Climate Data Forecasting -

Atmospheric CO_2 Concentration / Temperature / Precipitation

Wolfgang Vollmer

2023-02-20

Contents

1 Forecasting of Cottbus - 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

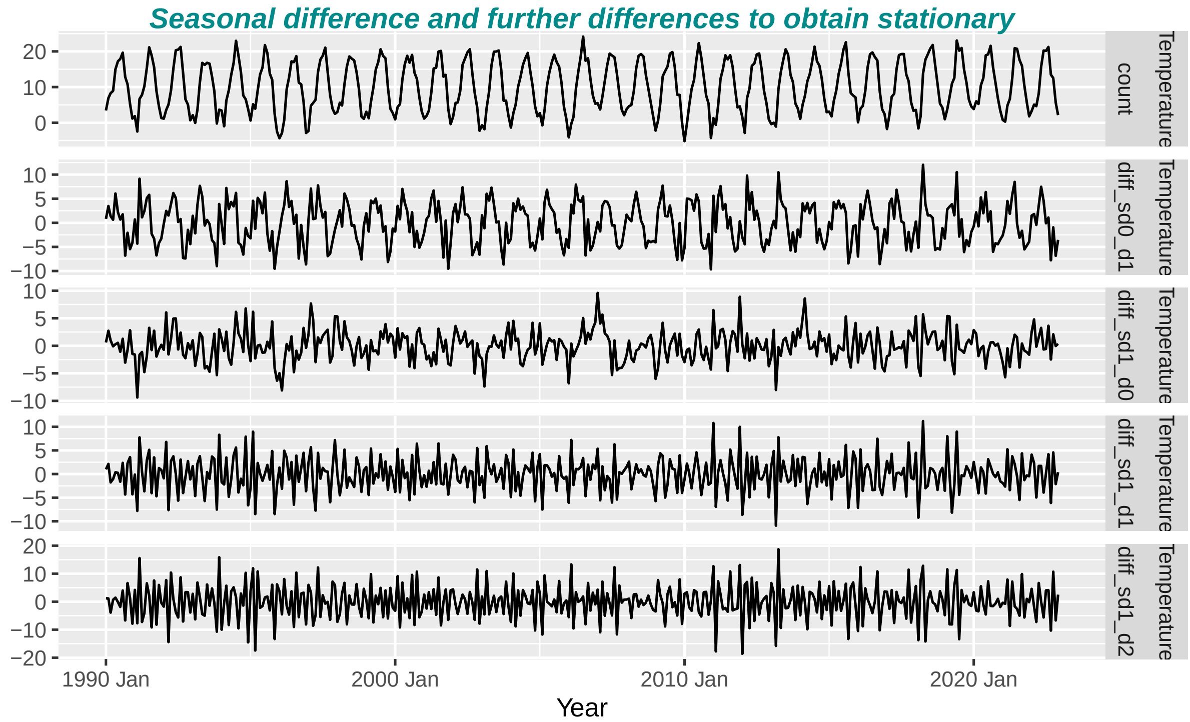
The model fit residuals have to be stationary. For good forecasting this has to be verified with residual diagnostics.

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 rsp. 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   5707.         0
#> Ljung-Box test on (difference(count, 12))
#> # A tibble: 1 x 3
#>   Measure     lb_stat lb_pvalue
#>   <fct>         <dbl>     <dbl>
#> 1 Temperature    100.         0
#> Ljung-Box test on (difference(count, 12) + difference())
#> # A tibble: 1 x 3
#>   Measure     lb_stat lb_pvalue
#>   <fct>         <dbl>     <dbl>
#> 1 Temperature    285.         0
```

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

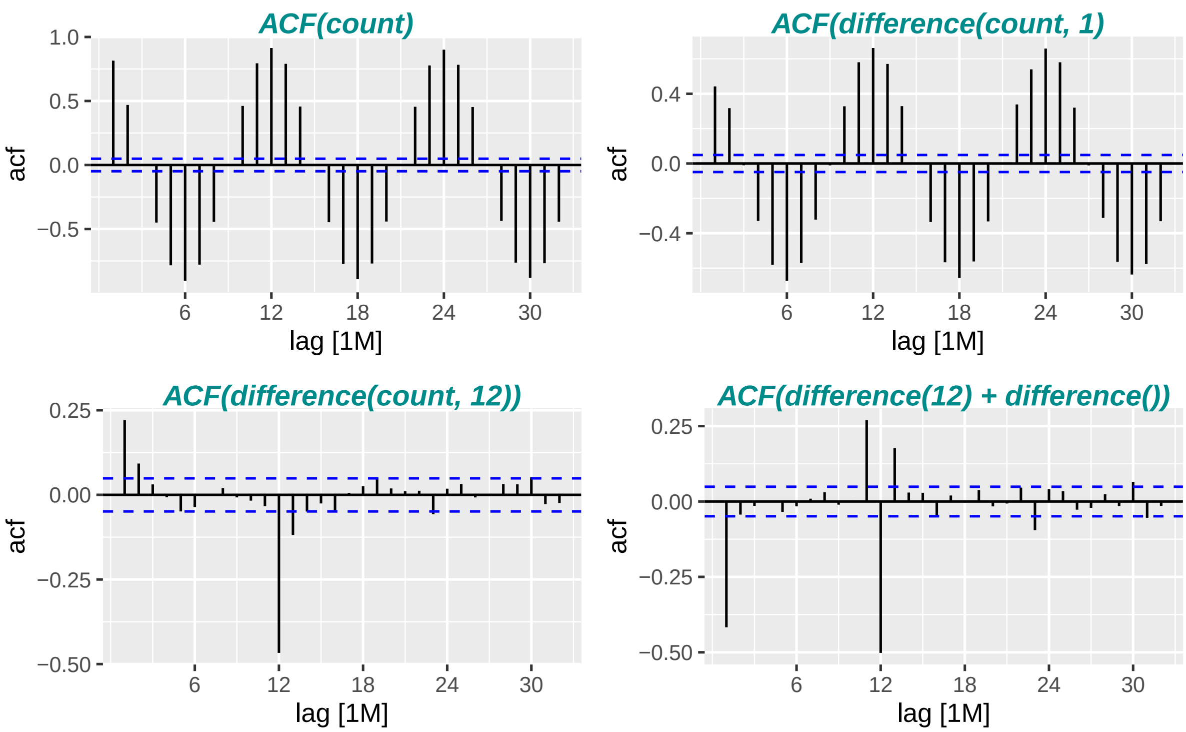
kpss test of stationary
Null Hypothesis of stationary in a given time series
=> H_0 to be rejected for $p < \alpha = 0.05$

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.

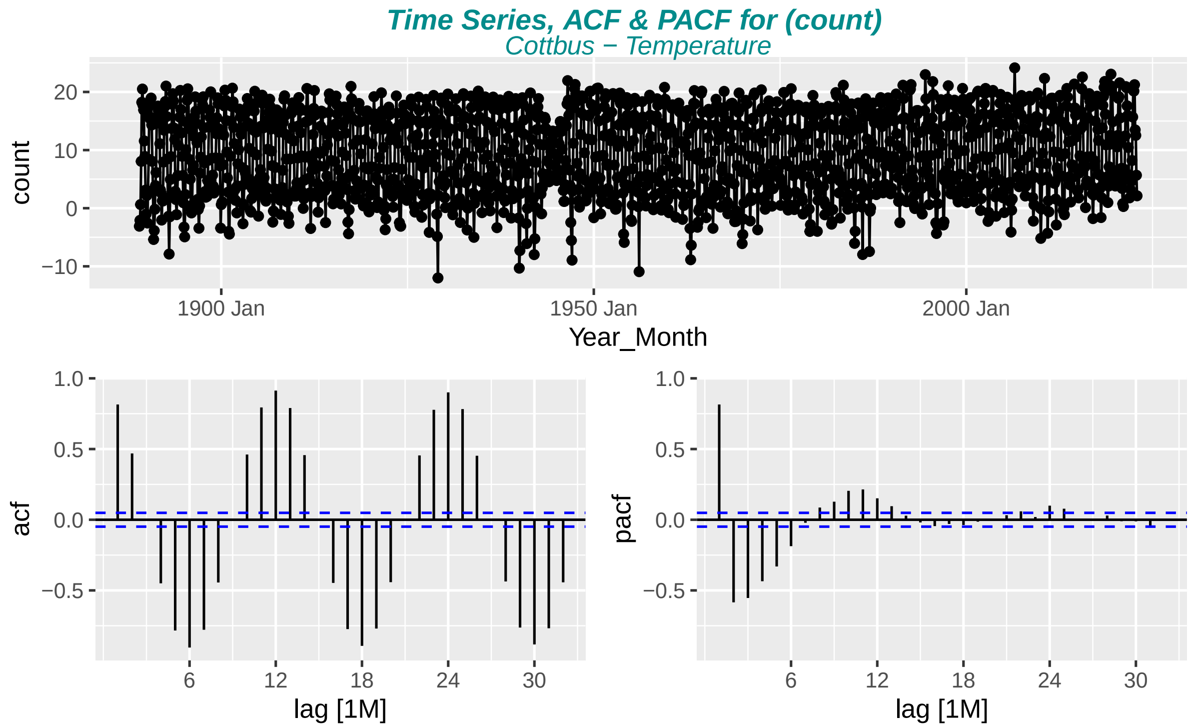
- returns 1 => for stationarity one seasonal difference rsp. difference is required

