Iowa Liquor Sales Case Study

Data Exploration Notebook

Introduction

This Rmd file contains code to analyze the Iowa Liquor Sales data set. The analyses performed here extend past the analysis shown in the lecture videos. In particular, the bottom of the file explores association rules between different types of liquor.

Data source

Setup

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
liquor <- read_csv("Iowa-Liquor-Sales.csv")
```

```
## Rows: 1593369 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): Invoice/Item Number, Date, Store Name, Address, City, Store Locati...
## dbl (13): Store Number, Zip Code, County Number, Category, Item Number, Pack...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
liquor
```

```
## # A tibble: 1,593,369 x 24
##    Invoice/~1 Date  Store~2 Store~3 Address City  Zip C~4 Store~5 Count~6 County
##    <chr>      <chr>   <dbl> <chr>   <chr>   <chr>   <dbl> <chr>     <dbl> <chr>
##  1 INV-47935~ 06/0~    4014 Wal-Ma~ 510 C ~ Deni~   51442 <NA>         24 CRAWF~
##  2 INV-47935~ 06/0~    4014 Wal-Ma~ 510 C ~ Deni~   51442 <NA>         24 CRAWF~
##  3 INV-47936~ 06/0~    5343 Frohli~ 403 Ma~ Coon~   50058 POINT ~      14 CARRO~
##  4 INV-41517~ 11/0~    5417 Casey'~ 9001 6~ Ceda~   52404 <NA>         57 LINN
##  5 INV-41518~ 11/0~    4921 Market~ 5340 1~ Ceda~   52404 <NA>         57 LINN
##  6 INV-41518~ 11/0~    5687 Casey'~ 4560 1~ Ceda~   52404 <NA>         57 LINN
##  7 INV-47938~ 06/0~    4568 Select~ 4103 F~ Siou~   51108 POINT ~      97 WOODB~
##  8 INV-41521~ 11/0~    2648 Hy-Vee~ 555 S ~ West~   50265 POINT ~      77 POLK
##  9 INV-47940~ 06/0~    3831 The Ma~ 301 An~ Madr~   50156 POINT ~       8 BOONE
## 10 INV-41523~ 11/0~    4379 Kum & ~ 5969 A~ West~   50266 <NA>         77 POLK
## # ... with 1,593,359 more rows, 14 more variables: Category <dbl>,
## #   `Category Name` <chr>, `Vendor Number` <chr>, `Vendor Name` <chr>,
## #   `Item Number` <dbl>, `Item Description` <chr>, Pack <dbl>,
## #   `Bottle Volume (ml)` <dbl>, `State Bottle Cost` <dbl>,
## #   `State Bottle Retail` <dbl>, `Bottles Sold` <dbl>, `Sale (Dollars)` <dbl>,
## #   `Volume Sold (Liters)` <dbl>, `Volume Sold (Gallons)` <dbl>, and
## #   abbreviated variable names 1: `Invoice/Item Number`, 2: `Store Number`, ...
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Data Questions

Summary

- What are the most popular items/vendors/categories?
  - The total number of sales that include a particular item/vendor/category
  - The total volume sold
- What are the most profitable items/vendors/categories?
  - Per-item profit, profit margin
  - Per-sale profit, profit margin

Temporal

- What is the relationship between sales and time?

Spatial

- What is the relationship between sales and county/city?

Market Basket Analysis/Association Rules

- Are there any associations between particular items?

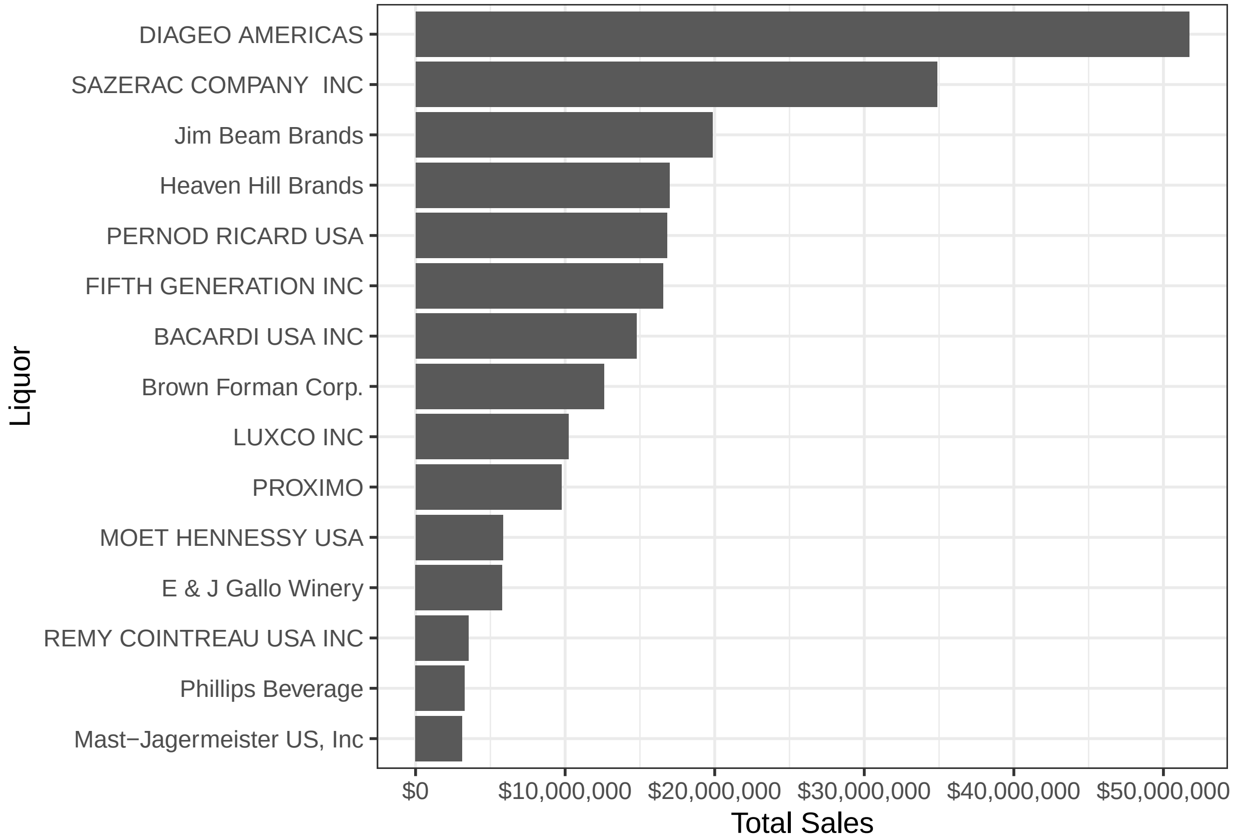
Data Cleaning

```
liquor <- liquor %>%
  select(-c(`Store Number`,
            `County Number`,
            Category,
            `Vendor Number`,
            `Item Number`,
            `Volume Sold (Liters)`)) %>%
  rename(invoice = `Invoice/Item Number`,
         storeName = `Store Name`,
         zip = `Zip Code`,
         location = `Store Location`,
         category = `Category Name`,
         vendor = `Vendor Name`,
         description = `Item Description`,
         bottleVolume = `Bottle Volume (ml)`,
         cost = `State Bottle Cost`,
         retail = `State Bottle Retail`,
         numSold = `Bottles Sold`,
         saleTotal = `Sale (Dollars)`,
         saleVolume = `Volume Sold (Gallons)`) %>%
  mutate(Date = lubridate::dmy(Date),
         location = location %>%
           str_remove(pattern = "POINT \\(") %>%
           str_remove(pattern = "\\)")) %>%
  tidyr::separate(col = location,into = c("long","lat"),sep = " ",convert = TRUE)

liquor
```

