ENSEMBLES OF ARIMA AND ARIMAX MODELS FOR FLU FORECASTING

Victor Felix

2024-06-11

R Markdown

```r
library("tidyr")
library("MMWRweek")
library("data.table")
library("caret")
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
library("purrr")
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:caret':
##
##     lift
```

```
## The following object is masked from 'package:data.table':
##
##     transpose
```

```r
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library("tseries")
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library("gtools")
library("forecast")
library("scoringutils")
```

```
## Note: scoringutils is currently undergoing major development changes (with an update planned for the
```

```r
library("covidHubUtils")
library("parallel")
library("future")#https://cran.r-project.org/web/packages/future/vignettes/future-4-issues.html
```

```
##
## Attaching package: 'future'
```

```
## The following object is masked from 'package:tseries':
##
##     value
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```r
library("listenv")
```

```
##
## Attaching package: 'listenv'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```r
library("epitools")
library("ggplot2")
library("sf")
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

in the document. You can embed an R code chunk like this:

```r
load("ARIMA_MODELS.Rdata")
load("ADJACENT_MODELS_LAG1.Rdata")
load("TEMPERATURE_MODELS_LAG1.Rdata")
load("EPIWEEK_MODELS_LAG1.Rdata")
```

SAVE AS FUNCTIONS LATER

```r
# Define the function
calculate_mean_abs_error <- function(state_list) {
  # Get the number of states in the list
  num_states <- length(state_list)

  # Initialize vectors to store state names and mean values
  state_names <- vector("character", length = num_states)
  mean_values <- vector("numeric", length = num_states)

  # Iterate through the states
  for (state in 1:num_states) {
    # Get the name of the state
    state_name <- names(state_list)[state]

    # Get the mean WIS value for the state
    mean_abs_error <- mean(state_list[[state]]$abs_error)

    # Store the state name and mean value in the vectors
    state_names[state] <- state_name
    mean_values[state] <- mean_abs_error
  }

  # Create a data frame
  results_df <- data.frame(State = state_names, Mean_abs_error = mean_values)

  # Return the data frame
  return(results_df)
}
#calculate_mean_abs_error
# Define the function
calculate_mean_wis <- function(state_list) {
  # Get the number of states in the list
  num_states <- length(state_list)

  # Initialize vectors to store state names and mean values
  state_names <- vector("character", length = num_states)
  mean_values <- vector("numeric", length = num_states)

  # Iterate through the states
  for (state in 1:num_states) {
    # Get the name of the state
    state_name <- names(state_list)[state]

    # Get the mean WIS value for the state
    mean_wis <- mean(state_list[[state]]$WIS)
    # Store the state name and mean value in the vectors
    state_names[state] <- state_name
    mean_values[state] <- mean_wis
  }

  # Create a data frame
  results_df <- data.frame(State = state_names, Mean_WIS = mean_values)

  # Return the data frame
  return(results_df)
}

# Define the function
calculate_mean_N_of_models <- function(state_list) {
  # Get the number of states in the list
  num_states <- length(state_list)

  # Initialize vectors to store state names and mean values
  state_names <- vector("character", length = num_states)
  mean_values <- vector("numeric", length = num_states)

  # Iterate through the states
  for (state in 1:num_states) {
    # Get the name of the state
    state_name <- names(state_list)[state]

    # Get the mean WIS value for the state
    mean_Number_of_models <- mean(state_list[[state]]$Number_of_models)

    # Store the state name and mean value in the vectors
    state_names[state] <- state_name
    mean_values[state] <- mean_Number_of_models
  }

  # Create a data frame
  results_df <- data.frame(State = state_names, Mean_Number_of_models = mean_values)

  # Return the data frame
  return(results_df)
}
```

COMPARING WIS RESULTS FOR WEEK1

```
W1<-NULL
calculate_mean_N_of_models
```

```
## function(state_list) {
##   # Get the number of states in the list
##   num_states <- length(state_list)
##
##   # Initialize vectors to store state names and mean values
##   state_names <- vector("character", length = num_states)
##   mean_values <- vector("numeric", length = num_states)
##
##   # Iterate through the states
##   for (state in 1:num_states) {
##     # Get the name of the state
##     state_name <- names(state_list)[state]
##
##     # Get the mean WIS value for the state
##     mean_Number_of_models <- mean(state_list[[state]]$Number_of_models)
##
##     # Store the state name and mean value in the vectors
##     state_names[state] <- state_name
##     mean_values[state] <- mean_Number_of_models
##   }
##
##   # Create a data frame
##   results_df <- data.frame(State = state_names, Mean_Number_of_models = mean_values)
##
##   # Return the data frame
##   return(results_df)
## }
```

```
# AUTO ARIMA RESULTS
AUTO_AR_W1 <- calculate_mean_wis(AUTO_ARIMA_WEEK1_list)
ES27_AR_W1 <- calculate_mean_wis(ES27_ARIMA_WEEK1_list)
ES64_AR_W1 <- calculate_mean_wis(ES64_ARIMA_WEEK1_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W1 <- calculate_mean_wis(AUTO_ADJACENT_WEEK1_list)
ES27_ADJ_W1 <- calculate_mean_wis(ES27_ADJACENT_WEEK1_list)
ES64_ADJ_W1 <- calculate_mean_wis(ES64_ADJACENT_WEEK1_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W1 <- calculate_mean_wis(AUTO_TEMPERATURE_WEEK1_list)
ES27_TMP_W1 <- calculate_mean_wis(ES27_TEMPERATURE_WEEK1_list)
ES64_TMP_W1 <- calculate_mean_wis(ES64_TEMPERATURE_WEEK1_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W1 <- calculate_mean_wis(AUTO_EPIWEEK_WEEK1_list)
ES27_EPI_W1 <- calculate_mean_wis(ES27_EPIWEEK_WEEK1_list)
ES64_EPI_W1 <- calculate_mean_wis(ES64_EPIWEEK_WEEK1_list)

# MERGE
# AUTO ARIMA RESULTS
W1 <- merge(AUTO_AR_W1, ES27_AR_W1, by = "State")
W1 <- merge(W1, ES64_AR_W1, by = "State")
# ADJACENT ARIMAX RESULTS
W1 <- merge(W1, AUTO_ADJ_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, AUTO_ADJ_W1, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

```
W1 <- merge(W1, ES27_ADJ_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES27_ADJ_W1, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

```r
W1 <- merge(W1, ES64_ADJ_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES64_ADJ_W1, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the
## result
```

```r
# TEMPERATURE ARIMAX RESULTS
W1 <- merge(W1, AUTO_TMP_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, AUTO_TMP_W1, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the
## result
```

```r
W1 <- merge(W1, ES27_TMP_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES27_TMP_W1, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’ are duplicated in the result
```

```r
W1 <- merge(W1, ES64_TMP_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES64_TMP_W1, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’ are duplicated in the result
```

```r
# EPIWEEK ARIMAX RESULTS
W1 <- merge(W1, AUTO_EPI_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, AUTO_EPI_W1, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

```r
W1 <- merge(W1, ES27_EPI_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES27_EPI_W1, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

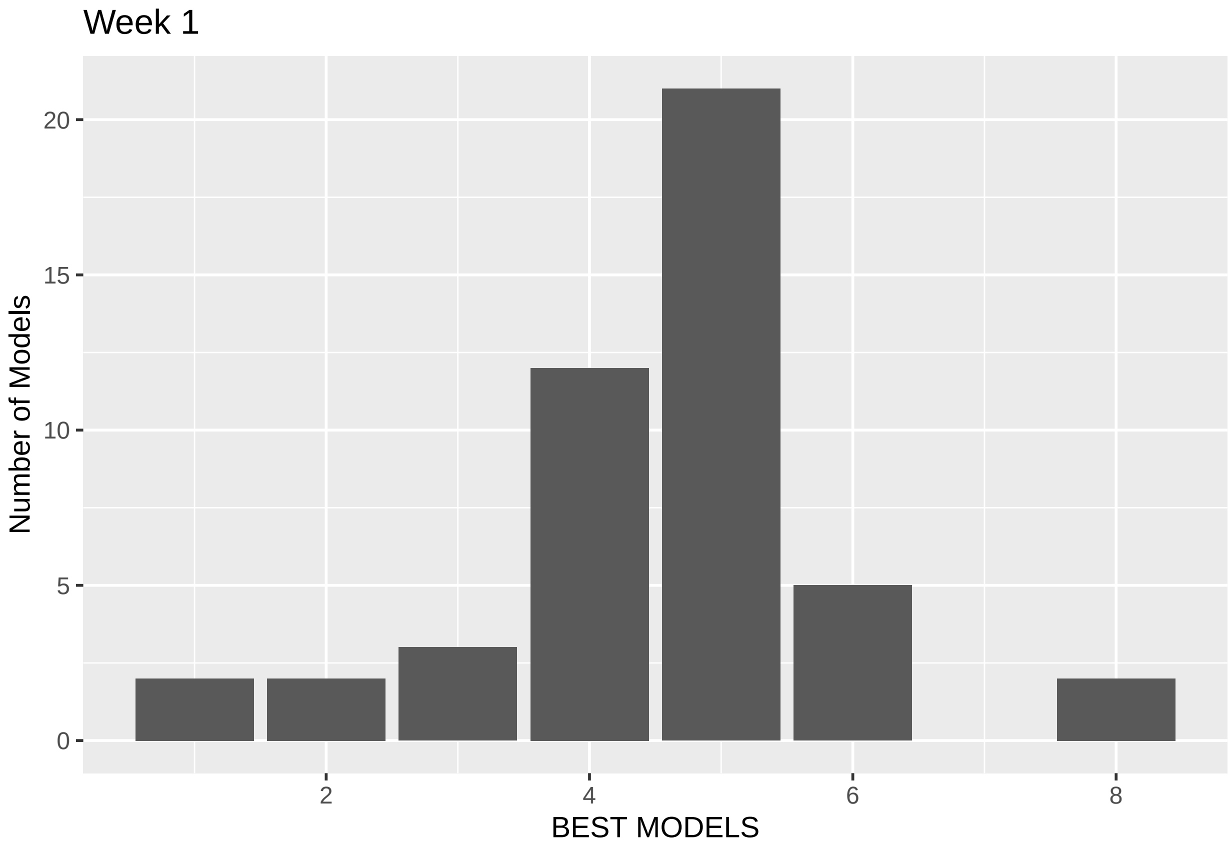
```r
W1 <- merge(W1, ES64_EPI_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES64_EPI_W1, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are
## duplicated in the result
```

```r
# Rename columns for clarity
colnames(W1)[1] <- "NAME"
colnames(W1)[2] <- "AUTO_AR"
colnames(W1)[3] <- "ES27_AR"
colnames(W1)[4] <- "ES64_AR"
colnames(W1)[5] <- "AUTO_ADJ"
colnames(W1)[6] <- "ES27_ADJ"
colnames(W1)[7] <- "ES64_ADJ"
colnames(W1)[8] <- "AUTO_TMP"
colnames(W1)[9] <- "ES27_TMP"
colnames(W1)[10] <- "ES64_TMP"
colnames(W1)[11] <- "AUTO_EPI"
colnames(W1)[12] <- "ES27_EPI"
colnames(W1)[13] <- "ES64_EPI"

# Identify the best result for each state
W1$Best_Result <- apply(W1[,2:13], 1, function(x) {
  which.min(x)
})

#W1$Best_Result <- apply(W1[, 2:13], 1, function(x) {
#  colnames(W1)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W1)
```

```
##              NAME    AUTO_AR    ES27_AR    ES64_AR   AUTO_ADJ   ES27_ADJ
## 1         Alabama 103.409318 104.093942 105.012400 101.546152 100.225158
## 2         Arizona  46.302257  46.476840  47.582193  45.440483  44.501971
## 3        Arkansas  24.370957  23.775337  23.892389  25.099721  24.922003
## 4      California 156.499353 155.686764 154.547088 155.732654 152.157004
## 5        Colorado  69.933717  69.357734  69.366517  66.800570  66.584179
## 6     Connecticut  51.271575  50.954436  52.112114  50.306399  51.084597
## 7        Delaware   8.305231   8.526467   8.625190   7.648442   7.619765
## 8         Georgia 161.960674 159.050378 159.000997 163.572080 160.554938
## 9           Idaho  14.337864  14.136988  14.154030  13.171992  13.002660
## 10       Illinois  94.221206  89.584781  90.434714  91.111630  89.821944
## 11        Indiana  42.445789  42.077190  42.215812  40.715814  40.997438
## 12           Iowa  11.822358  12.270603  12.425317  10.883856  11.414388
## 13         Kansas  29.315392  29.629395  29.356502  25.652730  26.341144
## 14       Kentucky  50.354427  50.767179  50.454917  52.106745  51.372688
## 15      Louisiana 129.221764 125.113062 127.365459 120.222435 119.505556
## 16          Maine  18.498170  17.836077  18.020419  18.212056  17.539372
## 17       Maryland  59.747855  68.908037  64.901110  59.911052  59.201471
## 18  Massachusetts  69.871965  66.557022  67.756392  65.006511  64.348748
## 19       Michigan  62.840660  62.926081  63.095649  60.871943  61.112588
## 20      Minnesota  28.314119  28.375812  28.568497  28.466913  28.712551
## 21    Mississippi  56.995763  56.257374  56.708294  54.486092  54.093453
## 22       Missouri  32.112775  32.280581  32.134688  31.976379  31.367902
## 23        Montana   8.092742   8.101631   8.136461   7.438520   7.451152
## 24       Nebraska  27.355127  27.028959  27.198969  25.312535  25.416448
## 25         Nevada  30.635113  31.017026  32.720279  30.939323  30.802592
## 26  New Hampshire   5.505017   5.594131   5.540229   5.177490   5.078553
## 27     New Jersey  59.965398  58.715084  58.501178  57.531297  57.485736
## 28     New Mexico  53.639531  53.621172  54.192405  51.069190  50.819572
## 29       New York  87.904225  88.543041  88.942887  83.636195  83.669515
## 30 North Carolina 127.668995 130.636261 127.848535 130.899734 129.639571
## 31   North Dakota   9.685183   9.896307  10.045193   9.574397   9.386909
## 32           Ohio  44.825429  44.892814  44.813525  43.374196  42.349803
## 33       Oklahoma  42.057668  41.671354  41.180721  41.400282  41.270291
## 34         Oregon  49.089592  49.860639  49.301600  46.183218  47.493492
## 35   Pennsylvania  71.227208  69.133252  72.146752  60.964980  60.822074
## 36   Rhode Island  15.141265  15.493542  15.512260  14.143232  14.501541
## 37 South Carolina  60.836149  59.155967  58.931107  56.712461  56.186468
## 38   South Dakota  16.042250  15.811570  15.644600  15.952404  15.858926
## 39      Tennessee  78.816135  73.230730  72.274990  72.857155  72.488680
## 40          Texas 153.139013 152.475883 149.334594 154.496798 152.689885
## 41           Utah  69.420469  68.995596  69.749752  63.052363  63.563518
## 42        Vermont   8.225015   8.178253   8.170470   7.461085   7.525942
## 43       Virginia 343.054028 342.779872 344.594059 306.675458 300.953800
## 44     Washington  39.398563  38.804666  38.503057  40.454282  40.162748
## 45  West Virginia  32.432192  32.444482  32.553516  33.545429  32.667212
## 46      Wisconsin  25.657324  25.529887  25.615115  25.127876  24.849991
## 47        Wyoming  11.362245  11.220385  11.361650  10.703540  10.754306
##      ES64_ADJ   AUTO_TMP   ES27_TMP   ES64_TMP   AUTO_EPI   ES27_EPI   ES64_EPI
## 1  109.006360 104.326739 105.677502 107.363015 109.151577 108.021187 107.249861
## 2   45.923142  46.792122  46.390313  46.862976  48.553419  47.425288  47.817300
## 3   24.849880  25.058075  24.276970  24.641728  25.415873  25.326815  25.047053
## 4  159.086508 158.704973 157.596654 156.441054 163.926472 160.315585 161.323143
## 5   71.395540  70.281718  70.238685  70.344102  72.153806  71.433583  71.595681
## 6   54.482050  52.188673  53.306279  55.106539  53.684705  52.956535  54.840143
## 7    7.718464   8.660477   8.463488   8.524026   8.846841   8.949536   8.938099
## 8  176.854467 164.464203 160.919355 160.508273 168.237783 165.118086 164.634149
## 9   14.237797  14.723314  14.324970  14.349149  14.426165  14.380011  14.375228
## 10  94.544026  93.834818  91.515116  91.478563 100.077096  96.804425  96.224484
## 11  42.878662  43.417411  42.141149  42.448102  44.786394  43.563741  43.771000
## 12  11.132076  12.647863  12.590814  12.711989  12.606368  12.325643  12.510155
## 13  26.179869  29.228394  29.917587  29.353212  31.086116  30.481501  30.710024
## 14  50.128377  50.591266  51.783602  51.503146  54.916712  53.940807  53.772426
## 15 138.023047 128.605435 126.968220 126.934019 137.247989 135.417776 136.203676
## 16  18.075635  18.467677  18.136546  18.332507  19.135026  18.479281  18.498002
## 17  61.335633  60.319243  65.120285  62.773668  60.942424  63.523074  62.374800
## 18  74.351341  68.948523  68.138229  69.101131  73.783211  71.203719  71.781918
## 19  64.572716  65.523861  64.397359  65.048600  68.684231  66.688676  67.140660
## 20  28.547059  29.020251  28.930288  29.130452  29.294419  28.999441  29.176323
## 21  59.191239  57.346384  57.255806  57.325514  61.353775  61.334410  61.195853
## 22  32.100647  32.505240  32.742576  32.742988  34.999505  34.315164  34.724530
## 23   7.976220   8.198884   8.297440   8.361919   8.338112   8.254457   8.266415
## 24  27.038178  27.814486  27.168813  27.187196  28.428159  28.319115  28.741628
## 25  32.348983  32.096231  32.006847  33.745080  31.617354  32.571774  34.257358
## 26   5.123453   5.493741   5.567185   5.580729   6.026916   5.835288   5.808275
## 27  61.589314  60.040156  59.398155  59.429211  60.626881  59.990129  60.445349
## 28  55.724204  55.175468  54.510884  55.591955  61.826658  60.373331  60.410913
## 29  84.945551  88.501401  89.110345  89.342659  88.215063  88.142202  88.452255
## 30 126.300516 128.604263 133.167735 131.313640 135.472886 138.204982 137.486174
## 31   9.697611   9.651947   9.866933  10.041912   9.792874   9.778198   9.817061
## 32  45.056212  45.512083  45.709820  46.162142  49.931989  48.812671  48.359497
## 33  38.584113  42.187429  41.832393  41.828756  45.004880  44.699927  45.024212
## 34  48.338062  49.195700  49.784957  49.620676  50.064929  50.017479  49.438661
## 35  68.002479  70.682909  69.630545  73.734472  78.600691  76.034389  77.746653
## 36  15.072088  15.389081  15.416217  15.591169  16.720193  16.651937  16.907866
## 37  58.355490  62.703121  60.782498  61.008910  61.249776  61.196684  61.278944
## 38  15.944698  16.485335  16.053226  16.257287  16.638241  16.247237  16.392666
## 39  75.293083  73.800433  72.218989  72.415613  76.150402  75.924803  76.466438
## 40 154.421569 151.652927 152.787389 149.653596 161.999150 158.937012 157.608434
## 41  62.994455  68.612550  69.102948  69.343871  68.634771  68.731431  69.775797
## 42   7.856087   7.970803   8.154640   8.115024   8.480890   8.511227   8.566663
## 43 311.199853 354.719481 347.576620 348.245780 357.818578 356.758868 359.157017
## 44  44.647382  38.732092  38.434832  38.448683  41.306389  40.436776  40.389818
## 45  32.121206  32.517067  32.868589  32.882305  34.280410  33.628071  33.868981
## 46  25.521427  26.192702  25.937983  26.012463  26.393515  26.177397  26.207278
## 47  11.251507  11.427373  11.484249  11.590464  11.875372  11.712284  11.772427
##    Best_Result
## 1            5
## 2            5
## 3            2
## 4            5
## 5            5
## 6            4
## 7            5
## 8            3
## 9            5
## 10           2
## 11           4
## 12           4
## 13           4
## 14           6
## 15           5
## 16           5
## 17           5
## 18           5
## 19           4
## 20           1
## 21           5
## 22           5
## 23           4
## 24           4
## 25           1
## 26           5
## 27           5
## 28           5
## 29           4
## 30           6
## 31           5
## 32           5
## 33           6
## 34           4
## 35           5
## 36           4
## 37           5
## 38             3
## 39             8
## 40             3
## 41             6
## 42             4
## 43             5
## 44             8
## 45             6
## 46             5
## 47             4
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W1,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```



