Climate

Elise Boos, Kelsey Husted, Taro Katayama

2023-04-11

Regional Weather Forecasting

Objectives

Forecasting temperatures is useful for energy usage projections in the future. Additionally, analyzing climate-related data at national-level is important since climate change may impact certain regions more significantly than others. Discrepancies between regions can potentially result in energy allotment adjustments.

Method

Temperature data was obtained from the International Energy Agency website and was readily available to download at the grid, national, and sub-national level. Daily and monthly resolution data was provided ranging from 2000 to the present. After importing and wrangling the data, a time series variable was created. Finally, a variety of models were fit to the data to find the closest fit for forecasting purposes.

(Discussion and Limitations at the bottom of rmd)

Load Data

```
require(tidyverse)
require(tseries)
require(forecast)
require(lubridate)
require(ggplot2)
require(Kendall)
library(kableExtra)
temp <- read.csv("./Data_Raw/Temperature_Data.csv")
```

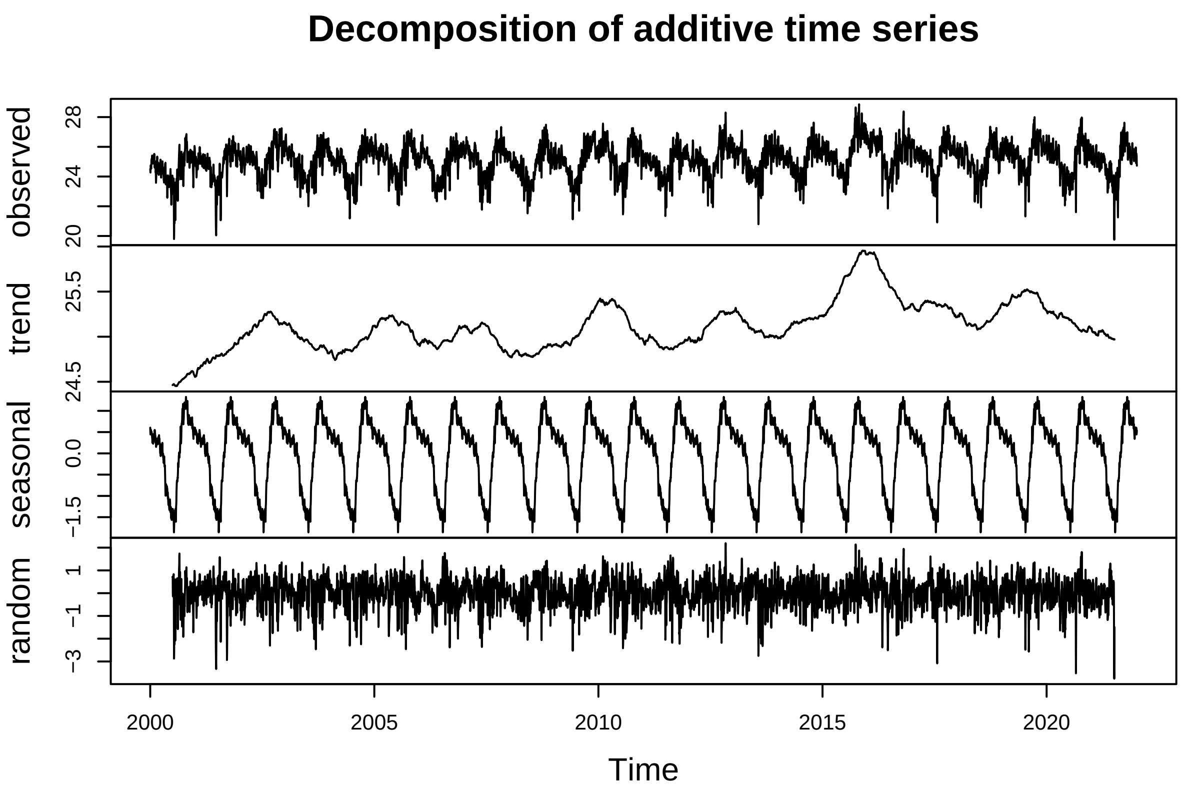
Tidy data

```
temp_processed<- temp%>%
  mutate(Date = as.Date(Date, format = "%d-%m-%y"))%>%
  rename(Temp= Temperaturedaily)%>%
  arrange(Date, Country)%>%
  select(Date, Country, Temp)
```

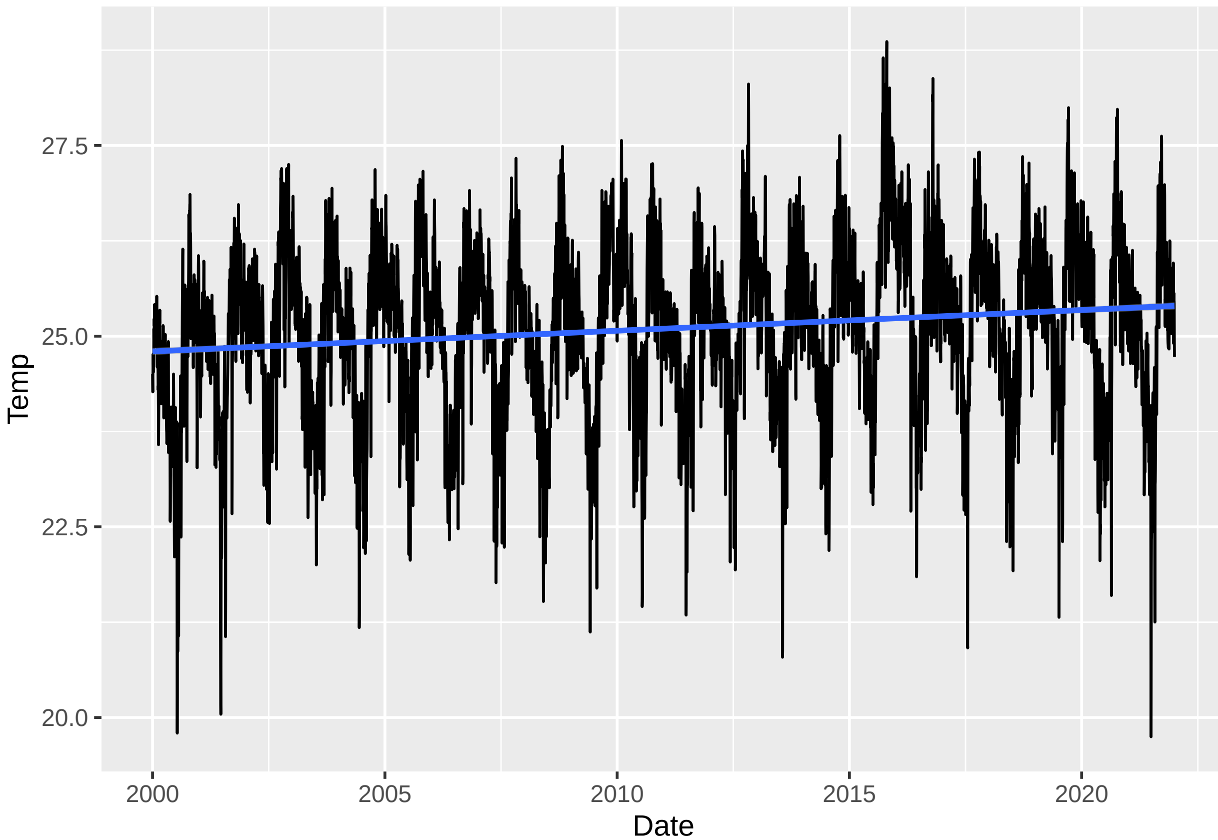
Time series transformation (BRA)

```
#Filter for Brazil
#Temperature
BRA_temp <- temp_processed%>%
  filter(Country == 'BRA')
```

```r
fday <- first(BRA_temp$Date)

#Time series transformation ts()
ts_temp_BRA <- ts(BRA_temp[,3], frequency = 365, start = c(year(fday), month(fday), day(fday)))

#Time series transformation msts()
ts_temp_BRA2 <- msts(BRA_temp[,3], seasonal.periods =c(7,365.25),
                     start=c(year(fday), month(fday), day(fday)))

#Decompose
decompose_temp_BRA <- decompose(ts_temp_BRA, type = "additive")
plot(decompose_temp_BRA)
```




```r
ggplot(BRA_temp, aes(x=Date, y =Temp))+
  geom_line()+
  geom_smooth(method = lm)
```

