Meeting 13 Presentation

Daniel L.

5/13/2022

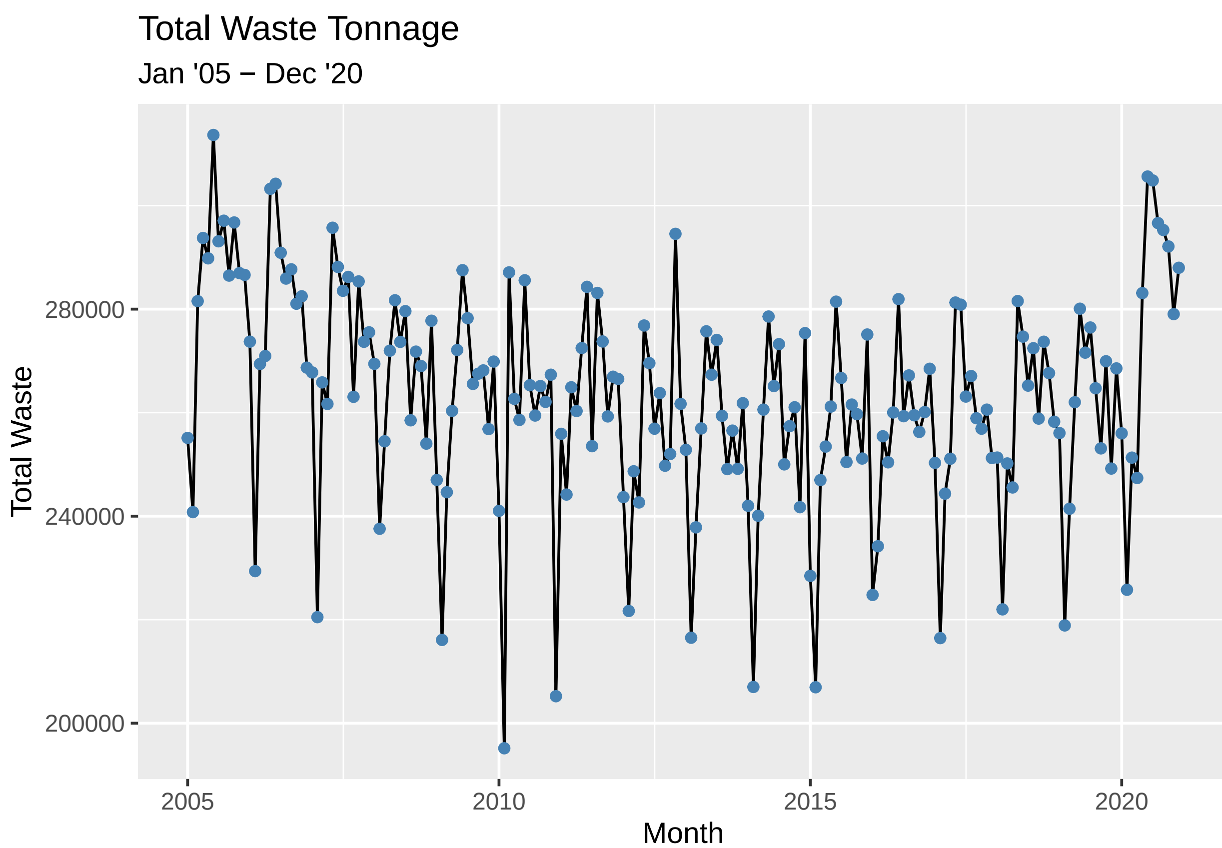
Dynamic Regression

When we estimate the parameters from the model, we need to minimize the sum of squared ϵ_t values.

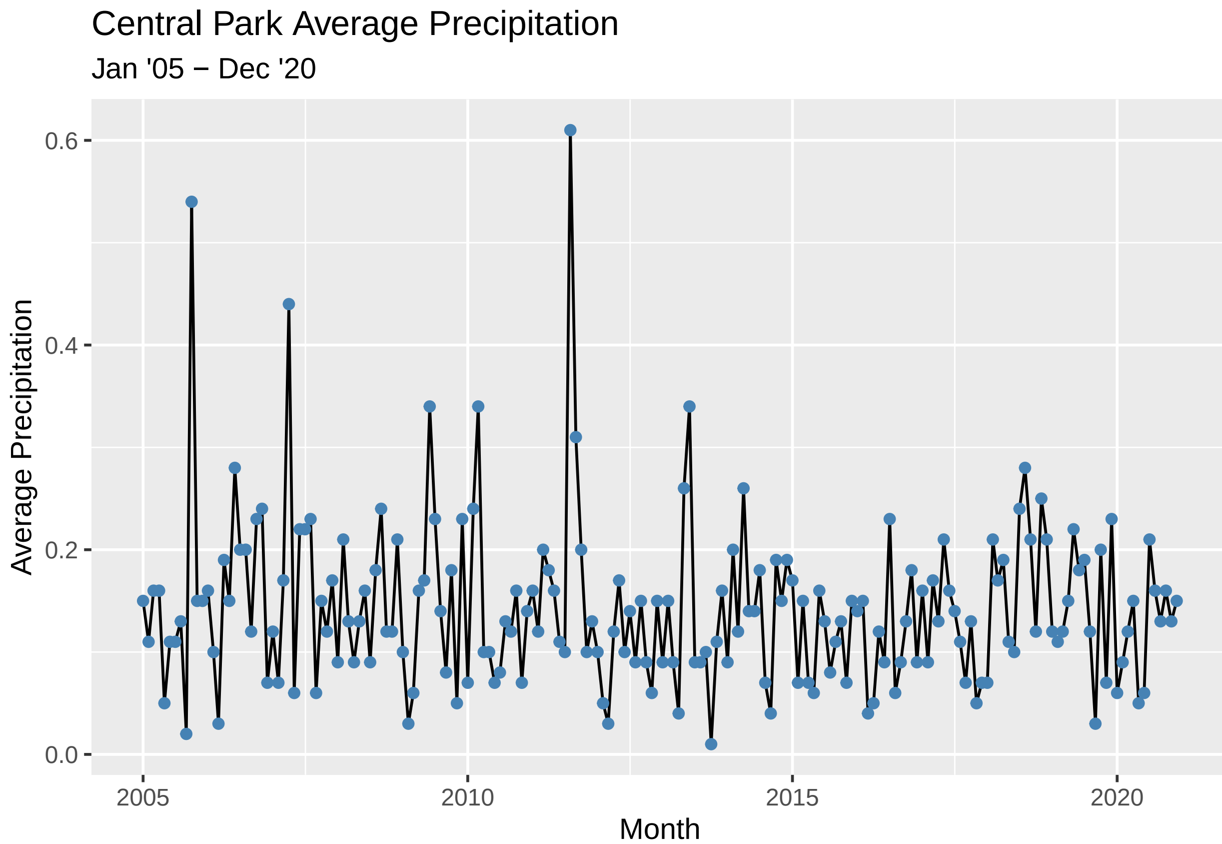
An important consideration when estimating a regression with ARMA errors is that all of the variables in the model must first be stationary. Thus, we first have to check that y_t and all of the predictors appear to be stationary. If we estimate the model when any of these are non-stationary, the estimated coefficients will not be consistent estimates (and therefore may not be meaningful).

Plots of the variables

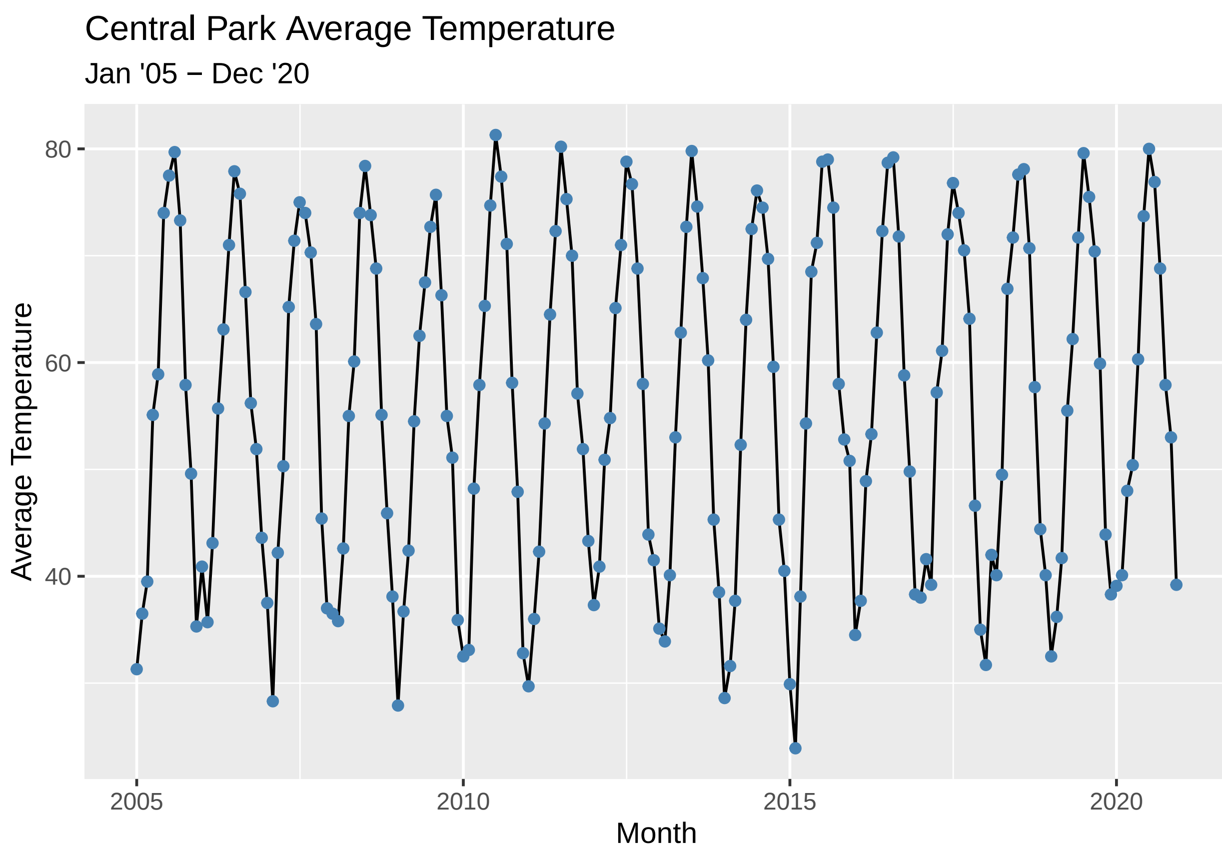
```
final_nyc_data %>% ggplot(., mapping = aes(x = month, y = total_waste_total)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       y = "Total Waste",
       title = "Total Waste Tonnage",
       subtitle = "Jan '05 - Dec '20")
```



