---

title: "Comcast-cust-rprog"

author: "Shiietal Palshetkar"

date: "14/07/2023"

output:

pdf\_document: default

word\_document: default

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the \*\*Knit\*\* button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r}

#Importing necessary packages

library(dplyr)

library(ggplot2)

library(lubridate)

library(readr)

library(tidyverse)

#Importing Comcast Dataset

comcast\_data <- read\_csv("/home/labsuser/Projects/Comcast Telecom Complaints data.csv")

View(comcast\_data)

```

```{r}

#Manipulating Field Names

names(comcast\_data)<-gsub(pattern = '\\.',replacement = "",x=names(comcast\_data))

names(comcast\_data)

View(comcast\_data)

#From the dataset, we can see that the format of Date column is not same throughout, so we need to make it same for analysis.

#Processing Date

comcast\_data$Date<- dmy(comcast\_data$Date)

View(comcast\_data)

#Now we need to get the complaints on a daily level basis and plot a trend chart for it.

ans<-comcast\_data %>% group\_by(Date) %>% summarize(NumOfComplaints=n())

#Plotting for daily granularity level

ggplot(data = ans,aes(as.POSIXct(Date),NumOfComplaints))+

geom\_line()+

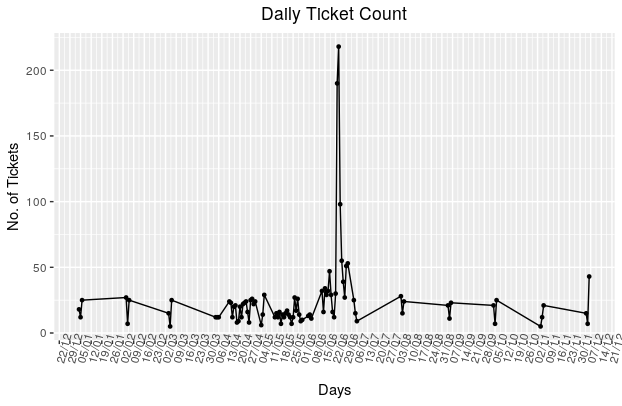
geom\_point(size = 1)+

scale\_x\_datetime(breaks = "1 weeks",date\_labels = "%d/%m")+

labs(title = "Daily Ticket Count",x= "Days",y ="No. of Tickets")+

theme(axis.text.x = element\_text(angle = 75),

plot.title = element\_text(hjust = 0.5))



#INSIGHTS:- From the above trend chart, we can clearly see that complaints for the month of June are maximum i.e.1046.

```

```{r}

#Now we need to make a frequency table basis the complaint types.

# Complaint Type Processing

network\_tickets<- contains(comcast\_data$`Customer Complaint`,match = 'network',ignore.case = T)

internet\_tickets<- contains(comcast\_data$`Customer Complaint`,match = 'internet',ignore.case = T)

billing\_tickets<- contains(comcast\_data$`Customer Complaint`,match = 'bill',ignore.case = T)

email\_tickets<- contains(comcast\_data$`Customer Complaint`,match = 'email',ignore.case = T)

charges\_ticket<- contains(comcast\_data$`Customer Complaint`,match = 'charge',ignore.case = T)

comcast\_data$ComplaintType[internet\_tickets]<- "Internet"

comcast\_data$ComplaintType[network\_tickets]<- "Network"

comcast\_data$ComplaintType[billing\_tickets]<- "Billing"

comcast\_data$ComplaintType[email\_tickets]<- "Email"

comcast\_data$ComplaintType[charges\_ticket]<- "Charges"

comcast\_data$ComplaintType[-c(internet\_tickets,network\_tickets,

billing\_tickets,charges\_ticket,email\_tickets)]<- "Others"

table(comcast\_data$ComplaintType)

View(comcast\_data)

#INSIGHTS:- From the above table we can see that the Internet type complaints are maximum.

```

```{r}

#Now we need to make a new categorical variable for Complaint Status.

open\_complaints<-(comcast\_data$Status == 'Open' | comcast\_data$Status == 'Pending')

closed\_complaints<-(comcast\_data$Status == 'Closed' | comcast\_data$Status == 'Solved')

comcast\_data$ComplaintStatus[open\_complaints]<-'Open'

comcast\_data$ComplaintStatus[closed\_complaints]<-'Closed'

# Now we need to plot state wise status of complaints in a stacked bar chart.

stack<-table(comcast\_data$ComplaintStatus,comcast\_data$State)

stack

comcast\_data<- group\_by(comcast\_data,State,ComplaintStatus)

chart\_data<- summarise(comcast\_data,Count = n())

#Plotting on stacked bar chart

ggplot(as.data.frame(chart\_data), mapping = aes(State, Count)) +

geom\_col(aes(fill = ComplaintStatus), width = 0.95) +

theme(

axis.text.x = element\_text(angle = 90),

axis.title.y = element\_text(size = 15),

axis.title.x = element\_text(size = 15),

title = element\_text(size = 16, colour = "#0073C2FF"),

plot.title = element\_text(hjust = 0.5)

) +

labs(

title = "Ticket Status Stacked Bar Chart",

x = "States",

y = "No of Tickets",

fill = "Status"

)

Chart

Description automatically generated

#INSIGHTS:- From the above chart, we can clearly see that Georgia has maximum complaints.

```

```{r}

#Now we need to see which state has maximum unresolved complaints

comcast\_data %>% filter(ComplaintStatus=='Open') %>% group\_by(State) %>% summarize(NumOfComplaints=n()) %>% arrange(desc(NumOfComplaints))

#INSIGHTS:- From the table generated above we can see that Georgia has maximum unresolved complaints i.e. 80.

```

```{r}

#Now we want to see the percentage of resolved complaints.

tot<-comcast\_data %>% group\_by(ComplaintStatus) %>% summarize(NumOfComplaints=n())

tot

slices<-tot$NumOfComplaints

pct<-round((slices/sum(slices)\*100),2)

pct

lbls<-paste(tot$ComplaintStatus," ",pct,"%",sep="")

#Plotting pie chart

pie(slices,labels=lbls)

A picture containing chart

Description automatically generated

#INSIGHTS:- From the above pie chart we can clearly see that there are total 76.75% Complaints resolved.

```

```

## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}

plot(pressure)

```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.