

Climate Data Forecasting - Atmospheric CO_2 Concentration / Temperature / Precipitation

Wolfgang Vollmer

2021-04-05

Contents

1	Forecasting of Hohenpeissenberg - Temperature Climate Analysis	2
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
2	ExponenTial Smoothing (ETS) Forecasting Models	6
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
3	ARIMA Forecasting Models - AutoRegressive-Integrated Moving Average	13
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

4	ARIMA vs ETS	19
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
5	Yearly Data Forecasts with ARIMA and ETS	21
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
6	Backup	23

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

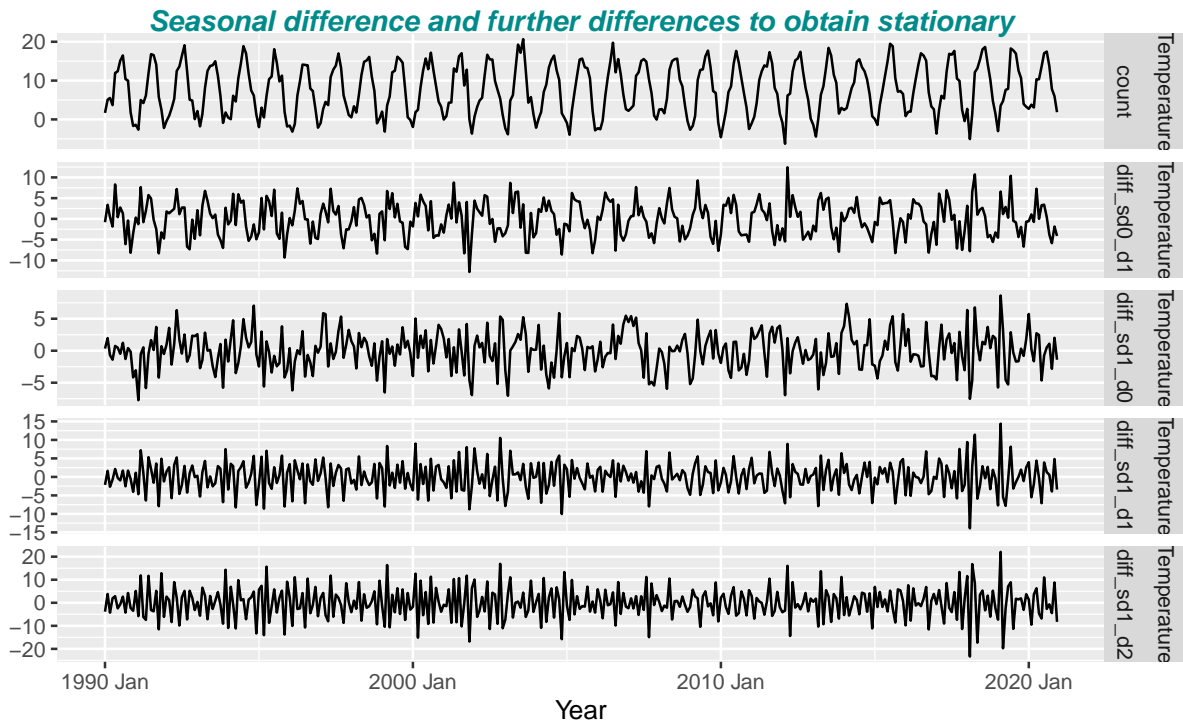
The model fit residuals have to be stationary. For good forecasting this has to be verified with residual diagnostics.

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 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 8981.      0
#> Ljung-Box test on (difference(count, 12))
#> # A tibble: 1 x 3
#>   Measure    lb_stat lb_pvalue
#>   <fct>      <dbl>    <dbl>
#> 1 Temperature 46.4 0.00000121
#> Ljung-Box test on (difference(count, 12) + difference())
#> # A tibble: 1 x 3
#>   Measure    lb_stat lb_pvalue
#>   <fct>      <dbl>    <dbl>
#> 1 Temperature 560.      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

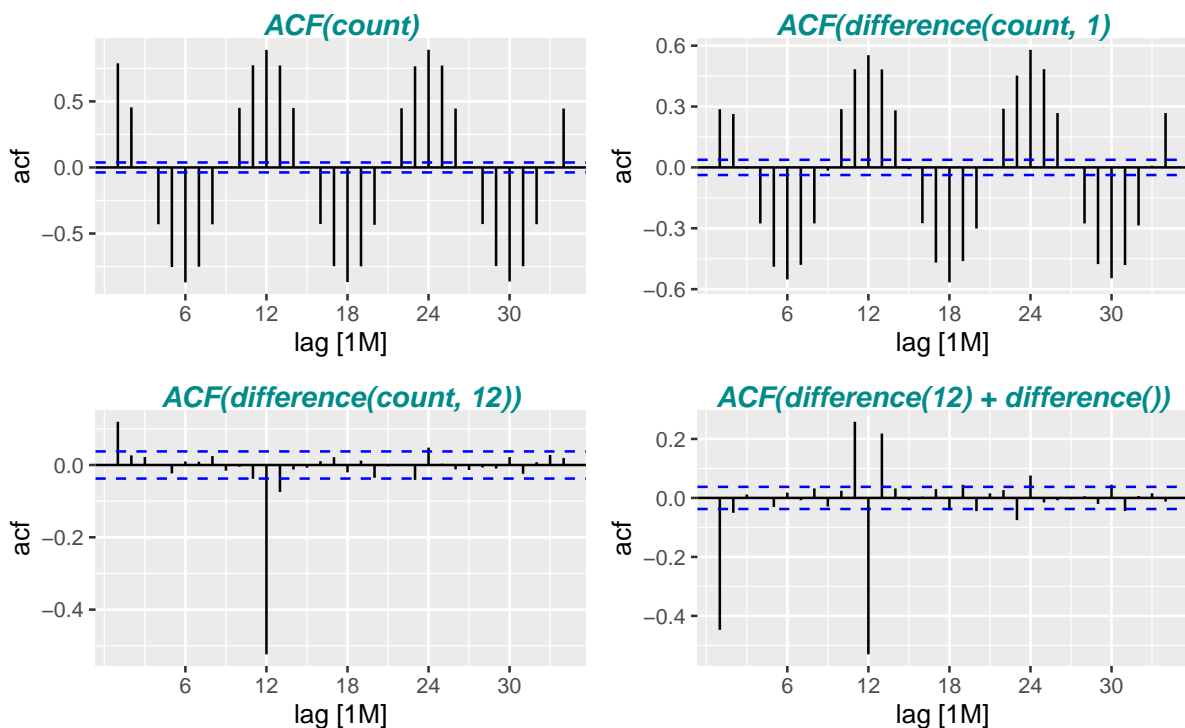
=> H_0 to be rejected for $p < \alpha = 0.05$

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.

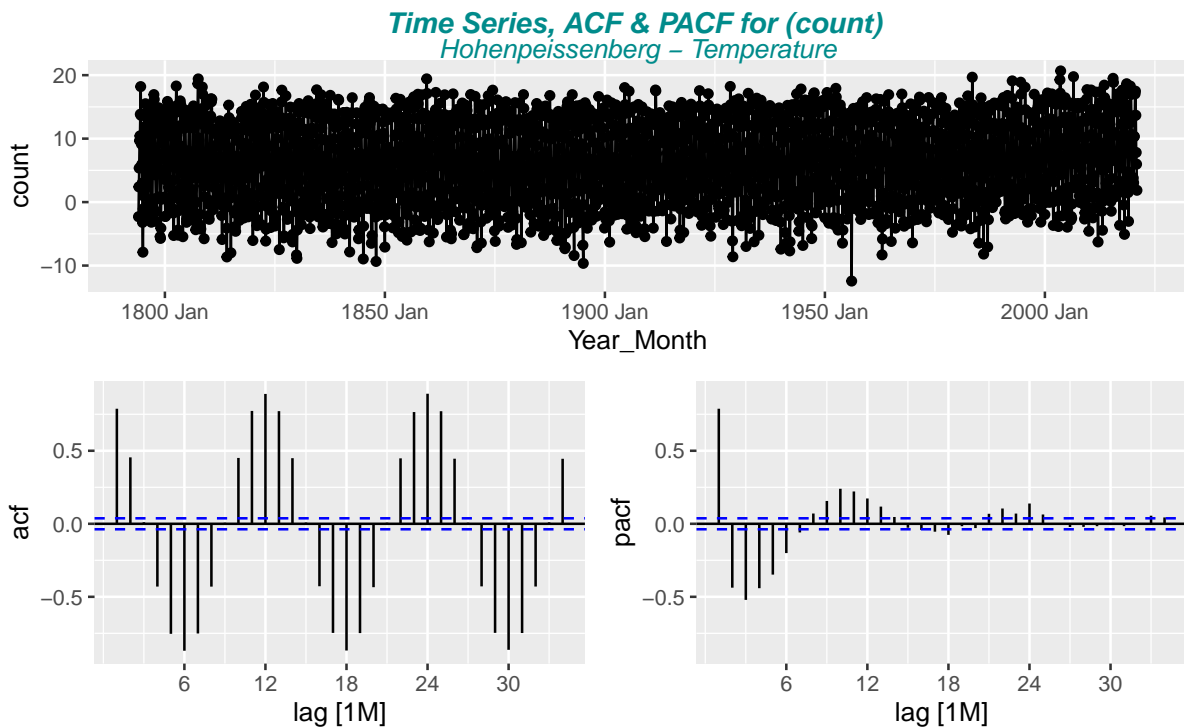
- returns 1 => for stationarity one seasonal difference resp. difference is required

```
#> 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      2.98          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    0.0177          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    0.00455          0.1       0     0
```

