R Notebook

Product Sentiment Analysis

# load libraries that you feel you need and explain why

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.4.2

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.4.3

## Warning: package 'readr' was built under R version 4.4.3

## Warning: package 'forcats' was built under R version 4.4.2

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.1  
## ✔ readr 2.1.5

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tm)

## Warning: package 'tm' was built under R version 4.4.3

## Loading required package: NLP

## Warning: package 'NLP' was built under R version 4.4.2

##   
## Attaching package: 'NLP'  
##   
## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.4.3

## Loading required package: RColorBrewer

library(RColorBrewer)  
library(tidytext)

## Warning: package 'tidytext' was built under R version 4.4.3

library(ggplot2)  
library(igraph)

## Warning: package 'igraph' was built under R version 4.4.3

##   
## Attaching package: 'igraph'  
##   
## The following objects are masked from 'package:purrr':  
##   
## compose, simplify  
##   
## The following object is masked from 'package:tidyr':  
##   
## crossing  
##   
## The following object is masked from 'package:tibble':  
##   
## as\_data\_frame  
##   
## The following objects are masked from 'package:lubridate':  
##   
## %--%, union  
##   
## The following objects are masked from 'package:dplyr':  
##   
## as\_data\_frame, groups, union  
##   
## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum  
##   
## The following object is masked from 'package:base':  
##   
## union

library(ggraph)

## Warning: package 'ggraph' was built under R version 4.4.3

# Load the data from a file

# Assess the data

head(posts)

## Transaction\_ID TimeKey Year Week Number\_of\_Items T\_Ounces T\_Price T\_Cost  
## 1 1 2018-1 2018 1 4 24 $21.00 $8.02   
## 2 2 2018-1 2018 1 4 36 $23.50 $11.71   
## 3 3 2018-1 2018 1 4 36 $22.50 $11.65   
## 4 4 2018-1 2018 1 3 24 $17.00 $7.86   
## 5 5 2018-1 2018 1 1 4 $3.75 $1.35   
## 6 6 2018-1 2018 1 5 24 $22.75 $8.12   
## CustomerID Star\_Rating Comment Chocolate Feature Superman Vanilla  
## 1 NA NA 1 2 0 1  
## 2 NA NA 1 2 0 1  
## 3 NA NA 0 3 1 0  
## 4 NA NA 0 0 0 3  
## 5 NA NA 1 0 0 0  
## 6 NA NA 1 1 1 2  
## FeatureFlavor ParlorLocation ProductionLocation  
## 1 Superman Alpena Mount Pleasant  
## 2 Superman Alpena Mount Pleasant  
## 3 Superman Alpena Mount Pleasant  
## 4 Superman Alpena Mount Pleasant  
## 5 Superman Alpena Mount Pleasant  
## 6 Superman Alpena Mount Pleasant

summary(posts)

## Transaction\_ID TimeKey Year Week   
## Min. : 1 Length:27669 Min. :2018 Min. : 1.000   
## 1st Qu.: 8622 Class :character 1st Qu.:2019 1st Qu.: 4.000   
## Median :17290 Mode :character Median :2020 Median : 7.000   
## Mean :17394 Mean :2020 Mean : 6.933   
## 3rd Qu.:26211 3rd Qu.:2021 3rd Qu.:10.000   
## Max. :34570 Max. :2021 Max. :14.000   
##   
## Number\_of\_Items T\_Ounces T\_Price T\_Cost   
## Min. :1.000 Min. : 4.00 Length:27669 Length:27669   
## 1st Qu.:2.000 1st Qu.:12.00 Class :character Class :character   
## Median :3.000 Median :16.00 Mode :character Mode :character   
## Mean :3.145 Mean :18.86   
## 3rd Qu.:4.000 3rd Qu.:24.00   
## Max. :6.000 Max. :60.00   
##   
## CustomerID Star\_Rating Comment Chocolate   
## Min. : 1.0 Min. :1.000 Length:27669 Min. :0.0000   
## 1st Qu.: 282.0 1st Qu.:4.000 Class :character 1st Qu.:0.0000   
## Median : 614.0 Median :5.000 Mode :character Median :0.0000   
## Mean : 706.9 Mean :4.497 Mean :0.6365   
## 3rd Qu.:1072.0 3rd Qu.:5.000 3rd Qu.:1.0000   
## Max. :1998.0 Max. :5.000 Max. :5.0000   
## NA's :12040 NA's :14100   
## Feature Superman Vanilla FeatureFlavor   
## Min. :0.000 Min. :0.000 Min. :0.0000 Length:27669   
## 1st Qu.:1.000 1st Qu.:0.000 1st Qu.:0.0000 Class :character   
## Median :1.000 Median :0.000 Median :0.0000 Mode :character   
## Mean :1.572 Mean :0.312 Mean :0.6246   
## 3rd Qu.:2.000 3rd Qu.:1.000 3rd Qu.:1.0000   
## Max. :6.000 Max. :4.000 Max. :6.0000   
##   
## ParlorLocation ProductionLocation  
## Length:27669 Length:27669   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##   
##

colSums(is.na(posts))

## Transaction\_ID TimeKey Year Week   
## 0 0 0 0   
## Number\_of\_Items T\_Ounces T\_Price T\_Cost   
## 0 0 0 0   
## CustomerID Star\_Rating Comment Chocolate   
## 12040 14100 0 0   
## Feature Superman Vanilla FeatureFlavor   
## 0 0 0 0   
## ParlorLocation ProductionLocation   
## 0 0

duplicate\_rows <- sum(duplicated(posts))  
print(paste("Number of duplicate rows:", duplicate\_rows))

## [1] "Number of duplicate rows: 0"

# Insights Gained:

1. There are 18 columns and 27669 row in the datset
2. There are 11 integer columns and 7 non integer columns. Among the non integer column T\_Price, T\_Cost columns should be numeric
3. There are 12040 null values in CustomerID, 14100 null values in star\_Rating columns
4. There are blank spaces in Comment column so replacing them with Not Available
5. There are no duplicates

