|  |
| --- |
| ### Loading Libraries  library(dplyr)  library(tidyr)  library(ggplot2)  #install.packages('lubridate')  library(lubridate)  library(forcats)  install.packages('qcc')  library(qcc)  # Set working directory to the working folder  setwd("C:\\ecommerce\_dataset")  # Importing the eCommerce dataset into R studio  ecommerce\_data <- read.csv(file = 'ecommerce\_dataset.csv', header = T)  #Display the data  #print(ecommerce\_data)  # print number of columns  cat("Number of variables: ",ncol(ecommerce\_data))  # print number of rows  cat("Number of Observations: ",nrow(ecommerce\_data))  #Lets check the summary of the data  summary(ecommerce\_data)  #Check for missing values in each column  cols\_missing\_val = sum(is.na(ecommerce\_data))  if (cols\_missing\_val > 0) {  #Get the list of column names using the function colnames()  col\_names = colnames(ecommerce\_data)    for (col in col\_names) {  bool <- is.na(ecommerce\_data[[col]])  if (TRUE %in% bool) {  ## Checking for numeric column with missing values  cat("Handling the missing column: ",col,"\n")  ## Handle the missing columns by replacing it with mean  ecommerce\_data[[col]][is.na(ecommerce\_data[[col]])]<-mean(ecommerce\_data[[col]],na.rm=TRUE)    }  }  }  ######### Convert Date fields from character  #Convert review\_creation\_date to date type  ecommerce\_data['review\_creation\_date'] <- as.Date(ecommerce\_data$review\_creation\_date, "%m/%d/%Y %H:%M")  ecommerce\_data['review\_creation\_date']  #Convert shipping\_limit\_date to date type  ecommerce\_data[['shipping\_limit\_date']] <- as.Date(ecommerce\_data$shipping\_limit\_date, "%m/%d/%Y %H:%M")  ecommerce\_data[['shipping\_limit\_date']]  ###### Drop all the rows which has datetime NULL and we will use the date for  #majority of our analysis  ecommerce\_data <-  ecommerce\_data %>% drop\_na()  cat("Rows and Columns after dropping NULL values: ", nrow(ecommerce\_data) , "X", ncol(ecommerce\_data))  #Note: We still have more than 33K rows which is good for our analysis  #Adding a new column to check the difference between shipping limite date and review creation date  ecommerce\_data['shipping\_limit\_creation\_difference'] = ecommerce\_data['review\_creation\_date']-  ecommerce\_data['shipping\_limit\_date']  ecommerce\_data['shipping\_limit\_creation\_difference']  #Convert shipping\_limit\_date to day\_of\_week  ecommerce\_data['shipping\_limit\_day\_of\_week'] = weekdays(ecommerce\_data$shipping\_limit\_date)  ecommerce\_data['shipping\_limit\_day\_of\_week']  #Lets check the summary of the data again  summary(ecommerce\_data)  ## Plotting Graphs Now  #########################################  #1. Box Plot: To see the price range in each of the product categories  # Give the chart file a name.  png(file = "1-boxplot.png")  # Plot the chart.  ggplot(data=ecommerce\_data, mapping=aes(x=product\_category\_name, y=price))+  geom\_boxplot(colour="blue")+  ggtitle("eCommerce Product")  # Save the file.  dev.off()  #########################################  #2. Pie chart to see the contribution of Payment Options  # Give the chart file a name.  png(file = "2-Pie Chart.jpg")  #Summarize the dataset by Payment options  df2.piechart <-  ecommerce\_data %>%  group\_by(payment\_type) %>%  summarise(payment\_value = sum(payment\_value))  #Plot Piechart  ggplot(df2.piechart, aes(x="", y=payment\_value, fill=payment\_type)) +  geom\_bar(stat="identity", width=1) +  coord\_polar("y", start=0)+  ggtitle("Payment Method Contribution")  # Save the file.  dev.off()  #####################################  #3. Histogram of Reviews  png(file = "3-Histogram.jpg")  #Will check how many each ratings are given to the produts  ggplot(ecommerce\_data, aes(x = review\_score)) +  geom\_histogram(colour = "red",  lwd = 0.75,  linetype = 1,  position = "identity")+  ggtitle("Review Score Count")  # Save the file.  dev.off()  #####################################  #4. Scatter Plot  png(file = "4-Scatter.jpg")  #Will check how many each ratings are given to the produts  ggplot(ecommerce\_data, aes(x=product\_width\_cm, y=freight\_value)) +  geom\_point(size=2, shape=23, color="pink") +  ggtitle("Scatter Plot")  # Save the file.  dev.off()  summary(ecommerce\_data)  #####################################  #5. Line Plot  png(file = "5-Line.jpg")  print("Assuming sale day and shipment limit day of the week is same")  df2.linechart <-  ecommerce\_data %>%  group\_by(shipping\_limit\_day\_of\_week) %>%  summarise(payment\_value = sum(payment\_value))  #Will check how many each ratings are given to the produts  ggplot(data=df2.linechart, aes(x=shipping\_limit\_day\_of\_week, y=payment\_value, group=1)) +  geom\_line(color="green")+  geom\_point() +  ggtitle("Sales by Day of the Week")  # Save the file.  dev.off()  #####################################  #6. Top selling Category Plot - Pareto Analysis  png(file = "6-Top Categories.jpg")  df2.pareto <-  ecommerce\_data %>%  group\_by(product\_category\_name) %>%  summarise(payment\_value = sum(payment\_value))  pareto.chart(df2.pareto$payment\_value,  main='Pareto Chart for Top Business Categories',  col=heat.colors(length(df2.pareto$payment\_value)))  # Save the file.  dev.off()  #####################################  #####################################  #7. Scatter Plot - Difference in days of Sale & delivery v Review rating  png(file = "7-Scatter.jpg")  #Will check how many each ratings are given to the produts  ggplot(ecommerce\_data, aes(x=review\_score, y=shipping\_limit\_creation\_difference)) +  geom\_point(size=2, shape=23, color="red") +  ggtitle("Scatter Plot")  # Save the file.  dev.off()  ##################################### |