Day 2, Part 2: Data Visualization

Brennan Terhune-Cotter and Matt Dye

Agenda

1. Aesthetic mappings in ggplot2
2. Building a simple ggplot
3. Comparing groups in clustered barplots
4. Plotting data with summary statistics

https://github.com/brennangitsit/2023_IAM3_R

Your Data!

You have a dataframe of the % change in CPI food prices over 2020-2022 for different foods, as well as a 2023 projection and a 20-year historical average.

item	annual_2020	annual_2021	annual_2022	predicted_2023	hist_avg_2003_2022
All food	3.4	3.9	9.9	7.5	2.8
Food away from home	3.4	4.5	7.7	8.3000000000000007	3.1
Food at home	3.5	3.5	11.4	7.8	2.5
Meats, poultry, and fish	6.3	6.8	9.6	2.1	3.4
Meats	7.4	7.7	8.1999999999999993	1.3	3.6
Beef and veal	9.6	9.3000000000000007	5.3	-1	4.5999999999999996
Pork	6.3	8.6	8.6999999999999993	-0.8	2.6
Other meats	4.4000000000000004	2.9	14.2	4.5	2.7
Poultry	5.6	5.0999999999999996	14.6	3.4	2.9
Fish and seafood	3.3	5.4	9.1	2.7	3.3

Your Data!

Here are all of the food categories in your data:

```r
# Turn a character vector into a string
paste(cpi$item, collapse = "; ")
```

```
[1] "All food; Food away from home; Food at home; Meats, poultry, and fish; Meats; Beef and veal;
Pork; Other meats; Poultry; Fish and seafood; Eggs; Dairy products; Fats and oils; Fruits and
vegetables; Fresh fruits and vegetables; Fresh fruits; Fresh vegetables; Processed fruits and
vegetables; Sugar and sweets; Cereals and bakery products; Nonalcoholic beverages; Other foods"
```

Aesthetic Mappings

- ggplot uses aesthetic mappings (aes) to connect data with visual features in the plot

```
?aes
```

- ggplot requires you to assign variable names to aes()

```
ggplot(aes(x = variable1, y = variable2, color = variable3)
```

Let's visualize the annual increase in prices of ALL FOOD from 2020-2023 as a line graph. What would go in aes()?

item	annual_2020	annual_2021	annual_2022	predicted_2023	hist_avg_2003_2022
All food	3.4	3.9	9.9	7.5	2.8

- The way the dataframe is set up currently, there is no way to assign variables to aes()
- We need to tidy our dataframe (tidying is covered later in the workshop)

Wide vs. Long Data

item	annual_2020	annual_2021	annual_2022	predicted_2023	hist_avg_20(
All food	3.4	3.9	9.9	7.5	2.8

item	year	increase	hist_avg_2003_2022
All food	2020	3.4	2.8
All food	2021	3.9	2.8
All food	2022	9.9	2.8
All food	2023	7.5	2.8

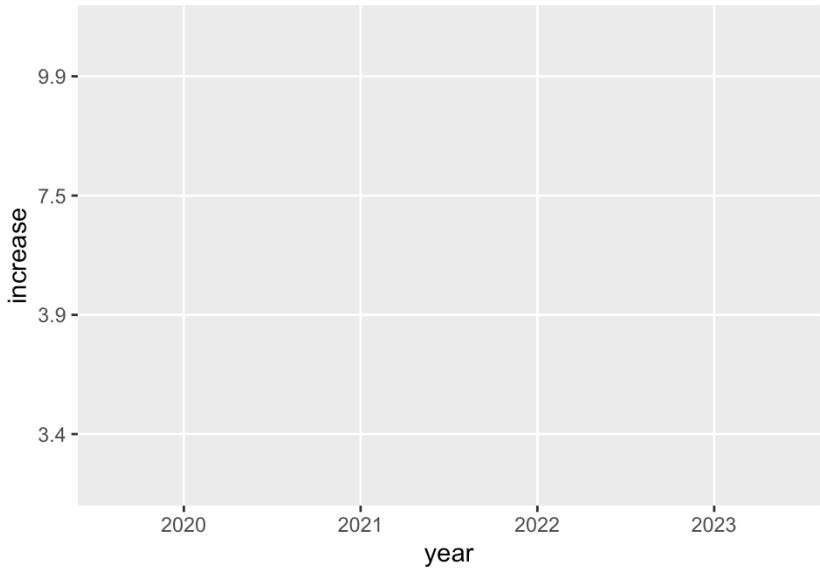
In long/tidy data:

1. Every column is a variable.
2. Every row is an observation.
3. Every cell is a single value.

```
ggplot(aes(x = year, y = increase))
```

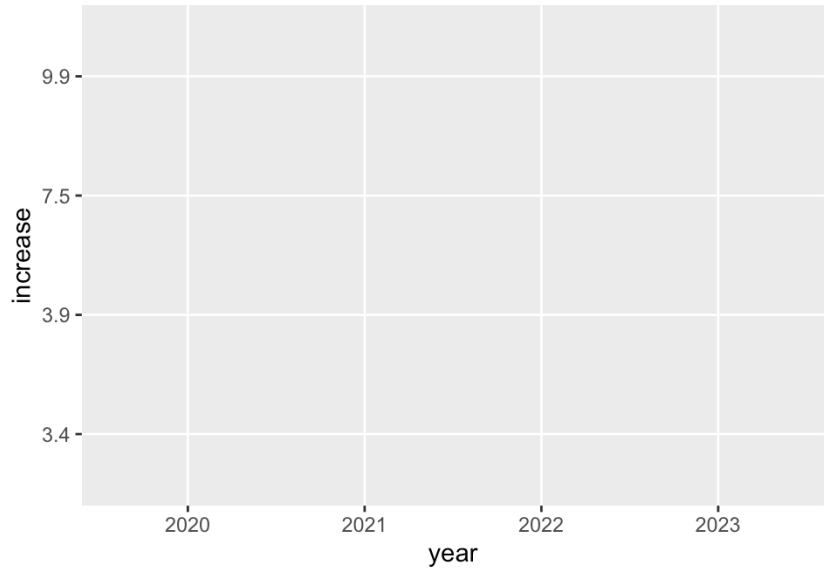
Building a ggplot

```r
cpi_allfood <- cpi_tidy %>%
  filter(item == "All food")

ggplot(cpi_allfood,
       mapping = aes(
         x = year,
         y = increase
       ))
```



