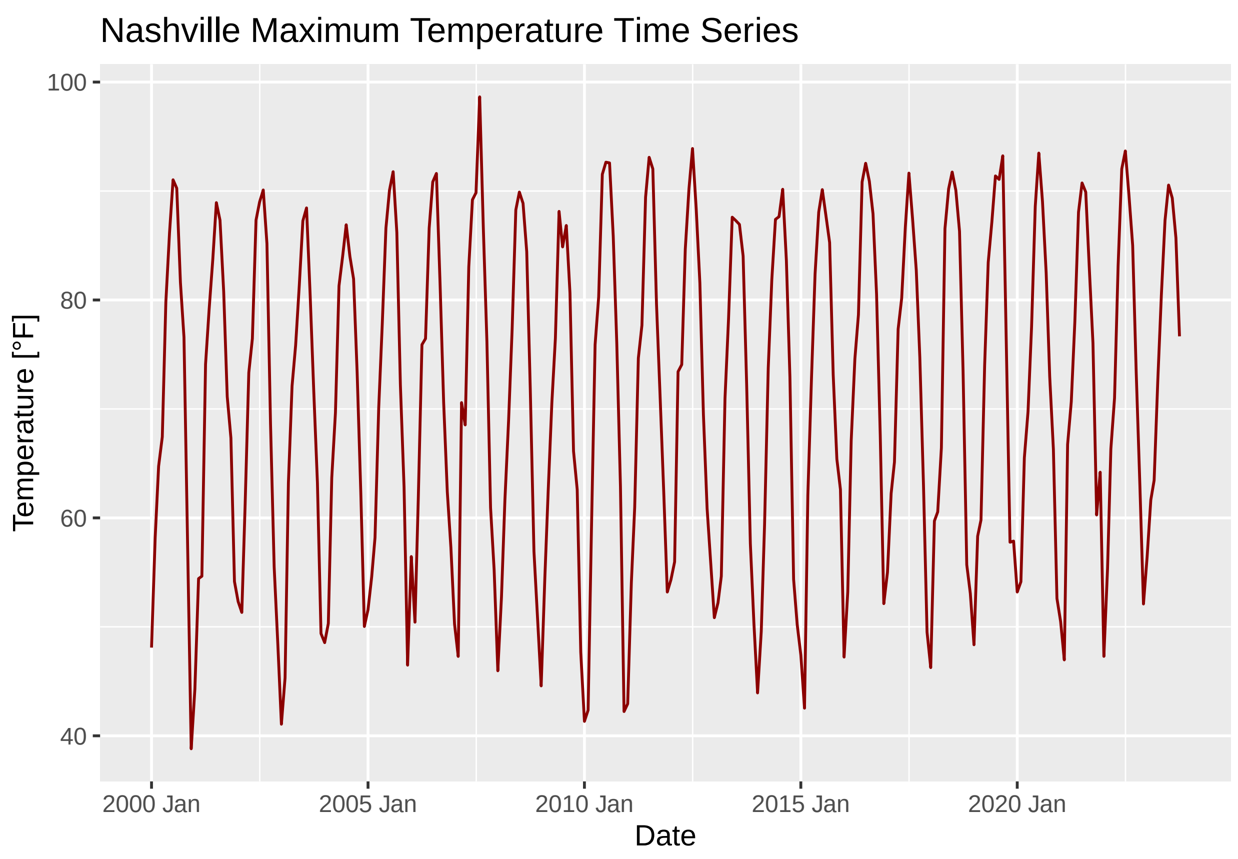
Forecasting Project

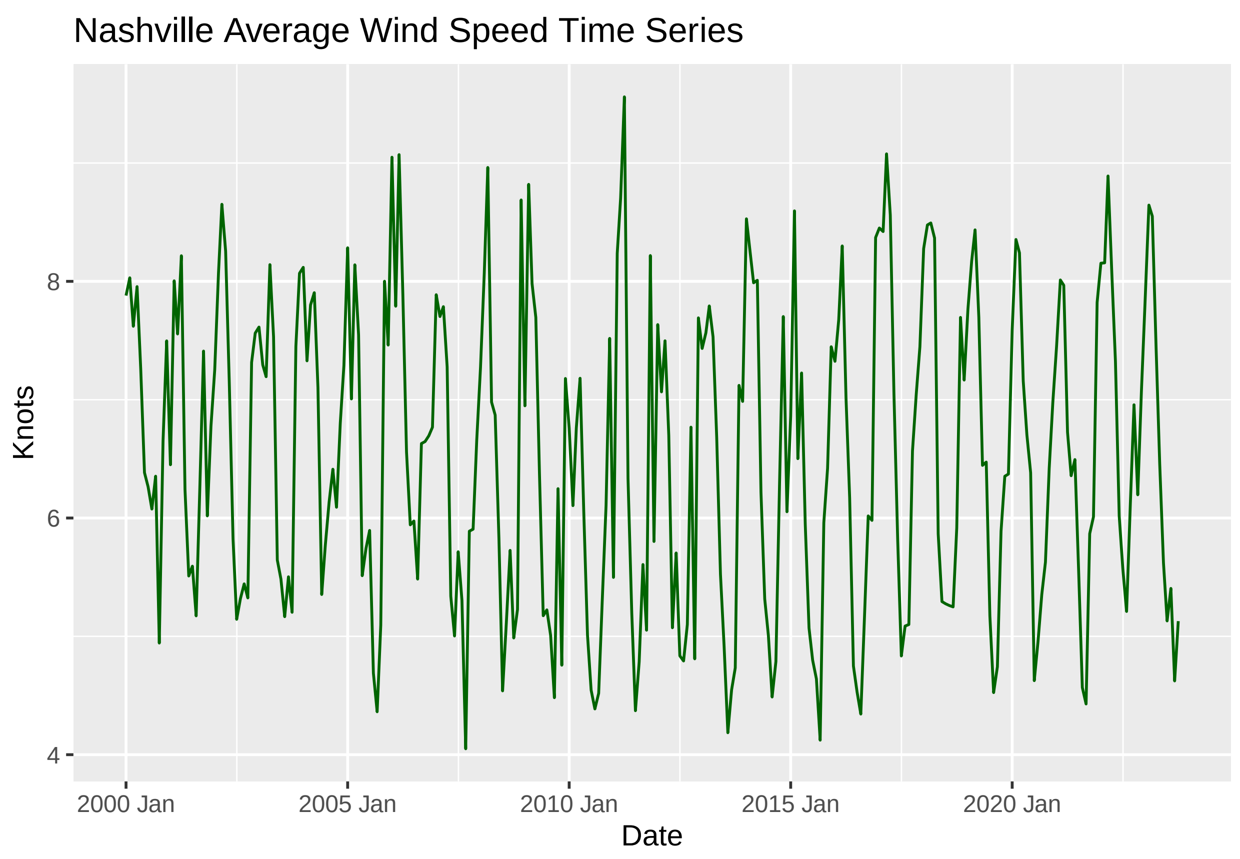
DANIEL LOPEZ

Forecasting Considerations

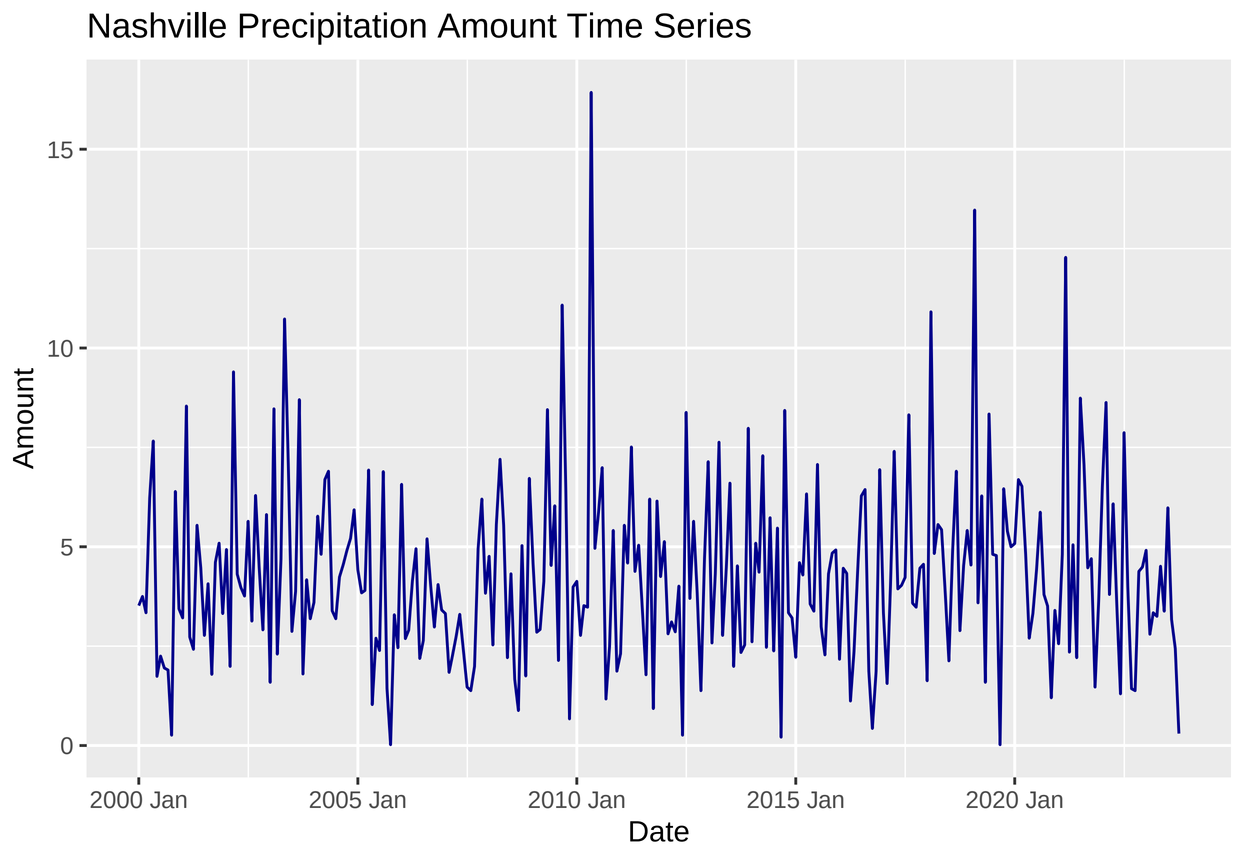
```r
data = read.csv("data.csv", stringsAsFactors = FALSE, header = TRUE)
# clean format
data$DATE <- ymd(data$DATE)
# remove NAs

# convert to tsibble
weather <- data |> as_tsibble(index = DATE)
weather <- subset(weather, select = c('DATE', 'TMAX', 'AWND','PRCP','SNOW'))
# aggregrate by month for more contextual forecasts.
monthly_weather <- data %>%
  mutate(DATE = yearmonth(DATE)) %>%  # Convert DATE to yearmonth class
  group_by(DATE) %>%
  summarise(
    TMAX = mean(TMAX, na.rm = TRUE),
    AWND = mean(AWND, na.rm = TRUE),
    PRCP = sum(PRCP, na.rm = TRUE),   # Change to sum if you prefer total monthly precipitation
    SNOW = sum(SNOW, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  as_tsibble(index = DATE)

weather_train <- monthly_weather %>%
      filter(DATE <= yearmonth("2020-01"))

weather_test <- monthly_weather %>%
      filter(DATE >= yearmonth("2020-02"))


ggplot(monthly_weather , aes(x = DATE, y = TMAX)) +
  geom_line(color = "darkred") +
  xlab("Date") +
  ylab("Temperature [°F]") +
  ggtitle("Nashville Maximum Temperature Time Series")
```



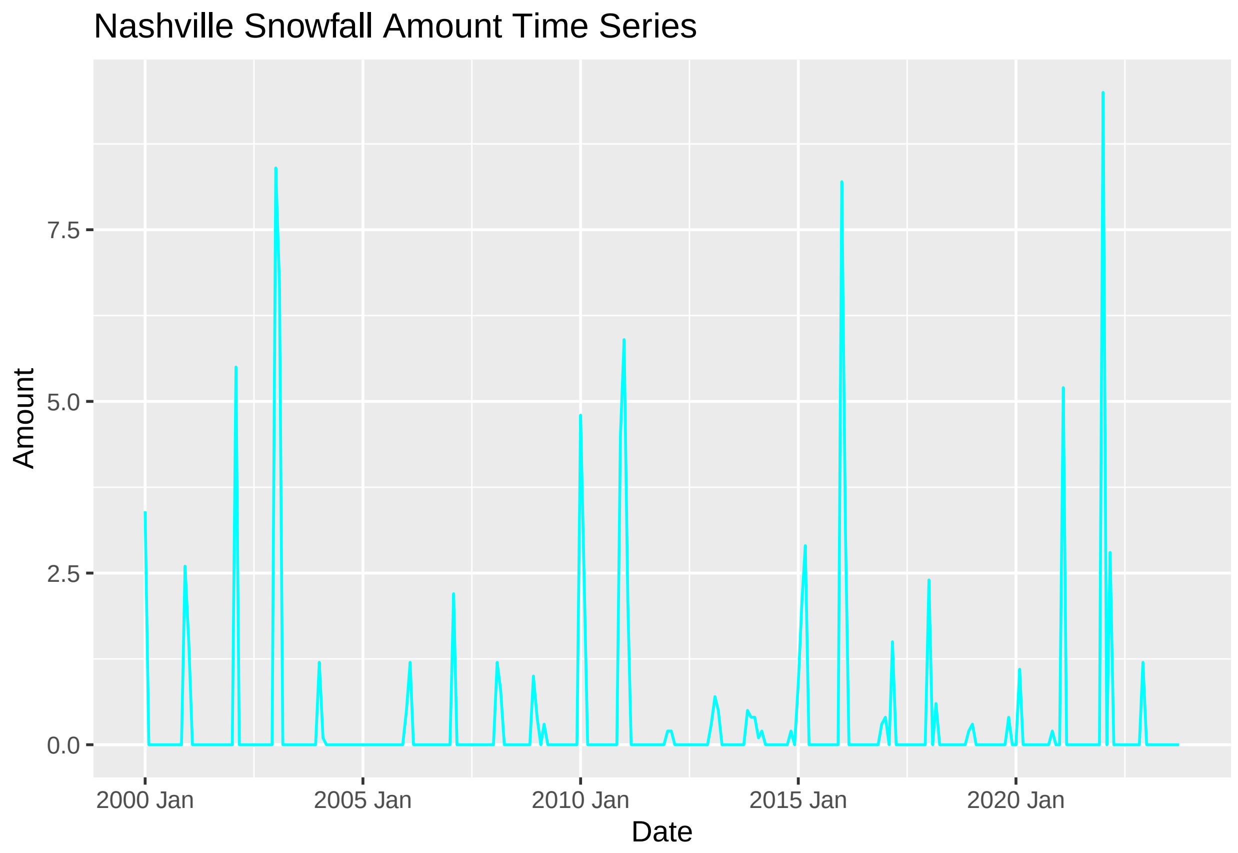
```
ggplot(monthly_weather , aes(x = DATE, y = AWND)) +
  geom_line(color = "darkgreen") +
  xlab("Date") +
  ylab("Knots") +
  ggtitle("Nashville Average Wind Speed Time Series")
```



```r
ggplot(monthly_weather , aes(x = DATE, y = PRCP)) +
  geom_line(color = "darkblue") +
  xlab("Date") +
  ylab("Amount") +
  ggtitle("Nashville Precipitation Amount Time Series")
```



```r
ggplot(monthly_weather , aes(x = DATE, y = SNOW)) +
  geom_line(color = "cyan") +
