Monte Carlo Simulation

Danny Ettelson

1/21/2019

```
#load libraries
library(tidyverse)
library(dplyr)
library(RColorBrewer)


#LOAD DATA
#Price Schedule
price_schedule <- read_csv("Model_Map/2018_Summer_TOU_EV_4.csv")

#Price Schedule Options (All in Model_Map Folder)

#2018 Summer 3
#read_csv("Model_Map/2018_Summer_TOU_EV_3.csv")

#2018 Summer 4
#read_csv("Model_Map/2018_Summer_TOU_EV_4.csv")

#2018 Winter 3
#read_csv("Model_Map/2018_Winter_TOU_EV_3.csv")

#2018 Winter 4
#read_csv("Model_Map/2018_Winter_TOU_EV_4.csv")

#2018 Winter D
#read_csv("Model_Map/2018_Winter_TOU_EV_D.csv")

#2019 Summer 8
#read_csv("Model_Map/2019_Summer_TOU_EV_8.csv")

#2019 Winter 8
#read_csv("Model_Map/2019_Winter_TOU_EV_8.csv")


#Baseline Usage
#baseline <- read_csv("Model_Map/03-18_WP_Avg.csv")

DC_baseline <- read_csv("Model_Map/DC_Avg_Usage.csv")
WP_baseline <- read_csv("Model_Map/Workplace_Avg_Usage.csv")
MUD_baseline <- read_csv("Model_Map/MUD_Avg_Usage.csv")
F_baseline <- read_csv("Model_Map/Fleet_Avg_Usage.csv")


baseline <- bind_rows("Destination Center" = DC_baseline, "Fleet" = F_baseline, "Multi Unit Dwelling" =
View(baseline)

Wokplace_Total_Usage <- read_csv("Model_Map/Wokplace_Total_Usage.csv")
```

```r
Workplace_Avg_Usage <- read_csv("Model_Map/Workplace_Avg_Usage.csv")

# Number of Chargers by Segment
#chargers <- read_csv("Model_Map/Chargers_Installed_03-18.csv")
Chargers <- read_csv("Model_Map/Chargers.csv")
Event_Chargers <- read_csv("Model_Map/Event_Chargers.csv")

add_baseline_chargers <- Chargers %>%
  filter(Market_Segment!= "Total") %>%
  slice(rep(1:n(),each=24))

#WHY REPEAT 24 TIMES?? time doesnt matter here

#Event Usage
DC_Event_Total_Usage <- read_csv("Model_Map/DC_Event_Total_Usage.csv")


#Elasticities with format 9X3 with columns Base_Hr, Changed_Hr, and Elasticity
#Changed_Hr is the Hour where the price change occurs, Base_Hr is the hour in which demand changes
Elasticities <- read_csv("SDGE_Elasticities.csv")
SDGE_P_SOP_Ratios <- read_csv("SDGE_P_SOP_Ratios.csv")


#Ratio for selecting Default Elasticities
P_SOP_Ratio <- max(price_schedule$P0)/min(price_schedule$P0)
#Matches our closest Ratio to Inputted Ratio
closest_schedule <- SDGE_P_SOP_Ratios$Rate_Schedule[which.min(abs(SDGE_P_SOP_Ratios$P_SOP_Ratio - P_SOP
closest_elasticities <- match(closest_schedule, names(Elasticities))
#Uses Elasticities of rate schedule with closest ratio
```

```r
#Set variables and values
#All defaults are set to MARCH 2018 WORKPLACE
p_c <- -0.05 #price change
i_h <- c(12:15) #intervention hours
t_a <- 0 #throttling amount
t_h <- c(7:11) #throttling hours
sch <- closest_elasticities #elasticities to use for price intervention (column in the elasticities dat
sg <- "Workplace" #segment
mth <- "Mar_18" #month
pwr <- 6.6 #charger power
pk <- c(17:21) #target window to shift out off (this is only used in the output calculations below, not
int_ch <- filter(Chargers, Market_Segment == sg) %>%
  select(mth) %>%
  as.numeric() # default is to MARCH 2018
int_e_b <- TRUE
i_c_e <- 1



hourly_demand <- function(segment = sg,
                          month = mth,
                          charger_power = pwr,
                          schedule = sch,
                          price_change = p_c,
```

```
                          intervention_hours = i_h,
                          intervention_chargers = int_ch,#if intervention_chargers is
                          int_equals_baseline = int_e_b,
                          throttle_amount = t_a,
                          throttle_hours = t_h,
                          intervention_comm_effect = i_c_e){

  #CONTEXT#####

  #This section puts the Hr, initial price (P0), period, initial unscaled load (Xi), and scaled load (X

  #Price Schedule is read in above

# Elasticity
chosen_elasticities <- Elasticities[c(1,2,schedule)] #this pulls out columns 1, 2, and the designated e
colnames(chosen_elasticities) <- c("Base_Hr","Changed_Hr","Elasticity")


#price_schedule$period <- factor(price_schedule$period, levels = c("P","MP","OP"))

#Baseline

#filter the number of chargers by market segment and month, change to numeric (have to set the month an
baseline_chargers <-filter(Chargers, Market_Segment == segment) %>%
  select(month) %>%
  as.numeric()


intervention_chargers <- ifelse(int_equals_baseline == TRUE, baseline_chargers, intervention_chargers)

#WP_Chargers <- chargers$Workplace #Number of Chargers (C)
#DC_Chargers <- chargers$Workplace #Number of Chargers (C)
baseline_month<- baseline %>%
  filter(Segment == segment) %>%
  select(month) %>%
  unlist()


EV_Demand <- mutate(price_schedule, I01 = 0 ,Xi = baseline_month, X0 = baseline_month/baseline_chargers

EV_Demand$I01[intervention_hours] <-1


  #MAX_THEORETICAL####
#Theoretical max is based on the current number of chargers in the SCE Charge Ready pilot program, mult


Max_Theory <- intervention_chargers*charger_power


####
```

```r
  #SPLINING####
x <- c(1:24) #used for the 24 hours in for loops (24 elasticity columns)


#This makes a table for each hour that lists the midpoint hours that will be splined, hours as <24, rat


#Finds the hours in the rate schedule just before the period changes
change_points <- which(price_schedule$Period != dplyr::lag(price_schedule$Period)) - 1

#Finds the midpoints of each "chunk" of rate periods unless the "chunk" spans over the end of the day
mid_points <- change_points[-length(change_points)] +diff(change_points)/2

#finds the midpoint of the "chunk" of rate period that spans over the end of the day
rollover_midpoint <- (change_points[1]+24 + change_points[length(change_points)])/2 -24

#adds rollover_midpoint but only if there is actually a rate period "chunk" that rolls over the day
if(price_schedule$Period[1] == price_schedule$Period[length(price_schedule$Period)]) {
  mid_points <- append(mid_points, rollover_midpoint)
}

  #create a dataframe of the own and cross elasticities of the midpoints and self point (i.e., put the
for(i in x) {
    nam <- paste("Midpoints", i, sep = ".")
    Hrs <- append(mid_points, i)
    Hrs <- Hrs[-match(price_schedule$Period[i],price_schedule$Period[mid_points])]
#The loop above selects a set of midpoints that leaves out one midpoint based on the hour that a table

    Hrs24 <- append(Hrs,i) #adds the end point (the starting hour 24 hours later)
    Hrs <- if_else(Hrs<i,Hrs+24,Hrs) %>%
      append(i+24)
    #lists "real hours" from the starting point, adding 24 to any hours before the start point

    periods <- price_schedule$Period[c(Hrs24)]
    #retrieves the rate periods of each hour listed

    own_period <- price_schedule$Period[i]
    #retrieves rate period of the current hour
    own_period_elasticities <- filter(chosen_elasticities, Base_Hr == own_period)
    midpoint_elasticities <- own_period_elasticities$Elasticity[match(periods, Elasticities$Changed_Hr)]


    assign(nam,data.frame(Hour=Hrs,Hrs24=Hrs24, Period=periods,Elasticity = midpoint_elasticities))
    #makes a data frame named after the current hour with each of the above variables

}

#spline the midpoint table
for (i in x) {
  current_hr <- eval(parse(text = sub("XX", i, "Midpoints.XX")))
  #calls current hours midpoint table

  Y = spline(x=current_hr$Hour,y=current_hr$Elasticity,xout=seq(min(current_hr$Hour),max(current_hr$Hou
```

```
  #splines elasticities to smoooth

  HR = Y$x

  ELAST = Y$y

  nam <- paste("Elasticities", i, sep = ".")

  assign(nam,data.frame(HR=HR,ELAST=ELAST,HR24 = if_else(HR<=24,HR,HR-24)))
  #makes a data frame with above variables: Hours, smoothed elasticities
}
####

  #MATRIX####

#creates our matrix based on the 24 smoothed elasticities for each hour.
