# W271 Assignment 7

```r
library(tidyverse)
library(magrittr)
library(patchwork)

library(lubridate)

library(tsibble)
library(feasts)
install.packages('forecast')
library(forecast)

library(sandwich)
library(lmtest)

library(nycflights13)
install.packages('blsR')
library(blsR)
```

```r
theme_set(theme_minimal())
```

## Question-1: AIC and BIC and "Stringency"

### (4 points) Part-1

In the async lecture, Jeffrey says "BIC is in general more stringent than AIC or AICc". Let's illustrate that and reason about it.

1. Produce a dataset, `d`, that includes 100 observations of pure white-noise.
    - The outcome variable should be a variable `y` that has 100 draws from `rnorm`, with `mean=0` and `sd=1`.
    - The input variables should be variables `x1` … `x10` that are also 100 draws from `rnorm` each with `mean=0` and `sd=1`.
    - There are fancy ways to write this code; the goal for this isn't to place a clever coding task in front of you, so feel free to use copy-paste to create the data object in any way that you can.
2. After producing data, fit 11 models against that data, stored as `model0` through `model10`. (The number appended to `model` corresponds to the number of parameters that you have used in your estimation).
3. After estimating your models, create a new dataset, `results_data`, that contains the number of parameters that you have used in an estimation, and the AIC and BIC values that you calculated for that number of parameters.
    1. Note – this is another place where the way that you create the data, and the way that the data is the most useful to use are incongruent.
    2. When we created the data, we created a dataset that has a column called `parameters`, a column called `aic` and a column called `bic`.
    3. However, it is much more useful to have "tidy" data that has these values stacked. If you find yourself creating the dataset in the "wide" form that we have described above, you can use the `dplyr::pivot_longer` function to pivot your data into a tidy format. Specifically, we used this call `pivot_longer(cols = c('aic', 'bic'))` with our input data structure.

4. Finally, produce a plot that shows the AIC and BIC values on the y-axis and the number of estimated parameters on the x-axis. In the subtitle to your plot, note whether a relatively higher or lower AIC or BIC means that a model is performing better or worse (i.e. either “Higher values are better” or “Lower values are better”). What do you notice about these plots, and what does this tell you about the “stringency” of AIC vs. BIC?

```r
d <- data.frame(
  y = rnorm(100, mean=0, sd=1),
  x1 = rnorm(100, mean=0, sd=1),
  x2 = rnorm(100, mean=0, sd=1),
  x3 = rnorm(100, mean=0, sd=1),
  x4 = rnorm(100, mean=0, sd=1),
  x5 = rnorm(100, mean=0, sd=1),
  x6 = rnorm(100, mean=0, sd=1),
  x7 = rnorm(100, mean=0, sd=1),
  x8 = rnorm(100, mean=0, sd=1),
  x9 = rnorm(100, mean=0, sd=1),
  x10 = rnorm(100, mean=0, sd=1)
)
head(d)
```

```
##            y         x1         x2         x3         x4         x5         x6
## 1 -1.4354407 -0.7580164  1.4459333  1.9199328 -1.6403735  0.5356636  0.7231867
## 2  0.6251068  0.1057225  0.3639325 -1.9540449  0.0571696 -0.1542156 -1.4159535
## 3 -1.7491418  2.0524703 -1.0145225 -0.1342037 -0.3082965 -0.7346283  0.7736912
## 4 -1.1145482  0.6688293 -0.7798759 -1.9806666  0.2522386 -0.4201904  1.3730310
## 5  0.5156526 -1.0931410  0.9359561  0.6050061  0.3199418 -0.1025155 -0.2502545
## 6 -1.6584507 -1.0573737  0.5136871 -0.3935659 -0.3904988  1.2389127 -0.2695691
##             x7          x8          x9        x10
## 1 -0.04320003  1.12927976  0.84794795 -0.6995596
## 2  0.77149598  1.47852853 -0.09137578 -0.6426201
## 3 -0.45886119 -0.51513427  0.64213080  1.7537416
## 4 -1.86144959  0.07304316 -0.64616061 -2.0741884
## 5  0.82740602  1.10986768 -1.04732511 -1.4638620
## 6 -1.20107770 -1.13985960  1.18713589  0.1917949
```

```r
model0 <- lm(y ~ 1, data=d)
model1 <- lm(y ~ x1, data=d)
model2 <- lm(y ~ x1 + x2, data=d)
model3 <- lm(y ~ x1 + x2 + x3, data=d)
model4 <- lm(y ~ x1 + x2 + x3 + x4, data=d)
model5 <- lm(y ~ x1 + x2 + x3 + x4
             + x5, data=d)
model6 <- lm(y ~ x1 + x2 + x3 + x4
             + x5 + x6, data=d)
model7 <- lm(y ~ x1 + x2 + x3 + x4
             + x5 + x6 + x7, data=d)
model8 <- lm(y ~ x1 + x2 + x3 + x4
             + x5 + x6 + x7 + x8, data=d)
model9 <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 +
               x8 + x9, data=d)
model10 <- lm(y ~ x1 + x2 + x3 + x4
              + x5 + x6 + x7 +
                x8 + x9 + x10, data=d)
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x1, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8923 -0.7460  0.1064  0.7229  3.0198
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.14737    0.10385  -1.419    0.159
## x1          -0.08148    0.10326  -0.789    0.432
##
## Residual standard error: 1.034 on 98 degrees of freedom
## Multiple R-squared:  0.006313,   Adjusted R-squared:  -0.003826
## F-statistic: 0.6226 on 1 and 98 DF,  p-value: 0.432
```

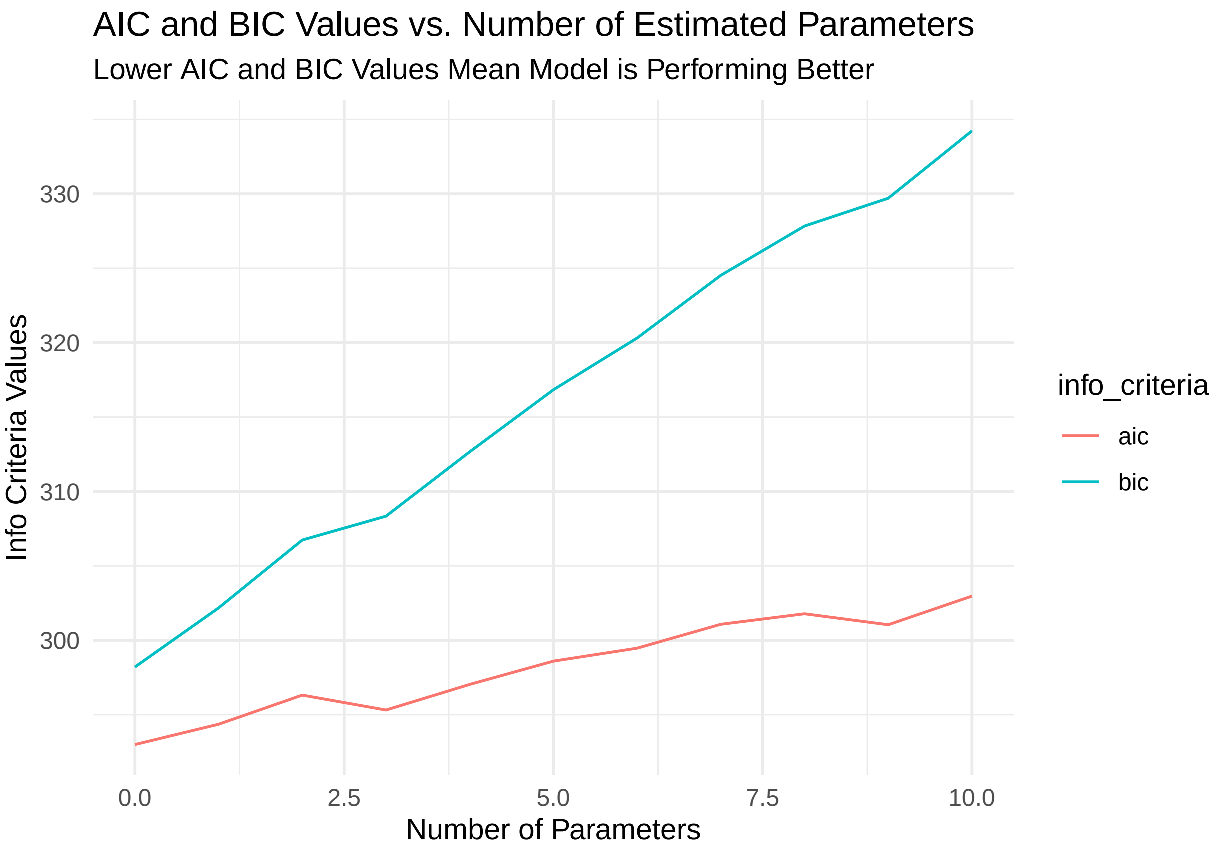
```r
results_data <- data.frame(
  parameters = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
  aic = c(AIC(model0), AIC(model1), AIC(model2),
          AIC(model3), AIC(model4),
          AIC(model5), AIC(model6), AIC(model7),
          AIC(model8), AIC(model9),
          AIC(model10)),
  bic = c(BIC(model0), BIC(model1), BIC(model2),
          BIC(model3), BIC(model4),
          BIC(model5), BIC(model6), BIC(model7),
          BIC(model8), BIC(model9),
          BIC(model10))
)
head(results_data)
```

```
##   parameters      aic      bic
## 1          0 292.9933 298.2036
## 2          1 294.3599 302.1754
## 3          2 296.3166 306.7372
## 4          3 295.3142 308.3401
## 5          4 297.0337 312.6648
## 6          5 298.5992 316.8354
```

```r
tidy_data <- pivot_longer(results_data,
                          cols = c(aic, bic),
                          names_to = "info_criteria",
                          values_to = "value")
head(tidy_data)
```

```
## # A tibble: 6 x 3
##   parameters info_criteria value
##        <dbl> <chr>         <dbl>
## 1          0 aic            293.
## 2          0 bic            298.
## 3          1 aic            294.
## 4          1 bic            302.
## 5          2 aic             296.
## 6          2 bic             307.
```

```
tidy_data %>% ggplot(aes(x=parameters, y = value, color= info_criteria)) +
  geom_line() + xlab("Number of Parameters") + ylab("Info Criteria Values") +
  ggtitle('AIC and BIC Values vs. Number of Estimated Parameters') +
  labs(subtitle = "Lower AIC and BIC Values Mean Model is Performing Better")
```



Overall, there is a positive association between the number of estimated model parameters used and the AIC values. More parameters are associated with higher AIC values. Similarly, there is a positive association between the number of estimated model parameters used and the BIC values. More parameters are associated with higher BIC values. We see that the BIC increases much faster than the AIC when we increase the number of estimated parameters. This means that the BIC is a more stringent metric than AIC since the larger BIC values would penalize the model more than AIC would.

This plot indicates that models with fewer parameters perform better than models with more parameters. This makes sense since we are modeling pure white noise, which is stationary by definition. Due to its constant mean, a constant linear model with 0 parameters would perform best.

### (2 points) Part-2

Now, suppose that you had data that, *in the population model* actually held a relationship between the input features and the outcome feature. Specifically, suppose that for every unit increase in `x1` there was a `0.1` increase in the outcome, for every unit increase in `x2` there was a `0.2` increase in the outcome, . . ., for every unit increase in `x10` there was a `1.0` unit increase in the outcome. Suppose that if all `x1` . . . `x10` were zero, that the outcome would have an expectation of zero, but with white-noise around it with $\mu = 0$ and $\sigma = 1$.

- Modify the code that you wrote above to create data according to this schedule.
- Estimate 11 models as before.

- Produce a new dataset `resutls_data` that contains the AIC and BIC values from each of these models.
- Produce the same plot as you did before with the white noise series. Comment on what, if anything is similar or different between this plot, and the plot you created before.

```
new_d <- d
new_d$y <- 0.1 * new_d$x1 + 0.2 * new_d$x2 + 0.3 * new_d$x3 + 0.4 * new_d$x4 +
    0.5 * new_d$x5 + 0.6 * new_d$x6 + 0.7 * new_d$x7 + 0.8 * new_d$x8 +
    0.9 * new_d$x9 + 1 * new_d$x10 + rnorm(100, mean=0, sd=1)
head(new_d)
```

```
##            y         x1         x2         x3         x4         x5         x6
## 1  2.6675946 -0.7580164  1.4459333  1.9199328 -1.6403735  0.5356636  0.7231867
## 2  1.0670484  0.1057225  0.3639325 -1.9540449  0.0571696 -0.1542156 -1.4159535
## 3  2.3637189  2.0524703 -1.0145225 -0.1342037 -0.3082965 -0.7346283  0.7736912
## 4 -3.5256161  0.6688293 -0.7798759 -1.9806666  0.2522386 -0.4201904  1.3730310
## 5 -0.9519386 -1.0931410  0.9359561  0.6050061  0.3199418 -0.1025155 -0.2502545
## 6 -0.8170592 -1.0573737  0.5136871 -0.3935659 -0.3904988  1.2389127 -0.2695691
##             x7          x8          x9        x10
## 1 -0.04320003  1.12927976  0.84794795 -0.6995596
## 2  0.77149598  1.47852853 -0.09137578 -0.6426201
## 3 -0.45886119 -0.51513427  0.64213080  1.7537416
## 4 -1.86144959  0.07304316 -0.64616061 -2.0741884
## 5  0.82740602  1.10986768 -1.04732511 -1.4638620
## 6 -1.20107770 -1.13985960  1.18713589  0.1917949
```

```
model0 <- lm(y ~ 1, data=new_d)
model1 <- lm(y ~ x1, data=new_d)
model2 <- lm(y ~ x1 + x2, data=new_d)
model3 <- lm(y ~ x1 + x2 + x3, data=new_d)
model4 <- lm(y ~ x1 + x2 + x3 + x4, data=new_d)
model5 <- lm(y ~ x1 + x2 + x3 + x4 + x5,
             data=new_d)
model6 <- lm(y ~ x1 + x2 + x3 + x4 + x5
             + x6, data=new_d)
model7 <- lm(y ~ x1 + x2 + x3 + x4 + x5
             + x6 + x7, data=new_d)
model8 <- lm(y ~ x1 + x2 + x3 + x4 + x5
             + x6 + x7 + x8, data=new_d)
model9 <- lm(y ~ x1 + x2 + x3 + x4 + x5
             + x6 + x7 +
               x8 + x9, data=new_d)
model10 <- lm(y ~ x1 + x2 + x3 + x4 + x5
              + x6 + x7 +
                x8 + x9 + x10, data=new_d)
```

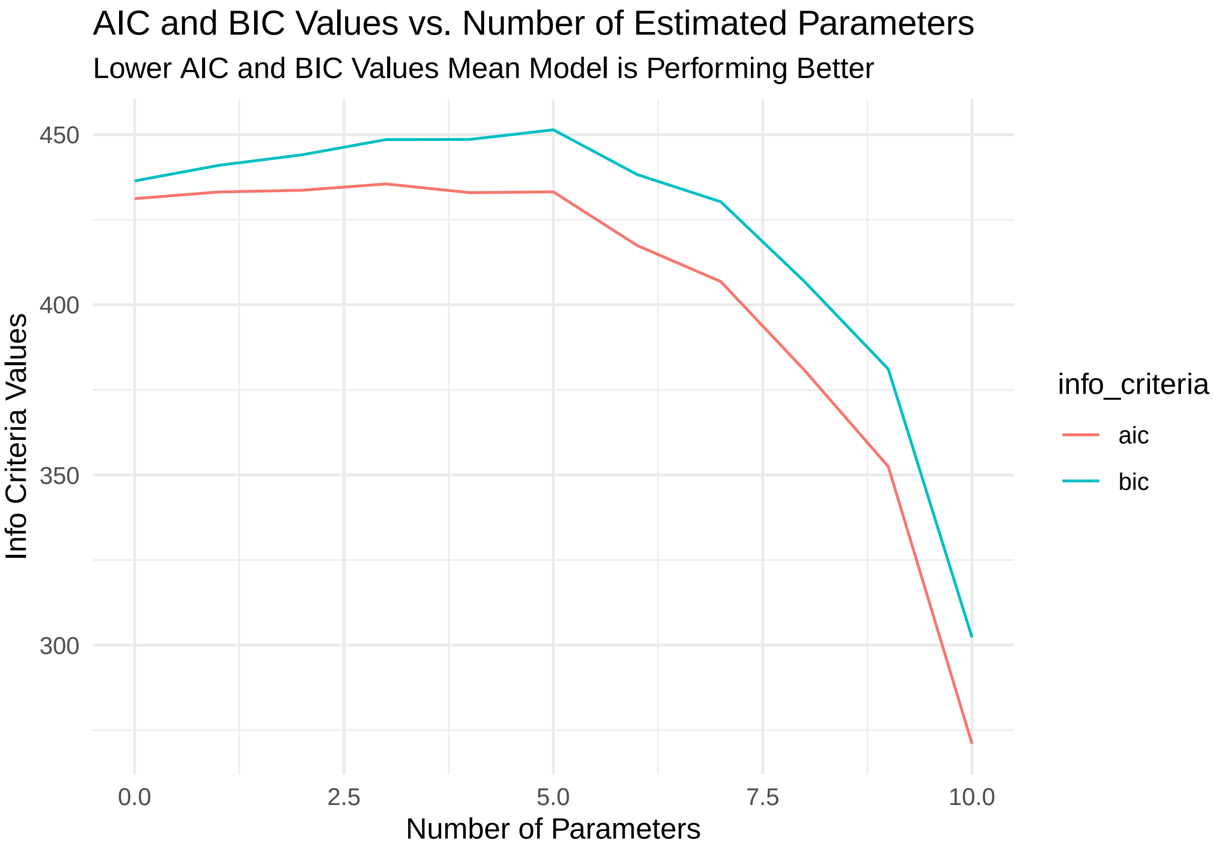
```
results_data <- data.frame(
  parameters = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
  aic = c(AIC(model0), AIC(model1), AIC(model2),
          AIC(model3), AIC(model4),
          AIC(model5), AIC(model6), AIC(model7),
          AIC(model8), AIC(model9),
          AIC(model10)),
  bic = c(BIC(model0), BIC(model1), BIC(model2),
          BIC(model3), BIC(model4),
          BIC(model5), BIC(model6), BIC(model7),
          BIC(model8), BIC(model9),
          BIC(model10))
)
head(results_data)
```

```
##   parameters      aic      bic
## 1          0 431.1797 436.3900
## 2          1 433.1332 440.9487
## 3          2 433.6531 444.0738
## 4          3 435.4991 448.5249
## 5          4 432.9467 448.5777
## 6          5 433.1753 451.4115
```

```
tidy_data <- pivot_longer(results_data,
                          cols = c(aic, bic),
                          names_to = "info_criteria",
                          values_to = "value")
head(tidy_data)
```

```
## # A tibble: 6 x 3
##   parameters info_criteria value
##        <dbl> <chr>         <dbl>
## 1          0 aic            431.
## 2          0 bic            436.
## 3          1 aic            433.
## 4          1 bic            441.
## 5          2 aic            434.
## 6          2 bic            444.
```

```
tidy_data %>% ggplot(aes(x=parameters, y = value, color= info_criteria)) +
  geom_line() + xlab("Number of Parameters") + ylab("Info Criteria Values") +
  ggtitle('AIC and BIC Values vs. Number of Estimated Parameters') +
  labs(subtitle = "Lower AIC and BIC Values Mean Model is Performing Better")
```



Compared to the plot for the white noise series, this plot shows that increasing the number of estimated model parameters will eventually drive down the AIC and BIC values. The BIC values slightly increase but fall drastically past 6 parameters. The AIC values begin with a slight decrease but then begin to fall much quicker for number of parameters greater than 6.

According to the AIC, the constant model with 0 parameters would have the worst performance since it has the highest AIC value. According to the BIC, the model with 6 parameters would have the worst performance since it has the highest BIC value. However, the constant model with 0 parameters also performs poorly compared to models with more than 6 parameters. For both AIC and BIC metrics, the linear model with 10 parameters performs best. This makes sense since we defined the outcome to be a linear function of x1 through x10 plus a white noise term.

