

# Climate Data Forecasting - Atmospheric $CO_2$ Concentration / Temperature / Precipitation

Wolfgang Vollmer

2023-02-20

## Contents

<b>1</b>	<b>Forecasting of Potsdam - Temperature Climate Analysis</b>	<b>2</b>
1.1	Stationarity and differencing . . . . .	2
1.1.1	Ljung-Box Test - independence/white noise of the time series . . . . .	3
1.1.2	Unitroot KPSS Test - fix number of seasonal differences/differences required . . .	3
1.1.3	ACF Plots of Differences . . . . .	4
1.1.4	Time Series, ACF and PACF Plots of Differences - for ARIMA p, q check . . . . .	5
<b>2</b>	<b>ExponenTial Smoothing (ETS) Forecasting Models</b>	<b>6</b>
2.1	ETS Models and their componentes . . . . .	7
2.1.1	Residual Accuracy with one-step-ahead fitted residuals - check RMSE, MAE . . .	8
2.1.2	Ljung-Box Test - independence/white noise of the forecasts residuals . . . . .	9
2.1.3	ETS Models - components of ETS(A,N,A), ETS(A,A,A), ETS(A,Ad,A), models .	9
2.1.4	Forecast Accuracy with Training/Test Data . . . . .	9
2.2	Forecasting with selected ETS model <ETS(A,A,A)> . . . . .	10
2.2.1	Forecast Plot of selected ETS model . . . . .	10
2.2.2	Residual Stationarity . . . . .	11
2.2.3	Histogram of forecast residuals with overlaid normal curve . . . . .	12
<b>3</b>	<b>ARIMA Forecasting Models - AutoRegressive-Integrated Moving Average</b>	<b>13</b>
3.1	Seasonal ARIMA models . . . . .	13
3.1.1	Residual Accuracy with one-step-ahead fitted residuals - check RMSE, MAE . . .	15
3.1.2	Ljung-Box Test - independence/white noise of the forecasts residuals . . . . .	15
3.1.3	Forecast Accuracy with Training/Test Data . . . . .	15
3.2	Temperature - Forecasting with selected ARIMA model <ARIMA(0,1,2)(0,1,2)[12]> . . .	16
3.2.1	Forecast Plot of selected ARIMA model . . . . .	16
3.2.2	Residual Stationarity . . . . .	17
3.2.3	Histogram of forecast residuals with overlaid normal curve . . . . .	18

<b>4</b>	<b>ARIMA vs ETS</b>	<b>19</b>
4.0.1	Comparing Residual and Forecast Accuracy of selected ETS and ARIMA model . . . . .	19
4.0.2	Forecast Plot of selected ETS and ARIMA model . . . . .	19
4.0.3	Ljung-Box Test - independence/white noise of the forecasts residuals . . . . .	21
<b>5</b>	<b>Yearly Data Forecasts with ARIMA and ETS</b>	<b>21</b>
5.0.1	Comparing Residual and Forecast Accuracy of selected ETS and ARIMA model . . . . .	21
5.0.2	Forecast Plot of selected ETS and ARIMA model . . . . .	22
5.0.3	Ljung-Box Test - independence/white noise of the forecasts residuals . . . . .	23
<b>6</b>	<b>Backup</b>	<b>23</b>

# 1 Forecasting of Potsdam - Temperature Climate Analysis

## 1.1 Stationarity and differencing

Stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times. On the other hand, a white noise series is stationary — it does not matter when you observe it, it should look much the same at any point in time.

Stationary time series will have no predictable patterns in the long-term. Time plots will show the series to be roughly horizontal (although some cyclic behaviour is possible), with constant variance.

If Time Series data with seasonality are non-stationary

- => first take a seasonal difference
- if seasonally differenced data appear are still non-stationary
- => take an additional first seasonal difference

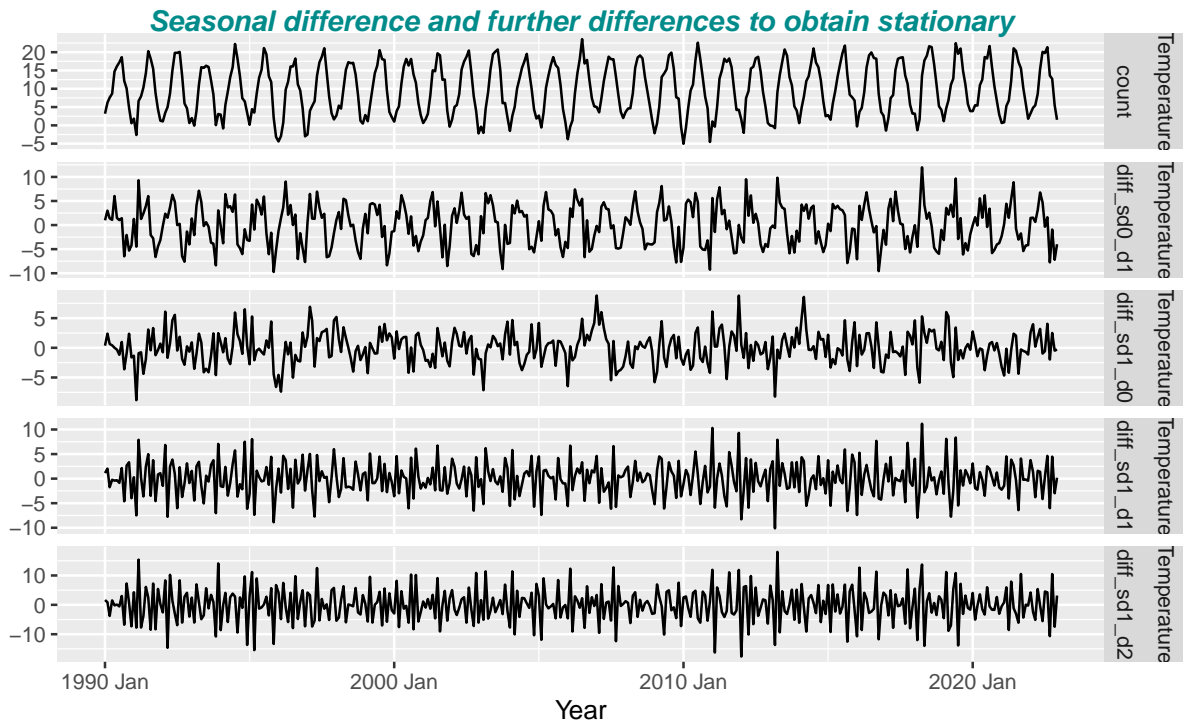
The model fit residuals have to be stationary. For good forecasting this has to be verified with residual diagnostics.

Essential:

- Residuals are uncorrelated
- The residuals have zero mean

Useful (but not necessary):

- The residuals have constant variance.
- The residuals are normally distributed.



### 1.1.1 Ljung-Box Test - independence/white noise of the time series

The Ljung-Box Test becomes important when checking independence/white noise of the forecasts residuals of the fitted ETS resp. ARIMA models. There we have to check whether the forecast errors are normally distributed with mean zero

Null Hypothesis of independence/white noise in a given time series

=>  $H_0$  to be rejected for  $p < \alpha = 0.05$

=> data in the given time series are dependent

=> even differenced data are dependent if  $p < \alpha = 0.05$

=> independence/white noise of residuals of fitted models to be verified

```
#> Ljung-Box test with (count), w/o differences
#> # A tibble: 1 x 3
#>   Measure    lb_stat lb_pvalue
#>   <fct>      <dbl>    <dbl>
#> 1 Temperature 5535.      0
#> Ljung-Box test on (difference(count, 12))
#> # A tibble: 1 x 3
#>   Measure    lb_stat lb_pvalue
#>   <fct>      <dbl>    <dbl>
#> 1 Temperature 86.2 3.05e-14
#> Ljung-Box test on (difference(count, 12) + difference())
#> # A tibble: 1 x 3
#>   Measure    lb_stat lb_pvalue
#>   <fct>      <dbl>    <dbl>
#> 1 Temperature 284.      0
```

### 1.1.2 Unitroot KPSS Test - fix number of seasonal differences/differences required

kpss test of stationary

Null Hypothesis of stationary in a given time series

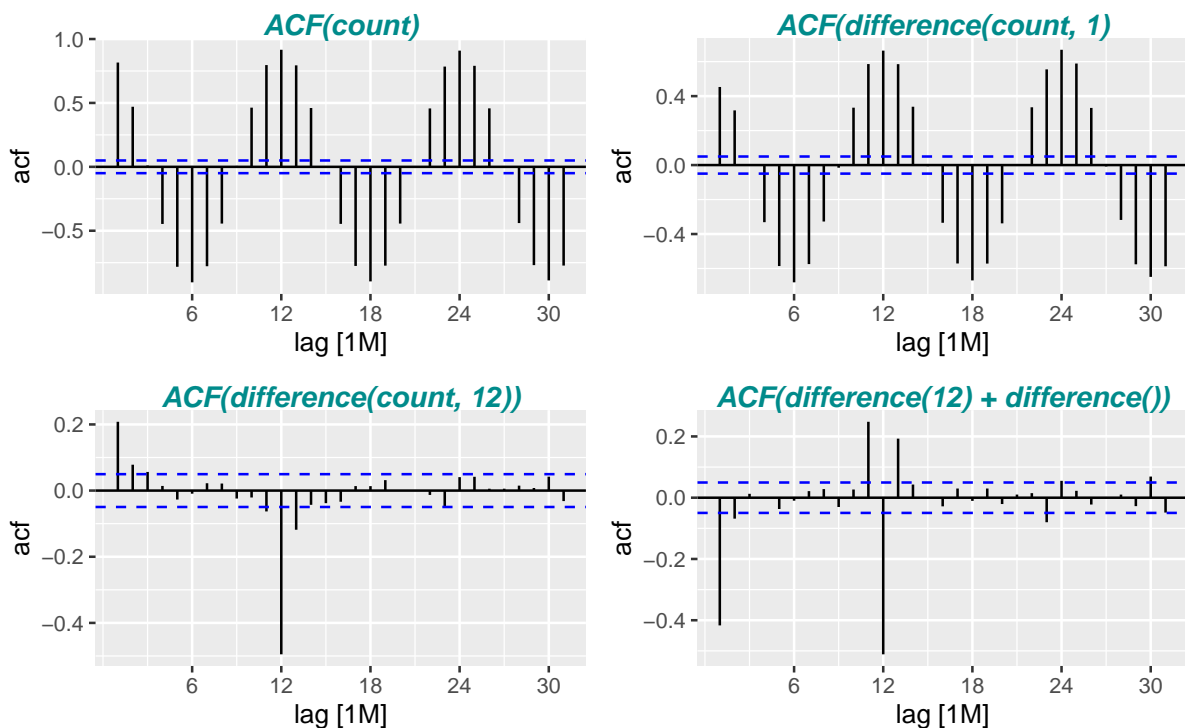
=>  $H_0$  to be rejected for  $p < \alpha = 0.05$

unitroot\_nsdiffs/ndiff provides minimum number of seasonal differences/differences required for a stationary series. First fix required seasonal differences and then apply ndiffs to the seasonally differenced data.

