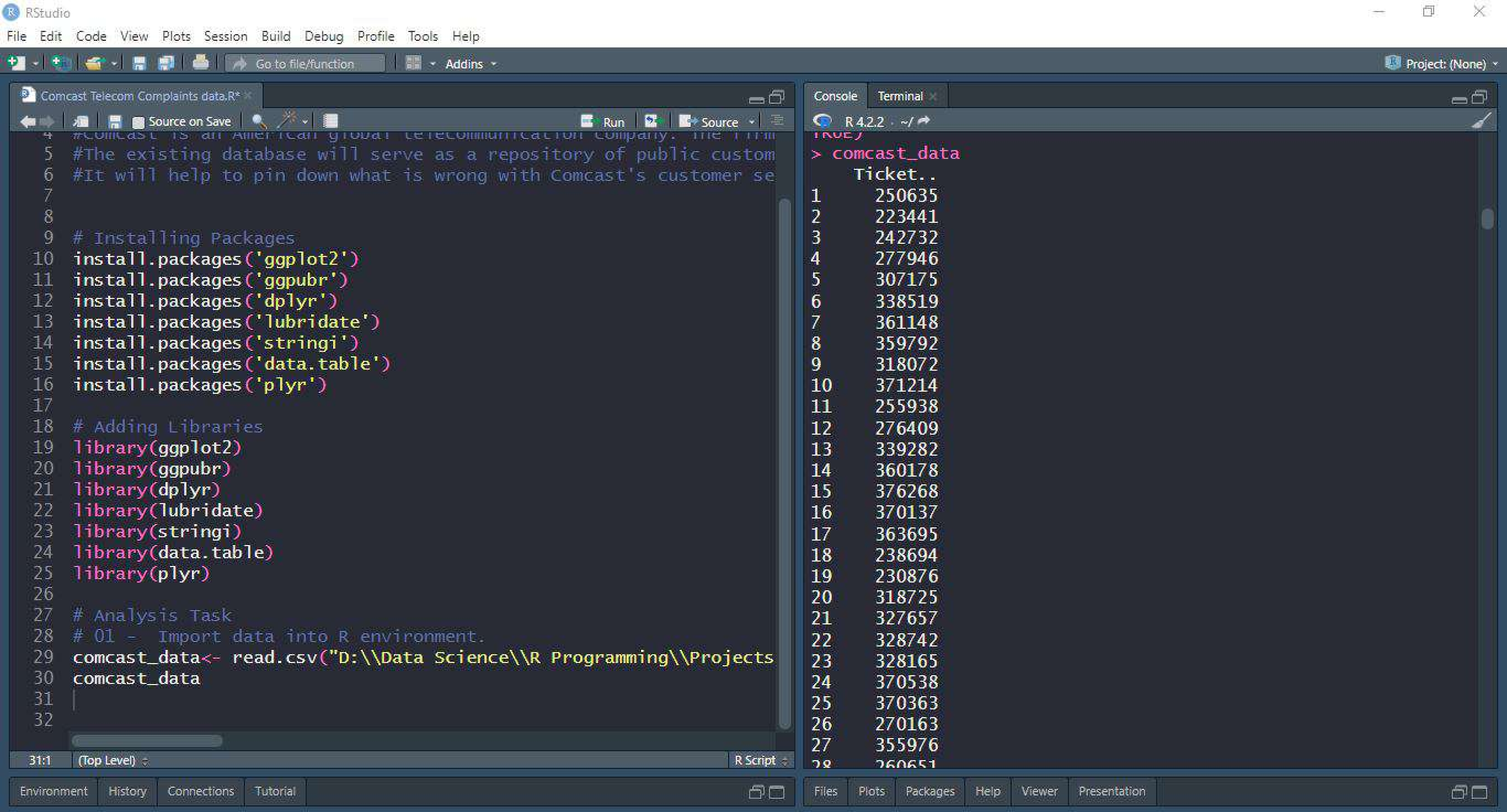
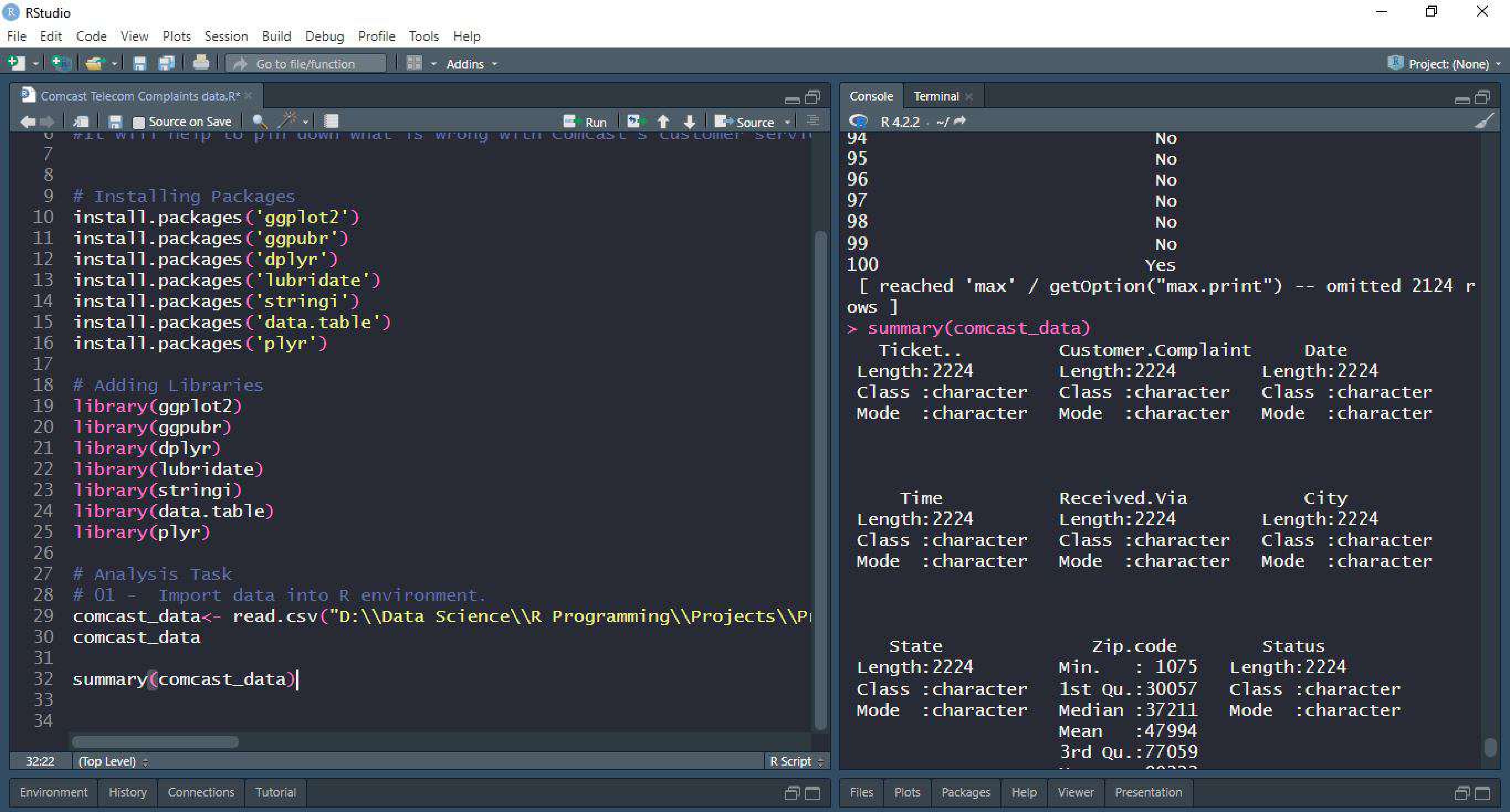

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Comcast Telecom Complaints data.R*
Source on Save
Run
Source
Console
Terminal
R 4.2.2 · ~/
31:1
(Top Level)
R Script
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation

```r
#Comcast is an American global telecommunication company. The firm
#The existing database will serve as a repository of public custom
#It will help to pin down what is wrong with Comcast's customer se


# Installing Packages
install.packages('ggplot2')
install.packages('ggpubr')
install.packages('dplyr')
install.packages('lubridate')
install.packages('stringi')
install.packages('data.table')
install.packages('plyr')

# Adding Libraries
library(ggplot2)
library(ggpubr)
library(dplyr)
library(lubridate)
library(stringi)
library(data.table)
library(plyr)

# Analysis Task
# 01 -  Import data into R environment.
comcast_data<- read.csv("D:\\Data Science\\R Programming\\Projects
comcast_data

