

W271 Assignment 8

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

(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? > Knowing what model fits the time-series better will allow us to select the model that is more likely to make valid predictions. Autoregressive models imply that there is a strong influence of one day over another, while moving average models assume that there are random shocks that happen independently. Therefore, AR model will take into account shocks that go beyond its order p , while MA has no memory whatsoever about what happened beyond the lag of its order q . Producing forecast for AR(p) model we would use a linear regression and last p values, while forecasting with MA(q) model we would use the last q shocks.

(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???)

```
# Generate a list of timeseries we would like to obtain
series_ids <- list(uer.total = "LNS14000000",
                  uer.men = "LNS14000001",
                  uer.women = 'LNS14000002')

# API call to retrieve the data
uer_series <- get_n_series_table(series_ids,
                                '913c5fe448924ea584e043f736c95080',
                                start_year = 2000,
                                end_year = 2023)
```

```
## Year 2000 to 2023 is longer than 20 year API limit. Performing 2 requests.
```

```
# Generate time period variable from the text column for month
uer_series$period <- as.integer(str_sub(uer_series$period, 2, 3))
```

```

# Set descriptive names
names(uer_series) <- c("year", "month", "overall", "male", "female")

# Produce tidyverse-friendly long form of the data frame
unemployment <- pivot_longer(uer_series, cols = 3:5,
                             names_to = "name",
                             values_to = "value")

# Generate timeindex
unemployment$time_index <- yearmonth(paste(unemployment$year, unemployment$month))

unemployment <- as_tsibble(unemployment[3:5], key = name, index = time_index)

head(unemployment)

## # A tsibble: 6 x 3 [1M]
## # Key:      name [1]
##   name    value time_index
##   <chr>  <dbl>    <mth>
## 1 female  4.1     2000 Jan
## 2 female  4.1     2000 Feb
## 3 female  4.3     2000 Mar
## 4 female  4       2000 Apr
## 5 female  4.2     2000 May
## 6 female  4.1     2000 Jun

```

(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-series data. Produce each of these plots, and comment on what you see.

```

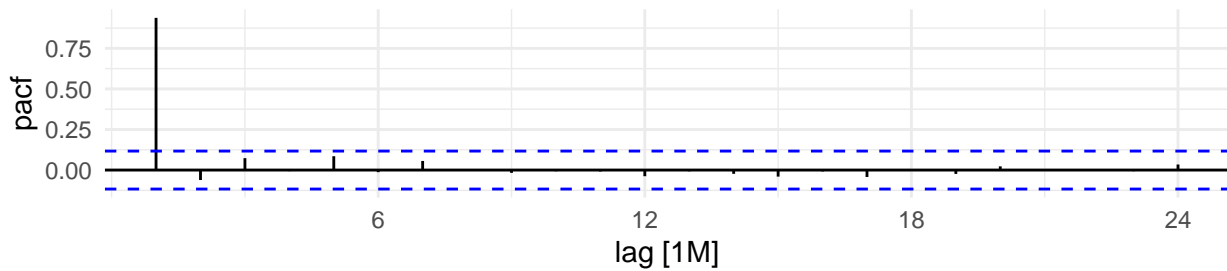
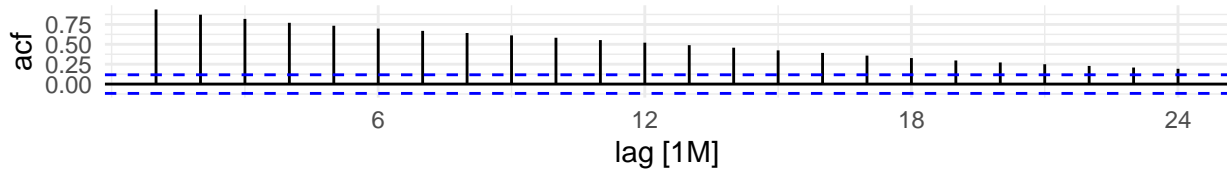
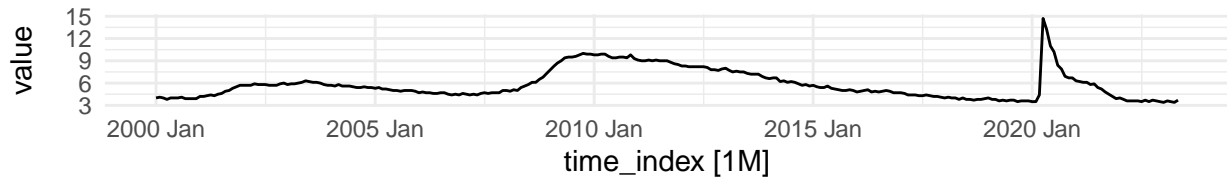
TS_plot <- unemployment %>% filter (name == 'overall') %>% autoplot()

## Plot variable not specified, automatically selected `var = value`
ACF_plot <- unemployment %>% filter(name == 'overall') %>% ACF() %>% autoplot()

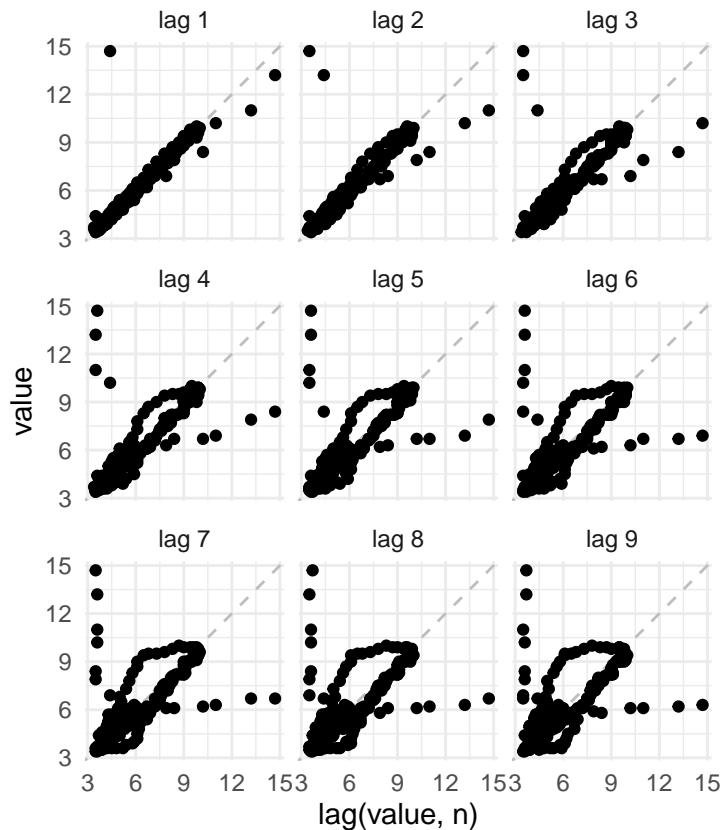
## Response variable not specified, automatically selected `var = value`
PACF_plot <- unemployment %>% filter(name == 'overall') %>% PACF() %>% autoplot()

## Response variable not specified, automatically selected `var = value`
plot_grid(TS_plot, ACF_plot, PACF_plot, align = "v", nrow = 4, rel_heights = c(2, 2, 3), scale = c(1, 1, 1, 1))

```



```
unemployment %>% filter(name == 'overall') %>% gg_lag(value, lag = 1:9, geom = "point", color = 1)
```



I assume that the conclusions about overall unemployment can be extrapolated to gender-specific data, therefore I focus on overall unemployment here. From the look of the series it is obvious that expectation of the series is not constant, i.e. the series is not stationary. This is also reflected on the ACF plot that has many non-zero lags and only gradual decrease in significance. That might mean an AR component in the series, but in this case is more likely is due to strong trend. PACF

