EDLD651_final_project

Kyla & Erika & Cheyna

«««< Updated upstream # Load Packages

======= »»»> Stashed changes

```
library(here)
library(janitor)
library(rio)
library(tidyverse)
library(tm)
library(dplyr)
#install.packages("tm") <- i didn't have this package
```

Import Data

«««< Updated upstream

```
# Fruit data

fruit_files <- list.files(pattern = ".xlsx", path = "../data/fruit-2020", full.names = TRU

fruit_2020 <- do.call(rbind, lapply
        (fruit_files, import, as.is=T, header = FALSE))

# Vegetable Data
veg_files <- list.files(pattern = ".xlsx", path = "../data/vegetables-2020", full.names =

veg_2020 <- do.call(rbind, lapply
        (veg_files, import, as.is=T, header = FALSE))
```

Removing rows without data

```
# Fruit cleaning
# Febe: To make the code cleaner and more readable, I would like to suggest combining the

fruit_2020_clean <- fruit_2020 |>
  rename(
    fruit = ...1,
    retail_price = ...2,
    retail_price_unit = ...3,
    yield = ...4,
    cup_eq_size = ...5,
    cup_eq_unit = ...6,
    cup_eq_price = ...7
  ) |>
  filter(!str_detect(fruit, "^(1|2|3|4|Source)"))


# Veggie cleaning
# Febe: The same here

veg_2020_clean <- veg_2020 %>%
  rename(
    veg = ...1,
    retail_price = ...2,
    retail_price_unit = ...3,
    yield = ...4,
    cup_eq_size = ...5,
    cup_eq_unit = ...6,
    cup_eq_price = ...7
  ) %>%
  filter(!str_detect(veg, "^(1|2|3|4|Source)"))
```

Tidying the data

```
# Febe: I really admire the authors' willingness to tackle such a messy dataset. It requir

# Fruit
fruit_names <- c("Apples", "Apricots", "Bananas", "Mixed berries", "Blackberries", "Bluebe
```

```r
# Febe: We can combine the mutate functions together to make it more visiable
fruit_2020_clean <- fruit_2020_clean |>
  # fix the fruit column to isolate the fruit name
  mutate(fruit = str_extract(fruit, "[^-]+"),
         is_fruit = fruit %in% fruit_names,
         fruit_name = if_else(is_fruit, fruit, NA_character_)) |>
  fill(fruit_name, .direction = "down") |>
  # remove rows where fruit column is the fruit name or "Form"
  filter(!is_fruit, fruit != "Form") |>
  # rename columns
  rename(form = fruit) |>
  # move the fruit column to the beginning
  relocate(fruit_name, .before = "form") |>
  # remove the is_fruit column, we don't need it anymore
  select(-is_fruit) |>
  # remove any rows with NA in the retail_price column
  filter(!is.na(retail_price))

# Vegetable
veg_names <- c("Acorn squash", "Artichoke", "Asparagus", "Avocados", "Beets", "Black beans

veg_2020_clean <- veg_2020_clean |>
  # fix the veg column to isolate the veg name
  mutate(veg = str_extract(veg, "[^-]+"),
         is_veg = veg %in% veg_names,
         veg_name = if_else(is_veg, veg, NA_character_)) |>
  fill(veg_name, .direction = "down") |>
  # remove rows where veg column is the veg name or "Form"
  filter(!is_veg, veg != "Form") |>
  # rename columns
  rename(form = veg) |>
  # move the veg column to the beginning
  relocate(veg_name, .before = "form") |>
  # remove the is_veg column, we don't need it anymore
  select(-is_veg) |>
  # remove any rows with NA in the reatil_price column
  filter(!is.na(retail_price))
```