WEEK 2

```
W2<-NULL

# AUTO ARIMA RESULTS
AUTO_AR_W2 <- calculate_mean_wis(AUTO_ARIMA_WEEK2_list)
ES27_AR_W2 <- calculate_mean_wis(ES27_ARIMA_WEEK2_list)
ES64_AR_W2 <- calculate_mean_wis(ES64_ARIMA_WEEK2_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W2 <- calculate_mean_wis(AUTO_ADJACENT_WEEK2_list)
```

```
ES27_ADJ_W2 <- calculate_mean_wis(ES27_ADJACENT_WEEK2_list)
ES64_ADJ_W2 <- calculate_mean_wis(ES64_ADJACENT_WEEK2_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W2 <- calculate_mean_wis(AUTO_TEMPERATURE_WEEK2_list)
ES27_TMP_W2 <- calculate_mean_wis(ES27_TEMPERATURE_WEEK2_list)
ES64_TMP_W2 <- calculate_mean_wis(ES64_TEMPERATURE_WEEK2_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W2 <- calculate_mean_wis(AUTO_EPIWEEK_WEEK2_list)
ES27_EPI_W2 <- calculate_mean_wis(ES27_EPIWEEK_WEEK2_list)
ES64_EPI_W2 <- calculate_mean_wis(ES64_EPIWEEK_WEEK2_list)

# MERGE
# AUTO ARIMA RESULTS
W2 <- merge(AUTO_AR_W2, ES27_AR_W2, by = "State")
W2 <- merge(W2, ES64_AR_W2, by = "State")
# ADJACENT ARIMAX RESULTS
W2 <- merge(W2, AUTO_ADJ_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, AUTO_ADJ_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

```
W2 <- merge(W2, ES27_ADJ_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES27_ADJ_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

```
W2 <- merge(W2, ES64_ADJ_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES64_ADJ_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the
## result
```

```
# TEMPERATURE ARIMAX RESULTS
W2 <- merge(W2, AUTO_TMP_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, AUTO_TMP_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the
## result
```

```
W2 <- merge(W2, ES27_TMP_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES27_TMP_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’ are duplicated in the result
```

```
W2 <- merge(W2, ES64_TMP_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES64_TMP_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’ are duplicated in the result
```

```r
# EPIWEEK ARIMAX RESULTS
W2 <- merge(W2, AUTO_EPI_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, AUTO_EPI_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

```r
W2 <- merge(W2, ES27_EPI_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES27_EPI_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

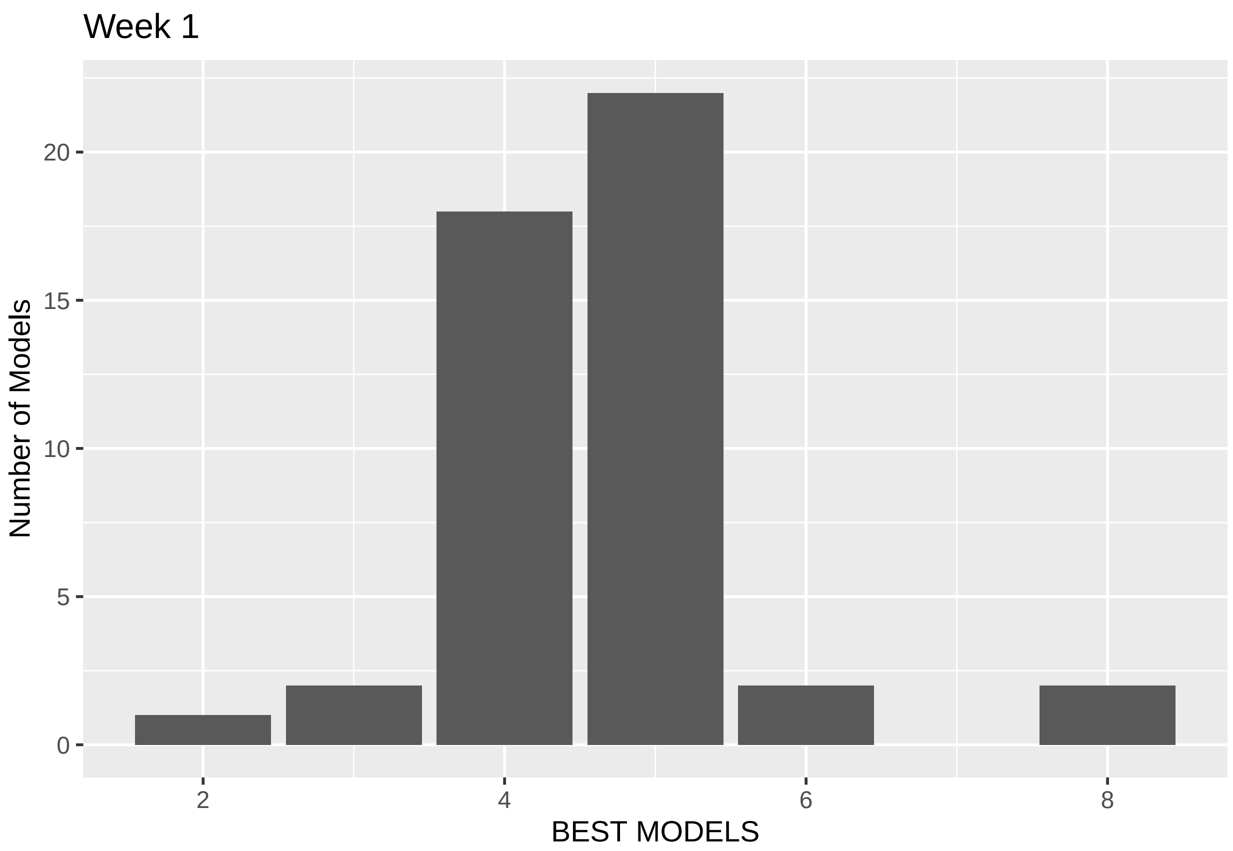
```r
W2 <- merge(W2, ES64_EPI_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES64_EPI_W2, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are
## duplicated in the result
```

```r
# Rename columns for clarity
colnames(W2)[1] <- "NAME"
colnames(W2)[2] <- "AUTO_AR"
colnames(W2)[3] <- "ES27_AR"
colnames(W2)[4] <- "ES64_AR"
colnames(W2)[5] <- "AUTO_ADJ"
colnames(W2)[6] <- "ES27_ADJ"
colnames(W2)[7] <- "ES64_ADJ"
colnames(W2)[8] <- "AUTO_TMP"
colnames(W2)[9] <- "ES27_TMP"
colnames(W2)[10] <- "ES64_TMP"
colnames(W2)[11] <- "AUTO_EPI"
colnames(W2)[12] <- "ES27_EPI"
colnames(W2)[13] <- "ES64_EPI"

# Identify the best result for each state
W2$Best_Result <- apply(W2[,2:13], 1, function(x) {
  which.min(x)
})

#W2$Best_Result <- apply(W2[, 2:13], 1, function(x) {
#  colnames(W2)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W2)
```

```
##              NAME    AUTO_AR    ES27_AR    ES64_AR   AUTO_ADJ   ES27_ADJ
## 1         Alabama 176.537953 176.723664 178.270971 171.945828 172.523187
## 2         Arizona  66.815674  67.011163  67.569602  67.073182  63.086388
## 3        Arkansas  35.988069  35.233747  35.462207  35.124756  35.530550
## 4      California 260.190969 254.821031 252.380743 255.551566 247.323178
## 5        Colorado 105.104900 103.190001 103.309532  99.597100 100.354377
## 6     Connecticut  81.703350  82.351383  83.307293  79.081846  80.249848
## 7        Delaware  11.335572  11.676090  11.703639  10.581663  10.325811
## 8         Georgia 286.788145 284.497118 283.876076 286.012780 282.468759
## 9           Idaho  20.383488  20.184826  20.167822  18.893228  18.493967
## 10       Illinois 163.036658 148.222347 149.239458 156.568150 149.514734
## 11        Indiana  68.632869  67.558880  67.899124  65.340748  66.909407
## 12           Iowa  15.678834  16.702021  17.207221  14.174478  15.430881
## 13         Kansas  43.210269  44.385918  43.823589  38.292898  38.585370
## 14       Kentucky  77.823414  78.795003  76.587965  76.785424  76.190543
## 15      Louisiana 220.288563 207.306981 211.144748 206.945884 202.096916
## 16          Maine  26.564275  25.557656  26.060098  26.610437  24.946784
## 17       Maryland  94.579331 135.697877 106.160551  93.381153  89.446630
## 18  Massachusetts 125.668162 116.365472 119.685813 114.729407 113.610555
## 19       Michigan  96.600006  96.721650  97.186896  91.388586  93.656459
## 20      Minnesota  41.979617  41.696685  41.850570  41.235795  41.804507
## 21    Mississippi  83.434981  81.900136  81.680289  83.693821  83.067016
## 22       Missouri  48.548585  49.277064  48.654850  46.061768  45.934974
## 23        Montana  11.823312  11.832042  11.853525  10.825429  10.754701
## 24       Nebraska  39.493531  38.584886  38.596887  36.770920  37.011602
## 25         Nevada  42.822686  42.502291  46.416569  41.220605  41.725533
## 26  New Hampshire   7.422284   7.529719   7.467935   6.831063   6.647625
## 27     New Jersey 102.319460  99.423556  98.882156  97.322599  97.054351
## 28     New Mexico  88.486853  88.239186  89.544719  81.370033  82.026672
## 29       New York 141.114246 143.460016 145.487807 131.873309 132.445692
## 30 North Carolina 213.131452 220.065989 211.956697 210.169441 205.374862
## 31   North Dakota  12.310758  12.603470  12.807870  11.863921  11.640484
## 32           Ohio  71.797372  70.931577  70.695491  70.016408  67.395008
## 33       Oklahoma  56.052265  54.572391  53.557412  51.921548  52.242312
## 34         Oregon  73.535050  74.060843  73.935678  68.059521  69.520345
## 35   Pennsylvania 110.338439 104.276590 108.902767  95.408164  95.111392
## 36   Rhode Island  22.192180  23.367774  23.152091  20.865902  21.539149
## 37 South Carolina  86.883526  85.230721  83.671310  83.283351  85.507441
## 38   South Dakota  20.983752  20.502093  20.433740  20.752354  20.327296
## 39      Tennessee 128.675991 114.994598 113.841298 114.966168 114.647497
## 40          Texas 215.773614 215.000594 207.596274 224.787600 220.735901
## 41           Utah  84.980759  83.540837  84.020632  73.688008  74.642665
## 42        Vermont  11.048837  10.991187  10.924071  10.055391  10.107192
## 43       Virginia 485.768052 487.812793 489.334907 458.507131 453.306255
## 44     Washington  67.336245  66.070139  65.778381  68.852334  68.411147
## 45  West Virginia  48.220102  47.328666  46.986035  47.022824  46.215014
## 46      Wisconsin  38.134117  38.384247  38.252974  36.976331  36.274873
## 47        Wyoming  15.733536  15.283358  15.702952  14.365236  14.677665
##      ES64_ADJ   AUTO_TMP   ES27_TMP   ES64_TMP  AUTO_EPI   ES27_EPI   ES64_EPI
## 1  177.759917 175.999346 175.204711 179.157414 180.54717 178.182082 174.874866
## 2   66.231978  66.829239  65.878244  66.159052  69.35285  67.996908  67.710822
## 3   36.716284  36.459882  35.665916  35.159198  36.66010  36.892762  36.666106
## 4  261.979447 263.998791 259.190185 254.484011 270.69366 260.596946 261.695345
## 5  106.686906 105.292494 104.495029 104.660565 107.52025 105.350861 104.828701
## 6   87.335629  82.326611  85.302906  88.433513  84.00331  82.093527  85.107705
## 7   10.562091  11.687616  11.587710  11.571581  11.84038  11.963709  11.895183
## 8  304.913092 292.741004 288.538447 287.002865 299.33805 290.161846 289.802951
## 9   19.935513  20.753086  20.322290  20.287852  20.11235  20.072261  20.071252
## 10 157.680262 159.456790 151.308448 148.701505 166.85986 159.640016 159.149693
## 11  69.133104  70.200476  67.827601  68.092977  69.56768  69.633397  69.993989
## 12  14.951115  17.693839  17.086857  17.519532  16.29632  16.259438  16.811977
## 13  38.210673  43.445983  44.758418  43.274181  43.96564  43.405009  43.037156
## 14  76.613197  77.240942  79.881898  77.819243  82.91322  81.284691  80.576202
## 15 229.088538 217.767399 209.701846 210.356882 223.90444 218.249150 218.718888
## 16  26.003109  26.261745  26.019511  26.202462  26.97331  25.574103  25.602158
## 17  92.183836  93.650010 125.915729 101.867208  91.56233 109.486318  98.565707
## 18 126.455889 119.583333 116.527780 119.814225 125.87846 121.199458 123.547505
## 19 100.415309  99.996684  97.562777  98.809899 103.75573 100.784903 101.827388
## 20  41.276790  42.061427  41.504808  41.667529  42.85301  42.362353  42.295161
## 21  89.376788  83.787122  83.072094  82.211854  88.75142  88.845873  88.769248
## 22  47.693059  49.322636  49.755490  48.923013  50.58961  49.758569  50.338231
## 23  11.343138  11.543813  11.747071  11.809004  12.00583  11.816055  11.807627
## 24  39.207577  39.280771  38.231621  38.332611  39.53577  39.562344  39.922863
## 25  44.026275  43.767378  42.254758  45.359992  43.50707  43.897335  45.105149
## 26   6.738505   7.147625   7.369604   7.315636   7.74029   7.601517   7.534639
## 27 102.302313 101.288635 100.149978 100.183649 101.18181  99.796314 100.548672
## 28  89.469402  90.194277  89.386450  91.001371  97.43801  94.831447  94.857281
## 29 133.002102 137.340480 142.239389 142.749431 136.13942 137.246541 137.616144
## 30 206.190326 209.009562 222.497763 215.132556 212.33106 216.450558 211.379569
## 31  12.195738  12.038334  12.349878  12.590092  12.26224  12.464026  12.683321
## 32  73.731760  72.297576  72.069795  72.196264  77.04502  74.952155  74.289041
## 33  49.541307  55.193883  55.026923  54.552063  55.75921  55.915764  56.168437
## 34  72.557767  72.288591  73.464198  73.270738  74.85869  73.259424  73.292834
## 35 104.847122 108.012073 105.555054 112.668580 115.16108 110.657381 114.686379
## 36  21.969431  22.487478  22.767756  23.026000  24.44596  24.342970  24.655375
## 37  85.666309  89.016934  85.651525  85.132177  84.88862  85.786358  85.167163
## 38  20.691892  21.348947  20.552581  20.951146  20.83500  20.374750  20.552212
## 39 120.881678 116.305409 113.778811 113.969551 118.91600 117.139713 119.030238
## 40 223.830870 213.081206 215.432380 209.287315 224.04758 219.963653 219.004113
## 41  75.187079  79.048537  79.912061  80.634291  81.39304  80.453997  81.160573
## 42  10.670236  10.608157  10.843861  10.726058  11.26278  11.276611  11.276699
## 43 467.173161 503.631833 492.351906 490.795868 501.24830 501.029044 500.283609
## 44  73.762702  66.262276  64.632992  65.032837  69.46437  67.471431  67.794534
## 45  47.367625  47.589558  47.570605  47.062501  48.25616  48.061682  47.978103
## 46  37.513110  38.770255  38.729582  38.582678  38.69105  38.107989  38.067228
## 47  15.298264  15.514178  15.642927  16.013436  15.96742  15.573718  15.800940
##    Best_Result
## 1            4
## 2            5
## 3            4
## 4            5
## 5            4
## 6            4
## 7            5
## 8            5
## 9            5
## 10           2
## 11           4
## 12           4
## 13           6
## 14           5
## 15           5
## 16           5
## 17           5
## 18           5
## 19           4
## 20           4
## 21           3
## 22           5
## 23           5
## 24           4
## 25           4
## 26           5
## 27           5
## 28           4
## 29           4
## 30           5
## 31           5
## 32           5
## 33           6
## 34           4
## 35           5
## 36           4
## 37           4
## 38           5
## 39           8
## 40           3
## 41           4
## 42           4
## 43           5
## 44           8
## 45           5
## 46           5
## 47           4
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W2,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```



WEEK 3

```
W3<-NULL

# AUTO ARIMA RESULTS
AUTO_AR_W3 <- calculate_mean_wis(AUTO_ARIMA_WEEK3_list)
ES27_AR_W3 <- calculate_mean_wis(ES27_ARIMA_WEEK3_list)
ES64_AR_W3 <- calculate_mean_wis(ES64_ARIMA_WEEK3_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W3 <- calculate_mean_wis(AUTO_ADJACENT_WEEK3_list)
ES27_ADJ_W3 <- calculate_mean_wis(ES27_ADJACENT_WEEK3_list)
ES64_ADJ_W3 <- calculate_mean_wis(ES64_ADJACENT_WEEK3_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W3 <- calculate_mean_wis(AUTO_TEMPERATURE_WEEK3_list)
ES27_TMP_W3 <- calculate_mean_wis(ES27_TEMPERATURE_WEEK3_list)
ES64_TMP_W3 <- calculate_mean_wis(ES64_TEMPERATURE_WEEK3_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W3 <- calculate_mean_wis(AUTO_EPIWEEK_WEEK3_list)
ES27_EPI_W3 <- calculate_mean_wis(ES27_EPIWEEK_WEEK3_list)
ES64_EPI_W3 <- calculate_mean_wis(ES64_EPIWEEK_WEEK3_list)

# MERGE
# AUTO ARIMA RESULTS
W3 <- merge(AUTO_AR_W3, ES27_AR_W3, by = "State")
W3 <- merge(W3, ES64_AR_W3, by = "State")
# ADJACENT ARIMAX RESULTS
```

```r
W3 <- merge(W3, AUTO_ADJ_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, AUTO_ADJ_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

```r
W3 <- merge(W3, ES27_ADJ_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES27_ADJ_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

```r
W3 <- merge(W3, ES64_ADJ_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES64_ADJ_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the
## result
```

```r
# TEMPERATURE ARIMAX RESULTS
W3 <- merge(W3, AUTO_TMP_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, AUTO_TMP_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the
## result
```

```r
W3 <- merge(W3, ES27_TMP_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES27_TMP_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y' are duplicated in the result
```

```r
W3 <- merge(W3, ES64_TMP_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES64_TMP_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y' are duplicated in the result
```

```r
# EPIWEEK ARIMAX RESULTS
W3 <- merge(W3, AUTO_EPI_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, AUTO_EPI_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

```r
W3 <- merge(W3, ES27_EPI_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES27_EPI_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

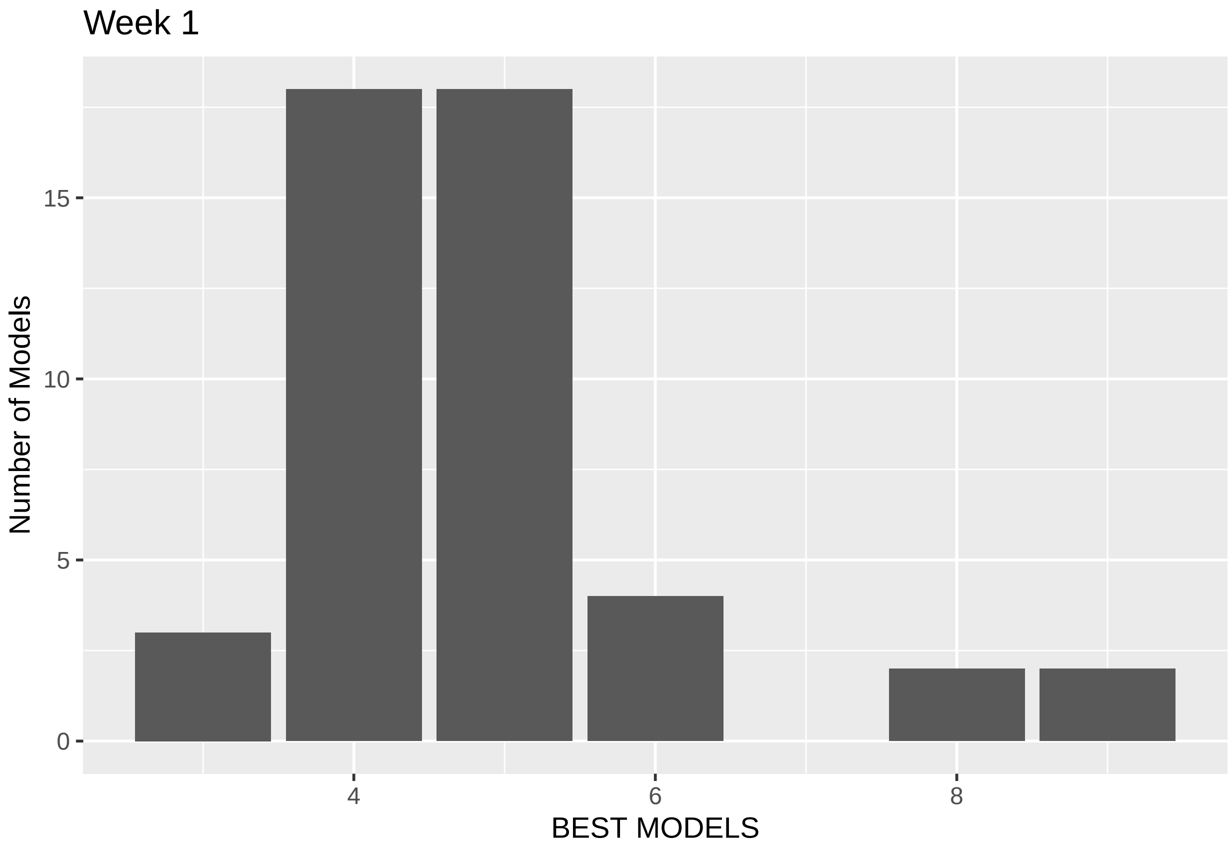
```
W3 <- merge(W3, ES64_EPI_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES64_EPI_W3, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are
## duplicated in the result
```

