ENSEMBLES OF ARIMA AND ARIMAX MODELS FOR FLU FORECASTING

Victor Felix

2024-06-11

LOADING LIBRARIES

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

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

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

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

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

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

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

```
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
```

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

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

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

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

```
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
```

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

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

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

```
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
```

```
library("forcats")
```

LOADING DATA

```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
#calculate_mean_abs_error
# Define the function
calculate_mean_wis <- function(state_list) {
  # Capture the name of the input list
  list_name <- deparse(substitute(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, na.rm = TRUE)  # Use na.rm = TRUE to handle NA values

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

  # Create a data frame with a dynamic column name
  results_df <- data.frame(State = state_names)
  results_df[[list_name]] <- mean_values

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

COMPARING WIS RESULTS FOR 1-4 WEEKS AHEAD

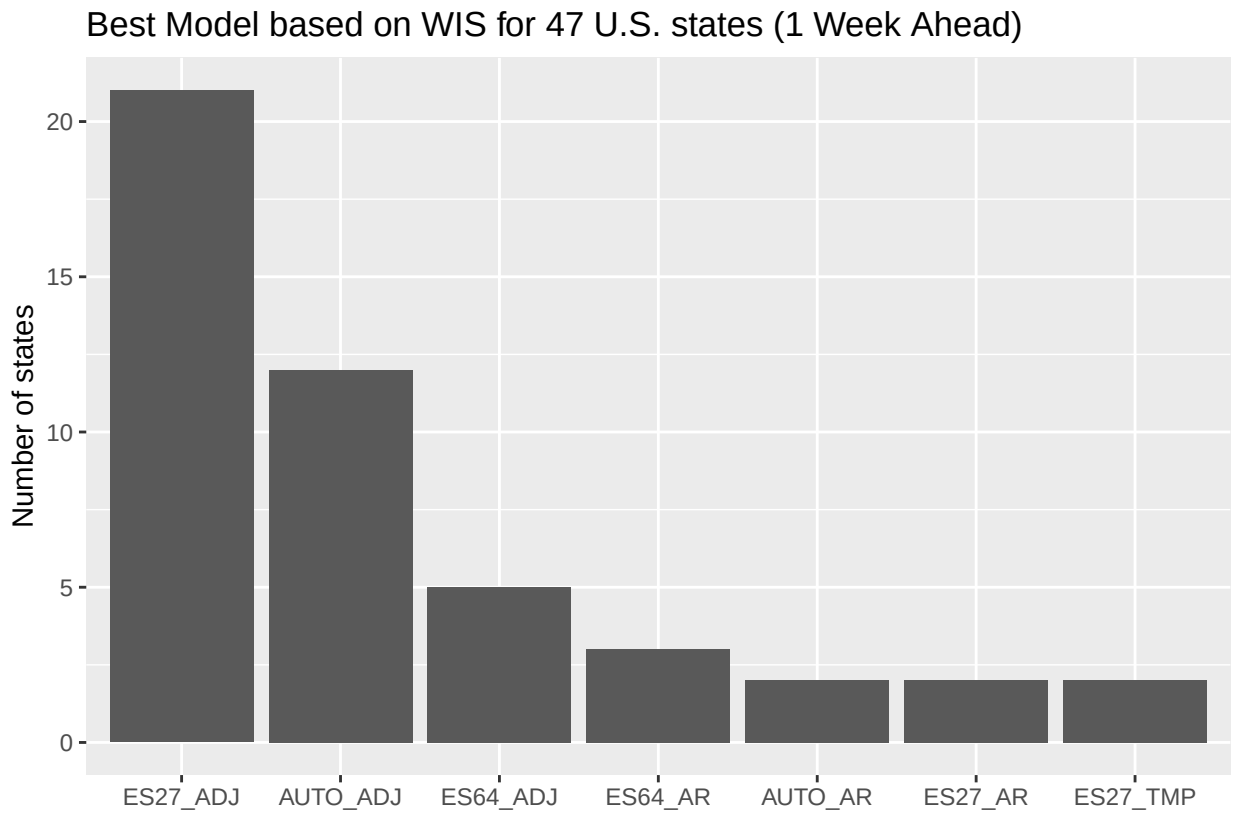
```r
W1<-NULL

# 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)
```

```r
# 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)

# 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")
W1 <- merge(W1, ES27_ADJ_W1, by = "State")
W1 <- merge(W1, ES64_ADJ_W1, by = "State")
# TEMPERATURE ARIMAX RESULTS
W1 <- merge(W1, AUTO_TMP_W1, by = "State")
W1 <- merge(W1, ES27_TMP_W1, by = "State")
W1 <- merge(W1, ES64_TMP_W1, by = "State")
# EPIWEEK ARIMAX RESULTS
W1 <- merge(W1, AUTO_EPI_W1, by = "State")
W1 <- merge(W1, ES27_EPI_W1, by = "State")
W1 <- merge(W1, ES64_EPI_W1, by = "State")

# 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
})

# REORDER BY FREQUENCY
W1$Best_Result <- fct_infreq(W1$Best_Result)
```

```r
# Print merged results
head(W1)
```

```
##          NAME   AUTO_AR   ES27_AR   ES64_AR  AUTO_ADJ  ES27_ADJ  ES64_ADJ
## 1     Alabama 103.40932 104.09394 105.01240 101.54615 100.22516 109.00636
## 2     Arizona  46.30226  46.47684  47.58219  45.44048  44.50197  45.92314
## 3    Arkansas  24.37096  23.77534  23.89239  25.09972  24.92200  24.84988
## 4  California 156.49935 155.68676 154.54709 155.73265 152.15700 159.08651
## 5    Colorado  69.93372  69.35773  69.36652  66.80057  66.58418  71.39554
## 6 Connecticut  51.27157  50.95444  52.11211  50.30640  51.08460  54.48205
##    AUTO_TMP  ES27_TMP  ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI Best_Result
## 1 104.32674 105.67750 107.36301 109.15158 108.02119 107.24986    ES27_ADJ
## 2  46.79212  46.39031  46.86298  48.55342  47.42529  47.81730    ES27_ADJ
## 3  25.05808  24.27697  24.64173  25.41587  25.32682  25.04705     ES27_AR
## 4 158.70497 157.59665 156.44105 163.92647 160.31558 161.32314    ES27_ADJ
## 5  70.28172  70.23868  70.34410  72.15381  71.43358  71.59568    ES27_ADJ
## 6  52.18867  53.30628  55.10654  53.68471  52.95653  54.84014    AUTO_ADJ
```