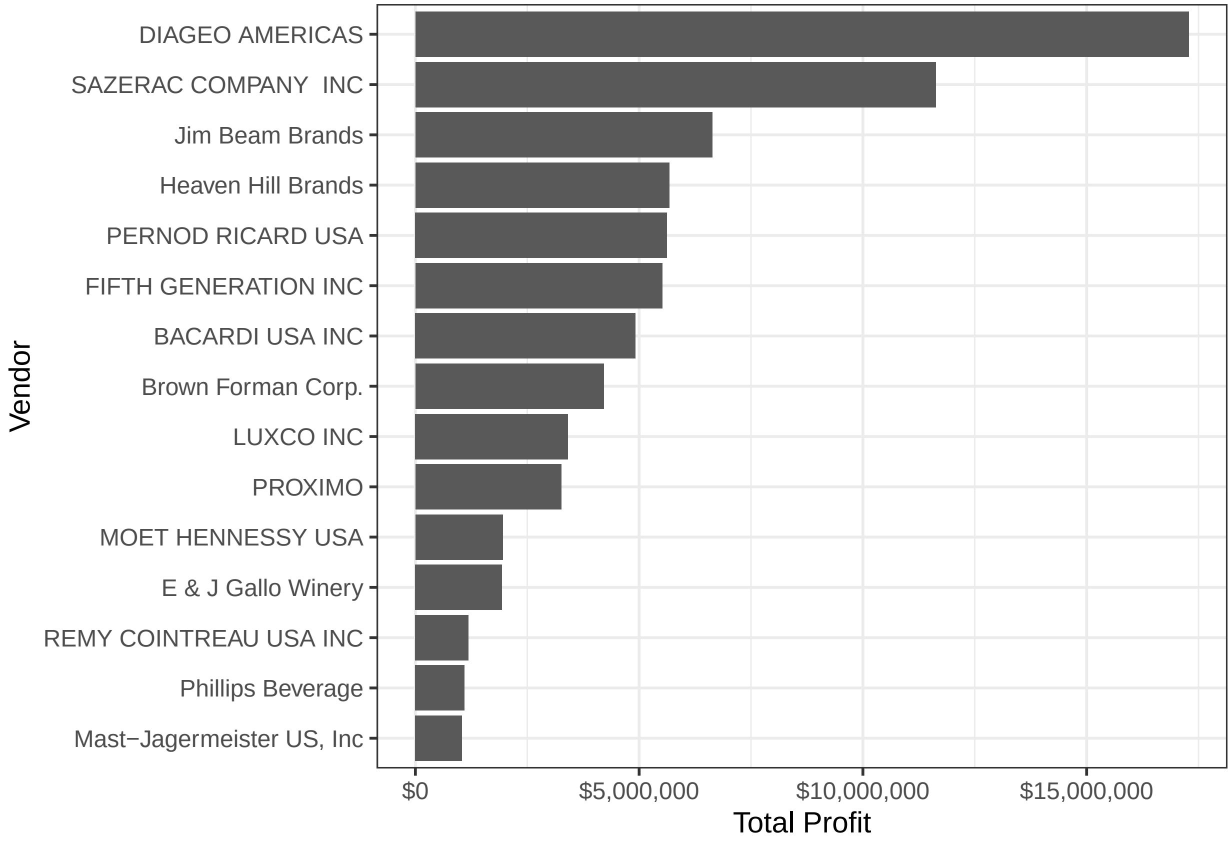
```
## # A tibble: 1,593,369 x 19
##    invoice     Date       store~1 Address City      zip  long   lat County categ~2
##    <chr>       <date>     <chr>   <chr>   <chr>   <dbl> <dbl> <dbl> <chr>  <chr>
##  1 INV-479350~ 2022-01-06 Wal-Ma~ 510 C ~ Deni~ 51442  NA    NA    CRAWF~ White ~
##  2 INV-479350~ 2022-01-06 Wal-Ma~ 510 C ~ Deni~ 51442  NA    NA    CRAWF~ Canadi~
##  3 INV-479361~ 2022-01-06 Frohli~ 403 Ma~ Coon~ 50058 -94.7  41.9 CARRO~ Import~
##  4 INV-415171~ 2021-01-11 Casey'~ 9001 6~ Ceda~ 52404  NA    NA    LINN   Americ~
##  5 INV-415188~ 2021-01-11 Market~ 5340 1~ Ceda~ 52404  NA    NA    LINN   Straig~
##  6 INV-415189~ 2021-01-11 Casey'~ 4560 1~ Ceda~ 52404  NA    NA    LINN   Canadi~
##  7 INV-479389~ 2022-01-06 Select~ 4103 F~ Siou~ 51108 -96.4  42.5 WOODB~ Americ~
##  8 INV-415212~ 2021-01-11 Hy-Vee~ 555 S ~ West~ 50265 -93.8  41.6 POLK   Triple~
##  9 INV-479408~ 2022-01-06 The Ma~ 301 An~ Madr~ 50156 -93.8  41.9 BOONE  Canadi~
## 10 INV-415238~ 2021-01-11 Kum & ~ 5969 A~ West~ 50266  NA    NA    POLK   Americ~
## # ... with 1,593,359 more rows, 9 more variables: vendor <chr>,
## #   description <chr>, Pack <dbl>, bottleVolume <dbl>, cost <dbl>,
## #   retail <dbl>, numSold <dbl>, saleTotal <dbl>, saleVolume <dbl>, and
## #   abbreviated variable names 1: storeName, 2: category
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Visualizations

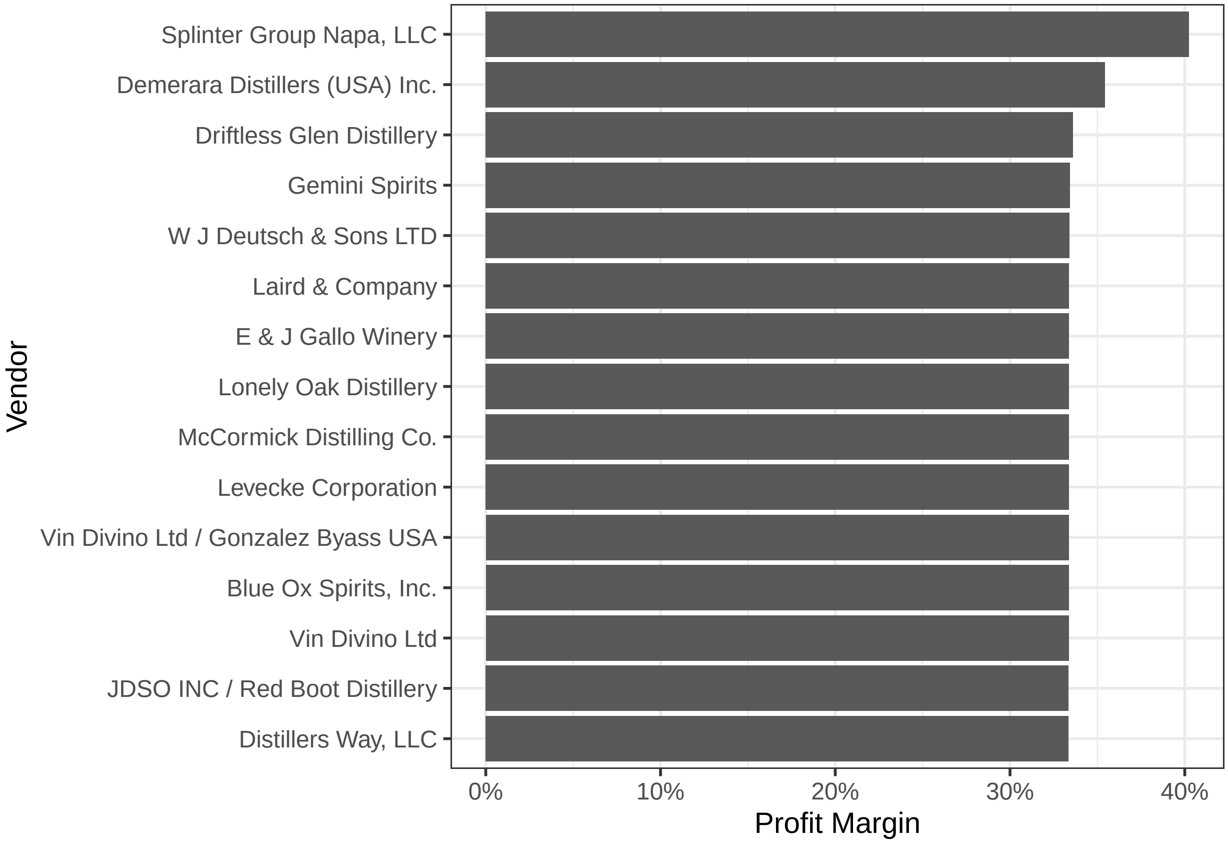
```
liquor %>%
  group_by(vendor) %>%
  # summarise(n = n()) %>%
  summarise(saleTotal = sum(saleTotal)) %>%
  top_n(n = 15,wt = saleTotal) %>%
  ggplot(aes(x=reorder(vendor,saleTotal),y=saleTotal)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_bw() +
  labs(y = "Total Sales",
       x = "Liquor") +
  scale_y_continuous(labels = scales::dollar)
```



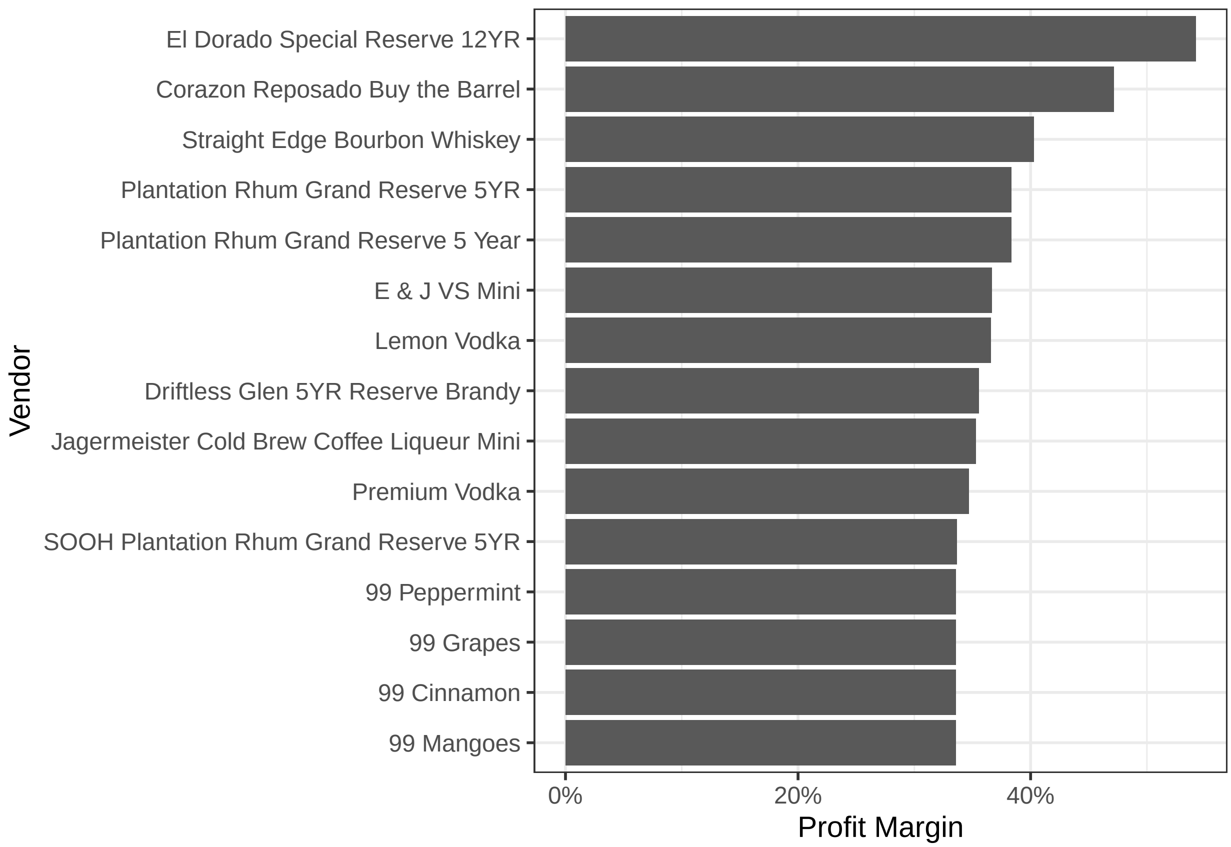
```
liquor %>%
  mutate(profit = retail - cost,
         totalProfit = profit*numSold) %>%
  group_by(vendor) %>%
  summarise(totalProfit = sum(totalProfit)) %>%
  top_n(n = 15,wt = totalProfit) %>%
  ggplot(aes(x=reorder(vendor,totalProfit),y=totalProfit)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_bw() +
  labs(y = "Total Profit",
       x = "Vendor") +
  scale_y_continuous(labels = scales::dollar)
```



