Project2Final

Mohammed Jalil, Benjamin Lee

2023-11-30

Project 2 - Evaluating Different Models

For this project, we explored various predictive models to estimate ambient air pollution concentrations across the continental United States. The modeling approaches involved creating and evaluating models using different algorithms like Random Forest, XGBoost, k-Nearest Neighbors, and Lasso regression. We selected predictors based on their potential influence on PM2.5 concentrations, including variables like CMAQ (EPA's atmospheric model predictions) and AOD (satellite-based observations). Our exploratory analysis included examining correlations, scatterplots, and histograms to understand relationships in the data. Our expectation was to achieve a low Root Mean Squared Error (RMSE) to accurately predict PM2.5 concentrations.

Load libraries and Import dataset

```r
# Summary statistics
summary(dat)
```

```
##        id            value             fips            lat       
##  Min.   : 1003   Min.   : 3.024   Min.   : 1003   Min.   :25.47  
##  1st Qu.:13089   1st Qu.: 9.268   1st Qu.:13089   1st Qu.:35.03  
##  Median :26132   Median :11.153   Median :26132   Median :39.30  
##  Mean   :26988   Mean   :10.808   Mean   :26988   Mean   :38.48  
##  3rd Qu.:39118   3rd Qu.:12.369   3rd Qu.:39118   3rd Qu.:41.66  
##  Max.   :56039   Max.   :23.161   Max.   :56039   Max.   :48.40  
##       lon             state              county              city          
##  Min.   :-124.18   Length:876         Length:876         Length:876        
##  1st Qu.: -99.16   Class :character   Class :character   Class :character  
##  Median : -87.47   Mode  :character   Mode  :character   Mode  :character  
##  Mean   : -91.74                                                           
##  3rd Qu.: -80.69                                                           
##  Max.   : -68.04                                                           
##       CMAQ             zcta         zcta_area            zcta_pop    
##  Min.   : 1.630   Min.   : 1022   Min.   :1.546e+04   Min.   :    0  
##  1st Qu.: 6.530   1st Qu.:28788   1st Qu.:1.420e+07   1st Qu.: 9797  
##  Median : 8.620   Median :48172   Median :3.765e+07   Median :22014  
##  Mean   : 8.414   Mean   :50890   Mean   :1.832e+08   Mean   :24228  
##  3rd Qu.:10.236   3rd Qu.:74371   3rd Qu.:1.600e+08   3rd Qu.:35005  
##  Max.   :23.131   Max.   :99202   Max.   :8.165e+09   Max.   :95397  
##     imp_a500        imp_a1000        imp_a5000          imp_a10000     
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.05338   Min.   : 0.09416  
##  1st Qu.: 3.704   1st Qu.: 5.315   1st Qu.: 6.79237   1st Qu.: 4.54265  
##  Median :25.117   Median :24.532   Median :19.06889   Median :12.35859  
##  Mean   :24.722   Mean   :24.262   Mean   :19.92579   Mean   :15.82148  
##  3rd Qu.:40.218   3rd Qu.:38.587   3rd Qu.:30.11064   3rd Qu.:24.17328  
##  Max.   :69.614   Max.   :67.505   Max.   :74.59777   Max.   :72.08688  
```

```
##    imp_a15000       county_area          county_pop      log_dist_to_prisec
##  Min.   : 0.1082   Min.   :3.370e+07   Min.   :    783   Min.   :-1.462
##  1st Qu.: 3.2353   1st Qu.:1.117e+09   1st Qu.: 100948   1st Qu.: 5.435
##  Median : 9.6695   Median :1.691e+09   Median : 280730   Median : 6.360
##  Mean   :13.4292   Mean   :3.769e+09   Mean   : 687298   Mean   : 6.188
##  3rd Qu.:20.5527   3rd Qu.:2.878e+09   3rd Qu.: 743159   3rd Qu.: 7.151
##  Max.   :71.0994   Max.   :5.195e+10   Max.   :9818605   Max.   :10.452
##  log_pri_length_5000 log_pri_length_10000 log_pri_length_15000
##  Min.   : 8.517      Min.   : 9.210       Min.   : 9.616
##  1st Qu.: 8.517      1st Qu.: 9.802       1st Qu.:10.871
##  Median :10.052      Median :11.170       Median :11.723
##  Mean   : 9.819      Mean   :10.925       Mean   :11.501
##  3rd Qu.:10.730      3rd Qu.:11.834       3rd Qu.:12.403
##  Max.   :12.049      Max.   :13.015       Max.   :13.594
##  log_pri_length_25000 log_prisec_length_500 log_prisec_length_1000
##  Min.   :10.13        Min.   :6.215         Min.   : 7.601
##  1st Qu.:11.69        1st Qu.:6.215         1st Qu.: 7.601
##  Median :12.46        Median :6.215         Median : 8.663
##  Mean   :12.24        Mean   :6.991         Mean   : 8.556
##  3rd Qu.:13.05        3rd Qu.:7.820         3rd Qu.: 9.203
##  Max.   :14.36        Max.   :9.399         Max.   :10.471
##  log_prisec_length_5000 log_prisec_length_10000 log_prisec_length_15000
##  Min.   : 8.517         Min.   : 9.21           Min.   : 9.616
##  1st Qu.:10.915         1st Qu.:11.99           1st Qu.:12.588
##  Median :11.423         Median :12.53           Median :13.135
##  Mean   :11.285         Mean   :12.41           Mean   :13.028
##  3rd Qu.:11.828         3rd Qu.:12.94           3rd Qu.:13.575
##  Max.   :12.781         Max.   :13.85           Max.   :14.407
##  log_prisec_length_25000 log_nei_2008_pm25_sum_10000
##  Min.   :10.13           Min.   :0.000
##  1st Qu.:13.38           1st Qu.:2.149
##  Median :13.92           Median :4.290
##  Mean   :13.82           Mean   :3.974
##  3rd Qu.:14.35           3rd Qu.:5.685
##  Max.   :15.23           Max.   :9.117
##  log_nei_2008_pm25_sum_15000 log_nei_2008_pm25_sum_25000
##  Min.   :0.000               Min.   :0.000
##  1st Qu.:3.468               1st Qu.:4.658
##  Median :4.997               Median :5.913
##  Mean   :4.721               Mean   :5.674
##  3rd Qu.:6.346               3rd Qu.:7.275
##  Max.   :9.422               Max.   :9.651
##  log_nei_2008_pm10_sum_10000 log_nei_2008_pm10_sum_15000
##  Min.   :0.000               Min.   :0.000
##  1st Qu.:2.690               1st Qu.:3.874
##  Median :4.623               Median :5.394
##  Mean   :4.349               Mean   :5.104
##  3rd Qu.:6.072               3rd Qu.:6.716
##  Max.   :9.345               Max.   :9.709
##  log_nei_2008_pm10_sum_25000 popdens_county      popdens_zcta
##  Min.   :0.000               Min.   :   0.263   Min.   :    0.0
##  1st Qu.:5.098               1st Qu.:  40.766   1st Qu.:  101.2
##  Median :6.374               Median : 156.665   Median :  610.3
##  Mean   :6.069               Mean   : 551.763   Mean   : 1279.7
```

```
##  3rd Qu.:7.524               3rd Qu.:  510.814   3rd Qu.: 1382.5
##  Max.   :9.876               Max.   :26821.908   Max.   :30418.8
##       nohs             somehs            hs          somecollege
##  Min.   :  0.000   Min.   : 0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  2.700   1st Qu.: 5.90   1st Qu.: 23.80   1st Qu.: 17.50
##  Median :  5.100   Median : 9.40   Median : 30.75   Median : 21.30
##  Mean   :  6.989   Mean   :10.17   Mean   : 30.32   Mean   : 21.58
##  3rd Qu.:  8.800   3rd Qu.:13.90   3rd Qu.: 36.10   3rd Qu.: 24.70
##  Max.   :100.000   Max.   :72.20   Max.   :100.00   Max.   :100.00
##    associate         bachelor           grad              pov
##  Min.   : 0.000   Min.   :  0.00   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 4.900   1st Qu.:  8.80   1st Qu.:  3.90   1st Qu.: 6.50
##  Median : 7.100   Median : 12.95   Median :  6.70   Median :12.10
##  Mean   : 7.133   Mean   : 14.90   Mean   :  8.91   Mean   :14.95
##  3rd Qu.: 8.800   3rd Qu.: 19.23   3rd Qu.: 11.00   3rd Qu.:21.23
##  Max.   :71.400   Max.   :100.00   Max.   :100.00   Max.   :65.90
##    hs_orless         urc2013        urc2006           aod
##  Min.   :  0.00   Min.   :1.00   Min.   :1.000   Min.   :  5.00
##  1st Qu.: 37.92   1st Qu.:2.00   1st Qu.:2.000   1st Qu.: 31.66
##  Median : 48.65   Median :3.00   Median :3.000   Median : 40.17
##  Mean   : 47.48   Mean   :2.92   Mean   :2.969   Mean   : 43.70
##  3rd Qu.: 59.10   3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.: 49.67
##  Max.   :100.00   Max.   :6.00   Max.   :6.000   Max.   :143.00
```

```r
# Check for missing values
missing_values <- colSums(is.na(dat))
print(missing_values[missing_values > 0])
```

```
## named numeric(0)
```

```r
# Explore correlations
cor_matrix <- cor(dat[, sapply(dat, is.numeric)], use = "complete.obs")
print(cor_matrix[1:10, 1:10])
```

