W271 Assignment 7 (Solutions)

(14 points total) Question-1: Is Unemployment an Autoregressive or a Moving Average Process?

You did work in a previous homework to produce a data pipeline that pulled the unemployment rate from official BLS sources. Reuse that pipeline to answer this final question in the homework:

> "Are unemployment claims in the US an autoregressive, or a moving average process?

(1 point) Part-1: Why is the distinction important?

Why is it important to know whether a process is a AR or an MA (or a combination of the two) process? What changes in the ways that you would talk about the process, what changes in the ways that you would fit a model to the process, and what changes with how you would produce a forecast for this process?

- The AR(p) and the MA(q) models address different forms of dependence between observations over time. The class of AR(p) models describes the direct (and indirect) linear dependence between observations over time while MA(q) models describe the dependence in the innovation processes rather than the observations themselves. Separately, these classes of models therefore allow to take into account two extremely common dependence settings within time series.
- AR(p) models need to respect the condition of statianry (since they are always invertible) while MA(q) models need to respect the condition of invertibility (since they are always stationary).
- The ACF and PACF plots of AR(p) and MA(q) models are quite informative in terms of understanding which of the two classes of models an observed time series could have been generated from. The roles of the ACF and PACF in identifying the kind of AR(p) models underlying an observed time series is completely inversed when considering MA(q) models.
- AR(P) could be estimated using differnt methods including:

1) Method of moments estimator (e.g. Yule-Walker estimator)
2) Maximum Likelihood Estimation (MLE) estimator
3) Ordinary Least Squares (OLS) estimator

- MA(q) estimation is more difficult than AR model, but it could be estimated using MLE and Method of moment.

(1 point) Part-2: Pull in (and clean up) your data pipeline.

In the previous homework, you built a data pipeline to draw data from the BLS. We are asking you to re-use, and if you think it is possible, to improve the code that you wrote for this pipeline in the previous homework.

- Are there places where you took "shortcuts" that could be more fully developed?
- Are the processes that could be made more modular, or better documented so that they are easier for you to understand what they are doing? You've been away from the code that you wrote for a few weeks, and so it might feel like "discovering" the code of a *mad-person* (Who even wrote this???)

```
unemployment <- get_n_series_table(
  series_ids=list(
    overall='LNS14000000',
```

```
    male='LNS14000001',
    female='LNS14000002'),
  api_key = '21c01016e3a14d2888519292883a447a',
  start_year=2000,
  end_year=2023,
  tidy = TRUE
)
```

```
unemployment <- unemployment %>%
  mutate(time_index = make_datetime(year,month)) %>%
  mutate(time_index = yearmonth(time_index)) %>%
  as_tsibble(index=time_index) %>%
  select(time_index, overall)
```

```
head(unemployment)
```

```
## # A tsibble: 6 x 2 [1M]
##   time_index overall
##        <mth>   <dbl>
## 1   2000 Jan     4
## 2   2000 Feb     4.1
## 3   2000 Mar     4
## 4   2000 Apr     3.8
## 5   2000 May     4
## 6   2000 Jun     4
```

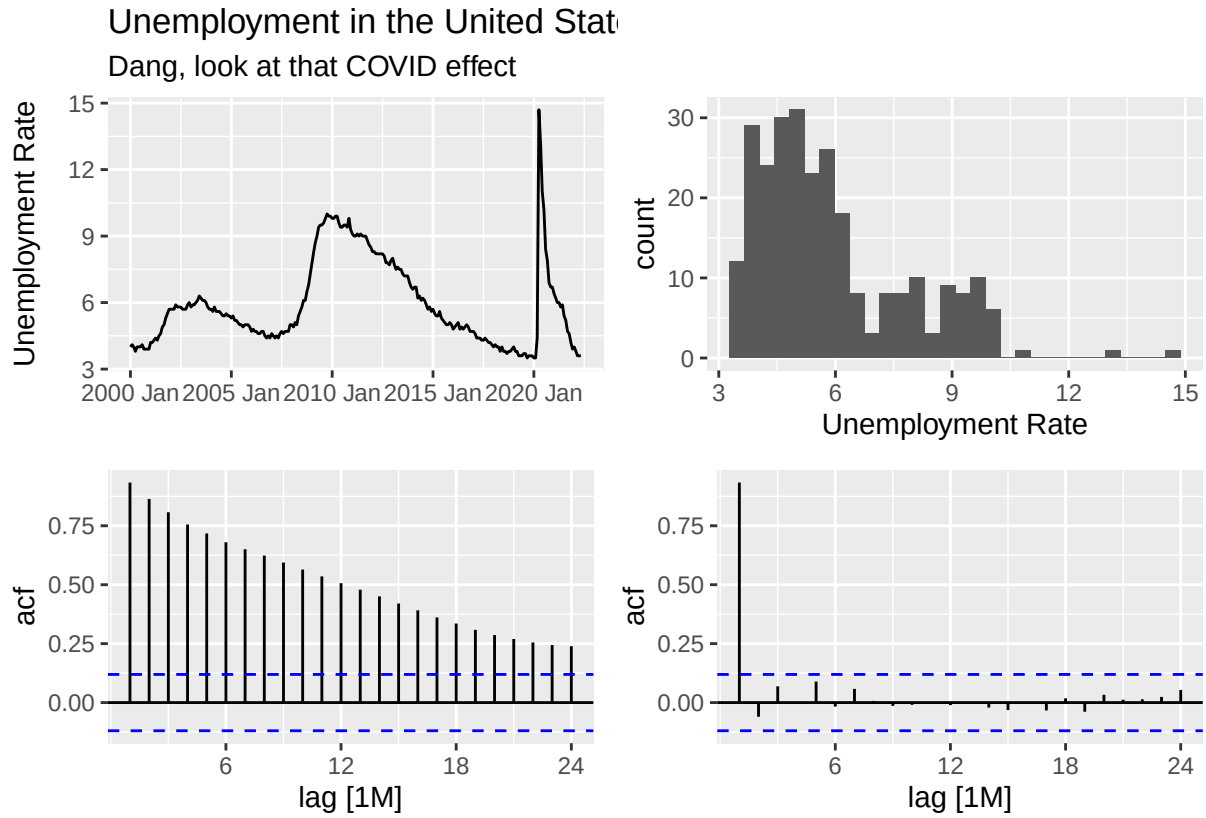
(5 points) Part-3: Conduct an EDA of the data and comment on what you see.

We have presented four **core** plots that are a part of the EDA for time-serires data. Produce each of these plots, and comment on what you see.

```
time_plot <- unemployment %>%
  ggplot() +
  aes(x=yearmonth(time_index),y=overall) +
  geom_line() +
  labs(
    title = 'Unemployment in the United States',
    subtitle = 'Dang, look at that COVID effect',
    x = NULL, y = 'Unemployment Rate',
    color = 'Emplyoment Group') +
  theme(legend.position = c(.2,.8))

overall_acf <- unemployment %>%
   ACF(y=overall) %>%
  autoplot()

overall_pacf <- unemployment %>%
  ACF(y=overall, type = "partial") %>%
  autoplot()

hist<- unemployment %>%
  ggplot() +
  geom_histogram(aes(x=overall)) +
  labs(
    x = 'Unemployment Rate') +
```