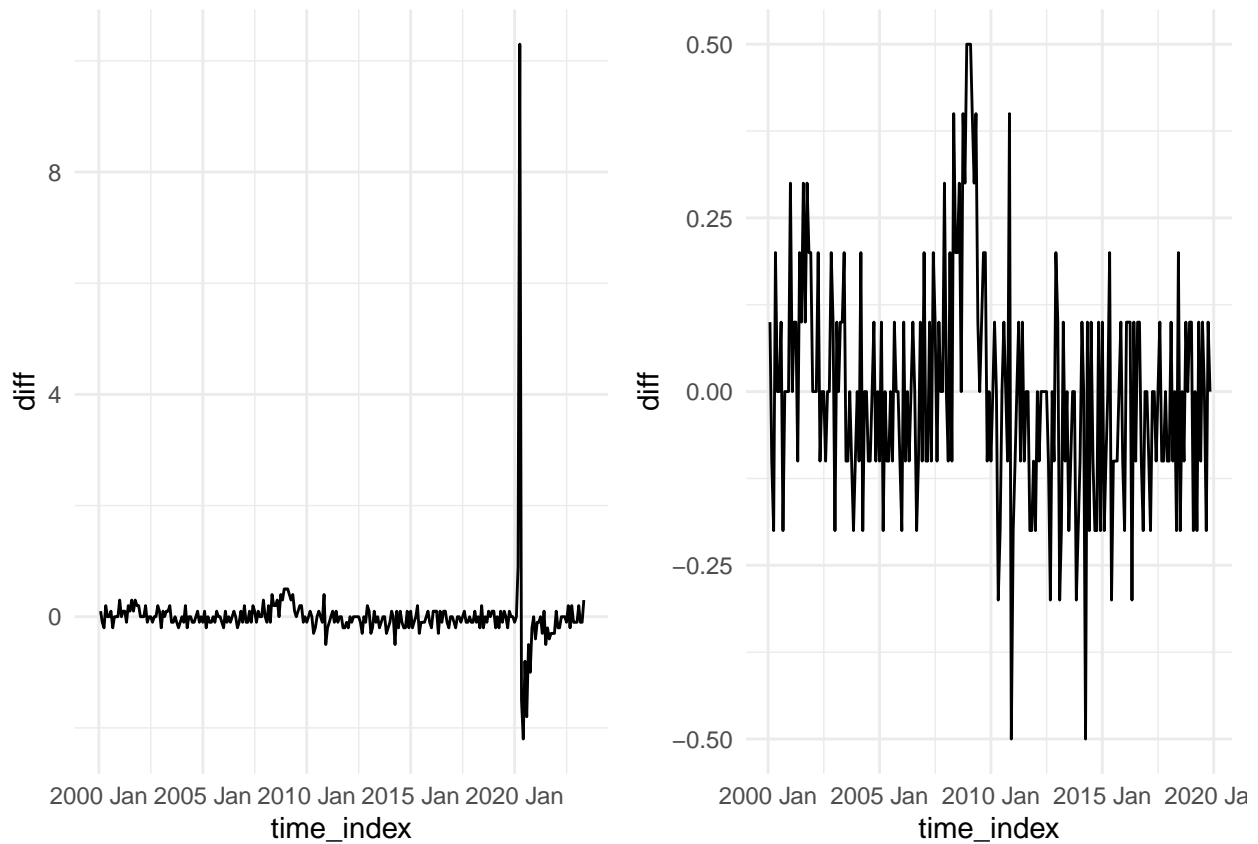
plot also exhibit strong correlation with lag 1, but no oscillatory behaviour typical for MA model.

```
overall_ts <- unemployment %>% filter(name == "overall")
overall_ts <- mutate(overall_ts, diff = difference(value, lag=1))
overall_ts <- overall_ts[3:4]

short <- overall_ts %>% filter(time_index < ym('2019 Dec')) %>%
  ggplot() + aes(x = time_index, y = diff) + geom_line()

full <- overall_ts %>% ggplot() + aes(x = time_index, y = diff) + geom_line()

grid.arrange(full,short,nrow=1)
```

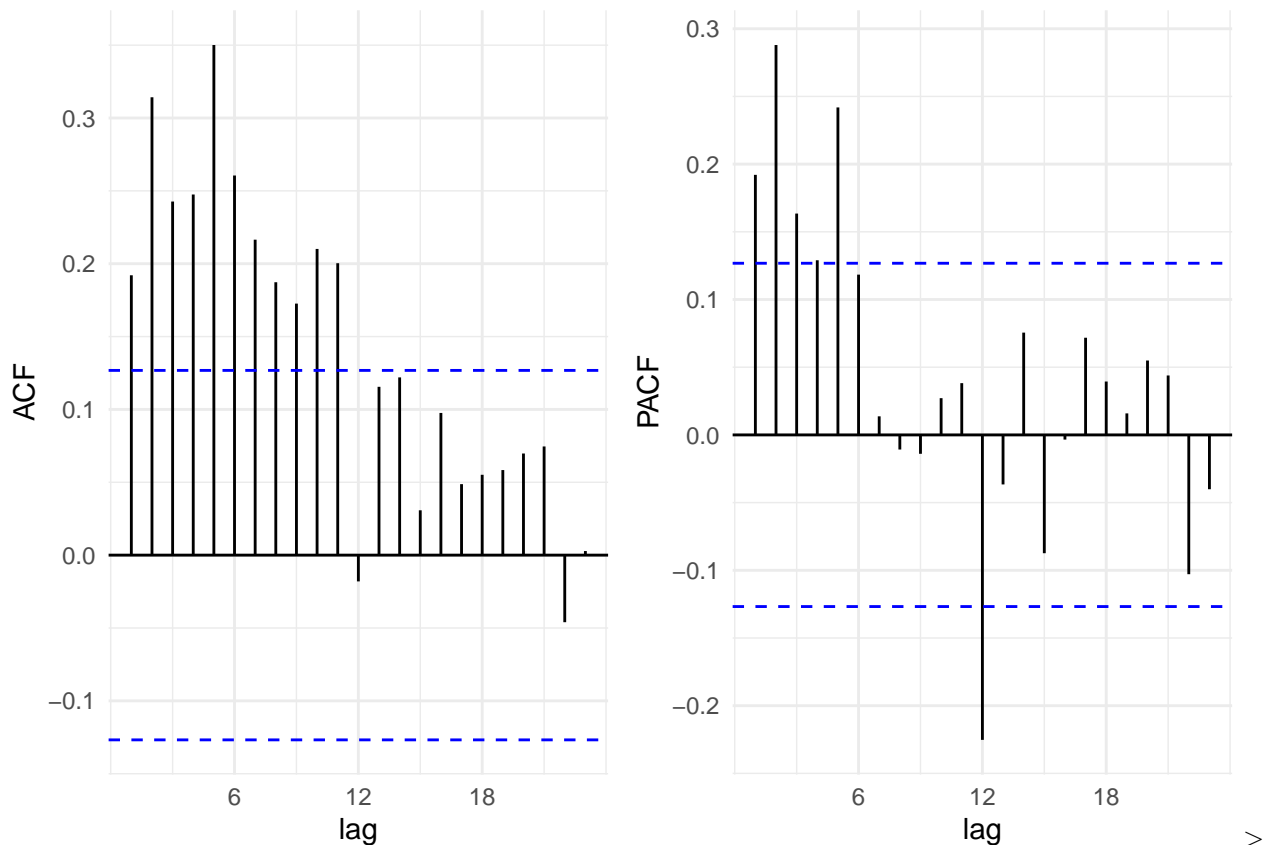


Differencing yields an almost stationary series, but a huge spike at the beginning of the 2020 introduces perturbation that is not characteristic of the series and can skew predictions.

```
g1<-overall_ts %>% filter(time_index < ym('2019 Dec')) %>%
  ACF(diff, type="correlation") %>% autoplot() + labs (x = "lag", y = "ACF")

g2<-overall_ts %>% filter(time_index < ym('2019 Dec')) %>%
  ACF(diff, type="partial") %>% autoplot() + labs (x = "lag", y = "PACF")

grid.arrange(g1,g2,nrow=1)
```



Focusing on detrended series, we can see that ACF plot decays quickly, losing significance after lag 11. This is not consistent with a pure AR process that should show exponential decay in ACF plot. Therefore, based on ACF plot, we might conclude MA(11) model. At the same time, PACF plot is consistently significant up to lag 6, with a negative spike at lag 12. This is not consistent with a pure MA process that should have oscillations in PACF plot and based on ACF plot we might conclude AR(6), possibly up to AR(12) process.

(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? Form the analysis above it is difficult to be completely certain what degrees of AR and MA would be the most appropriate. It is likely to be a mix of the two, and detrending as also necessary. ## (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.

