DATA 607 Project 2

Samuel C

2025-03-09

Overview

In this project I have chosen three datasets provided from my classmates to tidy and clean. These datasets involve sales data, weather data across different cities, and lastly emissions data across different countries. Not only will these datasets be tidied, I will also perform a bit of exploratory data analysis in order to consider possible relationships among the data in each dataset.

Getting Started

First, we must load the packages and data we will use. I have stored the data on my github across three separate .csv files.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## 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
```

```
untidy_emissions <- read.csv("https://raw.githubusercontent.com/scrummett/DATA607/refs/heads/main/Total
untidy_sales <- read.csv("https://raw.githubusercontent.com/scrummett/DATA607/refs/heads/main/salesdata
untidy_weather <- read.csv("https://raw.githubusercontent.com/scrummett/DATA607/refs/heads/main/weather
```

Now with the data loaded, I will begin tidying and EDA from the easiest to the most intensive.

Sales Data

```
head(untidy_sales)
```

```
##   Product.Name Region Jan.Sales Feb.Sales Mar.Sales Apr.Sales May.Sales
## 1    Product A  North       100       110       120       130       140
## 2    Product A  South       200       210       220       230       240
## 3    Product A   East       300       310       320       330       340
## 4    Product B  North       150       160       170       180       190
## 5    Product B  South       250       260       270       280       290
```

```
## 6    Product B   East       350       360       370       380       390
##   Jun.Sales
## 1       150
## 2       250
## 3       350
## 4       200
## 5       300
## 6       400
```

This dataset has sales per month separated out across different columns, however a clean version of this dataset would have "months" be a column itself, and total sales figures being a separate column as well.

```r
untidy_sales <- untidy_sales |>
  pivot_longer(
    cols = ends_with(".Sales"),
    names_to = "Month",
    values_to = "Sales")
head(untidy_sales)
```

```
## # A tibble: 6 x 4
##   Product.Name Region Month     Sales
##   <chr>        <chr>  <chr>     <int>
## 1 Product A    North  Jan.Sales   100
## 2 Product A    North  Feb.Sales   110
## 3 Product A    North  Mar.Sales   120
## 4 Product A    North  Apr.Sales   130
## 5 Product A    North  May.Sales   140
## 6 Product A    North  Jun.Sales   150
```

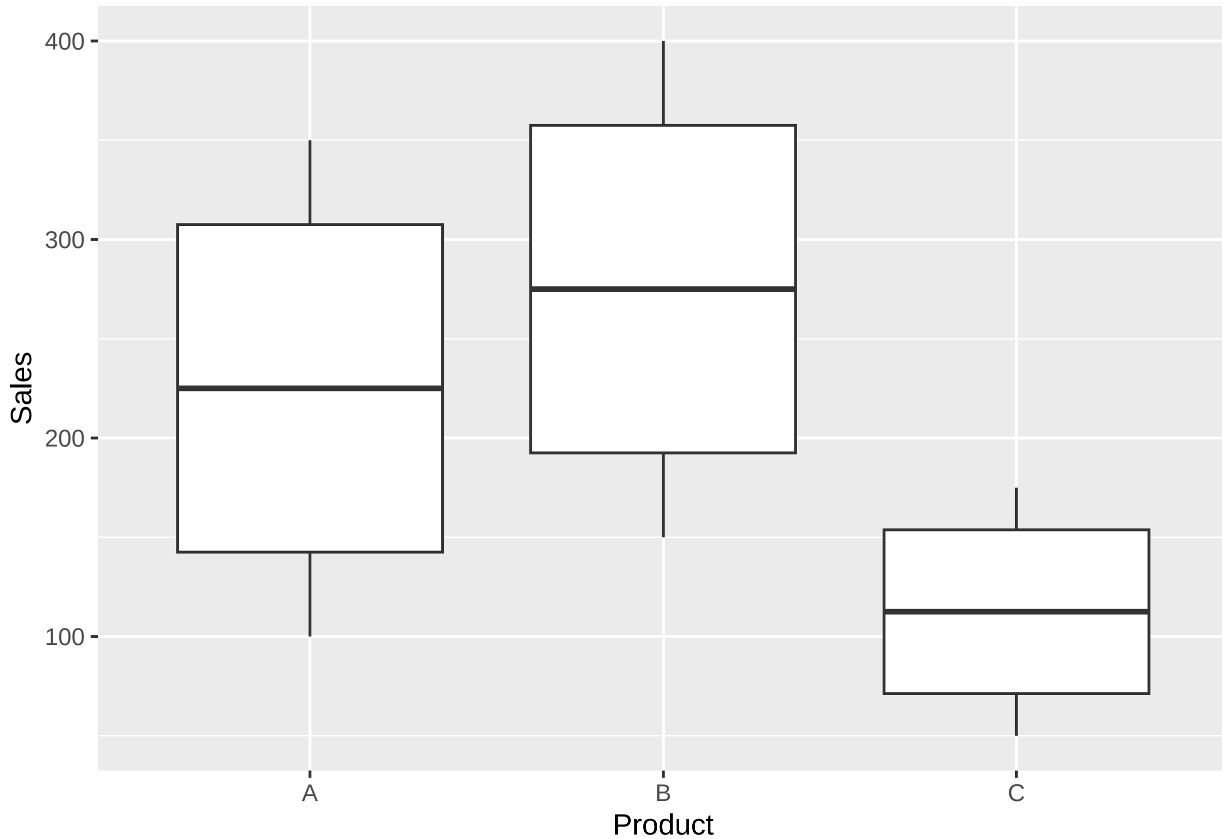
We now have a "tidy" dataset, however the data in the columns can be cleaned up to avoid redundancy.

