Comcast project

Swathi V Hebbar

17/12/2020

Comcast is an American global telecommunication company. The firm has been providing terrible customer service. They continue to fall short despite repeated promises to improve. Only last month (October 2016) the authority fined them a $2.3 million, after receiving over 1000 consumer complaints.

The existing database will serve as a repository of public customer complaints filed against Comcast. It will help to pin down what is wrong with Comcast's customer service.

Task Need to be performed:

- Importing data into R environment.
- Provide the trend chart for the number of complaints at monthly and daily granularity levels.
- Provide a table with the frequency of complaint types. -Which complaint types are maximum i.e., around internet, network issues, or across any other domains.
- Create a new categorical variable with value as Open and Closed. Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed
- Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on: -Which state has the maximum complaints -Which state has the highest percentage of unresolved complaints
- Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

1. Import the data into R environment

```
# 1.   Import the data into R environment
Comcast <- read.csv(file.choose())
str(Comcast)
```

```
## 'data.frame':    2224 obs. of  10 variables:
##  $ Ticket..                   : chr  "250635" "223441" "242732" "277946" ...
##  $ Customer.Complaint         : chr  "Comcast Cable Internet Speeds" "Payment disappear - service go
##  $ Date                       : chr  "22-04-2015" "4/8/2015" "18-04-2015" "5/7/2015" ...
##  $ Time                       : chr  "3:53:50 PM" "10:22:56 AM" "9:55:47 AM" "11:59:35 AM" ...
##  $ Received.Via               : chr  "Customer Care Call" "Internet" "Internet" "Internet" ...
##  $ City                       : chr  "Abingdon" "Acworth" "Acworth" "Acworth" ...
##  $ State                      : chr  "Maryland" "Georgia" "Georgia" "Georgia" ...
##  $ Zip.code                   : int  21009 30102 30101 30101 30101 30101 30101 49221 94502 94501 ...
##  $ Status                     : chr  "Closed" "Closed" "Closed" "Open" ...
##  $ Filing.on.Behalf.of.Someone: chr  "No" "No" "Yes" "Yes" ...
```

Now that we have uploaded the file , we can check for proper variable name in the data frame

```
# to correct all the variable names in the data frame
names(Comcast) <- c("Ticket","Customer complaint","Date","Time",
        "Received Via","City","State","Zip Code","Status","Filling for others")
View(Comcast)
str(Comcast)
```

```
## 'data.frame':    2224 obs. of  10 variables:
##  $ Ticket            : chr  "250635" "223441" "242732" "277946" ...
##  $ Customer complaint: chr  "Comcast Cable Internet Speeds" "Payment disappear - service got disconn
##  $ Date              : chr  "22-04-2015" "4/8/2015" "18-04-2015" "5/7/2015" ...
##  $ Time              : chr  "3:53:50 PM" "10:22:56 AM" "9:55:47 AM" "11:59:35 AM" ...
##  $ Received Via      : chr  "Customer Care Call" "Internet" "Internet" "Internet" ...
##  $ City              : chr  "Abingdon" "Acworth" "Acworth" "Acworth" ...
##  $ State             : chr  "Maryland" "Georgia" "Georgia" "Georgia" ...
##  $ Zip Code          : int  21009 30102 30101 30101 30101 30101 30101 49221 94502 94501 ...
##  $ Status            : chr  "Closed" "Closed" "Closed" "Open" ...
##  $ Filling for others: chr  "No" "No" "Yes" "Yes" ...
```