```r
# --------- WEEK1 MODELS ------------- #
ggplot(W1,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Best Model based on WIS for 47 U.S. states (1 Week Ahead)",
       x = "", y="Number of states") #+ scale_x_discrete()
```



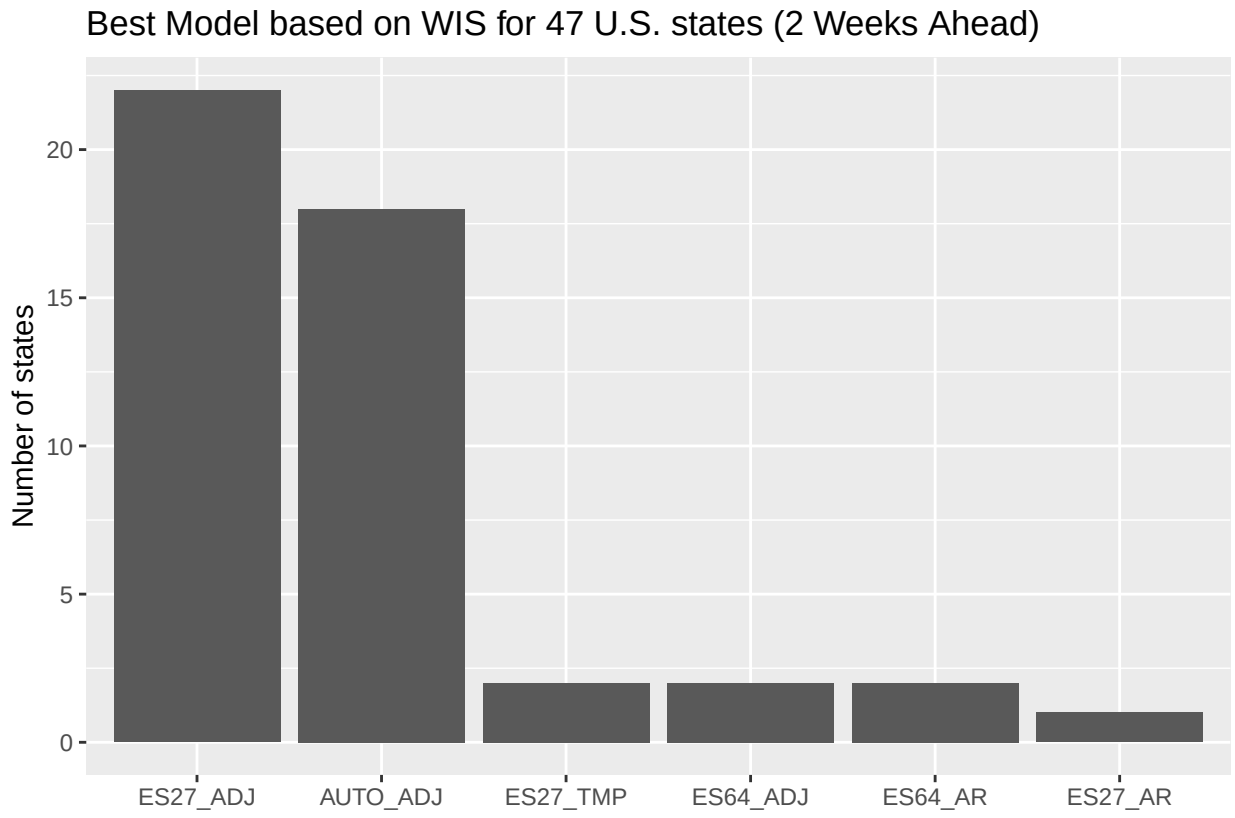
WEEK 2

```r
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")
W2 <- merge(W2, ES27_ADJ_W2, by = "State")
W2 <- merge(W2, ES64_ADJ_W2, by = "State")
# TEMPERATURE ARIMAX RESULTS
W2 <- merge(W2, AUTO_TMP_W2, by = "State")
W2 <- merge(W2, ES27_TMP_W2, by = "State")
W2 <- merge(W2, ES64_TMP_W2, by = "State")
# EPIWEEK ARIMAX RESULTS
W2 <- merge(W2, AUTO_EPI_W2, by = "State")
W2 <- merge(W2, ES27_EPI_W2, by = "State")
W2 <- merge(W2, ES64_EPI_W2, by = "State")

# 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) {
```

```r
#  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
})

# REORDER BY FREQUENCY
W2$Best_Result <- fct_infreq(W2$Best_Result)

# Print merged results
head(W2)
```

```
##          NAME   AUTO_AR   ES27_AR   ES64_AR  AUTO_ADJ  ES27_ADJ  ES64_ADJ
## 1     Alabama 176.53795 176.72366 178.27097 171.94583 172.52319 177.75992
## 2     Arizona  66.81567  67.01116  67.56960  67.07318  63.08639  66.23198
## 3    Arkansas  35.98807  35.23375  35.46221  35.12476  35.53055  36.71628
## 4  California 260.19097 254.82103 252.38074 255.55157 247.32318 261.97945
## 5    Colorado 105.10490 103.19000 103.30953  99.59710 100.35438 106.68691
## 6 Connecticut  81.70335  82.35138  83.30729  79.08185  80.24985  87.33563
##    AUTO_TMP  ES27_TMP  ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI Best_Result
## 1 175.99935 175.20471 179.15741 180.54717 178.18208 174.87487    AUTO_ADJ
## 2  66.82924  65.87824  66.15905  69.35285  67.99691  67.71082    ES27_ADJ
## 3  36.45988  35.66592  35.15920  36.66010  36.89276  36.66611    AUTO_ADJ
## 4 263.99879 259.19018 254.48401 270.69366 260.59695 261.69534    ES27_ADJ
## 5 105.29249 104.49503 104.66056 107.52025 105.35086 104.82870    AUTO_ADJ
## 6  82.32661  85.30291  88.43351  84.00331  82.09353  85.10770    AUTO_ADJ
```

