University of Connecticut

Advanced BAPM: Sec B-13

****

**LAZ Parking, Boston**

**Team 4**

**Under the guidance of:**

**Prof. John Wilson**

**&**

**Prof. Mark Tschiegg**

By:

Alok Singh

Avinav Neer

Danish Thottathil

Vinod Nakkala

Ruthwick Kuchibhotla

**“The work presented here is our work and our work alone”**

Contents

[1. Executive Summary 3](#_Toc468738322)

[2. Background 4](#_Toc468738323)

[3. Business Problem 4](#_Toc468738324)

[4. Methodology 5](#_Toc468738325)

[5. Exploratory Data Analysis 6](#_Toc468738326)

[6. The Solution- Price Dashboard 13](#_Toc468738327)

[6.1 Concepts Involved: 14](#_Toc468738328)

[6.2 Structure: 15](#_Toc468738329)

[6.3 Test Case: 16](#_Toc468738330)

[7. Recommendations 17](#_Toc468738331)

[8. Conclusion 21](#_Toc468738332)

[9. References 21](#_Toc468738333)

[10. Appendix 22](#_Toc468738334)

# Executive Summary

LAZ Parking wants to use the parking data from its Central Business District parking lot in Boston, to better serve customers and improve profitability. To achieve this objective, they collaborated with UConn and provided the data generated in the year 2015-2016 related to Occupancy, Parking Transactions, Validation, Credit Card payments etc. for analysis.

We identified that most customers park for a duration more than 80 minutes on weekdays- intuitively, office goers. We then developed a single window solution in the form of an R-Shiny dashboard, employing a price elasticity model, to give LAZ control on analyzing the customer attrition in the view of a price increase or gauging the increase in number of customers in the event of a price cut or discount. In order to put this to the best use we would suggest LAZ to use historical calculations of customer sensitivity- either from the Boston lot or from a city with similar demographics and customer patterns- like New York- where such studies have been conducted. Apart from suggesting the use of this interactive tool, we would also recommend some actions which could help LAZ explore unchartered avenues to draw more customers over a longer period.

# Background

LAZ Parking was founded in mid-1970’s on the foundations of a childhood friendship between three friends. It advanced from being a valet parking business to one of the highly-trusted parking garages in the country with a simple motto of “Create opportunities for our employees and value for our clients.” It provides a diverse set of parking management services for office buildings, hospitality and valet, healthcare services, government and municipal, campus, residential buildings, airport and transportation etc. Their solutions include managing a parking facility, leasing and ownership. They are spread across 26 states with an annual managed revenue of $1.2 billion. Some of their valuable clients are the First Baptist Church of Austin, The Beverly Hilton, Yale University, XL Center, University of Connecticut, Ohio State University, Chase Enterprises and many more.

# Business Problem

The decisions taken by LAZ on the price changes depend mostly on the market conditions and operating revenues of the company. They are not currently sure about what the customers are looking for in a typical parking. The decisions might vary based on geographical location, seasonality and ease of access- something which if understood can in-turn help them reach their customers and ultimately maximize the revenue. LAZ wanted a data driven solution for analyzing these customer parking trends and fix the prices based on not only the market conditions, but also on constant inflow of data.

# Methodology

The analysis was done through the following steps:

1. The Parking Transaction dataset (Parking\_Txns) and Payments dataset (PPA\_Payments) were joined using Payment Number as the unique ID. This gave the complete detailed information about the entry and exit times as well as the charges levied on individual customers at the same time. Using the parking date the records were labelled on the basis of the day of the week (as weekday or weekend). Further, on the basis of entry time the records were further classified as belonging to the weeknight slot if entry was after 4 pm. The parking rates in effect presently- as mentioned on the LAZ Boston website- were used to classify the records. The duration of stay in the lot was calculated using entry and exit times, using which the records were segmented in different time bands as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Day of Week** | **Duration of Stay** | **Rate** | **Time band** |
| Weekday | 0 - 20 minutes | $9 | 0-20 mins |
| 20 - 40 minutes | $19 | 20-40 mins |
| 40 - 60 minutes | $29 | 40-60 mins |
| 60 - 80 minutes | $36 | 60-80 mins |
| Over 80 minutes | $41 | >80 mins |
| Between 4 PM-6 AM | $13 | Weeknight |
| Weekend | 6 AM - 6 AM | $11 | Weekend |

The open source scripting language R was used for writing the code to bring about these modifications. The code has been provided in the Appendix.

1. The trends and distribution of customers in different time bands were analyzed using visualization techniques in R and Tableau. K-means clustering technique in SAS JMP was used to further segment the customers in increasingly profitable clusters and thus clustered records were analyzed for underlying insights.
2. A price elacticity model was developed using the open source R Shiny package. The interactive tool would help run simulations to determine the effect of price changes on revenues due to a shrink or increase in customer base. Factors affecting response to price changes by customers were understood through secondary research via academic papers on this topic.
3. Based on uncovered trends and consideration of steps taken by other parking providers in Boston and eslewhere some recommednations have been made to help LAZ attract more customers over a long time.

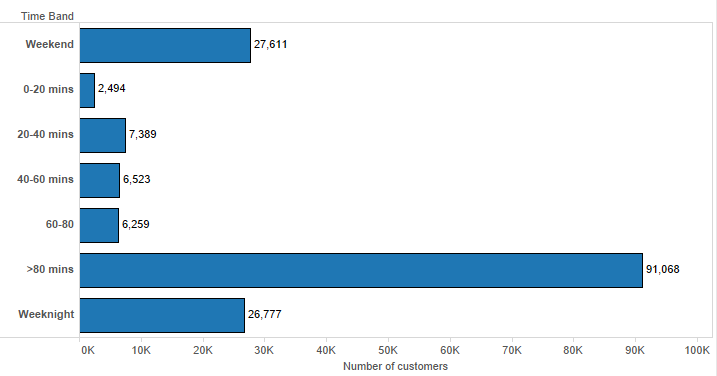
# Exploratory Data Analysis

Based on the time bands in the data manipulation phase, parking trends across these time bands were analyzed.

