U.S. Electric Grid: Will it power the future?

Ben Mayes

2025-06-01

# Install and Load Required Libraries

options(repos=c(CRAN="https://cran.r-project.org"))  
install.packages("forecast")

## Installing package into 'C:/Users/00815/AppData/Local/R/win-library/4.5'  
## (as 'lib' is unspecified)

## package 'forecast' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'forecast'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\00815\AppData\Local\R\win-library\4.5\00LOCK\forecast\libs\x64\forecast.dll  
## to  
## C:\Users\00815\AppData\Local\R\win-library\4.5\forecast\libs\x64\forecast.dll:  
## Permission denied

## Warning: restored 'forecast'

##   
## The downloaded binary packages are in  
## C:\Users\00815\AppData\Local\Temp\RtmpeiibTc\downloaded\_packages

install.packages("readr")

## Installing package into 'C:/Users/00815/AppData/Local/R/win-library/4.5'  
## (as 'lib' is unspecified)

## package 'readr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'readr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\00815\AppData\Local\R\win-library\4.5\00LOCK\readr\libs\x64\readr.dll  
## to C:\Users\00815\AppData\Local\R\win-library\4.5\readr\libs\x64\readr.dll:  
## Permission denied

## Warning: restored 'readr'

##   
## The downloaded binary packages are in  
## C:\Users\00815\AppData\Local\Temp\RtmpeiibTc\downloaded\_packages

install.packages("stats")

## Warning: package 'stats' is in use and will not be installed

install.packages("tseries")

## Installing package into 'C:/Users/00815/AppData/Local/R/win-library/4.5'  
## (as 'lib' is unspecified)

## package 'tseries' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'tseries'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\00815\AppData\Local\R\win-library\4.5\00LOCK\tseries\libs\x64\tseries.dll  
## to C:\Users\00815\AppData\Local\R\win-library\4.5\tseries\libs\x64\tseries.dll:  
## Permission denied

## Warning: restored 'tseries'

##   
## The downloaded binary packages are in  
## C:\Users\00815\AppData\Local\Temp\RtmpeiibTc\downloaded\_packages

install.packages("lmtest")

## Installing package into 'C:/Users/00815/AppData/Local/R/win-library/4.5'  
## (as 'lib' is unspecified)

## package 'lmtest' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'lmtest'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\00815\AppData\Local\R\win-library\4.5\00LOCK\lmtest\libs\x64\lmtest.dll  
## to C:\Users\00815\AppData\Local\R\win-library\4.5\lmtest\libs\x64\lmtest.dll:  
## Permission denied

## Warning: restored 'lmtest'

##   
## The downloaded binary packages are in  
## C:\Users\00815\AppData\Local\Temp\RtmpeiibTc\downloaded\_packages

library(forecast)

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

library(readr)  
library(forecast)  
library(stats)  
library(tseries)  
library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

# Read in Demand Data

data <- read.csv("C:/Users/00815/OneDrive/Desktop/MSU/CIS 663/CIS 663 Project/Demand\_per\_month.csv")  
data$date <- as.Date(data$date, format = "%Y-%m-%d")

# Read in Generation Data

data2 <- read.csv("C:/Users/00815/OneDrive/Desktop/MSU/CIS 663/CIS 663 Project/Generation\_per\_month.csv")  
data2$date <- as.Date(data2$date, format = "%Y-%m-%d")

# View Data Head

head(data)

## date demand  
## 1 2010-01-01 364.13  
## 2 2010-02-01 322.21  
## 3 2010-03-01 314.22  
## 4 2010-04-01 289.53  
## 5 2010-05-01 328.31  
## 6 2010-06-01 377.14

head(data2)

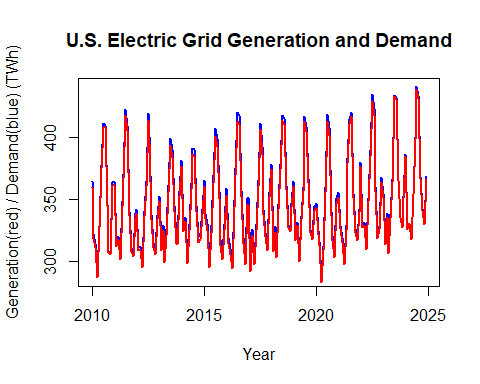
## date generation  
## 1 2010-01-01 359.99  
## 2 2010-02-01 318.82  
## 3 2010-03-01 311.17  
## 4 2010-04-01 286.92  
## 5 2010-05-01 326.99  
## 6 2010-06-01 374.64