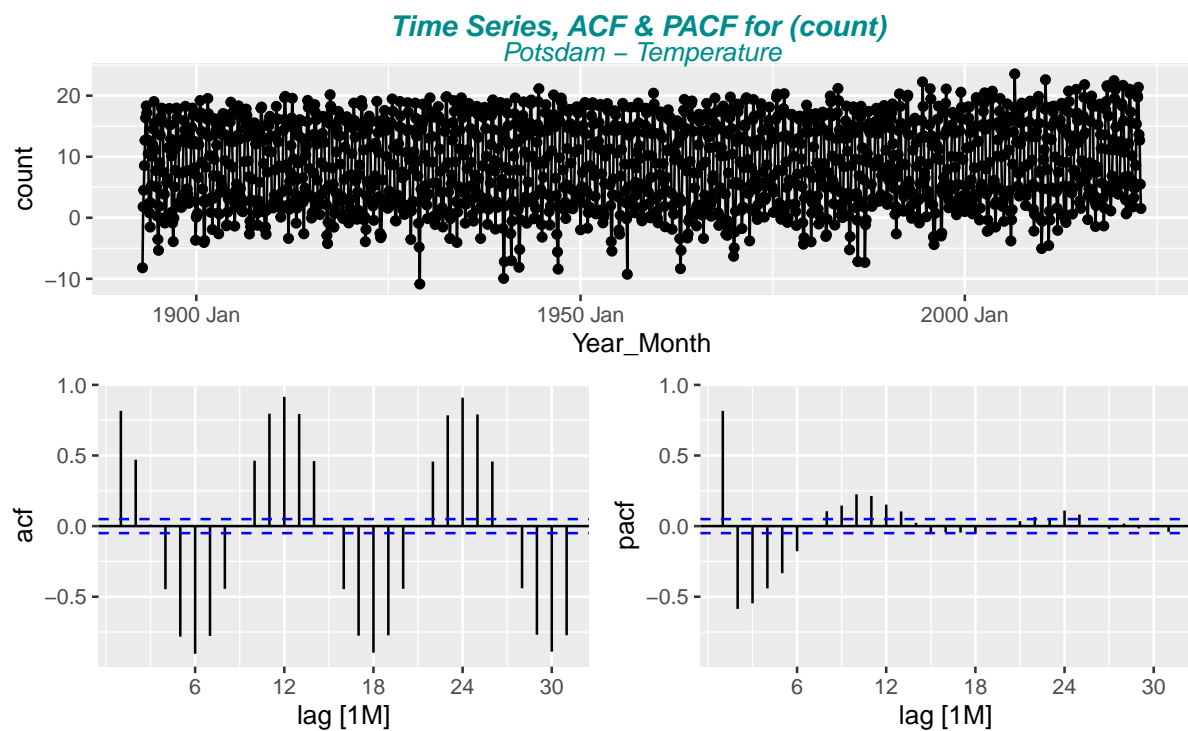
- returns 1 => for stationarity one seasonal difference resp. difference is required

```
#> ndiffs gives the number of differences required resp.
#> nsdiffs gives the number of seasonal differences required to make
#> a series stationary (test is based on the KPSS test)
#> kpss test, nsdiffs & ndiffs on (count), w/o differences
#> # A tibble: 1 x 5
#>   Measure      kpss_stat kpss_pvalue nsdiffs ndiffs
#>   <fct>         <dbl>         <dbl>   <int> <int>
#> 1 Temperature    0.613         0.0215     1     1
#> kpss test, nsdiffs & ndiffs on (difference(count, 12))
#> # A tibble: 1 x 5
#>   Measure      kpss_stat kpss_pvalue nsdiffs ndiffs
#>   <fct>         <dbl>         <dbl>   <int> <int>
#> 1 Temperature  0.00965         0.1         0     0
#> kpss test, nsdiffs & ndiffs on (difference(count, 12) %>% difference(1))
#> # A tibble: 1 x 5
#>   Measure      kpss_stat kpss_pvalue nsdiffs ndiffs
#>   <fct>         <dbl>         <dbl>   <int> <int>
#> 1 Temperature  0.00753         0.1         0     0
```

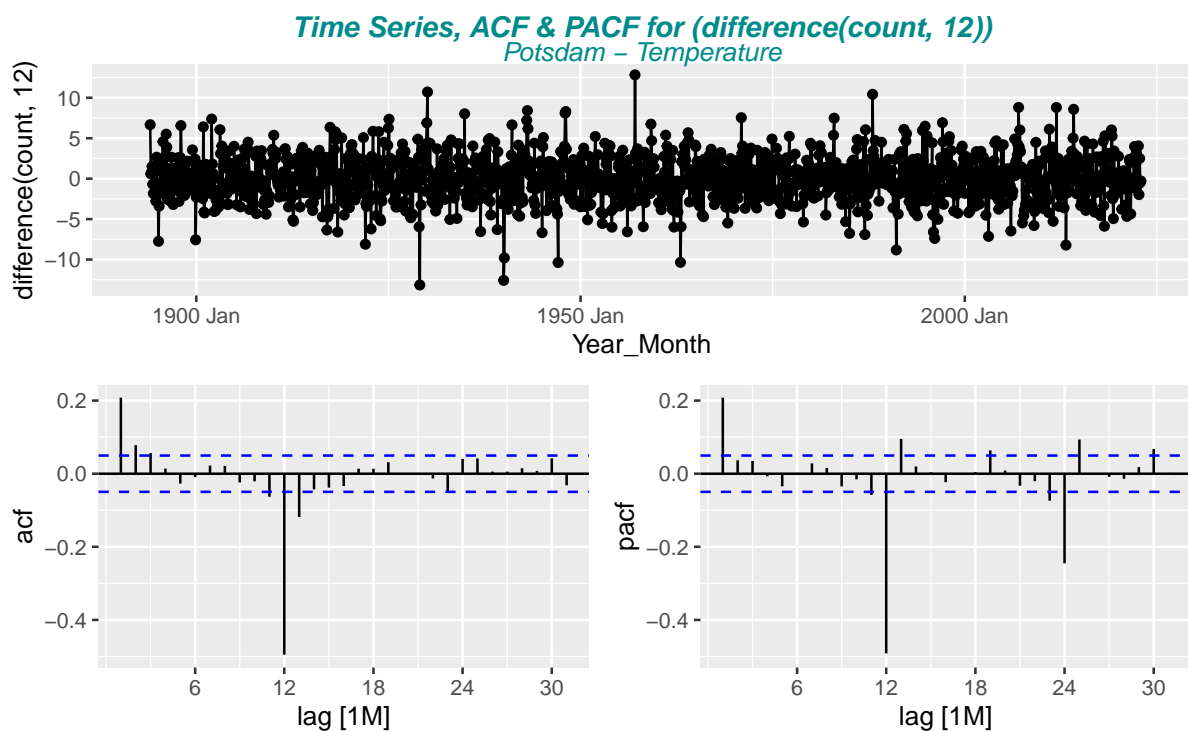
### 1.1.3 ACF Plots of Differences



#### 1.1.4 Time Series, ACF and PACF Plots of Differences - for ARIMA p, q check

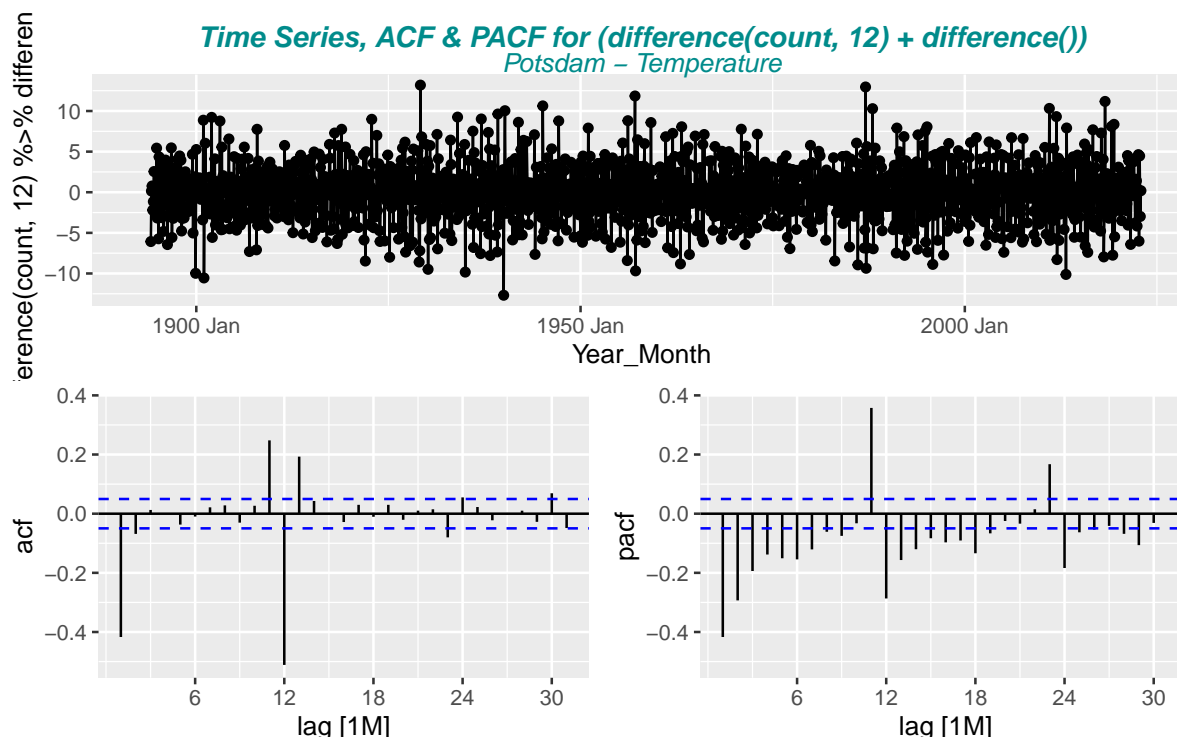


```
#> # A tibble: 1 x 2
#>   Sum   Mean
#>   <dbl> <dbl>
#> 1  34.4  0.0222
```



```
#> # A tibble: 1 x 2
#>   Sum   Mean
```

```
#> <dbl> <dbl>
#> 1 34.4 0.0222
```



```
#> # A tibble: 1 x 2
#>   Sum      Mean
#>   <dbl>   <dbl>
#> 1 -6.96 -0.00450
```

## 2 Exponential Smoothing (ETS) Forecasting Models

Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older.

The parameters are estimated by maximising the “likelihood”. The likelihood is the probability of the data arising from the specified model. AIC, AICc and BIC can be used here to determine which of the ETS models is most appropriate for a given time series (see output `glance(fit_ets)`).

The model selection is based on recognising key components of the time series (trend and seasonal) and the way in which these enter the smoothing method (e.g., in an additive, damped or multiplicative manner).

- Mauna Loa  $CO_2$  data best Models: ETS(M,A,A) & ETS(A,A,A)
- Basel Temperature data best Models: ETS(A,N,A), ETS(A,A,A), ETS(A,Ad,A) (close together). Best Forecast accuracy is with ETS(A,A,A), ETS(A,Ad,A).
- Basel Precipitation data best Models: ETS(A,N,A), ETS(A,Ad,A), ETS(A,A,A) (close together). Best Forecast accuracy is with ETS(A,A,A), ETS(A,Ad,A), ETS(A,N,A),

Trend term “N” for Basel Temperature/Precipitation corresponds to a “pure” exponential smoothing which results in a slope  $\beta = 0$ . This results in a forecast predicting a constant level. This does not fit to the result of the STL decomposition. Therefore best model choice is **ETS(A,A,A)**.

### Method Selection

*Error term:* either additive (“A”) or multiplicative (“M”).

Both methods provide identical point forecasts, but different prediction intervals and different likelihoods. AIC & BIC are able to select between the error types because they are based on likelihood.

Nevertheless, difference is for

- Mauna Loa  $CO_2$  not relevant and AIC/AICc/BIC values are only a little bit smaller for multiplicative errors. The prediction interval plots are fully overlapping.
- Basel Temperature AIC/AICc/BIC of additive error types are much better than the multiplicative ones.
- Basel Precipitation AIC/AICc/BIC of additive error types are much better than the multiplicative ones.

Note: For Basel Temperature and Precipitation Forecast plots the models ETS\_MAdA, ETS\_MMA, ETS\_MMA, ETS\_MNA are to be taken out since forecasts with multiplicative errors are exploding (forecast > 3 years impossible !!)

Therefore finally **Error term** = “A” is chosen in general.

*Trend term*: either none (“N”), additive (“A”), multiplicative (“M”) or damped variants (“Ad”, “Md”).

Note: Mauna Loa  $CO_2$  model ETS(A,Ad,A) fit plot shows to strong damping. For Basel Temperature model ETS(A,N,A) and ETS(A,Ad,A) are providing more or less the same forecast. This means that forecast remains on constant level since Trend “N” means “pure” exponentiell smoothing without trend (see above).

Therefore finally **Trend term** = “A” is chosen in general.

*Seasonal term*: either none (“N”), additive (“A”) or multiplicative (“M”).

