Climate Data Forecasting -

Atmospheric CO_2 Concentration / Temperature / Precipitation

Wolfgang Vollmer

2025-01-03

Contents

1 Forecasting of Hohenpeissenberg - 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

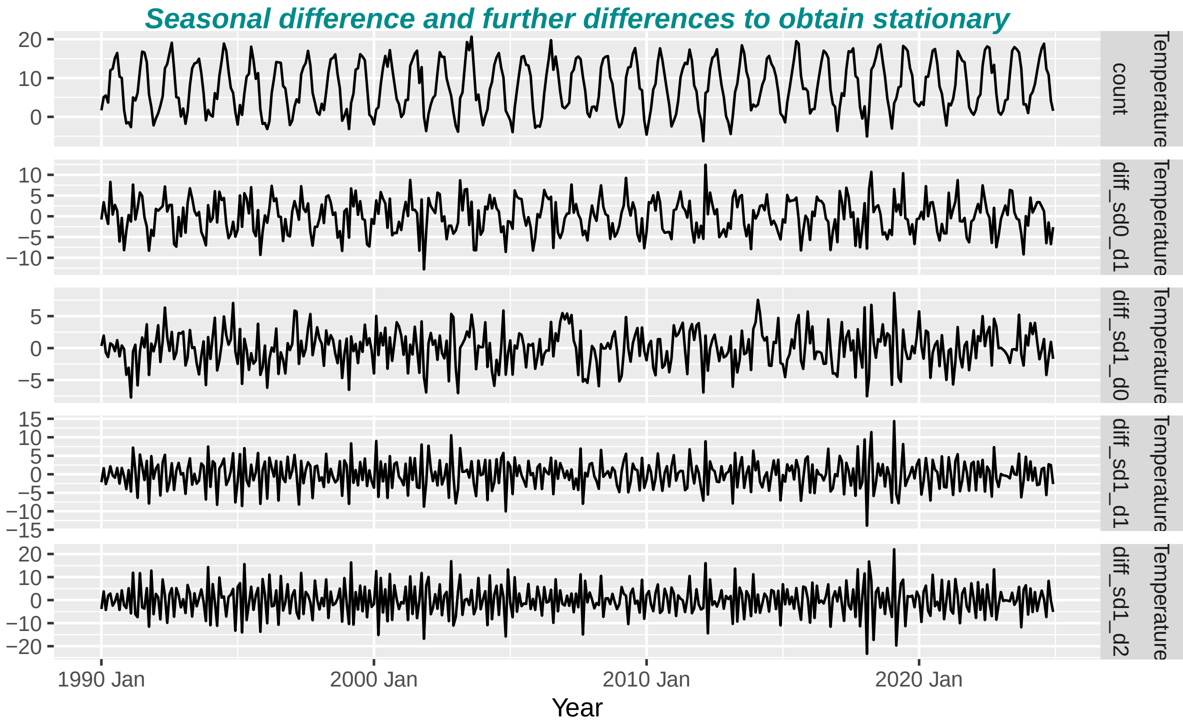
The model fit residuals have to be stationary. For good forecasting this has to be verified with residual diagnostics.

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   9589.         0
#> Ljung-Box test on (difference(count, 12))
#> # A tibble: 1 x 3
#>   Measure     lb_stat   lb_pvalue
#>   <fct>         <dbl>       <dbl>
#> 1 Temperature    50.6 0.000000211
#> Ljung-Box test on (difference(count, 12) + difference())
#> # A tibble: 1 x 3
#>   Measure     lb_stat lb_pvalue
#>   <fct>         <dbl>     <dbl>
#> 1 Temperature    620.         0
```

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

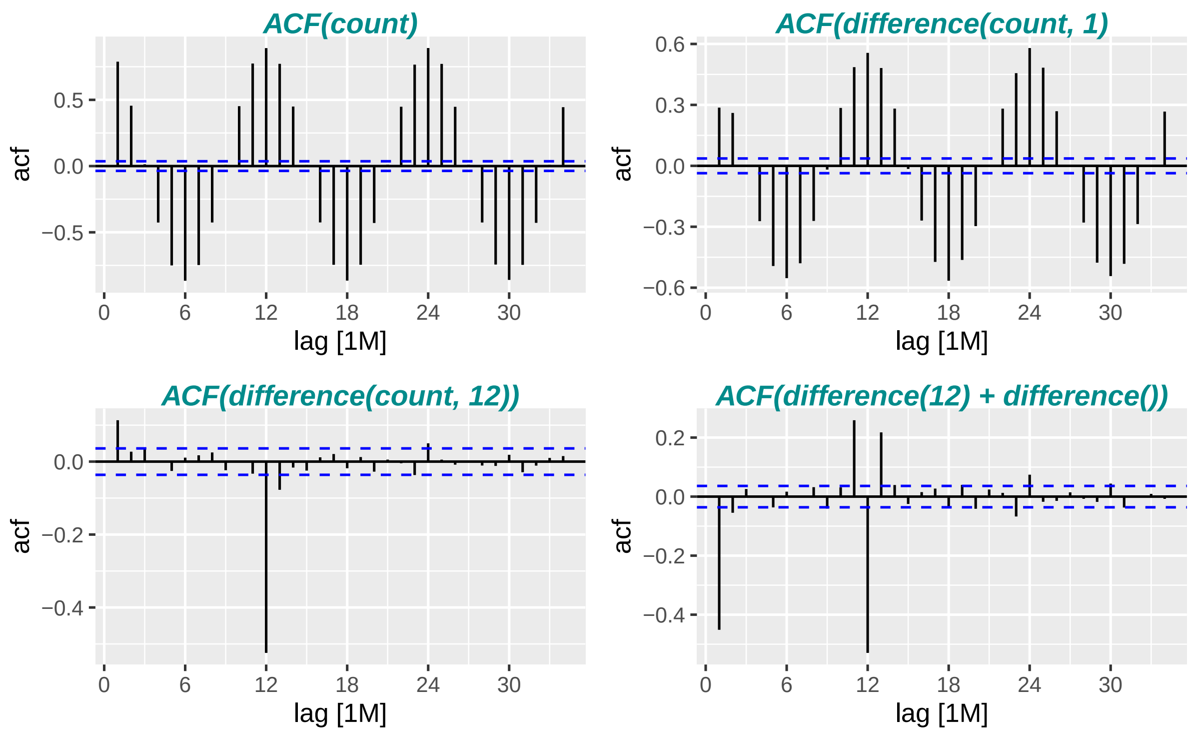
kpss test of stationary
Null Hypothesis of stationary in a given time series
=> H_0 to be rejected for $p < \alpha = 0.05$

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.

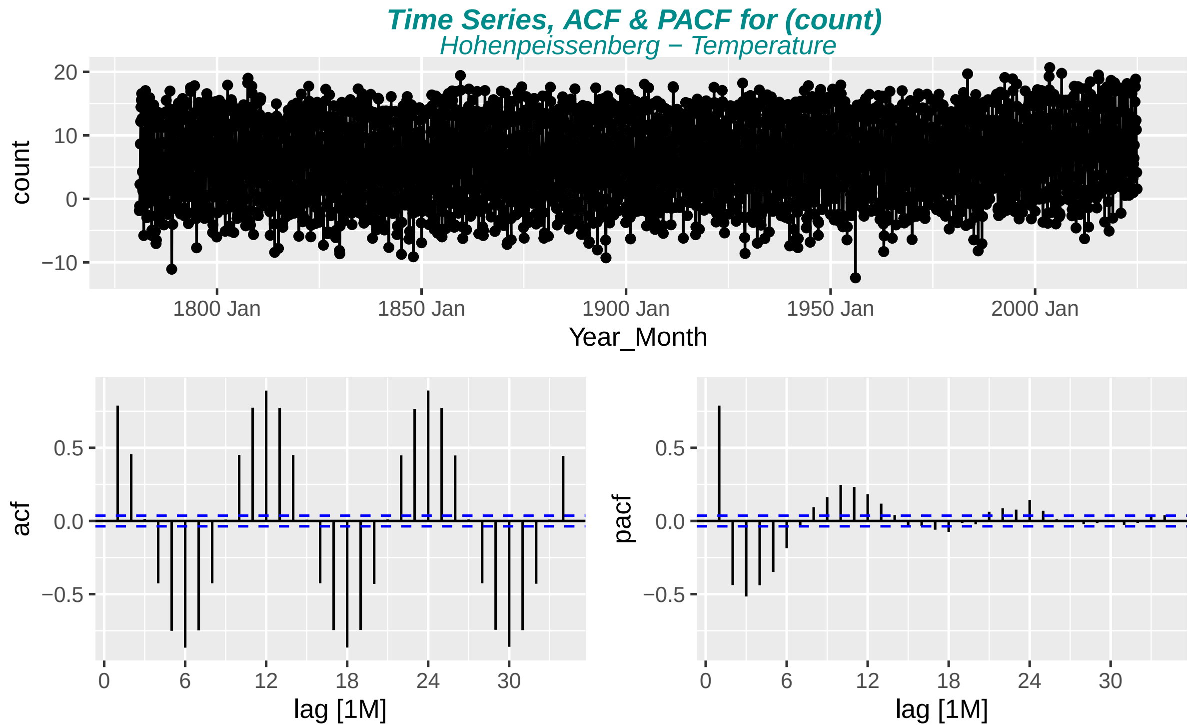
- returns 1 => for stationarity one seasonal difference rsp. difference is required

