Project 2: Data Transformation

Group Members: Alina Vikhnevich, Olivia Azevedo, Alyssa Gurkas, Musrat Jahan

2025-03-15

Contents

Loading libraries:

```r
library(kableExtra)
library(RSocrata)
library(tidyverse)
library(viridis)
library(readr)
library(readxl)
library(janitor)
library(lubridate)
library(ggplot2)
library(scales)
library(stringr)
library(forcats)
```

Introduction

Project Overview

The goal of this project is to tidy, transform, and analyze three different datasets using R, leveraging the `tidyverse`, `tidyr`, and `dplyr` packages. These datasets, originally in an untidy "wide" format, require cleaning, restructuring, and standardization before analysis can be performed.

By the end of this project, we will:

- Convert three untidy datasets into a structured format for analysis.
- Perform data wrangling using `tidyr` and `dplyr` to clean and reshape the data.
- Conduct exploratory data analysis (EDA) to uncover insights and trends.
- Document the transformation process and provide meaningful conclusions.

This project is a collaborative effort, and each dataset presents a unique challenge in terms of data cleaning, structuring, and interpretation. The final results will be published as an R Markdown report, demonstrating the power of data transformation techniques.

Overview of Datasets

Each dataset represents a different domain and requires a unique transformation approach. Below is a summary of the datasets used in this project:

1. **Dataset #1: Emissions Data**

- **Description**: This dataset provides information on pollutant emissions over multiple years. The data includes various emission sources and their impact over time.
- **Data Issues**: The dataset is in wide format, with emissions spread across multiple columns by year.
- **Transformation Steps**: We will convert it into a long format, making it easier to analyze trends over time.

2. **Dataset #2: New York State Department of Environmental Conservation's Application Review & Tracking System from 2020-2025 (DART)**

- **Description**: This dataset contains public about environment permits issued by New York State's Department of Environmental Conservation. This report explores water permits, regulated under the National Pollutant Discharge Elimination System (NPDES).
- **Data Issues**: The dataset is not normalized, and some entries are duplicated.
- **Transformation Steps**: We will standardize date formats, remove redundant data, and ensure consistency across permit records.

3. **Dataset #3: Cheese Nutritional Data**

- **Description**: This dataset provides nutritional information on various types of cheese.
- **Data Issues**: The dataset is structured as a wide table, making it difficult to compare across different cheese types.
- **Transformation Steps**: We will reshape the data into a long format, making it easier to compare nutritional values across different cheese varieties.

Relevance of These Datasets

Each dataset requires different data transformation techniques, making them ideal for practicing `tidyr` and `dplyr` functions. The common themes across these datasets include:

- Converting wide-format data into long format.
- Standardizing date and time fields.
- Handling missing values and duplicates.
- Preparing the data for downstream statistical analysis and visualization.

By applying tidy data principles, we ensure that each dataset is structured, organized, and ready for analysis. The insights gained from this project can be used for policy recommendations, water quality analysis, and compliance to the Clean Water Act.

Data Preparation and Cleaning

Emissions Data:

```
df <- read.csv("https://raw.githubusercontent.com/justin-2028/Total-Emissions-Per-Country-2000-2020/ref

colnames(df) <- gsub("^X", "", colnames(df))

print(head(df))
```

```
##          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
##      2000     2001     2002     2003     2004     2005     2006     2007
## 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
##       2008     2009     2010     2011     2012     2013     2014     2015
## 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
##       2016     2017     2018     2019     2020
## 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
```

```
unique(df$Unit)
```

```
## [1] "kilotonnes"
```

Make longer All the year columns were changed to one column under year. The dataset was made longer. The Unit column was deleted and I put the unit in parentheses in the column name for total emissions since there is just one unit and it is same for all the records. This is now a tidy dataset.

```
df_longer <- df |>
  pivot_longer(
    cols = starts_with("2"),
    names_to = "year",
    values_to = "total emissions (in kilotonnes)",
    values_drop_na = TRUE
  )
df_longer$Unit <- NULL
```

```r
print(head(df_longer))
```

```
## # A tibble: 6 x 5
##   Area        Item           Element                 year  total emissions (in k~1
##   <chr>       <chr>          <chr>                   <chr>                   <dbl>
## 1 Afghanistan Crop Residues Direct emissions (N2O) 2000                     0.52
## 2 Afghanistan Crop Residues Direct emissions (N2O) 2001                     0.527
## 3 Afghanistan Crop Residues Direct emissions (N2O) 2002                     0.82
## 4 Afghanistan Crop Residues Direct emissions (N2O) 2003                     0.999
## 5 Afghanistan Crop Residues Direct emissions (N2O) 2004                     0.822
## 6 Afghanistan Crop Residues Direct emissions (N2O) 2005                     1.18
## # i abbreviated name: 1: `total emissions (in kilotonnes)`
```

```r

yearly_emissions_by_area <- aggregate(df_longer$'total emissions (in kilotonnes)', by = list(df_longer$y

yearly_emissions_by_area

#rename columns
yearly_emissions_by_area <-
yearly_emissions_by_area %>%
  rename(
    year = Group.1,
    country = Group.2,
    emissions = x
    )

yearly_emissions_by_area

print(head(df_longer))
```

Total emissions per country for each year

Analyze overall total emissions per country for each year Too many different countries

```r
ggplot(yearly_emissions_by_area, aes(x = year, y = emissions,
                                      fill = country)) +
  geom_tile()
```

Jordan
Kazakhstan
Kenya
Kiribati
Kuwait
Kyrgyzstan
Land Locked Developing Countries
Lao People's Democratic Republic
Latvia
Least Developed Countries
Lebanon
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Low Income Food Deficit Countries
Luxembourg
Madagascar

Malaysia
Maldives
Mali
Malta
Marshall Islands
Martinique
Mauritania
Mauritius
Mayotte
Melanesia
Mexico
Micronesia
Micronesia (Federated States of)
Middle Africa
Monaco
Mongolia
Montenegro
Montserrat
Morocco

Myanmar
Namibia
Nauru
Nepal
Net Food Importing Developing Co
Netherlands
Netherlands Antilles (former)
New Caledonia
New Zealand
Nicaragua
Niger
Nigeria
Niue
Non−Annex I countries
Norfolk Island
North Macedonia
Northern Africa
Northern America
Northern Europe

```r
print(head(df))
```

```
##          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
##      2000     2001     2002     2003     2004     2005     2006     2007
## 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
##        2008     2009     2010     2011     2012     2013     2014     2015
## 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
##        2016     2017     2018     2019     2020
## 1   1.1940   1.0617   0.8988   1.2176   1.3170
## 2   0.2687   0.2389   0.2022   0.2740   0.2963
```

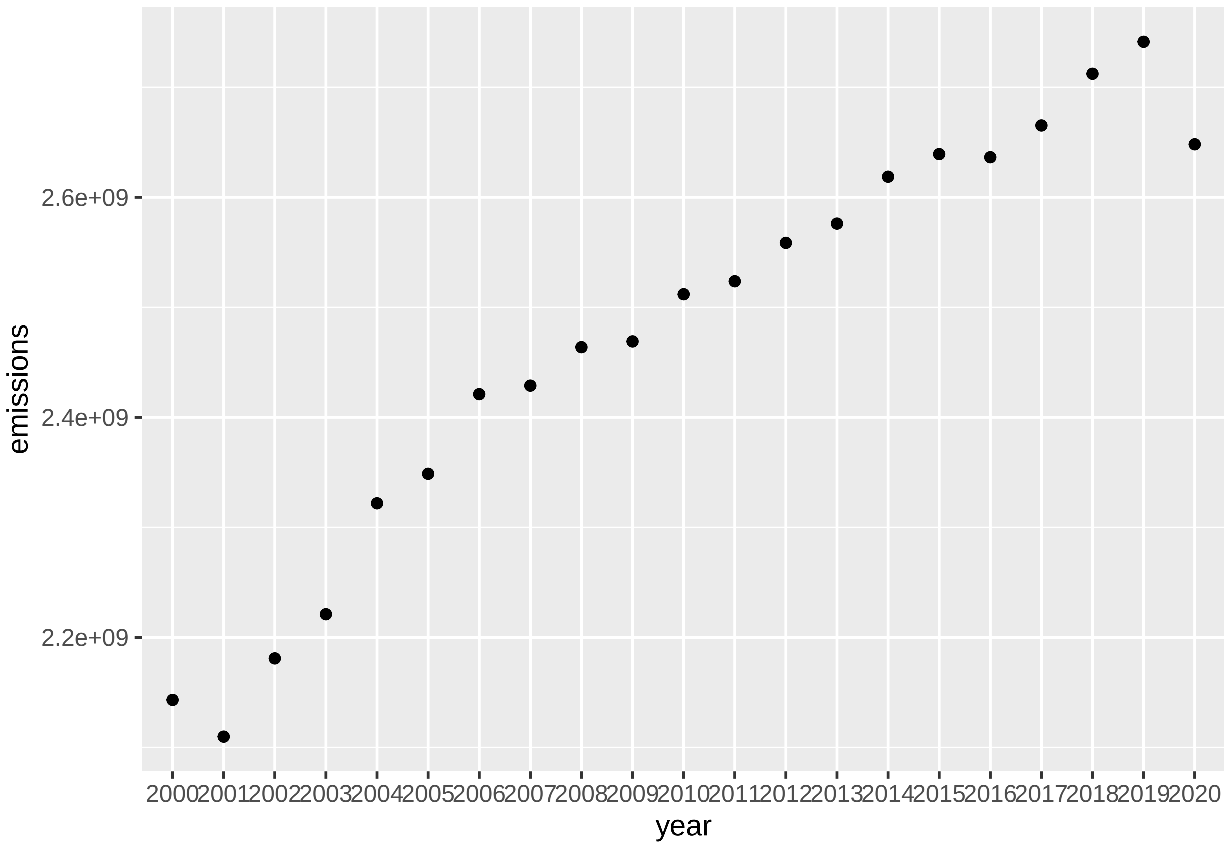
```
## 3   1.4627   1.3005   1.1011   1.4916   1.6133
## 4 387.6130 344.6447 291.7838 395.2689 427.5284
## 5 387.6130 344.6447 291.7838 395.2689 427.5284
## 6  16.6600  15.3233  16.4555  17.8542  20.6577
```

Analyze total emissions over time As you can see from the graph, total emissions have gone up steadily from 2000 to 2019, but in 2020, it decreased a significant amount. This might be due to more awareness about climate change and global warming.

```r
yearly_emissions <- aggregate(df_longer$'total emissions (in kilotonnes)', by=list(df_longer$year), FUN
yearly_emissions
```

```
##    Group.1          x
## 1     2000 2143119367
## 2     2001 2109760240
## 3     2002 2180860383
## 4     2003 2220985897
## 5     2004 2321819994
## 6     2005 2348671376
## 7     2006 2421011446
## 8     2007 2428782054
## 9     2008 2463686673
## 10    2009 2468939225
## 11    2010 2511864950
## 12    2011 2523633556
## 13    2012 2558572795
## 14    2013 2576071615
## 15    2014 2618685489
## 16    2015 2639203455
## 17    2016 2636367158
## 18    2017 2665248135
## 19    2018 2712258358
## 20    2019 2741323659
## 21    2020 2648131930
```

```r
#rename columns
yearly_emissions <-
yearly_emissions %>%
  rename(
    year = Group.1,
    emissions = x
    )

ggplot(yearly_emissions, aes(x = year, y = emissions)) +
  geom_point()
```



Total emissions per country Some of the top countries that contributed to emissions are China, USA, Brazil, India, Indonesia, and Democratic Republic of the Congo.

