Final project

Padmaja gorijala

2023-12-14

Recording:https://cmich.webex.com/webappng/sites/cmich/recording/8d0df0e57d12103cbdfe0a1e9d02d160/playback

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

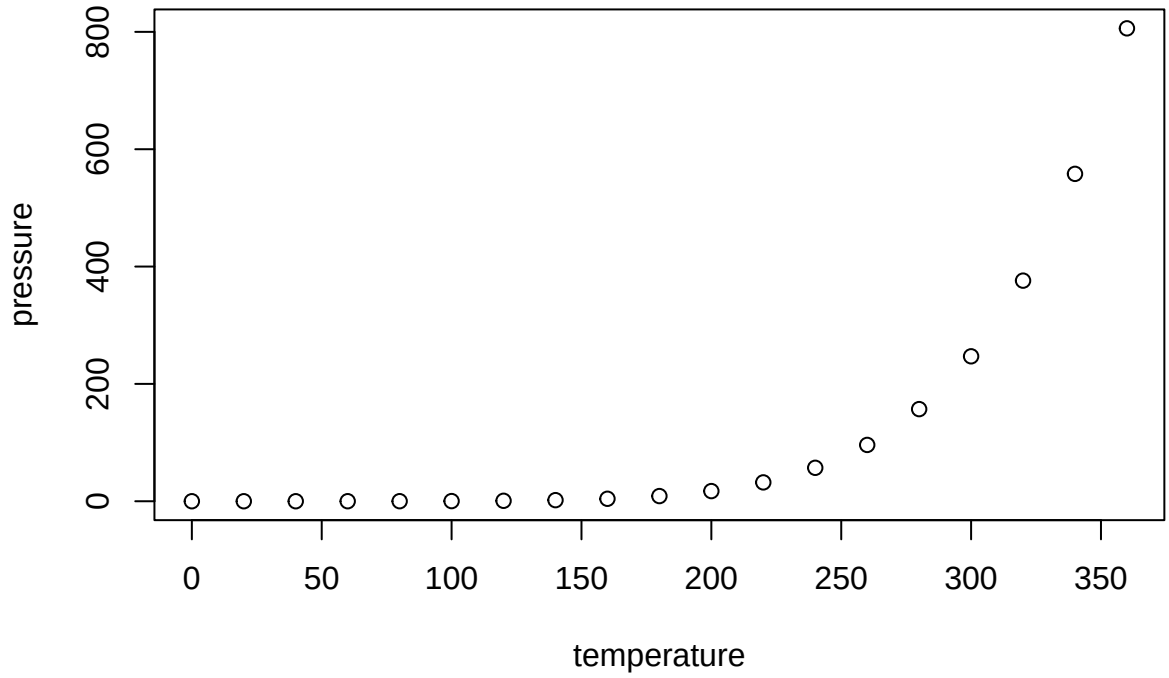
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library("ggplot2")
library("readxl")
```

Reading Sales Order data from the Sales Order Excel Sheet

```
US_Regional_Sales_Data <- read_excel("C:/Users/gorij1p/Downloads/US_Regional_Sales_Data.xlsx",
                                     sheet = "Sales Orders Sheet")
```

Reading Customer data from the Customers Excel Sheet

```
us_customer_details <- read_excel("C:/Users/gorij1p/Downloads/US_Regional_Sales_Data.xlsx",
                                  sheet = "Customers Sheet")
```

Reading Store Locations data from Store Locations Sheet

```
us_store_details <- read_excel("C:/Users/gorij1p/Downloads/US_Regional_Sales_Data.xlsx",
                               sheet = "Store Locations Sheet")
```

#Joining Sales Order Data and the customer # data with the CustomerID column

```
us_regional_customer_sales_data <- inner_join(US_Regional_Sales_Data,us_customer_details,by="_CustomerI
```

Joining combined Sales Order Data and the customer

data with the store details based on StoreID column

```
us_regional_customer_store_sales_data <- inner_join(us_regional_customer_sales_data,us_store_details,
                                                    by="_StoreID")
```

#Save the Final data in the R data format, # so that it can be read later for further use.

```
saveRDS(us_regional_customer_store_sales_data,file = "C:\\Users\\gorij1p\\Downloads\\data\\final_data.r
```

Write the final data back to a CSV file.