```
#> ndiffs gives the number of differences required rsp.
#>  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     0.633      0.0197       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   0.00818         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   0.00568         0.1       0      0
```

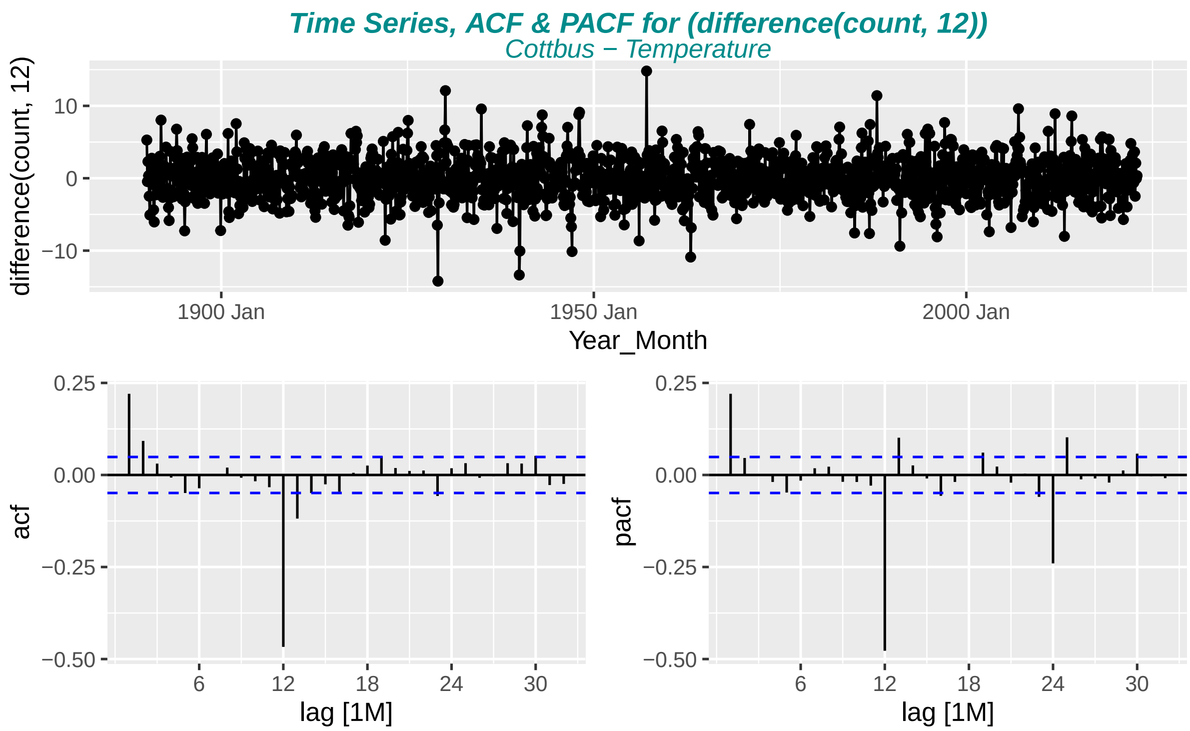
1.1.3 ACF Plots of Differences



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

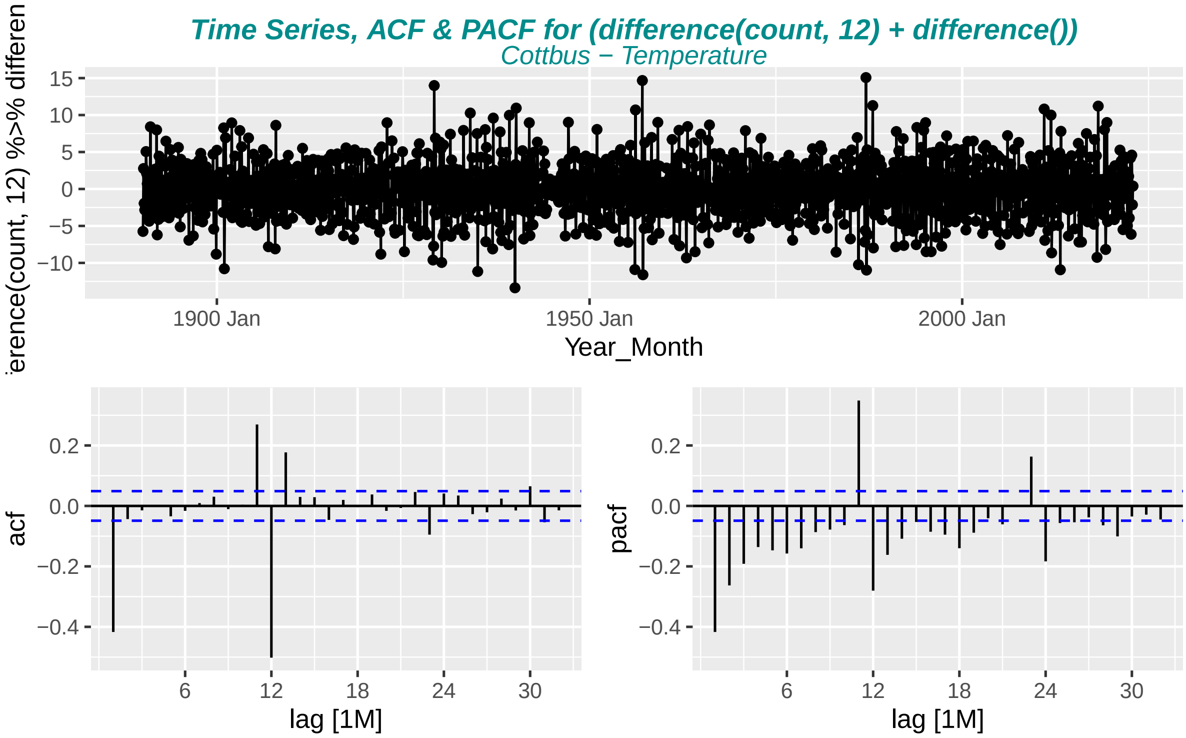



```
#> # A tibble: 1 x 2
#>     Sum   Mean
#>   <dbl>  <dbl>
#> 1  33.0 0.0207
```




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

```
#>    <dbl>  <dbl>
#> 1  33.0 0.0207
```




```
#> # A tibble: 1 x 2
#>      Sum     Mean
#>    <dbl>    <dbl>
#> 1 -4.93 -0.00309
```

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 togehter). 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 togehter). Best Forecast accuracy is with ETS(A,A,A), ETS(A,Ad,A), ETS(A,N,A),

Trend term "N" for Basel Temperature/Precipitation correspondends to a "pure" exponential smooothing 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 intervall 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").

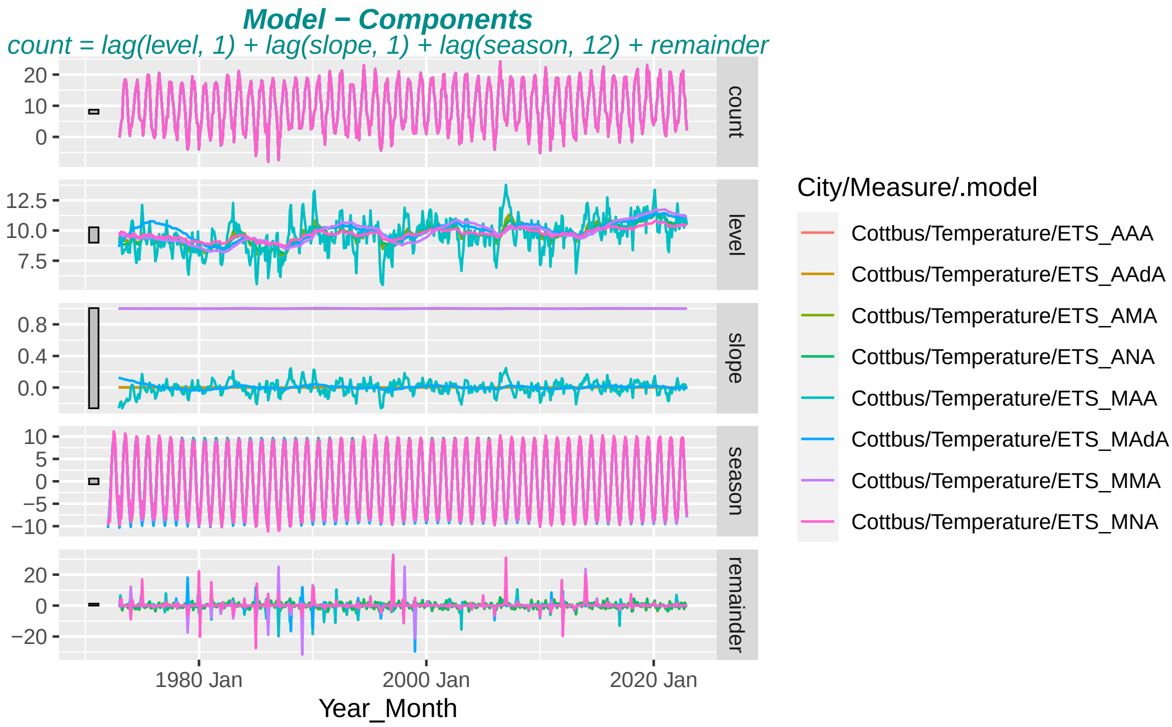
For CO2 and Temperature Data we have a clear seasonal pattern and seasonal term adds always a (more or less) fix amount on level and trend component. Therefore "A" additve term is chosen. For Precipitation the seasonal patttern is only slight. Indead, a multiplicative seasonal term results in "exploding" 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> <dbl>
#> 1 Cottbus Temperature ETS(co~   3.86  -2317. 4664. 4664. 4730.  3.77  3.83  1.50
#> Series: count
#> Model: ETS(A,N,A)
#>   Smoothing parameters:
#>     alpha = 0.05714372
#>     gamma = 0.0001000122
#>
#>   Initial states:
#>     l[0]      s[0]     s[-1]     s[-2]   s[-3]    s[-4]    s[-5]    s[-6]
#>  8.83005 -7.809344 -4.755159 0.0331151 4.66287 9.067921 9.659431 7.916954
#>     s[-7]     s[-8]     s[-9]    s[-10]    s[-11]
#>  4.308015 -0.409232 -5.117443 -8.501505 -9.055624
#>
#>   sigma^2:  3.855
#>
#>      AIC     AICc      BIC
#> 4663.615 4664.437 4729.569
#> 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 Cottbus Temperature ETS_ANA   3.86  -2317. 4664. 4664. 4730.  3.77  3.83 1.50
#> 2 Cottbus Temperature ETS_AMA   3.86  -2316. 4666. 4667. 4741.  3.76  3.84 1.51
#> 3 Cottbus Temperature ETS_AAA   3.86  -2316. 4667. 4668. 4742.  3.76  3.85 1.51
#> 4 Cottbus Temperature ETS_AA~   3.86  -2316. 4668. 4669. 4747.  3.75  3.83 1.51
#> 5 Cottbus Temperature ETS_MAA   6.02  -3549. 7132. 7133. 7207.  4.79  5.33 0.861
#> 6 Cottbus Temperature ETS_MA~   7.27  -3617. 7270. 7271. 7349.  4.28  4.32 0.849
#> 7 Cottbus Temperature ETS_MNA  12.7   -3764. 7558. 7559. 7624.  4.47  4.51 1.06
#> 8 Cottbus Temperature ETS_MMA  13.0   -3774. 7581. 7583. 7656.  4.27  4.31 1.04
```