```
liquor %>%
  mutate(profit = retail - cost,
         # totalProfit = profit*numSold,
         profitMargin = profit/retail) %>%
  group_by(vendor) %>%
  summarise(profitMargin = mean(profitMargin)) %>%
  top_n(n = 15,wt = profitMargin) %>%
  ggplot(aes(x=reorder(vendor,profitMargin),y=profitMargin)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_bw() +
  labs(y = "Profit Margin",
       x = "Vendor") +
  scale_y_continuous(labels = scales::percent)
```

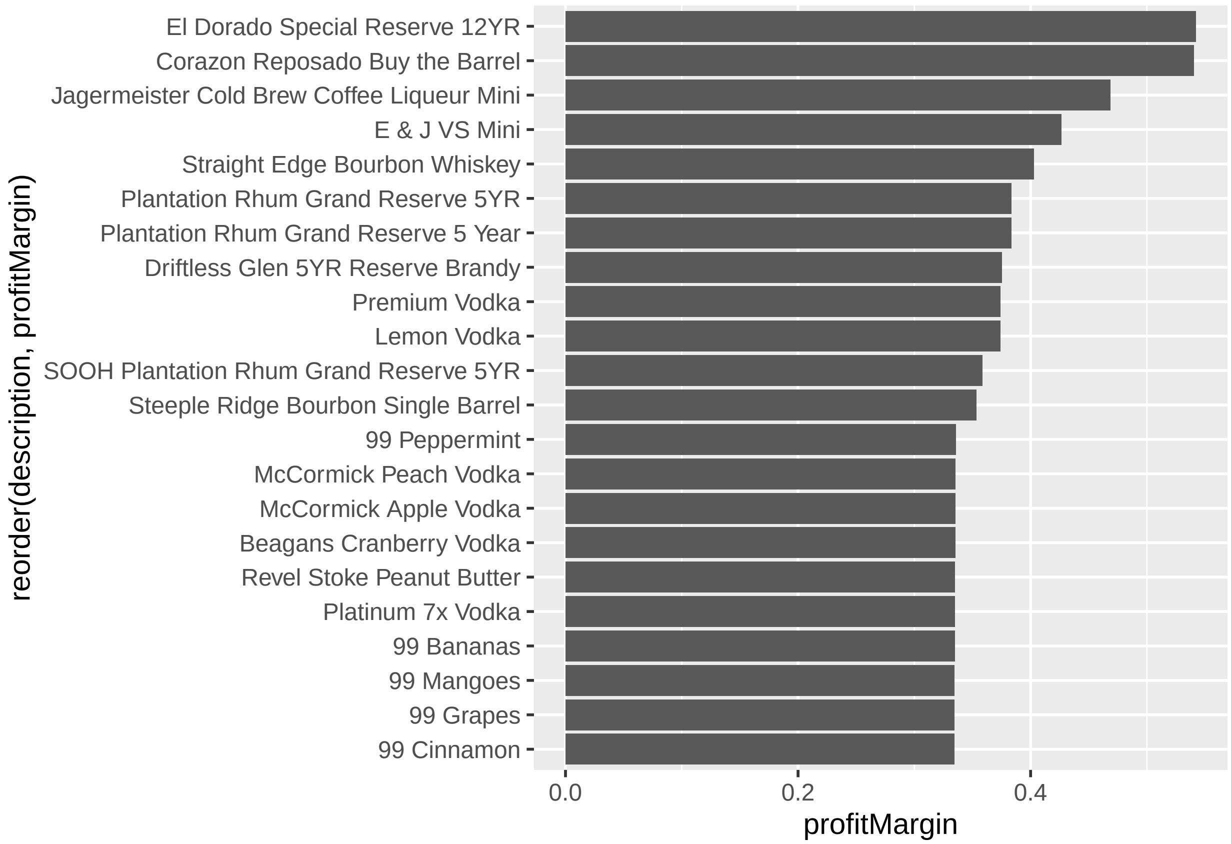


```
liquor %>%
  mutate(profit = retail - cost,
         # totalProfit = profit*numSold,
         profitMargin = profit/retail) %>%
  group_by(description) %>%
  summarise(profitMargin = mean(profitMargin)) %>%
  top_n(n = 15,wt = profitMargin) %>%
  ggplot(aes(x=reorder(description,profitMargin),y=profitMargin)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_bw() +
  labs(y = "Profit Margin",
       x = "Vendor") +
  scale_y_continuous(labels = scales::percent)
```

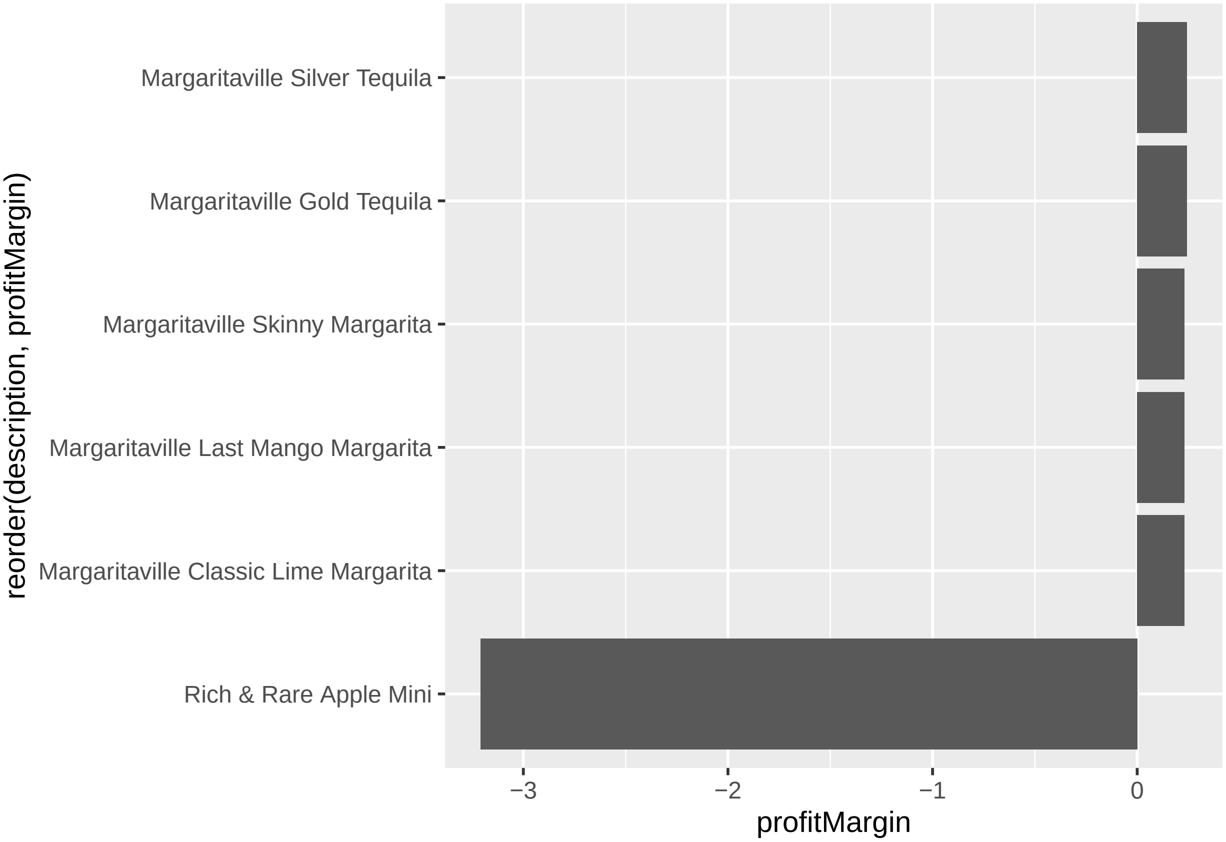


The plots below explore the most and least profitable liquors (per unit)

