Practicum1

Thomas Faria

2024-06-05

Part 1: Questions

Question 1

```
# Define dataframe
doc_df <- data.frame(
  doctor_type = c("PCP", "Psychiatrist", "Surgeon", "Anesthesia"),
  doctor_lastname = c("Smith", "Dame", "Jones", "Zayas"),
  location = c("MA", "ME", "NH", "VT"),
  AVG_Rating = c("7", "9", "8", "9")
)
print(doc_df)
```

```
##    doctor_type doctor_lastname location AVG_Rating
## 1          PCP           Smith       MA          7
## 2 Psychiatrist            Dame       ME          9
## 3      Surgeon           Jones       NH          8
## 4   Anesthesia           Zayas       VT          9
```

Question 2

```
# Index with brackets
doc_df[1, 2]
```

```
## [1] "Smith"
```

```
doc_df[2:4, ]
```

```
##    doctor_type doctor_lastname location AVG_Rating
## 2 Psychiatrist            Dame       ME          9
## 3      Surgeon           Jones       NH          8
## 4   Anesthesia           Zayas       VT          9
```

```
doc_df[, 4]
```

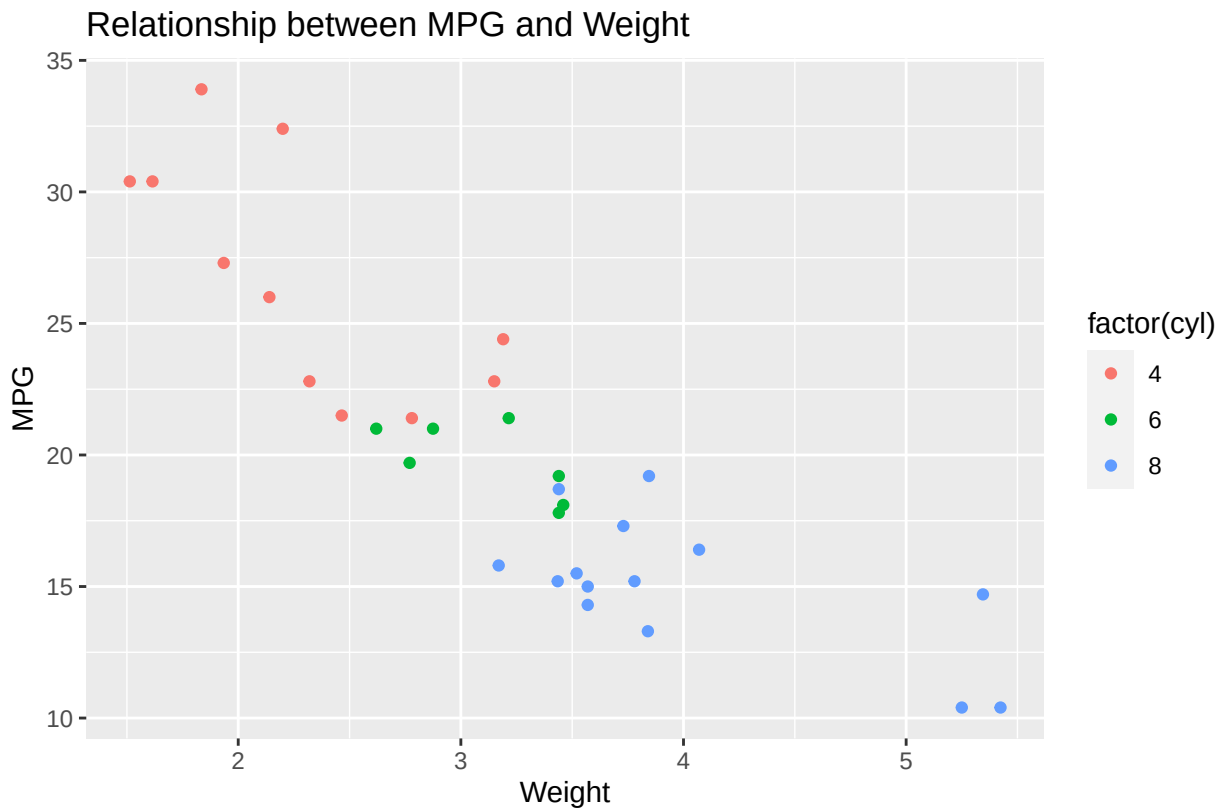
```
## [1] "7" "9" "8" "9"
```

Question 3

```
library(ggplot2)
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
ggplot(mtcars,aes(x=wt, y=mpg, color=factor(cyl))) + geom_point() + labs(x="Weight", y="MPG", title = "
```



This graph looks at the relationship between MPG and weight using cylinders as the color scheme.

Question 4

```
# Examine variable summary stats
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
```

```
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
mpg_weight_cor <- mtcars %>%
select(wt, mpg) %>%
drop_na() %>%
summarize(correlation = cor(wt, mpg))
mpg_weight_cor
```

```
##   correlation
## 1  -0.8676594
```

- It is known that the weight of a vehicle plays a key role in how fuel efficient it is, so both wt and mpg were selected to identify a potential correlation
- The Pearson coefficient is a descriptive statistic that reveals the linear correlation between two variables
- A value between 0 and 1 shows how strong the correlation is, with 1 indicating a strong correlation and 0 an absence of correlation
- The sign of the value indicates the type of correlation (positive or negative)
- In this case, the coefficient value returned is approximately -0.87, which indicates a strong, negative correlation
    - As vehicle weight increases, fuel efficiency (mpg) decreases

---

Part 2: Practicum Tasks

```
# Documentation from data.world recommends package installation directly from Github
devtools::install_github("datadotworld/data.world-r", build_vignettes = TRUE)
```

Load data from provided URL

```
## Using GitHub PAT from the git credential store.

## Skipping install of 'data.world' from a github remote, the SHA1 (a1fd7656) has not changed since last
##   Use 'force = TRUE' to force installation
```

```
# Load the requisite API token obtained from data.world advanced settings (Thomas's account)
# Original code: token <- readLines('~/RStudioProjects/Summer24_DA5020_Group7_Practicum1/API_token')
# Changed
token <- readLines('API_token')
saved_cfg <- data.world::save_config(token)
data.world::set_config(saved_cfg)

# From data.world R and RStudio integration:
library("data.world")
```

```
## Loading required package: dwapi

##
## Attaching package: 'dwapi'

## The following object is masked from 'package:usethis':
##
##     create_project

## The following object is masked from 'package:dplyr':
##
##     sql
```

```
sql_stmt <- data.world::qry_sql("SELECT * FROM chemical_dependence_treatment_program_admissions_beginni
query_results_df <- data.world::query(
  sql_stmt, "https://data.world/data-ny-gov/ngbt-9rwf")
```

```
## Rows: 72463 Columns: 7

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): county_of_program_location, program_category, service_type, age_gro...
## dbl (2): year, admissions
##
## 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.
```

