station-analysis

Julian DeGroot-Lutzner

12/14/2017

```
station_432 <- read.csv("station-432.csv")
station_521 <- read.csv("station-521.csv")
Weather_NYC <- read_csv("~/Documents/math154/ma154-project24-teambike/station-analysis/Weather_NYC.csv"
```

HERE COULD BE MORE ANALYSIS ABOUT EACH STATION

How far does the average citibike move in a week? Change avg. trip duration to median.

```
# Parsing all the start times into one format
mdy <- mdy_hms(station_432$starttime)
ymd <- ymd_hms(station_432$starttime)
f1 <- mdy_hm(station_432$starttime)
mdy[is.na(mdy)] <- ymd[is.na(mdy)]
station_432$starttime <- mdy
station_432$starttime[is.na(station_432$starttime)] <- f1[is.na(station_432$starttime)]

# Parsing all the start times into one format
mdy <- mdy_hms(station_521$starttime)
ymd <- ymd_hms(station_521$starttime)
f1 <- mdy_hm(station_521$starttime)
mdy[is.na(mdy)] <- ymd[is.na(mdy)]
station_521$starttime <- mdy
station_521$starttime[is.na(station_521$starttime)] <- f1[is.na(station_521$starttime)]
```

We wrote code to wrangle the data so that it is ready to make interesting graphs about each station but we never had time to make the graphs.

```
# Here is code that could be used to make graphs and understand
# the data at hand

# We took out rides that looped to the same station because these
# rides don't impact our prediction model
median_ <- function(...) median(..., na.rm=T)

# weekly sums of station 521
weekly_sums_521 <- station_521 %>%
  select(tripduration, starttime, start.station.id,
         end.station.id, usertype) %>%
  mutate(starttime = floor_date(starttime, "hour"),
         started.here = (start.station.id == 521),
         ended.here = (end.station.id == 521),
         subscriber = (usertype == "Subscriber"),
         customer = (usertype == "Customer")) %>%
  mutate(subscriber.started.here =
           (started.here & !ended.here & subscriber),
         subscriber.ended.here =
           (!started.here & ended.here & subscriber),
         customer.started.here =
           (started.here & !ended.here & subscriber),
         customer.ended.here =
           (!started.here & ended.here & subscriber)) %>%
  mutate_all(funs(ifelse(is.na(.), 0, .))) %>%
  mutate(duration.from.start =
          ifelse(started.here & !ended.here, tripduration, NA),
         duration.to.finish =
          ifelse(!started.here & ended.here, tripduration, NA)) %>%
  group_by(starttime) %>%
  summarize(median.trip.from.521 =
              median_(duration.from.start),
            median.trip.to.521 =
              median_(duration.to.finish),
            num.subscribers.started.521 =
              sum(subscriber.started.here),
            num.subscribers.ended.521 =
              sum(subscriber.ended.here),
            num.customers.started.521 =
              sum(customer.started.here),
            num.customers.ended.521 =
              sum(customer.ended.here),
            total.trips.started.521 =
              sum(started.here & !ended.here),
            total.trips.ended.521 =
              sum(!started.here & ended.here))

# weekly sums for station 432
weekly_sums_432 <- station_432 %>%
  select(tripduration, starttime, start.station.id,
         end.station.id, usertype) %>%
  mutate(starttime = floor_date(starttime, "week"),
         started.here = (start.station.id == 432),
         ended.here = (end.station.id == 432),
         subscriber = (usertype == "Subscriber"),
         customer = (usertype == "Customer")) %>%
  mutate(subscriber.started.here =
           (started.here & !ended.here & subscriber),
         subscriber.ended.here =
           (!started.here & ended.here & subscriber),
         customer.started.here =
           (started.here & !ended.here & subscriber),
         customer.ended.here =
           (!started.here & ended.here & subscriber)) %>%
  mutate_all(funs(ifelse(is.na(.), 0, .))) %>%
  mutate(duration.from.start =
          ifelse(started.here & !ended.here, tripduration, NA),
         duration.to.finish =
          ifelse(!started.here & ended.here, tripduration, NA)) %>%
  group_by(starttime) %>%
  summarize(median.trip.from.432 =
              median_(duration.from.start),
            median.trip.to.432 =
              median_(duration.to.finish),
            num.subscribers.started.432 =
              sum(subscriber.started.here),
            num.subscribers.ended.432 =
              sum(subscriber.ended.here),
            num.customers.started.432 =
              sum(customer.started.here),
            num.customers.ended.432 =
              sum(customer.ended.here),
            total.trips.started.432 =
              sum(started.here & !ended.here),
            total.trips.ended.432 =
              sum(!started.here & ended.here))
hourly_station_data <-
  left_join(hour_sums_432, hour_sums_521, by ="starttime")
```

```