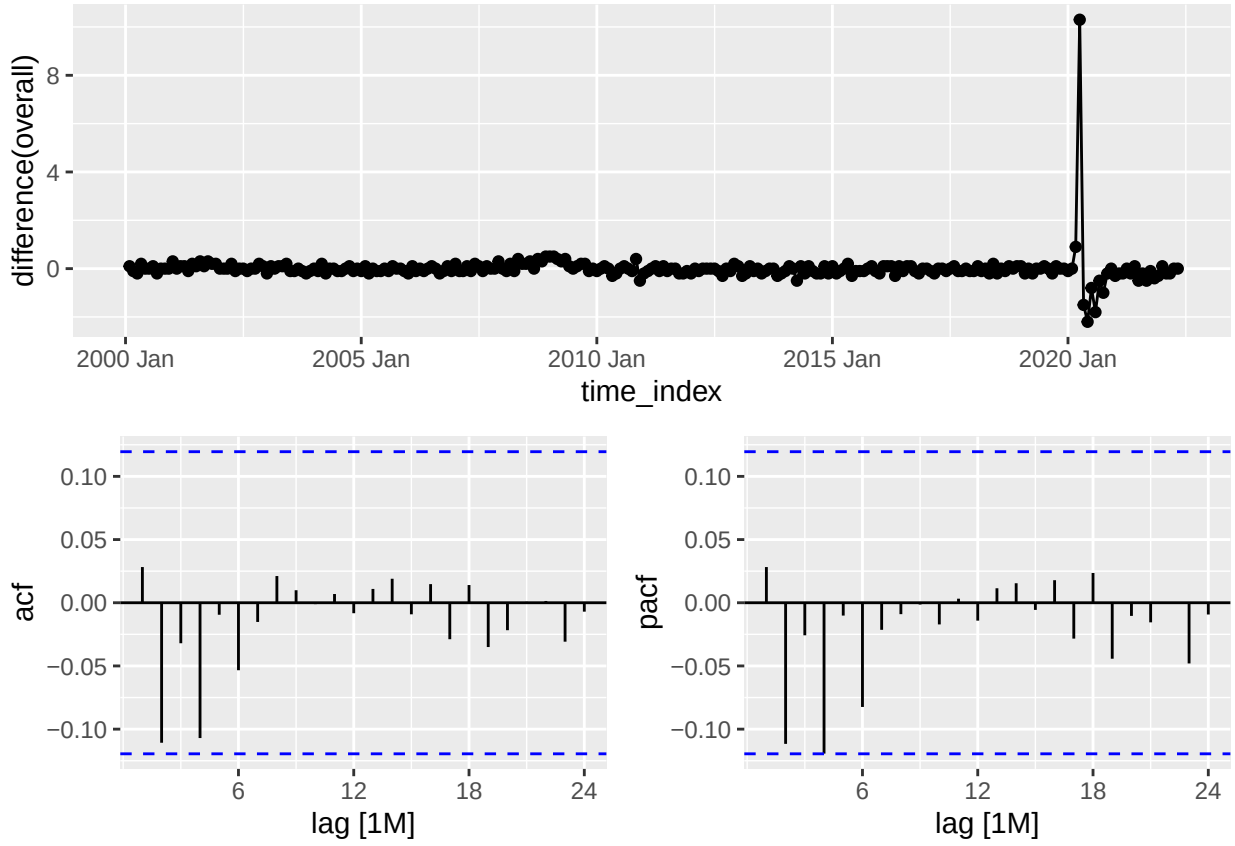
```
  theme(legend.position = c(.2,.8))

(time_plot + hist) /
  (overall_acf + overall_pacf)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- The time plot shows some non-stationarity. It has apparent trends up or down with sudden and unpredictable changes in direction.
- There is no evidence of changing variance, so we will not do a log transformation.
- Since this is probably a non-stationary series, its distribution change over and the histogram is not really informative.
- The ACF decays slowly, and the first bar of the PACF is almost equal to 1.
- To address the non-stationarity, we will take a first difference of the data. The differenced data are shown blow:

```
unemployment %>%
  gg_tsdisplay(difference(overall), plot_type='partial')
```



- The differnced series now appear to be stationary.
- The ACF and PACF shows that all autocorrelations are within the threshold limits, indicating that the differenced series is behaving like white noise.
- So an initial candidate model is an ARIMA(0,1,0).

(1 point) Part-4: Make a Call

Based on what you have plotted and written down in the previous section, would you say that the unemployment rate is an AR, MA or a mix of the two?

The first-differncing of the time series has allowed to make it apparently stationary so an ARIMA(p,d,q) model with d=1 could be a good class of models to explain and predict the time series. The ACF and PACF plots would suggest ARIMA(0,1,0) since both the ACF and PACF are insignificant starting from the first lags.

This means that unemployment rate is a random walk process, and the changes in unemployment rate is a white noise process and unpredictable, and our best guess about the future uenmployment rate is today's unemployment rate.

(6 points total) Part-5: Estimate a model

Report the best-fitting parameters from the best-fitting model, and then describe what your model is telling you. In this description, you should:

- (1 point) State, and justify your model selection criteria.
- (1 point) Interpret the model selection criteria in context of the other models that you also fitted.

- (2 points) Interpret the coefficients of the model that you have estimated.
- (2 points) Produce and interpret the model diagnostic plots to evaluate how well your best-fitting model is performing.
- (1 (optional) point) If, after fitting the models, and interpreting their diagnostics plots, you determine that the model is doing poorly – for example, you notice that the residuals are not following a white-noise process – then, make a note of the initial model that you fitted then propose a change to the data or the model in order to make the model fit better. If you take this action, you should focus your interpretation of the model's coefficients on the model that you think does the best job, which might be the model after some form of variable transformation.
- We need to use a unit root test to confirm whether differencing is required:

```
unemployment %>%
  features(overall, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1     0.502      0.0412
```

```
unemployment %>%
  mutate(diff_value = difference(overall)) %>%
  features(diff_value, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1    0.0830         0.1
```

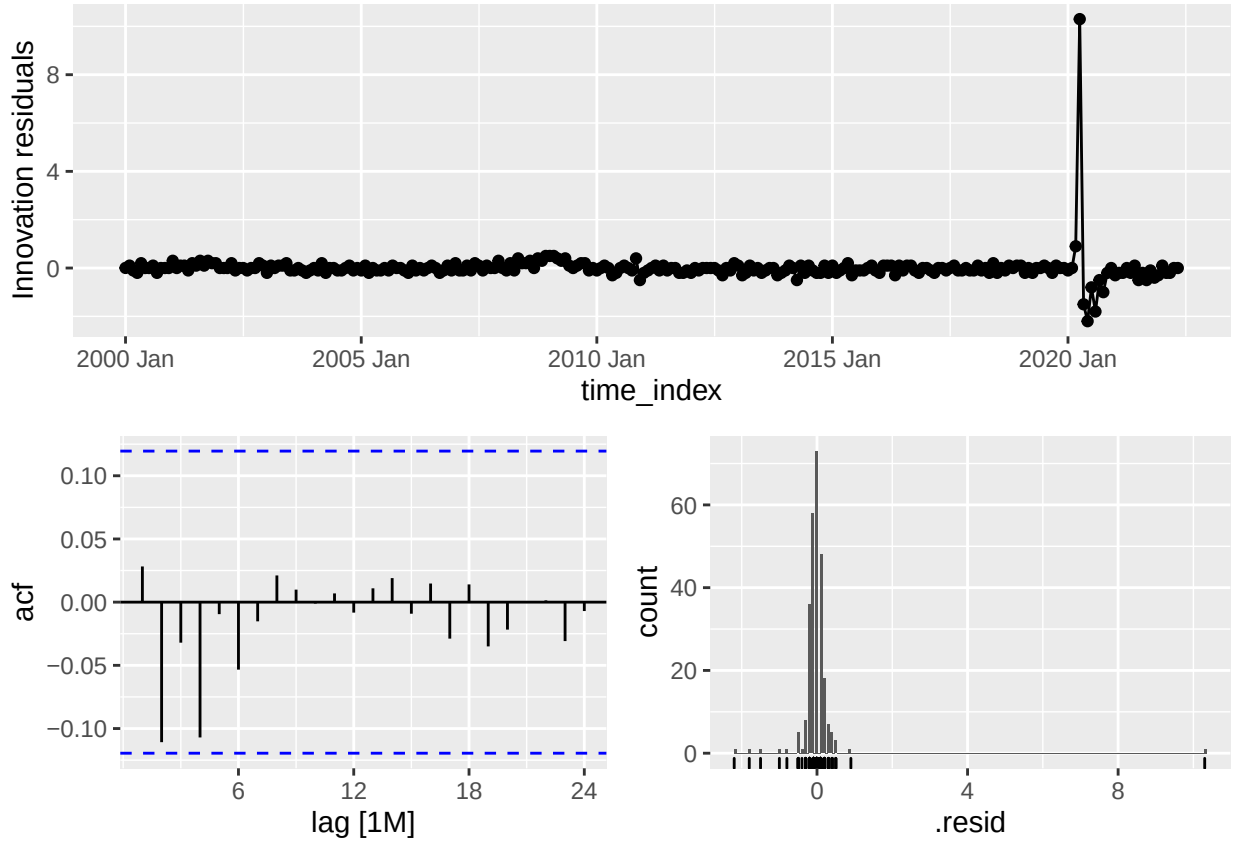
- In KPSS unit root test the null hypothesis is that the series is stationary.
- For the unemployment rate, the p-value is 0.04122915, which is less than 0.05, indicating that the null hypothesis of stationary is rejected. So first difference is required.
- For the difference of the unemployment rate, the p-value is 0.1, and it is greater than 0.05, so we fail to reject the null hypothesis of stationary.
- Only one difference is required to make the uenployemnt stationary.
- We use ARIMA() to estimate a ARIMA model with lowest corrected AIC.

