**<u>All source code below</u>**

```
#######Comcast Telecom Consumer Complaints##########

#load all Libraries once

library(lettercase)
library(lubridate)
library(ggplot2)
library(dplyr)
library(hrbrthemes)

setwd("D:/google_downloads/take/r/r-project/final")
getwd()

#connect to CSV file
Comcast_data <- read.csv("D:/google_downloads/take/r/r-project/final/Comcast
Telecom Complaints data.csv")
#view file data
View(Comcast_data)
#get data stucture--info
str(Comcast_data)

#Change all date in the same format from
#Use lubridate Library
# convert date
Comcast_data$Date<-gsub("/","-",Comcast_data$Date)
parse_date_time(Comcast_data$Date, orders="dmY")

#####performing EDA and Formating Data First to desirable formats###########
#Adding a month column first

#tail(Comcast_data)
# extract month only from date field
Comcast_data_month <- format(as.Date(Comcast_data$Date), "%m")
#Create extra column called Month1 where Full month Names can be tagged
add_month1 <- transform(Comcast_data,Month1 = ifelse(Comcast_data_month
=="01","01-January",ifelse(Comcast_data_month =="02","02-
February",ifelse(Comcast_data_month =="03","03-
March",ifelse(Comcast_data_month =="04","04-April",ifelse(Comcast_data_month
=="05","05-May",ifelse(Comcast_data_month =="06","06-
June",ifelse(Comcast_data_month =="07","07-July",ifelse(Comcast_data_month
=="08","08-August",ifelse(Comcast_data_month =="09","09-
September",ifelse(Comcast_data_month =="10","10-
October",ifelse(Comcast_data_month =="11","11-
November",ifelse(Comcast_data_month =="12","12-December","Not Yet Set"
))))))))))))
View(add_month1)
#add CountCustomerComplaints count field
Comcast_data_final <- transform(add_month1,CountCustomerComplaints=1)
View(Comcast_data_final)

Comcast_data_SUmm_plot_data <- Comcast_data_final %>% group_by(Month1)
%>%
summarise(CountCustomerComplaints=sum(CountCustomerComplaints,na.rm=TR
UE))
View(Comcast_data_SUmm_plot_data)



#########Trend : Customer Complaints Per Month###########
# Use the ggplot2,hrbrthemes Libraries
# Used and modified : http://www.sthda.com/english/wiki/ggplot2-line-types-how-to-
change-line-types-of-a-graph-in-r-software
df <- data.frame(Month=c("01-January","02-February","03-March","04-April","05-
May","06-June","07-July","08-August","09-September","10-October","11-
November","12-December"),

Count_Customer_Complaints=c(Comcast_data_SUmm_plot_data$CountCustomer
Complaints))
#head(df)
# Basic line plot with points
ggplot(data=df, aes(x=Month, y=Count_Customer_Complaints, group=1)) +
  geom_line()+
  geom_point()
# Change the line type
ggplot(data=df, aes(x=Month, y=Count_Customer_Complaints, group=1)) +
  geom_line(linetype = "dashed",size = 1, alpha = 0.8,color="darkgreen")+
  geom_point(color="#69b3a2", size=4) +
  ggtitle("Trend : Customer Complaints Per Month") +
  ylab("Customer Complaints (Count)") +
  geom_point()

#######daily Trend

Comcast_data_day <- format(as.Date(Comcast_data_final$Date,format="%d-%m-
%Y"), format = "%d")
#Create extra column called Day1 where days can be tagged
add_day1 <- transform(Comcast_data_final,Day1 = ifelse(Comcast_data_day
=="01","01",ifelse(Comcast_data_day =="02","02",ifelse(Comcast_data_day
=="03","03",ifelse(Comcast_data_day =="04","04",ifelse(Comcast_data_day
=="05","05",ifelse(Comcast_data_day =="06","06",ifelse(Comcast_data_day
=="07","07",ifelse(Comcast_data_day =="08","08",ifelse(Comcast_data_day
=="09","09",ifelse(Comcast_data_day =="10","10",ifelse(Comcast_data_day
=="11","11",ifelse(Comcast_data_day =="12","12",ifelse(Comcast_data_day
=="13","13",ifelse(Comcast_data_day =="14","14",ifelse(Comcast_data_day
=="15","15",ifelse(Comcast_data_day =="16","16",ifelse(Comcast_data_day
=="17","17",ifelse(Comcast_data_day =="18","18",ifelse(Comcast_data_day
=="19","19",ifelse(Comcast_data_day =="20","20",ifelse(Comcast_data_day
=="21","21",ifelse(Comcast_data_day =="22","22",ifelse(Comcast_data_day