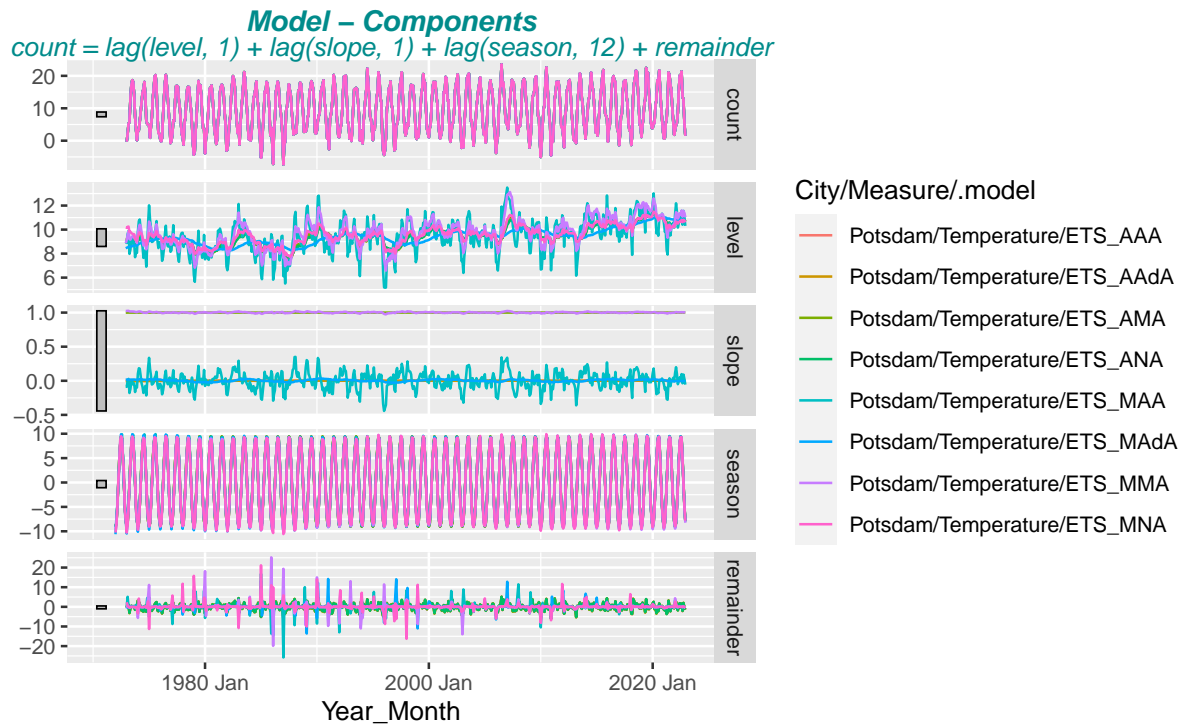
For CO2 and Temperature Data we have a clear seasonal pattern and seasonal term adds always a (more or less) fix amount on level and trend component. Therefore “A” additve term is chosen. For Precipitation the seasonal patttern is only slight. Inead, a multiplicative seasonal term results in “exploding” forecasts.

Since monthly data are strongly seasonal **seasonal term** “A” is chosen.

## 2.1 ETS Models and their componentes

```
#> [1] "model(ETS(count)) => provides best automatically chosen model"
#> # A tibble: 1 x 11
#>   City      Measure      .model sigma2 log_lik  AIC  AICc  BIC  MSE  AMSE  MAE
#>   <chr>    <fct>        <chr>   <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 Potsdam Temperature ETS(co~    3.64  -2299. 4628. 4629. 4694.   3.55  3.62  1.47
#> Series: count
#> Model: ETS(A,N,A)
#> Smoothing parameters:
#>   alpha = 0.06880272
#>   gamma = 0.0001001102
#>
#> Initial states:
#>   l[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]      s[-5]      s[-6]
#> 8.956222 -7.879422 -4.811496 0.06062935 4.847677 9.007473 9.368461 7.608746
#>   s[-7]      s[-8]      s[-9]      s[-10]     s[-11]
#> 4.34736 -0.3127315 -4.974572 -8.264206 -8.997919
#>
#> sigma^2: 3.6352
#>
#>   AIC      AICc      BIC
#> 4628.393 4629.215 4694.347
#> Model Selection by Information Criterion - lowest AIC, AICc, BIC
#> # A tibble: 8 x 11
```

```
#>   City      Measure      .model  sigma2 log_lik  AIC  AICc  BIC  MSE  AMSE  MAE
#>   <chr>    <fct>      <chr>    <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 Potsdam Temperature ETS_ANA    3.64  -2299. 4628. 4629. 4694.  3.55  3.62 1.47
#> 2 Potsdam Temperature ETS_AMA    3.65  -2300. 4633. 4634. 4708.  3.56  3.65 1.48
#> 3 Potsdam Temperature ETS_AAA    3.65  -2300. 4633. 4634. 4708.  3.56  3.64 1.48
#> 4 Potsdam Temperature ETS_AA~    3.66  -2299. 4635. 4636. 4714.  3.55  3.64 1.48
#> 5 Potsdam Temperature ETS_MA~    4.38  -3441. 6919. 6920. 6998.  4.01  4.04 0.766
#> 6 Potsdam Temperature ETS_MNA    5.66  -3505. 7041. 7041. 7107.  4.02  4.11 0.845
#> 7 Potsdam Temperature ETS_MAA    5.49  -3510. 7054. 7055. 7128.  4.75  5.51 0.828
#> 8 Potsdam Temperature ETS_MMA    6.69  -3566. 7167. 7168. 7241.  4.13  4.47 0.839
```



### 2.1.1 Residual Accuracy with one-step-ahead fitted residuals - check RMSE, MAE

Residual accuracy can be computed directly from models as the one-step-ahead fitted residuals are available. Select forecast models that minimises for lowest

- MAE (Mean absolute error, will lead to forecasts of the median) and
- RMSE (Root mean squared error, lead to forecasts of the mean)

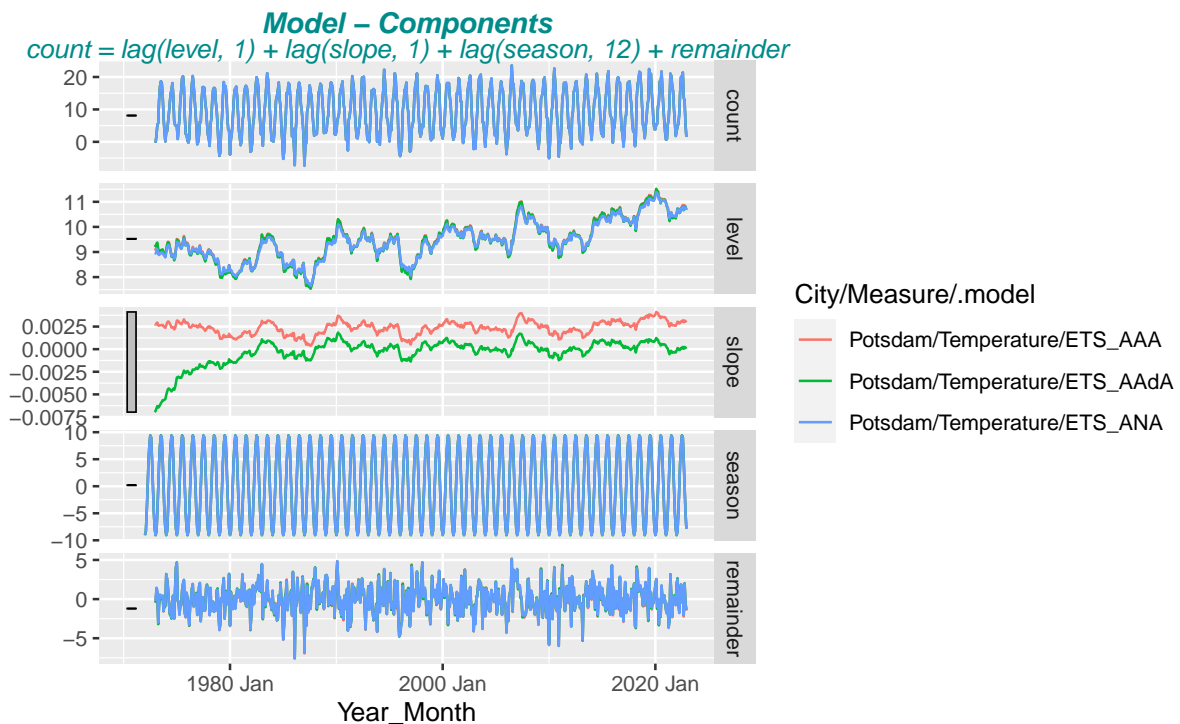
```
#> # A tibble: 8 x 12
#>   City      Measure      .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE
#>   <chr>    <fct>      <chr> <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 Potsdam Temperature ETS_A~ Trai~  0.0417  1.88  1.47  1.97  69.5  0.724 0.706
#> 2 Potsdam Temperature ETS_A~ Trai~  0.0321  1.88  1.48  1.25  69.0  0.725 0.706
#> 3 Potsdam Temperature ETS_A~ Trai~  0.0192  1.89  1.48  1.72  68.3  0.728 0.706
#> 4 Potsdam Temperature ETS_A~ Trai~  0.00315 1.89  1.48  0.868 69.6  0.725 0.706
#> 5 Potsdam Temperature ETS_M~ Trai~  0.0268  2.00  1.55  0.162 76.9  0.761 0.751
#> 6 Potsdam Temperature ETS_M~ Trai~  0.00807 2.00  1.57  1.29  68.9  0.768 0.751
#> 7 Potsdam Temperature ETS_M~ Trai~ -0.0917 2.03  1.62 -1.39 69.9  0.794 0.761
#> 8 Potsdam Temperature ETS_M~ Trai~  0.00104 2.18  1.72  4.00 69.1  0.846 0.817
#> # ... with 1 more variable: ACF1 <dbl>
```



### 2.1.2 Ljung-Box Test - independence/white noise of the forecasts residuals

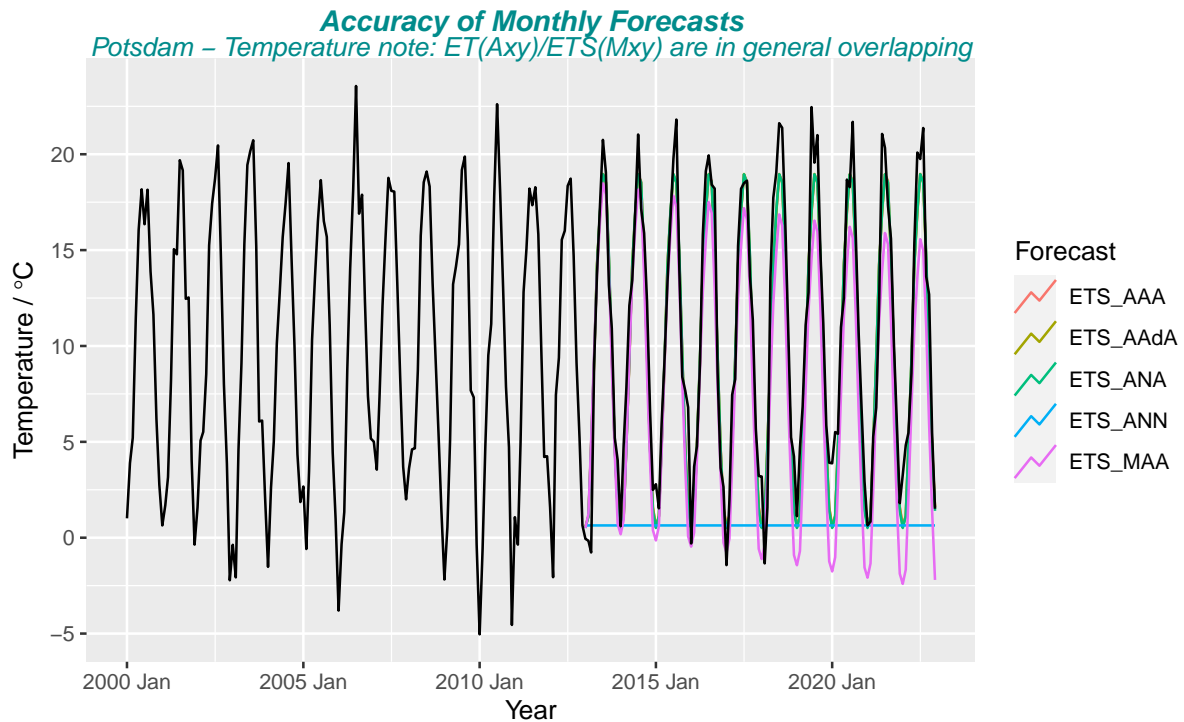
```
#> Null Hypothesis of independence/white noise for residuals - for p < 0.05: reject H_0
#> # A tibble: 8 x 5
#>   City      Measure      .model lb_stat lb_pvalue
#>   <chr>    <fct>      <chr>    <dbl>    <dbl>
#> 1 Potsdam Temperature ETS_AMA      44.1 0.0465
#> 2 Potsdam Temperature ETS_AAA      44.3 0.0444
#> 3 Potsdam Temperature ETS_AAdA     44.5 0.0425
#> 4 Potsdam Temperature ETS_ANA      45.1 0.0376
#> 5 Potsdam Temperature ETS_MNA      48.8 0.0165
#> 6 Potsdam Temperature ETS_MMA      53.7 0.00493
#> 7 Potsdam Temperature ETS_MAA      63.7 0.000321
#> 8 Potsdam Temperature ETS_MAdA     83.2 0.000000675
```

