Unemployment in Chicago

Melanie McCord

Contents

Introduction

Motivation

The purpose of this project was to explore unemployment in Chicago and to understand how it changed in relation to the pandemic. The unemployment rate tends to follow a cyclic trend, tending to peak with recessions and fall during economic growth. Labonte (n.d.) However, the pandemic created major economic changes, including a large peak in the unemployment rate. Ma et al. (2020) In order to understand these trends and how drastic they were, for my project, I chose to explore the unemployment rate in Chicago.

Dataset

This dataset comes from the non-seasonally adjusted Chicago unemployment metrics from the US bureau of statistics. The period of the data is monthly. I chose to focus on Chicago specifically because the city of Chicago publishes a great deal of metrics and future work aims to extend the analysis of this data into other variables. The time period of the dataset goes from January 1994 to January 2023.

Data Exploration

```
unemployment_chicago <- readxl::read_excel("data/Chicago-Naperville-ArlingtonHeights_MD_notseasadj.xls"
  drop_na() %>%
  rename(Month = `Month/Year`) %>%
```

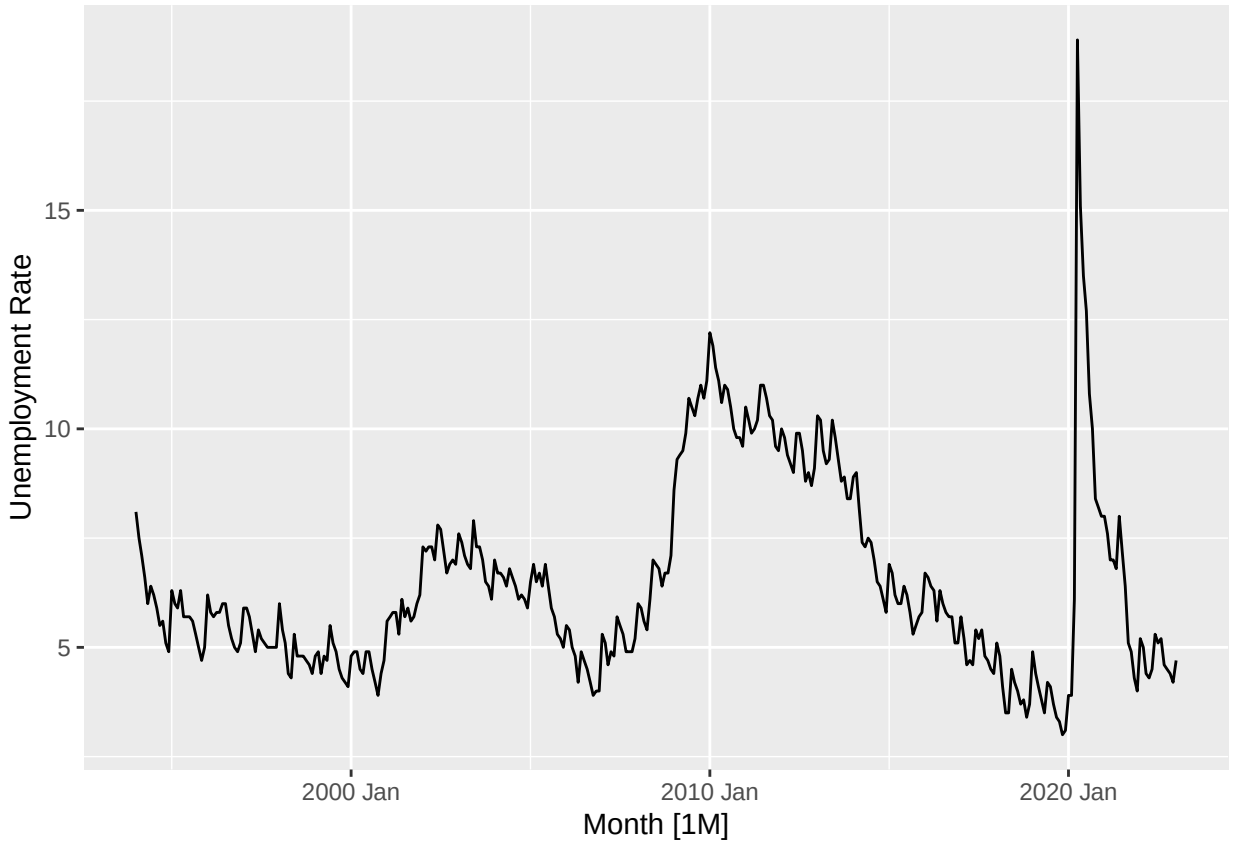
```
  mutate(`Month` = yearmonth(Month)) %>%
  as_tsibble(index = Month)
head(unemployment_chicago)
```

```
## # A tsibble: 6 x 8 [1M]
##      Month `Labor Force` Labor Force Participa~1 Employed EmploymentParticipat~2
##      <mth>         <dbl>                   <dbl>    <dbl>                   <dbl>
## 1 1994 Jan       3489100                    67.9  3207300                    62.4
## 2 1994 Feb       3504900                    68.2  3241900                    63
## 3 1994 Mar       3483800                    67.7  3237500                    62.9
## 4 1994 Apr       3473900                    67.5  3246100                    63.1
## 5 1994 May       3510600                    68.2  3300400                    64.1
## 6 1994 Jun       3566400                    69.2  3338500                    64.8
## # i abbreviated names: 1: `Labor Force Participation Rate`,
## #   2: `EmploymentParticipation Rate`
## # i 3 more variables: Unemployed <dbl>, `Unemployment Rate` <dbl>,
## #   `IL Rate` <dbl>
```

The data is seasonal, in this case monthly. The data consists of 7 other variables, labor force, labor force participation rate, employed, employment participation rate, unemployed, and unemployment rate. For the purposes of this project, I will simply focus on the unemployment rate.

