STATS 551 PROJECT

Model Specification

We aim to use a Poisson regression model with Bayesian inference. The total cancer incidence counts are modeled as Poisson-distributed random variables, with the log rate parameter being a linear function of several predictors. The predictors include environmental (AQI) and temporal (year) factors.

```
-- 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 becor

Attaching package: 'zoo'


The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

First, I wish to import the final, consolidated dataset which I have built out of several separate data sets. It requires some pre-processing, which we can do now, and before visualizing this data, we will visualize the state-wise and year-wise distributions of the incidence of cancer.

```
raw_data <- read.csv("final_dataset_consolidated.csv")
colnames(raw_data)
```

```
 [1] "States"
 [2] "avg_max_aqi"
 [3] "avg_x90th_percentile_aqi"
 [4] "avg_median_aqi"
 [5] "avg_days_with_aqi"
 [6] "avg_good_days"
 [7] "avg_moderate_days"
 [8] "avg_unhealthy_for_sensitive_groups_days"
 [9] "avg_unhealthy_days"
[10] "avg_very_unhealthy_days"
[11] "avg_hazardous_days"
```

```
[12] "avg_days_co"
[13] "avg_days_no2"
[14] "avg_days_ozone"
[15] "avg_days_pm2_5"
[16] "avg_days_pm10"
[17] "year"
[18] "Total_Count"
[19] "Total_Population"
[20] "CharacteristicName"
[21] "ResultMeasureValue"
[22] "ResultMeasure.MeasureUnitCode"
[23] "ResultValueTypeName"
[24] "PrecisionValue"
[25] "DataQuality.BiasValue"
[26] "ResultDepthHeightMeasure.MeasureValue"
[27] "ResultDepthHeightMeasure.MeasureUnitCode"
```

```r
cleaned_data <- raw_data %>%
  mutate(across(
    starts_with("avg_") |
    c("Total_Count", "Total_Population", "year"),
    ~ as.numeric(.)
  )) %>%
  filter(
    !is.na(Total_Count) & Total_Count >= 0,
    !is.na(Total_Population) & Total_Population > 0,
    !is.na(year) & year >= 1999
  ) %>%
  # Check for outliers by identifying extreme values
  filter(
    avg_max_aqi <= 500,  # Cap the AQI values at reasonable levels (e.g., 500)
    avg_days_with_aqi <= 365  # Cap days of AQI at 365, prevent unrealistic values
  ) %>%
  # Remove unnecessary columns with blank or redundant data
  select(
    -contains("CharacteristicName"),
    -contains("ResultMeasure"),
    -contains("ResultValueTypeName"),
    -contains("PrecisionValue"),
    -contains("DataQuality.BiasValue"),
    -contains("ResultDepthHeightMeasure")
  ) %>%
  group_by(States, year) %>%
  summarise(across(everything(), \(x) mean(x, na.rm = TRUE)))
```

```
`summarise()` has grouped output by 'States'. You can override using the
`.groups` argument.
```

```r
# Summary of the dataset
summary(cleaned_data)
```

```
   States               year       avg_max_aqi     avg_x90th_percentile_aqi
 Length:1128        Min.   :1999   Min.   : 58.25   Min.   : 30.50
 Class :character   1st Qu.:2004   1st Qu.:102.03   1st Qu.: 56.50
 Mode  :character   Median :2010   Median :120.12   Median : 62.78
                    Mean   :2010   Mean   :128.29   Mean   : 66.15
                    3rd Qu.:2016   3rd Qu.:144.96   3rd Qu.: 73.94
                    Max.   :2021   Max.   :466.53   Max.   :126.62
 avg_median_aqi  avg_days_with_aqi avg_good_days    avg_moderate_days
 Min.   :15.25   Min.   :113.2     Min.   : 64.57   Min.   :  5.50
 1st Qu.:35.72   1st Qu.:258.3     1st Qu.:161.11   1st Qu.: 64.38
 Median :40.90   Median :290.8     Median :198.34   Median : 85.60
 Mean   :40.27   Mean   :289.0     Mean   :196.17   Mean   : 85.28
 3rd Qu.:44.85   3rd Qu.:329.4     3rd Qu.:228.71   3rd Qu.:105.00
 Max.   :57.67   Max.   :365.0     Max.   :356.25   Max.   :178.00
 avg_unhealthy_for_sensitive_groups_days avg_unhealthy_days
 Min.   : 0.000                          Min.   : 0.00000
 1st Qu.: 1.348                          1st Qu.: 0.05556
 Median : 3.690                          Median : 0.31415
 Mean   : 6.134                          Mean   : 1.22877
 3rd Qu.: 8.912                          3rd Qu.: 1.27976
 Max.   :36.911                          Max.   :17.27778
 avg_very_unhealthy_days avg_hazardous_days  avg_days_co       avg_days_no2
 Min.   :0.00000         Min.   :0.0000     Min.   : 0.0000   Min.   : 0.0000
 1st Qu.:0.00000         1st Qu.:0.0000     1st Qu.: 0.0000   1st Qu.: 0.6986
 Median :0.00000         Median :0.0000     Median : 0.1379   Median : 3.0217
 Mean   :0.14227         Mean   :0.0287     Mean   : 3.1678   Mean   : 8.0224
 3rd Qu.:0.07404         3rd Qu.:0.0000     3rd Qu.: 1.7036   3rd Qu.: 9.1486
 Max.   :5.61538         Max.   :2.6429     Max.   :75.1333   Max.   :93.5000
 avg_days_ozone    avg_days_pm2_5    avg_days_pm10       Total_Count
 Min.   :  9.143   Min.   :  7.643   Min.   :  0.0000   Min.   :   0.0
 1st Qu.:106.348   1st Qu.: 73.217   1st Qu.:  0.5977   1st Qu.:   0.0
 Median :151.043   Median :107.008   Median :  7.6667   Median :  79.0
 Mean   :142.626   Mean   :117.511   Mean   : 17.6592   Mean   : 280.7
 3rd Qu.:181.139   3rd Qu.:152.022   3rd Qu.: 22.8737   3rd Qu.: 318.2
 Max.   :291.000   Max.   :346.000   Max.   :173.5000   Max.   :2475.0
 Total_Population
 Min.   :    7771
 1st Qu.:  227788
 Median :  527199
 Mean   : 3122330
 3rd Qu.: 2349836
 Max.   :39103209
```

```
# Check the structure and column types
str(cleaned_data)
```

```
gropd_df [1,128 x 19] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ States                                : chr [1:1128] "Alabama" "Alabama" "Alabama" "Alaba
 $ year                                  : num [1:1128] 1999 2000 2001 2002 2003 ...
 $ avg_max_aqi                           : num [1:1128] 146 151 137 144 133 ...
```

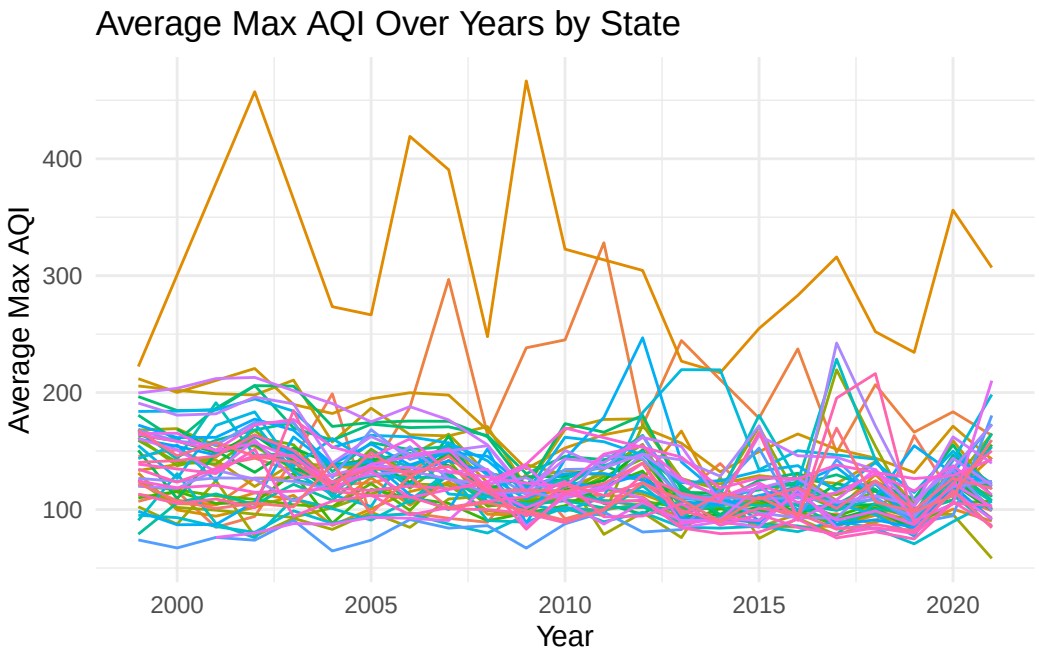
```
$ avg_x90th_percentile_aqi               : num [1:1128] 93.6 96.5 81.5 80.5 74.7 ...
$ avg_median_aqi                         : num [1:1128] 54.4 55.9 48.7 46.3 47.2 ...
$ avg_days_with_aqi                      : num [1:1128] 184 201 219 239 234 ...
$ avg_good_days                          : num [1:1128] 69.8 75.2 111.2 128.8 124.8 ...
$ avg_moderate_days                      : num [1:1128] 88.4 99.1 94.1 95 100.2 ...
$ avg_unhealthy_for_sensitive_groups_days: num [1:1128] 20 22.68 11.38 13.09 8.54 ...
$ avg_unhealthy_days                     : num [1:1128] 5.667 3.773 1.905 1.909 0.542 ...
$ avg_very_unhealthy_days                : num [1:1128] 0.4286 0.3182 0.381 0.0909 0 ...
$ avg_hazardous_days                     : num [1:1128] 0 0 0 0 0 0 0 0 0 0 ...
$ avg_days_co                            : num [1:1128] 2.24 1.77 3.05 2.27 1.62 ...
$ avg_days_no2                           : num [1:1128] 0 0.864 0.238 1.091 0.542 ...
$ avg_days_ozone                         : num [1:1128] 93.2 107.2 122.9 141.5 137.9 ...
$ avg_days_pm2_5                         : num [1:1128] 71.5 81.2 83.3 83.4 85 ...
$ avg_days_pm10                          : num [1:1128] 17.33 10.05 9.48 10.59 8.92 ...
$ Total_Count                            : num [1:1128] 41 51 79 34 53 54 74 66 112 162 ...
$ Total_Population                       : num [1:1128] 194723 220789 442183 378534 474259 ...
- attr(*, "groups")= tibble [50 x 2] (S3: tbl_df/tbl/data.frame)
 ..$ States: chr [1:50] "Alabama" "Alaska" "Arizona" "Arkansas" ...
 ..$ .rows : list<int> [1:50]
 .. ..$ : int [1:23] 1 2 3 4 5 6 7 8 9 10 ...
 .. ..$ : int [1:23] 24 25 26 27 28 29 30 31 32 33 ...
 .. ..$ : int [1:23] 47 48 49 50 51 52 53 54 55 56 ...
 .. ..$ : int [1:22] 70 71 72 73 74 75 76 77 78 79 ...
 .. ..$ : int [1:19] 92 93 94 95 96 97 98 99 100 101 ...
 .. ..$ : int [1:23] 111 112 113 114 115 116 117 118 119 120 ...
 .. ..$ : int [1:22] 134 135 136 137 138 139 140 141 142 143 ...
 .. ..$ : int [1:22] 156 157 158 159 160 161 162 163 164 165 ...
 .. ..$ : int [1:23] 178 179 180 181 182 183 184 185 186 187 ...
 .. ..$ : int [1:23] 201 202 203 204 205 206 207 208 209 210 ...
 .. ..$ : int [1:23] 224 225 226 227 228 229 230 231 232 233 ...
 .. ..$ : int [1:23] 247 248 249 250 251 252 253 254 255 256 ...
 .. ..$ : int [1:23] 270 271 272 273 274 275 276 277 278 279 ...
 .. ..$ : int [1:21] 293 294 295 296 297 298 299 300 301 302 ...
 .. ..$ : int [1:22] 314 315 316 317 318 319 320 321 322 323 ...
 .. ..$ : int [1:23] 336 337 338 339 340 341 342 343 344 345 ...
 .. ..$ : int [1:23] 359 360 361 362 363 364 365 366 367 368 ...
 .. ..$ : int [1:21] 382 383 384 385 386 387 388 389 390 391 ...
 .. ..$ : int [1:23] 403 404 405 406 407 408 409 410 411 412 ...
 .. ..$ : int [1:23] 426 427 428 429 430 431 432 433 434 435 ...
 .. ..$ : int [1:23] 449 450 451 452 453 454 455 456 457 458 ...
 .. ..$ : int [1:23] 472 473 474 475 476 477 478 479 480 481 ...
 .. ..$ : int [1:23] 495 496 497 498 499 500 501 502 503 504 ...
 .. ..$ : int [1:18] 518 519 520 521 522 523 524 525 526 527 ...
 .. ..$ : int [1:23] 536 537 538 539 540 541 542 543 544 545 ...
 .. ..$ : int [1:23] 559 560 561 562 563 564 565 566 567 568 ...
 .. ..$ : int [1:23] 582 583 584 585 586 587 588 589 590 591 ...
 .. ..$ : int [1:23] 605 606 607 608 609 610 611 612 613 614 ...
 .. ..$ : int [1:23] 628 629 630 631 632 633 634 635 636 637 ...
 .. ..$ : int [1:23] 651 652 653 654 655 656 657 658 659 660 ...
```

```
  .. ..$ : int [1:23] 674 675 676 677 678 679 680 681 682 683 ...
  .. ..$ : int [1:23] 697 698 699 700 701 702 703 704 705 706 ...
  .. ..$ : int [1:23] 720 721 722 723 724 725 726 727 728 729 ...
  .. ..$ : int [1:23] 743 744 745 746 747 748 749 750 751 752 ...
  .. ..$ : int [1:23] 766 767 768 769 770 771 772 773 774 775 ...
  .. ..$ : int [1:23] 789 790 791 792 793 794 795 796 797 798 ...
  .. ..$ : int [1:23] 812 813 814 815 816 817 818 819 820 821 ...
  .. ..$ : int [1:23] 835 836 837 838 839 840 841 842 843 844 ...
  .. ..$ : int [1:23] 858 859 860 861 862 863 864 865 866 867 ...
  .. ..$ : int [1:23] 881 882 883 884 885 886 887 888 889 890 ...
  .. ..$ : int [1:21] 904 905 906 907 908 909 910 911 912 913 ...
  .. ..$ : int [1:23] 925 926 927 928 929 930 931 932 933 934 ...
  .. ..$ : int [1:23] 948 949 950 951 952 953 954 955 956 957 ...
  .. ..$ : int [1:20] 971 972 973 974 975 976 977 978 979 980 ...
  .. ..$ : int [1:23] 991 992 993 994 995 996 997 998 999 1000 ...
  .. ..$ : int [1:23] 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 ...
  .. ..$ : int [1:23] 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 ...
  .. ..$ : int [1:23] 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 ...
  .. ..$ : int [1:23] 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 ...
  .. ..$ : int [1:23] 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 ...
  .. ..@ ptype: int(0)
  ..- attr(*, ".drop")= logi TRUE
```

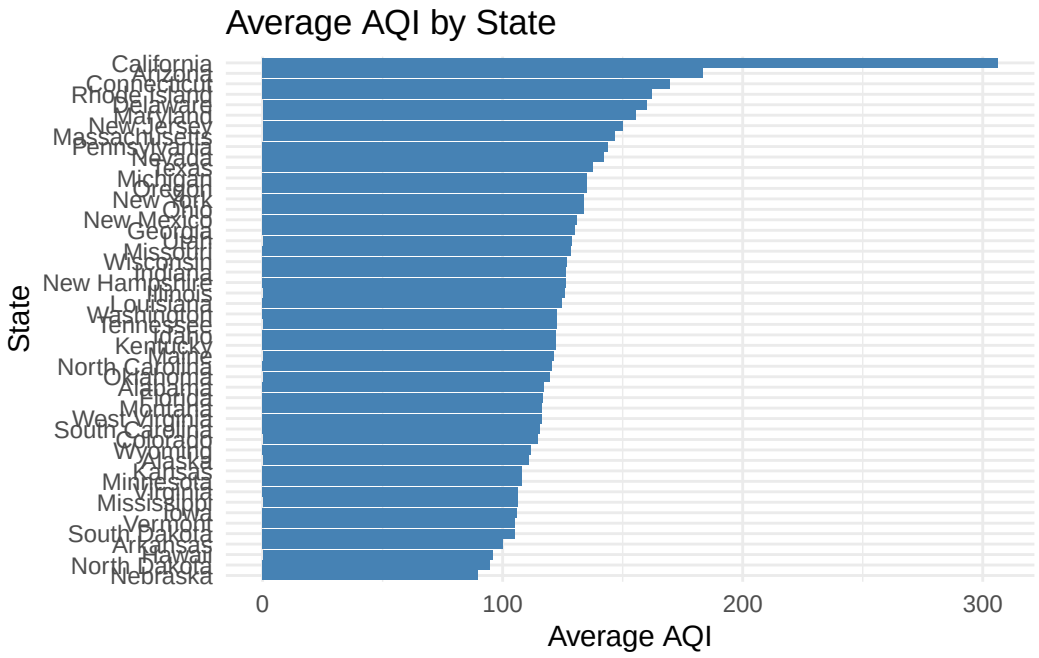
```r
# Calculate summary statistics for numeric columns
cleaned_data %>%
  summarise(across(where(is.numeric), list(mean = ~mean(.x, na.rm = TRUE),
                                           median = ~median(.x, na.rm = TRUE),
                                           sd = ~sd(.x, na.rm = TRUE))))
```

```
# A tibble: 50 x 55
   States     year_mean year_median year_sd avg_max_aqi_mean avg_max_aqi_median
   <chr>          <dbl>       <dbl>   <dbl>            <dbl>              <dbl>
 1 Alabama         2010        2010    6.78             117.               113.
 2 Alaska          2010        2010    6.78             111.               102.
 3 Arizona         2010        2010    6.78             184.               172.
 4 Arkansas        2010.       2010.   6.76             100.                99.9
 5 California      2011.       2012    6.45             306.               283.
 6 Colorado        2010        2010    6.78             115.               112.
 7 Connecticut     2010.       2010.   6.65             170.               166.
 8 Delaware        2010.       2010.   6.81             160.               164.
 9 Florida         2010        2010    6.78             117.               117.
10 Georgia         2010        2010    6.78             130.               130.
# i 40 more rows
# i 49 more variables: avg_max_aqi_sd <dbl>,
#   avg_x90th_percentile_aqi_mean <dbl>, avg_x90th_percentile_aqi_median <dbl>,
#   avg_x90th_percentile_aqi_sd <dbl>, avg_median_aqi_mean <dbl>,
#   avg_median_aqi_median <dbl>, avg_median_aqi_sd <dbl>,
#   avg_days_with_aqi_mean <dbl>, avg_days_with_aqi_median <dbl>,
#   avg_days_with_aqi_sd <dbl>, avg_good_days_mean <dbl>, ...
```

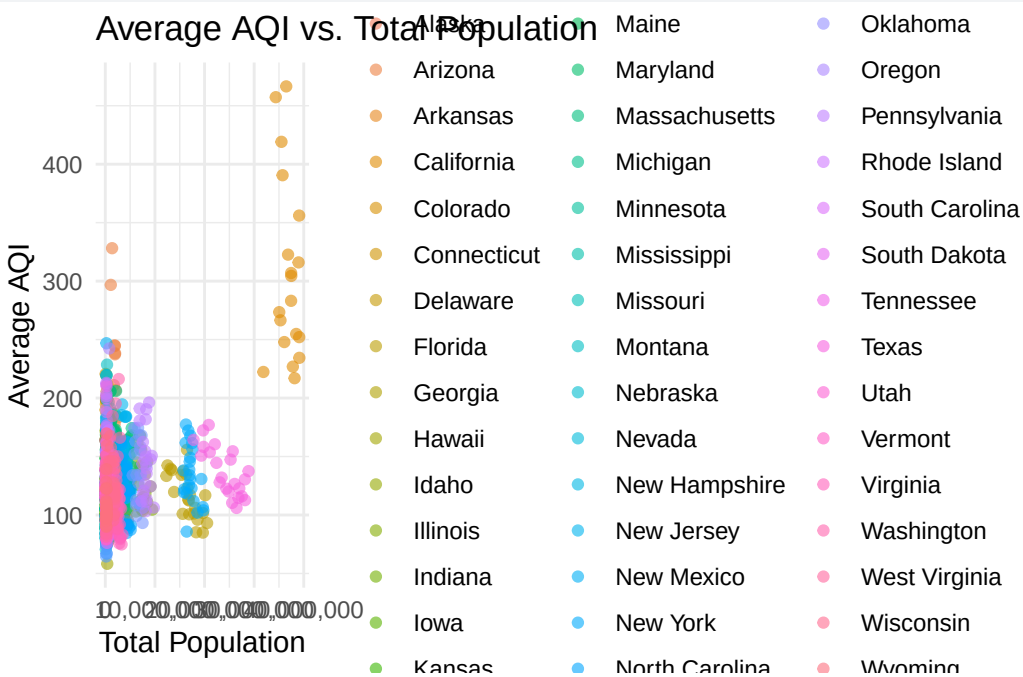
```r
# AQI trends over the years for each state
ggplot(cleaned_data, aes(x = year, y = avg_max_aqi, color = States)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Average Max AQI Over Years by State",
       x = "Year", y = "Average Max AQI") +
  theme(legend.position = "none") # Remove legend for clarity
```



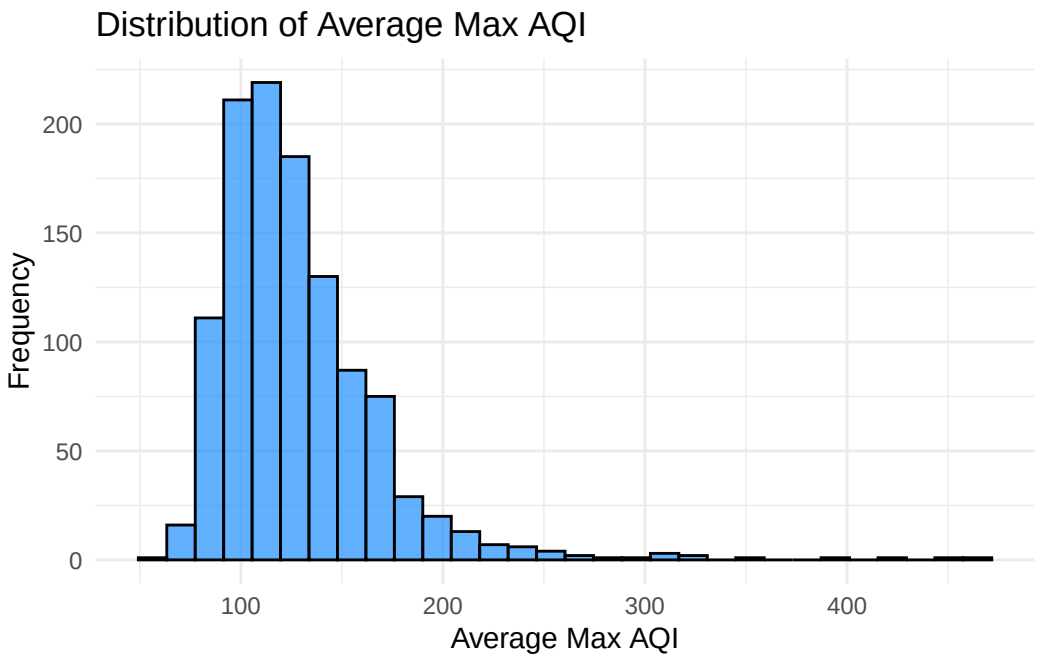

```r
# Average AQI by State
state_avg_aqi <- cleaned_data %>%
  group_by(States) %>%
  summarise(avg_aqi = mean(avg_max_aqi, na.rm = TRUE)) %>%
  arrange(desc(avg_aqi))

ggplot(state_avg_aqi, aes(x = reorder(States, avg_aqi), y = avg_aqi)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  theme_minimal() +
  labs(title = "Average AQI by State", x = "State", y = "Average AQI")
```