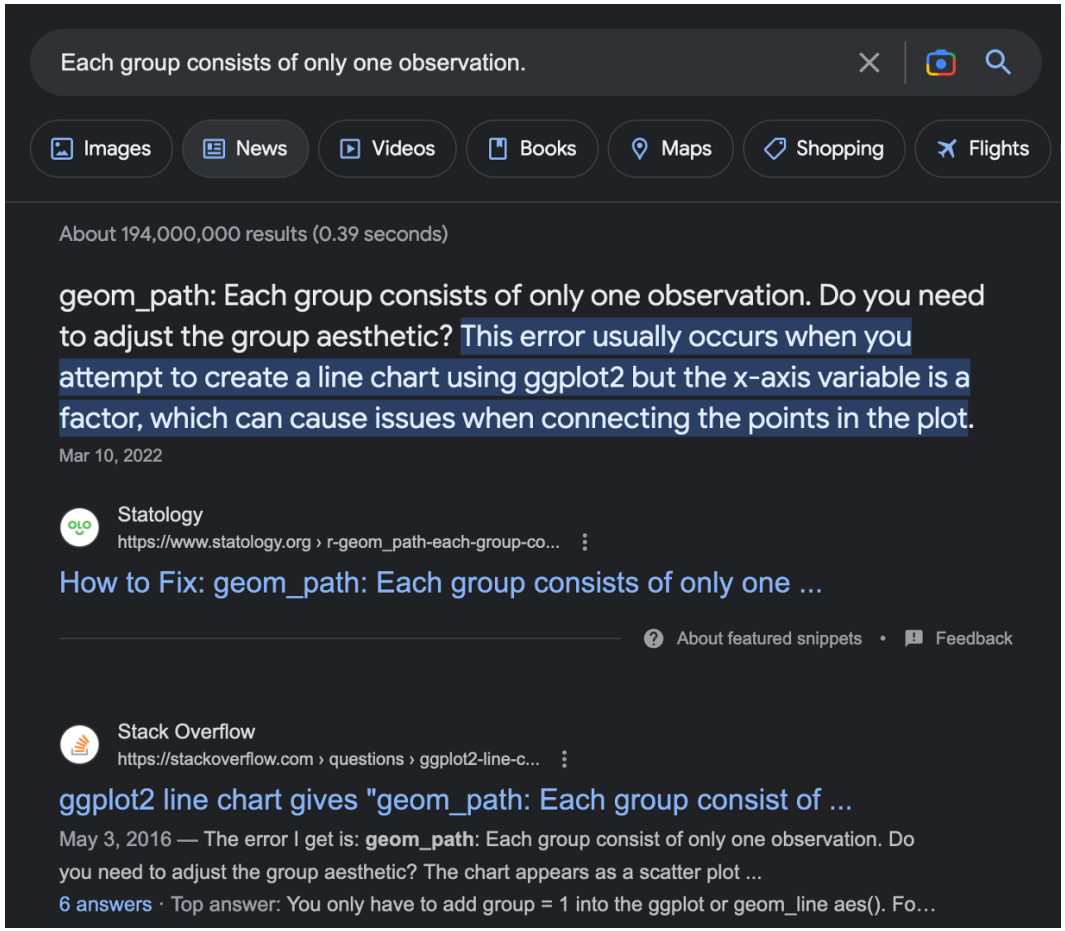
Building a ggplot

```r
cpi_allfood <- cpi_tidy %>%
  filter(item == "All food")

ggplot(cpi_allfood,
       mapping = aes(
         x = year,
         y = increase
       )) +
  geom_line()
```



`geom_line()`: Each group consists of only one observation. Do you need to adjust the group aesthetic?

A Slight Tangent: Troubleshooting

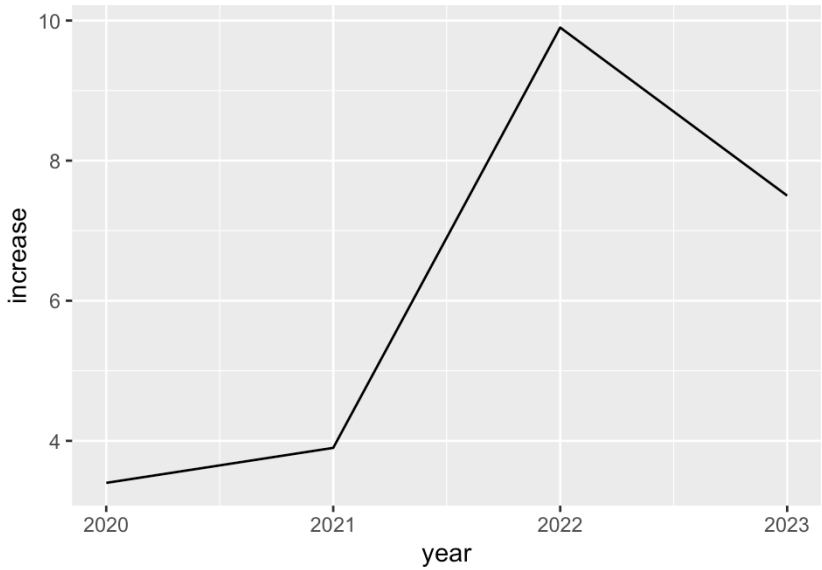


Two possible solutions here (and both work!)