```
unemployment_fit <- unemployment %>%
  model(
arima_fit = ARIMA(overall)
 ) %>%
  report()
```

```
## Series: overall
## Model: ARIMA(0,1,0)
##
## sigma^2 estimated as 0.4716:  log likelihood=-279.57
## AIC=561.14   AICc=561.15   BIC=564.73
```

- The ARIMA() has found that an ARIMA(0,1,0) gives the lowest AICc value, which confirmS our initial guess.
- The standard deviation of the residual is 0.460 which is an estimate for the varaince of white noise process.
- In th next step we need to do the model diagnostic using residuals plot and the Ljung-box test.

```
unemployment_fit %>%
  gg_tsresiduals()
```



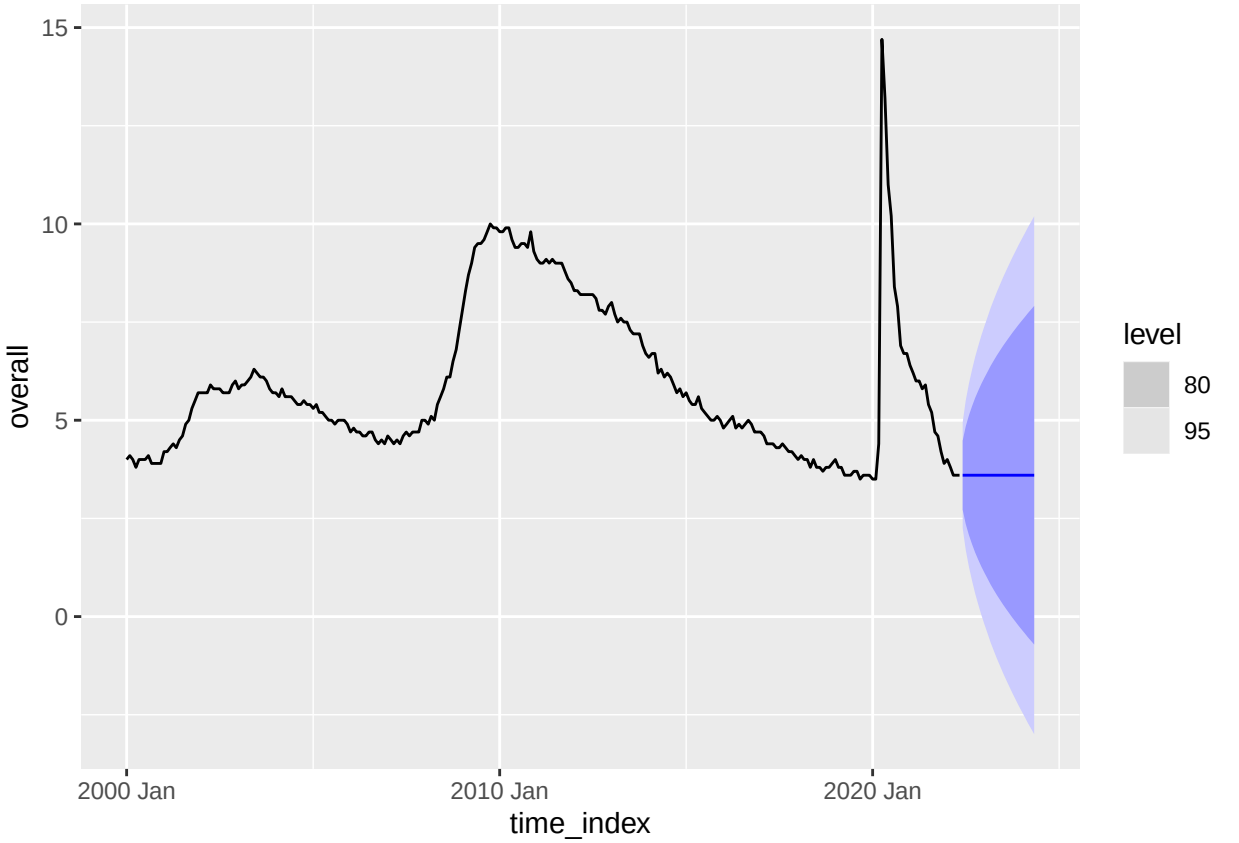
- The ACF plot of the residuals from the ARIMA(0,1,0) model shows that all autocorrelations are within the threshold limits, indicating that the residuals are behaving like white noise.

```
augment(unemployment_fit) %>%
  features(.innov, ljung_box, lag = 10, dof = 0)
```

```
## # A tibble: 1 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 arima_fit    8.03     0.626
```

- A ljung-box test returns a large p-value, also suggesting that the residuals are white noise.
- Finally let's use our fitted model to forecast next two years unemployment rate.

```
unemployment_fit %>%
  forecast(h=24) %>%
  autoplot(unemployment)
```



- As we expect for a random walk process, all the forecasts of unemployment rate are constant and equal to the last observed unemployemnt rate , and the prediction interval increase with the forecast's horizon.

(14 Points Total) Question-2: COVID-19

The United States Centers for Disease Control maintains the authoritative dataset of confirmed and probable COVID-19 cases.

- This data is described on this page [link].
- The data is made available via an API link on this page as well.

(1 point) Part-1: Access Data

Use the public API to download the CDC COVID-19 data and store in a useful dataframe. A useful dataframe:

- Should have useful variable names;
- Should be in a format that can be used for time series modeling;

- Should have appropriate time indexes (and possibly keys) set; but,
- At this point, should not have derivative features mutated onto the data frame; nor,
- Should it be aggregated or summarized.

