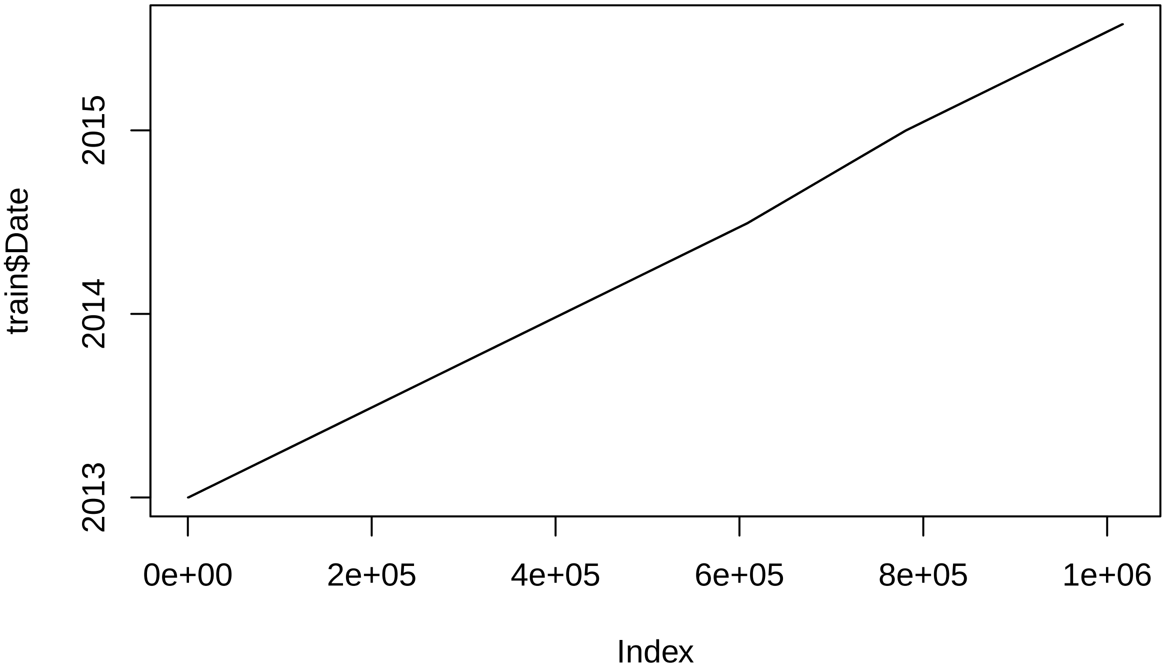
```
#major difference observed

#percentage of state holidays in train and test.
table(train$StateHoliday)/nrow(train)
```

```
##
##           0           a           b           c
## 0.969475300 0.019917244 0.006576820 0.004030637
```

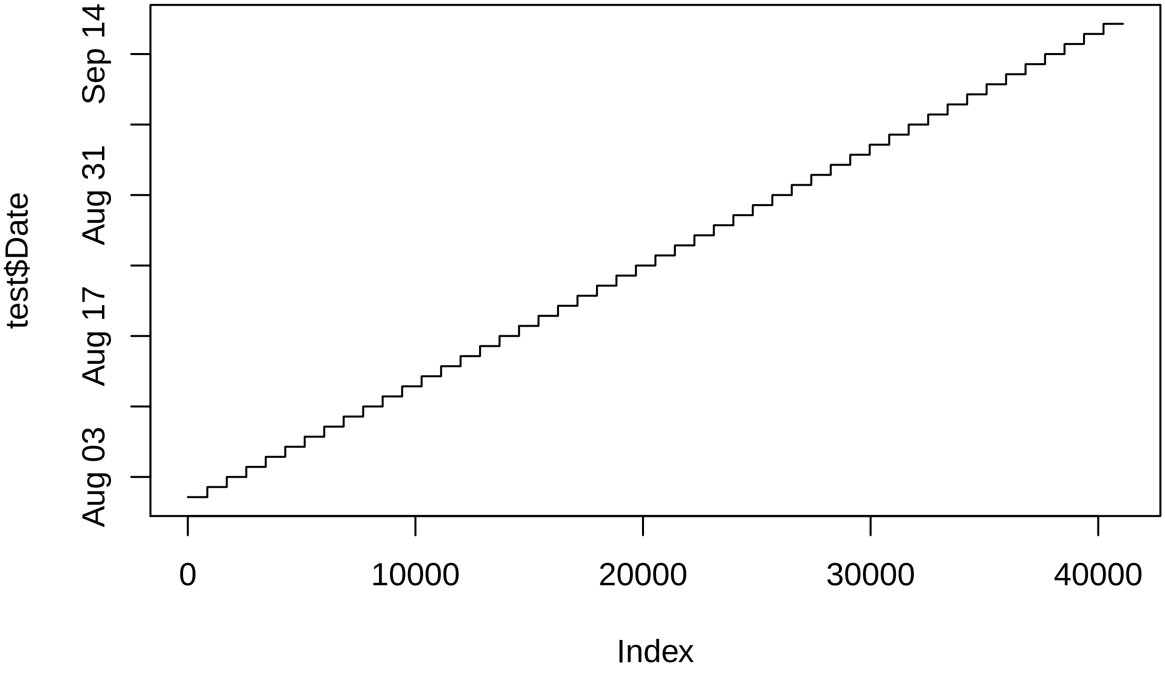
```
table(test$StateHoliday)/nrow(test)
