Homework 3 Solution

Disclaimer

This homework solution is for the sole benefit of students taking Stat 220 from Prof. Bastola during Spring term 2024. Dissemination of this solution to people who are not registered for this course is not permitted and will be considered grounds for Academic Dishonesty for the all individuals involved in the giving and receiving of the solution.

Problem 1: babynames

Use the `babynames` dataset and `dplyr` to answer the following:

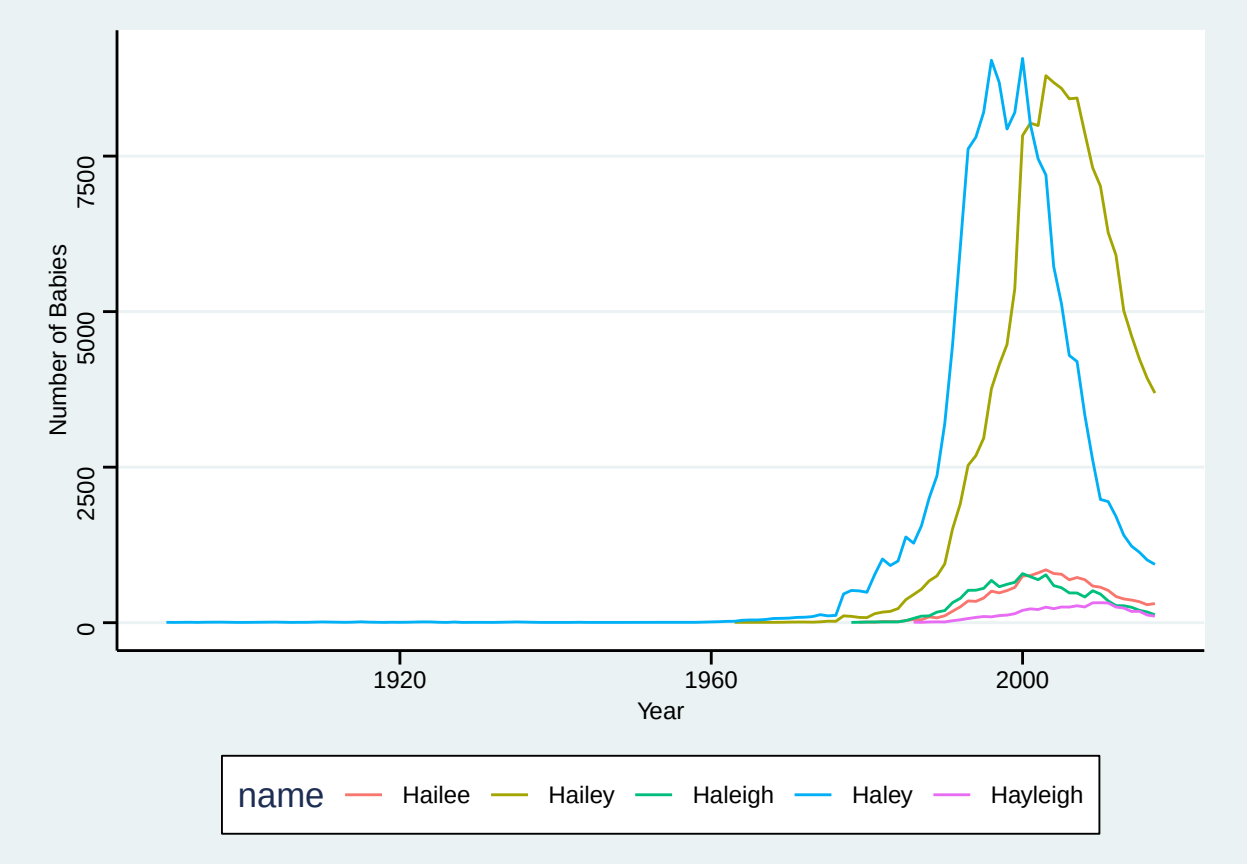
a.

Create a data set `babynames_hailey` that contains data on female babies named Hailey and some other variations including Hailee, Haleigh, Haley, or Hayleigh. Create one graph that shows the how the number of babies with each of these five names varies over time. Summarize what this graph shows in 1-2 sentences.

answer:

The names `Hailey` and `Haley` are the more prominent variations among the five. These names rose to popularity sometime after 1960s, peaked around 200s, and have been waning since.

```
> babynames_hailey <- babynames %>%
+   filter(name == "Hailey" | name == "Hailee" | name == "Haleigh" | name == "Haley" | name == "Hayleigh
> ggplot(data=babynames_hailey, aes(x=year, y=n)) +
+   geom_line(aes(colour=name)) +
+   xlab('Year') +
+   ylab('Number of Babies')
```



b.

Historically, which name is most balanced between the number of males and females with that name aggregated over all years? To answer this, aggregate `babyname` name counts for each name over all years by sex, then filter to names that have at least 10,000 occurrences by sex. For each name, compute the proportion of male and female names and find which name (or names) have male/female proportions closest to 0.50.

