Final Project - ST 308

Alan Ramirez

2025-04-24

Getting Started

Reading the `tidyverse` collection of packages to enable following code chunks to function:

```
library(tidyverse)
```

I also added the *message=FALSE* global option to prevent code messages from displaying in the outputted document. There was no mention of any incentive to do so in the project, but I figured the outputted PDF might as well look cleaner than otherwise.

Part 1

Response to Question 1

R-markdown is a tool that enables users to combine code, the code's output (graphs, images, figures, etc.), and narrative text that clarifies the function behind said codes in outputted files (HTML, PDF, or MS word), with said outputted files serving as reproducible research to be shared among fellow researchers who seek to read/run one's code or reproduce their work.

Response to Question 2

I uploaded the `.csv` data set that I downloaded from Moodle, read its data into the `house` object with the `read_csv` function, and then displayed the first six rows of `house` (the reiterated data set) using the `slice()` function.

```
house <-read_csv("aacline_house.csv")
```

```
## Rows: 1000 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): HouseStyle, BsmtFinType2, Foundation, RoofStyle, Exterior1st, LandS...
## dbl (7): SalePrice, BsmtFinSF1, GarageCars, YearBuilt, OpenPorchSF, WoodDeck...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
slice(house, 1:6)
```

```
## # A tibble: 6 x 13
##   SalePrice BsmtFinSF1 GarageCars YearBuilt OpenPorchSF WoodDeckSF YearRemodAdd
##       <dbl>      <dbl>      <dbl>     <dbl>       <dbl>      <dbl>        <dbl>
## 1    223500        486          2      2001          42          0         2002
## 2    140000        216          3      1915          35          0         1970
## 3    250000        655          3      2000          84        192         2000
## 4    200000        859          2      1973         204        235         1973
## 5    129500        906          1      1965           0          0         1965
## 6    345000        998          3      2005          21        147         2006
## # i 6 more variables: HouseStyle <chr>, BsmtFinType2 <chr>, Foundation <chr>,
## #   RoofStyle <chr>, Exterior1st <chr>, LandSlope <chr>
```

Response to Question 3

`GarageCars` corresponds to a quantitative variable given that it quantitatively measures the number of cars that can fit in a garage.

Conversely, `HouseStyle` is a categorical variable given how it characterizes the kinds of houses that quantitatively has a given number of cars in their garages.

Response to Question 4

Ensuring that this tibble follows the analytic aspects of the prompt before the presentation aspects, I pipe `house` with a `filter()` function that seeks to only induce findings that effectively adheres to the said conditions of the prompt. Given how the prompt does not ask these new findings to be reiterated into a R object and to simply display it, I simply then house this piped function in a `print` function.

```r
print(
  house %>%
  filter(
    YearBuilt > median(YearBuilt, na.rm = TRUE),
    SalePrice < mean(SalePrice, na.rm = TRUE),
    HouseStyle == "2Story"
  )
)
```

```
## # A tibble: 66 x 13
##    SalePrice BsmtFinSF1 GarageCars YearBuilt OpenPorchSF WoodDeckSF YearRemodAdd
##        <dbl>      <dbl>      <dbl>     <dbl>       <dbl>      <dbl>        <dbl>
##  1    172500        649          2      1999           0        115         2000
##  2    185000          0          2      1998          94          0         1998
##  3    174000          0          2      2005          38        100         2005
##  4    178000          0          2      1985          46        192         1985
##  5    176000        419          2      1999          32          0         1999
##  6    163990          0          2      2005          24          0         2006
##  7    130000          0          2      2004          40          0         2006
##  8    130000          0          2      2004          40          0         2004
##  9    148500        566          2      1976           0         87         1976
## 10    161750        378          1      1977          36          0         1977
## # i 56 more rows
## # i 6 more variables: HouseStyle <chr>, BsmtFinType2 <chr>, Foundation <chr>,
## #   RoofStyle <chr>, Exterior1st <chr>, LandSlope <chr>
```