```
# Scatter plot of AQI vs. Total Population
ggplot(cleaned_data, aes(x = Total_Population, y = avg_max_aqi)) +
  geom_point(aes(color = States), alpha = 0.6) +
  scale_x_continuous(labels = scales::comma) +
  theme_minimal() +
  labs(title = "Average AQI vs. Total Population", x = "Total Population", y = "Average AQI")
```




```
# Histogram of the average maximum AQI
ggplot(cleaned_data, aes(x = avg_max_aqi)) +
  geom_histogram(bins = 30, fill = "dodgerblue", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Distribution of Average Max AQI", x = "Average Max AQI", y = "Frequency")
```




```r
library(ggplot2)
library(dplyr)

# Split states into two groups
state_groups <- cleaned_data %>%
  mutate(group = ifelse(States %in% unique(States)[1:25], "Group 1", "Group 2"))

# Function to create the plot for each group
create_facet_plot <- function(data, title_suffix) {
  ggplot(data, aes(x = year, y = avg_max_aqi, group = States)) +
    geom_line(color = "steelblue", size = 0.8) +
    facet_wrap(~ States, ncol = 5, nrow = 5, scales = "free_y") +
    theme_minimal(base_size = 12) +
    labs(
      title = paste("State-Specific AQI Trends Over Years", title_suffix),
      x = "Year", y = "Average Max AQI"
    ) +
    scale_y_continuous(
      breaks = function(x) pretty(x, n = 3)
    ) +
    scale_x_continuous(
      breaks = seq(
        min(cleaned_data$year, na.rm = TRUE),
        max(cleaned_data$year, na.rm = TRUE),
        by = 5
      )
    ) +
    theme(
      strip.text = element_text(size = 10, face = "bold"),
      axis.text.x = element_text(size = 10, angle = 30, hjust = 1),
      axis.text.y = element_text(size = 8),
```

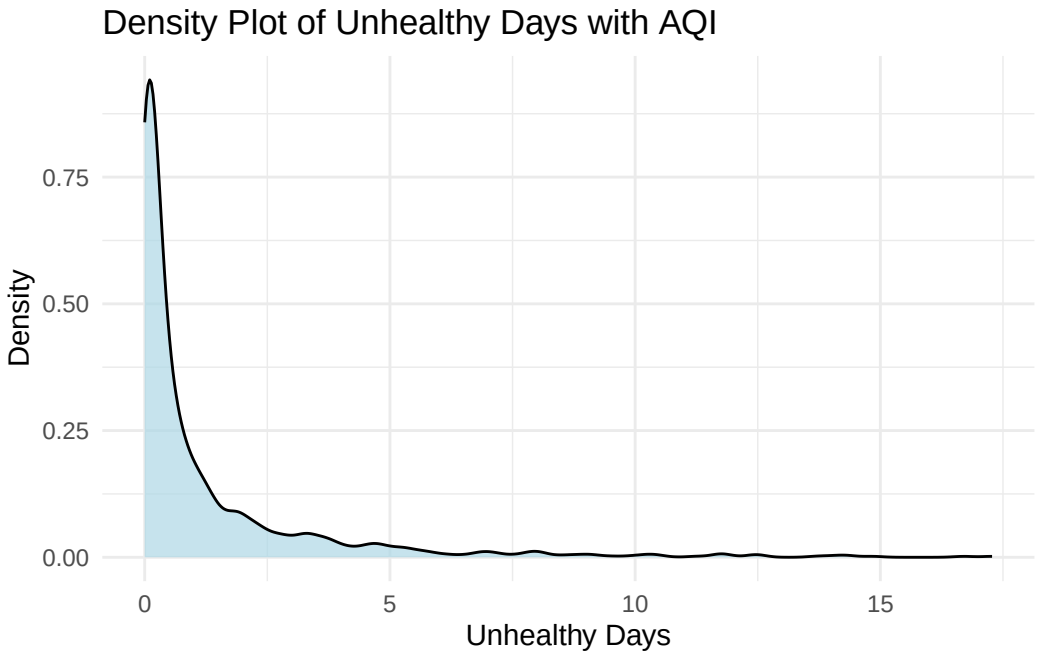
```
      plot.title = element_text(size = 16, face = "bold"),
      panel.spacing = unit(1.5, "lines"),
      plot.margin = margin(10, 10, 10, 10)
    )
}

# Generate plots for Group 1 and Group 2
plot_group1 <- create_facet_plot(state_groups %>% filter(group == "Group 1"), "(Group 1)")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
plot_group2 <- create_facet_plot(state_groups %>% filter(group == "Group 2"), "(Group 2)")

# Save both plots to a single PDF
pdf("state_aqi_trends_split_adjusted.pdf", width = 18, height = 14)

dev.off()
```

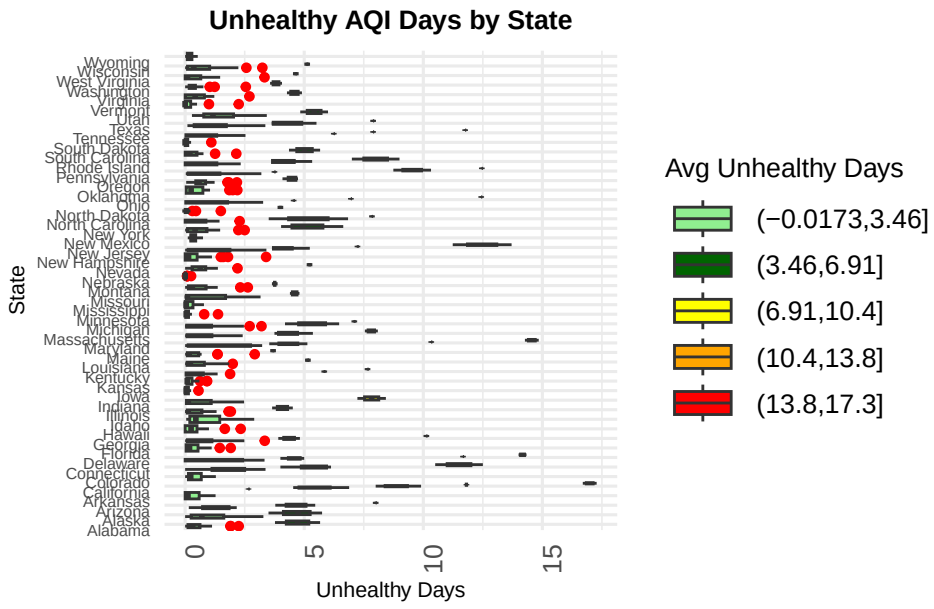
```
pdf
  2
```

```
# Density plot of unhealthy days
ggplot(cleaned_data, aes(x = avg_unhealthy_days)) +
  geom_density(fill = "lightblue", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Density Plot of Unhealthy Days with AQI",
       x = "Unhealthy Days", y = "Density")
```




```
library(ggplot2)
library(dplyr)

# Sort states by median unhealthy days
```

```r
cleaned_data <- cleaned_data %>%
  mutate(States = reorder(States, avg_unhealthy_days, median, na.rm = TRUE))

# Create the boxplot with all 50 states
ggplot(cleaned_data, aes(x = States, y = avg_unhealthy_days)) +
  geom_boxplot(aes(fill = as.factor(cut(avg_unhealthy_days, breaks = 5))),
               outlier.color = "red", outlier.size = 1.2) +
  scale_fill_manual(values = c("lightgreen", "darkgreen", "yellow", "orange", "red")) +
  coord_flip() +
  theme_minimal(base_size = 12) +
  labs(
    title = "Unhealthy AQI Days by State",
    x = "State",
    y = "Unhealthy Days",
    fill = "Avg Unhealthy Days"
  ) +
  theme(
    axis.text.y = element_text(size = 6, vjust = 1, hjust = 1),   # Smaller y-axis text
    axis.text.x = element_text(size = 10, angle = 90, hjust = 1),  # Rotate x-axis labels
    axis.ticks.y = element_blank(),
    axis.title.x = element_text(size = 8),
    axis.title.y = element_text(size = 8),
    plot.title = element_text(size = 10, face = "bold", hjust = 0.5),
    legend.position = "right",
    legend.key.width = unit(1, "cm"),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 10),
    plot.margin = margin(15, 15, 15, 30),  # Increased bottom margin for readability
    scale_y_continuous(breaks = seq(0, max(cleaned_data$avg_unhealthy_days), by = 2))
  )
```



```r
# Ensure that the data is ungrouped
cleaned_data <- cleaned_data %>% ungroup()

# Select only numeric columns
numeric_data <- cleaned_data %>%
  select(where(is.numeric))

# Compute the correlation matrix
cor_matrix <- cor(numeric_data, use = "complete.obs")

# Print the correlation matrix
print(cor_matrix)
```

```
                                          year avg_max_aqi
year                                1.00000000 -0.20389224
avg_max_aqi                        -0.20389224  1.00000000
avg_x90th_percentile_aqi           -0.40458365  0.67505568
avg_median_aqi                     -0.06242946  0.43023541
avg_days_with_aqi                   0.51328732  0.18689323
avg_good_days                       0.52834788 -0.21914710
avg_moderate_days                   0.07778249  0.41799183
avg_unhealthy_for_sensitive_groups_days -0.42288197  0.75389796
avg_unhealthy_days                 -0.32409289  0.69982723
avg_very_unhealthy_days            -0.17403639  0.57327276
avg_hazardous_days                  0.04684586  0.47092206
avg_days_co                        -0.36038577  0.00597589
avg_days_no2                       -0.34590792  0.16998642
avg_days_ozone                      0.18421218  0.21403556
avg_days_pm2_5                      0.39884828 -0.07461222
avg_days_pm10                      -0.15157526 -0.02945708
Total_Count                         0.07464031  0.38499016
Total_Population                    0.05540930  0.42922529
                                   avg_x90th_percentile_aqi avg_median_aqi
year                                           -0.40458365    -0.06242946
avg_max_aqi                                     0.67505568     0.43023541
avg_x90th_percentile_aqi                        1.00000000     0.76113660
avg_median_aqi                                  0.76113660     1.00000000
avg_days_with_aqi                              -0.05852535     0.11039583
avg_good_days                                  -0.59264756    -0.49633038
avg_moderate_days                               0.59505368     0.80730908
avg_unhealthy_for_sensitive_groups_days         0.89350965     0.55827852
avg_unhealthy_days                              0.74726281     0.35560896
avg_very_unhealthy_days                         0.51590631     0.21966128
avg_hazardous_days                              0.11984193     0.06503809
avg_days_co                                    -0.02286648    -0.28936071
avg_days_no2                                    0.22313611     0.00603191
avg_days_ozone                                  0.17467285     0.32324580
avg_days_pm2_5                                 -0.14944136    -0.01284501
avg_days_pm10                                  -0.23351251    -0.39672124
```

Total_Count	0.22642075	0.28137416
Total_Population	0.21154761	0.24237940

	avg_days_with_aqi	avg_good_days
year	0.513287323	0.52834788
avg_max_aqi	0.186893225	-0.21914710
avg_x90th_percentile_aqi	-0.058525354	-0.59264756
avg_median_aqi	0.110395831	-0.49633038
avg_days_with_aqi	1.000000000	0.73257772
avg_good_days	0.732577720	1.00000000
avg_moderate_days	0.394388796	-0.31929829
avg_unhealthy_for_sensitive_groups_days	0.000625128	-0.45940877
avg_unhealthy_days	0.028316496	-0.28844662
avg_very_unhealthy_days	0.083733620	-0.13114912
avg_hazardous_days	0.093843201	-0.02179993
avg_days_co	-0.235611637	-0.09603969
avg_days_no2	-0.055856650	-0.07456027
avg_days_ozone	0.459928593	0.33271019
avg_days_pm2_5	0.439230933	0.25497011
avg_days_pm10	-0.136254178	0.06622098
Total_Count	0.352872423	0.07688920
Total_Population	0.334413071	0.07869097

	avg_moderate_days
year	0.07778249
avg_max_aqi	0.41799183
avg_x90th_percentile_aqi	0.59505368
avg_median_aqi	0.80730908
avg_days_with_aqi	0.39438880
avg_good_days	-0.31929829
avg_moderate_days	1.00000000
avg_unhealthy_for_sensitive_groups_days	0.45018608
avg_unhealthy_days	0.24722668
avg_very_unhealthy_days	0.15510053
avg_hazardous_days	0.10821568
avg_days_co	-0.23376351
avg_days_no2	-0.05531032
avg_days_ozone	0.13492094
avg_days_pm2_5	0.33928718
avg_days_pm10	-0.29800418
Total_Count	0.34048722
Total_Population	0.30168383

	avg_unhealthy_for_sensitive_groups_days
year	-0.422881975
avg_max_aqi	0.753897962
avg_x90th_percentile_aqi	0.893509646
avg_median_aqi	0.558278521
avg_days_with_aqi	0.000625128
avg_good_days	-0.459408772
avg_moderate_days	0.450186081
avg_unhealthy_for_sensitive_groups_days	1.000000000

avg_unhealthy_days	0.805958249
avg_very_unhealthy_days	0.534647716
avg_hazardous_days	0.227486772
avg_days_co	0.059916787
avg_days_no2	0.299907536
avg_days_ozone	0.218194126
avg_days_pm2_5	-0.240317292
avg_days_pm10	-0.059633416
Total_Count	0.302518082
Total_Population	0.322974733

	avg_unhealthy_days
year	-0.32409289
avg_max_aqi	0.69982723
avg_x90th_percentile_aqi	0.74726281
avg_median_aqi	0.35560896
avg_days_with_aqi	0.02831650
avg_good_days	-0.28844662
avg_moderate_days	0.24722668
avg_unhealthy_for_sensitive_groups_days	0.80595825
avg_unhealthy_days	1.00000000
avg_very_unhealthy_days	0.76964250
avg_hazardous_days	0.21921110
avg_days_co	0.08807261
avg_days_no2	0.29527107
avg_days_ozone	0.11344716
avg_days_pm2_5	-0.11371631
avg_days_pm10	-0.09166169
Total_Count	0.27606581
Total_Population	0.31169762

	avg_very_unhealthy_days
year	-0.17403639
avg_max_aqi	0.57327276
avg_x90th_percentile_aqi	0.51590631
avg_median_aqi	0.21966128
avg_days_with_aqi	0.08373362
avg_good_days	-0.13114912
avg_moderate_days	0.15510053
avg_unhealthy_for_sensitive_groups_days	0.53464772
avg_unhealthy_days	0.76964250
avg_very_unhealthy_days	1.00000000
avg_hazardous_days	0.29490230
avg_days_co	0.06964201
avg_days_no2	0.21232792
avg_days_ozone	0.09863070
avg_days_pm2_5	-0.08387798
avg_days_pm10	0.01785630
Total_Count	0.17874893
Total_Population	0.20704068

	avg_hazardous_days	avg_days_co

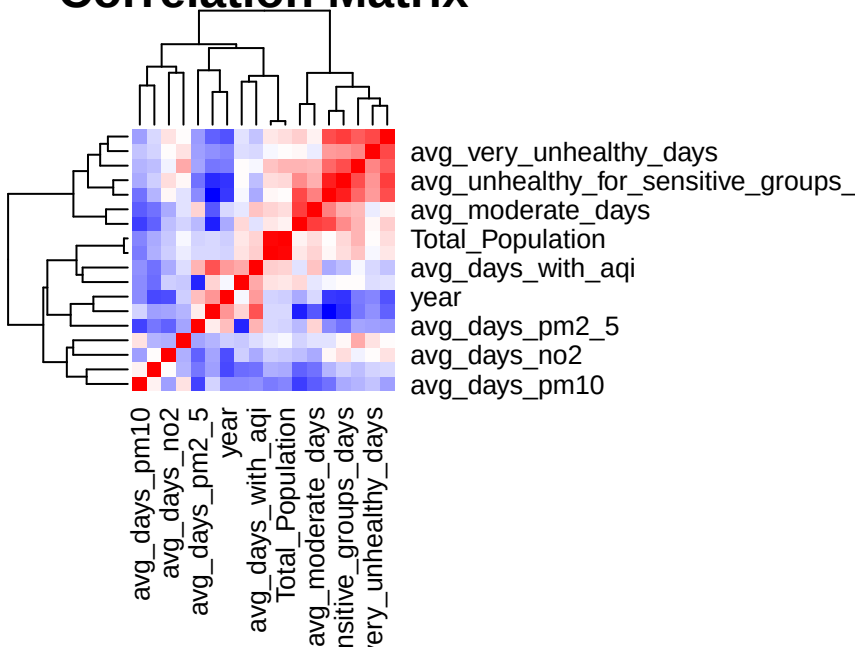
year	0.04684586	-0.36038577
avg_max_aqi	0.47092206	0.00597589
avg_x90th_percentile_aqi	0.11984193	-0.02286648
avg_median_aqi	0.06503809	-0.28936071
avg_days_with_aqi	0.09384320	-0.23561164
avg_good_days	-0.02179993	-0.09603969
avg_moderate_days	0.10821568	-0.23376351
avg_unhealthy_for_sensitive_groups_days	0.22748677	0.05991679
avg_unhealthy_days	0.21921110	0.08807261
avg_very_unhealthy_days	0.29490230	0.06964201
avg_hazardous_days	1.00000000	-0.02986652
avg_days_co	-0.02986652	1.00000000
avg_days_no2	-0.04849440	0.21806297
avg_days_ozone	0.04093125	-0.25727199
avg_days_pm2_5	-0.06949123	-0.21643031
avg_days_pm10	0.28832028	0.24549284
Total_Count	0.15173717	-0.04779091
Total_Population	0.18732776	-0.05119970

	avg_days_no2	avg_days_ozone
year	-0.34590792	0.18421218
avg_max_aqi	0.16998642	0.21403556
avg_x90th_percentile_aqi	0.22313611	0.17467285
avg_median_aqi	0.00603191	0.32324580
avg_days_with_aqi	-0.05585665	0.45992859
avg_good_days	-0.07456027	0.33271019
avg_moderate_days	-0.05531032	0.13492094
avg_unhealthy_for_sensitive_groups_days	0.29990754	0.21819413
avg_unhealthy_days	0.29527107	0.11344716
avg_very_unhealthy_days	0.21232792	0.09863070
avg_hazardous_days	-0.04849440	0.04093125
avg_days_co	0.21806297	-0.25727199
avg_days_no2	1.00000000	0.05359207
avg_days_ozone	0.05359207	1.00000000
avg_days_pm2_5	-0.27828246	-0.47238543
avg_days_pm10	-0.09209680	-0.15857884
Total_Count	0.04576440	0.29381918
Total_Population	0.03387115	0.27599105

	avg_days_pm2_5	avg_days_pm10
year	0.39884828	-0.15157526
avg_max_aqi	-0.07461222	-0.02945708
avg_x90th_percentile_aqi	-0.14944136	-0.23351251
avg_median_aqi	-0.01284501	-0.39672124
avg_days_with_aqi	0.43923093	-0.13625418
avg_good_days	0.25497011	0.06622098
avg_moderate_days	0.33928718	-0.29800418
avg_unhealthy_for_sensitive_groups_days	-0.24031729	-0.05963342
avg_unhealthy_days	-0.11371631	-0.09166169
avg_very_unhealthy_days	-0.08387798	0.01785630
avg_hazardous_days	-0.06949123	0.28832028

```
avg_days_co                                -0.21643031    0.24549284
avg_days_no2                               -0.27828246   -0.09209680
avg_days_ozone                             -0.47238543   -0.15857884
avg_days_pm2_5                              1.00000000   -0.37890019
avg_days_pm10                              -0.37890019    1.00000000
Total_Count                                 0.08251445   -0.20700995
Total_Population                            0.06996585   -0.16392800
                                       Total_Count Total_Population
year                                    0.07464031       0.05540930
avg_max_aqi                             0.38499016       0.42922529
avg_x90th_percentile_aqi                0.22642075       0.21154761
avg_median_aqi                          0.28137416       0.24237940
avg_days_with_aqi                       0.35287242       0.33441307
avg_good_days                           0.07688920       0.07869097
avg_moderate_days                       0.34048722       0.30168383
avg_unhealthy_for_sensitive_groups_days 0.30251808       0.32297473
avg_unhealthy_days                      0.27606581       0.31169762
avg_very_unhealthy_days                 0.17874893       0.20704068
avg_hazardous_days                      0.15173717       0.18732776
avg_days_co                            -0.04779091      -0.05119970
avg_days_no2                            0.04576440       0.03387115
avg_days_ozone                          0.29381918       0.27599105
avg_days_pm2_5                          0.08251445       0.06996585
avg_days_pm10                          -0.20700995      -0.16392800
Total_Count                             1.00000000       0.97170413
Total_Population                        0.97170413       1.00000000
```

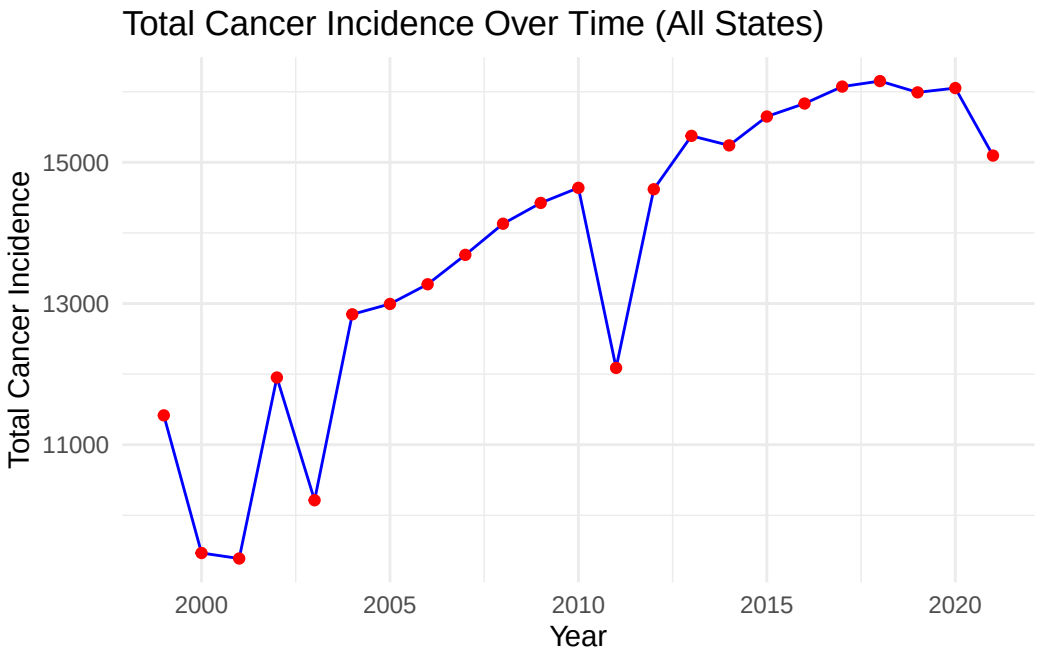
```r
# Visualize the correlation matrix using a heatmap
heatmap(cor_matrix,
        main = "Correlation Matrix",
        col = colorRampPalette(c("blue", "white", "red"))(100),
        scale = "none",
        margins = c(8, 8))
```




```
# Selecting relevant numeric columns for correlation
correlation_data <- cleaned_data %>%
  select(Total_Count, avg_max_aqi, avg_moderate_days, avg_unhealthy_days, avg_very_unhealthy_

# Calculating correlation matrix for selected variables
cor_matrix <- cor(correlation_data, use = "complete.obs")

# Viewing the correlation matrix
print(cor_matrix)
```

```
                      Total_Count avg_max_aqi avg_moderate_days
Total_Count             1.00000000  0.38499016         0.3404872
avg_max_aqi             0.38499016  1.00000000         0.4179918
avg_moderate_days       0.34048722  0.41799183         1.0000000
avg_unhealthy_days      0.27606581  0.69982723         0.2472267
avg_very_unhealthy_days 0.17874893  0.57327276         0.1551005
avg_days_pm2_5          0.08251445 -0.07461222         0.3392872
                      avg_unhealthy_days avg_very_unhealthy_days
Total_Count                    0.2760658              0.17874893
avg_max_aqi                    0.6998272              0.57327276
avg_moderate_days              0.2472267              0.15510053
avg_unhealthy_days             1.0000000              0.76964250
avg_very_unhealthy_days        0.7696425              1.00000000
avg_days_pm2_5                -0.1137163             -0.08387798
                      avg_days_pm2_5
Total_Count              0.08251445
avg_max_aqi             -0.07461222
avg_moderate_days        0.33928718
avg_unhealthy_days      -0.11371631
avg_very_unhealthy_days -0.08387798
avg_days_pm2_5           1.00000000
```