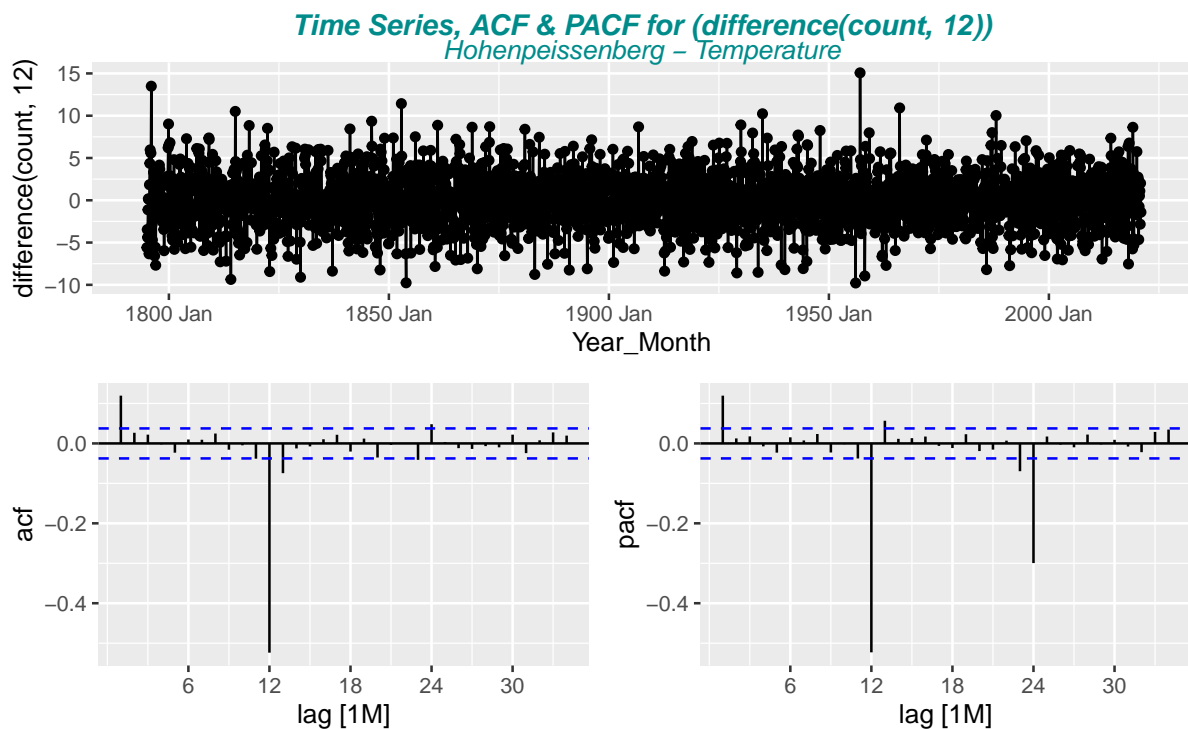
1.1.3 ACF Plots of Differences



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

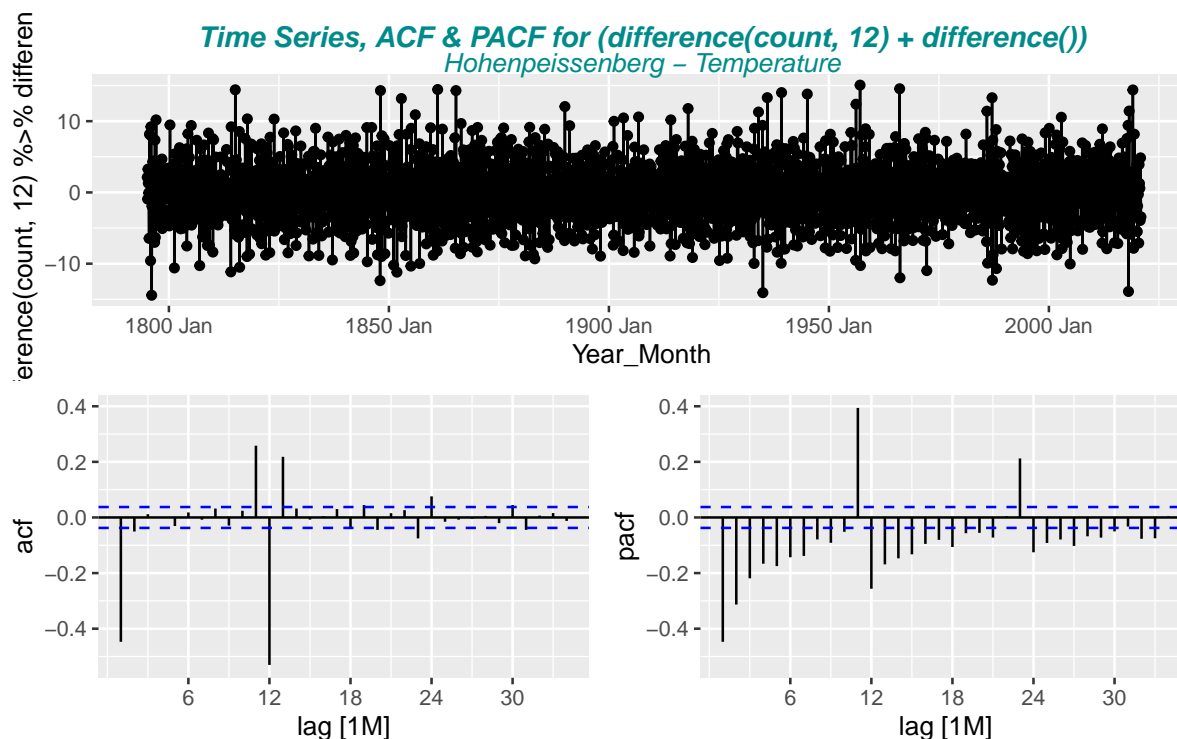


```
#> # A tibble: 1 x 2
#>   Sum      Mean
#>   <dbl>   <dbl>
#> 1  21.6 0.00796
```



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

```
#> <dbl> <dbl>
#> 1 21.6 0.00796
```



```
#> # A tibble: 1 x 2
#>   Sum      Mean
#>   <dbl>   <dbl>
#> 1  4.18 0.00154
```

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” exponential smoothing without trend (see above).

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

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

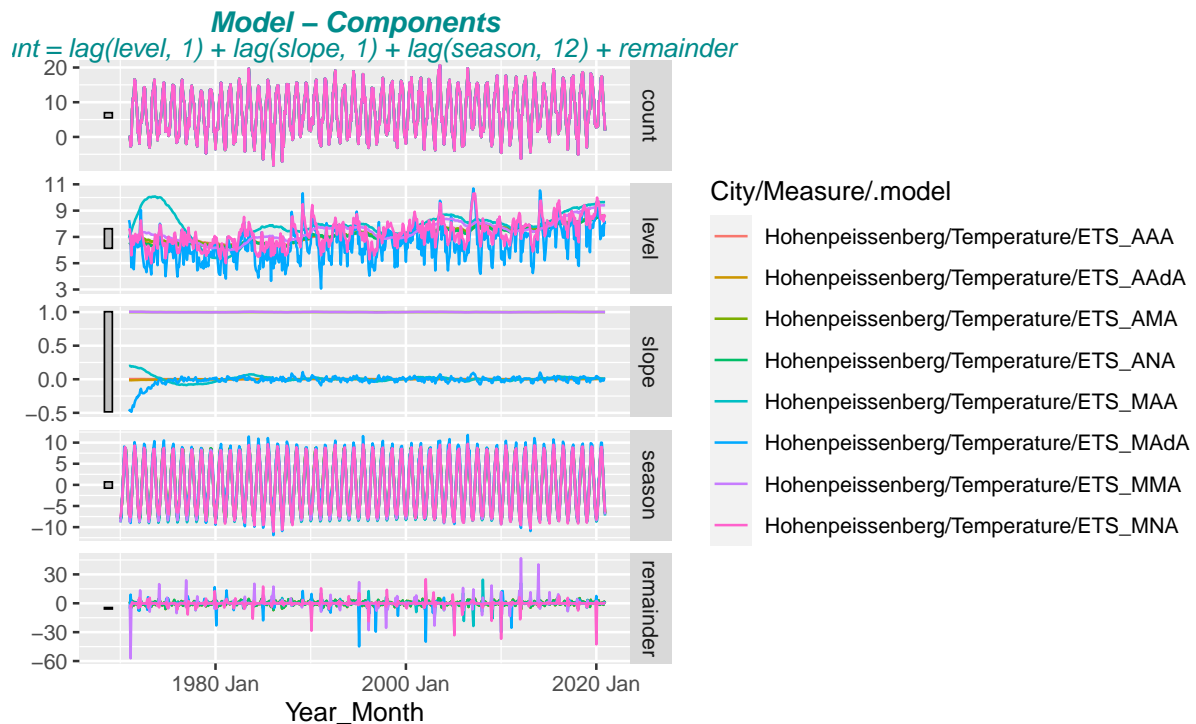
For CO2 and Temperature Data we have a clear seasonal pattern and seasonal term adds always a (more or less) fix amount on level and trend component. Therefore “A” additive term is chosen. For Precipitation the seasonal pattern is only slight. Instead, a multiplicative seasonal term results in “exploding”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

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

2.1 ETS Models and their components

```
#> [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 Hohenpeis~ Tempera~ ETS(co~  4.20  -2342. 4717. 4718. 4792.  4.09  4.10  1.61
#> Series: count
#> Model: ETS(A,A,A)
#> Smoothing parameters:
#>   alpha = 0.014561
#>   beta  = 0.0001004411
#>   gamma = 0.000101192
#>
#> Initial states:
#>      1      b      s1      s2      s3      s4      s5      s6
#> 6.766848 0.003163054 -6.733642 -4.165398 0.7660276 4.747525 8.510702 8.572779
#>      s7      s8      s9      s10     s11     s12
#> 6.403173 3.235912 -1.124447 -4.528056 -7.691808 -7.992768
#>
#> sigma^2: 4.2016
#>
#>      AIC      AICc      BIC
#> 4717.224 4718.276 4791.972
#> 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 Hohenpeis~ Temperat~ ETS_A~  4.20 -2342. 4717. 4718. 4792.  4.09  4.10  1.61
#> 2 Hohenpeis~ Temperat~ ETS_A~  4.21 -2342. 4718. 4719. 4793.  4.10  4.11  1.61
#> 3 Hohenpeis~ Temperat~ ETS_A~  4.23 -2344. 4719. 4720. 4785.  4.13  4.14  1.62
#> 4 Hohenpeis~ Temperat~ ETS_A~  4.22 -2343. 4721. 4722. 4800.  4.10  4.11  1.61
#> 5 Hohenpeis~ Temperat~ ETS_M~ 10.4  -3522. 7078. 7079. 7152.  5.39  5.44  1.24
#> 6 Hohenpeis~ Temperat~ ETS_M~ 17.2  -3647. 7324. 7324. 7390.  5.56  5.64  1.42
#> 7 Hohenpeis~ Temperat~ ETS_M~ 18.0  -3695. 7427. 7428. 7506.  7.40  7.69  1.39
#> 8 Hohenpeis~ Temperat~ ETS_M~ 27.2  -3751. 7536. 7537. 7610.  4.43  4.44  1.69
```



