Summer Research Project

Yi (Max) Ren

2023-12-05

Data Cleaning

The data from three points in time was read in and hence cleaned. Over the three-year period, some item may have become unavailable and the closest substitute items were recorded to make up for this. This requires us to clean the data before conducting analysis.

```r
ww_Jul <- read.csv("WW_2023Jul.csv")
coles_Jul <- read.csv("Coles_2023Jul.csv")
ww_Dec <- read.csv("WW_2023Dec.csv")
coles_Dec <- read.csv("Coles_2023Dec.csv")
data_2020 <- read.csv("2020Dec.csv")
```

We mainly focus on direct matches so that any substituted items need to be filtered out. Two data sets (July and December 2023) are created, containing all direct matches from Cole's and Woolworth,

```r
match_coles_Jul <- coles_Jul %>%
  filter(is.na(substitution) & !is.na(price))
match_ww_Jul <- ww_Jul %>%
  filter(is.na(substitution) & !is.na(price))
matched_Jul <- rbind(match_coles_Jul, match_ww_Jul)

match_coles_Dec <- coles_Dec %>%
  filter(is.na(substitution) & !is.na(price))
match_coles_Dec <- match_coles_Dec[,1:14]
match_ww_Dec <- ww_Dec %>%
  filter(is.na(substitution) & !is.na(price))
matched_Dec <- rbind(match_coles_Dec, match_ww_Dec)
```

The category and desc columns are extracted from the 2020 data set and added to December 2023 data set for future inflation analysis.

```r
data_2020 <- data_2020 %>%
  group_by(brand, item_name, category, desc)%>%
  summarise(
    min_price = min(product_cost),
    max_price = max(product_cost),
    min_product_size = min(product_size),
    max_product_size = max(product_size)
  ) %>%
  ungroup() %>%
  select(category, desc, brand, item_name, min_price, max_price,
         min_product_size, max_product_size)

matched_Dec <- matched_Dec %>%
  left_join(data_2020)
```

For prices and product sizes in 2020, the max and min were recorded during data collection. This creates confusions in later analysis so we simply take out the observations where there are distinctive min and max prices in 2020 (very few observations). By observing the data set, there are a number of NAs in the product size column, as some products are sold not according to weight. The unit price cannot be calculated without the product sizes, hence, we remove the observations with NAs.

```
matched_Dec <- matched_Dec %>%
  filter(min_price == max_price) %>%
  filter(min_product_size == max_product_size) %>%
  filter(is.na(product_size) == F) %>%
  select(category, desc, brand, item_name, min_price, min_product_size,
         store, price, product_size)
colnames(matched_Dec) <- c("category", "desc", "brand", "item_name",
                           "price_20", "size_20", "store", "price_Dec", "size_Dec")
matched_Dec$size_Dec <- as.numeric(matched_Dec$size_Dec)
```

Larger Categories Added

After browsing through the data set, it is noticed that there are more than 50 categories of grocery products. This is helpful when we want to look at specific groups of items. However, when it comes to visualization of our findings, confusions will arise. Therefore, we added a new column named large-category, using the categorization of grocery product from the ABS's CPI report. This additional method categorizes the products into six groups, namely beverages, bread and cereal products, dairy and related product, food product n.e.c, fruit and vegetables, and, meat and seafood. This will help with the visualization of data, and as it aligns with the ABS's categorization, can be compared to the ABS's CPI data.

Overall Inflation Calculations (Dec 2020 vs Dec 2023)

```
matched_Dec <- matched_Dec %>%
  mutate(unitprice_20 = price_20*100/size_20,
         unitprice_Dec = price_Dec*100/size_Dec,
         inflation = unitprice_Dec/unitprice_20 - 1)
```

```
inflation_summary <- matched_Dec %>%
  group_by(category)%>%
  summarise(mean_inflation = mean(inflation, na.rm = TRUE))

mean(matched_Dec$inflation)
```

```
## [1] 0.2939225
```

```
summary(matched_Dec$inflation)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.58484  0.05799  0.25000  0.29392  0.42857  2.46154
```

The overall inflation rate of the groceries from Woolworth and Cole's between December 2020 and December 2023 is approximately 29.4%. The greatest price increase occurs for the dips, which increased about 187.9% over three years. Another category that has increased significantly is milk and dairy products, this includes reduced fat or skim milk, full fat milk, cream, and cheese, with inflation rates 70.2%, 61,6%, 54% and 41.9% respectively.