# Here is code that could be used to make graphs and understand
# the data at hand

# We took out rides that looped to the same station because these
# rides don't impact our prediction model
median_ <- function(...) median(..., na.rm=T)

# weekly sums of station 521
weekly_sums_521 <- station_521 %>%
  select(tripduration, starttime, start.station.id,
         end.station.id, usertype) %>%
  mutate(starttime = floor_date(starttime, "hour"),
         started.here = (start.station.id == 521),
         ended.here = (end.station.id == 521),
         subscriber = (usertype == "Subscriber"),
         customer = (usertype == "Customer")) %>%
  mutate(subscriber.started.here =
           (started.here & !ended.here & subscriber),
         subscriber.ended.here =
           (!started.here & ended.here & subscriber),
         customer.started.here =
           (started.here & !ended.here & subscriber),
         customer.ended.here =
           (!started.here & ended.here & subscriber)) %>%
  mutate_all(funs(ifelse(is.na(.), 0, .))) %>%
  mutate(duration.from.start =
          ifelse(started.here & !ended.here, tripduration, NA),
         duration.to.finish =
          ifelse(!started.here & ended.here, tripduration, NA)) %>%
  group_by(starttime) %>%
  summarize(median.trip.from.521 =
              median_(duration.from.start),
            median.trip.to.521 =
              median_(duration.to.finish),
            num.subscribers.started.521 =
              sum(subscriber.started.here),
            num.subscribers.ended.521 =
              sum(subscriber.ended.here),
            num.customers.started.521 =
              sum(customer.started.here),
            num.customers.ended.521 =
              sum(customer.ended.here),
            total.trips.started.521 =
              sum(started.here & !ended.here),
            total.trips.ended.521 =
              sum(!started.here & ended.here))

# weekly sums for station 432
weekly_sums_432 <- station_432 %>%
  select(tripduration, starttime, start.station.id,
         end.station.id, usertype) %>%
  mutate(starttime = floor_date(starttime, "week"),
         started.here = (start.station.id == 432),
         ended.here = (end.station.id == 432),
         subscriber = (usertype == "Subscriber"),
         customer = (usertype == "Customer")) %>%
  mutate(subscriber.started.here =
           (started.here & !ended.here & subscriber),
         subscriber.ended.here =
           (!started.here & ended.here & subscriber),
         customer.started.here =
           (started.here & !ended.here & subscriber),
         customer.ended.here =
           (!started.here & ended.here & subscriber)) %>%
  mutate_all(funs(ifelse(is.na(.), 0, .))) %>%
  mutate(duration.from.start =
          ifelse(started.here & !ended.here, tripduration, NA),
         duration.to.finish =
          ifelse(!started.here & ended.here, tripduration, NA)) %>%
  group_by(starttime) %>%
  summarize(median.trip.from.432 =
              median_(duration.from.start),
            median.trip.to.432 =
              median_(duration.to.finish),
            num.subscribers.started.432 =
              sum(subscriber.started.here),
            num.subscribers.ended.432 =
              sum(subscriber.ended.here),
            num.customers.started.432 =
              sum(customer.started.here),
            num.customers.ended.432 =
              sum(customer.ended.here),
            total.trips.started.432 =
              sum(started.here & !ended.here),
            total.trips.ended.432 =
              sum(!started.here & ended.here))
hourly_station_data <-
  left_join(hour_sums_432, hour_sums_521, by ="starttime")
```

Hourly Sums

```r
# took out rides that looped to the same station because these
# rides don't impact our prediction model
median_ <- function(...) median(..., na.rm=T)

# hourly sums of station 521
hourly_sums_521 <- station_521 %>%
  select( starttime, start.station.id,
         end.station.id) %>%
  mutate(starttime = floor_date(starttime, "hour"),
         started.here = (start.station.id == 521),
         ended.here = (end.station.id == 521)) %>%
  group_by(starttime) %>%
  summarize(total.trips.started.521 =
              sum(started.here & !ended.here),
            total.trips.ended.521 =
              sum(!started.here & ended.here)) %>%
  mutate(net.change.521 =
  total.trips.started.521 -total.trips.ended.521 )

# hourly sums for station 432
hourly_sums_432 <- station_432 %>%
  select(starttime, start.station.id,
         end.station.id) %>%
  mutate(starttime = floor_date(starttime, "hour"),
         started.here = (start.station.id == 432),
         ended.here = (end.station.id == 432)) %>%
  group_by(starttime) %>%
  summarize(total.trips.started.432 =
              sum(started.here & !ended.here),
            total.trips.ended.432 =
              sum(!started.here & ended.here)) %>%
  mutate(net.change.432 =
  total.trips.started.432 -total.trips.ended.432 )
```