  xlab("Date") +
  ylab("Amount") +
  ggtitle("Nashville Snowfall Amount Time Series")
```



This project aims to forecast the monthly maximum temperatures in Nashville Tennessee. Knowing the maximum temperaturecan help farmers predict crop yields, help health officials predict heat waves and disease risk, help city planners predict energy demand andtourism numbers, help researchers understand climate trends, and more. For forecasting the maximum temperature in Nashville, we will utilize the following variables: average wind speed, precipitation amount, snowfall amount, and date. For the purpose of this forecast, we omit average temperature and minimum temperature. These variables are correlated and might provide “unfair” information into the model, as knowing the minimum temperature may provide a too strong assumption about the max temperature. Additionally, I want to make the forecast more difficult. For simplification, we agreggate the data into monthly intervals. It will help make forecasting trends much easier and more effective given our known variables.

Wind speed is a strong component in understanding the maximum temperature of a particular day. While it doesn’t directly affect temperature, wind helps distribute heat across different areas through the process of advection. Depending on which air mass is being advected, we could see higher or lower temperatures in Nashville, though this knowledge would be difficult to model in a time series here. Generally, though, wind can help evenly distribute the temperature of an area and lead to less hot or cold spots. In the plot above, there appears to be seasonality to the wind speed, which may be attributed to periods of high-action weather and the movement of air masses.