=="23","23",ifelse(Comcast_data_day =="24","24",ifelse(Comcast_data_day
=="25","25",ifelse(Comcast_data_day =="26","26",ifelse(Comcast_data_day
=="27","27",ifelse(Comcast_data_day =="28","28",ifelse(Comcast_data_day
=="29","29",ifelse(Comcast_data_day =="30","30",ifelse(Comcast_data_day
=="31","31","Not Yet Set" ))))))))))))))))))))))))))))))))
#View(add_day1)
#summarise count
day_Comcast_data_SUmm_plot_data <- add_day1 %>% group_by(Day1,Month1)
%>%
summarise(CountCustomerComplaints=sum(CountCustomerComplaints,na.rm=TR
UE))
#View(day_Comcast_data_SUmm_plot_data)

#graph
ggplot(data=day_Comcast_data_SUmm_plot_data) +
  geom_line(mapping=aes(x=Day1, y= CountCustomerComplaints, group = Month1,
                      color=Month1), size = 1) +
  labs(y="Customer Complaints (Count)", x="Days", title="Trend : Customer
Complaints Per Day")
```

```
####frequency_table for Customer.Complaint
# use lettercase Library
#frequency_table <- Comcast_data_final %>%
group_by(str_title_case(Customer.Complaint)) %>% summarise(Freq=n())
frequency_table <- Comcast_data_final %>%
group_by(str_to_upper(Customer.Complaint)) %>% summarise(Freq=n())
View(frequency_table)


####Which complaint types are maximum i.e., around internet, network issues, or
across any other domains######
f_tab_1 <- Comcast_data_final %>% group_by(Customer.Complaint,Received.Via)
%>%
summarise(CountCustomerComplaints=sum(CountCustomerComplaints,na.rm=TR
UE))
#View(f_tab_1)
#--sort---order by CountCustomerComplaints descending
df <-f_tab_1[order(-f_tab_1$CountCustomerComplaints),]
#View(df)
# Remove duplicate rows of the dataframe using carb variable
max_unq <- df[!duplicated(df$Received.Via), ]
View(max_unq)

########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#

New_Status_data <- transform(Comcast_data_final,New_Status = ifelse(Status
=="Closed","Closed",ifelse(Status =="Solved","Closed",ifelse(Status
=="Open","Open",ifelse(Status =="Pending","Open","No available status")))))
View(New_Status_data)

####Provide state wise status of complaints in a stacked bar chart. Use the
categorized variable from Q3. Provide insights on###

ggplot(data = New_Status_data, aes(x = State, y = CountCustomerComplaints, fill =
New_Status)) +
 geom_bar(stat="identity") + coord_flip() + labs(title = "state wise status of
complaints in a stacked bar chart",
                                                  y = "Customer Complaints (Count)", x = "All States",
fill = "Complaints Status")



#####Which state has the maximum complaints#####
state_tab_1 <- New_Status_data %>% group_by(State) %>%
summarise(Customer_Complaints=sum(CountCustomerComplaints,na.rm=TRUE))
#View(state_tab_1)
#--sort---order by CountCustomerComplaints descending
state_high_complaint_1 <-state_tab_1[order(-
state_tab_1$CountCustomerComplaints),]
View(state_high_complaint_1)


########Which state has the highest percentage of unresolved complaints
state_per_1 <- New_Status_data %>% group_by(State,New_Status) %>%
filter(New_Status =='Open') %>%
summarise(CountCustomerComplaints=sum(CountCustomerComplaints,na.rm=TR
UE))
#View(state_per_1)
#--sort---order by CountCustomerComplaints descending
df_unresolved <-state_per_1[order(-state_per_1$CountCustomerComplaints),]
View(df_unresolved)


########Provide the percentage of complaints resolved till date, which were
received through theInternet and customer care calls

resolv_1 <- New_Status_data %>%  filter(New_Status =='Closed') %>%
summarise(CountCustomerComplaints=sum(CountCustomerComplaints,na.rm=TR
UE))
#View(resolv_1)
#get all the total cases
all_cases_1 <- New_Status_data %>%
summarise(CountCustomerComplaints=sum(CountCustomerComplaints,na.rm=TR
UE))
#View(all_cases_1)
```

```
#percentage of resolved
per_resolved <-
(resolv_1$CountCustomerComplaints/all_cases_1$CountCustomerComplaints
*100)
View(per_resolved)
```