The plot below shows the relationship between the unit price in 2020 and the inflation rate they have over the past three years (Figure 1). Slight negative trends are shown which suggest that the higher-priced items tend to have a lower inflation rate and lower-priced items tend to have a greater inflation. However, the difference is not significant as the slopes of the trendllines are quite small.

```
ggplot(matched_Dec, aes(x = unitprice_20, y = inflation)) +
  geom_point() + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

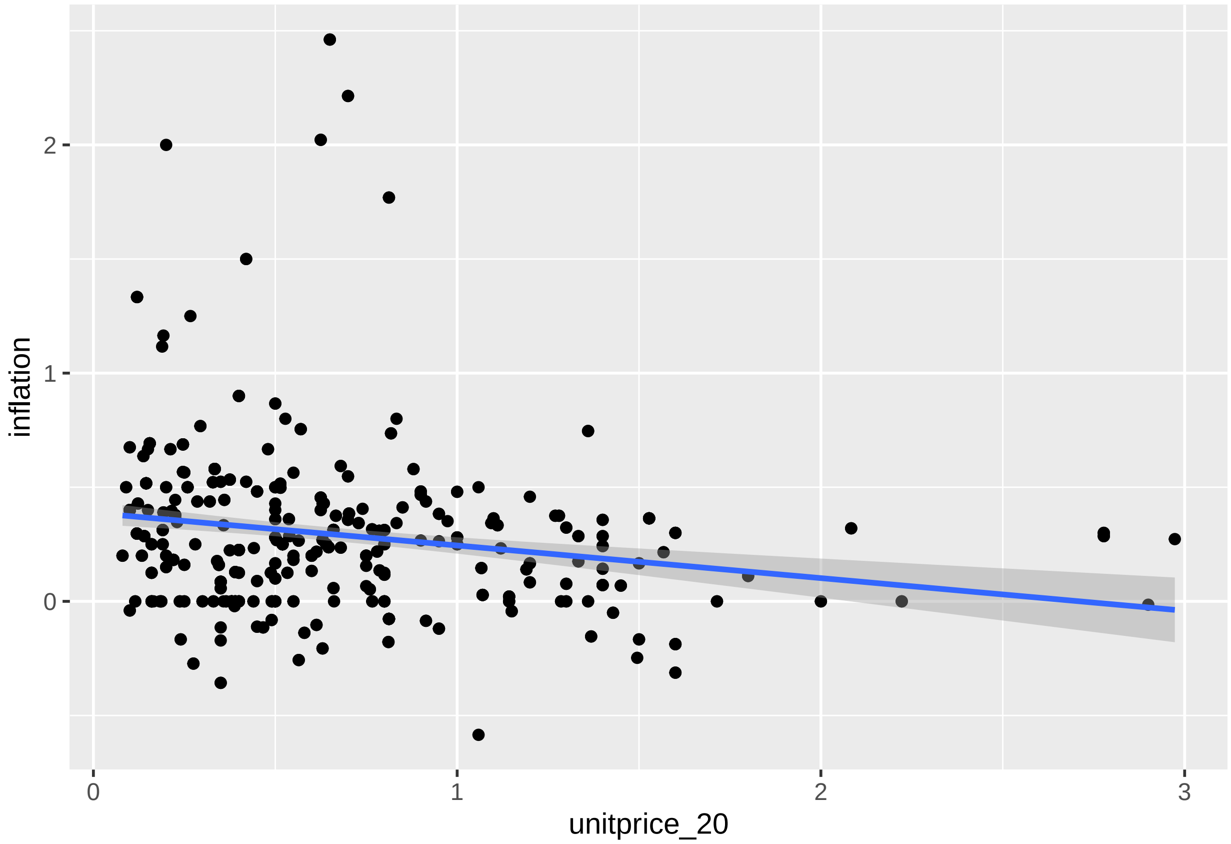

Figure 1: Unit price in 2020 vs the inflation rate over three years

As mentioned earlier, the larger categories from ABS are used here to show a sense of clustering (Figure 2). It is obvious that dairy and related products, and food product n.e.c have the greatest inflation rate. It is also noticed that items from the fruit and vegetables category scattered in the bottom left corner, as they have the least inflation rate and they tend to be the cheapest items among all. This result aligns with the column graph that we created to show the inflation rates of the categories (Figure 3).