**Number of customers in each Time band:**

*Findings*:

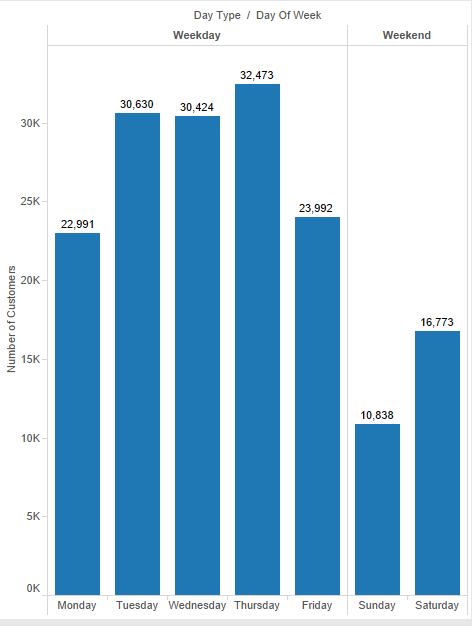
* Majority of the customers park for more than 80 minutes falling in the highest price band of $ 41 followed by discounted parking in the weekend and weeknight
* Parking time between 20 and 80 minutes have even distribution of customers



**Number of customers per day:**

*Findings*:

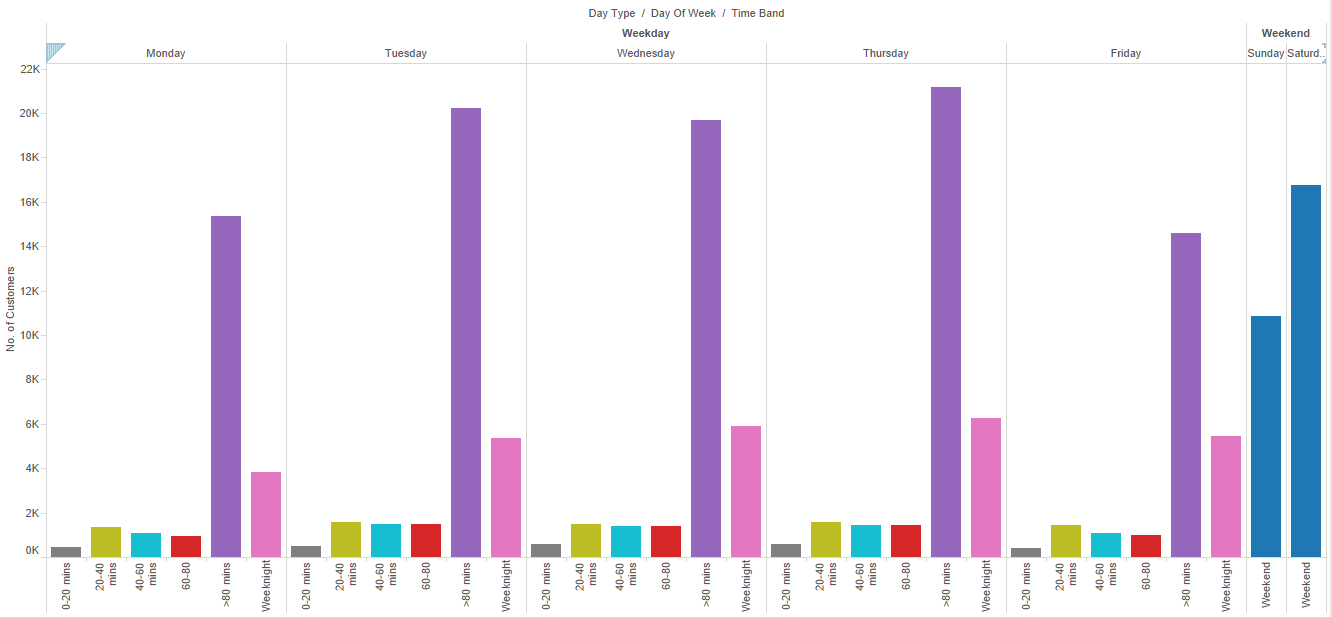
* Parking trend during mid days of the weeks (Tuesday, Wednesday, Thursday) are evenly distributed
* Lower demand on Mondays and Fridays compared to other days may be due to extended holiday weekends



**Number of customers per day across Time Bands:**

*Findings*:

* Similar to overall trend, most customers tend to park for more than 80 minutes during weekdays followed by weeknight parking
* During weekdays on an average only 20% of the customers park for less than 80 minutes

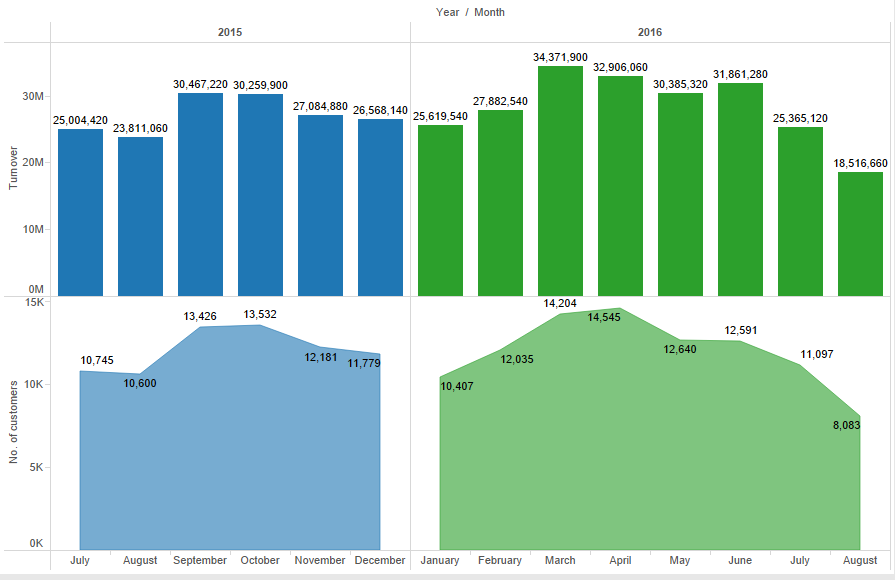


**Number of customers/ Turnover per month:**

*Findings*:

* Parking turnover across months are consistent from July 2015 to August 2016 with spikes March and April 2016
* Significant dip in the number of customers in August 2016 is due to incomplete data for that month.