```r
# --------- WEEK1 MODELS ------------- #
ggplot(W2,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Best Model based on WIS for 47 U.S. states (2 Weeks Ahead)",
       x = "", y="Number of states")
```



WEEK 3

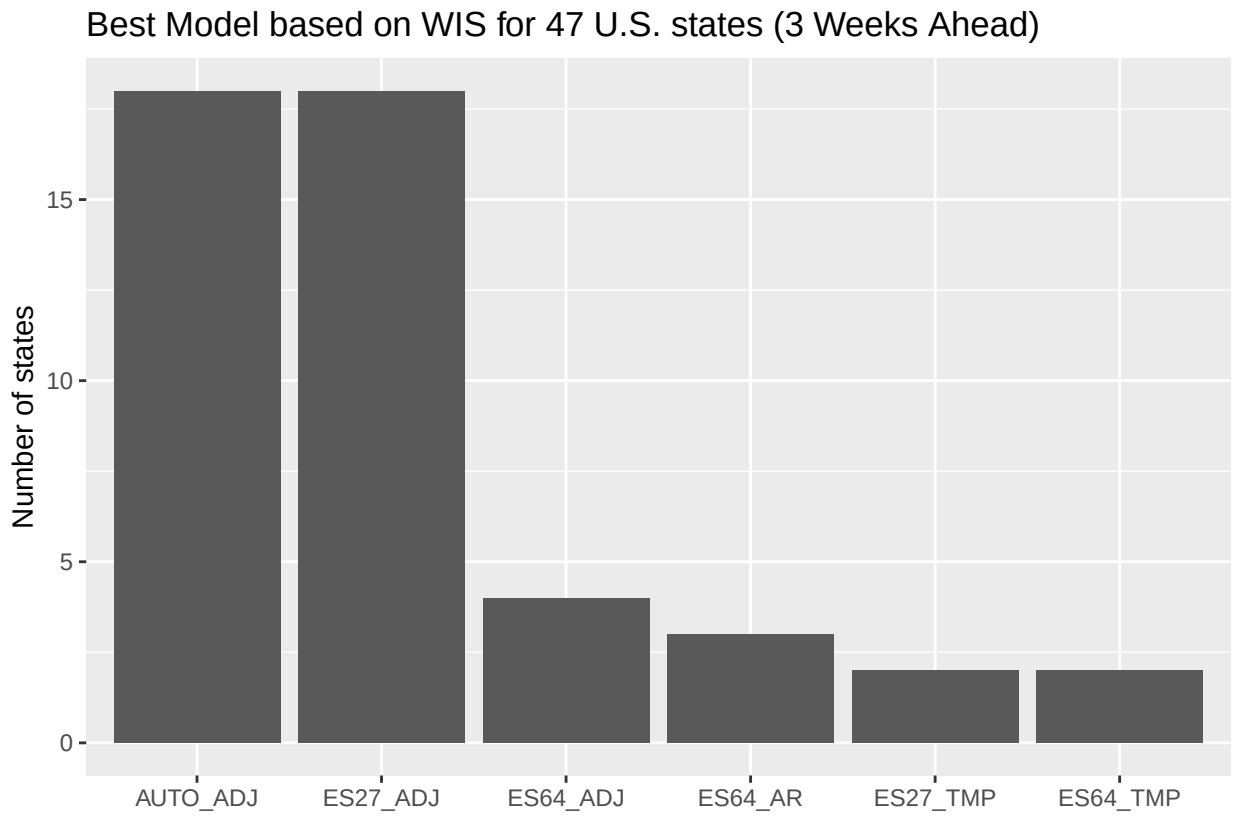
```
W3<-NULL

# 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")
W3 <- merge(W3, ES27_ADJ_W3, by = "State")
W3 <- merge(W3, ES64_ADJ_W3, by = "State")
# TEMPERATURE ARIMAX RESULTS
W3 <- merge(W3, AUTO_TMP_W3, by = "State")
W3 <- merge(W3, ES27_TMP_W3, by = "State")
W3 <- merge(W3, ES64_TMP_W3, by = "State")
# EPIWEEK ARIMAX RESULTS
W3 <- merge(W3, AUTO_EPI_W3, by = "State")
W3 <- merge(W3, ES27_EPI_W3, by = "State")
W3 <- merge(W3, ES64_EPI_W3, by = "State")

# 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
})

# REORDER BY FREQUENCY
W3$Best_Result <- fct_infreq(W3$Best_Result)

# Print merged results
head(W3)
```

```
##          NAME   AUTO_AR   ES27_AR   ES64_AR  AUTO_ADJ  ES27_ADJ  ES64_ADJ
## 1     Alabama 234.88777 232.77346 232.89241 225.50942 228.68937 233.74237
## 2     Arizona  85.84635  86.91635  86.33633  89.32789  81.79644  86.83405
## 3    Arkansas  46.63206  45.93368  46.03893  43.84612  44.93047  47.18447
## 4  California 356.94659 347.46636 342.95515 353.55475 338.96871 360.00363
## 5    Colorado 142.10265 138.55124 138.78039 133.70220 136.43488 142.01284
## 6 Connecticut 106.59585 111.68002 110.47228 102.60351 105.20829 115.11376
##    AUTO_TMP  ES27_TMP  ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI Best_Result
## 1 233.29858 230.85754 232.48676 236.44039 232.47689 227.20783    AUTO_ADJ
## 2  85.40273  84.04960  83.72061  90.40414  89.00180  88.16697    ES27_ADJ
## 3  46.92462  46.37374  43.97295  46.74298  47.21512  46.45097    AUTO_ADJ
```

```
## 4 361.10793 355.07486 348.09042 374.25618 359.89251 360.09505    ES27_ADJ
## 5 141.47293 140.57502 140.36094 143.39508 139.77563 139.55267    AUTO_ADJ
## 6 107.29540 115.28619 116.48950 108.21244 108.31155 110.32945    AUTO_ADJ
```