```
ggplot(matched_Dec, aes(x = unitprice_20, y = inflation, col = large_category)) +
  geom_point()
```

```
CPI_category <- matched_Dec %>%
  group_by(large_category) %>%
  summarise(mean_inflation = mean(inflation, na.rm = TRUE))

ggplot(data = CPI_category, aes(x = large_category, y = mean_inflation)) +
  geom_col(fill = "lightblue", width = 0.5) + guides(x = guide_axis(n.dodge = 2))
```

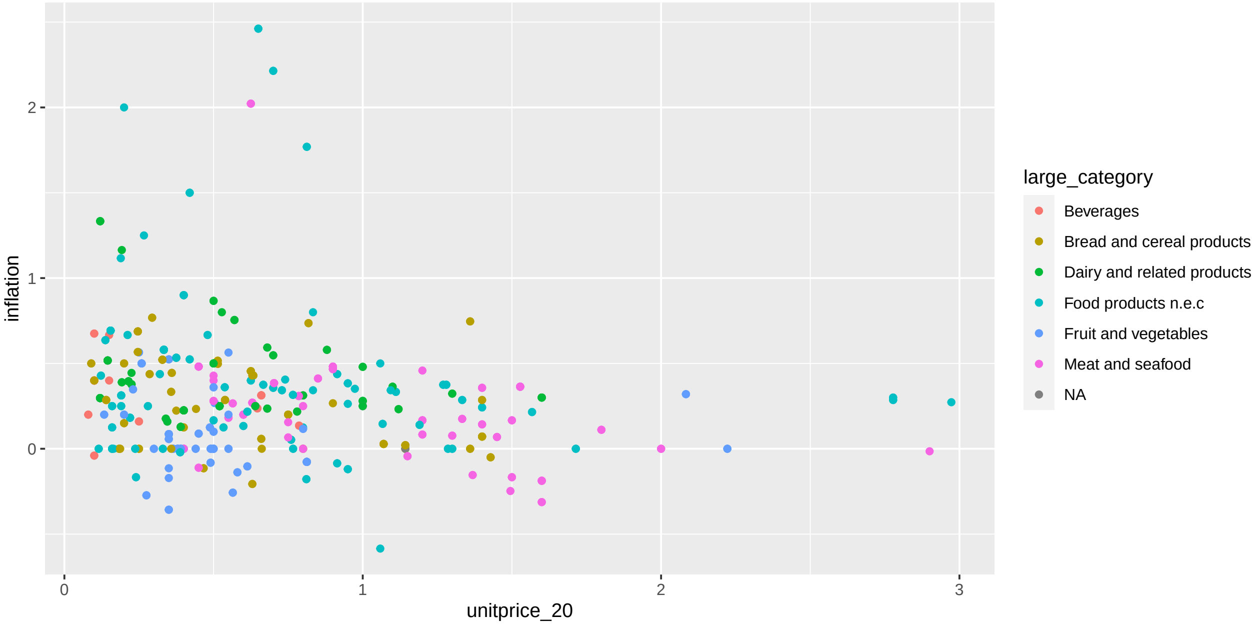