Response to Question 5

Using the original `house` data set, I make a new `house_subset` object that is not only characterized by the old `house` variables being renamed to those outlined in the prompt, but also catalyzed by the `rename()`function that is housed in the piped chain that gives way for `house_subset`. I then pipe (%>%) the `rename()` function with a `mutuate()` function to not just facilitate the formation of a new factor version of `garage_cars` (that of which is executed via `factor()`), but also create the new column for it necessary for `house_subset`. I then pipe that with a `select()` function composed of the three variables at hand, followed by the `head(house_subset)` that comes to print out the first six rows of the data set.

```
house_subset <- house %>%
  rename(
    sale_price = SalePrice,
    garage_cars = GarageCars,
    year_built = YearBuilt
  ) %>%
  mutate(garage_cars = factor(garage_cars)) %>%
  select(sale_price, garage_cars, year_built)
head(house_subset)
```

```
## # A tibble: 6 x 3
##   sale_price garage_cars year_built
##        <dbl> <fct>            <dbl>
## 1     223500 2                 2001
## 2     140000 3                 1915
## 3     250000 3                 2000
## 4     200000 2                 1973
## 5     129500 1                 1965
## 6     345000 3                 2005
```

Response to Question 6

Writing the function `exchange` first necessitates the usage of `function()`, with `usd` merely being a relevant describer of the argument name in the `exchange` function, given that other relevant names (like "US Dollar") bear no impact on the output. Having finalized the function of the `exchange` function, I then used it on the `house_subset` object (alongside the `$` operator that enables access to the essential `sales_price` variable) to get the converted sales values for the first five values of that set, that of which is simply obtained by writing "1:5" in square brackets.

```
exchange <- function(usd) {
  return(usd * 1.45)
}
exchange(house_subset$sale_price)[1:5]
```

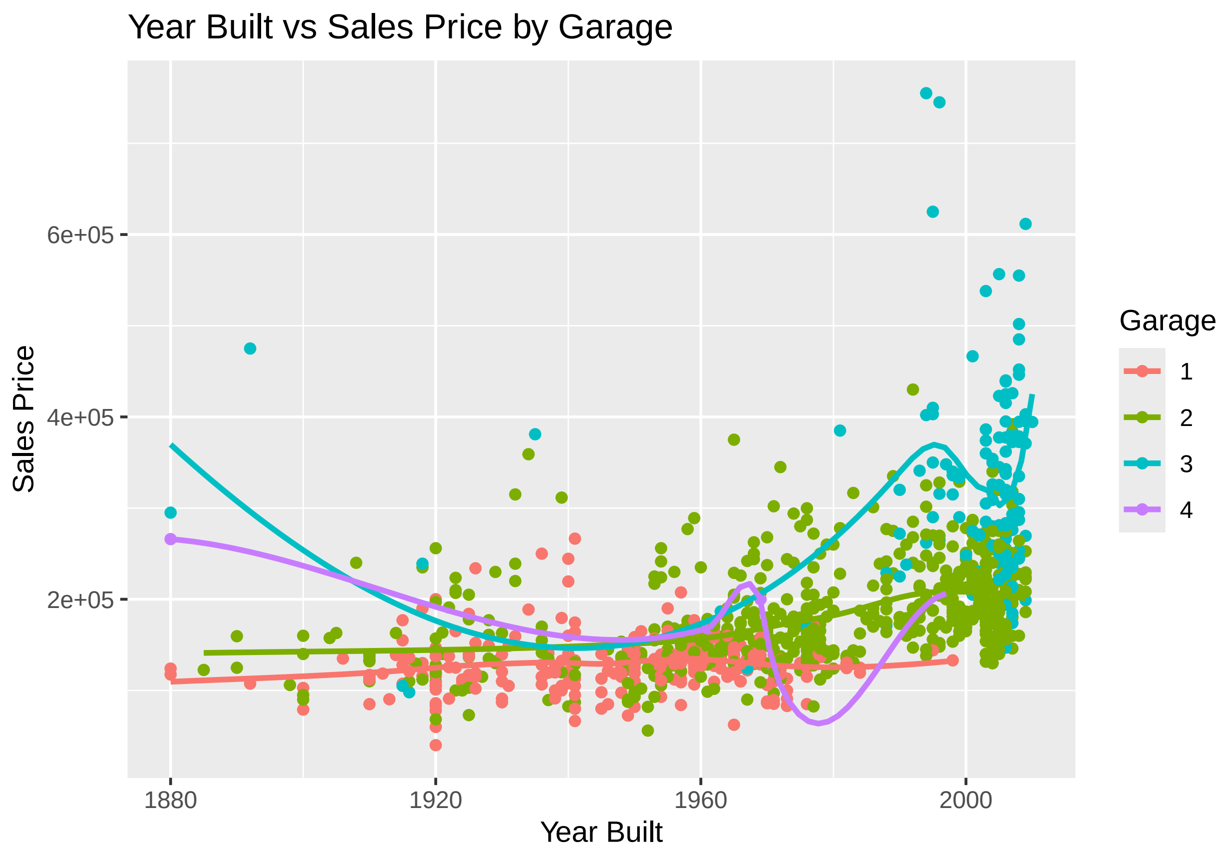
```
## [1] 324075 203000 362500 290000 187775
```