```

```
##
##           0           a
## 0.995619159 0.004380841
```

```
plot(train$Date, type="l")
```



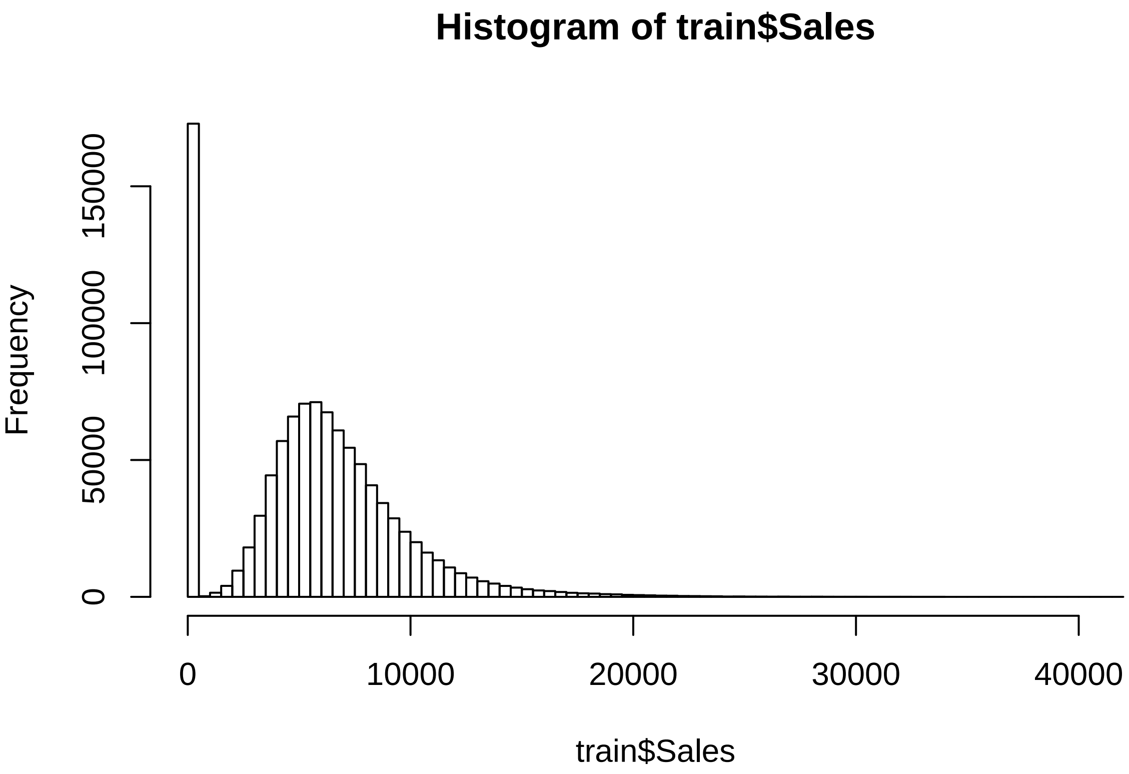
```
plot(test$Date, type="l")
```



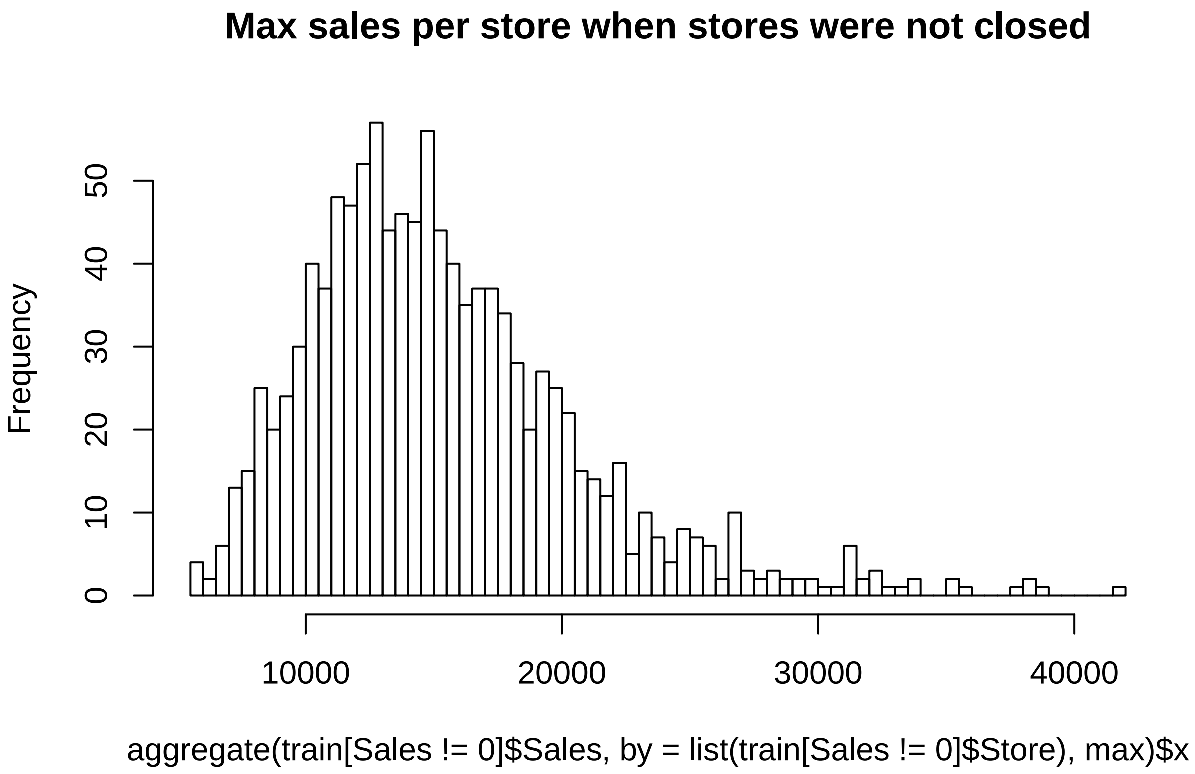
```
all(table(test$Date) == 856)
```

```
## [1] TRUE
```

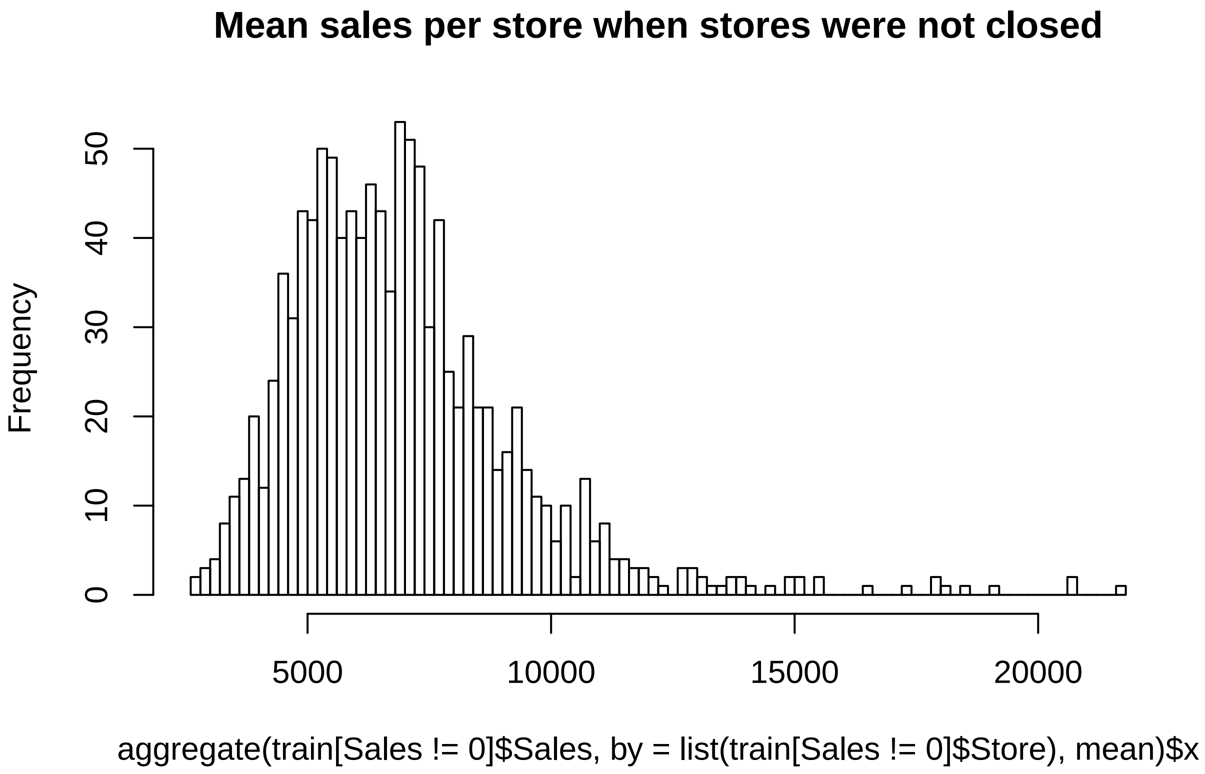
```