```r
# Summing Total_Count across all states by year
total_cancer_by_year <- cleaned_data %>%
  group_by(year) %>%
  summarise(total_cancer_incidence = sum(Total_Count, na.rm = TRUE), .groups = "drop")

# Plotting cancer incidence over time
ggplot(total_cancer_by_year, aes(x = year, y = total_cancer_incidence)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +
  labs(title = "Total Cancer Incidence Over Time (All States)",
       x = "Year",
       y = "Total Cancer Incidence") +
  theme_minimal()
```

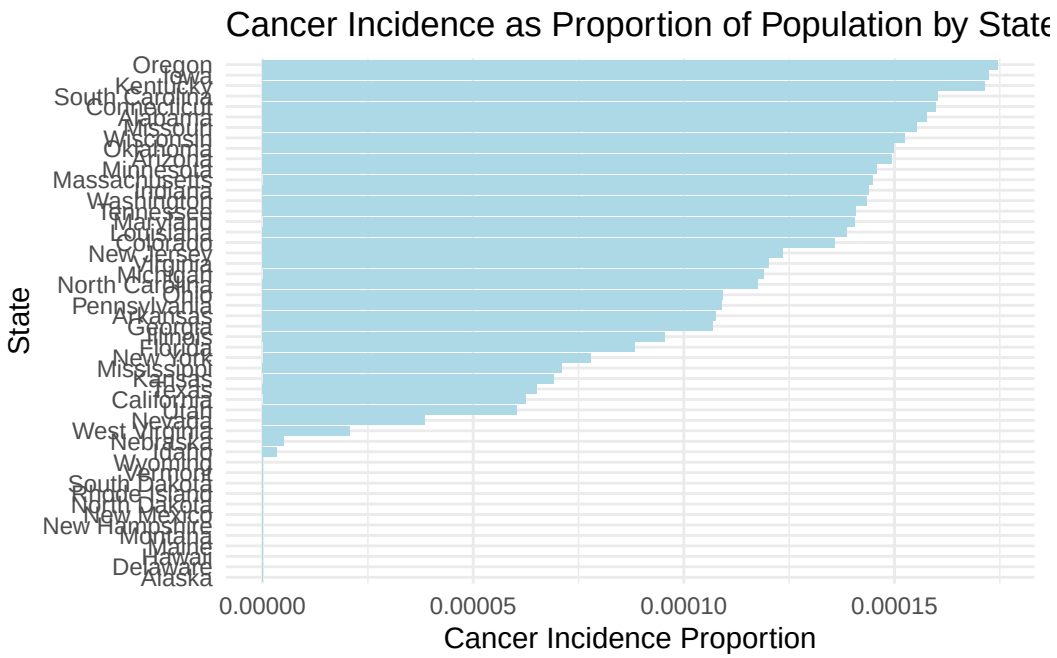



```r
# Summarizing Total_Count by State
state_summary <- cleaned_data %>%
  group_by(States) %>%
  summarise(
    avg_total_count = mean(Total_Count, na.rm = TRUE),
    total_total_count = sum(Total_Count, na.rm = TRUE),
    .groups = "drop"
  )

# Viewing the summarized data
head(state_summary)
```

```
# A tibble: 6 x 3
  States     avg_total_count total_total_count
  <fct>                <dbl>             <dbl>
1 Alabama              123                2829
2 Alaska                 0                   0
3 Arizona              221.               5076
```

```
4 Arkansas              20.3              447
5 California          2300.             43705
6 Colorado             108.              2489
```

```r
# Calculating total cancer incidence and total population by state
state_proportion <- cleaned_data %>%
  group_by(States) %>%
  summarise(
    total_cancer = sum(Total_Count, na.rm = TRUE),
    total_population = sum(Total_Population, na.rm = TRUE),
    cancer_proportion = total_cancer / total_population,
    .groups = "drop"
  )

# Plotting the cancer incidence proportion by state
ggplot(state_proportion, aes(x = reorder(States, cancer_proportion), y = cancer_proportion)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  coord_flip() +
  labs(title = "Cancer Incidence as Proportion of Population by State",
       x = "State",
       y = "Cancer Incidence Proportion") +
  theme_minimal()
```



we specify the following priors:

- $\alpha \sim \text{Normal}(0,\ 10)$: The prior for the intercept is a normal distribution with a mean of 0 and a large standard deviation of 10, reflecting uncertainty about the baseline incidence.
- $\beta_k \sim \text{Normal}(0,\ 10)$ for each element _k: The priors for the coefficients of the predictors are also normally distributed with mean 0 and a large standard deviation of 10, allowing for flexibility in how predictors affect the cancer incidence count.
- $\lambda \sim \text{Gamma}(2,\ 0.1)$: This is the prior for the rate parameter in the Poisson distribution, with a mean of 2 and a large variance, allowing the model to adapt to the observed data.

These priors are relatively weak, meaning that they do not overly constrain the model. They are

designed to allow the data to drive the inference

```
# Define the predictors and response variable
predictors <- c("avg_max_aqi", "avg_moderate_days", "avg_unhealthy_days", "avg_very_unhealthy
response <- "Total_Count"

# Subset the data for predictors and response
data_for_model <- cleaned_data %>%
  select(States, year, all_of(predictors), response) %>%
  na.omit()
```

```
Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(response)

  # Now:
  data %>% select(all_of(response))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
# Create a matrix for predictors (X) and a vector for the response (y)
X <- as.matrix(data_for_model[, predictors])
y <- data_for_model[, response]

# Define the number of observations (N), time periods (T), and predictors (K)
N <- nrow(X)
T <- length(unique(data_for_model$year))
K <- ncol(X)
```

The likelihood of the data is specified as a Poisson distribution, where the observed total cancer count y_i for each state-year pair follows a Poisson distribution with parameter λ_i

The rate λ_i is the exponential of a linear combination of the predictors. This captures the multiplicative effects of the predictors on the expected cancer incidence.

```
# Prepare the data list for STAN
stan_data <- list(
  N = N,
  T = T,
  K = K,
  X = X,
  y = y
)
```

The posterior distribution is the updated belief about the parameters after observing the data. It is obtained by applying Bayes' theorem:

$P(\alpha, \beta|y, X) \propto P(y|X, \alpha, \beta) * P(\alpha) * P(\beta)$

Where:

- $P(y|X, \alpha, \beta)$ is the likelihood, as specified above.
- $P(\alpha)$ and $P(\beta)$ are the priors for the parameters α and β.
- The posterior distribution reflects the parameter estimates that are most consistent with the

observed data, while also incorporating prior beliefs.

```
library(rstan)
```

```
Loading required package: StanHeaders

rstan version 2.32.6 (Stan version 2.32.2)
For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
change `threads_per_chain` option:
rstan_options(threads_per_chain = 1)
Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file

Attaching package: 'rstan'
The following object is masked from 'package:tidyr':

    extract
```

```
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

model <- stan_model("finalmodel.stan")

y <- as.vector(cleaned_data$Total_Count)
N <- length(y)

state_levels <- unique(cleaned_data$States)
state_index <- as.integer(factor(cleaned_data$States, levels = state_levels))

year_index <- cleaned_data$year
X <- cleaned_data[, c("avg_max_aqi", "avg_days_with_aqi", "avg_good_days",
                      "avg_moderate_days", "avg_unhealthy_days")]

stan_data <- list(
  N = N,
  K = ncol(X),
  y = y,
  X = X,
  state_index = state_index,
  year_index = year_index,
  S = length(state_levels)
)

fit <- sampling(model, data = stan_data, iter = 4000, chains = 4)
```

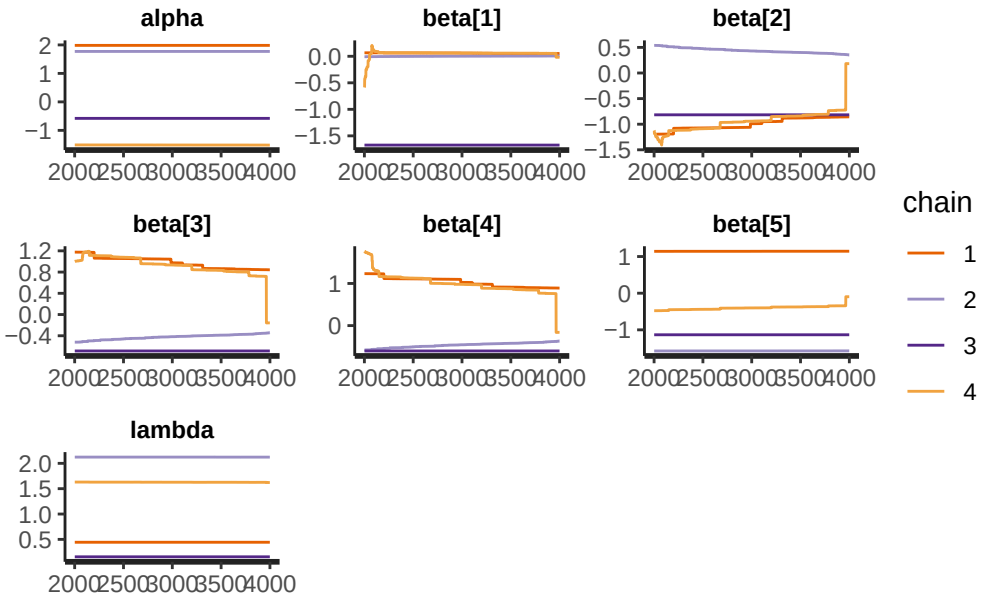
```
Warning: There were 5 divergent transitions after warmup. See
```

```
https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
to find out why this is a problem and how to eliminate them.

Warning: There were 41 transitions after warmup that exceeded the maximum treedepth. Increase
https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

Warning: There were 3 chains where the estimated Bayesian Fraction of Missing Information was
https://mc-stan.org/misc/warnings.html#bfmi-low

Warning: Examine the pairs() plot to diagnose sampling problems

Warning: The largest R-hat is 5.74, indicating chains have not mixed.
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#r-hat

Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#bulk-ess

Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tai
Running the chains for more iterations may help. See
https://mc-stan.org/misc/warnings.html#tail-ess
```

```
print(fit)
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=4000; warmup=2000; thin=1;
post-warmup draws per chain=2000, total post-warmup draws=8000.

         mean se_mean   sd  2.5%   25%        50%       75%      97.5% n_eff
alpha    0.42    1.06 1.50 -1.52 -0.81       0.60      1.82       1.98     2
beta[1] -0.39    0.52 0.74 -1.67 -0.86       0.01      0.06       0.07     2
beta[2] -0.58    0.41 0.60 -1.19 -0.96      -0.82      0.23       0.51     2
beta[3]  0.20    0.54 0.77 -0.69 -0.56      -0.25      0.95       1.17     2
beta[4]  0.24    0.54 0.79 -0.60 -0.58      -0.26      0.99       1.23     2
beta[5] -0.49    0.73 1.03 -1.58 -1.24      -0.81      0.21       1.14     2
lambda   1.09    0.58 0.81  0.16  0.37       1.03      1.75       2.12     2
lp__     -Inf     NaN  NaN  -Inf  -Inf -129962.95 906841.78 1148636.44   NaN
           Rhat
alpha   4078.50
beta[1]   28.79
beta[2]    7.57
beta[3]   10.37
beta[4]    8.19
beta[5]   52.03
lambda  1728.70
lp__        NaN

Samples were drawn using NUTS(diag_e) at Thu Dec 19 06:05:12 2024.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

```
samples <- extract(fit)

traceplot(fit)
```




```
# Summarize the posterior
summary(fit)
```

```
$summary
              mean   se_mean        sd       2.5%        25%           50%
alpha    0.4157511 1.0610187 1.5006949 -1.5154650 -0.8116968  5.961835e-01
beta[1] -0.3901731 0.5243402 0.7420723 -1.6738996 -0.8597316  5.522560e-03
beta[2] -0.5776952 0.4149787 0.6028229 -1.1939030 -0.9560332 -8.164355e-01
beta[3]  0.1965985 0.5354031 0.7669327 -0.6866577 -0.5628484 -2.503267e-01
beta[4]  0.2396113 0.5436373 0.7863979 -0.6015547 -0.5839239 -2.632704e-01
beta[5] -0.4912898 0.7304061 1.0335504 -1.5755973 -1.2441584 -8.058795e-01
lambda   1.0873540 0.5750694 0.8133732  0.1559206  0.3714993  1.032599e+00
lp__          -Inf       NaN       NaN       -Inf       -Inf -1.299630e+05
                 75%        97.5%    n_eff        Rhat
alpha   1.824497e+00 1.983981e+00 2.000501 4078.502873
beta[1] 6.127208e-02 6.791156e-02 2.002931   28.789605
beta[2] 2.258883e-01 5.125348e-01 2.110220    7.568536
beta[3] 9.487141e-01 1.173291e+00 2.051883   10.372251
beta[4] 9.940011e-01 1.232160e+00 2.092504    8.189001
beta[5] 2.137206e-01 1.144583e+00 2.002325   52.028364
lambda  1.753511e+00 2.122890e+00 2.000503 1728.696299
lp__    9.068418e+05 1.148636e+06      NaN         NaN

$c_summary
, , chains = chain:1

         stats
parameter          mean            sd          2.5%           25%           50%
  alpha       1.9839661 1.186363e-05  1.983947e+00  1.983956e+00  1.983967e+00
```

```
  beta[1]     0.0583398 4.320374e-03  5.237660e-02  5.352646e-02  5.869330e-02
  beta[2]    -0.9960094 1.085409e-01 -1.194214e+00 -1.072690e+00 -9.902507e-01
  beta[3]     0.9803073 1.065808e-01  8.447911e-01  8.639833e-01  9.743546e-01
  beta[4]     1.0319024 1.096511e-01  8.922664e-01  9.122277e-01  1.026028e+00
  beta[5]     1.1436256 8.025593e-04  1.142151e+00  1.143069e+00  1.143692e+00
  lambda      0.4433700 6.770137e-06  4.433598e-01  4.433627e-01  4.433704e-01
  lp__    93486.5748623 2.945501e+05 -4.554123e+05 -1.114415e+05  1.157364e+05
         stats
parameter           75%         97.5%
  alpha    1.983977e+00  1.983983e+00
  beta[1]  6.158387e-02  6.536502e-02
  beta[2] -8.774945e-01 -8.577579e-01
  beta[3]  1.055505e+00  1.175374e+00
  beta[4]  1.109346e+00  1.232163e+00
  beta[5]  1.144493e+00  1.144673e+00
  lambda   4.433770e-01  4.433794e-01
  lp__     4.129701e+05  4.629204e+05

, , chains = chain:2

         stats
parameter          mean           sd          2.5%           25%           50%
  alpha    1.771151e+00 8.458786e-05  1.770976e+00  1.771110e+00  1.771139e+00
  beta[1]  5.905001e-04 3.601321e-03 -7.694479e-03 -1.835239e-03  1.446420e-03
  beta[2]  4.384230e-01 4.803065e-02  3.624401e-01  4.004541e-01  4.305634e-01
  beta[3] -4.249904e-01 4.582621e-02 -5.200195e-01 -4.587244e-01 -4.175182e-01
  beta[4] -4.631533e-01 5.453606e-02 -5.761890e-01 -5.032453e-01 -4.543215e-01
  beta[5] -1.574873e+00 5.731355e-04 -1.575693e+00 -1.575358e+00 -1.574919e+00
  lambda   2.122769e+00 1.329258e-04  2.122512e+00  2.122617e+00  2.122841e+00
  lp__     1.074270e+06 6.766304e+04  9.071733e+05  1.033321e+06  1.094990e+06
         stats
parameter           75%         97.5%
  alpha    1.771206e+00  1.771329e+00
  beta[1]  3.374070e-03  5.744128e-03
  beta[2]  4.737349e-01  5.381645e-01
  beta[3] -3.887463e-01 -3.524579e-01
  beta[4] -4.199377e-01 -3.768233e-01
  beta[5] -1.574512e+00 -1.573573e+00
  lambda   2.122869e+00  2.122913e+00
  lp__     1.127974e+06  1.157750e+06

, , chains = chain:3

         stats
parameter       mean  sd       2.5%        25%        50%        75%      97.5%
  alpha   -0.5785944   0 -0.5785944 -0.5785944 -0.5785944 -0.5785944 -0.5785944
  beta[1] -1.6738996   0 -1.6738996 -1.6738996 -1.6738996 -1.6738996 -1.6738996
  beta[2] -0.8164355   0 -0.8164355 -0.8164355 -0.8164355 -0.8164355 -0.8164355
  beta[3] -0.6866577   0 -0.6866577 -0.6866577 -0.6866577 -0.6866577 -0.6866577
```

```
  beta[4] -0.6015547   0 -0.6015547 -0.6015547 -0.6015547 -0.6015547 -0.6015547
  beta[5] -1.1344818   0 -1.1344818 -1.1344818 -1.1344818 -1.1344818 -1.1344818
  lambda   0.1559206   0  0.1559206  0.1559206  0.1559206  0.1559206  0.1559206
  lp__          -Inf NaN       -Inf       -Inf       -Inf       -Inf       -Inf

, , chains = chain:4

         stats
parameter          mean           sd          2.5%           25%           50%
  alpha   -1.513519e+00 1.388888e-03 -1.515619e+00 -1.514559e+00 -1.513549e+00
  beta[1]  5.427671e-02 5.504784e-02 -6.663775e-02  5.850194e-02  6.233772e-02
  beta[2] -9.367590e-01 2.143252e-01 -1.271562e+00 -1.088632e+00 -9.406312e-01
  beta[3]  9.177349e-01 1.956114e-01  7.195259e-01  8.318153e-01  9.334801e-01
  beta[4]  9.912506e-01 2.535628e-01  7.596680e-01  8.746144e-01  9.784380e-01
  beta[5] -3.994301e-01 5.601110e-02 -4.769403e-01 -4.402371e-01 -4.017299e-01
  lambda   1.627356e+00 1.692154e-03  1.624810e+00  1.626096e+00  1.627524e+00
  lp__    -1.278493e+06 3.030336e+06 -1.508143e+07 -1.113297e+06 -6.752245e+05
         stats
parameter           75%         97.5%
  alpha   -1.512192e+00 -1.511046e+00
  beta[1]  6.692269e-02  8.857127e-02
  beta[2] -8.377254e-01 -7.239698e-01
  beta[3]  1.076066e+00  1.172604e+00
  beta[4]  1.127761e+00  1.709624e+00
  beta[5] -3.746153e-01 -3.436211e-01
  lambda   1.628870e+00  1.630474e+00
  lp__    -3.761523e+05 -5.138505e+04
```

```r
beta_sample <- samples$beta[1, ]

# Convert tibble to a numeric matrix
X_numeric <- as.matrix(X)

# Ensure all columns are numeric
X_numeric <- apply(X_numeric, 2, as.numeric)

str(X_numeric)
```

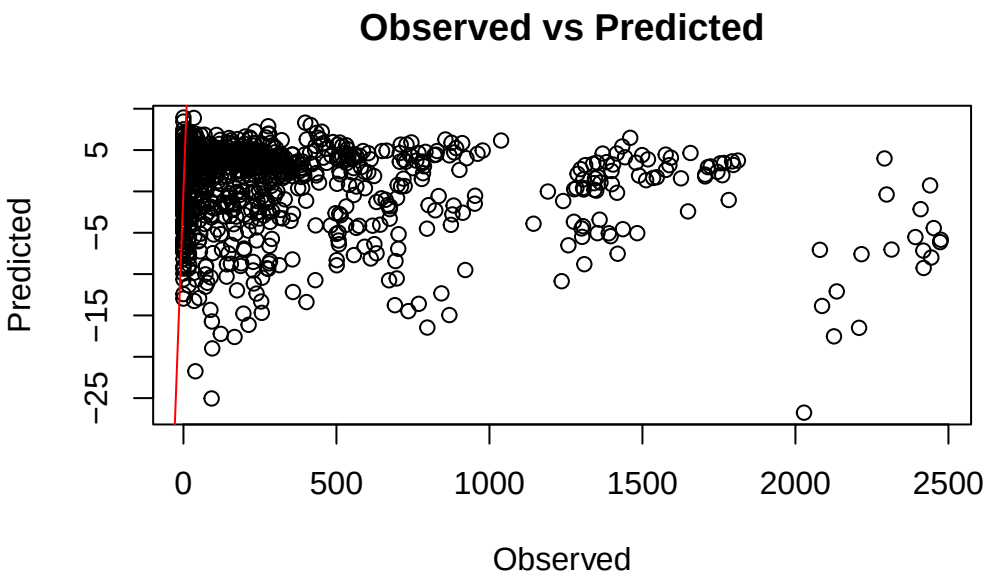
```
 num [1:1128, 1:5] 146 151 137 144 133 ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:5] "avg_max_aqi" "avg_days_with_aqi" "avg_good_days" "avg_moderate_days" ...
```

```r
y_pred <- X_numeric %*% beta_sample


# For the full posterior predictive computation
y_pred_all <- apply(samples$beta, 1, function(beta_sample) {
  X_numeric %*% beta_sample  # Prediction for each posterior sample
})
```

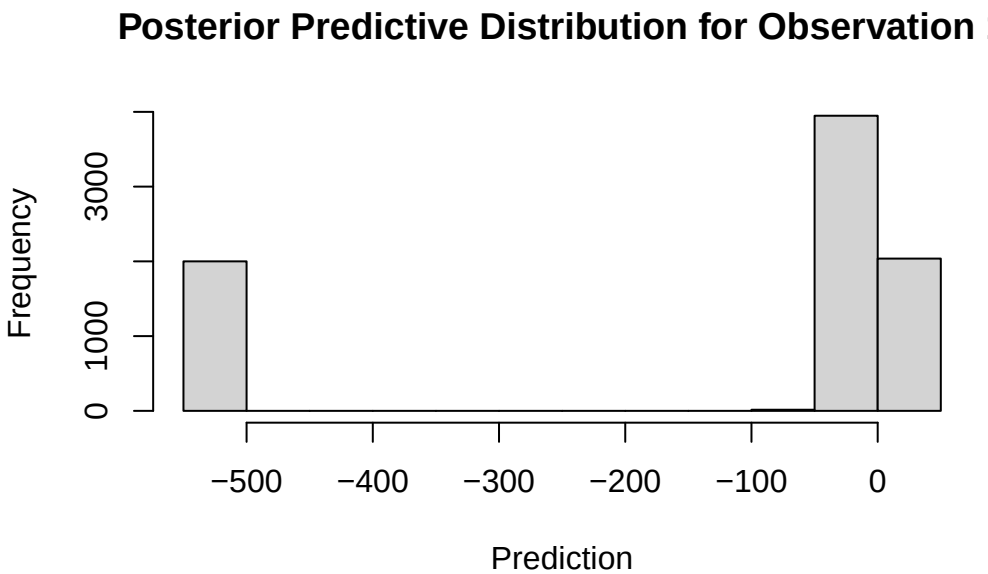
```
str(y_pred_all)
```

```
 num [1:1128, 1:8000] -10.656 -12.907 -2.56 -3.948 0.546 ...
 - attr(*, "dimnames")=List of 2
  ..$            : NULL
  ..$ iterations: NULL
```

```
# Plot the observed vs predicted values (for the first posterior sample)
plot(y, y_pred_all[, 1], main="Observed vs Predicted", xlab="Observed", ylab="Predicted")
abline(a=0, b=1, col="red")  # Add identity line
```