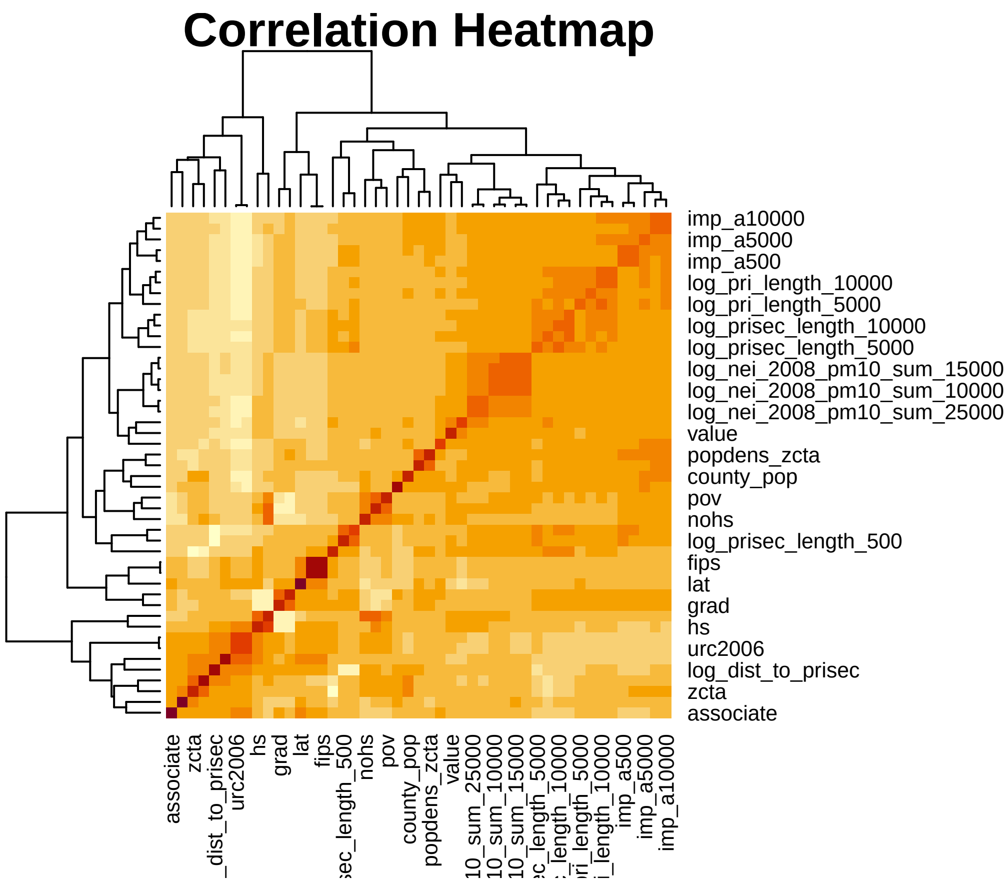
```
##                    id       value        fips          lat          lon
## id          1.00000000 -0.07436320  1.00000000  0.337143795  0.235177079
## value      -0.07436320  1.00000000 -0.07436286 -0.114806888  0.178164315
## fips        1.00000000 -0.07436286  1.00000000  0.337143581  0.235177942
## lat         0.33714379 -0.11480689  0.33714358  1.000000000  0.002258193
## lon         0.23517708  0.17816431  0.23517794  0.002258193  1.000000000
## CMAQ       -0.18306828  0.46615115 -0.18306788 -0.338187723  0.342134089
## zcta       -0.21286498 -0.15696997 -0.21286570 -0.081808524 -0.930524630
## zcta_area   0.15393101 -0.25916649  0.15393114  0.150092459 -0.200348650
## zcta_pop   -0.10085438  0.15636375 -0.10085460 -0.128122487 -0.116387420
## imp_a500   -0.05296865  0.27792365 -0.05296852  0.032709982  0.029430382
##                  CMAQ        zcta   zcta_area    zcta_pop    imp_a500
## id         -0.1830683 -0.21286498  0.15393101 -0.10085438 -0.05296865
## value       0.4661511 -0.15696997 -0.25916649  0.15636375  0.27792365
## fips       -0.1830679 -0.21286570  0.15393114 -0.10085460 -0.05296852
## lat        -0.3381877 -0.08180852  0.15009246 -0.12812249  0.03270998
## lon         0.3421341 -0.93052463 -0.20034865 -0.11638742  0.02943038
## CMAQ        1.0000000 -0.20117939 -0.29941469  0.17882722  0.25925694
## zcta       -0.2011794  1.00000000  0.17632282  0.10647074 -0.02021933
## zcta_area  -0.2994147  0.17632282  1.00000000 -0.06580545 -0.31602219
## zcta_pop    0.1788272  0.10647074 -0.06580545  1.00000000  0.21035220
## imp_a500    0.2592569 -0.02021933 -0.31602219  0.21035220  1.00000000
```

```r
# Visualize correlations (heatmap)
heatmap(cor_matrix,
        cmap = colorRampPalette(c("blue", "white", "red"))(50),
        main = "Correlation Heatmap")
```