Initial evaluation of dataset

- Determine necessary preparation steps and perform them
- Discuss distribution, outliers, and prepare summary stats

```r
# Evaluate data distribution, outliers, and prepare summary stats

# Reassign query results to more descriptive variable
admissions_data <- query_results_df

# Overview data
glimpse(admissions_data)
```

```
## Rows: 72,463
## Columns: 7
## $ year                       <dbl> 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2~
## $ county_of_program_location <chr> "Albany", "Albany", "Albany", "Albany", "Al~
## $ program_category           <chr> "Crisis", "Crisis", "Crisis", "Crisis", "Cr~
## $ service_type               <chr> "Medically Managed Detoxification", "Medica~
## $ age_group                  <chr> "18 thru 24", "18 thru 24", "18 thru 24", "~
## $ primary_substance_group    <chr> "Alcohol", "All Others", "Cocaine incl Crac~
## $ admissions                 <dbl> 25, 7, 1, 64, 20, 140, 10, 4, 244, 63, 230,~
```

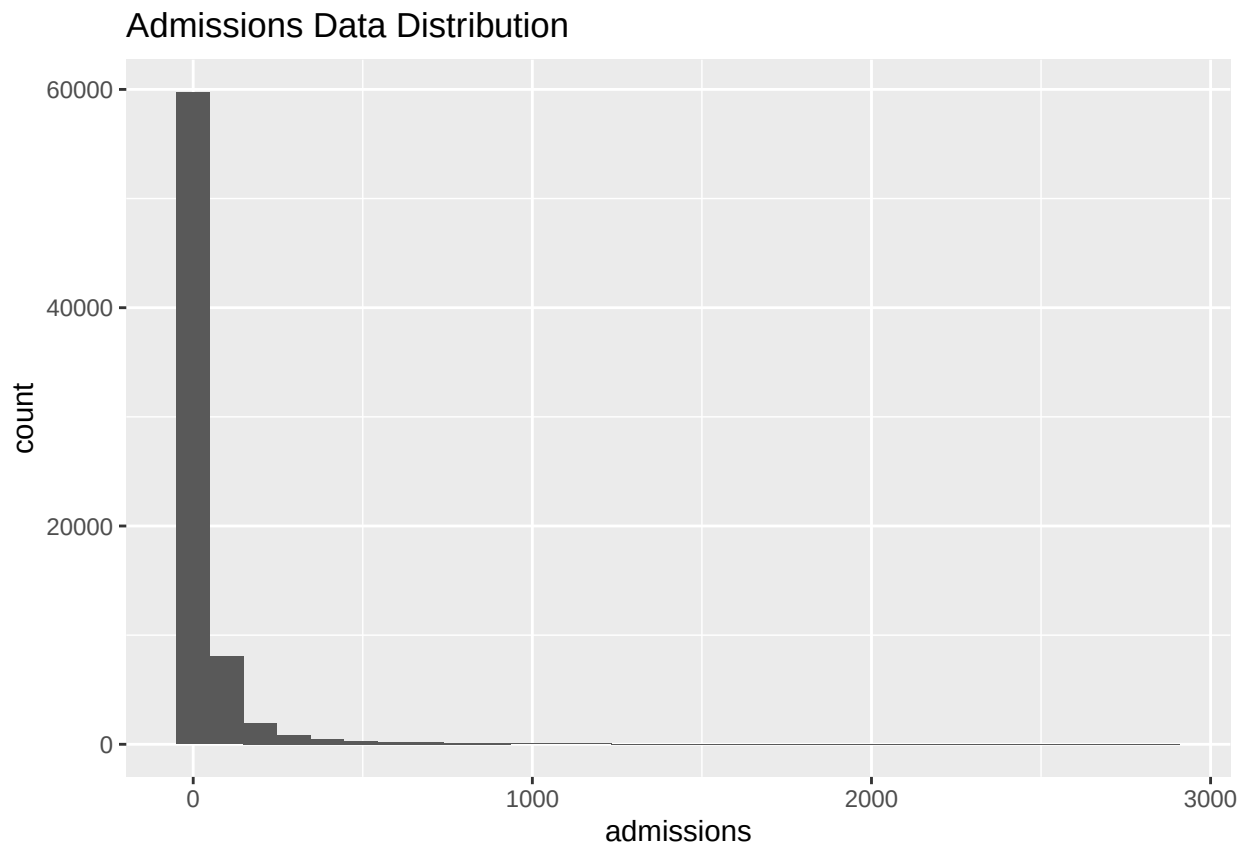
```r
summary(admissions_data)
```

```
##       year      county_of_program_location program_category
##  Min.   :2007   Length:72463               Length:72463
##  1st Qu.:2009   Class :character           Class :character
##  Median :2012   Mode  :character           Mode  :character
##  Mean   :2012
##  3rd Qu.:2015
##  Max.   :2017
##  service_type        age_group         primary_substance_group
##  Length:72463       Length:72463       Length:72463
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##    admissions
##  Min.   :   1.00
##  1st Qu.:   3.00
##  Median :   8.00
##  Mean   :  44.62
##  3rd Qu.:  30.00
##  Max.   :2862.00
```