```

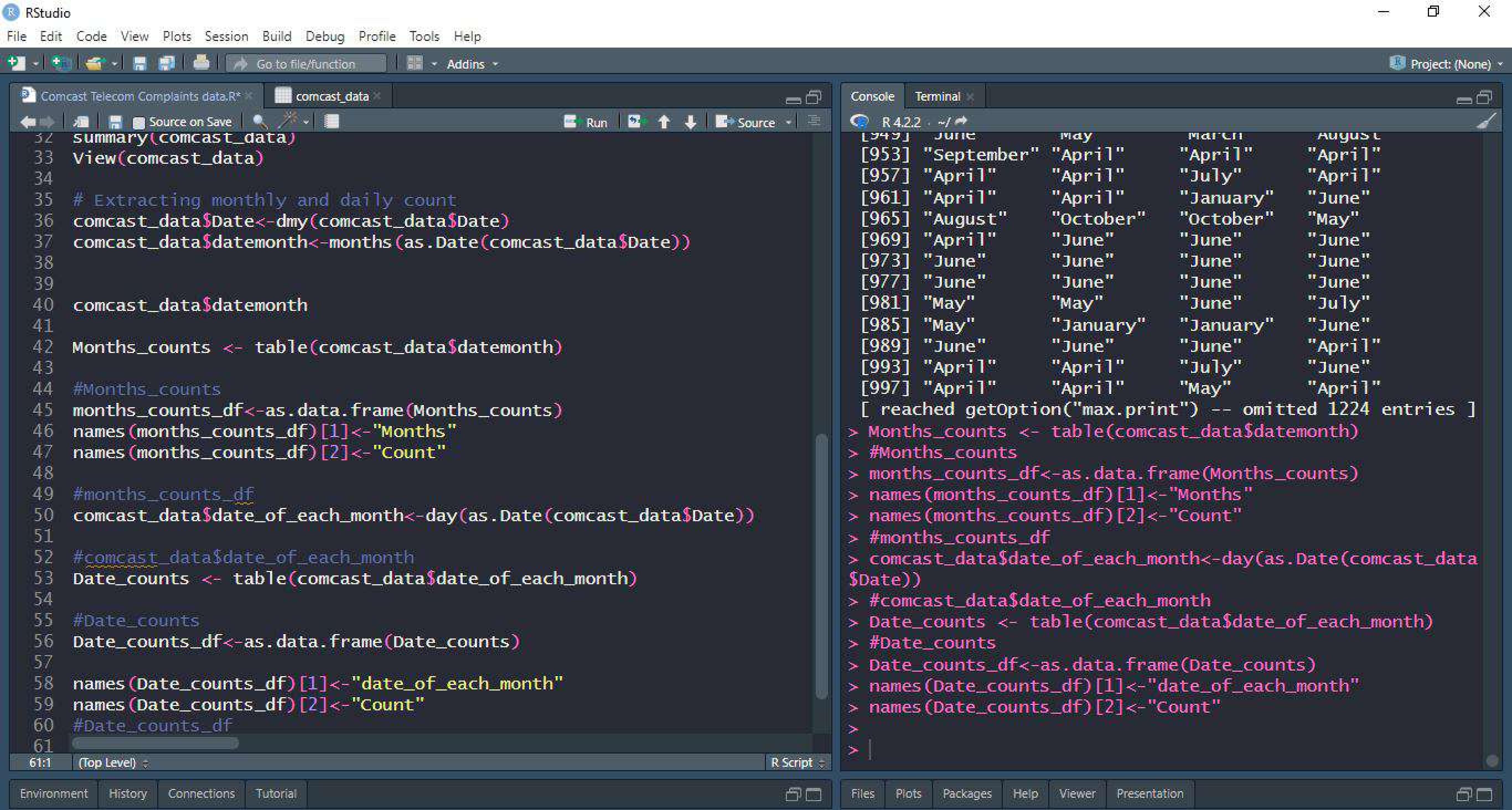
```
TRUE)
> comcast_data
   Ticket..
1    250635
2    223441
3    242732
4    277946
5    307175
6    338519
7    361148
8    359792
9    318072
10   371214
11   255938
12   276409
13   339282
14   360178
15   376268
16   370137
17   363695
18   238694
19   230876
20   318725
21   327657
22   328742
23   328165
24   370538
25   370363
26   270163
27   355976
28   260651
```



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Comcast Telecom Complaints data.R*
Source on Save
Run
Source
Console
Terminal
R 4.2.2 · ~/
32:22
(Top Level)
R Script
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation

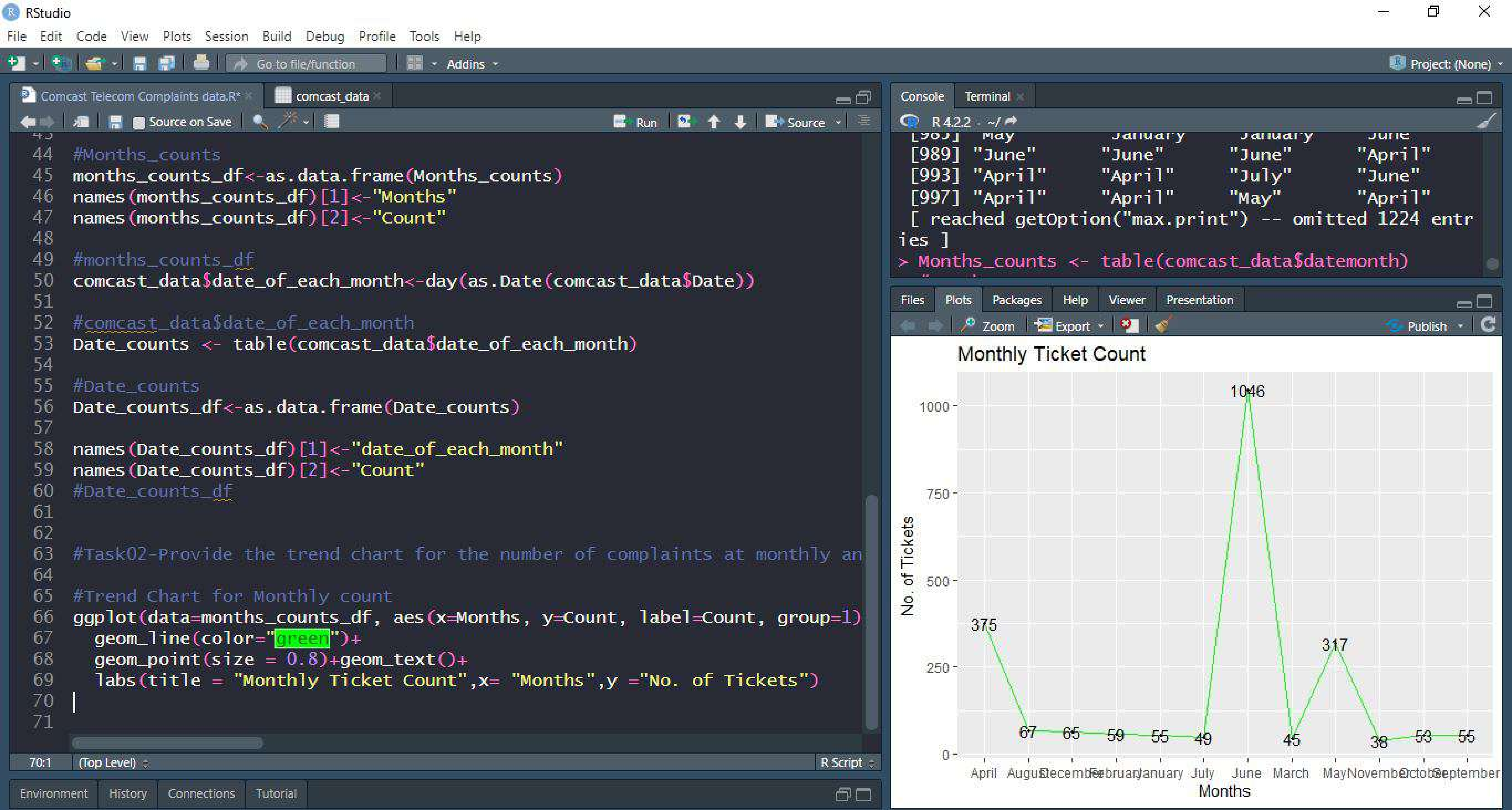
```r
#It will help to pin down what is wrong with Comcast's customer servi


# Installing Packages
install.packages('ggplot2')
install.packages('ggpubr')
install.packages('dplyr')
install.packages('lubridate')
install.packages('stringi')
install.packages('data.table')
install.packages('plyr')

# Adding Libraries
library(ggplot2)
library(ggpubr)
library(dplyr)
library(lubridate)
library(stringi)
library(data.table)
library(plyr)

# Analysis Task
# 01 -  Import data into R environment.
comcast_data<- read.csv("D:\\Data Science\\R Programming\\Projects\\Pr
comcast_data

summary(comcast_data)
```

```
94                         No
95                         No
96                         No
97                         No
98                         No
99                         No
100                       Yes
 [ reached 'max' / getOption("max.print") -- omitted 2124 r
ows ]
> summary(comcast_data)
   Ticket..          Customer.Complaint     Date          
 Length:2224        Length:2224        Length:2224       
 Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character  




     Time           Received.Via           City          
 Length:2224        Length:2224        Length:2224       
 Class :character   Class :character   Class :character  
 Mode  :character   Mode  :character   Mode  :character  




    State              Zip.code        Status         
 Length:2224        Min.   : 1075   Length:2224       
 Class :character   1st Qu.:30057   Class :character  
 Mode  :character   Median :37211   Mode  :character  
                    Mean   :47994                     
                    3rd Qu.:77059                     
```