Prepare the Weather Data

```r
# Choosing important variables
Weather_NYC <- Weather_NYC %>%
  select(valid, tmpf, dwpf, relh, vsby)
```

```r
Weather_NYC <- Weather_NYC %>%
  mutate(valid = ymd_hms(valid)) %>%
  filter(minute(valid)=="51")  %>%
  mutate(valid = ceiling_date(valid, unit = "hour"),
         Month=month(valid)) %>%
  mutate( summer=ifelse(Month=="6"|Month=="7"|Month=="8",1,0),
          spring=ifelse((Month=="3"|Month=="4"|Month=="5"),1,0),
          winter=ifelse((Month=="1"|Month=="2"|Month=="12"),1,0),
          fall=ifelse((Month=="9"|Month=="10"|Month=="11"),1,0),
          day.of.week=wday(valid),
          hour = hour(valid)) %>%
  mutate(week.day=
           ifelse(day.of.week > 1 & day.of.week < 7,TRUE,FALSE),
         weekend.day=
           ifelse(day.of.week == 1 | day.of.week == 7,TRUE,FALSE),
         EarlyMorning=
           ifelse(hour=="0"|hour=="1"|hour=="2"|hour=="3"|hour=="4"|hour=="5"|hour=="6",1,0),
         Commuting=
           ifelse((hour=="7"|hour=="8"|hour=="9"),1,0),
         DayTime=
           ifelse((hour=="10"|hour=="11"|
              hour=="12"|hour=="13"|hour=="14"|hour=="15"),1,0),
         Evening=ifelse((hour=="16"|hour=="17"|
                           hour=="18"|hour=="19"),1,0),
         Night=ifelse((hour== "20"|
                         hour=="21"|hour=="22"|hour=="23"),1,0),
         starttime = valid) %>%
  select(-valid, -hour, -Month, -day.of.week)
```

```r
station_432_combined <- hourly_sums_432 %>%
  select(starttime, net.change.432 ) %>%
  inner_join(Weather_NYC, by= "starttime" )
```

```r
station_521_combined <- hourly_sums_521 %>%
  select(starttime, net.change.521 ) %>%
  inner_join(Weather_NYC, by= "starttime" )
```

```r
print.data.frame(head(station_521_combined,1))
```

```
##             starttime net.change.521  tmpf  dwpf  relh vsby summer spring
## 1 2013-07-01 02:00:00             -1 75.02 69.98 84.34    8      1      0
##   winter fall week.day weekend.day EarlyMorning Commuting DayTime Evening
## 1      0    0     TRUE       FALSE            1         0       0       0
##   Night
## 1     0
```

```r
dim(station_521_combined)
```

```
## [1] 21646    17
```

```r
print.data.frame(head(station_432_combined,1))
```

```
##             starttime net.change.432  tmpf  dwpf  relh vsby summer spring
## 1 2013-07-01 02:00:00              1 75.02 69.98 84.34    8      1      0
##   winter fall week.day weekend.day EarlyMorning Commuting DayTime Evening
## 1      0    0     TRUE       FALSE            1         0       0       0
##   Night
## 1     0
```

```r
dim(station_432_combined)
```

```
## [1] 31743    17
```

```r
nrow(na.omit(station_521_combined))
```

```
## [1] 21107
```

```r
station_521_combined <- station_521_combined[complete.cases(station_521_combined),]
```

```
station_521_combined <- station_521_combined %>%
  select(-starttime)
print.data.frame(head(station_521_combined,1))
```

```
##   net.change.521  tmpf  dwpf  relh vsby summer spring winter fall week.day
## 1             -1 75.02 69.98 84.34    8      1      0      0    0     TRUE
##   weekend.day EarlyMorning Commuting DayTime Evening Night
## 1       FALSE            1         0       0       0     0
```

```
dim(station_521_combined)
```

```
## [1] 21107    16
```