How does the price of fruits and vegetables vary based on form? (Kyla)

```
# remove the numbers at the end of the forms
fruit_2020_clean$form <- removeNumbers(fruit_2020_clean$form)
veg_2020_clean$form <- removeNumbers(veg_2020_clean$form)

# remove additional specifiers from form column that are not important
fruit_2020_clean$form <- gsub("\\s*\\([^\\)]+\\)", "", fruit_2020_clean$form)
fruit_2020_clean$form <- sub(",.*", "", fruit_2020_clean$form)
veg_2020_clean$form <- sub(" .*$", "", veg_2020_clean$form)
veg_2020_clean$form <- gsub(",", "", veg_2020_clean$form)

# change the tomato forms from "Grape", "Roma", and "Large to all be "Fresh"
# and filter to only look at price differences between fresh, frozen, canned, and dried
veg_2020_clean <- veg_2020_clean |>
  mutate(form = case_when(form == "Canned" ~ "Canned",
                    form == "Grape" ~ "Fresh",
                    form == "Roma" ~ "Fresh",
                    form == "Large" ~ "Fresh",
                    TRUE ~ form)) |>
  filter(form == "Fresh" | form == "Frozen" | form == "Canned" | form == "Dried")

# remove the specific rows for raisins and applesauce because they don't fit cleanly into
# and change "Ready to drink" to "Juice"
fruit_2020_clean <- fruit_2020_clean |>
  filter(form != "Raisins" & form != "Applesauce") |>
  mutate(form = case_when(form == "Ready to drink" ~ "Juice",
                          TRUE ~ form))

# make sure retail_price is a numeric variable
fruit_2020_clean <- fruit_2020_clean |>
  mutate(retail_price = as.numeric(retail_price))
veg_2020_clean <- veg_2020_clean |>
  mutate(retail_price = as.numeric(retail_price))

# examine retail price by form
fruit_2020_clean |>
  group_by(form) |>
  summarize(mean_retail_price = mean(retail_price, na.rm = TRUE))
```

```
# A tibble: 7 x 2
  form                      mean_retail_price
  <chr>                                 <dbl>
1 Canned                                 4.53
2 Dried                                  6.51
3 Fresh                                  2.08
4 Frozen                                 2.37
5 Juice                                  1.35
6 Packed in juice                        1.76
7 Packed in syrup or water               1.75
```

```
veg_2020_clean |>
  group_by(form) |>
  summarize(mean_retail_price = mean(retail_price, na.rm = TRUE))
```

```
# A tibble: 4 x 2
  form   mean_retail_price
  <chr>               <dbl>
1 Canned               1.54
2 Dried                1.65
3 Fresh                1.85
4 Frozen               2.16
```

Do fruit and vegetable prices vary by family (berries, stone fruit etc)? (Erika)

```
#making a new column for the fruit groups -- I can add where I found this info in our intr

# Febe: As a small suggestion, since the fruit and vegetable pipelines are almost identica


fruit_family_data <- fruit_2020_clean %>%
   mutate(
    family = case_when(
      fruit_name %in% c("Peaches", "Plums", "Apricots", "Cherries",
                        "Nectarines", "Mangoes", "Dates") ~ "stone fruit",

      fruit_name %in% c("Blackberries", "Blueberries", "Raspberries",
                        "Strawberries", "Pomegranate", "Bananas", "Kiwi",
```

```
                          "Pineapple", "Cranberries", "Grapes", "Papaya") ~ "berries",

      fruit_name %in% c("Apples", "Pears") ~ "pome",

      fruit_name %in% c("Clementines", "Oranges", "Grapefruit") ~ "citrus",

      fruit_name %in% c("Cantaloupe", "Honeydew melon", "Watermelon") ~ "melon",

      fruit_name %in% c("Mixed berries", "Fruit cocktail") ~ "mixed fruits",

      fruit_name %in% c("Figs") ~ "syconium",

      TRUE ~ "other"
    )
  )

#looking at summary stats by family ... using cup eq. price so we have a relatively simila
str(fruit_family_data$cup_eq_price)
```

```
 chr [1:60] "0.40938921329397582" "0.39020183894115951" ...
```

```
fruit_family_data <- fruit_family_data %>%
  mutate(cup_eq_price = as.numeric(cup_eq_price))

fruit_family_data %>%
  group_by(family) %>%
  summarize(
    mean_price = mean(cup_eq_price, na.rm = TRUE),
    median_price = median(cup_eq_price, na.rm = TRUE),
    sd_price = sd(cup_eq_price, na.rm = TRUE),
    n = n()
  )
```

```
# A tibble: 7 x 5
  family       mean_price median_price sd_price     n
  <chr>             <dbl>        <dbl>    <dbl> <int>
1 berries           0.958        0.905    0.513    24
2 citrus            0.677        0.624    0.266     6
3 melon             0.464        0.424    0.257     3
4 mixed fruits      1.06         1.08     0.125     3
```

```
5 pome              0.679        0.525    0.405     6
6 stone fruit       1.11         1.07     0.562    17
7 syconium          1.18         1.18    NA         1
```

```r
# running a one way anova to see if price (continuous) varies by family (categorical). I f
anova_result <- aov(cup_eq_price ~ family, data = fruit_family_data)
summary(anova_result)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
family       6  2.125  0.3541    1.51  0.193
Residuals   53 12.427  0.2345
```

```r
#plot
plot_table1 <- fruit_family_data %>%
  group_by(family) %>%
  summarize(
    mean_price = mean(cup_eq_price, na.rm = TRUE),
    sd_price = sd(cup_eq_price, na.rm = TRUE),
    n = n(),
    se_price = sd_price / sqrt(n)
  )

ggplot(plot_table1, aes(x = reorder(family, mean_price), y = mean_price)) +
  geom_col(fill = "deeppink4") +
  geom_errorbar(aes(ymin = mean_price - se_price, ymax = mean_price + se_price), width = 0
  labs(x = "Fruit Family", y = "Mean Price per Unit", title = "Mean Fruit Prices by Family
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