```r
untidy_sales <- untidy_sales |>
  mutate(Month = str_remove(Month, ".Sales"),
         Product.Name = str_remove(Product.Name, "Product "))
tidy_sales <- untidy_sales |>
  rename("Product" = "Product.Name")
head(tidy_sales)
```

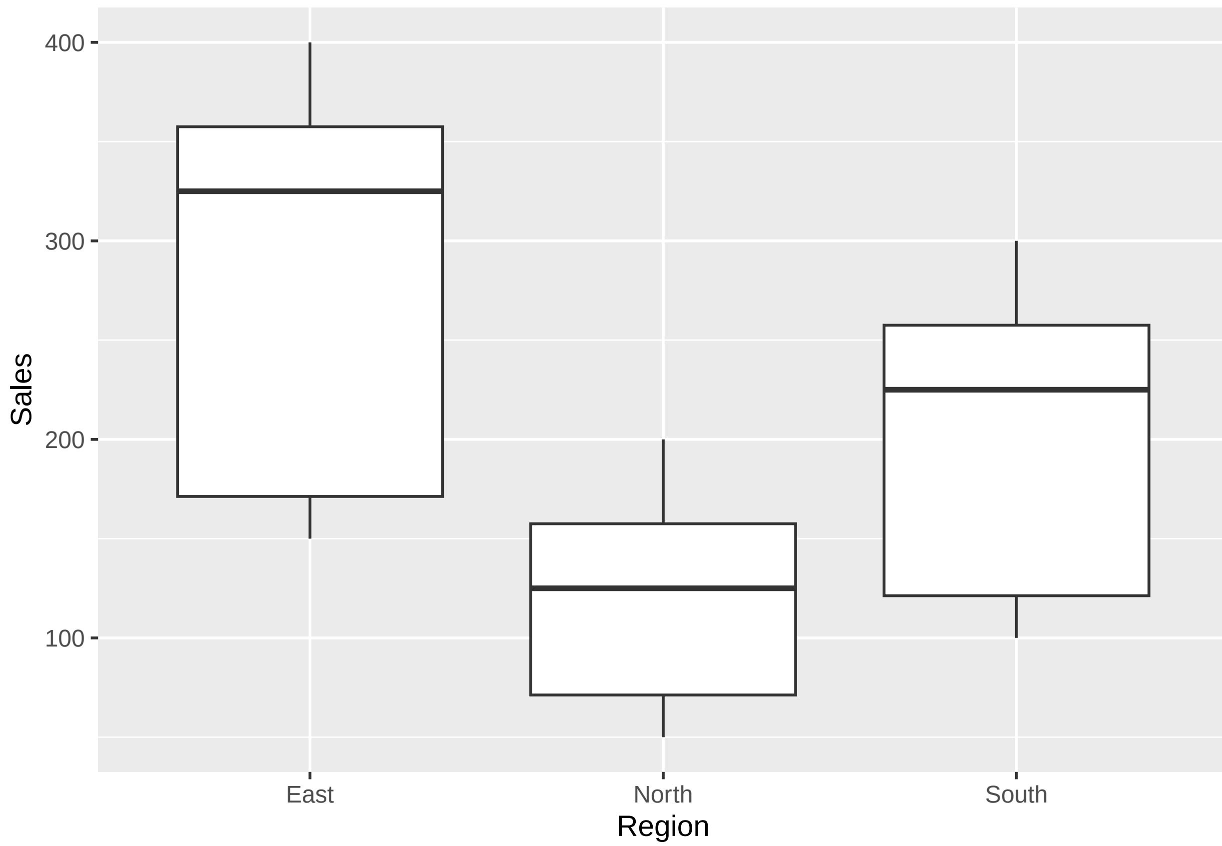
```
## # A tibble: 6 x 4
##   Product Region Month Sales
##   <chr>   <chr>  <chr> <int>
## 1 A       North  Jan     100
## 2 A       North  Feb     110
## 3 A       North  Mar     120
## 4 A       North  Apr     130
## 5 A       North  May     140
## 6 A       North  Jun     150
```

After cleaning up observations and changing the title of a column, we now have a tidy dataset of sales data.

```r
tidy_sales |>
  ggplot(aes(x = Product, y = Sales)) +
  geom_boxplot()
```



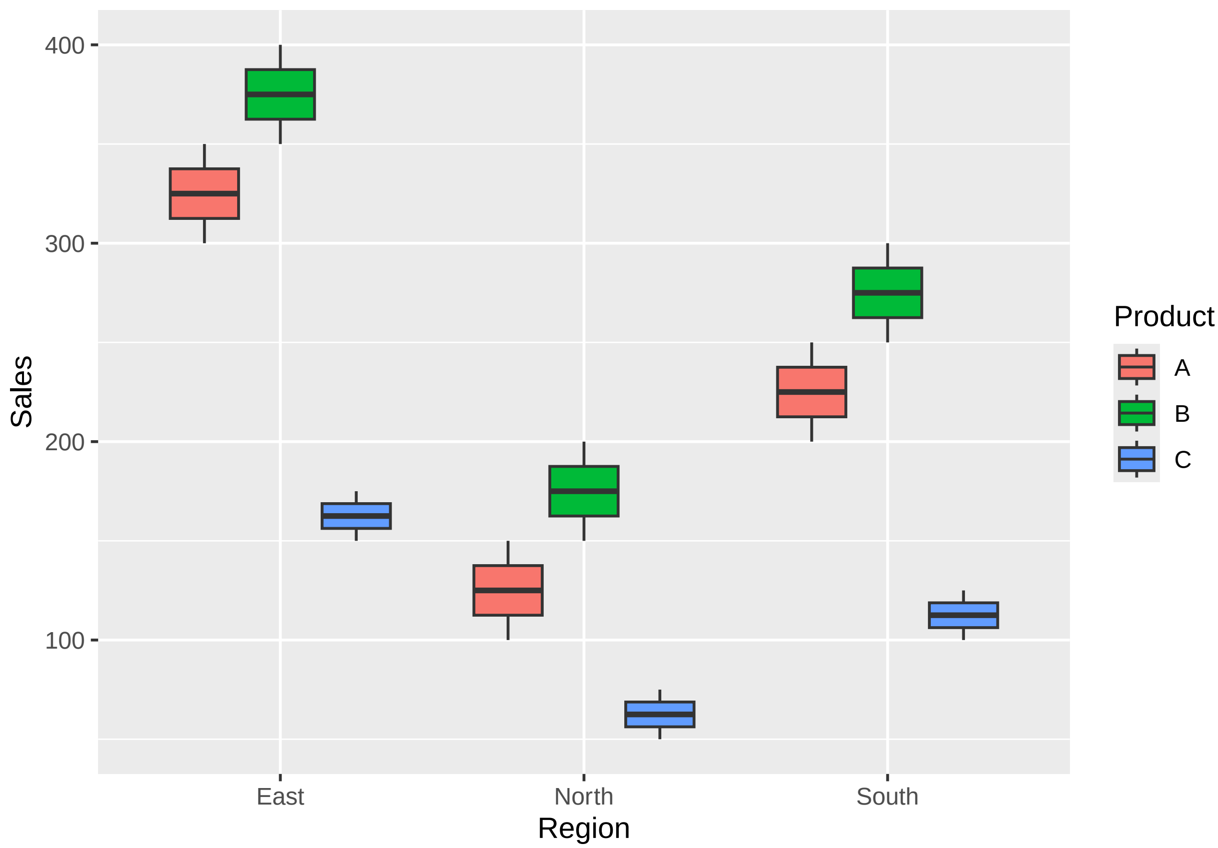

```r
tidy_sales |>
  ggplot(aes(x = Region, y = Sales)) +
  geom_boxplot()
```



Here we have graphs showing total sales by product and then by region. The graphs show that while products A and B sell similar albeit different amounts, sales of product C lag behind severely. Additionally, the North has far fewer sales than both the South and the East.

One question posed in our discussion forum asked about sales performance by product across regions, which can be seen in the following graph.