#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

matrix <- data.frame(Hr = c(1:24))
for (i in x) {
  El <- eval(parse(text = sub("YY", i, "Elasticities.YY")))
  El <- El[-1,]
  El <- El[order(El$HR24),]
  matrix <- cbind(matrix, El$ELAST)
}
matrix<-matrix[,-1]
colnames(matrix) <- c(1:24)
####

###set matrix to no cross_elasticititities
#matrix <- read_csv("No_Cross_Matrix.csv")


  #INTERVENTION & COMMUNICATION####


#price_change <- -0.05
#intervention_hours <- c(12:15)
EV_Demand <- EV_Demand %>%
  mutate(P1 = price_schedule$P0) #Copies the initial price schedule into a new column (P1) that can the

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


#Adds percentage change in price (P1p)
EV_Demand <- EV_Demand %>%
  mutate(P1p = (P1-P0)/P0) %>%
  mutate(P1pC = P1p*intervention_comm_effect)


X1p <- as.vector(0)
for (val in x) {
```

```
  mat <- sub("XX",val, "matrix$`XX`")
  sum_prod <- crossprod(EV_Demand$P1pC,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 (X1

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

####


  #THROTTLING####

#throttle_amount <- 0 #throttling amount -0.5 - 50%
Tp <- rep(0,24)
#throttle_hours <- c(7:11) #hours that throttling occurs
Tp[throttle_hours] <- throttle_amount #Assigns the throttling intervention percentage chosen at the inp

#Adds throttling percentage to each hour (Tp)

EV_Demand <- EV_Demand %>%
  mutate(Tp=Tp) %>%
  mutate(Xt = (1+Tp)*X1)


####


  #SHIFTING/FINAL####

#The variables below quantify the shift and net change in demand as a result of interventions, and need

Total_x0 <- sum(EV_Demand$X0)
Total_xt <-sum(EV_Demand$Xt)

Net_Change <- Total_xt-Total_x0
Change_intervention <- sum(EV_Demand$Xt[intervention_hours]) - sum(EV_Demand$X0[intervention_hours])
Change_outside_intervention <- sum(EV_Demand$Xt[-intervention_hours])- sum(EV_Demand$X0[-intervention_h




EV_Demand <- mutate(EV_Demand, Xint_effect = Xt - X0)

EV_Demand <- mutate(EV_Demand, MT = Max_Theory ,Xf = Xt)




####
```

```
return(list(EV_Demand=EV_Demand,matrix=matrix)) #This is how to output multiple data frames from the fc

}
EV_Demand_run1 <- hourly_demand(month = "Nov_18",intervention_comm_effect = 1, throttle_amount = -0.5, p
#EV_Demand_run1 <- hourly_demand(month = "Jul_18", int_equals_baseline = FALSE, intervention_chargers =
```

```
#Load Emissions Factors
Hourly_EF <- read_csv("Hrly_EF.csv")
```

```
## Parsed with column specification:
## cols(
##   HR = col_double(),
##   CO2Baseline = col_double(),
##   NOXBaseline = col_double(),
##   CO2Marginal = col_double(),
##   NOXMarginal = col_double()
## )
```

```
# Current hourly EF csv is fake. Numbers will be in lbs/kwh for both CO2 and NOX.

#Load default periods (intervention hour is set above); decide if moving these defaults above
pk <- c(17:21) #target window to shift out off (this is only used in the output calculations below, not

#Insert costs ($/kg)
NOXcost <- 21.93 #$/kg
Curtailmentcost <- 0.15 #$/MWH This number needs to be sourced and updated!

#emissions_x <- emission_output(run_x$EV_Demand)
# emissions_x<- emission_output(hourly_demand())
# Emissionfcn <- function (EV_Demand_run1, peak_hour = pk) {
# add a column to output of hourly demand that specifies intervention hours (binary 1: 0; intervention
#}

emissions_fcn <- function(EVDemand, peak_hours = pk) {
#Create Date Frame for Emissions Outputs
Emissions <- Hourly_EF %>%
  mutate(I01 = EVDemand$EV_Demand$I01) %>%
  mutate(Xi = EVDemand$EV_Demand$Xi) %>% # column for initial demand at each hour not scaled
  mutate(X0 = EVDemand$EV_Demand$X0) %>% # column for initial demand at each hour scaled by chargers
  mutate(Xf = EVDemand$EV_Demand$Xf)
     # column for new demand at each hour. we'll need to update "Xt" to reflect the new column post com

Emissions <- Emissions %>%
  mutate(Xdelta = Xf - X0) #add in delta (change) row. The function could spit this out in its final da

#Calculate emissions for baseline demand (scaled and not)
Emissions <- Emissions %>%
  mutate (CO2Xi = Xi*CO2Baseline) %>%
  mutate (NOXXi = Xi*NOXBaseline) %>%
  mutate (CO2X0 = X0*CO2Baseline) %>%
  mutate (NOXX0 = X0*NOXBaseline)

#calculate change in emissions
```

```
Emissions <- Emissions %>%
  mutate (CO2Xdelta = Xdelta*CO2Marginal) %>%
  mutate (NOXXdelta= Xdelta*NOXMarginal)

#calculate emissions for final
Emissions <- Emissions %>%
  mutate (CO2Xf = CO2X0+CO2Xdelta) %>% # baseline emissions - change for emissions associated with fina
  mutate (NOXXf = NOXX0+NOXXdelta)

#Set intervention hours, peak hours, other hours based on intervention hours selected for hourly demand
#Find interventions hours
intervention_emission_hours <- which(Emissions[,6] == 1) #pulls intervention hours from initial hourly
#Find other hours based on intervention hours set in hourly function and peak hour input
other_emission_hours <- c(1:24) #create 1:24 for all other hours
other_emission_hours <-other_emission_hours[!other_emission_hours %in% intervention_emission_hours] #re
other_emission_hours <-other_emission_hours[!other_emission_hours %in% peak_hours] #remove the pk hours

#Sum base emissions by period (intervention window, peak window, other)
 CO2X0_i_h <- sum(Emissions$CO2X0[intervention_emission_hours])
 CO2X0_pk <- sum(Emissions$CO2X0[peak_hours])
 CO2X0_o_h <- sum(Emissions$CO2X0[other_emission_hours])
 CO2X0_sum <- sum(Emissions$CO2X0)
 NOXX0_i_h <- sum(Emissions$NOXX0[intervention_emission_hours])
 NOXX0_pk <- sum(Emissions$NOXX0[peak_hours])
 NOXX0_o_h <- sum(Emissions$NOXX0[other_emission_hours])
 NOXX0_sum <- sum(Emissions$NOXX0)

#Sum final emissions by period (intervention window, peak window, other)
 CO2Xf_i_h <- sum(Emissions$CO2Xf[intervention_emission_hours]) #11-3 window
 CO2Xf_pk <- sum(Emissions$CO2Xf[peak_hours]) #peak 4pm-9pm window
 CO2Xf_o_h <- sum(Emissions$CO2Xf[other_emission_hours]) #all other hours
 CO2Xf_sum <- sum(Emissions$CO2Xf)
 NOXXf_i_h <- sum(Emissions$NOXXf[intervention_emission_hours]) #11-3 window
 NOXXf_pk <- sum(Emissions$NOXXf[peak_hours]) #peak 4pm-9pm window
 NOXXf_o_h <- sum(Emissions$NOXXf[other_emission_hours]) #all other hours
 NOXXf_sum <- sum(Emissions$NOXXf)

#Sum change in emissions by period (intervention window, peak window, other)
 CO2Xdelta_i_h <- sum(Emissions$CO2Xdelta[intervention_emission_hours]) #11-3 window
 CO2Xdelta_pk <- sum(Emissions$CO2Xdelta[peak_hours]) #peak 4pm-9pm window
 CO2Xdelta_o_h <- sum(Emissions$CO2Xdelta[other_emission_hours]) #all other hours
 CO2Xdelta_sum <- sum(Emissions$CO2Xdelta)
 NOXXdelta_i_h <- sum(Emissions$NOXXdelta[intervention_emission_hours]) #11-3 window
 NOXXdelta_pk <- sum(Emissions$NOXXdelta[peak_hours]) #peak 4pm-9pm window
 NOXXdelta_o_h <- sum(Emissions$NOXXdelta[other_emission_hours]) #all other hours
 NOXXdelta_sum <- sum(Emissions$NOXXdelta)

#Put into new df
Emissions_Table <- data.frame(Time= c("intervention hours", "peak period", "other", "Total"), CO2Initia
#Note: there's got to be a cleaner way to calculate these totals.