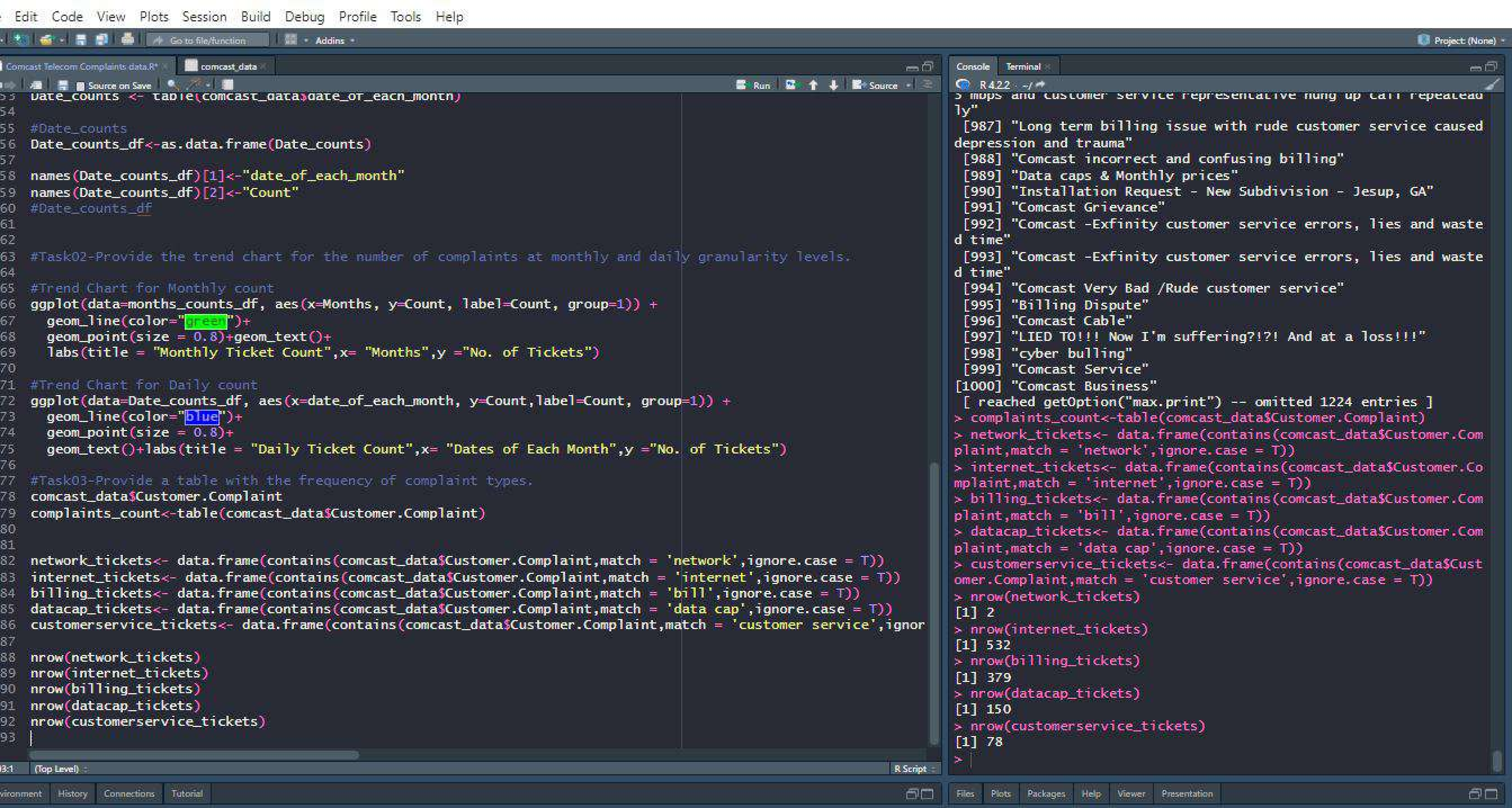
```
summary(comcast_data)
View(comcast_data)

# Extracting monthly and daily count
comcast_data$Date<-dmy(comcast_data$Date)
comcast_data$datemonth<-months(as.Date(comcast_data$Date))


comcast_data$datemonth

Months_counts <- table(comcast_data$datemonth)

#Months_counts
months_counts_df<-as.data.frame(Months_counts)
names(months_counts_df)[1]<-"Months"
names(months_counts_df)[2]<-"Count"

#months_counts_df
comcast_data$date_of_each_month<-day(as.Date(comcast_data$Date))

#comcast_data$date_of_each_month
Date_counts <- table(comcast_data$date_of_each_month)

#Date_counts
Date_counts_df<-as.data.frame(Date_counts)

names(Date_counts_df)[1]<-"date_of_each_month"
names(Date_counts_df)[2]<-"Count"
#Date_counts_df

```

```
[949] "June"      "May"       "March"     "August"
[953] "September" "April"     "April"     "April"
[957] "April"     "April"     "July"      "April"
[961] "April"     "April"     "January"   "June"
[965] "August"    "October"   "October"   "May"
[969] "April"     "June"      "June"      "June"
[973] "June"      "June"      "June"      "June"
[977] "June"      "June"      "June"      "June"
[981] "May"       "May"       "June"      "July"
[985] "May"       "January"   "January"   "June"
[989] "June"      "June"      "June"      "April"
[993] "April"     "April"     "July"      "June"
[997] "April"     "April"     "May"       "April"
 [ reached getOption("max.print") -- omitted 1224 entries ]
> Months_counts <- table(comcast_data$datemonth)
> #Months_counts
> months_counts_df<-as.data.frame(Months_counts)
> names(months_counts_df)[1]<-"Months"
> names(months_counts_df)[2]<-"Count"
> #months_counts_df
> comcast_data$date_of_each_month<-day(as.Date(comcast_data
$Date))
> #comcast_data$date_of_each_month
> Date_counts <- table(comcast_data$date_of_each_month)
> #Date_counts
> Date_counts_df<-as.data.frame(Date_counts)
> names(Date_counts_df)[1]<-"date_of_each_month"
> names(Date_counts_df)[2]<-"Count"
>
>
```