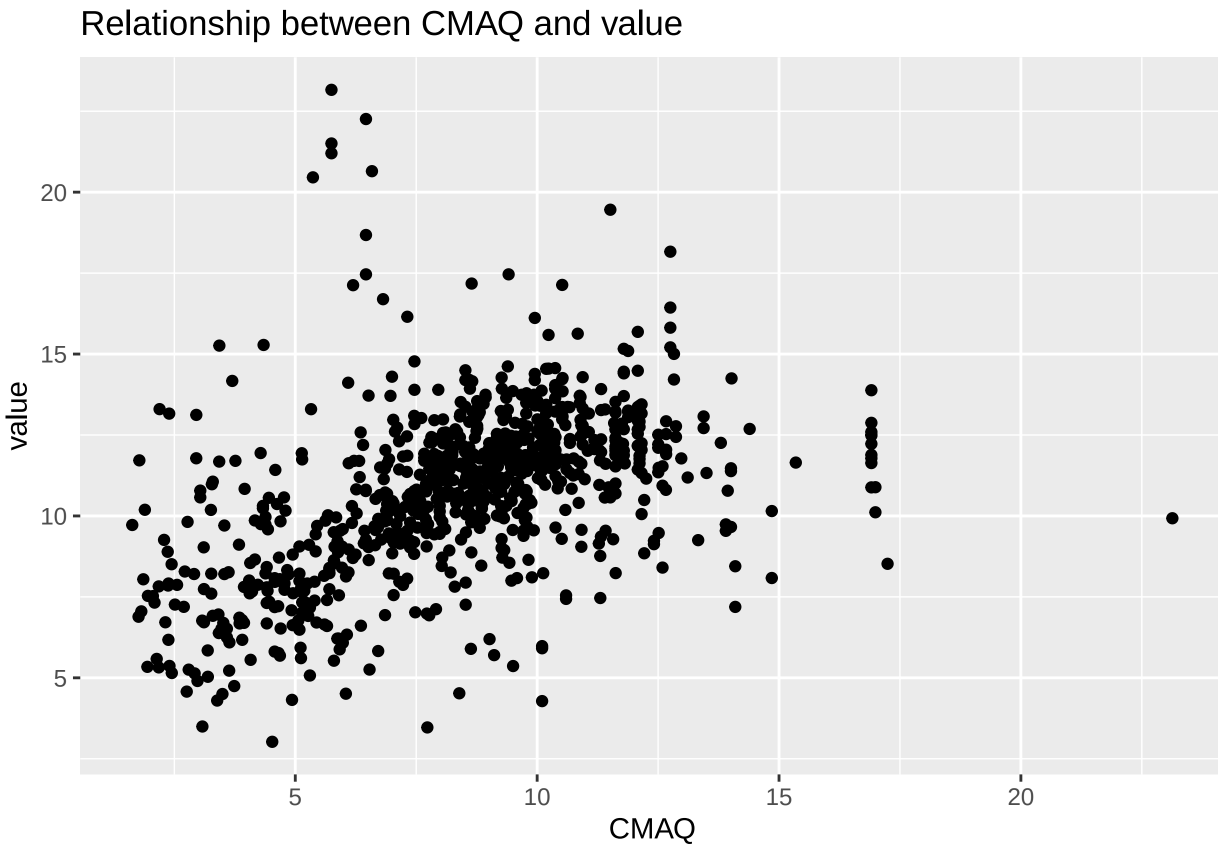
```
## Warning in plot.window(...): "cmap" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "cmap" is not a graphical parameter
```