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

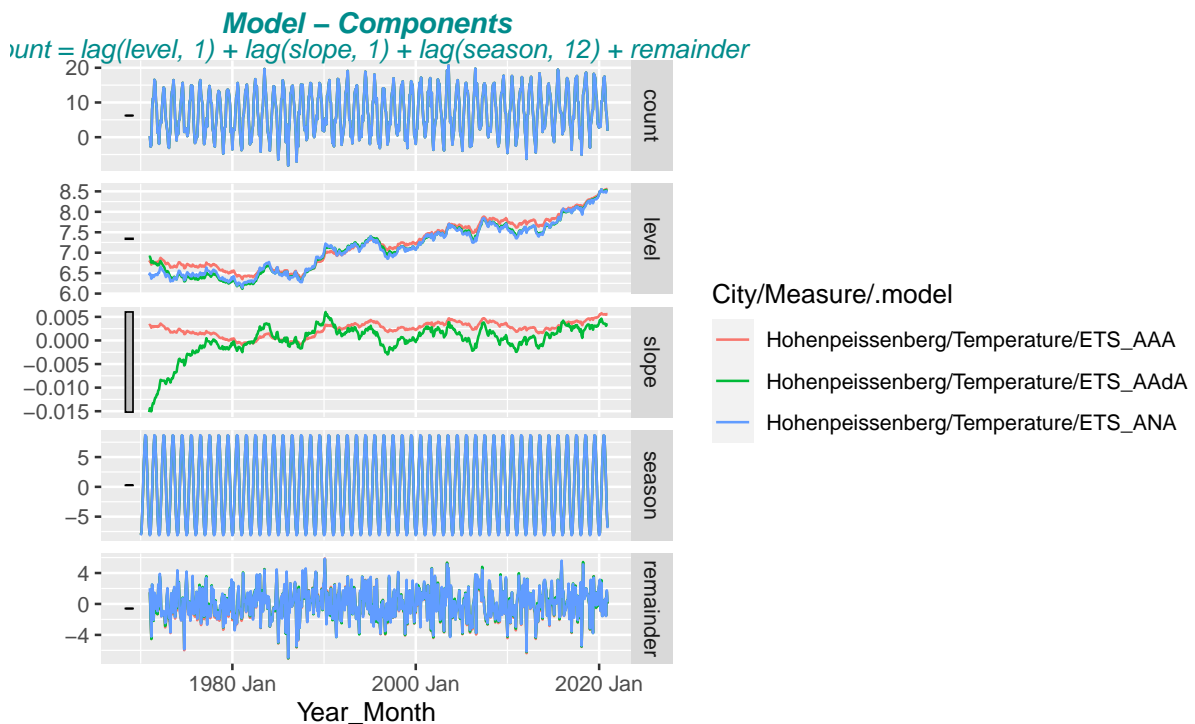
- MAE (Mean absolute error, will lead to forecasts of the median) and
- RMSE (Root mean squared error, lead to forecasts of the mean)

```
#> # A tibble: 8 x 11
#>   City      Measure .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  ACF1
#>   <chr>    <fct>    <chr> <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 Hohenpe~ Tempera~ ETS_AAA Trai~  0.0406  2.02  1.61 -47.5  137.  0.710  0.0806
#> 2 Hohenpe~ Tempera~ ETS_AMA Trai~  0.0352  2.02  1.61 -46.3  137.  0.710  0.0774
#> 3 Hohenpe~ Tempera~ ETS_AA~ Trai~  0.141  2.03  1.61 -44.9  135.  0.709  0.0752
#> 4 Hohenpe~ Tempera~ ETS_ANA Trai~  0.149  2.03  1.62 -46.0  137.  0.713  0.0721
#> 5 Hohenpe~ Tempera~ ETS_MMA Trai~ -0.0602  2.10  1.66 -41.0  143.  0.732  0.108
#> 6 Hohenpe~ Tempera~ ETS_MAA Trai~ -0.183  2.32  1.83 -45.7  169.  0.805  0.254
#> 7 Hohenpe~ Tempera~ ETS_MNA Trai~  0.00755  2.36  1.89 -26.4  159.  0.834 -0.0491
#> 8 Hohenpe~ Tempera~ ETS_MA~ Trai~  0.0353  2.72  2.15 -38.0  179.  0.949 -0.0935
```


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_AMA    39.0 0.126
#> 2 Hohenpeissenberg Temperature ETS_AAA    39.1 0.124
#> 3 Hohenpeissenberg Temperature ETS_AAdA   40.2 0.102
#> 4 Hohenpeissenberg Temperature ETS_ANA    40.4 0.0976
#> 5 Hohenpeissenberg Temperature ETS_MMA    44.8 0.0400
#> 6 Hohenpeissenberg Temperature ETS_MNA    99.2 0.00000000246
#> 7 Hohenpeissenberg Temperature ETS_MAA   380. 0
#> 8 Hohenpeissenberg Temperature ETS_MAdA   165. 0
```

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

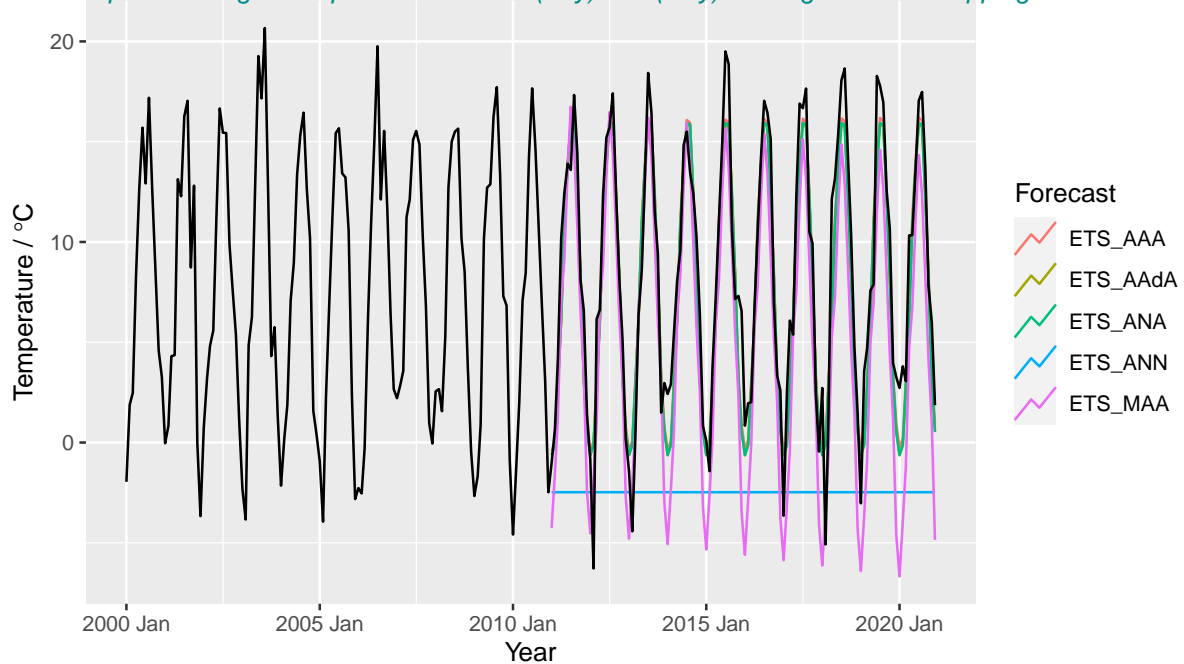


2.1.4 Forecast Accuracy with Training/Test Data