# Changes:

1. Remove $ symbol in T\_Price and T\_Cost then convert to numeric for calculations
2. Replacing CustomerID NA values with “Not Available”. There are 14100 Null Values in Star\_Rating so deleting them affects the analysis results so I decided to replace with their mean
3. There are Empty balcks which means they havent rated so no review so Im replacing empty black with “No Review Given”
4. Creating new column called Profits where Profits = T\_Price - T\_Cost

# 1. Remove $ symbol in T\_Price and T\_Cost then convert to numeric

posts$T\_Price <- as.numeric(gsub("\\$", "", posts$T\_Price))  
posts$T\_Cost <- as.numeric(gsub("\\$", "", posts$T\_Cost))

# 2. Replacing CustomerID NA values with “Not Available” and Null Values in Star\_Rating by mean

posts$CustomerID[is.na(posts$CustomerID)] <- "Not Available"  
Mean\_Star\_Rating <- mean(posts$Star\_Rating, na.rm = TRUE)  
posts$Star\_Rating[is.na(posts$Star\_Rating)] <- Mean\_Star\_Rating

# 3. There are Empty balcks which means they havent rated so no review so Im replacing empty black with “No Review Given”

posts$Comment[posts$Comment == ""] <- "No Review Given"

# 4 Creating new column called Profits

posts$Profit <- posts$T\_Price - posts$T\_Cost

# Check Data

head(posts)

## Transaction\_ID TimeKey Year Week Number\_of\_Items T\_Ounces T\_Price T\_Cost  
## 1 1 2018-1 2018 1 4 24 21.00 8.02  
## 2 2 2018-1 2018 1 4 36 23.50 11.71  
## 3 3 2018-1 2018 1 4 36 22.50 11.65  
## 4 4 2018-1 2018 1 3 24 17.00 7.86  
## 5 5 2018-1 2018 1 1 4 3.75 1.35  
## 6 6 2018-1 2018 1 5 24 22.75 8.12  
## CustomerID Star\_Rating Comment Chocolate Feature Superman Vanilla  
## 1 Not Available 4.497236 No Review Given 1 2 0 1  
## 2 Not Available 4.497236 No Review Given 1 2 0 1  
## 3 Not Available 4.497236 No Review Given 0 3 1 0  
## 4 Not Available 4.497236 No Review Given 0 0 0 3  
## 5 Not Available 4.497236 No Review Given 1 0 0 0  
## 6 Not Available 4.497236 No Review Given 1 1 1 2  
## FeatureFlavor ParlorLocation ProductionLocation Profit  
## 1 Superman Alpena Mount Pleasant 12.98  
## 2 Superman Alpena Mount Pleasant 11.79  
## 3 Superman Alpena Mount Pleasant 10.85  
## 4 Superman Alpena Mount Pleasant 9.14  
## 5 Superman Alpena Mount Pleasant 2.40  
## 6 Superman Alpena Mount Pleasant 14.63

summary(posts)

## Transaction\_ID TimeKey Year Week   
## Min. : 1 Length:27669 Min. :2018 Min. : 1.000   
## 1st Qu.: 8622 Class :character 1st Qu.:2019 1st Qu.: 4.000   
## Median :17290 Mode :character Median :2020 Median : 7.000   
## Mean :17394 Mean :2020 Mean : 6.933   
## 3rd Qu.:26211 3rd Qu.:2021 3rd Qu.:10.000   
## Max. :34570 Max. :2021 Max. :14.000   
## Number\_of\_Items T\_Ounces T\_Price T\_Cost   
## Min. :1.000 Min. : 4.00 Min. : 3.75 Min. : 1.280   
## 1st Qu.:2.000 1st Qu.:12.00 1st Qu.: 9.50 1st Qu.: 3.950   
## Median :3.000 Median :16.00 Median :14.25 Median : 5.450   
## Mean :3.145 Mean :18.86 Mean :15.49 Mean : 6.244   
## 3rd Qu.:4.000 3rd Qu.:24.00 3rd Qu.:20.75 3rd Qu.: 8.080   
## Max. :6.000 Max. :60.00 Max. :40.75 Max. :19.590   
## CustomerID Star\_Rating Comment Chocolate   
## Length:27669 Min. :1.000 Length:27669 Min. :0.0000   
## Class :character 1st Qu.:4.497 Class :character 1st Qu.:0.0000   
## Mode :character Median :4.497 Mode :character Median :0.0000   
## Mean :4.497 Mean :0.6365   
## 3rd Qu.:5.000 3rd Qu.:1.0000   
## Max. :5.000 Max. :5.0000   
## Feature Superman Vanilla FeatureFlavor   
## Min. :0.000 Min. :0.000 Min. :0.0000 Length:27669   
## 1st Qu.:1.000 1st Qu.:0.000 1st Qu.:0.0000 Class :character   
## Median :1.000 Median :0.000 Median :0.0000 Mode :character   
## Mean :1.572 Mean :0.312 Mean :0.6246   
## 3rd Qu.:2.000 3rd Qu.:1.000 3rd Qu.:1.0000   
## Max. :6.000 Max. :4.000 Max. :6.0000   
## ParlorLocation ProductionLocation Profit   
## Length:27669 Length:27669 Min. : 2.400   
## Class :character Class :character 1st Qu.: 5.740   
## Mode :character Mode :character Median : 8.890   
## Mean : 9.246   
## 3rd Qu.:12.320   
## Max. :22.420

colSums(is.na(posts))

## Transaction\_ID TimeKey Year Week   
## 0 0 0 0   
## Number\_of\_Items T\_Ounces T\_Price T\_Cost   
## 0 0 0 0   
## CustomerID Star\_Rating Comment Chocolate   
## 0 0 0 0   
## Feature Superman Vanilla FeatureFlavor   
## 0 0 0 0   
## ParlorLocation ProductionLocation Profit   
## 0 0 0

duplicate\_rows <- sum(duplicated(posts))  
print(paste("Number of duplicate rows:", duplicate\_rows))