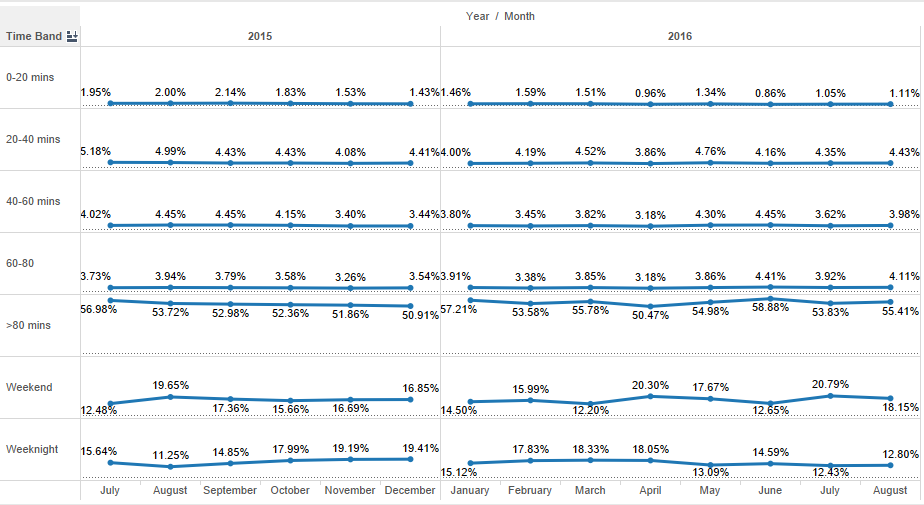
The trend of number of customers and the turnover (as in the original data, in cents) is plotted below:



**Proportion of customers across Time band per month:**

*Findings*:

* Proportion of customers who park for less than 80 minutes during weekdays follows a consistent trend and constitutes ~14% of the overall customers
* Weekday customers who park for more than 80 minutes constitutes more than 50% of the overall and has remained fairly consistent over last year
* Major fluctuations in proportion is found among weekend customers. This may be due to surge in demand due to events



**Most Profitable Customer Clusters**

The credit card transaction details were joined with Payments data using Payment number. Further, unsupervised learning method called K-Means clustering was used to group the data based on underlying similarities. The clusters were then arranged on the basis of increasing average turnovers. The most profitable cluster was found to comprise mostly of weekday customers parking for more than 80 minutes. The most profitable clusters also comprised a large number of weeknight and weekend customers apart from the “>80 mins” weekday customers. The time band distribution of the top three clusters is as shown below:

The third top cluster, however, contained largely of weekend and weeknight customers.

The credit card distribution of customers lying in this cluster was as follows:

While other networks show a distribution which is fairly similar to their total market share in the US, Discover shows a very low percentage (1%) when compared to its total US market share (9%).**[1]** This may be due to the underlying character of the Discover network which makes it less appealing to the bulk of customers of LAZ- office goers.

# The Solution- Price Dashboard

R Shiny provides a novel approach to develop interactive user interfaces (UIs) to carry out sophisticated data analysis using a free to use and open source framework. It has a very active community which takes upon itself to continuously improve the features and provide support and help to learners. Using Shiny web based applications can be developed without HTML, CSS or Java Script expertise.

As part of our solution, an interactive dashboard has been developed to give a one stop solution to LAZ Parking for conducting future analysis on changes in parking rates and their effects. The price changes and increase or decrease in turnover due to them could be simulated.

All the visual features and analytic calculations have been exhaustively coded in R Shiny, along with other packages from R (data.table, dplyr etc.). The complete code has been provided in the Appendix and the different aspects have been explained in the following pages.

## 6.1 Concepts Involved:

Price Elasticity:

It is a measure of responsiveness of the quantity of a raw good or service demanded to changes in its price. It could be measured as the change in demand of a quantity or service with a change in its price. An important factor involved in the price elasticity modelling is sensitivity.

Sensitivity:

Sensitivity indicates how sensitive customers are to a change in price of a commodity or service. Generally sensitivity is always negative. For example with increase in price of parking ticket, the customer base would decrease. But what is important here is to manage that trade-off between price and the number of customers.

Ways to calculate sensitivity:

1. Best way to calculate the sensitivity would have been to measure the response of ticket price change in the past data. This couldn’t be done with the data provided. It was also a project risk indicated earlier in our project charter.
2. Pull out sensitivity of previous price changes from data of a parking garage with similar customer demographics and distribution like that of the CBD, Boston garage.
3. To perform simulation for best and worst case scenarios.

Shiny Application:

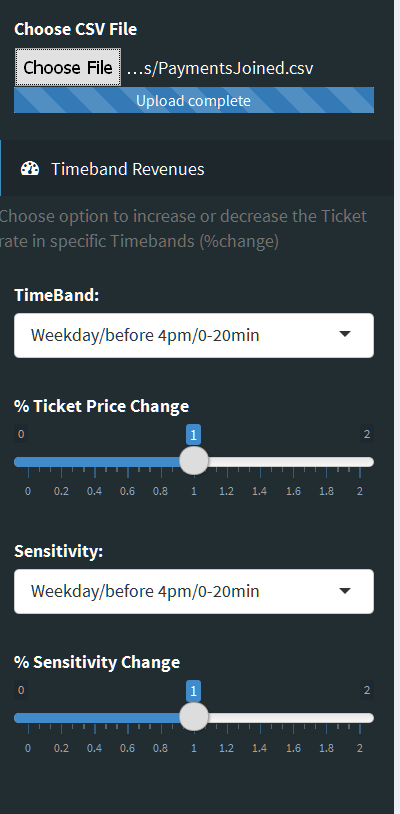
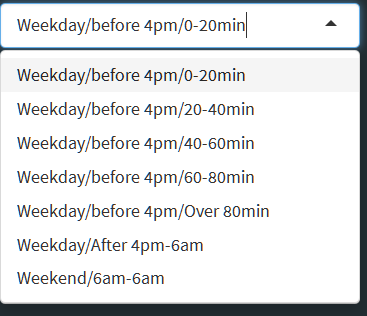
Shiny was used for building an application which was not only interactive in nature but could be scaled in future for new cases.

## Structure:

The application is divided into two functional parts- the sidebar and the main body.

