Final Script

Rachel Zhang (yz2334)

4/29/2020

```
convert_date = function(x){ #to convert date
  require(dplyr)
  new = as.Date(x, '%y.%m.%d') %>%
     format(., '20%y-%m-%d') %>%
    as.Date()
  return (new)
}

materr = function(data){
  require(dplyr)
  m = data
  for (j in 2:ncol(data)){
    m[,j] = (data[,j] - data[,1]) #calculating errors for each forecast
  }
  ME = colMeans(m)
  MAE = colMeans(abs(m))
  SD = apply(m,2,sd)
  p = data
  for (i in 2:ncol(p)){
    p[, i] = m[,i] / data[,1]
  }
  MPE = colMeans(p)
  MAPE = colMeans(abs(p))
  return(rbind(ME,MAE,MPE,MAPE,SD))
}
```

Data Importing and Cross-Validation

```
dt <- read.csv("~/Documents/Research/Thesis writing/NewData/Arr.csv") %>%
  data.frame()%>%
  select(Arrival.Date, Booking.Window, Quantity) %>%
  mutate_at(., vars(Arrival.Date),  funs(as.Date(., "%m/%d/%Y"))) %>%
  group_by(Arrival.Date, Booking.Window) %>%
  mutate(Quan = sum(Quantity)) %>%
  dplyr::arrange(., Arrival.Date, Booking.Window, Quan)
dt = dt[-(1:2),-3] #throw out the first two observations to avoid inconsistency
dt$Arrival.Date = convert_date(dt$Arrival.Date)

wide = dcast(dt, Arrival.Date ~ Booking.Window, value.var='Quan') %>%
  data.frame() %>%
  arrange(., Arrival.Date)

for (i in (ncol(wide)-1):2){
  wide[i] = wide[i] + wide[i+1]
}
```

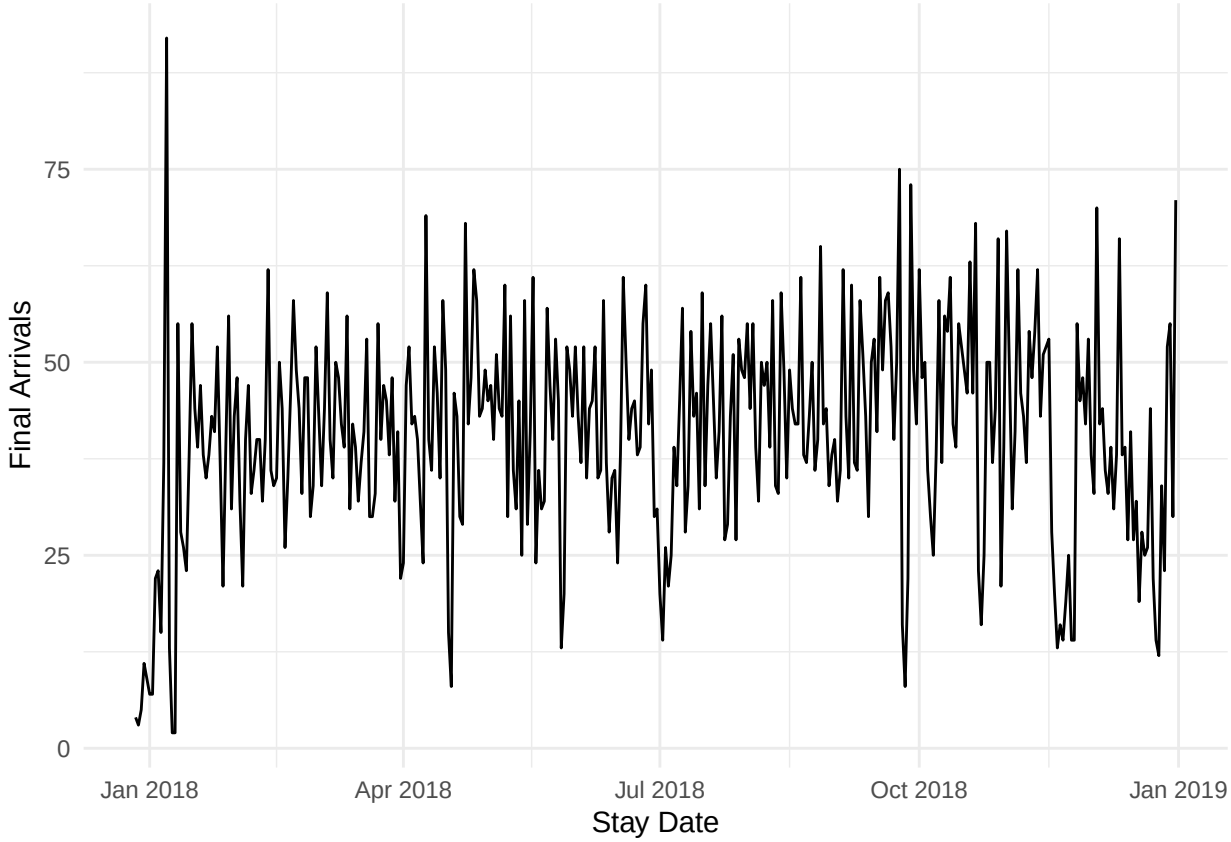
```
colnames(wide) = c('Arrival.Date', 'ROH0', paste0('ROH', 1:(ncol(wide)-2)))
wide = wide %>%
  remove_rownames %>% column_to_rownames('Arrival.Date')

kable(wide[1:10, 1:10], 'latex', caption = 'Data Overview',
        longtable = F, booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position", "scale_down" ,"repeat_header"))
```