Visualizing the Data

```
unemployment_chicago %>%
  autoplot(`Unemployment Rate`)
```

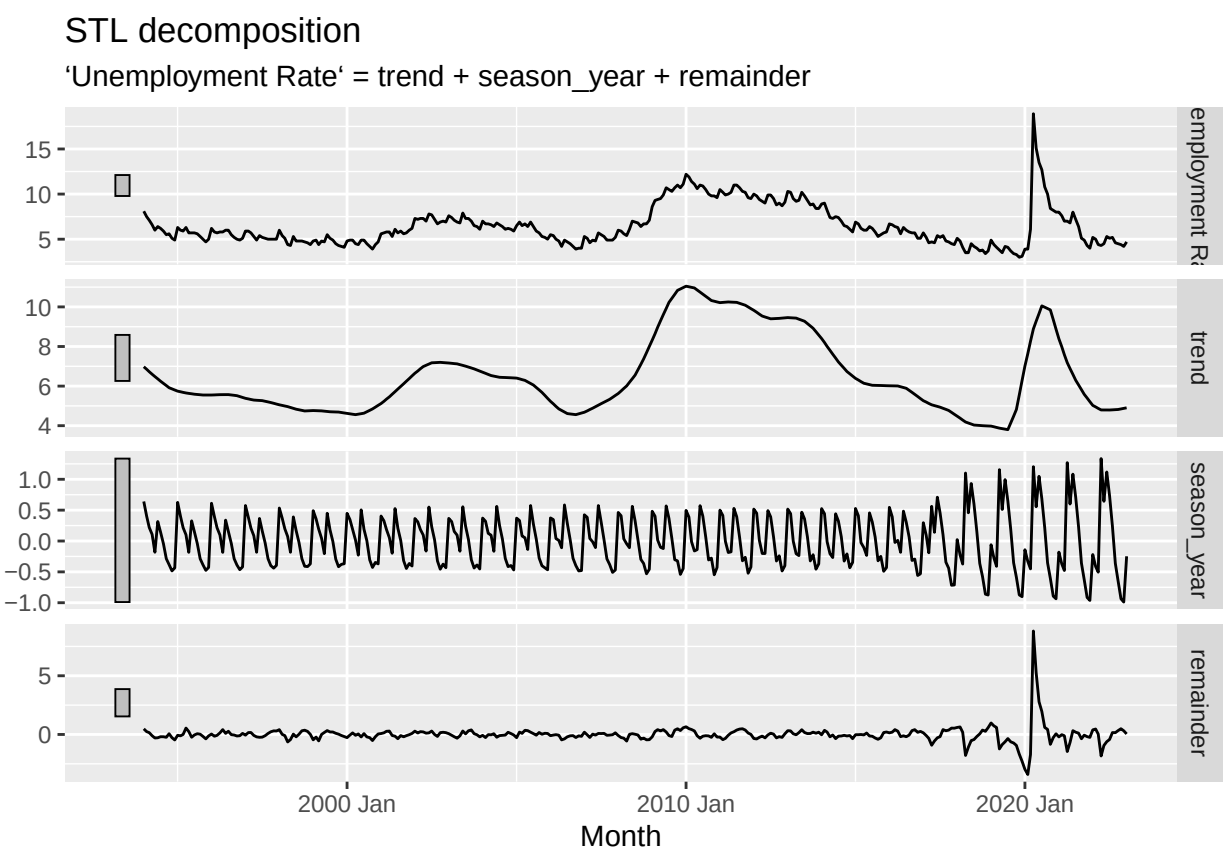


The data shows clear seasonality, as well as cyclicity in drops and falls of the unemployment rate, i.e. recessions vs economic growth. There doesn't appear to be a trend in the data, as the sudden growth followed by sharp dips (i.e. economic growth vs. recessions) seems evident in the cyclicity.

Decomposing the Seasonal and Trend Components

STL Decomposition

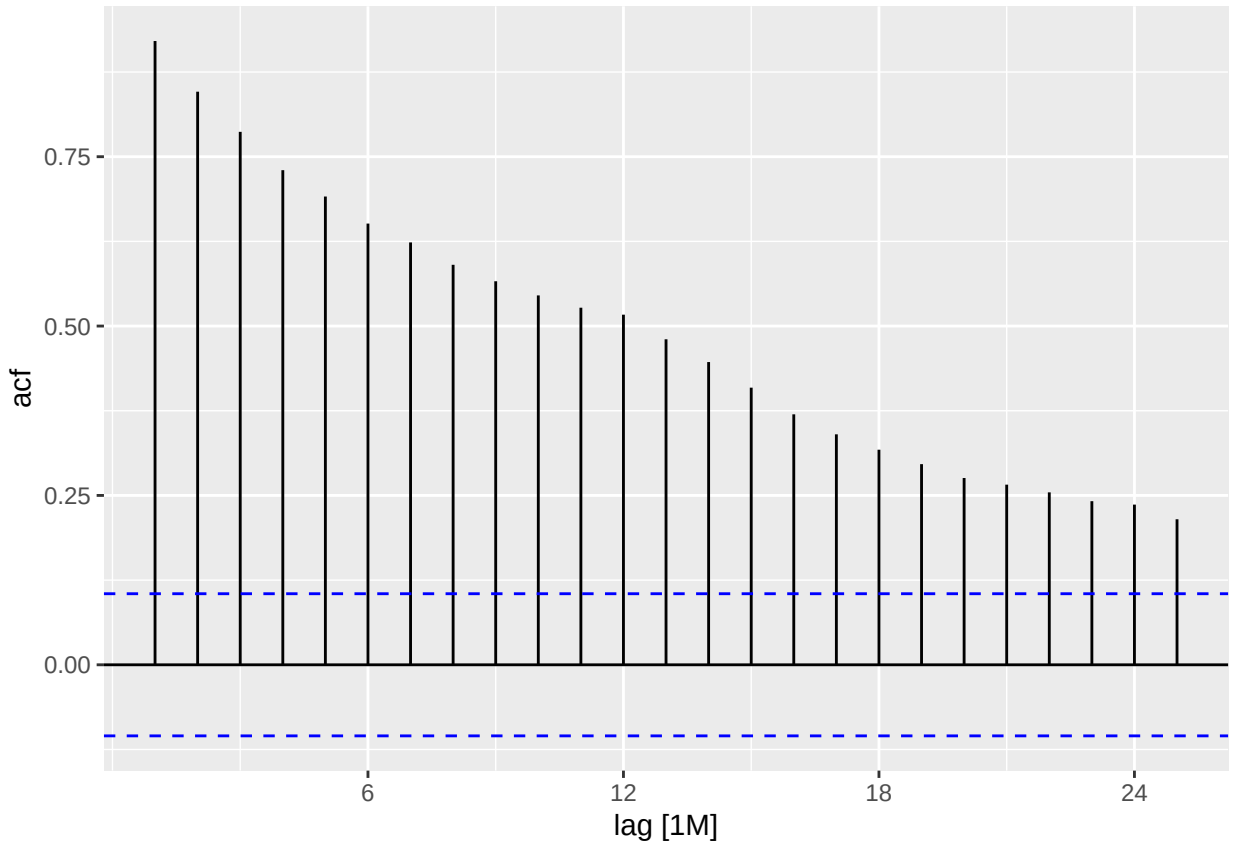
```
unemployment_chicago %>%
  model (
    STL(`Unemployment Rate`)
  ) %>%
  components() %>%
  autoplot()
```