#Calculate cost reduction from NOX reduction (only doing it for the change)
Emissions_Table <- Emissions_Table %>%
```

```
   mutate(NOXChangeCost = NOXChange*NOXcost) #this is in dollars

#Curtailment Cost Reduction
#sum hours of demand change in target window
Xdelta_i_h <- sum(Emissions$Xdelta[intervention_emission_hours])
Curtailmentcost_i_h <- Xdelta_i_h*-Curtailmentcost
#add to output table
Emissions_Table <- Emissions_Table %>%
   mutate(ChangeCurtCost = c(Curtailmentcost_i_h, "NA", "NA", Curtailmentcost_i_h))

return(list(Emissions=Emissions, Emissions_Table=Emissions_Table))

}
#Emissions_run1 <- emissions_fcn(EVDemand = hourly_demand(SDGE_commm_effect = 0,intervention_comm_effec
```

```
simulation <- function(simulations = 12,
                       sim_segment = sg,
                       sim_month = mth,
                       sim_charger_power = pwr,
                       sim_schedule = sch,
                       sim_price_change = p_c,
                       sim_intervention_hours = i_h,
                       sim_intervention_chargers = int_ch,
                       sim_int_equals_baseline = int_e_b,
                       sim_throttle_amount = t_a,
                       sim_throttle_hours = t_h,
                       sim_intervention_comm_effect = i_c_e ) {

  intervention_draws <- sample(4:length(Elasticities),simulations,replace = FALSE)
  comm_possibilities <- seq(from=0.2, to=1, by=.01)
  comm_draws <-  sample(comm_possibilities,simulations, replace = TRUE)

  sim_result <- hourly_demand()
  sim_result_EV_Demand <- mutate(sim_result$EV_Demand, i_draw = intervention_draws[1], c_draw = comm_dra
  sim_result_EV_Demand <- sim_result_EV_Demand[0,]


  for(i in seq(simulations)){
    run_i <- hourly_demand(segment = sim_segment,
                           month = sim_month,
                           charger_power = sim_charger_power,
                           schedule = intervention_draws[i],
                           price_change = sim_price_change,
                           intervention_hours = sim_intervention_hours,
                           intervention_chargers = sim_intervention_chargers,
                           int_equals_baseline = sim_int_equals_baseline,
                           throttle_amount = sim_throttle_amount,
                           throttle_hours = sim_throttle_hours,
                           intervention_comm_effect = comm_draws[i])

    run_i_EV_Demand <- run_i$EV_Demand %>%
      mutate(i_draw = intervention_draws[i], c_draw = comm_draws[i])

    sim_result_EV_Demand <- rbind(sim_result_EV_Demand,run_i_EV_Demand)
```

```
  }

    summary <- group_by(sim_result_EV_Demand, Hr) %>%
    summarise_at(vars(Xf),funs(mean,min,max))# %>%


  sim_result_EV_Demand <- sim_result_EV_Demand %>%
    mutate(Hr_avg = summary$mean[Hr], min = summary$min[Hr], max = summary$max[Hr])


  return(sim_result_EV_Demand)
}