Building a ggplot

```
cpi_tidy <- cpi_tidy %>%
  mutate(year = as.numeric(year),
         increase = as.numeric(increase))

cpi_allfood <- cpi_tidy %>%
  filter(item == "All food")

ggplot(cpi_allfood,
       mapping = aes(
         x = year,
         y = increase
       )) +
  geom_line()
```

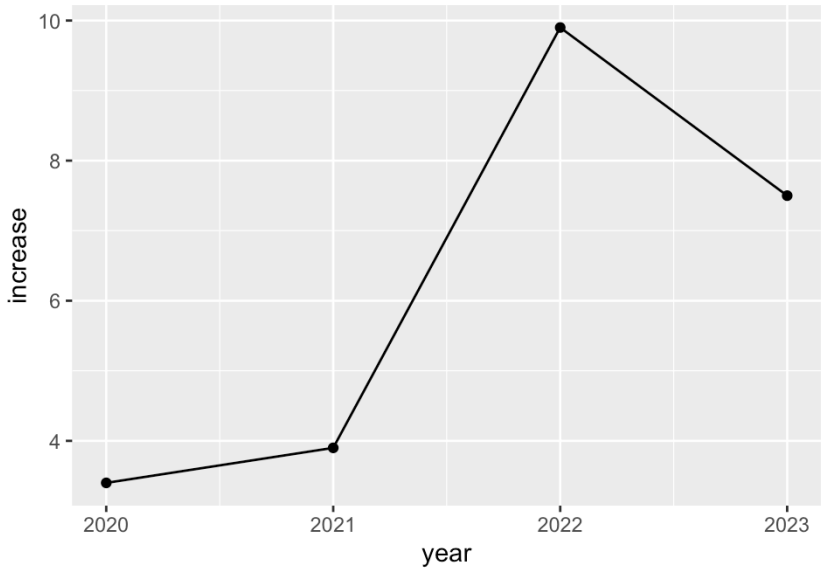
To fix the issue, I changed *year* and *increase* from char to numeric.



Let's add points to each datapoint.

Building a ggplot

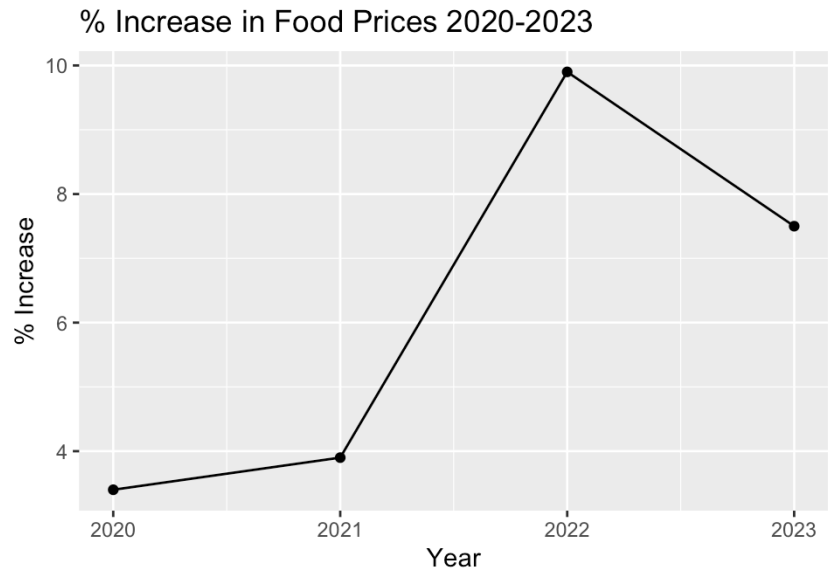
```
ggplot(cpi_allfood,
       mapping = aes(
         x = year,
         y = increase
       )) +
  geom_line() +
  geom_point()
```



Let's add appropriate axis labels and a title.