```
# doing the same exact thing but for vegetables
veg_family_data <- veg_2020_clean %>%
   mutate(
    family = case_when(
      veg_name %in% c("Broccoli", "Brussels sprouts", "Cabbage", "Cauliflower",
                      "Collard greens", "Kale", "Turnip greens", "Radish", "Mustard gree

      veg_name %in% c("Carrots", "Celery") ~ "carrot family",

      veg_name %in% c("Onions", "Asparagus") ~ "liliceae family",

      veg_name %in% c("Acorn squash", "Cucumbers", "Butternut squash", "Pumpkin", "Zucchin

      veg_name %in% c("Black beans", "Blackeye peas", "Great northern beans", "Green beans

      veg_name %in% c("Tomatoes", "Sweet potatoes", "Red peppers", "Potatoes", "Green pepp

      veg_name %in% c("Okra") ~ "hibiscus family",

      veg_name %in% c("Sweet corn") ~ "grass family",

      veg_name %in% c("Artichoke", "Iceberg lettuce", "Romaine lettuce") ~ "aster family",
```

```
      veg_name %in% c("Spinach", "Beets") ~ "goosefoot family",

      veg_name %in% c("Avocados", "Olives") ~ "botanically considered a fruit",

      veg_name %in% c("Mushrooms") ~ "fungus",

      veg_name %in% c("Mixed vegetables") ~ "mixed",

      TRUE ~ "other"
    )
  )

str(veg_family_data$cup_eq_price)
```

```
 chr [1:77] "1.1633264507895209" "2.2544626846324833" "2.0251409024769567" ...
```

```
veg_family_data <- veg_family_data %>%
  mutate(cup_eq_price = as.numeric(cup_eq_price))

anova_result2 <- aov(cup_eq_price ~ family, data = veg_family_data)
summary(anova_result2) # this one DID have a significant difference (p < 0.001) so i did p
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
family      11  8.882  0.8075   5.277 6.82e-06 ***
Residuals   65  9.946  0.1530
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pairwise.t.test(veg_family_data$cup_eq_price, veg_family_data$family,
                p.adjust.method = "bonferroni")
```

```
    Pairwise comparisons using t tests with pooled SD

data:  veg_family_data$cup_eq_price and veg_family_data$family

                               aster family botanically considered a fruit
botanically considered a fruit 1.00000      -
```