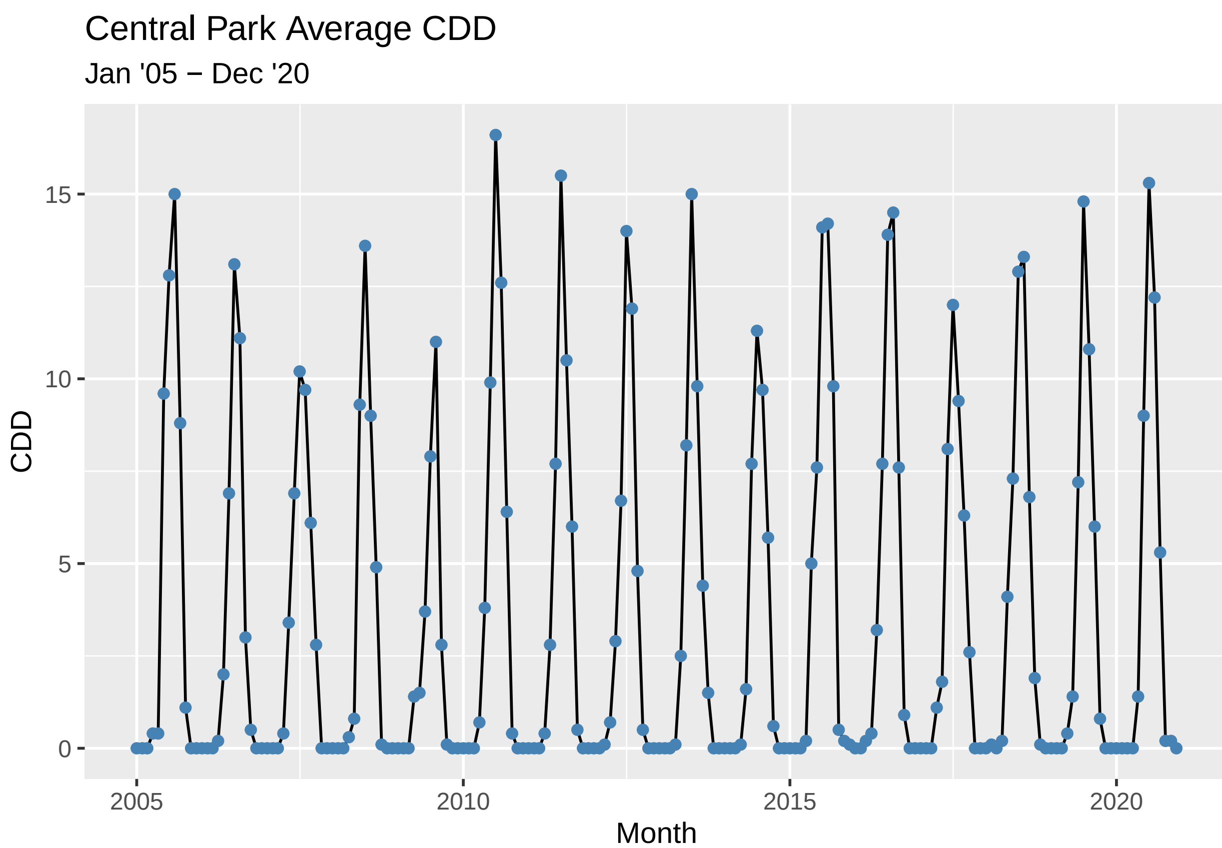

```
final_nyc_data %>% ggplot(., mapping = aes(x = month, y = avg_precip)) + geom_line() +
  geom_point(color="steelblue")+
  labs(x = "Month",
       y = "Average Precipitation",
       title = "Central Park Average Precipitation",
       subtitle = "Jan '05 - Dec '20")
```



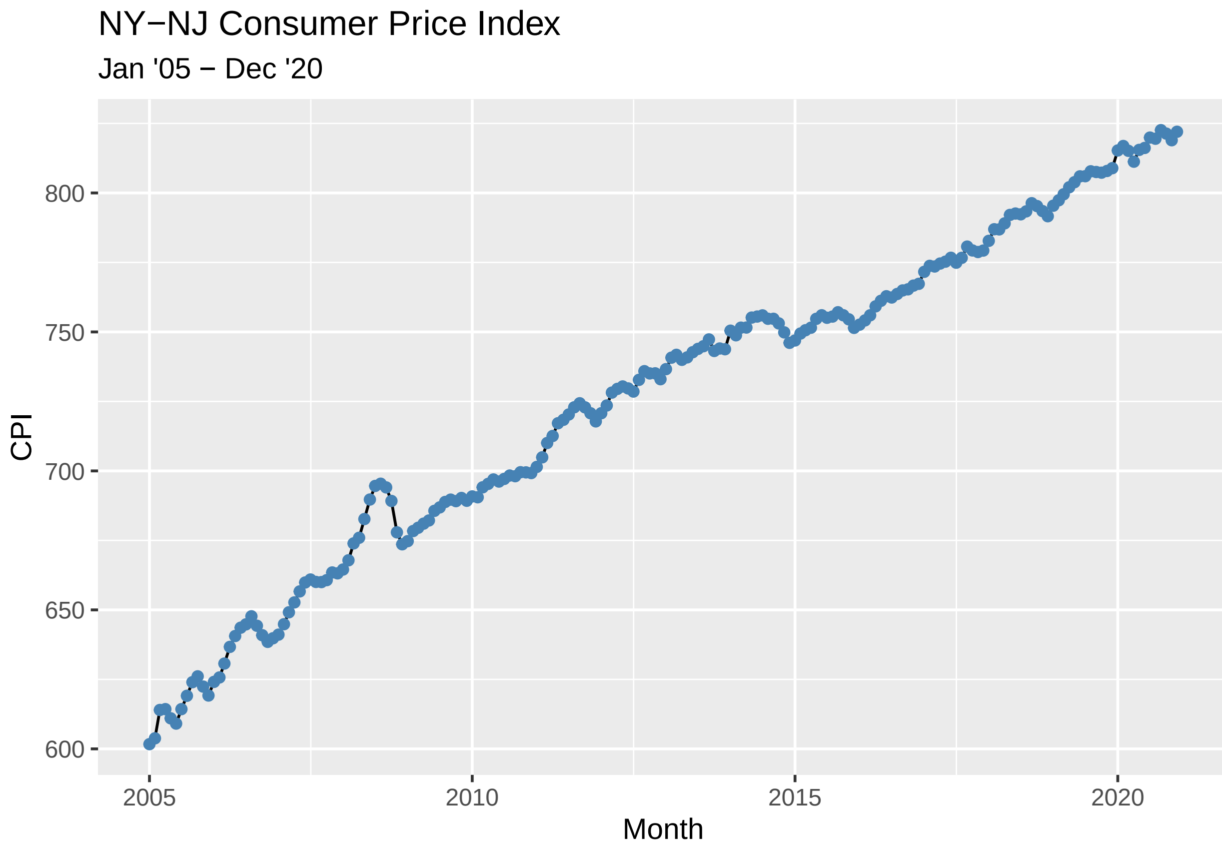

```
final_nyc_data %>% ggplot(., mapping = aes(x = month, y = avg_temp)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       y = "Average Temperature",
       title = "Central Park Average Temperature",
       subtitle = "Jan '05 - Dec '20")
```



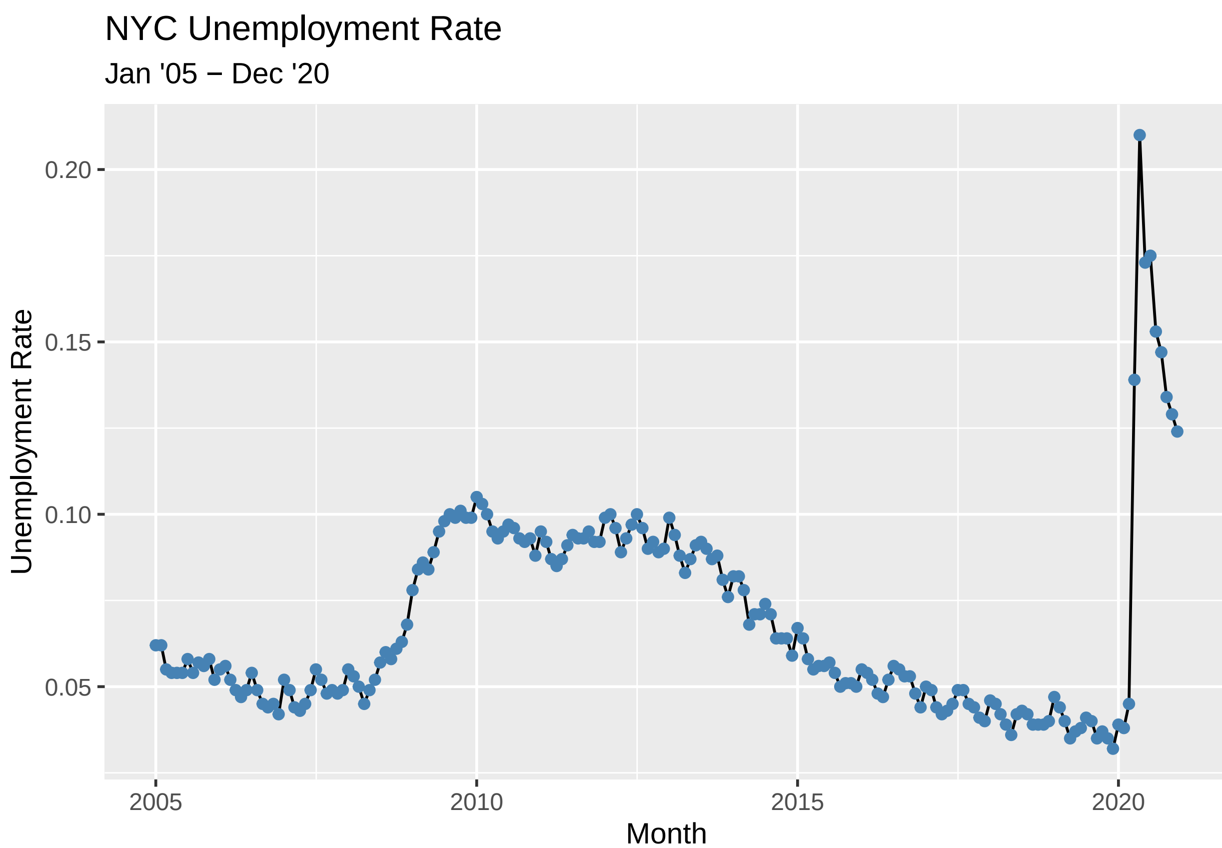

```r
final_nyc_data %>% ggplot(., mapping = aes(x = month, y = avg_CDD)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       y = "CDD",
       title = "Central Park Average CDD",
       subtitle = "Jan '05 - Dec '20")
```




```
final_nyc_data %>% ggplot(., mapping = aes(x = month, y = cpi)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       y = "CPI",
       title = "NY-NJ Consumer Price Index",
       subtitle = "Jan '05 - Dec '20")
```




```
final_nyc_data %>% ggplot(., mapping = aes(x = month, y = unemp_rate)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       y = "Unemployment Rate",
       title = "NYC Unemployment Rate",
       subtitle = "Jan '05 - Dec '20")
```



