DASC6510 Time Series Course Final Project

Changda Li (T00705321) & Duo Feng (T00704552)

2023-11-30

0. Laoding libaries and dataset

```
library(fable)
```

```
## Loading required package: fabletools
```

```
library(fabletools)
library(fpp3)
```

```
## -- Attaching packages -------------------------------------------- fpp3 0.5 --
```

```
## v tibble      3.2.1     v ggplot2     3.4.3
## v dplyr       1.1.3     v tsibble     1.1.3
## v tidyr       1.3.0     v tsibbledata 0.4.1
## v lubridate   1.9.2     v feasts      0.3.1
```

```
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(dplyr)
library(tsibble)
library(readxl)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:fabletools':
##
##     accuracy
```

```r
library(tsintermittent)
library(expsmooth)
library(fable.prophet)
```

```
## Loading required package: Rcpp
```

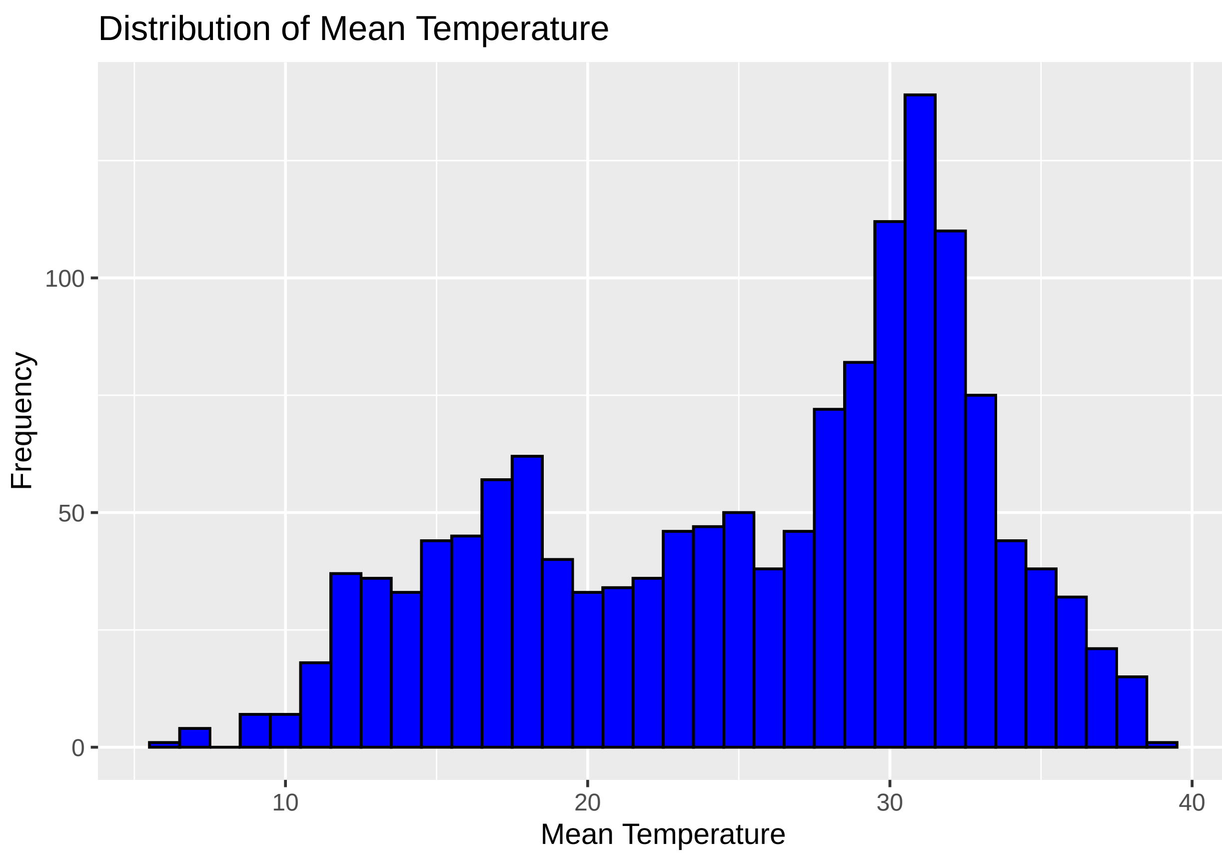
```r
library(ggplot2)
library(pheatmap)
library(imputeTS)
```

```r
setwd("./")
data_train <- read.csv("data/DailyDelhiClimateTrain.csv")
data_test <- read.csv("data/DailyDelhiClimateTest.csv")
```

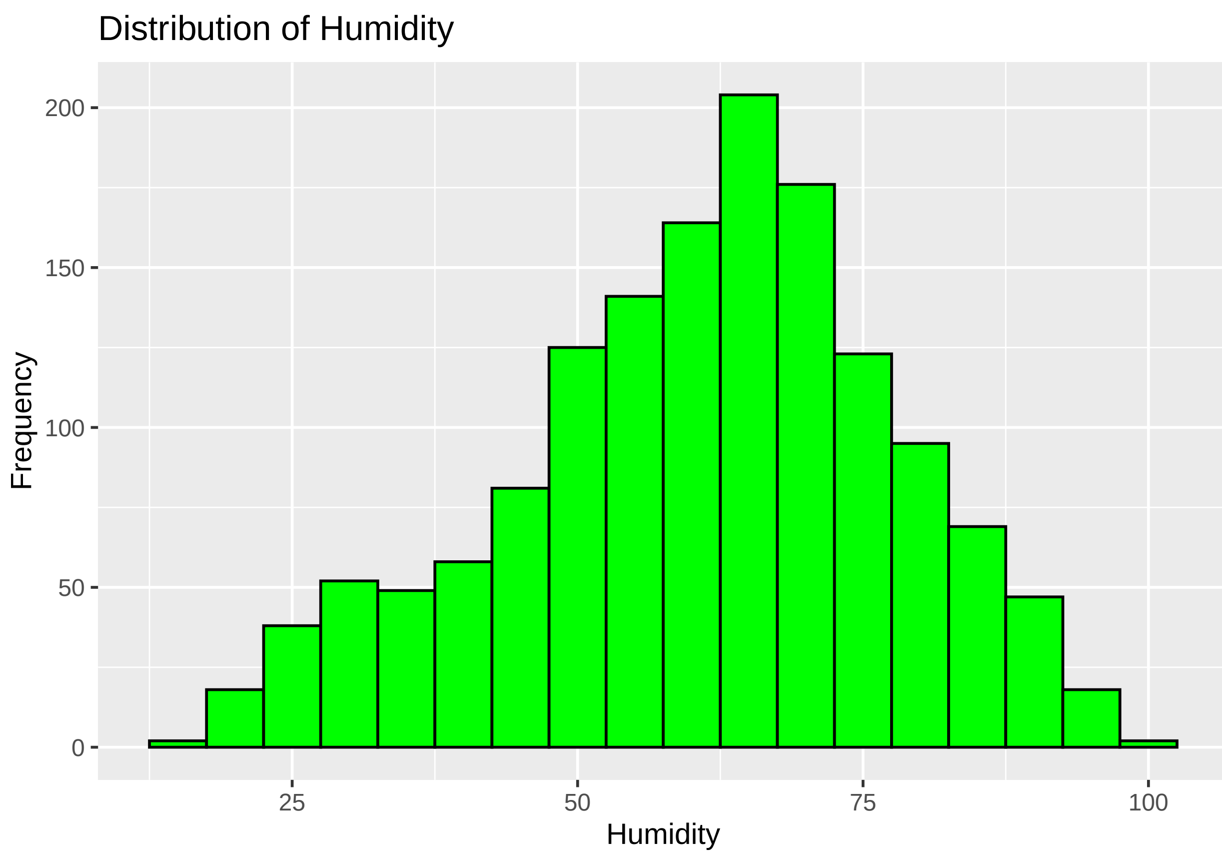
I. Exploratory data analysis

1. Distribution of the each variable

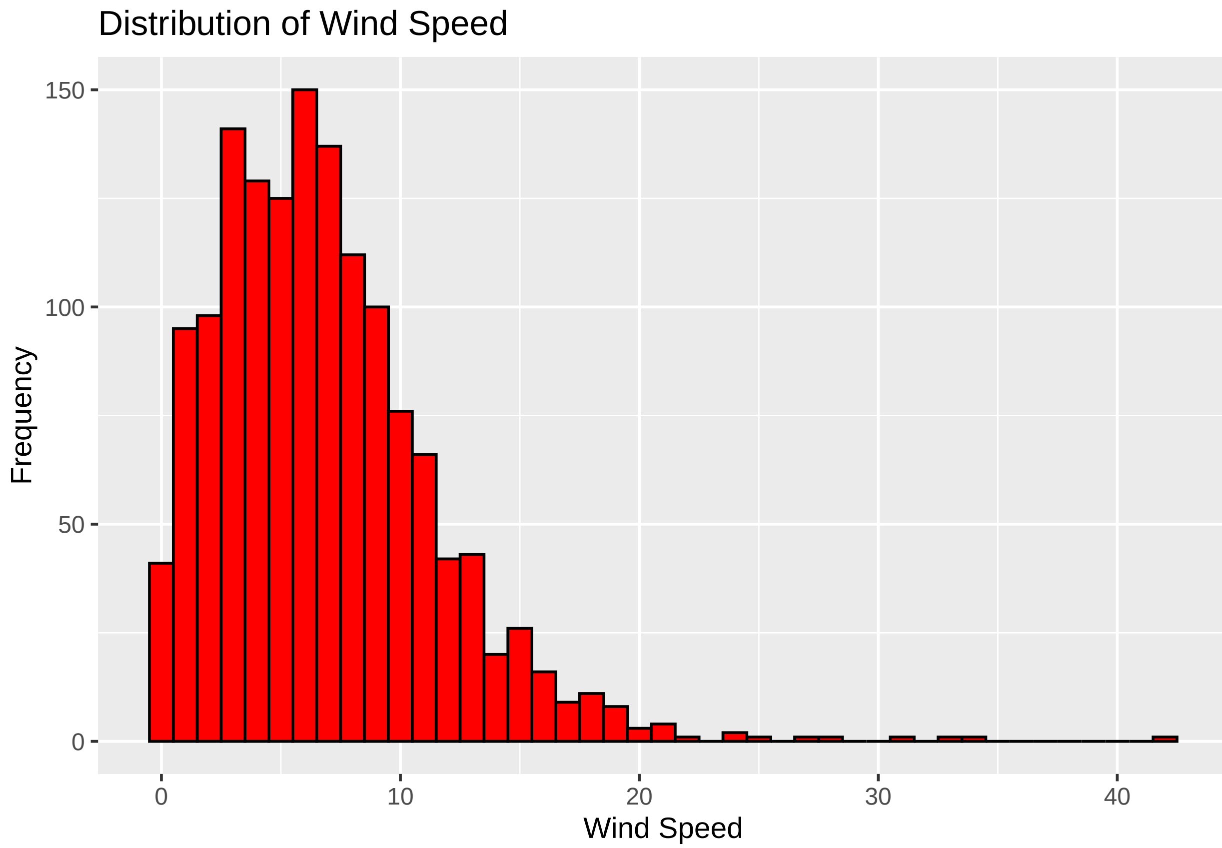
```r
# Plot the distribution of each variable
ggplot(data_train, aes(x = meantemp)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black") +
  labs(title = "Distribution of Mean Temperature", x = "Mean Temperature", y = "Frequency")
```



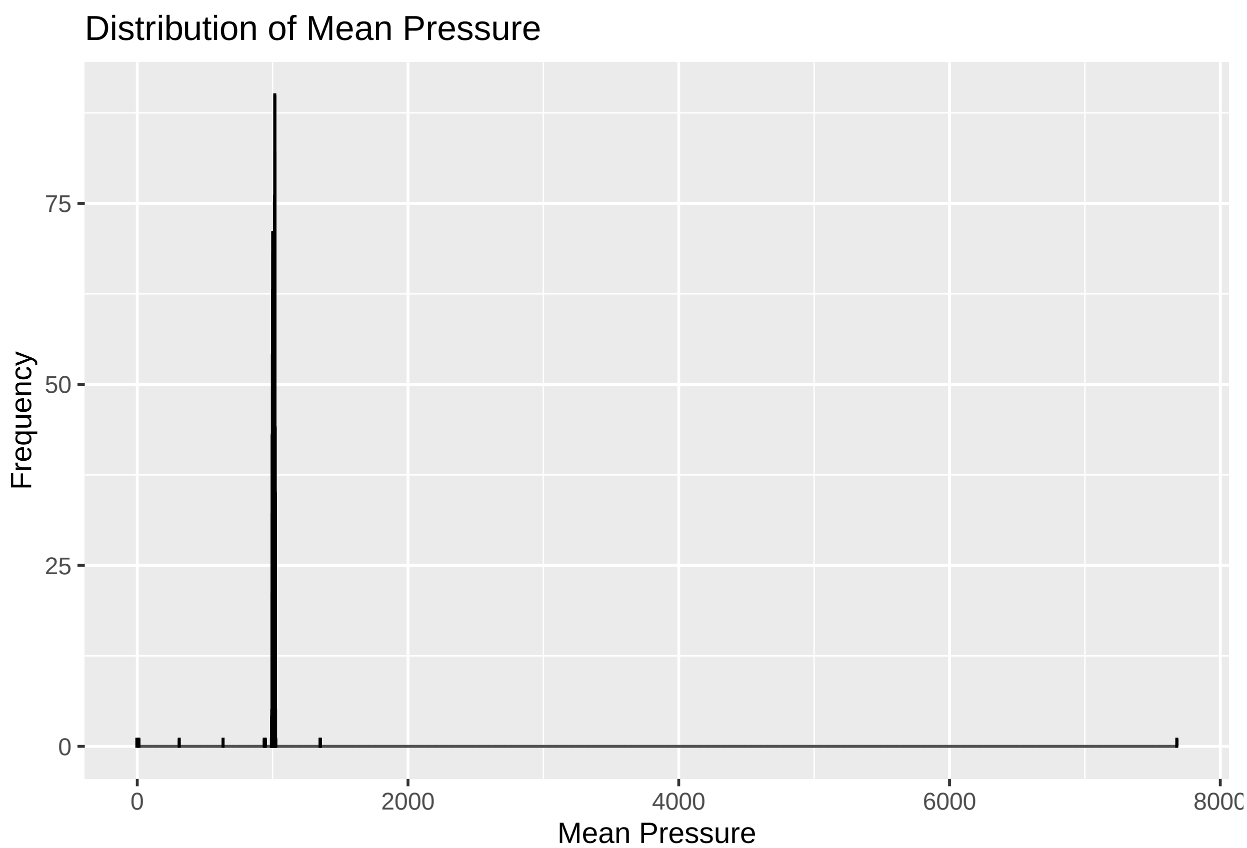

```r
ggplot(data_train, aes(x = humidity)) +
  geom_histogram(binwidth = 5, fill = "green", color = "black") +
  labs(title = "Distribution of Humidity", x = "Humidity", y = "Frequency")
```




```r
ggplot(data_train, aes(x = wind_speed)) +
  geom_histogram(binwidth = 1, fill = "red", color = "black") +
  labs(title = "Distribution of Wind Speed", x = "Wind Speed", y = "Frequency")
```

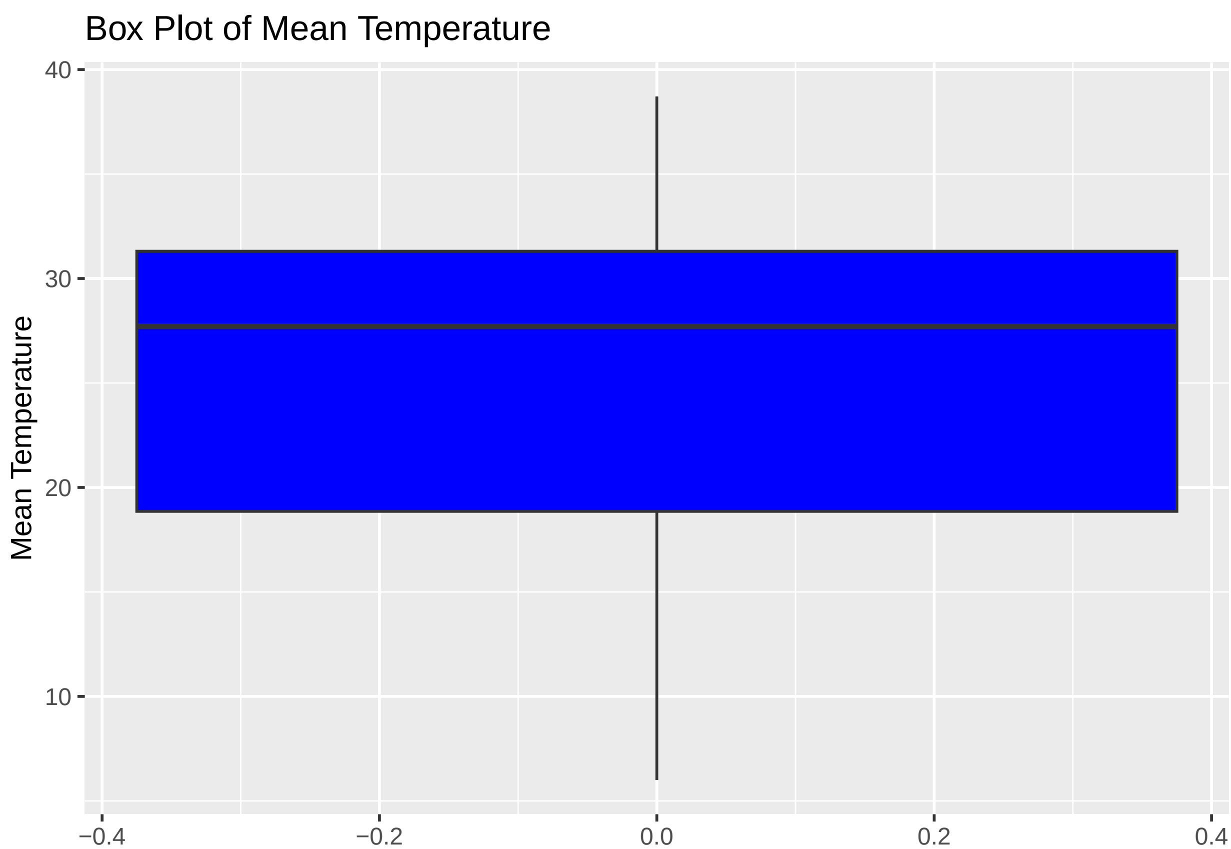



```r
ggplot(data_train, aes(x = meanpressure)) +
  geom_histogram(binwidth = 1, fill = "purple", color = "black") +
  labs(title = "Distribution of Mean Pressure", x = "Mean Pressure", y = "Frequency")
```