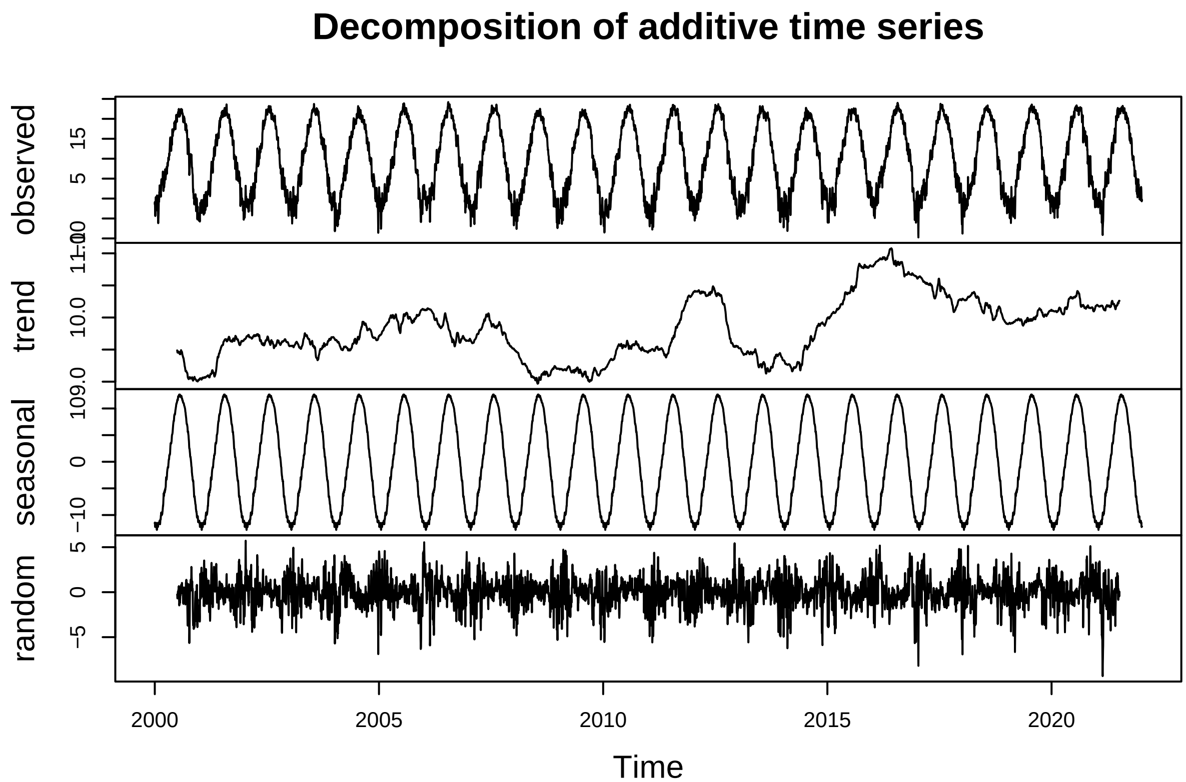



```
summary(SeasonalMannKendall(ts_temp_BRA))
```

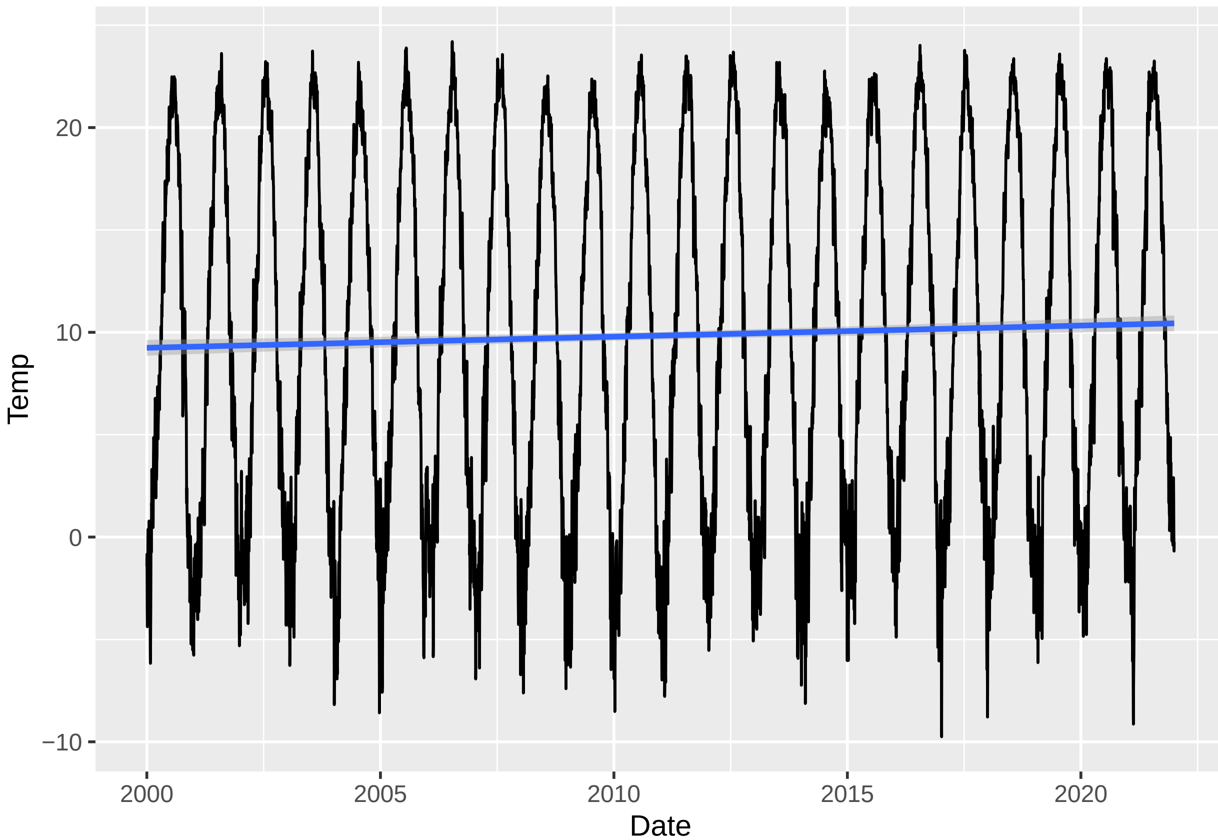
```
## Score =  13277 , Var(Score) = 460104.3
## denominator =  84446.99
## tau = 0.157, 2-sided pvalue =< 2.22e-16
```

Time series transformation (USA)

```
#Filter for USA
#Temperature
USA_temp <- temp_processed%>%
  filter(Country == 'USA')


#Time series transformation
ts_temp_USA <- ts(USA_temp[,3], frequency = 365, start = c(2000,1))

#Time series transformation msts()
ts_temp_USA2 <- msts(USA_temp[,3], seasonal.periods =c(7,365.25),
                           start=c(year(fday), month(fday), day(fday)))
#Decompose
decompose_temp_USA <- decompose(ts_temp_USA, type = "additive")
plot(decompose_temp_USA)
```




```
ggplot(USA_temp, aes(x=Date, y =Temp))+
  geom_line()+
  geom_smooth(method = lm)
```

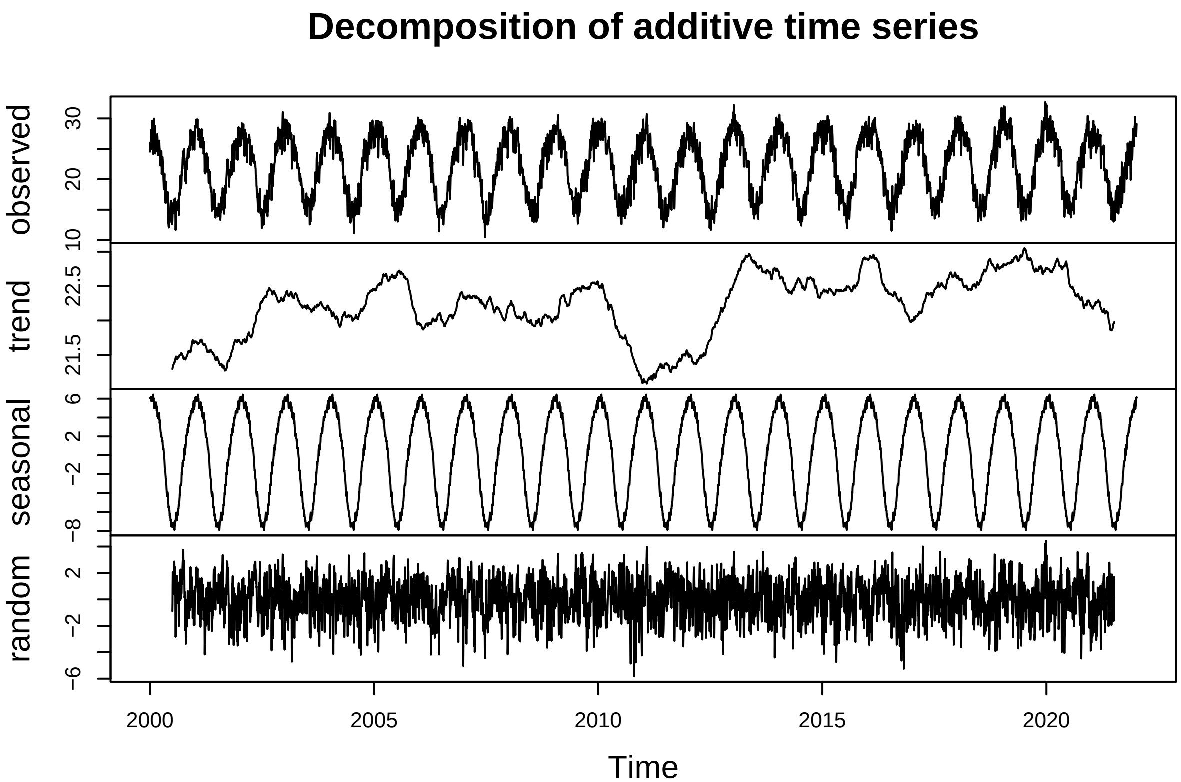



```
summary(SeasonalMannKendall(ts_temp_USA))
```