```
# most profitable
liquor %>%
  mutate(profit = retail - cost,
         profitMargin = profit/retail) %>%
  select(description,profitMargin) %>%
  distinct() %>%
  group_by(description) %>%
  summarise(profitMargin = mean(profitMargin)) %>%
  top_n(n = 20,wt = profitMargin) %>%
  ggplot(aes(x=reorder(description,profitMargin),y=profitMargin)) +
  geom_bar(stat = "identity") +
  coord_flip()
```



```
# least profitable
liquor %>%
  mutate(profit = retail - cost,
         profitMargin = profit/retail) %>%
  select(description,profitMargin) %>%
  distinct() %>%
  group_by(description) %>%
  summarise(profitMargin = mean(profitMargin)) %>%
  top_n(n = 5,wt = -profitMargin) %>%
  ggplot(aes(x=reorder(description,profitMargin),y=profitMargin)) +
  geom_bar(stat = "identity") +
  coord_flip()
```

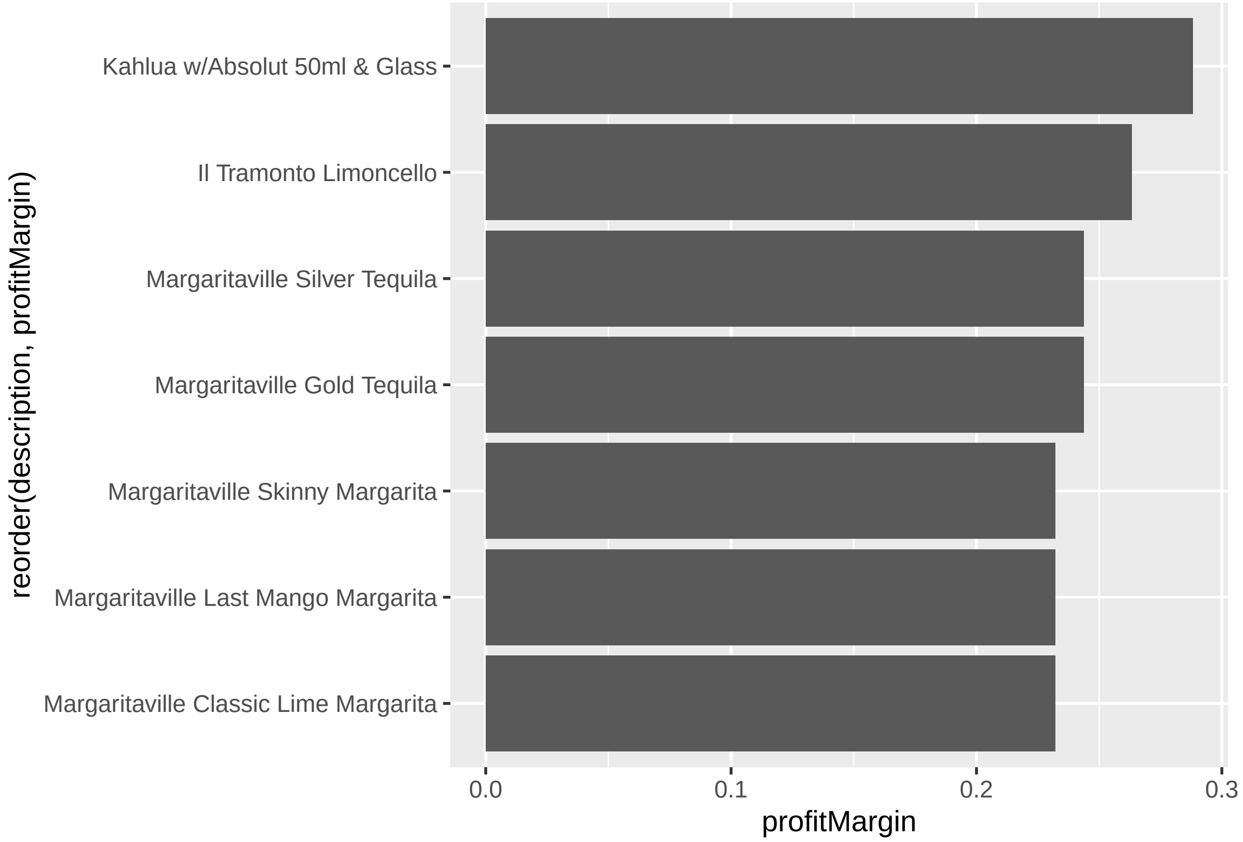


You'll notice in the least profitable plot that the 'Rich & Rare Apple Mini' is suspiciously not profitable (over -300%). Looking deeper into this liquor, it appears that there is a recording error where some of the costs were recorded as $60 based on the output below. Removing these outliers makes a nicer plot.

```
liquor %>%
  filter(description == "Rich & Rare Apple Mini") %>%
  select(storeName,City,Date,cost,retail) %>%
  arrange(storeName)
```