Note: Filtering to "common" names avoids finding rare names that, for example, have 1 instance of a male and 1 of a female (which would be 50% male and 50% female).

answer:

First aggregate names and filter to "common" names:

```
> ag_names <- babynames %>%
+   group_by(name, sex) %>%
+   summarize(tot_n_bysex = sum(n)) %>% # for each sex/year get number of babies
+   filter(tot_n_bysex > 10000)
> ag_names
# A tibble: 2,569 x 3
# Groups:   name [2,456]
   name     sex   tot_n_bysex
   <chr>    <chr>       <int>
 1 Aaliyah  F           83580
 2 Aaron    M          575297
 3 Abbey    F           17139
 4 Abbie    F           21391
 5 Abbigail F           11465
```

```
 6 Abby     F           57575
 7 Abel     M           50123
 8 Abigail  F          356404
 9 Abraham  M           88644
10 Abram    M           16391
# i 2,559 more rows
```

By name, computer male/female proportions and look at absolute difference from 0.5:

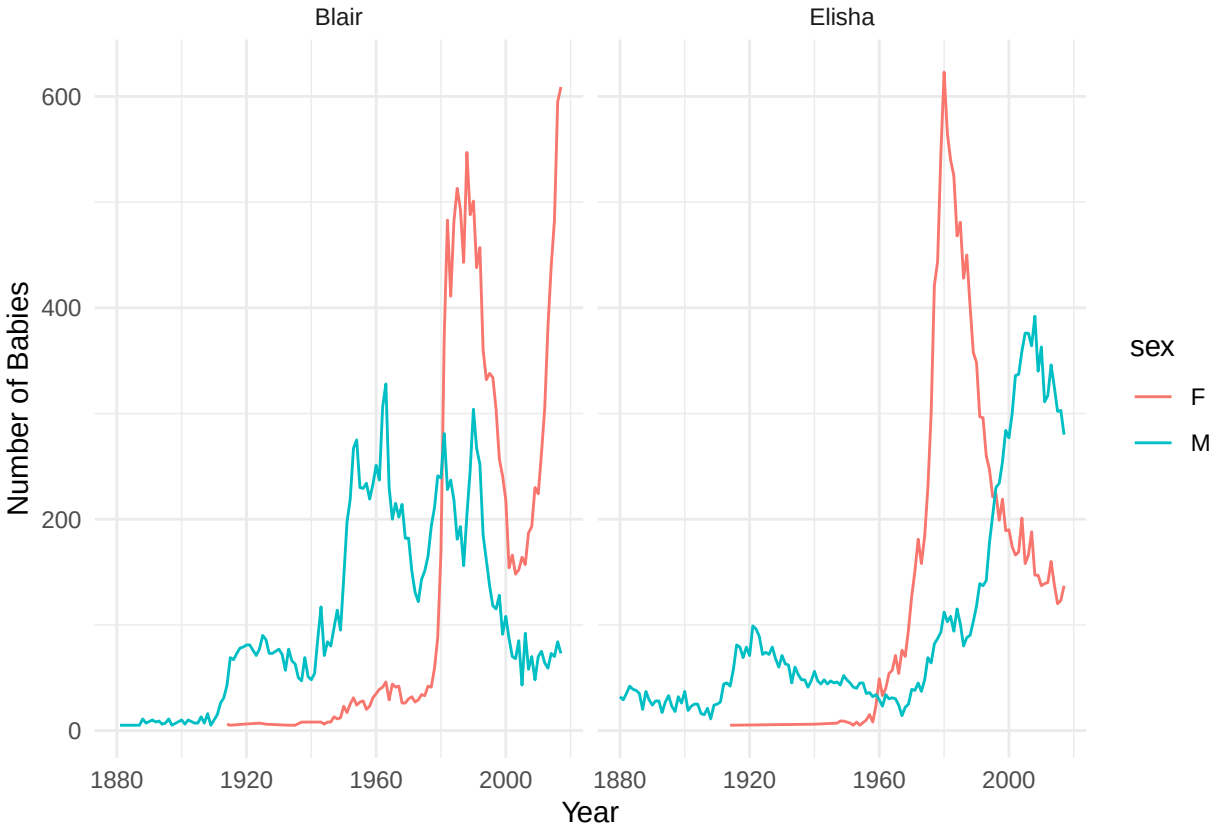
```
> ag_names %>%
+   group_by(name) %>%   # focus = name
+   mutate(name_prop = tot_n_bysex/sum(tot_n_bysex),    # num. babies of a sex/num both sex for each na
+          abs_diff = abs(.5 - name_prop)) %>% # how does this proportion differ from 0.50
+   ungroup() %>%  # need to ungroup to sort entire data
+   arrange(abs_diff)  # sort from small to large
# A tibble: 2,569 x 5
   name   sex   tot_n_bysex name_prop abs_diff
   <chr>  <chr>       <int>     <dbl>    <dbl>
 1 Blair  M           14470     0.505  0.00480
 2 Blair  F           14195     0.495  0.00480
 3 Elisha F           13599     0.505  0.00499
 4 Elisha M           13330     0.495  0.00499
 5 Kerry  F           48534     0.495  0.00541
 6 Kerry  M           49596     0.505  0.00541
 7 Kris   F           13490     0.491  0.00895
 8 Kris   M           13982     0.509  0.00895
 9 Robbie F           22264     0.516  0.0162
10 Robbie M           20863     0.484  0.0162
# i 2,559 more rows
```

Here we see that Blair is a "common" name with an almost 50/50 split between male and female birth records. Elisha is a close second.

c. For the names identified in part (b) as most balanced between males and females, find the time period during which these names were the most popular. Plot the number of babies with these names by year and sex.

answer: The graph shows that the names "Blair" and "Elisha" were most popular during the late 20th century, with "Blair" peaking in the 1990s and "Elisha" in the 1980s.

```
> balanced_names <- c("Blair", "Elisha")
> popular_period <- babynames %>%
+   filter(name %in% balanced_names) %>%
+   group_by(name, sex, year) %>%
+   summarize(n = sum(n))
>
> ggplot(data = popular_period, aes(x = year, y = n, color = sex)) +
+   geom_line() +
+   facet_wrap(~name) +
+   xlab('Year') +
+   ylab('Number of Babies') +
+   theme_minimal()
```



d. Find the names that have consistently increased in popularity from 1950 to 2012. Create a graph that shows the trends of these names over time.