```
#> 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      3.91        0.01       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    0.0203         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   0.00186         0.1       0      0
```

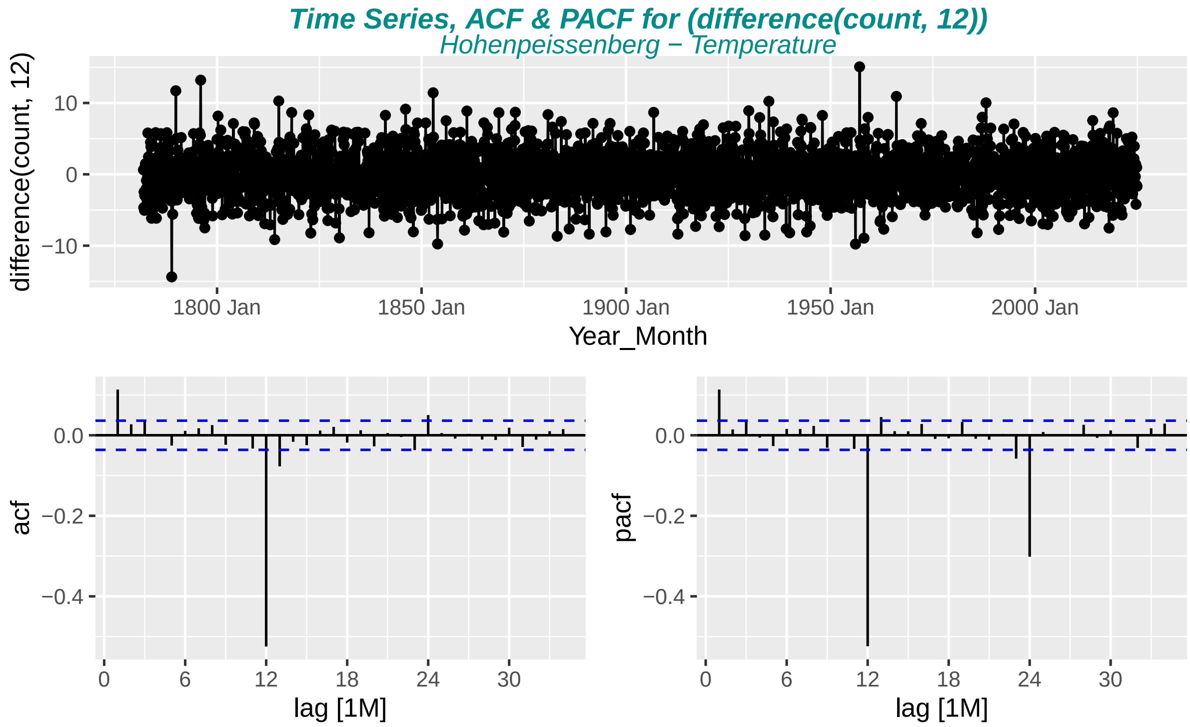
1.1.3 ACF Plots of Differences



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

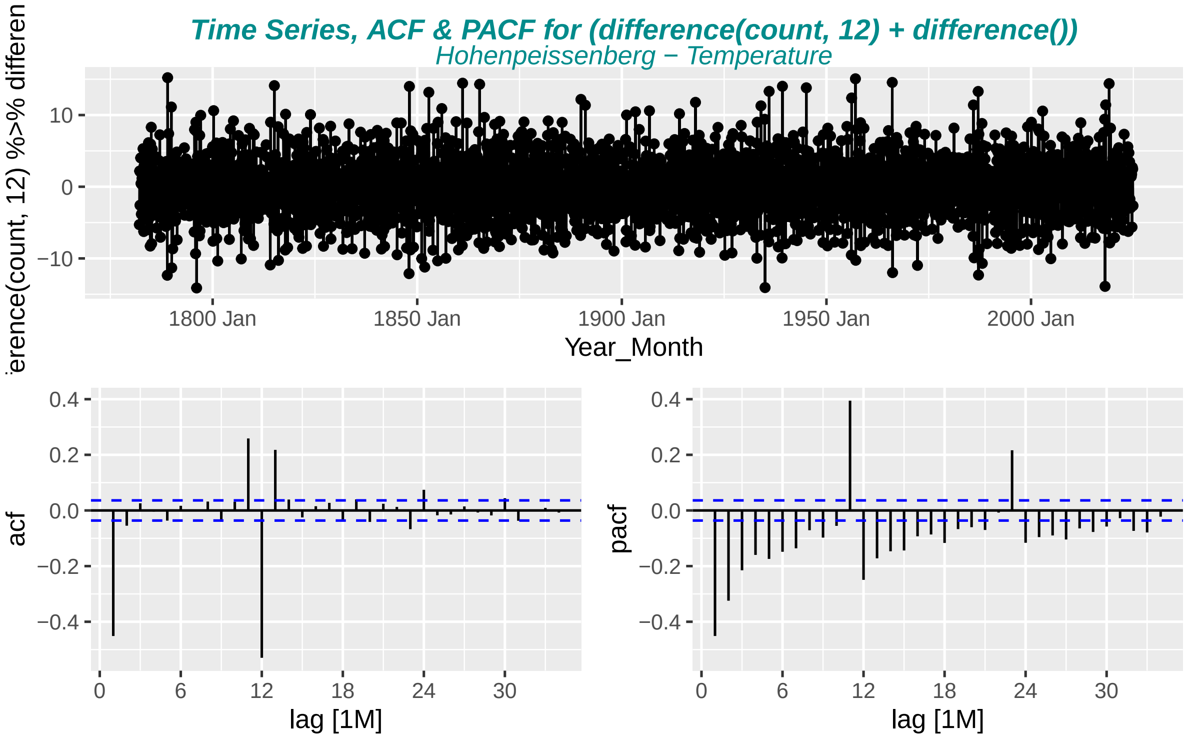



```
#> # A tibble: 1 x 4
#> # Groups:   City [1]
#>   City             Measure       Sum    Mean
#>   <chr>            <fct>       <dbl>   <dbl>
#> 1 Hohenpeissenberg Temperature  27.7 0.00951
```




```
#> # A tibble: 1 x 4
```

```
#> # Groups:   City [1]
#>   City              Measure       Sum    Mean
#>   <chr>             <fct>       <dbl>   <dbl>
#> 1 Hohenpeissenberg Temperature  27.7 0.00951
```




```
#> # A tibble: 1 x 4
#> # Groups:   City [1]
#>   City              Measure       Sum      Mean
#>   <chr>             <fct>       <dbl>     <dbl>
#> 1 Hohenpeissenberg Temperature -2.31 -0.000792
```

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").

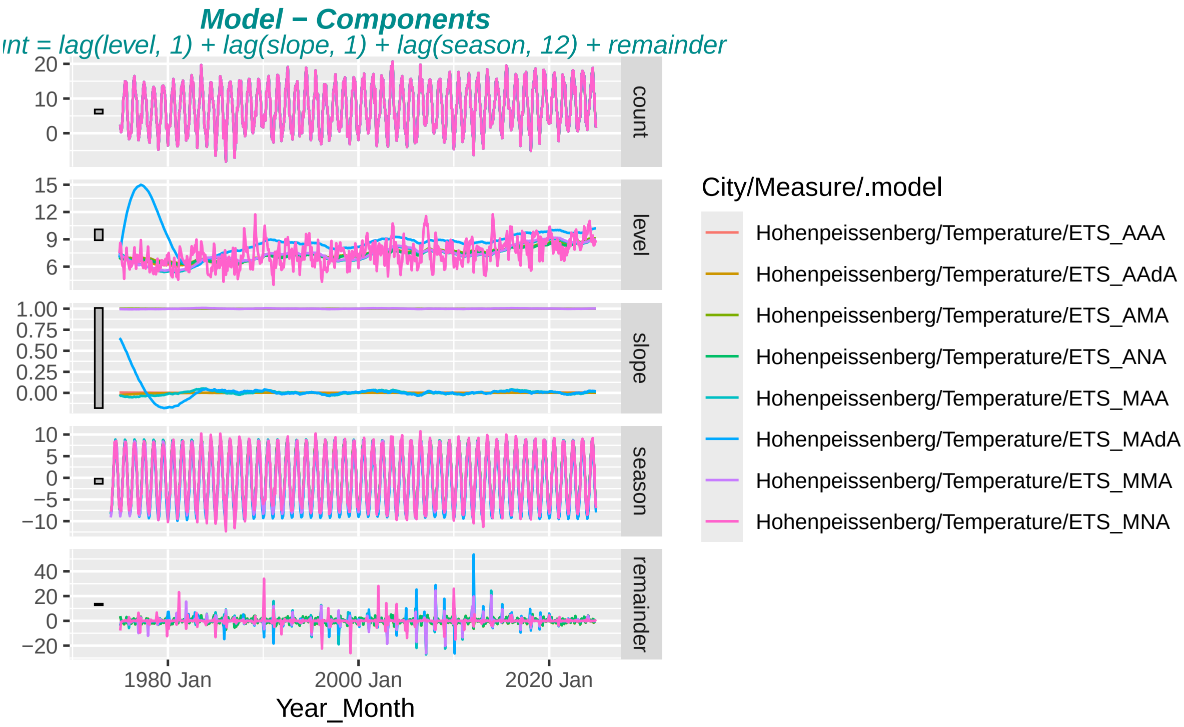
For CO2 and Temperature Data we have a clear seasonal pattern and seasonal term adds always a (more or less) fix amount on level and trend component. Therefore "A" additve term is chosen. For Precipitation the seasonal patttern is only slight. Indead, a multiplicative seasonal term results in "exploding" 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 Hohenpeisse~ Temper~ ETS(c~   4.11  -2336. 4703. 4703. 4769.  4.02  4.02  1.60
#> Series: count
#> Model: ETS(A,N,A)
#>   Smoothing parameters:
#>     alpha = 0.02487034
#>     gamma = 0.0001000122
#>
#>   Initial states:
#>      l[0]      s[0]     s[-1]    s[-2]    s[-3]    s[-4]    s[-5]    s[-6]
#>  6.792878 -6.766858 -4.147659 1.035374 4.596599 8.446444 8.670254 6.623345
#>     s[-7]     s[-8]     s[-9]    s[-10]    s[-11]
#>  3.197426 -1.175033 -4.600845 -7.704995 -8.174053
#>
#>   sigma^2:  4.1141
#>
#>      AIC     AICc      BIC
```

```
#> 4702.644 4703.466 4768.598
#> 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 Hohenpeisse~ Temper~ ETS_A~   4.11   -2336. 4703. 4703. 4769.  4.02  4.02  1.60
#> 2 Hohenpeisse~ Temper~ ETS_A~   4.12   -2335. 4705. 4706. 4780.  4.01  4.01  1.59
#> 3 Hohenpeisse~ Temper~ ETS_A~   4.12   -2336. 4705. 4706. 4780.  4.01  4.01  1.59
#> 4 Hohenpeisse~ Temper~ ETS_A~   4.13   -2336. 4707. 4709. 4787.  4.01  4.01  1.59
#> 5 Hohenpeisse~ Temper~ ETS_M~  10.8    -3534. 7102. 7104. 7177.  4.44  4.44  1.21
#> 6 Hohenpeisse~ Temper~ ETS_M~  12.6    -3561. 7156. 7157. 7231.  4.34  4.35  1.29
#> 7 Hohenpeisse~ Temper~ ETS_M~  12.8    -3622. 7274. 7275. 7340.  6.95  7.04  1.24
#> 8 Hohenpeisse~ Temper~ ETS_M~  17.7    -3695. 7426. 7427. 7505.  8.16  8.32  1.44
```