## Question-2: Weather in NYC

Our goals with this question are to:

- (If necessary) Clean up code that we've written before to re-use. This task of writing code, and then coming back and using it later is often overlooked in the MIDS program. Here's a chance to practice!
- Estimate several different polynomial regressions against a time series and evaluate at what point we have produced a model with "enough complexity" that the model evaluation scores cease to tell us that additional model parameters are improving the model fit.

### (1 point) Part-1: Load the Weather Data

Load the weather data in the same way as you did in the previous assignment, recalling that there was some weird duplication of data for one of the days. Then, create an object, `weather_weekly` that aggregates the data to have two variables `average_temperature` and `average_dewpoint` at the year-week level, for each airport. After your aggregation is complete, you should have a `tsibble` that has the following shape:

```
A tsibble: 159 x 4 [1W]
# Key:       origin [3]
   origin week_index average_temperature average_dewpoint
   <chr>      <week>               <dbl>            <dbl>
 1 EWR      2013 W01                34.3             19.4
 2 EWR      2013 W02                42.7             33.3
 3 EWR      2013 W03                39.6             26.5
```

```
head(weather)
```

```
## # A tibble: 6 x 15
##   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed wind_gust
##   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>     <dbl>
## 1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4         NA
## 2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06        NA
## 3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5         NA
## 4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7         NA
## 5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7         NA
## 6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5         NA
## # i 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,
## #   time_hour <dttm>
```

```
weather_weekly <- weather %>%
   mutate(time_index = make_datetime(year, month, day, hour)) %>%
   mutate(week_index = yearweek(time_index)) %>%
   select(origin, week_index, temp, dewp) %>% group_by(origin, week_index) %>%
   summarise(average_temperature = round(mean(temp), 1),
            average_dewpoint=round(mean(dewp), 1)) %>% ungroup() %>%
   as_tsibble(index=week_index, key = c(origin))
```

```
## `summarise()` has grouped output by 'origin'. You can override using the
## `.groups` argument.
```

```
weather_weekly
```

```
## # A tsibble: 159 x 4 [1W]
## # Key:       origin [3]
##    origin week_index average_temperature average_dewpoint
##    <chr>      <week>               <dbl>            <dbl>
##  1 EWR      2013 W01                34.3             19.4
##  2 EWR      2013 W02                42.7             33.3
##  3 EWR      2013 W03                39.6             26.5
##  4 EWR      2013 W04                21.3              3.8
##  5 EWR      2013 W05                36               25.7
##  6 EWR      2013 W06                30               17.6
##  7 EWR      2013 W07                37.8             25.3
##  8 EWR      2013 W08                34.2             21.3
##  9 EWR      2013 W09                39.2             27.8
## 10 EWR      2013 W10                38.7             26.8
## # i 149 more rows
```

## (2 points) Part-2: Fit Polynomial Regression Models

For each of the `average_temperature` and `average_dewpoint` create ten models that include polynomials of increasing order.

- One issue that you're likely to come across is dealing with how to make the time index that you're

using in your `tsibble` work with either `poly` or some other function to produce the polynomial terms; this arises because although the time index is ordered, it isn't really a "numeric" feature so when you call for something like, `poly(week_index, degree=2)` you will be met with an error.
- Cast the index to a numeric variable, where the first week is indexed to be `0`. Recall that Jeffrey notes that this form of translation only changes the way that the intercept is interpreted; we will note that because the `as.numeric(week_index)` creates input variables that are in the vicinity, it also changes the magnitude of the higher-order polynomial terms that are estimated, though it does not change the regression diagnostics and model scoring to transform (or not) these time index variables.

Additionally, you might recall that in 203, we actually recommended you away from using the `poly` function. That was a recommendation based on students' knowledge at the time, when we were considering fitting log and square root transformations of data. At this point, you can handle the additional complexity and can take the recommendation that `poly` is nice for working with polynomial translations of time.

```
weather_tsib_cast <- weather_weekly
weather_tsib_cast$week_index <- as.numeric(weather_tsib_cast$week_index) - 2244
weather_tsib_cast
```

```
## # A tibble: 159 x 4
##    origin week_index average_temperature average_dewpoint
##    <chr>       <dbl>               <dbl>            <dbl>
##  1 EWR             0                34.3             19.4
##  2 EWR             1                42.7             33.3
##  3 EWR             2                39.6             26.5
##  4 EWR             3                21.3              3.8
##  5 EWR             4                36               25.7
##  6 EWR             5                30               17.6
##  7 EWR             6                37.8             25.3
##  8 EWR             7                34.2             21.3
##  9 EWR             8                39.2             27.8
## 10 EWR             9                38.7             26.8
## # i 149 more rows
```

In order to map the first week to an index of 0, we subtract 2244 from the as.numeric(week_index) result since the output for the first week is 2244.

```
# Models for average_temperature

avg_temp_mod1 <- lm(average_temperature ~ poly(week_index, 1),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod1
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 1), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##         (Intercept)  poly(week_index, 1)
##               54.82                51.32
```

```
avg_temp_mod2 <- lm(average_temperature ~ poly(week_index, 2),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod2
```

```
##
## Call:
```

```
## lm(formula = average_temperature ~ poly(week_index, 2), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 2)1  poly(week_index, 2)2
##                54.89                 51.77               -175.10
```

```
avg_temp_mod3 <- lm(average_temperature ~ poly(week_index, 3),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod3
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 3), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 3)1  poly(week_index, 3)2
##                54.92                 51.90               -175.35
## poly(week_index, 3)3
##               -47.92
```

```
avg_temp_mod4 <- lm(average_temperature ~ poly(week_index, 4),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod4
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 4), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 4)1  poly(week_index, 4)2
##                54.93                 51.97               -175.50
## poly(week_index, 4)3  poly(week_index, 4)4
##               -48.07                 65.90
```

```
avg_temp_mod5 <- lm(average_temperature ~ poly(week_index, 5),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod5
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 5), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 5)1  poly(week_index, 5)2
##                54.94                 52.01               -175.57
## poly(week_index, 5)3  poly(week_index, 5)4  poly(week_index, 5)5
##               -48.15                 65.93                 11.05
```

```
avg_temp_mod6 <- lm(average_temperature ~ poly(week_index, 6),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod6
```

```
##
```

```
## Call:
## lm(formula = average_temperature ~ poly(week_index, 6), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 6)1  poly(week_index, 6)2
##              54.9365               52.0108             -175.5711
## poly(week_index, 6)3  poly(week_index, 6)4  poly(week_index, 6)5
##             -48.1492               65.9309               11.0490
## poly(week_index, 6)6
##               0.4191
```

```
avg_temp_mod7 <- lm(average_temperature ~ poly(week_index, 7),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod7
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 7), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 7)1  poly(week_index, 7)2
##              54.9347               52.0007             -175.5514
## poly(week_index, 7)3  poly(week_index, 7)4  poly(week_index, 7)5
##             -48.1286               65.9219               11.0236
## poly(week_index, 7)6  poly(week_index, 7)7
##               0.4147                3.2951
```

```
avg_temp_mod8 <- lm(average_temperature ~ poly(week_index, 8),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod8
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 8), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 8)1  poly(week_index, 8)2
##              54.9322               51.9862             -175.5234
## poly(week_index, 8)3  poly(week_index, 8)4  poly(week_index, 8)5
##             -48.0992               65.9092               10.9873
## poly(week_index, 8)6  poly(week_index, 8)7  poly(week_index, 8)8
##               0.4083                3.3282                6.4227
```

```
avg_temp_mod9 <- lm(average_temperature ~ poly(week_index, 9),
                    data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod9
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 9), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
```

```
##          (Intercept)  poly(week_index, 9)1  poly(week_index, 9)2
##              54.9326               51.9885             -175.5279
## poly(week_index, 9)3  poly(week_index, 9)4  poly(week_index, 9)5
##             -48.1039               65.9112               10.9931
## poly(week_index, 9)6  poly(week_index, 9)7  poly(week_index, 9)8
##               0.4093                3.3229                6.4188
## poly(week_index, 9)9
##               1.2166
```

```
avg_temp_mod10 <- lm(average_temperature ~ poly(week_index, 10),
                     data = weather_tsib_cast, na.action = na.omit)
avg_temp_mod10
```

```
##
## Call:
## lm(formula = average_temperature ~ poly(week_index, 10), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##            (Intercept)   poly(week_index, 10)1   poly(week_index, 10)2
##                54.9249                 51.9440               -175.4420
##  poly(week_index, 10)3   poly(week_index, 10)4   poly(week_index, 10)5
##               -48.0137                 65.8720                 10.8819
##  poly(week_index, 10)6   poly(week_index, 10)7   poly(week_index, 10)8
##                 0.3897                  3.4240                  6.4924
##  poly(week_index, 10)9  poly(week_index, 10)10
##                 1.1545                -13.4193
```

```
# Models for average_dewpoint
avg_dewp_mod1 <- lm(average_dewpoint ~ poly(week_index, 1),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod1
```

```
##
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 1), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##         (Intercept)  poly(week_index, 1)
##               41.05                55.81
```

```
avg_dewp_mod2 <- lm(average_dewpoint ~ poly(week_index, 2),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod2
```

```
##
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 2), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 2)1  poly(week_index, 2)2
##                41.12                 56.24               -170.58
```

```
avg_dewp_mod3 <- lm(average_dewpoint ~ poly(week_index, 3),
                    data = weather_tsib_cast, na.action = na.omit)
```

```
avg_dewp_mod3
```

```
##
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 3), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 3)1  poly(week_index, 3)2
##                41.15                 56.37               -170.83
## poly(week_index, 3)3
##               -48.00
```

```
avg_dewp_mod4 <- lm(average_dewpoint ~ poly(week_index, 4),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod4
```

```
##
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 4), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 4)1  poly(week_index, 4)2
##                41.16                 56.46               -171.01
## poly(week_index, 4)3  poly(week_index, 4)4
##               -48.18                 75.06
```

```
avg_dewp_mod5 <- lm(average_dewpoint ~ poly(week_index, 5),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod5
```

```
##
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 5), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 5)1  poly(week_index, 5)2
##                41.17                 56.52               -171.12
## poly(week_index, 5)3  poly(week_index, 5)4  poly(week_index, 5)5
##               -48.30                 75.11                 17.19
```

```
avg_dewp_mod6 <- lm(average_dewpoint ~ poly(week_index, 6),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod6
```

```
##
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 6), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##          (Intercept)  poly(week_index, 6)1  poly(week_index, 6)2
##               41.170                56.515              -171.110
## poly(week_index, 6)3  poly(week_index, 6)4  poly(week_index, 6)5
##              -48.289                75.106                17.181
## poly(week_index, 6)6
##               -6.971
```

```r
avg_dewp_mod7 <- lm(average_dewpoint ~ poly(week_index, 7),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod7
```

```
## 
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 7), data = weather_tsib_cast, 
##     na.action = na.omit)
## 
## Coefficients:
##          (Intercept)  poly(week_index, 7)1  poly(week_index, 7)2  
##               41.167                56.496              -171.073  
## poly(week_index, 7)3  poly(week_index, 7)4  poly(week_index, 7)5  
##              -48.250                75.089                17.133  
## poly(week_index, 7)6  poly(week_index, 7)7  
##               -6.979                 6.250  
```

```r
avg_dewp_mod8 <- lm(average_dewpoint ~ poly(week_index, 8),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod8
```

```
## 
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 8), data = weather_tsib_cast, 
##     na.action = na.omit)
## 
## Coefficients:
##          (Intercept)  poly(week_index, 8)1  poly(week_index, 8)2  
##               41.164                56.483              -171.047  
## poly(week_index, 8)3  poly(week_index, 8)4  poly(week_index, 8)5  
##              -48.224                75.078                17.100  
## poly(week_index, 8)6  poly(week_index, 8)7  poly(week_index, 8)8  
##               -6.985                 6.280                 5.790  
```

```r
avg_dewp_mod9 <- lm(average_dewpoint ~ poly(week_index, 9),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod9
```

```
## 
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 9), data = weather_tsib_cast, 
##     na.action = na.omit)
## 
## Coefficients:
##          (Intercept)  poly(week_index, 9)1  poly(week_index, 9)2  
##               41.161                56.462              -171.007  
## poly(week_index, 9)3  poly(week_index, 9)4  poly(week_index, 9)5  
##              -48.181                75.060                17.048  
## poly(week_index, 9)6  poly(week_index, 9)7  poly(week_index, 9)8  
##               -6.994                 6.327                 5.824  
## poly(week_index, 9)9  
##              -10.861  
```

```
avg_dewp_mod10 <- lm(average_dewpoint ~ poly(week_index, 10),
                    data = weather_tsib_cast, na.action = na.omit)
avg_dewp_mod10
```

```
##
## Call:
## lm(formula = average_dewpoint ~ poly(week_index, 10), data = weather_tsib_cast,
##     na.action = na.omit)
##
## Coefficients:
##            (Intercept)   poly(week_index, 10)1   poly(week_index, 10)2
##                 41.152                  56.412                -170.911
##  poly(week_index, 10)3   poly(week_index, 10)4   poly(week_index, 10)5
##                -48.080                  75.016                  16.923
##  poly(week_index, 10)6   poly(week_index, 10)7   poly(week_index, 10)8
##                 -7.016                   6.441                   5.907
##  poly(week_index, 10)9  poly(week_index, 10)10
##                -10.931                 -15.096
```

## (2 points) Part-3: Evalute the model fits best for each outcomes

For each of the outcomes – `average_temperature` at the weekly level, and `average_dewpoint` at the weekly level – make an assessment based on either AIC or BIC for why one polynomial degree produces the best fitting model. In doing so, describe why you have chosen to use either AIC or BIC, what the particular scoring of this metric is doing (i.e. write the formula, and explain to your reader what is happening in that formula). Especially compelling in producing your argument for why you prefer a particular model form is to create a plot of the polynomial degree on the x-axis and the metric score on the y-axis.

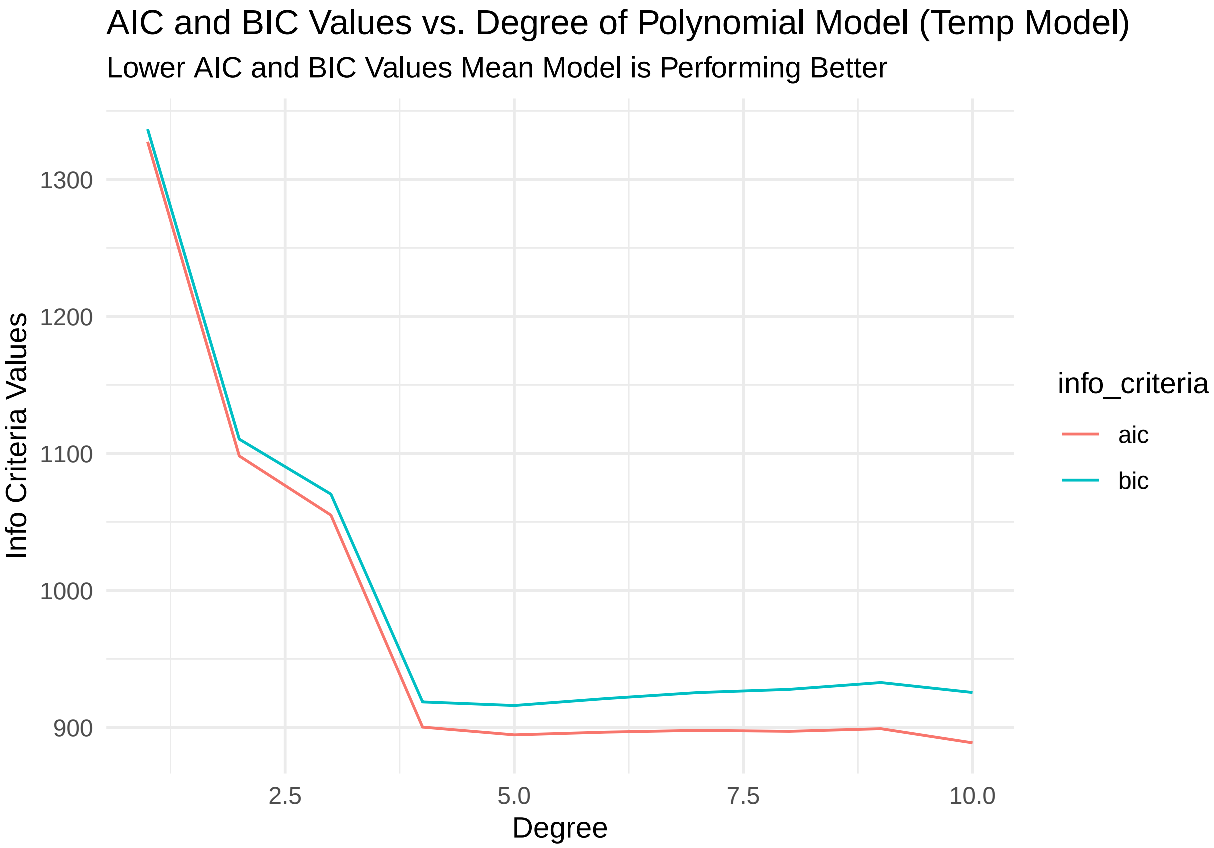
```
model_temp_results_data <- data.frame(
  degree = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
  aic = c(AIC(avg_temp_mod1), AIC(avg_temp_mod2), AIC(avg_temp_mod3),
          AIC(avg_temp_mod4), AIC(avg_temp_mod5),
          AIC(avg_temp_mod6), AIC(avg_temp_mod7), AIC(avg_temp_mod8),
          AIC(avg_temp_mod9), AIC(avg_temp_mod10)),
  bic = c(BIC(avg_temp_mod1), BIC(avg_temp_mod2), BIC(avg_temp_mod3),
          BIC(avg_temp_mod4), BIC(avg_temp_mod5),
          BIC(avg_temp_mod6), BIC(avg_temp_mod7), BIC(avg_temp_mod8),
          BIC(avg_temp_mod9), BIC(avg_temp_mod10))
)
head(model_temp_results_data)
```

```
##   degree       aic       bic
## 1      1 1327.5163 1336.7040
## 2      2 1098.1817 1110.4321
## 3      3 1054.9866 1070.2996
## 4      4  900.2569  918.6324
## 5      5  894.5949  916.0331
## 6      6  896.5835  921.0843
```

```
tidy_temp_data <- pivot_longer(model_temp_results_data,
                               cols = c(aic, bic),
                          names_to = "info_criteria", values_to = "value")
head(tidy_temp_data)
```

```
## # A tibble: 6 x 3
##   degree info_criteria value
##    <dbl> <chr>         <dbl>
## 1      1 aic           1328.
## 2      1 bic           1337.
## 3      2 aic           1098.
## 4      2 bic           1110.
## 5      3 aic           1055.
## 6      3 bic           1070.
```

```
tidy_temp_data %>% ggplot(aes(x=degree, y = value, color= info_criteria)) +
  geom_line() + xlab("Degree") + ylab("Info Criteria Values") +
  ggtitle('AIC and BIC Values vs. Degree of Polynomial Model (Temp Model)') +
  labs(subtitle = "Lower AIC and BIC Values Mean Model is Performing Better")
```