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

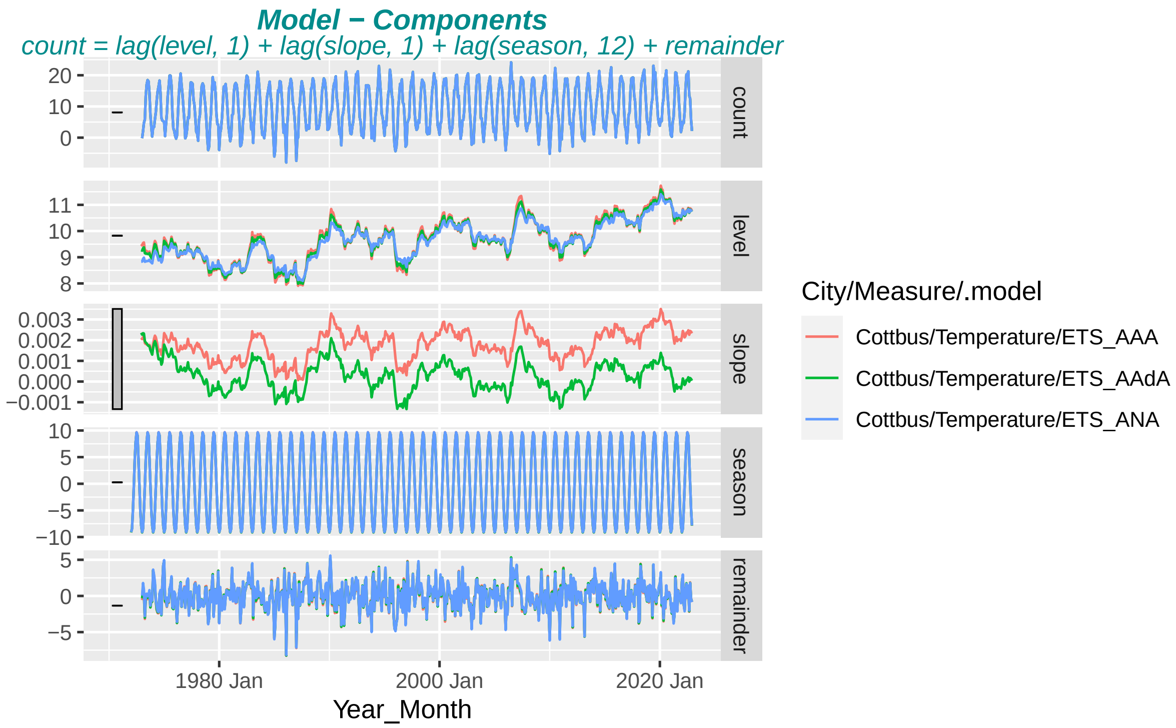
- MAE (Mean absolute error, will lead to forecasts of the median) and
- RMSE (Root mean squared error, lead to forecasts of the mean)

```
#> # 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 Cottbus Temperature ETS_AA~ Trai~  0.0293   1.94  1.51 -68.0  145. 0.724 0.708
#> 2 Cottbus Temperature ETS_AMA Trai~  0.0194   1.94  1.51 -66.9  142. 0.726 0.708
#> 3 Cottbus Temperature ETS_AAA Trai~  0.00372  1.94  1.51 -68.6  145. 0.727 0.709
#> 4 Cottbus Temperature ETS_ANA Trai~  0.0558   1.94  1.50 -68.2  143. 0.723 0.709
#> 5 Cottbus Temperature ETS_MMA Trai~ -0.0618   2.07  1.58 -66.3  151. 0.761 0.755
#> 6 Cottbus Temperature ETS_MA~ Trai~ -0.0692   2.07  1.59 -67.2  157. 0.767 0.756
#> 7 Cottbus Temperature ETS_MNA Trai~  0.0359   2.11  1.62 -52.9  142. 0.781 0.773
#> 8 Cottbus Temperature ETS_MAA Trai~  0.0144   2.19  1.72 -82.9  152. 0.826 0.800
#> # ... with 1 more variable: ACF1 <dbl>
```

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

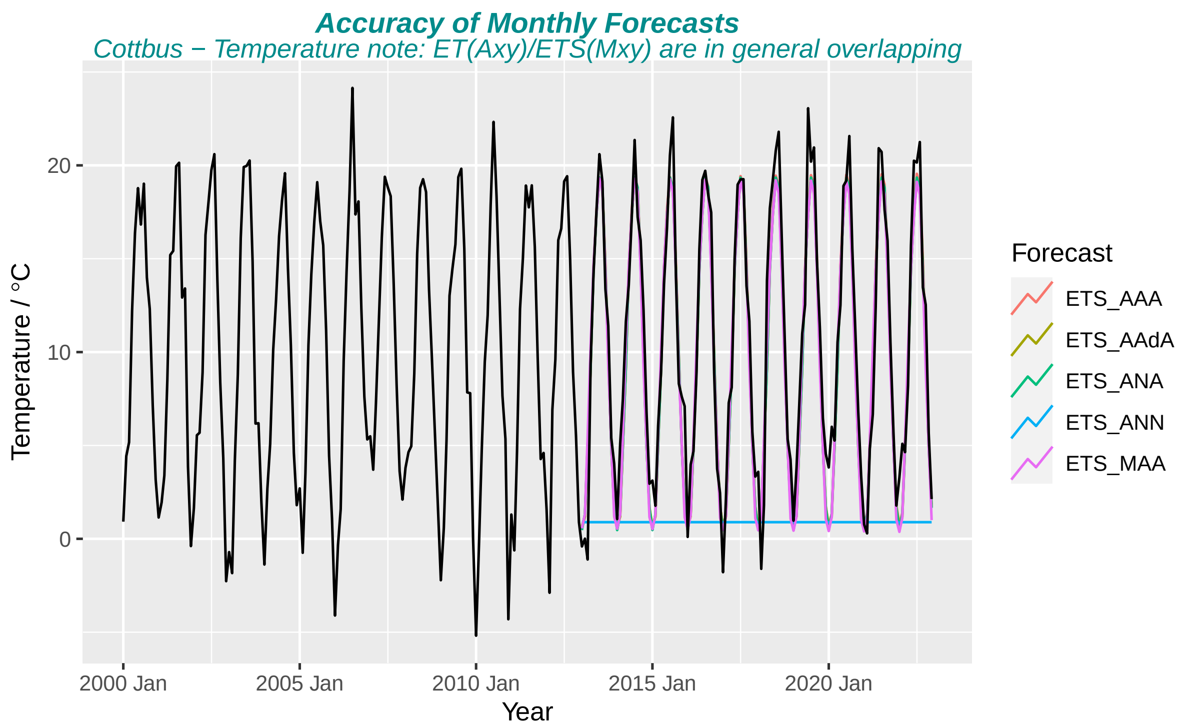
```
#> Null Hypothesis of independence/white noise for residuals - for p < 0.05: reject H_0
#> # A tibble: 8 x 5
#>   City    Measure     .model   lb_stat lb_pvalue
#>   <chr>   <fct>       <chr>      <dbl>     <dbl>
#> 1 Cottbus Temperature ETS_AAA     38.1  1.46e- 1
#> 2 Cottbus Temperature ETS_AMA     38.3  1.43e- 1
#> 3 Cottbus Temperature ETS_AAdA    38.8  1.31e- 1
#> 4 Cottbus Temperature ETS_ANA     40.0  1.06e- 1
#> 5 Cottbus Temperature ETS_MNA     43.0  5.90e- 2
#> 6 Cottbus Temperature ETS_MAA     61.4  6.29e- 4
#> 7 Cottbus Temperature ETS_MAdA    93.0  2.27e- 8
#> 8 Cottbus Temperature ETS_MMA    130.   1.74e-14
```

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

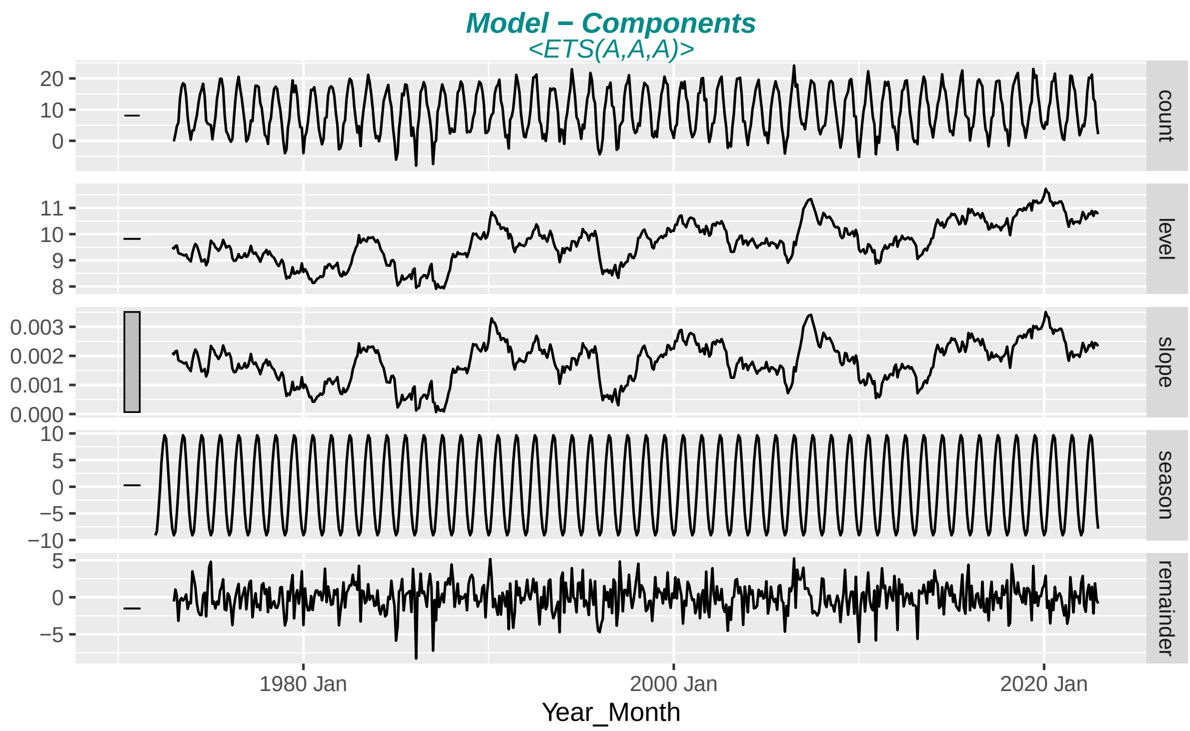
```
#> # A tibble: 5 x 12
#>   .model   City  Measure .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
#>   <chr>    <chr> <fct>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
#> 1 ETS_AAA  Cott~ Temper~ Test  0.767  1.98  1.57 -83.5  135. 0.754 0.722 0.0439
#> 2 ETS_ANA  Cott~ Temper~ Test  0.924  2.05  1.64 -92.9  147. 0.788 0.747 0.0636
#> 3 ETS_AAdA Cott~ Temper~ Test  0.910  2.05  1.63 -83.0  137. 0.787 0.747 0.0490
#> 4 ETS_MAA  Cott~ Temper~ Test  0.949  2.18  1.76 -90.1  149. 0.850 0.795 0.126
#> 5 ETS_ANN  Cott~ Temper~ Test  9.76  12.0   9.94   5.55 169. 4.79  4.37  0.795
```