Now the variables of Comcast data are proper. Now we can observe the variable "Date" is not sorted properly. So we will create a new variable in Comcast with the name Date_Modified whose entries are neatly arranged with the date format %dd/%mm/%yy.

```
#install.packages("lubridate")
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
Comcast$Date_Modified <- parse_date_time(Comcast$Date,"%d!/%m!/%Y!")
View(Comcast)
str(Comcast)
```

```
## 'data.frame':    2224 obs. of  11 variables:
##  $ Ticket            : chr  "250635" "223441" "242732" "277946" ...
##  $ Customer complaint: chr  "Comcast Cable Internet Speeds" "Payment disappear - service got disconn
##  $ Date              : chr  "22-04-2015" "4/8/2015" "18-04-2015" "5/7/2015" ...
##  $ Time              : chr  "3:53:50 PM" "10:22:56 AM" "9:55:47 AM" "11:59:35 AM" ...
##  $ Received Via      : chr  "Customer Care Call" "Internet" "Internet" "Internet" ...
##  $ City              : chr  "Abingdon" "Acworth" "Acworth" "Acworth" ...
##  $ State             : chr  "Maryland" "Georgia" "Georgia" "Georgia" ...
##  $ Zip Code          : int  21009 30102 30101 30101 30101 30101 30101 49221 94502 94501 ...
##  $ Status            : chr  "Closed" "Closed" "Closed" "Open" ...
##  $ Filling for others: chr  "No" "No" "Yes" "Yes" ...
##  $ Date_Modified     : POSIXct, format: "2015-04-22" "2015-08-04" ...
```

Now we check for any NA present in the data.

```
sum(is.na(Comcast))
```

```
## [1] 0
```

The count of missing values in the dataset is zero. So we can proceed further. Now we define the variables as factor, character and numeric depending on their behaviour.

```
Comcast$Ticket<- as.numeric(Comcast$Ticket)
```

```
## Warning: NAs introduced by coercion
```

```
Comcast$`Received Via`<- as.factor(Comcast$`Received Via`)
Comcast$City<- as.factor(Comcast$City)
Comcast$State<- as.factor(Comcast$State)
Comcast$`Zip Code`  <- as.factor(Comcast$`Zip Code`)
Comcast$Status <- as.factor(Comcast$Status)
Comcast$`Filling for others` <- as.factor(Comcast$`Filling for others`)

str(Comcast)
```

```
## 'data.frame':    2224 obs. of  11 variables:
##  $ Ticket            : num  250635 223441 242732 277946 307175 ...
##  $ Customer complaint: chr  "Comcast Cable Internet Speeds" "Payment disappear - service got disconn
##  $ Date              : chr  "22-04-2015" "4/8/2015" "18-04-2015" "5/7/2015" ...
##  $ Time              : chr  "3:53:50 PM" "10:22:56 AM" "9:55:47 AM" "11:59:35 AM" ...
##  $ Received Via      : Factor w/ 2 levels "Customer Care Call",..: 1 2 2 2 2 2 1 2 1 1 ...
##  $ City              : Factor w/ 928 levels "Abingdon","Acworth",..: 1 2 2 2 2 2 2 3 4 4 ...
##  $ State             : Factor w/ 43 levels "Alabama","Arizona",..: 19 11 11 11 11 11 11 21 4 4 ...
##  $ Zip Code          : Factor w/ 1543 levels "1075","1082",..: 298 438 437 437 437 437 437 945 1317
##  $ Status            : Factor w/ 4 levels "Closed","Open",..: 1 1 1 2 4 4 3 4 1 2 ...
##  $ Filling for others: Factor w/ 2 levels "No","Yes": 1 1 2 2 1 1 1 1 1 2 ...
##  $ Date_Modified     : POSIXct, format: "2015-04-22" "2015-08-04" ...
```