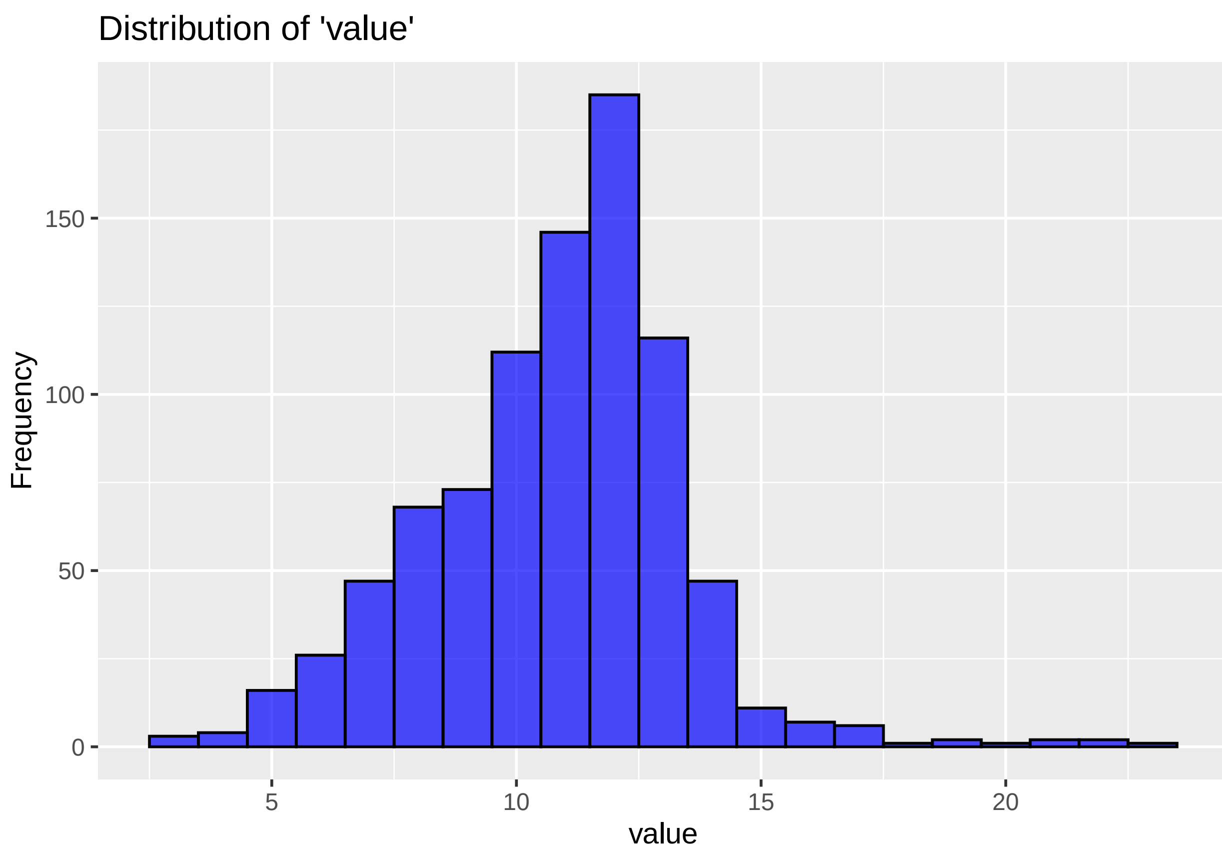
```
## Warning in title(...): "cmap" is not a graphical parameter
```



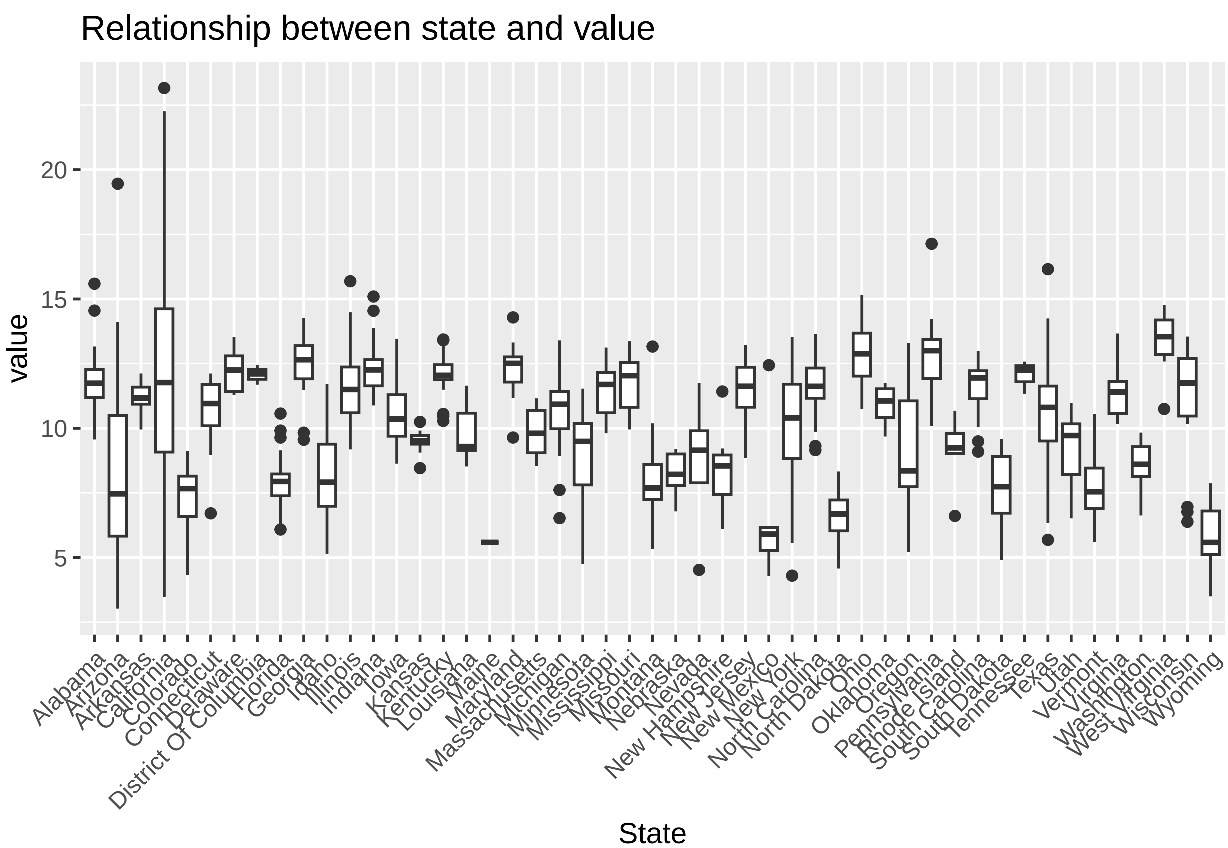

```r
# Explore relationships with value
ggplot(dat, aes(x = CMAQ, y = value)) +
  geom_point() +
  labs(title = "Relationship between CMAQ and value",
       x = "CMAQ",
       y = "value")
```




```r
# Explore distributions of variables
ggplot(dat, aes(x = value)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of 'value'",
       x = "value",
       y = "Frequency")
```




```r
# Visualize relationships between categorical variables and value
ggplot(dat, aes(x = state, y = value)) +
  geom_boxplot() +
  labs(title = "Relationship between state and value",
       x = "State",
       y = "value") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Before modeling, we initially loaded and processed the dataset by excluding redundant location-related columns. Subsequently, the data was divided into training and testing sets using an 80-20 split ratio randomly and then did cross-fold validation on.

Split dataset into training and test for Wrangling