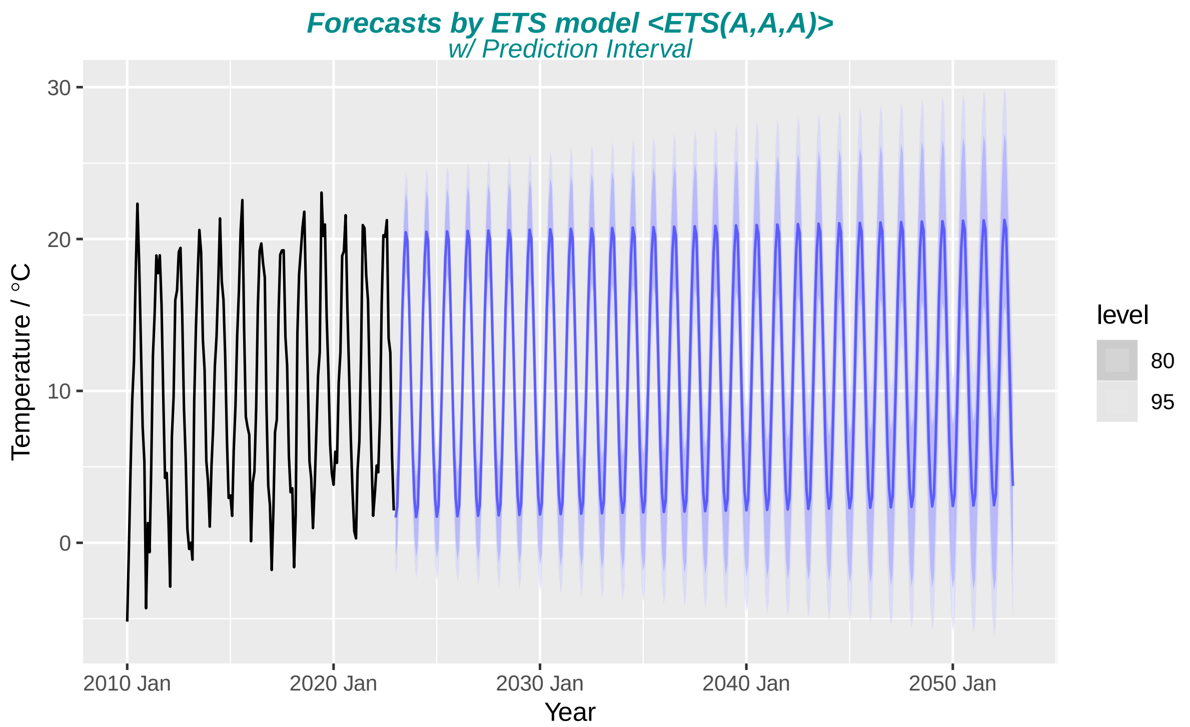


2.2 Forecasting with selected ETS model <ETS(A,A,A)>

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 = 0.08922747
#>     beta  = 0.0001000092
#>     gamma = 0.0001000074
#>
#>   Initial states:
#>      l[0]         b[0]       s[0]      s[-1]     s[-2]     s[-3]     s[-4]     s[-5]
#>  9.497501 0.002108919 -7.869804 -4.876889 -0.04908 4.763188 9.081588 9.671608
#>     s[-6]     s[-7]      s[-8]     s[-9]     s[-10]     s[-11]
#>  7.923166 4.473912 -0.5832224 -5.037017 -8.387651 -9.109798
#>
#>   sigma^2:  3.8634
#>
#>      AIC     AICc      BIC
#> 4666.864 4667.915 4741.612
```

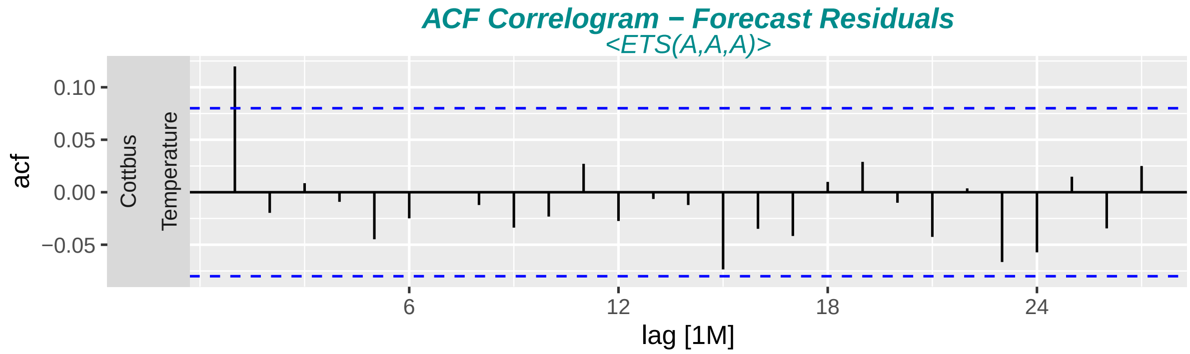


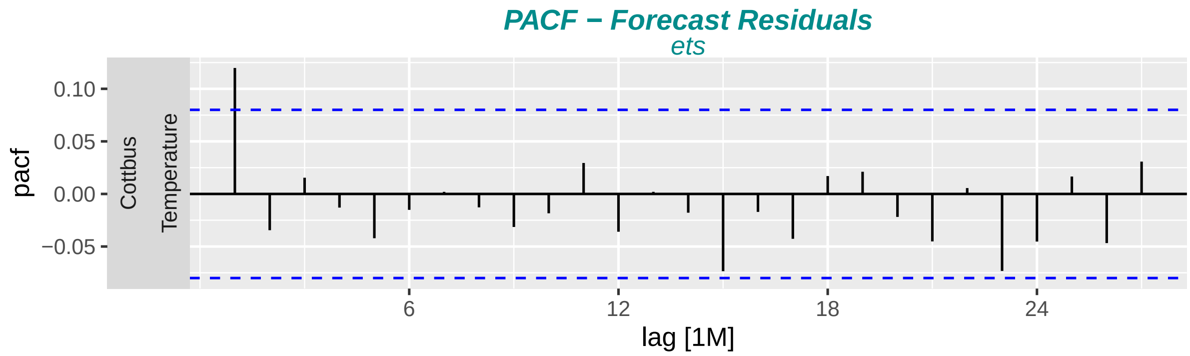


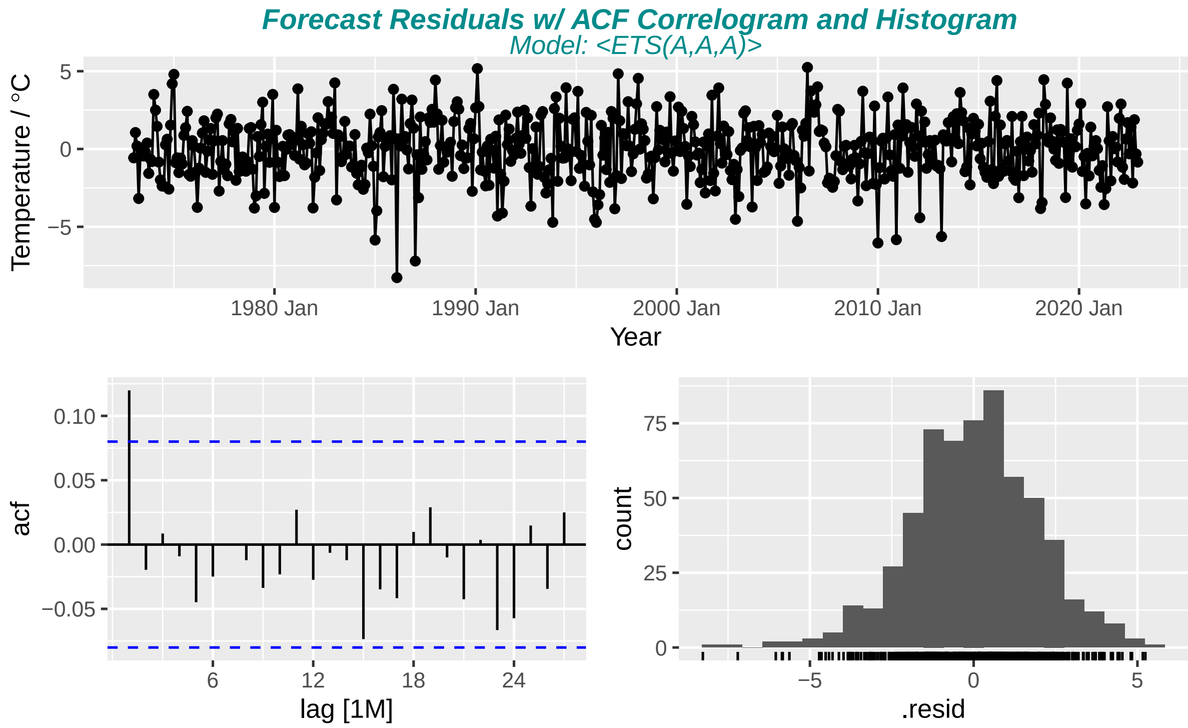
2.2.2 Residual Stationarity

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

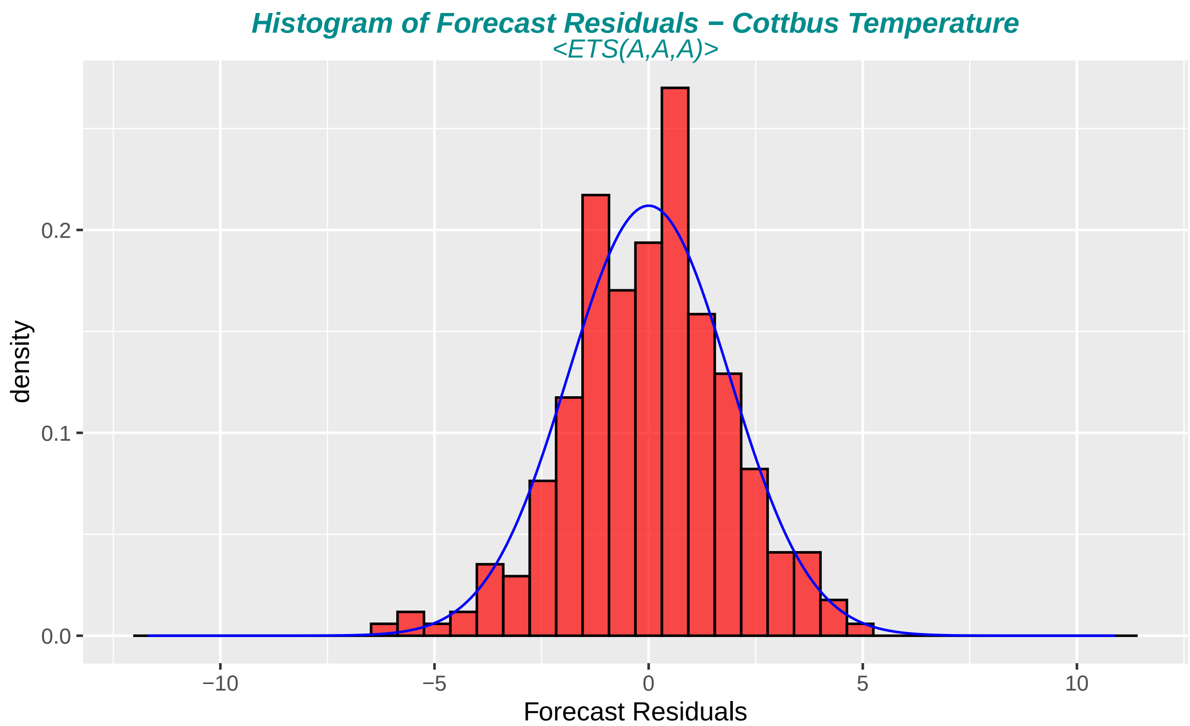






2.2.3 Histogram of forecast residuals with overlaid normal curve