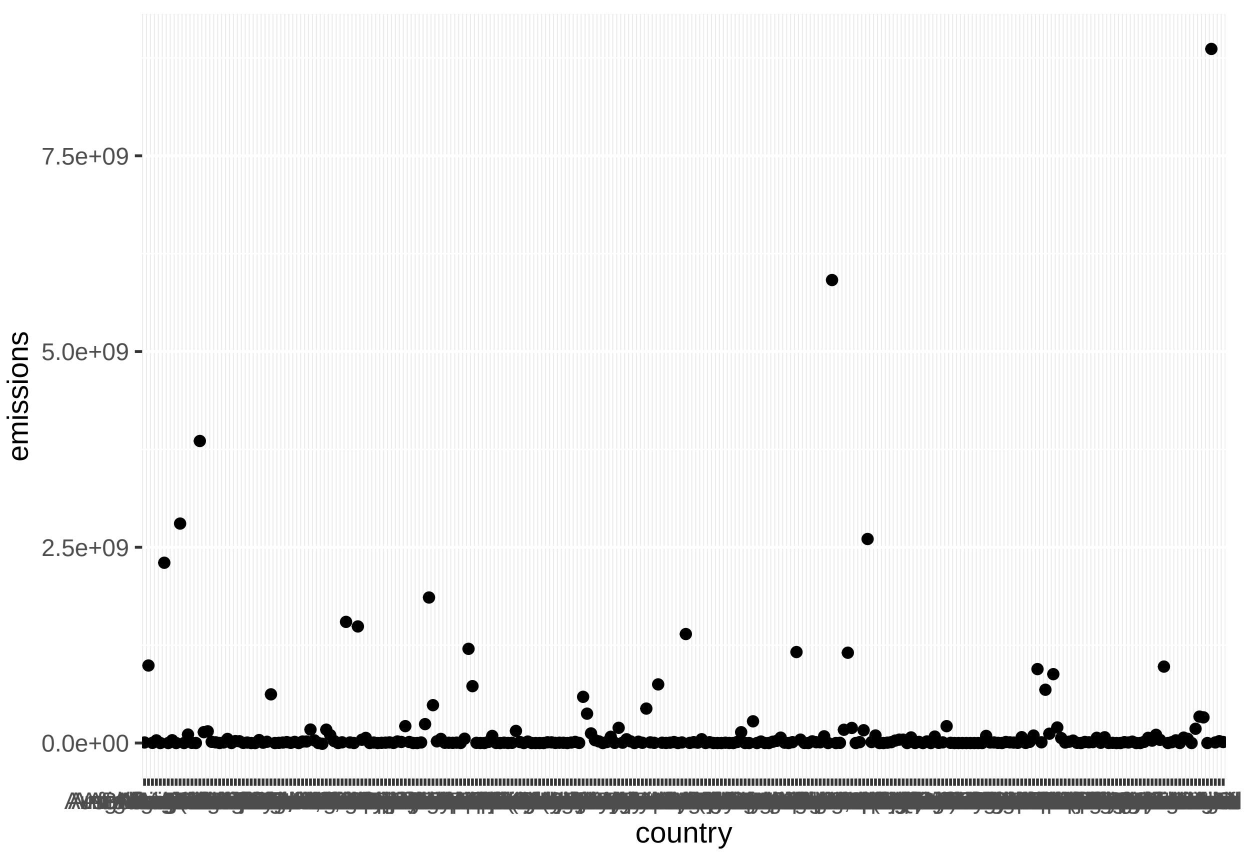
```
emissions_by_area <- aggregate(df_longer$'total emissions (in kilotonnes)', by = list(df_longer$Area), 

#rename columns
emissions_by_area <-
emissions_by_area %>%
  rename(
    country = Group.1,
    emissions = x
    )

top <- emissions_by_area[order(-emissions_by_area$emissions),]

print(head(df))
```

```
##          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
##      2000     2001     2002     2003     2004     2005     2006     2007
## 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
##        2008     2009     2010     2011     2012     2013     2014     2015
## 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
##        2016     2017     2018     2019     2020
## 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
```

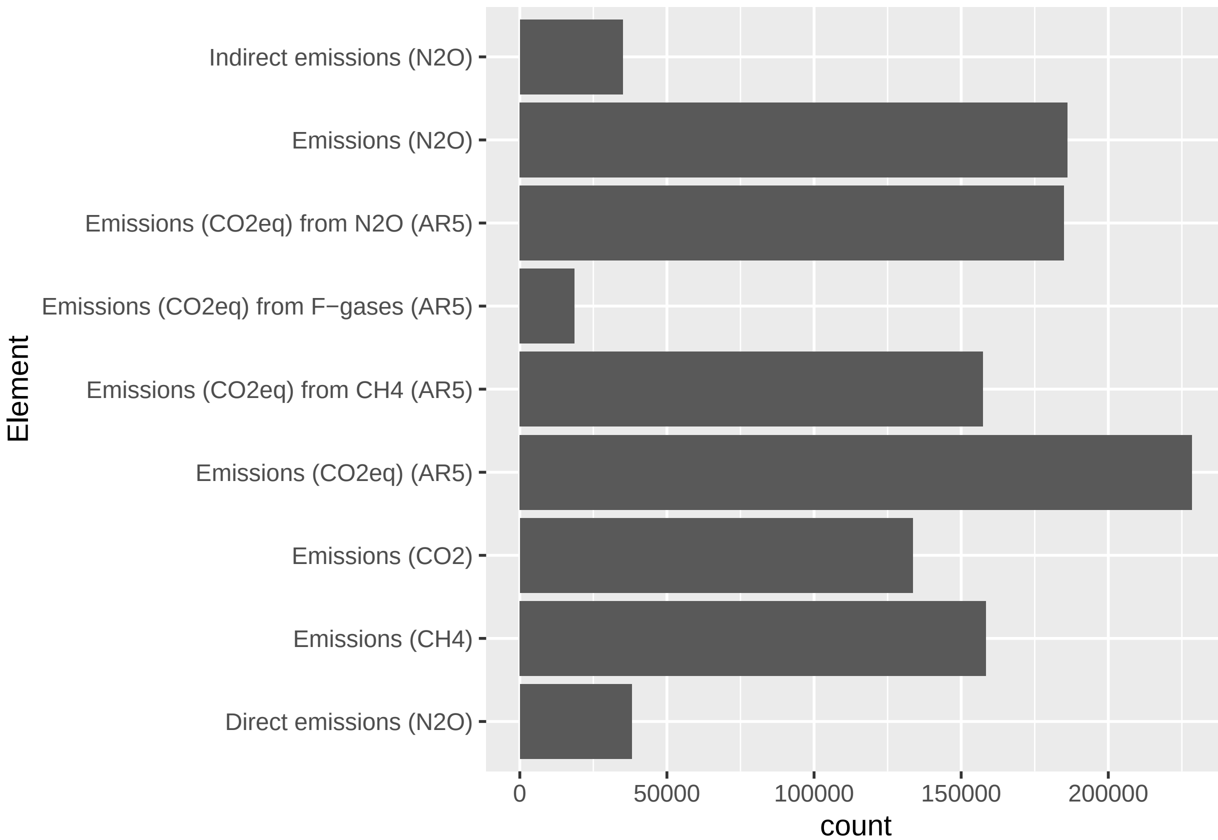
Analysis of Total emissions per country too many countries, cant read

```
ggplot(emissions_by_area, aes(x = country, y = emissions), label=NA)+ geom_point()
```



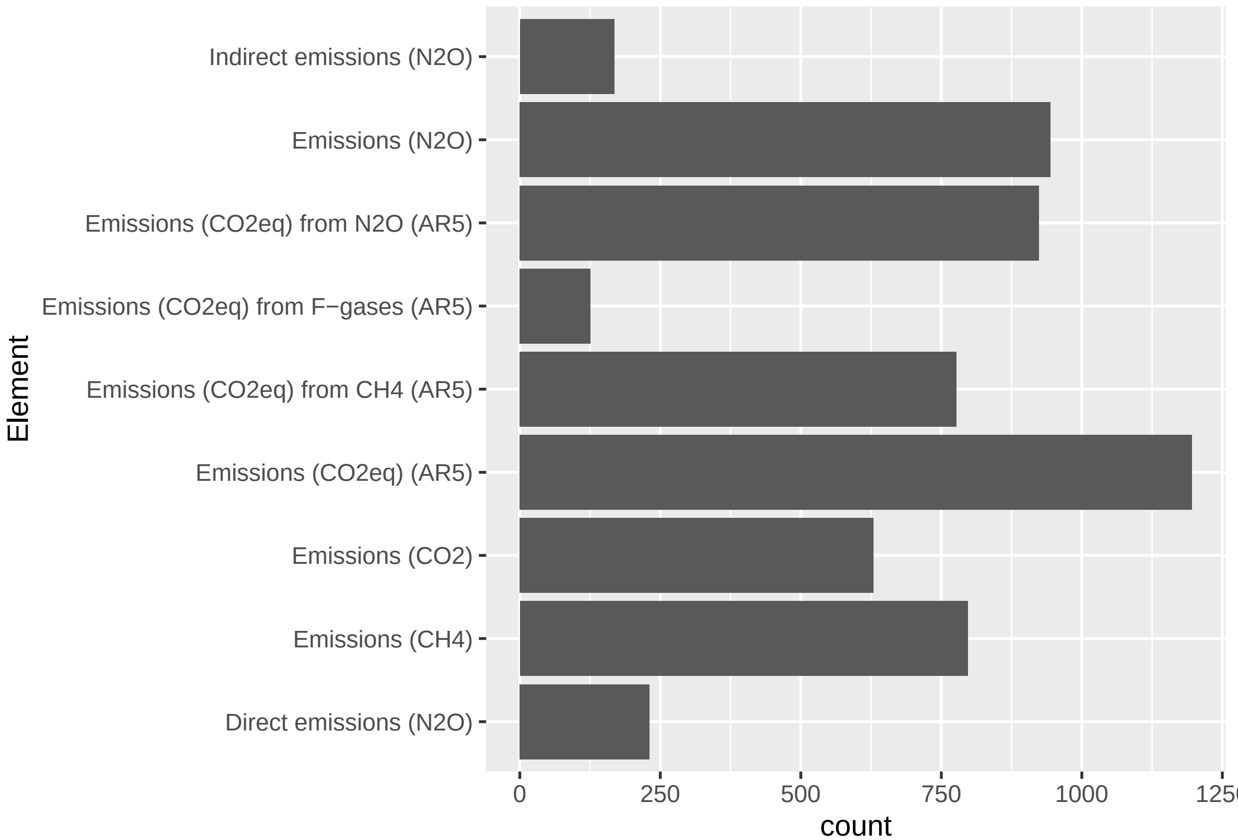
Aanalyze by emission type Emissions (CO2eq) (AR5) are highest. They are over 200,000 kilotonnes. The second highest place is tied with emissions (N20) and emissions (CO2eq) from N20 (AR5). Lowest emissions are (CO2eq) from F-gases, less than 25,000 kilotonnes.

```r
ggplot(df_longer, aes(y=Element)) +
         geom_bar()
```