Because diagnostic plots did not yield a definitive indication of the degree of MA and AR

component of the models, we will perform a grid search of various combinations of AR, MA and detrending parameters. We will use BIC criterion as it is more stringent than AIC. We will be fitting a large number of models, therefore stringent criterion is needed to avoid unreliable conclusions.

```
# Generate a set of models AR(1)-AR(10) and MA(1)-MA(10)
# with first and second degree of differencing.
# Select the model with the lowest BIC
model.bic<-unemployment %>% filter(name == "overall") %>%
model(ARIMA(value ~ 1 + pdq(1:10,1:2,1:10) + PDQ(0,0,0), ic="bic", stepwise=F, greedy=F))

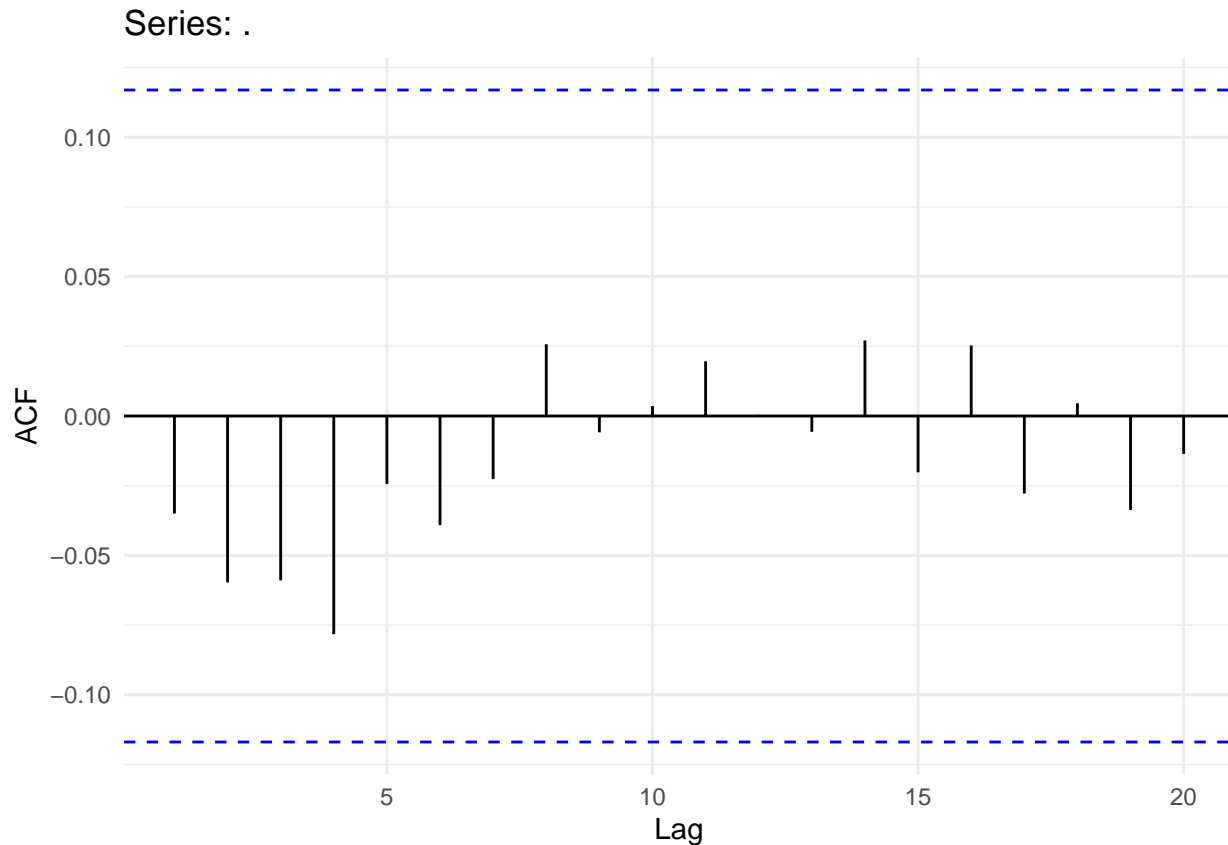
# Report the winning model parameters
model.bic %>% report()
```

```
## Series: value
## Model: ARIMA(1,1,1) w/ drift
##
## Coefficients:
##          ar1      ma1  constant
##      -0.7277  0.8019  -0.0018
## s.e.   0.1594  0.1363   0.0720
##
## sigma^2 estimated as 0.4521:  log likelihood=-284.66
## AIC=577.33  AICc=577.47  BIC=591.87
```

With the winning ARIMA model (1,1,1) in hand we will check its residuals to make sure that the residuals are as close to white noise as possible.

```
# Obtain the winning model as a separate instance of Arima class
dat <- unemployment %>% filter(name == "overall")
arima.model <- arima(dat$value, order = c(1,1,1), include.mean = T)

# Visualize ACF plot of its residuals
arima.model %>% residuals() %>% acf(plot = FALSE) %>% autoplot() + coord_cartesian(xlim = c(1, 20))
```



The residuals do not appear to be a white noise, but we will use Ljung Box test to investigate if they are indeed not random.

```
resid.ts <- as.ts(arima.model %>% residuals())
Box.test(resid.ts, lag = 1, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: resid.ts
## X-squared = 0.34698, df = 1, p-value = 0.5558
```

The test fails to reject the null hypothesis, therefore we do not have statistical evidence against random distribution of the residuals of this model.

The selected model, therefore, is ARIMA (1,1,1) with coefficients autoregressive coefficient -0.7277, moving average coefficient and a constant term (average) of -0.0018. I.e. after one level of differencing, in the resulting autoregressive model the predicted value will depend on the lag 1 series with coeff -0.7277, lag 1 error with coeff 0.8019, and the average of the model is almost zero.

```
# Obtain residuals ACF plot for selected model
dat <- unemployment %>% filter(name == "overall")
arima1_1_1 <- arima(dat$value, order = c(1,1,1), include.mean = T)
plot111 <- arima1_1_1 %>% residuals() %>% acf(plot = FALSE) %>% autoplot() +
  coord_cartesian(xlim = c(1, 15)) + labs(x = "lag", y = "ACF", title = "1.1.1")
```

```
# Obtain residuals plot for alternative ARIMA(1,1,2)
dat <- unemployment %>% filter(name == "overall")
```

```

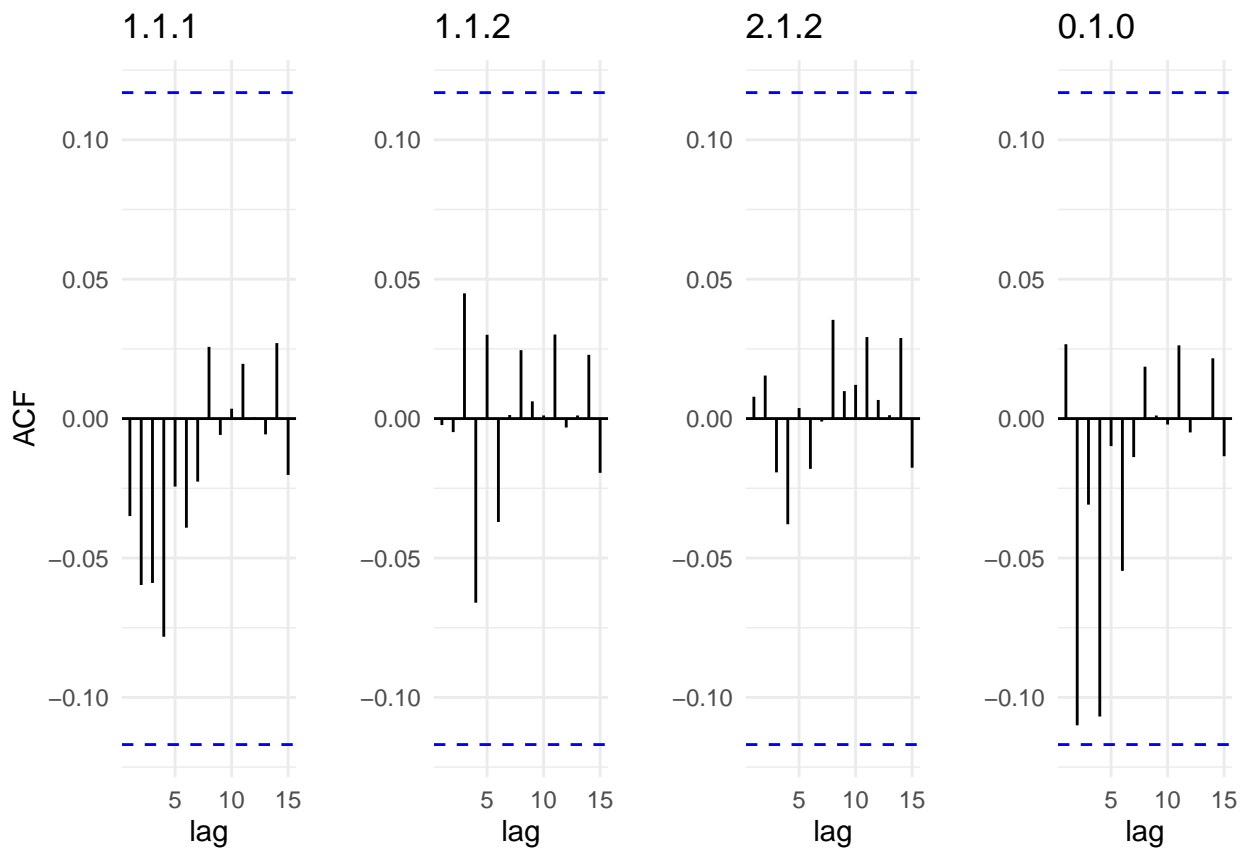
arima1_1_2 <- arima(dat$value, order = c(1,1,2), include.mean = T)
plot112 <- arima1_1_2 %>% residuals() %>% acf(plot = FALSE) %>% autoplot() +
  coord_cartesian(xlim = c(1, 15)) + labs(x = "lag", y = "", title = "1.1.2")

# Obtain residuals plot for for alternative ARIMA(2,1,2)
dat <- unemployment %>% filter(name == "overall")
arima2_1_2 <- arima(dat$value, order = c(2,1,2), include.mean = T)
plot212 <- arima2_1_2 %>% residuals() %>% acf(plot = FALSE) %>% autoplot() +
  coord_cartesian(xlim = c(1, 15)) + labs(x = "lag", y = "", title = "2.1.2")

# Obtain residuals plot for random walk
dat <- unemployment %>% filter(name == "overall")
arima0_1_0 <- arima(dat$value, order = c(0,1,0), include.mean = T)
plot010 <- arima0_1_0 %>% residuals() %>% acf(plot = FALSE) %>% autoplot() +
  coord_cartesian(xlim = c(1, 15)) + labs(x = "lag", y = "", title = "0.1.0")

grid.arrange(plot111, plot112, plot212, plot010, nrow=1)