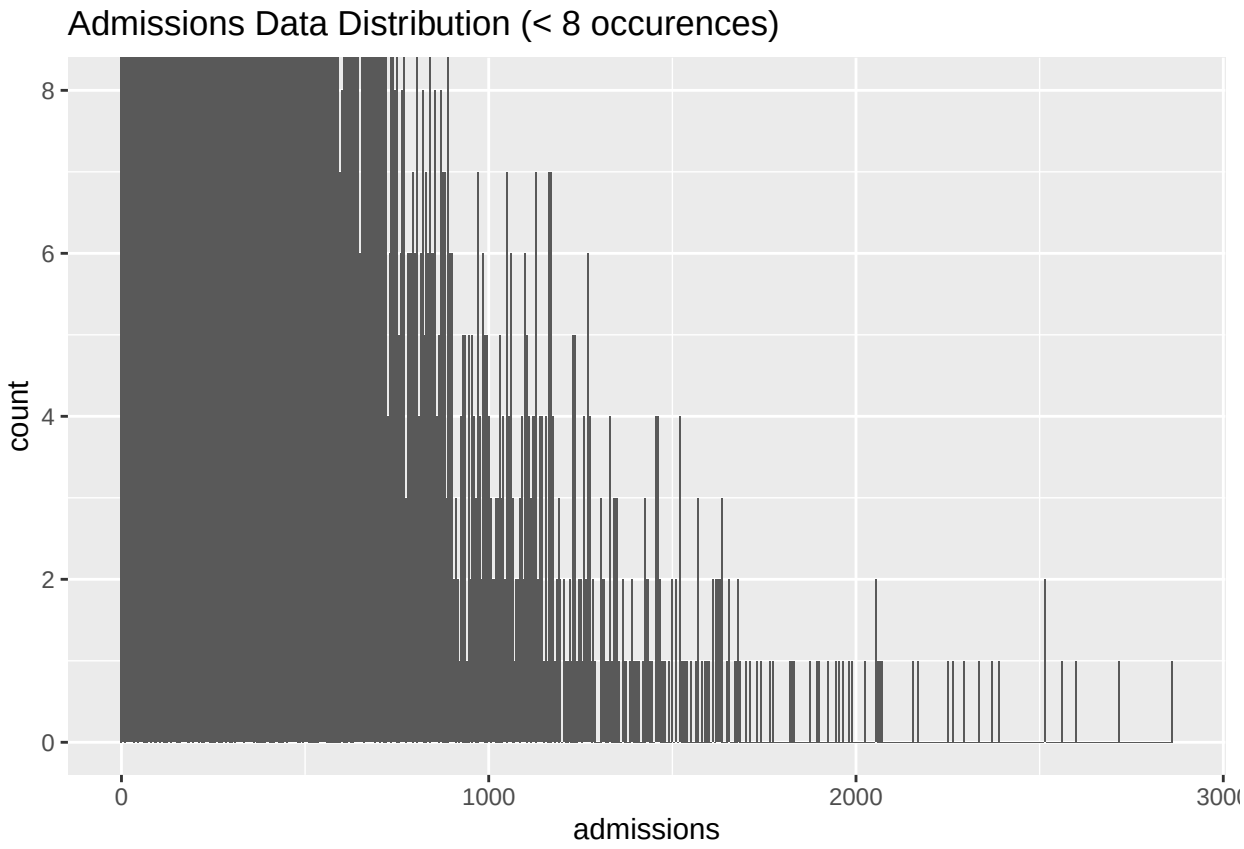
```r
# Visualize outliers in the admissions column
ggplot(admissions_data) +
  labs(title = "Admissions Data Distribution") +
  geom_histogram(mapping = aes(x = admissions), bindwidth = 5) +
  scale_y_continuous()
```

```
## Warning: Ignoring unknown parameters: bindwidth
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Zoom in on low occuring values
ggplot(admissions_data) +
  labs(title = "Admissions Data Distribution (< 8 occurences)") +
  geom_histogram(mapping = aes(x = admissions), binwidth = 5) +
  coord_cartesian(ylim = c(0,8))
```




```r
# Designate all columns except for year, admissions, and county as categorical (using as.factor())
admissions_data_factors <- admissions_data %>%
  mutate(
    program_category = as.factor(program_category),
    service_type = as.factor(service_type),
    age_group = as.factor(age_group),
    primary_substance_group = as.factor(primary_substance_group)
  )

# Compute summaries per factor
program_category_summary <- admissions_data_factors %>%
  group_by(program_category) %>%
  summarize(
    min_admissions = min(admissions),
    median_admissions = median(admissions),
    mean_admissions = mean(admissions),
    max_admissions = max(admissions)
  )
print(program_category_summary)
```

```
## # A tibble: 5 x 5
##   program_category         min_admissions median_admissions mean_admis~1 max_a~2
##   <fct>                             <dbl>             <dbl>        <dbl>   <dbl>
## 1 Crisis                                1                15         67.2    2862
## 2 Inpatient                             1                15         43.8    1106
```

```
## 3 Opioid Treatment Program              1                 9         55.4    1582
## 4 Outpatient                            1                11         56.1    1876
## 5 Residential                           1                 3         11.7     516
## # ... with abbreviated variable names 1: mean_admissions, 2: max_admissions
```

```
service_type_summary <- admissions_data_factors %>%
  group_by(service_type) %>%
  summarize(
    min_admissions = min(admissions),
    median_admissions = median(admissions),
    mean_admissions = mean(admissions),
    max_admissions = max(admissions)
  )
print(service_type_summary)
```

```
## # A tibble: 28 x 5
##    service_type                    min_admissions median_admi~1 mean_~2 max_a~3
##    <fct>                                    <dbl>         <dbl>   <dbl>   <dbl>
##  1 Community Residential                        1             3    6.70     143
##  2 Inpatient Rehabilitation                     1            15   43.8     1106
##  3 Intensive Residential                        1             7   25.0      516
##  4 Limited Outpatient/KEEP                      1             7   15.0      151
##  5 Long Term Res CD/Youth                       1             2    4.79      31
##  6 Med Sup Withdrawal - Inpatient               1            14   70.6     2058
##  7 Med Sup Withdrawal - Outpatient              1             8   32.1      341
##  8 Medically Managed Detoxification             1            19   95.1     2862
##  9 Medically Monitored Withdrawal               1            13   38.9     2516
## 10 Meth to Abst - Residential                   1            16   27.4       79
## # ... with 18 more rows, and abbreviated variable names 1: median_admissions,
## #   2: mean_admissions, 3: max_admissions
```

```
age_group_summary <- admissions_data_factors %>%
  group_by(age_group) %>%
  summarize(
    min_admissions = min(admissions),
    median_admissions = median(admissions),
    mean_admissions = mean(admissions),
    max_admissions = max(admissions)
  )
print(age_group_summary)
```

```
## # A tibble: 6 x 5
##   age_group    min_admissions median_admissions mean_admissions max_admissions
##   <fct>                 <dbl>             <dbl>           <dbl>          <dbl>
## 1 18 thru 24                1                 8            31.3           1518
## 2 25 thru 34                1                13            52.9           1876
## 3 35 thru 44                1                10            53.1           2862
## 4 45 thru 54                1                 8            58.8           2716
## 5 55 and Older              1                 5            30.0           1277
## 6 Under 18                  1                 4            23.5            661
```

```r
primary_substance_group_summary <- admissions_data_factors %>%
  group_by(primary_substance_group) %>%
  summarize(
    min_admissions = min(admissions),
    median_admissions = median(admissions),
    mean_admissions = mean(admissions),
    max_admissions = max(admissions)
  )
print(primary_substance_group_summary)
```

```
## # A tibble: 6 x 5
##   primary_substance_group min_admissions median_admissions mean_admiss~1 max_a~2
##   <fct>                            <dbl>              <dbl>         <dbl>   <dbl>
## 1 Alcohol                              1                 21          91.6    2862
## 2 All Others                           1                  3           9.92    341
## 3 Cocaine incl Crack                   1                  7          29.3    1489
## 4 Heroin                               1                 13          55.0    1582
## 5 Marijuana incl Hashish               1                  8          46.0    1876
## 6 Other Opioids                        1                  6          15.4     672
## # ... with abbreviated variable names 1: mean_admissions, 2: max_admissions
```

```r
# Compute outliers for admissions
admissions_outliers <- admissions_data_factors %>%
  mutate(
    mean_admissions = mean(admissions, na.rm = TRUE),
    sd_admissions = sd(admissions, na.rm = TRUE)
  ) %>%
  # Relative to the mean, any values on the lower or upper bounds that are 3 times the standard deviati
  filter(admissions < mean_admissions - 3 * sd_admissions | admissions > mean_admissions + 3 * sd_admiss
  select(admissions)
admissions_outliers
```

```
## # A tibble: 1,380 x 1
##    admissions
##         <dbl>
##  1        526
##  2        468
##  3        515
##  4        501
##  5        752
##  6        496
##  7        566
##  8        442
##  9        564
## 10        469
## # ... with 1,370 more rows
```

```r
# Remove outliers from dataset
rmv_admissions_outliers <- admissions_data_factors %>%
  mutate(
    mean_admissions = mean(admissions, na.rm = TRUE),
    sd_admissions = sd(admissions, na.rm = TRUE)
```

```
  ) %>%
  filter(!(admissions < mean_admissions - 3 * sd_admissions | admissions > mean_admissions + 3 * sd_admi
# Note subtracted outliers from new dataframe
str(admissions_data_factors)
```

```
## tibble [72,463 x 7] (S3: tbl_df/tbl/data.frame)
##  $ year                       : num [1:72463] 2017 2017 2017 2017 2017 ...
##  $ county_of_program_location: chr [1:72463] "Albany" "Albany" "Albany" "Albany" ...
##  $ program_category           : Factor w/ 5 levels "Crisis","Inpatient",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ service_type               : Factor w/ 28 levels "Community Residential",..: 8 8 8 8 8 8 8 8 8 8 .
##  $ age_group                  : Factor w/ 6 levels "18 thru 24","25 thru 34",..: 1 1 1 1 1 2 2 2 2 2
##  $ primary_substance_group    : Factor w/ 6 levels "Alcohol","All Others",..: 1 2 3 4 6 1 2 3 4 6 ...
##  $ admissions                 : num [1:72463] 25 7 1 64 20 140 10 4 244 63 ...
```

```
str(rmv_admissions_outliers$admissions)
```

```
##  num [1:71083] 25 7 1 64 20 140 10 4 244 63 ...
```

```
# Read in .csv created from https://www.dot.ny.gov/main/business-center/engineering/specifications/loca
county_codes <- read_csv("county_codes.csv")
```