```
# --------- WEEK1 MODELS ------------- #
ggplot(W3,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Best Model based on WIS for 47 U.S. states (3 Weeks Ahead)",
       x = "", y="Number of states")
```



WEEK 4

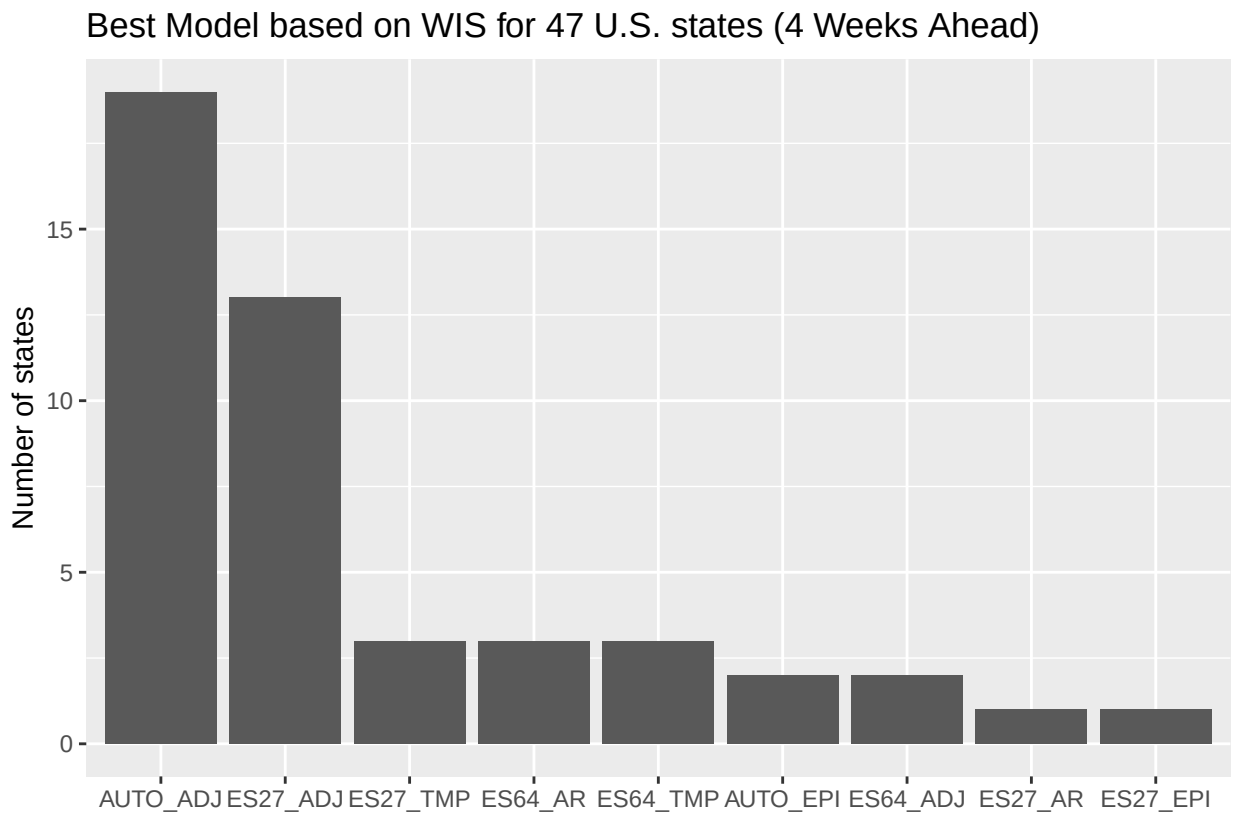
```
W4<-NULL

# 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
```

```r
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")
W4 <- merge(W4, ES27_ADJ_W4, by = "State")
W4 <- merge(W4, ES64_ADJ_W4, by = "State")
# TEMPERATURE ARIMAX RESULTS
W4 <- merge(W4, AUTO_TMP_W4, by = "State")
W4 <- merge(W4, ES27_TMP_W4, by = "State")
W4 <- merge(W4, ES64_TMP_W4, by = "State")
# EPIWEEK ARIMAX RESULTS
W4 <- merge(W4, AUTO_EPI_W4, by = "State")
W4 <- merge(W4, ES27_EPI_W4, by = "State")
W4 <- merge(W4, ES64_EPI_W4, by = "State")

# 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
})


W4$Best_Result <- fct_infreq(W4$Best_Result)

# Print merged results
head(W4)
```

```
##          NAME  AUTO_AR   ES27_AR  ES64_AR  AUTO_ADJ  ES27_ADJ  ES64_ADJ
## 1     Alabama 275.2895 271.45712 269.0416 261.92524 267.68264 276.10858
```