## [1] "Number of duplicate rows: 0"

# Insights Gained:

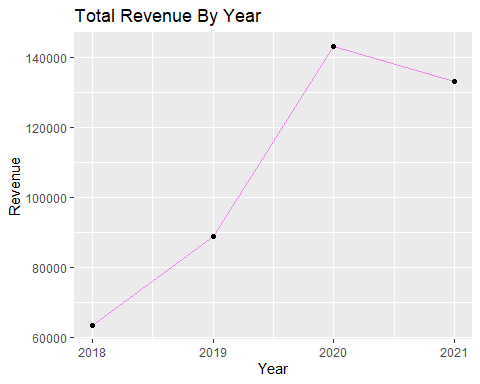
There are no null values and Duplicates and all the mentioned issues in data are cleared so we have succesfully cleaned the data now proceedingwith analysis

View(posts)

# Visualizations

# 1. Total Revenue from Each Year

yr\_rev <- posts %>% group\_by(Year) %>% summarise(total\_revenue = sum(T\_Price))   
  
ggplot(yr\_rev, aes(x = Year, y = total\_revenue)) + geom\_line(color = "violet") +  
 geom\_point() + labs(title = "Total Revenue By Year", x = "Year", y = "Revenue")



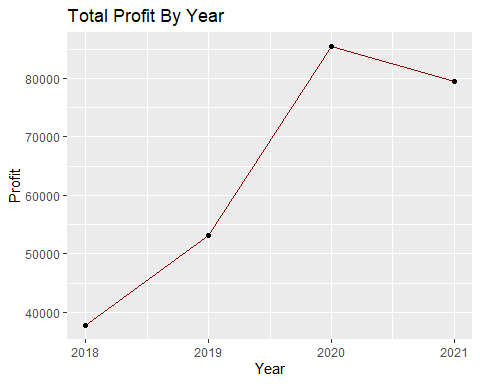
Insights Gained: 2020 is the year with more Revenue then other years. From 2018 to 2020 there is good increase in revenues but after 2020 there is downfall in revenue

# 2. Total Profits by Year

yr\_profit <- posts %>% group\_by(Year) %>% summarise(total\_profit = sum(Profit))  
print(yr\_profit)

## # A tibble: 4 × 2  
## Year total\_profit  
## <int> <dbl>  
## 1 2018 37770.  
## 2 2019 53094.  
## 3 2020 85434.  
## 4 2021 79541.

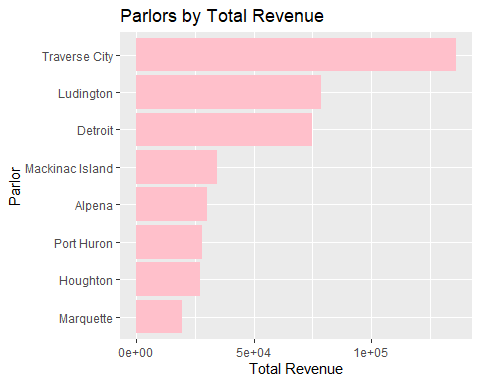
ggplot(yr\_profit, aes(x = Year, y = total\_profit)) + geom\_line(color = "darkred") +   
 geom\_point() + labs(title = "Total Profit By Year", x = "Year", y = "Profit")



Insights Gained: There is positive increase in profit till 2020 then profit has slightly decreased

# 3. Parlors by Total Revenue

t\_parlors <- posts %>% group\_by(ParlorLocation) %>%  
 summarise(total\_revenue = sum(T\_Price, na.rm = TRUE)) %>% arrange(desc(total\_revenue))  
  
ggplot(t\_parlors, aes(x = reorder(ParlorLocation, total\_revenue), y = total\_revenue)) +  
 geom\_bar(stat = "identity", fill = "pink") + coord\_flip() +  
 labs(title = "Parlors by Total Revenue", x = "Parlor", y = "Total Revenue")



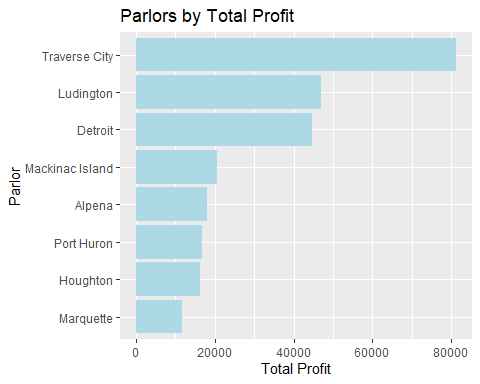
Insights Gained: Traverse City has highest revenue and Port Huron Has lowest revenue

# 4. Parlors by Total Profit

tp\_parlors <- posts %>% group\_by(ParlorLocation) %>%  
 summarise(total\_profit = sum(Profit, na.rm = TRUE)) %>% arrange(desc(total\_profit))  
  
print(tp\_parlors)

## # A tibble: 8 × 2  
## ParlorLocation total\_profit  
## <chr> <dbl>  
## 1 Traverse City 81280.  
## 2 Ludington 46871.  
## 3 Detroit 44669.  
## 4 Mackinac Island 20452.  
## 5 Alpena 17974.  
## 6 Port Huron 16775.  
## 7 Houghton 16150.  
## 8 Marquette 11668.

ggplot(tp\_parlors, aes(x = reorder(ParlorLocation, total\_profit), y = total\_profit)) +  
 geom\_bar(stat = "identity", fill = "lightblue") +   
 coord\_flip() + labs(title = "Parlors by Total Profit", x = "Parlor", y = "Total Profit")



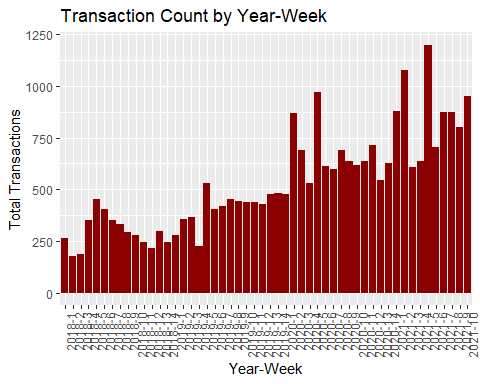
Insights Gained: Traverse city has the highest Profit and Port Huron has the lowest profis