```
cabbage family                0.10115      1.00000
carrot family                 0.60324      1.00000
goosefoot family              1.00000      1.00000
gourd family                  0.58595      1.00000
grass family                  1.00000      1.00000
hibiscus family               1.00000      1.00000
legume family                 0.00483      0.29373
liliceae family               1.00000      1.00000
mixed                         0.28888      1.00000
nightshade family             0.13568      1.00000
                              cabbage family carrot family goosefoot family
botanically considered a fruit -              -             -
cabbage family                -              -             -
carrot family                 1.00000        -             -
goosefoot family              1.00000        1.00000       -
gourd family                  1.00000        1.00000       1.00000
grass family                  1.00000        1.00000       1.00000
hibiscus family               1.00000        1.00000       1.00000
legume family                 1.00000        1.00000       1.00000
liliceae family               0.00041        0.04096       0.18114
mixed                         1.00000        1.00000       1.00000
nightshade family             1.00000        1.00000       1.00000
                              gourd family grass family hibiscus family
botanically considered a fruit -             -            -
cabbage family                -              -            -
carrot family                 -              -            -
goosefoot family              -              -            -
gourd family                  -              -            -
grass family                  1.00000        -            -
hibiscus family               1.00000        1.00000      -
legume family                 1.00000        1.00000      0.44348
liliceae family               0.01156        0.15773      1.00000
mixed                         1.00000        1.00000      1.00000
nightshade family             1.00000        1.00000      1.00000
                              legume family liliceae family mixed
botanically considered a fruit -             -               -
cabbage family                -              -               -
carrot family                 -              -               -
goosefoot family              -              -               -
gourd family                  -              -               -
grass family                  -              -               -
hibiscus family               -              -               -
legume family                 -              -               -
```

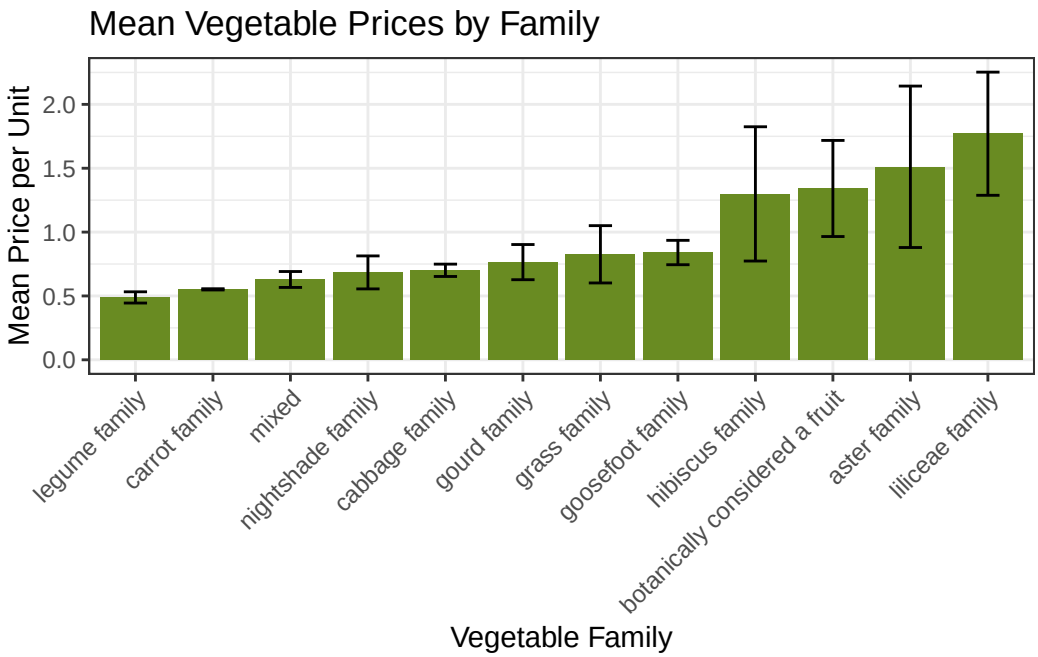
```
liliceae family              6.2e-06       -              -
mixed                        1.00000       0.00706        -
nightshade family            1.00000       0.00095        1.00000

P value adjustment method: bonferroni
```

```r
#confusing to read but the significant groups are: liliceae is signnificantly different fr
#difference mostly because of liliceae (onions + asparagus) being higher in price than oth