Restructure data into appropriate tibbles

```
## Rows: 62 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): county_of_program_location, county_code
##
## 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.
```

```
county_codes
```

```
## # A tibble: 62 x 2
##    county_of_program_location county_code
##    <chr>                      <chr>
##  1 Albany                     AL
##  2 Cattaraugus                CA
##  3 Chenango                   CN
##  4 Delaware                   DE
##  5 Franklin                   FR
##  6 Hamilton                   HA
##  7 Lewis                      LE
##  8 Montgomery                 MG
##  9 Oneida                     ON
## 10 Orleans                    OL
## # ... with 52 more rows
```

```r
county <- admissions_data_factors %>%
  select(county_of_program_location) %>%
  distinct() %>%
  # Join codes with respective counties from county_codes
  left_join(county_codes, by = "county_of_program_location") %>%
  mutate(county_code = case_when(
      # Tagging counties with first two characters and "-NYC"
      county_of_program_location %in% c("Bronx", "Queens", "Kings") ~ paste(str_to_upper(str_sub(county
      county_of_program_location == "New York" ~ "NYC",
      # Handle to not treat as NA value, changed code to NS instead
      county_of_program_location == "Nassau" ~ "NS",
      # Handle to not treat as NA value, manually assigned SL
      county_of_program_location == "St Lawrence" ~ "SL",
      TRUE ~ county_code
    )
  )
# Note: The county "Hamilton" is included in the county_codes csv, but it is not found in the admission
county
```

```
## # A tibble: 61 x 2
##    county_of_program_location county_code
##    <chr>                      <chr>
##  1 Albany                     AL
##  2 Bronx                      BR-NYC
##  3 Broome                     BM
##  4 Dutchess                   DU
##  5 Erie                       ER
##  6 Kings                      KI-NYC
##  7 Monroe                     MO
##  8 Nassau                     NS
##  9 New York                   NYC
## 10 Niagara                    NI
## # ... with 51 more rows
```

```r
# Define abbreviations for recoding
program_category_index <- c(
  "Crisis" = "C",
  "Inpatient" = "I",
  "Opioid Treatment Program" = "OTP",
  "Outpatient" = "O",
  "Residential" = "R"
)

# Ceate new column called program_code
admissions_data_coded <- admissions_data_factors %>%
  mutate(program_code = recode(program_category,
       "Crisis" = "C",
       "Inpatient" = "I",
       "Opioid Treatment Program" = "OTP",
       "Outpatient" = "O",
       "Residential" = "R"))

# Create program_category tibble and recode based on the index directly
```

```r
program_category_df <- admissions_data_coded %>%
  distinct(program_category, .keep_all = TRUE) %>%
  select(program_code, program_category)

program_category_df
```

```
## # A tibble: 5 x 2
##   program_code program_category
##   <fct>        <fct>
## 1 C            Crisis
## 2 I            Inpatient
## 3 OTP          Opioid Treatment Program
## 4 O            Outpatient
## 5 R            Residential
```

```r
# Sairah
# Define the index for recoding
primary_substance_group_index <- c(
  "Alcohol" = "A",
  "All Others" = "AO",
  "Cocaine incl Crack" = "CC",
  "Heroin" = "H",
  "Marijuana incl Hashish" = "MH",
  "Other Opioids" = "OO"
)

# Update admissions_data_coded with new column called substance_code
admissions_data_coded <- admissions_data_coded %>%
  mutate(substance_code = recode(primary_substance_group,
        "Alcohol" = "A",
        "All Others" = "AO",
        "Cocaine incl Crack" = "CC",
        "Heroin" = "H",
        "Marijuana incl Hashish" = "MH",
        "Other Opioids" = "OO"))

# Create Primary Substance Group tibble and recode based on the index directly
primary_substance_group_df <- admissions_data_coded %>%
  distinct(primary_substance_group, .keep_all = TRUE) %>%
  select(substance_code, primary_substance_group)

primary_substance_group_df
```

```
## # A tibble: 6 x 2
##   substance_code primary_substance_group
##   <fct>          <fct>
## 1 A              Alcohol
## 2 AO             All Others
## 3 CC             Cocaine incl Crack
## 4 H              Heroin
## 5 OO             Other Opioids
## 6 MH             Marijuana incl Hashish
```