hist(train$Sales, 100)
```



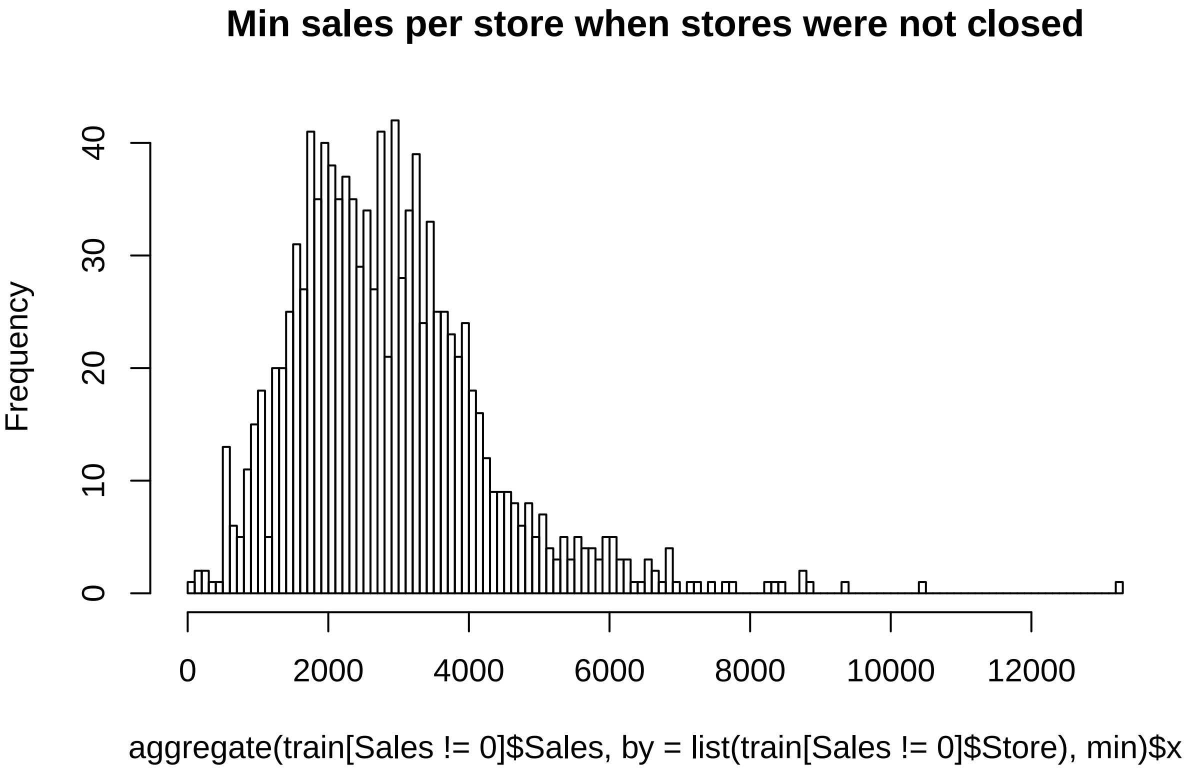
```
maxhist = hist(aggregate(train[Sales != 0]$Sales,
          by = list(train[Sales != 0]$Store), max)$x, 100,
     main = "Max sales per store when stores were not closed")
```



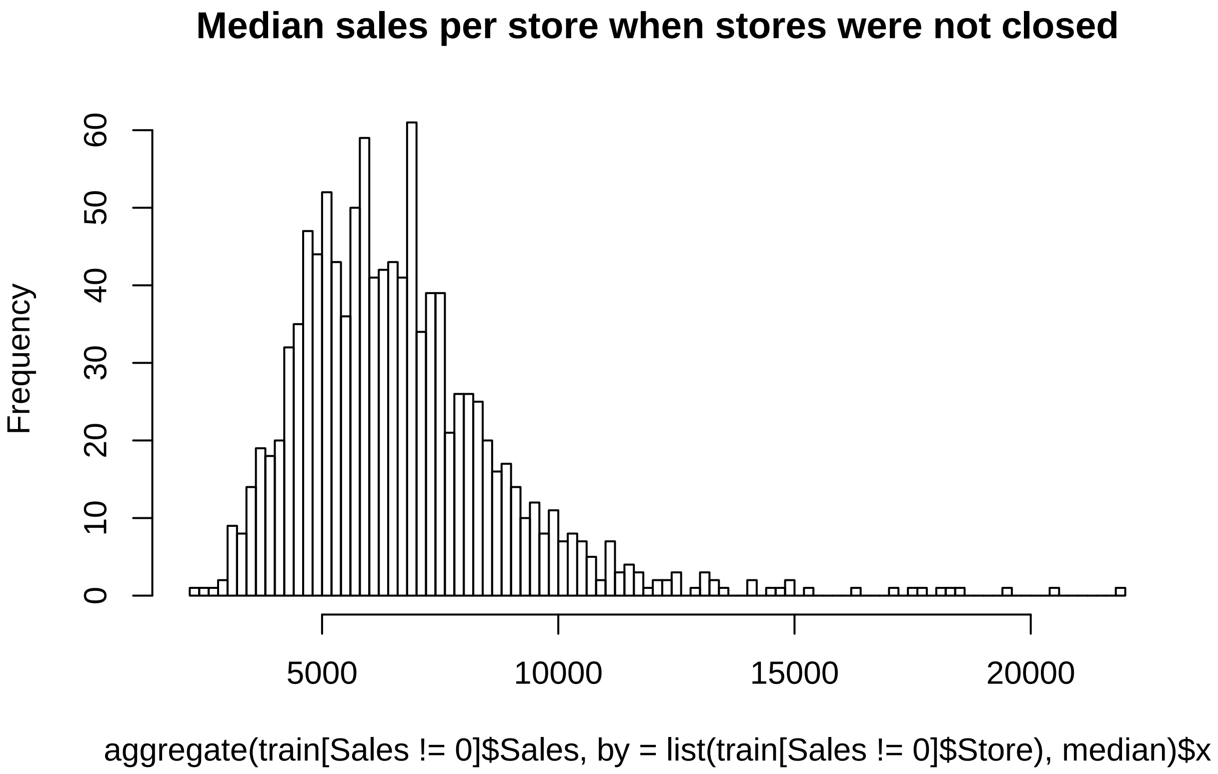
```
meanhist = hist(aggregate(train[Sales != 0]$Sales,