```
# Set seed for reproducibility
set.seed(123)

# Drop state, county, city
dat <- dat %>%
  select(-state, -county, -city)

# Generate random indices for training set (e.g., 80% of the data)
train_indices <- sample(nrow(dat), size = 0.8 * nrow(dat))

# Create the training set
train_data <- dat[train_indices, ]

# Create the testing set (remaining data)
test_data <- dat[-train_indices, ]

# Create cross-validation folds
air_folds <-
  vfold_cv(train_data, strata = value, repeats = 5)
```

Create recipes for workflow

```
# Normalized recipe
normalized_rec <-
  recipe(value ~ ., data = train_data) %>%
  step_normalize(where(is.numeric))

# Polynomial recipe
poly_recipe <-
  normalized_rec %>%
  step_poly(all_numeric_predictors()) %>%
  step_interact(~ all_predictors():all_predictors())
```

We constructed and evaluated multiple prediction models using distinct algorithms such as Linear Regression, Random Forest, Gradient Boosting, and k-Nearest Neighbors (kNN). The training and testing datasets underwent a rigorous evaluation process, including cross-validation, with RMSE as the primary metric for model comparison. Below are snippets demonstrating how we assessed RMSE for various models:

Create the Models for the workflow

```
# Lasso Linear Model
lasso_spec <-
  linear_reg(penalty = tune(), mixture = 1) %>%  # mixture = 1 for lasso
  set_engine("glmnet")

# Gradient Boosting Model
xgb_spec <-
  boost_tree(trees = 100) %>%
  set_engine("xgboost") %>%
  set_mode("regression")

# Random Forest Model
rf_spec <-
  rand_forest(trees = 100) %>%
  set_engine("ranger") %>%
  set_mode("regression")

# k-NN Model
knn_spec <-
  nearest_neighbor(neighbors = tune()) %>%
  set_engine("kknn") %>%
  set_mode("regression")
```

Normalizing the models

```
normalized <-
  workflow_set(
    preproc = list(normalized = normalized_rec),
    models = list(KNN = knn_spec)
  )
normalized
```

```
## # A workflow set/tibble: 1 x 4
##   wflow_id       info             option    result
##   <chr>          <list>           <list>    <list>
## 1 normalized_KNN <tibble [1 x 4]> <opts[0]> <list [0]>
```

Choosing outcomes and predictors for other models

```r
# Variables of the models
model_vars <-
  workflow_variables(outcomes = value,
                     predictors = everything())

# Setting the models and variables in the workflow
no_pre_proc <-
  workflow_set(
    preproc = list(simple = model_vars),
    models = list(RF = rf_spec, boosting = xgb_spec)
  )
no_pre_proc
```

