Data 607 - Project 2

Richie R.

Overview

In this project, we will explore three of the datasets and analysis question presented within Discussion 5.

The three analysis that will be performed in this will be:

1. G. Schneider's "Consumer Price Index 2024"
2. Z. Liang's "U.S Vehicle Model Sales"
3. R. Rivera's "Car Crash information over time (1994-Present)"

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)

# For HTML web scraping
library(rvest)
```

```
##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
# R has a useful package to read excel files
library(readxl)
```

1. G. Schneider's "Consumer Price Index 2024"

```
data_url_1 <- "https://github.com/GuillermoCharlesSchneider/DATA-607/raw/main/CPI%202024.xlsx"

# the read_excel function can only work on local files,
# so it'll need to be downloaded
download.file(data_url_1, "guis_cpi.xlsx", mode = "wb")

cpi_df <- read_excel("guis_cpi.xlsx", sheet = "Sheet0")
```

```
## New names:
## * `` -> `...1`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
```

```
head(cpi_df)
```

```
## # A tibble: 6 x 11
##   ...1    Table 1. Consumer Pr~1 ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10
##   <chr>   <chr>                  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 <NA>    [1982-84=100, unless ~  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## 2 <NA>    <NA>                    <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## 3 Indent~ Expenditure category   "Rel~ "Una~ "Una~ "Una~ "Una~ "Una~ "Sea~ "Sea~
## 4 <NA>    <NA>                    <NA> "Jan~ "Dec~ "Jan~ "Jan~ "Dec~ "Oct~ "Nov~
## 5 <NA>    <NA>                    <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
## 6 0       All items              "100" "299~ "306~ "308~ "3.1" "0.5" "0.2" "0.2"
## # i abbreviated name:
## #   1: `Table 1. Consumer Price Index for All Urban Consumers (CPI-U): U.S. city average, by expendi
## # i 1 more variable: ...11 <chr>
```

This data is pretty messy. So to clean it up, let's start with the unecessary rows. We'll do this by removing any `NA` entries in the fourth column `...4`:

```
cpi_df <- cpi_df |>
  drop_na(...4)

head(cpi_df)
```

```
## # A tibble: 6 x 11
##   ...1    Table 1. Consumer Pr~1 ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10
##   <chr>   <chr>                  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Indent~ Expenditure category   "Rel~ "Una~ "Una~ "Una~ "Una~ "Una~ "Sea~ "Sea~
## 2 <NA>    <NA>                    <NA> "Jan~ "Dec~ "Jan~ "Jan~ "Dec~ "Oct~ "Nov~
## 3 0       All items              "100" "299~ "306~ "308~ "3.1" "0.5" "0.2" "0.2"
## 4 1       Food                   "13.~ "319~ "325~ "327~ "2.6" "0.6" "0.2" "0.2"
## 5 2       Food at home           "8.1~ "301~ "303~ "305~ "1.2" "0.7" "0"   "0.1"
```

```
## 6 3       Cereals and bakery pr~ "1.0~ "349~ "353~ "354~ "1.5" "0.2" "0.3" "-0.~
## # i abbreviated name:
## #   1: 'Table 1. Consumer Price Index for All Urban Consumers (CPI-U): U.S. city average, by expendi
## # i 1 more variable: ...11 <chr>
```

There's good descriptive information in the first two rows of data. In order to ensure that we don't lose any information, we will need to combine the first two rows and turn that into the header then remove the first two rows from the dataset. I discovered that I can simply use the paste function as *paste(row_1, row_2)* in order to combine each columns values in these rows. Using that, I can change the column name and remove these two rows.

```
colnames(cpi_df) <- paste(cpi_df[1, ], cpi_df[2, ])

# Removing a row will need to be done twice.
# This will be performed with a for loop
rows_removed <- 0
while (rows_removed < 2) {
  cpi_df <- cpi_df[-1, ]
  rows_removed <- rows_removed + 1
}

head(cpi_df)
```

```
## # A tibble: 6 x 11
##   'Indent Level NA' 'Expenditure category NA'      Relative\r\nimportance\r\nD~1
##   <chr>             <chr>                          <chr>
## 1 0                 All items                      100
## 2 1                 Food                           13.555
## 3 2                 Food at home                   8.1669999999999998
## 4 3                 Cereals and bakery products    1.0660000000000001
## 5 3                 Meats, poultry, fish, and eggs 1.722
## 6 3                 Dairy and related products     0.748
## # i abbreviated name: 1: 'Relative\r\nimportance\r\nDec.\r\n2023 NA'
## # i 8 more variables: 'Unadjusted indexes Jan.\r\n2023' <chr>,
## #   'Unadjusted indexes Dec.\r\n2023' <chr>,
## #   'Unadjusted indexes Jan.\r\n2024' <chr>,
## #   'Unadjusted percent change Jan.\r\n2023-\r\nJan.\r\n2024' <chr>,
## #   'Unadjusted percent change Dec.\r\n2023-\r\nJan.\r\n2024' <chr>,
## #   'Seasonally adjusted percent change Oct.\r\n2023-\r\nNov.\r\n2023' <chr>, ...
```

I really don't like these column names, so let's clean these up by removing the \r, \n, and \t values.

This can be done using gsub and regex looking for any non-alpha-numeric items of length 1 or more and replacing them with an _ character.

```
colnames(cpi_df) <- gsub(
  "[^a-zA-Z0-9]{1,}",
  "_",
  colnames(cpi_df)
)

colnames(cpi_df)
```

```
##  [1] "Indent_Level_NA"
##  [2] "Expenditure_category_NA"
##  [3] "Relative_importance_Dec_2023_NA"
##  [4] "Unadjusted_indexes_Jan_2023"
##  [5] "Unadjusted_indexes_Dec_2023"
##  [6] "Unadjusted_indexes_Jan_2024"
##  [7] "Unadjusted_percent_change_Jan_2023_Jan_2024"
##  [8] "Unadjusted_percent_change_Dec_2023_Jan_2024"
##  [9] "Seasonally_adjusted_percent_change_Oct_2023_Nov_2023"
## [10] "Seasonally_adjusted_percent_change_Nov_2023_Dec_2023"
## [11] "Seasonally_adjusted_percent_change_Dec_2023_Jan_2024"
```

Now that we have our columns, let's go ahead and convert numeric columns into numeric datatypes. This seems to be all the columns in the dataframe. In order to pass all the columns in the dataframe, we'll use the `setdiff(columns, columnn_to_exclude)` function to select all columns except the one we're interested in.

Then, we can convert the remaining columns using the `mutate_at()` function:

```
numeric_cols <- setdiff(colnames(cpi_df), "Expenditure_category_NA")

cpi_df <- cpi_df |>
  mutate_at(vars(numeric_cols), as.numeric)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(numeric_cols)
##
##   # Now:
##   data %>% select(all_of(numeric_cols))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
str(cpi_df)
```

```
## tibble [37 x 11] (S3: tbl_df/tbl/data.frame)
##  $ Indent_Level_NA                                  : num [1:37] 0 1 2 3 3 3 3 3 3 2 ...
##  $ Expenditure_category_NA                          : chr [1:37] "All items" "Food" "Food at home
##  $ Relative_importance_Dec_2023_NA                  : num [1:37] 100 13.55 8.17 1.07 1.72 ...
##  $ Unadjusted_indexes_Jan_2023                      : num [1:37] 299 319 301 349 323 ...
##  $ Unadjusted_indexes_Dec_2023                      : num [1:37] 307 325 303 354 320 ...
##  $ Unadjusted_indexes_Jan_2024                      : num [1:37] 308 327 305 355 320 ...
##  $ Unadjusted_percent_change_Jan_2023_Jan_2024      : num [1:37] 3.1 2.6 1.2 1.5 -0.9 -1.1 1.1 3.4
##  $ Unadjusted_percent_change_Dec_2023_Jan_2024      : num [1:37] 0.5 0.6 0.7 0.2 -0.1 0.4 1.3 2.2
##  $ Seasonally_adjusted_percent_change_Oct_2023_Nov_2023: num [1:37] 0.2 0.2 0 0.3 -0.2 0 0.1 0.4 -0.1
##  $ Seasonally_adjusted_percent_change_Nov_2023_Dec_2023: num [1:37] 0.2 0.2 0.1 -0.1 0.3 0.1 0 0.2 0
##  $ Seasonally_adjusted_percent_change_Dec_2023_Jan_2024: num [1:37] 0.3 0.4 0.4 -0.2 0 0.2 0.4 1.2 0
```

In most instances, I find that the percent change isn't available but this dataset seems to already have these fields calculated. So I will proceed by simply sorting the data by the `Unadjusted indexes Dec 2023` and

then resort by `Unadjusted indexes Jan 2024` in descending order to see which categories had the greatest increase.

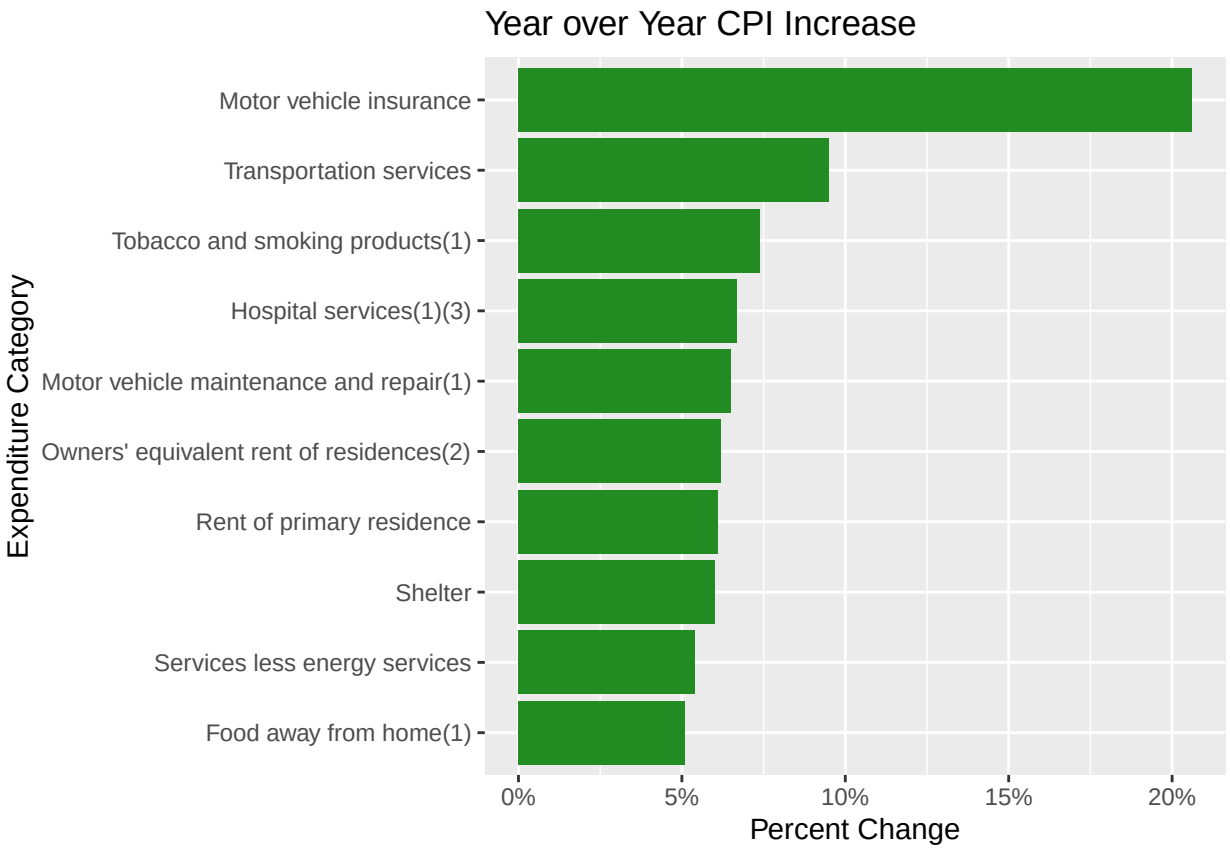
```
top_yoy_changes <- cpi_df |>
  arrange(desc(Unadjusted_percent_change_Jan_2023_Jan_2024)) |>
  select(
    Indent_Level_NA,
    Expenditure_category_NA,
    Unadjusted_percent_change_Jan_2023_Jan_2024
  ) |>
  top_n(10)
```