```
#> 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 Cottbus Temperature ets       25.5     0.699
```



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

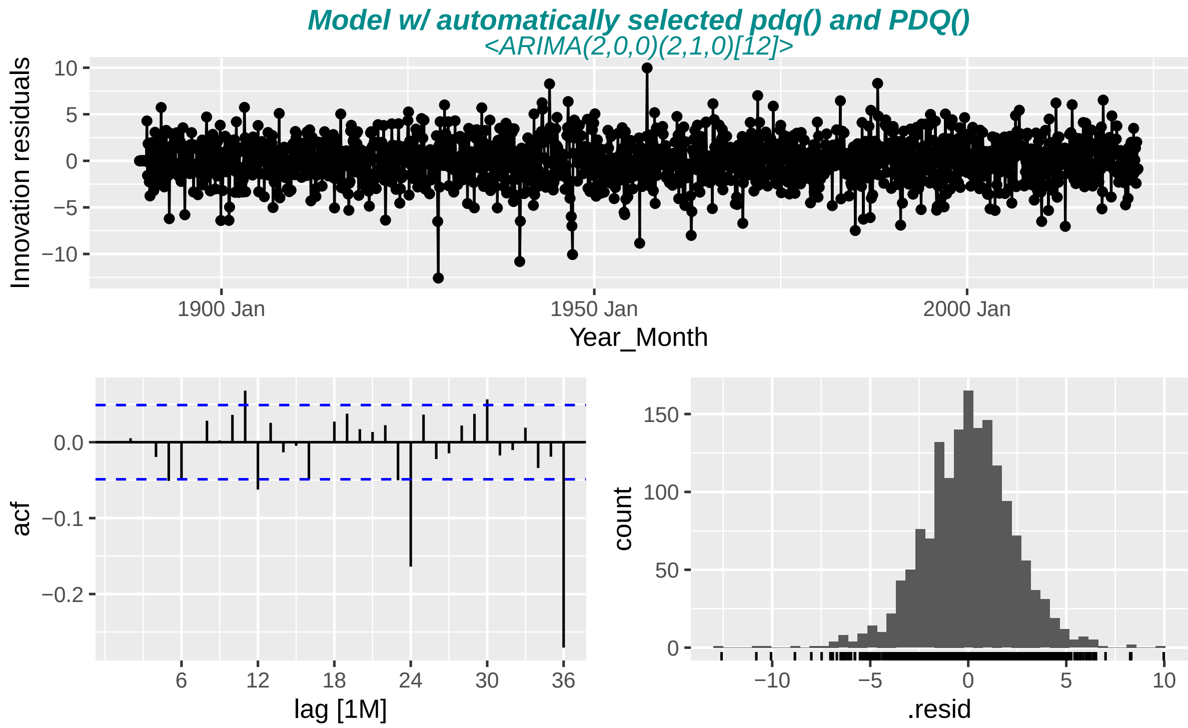
Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, * p is the order (number of time lags) of the autoregressive model * d is the degree of differencing (number of times the data have had past values subtracted) * q is the order of the 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 Cottbus Temperature arima    5.39  -3610. 7229. 7230. 7256. <cpl [26]> <cpl>
#> # ... with abbreviated variable name 1: ma_roots
#> Series: count
```

```
#> Model: ARIMA(2,0,0)(2,1,0)[12]
#>
#> Coefficients:
#>          ar1     ar2     sar1     sar2
#>       0.2546  0.0475  -0.6141  -0.2684
#> s.e.  0.0251  0.0250   0.0242   0.0242
#>
#> sigma^2 estimated as 5.392:  log likelihood=-3609.74
#> AIC=7229.48   AICc=7229.52   BIC=7256.36
```




```
#> 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 Cottbus Temperature ARIMA_2~   3.71  -1245. 2499. 2499. 2521. <cpl>   <cpl>
#>  2 Cottbus Temperature ARIMA_1~   3.71  -1245. 2499. 2500. 2521. <cpl>   <cpl>
#>  3 Cottbus Temperature ARIMA_1~   3.70  -1244. 2499. 2500. 2526. <cpl>   <cpl>
#>  4 Cottbus Temperature ARIMA_0~   3.72  -1245. 2501. 2501. 2523. <cpl>   <cpl>
#>  5 Cottbus Temperature ARIMA_1~   5.06  -1312. 2632. 2632. 2650. <cpl>   <cpl>
#>  6 Cottbus Temperature ARIMA_1~   5.65  -1343. 2696. 2697. 2718. <cpl>   <cpl>
#>  7 Cottbus Temperature ARIMA_2~   5.65  -1343. 2696. 2697. 2718. <cpl>   <cpl>
#>  8 Cottbus Temperature ARIMA_3~   5.54  -1374. 2764. 2764. 2799. <cpl>   <cpl>
#>  9 Cottbus Temperature ARIMA_2~   6.95  -1402. 2813. 2813. 2830. <cpl>   <cpl>
#> 10 Cottbus Temperature ARIMA_1~   7.35  -1421. 2847. 2847. 2860. <cpl>   <cpl>
#> 11 Cottbus Temperature ARIMA_0~   7.36  -1421. 2848. 2848. 2861. <cpl>   <cpl>
#> 12 Cottbus Temperature ARIMA_0~   9.51  -1495. 2994. 2994. 3003. <cpl>   <cpl>
#> 13 Cottbus Temperature ARIMA_1~  10.3   -1517. 3039. 3039. 3047. <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_arima) output). The preference is to use the AICc to selec 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 Cottbus Temperature ARIMA~ Trai~   0.0657    1.90   1.45  -86.8  157.   0.698
#>  2 Cottbus Temperature ARIMA~ Trai~   0.0640    1.90   1.45  -84.3  154.   0.697
#>  3 Cottbus Temperature ARIMA~ Trai~   0.0641    1.90   1.45  -85.6  156.   0.698
#>  4 Cottbus Temperature ARIMA~ Trai~   0.0674    1.90   1.45  -88.3  159.   0.698
#>  5 Cottbus Temperature ARIMA~ Trai~   0.0573    2.22   1.71  -47.5  176.   0.822
#>  6 Cottbus Temperature ARIMA~ Trai~   0.0322    2.34   1.83  -57.4  204.   0.879
#>  7 Cottbus Temperature ARIMA~ Trai~   0.00333   2.35   1.80  -38.6  192.   0.865
#>  8 Cottbus Temperature ARIMA~ Trai~   0.00333   2.35   1.80  -38.6  192.   0.865
#>  9 Cottbus Temperature ARIMA~ Trai~  -0.0130    2.60   2.03  -60.6  214.   0.978
#> 10 Cottbus Temperature ARIMA~ Trai~  -0.0265    2.68   2.01  -11.8  221.   0.968
#> 11 Cottbus Temperature ARIMA~ Trai~  -0.0263    2.68   2.01  -14.7  221.   0.969
#> 12 Cottbus Temperature ARIMA~ Trai~  -0.00915   3.05   2.34 -103.   317.   1.13
#> 13 Cottbus Temperature ARIMA~ Trai~  -0.00749   3.17   2.45   15.1  283.   1.18
#> 14 Cottbus Temperature ARIMA~ Trai~ NaN       NaN    NaN     NaN    NaN  NaN
#> # ... with 2 more variables: RMSSE <dbl>, ACF1 <dbl>
```