```



Interestingly, simple random walk model ARIMA (0,1,0) has lower BIC than the selected model (583.87 vs 591.87), but shows clear autoregressive features in its residuals (see plot above). Also, ARIMA (1,1,2) (BIC 593.54) visually has residuals closer to white noise and ARIMA (2,1,2) ACF plot has no significant lags at all, albeit at the expense of higher BIC. More detailed evaluations and testing of predictions would be needed to make the final judgement on the suitability of these models.

(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.

```
covid_df <- read.socrata("https://data.cdc.gov/resource/pwn4-m3yp.json")

columns_to_select <- c(1,2,5,6,7,8)
covid_df$date_updated <- ymd(covid_df$date_updated)
covid_df <- covid_df %>% mutate_at(c(5,6,7,8), as.numeric)
covid_ts <- covid_df[columns_to_select] %>% as_tsibble(key = 'state', index = 'date_updated')
```

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

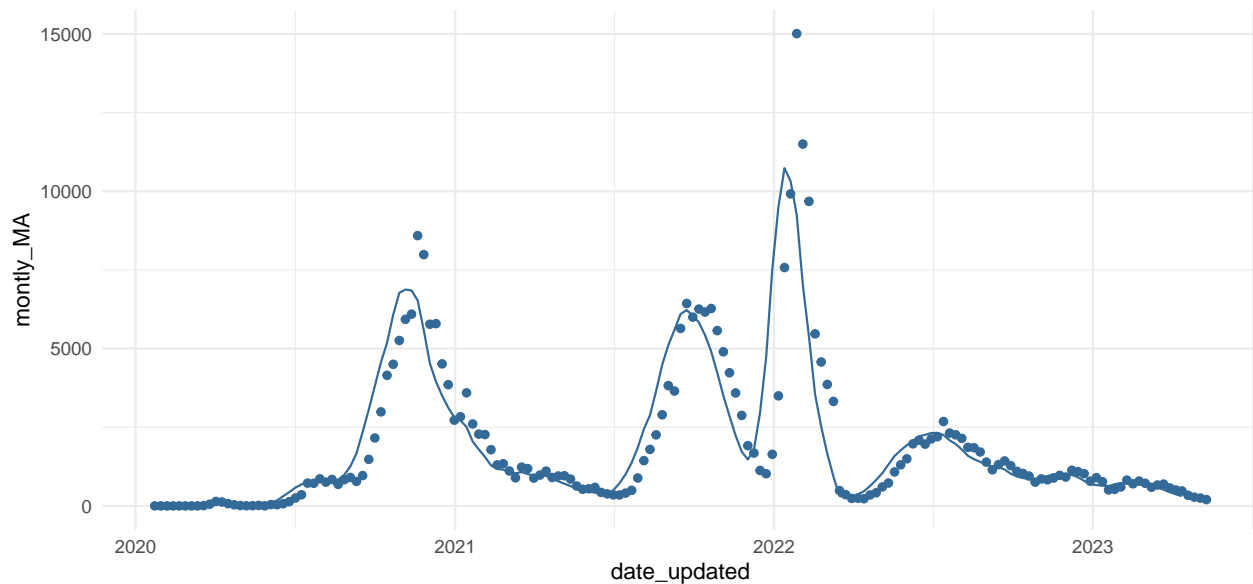
1. Choose a state that is not California (we are putting this criteria in so that we see many different states chosen);
2. Produce a 7-day, backward smoother of the total case rate; then,
3. Produce a model of COVID cases in that state. This should include:
 - Conducting a full EDA and description of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

```
# Let's pick Montana, home of the Glacier NP
covid_MT <- covid_ts %>% filter(state == "MT")

# The data that we ended up using is already aggregated by week
# Therefore weekly moving average is pointless
# Smoothing over 4-week periods (not exactly monthly, but close enough):

# Produce new column with smoothed values
covid_MT <- covid_MT %>% mutate(
  `montly_MA` = slider::slide_dbl(new_cases, mean,
    .before = 0, .after = 4, .complete = TRUE))

# Generate a plot to show smoothing effect
covid_MT %>% ggplot() +
  geom_line(aes(x = date_updated, y = montly_MA , colour = origin)) +
  geom_point(aes(x = date_updated, y = new_cases , colour = origin)) +
  theme_minimal() +
  theme(legend.position = "none")
```



Smoothing effect does not appear to add any value to the analysis, therefore in the following analysis we will use unsmoothed weekly data, equivalent to what we would have, had we gotten the daily data from the source.

We first conduct EDA on the timeseries

```
# Create a differenced version of the Montana TS.
# covid_MT <- mutate(covid_MT, diff = difference(new_cases, lag=26))

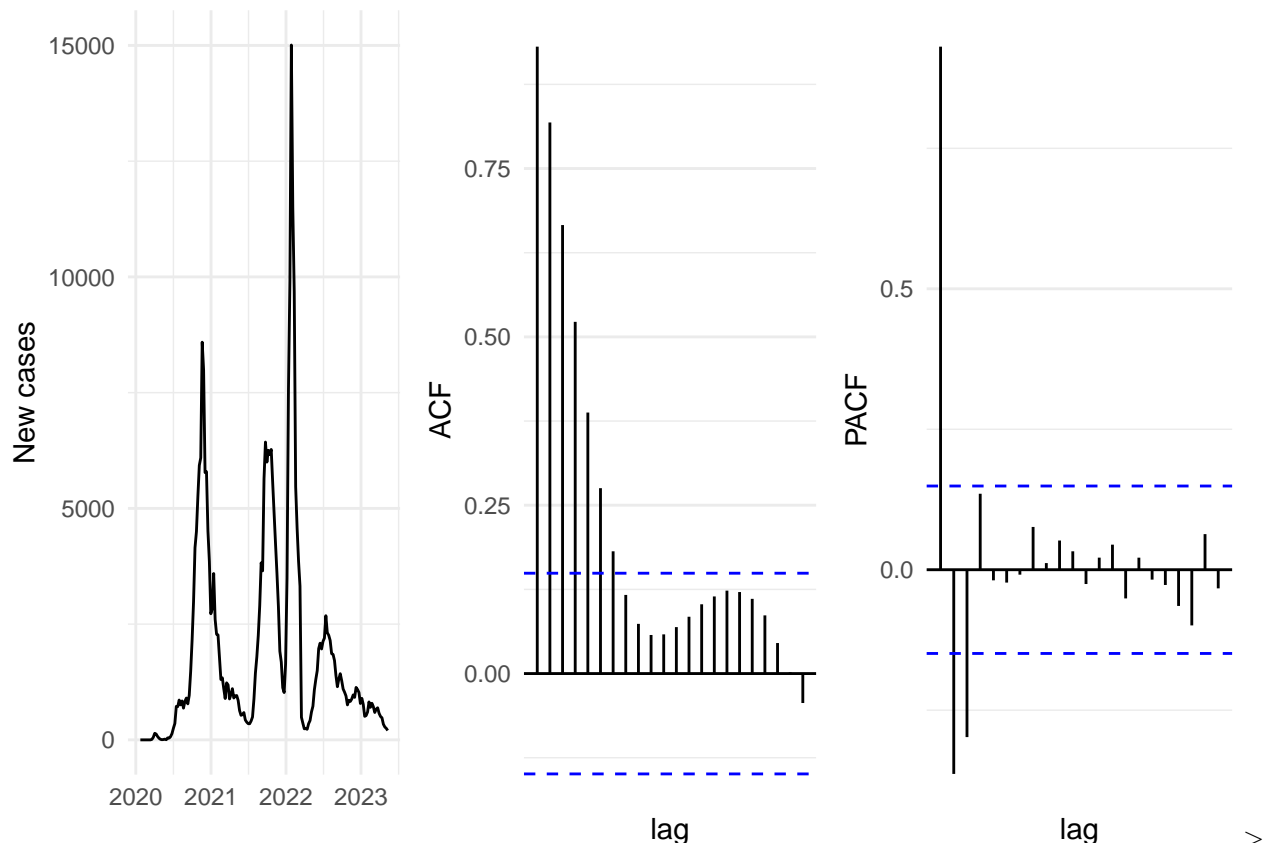
#Generate TS plot
ts_plot <- covid_MT %>% ggplot() + aes(x = date_updated, y = new_cases) +
  geom_line() + labs (x = "", y = "New cases")

#Generate differenced TS plot
ts_diff_plot <- covid_MT %>% ggplot() + aes(x = date_updated, y = diff) +
  geom_line() + labs (x = "", y = "1 week diff")

# Create an ACF plot for the differenced timeseries
acf_plot<-covid_MT %>%
  ACF(new_cases, type="correlation") %>% autoplot() + labs (x = "lag", y = "ACF")

# Create an PACF plot for the differenced timeseries
pacf_plot<-covid_MT %>%
  ACF(new_cases, type="partial") %>% autoplot() + labs (x = "lag", y = "PACF")

grid.arrange(ts_plot , acf_plot, pacf_plot, nrow=1)
```