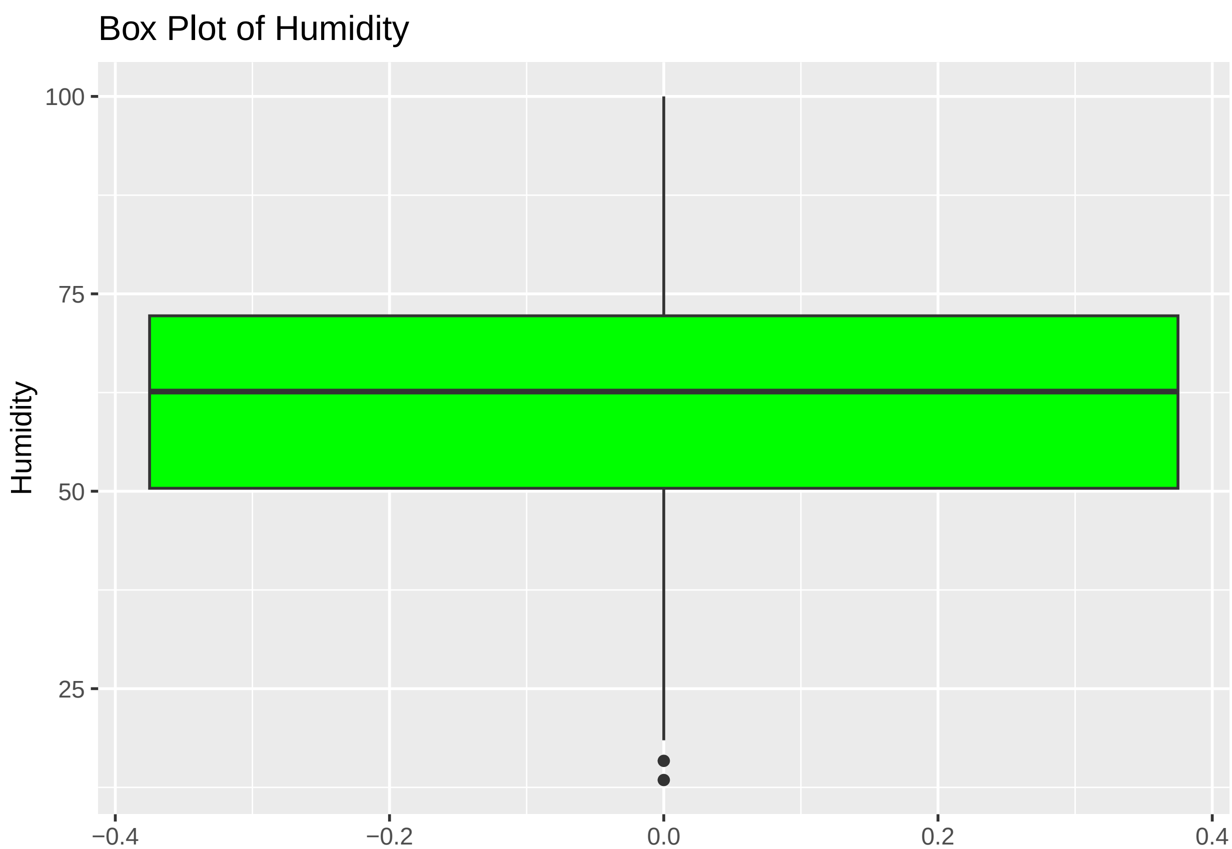


2. Boxplot of the each variable

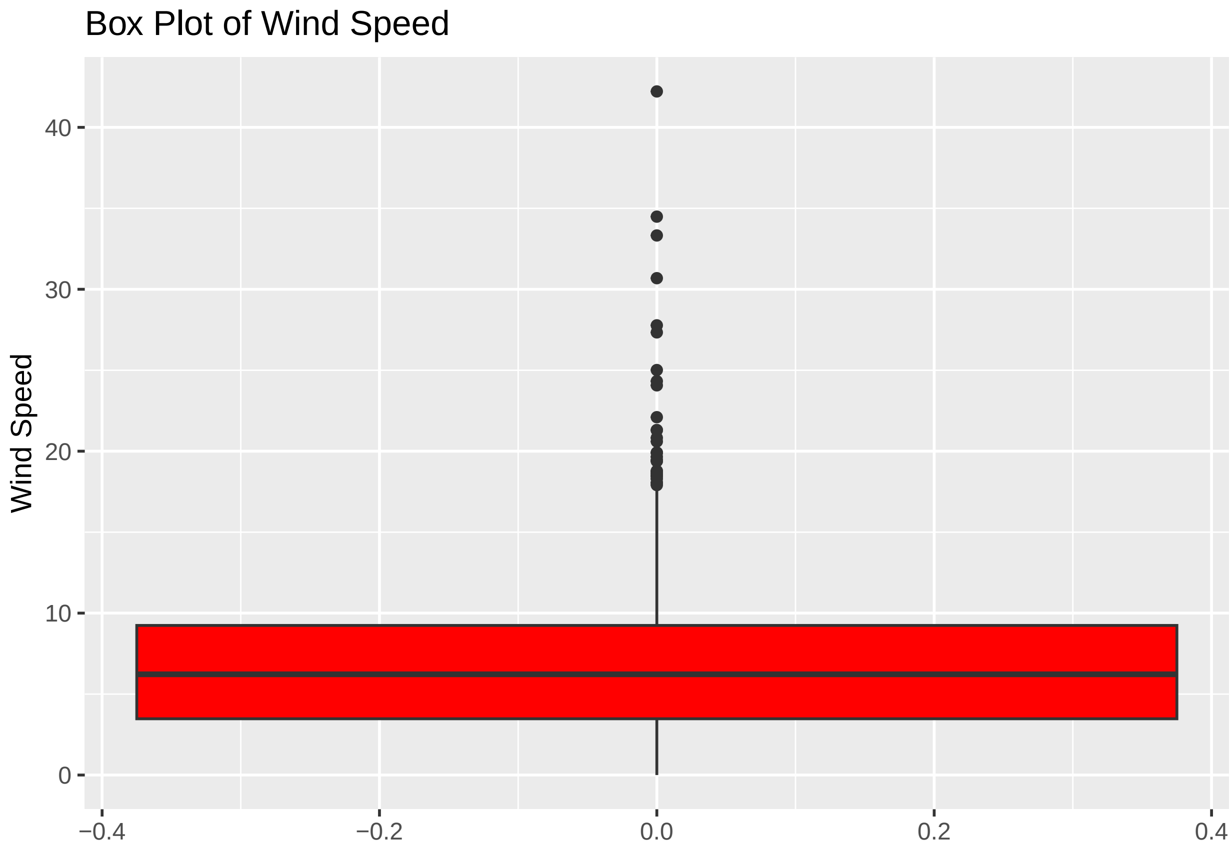
```
# Create a box plot for mean temperature
ggplot(data_train, aes(y = meantemp)) +
  geom_boxplot(fill = "blue") +
  labs(title = "Box Plot of Mean Temperature", y = "Mean Temperature")
```



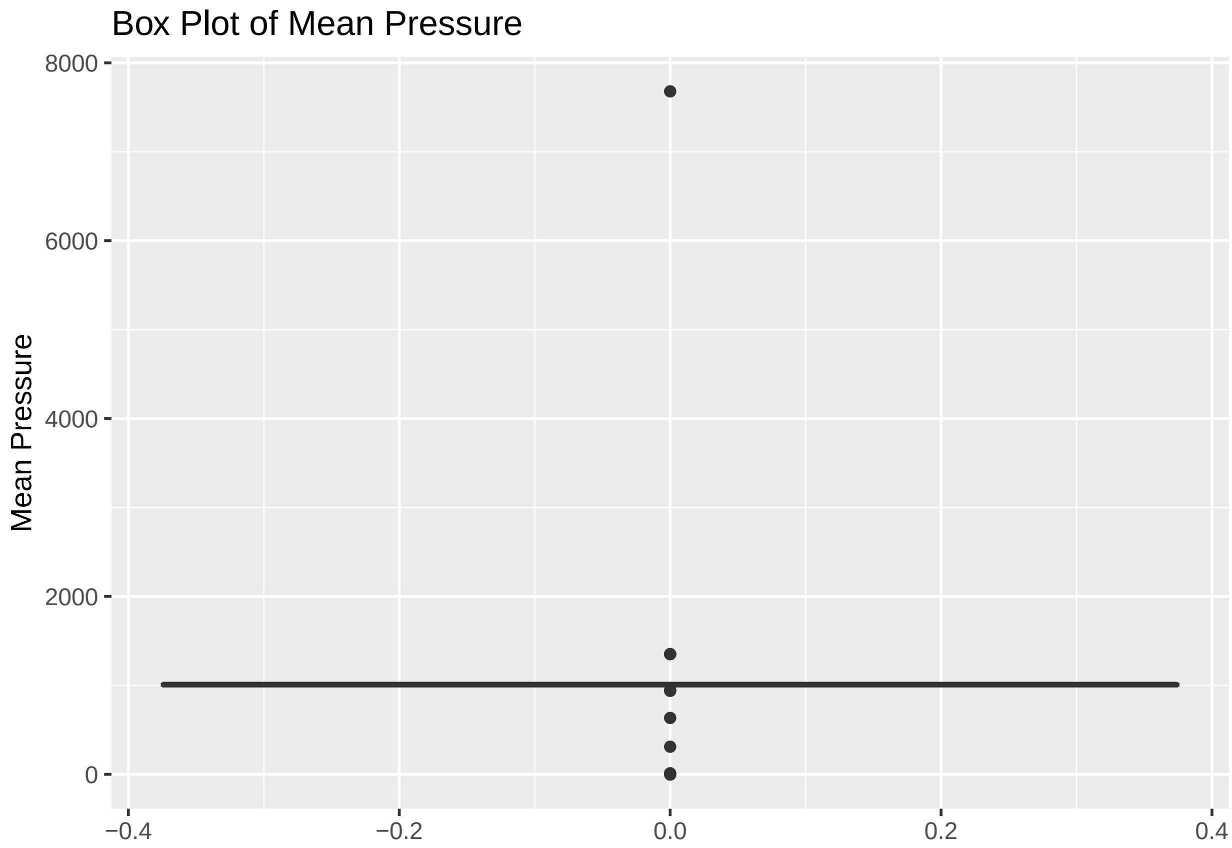

```r
# Create a box plot for humidity
ggplot(data_train, aes(y = humidity)) +
  geom_boxplot(fill = "green") +
  labs(title = "Box Plot of Humidity", y = "Humidity")
```




```r
# Create a box plot for wind speed
ggplot(data_train, aes(y = wind_speed)) +
  geom_boxplot(fill = "red") +
  labs(title = "Box Plot of Wind Speed", y = "Wind Speed")
```




```r
# Create a box plot for mean pressure
ggplot(data_train, aes(y = meanpressure)) +
  geom_boxplot(fill = "purple") +
  labs(title = "Box Plot of Mean Pressure", y = "Mean Pressure")
```



3. Find the outlier of mean pressure:

```
q1_meanpressure <- quantile(data_train$meanpressure, 0.25)
q3_meanpressure <- quantile(data_train$meanpressure, 0.75)
iqr_meanpressure <- q3_meanpressure - q1_meanpressure
lower_bound_meanpressure <- q1_meanpressure - 1.5 * iqr_meanpressure
upper_bound_meanpressure <- q3_meanpressure + 1.5 * iqr_meanpressure
outliers_meanpressure <- data_train$meanpressure[data_train$meanpressure < lower_bound_meanpressure | da
outliers_meanpressure
```

```
## [1] 7679.333333  938.066667  946.312500  310.437500  633.900000   -3.041667
## [7] 1352.615385 1350.296296   12.045455
```

We can see that there is a extreme large value of meanpressure, so we need to drop it because it is not reasonable.

4. Drop the outlier the outlier of mean pressure:

```
outlier_row_num <- which(data_train$meanpressure > 7000)
data_train_no_outlier <- data_train[-outlier_row_num, ]
```

5. Chekc missing values

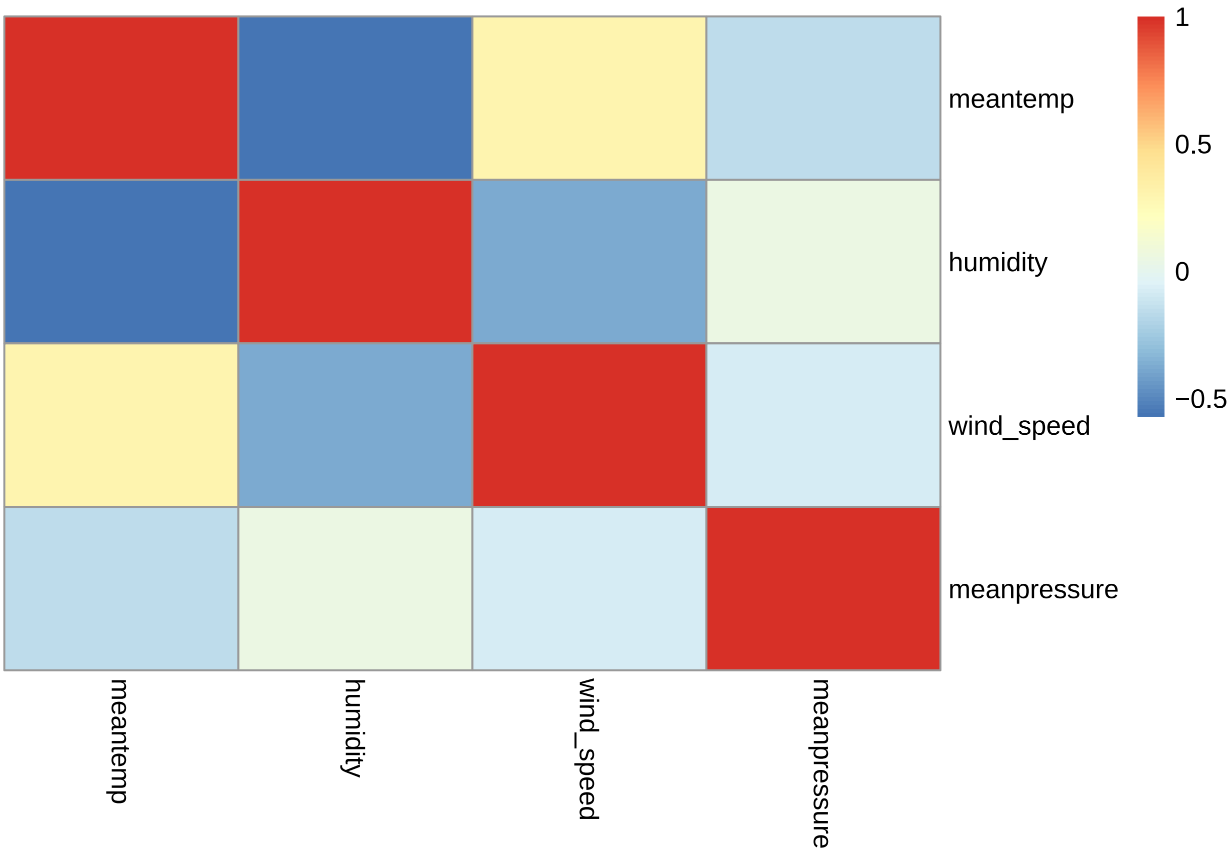
```r
sum(is.na(data_train_no_outlier))
```

```
## [1] 0
```

There is no missing values in the data set.

6. Heat map

```r
pheatmap(cor(data_train_no_outlier[,-1]),cluster_rows = FALSE, cluster_cols =FALSE)
```



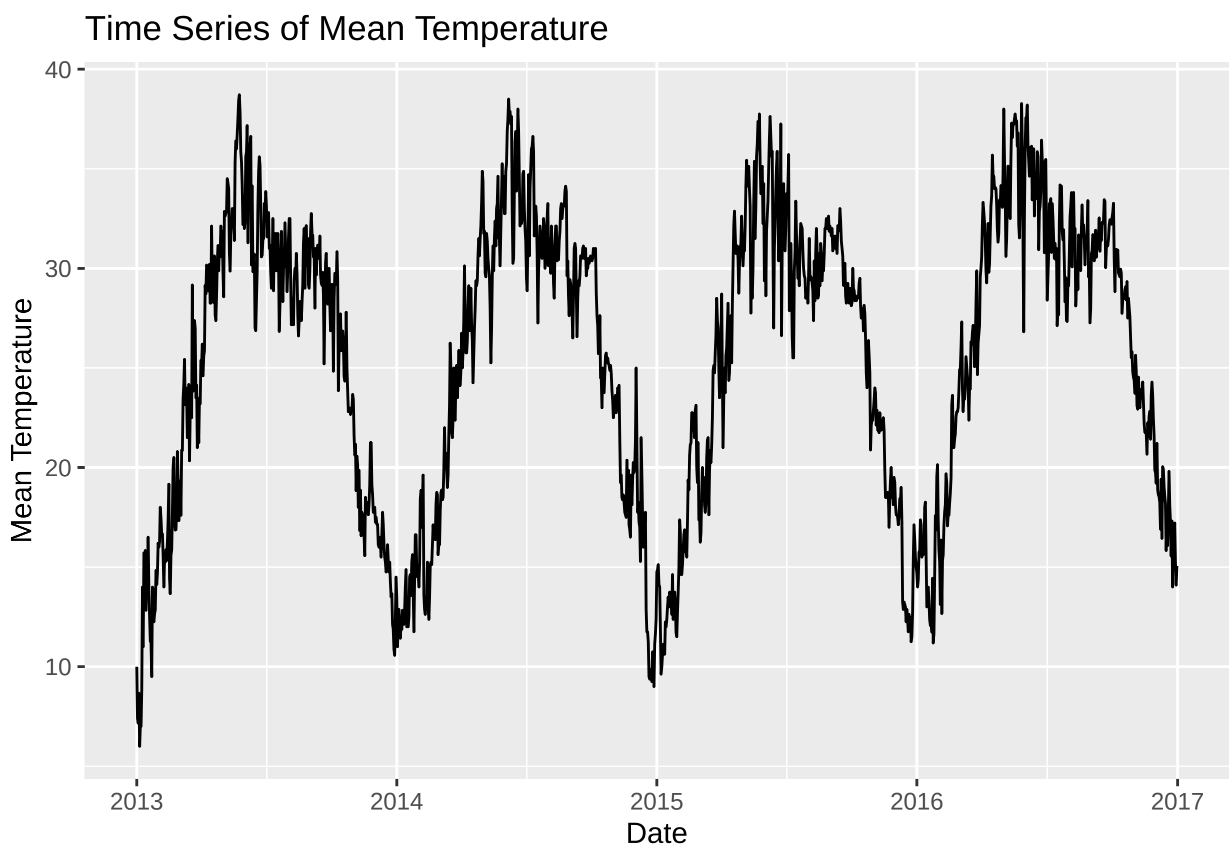
7. Transform the data to tsibble

```r
data_train_no_outlier <- data_train_no_outlier[-nrow(data_train_no_outlier), ]

train_ts <- tsibble(
  Date = as.Date(data_train_no_outlier$date),
  #Day_number = 1:length(data_train_no_outlier$date),
```

```r
  Mean_temp = data_train_no_outlier$meantemp,
  Humidity = data_train_no_outlier$humidity,
  Wind_speed = data_train_no_outlier$wind_speed,
  Mean_pressure = data_train_no_outlier$meanpressure,
  index = Date
)

test_ts <- tsibble(
  Date = as.Date(data_test$date),
  #Day_number = (length(data_train_no_outlier$date)+1):(length(data_test$date) +
    #length(data_train_no_outlier$date)),
  Mean_temp = data_test$meantemp,
  Humidity = data_test$humidity,
  Wind_speed = data_test$wind_speed,
  Mean_pressure = data_test$meanpressure,
  index = Date
)
```

8. Plot the meantemp

```r
train_ts %>%
  autoplot(Mean_temp) +
  labs(
    x = "Date",
    y = "Mean Temperature",
    title = "Time Series of Mean Temperature"
  )
```



We can see that plot shows a strong cycle.