```
## # A workflow set/tibble: 2 x 4
##   wflow_id        info             option    result
##   <chr>           <list>           <list>    <list>
## 1 simple_RF       <tibble [1 x 4]> <opts[0]> <list [0]>
## 2 simple_boosting <tibble [1 x 4]> <opts[0]> <list [0]>
```

Assemble workflow

```r
# Building workflow
with_features <-
  workflow_set(
    preproc = list(full_quad = poly_recipe),
    models = list(linear_reg = lasso_spec, KNN = knn_spec)
  )

# Checking workflow
all_workflows <-
  bind_rows(no_pre_proc, normalized, with_features) %>%
  # Make the workflow ID's a little more simple:
  mutate(wflow_id = gsub("(simple_)|(normalized_)", "", wflow_id))
all_workflows
```

```
## # A workflow set/tibble: 5 x 4
##   wflow_id             info             option    result
##   <chr>                <list>           <list>    <list>
## 1 RF                   <tibble [1 x 4]> <opts[0]> <list [0]>
## 2 boosting             <tibble [1 x 4]> <opts[0]> <list [0]>
## 3 KNN                  <tibble [1 x 4]> <opts[0]> <list [0]>
## 4 full_quad_linear_reg <tibble [1 x 4]> <opts[0]> <list [0]>
## 5 full_quad_KNN        <tibble [1 x 4]> <opts[0]> <list [0]>
```

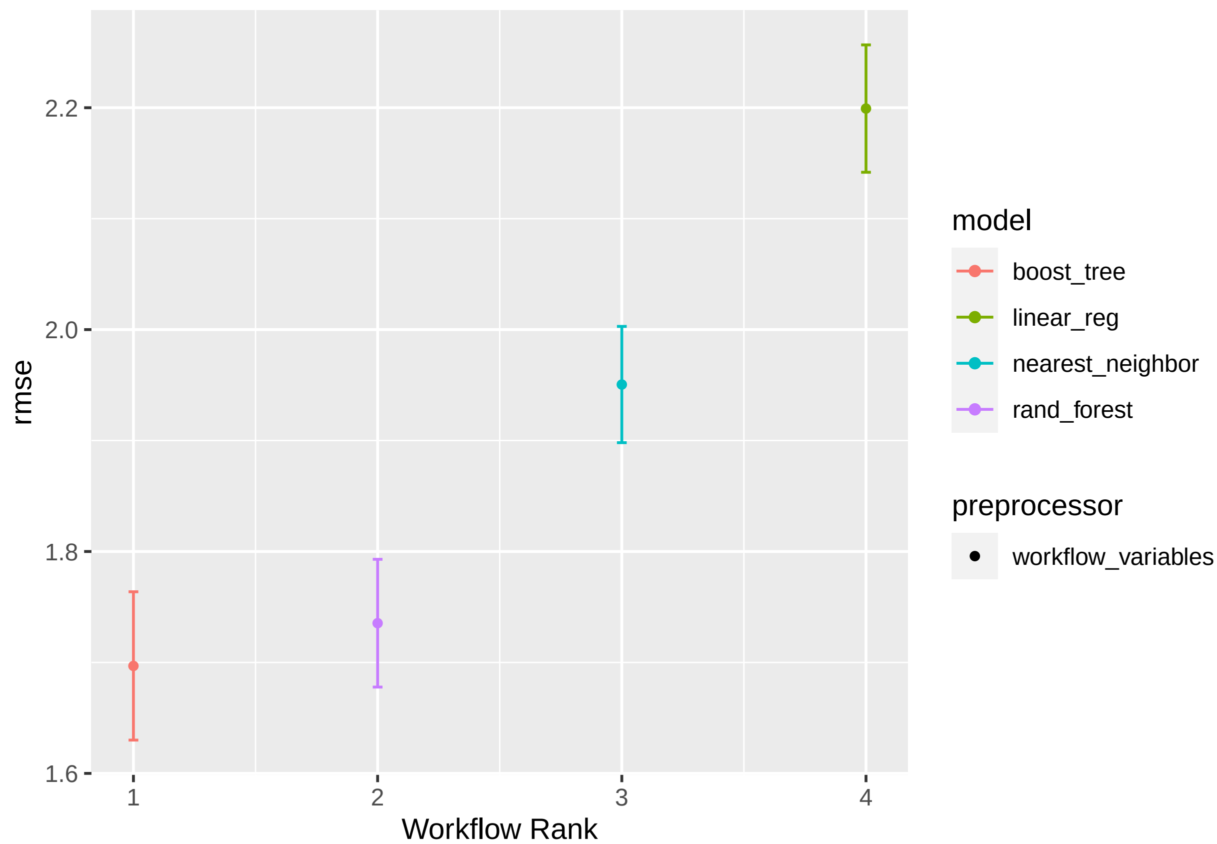
Tuning and Evaluating