```
## Selecting by Unadjusted_percent_change_Jan_2023_Jan_2024
```

```
top_yoy_changes
```

```
## # A tibble: 10 x 3
##    Indent_Level_NA Expenditure_category_NA                Unadjusted_percent_c~1
##              <dbl> <chr>                                                   <dbl>
##  1               4 Motor vehicle insurance                                  20.6
##  2               3 Transportation services                                   9.5
##  3               3 Tobacco and smoking products(1)                           7.4
##  4               4 Hospital services(1)(3)                                   6.7
##  5               4 Motor vehicle maintenance and repair(~                    6.5
##  6               4 Owners' equivalent rent of residences~                    6.2
##  7               4 Rent of primary residence                                 6.1
##  8               3 Shelter                                                   6
##  9               2 Services less energy services                             5.4
## 10               2 Food away from home(1)                                    5.1
## # i abbreviated name: 1: Unadjusted_percent_change_Jan_2023_Jan_2024
```

```
ggplot(
  data = top_yoy_changes,
  aes(
    x = Unadjusted_percent_change_Jan_2023_Jan_2024 / 100,
    y = reorder(
      Expenditure_category_NA,
      Unadjusted_percent_change_Jan_2023_Jan_2024
    )
  )
) +
  geom_bar(
    stat = "identity",
    fill = "forestgreen"
  ) +
  labs(
    x = "Percent Change",
    y = "Expenditure Category",
    title = "Year over Year CPI Increase "
  ) +
  scale_x_continuous(labels = scales::percent_format())
```



The code above has returned the top 10 expense categories and their year over year percent change. From here we can see that motor vehicle insurance has gone up **a whopping 20.6%**! The New York Times seems to have an article where they go over some reasons for this but this was something I wasn't aware of at all, thanks to living in NYC. It was unintentional, but it seems that each of the datasets I chosen will have an element investigating cars.

Following Motor Vehicle insurance, I see that transportation services and tobacco take the second and third place spots for increases in cost at 9.5% and 7.4%, respectively.

Now I'm curious, did anything get cheaper? We can investigate this by flipping our `arrange()`. This can be done by removing the `desc()`. Additionally, we will need to change our `top_n(10)` to be a `-10`:

```
bot_yoy_changes <- cpi_df |>
  arrange(Unadjusted_percent_change_Jan_2023_Jan_2024) |>
  select(
    Indent_Level_NA,
    Expenditure_category_NA,
    Unadjusted_percent_change_Jan_2023_Jan_2024
  ) |>
  top_n(-10)
```

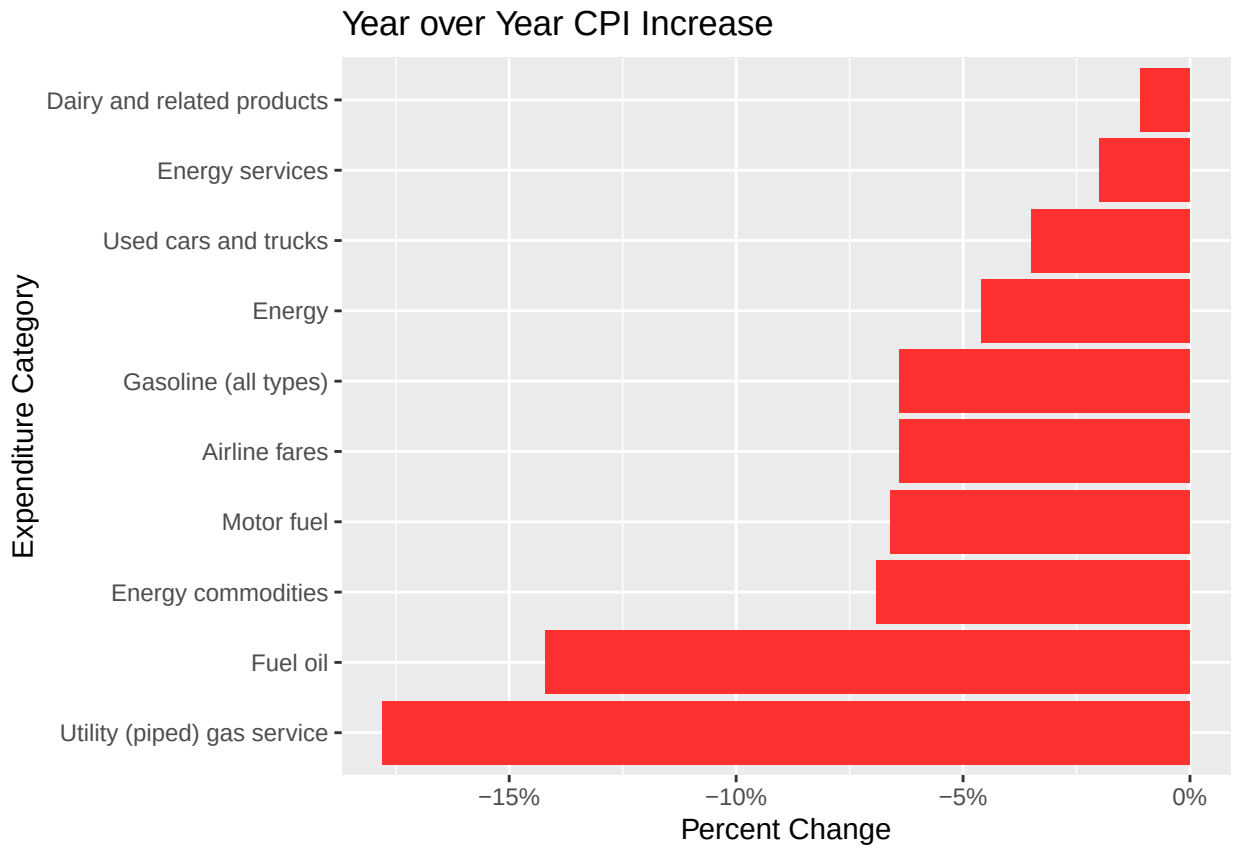
```
## Selecting by Unadjusted_percent_change_Jan_2023_Jan_2024
```

```
bot_yoy_changes
```

```
## # A tibble: 10 x 3
```

```
##    Indent_Level_NA Expenditure_category_NA     Unadjusted_percent_change_Jan_2~1
##              <dbl> <chr>                                                   <dbl>
##  1               3 Utility (piped) gas service                            -17.8
##  2               3 Fuel oil                                               -14.2
##  3               2 Energy commodities                                      -6.9
##  4               3 Motor fuel                                              -6.6
##  5               4 Gasoline (all types)                                    -6.4
##  6               4 Airline fares                                           -6.4
##  7               1 Energy                                                  -4.6
##  8               3 Used cars and trucks                                    -3.5
##  9               2 Energy services                                         -2
## 10               3 Dairy and related products                              -1.1
## # i abbreviated name: 1: Unadjusted_percent_change_Jan_2023_Jan_2024
```

```
ggplot(
  data = bot_yoy_changes,
  aes(
    x = Unadjusted_percent_change_Jan_2023_Jan_2024 / 100,
    y = reorder(
      Expenditure_category_NA,
      Unadjusted_percent_change_Jan_2023_Jan_2024
    )
  )
) +
  geom_bar(
    stat = "identity",
    fill = "firebrick1"
  ) +
  labs(
    x = "Percent Change",
    y = "Expenditure Category",
    title = "Year over Year CPI Increase "
  ) +
  scale_x_continuous(labels = scales::percent_format())
```



My first impression here is my sympathy for the environment. Each of the items on this list are large contributors of climate change with 8 of the 10 top items directly burning hydrocarbons. Admittedly, two of these entries are "parent" categories to others on this list, it's not a good sign that it's getting even cheaper to pollute the atmosphere.

Moving on, let's investigate the change From December 2023 to January 2024. We will do this by applying the same approach as year over year:

```
top_mom_changes <- cpi_df |>
  arrange(desc(Unadjusted_percent_change_Dec_2023_Jan_2024)) |>
  select(
    Indent_Level_NA,
    Expenditure_category_NA,
    Unadjusted_percent_change_Dec_2023_Jan_2024
  ) |>
  top_n(10)
```

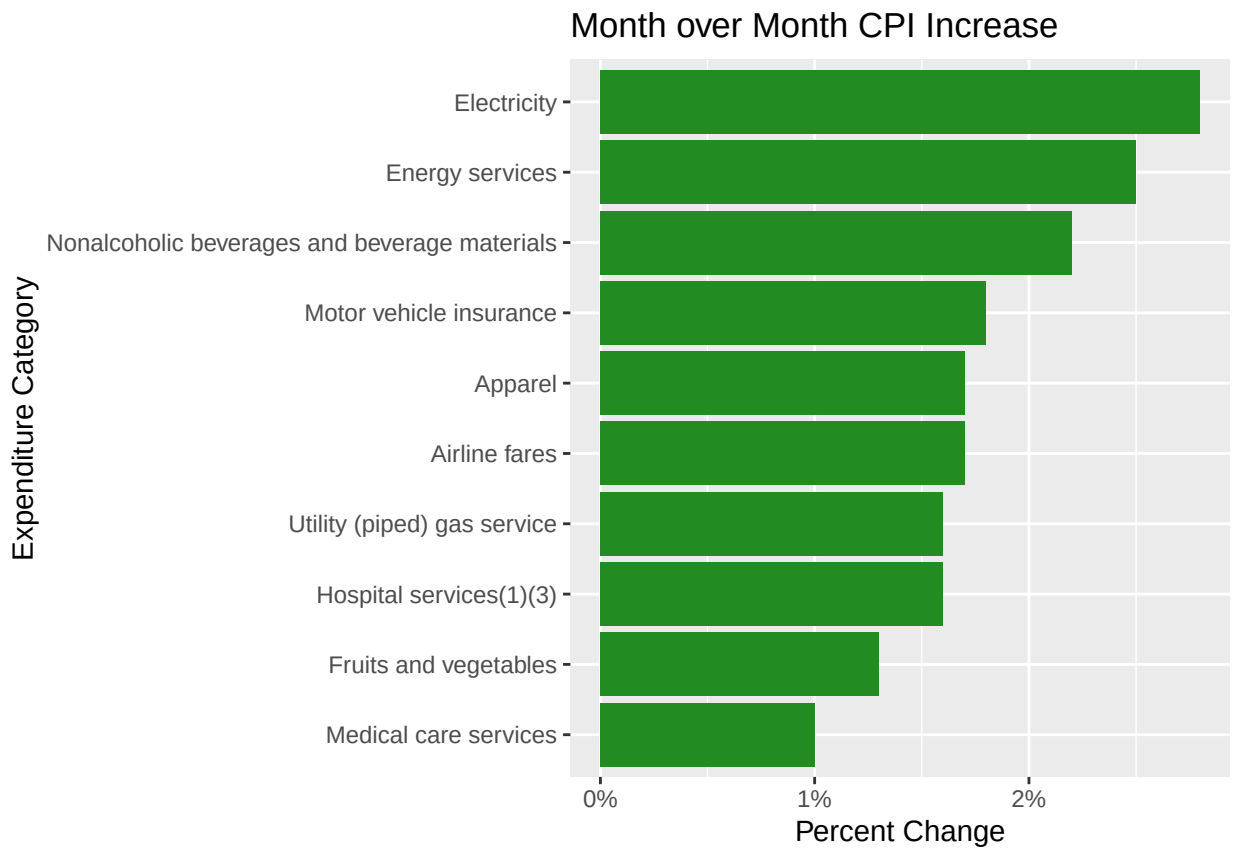
```
## Selecting by Unadjusted_percent_change_Dec_2023_Jan_2024
```

```
top_mom_changes
```

```
## # A tibble: 10 x 3
##    Indent_Level_NA Expenditure_category_NA              Unadjusted_percent_c~1
##              <dbl> <chr>                                                 <dbl>
##  1               3 Electricity                                             2.8
##  2               2 Energy services                                         2.5
```

```
##  3              3 Nonalcoholic beverages and beverage m~                  2.2
##  4              4 Motor vehicle insurance                                 1.8
##  5              3 Apparel                                                 1.7
##  6              4 Airline fares                                           1.7
##  7              3 Utility (piped) gas service                             1.6
##  8              4 Hospital services(1)(3)                                 1.6
##  9              3 Fruits and vegetables                                   1.3
## 10              3 Medical care services                                   1
## # i abbreviated name: 1: Unadjusted_percent_change_Dec_2023_Jan_2024
```

```r
ggplot(
  data = top_mom_changes,
  aes(
    x = Unadjusted_percent_change_Dec_2023_Jan_2024 / 100,
    y = reorder(
      Expenditure_category_NA,
      Unadjusted_percent_change_Dec_2023_Jan_2024
    )
  )
) +
  geom_bar(
    stat = "identity",
    fill = "forestgreen"
  ) +
  labs(
    x = "Percent Change",
    y = "Expenditure Category",
    title = "Month over Month CPI Increase "
  ) +
  scale_x_continuous(labels = scales::percent_format())
```



From here, we see that month over month, Electricity and Energy services have increased the most at 2.5% and 2.5%, respectively. These are pretty significant monthly changes although I theorize that these two categories may be correlated with the U.S experiencing winter. As the months get colder, energy demand increases across the board. This is backed up by the U.S. Energy Information Administration.

Now, as I'm curious, I'd also like to see what has gotten cheaper month to month:

```
bot_mom_changes <- cpi_df |>
  arrange(Unadjusted_percent_change_Dec_2023_Jan_2024) |>
  select(
    Indent_Level_NA,
    Expenditure_category_NA,
    Unadjusted_percent_change_Dec_2023_Jan_2024
  ) |>
  top_n(-10)
```

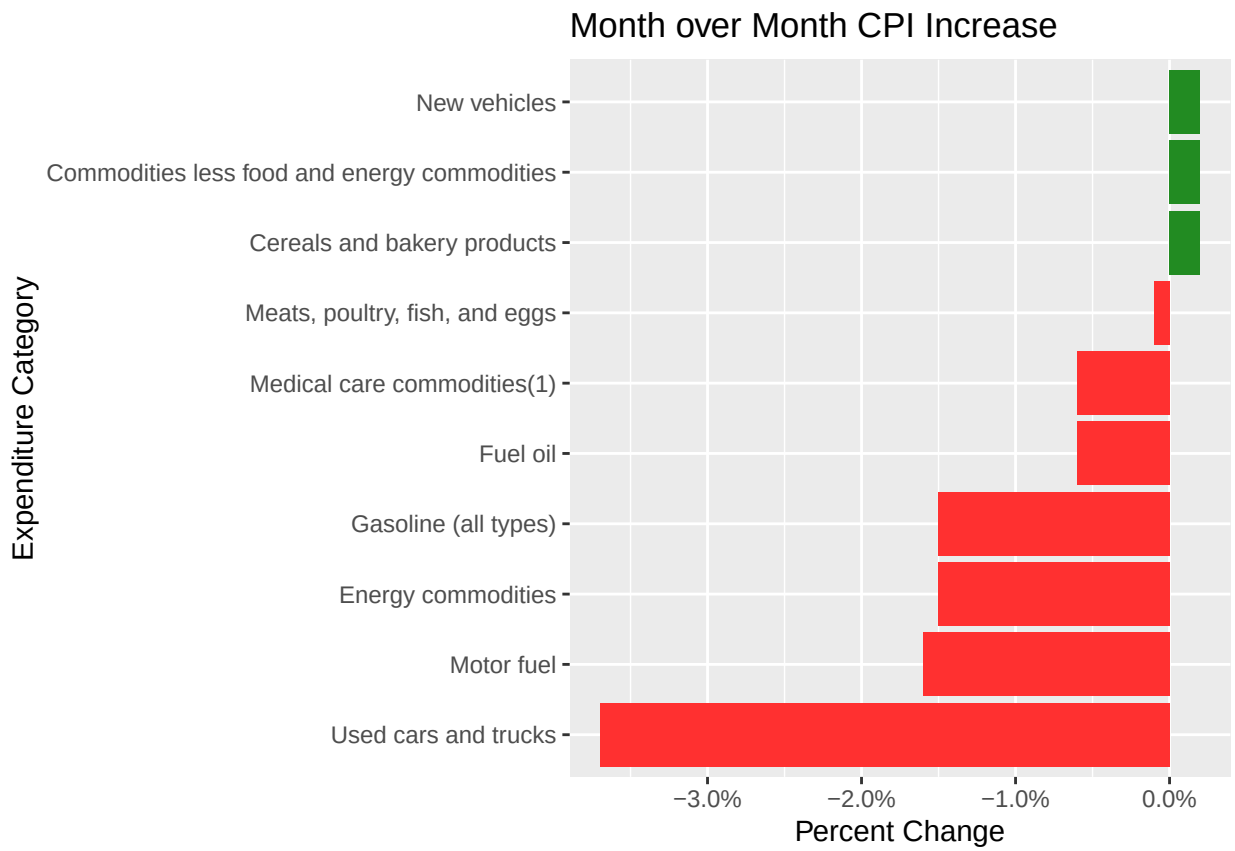
```
## Selecting by Unadjusted_percent_change_Dec_2023_Jan_2024
```