```
## # A tibble: 53 x 5
##    storeName                                City         Date        cost retail
##    <chr>                                    <chr>        <date>     <dbl>  <dbl>
##  1 Brother's Market / Clarion               Clarion      2021-07-07  5.16   7.74
##  2 Casey's General Store #2481 / Bloomfield Bloomfield   2021-07-05  5.16   7.74
##  3 Casey's General Store #2644 / Earlham    Earlham      2021-01-12  5.16   7.74
##  4 Casey's General Store #3098 / WDM        West Des Mo~ 2021-11-02 60      7.74
##  5 Central City Liquor, Inc.                Des Moines   2021-03-02 60      7.74
##  6 Central City Liquor, Inc.                Des Moines   2021-08-03 60      7.74
##  7 Central City Liquor, Inc.                Des Moines   2021-08-04  5.16   7.74
##  8 Central City Liquor, Inc.                Des Moines   2021-01-06  5.16   7.74
##  9 Central City Liquor, Inc.                Des Moines   2021-12-08  5.16   7.74
## 10 East End Liquor / Des Moines             Des Moines   2021-01-04  5.16   7.74
## # ... with 43 more rows
## # i Use `print(n = ...)` to see more rows
```

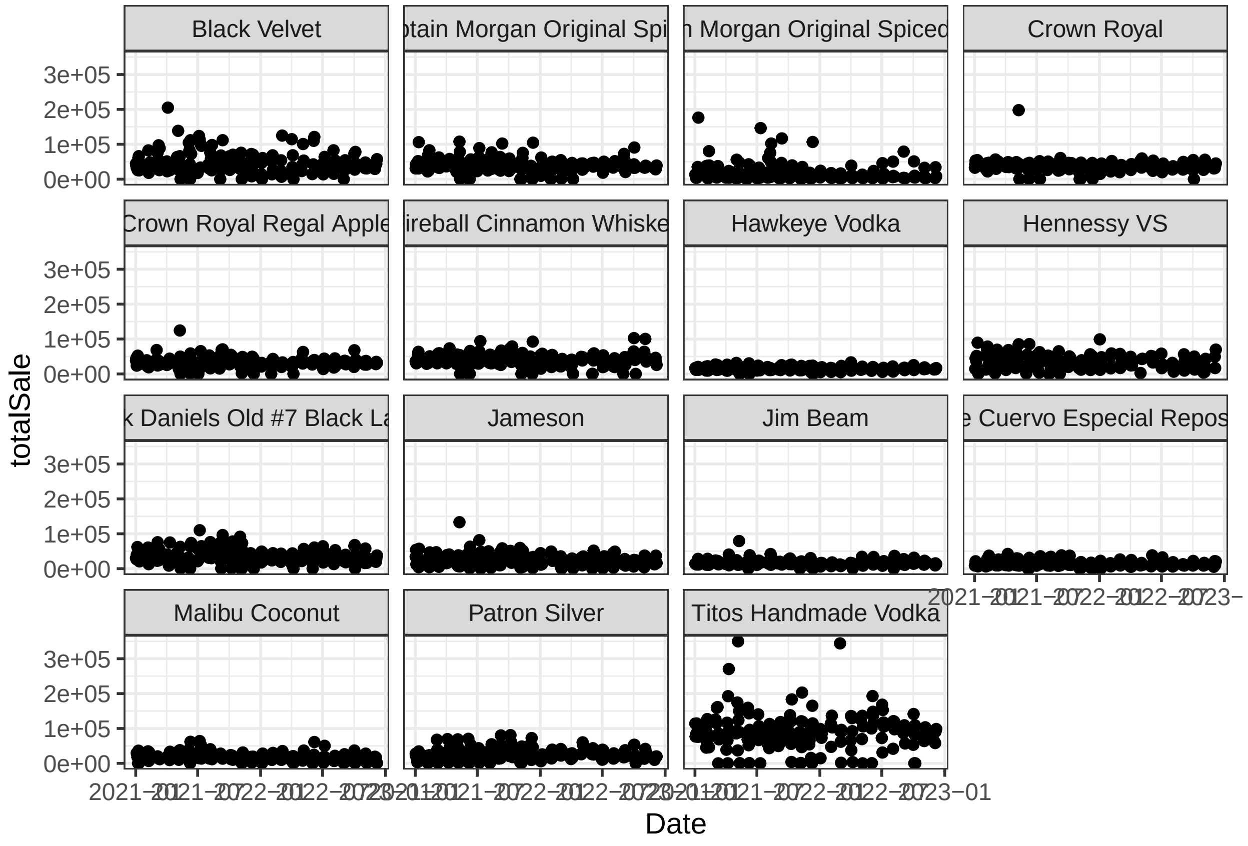
```r
# least profitable, fixed
liquor %>%
  filter(!(description == "Rich & Rare Apple Mini" & cost == 60)) %>%
  mutate(profit = retail - cost,
         profitMargin = profit/retail) %>%
  select(description,profitMargin) %>%
  distinct() %>%
  group_by(description) %>%
  summarise(profitMargin = mean(profitMargin)) %>%
  top_n(n = 7,wt = -profitMargin) %>%
  ggplot(aes(x=reorder(description,profitMargin),y=profitMargin)) +
  geom_bar(stat = "identity") +
  coord_flip()
```



Temporal data

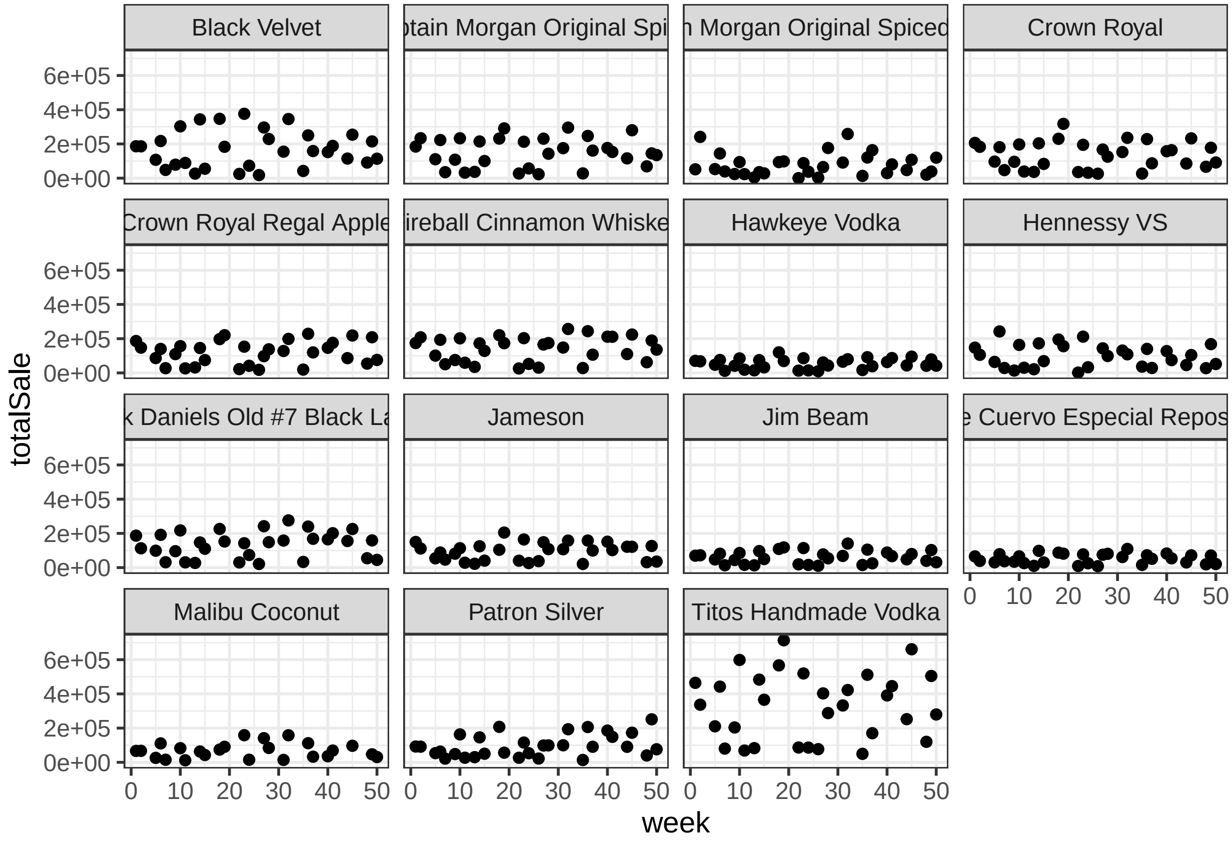
```r
popularLiquors <- liquor %>%
  group_by(description) %>%
  summarise(totalSale = sum(saleTotal)) %>%
  top_n(15,wt = totalSale) %>%
  pull(description)

liquor %>%
  filter(description %in% popularLiquors) %>%
  group_by(description,Date) %>%
  summarise(totalSale = sum(saleTotal)) %>%
  ggplot(aes(x=Date,y = totalSale)) +
  geom_point() +
  facet_wrap(~description) +
  theme_bw()
```

```
## `summarise()` has grouped output by 'description'. You can override using the
## `.groups` argument.
```



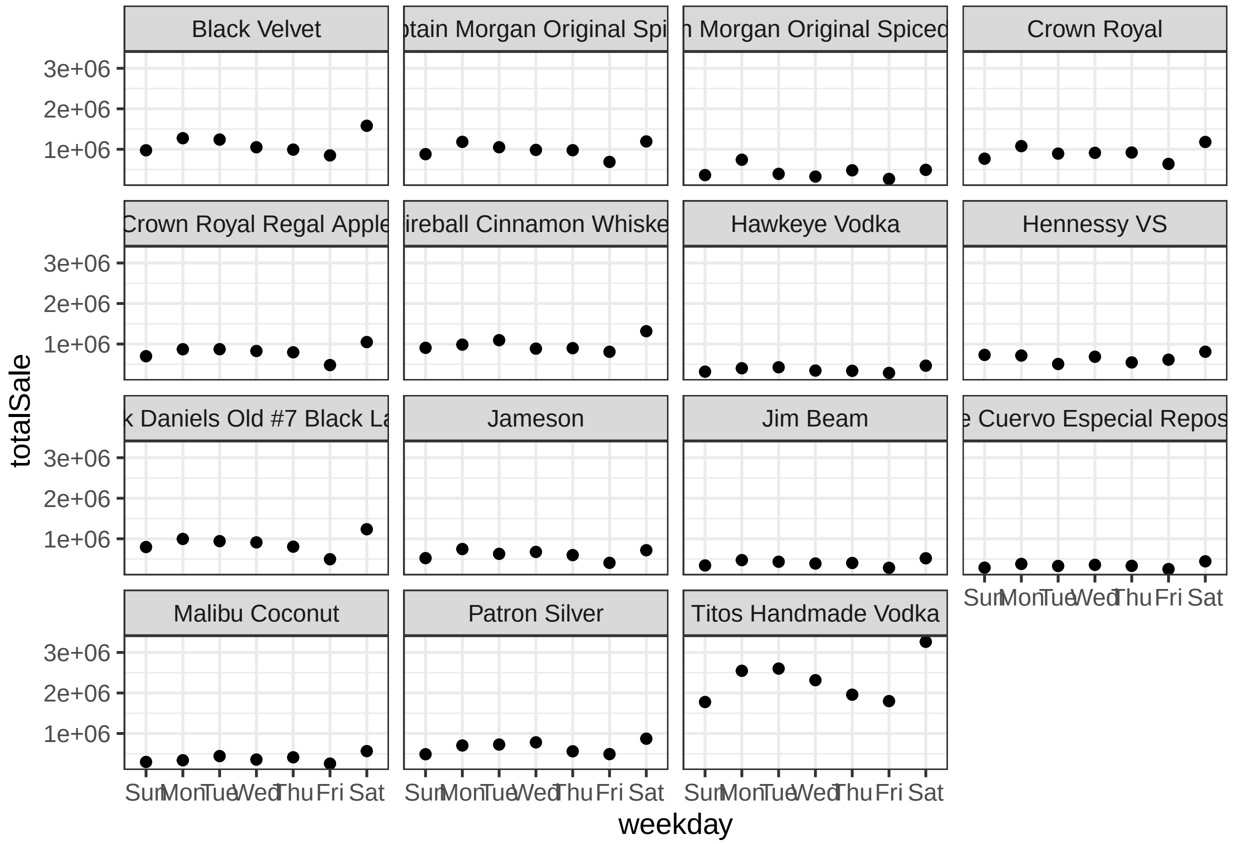
```
liquor %>%
  filter(description %in% popularLiquors & lubridate::year(Date) == 2021) %>%
  mutate(week = lubridate::week(Date)) %>%
  group_by(description,week) %>%
  summarise(totalSale = sum(saleTotal)) %>%
  ggplot(aes(x=week,y = totalSale)) +
  geom_point() +
  facet_wrap(~description) +
  theme_bw()
```

```
## `summarise()` has grouped output by 'description'. You can override using the
## `.groups` argument.
```



```
liquor %>%
  filter(description %in% popularLiquors) %>%
  mutate(weekday = lubridate::wday(Date,label = TRUE)) %>%
  group_by(description,weekday) %>%
  summarise(totalSale = sum(saleTotal)) %>%
  ggplot(aes(x=weekday,y = totalSale)) +
  geom_point() +
  facet_wrap(~description) +
  theme_bw()
```