2. Provide the trend chart for the number of complaints at daily and monthly granularity levels.

```
#install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages("ggplot2")
library(ggplot2)
library(magrittr)

Dialy_report<- Comcast %>% group_by(Date_Modified) %>% summarise(No_of_complaints_Dialy=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

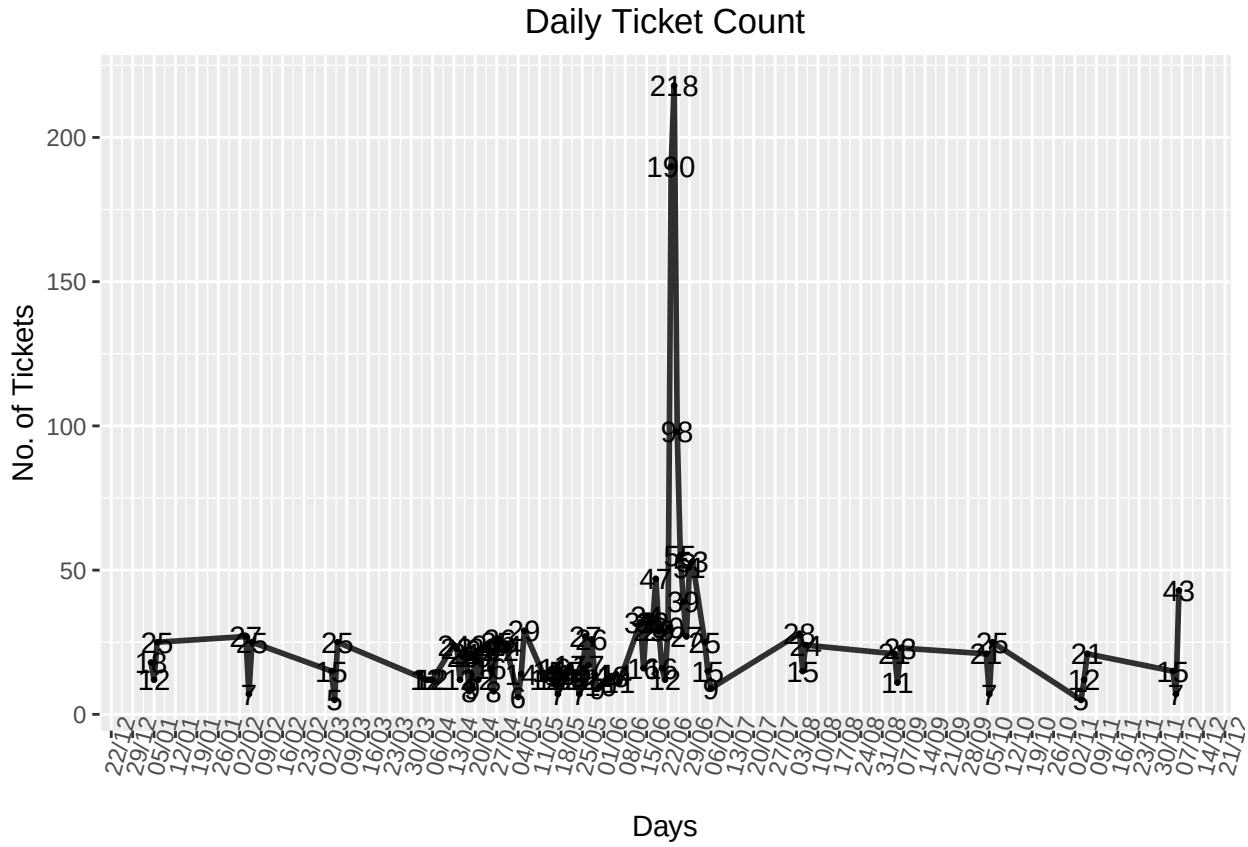