Figure 2: Clustering of items from different categories

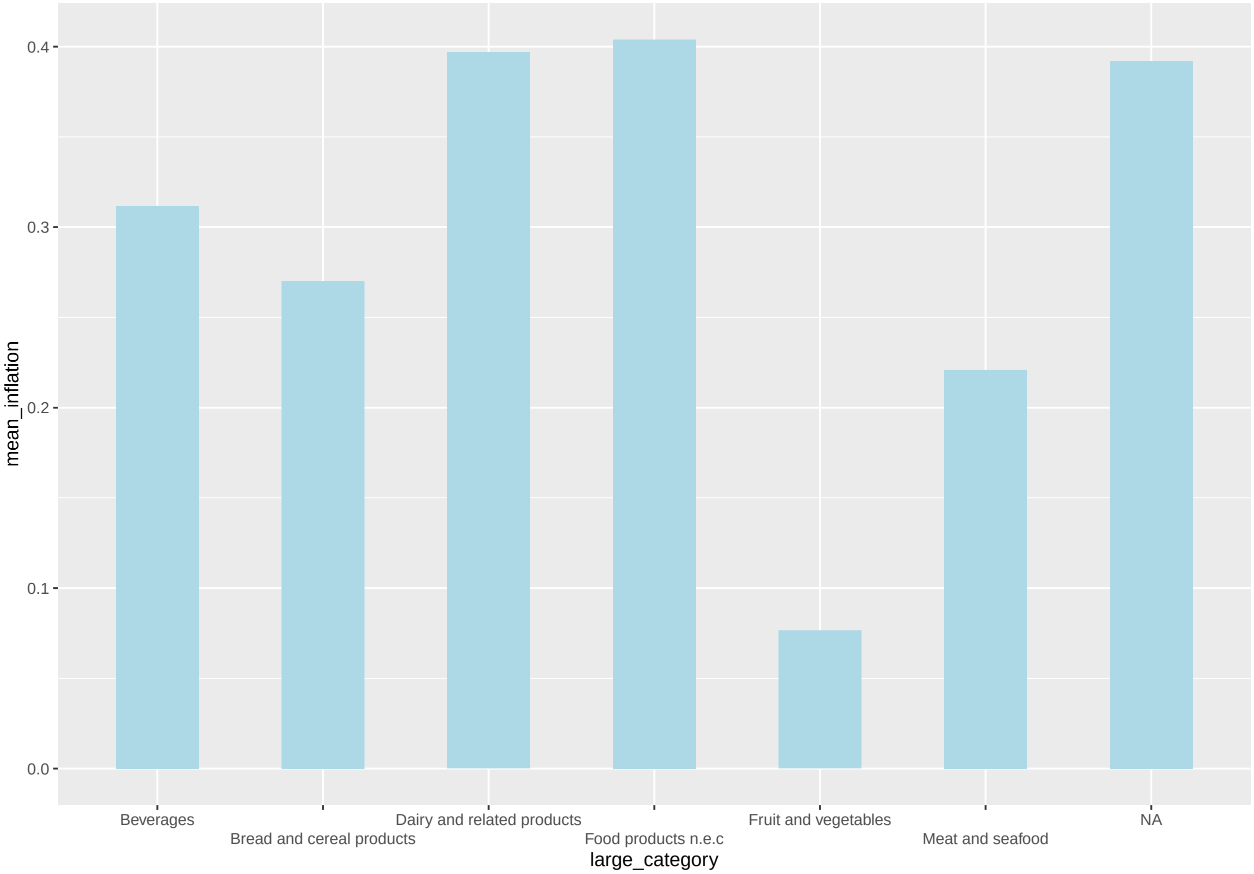

Figure 3: Inflation rate of grocery products over three years (categorized according to the ABS's CPI page)

Half Yearly Inflation Calaulation (Jul 2023 vs Dec 2023)

After some basic data cleaning, we can compare the prices between July and December this year. It is found that the overall inflation rate is around 3.1%, showing a lower rate of increase of prices at recent time.