Random Forest

Random Forest

How many trees is enough?

```
#results <- data.frame()
#for(n in seq(from= 50, to = 750, by= 200)){
set.seed(47)
rf.model <- train(net.change.521~.,
                  data = training.521,
                  method = "rf",
                  trControl=trainControl(method="oob"),
                  ntree = 200,
                  tuneGrid=data.frame(mtry=6))
prediction <- predict(rf.model, testing.521)
#cm <- confusionMatrix(prediction, testing.521$)
#results <- rbind(results , cm$overall)
#}
#results

RMSE(prediction, testing.521$net.change.521)
```

```
## [1] 16.88
```

```
actual_positive_change <- (testing.521$net.change.521 > 0)
pred_positive_change <- (prediction > 0)
confusionMatrix(pred_positive_change, actual_positive_change)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2241  925
##      TRUE   1230 1934
##
##                Accuracy : 0.66
##                  95% CI : (0.648, 0.671)
##     No Information Rate : 0.548
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.319
##  Mcnemar's Test P-Value : 5.81e-11
## 
##             Sensitivity : 0.646
##             Specificity : 0.676
##          Pos Pred Value : 0.708
##          Neg Pred Value : 0.611
##              Prevalence : 0.548
##          Detection Rate : 0.354
##    Detection Prevalence : 0.500
##       Balanced Accuracy : 0.661
## 
##        'Positive' Class : FALSE
## 
```

```r
all_true <- (prediction < -10000)
confusionMatrix(all_true, actual_positive_change)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction FALSE TRUE
##      FALSE  3471 2859
##      TRUE      0    0
## 
##                Accuracy : 0.548
##                  95% CI : (0.536, 0.561)
##     No Information Rate : 0.548
##     P-Value [Acc > NIR] : 0.505
## 
##                   Kappa : 0
##  Mcnemar's Test P-Value : <2e-16
## 
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.548
##          Neg Pred Value :   NaN
##              Prevalence : 0.548
##          Detection Rate : 0.548
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
## 
##        'Positive' Class : FALSE
## 
```

```r
head(prediction)
```

```
##          1          2          3          4          5          6
##   3.130707 -10.755238   3.047207  21.924906  -0.002622 -24.097631
```

```r
head(testing.521$net.change.521)
```

```
## [1]   7 -10   7  27  -1  -5
```

```r
mean(prediction)
```

```
## [1] 1.86
```

linear model

SVM

```
station.521.svm<- station_521_combined %>%
  mutate(net.change.521 = ifelse(net.change.521 > 0, "P","N"))

set.seed(4747)
inTrain <-
  createDataPartition(station.521.svm$net.change.521,
                      p = 0.7, list=FALSE)
training.svm.521 <- station.521.svm[inTrain, ]
testing.svm.521 <- station.521.svm[-inTrain,]
head(training.svm.521)
```

```
## # A tibble: 6 x 16
##   net.change.521  tmpf  dwpf  relh  vsby summer spring winter  fall
##            <chr> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl>
## 1              N 73.94 69.98 87.45     9      1      0      0     0
## 2              P 75.02 69.98 84.34     7      1      0      0     0
## 3              N 75.02 71.06 87.50     6      1      0      0     0
## 4              P 75.92 71.96 87.55     7      1      0      0     0
## 5              P 73.04 71.06 93.52     6      1      0      0     0
## 6              N 73.04 71.06 93.52     8      1      0      0     0
## # ... with 7 more variables: week.day <lgl>, weekend.day <lgl>,
## #   EarlyMorning <dbl>, Commuting <dbl>, DayTime <dbl>, Evening <dbl>,
## #   Night <dbl>
```

```
head(testing.svm.521)
```

```
## # A tibble: 6 x 16
##   net.change.521  tmpf  dwpf  relh  vsby summer spring winter  fall
##            <chr> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl>
## 1              N 75.02 69.98 84.34     8      1      0      0     0
## 2              P 73.94 69.98 87.45     9      1      0      0     0
## 3              N 75.02 73.04 93.57     6      1      0      0     0
## 4              N 75.02 73.04 93.57     4      1      0      0     0
## 5              P 73.04 71.06 93.52     8      1      0      0     0
## 6              P 73.04 71.06 93.52     7      1      0      0     0
## # ... with 7 more variables: week.day <lgl>, weekend.day <lgl>,
## #   EarlyMorning <dbl>, Commuting <dbl>, DayTime <dbl>, Evening <dbl>,
## #   Night <dbl>
```

```
set.seed(47)
svm.linear.model <- train(net.change.521~., data = training.svm.521, method="svmLinear",
                 trControl = trainControl(method="cv"),
                 tuneGrid= expand.grid(C= (0.1)),
                 preProcess = c("center", "scale"))

svm.linear.pred <- predict(svm.linear.model, testing.svm.521)

confusionMatrix(svm.linear.pred, testing.svm.521$net.change.521)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction    N    P
##          N 2614 1451
##          P  858 1408
##
##                Accuracy : 0.635
##                  95% CI : (0.623, 0.647)
##     No Information Rate : 0.548
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.25
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.753
##             Specificity : 0.492
##          Pos Pred Value : 0.643
##          Neg Pred Value : 0.621
##              Prevalence : 0.548
##          Detection Rate : 0.413
##    Detection Prevalence : 0.642
##       Balanced Accuracy : 0.623
##
##        'Positive' Class : N
##
```