```
## `summarise()` has grouped output by 'description'. You can override using the
## `.groups` argument.
```



Spatial

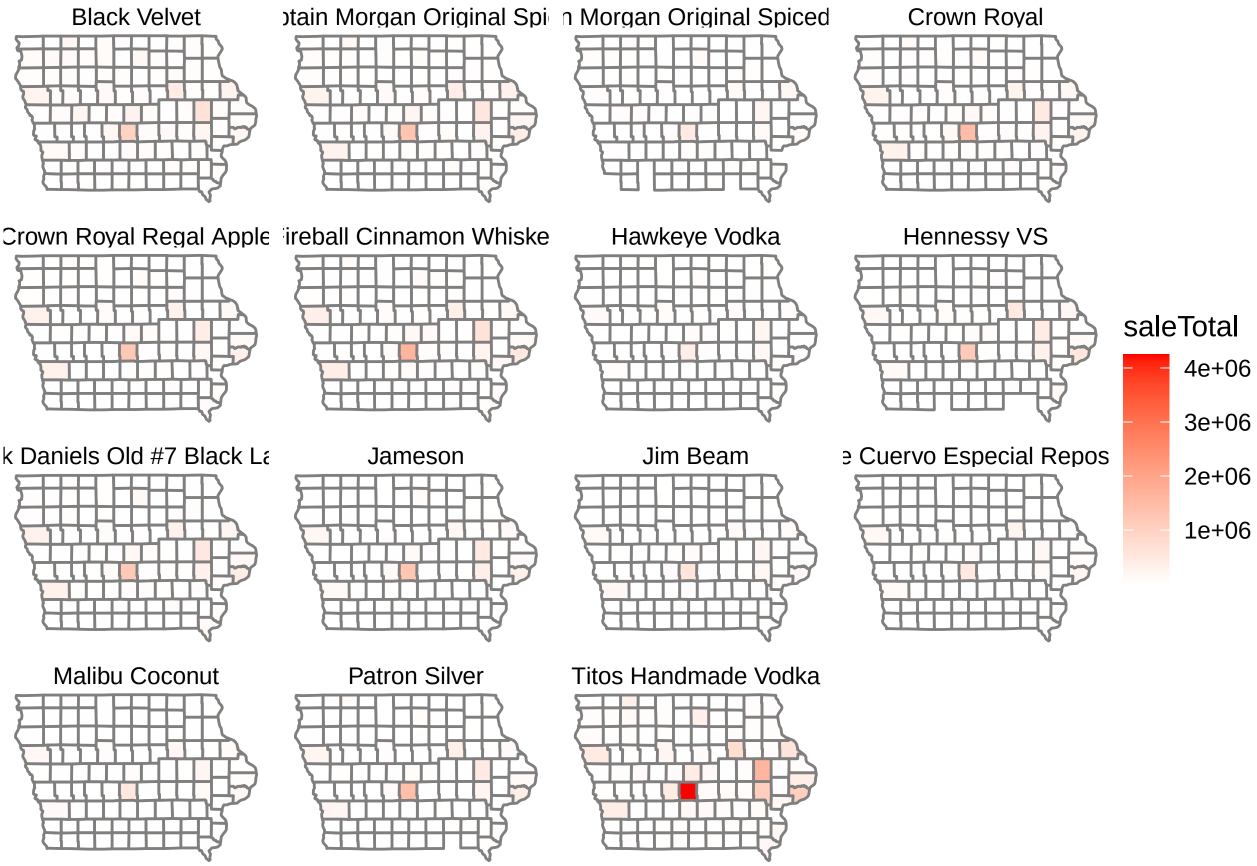
```
iowaCounty <- ggplot2::map_data(map = "county",region = "iowa")

iowaCounty %>%
  mutate(subregion = toupper(subregion)) %>%
  head()
```

```
##        long      lat group order region subregion
## 1 -94.24583 41.50506     1     1   iowa     ADAIR
## 2 -94.24583 41.16129     1     2   iowa     ADAIR
## 3 -94.24583 41.16129     1     3   iowa     ADAIR
## 4 -94.48647 41.16129     1     4   iowa     ADAIR
## 5 -94.70992 41.16129     1     5   iowa     ADAIR
## 6 -94.70992 41.50506     1     6   iowa     ADAIR
```

```
liquor <- liquor %>%
  mutate(County = ifelse(County == "BUENA VIST","BUENA VISTA",
                         ifelse(County == "CERRO GORD","CERRO GORDO",
                                ifelse(str_detect(toupper(County),pattern = "POTTAWATTA"),"POTTAWATTAMI

liquorFiltered <- liquor %>%
  mutate(County = toupper(County)) %>%
  filter(description %in% popularLiquors) %>%
  group_by(description,County) %>%
  summarise(saleTotal = sum(saleTotal))
```

```
## `summarise()` has grouped output by 'description'. You can override using the
## `.groups` argument.
```

```
iowaCounty %>%
  mutate(subregion = toupper(subregion)) %>%
  left_join(y = liquorFiltered,
            by = c("subregion" = "County")) %>%
  ggplot(aes(x = long,y = lat)) +
  geom_polygon(aes(group = group,fill = saleTotal),
               colour = "gray50") +
  facet_wrap(~description) +
  theme_void() +
  scale_fill_gradient(low = "white",high = "red")
```



```
unique(iowaCounty$subregion)
```

```
##  [1] "adair"        "adams"        "allamakee"    "appanoose"
##  [5] "audubon"      "benton"       "black hawk"   "boone"
##  [9] "bremer"       "buchanan"     "buena vista"  "butler"
## [13] "calhoun"      "carroll"      "cass"         "cedar"
## [17] "cerro gordo"   "cherokee"      "chickasaw"     "clarke"
## [21] "clay"          "clayton"       "clinton"       "crawford"
## [25] "dallas"        "davis"         "decatur"       "delaware"
## [29] "des moines"    "dickinson"     "dubuque"       "emmet"
## [33] "fayette"       "floyd"         "franklin"      "fremont"
## [37] "greene"        "grundy"        "guthrie"       "hamilton"
## [41] "hancock"       "hardin"        "harrison"      "henry"
## [45] "howard"        "humboldt"      "ida"           "iowa"
## [49] "jackson"       "jasper"        "jefferson"     "johnson"
## [53] "jones"         "keokuk"        "kossuth"       "lee"
## [57] "linn"          "louisa"        "lucas"         "lyon"
## [61] "madison"       "mahaska"       "marion"        "marshall"
## [65] "mills"         "mitchell"      "monona"        "monroe"
## [69] "montgomery"    "muscatine"     "obrien"        "osceola"
## [73] "page"          "palo alto"     "plymouth"      "pocahontas"
## [77] "polk"          "pottawattamie" "poweshiek"     "ringgold"
## [81] "sac"           "scott"         "shelby"        "sioux"
## [85] "story"         "tama"          "taylor"        "union"
## [89] "van buren"     "wapello"       "warren"        "washington"
## [93] "wayne"         "webster"       "winnebago"     "winneshiek"
## [97] "woodbury"      "worth"         "wright"
```