Degree days are measures of how cold or warm a location is. A degree day compares the mean (the average of the high and low) outdoor temperatures recorded for a location to a standard temperature, usually 65° Fahrenheit (F) in the United States. The average DD per month, from 2005 through 2020, is included in our data set.

The NYC unemployment rate was provided by a NYS labor site

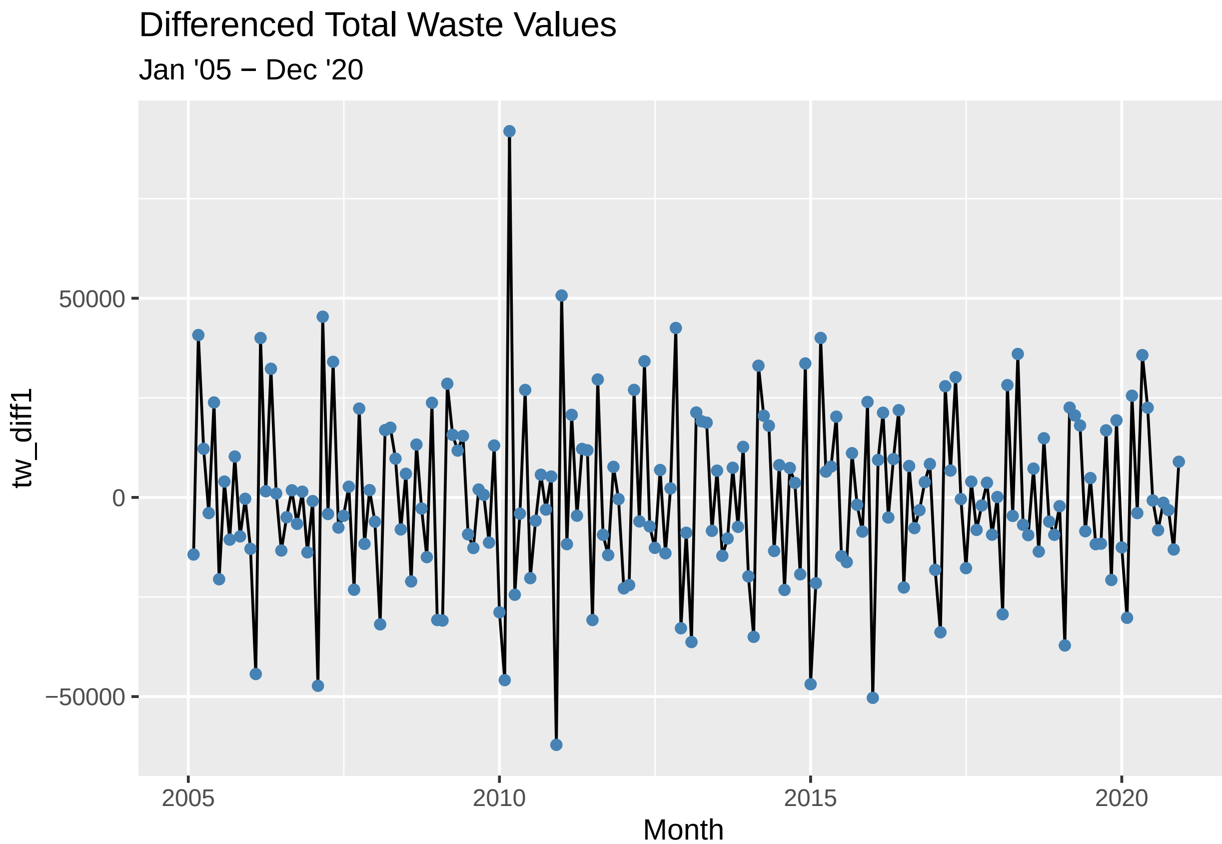
CPI for "All-Urban Consumers", for NYS-NJ region provided by the BLS

Differencing three variables

```
final_nyc_data_small %>%
  mutate(tw_diff1 = difference(total_waste_total,
                               differences = 1,
                               order_by = month_num)) %>%
  ggplot(.,
         mapping = aes(x = month,
                       y = tw_diff1)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       title = "Differenced Total Waste Values",
       subtitle = "Jan '05 - Dec '20")
```

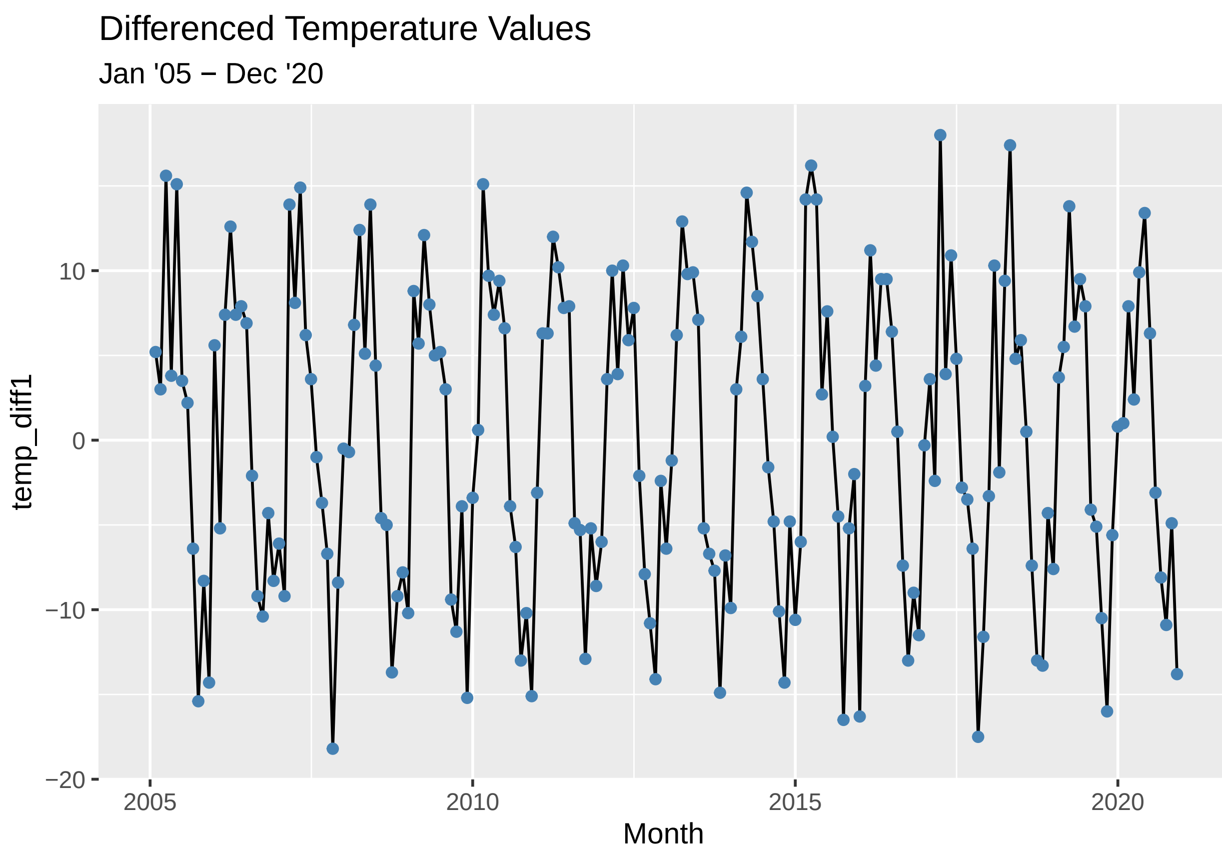
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