```
# Rename columns for clarity
colnames(W3)[1] <- "NAME"
colnames(W3)[2] <- "AUTO_AR"
colnames(W3)[3] <- "ES27_AR"
colnames(W3)[4] <- "ES64_AR"
colnames(W3)[5] <- "AUTO_ADJ"
colnames(W3)[6] <- "ES27_ADJ"
colnames(W3)[7] <- "ES64_ADJ"
colnames(W3)[8] <- "AUTO_TMP"
colnames(W3)[9] <- "ES27_TMP"
colnames(W3)[10] <- "ES64_TMP"
colnames(W3)[11] <- "AUTO_EPI"
colnames(W3)[12] <- "ES27_EPI"
colnames(W3)[13] <- "ES64_EPI"

# Identify the best result for each state
W3$Best_Result <- apply(W3[,2:13], 1, function(x) {
  which.min(x)
})

#W3$Best_Result <- apply(W3[, 2:13], 1, function(x) {
#  colnames(W3)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W3)
```

```
##              NAME   AUTO_AR   ES27_AR    ES64_AR   AUTO_ADJ   ES27_ADJ
## 1         Alabama 234.88777 232.77346 232.892408 225.509424 228.689368
## 2         Arizona  85.84635  86.91635  86.336327  89.327887  81.796437
## 3        Arkansas  46.63206  45.93368  46.038931  43.846119  44.930467
## 4      California 356.94659 347.46636 342.955148 353.554748 338.968707
## 5        Colorado 142.10265 138.55124 138.780390 133.702200 136.434875
## 6     Connecticut 106.59585 111.68002 110.472275 102.603506 105.208292
## 7        Delaware  13.95322  14.69995  14.552080  13.682627  13.113199
## 8         Georgia 382.45172 382.33572 381.387649 381.820434 382.970453
## 9           Idaho  25.77925  25.75376  25.634540  24.107784  23.396980
## 10       Illinois 223.22321 199.63879 198.893416 216.165930 202.803437
## 11        Indiana  92.26381  90.35761  90.803566  87.560174  90.950955
## 12           Iowa  18.86642  20.34778  20.759427  17.248982  18.806055
## 13         Kansas  56.99840  61.57436  59.109262  52.836209  52.968726
## 14       Kentucky 102.59049 105.28931 100.063481 100.776646  99.301234
## 15      Louisiana 297.32004 275.64355 277.384035 284.819363 274.414935
## 16          Maine  34.74827  33.91935  34.525280  34.855741  33.281350
## 17       Maryland 122.14664 198.78245 135.065222 119.841278 113.683489
## 18  Massachusetts 178.09434 162.45365 166.534324 161.269305 161.160483
## 19       Michigan 130.61401 130.58946 130.999004 123.668804 127.064715
## 20      Minnesota  54.51208  53.57193  53.518858  52.540309  53.462011
## 21    Mississippi 107.97358 105.09892 104.236017 111.618686 110.858837
## 22       Missouri  64.15799  66.85271  64.536243  61.165787  61.192385
## 23        Montana  15.32291  15.36393  15.338157  13.926956  13.905109
## 24       Nebraska  51.31918  50.37280  50.198524  47.209828  47.473513
## 25         Nevada  54.46662  53.21834  54.486630  52.566309  52.614950
## 26  New Hampshire   9.18898   9.36836   9.263103   8.587303   8.243472
## 27     New Jersey 137.07957 132.39404 131.925562 130.529376 129.773633
## 28     New Mexico 121.19504 121.08841 123.136606 112.206792 112.673937
## 29       New York 191.69772 201.39119 210.983768 171.465968 175.318664
## 30 North Carolina 282.97467 308.24313 285.760362 280.994460 277.009764
## 31   North Dakota  14.76193  15.18298  15.345313  14.092342  13.917991
## 32           Ohio  96.47758  95.08914  93.935820  96.325567  91.352276
## 33       Oklahoma  68.74801  67.53286  65.112828  61.352577  62.561973
## 34         Oregon  97.42877  98.13785  97.047406  89.871315  92.247560
## 35   Pennsylvania 147.07249 138.99821 142.205900 127.477726 127.058103
## 36   Rhode Island  28.75141  30.98661  30.569703  27.227060  28.279085
## 37 South Carolina 109.54230 107.38710 103.978216 106.796410 108.762610
## 38   South Dakota  26.42528  25.71227  25.733651  25.702586  25.202925
## 39      Tennessee 174.26536 147.08732 148.516072 149.221653 149.946325
## 40          Texas 273.45252 274.63644 263.635194 284.762506 281.456800
## 41           Utah 100.75554  98.09596  97.996969  86.298930  86.762056
## 42        Vermont  14.03092  13.95116  13.728550  12.848965  12.900337
## 43       Virginia 622.00344 622.00660 620.795190 595.960401 596.306662
## 44     Washington  94.74162  93.54929  92.831934  95.432880  95.614698
## 45  West Virginia  62.50768  60.49049  59.176679  59.334265  58.276924
## 46      Wisconsin  48.84304  50.16748  49.891798  47.890966  47.129473
## 47        Wyoming  19.35988  18.92365  18.979181  17.812543  18.197674
##      ES64_ADJ   AUTO_TMP   ES27_TMP   ES64_TMP   AUTO_EPI   ES27_EPI   ES64_EPI
## 1  233.742374 233.298582 230.857535 232.486760 236.440386 232.476890 227.207831
## 2   86.834053  85.402729  84.049601  83.720615  90.404135  89.001799  88.166973
## 3   47.184470  46.924621  46.373742  43.972949  46.742982  47.215122  46.450968
## 4  360.003627 361.107930 355.074858 348.090425 374.256183 359.892507 360.095054
## 5  142.012837 141.472933 140.575022 140.360941 143.395079 139.775631 139.552670
## 6  115.113760 107.295405 115.286188 116.489498 108.212440 108.311549 110.329453
## 7   13.416528  14.438258  14.572935  14.469553  14.623885  14.945571  14.715249
## 8  405.305627 388.901019 386.923556 384.900314 397.930583 386.107650 385.906455
## 9   25.197945  26.314003  25.840118  25.700186  25.350023  25.183075  25.202334
## 10 212.318961 217.946678 201.304891 196.350007 224.847257 214.575591 213.077878
## 11  91.770591  93.781861  90.658732  91.219310  90.738747  93.257986  93.484980
## 12  18.049196  21.409317  20.590238  21.188147  19.410519  19.461394  19.893449
## 13  51.171634  57.448064  61.667764  57.811724  58.274950  58.616703  57.740675
## 14 100.785081 102.167043 105.987610 100.800259 110.163747 107.624610 105.685968
## 15 304.781129 294.650392 279.629560 279.616272 295.573282 288.864916 287.970641
## 16  34.304260  34.360803  33.906642  34.173851  34.733917  33.352648  33.615529
## 17 118.566146 122.375412 178.277626 129.172214 118.568081 147.899058 122.988905
## 18 173.079833 165.780961 161.196423 165.523239 174.499744 166.058243 168.466868
## 19 136.050196 135.582557 130.818963 133.075570 137.919336 133.662115 134.402668
## 20  52.768540  54.013129  52.819484  52.880563  54.860069  54.159561  53.813844
## 21 117.734879 106.725739 106.025730 103.944805 115.737078 114.660702 114.626940
## 22  63.241271  64.898867  66.953379  64.798451  65.846006  64.793824  64.156234
## 23  14.407759  14.831780  15.072418  15.093064  15.479671  15.177351  15.154206
## 24  50.641474  50.694029  49.414149  49.541685  51.791903  50.867067  51.249549
## 25  53.271810  55.344624  52.573786  53.676224  55.876923  54.161054  52.892242
## 26   8.308399   8.723703   9.076238   8.968081   9.411445   9.321972   9.223003
## 27 135.838039 135.088901 133.403375 133.308321 133.171069 130.955892 132.283198
## 28 120.974336 123.413674 122.845566 125.155525 133.320974 128.719672 129.211245
## 29 174.957398 181.529390 196.793160 199.977195 177.035562 181.957163 182.546800
## 30 275.038675 279.510594 311.095730 289.468893 276.697806 287.654944 278.097641
## 31  14.533661  14.354552  14.806029  14.977569  14.718282  15.056793  15.279096
## 32 100.329516  96.390267  95.785898  95.332418 102.420559  99.290142  98.135202
## 33  60.402618  67.537322  68.176558  66.003859  67.975551  68.546102  69.022222
## 34  95.190478  93.408523  95.871991  94.963319  99.986745  97.096238  96.854628
## 35 139.342244 142.550403 138.915417 146.446857 148.607017 141.706878 147.882086
## 36  27.864428  28.960032  29.602997  30.037079  31.852133  31.547313  31.976855
## 37 109.131918 112.213453 107.600452 105.957442 105.977715 107.089747 105.666524
## 38  25.788388  26.152071  25.522818  26.376190  27.148684  26.131302  26.464438
## 39 156.533794 151.102104 146.980564 149.367532 153.364759 149.589433 154.029723
## 40 283.211480 271.375261 273.468383 265.645310 286.806219 280.744134 275.937668
## 41  86.846443  90.560710  91.672803  92.359031  94.154108  93.135694  93.035248
## 42  13.518928  13.372936  13.690038  13.364248  14.368526  14.431681  14.441384
## 43 591.539935 646.927848 626.148345 620.026940 640.077386 642.180412 635.218559
## 44 100.465233  94.236885  91.452267  92.098756  96.385151  93.688977  94.480084
## 45  60.661698  61.180763  60.554213  59.022032  61.793951  60.986596  60.561643
## 46  47.899661  49.855705  50.391890  50.140282  49.490803  49.198424  49.041052
## 47  18.806280  18.955759  19.458908  19.420472  19.664465  19.089520  19.071822
##    Best_Result
## 1            4
## 2            5
## 3            4
## 4            5
## 5            4
## 6            4
## 7            5
## 8            3
## 9            5
## 10           9
## 11           4
## 12           4
## 13           6
## 14           5
## 15           5
## 16           5
## 17           5
## 18           5
## 19           4
## 20           4
## 21           9
## 22           4
## 23           5
## 24           4
## 25           4
## 26           5
## 27           5
## 28           4
## 29           4
## 30             6
## 31             5
## 32             5
## 33             6
## 34             4
## 35             5
## 36             4
## 37             3
## 38             5
## 39             8
## 40             3
## 41             4
## 42             4
## 43             6
## 44             8
## 45             5
## 46             5
## 47             4
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W3,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```



WEEK 4

```
W4<-NULL

# AUTO ARIMA RESULTS
AUTO_AR_W4 <- calculate_mean_wis(AUTO_ARIMA_WEEK4_list)
ES27_AR_W4 <- calculate_mean_wis(ES27_ARIMA_WEEK4_list)
ES64_AR_W4 <- calculate_mean_wis(ES64_ARIMA_WEEK4_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W4 <- calculate_mean_wis(AUTO_ADJACENT_WEEK4_list)
ES27_ADJ_W4 <- calculate_mean_wis(ES27_ADJACENT_WEEK4_list)
ES64_ADJ_W4 <- calculate_mean_wis(ES64_ADJACENT_WEEK4_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W4 <- calculate_mean_wis(AUTO_TEMPERATURE_WEEK4_list)
ES27_TMP_W4 <- calculate_mean_wis(ES27_TEMPERATURE_WEEK4_list)
ES64_TMP_W4 <- calculate_mean_wis(ES64_TEMPERATURE_WEEK4_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W4 <- calculate_mean_wis(AUTO_EPIWEEK_WEEK4_list)
ES27_EPI_W4 <- calculate_mean_wis(ES27_EPIWEEK_WEEK4_list)
ES64_EPI_W4 <- calculate_mean_wis(ES64_EPIWEEK_WEEK4_list)

# MERGE
# AUTO ARIMA RESULTS
W4 <- merge(AUTO_AR_W4, ES27_AR_W4, by = "State")
W4 <- merge(W4, ES64_AR_W4, by = "State")
# ADJACENT ARIMAX RESULTS
W4 <- merge(W4, AUTO_ADJ_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, AUTO_ADJ_W4, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

```
W4 <- merge(W4, ES27_ADJ_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES27_ADJ_W4, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the result
```

```
W4 <- merge(W4, ES64_ADJ_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES64_ADJ_W4, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the
## result
```

```
# TEMPERATURE ARIMAX RESULTS
W4 <- merge(W4, AUTO_TMP_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, AUTO_TMP_W4, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’ are duplicated in the
## result
```

```
W4 <- merge(W4, ES27_TMP_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES27_TMP_W4, by = "State"): column names
## ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’, ’Mean_WIS.y’, ’Mean_WIS.x’,
## ’Mean_WIS.y’ are duplicated in the result
```

```r
W4 <- merge(W4, ES64_TMP_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES64_TMP_W4, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y' are duplicated in the result
```

```r
# EPIWEEK ARIMAX RESULTS
W4 <- merge(W4, AUTO_EPI_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, AUTO_EPI_W4, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

```r
W4 <- merge(W4, ES27_EPI_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES27_EPI_W4, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are duplicated in the result
```

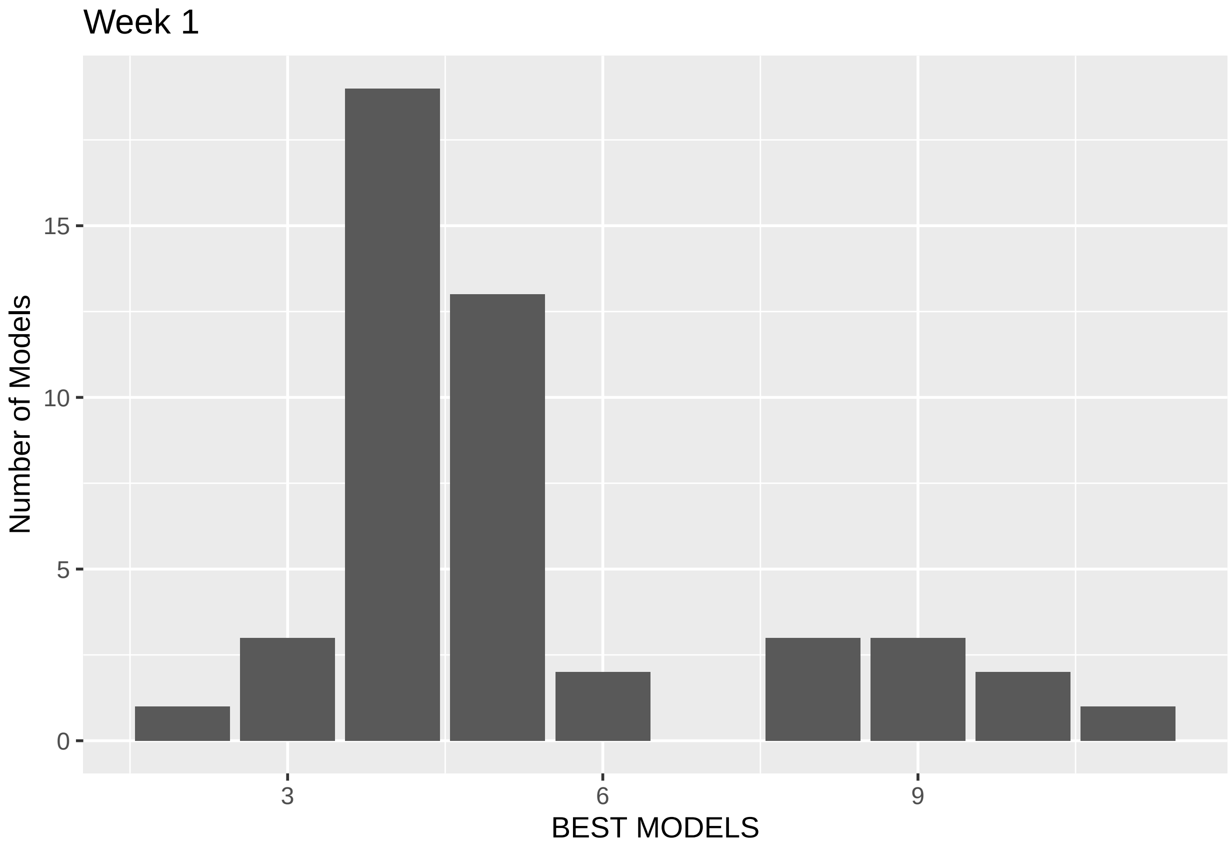
```r
W4 <- merge(W4, ES64_EPI_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES64_EPI_W4, by = "State"): column names
## 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x',
## 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y', 'Mean_WIS.x', 'Mean_WIS.y' are
## duplicated in the result
```