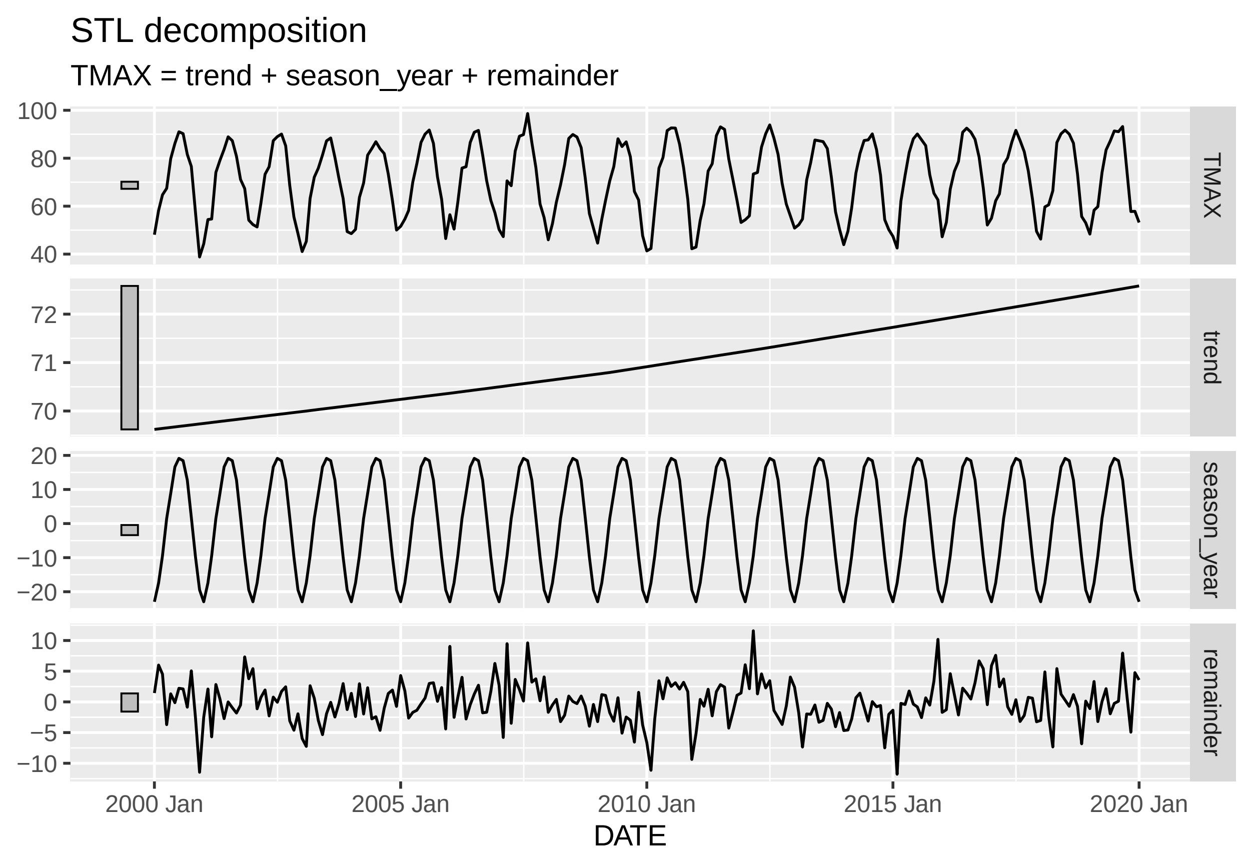
Precipitation is a strong component to the daily maximum temperature. If there is high precpitation, the atmosphere cools due to the latent heat of evaporation when water falls from the clouds. Additionally, precipitation is analagous to cloud cover, which typically helps reflect solar radiation and reduce temperatures on the surface of the Earth. From the plot above, we see precipitation pretty randomly distributed, In the winter, snowfall can dramatically affect the temperature of the day. Since snow has a high albedo, it reflects a huge portion of the solar radiation that hits Earth back into space. Thus, less solar radiation reaches the Earth and absorbed by the ground, leading to less ground heating, less reabsorption of radiation into the atmosphere, and lower temperatures. Snowfall for most of the year is going to be near zero, but during the winter months above, we see an increase in snow amount.

The last factor is the date, which provides a seasonal component to the model. Date is a strong climatological variables that can help provide a strong temperature forecast, as typically we should see similar temperatures during similar times of the year. During the summer, we should see high temperatures in contrast to colder temperatures during the winter. Transitional seasons like spring and autumn should be somewhere in the middle. Because I am interested in the Nashville location, there should a strong seasonal, as opposed to non-seasonal regions like in the tropics.

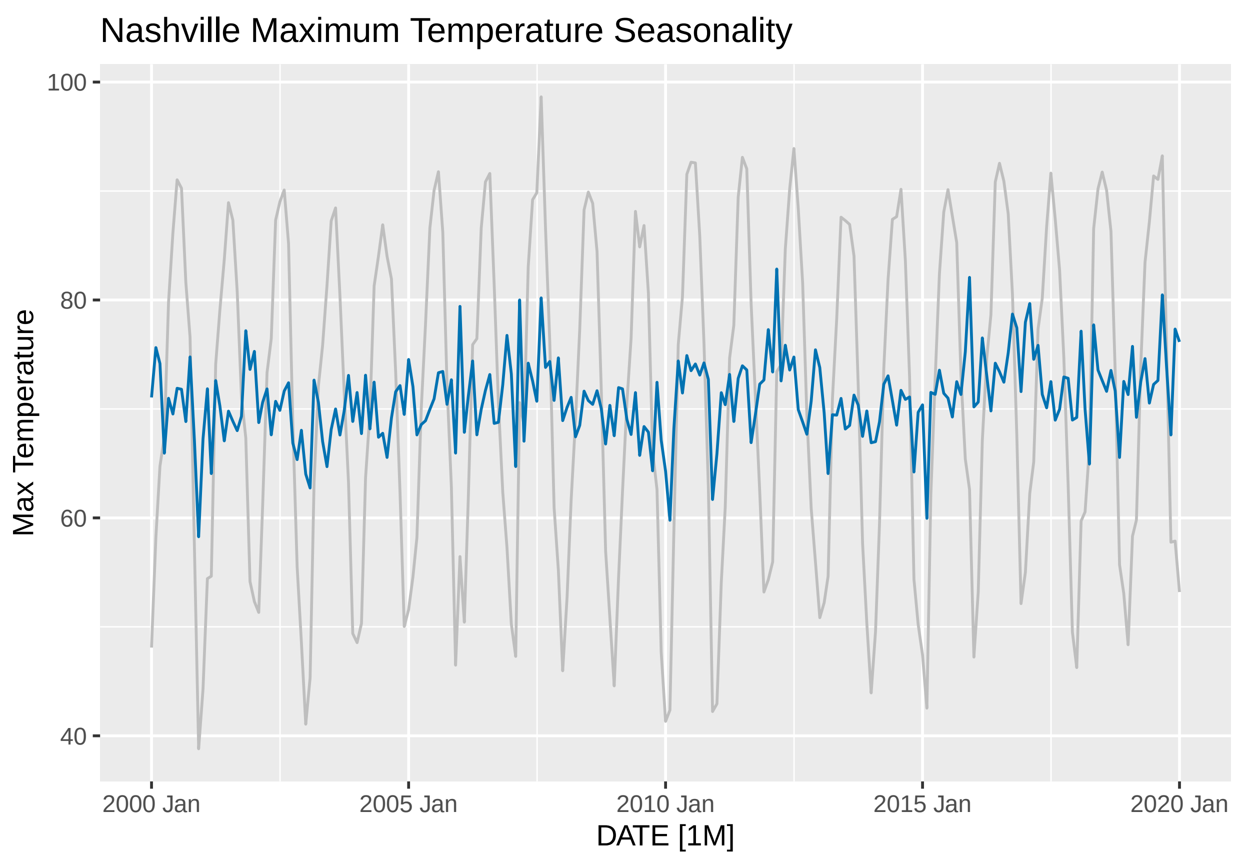
For this project, I will initially develop and compare models based on only the seasonality and the trend. Then, I will start incorporation the additional factors into the model, seeing if they provide stronger forecasts and allowing me to see if more complex models improve the forecast accuracy.

Decomposition

```r
dcmp <- weather_train |>
  model(stl = STL(TMAX ~ trend(window = 365) +
          season(window = 365),
        robust = TRUE))

components(dcmp) |>
  autoplot()
```



```r
components(dcmp) |>
  as_tsibble() |>
  autoplot(TMAX, colour = "gray") +
  geom_line(aes(y=season_adjust), colour = "#0072B2") +
  labs(y = "Max Temperature",
       title = "Nashville Maximum Temperature Seasonality")
```



```r
weather_train |>
  features(TMAX, unitroot_kpss)
```

```
# A tibble: 1 x 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1    0.0384         0.1
```

```r
weather_train |>
  features(TMAX, unitroot_ndiffs)
```

```
# A tibble: 1 x 1
  ndiffs
   <int>
1      0
```

The topmost plot of the STL decomposition shows the actual values of tmax in the trainind dataset. Similar to discussed above, there appears to be strong oscillations on a year-by year basis due to the changing seasons, with low temperatures in the winter and high temperatures in the summer. Looking at the trend, which I selected to be a year-by-year window, we see a gradual increase in temperature of about 3 degrees F. This can be indicative of several factors, including climate change, urban heat island effects, or other similarly related trends. With the trend window selected at 365, we can see a clear seasonality in the season-year plot. Again, the seasonal pattern appears consistent over time, suggesting that the yearly fluctuations in temperature is small. The bottom remainder/residual plot looks to have random fluctuations, is centered around 0, and looks similar to white noise. Thus, the decomposition is effectively capturing the main trend and seasonality structures in the dataset. However, there are some spikes, typically in the winter and summer that isn't captured by the seasonality or trend. With my modeling here, I hope to better capture these extreme events with the inclusion of other meteorological variables.