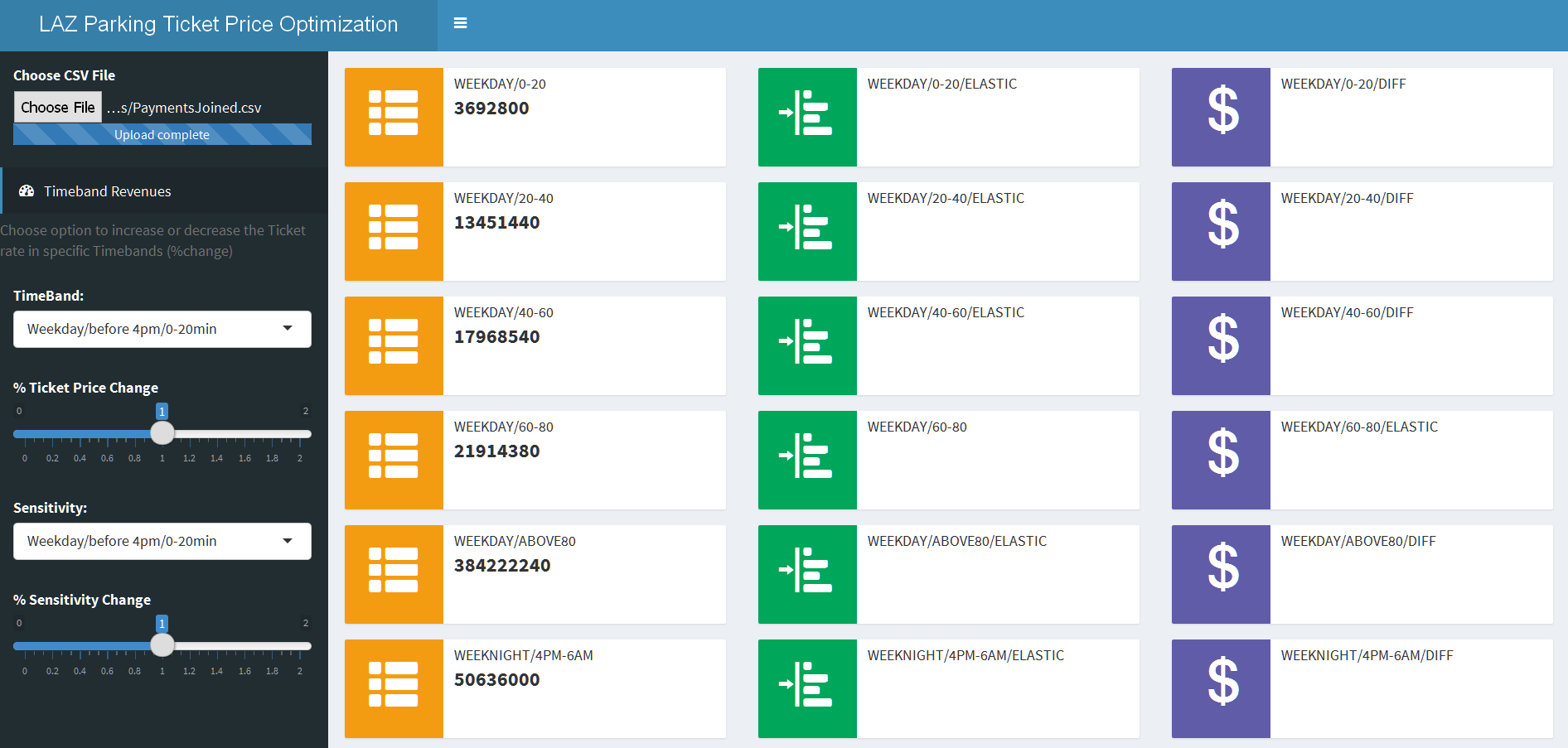
Sidebar:

* Sidebar contains the 3 tabs. One is to choose the CSV file. Now this file could be of weekly, monthly or yearly recorded parking iterations depending on the user case.
* Second is to select the time band in which the user feels that increase in price can get better revenue results.
* Third is to select the corresponding sensitivity for the time band. This feature is optional in case the user wants to run the simulations.
* There are 2 sliders added to the sidebar to help the user to perform percentage change in price of particular time bands and sensitivity.



Main Body:

* Main Body consists of 3 columns and 8 Rows of info boxes. The first column info boxes which are yellow in color give the total of a particular time band as listed on the info box.
* The second info box which is green in color calculates the total revenue when there is a change in the ticket price corresponding to a time band.
* The last info box which is purple in color gives the difference between the yellow and the green info boxes and depicts where a price change is making profit or loss for a price change.



## Test Case:

A Test case has been presented to better illustrate how the application works. Running these simulations with actual sensitivity and revenue values could help LAZ decide the optimum price for a time band.

In the first row the change in price and sensitivity is zero that means there is no change in the elastic revenue given in green info box. Hence the difference is zero represented by the purple info box.

Similarly for the next rows the effect of 10 % price change with 0 sensitivity change and a 20% price change with a 10% sensitivity can be seen. The simulation depicts the impact on revenue in a scenario where if you increase the price by 20% the customer base shrinks by 10%. The motive of this application is to make LAZ parking have full control over their price changes.



# Recommendations

Based on the analysis of the data and secondary research about the current practices by other parking services, we would like to make the following recommendations.

1. **Using the Shiny Dashboard:**

LAZ could use past price change data and the effect which they caused on the customer footfall to determine a decent approximation of the sensitivity parameter. This sensitivity parameter could be then used to analyze how a change of price in a particular time band would affect the customers. If the customer response can be better estimated in advance, it would give LAZ further scope to balance the tradeoff between increased charges and lesser customers by giving tailor made incentives to the group of customers it wants to target- for example office goers.