```
unique(liquor$County)
```

```
##   [1] "CRAWFORD"      "CARROLL"       "LINN"          "WOODBURY"
##   [5] "POLK"          "BOONE"         "BLACK HAWK"    "MADISON"
##   [9] "BUENA VISTA"   "HOWARD"        "SCOTT"         "JASPER"
##  [13] "SIOUX"         "OBRIEN"        "DELAWARE"      "CEDAR"
##  [17] "JOHNSON"       "MARION"        "JACKSON"       "DUBUQUE"
##  [21] "STORY"         "WARREN"        "MARSHALL"      "WINNEBAGO"
##  [25] "LEE"           "WAPELLO"       "CLAY"          "PLYMOUTH"
##  [29] "WRIGHT"        "DALLAS"        "POTTAWATTAMIE" "LUCAS"
##  [33] "KEOKUK"        "HUMBOLDT"      "CLINTON"       "IOWA"
##  [37] "CERRO GORDO"   "MUSCATINE"     "DES MOINES"    "MILLS"
##  [41] "MONTGOMERY"    "TAYLOR"        "UNION"         "POWESHIEK"
##  [45] "BENTON"        "ALLAMAKEE"     "APPANOOSE"     "DAVIS"
##  [49] "BUCHANAN"      "FLOYD"         "SAC"           "CHEROKEE"
##  [53] "CALHOUN"       "BUTLER"        "GRUNDY"        "HARDIN"
##  [57] "KOSSUTH"       "MAHASKA"       "WASHINGTON"    "HAMILTON"
##  [61] "BREMER"        "AUDUBON"       "LYON"          "CASS"
##  [65] "JEFFERSON"     "WEBSTER"       "PALO ALTO"     "PAGE"
##  [69] "JONES"         "OSCEOLA"       "IDA"           "HARRISON"
##  [73] "MONONA"        "FAYETTE"       NA              "CHICKASAW"
##  [77] "TAMA"          "CLARKE"        "WINNESHIEK"    "WORTH"
##  [81] "SHELBY"        "FRANKLIN"      "MITCHELL"      "GREENE"
##  [85] "CLAYTON"       "WAYNE"         "DICKINSON"     "HANCOCK"
##  [89] "HENRY"         "EMMET"         "VAN BUREN"     "GUTHRIE"
##  [93] "MONROE"        "LOUISA"        "ADAIR"         "DECATUR"
##  [97] "POCAHONTAS"    "FREMONT"       "RINGGOLD"      "ADAMS"
```

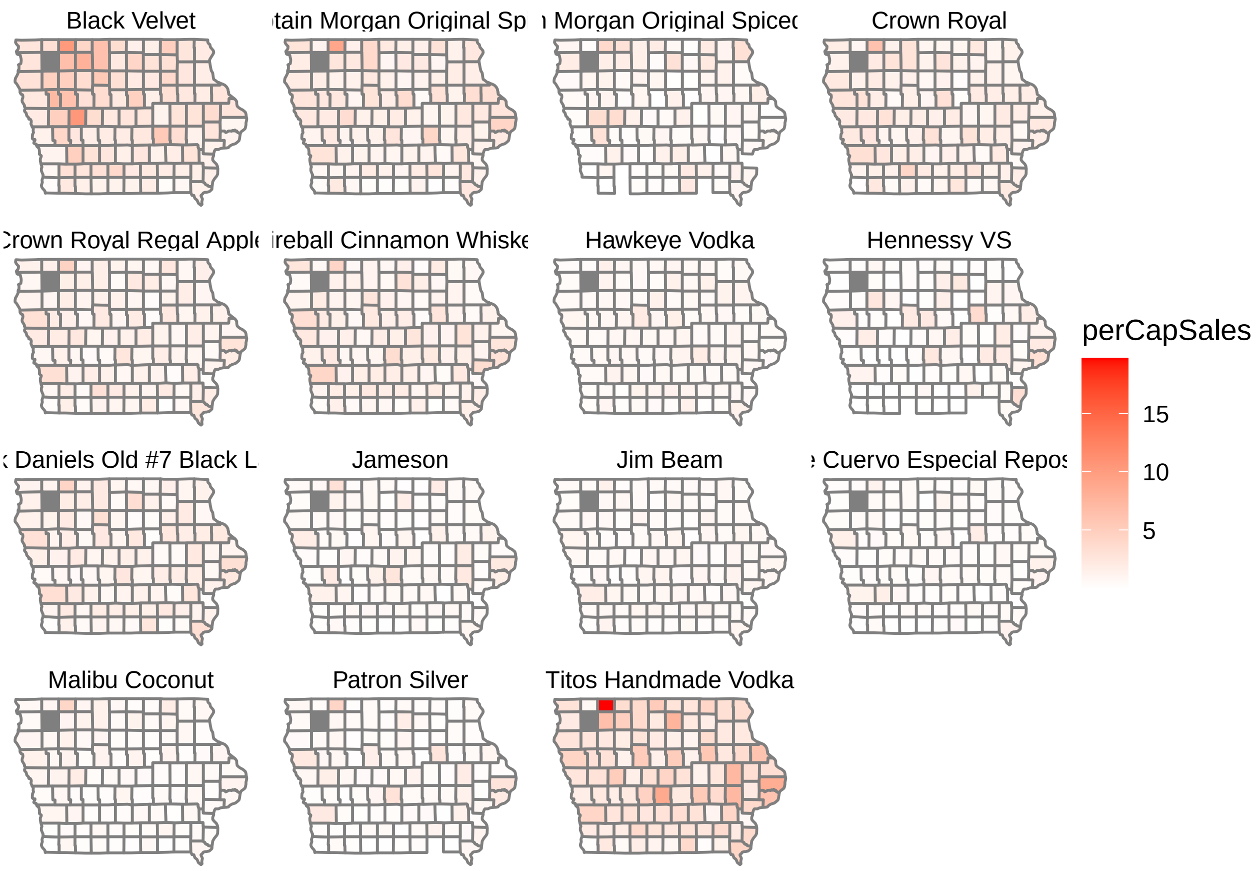
```
library(rvest)
```

```
##
## Attaching package: 'rvest'
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
iowaCountyPop <- rvest::read_html("https://www.iowa-demographics.com/counties_by_population") %>%
  rvest::html_element("table") %>%
  rvest::html_table()

iowaCountyPop <- iowaCountyPop %>%
  select(County,Population) %>%
  mutate(County = County %>%
           str_remove(pattern = " County") %>%
           toupper(),
         Population = Population %>%
           str_remove(pattern = ",") %>%
           as.numeric()) %>%
  slice(-100)
```

```
## Warning in Population %>% str_remove(pattern = ",") %>% as.numeric(): NAs
## introduced by coercion
```

```
iowaCounty %>%
  mutate(subregion = toupper(subregion)) %>%
  left_join(y = liquorFiltered,
            by = c("subregion" = "County")) %>%
  left_join(iowaCountyPop,
            by = c("subregion" = "County" )) %>%
  mutate(perCapSales = saleTotal/Population) %>%
  ggplot(aes(x = long,y = lat)) +
  geom_polygon(aes(group = group,fill = perCapSales),
               colour = "gray50") +
  facet_wrap(~description) +
  theme_void() +
  scale_fill_gradient(low = "white",high = "red")
```



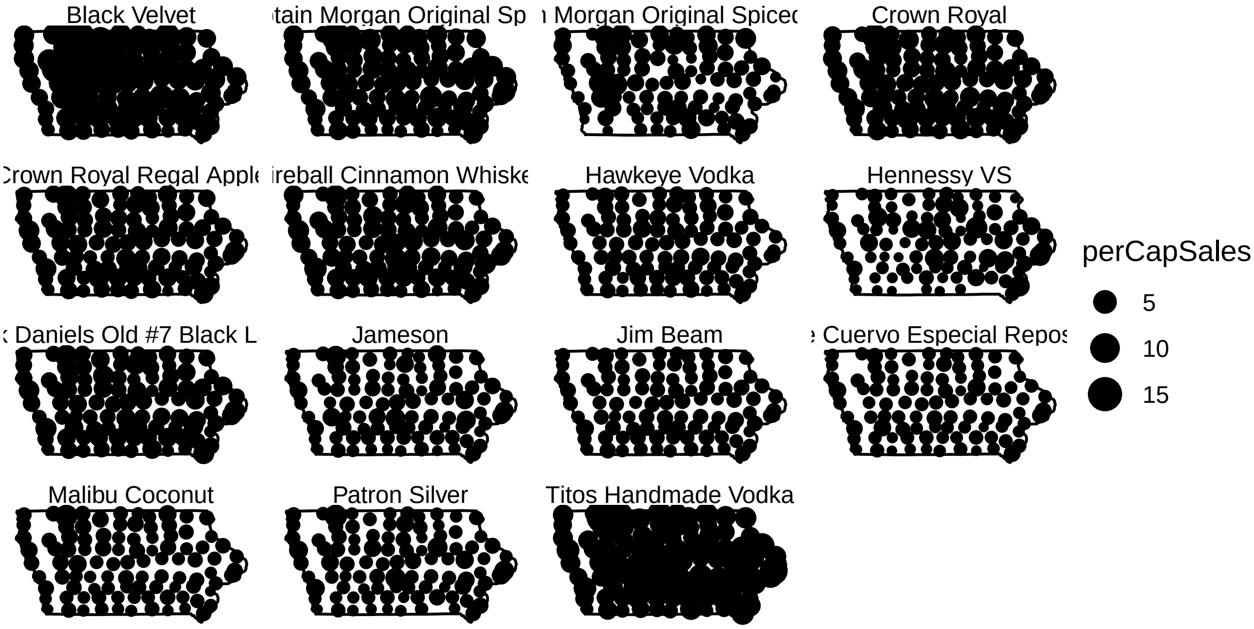
```
write_csv(x = liquor,file = "liquorCleaned.csv")
write_csv(x = iowaCountyPop,file = "iowaCountyPop.csv")
```

Below is an alternative visualization that draws individual stores as points and maps the variable of interest (e.g., per capita sales in $) to the size of each point

```
iowaState <- map_data(map = "state",region = "iowa")

iowaCounty %>%
  mutate(subregion = toupper(subregion)) %>%
  left_join(y = liquorFiltered,
            by = c("subregion" = "County")) %>%
  left_join(iowaCountyPop,
            by = c("subregion" = "County" )) %>%
  mutate(perCapSales = saleTotal/Population)  %>%
  group_by(subregion,description) %>%
  summarise(countyCenter_long = mean(long),
            countyCenter_lat = mean(lat),
            perCapSales = unique(perCapSales)) %>%
  ggplot(aes(x=countyCenter_long,y=countyCenter_lat)) +
  geom_point(aes(size = perCapSales)) +
  geom_path(data = iowaState,
            aes(x = long,y = lat),
            inherit.aes = FALSE) +
  facet_wrap(~description) +
  coord_fixed(xlim = c(-96.639704,-90.140061),
              ylim = c(40.375501,43.501196)) +
  theme_void()
```

```
## `summarise()` has grouped output by 'subregion'. You can override using the
## `.groups` argument.
```

```
## Warning: Removed 15 rows containing missing values (geom_point).
```



Market Basket Analysis

The first 6 digits in the invoice code detail an individual purchaser - for example, 339132 in the table below - followed by 5 digits that detail the individual item purchased, probably associated with a unique SKU.

This means we can analyze which items tend to be purchased together. This is useful in a marketing capacity to build something called a 'recommender system.' If you've ever seen on an online store page a menu that says 'Customers who bought [x] also bought [y],' then you've witnessed a recommender system at-work. Other terms for this type of analysis are 'Market Basket Analysis' or 'Association Rule Learning.'