Although there is definitely a seasonal component, it appears to be less significant than the trend. Mostly the remainder from the STL decomposition looks very constant, but there is a very sharp peak in the remainder in 2020. The seasonality appears to follow a similar pattern, however the 5 more recent years show a growth in the variance of the seasonality.

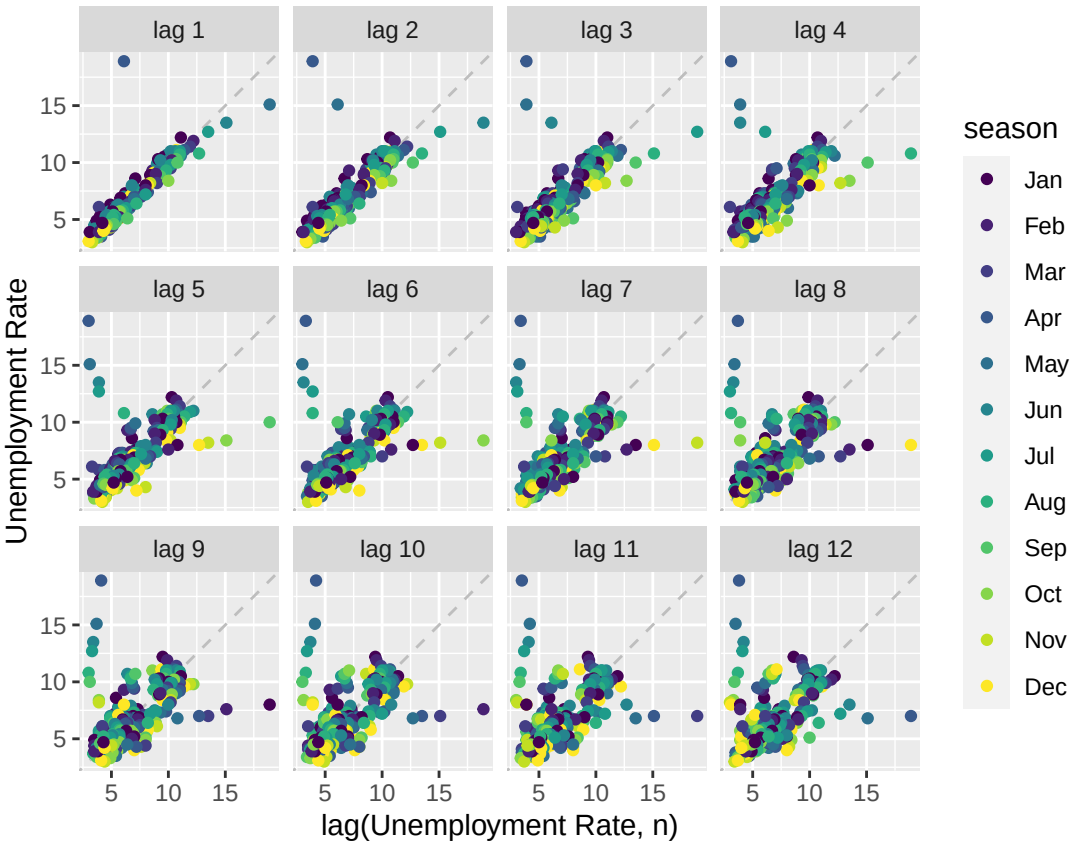
Exploring the Autocorrelation

```
unemployment_chicago %>%
  ACF(`Unemployment Rate`) %>%
  autoplot()
```