```
write.csv(us_regional_customer_store_sales_data,file = "C:\\Users\\gorij1p\\Downloads\\data\\cleaneddat
```

Prints the structure of the data

```
str(us_regional_customer_store_sales_data)
```

```
## tibble [7,991 x 31] (S3: tbl_df/tbl/data.frame)
##  $ OrderNumber     : chr [1:7991] "SO - 000101" "SO - 000102" "SO - 000103" "SO - 000104" ...
##  $ Sales Channel   : chr [1:7991] "In-Store" "Online" "Distributor" "Wholesale" ...
##  $ WarehouseCode   : chr [1:7991] "WARE-UHY1004" "WARE-NMK1003" "WARE-UHY1004" "WARE-NMK1003" ...
##  $ ProcuredDate    : POSIXct[1:7991], format: "2017-12-31" "2017-12-31" ...
##  $ OrderDate       : POSIXct[1:7991], format: "2018-05-31" "2018-05-31" ...
##  $ ShipDate        : POSIXct[1:7991], format: "2018-06-14" "2018-06-22" ...
##  $ DeliveryDate    : POSIXct[1:7991], format: "2018-06-19" "2018-07-02" ...
##  $ CurrencyCode    : chr [1:7991] "USD" "USD" "USD" "USD" ...
##  $ _SalesTeamID    : num [1:7991] 6 14 21 28 22 12 10 6 4 10 ...
##  $ _CustomerID     : num [1:7991] 15 20 16 48 49 21 14 9 9 33 ...
##  $ _StoreID        : num [1:7991] 259 196 213 107 111 285 6 280 299 261 ...
##  $ _ProductID      : num [1:7991] 12 27 16 23 26 1 5 46 47 13 ...
##  $ Order Quantity  : num [1:7991] 5 3 1 8 8 5 4 5 4 8 ...
##  $ Discount Applied: num [1:7991] 0.075 0.075 0.05 0.075 0.1 0.05 0.15 0.05 0.3 0.05 ...
##  $ Unit Price      : num [1:7991] 1963 3940 1776 2325 1822 ...
##  $ Unit Cost       : num [1:7991] 1001 3349 781 1465 1476 ...
##  $ Customer Names  : chr [1:7991] "Rochester Ltd" "Pacific Ltd" "3LAB, Ltd" "Fenwal, Corp" ...
##  $ City Name       : chr [1:7991] "Babylon (Town)" "Overland Park" "Ann Arbor" "New Haven" ...
##  $ County          : chr [1:7991] "Suffolk County" "Johnson County" "Washtenaw County" "New Haven Cou
##  $ StateCode       : chr [1:7991] "NY" "KS" "MI" "CT" ...
##  $ State           : chr [1:7991] "New York" "Kansas" "Michigan" "Connecticut" ...
##  $ Type            : chr [1:7991] "Town" "City" "City" "City" ...
##  $ Latitude        : num [1:7991] 40.6 39 42.3 41.3 41.6 ...
##  $ Longitude       : num [1:7991] -73.3 -94.7 -83.7 -72.9 -73.1 ...
##  $ AreaCode        : chr [1:7991] "631" "913" "734" "203" ...
##  $ Population      : num [1:7991] 213776 186515 117070 130322 108802 ...
##  $ Household Income: num [1:7991] 68789 74830 47179 49771 40213 ...
##  $ Median Income   : num [1:7991] 80327 72463 55990 37192 40467 ...
##  $ Land Area       : num [1:7991] 1.35e+08 1.95e+08 7.27e+07 4.84e+07 7.39e+07 ...
##  $ Water Area      : num [1:7991] 1.60e+08 1.31e+06 2.25e+06 3.74e+06 1.09e+06 ...
##  $ Time Zone       : chr [1:7991] "America/New York" "America/Chicago" "America/Detroit" "America/Ne
```

Removing the NA rows in the data

```
us_regional_customer_store_sales_data <- na.omit(us_regional_customer_store_sales_data)
```

Replacing the column name with space to underscore("_")

```
colnames(us_regional_customer_store_sales_data) <- sub(" ", "_", colnames(us_regional_customer_store_sa
```

Converting the string data format to the Date data format for the Order Date field

```
us_regional_customer_store_sales_data$OrderDate<-as.Date(us_regional_customer_store_sales_data$OrderDat
```

Converting the string data format to the Date data format for the Procured Date field