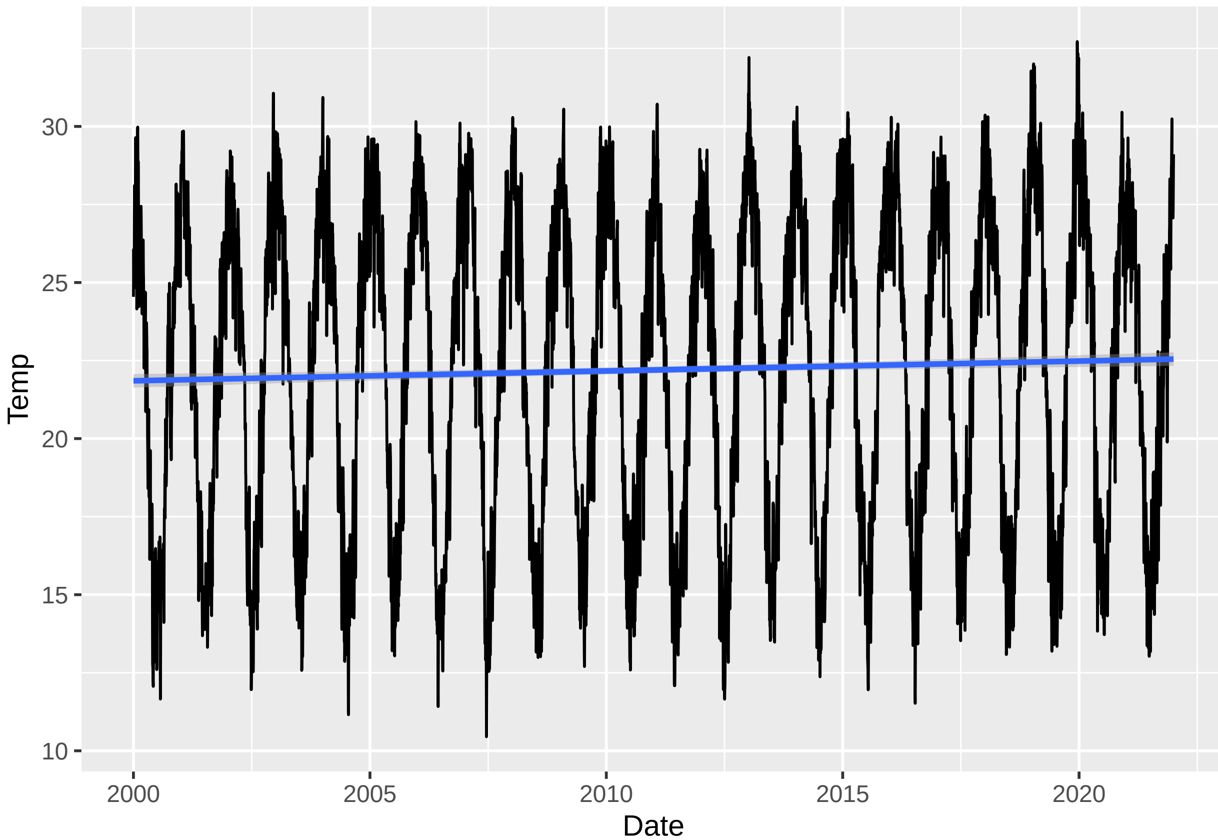
```
## Score =  10871 , Var(Score) = 460104.3
## denominator =  84446.99
## tau = 0.129, 2-sided pvalue =< 2.22e-16
```

###Time series transformation (AUS)

```
#Filter for Australia
#Temperature
AUS_temp <- temp_processed%>%
  filter(Country == 'AUS')

#Time series transformation
ts_temp_AUS <- ts(AUS_temp[,3], frequency = 365, start = c(2000,1))

#Time series transformation msts()
ts_temp_AUS2 <- msts(AUS_temp[,3], seasonal.periods =c(7,365.25),
                           start=c(year(fday), month(fday), day(fday)))

#Decompose
decompose_temp_AUS <- decompose(ts_temp_AUS, type = "additive")
plot(decompose_temp_AUS)
```




```
ggplot(AUS_temp, aes(x=Date, y =Temp))+
  geom_line()+
  geom_smooth(method = lm)
```




```
summary(SeasonalMannKendall(ts_temp_AUS))
```

```
## Score =  8085 , Var(Score) = 460104.3
## denominator =  84446.99
## tau = 0.0957, 2-sided pvalue =< 2.22e-16
```

Temperature Forecasting Test (BRA)