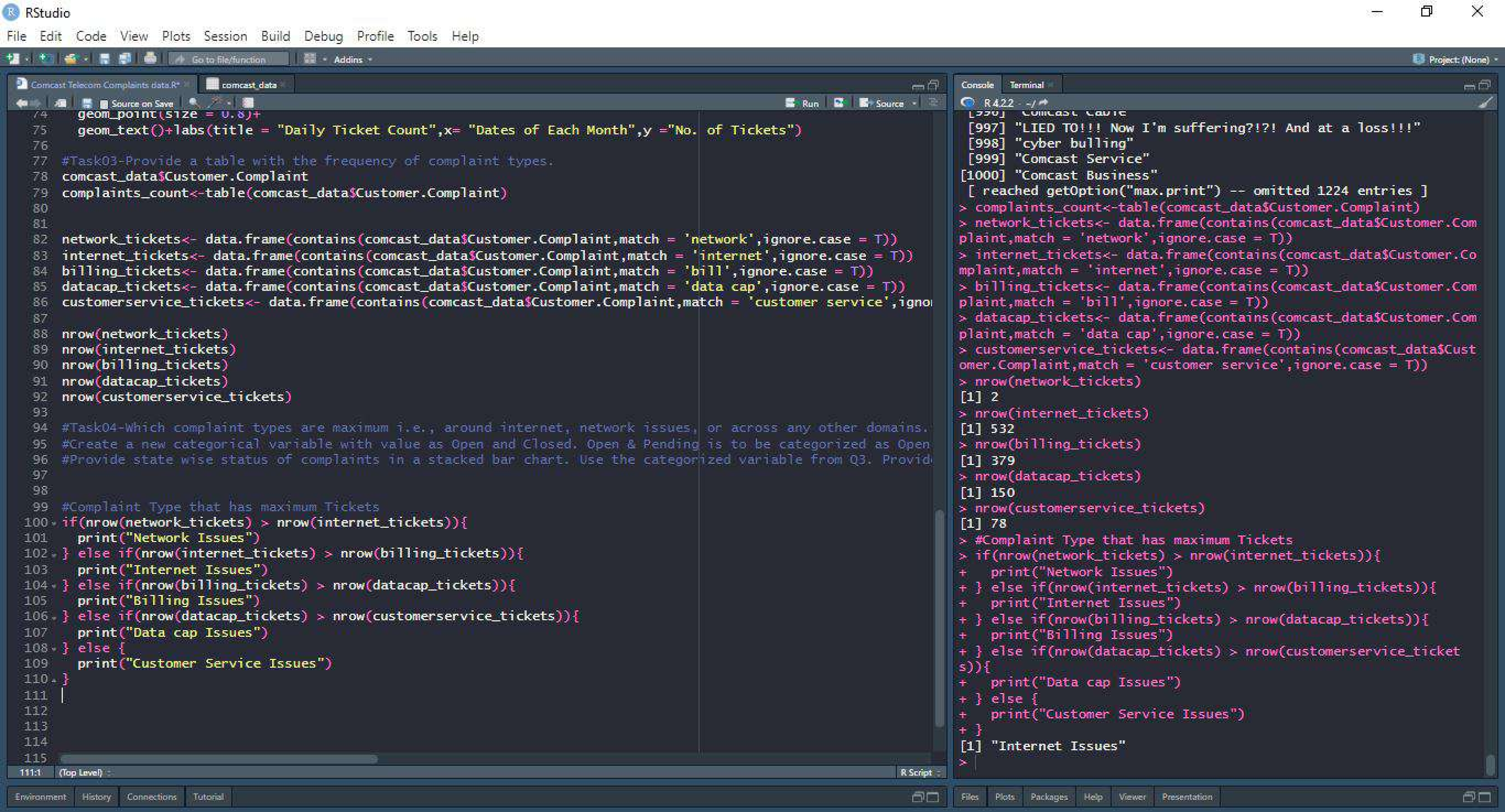
```
#Months_counts
months_counts_df<-as.data.frame(Months_counts)
names(months_counts_df)[1]<-"Months"
names(months_counts_df)[2]<-"Count"

#months_counts_df
comcast_data$date_of_each_month<-day(as.Date(comcast_data$Date))

#comcast_data$date_of_each_month
Date_counts <- table(comcast_data$date_of_each_month)

#Date_counts
Date_counts_df<-as.data.frame(Date_counts)

names(Date_counts_df)[1]<-"date_of_each_month"
names(Date_counts_df)[2]<-"Count"
#Date_counts_df



#Task02-Provide the trend chart for the number of complaints at monthly an

#Trend Chart for Monthly count
ggplot(data=months_counts_df, aes(x=Months, y=Count, label=Count, group=1)
  geom_line(color="green")+
  geom_point(size = 0.8)+geom_text()+
  labs(title = "Monthly Ticket Count",x= "Months",y ="No. of Tickets")

```

```
 [985] "May"          "January"      "January"      "June"
 [989] "June"         "June"         "June"         "April"
 [993] "April"        "April"        "July"         "June"
 [997] "April"        "April"        "May"          "April"
 [ reached getOption("max.print") -- omitted 1224 entr
ies ]
> Months_counts <- table(comcast_data$datemonth)
```



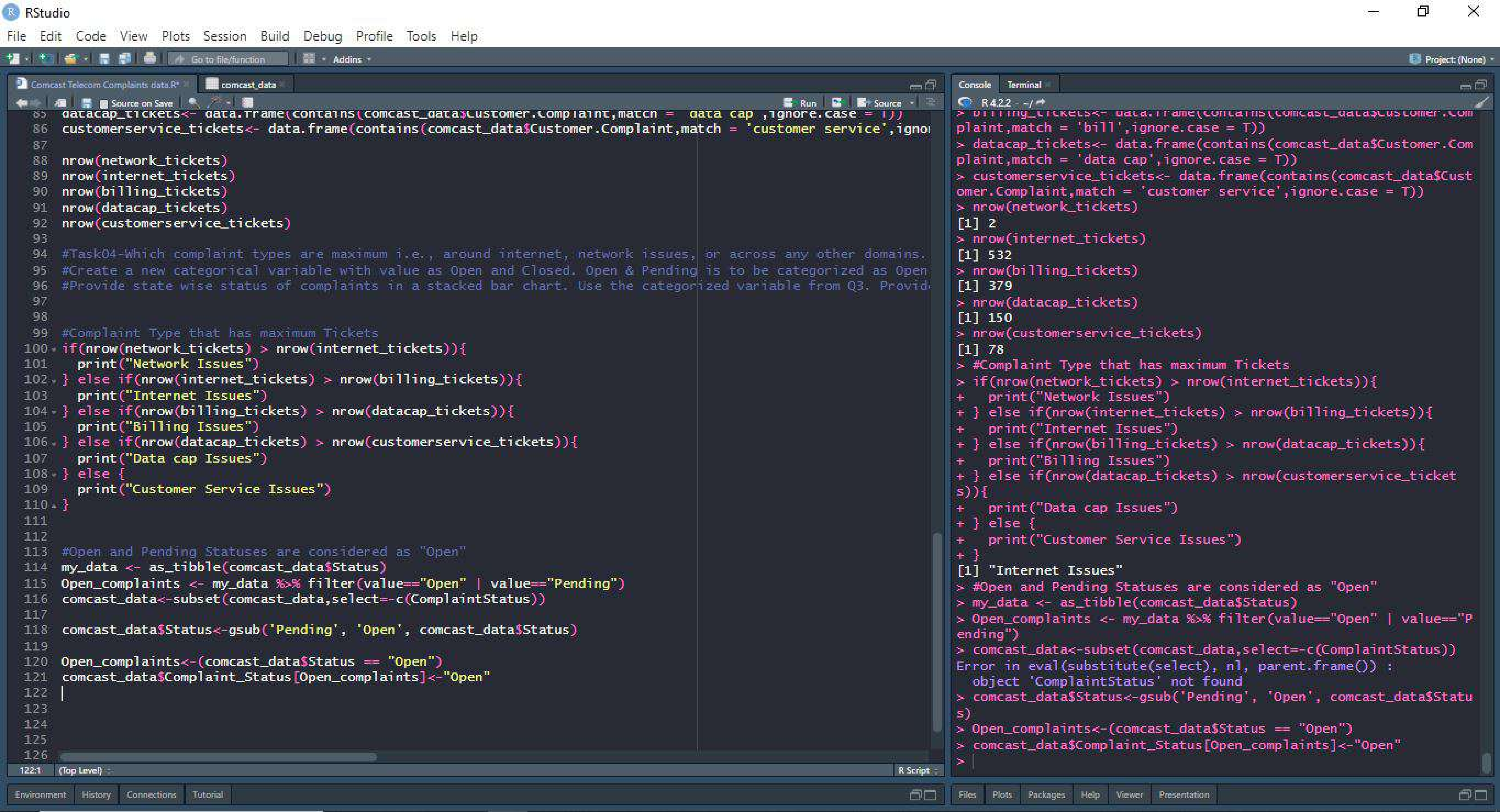
```r
Date_counts <- table(comcast_data$date_of_each_month)

#Date_counts
Date_counts_df<-as.data.frame(Date_counts)

names(Date_counts_df)[1]<-"date_of_each_month"
names(Date_counts_df)[2]<-"Count"
#Date_counts_df


#Task02-Provide the trend chart for the number of complaints at monthly and daily granularity levels.

#Trend Chart for Monthly count
ggplot(data=months_counts_df, aes(x=Months, y=Count, label=Count, group=1)) +
  geom_line(color="green")+
  geom_point(size = 0.8)+geom_text()+
  labs(title = "Monthly Ticket Count",x= "Months",y ="No. of Tickets")

#Trend Chart for Daily count
ggplot(data=Date_counts_df, aes(x=date_of_each_month, y=Count,label=Count, group=1)) +
  geom_line(color="blue")+
  geom_point(size = 0.8)+
  geom_text()+labs(title = "Daily Ticket Count",x= "Dates of Each Month",y ="No. of Tickets")

#Task03-Provide a table with the frequency of complaint types.
comcast_data$Customer.Complaint
complaints_count<-table(comcast_data$Customer.Complaint)


network_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'network',ignore.case = T))
internet_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'internet',ignore.case = T))
billing_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'bill',ignore.case = T))
datacap_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'data cap',ignore.case = T))
customerservice_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'customer service',ignor

nrow(network_tickets)
nrow(internet_tickets)
nrow(billing_tickets)
nrow(datacap_tickets)
nrow(customerservice_tickets)
```

```
s mops and customer service representative hung up call repeated
ly"
 [987] "Long term billing issue with rude customer service caused
depression and trauma"
 [988] "Comcast incorrect and confusing billing"
 [989] "Data caps & Monthly prices"
 [990] "Installation Request - New Subdivision - Jesup, GA"
 [991] "Comcast Grievance"
 [992] "Comcast -Exfinity customer service errors, lies and waste
d time"
 [993] "Comcast -Exfinity customer service errors, lies and waste
d time"
 [994] "Comcast Very Bad /Rude customer service"
 [995] "Billing Dispute"
 [996] "Comcast Cable"
 [997] "LIED TO!!! Now I'm suffering?!?! And at a loss!!!"
 [998] "cyber bulling"
 [999] "Comcast Service"
[1000] "Comcast Business"
 [ reached getOption("max.print") -- omitted 1224 entries ]
> complaints_count<-table(comcast_data$Customer.Complaint)
> network_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'network',ignore.case = T))
> internet_tickets<- data.frame(contains(comcast_data$Customer.Co
mplaint,match = 'internet',ignore.case = T))
> billing_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'bill',ignore.case = T))
> datacap_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'data cap',ignore.case = T))
> customerservice_tickets<- data.frame(contains(comcast_data$Cust
omer.Complaint,match = 'customer service',ignore.case = T))
> nrow(network_tickets)
[1] 2
> nrow(internet_tickets)
[1] 532
> nrow(billing_tickets)
[1] 379
> nrow(datacap_tickets)
[1] 150
> nrow(customerservice_tickets)
[1] 78
> 
```