1. **Credit card promotions:**

As an extension of the use of the dashboard, we would like to suggest giving special promotional discounts to customers having credit cards of the Discover network. As discussed earlier, the disproportionally low number of Discover customers could be due to some inherent patterns. Our analysis has revealed that the weeknight and weekend time slots are the second most profitable after the office hours weekday slots. A discount to Discover card holders for weeknights and weekends would help by attracting new customers who would have come specially for some cultural or sporting events in the vicinity or vacations, while ensuring that the increased inflow does not disturb the occupancy of the lot during peak hours thus posing a hassle to the most important customers- the office goers

1. **Customer Profiles:**

LAZ could benefit by maintaining profiles of repeat customers- those using validation cards or making booking through the app on their phones or even parking multiple times using their credit card for payments. This would help in tracking their behavior over a long time in this lot as well as other LAZ lots in Boston and elsewhere. This could help in effective loyalty benefits and predicting usage patterns by using aggregated data and using advanced predictive modeling. An important consideration here is the security of the customer data, especially their credit card details.

1. **Real-time Parking Status:**

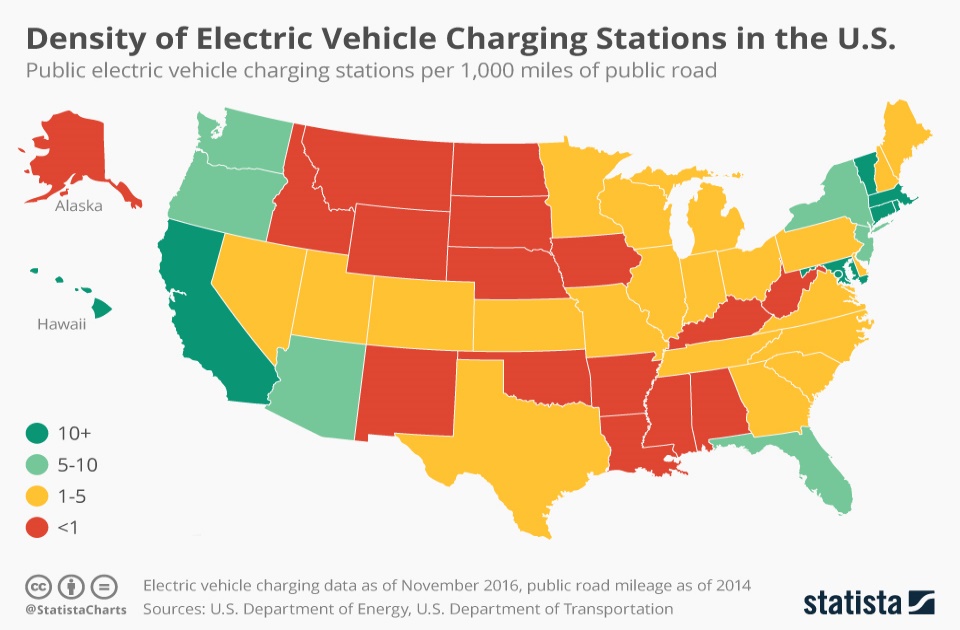
Upon analyzing the Occupancy data, a large number of erroneous values were observed. These could be attributed to the manual process of counting available parking spots which are difficult to update instantly and error prone. Actual real-time occupancy could help both LAZ and customers. This can be implemented using close circuit cameras and using big-data technologies to run image recognition on the live stream of data using tools like Apache Spark. The number of cameras required and installation charges would depend on the layout of the garage. Integration of this information with mobile application would help customers see if the slots are available even before coming to the garage entrance, thus providing hassle free parking and improved customer satisfaction. The application can be customized for each segment of customers and helps in customer retention. Also the image recognition using cameras can help in preventing fraud and improved security and would be cheaper than motion sensors.

1. **Collaborations with local businesses:**

LAZ can collaborate with local businesses like coffee shots, pub/bars and restaurants who have limited parking. We observed that one of the local Irish pub & restaurant, “the KINSALE Irish pub & Restaurant” is directing their customers to LAZ parking as they don’t have enough parking spaces and compensating the parking fee for three hours if they are spending a minimum of twenty dollars. This information was not available anywhere on the LAZ website or mobile application. Embedding this kind of information will help LAZ to better engage with businesses and customers. It can expand this kind of collaboration with other local businesses and put that information in the local business spot, LAZ website and LAZ mobile application. This will be help customers know about local businesses through LAZ and vice versa. It can also present novel marketing opportunities to both LAZ and the businesses. We have identified some of the businesses around LAZ commercial parking garage who don’t have parking spaces and LAZ can collaborate with them- the Tip Tap Room, Capitol Coffee House and the Oceanaire Seafood Room.

1. **Provision for electric charging of cars:**

There are more than half a million Electric Charging cars in the country and the count is increasing day by day. Major car makers are undertaking projects to get their models in the market. LAZ must provide provision for Electric Vehicle (EV) charging in the garage to attract those customers to keep up with competitors who have already started making such provisions. With both the Federal and State governments giving subsidies for installation of EV charging points the incurred costs are expected to be not too high. Massachusetts is one of the states having highest number of public charging points per square mile as is evident from the following infographic.



1. **Value Added Services:**

We would like to recommend a few value added services for LAZ parking which could be implemented after a thorough cost benefit analysis- for example, a car wash service or an oil changing service in the garage, or in collaboration with a provider to attract more customers. Also, providing wheel chair assist for differently abled persons apart from reserved slots to park would help both the customers and the reputation of LAZ as a thoughtful service provider.

# Conclusion

LAZ needs to streamline the data collection and analysis processes to be able to better utilize the vast amount of data generated in a way which helps to increase footfalls and revenues. One of the most common methods is by increasing prices to increase revenues or decrease prices to increase customer attraction. Keeping in mind that the prices are not entirely flexible and depend on a myriad of factors- from local agreements to competitor behavior- an analysis of the impact of change of prices on the number of customers through our dashboard can help in giving a broad initial idea about the efficacy of the change and the tradeoffs involved. Improved data collection can help in understanding customer behavior and cater to loyal customers in a better way as well as identify possible new customers who share features and are similar to several existing customers

# References

1. US Credit Card Networks Market Share- <https://wallethub.com/edu/market-share-by-credit-card-network/25531>
2. Getting the prices right: An evaluation of pricing parking by demand in San Francisco by Pierce and Shoup
3. Econometric Analysis of Public Parking Price Elasticity in Eugene, Oregon by Moshe Farber and Erin Weld
4. Electric Vehicle Charging Infrastructure- <https://www.statista.com/chart/6586/electric-vehicle-charging-infrastructure>
5. R-Shiny tutorials and help - <https://shiny.rstudio.com/>