```
us_regional_customer_store_sales_data$ProcuredDate <- as.Date(us_regional_customer_store_sales_data$Proc
```

Splitting up the month from the order and storing it in the month column

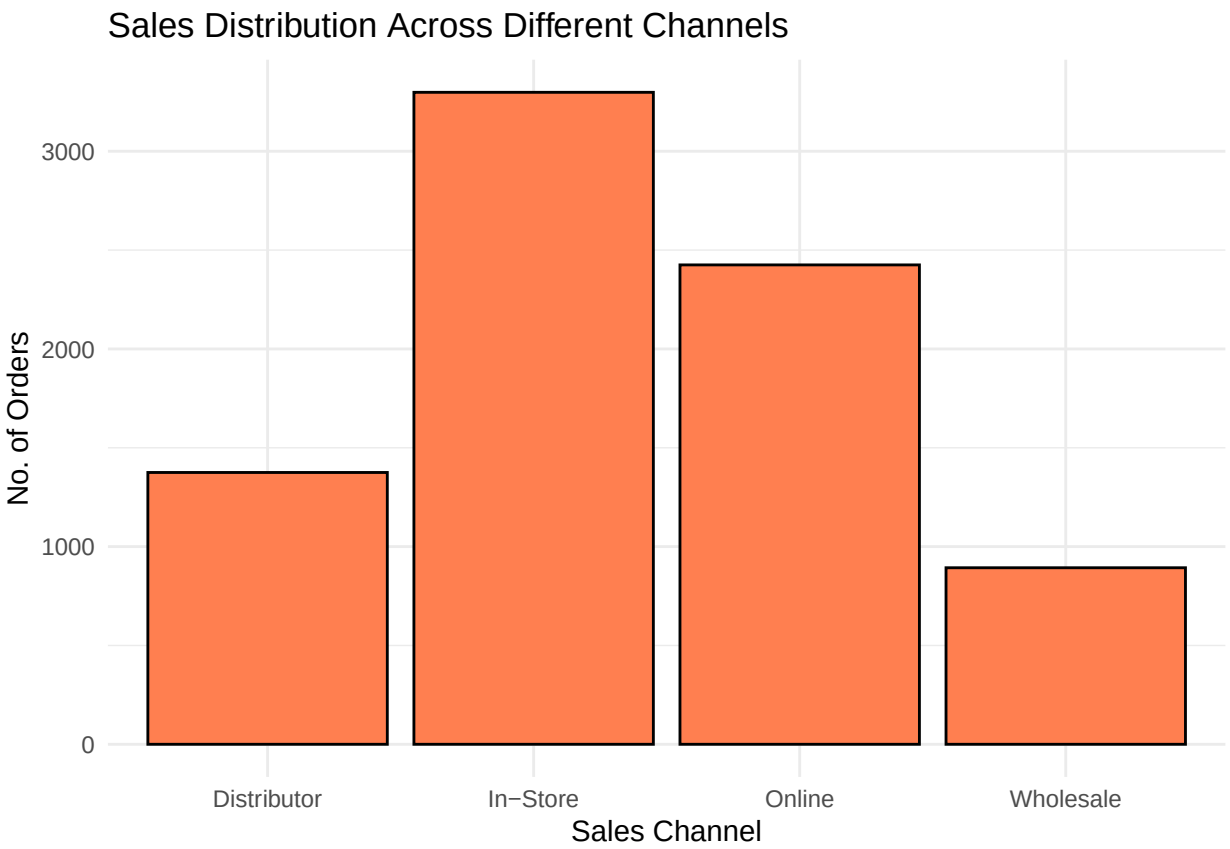
```
us_regional_customer_store_sales_data$Month <- format(us_regional_customer_store_sales_data$OrderDate,
```

Calculating the overall price based on quantity, unit price and discount

```
us_regional_customer_store_sales_data$overall_price <- (us_regional_customer_store_sales_data$Order_Quan
                                                        (us_regional_customer_store_sales_data$Unit_Pr
```

Barchart for Sales Distribution Channel and No. of Order