9. Decompose the time series of meantemp

Because there is a gap in the train data set, we need to fill the gap and impute the missing values

```
train_ts_fill_gap <- train_ts %>% fill_gaps()
missing_row_num <- which(is.na(train_ts_fill_gap["Mean_temp"]))

train_ts_fill_gap[missing_row_num,]$Mean_temp <-
  data_train[missing_row_num,]$meantemp
train_ts_fill_gap[missing_row_num,]$Humidity <-
  data_train[missing_row_num,]$humidity
train_ts_fill_gap[missing_row_num,]$Wind_speed <-
  data_train[missing_row_num,]$wind_speed

pressure_imputed <- train_ts_fill_gap %>%
  model(
    ARIMA(Mean_pressure)
    ) %>% interpolate(train_ts_fill_gap)
pressure_imputed[missing_row_num,]
```

```
## # A tsibble: 1 x 2 [1D]
##   Date       Mean_pressure
##   <date>             <dbl>
## 1 2016-03-28         1009.
```

```r
train_ts_fill_gap[missing_row_num,]$Mean_pressure <-
  pressure_imputed[missing_row_num,]$Mean_pressure

train_ts_imputed <- train_ts_fill_gap
```

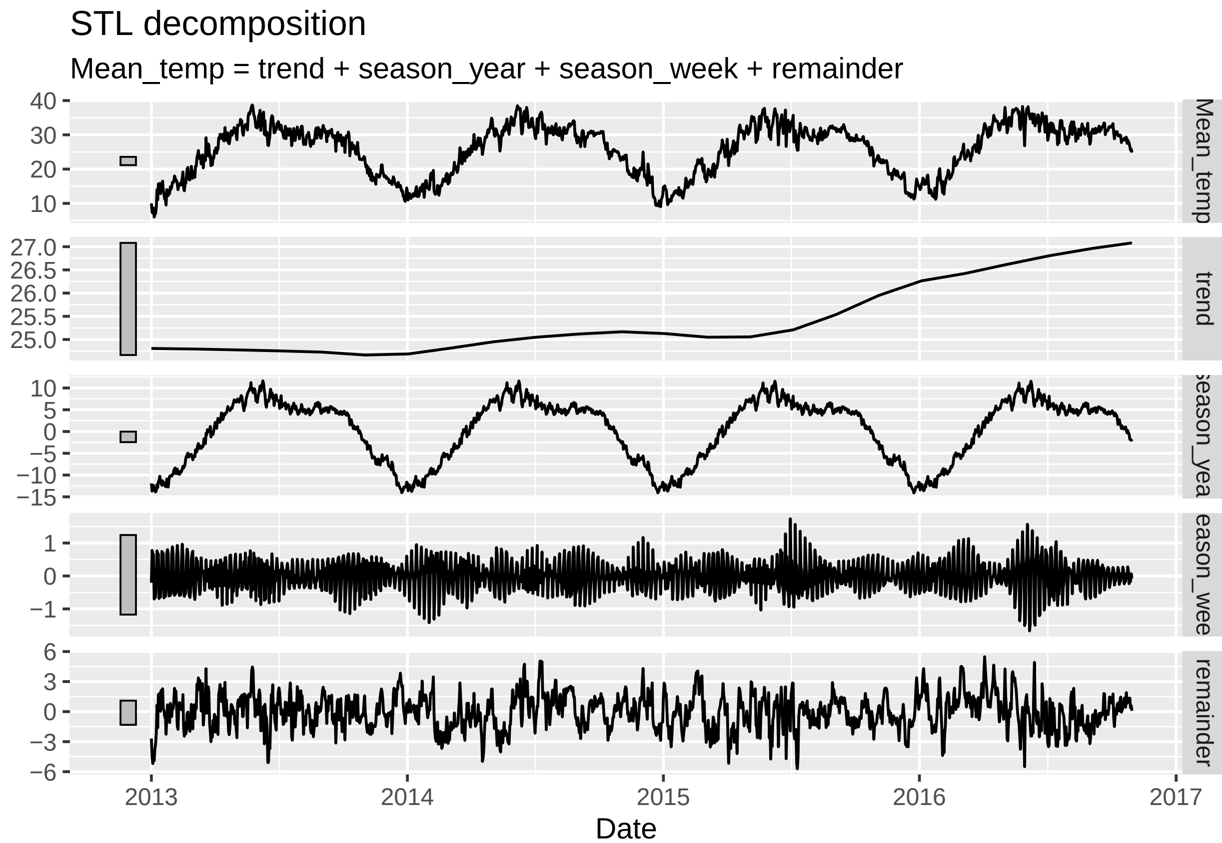
Redo the train test split, assign more number of rows to the test set

```r
data_whole <-  bind_rows(train_ts_imputed, test_ts)
train_ts_imputed <- data_whole %>% filter(Date <= "2016-10-30")
test_ts <- data_whole %>% filter(Date > "2016-10-30")
```

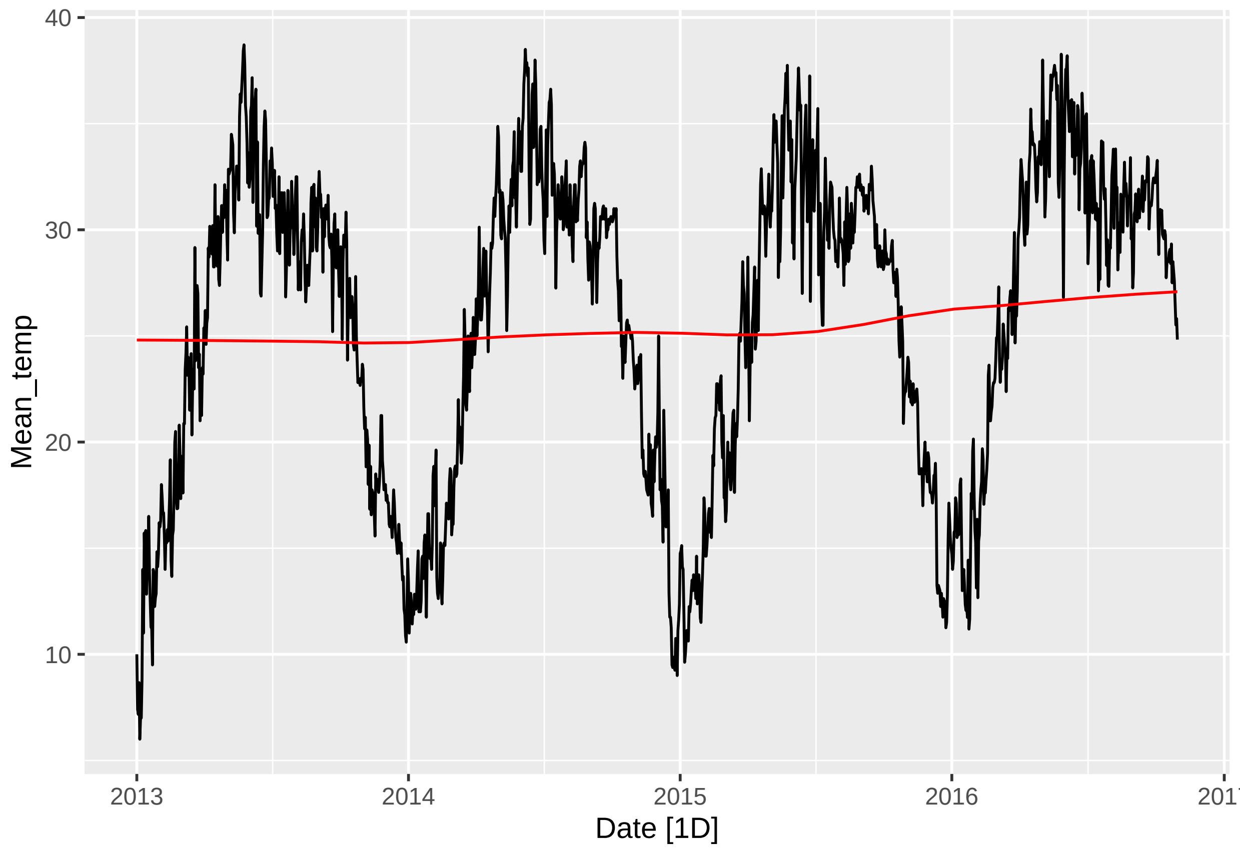
```r
dcmp_train <- train_ts_imputed %>%
  model(stl = STL(Mean_temp))
components(dcmp_train)
```

```
## # A dable: 1,399 x 8 [1D]
## # Key:     .model [1]
## # :        Mean_temp = trend + season_year + season_week + remainder
##    .model Date       Mean_temp trend season_week season_year remainder
##    <chr>  <date>         <dbl> <dbl>       <dbl>       <dbl>     <dbl>
##  1 stl    2013-01-01     10     24.8      -0.205       -11.9    -2.69
##  2 stl    2013-01-02      7.4   24.8       0.778       -13.7    -4.48
##  3 stl    2013-01-03      7.17  24.8       0.675       -13.1    -5.20
##  4 stl    2013-01-04      8.67  24.8       0.657       -12.4    -4.44
##  5 stl    2013-01-05      6     24.8      -0.701       -13.2    -4.91
##  6 stl    2013-01-06      7     24.8      -0.535       -13.5    -3.75
##  7 stl    2013-01-07      7     24.8      -0.649       -13.9    -3.27
##  8 stl    2013-01-08      8.86  24.8      -0.231       -13.6    -2.12
##  9 stl    2013-01-09     14     24.8       0.761       -12.2     0.641
## 10 stl    2013-01-10     11     24.8       0.693       -12.9    -1.62
## # i 1,389 more rows
## # i 1 more variable: season_adjust <dbl>
```

```r
components(dcmp_train) %>%
  autoplot
```




```r
train_ts_imputed %>%
  autoplot(Mean_temp) +
  autolayer(components(dcmp_train), trend, color='red')
```




```
  labs(
    x = "Date",
    y = "Mean Temperature",
    title = "Time Series of Mean Temperature"
  )
```

```
## $x
## [1] "Date"
##
## $y
## [1] "Mean Temperature"
##
## $title
## [1] "Time Series of Mean Temperature"
##
## attr(,"class")
## [1] "labels"
```

II. Training model

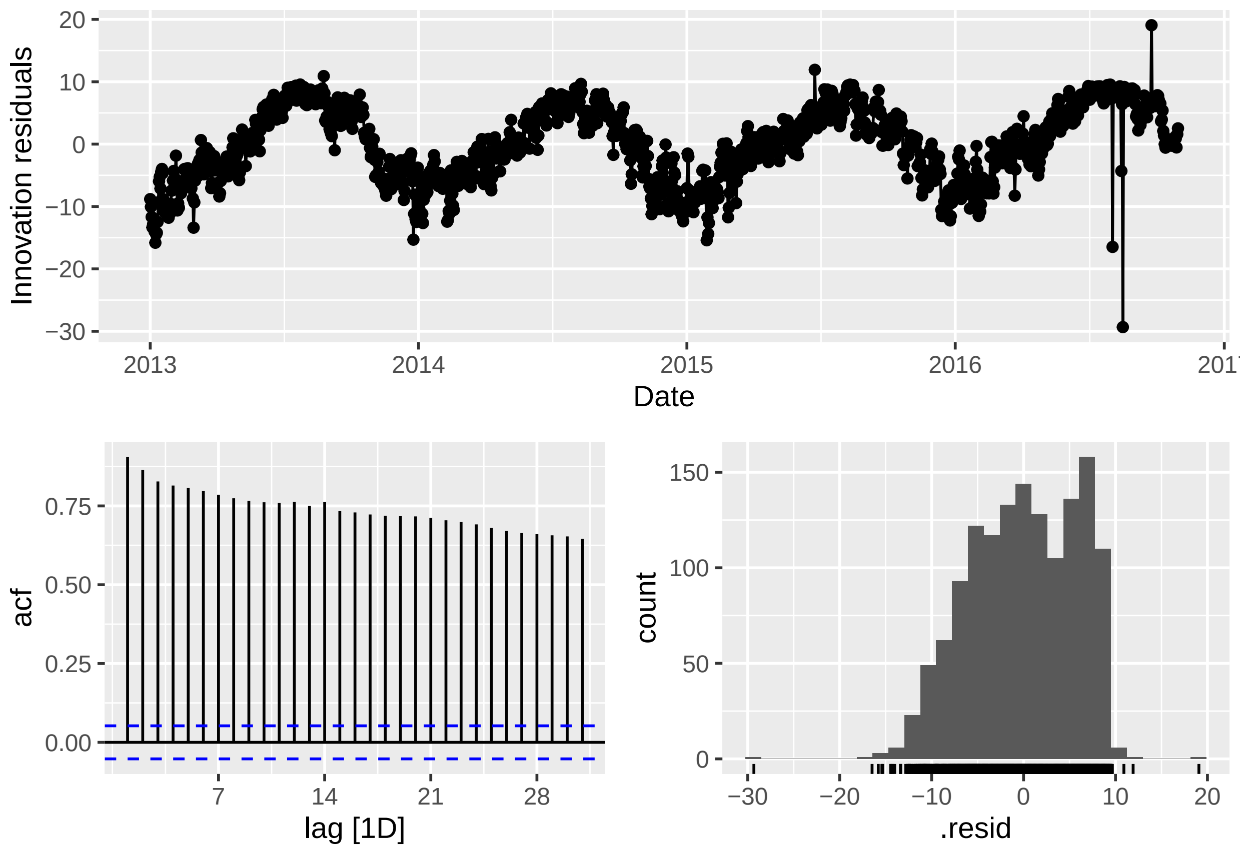
1. TSLM

```r
fit_lm <- train_ts_imputed %>%
  model(
    tslm = TSLM(Mean_temp ~ Mean_pressure + Wind_speed + Humidity)
  )
report(fit_lm)
```

```
## Series: Mean_temp
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.3331  -4.4425   0.1376   5.1594  19.0682
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    73.742841   4.439086  16.612  < 2e-16 ***
## Mean_pressure -0.034708   0.004332  -8.012 2.36e-15 ***
## Wind_speed     0.155739   0.037008   4.208 2.74e-05 ***
## Humidity      -0.232865   0.010008 -23.268  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.866 on 1395 degrees of freedom
## Multiple R-squared: 0.371,   Adjusted R-squared: 0.3697
## F-statistic: 274.3 on 3 and 1395 DF, p-value: < 2.22e-16
```

Residual:

```r
gg_tsresiduals(fit_lm)
```



We can see there is an obvious pattern in the residual plot, and the acf plot shows strong autocorrelation between observations. The distribution is also not normally distributed. In other words, there are some useful information left in the residuals.

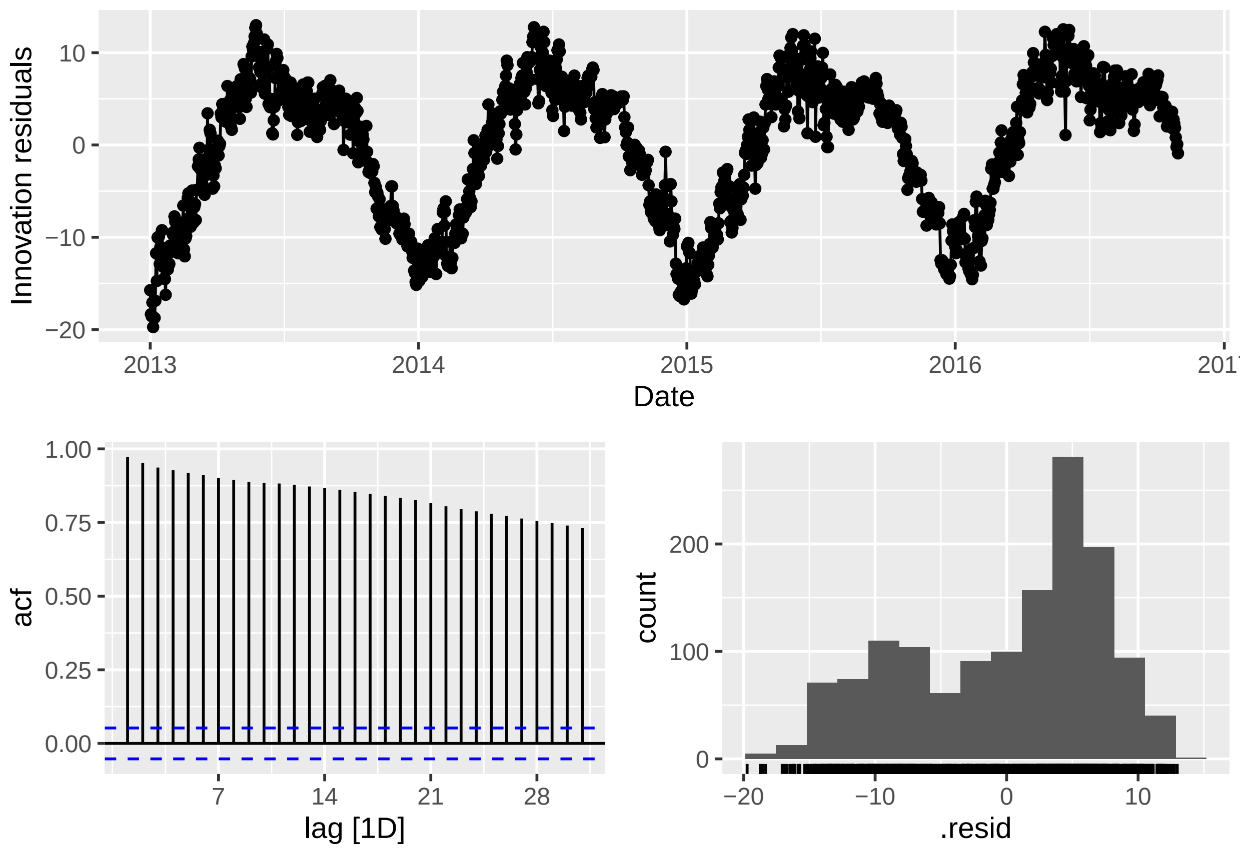
2. Benchmark methods(Mean, Naive, Seanonal Naive)

Mean method:

```
fit_mean <- train_ts_imputed %>%
  model(
    mean = MEAN(Mean_temp)
  )
```

Residuals:

```
gg_tsresiduals(fit_mean)
```



Similar to TSLM.