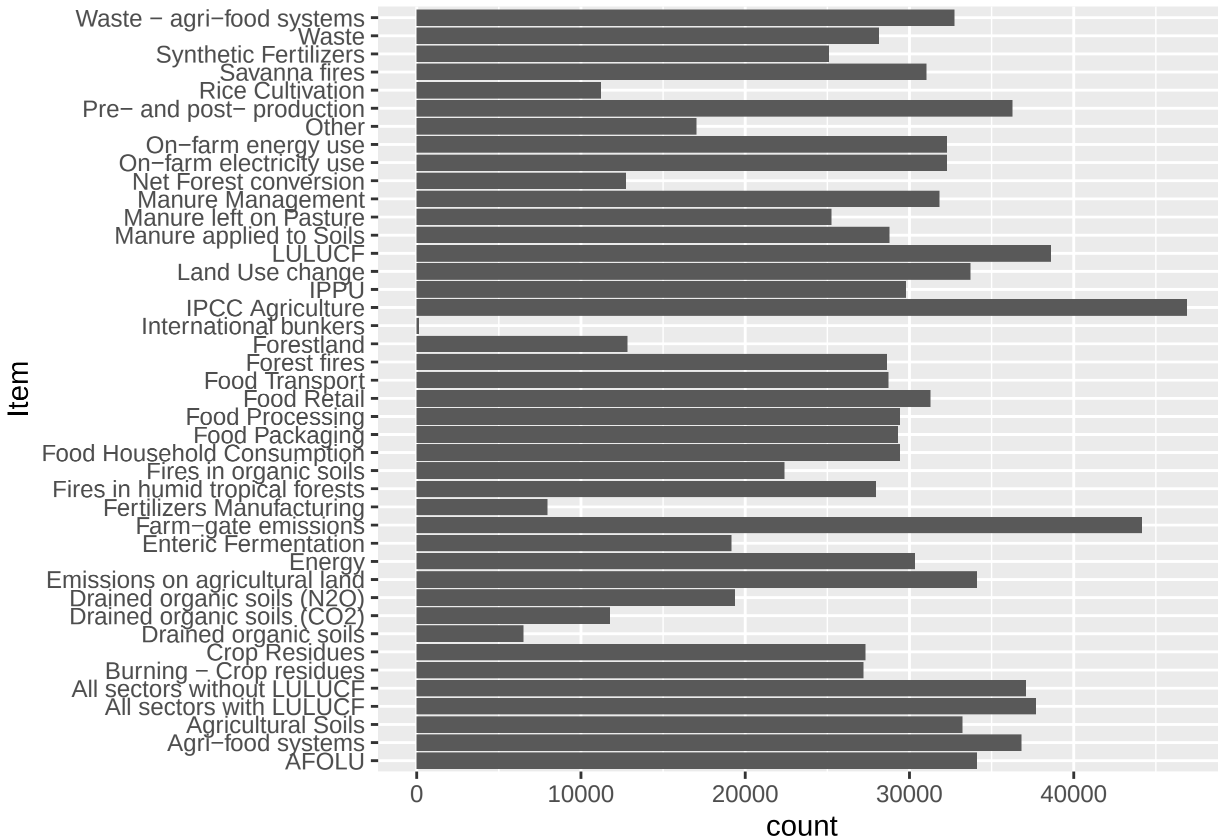
USA emission types distribution The distribution looks very similar to the distribution with the data from all the regions. For USA, the counts are smaller. Highest are emissions (CO2eq) (AR5), a little less than 1250 kilotonnes. Lowest emissions are (CO2eq) from F-gases, around 125 kilotonnes. The second highest place is from emissions (N20), around 950 kilotonnes.

```r
usa <- df_longer %>%
  filter(Area == "United States of America")

ggplot(usa, aes(y=Element)) +
         geom_bar()
```



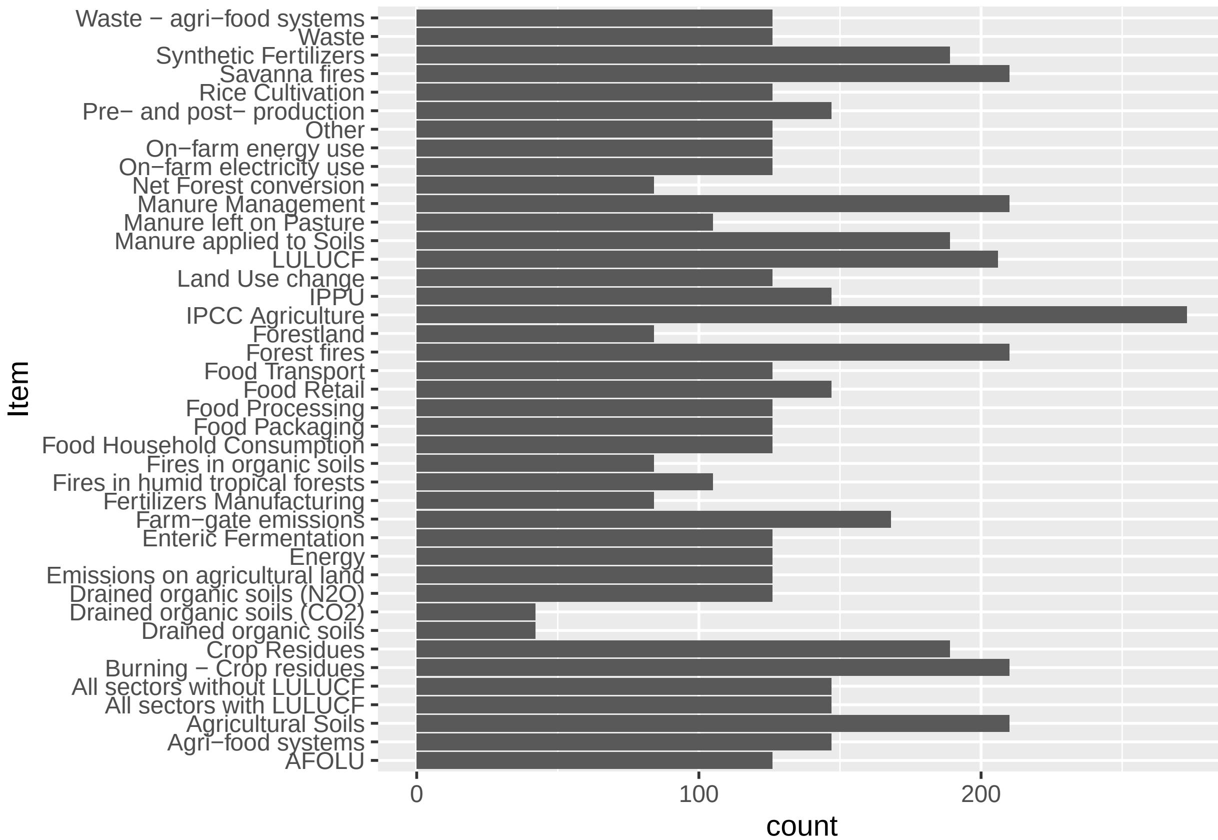
Item Analysis Highest item is IPCC Agriculture. Second highest is farm gate emissions. Lowest is international bunkers.

```
ggplot(df_longer, aes(y=Item)) +
        geom_bar()
```



Item Analysis USA Highest item is IPCC Agriculture, just like in the overall data. Lowest is drained organic soils (C02) and drained organic soils.

```
ggplot(usa, aes(y=Item)) +
         geom_bar()
```


Waste − agri−food systems
Waste
Synthetic Fertilizers
Savanna fires
Rice Cultivation
Pre− and post− production
Other
On−farm energy use
On−farm electricity use
Net Forest conversion
Manure Management
Manure left on Pasture
Manure applied to Soils
LULUCF
Land Use change
IPPU
IPCC Agriculture
Forestland
Forest fires
Food Transport
Food Retail
Food Processing
Food Packaging
Food Household Consumption
Fires in organic soils
Fires in humid tropical forests
Fertilizers Manufacturing
Farm−gate emissions
Enteric Fermentation
Energy
Emissions on agricultural land
Drained organic soils (N2O)
Drained organic soils (CO2)
Drained organic soils
Crop Residues
Burning − Crop residues
All sectors without LULUCF
All sectors with LULUCF
Agricultural Soils
Agri−food systems
AFOLU
Item
0
100
200
count

NYSDEC Water Permit Data (DART):

Loading NYSDEC DART Data Data used in this section comes from New York State Department of Environmental Conservation's Application Review & Tracking System (DART on the Web).

DART is a web-based application and tracking system that is designed for the general public. DART hosts information about NYSDEC's processing and issuance of environmental permits under the Uniform Procedures Act. The data is updated daily, and more information about the data can be found in the data dictionary.

In this section, data was previously filtered to only include DART entries from 2020-2025, and will be focused on waste water permits that discharge to surface water.

```
library(readr)
dart <- read_csv("https://raw.githubusercontent.com/AlinaVikhnevich/data_607/refs/heads/main/Project%20
```

Defining Regex Patterns to Detect NPDES IDs To identify wastewater permits, there are three regex patterns to identify:

1. NPDES Permit (meaning a regular permit).
2. General Permit
3. Individual Permit (these are permits that are processed under general permits).