It is clear that the timeseries is not stationary, as mean of the series has lots of spikes and valeys and does not stay constant. Therefore, application of ARIMA family models is questionable, but we will perform a grid search of the best model anyway.

```
model.bic<-covid_MT %>%
model(ARIMA(new_cases~ 1 + pdq(0:10,0:2,0:10) + PDQ(0,0,0), ic="bic", stepwise=FALSE))

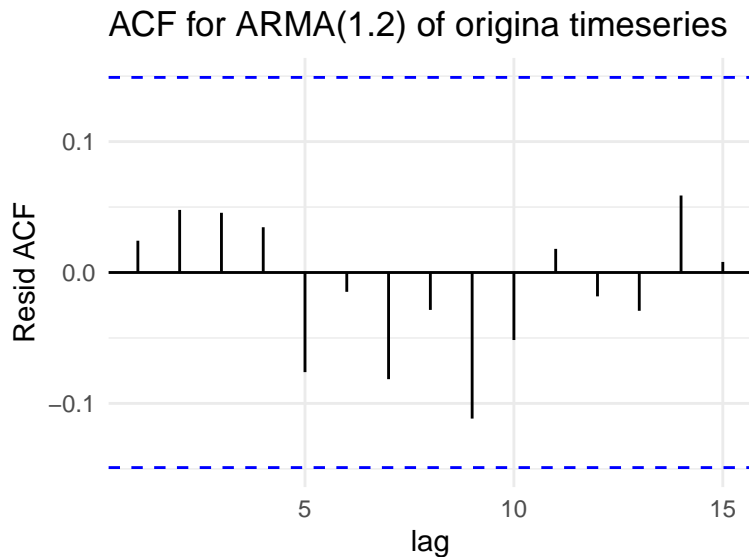
model.bic %>% report()
```

```
## Series: new_cases
## Model: ARIMA(1,0,2) w/ mean
##
## Coefficients:
##          ar1      ma1      ma2  constant
##          0.8502  0.3580  0.4220  270.2374
## s.e.    0.0430  0.0756  0.0674   97.5784
##
## sigma^2 estimated as 569559:  log likelihood=-1391.15
## AIC=2792.3   AICc=2792.66   BIC=2808.07
```

Grid search finds the lowes BIC value for ARMA(1.2) model with BIC 2808.0679916 Next we will examin the residuals of the resultign ARMA model

```
# Recreate model to hgave access to residuals
MT_original <- arima(covid_MT$new_cases,
                     order = c(1,0,2),
                     seasonal = c(0, 0, 0),
                     include.mean = T)
```

```
#Make a plot of residuals ACF
MT_original_plot <- MT_original %>% residuals() %>%
  acf(plot = FALSE, na.action = na.pass) %>%
  autoplot() + coord_cartesian(xlim = c(1, 15)) +
  labs(x = "lag", y = "Resid ACF", title = "ACF for ARMA(1.2) of origina timeseries")
```



```
resid.ar3 <- as.ts(MT_original %>% residuals())
LBT <- Box.test(resid.ar3, lag = 1, type = "Ljung-Box")
```

The residuals look like random white noise. Ljung-Box test fails to reject hypothesis that the residuals are random (p value 0.747). Despite reasonably good fit with ARMA(1,2), we will investigate a detrending in attempt to produce a more stationary timeseries.

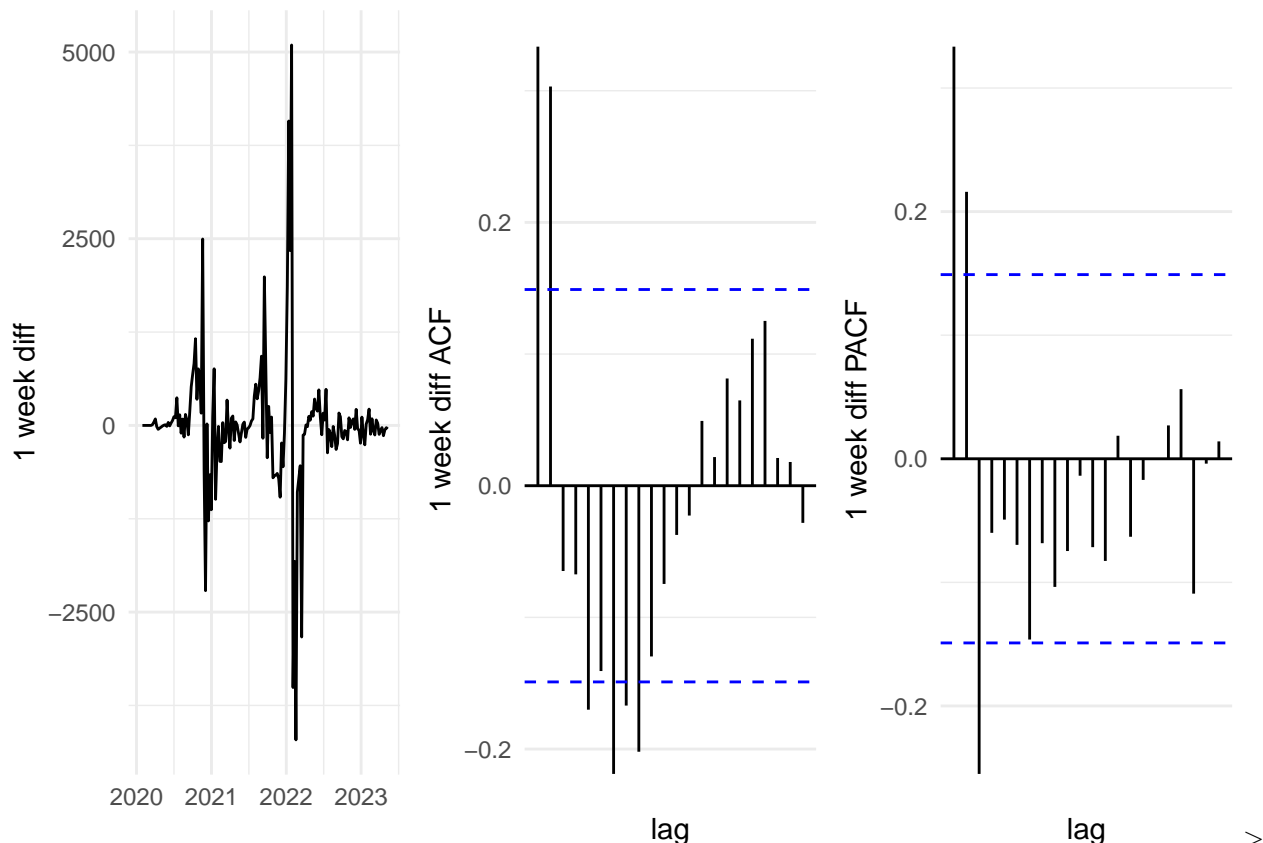
```
# Create a differenced version of the Montana TS.
covid_MT <- mutate(covid_MT, diff = difference(new_cases, lag=1))

#Generate differenced TS plot
ts_diff_plot <- covid_MT %>% ggplot() + aes(x = date_updated, y = diff) +
  geom_line() + labs (x = "", y = "1 week diff")

# Create an ACF plot for the differenced timeseries
acf_plot<-covid_MT %>%
  ACF(diff, type="correlation") %>% autoplot() + labs (x = "lag", y = "1 week diff ACF")

# Create an PACF plot for the differenced timeseries
pacf_plot<-covid_MT %>%
  ACF(diff, type="partial") %>% autoplot() + labs (x = "lag", y = "1 week diff PACF")

grid.arrange(ts_diff_plot, acf_plot, pacf_plot, nrow=1)
```