NAIVE method:

```
fit_naive <- train_ts_imputed %>%
  model(
    naive = NAIVE(Mean_temp)
  )
```

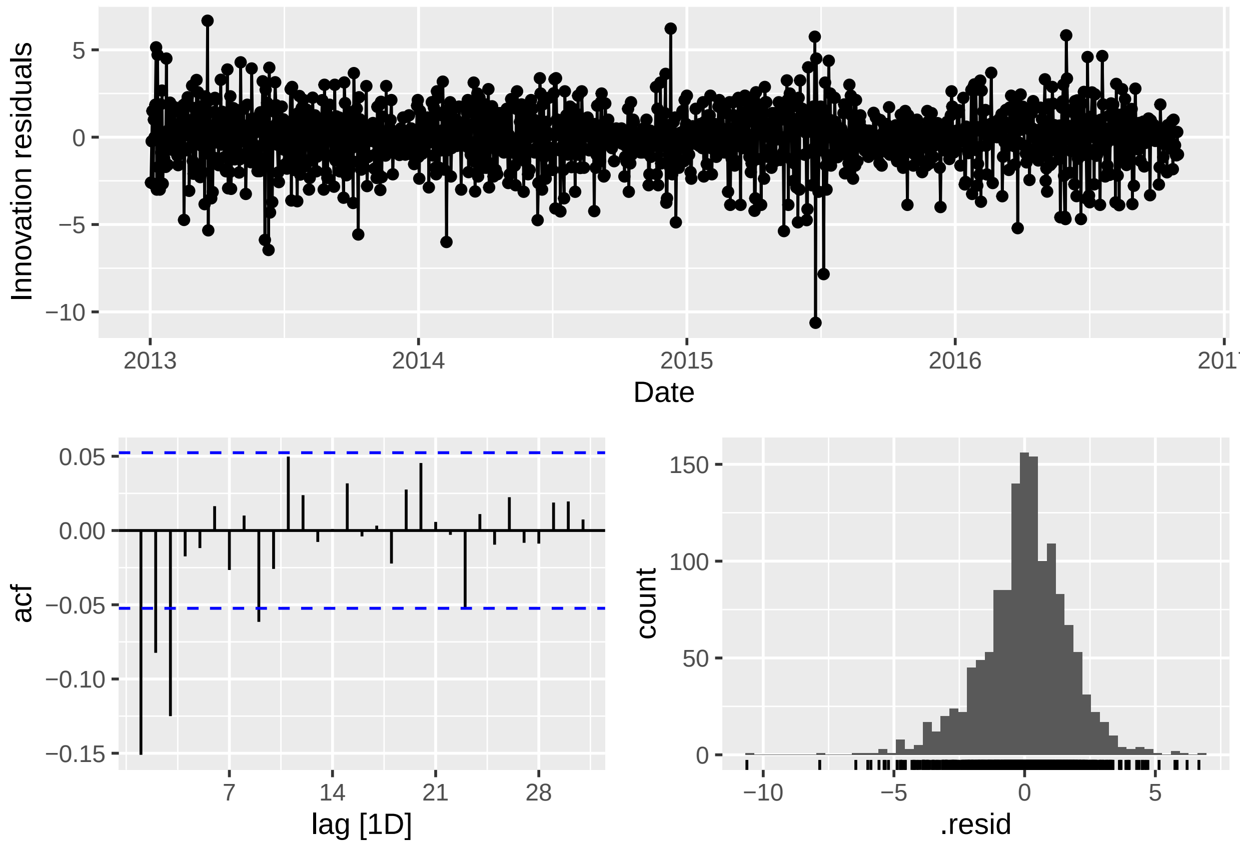
Residuals:

```
gg_tsresiduals(fit_naive)
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```



Better than mean method, let's the result of Ljung-Box test:

```
augment(fit_naive) %>%
  features(.resid, ljung_box, lag=10, dof=0)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 naive     71.8  1.98e-11
```

We got an extreme small p-value, so the residuals are not white noise.

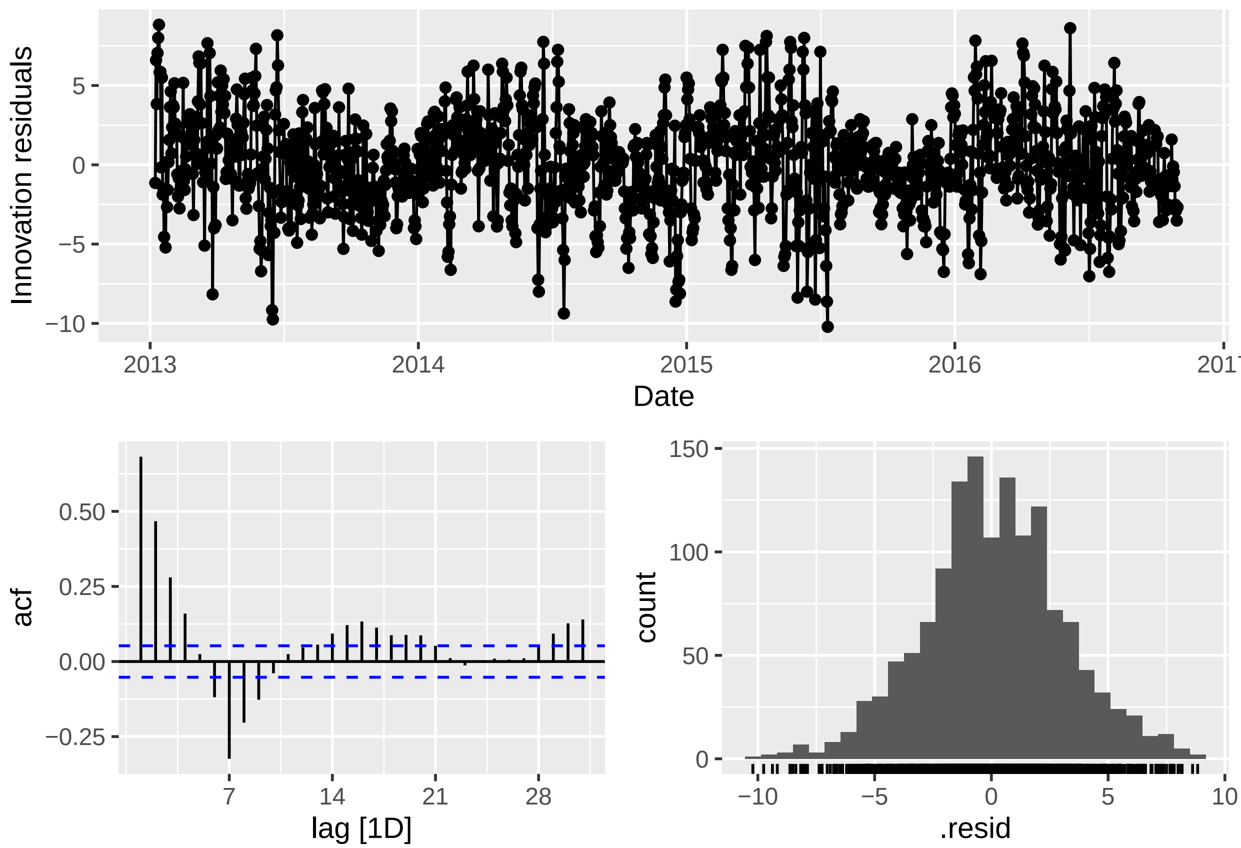
Snaive method:

```
fit_snaive <- train_ts_imputed %>%
  model(
    snaive = SNAIVE(Mean_temp)
  )
```

Residuals:

```r
gg_tsresiduals(fit_snaive)
```

```
## Warning: Removed 7 rows containing missing values (`geom_line()`).

## Warning: Removed 7 rows containing missing values (`geom_point()`).

## Warning: Removed 7 rows containing non-finite values (`stat_bin()`).
```



Drift method:

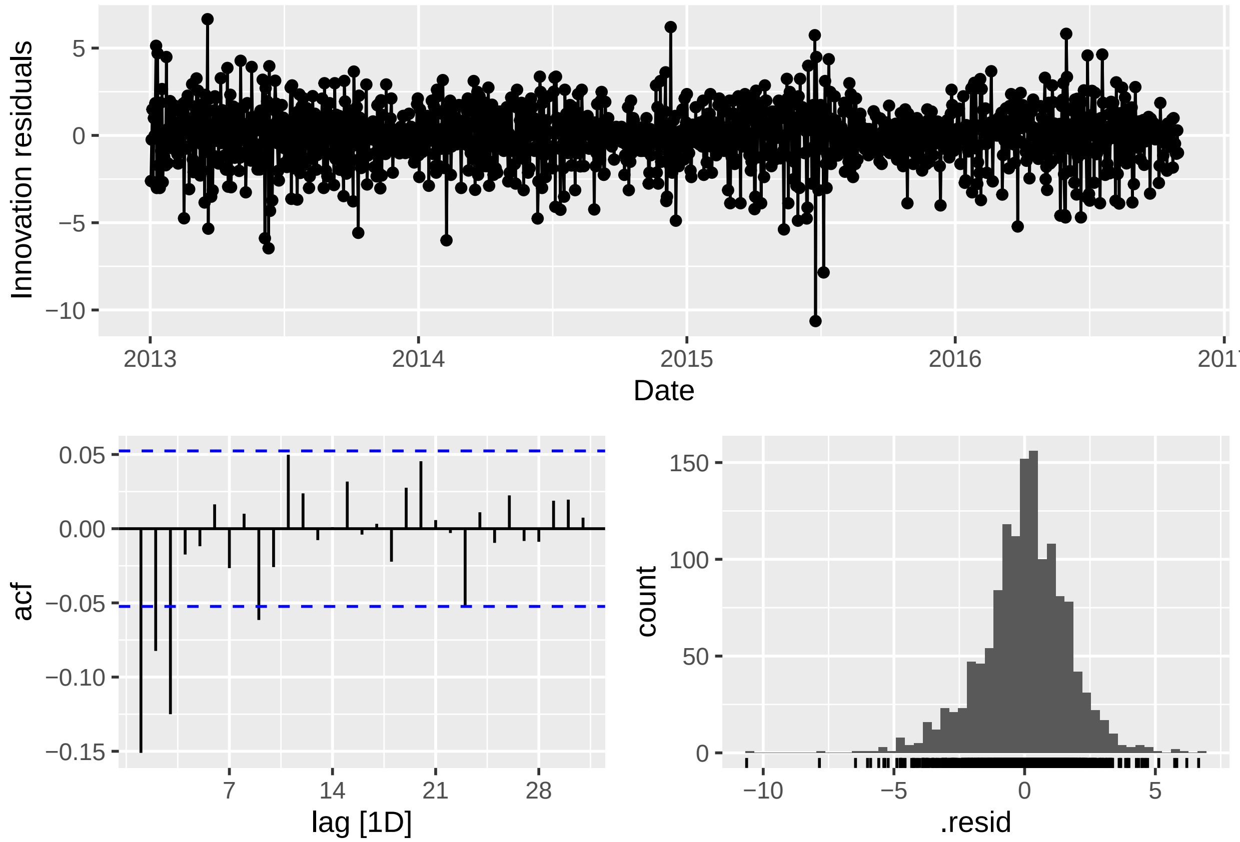
```r
fit_drift <- train_ts_imputed %>%
  model(
    drift = NAIVE(Mean_temp ~ drift())
  )
```

Residual:

```r
gg_tsresiduals(fit_drift)
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```



Similar to naive

3. ARIMA

With a constant

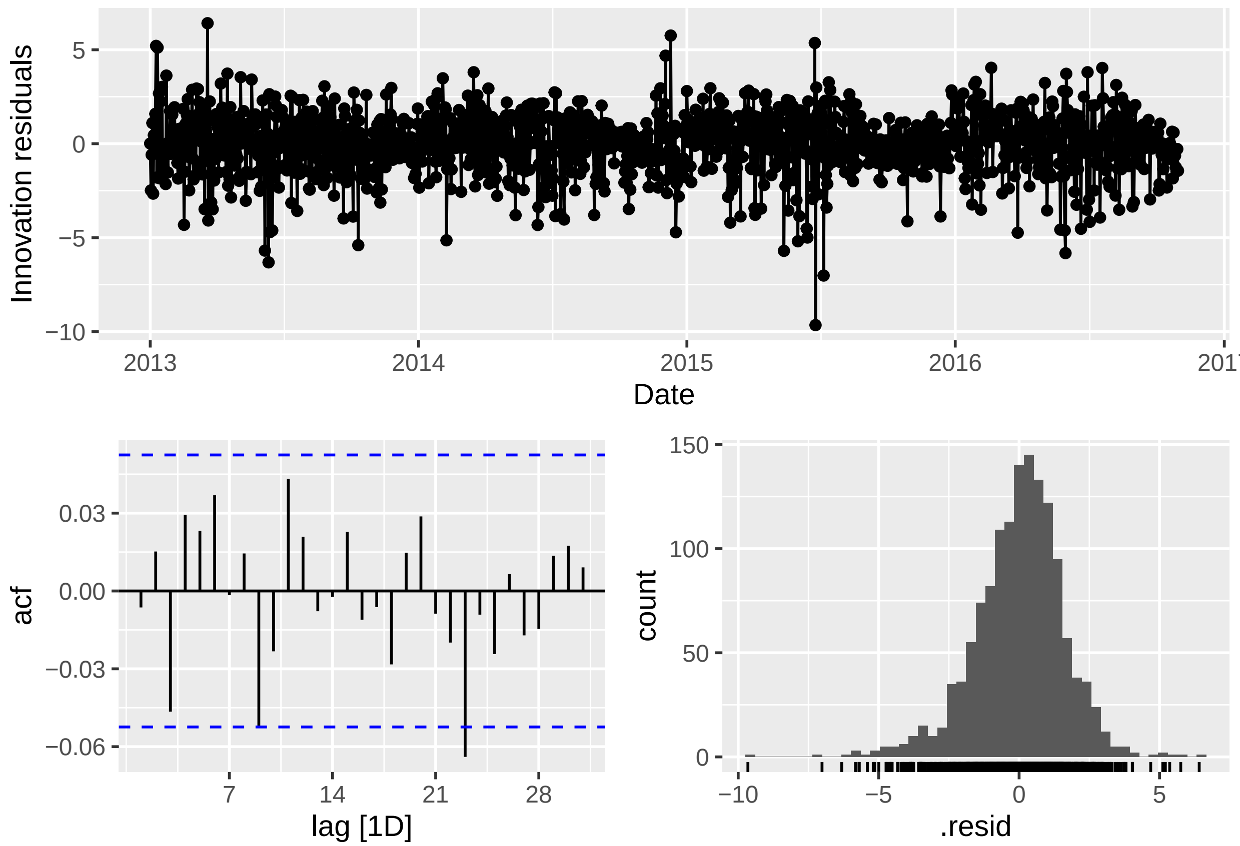
```
fit_arima <- train_ts_imputed %>%
  model(
    arima = ARIMA(Mean_temp ~ 1 + pdq() + PDQ(), stepwise=FALSE,approx=FALSE)
  )
report(fit_arima)
```

```
## Series: Mean_temp
## Model: ARIMA(2,1,2)(0,0,1)[7] w/ drift
##
## Coefficients:
##          ar1      ar2      ma1     ma2     sma1  constant
##       1.6921  -0.6991  -1.9202  0.9267  -0.0074     1e-04
## s.e.  0.0319   0.0317   0.0179  0.0177   0.0282     3e-04
##
```

```
## sigma^2 estimated as 2.56:  log likelihood=-2638.14
## AIC=5290.28   AICc=5290.36   BIC=5326.98
```

Residuals:

```
gg_tsresiduals(fit_arima)
```



Seems good, let's see the Ljung-box test:

```
augment(fit_arima) %>%
  features(.resid, ljung_box, lag=10, dof=4)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 arima     12.2    0.0580
```

The p-value not large but greater than 0.05, so the residuals are dishtinguishable for white noise.