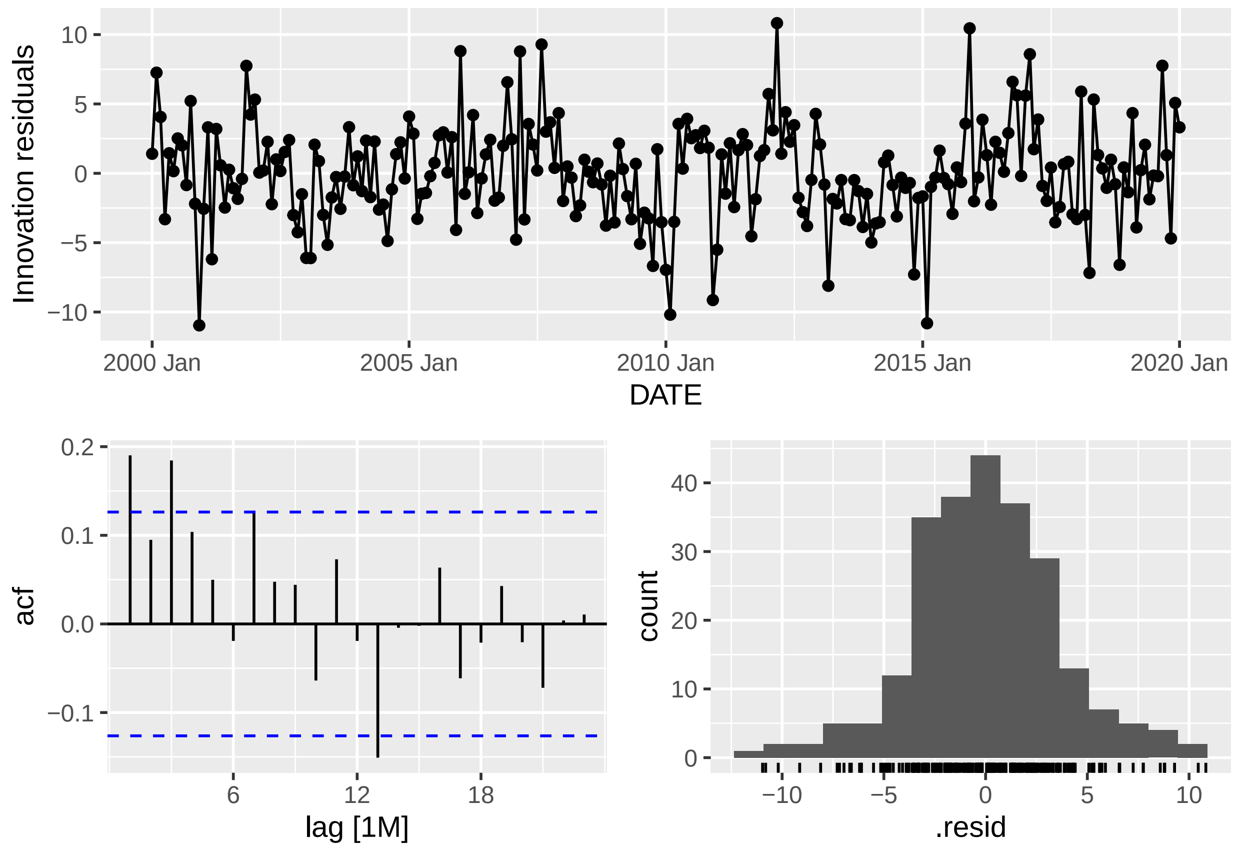
An important component to undersanding the viability of the data for ARIMA model input is knowing if the dataset requires differencing. For a valid model, we would like the dataset to be stationary and to no longer require seasonal-differencing. Here, I perform a kpss unitroot rest and a numer differencing unitroot test. The kpss_pvalue of .1 suggests that we may reject the null hypothesis, and that the data here is stationary and does not require differencing. The ndiffs value of 0 suggests that no seasonal differencing is requires. As such, we may proceed.

Model Selection

```r
tslm_date <- weather_train |>
  model(tslm = TSLM(
    TMAX ~ trend() + season()
  ))

report(tslm_date)
```

```
Series: TMAX
Model: TSLM

Residuals:
    Min      1Q  Median      3Q     Max
-10.962  -2.226  -0.162   2.169  10.825

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    46.675918   0.906478  51.492  < 2e-16 ***
trend()         0.013291   0.003424   3.882 0.000136 ***
season()year2   4.251183   1.154498   3.682 0.000289 ***
season()year3  13.964149   1.154453  12.096  < 2e-16 ***
season()year4  24.014029   1.154417  20.802  < 2e-16 ***
season()year5  31.616598   1.154392  27.388  < 2e-16 ***
season()year6  39.255780   1.154376  34.006  < 2e-16 ***
season()year7  41.749693   1.154371  36.167  < 2e-16 ***
season()year8  41.458982   1.154376  35.915  < 2e-16 ***
season()year9  35.654239   1.154392  30.886  < 2e-16 ***
season()year10 24.561431   1.154417  21.276  < 2e-16 ***
season()year11 12.605990   1.154453  10.919  < 2e-16 ***
season()year12  2.933236   1.154498   2.541 0.011728 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.695 on 228 degrees of freedom
Multiple R-squared: 0.9455, Adjusted R-squared: 0.9427
F-statistic: 329.9 on 12 and 228 DF, p-value: < 2.22e-16
```

```r
tslm_date |> gg_tsresiduals()
```



The report for the time series linear model only using trend and seasonality shows that for each month, TMAX is expected to increase by about 0.0133 units, and this effect is statistically significant (p-value < 0.05). Additionally all the seasonal component are statistically significant with p-values < 0.05. Overall, the model fits well, capturing about 94.55% of the variability. The residuals' spread suggests that the model might not capture all patterns or might have some outliers. It seems that the model is performing very well, capturing both seasonal and trend patterns.

From the residual plot, we see the residuals centered around 0, with no clear trend or seasonality, suggesting that the model has likely captured the major trend and seasonal components of the data. There are still some major fluctuations, however, that the model misses. In the ACF plot, there are bars exceeding the significance threshold, which indicate potential autocorrelation between variables and this will be looked at next. The residuals appear normally distributed, which is necessary as time series models assumes that the residuals are normally distributed.