Response to Question 7

Generally speaking, it seems as though as time passes, more two- and three-car garage houses are comprising of the costliest houses. This additionally substantiated by the trend lines, with that also showing a re-emergence of four-car garages in the market from a price perspective. I then write the following code to produce the graph at hand.

```
ggplot(house_subset, aes(x = year_built, y = sale_price, color = garage_cars)) +
  geom_point()+ #for the scatter plot layer
  geom_smooth(se=FALSE)+ #for trend line layer, with (se=FALSE) preventing default logical intervals
# from showing
  labs( #for adding labels in place of graph's title, x-axis label, and y-axis label
    title = "Year Built vs Sales Price by Garage",
    x = "Year Built",
    y = "Sales Price",
    color = "Garage" #for color scheming to correspond with `garage_cars` variable
  )
```

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



Part 2

Response to Question 1

For the data that I am going to be using, I went to Kaggle and downloaded the `indicator-3-4-0-3-1.csv` dataset from the link `http://stateoftheenvironment.des.qld.gov.au/2017/datasets/indicator-3-4-0-3-1.csv`. In this data set, I analyze the greenhouse gas emissions (in millions of tons of carbon dioxide) of all eight of Queensland, AU's transport sectors from 1990-2016. What this made me question, from a research perspective, is what sector had the highest average emissions figures throughout that 26-year period. I figured this question was worth asking given how it can provide insight into how this state's emissions compares to other states in Australia.

As for visualizing this, I figured a bar chart showing the average emissions per sector (over a 26-year period) is an adequate choice. Here is the following code I wrote to execute my visualization:

```r
#Reading in data uploaded into a new object.
QLDemissions <- read_csv("indicator-3-4-0-3-1.csv")
```