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

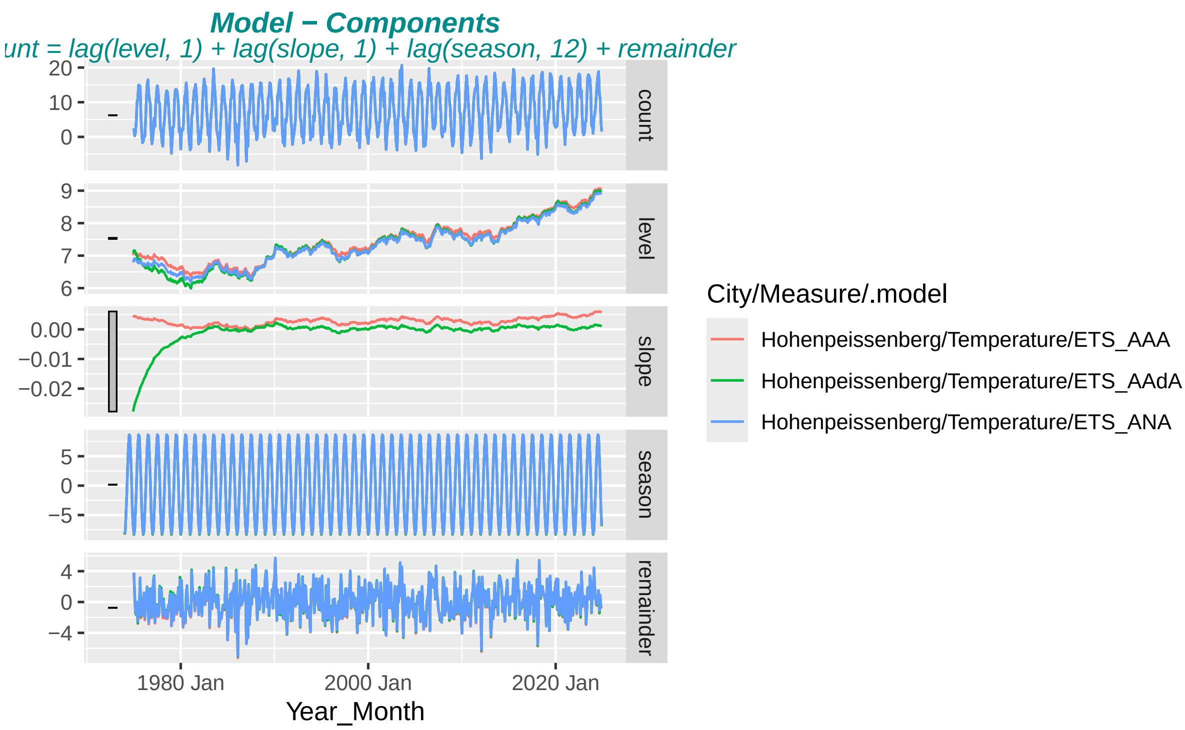
- MAE (Mean absolute error, will lead to forecasts of the median) and
- RMSE (Root mean squared error, lead to forecasts of the mean)

```
#> # 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 Hohenpeisse~ Temper~ ETS_A~ Trai~  0.0313    2.00  1.59 -53.8  136. 0.713 0.701
#> 2 Hohenpeisse~ Temper~ ETS_A~ Trai~  0.0292    2.00  1.59 -54.1  136. 0.714 0.701
#> 3 Hohenpeisse~ Temper~ ETS_A~ Trai~  0.145     2.00  1.59 -51.5  135. 0.714 0.701
#> 4 Hohenpeisse~ Temper~ ETS_A~ Trai~  0.141     2.00  1.60 -48.7  134. 0.716 0.702
#> 5 Hohenpeisse~ Temper~ ETS_M~ Trai~  0.0533    2.08  1.65 -40.7  141. 0.739 0.729
#> 6 Hohenpeisse~ Temper~ ETS_M~ Trai~  0.0161    2.11  1.67 -36.4  151. 0.747 0.738
#> 7 Hohenpeisse~ Temper~ ETS_M~ Trai~  0.00467   2.64  2.08 -52.6  181. 0.933 0.923
#> 8 Hohenpeisse~ Temper~ ETS_M~ Trai~ -0.410     2.86  2.13 -77.3  193. 0.954 1.00
#> # i 1 more variable: ACF1 <dbl>
```

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

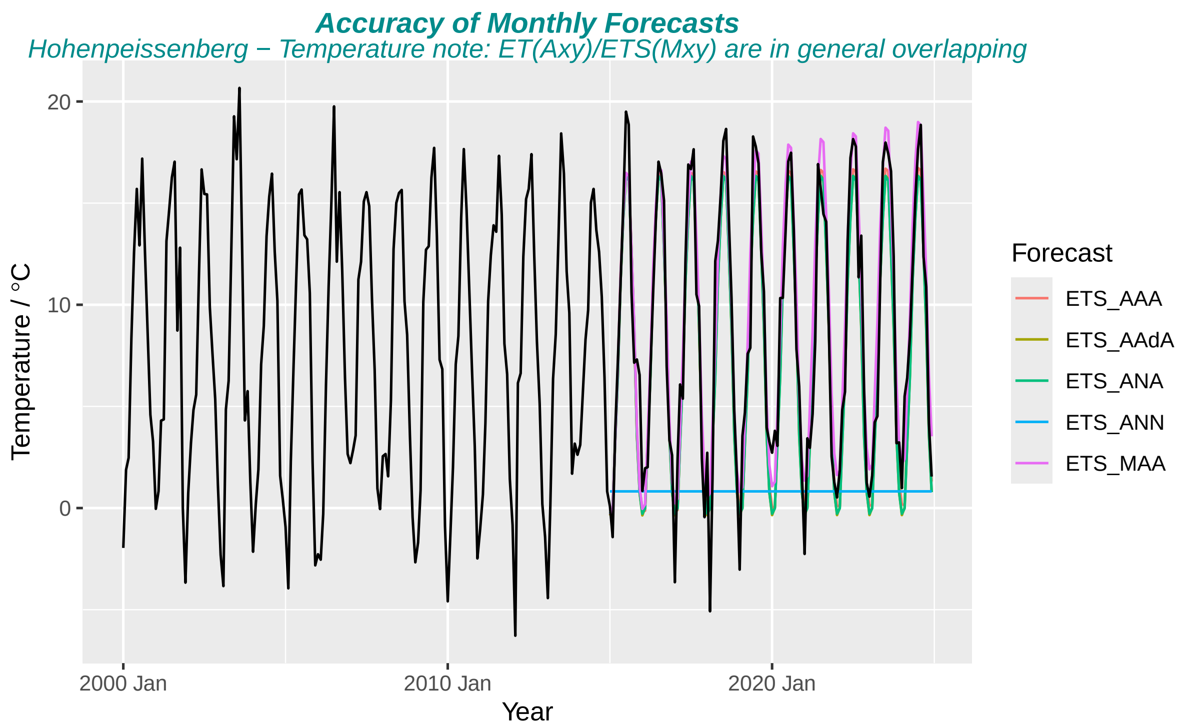
```
#> Null Hypothesis of independence/white noise for residuals - for p < 0.05: reject H_0
#> # A tibble: 8 x 5
#>   City             Measure     .model    lb_stat lb_pvalue
#>   <chr>            <fct>       <chr>       <dbl>     <dbl>
#> 1 Hohenpeissenberg Temperature ETS_AAA      40.0   0.105
#> 2 Hohenpeissenberg Temperature ETS_AMA      40.2   0.101
#> 3 Hohenpeissenberg Temperature ETS_ANA      40.6   0.0938
#> 4 Hohenpeissenberg Temperature ETS_AAdA     41.7   0.0752
#> 5 Hohenpeissenberg Temperature ETS_MMA      51.9   0.00779
#> 6 Hohenpeissenberg Temperature ETS_MAA      53.1   0.00571
#> 7 Hohenpeissenberg Temperature ETS_MAdA   2406.    0
#> 8 Hohenpeissenberg Temperature ETS_MNA     168.    0
```

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

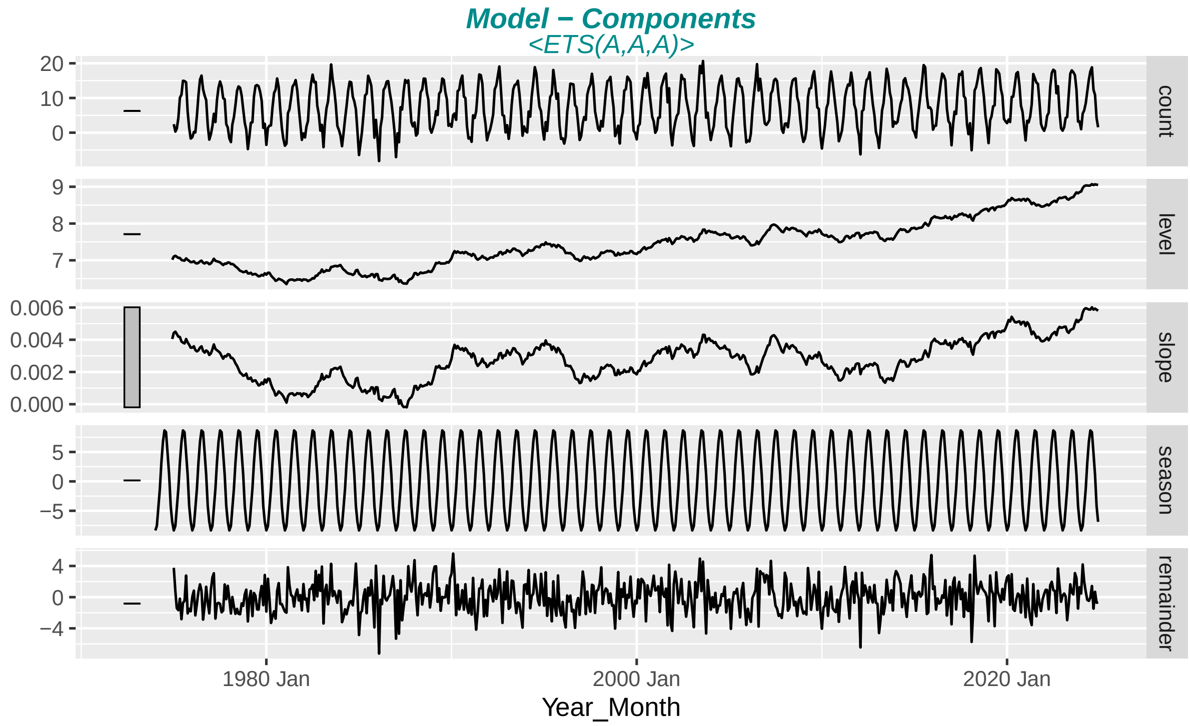
```
#> # A tibble: 5 x 12
#>   .model City  Measure .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE    ACF1
#>   <chr>  <chr> <fct>   <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
#> 1 ETS_A~ Hohe~ Temper~ Test   0.644  2.05  1.65 23.0    36.7 0.748 0.727 -0.0272
#> 2 ETS_M~ Hohe~ Temper~ Test  -0.509  2.07  1.62 -0.591  45.3 0.735 0.732  0.0355
#> 3 ETS_A~ Hohe~ Temper~ Test   0.939  2.15  1.76 31.4    42.3 0.799 0.763 -0.0207
#> 4 ETS_A~ Hohe~ Temper~ Test   0.934  2.15  1.76 30.2    41.3 0.800 0.763 -0.0232
#> 5 ETS_A~ Hohe~ Temper~ Test   7.88  10.1   8.25 77.4    92.7 3.75  3.59   0.770
```