# 5. Highest Number of Transactions in Year-Weaak

trans\_cnt <- posts %>% group\_by(Year, Week) %>% summarise(trans\_cnt = n(), .groups = "drop")   
trans\_cnt\_order <- trans\_cnt %>% arrange(Year, Week) %>% mutate(TimeKey = paste(Year, Week, sep = "-"))  
trans\_cnt\_order$TimeKey <- factor(trans\_cnt\_order$TimeKey, levels = trans\_cnt\_order$TimeKey)  
print(trans\_cnt\_order)

## # A tibble: 52 × 4  
## Year Week trans\_cnt TimeKey  
## <int> <int> <int> <fct>   
## 1 2018 1 263 2018-1   
## 2 2018 2 177 2018-2   
## 3 2018 3 187 2018-3   
## 4 2018 4 351 2018-4   
## 5 2018 5 452 2018-5   
## 6 2018 6 404 2018-6   
## 7 2018 7 349 2018-7   
## 8 2018 8 330 2018-8   
## 9 2018 9 295 2018-9   
## 10 2018 10 279 2018-10  
## # ℹ 42 more rows

ggplot(trans\_cnt\_order, aes(x = TimeKey, y = trans\_cnt)) +  
 geom\_bar(stat = "identity", fill = "darkred") +  
 labs(title = "Transaction Count by Year-Week", x = "Year-Week", y = "Total Transactions") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



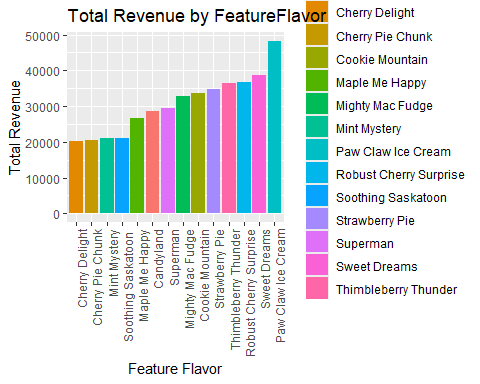
Insights Gained: in 2015, week 5 has more transactions and in 2018, week 2 has least number of transactions

# 6. Total Revenue by FeatureFlavor

rev\_flavor <- posts %>% group\_by(FeatureFlavor) %>%   
 summarise(total\_revenue = sum(T\_Price), .groups = "drop") %>% arrange(desc(total\_revenue))  
print(rev\_flavor)

## # A tibble: 14 × 2  
## FeatureFlavor total\_revenue  
## <chr> <dbl>  
## 1 "Paw Claw Ice Cream" 48267.  
## 2 "Sweet Dreams" 38650.  
## 3 "Robust Cherry Surprise" 36796.  
## 4 "Thimbleberry Thunder" 36410   
## 5 "Strawberry Pie" 34759.  
## 6 "Cookie Mountain" 33646.  
## 7 "Mighty Mac Fudge" 32699   
## 8 "Superman" 29350.  
## 9 "Candyland" 28648.  
## 10 "Maple Me Happy" 26704   
## 11 "Soothing Saskatoon " 21116.  
## 12 "Mint Mystery" 20978.  
## 13 "Cherry Pie Chunk" 20394.  
## 14 "Cherry Delight" 20190.

ggplot(rev\_flavor, aes(x = reorder(FeatureFlavor, total\_revenue), y = total\_revenue, fill = FeatureFlavor)) +  
 geom\_bar(stat = "identity") + labs( title = "Total Revenue by FeatureFlavor", x = "Feature Flavor", y = "Total Revenue") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



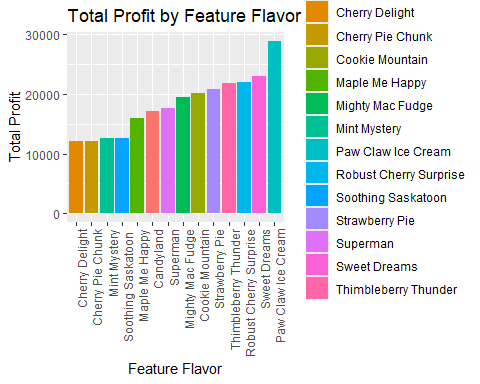
Insights Gained: Paw Claw Ice Cream has the highest revenue and Cherry Delight has the lowest revenue

# 7. Total Profits by FeatureFlavor

profit\_flavor <- posts %>% group\_by(FeatureFlavor) %>% summarise(total\_profit = sum(Profit), .groups = "drop") %>%  
 arrange(desc(total\_profit))  
  
print(profit\_flavor)

## # A tibble: 14 × 2  
## FeatureFlavor total\_profit  
## <chr> <dbl>  
## 1 "Paw Claw Ice Cream" 28845.  
## 2 "Sweet Dreams" 23007.  
## 3 "Robust Cherry Surprise" 21964.  
## 4 "Thimbleberry Thunder" 21708.  
## 5 "Strawberry Pie" 20772.  
## 6 "Cookie Mountain" 20058.  
## 7 "Mighty Mac Fudge" 19506.  
## 8 "Superman" 17534.  
## 9 "Candyland" 17116.  
## 10 "Maple Me Happy" 15933.  
## 11 "Soothing Saskatoon " 12617.  
## 12 "Mint Mystery" 12558.  
## 13 "Cherry Pie Chunk" 12134.  
## 14 "Cherry Delight" 12087.

ggplot(profit\_flavor, aes(x = reorder(FeatureFlavor, total\_profit), y = total\_profit, fill = FeatureFlavor)) +  
 geom\_bar(stat = "identity") + labs(title = "Total Profit by Feature Flavor", x = "Feature Flavor", y = "Total Profit") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