```
tidy_sales |>
  ggplot(aes(x = Region, y = Sales, fill = Product)) +
  geom_boxplot()
```



From this we can see that product B sells the most across every region, and that product C sells the least.

Weather

While our sales data only had months and sales to elongate, the dataset for weather has one more.

```
head(untidy_weather)
```

```
##          City Temp_Jan Temp_Feb Temp_Mar Humid_Jan Humid_Feb Humid_Mar
## 1    New York     32°F     35°F     42°F       75%       72%       68%
## 2 Los Angeles     58°F     60°F     65°F       65%       63%       60%
## 3     Chicago     28°F     30°F     40°F       80%       78%       75%
```

Again, months are spread across the data as separate variables, however we also have temperature and humidity to separate out into individual columns as well.

```
untidy_weather_left <- untidy_weather |>
  pivot_longer(
    cols = starts_with("Temp_"),
    names_to = c("Month"),
    names_prefix = "Temp_",
    values_to = "Temp_F")
untidy_weather_right <- untidy_weather |>
  pivot_longer(
    cols = starts_with("Humid_"),
    names_to = "Month",
```

```r
    names_prefix = "Humid_",
    values_to = "Humidity_Percent"
  )
head(untidy_weather_left)
```

```
## # A tibble: 6 x 6
##   City        Humid_Jan Humid_Feb Humid_Mar Month Temp_F
##   <chr>       <chr>     <chr>     <chr>     <chr> <chr>
## 1 New York    75%       72%       68%       Jan   32°F
## 2 New York    75%       72%       68%       Feb   35°F
## 3 New York    75%       72%       68%       Mar   42°F
## 4 Los Angeles 65%       63%       60%       Jan   58°F
## 5 Los Angeles 65%       63%       60%       Feb   60°F
## 6 Los Angeles 65%       63%       60%       Mar   65°F
```

```r
head(untidy_weather_right)
```

```
## # A tibble: 6 x 6
##   City        Temp_Jan Temp_Feb Temp_Mar Month Humidity_Percent
##   <chr>       <chr>    <chr>    <chr>    <chr> <chr>
## 1 New York    32°F     35°F     42°F     Jan   75%
## 2 New York    32°F     35°F     42°F     Feb   72%
## 3 New York    32°F     35°F     42°F     Mar   68%
## 4 Los Angeles 58°F     60°F     65°F     Jan   65%
## 5 Los Angeles 58°F     60°F     65°F     Feb   63%
## 6 Los Angeles 58°F     60°F     65°F     Mar   60%
```

```r
untidy_weather <- left_join(untidy_weather_left, untidy_weather_right, by = c("City", "Month"))
head(untidy_weather)
```

```
## # A tibble: 6 x 10
##   City     Humid_Jan Humid_Feb Humid_Mar Month Temp_F Temp_Jan Temp_Feb Temp_Mar
##   <chr>    <chr>     <chr>     <chr>     <chr> <chr>  <chr>    <chr>    <chr>
## 1 New York 75%       72%       68%       Jan   32°F   32°F     35°F     42°F
## 2 New York 75%       72%       68%       Feb   35°F   32°F     35°F     42°F
## 3 New York 75%       72%       68%       Mar   42°F   32°F     35°F     42°F
## 4 Los Ang~ 65%       63%       60%       Jan   58°F   58°F     60°F     65°F
## 5 Los Ang~ 65%       63%       60%       Feb   60°F   58°F     60°F     65°F
## 6 Los Ang~ 65%       63%       60%       Mar   65°F   58°F     60°F     65°F
## # i 1 more variable: Humidity_Percent <chr>
```

We now have our individual columns for Month, Temperature and Humidity, however we must trim the fat and get rid of each column that has a single months values.

```r
untidy_weather <- untidy_weather |>
  select("City",
         "Month",
         "Temp_F",
         "Humidity_Percent")
head(untidy_weather)
```

```
## # A tibble: 6 x 4
##   City        Month Temp_F Humidity_Percent
##   <chr>       <chr> <chr>  <chr>
## 1 New York    Jan   32°F   75%
## 2 New York    Feb   35°F   72%
## 3 New York    Mar   42°F   68%
## 4 Los Angeles Jan   58°F   65%
## 5 Los Angeles Feb   60°F   63%
## 6 Los Angeles Mar   65°F   60%
```

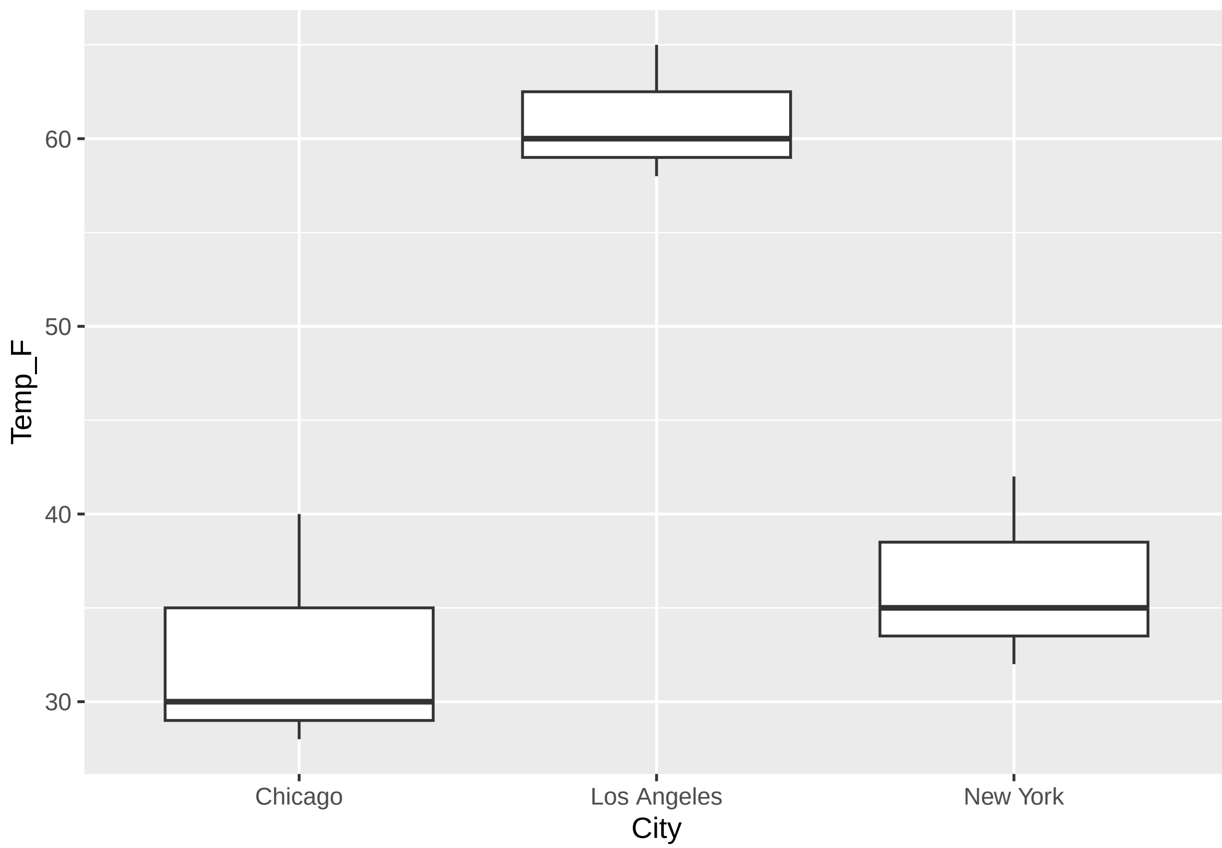
With this, we have our data tidy, but to make sure we can examine with EDA we must change Temperature and Humidity from character values to integers.