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



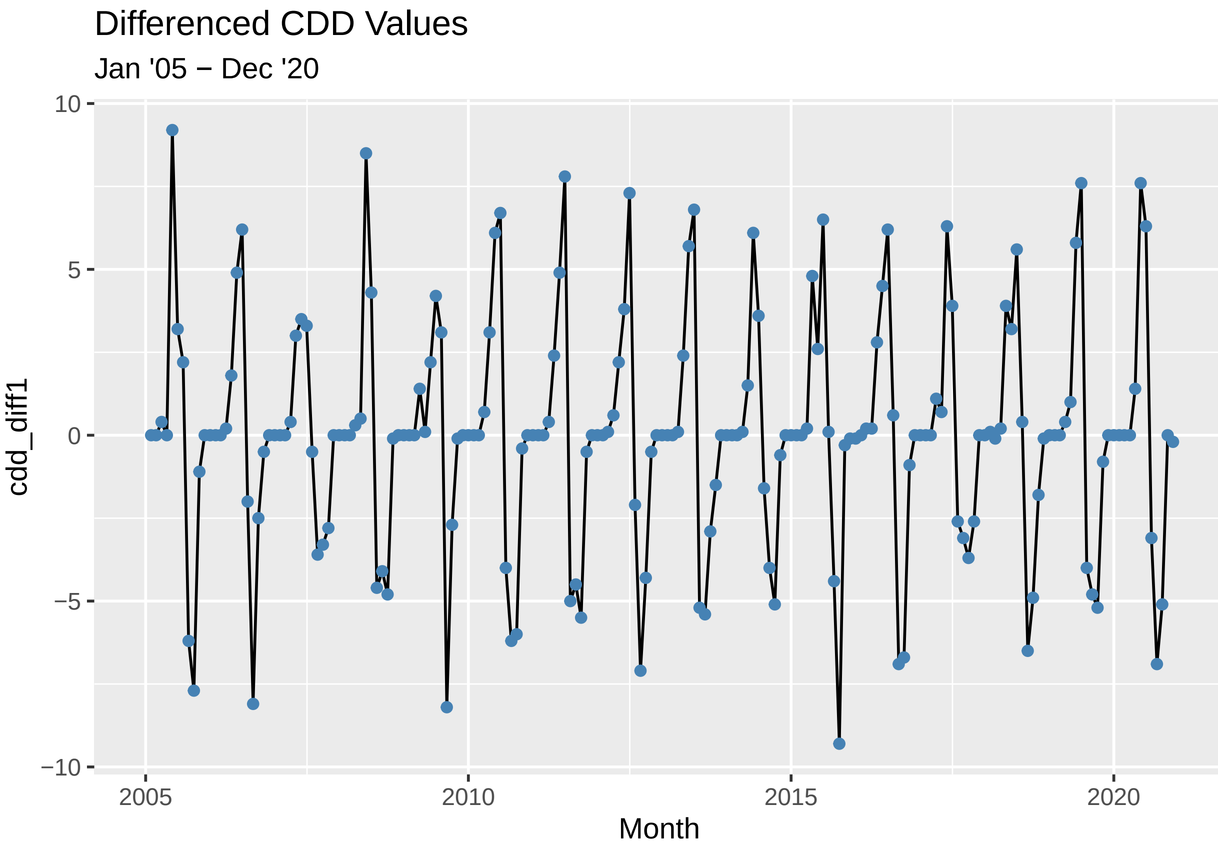

```
final_nyc_data_small %>%
  mutate(temp_diff1 = difference(avg_temp,
                                 differences = 1,
                                 order_by = month_num)) %>%
  ggplot(.,
         mapping = aes(x = month,
                       y = temp_diff1)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       title = "Differenced Temperature Values",
       subtitle = "Jan '05 - Dec '20")
```

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




```
final_nyc_data_small %>%
  mutate(cdd_diff1 = difference(avg_CDD,
                                differences = 1,
                                order_by = month_num)) %>%
  ggplot(.,
         mapping = aes(x = month,
                       y = cdd_diff1)) + geom_line() +
  geom_point(color="steelblue") +
  labs(x = "Month",
       title = "Differenced CDD Values",
       subtitle = "Jan '05 - Dec '20")
```

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



First attempt at a TSLM fit

https://otexts.com/fpp3/forecasting.html https://otexts.com/fpp3/forecasting-regression.html

The TSLM() function fits a linear regression model to time series data. It is similar to the lm() function which is widely used for linear models, but TSLM() provides additional facilities for handling time series.

```
lin_model_fit <- nyc_ts_2 %>%
  model(
    linear = TSLM(tw_diff1 ~ cpi + unemp_rate + avg_precip + temp_diff1 + cdd_diff1)
    # exponential = TSLM(log(tw_diff1) ~  cpi + unemp_rate + avg_precip + temp_diff1 + cdd_diff1)
  )
report(lin_model_fit)
```

```
## Series: tw_diff1
## Model: TSLM
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -49957 -12098   1057  10983  67896
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13483.73   18714.96  -0.720  0.47214
```

```
## cpi             10.89      25.08   0.434  0.66461
## unemp_rate   14878.05   50233.60   0.296  0.76743
## avg_precip   32610.92   17840.63   1.828  0.06918 .
## temp_diff1    1152.36     210.53   5.474 1.42e-07 ***
## cdd_diff1    -1612.58     539.13  -2.991  0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19840 on 185 degrees of freedom
## Multiple R-squared: 0.1564,  Adjusted R-squared: 0.1336
## F-statistic:  6.86 on 5 and 185 DF, p-value: 6.7559e-06
```

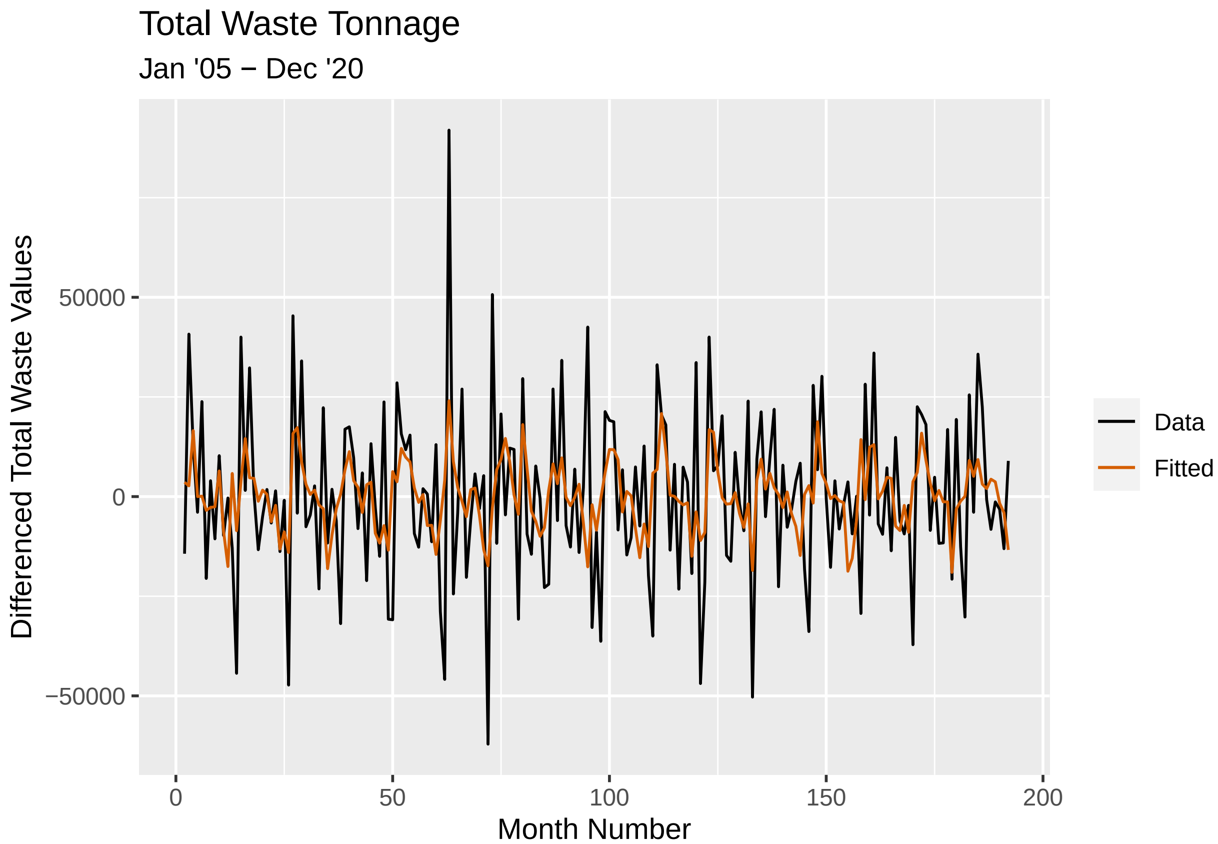
```
# fit_trends
#
# nyc_ts_2 %>%
#   autoplot(total_waste_total) +
#   geom_line(data = fitted(fit_trends),
#             aes(y = .fitted, colour = .model)) +
#   autolayer(fit_trends, alpha = 0.5, level = 95) +
#   labs(y = "Tons",
#        title = "Blank")
```

Adjusted R2 = 0.1336

Plots of the model

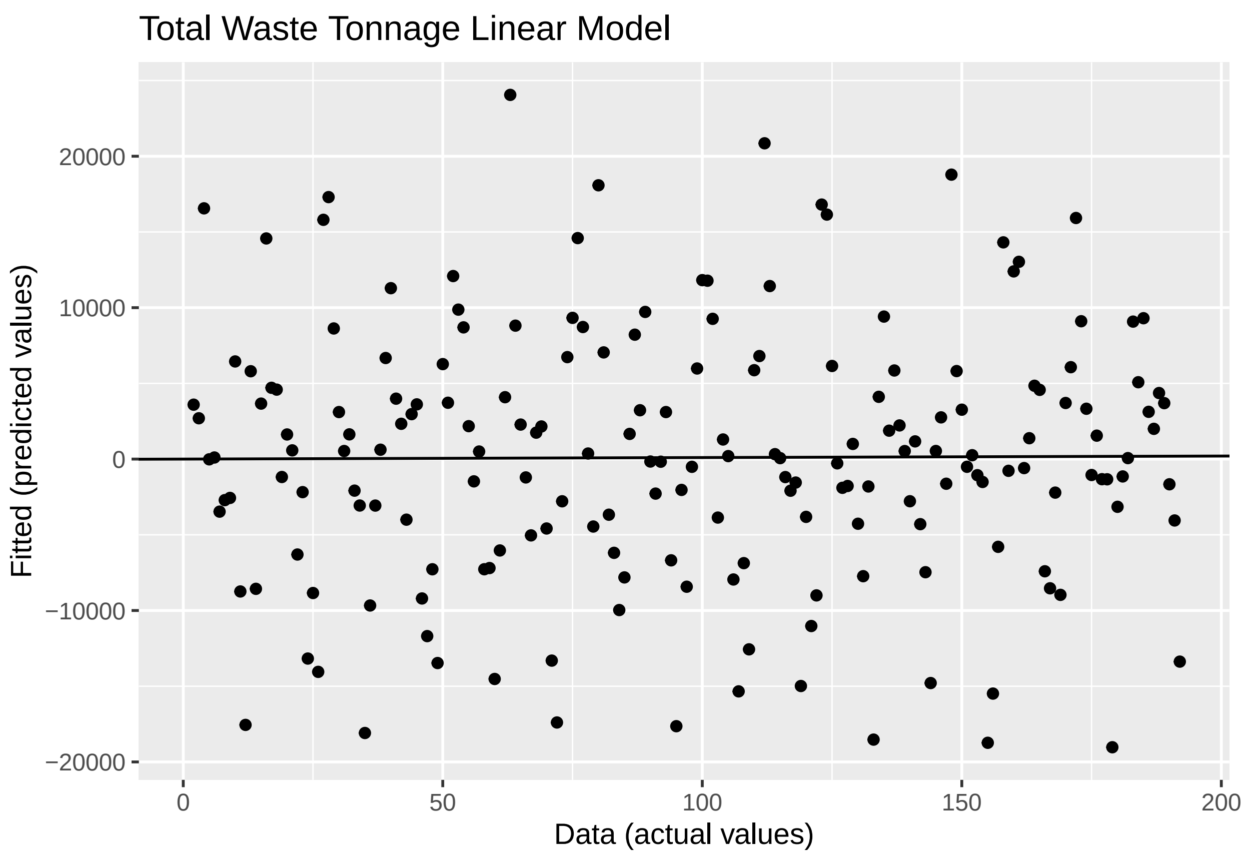
```
augment(lin_model_fit) %>%
  ggplot(aes(x = month_num)) +
  geom_line(aes(y = tw_diff1, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(x = "Month Number"
       ,y = "Differenced Total Waste Values",
       title = "Total Waste Tonnage",
       subtitle = "Jan '05 - Dec '20") +
  scale_colour_manual(values=c(Data="black",Fitted="#D55E00")) +
  guides(colour = guide_legend(title = NULL))
```