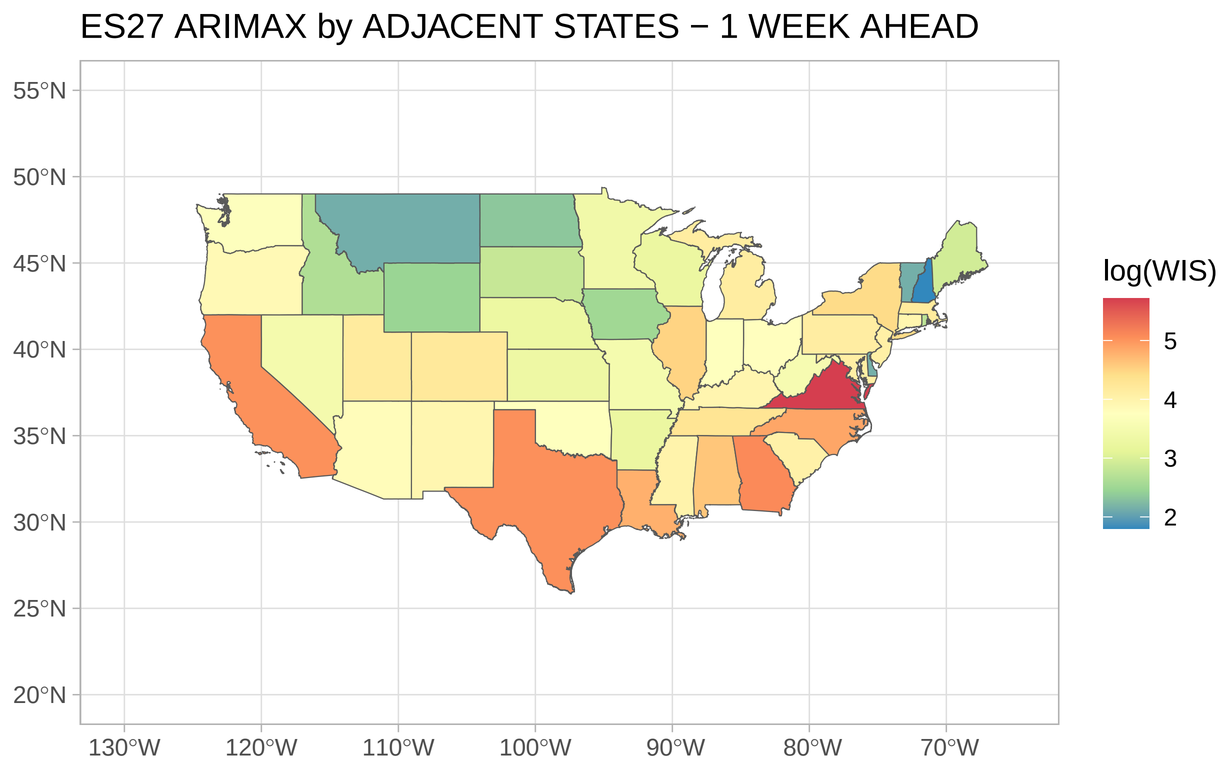
```r
# Rename columns for clarity
colnames(W4)[1] <- "NAME"
colnames(W4)[2] <- "AUTO_AR"
colnames(W4)[3] <- "ES27_AR"
colnames(W4)[4] <- "ES64_AR"
colnames(W4)[5] <- "AUTO_ADJ"
colnames(W4)[6] <- "ES27_ADJ"
colnames(W4)[7] <- "ES64_ADJ"
colnames(W4)[8] <- "AUTO_TMP"
colnames(W4)[9] <- "ES27_TMP"
colnames(W4)[10] <- "ES64_TMP"
colnames(W4)[11] <- "AUTO_EPI"
colnames(W4)[12] <- "ES27_EPI"
colnames(W4)[13] <- "ES64_EPI"

# Identify the best result for each state
W4$Best_Result <- apply(W4[,2:13], 1, function(x) {
  which.min(x)
})

#W4$Best_Result <- apply(W4[, 2:13], 1, function(x) {
#  colnames(W4)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W4)
```

```
##              NAME   AUTO_AR   ES27_AR   ES64_AR  AUTO_ADJ   ES27_ADJ   ES64_ADJ
## 1         Alabama 275.28952 271.45712 269.04162 261.92524 267.682639 276.108582
## 2         Arizona 102.65395 105.12352 103.90901 110.11321  99.902263 105.716014
## 3        Arkansas  55.66950  55.81443  55.47700  52.27615  53.981024  56.752128
## 4      California 453.37949 440.40877 430.85543 448.98797 429.570798 453.964446
## 5        Colorado 173.99421 170.30228 170.07398 162.98801 167.801306 172.221817
## 6     Connecticut 130.37342 146.30924 141.14141 122.83695 132.297504 142.936146
## 7        Delaware  16.36467  17.76383  17.40695  16.48705  16.003169  16.075612
## 8         Georgia 449.46077 450.56964 448.39217 445.04795 452.022182 477.563164
## 9           Idaho  30.37431  30.63553  30.16207  28.94752  28.098265  29.670479
## 10       Illinois 278.73206 246.78286 244.37859 269.51249 251.417538 262.853383
## 11        Indiana 112.92707 109.79137 110.51230 105.79183 111.284681 111.207831
## 12           Iowa  22.57385  24.60541  24.57803  20.67312  22.579474  21.299611
## 13         Kansas  69.46565  78.78503  73.45188  65.99016  66.671073  62.800100
## 14       Kentucky 125.34396 129.34715 120.67508 120.85554 120.090601 121.393174
## 15      Louisiana 365.74703 335.93293 334.46851 355.76238 340.913197 374.799991
## 16          Maine  41.78094  41.46243  42.12398  41.51798  40.730326  41.698353
## 17       Maryland 149.24724 733.69421 234.48965 145.72381 140.083223 145.023930
## 18  Massachusetts 226.40215 201.33029 206.68924 203.08672 203.535597 214.901084
## 19       Michigan 159.96901 158.62809 160.38104 151.22290 156.299964 165.495224
## 20      Minnesota  66.11422  64.58375  64.57388  62.49104  63.991585  63.143208
## 21    Mississippi 130.11600 124.30438 122.05993 136.94496 135.641761 142.506706
## 22       Missouri  78.18526  84.11680  79.91374  75.66274  76.130928  77.842996
## 23        Montana  18.56805  18.63218  18.54294  16.76713  16.791032  17.233154
## 24       Nebraska  63.24425  62.02816  61.72105  57.84857  57.987423  61.218653
## 25         Nevada  64.50978  63.33068  67.01789  64.44030  64.126041  65.497023
## 26  New Hampshire  10.65167  11.00257  10.86133  10.24920   9.701992   9.760411
## 27     New Jersey 165.92795 160.28125 160.14978 158.47891 156.941639 164.075768
## 28     New Mexico 148.46881 148.95697 151.29844 139.36502 140.368566 146.806472
## 29       New York 244.30413 297.18880 405.77634 205.58741 218.735111 223.248499
## 30 North Carolina 350.40558 414.32465 356.89108 339.50127 335.717818 333.817042
## 31   North Dakota  17.05547  17.69706  17.81904  16.18920  16.087761  16.680732
## 32           Ohio 118.26267 116.82928 114.79082 121.20163 113.826267 125.355154
## 33       Oklahoma  80.13939  79.00770  75.38583  70.62465  72.279888  70.785027
## 34         Oregon 122.14616 122.71701 121.46414 111.04903 114.288542 116.117474
## 35   Pennsylvania 178.71436 168.57082 170.02326 155.64085 155.455868 169.431438
## 36   Rhode Island  34.18871  38.10496  37.09671  32.54901  34.201514  32.705373
## 37 South Carolina 132.12477 129.21029 124.03297 126.94787 129.369571 130.191404
## 38   South Dakota  31.18774  30.46427  30.09064  30.27982  29.598961  30.019995
## 39      Tennessee 217.97053 176.34099 179.64421 178.28043 179.372492 186.138565
## 40          Texas 322.20884 328.09457 312.78338 338.14553 333.162205 335.597156
## 41           Utah 114.99037 112.24201 112.39467  98.64905  98.843230  99.148264
## 42        Vermont  16.71528  16.75434  16.29635  15.49278  15.485540  16.174226
## 43       Virginia 719.64990 722.47928 717.41429 710.45777 716.311655 701.121805
## 44     Washington 119.47259 118.71360 117.92729 120.06737 120.310094 123.330290
## 45  West Virginia  74.72365  71.91401  69.26481  70.35248  70.133048  72.650458
## 46      Wisconsin  58.51381  60.69611  60.25077  57.49134  56.605055  57.215671
## 47        Wyoming  22.86534  22.40381  22.57182  21.54896  21.882197  22.201858
##     AUTO_TMP  ES27_TMP  ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI Best_Result
## 1  274.93785 270.40965 268.35784 278.85890 269.67093 263.56600           4
## 2  102.77459 100.92540  99.46926 109.09777 107.06585 105.55045           9
## 3   56.16726  56.44215  52.45936  56.43826  57.19076  55.85982           4
## 4  454.72572 447.55868 437.97009 478.56280 456.37471 455.32237           5
## 5  173.44942 172.78770 171.68062 174.27810 171.00633 170.60375           4
## 6  134.29882 150.17829 147.67298 133.59970 137.09017 139.77165           4
## 7   16.91836  17.37766  17.18011  17.21451  17.84350  17.47414           5
## 8  455.00149 455.28815 452.21438 463.41017 453.01010 451.12173           4
## 9   31.10590  30.66071  30.21985  30.09292  29.57366  29.49922           5
## 10 269.61454 245.34264 238.55803 278.84152 262.81420 261.76433           9
## 11 113.65792 110.25842 110.83091 109.55940 113.37926 113.36125           4
## 12  24.75662  24.33989  24.93441  23.45594  23.25002  23.79166           4
## 13  69.73908  79.09568  71.09335  71.66596  72.70940  70.45294           6
## 14 125.22878 130.22618 121.11456 134.78976 130.85609 127.28401           5
## 15 362.49402 340.98745 340.67492 360.89468 350.90908 347.46001           3
## 16  42.06156  41.21665  41.62816  41.29480  40.70106  40.98677          11
## 17 151.15576 622.81041 215.06928 146.68177 400.26870 191.14660           5
## 18 208.05875 199.83094 204.27735 217.89918 205.20754 207.75422           8
## 19 165.29750 158.93358 162.48054 166.53661 162.34406 164.20291           4
## 20  65.30063  63.53223  63.50784  65.96780  64.91319  64.70121           4
## 21 126.31930 126.01065 122.19482 138.86138 136.75980 136.43300           3
## 22  79.30001  84.23496  80.70812  79.81598  80.16727  78.57661           4
## 23  17.84534  18.13965  18.07689  18.61704  18.22965  18.18081           4
## 24  61.97667  60.81238  61.09384  63.75490  62.39947  63.03133           4
## 25  65.91617  62.67368  66.31137  66.95035  64.32181  63.11947           8
## 26  10.03544  10.63771  10.45376  10.92107  10.77389  10.66851           5
## 27 162.52093 161.14724 161.02008 159.00961 157.92357 159.44994           5
## 28 151.38789 151.45425 153.45689 164.66539 158.36048 157.87726           4
## 29 226.81573 281.03479 325.40929 215.48766 233.21022 246.78167           4
## 30 341.45141 418.40435 362.10590 329.43163 348.23195 333.79510          10
## 31  16.70007  17.37392  17.43998  17.13577  17.51611  17.70945           5
## 32 117.81400 117.32933 116.12172 125.46329 121.71854 119.66706           5
## 33  79.05033  80.45143  77.30958  80.01183  80.26801  82.83463           4
## 34 115.66801 118.19371 117.64279 122.17397 121.26655 122.21727           4
## 35 172.57510 166.97476 173.36632 179.34991 170.23680 178.33552           5
## 36  34.22888  35.77666  36.21281  37.80820  37.79716  38.26965           4
## 37 134.47243 128.95789 126.05023 123.99649 126.12503 124.35499          10
## 38  30.98946  30.14343  31.06984  32.57962  31.01956  31.28705           5
## 39 183.69164 176.58330 179.96154 183.21005 178.21473 183.28417           2
## 40 325.52035 327.19978 316.38432 344.37767 336.46845 327.20954           3
## 41 101.08233 102.70128 103.79875 106.40477 105.70162 106.18981           4
## 42  15.97944  16.47478  15.92683  17.22410  17.28672  17.23208           5
## 43 752.35768 729.29860 716.25547 747.81900 770.11571 759.75997           6
## 44 119.56465 116.03877 117.04231 119.72096 117.34819 118.81459           8
## 45  72.87408  72.04224  69.16188  73.85488  73.89317  71.99868           9
## 46  59.50334  61.20282  60.53753  58.98343  58.73286  58.40575           5
## 47  22.77595  23.25224  23.16213  23.54689  22.87595  22.67441           4
```

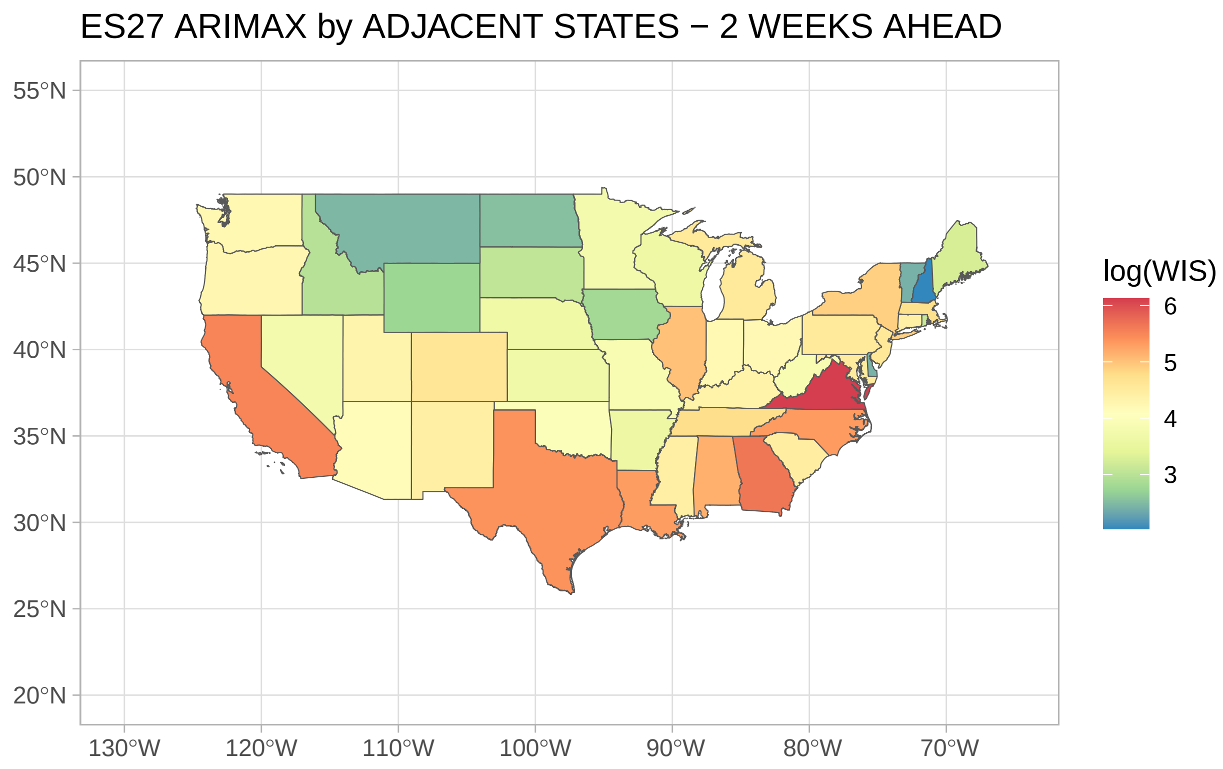
```r
# --------- WEEK1 MODELS ------------- #
ggplot(W4,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```



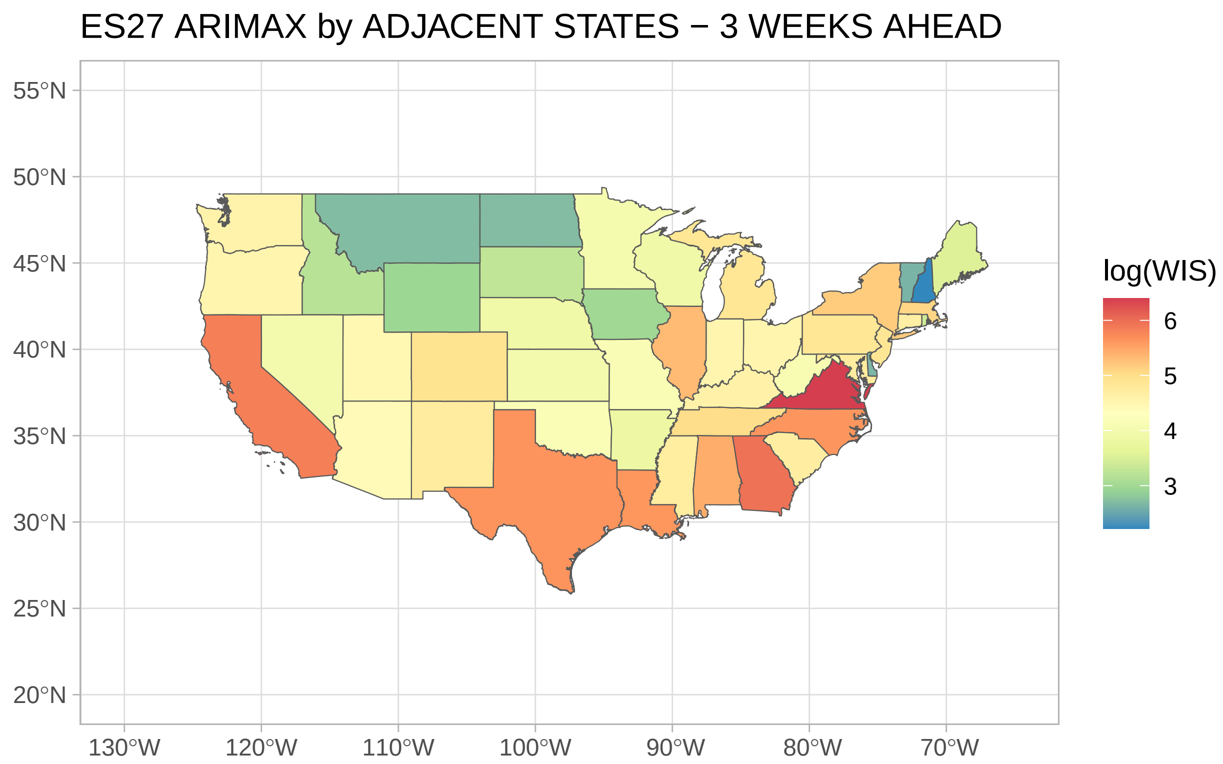
```
#######################################################
# MAPPING THE MEAN PERFORMANCE OF THE ES64, ES27      #
# AND AUTO ARIMA MODELS FOR THE 50 STATES OF THE U.S. #
# BASED ON THE SUMMARY RESULTS OF THE CURRENT MODELS. #
#######################################################

states <- read_sf("cb_2018_us_state_500k/cb_2018_us_state_500k.shp")

#################################################
# ES27 ARIMAX by ADJACENT STATES - 1 WEEK AHEAD #
#################################################

map_week1<-left_join(states, W1, by=join_by("NAME"))%>%
  drop_na()

ES_1WEEK<- ggplot(map_week1, fill ="lightgrey") +  theme_light()  + geom_sf(aes(fill=log1p(ES27_ADJ))) +

x_limits <- c(-130, -65)  # Set the desired longitude range
y_limits <- c(20, 55)     # Set the desired latitude range

ES_1WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
```



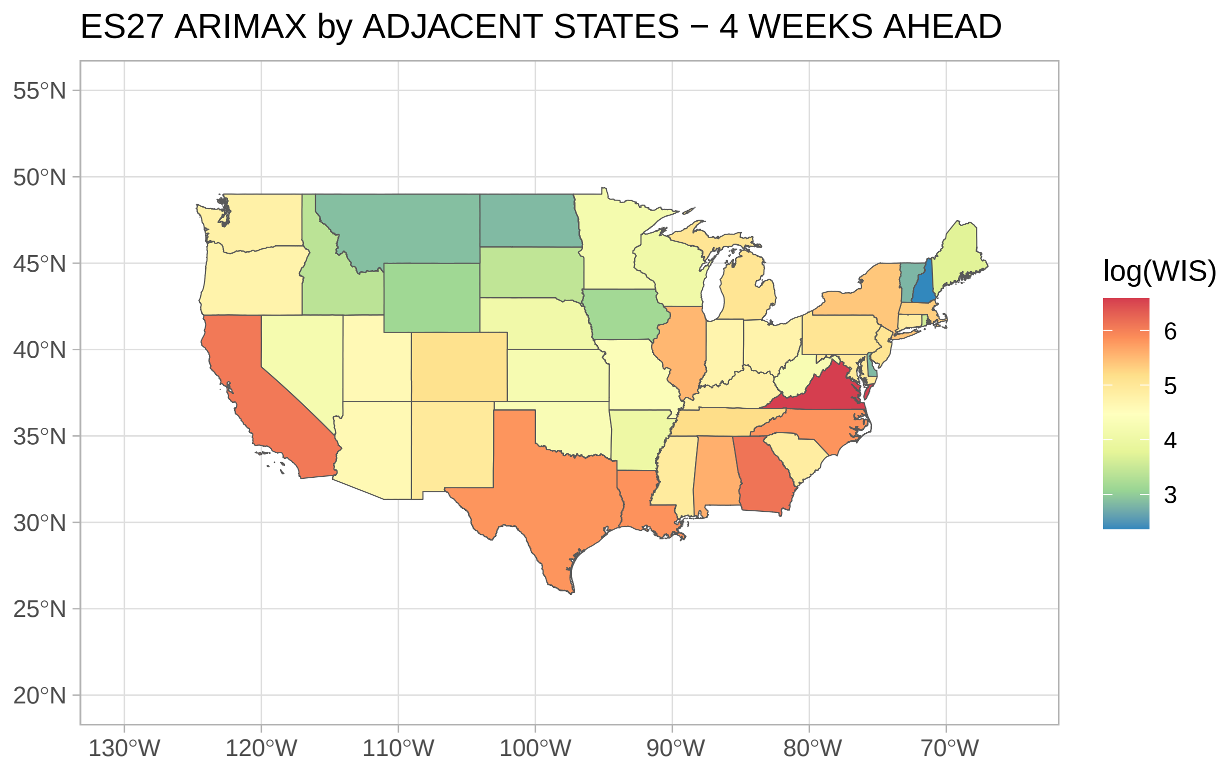
```
#####################################################
# ES27 ARIMAX by ADJACENT STATES - 2 WEEKS AHEAD #
###################################################

map_week2<-left_join(states, W2, by=join_by("NAME"))%>%
  drop_na()