sim1 <- simulation(sim_intervention_comm_effect = 0)
```

```
#No Cross Elasticities
#needs to use No_Cross_Matrix for this to run correctly
no_cross_discount <- hourly_demand(segment = "Workplace",
                                   month = "Nov_18",
                                   price_change = -0.05,
                                   intervention_hours = c(12:15),
                                   throttle_amount = 0,
                                   throttle_hours = c(6:11))

no_cross_results <- read_csv("WP_Discount_No_Cross_Nov_18.csv")
```

```
## Parsed with column specification:
## cols(
##   Hr = col_double(),
##   P0 = col_double(),
##   Period = col_character(),
##   I01 = col_double(),
##   Xi = col_double(),
##   X0 = col_double(),
##   P1 = col_double(),
##   P1p = col_double(),
##   P1pC = col_double(),
##   X1p = col_double(),
##   X1 = col_double(),
##   Tp = col_double(),
##   Xt = col_double(),
##   Xint_effect = col_double(),
##   MT = col_double(),
##   Xf = col_double()
## )
```

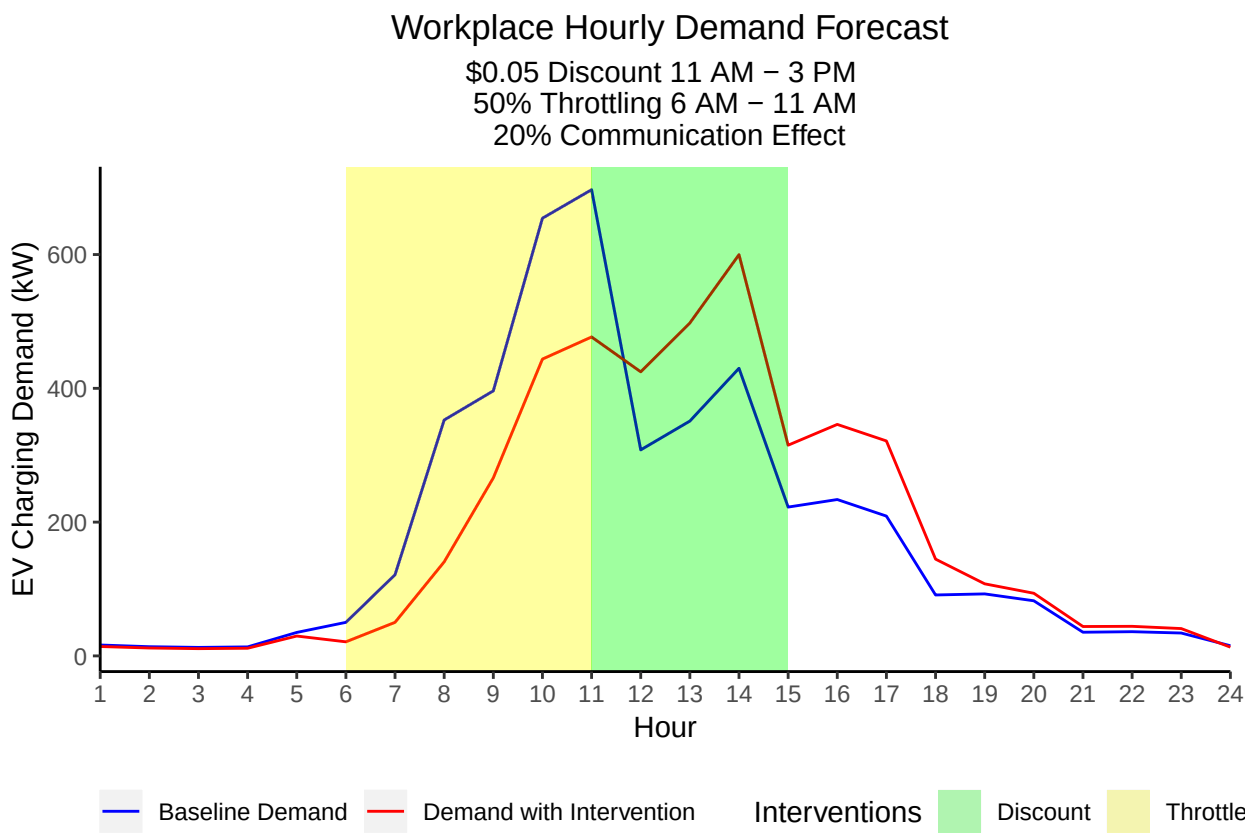
```
#-Workplaces: Communication + Discount on current TOU rate

WP_C_D <- hourly_demand(segment = "Workplace",
                                   month = "Nov_18",
                                   price_change = -0.05,
                                   intervention_hours = c(12:15),
```

```
                                        throttle_amount = 0)


#-Workplaces: Communication + Discount + Throttling on current TOU rate

WP_C_D_T_TOU_SUmmer_4 <- hourly_demand(segment = "Workplace",
                                        month = "Nov_18",
                                        price_change = -0.05,
                                        intervention_hours = c(12:15),
                                        throttle_amount = -0.5,
                                        throttle_hours = c(6:11))


#-Workplaces: TOU 2019
#Having issues w/ implementing new TOU

#TOU_summer_19 <- read_csv("Model_Map/2019_Summer_TOU_EV_8.csv")

#WP_TOU_19 <- hourly_demand(segment = "Workplace",
#                           month = "June_18",
#                           new_TOU = TOU_summer_19$P0,
#                           price_change = 0)

#-MUDs: rebate+ communication

MUD_R_C <- hourly_demand(segment = "Multi Unit Dwelling",
                         month = "June_18",
                         price_change = 0.1 ,
                         intervention_hours = c(17:21))

#-Destination centers: discount + throttling + no comms

DC_D_T_NC <- hourly_demand(segment = "Destination Center",
                           month = "June_18",
                           intervention_hours = c(12:15),
                           price_change = -0.05,
                           throttle_amount = -0.5, throttle_hours = c(6:11))

#-Fleets: rebate + communication

F_R_C <-  hourly_demand(segment = "Fleet",
                        month = "Aug_18",
                        intervention_hours = c(17:21),