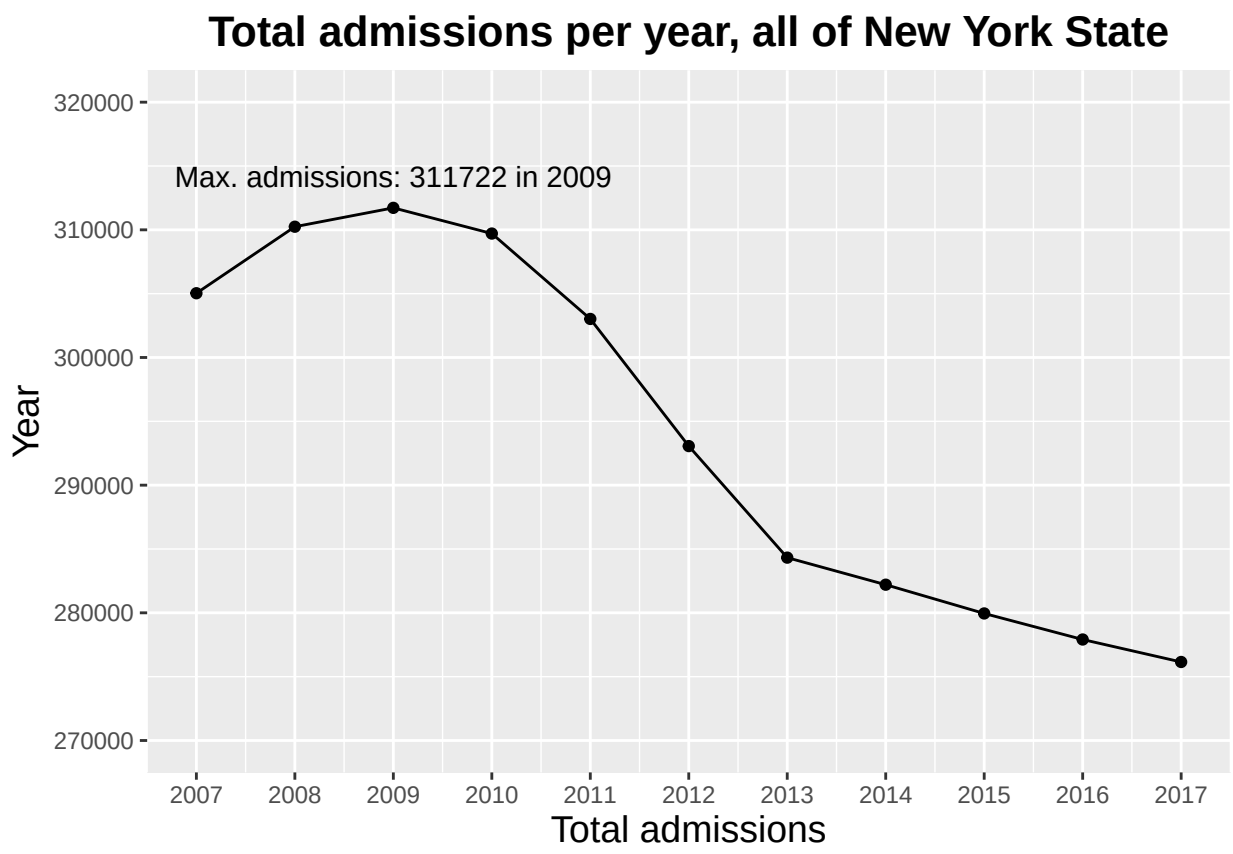
```r
# Thomas

# Join county_code data onto main tibble using a full_join by county name
admissions_data_coded_joined <- admissions_data_coded %>%
  full_join(county, by = "county_of_program_location")

# Final tibble: admissions_data_df
admissions_data_df <- admissions_data_coded_joined %>%
  select(
    year,
    county_code,
    program_code,
    service_type,
    age_group,
    substance_code,
    admissions
  )
```

Define annualAdmissions()

- Function should derive the total # of reported admissions per year for the entire state of NY and display these results on a line graph
- Annotate to show year with highest admissions
- Explain results

```r
# Thomas -- NEEDS WORK

# This function uses aggregate() to sum the total admissions for every year in the admissions_data_df
# The max point is computed from the aggregated tibble and held for later reference on the graph
# A line graph is prepared using ggplot2 with appropriate labeling

annualAdmissions <- function() {
  # Get aggregated data as its own tibble for easy ref
  total_admissions <- aggregate(admissions_data_df$admissions,
            by = list(year = admissions_data_df$year),
            sum) %>%
    rename(total = x)

  # Get maximum
  max_point <- total_admissions[which.max(total_admissions$total), ]

  # Plot a line graph
  total_admissions %>%
    ggplot(mapping = aes(x = year, y = total)) +
    geom_line() +
    geom_point() +
    scale_x_continuous(breaks = 2007:2017) +
    scale_y_continuous(limits = c(270000, 320000)) +
    labs(title = "Total admissions per year, all of New York State",
         x = "Total admissions",
         y = "Year") +
```

```
    theme(
      plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
      axis.title.x = element_text(size = 14),
      axis.title.y = element_text(size = 14)
    ) +
    annotate("text",
             x = max_point$year,
             y = max_point$total,
             label = paste("Max. admissions:", max_point$total, "in", max_point$year),
             vjust = -1)
}

annualAdmissions()
```