For more information about permit types please see the question "What are the primary differences between a NODES individual permit and a NPDES general permit" under EPA's NPDES Permit Basics Site.

```
# p_type = permit type, in this exercise we are filtering for wastewater permits
p_type <- c("P/C/I SPDES - Surface Discharge",
            "Municipal SPDES - Surface Discharge",
            "Industrial SPDES - Surface Discharge")

# defining the regex patterns for the IDs we want to track
npdes_pattern <- "NY\\d{7}"
gp_pattern <- "GP\\d{7}"
individual_pattern <- "NY[A-Z]\\d{2}[A-Z]\\d{3}"
all_patterns <- paste(npdes_pattern,gp_pattern,individual_pattern, sep="|")
```

Creating the NPDES Universe Creating the permit universe pulling from NYSDEC's DART System and detecting the string patterns within DART to assign permit type: npdes, individual(i.e., a permit covered under a general permit), general, or multi (meaning the DART entry had multiple associated IDs).

```
universe <- dart |>
  filter(`permit_type` %in% p_type) |>
  mutate(
    npdes = str_count(`other_known_ids`, npdes_pattern), # the str_counts are taking count of permit ID
    individual = str_count(`other_known_ids`, individual_pattern),
    gp =  str_count(`other_known_ids`, gp_pattern),
    sum_ids = rowSums(across(c(`npdes`, `individual`,`gp`))),
    npdes_id = str_extract_all(`other_known_ids`, all_patterns),
    date_received=as.Date(date_received, format = "%d-%m-%Y")
    ) |>
  mutate(applicant_id =cur_group_id(),.by = applicant) |> # creating applicant id
  mutate(facility_id = cur_group_id(),.by = c(facility,location,town_or_city)) |> # creating facility i
  distinct() |> # removing duplicate rows
  mutate(
        dart_p_type = case_when(sum_ids  > 1 ~ "multi", # if entry is associated with multiple ids, i
                                sum_ids & npdes == 1 ~ "npdes",
```

```
                                sum_ids & individual == 1 ~ "individual",
                                sum_ids & gp == 1 ~ "gp")) |>
  unnest_longer(npdes_id, keep_empty = FALSE) |>
  filter(!is.na(npdes_id))
```

Note: The code above filters entries that did not have a NPDES ID listed in the "Other Known IDs" column, however, were listed as NPDES permits in the Permit Type Column. However, out of 35,642 entries, only 69 were missing NPDES IDs.

Tidy Data

Table 1: Permit Level Data This table shows the most recent permit information

```
tbl1_permit_lvl <- universe  |>
  group_by(npdes_id) |>
  slice(which.max(date_received)) |>
  select(npdes_id,facility_id,application_id,applicant,applicant_id,permit_type,
         status,date_received,upa_class,seqr_class,seqr_determination,
         lead_agency,coastal_zone_status, final_disposition,permit_effective_date,
         permit_expration_date,dec_contact,shpa_status,enivronmental_justice)
```

Permit Data Head

npdes_id

facility_id

application_id

applicant

applicant_id

permit_type

status

date_received

upa_class

seqr_class

seqr_determination

lead_agency

coastal_zone_status

final_disposition

permit_effective_date

permit_expration_date

dec_contact

shpa_status

enivronmental_justice

NY0000044

574

3-3724-00045/00004

WATCHTOWER BIBLE AND TRACT SOCIETY OF NEW YORK INC

158

P/C/I SPDES - Surface Discharge

Issued

2024-02-16

MINOR

Type II Action

Not Applicable

None Designated

This project is not located in a Coastal Management area.

Issued

2024-09-01 04:00:00

2029-08-31 04:00:00

KATHERINE M MURRAY

NA

NA

NY0000078

706

8-5436-00007/00014

GARLOCK SEALING TECHNOLOGIES LLC

593

Industrial SPDES - Surface Discharge

SAPA Extended

2021-02-03

MAJOR

Type II Action

Not Applicable

None Designated

This project is not located in a Coastal Management area.

Issued

2023-07-01 04:00:00

2023-08-31 04:00:00

GUILLERMO R SAAR

NA

The application is subject to the Department Environmental Justice policy (CP-29). Either the permits needed for the project are not subject to the policy or it has been determined that the project would not affect a Potential Environmental Justice Area.

NY0000167

1906

8-2499-00039/00002

RETSOF REALTY LLC

1571

Industrial SPDES - Surface Discharge

Issued

2021-04-15

MINOR

Type II Action

Not Applicable

None Designated

This project is not located in a Coastal Management area.

Issued

2022-01-01 05:00:00

2026-12-31 05:00:00

MICHAEL R SCHAEFER

NA

NA

NY0000247

1556

3-5518-00680/00001

CONSOLIDATED EDISON COMPANY OF NEW YORK, INC.

914

Industrial SPDES - Surface Discharge

Suspended Indefinitely

2024-04-22

MINOR

Type II Action

Not Applicable

None Designated

This project is not located in a Coastal Management area.

NA

NA

NA

KATHERINE M MURRAY

NA

NA

NY0000281

539

9-1464-00117/00013

LINDE INC

455

Industrial SPDES - Surface Discharge

Issued

2021-04-29

MINOR

Type II Action

Not Applicable

None Designated

This project is not located in a Coastal Management area.

Issued

2022-01-11 05:00:00

2026-12-31 05:00:00

MICHAEL R SCHAEFER

NA

NA

NY0000311

1420

8-3224-00108/00031

PACTIV LLC

1186

Industrial SPDES - Surface Discharge

Issued

2021-08-02

MINOR

Type II Action

Not Applicable

None Designated

This project is not located in a Coastal Management area.

Issued

2022-03-01 05:00:00

2027-02-28 05:00:00

MICHAEL R SCHAEFER

NA

NA

Table 2: Permit Action Level Data This table shows the permit history. each observation in this table represents a permit action.

```
tbl2_permit_act_lvl <- universe |>
    mutate(action_id = paste(npdes_id,date_received, sep = "_")) |>
    distinct() |>
    mutate(dup_flag = duplicated(action_id),
           transfer_flag=str_detect(toupper(short_description),"TRANSFER")) |>
  select(action_id,facility,facility_id,npdes_id,application_id,applicant,
         application_type,date_received,status,short_description,
         enb_publication_date,written_comments_due,dup_flag,transfer_flag)

tbl2_permit_act_lvl$short_description <- tolower(tbl2_permit_act_lvl$short_description)
```

Permit Action Data Head

action_id

facility

facility_id

npdes_id

application_id

applicant

application_type

date_received

status

short_description

enb_publication_date

written_comments_due

dup_flag

transfer_flag

NY0101915_2022-10-05

B O C E S

1

NY0101915

3-1332-00172/00001

DUTCHESS BOCES

Renewal Treat as New

2022-10-05

Issued

spdes fast track renewal for ny0101915

2023-02-15 05:00:00

2023-03-17 04:00:00

FALSE

FALSE

NYR10L635_2023-03-24

NYC EDC-SOUTH BROOKLYN MARINE TERMINAL

2

NYR10L635

2-6102-00120/00032

NYC ECONOMIC DEVELOPMENT CORP

New

2023-03-24

Issued

spdes application for substation test pits and construction

2024-03-20 04:00:00

2024-04-19 04:00:00

FALSE

FALSE

NY0313149_2023-03-24

NYC EDC-SOUTH BROOKLYN MARINE TERMINAL

2

NY0313149

2-6102-00120/00032

NYC ECONOMIC DEVELOPMENT CORP

New

2023-03-24

Issued

spdes application for substation test pits and construction

2024-03-20 04:00:00

2024-04-19 04:00:00

FALSE

FALSE

NY0035441_2021-06-30

CHAUTAUQUA FISH HATCHERY

3

NY0035441

9-0628-00098/00004

NYS Dept of Environmental Conservation

New

2021-06-30

Issued

reduce conc. limit of formalin & change in phos. loading limit

2021-09-29 04:00:00

2021-10-29 04:00:00

FALSE

FALSE

NY0071897_2021-03-30

EFFRON FUEL OIL CORP TERMINAL

4

NY0071897

3-1313-00015/00002

PETRO INC

Minor Modification

2021-03-30

Expired

transfer

NA

NA

FALSE

TRUE

NY0071897_2022-09-08

EFFRON FUEL OIL CORP TERMINAL

4

NY0071897

3-1313-00015/00002

PETRO INC

Renewal Treat as New

2022-09-08

Issued

spdes fast track renewal for ny 0071897

2022-11-02 04:00:00

2022-12-02 05:00:00

FALSE

FALSE

Table 3: Facility Level Data This table shows the facility information. Each observation in this table represents a facility associated with NPDES permits.

```
tbl3_facility_lvl <- universe |>
  select(facility_id,facility,
         location,town_or_city) |>
  distinct() |>
  arrange(facility_id)
```

Facility Data Head

facility_id

facility

location

town_or_city

1

B O C E S

578 SALT POINT TURNPIKE HYDE PARK 12538

HYDE PARK

2

NYC EDC-SOUTH BROOKLYN MARINE TERMINAL

Sunset Park and Greenwood Heights 29th to 39th St|2nd Ave To Ny Harbor Brooklyn (6102) 11232

BROOKLYN

3

CHAUTAUQUA FISH HATCHERY

5875 PRENDERGAST RD MAYVILLE 14757

CHAUTAUQUA

4

EFFRON FUEL OIL CORP TERMINAL

FOOT OF PROSPECT ST POUGHKEEPSIE 12602

POUGHKEEPSIE

5

POUGHKEEPSIE STP

173 KITTREDGE PL POUGHKEEPSIE 12601

POUGHKEEPSIE

6

AMENIA S & G-LEEDSVILLE PROCESSING PLANT

307 LEEDSVILLE RD AMENIA 12501

AMENIA

Table 4: NPDES Permit Applicant Table This table shows the applicant information. Each observation in this table represents a permit applicant for NPDES permits.

```
tbl4_app_lvl <- universe |>
  group_by(applicant_id) |>
  slice(which.max(date_received)) |>
  select(applicant_id,applicant,application_id)
```

NPDES Permit Applicant Data Head

applicant_id

applicant

application_id

1

DUTCHESS BOCES

3-1332-00172/00001

2

NYC ECONOMIC DEVELOPMENT CORP

2-6402-00004/00100

3

NYS Dept of Environmental Conservation

3-4844-00112/00001

4

PETRO INC

3-1334-00136/00001

5

CITY OF POUGHKEEPSIE

3-1346-00364/00003

6

DOLOMITE PRODUCTS COMPANY INC.