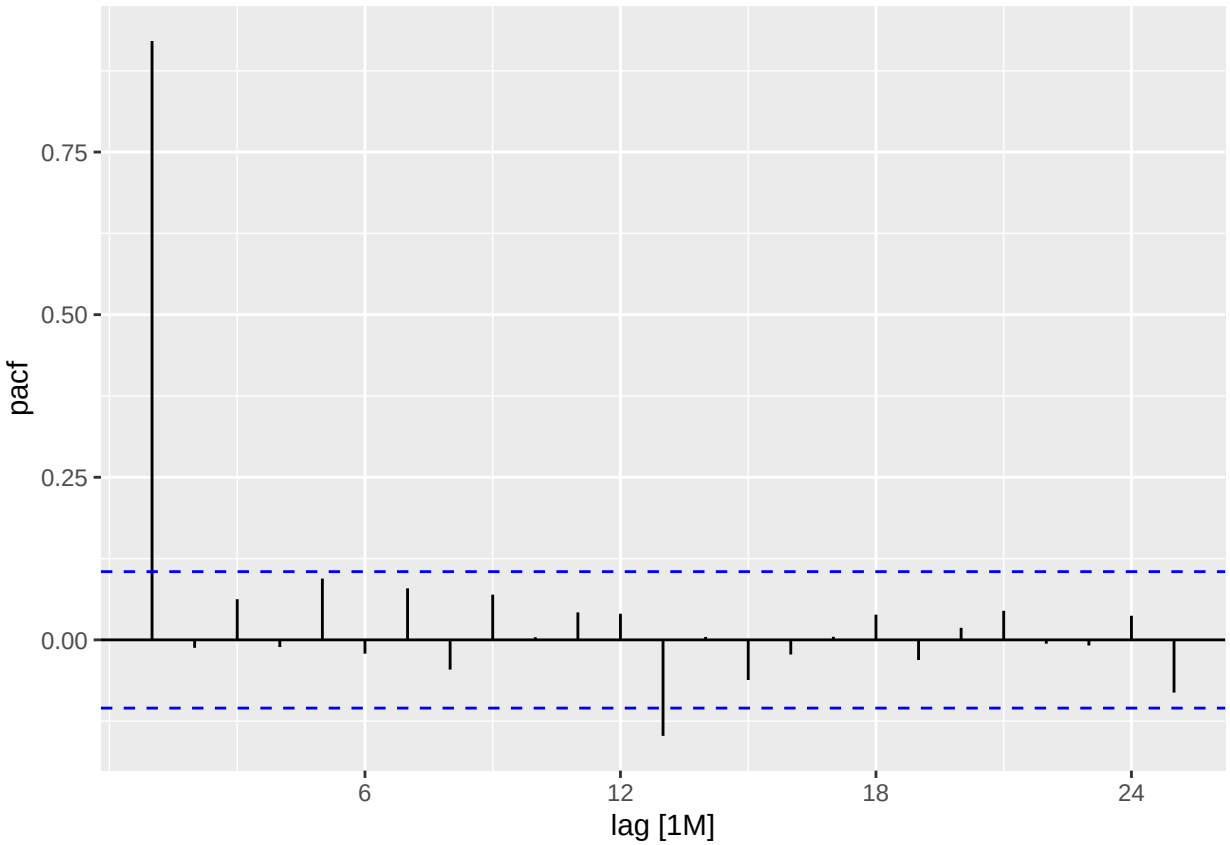
There is very significant autocorrelation throughout the entire cycles, that is steadily decreasing with each season. The autocorrelation is strictly positive.

```
unemployment_chicago %>%
  gg_lag(`Unemployment Rate`, lags = 1:12, geom = "point")
```



The lag plots also show strong positive autocorrelation, that is steadily decreasing with each lag, though this may be due to the outliers among each lag.

```
unemployment_chicago %>%
  PACF(`Unemployment Rate`) %>%
  autoplot()
```



The partial autocorrelation indicates that the first lag has a very strong partial autocorrelation, and the 13th lag also appears to have a significant partial autocorrelation, although it is less significant than the first one.

Identifying Appropriate Models

For the purposes of this project, I've chosen to compare variations of two common models, ARIMA and ETS, and compare their performance on this dataset.

First, I will consider whether to apply a box cox transformation. The box cox transformation can help with unstable variance by applying a power transformation to the data. For this purpose, Guerrero's method for defining the optimal value of lambda was used. Guerrero (1993)

Box Cox Transformation

```
unemployment_chicago %>%
  features(`Unemployment Rate`, guerrero)
```

```
## # A tibble: 1 x 1
##   lambda_guerrero
##             <dbl>
## 1          -0.900
```

Since the suggested lambda is close to -1, I will apply an inverse transformation on the unemployment data, since box_cox(X, -1) = inv(X, -1). I will compare both the models with the inverse transformation and the models without.

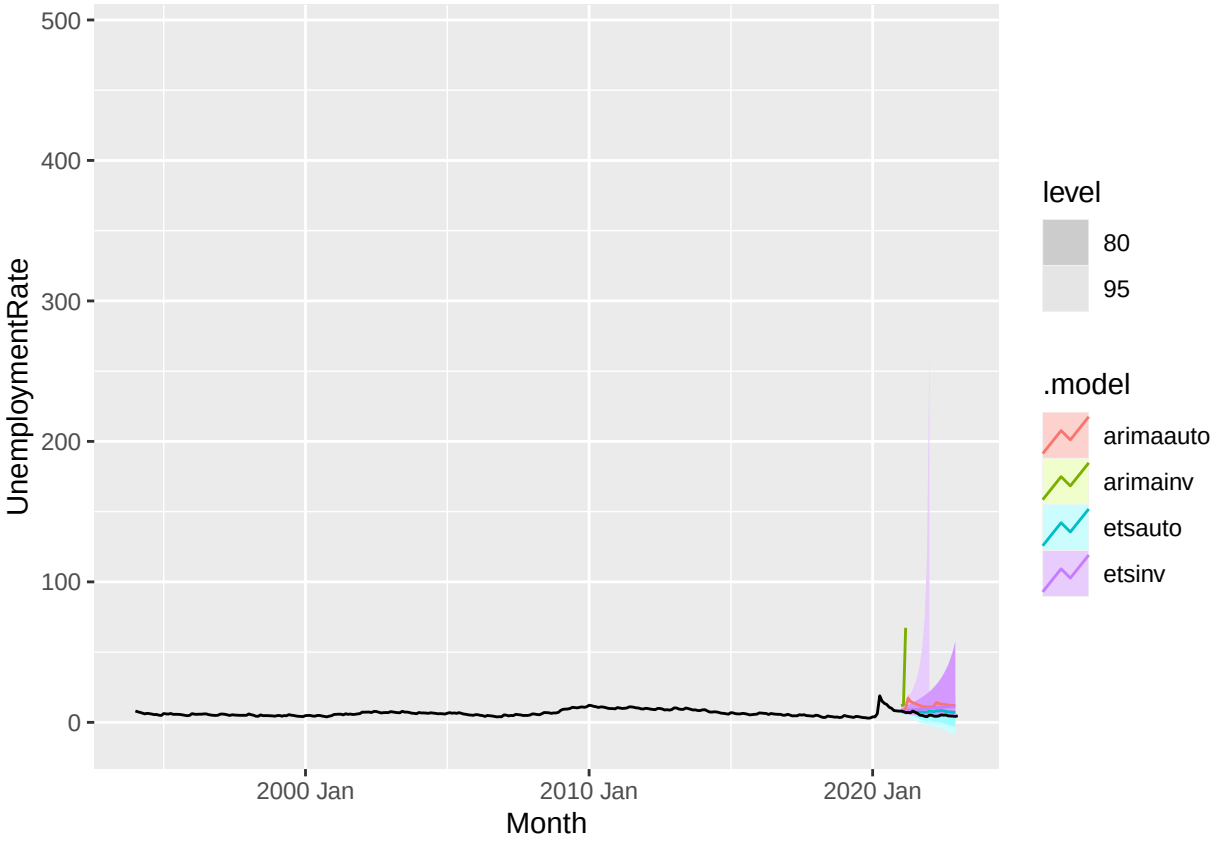
Fitting an ARIMA and ETS Model

```
unemployment_chicago <- unemployment_chicago %>%
  mutate(UnemploymentRate = `Unemployment Rate`)
unemployment_chicago_train <- unemployment_chicago %>%
  filter(year(Month) < 2021)
unemployment.fit <- unemployment_chicago_train %>%
  mutate(UnemploymentRate = `Unemployment Rate`) %>%
  model (
    etsinv = ETS(box_cox(UnemploymentRate, lambda = -1)),
    arimainv = ARIMA(box_cox(UnemploymentRate, lambda = -1)),
    arimaauto = ARIMA(UnemploymentRate),
    etsauto = ETS(UnemploymentRate)
  )
```