```
#we get the data directly from the website as a CSV instead of using the API endpoint
dat<-read.csv("./covid_case_data.csv")
```

```
head(dat)
```

```
##   submission_date state tot_cases conf_cases prob_cases new_case pnew_case
## 1      12/01/2021    ND    163565     135705      27860      589       220
## 2      11/07/2021    DE    143685     132310      11375      296        30
## 3      05/12/2022    CT    777064     696528      80536     1963       173
## 4      10/04/2020    MD    127290         NA         NA      471         0
## 5      02/06/2020    NE         0         NA         NA        0        NA
## 6      02/02/2021    IL   1130917    1130917          0     2304         0
##   tot_death conf_death prob_death new_death pnew_death             created_at
## 1      1907         NA         NA         9          0 12/02/2021 02:35:20 PM
## 2      2186       1992        194         3          0 11/09/2021 12:00:00 AM
## 3     10883       8906       1977         0          0 05/13/2022 01:28:57 PM
## 4      4092       3933        159         3          0 10/06/2020 12:00:00 AM
## 5         0         NA         NA         0         NA 03/26/2020 04:22:39 PM
## 6     21336      19306       2030        63         16 02/03/2021 02:55:58 PM
##   consent_cases consent_deaths
## 1         Agree      Not agree
## 2         Agree          Agree
## 3         Agree          Agree
## 4           N/A          Agree
## 5         Agree          Agree
## 6         Agree          Agree
```

```
colnames(dat)
```

```
##  [1] "submission_date" "state"           "tot_cases"       "conf_cases"
##  [5] "prob_cases"      "new_case"        "pnew_case"       "tot_death"
##  [9] "conf_death"      "prob_death"      "new_death"       "pnew_death"
## [13] "created_at"      "consent_cases"   "consent_deaths"
```

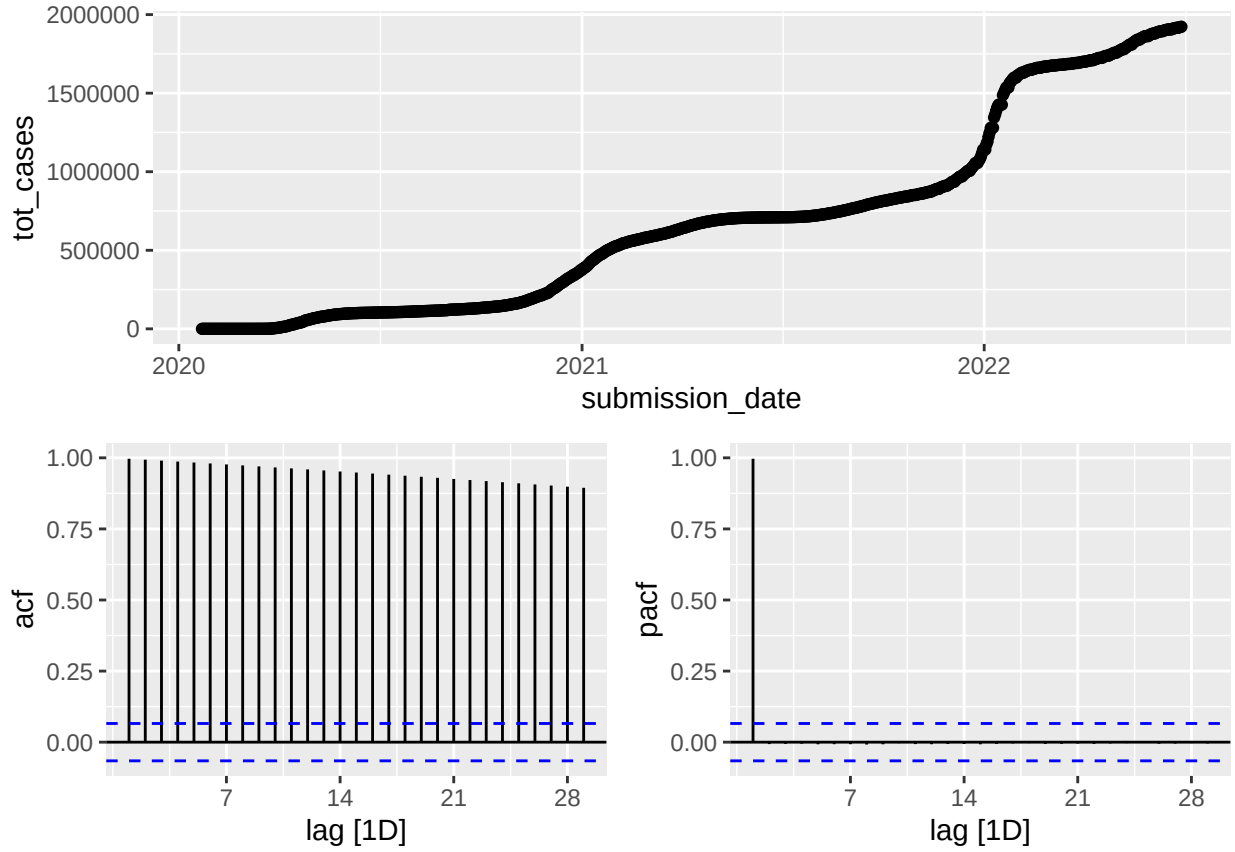
```
dat<-dat %>%
  mutate(submission_date=as.Date(submission_date,format="%m/%d/%Y")) %>%
  select(submission_date,state,tot_cases) %>%
  arrange(state,submission_date) %>%
  as_tsibble(key=state,index=submission_date)
```

(5 points) Part-2: Pick a State and Produce a Model

1. Choose a state that is not California (we are putting this criteria in so that we see many different states chosen);
2. Produce a 7-day, backward smoother of the total case rate; then,
3. Produce a model of COVID cases in that state. This should include:

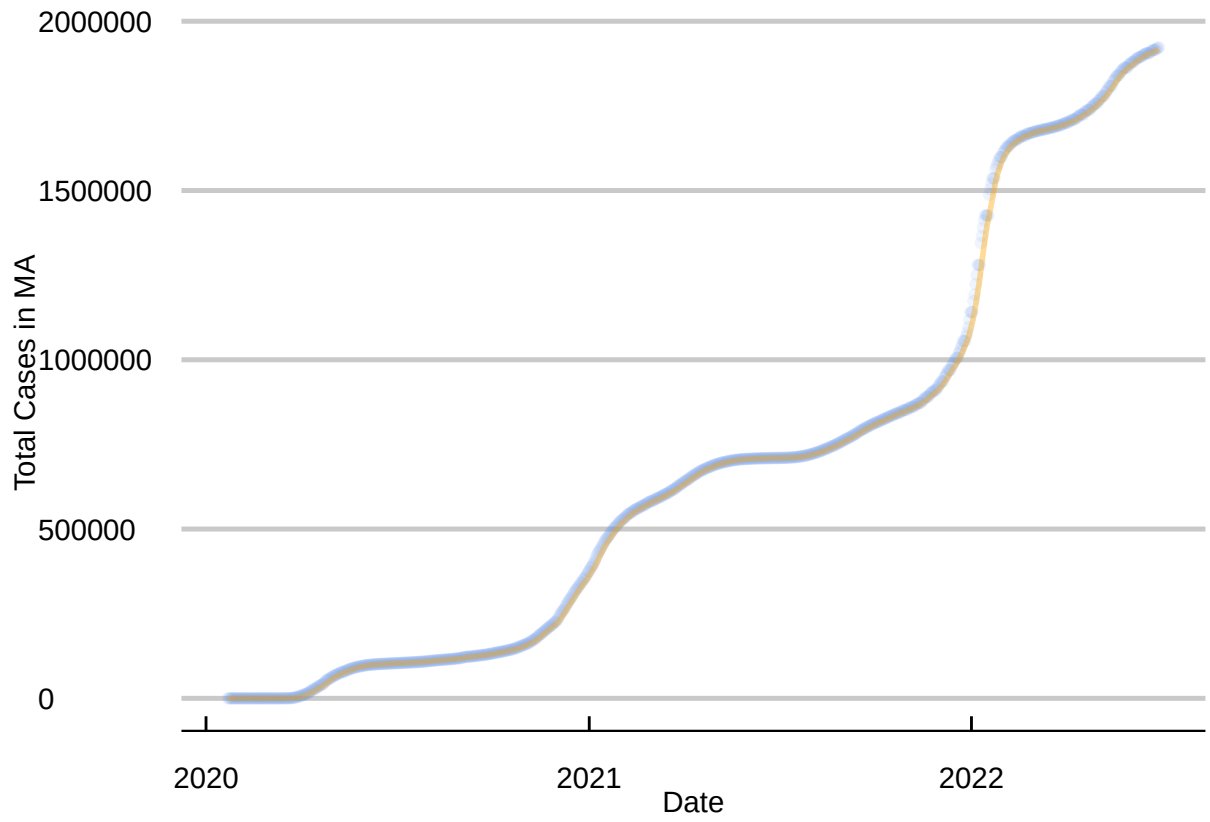
- Conducting a full EDA and description of the data that you observe
- Estimating a model (either AR or MA) that you believe is appropriate after conducting your EDA
- Evaluating the model performance through diagnostic plots
- We pull COVID cases in the state of Massachusetts.
- The plot below shows that the total cases has a non linear trend. The ACF is characteristic of a process with a trend. The PACF plot only has a significant term at lag 1.
- When data is nonlinear / nonconstant like this, it can be helpful to work off of the log of the series.