MAP_WEEK2<- ggplot(map_week2, fill ="lightgrey") +  theme_light()  + geom_sf(aes(fill=log1p(ES27_ADJ)))

x_limits <- c(-130, -65)  # Set the desired longitude range
y_limits <- c(20, 55)     # Set the desired latitude range

MAP_WEEK2 + coord_sf(xlim = x_limits, ylim = y_limits)
```



```
###################################################
# ES27 ARIMAX by ADJACENT STATES - 3 WEEKS AHEAD #
#################################################

map_week3<-left_join(states, W3, by=join_by("NAME"))%>%
  drop_na()

MAP_WEEK3<- ggplot(map_week3, fill ="lightgrey") +  theme_light()  + geom_sf(aes(fill=log1p(ES27_ADJ)))

x_limits <- c(-130, -65)  # Set the desired longitude range
y_limits <- c(20, 55)     # Set the desired latitude range

MAP_WEEK3 + coord_sf(xlim = x_limits, ylim = y_limits)
```



```
#####################################################
# ES27 ARIMAX by ADJACENT STATES - 4 WEEKS AHEAD #
###################################################

map_week4<-left_join(states, W4, by=join_by("NAME"))%>%
  drop_na()

MAP_WEEK4<- ggplot(map_week4, fill ="lightgrey") +  theme_light()  + geom_sf(aes(fill=log1p(ES27_ADJ)))

x_limits <- c(-130, -65)  # Set the desired longitude range
y_limits <- c(20, 55)    # Set the desired latitude range

MAP_WEEK4 + coord_sf(xlim = x_limits, ylim = y_limits)
```



COMPARING ABSOLUTE ERRORS RESULTS FOR WEEK1

```
W1<-NULL

# AUTO ARIMA RESULTS
AUTO_AR_W1 <- calculate_mean_abs_error(AUTO_ARIMA_WEEK1_list)
ES27_AR_W1 <- calculate_mean_abs_error(ES27_ARIMA_WEEK1_list)
ES64_AR_W1 <- calculate_mean_abs_error(ES64_ARIMA_WEEK1_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W1 <- calculate_mean_abs_error(AUTO_ADJACENT_WEEK1_list)
ES27_ADJ_W1 <- calculate_mean_abs_error(ES27_ADJACENT_WEEK1_list)
ES64_ADJ_W1 <- calculate_mean_abs_error(ES64_ADJACENT_WEEK1_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W1 <- calculate_mean_abs_error(AUTO_TEMPERATURE_WEEK1_list)
ES27_TMP_W1 <- calculate_mean_abs_error(ES27_TEMPERATURE_WEEK1_list)
ES64_TMP_W1 <- calculate_mean_abs_error(ES64_TEMPERATURE_WEEK1_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W1 <- calculate_mean_abs_error(AUTO_EPIWEEK_WEEK1_list)
ES27_EPI_W1 <- calculate_mean_abs_error(ES27_EPIWEEK_WEEK1_list)
ES64_EPI_W1 <- calculate_mean_abs_error(ES64_EPIWEEK_WEEK1_list)

# MERGE
# AUTO ARIMA RESULTS
W1 <- merge(AUTO_AR_W1, ES27_AR_W1, by = "State")
W1 <- merge(W1, ES64_AR_W1, by = "State")
# ADJACENT ARIMAX RESULTS
```

```r
W1 <- merge(W1, AUTO_ADJ_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, AUTO_ADJ_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
W1 <- merge(W1, ES27_ADJ_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES27_ADJ_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
W1 <- merge(W1, ES64_ADJ_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES64_ADJ_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```r
# TEMPERATURE ARIMAX RESULTS
W1 <- merge(W1, AUTO_TMP_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, AUTO_TMP_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```r
W1 <- merge(W1, ES27_TMP_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES27_TMP_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
W1 <- merge(W1, ES64_TMP_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES64_TMP_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
# EPIWEEK ARIMAX RESULTS
W1 <- merge(W1, AUTO_EPI_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, AUTO_EPI_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```r
W1 <- merge(W1, ES27_EPI_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES27_EPI_W1, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

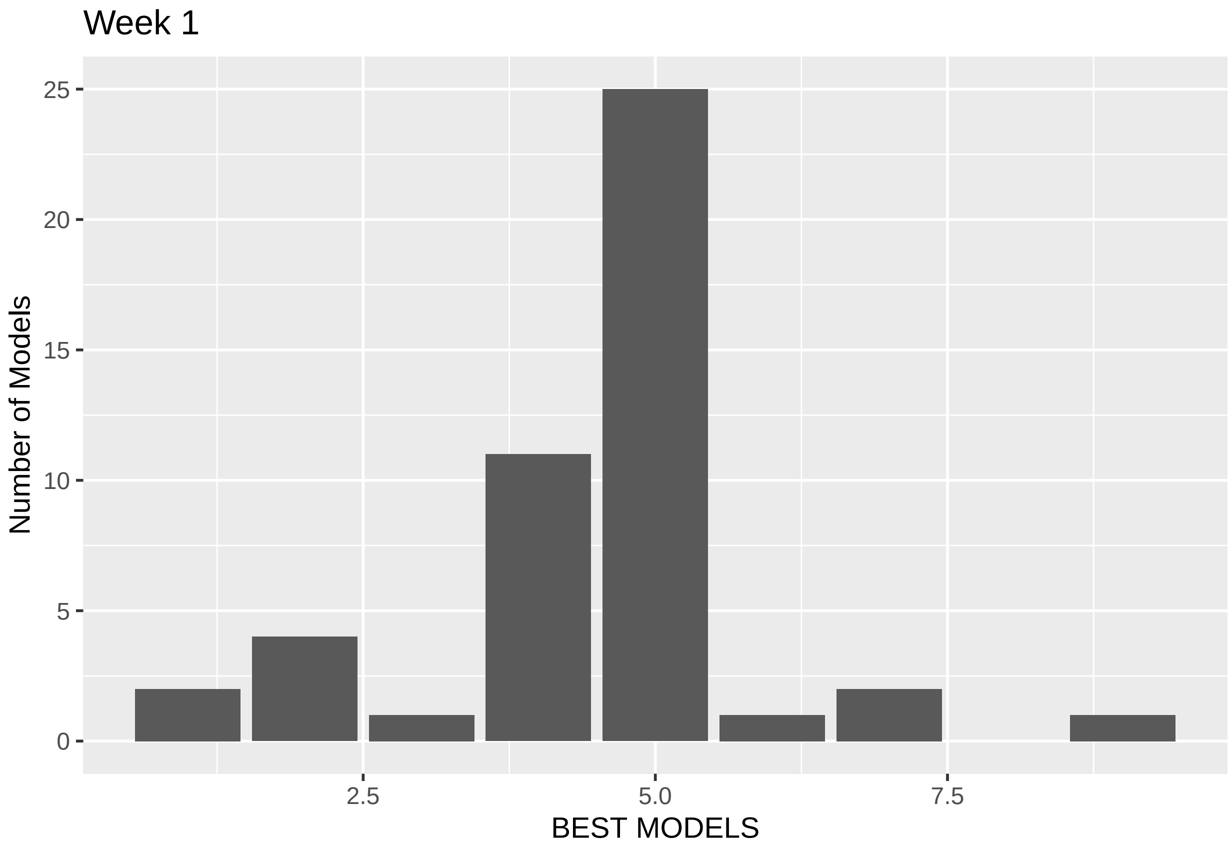
```
W1 <- merge(W1, ES64_EPI_W1, by = "State")
```

```
## Warning in merge.data.frame(W1, ES64_EPI_W1, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y' are duplicated in the result
```

```
# Rename columns for clarity
colnames(W1)[1] <- "NAME"
colnames(W1)[2] <- "AUTO_AR"
colnames(W1)[3] <- "ES27_AR"
colnames(W1)[4] <- "ES64_AR"
colnames(W1)[5] <- "AUTO_ADJ"
colnames(W1)[6] <- "ES27_ADJ"
colnames(W1)[7] <- "ES64_ADJ"
colnames(W1)[8] <- "AUTO_TMP"
colnames(W1)[9] <- "ES27_TMP"
colnames(W1)[10] <- "ES64_TMP"
colnames(W1)[11] <- "AUTO_EPI"
colnames(W1)[12] <- "ES27_EPI"
colnames(W1)[13] <- "ES64_EPI"

# Identify the best result for each state
W1$Best_Result <- apply(W1[,2:13], 1, function(x) {
  which.min(x)
})

#W1$Best_Result <- apply(W1[, 2:13], 1, function(x) {
#  colnames(W1)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W1)
```

```
##              NAME   AUTO_AR   ES27_AR    ES64_AR   AUTO_ADJ   ES27_ADJ
## 1         Alabama 151.48877 149.42716 152.383947 147.651281 146.444946
## 2         Arizona  68.20937  68.58018  70.665752  66.735532  64.460345
## 3        Arkansas  35.44634  34.41291  35.160342  36.192962  35.896407
## 4      California 228.97628 225.58583 222.918813 217.808666 212.272706
## 5        Colorado  94.47620  93.62881  94.257856  88.162698  86.533939
## 6     Connecticut  71.88757  66.11095  71.952815  69.460141  65.247560
## 7        Delaware  11.60498  11.96285  12.188690  10.535077  10.493166
## 8         Georgia 246.73295 240.39612 240.521285 246.242198 242.828513
## 9           Idaho  19.02588  18.72373  18.967321  17.918682  17.469907
## 10       Illinois 143.80910 133.66594 134.825128 137.128628 133.907175
## 11        Indiana  59.74954  58.97870  59.404656  55.645537  55.476754
## 12           Iowa  16.49067  17.16090  17.607296  14.553317  15.460221
## 13         Kansas  42.88743  42.61874  44.675746  34.576508  34.906137
## 14       Kentucky  72.60707  74.14009  75.155967  76.862600  74.371206
## 15      Louisiana 198.63080 189.96786 192.694877 182.200387 181.124556
## 16          Maine  26.01580  24.28679  24.736040  25.102301  23.860411
## 17       Maryland  81.53018  89.41175  87.261039  81.738998  78.696999
## 18  Massachusetts 105.34787 100.48120 102.960165  97.994551  97.701997
## 19       Michigan  87.45293  87.29822  88.338672  84.402581  82.754779
## 20      Minnesota  37.49001  37.49393  37.979709  38.273603  38.696656
## 21    Mississippi  87.43033  84.56155  84.828823  84.317868  83.571724
## 22       Missouri  41.38787  40.64536  41.627378  39.901780  38.630343
## 23        Montana  11.04965  10.96269  11.003570   9.674652   9.775759
## 24       Nebraska  39.21303  39.09292  39.641018  36.400633  36.536012
## 25         Nevada  43.94306  44.86251  48.149989  44.349545  45.017648
## 26  New Hampshire   7.92355   7.95180   7.948873   7.397984   7.119044
## 27     New Jersey  86.42591  82.99648  82.549504  80.498104  79.790709
## 28     New Mexico  81.90426  83.06968  84.357313  79.109073  78.576295
## 29       New York 118.92522 119.68811 121.573799 109.104523 109.573940
## 30 North Carolina 180.09363 182.22364 182.285945 186.184551 181.486815
## 31   North Dakota  13.90309  14.25661  14.617651  13.908457  13.451147
## 32           Ohio  65.08385  65.34843  65.365265  62.816164  59.893082
## 33       Oklahoma  60.23788  59.94219  59.762113  58.187869  57.100971
## 34         Oregon  64.23794  65.95026  65.213075  60.951532  63.355207
## 35   Pennsylvania 107.00028 102.53822 108.421945  88.523723  88.241999
## 36   Rhode Island  21.94064  22.95167  22.899272  20.097871  20.487126
## 37 South Carolina  88.11963  85.84728  85.536843  80.773106  80.353706
## 38   South Dakota  22.82589  22.64221  22.309102  22.290572  22.383820
## 39      Tennessee 120.78009 111.16335 108.608484 107.805772 108.312825
## 40          Texas 230.63096 229.64147 225.646325 230.417742 227.931609
## 41           Utah  86.29350  83.83603  85.428730  68.395223  71.105992
## 42        Vermont  12.44045  12.50044  12.339536  11.323872  11.472441
## 43       Virginia 451.96480 451.40164 454.866660 415.960638 403.541938
## 44     Washington  57.83461  56.97027  56.248448  60.525332  60.082336
## 45  West Virginia  48.07968  47.55352  47.878003  50.299278  48.621005
## 46      Wisconsin  34.99383  35.05032  35.738055  34.759349  34.137617
## 47        Wyoming  16.92944  16.56148  17.301359  15.784391  15.736752
##     ES64_ADJ   AUTO_TMP   ES27_TMP   ES64_TMP   AUTO_EPI   ES27_EPI   ES64_EPI
## 1  156.05074 151.554699 153.964880 155.742227 159.842265 155.569982 154.133581
## 2   68.15471  68.751788  68.211223  69.367157  71.454359  69.886860  71.091252
## 3   35.99479  36.127132  34.820892  36.607513  36.828082  36.779750  37.217507
## 4  241.51981 232.109357 227.435419 225.151000 241.480913 234.045331 235.698514
## 5   97.84971  94.954796  94.585896  94.227869  99.510423  98.555036  98.905187
## 6   74.54275  70.824713  71.213227  78.779686  75.560428  71.915604  77.157591
## 7   10.88305  12.298118  11.830231  11.967777  12.539691  12.793252  12.753110
## 8  255.92023 250.082913 244.472944 244.689515 254.648832 250.047501 249.959593
## 9   18.85276  19.524167  18.908083  19.107265  19.442538  19.169310  19.275902
## 10 143.11554 142.627862 137.872063 136.972068 149.581879 144.340799 143.284846
## 11  60.04154  61.596148  58.963653  59.588965  63.071670  61.613031  62.126778
## 12  15.62799  17.870156  17.619892  18.029748  17.435198  17.057329  17.520937
## 13  36.00393  43.068578  42.323321  44.222501  45.231571  45.804443  46.920248
## 14  74.84398  71.591475  75.187566  76.004339  80.723289  79.600050  80.142504
## 15 210.95920 196.850584 192.538740 191.983014 210.178051 207.453202 208.285692
## 16  24.68699  25.880230  24.853310  25.365208  25.920903  25.021724  25.543507
## 17  82.10637  80.418735  83.673190  82.113067  85.343424  85.183891  84.351789
## 18 110.65533 105.050558 103.993560 105.441994 112.226093 109.148716 109.721919
## 19  91.21918  90.876659  89.278680  90.751122  98.072940  95.093332  95.528486
## 20  38.39363  39.154860  39.121375  39.679666  39.207649  38.293854  38.727069
## 21  90.89400  88.680871  87.333485  87.390368  93.912779  93.172482  92.715736
## 22  41.40511  41.419749  41.033642  42.315883  47.413206  45.381409  47.114907
## 23  11.01958  11.454801  11.557989  11.645431  11.400777  11.218611  11.241321
## 24  39.27076  40.844187  39.741640  39.858654  41.667764  41.534241  42.346521
## 25  48.90829  47.556601  46.727898  48.789650  45.916779  48.825344  51.647962
## 26   7.58504   8.060808   8.104347   8.189044   8.884114   8.482133   8.514564
## 27  87.99266  85.799540  83.611982  83.619954  86.848395  85.422996  86.335576
## 28  85.23837  85.287491  84.773225  86.660823  95.541213  92.860266  93.087863
## 29 115.54746 119.630323 120.005718 121.473110 119.922829 119.468775 120.823151
## 30 177.50866 177.399485 180.860788 184.416410 191.449740 196.391973 196.902750
## 31  14.12184  13.705814  14.133173  14.628627  13.896330  14.101480  14.219362
## 32  69.13029  65.861248  66.214316  67.052083  72.171395  71.056354  70.756703
## 33  53.04152  58.276849  57.433731  59.599725  64.722717  64.099196  65.019299
## 34  65.39139  65.874613  67.641587  67.338821  64.049362  65.321715  64.756321
## 35 102.37346 106.908196 102.610614 110.806319 120.151110 115.851486 118.153941
## 36  22.07084  22.611755  22.673049  22.971141  25.112933  25.198130  25.657410
## 37  84.68774  90.586985  87.887293  88.197416  90.484271  89.841303  90.253577
## 38  22.76260  23.619146  23.020454  23.284559  24.304764  23.730851  23.983976
## 39 110.28748 110.595903 109.321674 109.464422 112.885874 114.086189 114.551656
## 40 233.95080 227.446808 230.278879 224.706636 246.111775 240.242001 238.421963
## 41  72.55443  90.192426  90.387863  90.997715  85.391545  83.624770  86.188380
## 42  12.09941  12.009657  12.478781  12.272307  12.967818  12.981007  12.995509
## 43 417.91937 465.497059 454.023707 457.000358 478.008419 475.828519 477.733163
## 44  63.53886  58.270767  56.639002  56.731142  60.351139  58.876870  58.853029
## 45  48.30621  48.272700  48.812009  48.626859  51.478362  49.927350  50.371975
## 46  35.98670  35.707289  35.253873  35.751896  36.497054  36.166582  36.468706
## 47  17.17775  16.908621  17.027651  17.674745  17.740754  17.496189  17.825901
##    Best_Result
## 1            5
## 2            5
## 3            2
## 4            5
## 5            5
## 6            5
## 7            5
## 8            2
## 9            5
## 10           2
## 11           5
## 12           4
## 13           4
## 14           7
## 15           5
## 16           5
## 17           5
## 18           5
## 19           5
## 20           1
## 21           5
## 22           5
## 23           4
## 24           4
## 25           1
## 26           5
## 27           5
## 28           5
## 29           4
## 30           7
## 31           5
## 32           5
## 33           6
## 34           4
## 35           5
## 36           4
## 37           5
## 38           4
## 39           4
## 40           9
## 41           4
## 42           4
## 43           5
## 44           3
## 45           2
## 46           5
## 47           5
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W1,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```



COMPARING ABSOLUTE ERRORS RESULTS FOR WEEK2

```r
W2<-NULL

# AUTO ARIMA RESULTS
AUTO_AR_W2 <- calculate_mean_abs_error(AUTO_ARIMA_WEEK2_list)
ES27_AR_W2 <- calculate_mean_abs_error(ES27_ARIMA_WEEK2_list)
ES64_AR_W2 <- calculate_mean_abs_error(ES64_ARIMA_WEEK2_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W2 <- calculate_mean_abs_error(AUTO_ADJACENT_WEEK2_list)
ES27_ADJ_W2 <- calculate_mean_abs_error(ES27_ADJACENT_WEEK2_list)
ES64_ADJ_W2 <- calculate_mean_abs_error(ES64_ADJACENT_WEEK2_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W2 <- calculate_mean_abs_error(AUTO_TEMPERATURE_WEEK2_list)
ES27_TMP_W2 <- calculate_mean_abs_error(ES27_TEMPERATURE_WEEK2_list)
ES64_TMP_W2 <- calculate_mean_abs_error(ES64_TEMPERATURE_WEEK2_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W2 <- calculate_mean_abs_error(AUTO_EPIWEEK_WEEK2_list)
ES27_EPI_W2 <- calculate_mean_abs_error(ES27_EPIWEEK_WEEK2_list)
ES64_EPI_W2 <- calculate_mean_abs_error(ES64_EPIWEEK_WEEK2_list)

# MERGE
# AUTO ARIMA RESULTS
W2 <- merge(AUTO_AR_W2, ES27_AR_W2, by = "State")
W2 <- merge(W2, ES64_AR_W2, by = "State")
# ADJACENT ARIMAX RESULTS
W2 <- merge(W2, AUTO_ADJ_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, AUTO_ADJ_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
W2 <- merge(W2, ES27_ADJ_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES27_ADJ_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
W2 <- merge(W2, ES64_ADJ_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES64_ADJ_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```r
# TEMPERATURE ARIMAX RESULTS
W2 <- merge(W2, AUTO_TMP_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, AUTO_TMP_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```r
W2 <- merge(W2, ES27_TMP_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES27_TMP_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
W2 <- merge(W2, ES64_TMP_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES64_TMP_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
# EPIWEEK ARIMAX RESULTS
W2 <- merge(W2, AUTO_EPI_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, AUTO_EPI_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```r
W2 <- merge(W2, ES27_EPI_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES27_EPI_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