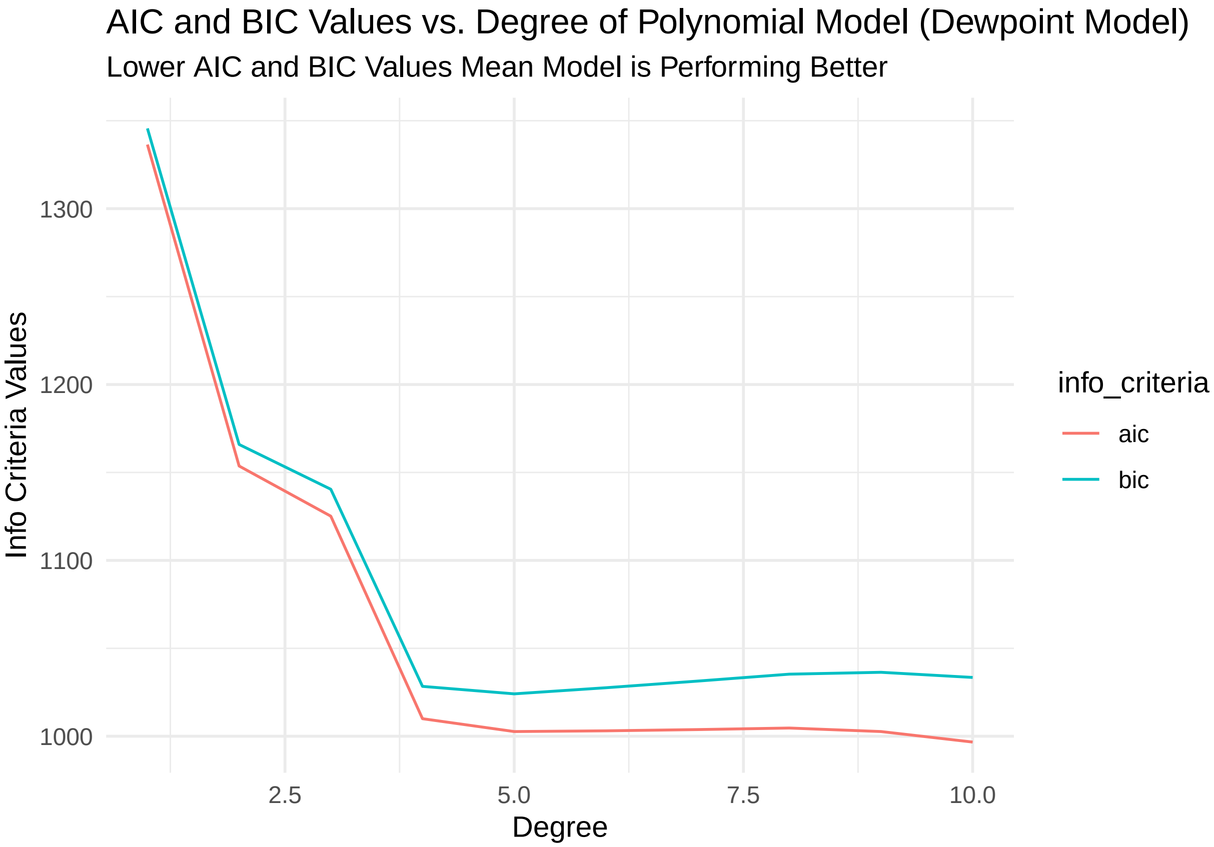
```
model_dewp_results_data <- data.frame(
  degree = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
  aic = c(AIC(avg_dewp_mod1), AIC(avg_dewp_mod2), AIC(avg_dewp_mod3),
          AIC(avg_dewp_mod4), AIC(avg_dewp_mod5),
          AIC(avg_dewp_mod6), AIC(avg_dewp_mod7), AIC(avg_dewp_mod8),
          AIC(avg_dewp_mod9), AIC(avg_dewp_mod10)),
  bic = c(BIC(avg_dewp_mod1), BIC(avg_dewp_mod2), BIC(avg_dewp_mod3),
          BIC(avg_dewp_mod4), BIC(avg_dewp_mod5),
          BIC(avg_dewp_mod6), BIC(avg_dewp_mod7), BIC(avg_dewp_mod8),
          BIC(avg_dewp_mod9), BIC(avg_dewp_mod10))
)
head(model_dewp_results_data)
```

```
##   degree      aic      bic
## 1      1 1336.453 1345.641
## 2      2 1153.672 1165.922
## 3      3 1125.140 1140.453
## 4      4 1009.987 1028.363
## 5      5 1002.676 1024.115
## 6      6 1003.079 1027.579
```

```
tidy_dewp_data <- pivot_longer(model_dewp_results_data,
                             cols = c(aic, bic),
                          names_to = "info_criteria", values_to = "value")
head(tidy_dewp_data)
```

```
## # A tibble: 6 x 3
##   degree info_criteria value
##    <dbl> <chr>         <dbl>
## 1      1 aic           1336.
## 2      1 bic           1346.
## 3      2 aic           1154.
## 4      2 bic           1166.
## 5      3 aic           1125.
## 6      3 bic           1140.
```

```
tidy_dewp_data %>% ggplot(aes(x=degree, y = value, color= info_criteria)) +
  geom_line() + xlab("Degree") + ylab("Info Criteria Values") +
  ggtitle('AIC and BIC Values vs. Degree of Polynomial Model (Dewpoint Model)') +
  labs(subtitle = "Lower AIC and BIC Values Mean Model is Performing Better")
```



Looking at these two BIC scoring criteria there seems to be a clear **lack** of improvement beyond a polynomial order of four. *Perhaps* moving from four to five would still increase the model's performance, but it is small compared to the polynomials 2-4. For us, if we were fitting this model, we would be likely to stop at `poly( , degree = 4)`.

Based on the plots for both average_temperature and average_dewpoint, we would use the AIC to help decide the best fitting polynomial model. While the BIC does not really improve beyond degree 4, the AIC drops slightly below 900 for average_temperature and slightly below 1000 for average_dewpoint, the lowest points for the AIC for both models respectively. Hence, the AIC helps differentiate which degree polynomial is better when the BIC fails to improve. Based on the plots, if we were to use the AIC, we would choose the polynomial of degree 10 since the model for both average_temperature and average_depoint corresponds to the lowest AIC values.

The AIC consists of the negative log likelihood and a term proportional to the number of parameters in the model. A lower AIC means that the log likelihood is large compared to the effect of the number of parameters on the AIC value, indicating that the log likelihood is larger for lower AIC values. Since the degree 10 polynomial corresponds to the lowest AIC values for both average_temperature and average_dewpoint, the degree 10 polynomial is the best fitting model.

# Question-3: Smooth Moves

In the async lecture, Jeffrey proposes four different smoothers that might be used:

1. **Moving Average**: These moving average smoothers can be either symmetric or, often preferred, backward smoothers. Please use a backward smoother, and make the choice about the number of periods based off of some evaluation of different choices. You might consult [this page] in *Forecasting Principles and Practice 3*.
2. **Regression Smoothers**: Please use the polynomial regression that you stated you most preferred from your BIC analysis to the last question.
3. (Optional) **Spline Smoothers**: There is a reading in the repository that provides some more background (it is a review from 2019) on using spline smoothers. The current implementation that we prefer in R is the `splines2` library. For your spline smoother, use the `splines2::naturalSpline` function. Once you have fitted this spline, you can use the `predict` method to produce values. A good starting place for this is [here]. We'll note that this is the most challenging of the smoothers to get running in this assignment, and so getting it running successfully is optional.
4. **Kernel Smoothers**.: Please use the `ksmooth` function that is available to you in the `stats` library. Because `stats` is always loaded in R, we have not referred to it using the `::` notation.

## (6 points, with 2 optional) Part-1: Create Smoothers

With the weekly weather data that you used for the previous question, produce a smoothed variable for `average_temperature` and `average_dewpoint` using each of the four smoothers described in the async. Three smoothers are required of this question – (1) Moving Average; (2) Regression Smoothers; and, (3) Kernel Smoothers. The fourth, splines, is optional but if you produce a spline smoother that is working effectively, you can earn two bonus points. (Note that the homework maximum score is still 100%.)

When you are done with this task, you should have created eight new variables that are each a smoothed version of this series.

For each smoother that you produce:

- Fit the smoother **within** each origin. That is, fit the smoother for JFK separately from LaGuardia and Newark.
- Attach the values that are produced by the smoother onto the `weekly_weather` dataframe.
- Produce a plot that shows the original data as `geom_point()`, and the smoother's predictions as `geom_line()`.

- Your goal is not to produce **any** smoother, but instead, for each class of smoother, the version that is doint the best job that is possible by this smoother. That is, you are working through the hyperparametrs to these algorithms to produce their most effective output.

```
install.packages('slider')
```

```
## Installing package into '/usr/local/lib/R/site-library'
## (as 'lib' is unspecified)
```

```
library(slider)
```

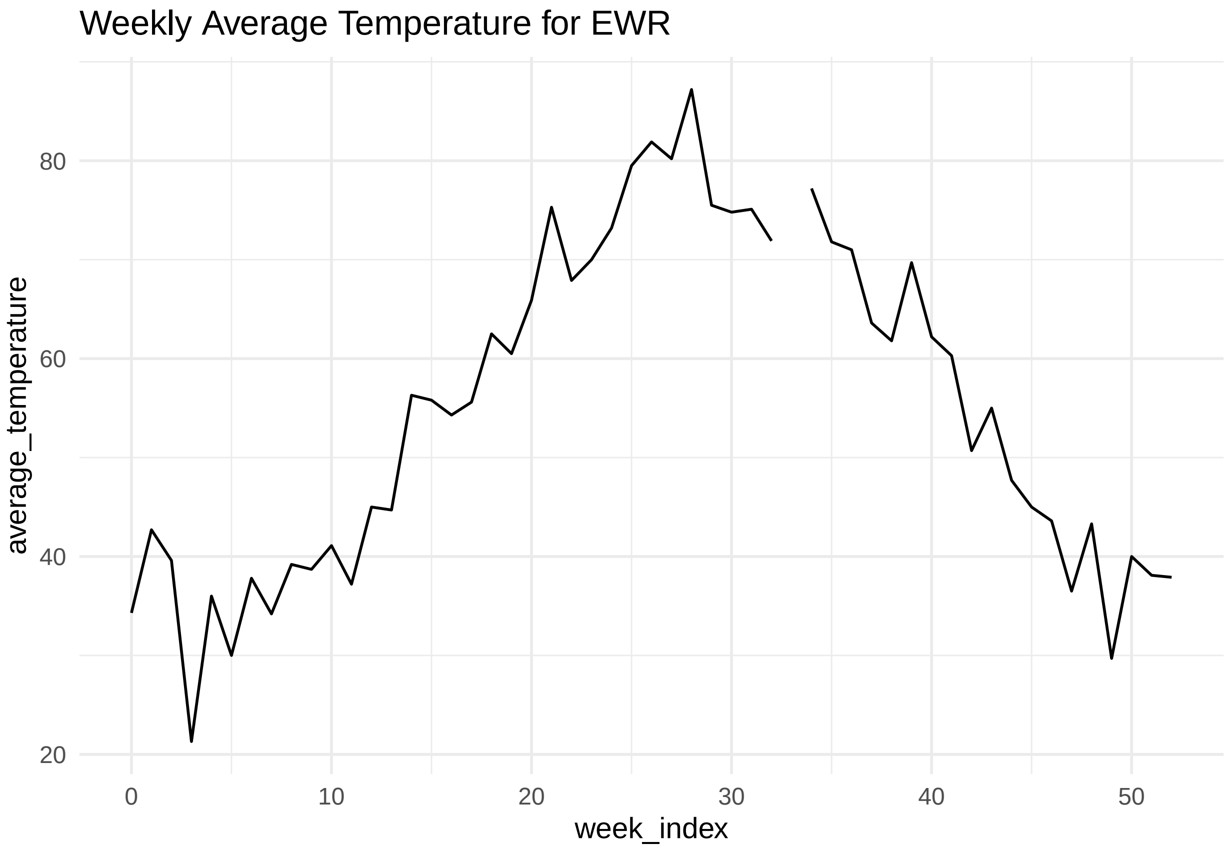
```
weekly_weather <- weather_weekly
weekly_weather$week_index <- as.numeric(weekly_weather$week_index) - 2244
head(weekly_weather)
```

```
## # A tibble: 6 x 4
##   origin week_index average_temperature average_dewpoint
##   <chr>       <dbl>               <dbl>            <dbl>
## 1 EWR             0                34.3             19.4
## 2 EWR             1                42.7             33.3
## 3 EWR             2                39.6             26.5
## 4 EWR             3                21.3              3.8
## 5 EWR             4                36               25.7
## 6 EWR             5                30               17.6
```

```
unique(weekly_weather$origin)
```

```
## [1] "EWR" "JFK" "LGA"
```

```
weekly_weather %>% filter(origin == 'EWR') %>%
  ggplot(aes(x=week_index, average_temperature)) +
  geom_line() + ggtitle("Weekly Average Temperature for EWR")
```



```
ewr_ma_temp <- weekly_weather %>%
  filter(origin == 'EWR') %>%
  mutate(
    backwards_ma = slider::slide_dbl(average_temperature, mean,
                .before = 1, .after = 0, .complete = TRUE)
  )
ewr_ma_temp
```

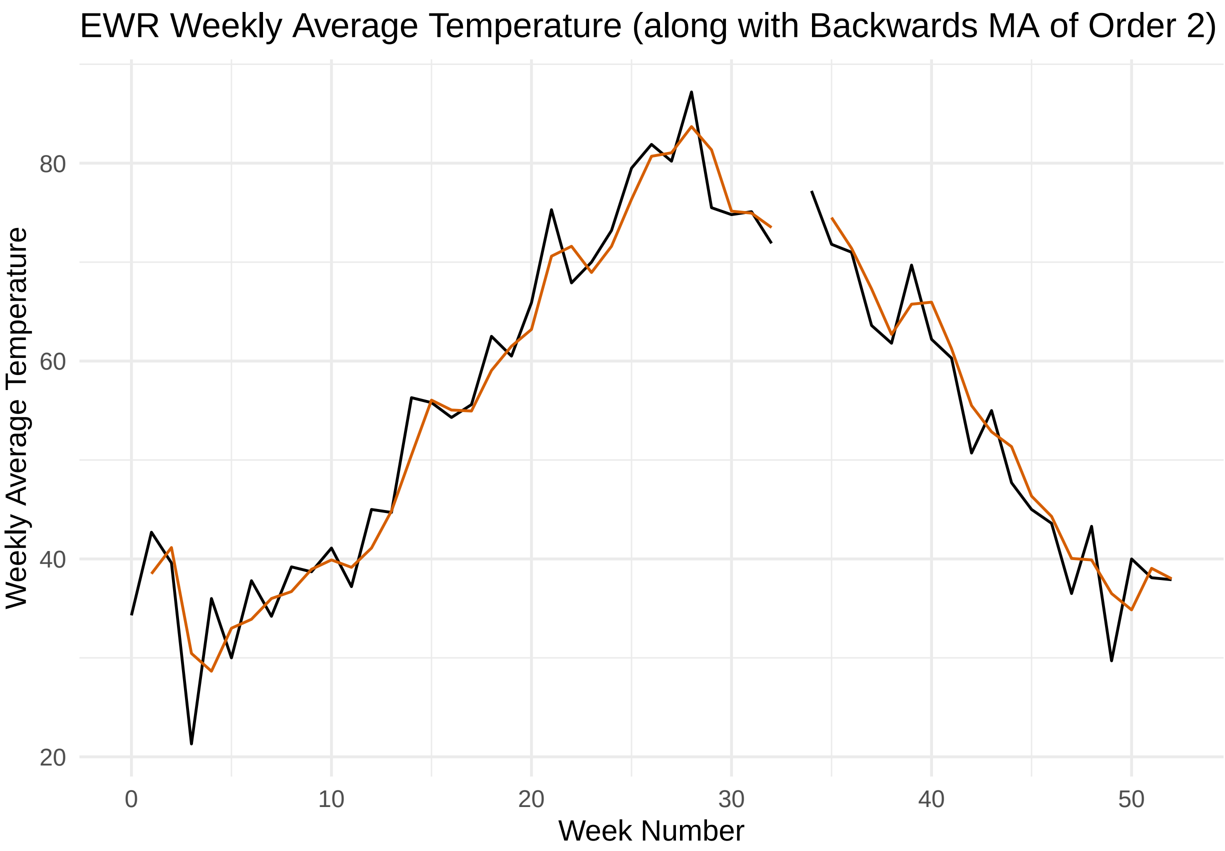
```
## # A tibble: 53 x 5
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 EWR             0                34.3             19.4         NA
##  2 EWR             1                42.7             33.3         38.5
##  3 EWR             2                39.6             26.5         41.2
##  4 EWR             3                21.3              3.8         30.4
##  5 EWR             4                36               25.7         28.6
##  6 EWR             5                30               17.6         33
##  7 EWR             6                37.8             25.3         33.9
##  8 EWR             7                34.2             21.3         36
##  9 EWR             8                39.2             27.8         36.7
## 10 EWR             9                38.7             26.8         39.0
## # i 43 more rows
```

```
ewr_ma_temp <- ewr_ma_temp %>% as_tsibble(index=week_index, key=origin)
ewr_ma_temp
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 EWR             0                34.3             19.4         NA
##  2 EWR             1                42.7             33.3         38.5
##  3 EWR             2                39.6             26.5         41.2
##  4 EWR             3                21.3              3.8         30.4
##  5 EWR             4                36               25.7         28.6
##  6 EWR             5                30               17.6         33
##  7 EWR             6                37.8             25.3         33.9
##  8 EWR             7                34.2             21.3         36
##  9 EWR             8                39.2             27.8         36.7
## 10 EWR             9                38.7             26.8         39.0
## # i 43 more rows
```

```
ewr_ma_temp |>
  autoplot(average_temperature) +
  geom_line(aes(y = backwards_ma), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "EWR Weekly Average Temperature (along with Backwards MA of Order 2)")
```

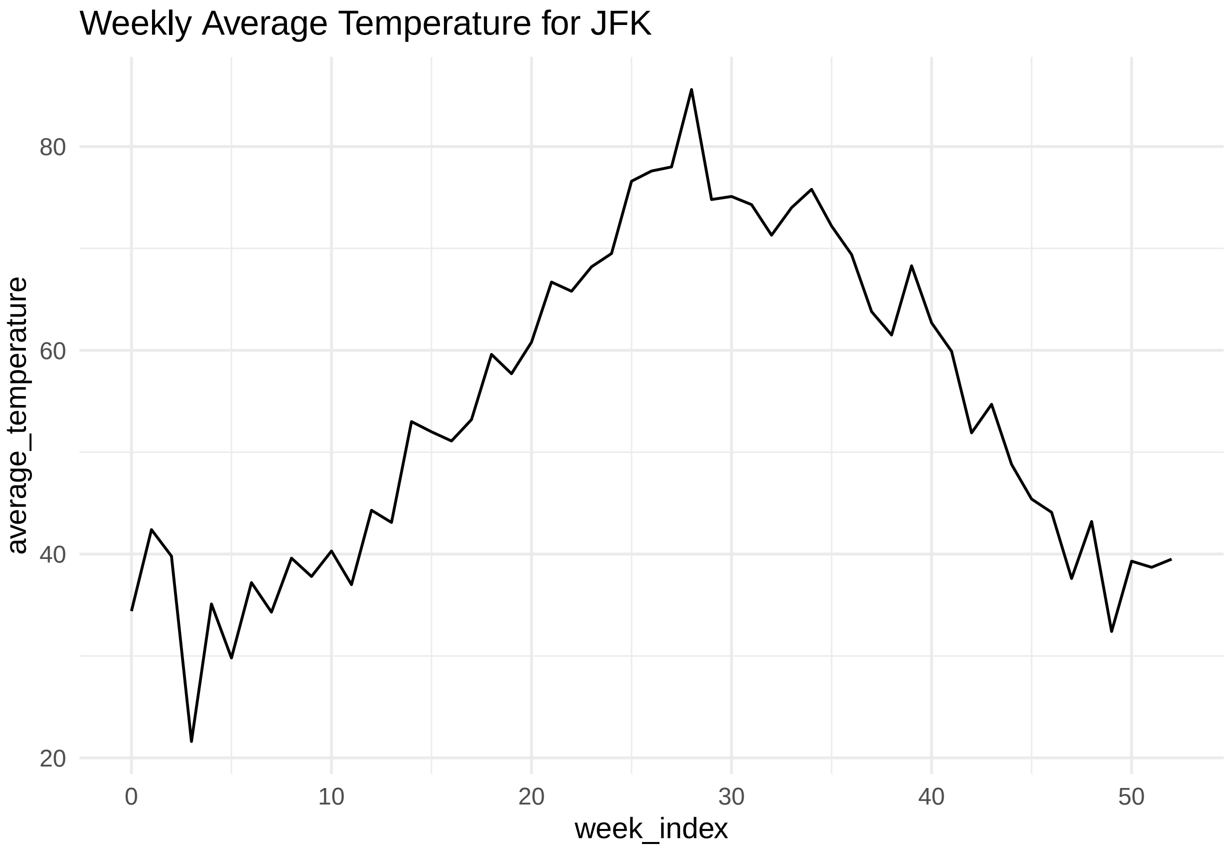
```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