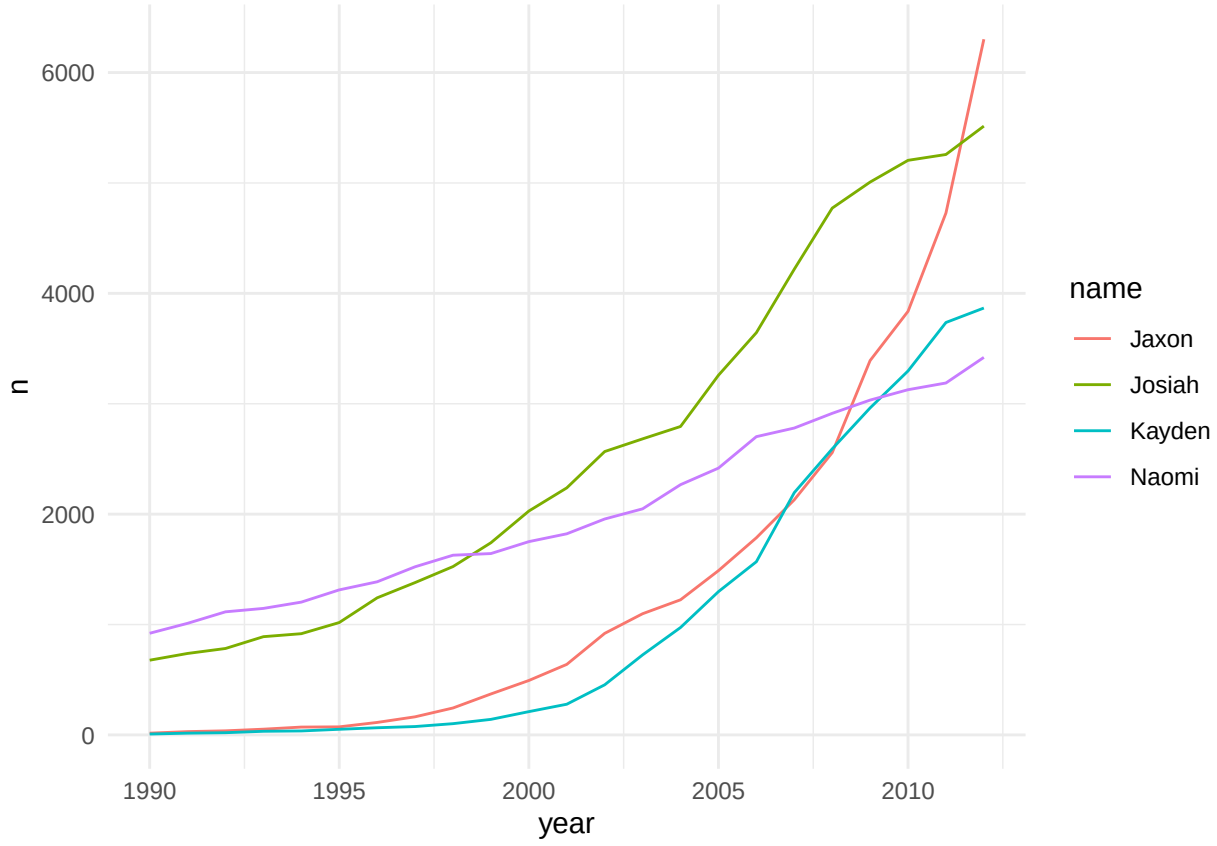
Hint: you may need to use `lag()` function that allows you to access and manipulate previous observations in a sequence. `lag()` function shifts the data by one position, padding the beginning with an `NA`. So, when manipulating data using `lag()`, it's advisable to use `tidyr::drop_na()` post-lag operation to get rid of the `NA`'s.

answer: The graph displays the names `Kaiden`, `Kayden`, and `Zayden` have consistently increased in popularity from 1950 to 2012. These names show a general upward trend over time, with some fluctuations. Note that the popularity of some names may have plateaued or declined after 2012.

```
> consistent_growth <- babynames %>%
+   filter(year >= 1990, year <= 2012) %>%
+   group_by(name, sex, year) %>%
+   summarize(n = sum(n)) %>%
+   arrange(year) %>%
+   group_by(name, sex) %>%
+   mutate(growth = n - lag(n)) %>%
+   drop_na(growth) %>%
+   group_by(name, sex) %>%
+   summarize(total_growth = sum(growth),
+             min_growth = min(growth)) %>%
+   filter(min_growth > 0, total_growth > 0)
```