```
tidy_weather <- untidy_weather |>
  mutate(Temp_F = parse_number(Temp_F),
         Humidity_Percent = parse_number(Humidity_Percent))
head(tidy_weather)
```

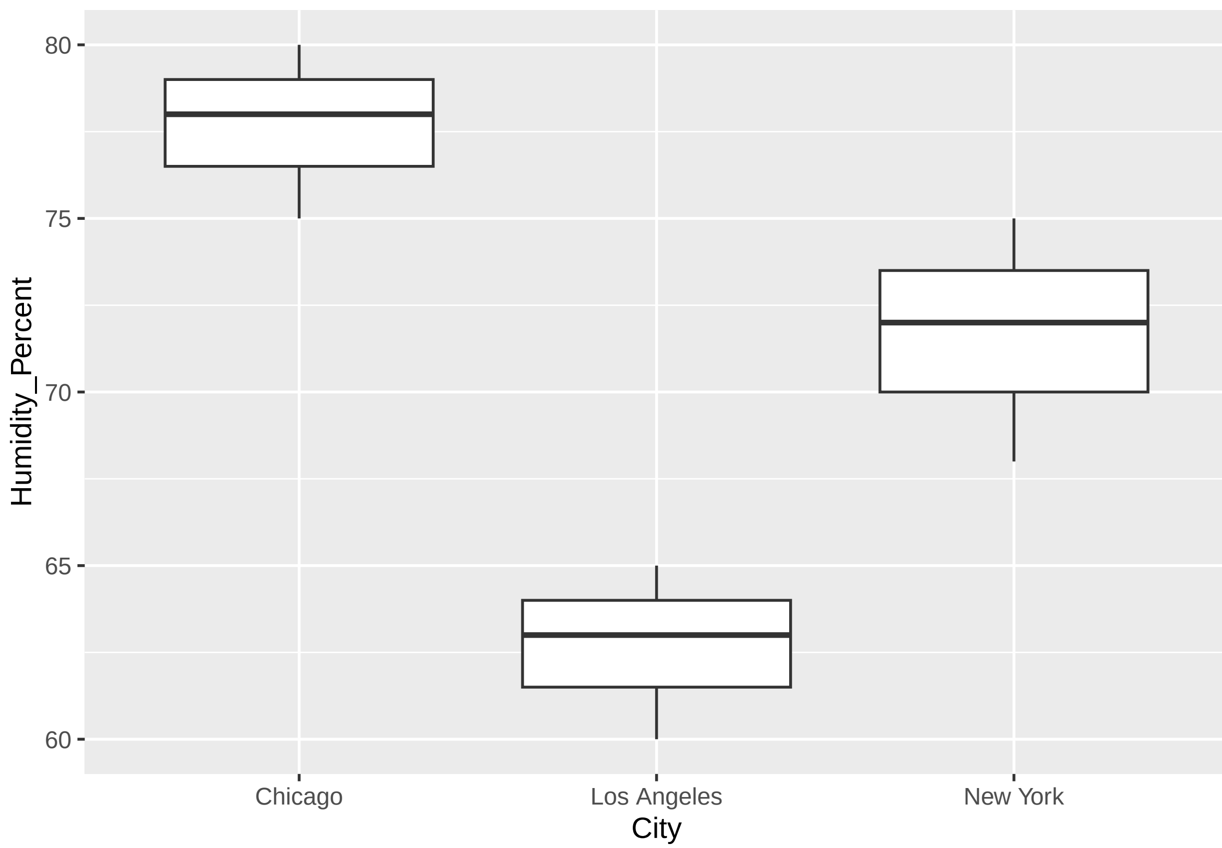
```
## # A tibble: 6 x 4
##   City        Month Temp_F Humidity_Percent
##   <chr>       <chr>  <dbl>            <dbl>
## 1 New York    Jan       32               75
## 2 New York    Feb       35               72
## 3 New York    Mar       42               68
## 4 Los Angeles Jan       58               65
## 5 Los Angeles Feb       60               63
## 6 Los Angeles Mar       65               60
```

Now we can begin EDA.

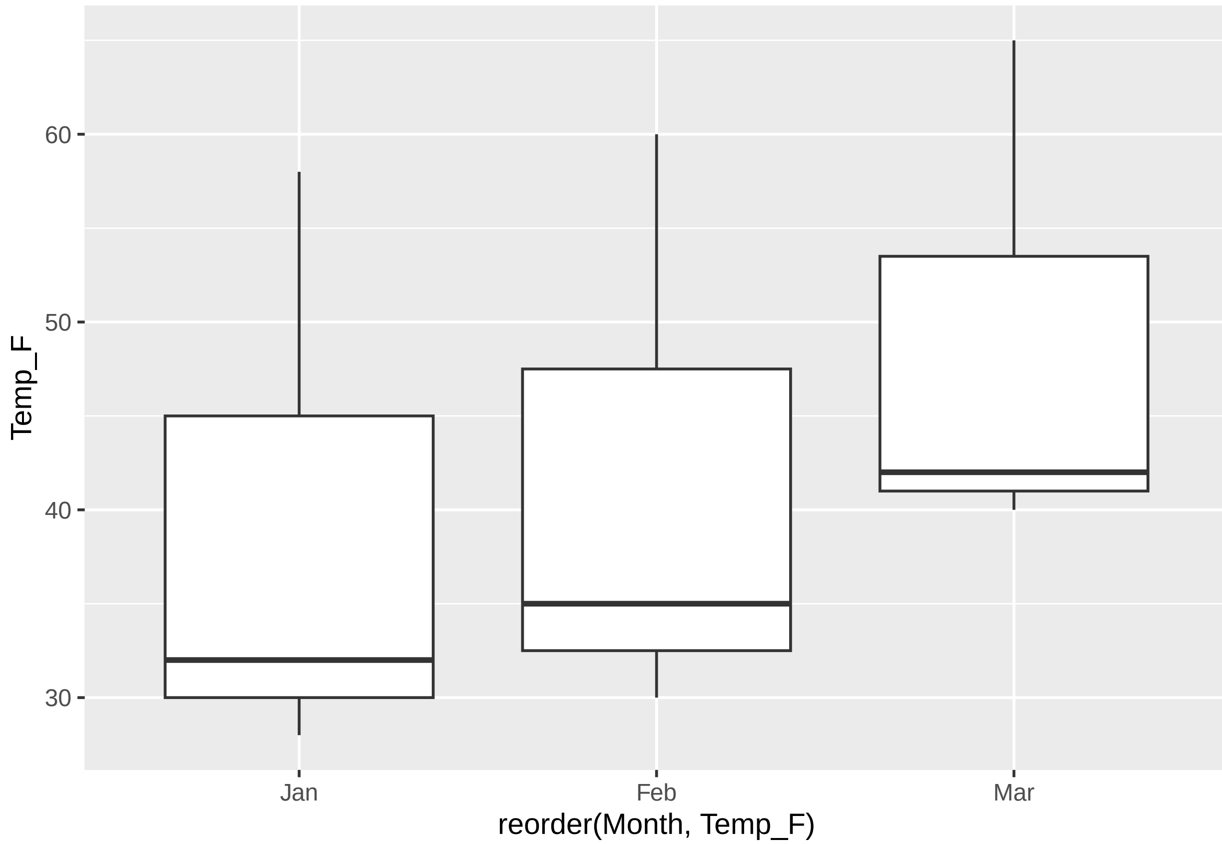
```
tidy_weather |>
  ggplot(aes(x = City, y = Temp_F)) +
  geom_boxplot()
```



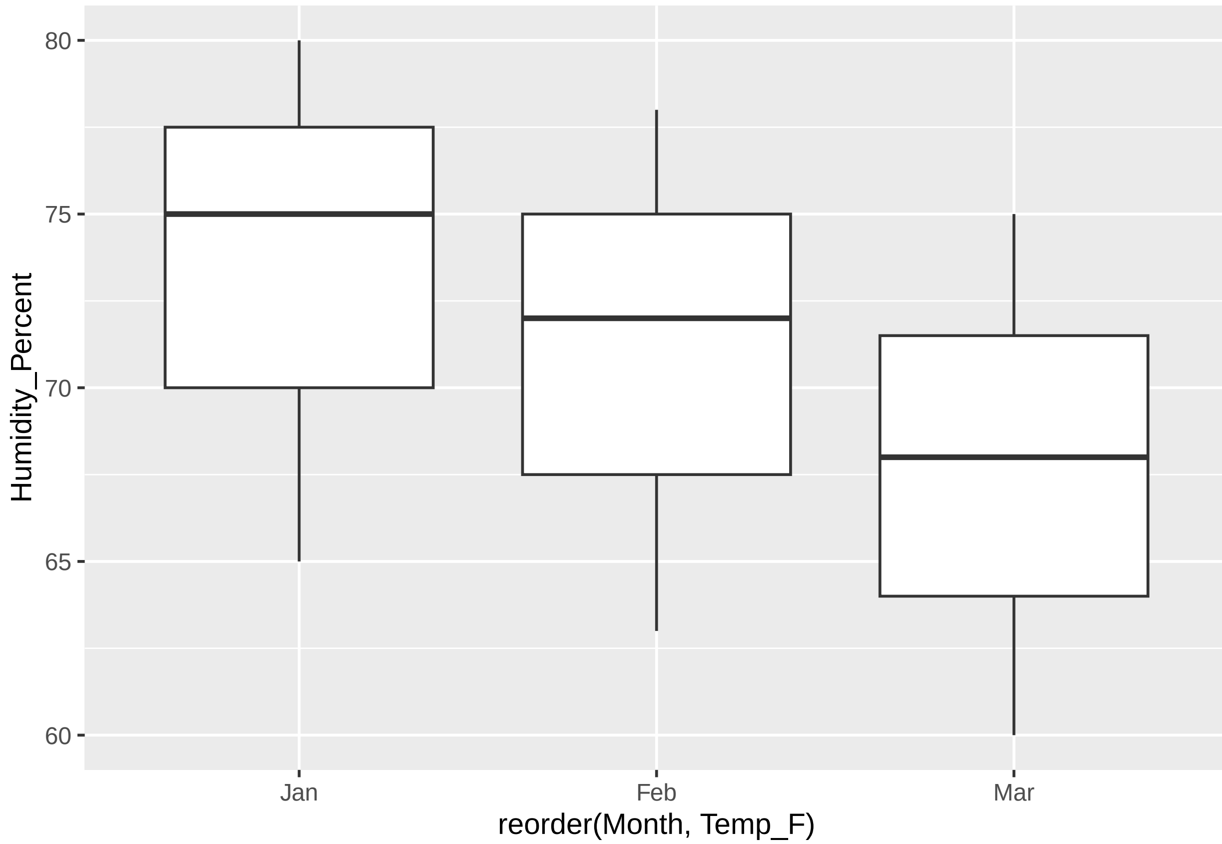