Forecasting

```
unemployment.fit %>%
  forecast(h = "2 years") %>%
  autoplot(unemployment_chicago)
```

```
## Warning: Removed 21 rows containing missing values (`()`).
```



Both inverse methods, as well as the ETS method, resulted in nonsensical prediction intervals. Unemployment rate is constrained to $(0, 1)$. Multiplying this by 100, as the US bureau of labor statistics appears to have done, results in prediction intervals that include impossibly high values for both inverse models, and negative values for the normal ETS model. Therefore, the model whose prediction intervals appear to make the most

sense in context is the ARIMA forecast.

```
unemployment.fit %>%
  forecast(h = "2 years") %>%
  accuracy(unemployment_chicago)
```

```
## # A tibble: 4 x 10
##   .model    .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
##   <chr>     <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 arimaauto Test   -6.59  6.99  6.59 -130.  130.   5.95  3.62  0.525
## 2 arimainv  Test  -23.0  35.0  23.0  -324.  324.  20.7  18.1  -0.159
## 3 etsauto   Test   -2.34  2.53  2.34  -47.9  47.9  2.11  1.31  0.835
## 4 etsinv    Test   -4.34  4.88  4.34  -90.6  90.6  3.92  2.53  0.843
```

However, the best performing model on the future data is the ETS model, with a RMSE of around 2.53 percentage points. However, it may not perform as well on past data.

```
augment(unemployment.fit) %>%
  features(.innov, ljung_box)
```

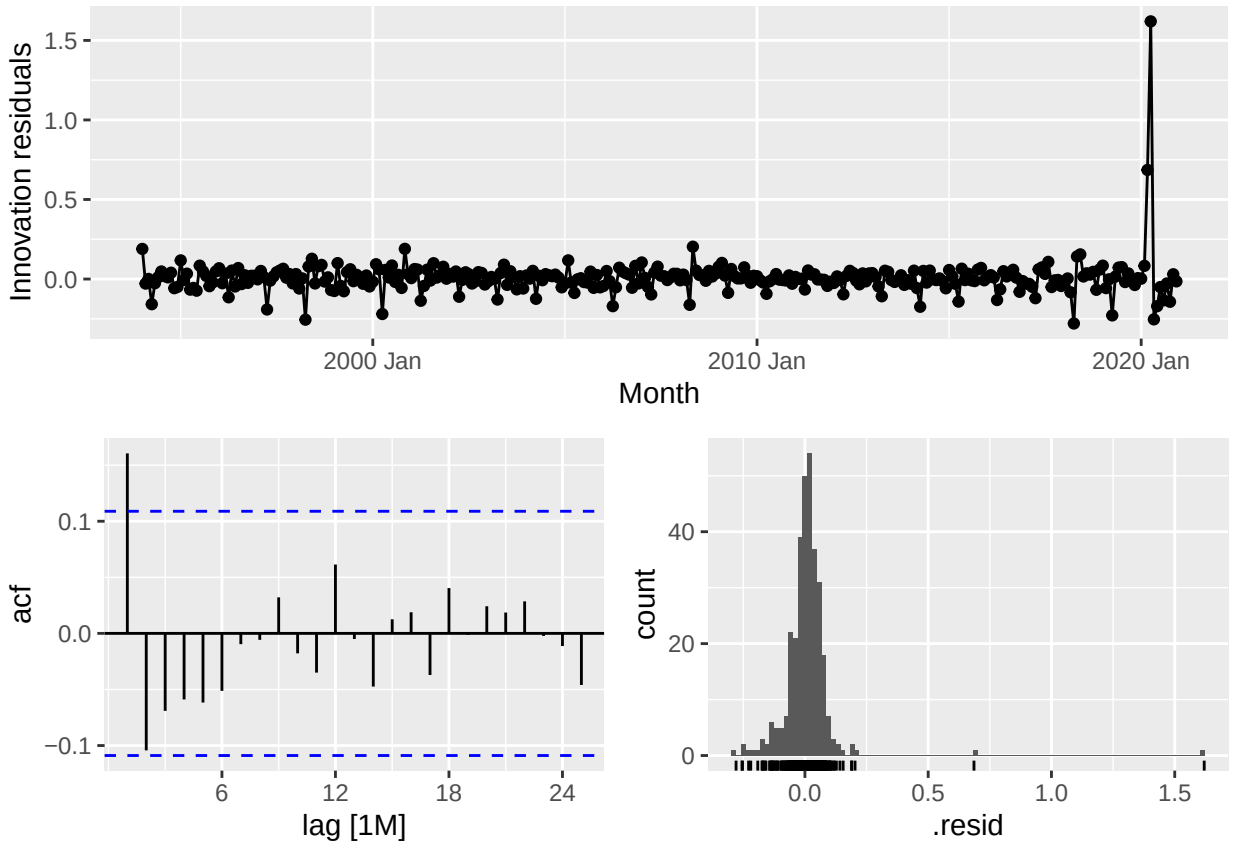
```
## # A tibble: 4 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 arimaauto  0.0938   0.759
## 2 arimainv   0.0182   0.893
## 3 etsauto    8.42     0.00371
## 4 etsinv     2.08     0.149
```

Although the ETS model performs the best on the future data, the ETS model appears to have very significant autocorrelation in the residuals of the past data, with a p-value of > 0.01. Further, the prediction intervals for the auto ETS data are very wide, stretching to negative numbers, which does not make sense to interpret in context since unemployment rate should be between 0 and 100%.