```
#Dialy report of the complaints
View(Dialy_report)
Monthly_report<-Comcast %>% group_by(Month=month(Comcast$Date_Modified)) %>%  summarise(No_of_complaints
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#Monthly report of the complaints
View(Monthly_report)
```

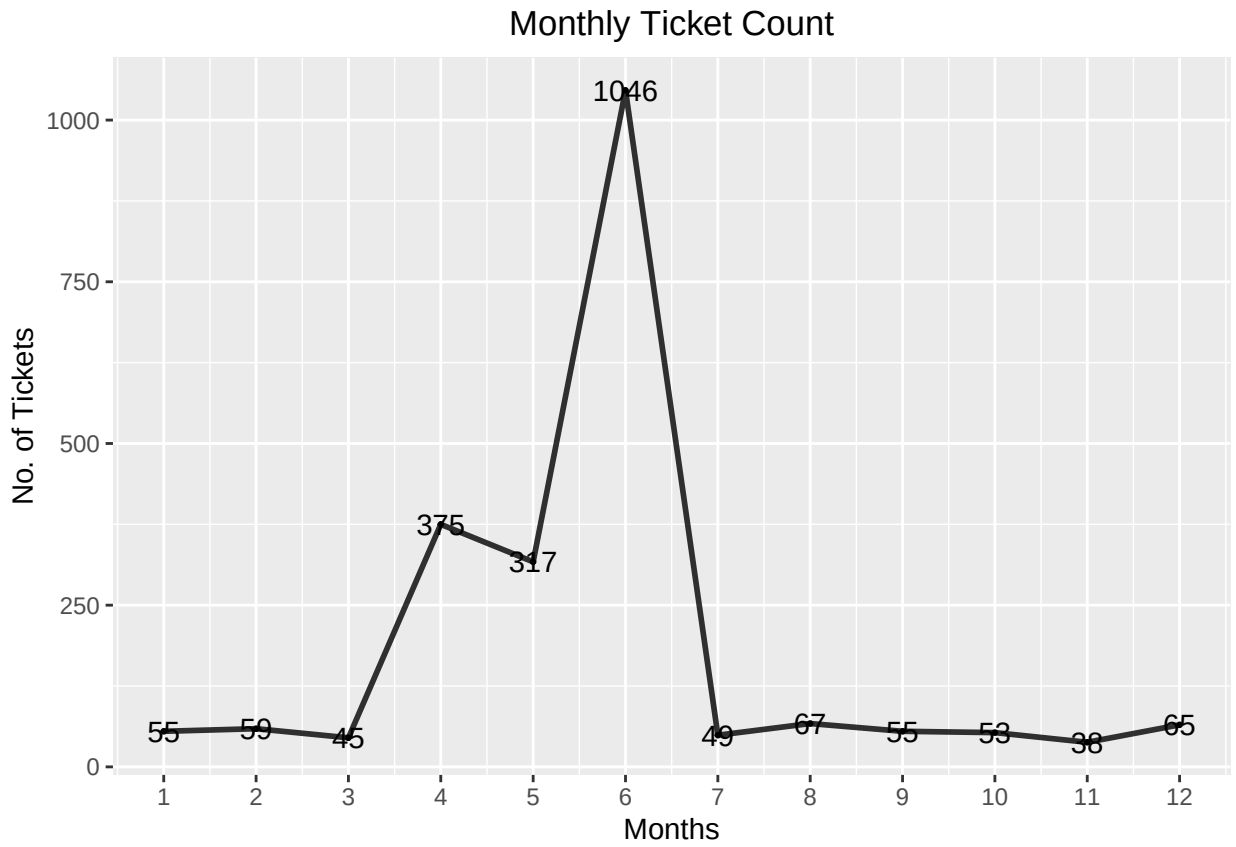
Now plotting the trend chart for number of complaints on daily basis.