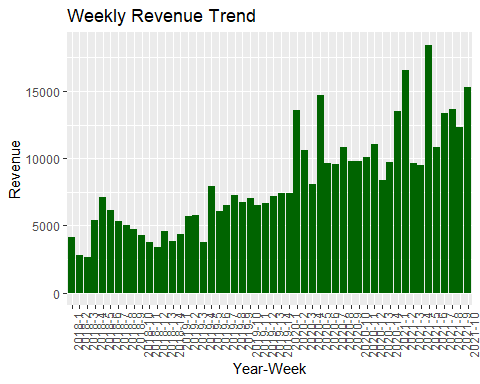
Insights Gained: Paw Claw Ice Cream has the highest Profits where Cherry Delight has the lowest profits

# 8. Total Year-Weekly Revenue

week\_rev <- posts %>% group\_by(Year, Week) %>% summarise(week\_rev = sum(T\_Price), .groups = "drop")  
week\_rev\_ord <- week\_rev %>% arrange(Year, Week) %>% mutate(TimeKey = paste(Year, Week, sep = "-"))  
week\_rev\_ord$TimeKey <- factor(week\_rev\_ord$TimeKey, levels = week\_rev\_ord$TimeKey)  
print(week\_rev\_ord)

## # A tibble: 52 × 4  
## Year Week week\_rev TimeKey  
## <int> <int> <dbl> <fct>   
## 1 2018 1 4128. 2018-1   
## 2 2018 2 2819 2018-2   
## 3 2018 3 2675 2018-3   
## 4 2018 4 5391. 2018-4   
## 5 2018 5 7129 2018-5   
## 6 2018 6 6110. 2018-6   
## 7 2018 7 5319. 2018-7   
## 8 2018 8 5050 2018-8   
## 9 2018 9 4754. 2018-9   
## 10 2018 10 4319. 2018-10  
## # ℹ 42 more rows

ggplot(week\_rev\_ord, aes(x = TimeKey, y = week\_rev)) + geom\_bar(stat = "identity", fill = "darkgreen") +  
 labs(title = "Weekly Revenue Trend", x = "Year-Week", y = "Revenue") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))

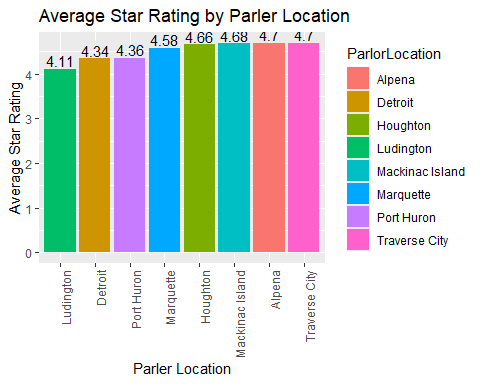
 Insights Gained: In 2021, week 5 has the highest revenue and in 2018, week 3 has the lowest revenue

# 9. Average Star\_Rating by ParlorLocation

avg\_rate\_loc <- posts %>% group\_by(ParlorLocation) %>%   
 summarise(avg\_rate = mean(Star\_Rating, na.rm = TRUE), .groups = "drop") %>%  
 arrange(desc(avg\_rate))  
print(avg\_rate\_loc)

## # A tibble: 8 × 2  
## ParlorLocation avg\_rate  
## <chr> <dbl>  
## 1 Traverse City 4.70  
## 2 Alpena 4.70  
## 3 Mackinac Island 4.68  
## 4 Houghton 4.66  
## 5 Marquette 4.58  
## 6 Port Huron 4.36  
## 7 Detroit 4.34  
## 8 Ludington 4.11

ggplot(avg\_rate\_loc, aes(x = reorder(ParlorLocation, avg\_rate), y = avg\_rate, fill = ParlorLocation)) +  
 geom\_bar(stat = "identity") + geom\_text(aes(label = round(avg\_rate, 2)), vjust = -0.2) +   
 labs( title = "Average Star Rating by Parler Location", x = "Parler Location", y = "Average Star Rating") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



Insights Gained: Almost all Parlor locations has 4+ average reviews Traverse City has the highest average Star Rating which is 4.698 out of 5

# 10. Top 10 Customers by Transaction Count and Their Preffered Flavor

top10\_cust <- posts %>% group\_by(CustomerID) %>% summarise(Transaction\_Count = n(), .groups = "drop") %>%  
 arrange(desc(Transaction\_Count)) %>% head(10)  
  
print(top10\_cust)

## # A tibble: 10 × 2  
## CustomerID Transaction\_Count  
## <chr> <int>  
## 1 Not Available 12040  
## 2 101 38  
## 3 180 38  
## 4 251 38  
## 5 99 38  
## 6 301 37  
## 7 474 37  
## 8 662 37  
## 9 740 37  
## 10 94 37

posts %>% filter(CustomerID %in% top10\_cust$CustomerID) %>% group\_by(CustomerID) %>%  
 summarise(  
 Total\_Chocolate = sum(Chocolate, na.rm = TRUE),  
 Total\_Feature = sum(Feature, na.rm = TRUE),  
 total\_Superman = sum(Superman, na.rm = TRUE),  
 total\_Vanilla = sum(Vanilla, na.rm = TRUE),.groups = "drop") -> top10\_customer\_sum  
  
print(top10\_customer\_sum)

## # A tibble: 10 × 5  
## CustomerID Total\_Chocolate Total\_Feature total\_Superman total\_Vanilla  
## <chr> <int> <int> <int> <int>  
## 1 101 36 93 18 43  
## 2 180 16 52 17 29  
## 3 251 12 41 12 11  
## 4 301 39 79 26 41  
## 5 474 11 36 13 14  
## 6 662 21 79 11 37  
## 7 740 34 83 20 48  
## 8 94 10 45 7 12  
## 9 99 14 35 9 18  
## 10 Not Available 8450 20860 4147 8308

Insights Gained: Customer with customerID 101, 180, 251,99 has the highest transactions which is 38 and all of them most preferred Flavor was Feature and second most preferred flavor was Vanilla