```
unique(weekly_weather$origin)
```

```
## [1] "EWR" "JFK" "LGA"
```

```
weekly_weather %>% filter(origin == 'JFK') %>%
  ggplot(aes(x=week_index, average_temperature)) +
  geom_line() + ggtitle("Weekly Average Temperature for JFK")
```



```
jfk_ma_temp <- weekly_weather %>%
  filter(origin == 'JFK') %>%
  mutate(
    backwards_ma = slider::slide_dbl(average_temperature, mean,
                .before = 1, .after = 0, .complete = TRUE)
  )
jfk_ma_temp
```

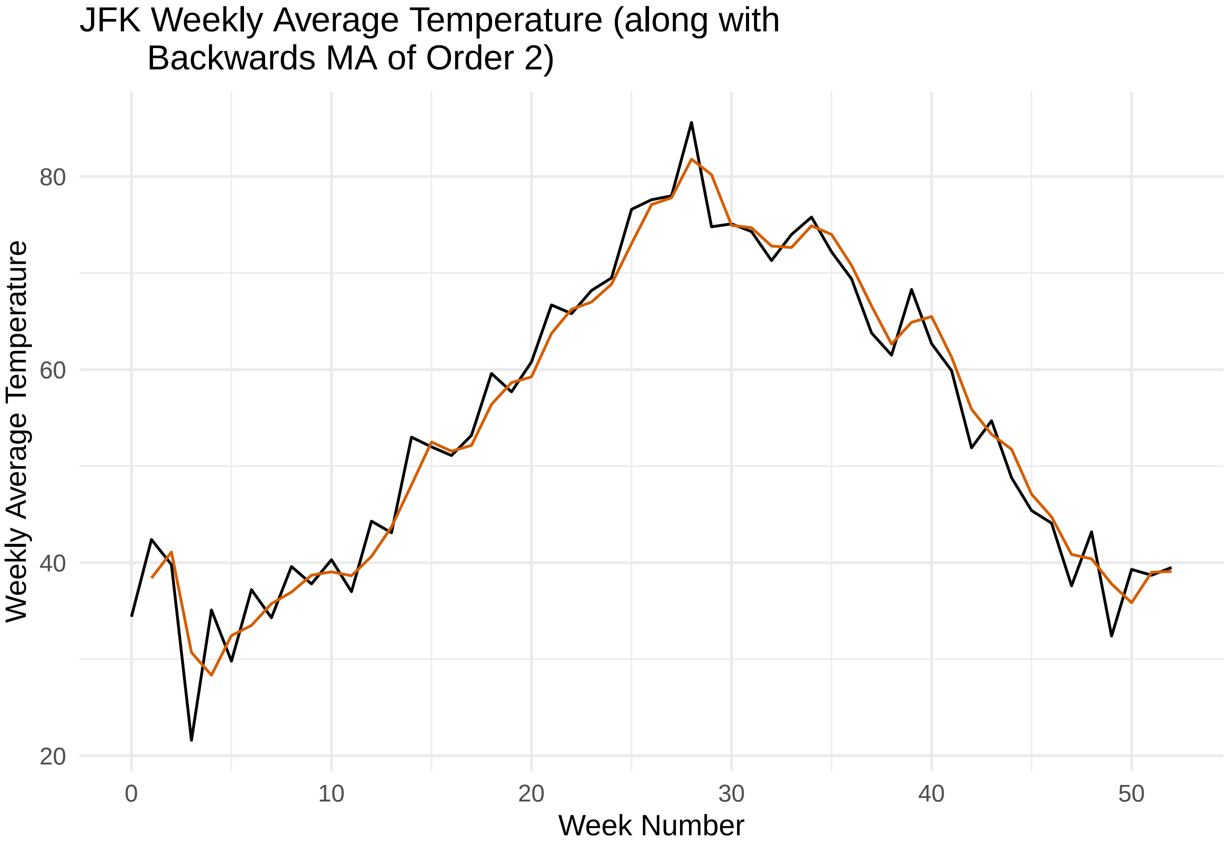
```
## # A tibble: 53 x 5
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 JFK             0                34.4             19.2         NA
##  2 JFK             1                42.4             33.3         38.4
##  3 JFK             2                39.8             26.3         41.1
##  4 JFK             3                21.6              3.2         30.7
##  5 JFK             4                35.1             24.7         28.4
##  6 JFK             5                29.8             16.9         32.4
##  7 JFK             6                37.2             25.2         33.5
##  8 JFK             7                34.3             21           35.8
##  9 JFK             8                39.6             27.7         37.0
## 10 JFK             9                37.8             26.2         38.7
## # i 43 more rows
```

```
jfk_ma_temp <- jfk_ma_temp %>% as_tsibble(index=week_index, key=origin)
jfk_ma_temp
```

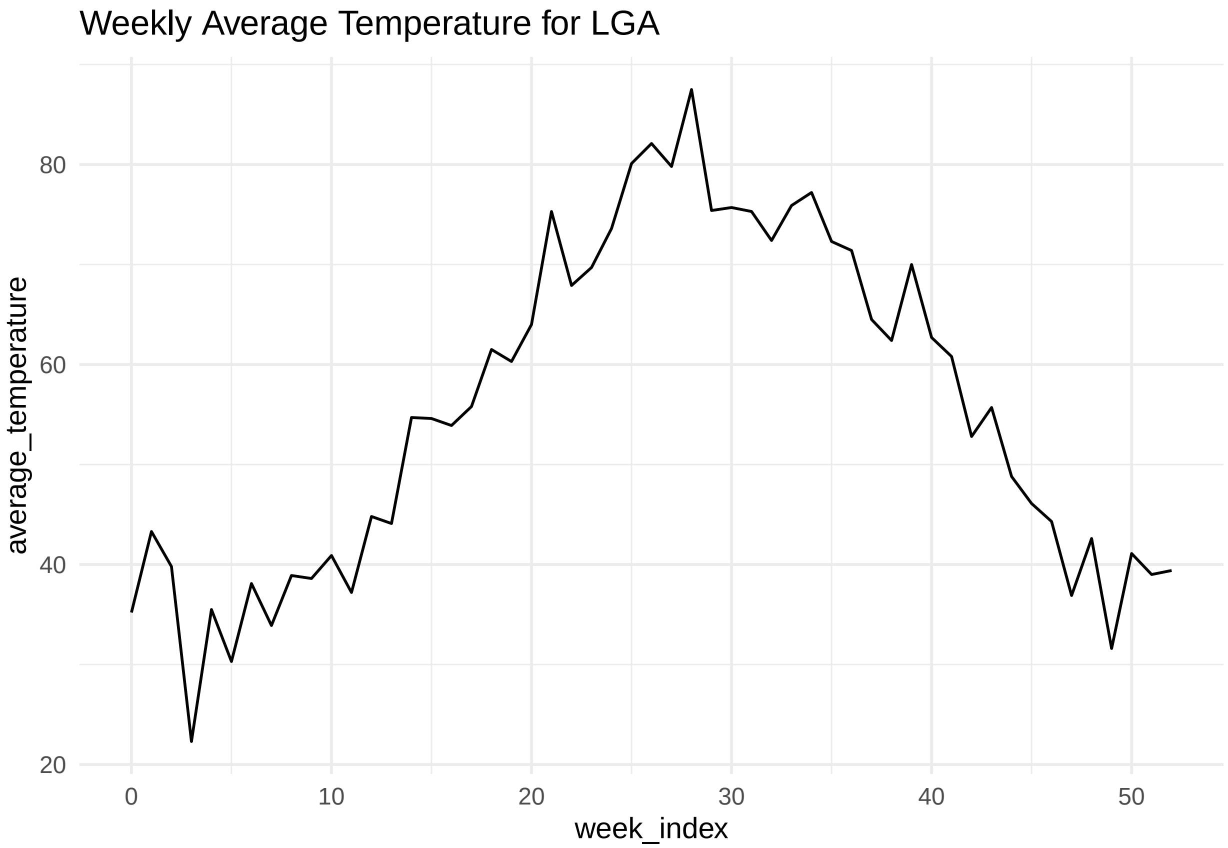
```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 JFK             0                34.4             19.2         NA  
##  2 JFK             1                42.4             33.3         38.4
##  3 JFK             2                39.8             26.3         41.1
##  4 JFK             3                21.6              3.2         30.7
##  5 JFK             4                35.1             24.7         28.4
##  6 JFK             5                29.8             16.9         32.4
##  7 JFK             6                37.2             25.2         33.5
##  8 JFK             7                34.3             21           35.8
##  9 JFK             8                39.6             27.7         37.0
## 10 JFK             9                37.8             26.2         38.7
## # i 43 more rows
```

```
jfk_ma_temp |>
  autoplot(average_temperature) +
  geom_line(aes(y = backwards_ma), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "JFK Weekly Average Temperature (along with
       Backwards MA of Order 2)")
```

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



```
weekly_weather %>% filter(origin == 'LGA') %>%
  ggplot(aes(x=week_index, average_temperature)) + geom_line() +
  ggtitle("Weekly Average Temperature for LGA")
```



```
lga_ma_temp <- weekly_weather %>%
  filter(origin == 'LGA') %>%
  mutate(
    backwards_ma = slider::slide_dbl(average_temperature, mean,
                .before = 1, .after = 0, .complete = TRUE)
  )
lga_ma_temp
```

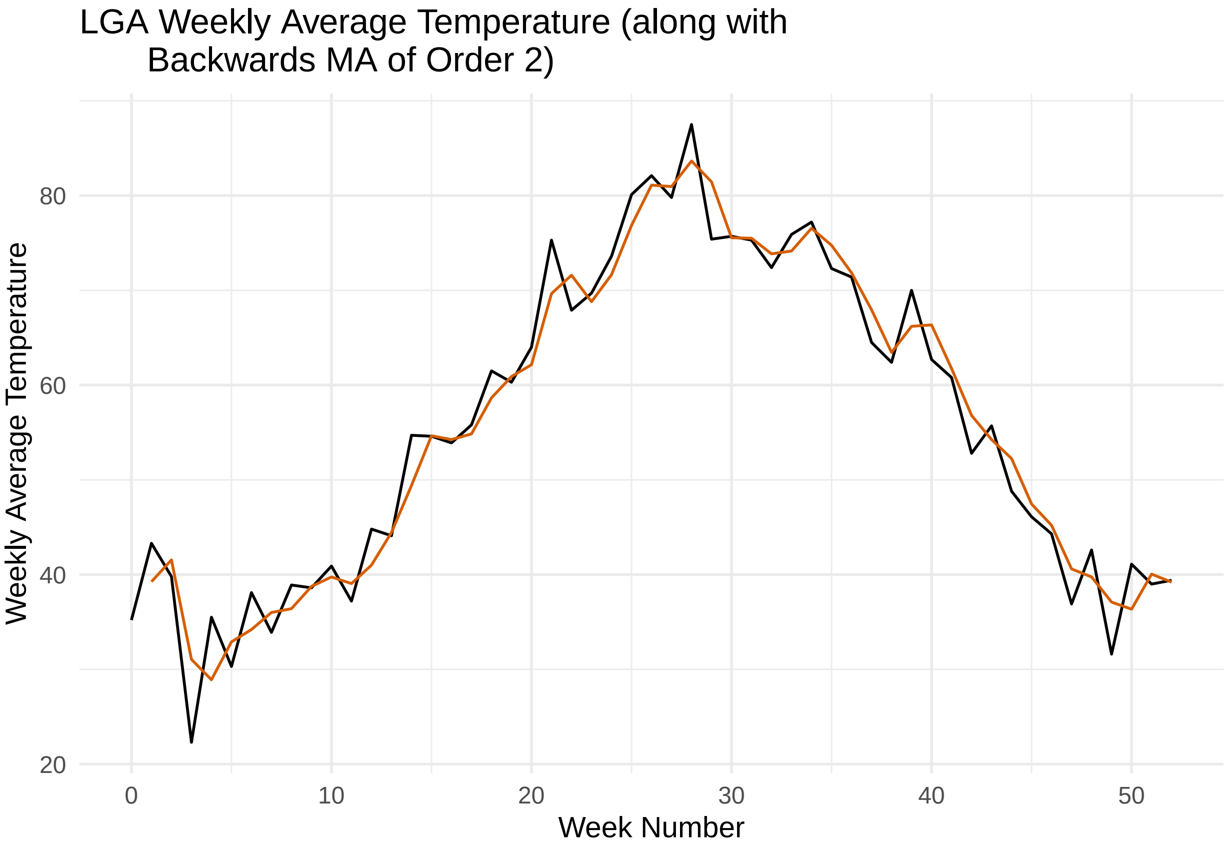
```
## # A tibble: 53 x 5
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 LGA             0                35.2             18.8         NA
##  2 LGA             1                43.3             32.3         39.2
##  3 LGA             2                39.8             25.3         41.6
##  4 LGA             3                22.3              5           31.0
##  5 LGA             4                35.5             23.9         28.9
##  6 LGA             5                30.3             16.6         32.9
##  7 LGA             6                38.1             25           34.2
##  8 LGA             7                33.9             20.2         36
##  9 LGA             8                38.9             27.4         36.4
## 10 LGA             9                38.6             25.9         38.8
## # i 43 more rows
```

```
lga_ma_temp <- lga_ma_temp %>% as_tsibble(index=week_index, key=origin)
lga_ma_temp
```

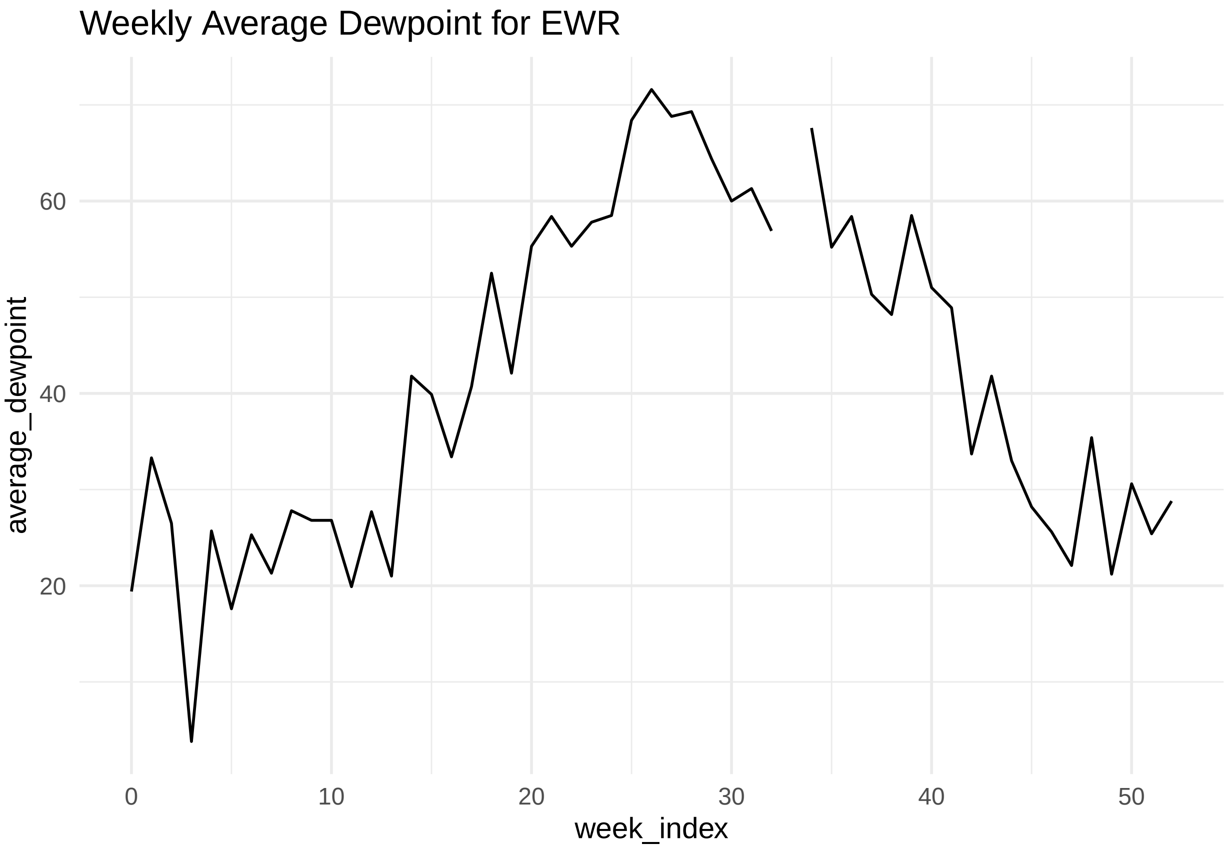
```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 LGA             0                35.2             18.8         NA
##  2 LGA             1                43.3             32.3         39.2
##  3 LGA             2                39.8             25.3         41.6
##  4 LGA             3                22.3              5           31.0
##  5 LGA             4                35.5             23.9         28.9
##  6 LGA             5                30.3             16.6         32.9
##  7 LGA             6                38.1             25           34.2
##  8 LGA             7                33.9             20.2         36
##  9 LGA             8                38.9             27.4         36.4
## 10 LGA             9                38.6             25.9         38.8
## # i 43 more rows
```

```
lga_ma_temp |>
  autoplot(average_temperature) +
  geom_line(aes(y = backwards_ma), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "LGA Weekly Average Temperature (along with
       Backwards MA of Order 2)")
```

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



```
weekly_weather %>% filter(origin == 'EWR') %>%
  ggplot(aes(x=week_index, average_dewpoint)) + geom_line() +
  ggtitle("Weekly Average Dewpoint for EWR")
```