```
ggplot(Dialy_report,aes(Date_Modified,No_of_complaints_Dialy,label=No_of_complaints_Dialy)) +
  geom_line(size = 1, alpha = 0.8) +
  geom_point(size = 0.5) +
  geom_text()+
  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))
```



From the help of Daily report,we can observe that from second half of June, we received more complaints than normal days. Now we plot the trend chart for monthly report

```
ggplot(Monthly_report,aes(Month,No_of_complaints_monthly,label=No_of_complaints_monthly)) +
  geom_line(size = 1, alpha = 0.8) +
  geom_point(size = 0.5) +
  geom_text()+
  scale_x_continuous(breaks = Monthly_report$Month)+
  labs(title = "Monthly Ticket Count",x= "Months",y ="No. of Tickets")+
  theme(plot.title = element_text(hjust = 0.5))
```



Now we can see from the report that, in the mont of April, May and June the complaints were high.

3. Provide a table with the frequency of complaint types.

In order to find the complaint types, we need to analyze the Complaints of customer received. If we know the most common words used in the complaints then we can create a new variable with Complaint types as the factor.To know the most common words used in the complaints we can use data visualization.

```
# install.packages("tm")
# install.packages("SnowballC")
# install.packages("RColorBrewer")
# install.packages("wordcloud")
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(SnowballC)
library("wordcloud")
```

```
## Loading required package: RColorBrewer
```

```
library("RColorBrewer")

# Column containing text into a corpus for pre processing
corpus = Corpus(VectorSource(Comcast$`Customer complaint`))
#converting to lower case
corpus = tm_map(corpus, PlainTextDocument)
```

```
## Warning in tm_map.SimpleCorpus(corpus, PlainTextDocument): transformation drops
## documents
```

```
corpus = tm_map(corpus, tolower)
```

```
## Warning in tm_map.SimpleCorpus(corpus, tolower): transformation drops documents
```

```
#Removing Punctuation
corpus = tm_map(corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents
```

```
#removing words
corpus = tm_map(corpus, removeWords, c("and", stopwords("english")))
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeWords, c("and",
## stopwords("english"))): transformation drops documents
```

```
# Stemming
corpus = tm_map(corpus, stemDocument)
```

```
## Warning in tm_map.SimpleCorpus(corpus, stemDocument): transformation drops
## documents
```

```
# Eliminate white spaces
corpus = tm_map(corpus, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(corpus, stripWhitespace): transformation drops
## documents
```

```
#creating term doccument matrix
DTM <- TermDocumentMatrix(corpus)
mat <- as.matrix(DTM)
f <- sort(rowSums(mat),decreasing=TRUE)
dat <- data.frame(word = names(f),freq=f)
#printing the first 5 most used words
head(dat, 5)
```

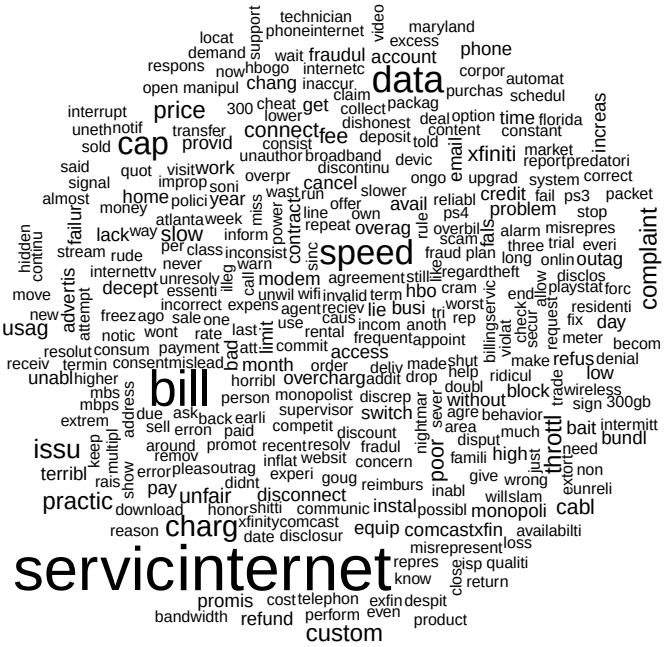
```
##              word freq
## comcast   comcast 1200
## internet internet  517
## servic     servic  496
## bill         bill  361
## data         data  219
```

Here we got to know that most common used words in Complaints of Comcast data set. But still we can have a better idea if we look into the wordcloud.

```
set.seed(1234)
wordcloud(words = dat$word, freq = dat$freq, random.order=TRUE)
```

```
## Warning in wordcloud(words = dat$word, freq = dat$freq, random.order = TRUE):
```

```
## comcast could not be fit on page. It will not be plotted.
```



From analyzing the Wordcloud, we get most commonly used words in the complaints are Internet, Service,Billing and Charges. So we can create a new categorical variable in Comcast which mentions its complaint types.The complaint types which doesn't fall on any above category are grouped as others.