```
ma.dat<-dat %>%
  filter(state=="MA")

ma.dat %>%
  gg_tsdisplay(tot_cases,plot_type="partial")
```



- The plot below shows both the original daily total cases in MA and also the 7 day moving average over time. Both overap a lot because the time series is already overly smooth given it is total daily cases which does not change much day to day.

```
ma.dat<-ma.dat %>%
  mutate(avg7d=rollmean(tot_cases,k=7,fill=0,align="right"))

ma.dat %>%
  ggplot(aes(submission_date,tot_cases)) +
  geom_point(alpha=0.1,color="cornflowerblue") +
  geom_line(aes(submission_date,avg7d),lwd=1,alpha=0.4,col="orange1") +
  theme_economist_white(gray_bg=F) +
  xlab("Date") +
  ylab("Total Cases in MA")
```



- We use BIC to fit the optimal model below, which results in an AR model with 6 lags and an order of differencing of 2. All selected coefficients are significant.

```
model.fit<-ma.dat %>%
  model(ts.model=ARIMA(tot_cases~0+pdq(0:10,0:2,0:10)+PDQ(0,0,0),ic="bic",greedy=F,stepwise=F))

model.fit$ts.model
```

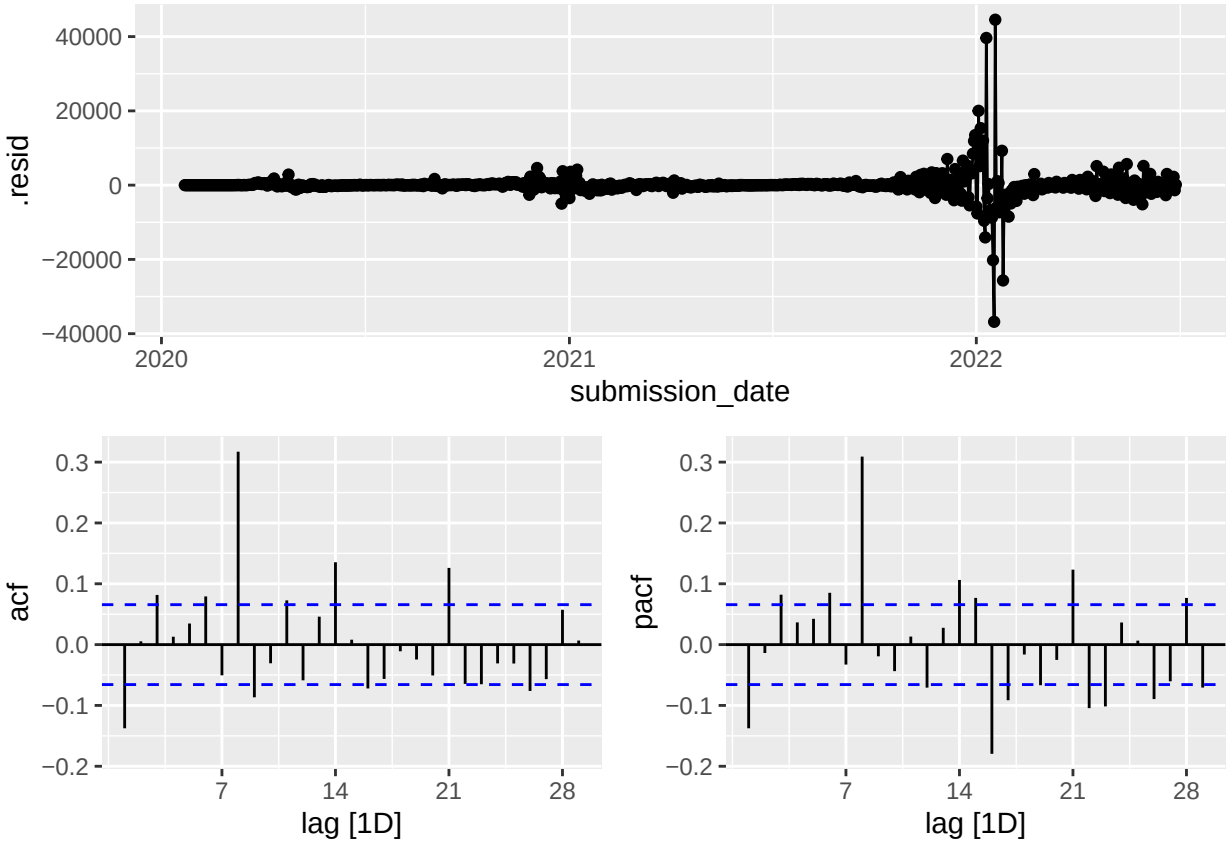
```
## <lst_mdl[1]>
## [1] <ARIMA(6,2,0)>
```

```
model.fit %>% coef()
```

```
## # A tibble: 6 x 7
##   state .model   term  estimate std.error statistic   p.value
##   <chr> <chr>    <chr>    <dbl>     <dbl>     <dbl>     <dbl>
## 1 MA    ts.model ar1     -0.901    0.0302     -29.8 5.89e-136
## 2 MA    ts.model ar2     -0.885    0.0365     -24.2 6.39e-100
## 3 MA    ts.model ar3     -0.763    0.0407     -18.7 3.45e- 66
## 4 MA    ts.model ar4     -0.706    0.0406     -17.4 2.05e- 58
## 5 MA    ts.model ar5     -0.663    0.0364     -18.2 4.33e- 63
## 6 MA    ts.model ar6     -0.430    0.0301     -14.3 8.70e- 42
```

- The plot below shows the residuals are not well behaved and there are some significant lags still.

```
model.fit %>%
  augment() %>%
  gg_tsdisplay(.resid,plot_type="partial")
```



- The Box Ljung test also rejects that the null hypothesis that the autocorrelation in the residuals is zero as expected.

```
model.fit %>%
  augment() %>%
  features(.resid,ljung_box,lag=10)
```

```
## # A tibble: 1 x 4
##   state .model   lb_stat lb_pvalue
##   <chr> <chr>      <dbl>     <dbl>
## 1 MA    ts.model    130.         0
```

- Given the results above we fit a new model with additional parameters from the optimal version chosen by BIC. Because the ACF plot exhibits some cylicality and significant lags, we opt to add 8 MA lags.