                        price_change = 0.1)

#-Workplaces: Discount + throttling



WP_D_T <- hourly_demand(segment = "Workplace",
                        month = "June_18",
                        intervention_hours = c(12:15),
                        price_change = -0.05,
```

```
                        throttle_amount = -0.5,
                        throttle_hours = c(6:11))

WP_D_R_T <- hourly_demand(segment = "Workplace",
                        month = "June_18",
                        intervention_hours = c(12:15, 17:21),
                        price_change = c(rep(-0.05,4),rep(0.1,5)),
                        throttle_amount = -0.5,
                        throttle_hours = c(6:11))

#Scale baseline to July 11,2018 event
jul_11_18_event <- Event_Chargers %>%
  filter(Market_Segment=="DC") %>%
  select(Jul_11_18_LR) %>%
  unlist()

Jul_11_18_baseline_usage_scaled <- hourly_demand(price_change = 0,
                                                 throttle_amount = 0,
                                                 month = "Jul_18",
                                                 segment =  "Destination Center",
                                                 intervention_chargers =
                                                   jul_11_18_event,
                                                 int_equals_baseline = FALSE)

#Scale baseline to Nov 14, 2018 event
nov_14_18_event <- Event_Chargers %>%
  filter(Market_Segment=="Workplace") %>%
  select(Nov_14_18_LS) %>%
  unlist()

Nov_14_18_baseline_usage_scaled <- hourly_demand(price_change = 0, throttle_amount = 0, month = "Nov_18
```

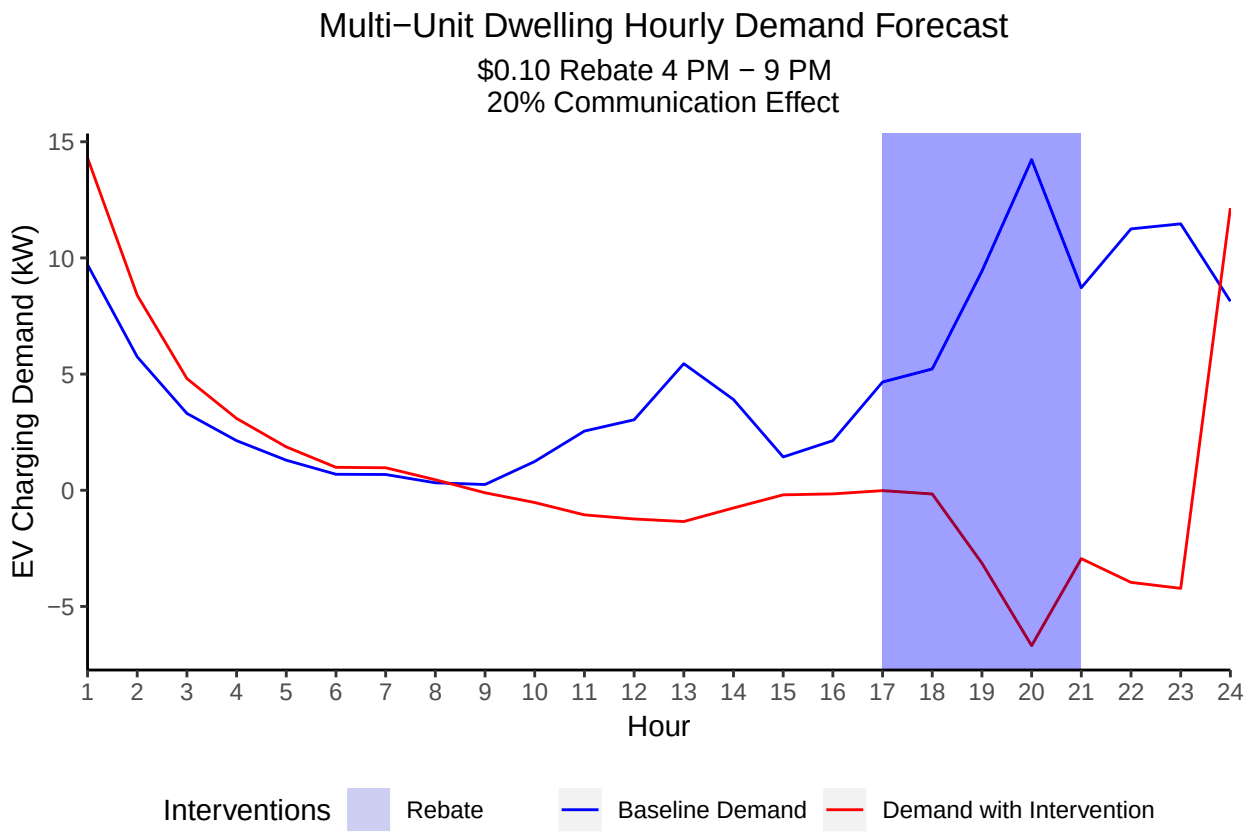
```
#Scenario 1
#-Workplaces: Communication + Discount + Throttling on current TOU rate


graph_table1 <- WP_C_D_T_TOU_SUmmer_4$EV_Demand[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=WP_C_D_T_TOU_SUmmer_4$EV_Demand$MT[1])


Demand_Graph1 <- ggplot(data = graph_table1, 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=" Workplace Hourly Demand Forecast",
       subtitle="$0.05 Discount 11 AM - 3 PM \n 50% Throttling 6 AM - 11 AM \n 20% Communication Effect
       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=6,xmax=11,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(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('green','yellow'),  guide = guide_legend(override.aes = l

Potential_Graph1 <- Demand_Graph1 +
  geom_line(aes(y=Theoretical_Max))

Demand_Graph1
```



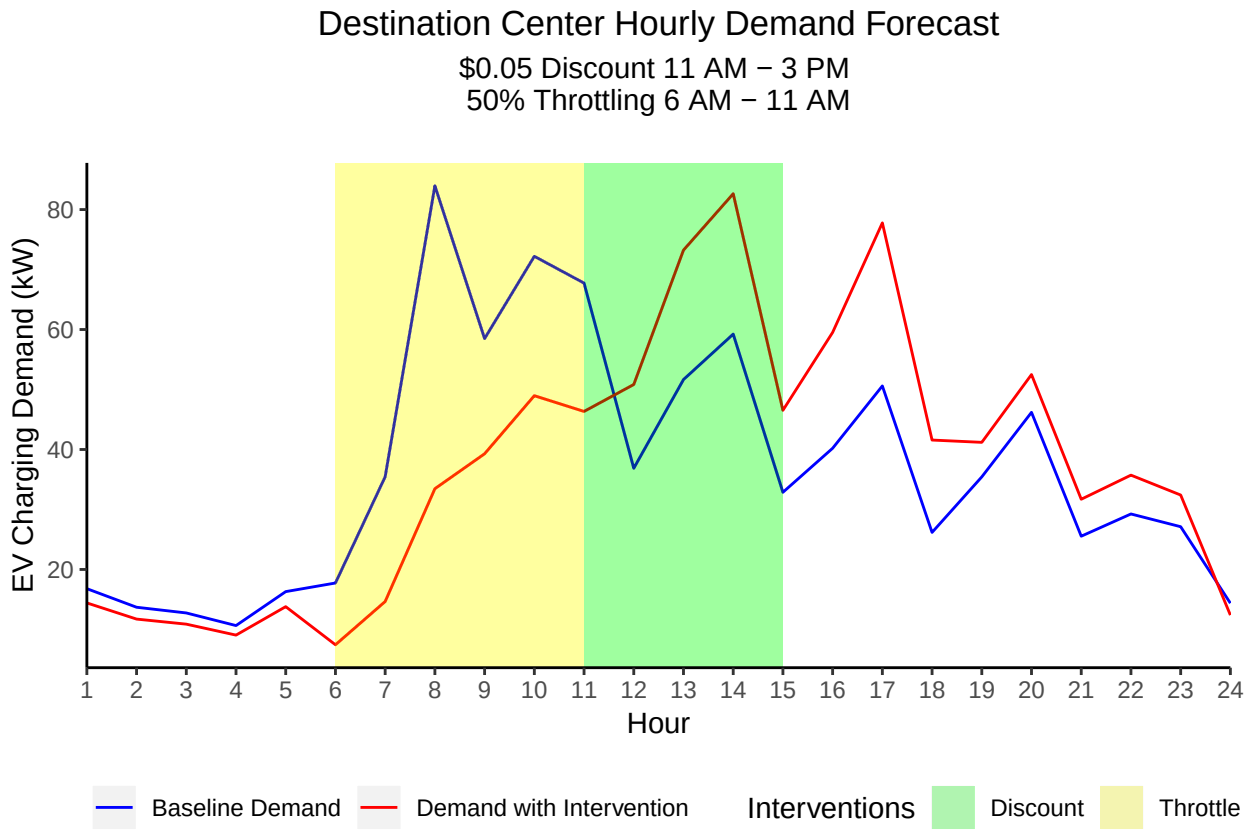

```
#Potential_Graph1

#Scenario 2 UNDER CONSTRUCTION
#-Workplaces: TOU 2019


#graph_table2 <- WP_TOU_19$EV_Demand[c("Hr","X0","Xf")] %>%
#  gather(condition,value,X0:Xf) %>%
#  mutate(Theoretical_Max=WP_TOU_19$EV_Demand$MT[1])
#
#
#Demand_Graph2 <- ggplot(data = graph_table2, 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=" Workplace Hourly Demand Forecast",
```

```