Building a ggplot

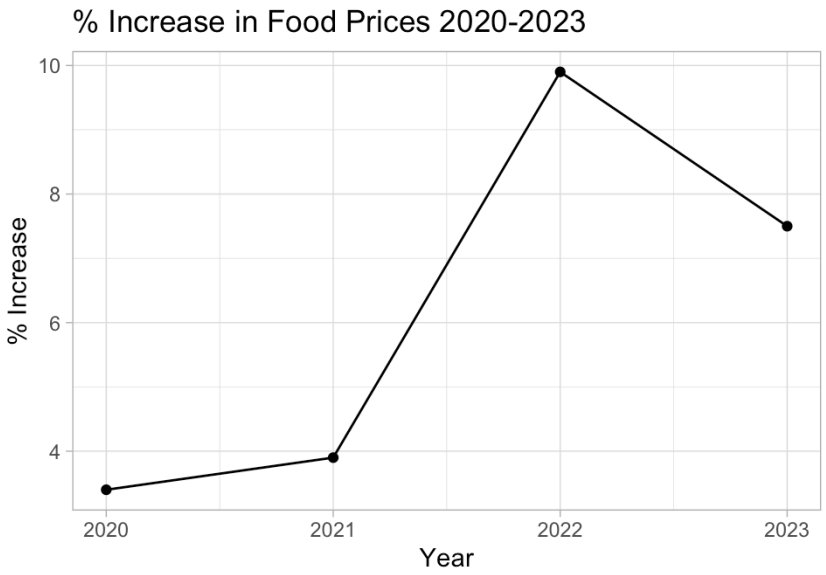
```
ggplot(cpi_allfood,
       mapping = aes(
         x = year,
         y = increase
       )) +
  geom_line() +
  geom_point() +
  labs(title = "% Increase in Food Prices 202
       x = "Year",
       y = "% Increase")
```



We can apply a theme to our plot to make it look better…

Building a ggplot

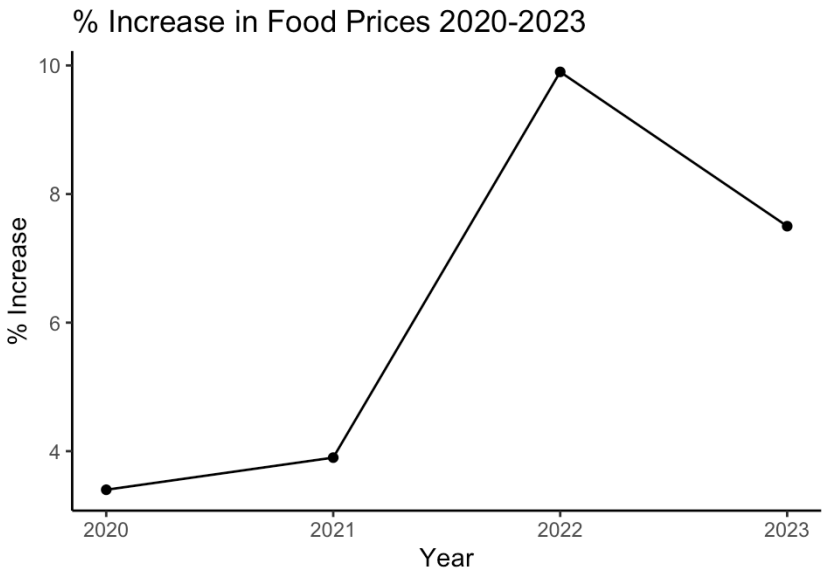
```
ggplot(cpi_allfood,
       mapping = aes(
         x = year,
         y = increase
       )) +
  geom_line() +
  geom_point() +
  labs(title = "% Increase in Food Prices 202
       x = "Year",
       y = "% Increase") +
  theme_light()
```



There are various themes which can be edited using **theme()**.