```
liquor %>%
  filter(Date == "2021-01-02") %>%
  arrange(invoice) %>%
  slice(7:14)
```

```
## # A tibble: 8 x 19
##   invoice      Date       store~1 Address City    zip  long   lat County categ~2
##   <chr>        <date>     <chr>   <chr>   <chr> <dbl> <dbl> <dbl> <chr>  <chr>
## 1 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Americ~
## 2 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Blende~
## 3 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Straig~
## 4 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Americ~
## 5 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Americ~
## 6 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Americ~
## 7 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Whiske~
## 8 INV-3391320~ 2021-01-02 Marsha~ 11 N 3~ Mars~ 50158 -92.9  42.1 MARSH~ Scotch~
## # ... with 9 more variables: vendor <chr>, description <chr>, Pack <dbl>,
## #   bottleVolume <dbl>, cost <dbl>, retail <dbl>, numSold <dbl>,
## #   saleTotal <dbl>, saleVolume <dbl>, and abbreviated variable names
## #   1: storeName, 2: category
## # i Use `colnames()` to see all variable names
```

We will perform a very rudimentary market basket analysis. Note that there are more robust techniques available in R packages such as `arules` that you may be interested in exploring.

For the sake of an example, we'll consider association rules between Tito's Handmade Vodka and various liquors. When we go to implement this in the app, we'll allow the user to select a specific liquor other than Tito's, but we need to start somewhere.

Support

First, let's calculate the support between Tito's and each liquor. For liquor Y, the support is the probability of observing a sale that includes both Tito's and Y.

$$Support(Titos, Y) = P(Titos, Y) = \frac{\text{\# sales with Titos and Y}}{\text{total \# sales}}$$

We first search the data set for all purchases that included Tito's Handmade Vodka. We filter the liquor data set to only these purchases and save this to `salesIncludingTitos`.

```r
liquor <- liquor %>%
  mutate(saleID = str_sub(invoice,5,10))

allSaleID <- unique(liquor$saleID)

titosSaleID <- liquor %>%
  select(saleID,description) %>%
  filter(description == "Titos Handmade Vodka") %>%
  distinct() %>%
  pull(saleID) %>%
  unique()

salesIncludingTitos <- liquor %>%
  filter(saleID %in% titosSaleID) %>%
  select(saleID,description) %>%
  arrange(saleID) %>%
  distinct()

salesIncludingTitos
```

```
## # A tibble: 912,556 x 2
##    saleID description
##    <chr>  <chr>
##  1 331682 Southern Comfort
##  2 331682 Red Stag Black Cherry
##  3 331682 Jack Daniels Old #7 Black Label
##  4 331682 Titos Handmade Vodka
##  5 331682 Bacardi Limon
##  6 331696 Corralejo Reposado
##  7 331696 Knob Creek
##  8 331696 Smirnoff Spicy Tamarind
##  9 331696 Dr McGillicuddys Apple Pie
## 10 331696 Arrow Mcdales Butterscotch Schnapps
## # ... with 912,546 more rows
## # i Use `print(n = ...)` to see more rows
```

To calculate support, we first tally the number of sale IDs that included each type of liquor in the salesIncludingTitos data set. This is equivalent to calculating the numerator of the support. Then, we simply divide by the total number of sale IDs in the data set.

Considering the output below, Tito's Handmade Vodka unsurprisingly has the largest support with itself. You'll notice that the liquors with the highest support with Tito's also happen to be the most popular liquors overall. The support is affected by the overall prevalence of each liquor in the data set - more popular liquors will naturally have higher support – so it's not really the most insightful statistic. Support by itself provides the "no duh" associations between items like "customers who bought eggs also bought bread." However, we will use the support as a building block to calculating the lift.

```
titosSupport <- salesIncludingTitos %>%
  group_by(description) %>%
  summarise(n = n()) %>%
  mutate(support = n/length(allSaleID))

titosSupport %>%
  arrange(desc(support))
```

```
## # A tibble: 2,819 x 3
##    description                     n support
##    <chr>                       <int>   <dbl>
##  1 Titos Handmade Vodka        24655   0.396
##  2 Black Velvet                15498   0.249
##  3 Fireball Cinnamon Whiskey   14072   0.226
##  4 Hawkeye Vodka               13319   0.214
##  5 Captain Morgan Original Spiced 11850   0.190
##  6 Crown Royal                  9772   0.157
##  7 Crown Royal Regal Apple      9211   0.148
##  8 Smirnoff 80prf               8885   0.143
##  9 Jim Beam                     8666   0.139
## 10 Seagrams 7 Crown             8544   0.137
## # ... with 2,809 more rows
## # i Use `print(n = ...)` to see more rows
```

Confidence

Next, we'll consider the confidence: the probability that a sale included liquor Y *given* (denoted by a vertical bar |) that it included Tito's.

$$Confidence(Y|Titos) = P(Y|Tito's) = \frac{P(Y, Tito's)}{P(Tito's)} = \frac{Support(Y, Tito's)}{\frac{\text{\# sales with Tito's}}{\text{total \# sales}}}$$

We already have calculated the numerator of the confidence for each liquor by calculating the support. Thus, we simply need to divide the support column by the probability that a sale included Tito's.

Considering the output below, we see that the order of the highest confidence liquors is the same as the highest support liquor. Thus, confidence is also affected by the prevalence of liquors in the data set – more popular liquors will naturally have a higher support/confidence than less popular liquors. The confidence of Tito's with itself is obviously 1.

```r
titosConfidence <- titosSupport %>%
  mutate(confidence = support/(length(titosSaleID)/length(allSaleID)))

titosConfidence %>%
  arrange(desc(confidence))
```

```
## # A tibble: 2,819 x 4
##    description                        n support confidence
##    <chr>                          <int>   <dbl>      <dbl>
##  1 Titos Handmade Vodka           24655   0.396      1
##  2 Black Velvet                   15498   0.249      0.629
##  3 Fireball Cinnamon Whiskey      14072   0.226      0.571
##  4 Hawkeye Vodka                  13319   0.214      0.540
##  5 Captain Morgan Original Spiced 11850   0.190      0.481
##  6 Crown Royal                     9772   0.157      0.396
##  7 Crown Royal Regal Apple         9211   0.148      0.374
##  8 Smirnoff 80prf                  8885   0.143      0.360
##  9 Jim Beam                        8666   0.139      0.351
## 10 Seagrams 7 Crown                8544   0.137      0.347
## # ... with 2,809 more rows
## # i Use `print(n = ...)` to see more rows
```

Lift

Finally, we'll consider the lift of liquor Y given Tito's. Think of this as a measure of association between the occurrence of Tito's and of liquor Y. The larger the lift, the more often liquor Y and Tito's are observed together **after accounting for the frequency of both liquors.** As opposed to support/confidence, which can be deceptive if two liquors are overall very popular in a data set (e.g., Tito's and Black Velvet), lift accounts for the rarity of both liquors and can therefore uncover associations that support/confidence do not.

$$Lift(Y|Tito's) = \frac{P(Y|Tito's)}{P(Y)} = \frac{Confidence(Y|Tito's)}{\frac{\text{\# sales including liquor Y}}{\text{total \# sales}}}$$

Again, we have already calculated the numerator of the lift in calculating the confidence. We then divide the confidence by the probability of observing liquor Y in the data set – this is what we mean by "taking into account the frequency of both liquors." The `liquorYFrequency` data set contains the number of sales that include each liquor. We use this in calculating the denominator of the lift.

```
liquorYFrequency <- liquor %>%
  select(saleID,description) %>%
  distinct() %>%
  group_by(description) %>%
  tally(name = "freq")

liquorYFrequency
```

```
## # A tibble: 4,048 x 2
##    description                             freq
##    <chr>                                  <int>
##  1 135<U+FFFD> East Hyogo Japanese Dry Gin          71
##  2 173 Craft Distillery Broken Beaker Rum     1
##  3 173 Craft Distillery Volumetric Vodka      1
##  4 1792 12YR Old Bourbon                    102
##  5 1792 Bottled in Bond Bourbon              44
##  6 1792 Full Proof                          315
##  7 1792 Full Proof Buy the Barrel             3
##  8 1792 Sweet Wheat Bourbon                  93
##  9 1800 Anejo                               281
## 10 1800 Coconut                             335
## # ... with 4,038 more rows
## # i Use `print(n = ...)` to see more rows
```

We see that the liquors with the highest associated lift with Tito's are rather different than the highest support/confidence. If one were to develop a recommender system for Tito's purchasers, it seems reasonable to recommend liquors that have the highest lift and, amongst these liquors, those with the highest support/confidence.

```
titosConfidence %>%
  left_join(liquorYFrequency,
            by = "description") %>%
  mutate(lift = confidence/(freq/length(allSaleID))) %>%
  select(-c(support,n,freq)) %>%
  arrange(desc(lift),desc(confidence)) %>%
  # filter(description != "Titos Handmade Vodka") %>%
  top_n(n = 30,wt = lift)
```

```
## # A tibble: 31 x 3
##    description                                         confidence  lift
##    <chr>                                                    <dbl> <dbl>
##  1 Titos Handmade Vodka                                 1          2.53
##  2 Luster Lavender Limoncello                           0.000284   2.53
##  3 RumHaven VAP                                         0.000203   2.53
##  4 GOTCHA BLENDED WHISKEY 750ML                         0.000162   2.53
##  5 Dueces Wild Vodka                                    0.000122   2.53
##  6 Smirnoff Zero Sugar Infusion Watermelon & Mint Mini  0.000122   2.53
##  7 Belvedere Pure Cutting Board VAP                     0.0000811  2.53
##  8 Century Farms Captains Vodka                         0.0000811  2.53
##  9 Kentucky Tavern                                      0.0000811  2.53
## 10 Luksusowa Potato Vodka                               0.0000811  2.53
## # ... with 21 more rows
## # i Use `print(n = ...)` to see more rows
```