#       subtitle="2019 TOU Rates",
#       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"))
#  theme(plot.title = element_text(hjust = 0.5)) +
#  theme(plot.subtitle = element_text(hjust = 0.5))+
#  theme(legend.position="bottom")
#
#Potential_Graph2 <- Demand_Graph2 +
#  geom_line(aes(y=Theoretical_Max))
#
#Demand_Graph2
#Potential_Graph2

#Scenario 3
#-MUDs: rebate+ communication

graph_table3 <- MUD_R_C$EV_Demand[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=MUD_R_C$EV_Demand$MT[1])


Demand_Graph3 <- ggplot(data = graph_table3, 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=" Multi-Unit Dwelling Hourly Demand Forecast",
       subtitle="$0.10 Rebate 4 PM - 9 PM \n 20% Communication Effect",
       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=17,xmax=21,ymin=-Inf,ymax=Inf, fill = "Rebate"),alpha=0.0075) +
  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('blue'),  guide = guide_legend(override.aes = list(alpha

Potential_Graph3 <- Demand_Graph3 +
  geom_line(aes(y=Theoretical_Max))

Demand_Graph3
```



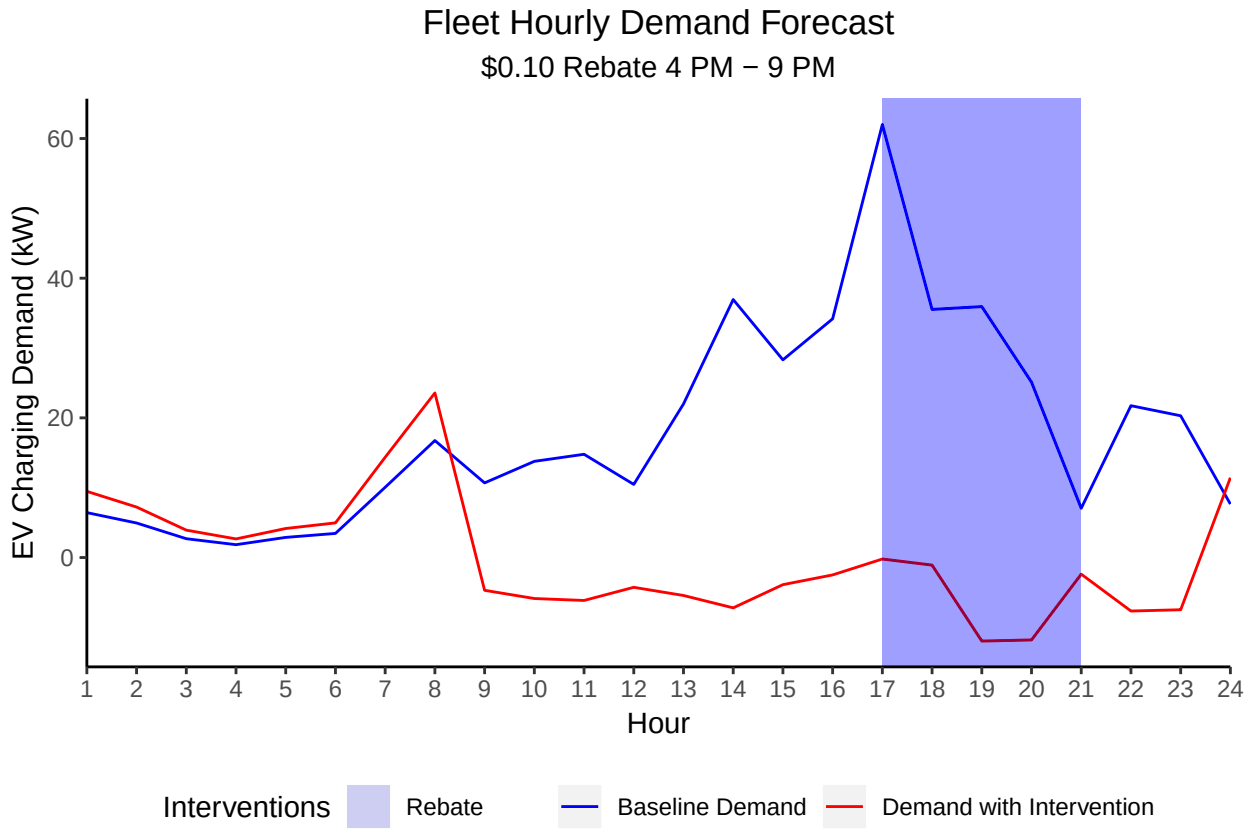

```
#Potential_Graph3


#Scenario 4
#-Destination centers: discount + throttling + no comms


graph_table4 <- DC_D_T_NC$EV_Demand[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=DC_D_T_NC$EV_Demand$MT[1])


Demand_Graph4 <- ggplot(data = graph_table4, 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="Destination Center Hourly Demand Forecast",
       subtitle="$0.05 Discount 11 AM - 3 PM \n 50% Throttling 6 AM - 11 AM \n",
       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=11,xmax=15,ymin=-Inf,ymax=Inf, fill = "Discount"),alpha=0.0075) +
  geom_rect(aes(xmin=6,xmax=11,ymin=-Inf,ymax=Inf, fill = "Throttle"),alpha=0.0075) +
  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('green','yellow'),  guide = guide_legend(override.aes = l

Potential_Graph4 <- Demand_Graph4 +
  geom_line(aes(y=Theoretical_Max))

Demand_Graph4
```



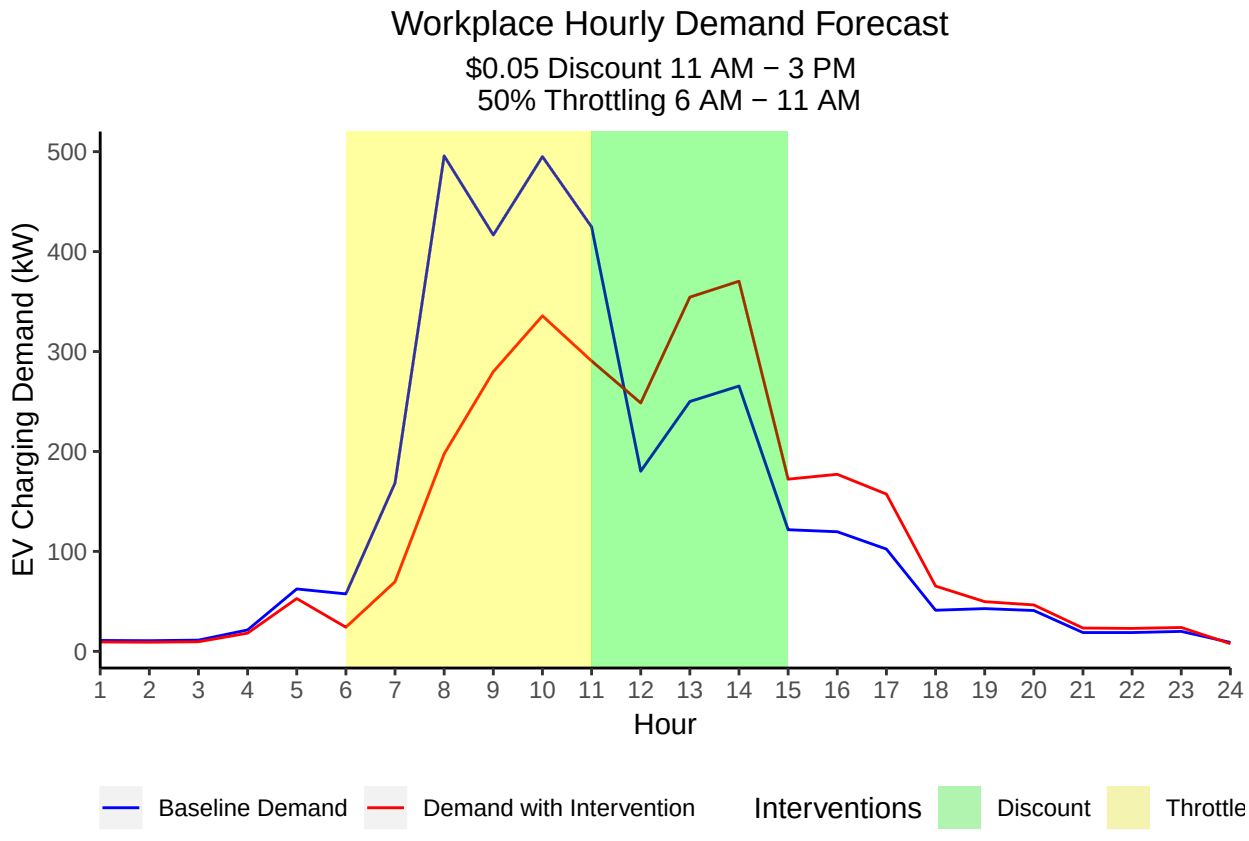

```
#Potential_Graph4


#Scenario 5
#-Fleets: rebate + communication

graph_table5 <- F_R_C$EV_Demand[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=F_R_C$EV_Demand$MT[1])


Demand_Graph5 <- ggplot(data = graph_table5, 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="Fleet Hourly Demand Forecast",
       subtitle="$0.10 Rebate 4 PM - 9 PM",
```

```
       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=17,xmax=21,ymin=-Inf,ymax=Inf, fill = "Rebate"),alpha=0.0075) +
  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('blue'),  guide = guide_legend(override.aes = list(alpha

Potential_Graph5 <- Demand_Graph5 +
  geom_line(aes(y=Theoretical_Max))

Demand_Graph5
```




```
#Potential_Graph5

#Scenario 6
#-Workplaces: Discount + throttling

graph_table6 <- WP_D_T$EV_Demand[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=WP_D_T$EV_Demand$MT[1])
```

```
Demand_Graph6 <- ggplot(data = graph_table6, 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=" Workplace Hourly Demand Forecast",