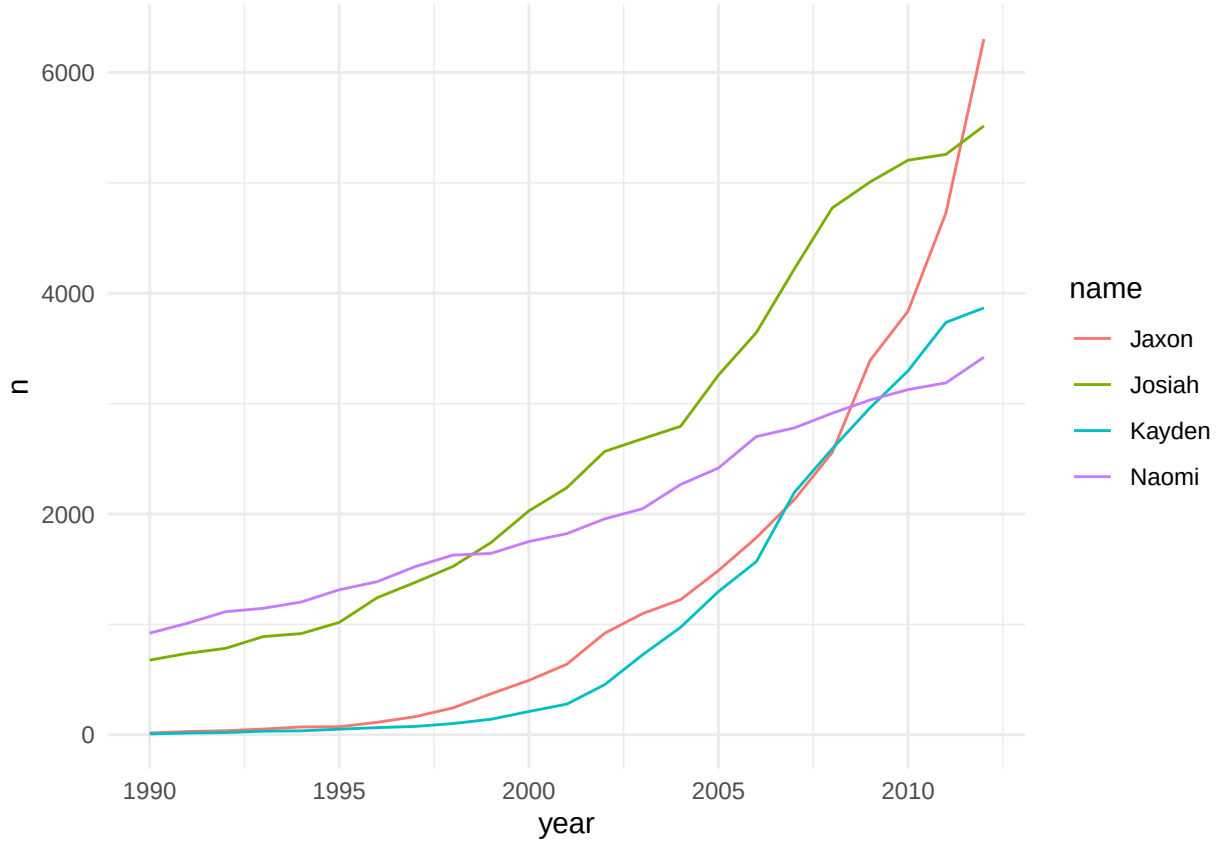
```
> consistent_growth %>%
+   left_join(babynames, by = c("name", "sex"), multiple = "all") %>%
```

```
+   filter(year %in% c(1990:2012)) %>%
+   group_by(name, sex) %>%
+   summarize(count = n()) %>%
+   filter(count == 23) %>%
+   left_join(babynames, by = c("name", "sex"), multiple = "all") %>%
+   filter(year %in% c(1990:2012)) %>%
+   ggplot(aes(x = year, y = n, color = name)) +
+   geom_line() +
+   theme_minimal()
```