```
## 2     Arizona 102.6539 105.12352 103.9090 110.11321  99.90226 105.71601
## 3    Arkansas  55.6695  55.81443  55.4770  52.27615  53.98102  56.75213
## 4  California 453.3795 440.40877 430.8554 448.98797 429.57080 453.96445
## 5    Colorado 173.9942 170.30228 170.0740 162.98801 167.80131 172.22182
## 6 Connecticut 130.3734 146.30924 141.1414 122.83695 132.29750 142.93615
##    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    AUTO_ADJ
## 2 102.77459 100.92540  99.46926 109.09777 107.06585 105.55045    ES64_TMP
## 3  56.16726  56.44215  52.45936  56.43826  57.19076  55.85982    AUTO_ADJ
## 4 454.72572 447.55868 437.97009 478.56280 456.37471 455.32237    ES27_ADJ
## 5 173.44942 172.78770 171.68062 174.27810 171.00633 170.60375    AUTO_ADJ
## 6 134.29882 150.17829 147.67298 133.59970 137.09017 139.77165    AUTO_ADJ
```

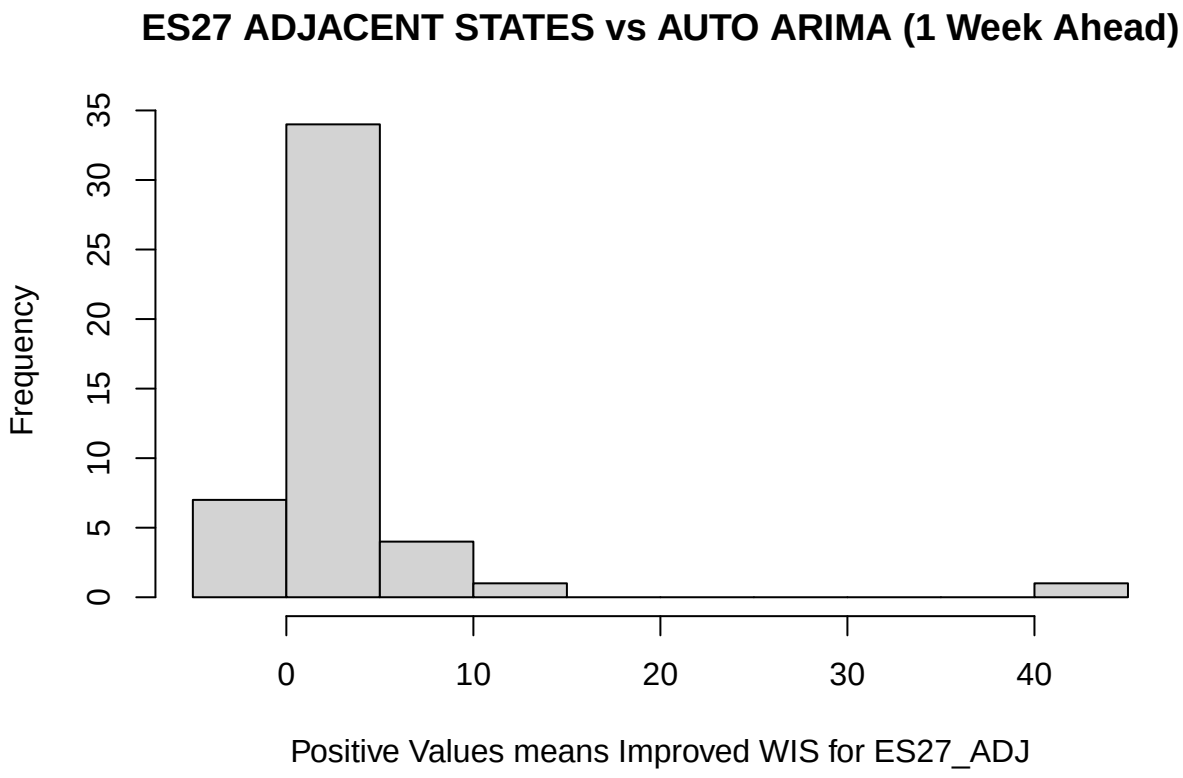
```
# --------- WEEK1 MODELS ------------- #
ggplot(W4,aes(x=Best_Result)) + geom_bar()+
  labs(title = "Best Model based on WIS for 47 U.S. states (4 Weeks Ahead)",
       x = "", y="Number of states")
```



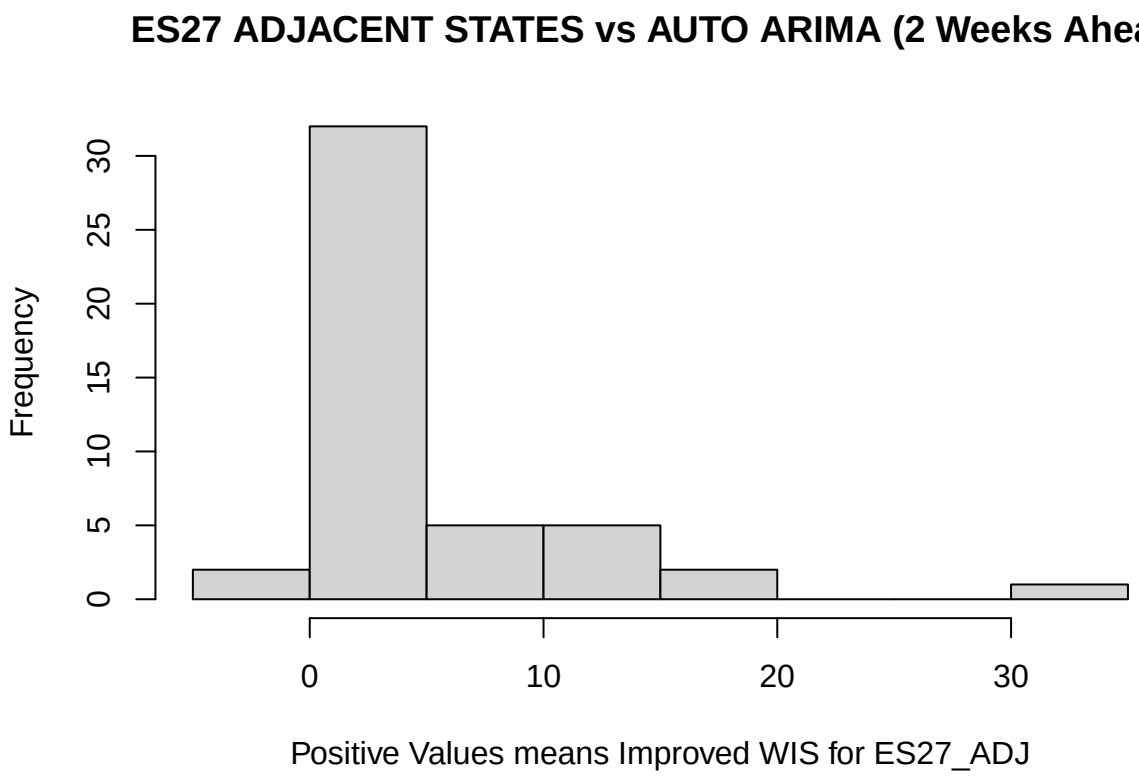
COMPARING AUTO ARIMA AND ADJACENT STATES ARIMAX WIS RESULTS FOR 1-4 WEEKS AHEAD