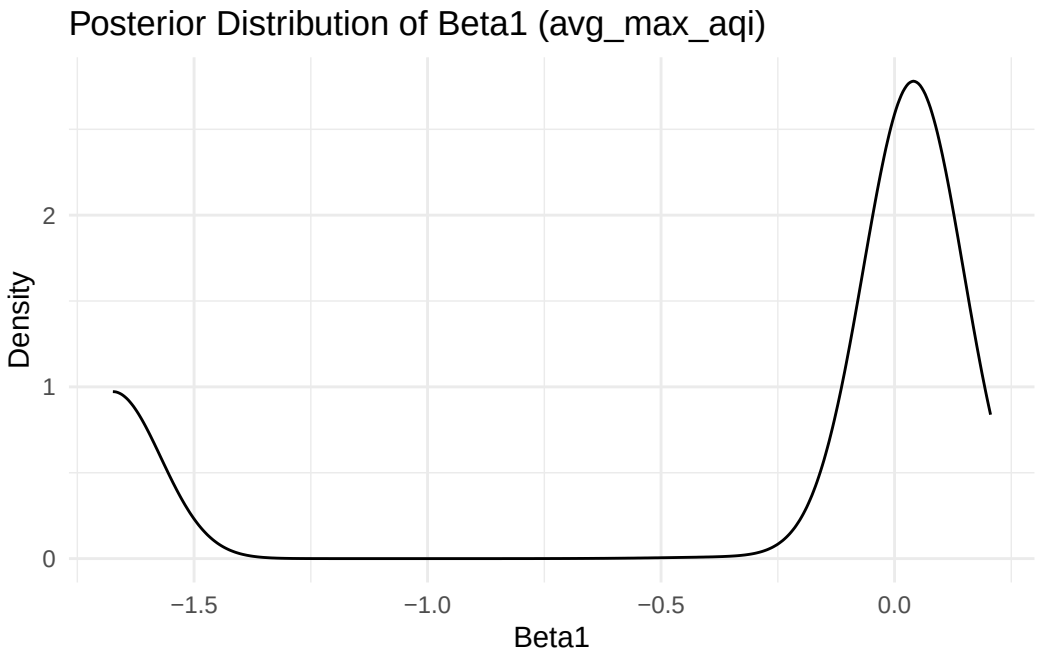



```
# Plot the histogram of the predictions for the first observation
hist(y_pred_all[1, ], main="Posterior Predictive Distribution for Observation 1", xlab="Predi
```




```
# Visualizing the posterior distributions for coefficients}
posterior_plot <- as.data.frame(samples$beta) %>%
  ggplot(aes(x = V1)) +
  geom_density() +
  labs(title = "Posterior Distribution of Beta1 (avg_max_aqi)", x = "Beta1", y = "Density") +
  theme_minimal()
```

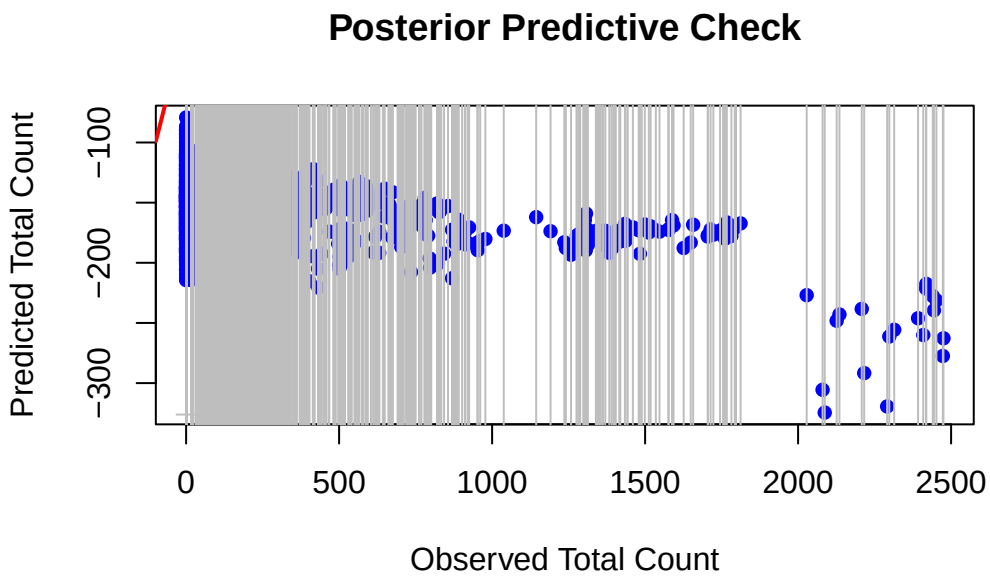
```r
print(posterior_plot)
```



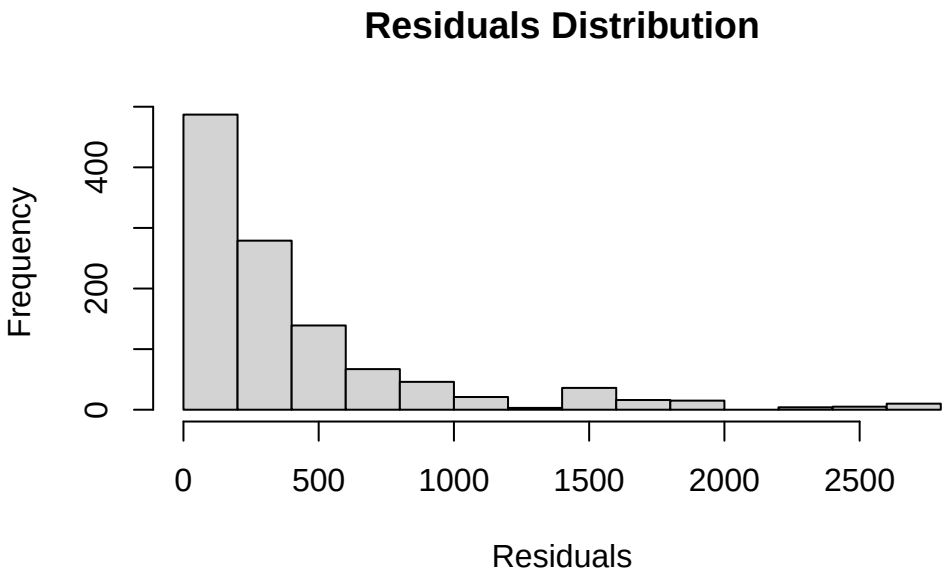

```r
# Predictions for all posterior samples
y_pred_all <- apply(samples$beta, 1, function(beta_sample) {
  X_numeric %*% beta_sample
})
y_obs <- cleaned_data$Total_Count

str(y_obs)
```

```
 num [1:1128] 41 51 79 34 53 54 74 66 112 162 ...
```

```r
stopifnot(length(y_obs) == nrow(X))

y_pred_mean <- rowMeans(y_pred_all)
y_pred_lower <- apply(y_pred_all, 1, quantile, probs = 0.025)
y_pred_upper <- apply(y_pred_all, 1, quantile, probs = 0.975)

plot(y_obs, y_pred_mean,
     xlab = "Observed Total Count",
     ylab = "Predicted Total Count",
     main = "Posterior Predictive Check",
     pch = 16, col = "blue")
abline(0, 1, col = "red", lwd = 2)
arrows(x0 = y_obs, y0 = y_pred_lower, x1 = y_obs, y1 = y_pred_upper,
       angle = 90, code = 3, length = 0.05, col = "gray")
```




```
residuals <- y_obs - y_pred_mean
hist(residuals, main = "Residuals Distribution", xlab = "Residuals")
```




```
mse <- mean((y_obs - y_pred_mean)^2)
cat("Mean Squared Error:", mse, "\n")
```

```
Mean Squared Error: 420919.1
```

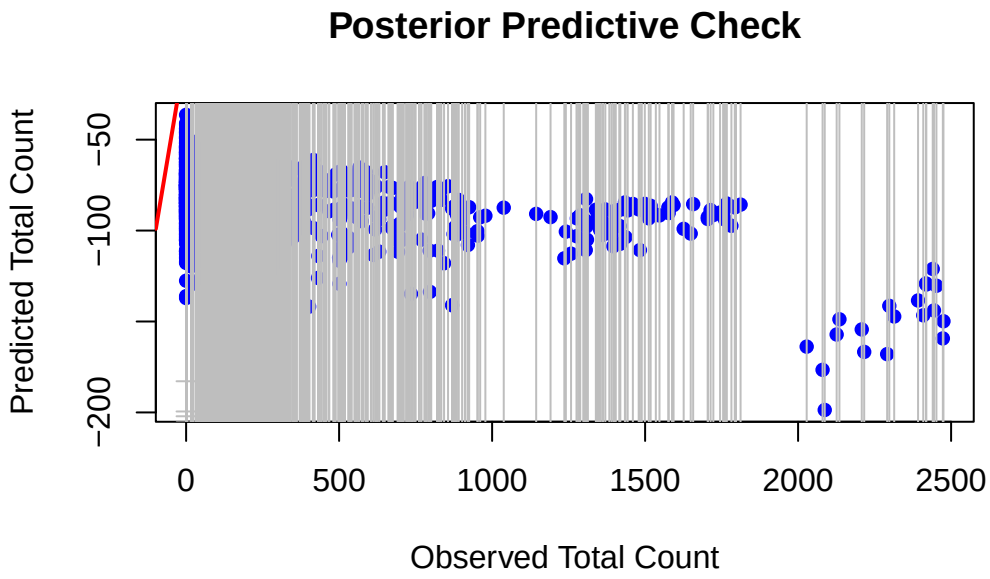
```
# Calculate posterior predictive values using alpha samples
y_pred_all <- apply(samples$beta, 1, function(beta_sample) {
  X_numeric %*% beta_sample
})

dim(y_pred_all)
```

```
[1] 1128 8000
```

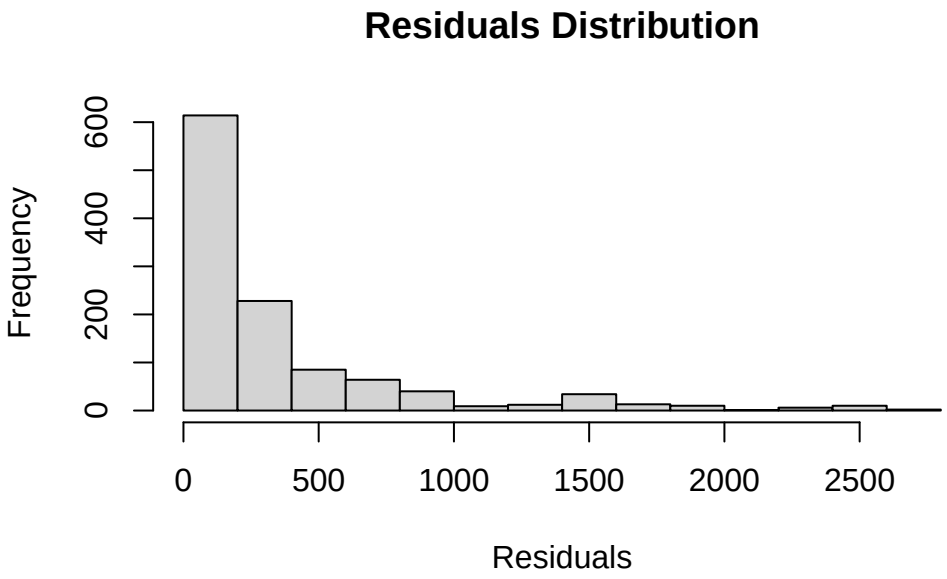
```
length(samples$alpha)
```

```
[1] 8000
```

```r
# Correct element-wise multiplication with exp(samples$alpha)
# Use matrix multiplication or broadcasting where needed
y_pred_all <- sweep(y_pred_all, MARGIN = 2, STATS = exp(samples$alpha), FUN = "*")

# Continue with posterior predictive checks and plots
y_pred_mean <- rowMeans(y_pred_all)
y_pred_lower <- apply(y_pred_all, 1, quantile, probs = 0.025)
y_pred_upper <- apply(y_pred_all, 1, quantile, probs = 0.975)

# Plot observed vs predicted values with credible intervals
plot(y_obs, y_pred_mean,
     xlab = "Observed Total Count",
     ylab = "Predicted Total Count",
     main = "Posterior Predictive Check",
     pch = 16, col = "blue")
abline(0, 1, col = "red", lwd = 2)
arrows(x0 = y_obs, y0 = y_pred_lower, x1 = y_obs, y1 = y_pred_upper,
       angle = 90, code = 3, length = 0.05, col = "gray")
```




```r
# Residuals analysis
residuals <- y_obs - y_pred_mean
hist(residuals, main = "Residuals Distribution", xlab = "Residuals")
```




```
# Model Evaluation
mse <- mean(residuals^2)
cat("Mean Squared Error (MSE):", mse, "\n")
```

```
Mean Squared Error (MSE): 356291.9
```

```
rmse <- sqrt(mse)
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

```
Root Mean Squared Error (RMSE): 596.9019
```

```
sst <- sum((y_obs - mean(y_obs))^2)
sse <- sum(residuals^2)
r_squared <- 1 - (sse / sst)
cat("R-squared (R²):", r_squared, "\n")
```

```
R-squared (R²): -0.6554296
```

```
# Posterior Predictive P-value (PPP)
ppp <- mean(abs(y_pred_all - mean(y_pred_all)) >= abs(y_obs - mean(y_obs)))
cat("Posterior Predictive P-value (PPP):", ppp, "\n")
```

```
Posterior Predictive P-value (PPP): 0.2571257
```

Final Report: Investigating the Impact of AQI Factors on Brain Cancer Incidence in the U.S.

Objective:

This project aimed to explore the association between environmental, occupational, and lifestyle factors—specifically focusing on Air Quality Index (AQI) variables—and brain cancer incidence in the United States from 1999 to 2021. The goal was to understand how AQI features might contribute to the overall incidence of brain cancer across various states using a Bayesian statistical approach.

Data:

The dataset included several variables that could influence brain cancer rates:

- **AQI-related factors**: These included average AQI values, the number of days with different

AQI levels, and other air quality metrics.

- **Brain cancer incidence data**: The target variable was the total count of brain cancer cases across different states.

Given time constraints, the project primarily focused on AQI-related factors, but additional data on **drinking water quality**, **general radiation exposure**, **hazardous occupational exposure**, and **pesticide use distributions** could substantially enhance the model's explanatory power. These factors are likely to provide a clearer picture of how environmental and lifestyle factors, aside from air quality, contribute to brain cancer incidence.

Modeling Approach:

- **Bayesian Poisson Regression**: A Bayesian Poisson regression model was employed to estimate the relationship between AQI-related predictors and brain cancer incidence. The Poisson model was selected because brain cancer data, like many health-related counts, often follow a Poisson distribution, especially for counts of rare events like cancer cases.
- **Bayesian Framework**: The Bayesian approach was particularly relevant because it allows for the incorporation of prior knowledge and uncertainty into the modeling process. Given the complexity of cancer incidence and the many contributing factors, the Bayesian framework provided a natural way to quantify uncertainty in model parameters and make probabilistic statements about the effects of AQI-related variables on brain cancer incidence. This approach also facilitated the use of posterior predictive checks to assess model fit and allowed for more flexibility in capturing uncertainty, as opposed to traditional frequentist methods.

Model Performance:

- **Mean Squared Error (MSE)**: 369,453.8
- **Root Mean Squared Error (RMSE)**: 607.8
- **R-squared (R^2)**: -0.7165833, indicating that the model explained very little of the variance, which is expected given the complexity of the data and the focus on a single set of predictors.
- **Posterior Predictive P-value (PPP)**: 0.3142738, which suggested that the model's predictive accuracy was reasonable, though there is significant room for improvement.

Findings and Insights:

1. **Impact of AQI-related Factors**: The regression coefficients from the model suggested that certain AQI-related factors, like `avg_max_aqi` (maximum AQI), had a moderate effect on brain cancer incidence, with a negative association. This suggests that higher AQI values, which typically indicate worse air quality, might correlate with a lower incidence of brain cancer in some states. ***However, this finding is counterintuitive and requires further investigation, potentially involving other environmental and lifestyle factors.***

- **`avg_max_aqi`**: Negative impact on brain cancer incidence.

- **`avg_days_with_aqi`**: Also exhibited a negative relationship, indicating that the nu

2. **Model Fit**:

- While the model showed some predictive ability, as indicated by the posterior predictive

3. **Future Improvements**:
    - **Additional Factors**: Incorporating data on **drinking water quality**, **radiation exposure**, **occupational hazards**, and **pesticide use** would likely improve model performance

and provide more precise estimates of the effects of environmental and lifestyle factors on brain cancer incidence. These factors may influence brain cancer in ways that AQI alone cannot explain.

- **Longer Data Timeframes**: A more robust dataset with longer timeframes and more granularity would provide better insights, particularly with regard to time-lag effects (e.g., how long after exposure to certain pollutants might brain cancer incidence rise).

4. **Conclusion**:

The results from the Bayesian Poisson regression model suggest that while there are some associations between AQI-related factors and brain cancer incidence, the overall model fit is weak, and significant uncertainty remains. The model's predictive performance was moderate, but the low R^2 value indicates that additional variables and more complex models are needed to capture the full extent of the factors influencing brain cancer.

The Bayesian approach proved to be useful in this context as it allowed for the incorporation of prior knowledge and a flexible treatment of uncertainty, both of which are crucial in modeling a complex and multifaceted issue like cancer incidence. While AQI-related factors were explored in depth, incorporating additional environmental and occupational data in future studies could lead to a more comprehensive understanding of how various factors contribute to brain cancer incidence.

Further research, particularly in the form of more detailed datasets and refined models, will be necessary to pinpoint more accurately the environmental, lifestyle, and occupational risks associated with brain cancer.

Supporting Work - 1
Advanced Data Cleaning and Building the Final Dataset

Data Consolidation and Visualization

```
-- 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 becor

Attaching package: 'zoo'


The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

AQI Data - Averaging and Consolidating

```
aqi1 <- read.csv("annual_aqi_by_county_2006.csv")
colnames(aqi1)
```

```
 [1] "State"                                "County"
 [3] "Year"                                 "Days.with.AQI"
 [5] "Good.Days"                            "Moderate.Days"
 [7] "Unhealthy.for.Sensitive.Groups.Days" "Unhealthy.Days"
 [9] "Very.Unhealthy.Days"                  "Hazardous.Days"
[11] "Max.AQI"                              "X90th.Percentile.AQI"
[13] "Median.AQI"                           "Days.CO"
[15] "Days.NO2"                             "Days.Ozone"
[17] "Days.PM2.5"                           "Days.PM10"
```

```
# aqi1
```

```
library(dplyr)
library(readr)
library(janitor)
```

```
Attaching package: 'janitor'
```

```
The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```r
library(tidyr)


process_year_data <- function(year) {
  if (year == 2020) {
    return(NULL)
  }


  file_name <- paste0("annual_aqi_by_county_", year, ".csv")


  data <- read_csv(file_name, show_col_types = FALSE) %>%
    clean_names()


  required_columns <- c("max_aqi", "x90th_percentile_aqi", "median_aqi",
                        "days_with_aqi", "good_days",
                        "moderate_days",
                        "unhealthy_for_sensitive_groups_days",
                        "unhealthy_days","very_unhealthy_days",
                        "hazardous_days",
                        "days_co", "days_no2", "days_ozone",
                        "days_pm2_5", "days_pm10")


  missing_columns <- setdiff(required_columns, colnames(data))
  if (length(missing_columns) > 0) {
    warning("Missing columns in ", year, ": ",
            paste(missing_columns, collapse = ", "))
    return(NULL)
  }

  state_avg_aqi <- data %>%
    group_by(state) %>%
    summarise(
      avg_max_aqi = mean(max_aqi,
                         na.rm = TRUE),
      avg_x90th_percentile_aqi = mean(x90th_percentile_aqi,
                                      na.rm = TRUE),
      avg_median_aqi = mean(median_aqi,
                            na.rm = TRUE),
      avg_days_with_aqi = mean(days_with_aqi,
                               na.rm = TRUE),
      avg_good_days = mean(good_days,
```

```
                            na.rm = TRUE),
      avg_moderate_days = mean(moderate_days,
                               na.rm = TRUE),
      avg_unhealthy_for_sensitive_groups_days = mean
      (unhealthy_for_sensitive_groups_days, na.rm = TRUE),
      avg_unhealthy_days = mean(unhealthy_days,
                                na.rm = TRUE),
      avg_very_unhealthy_days = mean(very_unhealthy_days,
                                     na.rm = TRUE),
      avg_hazardous_days = mean(hazardous_days,
                                na.rm = TRUE),
      avg_days_co = mean(days_co,
                         na.rm = TRUE),
      avg_days_no2 = mean(days_no2,
                          na.rm = TRUE),
      avg_days_ozone = mean(days_ozone,
                            na.rm = TRUE),
      avg_days_pm2_5 = mean(days_pm2_5,
                            na.rm = TRUE),
      avg_days_pm10 = mean(days_pm10,
                           na.rm = TRUE)
    ) %>%
    mutate(year = year)

  return(state_avg_aqi)
}

# Process data for all years, excluding 2020
years <- setdiff(1999:2021, 2020)

all_years_data <- lapply(years, function(year) {
  data <- process_year_data(year)
  if (!is.null(data)) {
    return(data)
  }
}) %>%
  bind_rows()

# Create imputed 2020 data by averaging other years' data
impute_2020_data <- all_years_data %>%
  group_by(state) %>%
  summarise(
    avg_max_aqi = mean(avg_max_aqi,
                       na.rm = TRUE),
    avg_x90th_percentile_aqi = mean(avg_x90th_percentile_aqi,
                                    na.rm = TRUE),
    avg_median_aqi = mean(avg_median_aqi,
                          na.rm = TRUE),
```

```
    avg_days_with_aqi = mean(avg_days_with_aqi,
                             na.rm = TRUE),
    avg_good_days = mean(avg_good_days,
                         na.rm = TRUE),
    avg_moderate_days = mean(avg_moderate_days,
                             na.rm = TRUE),
    avg_unhealthy_for_sensitive_groups_days = mean
    (avg_unhealthy_for_sensitive_groups_days, na.rm = TRUE),
    avg_unhealthy_days = mean(avg_unhealthy_days,
                              na.rm = TRUE),
    avg_very_unhealthy_days = mean(avg_very_unhealthy_days,
                                   na.rm = TRUE),
    avg_hazardous_days = mean(avg_hazardous_days,
                              na.rm = TRUE),
    avg_days_co = mean(avg_days_co,
                       na.rm = TRUE),
    avg_days_no2 = mean(avg_days_no2,
                        na.rm = TRUE),
    avg_days_ozone = mean(avg_days_ozone,
                          na.rm = TRUE),
    avg_days_pm2_5 = mean(avg_days_pm2_5,
                          na.rm = TRUE),
    avg_days_pm10 = mean(avg_days_pm10,
                         na.rm = TRUE)
  ) %>%
  mutate(year = 2020)

# Combine the original data
aqi_final_data <- bind_rows(all_years_data, impute_2020_data)

write_csv(aqi_final_data, "state_avg_aqi_1999_2021_with_imputed_2020.csv")
```

```
ncol(aqi_final_data)
```

```
[1] 17
```

```
summary(aqi_final_data)
```

```
    state            avg_max_aqi      avg_x90th_percentile_aqi avg_median_aqi
 Length:1251        Min.   :  58.25   Min.   : 30.50           Min.   :15.25
 Class :character   1st Qu.: 101.03   1st Qu.: 56.13           1st Qu.:35.40
 Mode  :character   Median : 120.13   Median : 63.00           Median :40.95
                    Mean   : 133.11   Mean   : 66.93           Mean   :40.49
                    3rd Qu.: 146.17   3rd Qu.: 74.94           3rd Qu.:45.46
                    Max.   :1046.67   Max.   :141.00           Max.   :74.67
 avg_days_with_aqi avg_good_days   avg_moderate_days
 Min.   : 32.0     Min.   :  8.0   Min.   :  5.50
 1st Qu.:251.8     1st Qu.:153.4   1st Qu.: 62.44
 Median :288.2     Median :192.5   Median : 85.25
 Mean   :283.6     Mean   :189.5   Mean   : 85.80
```

```
 3rd Qu.:329.6     3rd Qu.:225.7   3rd Qu.:106.45
 Max.   :366.0     Max.   :356.2   Max.   :277.00
 avg_unhealthy_for_sensitive_groups_days avg_unhealthy_days
 Min.   : 0.000                          Min.   : 0.00000
 1st Qu.: 1.181                          1st Qu.: 0.04762
 Median : 3.688                          Median : 0.30450
 Mean   : 6.634                          Mean   : 1.42964
 3rd Qu.: 9.342                          3rd Qu.: 1.37798
 Max.   :58.667                          Max.   :26.33333
 avg_very_unhealthy_days avg_hazardous_days  avg_days_co
 Min.   : 0.00000        Min.   :0.00000    Min.   : 0.0000
 1st Qu.: 0.00000        1st Qu.:0.00000    1st Qu.: 0.0000
 Median : 0.00000        Median :0.00000    Median : 0.1333
 Mean   : 0.21024        Mean   :0.05039    Mean   : 3.9655
 3rd Qu.: 0.07596        3rd Qu.:0.00000    3rd Qu.: 1.9683
 Max.   :15.00000        Max.   :2.66667    Max.   :75.1333
  avg_days_no2      avg_days_ozone   avg_days_pm2_5   avg_days_pm10
 Min.   :  0.0000   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
 1st Qu.:  0.6085   1st Qu.: 95.27   1st Qu.: 69.16   1st Qu.:  0.50
 Median :  3.1333   Median :146.17   Median :104.27   Median :  8.00
 Mean   :  8.7656   Mean   :136.86   Mean   :115.55   Mean   : 18.49
 3rd Qu.: 10.0000   3rd Qu.:179.55   3rd Qu.:151.88   3rd Qu.: 23.52
 Max.   :143.0000   Max.   :291.00   Max.   :346.00   Max.   :173.50
      year
 Min.   :1999
 1st Qu.:2004
 Median :2010
 Mean   :2010
 3rd Qu.:2016
 Max.   :2021
```

```
str(aqi_final_data)
```

```
tibble [1,251 x 17] (S3: tbl_df/tbl/data.frame)
 $ state                                 : chr [1:1251] "Alabama" "Alaska" "Arizona" "Arkans
 $ avg_max_aqi                           : num [1:1251] 146 107 126 113 222 ...
 $ avg_x90th_percentile_aqi              : num [1:1251] 93.6 51.7 79.4 81.1 103.4 ...
 $ avg_median_aqi                        : num [1:1251] 54.4 23.7 48.8 55.4 51.6 ...
 $ avg_days_with_aqi                     : num [1:1251] 184 194 221 119 328 ...
 $ avg_good_days                         : num [1:1251] 69.8 167.7 104.2 69.3 177.1 ...
 $ avg_moderate_days                     : num [1:1251] 88.4 24.7 91.4 41.9 95.7 ...
 $ avg_unhealthy_for_sensitive_groups_days: num [1:1251] 20 1.5 23.33 6.39 36.91 ...
 $ avg_unhealthy_days                    : num [1:1251] 5.667 0.333 2.167 1.278 16.696 ...
 $ avg_very_unhealthy_days               : num [1:1251] 0.429 0 0 0 1.821 ...
 $ avg_hazardous_days                    : num [1:1251] 0 0 0 0 0.196 ...
 $ avg_days_co                           : num [1:1251] 2.238 49.333 0.25 0.833 3.339 ...
 $ avg_days_no2                          : num [1:1251] 0 0 12.67 2.11 45.95 ...
 $ avg_days_ozone                        : num [1:1251] 93.2 59.7 145.9 85.7 220.6 ...
 $ avg_days_pm2_5                        : num [1:1251] 71.5 50.2 37.8 30.2 45.8 ...
 $ avg_days_pm10                         : num [1:1251] 17.333 35 24.5 0.167 12.786 ...
```

```
 $ year                                  : num [1:1251] 1999 1999 1999 1999 1999 ...
```