```
bot_mom_changes
```

```
## # A tibble: 10 x 3
##    Indent_Level_NA Expenditure_category_NA              Unadjusted_percent_c~1
##              <dbl> <chr>                                                 <dbl>
##  1               3 Used cars and trucks                                   -3.7
##  2               3 Motor fuel                                             -1.6
##  3               2 Energy commodities                                     -1.5
```

```
##  4               4 Gasoline (all types)                                    -1.5
##  5               3 Fuel oil                                                -0.6
##  6               3 Medical care commodities(1)                             -0.6
##  7               3 Meats, poultry, fish, and eggs                          -0.1
##  8               3 Cereals and bakery products                              0.2
##  9               2 Commodities less food and energy comm~                   0.2
## 10               3 New vehicles                                             0.2
## # i abbreviated name: 1: Unadjusted_percent_change_Dec_2023_Jan_2024
```

```r
ggplot(
  data = bot_mom_changes,
  aes(
    x = Unadjusted_percent_change_Dec_2023_Jan_2024 / 100,
    y = reorder(
      Expenditure_category_NA,
      Unadjusted_percent_change_Dec_2023_Jan_2024
    )
  )
) +
  geom_bar(
    aes(
      fill = ifelse(
        Unadjusted_percent_change_Dec_2023_Jan_2024 < 0,
        "firebrick1",
        "forestgreen"
      )
    ),
    stat = "identity"
  ) +
  labs(
    x = "Percent Change",
    y = "Expenditure Category",
    title = "Month over Month CPI Increase "
  ) +
  scale_x_continuous(labels = scales::percent_format()) +
  scale_fill_identity()
```



I'm not sure exactly what I was expecting, but significantly lowered used car prices was not one of them although many of the energy products which were cheaper year over year were also cheaper month to month. After the 6th entry on the list (Medical care commodities), there is a pretty small change in the month-to-month cost, which I was anticipating.

2. Z. Liang's "U.S Vehicle Model Sales"

In this dataset, Zixian suggest that we investigate the change in sales between motor vehicles. Using data from goodbadcar.net, we can compare relative year over year growth of cars as this dataset represents total sales volume per vehicle brand and model.

Doing research for this, I found a useful package called `rvest` which allows me to easily scrape data from an online table by simply specifying the table tag.

In order to read this data, we'll use the HTML of the website along with the `read_html` and `html_table` functions to take the raw html from the website and parse the contents to obtain the data from the second table. This second table contains the year over year growth of sales per model of car.

```
data_url_2 <- "https://www.goodcarbadcar.net/2023-us-vehicle-sales-figures-by-model/"
webpage <- read_html(data_url_2)

table_id <- "table_2"
selected_table <- html_table(html_nodes(webpage, sprintf("#%s", table_id)))

car_sales_data <- selected_table[[1]]
```

```r
colnames(car_sales_data) <- make.names(colnames(car_sales_data))

car_sales_data
```

```
## # A tibble: 328 x 5
##    modelName          Q4.2023 Q4.2022 Year.To.Date Year.to.Date.Previous.Year
##    <chr>              <chr>   <chr>   <chr>        <chr>
##  1 Acura ILX          0       0       2            6,296
##  2 Acura Integra      7,256   6,989   32,090       13,027
##  3 Acura MDX          12,680  13,211  57,599       46,425
##  4 Acura NSX          0       87      5            298
##  5 Acura RDX          12,026  5,013   39,228       24,749
##  6 Acura RLX          0       0       0            3
##  7 Acura TLX          3,158   2,354   16,731       11,508
##  8 Alfa Romeo Giulia  766     1,431   3,461        5,091
##  9 Alfa Romeo Stelvio 1,307   1,601   5,338        7,752
## 10 Alfa Romeo Tonale  1,234   0       2,098        0
## # i 318 more rows
```

We must convert the sales volume columns into numeric columns:

```r
car_numeric_cols <- setdiff(colnames(car_sales_data), "modelName")

car_sales_data <- car_sales_data |>
  mutate(across(car_numeric_cols, ~as.numeric(gsub(",", "", .))))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(car_numeric_cols, ~as.numeric(gsub(",", "", .)))`.
## Caused by warning:
## ! Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(car_numeric_cols)
##
##   # Now:
##   data %>% select(all_of(car_numeric_cols))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```r
str(car_sales_data)
```

```
## tibble [328 x 5] (S3: tbl_df/tbl/data.frame)
##  $ modelName                : chr [1:328] "Acura ILX" "Acura Integra" "Acura MDX" "Acura NSX" ...
##  $ Q4.2023                  : num [1:328] 0 7256 12680 0 12026 ...
##  $ Q4.2022                  : num [1:328] 0 6989 13211 87 5013 ...
##  $ Year.To.Date             : num [1:328] 2 32090 57599 5 39228 ...
##  $ Year.to.Date.Previous.Year: num [1:328] 6296 13027 46425 298 24749 ...
```

This dataset seems relatively complete. With this, we are able to determine much about cars such as the total year over year change and percent change. With that, we will look at the top 10 and bottom 10 models by total year over change and compare that to the percent change.

```r
car_sales_data <- car_sales_data |>
  mutate(yoy_change = Q4.2023 - Q4.2022) |>
  mutate(yoy_pct_change = round(yoy_change / Q4.2022, 2))

average_growth <- sum(car_sales_data$yoy_change) / sum(car_sales_data$Q4.2022)

average_growth
```

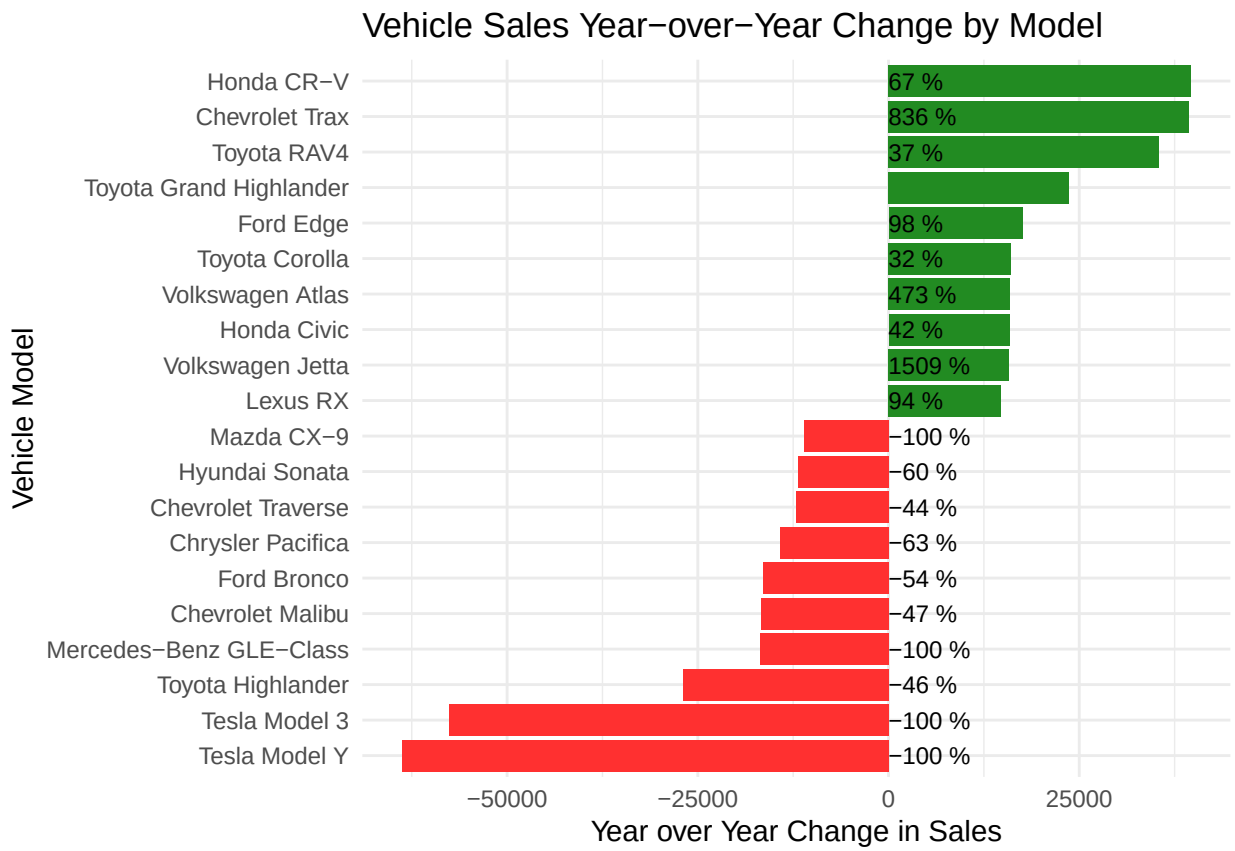
```
## [1] 0.01684737
```

```r
top_car_sales <- car_sales_data |>
  top_n(10, yoy_change)

bot_car_sales <- car_sales_data |>
  top_n(-10, yoy_change)

visualization_data <- rbind(
  top_car_sales,
  bot_car_sales
)

visualization_data |>
  select(
    modelName,
    yoy_change,
    yoy_pct_change
  ) |>
  arrange(
    desc(yoy_change)
  )
```

```
## # A tibble: 20 x 3
##    modelName               yoy_change yoy_pct_change
##    <chr>                        <dbl>          <dbl>
##  1 Honda CR-V                   39638           0.67
##  2 Chevrolet Trax               39410           8.36
##  3 Toyota RAV4                  35512           0.37
##  4 Toyota Grand Highlander      23660         Inf
##  5 Ford Edge                    17555           0.98
##  6 Toyota Corolla               16017           0.32
##  7 Volkswagen Atlas             15974           4.73
##  8 Honda Civic                  15854           0.42
##  9 Volkswagen Jetta             15804          15.1
## 10 Lexus RX                     14800           0.94
## 11 Mazda CX-9                  -11100          -1
## 12 Hyundai Sonata              -11890          -0.6
## 13 Chevrolet Traverse          -12134          -0.44
## 14 Chrysler Pacifica           -14255          -0.63
## 15 Ford Bronco                 -16473          -0.54
## 16 Chevrolet Malibu            -16775          -0.47
## 17 Mercedes-Benz GLE-Class     -16800          -1
## 18 Toyota Highlander           -26989          -0.46
## 19 Tesla Model 3               -57600          -1
```