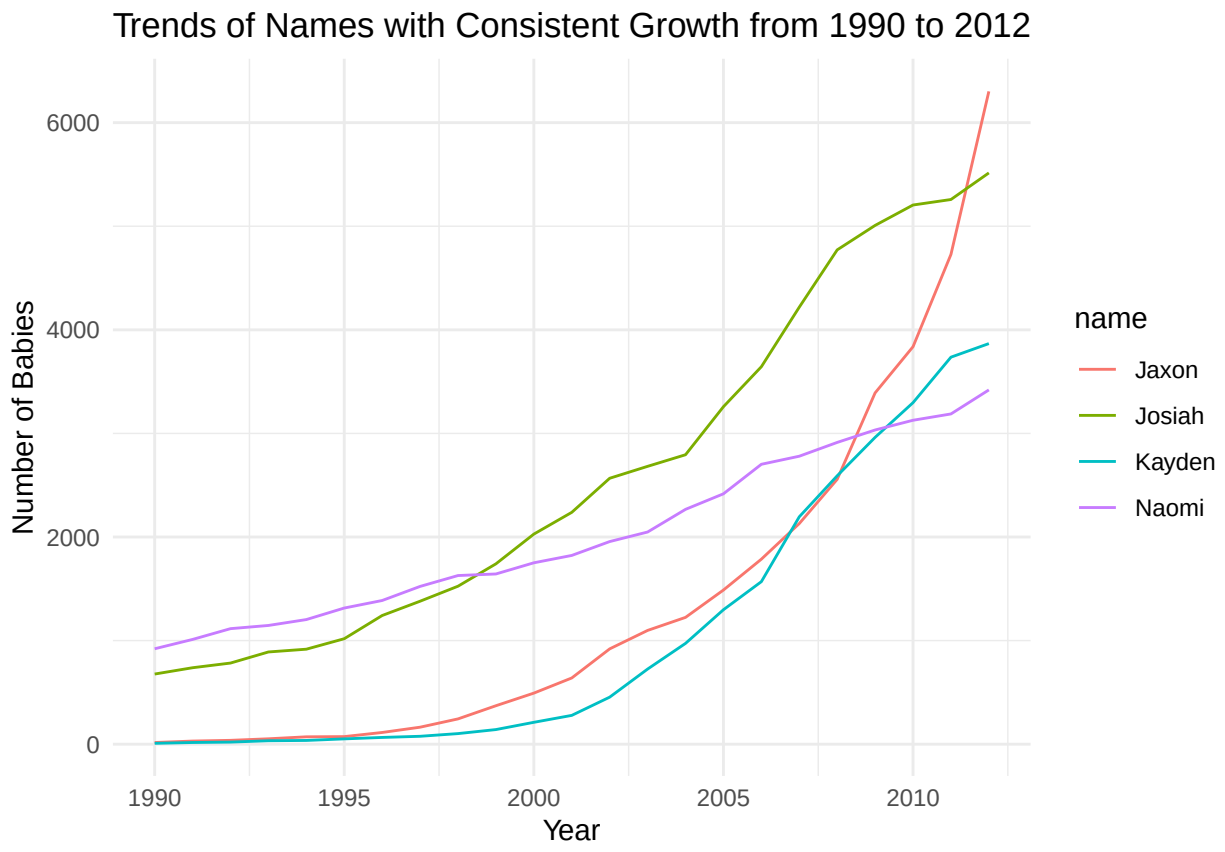

```
> library(tidyverse)
> library(babynames)
>
> library(dplyr)
>
> consistent_growth <- babynames %>%
+   filter(year >= 1990, year <= 2012) %>%
+   group_by(name, sex, year) %>%
+   summarize(n = sum(n)) %>%
+   arrange(year) %>%
+   group_by(name, sex) %>%
+   mutate(growth = n - lag(n)) %>%
+   drop_na(growth) %>%
+   group_by(name, sex) %>%
+   summarize(total_growth = sum(growth),
+             min_growth = min(growth)) %>%
```

```
+   filter(min_growth > 0, total_growth > 0)
>
>
> consistent_growth %>%
+   inner_join(babynames %>% filter(year %in% 1990:2012), by = c("name", "sex")) %>%
+   group_by(name, sex) %>%
+   filter(n() == 23)  %>%
+   ggplot(aes(x = year, y = n, color = name)) +
+   geom_line() +
+   theme_minimal()
```




```
> library(dplyr)
> library(ggplot2)
>
> babynames %>%
+   filter(year >= 1990, year <= 2012) %>%
+   group_by(name, sex) %>%
+   arrange(name, sex, year) %>%
+   mutate(growth = n - lag(n)) %>%
+   drop_na() %>%
+   summarize(total_growth = sum(growth), min_growth = min(growth)) %>%
+   filter(min_growth > 0, total_growth > 0) %>%
+   inner_join(babynames %>% filter(year %in% 1990:2012), by = c("name", "sex")) %>%
+   group_by(name, sex) %>%
+   filter(n() == 23) %>%
+   ggplot(aes(x = year, y = n, color = name)) +
```

```
+     geom_line() +
+     theme_minimal() +
+     labs(title = "Trends of Names with Consistent Growth from 1990 to 2012",
+          x = "Year",
+          y = "Number of Babies")
```



Problem 2: Consumption

The data set `food_consumption.csv` was compiled from the website nu3 and contains the following measurements from 2018:

variable	class	description
country	character	Country Name
food_category	character	Food Category
consumption	double	Consumption (kg/person/year)
co2_emmission	double	Co2 Emission (Kg CO2/person/year)

```
> food_consumption <- read.csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/foodconsump
```

a.

Create a data frame that shows the 3 food category types with the highest total food consumption in 2018.

answer:

```
> food_consumption %>%
+   group_by(food_category) %>%
+   summarize(total_consumption = sum(consumption)) %>%
+   slice_max(total_consumption, n = 3)
# A tibble: 3 x 2
  food_category            total_consumption
  <chr>                                <dbl>
1 Milk - inc. cheese                  16351.
2 Wheat and Wheat Products             9301.
3 Rice                                 3819.
```

b.

For each country, compute the percentage of consumption for each food category out of that country's total food consumption (per person). Then create a data frame that shows the 5 countries with the highest proportion of total consumption in the category `Milk - inc. cheese`.

answer:

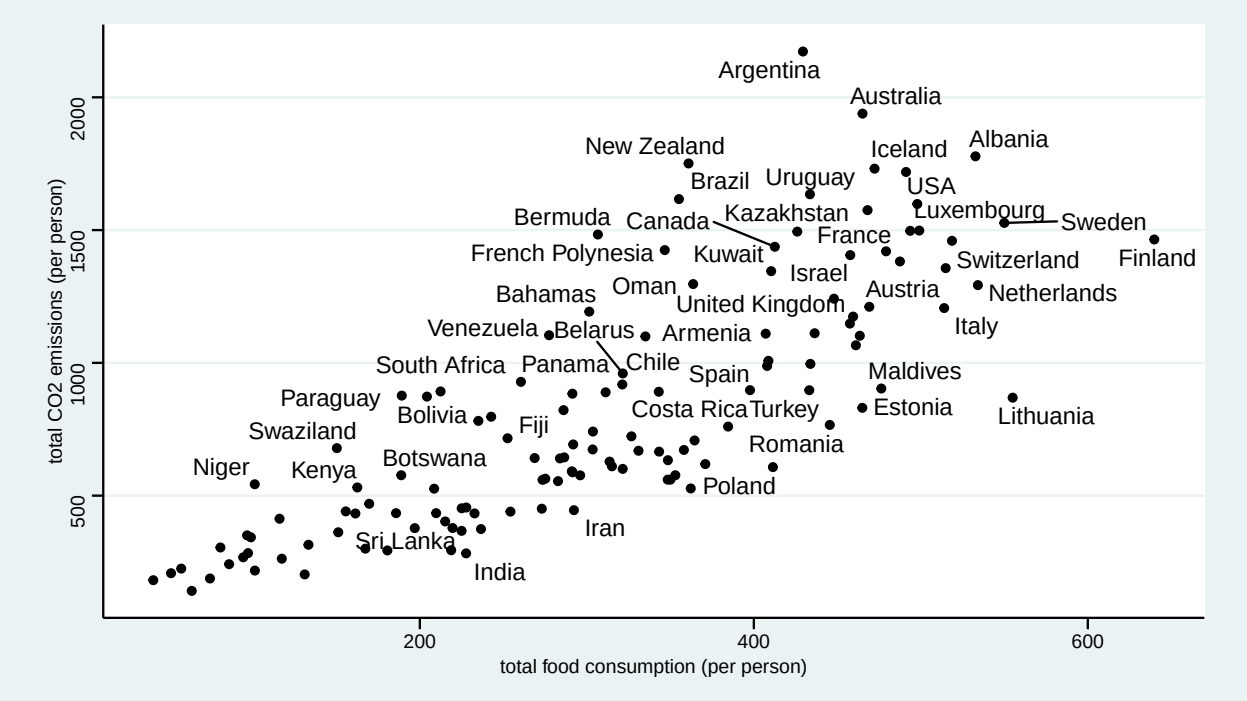
```
> food_consumption %>%
+   group_by(country) %>%
+   mutate(total_consumption = consumption/sum(consumption)) %>%
+   ungroup() %>%
+   filter(food_category == "Milk - inc. cheese") %>%
+   slice_max(total_consumption, n = 5)
# A tibble: 5 x 6
      X country     food_category    consumption co2_emmission total_consumption
  <int> <chr>       <chr>                  <dbl>         <dbl>             <dbl>
1   150 Finland     Milk - inc. che~         431.          614.             0.673
2   260 Netherlands Milk - inc. che~         341.          486.             0.639
3   788 Botswana    Milk - inc. che~         118.          168.             0.623
4   117 Sweden      Milk - inc. che~         341.          486.             0.620
5   249 Switzerland Milk - inc. che~         319.          454.             0.619
```

c.

For each country, compute total food consumption and total CO2 (per person) across all categories, then create a scatterplot of total CO2 vs. total consumption. Label points by country name using the `geom_text_repel` from the `ggrepel` package (which you may need to install). Describe the trend you see.

answer: There is a positive relationship that is fairly linear, as food consumption increases so do CO2 emissions.

```
> food_consumption %>%
+   group_by(country) %>%
+   summarize(total_consumption = sum(consumption), total_co2 = sum(co2_emmission)) %>%
+   ggplot(aes(x = total_consumption, y = total_co2)) +
+   geom_point() +
+   ggrepel::geom_text_repel(aes(label = country)) +
+   labs(x = "total food consumption (per person)",
+        y = "total CO2 emissions (per person)")
Warning: ggrepel: 81 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```



Problem 3: restaurant violations

The data set `Violations` contains info about the outcome of health inspections of restaurants in New York City. See `?mdsr::Violations` for more details.

```
> #install.packages("mdsr")
> data("Violations", package = "mdsr")
```

a.

Use `dplyr` to construct a data frame with the following:

- only cases from Manhattan (`boro`)
- includes the median violation score (`score`) by zip code (`zipcode`)
- includes the number of inspections by zip code
- only includes zip codes with with 50 or more inspections

Note: The first line of your pipe should be `tidyr::drop_na(score)` to remove rows with missing values for `score`. `drop_na` is from the `tidyr` package.

answer:

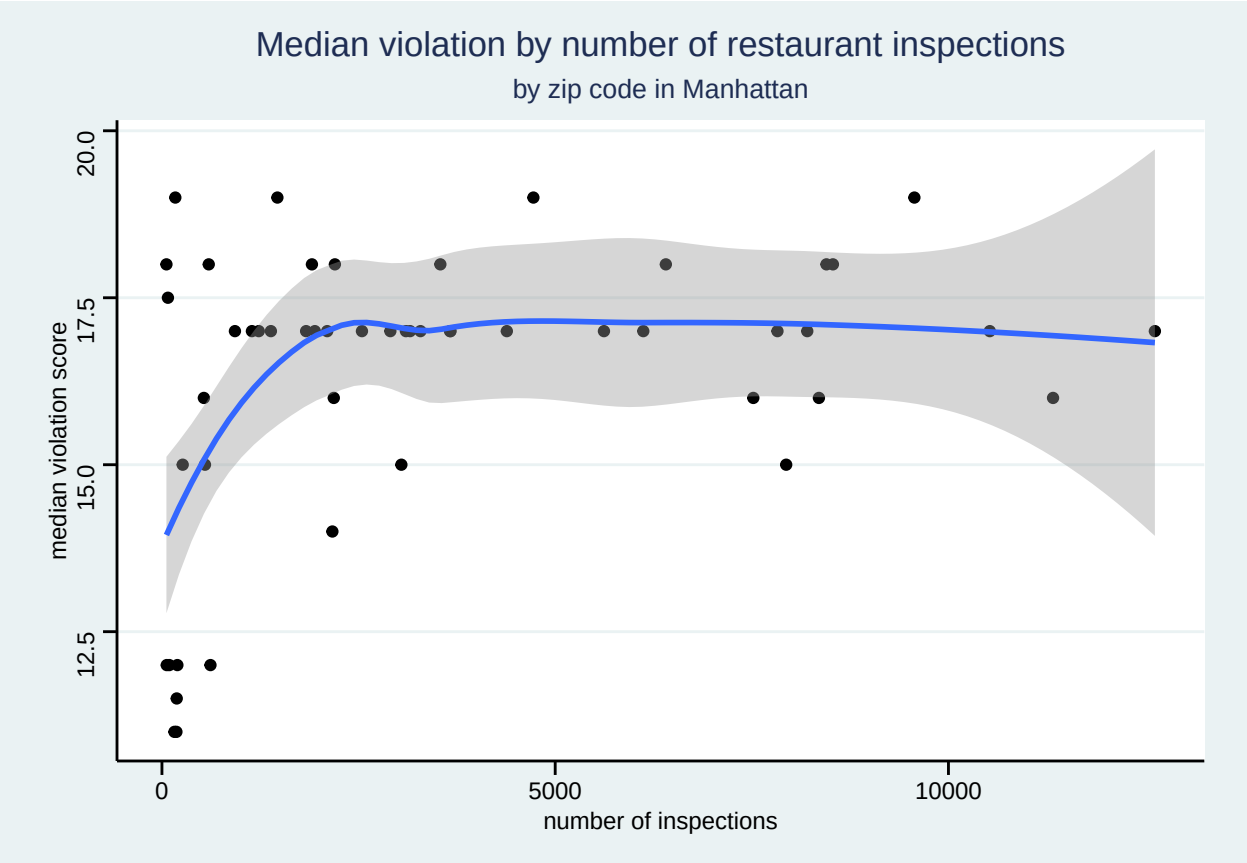
```
>
> viol_med <- Violations %>%
+   drop_na(score) %>%  # remove rows with NA for score
+   filter(boro == "MANHATTAN") %>%  # omit rows with NA score
+   group_by(zipcode) %>%
+   summarize(
+     med_score = median(score), # median by zip
+     N=n()) %>%                 # number of inspections by zip
+   filter(N >= 50)              # 50 or more inspections
```

b.

Create and interpret a `ggplot2` that shows the relationship between the number of inspections (x) and the median score (y). Use both the `point` and `smooth` geometries in your graph and add an information title and axes labels to your plot.

answer: There isn't a strong trend between the number of inspections in a zip code and the median number of violations. The only potentially interesting feature is the cluster of zip codes with lower median violations and fewer inspections.