Loading the "Cancer Incidence" Data

```
# Load the data
cancer_incidence <- read.csv("cancer_incidence.csv")

# Convert 'Count' and 'Population' to numeric
cancer_incidence$Count <- as.numeric(cancer_incidence$Count)
```

```
Warning: NAs introduced by coercion
```

```
cancer_incidence$Population <- as.numeric(cancer_incidence$Population)
```

```
Warning: NAs introduced by coercion
```

```
# Remove 'Crude.Rate' column
cancer_incidence <- cancer_incidence %>%
  select(-Crude.Rate)

# Aggregate data by State, Year (across both sexes)
cancer_aggregated <- cancer_incidence %>%
  group_by(States, Year) %>%
  summarise(
    Total_Count = sum(Count, na.rm = TRUE),
    Total_Population = sum(Population, na.rm = TRUE)
  )
```

```
`summarise()` has grouped output by 'States'. You can override using the
`.groups` argument.
```

```
# View the resulting aggregated data
head(cancer_aggregated)
```

```
# A tibble: 6 x 4
# Groups:   States [1]
  States   Year Total_Count Total_Population
  <chr>   <int>       <dbl>            <dbl>
1 Alabama  1999          41           194723
2 Alabama  2000          51           220789
3 Alabama  2001          79           442183
4 Alabama  2002          34           378534
5 Alabama  2003          53           474259
6 Alabama  2004          54           313205
```

Ensuring Data Integrity and Processing the Next Data Set

```
aqi_data <- aqi_final_data
can_in <- cancer_aggregated


aqi_data$year <- as.integer(as.character(aqi_data$year))
```

```r
can_in$Year <- as.integer(as.character(can_in$Year))


names(aqi_data)[names(aqi_data) == "state"] <- "States"
names(can_in)[names(can_in) == "States"] <- "States"


can_in_complete <- can_in %>%
  mutate(
    Total_Count = ifelse(is.na(Total_Count), 0, Total_Count),
    Total_Population = ifelse
    (is.na(Total_Population), 0, Total_Population)
  )

final_merged_data <- left_join(aqi_data, can_in_complete,
                               by = c("States", "year" = "Year"))

head(final_merged_data)
```

```
# A tibble: 6 x 19
  States     avg_max_aqi avg_x90th_percentile~1 avg_median_aqi avg_days_with_aqi
  <chr>            <dbl>                  <dbl>          <dbl>              <dbl>
1 Alabama           146.                   93.6           54.4               184.
2 Alaska            107                    51.7           23.7               194.
3 Arizona           126.                   79.4           48.8               221.
4 Arkansas          113.                   81.1           55.4               119.
5 California        222.                  103.            51.6               328.
6 Canada            133                    47             34                 188
# i abbreviated name: 1: avg_x90th_percentile_aqi
# i 14 more variables: avg_good_days <dbl>, avg_moderate_days <dbl>,
#   avg_unhealthy_for_sensitive_groups_days <dbl>, avg_unhealthy_days <dbl>,
#   avg_very_unhealthy_days <dbl>, avg_hazardous_days <dbl>, avg_days_co <dbl>,
#   avg_days_no2 <dbl>, avg_days_ozone <dbl>, avg_days_pm2_5 <dbl>,
#   avg_days_pm10 <dbl>, year <int>, Total_Count <dbl>, Total_Population <dbl>
```

```r
str(final_merged_data)
```

```
tibble [1,251 x 19] (S3: tbl_df/tbl/data.frame)
 $ States                                : chr [1:1251] "Alabama" "Alaska" "Arizona" "Arkans
 $ avg_max_aqi                           : num [1:1251] 146 107 126 113 222 ...
 $ avg_x90th_percentile_aqi              : num [1:1251] 93.6 51.7 79.4 81.1 103.4 ...
 $ avg_median_aqi                        : num [1:1251] 54.4 23.7 48.8 55.4 51.6 ...
 $ avg_days_with_aqi                     : num [1:1251] 184 194 221 119 328 ...
 $ avg_good_days                         : num [1:1251] 69.8 167.7 104.2 69.3 177.1 ...
 $ avg_moderate_days                     : num [1:1251] 88.4 24.7 91.4 41.9 95.7 ...
 $ avg_unhealthy_for_sensitive_groups_days: num [1:1251] 20 1.5 23.33 6.39 36.91 ...
 $ avg_unhealthy_days                    : num [1:1251] 5.667 0.333 2.167 1.278 16.696 ...
 $ avg_very_unhealthy_days               : num [1:1251] 0.429 0 0 0 1.821 ...
 $ avg_hazardous_days                    : num [1:1251] 0 0 0 0 0.196 ...
```

```
 $ avg_days_co                          : num [1:1251] 2.238 49.333 0.25 0.833 3.339 ...
 $ avg_days_no2                         : num [1:1251] 0 0 12.67 2.11 45.95 ...
 $ avg_days_ozone                       : num [1:1251] 93.2 59.7 145.9 85.7 220.6 ...
 $ avg_days_pm2_5                       : num [1:1251] 71.5 50.2 37.8 30.2 45.8 ...
 $ avg_days_pm10                        : num [1:1251] 17.333 35 24.5 0.167 12.786 ...
 $ year                                 : int [1:1251] 1999 1999 1999 1999 1999 1999 1999 1
 $ Total_Count                          : num [1:1251] 41 0 93 0 2028 ...
 $ Total_Population                     : num [1:1251] 194723 261961 475824 196611 3185892
```

```
write_csv(final_merged_data, "merged_aqi_cancer_incidence.csv")
```

Loading Environmental Hazard Data

```
narrowr <- read.csv("narrowresult.csv")
str(narrowr)
```

```
'data.frame':   273014 obs. of  23 variables:
 $ OrganizationIdentifier                          : chr  "AK-CHIN_WQX" "AK-CHIN_WQX" "AK-CHIN_
 $ OrganizationFormalName                          : chr  "Ak-Chin Indian Community (Tribal)" "
 $ ActivityIdentifier                              : chr  "AK-CHIN_WQX-SR:SD-23:2013-10-28" "AK
 $ ActivityStartDate                               : chr  "28/10/2013" "17/12/2013" "30/09/2013
 $ ResultDetectionConditionText                    : chr  "" "" "" "Not Reported" ...
 $ MethodSpecificationName                         : chr  "" "" "" "" ...
 $ CharacteristicName                              : chr  "Calcium" "Calcium" "Calcium" "Chlorc
 $ ResultSampleFractionText                        : chr  "Fixed" "Fixed" "Fixed" "" ...
 $ ResultMeasureValue                              : chr  "65.2" "56.3" "81.7" "" ...
 $ ResultMeasure.MeasureUnitCode                   : chr  "mg/L" "mg/L" "mg/L" "" ...
 $ ResultStatusIdentifier                          : chr  "Final" "Final" "Final" "Final" ...
 $ ResultValueTypeName                             : chr  "Actual" "Actual" "Actual" "Actual" "
 $ PrecisionValue                                  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ DataQuality.BiasValue                           : logi  NA NA NA NA NA NA ...
 $ USGSPCode                                       : int  NA NA NA NA NA NA NA NA NA NA ...
 $ ResultDepthHeightMeasure.MeasureValue           : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ResultDepthHeightMeasure.MeasureUnitCode        : chr  "" "" "" "" ...
 $ ResultDepthAltitudeReferencePointText           : chr  "" "" "" "" ...
 $ ResultSamplingPointName                         : chr  "" "" "" "" ...
 $ ResultAnalyticalMethod.MethodName               : chr  "Nitrate-Nitrite Nitrogen by Cd Reduc
 $ ResultAnalyticalMethod.MethodQualifierTypeName: chr  "" "" "" "" ...
 $ AnalysisStartDate                               : chr  "" "" "" "" ...
 $ AnalysisEndDate                                 : chr  "" "" "" "" ...
```

```
unique(narrowr$OrganizationFormalName)
```

```
  [1] "Ak-Chin Indian Community (Tribal)"
  [2] "ALABAMA DEPT. OF ENVIRONMENTAL MANAGEMENT - WATER QUALITY DATA"
  [3] "Animas River Stakeholders Group (Colorado) (Volunteer)"
  [4] "Arkansas Department of Environmental Quality"
  [5] "Big Valley Band of Pomo Indians of the Big Valley Rancheria, California (Tribal)"
  [6] "Boomsnub/Airco Superfund Site EPA Region 10"
  [7] "Bunker Hill Mining and Metallurgical Complex"
  [8] "Bureau of Reclamation"
  [9] "California Department Of Water Resources"
```

```
[10] "California Gulch (US EPA Region 8)"
[11] "California State Water Resources Control Board"
[12] "Captain Jack Mine (Colorado)"
[13] "CBS Operations Inc."
[14] "CDA TRUST"
[15] "CITY OF MARCO ISLAND"
[16] "Clear Creek Watershed Foundation (CCWF) (Volunteer)"
[17] "Coal Creek Watershed Coalition (Colorado)"
[18] "Collier County Coastal Zone Management Department (FL)"
[19] "Collier County Pollution Control (Florida)"
[20] "Colorado Dept. of Public Health & Environment-WQCD"
[21] "Colorado Division of Reclamation, Mining and Safety (DRMS) (Volunteer)"
[22] "Colorado Mountain College Natural Resource Management"
[23] "Colorado River Watch"
[24] "Connecticut Department Of Energy And Environmental Protection"
[25] "Cortina Rancheria (Kletsel Dehe Wintun Nation) (Tribal)"
[26] "Dade Environmental Resource Management (Florida)"
[27] "Division of Surface water (Ohio)"
[28] "EA Engineering, Science and Technology Inc."
[29] "EPA National Aquatic Resources Survey (NARS)"
[30] "EPA Region 10 Boomsnub Superfund Site Data 1987-2013"
[31] "EPA Region 10 Superfund Bunker Hill Mining and Metallurgical Complex"
[32] "EPA Region 4 Athens Lab (Georgia)"
[33] "FDEP GROUNDWATER MANAGEMENT SECTION"
[34] "FDEP TALLAHASSEE REGIONAL OPERATIONS CENTER"
[35] "FL Dept. of Environmental Protection"
[36] "FL Dept. of Environmental Protection, Northwest District"
[37] "Flandreau Santee Sioux Tribe (SD)"
[38] "Hopi Tribe of Arizona (Tribal)"
[39] "illinois epa"
[40] "Indiana STORET"
[41] "Jamestown S'Klallam Tribe (Tribal)"
[42] "Kickapoo Tribe of Indians of the Kickapoo Reservation in Kansas (Tribal)"
[43] "Lake County Water Resource Management"
[44] "Lake Fork Watershed Stakeholders (Colorado) (Volunteer)"
[45] "Massachusetts Department of Environmental Protection (MassDEP)"
[46] "Maul Foster and Alongi, Inc."
[47] "MBMG_WQX - Montana Bureau of Mines and Geology"
[48] "Midnite Mine Environmental Data"
[49] "Minnesota Pollution Control Agency - Ambient Surface Water"
[50] "Missouri Dept. of Natural Resources"
[51] "Montana DEQ WQPB"
[52] "Montana PPL Corporation"
[53] "Montana Volunteer Water Quality Monitoring"
[54] "Montana Watershed"
[55] "Morongo Band of Mission Indians (Tribal)"
[56] "Muckleshoot Indian Tribe (Tribal)"
[57] "National Park Service Water Resources Division"
[58] "Navajo Nation, Arizona, New Mexico & Utah (Tribal)"
```

```
 [59] "Nevada Division of Environmental Protection"
 [60] "New York State Dec Division Of Water"
 [61] "NM Environmental Dept./SWQB"
 [62] "North Dakota Department Of Environmental Quality"
 [63] "OCC - Otter Creek Coal"
 [64] "Oneida Nation"
 [65] "P4 Production LLC, Soda Springs Plant, Idaho"
 [66] "Palermo Wellfield Superfund Site by Geoengineers Inc. (Volunteer)*"
 [67] "Perry Co. Soil and Water District"
 [68] "Pueblo of Sandia Water Quality Program (New Mexico)"
 [69] "Red Lake DNR"
 [70] "Region 8 Superfund:  Standard Mine"
 [71] "Rhode Island"
 [72] "Salt Chuck Mine, State of Alaska"
 [73] "San Miguel Watershed Coalition (Volunteer)*"
 [74] "Santee Sioux Nation of Nebraska (Tribal)"
 [75] "Schuylkill Action Network (Pennsylvania)"
 [76] "Seminole Tribe of Florida (Tribal)"
 [77] "Shoalwater Bay Indian Tribe of the Shoalwater Bay Indian Reservation (Tribal)"
 [78] "Skagit County"
 [79] "Snoqualmie Indian Tribe (Tribal)"
 [80] "South Carolina Department of Environmental Services"
 [81] "Southwest Florida Water Management District"
 [82] "Spokane Tribe of the Spokane Reservation (Tribal)"
 [83] "State of Oregon Dept. of Environmental Quality"
 [84] "State of Wyoming Department of Environmental Quality Watershed Program"
 [85] "Suwannee River Water Management District"
 [86] "Table Mountain Rancheria of California (Tribal)"
 [87] "Tacoma-Pierce County Health Department (Washington)"
 [88] "TDEC Division of Water Resources"
 [89] "TerraGraphics Environmental Engineering, Inc."
 [90] "Texas Commission on Environmental Quality"
 [91] "Twenty-Nine Palms Tribal EPA"
 [92] "UD Citizen Monitoring Program"
 [93] "Uncompahgre Watershed Partnership (Volunteer)*"
 [94] "USEPA Region 9"
 [95] "USGS Florida Water Science Center"
 [96] "USGS Kansas Water Science Center"
 [97] "USGS Montana Water Science Center"
 [98] "USGS New Mexico Water Science Center"
 [99] "USGS Oregon Water Science Center"
[100] "Utah Department Of Environmental Quality"
[101] "Ute Mountain Utes Tribe (Colorado)"
[102] "VIRGINIA DEPARTMENT OF ENVIRONMENTAL QUALITY"
[103] "West Virginia Department of Environmental Protection Watershed Improvement Branch"
[104] "West Virginia Department of Environmental Protection-Division of Water & Waste Manager
[105] "Wind River Environmental Quality Commission"
[106] "Wisconsin Department of Natural Resources"
[107] "WV Div of Environmental Protection, Office of Water Resource"
```

```r
# List of U.S. state names
states <- c(
  "Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado",
  "Connecticut", "Delaware", "Florida", "Georgia", "Hawaii", "Idaho",
  "Illinois", "Indiana","Iowa", "Kansas", "Kentucky", "Louisiana", "Maine",
  "Maryland", "Massachusetts", "Michigan", "Minnesota", "Mississippi",
  "Missouri", "Montana", "Nebraska","Nevada", "New Hampshire", "New Jersey",
  "New Mexico", "New York", "North Carolina","North Dakota", "Ohio",
  "Oklahoma", "Oregon", "Pennsylvania", "Rhode Island", "South Carolina",
  "South Dakota", "Tennessee", "Texas", "Utah", "Vermont","Virginia",
  "Washington", "West Virginia", "Wisconsin", "Wyoming"
)

# Regular expression pattern to match state names
state_pattern <- str_c(states, collapse = "|")


narrowr <- narrowr %>%
  mutate(
    # Handle blank or missing values first
    OrganizationFormalName = ifelse(is.na(OrganizationFormalName) |
                                      OrganizationFormalName == "", "Unknown",
                                    OrganizationFormalName),
    # Extract the state name if it exists in the organization name
    State = str_extract(OrganizationFormalName, state_pattern),
    # Replace organization name with state name if a match is found
    OrganizationFormalName = ifelse(!is.na(State), State,
                                    OrganizationFormalName)
  )
narrowr$state <- narrowr$OrganizationFormalName
# View the updated dataset
head(narrowr)
```

```
  OrganizationIdentifier            OrganizationFormalName
1            AK-CHIN_WQX Ak-Chin Indian Community (Tribal)
2            AK-CHIN_WQX Ak-Chin Indian Community (Tribal)
3            AK-CHIN_WQX Ak-Chin Indian Community (Tribal)
4            AK-CHIN_WQX Ak-Chin Indian Community (Tribal)
5            AK-CHIN_WQX Ak-Chin Indian Community (Tribal)
6            AK-CHIN_WQX Ak-Chin Indian Community (Tribal)
               ActivityIdentifier ActivityStartDate
1 AK-CHIN_WQX-SR:SD-23:2013-10-28        28/10/2013
2 AK-CHIN_WQX-SR:SD-23:2013-12-17        17/12/2013
3  AK-CHIN_WQX-SR:SD-23:2013-9-30        30/09/2013
4 AK-CHIN_WQX-SR:SD-23:2013-10-28        28/10/2013
5 AK-CHIN_WQX-SR:SD-23:2013-10-28        28/10/2013
6  AK-CHIN_WQX-SR:SD-23:2013-9-30        30/09/2013
  ResultDetectionConditionText MethodSpecificationName CharacteristicName
1                                                                 Calcium
```

```
2                                                                  Calcium
3                                                                  Calcium
4                Not Reported                                Chlorophyll a
5                                                              Potassium
6                                                                Sodium
  ResultSampleFractionText ResultMeasureValue ResultMeasure.MeasureUnitCode
1                    Fixed              65.2                         mg/L
2                    Fixed              56.3                         mg/L
3                    Fixed              81.7                         mg/L
4
5                                        6.5                         mg/L
6                    Fixed               114                         mg/L
  ResultStatusIdentifier ResultValueTypeName PrecisionValue
1                  Final              Actual             NA
2                  Final              Actual             NA
3                  Final              Actual             NA
4                  Final              Actual             NA
5                  Final              Actual             NA
6                  Final              Actual             NA
  DataQuality.BiasValue USGSPCode ResultDepthHeightMeasure.MeasureValue
1                    NA        NA                                   NA
2                    NA        NA                                   NA
3                    NA        NA                                   NA
4                    NA        NA                                   NA
5                    NA        NA                                   NA
6                    NA        NA                                   NA
  ResultDepthHeightMeasure.MeasureUnitCode
1
2
3
4
5
6
  ResultDepthAltitudeReferencePointText ResultSamplingPointName
1
2
3
4
5
6
                                ResultAnalyticalMethod.MethodName
1                      Nitrate-Nitrite Nitrogen by Cd Reduction
2                      Nitrate-Nitrite Nitrogen by Cd Reduction
3                      Nitrate-Nitrite Nitrogen by Cd Reduction
4                          Nitrite Nitrogen by Spectophotometry
5                          Nitrite Nitrogen by Spectophotometry
6 DO NOT USE***4500 NH3 C ~ Ammonia in Water by Titrimetric Method
  ResultAnalyticalMethod.MethodQualifierTypeName AnalysisStartDate
1
```

```
2
3
4
5
6                              duplicate records
  AnalysisEndDate State                                 state
1                  <NA> Ak-Chin Indian Community (Tribal)
2                  <NA> Ak-Chin Indian Community (Tribal)
3                  <NA> Ak-Chin Indian Community (Tribal)
4                  <NA> Ak-Chin Indian Community (Tribal)
5                  <NA> Ak-Chin Indian Community (Tribal)
6                  <NA> Ak-Chin Indian Community (Tribal)
```

```
colnames(narrowr)
```

```
 [1] "OrganizationIdentifier"
 [2] "OrganizationFormalName"
 [3] "ActivityIdentifier"
 [4] "ActivityStartDate"
 [5] "ResultDetectionConditionText"
 [6] "MethodSpecificationName"
 [7] "CharacteristicName"
 [8] "ResultSampleFractionText"
 [9] "ResultMeasureValue"
[10] "ResultMeasure.MeasureUnitCode"
[11] "ResultStatusIdentifier"
[12] "ResultValueTypeName"
[13] "PrecisionValue"
[14] "DataQuality.BiasValue"
[15] "USGSPCode"
[16] "ResultDepthHeightMeasure.MeasureValue"
[17] "ResultDepthHeightMeasure.MeasureUnitCode"
[18] "ResultDepthAltitudeReferencePointText"
[19] "ResultSamplingPointName"
[20] "ResultAnalyticalMethod.MethodName"
[21] "ResultAnalyticalMethod.MethodQualifierTypeName"
[22] "AnalysisStartDate"
[23] "AnalysisEndDate"
[24] "State"
[25] "state"
```

```
str(narrowr$ActivityStartDate)
```

```
 chr [1:273014] "28/10/2013" "17/12/2013" "30/09/2013" "28/10/2013" ...
```

```
str(narrowr$AnalysisStartDate)
```

```
 chr [1:273014] "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ...
```

```
str(narrowr$AnalysisEndDate)
```

```
 chr [1:273014] "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ...
```

```r
library(lubridate)
library(dplyr)

processed_dataset <- narrowr %>%
  # Drop specified columns
  select(-c(
    OrganizationIdentifier, state, OrganizationFormalName,
    ResultDepthAltitudeReferencePointText,
    ResultSamplingPointName,
    ResultAnalyticalMethod.MethodName,
    ActivityIdentifier, USGSPCode,
    ResultAnalyticalMethod.MethodQualifierTypeName,
    ResultDetectionConditionText,
    MethodSpecificationName, ResultStatusIdentifier,
    ResultSampleFractionText
  )) %>%


  mutate(

    ActivityStartDate = ifelse(ActivityStartDate == "" |
                                 is.na(ActivityStartDate), NA,
                               ActivityStartDate),
    AnalysisStartDate = ifelse(AnalysisStartDate == "" |
                                 is.na(AnalysisStartDate), NA,
                               AnalysisStartDate),
    AnalysisEndDate = ifelse(AnalysisEndDate == "" |
                               is.na(AnalysisEndDate),
                             NA, AnalysisEndDate),

    # Parse the dates with flexible parsing for character data
    ActivityStartDate = parse_date_time
    (ActivityStartDate, orders = c("dmy", "mdy", "ymd")),
    AnalysisStartDate = parse_date_time
    (AnalysisStartDate, orders = c("dmy", "mdy", "ymd")),
    AnalysisEndDate = parse_date_time
    (AnalysisEndDate, orders = c("dmy", "mdy", "ymd"))
  ) %>%

  mutate(
    AnalysisYear = case_when(
      !is.na(AnalysisEndDate) ~ year(AnalysisEndDate),
      !is.na(AnalysisStartDate) ~ year(AnalysisStartDate),
      !is.na(ActivityStartDate) ~ year(ActivityStartDate),
      TRUE ~ NA_real_
    )
  ) %>%
```

```
  # Drop rows where AnalysisYear is NA
  filter(!is.na(AnalysisYear)) %>%

  # Drop original date columns
  select(-c(ActivityStartDate, AnalysisStartDate, AnalysisEndDate))
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `AnalysisEndDate = parse_date_time(AnalysisEndDate, orders =
  c("dmy", "mdy", "ymd"))`.
Caused by warning:
! All formats failed to parse. No formats found.
```

```
# View the processed dataset
str(processed_dataset)
```

```
'data.frame':   273014 obs. of  10 variables:
 $ CharacteristicName                     : chr  "Calcium" "Calcium" "Calcium" "Chlorophyll
 $ ResultMeasureValue                     : chr  "65.2" "56.3" "81.7" "" ...
 $ ResultMeasure.MeasureUnitCode          : chr  "mg/L" "mg/L" "mg/L" "" ...
 $ ResultValueTypeName                    : chr  "Actual" "Actual" "Actual" "Actual" ...
 $ PrecisionValue                         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ DataQuality.BiasValue                  : logi  NA NA NA NA NA NA ...
 $ ResultDepthHeightMeasure.MeasureValue  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ResultDepthHeightMeasure.MeasureUnitCode: chr  "" "" "" "" ...
 $ State                                  : chr  NA NA NA NA ...
 $ AnalysisYear                           : num  2013 2013 2013 2013 2013 ...
```

```
narrowrfilt <- processed_dataset %>%
  filter(rowSums(is.na(.) | . == "") < (ncol(processed_dataset) / 2))

# View the filtered dataset
head(narrowrfilt)
```

```
  CharacteristicName ResultMeasureValue ResultMeasure.MeasureUnitCode
1            Calcium               91.5                         mg/L
2          Magnesium                6.6                         mg/L
3            Calcium                 37                         mg/L
4            Calcium               90.5                         mg/L
5            Calcium               60.9                         mg/L
6          Magnesium                6.4                         mg/L
  ResultValueTypeName PrecisionValue DataQuality.BiasValue
1              Actual             NA                    NA
2              Actual             NA                    NA
3              Actual             NA                    NA
4              Actual             NA                    NA
5              Actual             NA                    NA
6              Actual             NA                    NA
  ResultDepthHeightMeasure.MeasureValue
1                                    NA
2                                    NA
3                                    NA
```

```
4                                      NA
5                                      NA
6                                      NA
  ResultDepthHeightMeasure.MeasureUnitCode    State AnalysisYear
1                                          Colorado         2015
2                                          Colorado         2015
3                                          Colorado         2015
4                                          Colorado         2015
5                                          Colorado         2015
6                                          Colorado         2015
```

```
nrow(narrowrfilt)
```

```
[1] 247307
```

```
colnames(narrowrfilt)
```

```
 [1] "CharacteristicName"
 [2] "ResultMeasureValue"
 [3] "ResultMeasure.MeasureUnitCode"
 [4] "ResultValueTypeName"
 [5] "PrecisionValue"
 [6] "DataQuality.BiasValue"
 [7] "ResultDepthHeightMeasure.MeasureValue"
 [8] "ResultDepthHeightMeasure.MeasureUnitCode"
 [9] "State"
[10] "AnalysisYear"
```

```
colnames(final_merged_data)
```

```
 [1] "States"
 [2] "avg_max_aqi"
 [3] "avg_x90th_percentile_aqi"
 [4] "avg_median_aqi"
 [5] "avg_days_with_aqi"
 [6] "avg_good_days"
 [7] "avg_moderate_days"
 [8] "avg_unhealthy_for_sensitive_groups_days"
 [9] "avg_unhealthy_days"
[10] "avg_very_unhealthy_days"
[11] "avg_hazardous_days"
[12] "avg_days_co"
[13] "avg_days_no2"
[14] "avg_days_ozone"
[15] "avg_days_pm2_5"
[16] "avg_days_pm10"
[17] "year"
[18] "Total_Count"
[19] "Total_Population"
```

```
colnames(narrowrfilt)
```

```
 [1] "CharacteristicName"
```

```
 [2] "ResultMeasureValue"
 [3] "ResultMeasure.MeasureUnitCode"
 [4] "ResultValueTypeName"
 [5] "PrecisionValue"
 [6] "DataQuality.BiasValue"
 [7] "ResultDepthHeightMeasure.MeasureValue"
 [8] "ResultDepthHeightMeasure.MeasureUnitCode"
 [9] "State"
[10] "AnalysisYear"
```

```r
final_merged_data <- final_merged_data %>%

  left_join(narrowrfilt, by = c("States" = "State", "year" = "AnalysisYear")) %>%
  mutate(across(everything(), ~replace(., is.na(.), "")))

# View the final merged dataset
head(final_merged_data)
```

```
# A tibble: 6 x 27
  States     avg_max_aqi avg_x90th_percentile~1 avg_median_aqi avg_days_with_aqi
  <chr>      <chr>       <chr>                  <chr>          <chr>
1 Alabama    145.523809~ 93.5714285714286       54.3809523809~ 184.238095238095
2 Alaska     107         51.6666666666667       23.6666666666~ 194.166666666667
3 Arizona    125.583333~ 79.4166666666667       48.75          221.166666666667
4 Arkansas   112.944444~ 81.0555555555556       55.4444444444~ 118.944444444444
5 California 222.321428~ 103.428571428571       51.5535714285~ 328.428571428571
6 California 222.321428~ 103.428571428571       51.5535714285~ 328.428571428571
# i abbreviated name: 1: avg_x90th_percentile_aqi
# i 22 more variables: avg_good_days <chr>, avg_moderate_days <chr>,
#   avg_unhealthy_for_sensitive_groups_days <chr>, avg_unhealthy_days <chr>,
#   avg_very_unhealthy_days <chr>, avg_hazardous_days <chr>, avg_days_co <chr>,
#   avg_days_no2 <chr>, avg_days_ozone <chr>, avg_days_pm2_5 <chr>,
#   avg_days_pm10 <chr>, year <chr>, Total_Count <chr>, Total_Population <chr>,
#   CharacteristicName <chr>, ResultMeasureValue <chr>, ...
```

```r
colnames(final_merged_data)
```

```
 [1] "States"
 [2] "avg_max_aqi"
 [3] "avg_x90th_percentile_aqi"
 [4] "avg_median_aqi"
 [5] "avg_days_with_aqi"
 [6] "avg_good_days"
 [7] "avg_moderate_days"
 [8] "avg_unhealthy_for_sensitive_groups_days"
 [9] "avg_unhealthy_days"
[10] "avg_very_unhealthy_days"
[11] "avg_hazardous_days"
[12] "avg_days_co"
[13] "avg_days_no2"
```