```
matched_Jul <- matched_Jul %>%
  left_join(matched_Dec) %>%
  filter(!is.na(price_Dec)) %>%
  filter(!is.na(product_size)) %>%
  select(category, desc, brand, item_name, store, price, product_size, price_Dec, size_Dec)
colnames(matched_Jul) <- c("category", "desc", "brand", "item_name", "store", "price_Jul",
                           "size_Jul", "price_Dec", "size_Dec")
```

```
matched_Jul <- matched_Jul %>%
  mutate(unitprice_Jul = price_Jul*100/size_Jul,
         unitprice_Dec = price_Dec*100/size_Dec,
         inflation = unitprice_Dec/unitprice_Jul - 1)

infla_JvD_summary <- matched_Jul %>%
  group_by(category) %>%
  summarise(mean_inflation = mean(inflation, na.rm = TRUE))

mean(matched_Jul$inflation)
```

```
## [1] 0.03058515
```

No-name Brands vs Named Brands

To assess the difference between the price changes of no-name brands and named brands, we need to split the cleaned data set and find the inflation rate for the two groups. The so called no-name brands are the products that Woolworth and Cole's produced and the named brands are the products that third-parties have produced and selling at the supermarkets.

```
named <- matched_Dec %>%
  filter(brand != "Woolworths") %>%
  filter(brand != "Coles")

wool <- matched_Dec %>%
  filter(brand == "Woolworths")
col <- matched_Dec %>%
  filter(brand == "Coles")
no_name <- rbind(wool, col)

infla_named <- named %>%
  mutate(inflation = unitprice_Dec/unitprice_20-1)%>%
  select(brand, item_name, category, desc, inflation, unitprice_Dec, unitprice_20)

infla_no_name <- no_name %>%
  mutate(inflation = unitprice_Dec/unitprice_20-1)%>%
  select(brand, item_name, category, desc, inflation, unitprice_Dec, unitprice_20)

mean(named$inflation)
```

```
## [1] 0.3553381
```

```
mean(no_name$inflation)
```

```
## [1] 0.2604231
```

It is found that the difference of price changes between no-name brands and named brand is quite significant. Products that are produced by Woolworth and Cole's have an overall inflation rate of 26% whereas the named brands have an overall inflation rate of 35.5%.

Store Location

When the price data was originally collected in 2020, it was designed that two supermarkets are located in medium socio-economic status suburbs and the other two are located in low socio-economic status suburbs. The aim was to discover whether the store location influence the price changes. However, due to the nature of data collection in July 2023 and December 2023 (collected online), there appears to have no apparent difference in prices between different store locations in 2023, apart from the unavailability of some products.

```
a <- filter(matched_Dec, store == "Merrylands")
b <- filter(matched_Dec, store == "Auburn")
c <- filter(matched_Dec, store == "East Village")
d <- filter(matched_Dec, store == "Burwood")

low_eco <- rbind(a,b)
med_eco <- rbind(c,d)

infla_loweco <- low_eco %>%
  mutate(inflation = unitprice_Dec/unitprice_20-1)%>%
  select(brand, item_name, category, desc, inflation, unitprice_Dec, unitprice_20)

infla_medeco <- med_eco %>%
  mutate(inflation = unitprice_Dec/unitprice_20-1)%>%
  select(brand, item_name, category, desc, inflation, unitprice_Dec, unitprice_20)

mean(infla_loweco$inflation)
```

```
## [1] 0.2966526
```

```
mean(infla_medeco$inflation)
```

```
## [1] 0.291163
```

Although there is no price difference between store locations for online shopping, the prices were distinctive in 2020 and hence led to a slight difference in inflation rate. It is found that the inflation rates of stores in low socio-economic status suburbs and stores in medium socio-economic status suburbs are 29.7% and 29.1% respectively. There is a slight gap between the low and medium socio-economic status, it is suspected that the grocery prices in the low socio-economic status suburbs was moderately lower, hence a greater inflation rate is observed.

Woolworth's vs Cole's

We would like to discover the journalistic questions: which of Woolworth's and Cole's has increased grocery prices more over three years? We will assess the price changes of the items sold in Woolworth's and Cole's and hence find the inflation rate.

```
match_coles_Dec <- match_coles_Dec %>%
  left_join(data_2020) %>%
  filter(min_price == max_price) %>%
  filter(min_product_size == max_product_size) %>%
  filter(is.na(product_size) == F) %>%
  select(category, desc, brand, item_name, min_price, min_product_size,
         store, price, product_size)
colnames(match_coles_Dec) <- c("category", "desc", "brand", "item_name",
                           "price_20", "size_20", "store", "price_Dec", "size_Dec")
match_coles_Dec$size_Dec <- as.numeric(match_coles_Dec$size_Dec)