```
> viol_med %>%
+   ggplot(aes(x = N, y = med_score)) +
+   geom_point() +
+   geom_smooth() +
+   labs(
+     x = "number of inspections",
+     y = "median violation score",
+     title = "Median violation by number of restaurant inspections",
+     subtitle = "by zip code in Manhattan")
```

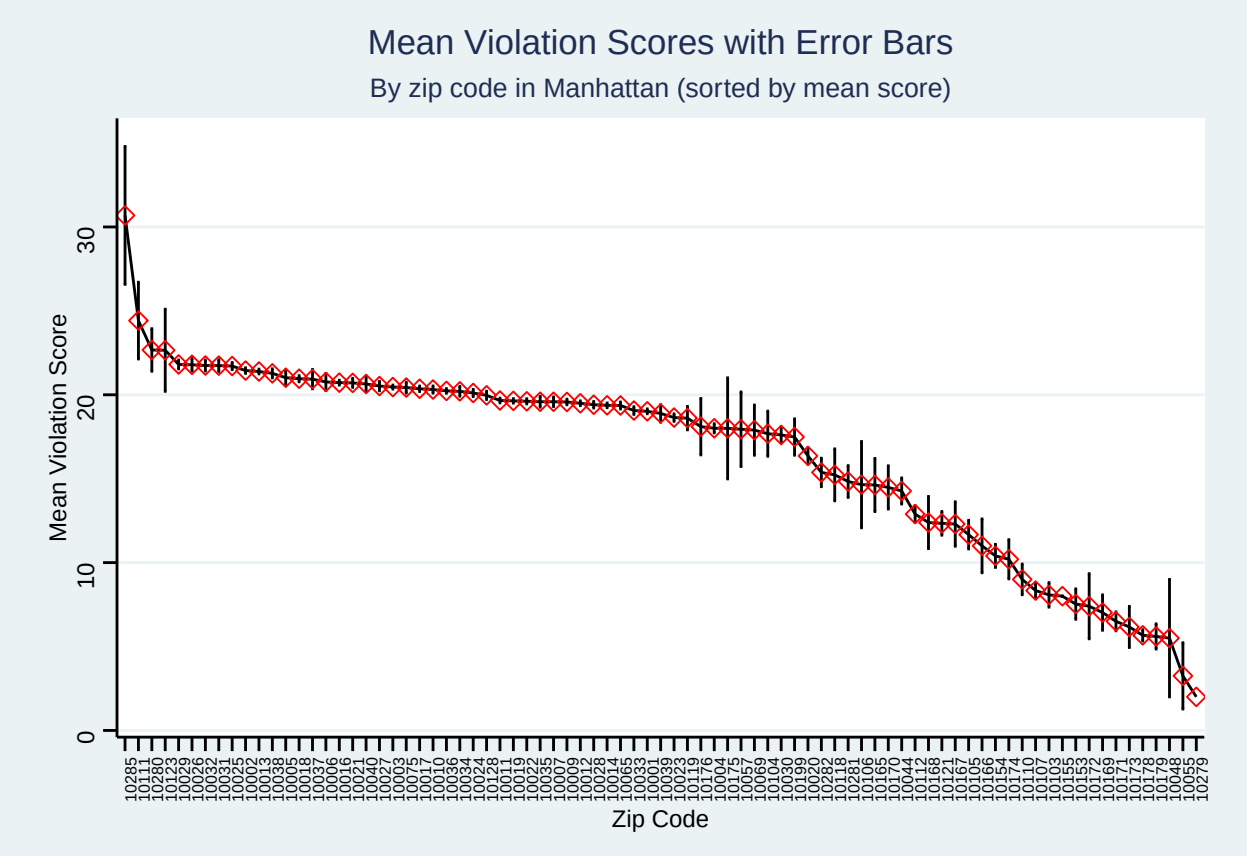


c. Now, let's calculate the mean and standard error for the violation scores across all the zip codes of Manhattan. Then, create and interpret a `ggplot2` plot that shows the relationship between the zip code (x) and the ordered mean violation scores (y), along with error bars representing one standard error. Also based on this plot, identify the neighborhood with the highest and lowest mean violation scores.

answer: The neighborhood with the highest and lowest mean violation scores are 10285 and 10279, respectively.