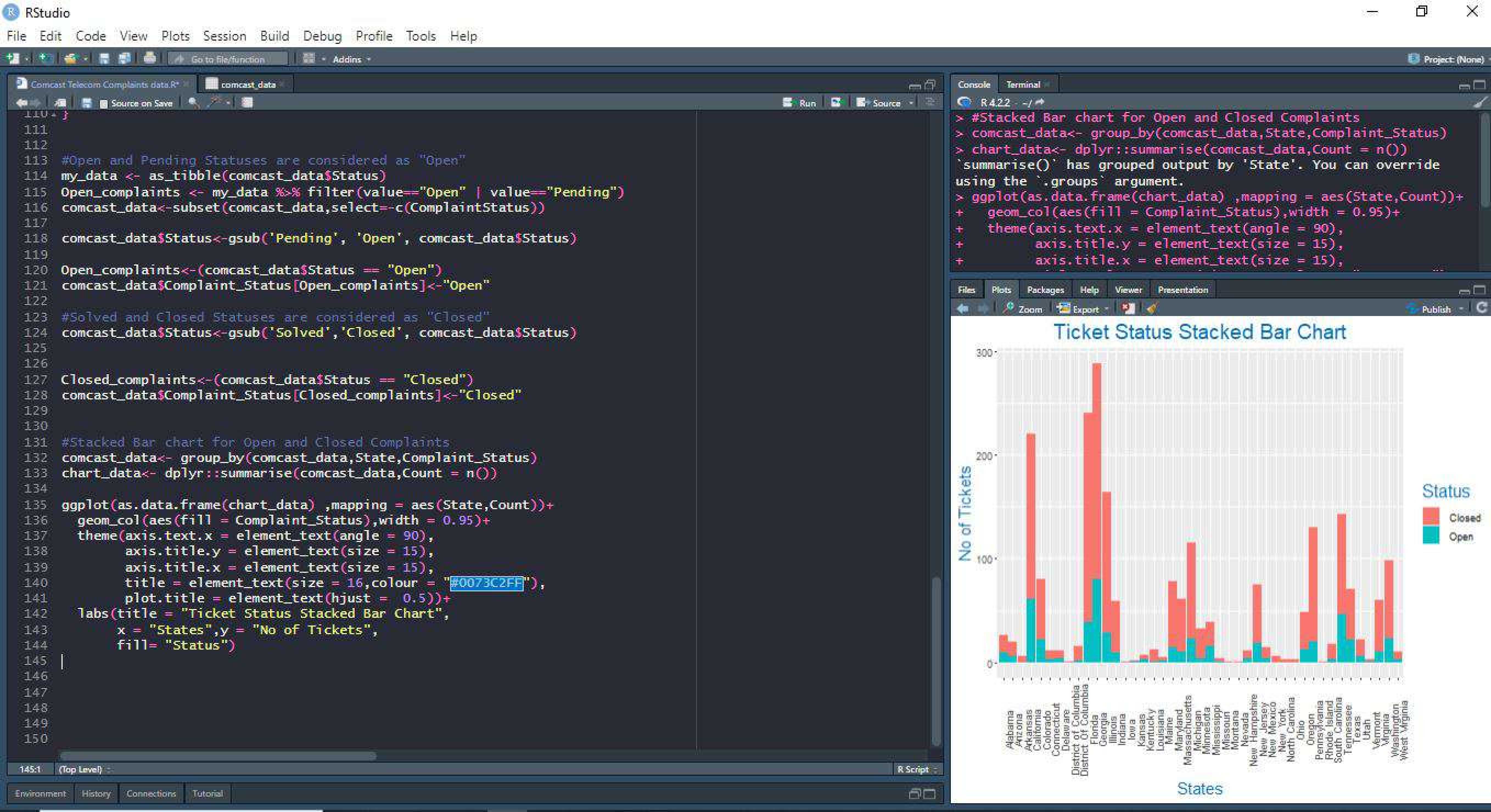
```r
  geom_point(size = 0.8)+
  geom_text()+labs(title = "Daily Ticket Count",x= "Dates of Each Month",y ="No. of Tickets")

#Task03-Provide a table with the frequency of complaint types.
comcast_data$Customer.Complaint
complaints_count<-table(comcast_data$Customer.Complaint)


network_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'network',ignore.case = T))
internet_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'internet',ignore.case = T))
billing_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'bill',ignore.case = T))
datacap_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'data cap',ignore.case = T))
customerservice_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'customer service',ignoi

nrow(network_tickets)
nrow(internet_tickets)
nrow(billing_tickets)
nrow(datacap_tickets)
nrow(customerservice_tickets)

#Task04-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
#Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provid


#Complaint Type that has maximum Tickets
if(nrow(network_tickets) > nrow(internet_tickets)){
  print("Network Issues")
} else if(nrow(internet_tickets) > nrow(billing_tickets)){
  print("Internet Issues")
} else if(nrow(billing_tickets) > nrow(datacap_tickets)){
  print("Billing Issues")
} else if(nrow(datacap_tickets) > nrow(customerservice_tickets)){
  print("Data cap Issues")
} else {
  print("Customer Service Issues")
}
```

```
 [996] "Comcast Cable"
 [997] "LIED TO!!! Now I'm suffering?!?! And at a loss!!!"
 [998] "cyber bulling"
 [999] "Comcast Service"
[1000] "Comcast Business"
 [ reached getOption("max.print") -- omitted 1224 entries ]
> complaints_count<-table(comcast_data$Customer.Complaint)
> network_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'network',ignore.case = T))
> internet_tickets<- data.frame(contains(comcast_data$Customer.Co
mplaint,match = 'internet',ignore.case = T))
> billing_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'bill',ignore.case = T))
> datacap_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'data cap',ignore.case = T))
> customerservice_tickets<- data.frame(contains(comcast_data$Cust
omer.Complaint,match = 'customer service',ignore.case = T))
> nrow(network_tickets)
[1] 2
> nrow(internet_tickets)
[1] 532
> nrow(billing_tickets)
[1] 379
> nrow(datacap_tickets)
[1] 150
> nrow(customerservice_tickets)
[1] 78
> #Complaint Type that has maximum Tickets
> if(nrow(network_tickets) > nrow(internet_tickets)){
+   print("Network Issues")
+ } else if(nrow(internet_tickets) > nrow(billing_tickets)){
+   print("Internet Issues")
+ } else if(nrow(billing_tickets) > nrow(datacap_tickets)){
+   print("Billing Issues")
+ } else if(nrow(datacap_tickets) > nrow(customerservice_ticket
s)){
+   print("Data cap Issues")
+ } else {
+   print("Customer Service Issues")
+ }
[1] "Internet Issues"
> 
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Comcast Telecom Complaints data.R*
comcast_data
Source on Save
Run
Source
Console
Terminal
R 4.2.2 · ~/
122:1
(Top Level)
R Script
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation

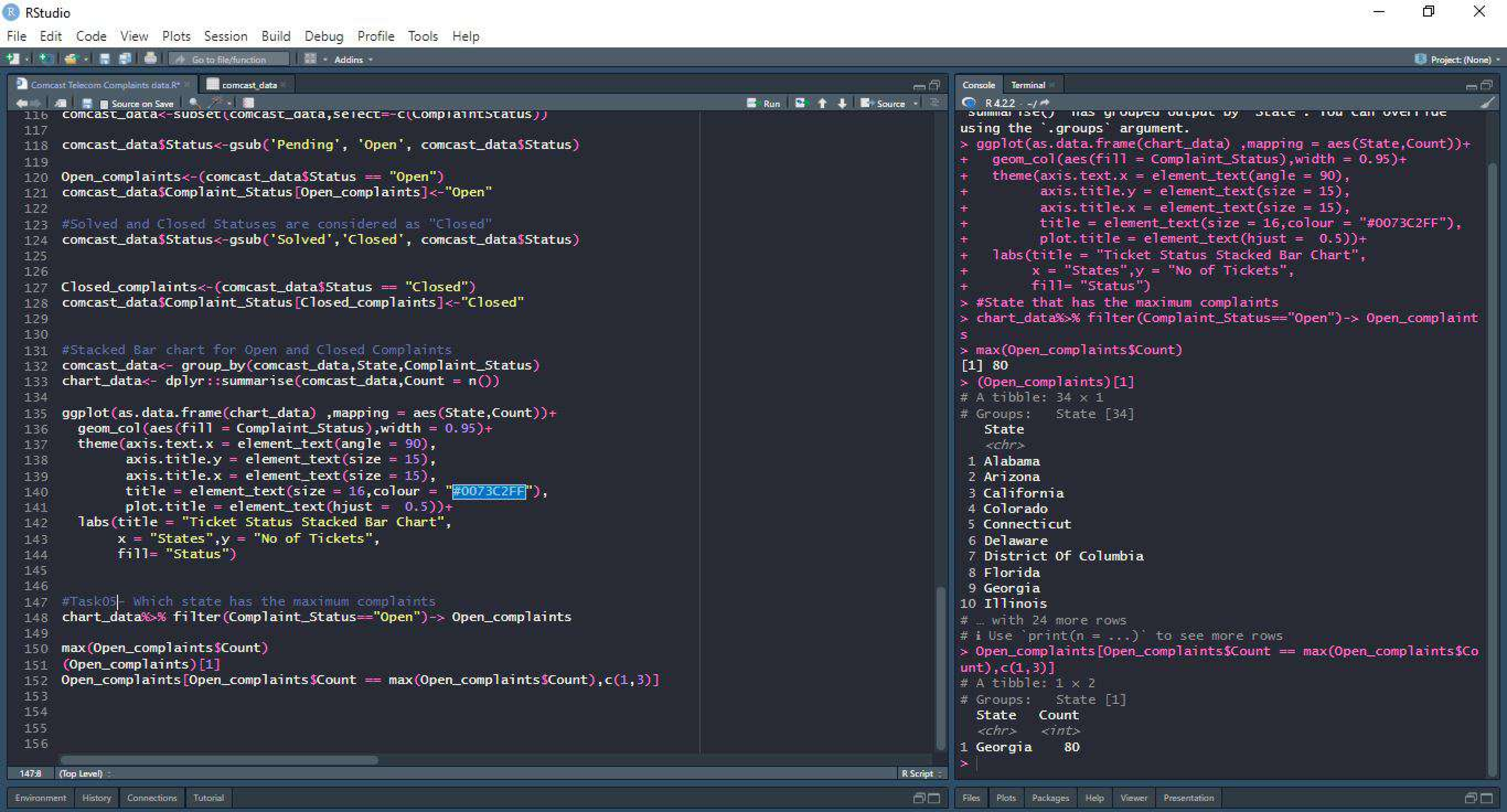
```
datacap_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'data cap',ignore.case = T))
customerservice_tickets<- data.frame(contains(comcast_data$Customer.Complaint,match = 'customer service',ignor