```
ggplot(us_regional_customer_store_sales_data, aes(x = Sales_Channel)) +
  geom_bar(fill = "coral", color = "black") +
  theme_minimal() +
  labs(title = "Sales Distribution Across Different Channels", x = "Sales Channel", y = "No. of Orders"
```

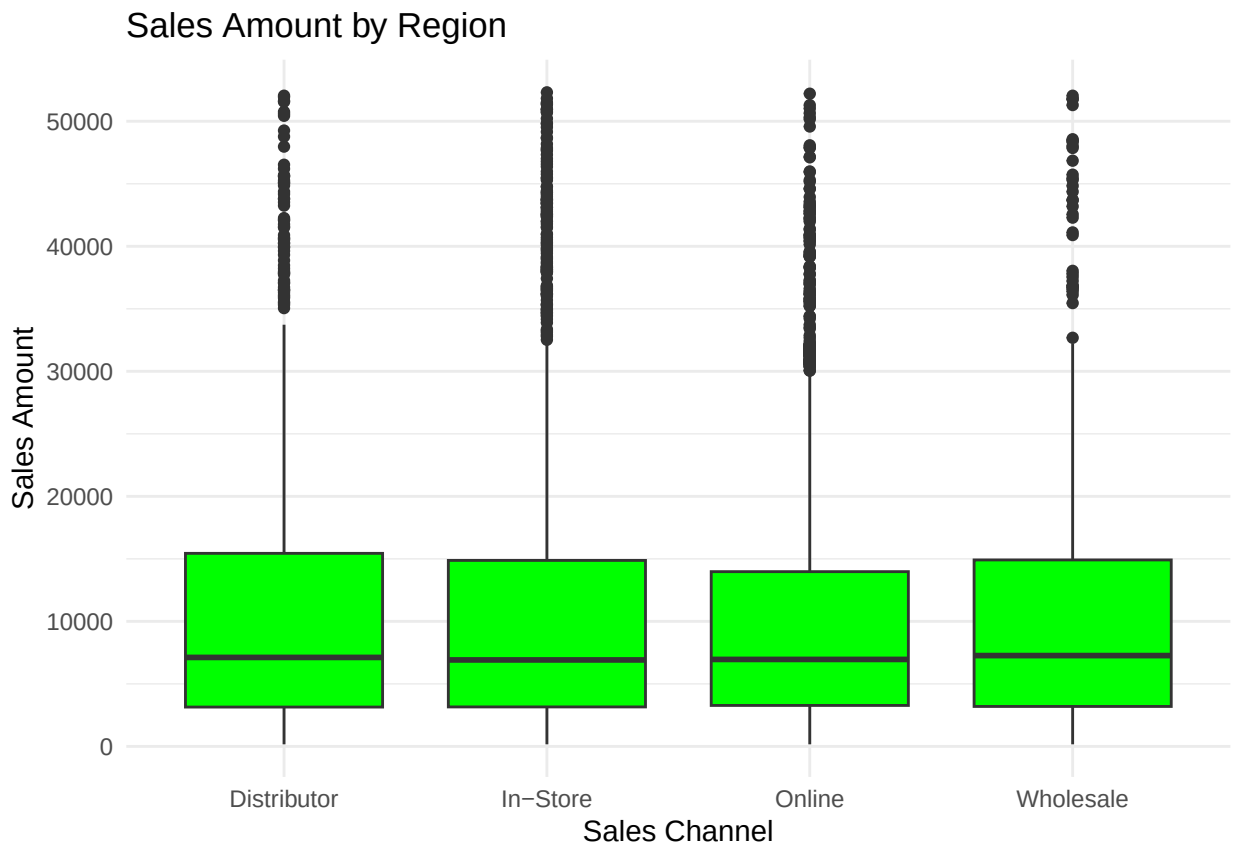


Histogram for an Order Quantity

```
ggplot(us_regional_customer_store_sales_data, aes(x = Order_Quantity)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  theme_minimal() +
  labs(title = "Histogram of Order Quantities", x = "Order Quantity", y = "Frequency")
```

Boxplot for Sales channel and overall price

```
ggplot(us_regional_customer_store_sales_data, aes(x = Sales_Channel, y = overall_price)) +
  geom_boxplot(fill = "green") +
  theme_minimal() +
  labs(title = "Sales Amount by Region", x = "Sales Channel", y = "Sales Amount")
```



Scatter Plot for Unit Price and Unit Cost

```
ggplot(us_regional_customer_store_sales_data, aes(x = Unit_Cost, y = Unit_Price)) +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Unit Cost vs Unit Price", x = "Unit Cost", y = "Unit Price")
```

Linear model to predict the unit price

based on the unit cost

```
model <- lm(`Unit_Price` ~  `Unit_Cost`, data = us_regional_customer_store_sales_data)
```

Print the summary about the model

```
summary(model)
```

```
##
## Call:
## lm(formula = Unit_Price ~ Unit_Cost, data = us_regional_customer_store_sales_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1576.2  -308.9  -134.6   193.7  2533.7
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.555e+02  1.023e+01   24.98   <2e-16 ***
## Unit_Cost   1.417e+00  5.639e-03  251.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 560.8 on 7989 degrees of freedom
## Multiple R-squared:  0.8877, Adjusted R-squared:  0.8877
## F-statistic: 6.314e+04 on 1 and 7989 DF,  p-value: < 2.2e-16
```

Defines a new data frame with the unit cost

```
new_data <- data.frame(
  `Unit_Cost` = c(3)
)
```

Prediciting the unit price based on the unit cost

```
predictions <- predict(model, newdata = new_data)
```

Print the value

```
print(predictions)
```

```
##       1
## 259.709
```