```
Internet_Complaints<- contains(Comcast$`Customer complaint`,match="Internet",ignore.case = T)
Service_Complaints <- contains(Comcast$`Customer complaint`,match="Service",ignore.case = T)
Billing_Complaints <- contains(Comcast$`Customer complaint`,match="Billing",ignore.case = T)
Charges_Complaints <- contains(Comcast$`Customer complaint`,match="Charges",ignore.case=T)

Comcast$ComplaintType[Internet_Complaints]<- "Internet"
Comcast$ComplaintType[Service_Complaints]<- "Service"
Comcast$ComplaintType[Billing_Complaints]<- "Billing"
Comcast$ComplaintType[Charges_Complaints]<- "Charges"

Comcast$ComplaintType[-c(Internet_Complaints,
                         Service_Complaints,Billing_Complaints,Charges_Complaints)]<- "Others"
# to view the complaint type of each data which was created
View(Comcast)
```

4 Provide a table with the frequency of complaint types.

```
table(Complaint_type=Comcast$ComplaintType)
```

```
## Complaint_type
##  Billing  Charges Internet   Others  Service
```

```
##      294       77      368     1068      417
```

From the table, we get to know that apart from others(Which are not specified exactly), service related complaints are more.

5 Create a new categorical variable with value as Open and Closed, Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed.

```
Comcast<- transform(Comcast,Status_New = ifelse(Status %in% c("Pending","Open"),"Open",
                                                if(Comcast$Status %in% c("Closed","Solved")){
                                                  "Closed"
                                                }
                    ))
```

```
## Warning in if (Comcast$Status %in% c("Closed", "Solved")) {: the condition has
## length > 1 and only the first element will be used
```

```
Comcast$Status_New<- as.factor(Comcast$Status_New)
View(Comcast)
str(Comcast)
```