```
model.fit2<-ma.dat %>%
  model(ts.model=ARIMA(tot_cases~0+pdq(6,2,8)+PDQ(0,0,0),ic="bic",greedy=F,stepwise=F))

model.fit2$ts.model
```

```
## <lst_mdl[1]>
## [1] <ARIMA(6,2,8)>
```

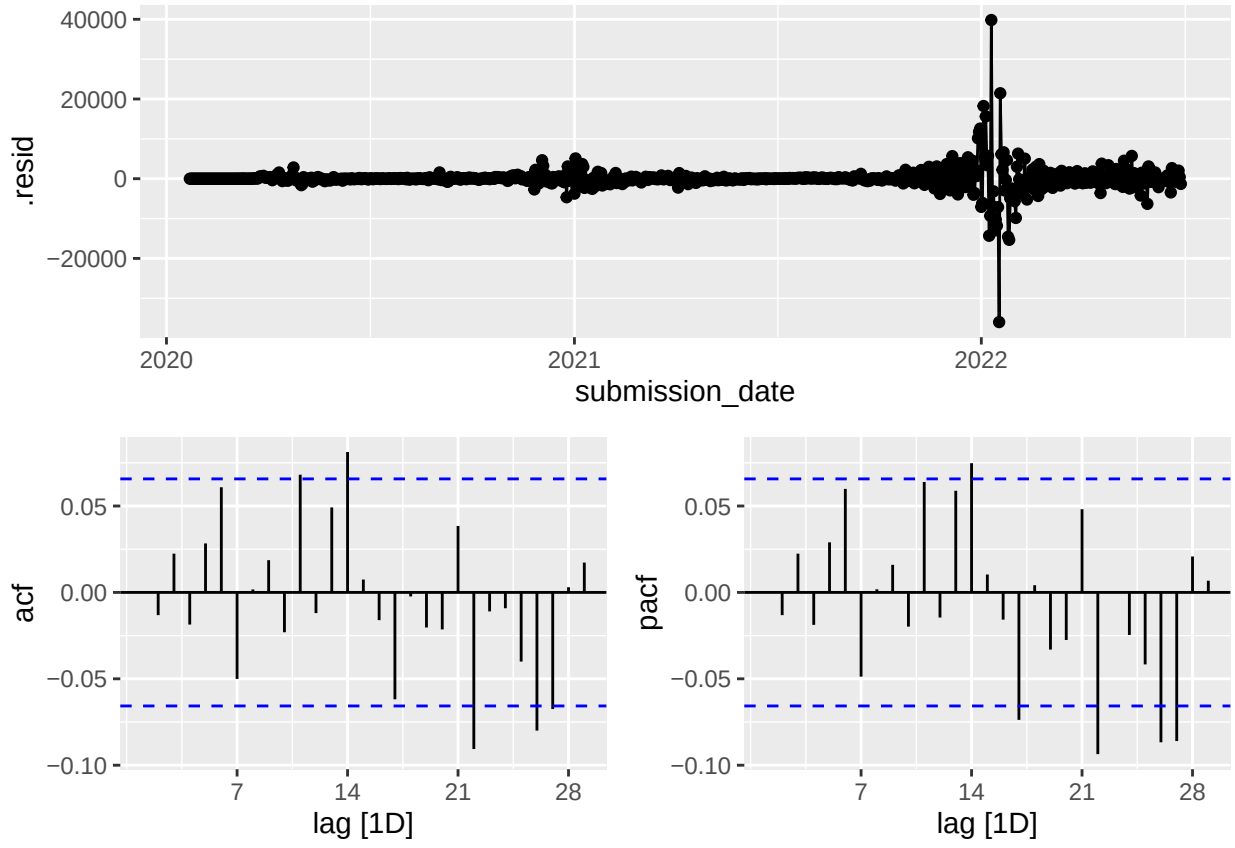
```
model.fit2 %>% coef()
```

```
## # A tibble: 14 x 7
##    state .model   term  estimate std.error statistic   p.value
##    <chr> <chr>    <chr>    <dbl>     <dbl>     <dbl>     <dbl>
##  1 MA    ts.model ar1    -1.09      0.0404   -26.9   4.52e-117
```

```
##  2 MA    ts.model ar2    -1.09      0.0440   -24.8   1.79e-103
##  3 MA    ts.model ar3    -1.11      0.0509   -21.7   2.56e- 84
##  4 MA    ts.model ar4    -0.925     0.0573   -16.1   1.26e- 51
##  5 MA    ts.model ar5    -0.926     0.0370   -25.0   5.26e-105
##  6 MA    ts.model ar6    -0.743     0.0496   -15.0   2.06e- 45
##  7 MA    ts.model ma1     0.0767    0.0401     1.91  5.60e-  2
##  8 MA    ts.model ma2     0.0920    0.0378     2.43  1.51e-  2
##  9 MA    ts.model ma3     0.172     0.0391     4.40  1.21e-  5
## 10 MA    ts.model ma4    -0.0591    0.0432    -1.37  1.72e-  1
## 11 MA    ts.model ma5     0.127     0.0440     2.89  3.99e-  3
## 12 MA    ts.model ma6     0.0284    0.0516     0.551 5.82e-  1
## 13 MA    ts.model ma7    -0.285     0.0397    -7.19  1.41e- 12
## 14 MA    ts.model ma8     0.539     0.0333    16.2   6.98e- 52
```

- The residuals plots seem much better behaved, especially up to lag 10, and the Box Ljung test does not reject the null hypothesis up to lag 10, meaning we have statistical evidence that the residuals behave like white noise.

```
model.fit2 %>%
  augment() %>%
  gg_tsdisplay(.resid,plot_type="partial")
```