# Now Prforming analysis on Comments

Before that Im removing rows in Comments where value is No Review Given

posts <- posts %>% filter(Comment != "No Review Given")  
head(posts)

## Transaction\_ID TimeKey Year Week Number\_of\_Items T\_Ounces T\_Price T\_Cost  
## 1 5057 2019-1 2019 1 2 8 8.50 2.69  
## 2 5058 2019-1 2019 1 2 16 10.25 5.20  
## 3 5059 2019-1 2019 1 3 16 15.25 5.42  
## 4 5061 2019-1 2019 1 3 16 15.25 5.42  
## 5 5066 2019-1 2019 1 3 20 15.00 6.61  
## 6 5069 2019-1 2019 1 6 32 27.25 10.65  
## CustomerID Star\_Rating  
## 1 903 4  
## 2 959 5  
## 3 5 5  
## 4 1123 4  
## 5 1375 4  
## 6 642 5  
## Comment  
## 1 If you like cherries this one's for you  
## 2 Ice cream is my love language  
## 3 When life gives you lemons, make a lemon sorbet to go with your ice cream  
## 4 Superman you are flying high  
## 5 So good it's bad. What an ice cream shop  
## 6 Capturing the essence of indulgence in every scoop  
## Chocolate Feature Superman Vanilla FeatureFlavor ParlorLocation  
## 1 1 0 0 1 Superman Alpena  
## 2 0 2 0 0 Superman Alpena  
## 3 1 2 0 0 Superman Alpena  
## 4 0 2 1 0 Superman Alpena  
## 5 0 3 0 0 Superman Alpena  
## 6 1 4 1 0 Superman Alpena  
## ProductionLocation Profit  
## 1 Mount Pleasant 5.81  
## 2 Mount Pleasant 5.05  
## 3 Mount Pleasant 9.83  
## 4 Mount Pleasant 9.83  
## 5 Mount Pleasant 8.39  
## 6 Mount Pleasant 16.60

# Extracting words from comments

# Unnest words from Comment  
posts\_words <- posts %>% unnest\_tokens(word, Comment)  
  
# Removing stopwords  
posts\_words <- posts\_words %>% anti\_join(stop\_words)

## Joining with `by = join\_by(word)`

posts\_words %>% inner\_join(get\_sentiments("bing")) %>% head(5)

## Joining with `by = join\_by(word)`

## Transaction\_ID TimeKey Year Week Number\_of\_Items T\_Ounces T\_Price T\_Cost  
## 1 5058 2019-1 2019 1 2 16 10.25 5.20  
## 2 5059 2019-1 2019 1 3 16 15.25 5.42  
## 3 5066 2019-1 2019 1 3 20 15.00 6.61  
## 4 5069 2019-1 2019 1 6 32 27.25 10.65  
## 5 5076 2019-1 2019 1 3 12 11.25 4.05  
## CustomerID Star\_Rating Chocolate Feature Superman Vanilla FeatureFlavor  
## 1 959 5 0 2 0 0 Superman  
## 2 5 5 1 2 0 0 Superman  
## 3 1375 4 0 3 0 0 Superman  
## 4 642 5 1 4 1 0 Superman  
## 5 180 5 0 1 0 2 Superman  
## ParlorLocation ProductionLocation Profit word sentiment  
## 1 Alpena Mount Pleasant 5.05 love positive  
## 2 Alpena Mount Pleasant 9.83 lemon negative  
## 3 Alpena Mount Pleasant 8.39 bad negative  
## 4 Alpena Mount Pleasant 16.60 indulgence positive  
## 5 Detroit Kalamazoo 7.20 indulgence positive

posts\_words %>% inner\_join(get\_sentiments("afinn")) %>% head(5)

## Joining with `by = join\_by(word)`

## Transaction\_ID TimeKey Year Week Number\_of\_Items T\_Ounces T\_Price T\_Cost  
## 1 5058 2019-1 2019 1 2 16 10.25 5.20  
## 2 5066 2019-1 2019 1 3 20 15.00 6.61  
## 3 5069 2019-1 2019 1 6 32 27.25 10.65  
## 4 5076 2019-1 2019 1 3 12 11.25 4.05  
## 5 5077 2019-1 2019 1 3 16 14.25 5.36  
## CustomerID Star\_Rating Chocolate Feature Superman Vanilla FeatureFlavor  
## 1 959 5 0 2 0 0 Superman  
## 2 1375 4 0 3 0 0 Superman  
## 3 642 5 1 4 1 0 Superman  
## 4 180 5 0 1 0 2 Superman  
## 5 1283 5 0 2 0 1 Superman  
## ParlorLocation ProductionLocation Profit word value  
## 1 Alpena Mount Pleasant 5.05 love 3  
## 2 Alpena Mount Pleasant 8.39 bad -3  
## 3 Alpena Mount Pleasant 16.60 scoop 3  
## 4 Detroit Kalamazoo 7.20 scoop 3  
## 5 Detroit Kalamazoo 8.89 scoop 3

posts\_words %>% inner\_join(get\_sentiments("nrc")) %>% head(10)

## Joining with `by = join\_by(word)`

## Warning in inner\_join(., get\_sentiments("nrc")): Detected an unexpected many-to-many relationship between `x` and `y`.  
## ℹ Row 4 of `x` matches multiple rows in `y`.  
## ℹ Row 2738 of `y` matches multiple rows in `x`.  
## ℹ If a many-to-many relationship is expected, set `relationship =  
## "many-to-many"` to silence this warning.