                          by = list(train[Sales != 0]$Store), mean)$x, 100,
     main = "Mean sales per store when stores were not closed")
```



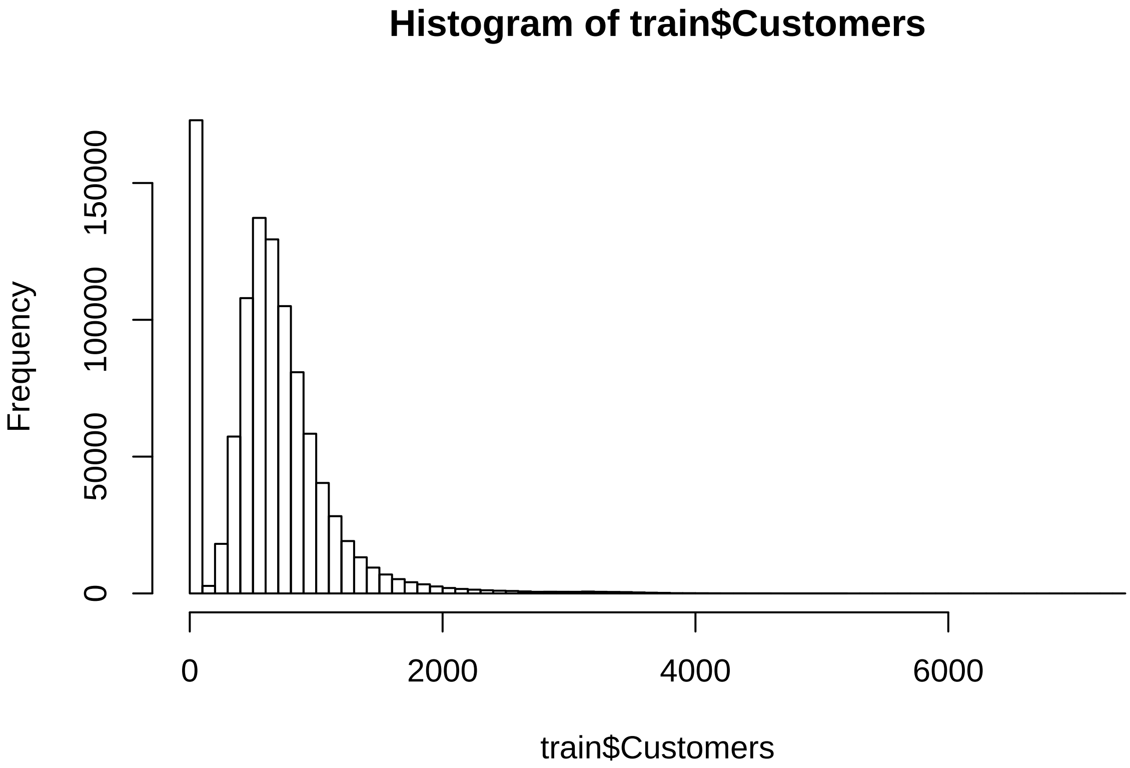
```
minhist = hist(aggregate(train[Sales != 0]$Sales,
                by = list(train[Sales != 0]$Store), min)$x, 100,