```
########
# WEEK1
W1_<-NULL
# AUTO ARIMA RESULTS
AUTO_AR_W1_ <- calculate_mean_wis(AUTO_ARIMA_WEEK1_list)
```

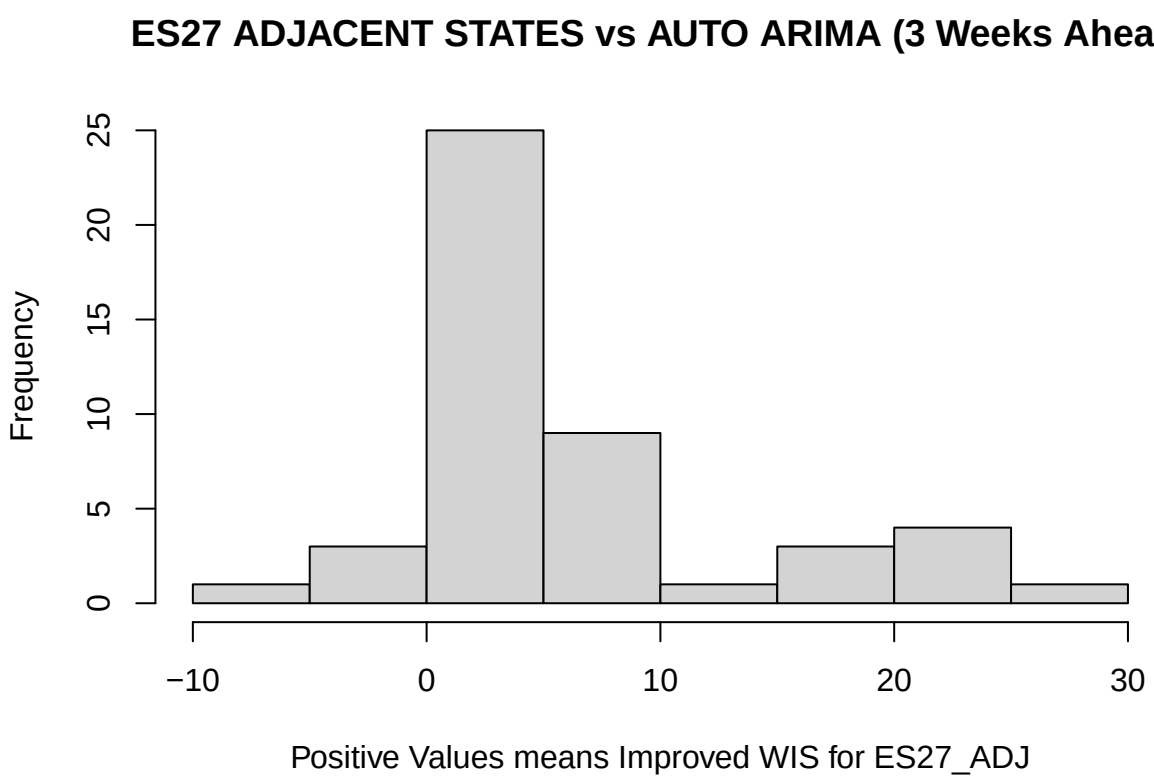
```r
ES27_ADJ_W1_ <- calculate_mean_wis(ES27_ADJACENT_WEEK1_list)
# AUTO ARIMA RESULTS
W1_ <- merge(AUTO_AR_W1_, ES27_ADJ_W1_, by = "State")
# Rename columns for clarity
colnames(W1_)[1] <- "NAME"
colnames(W1_)[2] <- "AUTO_AR"
colnames(W1_)[3] <- "ES27_ADJ"

########
# 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_WEEK3_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 ADJACENT STATES vs AUTO ARIMA (1 Week Ahead)", xlab = "Positi
```



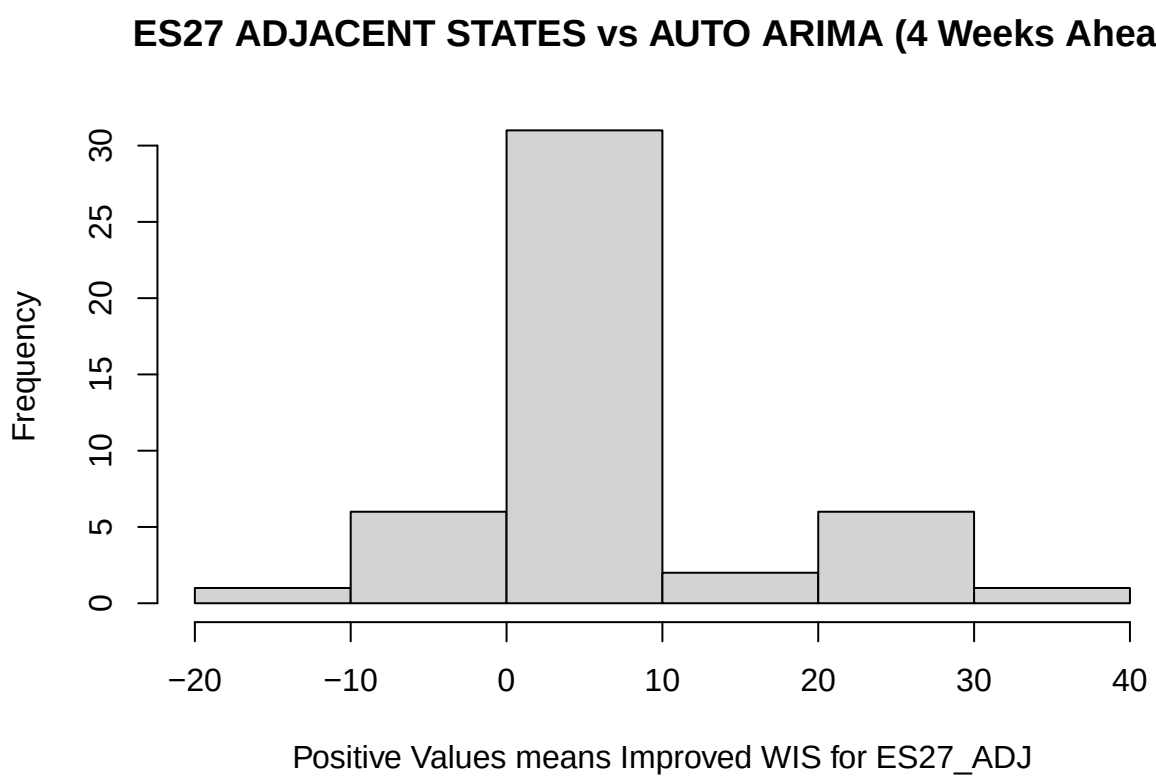

```
hist(W2_$AUTO_AR-W2_$ES27_ADJ, main="ES27 ADJACENT STATES vs AUTO ARIMA (2 Weeks Ahead)", xlab = "Positi
```