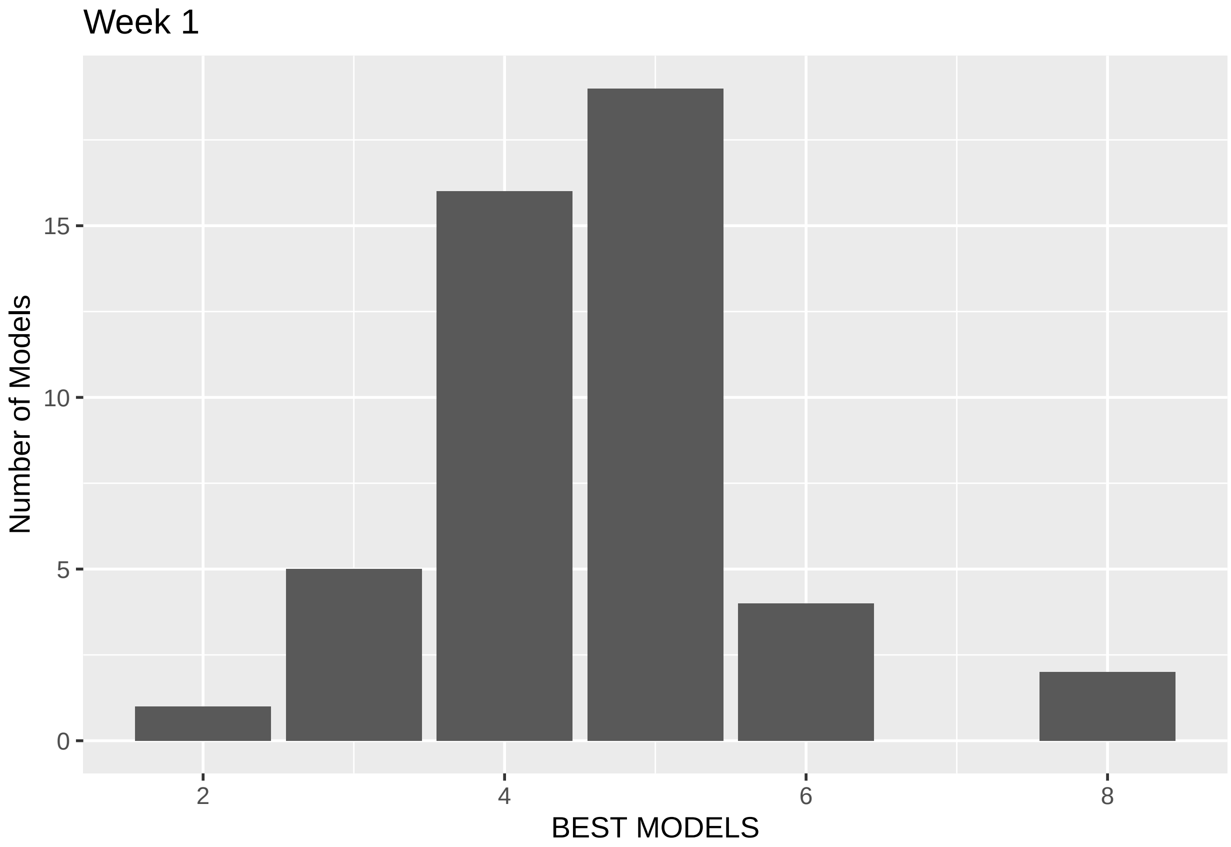
```r
W2 <- merge(W2, ES64_EPI_W2, by = "State")
```

```
## Warning in merge.data.frame(W2, ES64_EPI_W2, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```r
# Rename columns for clarity
colnames(W2)[1] <- "NAME"
colnames(W2)[2] <- "AUTO_AR"
colnames(W2)[3] <- "ES27_AR"
colnames(W2)[4] <- "ES64_AR"
colnames(W2)[5] <- "AUTO_ADJ"
colnames(W2)[6] <- "ES27_ADJ"
colnames(W2)[7] <- "ES64_ADJ"
colnames(W2)[8] <- "AUTO_TMP"
colnames(W2)[9] <- "ES27_TMP"
colnames(W2)[10] <- "ES64_TMP"
colnames(W2)[11] <- "AUTO_EPI"
colnames(W2)[12] <- "ES27_EPI"
colnames(W2)[13] <- "ES64_EPI"

# Identify the best result for each state
W2$Best_Result <- apply(W2[,2:13], 1, function(x) {
  which.min(x)
})

#W2$Best_Result <- apply(W2[, 2:13], 1, function(x) {
#  colnames(W2)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W2)
```

```
##              NAME   AUTO_AR   ES27_AR   ES64_AR   AUTO_ADJ   ES27_ADJ
## 1         Alabama 259.95188 255.93699 255.84264 247.050658 248.042752
## 2         Arizona  98.70302  97.97497  97.79662  98.578121  92.475798
## 3        Arkansas  53.01140  51.51257  52.14332  51.637597  52.937068
## 4      California 385.61969 374.00944 367.42927 376.247269 364.716724
## 5        Colorado 145.91546 144.24394 145.26594 138.441364 140.647640
## 6     Connecticut 116.02944 108.05691 114.50148 107.899466 104.663781
## 7        Delaware  15.78923  16.40436  16.56874  14.630552  14.021795
## 8         Georgia 439.32015 429.31946 428.12131 429.095935 426.980985
## 9           Idaho  26.48769  25.80980  26.00380  25.083553  24.005484
## 10       Illinois 252.73273 227.90800 224.00439 240.218191 229.124498
## 11        Indiana  97.10810  94.17416  95.21237  89.829264  92.205621
## 12           Iowa  22.15807  23.48895  24.33299  19.679257  21.378862
## 13         Kansas  65.71250  66.19775  68.04593  56.890852  56.616284
## 14       Kentucky 115.56431 117.47302 116.50350 114.096608 113.301073
## 15      Louisiana 338.93964 315.02751 318.33746 317.622229 311.286259
## 16          Maine  37.88056  35.68618  36.65540  37.381706  34.509644
## 17       Maryland 131.96340 178.74545 144.71921 129.482666 124.196531
## 18  Massachusetts 191.59261 175.88027 181.20202 170.719185 169.333378
## 19       Michigan 140.39911 142.17620 142.18628 132.261576 133.884842
## 20      Minnesota  58.25772  56.38945  56.33904  57.736767  58.500553
## 21    Mississippi 129.01553 126.62823 124.79093 127.307337 127.425516
## 22       Missouri  62.07609  61.34101  62.97545  57.218930  56.213100
## 23        Montana  16.39781  16.17731  16.29169  14.671172  14.789120
## 24       Nebraska  58.42699  57.45675  57.47848  55.216803  55.171256
## 25         Nevada  62.60169  61.73541  67.09255  59.908806  61.620590
## 26  New Hampshire  10.63736  10.72522  10.74985   9.865231   9.516575
## 27     New Jersey 146.75987 142.02233 141.53794 139.534906 138.836592
## 28     New Mexico 136.59758 137.20341 138.32347 124.138667 125.645690
## 29       New York 198.37581 199.32772 204.20673 178.747051 180.038076
## 30 North Carolina 309.80103 314.15189 306.26060 299.470493 295.623308
## 31   North Dakota  17.97207  18.30228  18.63794  17.143477  16.775670
## 32           Ohio 106.25629 104.44650 103.87243 104.406819  99.221739
## 33       Oklahoma  82.90667  80.86763  79.63449  75.623957  75.858222
## 34         Oregon 101.26964 101.73241 102.16428  92.667275  96.346454
## 35   Pennsylvania 168.73254 156.46053 164.72014 143.646750 143.461919
## 36   Rhode Island  32.54797  34.50458  34.52094  30.530971  31.160976
## 37 South Carolina 128.69849 125.90976 124.34348 119.326637 124.197765
## 38   South Dakota  30.00180  29.52711  29.23223  29.686161  29.128481
## 39      Tennessee 192.51512 173.06751 170.74362 172.526607 173.517846
## 40          Texas 325.87563 327.34910 316.34497 338.162925 333.176126
## 41           Utah 112.74406 108.80582 109.97258  92.074907  94.114828
## 42        Vermont  16.71125  16.68321  16.55090  15.294730  15.359645
## 43       Virginia 675.41641 670.01454 672.44241 655.382697 643.355465
## 44     Washington  96.70699  93.97757  93.72226  99.338703  97.612328
## 45  West Virginia  71.66397  68.34277  68.02137  70.945315  69.071926
## 46      Wisconsin  52.36697  51.99032  52.13119  49.895376  50.219099
## 47        Wyoming  24.30896  22.82937  23.92288  21.343818  21.599883
##      ES64_ADJ  AUTO_TMP  ES27_TMP  ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI
## 1  249.002754 259.11645 257.47737 261.37962 264.56821 258.64384 251.64946
## 2   98.623028  98.86230  97.35252  97.13957 101.40979  99.24293  99.04786
## 3   54.237682  53.17555  51.86450  52.24410  54.17214  54.94260  54.99683
## 4  392.621157 390.82615 379.63847 370.41280 399.18323 381.38183 382.15602
## 5  153.453939 146.64432 145.24397 145.58098 153.60058 150.62242 149.72891
## 6  118.403110 114.37558 115.28776 125.94553 121.62552 112.89131 119.56790
## 7   14.653683  16.38050  16.22120  16.23003  16.63432  16.78869  16.71407
## 8  437.648057 448.45662 436.94774 434.65691 455.32111 442.59599 441.82929
## 9   25.756609  26.85180  25.86610  26.11819  26.60308  26.22811  26.25689
## 10 237.511235 248.00990 231.79268 225.29165 254.93686 239.52582 238.15554
## 11  96.159666  98.60473  94.11347  95.27369  97.63859  97.66360  97.70174
## 12  21.511428  24.71515  24.01868  24.64382  23.23946  22.97042  23.93891
## 13  54.508001  66.53447  65.99048  66.81112  67.10018  67.73723  67.40915
## 14 113.312029 114.18705 119.60211 118.68933 124.23367 122.13441 122.47793
## 15 341.931768 333.19151 318.99879 320.61275 341.95919 332.64295 332.58572
## 16  36.477481  37.33407  36.43024  36.82972  37.33744  35.51713  36.07755
## 17 123.957167 128.68793 162.47197 136.65732 130.15794 142.32217 135.12738
## 18 181.208107 181.78298 175.57648 181.62410 191.23672 183.58398 187.02916
## 19 146.270313 145.49489 142.04022 143.24084 154.59035 149.80943 151.24913
## 20  57.044440  57.77008  57.15286  57.44221  58.71414  57.75752  57.71419
## 21 130.962952 128.36918 127.89491 125.95347 133.60151 134.21415 133.68979
## 22  62.255832  63.18954  61.76622  62.61403  67.88726  65.85800  68.44885
## 23  16.109157  16.03755  16.31948  16.48861  16.53234  16.33616  16.38716
## 24  58.095061  58.34304  56.74854  56.88791  59.52773  60.00797  60.52347
## 25  66.870295  66.23876  61.99576  65.52418  64.53300  65.80975  67.97120
## 26   9.937941  10.44759  10.68022  10.73327  11.12972  10.90263  10.86020
## 27 147.272194 145.70541 143.84094 143.78772 144.53016 143.07489 144.47173
## 28 137.194050 139.30819 139.45911 140.89315 148.62980 145.45683 145.79212
## 29 185.259740 190.32099 196.41075 198.35459 193.65812 194.43378 196.24651
## 30 292.038301 300.49360 312.03374 309.63088 306.32015 308.54502 306.03570
## 31  17.637916  17.43484  17.75734  18.32141  17.65527  18.06010  18.36064
## 32 109.678813 106.20665 105.64695 106.01242 112.66502 111.49956 110.23682
## 33  73.612493  81.63494  79.48679  81.50411  83.26782  82.50242  82.51122
## 34 100.849921 101.08477 102.96042 102.56092  99.32161  97.42961  97.60676
## 35 156.889849 164.87167 158.34194 168.26665 176.30631 167.94930 172.67569
## 36  32.829803  33.21811  33.35840  34.22456  36.14376  36.01758  36.78764
## 37 122.555550 130.99857 126.27250 126.26471 125.34663 126.89020 125.69522
## 38  29.981001  30.99939  29.52938  29.73967  30.18268  29.83301  30.34825
## 39 172.957953 172.70854 169.75158 170.78374 177.33823 176.20051 178.23792
## 40 337.773790 319.57961 328.27782 317.56749 336.82637 333.84600 330.85333
## 41  97.673971 107.17722 108.52997 110.05136 109.38332 106.26834 107.78035
## 42  16.189072  16.15551  16.56497  16.30801  17.22269  17.26932  17.23793
## 43 651.362473 691.54541 672.43708 669.97146 696.04846 694.24711 689.46261
## 44 100.930182  95.75863  93.42596  93.57054  99.33218  96.81204  97.28855
## 45  70.470901  69.97879  68.66840  68.23072  70.64109  70.23248  70.10474
## 46  52.721163  52.63002  52.23308  52.45461  52.33040  52.06532  52.26081
## 47  23.346513  23.11027  23.20768  23.96477  24.56120  23.64900  23.97633
##    Best_Result
## 1            4
## 2            5
## 3            2
## 4            5
## 5            4
## 6            5
## 7            5
## 8            5
## 9            5
## 10           3
## 11           4
## 12           4
## 13           6
## 14           5
## 15           5
## 16           5
## 17           6
## 18           5
## 19           4
## 20           3
## 21           3
## 22           5
## 23           4
## 24           5
## 25           4
## 26           5
## 27           5
## 28           4
## 29           4
## 30           6
## 31           5
## 32           5
## 33           6
## 34           4
## 35           5
## 36           4
## 37           4
## 38           5
## 39           8
## 40           3
## 41           4
## 42           4
## 43           5
## 44           8
## 45           3
## 46           4
## 47           4
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W2,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```



COMPARING ABSOLUTE ERRORS RESULTS FOR WEEK3

```
W3<-NULL

# AUTO ARIMA RESULTS
AUTO_AR_W3 <- calculate_mean_abs_error(AUTO_ARIMA_WEEK3_list)
ES27_AR_W3 <- calculate_mean_abs_error(ES27_ARIMA_WEEK3_list)
ES64_AR_W3 <- calculate_mean_abs_error(ES64_ARIMA_WEEK3_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W3 <- calculate_mean_abs_error(AUTO_ADJACENT_WEEK3_list)
ES27_ADJ_W3 <- calculate_mean_abs_error(ES27_ADJACENT_WEEK3_list)
ES64_ADJ_W3 <- calculate_mean_abs_error(ES64_ADJACENT_WEEK3_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W3 <- calculate_mean_abs_error(AUTO_TEMPERATURE_WEEK3_list)
ES27_TMP_W3 <- calculate_mean_abs_error(ES27_TEMPERATURE_WEEK3_list)
ES64_TMP_W3 <- calculate_mean_abs_error(ES64_TEMPERATURE_WEEK3_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W3 <- calculate_mean_abs_error(AUTO_EPIWEEK_WEEK3_list)
ES27_EPI_W3 <- calculate_mean_abs_error(ES27_EPIWEEK_WEEK3_list)
ES64_EPI_W3 <- calculate_mean_abs_error(ES64_EPIWEEK_WEEK3_list)

# MERGE
# AUTO ARIMA RESULTS
W3 <- merge(AUTO_AR_W3, ES27_AR_W3, by = "State")
W3 <- merge(W3, ES64_AR_W3, by = "State")
# ADJACENT ARIMAX RESULTS
```

```
W3 <- merge(W3, AUTO_ADJ_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, AUTO_ADJ_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```
W3 <- merge(W3, ES27_ADJ_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES27_ADJ_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```
W3 <- merge(W3, ES64_ADJ_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES64_ADJ_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```
# TEMPERATURE ARIMAX RESULTS
W3 <- merge(W3, AUTO_TMP_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, AUTO_TMP_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```
W3 <- merge(W3, ES27_TMP_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES27_TMP_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```
W3 <- merge(W3, ES64_TMP_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES64_TMP_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```
# EPIWEEK ARIMAX RESULTS
W3 <- merge(W3, AUTO_EPI_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, AUTO_EPI_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

```
W3 <- merge(W3, ES27_EPI_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES27_EPI_W3, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’
## are duplicated in the result
```