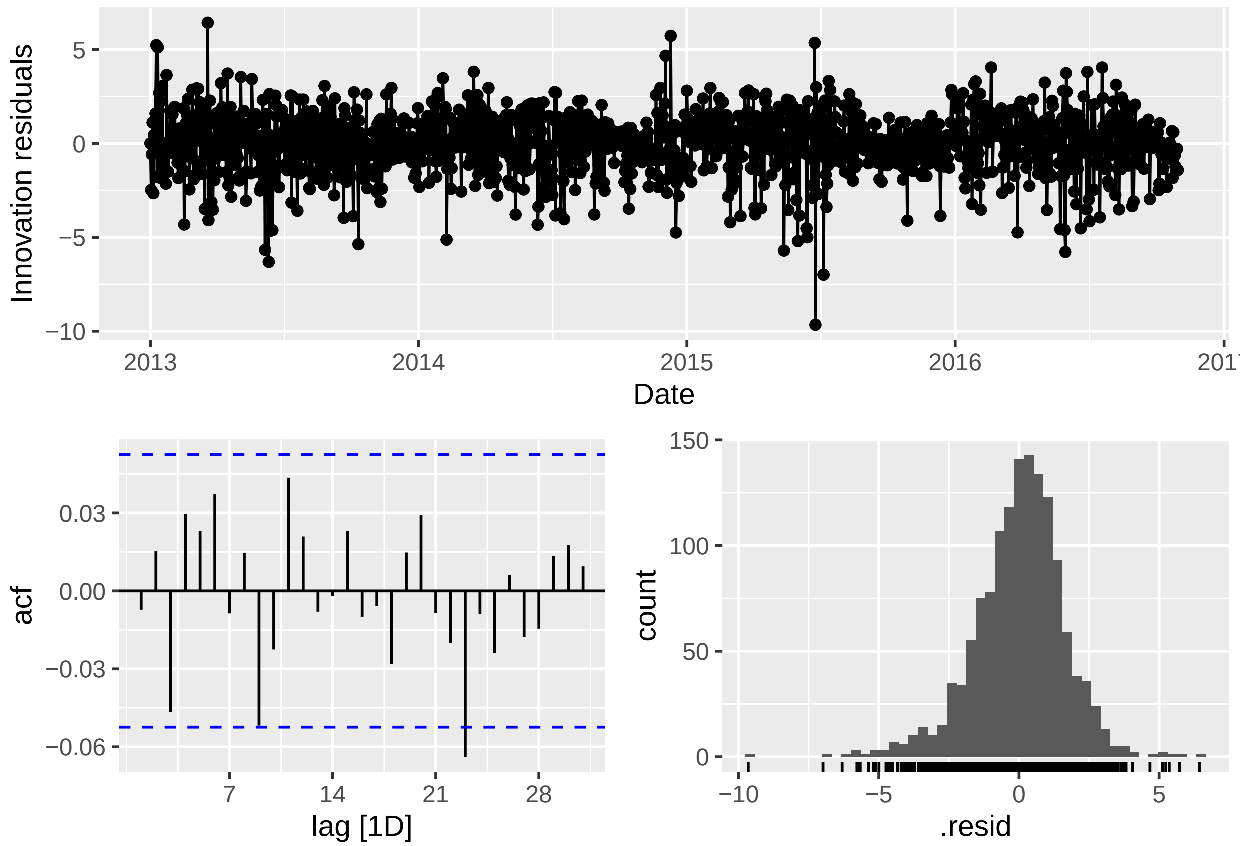
Without a constant

```r
fit_arima_no_cons <- train_ts_imputed %>%
  model(
    arima_no_cons = ARIMA(Mean_temp ~ pdq() + PDQ(),
                          stepwise=FALSE, approx=FALSE)
  )
report(fit_arima_no_cons )
```

```
## Series: Mean_temp
## Model: ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.6933  -0.7003  -1.9209  0.9274
## s.e.  0.0315   0.0312   0.0173  0.0172
##
## sigma^2 estimated as 2.556:  log likelihood=-2638.21
## AIC=5286.43   AICc=5286.47   BIC=5312.64
```

Residuals:

```r
gg_tsresiduals(fit_arima_no_cons)
```



Also seems good, let's see the Ljung-box test:

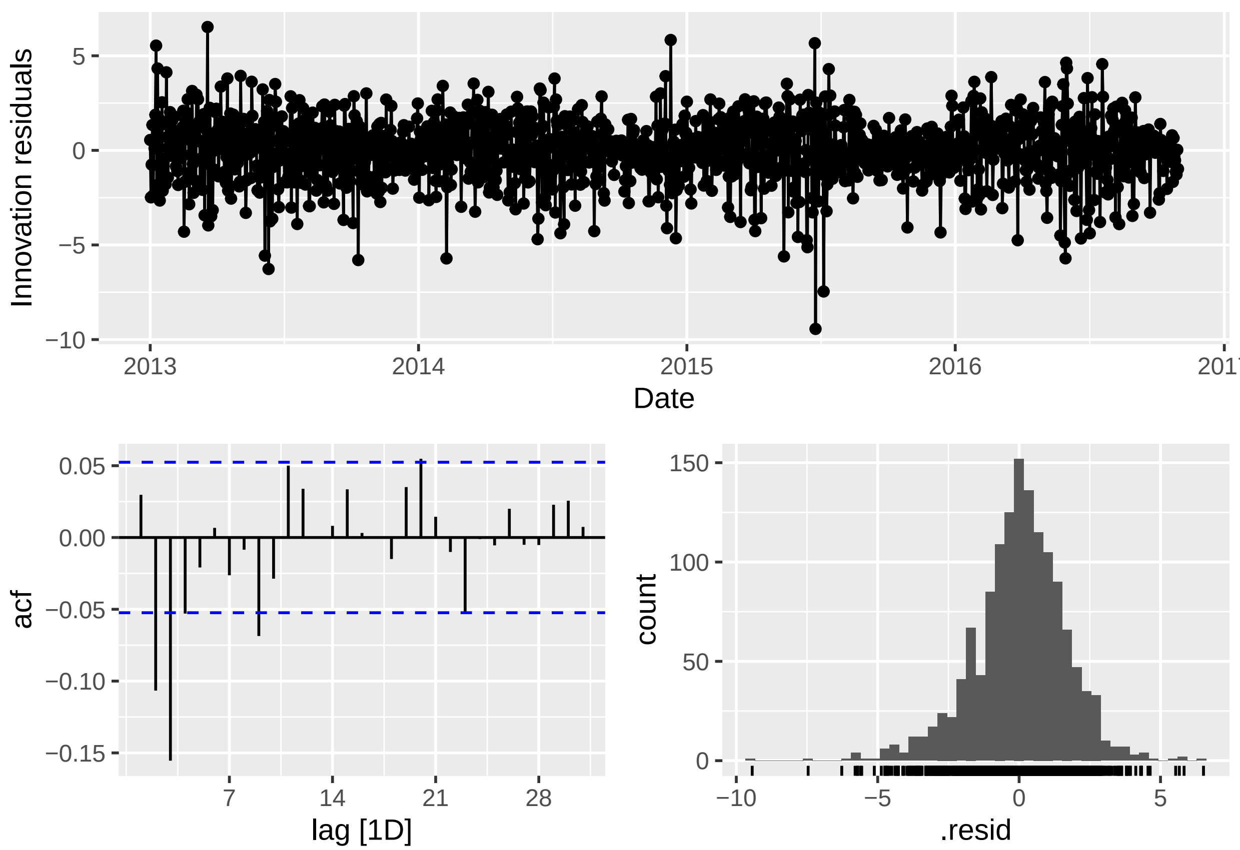
```r
augment(fit_arima_no_cons) %>%
  features(.resid, ljung_box, lag=10, dof=4)
```

```
## # A tibble: 1 x 3
##   .model        lb_stat lb_pvalue
##   <chr>           <dbl>     <dbl>
## 1 arima_no_cons    12.3    0.0560
```

Similar to the ARIMA with a constant.

4.EWMA

```r
fit_ewma <- train_ts_imputed |>
  model(ewma = ETS(Mean_temp))
gg_tsresiduals(fit_ewma)
```



Check white noise by ljung box:

```r
augment(fit_ewma) %>%
  features(.resid, ljung_box, lag=10, dof=0)
```

```
## # A tibble: 1 x 3
```

```
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 ewma      64.6  4.84e-10
```

The Ha is rejected by low p-value, thus the residual of EWMA model is not white noise here.

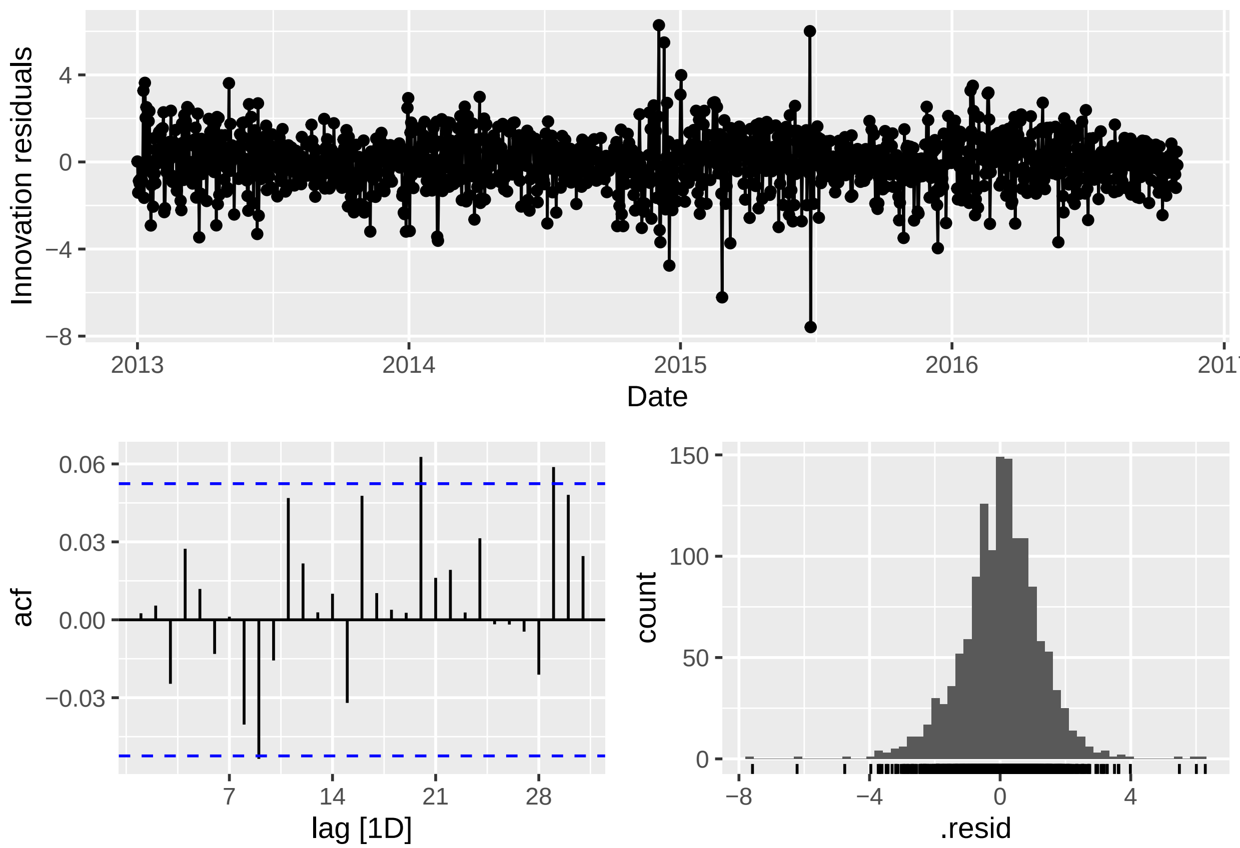
5.Dynamic regression

```r
fit_darima <- train_ts_imputed%>%
  model(
    dynamic_arima = ARIMA(Mean_temp ~ Mean_pressure + Humidity + Wind_speed)
  )
report(fit_darima)
```

```
## Series: Mean_temp
## Model: LM w/ ARIMA(3,1,1)(1,0,1)[7] errors
##
## Coefficients:
##          ar1      ar2      ar3      ma1     sar1    sma1  Mean_pressure
##       0.6289  -0.1344  -0.0684  -0.6946  -0.2744  0.2561          1e-04
## s.e.  0.0930   0.0321   0.0398   0.0899   0.6401  0.6457          7e-04
##       Humidity  Wind_speed
##        -0.1407     -0.0289
## s.e.    0.0043      0.0071
##
## sigma^2 estimated as 1.482:  log likelihood=-2254.32
## AIC=4528.64   AICc=4528.8   BIC=4581.07
```

Residuals:

```r
gg_tsresiduals(fit_darima, type = 'innovation')
```



Ljung-box:

```
augment(fit_darima) %>%
  features(.resid, ljung_box, lag=10, dof=7)
```

```
## # A tibble: 1 x 3
##   .model        lb_stat lb_pvalue
##   <chr>           <dbl>     <dbl>
## 1 dynamic_arima    9.08    0.0283
```

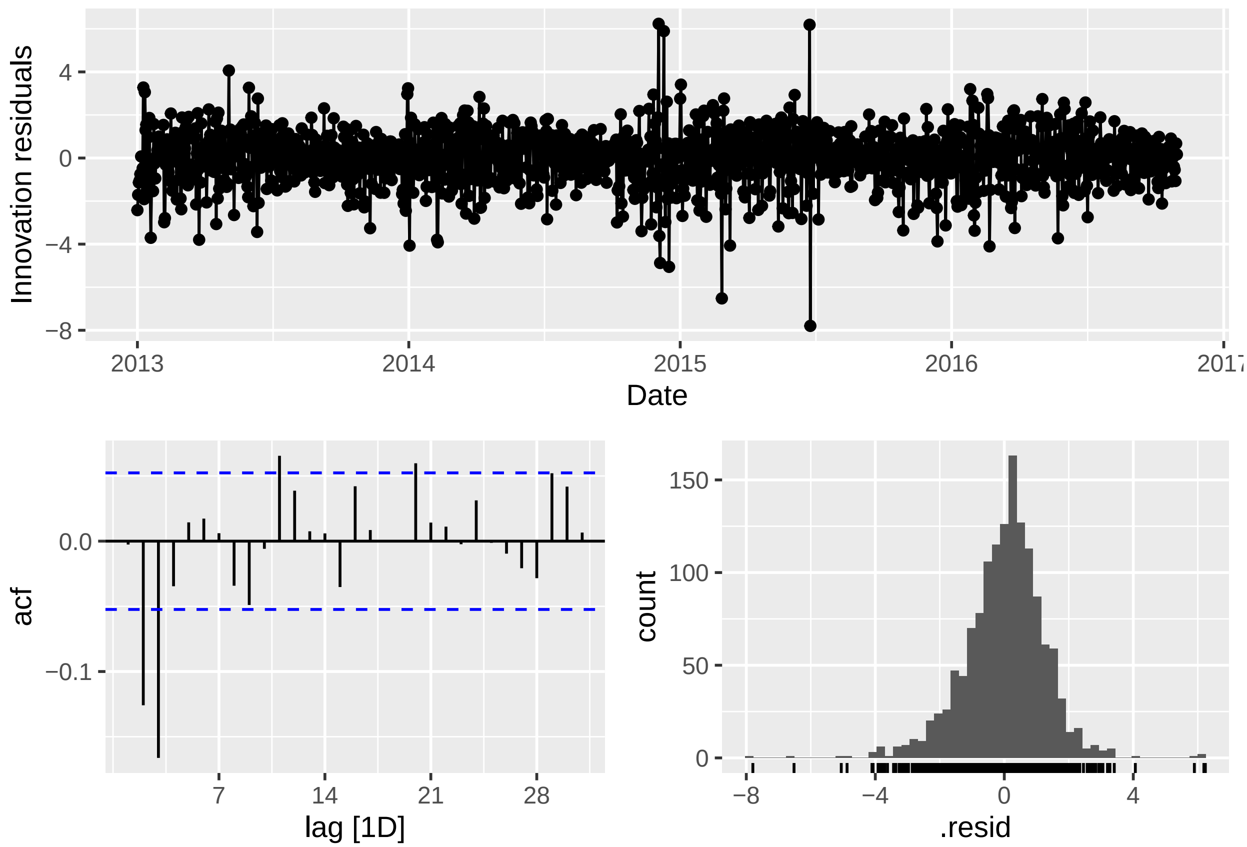
Arima error with differnece is 0

```
fit_darima_diff0 <- train_ts_imputed %>%
  model(
    dynamic_arima_diff0 = ARIMA(Mean_temp ~ Mean_pressure + Humidity + Wind_speed +
                        pdq(d = 0) + PDQ())
  )
report(fit_darima_diff0)
```

```
## Series: Mean_temp
## Model: LM w/ ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1  Mean_pressure  Humidity  Wind_speed  intercept
```

```
##        0.9811          0e+00    -0.1381      -0.0302    34.1278
## s.e.  0.0051          7e-04     0.0042       0.0071     1.8552
##
## sigma^2 estimated as 1.567:  log likelihood=-2298.58
## AIC=4609.16   AICc=4609.22   BIC=4640.62
```

```
gg_tsresiduals(fit_darima_diff0)
```



Ljung-box:

```
augment(fit_darima_diff0) %>%
  features(.resid, ljung_box, lag=10, dof=5)
```

```
## # A tibble: 1 x 3
##   .model              lb_stat lb_pvalue
##   <chr>                 <dbl>     <dbl>
## 1 dynamic_arima_diff0    68.6  2.03e-13
```

The residual is neither white noise.

6. NNAR

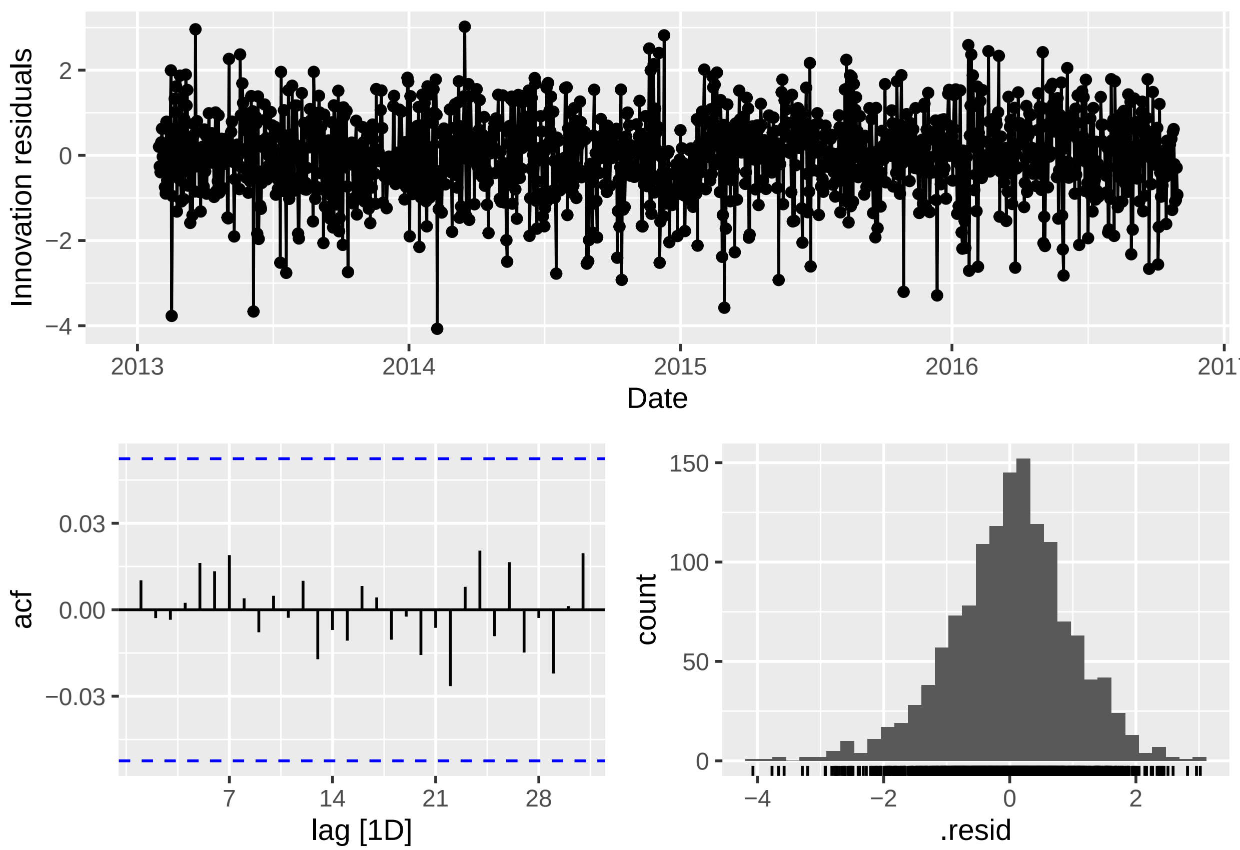
```r
fit_nnar <- train_ts_imputed %>%
  model(
    nnar = NNETAR(Mean_temp)
  )
report(fit_nnar)
```

```
## Series: Mean_temp
## Model: NNAR(29,1,15)[7]
##
## Average of 20 networks, each of which is
## a 29-15-1 network with 466 weights
## options were - linear output units
##
## sigma^2 estimated as 0.9295
```