### 2.1.3 ETS Models - components of ETS(A,N,A), ETS(A,A,A), ETS(A,Ad,A), models



### 2.1.4 Forecast Accuracy with Training/Test Data

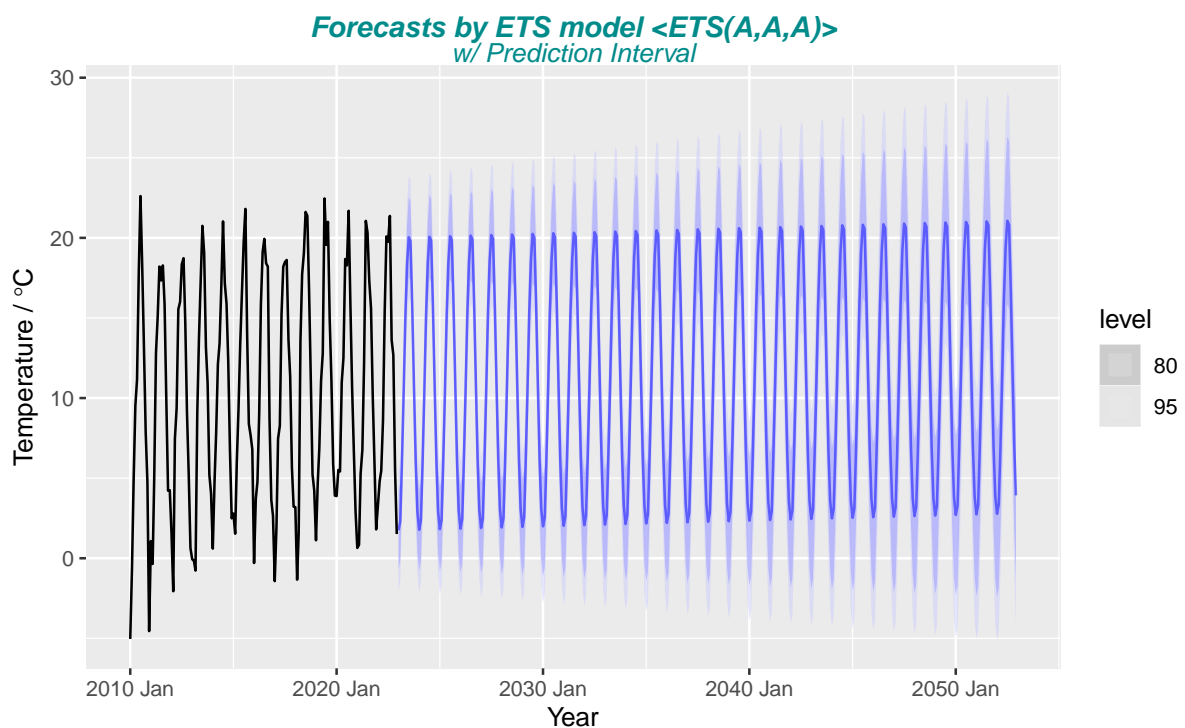
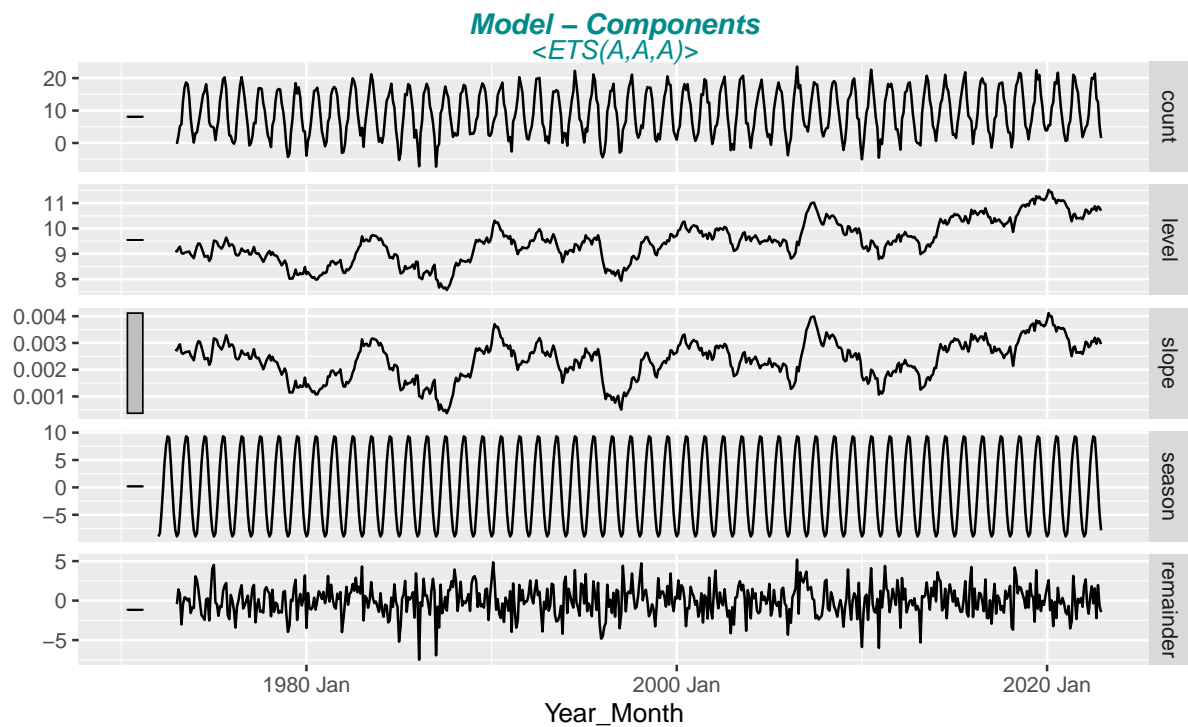
```
#> # A tibble: 5 x 12
#>   .model City      Measure .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
#>   <chr>   <chr>    <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 ETS_AAA Potsd~ Temper~ Test  0.992 2.06 1.65 40.7 48.6 0.816 0.774 0.0409
#> 2 ETS_ANA Potsd~ Temper~ Test  0.997 2.07 1.67 40.3 48.2 0.826 0.779 0.0559
#> 3 ETS_AAdA Potsd~ Temper~ Test  1.00 2.07 1.66 40.8 48.7 0.822 0.780 0.0404
#> 4 ETS_MAA Potsd~ Temper~ Test  2.78 3.55 3.01 68.1 71.4 1.48 1.34 0.336
#> 5 ETS_ANN Potsd~ Temper~ Test  9.93 12.1 10.1 104. 104. 4.97 4.54 0.795
```



## 2.2 Forecasting with selected ETS model <ETS(A,A,A)>

### 2.2.1 Forecast Plot of selected ETS model

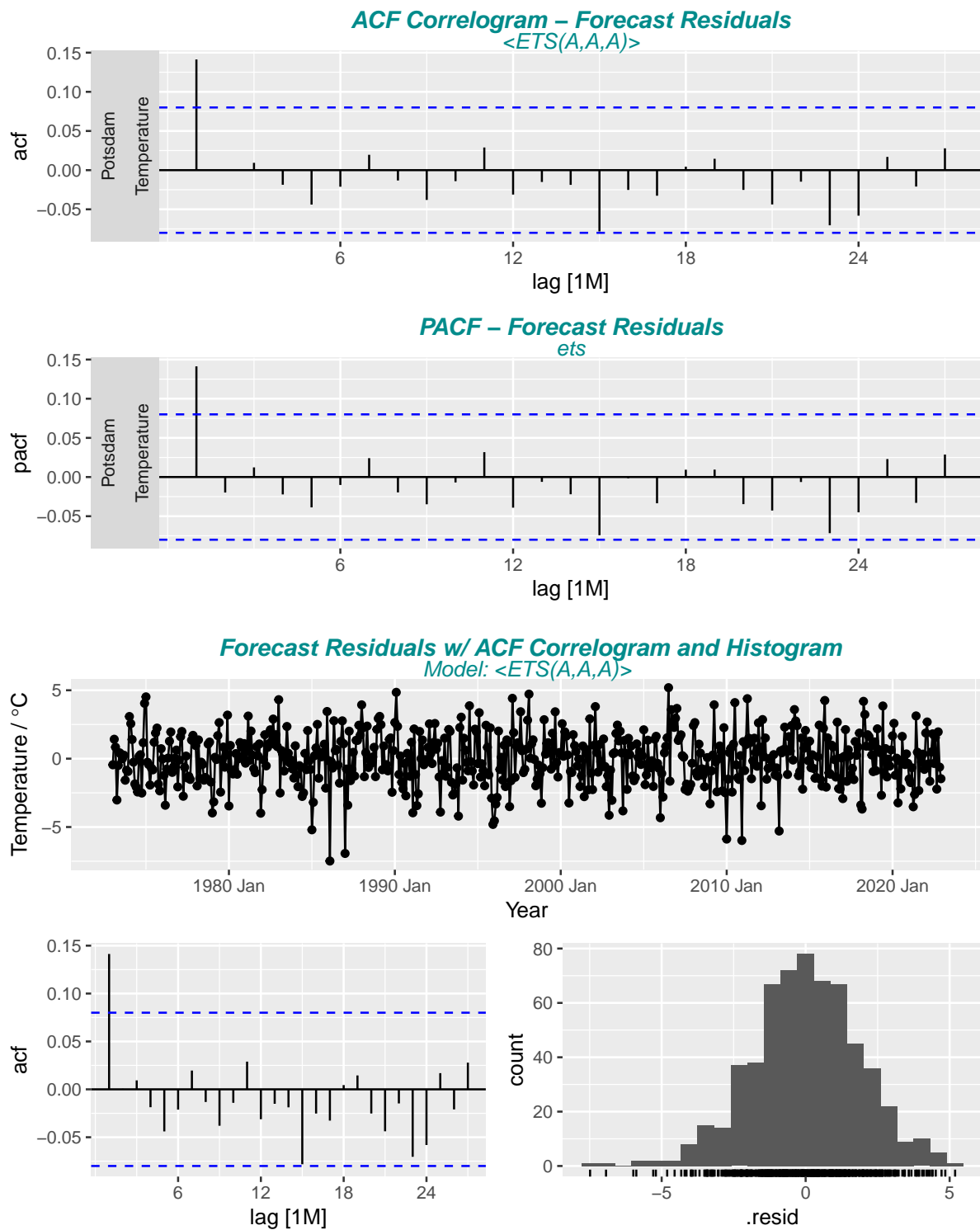
```
#> Provide model coefficients by report(fit_model)
#> Series: count
#> Model: ETS(A,A,A)
#> Smoothing parameters:
#>   alpha = 0.08172626
#>   beta  = 0.0001018886
#>   gamma = 0.0001055238
#>
#> Initial states:
#>   l[0]      b[0]      s[0]      s[-1]      s[-2]      s[-3]      s[-4]      s[-5]
#> 9.126133 0.002763414 -7.843971 -4.758141 0.02923505 4.964201 9.071047 9.31864
#>   s[-6]      s[-7]      s[-8]      s[-9]      s[-10]      s[-11]
#> 7.650969 4.292687 -0.4842531 -4.896102 -8.38397 -8.960341
#>
#> sigma^2: 3.653
#>
#>      AIC      AICc      BIC
#> 4633.276 4634.327 4708.024
```



### 2.2.2 Residual Stationarity

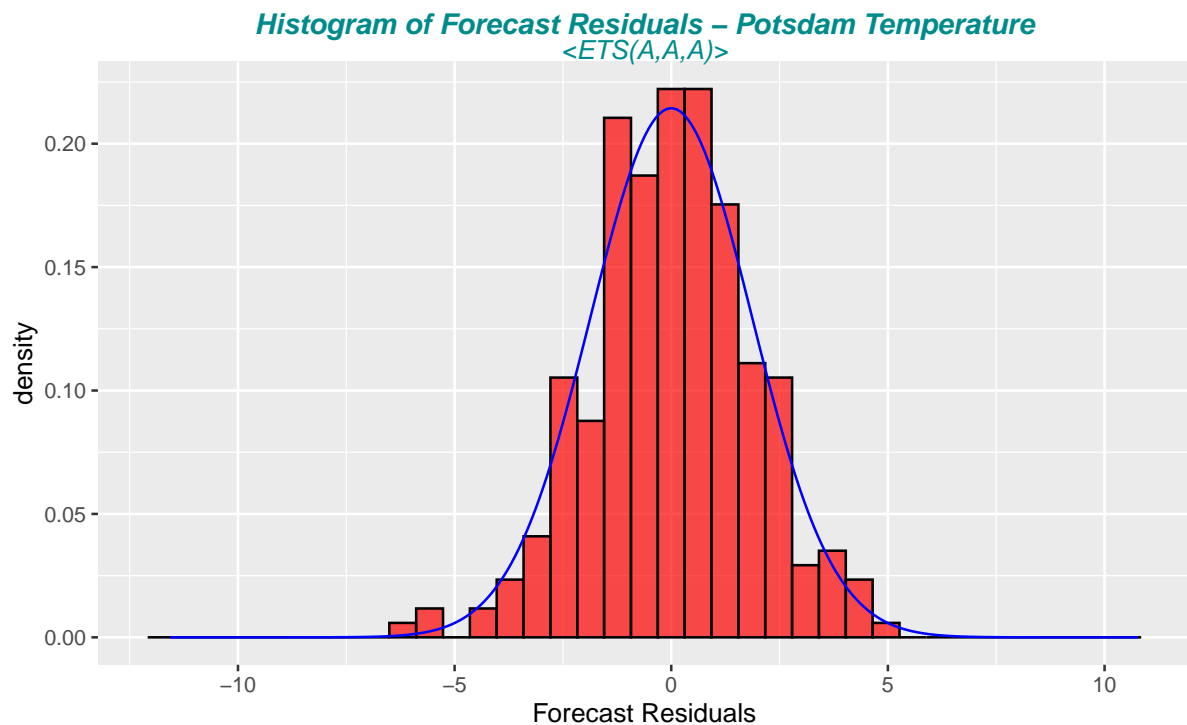
Required checks to be ready for forecasting:

- ACF Forecast Residual: all spikes are within the significance limits, so the residuals appear to be white noise
- The Ljung-Box test also shows that the residuals have no remaining autocorrelations
- Forecast Residuals are more or less normally distributed with roughly centred on zero



### 2.2.3 Histogram of forecast residuals with overlaid normal curve

```
#> Null Hypothesis of independence/white noise for residuals - for p < 0.05: reject H_0
#> # A tibble: 1 x 5
#>   City      Measure      .model lb_stat lb_pvalue
#>   <chr>    <fct>      <chr>    <dbl>    <dbl>
#> 1 Potsdam Temperature ets        28.5    0.544
```



### 3 ARIMA Forecasting Models - AutoRegressive-Integrated Moving Average

Exponential smoothing and ARIMA (AutoRegressive-Integrated Moving Average) models are the two most widely used approaches to time series forecasting, and provide complementary approaches to the problem.

While exponential smoothing models are based on a description of the trend and seasonality in the data, ARIMA models aim to describe the autocorrelations in the data.

#### 3.1 Seasonal ARIMA models

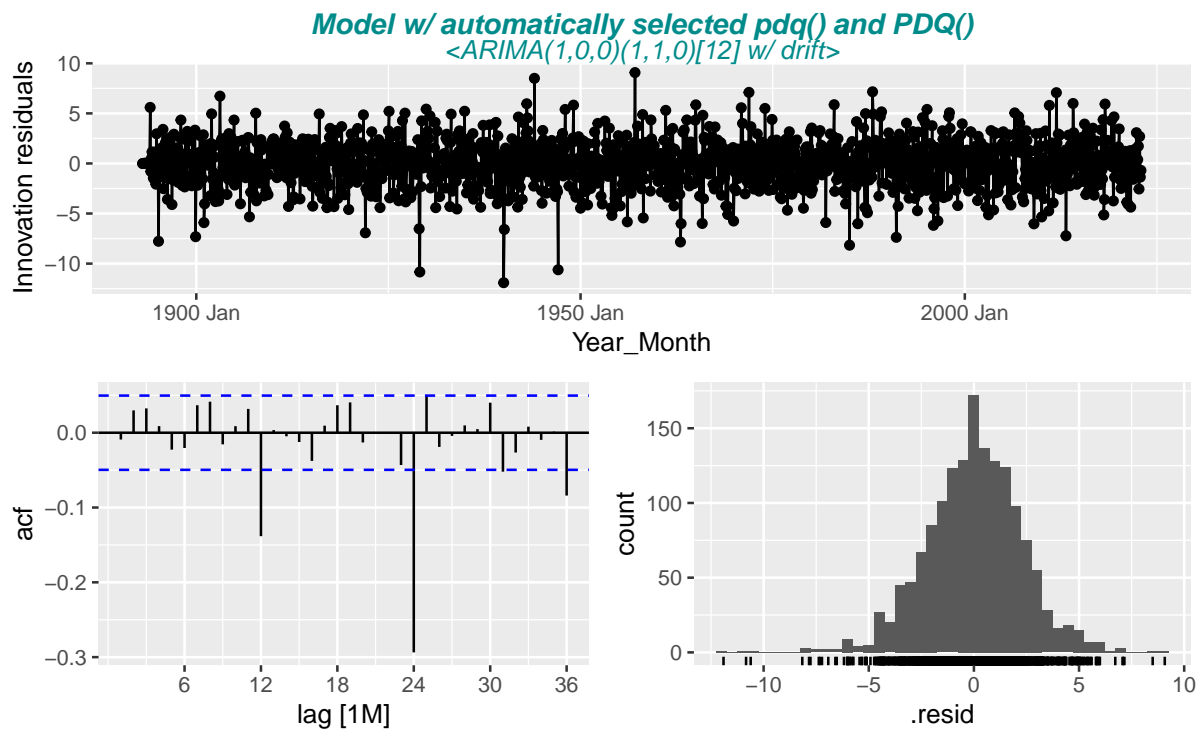
Non-seasonal ARIMA models are generally denoted  $ARIMA(p,d,q)$  where parameters  $p$ ,  $d$ , and  $q$  are non-negative integers, \*  $p$  is the order (number of time lags) of the autoregressive model \*  $d$  is the degree of differencing (number of times the data have had past values subtracted) \*  $q$  is the order of the moving-average model of past forecast errors .

The value of  $d$  has an effect on the prediction intervals — the higher the value of  $d$ , the more rapidly the prediction intervals increase in size. For  $d=0$ , the point forecasts are equal to the mean of the data and the long-term forecast standard deviation will go to the standard deviation of the historical data, so the prediction intervals will all be essentially the same.

Seasonal ARIMA models are usually denoted  $ARIMA(p,d,q)(P,D,Q)_m$ , where  $m$  refers to the number of periods in each season, and the uppercase  $P,D,Q$  refer to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.

```
#> # A tibble: 1 x 10
#>   City      Measure   .model sigma2 log_lik   AIC   AICc   BIC ar_roots  ma_ro~1
#>   <chr>    <fct>      <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>    <list>
#> 1 Potsdam Temperature arima     5.36 -3497. 7001. 7001. 7023. <cpl [13]> <cpl>
#> # ... with abbreviated variable name 1: ma_roots
#> Series: count
```

```
#> Model: ARIMA(1,0,0)(1,1,0)[12] w/ drift
#>
#> Coefficients:
#>          ar1      sar1  constant
#>       0.2263  -0.5035   0.0222
#> s.e.  0.0248   0.0220   0.0589
#>
#> sigma^2 estimated as 5.362:  log likelihood=-3496.64
#> AIC=7001.28  AICc=7001.31  BIC=7022.66
```



```
#> Model Selection by Information Criterion - lowest AIC, AICc, BIC
#> choose p, q parameter accordingly - but only for same d, D values
#> # A tibble: 13 x 10
#>   City      Measure .model  sigma2 log_lik  AIC  AICc  BIC ar_ro~1 ma_ro~2
#>   <chr>    <fct>      <chr>    <dbl>  <dbl>  <dbl> <dbl> <dbl> <list>  <list>
#> 1 Potsdam Temperature ARIMA_2~  3.51 -1227. 2464. 2464. 2486. <cpl> <cpl>
#> 2 Potsdam Temperature ARIMA_1~  3.51 -1226. 2464. 2464. 2490. <cpl> <cpl>
#> 3 Potsdam Temperature ARIMA_1~  3.52 -1227. 2464. 2464. 2486. <cpl> <cpl>
#> 4 Potsdam Temperature ARIMA_0~  3.54 -1228. 2466. 2466. 2488. <cpl> <cpl>
#> 5 Potsdam Temperature ARIMA_1~  4.77 -1295. 2599. 2599. 2616. <cpl> <cpl>
#> 6 Potsdam Temperature ARIMA_1~  5.33 -1326. 2661. 2661. 2683. <cpl> <cpl>
#> 7 Potsdam Temperature ARIMA_2~  5.33 -1326. 2661. 2661. 2683. <cpl> <cpl>
#> 8 Potsdam Temperature ARIMA_3~  5.23 -1356. 2729. 2729. 2764. <cpl> <cpl>
#> 9 Potsdam Temperature ARIMA_2~  6.54 -1384. 2777. 2777. 2794. <cpl> <cpl>
#> 10 Potsdam Temperature ARIMA_1~  6.95 -1404. 2814. 2814. 2827. <cpl> <cpl>
#> 11 Potsdam Temperature ARIMA_0~  6.96 -1405. 2815. 2815. 2828. <cpl> <cpl>
#> 12 Potsdam Temperature ARIMA_0~  8.83 -1474. 2951. 2951. 2960. <cpl> <cpl>
#> 13 Potsdam Temperature ARIMA_1~  9.64 -1498. 2999. 2999. 3008. <cpl> <cpl>
#> # ... with abbreviated variable names 1: ar_roots, 2: ma_roots
```

Good models are obtained by minimising the AIC, AICc or BIC (see `glance(fit_arma)` output). The preference is to use the AICc to select  $p$  and  $q$ .

These information criteria tend not to be good guides to selecting the appropriate order of differencing ( $d$ ) of a model, but only for selecting the values of  $p$  and  $q$ . This is because the differencing changes the

data on which the likelihood is computed, making the AIC values between models with different orders of differencing not comparable.

### 3.1.1 Residual Accuracy with one-step-ahead fitted residuals - check RMSE, MAE

Residual accuracy can be computed directly from models as the one-step-ahead fitted residuals are available. Select forecast models that minimises for lowest

- MAE (Mean absolute error, will lead to forecasts of the median) and
- RMSE (Root mean squared error, lead to forecasts of the mean)

```
#> # A tibble: 14 x 12
#>   City      Measure .model .type      ME    RMSE    MAE      MPE    MAPE    MASE
#>   <chr>    <fct>      <chr> <chr>    <dbl> <dbl> <dbl>    <dbl> <dbl>    <dbl>
#> 1 Potsdam Temperat~ ARIMA~ Trai~  0.0947  1.85  1.43  6.13  68.8  0.703
#> 2 Potsdam Temperat~ ARIMA~ Trai~  0.0986  1.85  1.43  7.13  68.2  0.701
#> 3 Potsdam Temperat~ ARIMA~ Trai~  0.0944  1.85  1.43  6.87  68.8  0.702
#> 4 Potsdam Temperat~ ARIMA~ Trai~  0.0689  1.85  1.44  5.85  70.2  0.705
#> 5 Potsdam Temperat~ ARIMA~ Trai~  0.0568  2.16  1.68  9.89  81.2  0.826
#> 6 Potsdam Temperat~ ARIMA~ Trai~  0.0281  2.27  1.80  7.87  95.3  0.882
#> 7 Potsdam Temperat~ ARIMA~ Trai~  0.00243  2.28  1.77  10.5  87.5  0.867
#> 8 Potsdam Temperat~ ARIMA~ Trai~  0.00243  2.28  1.77  10.5  87.5  0.867
#> 9 Potsdam Temperat~ ARIMA~ Trai~ -0.0124  2.52  1.98  11.5  88.7  0.971
#> 10 Potsdam Temperat~ ARIMA~ Trai~ -0.0113  2.60  1.96  5.87  103.  0.964
#> 11 Potsdam Temperat~ ARIMA~ Trai~ -0.0104  2.61  1.97  5.94  103.  0.965
#> 12 Potsdam Temperat~ ARIMA~ Trai~ -0.00915  2.94  2.28 -0.0179 113.  1.12
#> 13 Potsdam Temperat~ ARIMA~ Trai~ -0.00816  3.07  2.38  5.27  131.  1.17
#> 14 Potsdam Temperat~ ARIMA~ Trai~ NaN      NaN    NaN    NaN    NaN    NaN
#> # ... with 2 more variables: RMSSE <dbl>, ACF1 <dbl>
```