Building a ggplot

```
ggplot(cpi_allfood,
       mapping = aes(
         x = year,
         y = increase
       )) +
  geom_line() +
  geom_point() +
  labs(title = "% Increase in Food Prices 202
       x = "Year",
       y = "% Increase") +
  theme_classic()
```



There are various themes which can be edited using **theme()**.

What if we wanted to compare different types of foods?

Comparing groups in a ggplot

We make a dataframe with the types of foods we're interested in.

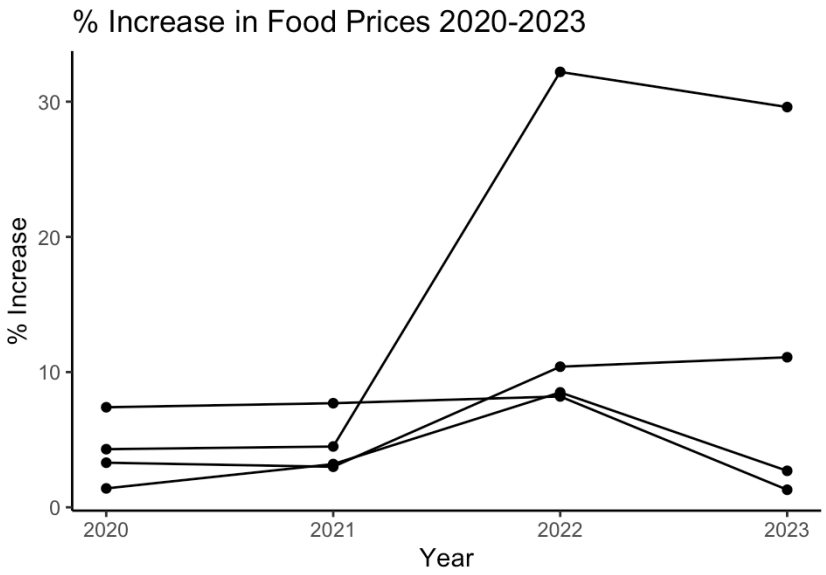
```
food_items <- c("Meats", "Eggs",
                "Fruits and vegetables",
                "Sugar and sweets")

cpi_foods <- cpi_tidy %>%
  filter(item %in% food_items)
```

item	year	increase	hist_avg_2003_2022
Meats	2020	7.4	3.6
Meats	2021	7.7	3.6
Meats	2022	8.2	3.6
Meats	2023	1.3	3.6
Eggs	2020	4.3	4.7
Eggs	2021	4.5	4.7
Eggs	2022	32.2	4.7
Eggs	2023	29.6	4.7
Fruits and vegetables	2020	1.4	2.2000000000000002
Fruits and vegetables	2021	3.2	2.2000000000000002
Fruits and vegetables	2022	8.5	2.2000000000000002
Fruits and vegetables	2023	2.7	2.2000000000000002
Sugar and sweets	2020	3.3	2.5
Sugar and sweets	2021	3.0	2.5
Sugar and sweets	2022	10.4	2.5
Sugar and sweets	2023	11.1	2.5

Comparing groups in a ggplot

```
ggplot(cpi_foods,
       mapping = aes(
         x = year,
         y = increase,
         group = item
       )) +
  geom_line() +
  geom_point() +
  labs(title = "% Increase in Food Prices 202
       x = "Year",
       y = "% Increase") +
  theme_classic()
```