# Create time series for demand and generation

data\_ts <- ts(data$demand, start = c(2010, 1), frequency = 12)  
data2\_ts <- ts(data2$generation, start = c(2010, 1), frequency = 12)

# Plot demand and generation time series

plot(data\_ts, main = "U.S. Electric Grid Generation and Demand", ylab = "Generation(red) / Demand(blue) (TWh)", xlab = "Year", col = "blue", lwd = 2)  
lines(data2\_ts, col = "red", lwd = 2)



# View summary of time series

summary(data\_ts)

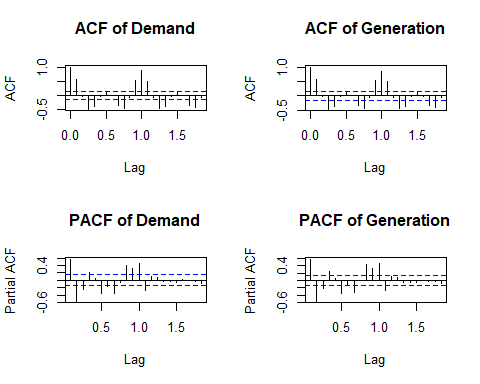
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 286.2 319.9 340.1 348.6 367.0 441.3

summary(data2\_ts)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 283.0 317.4 336.4 345.0 363.6 438.8

# Acf and Pacf plots for demand and generation

par(mfrow = c(2, 2))  
acf(data\_ts, main = "ACF of Demand")  
acf(data2\_ts, main = "ACF of Generation")  
pacf(data\_ts, main = "PACF of Demand")  
pacf(data2\_ts, main = "PACF of Generation")



# Check for stationarity

adf\_test <- adf.test(data\_ts)

## Warning in adf.test(data\_ts): p-value smaller than printed p-value

if (adf\_test$p.value < 0.05) {  
 print("The time series is stationary.")  
} else {  
 print("The time series is not stationary. Differencing may be required.")  
}

## [1] "The time series is stationary."

adf\_test2 <- adf.test(data2\_ts)

## Warning in adf.test(data2\_ts): p-value smaller than printed p-value

if (adf\_test2$p.value < 0.05) {  
 print("The time series is stationary.")  
} else {  
 print("The time series is not stationary. Differencing may be required.")  
}

## [1] "The time series is stationary."

# Train and test split demand data

train\_size <- floor(0.8 \* length(data\_ts))  
train\_data <- window(data\_ts, end = c(2019, 12))  
test\_data <- window(data\_ts, start = c(2020, 1))

# Summary of the training and test data

summary(train\_data)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 289.5 319.0 338.1 345.6 364.7 422.3

summary(test\_data)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 286.2 326.3 344.9 354.6 371.5 441.3

# Fit ARIMA model

fit <- auto.arima(train\_data)

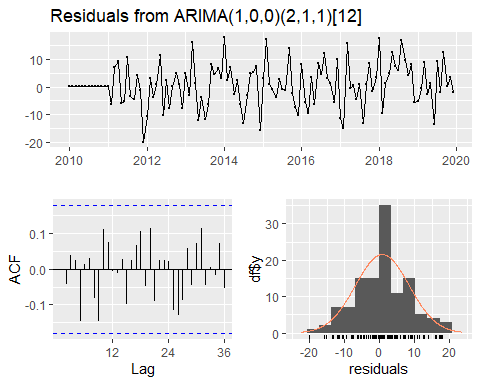
# View Arima model summary

summary(fit)

## Series: train\_data   
## ARIMA(1,0,0)(2,1,1)[12]   
##   
## Coefficients:  
## ar1 sar1 sar2 sma1  
## 0.3991 0.1836 -0.2872 -0.8301  
## s.e. 0.0886 0.1297 0.1083 0.1963  
##   
## sigma^2 = 68.13: log likelihood = -387.75  
## AIC=785.5 AICc=786.09 BIC=798.91  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.7835905 7.684274 5.866767 0.2052533 1.68668 0.6593725  
## ACF1  
## Training set 0.0008662121