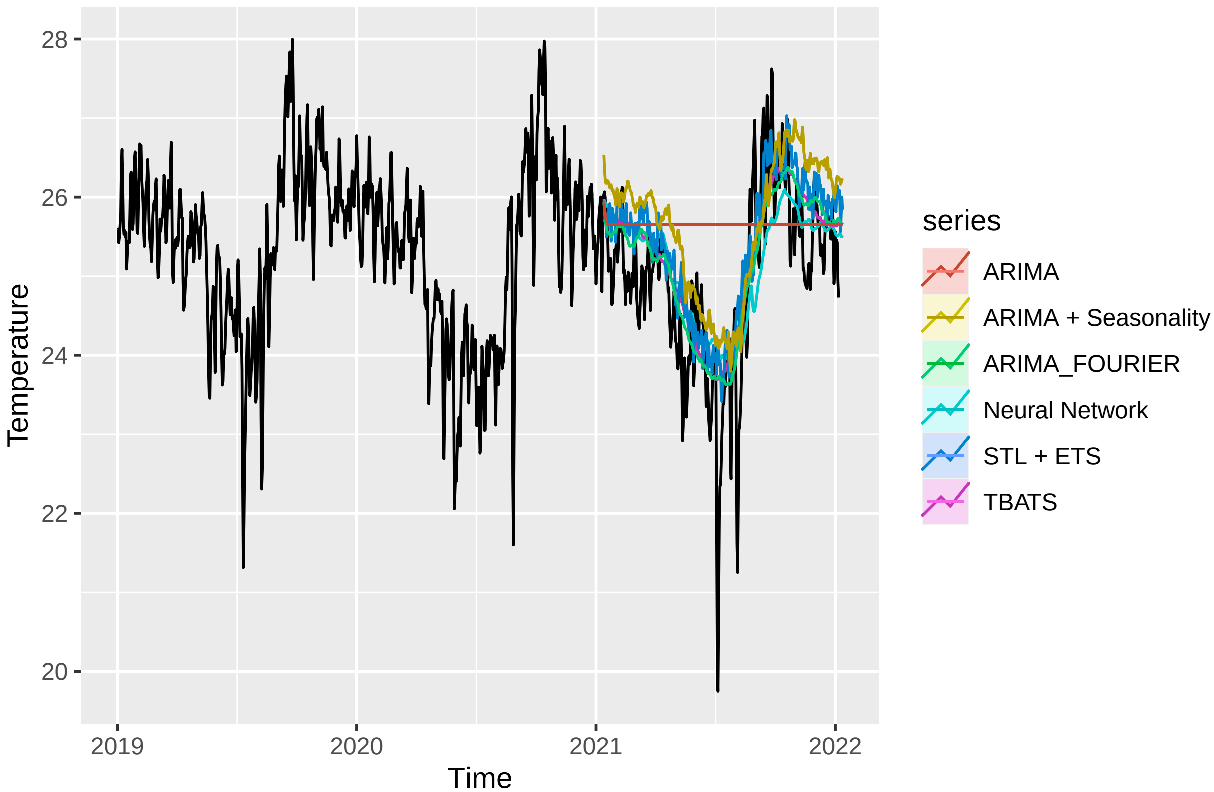
```
#AUTO ARIMA
ts_temp_BRA_sub <-window(ts_temp_BRA, start = c(2000,1), end = c(2021,12))

ts_actual_forecast<- window(ts_temp_AUS, start = c(2022,1), end = c(2022,12))

arima <- auto.arima(ts_temp_BRA_sub,seasonal=FALSE,
                             lambda=0)

ARIMA_forecast <- forecast(arima,h=365)

#ARIMA with seasonality added back

seasonality<- decompose_temp_BRA$seasonal[1:365]

for_and_seasonality<- ARIMA_forecast$mean + seasonality

ts_temp_BRA2_sub <-window(ts_temp_BRA2, start = c(2000,1), end = c(2021,12))
### ARIMA + Fourier Terms
ARIMA_Four <- auto.arima(ts_temp_BRA2_sub,
```

```
                         seasonal=FALSE,
                         lambda=0,
                         xreg=fourier(ts_temp_BRA2_sub,
                                      K=c(2,12)))
ARIMA_Four_for <- forecast(ARIMA_Four,
                           xreg=fourier(ts_temp_BRA2_sub,
                                        K=c(2,12),
                                        h=365),
                           h=365)


#### NEURAL NETWORK
NN_fit <- nnetar(ts_temp_BRA2_sub,p=1,P=1)

#5 year forecast
NN_for <- forecast(NN_fit, h=365)


### TBATS
TBATS_fit <- tbats(ts_temp_BRA2_sub)

TBATS_for <- forecast(TBATS_fit, h=365)

### STL + ETS
ETS_fit <- stlf(ts_temp_BRA2_sub, h=365)

a_score <- accuracy(ARIMA_forecast, ts_actual_forecast)
a2_score <- accuracy(ARIMA_Four_for, ts_actual_forecast)
NN_score <- accuracy(NN_for, ts_actual_forecast)
TBATS_score <- accuracy(TBATS_for, ts_actual_forecast)
ETS_score <- accuracy(ETS_fit, ts_actual_forecast)

scores <- as.data.frame(rbind(a_score, a2_score, NN_score, TBATS_score, ETS_score))
row.names(scores) <- c("ARIMA_training", "ARIMA_test", "ARIMA FOURIER_training","ARIMA FOURIER_TEST",
"NN_training", "NN_test","TBATS_training", "TBATS_test", "STL + ETS_training", "STL + ETS_test")
scores
```

```
##                                 ME      RMSE       MAE           MPE      MAPE
## ARIMA_training        0.0013044353 0.2933913 0.2212043 -0.0056612608 0.8863639
## ARIMA_test            2.2592309108 2.3852210 2.2592309  8.0256965136 8.0256965
## ARIMA FOURIER_training 0.0030258725 0.2861840 0.2156922 -0.0006063623 0.8643516
## ARIMA FOURIER_TEST    2.2055680468 2.3284069 2.2055680  7.8351177842 7.8351178
## NN_training           0.0001341801 0.3473734 0.2574637 -0.0192514493 1.0328259
## NN_test               2.3924406673 2.5133237 2.3924407  8.5027938626 8.5027939
## TBATS_training        0.0017179633 0.2860879 0.2149848 -0.0064489408 0.8616030
## TBATS_test            2.2629036266 2.3834168 2.2629036  8.0404807140 8.0404807
## STL + ETS_training    0.0001832022 0.3019373 0.2280123 -0.0073721275 0.9163490
## STL + ETS_test        1.9109080994 2.0470942 1.9109081  6.7804998859 6.7804999
##                            MASE          ACF1 Theil's U
## ARIMA_training        0.2877435 -9.221182e-03        NA
## ARIMA_test            2.9388165  5.655160e-01  5.027844
## ARIMA FOURIER_training 0.2805732 -9.872358e-03        NA
## ARIMA FOURIER_TEST    2.8690116  5.640541e-01  4.906504
## NN_training           0.3349098  3.964328e-01        NA
```

```
## NN_test                3.1120963  5.677864e-01  5.295469
## TBATS_training         0.2796531 -6.969618e-05        NA
## TBATS_test             2.9435940  5.657727e-01  5.020404
## STL + ETS_training     0.2965993  3.874788e-01        NA
## STL + ETS_test         2.4857168  5.421575e-01  4.344678
```

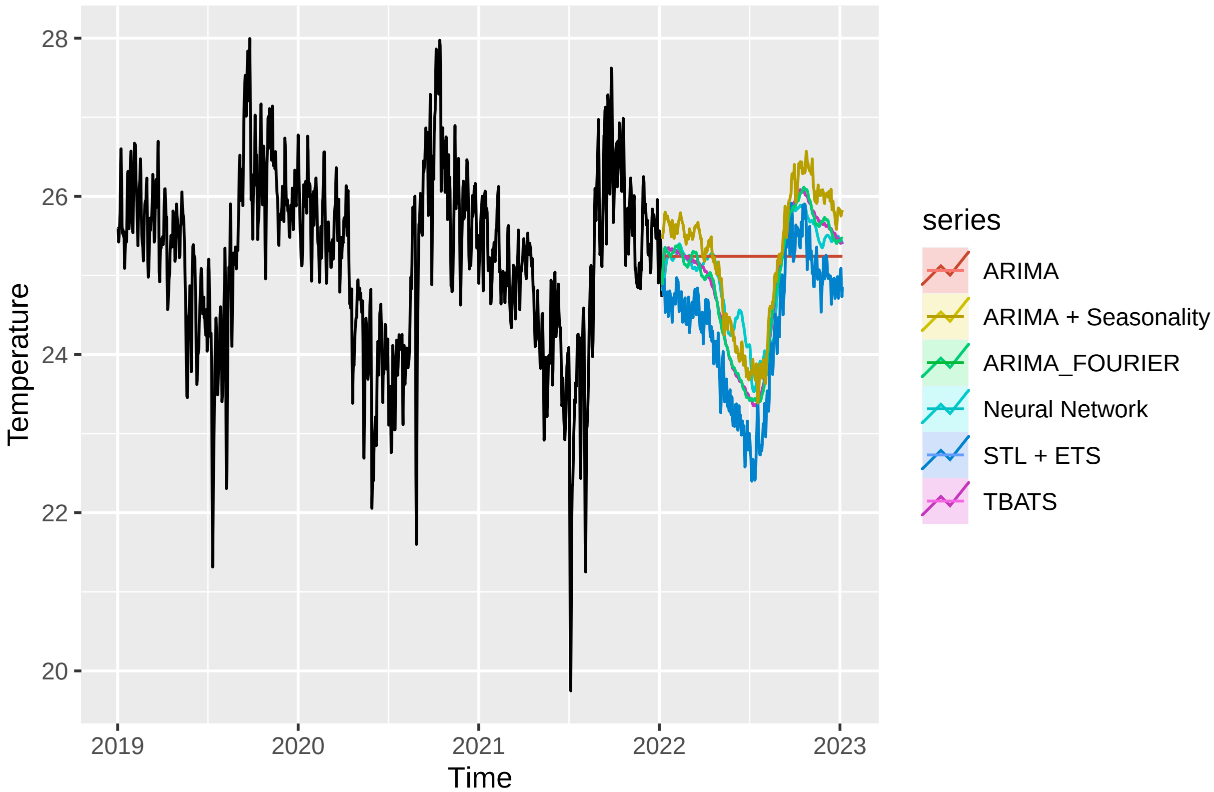
```
autoplot(window(ts_temp_BRA, start = c(2019,1))) +
  autolayer(NN_for, series="Neural Network",PI=FALSE)+
  autolayer(TBATS_for,series="TBATS",PI=FALSE)+
  autolayer(ETS_fit,series="STL + ETS",PI=FALSE)+
  autolayer(ARIMA_Four_for, series="ARIMA_FOURIER",PI=FALSE)+
  autolayer(ARIMA_forecast, series="ARIMA",PI=FALSE)+
  autolayer(for_and_seasonality, series = "ARIMA + Seasonality")+
  ylab("Temperature")
```