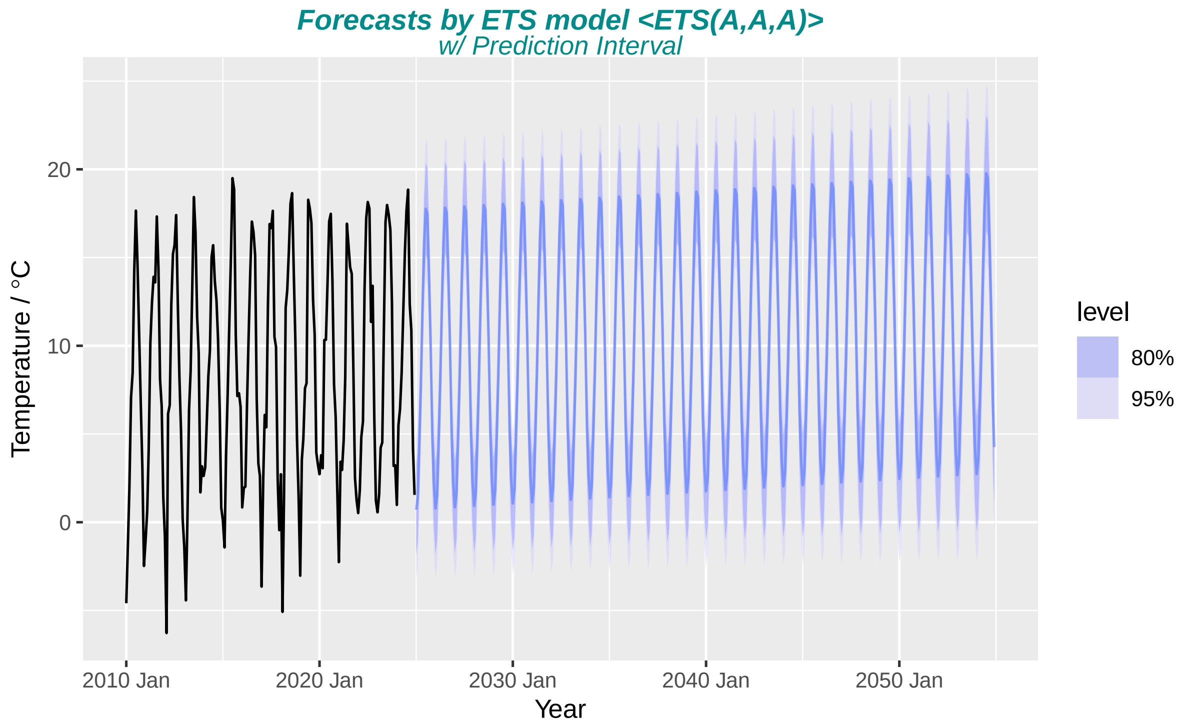


2.2 Forecasting with selected ETS model <ETS(A,A,A)>

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 = 0.02163066
#>     beta  = 0.0001001477
#>     gamma = 0.0001005095
#>
#>   Initial states:
#>      l[0]        b[0]       s[0]     s[-1]    s[-2]    s[-3]    s[-4]    s[-5]
#>  7.016633 0.004047721 -6.875202 -4.193987 1.135854 4.761125 8.377214 8.670147
#>     s[-6]   s[-7]     s[-8]     s[-9]    s[-10]    s[-11]
#>  6.720997 3.16646 -1.251049 -4.592603 -7.574575 -8.344381
#>
#>   sigma^2:  4.1178
#>
#>      AIC     AICc      BIC
#> 4705.139 4706.191 4779.887
```

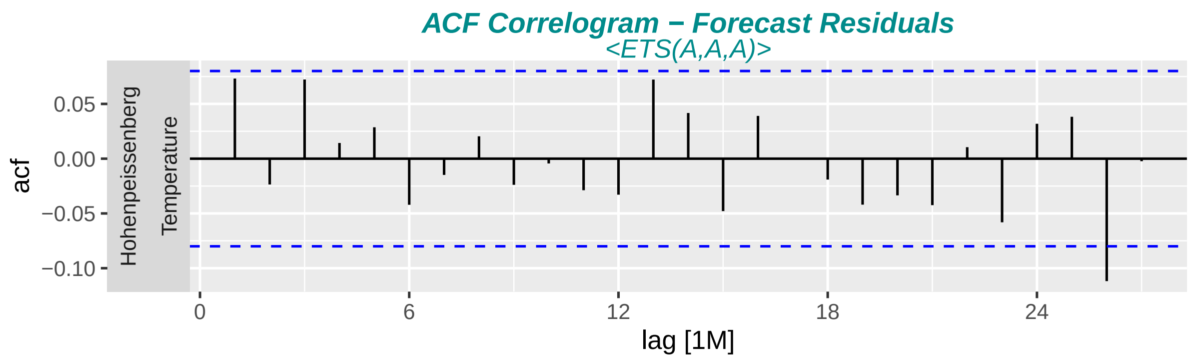


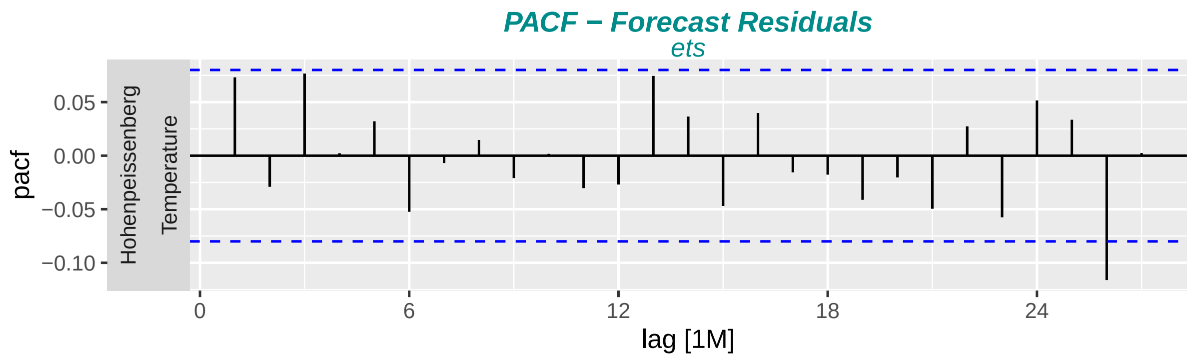


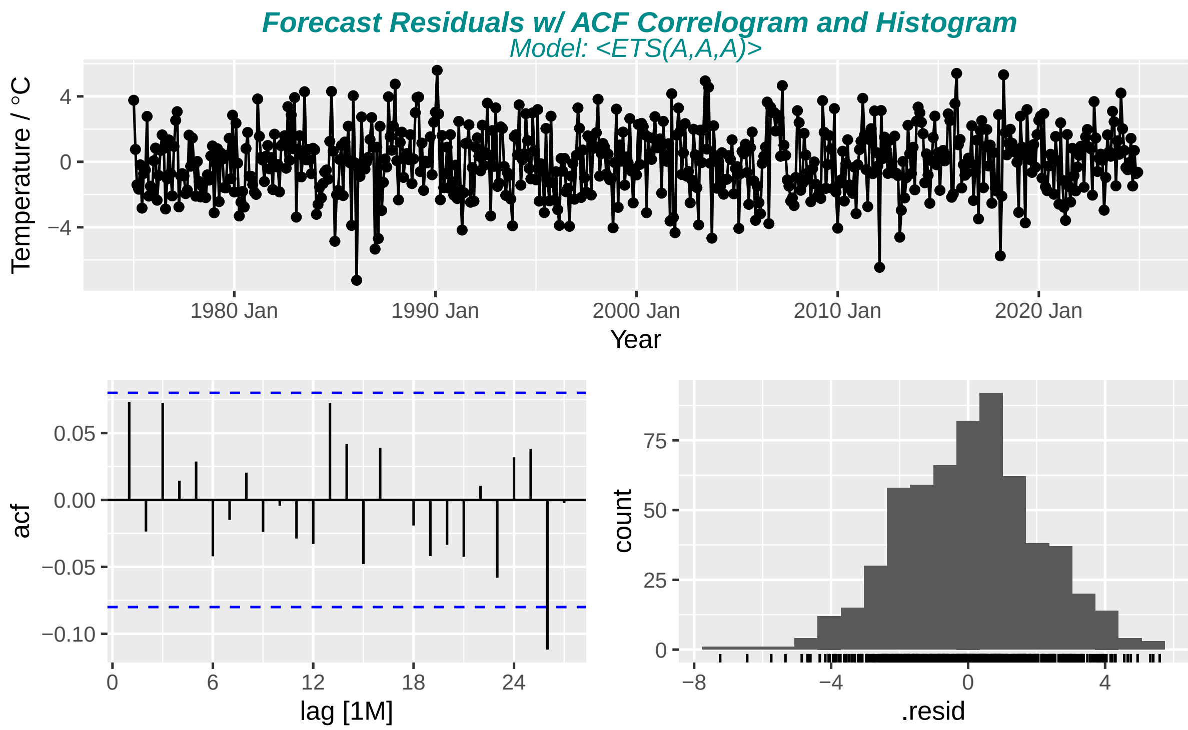
2.2.2 Residual Stationarity

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

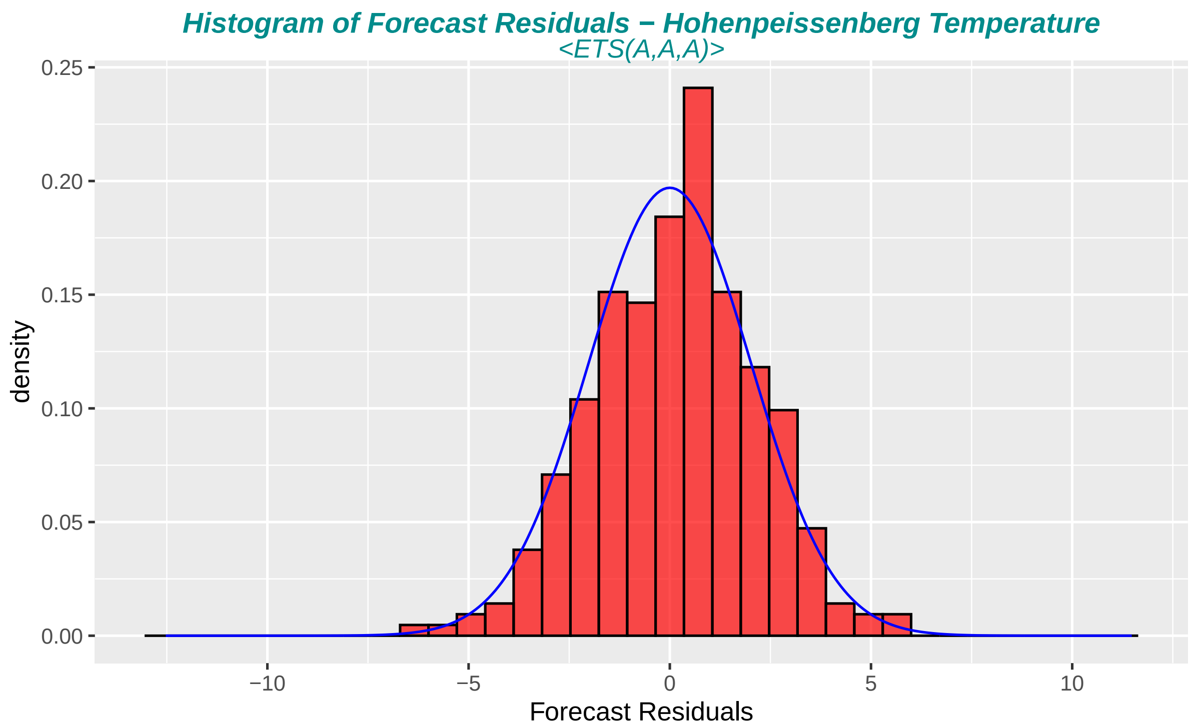






2.2.3 Histogram of forecast residuals with overlaid normal curve