     main = "Min sales per store when stores were not closed")
```



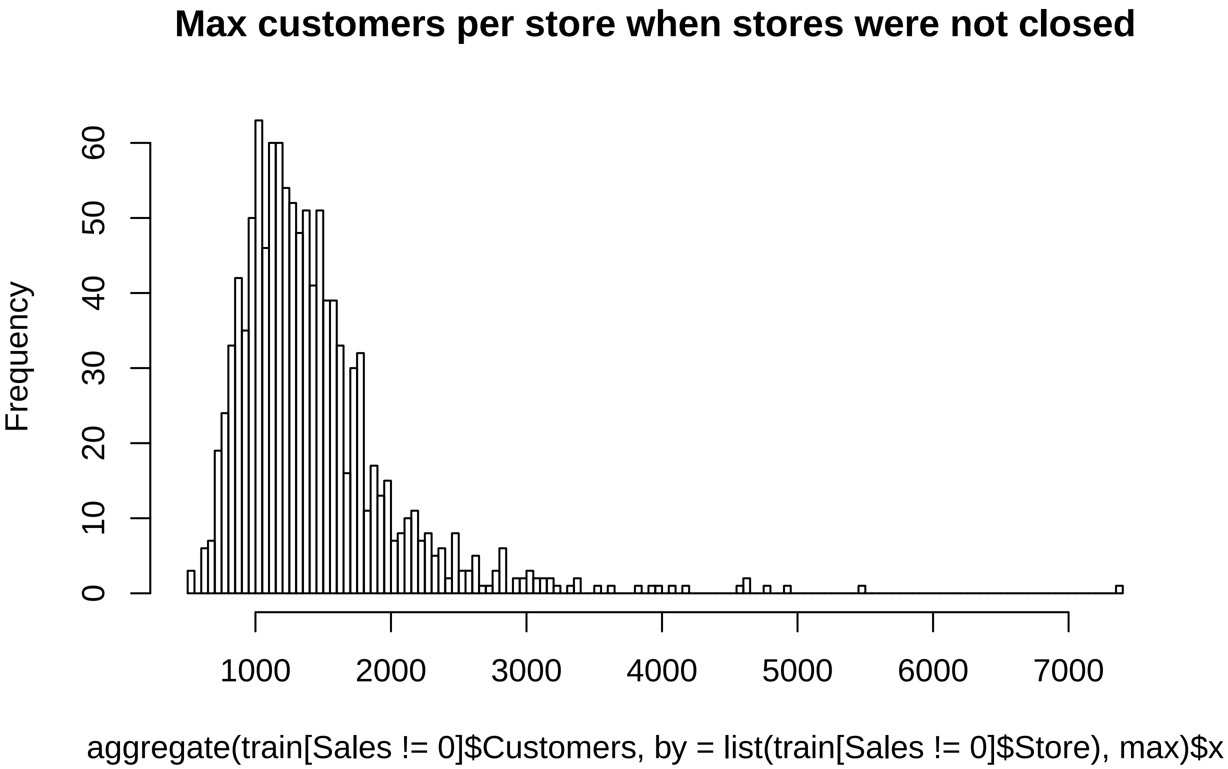
```
medianhist = hist(aggregate(train[Sales != 0]$Sales,
                by = list(train[Sales != 0]$Store), median)$x, 100,
     main = "Median sales per store when stores were not closed")
```



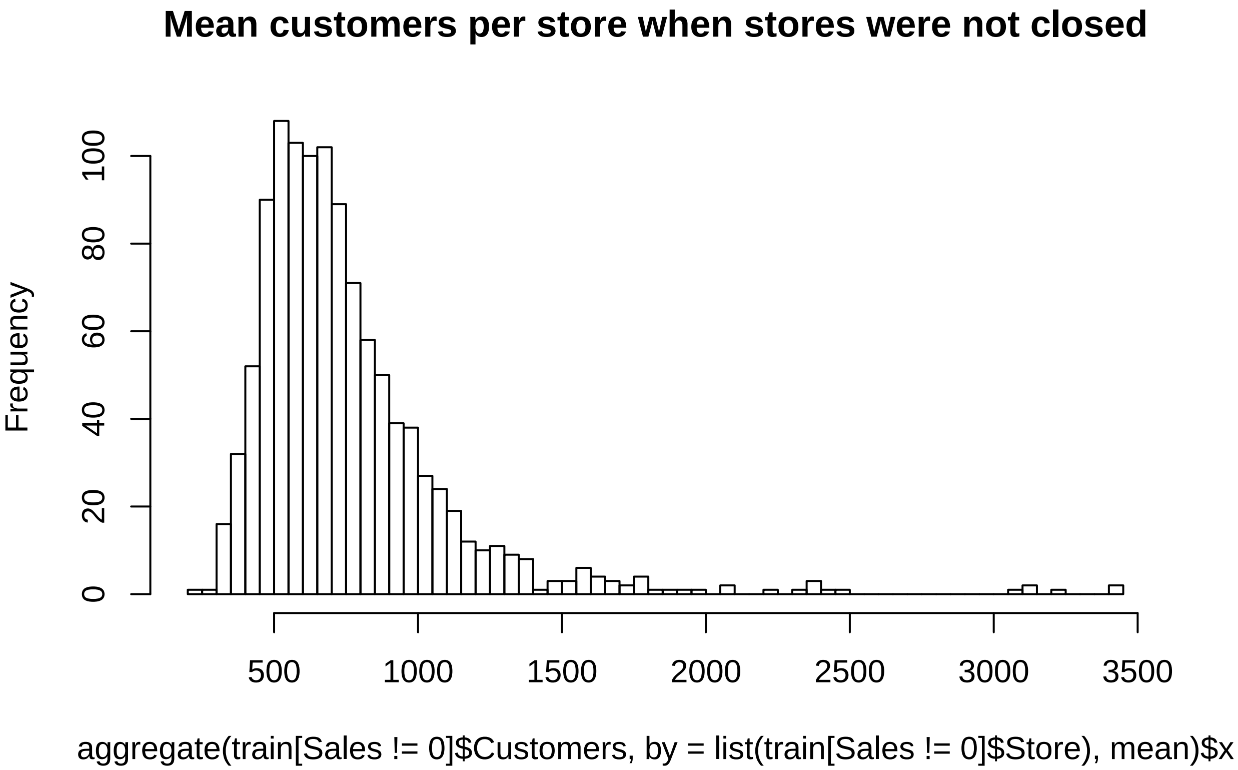
```
hist(train$Customers, 100)