```
ewr_ma_dewp <- weekly_weather %>%
  filter(origin == 'EWR') %>%
  mutate(
    backwards_ma = slider::slide_dbl(average_dewpoint, mean,
                 .before = 1, .after = 0, .complete = TRUE)
  )
ewr_ma_dewp
```

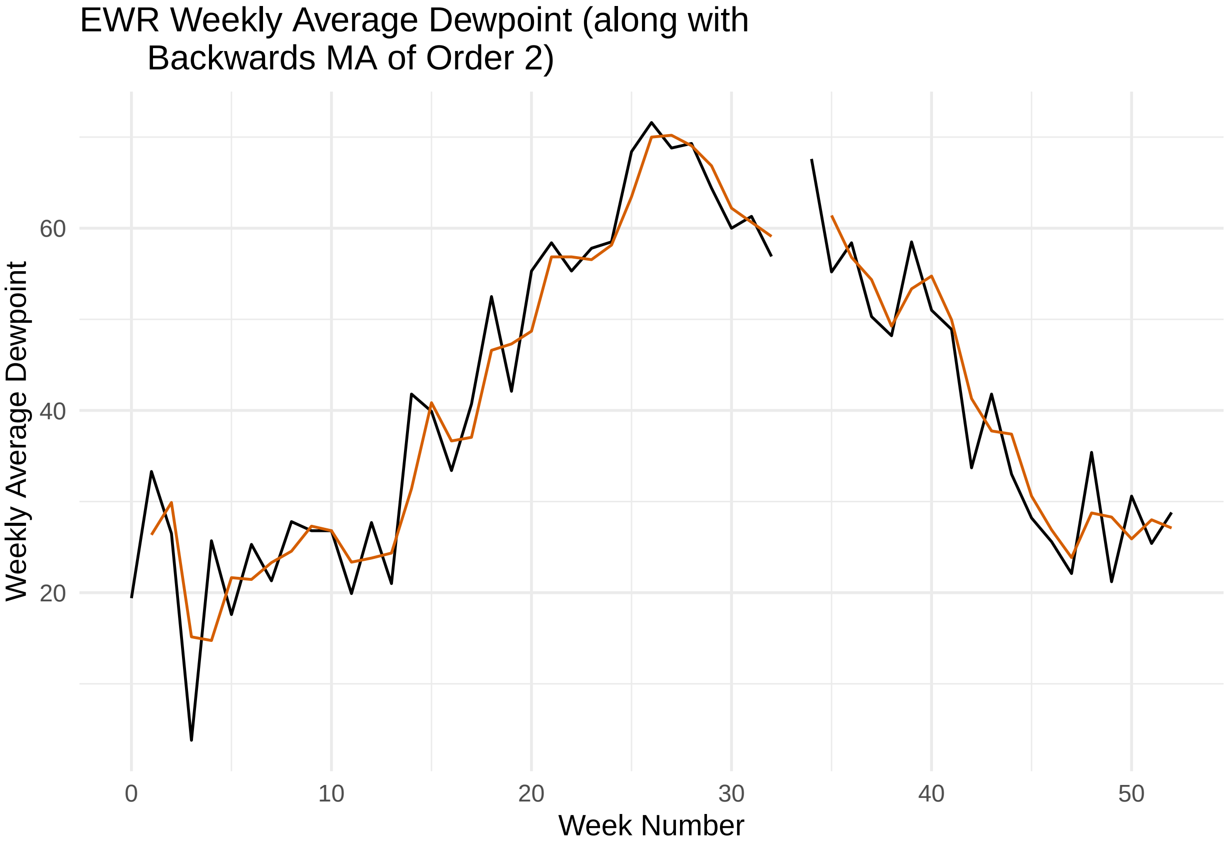
```
## # A tibble: 53 x 5
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 EWR             0                34.3             19.4         NA
##  2 EWR             1                42.7             33.3         26.4
##  3 EWR             2                39.6             26.5         29.9
##  4 EWR             3                21.3              3.8         15.2
##  5 EWR             4                36               25.7         14.8
##  6 EWR             5                30               17.6         21.6
##  7 EWR             6                37.8             25.3         21.5
##  8 EWR             7                34.2             21.3         23.3
##  9 EWR             8                39.2             27.8         24.6
## 10 EWR             9                38.7             26.8         27.3
## # i 43 more rows
```

```
ewr_ma_dewp <- ewr_ma_dewp %>% as_tsibble(index=week_index, key=origin)
ewr_ma_dewp
```

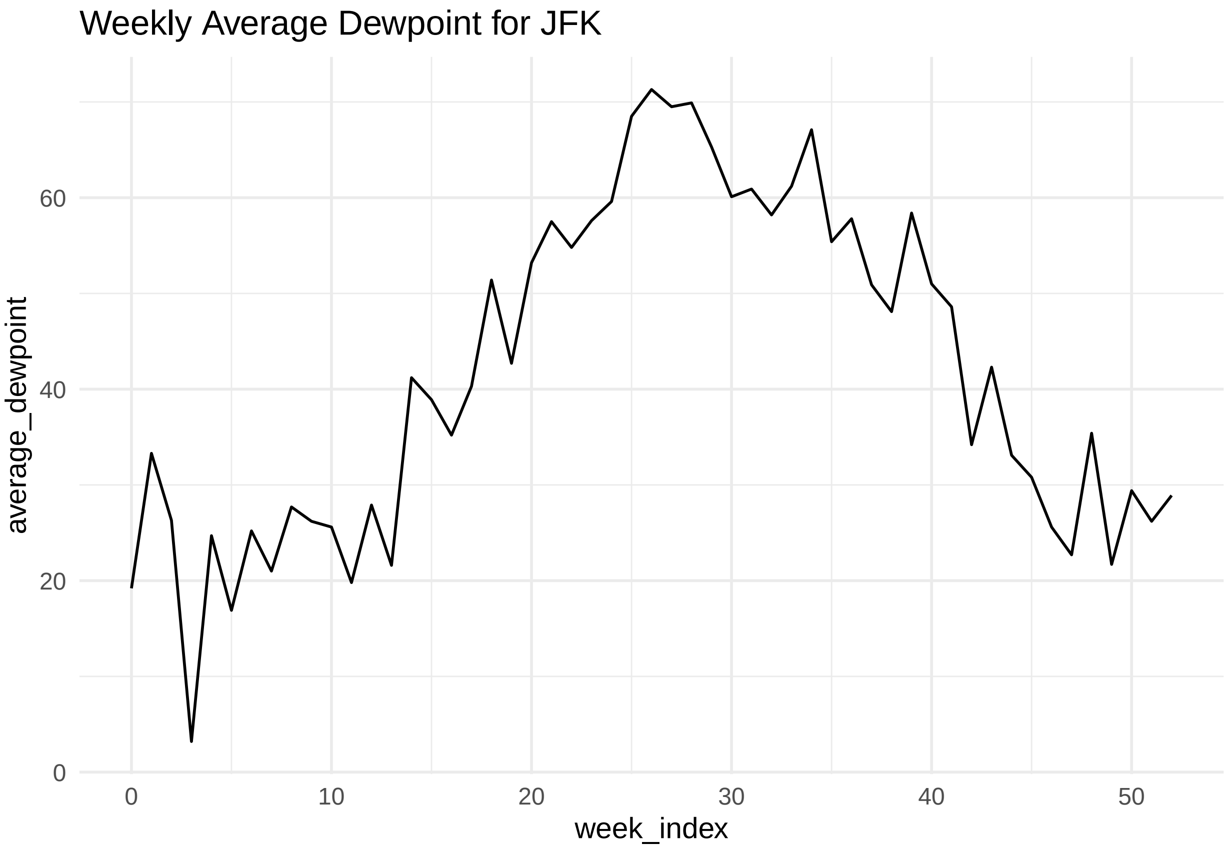
```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 EWR             0                34.3             19.4         NA
##  2 EWR             1                42.7             33.3         26.4
##  3 EWR             2                39.6             26.5         29.9
##  4 EWR             3                21.3              3.8         15.2
##  5 EWR             4                36               25.7         14.8
##  6 EWR             5                30               17.6         21.6
##  7 EWR             6                37.8             25.3         21.5
##  8 EWR             7                34.2             21.3         23.3
##  9 EWR             8                39.2             27.8         24.6
## 10 EWR             9                38.7             26.8         27.3
## # i 43 more rows
```

```
ewr_ma_dewp |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = backwards_ma), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "EWR Weekly Average Dewpoint (along with
       Backwards MA of Order 2)")
```

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



```
weekly_weather %>% filter(origin == 'JFK') %>%
  ggplot(aes(x=week_index, average_dewpoint)) + geom_line() +
  ggtitle("Weekly Average Dewpoint for JFK")
```



```
jfk_ma_dewp <- weekly_weather %>%
  filter(origin == 'JFK') %>%
  mutate(
    backwards_ma = slider::slide_dbl(average_dewpoint, mean,
                .before = 1, .after = 0, .complete = TRUE)
  )
jfk_ma_dewp
```

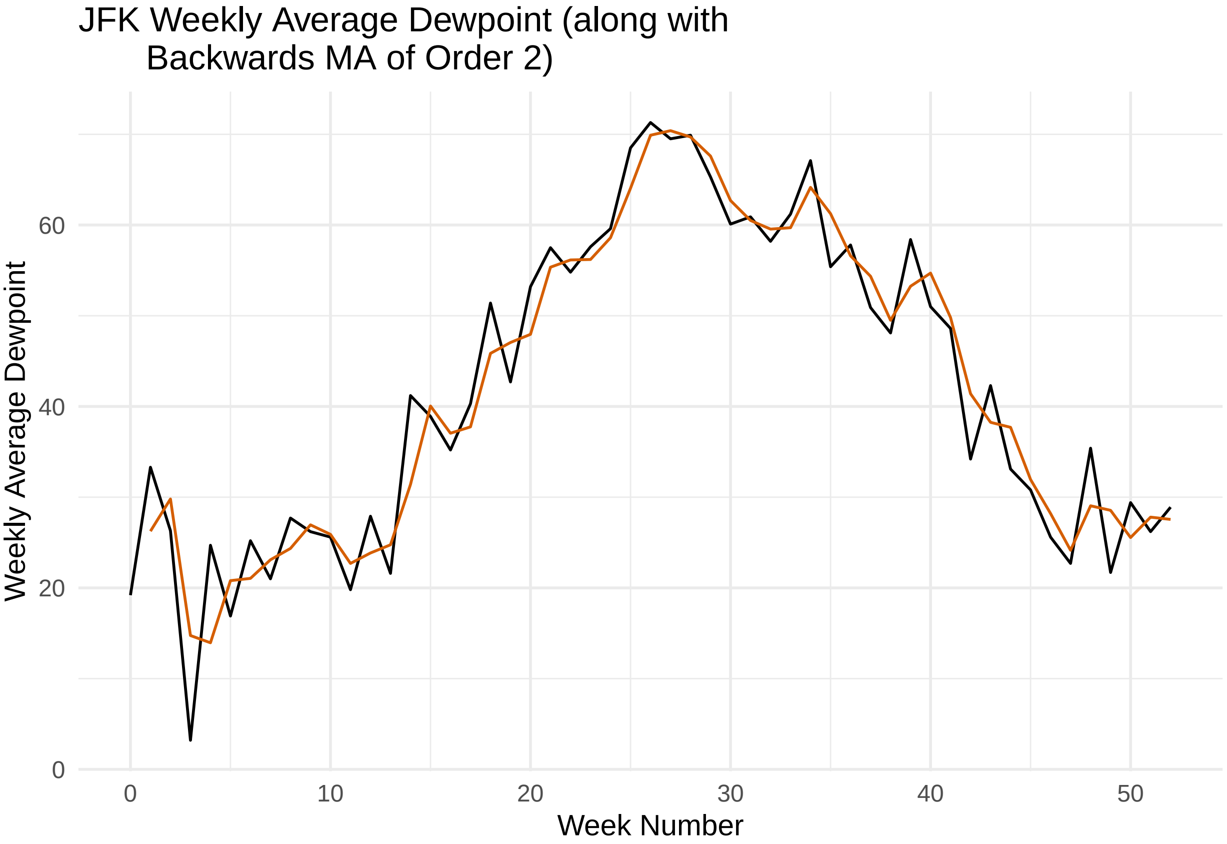
```
## # A tibble: 53 x 5
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 JFK             0                34.4             19.2         NA
##  2 JFK             1                42.4             33.3         26.2
##  3 JFK             2                39.8             26.3         29.8
##  4 JFK             3                21.6              3.2         14.8
##  5 JFK             4                35.1             24.7         14.0
##  6 JFK             5                29.8             16.9         20.8
##  7 JFK             6                37.2             25.2         21.0
##  8 JFK             7                34.3             21           23.1
##  9 JFK             8                39.6             27.7         24.4
## 10 JFK             9                37.8             26.2         27.0
## # i 43 more rows
```

```
jfk_ma_dewp <- jfk_ma_dewp %>% as_tsibble(index=week_index, key=origin)
jfk_ma_dewp
```

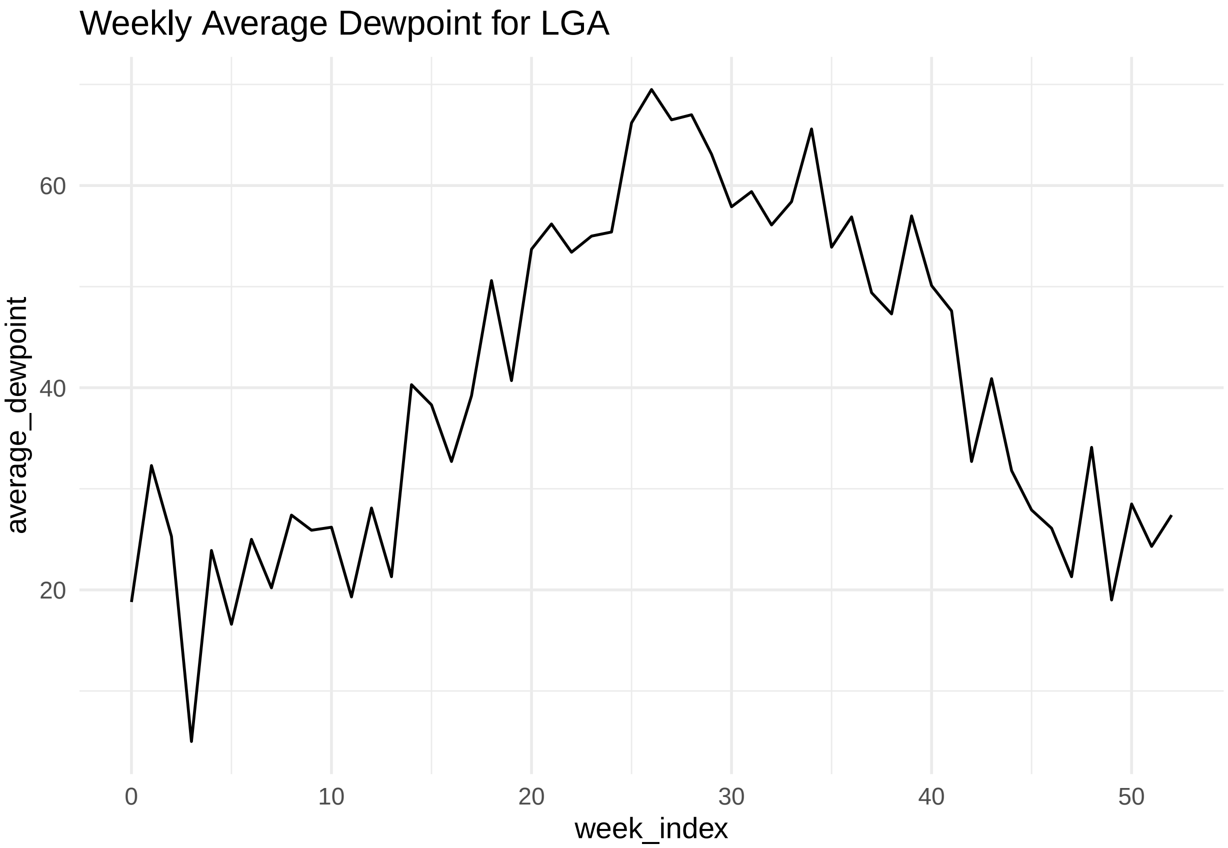
```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 JFK             0                34.4             19.2         NA  
##  2 JFK             1                42.4             33.3         26.2
##  3 JFK             2                39.8             26.3         29.8
##  4 JFK             3                21.6              3.2         14.8
##  5 JFK             4                35.1             24.7         14.0
##  6 JFK             5                29.8             16.9         20.8
##  7 JFK             6                37.2             25.2         21.0
##  8 JFK             7                34.3             21           23.1
##  9 JFK             8                39.6             27.7         24.4
## 10 JFK             9                37.8             26.2         27.0
## # i 43 more rows
```

```
jfk_ma_dewp |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = backwards_ma), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "JFK Weekly Average Dewpoint (along with
       Backwards MA of Order 2)")
```

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



```
weekly_weather %>% filter(origin == 'LGA') %>%
  ggplot(aes(x=week_index, average_dewpoint)) + geom_line() +
  ggtitle("Weekly Average Dewpoint for LGA")
```



```
lga_ma_dewp <- weekly_weather %>%
  filter(origin == 'LGA') %>%
  mutate(
    backwards_ma = slider::slide_dbl(average_dewpoint, mean,
                  .before = 1, .after = 0, .complete = TRUE)
  )
lga_ma_dewp
```

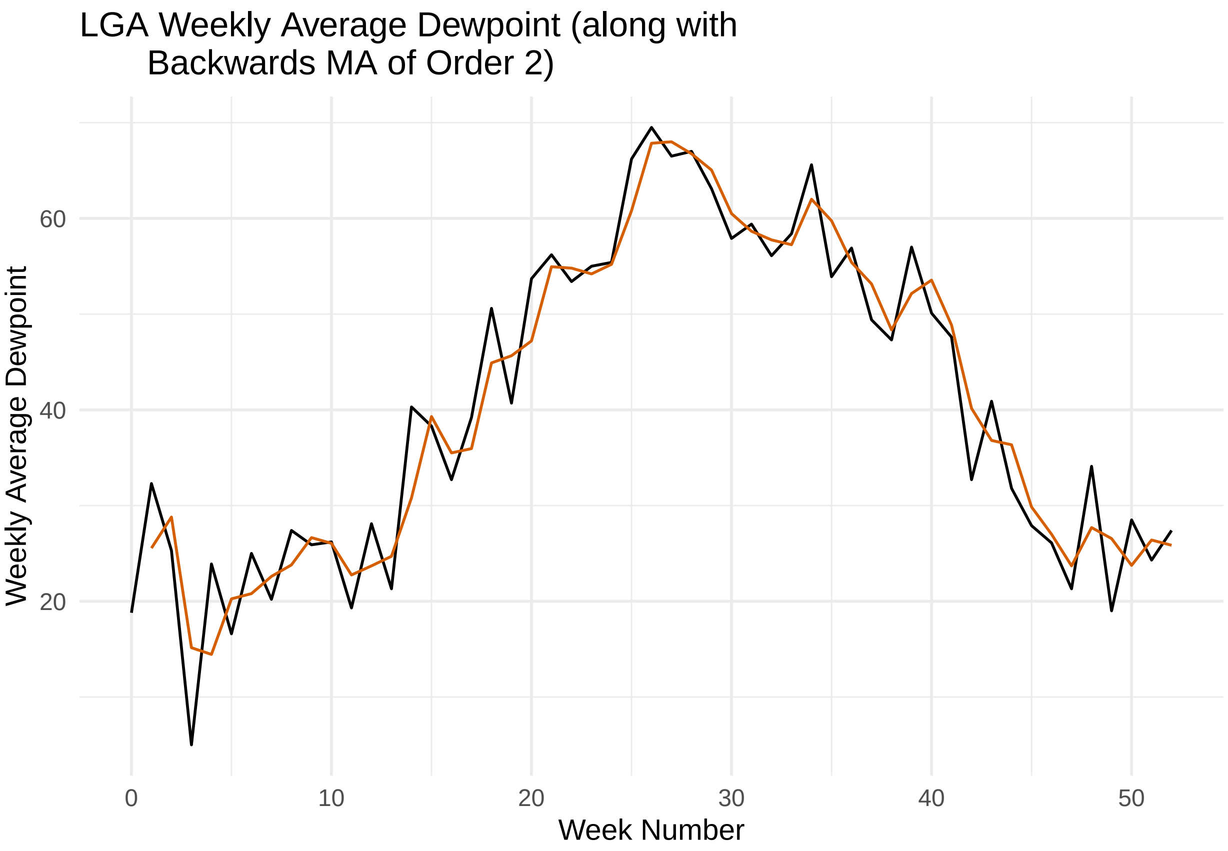
```
## # A tibble: 53 x 5
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 LGA             0                35.2             18.8         NA  
##  2 LGA             1                43.3             32.3         25.5
##  3 LGA             2                39.8             25.3         28.8
##  4 LGA             3                22.3              5           15.2
##  5 LGA             4                35.5             23.9         14.4
##  6 LGA             5                30.3             16.6         20.2
##  7 LGA             6                38.1             25           20.8
##  8 LGA             7                33.9             20.2         22.6
##  9 LGA             8                38.9             27.4         23.8
## 10 LGA             9                38.6             25.9         26.6
## # i 43 more rows
```