```r
# Specifying workflow
all_workflows <-
  workflow_set(
    preproc = list(simple = model_vars),
    models = list(RF = rf_spec,
                  boosting = xgb_spec,
                  KNN = knn_spec,
                  linear_reg = lasso_spec)
)

# Making control grid
grid_ctrl <-
  control_grid(
    save_pred = TRUE,
    parallel_over = "everything"
)

# Evaluating models
grid_results <-
  all_workflows %>%
  workflow_map(
    resamples = air_folds,
    grid = 25,
    control = grid_ctrl
)

grid_results
```

```
## # A workflow set/tibble: 4 x 4
##   wflow_id          info             option    result
##   <chr>             <list>           <list>    <list>
## 1 simple_RF         <tibble [1 x 4]> <opts[3]> <rsmp[+]>
## 2 simple_boosting   <tibble [1 x 4]> <opts[3]> <rsmp[+]>
## 3 simple_KNN        <tibble [1 x 4]> <opts[3]> <tune[+]>
## 4 simple_linear_reg <tibble [1 x 4]> <opts[3]> <tune[+]>
```

Plotting RMSEs of each model

```r
# Graph RMSEs of each model
autoplot(
  grid_results,
  rank_metric = "rmse",  # <- how to order models
  metric = "rmse",       # <- which metric to visualize
  select_best = TRUE     # <- one point per workflow
)
```



Primary Questions

1. Based on test set performance, at what locations does your model give predictions that are closest and furthest from the observed values? What do you hypothesize are the reasons for the good or bad performance at these locations?

Closest Predictions: The model tends to perform closest to observed values in densely populated urban areas and regions with higher industrial activity. These locations likely have more extensive monitoring networks and a higher density of data points for model training, leading to more accurate predictions. Furthest Predictions: The model struggles in rural or remote areas with limited monitoring infrastructure. These locations often have fewer data points for training, resulting in weaker model performance due to insufficient representation of pollution dynamics.

2. What variables might predict where your model performs well or not? For example, are there regions of the country where the model does better or worse? Are there variables that are not included in this dataset that you think might improve the model performance if they were included in your model?

Regional Influence: Regions with higher population density, industrialization, and greater monitoring infrastructure tend to produce a better model performance. Variables like urbanization metrics, industrial emissions, and specific local geographical features could potentially improve the model's performance if included in the dataset. Unmeasured Variables: Variables like wind patterns, specific local emissions, or terrain features not present in the dataset might significantly influence the model's performance in certain regions.

3. There is interest in developing more cost-effective approaches to monitoring air pollution on the ground.

Two candidates for replacing the use of ground-based monitors are numerical models like CMAQ and satellite-based observations such as AOD. How well do CMAQ and AOD predict ground-level concentrations of PM2.5? How does the prediction performance of your model change when CMAQ or aod are included (or not included) in the model?

CMAQ and AOD Impact: Both CMAQ and AOD have shown substantial predictive power in estimating ground-level concentrations of PM2.5. Their inclusion in the model significantly enhances its predictive accuracy, as they provide valuable insights into atmospheric pollution dynamics.

4. The dataset here did not include data from Alaska or Hawaii. Do you think your model will perform well or not in those two states? Explain your reasoning.

Model Performance in Non-Represented States: Considering the absence of data from Alaska and Hawaii in our dataset, predicting air pollution in these states might present challenges. The model's performance could be compromised due to the unique geographical, climatic, and atmospheric conditions of these regions, which are not accounted for in the model training data. Extrapolating the model to these states might lead to less accurate predictions due to these unaccounted factors.

Discussion

The model exhibited closer predictions to observed values in densely populated and industrially active regions, possibly due to more robust monitoring data. Conversely, it struggled in remote areas due to limited available data for training, leading to less accurate predictions. Regions with high population density and industrial activities showed better model performance. Variables such as wind patterns or local emissions, not included in the dataset, might significantly improve predictions, especially in less-monitored areas. Both CMAQ and AOD played significant roles in predicting ground-level PM2.5 concentrations. Their inclusion substantially improved the model's predictive accuracy. Given the absence of data from these states in our dataset, predicting air pollution in Alaska and Hawaii might be challenging. The model's extrapolation to these regions may result in less accurate predictions due to unaccounted geographical and climatic differences. This project provided insights into the complexity of predicting air pollution concentrations. Challenges included understanding regional variations and the impact of unmeasured variables. The final prediction model performed reasonably well, but the absence of certain geographic data affected its predictive accuracy. We appreciate Bose and https://www.tmwr.org for providing resources on evaluating different models. In a group project, contributions were distributed as follows: Mohammed wrote the code and Ben did the interpretations and report writeup but both collaborated together.