The mean of the differenced series is now close to 0, but the variance is clearly not constant throughout the range. Additionally, ACF plot has strong oscillations, indicating seasonality. Seasonality is unlikely to be caused by the actual astronomical season, but probably by an interaction of virus spread, consequent political decisions to shut downs that in turn causes new wave of virus spread. After a few experiments, we found that de-seasoning by differencing with lag 26 provides a more stationary timeseries

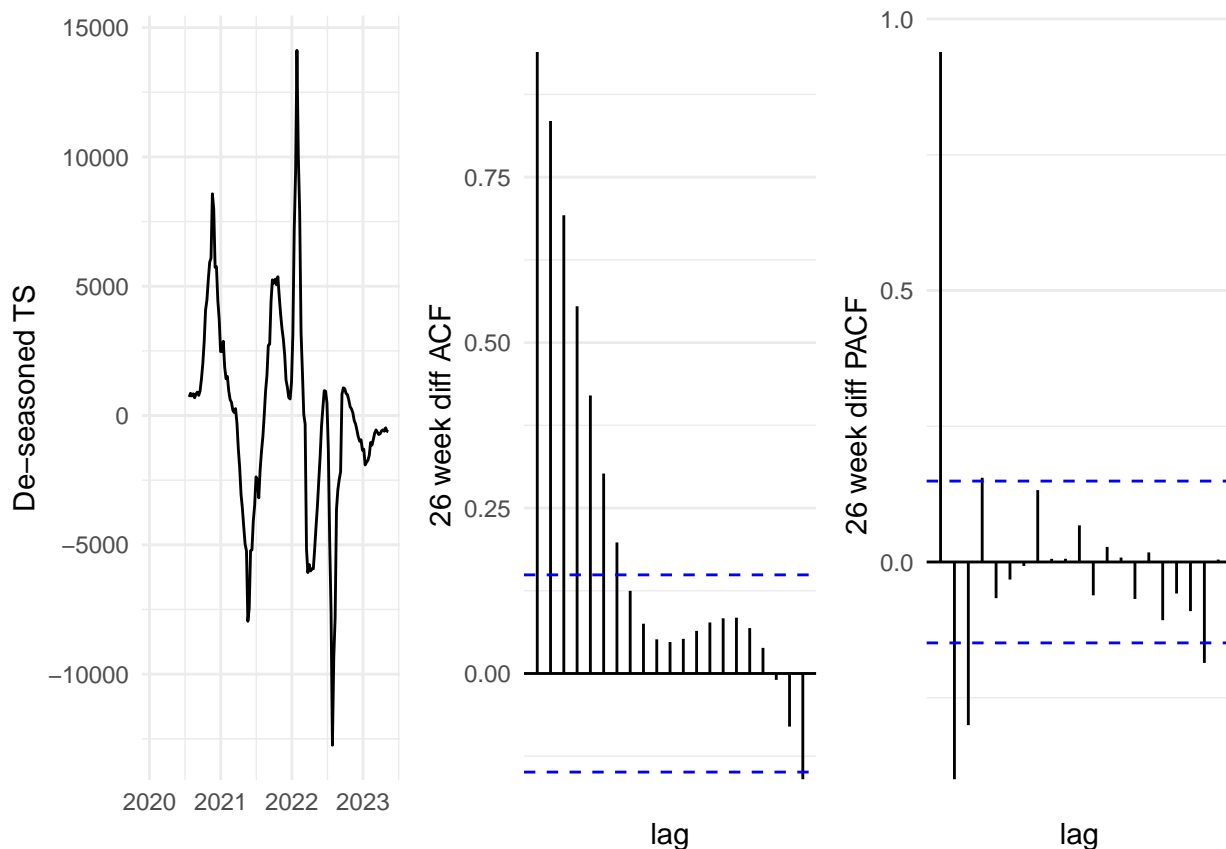
```
# Create a differenced version of the Montana TS.
covid_MT <- mutate(covid_MT, diff = difference(new_cases, lag=26))

#Generate differenced TS plot
ts_diff_plot <- covid_MT %>% ggplot() + aes(x = date_updated, y = diff) +
  geom_line() + labs (x = "", y = "De-seasoned TS")

# Create an ACF plot for the differenced timeseries
acf_plot<-covid_MT %>%
  ACF(diff, type="correlation") %>% autoplot() + labs (x = "lag", y = "26 week diff ACF")

# Create an PACF plot for the differenced timeseries
pacf_plot<-covid_MT %>%
  ACF(diff, type="partial") %>% autoplot() + labs (x = "lag", y = "26 week diff PACF")

grid.arrange(ts_diff_plot, acf_plot, pacf_plot, nrow=1)
```



After de-seasoning, the time series becomes more stationary, with mean around 0 and standard deviation without much change. Characteristic decay of ACF function indicates an AR process, which is consistent with quick decay of PACF plot. From these graphs it appears that it might be an AR(3) process, because 3'd lag is the last one strongly significant on PACF plot. To confirm these observations we will perform a grid search using BIC to evaluate goodness of fit.

```
MT_deseasoned <- covid_MT %>%
model(ARIMA(diff~ 1 + pdq(0:10,0:2,0:10) + PDQ(0,0,0), ic="bic", stepwise=FALSE))

MT_deseasoned %>% report()

## Series: diff
## Model: ARIMA(3,0,0) w/ mean
##
## Coefficients:
##          ar1          ar2          ar3  constant
##          1.1874   -0.0040   -0.2958    11.1739
## s.e.    0.0779    0.1254    0.0778    94.9349
##
## sigma^2 estimated as 1193134:  log likelihood=-1248.61
## AIC=2507.22  AICc=2507.58  BIC=2522.99
```

Grid search finds the lowest BIC value for AR(3) model with BIC 2523, a dramatic improvement over best ARMA (1,2) model for original TS (BIC 2808.1) or simply differenced timeseries (BIC 2800.9)

Next we will examine the residuals of the resulting AR model

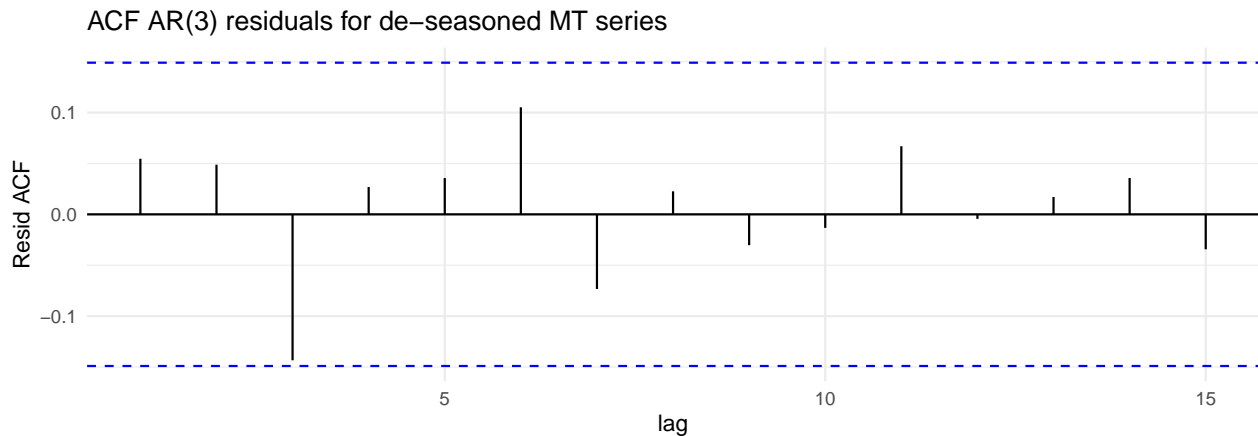
```

# Fit the model one more time to have access to residuals
MT_AR3 <- arima(covid_MT$diff, order = c(3,0,0), seasonal = c(0, 0, 0), include.mean = T)

# Plot ACF for residuals
MT_AR3_plot <- MT_AR3 %>% residuals() %>% acf(plot = FALSE, na.action = na.pass) %>% autoplot() +
  coord_cartesian(xlim = c(1, 15)) + labs(x = "lag", y = "Resid ACF", title = "ACF AR(3) residuals for c

#Perform LB test on residuals
resid.mtar3 <- as.ts(MT_AR3 %>% residuals())
LBT_deseasoned <- Box.test(resid.mtar3, lag = 1, type = "Ljung-Box")

```



The residuals resemble white noise. Ljung-Box test fails to reject hypothesis that the residuals are random (p value 0.504). Therefore we believe that the best model is a simple AR(3) after deseasoning of the timeseries with lag 26.