Temperature Forecasting (BRA)

```
#AUTO ARTIMA
arima <- auto.arima(ts_temp_BRA,seasonal=FALSE,
                             lambda=0)

ARIMA_forecast <- forecast(arima,h=365)

#ARIMA with seasonality added back

seasonality<- decompose_temp_BRA$seasonal[1:365]

for_and_seasonality<- ARIMA_forecast$mean + seasonality
```

```r
### ARIMA + Fourier Terms
ARIMA_Four <- auto.arima(ts_temp_BRA2,
                         seasonal=FALSE,
                         lambda=0,
                         xreg=fourier(ts_temp_BRA2,
                                      K=c(2,12)))
ARIMA_Four_for <- forecast(ARIMA_Four,
                           xreg=fourier(ts_temp_BRA2,
                                        K=c(2,12),
                                        h=365),
                           h=365)

#### NEURAL NETWORK
NN_fit <- nnetar(ts_temp_BRA2,p=1,P=1)

#5 year forecast
NN_for <- forecast(NN_fit, h=365)


### TBATS
TBATS_fit <- tbats(ts_temp_BRA2)

TBATS_for <- forecast(TBATS_fit, h=365)

### STL + ETS
ETS_fit <- stlf(ts_temp_BRA2, h=365)


###PLOT
autoplot(window(ts_temp_BRA, start = c(2019,1))) +
  autolayer(NN_for, series="Neural Network",PI=FALSE)+
  autolayer(ARIMA_forecast, series="ARIMA",PI=FALSE)+
  autolayer(TBATS_for,series="TBATS",PI=FALSE)+
  autolayer(ETS_fit,series="STL + ETS",PI=FALSE)+
  autolayer(ARIMA_Four_for, series="ARIMA_FOURIER",PI=FALSE)+
  autolayer(for_and_seasonality, series="ARIMA + Seasonality")+
  ylab("Temperature")
```




```r
#create accuracy test for each forecast
a_score <- accuracy(ARIMA_forecast)
a2_score <- accuracy(ARIMA_Four_for)
NN_score <- accuracy(NN_for)
TBATS_score <- accuracy(TBATS_for)
ETS_score <- accuracy(ETS_fit)


#create accuracy test matrix
scores <- as.data.frame(rbind(a_score, a2_score, NN_score, TBATS_score, ETS_score))
row.names(scores) <- c("ARIMA", "ARIMA FOURIER","NN", "TBATS", "STL + ETS")
scores
```

```
##                         ME      RMSE       MAE          MPE      MAPE
## ARIMA          1.155240e-03 0.2933117 0.2209756 -0.006195613 0.8857841
## ARIMA FOURIER  2.548022e-03 0.2861532 0.2153629 -0.002532060 0.8633544
## NN            -4.406821e-05 0.3475254 0.2570008 -0.020056795 1.0316260
## TBATS          1.016324e-03 0.2861009 0.2150811 -0.008752422 0.8623018
## STL + ETS      4.462651e-05 0.3015785 0.2274305 -0.007920840 0.9146430
##                    MASE          ACF1
## ARIMA         0.2888160 -1.035830e-02
## ARIMA FOURIER 0.2814801 -1.092407e-02
## NN            0.3359009  3.996922e-01
## TBATS         0.2811117  7.736454e-05
## STL + ETS     0.2972525  3.902598e-01
```

```r
kbl(scores,
      caption = "Forecast Accuracy for Brazil Temperature Data",
      digits = array(5,ncol(scores))) %>%
  kable_styling(full_width = FALSE, position = "center", html_font="Cambria")
```

Table 1: Forecast Accuracy for Brazil Temperature Data

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
ARIMA	0.00116	0.29331	0.22098	-0.00620	0.88578	0.28882	-0.01036
ARIMA FOURIER	0.00255	0.28615	0.21536	-0.00253	0.86335	0.28148	-0.01092
NN	-0.00004	0.34753	0.25700	-0.02006	1.03163	0.33590	0.39969
TBATS	0.00102	0.28610	0.21508	-0.00875	0.86230	0.28111	0.00008
STL + ETS	0.00004	0.30158	0.22743	-0.00792	0.91464	0.29725	0.39026

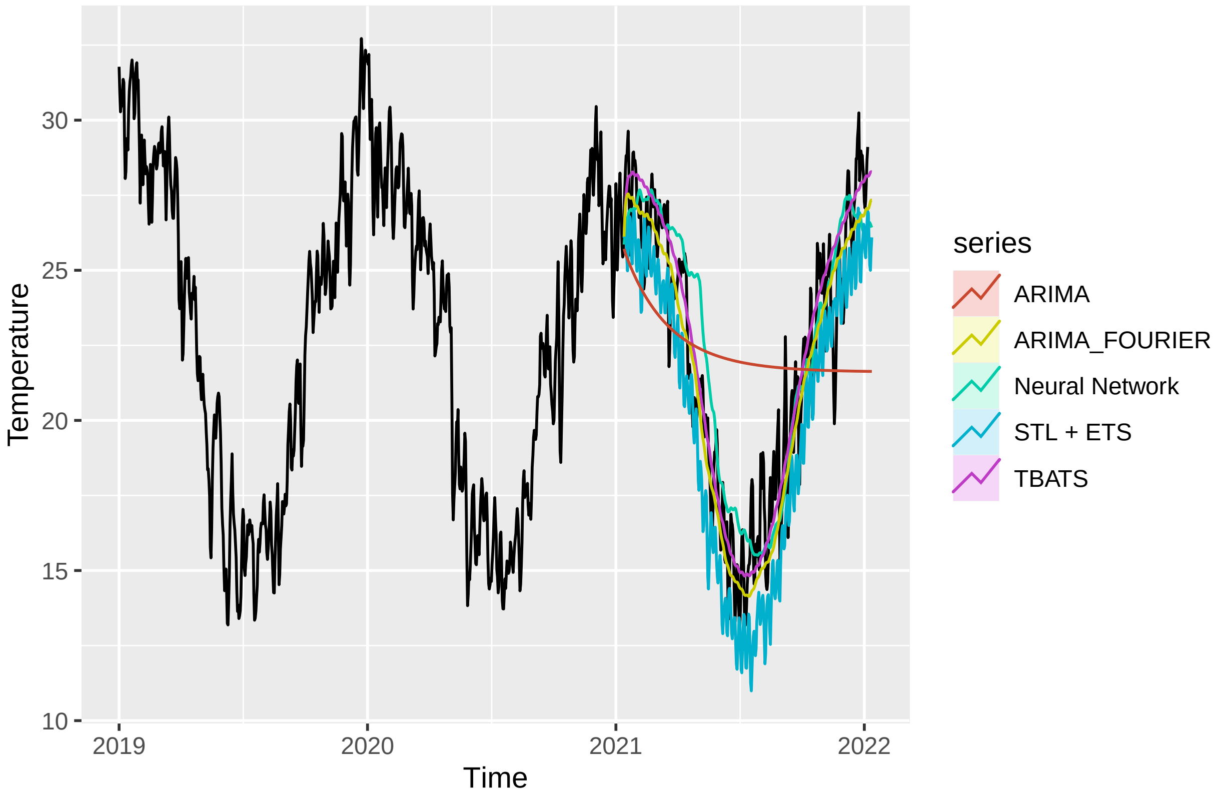
Temperature Forecasting Test (AUS)

```
#AUTO ARTIMA
ts_temp_AUS_sub <-window(ts_temp_AUS, start = c(2000,1), end = c(2021,12))

ts_actual_forecast<- window(ts_temp_AUS, start = c(2022,1), end = c(2022,12))

arima_AUS <- auto.arima(ts_temp_AUS_sub,seasonal=FALSE,
                             lambda=0)

ARIMA_forecast_AUS <- forecast(arima_AUS,h=365)


ts_temp_AUS2_sub <-window(ts_temp_AUS2, start = c(2000,1), end = c(2021,12))
### ARIMA + Fourier Terms
ARIMA_Four_AUS <- auto.arima(ts_temp_AUS2_sub,
                         seasonal=FALSE,
                         lambda=0,
                         xreg=fourier(ts_temp_AUS2_sub,
                                      K=c(2,12)))
ARIMA_Four_AUS_for <- forecast(ARIMA_Four_AUS,
                           xreg=fourier(ts_temp_AUS2_sub,
                                        K=c(2,12),
                                        h=365),
                           h=365)

#### NEURAL NETWORK
NN_fit_AUS <- nnetar(ts_temp_AUS2_sub,p=1,P=1)

#5 year forecast
NN_for_AUS <- forecast(NN_fit_AUS, h=365)


### TBATS
TBATS_fit_AUS <- tbats(ts_temp_AUS2_sub)

TBATS_for_AUS <- forecast(TBATS_fit_AUS, h=365)

### STL + ETS
ETS_fit_AUS <- stlf(ts_temp_AUS2_sub, h=365)
```

```r
###ACCURACY
a_score_AUS <- accuracy(ARIMA_forecast_AUS, ts_actual_forecast)
a2_score_AUS <- accuracy(ARIMA_Four_AUS_for, ts_actual_forecast)
NN_score_AUS <- accuracy(NN_for_AUS, ts_actual_forecast)
TBATS_score_AUS <- accuracy(TBATS_for_AUS, ts_actual_forecast)
ETS_score_AUS <- accuracy(ETS_fit_AUS, ts_actual_forecast)


scores_AUS <- as.data.frame(rbind(a_score_AUS, a2_score_AUS, NN_score_AUS, TBATS_score_AUS, ETS_score_AU
row.names(scores_AUS) <- c("ARIMA_training", "ARIMA_test", "ARIMA FOURIER_training","ARIMA FOURIER_TEST
"NN_training", "NN_test","TBATS_training", "TBATS_test", "STL + ETS_training", "STL + ETS_test")
scores_AUS
```

```
##                                  ME      RMSE       MAE         MPE      MAPE
## ARIMA_training         1.045432e-02 0.6545550 0.5010596 -0.04860487  2.364876
## ARIMA_test             6.278368e+00 6.3248621 6.2783676 22.43545372 22.435454
## ARIMA FOURIER_training  1.059049e-02 0.6275716 0.4785515 -0.04113338  2.260244
## ARIMA FOURIER_TEST     9.252352e-01 1.1690698 0.9252352  3.24806023  3.248060
## NN_training            1.143055e-04 0.7521522 0.5850538 -0.12977070  2.769129
## NN_test                1.519491e+00 1.7004432 1.5194911  5.37373873  5.373739
## TBATS_training         4.506664e-03 0.6210546 0.4725630 -0.06724323  2.232773
## TBATS_test            -1.824220e-01 0.7495312 0.6922706 -0.72449958  2.485058
## STL + ETS_training     6.368236e-05 0.6391497 0.5010362 -0.05026571  2.368057
## STL + ETS_test         1.750246e+00 1.7958331 1.7502455  6.24618461  6.246185
##                             MASE          ACF1 Theil's U
## ARIMA_training         0.2932795  0.0053281237        NA
## ARIMA_test             3.6748449  0.5655169044 13.025078
## ARIMA FOURIER_training 0.2801051  0.0059867330        NA
## ARIMA FOURIER_TEST     0.5415573  0.5525998989  2.512021
## NN_training            0.3424428  0.4126621689        NA
## NN_test                0.8893863  0.5814544448  3.640688
## TBATS_training         0.2765999 -0.0001569192        NA
## TBATS_test             0.4051988  0.5540654717  1.532724
## STL + ETS_training     0.2932658  0.4011044859        NA
## STL + ETS_test         1.0244511  0.3842713382  3.875409
```