```
#> 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 Hohenpeissenberg Temperature ets       37.6     0.161
```



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

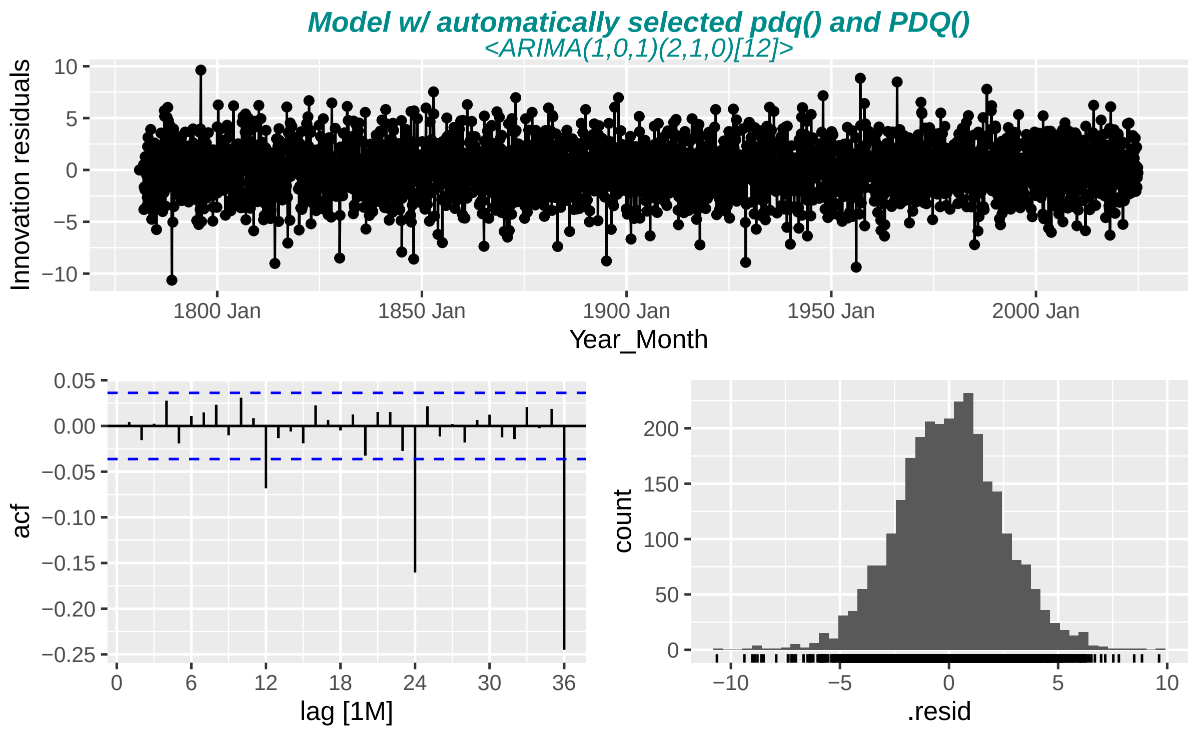
Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, * p is the order (number of time lags) of the autoregressive model * d is the degree of differencing (number of times the data have had past values subtracted) * q is the order of the 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_roots
#>   <chr>      <fct>   <chr>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>   <list>
#> 1 Hohenpei~ Temper~ arima    5.82  -6707. 13424. 13424. 13454. <cpl>    <cpl>
#> Series: count
#> Model: ARIMA(1,0,1)(2,1,0)[12]
```

```
#>
#> Coefficients:
#>          ar1      ma1     sar1     sar2
#>       0.4888  -0.3927  -0.6891  -0.3074
#> s.e.  0.1801   0.1905   0.0176   0.0177
#>
#> sigma^2 estimated as 5.821:  log likelihood=-6706.99
#> AIC=13423.97   AICc=13423.99   BIC=13453.86
```




```
#> Model Selection by Information Criterion - lowest AIC, AICc, BIC
#>  choose p, q parameter accordingly - but only for same d, D values
#> # A tibble: 12 x 10
#>    City        Measure .model sigma2 log_lik   AIC  AICc   BIC ar_roots ma_roots
#>    <chr>       <fct>   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>   <list>
#>  1 Hohenpeiss~ Temper~ ARIMA~   4.10  -1269. 2547. 2547. 2569. <cpl>    <cpl>
#>  2 Hohenpeiss~ Temper~ ARIMA~   4.11  -1269. 2547. 2547. 2569. <cpl>    <cpl>
#>  3 Hohenpeiss~ Temper~ ARIMA~   4.11  -1269. 2547. 2547. 2569. <cpl>    <cpl>
#>  4 Hohenpeiss~ Temper~ ARIMA~   5.31  -1327. 2662. 2662. 2679. <cpl>    <cpl>
#>  5 Hohenpeiss~ Temper~ ARIMA~   5.87  -1355. 2719. 2719. 2741. <cpl>    <cpl>
#>  6 Hohenpeiss~ Temper~ ARIMA~   5.87  -1355. 2719. 2719. 2741. <cpl>    <cpl>
#>  7 Hohenpeiss~ Temper~ ARIMA~   5.64  -1378. 2773. 2773. 2808. <cpl>    <cpl>
#>  8 Hohenpeiss~ Temper~ ARIMA~   7.26  -1416. 2840. 2840. 2857. <cpl>    <cpl>
#>  9 Hohenpeiss~ Temper~ ARIMA~   8.18  -1452. 2910. 2910. 2923. <cpl>    <cpl>
#> 10 Hohenpeiss~ Temper~ ARIMA~   8.18  -1452. 2910. 2910. 2923. <cpl>    <cpl>
#> 11 Hohenpeiss~ Temper~ ARIMA~  10.9   -1534. 3073. 3073. 3082. <cpl>    <cpl>
#> 12 Hohenpeiss~ Temper~ ARIMA~  12.4   -1571. 3146. 3146. 3155. <cpl>    <cpl>
```

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)

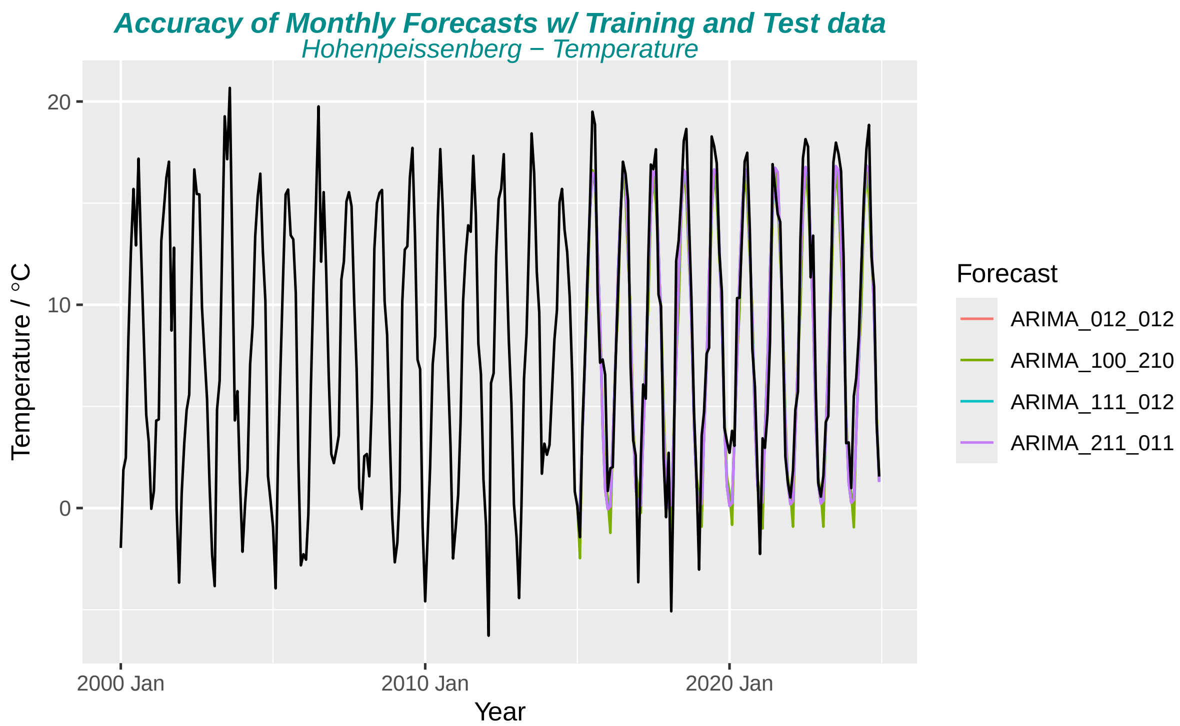
```
#> # A tibble: 14 x 12
#>    City        Measure .model .type        ME   RMSE    MAE    MPE  MAPE    MASE
#>    <chr>       <fct>   <chr>  <chr>     <dbl>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>
#>  1 Hohenpeiss~ Temper~ ARIMA~ Trai~   1.04e-1   2.00   1.56  -34.7  131.   0.701
#>  2 Hohenpeiss~ Temper~ ARIMA~ Trai~   1.04e-1   2.00   1.56  -31.6  128.   0.700
#>  3 Hohenpeiss~ Temper~ ARIMA~ Trai~   1.04e-1   2.00   1.56  -32.9  128.   0.700
#>  4 Hohenpeiss~ Temper~ ARIMA~ Trai~   1.01e-1   2.28   1.80   16.8  139.   0.807
#>  5 Hohenpeiss~ Temper~ ARIMA~ Trai~   4.55e-2   2.36   1.88  -40.6  162.   0.842
#>  6 Hohenpeiss~ Temper~ ARIMA~ Trai~  -2.39e-4   2.39   1.87  -20.0  154.   0.839
#>  7 Hohenpeiss~ Temper~ ARIMA~ Trai~  -2.39e-4   2.39   1.87  -20.0  154.   0.839
#>  8 Hohenpeiss~ Temper~ ARIMA~ Trai~   9.17e-3   2.66   2.10  -41.9  180.   0.943
#>  9 Hohenpeiss~ Temper~ ARIMA~ Trai~   5.33e-2   2.82   2.19  -39.6  237.   0.982
#> 10 Hohenpeiss~ Temper~ ARIMA~ Trai~   5.34e-2   2.82   2.19  -38.7  237.   0.981
#> 11 Hohenpeiss~ Temper~ ARIMA~ Trai~   7.46e-3   3.26   2.52 -142.   283.   1.13
#> 12 Hohenpeiss~ Temper~ ARIMA~ Trai~   6.36e-3   3.48   2.70  -62.7  252.   1.21
#> 13 Hohenpeiss~ Temper~ ARIMA~ Trai~ NaN       NaN    NaN     NaN    NaN  NaN
#> 14 Hohenpeiss~ Temper~ ARIMA~ Trai~ NaN       NaN    NaN     NaN    NaN  NaN
#> # i 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 Hohenpeissenberg Temperature ARIMA_211_011    38.9  1.29e- 1
#>  2 Hohenpeissenberg Temperature ARIMA_012_012    39.5  1.15e- 1
#>  3 Hohenpeissenberg Temperature ARIMA_111_012    39.9  1.08e- 1
#>  4 Hohenpeissenberg Temperature ARIMA_301_200    84.8  3.88e- 7
#>  5 Hohenpeissenberg Temperature ARIMA_100_210   101.   1.31e- 9
#>  6 Hohenpeissenberg Temperature ARIMA_100_110   109.   5.66e-11
#>  7 Hohenpeissenberg Temperature ARIMA_200_110   109.   5.66e-11
#>  8 Hohenpeissenberg Temperature ARIMA_010_110   296.   0
#>  9 Hohenpeissenberg Temperature ARIMA_012_010   226.   0
#> 10 Hohenpeissenberg Temperature ARIMA_110_010   436.   0
#> 11 Hohenpeissenberg Temperature ARIMA_111_010   226.   0
#> 12 Hohenpeissenberg Temperature ARIMA_210_110   159.   0
#> 13 Hohenpeissenberg Temperature ARIMA_002_200    NA   NA
#> 14 Hohenpeissenberg Temperature ARIMA_111_112    NA   NA
```