```
## 'data.frame':    2224 obs. of  13 variables:
##  $ Ticket            : num  250635 223441 242732 277946 307175 ...
##  $ Customer.complaint: chr  "Comcast Cable Internet Speeds" "Payment disappear - service got disconn
##  $ Date              : chr  "22-04-2015" "4/8/2015" "18-04-2015" "5/7/2015" ...
##  $ Time              : chr  "3:53:50 PM" "10:22:56 AM" "9:55:47 AM" "11:59:35 AM" ...
##  $ Received.Via      : Factor w/ 2 levels "Customer Care Call",..: 1 2 2 2 2 2 1 2 1 1 ...
##  $ City              : Factor w/ 928 levels "Abingdon","Acworth",..: 1 2 2 2 2 2 2 3 4 4 ...
##  $ State             : Factor w/ 43 levels "Alabama","Arizona",..: 19 11 11 11 11 11 11 21 4 4 ...
##  $ Zip.Code          : Factor w/ 1543 levels "1075","1082",..: 298 438 437 437 437 437 437 945 1317
##  $ Status            : Factor w/ 4 levels "Closed","Open",..: 1 1 1 2 4 4 3 4 1 2 ...
##  $ Filling.for.others: Factor w/ 2 levels "No","Yes": 1 1 2 2 1 1 1 1 1 2 ...
##  $ Date_Modified     : POSIXct, format: "2015-04-22" "2015-08-04" ...
##  $ ComplaintType     : chr  "Internet" "Service" "Service" "Others" ...
##  $ Status_New        : Factor w/ 2 levels "Closed","Open": 1 1 1 2 1 1 2 1 1 2 ...
```

New variable is created with the name Status_New which gives status as open and closed only.

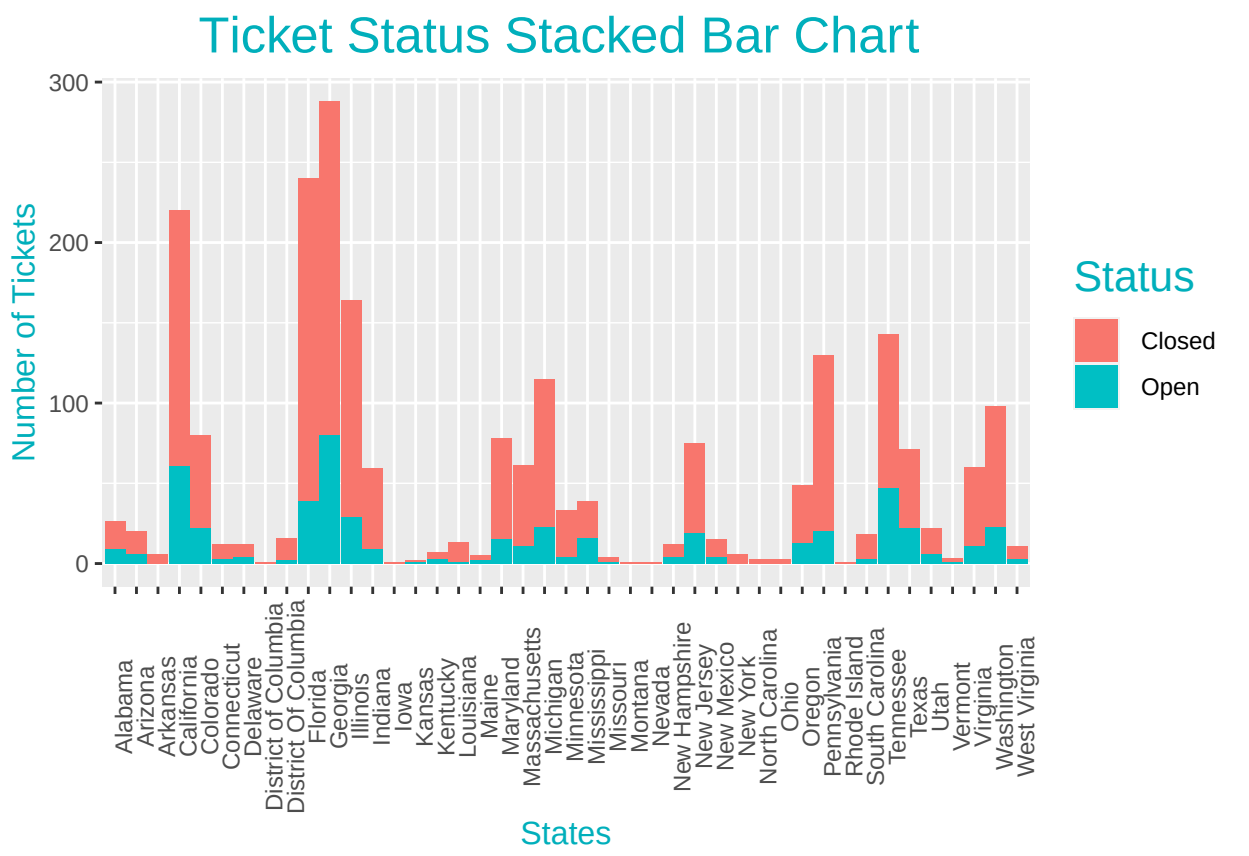
6 Provide state wise status of complaints in a stacked bar chart.

```
library(ggplot2)
#Getting statewise com[plaints
Comcast<- group_by(Comcast,State,Status_New)
X_axis_Chart<- summarise(Comcast,Count = n())
```

```
## `summarise()` regrouping output by 'State' (override with `.groups` argument)
```

```
#plotting stacked bar plot
ggplot(as.data.frame(X_axis_Chart) , aes(State,Count))+
  geom_col(aes(fill = Status_New),width = 0.95)+
  theme(axis.text.x = element_text(angle = 90),
        axis.title.y = element_text(size = 12),
        axis.title.x = element_text(size = 12),
        title = element_text(size = 16,colour = "#00AFBB"),
```

```
      plot.title = element_text(hjust =  0.5))+
 labs(title = "Ticket Status Stacked Bar Chart ",
      x = "States",y = "Number of Tickets",
      fill= "Status")
```



Here, from the stacked bar chart, it gives clear information about the status of the complaints state wise. And we can see that Georgia has the highest number of complaints.

7 Which state has the maximum complaints

It is well known from the graph that Georgia has the most complaints, still if you want you can actually check it from the command below.