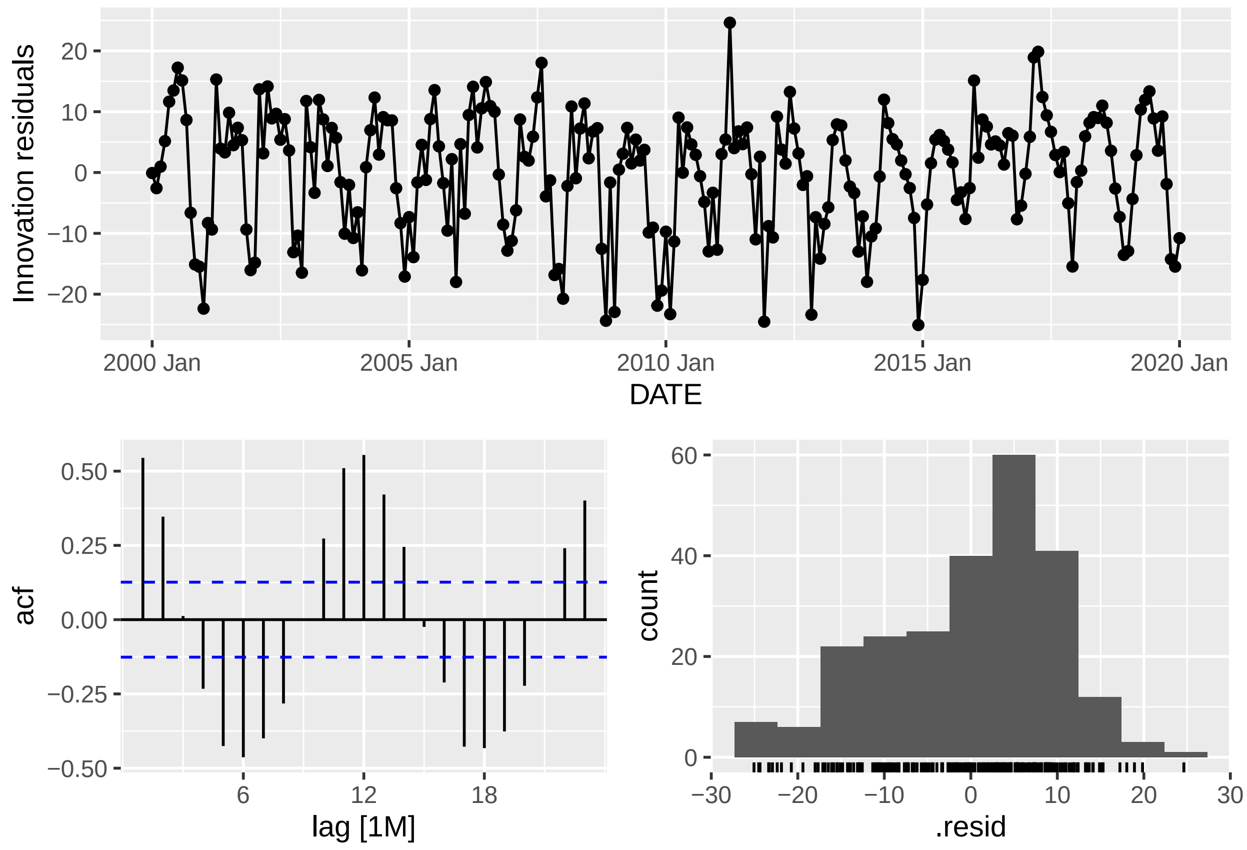
```r
model <- lm(TMAX ~ PRCP + SNOW + AWND, data=weather_train)
vif_values <- VIF(model)
print(vif_values)
```

```
    PRCP     SNOW     AWND 
1.050572 1.065912 1.085744 
```

```r
tslm_date_prcp_snow_awnd <- weather_train |>
  model(tslm = TSLM(
    TMAX ~ DATE + PRCP + SNOW + AWND
  ))

report(tslm_date_prcp_snow_awnd)
```

```
Series: TMAX
Model: TSLM

Residuals:
    Min      1Q  Median      3Q     Max
-25.062  -7.331   1.976   7.348  24.624

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.222e+02  5.798e+00  21.073  < 2e-16 ***
DATE        -9.272e-05  3.065e-04  -0.302    0.763
PRCP         3.738e-01  2.903e-01   1.288    0.199
SNOW        -4.021e+00  5.623e-01  -7.152 1.07e-11 ***
AWND        -7.694e+00  5.219e-01 -14.741  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10 on 236 degrees of freedom
Multiple R-squared: 0.5869, Adjusted R-squared: 0.5799
F-statistic: 83.83 on 4 and 236 DF, p-value: < 2.22e-16
```

```r
tslm_date_prcp_snow_awnd |> gg_tsresiduals()
```



Here, we conduct a study to see if any of the meteorological variables are autocorrelated. Since none of the VIFs > 10 and they are all evenly distributed among the variables, we can continue using all the variables here.

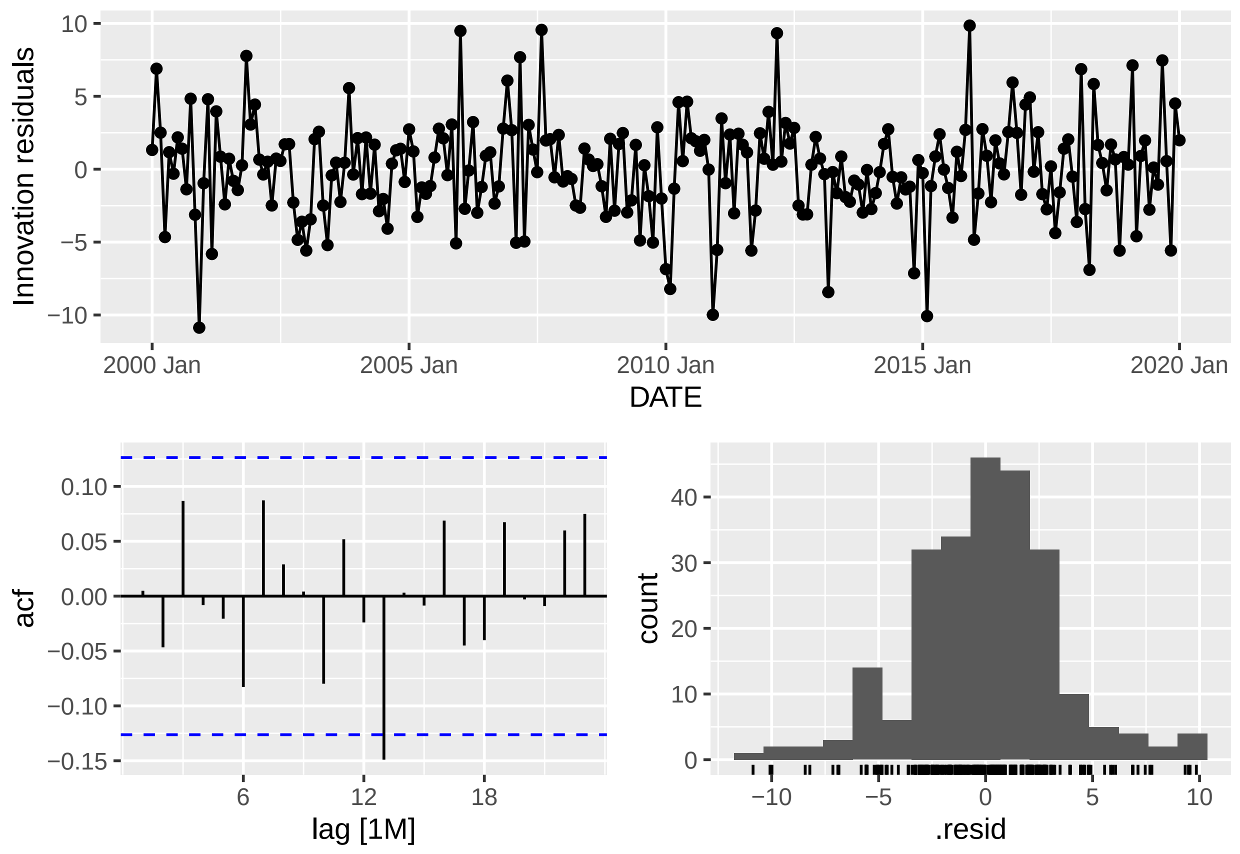
The report for the time series linear model only using date, precipitation, average wind, and snowfall shows that for each month. Here the coefficient for date and precipitation are not significant - indicating that precipitation does not have a significant impact on TMAX. Snow and wind do however, and both are negatively correlated to the maximum temperature of a day in Nashville. The adjusted R-square value here shows that 57.99% of the variability in the maximum temperature is captured by the model here. Overall, the model is statistically significant with a p-value $< .05$.

From the residual plot, we see the residuals centered around 0, with no clear trend or seasonality, suggesting that the model has likely captured the major trend and seasonal components of the data. There are still some major fluctuations, however, that the model misses. The residuals appear normally distributed again. The ACF plot shows a sinusoidal and bars past the significance level, indicating potential seasonality not captured by the model.

```
arima_date <- weather_train |>
  model(arima = ARIMA(
    TMAX ~ trend() + season()
  ))

report(arima_date)
```

```
Series: TMAX
Model: LM w/ ARIMA(1,0,1)(2,0,0)[12] errors

Coefficients:
         ar1      ma1      sar1      sar2  trend()  season()year2  season()year3
      0.7453  -0.5952  -0.0271  -0.2513   0.0131         4.0650        14.0904
s.e.  0.1486   0.1769   0.0679   0.0692   0.0041         0.7887         0.8075
      season()year4  season()year5  season()year6  season()year7  season()year8
            24.1197        31.5440        39.3143        41.7448        41.4909
s.e.         0.8214         0.8304         0.8352         0.8368         0.8351
      season()year9  season()year10  season()year11  season()year12  intercept
            35.6197         24.5412         12.7234          3.0162    46.6840
s.e.         0.8302          0.8212          0.8079          0.7875     0.7846

sigma^2 estimated as 12.36:  log likelihood=-636.97
AIC=1309.94   AICc=1313.02   BIC=1372.66
```

```r
arima_date |> gg_tsresiduals()
```



```r
augment(arima_date) |>
  filter(.model == "arima") |>
  features(.innov, ljung_box, lag=24, dof=1)
```

```
# A tibble: 1 x 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 arima     20.4     0.617
```

Here, we autofit into a linear model with ARIMA errors. The armia errors are ARIMA(1,0,1)(2,0,0)[12], indicating a non-seasonal autoregressive term of order 1, a non-seasonal moving average term of order 1, and seasonal autoregressive terms of order 2 with a seasonal period of 12 months. For the coefficients, ar1 and ma1 suggest positive autocorrelation and negative moving average behavior in the series. sar1 and sar2 indicate a negative seasonal autoregressive effect. The trend tell us that there is a linear trend in max temperature, with an increase of .0131.