```
## 20 Tesla Model Y               -63800          -1
```

```
car_sales_yoy <- ggplot(
  visualization_data,
  aes(y = reorder(modelName, yoy_change))
) +
  geom_bar(
    aes(
      x = yoy_change,
      fill = ifelse(
        yoy_change < 0,
        "firebrick1",
        "forestgreen"
      )
    ),
    stat = "identity",
    position = "dodge"
  ) +
  geom_text(
    aes(
      x = yoy_pct_change,
      label = paste(100 * yoy_pct_change, "%")
    ),
    position = position_dodge(width = 0.75),
    hjust = 0,
    size = 3
  ) +
  theme_minimal() +
  labs(
    x = "Year over Year Change in Sales",
    y = "Vehicle Model",
    title = "Vehicle Sales Year-over-Year Change by Model"
  ) +
  scale_fill_identity()

car_sales_yoy
```



The graph above shows the vehicle models with the greatest increase in total car sales and the greatest decrease in sales. From here, we can see that despite the Honda CR-V having the greatest year over year increase, it's only had a 67% growth. The second entry in the list, the Chevrolet Trax, has a comparable increase in sales year over year but a staggering 836% increase! Even more impressive, the Volkswagen Jetta has a 15100% Increase in sales! Additionally, we can see that there was an infinite increase in the Toyota Grant Highlander. I assume that this means that there were no sales the previous year:

```
car_sales_data |>
  filter(modelName == "Toyota Grand Highlander")
```

```
## # A tibble: 1 x 7
##   modelName       Q4.2023 Q4.2022 Year.To.Date Year.to.Date.Previou~1 yoy_change
##   <chr>             <dbl>   <dbl>        <dbl>                  <dbl>      <dbl>
## 1 Toyota Grand H~   23660       0        39373                      0      23660
## # i abbreviated name: 1: Year.to.Date.Previous.Year
## # i 1 more variable: yoy_pct_change <dbl>
```

It looks light I was right, there were no sales in 2022!

3. R. Rivera's "Car Crash information over time (1994-Present)"

For this analysis, I will be attempting to investigate car crashes by vehicle type and people type fatalities. To outline a few definitions, vehicle type is a class of vehicle. These can be:

- Passenger Cars

- Light Trucks
- Large Trucks
- Motor Cycles
- Busses
- Other Vehicles (Limousines, Motorhomes, Farm Equipment, etc.)

Also, for people types, we will use the people types that are outlined in this data:

- Vehicle Occupants (Driver, Passenger)
- Motorcyclists
- Nonmotorists - Pedestrian
- Nonmotorists - Pedalcyclist

For a full list of descriptive definitions, please refer to the NHTSA terms help website.

Although there was an API offered, it appears that there isn't a sufficient amount of information available to utilize it for these purposes. To ensure that I had the sufficient amount of granularity for the analysis, I needed to use the Fatality and Injury Reporting System Tool Query offered. Exports of this file can be found in this repository:

The datasets we will read below are:

1. Vehicles Involved in Fatal Crashes (`vehicle_data`) - This dataset contains the number of vehicles in fatal crashes by year, month, and vehicle type.
2. Persons Involved in Fatal Crashes (`people_data`) - This dataset contains the number of people involved in fatal crashes by year, month, and person type.
3. Fatal Motor Vehicle Crashes (`crash_data`) - This dataset contains the count of motor crash incidents by month, year, and the type of crash.

First we will need to download the data:

```
# set up the URLs for the file's we need
urls <- list(
  c(
    "https://github.com/riverar9/cuny-msds/raw/main/data607/projects/project-2/vehicle_crash_data.xlsx"
    "vehicle_crash_data.xlsx"
  ),
  c(
    "https://github.com/riverar9/cuny-msds/raw/main/data607/projects/project-2/fatal_crash_data.xlsx",
    "fatal_crash_data.xlsx"
  ),
  c(
    "https://github.com/riverar9/cuny-msds/raw/main/data607/projects/project-2/people_fatality_data.xls
    "people_fatality_data.xlsx"
  )
)

# Iterate through the URLs and download them to the working directory
for (url in urls) {
  download.file(url[1], url[2], mode = "wb")
}
```

With all the files downloaded, we can read each one and format the data. We will do that with the below cells.

While working, I noticed that I should create a function that will convert the month names into an ordered factor.

```
apply_month_factors <- function(df, month_column_name) {
  # Convert month_column_name into an ordered factor using months
  df <- df |>
     mutate(
      {{ month_column_name }} := factor({{ month_column_name }},
      levels = month.name,
      ordered = TRUE)
    )

  return(df)
}
```

```
vehicle_data <- read_excel(
  "vehicle_crash_data.xlsx",
  sheet = "CrashReport - Table 1",
  skip = 6
)
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
```

```
# specifying the column names
colnames(vehicle_data) <- c(
  "year",
  "month",
  "passenger_car",
  "light_truck_pickup",
  "light_truck_utility",
  "light_truck_van",
  "light_truck_other",
  "large_truck",
  "motorcycle",
  "bus",
  "other_vehicle",
  "vehicle_total"
)

# performing a forward fill on the year column
# and remove total from the month or year column.
# Finally, convert year into an integer
vehicle_data <- vehicle_data |>
  fill(year, .direction = "down") |>
  filter(month != "Total") |>
  filter(year != "Total") |>
  mutate(year = as.integer(year)) |>
  mutate(light_truck = light_truck_pickup +
           light_truck_utility + light_truck_van +
           light_truck_other) |>
  select(
```

```
    year,
    month,
    passenger_car,
    light_truck,
    large_truck,
    motorcycle,
    bus,
    other_vehicle,
    vehicle_total
  )

vehicle_data <- apply_month_factors(
  vehicle_data,
  month
)

str(vehicle_data)
```

```
## tibble [180 x 9] (S3: tbl_df/tbl/data.frame)
##  $ year         : int [1:180] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  $ month        : Ord.factor w/ 12 levels "January"<"February"<..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ passenger_car: num [1:180] 1791 1730 1925 1959 1923 ...
##  $ light_truck  : num [1:180] 1687 1593 1880 1762 1840 ...
##  $ large_truck  : num [1:180] 328 379 393 336 348 423 392 424 401 482 ...
##  $ motorcycle   : num [1:180] 138 151 392 438 647 626 686 660 635 489 ...
##  $ bus          : num [1:180] 26 26 38 19 26 17 13 19 20 28 ...
##  $ other_vehicle: num [1:180] 90 89 118 100 115 126 137 129 134 120 ...
##  $ vehicle_total: num [1:180] 4060 3968 4746 4614 4899 ...
```

```
people_data <- read_excel(
  "people_fatality_data.xlsx",
  sheet = "CrashReport - Table 1",
  skip = 6
)
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
```

```
# specifying the column names
colnames(people_data) <- c(
  "year",
  "month",
  "car_driver",
  "car_passenger",
  "car_occupant",
  "other_1",
  "pedestrian",
  "bicyclist",
  "other_2",
  "other_3",
  "other_4",
```

```
  "other_5",
  "other_6",
  "other_7",
  "other_8",
  "other_9",
  "people_total"
)

# performing a forward fill on the year column
# and remove total from the month or year column.
# Then, convert year into an integer.
# Finally, combine all the others into one
# and remove the rest
people_data <- people_data |>
  fill(year, .direction = "down") |>
  filter(month != "Total") |>
  filter(year != "Total") |>
  mutate(year = as.integer(year)) |>
  mutate(other_person = other_1 + other_2 +
           other_3 + other_4 + other_5 +
           other_6 + other_7 + other_8 +
           other_9) |>
  select(
    year,
    month,
    car_driver,
    car_passenger,
    car_occupant,
    pedestrian,
    bicyclist,
    other_person,
    people_total
  )

people_data <- apply_month_factors(
  people_data,
  month
)

str(people_data)
```

```
## tibble [180 x 9] (S3: tbl_df/tbl/data.frame)
##  $ year         : int [1:180] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  $ month        : Ord.factor w/ 12 levels "January"<"February"<..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ car_driver   : num [1:180] 4034 3950 4729 4593 4885 ...
##  $ car_passenger: num [1:180] 2349 2107 2606 2681 2741 ...
##  $ car_occupant : num [1:180] 22 30 31 19 13 26 23 28 14 28 ...
##  $ pedestrian   : num [1:180] 447 367 448 339 353 331 381 417 381 521 ...
##  $ bicyclist    : num [1:180] 40 50 48 47 66 81 74 65 78 71 ...
##  $ other_person : num [1:180] 23 39 26 37 29 21 21 28 46 26 ...
##  $ people_total : num [1:180] 6915 6543 7888 7716 8087 ...
```

```
crash_data <- read_excel(
  "fatal_crash_data.xlsx",
  sheet = "CrashReport - Table 1",
  skip = 6
)
```

```
## New names:
## * `` -> `...1`
## * `` -> `...2`
```

```
# specifying the column names
colnames(crash_data) <- c(
  "year",
  "month",
  "with_pedestrian",
  "without_pedestrian",
  "crash_total"
)