```r
tidy_weather |>
  ggplot(aes(x = City, y = Humidity_Percent)) +
  geom_boxplot()
```




```r
tidy_weather |>
  ggplot(aes(x = reorder(Month, Temp_F), y = Temp_F)) +
  geom_boxplot()
```



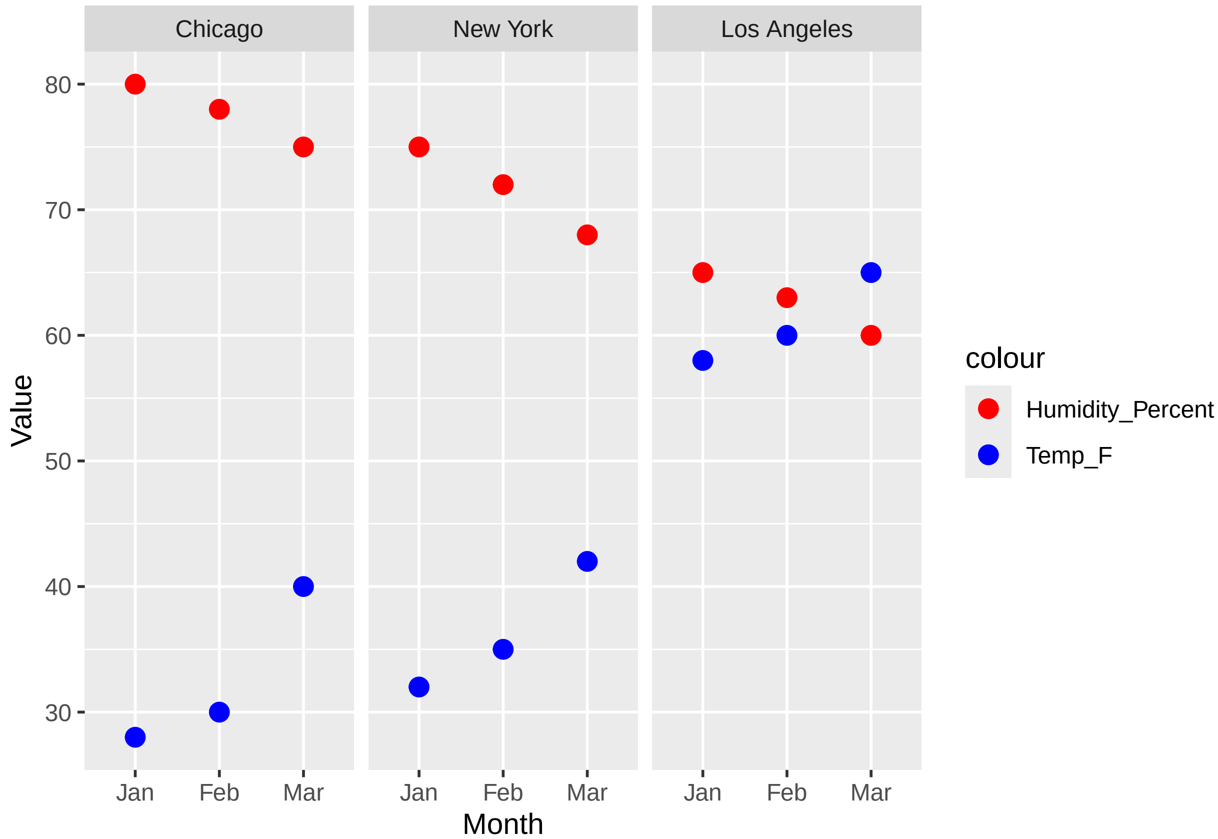