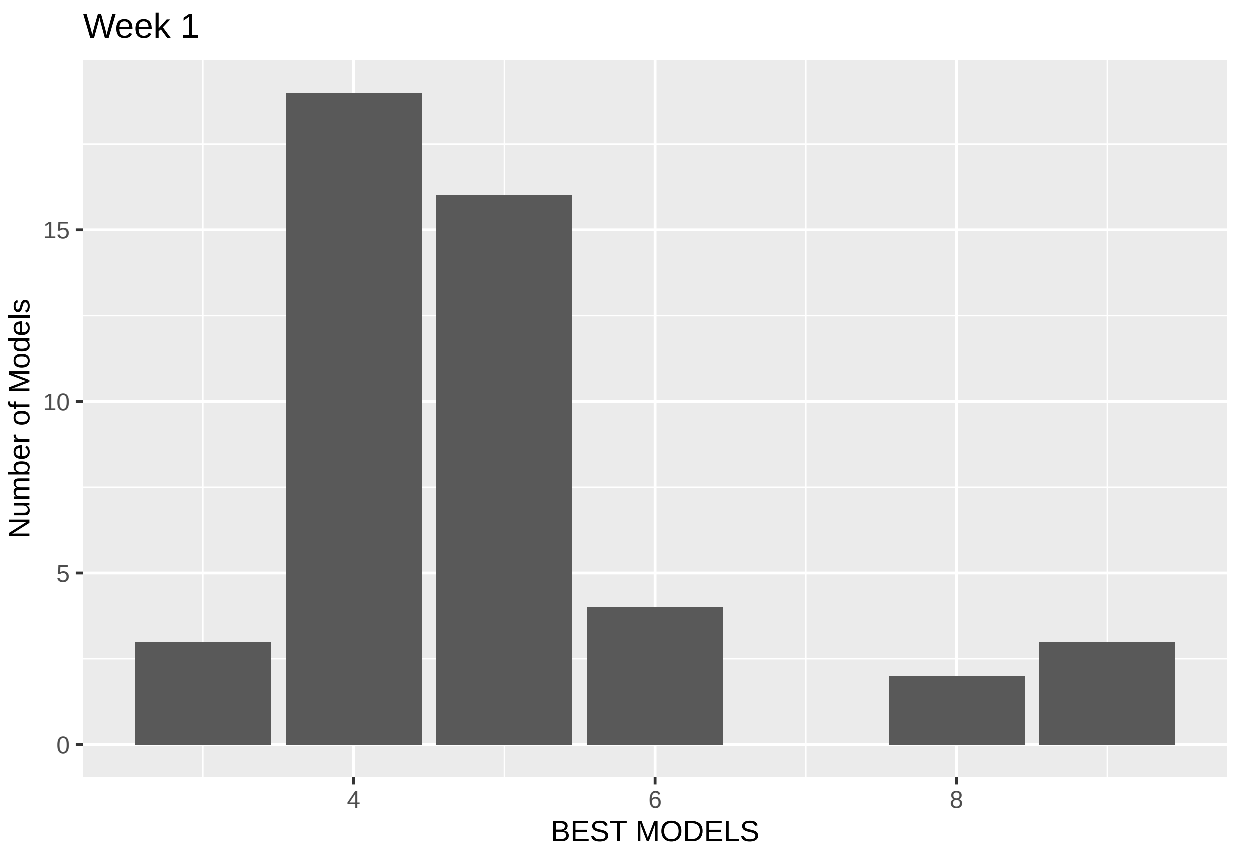
```
W3 <- merge(W3, ES64_EPI_W3, by = "State")
```

```
## Warning in merge.data.frame(W3, ES64_EPI_W3, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y' are duplicated in the result
```

```
# Rename columns for clarity
colnames(W3)[1] <- "NAME"
colnames(W3)[2] <- "AUTO_AR"
colnames(W3)[3] <- "ES27_AR"
colnames(W3)[4] <- "ES64_AR"
colnames(W3)[5] <- "AUTO_ADJ"
colnames(W3)[6] <- "ES27_ADJ"
colnames(W3)[7] <- "ES64_ADJ"
colnames(W3)[8] <- "AUTO_TMP"
colnames(W3)[9] <- "ES27_TMP"
colnames(W3)[10] <- "ES64_TMP"
colnames(W3)[11] <- "AUTO_EPI"
colnames(W3)[12] <- "ES27_EPI"
colnames(W3)[13] <- "ES64_EPI"

# Identify the best result for each state
invisible(W3$Best_Result <- apply(W3[,2:13], 1, function(x) {
  which.min(x)
}))

#W3$Best_Result <- apply(W3[, 2:13], 1, function(x) {
#  colnames(W3)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W3)
```

```
##               NAME   AUTO_AR   ES27_AR   ES64_AR  AUTO_ADJ  ES27_ADJ  ES64_ADJ
## 1         Alabama 349.86643 343.58328 339.56618 324.67647 328.30850 325.32192
## 2         Arizona 128.18072 126.89568 124.69638 130.07970 120.49111 128.97881
## 3        Arkansas  69.46577  67.94507  68.03305  63.75847  66.49926  68.83715
## 4      California 526.42763 508.95525 498.71229 522.33860 497.25797 540.45924
## 5        Colorado 199.89635 195.25998 197.18771 189.17136 192.16287 204.69856
## 6     Connecticut 155.36769 148.60913 152.94020 143.87305 137.14369 153.82955
## 7        Delaware  19.62929  20.62620  20.59739  19.08371  17.85508  18.82953
## 8         Georgia 588.56479 580.09854 578.60257 573.09918 581.04147 579.22188
## 9           Idaho  32.97925  32.25496  32.26099  31.61760  29.17377  31.79724
## 10       Illinois 345.43976 304.88189 298.87307 331.71250 307.28020 314.98102
## 11        Indiana 131.73791 127.23819 127.95668 120.60043 125.23814 126.96635
## 12           Iowa  27.00782  28.37321  29.27435  24.50684  26.31543  26.10615
## 13         Kansas  87.68507  92.22577  91.56717  79.45908  79.09791  74.97244
## 14       Kentucky 152.61159 155.37692 151.15687 148.64592 148.85975 147.49421
## 15      Louisiana 449.59250 417.16960 417.64380 428.97269 413.41165 441.19850
## 16          Maine  48.96734  46.57500  47.73617  48.57624  45.59359  47.56688
## 17       Maryland 170.10103 247.29693 177.31694 167.57221 154.07920 157.08903
## 18  Massachusetts 268.92954 247.70808 252.25184 237.53459 240.22209 240.61420
## 19       Michigan 190.00545 191.17862 190.66910 178.85351 179.62681 196.02759
## 20      Minnesota  75.99067  73.17699  72.56583  73.10831  74.94737  73.85138
## 21    Mississippi 166.17868 159.07843 156.73907 165.01251 164.98719 169.22985
## 22       Missouri  82.14503  82.26308  81.98037  75.90177  74.02377  81.38796
## 23        Montana  21.17954  21.01340  21.05051  18.90982  19.29267  20.37311
## 24       Nebraska  75.23252  73.54345  73.47647  71.02157  70.86932  74.20769
## 25         Nevada  79.18912  78.39789  79.73633  76.70987  77.52892  78.65571
## 26  New Hampshire  12.88144  12.92527  12.91214  12.07036  11.80092  11.99465
## 27     New Jersey 198.86914 189.39034 189.15935 187.54032 186.56962 195.79063
## 28     New Mexico 185.87370 186.21645 186.74775 166.88262 170.25024 181.02225
## 29       New York 264.05837 267.51970 279.87730 233.50985 237.22857 235.21374
## 30 North Carolina 412.33793 427.77879 407.67981 394.68985 398.02974 386.00994
## 31   North Dakota  21.58925  21.71065  22.01690  20.56015  20.35450  20.96424
## 32           Ohio 143.10411 141.39731 139.64573 143.17641 135.63695 145.94264
## 33       Oklahoma 102.82606 100.15470  97.53448  90.24325  91.71550  90.36981
## 34         Oregon 136.73117 134.98155 134.25791 124.73666 129.30212 134.33436
## 35   Pennsylvania 226.38558 211.34923 212.16081 193.02373 192.16660 205.12963
## 36   Rhode Island  41.20994  44.37100  44.37837  38.93753  40.31407  40.35134
## 37 South Carolina 162.37321 157.48269 152.58719 153.02507 157.89399 156.68653
## 38   South Dakota  38.20512  36.95056  36.85851  36.85538  35.74951  37.36264
## 39      Tennessee 259.10988 218.03672 216.64734 220.73939 225.28399 218.97907
## 40          Texas 420.20419 418.20852 403.60697 425.57812 422.60201 422.72630
## 41           Utah 139.46419 132.39370 132.78438 116.93621 118.32384 120.83565
## 42        Vermont  20.80793  20.82116  20.36304  19.17732  19.38386  20.14313
## 43       Virginia 883.43517 867.60163 863.89876 857.65513 851.47517 835.42771
## 44     Washington 132.98583 129.67300 129.81110 133.09048 134.73786 136.14439
## 45  West Virginia  92.17484  86.40724  85.48431  88.27215  85.71685  90.09116
## 46      Wisconsin  67.35232  68.59389  68.62425  64.96773  65.95098  67.40550
## 47        Wyoming  29.89547  28.12043  28.79878  26.50398  26.88324  28.63994
##     AUTO_TMP  ES27_TMP  ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI Best_Result
## 1  346.87292 341.11117 342.49709 348.28448 341.03653 331.59245           4
## 2  126.46708 123.52562 121.73940 131.68381 129.57585 127.65571           5
## 3   68.54038  67.38075  64.39930  68.34895  69.98469  69.04957           4
## 4  537.89737 520.96678 507.58534 547.84048 527.84161 525.25188           5
## 5  199.51489 198.20897 198.80991 205.72948 201.63098 202.79090           4
## 6  151.85203 154.15569 164.93246 161.18818 150.61247 157.38529           5
## 7   20.43297  20.46833  20.34587  20.58203  20.93173  20.74206           5
## 8  595.51184 588.32208 583.04186 604.69279 587.40003 587.64600           4
## 9   33.38839  32.27199  32.31331  33.08189  32.37705  32.42156           5
## 10 337.09531 306.33489 298.11376 342.73306 322.66889 318.76852           9
## 11 133.71392 127.34806 128.47780 126.15116 131.49064 130.75106           4
## 12  30.57089  29.07819  30.06445  27.72734  27.49912  28.24637           4
## 13  88.63583  91.69597  89.81073  88.67915  91.13905  90.13107           6
## 14 151.92970 157.11651 153.13382 161.28653 158.90862 156.94202           6
## 15 445.80715 423.67533 423.21148 443.05840 436.58447 433.68472           5
## 16  47.34349  46.47093  47.14318  48.08426  46.28345  46.91891           5
## 17 168.04766 215.49604 166.86765 167.31962 179.76560 162.73676           5
## 18 250.05962 244.98444 251.74687 262.20822 250.07673 253.54939           4
## 19 197.75104 191.72468 194.37360 201.95824 197.05792 197.80417           4
## 20  74.24762  73.68286  73.44281  74.75601  74.60710  73.95516           3
## 21 163.54992 159.82338 155.30061 171.71590 171.35667 170.98417           9
## 22  82.91893  81.60403  81.34048  87.57742  83.87170  84.91460           5
## 23  20.41295  20.86625  21.08651  21.19155  20.91788  21.00243           4
## 24  74.30856  72.82466  72.96144  76.54012  75.66470  76.27989           5
## 25  82.43833  78.05121  79.10359  82.10147  80.69805  79.25882           4
## 26  12.39326  12.75387  12.73763  13.00157  13.01424  12.92533           5
## 27 196.77207 191.35960 191.12662 190.97461 187.75641 190.55838           5
## 28 189.01837 190.27247 191.91862 198.00858 193.40555 193.69640           4
## 29 244.88274 262.14709 267.10171 246.25887 246.86467 248.85697           4
## 30 400.64042 429.51956 413.39160 402.63892 406.93303 402.22478           6
## 31  21.05104  21.22048  21.79259  21.21712  21.83048  22.12691           5
## 32 142.37849 141.73318 141.32529 151.05730 146.90898 145.43451           5
## 33 100.08950  98.44570  97.59801 102.55576 101.42667 102.13806           4
## 34 132.06509 134.89126 132.81726 134.78263 131.22457 129.90365           4
## 35 220.30523 210.45010 218.34394 227.23393 216.24528 221.52387           5
## 36  41.79088  42.28933  43.64550  45.30036  44.75010  45.69772           4
## 37 165.42206 156.76387 154.70622 157.11666 158.25732 155.45795           3
## 38  38.01054  36.46227  37.67880  39.80795  38.70339  39.48663           5
## 39 220.64242 216.19329 218.01821 227.09794 222.25309 225.89737           8
## 40 413.01415 416.40755 406.55458 426.74321 420.55683 412.13963           3
## 41 126.80090 128.45277 130.20619 133.41463 130.71821 131.05985           4
## 42  20.05930  20.48652  19.96318  21.29586  21.36242  21.32670           4
## 43 917.86604 876.06043 864.47187 899.55980 902.21140 890.97061           6
## 44 133.25323 128.33405 130.03726 133.87995 131.40018 132.69235           8
## 45  88.87630  86.82729  85.21111  90.16150  88.39589  88.94463           9
## 46  68.21816  68.97991  68.96298  67.39622  67.93673  68.00616           4
## 47  28.50496  28.79942  29.22325  30.55575  29.14468  29.20414           4
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W3,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```



COMPARING ABSOLUTE ERRORS RESULTS FOR WEEK4

```r
W4<-NULL

# AUTO ARIMA RESULTS
AUTO_AR_W4 <- calculate_mean_abs_error(AUTO_ARIMA_WEEK4_list)
ES27_AR_W4 <- calculate_mean_abs_error(ES27_ARIMA_WEEK4_list)
ES64_AR_W4 <- calculate_mean_abs_error(ES64_ARIMA_WEEK4_list)
# ADJACENT ARIMAX RESULTS
AUTO_ADJ_W4 <- calculate_mean_abs_error(AUTO_ADJACENT_WEEK4_list)
ES27_ADJ_W4 <- calculate_mean_abs_error(ES27_ADJACENT_WEEK4_list)
ES64_ADJ_W4 <- calculate_mean_abs_error(ES64_ADJACENT_WEEK4_list)
# TEMPERATURE ARIMAX RESULTS
AUTO_TMP_W4 <- calculate_mean_abs_error(AUTO_TEMPERATURE_WEEK4_list)
ES27_TMP_W4 <- calculate_mean_abs_error(ES27_TEMPERATURE_WEEK4_list)
ES64_TMP_W4 <- calculate_mean_abs_error(ES64_TEMPERATURE_WEEK4_list)
# EPIWEEK ARIMAX RESULTS
AUTO_EPI_W4 <- calculate_mean_abs_error(AUTO_EPIWEEK_WEEK4_list)
ES27_EPI_W4 <- calculate_mean_abs_error(ES27_EPIWEEK_WEEK4_list)
ES64_EPI_W4 <- calculate_mean_abs_error(ES64_EPIWEEK_WEEK4_list)

# MERGE
# AUTO ARIMA RESULTS
W4 <- merge(AUTO_AR_W4, ES27_AR_W4, by = "State")
W4 <- merge(W4, ES64_AR_W4, by = "State")
# ADJACENT ARIMAX RESULTS
```

```
W4 <- merge(W4, AUTO_ADJ_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, AUTO_ADJ_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y' are duplicated in the result
```

```
W4 <- merge(W4, ES27_ADJ_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES27_ADJ_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y' are duplicated in the result
```

```
W4 <- merge(W4, ES64_ADJ_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES64_ADJ_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y'
## are duplicated in the result
```

```
# TEMPERATURE ARIMAX RESULTS
W4 <- merge(W4, AUTO_TMP_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, AUTO_TMP_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y'
## are duplicated in the result
```

```
W4 <- merge(W4, ES27_TMP_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES27_TMP_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y' are duplicated in the result
```

```
W4 <- merge(W4, ES64_TMP_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES64_TMP_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y' are duplicated in the result
```

```
# EPIWEEK ARIMAX RESULTS
W4 <- merge(W4, AUTO_EPI_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, AUTO_EPI_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y'
## are duplicated in the result
```

```
W4 <- merge(W4, ES27_EPI_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES27_EPI_W4, by = "State"): column names
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y',
## 'Mean_abs_error.x', 'Mean_abs_error.y', 'Mean_abs_error.x', 'Mean_abs_error.y'
## are duplicated in the result
```

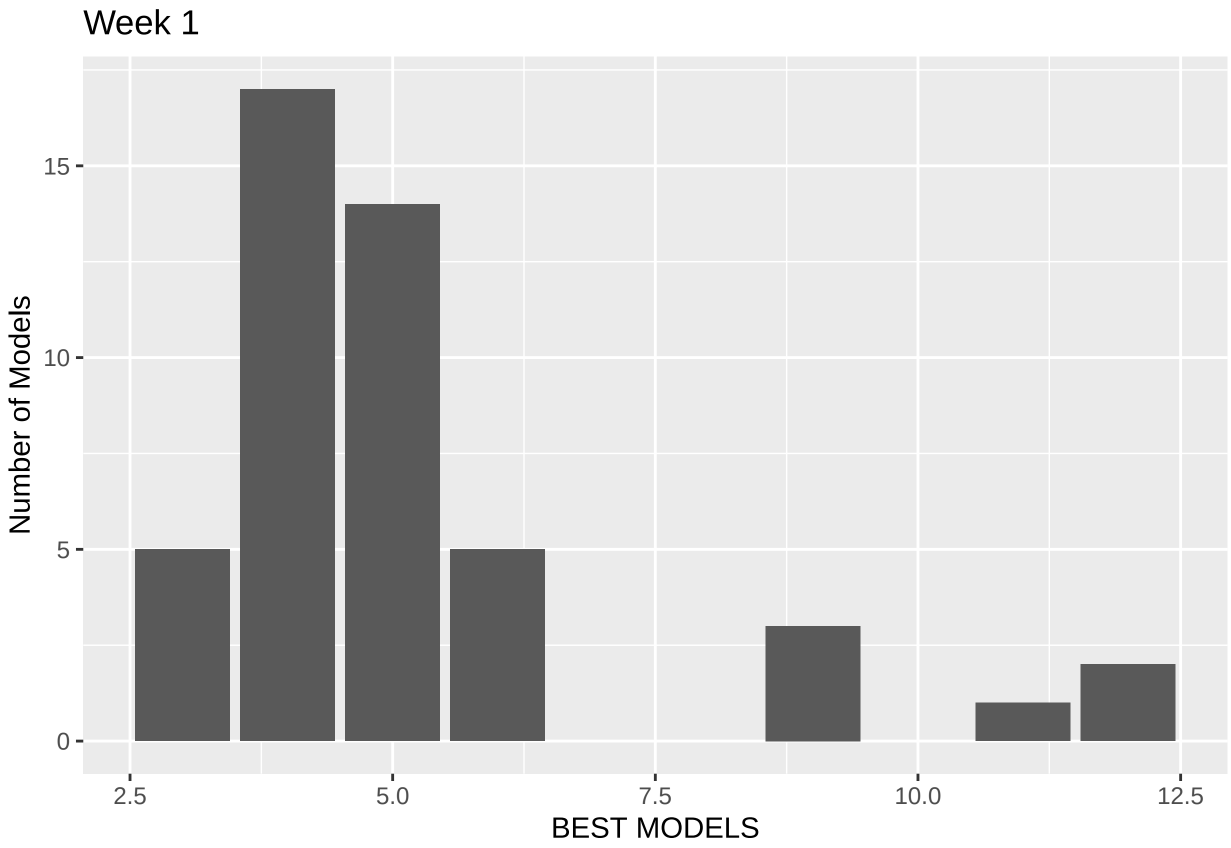
```
W4 <- merge(W4, ES64_EPI_W4, by = "State")
```

```
## Warning in merge.data.frame(W4, ES64_EPI_W4, by = "State"): column names
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’, ’Mean_abs_error.x’, ’Mean_abs_error.y’,
## ’Mean_abs_error.x’, ’Mean_abs_error.y’ are duplicated in the result
```

```
# Rename columns for clarity
colnames(W4)[1] <- "NAME"
colnames(W4)[2] <- "AUTO_AR"
colnames(W4)[3] <- "ES27_AR"
colnames(W4)[4] <- "ES64_AR"
colnames(W4)[5] <- "AUTO_ADJ"
colnames(W4)[6] <- "ES27_ADJ"
colnames(W4)[7] <- "ES64_ADJ"
colnames(W4)[8] <- "AUTO_TMP"
colnames(W4)[9] <- "ES27_TMP"
colnames(W4)[10] <- "ES64_TMP"
colnames(W4)[11] <- "AUTO_EPI"
colnames(W4)[12] <- "ES27_EPI"
colnames(W4)[13] <- "ES64_EPI"

# Identify the best result for each state
W4$Best_Result <- apply(W4[,2:13], 1, function(x) {
  which.min(x)
})

#W4$Best_Result <- apply(W4[, 2:13], 1, function(x) {
#  colnames(W4)[which.min(x) + 1]  # +1 to shift the index to account for column 1
#})