# performing a forward fill on the year column
# and remove total from the month or year column.
# Finally, convert year into an integer
crash_data <- crash_data |>
  fill(year, .direction = "down") |>
  filter(month != "Total") |>
  filter(year != "Total") |>
  mutate(year = as.integer(year))

crash_data <- apply_month_factors(
  crash_data,
  month
)

str(crash_data)
```

```
## tibble [180 x 5] (S3: tbl_df/tbl/data.frame)
##  $ year              : int [1:180] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  $ month             : Ord.factor w/ 12 levels "January"<"February"<..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ with_pedestrian   : num [1:180] 428 349 414 322 330 299 352 370 354 475 ...
##  $ without_pedestrian: num [1:180] 2304 2269 2678 2719 2982 ...
##  $ crash_total       : num [1:180] 2732 2618 3092 3041 3312 ...
```

For ease of use, let's combine all of this data into one wide table. Because all of these datasets cover every month from 2007 to 2021, we can use a left join and expect a one-to-one relationship between each dataset.

As we are combining 3 datasets, we will nest a join within another join and this should return a dataset with Year and Month are the unique row identifier and we have columns from each dataset:

```
combined_data <- left_join(
  crash_data,
  left_join(
    vehicle_data,
    people_data,
```

```
    by = c(
      "year",
      "month"
    )
  ),
  by = c(
    "year",
    "month"
  )
)

str(combined_data)
```

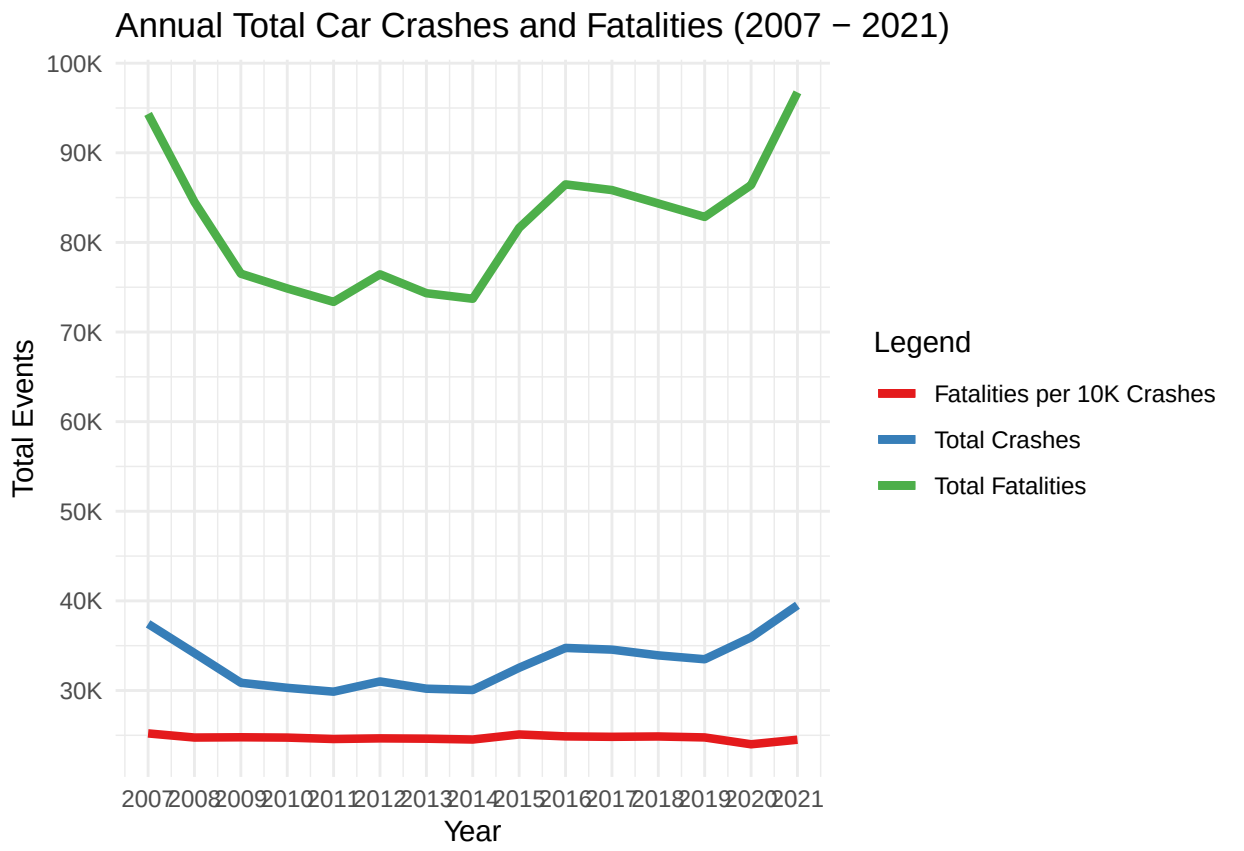
```
## tibble [180 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year              : int [1:180] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  $ month             : Ord.factor w/ 12 levels "January"<"February"<..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ with_pedestrian   : num [1:180] 428 349 414 322 330 299 352 370 354 475 ...
##  $ without_pedestrian: num [1:180] 2304 2269 2678 2719 2982 ...
##  $ crash_total       : num [1:180] 2732 2618 3092 3041 3312 ...
##  $ passenger_car     : num [1:180] 1791 1730 1925 1959 1923 ...
##  $ light_truck       : num [1:180] 1687 1593 1880 1762 1840 ...
##  $ large_truck       : num [1:180] 328 379 393 336 348 423 392 424 401 482 ...
##  $ motorcycle        : num [1:180] 138 151 392 438 647 626 686 660 635 489 ...
##  $ bus               : num [1:180] 26 26 38 19 26 17 13 19 20 28 ...
##  $ other_vehicle     : num [1:180] 90 89 118 100 115 126 137 129 134 120 ...
##  $ vehicle_total     : num [1:180] 4060 3968 4746 4614 4899 ...
##  $ car_driver        : num [1:180] 4034 3950 4729 4593 4885 ...
##  $ car_passenger     : num [1:180] 2349 2107 2606 2681 2741 ...
##  $ car_occupant      : num [1:180] 22 30 31 19 13 26 23 28 14 28 ...
##  $ pedestrian        : num [1:180] 447 367 448 339 353 331 381 417 381 521 ...
##  $ bicyclist         : num [1:180] 40 50 48 47 66 81 74 65 78 71 ...
##  $ other_person      : num [1:180] 23 39 26 37 29 21 21 28 46 26 ...
##  $ people_total      : num [1:180] 6915 6543 7888 7716 8087 ...
```

Now that we have all of our data, we can plot to see how car crashes have changed over time:

```
ggplot() +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = crash_total,
      color = "Total Crashes"
    ),
    fun = sum,
    geom = "line",
    size = 1.5,
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = people_total,
```

```
      color = "Total Fatalities"
    ),
    fun = sum,
    geom = "line",
    size = 1.5,
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = 10000 * (people_total / crash_total),
      color = "Fatalities per 10K Crashes"
    ),
    fun = mean,
    geom = "line",
    size = 1.5
  ) +
  labs(
    x = "Year",
    y = "Total Events",
    title = "Annual Total Car Crashes and Fatalities (2007 - 2021)",
    color = "Legend"
  ) +
  scale_color_brewer(
    palette = "Set1"
  ) +
  theme_minimal() +
  scale_x_continuous(
    breaks = seq(
      2007,
      2022,
      by = 1
    )
  ) +
  scale_y_continuous(
    breaks = seq(
      0,
      100000,
      by = 10000
    ),
    labels = scales::comma_format(
      scale = 1e-3,
      suffix = "K"
    )
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
total_crashes_2007 <-  combined_data |>
  filter(year == 2007) |>
  summarise(total_crashes = sum(crash_total)) |>
  select(total_crashes)
total_crashes_2021 <- combined_data |>
  filter(year == 2021) |>
  summarise(total_crashes = sum(crash_total)) |>
  select(total_crashes)

print(
  paste(
    "There has been a",
    round(
      100 * (total_crashes_2021$total_crashes / total_crashes_2007$total_crashes - 1), 1
    ),
    "% increase in crashes."
  )
)
```