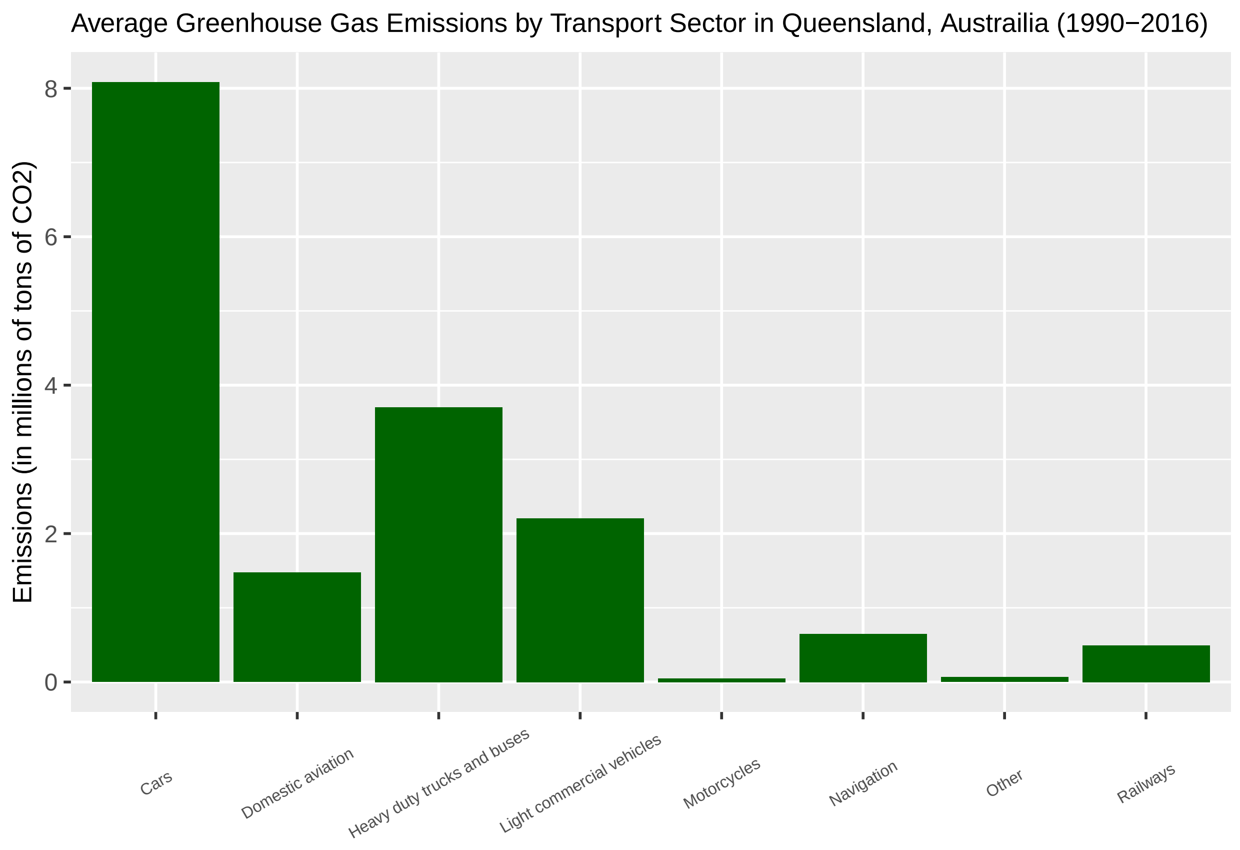
```
## Rows: 9 Columns: 28
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Category
## dbl (27): 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#Converting from wide to long format to enable easier mean calculations
QLDemissionsLong <- QLDemissions %>%
  pivot_longer(cols = 2:28,
               names_to = "Year",
               values_to = "Emissions")

#Calculating average emissions per sector
QLDavg_emissions <- QLDemissionsLong %>%
  group_by(Category) %>%
  summarise(average_emissions = mean(Emissions, na.rm = TRUE))
RevisedQLDavg_emissions <- filter(QLDavg_emissions, average_emissions < 16)

#Creating the bar plot
g <- ggplot(data = RevisedQLDavg_emissions,
       aes(x = Category, y = average_emissions))
g + geom_col(fill = "darkgreen", size = 10) + #here, geom_col was used instead of geom_bar
#since the former ensures the heights of the bars represent the values in the data more accordingly,
#given how the latter uses stat_count() by default when unnecessary
  labs(subtitle =
"Average Greenhouse Gas Emissions by Transport Sector in Queensland, Austrailia (1990-2016)",
    y = "Emissions (in millions of tons of CO2)"
  ) +
  theme(axis.text.x=element_text(size=6,
                                 angle = 30,
                                 vjust=.5),
        axis.title.y = element_text(size=10),
        axis.title.x = element_text(size=0), #here, the term "Transport Sector" appears by default,
#and I figured this line would get rid of it simply due to redundancy
        plot.subtitle = element_text(size=10))
```

```
## 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.
```


Average Greenhouse Gas Emissions by Transport Sector in Queensland, Austrailia (1990−2016)
Emissions (in millions of tons of CO2)
Cars
Domestic aviation
Heavy duty trucks and buses
Light commercial vehicles
Motorcycles
Navigation
Other
Railways