Hourly EV Charging Demand Model

Danny Ettelson

11/27/2018

```r
#load libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0      v purrr   0.2.5
## v tibble  1.4.2      v dplyr   0.7.8
## v tidyr   0.8.2      v stringr 1.3.1
## v readr   1.1.1      v forcats 0.3.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
library(RColorBrewer)


#load data
#Price Schedule
Winter_TOU_2018 <- read.csv("Model_Map/2018_Winter_TOU_EV_D.csv")
Baseline_03_18_WP <- read.csv("Model_Map/03-18_WP_Avg.csv")
Chargers <- read.csv("Model_Map/Chargers_Installed_03-18.csv")
```

```r
#Price Schedule
price_schedule <- data.frame(Hr = c(1:24)) %>%
  mutate(P0 = Winter_TOU_2018$P0)

#Baseline
WP_Chargers <- Chargers$Workplace #Number of Chargers (C)
EV_Demand <- mutate(price_schedule, Xi = Baseline_03_18_WP$Xi, X0 = Baseline_03_18_WP$Xi/340*WP_Chargers
```

```r
#creates our matrix based on the 24 elasticity .csv files.
#uses a for loop to call files rather than individually
#NOTE this matrix has each COLUMN to be used for each hour. Our excel used each ROW if trying to compar

x <- c(1:24) #used for the 24 hours in for loops (24 elasticity columns)
matrix <- data.frame(Hr = c(1:24))
for (val in x) {
  El <- read_csv(sub("XX", val, "Model_Map/Elasticities_XX.csv"))
  El <- El[-1,]
  El <- El[order(El$HR24),]
  matrix <- cbind(matrix, El$ELAST)
}
```

```
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
```

```
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
```

```
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
```

```
##   HR24 = col_integer()
## )
## Parsed with column specification:
## cols(
##   ELAST = col_double(),
##   HR24 = col_integer()
## )
matrix<-matrix[,-1]
colnames(matrix) <- c(1:24)

price_change <- 0
intervention_hours <- c(12:15)
EV_Demand <- mutate(EV_Demand, P1 = price_schedule$P0) #Adds price schedule with intervention (P1)

EV_Demand$P1[intervention_hours] <-EV_Demand$P1[intervention_hours] + price_change #updates interventio

EV_Demand <- mutate(EV_Demand, P1p = (P1-P0)/P0) #Adds percentage change in price (P1p)

X1p <- as.vector(0)
for (val in x) {
  mat <- sub("XX",val, "matrix$`XX`")
  sum_prod <- crossprod(EV_Demand$P1p,eval(parse(text = mat)))
  X1p<- append(X1p,sum_prod)

} #crossprod() multiplies sumproduct of the percent change in price with each column in the matrix. Thi

X1p <- X1p[-1] # gets rid of the first dummy entry to the variable
EV_Demand <- mutate(EV_Demand, X1p = X1p) #add percent change in demand due to price onto EV_Demand (X1p

EV_Demand <- mutate(EV_Demand, X1 = (1+X1p)*X0) #adds new demand in kW variable (X1)

throttle_amount <- -0.5 #throttling amount -0.5 - 50%
Tp <- rep(0,24)
throttle_hours <- c(10) #hours that throttling occurs
Tp[throttle_hours] <- throttle_amount #Assigns each hour a throttling percentage
EV_Demand <- mutate(EV_Demand, Tp=Tp) #Adds throttling percentage to each hour (Tp)

E <- as.vector(0) #dummy for self-elasticities (E)

for (val in x) {
  self <- matrix[val,val]
  E <- append(E,self)
} #checks the matrix for the self elasticity and adds it to a vector, repeated 24 times as a for loop r

E <- E[-1] #removing dummy

EV_Demand <- mutate(EV_Demand, Ptp = Tp/E, Pt = (1+Ptp)*P1)#Creates an equivalent change in price durin

Xtp <- as.vector(0) #Dummy for change in demand due to throttling (Xtp)

for (val in x) {
  mat <- sub("XX",val, "matrix$`XX`")
  sum_prod <- crossprod(EV_Demand$Ptp,eval(parse(text = mat)))
  Xtp<- append(Xtp,sum_prod)
```

```r
} #uses new price equivalent sumproduct with each hour's elasticities to find percent change in demand
Xtp <- Xtp[-1] #gets rid of dummy

EV_Demand <- mutate(EV_Demand, Xtp = Xtp, Xt = (Xtp+1)*X1) #adds (Xtp) and the new demand in kW due to

#to be completed...

#need to stack both demand curves in one data frame for a legend

graph_table <- EV_Demand[c(1,4,13)] %>%
  gather(condition,value,X0:Xt)


Demand_Graph <- ggplot(data = graph_table, aes(x = Hr)) +
  geom_line(aes(y = value, color=condition)) +
 theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
       panel.background = element_blank(), axis.line = element_line(colour = "black")) +
  labs(title="Hourly Demand Forecast",
       subtitle="50% Throttling 9 AM - 10 AM",
       y="EV Charging Demand (kW)",
       x="Hour",
       color=NULL) +
  scale_x_continuous(breaks = 1:24, limits = c(1,24), expand = c(0, 0)) +
  scale_color_manual(labels=c("Baseline Demand","Demand with Intervention"), values = c("blue", "red"))
  geom_rect(aes(xmin=9,xmax=10,ymin=-Inf,ymax=Inf,fill="Throttle"),alpha=0.0075) +
  #geom_rect(aes(xmin=11,xmax=15,ymin=-Inf,ymax=Inf, fill = "Discount"),alpha=0.0075) +
  #theme_bw() + # theme_bw() gives grid lines, could remove
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))+
 theme(legend.position="bottom") +
  scale_fill_manual('Interventions',values = c('yellow'),  guide = guide_legend(override.aes = list(alph


# Aesthetic Notes:
# might try to remove space b/w Y-axis and hr0, as well as past hr24
# bold axis titles
# color brewer
# Midnight to 1AM is missing!

Demand_Graph
```

Hourly Demand Forecast

50% Throttling 9 AM – 10 AM