8-1836-00001/02002

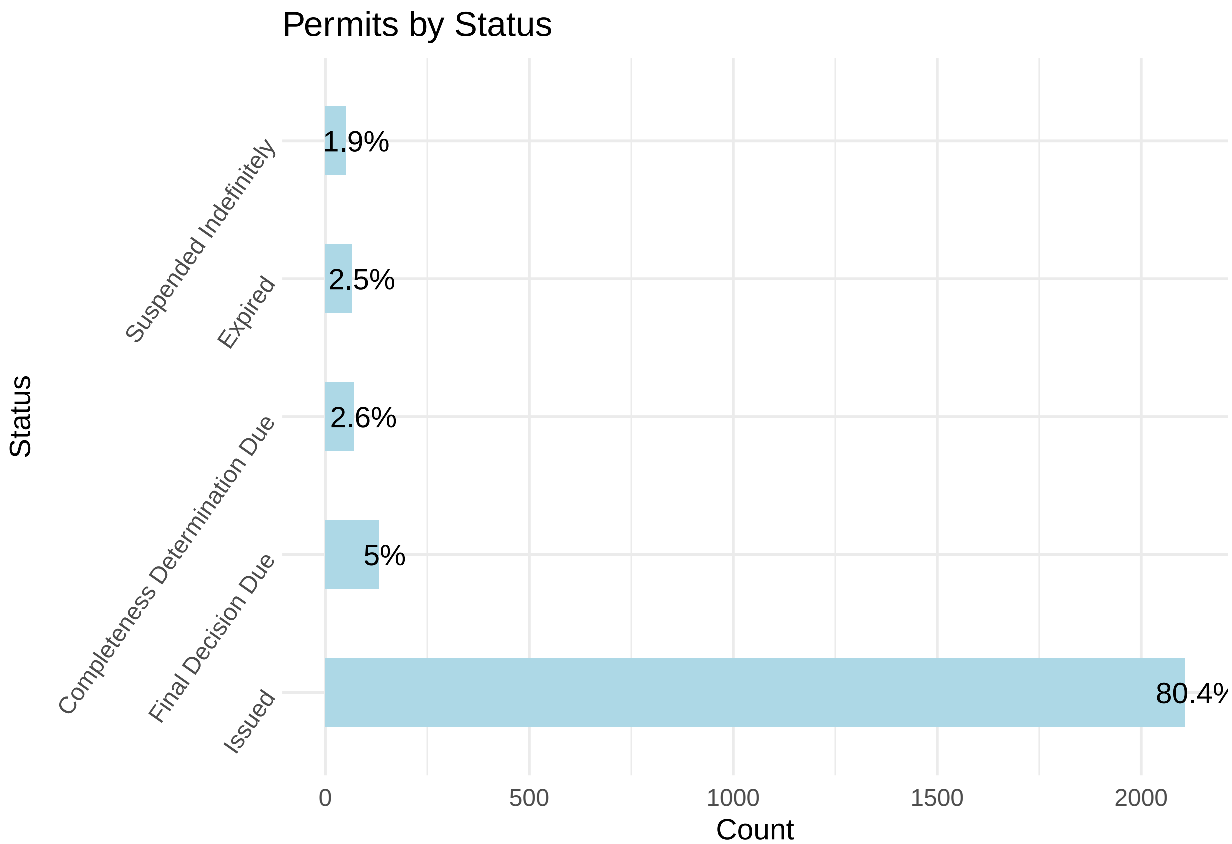
Data Tables and Structure

```
(1) Table 1 - permit table: the purpose of this table is to have the most
recent permit information. This will have one row per permit.

(2) Table 2 - permit action table: the purpose of this table is to have a
table with every permit-action. This means there should be one row per
permit action.

(3) Table 3 - facility table: the purpose of this table is to have
information on the facility.

(4) Table 4 - applicant table: the purpose of this table is to have
information about the applicant.
```

Data Considerations:

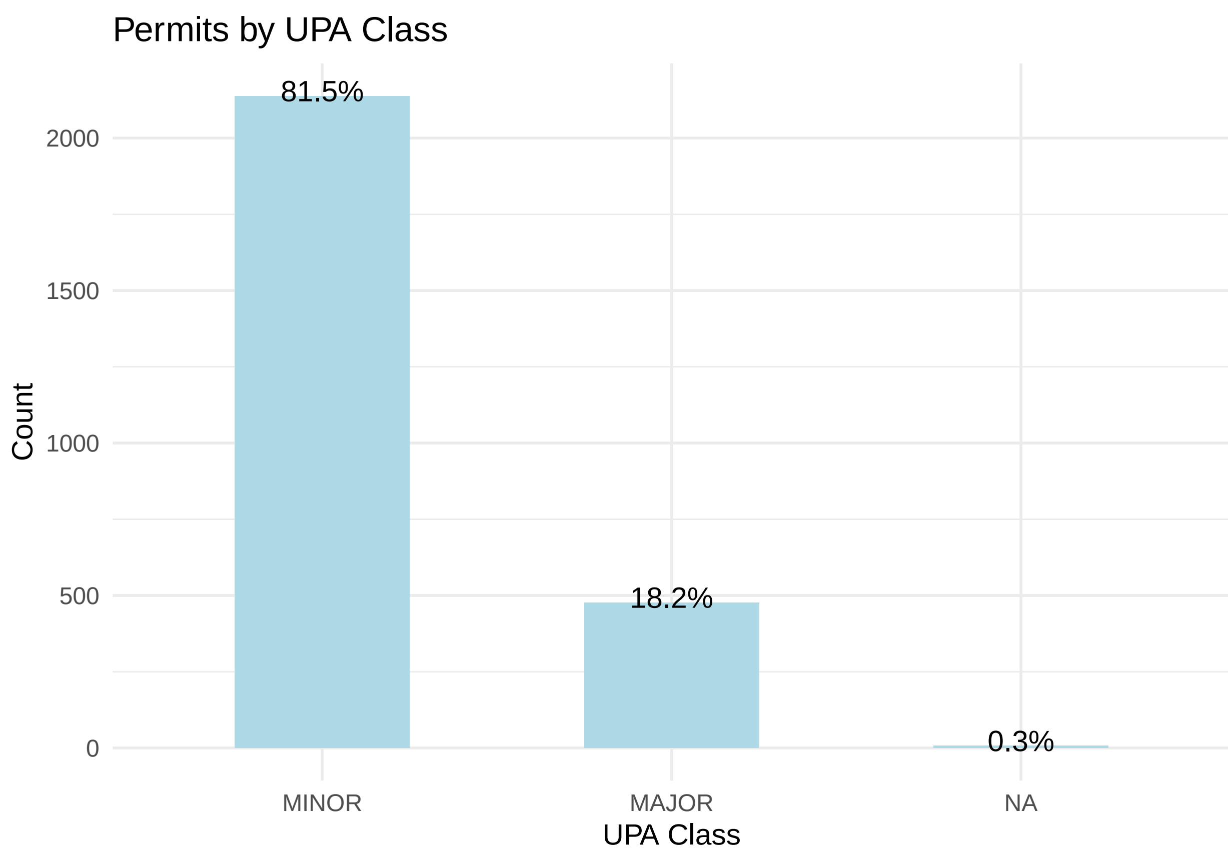
- There was missing data, such as NPDES IDs. This means that some permit information may not be available.
- There may be facilities that are listed as different facilities due to address changes. This information should be verified. Databases like EPA's Enforcement and Compliance History Online (ECHO)) may be helpful for verifying facility information.
- For entries that were made on the same day for a particular permit, it is not possible to identify which entry was made first. Permit transfer actions are largely affected by this. Due to this, duplicates and transfers are flagged for manual review.

```r
permit_status <- tbl1_permit_lvl |>
group_by(status) |>
  summarize(
    Count = n(),
    Proportion = (n()/nrow(tbl1_permit_lvl))*100
  ) |>
  arrange(desc(Proportion)) |>
  head(5) |>
  rename("Status" = "status")

permit_status$Proportion <- paste0(round(permit_status$Proportion, digits=1),"%")

ggplot(permit_status,aes(x = reorder(Status, -Count), y= Count)) +
    geom_bar(stat="identity", fill="lightblue", width=0.5)+
    geom_text(aes(label=Proportion),
              hjust=.35)+
    theme_minimal()+
    labs(title="Permits by Status",x="Status")+
    theme(axis.text.y =element_text(angle = 55,hjust=1))+
    coord_flip()
```