# Appendix

**## Data Wrangling using R##**

# PPA Payments

rm(list=ls())

library(dplyr)

ppa\_payments<-read.csv("C:/Users/Avinav/Desktop/Capstone/Data Warehouse Files/Data Warehouse Files/PPA\_Payments.csv")

str(ppa\_payments)

ppa\_payments$DeviceNumber = as.factor(ppa\_payments$DeviceNumber)

ppa\_payments$PaymentType = as.factor(ppa\_payments$PaymentType)

ppa\_payments$PaymentMethod = as.factor(ppa\_payments$PaymentMethod)

parking<- read.csv("C:/Users/Avinav/Desktop/Capstone/Data Warehouse Files/Data Warehouse Files/PAR\_ParkingTxns.csv")

str(parking)

parking$SiteCarParkNumber = as.factor(parking$SiteCarParkNumber)

parking$TicketType = as.factor(parking$TicketType)

parking$ExitDeviceNumber = as.factor(parking$ExitDeviceNumber)

parking$EntryDeviceNumber = as.factor(parking$EntryDeviceNumber)

# Date time manipulation

parking$TradingDate = strptime(parking$TradingDate,"%m/%d/%Y")

parking$EntryDateTime = strptime(parking$EntryDateTime,"%m/%d/%Y %H:%M")

parking$ExitDateTime = strptime(parking$ExitDateTime,"%m/%d/%Y %H:%M")

parking$LastPayDateTime = strptime(parking$LastPayDateTime,"%m/%d/%Y %H:%M")

parking$duration = difftime(parking$ExitDateTime,parking$EntryDateTime,units = "mins")

parking$DayOfWeek = strftime(parking$EntryDateTime,"%A")

parking$Month = strftime(parking$EntryDateTime,"%B")

parking$Year = strftime(parking$EntryDateTime, "%Y")

parking$Time = strftime(parking$EntryDateTime,"%H:%M")

str(parking)

# COnverting Date/Time to suitable format for arranging

parking$TradingDate = as.POSIXct(parking$TradingDate)

parking$EntryDateTime = as.POSIXct(parking$EntryDateTime)

parking$ExitDateTime = as.POSIXct(parking$ExitDateTime)

parking$LastPayDateTime = as.POSIXct(parking$LastPayDateTime)

# Joining the two tables

parking\_payments\_joined = parking%>%select(PaymentNumber,duration,Time,DayOfWeek,Month,

Year,ExitDateTime,Misuse,TicketType)%>%

inner\_join(ppa\_payments,by="PaymentNumber")

# Creating a TimeBand for suitable categorisation

parking\_payments\_joined<-mutate(parking\_payments\_joined, DayType=ifelse(DayOfWeek %in%

c("Monday","Tuesday","Wednesday","Thursday","Friday"),"Weekday","Weekend"))

parking\_payments\_joined = parking\_payments\_joined%>%

mutate(TimeBand = ifelse(DayType=="Weekend","Weekend",

ifelse(Time> '16:00',"Weeknight",

ifelse(duration < 20,"0-20 mins",

ifelse(duration > 20 & duration <40,"20-40 mins",

ifelse(duration > 40 & duration <60,"40-60 mins",

ifelse(duration > 60 & duration <80,"60-80",">80 mins")))))))

# Printing selected variables of the modified tables with the TimeBands

# Weekend

head(select(parking\_payments\_joined,PaymentNumber,duration,Time,DayOfWeek,Month,

DayType,TimeBand,Turnover,Price,NetTurnover,NetPrice,Discounted),20)

# Weekday

head(select(parking\_payments\_joined[parking\_payments\_joined$DayType=="Weekday",],

PaymentNumber,duration,Time,DayOfWeek,Month,

DayType,TimeBand,Turnover,Price,NetTurnover,NetPrice,Discounted),20)

write.csv(parking\_payments\_joined,file="C://Users/Avinav/Desktop/PaymentsJoined.csv")

**## Shiny Dashboard##**

**## app.R ##**

library(shiny)

library(dplyr)

library(shinydashboard)

options(shiny.maxRequestSize=30\*1024^2)

ui <- dashboardPage(

dashboardHeader(title = "LAZ Parking Ticket Price Optimization", titleWidth = 400),

dashboardSidebar(

width = 300,

sidebarMenu(

tags$head(tags$style(".wrapper {overflow: visible !important;}")),

fileInput('file1', 'Choose CSV File',

accept=c('text/csv',

'text/comma-separated-values,text/plain',

'.csv')),

menuItem("Timeband Revenues", tabName = "elastic", icon = icon("dashboard"))

# menuItem("Price Elasticity", tabName = "dataset", icon = icon("dashboard"))

),

helpText("Choose option to increase or",

"decrease the Ticket rate ",

"in specific Timebands (%change)"),

sliderInput("band1", label = ("Weekday/before 4pm/0-20min"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("band2", label = ("Weekday/before 4pm/20-40min"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("band3", label = ("Weekday/before 4pm/40-60min"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("band4", label = ("Weekday/before 4pm/60-80min"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("band5", label = ("Weekday/before 4pm/Over 80min"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("band6", label = ("Weekday/After 4pm-6am"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("band7", label = ("Weekend/6am-6am"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("sensitivity7", label = ("Weekend/6am-6am/sensitivity7"),

min = 0, max = 2, value = 1, step= 0.1),

selectInput("Time Band", "TimeBand:",

c("Weekday/before 4pm/0-20min" = "band1",

"Weekday/before 4pm/20-40min" = "band2",

"Weekday/before 4pm/40-60min" = "band3",

"Weekday/before 4pm/60-80min" = "band4",

"Weekday/before 4pm/Over 80min" = "band5",

"Weekday/After 4pm-6am" = "band6",

"Weekend/6am-6am" = "band7"

)),

sliderInput("sensitivity7", label = ("% Ticket Price Change"),

min = 0, max = 2, value = 1, step= 0.1),

selectInput("Sensitivity", "Sensitivity:",

c("Weekday/before 4pm/0-20min" = "band1",

"Weekday/before 4pm/20-40min" = "band2",

"Weekday/before 4pm/40-60min" = "band3",

"Weekday/before 4pm/60-80min" = "band4",

"Weekday/before 4pm/Over 80min" = "band5",

"Weekday/After 4pm-6am" = "band6",

"Weekend/6am-6am" = "band7"