Analyze % of admissions by county

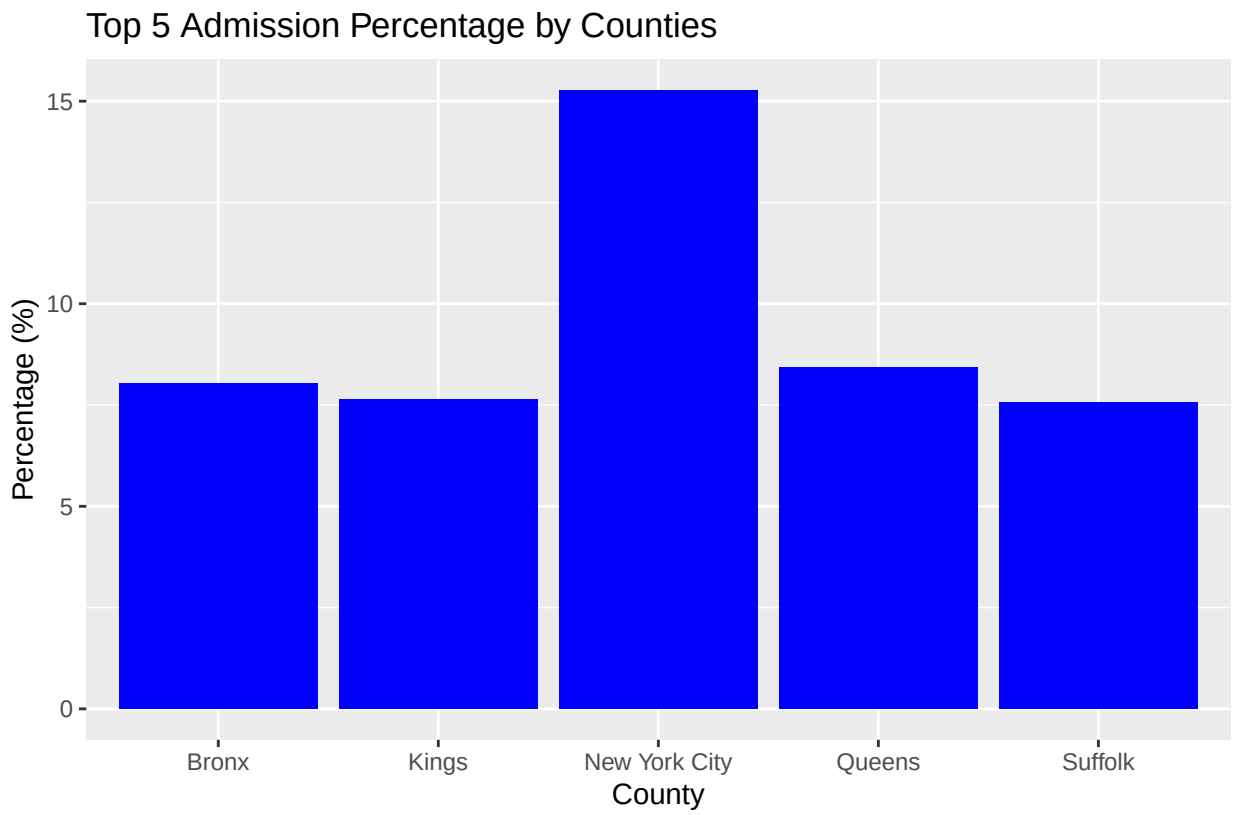
- Visualize top 5 counties using a bar chart
- Explain results

```
# Sairah

#total number of admission in the NYS
total= sum(admissions_data_df$admissions)
```

```r
#calculate percentage of admissions in each county
percentage_admissions <- admissions_data_df %>%
  select(county_code, admissions) %>%
  group_by(county_code) %>%
  summarize(percentage=((sum(admissions)/total) * 100))

top_5_admission_counties <- percentage_admissions %>%
  slice_max(n=5, percentage)

print(top_5_admission_counties)
```

```
## # A tibble: 5 x 2
##   county_code percentage
##   <chr>            <dbl>
## 1 NYC              15.3
## 2 QU-NYC            8.42
## 3 BR-NYC            8.03
## 4 KI-NYC            7.64
## 5 SU                7.57
```

```r
library(ggplot2)

ggplot(top_5_admission_counties, aes(x=county_code, y = percentage)) + geom_col(fill = "blue") +
    labs(x = "County", y= "Percentage (%)", title = "Top 5 Admission Percentage by Counties", caption =
  scale_x_discrete(labels = c("NYC" = "New York City", "QU-NYC" = "Queens", "BR-NYC" = "Bronx", "KI-NYC
```



ties in New York City that had the highest percentage of admissions to the chemical dependence treatment program.

Extract various "Rehab" facilities information

- Use a regex to match all facilities that include the word rehab, rehabilitation, etc.
- Using filtered data, identify the most prominent (common) substance related to admission for each age group
- Visualize and explain results

```
rehab_df <- admissions_data_factors %>%
  # Only show rehabilitation services
  filter(str_detect(service_type, regex("Rehab|Rehabilitation", ignore_case = TRUE))) %>%
  select(service_type, age_group, primary_substance_group, admissions)
rehab_df
```

```
## # A tibble: 17,319 x 4
##    service_type             age_group  primary_substance_group admissions
##    <fct>                    <fct>      <fct>                          <dbl>
##  1 Inpatient Rehabilitation 18 thru 24 Alcohol                           11
##  2 Inpatient Rehabilitation 18 thru 24 All Others                         2
##  3 Inpatient Rehabilitation 18 thru 24 Cocaine incl Crack                 4
##  4 Inpatient Rehabilitation 18 thru 24 Heroin                            21
##  5 Inpatient Rehabilitation 18 thru 24 Marijuana incl Hashish             6
##  6 Inpatient Rehabilitation 18 thru 24 Other Opioids                      5
##  7 Inpatient Rehabilitation 25 thru 34 Alcohol                           49
##  8 Inpatient Rehabilitation 25 thru 34 All Others                         7
##  9 Inpatient Rehabilitation 25 thru 34 Cocaine incl Crack                31
## 10 Inpatient Rehabilitation 25 thru 34 Heroin                           101
## # ... with 17,309 more rows
```

```
top_substance_df <- rehab_df %>%
  # Only interested in these combinations
  group_by(service_type, age_group, primary_substance_group) %>%
  # Take count to show how many admissions exist for each substance in each age group
  summarize(substance_count = sum(admissions)) %>%
  # Limit to age group
  group_by(age_group) %>%
  # Filter for the substances with the highest count
  filter(substance_count == max(substance_count)) %>%
  # Show relevant columns
  select(service_type, age_group, primary_substance_group, substance_count)
```

```
## `summarise()` has grouped output by 'service_type', 'age_group'. You can
## override using the `.groups` argument.
```

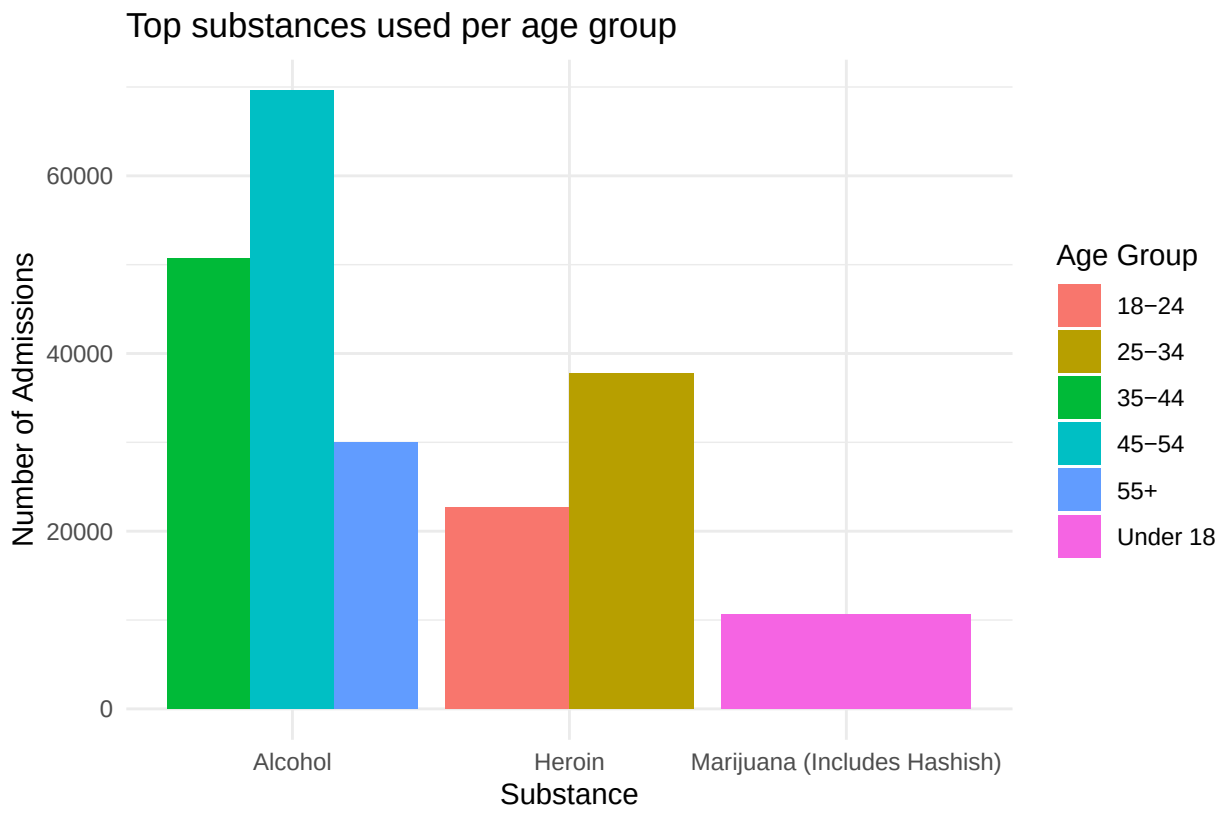
```
top_substance_df
```

```
## # A tibble: 6 x 4
## # Groups:   age_group [6]
##   service_type             age_group    primary_substance_group substance_count
##   <fct>                    <fct>        <fct>                            <dbl>
## 1 Inpatient Rehabilitation 18 thru 24   Heroin                           22705
```

```
## 2 Inpatient Rehabilitation 25 thru 34   Heroin                          37753
## 3 Inpatient Rehabilitation 35 thru 44   Alcohol                        50698
## 4 Inpatient Rehabilitation 45 thru 54   Alcohol                        69590
## 5 Inpatient Rehabilitation 55 and Older Alcohol                        30051
## 6 Res Rehab for Youth      Under 18     Marijuana incl Hashish          10643
```