```
#> # A tibble: 5 x 11
#>   .model City          Measure .type ME RMSE MAE MPE MAPE MASE ACF1
#>   <chr>   <chr>         <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 ETS_AAA Hohenpeis~ Tempera~ Test  0.692 2.17 1.73 2.34 66.3 0.784 0.0210
#> 2 ETS_ANA Hohenpeis~ Tempera~ Test  0.929 2.25 1.81 12.2 66.0 0.822 0.0386
#> 3 ETS_AA~ Hohenpeis~ Tempera~ Test  0.938 2.26 1.82 10.5 67.5 0.826 0.0343
#> 4 ETS_MAA Hohenpeis~ Tempera~ Test  3.27 4.13 3.54 111. 158. 1.61 0.236
#> 5 ETS_ANN Hohenpeis~ Tempera~ Test 10.8 12.5 11.0 184. 199. 4.98 0.768
```

Accuracy of Monthly Forecasts

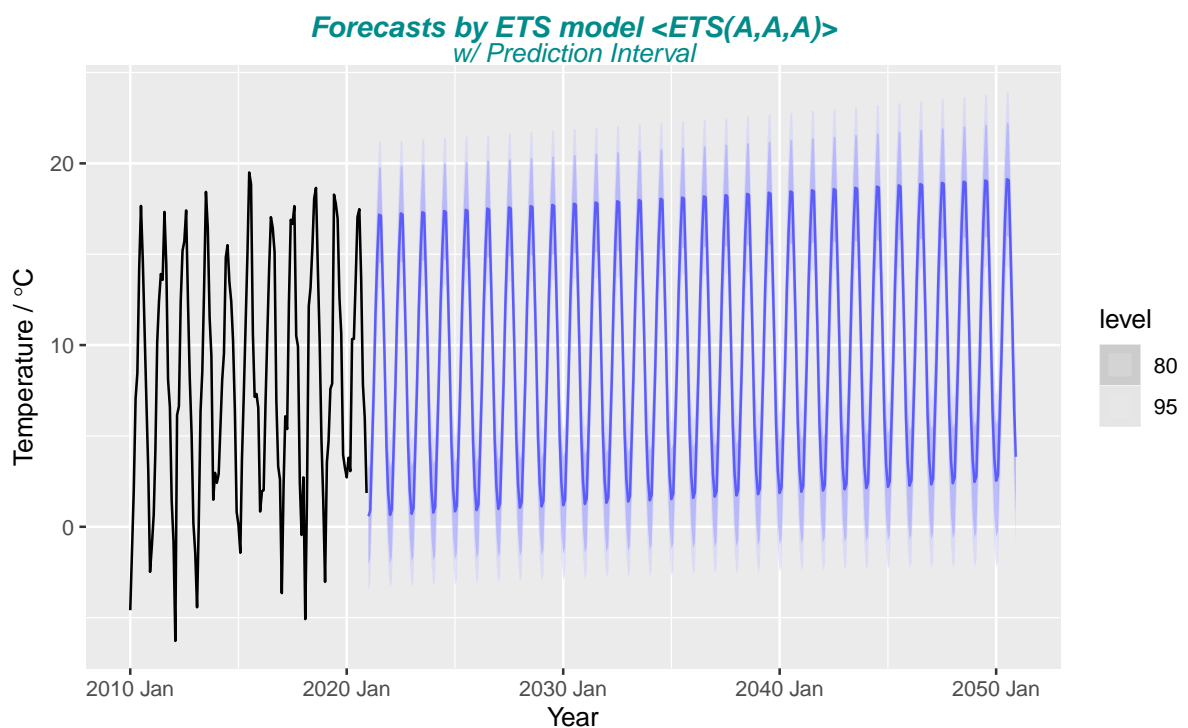
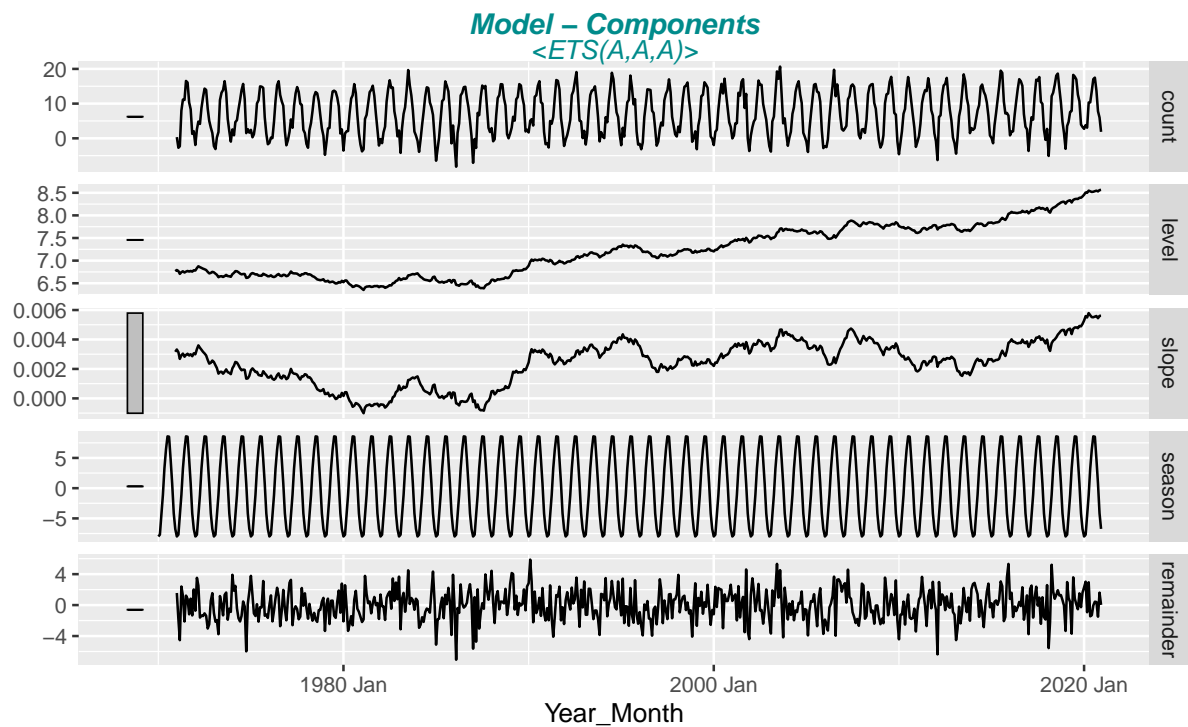
Hohenpeissenberg – Temperature note: ET(Axy)/ETS(Mxy) are in general overlapping



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

2.2.1 Forecast Plot of selected ETS model

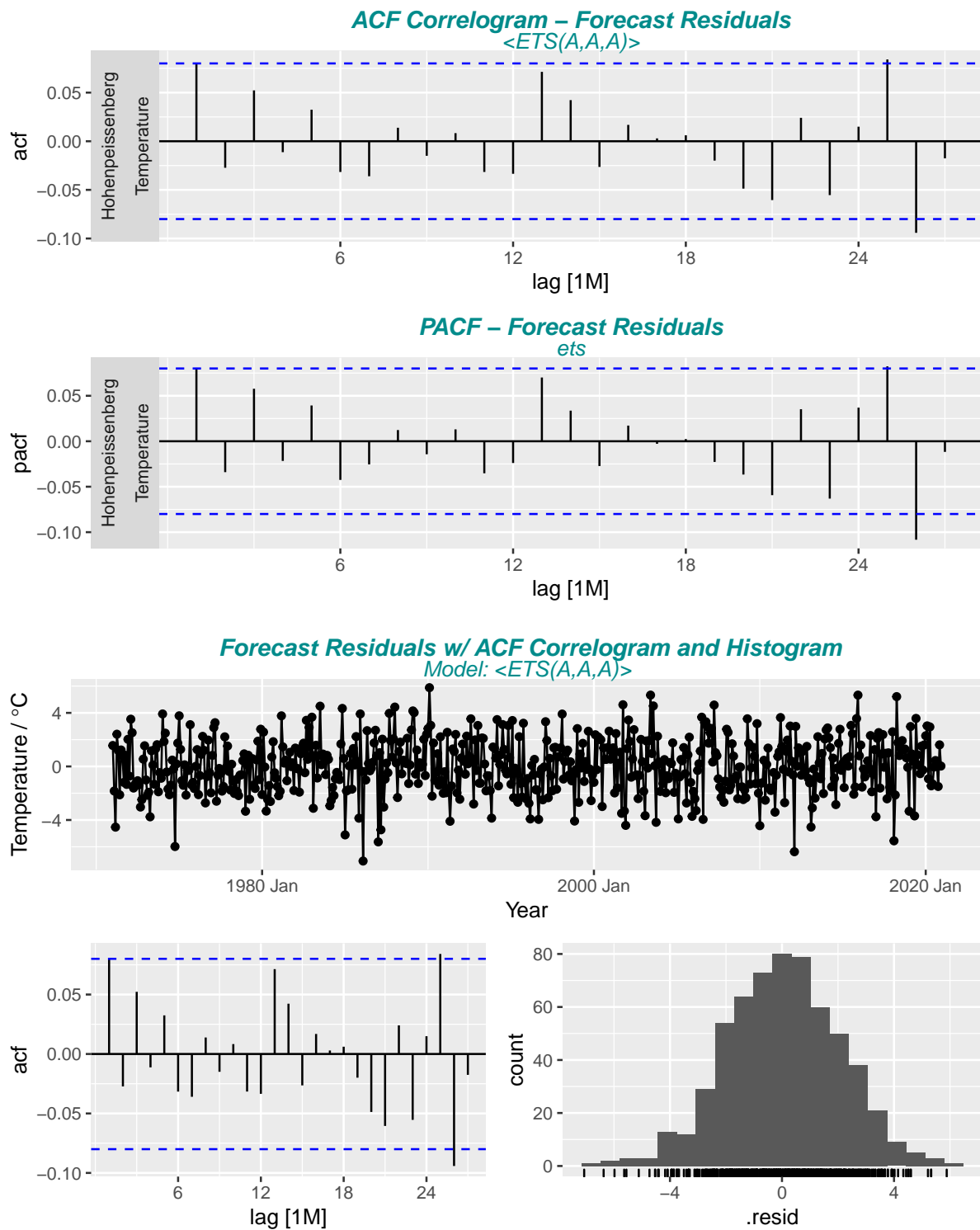
```
#> Provide model coefficients by report(fit_model)
#> Series: count
#> Model: ETS(A,A,A)
#> Smoothing parameters:
#>   alpha = 0.014561
#>   beta  = 0.0001004411
#>   gamma = 0.000101192
#>
#> Initial states:
#>      l      b      s1      s2      s3      s4      s5      s6
#> 6.766848 0.003163054 -6.733642 -4.165398 0.7660276 4.747525 8.510702 8.572779
#>      s7      s8      s9      s10     s11     s12
#> 6.403173 3.235912 -1.124447 -4.528056 -7.691808 -7.992768
#>
#> sigma^2: 4.2016
#>
#>      AIC      AICc      BIC
#> 4717.224 4718.276 4791.972
```



2.2.2 Residual Stationarity

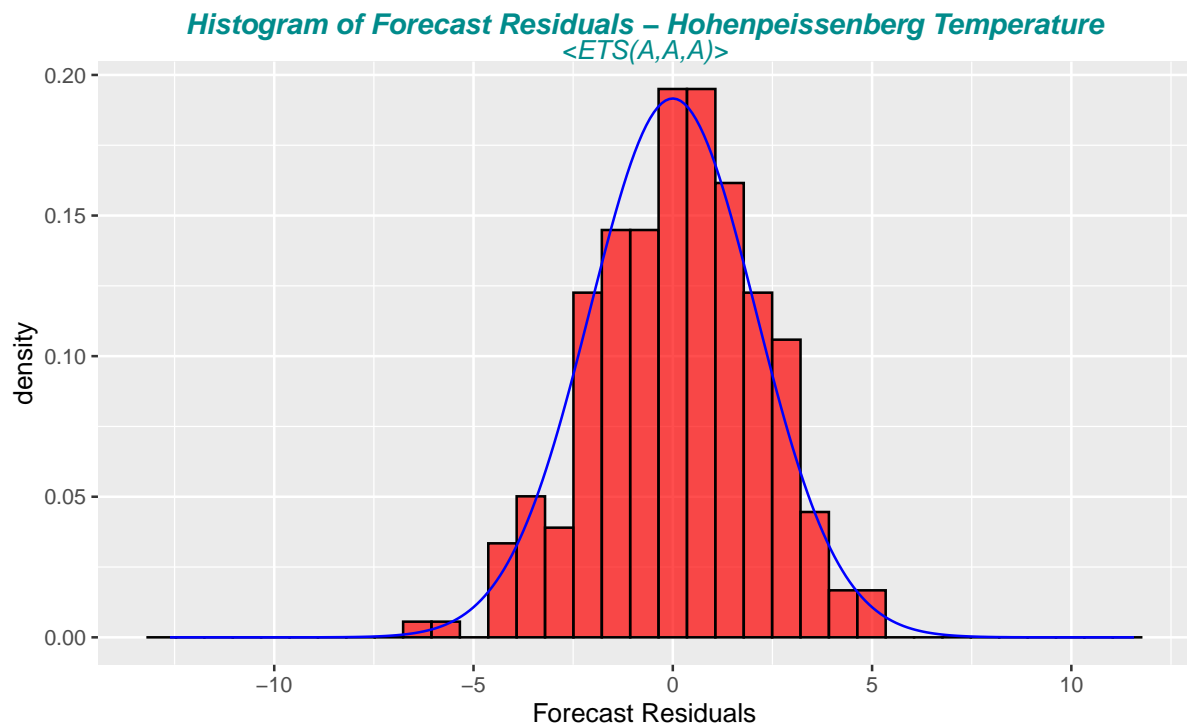
Required checks to be ready for forecasting:

- 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      31.2    0.404
```