3.1.3 Forecast Accuracy with Training/Test Data

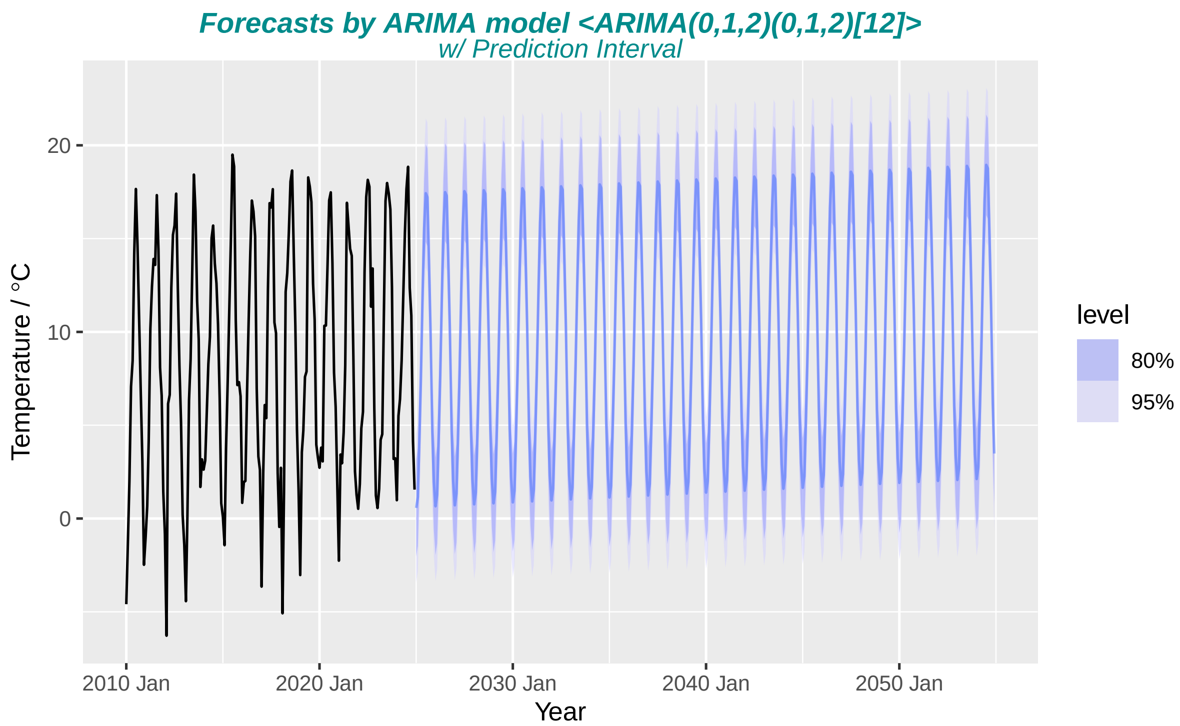
```
#> # A tibble: 4 x 12
#>   .model   City  Measure .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE    ACF1
#>   <chr>    <chr> <fct>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
#> 1 ARIMA_1~ Hohe~ Temper~ Test  0.554  2.02  1.61  20.4  35.4 0.734 0.714 -0.0330
#> 2 ARIMA_0~ Hohe~ Temper~ Test  0.555  2.02  1.62  20.7  35.7 0.734 0.714 -0.0330
#> 3 ARIMA_2~ Hohe~ Temper~ Test  0.558  2.02  1.62  21.2  36.3 0.736 0.714 -0.0320
#> 4 ARIMA_1~ Hohe~ Temper~ Test  0.792  2.22  1.78  17.6  36.8 0.811 0.786 -0.0391
```



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

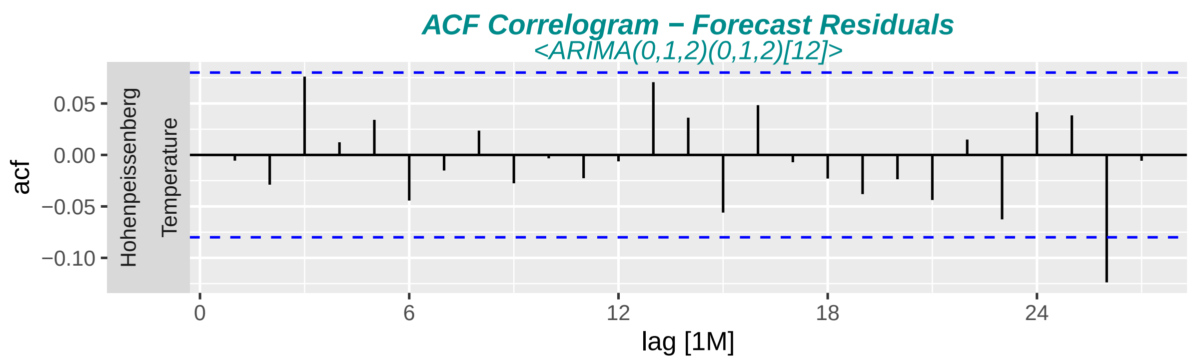
```
#> Provide model coefficients by report(fit_model)
#> Series: count
#> Model: ARIMA(0,1,2)(0,1,2)[12]
#>
#> Coefficients:
#>           ma1      ma2     sma1    sma2
#>       -0.9108  -0.0835  -0.9951  0.0180
#> s.e.   0.0444   0.0436   0.0500  0.0413
#>
#> sigma^2 estimated as 4.113:  log likelihood=-1268.56
#> AIC=2547.12   AICc=2547.22   BIC=2568.99
```