```
lga_ma_dewp <- lga_ma_dewp %>% as_tsibble(index=week_index, key=origin)
lga_ma_dewp
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint backwards_ma
##    <chr>       <dbl>               <dbl>            <dbl>        <dbl>
##  1 LGA             0                35.2             18.8         NA  
##  2 LGA             1                43.3             32.3         25.5
##  3 LGA             2                39.8             25.3         28.8
##  4 LGA             3                22.3              5           15.2
##  5 LGA             4                35.5             23.9         14.4
##  6 LGA             5                30.3             16.6         20.2
##  7 LGA             6                38.1             25           20.8
##  8 LGA             7                33.9             20.2         22.6
##  9 LGA             8                38.9             27.4         23.8
## 10 LGA             9                38.6             25.9         26.6
## # i 43 more rows
```

```
lga_ma_dewp |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = backwards_ma), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "LGA Weekly Average Dewpoint (along with
       Backwards MA of Order 2)")
```

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



```
ewr_data <- weekly_weather %>%
  filter(origin == 'EWR')
ewr_reg_smoother <- lm(average_temperature ~ poly(week_index, 10),
                       data = ewr_data, na.action = na.omit)
```

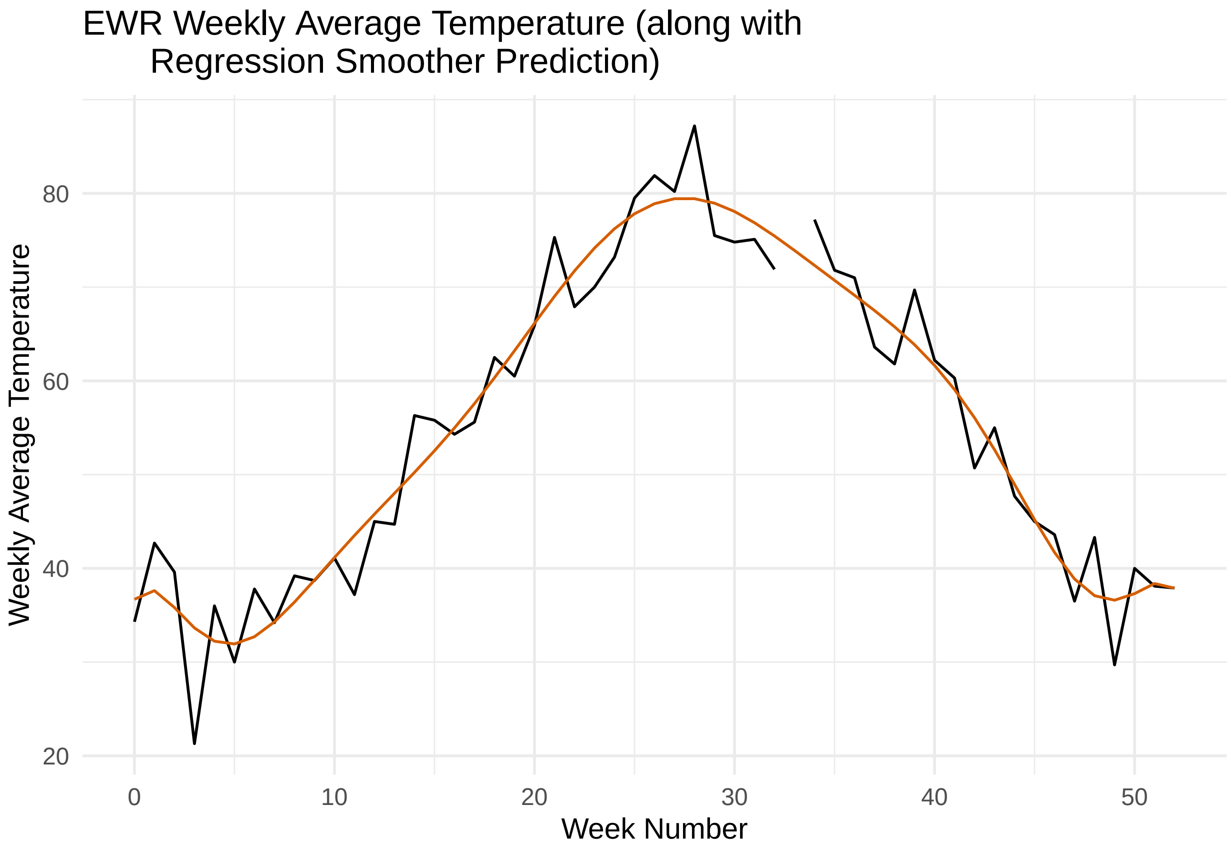
```r
ewr_data$temp_predictions <- predict(ewr_reg_smoother, newdata = ewr_data)
head(ewr_data)
```

```
## # A tibble: 6 x 5
##   origin week_index average_temperature average_dewpoint temp_predictions
##   <chr>       <dbl>               <dbl>            <dbl>            <dbl>
## 1 EWR             0                34.3             19.4             36.7
## 2 EWR             1                42.7             33.3             37.6
## 3 EWR             2                39.6             26.5             35.8
## 4 EWR             3                21.3              3.8             33.6
## 5 EWR             4                36               25.7             32.2
## 6 EWR             5                30               17.6             31.9
```

```r
ewr_data <- ewr_data %>% as_tsibble(index=week_index, key=origin)
ewr_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint temp_predictions
##    <chr>       <dbl>               <dbl>            <dbl>            <dbl>
##  1 EWR             0                34.3             19.4             36.7
##  2 EWR             1                42.7             33.3             37.6
##  3 EWR             2                39.6             26.5             35.8
##  4 EWR             3                21.3              3.8             33.6
##  5 EWR             4                36               25.7             32.2
##  6 EWR             5                30               17.6             31.9
##  7 EWR             6                37.8             25.3             32.7
##  8 EWR             7                34.2             21.3             34.3
##  9 EWR             8                39.2             27.8             36.4
## 10 EWR             9                38.7             26.8             38.7
## # i 43 more rows
```

```r
ewr_data |>
  autoplot(average_temperature) +
  geom_line(aes(y = temp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "EWR Weekly Average Temperature (along with
       Regression Smoother Prediction)")
```



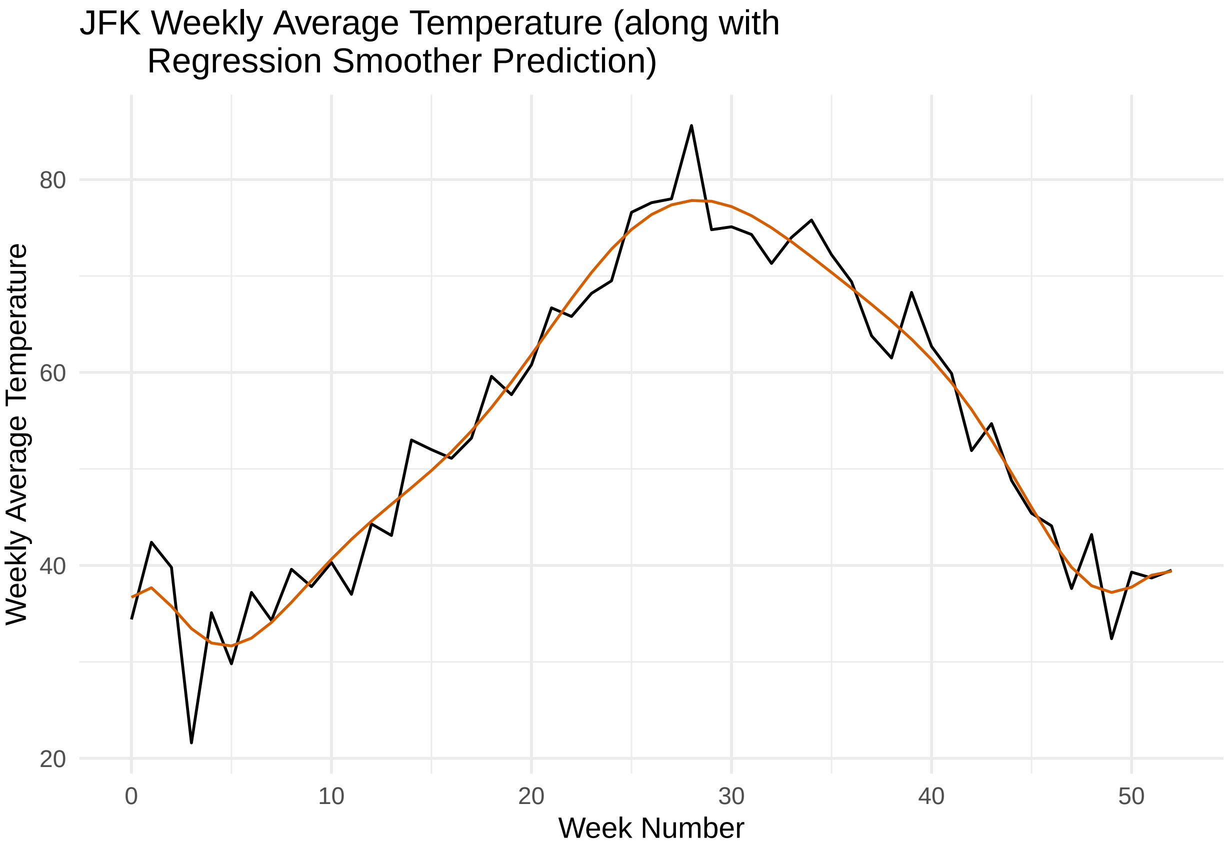
```
jfk_data <- weekly_weather %>%
  filter(origin == 'JFK')
jfk_reg_smoother <- lm(average_temperature ~ poly(week_index, 10),
                       data = jfk_data, na.action = na.omit)

jfk_data$temp_predictions <- predict(jfk_reg_smoother, newdata = jfk_data)
head(jfk_data)
```

```
## # A tibble: 6 x 5
##   origin week_index average_temperature average_dewpoint temp_predictions
##   <chr>       <dbl>               <dbl>            <dbl>            <dbl>
## 1 JFK             0                34.4             19.2             36.7
## 2 JFK             1                42.4             33.3             37.7
## 3 JFK             2                39.8             26.3             35.8
## 4 JFK             3                21.6              3.2             33.4
## 5 JFK             4                35.1             24.7             32.0
## 6 JFK             5                29.8             16.9             31.7
```

```
jfk_data <- jfk_data %>% as_tsibble(index=week_index, key=origin)
jfk_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint temp_predictions
##    <chr>       <dbl>               <dbl>            <dbl>            <dbl>
##  1 JFK             0                34.4             19.2             36.7
##  2 JFK             1                42.4             33.3             37.7
##  3 JFK             2                39.8             26.3             35.8
##  4 JFK             3                21.6              3.2             33.4
##  5 JFK             4                35.1             24.7             32.0
##  6 JFK             5                29.8             16.9             31.7
##  7 JFK             6                37.2             25.2             32.5
##  8 JFK             7                34.3             21               34.1
##  9 JFK             8                39.6             27.7             36.2
## 10 JFK             9                37.8             26.2             38.4
## # i 43 more rows
```

```
jfk_data |>
  autoplot(average_temperature) +
  geom_line(aes(y = temp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "JFK Weekly Average Temperature (along with
       Regression Smoother Prediction)")
```



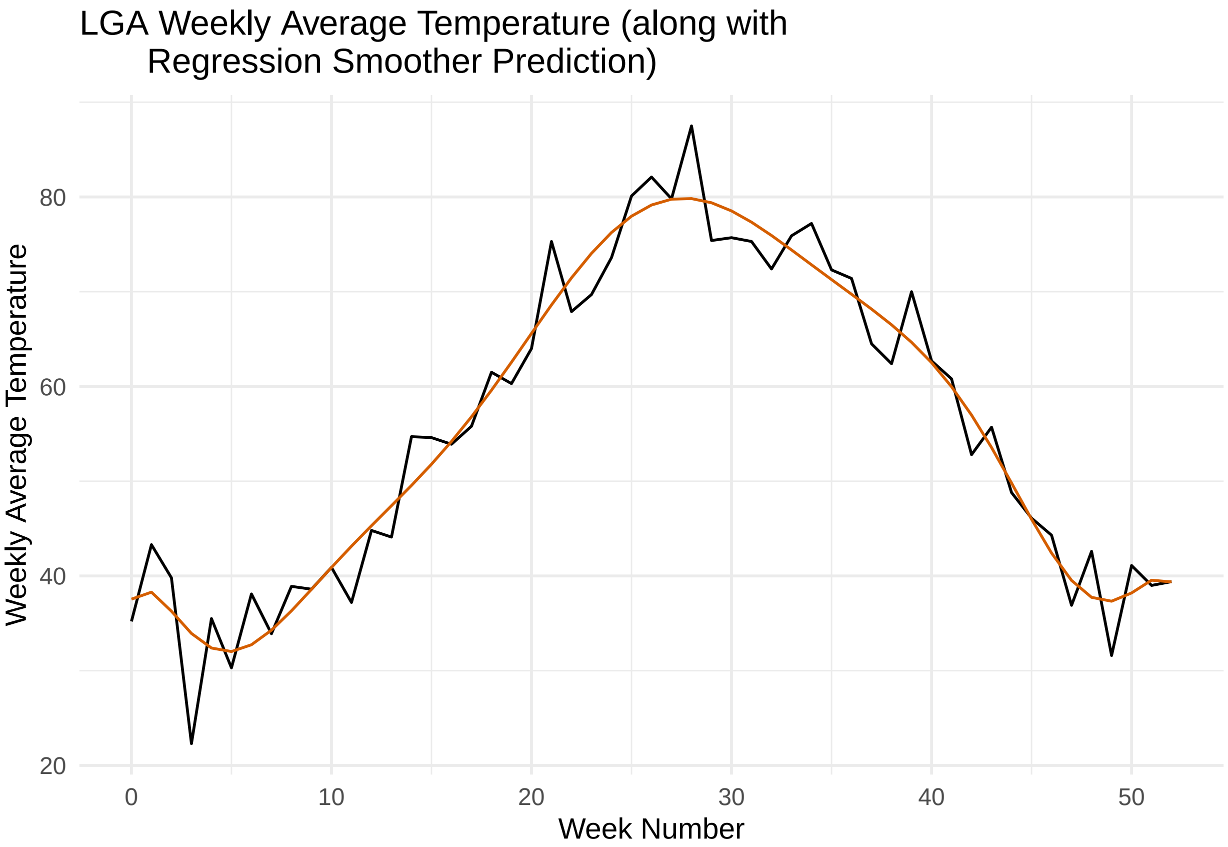
```
lga_data <- weekly_weather %>%
  filter(origin == 'LGA')
lga_reg_smoother <- lm(average_temperature ~ poly(week_index, 10),
                    data = lga_data, na.action = na.omit)

lga_data$temp_predictions <- predict(lga_reg_smoother, newdata = lga_data)
head(lga_data)
```

```
## # A tibble: 6 x 5
##   origin week_index average_temperature average_dewpoint temp_predictions
##   <chr>       <dbl>               <dbl>            <dbl>            <dbl>
## 1 LGA             0                35.2             18.8             37.5
## 2 LGA              1                 43.3              32.3              38.3
## 3 LGA              2                 39.8              25.3              36.3
## 4 LGA              3                 22.3               5                33.9
## 5 LGA              4                 35.5              23.9              32.4
## 6 LGA              5                 30.3              16.6              32.0
```

```
lga_data <- lga_data %>% as_tsibble(index=week_index, key=origin)
lga_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint temp_predictions
##    <chr>       <dbl>               <dbl>            <dbl>            <dbl>
##  1 LGA             0                35.2             18.8             37.5
##  2 LGA             1                43.3             32.3             38.3
##  3 LGA             2                39.8             25.3             36.3
##  4 LGA             3                22.3              5               33.9
##  5 LGA             4                35.5             23.9             32.4
##  6 LGA             5                30.3             16.6             32.0
##  7 LGA             6                38.1             25               32.7
##  8 LGA             7                33.9             20.2             34.3
##  9 LGA             8                38.9             27.4             36.3
## 10 LGA             9                38.6             25.9             38.6
## # i 43 more rows
```

```
lga_data |>
  autoplot(average_temperature) +
  geom_line(aes(y = temp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "LGA Weekly Average Temperature (along with
       Regression Smoother Prediction)")
```



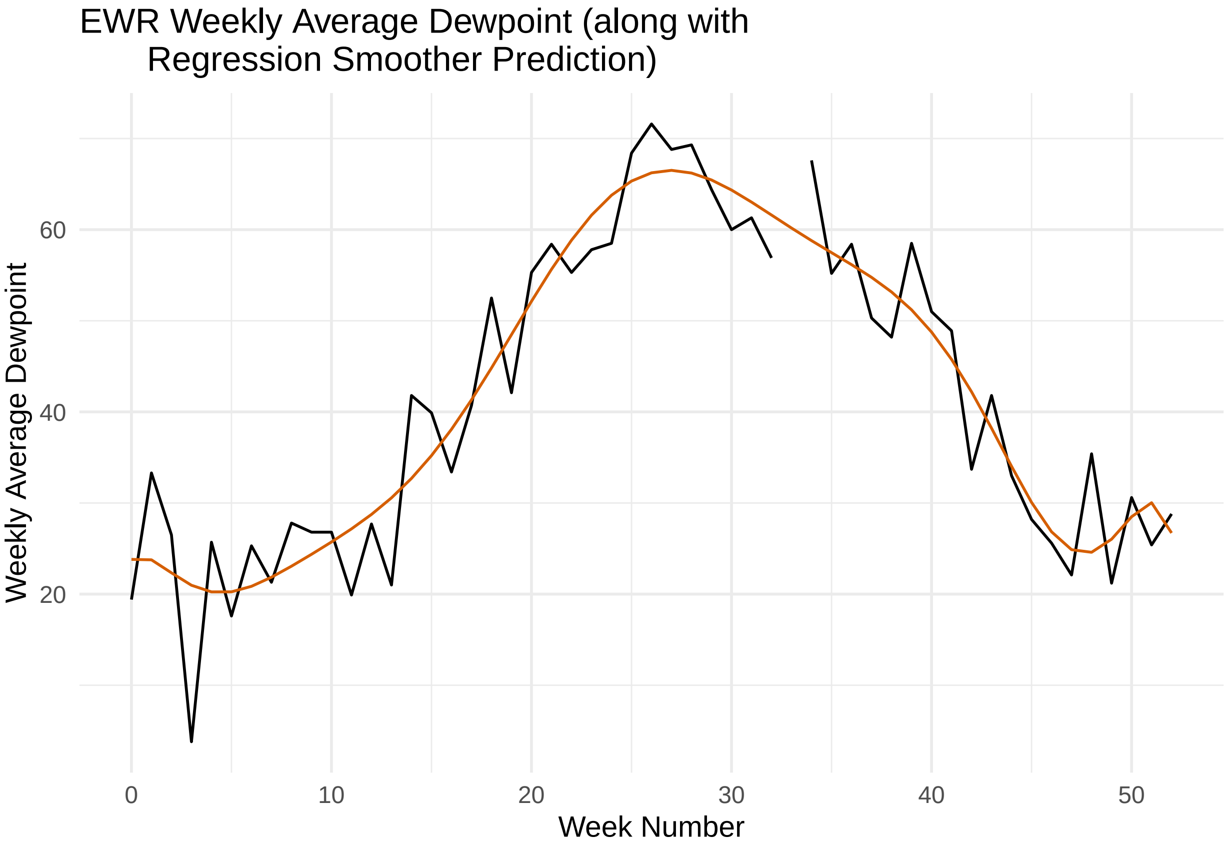
```
ewr_dewp_data <- weekly_weather %>%
  filter(origin == 'EWR')
ewr_reg_smoother <- lm(average_dewpoint ~ poly(week_index, 10),
                    data = ewr_dewp_data, na.action = na.omit)

ewr_dewp_data$dewp_predictions <- predict(ewr_reg_smoother,
                                          newdata = ewr_dewp_data)
head(ewr_dewp_data)
```

```
## # A tibble: 6 x 5
##   origin week_index average_temperature average_dewpoint dewp_predictions
##   <chr>       <dbl>               <dbl>            <dbl>            <dbl>
## 1 EWR             0                34.3             19.4             23.8
## 2 EWR             1                42.7             33.3             23.8
## 3 EWR             2                39.6             26.5             22.3
## 4 EWR             3                21.3              3.8             21.0
## 5 EWR             4                36               25.7             20.2
## 6 EWR             5                30               17.6             20.3
```