```
tidy_weather |>
  ggplot(aes(x = reorder(Month, Temp_F), y = Humidity_Percent)) +
  geom_boxplot()
```



Here we have graphs showing temperature and humidity broken down by either city or month. These graphs show that Chicago was the coldest across sampled months while LA was the hottest, while the opposite is true regarding humidity. These graphs also show that temperature increases as months go on while humidity decreases.

We can also look at how temperature and humidity fluctuate over these months across different cities with the following graph.

```r
tidy_weather |>
  ggplot(aes(x = reorder(Month, Temp_F))) +
    geom_point(aes(y = Temp_F, color = "Temp_F"), size = 3) +
    geom_point(aes(y = Humidity_Percent, color = "Humidity_Percent"), size = 3) +
    facet_wrap(~ factor(City, levels = c("Chicago", "New York", "Los Angeles"))) +
    labs(x = "Month", y = "Value") +
    scale_color_manual(values = c("Temp_F" = "blue", "Humidity_Percent" = "red"))
```



Here we can see that across all cities, as the months continue from January to March, temperature increases and humidity decreases.

Emissions

This is the largest dataset of the three, and needs the most work tidying and cleaning before EDA.

```
head(untidy_emissions)
```

```
##          Area             Item                       Element       Unit
## 1 Afghanistan    Crop Residues           Direct emissions (N2O) kilotonnes
## 2 Afghanistan    Crop Residues         Indirect emissions (N2O) kilotonnes
## 3 Afghanistan    Crop Residues                  Emissions (N2O) kilotonnes
## 4 Afghanistan    Crop Residues Emissions (CO2eq) from N2O (AR5) kilotonnes
## 5 Afghanistan    Crop Residues          Emissions (CO2eq) (AR5) kilotonnes
## 6 Afghanistan Rice Cultivation                  Emissions (CH4) kilotonnes
##     X2000    X2001    X2002    X2003    X2004    X2005    X2006    X2007
## 1   0.520   0.5267   0.8200   0.9988   0.8225   1.1821   1.0277   1.2426
## 2   0.117   0.1185   0.1845   0.2247   0.1851   0.2660   0.2312   0.2796
## 3   0.637   0.6452   1.0045   1.2235   1.0075   1.4481   1.2589   1.5222
## 4 168.807 170.9884 266.1975 324.2195 266.9995 383.7498 333.6093 403.3749
## 5 168.807 170.9884 266.1975 324.2195 266.9995 383.7498 333.6093 403.3749
## 6  18.200  16.9400  18.9000  20.3000  27.3000  22.4000  22.4000  23.8000
##      X2008    X2009    X2010    X2011    X2012    X2013    X2014    X2015
## 1   0.8869   1.3920   1.2742   1.0321   1.3726   1.4018   1.4584   1.2424
## 2   0.1996   0.3132   0.2867   0.2322   0.3088   0.3154   0.3281   0.2795
## 3   1.0865   1.7051   1.5609   1.2643   1.6815   1.7173   1.7865   1.5220
```

```
## 4 287.9099 451.8647 413.6467 335.0379 445.5958 455.0727 473.4174 403.3181
## 5 287.9099 451.8647 413.6467 335.0379 445.5958 455.0727 473.4174 403.3181
## 6  26.6000  28.0000  29.1200  29.4000  28.7000  28.7000  30.8000  22.9600
##      X2016    X2017    X2018    X2019    X2020
## 1   1.1940   1.0617   0.8988   1.2176   1.3170
## 2   0.2687   0.2389   0.2022   0.2740   0.2963
## 3   1.4627   1.3005   1.1011   1.4916   1.6133
## 4 387.6130 344.6447 291.7838 395.2689 427.5284
## 5 387.6130 344.6447 291.7838 395.2689 427.5284
## 6  16.6600  15.3233  16.4555  17.8542  20.6577
```

While "Item" and "Element" are extended across a "long" format, years are given their own columns, so we can tidy that first. We can also get rid of "Unit" as every "amt" is measured in kilotons and nothing else.

```
untidy_emissions <- untidy_emissions |>
  pivot_longer(
    cols = starts_with("X"),
    names_to = "Year",
    values_to = "Amt_kt"
  )
untidy_emissions <- untidy_emissions |>
  select(!Unit)
head(untidy_emissions)
```