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

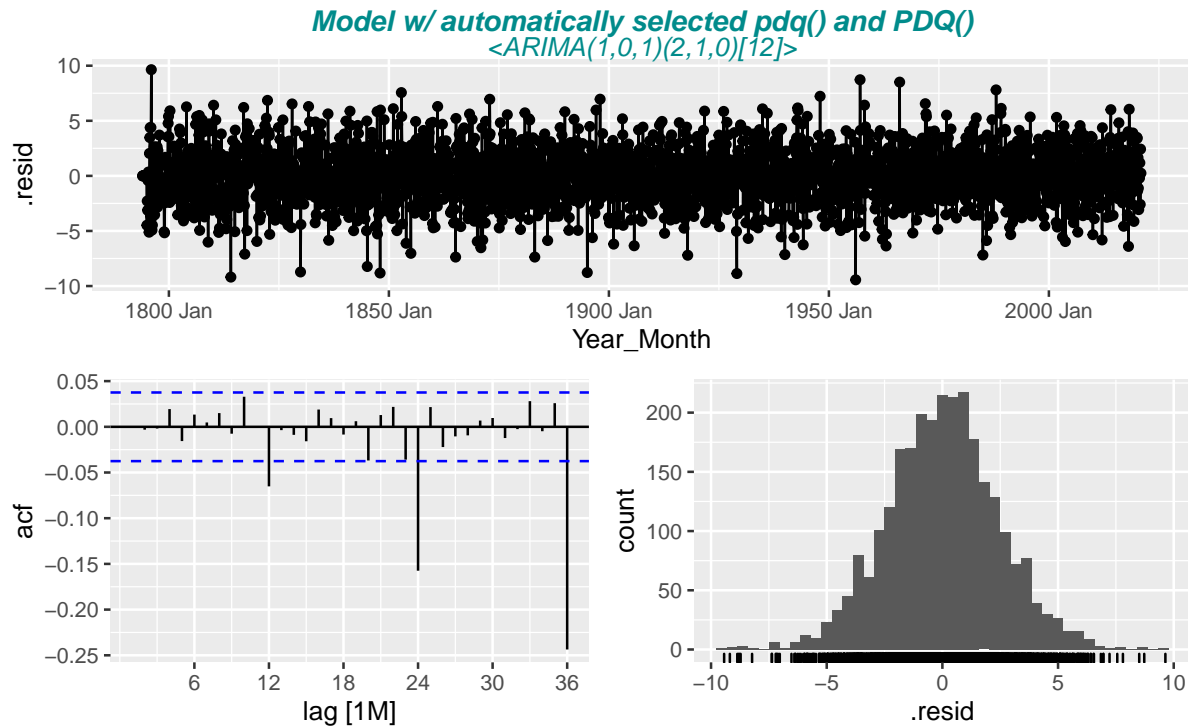
Non-seasonal ARIMA models are generally denoted $ARIMA(p,d,q)$ where parameters p , d , and q are non-negative integers, * p is the order (number of time lags) of the autoregressive model * d is the degree of differencing (number of times the data have had past values subtracted) * q is the order of 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 Hohenpe~ Tempera~ arima    5.92 -6260. 12530. 12530. 12560. <cpl [2~ <cpl [1~
#> Series: count
#> Model: ARIMA(1,0,1)(2,1,0)[12]
```

```
#>
#> Coefficients:
#>      ar1      ma1      sar1      sar2
#>    0.3362 -0.2267 -0.6928 -0.3121
#> s.e.  0.1664  0.1723  0.0183  0.0185
#>
#> sigma^2 estimated as 5.918:  log likelihood=-6260.23
#> AIC=12530.45  AICc=12530.47  BIC=12559.98
```