- To identify the most prominent substance used in each age group, we first define a regular expression in a new dataframe that filters all services containing "Rehab" or "Rehabilitation" in the name
- A separate dataframe is then defined to find the top substance per age group
  - The data is grouped by age_group and primary_substance_group because we are only interested in analyses in the context of these variables paired together

```
ggplot(top_substance_df, aes(primary_substance_group, substance_count, fill = age_group)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_x_discrete(
    name = "Substance",
    labels = c(
      "All Others" = "Other",
      "Cocaine incl Crack" = "Cocaine (Includes Crack)",
      "Marijuana incl Hashish" = "Marijuana (Includes Hashish)"
    )
  ) +
  scale_fill_discrete(
    name = "Age Group",
    labels = c(
      "18 thru 24" = "18-24",
      "25 thru 34" = "25-34",
      "35 thru 44" = "35-44",
      "45 thru 54" = "45-54",
      "55 and Older" = "55+",
      "Under 18" = "Under 18"
    )
  ) +
  labs(
    y = "Number of Admissions",
    title = "Top substances used per age group",
    caption = "Substances grouped by top use per age group with the number of admissions on the y-axis a
  ) +
  theme(
    axis.title = element_text(face = "bold", color = "black")
  ) +
  theme_minimal()
```



ıge group with the number of admissions on the y−axis and the type of substance on the x−axis.

```
all_substance_df <- rehab_df %>%
  group_by(service_type, age_group, primary_substance_group) %>%
  # Take count to show how many admissions exist for each substance in each age group
  summarize(substance_count = sum(admissions)) %>%
  # Limit to age group
  group_by(age_group) %>%
  # Filter for the substances with the highest count
  mutate(substance_count == max(substance_count)) %>%
  # Show relevant columns
  select(service_type, age_group, primary_substance_group, substance_count)
```

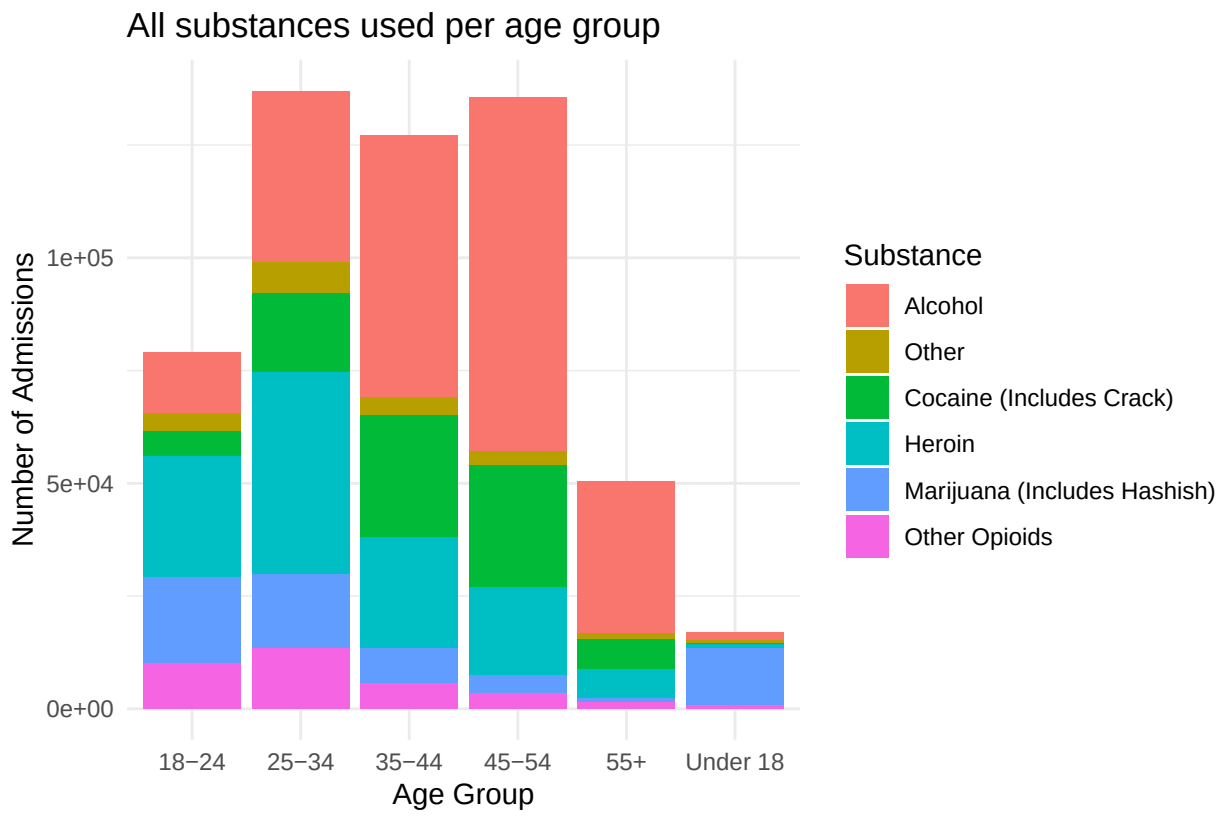
```
## `summarise()` has grouped output by 'service_type', 'age_group'. You can
## override using the `.groups` argument.
```

```
all_substance_df
```

```
## # A tibble: 236 x 4
## # Groups:   age_group [6]
##    service_type             age_group  primary_substance_group substance_count
##    <fct>                    <fct>      <fct>                                <dbl>
##  1 Inpatient Rehabilitation 18 thru 24 Alcohol                              10949
##  2 Inpatient Rehabilitation 18 thru 24 All Others                             3234
##  3 Inpatient Rehabilitation 18 thru 24 Cocaine incl Crack                     4583
##  4 Inpatient Rehabilitation 18 thru 24 Heroin                                22705
##  5 Inpatient Rehabilitation 18 thru 24 Marijuana incl Hashish                10209
```

```
##  6 Inpatient Rehabilitation 18 thru 24 Other Opioids                    8718
##  7 Inpatient Rehabilitation 25 thru 34 Alcohol                         32121
##  8 Inpatient Rehabilitation 25 thru 34 All Others                       5640
##  9 Inpatient Rehabilitation 25 thru 34 Cocaine incl Crack              14559
## 10 Inpatient Rehabilitation 25 thru 34 Heroin                          37753
## # ... with 226 more rows
```