```
hist(W3_$AUTO_AR-W3_$ES27_ADJ, main="ES27 ADJACENT STATES vs AUTO ARIMA (3 Weeks Ahead)", xlab = "Positi
```

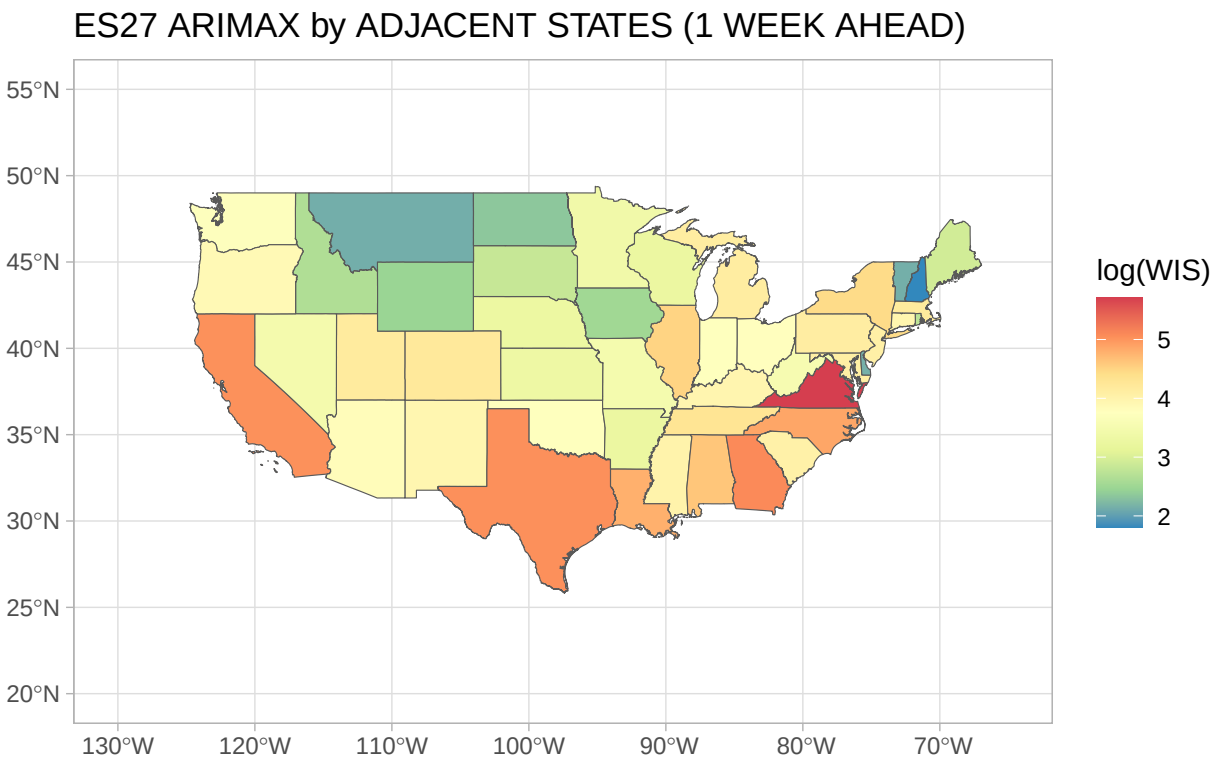



```
hist(W4_$AUTO_AR-W4_$ES27_ADJ, main="ES27 ADJACENT STATES vs AUTO ARIMA (4 Weeks Ahead)", xlab = "Positi
```

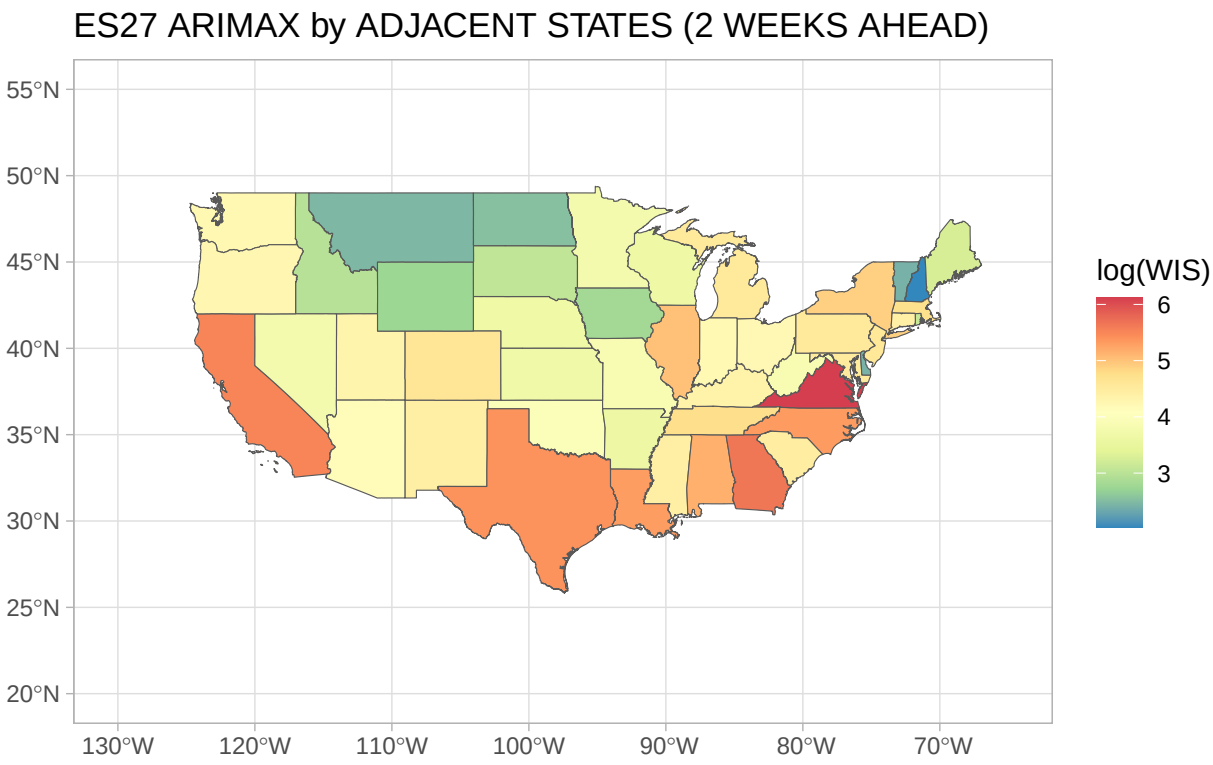


log(WIS) MAPS for 1-4 WEEKS AHEAD

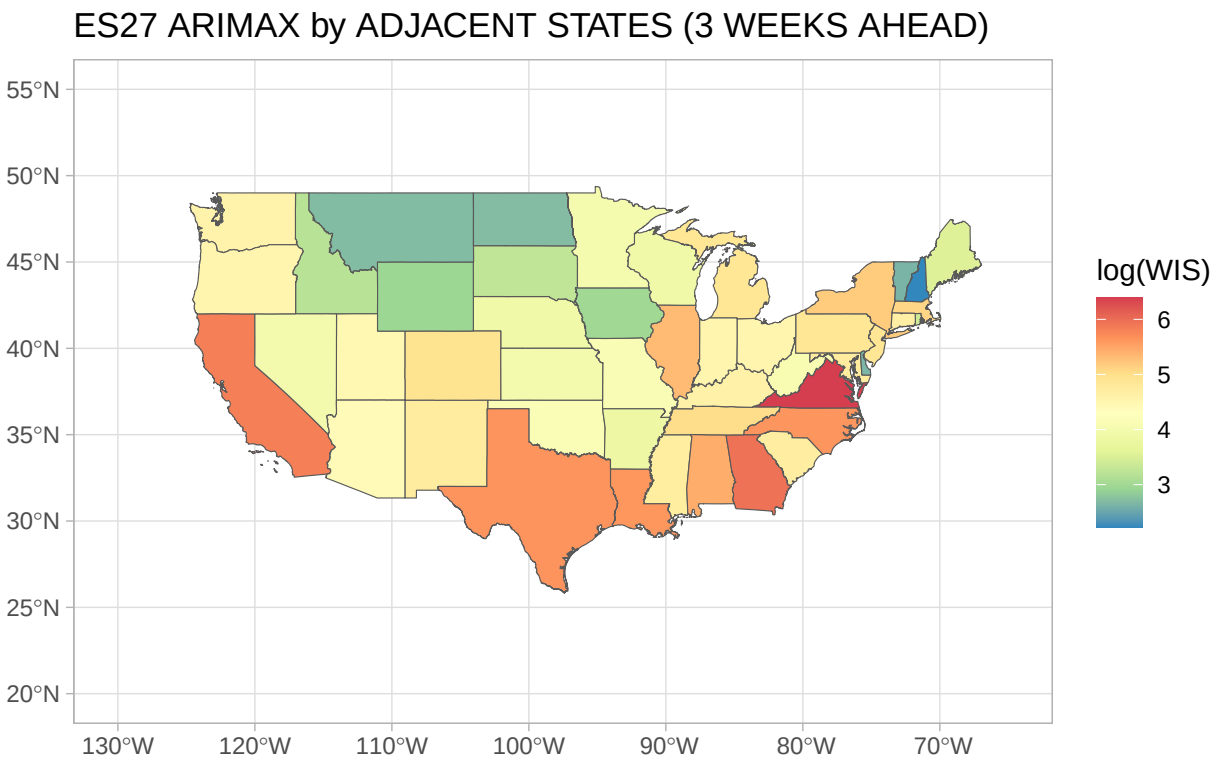
```
#########################################################
# MAPPING THE MEAN PERFORMANCE OF THE ES64, ES27      #
# AND AUTO ARIMA MODELS FOR THE 50 STATES OF THE U.S. #
# BASED ON THE SUMMARY RESULTS OF THE CURRENT MODELS. #
#########################################################