# plot
summary_table <- veg_family_data %>%
  group_by(family) %>%
  summarize(
    mean_price = mean(cup_eq_price, na.rm = TRUE),
    sd_price = sd(cup_eq_price, na.rm = TRUE),
    n = n(),
    se_price = sd_price / sqrt(n)
  )

ggplot(summary_table, aes(x = reorder(family, mean_price), y = mean_price)) +
  geom_col(fill = "olivedrab4") +
  geom_errorbar(aes(ymin = mean_price - se_price, ymax = mean_price + se_price), width = 0
  labs(x = "Vegetable Family", y = "Mean Price per Unit", title = "Mean Vegetable Prices b
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



What is the different between retail price for fruits and vegatbles by form? (Cheyna)

```
# examine difference between retail price for fruits and veggies grouped by form
fruit_2020_clean <- fruit_2020_clean %>%
  rename(name = fruit_name) %>% #rename col to join fruit df with veg df
  filter(form == "Fresh"| form == "Frozen" | form == "Canned" |form == "Dried") # select f
  fruit_2020_clean$fruit_veg <- "fruit"

veg_2020_clean <- veg_2020_clean %>%
  rename(name = veg_name)
  veg_2020_clean$fruit_veg <- "veg"

# make fruit and veg df
fruit_veg_df <- bind_rows(fruit_2020_clean, veg_2020_clean)

# test difference in price for fruits and veg
t.test(fruit_2020_clean$retail_price,veg_2020_clean$retail_price)
```

```
    Welch Two Sample t-test

data:  fruit_2020_clean$retail_price and veg_2020_clean$retail_price
t = 3.2614, df = 50.991, p-value = 0.00198
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.4669316 1.9622422
sample estimates:
mean of x mean of y
 3.030853  1.816266
```

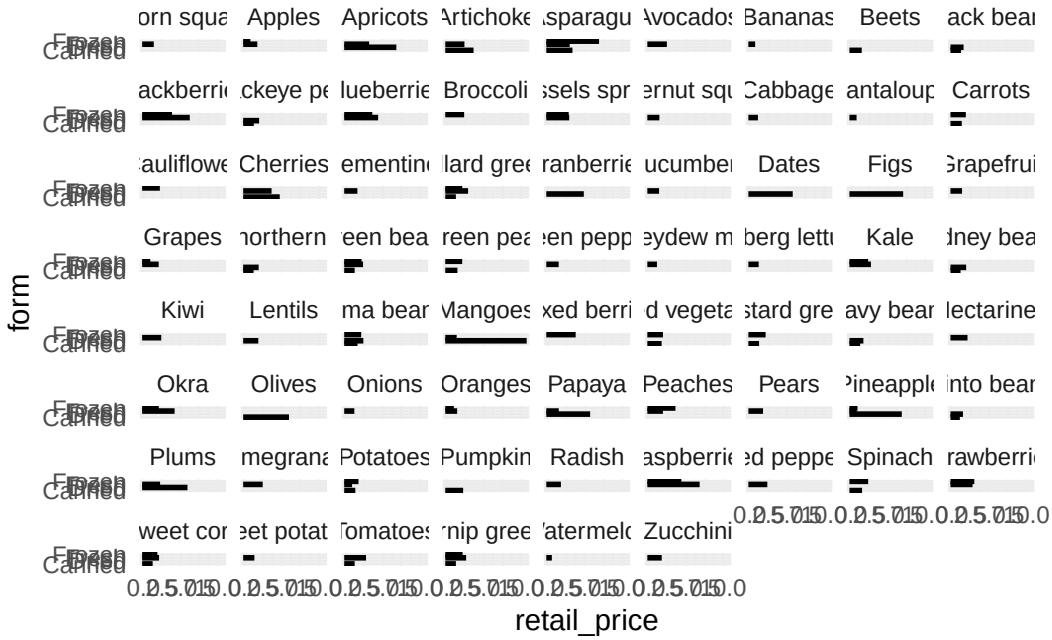
```r
# plot fruit and veg diff.
crazy_fruit_veg_plot <- fruit_veg_df %>%
  ggplot(aes(x = retail_price, y = form, fill = name)) +
  geom_bar(stat = 'summary', position = 'dodge', color = "black") +
  facet_wrap(~name)+
  theme_minimal()+
  theme(legend.position = "none")

crazy_fruit_veg_plot
```

```
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
```

```
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
```