# Check residuals of the fitted model demand

checkresiduals(fit)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,0)(2,1,1)[12]  
## Q\* = 16.964, df = 20, p-value = 0.6553  
##   
## Model df: 4. Total lags used: 24

# Forecast using the fitted model for the next 60 months

forecasted\_values <- forecast(fit, h = 120)

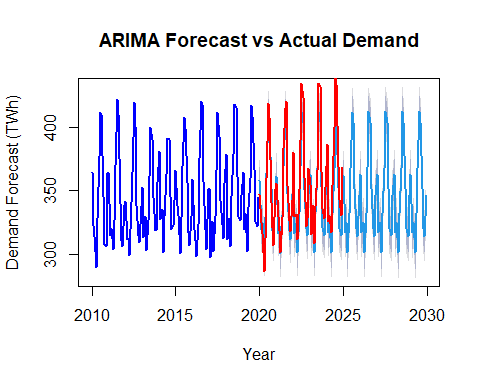
# View the forecasted values

print(forecasted\_values)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2020 358.2866 347.6509 368.9223 342.0207 374.5525  
## Feb 2020 319.0438 307.5959 330.4917 301.5358 336.5518  
## Mar 2020 325.0991 313.5271 336.6711 307.4012 342.7970  
## Apr 2020 299.6928 288.1011 311.2845 281.9649 317.4207  
## May 2020 328.7516 317.1568 340.3464 311.0189 346.4843  
## Jun 2020 363.2524 351.6571 374.8477 345.5190 380.9859  
## Jul 2020 411.2488 399.6534 422.8442 393.5152 428.9824  
## Aug 2020 401.1555 389.5601 412.7508 383.4218 418.8891  
## Sep 2020 351.9199 340.3246 363.5153 334.1863 369.6535  
## Oct 2020 318.4915 306.8962 330.0868 300.7580 336.2250  
## Nov 2020 312.0417 300.4467 323.6366 294.3088 329.7746  
## Dec 2020 346.0838 334.4914 357.6763 328.3547 363.8130  
## Jan 2021 361.1356 348.9358 373.3354 342.4776 379.7935  
## Feb 2021 316.5717 304.2760 328.8673 297.7671 335.3763  
## Mar 2021 322.8447 310.5338 335.1555 304.0168 341.6725  
## Apr 2021 300.3556 288.0423 312.6689 281.5241 319.1871  
## May 2021 330.0135 317.6998 342.3272 311.1814 348.8456  
## Jun 2021 369.2504 356.9367 381.5641 350.4182 388.0826  
## Jul 2021 410.5801 398.2664 422.8939 391.7479 429.4124  
## Aug 2021 401.7091 389.3954 414.0228 382.8769 420.5413  
## Sep 2021 347.6496 335.3359 359.9633 328.8174 366.4818  
## Oct 2021 318.4953 306.1816 330.8089 299.6632 337.3273  
## Nov 2021 311.6876 299.3745 324.0008 292.8564 330.5189  
## Dec 2021 347.0800 334.7702 359.3899 328.2537 365.9063  
## Jan 2022 363.3426 351.0450 375.6402 344.5350 382.1502  
## Feb 2022 316.3062 304.0073 328.6052 297.4966 335.1159  
## Mar 2022 324.3006 312.0014 336.5998 305.4907 343.1106  
## Apr 2022 301.1715 288.8723 313.4707 282.3615 319.9815  
## May 2022 332.4533 320.1541 344.7525 313.6432 351.2633  
## Jun 2022 369.3774 357.0782 381.6766 350.5674 388.1874  
## Jul 2022 412.0775 399.7783 424.3767 393.2675 430.8875  
## Aug 2022 403.9632 391.6639 416.2624 385.1531 422.7732  
## Sep 2022 351.1505 338.8513 363.4497 332.3405 369.9605  
## Oct 2022 320.4800 308.1809 332.7792 301.6701 339.2900  
## Nov 2022 314.3505 302.0517 326.6493 295.5411 333.1599  
## Dec 2022 346.7334 334.4366 359.0302 327.9271 365.5398  
## Jan 2023 362.9296 350.6252 375.2341 344.1116 381.7477  
## Feb 2023 316.9675 304.6598 329.2751 298.1445 335.7904  
## Mar 2023 325.2154 312.9072 337.5236 306.3917 344.0392  
## Apr 2023 301.1310 288.8227 313.4392 282.3071 319.9548  
## May 2023 332.5388 320.2305 344.8471 313.7149 351.3627  
## Jun 2023 367.6781 355.3699 379.9864 348.8542 386.5020  
## Jul 2023 412.5444 400.2362 424.8527 393.7206 431.3683  
## Aug 2023 404.2180 391.9097 416.5263 385.3941 423.0419  
## Sep 2023 353.0197 340.7114 365.3280 334.1958 371.8436  
## Oct 2023 320.8434 308.5351 333.1516 302.0196 339.6672  
## Nov 2023 314.9411 302.6331 327.2491 296.1177 333.7646  
## Dec 2023 346.3837 334.0773 358.6900 327.5627 365.2046  
## Jan 2024 362.2200 349.7363 374.7037 343.1278 381.3121  
## Feb 2024 317.1651 304.6520 329.6783 298.0279 336.3024  
## Mar 2024 324.9652 312.4474 337.4831 305.8208 344.1096  
## Apr 2024 300.8892 288.3706 313.4078 281.7437 320.0347  
## May 2024 331.8538 319.3351 344.3726 312.7081 350.9996  
## Jun 2024 367.3297 354.8109 379.8484 348.1839 386.4754  
## Jul 2024 412.2002 399.6814 424.7189 393.0544 431.3459  
## Aug 2024 403.6175 391.0987 416.1362 384.4717 422.7632  