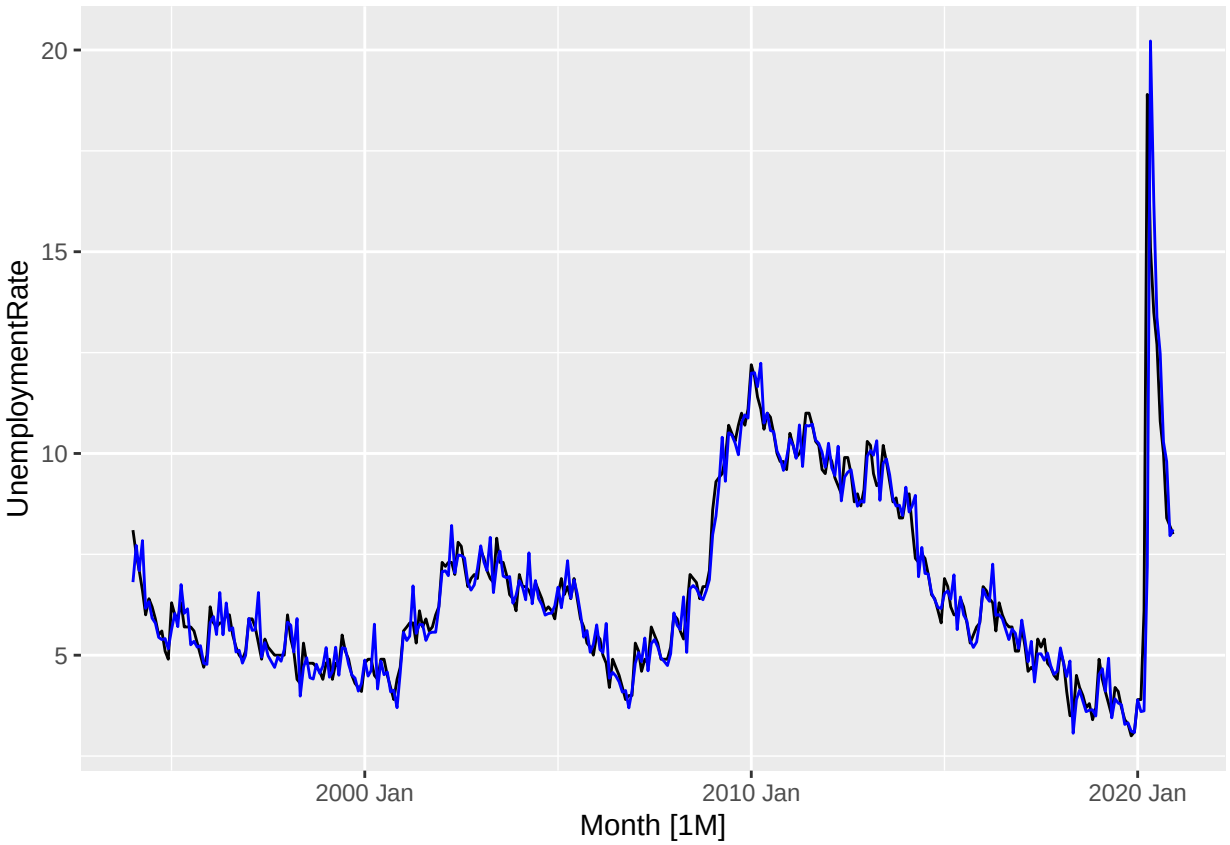
The models with the least significant autocorrelation are the two ARIMA models, which may indicate that the ARIMA model is a better fit for this dataset.

Exploring the Residuals

```
unemployment.fit %>%
  select(etsauto) %>%
  gg_tsresiduals()
```