nrow(network_tickets)
nrow(internet_tickets)
nrow(billing_tickets)
nrow(datacap_tickets)
nrow(customerservice_tickets)

#Task04-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
#Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provid


#Complaint Type that has maximum Tickets
if(nrow(network_tickets) > nrow(internet_tickets)){
  print("Network Issues")
} else if(nrow(internet_tickets) > nrow(billing_tickets)){
  print("Internet Issues")
} else if(nrow(billing_tickets) > nrow(datacap_tickets)){
  print("Billing Issues")
} else if(nrow(datacap_tickets) > nrow(customerservice_tickets)){
  print("Data cap Issues")
} else {
  print("Customer Service Issues")
}


#Open and Pending Statuses are considered as "Open"
my_data <- as_tibble(comcast_data$Status)
Open_complaints <- my_data %>% filter(value=="Open" | value=="Pending")
comcast_data<-subset(comcast_data,select=-c(ComplaintStatus))

comcast_data$Status<-gsub('Pending', 'Open', comcast_data$Status)

Open_complaints<-(comcast_data$Status == "Open")
comcast_data$Complaint_Status[Open_complaints]<-"Open"
```

```
> billing_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'bill',ignore.case = T))
> datacap_tickets<- data.frame(contains(comcast_data$Customer.Com
plaint,match = 'data cap',ignore.case = T))
> customerservice_tickets<- data.frame(contains(comcast_data$Cust
omer.Complaint,match = 'customer service',ignore.case = T))
> nrow(network_tickets)
[1] 2
> nrow(internet_tickets)
[1] 532
> nrow(billing_tickets)
[1] 379
> nrow(datacap_tickets)
[1] 150
> nrow(customerservice_tickets)
[1] 78
> #Complaint Type that has maximum Tickets
> if(nrow(network_tickets) > nrow(internet_tickets)){
+   print("Network Issues")
+ } else if(nrow(internet_tickets) > nrow(billing_tickets)){
+   print("Internet Issues")
+ } else if(nrow(billing_tickets) > nrow(datacap_tickets)){
+   print("Billing Issues")
+ } else if(nrow(datacap_tickets) > nrow(customerservice_ticket
s)){
+   print("Data cap Issues")
+ } else {
+   print("Customer Service Issues")
+ }
[1] "Internet Issues"
> #Open and Pending Statuses are considered as "Open"