## Sep 2024 352.3575 339.8387 364.8762 333.2117 371.5032  
## Oct 2024 320.3401 307.8214 332.8588 301.1944 339.4858  
## Nov 2024 314.2848 301.7664 326.8032 295.1396 333.4300  
## Dec 2024 346.4190 333.9024 358.9356 327.2765 365.5615  
## Jan 2025 362.2083 349.5298 374.8867 342.8183 381.5983  
## Feb 2025 317.0115 304.3058 329.7172 297.5799 336.4431  
## Mar 2025 324.6566 311.9466 337.3666 305.2183 344.0948  
## Apr 2025 300.8565 288.1458 313.5671 281.4171 320.2958  
## May 2025 331.7035 318.9927 344.4143 312.2640 351.1430  
## Jun 2025 367.7537 355.0429 380.4645 348.3142 387.1932  
## Jul 2025 412.0028 399.2920 424.7137 392.5633 431.4424  
## Aug 2025 403.4340 390.7232 416.1448 383.9945 422.8735  
## Sep 2025 351.6990 338.9882 364.4099 332.2595 371.1386  
## Oct 2025 320.1433 307.4326 332.8541 300.7039 339.5827  
## Nov 2025 313.9947 301.2842 326.7051 294.5557 333.4336  
## Dec 2025 346.5259 333.8174 359.2345 327.0899 365.9620  
## Jan 2026 362.4099 349.6055 375.2144 342.8272 381.9927  
## Feb 2026 316.9265 304.1050 329.7480 297.3178 336.5353  
## Mar 2026 324.6717 311.8475 337.4960 305.0588 344.2847  
## Apr 2026 300.9199 288.0952 313.7445 281.3063 320.5335  
## May 2026 331.8726 319.0479 344.6973 312.2589 351.4863  
## Jun 2026 367.9316 355.1069 380.7564 348.3179 387.5454  
## Jul 2026 412.0655 399.2408 424.8902 392.4518 431.6792  
## Aug 2026 403.5728 390.7481 416.3975 383.9591 423.1865  
## Sep 2026 351.7683 338.9436 364.5931 332.1546 371.3820  
## Oct 2026 320.2517 307.4271 333.0764 300.6381 339.8654  
## Nov 2026 314.1299 301.3055 326.9543 294.5167 333.7431  
## Dec 2026 346.5354 333.7129 359.3579 326.9251 366.1457  
## Jan 2027 362.4503 349.5400 375.3606 342.7057 382.1949  
## Feb 2027 316.9551 304.0290 329.8811 297.1864 336.7237  
## Mar 2027 324.7632 311.8347 337.6917 304.9907 344.5357  
## Apr 2027 300.9409 288.0120 313.8698 281.1678 320.7140  
## May 2027 331.9469 319.0179 344.8758 312.1737 351.7200  
## Jun 2027 367.8425 354.9135 380.7715 348.0693 387.6157  
## Jul 2027 412.1337 399.2047 425.0627 392.3605 431.9069  
## Aug 2027 403.6510 390.7220 416.5800 383.8778 423.4242  
## Sep 2027 351.9702 339.0412 364.8991 332.1970 371.7433  
## Oct 2027 320.3281 307.3992 333.2571 300.5550 340.1013  
## Nov 2027 314.2380 301.3094 327.1667 294.4654 334.0107  
## Dec 2027 346.5065 333.5796 359.4333 326.7365 366.2764  
## Jan 2028 362.3998 349.3713 375.4284 342.4744 382.3252  
## Feb 2028 316.9847 303.9383 330.0311 297.0320 336.9374  
## Mar 2028 324.7756 311.7264 337.8248 304.8186 344.7326  
## Apr 2028 300.9266 287.8769 313.9762 280.9688 320.8843  
## May 2028 331.9119 318.8622 344.9616 311.9541 351.8697  
## Jun 2028 367.7751 354.7253 380.8248 347.8172 387.7329  
## Jul 2028 412.1282 399.0784 425.1779 392.1703 432.0860  
## Aug 2028 403.6255 390.5757 416.6752 383.6676 423.5833  
## Sep 2028 351.9873 338.9376 365.0370 332.0295 371.9452  
## Oct 2028 320.3110 307.2614 333.3607 300.3533 340.2688  
## Nov 2028 314.2191 301.1697 327.2685 294.2617 334.1764  
## Dec 2028 346.4984 333.4508 359.5460 326.5438 366.4530  
## Jan 2029 362.3790 349.2256 375.5323 342.2627 382.4952  
## Feb 2029 316.9819 303.8102 330.1537 296.8375 337.1264  
## Mar 2029 324.7516 311.5769 337.9263 304.6027 344.9006  
## Apr 2029 300.9179 287.7427 314.0931 280.7682 321.0676  
## May 2029 331.8842 318.7089 345.0594 311.7344 352.0340  
## Jun 2029 367.7883 354.6130 380.9635 347.6385 387.9381  
## Jul 2029 412.1076 398.9323 425.2828 391.9578 432.2574  
## Aug 2029 403.5983 390.4231 416.7736 383.4485 423.7481  
## Sep 2029 351.9325 338.7573 365.1077 331.7827 372.0823  
## Oct 2029 320.2860 307.1108 333.4612 300.1362 340.4357  
## Nov 2029 314.1845 301.0096 327.3594 294.0352 334.3338  
## Dec 2029 346.5053 333.3321 359.6784 326.3587 366.6518