```
## [1] "There has been a 5.5 % increase in crashes."
```

This graph tells us a bit. Firstly what stands out to me is that there was a significant decrease in the number of fatal crashes from 2007 to 2014, but there was a quick reversal since 2014 with the latest data suggesting that we are now experiencing more vehicle crashes than ever before. Additionally, we can visually notice a correlation between the total number of crashes and fatalities with the number of fatalities being much

greater than the number of crashes. It is important to remember that this dataset is only the number of fatal crashes, so by definition we must expect that the number of fatalities must be at least equal to the number of crashes. The last piece plotted here is the number of fatalities per 10,000 crashes. We can see here that this number is pretty stable suggesting that there is a strong correlation between total fatalities and fatal crashes (honestly, duh) although we can also notice that it sits around 25,000. Meaning that, on average, each fatal crash results in at least 2 fatalities.

Although this graph did a lot to confirm what may have been initial assumptions, one big takeaway is that the average fatality per crash hasn't changed very much.

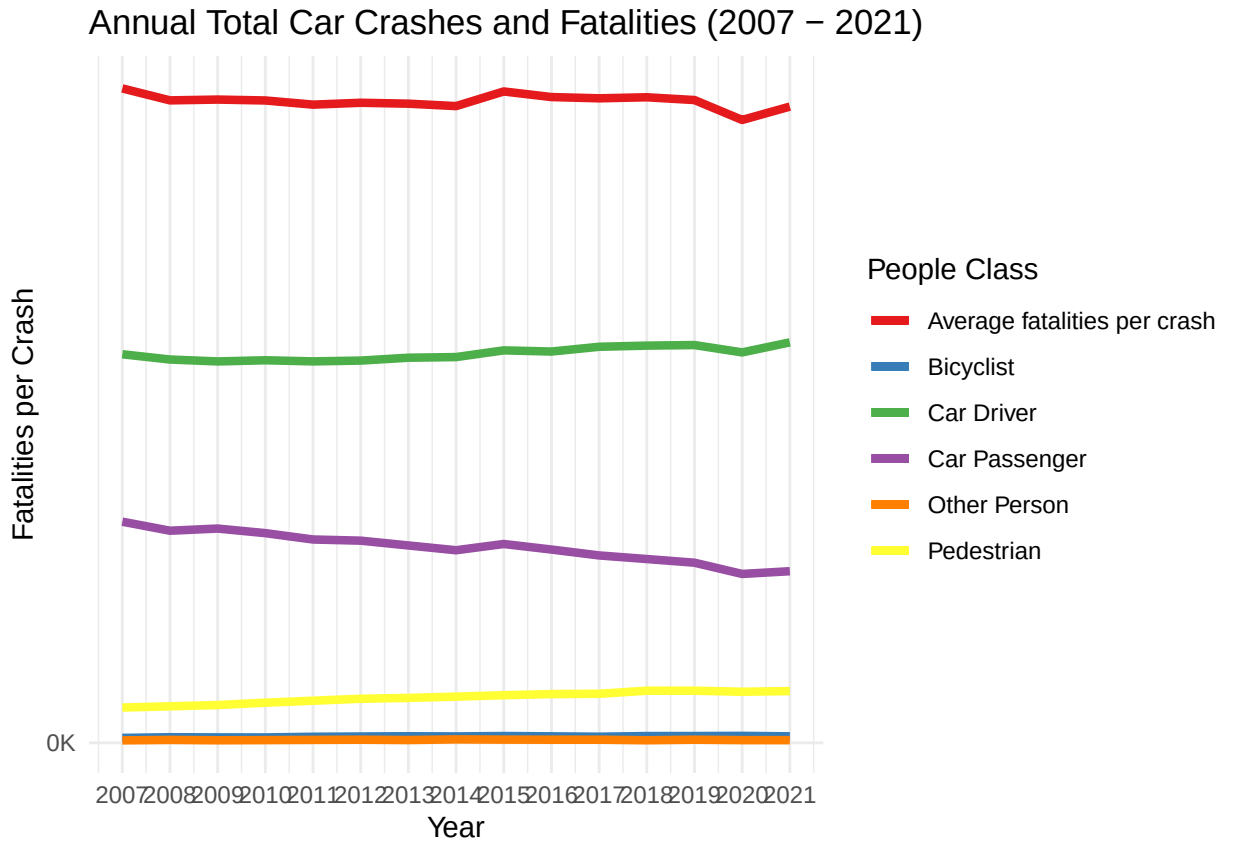
Here's an outstanding question, has there been any change in the groups of people who are experiencing fatalities? That is to say, is there a class of person who is experiencing a much higher or lower fatality rate?

In this next graph, we will plot the fatality rates by the person type. Fatality rate will be determined by a simple `(Person Class) / (Total Fatal Crashes)`

```
ggplot() +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = car_driver / crash_total,
      color = "Car Driver"
    ),
    fun = mean,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = car_passenger / crash_total,
      color = "Car Passenger"
    ),
    fun = mean,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = pedestrian / crash_total,
      color = "Pedestrian"
    ),
    fun = mean,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = bicyclist / crash_total,
      color = "Bicyclist"
```

```
  ),
  fun = mean,
  geom = "line",
  size = 1.5
) +
stat_summary(
  data = combined_data,
  aes(
    x = year,
    y = other_person / crash_total,
    color = "Other Person"
  ),
  fun = mean,
  geom = "line",
  size = 1.5
) +
stat_summary(
  data = combined_data,
  aes(
    x = year,
    y = people_total / crash_total,
    color = "Average fatalities per crash"
  ),
  fun = mean,
  geom = "line",
  size = 1.5
) +
labs(
  x = "Year",
  y = "Fatalities per Crash",
  title = "Annual Total Car Crashes and Fatalities (2007 - 2021)",
  color = "People Class"
) +
scale_color_brewer(
  palette = "Set1"
) +
theme_minimal() +
scale_x_continuous(
  breaks = seq(
    2007,
    2022,
    by = 1
  )
) +
scale_y_continuous(
  breaks = seq(
    0,
    100000,
    by = 10000
  ),
  labels = scales::comma_format(
    scale = 1e-3,
    suffix = "K"
```

```
    )
  )
```

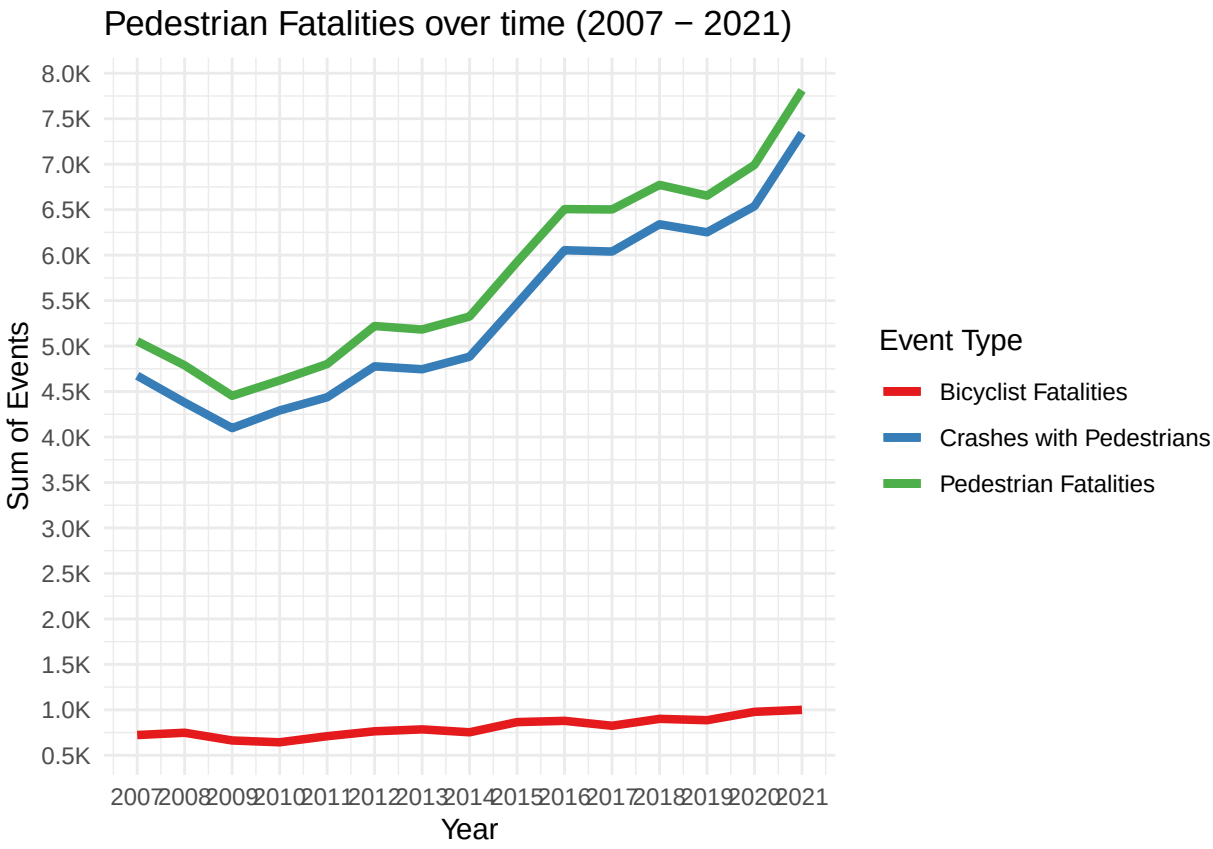


We've also plotted the total average number of fatalities per car crash to have a better visual feel for the relative contribution to the whole. From here, we can see that car drivers and pedestrians have seen an increase while car passengers have decreased. Although this doesn't help answer the question of why as there could be much fewer car passengers due to ride shares, loneliness, or other reasons.

I would like to zoom in on the pedestrian count, as they're fairly unprotected from vehicle crashes:

```
ggplot() +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = with_pedestrian,
      color = "Crashes with Pedestrians"
    ),
    fun = sum,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
```

```
    y = pedestrian,
    color = "Pedestrian Fatalities"
  ),
  fun = sum,
  geom = "line",
  size = 1.5
) +
stat_summary(
  data = combined_data,
  aes(
    x = year,
    y = bicyclist,
    color = "Bicyclist Fatalities"
  ),
  fun = sum,
  geom = "line",
  size = 1.5
) +
labs(
  x = "Year",
  y = "Sum of Events",
  title = "Pedestrian Fatalities over time (2007 - 2021)",
  color = "Event Type"
) +
scale_color_brewer(
  palette = "Set1"
) +
theme_minimal() +
scale_x_continuous(
  breaks = seq(
    2007,
    2022,
    by = 1
  )
) +
scale_y_continuous(
  breaks = seq(
    0,
    10000,
    by = 500
  ),
  labels = scales::comma_format(
    scale = 1e-3,
    suffix = "K"
  )
)
```




```
pedestrian_fatalities_2007 <-  combined_data |>
  filter(year == 2007) |>
  summarise(total_fatalities = sum(pedestrian))

pedestrian_fatalities_2021 <- combined_data |>
  filter(year == 2021) |>
  summarise(total_fatalities = sum(pedestrian))

print(
  paste(
    "There has been a",
    round(
      100 * (pedestrian_fatalities_2021$total_fatalities / pedestrian_fatalities_2007$total_fatalities
    ),
    "% increase in crashes."
  )
)
```