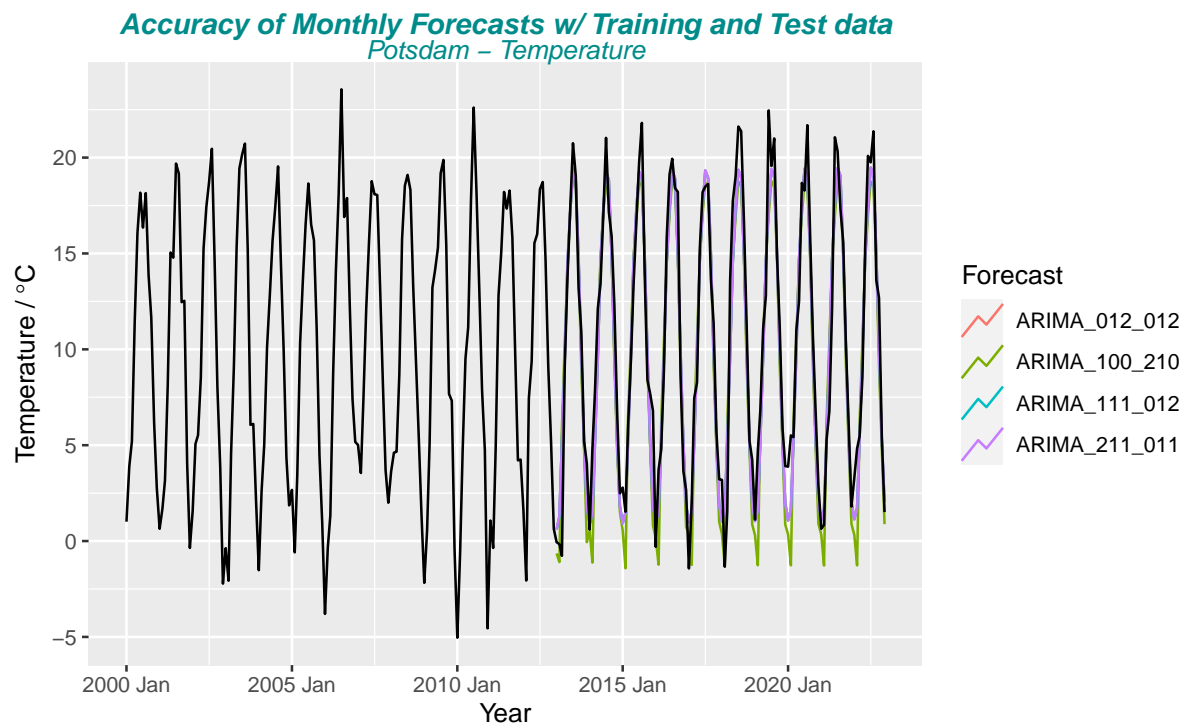
### 3.1.2 Ljung-Box Test - independence/white noise of the forecasts residuals

```
#> Null Hypothesis of independence/white noise for residuals - for p < 0.05: reject H_0
#> # A tibble: 14 x 5
#>   City      Measure .model      lb_stat lb_pvalue
#>   <chr>    <fct>      <chr>    <dbl>    <dbl>
#> 1 Potsdam Temperature ARIMA_111_012  27.3  6.06e- 1
#> 2 Potsdam Temperature ARIMA_012_012  27.4  6.03e- 1
#> 3 Potsdam Temperature ARIMA_211_011  27.6  5.91e- 1
#> 4 Potsdam Temperature ARIMA_111_112  28.6  5.40e- 1
#> 5 Potsdam Temperature ARIMA_301_200  88.5  1.10e- 7
#> 6 Potsdam Temperature ARIMA_100_210  96.9  5.63e- 9
#> 7 Potsdam Temperature ARIMA_100_110  107.  1.47e-10
#> 8 Potsdam Temperature ARIMA_200_110  107.  1.47e-10
#> 9 Potsdam Temperature ARIMA_010_110  256.  0
#> 10 Potsdam Temperature ARIMA_012_010  182.  0
#> 11 Potsdam Temperature ARIMA_110_010  312.  0
#> 12 Potsdam Temperature ARIMA_111_010  182.  0
#> 13 Potsdam Temperature ARIMA_210_110  166.  0
#> 14 Potsdam Temperature ARIMA_002_200    NA    NA
```

### 3.1.3 Forecast Accuracy with Training/Test Data

```
#> # A tibble: 4 x 12
#>   .model City Measure .type      ME    RMSE    MAE    MPE    MAPE    MASE RMSSE    ACF1
#>   <chr>   <chr> <fct>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
```

```
#> 1 ARIMA_2~ Pots~ Temper~ Test 0.604 1.90 1.51 36.9 50.6 0.744 0.713 0.0339
#> 2 ARIMA_1~ Pots~ Temper~ Test 0.606 1.90 1.50 36.4 50.2 0.743 0.714 0.0285
#> 3 ARIMA_0~ Pots~ Temper~ Test 0.604 1.90 1.51 36.8 50.6 0.744 0.714 0.0301
#> 4 ARIMA_1~ Pots~ Temper~ Test 1.06 2.42 1.88 16.9 54.6 0.931 0.909 0.00348
```

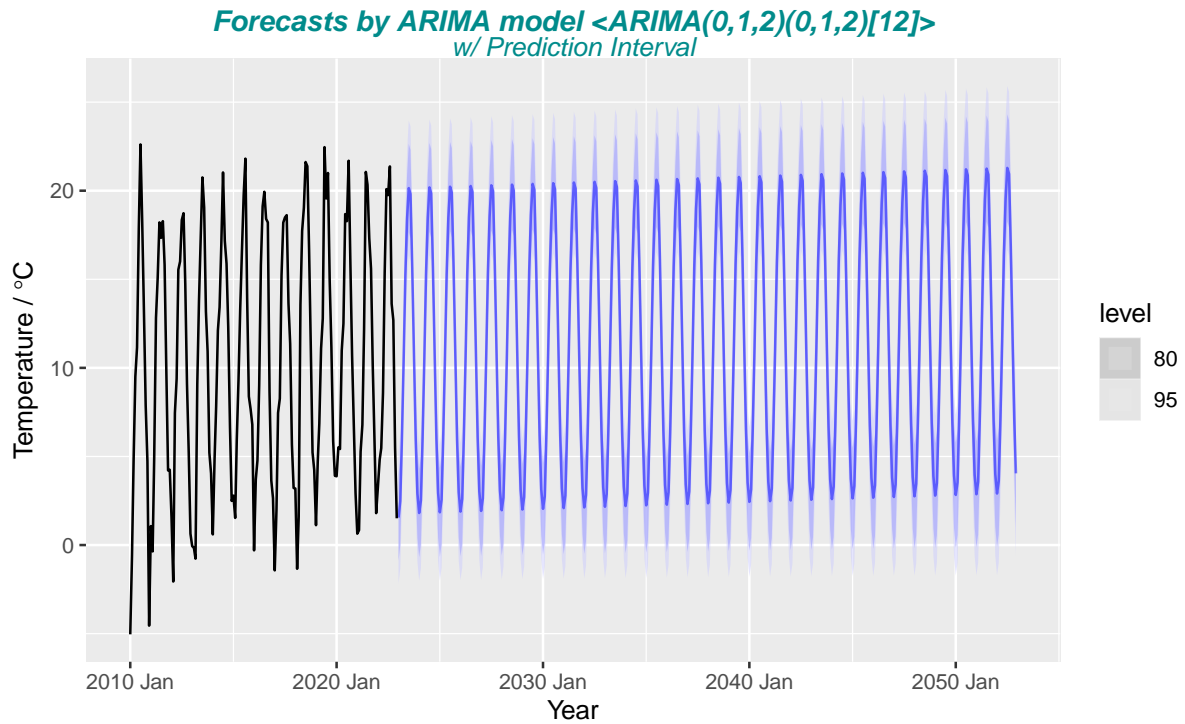


## 3.2 Temperature - Forecasting with selected ARIMA model <ARIMA(0,1,2)(0,1,2)[12]>

### 3.2.1 Forecast Plot of selected ARIMA model

```
#> Provide model coefficients by report(fit_model)
#> Series: count
#> Model: ARIMA(0,1,2)(0,1,2)[12]
#>
#> Coefficients:
#>      ma1      ma2      sma1      sma2
#>    -0.7921 -0.1800 -0.9974 -0.0025
#> s.e.    0.0417    0.0469    0.0663    0.0473
#>
#> sigma^2 estimated as 3.538:  log likelihood=-1227.91
#> AIC=2465.82  AICc=2465.92  BIC=2487.69
```

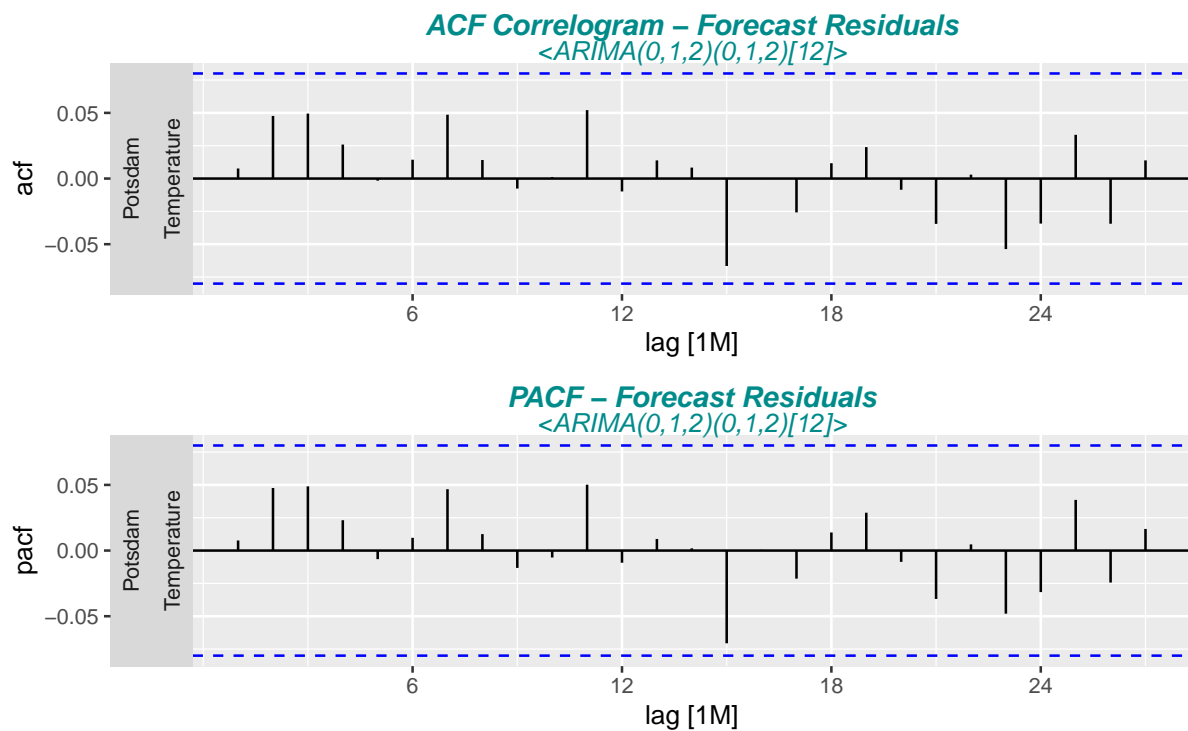


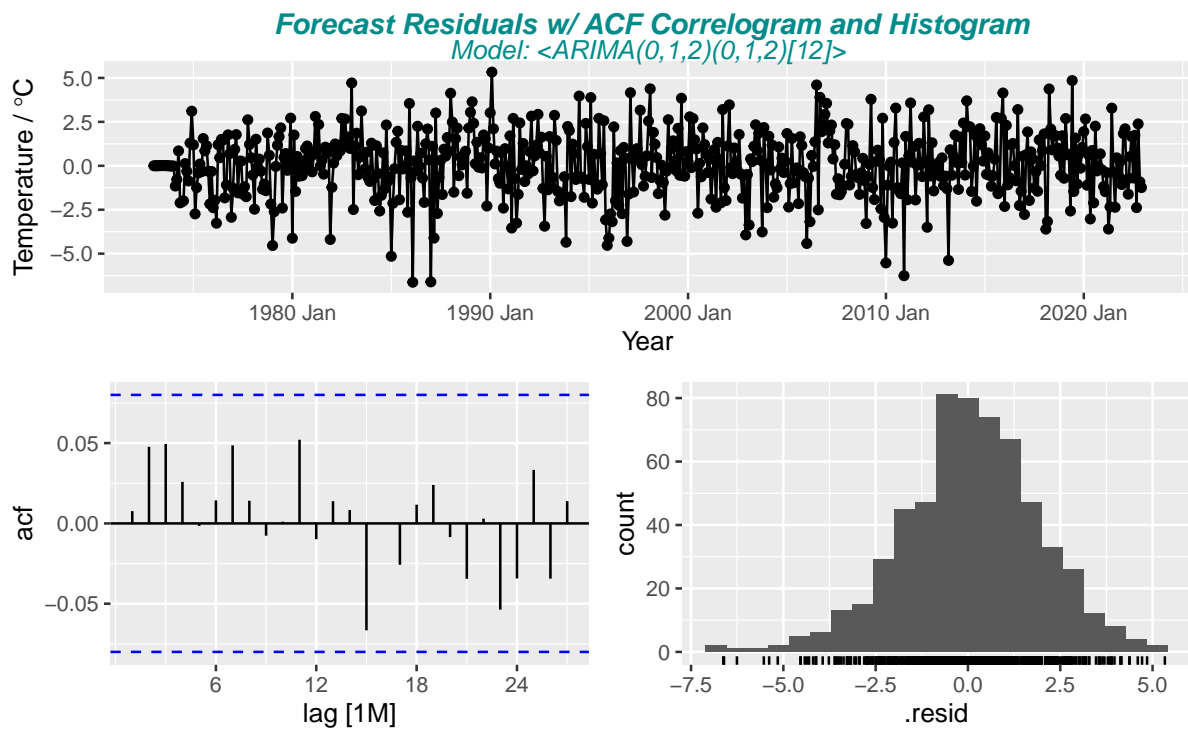


### 3.2.2 Residual Stationarity

Required checks to be ready for forecasting:

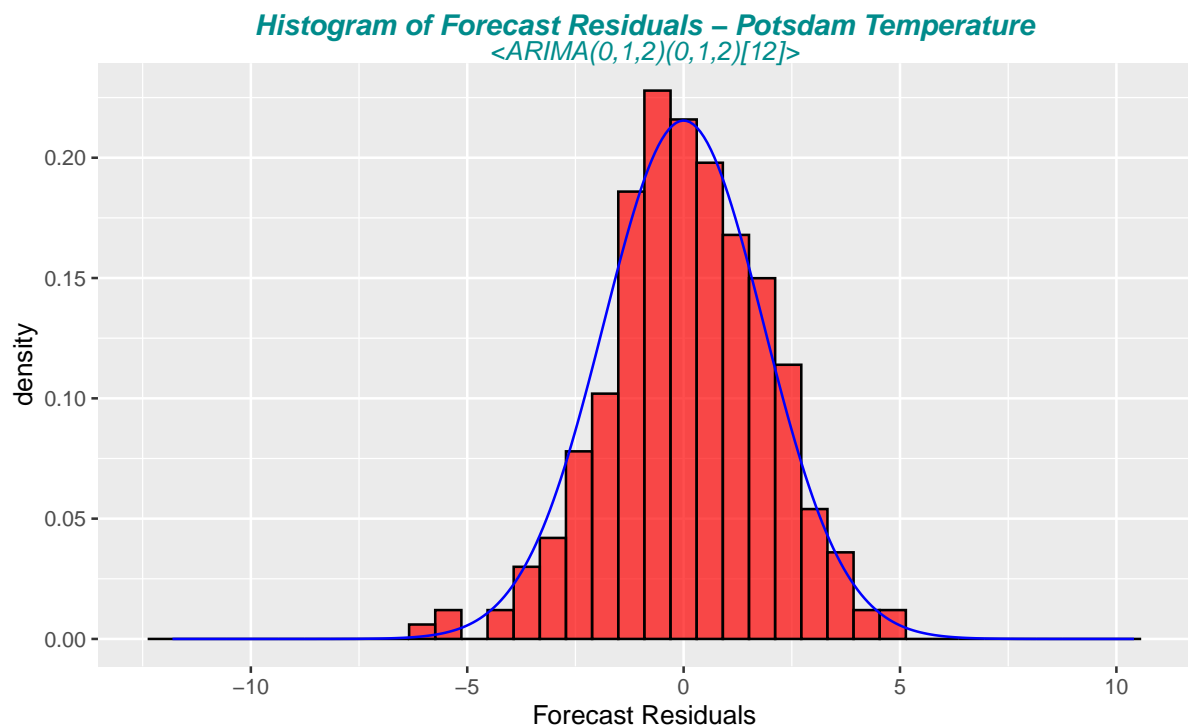
- ACF Forecast Residual: all spikes are within the significance limits, so the residuals appear to be white noise
- The Ljung-Box test also shows that the residuals have no remaining autocorrelations
- Forecast Residuals are more or less normally distributed with roughly centred on zero





### 3.2.3 Histogram of forecast residuals with overlaid normal curve

```
#> Null Hypothesis of independence/white noise for residuals - for p < 0.05: reject H_0
#> # A tibble: 1 x 5
#>   City      Measure      .model lb_stat lb_pvalue
#>   <chr>    <fct>      <chr>    <dbl>   <dbl>
#> 1 Potsdam Temperature arima      20.9    0.890
```



## 4 ARIMA vs ETS

In particular, all ETS models are non-stationary, while some ARIMA models are stationary.

The ETS models with seasonality or non-damped trend or both have two unit roots (i.e., they need two levels of differencing to make them stationary). All other ETS models have one unit root (they need one level of differencing to make them stationary).

We compare for the chosen ETS resp. ARIMA model the RMSE / MAE values. Lower values indicate a more accurate model based on the test set RMSE, ..., MASE.

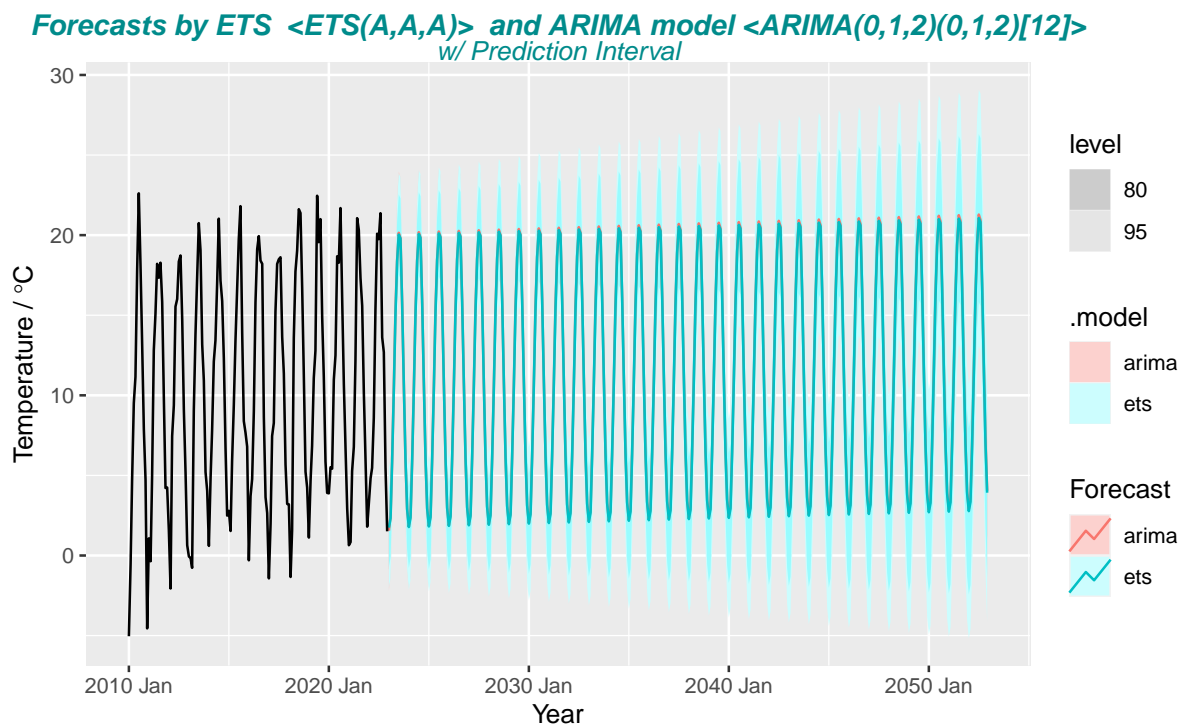
- Residual Accuracy with one-step-ahead fitted residuals
- Forecast Accuracy with Training/Test Data

Note: a good fit to training data is never an indication that the model will forecast well. Therefore the values of the Forecast Accuracy are the more relevant one.

### 4.0.1 Comparing Residual and Forecast Accuracy of selected ETS and ARIMA model

```
#> # A tibble: 4 x 12
#>   City      Measure      .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE
#>   <chr>    <fct>      <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 Potsdam Temperature ets      Trai~ 0.00315 1.89 1.48 0.868 69.6 0.725 0.706
#> 2 Potsdam Temperature arima     Trai~ 0.0689 1.85 1.44 5.85 70.2 0.705 0.695
#> 3 Potsdam Temperature ETS_AAA Test 0.992 2.06 1.65 40.7 48.6 0.816 0.774
#> 4 Potsdam Temperature ARIMA_~ Test 0.604 1.90 1.51 36.8 50.6 0.744 0.714
#> # ... with 1 more variable: ACF1 <dbl>
```

### 4.0.2 Forecast Plot of selected ETS and ARIMA model



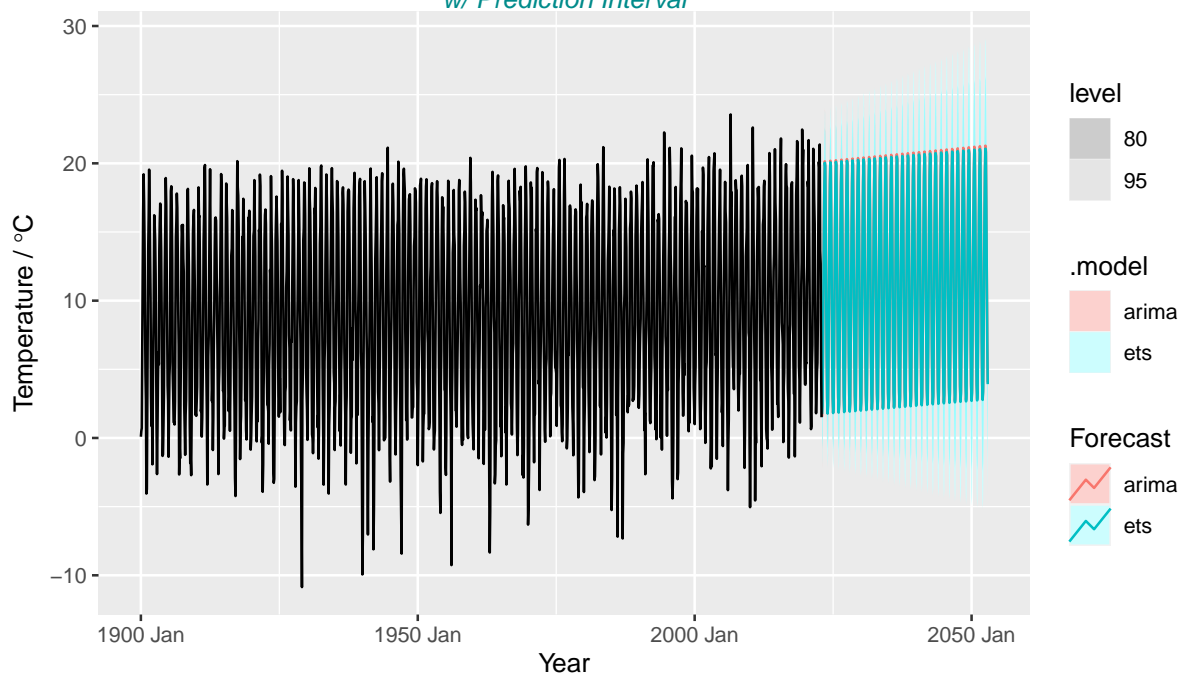
```
#> # A tsibble: 6 x 8 [1M]
#> # Key:      City, Measure, .model [2]
```

```

#> # Groups:   City, Measure, .model [2]
#>   City   Measure   .model Year_M~1      count .mean      '80%'
#>   <chr>   <fct>     <chr>   <mt>      <dist> <dbl>      <hilo>
#> 1 Potsdam Temperat~ arima  2023 Jan N(1.6, 3.6)  1.55 [-0.883396926, 3.986561]80
#> 2 Potsdam Temperat~ arima  2023 Feb N(2.5, 3.8)  2.48 [-0.007413563, 4.966834]80
#> 3 Potsdam Temperat~ arima  2023 Mrz N(5.9, 3.8)  5.88 [ 3.390960088, 8.367052]80
#> 4 Potsdam Temperat~ ets    2023 Jan N(1.7, 3.7)  1.74 [-0.707453035, 4.191386]80
#> 5 Potsdam Temperat~ ets    2023 Feb N(2.3, 3.7)  2.32 [-0.135661253, 4.779551]80
#> 6 Potsdam Temperat~ ets    2023 Mrz N(5.8, 3.7)  5.81 [ 3.346492026, 8.278064]80
#> # ... with 1 more variable: '95%' <hilo>, and abbreviated variable name
#> #   1: Year_Month
#> # A tsibble: 6 x 8 [1M]
#> # Key:       City, Measure, .model [2]
#> # Groups:   City, Measure, .model [2]
#>   City   Measure   .model Year_M~1      count .mean      '80%'
#>   <chr>   <fct>     <chr>   <mt>      <dist> <dbl>      <hilo>
#> 1 Potsdam Temperature arima  2052 Okt N(12, 5.6) 11.9 [ 8.912785, 14.969079]80
#> 2 Potsdam Temperature arima  2052 Nov N(7.1, 5.6)  7.09 [ 4.056072, 10.115621]80
#> 3 Potsdam Temperature arima  2052 Dez N(4, 5.6)  4.05 [ 1.014206,  7.077301]80
#> 4 Potsdam Temperature ets    2052 Okt N(12, 17) 11.8 [ 6.529199, 17.044515]80
#> 5 Potsdam Temperature ets    2052 Nov N(7, 17)  7.00 [ 1.736448, 12.267697]80
#> 6 Potsdam Temperature ets    2052 Dez N(3.9, 17)  3.92 [-1.354124,  9.193060]80
#> # ... with 1 more variable: '95%' <hilo>, and abbreviated variable name
#> #   1: Year_Month

