> library(stringi)

> library(lubridate)

Attaching package: ‘lubridate’

The following object is masked from ‘package:base’:

date

> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:lubridate’:

intersect, setdiff, union

The following objects are masked from ‘package:stats’:

filter, lag

The following objects are masked from ‘package:base’:

intersect, setdiff, setequal, union

> library(ggplot2)

> library(ggpubr)

Loading required package: magrittr

> comcast\_data<- read.csv("Comcast Telecom Complaints data.csv",header = TRUE)

> View(comcast\_data)

|  |
| --- |
| names(comcast\_data)<- stri\_replace\_all(regex = "\\.",replacement = "",names(comcast\_data))  > View(comcast\_data)  > comcast\_data$Date<- dmy(comcast\_data$Date)  > monthly\_count<- summarise(group\_by(comcast\_data,Month =as.integer(month(Date))),Count = n())  `summarise()` ungrouping output (override with `.groups` argument)  > daily\_count<- summarise(group\_by(comcast\_data,Date),Count =n())  `summarise()` ungrouping output (override with `.groups` argument)  >  > monthly\_count<-arrange(monthly\_count,Month)  >  > monthly\_count  # A tibble: 12 x 2  Month Count  *<int>* *<int>*  1 1 55  2 2 59  3 3 45  4 4 375  5 5 317  6 6 1046  7 7 49  8 8 67  9 9 55  10 10 53  11 11 38  12 12 65 |
|  |
| |  | | --- | | > | |

library(plotly)

Attaching package: ‘plotly’

The following object is masked from ‘package:ggplot2’:

last\_plot

The following object is masked from ‘package:stats’:

filter

The following object is masked from ‘package:graphics’:

layout

>

> # Create the ggplot objects for monthly and daily ticket counts

> monthly\_plot <- ggplot(data = monthly\_count, aes(Month, Count, label = Count)) +

+ geom\_line() +

+ geom\_point(size = 0.8) +

+ geom\_text() +

+ scale\_x\_continuous(breaks = monthly\_count$Month) +

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

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

|  |
| --- |
| daily\_plot <- ggplot(data = daily\_count, aes(as.POSIXct(Date), Count)) +  + geom\_line() +  + geom\_point(size = 0.8) +  + 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))  > daily\_count  # A tibble: 91 x 2  Date Count  *<date>* *<int>*  1 2015-01-04 18  2 2015-01-05 12  3 2015-01-06 25  4 2015-02-04 27  5 2015-02-05 7  6 2015-02-06 25  7 2015-03-04 15  8 2015-03-05 5  9 2015-03-06 25  10 2015-04-04 12  # … with 81 more rows  > daily\_plot |
|  |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | >   |  | | --- | | # Combine the ggplot plots into a single plotly plot  > combined\_plot <- subplot(monthly\_plot, daily\_plot, nrows = 2)  > combined\_plotly <- ggplotly(combined\_plot)  >  > # Display the plotly plot  > combined\_plotly | |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | >   |  | | --- | | # Complaint Type Processing  > network\_tickets<- contains(comcast\_data$CustomerComplaint,match = 'network',ignore.case = T)  > internet\_tickets<- contains(comcast\_data$CustomerComplaint,match = 'internet',ignore.case = T)  > billing\_tickets<- contains(comcast\_data$CustomerComplaint,match = 'bill',ignore.case = T)  > email\_tickets<- contains(comcast\_data$CustomerComplaint,match = 'email',ignore.case = T)  > charges\_ticket<- contains(comcast\_data$CustomerComplaint,match = 'charge',ignore.case = T)  >  > comcast\_data$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)  Billing Charges Email Internet Network Others  363 139 16 472 1 1233 | |  | | |  | | --- | | > | | | 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"  >  > library(plotly)  >  > comcast\_data <- group\_by(comcast\_data, State, ComplaintStatus)  > chart\_data <- summarise(comcast\_data, Count = n())  `summarise()` regrouping output by 'State' (override with `.groups` argument)  > ggplot\_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")  >  > # Convert ggplot chart to plotly  > plotly\_chart <- ggplotly(ggplot\_chart)  >  > # Display the plotly chart  > plotly\_chart | |  | | |  | | --- | | > | | | > #State having max unresolved complaints  > comcast\_data %>% filter(ComplaintStatus=='Open') %>% group\_by(State) %>% summarize(NumOfComplaints=n()) %>% arrange(desc(NumOfComplaints))  `summarise()` ungrouping output (override with `.groups` argument)  # A tibble: 34 x 2  State NumOfComplaints  *<fct>* *<int>*  1 Georgia 80  2 California 61  3 Tennessee 47  4 Florida 39  5 Illinois 29  6 Michigan 23  7 Washington 23  8 Colorado 22  9 Texas 22  10 Pennsylvania 20  # … with 24 more rows  >  >  > resolved\_data <- group\_by(comcast\_data,ComplaintStatus)  > total\_resloved<- summarise(resolved\_data ,percentage =(n()/nrow(resolved\_data)))  `summarise()` ungrouping output (override with `.groups` argument)  > resolved\_data <- group\_by(comcast\_data,ReceivedVia,ComplaintStatus)  > Category\_resloved<- summarise(resolved\_data ,percentage =(n()/nrow(resolved\_data)))  `summarise()` regrouping output by 'ReceivedVia' (override with `.groups` argument)  >  >  > par(mfrow = c(1,2))  > total<-ggplot(total\_resloved,  + aes(x= "",y =percentage,fill = ComplaintStatus))+  + 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))+  + theme\_classic()+theme(axis.line = element\_blank(),  + axis.text = element\_blank(),  + axis.ticks = element\_blank())  > library(ggpubr)  >  >  > # Pie Chart for Category wise Ticket Status  > category <- ggplot(Category\_resloved, aes(x = "", y = percentage, fill = ComplaintStatus)) +  + geom\_bar(stat = "identity", width = 1) +  + coord\_polar("y", start = 0) +  + geom\_text(aes(label = paste0(ReceivedVia, "-", round(percentage \* 100), "%")),  + position = position\_stack(vjust = 0.5)) +  + theme\_classic() +  + theme(axis.line = element\_blank(),  + axis.text = element\_blank(),  + axis.ticks = element\_blank())  >  > # Arrange and display the pie charts  > ggarrange(total, category, nrow = 1, ncol = 2)  > total  > category  > chart\_arrange <- ggarrange(total, category, nrow = 1, ncol = 2)  > chart\_arrange | |  | | |  | | --- | | > | | | | | |