```
## [1] "There has been a 54.6 % increase in crashes."
```

This is not good. We are seeing a pretty significant increase in pedestrian fatalities from 2007 to 2021, a 54.6% increase!

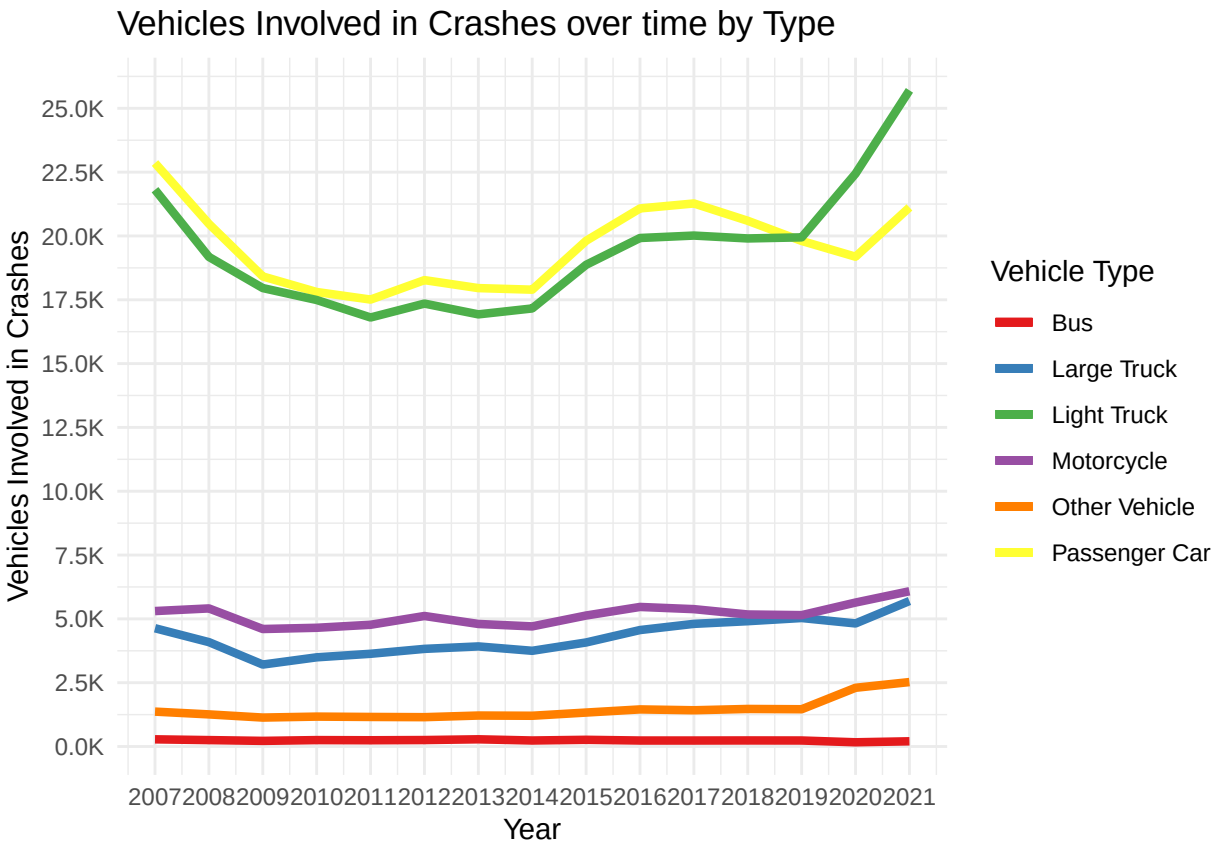
According to the US Department of Energy's AFDC, there has not been a significant change in the total number of vehicle miles in the US since 2007 (3.01 trillion miles) to 2021 (2.83 trillion miles).

Combining the crash data with the number of total miles driven it's pretty apparent that something is going on that is causing for more dangerous driving across the country. Additionally, pedestrians and car drivers are the major classes of people who are victims of these crashes.

Now finally, I would like to see if there is anything we can find by looking at the vehicle types:

```
ggplot() +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = passenger_car,
      color = "Passenger Car"
    ),
    fun = sum,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = light_truck,
      color = "Light Truck"
    ),
    fun = sum,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = large_truck,
      color = "Large Truck"
    ),
    fun = sum,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = motorcycle,
      color = "Motorcycle"
    ),
    fun = sum,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
```

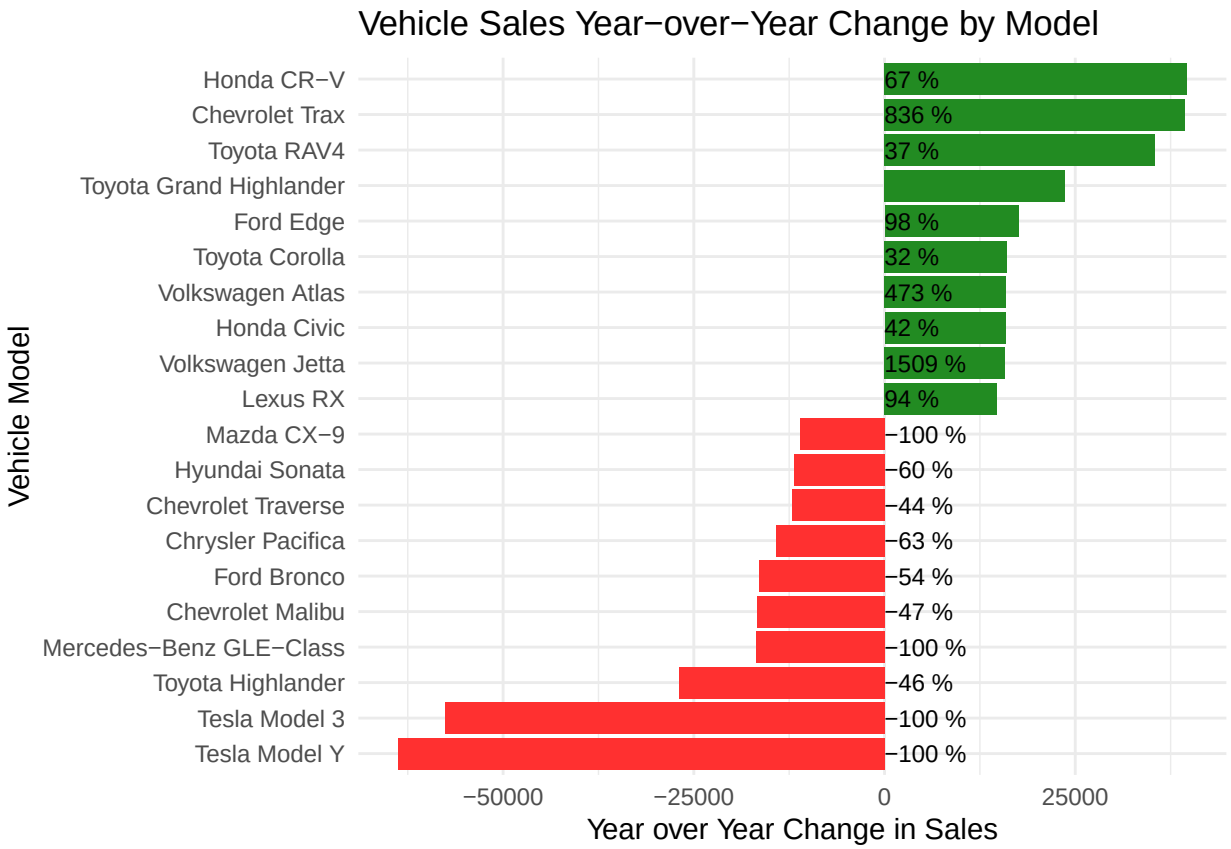
```
      x = year,
      y = bus,
      color = "Bus"
    ),
    fun = sum,
    geom = "line",
    size = 1.5
  ) +
  stat_summary(
    data = combined_data,
    aes(
      x = year,
      y = other_vehicle,
      color = "Other Vehicle"
    ),
    fun = sum,
    geom = "line",
    size = 1.5
  ) +
  labs(
    x = "Year",
    y = "Vehicles Involved in Crashes",
    title = "Vehicles Involved in Crashes over time by Type",
    color = "Vehicle Type"
  ) +
  scale_color_brewer(
    palette = "Set1"
  ) +
  theme_minimal() +
  scale_x_continuous(
    breaks = seq(
      2007,
      2022,
      by = 1
    )
  ) +
  scale_y_continuous(
    breaks = seq(
      0,
      30000,
      by = 2500
    ),
    labels = scales::comma_format(
      scale = 1e-3,
      suffix = "K"
    )
  )
```



Wow, again we see some major changes and a pretty worrying trend. My intuition tells me that the majority vehicle is a passenger car or a light truck and it seems that these two classes of vehicles are involved with the most crashes. Although, another interesting item to point out is that the number of light trucks has absolutely skyrocketed since 2019, not experiencing the same dip that is seen with passenger cars.

Here we may be able to find some overlap between that great increase in light truck crashes and the sales dataset. Revisiting the graph of the cars with the highest increase in sales:

```
car_sales_yoy
```



One thing I noticed here, is that all but 2 vehicles with the highest number of sales increase year over year would be classified as a light truck in the NHTSA data. Which means that light trucks may have a larger share of the vehicle type on the road which could help explain the increase in crashes with vehicles in this category.

These trends do not prove anything, but they are evidence that the roads are becoming more dangerous per mile driven for most people who interact with it.

Conclusion

In this exercise we've explored data across the Consumer Price Index, Vehicle Sales, and Fatal Crashes. Although there were quite a few things we've uncovered I will be breaking it down into two categories below:

In the respective dataset

This section contains the insights found in each individual dataset

1. The cost of energy (gas and electricity) has decreased year over year
2. The cost of used vehicles has decreased year over year
3. The cost of motor vehicle insurance has increased 20.6%
4. There have been a multitude of car models with incredible Year over Year sales growth
5. The majority of cars with the largest Year over Year sales growth are SUVs (Light Trucks)
6. The total number of total car crashes has increased by 5.5% from 2007 to present
7. The number of pedestrian fatalities has increased 54.6%

8. The number of trucks involved in crashes has increased

Combining some of these insights we can see:

1. The number of pedestrian fatalities has quickly outgrown the total number of car crashes, suggesting that something has happened causing for this increase.
2. The increase in sales of light trucks year over year may be a continuing trend, which could explain some of the increase light trucks have in fatal crashes.