# Plot the forecast values against the actual test data demand

plot(forecasted\_values, main = "ARIMA Forecast vs Actual Demand", ylab = "Demand Forecast (TWh)", xlab = "Year", col = "blue", lwd = 2)  
lines(test\_data, col = "red", lwd = 2)



#Train and fit generation data

train\_size2 <- floor(0.8 \* length(data2\_ts))  
train\_data2 <- window(data2\_ts, end = c(2019, 12))  
test\_data2 <- window(data2\_ts, start = c(2020, 1))

# Summary of the training and test data for generation

summary(train\_data2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 286.9 314.1 334.2 341.5 360.9 417.5

summary(test\_data2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 283.0 324.6 342.3 352.0 369.2 438.8

# Fit ARIMA model for generation data

fit2 <- auto.arima(train\_data2)

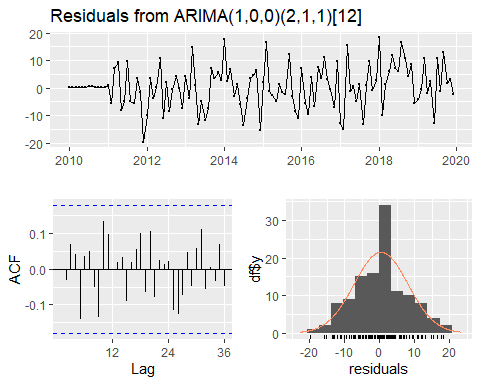
# View ARIMA model summary for generation data

summary(fit2)