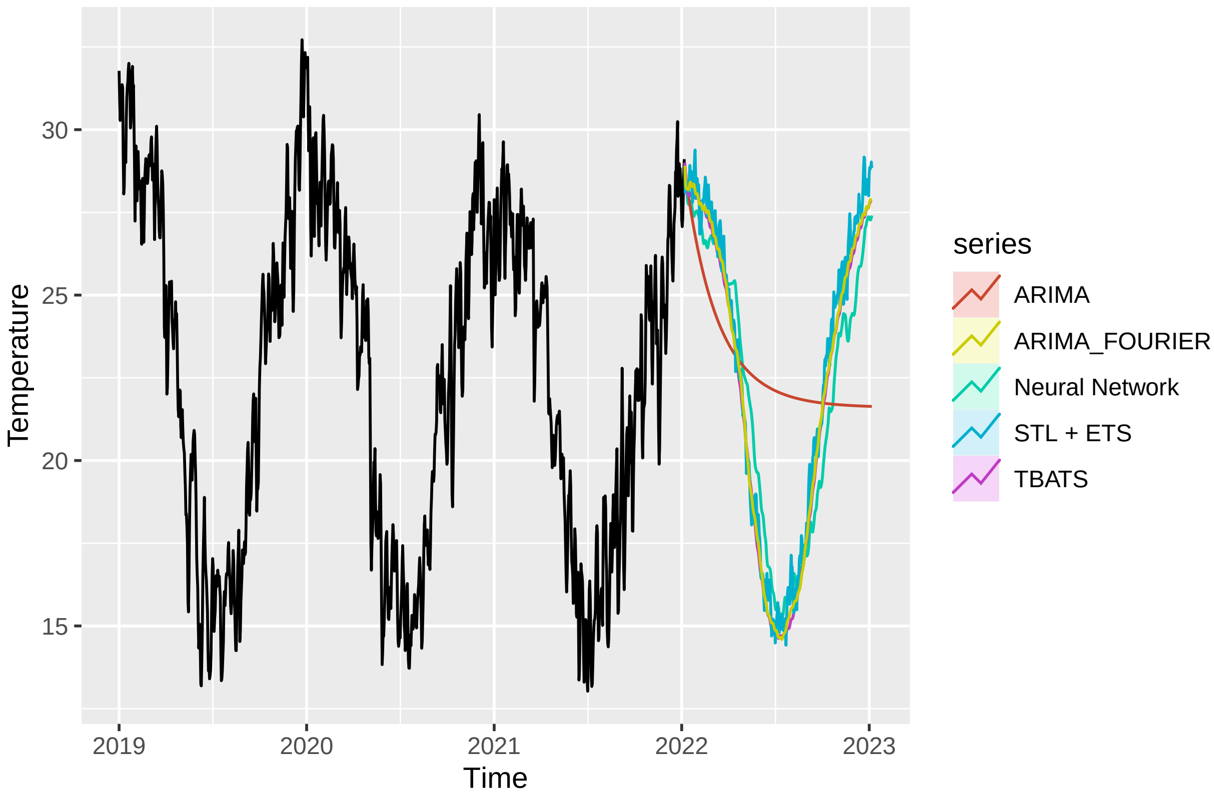
```r
autoplot(window(ts_temp_AUS, start = c(2019,1))) +
  autolayer(NN_for_AUS, series="Neural Network",PI=FALSE)+
  autolayer(TBATS_for_AUS,series="TBATS",PI=FALSE)+
  autolayer(ETS_fit_AUS,series="STL + ETS",PI=FALSE)+
  autolayer(ARIMA_Four_AUS_for, series="ARIMA_FOURIER",PI=FALSE)+
  autolayer(ARIMA_forecast_AUS, series="ARIMA",PI=FALSE)+
  ylab("Temperature")
```



Temperature Forecasting (AUS)

```
#AUTO ARTIMA
arima_AUS <- auto.arima(ts_temp_AUS,seasonal=FALSE,
                             lambda=0)

ARIMA_forecast <- forecast(arima_AUS,h=365)

### ARIMA + Fourier Terms
ARIMA_Four <- auto.arima(ts_temp_AUS2,
                         seasonal=FALSE,
                         lambda=0,
                         xreg=fourier(ts_temp_AUS2,
                                      K=c(2,12)))
ARIMA_Four_for <- forecast(ARIMA_Four,
                           xreg=fourier(ts_temp_AUS2,
                                        K=c(2,12),
                                        h=365),
                           h=365)

#### NEURAL NETWORK
NN_fit <- nnetar(ts_temp_AUS2,p=1,P=1)

#5 year forecast
NN_for <- forecast(NN_fit, h=365)


### TBATS
```

```r
TBATS_fit <- tbats(ts_temp_AUS2)

TBATS_for <- forecast(TBATS_fit, h=365)

### STL + ETS
ETS_fit <- stlf(ts_temp_AUS2, h=365)


###PLOT
autoplot(window(ts_temp_AUS, start = c(2019,1))) +
  autolayer(NN_for, series="Neural Network",PI=FALSE)+
  autolayer(ARIMA_forecast, series="ARIMA",PI=FALSE)+
  autolayer(TBATS_for,series="TBATS",PI=FALSE)+
  autolayer(ETS_fit,series="STL + ETS",PI=FALSE)+
  autolayer(ARIMA_Four_for, series="ARIMA_FOURIER",PI=FALSE)+
  ylab("Temperature")
```




```r
a_score <- accuracy(ARIMA_forecast)
a2_score <- accuracy(ARIMA_Four_for)
NN_score <- accuracy(NN_for)
TBATS_score <- accuracy(TBATS_for)
ETS_score <- accuracy(ETS_fit)


scores <- as.data.frame(rbind(a_score, a2_score, NN_score, TBATS_score, ETS_score))
row.names(scores) <- c("ARIMA", "ARIMA FOURIER","NN", "TBATS", "STL + ETS")
scores
```

```
##                            ME       RMSE         MAE          MPE       MAPE        MASE
```