Residuals:

```r
gg_tsresiduals(fit_nnar)
```

```
## Warning: Removed 29 rows containing missing values (`geom_line()`).

## Warning: Removed 29 rows containing missing values (`geom_point()`).

## Warning: Removed 29 rows containing non-finite values (`stat_bin()`).
```



ljung-box:

```r
augment(fit_nnar) %>%
  features(.resid, ljung_box, lag=10, dof=0)
```

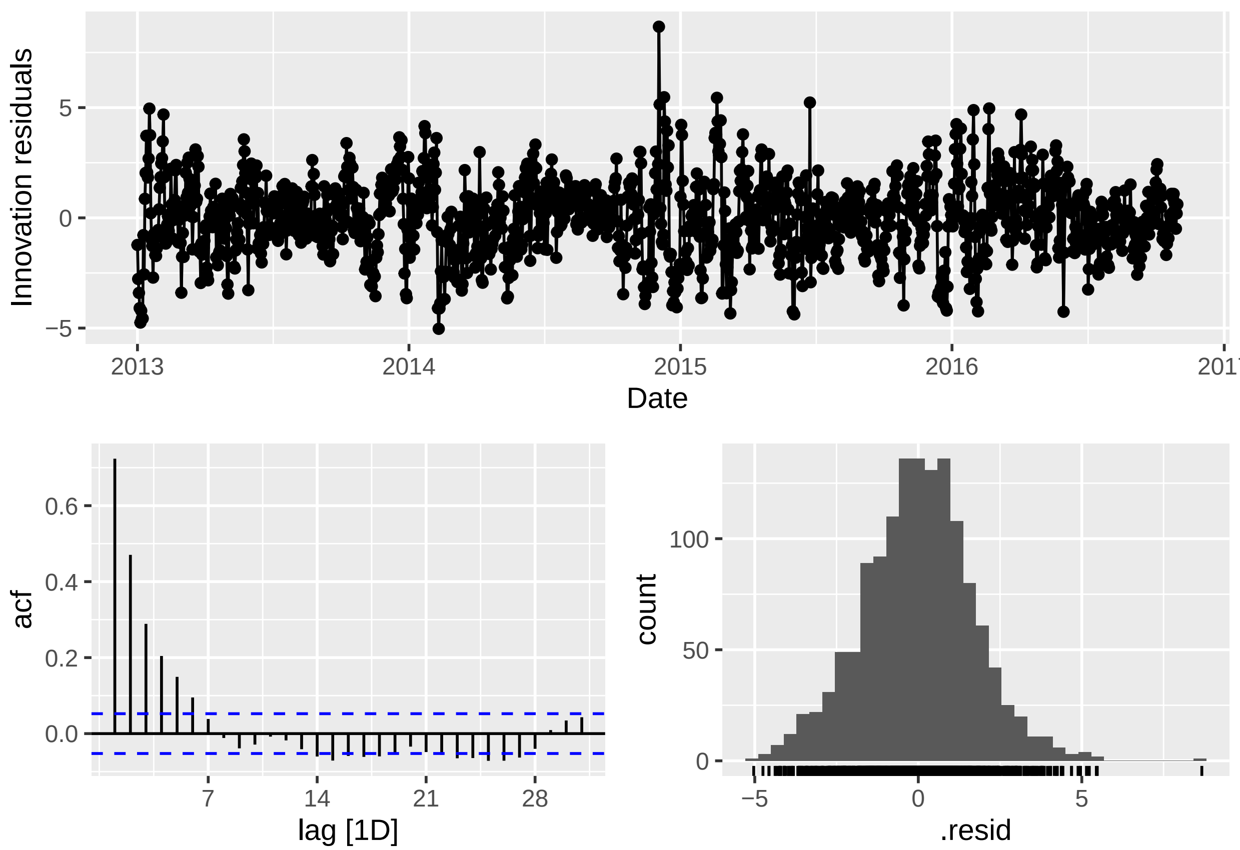
```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 nnar      1.42     0.999
```

7. Propht

```r
fit_prophet <- train_ts_imputed %>%
  model(
    prophet = prophet(Mean_temp ~ Mean_pressure
                      + Humidity + Wind_speed
                      + season(period = "week", order = 10)
                      + season(period = "year", order = 5))
  )
```

Residuals:

```r
gg_tsresiduals(fit_prophet)
```



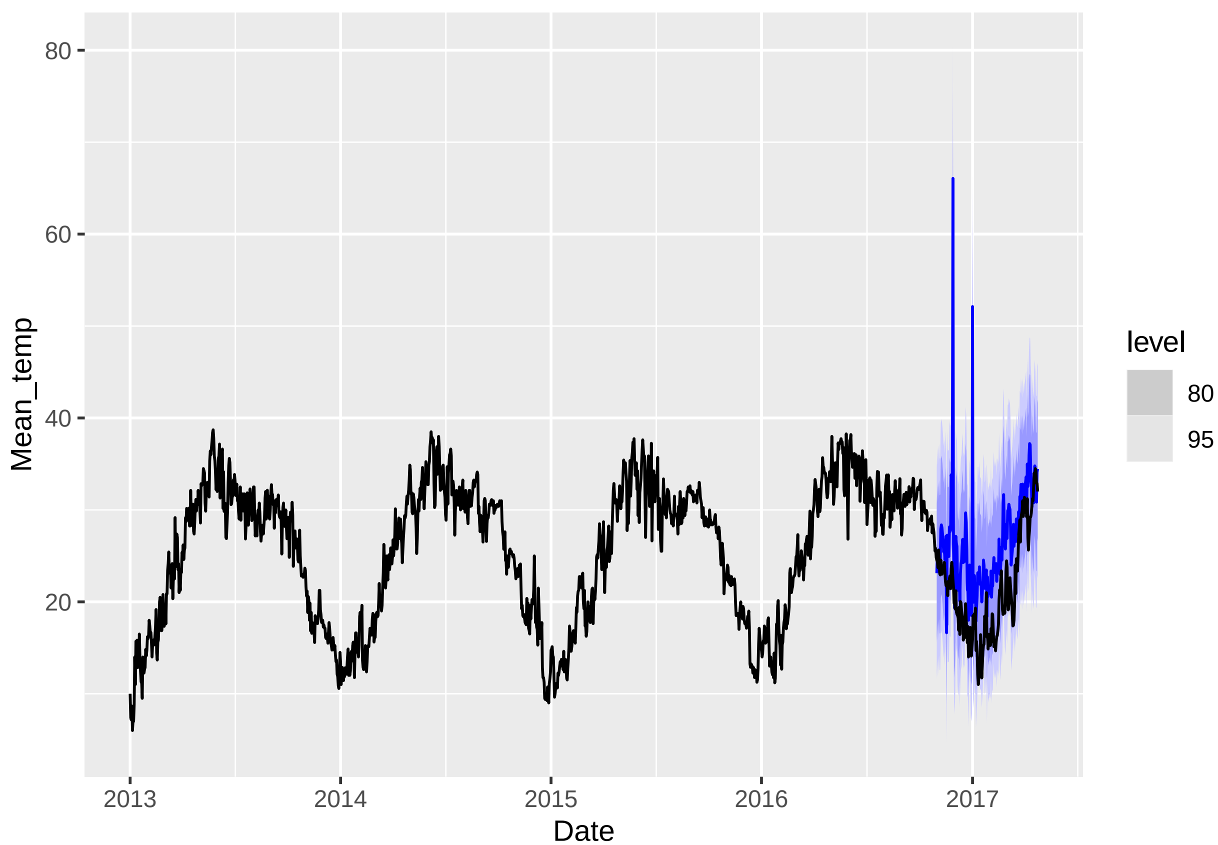
III. Evaluation of the models

1. TSLM

```r
tslm_forecast <- fit_lm %>% forecast(test_ts)
tslm_acc <- fabletools::accuracy(tslm_forecast, test_ts)
tslm_acc
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 tslm   Test  -5.27  7.24  5.46 -28.9  29.6   NaN   NaN 0.337
```

```r
# combine train and test tsibble
combined_tsibble <- bind_rows(train_ts_imputed, test_ts)
tslm_forecast %>%
   autoplot(combined_tsibble)
```



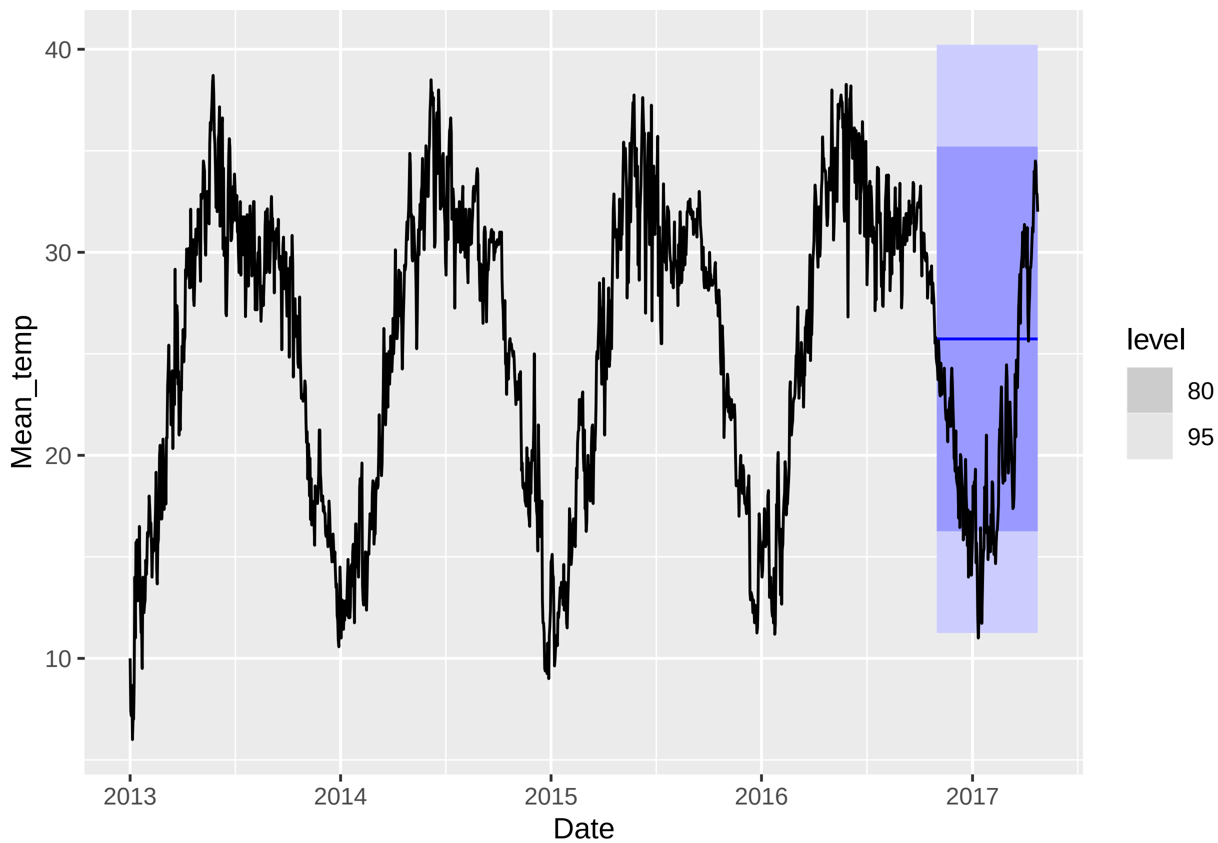
2. Benckmark

Mean

```r
mean_forecast <- fit_mean %>% forecast(test_ts)
mean_acc <- fabletools::accuracy(mean_forecast, test_ts)
mean_acc
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 mean   Test  -4.51  7.07  6.25 -29.2  34.8   NaN   NaN 0.945
```

```r
mean_forecast %>%
  autoplot(combined_tsibble)
```

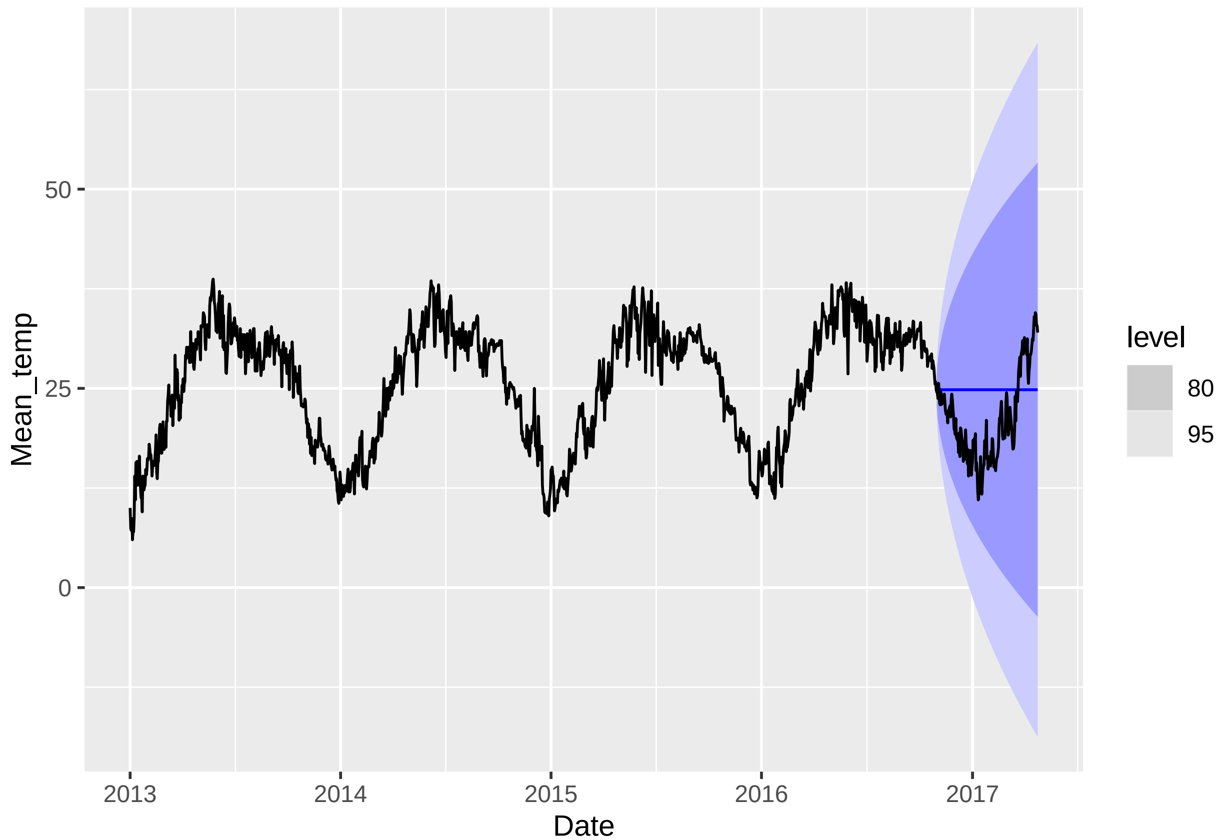


NAIVE

```r
naive_forecast <- fit_naive %>% forecast(test_ts)
naive_acc <- fabletools::accuracy(naive_forecast, test_ts)
naive_acc
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 naive  Test  -3.60  6.53  5.70 -24.6  31.4   NaN   NaN 0.945
```

```r
naive_forecast %>%
   autoplot(combined_tsibble)
```

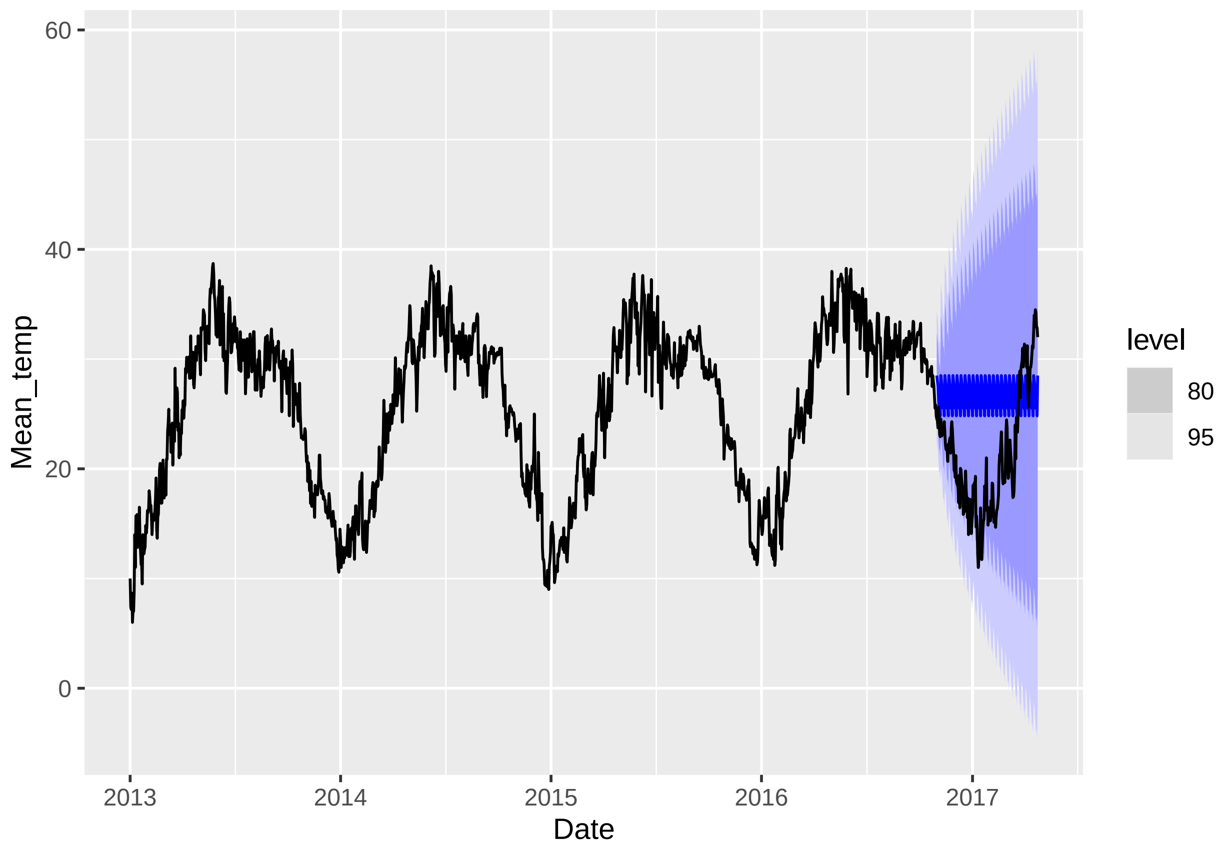


SNaive

```r
snaive_forecast <- fit_snaive %>% forecast(test_ts)
snaive_acc <- fabletools::accuracy(snaive_forecast, test_ts)
snaive_acc
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 snaive Test  -5.47  7.81  6.87 -34.0  38.4   NaN   NaN 0.915
```

```r
snaive_forecast %>%
   autoplot(combined_tsibble)
```