```
## # A tibble: 6 x 5
##   Area        Item           Element                 Year  Amt_kt
##   <chr>       <chr>          <chr>                   <chr>  <dbl>
## 1 Afghanistan Crop Residues Direct emissions (N2O) X2000  0.52
## 2 Afghanistan Crop Residues Direct emissions (N2O) X2001  0.527
## 3 Afghanistan Crop Residues Direct emissions (N2O) X2002  0.82
## 4 Afghanistan Crop Residues Direct emissions (N2O) X2003  0.999
## 5 Afghanistan Crop Residues Direct emissions (N2O) X2004  0.822
## 6 Afghanistan Crop Residues Direct emissions (N2O) X2005  1.18
```

Now that our data is tidy and in a "long" format, we can clean it up for EDA.

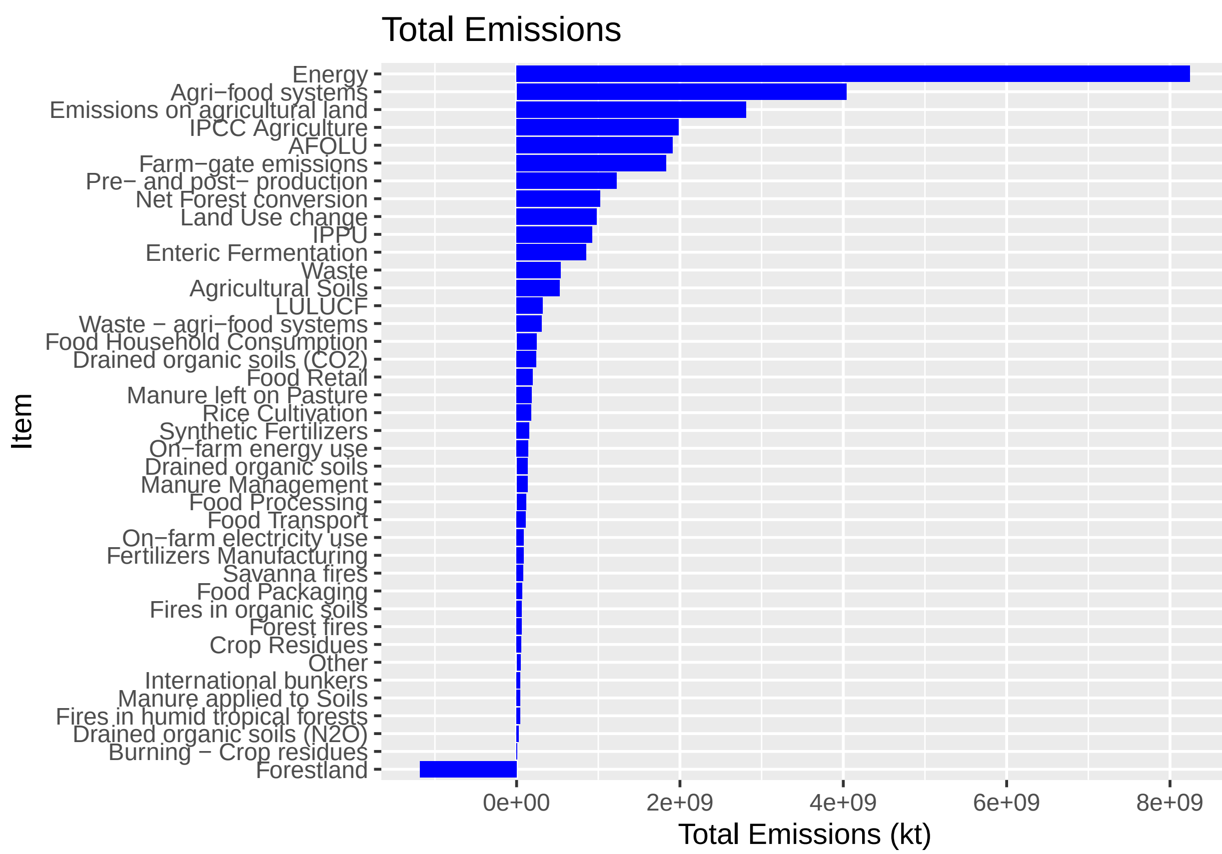
```
tidy_emissions <- untidy_emissions |>
  mutate(Element = str_remove(Element, regex("emissions", ignore_case = TRUE)),
         Element = str_remove(Element, " \\(AR5\\)"),
         Element = str_remove_all(Element, " \\(|\\)"),
         Year = str_remove(Year, "X"))
head(tidy_emissions)
```

```
## # A tibble: 6 x 5
##   Area        Item          Element    Year  Amt_kt
##   <chr>       <chr>         <chr>      <chr>  <dbl>
## 1 Afghanistan Crop Residues Direct N2O 2000   0.52
## 2 Afghanistan Crop Residues Direct N2O 2001   0.527
## 3 Afghanistan Crop Residues Direct N2O 2002   0.82
## 4 Afghanistan Crop Residues Direct N2O 2003   0.999
## 5 Afghanistan Crop Residues Direct N2O 2004   0.822
## 6 Afghanistan Crop Residues Direct N2O 2005   1.18
```

With the data tidied and observations cleaned, we can now begin EDA.

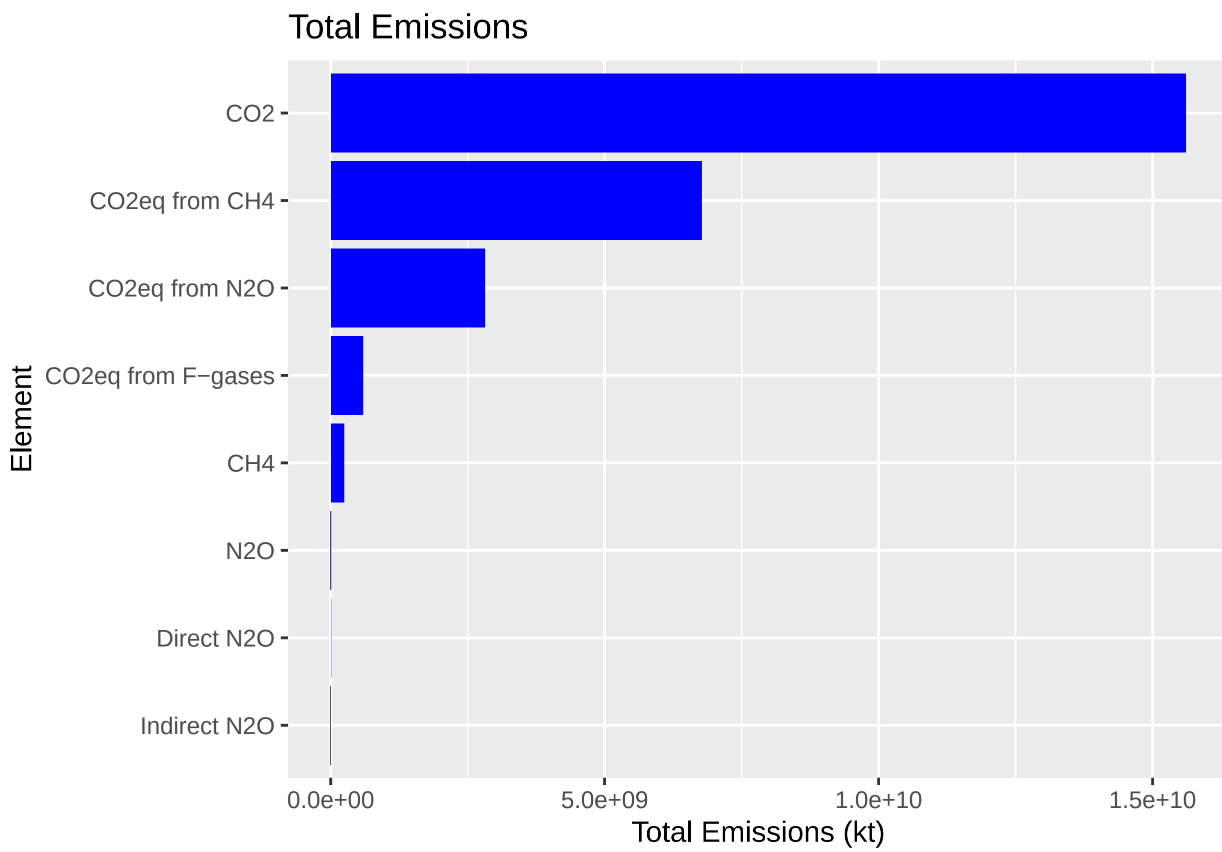
While my classmate suggested looking at overall total emissions per country for each year, and while I do think that would be insightful, processing and presenting that data has proved difficult. Therefore I will look at another suggested potential analysis, that of overall total emissions across source. Additionally, we can look at which element being polluted is the greatest.