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




```
augment(lin_model_fit) %>%
  ggplot(aes(x = month_num, y = .fitted)) +
  geom_point() +
  labs(
    y = "Fitted (predicted values)",
    x = "Data (actual values)",
    title = "Total Waste Tonnage Linear Model"
  ) + geom_abline(intercept = 0, slope = 1)
```

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



Linear model attempt 2

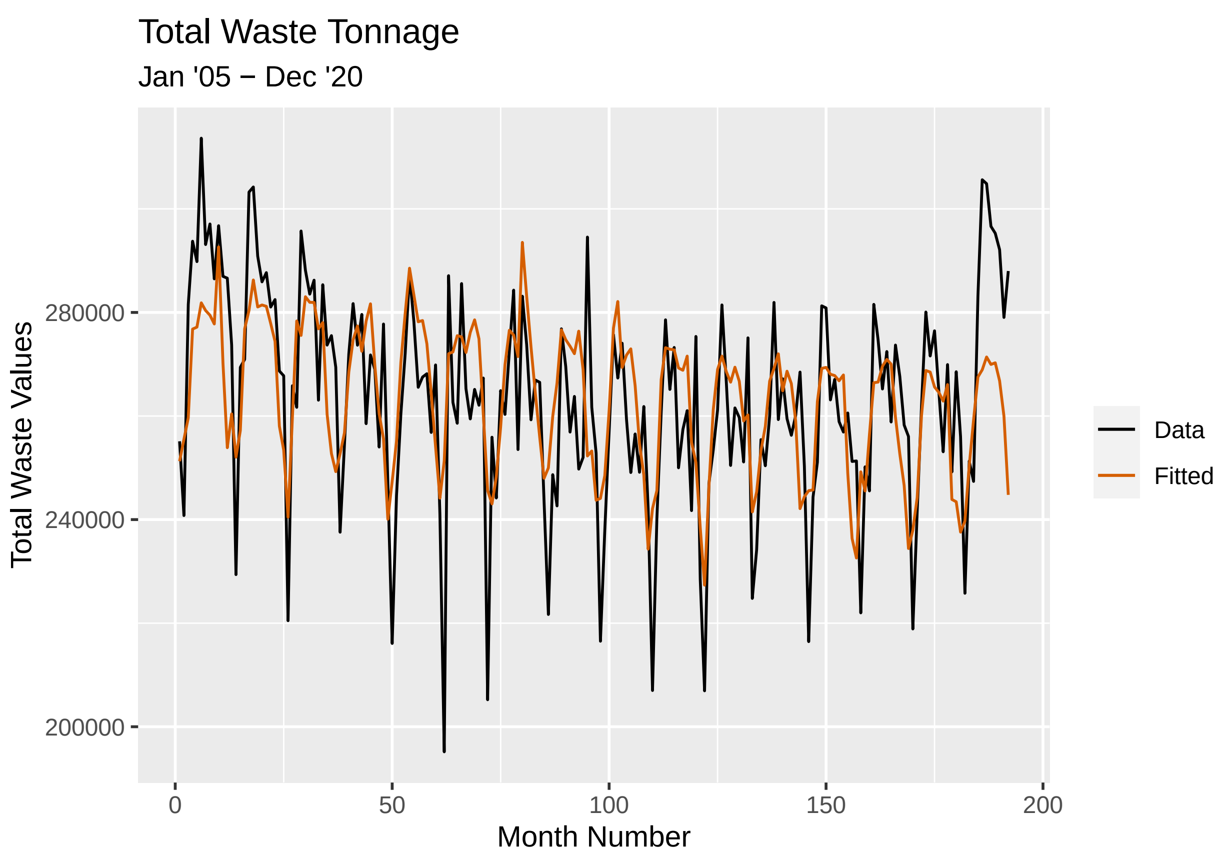
```
lin_model_fit2 <- nyc_ts_2 %>%
  model(
    linear = TSLM(total_waste_total ~ cpi + unemp_rate + avg_precip + avg_temp + avg_CDD)
    # exponential = TSLM(log(tw_diff1) ~  cpi + unemp_rate + avg_precip + temp_diff1 + cdd_diff1)
  )
report(lin_model_fit2)
```

```
## Series: total_waste_total
## Model: TSLM
##
## Residuals:
##       Min        1Q   Median        3Q       Max
## -56176.4 -10490.0   -752.2  10963.3  43220.2
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 257322.46   15707.89  16.382  < 2e-16 ***
## cpi            -84.31      19.71  -4.277 3.02e-05 ***
## unemp_rate   48960.81   39962.96   1.225 0.222066
## avg_precip   38695.86   14250.14   2.715 0.007242 **
## avg_temp      1144.89     139.80   8.189 4.10e-14 ***
```

```
## avg_CDD       -1639.11     453.63  -3.613 0.000389 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15750 on 186 degrees of freedom
## Multiple R-squared: 0.4314,  Adjusted R-squared: 0.4161
## F-statistic: 28.22 on 5 and 186 DF, p-value: < 2.22e-16
```

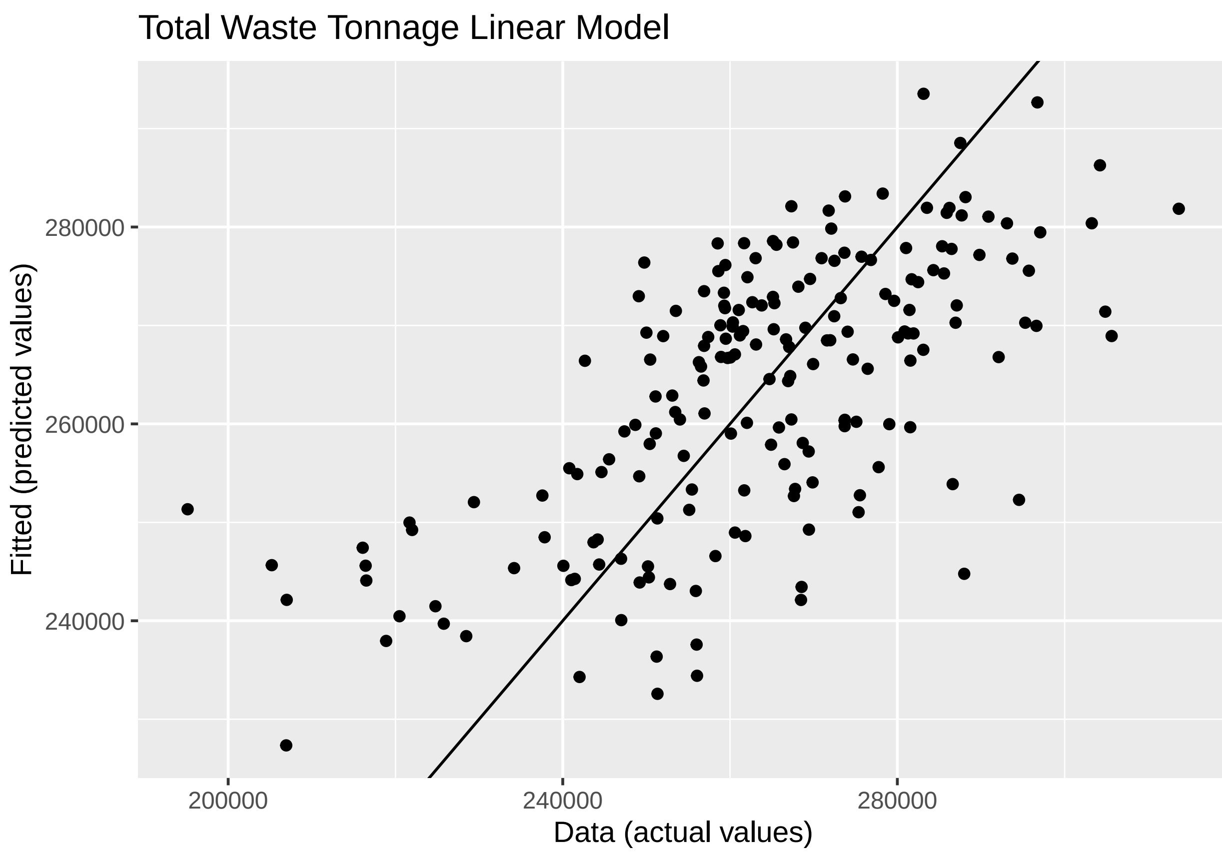
Adjusted R2 = 0.4161
Plots of the model

```
augment(lin_model_fit2) %>%
  ggplot(aes(x = month_num)) +
  geom_line(aes(y = total_waste_total, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(x = "Month Number"
       ,y = "Total Waste Values",
       title = "Total Waste Tonnage",
       subtitle = "Jan '05 - Dec '20") +
  scale_colour_manual(values=c(Data="black",Fitted="#D55E00")) +
  guides(colour = guide_legend(title = NULL))
```




```
augment(lin_model_fit2) %>%
  ggplot(aes(x = total_waste_total, y = .fitted)) +
  geom_point() +
```

```
  labs(
    y = "Fitted (predicted values)",
    x = "Data (actual values)",
    title = "Total Waste Tonnage Linear Model") +
  geom_abline(intercept = 0, slope = 1)
```