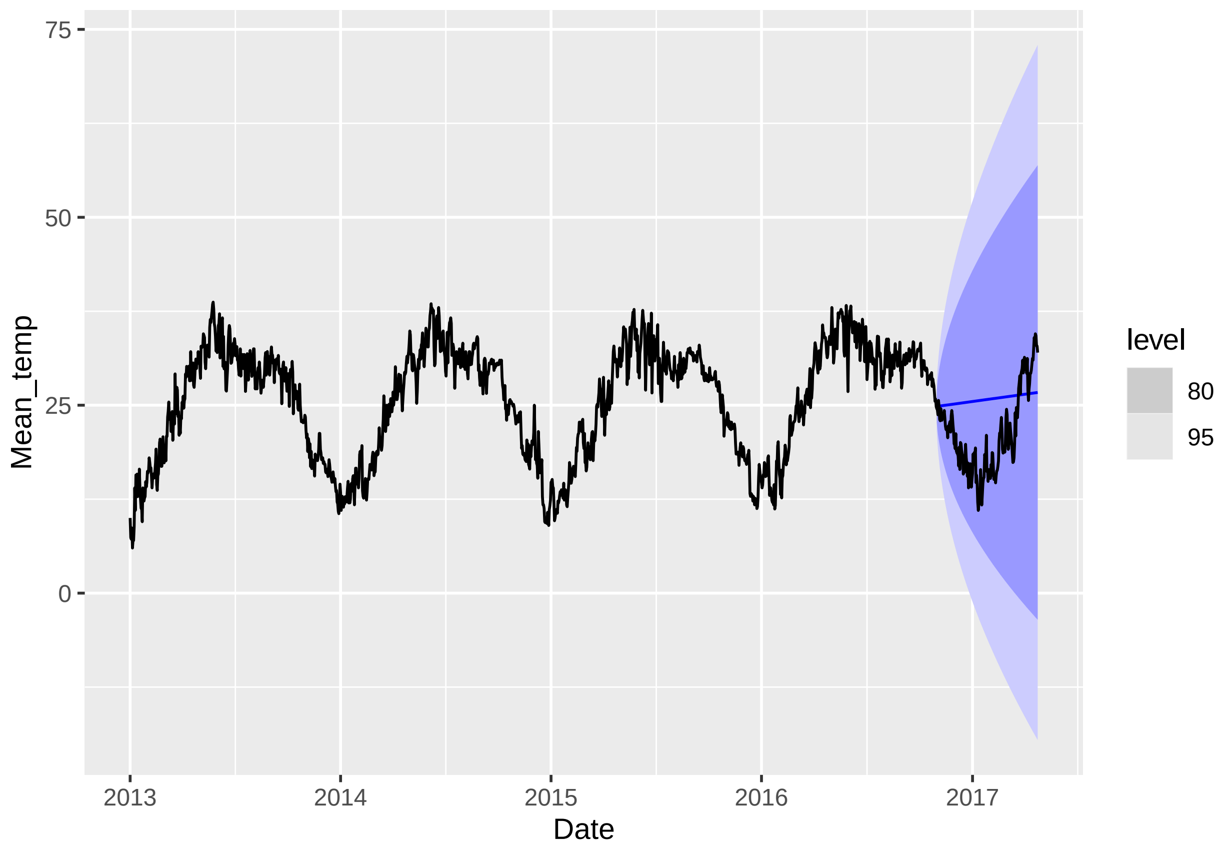


Drift

```
drift_forecast <- fit_drift %>% forecast(test_ts)
drift_acc <- fabletools::accuracy(drift_forecast, test_ts)
drift_acc
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 drift  Test  -4.54  6.92  5.99 -29.1  33.7   NaN   NaN 0.941
```

```
drift_forecast %>%
  autoplot(combined_tsibble)
```



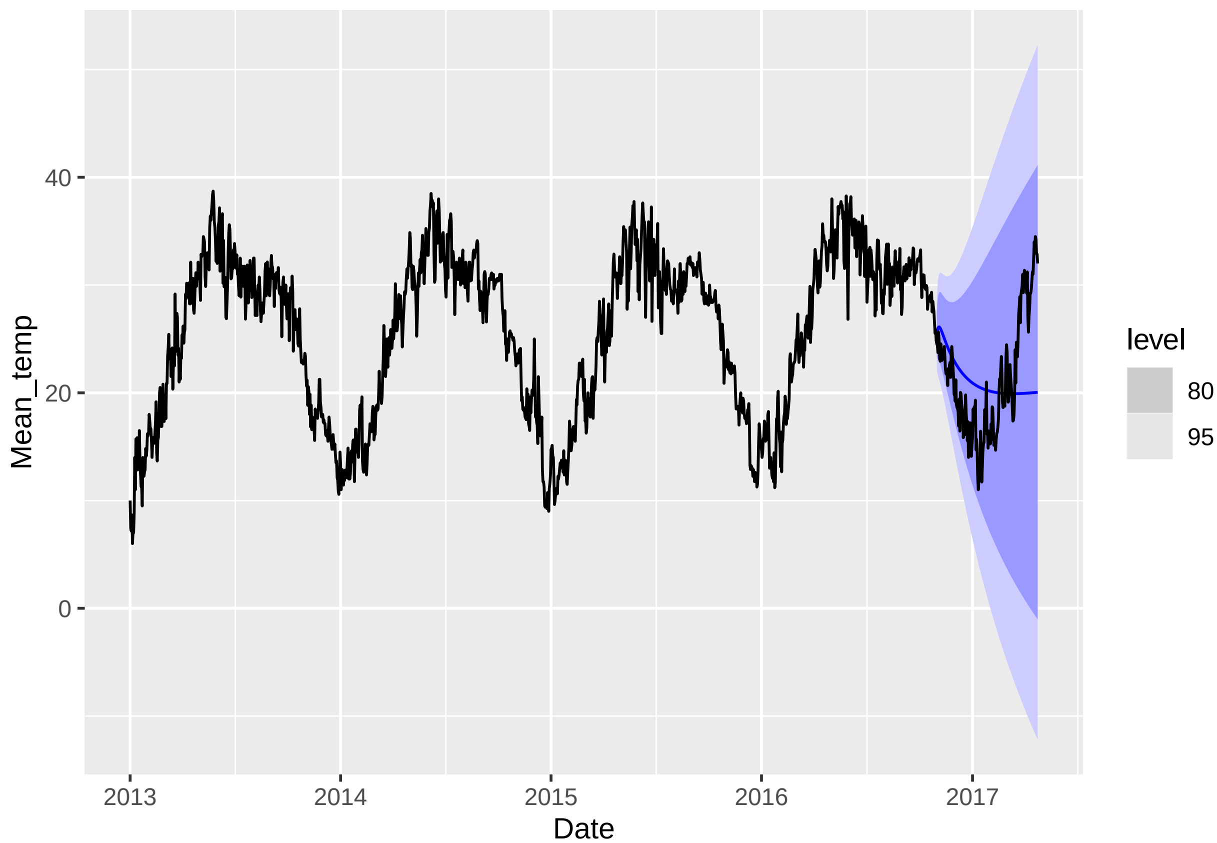
3. ARIMA

With a constant

```
arima_forecast <- fit_arima %>% forecast(test_ts)
arima_acc <- fabletools::accuracy(arima_forecast, test_ts)
arima_acc
```

```
## # A tibble: 1 x 10
##   .model .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima  Test  0.0161  5.73  4.55 -6.22  21.8   NaN   NaN 0.949
```

```
arima_forecast %>%
  autoplot(combined_tsibble)
```

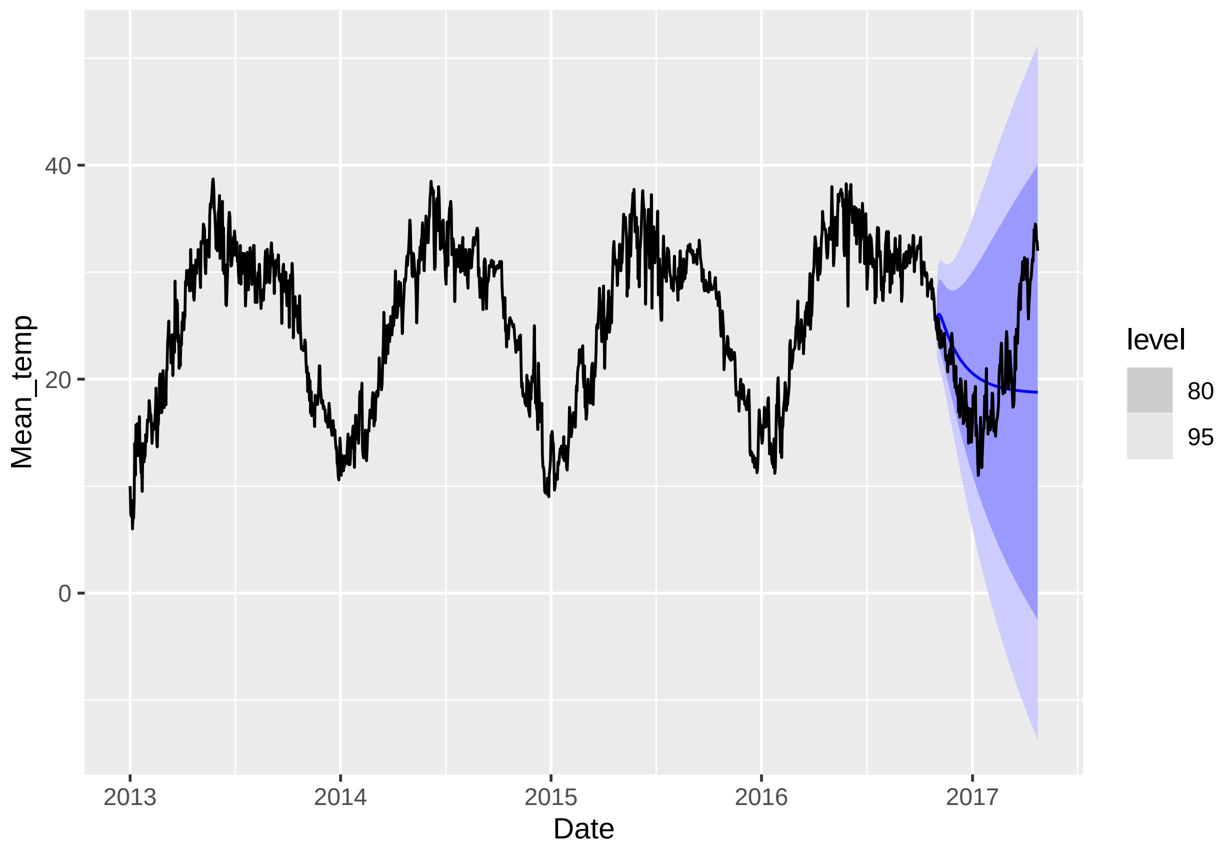


without a constant

```
arima_no_cons_forecast <- fit_arima_no_cons %>% forecast(test_ts)
arima_no_cons_acc <- fabletools::accuracy(arima_no_cons_forecast, test_ts)
arima_no_cons_acc
```

```
## # A tibble: 1 x 10
##   .model         .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>          <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima_no_cons Test  0.601  6.05  4.65 -3.48  21.6   NaN   NaN 0.952
```

```
arima_no_cons_forecast %>%
  autoplot(combined_tsibble)
```

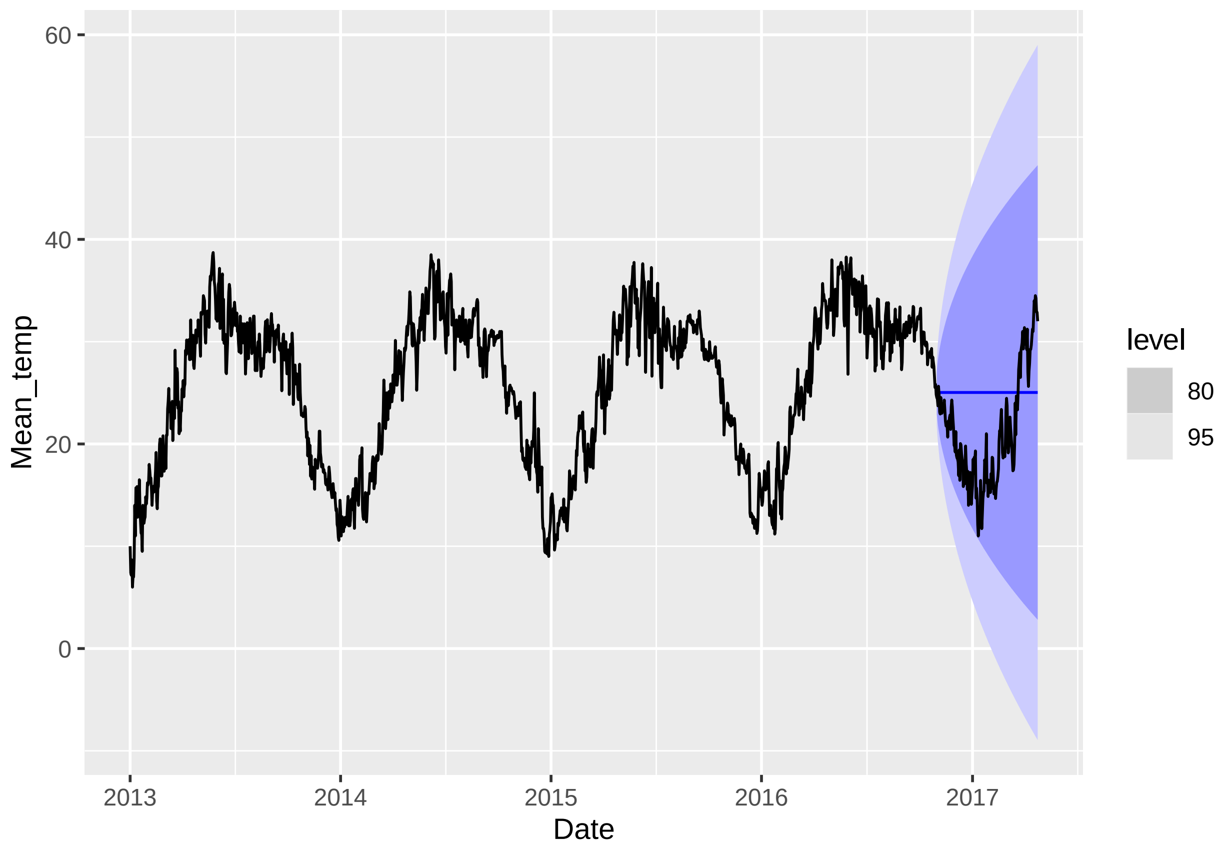


4. EWMA

```
ewma_forecast <- fit_ewma %>% forecast(test_ts)
ewma_acc <- fabletools::accuracy(ewma_forecast, test_ts)
ewma_acc
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ewma   Test  -3.80  6.65  5.82 -25.6  32.2   NaN   NaN 0.945
```

```
ewma_forecast %>%
   autoplot(combined_tsibble)
```

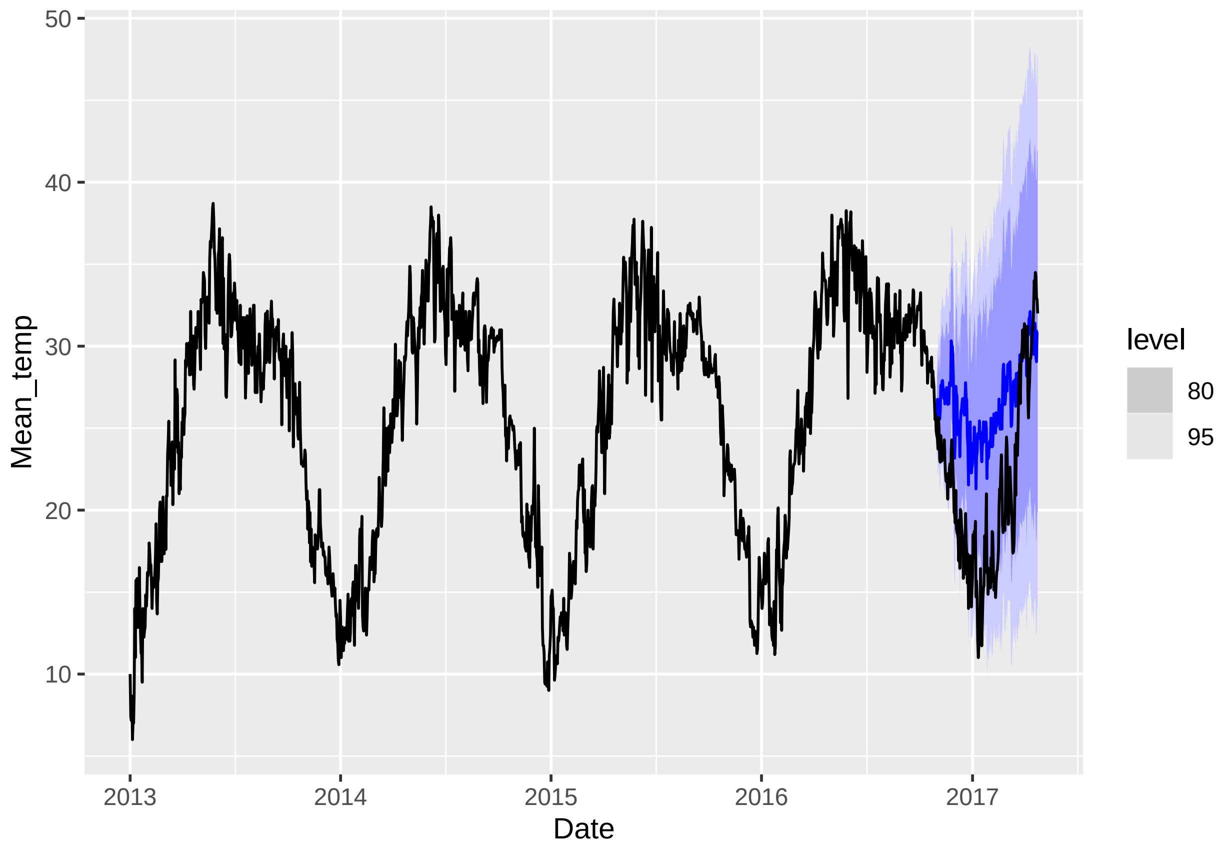


5. Dynamic regression

```
darima_forecast <- fit_darima %>% forecast(test_ts)
darima_acc <- fabletools::accuracy(darima_forecast, test_ts)
darima_acc
```

```
## # A tibble: 1 x 10
##   .model        .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>         <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 dynamic_arima Test  -5.44  6.58  5.81 -31.4  32.6   NaN   NaN 0.915
```

```
darima_forecast %>%
  autoplot(combined_tsibble)
```