```
#> 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_roots ma_roots
#>   <chr>    <fct>    <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <list>  <list>
#> 1 Hohenpe~ Tempera~ ARIMA_0~  4.22 -1275. 2559. 2559. 2581. <cpl [0~ <cpl [2~
#> 2 Hohenpe~ Tempera~ ARIMA_2~  4.21 -1275. 2559. 2559. 2581. <cpl [2~ <cpl [1~
#> 3 Hohenpe~ Tempera~ ARIMA_1~  4.22 -1275. 2560. 2560. 2581. <cpl [1~ <cpl [2~
#> 4 Hohenpe~ Tempera~ ARIMA_1~  4.22 -1275. 2562. 2562. 2588. <cpl [1~ <cpl [2~
#> 5 Hohenpe~ Tempera~ ARIMA_1~  5.44 -1334. 2676. 2676. 2694. <cpl [2~ <cpl [0~
#> 6 Hohenpe~ Tempera~ ARIMA_1~  6.08 -1365. 2740. 2740. 2762. <cpl [1~ <cpl [0~
#> 7 Hohenpe~ Tempera~ ARIMA_2~  6.08 -1365. 2740. 2740. 2762. <cpl [1~ <cpl [0~
#> 8 Hohenpe~ Tempera~ ARIMA_3~  5.92 -1393. 2802. 2803. 2837. <cpl [2~ <cpl [1~
#> 9 Hohenpe~ Tempera~ ARIMA_2~  7.66 -1431. 2871. 2871. 2888. <cpl [1~ <cpl [0~
#> 10 Hohenpe~ Tempera~ ARIMA_0~  8.37 -1459. 2923. 2923. 2936. <cpl [0~ <cpl [2~
#> 11 Hohenpe~ Tempera~ ARIMA_1~  8.37 -1459. 2923. 2923. 2936. <cpl [1~ <cpl [1~
#> 12 Hohenpe~ Tempera~ ARIMA_0~ 11.3  -1547. 3097. 3097. 3106. <cpl [1~ <cpl [0~
#> 13 Hohenpe~ Tempera~ ARIMA_1~ 12.7  -1578. 3160. 3160. 3169. <cpl [1~ <cpl [0~
```

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)

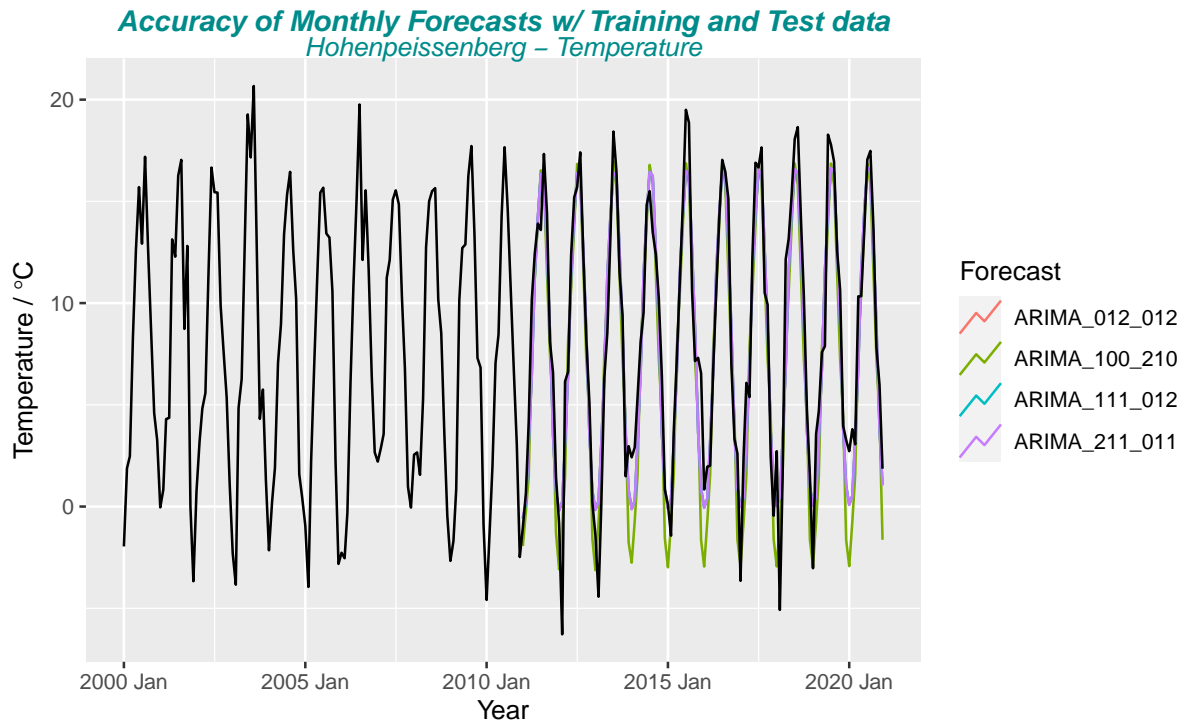
```
#> # A tibble: 14 x 11
#>   City Measure .model .type      ME    RMSE    MAE    MPE  MAPE    MASE
#>   <chr> <fct>   <chr> <chr>    <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>
#> 1 Hohe~ Temper~ ARIMA~ Trai~ 8.39e-2 2.02  1.58 -40.2 139.  0.695
#> 2 Hohe~ Temper~ ARIMA~ Trai~ 8.10e-2 2.02  1.58 -36.2 137.  0.695
#> 3 Hohe~ Temper~ ARIMA~ Trai~ 8.24e-2 2.02  1.58 -37.5 136.  0.694
#> 4 Hohe~ Temper~ ARIMA~ Trai~ 8.22e-2 2.02  1.58 -35.8 135.  0.694
#> 5 Hohe~ Temper~ ARIMA~ Trai~ 8.53e-2 2.30  1.82  15.4 147.  0.804
#> 6 Hohe~ Temper~ ARIMA~ Trai~ 3.81e-2 2.42  1.94 -33.8 178.  0.854
#> 7 Hohe~ Temper~ ARIMA~ Trai~ -2.12e-3 2.43  1.91 -29.5 167.  0.843
#> 8 Hohe~ Temper~ ARIMA~ Trai~ -2.12e-3 2.43  1.91 -29.5 167.  0.843
#> 9 Hohe~ Temper~ ARIMA~ Trai~ -4.95e-4 2.73  2.17 -49.3 190.  0.958
#> 10 Hohe~ Temper~ ARIMA~ Trai~ 3.81e-2 2.86  2.22 -51.2 242.  0.980
#> 11 Hohe~ Temper~ ARIMA~ Trai~ 3.79e-2 2.86  2.22 -49.8 241.  0.980
#> 12 Hohe~ Temper~ ARIMA~ Trai~ 4.51e-3 3.33  2.58 -149. 296.  1.14
#> 13 Hohe~ Temper~ ARIMA~ Trai~ 3.29e-3 3.52  2.74 -65.6 256.  1.21
#> 14 Hohe~ Temper~ ARIMA~ Trai~ NaN      NaN    NaN    NaN   NaN   NaN
#> # ... with 1 more variable: 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    34.5 2.62e- 1
#> 2 Hohenpeissenberg Temperature ARIMA_012_012    35.7 2.18e- 1
#> 3 Hohenpeissenberg Temperature ARIMA_111_012    36.1 2.04e- 1
#> 4 Hohenpeissenberg Temperature ARIMA_111_112    36.2 2.01e- 1
#> 5 Hohenpeissenberg Temperature ARIMA_301_200    78.6 3.12e- 6
#> 6 Hohenpeissenberg Temperature ARIMA_100_210    97.6 4.45e- 9
#> 7 Hohenpeissenberg Temperature ARIMA_100_110   109. 7.07e-11
#> 8 Hohenpeissenberg Temperature ARIMA_200_110   109. 7.07e-11
#> 9 Hohenpeissenberg Temperature ARIMA_010_110   291. 0.
#> 10 Hohenpeissenberg Temperature ARIMA_012_010   208. 0.
#> 11 Hohenpeissenberg Temperature ARIMA_110_010   383. 0.
#> 12 Hohenpeissenberg Temperature ARIMA_111_010   209. 0.
#> 13 Hohenpeissenberg Temperature ARIMA_210_110   157. 0.
#> 14 Hohenpeissenberg Temperature ARIMA_002_200    NA    NA
```

3.1.3 Forecast Accuracy with Training/Test Data

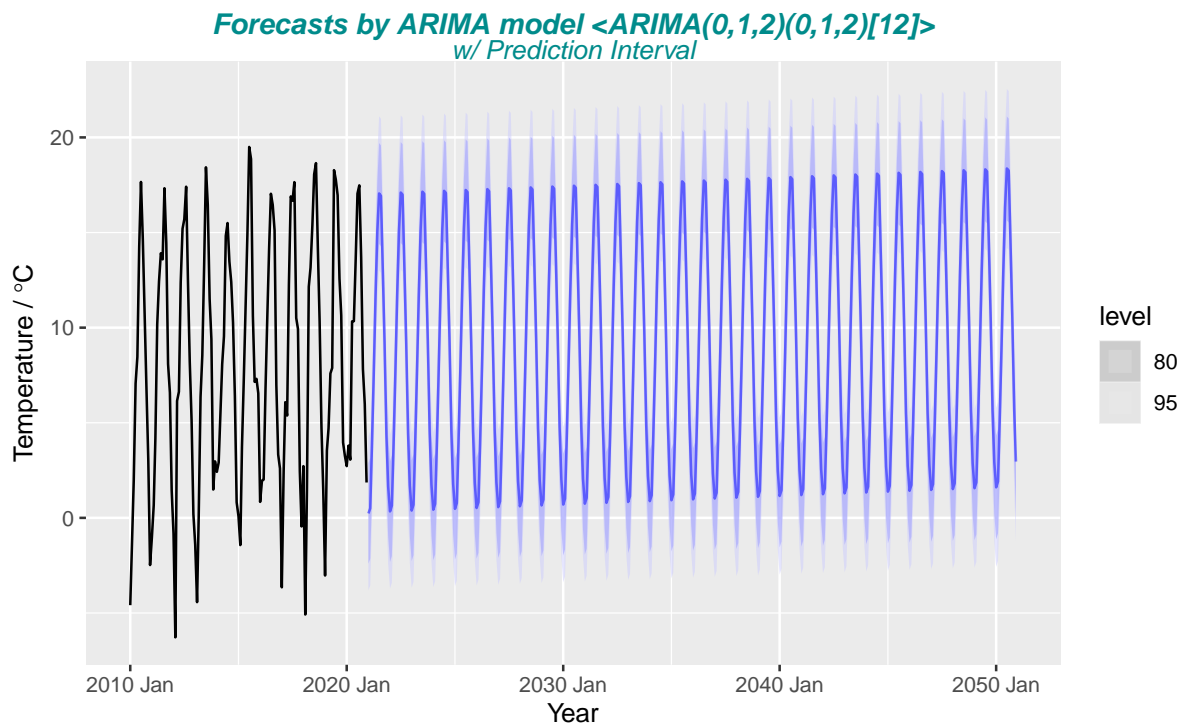
```
#> # A tibble: 4 x 11
#>   .model City Measure .type      ME    RMSE    MAE    MPE  MAPE    MASE  ACF1
#>   <chr>   <chr>   <fct>   <chr>    <dbl> <dbl>  <dbl> <dbl> <dbl>  <dbl>  <dbl>
#> 1 ARIMA_21~ Hohenpeis~ Tempera~ Test  0.418 2.08  1.64 -3.36 63.4 0.743 0.0210
#> 2 ARIMA_01~ Hohenpeis~ Tempera~ Test  0.422 2.09  1.64 -3.84 63.6 0.744 0.0204
#> 3 ARIMA_11~ Hohenpeis~ Tempera~ Test  0.423 2.09  1.64 -3.82 63.7 0.744 0.0203
#> 4 ARIMA_10~ Hohenpeis~ Tempera~ Test  1.27 2.56  2.04 53.0 100. 0.928 0.0313
```



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

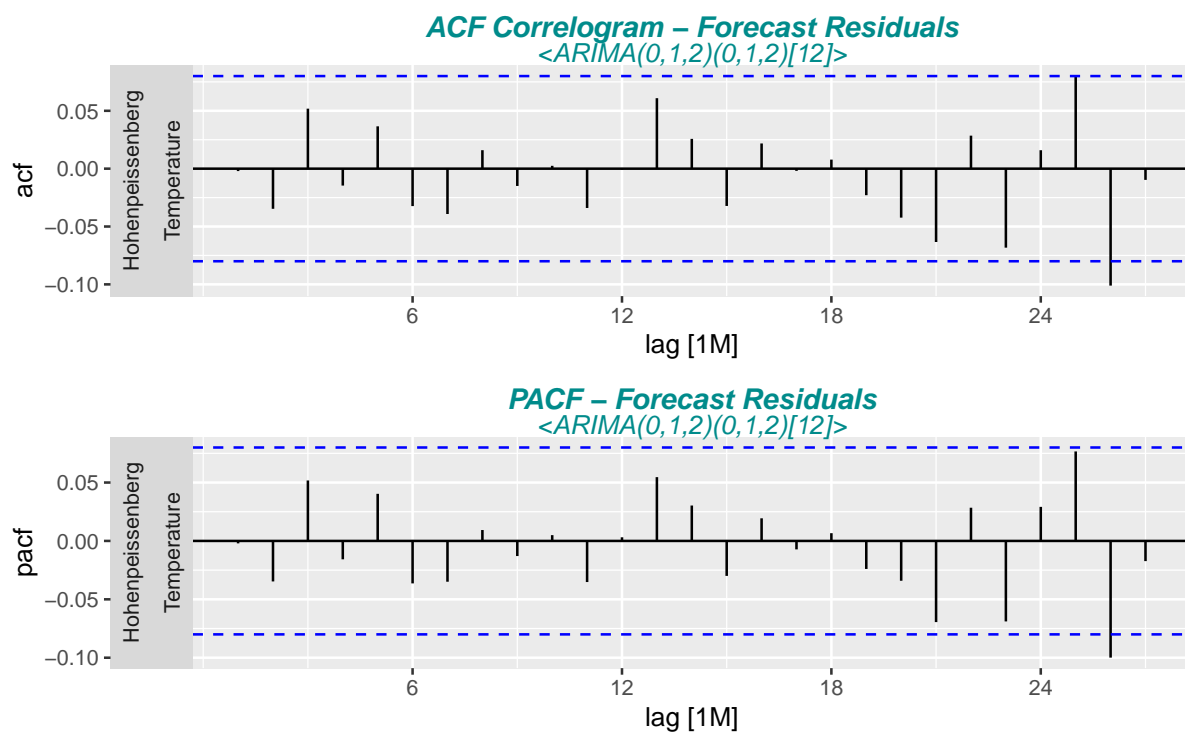
```
#> Provide model coefficients by report(fit_model)
#> Series: count
#> Model: ARIMA(0,1,2)(0,1,2)[12]
#>
#> Coefficients:
#>      ma1      ma2      sma1      sma2
#>    -0.9113 -0.0886 -0.9909  0.0285
#> s.e.    0.0449  0.0431  0.0432  0.0419
#>
#> sigma^2 estimated as 4.217:  log likelihood=-1274.61
#> AIC=2559.21  AICc=2559.31  BIC=2581.09
```

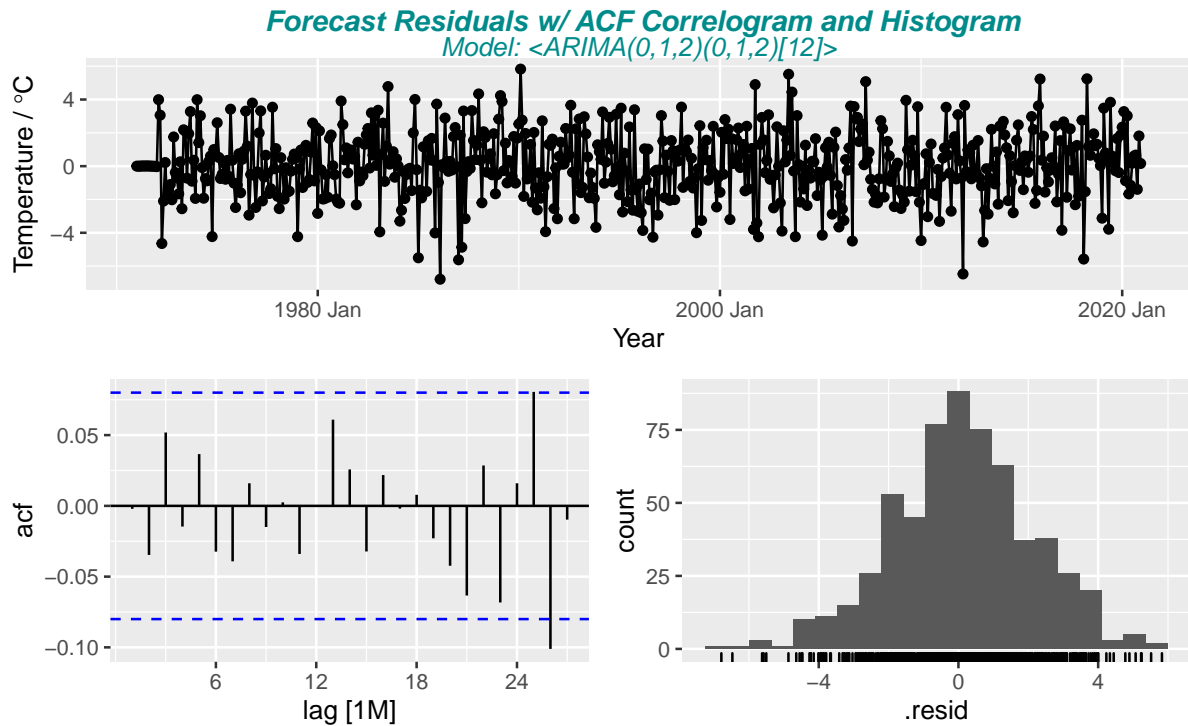



3.2.2 Residual Stationarity

Required checks to be ready for forecasting:

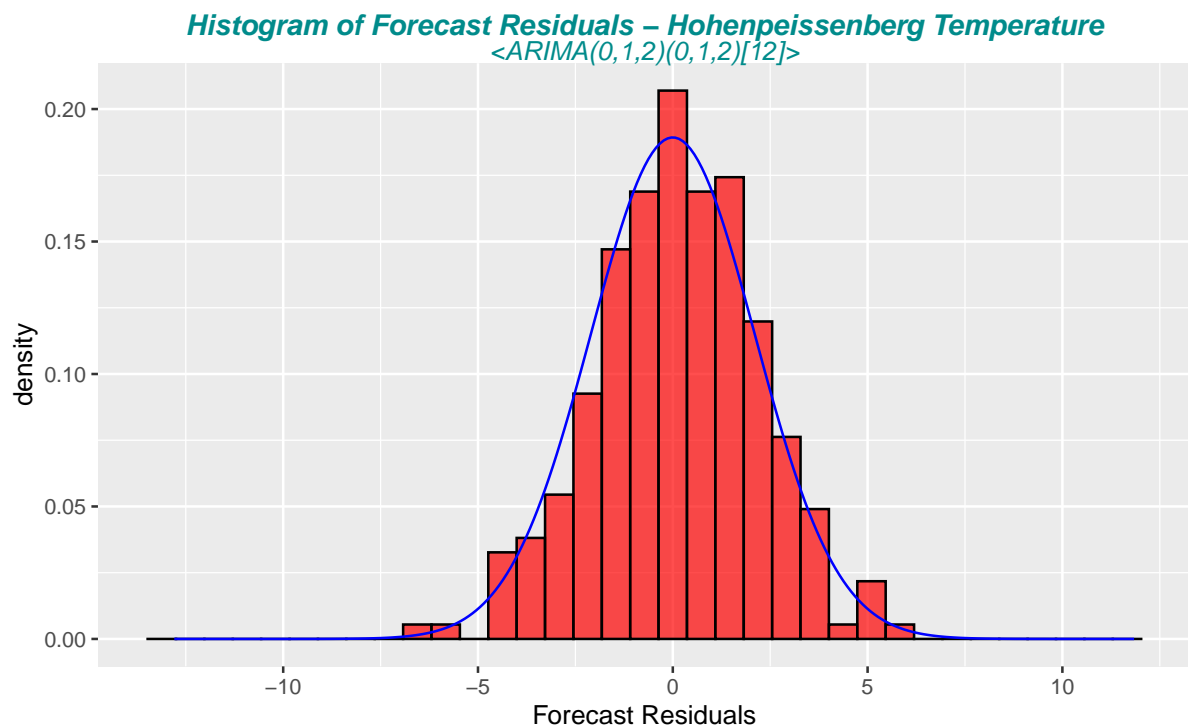
- ACF Forecast Residual: all spikes are within the significance limits, so the residuals appear to be white noise
- The Ljung-Box test also shows that the residuals have no remaining autocorrelations
- Forecast Residuals are more or less normally distributed with roughly centred on zero





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     26.1     0.667
```