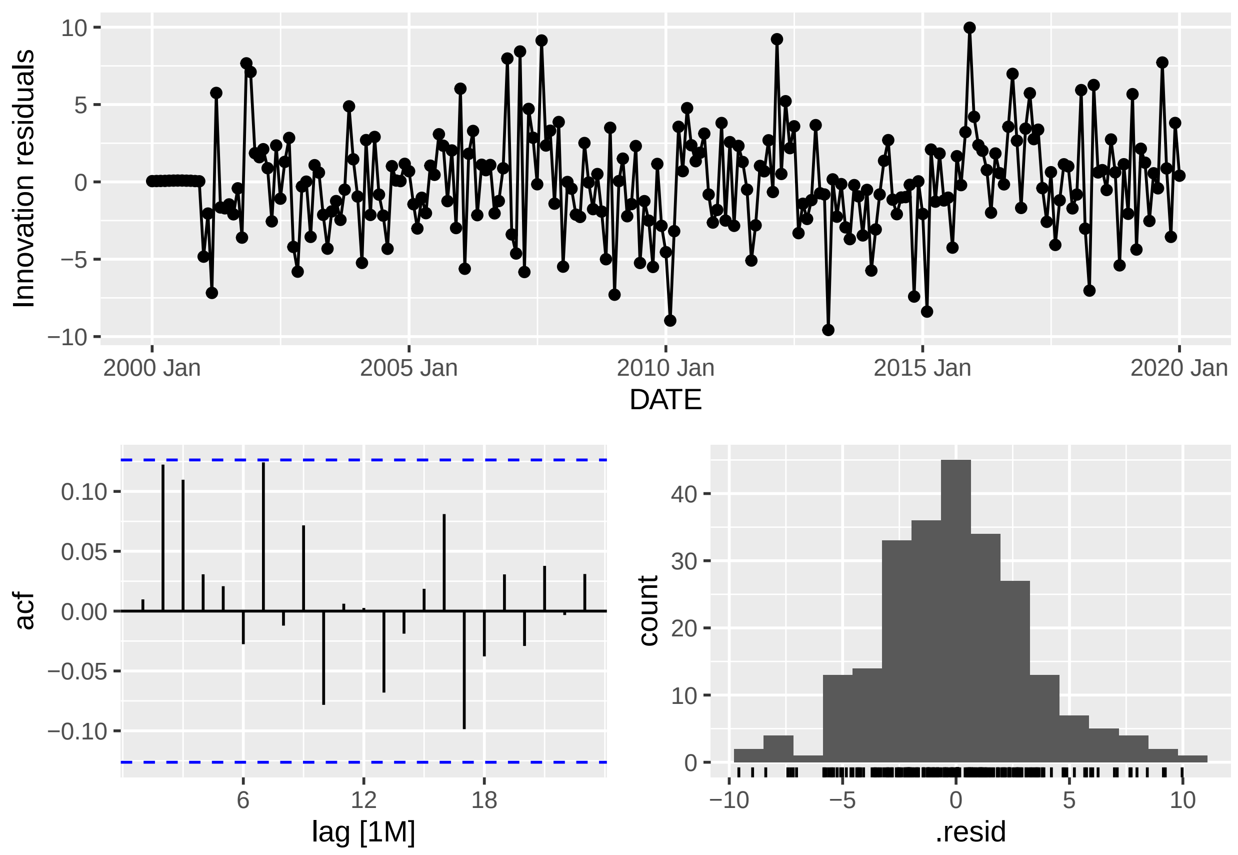
From the residual plot, we see the residuals centered around 0, with no clear trend or seasonality, suggesting that the model has likely captured the major trend and seasonal components of the data. There are still some major fluctuations, however, that the model misses. The residuals appear normally distributed again. The ACF plot shows that all the bars are are within the significance level.

The ljung box test with a p-value $>$ .05 indicates that this model is not significantly different from white noise, which is an assumption made for ARIMA models to be used.

```r
arima_date_prcp_snow_awnd <- weather_train |>
  model(arima = ARIMA(
    TMAX ~ DATE + PRCP + SNOW + AWND
  ))

report(arima_date_prcp_snow_awnd)
```

```
Series: TMAX
Model: LM w/ ARIMA(0,0,1)(2,1,2)[12] errors

Coefficients:
         ma1    sar1     sar2     sma1    sma2   DATE     PRCP     SNOW
      0.1479  0.2143  -0.2557  -1.1008  0.2022  4e-04  -0.0312  -1.3155
s.e.  0.0615  0.2479   0.0848   0.2431  0.2547  1e-04   0.0963   0.2232
         AWND
      -0.1741
s.e.   0.3381

sigma^2 estimated as 11.97:  log likelihood=-616.63
AIC=1253.27   AICc=1254.28   BIC=1287.61
```

```r
arima_date_prcp_snow_awnd |> gg_tsresiduals()
```



```r
augment(arima_date_prcp_snow_awnd) |>
  filter(.model == "arima") |>
  features(.innov, ljung_box, lag=24, dof=4)
```

```
# A tibble: 1 x 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 arima     21.1     0.389
```

Here, we autofit into a linear model with ARIMA errors. The armia errors are ARIMA(2,0,0)(2,0,0)[12], indicating 1 moving average in the non-seasonal part, 2 autoregressive terms in the seasonal part, 1 differencing in the seasonal part, 2 moving average terms in the seasonal part, and a seasonal period of 12 months. The coefficient for precipitation is -0.0312 and the coefficient for snow is -1.3155, suggesting that an increase in precipitation/snow is associated with a decrease in max temperature. The estimated variance of the residuals is 11.97.

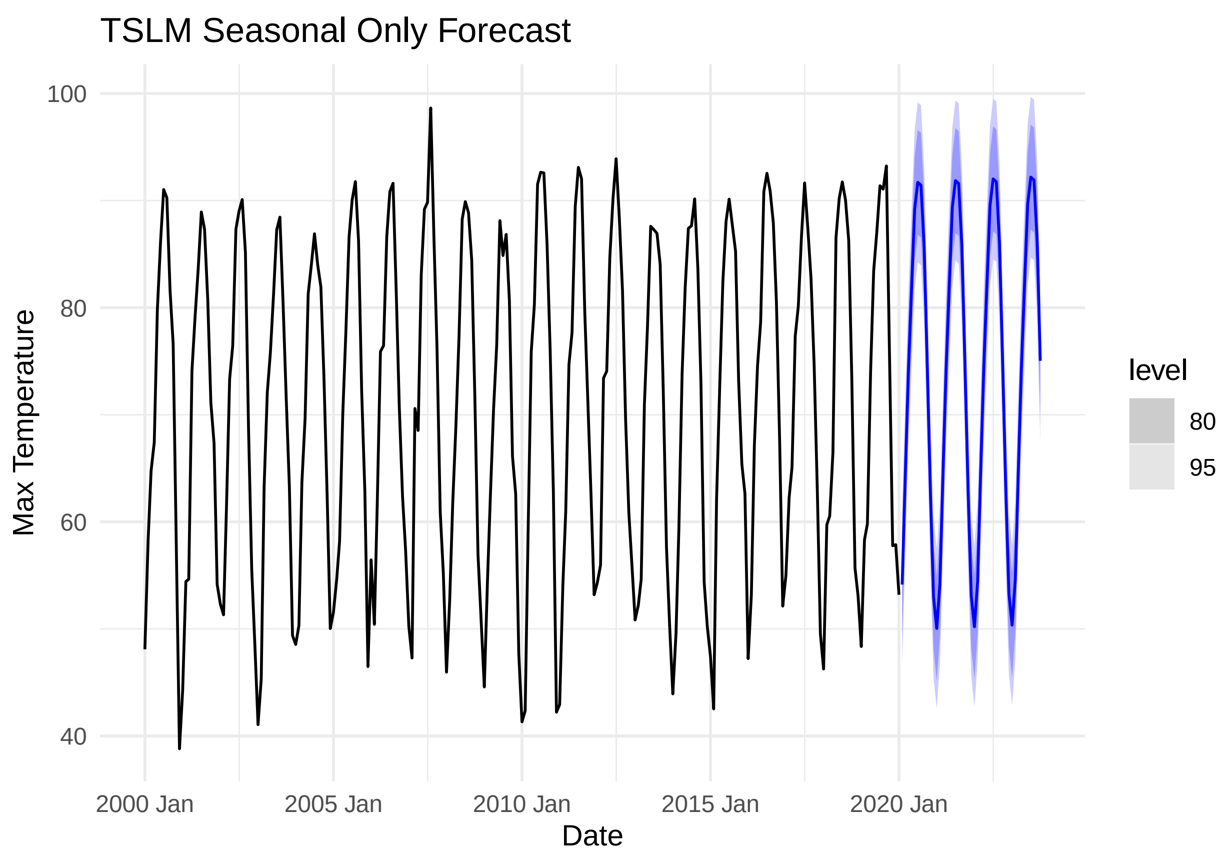
From the residual plot, we see the residuals centered around 0, with no clear trend or seasonality, suggesting that the model has likely captured the major trend and seasonal components of the data. There are still some major fluctuations, however, that the model misses. The residuals appear normally distributed again. The ACF plot shows that all the bars are are within the significance level. The residuals at the beginning of the plot are very near 0, which is interesting to see, suggesting that the model captures this data very well compared to all the other models.