## Transaction\_ID TimeKey Year Week Number\_of\_Items T\_Ounces T\_Price T\_Cost  
## 1 5058 2019-1 2019 1 2 16 10.25 5.20  
## 2 5058 2019-1 2019 1 2 16 10.25 5.20  
## 3 5058 2019-1 2019 1 2 16 10.25 5.20  
## 4 5058 2019-1 2019 1 2 16 10.25 5.20  
## 5 5058 2019-1 2019 1 2 16 10.25 5.20  
## 6 5058 2019-1 2019 1 2 16 10.25 5.20  
## 7 5059 2019-1 2019 1 3 16 15.25 5.42  
## 8 5059 2019-1 2019 1 3 16 15.25 5.42  
## 9 5059 2019-1 2019 1 3 16 15.25 5.42  
## 10 5059 2019-1 2019 1 3 16 15.25 5.42  
## CustomerID Star\_Rating Chocolate Feature Superman Vanilla FeatureFlavor  
## 1 959 5 0 2 0 0 Superman  
## 2 959 5 0 2 0 0 Superman  
## 3 959 5 0 2 0 0 Superman  
## 4 959 5 0 2 0 0 Superman  
## 5 959 5 0 2 0 0 Superman  
## 6 959 5 0 2 0 0 Superman  
## 7 5 5 1 2 0 0 Superman  
## 8 5 5 1 2 0 0 Superman  
## 9 5 5 1 2 0 0 Superman  
## 10 5 5 1 2 0 0 Superman  
## ParlorLocation ProductionLocation Profit word sentiment  
## 1 Alpena Mount Pleasant 5.05 cream anticipation  
## 2 Alpena Mount Pleasant 5.05 cream joy  
## 3 Alpena Mount Pleasant 5.05 cream positive  
## 4 Alpena Mount Pleasant 5.05 cream surprise  
## 5 Alpena Mount Pleasant 5.05 love joy  
## 6 Alpena Mount Pleasant 5.05 love positive  
## 7 Alpena Mount Pleasant 9.83 lemon disgust  
## 8 Alpena Mount Pleasant 9.83 lemon negative  
## 9 Alpena Mount Pleasant 9.83 cream anticipation  
## 10 Alpena Mount Pleasant 9.83 cream joy

# Now answer/create the following:

# Question 1. Top 15 meaningful words

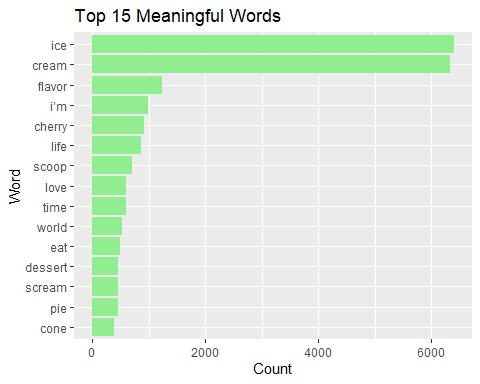
top15\_words <- posts\_words %>% count(word, sort = TRUE) %>% top\_n(15)

## Selecting by n

print(top15\_words)

## word n  
## 1 ice 6408  
## 2 cream 6343  
## 3 flavor 1238  
## 4 i’m 994  
## 5 cherry 919  
## 6 life 867  
## 7 scoop 712  
## 8 love 607  
## 9 time 606  
## 10 world 531  
## 11 eat 497  
## 12 dessert 463  
## 13 scream 462  
## 14 pie 461  
## 15 cone 381

top15\_words %>% mutate(word = reorder(word, n)) %>%  
 ggplot(aes(x = word, y = n)) +  
 geom\_col(fill = "lightgreen") +  
 coord\_flip() +  
 labs(title = "Top 15 Meaningful Words", x = "Word", y = "Count")

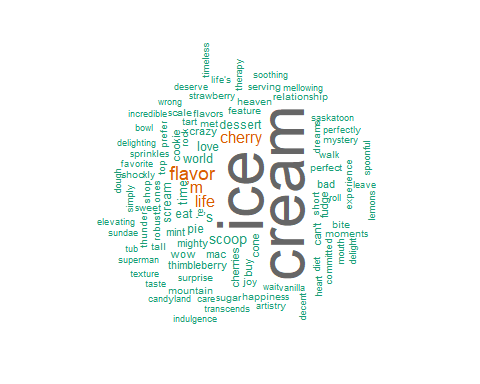
 Answer: ice is the most frequently used words followed by cream, flavor

# 2. Question Word cloud

set.seed(1234)  
wordcloud(words = posts\_words$word,   
 min.freq = 2,  
 max.words = 100,   
 random.order = FALSE,   
 rot.per = 0.35,  
 colors = brewer.pal(8, "Dark2"))

## Warning in tm\_map.SimpleCorpus(corpus, tm::removePunctuation): transformation  
## drops documents

## Warning in tm\_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,  
## tm::stopwords())): transformation drops documents



Answer: we can observe the word colud where it shows ice, cream are most used words followed by flavor, ’m, cherry, life and other words

# Question 3. Who are the top 5 customers by CustomerID who posts the most comments on products? For each these 5 are these positive or negative customers overall considering a measure using the afinn sentiment measure?

top5\_cust <- posts %>% group\_by(CustomerID) %>%  
 summarise(Comment\_Count = n()) %>% arrange(desc(Comment\_Count)) %>% head(5)  
print(top5\_cust)

## # A tibble: 5 × 2  
## CustomerID Comment\_Count  
## <chr> <int>  
## 1 180 38  
## 2 251 38  
## 3 99 38  
## 4 301 37  
## 5 474 37

Sentiment\_type <- posts %>% unnest\_tokens(word, Comment) %>% inner\_join(get\_sentiments("afinn")) %>% group\_by(CustomerID) %>%  
 summarise(Afinn\_Sentiment\_Score = sum(value))

## Joining with `by = join\_by(word)`

top5\_sentiment <- top5\_cust %>% left\_join(Sentiment\_type, by = "CustomerID")  
print(top5\_sentiment)

## # A tibble: 5 × 3  
## CustomerID Comment\_Count Afinn\_Sentiment\_Score  
## <chr> <int> <dbl>  
## 1 180 38 49  
## 2 251 38 46  
## 3 99 38 55  
## 4 301 37 39  
## 5 474 37 55

Answer: Customer with customerID 180,251,99,301, 474 the top 5 customers who posted more number comments, and they have positive Afinn Sentiment score