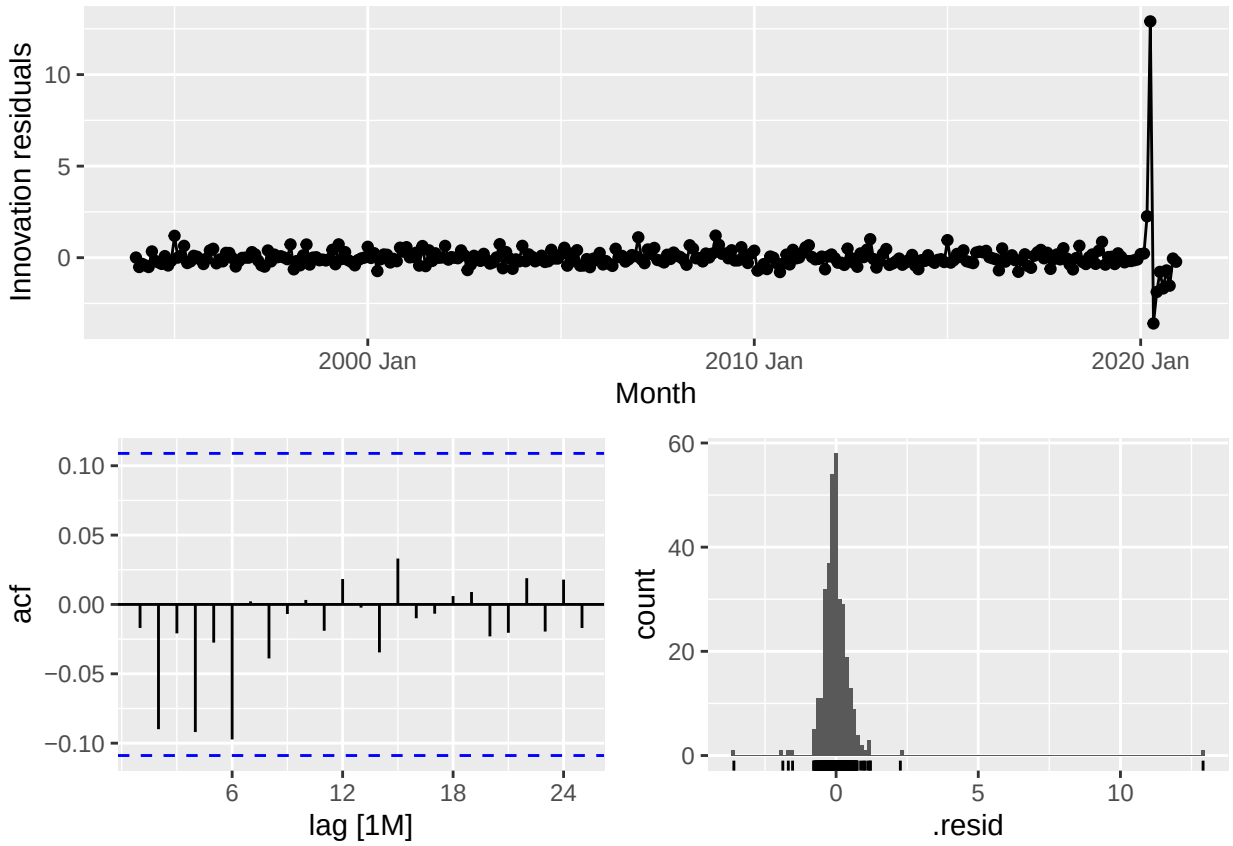



```
unemployment.fit %>%
  select(etsauto) %>%
  augment() %>%
  autoplot(UnemploymentRate) + geom_line(aes(y = .fitted), color = "blue")
```



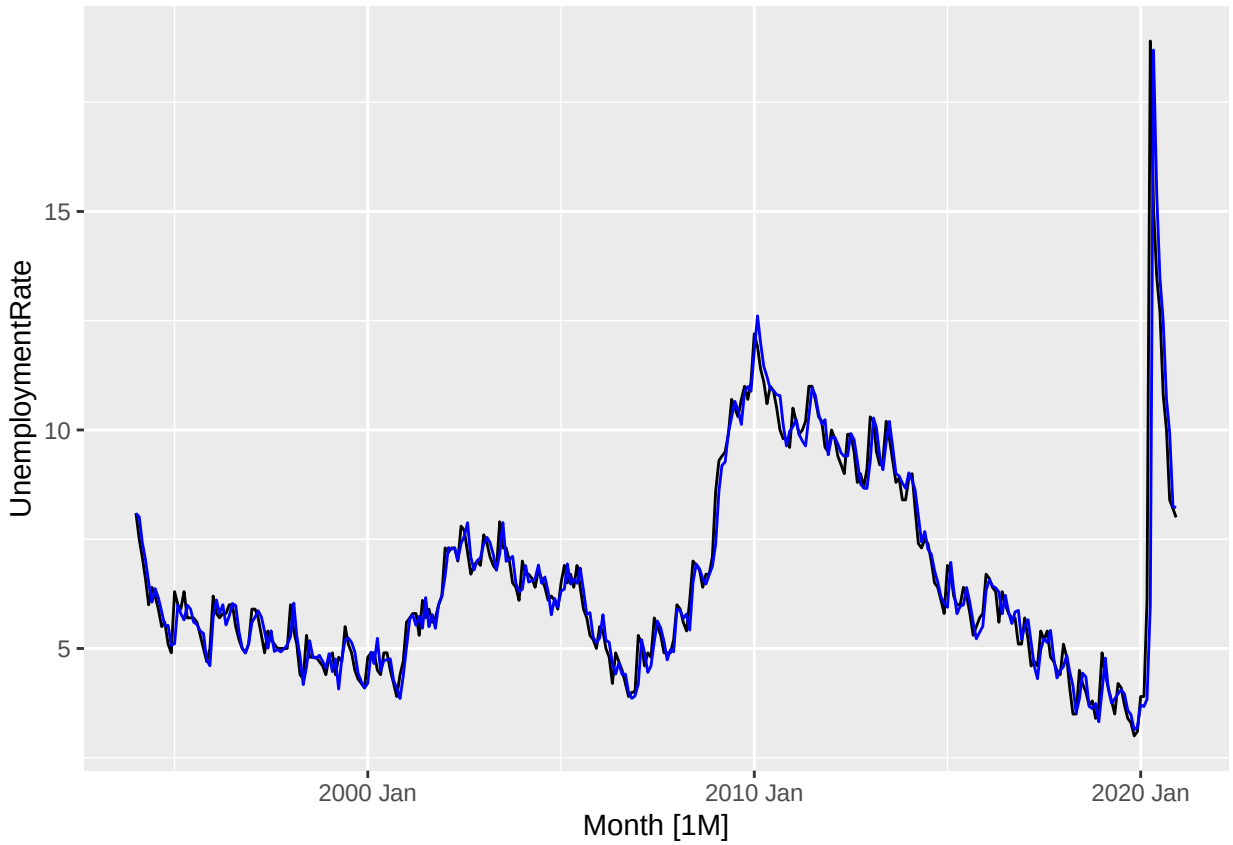
The ACF plot of the residuals of the automatically selected ETS model shows one very significant autocorrelation, and one borderline not statistically significant autocorrelation.

```
unemployment.fit %>%
  select(arimaauto) %>%
  gg_tsresiduals()
```



Although the ARIMA model performed worse on the test set, it appears to have captured the training set significantly better. There is no significant autocorrelation in the residuals, although the normally distributed assumption may be violated by the residuals, so bootstrapping may be more appropriate.

```
unemployment.fit %>%
  select(arimaauto) %>%
  augment() %>%
  autoplot(UnemploymentRate) + geom_line(aes(y = .fitted), color = "blue")
```