```
ewr_dewp_data <- ewr_dewp_data %>% as_tsibble(index=week_index, key=origin)
ewr_dewp_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint dewp_predictions
##    <chr>       <dbl>               <dbl>            <dbl>            <dbl>
##  1 EWR             0                34.3             19.4             23.8
##  2 EWR             1                42.7             33.3             23.8
##   3 EWR                2                 39.6              26.5              22.3
##   4 EWR                3                 21.3               3.8              21.0
##   5 EWR                4                 36                25.7              20.2
##   6 EWR                5                 30                17.6              20.3
##   7 EWR                6                 37.8              25.3              20.9
##   8 EWR                7                 34.2              21.3              21.9
##   9 EWR                8                 39.2              27.8              23.1
##  10 EWR                9                 38.7              26.8              24.4
## # i 43 more rows
```

```
ewr_dewp_data |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = dewp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "EWR Weekly Average Dewpoint (along with
       Regression Smoother Prediction)")
```



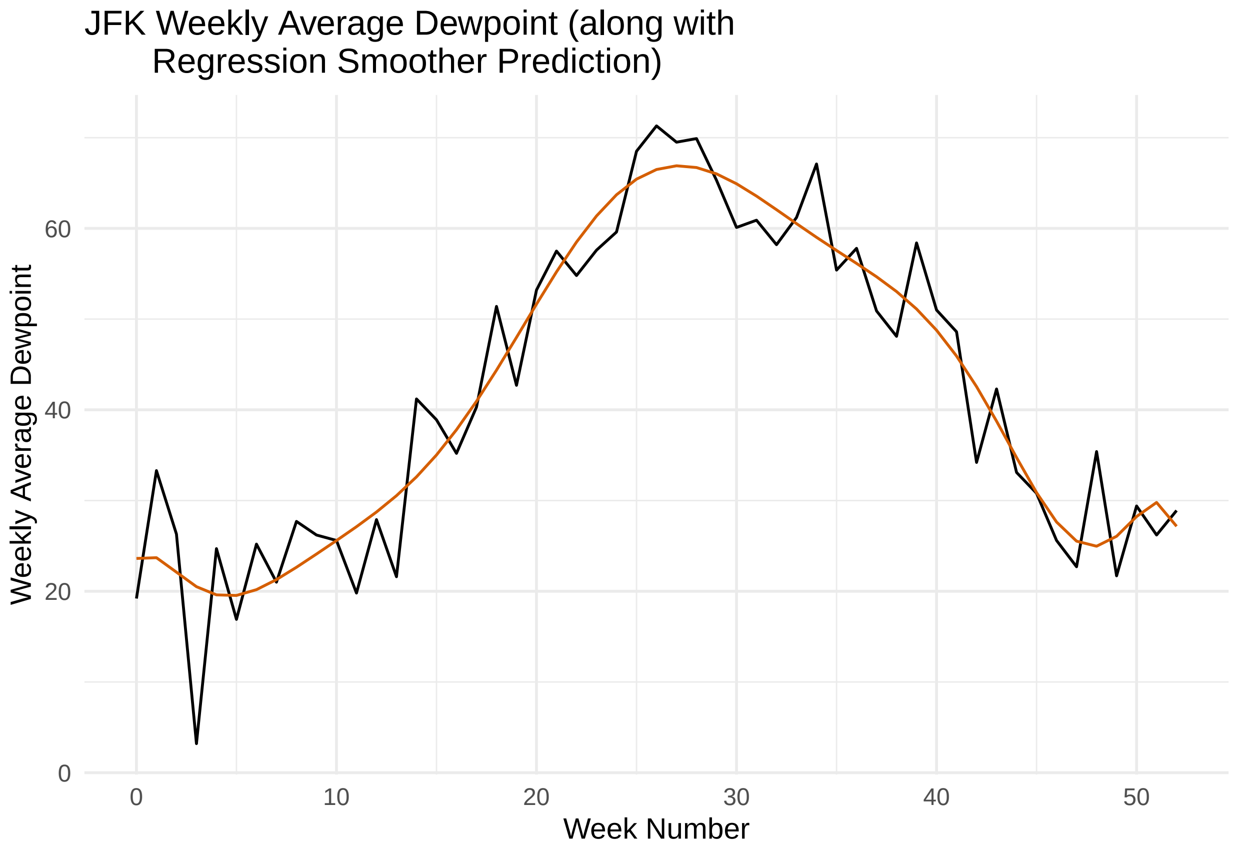
```
jfk_dewp_data <- weekly_weather %>%
  filter(origin == 'JFK')
jfk_reg_smoother <- lm(average_dewpoint ~ poly(week_index, 10),
                       data = jfk_dewp_data, na.action = na.omit)

jfk_dewp_data$dewp_predictions <- predict(jfk_reg_smoother,
                                          newdata = jfk_dewp_data)
head(jfk_dewp_data)
```

```
## # A tibble: 6 x 5
##   origin week_index average_temperature average_dewpoint dewp_predictions
##   <chr>       <dbl>               <dbl>            <dbl>            <dbl>
## 1 JFK             0                34.4             19.2             23.6
## 2 JFK             1                42.4             33.3             23.7
## 3 JFK             2                39.8             26.3             22.1
## 4 JFK             3                21.6              3.2             20.5
## 5 JFK             4                35.1             24.7             19.6
## 6 JFK             5                29.8             16.9             19.5
```

```
jfk_dewp_data <- jfk_dewp_data %>% as_tsibble(index=week_index, key=origin)
jfk_dewp_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint dewp_predictions
##    <chr>       <dbl>               <dbl>            <dbl>            <dbl>
##  1 JFK             0                34.4             19.2             23.6
##  2 JFK             1                42.4             33.3             23.7
##  3 JFK             2                39.8             26.3             22.1
##  4 JFK             3                21.6              3.2             20.5
##  5 JFK             4                35.1             24.7             19.6
##  6 JFK             5                29.8             16.9             19.5
##  7 JFK             6                37.2             25.2             20.2
##  8 JFK             7                34.3             21               21.3
##  9 JFK             8                39.6             27.7             22.6
## 10 JFK             9                37.8             26.2             24.1
## # i 43 more rows
```

```
jfk_dewp_data |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = dewp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "JFK Weekly Average Dewpoint (along with
       Regression Smoother Prediction)")
```



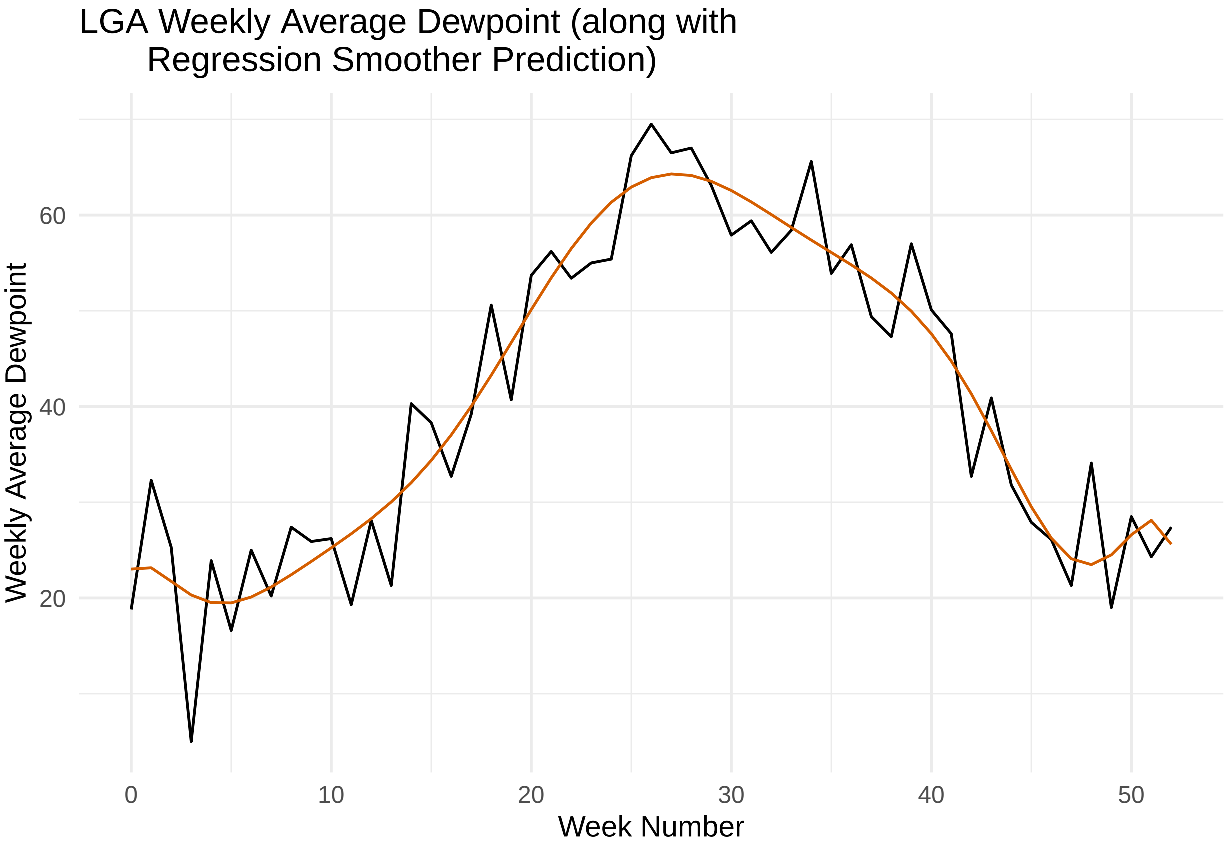
```
lga_dewp_data <- weekly_weather %>%
  filter(origin == 'LGA')
lga_reg_smoother <- lm(average_dewpoint ~ poly(week_index, 10),
                       data = lga_dewp_data, na.action = na.omit)

lga_dewp_data$dewp_predictions <- predict(lga_reg_smoother,
                                           newdata = lga_dewp_data)
head(lga_dewp_data)
```

```
## # A tibble: 6 x 5
##   origin week_index average_temperature average_dewpoint dewp_predictions
##   <chr>       <dbl>               <dbl>            <dbl>            <dbl>
## 1 LGA             0                35.2             18.8             23.0
## 2 LGA             1                43.3             32.3             23.2
## 3 LGA             2                39.8             25.3             21.7
## 4 LGA             3                22.3              5               20.3
## 5 LGA             4                35.5             23.9             19.5
## 6 LGA             5                30.3             16.6             19.5
```

```
lga_dewp_data <- lga_dewp_data %>% as_tsibble(index=week_index, key=origin)
lga_dewp_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint dewp_predictions
##    <chr>       <dbl>               <dbl>            <dbl>            <dbl>
##  1 LGA             0                35.2             18.8             23.0
##  2 LGA             1                43.3             32.3             23.2
##   3 LGA              2               39.8            25.3            21.7
##   4 LGA              3               22.3             5              20.3
##   5 LGA              4               35.5            23.9            19.5
##   6 LGA              5               30.3            16.6            19.5
##   7 LGA              6               38.1            25              20.1
##   8 LGA              7               33.9            20.2            21.1
##   9 LGA              8               38.9            27.4            22.4
##  10 LGA              9               38.6            25.9            23.8
## # i 43 more rows
```

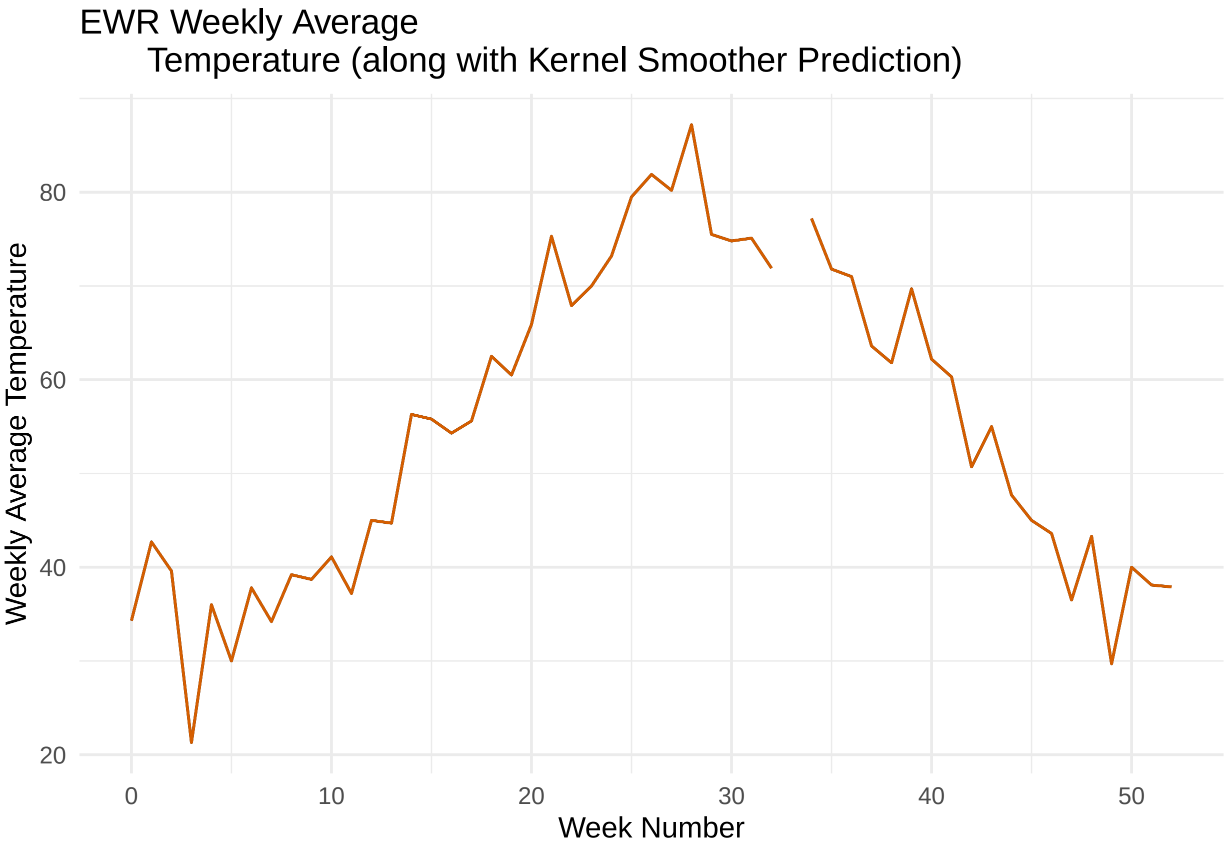
```
lga_dewp_data |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = dewp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "LGA Weekly Average Dewpoint (along with
       Regression Smoother Prediction)")
```



```
ewr_data <- weekly_weather %>%
  filter(origin == 'EWR')
smoothed_ewr_temp <- ksmooth(x=ewr_data$week_index,
                             y=ewr_data$average_temperature,
                             kernel = "normal", bandwidth=0.5, n.points = 53)
ewr_data$kernel_temp_predictions <- smoothed_ewr_temp$y

ewr_data <- ewr_data %>% as_tsibble(index=week_index, key=origin)
ewr_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint kernel_temp_predicti~1
##    <chr>       <dbl>               <dbl>            <dbl>                  <dbl>
##  1 EWR             0                34.3             19.4                   34.3
##  2 EWR             1                42.7             33.3                   42.7
##  3 EWR             2                39.6             26.5                   39.6
##  4 EWR             3                21.3              3.8                   21.3
##  5 EWR             4                36               25.7                   36
##  6 EWR             5                30               17.6                   30
##  7 EWR             6                37.8             25.3                   37.8
##  8 EWR             7                34.2             21.3                   34.2
##  9 EWR             8                39.2             27.8                   39.2
## 10 EWR             9                38.7             26.8                   38.7
## # i 43 more rows
## # i abbreviated name: 1: kernel_temp_predictions
```

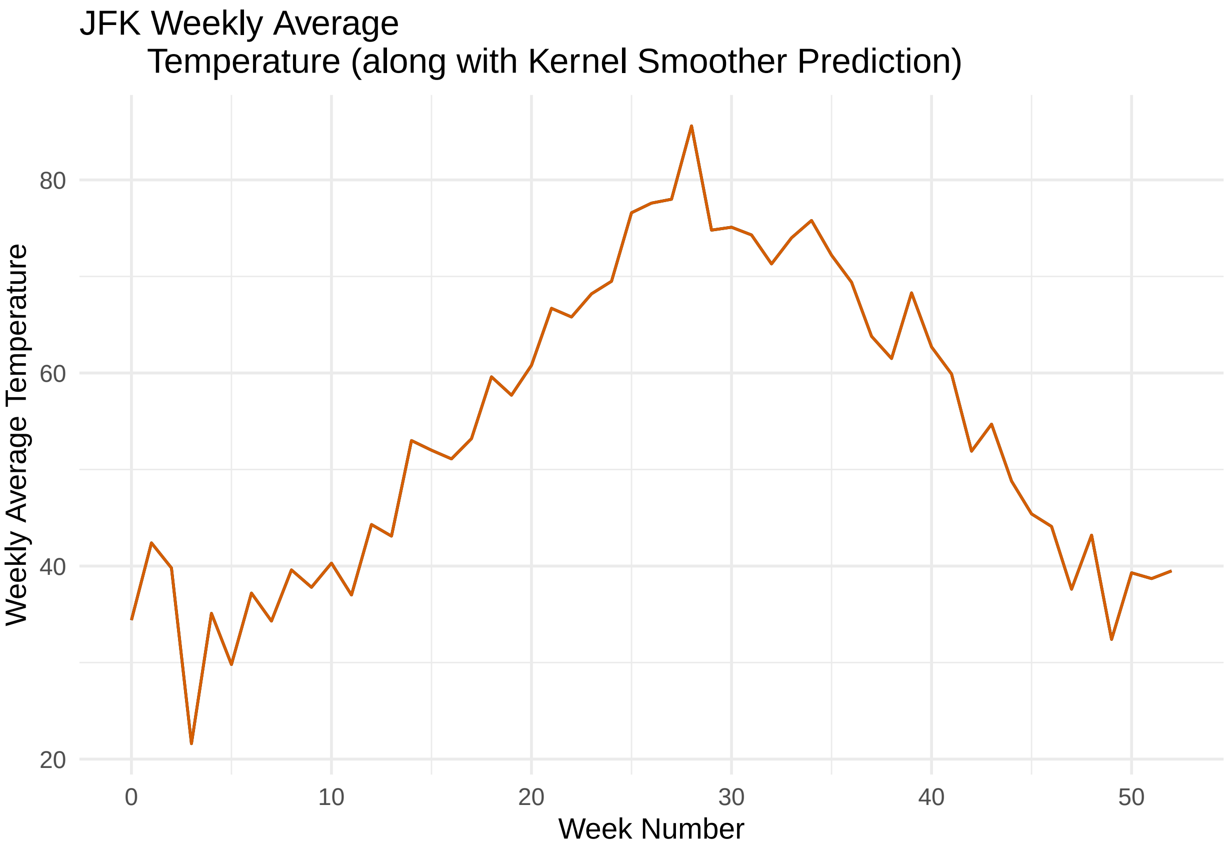
```
ewr_data |>
  autoplot(average_temperature) +
  geom_line(aes(y = kernel_temp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "EWR Weekly Average
       Temperature (along with Kernel Smoother Prediction)")
```