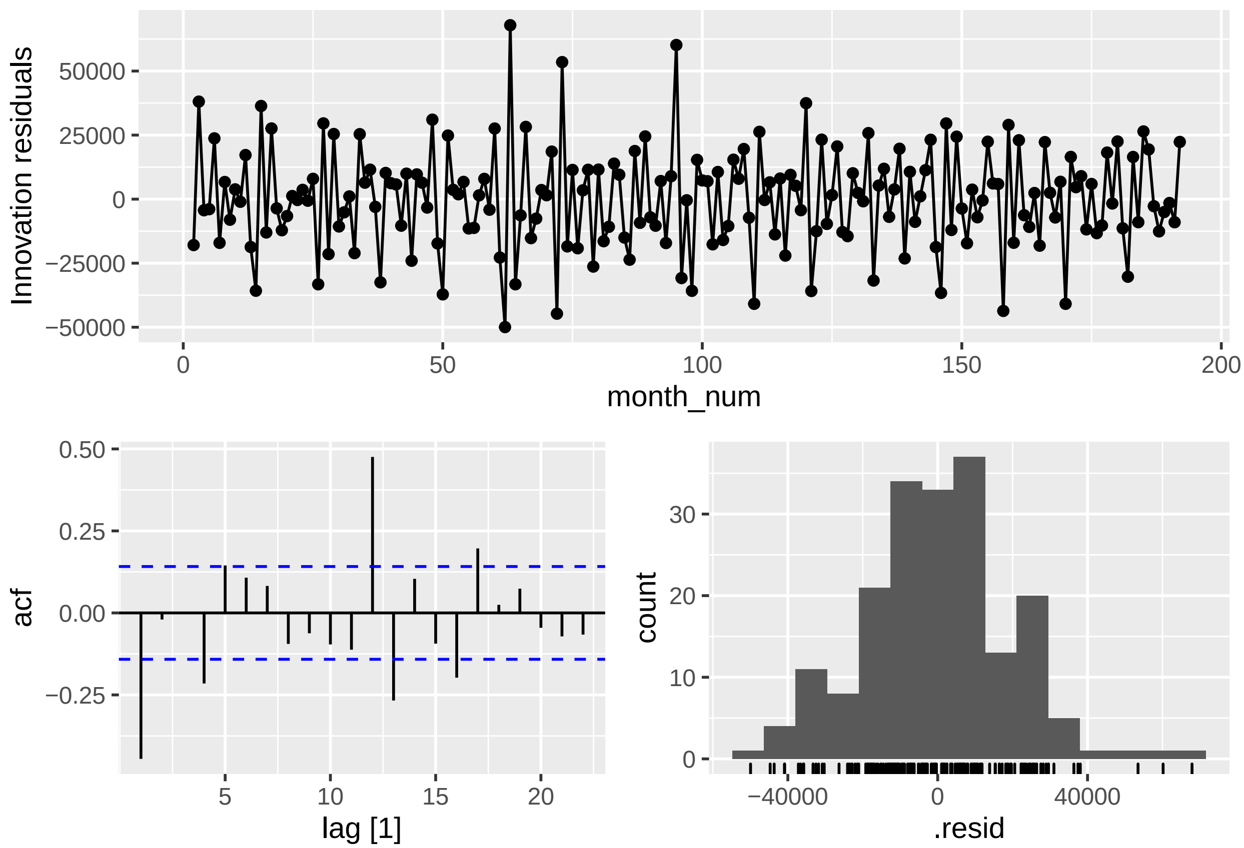
Evaluating the Regression Model

It is always a good idea to check whether the residuals are normally distributed. The authors explained earlier, that this is not essential for forecasting, but it does make the calculation of prediction intervals much easier.

Residuals of the differenced values

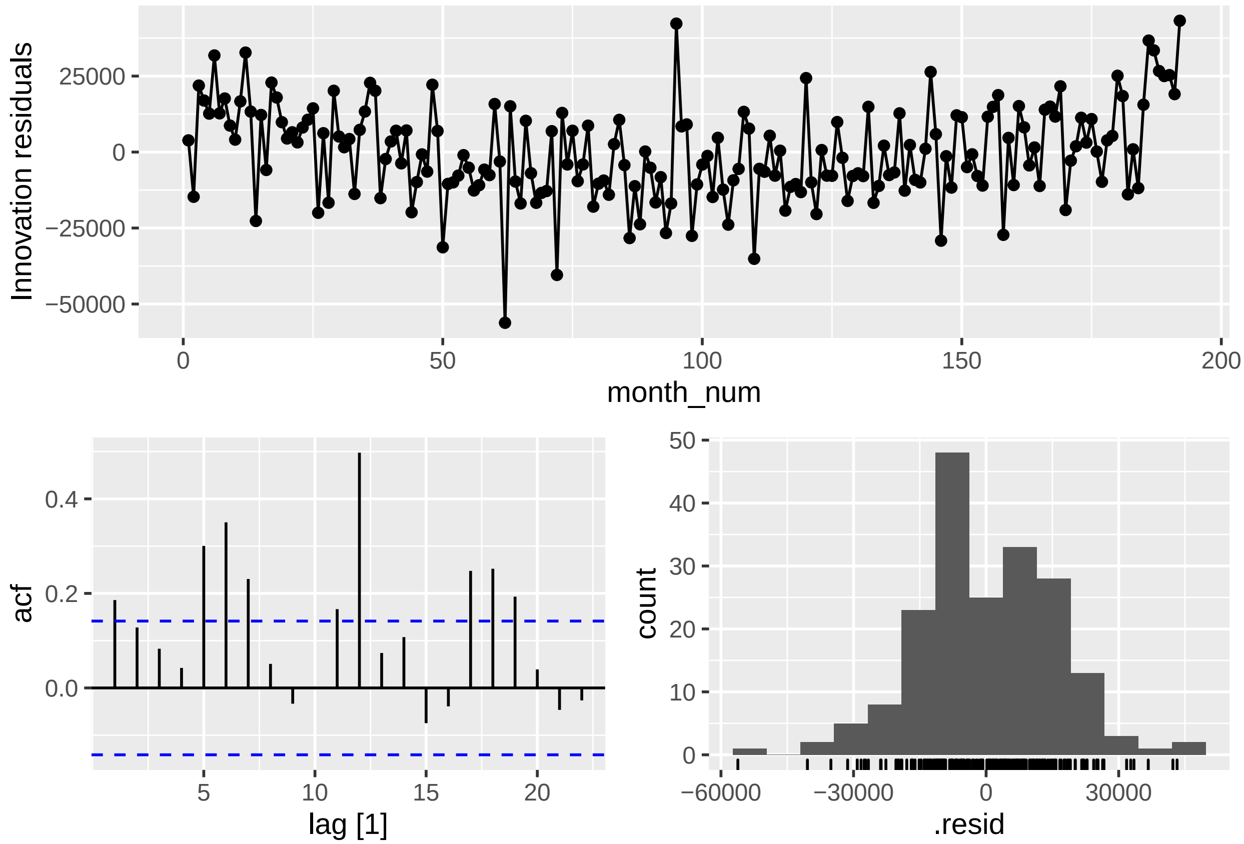
```
lin_model_fit %>%
  gg_tsresiduals()
```

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

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

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



Residuals of the non-differenced values

```
lin_model_fit2 %>%
  gg_tsresiduals()
```



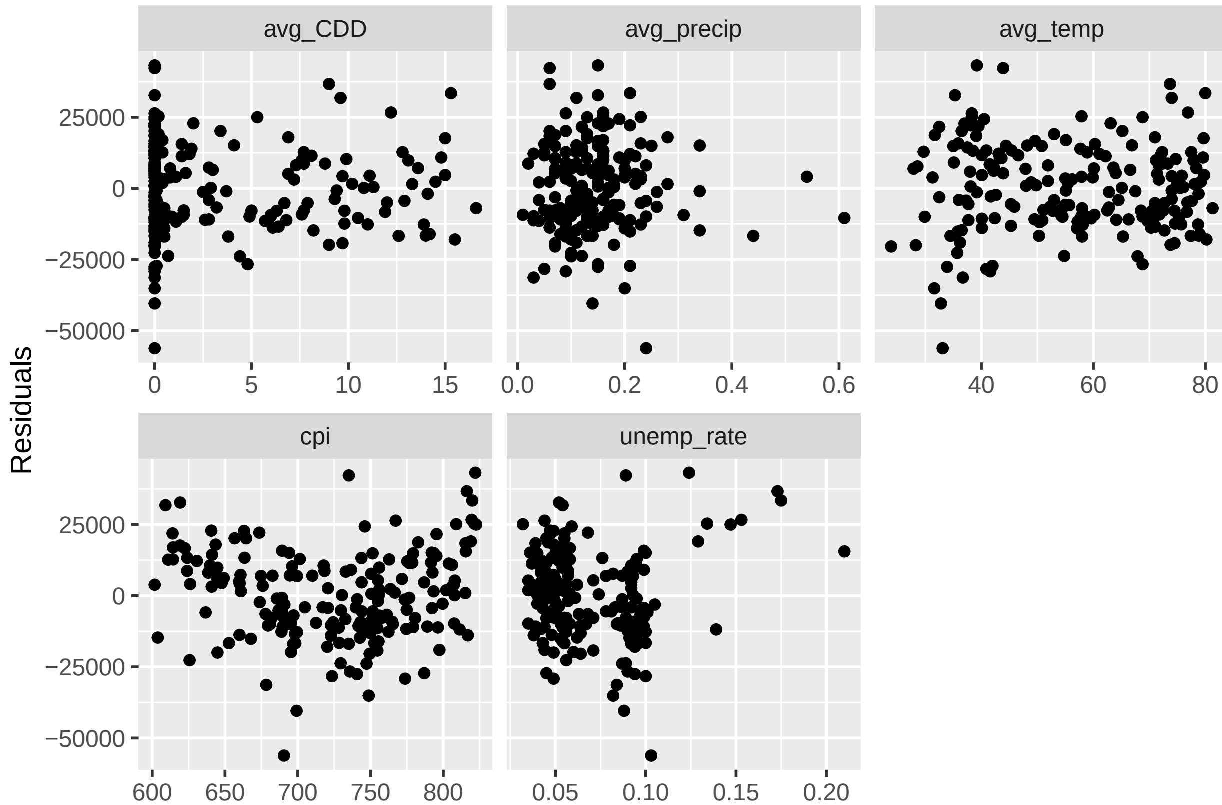
Lags of the second model

```
augment(lin_model_fit2) %>%
  features(.innov, ljung_box, lag = 10, dof = 5)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 linear    65.6  8.25e-13
```