The ljung box test with a p-value $>$ .05 indicates that this model is not significantly different from white noise, which is an assumption made for ARIMA models to be used.

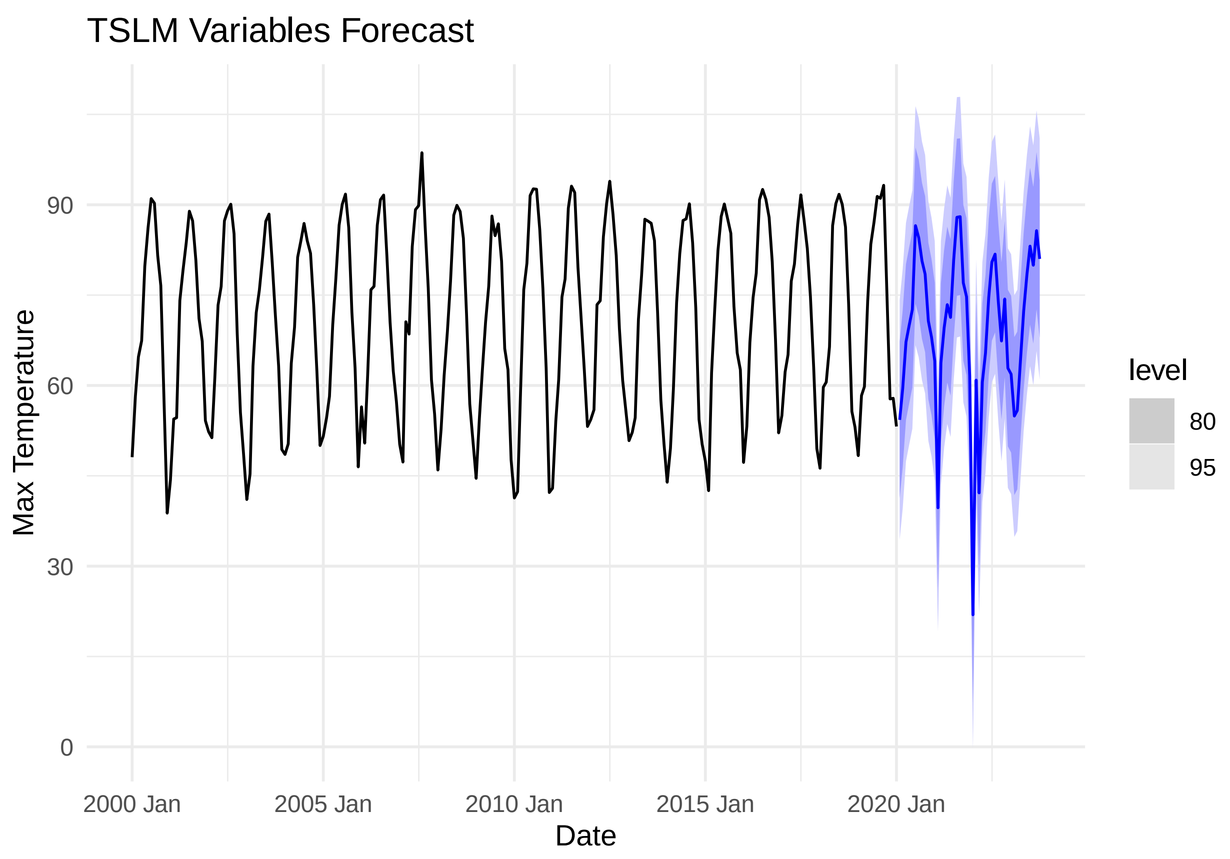
Forecast

```r
all_models <- weather_train |>
  model(
  tslm = TSLM(
    TMAX ~ trend() + season()
  ),
  tslm_all = TSLM(
    TMAX ~ DATE + PRCP + SNOW + AWND
  ),
  arima = ARIMA(
    TMAX ~ trend() + season()
  ),
  arima_all = ARIMA(
    TMAX ~ DATE  + PRCP + SNOW + AWND
  ),
)

fc <- all_models |> forecast(new_data = weather_test)
```

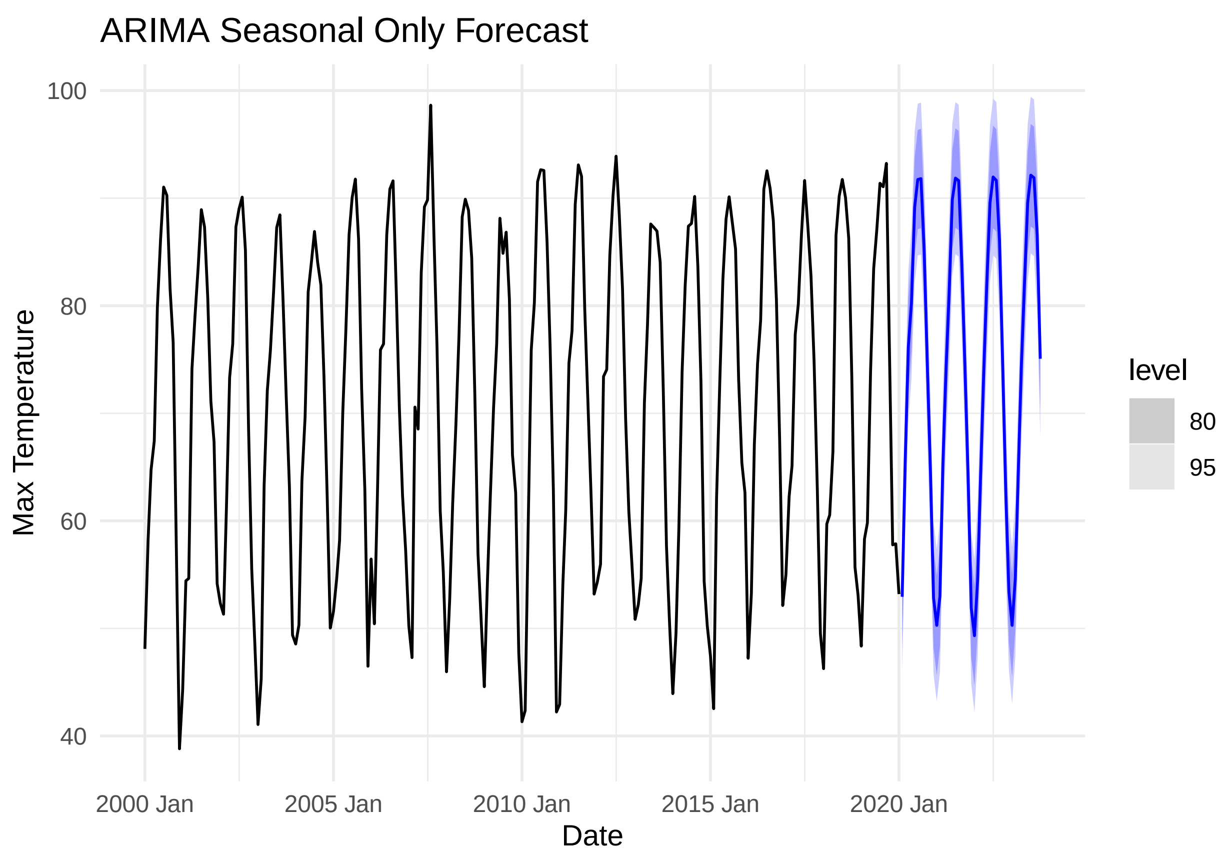
```r
fc_tslm_season <- fc |> filter(.model == "tslm")
fc_tslm_all <- fc |> filter(.model == "tslm_all")
fc_arima_season <- fc |> filter(.model == "arima")
fc_arima_all <- fc |> filter(.model == "arima_all")

autoplot(weather_train, series = "Observed") +
  autolayer(fc_tslm_season, series = "TSLM Seasonal + Trend  Forecast") +
  theme_minimal() +
  labs(title = "TSLM Seasonal Only Forecast",
    x = "Date",
    y = "Max Temperature")
```