       subtitle="$0.05 Discount 11 AM - 3 PM \n 50% Throttling 6 AM - 11 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=6,xmax=11,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(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('green','yellow'),  guide = guide_legend(override.aes = l

Potential_Graph6 <- Demand_Graph6 +
  geom_line(aes(y=Theoretical_Max))

Demand_Graph6
```

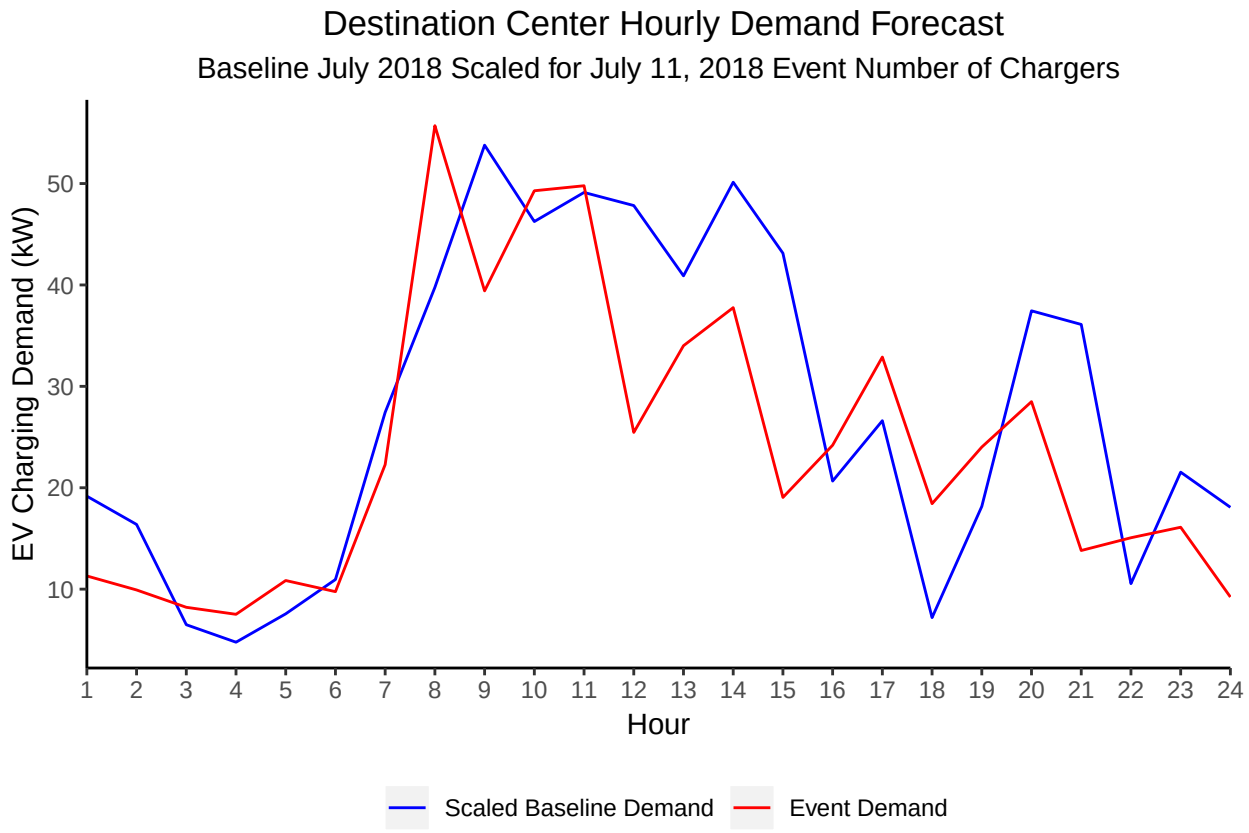



```
#Potential_Graph6
```

```r
#Scenario 7
#Scale baseline to July 11,2018 event

July_11_18_Event_Usage <- DC_Event_Total_Usage$Jul_11_18_LR
Jul_11_18_baseline_usage_scaled$EV_Demand <- Jul_11_18_baseline_usage_scaled$EV_Demand %>%
  mutate(Event_Usage = July_11_18_Event_Usage)

graph_table7 <- Jul_11_18_baseline_usage_scaled$EV_Demand[c("Hr","Xf","Event_Usage")] %>%
  gather(condition,value,Xf:Event_Usage) %>%
  mutate(Theoretical_Max=Jul_11_18_baseline_usage_scaled$EV_Demand$MT[1])
```

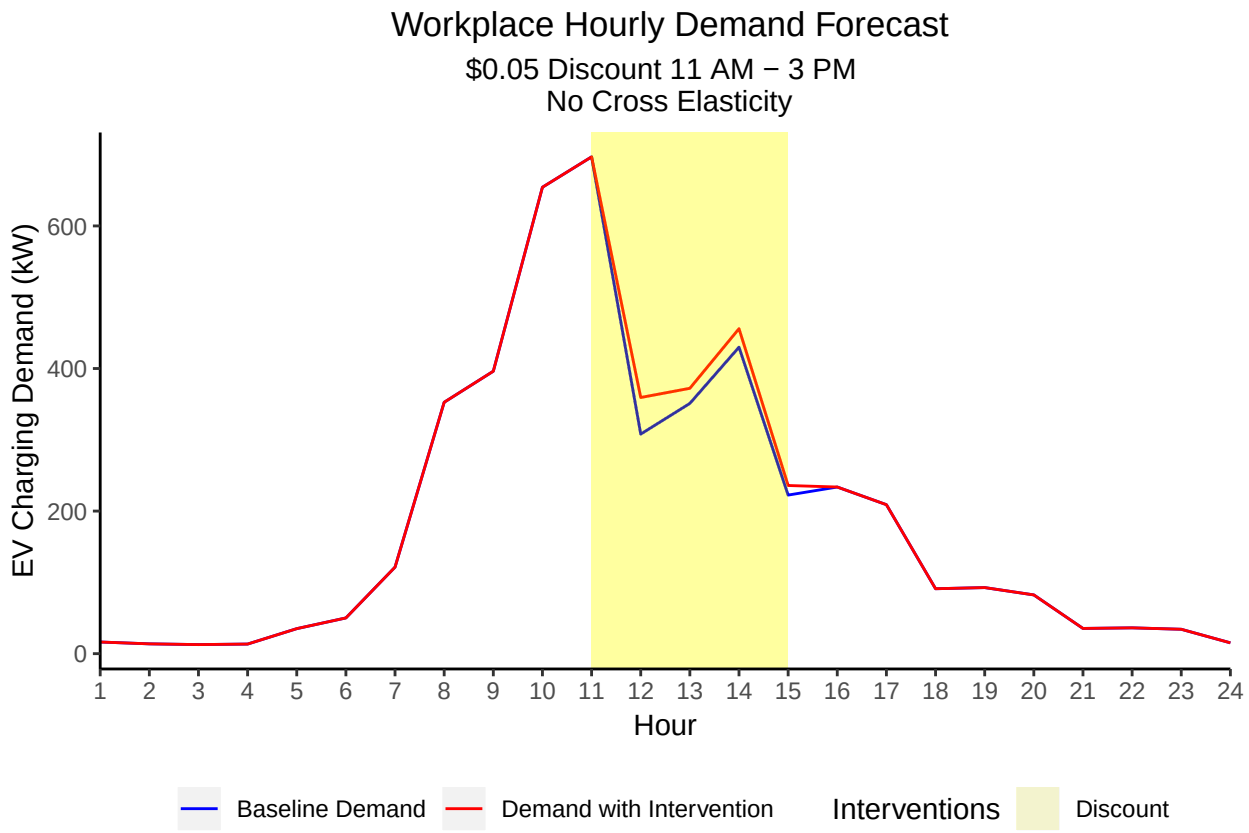
```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```r
Demand_Graph7 <- ggplot(data = graph_table7, 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=" Destination Center Hourly Demand Forecast",
       subtitle="Baseline July 2018 Scaled for July 11, 2018 Event Number of Chargers",
       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("Scaled Baseline Demand", "Event Demand"), values = c("blue", "red")) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))+
 theme(legend.position="bottom")

Potential_Graph7 <- Demand_Graph7 +
  geom_line(aes(y=Theoretical_Max))

Demand_Graph7
```



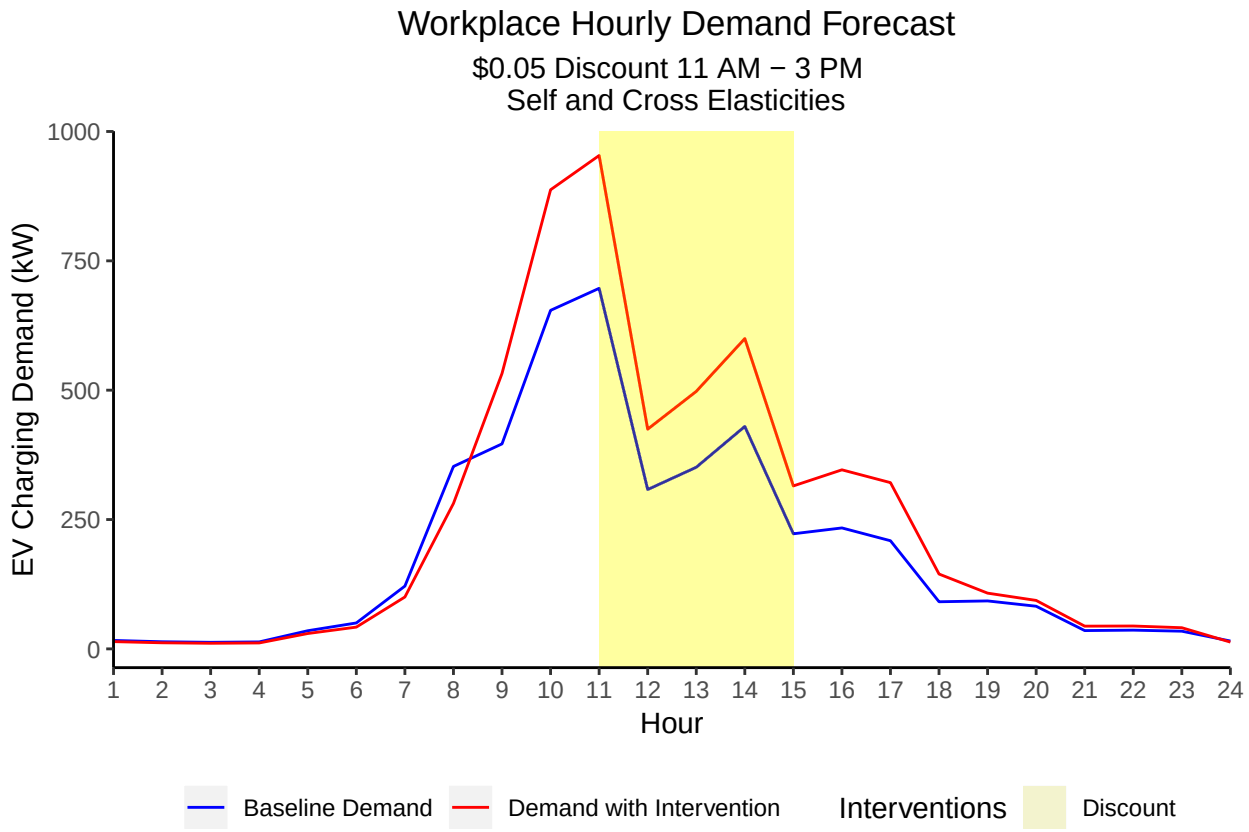

```
###Scenario 8

graph_table8 <- no_cross_results[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=no_cross_results$MT[1])


Demand_Graph8 <- ggplot(data = graph_table8, 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=" Workplace Hourly Demand Forecast",
       subtitle="$0.05 Discount 11 AM - 3 PM \n No Cross Elasticity",
       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=11,xmax=15,ymin=-Inf,ymax=Inf, fill = "Discount"),alpha=0.0075) +
  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(alp

Potential_Graph8 <- Demand_Graph8 +
  geom_line(aes(y=Theoretical_Max))
```

```
Demand_Graph8
```



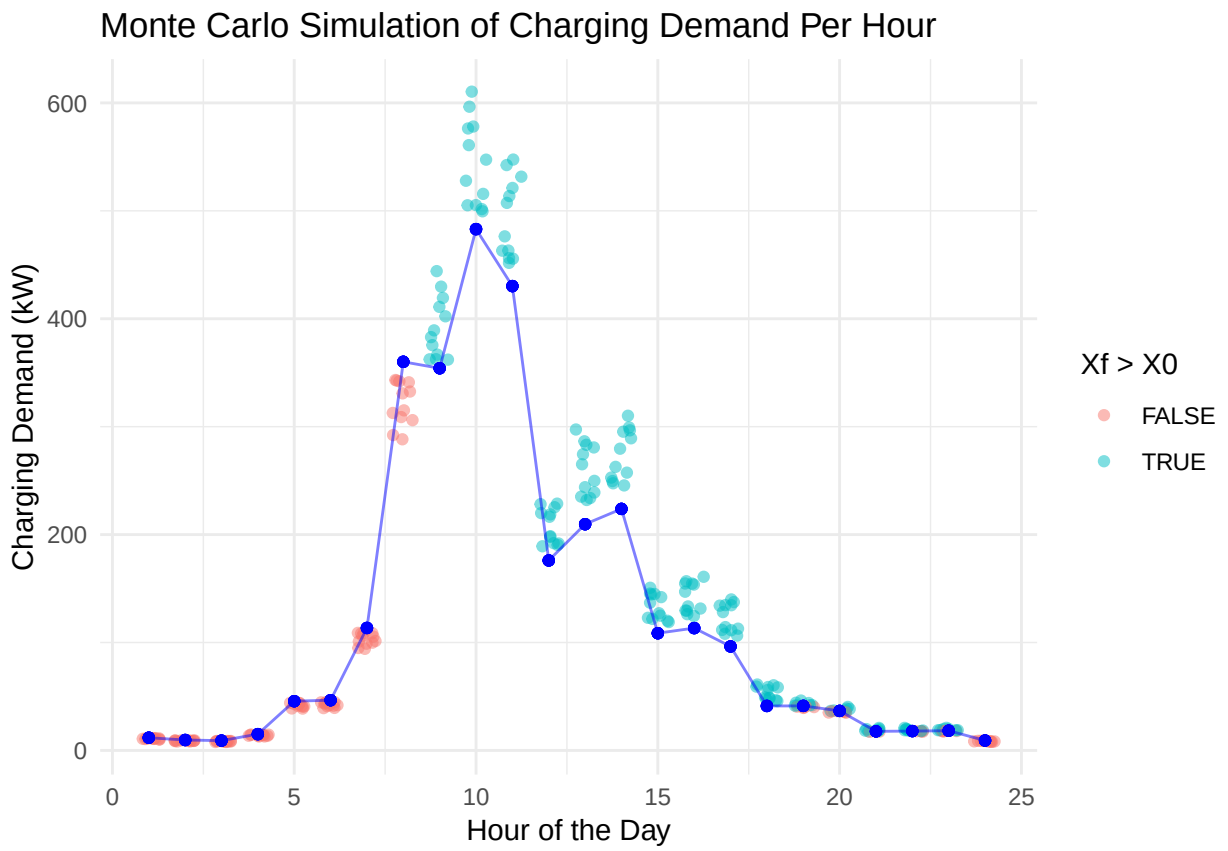

```
#Workplace discount only self and cross

graph_table9 <- WP_C_D$EV_Demand[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=WP_C_D$EV_Demand$MT[1])


Demand_Graph9 <- ggplot(data = graph_table9, 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=" Workplace Hourly Demand Forecast",
       subtitle="$0.05 Discount 11 AM - 3 PM \n Self and Cross Elasticities",
       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=11,xmax=15,ymin=-Inf,ymax=Inf, fill = "Discount"),alpha=0.0075) +
  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(alp
```

```