> my_data <- as_tibble(comcast_data$Status)
> Open_complaints <- my_data %>% filter(value=="Open" | value=="P
ending")
> comcast_data<-subset(comcast_data,select=-c(ComplaintStatus))
Error in eval(substitute(select), nl, parent.frame()) :
  object 'ComplaintStatus' not found
> comcast_data$Status<-gsub('Pending', 'Open', comcast_data$Statu
s)
> Open_complaints<-(comcast_data$Status == "Open")
> comcast_data$Complaint_Status[Open_complaints]<-"Open"
>
```



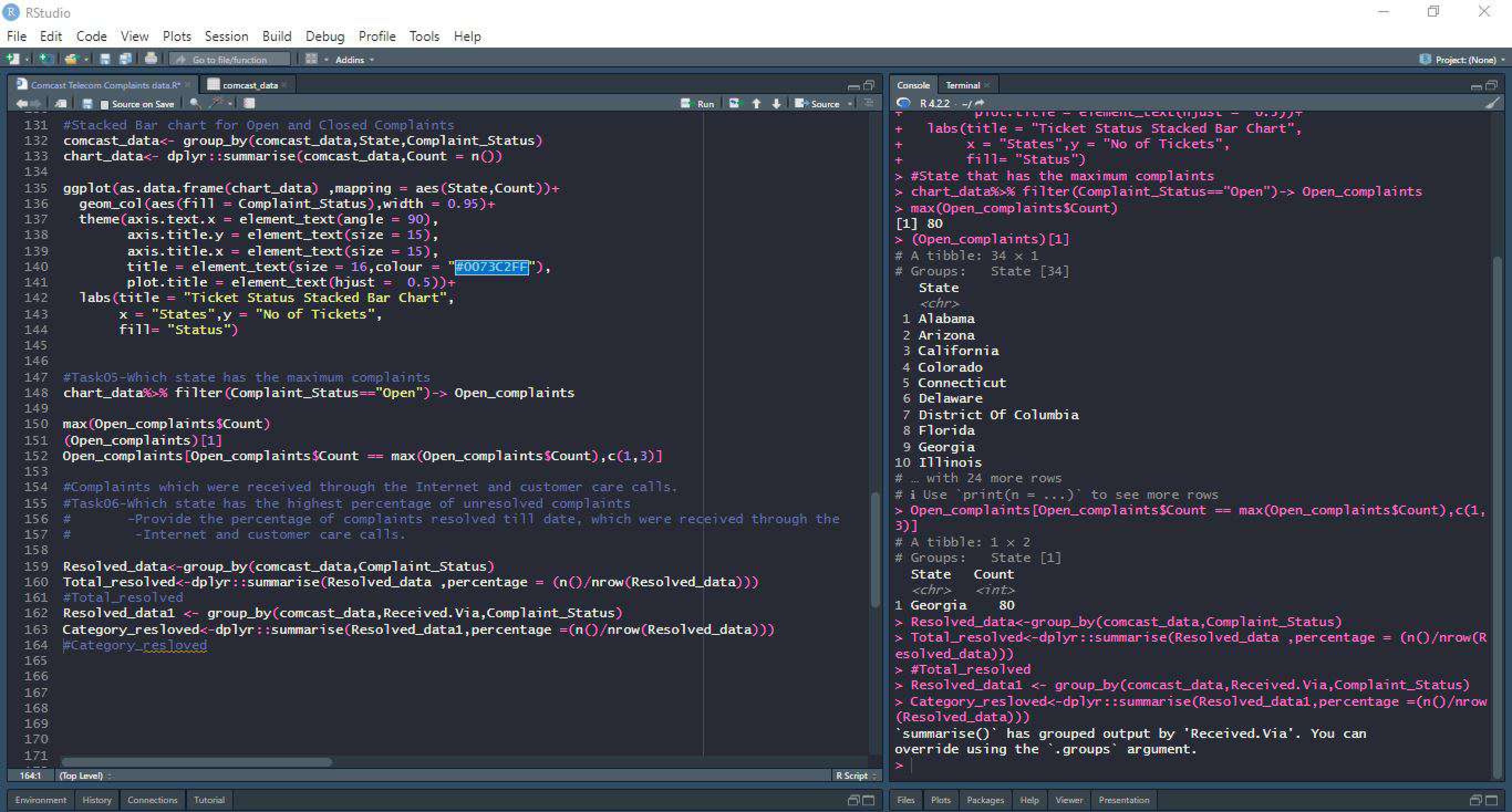
```r
}


#Open and Pending Statuses are considered as "Open"
my_data <- as_tibble(comcast_data$Status)
Open_complaints <- my_data %>% filter(value=="Open" | value=="Pending")
comcast_data<-subset(comcast_data,select=-c(ComplaintStatus))

comcast_data$Status<-gsub('Pending', 'Open', comcast_data$Status)

Open_complaints<-(comcast_data$Status == "Open")
comcast_data$Complaint_Status[Open_complaints]<-"Open"

#Solved and Closed Statuses are considered as "Closed"
comcast_data$Status<-gsub('Solved','Closed', comcast_data$Status)


Closed_complaints<-(comcast_data$Status == "Closed")
comcast_data$Complaint_Status[Closed_complaints]<-"Closed"


#Stacked Bar chart for Open and Closed Complaints
comcast_data<- group_by(comcast_data,State,Complaint_Status)
chart_data<- dplyr::summarise(comcast_data,Count = n())

ggplot(as.data.frame(chart_data) ,mapping = aes(State,Count))+
  geom_col(aes(fill = Complaint_Status),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")
```

```
> #Stacked Bar chart for Open and Closed Complaints
> comcast_data<- group_by(comcast_data,State,Complaint_Status)
> chart_data<- dplyr::summarise(comcast_data,Count = n())
`summarise()` has grouped output by 'State'. You can override
using the `.groups` argument.
> ggplot(as.data.frame(chart_data) ,mapping = aes(State,Count))+
+   geom_col(aes(fill = Complaint_Status),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),
```



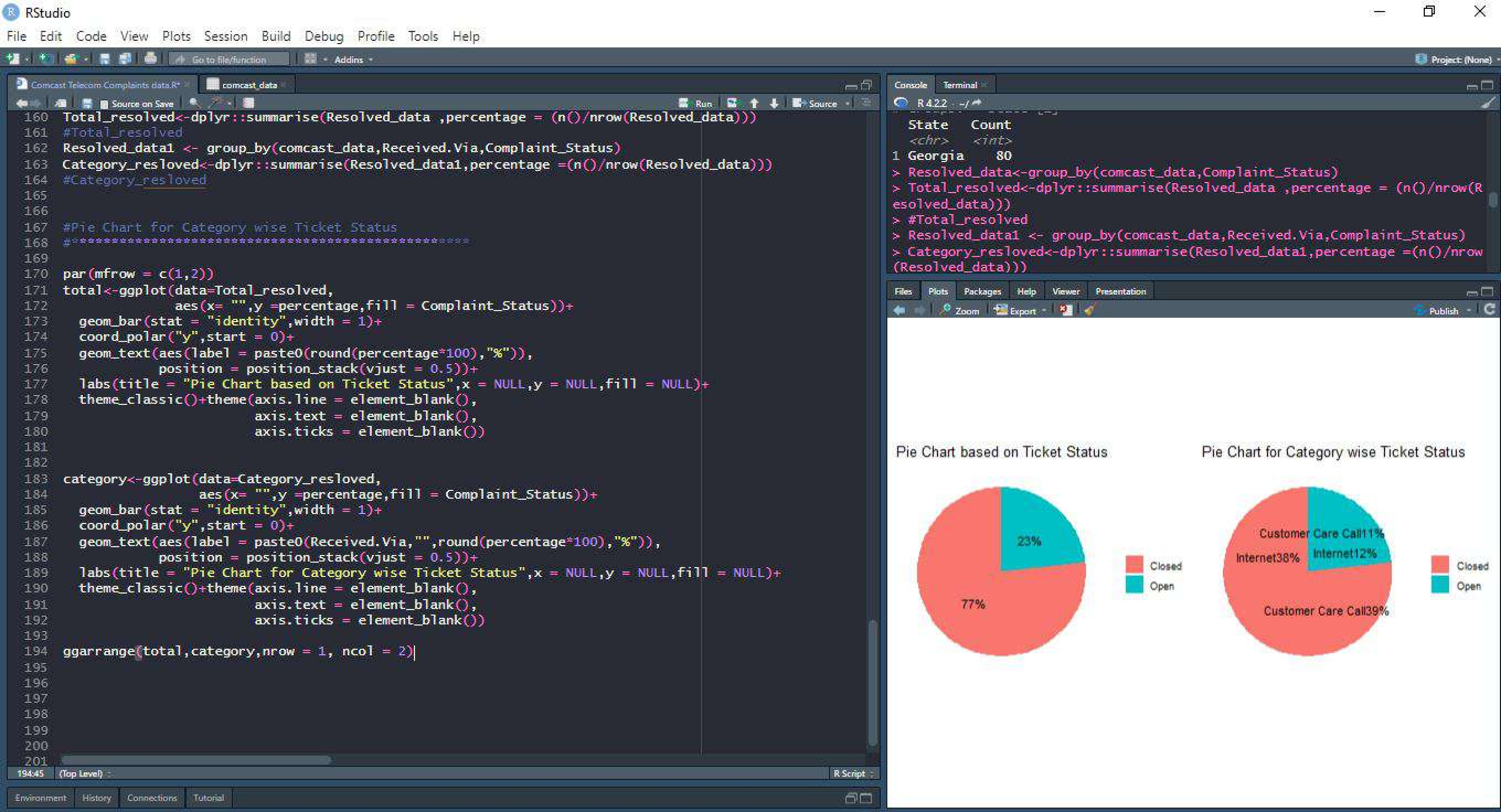
```r
comcast_data<-subset(comcast_data,select=-c(ComplaintStatus))

comcast_data$Status<-gsub('Pending', 'Open', comcast_data$Status)