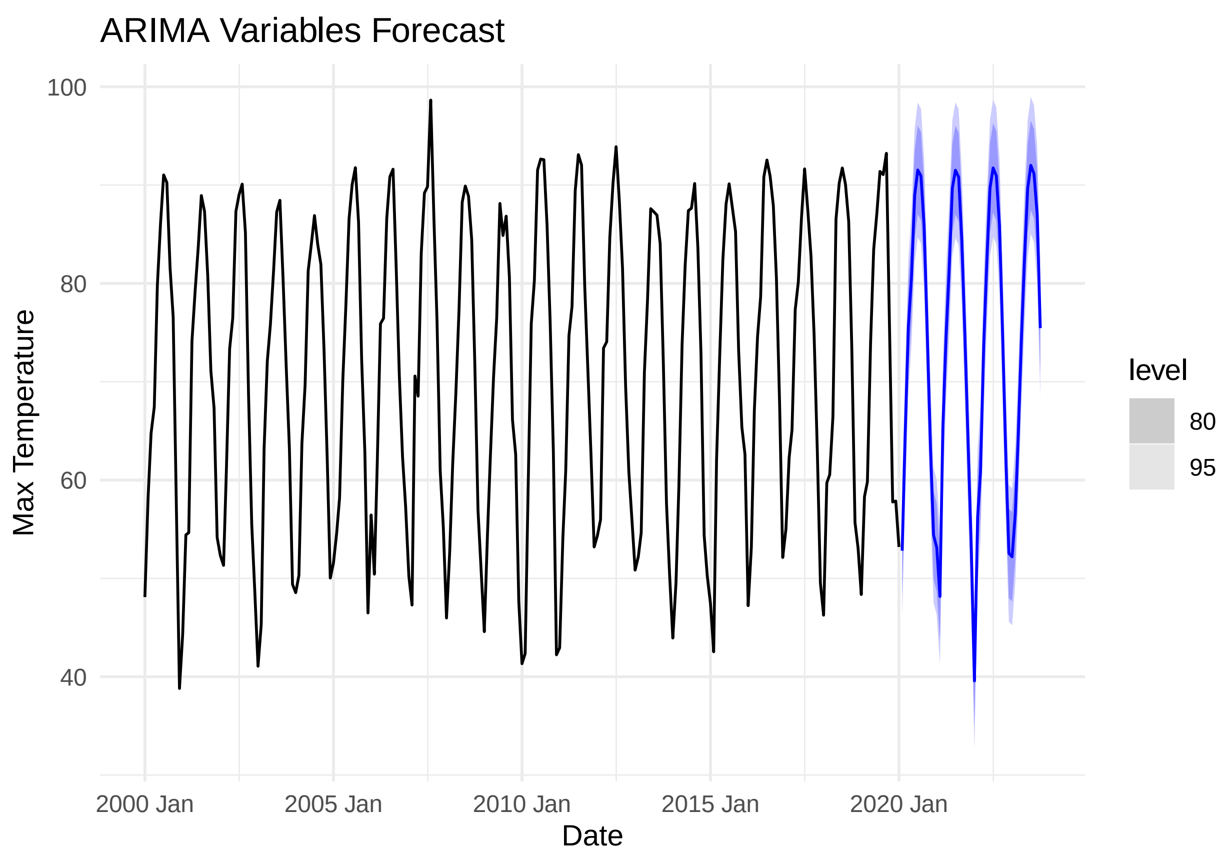
```r
autoplot(weather_train, series = "Observed") +
  autolayer(fc_tslm_all, series = "TSLM Variables Forecast") +
  theme_minimal() +
  labs(title = "TSLM Variables Forecast",
    x = "Date",
    y = "Max Temperature")
```



```r
autoplot(weather_train, series = "Observed") +
  autolayer(fc_arima_season, series = "ARIMA Seasonal + Trend Forecast") +
  theme_minimal() +
  labs(title = "ARIMA Seasonal Only Forecast",
    x = "Date",
    y = "Max Temperature")
```



```r
autoplot(weather_train, series = "Observed") +
  autolayer(fc_arima_all, series = "ARIMA Variables Forecast") +
  theme_minimal() +
  labs(title = "ARIMA Variables Forecast",
    x = "Date",
    y = "Max Temperature")
```



```r
fc |> accuracy(weather_test) |>
  select(.model, RMSE, ME, MAE)
```

```
# A tibble: 4 x 4
  .model      RMSE     ME   MAE
  <chr>      <dbl>  <dbl> <dbl>
1 arima       3.20 -0.383  2.35
2 arima_all   3.11 -0.131  2.38
3 tslm        3.20 -0.399  2.44
4 tslm_all   10.3   4.11   8.44
```

```r
fc |> accuracy(weather_test, list(crps = CRPS))
```

```
# A tibble: 4 x 3
  .model    .type  crps
  <chr>     <chr> <dbl>
1 arima     Test   1.72
2 arima_all Test   1.70
3 tslm      Test   1.76
4 tslm_all  Test   5.82
```

The TSLM seasonal and trend forcast has historical data exhibits clear seasonality with regular peaks and troughs of the max temperature, as discussed earlier. At first glance, the model performs very well, capturing the seasonality of the data and with approximately the same magnitude of max temperaturs as expected during the season. The confidence intervals here are quite tight, suggesting relative confidence in predictions.

Similarly, the TSLM data forecast exhibits the same seasonality, while also incorporating other meteorological variables. Here, the magnitude of the peaks are less than the previous model, with more irregularities. Additionally, the confidence intervals are wider, suggesting less confidence in this model.

Next, the lm ARIMA model only considering seasonality and trend. The plots are consistent and look very similar to the TSLM seasonal and trend forecast. Again, the model captures the seasonality of the data with similar magnitudes with relatively tight confidence intervals.

Lastly, the lm ARIMA model with meteorological variables. This model follows the same trends as the previous lm ARIMA model and the TSLM seasonal and trend model. However, Near the winter of 2022, there is a significantly different prediction for the max temperature. This indicates that the model is utilzing the data to make new inferences about what might occur during this time based on previous information.

Here, we analyze the model performance metrics. For RMSE, arima_all has the lowest RMSE (3.110366), followed closely by arima and tslm. tslm_all appears to have the highest RMSE, indicating poor performance. For Me, arima_all again has the closest to 0, suggesting it has the smallest bias. tslm_all has a high value of 4, indicating consistent over-prediction. For MAE, arima has the lowest, followed by arima_all and tslm. tslm_all has a significantly higher MAE. For CRPS, arima_all has the lowest value, indicating the best probabilistic forecast accuracy. tslm_all has the worst performance in this metric. The arima_all model generally performs the best across most of the metrics, with the lowest RMSE, ME close to zero, a low MAE, and the lowest CRPS. In contrast, the tslm_all model consistently performs the worst, particularly with a much higher RMSE and CRPS. The arima and tslm models perform similarly, with their metrics relatively close to each other. Given this, arima_all should be the chosen model, as it demonstrates the best overall performance across the metrics here.

Interpretation

Forecasts are based on historical data, which captures the trends, cycles, and patterns exhibited by the time series up to the present moment. As such, understanding past behavior can provide insights into future tendencies, especially in systems where certain patterns repeat over time. Here, I analyzed the forecasting performance that common weather stations have on the maximum temperatures in Nashville. A big knowledge base for this is looking at past climatology for a specific area. If you use climatology as a baseline estimation, the overall performance will already be pretty great, as shown by the seasonal + trend models I looked at here. The qeustion I wanted to answer is if using additional variables would help create an even more robust forecast. In agreement with my prior hypothesis, the inclusion of precipitation, snow, and average wind speed meteorological variables proved valuable. In both the TSLM and LM with ARIMA errors, the inclusion of these factors improved performance, even if only slightly. Even so, extreme weather events won't be fully captured by the model, as it will tend to create more safe predictions that are an average more correct than not. This is important because you don't want forecasts constantly raising alarms about extreme weather, as you will seem alarmist and poor at forecasting. On the flip side, however, a forecaster that fails to inform the public won't be seen as trustworthy in the opposite direction.

No matter how sophisticated a forecasting model is, there's always uncertainty about the future. Unforeseen events can lead to differences between the forecasted values and actual outcomes.

Weather is incredibly unpredictable. It is the source for what is known as the butterfly effect, that a small perturbation in a system far away can have big consequences elsewhere. In every model, it failed to capture the mild winter of 2022-2023. Here the model isn't capturing the global variability in weather, but perhaps it may learn the subtle underlying distributions in the season/trend. Last winter, La Nina was in effect, leading to overall higher temperatures in Nashville as well as the great lakes region/pacific northwest. Weather can also change very quickly. If this was a daily forecast of maximum temperature rather than monthly averages, each model would be much weaker because climatology is less effective. Daily weather is much more of a strong predictor here. However, the station data fails to provide a lot of this information, including air mass information, type of precipitation, cloud cover, and other variables. Including as many relevant variables as possible will lead to a better forecast. Knowing the current weather conditions and understanding where the model fails will also lead to a better forecast.