match_ww_Dec <- match_ww_Dec %>%
  left_join(data_2020) %>%
  filter(min_price == max_price) %>%
  filter(min_product_size == max_product_size) %>%
  filter(is.na(product_size) == F) %>%
  select(category, desc, brand, item_name, min_price, min_product_size,
         store, price, product_size)
colnames(match_ww_Dec) <- c("category", "desc", "brand", "item_name",
                           "price_20", "size_20", "store", "price_Dec", "size_Dec")

match_ww_Dec <- match_ww_Dec %>%
  mutate(large_category = case_when(
    category %in% names(category_mapping) ~ category_mapping[category],
    TRUE ~ as.character(category)))

match_coles_Dec <- match_coles_Dec %>%
  mutate(large_category = case_when(
    category %in% names(category_mapping) ~ category_mapping[category],
    TRUE ~ as.character(category)))

inflation_ww <- match_ww_Dec %>%
  mutate(unit_price_20 = price_20*100/size_20,
         unit_price_Dec = price_Dec*100/size_Dec,
         inflation = unit_price_Dec/unit_price_20-1)

inflation_coles <- match_coles_Dec %>%
  mutate(unit_price_20 = price_20*100/size_20,
         unit_price_Dec = price_Dec*100/size_Dec,
         inflation = unit_price_Dec/unit_price_20-1)

mean(inflation_ww$inflation)
```

```
## [1] 0.3031337
```

```
mean(inflation_coles$inflation)
```

```
## [1] 0.2803695
```

It is found that Woolworth has an overall inflation rate of 30.3% whereas Cole's has an overall inflation rate of 28.0%. This difference is more significant than the difference between different store locations and do affect people's decisions when it comes to choosing which supermarket to go to. The following graph shows a more detailed inflation rate between the two supermarkets (Figure 4). It is noticed that items that fall into the category food product n.e.c from Woolworth has increased significantly more than Cole's. Moreover, meat and seafood from Cole's has increased more tahn twice as much as Woolworth.

```
WvsC <- rbind(inflation_coles, inflation_ww)

store_mapping <- c("East Village" = "Cole's",
                   "Merrylands" = "Cole's",
                   "Auburn" = "Woolworth",
                   "Burwood" = "Woolworth")

WvsC <- WvsC %>%
  mutate(supermarket = case_when(
    store %in% names(store_mapping) ~ store_mapping[store],
    TRUE ~ as.character(store)))

by_category <- WvsC %>%
  group_by(large_category, supermarket) %>%
  summarise(mean_inflation = mean(inflation, na.rm = TRUE))