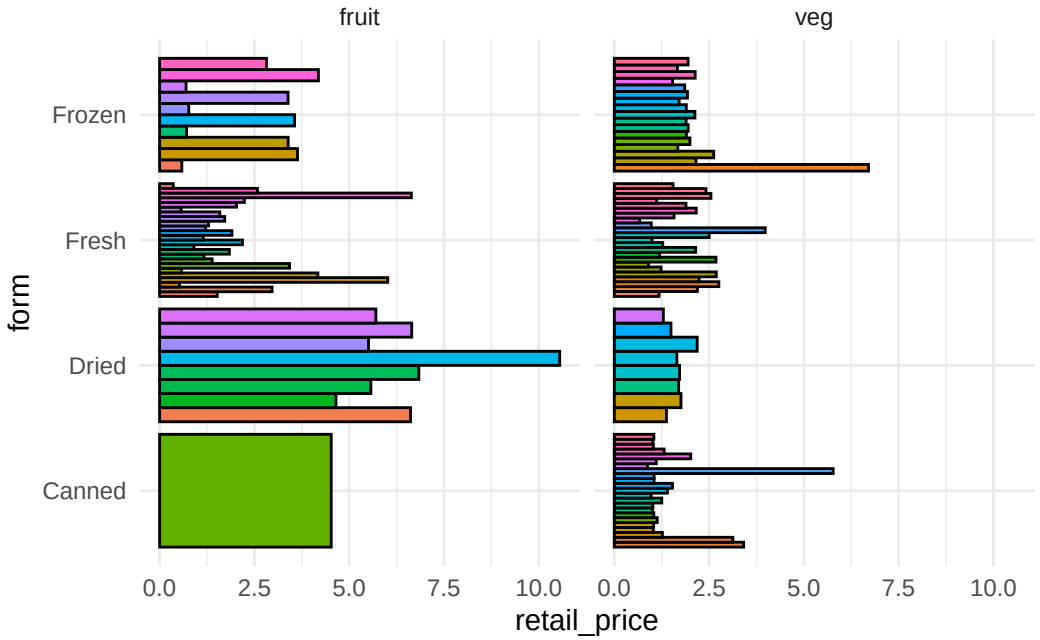
```
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
```




```
normal_fruit_veg_plot <- fruit_veg_df %>%
  ggplot(aes(x = retail_price, y = form, fill = name)) +
  geom_bar(stat = 'summary', position = 'dodge', color = "black") +
  facet_wrap(~fruit_veg)+
  theme_minimal()+
  theme(legend.position = "none")

normal_fruit_veg_plot
```

```
No summary function supplied, defaulting to `mean_se()`
No summary function supplied, defaulting to `mean_se()`
```




```
#Febe: Perhaps as an alternative, we could also consider plotting the mean price by form i

summary_form <- fruit_veg_df %>%
  group_by(fruit_veg, form) %>%
  summarize(
    mean_price = mean(retail_price, na.rm = TRUE),
    sd_price = sd(retail_price, na.rm = TRUE),
    n = n(),
    se_price = sd_price / sqrt(n),
    .groups = "drop"
  )

ggplot(summary_form, aes(x = form, y = mean_price, fill = fruit_veg)) +
  geom_col(position = "dodge", color = "black") +
  geom_errorbar(aes(ymin = mean_price - se_price,
                    ymax = mean_price + se_price),
                width = 0.2,
                position = position_dodge(0.9)) +
  scale_fill_manual(values = c("fruit" = "tomato3", "veg" = "seagreen3")) +
  labs(
    x = "Form",
    y = "Mean Retail Price",
```

```
  fill = "Type",
  title = "Mean Retail Price of Fruits vs Vegetables Across Forms"
) +
theme_bw() +
theme(text = element_text(size = 14))
```