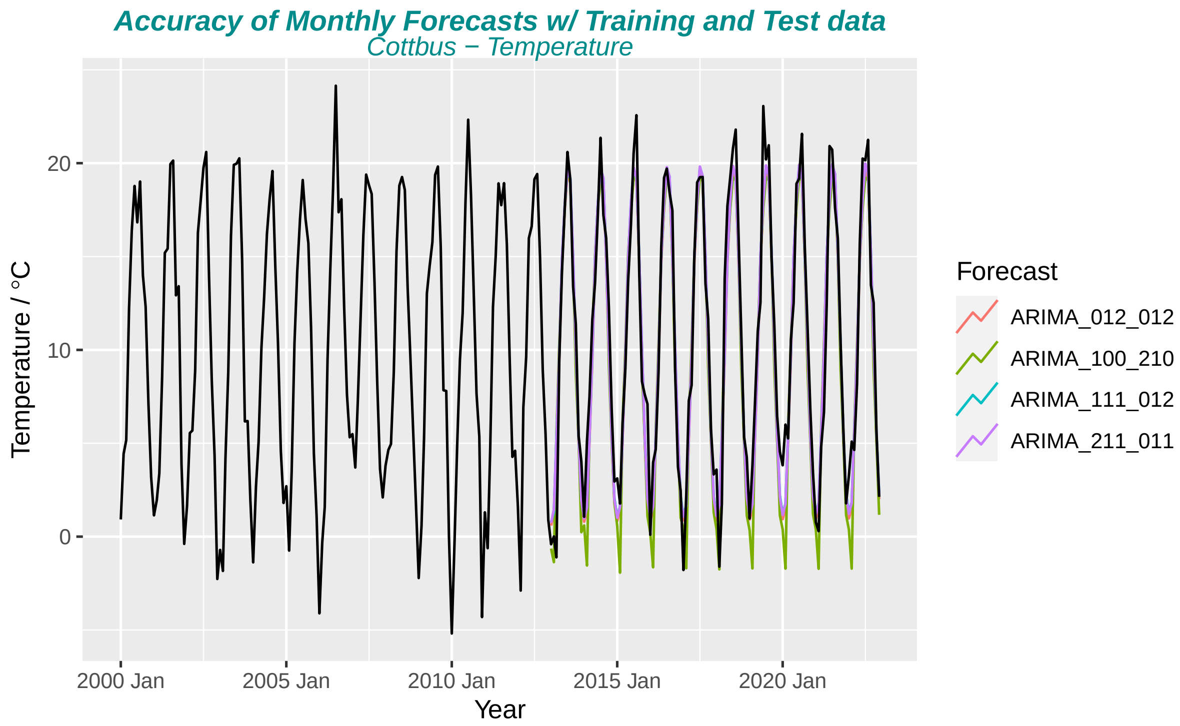
3.1.2 Ljung-Box Test - independence/white noise of the forecasts residuals

```
#> 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 Cottbus Temperature ARIMA_211_011    27.4  6.02e- 1
#>  2 Cottbus Temperature ARIMA_111_012    27.7  5.89e- 1
#>  3 Cottbus Temperature ARIMA_111_112    28.7  5.35e- 1
#>  4 Cottbus Temperature ARIMA_012_012    29.0  5.18e- 1
#>  5 Cottbus Temperature ARIMA_301_200    88.1  1.28e- 7
#>  6 Cottbus Temperature ARIMA_100_210    98.4  3.31e- 9
#>  7 Cottbus Temperature ARIMA_100_110   107.   1.26e-10
#>  8 Cottbus Temperature ARIMA_200_110   107.   1.26e-10
#>  9 Cottbus Temperature ARIMA_010_110   263.   0
#> 10 Cottbus Temperature ARIMA_012_010   179.   0
#> 11 Cottbus Temperature ARIMA_110_010   314.   0
#> 12 Cottbus Temperature ARIMA_111_010   179.   0
#> 13 Cottbus Temperature ARIMA_210_110   167.   0
#> 14 Cottbus 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>
```

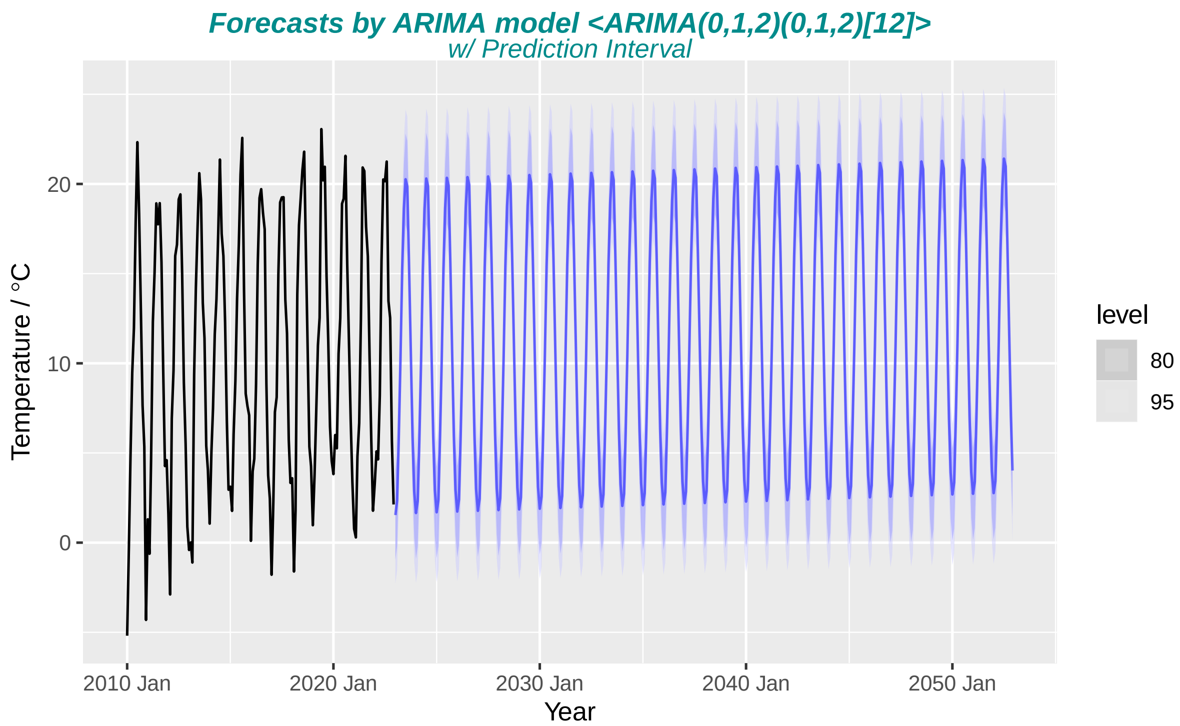
```
#> 1 ARIMA_1~ Cott~ Temper~ Test  0.414  1.87  1.46 -102.   148. 0.705 0.682 0.0394
#> 2 ARIMA_2~ Cott~ Temper~ Test  0.419  1.88  1.47 -116.   162. 0.710 0.684 0.0523
#> 3 ARIMA_0~ Cott~ Temper~ Test  0.690  1.95  1.54  -84.5  134. 0.743 0.712 0.0433
#> 4 ARIMA_1~ Cott~ Temper~ Test  1.01   2.46  1.87  144.   158. 0.902 0.897 0.0103
```



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

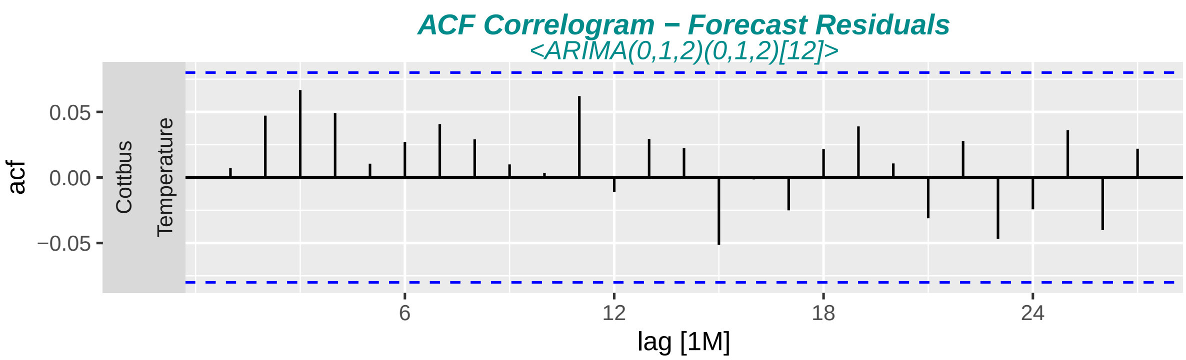
```
#> Provide model coefficients by report(fit_model)
#> Series: count
#> Model: ARIMA(0,1,2)(0,1,2)[12]
#>
#> Coefficients:
#>           ma1      ma2     sma1     sma2
#>       -0.8173  -0.1826  -0.9818  -0.0180
#> s.e.   0.0443   0.0398   0.0863   0.0429
#>
#> sigma^2 estimated as 3.72:  log likelihood=-1245.36
#> AIC=2500.73   AICc=2500.83   BIC=2522.6
```