(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

```

# I could not figure out how to aggregate on a tsibble
# So will aggregate on dataframe

# Subset the original df to make final ts smaller
columns_to_select <- c(1,6)
nation_df <- covid_df[columns_to_select]

# Aggregate on the dataframe
nation_ts <- aggregate(nation_df["new_cases"], by = nation_df["date_updated"], sum)

# Convert back to tsibble
nation_ts <- as_tsibble(nation_ts, index = date_updated)

# The data that we ended up using is already aggregated by week
# Therefore weekly moving average is pointless
# Smoothing over 4-week periods (not exactly monthly, but close enough):

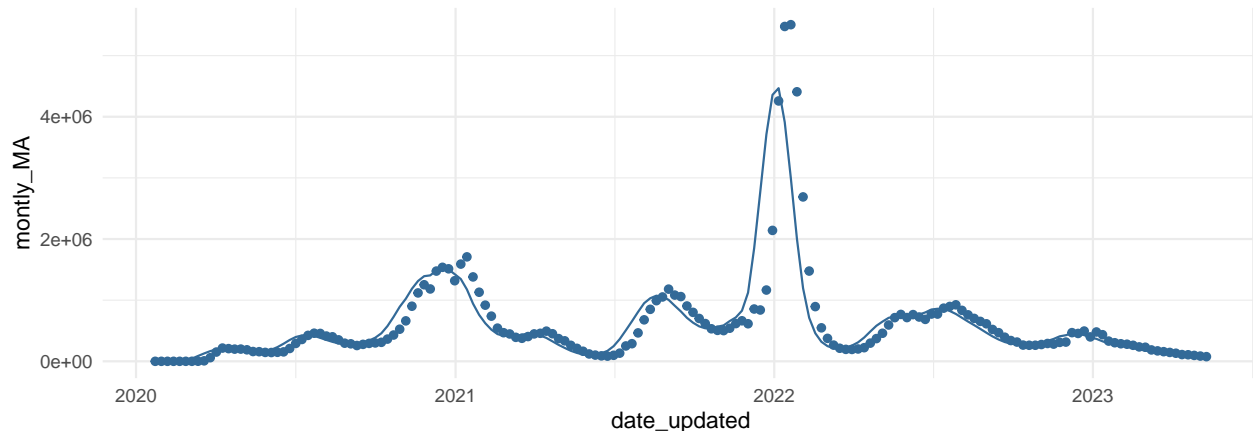
```

```

# Produce new column with smoothed values
nation_ts <- nation_ts%>% mutate(
  `montly_MA` = slider::slide_dbl(new_cases, mean,
    .before = 0, .after = 4, .complete = TRUE))

# Generate a plot to show smoothing effect
nation_ts %>% ggplot() +
  geom_line(aes(x = date_updated, y = montly_MA , colour = origin)) +
  geom_point(aes(x = date_updated, y = new_cases , colour = origin)) +
  theme_minimal() + theme(legend.position = "none")

```



Smoothing effect does not appear to add any value to the analysis, therefore in the following analysis we will use unsmoothed weekly data, equivalent to what we would have, had we gotten the daily data form the source.

We will follow the same pattern on nationwide analysis as we developed for Montana case. Below we plot the series itself, along side with the differenced series, ACF and PACF of the differenced series

```

nation_ts<- mutate(nation_ts, diff = difference(new_cases, lag=1))

ts_plot <- nation_ts %>% ggplot() + aes(x = date_updated, y = new_cases) +
  geom_line() + labs (x = "", y = "New cases")

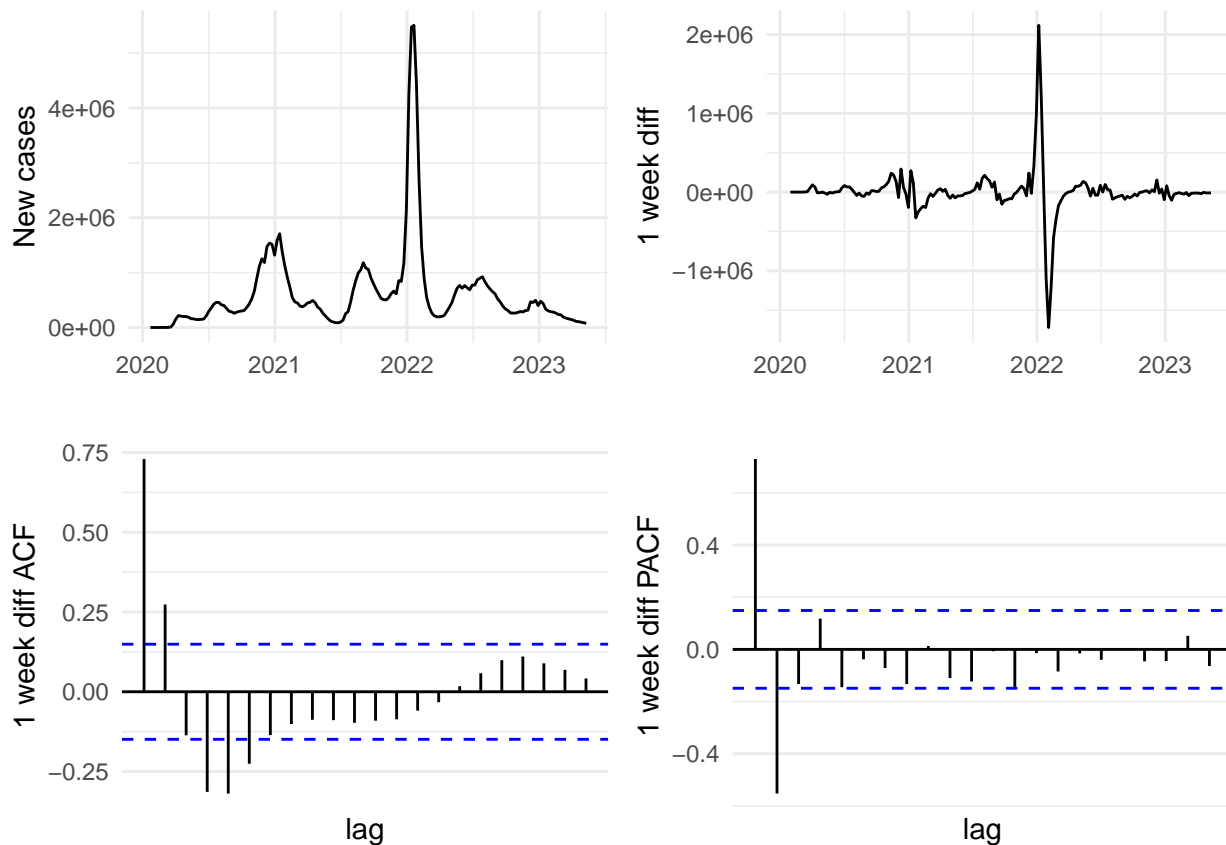
ts_diff_plot <- nation_ts %>% ggplot() + aes(x = date_updated, y = diff) +
  geom_line() + labs (x = "", y = "1 week diff")

acf_plot<-nation_ts %>%
  ACF(diff, type="correlation") %>% autoplot() + labs (x = "lag", y = "1 week diff ACF")

pacf_plot<-nation_ts %>%
  ACF(diff, type="partial") %>% autoplot() + labs (x = "lag", y = "1 week diff PACF")

grid.arrange(ts_plot , ts_diff_plot, acf_plot, pacf_plot, nrow=2)