```
> viol_mean_se <- Violations %>%
+   drop_na(score) %>%
+   filter(boro == "MANHATTAN") %>%
+   group_by(zipcode) %>%
+   summarize(
+     mean_score = mean(score), # Calculate mean of scores
+     count = n(),
+     se = sd(score) / sqrt(n()) # Calculate standard error
+   ) %>%
+    arrange(desc(mean_score))
```

```
> viol_mean_se %>%
+   ggplot(aes(x = reorder(zipcode, -mean_score), y = mean_score)) +
+   geom_line(group = 1) +
+   geom_errorbar(aes(ymin = mean_score - se, ymax = mean_score + se), width = 0.2) +
+   geom_point(color = "red", size = 2, pch = 5) + # Add tick marks for the mean scores
+   labs(
+     x = "Zip Code",
+     y = "Mean Violation Score",
+     title = "Mean Violation Scores with Error Bars",
+     subtitle = "By zip code in Manhattan (sorted by mean score)"
+   ) +
+   theme(axis.text.x = element_text(angle = 90, hjust = 1, size = 6))
```



Problem 4: joins

The data set below called `Students` contains information on five students with their ID number, first name and computer preference.

Id	Name	Computer
1	Arya	m
2	Gregor	m
3	Cersei	w
4	Jon	m
5	Jon	w

The data set below called `Classes` contains the roster information (student first name and ID) for two classes.

Class	Student	Stud_Id
CS	Jon	4
CS	Arya	1
CS	Cersei	3
Stats	Gregor	2
Stats	Jon	4
Stats	Jon	5
Stats	Arya	1

What data set will be produced by the following commands? Describe the data set in words and show what it looks like using an R Markdown table to display the new data set.

a.

```
> left_join(Classes, Students, by = c("Stud_Id" = "Id"))
```

answer

The student IDs are perfect keys so for each `Stud_Id` there is only one `Id` in the `Students` dataset. The data set produced has the same number of rows as `Classes` with the extra columns giving computer info (and the name from `Students`)

Class	Student	Stud_Id	Name	Computer
CS	Jon	4	Jon	m
CS	Arya	1	Arya	m
CS	Cersei	3	Cersei	w
Stats	Gregor	2	Gregor	m
Stats	Jon	4	Jon	m
Stats	Jon	5	Jon	w
Stats	Arya	1	Arya	m

b.

```
> CS <- Classes %>% filter(Class == "CS")
> Stats <- Classes %>% filter(Class == "Stats")
> semi_join(Stats, CS, by = "Stud_Id")
```

answer

A `semi_join` of the `Stats` class roster with `CS` will show `Stats` students who are also taking CS.

Class	Student	Stud_Id
Stats	Jon	4
Stats	Arya	1

c.

```
> anti_join(Stats, CS, by = "Stud_Id")
```

answer

An `anti_join` of the `Stats` class roster with `CS` will show `Stats` students who are not also taking CS.

Class	Student	Stud_Id
Stats	Gregor	2
Stats	Jon	5

Problem 5: restructure

Consider the `Lakes_wide` data set below that records lake clarity (in meters) for 2012 through 2014.

LakeId	2012	2013	2014
1	6.5	5.8	5.8
2	2.1	3.4	2.8

What data set will be produced by the following commands? Describe the data set in words and show what it looks like using an R Markdown table to display the new data set.

a.

```
> Lakes_wide %>%
+   pivot_longer(
+     cols = 2:4,
+     names_to = "Year",
+     values_to = "Clarity"
+   )
```

answer

This command gathers columns 2-4 (clarity measurements) and places the `values_to` in a variable called `Clarity` and creates a new `names_to` variable called `Year` that identifies the year of each value. The `LakeId` identifies which lake each measurement was taken from. (Note that the order of rows in your data frame can be different from the table below.)

LakeId	Year	Clarity
1	2012	6.5

LakeId	Year	Clarity
1	2013	5.8
1	2014	5.8
2	2012	2.1
2	2013	3.4
2	2014	2.8

b.

```
> Lakes_wide %>%
+   pivot_longer(
+     cols = 2:4,
+     names_to = "Year",
+     values_to = "Clarity"
+   ) %>%
+   group_by(LakeId) %>%
+   arrange(Year) %>%
+   mutate(Change_in_Clarity = Clarity - lag(Clarity))
```

answer

This takes the long data set from (a), arranges Clarity measures by year (first to last year), then for each lake it computes the difference between yearly measurements (current year minus previous year) using the `lag` function. Note that the 2012 change measurements are missing (`NA`) because we do not know the 2011 measurements. Note that these rows must be arranged by year (low to high).

LakeId	Year	Clarity	Change_in_Clarity
1	2012	6.5	NA
2	2012	2.1	NA
1	2013	5.8	-0.7
2	2013	3.4	1.3
1	2014	5.8	0.0
2	2014	2.8	-0.6