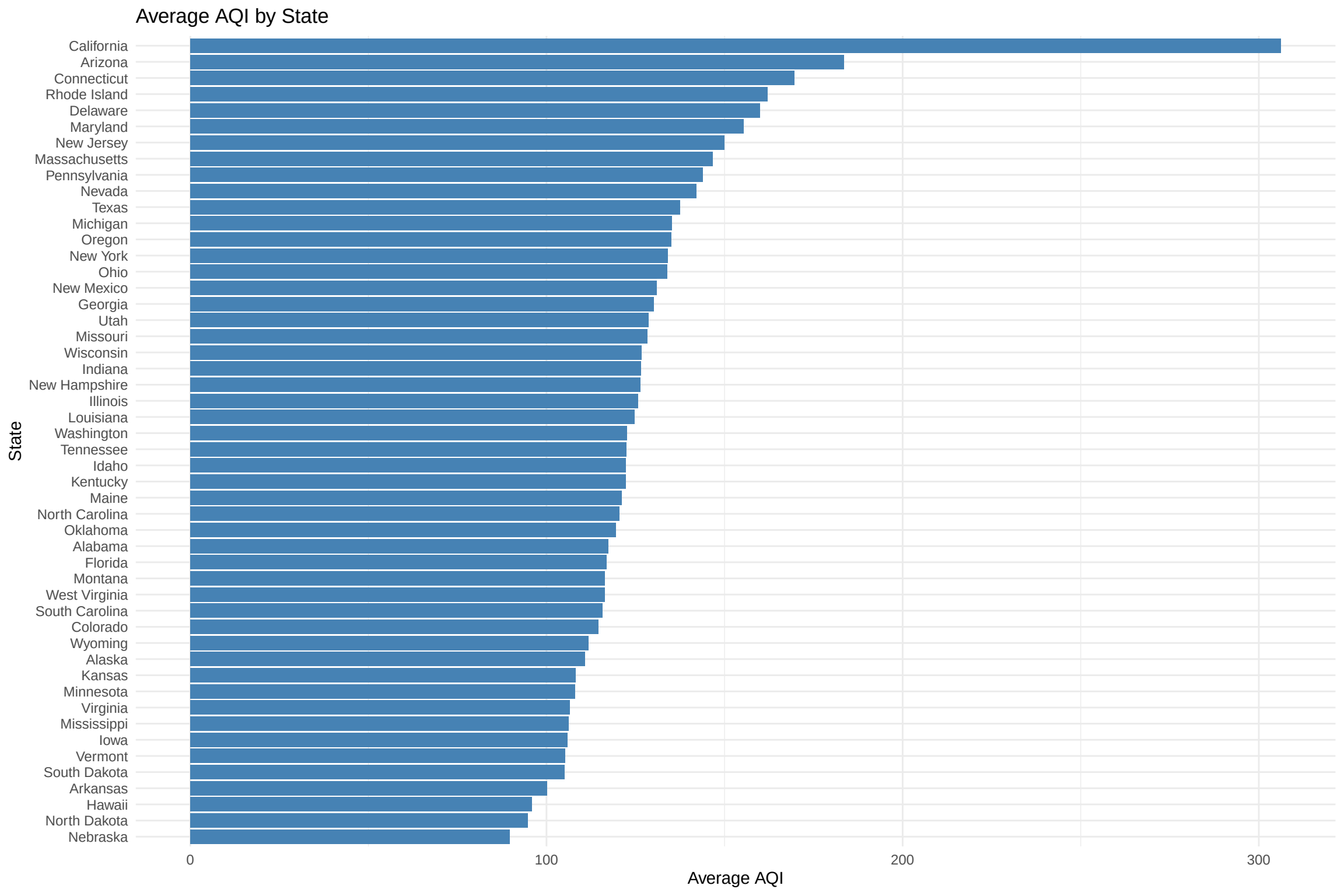
```
[14] "avg_days_ozone"
[15] "avg_days_pm2_5"
[16] "avg_days_pm10"
[17] "year"
[18] "Total_Count"
[19] "Total_Population"
[20] "CharacteristicName"
[21] "ResultMeasureValue"
[22] "ResultMeasure.MeasureUnitCode"
[23] "ResultValueTypeName"
[24] "PrecisionValue"
[25] "DataQuality.BiasValue"
[26] "ResultDepthHeightMeasure.MeasureValue"
[27] "ResultDepthHeightMeasure.MeasureUnitCode"
```

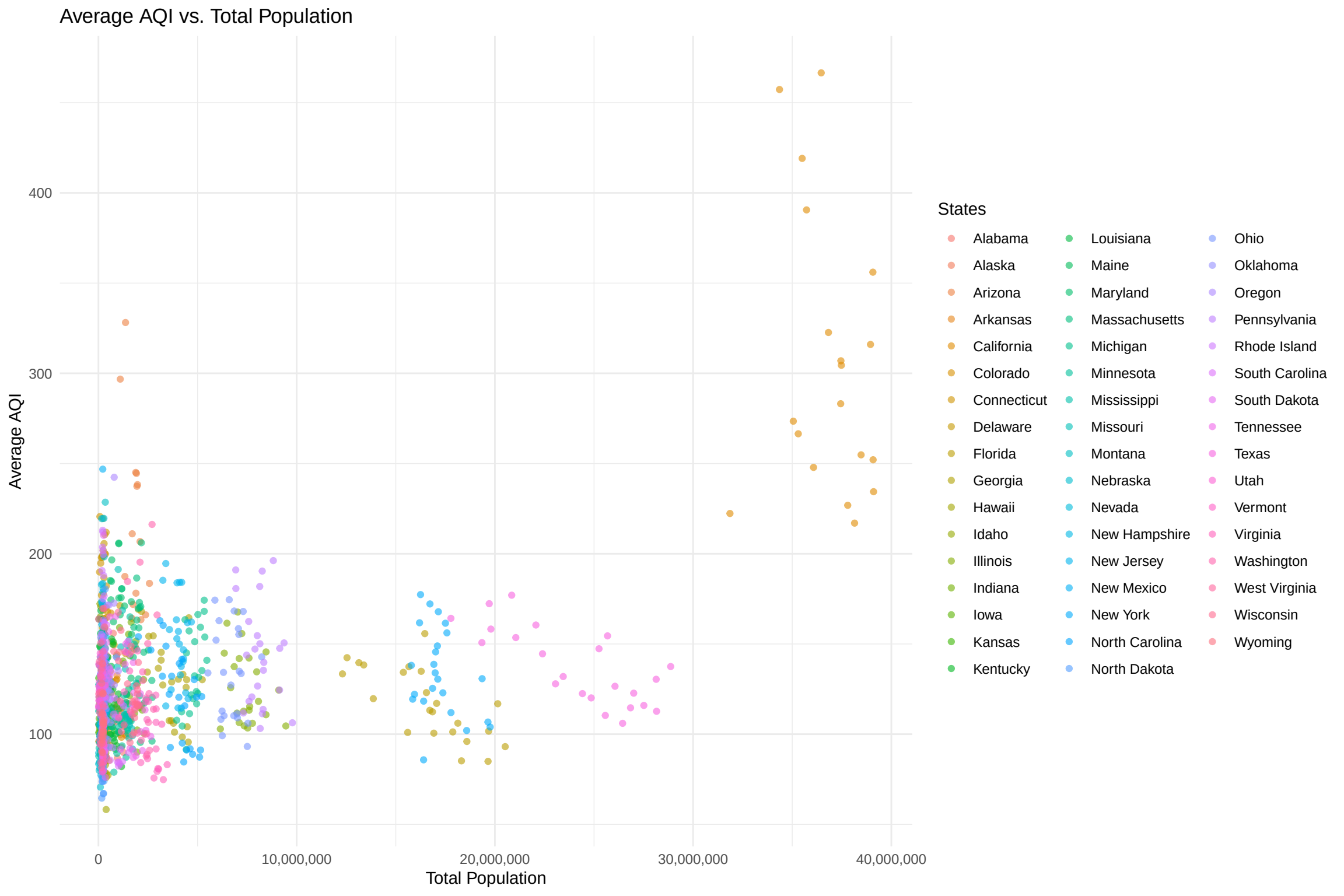
```
write_csv(final_merged_data, "final_dataset_consolidated.csv")
```

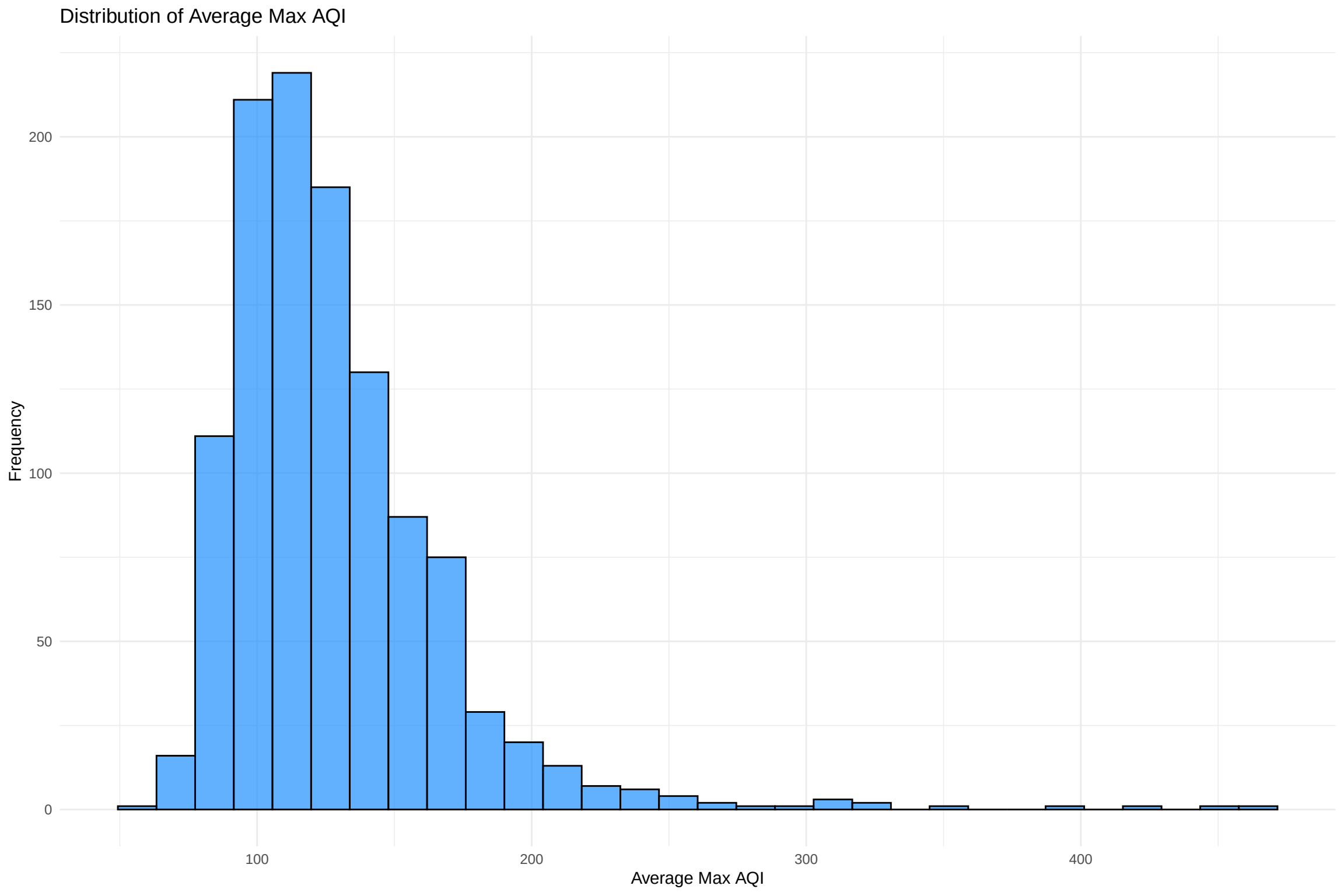
```
str(final_merged_data)
```

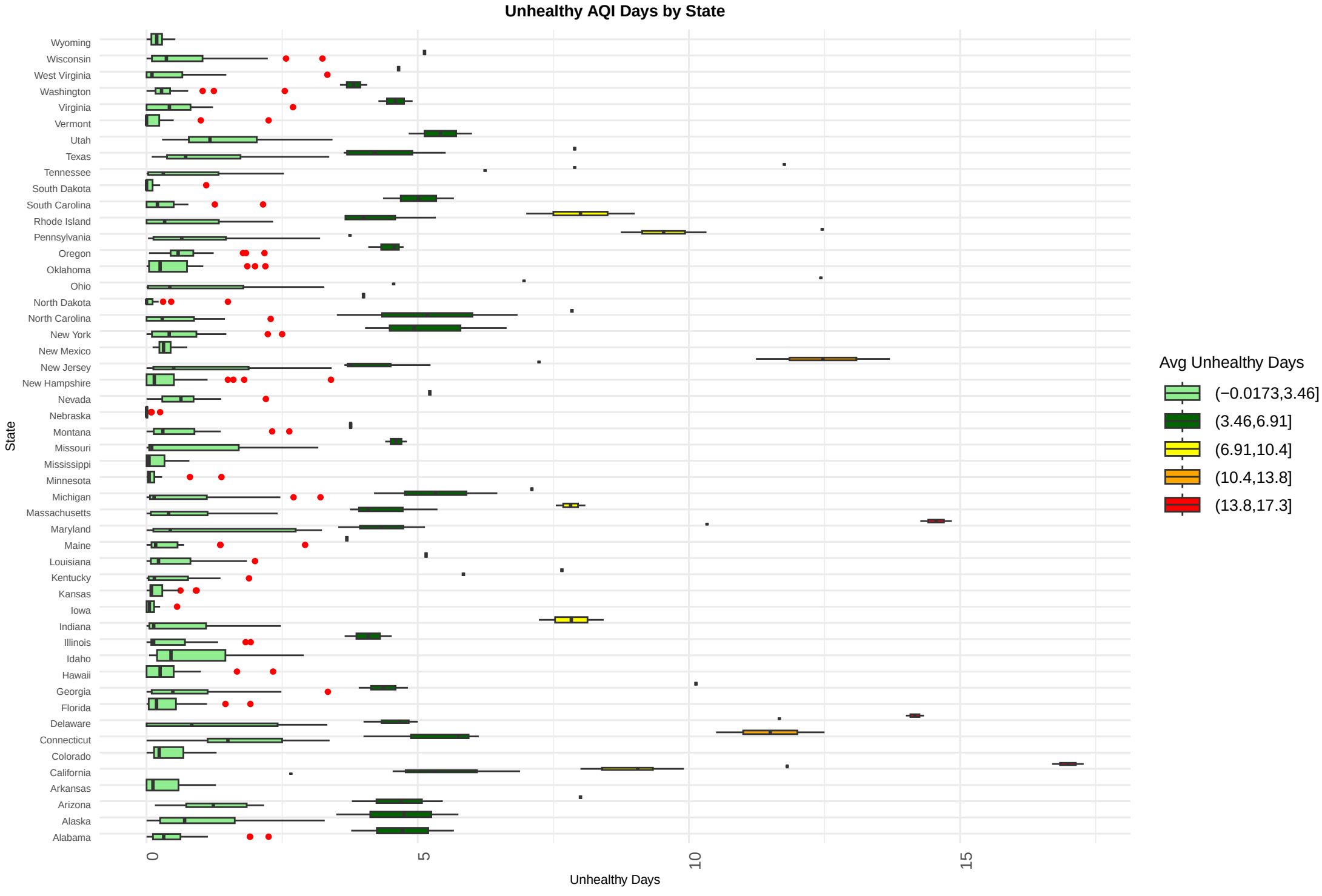
```
tibble [235,057 x 27] (S3: tbl_df/tbl/data.frame)
 $ States                                 : chr [1:235057] "Alabama" "Alaska" "Arizona" "Ark
 $ avg_max_aqi                            : chr [1:235057] "145.52380952381" "107" "125.5833
 $ avg_x90th_percentile_aqi               : chr [1:235057] "93.5714285714286" "51.666666666
 $ avg_median_aqi                         : chr [1:235057] "54.3809523809524" "23.666666666
 $ avg_days_with_aqi                      : chr [1:235057] "184.238095238095" "194.16666666
 $ avg_good_days                          : chr [1:235057] "69.7619047619048" "167.66666666
 $ avg_moderate_days                      : chr [1:235057] "88.3809523809524" "24.666666666
 $ avg_unhealthy_for_sensitive_groups_days : chr [1:235057] "20" "1.5" "23.3333333333333" "6
 $ avg_unhealthy_days                     : chr [1:235057] "5.66666666666667" "0.3333333333
 $ avg_very_unhealthy_days                : chr [1:235057] "0.428571428571429" "0" "0" "0"
 $ avg_hazardous_days                     : chr [1:235057] "0" "0" "0" "0" ...
 $ avg_days_co                            : chr [1:235057] "2.23809523809524" "49.333333333
 $ avg_days_no2                           : chr [1:235057] "0" "0" "12.6666666666667" "2.111
 $ avg_days_ozone                         : chr [1:235057] "93.1904761904762" "59.666666666
 $ avg_days_pm2_5                         : chr [1:235057] "71.4761904761905" "50.166666666
 $ avg_days_pm10                          : chr [1:235057] "17.3333333333333" "35" "24.5" "0
 $ year                                   : chr [1:235057] "1999" "1999" "1999" "1999" ...
 $ Total_Count                            : chr [1:235057] "41" "0" "93" "0" ...
 $ Total_Population                       : chr [1:235057] "194723" "261961" "475824" "19661
 $ CharacteristicName                     : chr [1:235057] "" "" "" "" ...
 $ ResultMeasureValue                     : chr [1:235057] "" "" "" "" ...
 $ ResultMeasure.MeasureUnitCode          : chr [1:235057] "" "" "" "" ...
 $ ResultValueTypeName                    : chr [1:235057] "" "" "" "" ...
 $ PrecisionValue                         : chr [1:235057] "" "" "" "" ...
 $ DataQuality.BiasValue                  : chr [1:235057] "" "" "" "" ...
 $ ResultDepthHeightMeasure.MeasureValue  : chr [1:235057] "" "" "" "" ...
 $ ResultDepthHeightMeasure.MeasureUnitCode: chr [1:235057] "" "" "" "" ...
```