```



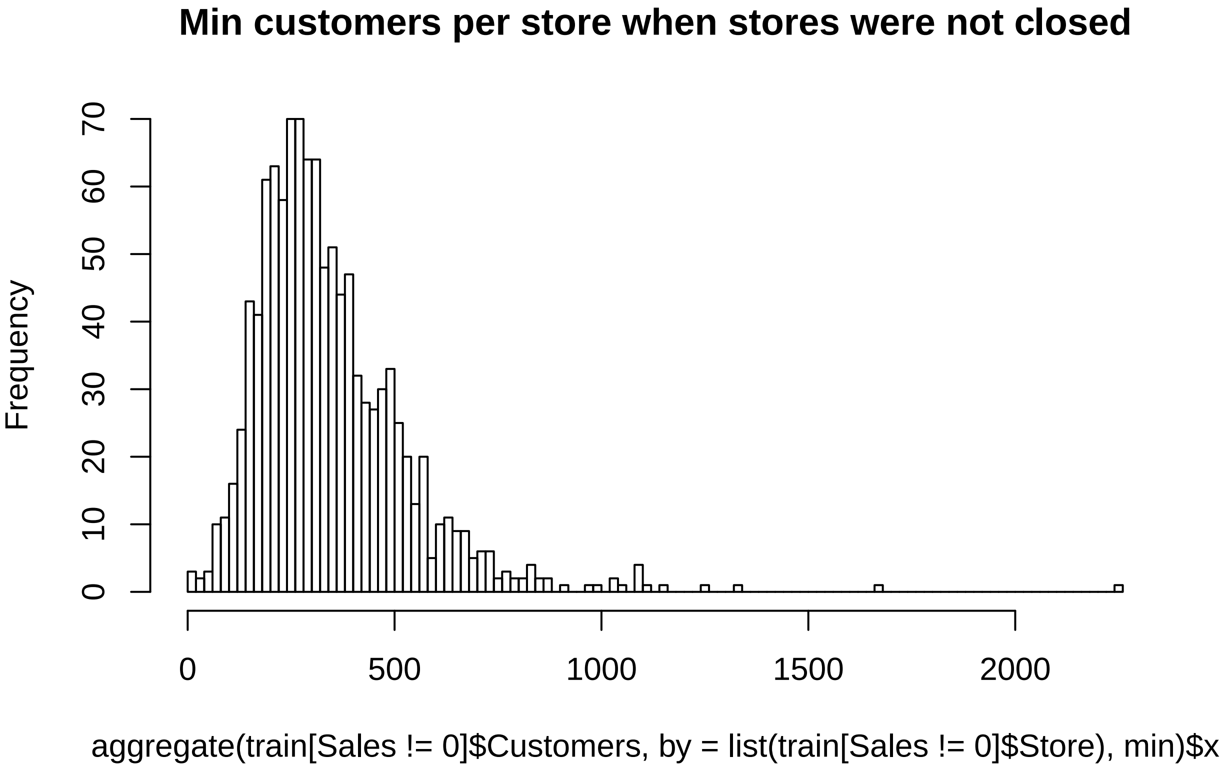
```
maxhist = hist(aggregate(train[Sales != 0]$Customers,
          by = list(train[Sales != 0]$Store), max)$x, 100,
     main = "Max customers per store when stores were not closed")
```



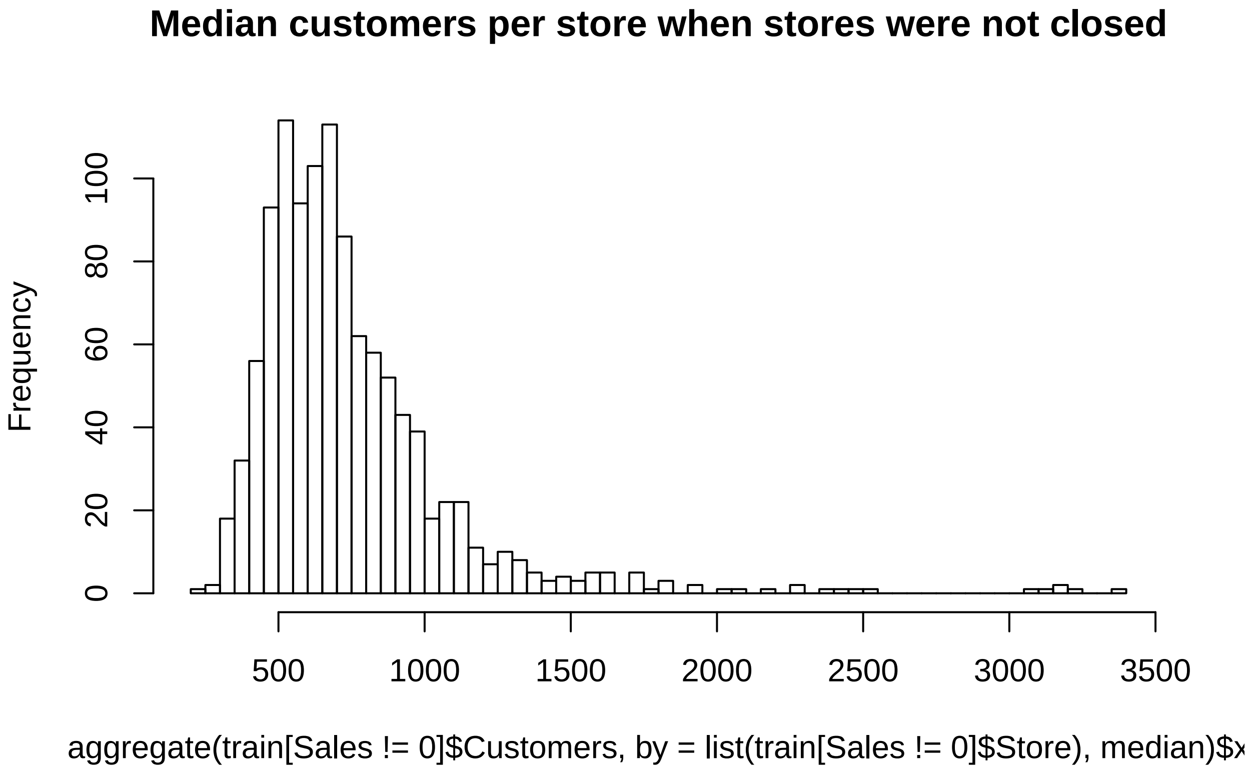
```
meanhist = hist(aggregate(train[Sales != 0]$Customers,
               by = list(train[Sales != 0]$Store), mean)$x, 100,
     main = "Mean customers per store when stores were not closed")
```



```
minhist = hist(aggregate(train[Sales != 0]$Customers,
                  by = list(train[Sales != 0]$Store), min)$x, 100,
     main = "Min customers per store when stores were not closed")
```



```
medianhist = hist(aggregate(train[Sales != 0]$Customers,
               by = list(train[Sales != 0]$Store), median)$x, 100,
     main = "Median customers per store when stores were not closed")
```


Median customers per store when stores were not closed
Frequency
aggregate(train[Sales != 0]$Customers, by = list(train[Sales != 0]$Store), median)$x