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.

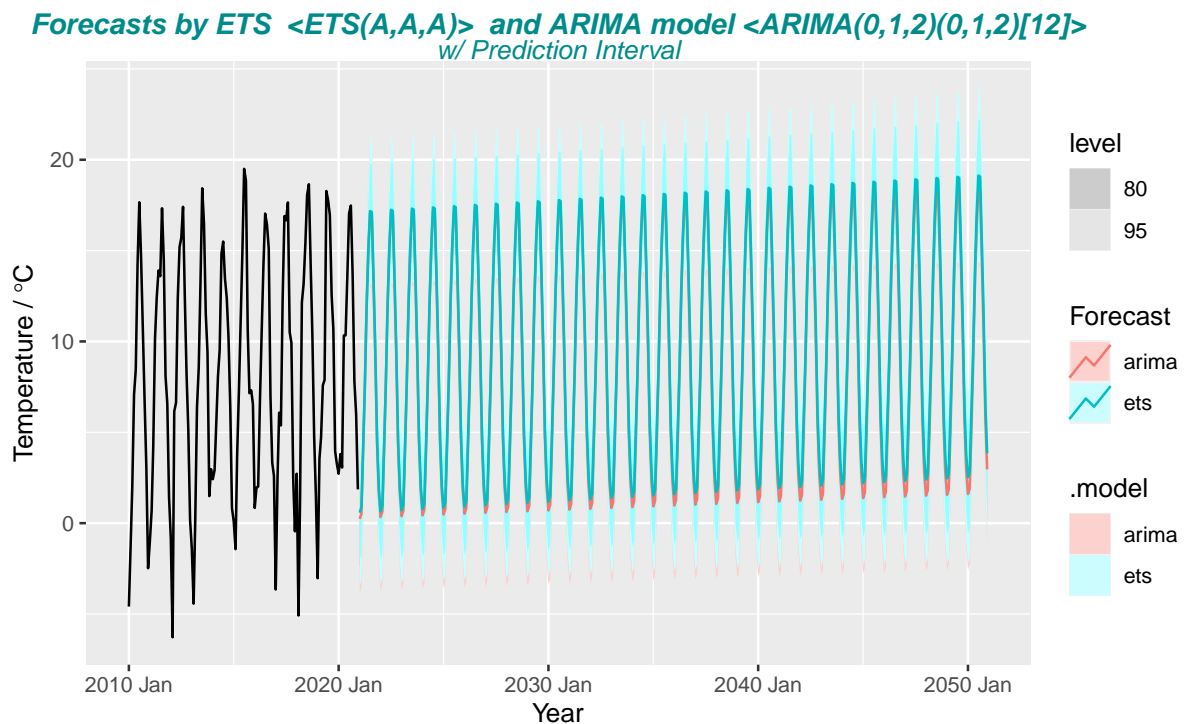
- Residual Accuracy with one-step-ahead fitted residuals
- Forecast Accuracy with Training/Test Data

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 11
#>   City      Measure .model .type    ME  RMSE  MAE  MPE  MAPE  MASE    ACF1
#>   <chr>    <fct>    <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 Hohenpe~ Tempera~ ets    Trai~ 0.0406 2.02 1.61 -47.5 137. 0.710 0.0806
#> 2 Hohenpe~ Tempera~ arima  Trai~ 0.0824 2.02 1.58 -37.5 136. 0.694 -0.00208
#> 3 Hohenpe~ Tempera~ ETS_AAA Test 0.692 2.17 1.73 2.34 66.3 0.784 0.0210
#> 4 Hohenpe~ Tempera~ ARIMA_~ Test 0.422 2.09 1.64 -3.84 63.6 0.744 0.0204
```

4.0.2 Forecast Plot of selected ETS and ARIMA model



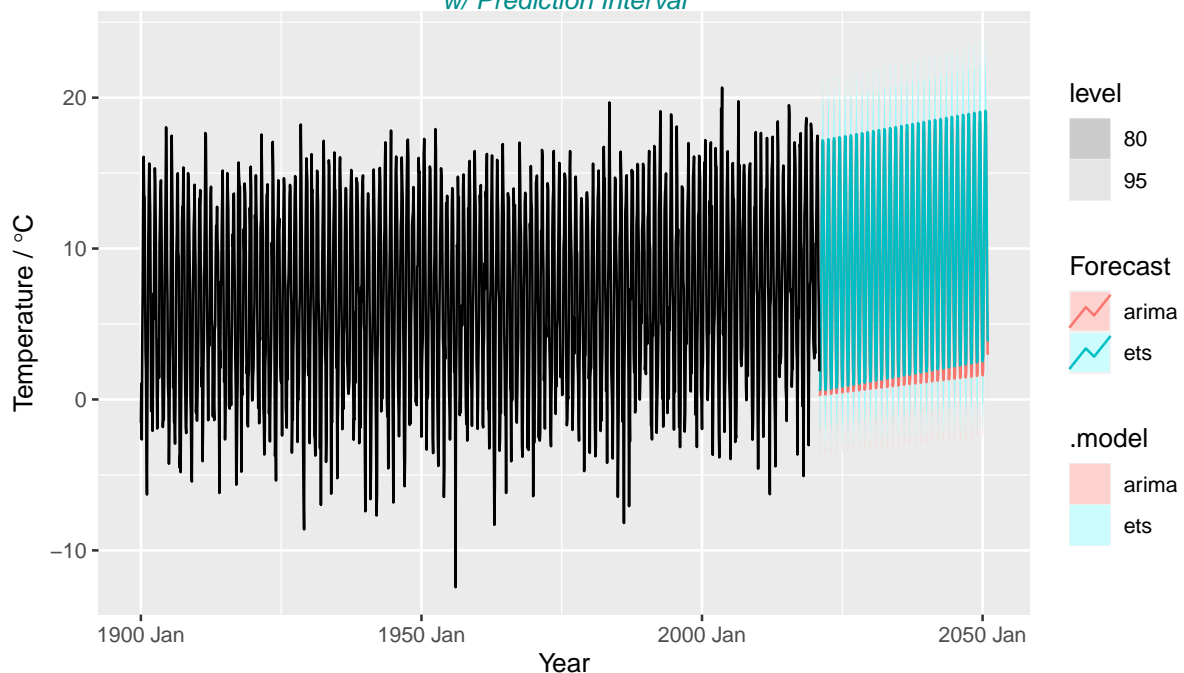
```
#> # A tsibble: 6 x 8 [1M]
#> # Key:      City, Measure, .model [2]
#> # Groups:   City, Measure, .model [2]
```