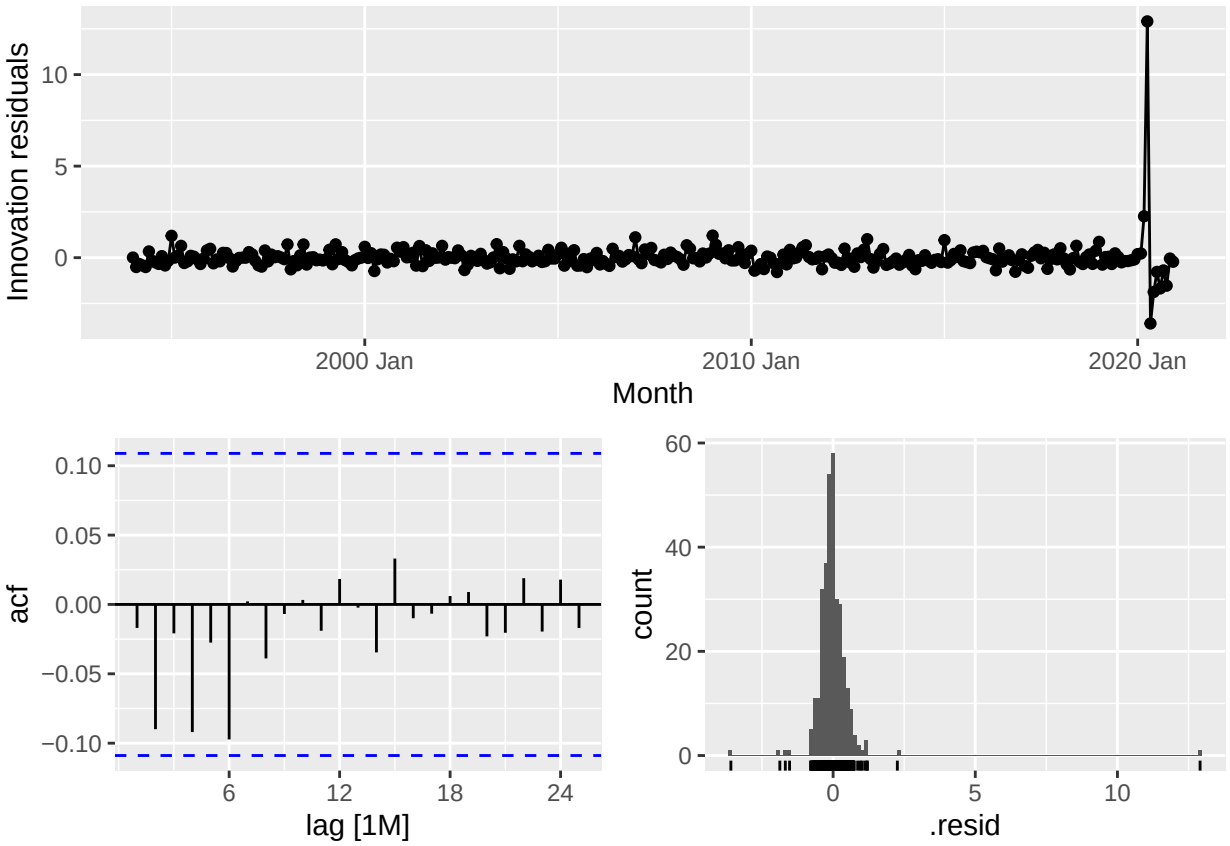
The ARIMA models appear to fit the training data better, and it's harder to distinguish $\hat{y}_t$ from y_t on the training data.

```
unemployment.fit
```

```
## # A mable: 1 x 4
##         etsinv                  arimainv                arimaauto       etsauto
##        <model>                   <model>                   <model>       <model>
## 1 <ETS(A,N,N)> <ARIMA(0,1,1)(1,0,0)[12]> <ARIMA(0,1,0)(0,0,2)[12]> <ETS(M,Ad,A)>
```

For the ETS model, the algorithm automatically selected multiplicative errors, additive damped trend, and additive seasonality. For the ARIMA model, it's chosen no autoregressive terms, 1 degree of differencing, and no lagged forecast errors for the non-seasonal part, whereas it's chosen no autoregressive terms, no differencing, and 2 lagged seasonal terms for the seasonal part.

```
unemployment.fit %>%
  select(arimaauto) %>%
  gg_tsresiduals()
```



Although the ARIMA model fares worse on future data, the residuals do indeed look like white noise, although the normality or skew assumption may be violated due to a few large outliers. It would appear that the ARIMA model may be the better of these models, but overall, none of the models appear to perform that well.

Conclusions and Future Work

In conclusion, the pandemic drastically changed the unemployment rate in Chicago. Although the ARIMA model was able to capture the past data, it failed to predict future data. The ETS model was closer to the actual data because its default predictions were lower than the ARIMA model, but the ARIMA model overall captured the unemployment data better. The inverse transformation did not improve the results for this particular dataset and in fact, the prediction intervals fared worse because of it.

However, there are some ways to get around that. Future work may include using a piecewise function for the dataset so that the years that the pandemic happened and/or other recessions are considered separately from other years. An issue with this data is that while cyclic recessions are indeed accounted for in other years, the pandemic had a much more drastic effect so that the models struggled to adapt. Additionally, I may consider using an indicator variable for the pandemic and/or other recessions, since in this case it was clear that the cause of the peak in unemployment was related to the pandemic.

Additionally, cross validation on the data exploration may improve the results somewhat since it's easier to predict data less far into the future, but even so, the changes with respect to the pandemic were pretty drastic and the models struggle to adapt.

Bibliography

Guerrero, V. M. (1993). Time-series analysis supported by power transformations. *Journal of Forecasting*, *12*(1), 37–48. https://doi.org/10.1002/for.3980120104

Labonte, M. (n.d.). *What Causes a Recession?*

Ma, C., Zhou, S., & Rogers, J. (2020). Modern Pandemics: Recession and Recovery. *International Finance Discussion Paper*, *2020*(1295). https://doi.org/10.17016/ifdp.2020.1295