```
State_Report<- Comcast %>% group_by(State) %>% summarise(Total=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
View(State_Report)
State_Report %>% filter(Total==max(State_Report$Total))
```

```
## # A tibble: 1 x 2
##   State   Total
##   <fct>   <int>
## 1 Georgia   288
```

8 Provide the percentage of complaints resolved till date

For this, if we just use the table function inorder to find the count of open and closed complaints,

```
table(Comcast$Status_New)
```

```
##
## Closed   Open
##   1707    517
```

As you saw, we will get to know only the counts of open and closed complaints, but not the %. So, we follow this procedure. So performing only these below commands are enough unless we want the count.

```
resolved_data <- group_by(Comcast,Status_New)
total_resloved<- summarise(resolved_data ,percentage =(n()/nrow(resolved_data))*100)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#total open and closed complaints
total_resloved
```

```
## # A tibble: 2 x 2
##   Status_New percentage
##   <fct>           <dbl>
## 1 Closed           76.8
## 2 Open             23.2
```

We got to know, over all, 76.75% of the complaints are closed. But 23.24% of the complaints are still open. In order to find the Complaints according to the received mode, we can do the same.

```
table(Comcast$Received.Via)
```

```
##
## Customer Care Call           Internet
##               1119               1105
```

We got to know only the count of complaints from Customer care and Internet. Instead we can follow this to to find he % of complaints received via different mode.

```
resolved_data <- group_by(Comcast,Received.Via,Status_New)
Category_resloved<- summarise(resolved_data ,percentage =(n()/nrow(resolved_data))*100)
```

```
## `summarise()` regrouping output by 'Received.Via' (override with `.groups` argument)
```

```
#total complaints received via different modes
Category_resloved
```

```
## # A tibble: 4 x 3
## # Groups:   Received.Via [2]
##   Received.Via       Status_New percentage
##   <fct>              <fct>           <dbl>
## 1 Customer Care Call Closed           38.8
## 2 Customer Care Call Open             11.5
## 3 Internet           Closed           37.9
## 4 Internet           Open             11.8
```

Clearly we can see both the categories have similar kind of frequency for complaints intake.

9 Which state has the highest percentage of unresolved complaints

Here its a tricky question! Because if we want to know overall % of open and closed complaints, then it will be meaningless in terms of States. Because we have 2223 complaints(among which 517 open and 1707 closed complaints-can see from the above table) and some states(Alabama) has 26 complaints(17 closed and 9 open), so if we calculate the % of overall unresolved complaints in that, it becomes, unresolved=9/517*100=1.74%. So instead of calculating overall % for the states, i tried calculating the % for each states. Depending on the Complaints each state received, we will be calculating the % for open and as well as closed complaints.

```
Table1<-table(Comcast$State,Comcast$Status_New)
table_df<- as.data.frame.matrix(Table1)
table_df2<-group_by(Comcast,State) %>% summarise(Total_val=n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
table_df2$open<-cbind(table_df$Open)
table_df2$closed<-cbind(table_df$Closed)
#statewise data for status of complaints
View(table_df2)
Statewise_df<-transform(table_df2,Unresolved_percentage=round(table_df2$open/table_df2$Total_val*100,2)
Statewise_df<-transform(Statewise_df,Resolved_percentage=round(table_df2$closed/table_df2$Total_val*100
#Statewise data of % of resolved and unresolved complaints
View(Statewise_df)
#Highest % of unresolved complaints statewise
Statewise_df %>% filter(Unresolved_percentage==max(Statewise_df$Unresolved_percentage))
```

```
##    State Total_val open closed Unresolved_percentage Resolved_percentage
## 1 Kansas         2    1      1                    50                  50
```

Here, for each state % will be calculated. But the drawback is that, as you can see, Kansas has only one open complaint. But overall complaint was 2. So it has 50% unresolved or resolved %. But we will get a clear picture about all the complaints for each state which will be helpfull if the complaints are more in states like Georgia.