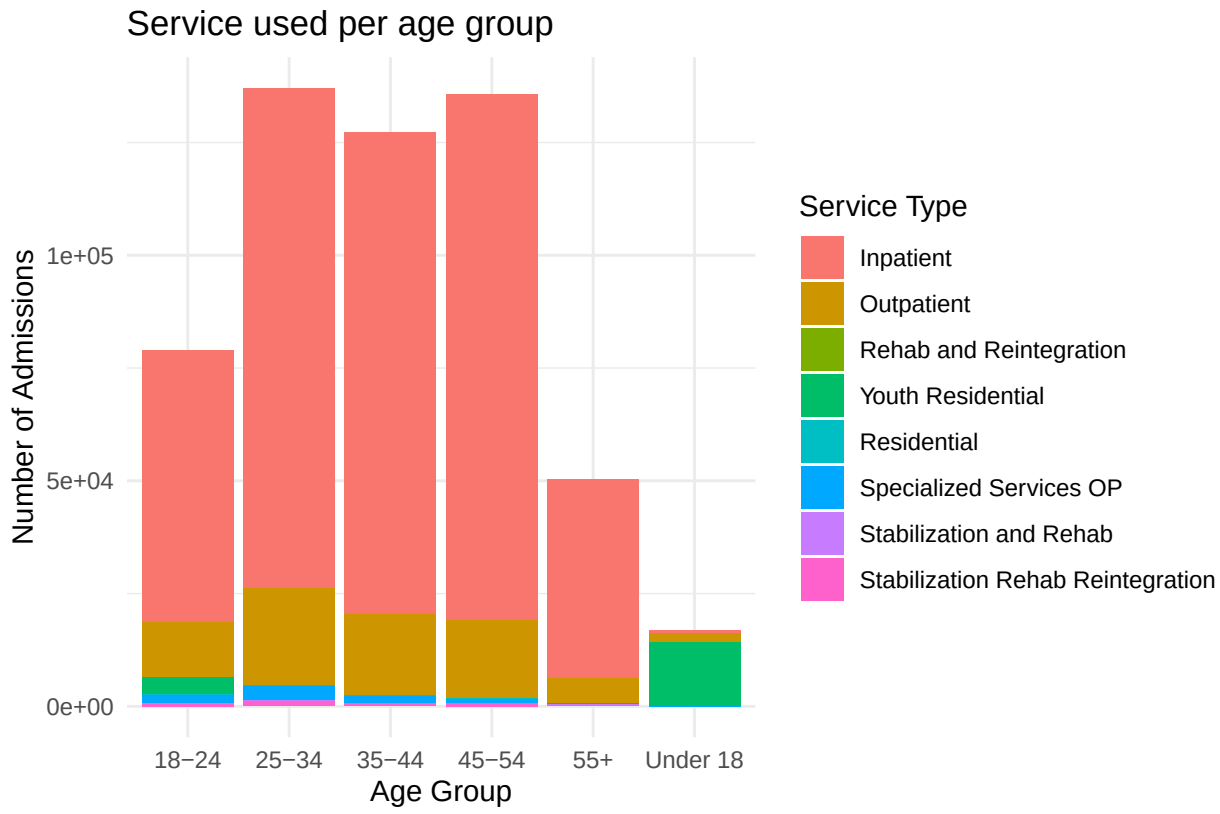
```r
ggplot(all_substance_df, aes(age_group, substance_count, fill = primary_substance_group)) +
  geom_bar(stat = "identity") +
  labs(
    x = "Age Group",
    y = "Number of Admissions",
    title = "All substances used per age group",
    caption = "Stacked breakdown of all substances used per age group with the number of admissions on t
  ) +
  scale_fill_discrete(
    name = "Substance",
    labels = c(
      "All Others" = "Other",
      "Cocaine incl Crack" = "Cocaine (Includes Crack)",
      "Marijuana incl Hashish" = "Marijuana (Includes Hashish)"
    )
  ) +
  scale_x_discrete(
    labels = c(
      "18 thru 24" = "18-24",
      "25 thru 34" = "25-34",
      "35 thru 44" = "35-44",
      "45 thru 54" = "45-54",
      "55 and Older" = "55+",
      "Under 18" = "Under 18"
    )
  ) +
  theme(
    axis.title = element_text(face = "bold", color = "black")
  ) +
  theme_minimal()
```



ı the number of admissions on the y−axis and the age groups on the x−axis.

```
ggplot(all_substance_df, aes(age_group, substance_count, fill = service_type)) +
  geom_bar(stat = "identity") +
  labs(
    x = "Age Group",
    y = "Number of Admissions",
    title = "Service used per age group",
    caption = "Stacked breakdown of services used for admissions per age group with number of admissions
  ) +
  scale_fill_discrete(
    name = "Service Type",
    labels = c(
      "Inpatient Rehabilitation" = "Inpatient",
      "Outpatient Rehabilitation" = "Outpatient",
      "Res Rehab for Youth" = "Youth Residential",
      "Residential Rehabilitation" = "Residential",
      "Specialized Services OP Rehab" = "Specialized Services OP"
    )
  ) +
  scale_x_discrete(
    labels = c(
      "18 thru 24" = "18-24",
      "25 thru 34" = "25-34",
      "35 thru 44" = "35-44",
      "45 thru 54" = "45-54",
      "55 and Older" = "55+",
      "Under 18" = "Under 18"
```

```
  )
) +
theme(
  axis.title = element_text(face = "bold", color = "black")
) +
theme_minimal()
```



with number of admissions on the y−axis and age group on the y−axis.