# Print merged results
print(W4)
```

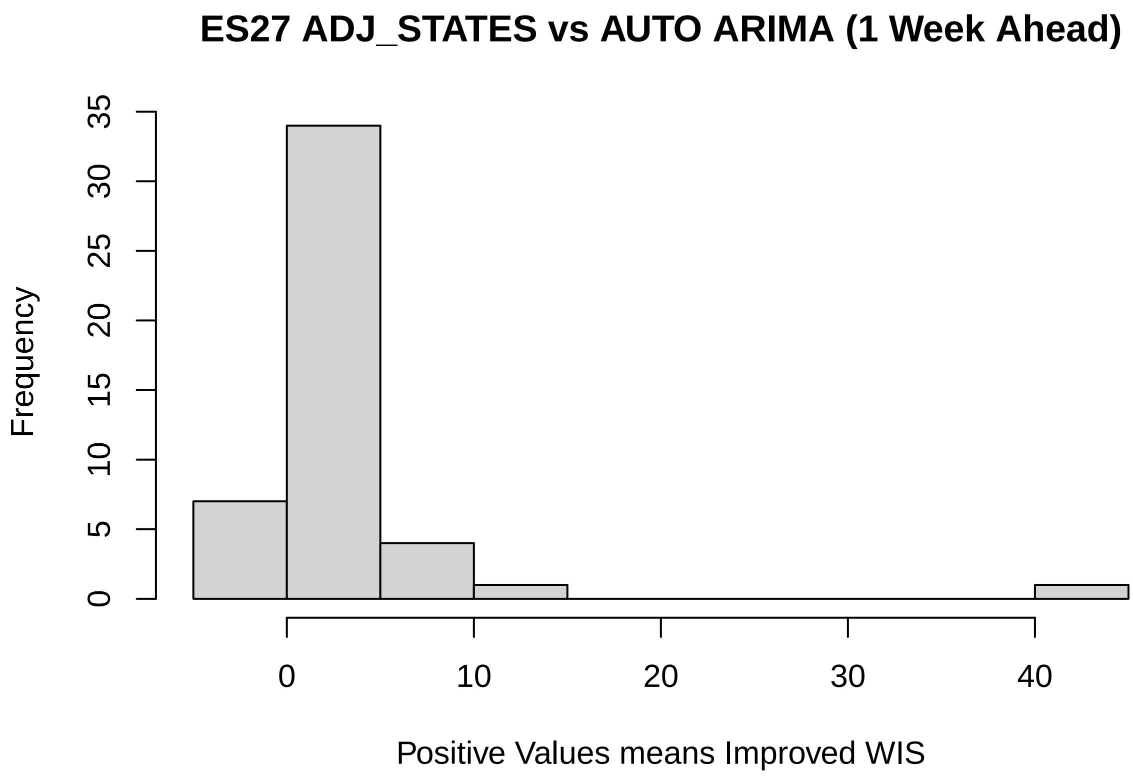
```
##              NAME   AUTO_AR   ES27_AR   ES64_AR  AUTO_ADJ  ES27_ADJ
## 1         Alabama 413.88230 400.73707 391.94808 381.68738 389.75197
## 2         Arizona 155.26097 156.40692 151.83746 161.36320 149.22714
## 3        Arkansas  80.97836  80.24648  80.04614  73.96920  78.18030
## 4      California 671.42215 647.58183 628.70287 661.55315 634.22341
## 5        Colorado 243.83827 239.83555 240.76861 228.04208 234.43555
## 6     Connecticut 187.72506 185.43976 188.80089 169.12149 168.93080
## 7        Delaware  22.75760  24.40142  24.18182  22.51529  21.56187
## 8         Georgia 693.93588 681.83455 677.83875 669.23492 683.32194
## 9           Idaho  38.26952  37.78159  37.33686  37.87708  34.74329
## 10       Illinois 433.59418 379.64568 369.09858 415.84350 383.96758
## 11        Indiana 163.48891 155.14640 156.41103 144.82661 154.19813
## 12           Iowa  31.89260  33.30811  33.84202  29.45967  31.55103
## 13         Kansas 104.95747 114.84086 109.97881  97.57734  99.43919
## 14       Kentucky 186.69831 186.03642 177.62240 176.70257 178.26851
## 15      Louisiana 548.14929 509.84455 502.71910 527.57655 506.60137
## 16          Maine  58.92535  56.72431  57.90613  58.20075  55.85245
## 17       Maryland 204.60864 662.02512 245.63180 201.91214 183.74678
## 18  Massachusetts  343.04392  305.52482  312.13744  297.42514  300.24189
## 19       Michigan  229.04571  232.68948  233.47725  215.67387  216.15167
## 20      Minnesota   92.52940   87.60910   86.71437   87.39236   90.08419
## 21    Mississippi  200.57256  189.10024  185.23326  200.29374  199.12711
## 22       Missouri   99.59544  100.61452   97.74458   93.47985   91.47680
## 23        Montana   25.41879   25.41640   25.34968   22.55043   23.24050
## 24       Nebraska   92.70202   90.69294   90.48193   84.74012   85.33616
## 25         Nevada   94.17776   93.56667   93.60375   93.71657   93.76908
## 26  New Hampshire   14.58181   14.92182   14.79091   14.11416   13.78266
## 27     New Jersey  244.45717  231.21269  231.68325  233.42525  229.11751
## 28     New Mexico  224.08834  228.82087  227.89548  205.55190  208.79192
## 29       New York  313.84753  343.13498  381.13673  269.73398  276.05852
## 30 North Carolina  499.58277  550.34040  496.10454  474.24555  480.97973
## 31   North Dakota   24.92641   25.06769   25.32756   23.43141   23.14464
## 32           Ohio  177.33711  175.03018  172.14396  180.20613  171.03029
## 33       Oklahoma  120.38985  116.65728  112.67193  106.12588  107.67038
## 34         Oregon  169.91182  167.18217  164.64936  153.28004  158.42844
## 35   Pennsylvania  275.17315  256.65776  252.57678  235.48050  233.81875
## 36   Rhode Island   48.77778   52.95832   51.94413   45.73934   47.27039
## 37 South Carolina  195.56388  188.72884  181.97388  181.18945  187.32105
## 38   South Dakota   45.10130   43.75578   42.96783   43.61832   42.02795
## 39      Tennessee  319.78391  257.66040  258.62385  253.12901  262.29573
## 40          Texas  497.26941  499.15394  477.86380  496.33242  489.52751
## 41           Utah  163.96537  156.43953  157.62974  141.20432  141.19689
## 42        Vermont   24.73010   24.84183   24.04280   22.86131   22.84963
## 43       Virginia 1036.42534 1022.54799 1012.54968 1027.06248 1030.24887
## 44     Washington  162.23477  159.29784  158.47996  161.60108  162.83122
## 45  West Virginia  110.16575  101.69800   98.80001  103.52748  103.04762
## 46      Wisconsin   82.16407   84.12614   83.98738   79.14300   79.86237
## 47        Wyoming   34.76514   33.00408   33.75744   32.38077   32.40521
##     ES64_ADJ  AUTO_TMP  ES27_TMP  ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI
## 1  384.17230 413.04489 398.38137 392.44665 415.48198 393.09770 380.18901
## 2  156.04356 154.24510 150.47835 146.61175 159.87617 157.80529 155.77767
## 3   82.10429  80.05437  80.17987  75.43289  80.27861  82.93429  80.82333
## 4  667.48554 679.05464 658.46479 639.28256 699.97186 671.25135 666.63152
## 5  249.25618 240.69508 242.08695 241.21535 249.88670 247.25072 248.26244
## 6  187.09644 187.08769 190.31553 199.38342 197.56367 183.53705 192.33282
## 7   22.16518  23.71981  24.01925  23.82780  24.00651  24.57367  24.21139
## 8  675.73221 694.65864 690.46422 681.26551 701.21221 682.51670 678.31963
## 9   37.25274  39.10174  37.75045  37.30113  39.00781  37.67113  37.42257
## 10 386.79205 419.15709 375.97313 363.26153 424.47730 393.27847 390.49988
## 11 152.44470 162.55873 155.22445 156.36272 154.45113 160.20852 158.77640
## 12  30.47339  35.17238  33.38360  34.35225  33.21403  32.48087  33.19940
## 13  92.46800 105.75971 114.71311 107.46039 107.90503 110.15820 107.79259
## 14 175.11737 184.20321 187.67602 178.06787 194.60741 190.25521 186.27601
## 15 534.95124 549.60078 516.23397 512.33407 536.63394 524.21269 517.78048
## 16  57.71700  58.18869  56.46779  57.52972  57.39589  56.21704  57.43653
## 17 190.38911 204.89388 481.80277 224.47091 203.70119 272.09340 210.70202
## 18 293.93051 316.90110 302.06865 311.40783 330.21106 310.47040 315.23405
## 19 233.46991 240.84071 232.95176 236.76978 241.93448 236.73328 238.39696
## 20  88.81562  89.42403  88.49239  87.49731  90.16750  89.31277  88.15147
## 21 201.70876 194.70369 191.49942 185.20667 205.65867 204.60975 204.56900
## 22  99.17870 100.53659 100.69948  98.35358 104.79882 100.40927 100.72794
## 23  24.46880  24.27044  25.01592  25.11801  25.42574  25.15300  25.21192
## 24   88.56609   90.99942   89.53446   89.69730   93.37952   91.90476   92.81261
## 25   93.57665   96.27072   93.53273   95.01154   97.74290   95.37375   93.48301
## 26   14.00803   13.91957   14.52616   14.35436   14.74547   14.77785   14.65378
## 27  235.86047  240.05557  232.71426  232.71215  233.06613  231.44672  233.88581
## 28  215.27228  228.25987  233.54403  233.02439  240.67783  234.77711  233.81658
## 29  281.57582  293.10189  326.49685  340.58533  289.36625  291.21944  294.14902
## 30  465.35987  484.05212  554.89515  503.07928  482.73876  489.75770  483.67024
## 31   23.52560   24.54480   24.69542   24.92576   24.99748   25.22572   25.41054
## 32  180.32240  175.54572  175.68853  173.56932  186.13328  182.31343  179.20471
## 33  107.55021  118.06578  116.70487  114.46940  122.12817  119.38092  119.62594
## 34  163.64495  162.62014  163.51369  161.07919  163.21589  161.20918  160.35651
## 35  246.22860  265.81190  253.50788  256.79845  275.10714  260.77131  266.17442
## 36   46.17584   48.76155   49.84385   50.82670   52.78954   52.14131   52.89257
## 37  184.26182  197.60706  186.63907  183.33535  184.32705  187.48719  183.02326
## 38   43.59576   45.42792   43.39862   44.34074   47.35514   45.67234   45.97002
## 39  253.24927  267.52949  256.62591  259.15589  266.13087  259.90383  261.46785
## 40  492.25888  498.24954  496.74710  483.47100  506.63731  501.45664  487.39464
## 41  144.58909  147.31250  149.48239  151.77176  156.50436  154.68673  155.93675
## 42   23.77525   23.95109   24.44691   23.56476   25.24374   25.38849   25.25784
## 43 1004.18345 1085.61522 1042.30934 1019.44232 1056.15890 1065.22139 1047.90290
## 44  162.57407  163.23647  158.70452  161.11721  160.76425  157.52292  160.04279
## 45  107.39994  106.89127  102.62669   99.21470  108.69837  105.69266  105.36667
## 46   81.53327   82.42174   84.24049   84.11227   81.13967   82.29600   82.61387
## 47   33.35251   33.90221   34.03242   34.34688   36.18882   34.41513   34.32894
##    Best_Result
## 1           12
## 2            9
## 3            4
## 4            3
## 5            4
## 6            5
## 7            5
## 8            4
## 9            5
## 10           9
## 11           4
## 12           4
## 13           6
## 14           6
## 15           3
## 16           5
## 17           5
## 18           6
## 19           4
## 20           3
## 21           9
## 22           5
## 23           4
## 24           4
## 25          12
## 26           5
## 27           5
## 28           4
## 29           4
## 30            6
## 31            5
## 32            5
## 33            4
## 34            4
## 35            5
## 36            4
## 37            4
## 38            5
## 39            4
## 40            3
## 41            5
## 42            5
## 43            6
## 44           11
## 45            3
## 46            4
## 47            4
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W4,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Week 1",
       x = "BEST MODELS", y="Number of Models")
```

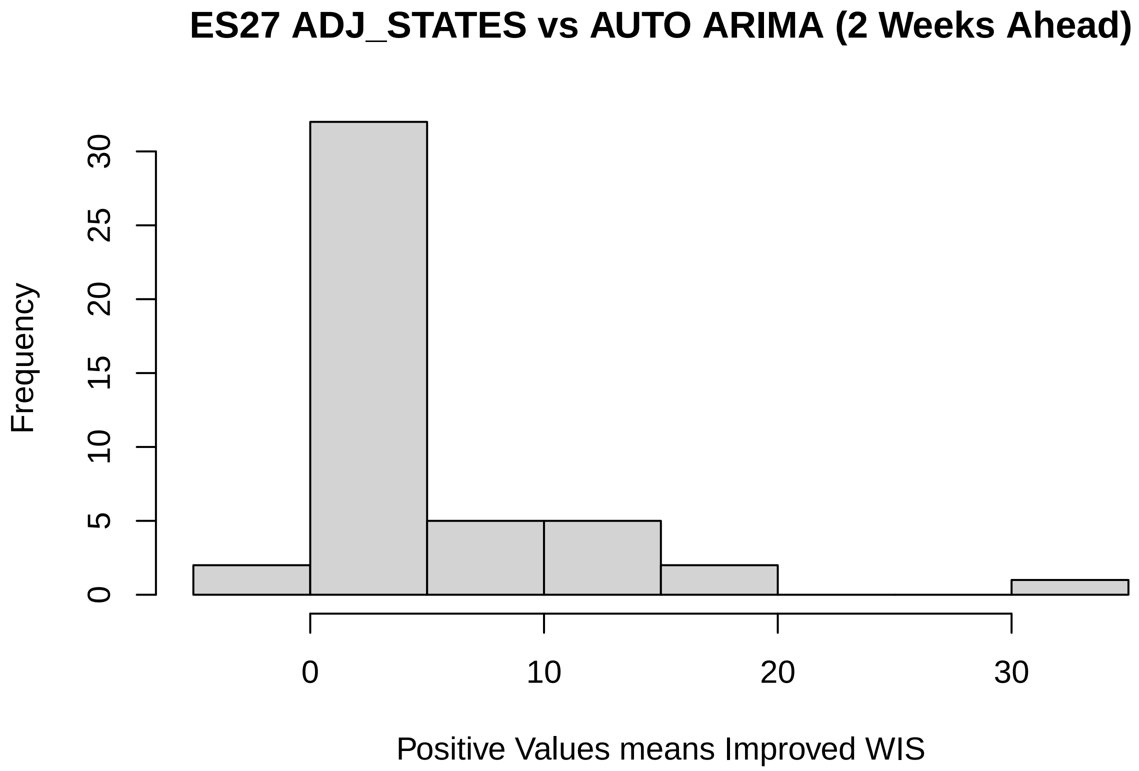


COMPARING WIS RESULTS FOR WEEK1

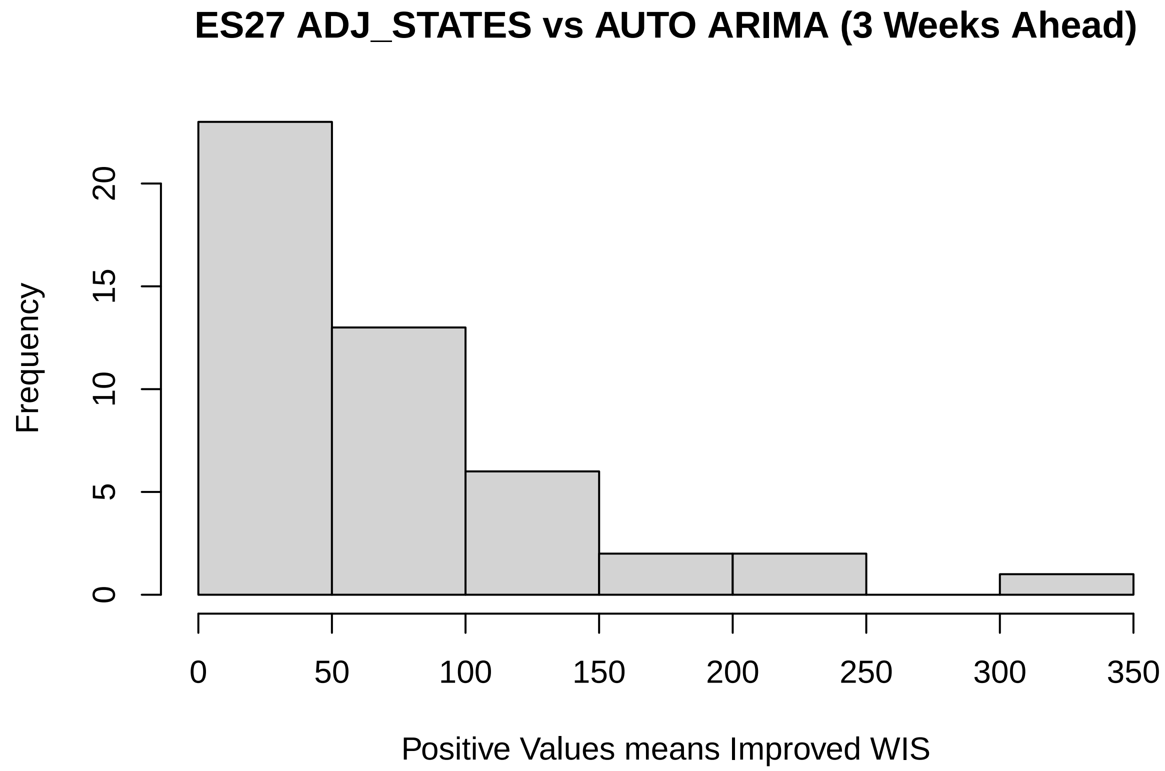
```
###########################
# WEEK1

W1_<-NULL
# AUTO ARIMA RESULTS
AUTO_AR_W1_ <- calculate_mean_wis(AUTO_ARIMA_WEEK1_list)
ES27_ADJ_W1_ <- calculate_mean_wis(ES27_ADJACENT_WEEK1_list)
# AUTO ARIMA RESULTS
W1_ <- merge(AUTO_AR_W1_, ES27_ADJ_W1_, by = "State")
# Rename columns for clarity
colnames(W1_)[1] <- "NAME"
colnames(W1_)[2] <- "AUTO_AR"
colnames(W1_)[3] <- "ES27_ADJ"

###########################
# WEEK2

W2_<-NULL
# AUTO ARIMA RESULTS
AUTO_AR_W2_ <- calculate_mean_wis(AUTO_ARIMA_WEEK2_list)
ES27_ADJ_W2_ <- calculate_mean_wis(ES27_ADJACENT_WEEK2_list)
# AUTO ARIMA RESULTS
W2_ <- merge(AUTO_AR_W2_, ES27_ADJ_W2_, by = "State")
# Rename columns for clarity
colnames(W2_)[1] <- "NAME"
colnames(W2_)[2] <- "AUTO_AR"
colnames(W2_)[3] <- "ES27_ADJ"

###########################
# WEEK3

W3_<-NULL
# AUTO ARIMA RESULTS
AUTO_AR_W3_ <- calculate_mean_wis(AUTO_ARIMA_WEEK3_list)
ES27_ADJ_W3_ <- calculate_mean_wis(ES27_ADJACENT_WEEK1_list)
# AUTO ARIMA RESULTS
W3_ <- merge(AUTO_AR_W3_, ES27_ADJ_W3_, by = "State")
# Rename columns for clarity
colnames(W3_)[1] <- "NAME"
colnames(W3_)[2] <- "AUTO_AR"
colnames(W3_)[3] <- "ES27_ADJ"

###########################
# WEEK4

W4_<-NULL
# AUTO ARIMA RESULTS
AUTO_AR_W4_ <- calculate_mean_wis(AUTO_ARIMA_WEEK4_list)
ES27_ADJ_W4_ <- calculate_mean_wis(ES27_ADJACENT_WEEK4_list)
# AUTO ARIMA RESULTS
W4_ <- merge(AUTO_AR_W4_, ES27_ADJ_W4_, by = "State")
# Rename columns for clarity
colnames(W4_)[1] <- "NAME"
colnames(W4_)[2] <- "AUTO_AR"
colnames(W4_)[3] <- "ES27_ADJ"

# Improved WIS

hist(W1_$AUTO_AR-W1_$ES27_ADJ, main="ES27 ADJ_STATES vs AUTO ARIMA (1 Week Ahead)", xlab = "Positive Va
```



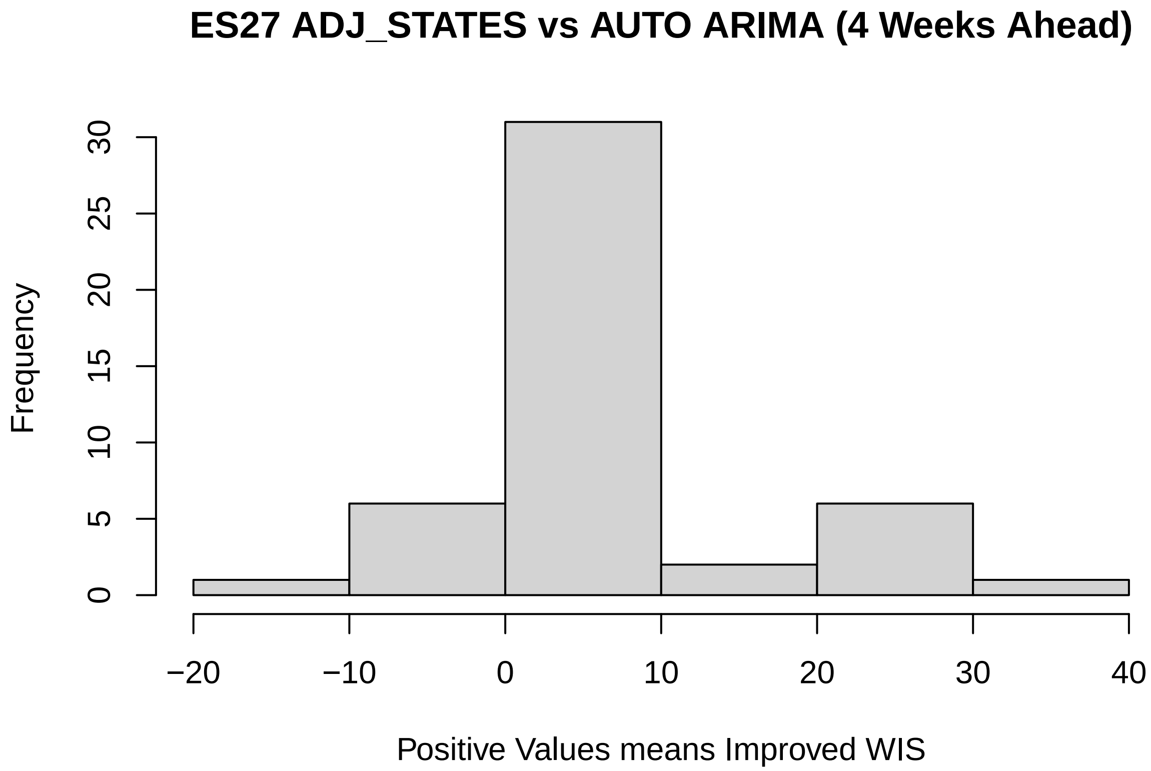
```
hist(W2_$AUTO_AR-W2_$ES27_ADJ, main="ES27 ADJ_STATES vs AUTO ARIMA (2 Weeks Ahead)", xlab = "Positive V
```



```
hist(W3_$AUTO_AR-W3_$ES27_ADJ, main="ES27 ADJ_STATES vs AUTO ARIMA (3 Weeks Ahead)", xlab = "Positive Va
```



```
hist(W4_$AUTO_AR-W4_$ES27_ADJ, main="ES27 ADJ_STATES vs AUTO ARIMA (4 Weeks Ahead)", xlab = "Positive Va
```


ES27 ADJ_STATES vs AUTO ARIMA (4 Weeks Ahead)
Frequency
0
5
10
15
20
25
30
−20
−10
0
10
20
30
40
Positive Values means Improved WIS