## Series: train\_data2   
## ARIMA(1,0,0)(2,1,1)[12]   
##   
## Coefficients:  
## ar1 sar1 sar2 sma1  
## 0.4172 0.1917 -0.2764 -0.8802  
## s.e. 0.0882 0.1290 0.1098 0.2607  
##   
## sigma^2 = 67.41: log likelihood = -388.66  
## AIC=787.32 AICc=787.91 BIC=800.73  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.423173 7.643647 5.866416 0.1001415 1.710069 0.6441891  
## ACF1  
## Training set 0.0009469745

# Check residuals of the fitted model for generation data

checkresiduals(fit2)



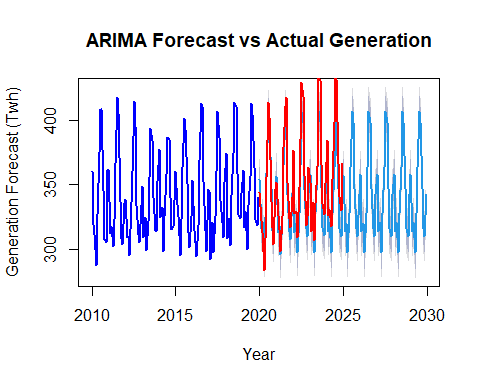
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,0)(2,1,1)[12]  
## Q\* = 17.049, df = 20, p-value = 0.6498  
##   
## Model df: 4. Total lags used: 24

# Forecast using the fitted model for generation data

forecasted\_values2 <- forecast(fit2, h = 120)

# Plot the forecasted values against the actual test data for generation

plot(forecasted\_values2, main = "ARIMA Forecast vs Actual Generation", ylab = "Generation Forecast (Twh)", xlab = "Year", col = "blue", lwd = 2)  
lines(test\_data2, col = "red", lwd = 2)



# Summary of forecasted\_values$mean

print(summary(forecasted\_values$mean))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 299.7 318.5 339.3 345.9 363.3 412.5

# Summary of test\_data

print(summary(test\_data))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 286.2 326.3 344.9 354.6 371.5 441.3

# T-test of forecasted values against actual values demand

t\_test\_demand <- t.test(forecasted\_values$mean, test\_data)  
if (t\_test\_demand$p.value < 0.05) {  
 print("The forecasted demand values are significantly different from the actual values.")  
} else {  
 print("The forecasted demand values are not significantly different from the actual values.")  
}

## [1] "The forecasted demand values are not significantly different from the actual values."

# Results of t-test

print(t\_test\_demand)

##   
## Welch Two Sample t-test  
##   
## data: forecasted\_values$mean and test\_data  
## t = -1.4638, df = 103.06, p-value = 0.1463  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -20.617521 3.107063  
## sample estimates:  
## mean of x mean of y   
## 345.8819 354.6372

# T-test of forecasted values against actual values for generation

t\_test\_generation <- t.test(forecasted\_values2$mean, test\_data2)  
if (t\_test\_generation$p.value < 0.05) {  
 print("The forecasted generation values are significantly different from the actual values.")  
} else {  
 print("The forecasted generation values are not significantly different from the actual values.")  
}

## [1] "The forecasted generation values are not significantly different from the actual values."

# Results of t-test

print(t\_test\_generation)

##   
## Welch Two Sample t-test  
##   
## data: forecasted\_values2$mean and test\_data2  
## t = -1.7983, df = 102.57, p-value = 0.07507  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -22.372318 1.094611  
## sample estimates:  
## mean of x mean of y   
## 341.3208 351.9597

# T-Test if predicted values of generation are significantly different from demand

t\_test\_comparison <- t.test(forecasted\_values$mean, forecasted\_values2$mean)  
if (t\_test\_comparison$p.value < 0.05) {  
 print("The forecasted demand and generation values are significantly different.")  
} else {  
 print("The forecasted demand and generation values are not significantly different.")  
}

## [1] "The forecasted demand and generation values are not significantly different."

# Results of t-test

print(t\_test\_comparison)

##   
## Welch Two Sample t-test  
##   
## data: forecasted\_values$mean and forecasted\_values2$mean  
## t = 1.0512, df = 237.94, p-value = 0.2942  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.986357 13.108605  
## sample estimates:  
## mean of x mean of y   
## 345.8819 341.3208