Table 2: Forecast Accuracy for Australia Temperature Data

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
ARIMA	0.01085	0.65790	0.50299	-0.04814	2.37658	0.29274	0.00551
ARIMA FOURIER	0.01126	0.63016	0.47987	-0.03926	2.26806	0.27929	0.00546
NN	-0.00023	0.75754	0.58888	-0.13404	2.79267	0.34273	0.41332
TBATS	0.00356	0.62380	0.47348	-0.07152	2.23921	0.27557	-0.00058
STL + ETS	0.00037	0.64305	0.50365	-0.05035	2.38454	0.29313	0.40302

```
## ARIMA          0.0108483121 0.6578969 0.5029867 -0.04814271 2.376581 0.2927408
## ARIMA FOURIER  0.0112589481 0.6301645 0.4798677 -0.03926399 2.268059 0.2792854
## NN            -0.0002317135 0.7575353 0.5888759 -0.13404098 2.792671 0.3427288
## TBATS          0.0035642093 0.6237952 0.4734809 -0.07152363 2.239215 0.2755683
## STL + ETS      0.0003747740 0.6430456 0.5036472 -0.05035214 2.384539 0.2931252
##                        ACF1
## ARIMA          0.0055065098
## ARIMA FOURIER  0.0054569276
## NN             0.4133165621
## TBATS         -0.0005765243
## STL + ETS      0.4030246418
```

```
kbl(scores,
      caption = "Forecast Accuracy for Australia Temperature Data",
      digits = array(5,ncol(scores))) %>%
  kable_styling(full_width = FALSE, position = "center", html_font="Cambria")
```

Temperature Forecasting Test (USA)

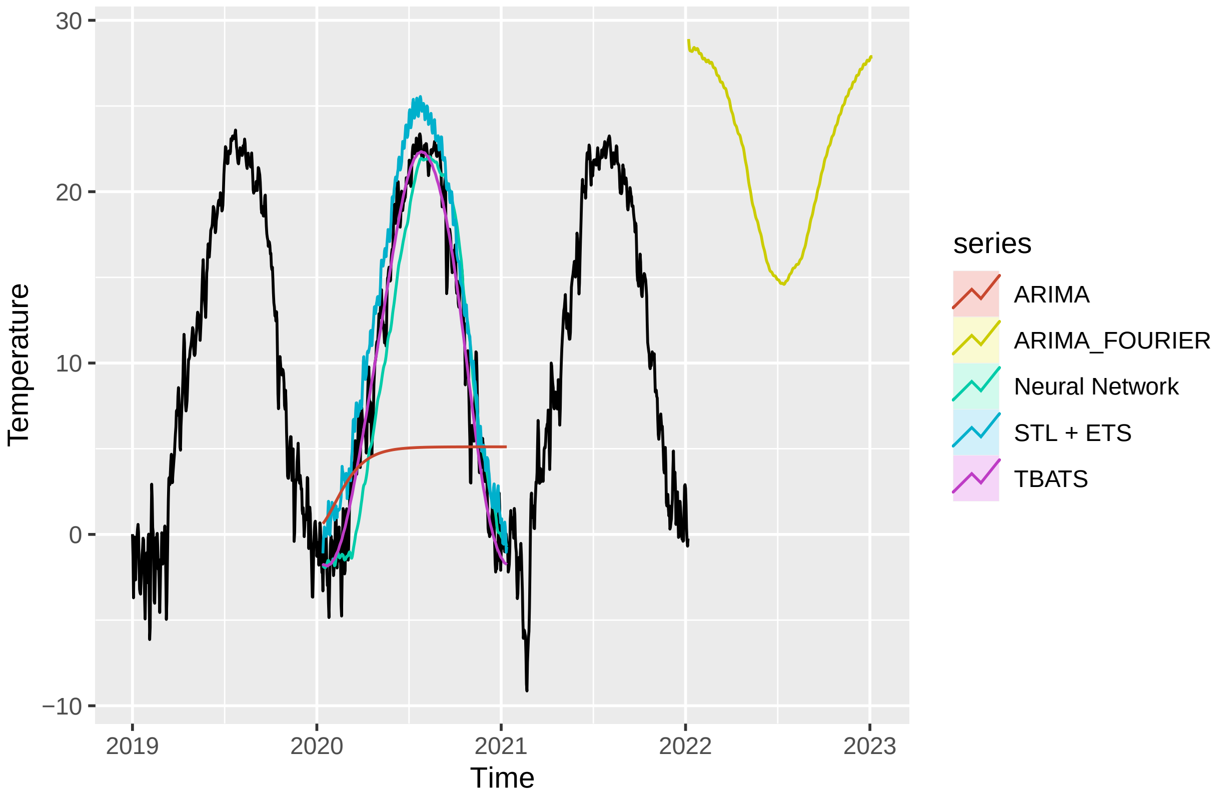
```
#AUTO ARTIMA
ts_temp_sub <-window(ts_temp_USA, start = c(2000,1), end = c(2020,12))

ts_actual_forecast<- window(ts_temp_USA, start = c(2021,1), end = c(2021,12))

arima <- auto.arima(ts_temp_sub,seasonal=FALSE,
                             lambda=0)

ARIMA_forecast <- forecast(arima,h=365)


ts_temp_USA2_sub <-window(ts_temp_USA2, start = c(2000,1), end = c(2020,12))
### ARIMA + Fourier Terms
ARIMA_Four <- auto.arima(ts_temp_USA2_sub,
                         seasonal=FALSE,
                         lambda=0,
                         xreg=fourier(ts_temp_USA2_sub,
                                      K=c(2,12)))
ARIMA_Four_USA_for <- forecast(ARIMA_Four,
                           xreg=fourier(ts_temp_USA2_sub,
                                        K=c(2,12),
                                        h=365),
                           h=365)

#### NEURAL NETWORK
```

```r
NN_fit <- nnetar(ts_temp_USA2_sub,p=1,P=1)

#5 year forecast
NN_for <- forecast(NN_fit, h=365)


### TBATS
TBATS_fit <- tbats(ts_temp_USA2_sub)

TBATS_for <- forecast(TBATS_fit, h=365)

### STL + ETS
ETS_fit <- stlf(ts_temp_USA2_sub, h=365)



###ACCURACY
a_score <- accuracy(ARIMA_forecast, ts_actual_forecast)
a2_score <- accuracy(ARIMA_Four_USA_for, ts_actual_forecast)
NN_score <- accuracy(NN_for, ts_actual_forecast)
TBATS_score <- accuracy(TBATS_for, ts_actual_forecast)
ETS_score <- accuracy(ETS_fit, ts_actual_forecast)


scores <- as.data.frame(rbind(a_score, a2_score, NN_score, TBATS_score, ETS_score))
row.names(scores) <- c("ARIMA_training", "ARIMA_test", "ARIMA FOURIER_training","ARIMA FOURIER_TEST",
"NN_training", "NN_test","TBATS_training", "TBATS_test", "STL + ETS_training", "STL + ETS_test")
scores
```

```
##                                   ME      RMSE       MAE         MPE       MAPE
## ARIMA_training          0.4022624071 0.8698973 0.7231891  -34.801494   47.58856
## ARIMA_test             -5.5345643077 5.5514387 5.5345643 -825.267057 3415.80246
## ARIMA FOURIER_training  0.0386925167 0.8500600 0.5433636  -30.618116   41.31104
## ARIMA FOURIER_TEST     -0.8934519773 0.9898870 0.8934520   10.814904  372.43783
## NN_training             0.0003649412 0.8361880 0.6184782  -13.734446   78.92531
## NN_test                -0.0485560631 0.4785913 0.4268295  196.441656  317.94835
## TBATS_training          0.0016804763 0.6462131 0.4736942   -3.767177   54.93815
## TBATS_test              1.1587086063 1.2408577 1.1587086  397.742535 1036.99997
## STL + ETS_training      0.0005901145 0.6897986 0.4996793   -4.229391   58.54421
## STL + ETS_test         -0.3753723755 0.9264636 0.7098821  316.885630  478.74808
##                             MASE          ACF1 Theil's U
## ARIMA_training         0.4143182  0.5428876718        NA
## ARIMA_test             3.1707761  0.2737916820 18.610848
## ARIMA FOURIER_training 0.3112954  0.2800202745        NA
## ARIMA FOURIER_TEST     0.5118625  0.2841178717  2.564677
## NN_training            0.3543289  0.4542305321        NA
## NN_test                0.2445325  0.2580209173  1.510431
## TBATS_training         0.2713815 -0.0007495747        NA
## TBATS_test             0.6638292  0.2211255087  5.308990
## STL + ETS_training     0.2862684  0.0406829081        NA
## STL + ETS_test         0.4066946  0.4651736022  1.916516
```

```r
autoplot(window(ts_temp_USA, start = c(2019,1))) +
  autolayer(NN_for, series="Neural Network",PI=FALSE)+
```