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)
```



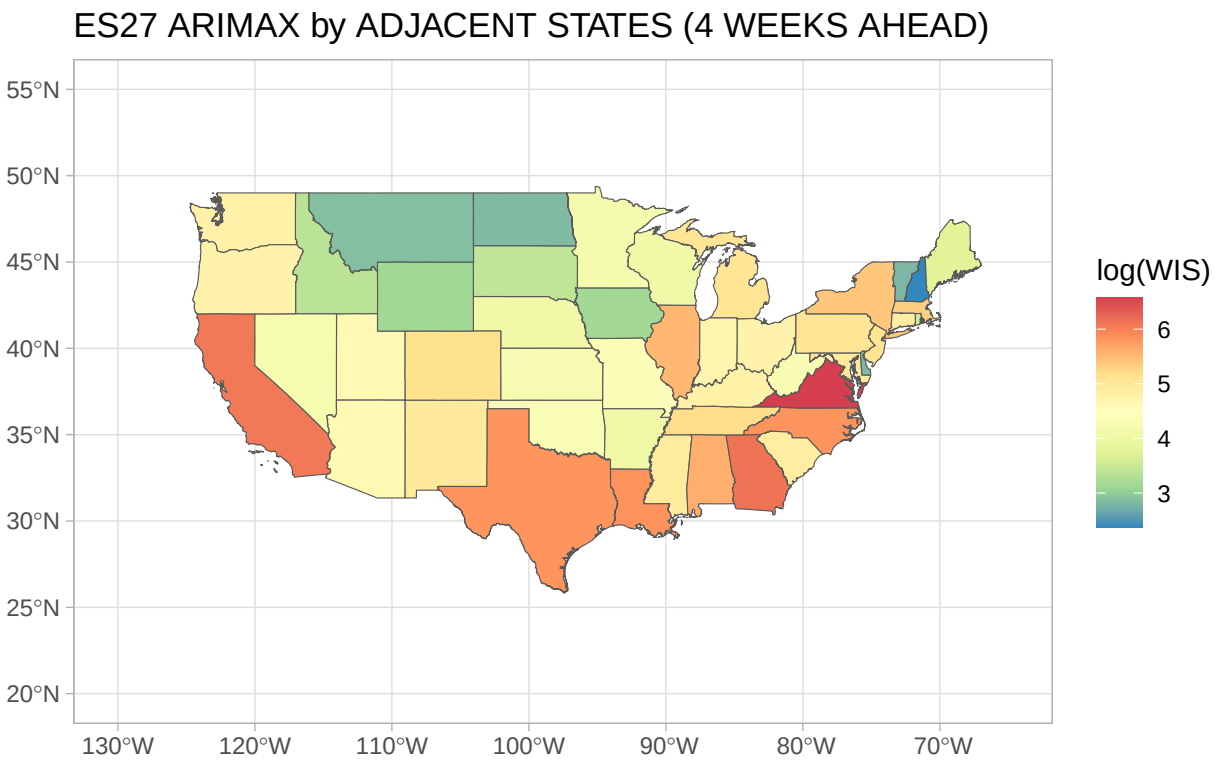

```
#####################################################
# ES27 ARIMAX by ADJACENT STATES - 2 WEEKS AHEAD #
#####################################################

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)
```


ES27 ARIMAX by ADJACENT STATES (4 WEEKS AHEAD)
55°N
50°N
45°N
40°N
35°N
30°N
25°N
20°N
130°W
120°W
110°W
100°W
90°W
80°W
70°W
log(WIS)
6
5
4
3