Residual Plots against predictors

```
nyc_ts_2 %>%
  left_join(residuals(lin_model_fit2), by = "month_num") %>%
  pivot_longer(avg_precip:unemp_rate,
               names_to = "regressor", values_to = "x") %>%
  ggplot(aes(x = x, y = .resid)) +
  geom_point() +
  facet_wrap(. ~ regressor, scales = "free_x") +
  labs(y = "Residuals", x = "")
```



Linear Model attempt 2 with trend

```
lin_model_fit2_trend <- nyc_ts_2 %>%
  model(
    linear = TSLM(total_waste_total ~ cpi + unemp_rate + avg_precip + avg_temp + avg_CDD + trend())
    # exponential = TSLM(log(tw_diff1) ~  cpi + unemp_rate + avg_precip + temp_diff1 + cdd_diff1)
  )
report(lin_model_fit2)
```

```
## Series: total_waste_total
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -56176.4 -10490.0   -752.2  10963.3  43220.2
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 257322.46   15707.89  16.382  < 2e-16 ***
## cpi            -84.31      19.71  -4.277 3.02e-05 ***
## unemp_rate   48960.81   39962.96   1.225 0.222066
## avg_precip   38695.86   14250.14   2.715 0.007242 **
## avg_temp      1144.89     139.80   8.189 4.10e-14 ***
```

```
## avg_CDD      -1639.11     453.63  -3.613 0.000389 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15750 on 186 degrees of freedom
## Multiple R-squared: 0.4314,  Adjusted R-squared: 0.4161
## F-statistic: 28.22 on 5 and 186 DF, p-value: < 2.22e-16
```

```
print("------------")
```

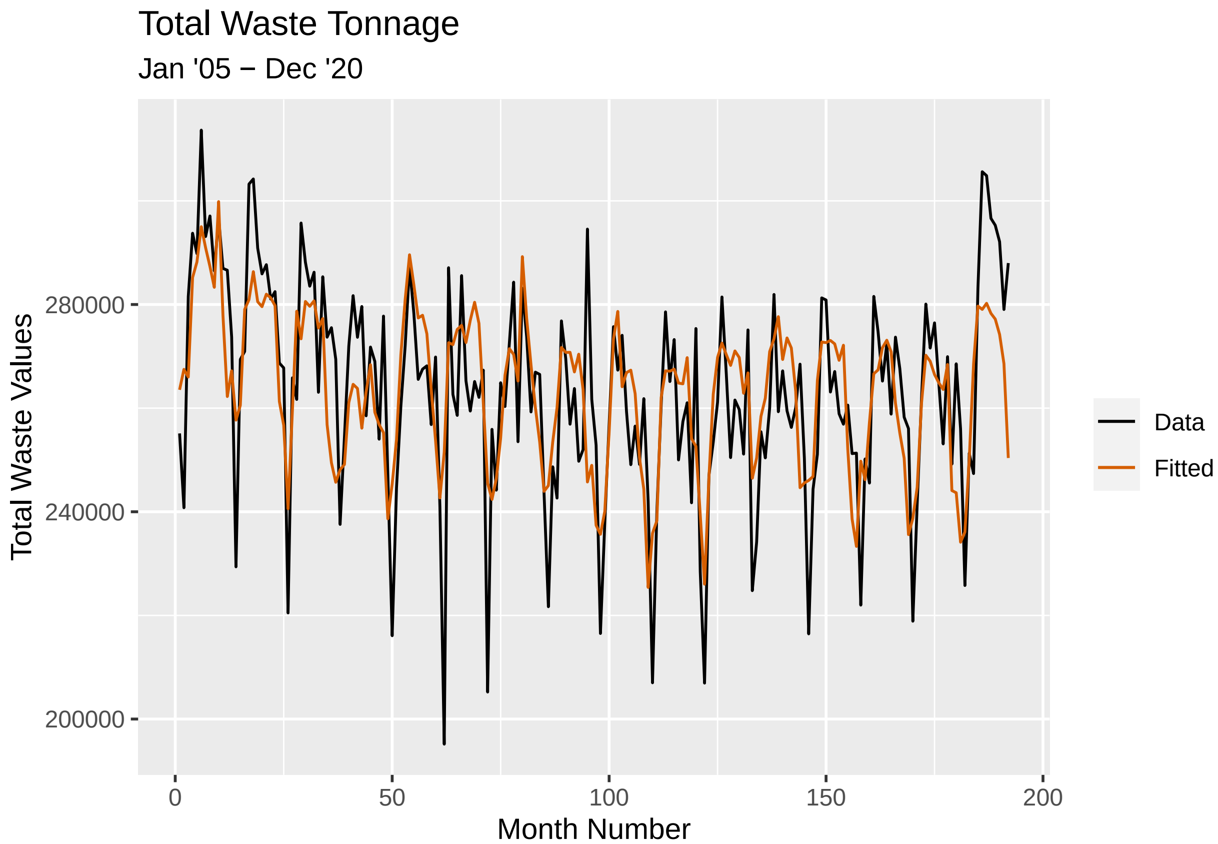
```
## [1] "------------"
```

```
report(lin_model_fit2_trend)
```

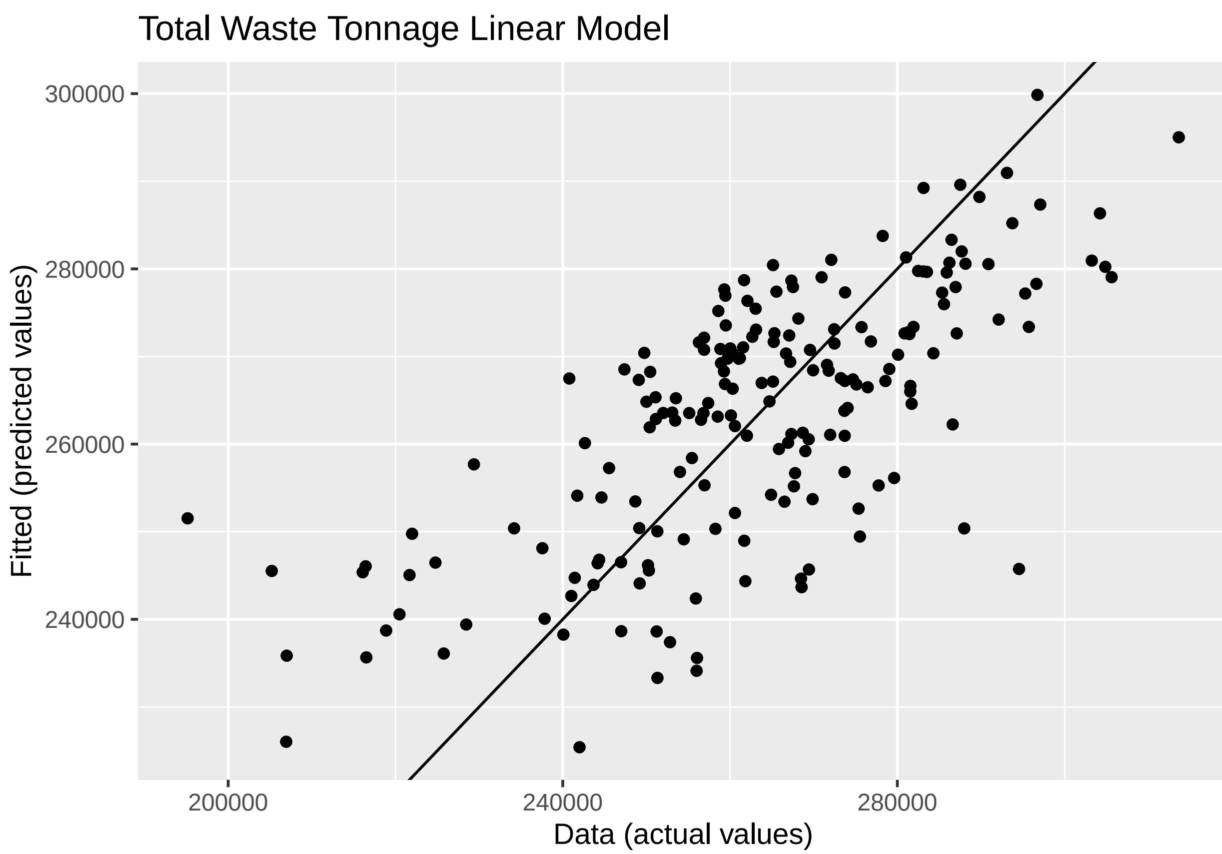
```
## Series: total_waste_total
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -56372.3 -10311.4   -208.4   9820.6  48803.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 641770.0    82288.4   7.799 4.41e-13 ***
## cpi           -717.7      134.6  -5.331 2.83e-07 ***
## unemp_rate  113510.6    40196.8   2.824  0.00526 **
## avg_precip   43763.5    13531.7   3.234  0.00145 **
## avg_temp      1256.5      134.4   9.348  < 2e-16 ***
## avg_CDD      -1829.1      431.3  -4.241 3.51e-05 ***
## trend()        670.1      141.1   4.750 4.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14910 on 185 degrees of freedom
## Multiple R-squared: 0.4932,  Adjusted R-squared: 0.4767
## F-statistic:    30 on 6 and 185 DF, p-value: < 2.22e-16
```

There is an average upward trend of 670

```
augment(lin_model_fit2_trend) %>%
  ggplot(aes(x = month_num)) +
  geom_line(aes(y = total_waste_total, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(x = "Month Number"
       ,y = "Total Waste Values",
       title = "Total Waste Tonnage",
       subtitle = "Jan '05 - Dec '20") +
  scale_colour_manual(values=c(Data="black",Fitted="#D55E00")) +
  guides(colour = guide_legend(title = NULL))
```




```r
augment(lin_model_fit2_trend) %>%
  ggplot(aes(x = total_waste_total, y = .fitted)) +
  geom_point() +
  labs(
    y = "Fitted (predicted values)",
    x = "Data (actual values)",
    title = "Total Waste Tonnage Linear Model") +
  geom_abline(intercept = 0, slope = 1)
```



Selecting Predictors

```
glance(lin_model_fit2) %>%
  select(adj_r_squared, CV, AIC, AICc, BIC)
```

```
## # A tibble: 1 x 5
##   adj_r_squared         CV   AIC  AICc   BIC
##           <dbl>      <dbl> <dbl> <dbl> <dbl>
## 1         0.416 259453200. 3719. 3720. 3742.
```

```
glance(lin_model_fit2_trend) %>%
  select(adj_r_squared, CV, AIC, AICc, BIC)
```

```
## # A tibble: 1 x 5
##   adj_r_squared         CV   AIC  AICc   BIC
##           <dbl>      <dbl> <dbl> <dbl> <dbl>
## 1         0.477 232255569. 3699. 3700. 3725.
```

We compare these values against the corresponding values from other models. For the CV, AIC, AICc and BIC measures, we want to find the model with the lowest value; for Adjusted R^2, we seek the model with the highest value. The adjusted R^2 it is not a good measure of the predictive ability of a model. It measures how well the model fits the historical data, but not how well the model will forecast future data.