Table 1: Data Overview

	ROH0	ROH1	ROH2	ROH3	ROH4	ROH5	ROH6	ROH7	ROH8	ROH9
2017-12-27	4	4	4	4	4	4	4	4	4	4
2017-12-28	3	3	3	3	3	3	3	3	3	3
2017-12-29	5	5	5	5	5	5	5	5	5	5
2017-12-30	11	11	11	11	11	11	11	11	11	11
2017-12-31	9	8	8	8	5	5	5	5	5	5
2018-01-01	7	5	5	5	3	3	3	3	3	3
2018-01-02	7	5	5	5	5	5	5	5	5	5
2018-01-03	22	18	17	16	16	15	15	15	15	15
2018-01-04	23	16	16	15	15	13	13	13	12	12
2018-01-05	15	11	10	9	9	9	9	8	6	6

```
ggplot(wide, aes(x=as.Date(unlist(rownames(wide))), y=ROH0)) +
  geom_line() +
  theme_minimal()+
  xlab("Stay Date") + ylab('Final Arrivals') +
  theme(plot.caption = element_text(hjust = 0))
```




```
agg = c(1, 2, 3, 4, 5, 6, 7, 14, 21, 30, 60, 90)
wide_agg = wide[1:nrow(wide), c(1,agg+1)]
```

```
set.seed(2020)
tr_ind = sample(nrow(wide), 0.8*nrow(wide))
train = wide_agg[tr_ind, ]
test = wide_agg[-tr_ind, ]
#comp_time =
```

Modeling

Additive Pick-up

```
#calculating additive pick up
apk = train
for (j in ncol(apk):1){
  apk[,j] = apk[,1] - apk[,j]
}
apk = colMeans(apk) %>% as.data.frame() %>% t()

apk.pre = test
apk.pre[, 2:ncol(apk.pre)] = NA

for (i in 1:(nrow(apk.pre))){
  apk.pre[i,2:ncol(apk.pre)] = test[i,2:ncol(apk.pre)] + apk
```

```
}
apk.err = t(materr(apk.pre)) %>% data.frame()
```

Multiplicative Pick-up

```
mpk = train
for (j in ncol(mpk):1){
  mpk[,j] = mpk[,j] / mpk[,1]
}
mpk = colMeans(mpk) %>% as.data.frame() %>% t()

mpk.pre = test
mpk.pre[, 2:ncol(mpk.pre)] = NA

for (i in 1:(nrow(mpk.pre))){
  mpk.pre[i,2:ncol(mpk.pre)] =
    test[i,2:ncol(mpk.pre)] / mpk
}

mpk.err = t(materr(mpk.pre)) %>% data.frame()
```

Regression

```
reg.pred = test
reg.pred[,2:ncol(reg.pred)] = NA
reg = vector(mode='list')
for (i in agg){
  this.predictor = paste0(paste0('ROH',agg[which(agg==i)]), collapse='+')
  lm.formula = paste('ROH0', this.predictor, sep = '~')
  reg[[this.predictor]] = lm(lm.formula, data = train)
  reg.pred[, which(names(reg.pred)==paste0('ROH',i))]=predict(reg[[this.predictor]], test)
}
reg.err = t(materr(reg.pred)) %>% data.frame()

texreg(reg, 'latex')
```