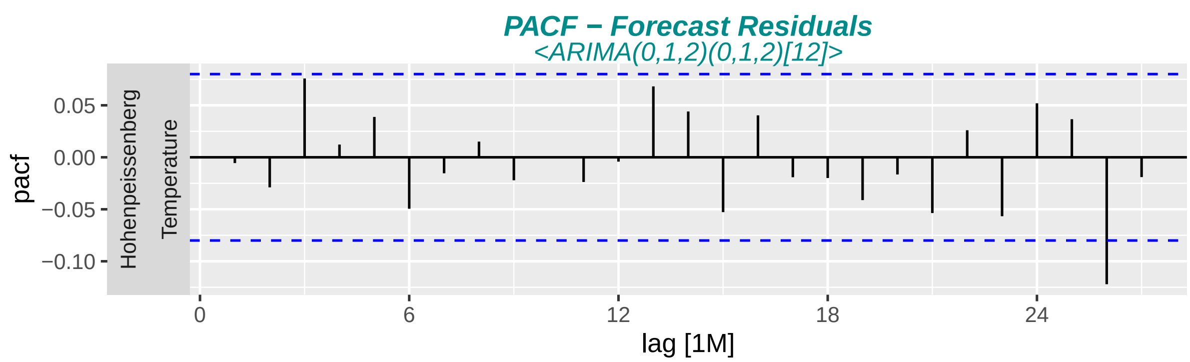


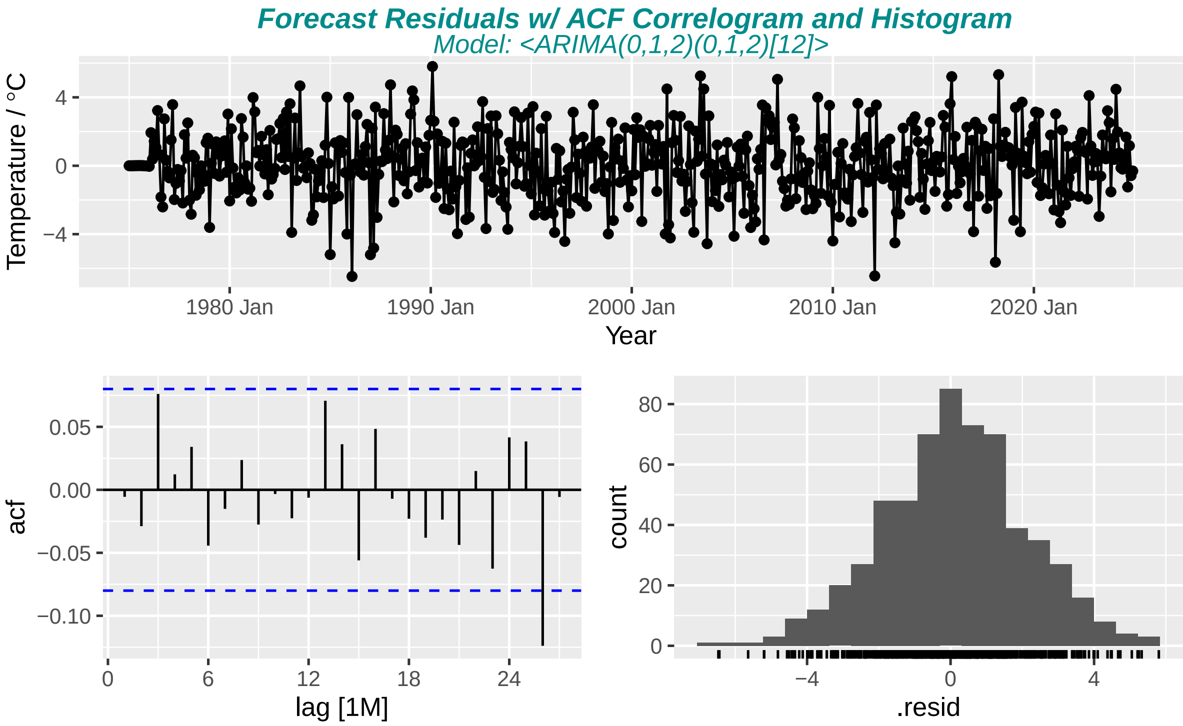
3.2.2 Residual Stationarity

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

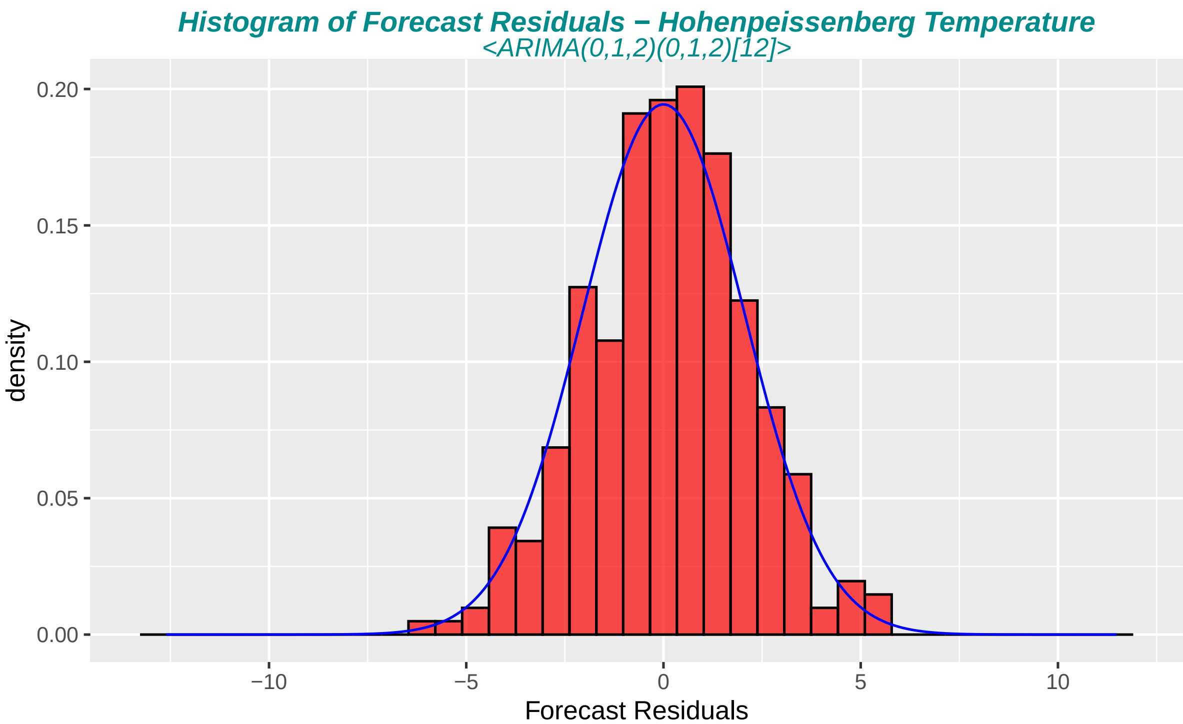






3.2.3 Histogram of forecast residuals with overlaid normal curve

```
#> 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 Hohenpeissenberg Temperature arima     31.3     0.401
```



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.

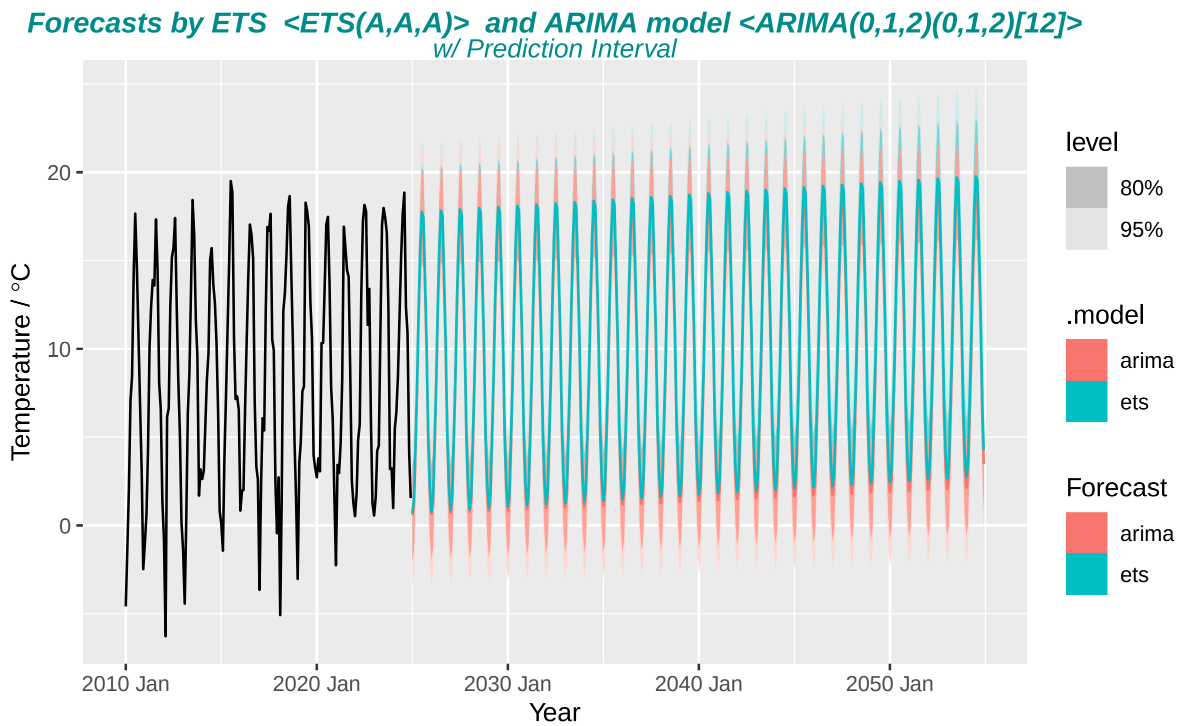
- Residual Accuracy with one-step-ahead fitted residuals
- Forecast Accuracy with Training/Test Data

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

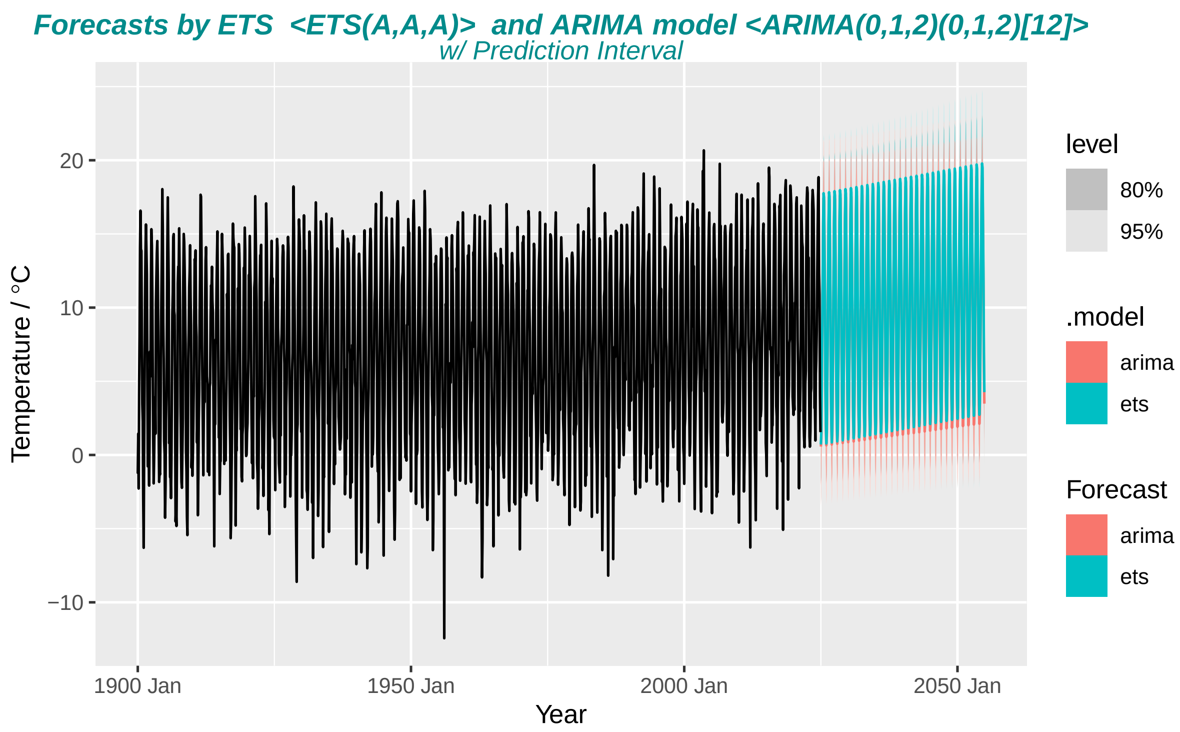
```
#> # A tibble: 4 x 12
#>   City  Measure .model .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
#>   <chr> <fct>   <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
#> 1 Hohe~ Temper~ ets    Trai~ 0.0292  2.00  1.59 -54.1 136.  0.714 0.701  0.0731
#> 2 Hohe~ Temper~ arima  Trai~ 0.104   2.00  1.56 -32.9 128.  0.700 0.700 -0.00555
#> 3 Hohe~ Temper~ ETS_A~ Test  0.644   2.05  1.65  23.0  36.7 0.748 0.727 -0.0272
#> 4 Hohe~ Temper~ ARIMA~ Test  0.555   2.02  1.62  20.7  35.7 0.734 0.714 -0.0330
```