In addition, R^2 does not allow for "degrees of freedom". Adding any variable tends to increase the value of R^2 even if that variable is irrelevant. For these reasons, forecasters should not use R^2 to determine whether a model will give good predictions, as it will lead to over-fitting.

In this case, the trended model beats out the non-trend total waste values model

Consequently, we recommend that one of the AICc, AIC, or CV statistics be used, each of which has forecasting as their objective. If the value of T is large enough, they will all lead to the same model. In most of the examples in this book, we use the AICc value to select the forecasting model.

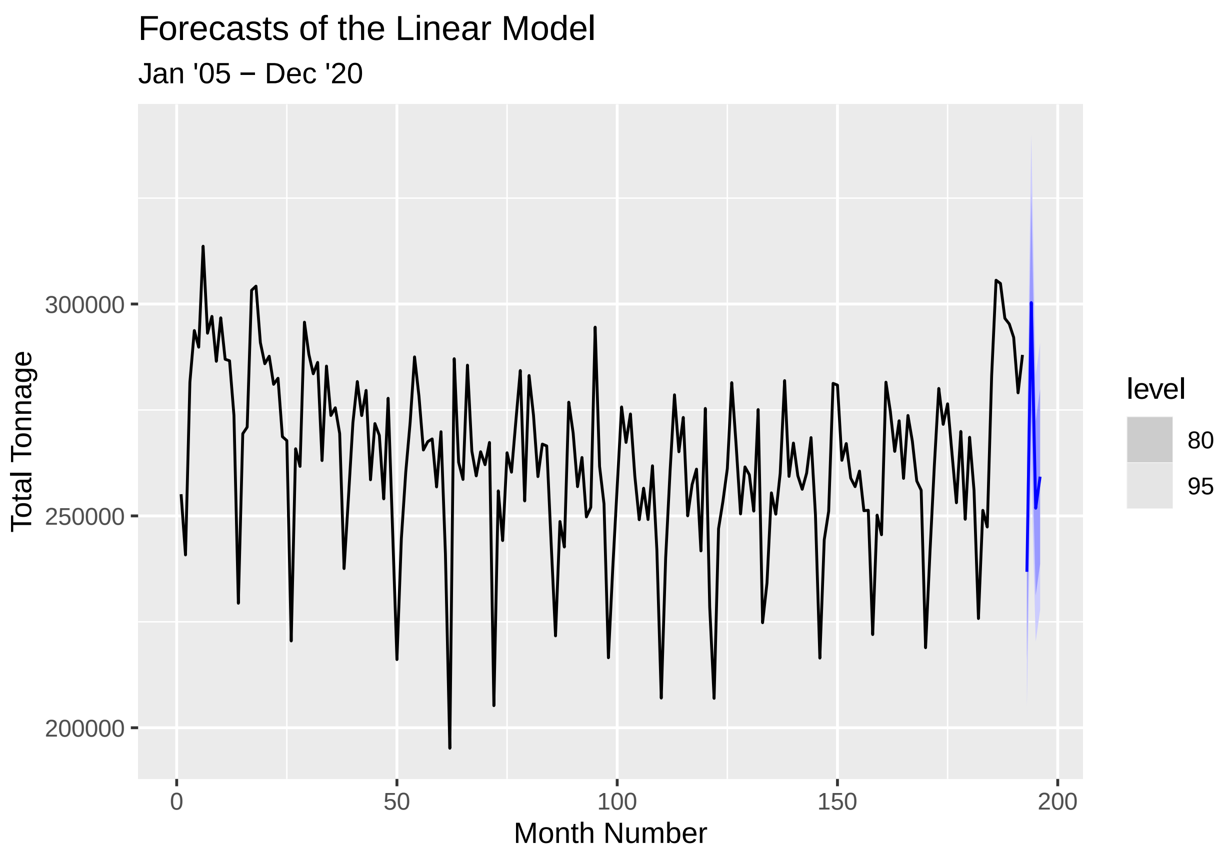
Forecast of the total-waste model and trend model

Ex-post forecasts are those that are made using later information on the predictors. For example, ex-post forecasts of consumption may use the actual observations of the predictors, once these have been observed

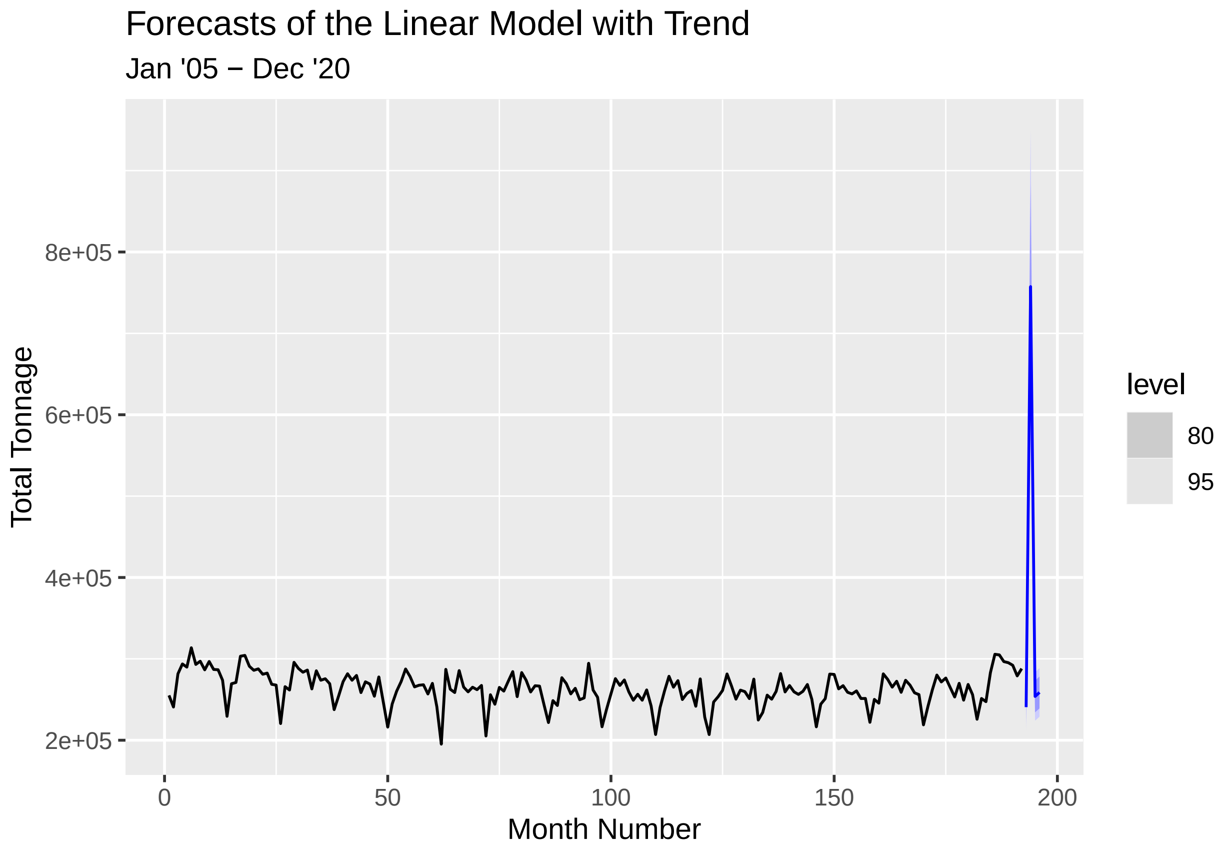
We should note that prediction intervals for scenario based forecasts do not include the uncertainty associated with the future values of the predictor variables. They assume that the values of the predictors are known in advance.

```
nyc_data_small_future <- new_data(nyc_ts_2, 4) %>%
  mutate(cpi = c(825.413,111, 831.067, 836.885),
         unemp_rate = c(0.1333,0.1280, 0.1130, 0.1090),
         avg_precip = c(0.07, 0.18, 0.17, 0.12),
         avg_temp = c(34.8, 34.2, 45.8, 54.6),
         avg_CDD = 0, 0, 0.1, 0.2)

nyc_data_small_future_diff <- new_data(nyc_ts_2,4) %>%
  mutate(cpi = c(825.413,111, 831.067, 836.885),
         unemp_rate = c(0.1333,0.1280, 0.1130, 0.1090),
         avg_precip = c(0.07, 0.18, 0.17, 0.12),
         temp_diff1 = c(-4.4, -0.6, 11.6, 8.8),
         cdd_diff1 = c(0,0,0.1, 0.1))
```

```
forecast(lin_model_fit2, new_data = nyc_data_small_future) %>%
  autoplot(nyc_ts_1) +
  labs(x  = "Month Number",
       y = "Total Tonnage",
       title = "Forecasts of the Linear Model",
       subtitle = "Jan '05 - Dec '20")
```




```
forecast(lin_model_fit2_trend, new_data = nyc_data_small_future) %>%
  autoplot(nyc_ts_1) +
  labs(x  = "Month Number",
       y = "Total Tonnage",
       title = "Forecasts of the Linear Model with Trend",
       subtitle = "Jan '05 - Dec '20")
```


Forecasts of the Linear Model with Trend
Jan '05 − Dec '20
Total Tonnage
8e+05
6e+05
4e+05
2e+05
0
50
100
150
200
Month Number
level
80
95