```

**Forecasts by ETS <ETS(A,A,A)> and ARIMA model <ARIMA(0,1,2)(0,1,2)[12]>**  
w/ Prediction Interval



```

#> # A tibble: 180 x 5
#> # Groups:   City, Measure, .model, Year [60]
#>   City   Measure   .model Year Year_avg
#>   <chr>   <fct>     <chr> <dbl>   <dbl>
#> 1 Potsdam Temperature arima  2023     1.55
#> 2 Potsdam Temperature arima  2023     2.48
#> 3 Potsdam Temperature arima  2023     5.88
#> 4 Potsdam Temperature arima  2024     1.81

```

```

#> 5 Potsdam Temperature arima 2024 2.51
#> 6 Potsdam Temperature arima 2024 5.92
#> 7 Potsdam Temperature arima 2025 1.85
#> 8 Potsdam Temperature arima 2025 2.55
#> 9 Potsdam Temperature arima 2025 5.96
#> 10 Potsdam Temperature arima 2026 1.89
#> # ... with 170 more rows
#> # A tibble: 180 x 5
#> # Groups:   City, Measure, .model, Year [60]
#>   City Measure .model Year Year_avg
#>   <chr> <fct> <chr> <dbl> <dbl>
#> 1 Potsdam Temperature arima 2023 10.8
#> 2 Potsdam Temperature arima 2023 5.95
#> 3 Potsdam Temperature arima 2023 2.90
#> 4 Potsdam Temperature arima 2024 10.8
#> 5 Potsdam Temperature arima 2024 5.99
#> 6 Potsdam Temperature arima 2024 2.95
#> 7 Potsdam Temperature arima 2025 10.9
#> 8 Potsdam Temperature arima 2025 6.03
#> 9 Potsdam Temperature arima 2025 2.99
#> 10 Potsdam Temperature arima 2026 10.9
#> # ... with 170 more rows

```

#### 4.0.3 Ljung-Box Test - independence/white noise of the forecasts residuals

```

#> # A tibble: 2 x 5
#>   City Measure .model lb_stat lb_pvalue
#>   <chr> <fct> <chr> <dbl> <dbl>
#> 1 Potsdam Temperature arima 27.4 0.603
#> 2 Potsdam Temperature ets 44.3 0.0444

```

## 5 Yearly Data Forecasts with ARIMA and ETS

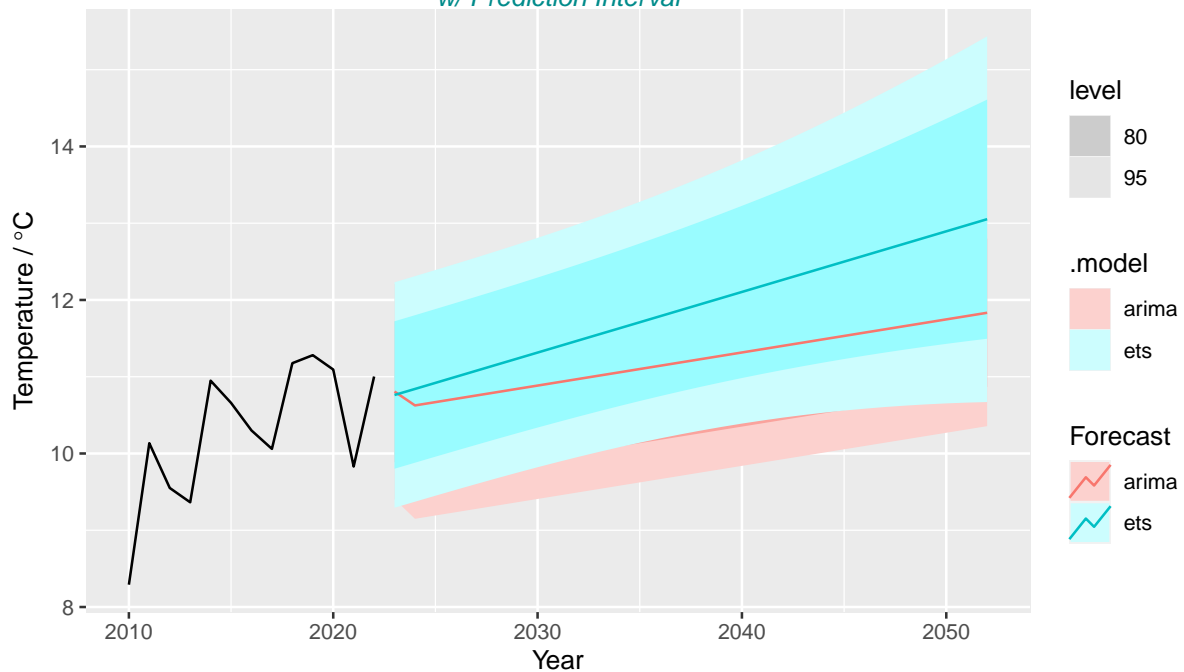
For yearly data the seasonal monthly data are replaced by the yearly average data. Therefore the seasonal component of the ETS and ARIMA model are to be taken out.

The ETS model  $\langle ETS(A, A, N) \rangle$  with seasonal term change “A” -> “N” is chosen. For ARIMA models the seasonal term (P,D,Q)<sub>m</sub> has to be taken out and an optimal ARIMA(p,1,q) with one differencing (d=1) is selected. However, for Mauna Loa two times differencing had to be selected  $\langle CO_2 \rangle \langle ARIMA(0,2,1) w/ poly \rangle$ . For Temperature and Precipitation the same model as for monthly data can be taken by leaving out the seasonal term  $\langle ARIMA(0,1,2)w/drift \rangle$ .

### 5.0.1 Comparing Residual and Forecast Accuracy of selected ETS and ARIMA model

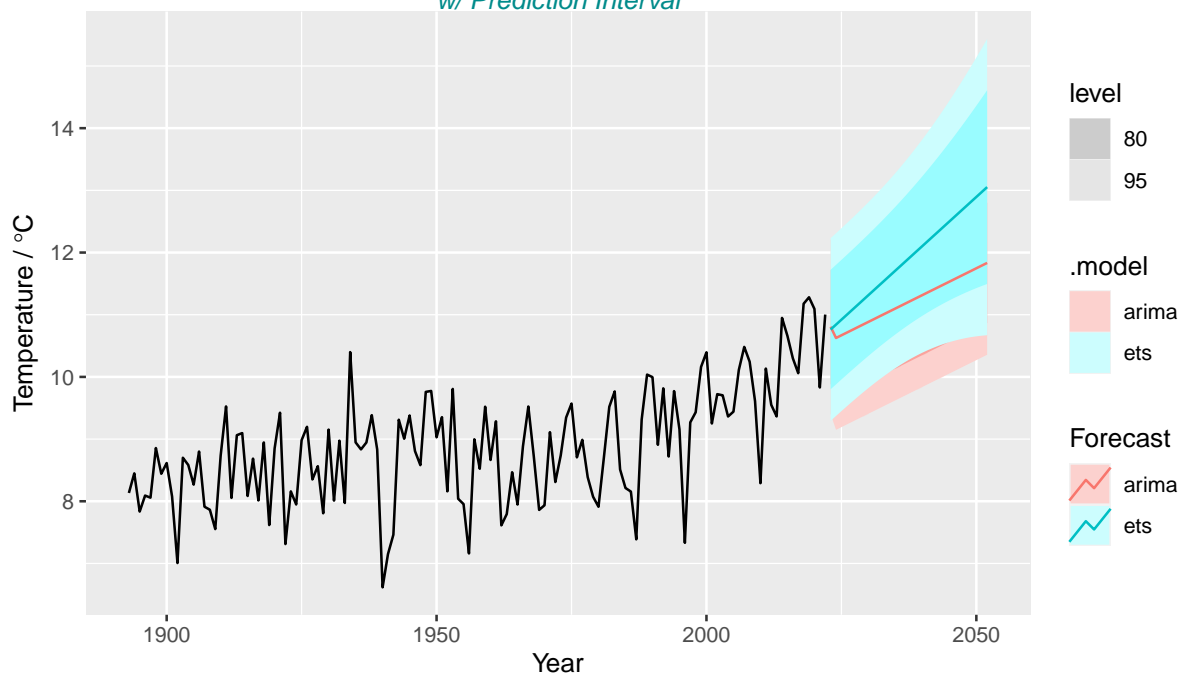
## 5.0.2 Forecast Plot of selected ETS and ARIMA model

arly Data Forecasts by ETS <ETS(A,A,N)> and ARIMA model <ARIMA(0,1,2) w/ drift>  
w/ Prediction Interval



```
#> # A tibble: 6 x 8 [1Y]
#> # Key:      City, Measure, .model [2]
#> # Groups:   City, Measure, .model [2]
#>   City      Measure      .model   Year   Year_avg .mean      '80%'
#>   <chr>    <fct>      <chr>   <dbl>   <dist> <dbl>    <hilo>
#> 1 Potsdam Temperature arima    2023 N(11, 0.51)  10.8 [9.890674, 11.72357]80
#> 2 Potsdam Temperature arima    2024 N(11, 0.57)  10.6 [9.660398, 11.59166]80
#> 3 Potsdam Temperature arima    2025 N(11, 0.57)  10.7 [9.703511, 11.63477]80
#> 4 Potsdam Temperature ets      2023 N(11, 0.56)  10.8 [9.801606, 11.72223]80
#> 5 Potsdam Temperature ets      2024 N(11, 0.56)  10.8 [9.880254, 11.80154]80
#> 6 Potsdam Temperature ets      2025 N(11, 0.56)  10.9 [9.958487, 11.88126]80
#> # ... with 1 more variable: '95%' <hilo>
#> # A tibble: 6 x 8 [1Y]
#> # Key:      City, Measure, .model [2]
#> # Groups:   City, Measure, .model [2]
#>   City      Measure      .model   Year   Year_avg .mean      '80%'
#>   <chr>    <fct>      <chr>   <dbl>   <dist> <dbl>    <hilo>
#> 1 Potsdam Temperature arima    2050 N(12, 0.57)  11.7 [10.78132, 12.71258]80
#> 2 Potsdam Temperature arima    2051 N(12, 0.57)  11.8 [10.82443, 12.75569]80
#> 3 Potsdam Temperature arima    2052 N(12, 0.57)  11.8 [10.86754, 12.79880]80
#> 4 Potsdam Temperature ets      2050 N(13, 1.3)   12.9 [11.42842, 14.36025]80
#> 5 Potsdam Temperature ets      2051 N(13, 1.4)   13.0 [11.46247, 14.48417]80
#> 6 Potsdam Temperature ets      2052 N(13, 1.5)   13.1 [11.49479, 14.60980]80
#> # ... with 1 more variable: '95%' <hilo>
```

### Early Data Forecasts by ETS <ETS(A,A,N)> and ARIMA model <ARIMA(0,1,2) w/ drift> w/ Prediction Interval



#### 5.0.3 Ljung-Box Test - independence/white noise of the forecasts residuals

```
#> # A tibble: 2 x 5
#>   City      Measure    .model lb_stat lb_pvalue
#>   <chr>    <fct>      <chr>   <dbl>   <dbl>
#> 1 Potsdam Temperature arima    39.8    0.109
#> 2 Potsdam Temperature ets      56.9    0.00213
```

## 6 Backup