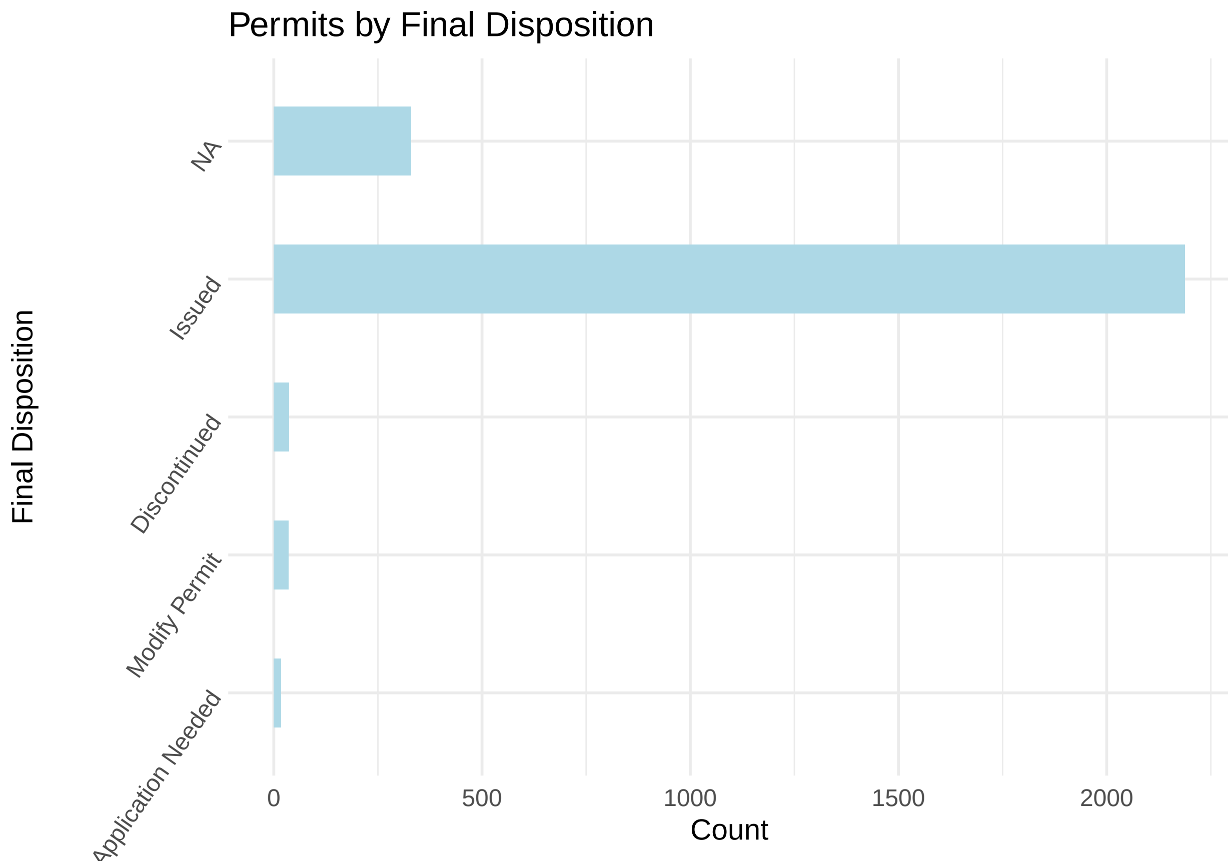


Analysis

```
upa_class <- tbl1_permit_lvl |>
group_by(upa_class) |>
  summarize(
    Count = n(),
    Proportion = (n()/nrow(tbl1_permit_lvl))*100
  )

upa_class$Proportion <- paste0(round(upa_class$Proportion, digits=1),"%")

ggplot(upa_class,aes(x = reorder(upa_class, -Count), y= Count)) +
    geom_bar(stat="identity", fill="lightblue", width=0.5)+
    geom_text(aes(label=Proportion),
              hjust=.5,
              vjust=0.25)+
    theme_minimal()+
    labs(title="Permits by UPA Class",x="UPA Class")
```




```r
final_dis <- tbl1_permit_lvl |>
group_by(final_disposition) |>
  summarize(
    Count = n(),
    Proportion = (n()/nrow(tbl1_permit_lvl))*100
  ) |>
  arrange(desc(Count)) |>
  head(5)

final_dis$Proportion <- paste0(round(final_dis$Proportion, digits=1),"%")
final_dis$Count <- as.numeric(final_dis$Count)

final_dis <- final_dis |>
  clean_names("title")

ggplot(final_dis,aes(x =reorder(`Final Disposition`,`Count`, .desc = TRUE), y= Count)) +
    geom_bar(stat="identity", fill="lightblue", width=0.5)+
    theme_minimal()+
    labs(title="Permits by Final Disposition",x="Final Disposition")+
    theme(axis.text.y =element_text(angle = 55,hjust=1))+
    coord_flip()
```




```
app_type <- tbl2_permit_act_lvl |>
group_by(application_type) |>
  summarize(
    Count = n(),
    Proportion = n()/nrow(tbl2_permit_act_lvl)
  ) |>
  clean_names("title") |>
  arrange(desc(Count))

knitr::kable(app_type, format ="markdown")
```

Application Type	Count	Proportion
Renewal Treat as New	2565	0.7753930
Modification Treat as New	274	0.0828295
Minor Modification	185	0.0559250
New	160	0.0483676
Modification	48	0.0145103
DIM Treat as New	47	0.0142080
Department Initiated Modification	29	0.0087666

```
short_desc <- tbl2_permit_act_lvl |>
  mutate(c_fast_track=coalesce(str_count(short_description,"fast track"),0)) |>
  summarize(
    "Fast Tracked Renewal Actions" = sum(c_fast_track),
    "Total Actions" = nrow(tbl2_permit_act_lvl),
```

```
    Proportion = sum(c_fast_track)/nrow(tbl2_permit_act_lvl)
  ) |>
  clean_names("title")

knitr::kable(short_desc, format ="markdown")
```

Fast Tracked Renewal Actions	Total Actions	Proportion
2204	3308	0.6662636

Cheese Dataset:

Import Data Read in raw csv file as data frame

```
data <- read.csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/refs/heads/main/data/20
```

View Raw Data Raw Data Head

cheese

url

milk

country

region

family

type

fat_content

calcium_content

texture

rind

color

flavor

aroma

vegetarian

vegan

synonyms

alt_spellings

producers

Aarewasser

https://www.cheese.com/aarewasser/

cow

Switzerland

NA

NA

semi-soft

NA

NA

buttery

washed

yellow

sweet

buttery

FALSE

FALSE

NA

NA

Jumi

Abbaye de Belloc

https://www.cheese.com/abbaye-de-belloc/

sheep

France

Pays Basque

NA

semi-hard, artisan

NA

NA

creamy, dense, firm

natural

yellow

burnt caramel

lanoline

TRUE

FALSE

Abbaye Notre-Dame de Belloc

NA

NA

Abbaye de Belval

https://www.cheese.com/abbaye-de-belval/

cow

France

NA

NA

semi-hard

40-46%

NA

elastic

washed

ivory

NA

aromatic

FALSE

FALSE

NA

NA

NA

Abbaye de Citeaux

https://www.cheese.com/abbaye-de-citeaux/

cow

France

Burgundy

NA

semi-soft, artisan, brined

NA

NA

creamy, dense, smooth

washed

white

acidic, milky, smooth

barnyardy, earthy

FALSE

FALSE

NA

NA

NA

Abbaye de Tamié

https://www.cheese.com/tamie/

cow

France

Savoie

NA

soft, artisan

NA

NA

creamy, open, smooth

washed

white

fruity, nutty

perfumed, pungent

FALSE

FALSE

NA

Tamié, Trappiste de Tamie, Abbey of Tamie

NA

Abbaye de Timadeuc

https://www.cheese.com/abbaye-de-timadeuc/

cow

France

province of Brittany

NA

semi-hard

NA

NA

soft

washed

pale yellow

salty, smooth

nutty

FALSE

FALSE

NA

NA

Abbaye Cistercienne NOTRE-DAME DE TIMADEUC

Raw Data Stats

Row_Count

Column_Count

Null_Count

None_Str_Count

1187

19

7133

0

Data Handling

- select columns needed to tidy and for analysis
- fill empty strings and null values with 'None' string

```
fill_empty_str = function(x){if_else(x=="", 'None' ,x)}

df = data |>
  select(cheese, milk, country, texture, aroma, flavor) |>
  mutate_all(fill_empty_str) |>
  mutate_all(replace_na, "None")
```

Data after Handling Stats

Row_Count

Column_Count

Null_Count

None_Str_Count

1187

6

0

340

Tidy Data Tidy data by ensuring each value has its own cell

- split out each row with listed values (milk, texture, aroma, flavor, country) into individual rows and lengthen the data frame

```
df = df |>
  mutate(cheese_id = row_number()) |>
  separate_rows(country, sep = ', ') |>
  separate_rows(milk, sep = ', ') |>
  separate_rows(texture, sep = ', ') |>
  separate_rows(aroma, sep = ', ') |>
  separate_rows(flavor, sep = ', ')
```

View Tidy Data Tidy Data Head

cheese

milk

country

texture

aroma

flavor

cheese_id

Aarewasser

cow

Switzerland

buttery

buttery

sweet

1

Abbaye de Belloc

sheep

France

creamy

lanoline

burnt caramel

2

Abbaye de Belloc

sheep

France

dense

lanoline

burnt caramel

2

Abbaye de Belloc

sheep

France

firm

lanoline

burnt caramel

2

Abbaye de Belval

cow

France

elastic

aromatic

None

3

Abbaye de Citeaux

cow

France

creamy

barnyardy

acidic

4

Tidy Data Stats

Row_Count

Column_Count

Null_Count

None_Str_Count

14394

7

0

2050

Normalize Data Normalize data to reduce redundancy and allow for more efficient analysis

- create a data frame for each column and create an associated id column for each
- replace all column values with respective id value in core data frame

```
create_id_dfs = function(id_prefix, col, df) {
  id_df = df |>
    select(all_of(col)) |>
    distinct() |>
    arrange(col) |>
    mutate(id = paste0(id_prefix, row_number()))
  return(id_df)
}

cheese_df = df |>
  select(cheese, cheese_id) |>
  distinct()

country_df = create_id_dfs('C', 'country', df)
colnames(country_df) = c('country', 'country_id')

milk_df = create_id_dfs('M', 'milk', df)
colnames(milk_df) = c('milk', 'milk_id')

texture_df = create_id_dfs('T', 'texture', df)
colnames(texture_df) = c('texture', 'texture_id')

aroma_df = create_id_dfs('A', 'aroma', df)
colnames(aroma_df) = c('aroma', 'aroma_id')

flavor_df = create_id_dfs('F', 'flavor', df)
colnames(flavor_df) = c('flavor', 'flavor_id')
```

```
df = left_join(df, country_df, by = join_by(country))
df = left_join(df, milk_df, by = join_by(milk))
df = left_join(df, texture_df, by = join_by(texture))
df = left_join(df, aroma_df, by = join_by(aroma))
df = left_join(df, flavor_df, by = join_by(flavor))

df = df |>
  select(cheese_id, country_id, milk_id, texture_id, aroma_id, flavor_id)
```

View Tidy and Normalized Data Normalized Tidy Data Head

cheese_id

country_id

milk_id

texture_id

aroma_id

flavor_id

1

C1

M1

T1

A1

F1

2

C2

M2

T2

A2

F2

2

C2

M2

T3

A2

F2

2

C2

M2

T4

A2

F2

3

C2

M1

T5

A3

F3

4

C2

M1

T2

A4

F4

Cheese Table Head

cheese

cheese_id

Aarewasser

1

Abbaye de Belloc

2

Abbaye de Belval

3

Abbaye de Citeaux

4

Abbaye de Tamié

5

Abbaye de Timadeuc

6

Country Table Head

country

country_id

Switzerland

C1

France

C2

England

C3

Great Britain

C4

United Kingdom

C5

Czech Republic

C6

Milk Table Head

milk

milk_id

cow

M1

sheep

M2

goat

M3

buffalo

M4

None

M5

water buffalo

M6

Texture Table Head

texture

texture_id

buttery

T1

creamy

T2

dense

T3

firm

T4

elastic

T5

smooth

T6

Aroma Table Head

aroma

aroma_id

buttery

A1

lanoline

A2

aromatic

A3

barnyardy

A4

earthy

A5

perfumed

A6

Flavor Table Head

flavor

flavor_id

sweet

F1

burnt caramel

F2

None

F3

acidic

F4

milky

F5

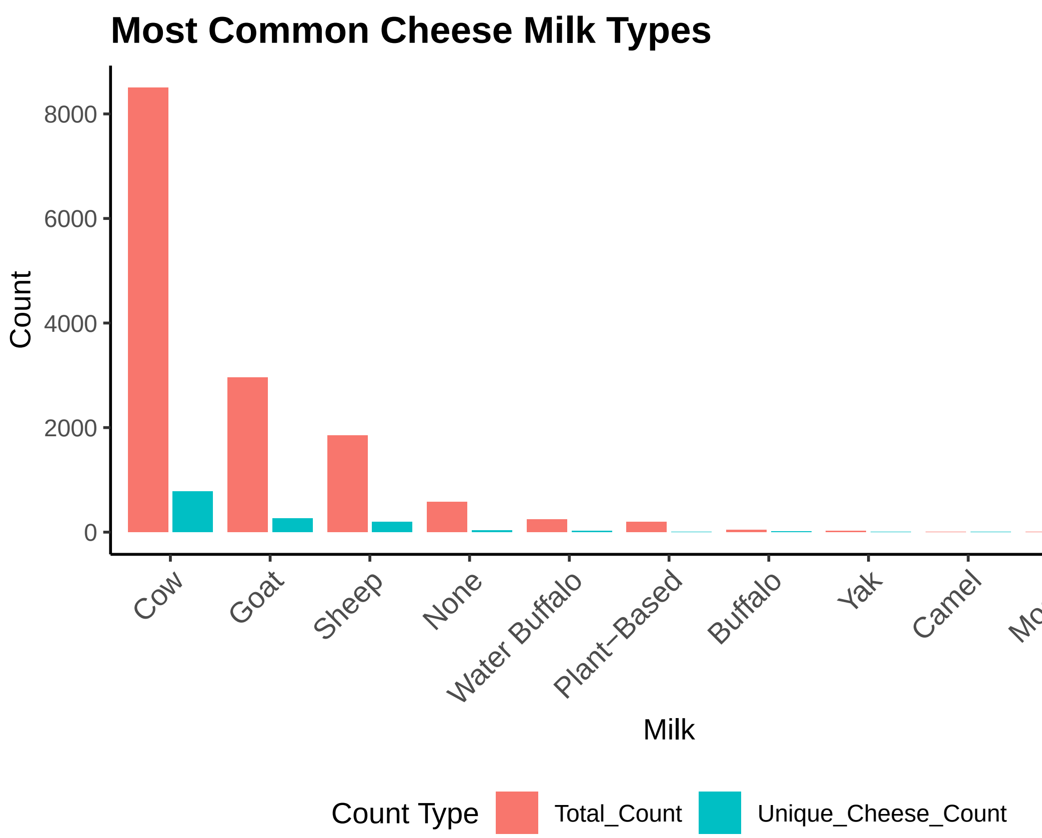
smooth

F6

Analysis Analysis Requested in Discussion Post:

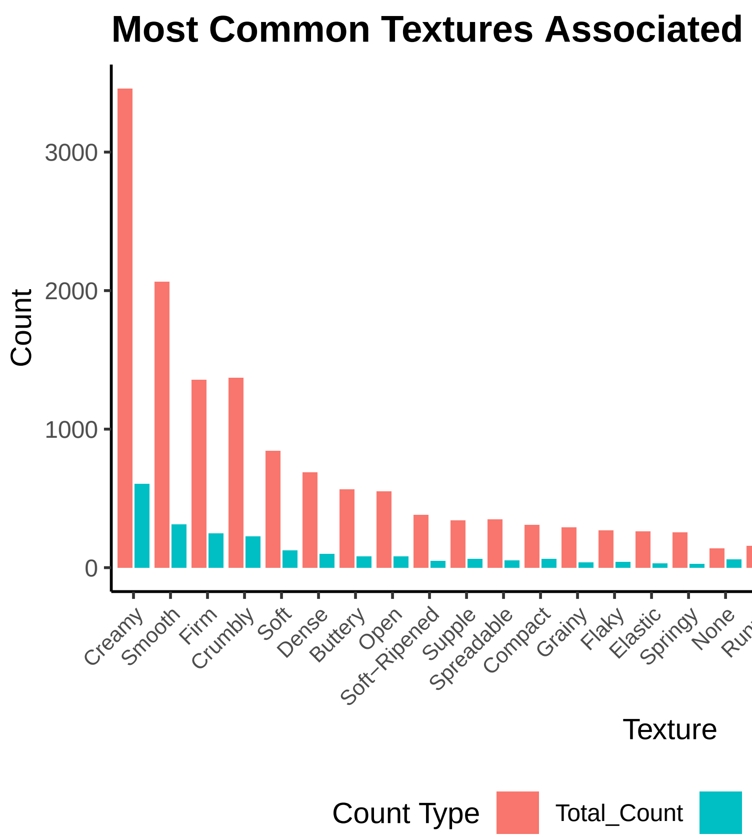
1. What are the most common milks used?
2. What are the more common textures associated with cheese?
3. Is there a country or region that produces more cheese?
4. Are there common aromas or flavors across cheeses made by different milks?

```r
df |>
  select(cheese_id, milk_id) |>
  left_join(cheese_df, by = join_by(cheese_id)) |>
  left_join(milk_df, by = join_by(milk_id)) |>
  mutate(milk = str_to_title(milk)) |>
  group_by(milk) |>
  summarise(
    Total_Count = n(),
    Unique_Cheese_Count = n_distinct(cheese)) |>
  pivot_longer(cols = c(Total_Count, Unique_Cheese_Count)) |>
  ggplot(aes(x = reorder(milk, -value), y = value, fill = name)) +
  geom_col(position = position_dodge2(width = 0.3, preserve = "single")) +
  labs(
    title = "Most Common Cheese Milk Types",
    x = "Milk",
    y = "Count",
    fill = 'Count Type'
  ) +
  theme_classic() +
  theme(
    axis.text.x = element_text(size = 11, angle = 45, vjust = 1, hjust=1),
    plot.title = element_text(size = 14, face = "bold"),
    legend.position = "bottom")
```



1. **Most Common Milks Used**

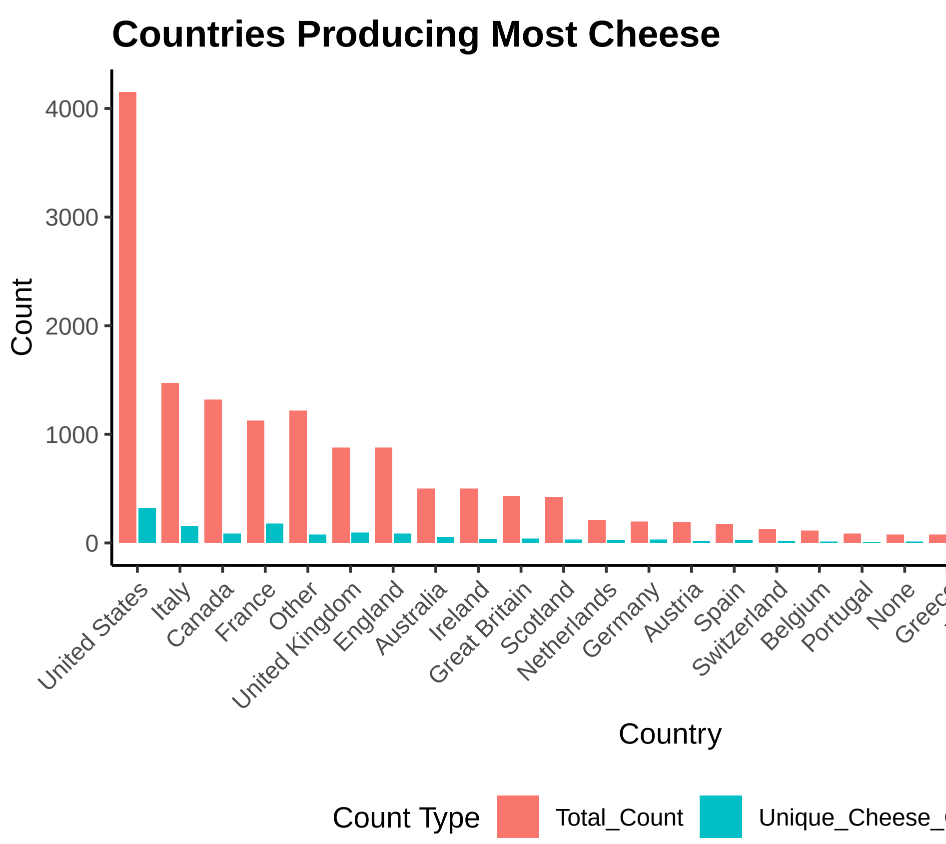
```
df |>
  select(cheese_id, texture_id) |>
  left_join(cheese_df, by = join_by(cheese_id)) |>
  left_join(texture_df, by = join_by(texture_id)) |>
  mutate(texture = str_to_title(texture)) |>
  group_by(texture) |>
  summarise(
    Total_Count = n(),
    Unique_Cheese_Count = n_distinct(cheese)) |>
  pivot_longer(cols = c(Total_Count, Unique_Cheese_Count)) |>
  ggplot(aes(x = reorder(texture, -value), y = value, fill = name)) +
  geom_col(position = position_dodge2(width = 0.2, preserve = "single")) +
  labs(
    title = "Most Common Textures Associated with Cheeses",
    x = "Texture",
    y = "Count",
    fill = 'Count Type'
  ) +
  theme_classic() +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust=1, size = 8),
    plot.title = element_text(size = 14, face = "bold"),
    legend.position = "bottom")
```



2. Most Common Textures Associated with Cheeses

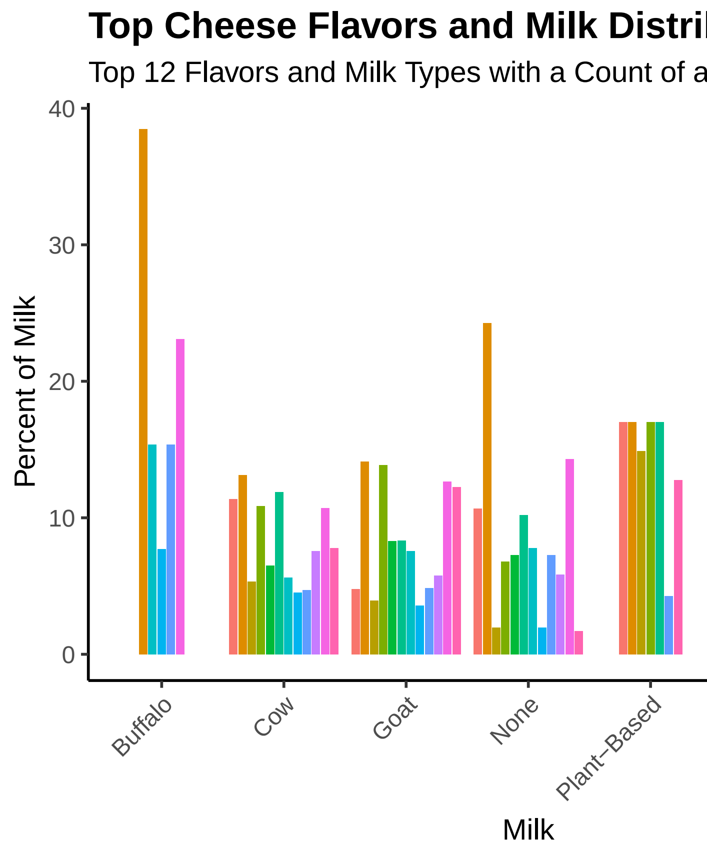
```
df |>
  select(cheese_id, country_id) |>
```

```r
left_join(cheese_df, by = join_by(cheese_id)) |>
left_join(country_df, by = join_by(country_id)) |>
mutate(country = str_to_title(country)) |>
group_by(country) |>
summarise(
  Total_Count = n(),
  Unique_Cheese_Count = n_distinct(cheese)) |>
mutate(country = ifelse(Unique_Cheese_Count <= 5, 'Other', country)) |>
group_by(country) |>
summarise(
  Total_Count = sum(Total_Count),
  Unique_Cheese_Count = sum(Unique_Cheese_Count)) |>
pivot_longer(cols = c(Total_Count, Unique_Cheese_Count)) |>
ggplot(aes(x = reorder(country, -value), y = value, fill = name)) +
geom_col(position = position_dodge2(width = 0.2, preserve = "single")) +
labs(
  title = "Countries Producing Most Cheese",
  x = "Country",
  y = "Count",
  fill = 'Count Type'
) +
theme_classic() +
theme(
  axis.text.x = element_text(angle = 45, vjust = 1, hjust=1),
  plot.title = element_text(size = 14, face = "bold"),
  legend.position = "bottom")
```