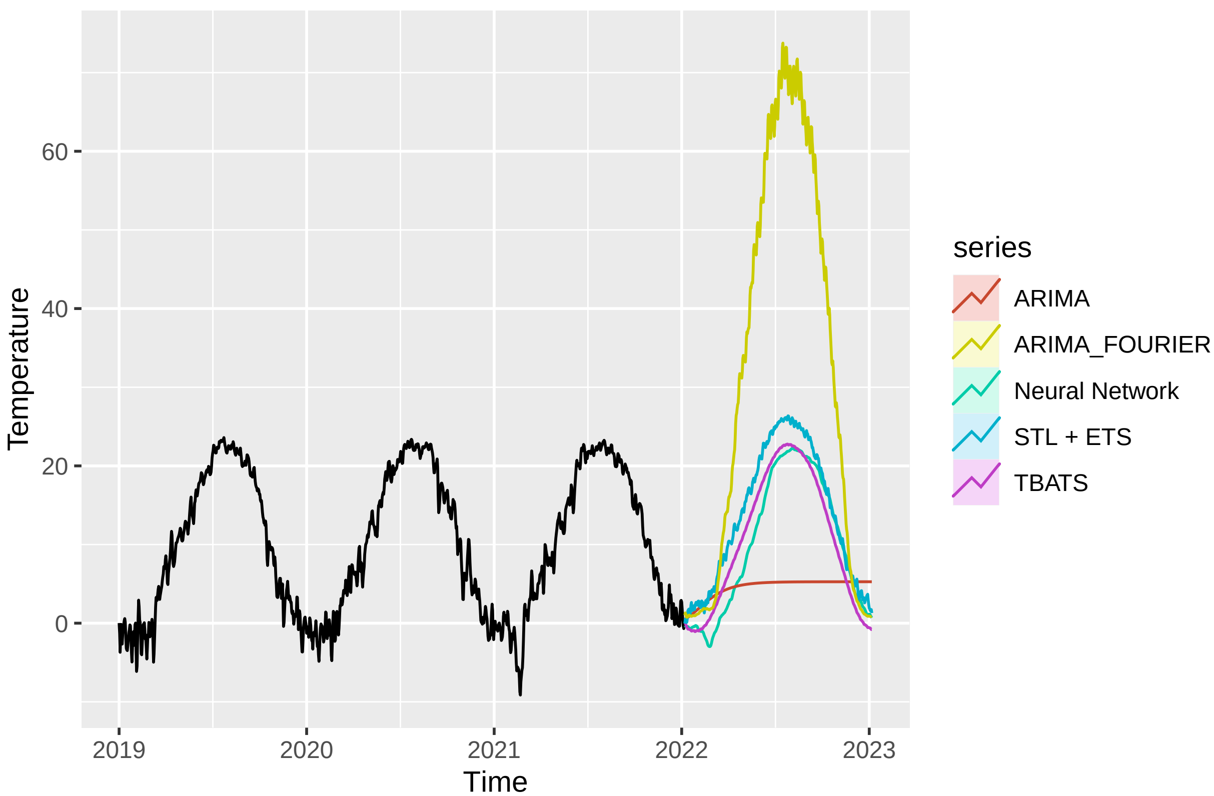
```
  autolayer(TBATS_for,series="TBATS",PI=FALSE)+
  autolayer(ETS_fit,series="STL + ETS",PI=FALSE)+
  autolayer(ARIMA_Four_for, series="ARIMA_FOURIER",PI=FALSE)+
  autolayer(ARIMA_forecast, series="ARIMA",PI=FALSE)+
  ylab("Temperature")
```



##Temperature Forecasting (USA)

```
#AUTO ARTIMA
arima <- auto.arima(ts_temp_USA,seasonal=FALSE,
                             lambda=0)

ARIMA_forecast <- forecast(arima,h=365)

### ARIMA + Fourier Terms
ARIMA_Four <- auto.arima(ts_temp_USA2,
                         seasonal=FALSE,
                         lambda=0,
                         xreg=fourier(ts_temp_USA2,
                                      K=c(2,12)))
ARIMA_Four_for <- forecast(ARIMA_Four,
                           xreg=fourier(ts_temp_USA2,
                                        K=c(2,12),
                                        h=365),
                           h=365)

#### NEURAL NETWORK
NN_fit <- nnetar(ts_temp_USA2,p=1,P=1)
```

```r
#5 year forecast
NN_for <- forecast(NN_fit, h=365)


### TBATS
TBATS_fit <- tbats(ts_temp_USA2)

TBATS_for <- forecast(TBATS_fit, h=365)

### STL + ETS
ETS_fit <- stlf(ts_temp_USA2, h=365)


###PLOT
autoplot(window(ts_temp_USA, start = c(2019,1))) +
  autolayer(NN_for, series="Neural Network",PI=FALSE)+
  autolayer(ARIMA_forecast, series="ARIMA",PI=FALSE)+
  autolayer(TBATS_for,series="TBATS",PI=FALSE)+
  autolayer(ETS_fit,series="STL + ETS",PI=FALSE)+
  autolayer(ARIMA_Four_for, series="ARIMA_FOURIER",PI=FALSE)+
  ylab("Temperature")
```




```r
a_score <- accuracy(ARIMA_forecast)
a2_score <- accuracy(ARIMA_Four_for)
NN_score <- accuracy(NN_for)
TBATS_score <- accuracy(TBATS_for)
ETS_score <- accuracy(ETS_fit)
```

Table 3: Forecast Accuracy for USA Temperature Data

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
ARIMA	0.39384	0.86979	0.72211	-32.76981	45.49040	0.41926	0.54020
ARIMA FOURIER	0.03909	0.85077	0.53775	-28.64724	39.30227	0.31223	0.26817
NN	0.00020	0.83360	0.61724	-13.09836	75.83975	0.35838	0.45527
TBATS	0.00029	0.64294	0.47190	-3.54110	53.48570	0.27399	-0.00006
STL + ETS	0.00055	0.68253	0.49678	-4.87451	57.25018	0.28844	0.03670

```r
scores <- as.data.frame(rbind(a_score, a2_score, NN_score, TBATS_score, ETS_score))
row.names(scores) <- c("ARIMA", "ARIMA FOURIER","NN", "TBATS", "STL + ETS")
scores
```

```
##                         ME      RMSE       MAE        MPE     MAPE      MASE
## ARIMA         0.3938400223 0.8697872 0.7221058 -32.769808 45.49040 0.4192647
## ARIMA FOURIER 0.0390854742 0.8507674 0.5377509 -28.647243 39.30227 0.3122256
## NN            0.0002005105 0.8336036 0.6172439 -13.098363 75.83975 0.3583804
## TBATS         0.0002864740 0.6429401 0.4719009  -3.541096 53.48570 0.2739922
## STL + ETS     0.0005496258 0.6825322 0.4967838  -4.874514 57.25018 0.2884396
##                        ACF1
## ARIMA          5.402049e-01
## ARIMA FOURIER  2.681685e-01
## NN             4.552667e-01
## TBATS         -5.846698e-05
## STL + ETS      3.670387e-02
```

```r
kbl(scores,
      caption = "Forecast Accuracy for USA Temperature Data",
      digits = array(5,ncol(scores))) %>%
  kable_styling(full_width = FALSE, position = "center", html_font="Cambria")
```

Discussion

The positive linear temperature trends for Brazil, United States, and Australia indicate the presence of climate change effects (Seasonal Mann-Kendall: p-values < 0.05). Temperature trends for the three countries follow a similar visual trend No one country looks to be increasing in temperature quicker than the other two.

The ARIMA + Fourier and TBATS model are generally the best predictors for temperature changes over time as displayed in the error terms. However, looking at the forecasts, the STL + ETS looks to do best in all three countries. This is not reflected in the error terms. One explanation for this might be because the accuracy function only uses the training set to provide the accuracy scores. It cannot get the accuracy score for forecasted values that do not have observed values yet. For this reason, the test sets might provide a more accurate representation of which forecasting model performs the best. In the test accuracy function, we were able to provide the test set error along with the training set error to provide the most accurate error scores.

Future steps might include looking into exogenous variables or events related to temperature such as precipitation, El Nino, and La Nina. Precipitation and temperature are closely linked and might provide a more accurate forecast model.

Furthermore, it might be interesting to model precipitation or model more localized temperature trends in the future. Particularly, climate change has changed precipitation regimes, and would be vital to understand these trends in the context of climate risks and vulnerability. We looked at temperature trends over a national scale, however temperature trends may be masked at this granularity. Future forecasting studies may delve into localized temperature predictions.

Limitations

Temperatures were generalized for the entire countries analyzed in this project. This approach can cause issues since regional climates in countries can contrast significantly. Countries were also selected bases on personal biases. Although daily data was available, only monthly temperatures were forecasted. Daily temperature would allow for a finer tuned model that would pick up weekly trends. Error terms were compared amongst forecasts, which only gave us a limited understanding of how good the forecast was.