```

Even after differencing the series remains non-stationary, with variance strongly dependent on time. Additionally, ACF resembles dampened oscillations, indicating underlying seasonality. It was not clear what would cause the seasonality, but running a few experiments we can find that de-trending with lag around half a year significantly reduces these artifacts. This comes as no surprise after findings in Montana data.

```
nation_ts <- mutate(nation_ts, diff = difference(new_cases, lag=30))

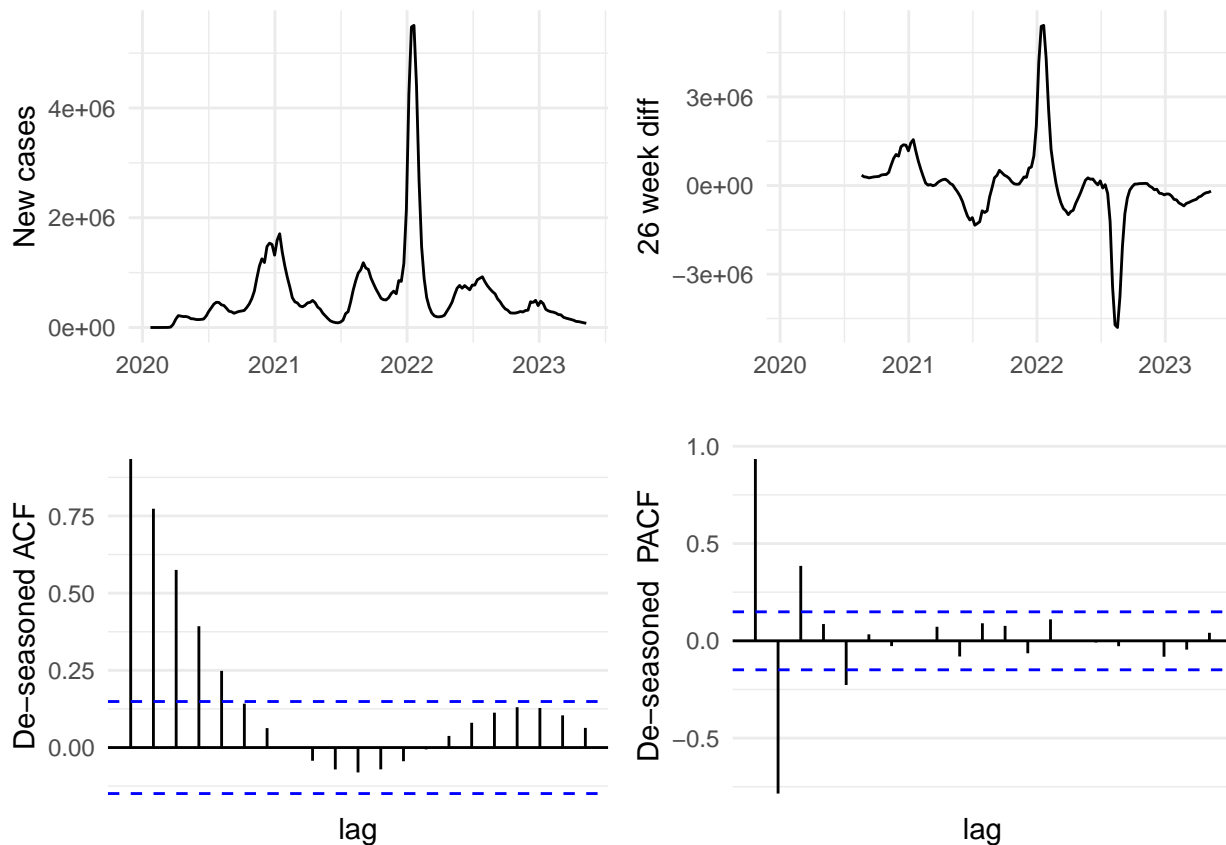
ts_plot <- nation_ts %>% ggplot() + aes(x = date_updated, y = new_cases) +
  geom_line() + labs (x = "", y = "New cases")

ts_diff_plot <- nation_ts %>% ggplot() + aes(x = date_updated, y = diff) +
  geom_line() + labs (x = "", y = "26 week diff")

acf_plot <- nation_ts %>%
  ACF(diff, type="correlation") %>% autoplot() + labs (x = "lag", y = "De-seasoned ACF")

pacf_plot <- nation_ts %>%
  ACF(diff, type="partial") %>% autoplot() + labs (x = "lag", y = "De-seasoned PACF")

grid.arrange(ts_plot , ts_diff_plot, acf_plot, pacf_plot, nrow=2)
```



Experiments on the aggregated series showed that for nationwide data lag 30 is more suitable. After de-seasoning, the timeseries appears to be closer to stationary, with mean around 0 and standard deviation relatively stable. Characteristic decay of ACF function indicates an AR process, which is consistent with quick decay of PACF plot. From these graphs it appears that it might be an AR(5) process, because 5'th lag is the last one significant on PACF plot. It is worth noting that ACF plot still has significant oscillations. More complex de-trending procedures than just differencing might be required.

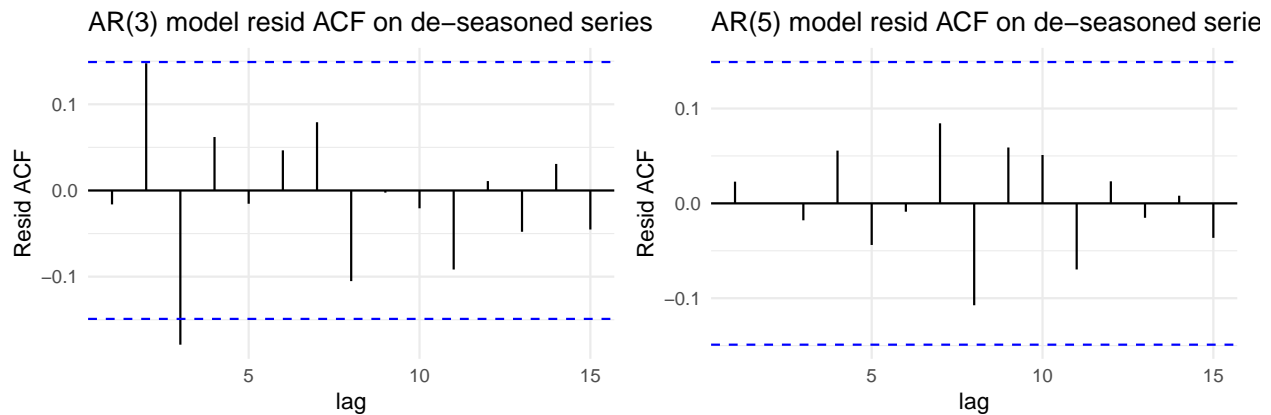
Next, we will perform a grid search using BIC to evaluate goodness of fit.

```
nation.bic<-nation_ts%>%
model(ARIMA(diff~ 1 + pdq(0:10,0:2,0:10) + PDQ(0,0,0), ic="bic", stepwise=F, greedy=F))
nation.bic %>%
report()
```

```
## Series: diff
## Model: ARIMA(3,0,0) w/ mean
##
## Coefficients:
##      ar1      ar2      ar3  constant
##      1.9776 -1.4447  0.3957   1833.972
## s.e.  0.0758   0.1348  0.0755  20514.645
##
## sigma^2 estimated as 5.384e+10:  log likelihood=-1983.81
## AIC=3977.61  AICc=3977.97  BIC=3993.38
```

```
Nation_AR3 <- arima(nation_ts$diff, order = c(3,0,0), seasonal = c(0, 0, 0), include.mean = T)
Nation_AR3_plot <- Nation_AR3 %>% residuals() %>% acf(plot = FALSE, na.action = na.pass) %>% autoplot()
  coord_cartesian(xlim = c(1, 15)) + labs(x = "lag", y = "Resid ACF", title = "AR(3) model resid ACF on")
```

```
Nation_AR5 <- arima(nation_ts$diff, order = c(5,0,0), seasonal = c(0, 0, 0), include.mean = T)
Nation_AR5_plot <- Nation_AR5 %>% residuals() %>% acf(plot = FALSE, na.action = na.pass) %>% autoplot()
coord_cartesian(xlim = c(1, 15)) + labs(x = "lag", y = "Resid ACF", title = "AR(5) model resid ACF on de-seasoned series")
```



> The grid search yielded the lowest BIC for AR(3) model, however, residual ACF plot for this model has a few strongly significant lags. ACF of residuals from AR(5) model that we inferred from the analysis of ACF and PACF of de-seasoned series demonstrates a lot more random behaviour. Therefore we select the AR(5) model after de-seasoning with lag 30 as the final answer.

(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.

Modeling nationwide data as a sum of state-level series almost certainly made the model less relevant. Montana is a reasonably populous state and the state-level model is sufficiently powered, there is little benefit from including all these additional data points. Modelling Guam or other small localities might actually benefit from adding more data, but they matter little for the overall picture.

At the same time, different states have different policies and circumstance, so averaging over this diverse set of data reduces usefulness of the model. The picture of how pandemic progressed through the nation is granular and averaging over the whole nation blurs out this granularity, making predictions less relevant.

In ideal world, we would need to incorporate a lot more external factors into the model. Examples of time-dependent factors include government mandated shutdowns and evolving attitude toward wearing masks. Time-independent factors might include population density that affects rate of transmission, and healthcare standards that affect how fast people recover etc.

Making a (shaky) assumption that these factors are constant within the state, I would estimate

independent models for each state and then used these 50+ time series as inputs into the nationwide model, regressing national new cases on the state-level new cases and state-level shocks. This would allow me to model missing information about those external factors without gathering additional data.