```

#>   City Measure .model Year_Month      count .mean      '80%'
#>   <chr> <fct>  <chr>      <mth>      <dist> <dbl>      <hilo>
#> 1 Hohe~ Temper~ arima    2021 Jan N(0.23, 4.2) 0.234 [-2.404544, 2.873214]80
#> 2 Hohe~ Temper~ arima    2021 Feb N(0.5, 4.3) 0.500 [-2.150387, 3.149956]80
#> 3 Hohe~ Temper~ arima    2021 Mrz N(3.7, 4.3) 3.73  [ 1.075390, 6.375732]80
#> 4 Hohe~ Temper~ ets      2021 Jan N(0.58, 4.2) 0.578 [-2.048710, 3.205107]80
#> 5 Hohe~ Temper~ ets      2021 Feb N(0.88, 4.2) 0.885 [-1.742634, 3.511747]80
#> 6 Hohe~ Temper~ ets      2021 Mrz N(4.1, 4.2) 4.05  [ 1.426025, 6.680978]80
#> # ... with 1 more variable: '95%' <hilo>
#> # A tibble: 6 x 8 [1M]
#> # Key:      City, Measure, .model [2]
#> # Groups:   City, Measure, .model [2]
#>   City Measure .model Year_Month      count .mean      '80%'
#>   <chr> <fct>  <chr>      <mth>      <dist> <dbl>      <hilo>
#> 1 Hohe~ Temper~ arima    2050 Okt N(11, 4.5) 10.5  [7.7764891, 13.240016]80
#> 2 Hohe~ Temper~ arima    2050 Nov N(5.7, 4.5)  5.71 [2.9756902,  8.439217]80
#> 3 Hohe~ Temper~ arima    2050 Dez N(3, 4.5)  2.96 [0.2284211,  5.691996]80
#> 4 Hohe~ Temper~ ets      2050 Okt N(11, 6) 11.3  [8.2125052, 14.465324]80
#> 5 Hohe~ Temper~ ets      2050 Nov N(6.4, 6)  6.41 [3.2838438,  9.542294]80
#> 6 Hohe~ Temper~ ets      2050 Dez N(3.9, 6)  3.85 [0.7182663,  6.982364]80
#> # ... with 1 more variable: '95%' <hilo>

```

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 Hohenpeissenberg Temperature arima    2021  0.234
#> 2 Hohenpeissenberg Temperature arima    2021  0.500
#> 3 Hohenpeissenberg Temperature arima    2021  3.73
#> 4 Hohenpeissenberg Temperature arima    2022  0.336
#> 5 Hohenpeissenberg Temperature arima    2022  0.644
#> 6 Hohenpeissenberg Temperature arima    2022  3.75
#> 7 Hohenpeissenberg Temperature arima    2023  0.382

```

```

#> 8 Hohenpeissenberg Temperature arima 2023 0.689
#> 9 Hohenpeissenberg Temperature arima 2023 3.80
#> 10 Hohenpeissenberg Temperature arima 2024 0.427
#> # ... 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 Hohenpeissenberg Temperature arima 2021 9.24
#> 2 Hohenpeissenberg Temperature arima 2021 4.35
#> 3 Hohenpeissenberg Temperature arima 2021 1.64
#> 4 Hohenpeissenberg Temperature arima 2022 9.24
#> 5 Hohenpeissenberg Temperature arima 2022 4.44
#> 6 Hohenpeissenberg Temperature arima 2022 1.69
#> 7 Hohenpeissenberg Temperature arima 2023 9.29
#> 8 Hohenpeissenberg Temperature arima 2023 4.49
#> 9 Hohenpeissenberg Temperature arima 2023 1.74
#> 10 Hohenpeissenberg Temperature arima 2024 9.33
#> # ... 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 Hohenpeissenberg Temperature arima    35.7    0.218
#> 2 Hohenpeissenberg Temperature ets      39.1    0.124

```

5 Yearly Data Forecasts with ARIMA and ETS

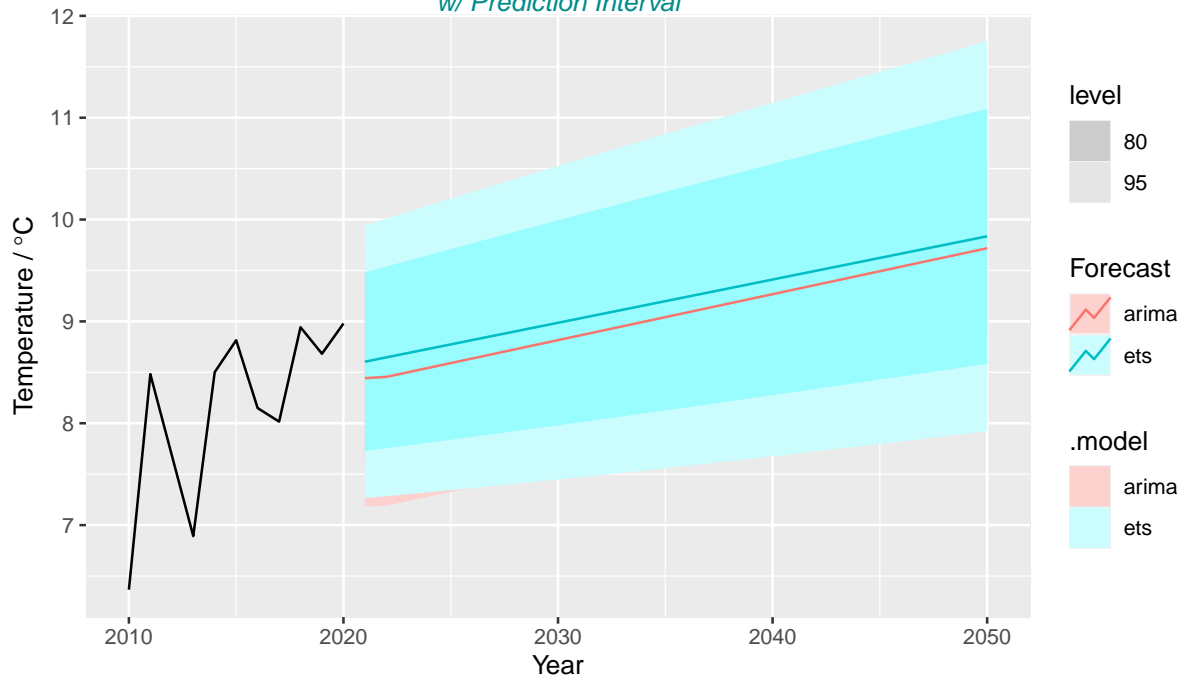
For yearly data the seasonal monthly data are replaced by the yearly average data. Therefore the seasonal component of the ETS and ARIMA model are to be taken out.

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)_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 differencing had to be selected $\$CO_2 \langle ARIMA(0,2,1) \text{ w/ poly} \rangle$. 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

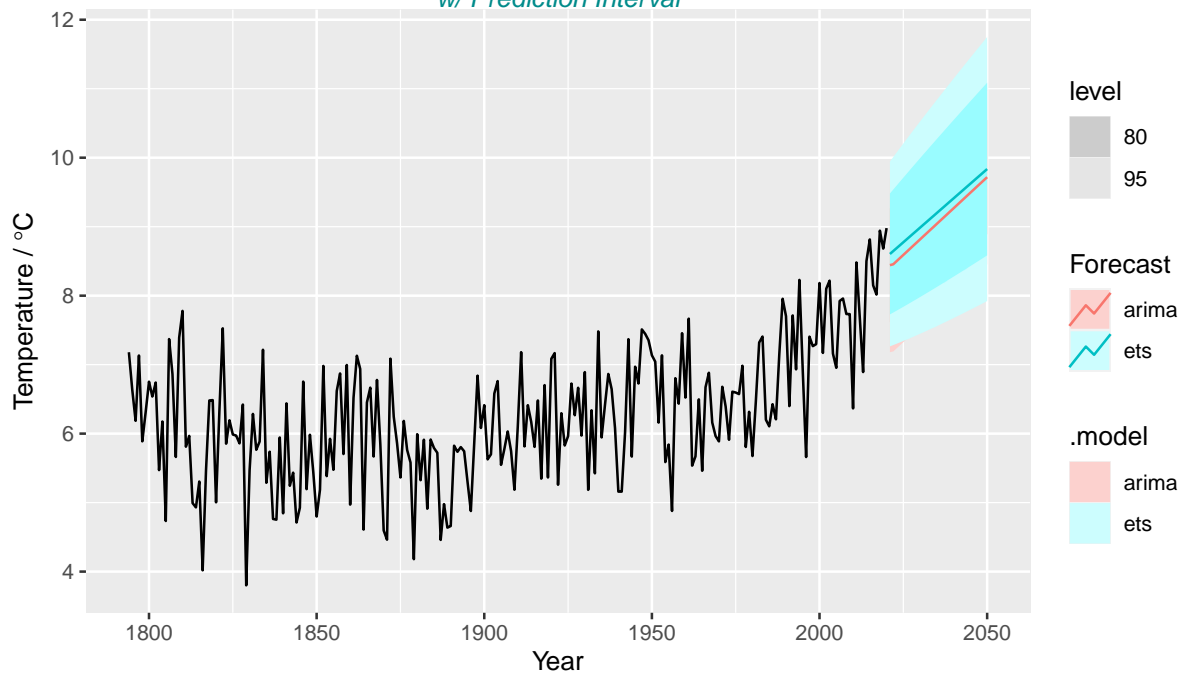
5.0.2 Forecast Plot of selected ETS and ARIMA model

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



```
#> # A 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 Hohe~ Temper~ arima    2021 N(8.4, 0.41)  8.44 [7.617625, 9.266683] 80
#> 2 Hohe~ Temper~ arima    2022 N(8.5, 0.42)  8.46 [7.628736, 9.281882] 80
#> 3 Hohe~ Temper~ arima    2023 N(8.5, 0.42)  8.50 [7.673841, 9.326987] 80
#> 4 Hohe~ Temper~ ets      2021 N(8.6, 0.47)  8.60 [7.728459, 9.480070] 80
#> 5 Hohe~ Temper~ ets      2022 N(8.6, 0.48)  8.65 [7.755549, 9.537901] 80
#> 6 Hohe~ Temper~ ets      2023 N(8.7, 0.5)   8.69 [7.782884, 9.595487] 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 Hohe~ Temper~ arima    2048 N(9.6, 0.42)  9.63 [8.801470, 10.45462] 80
#> 2 Hohe~ Temper~ arima    2049 N(9.7, 0.42)  9.67 [8.846575, 10.49972] 80
#> 3 Hohe~ Temper~ arima    2050 N(9.7, 0.42)  9.72 [8.891680, 10.54483] 80
#> 4 Hohe~ Temper~ ets      2048 N(9.8, 0.92)  9.75 [8.521664, 10.97973] 80
#> 5 Hohe~ Temper~ ets      2049 N(9.8, 0.94)  9.79 [8.552808, 11.03350] 80
#> 6 Hohe~ Temper~ ets      2050 N(9.8, 0.95)  9.84 [8.584043, 11.08719] 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 Hohenpeissenberg Temperature arima    33.3    0.310
#> 2 Hohenpeissenberg Temperature ets      33.2    0.313
```

6 Backup