)),

sliderInput("sensitivity7", label = ("% Sensitivity Change"),

min = 0, max = 2, value = 1, step= 0.1)

),

dashboardBody(

tabItems(

# First tab content

tabItem(tabName = "dataset",

fluidRow(

tabBox(

title = "Price Elasticity Model",

# The id lets us use input$tabset1 on the server to find the current tab

id = "tabset1", height = "250px",

tabPanel("Dataset", dataTableOutput('dataset'))

)

)

),

# Second tab content

tabItem(tabName = "elastic",

fluidRow(

column(width =4,

sliderInput("sensitivity1", label = ("Weekday/before 4pm/0-20min/sensitivity"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("sensitivity2", label = ("Weekday/before 4pm/20-40min/sensitivity"),

min = 0, max = 2, value = 1, step= 0.1)

),

column(width=4,

sliderInput("sensitivity3", label = ("Weekday/before 4pm/40-60min/sensitivity"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("sensitivity4", label = ("Weekday/before 4pm/60-80min/sensitivity"),

min = 0, max = 2, value = 1, step= 0.1)

),

column(width =4,

sliderInput("sensitivity5", label = ("Weekday/before 4pm/Over 80min/sensitivity"),

min = 0, max = 2, value = 1, step= 0.1),

sliderInput("sensitivity6", label = ("Weekday/After 4pm-6am/sensitivity"),

min = 0, max = 2, value = 1, step= 0.1)

),

infoBoxOutput("weekday9"),

infoBoxOutput("weekday9elastic"),

infoBoxOutput("weekday9diff"),

infoBoxOutput("weekday19"),

infoBoxOutput("weekday19elastic"),

infoBoxOutput("weekday19diff"),

infoBoxOutput("weekday29"),

infoBoxOutput("weekday29elastic"),

infoBoxOutput("weekday29diff"),

infoBoxOutput("weekday36"),

infoBoxOutput("weekday36elastic"),

infoBoxOutput("weekday36diff"),

infoBoxOutput("weekday41"),

infoBoxOutput("weekday41elastic"),

infoBoxOutput("weekday41diff"),

infoBoxOutput("weeknight13"),

infoBoxOutput("weeknight13elastic"),

infoBoxOutput("weeknight13diff"),

infoBoxOutput("weekend"),

infoBoxOutput("weekendelastic"),

infoBoxOutput("weekenddiff"),

infoBoxOutput("total"),

infoBoxOutput("totalelastic"),

infoBoxOutput("totaldiff")

)

)

)

)

)

server <- function(input, output) {

data<-reactive({

inFile <- input$file1

if (is.null(inFile))

return(NULL)

na.omit(read.csv(inFile$datapath))

})

#glmoutputs

output$dataset<-renderDataTable({select(data(),PaymentNumber, TranDateTime, DayOfWeek,DayType,duration,Discounted,

NetPrice,NetTurnover,TimeBand)},

options = list(lengthMenu = c(5, 10, 20), pageLength = 20))

output$weekday9 <- renderInfoBox({

infoBox(

"Weekday/0-20", sum(data()[which(data()$TimeBand=="Weekday9"),]$NetPrice), icon = icon("th-list", lib = "glyphicon"),

color = "yellow"

)

})

output$weekday9elastic <- renderInfoBox({

infoBox(

"Weekday/0-20/elastic", (sum(data()[which(data()$TimeBand=="Weekday9"),]$NetPrice)\*input$band1)/(input$sensitivity1), icon = icon("indent-left", lib = "glyphicon"),

color = "green"

)

})

output$weekday9diff <- renderInfoBox({

infoBox(

"Weekday/0-20/diff", ((sum(data()[which(data()$TimeBand=="Weekday9"),]$NetPrice)\*input$band1)/(input$sensitivity1)-sum(data()[which(data()$TimeBand=="Weekday9"),]$NetPrice)), icon = icon("usd", lib = "glyphicon"),

color = "purple"

)

})

output$weekday19 <- renderInfoBox({

infoBox(

"Weekday/20-40", sum(data()[which(data()$TimeBand=="Weekday19"),]$NetPrice), icon = icon("th-list", lib = "glyphicon"),

color = "yellow"

)

})

output$weekday19elastic <- renderInfoBox({

infoBox(

"Weekday/20-40/elastic", (sum(data()[which(data()$TimeBand=="Weekday19"),]$NetPrice)\*input$band2)/(input$sensitivity2), icon = icon("indent-left", lib = "glyphicon"),

color = "green"

)

})

output$weekday19diff <- renderInfoBox({

infoBox(

"Weekday/20-40/diff", ((sum(data()[which(data()$TimeBand=="Weekday19"),]$NetPrice)\*input$band2/(input$sensitivity2))-sum(data()[which(data()$TimeBand=="Weekday19"),]$NetPrice)), icon = icon("usd", lib = "glyphicon"),

color = "purple"

)

})

output$weekday29 <- renderInfoBox({

infoBox(

"Weekday/40-60", sum(data()[which(data()$TimeBand=="Weekday29"),]$NetPrice), icon = icon("th-list", lib = "glyphicon"),

color = "yellow"

)

})

output$weekday29elastic <- renderInfoBox({

infoBox(

"Weekday/40-60/elastic", (sum(data()[which(data()$TimeBand=="Weekday29"),]$NetPrice)\*input$band3)/(input$sensitivity3), icon = icon("indent-left", lib = "glyphicon"),

color = "green"

)

})

output$weekday29diff <- renderInfoBox({

infoBox(

"Weekday/40-60/diff", ((sum(data()[which(data()$TimeBand=="Weekday29"),]$NetPrice)\*input$band3/(input$sensitivity3))-sum(data()[which(data()$TimeBand=="Weekday29"),]$NetPrice)), icon = icon("usd", lib = "glyphicon"),

color = "purple"

)

})

output$weekday36 <- renderInfoBox({

infoBox(

"Weekday/60-80", sum(data()[which(data()$TimeBand=="Weekday36"),]$NetPrice), icon = icon("th-list", lib = "glyphicon"),

color = "yellow"