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_Month
#>   <chr>             <fct>       <chr>       <mth>
#> 1 Hohenpeissenberg Temperature arima    2025 Jan
#> 2 Hohenpeissenberg Temperature arima    2025 Feb
#> 3 Hohenpeissenberg Temperature arima    2025 Mrz
#> 4 Hohenpeissenberg Temperature ets      2025 Jan
#> 5 Hohenpeissenberg Temperature ets      2025 Feb
#> 6 Hohenpeissenberg Temperature ets      2025 Mrz
#> # i 4 more variables: count <dist>, .mean <dbl>, ‘80%‘ <hilo>, ‘95%‘ <hilo>
#> # A tsibble: 6 x 8 [1M]
#> # Key:       City, Measure, .model [2]
#> # Groups:    City, Measure, .model [2]
#>   City              Measure     .model Year_Month
#>   <chr>             <fct>       <chr>       <mth>
#> 1 Hohenpeissenberg Temperature arima    2054 Okt
#> 2 Hohenpeissenberg Temperature arima    2054 Nov
#> 3 Hohenpeissenberg Temperature arima    2054 Dez
#> 4 Hohenpeissenberg Temperature ets      2054 Okt
#> 5 Hohenpeissenberg Temperature ets      2054 Nov
#> 6 Hohenpeissenberg Temperature ets      2054 Dez
#> # i 4 more variables: count <dist>, .mean <dbl>, ‘80%‘ <hilo>, ‘95%‘ <hilo>
```




```
#> # A tibble: 180 x 5
#> # Groups:   City, Measure, .model, Year [60]
#>    City              Measure     .model  Year Year_avg
#>    <chr>             <fct>       <chr>  <dbl>    <dbl>
#>  1 Hohenpeissenberg Temperature arima   2025    0.570
#>  2 Hohenpeissenberg Temperature arima   2025    1.15
#>  3 Hohenpeissenberg Temperature arima   2025    4.15
#>  4 Hohenpeissenberg Temperature arima   2026    0.655
#>  5 Hohenpeissenberg Temperature arima   2026    1.28
#>  6 Hohenpeissenberg Temperature arima   2026    4.24
#>  7 Hohenpeissenberg Temperature arima   2027    0.707
```

```
#>  8 Hohenpeissenberg Temperature arima   2027    1.33
#>  9 Hohenpeissenberg Temperature arima   2027    4.30
#> 10 Hohenpeissenberg Temperature arima   2028    0.759
#> # i 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 Hohenpeissenberg Temperature arima   2025     9.92
#>  2 Hohenpeissenberg Temperature arima   2025     4.66
#>  3 Hohenpeissenberg Temperature arima   2025     1.98
#>  4 Hohenpeissenberg Temperature arima   2026     9.99
#>  5 Hohenpeissenberg Temperature arima   2026     4.70
#>  6 Hohenpeissenberg Temperature arima   2026     2.02
#>  7 Hohenpeissenberg Temperature arima   2027    10.0
#>  8 Hohenpeissenberg Temperature arima   2027     4.75
#>  9 Hohenpeissenberg Temperature arima   2027     2.07
#> 10 Hohenpeissenberg Temperature arima   2028    10.1
#> # i 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 Hohenpeissenberg Temperature arima     39.5     0.115
#> 2 Hohenpeissenberg Temperature ets       40.0     0.105
```

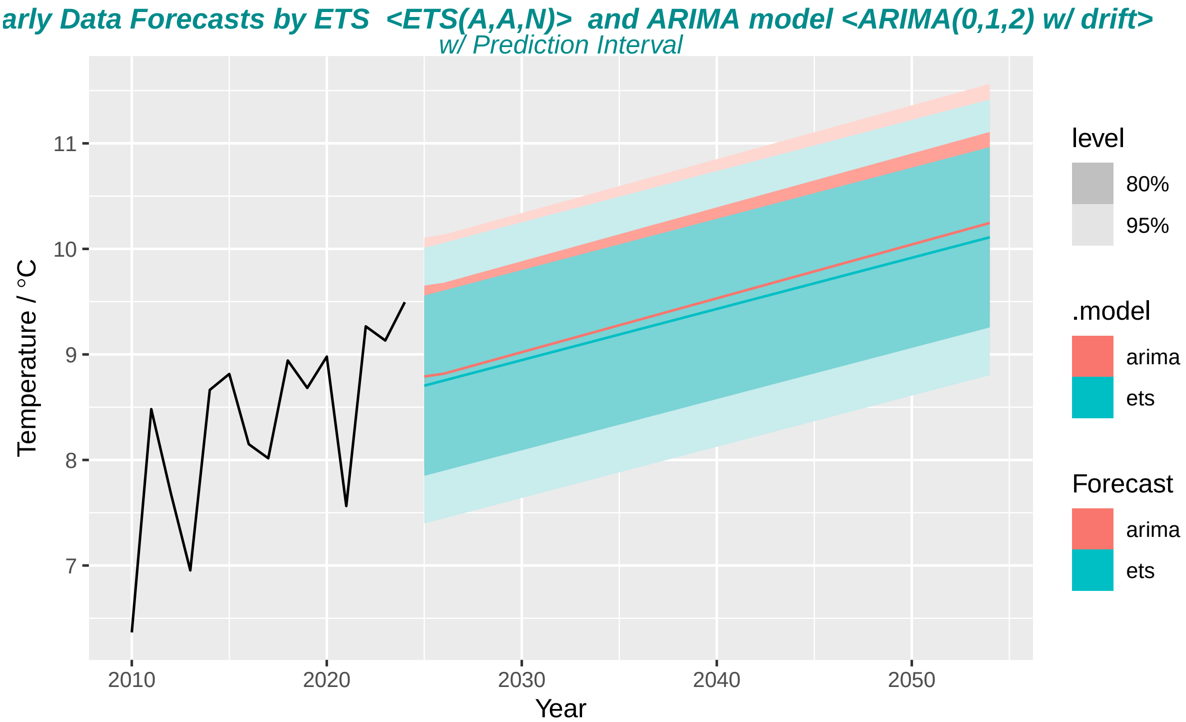
5 Yearly Data Forecasts with ARIMA and ETS

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.

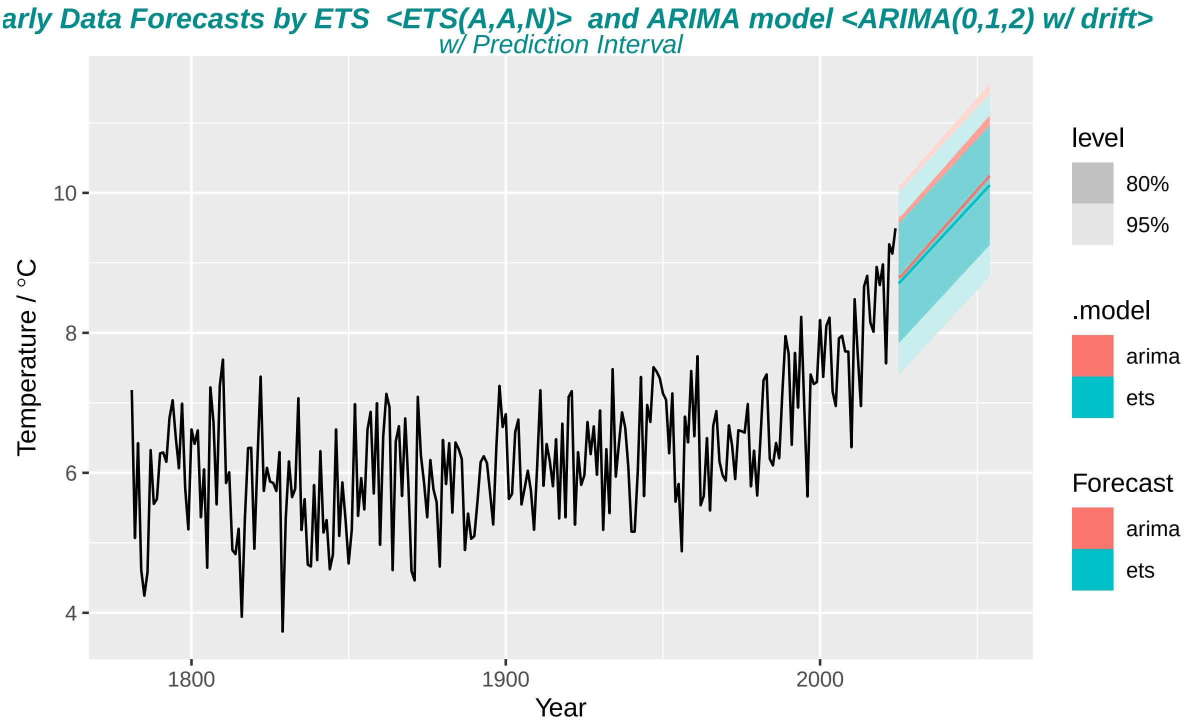
The ETS model $< ETS(A, A, N) >$ with seasonal term change "A" -> "N" is chosen. For ARIMA models the seasonal term (P,D,Q)m has to be taken out and an optimal ARIMA(p,1,q) with one differencing (d=1) is selected. However, for Mauna Loa two times differncing had to be selected $CO_2 <ARIMA(0,2,1) w/ 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
#>   <chr>            <fct>       <chr>  <dbl>
#> 1 Hohenpeissenberg Temperature arima   2025
#> 2 Hohenpeissenberg Temperature arima   2026
#> 3 Hohenpeissenberg Temperature arima   2027
#> 4 Hohenpeissenberg Temperature ets     2025
#> 5 Hohenpeissenberg Temperature ets     2026
#> 6 Hohenpeissenberg Temperature ets     2027
#> # i 4 more variables: Year_avg <dist>, .mean <dbl>, `80%` <hilo>, `95%` <hilo>
#> # A tsibble: 6 x 8 [1Y]
#> # Key:       City, Measure, .model [2]
#> # Groups:    City, Measure, .model [2]
#>   City             Measure     .model  Year
#>   <chr>            <fct>       <chr>  <dbl>
#> 1 Hohenpeissenberg Temperature arima   2052
#> 2 Hohenpeissenberg Temperature arima   2053
#> 3 Hohenpeissenberg Temperature arima   2054
#> 4 Hohenpeissenberg Temperature ets     2052
#> 5 Hohenpeissenberg Temperature ets     2053
#> 6 Hohenpeissenberg Temperature ets     2054
#> # i 4 more variables: Year_avg <dist>, .mean <dbl>, `80%` <hilo>, `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 Hohenpeissenberg Temperature arima     30.3     0.451
#> 2 Hohenpeissenberg Temperature ets       29.5     0.489
```

6 Backup