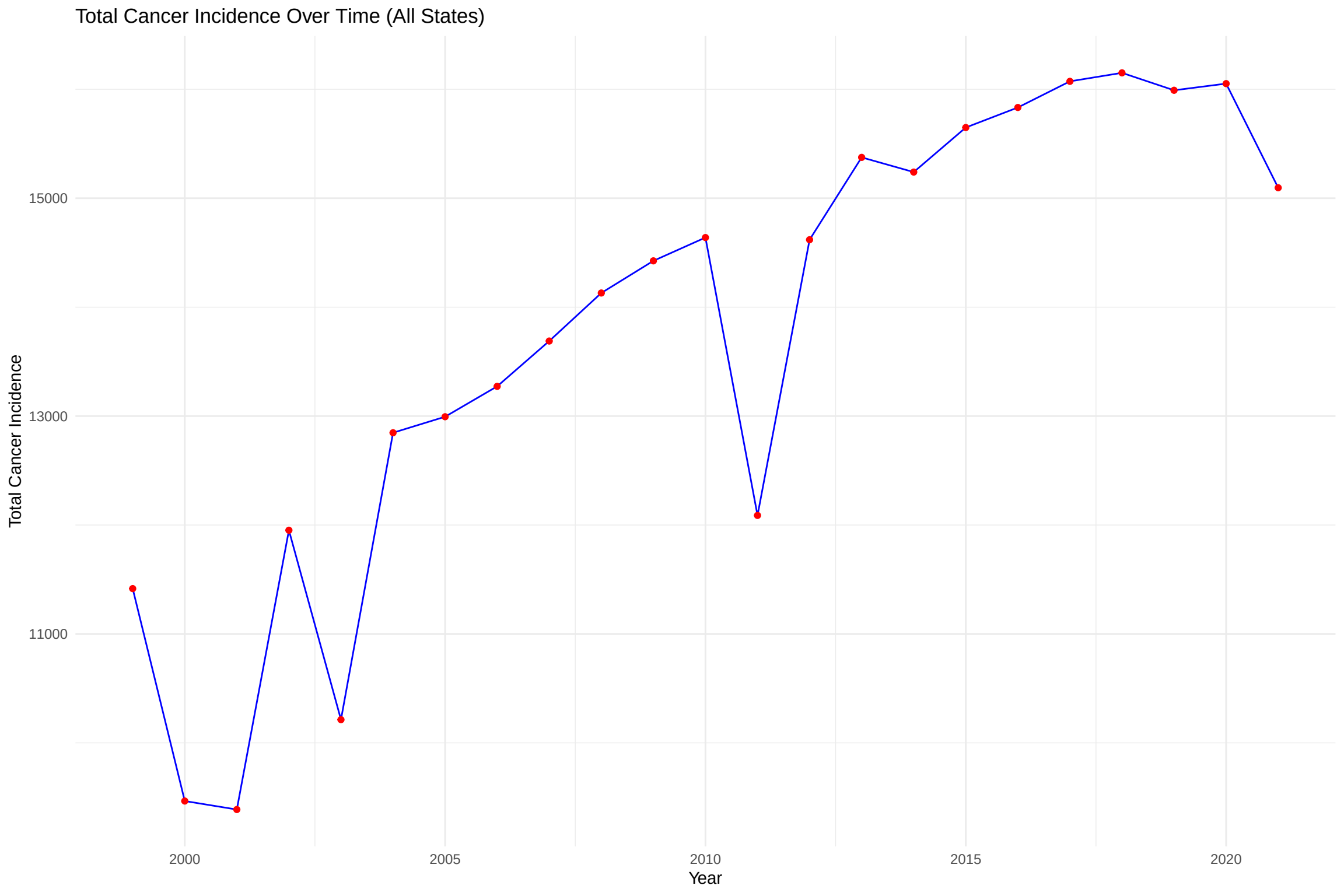
Supporting Work - 2
All Figures

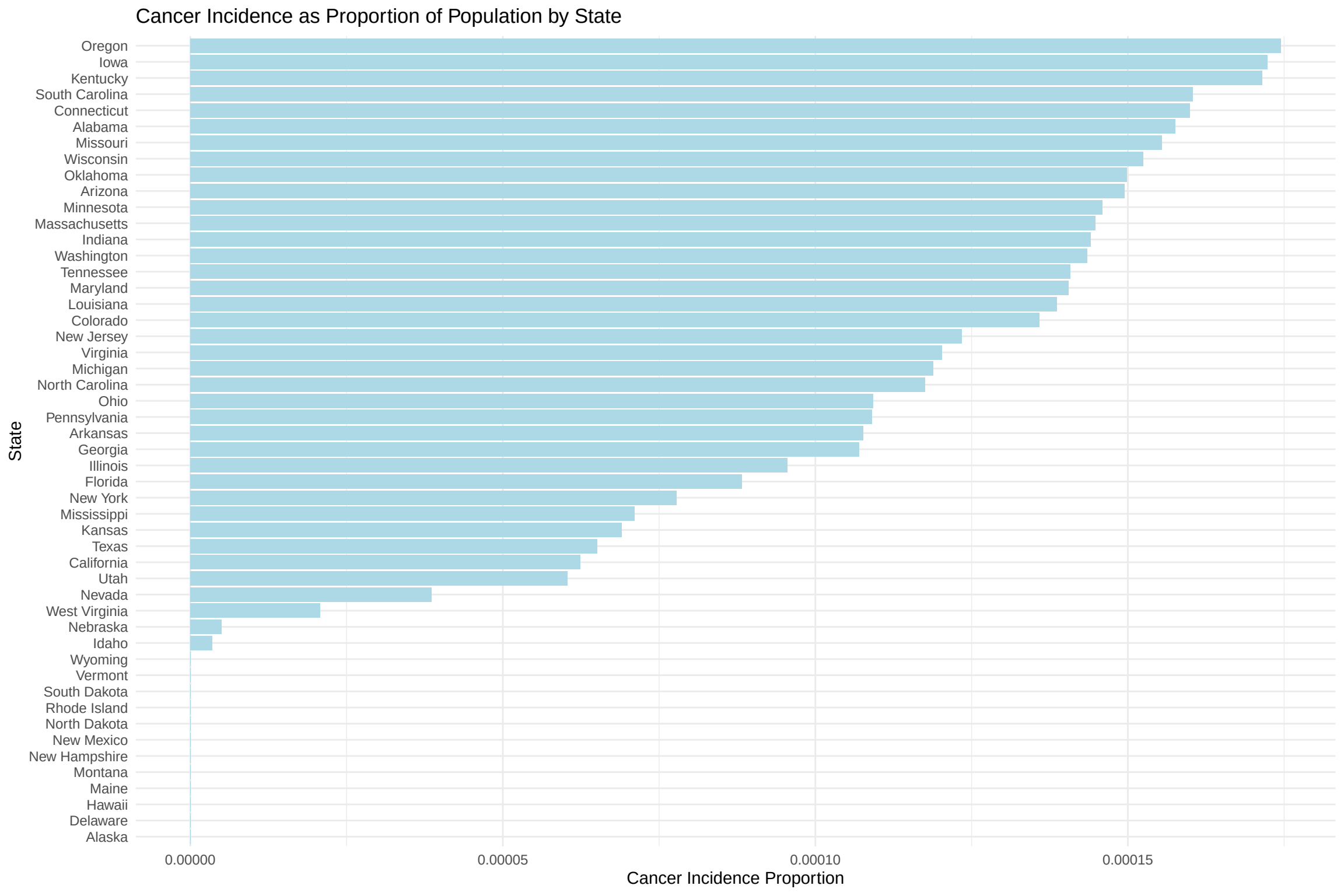

Average AQI by State
California
Arizona
Connecticut
Rhode Island
Delaware
Maryland
New Jersey
Massachusetts
Pennsylvania
Nevada
Texas
Michigan
Oregon
New York
Ohio
New Mexico
Georgia
Utah
Missouri
Wisconsin
Indiana
New Hampshire
Illinois
Louisiana
Washington
Tennessee
Idaho
Kentucky
Maine
North Carolina
Oklahoma
Alabama
Florida
Montana
West Virginia
South Carolina
Colorado
Wyoming
Alaska
Kansas
Minnesota
Virginia
Mississippi
Iowa
Vermont
South Dakota
Arkansas
Hawaii
North Dakota
Nebraska
State
0
100
200
300
Average AQI

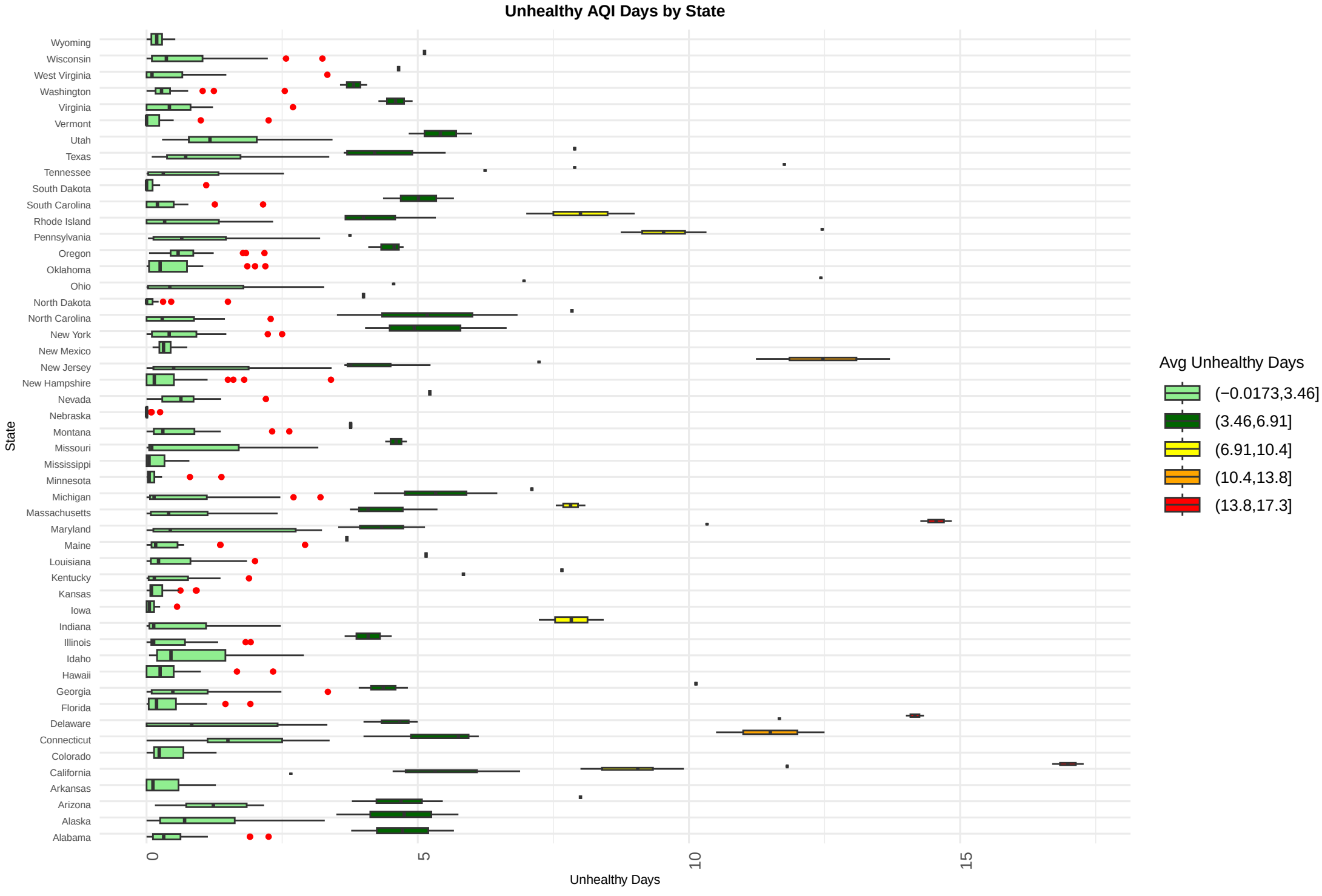

Average AQI vs. Total Population
Average AQI
400
300
200
100
0
10,000,000
20,000,000
30,000,000
40,000,000
Total Population
States
Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
Florida
Georgia
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Mississippi
Missouri
Montana
Nebraska
Nevada
New Hampshire
New Jersey
New Mexico
New York
North Carolina
North Dakota
Ohio
Oklahoma
Oregon
Pennsylvania
Rhode Island
South Carolina
South Dakota
Tennessee
Texas
Utah
Vermont
Virginia
Washington
West Virginia
Wisconsin
Wyoming

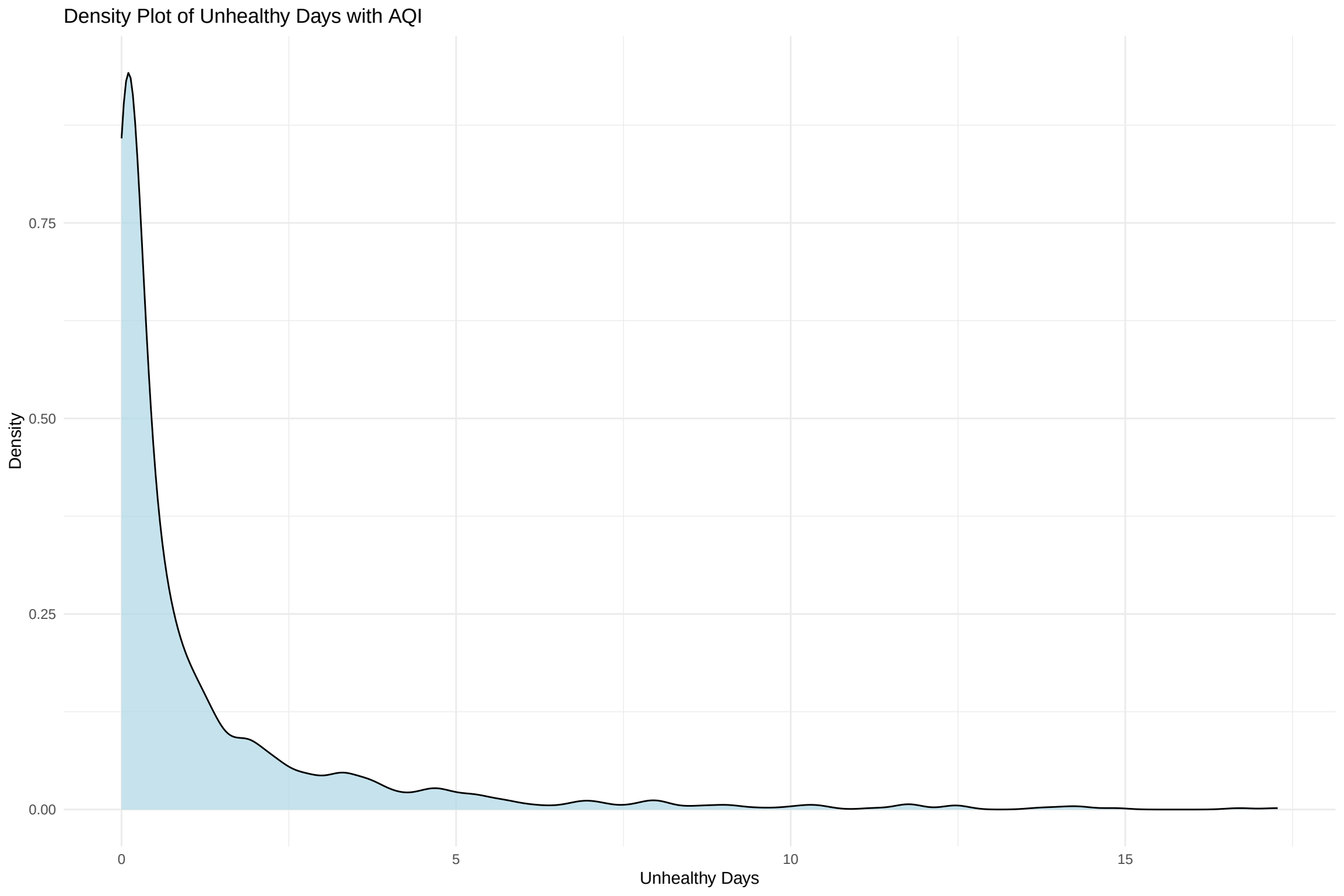

Distribution of Average Max AQI
Frequency
0
50
100
150
200
100
200
300
400
Average Max AQI

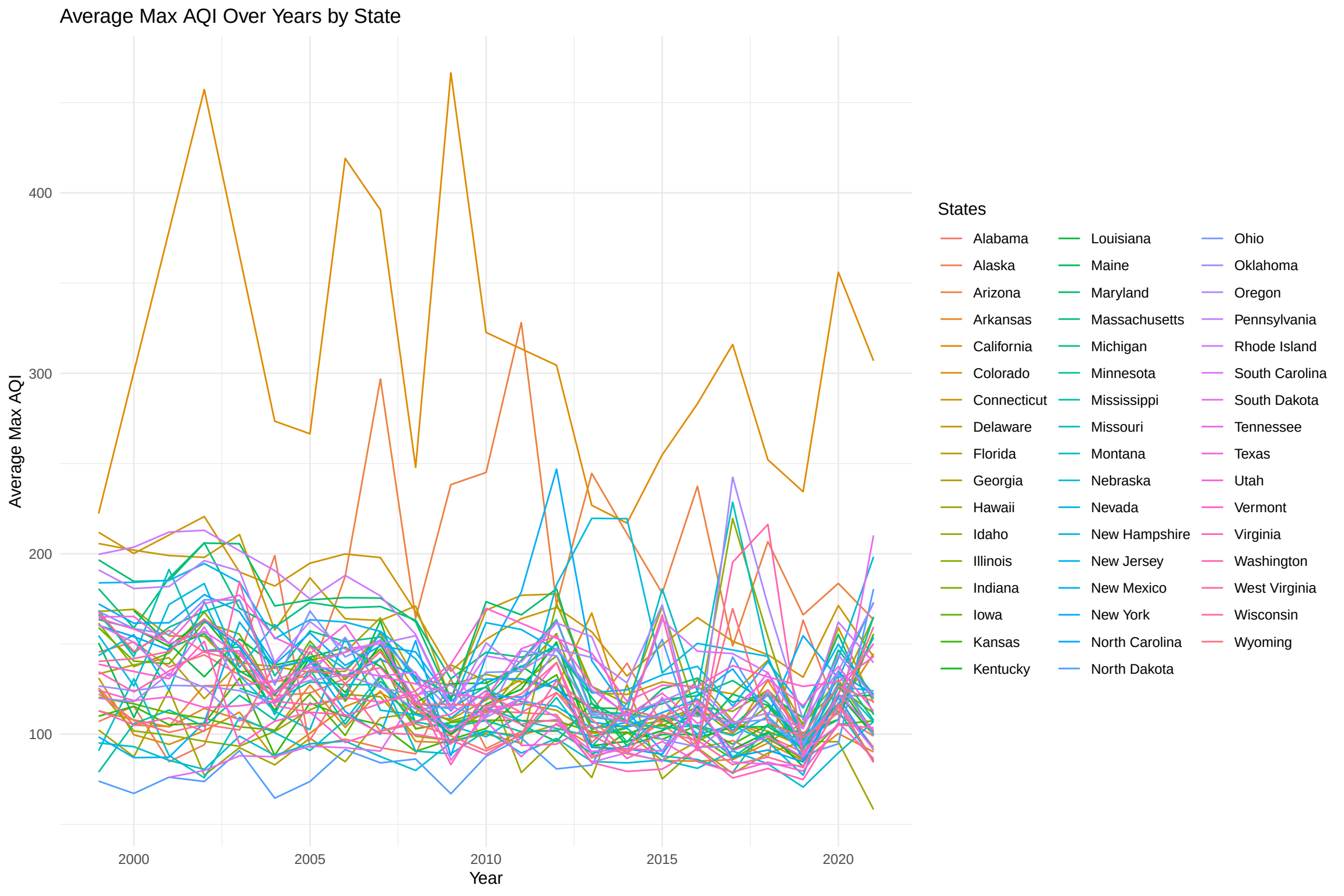

Unhealthy AQI Days by State
Wyoming
Wisconsin
West Virginia
Washington
Virginia
Vermont
Utah
Texas
Tennessee
South Dakota
South Carolina
Rhode Island
Pennsylvania
Oregon
Oklahoma
Ohio
North Dakota
North Carolina
New York
New Mexico
New Jersey
New Hampshire
Nevada
Nebraska
Montana
Missouri
Mississippi
Minnesota
Michigan
Massachusetts
Maryland
Maine
Louisiana
Kentucky
Kansas
Iowa
Indiana
Illinois
Idaho
Hawaii
Georgia
Florida
Delaware
Connecticut
Colorado
California
Arkansas
Arizona
Alaska
Alabama
State
0
5
10
15
Unhealthy Days
Avg Unhealthy Days
(−0.0173,3.46]
(3.46,6.91]
(6.91,10.4]
(10.4,13.8]
(13.8,17.3]