# Question 4. What Parlor site has the most comments? Are these negative or positive?

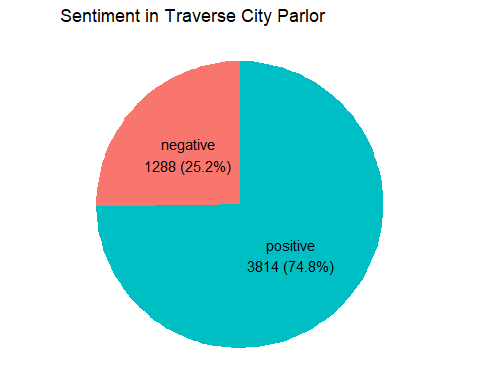
parlor\_cmt <- posts %>% group\_by(ParlorLocation) %>% summarise(comment\_count = n()) %>% arrange(desc(comment\_count))  
print(parlor\_cmt)

## # A tibble: 8 × 2  
## ParlorLocation comment\_count  
## <chr> <int>  
## 1 Traverse City 5467  
## 2 Ludington 2937  
## 3 Mackinac Island 1263  
## 4 Alpena 1222  
## 5 Detroit 1059  
## 6 Houghton 921  
## 7 Port Huron 362  
## 8 Marquette 338

top\_parlor <- parlor\_cmt$ParlorLocation[1]  
sen\_cnt <- posts %>% filter(ParlorLocation == top\_parlor) %>% unnest\_tokens(word, Comment) %>%  
 inner\_join(get\_sentiments("bing")) %>% count(sentiment)

## Joining with `by = join\_by(word)`

sen\_cnt <- sen\_cnt %>% mutate(percentage = round(n / sum(n) \* 100, 1), label = paste0(sentiment, "\n", n, " (", percentage, "%)"))  
ggplot(sen\_cnt, aes(x = "", y = n, fill = sentiment)) +  
 geom\_bar(width = 1, stat = "identity") +  
 coord\_polar("y", start = 0) +  
 geom\_text(aes(label = label), position = position\_stack(vjust = 0.5), size = 4) +  
 labs(title = paste("Sentiment in", top\_parlor, "Parlor")) +  
 theme\_void() +  
 theme(legend.position = "none")



Answer: Traverse City has the highest comments and among those comments 1288 are negative and 3814 are positive comments

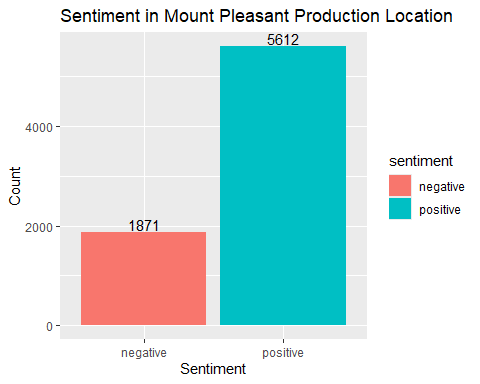
# Question 5. What production site has the most comments? Are these negative or positive?

prod\_cmt <- posts %>% group\_by(ProductionLocation) %>%   
 summarise(comment\_count = n()) %>% arrange(desc(comment\_count))  
print(prod\_cmt)

## # A tibble: 3 × 2  
## ProductionLocation comment\_count  
## <chr> <int>  
## 1 Mount Pleasant 7952  
## 2 Kalamazoo 4358  
## 3 Sault Ste. Marie 1259

top\_prod <- prod\_cmt$ProductionLocation[1]  
  
posts %>% filter(ProductionLocation == top\_prod) %>% unnest\_tokens(word, Comment) %>%  
 inner\_join(get\_sentiments("bing")) %>% count(sentiment) %>%  
 ggplot(aes(x = sentiment, y = n, fill = sentiment)) + geom\_col() +   
 geom\_text(aes(label = n), vjust = -0.2) +   
 labs(title = paste("Sentiment in", top\_prod, "Production Location"), x = "Sentiment", y = "Count")

## Joining with `by = join\_by(word)`

 Answer: Mount Pleasant has the highest Comments and it has 1871 negative and 5612 positive comments

# Question 6. What are the top 10 most frequent comments made by customers and how many times did each comment get made?

top10\_cmts <- posts %>% group\_by(Comment) %>% summarise(Frequency = n()) %>%  
 arrange(desc(Frequency)) %>% head(10)  
  
print(top10\_cmts)

## # A tibble: 10 × 2  
## Comment Frequency  
## <chr> <int>  
## 1 Hello ice cream heaven - where have you been all my life 263  
## 2 If you like cherries this one's for you 246  
## 3 This is crazy - whoever thought of this flavor 244  
## 4 Good 233  
## 5 Better than most 232  
## 6 What a shockly tart Robust Cherry Surprise flavor - you've got to … 232  
## 7 Wow. 230  
## 8 Perfect Strawberry Pie 227  
## 9 Really good ice cream 226  
## 10 So good it's bad. What an ice cream shop 226

Answer: We can observe top 10 most frequent comments made by customers and we can see their count. Most Frequent comment - Hello ice cream heaven - where have you been all my life which occurred 263 times And all those comments are positive comments

# Question 7. Create a network diagram based on bigrams

posts\_bigrams <- posts %>% unnest\_tokens(bigram, Comment, token = "ngrams", n = 2)  
bigrams\_sep <- posts\_bigrams %>% separate(bigram, c("word1", "word2"), sep = " ")  
bigrams\_fil <- bigrams\_sep %>% filter(!word1 %in% stop\_words$word, !word2 %in% stop\_words$word)  
bigram\_counts <- bigrams\_fil %>% count(word1, word2, sort = TRUE)  
bigram\_graph <- bigram\_counts %>% filter(n > 1) %>% graph\_from\_data\_frame()

## Warning: In `d`, `NA` elements were replaced with string "NA".

set.seed(123)  
ggraph(bigram\_graph, layout = "fr") + geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE) +  
 geom\_node\_point(color = "lightblue", size = 5) + geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void() + labs(title = "Bigram Network Diagram")