Open_complaints<-(comcast_data$Status == "Open")
comcast_data$Complaint_Status[Open_complaints]<-"Open"

#Solved and Closed Statuses are considered as "Closed"
comcast_data$Status<-gsub('Solved','Closed', comcast_data$Status)


Closed_complaints<-(comcast_data$Status == "Closed")
comcast_data$Complaint_Status[Closed_complaints]<-"Closed"


#Stacked Bar chart for Open and Closed Complaints
comcast_data<- group_by(comcast_data,State,Complaint_Status)
chart_data<- dplyr::summarise(comcast_data,Count = n())

ggplot(as.data.frame(chart_data) ,mapping = aes(State,Count))+
  geom_col(aes(fill = Complaint_Status),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")


#Task05- Which state has the maximum complaints
chart_data%>% filter(Complaint_Status=="Open")-> Open_complaints

max(Open_complaints$Count)
(Open_complaints)[1]
Open_complaints[Open_complaints$Count == max(Open_complaints$Count),c(1,3)]
```

```
`summarise()` has grouped output by 'State'. You can override
using the `.groups` argument.
> ggplot(as.data.frame(chart_data) ,mapping = aes(State,Count))+
+   geom_col(aes(fill = Complaint_Status),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")
> #State that has the maximum complaints
> chart_data%>% filter(Complaint_Status=="Open")-> Open_complaint
s
> max(Open_complaints$Count)
[1] 80
> (Open_complaints)[1]
# A tibble: 34 × 1
# Groups:   State [34]
   State
   <chr>
 1 Alabama
 2 Arizona
 3 California
 4 Colorado
 5 Connecticut
 6 Delaware
 7 District Of Columbia
 8 Florida
 9 Georgia
10 Illinois
# … with 24 more rows
# ℹ Use `print(n = ...)` to see more rows
> Open_complaints[Open_complaints$Count == max(Open_complaints$Co
unt),c(1,3)]
# A tibble: 1 × 2
# Groups:   State [1]
  State   Count
  <chr>   <int>
1 Georgia    80
>
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Comcast Telecom Complaints data.R*
comcast_data
Source on Save
Run
Source
Console
Terminal
R 4.2.2 · ~/
164:1
(Top Level)
R Script
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation

```
#Stacked Bar chart for Open and Closed Complaints
comcast_data<- group_by(comcast_data,State,Complaint_Status)
chart_data<- dplyr::summarise(comcast_data,Count = n())

ggplot(as.data.frame(chart_data) ,mapping = aes(State,Count))+
  geom_col(aes(fill = Complaint_Status),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")


#Task05-Which state has the maximum complaints
chart_data%>% filter(Complaint_Status=="Open")-> Open_complaints

max(Open_complaints$Count)
(Open_complaints)[1]
Open_complaints[Open_complaints$Count == max(Open_complaints$Count),c(1,3)]

#Complaints which were received through the Internet and customer care calls.
#Task06-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.

Resolved_data<-group_by(comcast_data,Complaint_Status)
Total_resolved<-dplyr::summarise(Resolved_data ,percentage = (n()/nrow(Resolved_data)))
#Total_resolved
Resolved_data1 <- group_by(comcast_data,Received.Via,Complaint_Status)
Category_resloved<-dplyr::summarise(Resolved_data1,percentage =(n()/nrow(Resolved_data)))
#Category_resloved
```

```
+         plot.title = element_text(hjust =  0.5))+
+   labs(title = "Ticket Status Stacked Bar Chart",
+        x = "States",y = "No of Tickets",
+        fill= "Status")
> #State that has the maximum complaints
> chart_data%>% filter(Complaint_Status=="Open")-> Open_complaints
> max(Open_complaints$Count)
[1] 80
> (Open_complaints)[1]
# A tibble: 34 × 1
# Groups:   State [34]
   State
   <chr>
 1 Alabama
 2 Arizona
 3 California
 4 Colorado
 5 Connecticut
 6 Delaware
 7 District Of Columbia
 8 Florida
 9 Georgia
10 Illinois
# … with 24 more rows
# ℹ Use `print(n = ...)` to see more rows
> Open_complaints[Open_complaints$Count == max(Open_complaints$Count),c(1,
3)]
# A tibble: 1 × 2
# Groups:   State [1]
  State   Count
  <chr>   <int>
1 Georgia    80
> Resolved_data<-group_by(comcast_data,Complaint_Status)
> Total_resolved<-dplyr::summarise(Resolved_data ,percentage = (n()/nrow(R
esolved_data)))
> #Total_resolved
> Resolved_data1 <- group_by(comcast_data,Received.Via,Complaint_Status)
> Category_resloved<-dplyr::summarise(Resolved_data1,percentage =(n()/nrow
(Resolved_data)))
`summarise()` has grouped output by 'Received.Via'. You can
override using the `.groups` argument.
>
```



RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Comcast Telecom Complaints data.R*
comcast_data
Source on Save
Run
Source
Console
Terminal
R 4.2.2 · ~/
Files Plots Packages Help Viewer Presentation
Zoom
Export
Publish
194:45
(Top Level)
R Script
Environment History Connections Tutorial

```r
Total_resolved<-dplyr::summarise(Resolved_data ,percentage = (n()/nrow(Resolved_data)))
#Total_resolved
Resolved_data1 <- group_by(comcast_data,Received.Via,Complaint_Status)
Category_resloved<-dplyr::summarise(Resolved_data1,percentage =(n()/nrow(Resolved_data)))
#Category_resloved


#Pie Chart for Category wise Ticket Status
#*************************************************

par(mfrow = c(1,2))
total<-ggplot(data=Total_resolved,
              aes(x= "",y =percentage,fill = Complaint_Status))+
  geom_bar(stat = "identity",width = 1)+
  coord_polar("y",start = 0)+
  geom_text(aes(label = paste0(round(percentage*100),"%")),
          position = position_stack(vjust = 0.5))+
  labs(title = "Pie Chart based on Ticket Status",x = NULL,y = NULL,fill = NULL)+
  theme_classic()+theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank())


category<-ggplot(data=Category_resloved,
                aes(x= "",y =percentage,fill = Complaint_Status))+
  geom_bar(stat = "identity",width = 1)+
  coord_polar("y",start = 0)+
  geom_text(aes(label = paste0(Received.Via,"",round(percentage*100),"%")),
          position = position_stack(vjust = 0.5))+
  labs(title = "Pie Chart for Category wise Ticket Status",x = NULL,y = NULL,fill = NULL)+
  theme_classic()+theme(axis.line = element_blank(),
                        axis.text = element_blank(),
                        axis.ticks = element_blank())

ggarrange(total,category,nrow = 1, ncol = 2)
```

```
  State   Count
  <chr>   <int>
1 Georgia    80
> Resolved_data<-group_by(comcast_data,Complaint_Status)
> Total_resolved<-dplyr::summarise(Resolved_data ,percentage = (n()/nrow(R
esolved_data)))
> #Total_resolved
> Resolved_data1 <- group_by(comcast_data,Received.Via,Complaint_Status)
> Category_resloved<-dplyr::summarise(Resolved_data1,percentage =(n()/nrow
(Resolved_data)))
```