Data Preparation

Iowa Liquor Sales

Hailey Ernest, James Irwin, Elliot Kunz, Kaatje Matthews-vanKoetsveld

2025-10-03

Data Collection

The data was collected from Iowa Liquor Sales using the API Socrata, it was filtered by transaction dates after the year 2020. For each client a different subset of the data was filtered to help them individually.

For Client A we collected "date" to extract the year, store "name" and "store" count for total bottles ordered per store, "bottle_volume_ml" and "state_bottle_cost" to feature engineer average bottle cost, "sale_bottles" the number of bottles sold, and "category_name" for the type of liquor.

For Client B we collected "date" to extract the year, "county" where the store sold liquor, "sale_liters" total volume sold, "category_name" for the type of liquor to find sale trends by category, and "sale_dollars" total cost. We collected an additional dataset of population that will be joined by county, we will analyse trends of liquor sold by county population.

Population: For Client B, we will be working with county data. Therefore, we have brought in population data by county by year for Iowa to the most current year (2025). The populations of cities are calculated every year by the state of Iowa government. To use this for data analysis, we will aggregate the city populations in each county to get a total population per county. https://catalog.data.gov/dataset/city-population-in-iowa-by-county-and-year

Feature Engeenering

As mention in the Data Collection section, the team will be keeping the aforementioned variables from the original dataset for purposes of analysis in their respective project. All other factors from the original dataset are not included.

Parsing out Company Name

In the given dataset, store name includes the name (Kum & Go, Hy-Vee, ect.), possibly the store number, and possibly the location of the store in one variable. To aggregate on store chains, we pulled out just the company name.

```
client_a <- read.csv(here::here("client_a_booze_r_us.csv"))

client_a2  <- client_a |>
  mutate(
    company = sub(" [0-9]+",
                  "",
                  sub("#.*",
                      "" ,
                      sub("/.*", "",name))),
    company = trimws(company, which="right")
  )

write.csv(client_a2, here::here("client_a_booze_r_us_transformed_.csv"))
```

Reducing Alcohol Category Type

The 'Category Name' field contains the category of the alcohol bought. These categories are overly specific, so we have reduced them into broader categories for easier aggregation. For example: "WHITE RUM" and "SPICED RUM" are both put in the "RUM" alcohol category. It is important to note that as of now, not all sales are placed into a reduced category. We have only parsed out the alcohol type for the 10 most common types in the dataset.

```
client_b <- read.csv(here::here("client_b_dead.csv"))

client_b2 <- client_b |>
  mutate(
    alcohol_type = case_when(
      str_detect(category_name, "WHISK") ~ "WHISKEY",
      str_detect(category_name, "VODKA") ~ "VODKA",
      str_detect(category_name, "RUM") ~ "RUM",
      str_detect(category_name, "TEQUILA") ~ "TEQUILA",
      str_detect(category_name, "SCHNAPPS") ~ "SCHNAPPS",
      str_detect(category_name, "GIN") ~ "GIN",
      str_detect(category_name, "BRANDIES") ~ "BRANDY",
      str_detect(category_name, "LIQUEURS") ~ "LIQUEUR",
```

```
      str_detect(category_name, "COCKTAIL") ~ "COCKTAILS",
      str_detect(category_name, "SCOTCH") ~ "SCOTCH",
      TRUE ~ NA
    )
  )

write.csv(client_b2, here::here("client_b_dead_transformed_.csv"))
```

Population by County by Year

Using the Iowa population data we've procured, we have just taken the relevant columns for all observations since 2020, and have summed the population by city to get population by county per year. This will be merged with the liquor sales by county per year so we can use it as a factor in our modeling.

```
population <- read.csv(here::here("City_Population_in_Iowa_by_County_and_Year.csv"))

population2 <- population |>
  select(County, City, Year, Estimate) |>
  mutate(DATE = as.Date(Year, format="%B %d %Y"),
         Year = year(DATE)) |>
  group_by(County, Year) |>
  summarize(Population = sum(Estimate)) |>
  filter(Year >= 2020)

write.csv(population2, here::here("Population_in_Iowa_Transformed.csv"))
```

Aggregations

For Booze 'R' Us, we aggregated transaction information by company group for each year between 2020-2025.

```
# Group and aggregate columns for Booze R Us
data = pd.read_csv("client_a_booze_r_us_transformed.csv")
data['date'] = pd.to_datetime(data['date'], errors='coerce')
data['date'] = data['date'].dt.strftime('%Y-%m')
grouped_data_booze = data.groupby(['company', 'date']).agg(
    sum_sale_dollars = ('sale_dollars', 'sum'),
    avg_sale_dollars = ('sale_dollars', 'mean'),
    bottles_sold = ('sale_bottles', 'sum'),
```

```
    num_transactions = ('name', 'count'),
    num_stores = ('store', 'nunique')
)

grouped_data_booze.to_csv("grouped_data_booze.csv", index=True)
```

For DEAD, we aggregated transaction information by county by alcohol type for each year between 2020-2025.

```
# Group and aggregate columns for DEAD
data = pd.read_csv("client_b_dead_transformed.csv")
pop_data = pd.read_csv('Population_in_Iowa_Transformed.csv')
data['county'] = data['county'].astype(str).str.strip().str.lower()
pop_data['County'] = pop_data['County'].astype(str).str.strip().str.lower()
data['date'] = pd.to_datetime(data['date'], errors='coerce')
data['date'] = data['date'].dt.strftime('%Y').astype(int)
combined_data =  pd.merge(data, pop_data, left_on = (['county', 'date']), right_on= (['County
grouped_data_dead = combined_data.groupby(['county', 'date','alcohol_type']).agg(
    sum_sale_dollars = ('sale_dollars', 'sum'),
    avg_sale_dollars = ('sale_dollars', 'mean'),
)

grouped_data_dead.to_csv("grouped_data_dead.csv", index=True)
```