ggplot(data = by_category[1:12,], aes(x = large_category, y = mean_inflation, fill = supermarket)) +
  geom_bar(stat = "identity", position = 'dodge') + guides(x = guide_axis(n.dodge = 2))
```

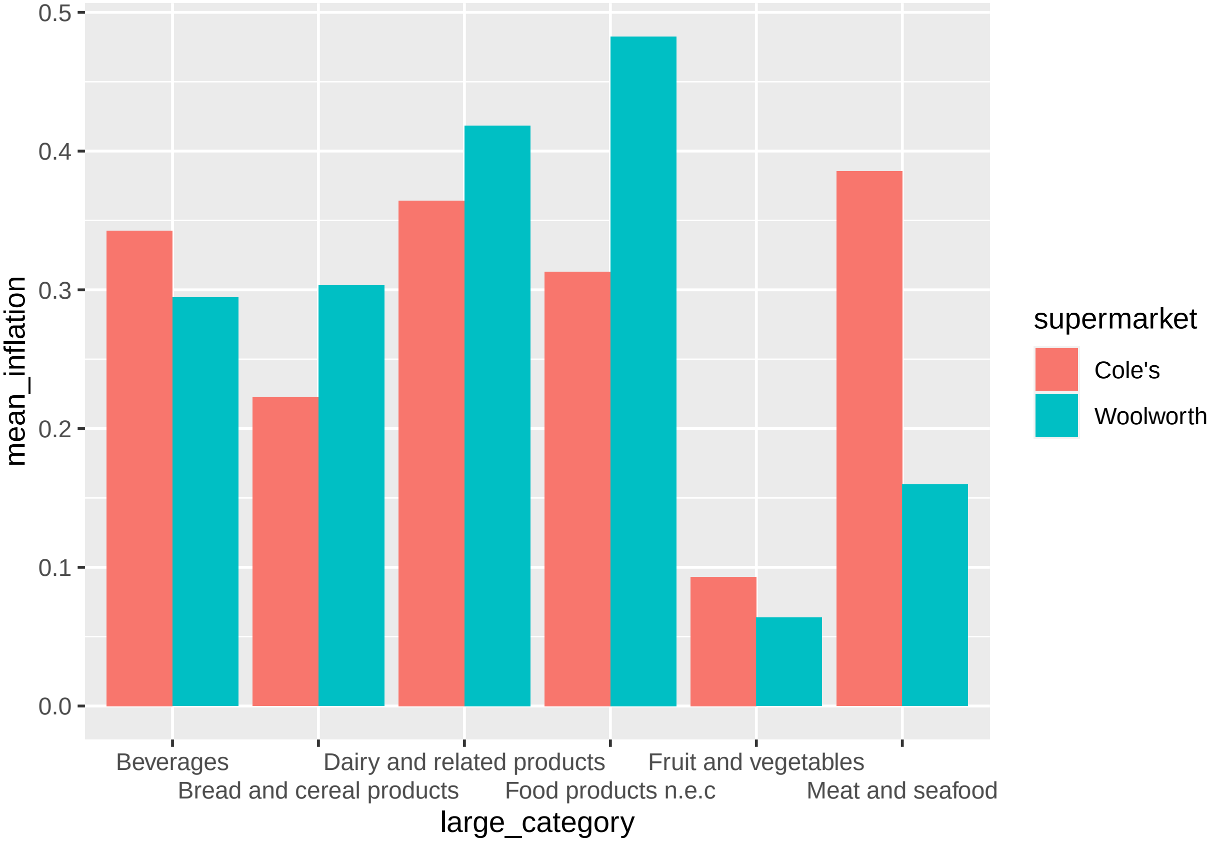

Figure 4: Inflation rate of grocery categories, compared between supermarkets

Average Unit Price in December 2023

After looking at the inflation rate, we will perform a simple analysis on the prices of the grocery items to have a sense of the current grocery prices in the super market. The average price in December 2023 from both supermarkets are calculated where Cole's has a lower average unit price of $0.755 per 100g and Woolworth has a average unit price of $0.856 per 100g.

```
price_ww <- inflation_ww %>%
  group_by(category) %>%
  summarise(mean_unitprice = mean(unit_price_Dec, na.rm = TRUE))
```

```r
price_coles <- inflation_coles %>%
  group_by(category) %>%
  summarise(mean_unitprice = mean(unit_price_Dec, na.rm = TRUE))

mean(inflation_ww$unit_price_Dec)
```

```
## [1] 0.8557573
```

```r
mean(inflation_coles$unit_price_Dec)
```

```
## [1] 0.7546938
```

Result Viewing

This section is solely for viewing the results from previous parts. This includes the data frames containing any results we had in previous part that are too large for viewing in the pdf file created by Markdown. To access them, please uncomment and run the following lines in R Studio.

```r
# Dec 2020 vs Dec 2023 inflation rate for each category
#View(inflation_summary)

# Dec 2020 vs Dec 2023 inflation rate for larger categories (ABS's category)
#View(CPI_category)

# Jul 2023 vs Dec 2023 inflation rate for each category
#View(infla_JvD_summary)

# Comparing inflation rates for larger categories of Woolworth and Cole's
#View(by_category)

# Dataframe containing all valid observations after data cleaning for Dec 2020 vs Dec 2023
#View(matched_Dec)

# Dataframe containing all valid observations after data cleaning for Dec 2020 vs Jul 2023
#View(matched_Jul)

# Dataframe for no-named brand items
#View(infla_no_name)

# Dataframe for named brand items
#View(infla_named)

# Dataframe for stores in low econ status area
#View(infla_loweco)

# Dataframe for store in medium econ status area
#View(infla_medeco)

# Dataframe for Woolworth items
#View(inflation_ww)

# Dataframe for Coles items
#View(inflation_coles)

# average prices of each category for Woolworth in Dec 2023
```

```
#View(price_ww)

# average prices of each category for Cole's in Dec 2023
#View(price_coles)
```