```
tidy_emissions |>
  filter(Item != "All sectors with LULUCF" &
           Item != "All sectors without LULUCF") |>
  group_by(Item) |>
  summarise(Total_Emissions = sum(Amt_kt, na.rm = TRUE), .groups = "drop") |>
  ggplot(aes(x = fct_reorder(Item, Total_Emissions, .desc = FALSE), y = Total_Emissions)) +
  geom_col(fill = "blue") +
  labs(title = "Total Emissions",
       x = "Item",
       y = "Total Emissions (kt)") +
  coord_flip()
```

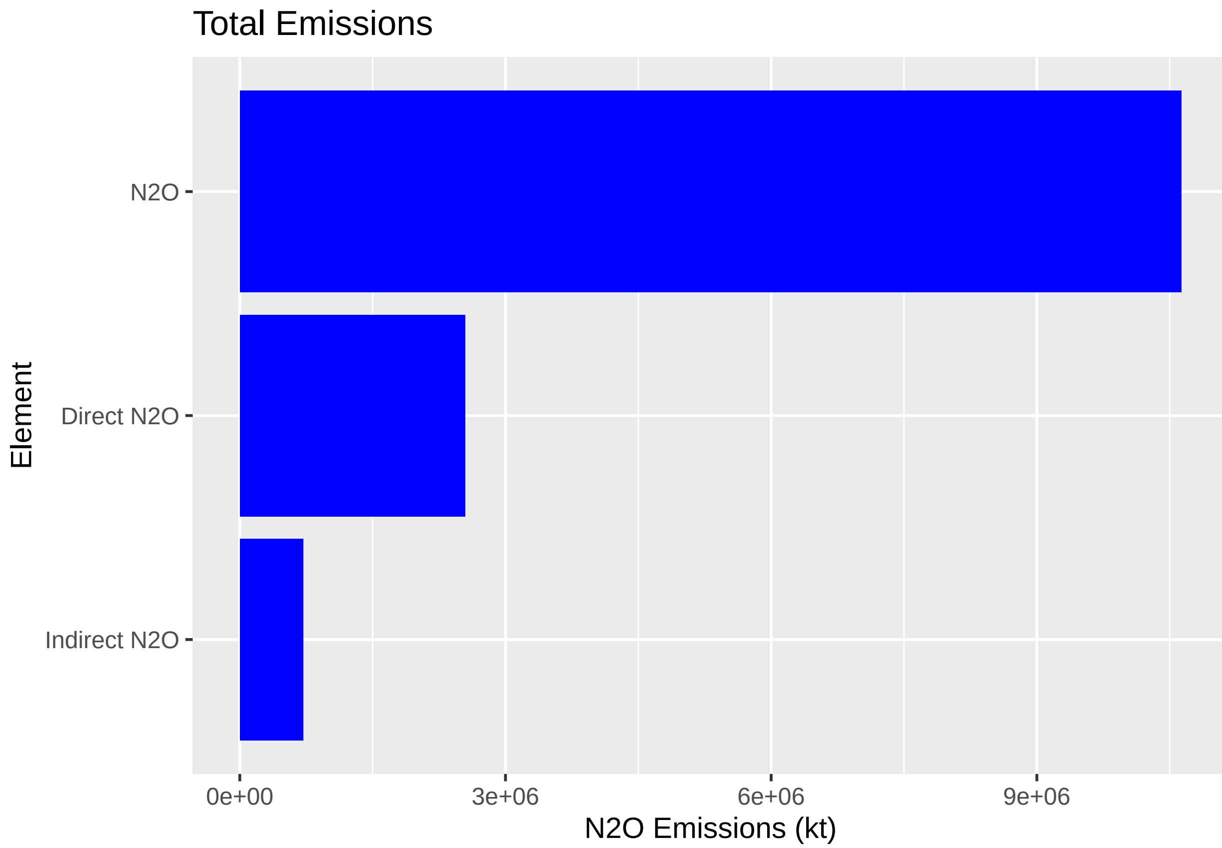



```
tidy_emissions |>
  filter(Element != "CO2eq") |>
  group_by(Element) |>
  summarise(Total_Emissions = sum(Amt_kt, na.rm = TRUE), .groups = "drop") |>
  ggplot(aes(x = fct_reorder(Element, Total_Emissions, .desc = FALSE), y = Total_Emissions)) +
  geom_col(fill = "blue") +
  labs(title = "Total Emissions",
```

```r
       x = "Element",
       y = "Total Emissions (kt)") +
  coord_flip()
```




```r
tidy_emissions |>
  filter(Element == "N2O" |
         Element == "Direct N2O" |
         Element == "Indirect N2O") |>
  group_by(Element) |>
  summarise(Total_Emissions = sum(Amt_kt, na.rm = TRUE), .groups = "drop") |>
  ggplot(aes(x = fct_reorder(Element, Total_Emissions, .desc = FALSE), y = Total_Emissions)) +
  geom_col(fill = "blue") +
  labs(title = "Total Emissions",
       x = "Element",
       y = "N2O Emissions (kt)") +
  coord_flip()
```



After filtering out variables accounting for all sectors with or without land use change, we find that energy is the greatest driver of emissions. Every item contributes to greater emissions except for one, that being forest land. Looking at the elements graph, we find that the single element emitted the most is CO2. N2O polutes the least, however we cannot get a good idea of how much compared to direct and indirect N2O, so we can break it down even further to find that N2O is producing a fraction of the other elements. Despite this, its equivalency in CO2 is the third largest on this list.

Conclusion

While all of these datasets had some stark differences, whether it be by the number of variables, or by the amount of sheer observations contained within, the process for tidying them was roughly the same across all three. Begin by identifying which columns can be addressed as the same variable, extend the data longer with said variable, and then transform the information into something to process. After that is when we see differences, as each dataset could be broken down differently. This leads to different forms of representation being better for one than another - I would not use columns as I did in emissions data to represent temperature and humidity in weather data. This was insightful on how managing and processing data can be very similar and very different!