)

})

output$weekday36elastic <- renderInfoBox({

infoBox(

"Weekday/60-80", (sum(data()[which(data()$TimeBand=="Weekday36"),]$NetPrice)\*input$band4)/(input$sensitivity4), icon = icon("indent-left", lib = "glyphicon"),

color = "green"

)

})

output$weekday36diff <- renderInfoBox({

infoBox(

"Weekday/60-80/elastic", ((sum(data()[which(data()$TimeBand=="Weekday36"),]$NetPrice)\*input$band4/(input$sensitivity4))-sum(data()[which(data()$TimeBand=="Weekday36"),]$NetPrice)), icon = icon("usd", lib = "glyphicon"),

color = "purple"

)

})

output$weekday41 <- renderInfoBox({

infoBox(

"Weekday/above80", sum(data()[which(data()$TimeBand=="Weekday41"),]$NetPrice), icon = icon("th-list", lib = "glyphicon"),

color = "yellow"

)

})

output$weekday41elastic <- renderInfoBox({

infoBox(

"Weekday/above80/elastic", (sum(data()[which(data()$TimeBand=="Weekday41"),]$NetPrice)\*input$band5)/(input$sensitivity5), icon = icon("indent-left", lib = "glyphicon"),

color = "green"

)

})

output$weekday41diff <- renderInfoBox({

infoBox(

"Weekday/above80/diff", ((sum(data()[which(data()$TimeBand=="Weekday41"),]$NetPrice)\*input$band5/(input$sensitivity5))-sum(data()[which(data()$TimeBand=="Weekday41"),]$NetPrice)), icon = icon("usd", lib = "glyphicon"),

color = "purple"

)

})

output$weeknight13 <- renderInfoBox({

infoBox(

"Weeknight/4pm-6am", sum(data()[which(data()$TimeBand=="Weeknight13"),]$NetPrice), icon = icon("th-list", lib = "glyphicon"),

color = "yellow"

)

})

output$weeknight13elastic <- renderInfoBox({

infoBox(

"Weeknight/4pm-6am/elastic", (sum(data()[which(data()$TimeBand=="Weeknight13"),]$NetPrice)\*input$band6)/(input$sensitivity6), icon = icon("indent-left", lib = "glyphicon"),

color = "green"

)

})

output$weeknight13diff <- renderInfoBox({

infoBox(

"Weeknight/4pm-6am/diff", ((sum(data()[which(data()$TimeBand=="Weeknight13"),]$NetPrice)\*input$band6/(input$sensitivity6))-sum(data()[which(data()$TimeBand=="Weeknight13"),]$NetPrice)), icon = icon("usd", lib = "glyphicon"),

color = "purple"

)

})

output$weekend <- renderInfoBox({

infoBox(

"Weekend", sum(data()[which(data()$TimeBand=="Weekend"),]$NetPrice), icon = icon("th-list", lib = "glyphicon"),

color = "yellow"

)

})

output$weekendelastic <- renderInfoBox({

infoBox(

"Weekend/elastic", (sum(data()[which(data()$TimeBand=="Weekend"),]$NetPrice)\*input$band7)/(input$sensitivity7), icon = icon("indent-left", lib = "glyphicon"),

color = "green"

)

})

output$weekenddiff <- renderInfoBox({

infoBox(

"Weekend/diff", ((sum(data()[which(data()$TimeBand=="Weekend"),]$NetPrice)\*input$band7/(input$sensitivity7))-sum(data()[which(data()$TimeBand=="Weekend"),]$NetPrice)), icon = icon("usd", lib = "glyphicon"),

color = "purple"

)

})

output$total <- renderInfoBox({

infoBox(

"totalRevenue", sum(data()$NetPrice/100), icon = icon("th-list", lib = "glyphicon"),

color = "yellow", fill=TRUE

)

})

output$totalelastic <- renderInfoBox({

infoBox(

"totalRevenue/elastic",(sum(data()[which(data()$TimeBand=="Weekend"),]$NetPrice)\*input$band7/(input$sensitivity7\*100)+ sum(data()[which(data()$TimeBand=="Weeknight13"),]$NetPrice)\*input$band6/(input$sensitivity6\*100)+ sum(data()[which(data()$TimeBand=="Weekday41"),]$NetPrice)\*input$band5/(input$sensitivity5\*100)+ sum(data()[which(data()$TimeBand=="Weekday36"),]$NetPrice)\*input$band4/(input$sensitivity4\*100)+ sum(data()[which(data()$TimeBand=="Weekday29"),]$NetPrice)\*input$band3/(input$sensitivity3\*100)+ sum(data()[which(data()$TimeBand=="Weekday19"),]$NetPrice)\*input$band2/(input$sensitivity2\*100)+ sum(data()[which(data()$TimeBand=="Weekday9"),]$NetPrice)\*input$band1/(input$sensitivity1\*100))

, icon = icon("indent-left", lib = "glyphicon"),

color = "green", fill=TRUE

)

})

output$totaldiff <- renderInfoBox({

infoBox(

"totalRevenue/diff",((sum(data()[which(data()$TimeBand=="Weekend"),]$NetPrice)\*input$band7/(input$sensitivity7\*100)+

sum(data()[which(data()$TimeBand=="Weeknight13"),]$NetPrice)\*input$band6/(input$sensitivity6\*100)+

sum(data()[which(data()$TimeBand=="Weekday41"),]$NetPrice)\*input$band5/(input$sensitivity5\*100)+ sum(data()[which(data()$TimeBand=="Weekday36"),]$NetPrice)\*input$band4/(input$sensitivity4\*100)+ sum(data()[which(data()$TimeBand=="Weekday29"),]$NetPrice)\*input$band3/(input$sensitivity3\*100)+ sum(data()[which(data()$TimeBand=="Weekday19"),]$NetPrice)\*input$band2/(input$sensitivity2\*100)+ sum(data()[which(data()$TimeBand=="Weekday9"),]$NetPrice)\*input$band1/(input$sensitivity1\*100))-sum(data()$NetPrice/100))

, icon = icon("usd", lib = "glyphicon"),

color = "purple", fill=TRUE

)

})

}

shinyApp(ui, server)