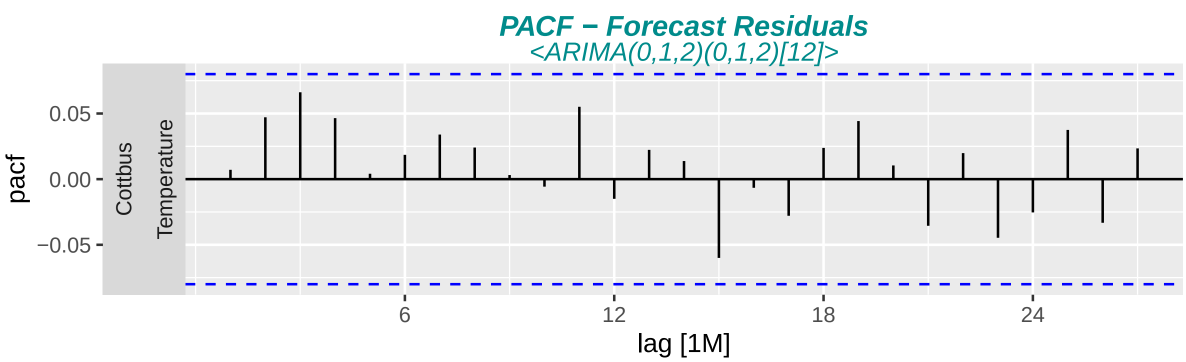


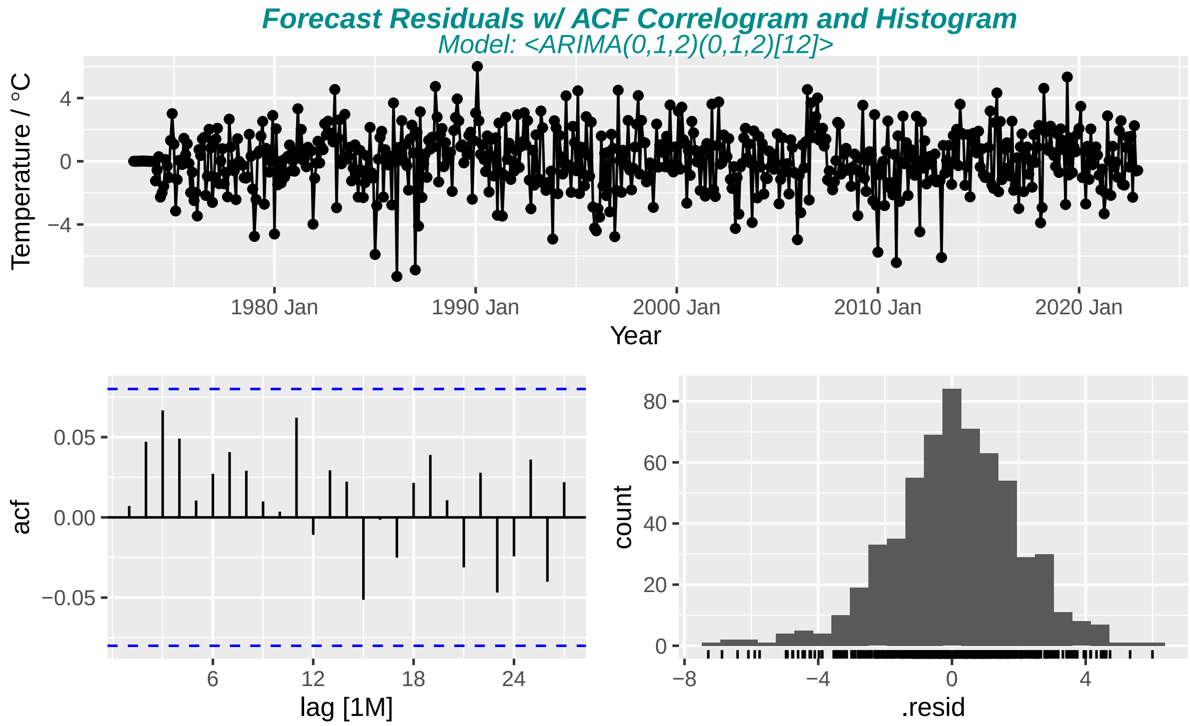
3.2.2 Residual Stationarity

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

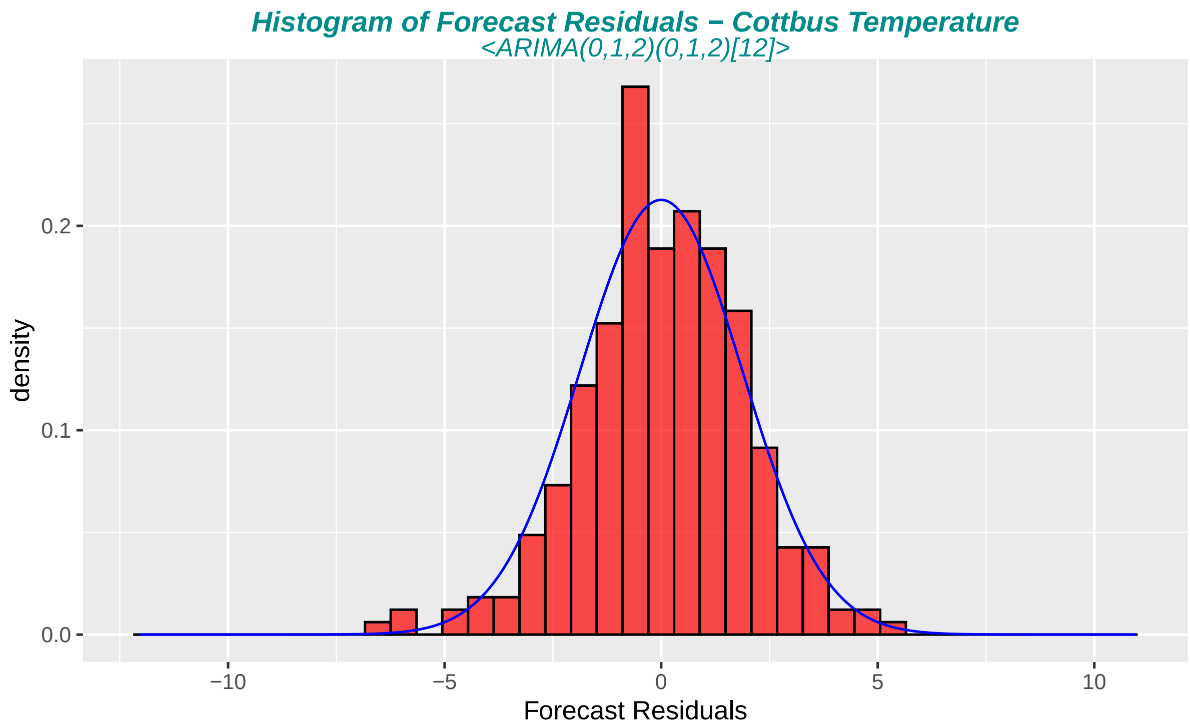






3.2.3 Histogram of forecast residuals with overlaid normal curve

```
#> 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 Cottbus Temperature arima     19.6     0.926
```



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 rsp. ARIMA model the RMSE / MAE values. Lower values indicate a more accurate model based on the test set RMSE, . . . , MASE.

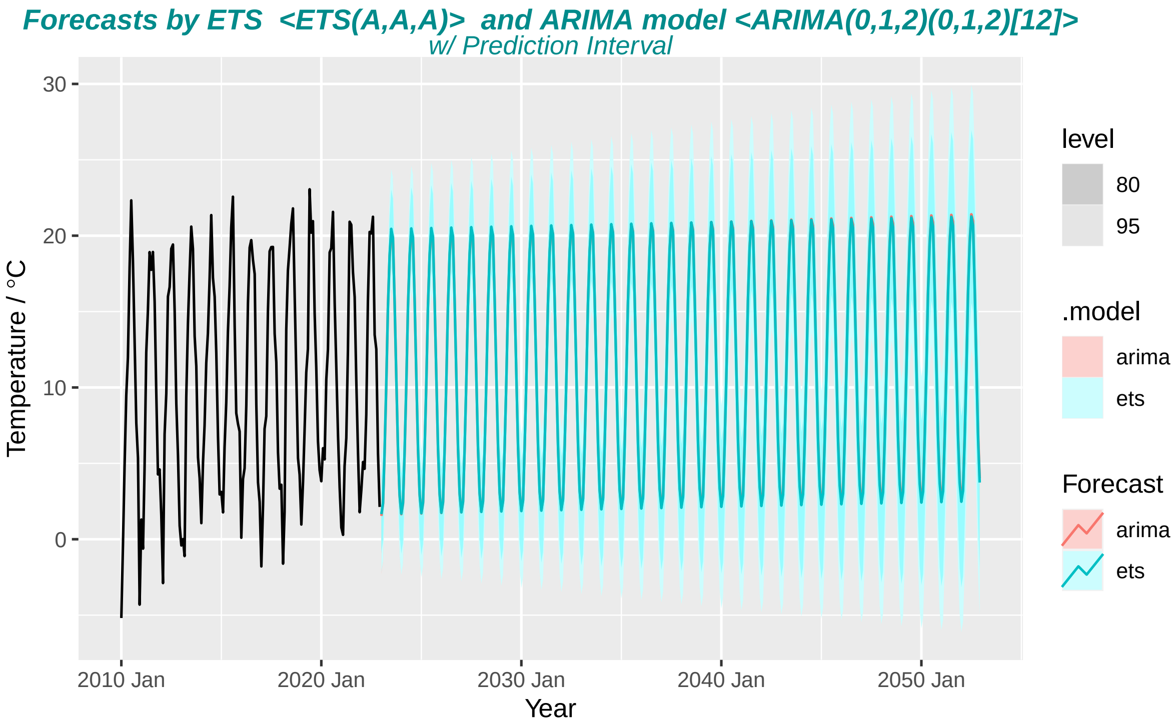
- Residual Accuracy with one-step-ahead fitted residuals
- Forecast Accuracy with Training/Test Data

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

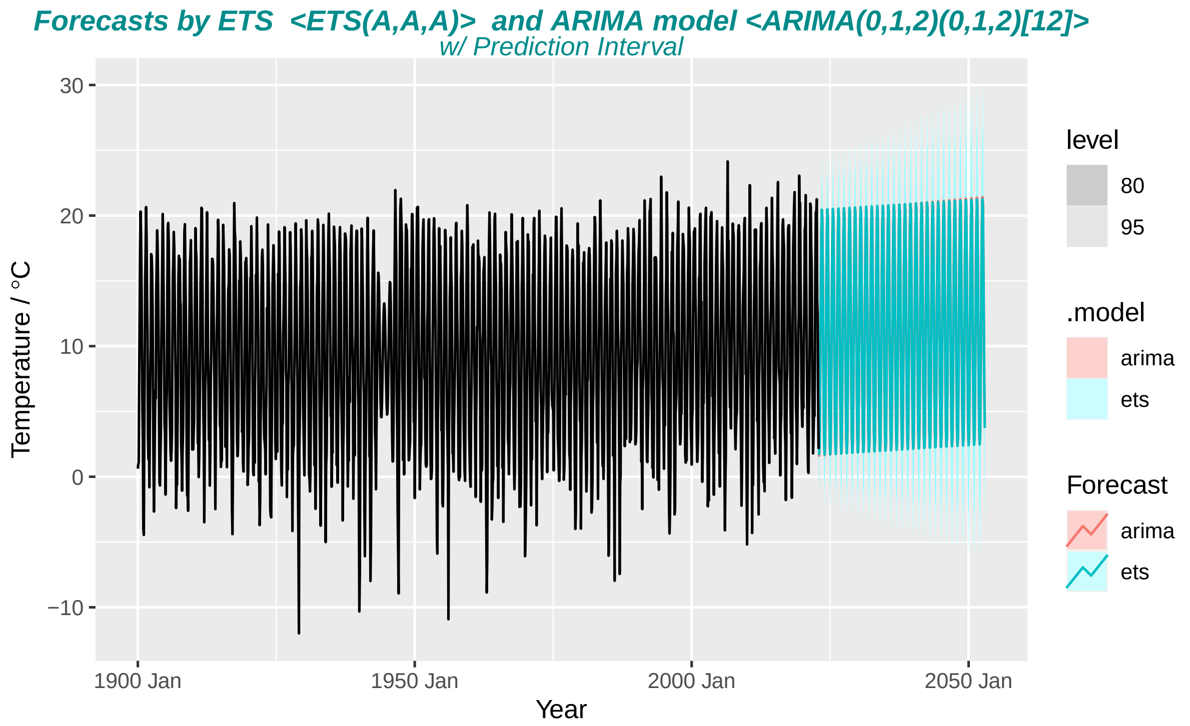
```
#> # A tibble: 4 x 12
#>   City  Measure .model .type      ME  RMSE   MAE   MPE  MAPE  MASE RMSSE    ACF1
#>   <chr> <fct>   <chr>  <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
#> 1 Cott~ Temper~ ets    Trai~ 0.00372  1.94  1.51 -68.6  145. 0.727 0.709 0.120
#> 2 Cott~ Temper~ arima  Trai~ 0.0674   1.90  1.45 -88.3  159. 0.698 0.695 0.00712
#> 3 Cott~ Temper~ ETS_A~ Test  0.767    1.98  1.57 -83.5  135. 0.754 0.722 0.0439
#> 4 Cott~ Temper~ ARIMA~ Test  0.690    1.95  1.54 -84.5  134. 0.743 0.712 0.0433
```