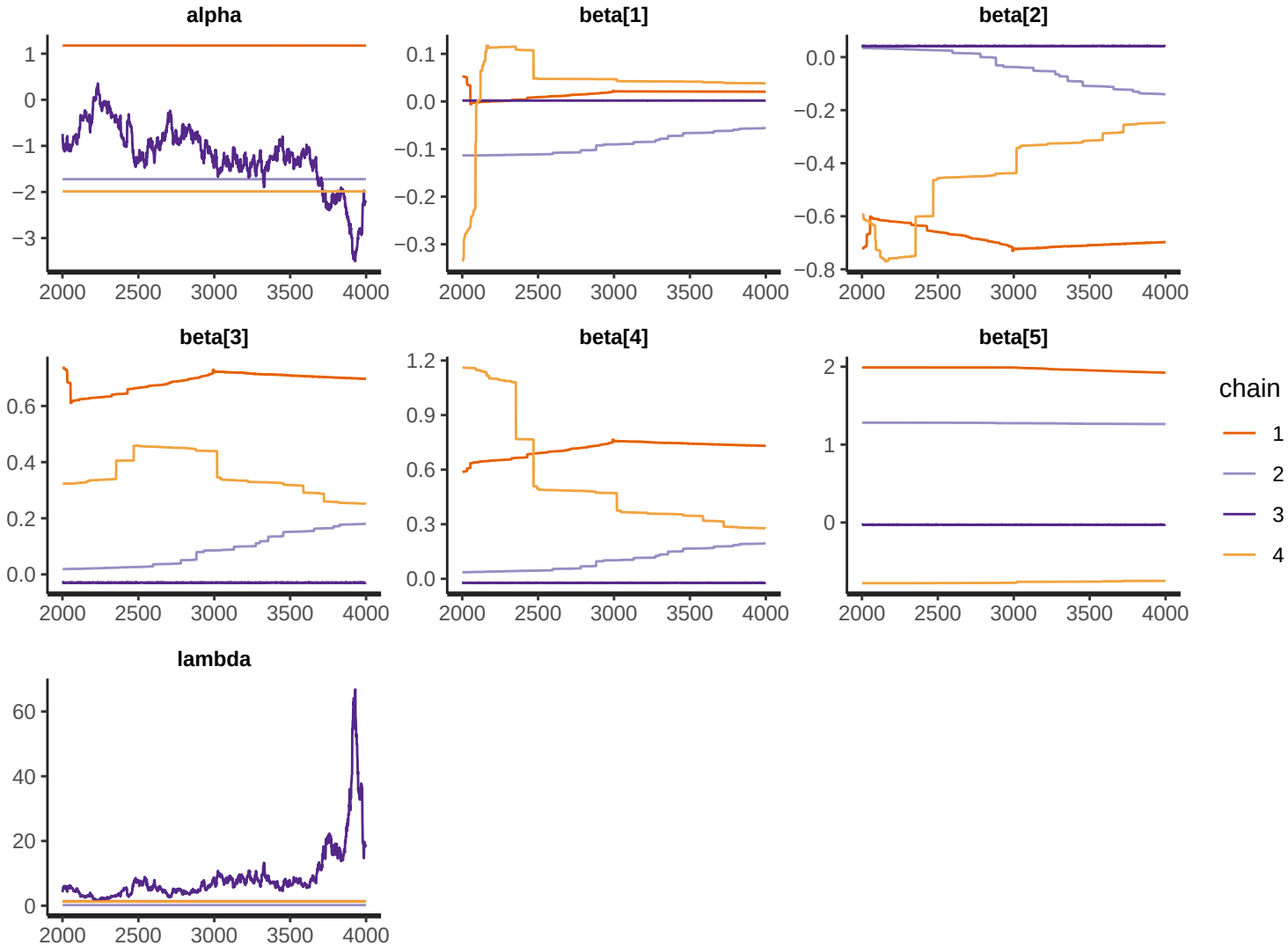

Total Cancer Incidence Over Time (All States)
Total Cancer Incidence
15000
13000
11000
2000
2005
2010
2015
2020
Year

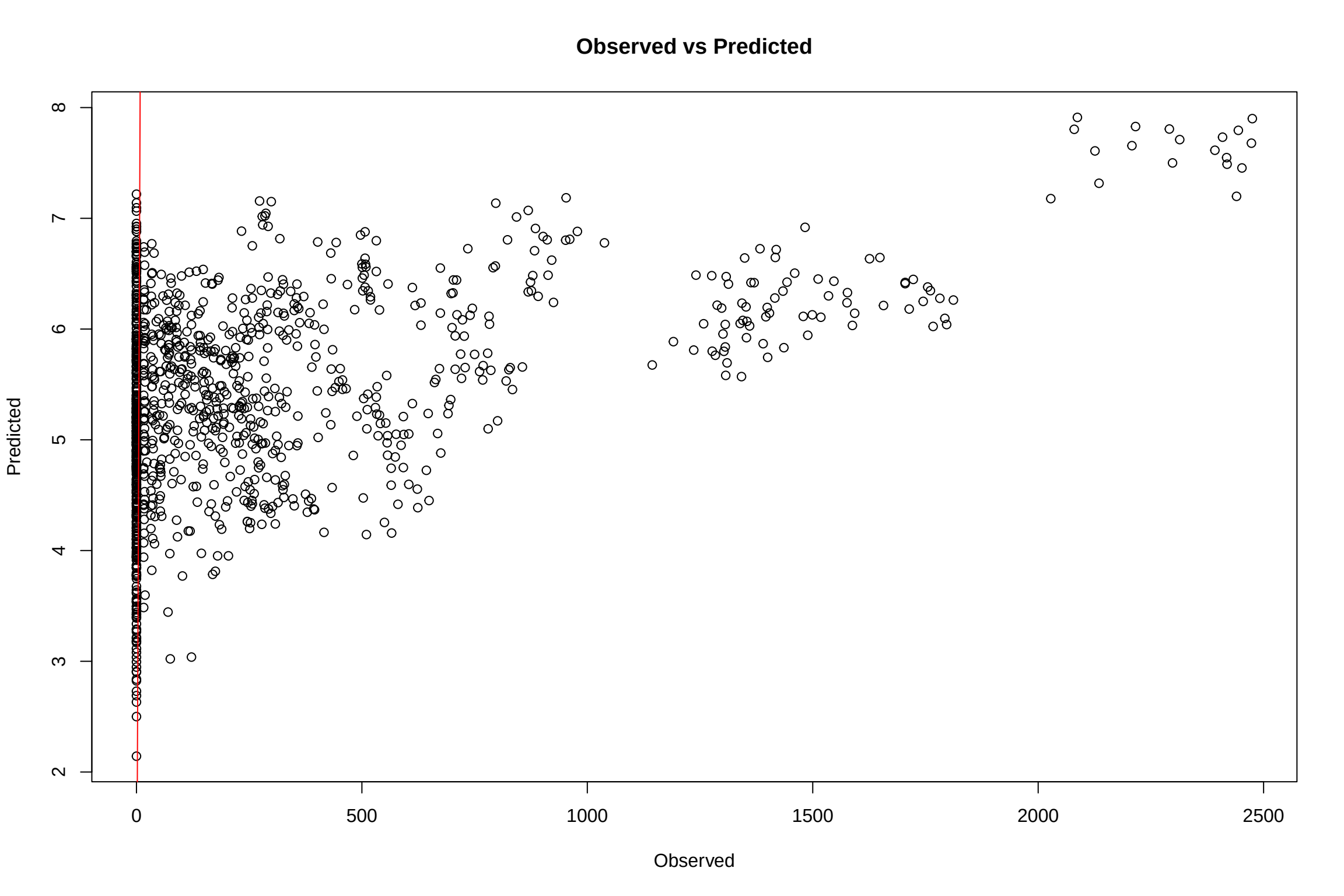

Cancer Incidence as Proportion of Population by State
Oregon
Iowa
Kentucky
South Carolina
Connecticut
Alabama
Missouri
Wisconsin
Oklahoma
Arizona
Minnesota
Massachusetts
Indiana
Washington
Tennessee
Maryland
Louisiana
Colorado
New Jersey
Virginia
Michigan
North Carolina
Ohio
Pennsylvania
Arkansas
Georgia
Illinois
Florida
New York
Mississippi
Kansas
Texas
California
Utah
Nevada
West Virginia
Nebraska
Idaho
Wyoming
Vermont
South Dakota
Rhode Island
North Dakota
New Mexico
New Hampshire
Montana
Maine
Hawaii
Delaware
Alaska
State
0.00000
0.00005
0.00010
0.00015
Cancer Incidence Proportion

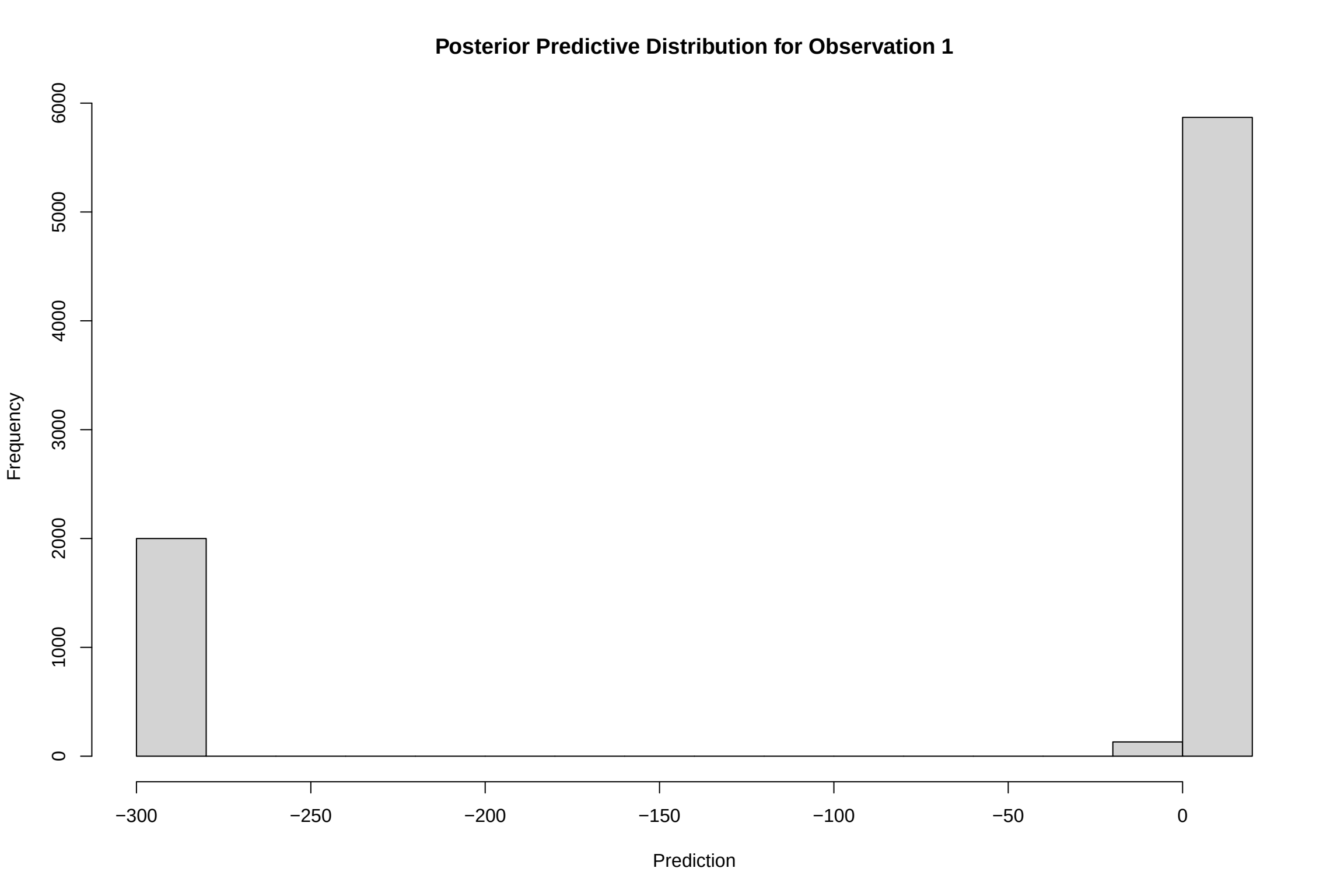

Unhealthy AQI Days by State
Wyoming
Wisconsin
West Virginia
Washington
Virginia
Vermont
Utah
Texas
Tennessee
South Dakota
South Carolina
Rhode Island
Pennsylvania
Oregon
Oklahoma
Ohio
North Dakota
North Carolina
New York
New Mexico
New Jersey
New Hampshire
Nevada
Nebraska
Montana
Missouri
Mississippi
Minnesota
Michigan
Massachusetts
Maryland
Maine
Louisiana
Kentucky
Kansas
Iowa
Indiana
Illinois
Idaho
Hawaii
Georgia
Florida
Delaware
Connecticut
Colorado
California
Arkansas
Arizona
Alaska
Alabama
State
0
5
10
15
Unhealthy Days
Avg Unhealthy Days
(−0.0173,3.46]
(3.46,6.91]
(6.91,10.4]
(10.4,13.8]
(13.8,17.3]

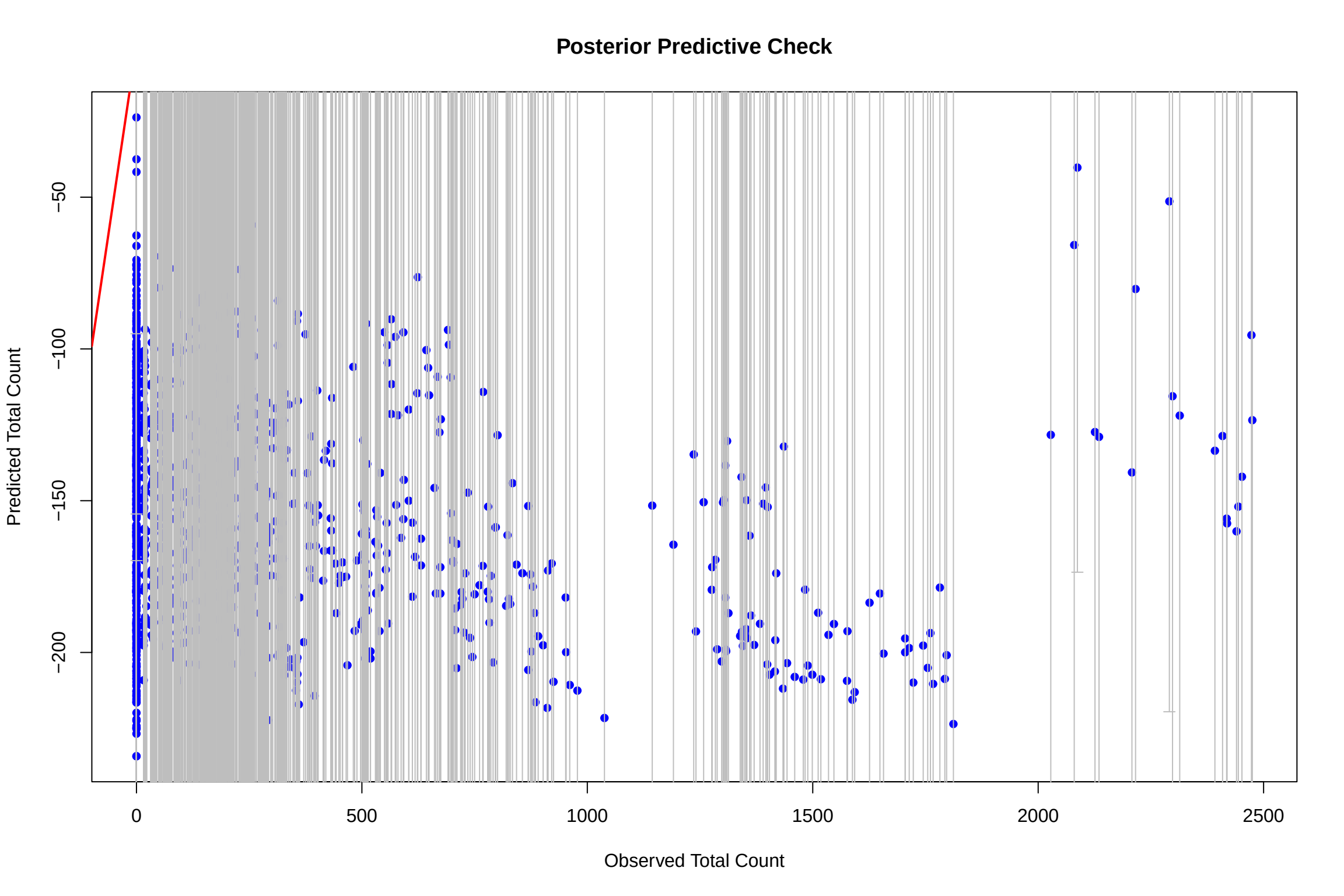

Density Plot of Unhealthy Days with AQI
Density
0.75
0.50
0.25
0.00
0
5
10
15
Unhealthy Days

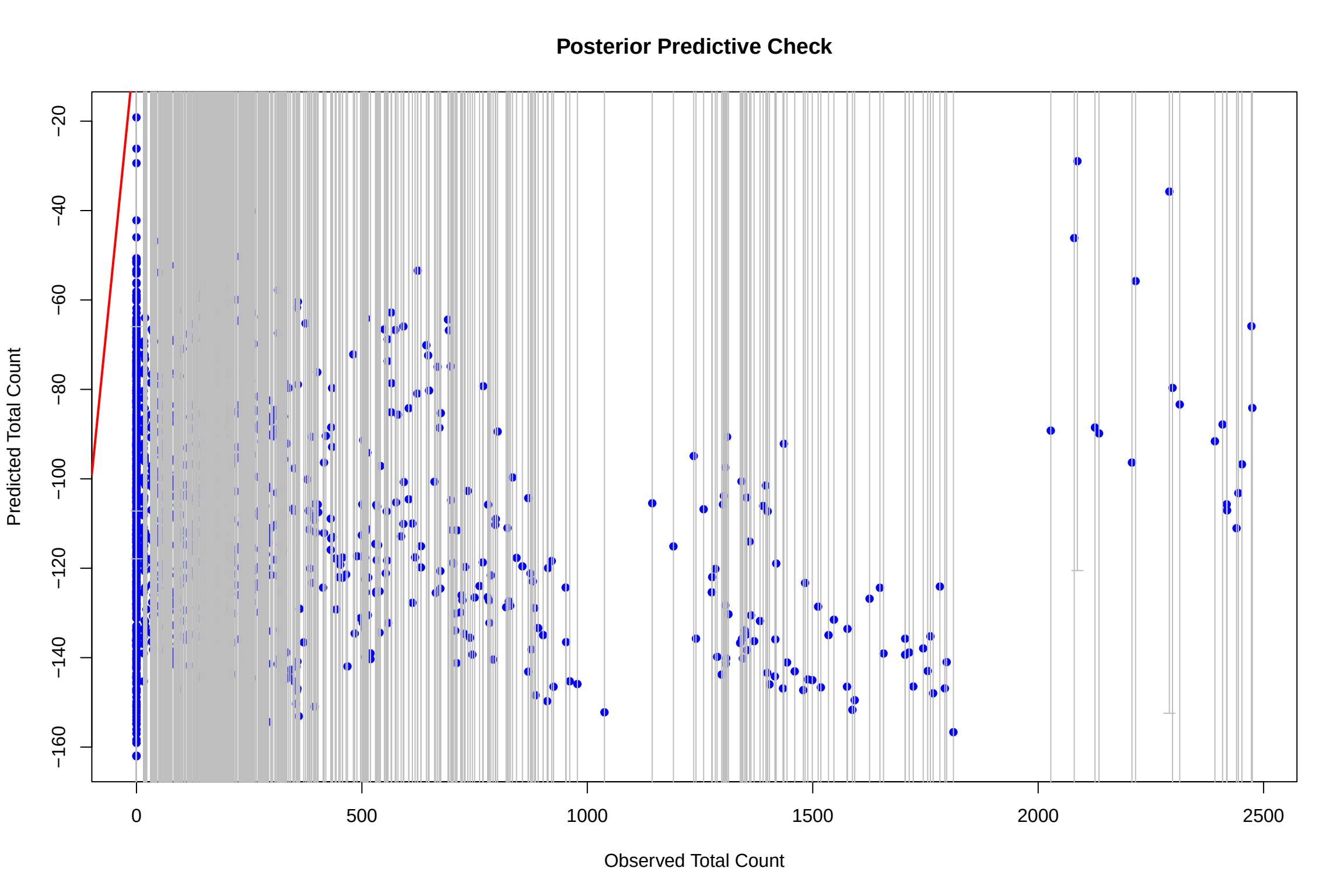

Average Max AQI Over Years by State
Average Max AQI
400
300
200
100
2000
2005
2010
2015
2020
Year
States
Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
Florida
Georgia
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Mississippi
Missouri
Montana
Nebraska
Nevada
New Hampshire
New Jersey
New Mexico
New York
North Carolina
North Dakota
Ohio
Oklahoma
Oregon
Pennsylvania
Rhode Island
South Carolina
South Dakota
Tennessee
Texas
Utah
Vermont
Virginia
Washington
West Virginia
Wisconsin
Wyoming

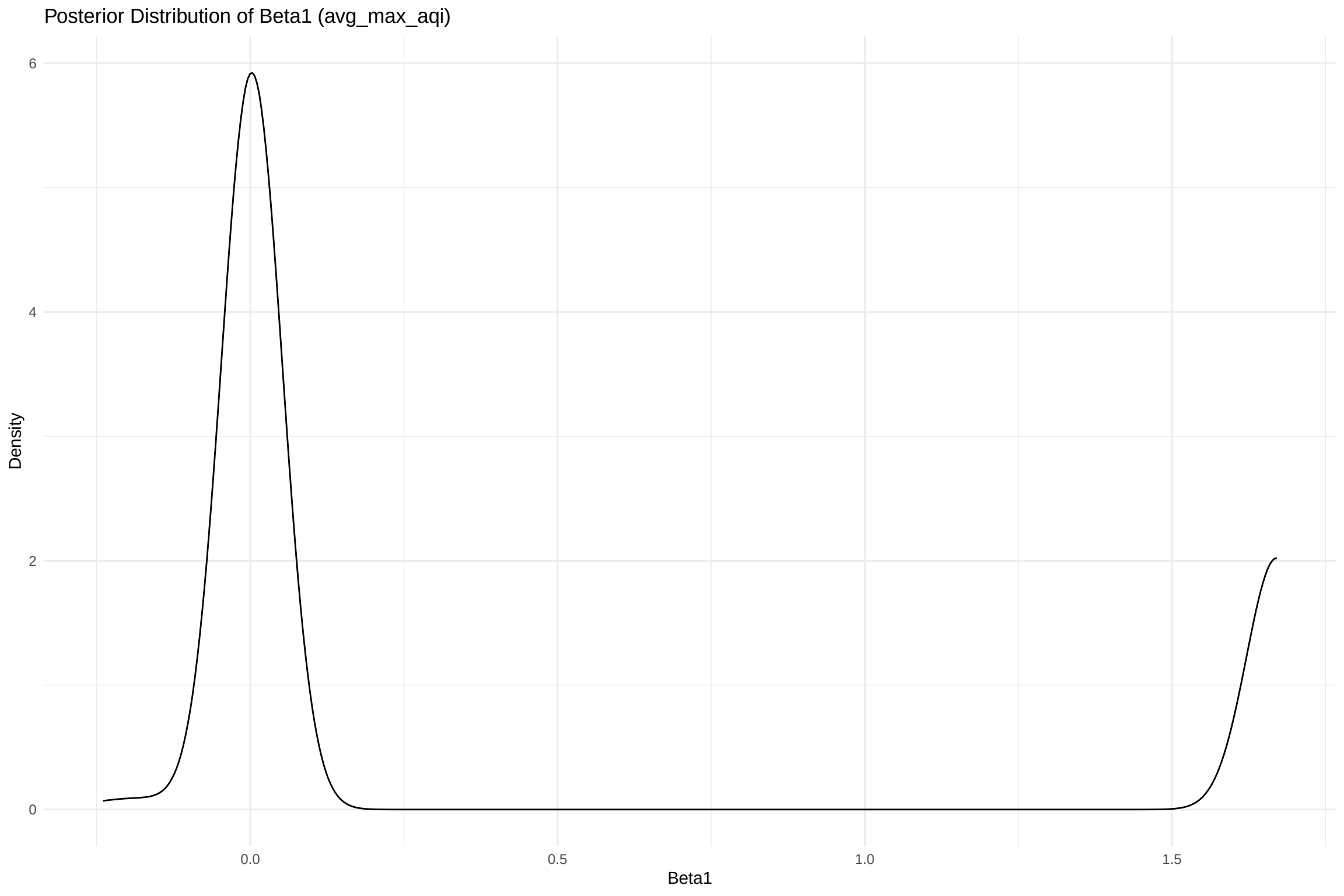

alpha
beta[1]
beta[2]
beta[3]
beta[4]
beta[5]
lambda
chain
1
2
3
4


Observed vs Predicted
Predicted
Observed
2
3
4
5
6
7
8
0
500
1000
1500
2000
2500


Posterior Predictive Distribution for Observation 1
Frequency
0
1000
2000
3000
4000
5000
6000
−300
−250
−200
−150
−100
−50
0
Prediction


Posterior Predictive Check
Predicted Total Count
−50
−100
−150
−200
0
500
1000
1500
2000
2500
Observed Total Count


Posterior Predictive Check
Predicted Total Count
−20
−40
−60
−80
−100
−120
−140
−160
0
500
1000
1500
2000
2500
Observed Total Count


Posterior Distribution of Beta1 (avg_max_aqi)
Density
6
4
2
0
0.0
0.5
1.0
1.5
Beta1