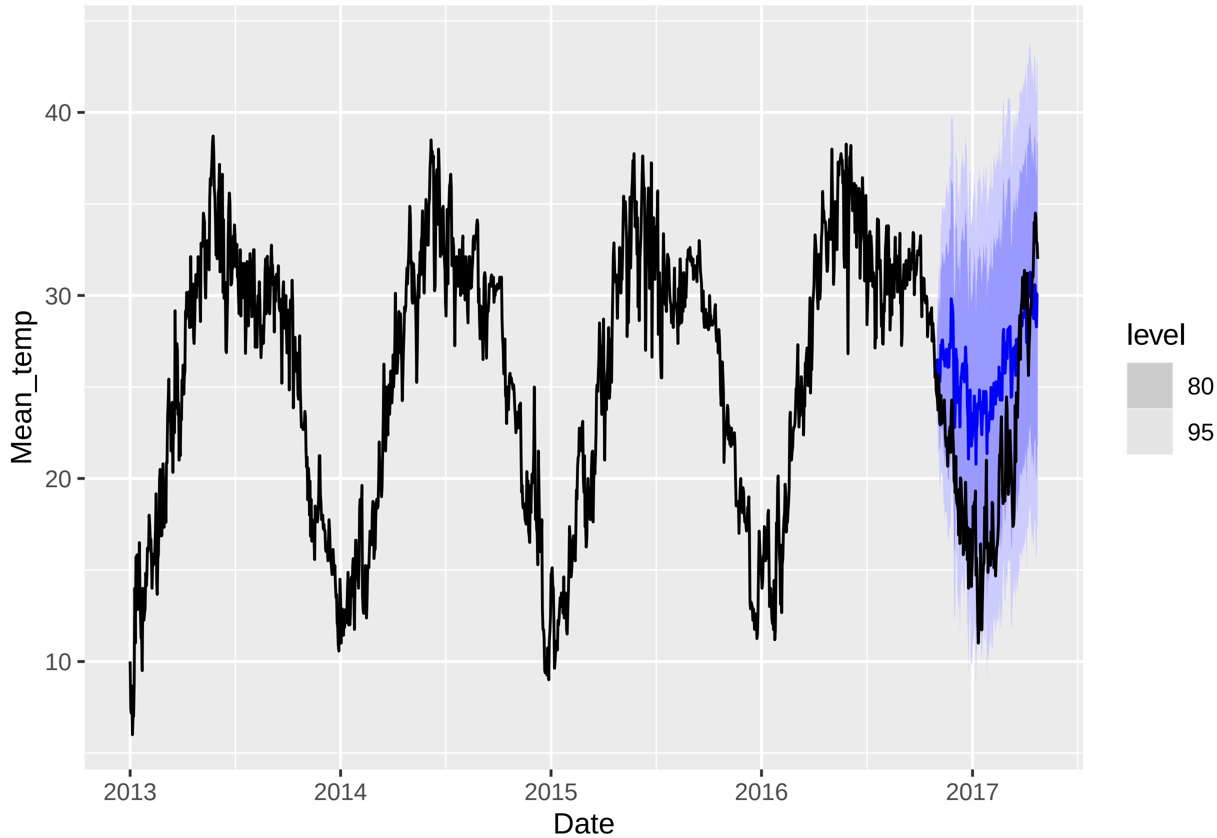


Arima error with difference 0

```
darima_diff0_forecast <- fit_darima_diff0 %>% forecast(test_ts)
darima_diff0_acc <- fabletools::accuracy(darima_diff0_forecast, test_ts)
darima_diff0_acc
```

```
## # A tibble: 1 x 10
##   .model               .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>                <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 dynamic_arima_diff0 Test  -4.84  6.12  5.41 -28.5  30.3   NaN   NaN 0.917
```

```
darima_diff0_forecast %>%
  autoplot(combined_tsibble)
```

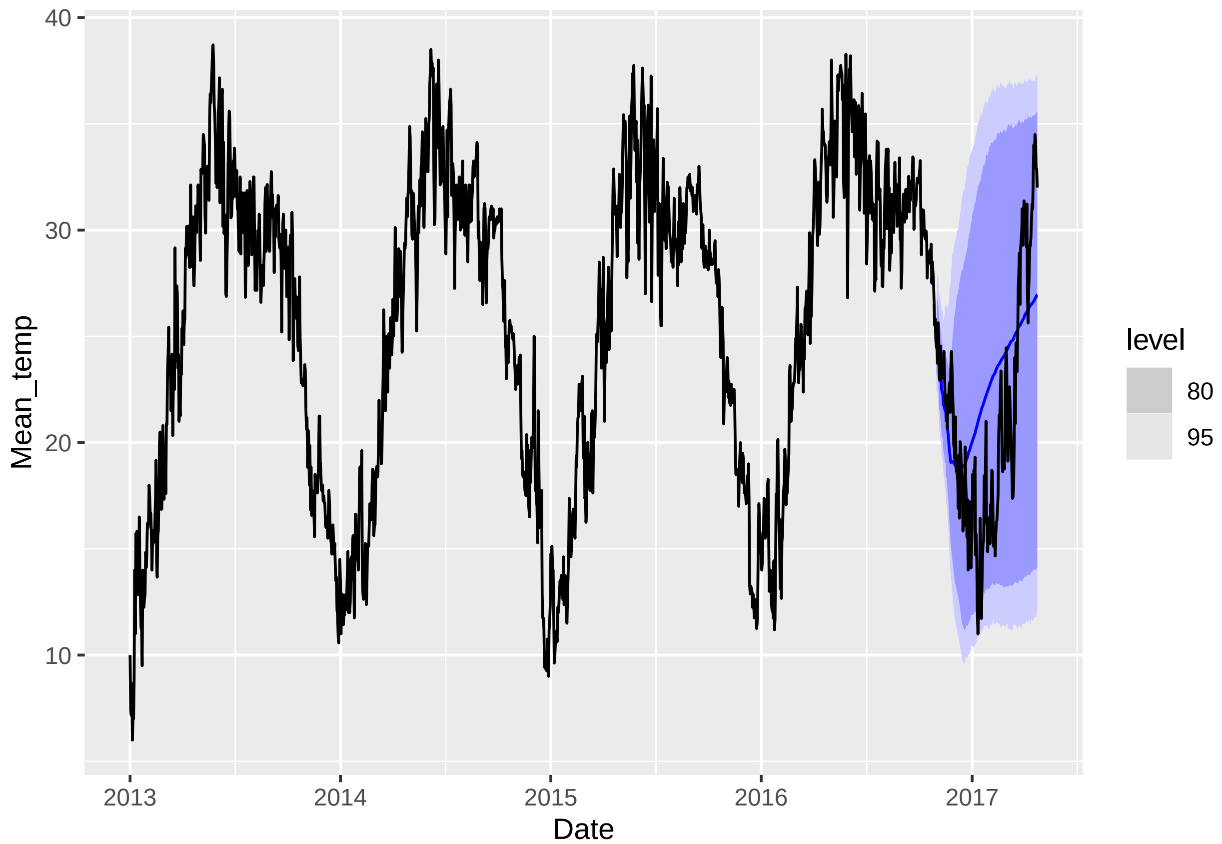


6. NNAR

```
nnar_forecast <- fit_nnar %>% forecast(test_ts)
nnar_acc <- fabletools::accuracy(nnar_forecast, test_ts)
nnar_acc
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 nnar   Test  -1.48  4.46  3.66 -12.1  19.8   NaN   NaN 0.920
```

```
nnar_forecast %>%
   autoplot(combined_tsibble)
```

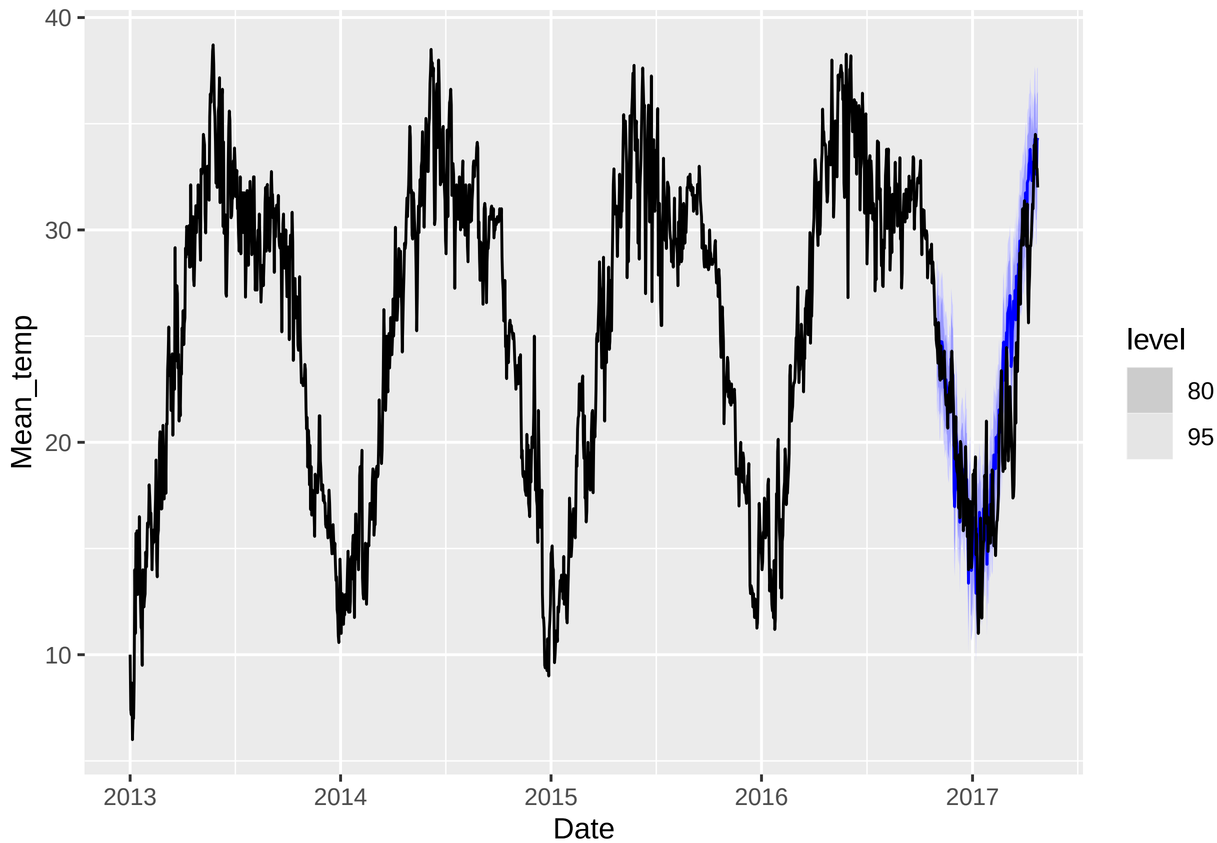


7. Prophet

```
prophet_forecast <- fit_prophet %>% forecast(test_ts)
prophet_acc <- fabletools::accuracy(prophet_forecast, test_ts)
prophet_acc
```

```
## # A tibble: 1 x 10
##   .model  .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 prophet Test  -1.20  2.76  1.98 -6.32  10.3   NaN   NaN 0.839
```

```
prophet_forecast %>%
  autoplot(combined_tsibble)
```



After evaluation of those models, we found that Prophet has the best performance among them.

IV. Tune prophet model

```
fit_multi_prophet <- train_ts_imputed %>%
  model(
    prophet_all = prophet(Mean_temp ~ Mean_pressure + Humidity + Wind_speed+
                      season(period = "week", order = 10) +
                      season(period = "year", order = 5)),
    prophet_no_pred = prophet(Mean_temp ~
                      season(period = "week", order = 10) +
                      season(period = "year", order = 5)),
    prophet_humi = prophet(Mean_temp ~ Humidity +
                      season(period = "week", order = 10) +
                      season(period = "year", order = 5)),
    prophet_wind = prophet(Mean_temp ~ Wind_speed +
                      season(period = "week", order = 10) +
                      season(period = "year", order = 5)),
    prophet_pressure = prophet(Mean_temp ~ Mean_pressure +
                      season(period = "week", order = 10) +
                      season(period = "year", order = 5)),
    prophet_h_w = prophet(Mean_temp ~ Humidity + Wind_speed+
                      season(period = "week", order = 10) +
                      season(period = "year", order = 5)),
```

```
    prophet_h_p = prophet(Mean_temp ~ Humidity + Mean_pressure+
                        season(period = "week", order = 10) +
                        season(period = "year", order = 5)),
    prophet_w_p = prophet(Mean_temp ~ Wind_speed + Mean_pressure+
                        season(period = "week", order = 10) +
                        season(period = "year", order = 5)),
  )
```

Evaluate:

```
multi_prophet_forecast <- fit_multi_prophet %>% forecast(test_ts)
multi_prophet_acc <- multi_prophet_forecast %>% fabletools::accuracy(test_ts)
multi_prophet_acc
```

```
## # A tibble: 8 x 10
##   .model           .type        ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <chr>            <chr>     <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 prophet_all      Test  -1.20      2.75  1.98 -6.32  10.3    NaN   NaN 0.839
## 2 prophet_h_p      Test  -1.06      2.67  1.94 -5.68  10.1    NaN   NaN 0.836
## 3 prophet_h_w      Test  -1.20      2.76  1.98 -6.31  10.4    NaN   NaN 0.840
## 4 prophet_humi     Test  -1.13      2.71  1.96 -6.01  10.2    NaN   NaN 0.836
## 5 prophet_no_pred  Test  -0.0407    2.30  1.82 -1.11   9.32   NaN   NaN 0.757
## 6 prophet_pressure Test   0.0455    2.28  1.82 -0.687  9.26   NaN   NaN 0.752
## 7 prophet_w_p      Test   0.000772  2.27  1.81 -0.929  9.23   NaN   NaN 0.751
## 8 prophet_wind     Test  -0.0314    2.28  1.81 -1.09   9.27   NaN   NaN 0.752
```

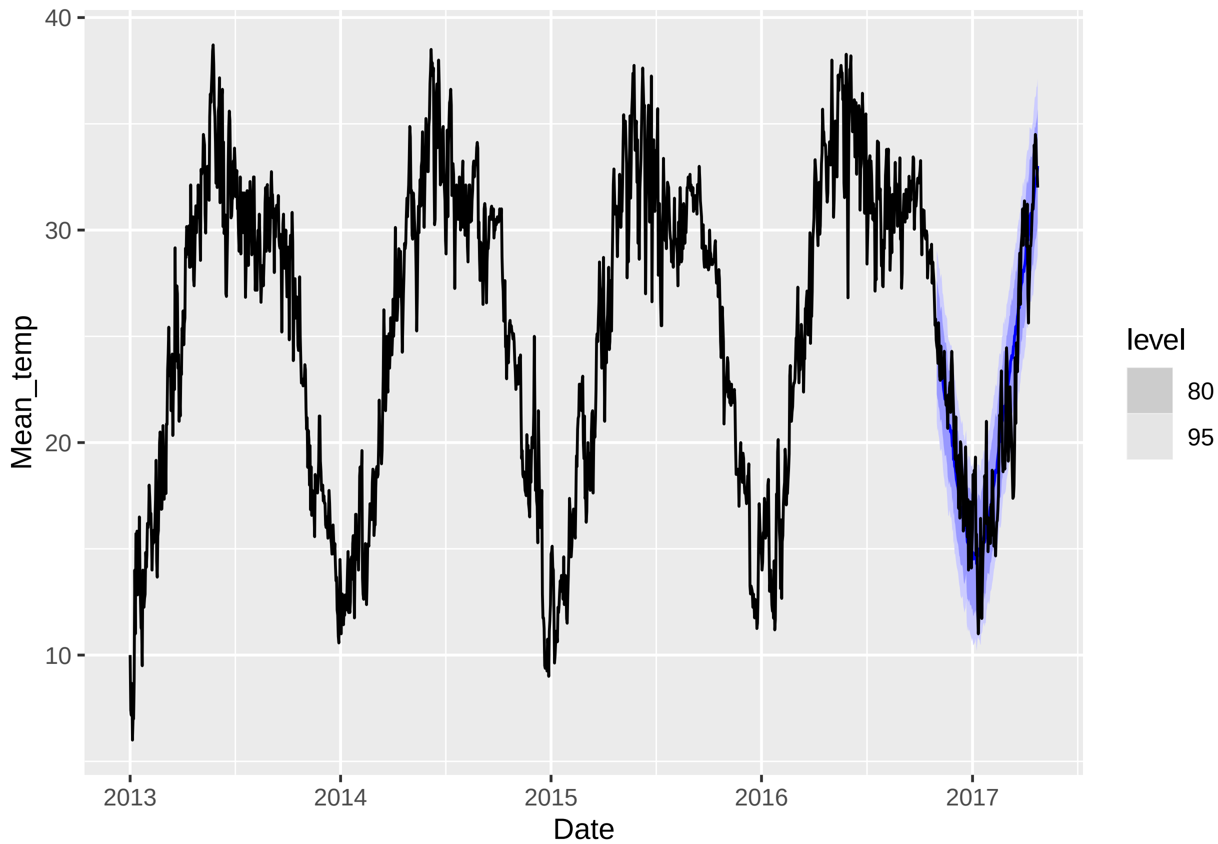
Model prophet_w_p has the best performance with a value of RMSE is equal to 2.29.

Plot:

```
w_p_row_number <- which(multi_prophet_forecast$.model == 'prophet_w_p')
multi_prophet_forecast[w_p_row_number,] %>% fabletools::accuracy(test_ts)
```

```
## # A tibble: 1 x 10
##   .model      .type       ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <chr>       <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 prophet_w_p Test  0.000772  2.27  1.81 -0.929  9.23   NaN   NaN 0.751
```

```
multi_prophet_forecast[w_p_row_number,] %>%
  autoplot(combined_tsibble)
```



V. Results

1. All models

```
# for final result recording
final_accs <- tslm_acc
final_accs <- rbind(final_accs, mean_acc)
final_accs <- rbind(final_accs, naive_acc)
final_accs <- rbind(final_accs, snaive_acc)
final_accs <- rbind(final_accs, drift_acc)
final_accs <- rbind(final_accs, arima_acc)
final_accs <- rbind(final_accs, arima_no_cons_acc)
final_accs <- rbind(final_accs, ewma_acc)
final_accs <- rbind(final_accs, darima_acc)
final_accs <- rbind(final_accs, darima_diff0_acc)
final_accs <- rbind(final_accs, nnar_acc)
final_accs <- rbind(final_accs, prophet_acc)

rs_dir <- "result"
if (!dir.exists(rs_dir)) {
  dir.create(rs_dir)
  cat("Directory created:", rs_dir, "\n")
} else {
```

```
  cat("Directory already exists:", rs_dir, "\n")
}
```

```
## Directory already exists: result
```

```
rs_file_path <- "result/all_model_accs.csv"
write.csv(final_accs,
          file = rs_file_path,
          row.names = FALSE)
```

2. Fine tune prophet models

```
prophet_file_path <- "result/prophet_fine_tune.csv"
write.csv(multi_prophet_acc,
          file = prophet_file_path,
          row.names = FALSE)
```