```
model.fit2 %>%
  augment() %>%
  features(.resid,ljung_box,lag=10)
```

```
## # A tibble: 1 x 4
##   state .model   lb_stat lb_pvalue
```

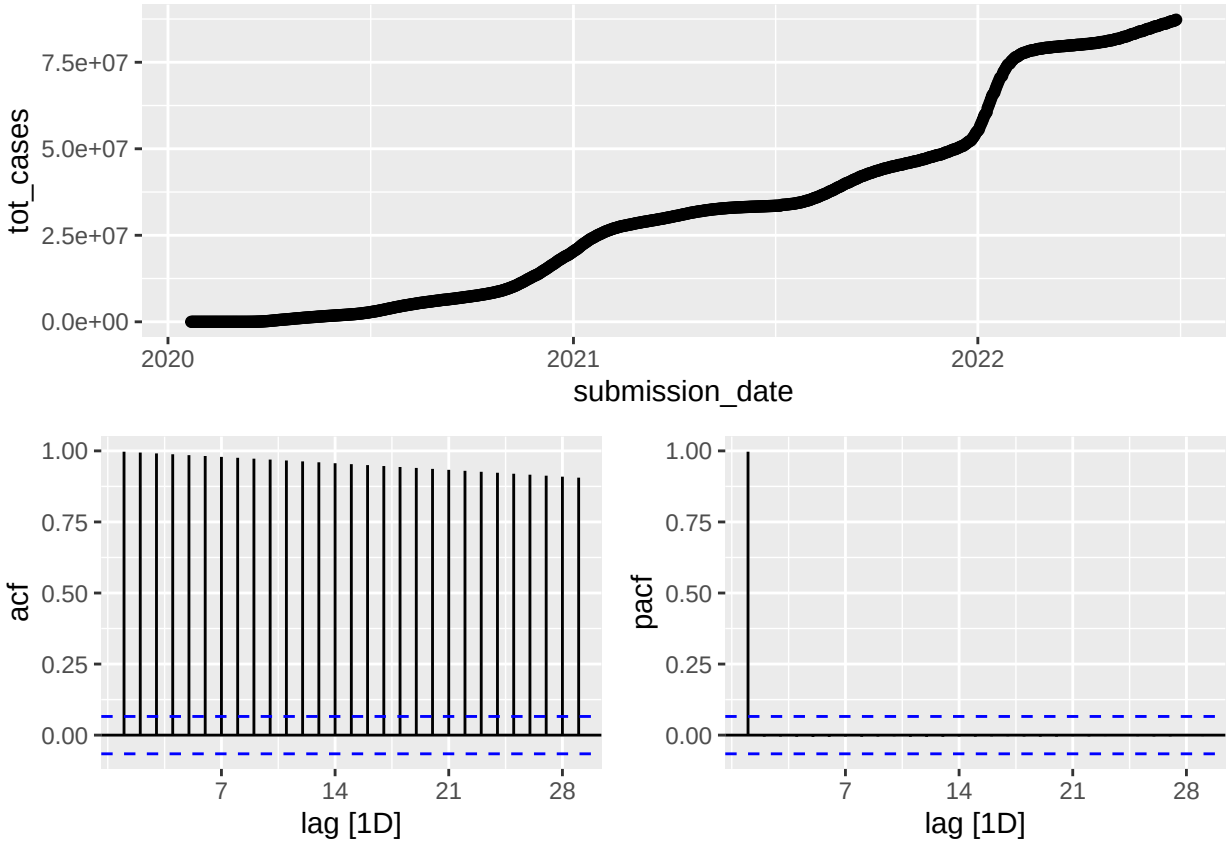
```
##   <chr> <chr>       <dbl>      <dbl>
## 1 MA    ts.model     8.01      0.628
```

#(5 points) Part-3: Produce a Nationwide Model

1. Aggregate the state-day data into nationwide-day level data;
2. Produce a 7-day, backward smoother of the total case rate; then,
3. Produce a model of COVID cases across the US. Like the state model, this should include:

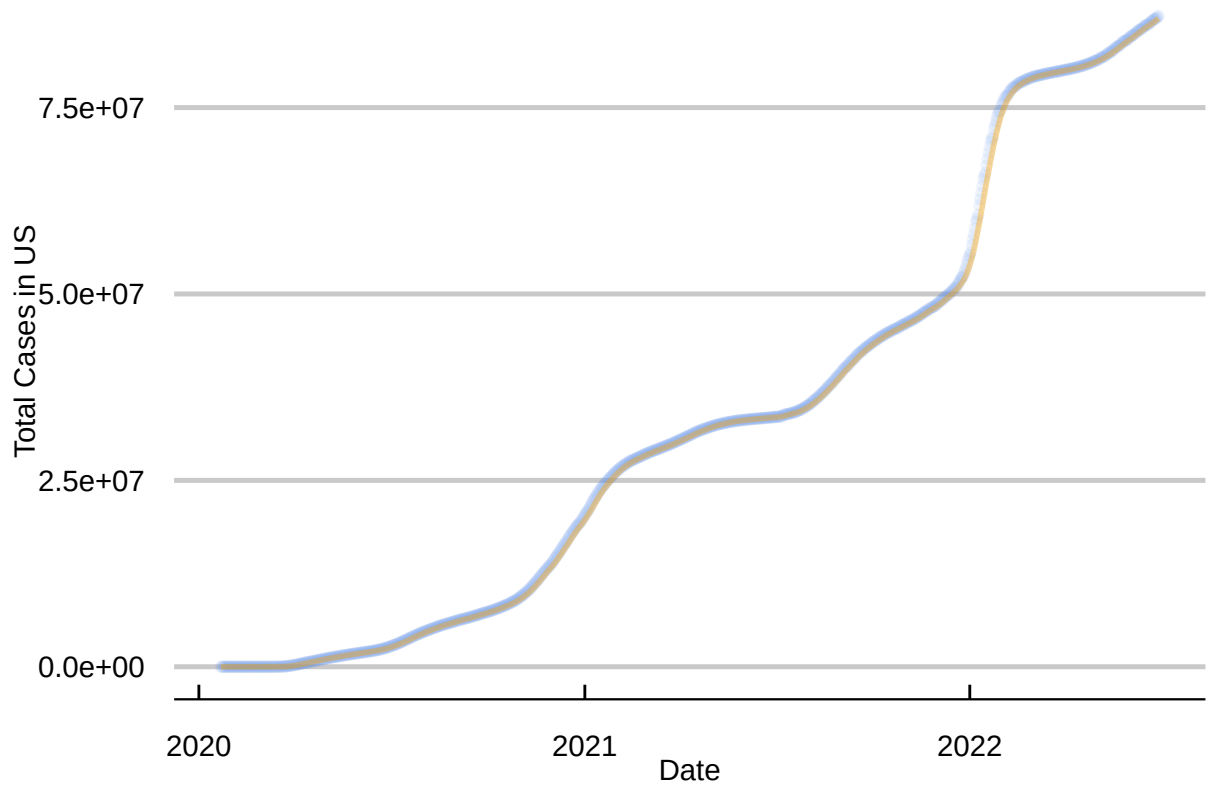
- Conducting a full EDA and description of the data that you observe
- Estimating a model (either AR or MA) that you believe is appropriate after conducting your EDA
- Evaluating the model performance through diagnostic plots
- We repeat the same steps as above. The plot of the national data is very similar to the plot of MA data, along with the ACF and PACF plots.

```
nat.dat<-dat %>%
  as_tibble() %>%
  group_by(submission_date) %>%
  summarize(tot_cases=sum(tot_cases),.groups="drop") %>%
  as_tsibble(index=submission_date)

nat.dat %>%
  gg_tsdisplay(tot_cases,plot_type="partial")
```



- Here is the time series plot with a 7 day smoother.

```
nat.dat<-nat.dat %>%
  mutate(avg7d=rollmean(tot_cases,k=7,fill=0,align="right"))

nat.dat %>%
  ggplot(aes(submission_date,tot_cases)) +
  geom_point(alpha=0.1,color="cornflowerblue") +
  geom_line(aes(submission_date,avg7d),lwd=1,alpha=0.4,col="orange1") +
  theme_economist_white(gray_bg=F) +
  xlab("Date") +
  ylab("Total Cases in US")
```



- Unsuprisingly, BIC chooses the same national model as the state model.

```
model.fit3<-nat.dat %>%
  model(ts.model=ARIMA(tot_cases~0+pdq(0:10,0:2,0:10)+PDQ(0,0,0),ic="bic",greedy=F,stepwise=F))

model.fit3$ts.model
```

```
## <lst_mdl[1]>
## [1] <ARIMA(6,2,0)>
```

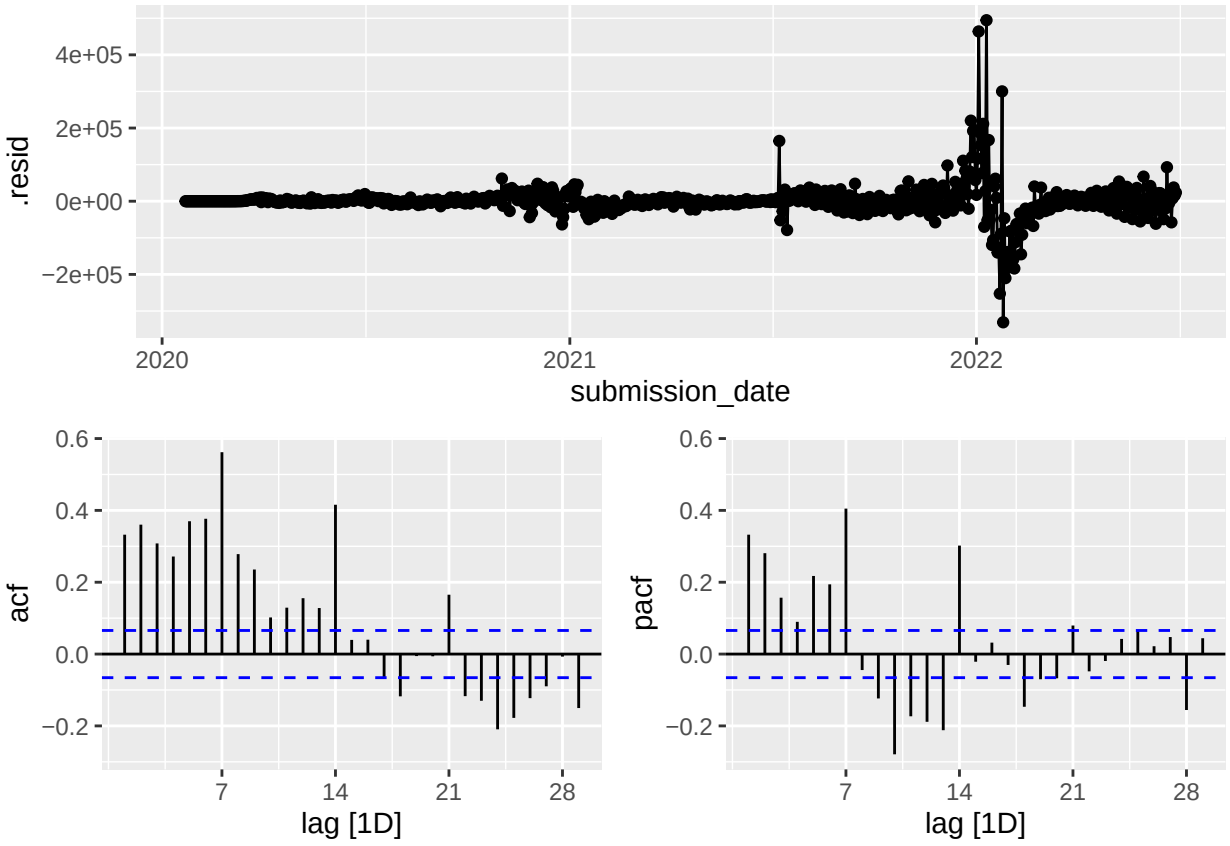
```
model.fit3 %>% coef()
```