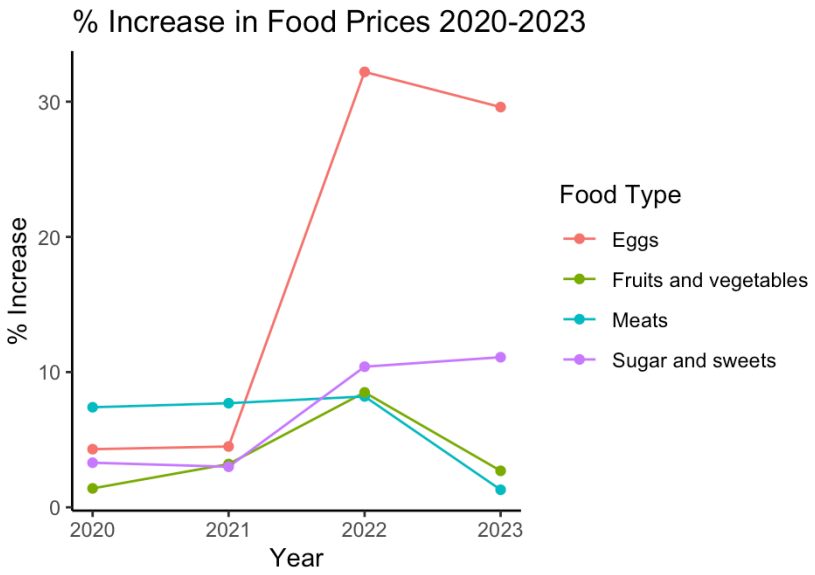
Then we add a new aesthetic mapping for *group*.



We need to add line colors and a legend to identify groups.

Comparing groups in a ggplot

```
ggplot(cpi_foods,
       mapping = aes(
         x = year,
         y = increase,
         color = item
       )) +
  geom_line() +
  geom_point() +
  labs(title = "% Increase in Food Prices 202
       x = "Year",
       y = "% Increase",
       color = "Food Type") +
  theme_classic()
```



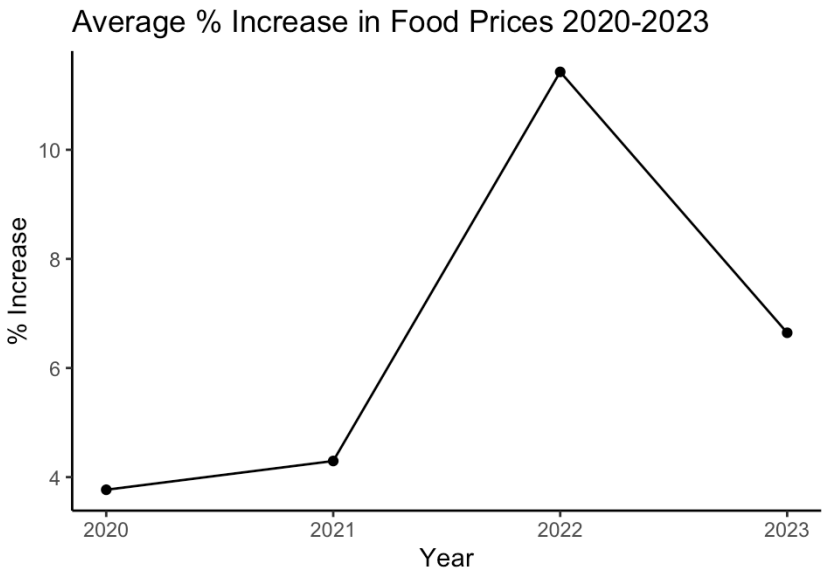
With ggplot, we can keep adding things as needed.

Plotting summary statistics

- We usually plot **summary statistics** like the *mean*.
- The simple ggplots above only plotted *identity*.
- You can make ggplots that summarize data…

Plotting summary statistics

```
ggplot(cpi_tidy,
       mapping = aes(
         x = year,
         y = increase
         )) +
  stat_summary(fun = mean,
               geom = "line") +
    stat_summary(fun = mean,
                 geom = "point") +
  labs(title = "Average % Increase in Food Pr
       x = "Year",
       y = "% Increase",
       color = "Food Type") +
  theme_classic()
```



- However, most of the time it's better to create a **summary dataset** to use for plotting.
- This allows us to keep an eye on the data.
- Summarizing data and plotting it will be covered on the last day :)