Potential_Graph9 <- Demand_Graph9 +
  geom_line(aes(y=Theoretical_Max))

Demand_Graph9
```



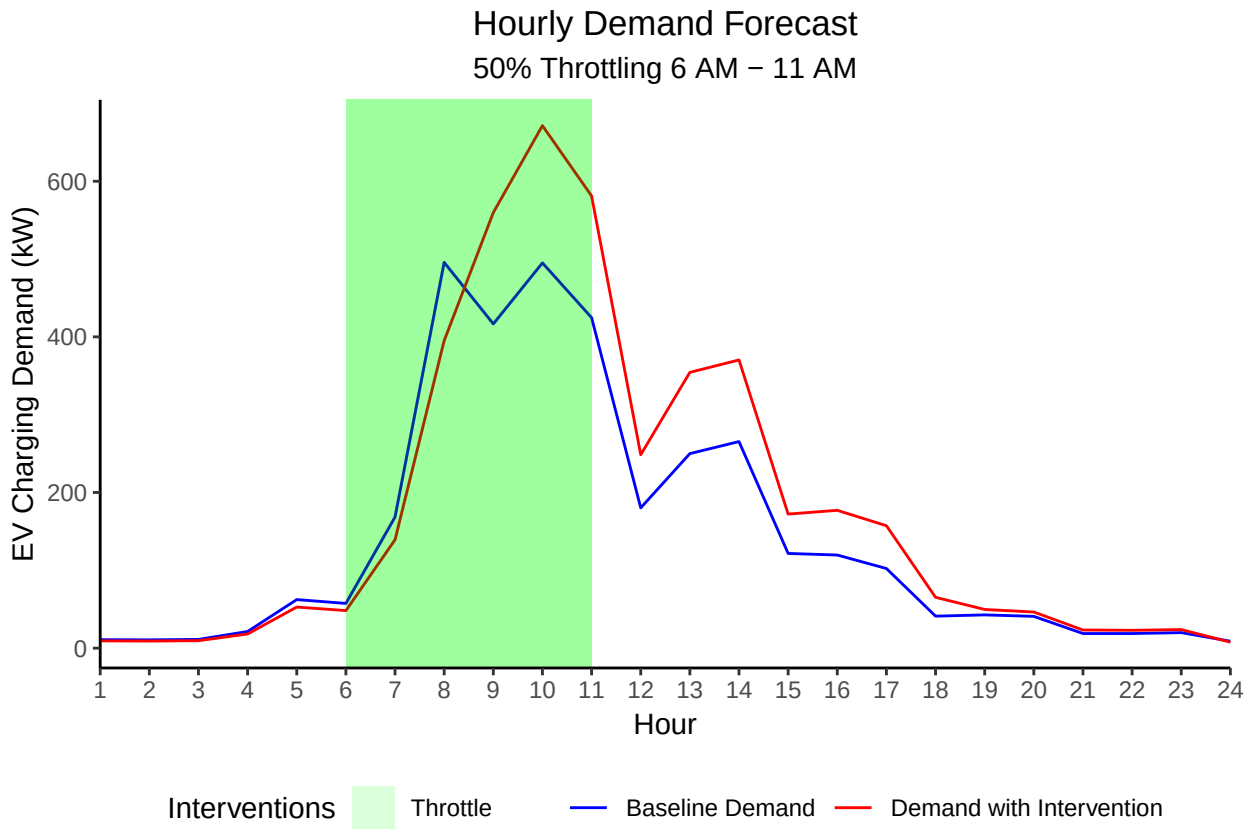

```
#Potential_Graph1
```

```
# plot of monte carlo output
library(ggplot2)
freq_graph <- ggplot(sim1, aes(x = Hr, y = X0))+
  geom_jitter(aes(y = Xf, color = Xf>X0), alpha = 0.5, width = 0.3) +
  geom_line(aes(x = Hr, y = X0), alpha = 0.5, color = "blue")+
  geom_point(aes(x = Hr, y = X0), alpha = 0.5, color = "blue")+
  #geom_ribbon(aes(ymin = min, ymax = max), alpha = 0.5, color ="blue") +
  #geom_line(aes(x = Hr, y = Hr_avg), alpha = 0.5, color = "red")+
  #geom_smooth(model = "lm") +
  theme_minimal() +
  xlab("Hour of the Day") +
  ylab("Charging Demand (kW)") +
  ggtitle("Monte Carlo Simulation of Charging Demand Per Hour")

freq_graph
```

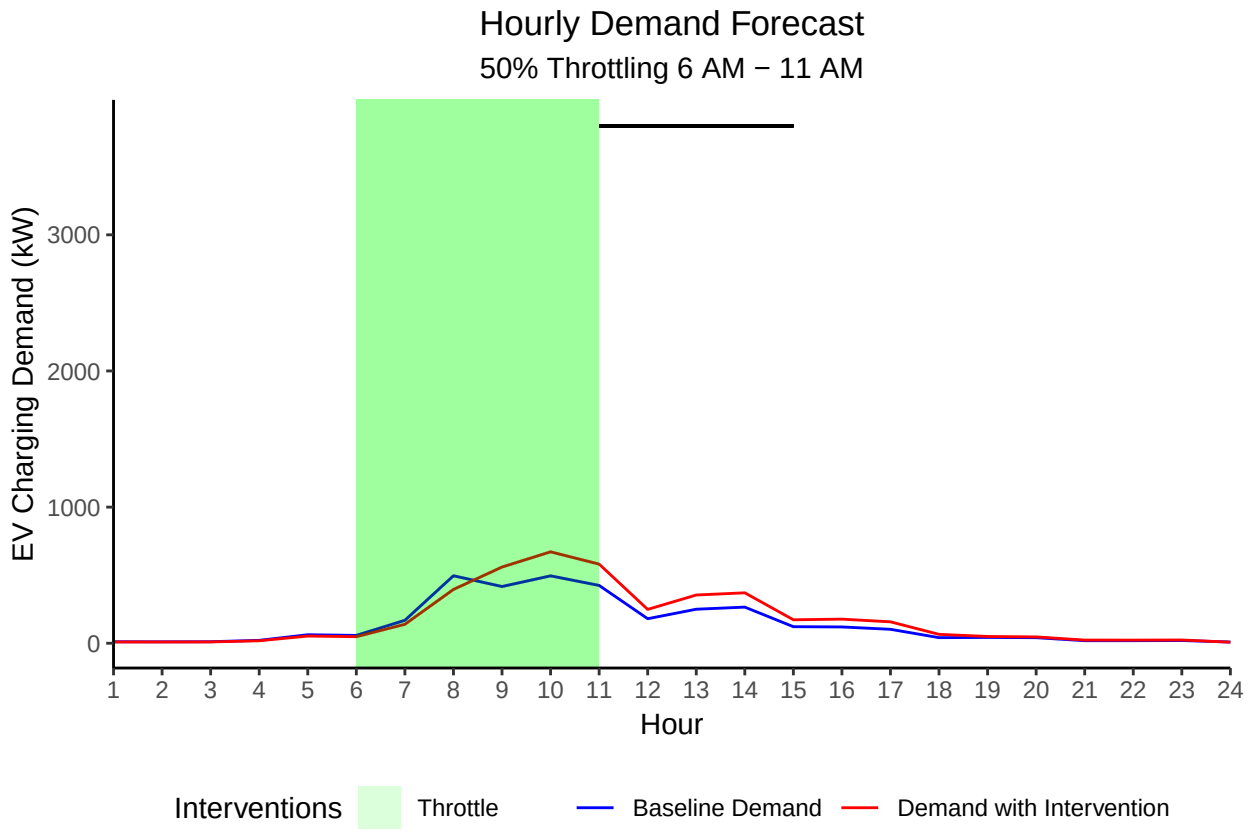



```
#need to stack both demand curves in one data frame for a legend
EV_Demand_run1 <- hourly_demand(segment = "Workplace", month = "June_18",intervention_comm_effect = 1, 

graph_table <- EV_Demand_run1$EV_Demand[c("Hr","X0","Xf")] %>%
  gather(condition,value,X0:Xf) %>%
  mutate(Theoretical_Max=EV_Demand_run1$EV_Demand$MT[1])


Demand_Graph <- ggplot(data = graph_table, aes(x = Hr)) +
  geom_line(aes(y = value, color=condition)) +
  theme_classic() +
  labs(title="Hourly Demand Forecast",
       subtitle="50% Throttling 6 AM - 11 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=6,xmax=11,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(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('green','yellow'),  guide = guide_legend(override.aes = 

Potential_Graph <- Demand_Graph +
```

```
  geom_segment(aes(y=Theoretical_Max, yend=Theoretical_Max, x=11, xend=15))

# \n 50% Throttling 6 AM - 11 AM label for throttling
# Aesthetic Notes:
# might try to remove space b/w Y-axis and hr0, as well as past hr24
# bold axis titles
# color brewer
# geom_line(aes(y=Theoretical_Max, x=11:15)) +

# If we want to graph the Max Theoretical Segment, we can use this:
# geom_segment(aes(y=Theoretical_Max, yend=Theoretical_Max, x=11, xend=15))


Demand_Graph
```




```
Potential_Graph
```




```
Total_max_theory <- pwr*Chargers$Nov_18[5]
Workplace_max_theory <- pwr*Chargers$Nov_18[4]
MUD_max_theory <- pwr*Chargers$Nov_18[3]
Fleet_max_theory <- pwr*Chargers$Nov_18[2]
DC_max_theory <- pwr*Chargers$Nov_18[1]

Max_theory_table <- Chargers %>%
  select(Market_Segment, Nov_18) %>%
  mutate(Theoretical_Max = pwr*Nov_18)

Max_theory_table #This shows the theoretical max per market segment (and total) for the SCE Charge Read
```

```
## # A tibble: 5 x 3
##   Market_Segment      Nov_18 Theoretical_Max
##   <chr>                <dbl>           <dbl>
## 1 Destination Center     234           1544.
## 2 Fleet                   83            548.
## 3 Multi Unit Dwelling     35            231
## 4 Workplace              596           3934.
## 5 Total                  948           6257.
```