```
## # A tibble: 6 x 6
##   .model   term  estimate std.error statistic  p.value
##   <chr>    <chr>    <dbl>     <dbl>     <dbl>    <dbl>
## 1 ts.model ar1     -0.631    0.0278     -22.7 3.68e-90
## 2 ts.model ar2     -0.626    0.0295     -21.2 3.30e-81
```

```
## 3 ts.model ar3     -0.449    0.0332     -13.5 5.00e-38
## 4 ts.model ar4     -0.436    0.0331     -13.2 2.90e-36
## 5 ts.model ar5     -0.560    0.0294     -19.0 7.28e-68
## 6 ts.model ar6     -0.556    0.0278     -20.0 8.96e-74
```

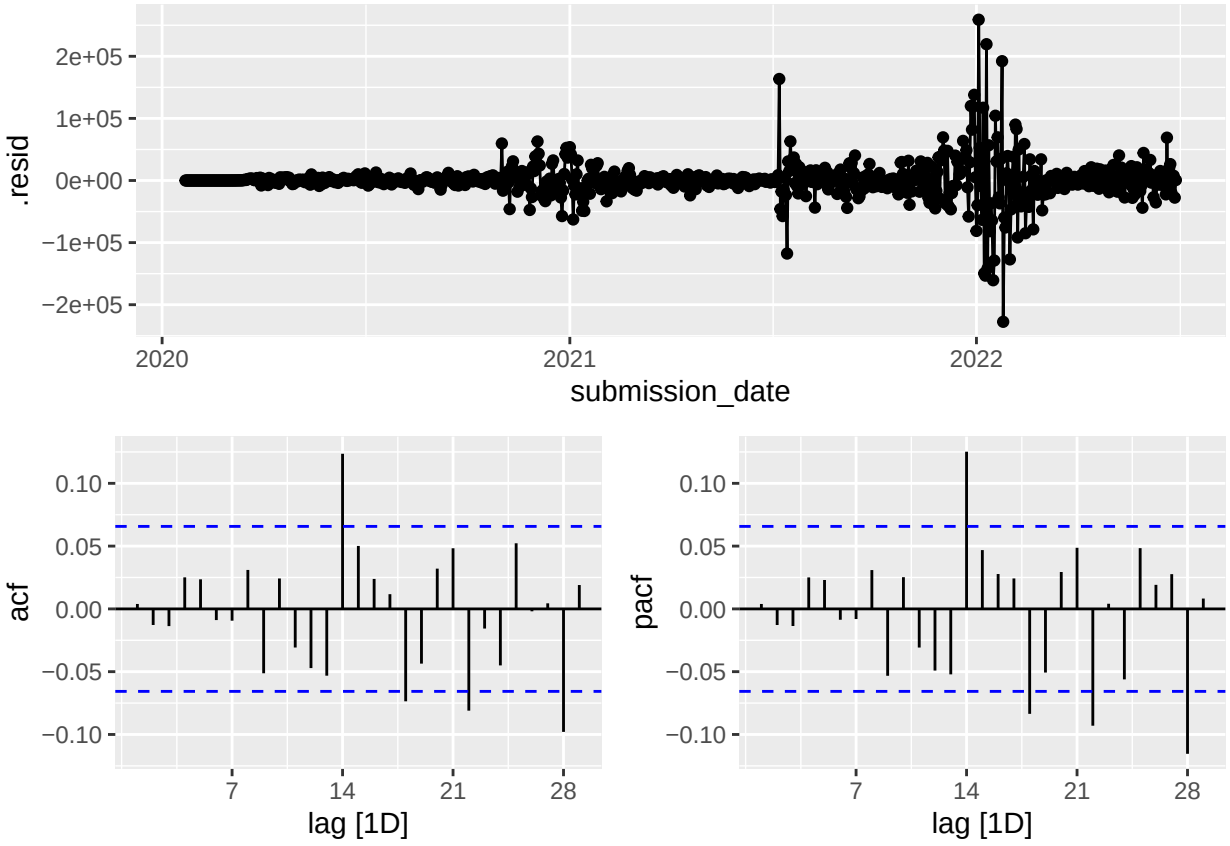
- The residual plot makes clear that this is not an adequete model. Given the strong ACF plot, we add 5 ma terms and given the PACF plot, we also add 2 more AR terms.

```
model.fit3 %>%
  augment() %>%
  gg_tsdisplay(.resid,plot_type="partial")
```



- The residual plots are better behaved now though there is somewhat of a seasonal pattern suggesting SARIMA may be more appropriate. We do fail to reject the null hypothesis of zero autocorrelation up to lag 10 in the Box Ljung test though.

```
model.fit4<-nat.dat %>%
  model(ts.model=ARIMA(tot_cases~0+pdq(10,2,5)+PDQ(0,0,0),ic="bic",greedy=F,stepwise=F))

model.fit4 %>%
  augment() %>%
  gg_tsdisplay(.resid,plot_type="partial")
```




```
model.fit4 %>%
  augment() %>%
  features(.resid,ljung_box,lag=10)
```

```
## # A tibble: 1 x 3
##   .model   lb_stat lb_pvalue
##   <chr>      <dbl>     <dbl>
## 1 ts.model    5.31     0.870
```

(3 points) Part-4: Write a few paragraphs about this modeling task

The nationwide model that you just produced contains much **more** data than went into your state-level model. Does this make it a better model? Why or why not?

Without a requirement that you actually produce the model that you propose: If you were trying to produce a nationwide model, knowing: (a) what you know about the state model that you fit; (b) what you know about the nationwide model that you fit; and (c) what you, as a citizen of this world who has lived through these past years: *propose a modeling strategy you think will produce the best nationwide forecasting model.*

This could be, for example, the nationwide model that you have fitted above. Or, you might propose some other forms of data aggregation before modeling, or model aggregation but not data aggregation. In writing about your strategy, justify choices that you are making.

Our goal with this question is to ask that you not only conduct the narrow technical work, but also that you do the higher-level reasoning about the technical work. We would like you to write in full paragraphs, rather than bullet points that address specific parts of the prompt above.

- The national model is not naturally better because even though it has more data, that data is not full

independent. We know state COVID cases are highly correlated because of spillover / infections from people traveling across state lines. Hence, there is not really a lot more data going into the national model vs. the state model.

- To best forecast national COVID cases, one option that generally works well is called ensembling. The idea is we combine forecasts from several time series models together. Averaging models tends to work well because it helps errors in model one be cancelled out by other models.
- For example, we could use an ensemble of (1) ARIMA model of nationwide covid cases and (2) a sum of 50 ARIMA models with one for each state. We could combine models by simply averaging their predictions or weight them by error on a test set.
- Model (2) is what is known as a hierarchical time series model because we are forecasting individual components of a total and summing to get a forecast of the total. In particular this is a bottoms up forecasting process.