```
jfk_data <- weekly_weather %>%
  filter(origin == 'JFK')
smoothed_jfk_temp <- ksmooth(x=jfk_data$week_index,
                             y=jfk_data$average_temperature,
                             kernel = "normal", bandwidth=0.5,
                             n.points = 53)
jfk_data$kernel_temp_predictions <- smoothed_jfk_temp$y

jfk_data <- jfk_data %>% as_tsibble(index=week_index, key=origin)
jfk_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint kernel_temp_predicti~1
##    <chr>       <dbl>               <dbl>            <dbl>                  <dbl>
##  1 JFK             0                34.4             19.2                   34.4
##  2 JFK             1                42.4             33.3                   42.4
##  3 JFK             2                39.8             26.3                   39.8
##  4 JFK             3                21.6              3.2                   21.6
##  5 JFK             4                35.1             24.7                   35.1
##  6 JFK             5                29.8             16.9                   29.8
##  7 JFK             6                37.2             25.2                   37.2
##  8 JFK             7                34.3             21                     34.3
##  9 JFK             8                39.6             27.7                   39.6
## 10 JFK             9                37.8             26.2                   37.8
## # i 43 more rows
## # i abbreviated name: 1: kernel_temp_predictions
```

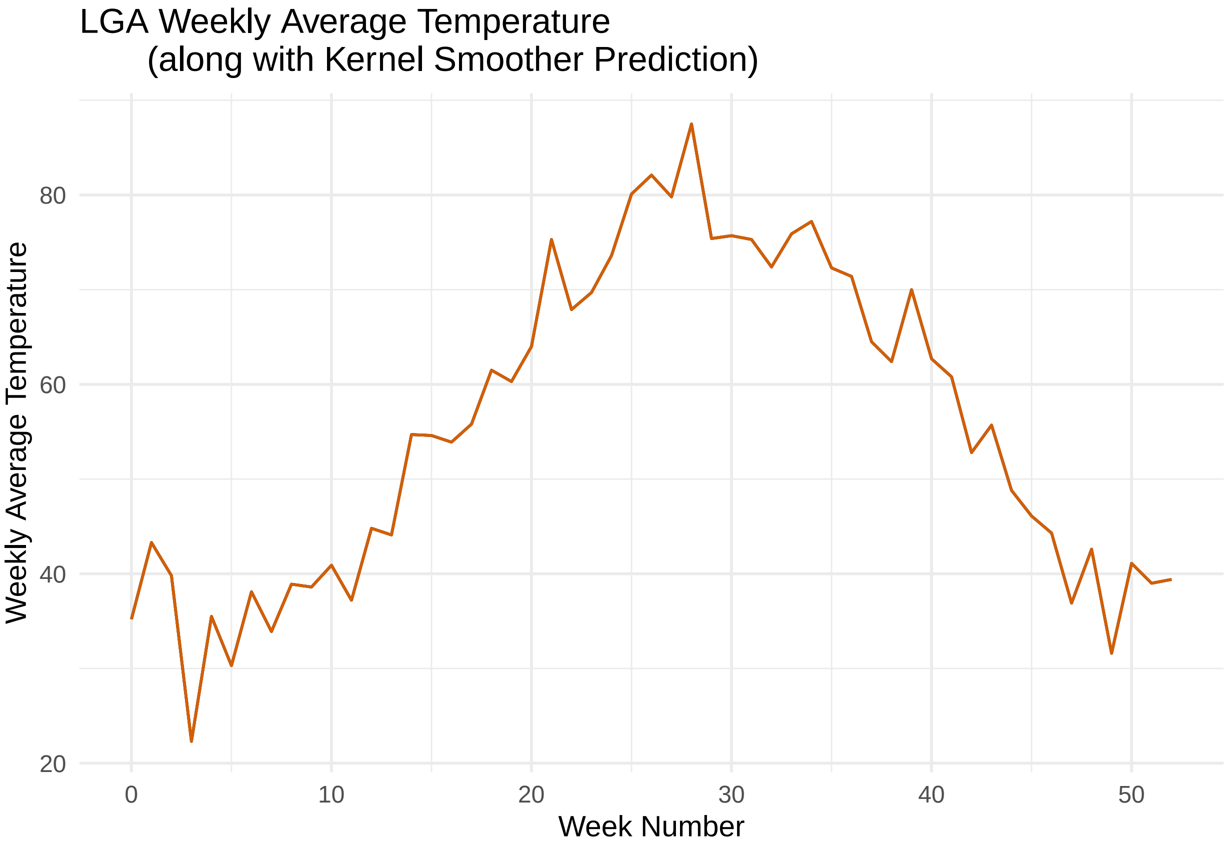
```
jfk_data |>
  autoplot(average_temperature) +
  geom_line(aes(y = kernel_temp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "JFK Weekly Average
       Temperature (along with Kernel Smoother Prediction)")
```



```
lga_data <- weekly_weather %>%
  filter(origin == 'LGA')
smoothed_lga_temp <- ksmooth(x=lga_data$week_index,
                             y=lga_data$average_temperature,
                             kernel = "normal", bandwidth=0.5, n.points = 53)
lga_data$kernel_temp_predictions <- smoothed_lga_temp$y

lga_data <- lga_data %>%
  as_tsibble(index=week_index, key=origin)
lga_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint kernel_temp_predicti~1
##    <chr>       <dbl>               <dbl>            <dbl>                  <dbl>
##  1 LGA             0                35.2             18.8                   35.2
##  2 LGA             1                43.3             32.3                   43.3
##  3 LGA             2                39.8             25.3                   39.8
##  4 LGA             3                22.3              5                     22.3
##  5 LGA             4                35.5             23.9                   35.5
##  6 LGA             5                30.3             16.6                   30.3
##  7 LGA             6                38.1             25                     38.1
##  8 LGA             7                33.9             20.2                   33.9
##  9 LGA             8                38.9             27.4                   38.9
## 10 LGA             9                38.6             25.9                   38.6
## # i 43 more rows
## # i abbreviated name: 1: kernel_temp_predictions
```

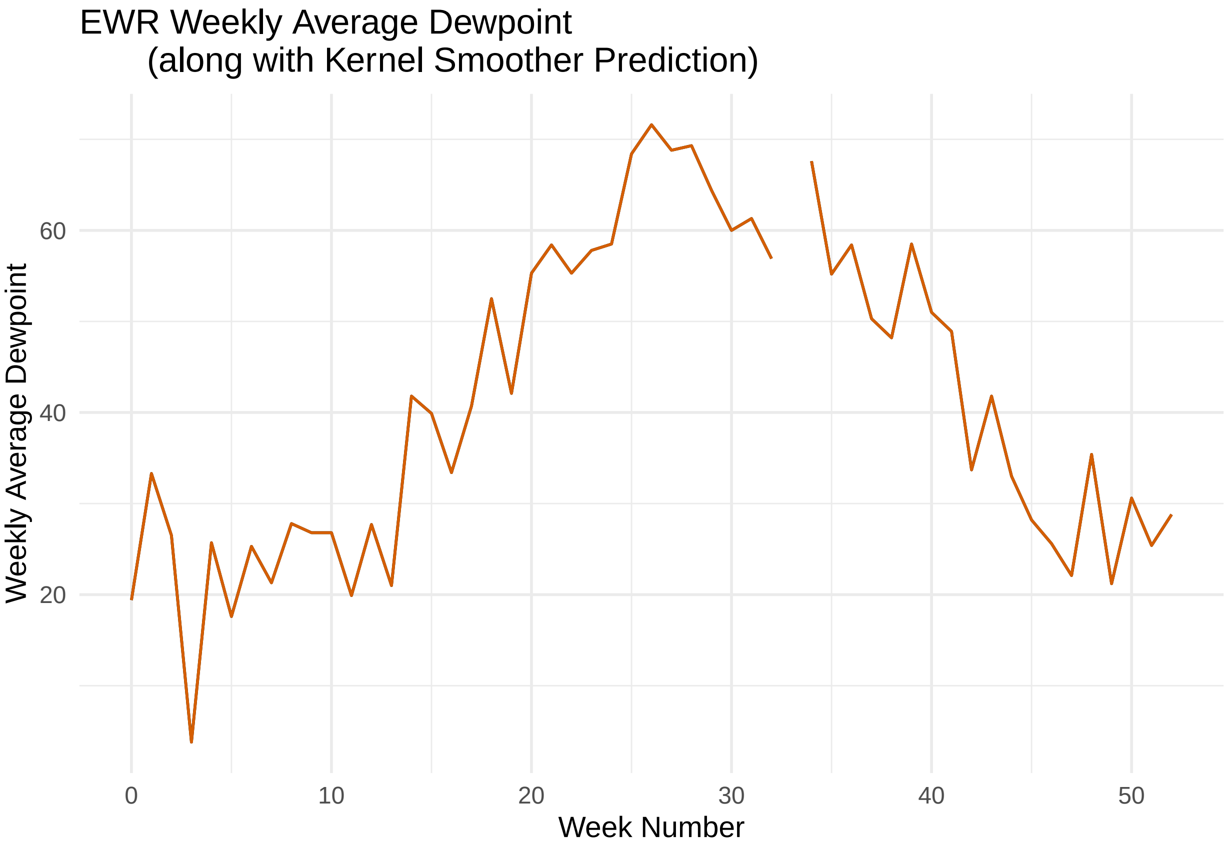
```r
lga_data |>
  autoplot(average_temperature) +
  geom_line(aes(y = kernel_temp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Temperature",
       title = "LGA Weekly Average Temperature
       (along with Kernel Smoother Prediction)")
```



```r
ewr_data <- weekly_weather %>%
  filter(origin == 'EWR')
smoothed_ewr_dewp <- ksmooth(x=ewr_data$week_index,
                             y=ewr_data$average_dewpoint, kernel = "normal",
                             bandwidth=0.5, n.points = 53)
ewr_data$kernel_dewp_predictions <- smoothed_ewr_dewp$y

ewr_data <- ewr_data %>% as_tsibble(index=week_index, key=origin)
ewr_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint kernel_dewp_predicti~1
##    <chr>       <dbl>               <dbl>            <dbl>                  <dbl>
##  1 EWR             0                34.3             19.4                   19.4
##  2 EWR             1                42.7             33.3                   33.3
##  3 EWR             2                39.6             26.5                   26.5
##  4 EWR             3                21.3              3.8                    3.8
##  5 EWR             4                36               25.7                   25.7
##  6 EWR             5                30               17.6                   17.6
##  7 EWR               6              37.8          25.3                 25.3
##  8 EWR               7              34.2          21.3                 21.3
##  9 EWR               8              39.2          27.8                 27.8
## 10 EWR               9              38.7          26.8                 26.8
## # i 43 more rows
## # i abbreviated name: 1: kernel_dewp_predictions
```

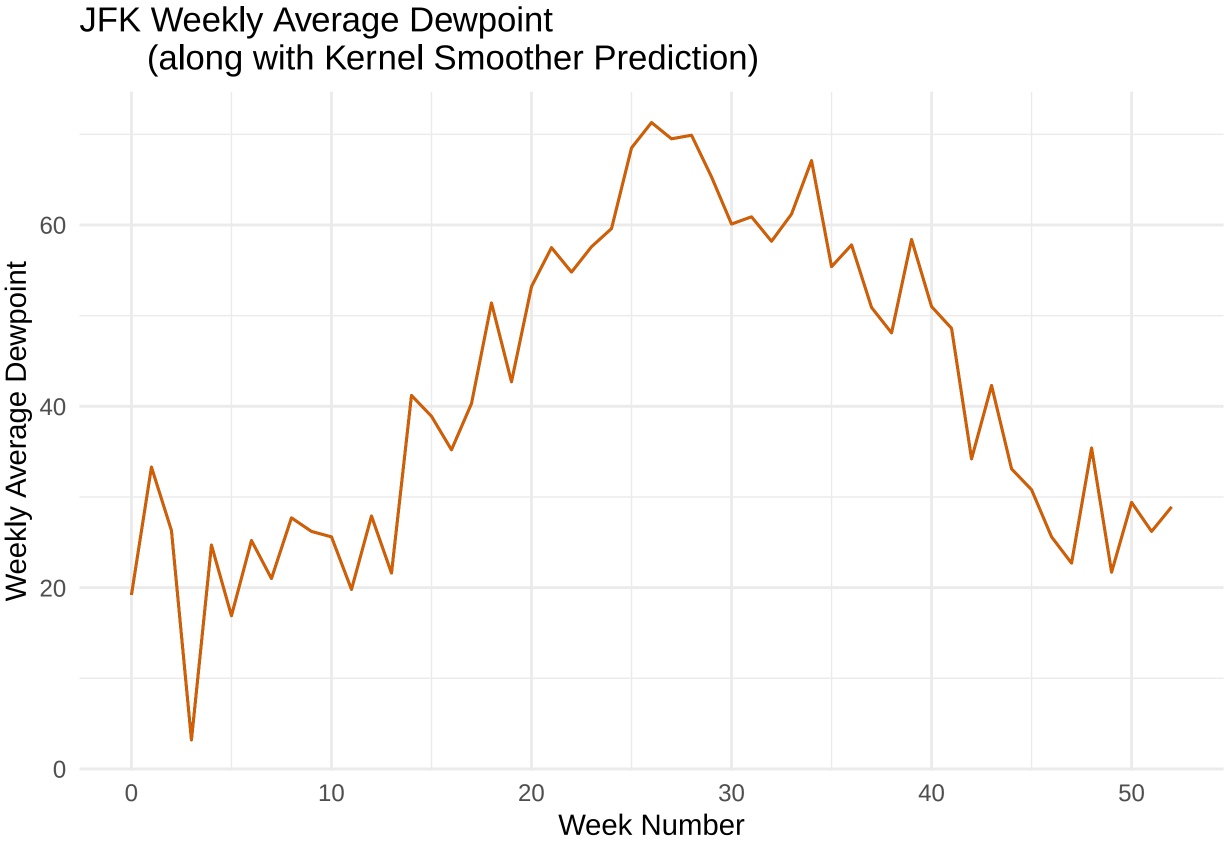
```
ewr_data |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = kernel_dewp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "EWR Weekly Average Dewpoint
       (along with Kernel Smoother Prediction)")
```



```
jfk_data <- weekly_weather %>%
  filter(origin == 'JFK')
smoothed_jfk_dewp <- ksmooth(x=jfk_data$week_index,
                             y=jfk_data$average_dewpoint, kernel = "normal",
                             bandwidth=0.5, n.points = 53)
jfk_data$kernel_dewp_predictions <- smoothed_jfk_dewp$y

jfk_data <- jfk_data %>% as_tsibble(index=week_index, key=origin)
jfk_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint kernel_dewp_predicti~1
##    <chr>       <dbl>               <dbl>            <dbl>                  <dbl>
##  1 JFK             0                34.4             19.2                   19.2
##  2 JFK             1                42.4             33.3                   33.3
##  3 JFK             2                39.8             26.3                   26.3
##  4 JFK             3                21.6              3.2                    3.2
##  5 JFK             4                35.1             24.7                   24.7
##  6 JFK             5                29.8             16.9                   16.9
##  7 JFK             6                37.2             25.2                   25.2
##  8 JFK             7                34.3             21                     21
##  9 JFK             8                39.6             27.7                   27.7
## 10 JFK             9                37.8             26.2                   26.2
## # i 43 more rows
## # i abbreviated name: 1: kernel_dewp_predictions
```

```
jfk_data |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = kernel_dewp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "JFK Weekly Average Dewpoint
       (along with Kernel Smoother Prediction)")
```

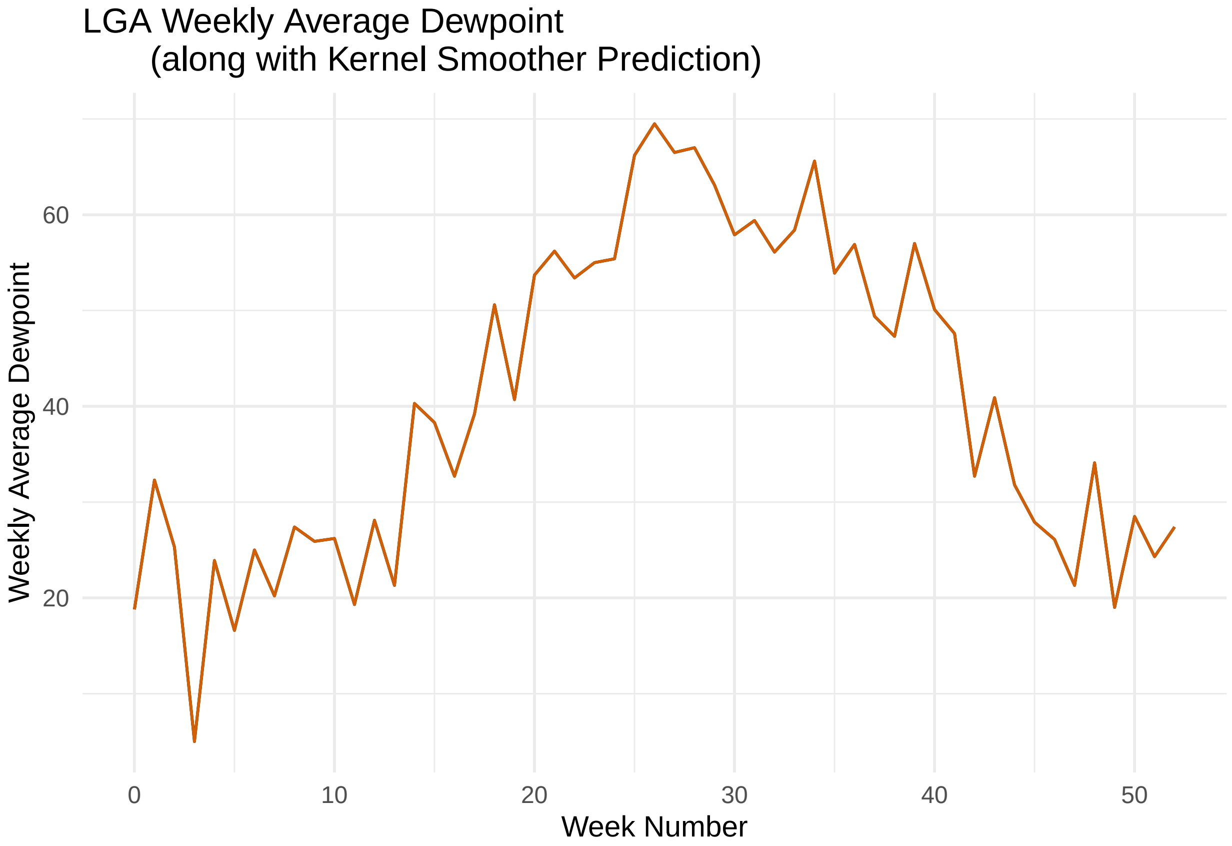


```
lga_data <- weekly_weather %>%
  filter(origin == 'LGA')
smoothed_lga_dewp <- ksmooth(x=lga_data$week_index,
                             y=lga_data$average_dewpoint,
                             kernel = "normal", bandwidth=0.5, n.points = 53)
lga_data$kernel_dewp_predictions <- smoothed_lga_dewp$y

lga_data <- lga_data %>% as_tsibble(index=week_index, key=origin)
```

```
lga_data
```

```
## # A tsibble: 53 x 5 [1]
## # Key:       origin [1]
##    origin week_index average_temperature average_dewpoint kernel_dewp_predicti~1
##    <chr>       <dbl>               <dbl>            <dbl>                  <dbl>
##  1 LGA             0                35.2             18.8                   18.8
##  2 LGA             1                43.3             32.3                   32.3
##  3 LGA             2                39.8             25.3                   25.3
##  4 LGA             3                22.3              5                      5
##  5 LGA             4                35.5             23.9                   23.9
##  6 LGA             5                30.3             16.6                   16.6
##  7 LGA             6                38.1             25                     25
##  8 LGA             7                33.9             20.2                   20.2
##  9 LGA             8                38.9             27.4                   27.4
## 10 LGA             9                38.6             25.9                   25.9
## # i 43 more rows
## # i abbreviated name: 1: kernel_dewp_predictions
```

```
lga_data |>
  autoplot(average_dewpoint) +
  geom_line(aes(y = kernel_dewp_predictions), colour = "#D55E00") +
  labs(x = "Week Number", y = "Weekly Average Dewpoint",
       title = "LGA Weekly Average Dewpoint
       (along with Kernel Smoother Prediction)")
```


LGA Weekly Average Dewpoint
(along with Kernel Smoother Prediction)
Weekly Average Dewpoint
Week Number