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>     <mth>      <dist> <dbl>                         <hilo>
#> 1 Cottbus Temperature arima  2023 Jan N(1.5, 3.8)  1.54 [-0.9633695, 4.041349]80
#> 2 Cottbus Temperature arima  2023 Feb N(2.3, 3.9)  2.33 [-0.2116345, 4.880317]80
#> 3 Cottbus Temperature arima  2023 Mrz N(5.7, 3.9)  5.67 [ 3.1194228, 8.211374]80
#> 4 Cottbus Temperature ets    2023 Jan N(1.7, 3.9)  1.66 [-0.8596157, 4.178277]80
#> 5 Cottbus Temperature ets    2023 Feb N(2.4, 3.9)  2.38 [-0.1454615, 4.912491]80
#> 6 Cottbus Temperature ets    2023 Mrz N(5.7, 3.9)  5.74 [ 3.1977534, 8.275731]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>     <mth>      <dist> <dbl>                         <hilo>
#> 1 Cottbus Temperature arima  2052 Okt  N(12, 4.1) 11.9  [ 9.327194, 14.486248]80
#> 2 Cottbus Temperature arima  2052 Nov N(7.1, 4.1)  7.06 [ 4.479732,  9.638786]80
#> 3 Cottbus Temperature arima  2052 Dez   N(4, 4.1)  4.02 [ 1.441216,  6.600464]80
#> 4 Cottbus Temperature ets    2052 Okt   N(12, 20) 11.6  [ 5.844455, 17.261379]80
#> 5 Cottbus Temperature ets    2052 Nov  N(6.7, 20)  6.73 [ 1.010146, 12.444434]80
#> 6 Cottbus Temperature ets    2052 Dez  N(3.7, 20)  3.74 [-1.989384,  9.462268]80
#> # ... with 1 more variable: `95%` <hilo>, and abbreviated variable name
#> #   1: Year_Month
```




```
#> # A tibble: 180 x 5
#> # Groups:   City, Measure, .model, Year [60]
#>    City    Measure     .model  Year Year_avg
#>    <chr>   <fct>       <chr>  <dbl>    <dbl>
#>  1 Cottbus Temperature arima   2023     1.54
#>  2 Cottbus Temperature arima   2023     2.33
#>  3 Cottbus Temperature arima   2023     5.67
#>  4 Cottbus Temperature arima   2024     1.65
#>  5 Cottbus Temperature arima   2024     2.32
```

```
#>  6 Cottbus Temperature arima   2024     5.72
#>  7 Cottbus Temperature arima   2025     1.69
#>  8 Cottbus Temperature arima   2025     2.36
#>  9 Cottbus Temperature arima   2025     5.76
#> 10 Cottbus Temperature arima   2026     1.73
#> # ... 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 Cottbus Temperature arima   2023    10.8
#>  2 Cottbus Temperature arima   2023     5.91
#>  3 Cottbus Temperature arima   2023     2.86
#>  4 Cottbus Temperature arima   2024    10.8
#>  5 Cottbus Temperature arima   2024     5.95
#>  6 Cottbus Temperature arima   2024     2.91
#>  7 Cottbus Temperature arima   2025    10.8
#>  8 Cottbus Temperature arima   2025     5.99
#>  9 Cottbus Temperature arima   2025     2.95
#> 10 Cottbus 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 Cottbus Temperature arima     29.0     0.518
#> 2 Cottbus Temperature ets       38.1     0.146
```

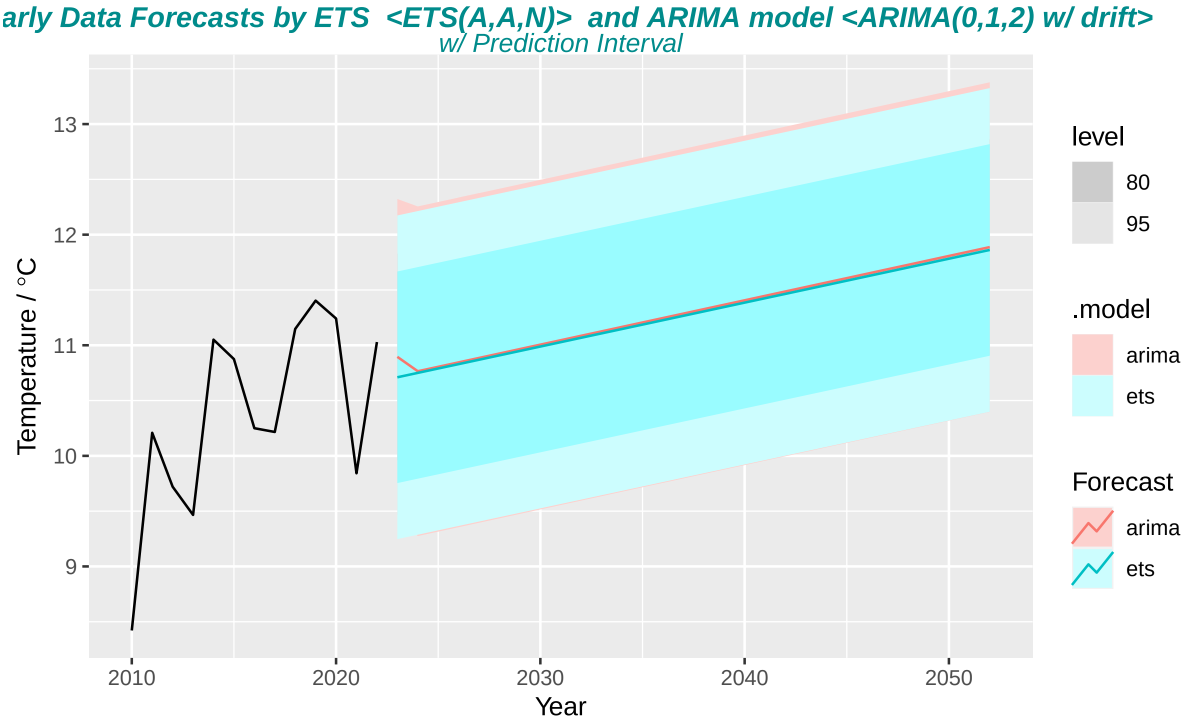
5 Yearly Data Forecasts with ARIMA and ETS

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.

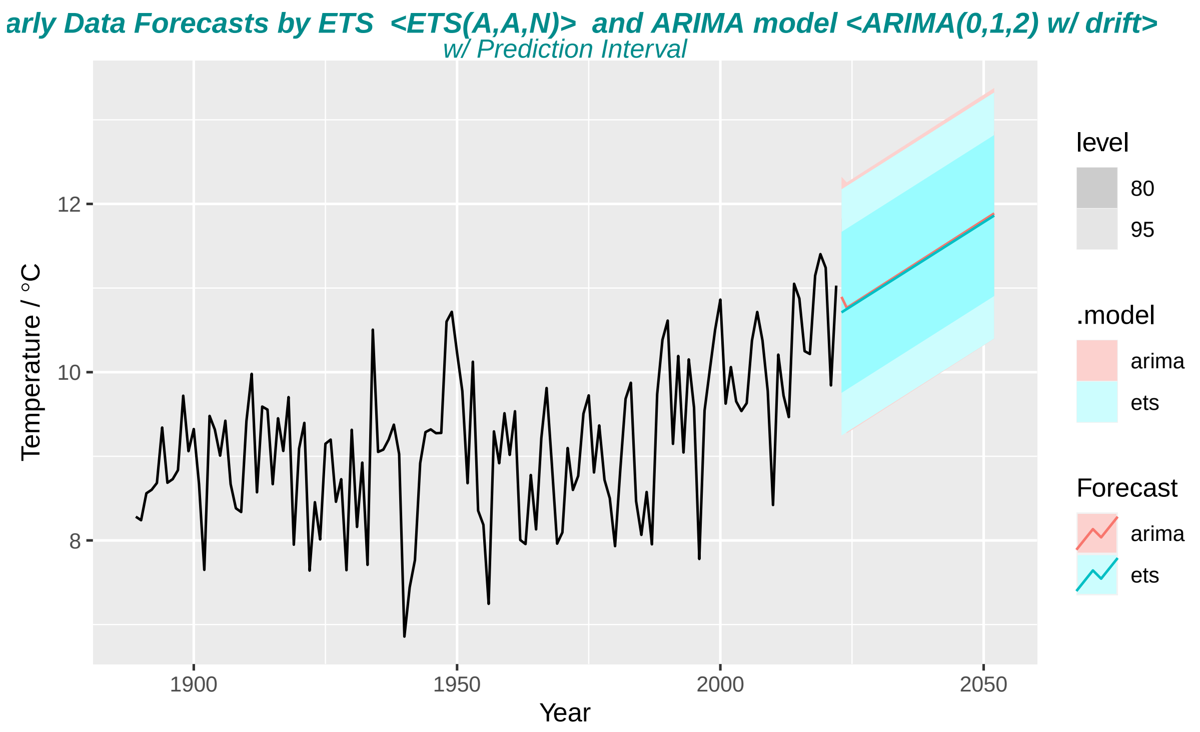
The ETS model $< ETS(A, A, N) >$ with seasonal term change "A" -> "N" is chosen. For ARIMA models the seasonal term (P,D,Q)m has to be taken out and an optimal ARIMA(p,1,q) with one differencing (d=1) is selected. However, for Mauna Loa two times differncing had to be selected $CO_2 <ARIMA(0,2,1) w/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

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




```
#> # A tsibble: 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 Cottbus Temperature arima   2023 N(11, 0.53)  10.9 [9.961655, 11.82869]80
#> 2 Cottbus Temperature arima   2024 N(11, 0.58)  10.8 [9.790333, 11.73954]80
#> 3 Cottbus Temperature arima   2025 N(11, 0.58)  10.8 [9.830426, 11.77963]80
#> 4 Cottbus Temperature ets     2023 N(11, 0.56)  10.7 [9.753664, 11.66702]80
#> 5 Cottbus Temperature ets     2024 N(11, 0.56)  10.8 [9.793373, 11.70673]80
#> 6 Cottbus Temperature ets     2025 N(11, 0.56)  10.8 [9.833081, 11.74644]80
#> # ... with 1 more variable: `95%` <hilo>
#> # A tsibble: 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 Cottbus Temperature arima   2050 N(12, 0.58)  11.8 [10.83275, 12.78196]80
#> 2 Cottbus Temperature arima   2051 N(12, 0.58)  11.8 [10.87285, 12.82205]80
#> 3 Cottbus Temperature arima   2052 N(12, 0.58)  11.9 [10.91294, 12.86215]80
#> 4 Cottbus Temperature ets     2050 N(12, 0.56)  11.8 [10.82572, 12.73927]80
#> 5 Cottbus Temperature ets     2051 N(12, 0.56)  11.8 [10.86543, 12.77899]80
#> 6 Cottbus Temperature ets     2052 N(12, 0.56)  11.9 [10.90513, 12.81871]80
#> # ... with 1 more variable: `95%` <hilo>
```



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 Cottbus Temperature arima     39.2   0.121
#> 2 Cottbus Temperature ets       52.3   0.00712
```

6 Backup