3. **Countries Producing Most Cheese**

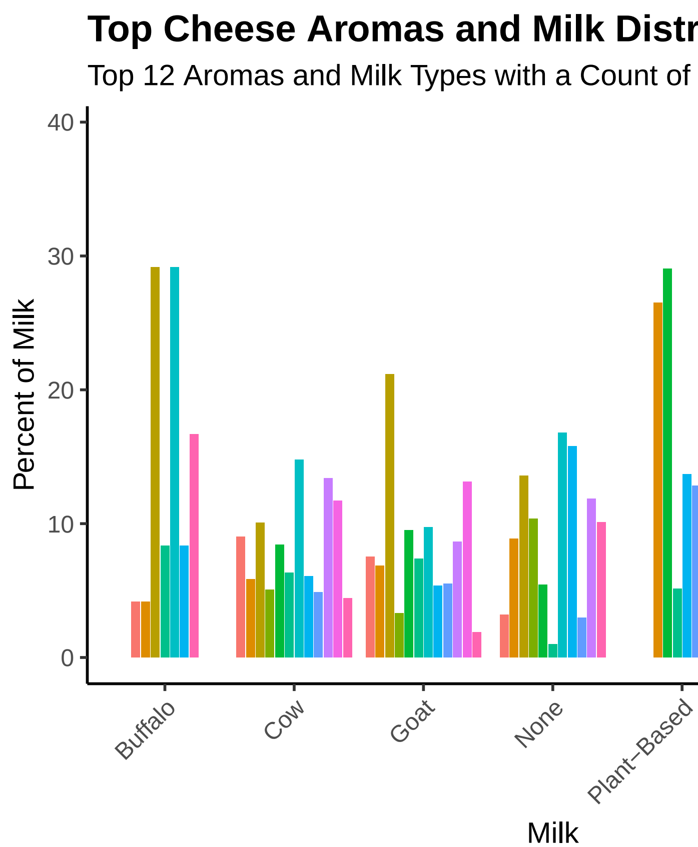
```r
top_flavor_df = df |>
  left_join(cheese_df, by = join_by(cheese_id)) |>
  left_join(flavor_df, by = join_by(flavor_id)) |>
  group_by(flavor, flavor_id) |>
  summarise(cnt = n(), .groups = 'keep') |>
  arrange(desc(cnt)) |>
  head(12)

milk_cnt_df = df |>
  left_join(cheese_df, by = join_by(cheese_id)) |>
  inner_join(top_flavor_df, by = join_by(flavor_id)) |>
  left_join(milk_df, by = join_by(milk_id)) |>
  group_by(milk, milk_id) |>
  summarise(milk_cnt = n(), .groups = 'keep') |>
  filter(milk_cnt>10)

df |>
  select(cheese_id, milk_id, flavor_id) |>
  left_join(cheese_df, by = join_by(cheese_id)) |>
  inner_join(top_flavor_df, by = join_by(flavor_id)) |>
  inner_join(milk_cnt_df, by = join_by(milk_id)) |>
  mutate(
    flavor = str_to_title(flavor),
    milk = str_to_title(milk)
    ) |>
  group_by(milk) |>
  mutate(y= n()) |>
  group_by(milk, flavor) |>
  mutate(x = n()) |>
  mutate(Unique_Cheese_Prct =  (x / y)*100) |>
  select(milk, flavor, Unique_Cheese_Prct, x, milk_cnt, y) |>
  distinct() |>
  ggplot(aes(x = milk, y = Unique_Cheese_Prct, fill = flavor)) +
  geom_col(position = position_dodge2(width = 0.3, preserve = "single")) +
  labs(
    title = "Top Cheese Flavors and Milk Distribution",
    subtitle = 'Top 12 Flavors and Milk Types with a Count of at Least 10',
    x = "Milk",
    y = "Percent of Milk",
    fill = 'Flavor'
  ) +
  theme_classic() +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust=1),
    plot.title = element_text(size = 14, face = "bold"))
```



4a. Common Flavors Across Cheeses By Different Milks

```
top_aroma_df = df |>
  left_join(cheese_df, by = join_by(cheese_id)) |>
  left_join(aroma_df, by = join_by(aroma_id)) |>
  group_by(aroma, aroma_id) |>
  summarise(cnt = n(), .groups = 'keep') |>
  arrange(desc(cnt)) |>
  head(12)

df |>
  select(cheese_id, milk_id, aroma_id) |>
  left_join(cheese_df, by = join_by(cheese_id)) |>
  inner_join(top_aroma_df, by = join_by(aroma_id)) |>
  inner_join(milk_cnt_df, by = join_by(milk_id)) |>
  mutate(
    aroma = str_to_title(aroma),
    milk = str_to_title(milk)
    ) |>
  group_by(milk) |>
  mutate(y= n()) |>
  group_by(milk, aroma) |>
  mutate(x = n()) |>
  mutate(Unique_Cheese_Prct =  (x / y)*100) |>
  select(milk, aroma, Unique_Cheese_Prct, x, milk_cnt, y) |>
  distinct() |>
  ggplot(aes(x = milk, y = Unique_Cheese_Prct, fill = aroma)) +
```

```r
  geom_col(position = position_dodge2(width = 0.3, preserve = "single")) +
  labs(
    title = "Top Cheese Aromas and Milk Distribution",
    subtitle = 'Top 12 Aromas and Milk Types with a Count of at Least 10',
    x = "Milk",
    y = "Percent of Milk",
    fill = 'Aroma'
  ) +
  theme_classic() +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 1, hjust=1),
    plot.title = element_text(size = 14, face = "bold"))
```



4b. Common Aromas Across Cheeses By Different Milks

Exporting Processed Data

```r
# Export cleaned Emissions dataset
write.csv(yearly_emissions_by_area, "yearly_emissions_by_area_cleaned.csv", row.names = FALSE)
write.csv(yearly_emissions, "yearly_emissions_cleaned.csv", row.names = FALSE)
write.csv(emissions_by_area, "emissions_by_area_cleaned.csv", row.names = FALSE)

# Export cleaned DART water permit dataset
write.csv(tbl1_permit_lvl, "tbl1_permit_lvl_cleaned.csv", row.names = FALSE)
write.csv(tbl2_permit_act_lvl, "tbl2_permit_act_lvl_cleaned.csv", row.names = FALSE)
write.csv(tbl3_facility_lvl, "tbl3_facility_lvl_cleaned.csv", row.names = FALSE)
write.csv(tbl4_app_lvl, "tbl4_app_lvl_cleaned.csv", row.names = FALSE)

# Export cleaned Cheese Quality dataset
write.csv(cheese_df, "cheese_df_cleaned.csv", row.names = FALSE)
write.csv(country_df, "country_df_cleaned.csv", row.names = FALSE)
write.csv(milk_df, "milk_df_cleaned.csv", row.names = FALSE)
write.csv(texture_df, "texture_df_cleaned.csv", row.names = FALSE)
write.csv(aroma_df, "aroma_df_cleaned.csv", row.names = FALSE)
write.csv(flavor_df, "flavor_df_cleaned.csv", row.names = FALSE)
write.csv(df, "final_cheese_data_cleaned.csv", row.names = FALSE)

# Confirm that the files were saved successfully
list.files(pattern = "*.csv")
```

```
##  [1] "all-ages.csv"
##  [2] "aroma_df_cleaned.csv"
##  [3] "cheese_df_cleaned.csv"
##  [4] "country_df_cleaned.csv"
##  [5] "emissions_by_area_cleaned.csv"
##  [6] "final_cheese_data_cleaned.csv"
##  [7] "fixed_file.csv"
##  [8] "flavor_df_cleaned.csv"
##  [9] "milk_df_cleaned.csv"
## [10] "project2-emissions-data.csv"
## [11] "tbl1_permit_lvl_cleaned.csv"
## [12] "tbl2_permit_act_lvl_cleaned.csv"
## [13] "tbl3_facility_lvl_cleaned.csv"
## [14] "tbl4_app_lvl_cleaned.csv"
## [15] "texture_df_cleaned.csv"
## [16] "tournament.csv"
## [17] "Week4_table.csv"
## [18] "week7_data.csv"
## [19] "yearly_emissions_by_area_cleaned.csv"
## [20] "yearly_emissions_cleaned.csv"
```

Why Exporting Matters?

- **Preserving Cleaning Efforts:** Once data transformation is complete, saving the cleaned versions prevents the need to redo preprocessing each time.
- **Improving Reproducibility:** The structured datasets can be shared with other analysts or data scientists for further analysis.
- **Facilitating Advanced Analytics:** The exported `.csv` files are now ready for machine learning models, visualization dashboards, and predictive analytics.

Conclusion

This project focused on transforming and analyzing three diverse datasets, demonstrating the importance of data wrangling techniques in preparing raw information for meaningful insights. By leveraging `tidyr` and `dplyr`, we efficiently cleaned, structured, and transformed the datasets into a tidy format, making them suitable for downstream analysis.

Key Takeaways from Each Dataset:

1. **Emissions Data:**
    - The data was reshaped to a long format, making it easier to analyze changes over time.
    - Trends in emissions were identified, providing insights into pollution levels and their environmental implications.
2. **DART Water Permits Data:**
    - The dataset was transformed to facilitate trend analysis in water permit issuance from 2020 to 2025.
    - Cleaning and standardization helped address inconsistencies, ensuring accurate comparisons across years.
3. **Cheese Quality Data:**
    - The dataset was normalized to separate variables, improving its usability.
    - Various transformations allowed us to explore relationships between cheese characteristics, quality ratings, and production factors.

Overall Insights and Future Applications:

- **Data Preparation Matters:**

The process of tidying and transforming data is crucial for accurate and meaningful analysis. Well-structured data improves efficiency in both visualization and modeling.

- **Standardization & Normalization:**

Converting datasets to tidy formats ensured that values were easily accessible for statistical computation.

- **Potential for Further Analysis:**

These datasets can now be used for deeper predictive modeling, trend forecasting, and policy recommendations based on their respective domains.

Through this project, we reinforced the significance of data wrangling techniques in real-world data science applications. The ability to tidy, transform, and analyze raw data is a fundamental skill that enhances decision-making and unlocks valuable insights across different industries.