W271 Assignment 8

```
Sys.setlocale("LC_TIME", "English")
```

```
## [1] "English_United States.1252"
```

```
library(tidyverse)
library(magrittr)
library(patchwork)

library(lubridate)

library(tsibble)
library(feasts)
library(fable)
library(forecast)

library(sandwich)
library(lmtest)

library(nycflights13)
library(blsR)
library(reshape2)
```

```
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?

The AR Model relates to the past values of the series, while the MA relates to past white noise (error) terms. Meaning that when forecasting an AR Model we could predict up to n values ahead (where n is the AR order of the model). While for MA models the best prediction we can do is the same value as the last observation.

(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(
  list(overall = 'LNS14000000'),
  start_year = 2000, end_year=2023, tidy=TRUE
)
```

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

```
# creating tsibble object with time_index
unemployment <- unemployment %>%
  mutate(time_index = yearmonth(make_datetime(year, month))) %>%
  as_tsibble(index = time_index)
# ordering columns
unemployment <- unemployment %>%
  select(year, month, time_index, overall)

unm.short <- unemployment[, 3:4]
head(unemployment, 10)
```

```
## # A tsibble: 10 x 4 [1M]
##     year month time_index overall
##    <int> <int>      <mth>   <dbl>
##  1  2000     1   2000 Jan     4
##  2  2000     2   2000 Feb     4.1
##  3  2000     3   2000 Mar     4
##  4  2000     4   2000 Apr     3.8
##  5  2000     5   2000 May     4
##  6  2000     6   2000 Jun     4
##  7  2000     7   2000 Jul     4
##  8  2000     8   2000 Aug     4.1
##  9  2000     9   2000 Sep     3.9
## 10  2000    10   2000 Oct     3.9
```

This time we've easily created the tsibble from the data, we have also creted a tsibble with just two columns for it to be manipulated in the steps to come.

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

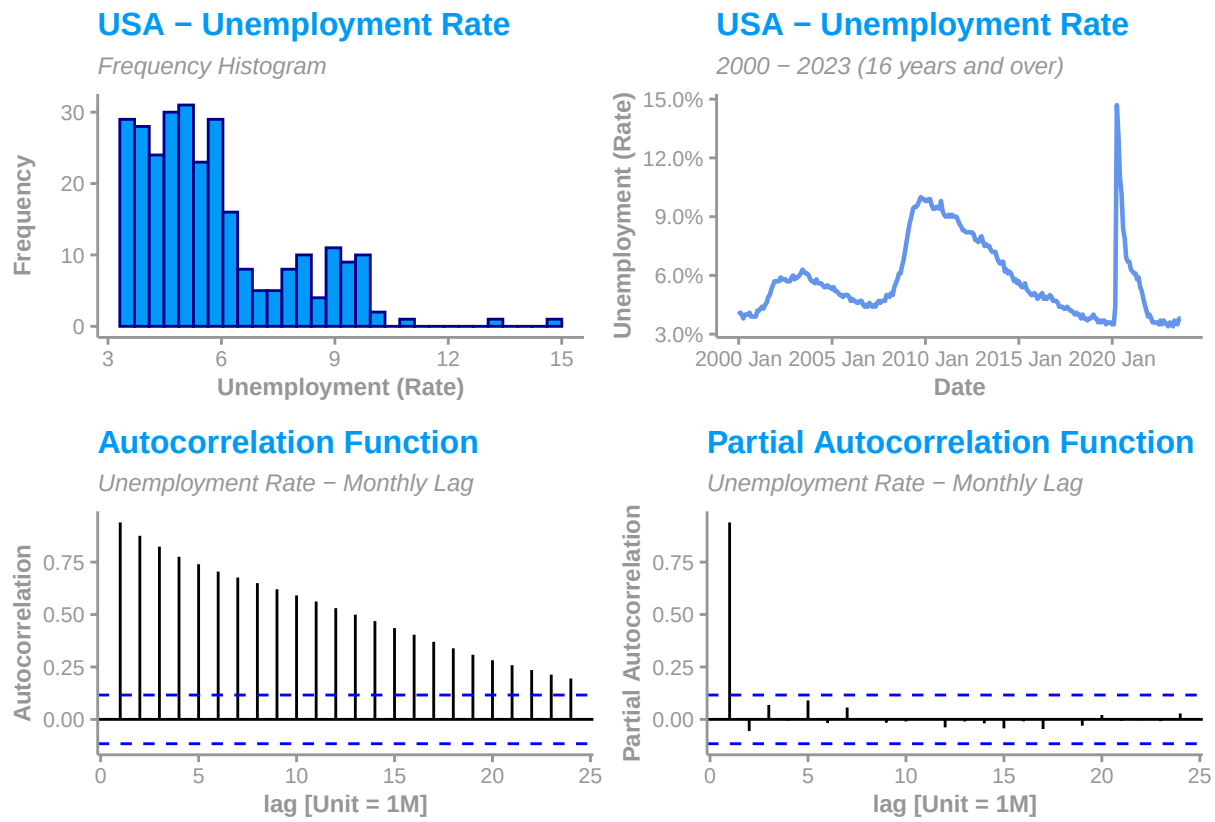
We have presented four **core** plots that are a part of the EDA for time-series data. Produce each of these plots, and comment on what you see.

```r
unm_hist <- ggplot(unemployment,
  aes(
    x = overall
    )) +
    geom_histogram(
      aes(y = ..count..),
      col = I("darkblue"),
      fill = "#0099F8") +
    labs(
      title = "USA - Unemployment Rate",
      subtitle = "Frequency Histogram",
      x = "Unemployment (Rate)",
      y = "Frequency"
    ) +
    theme_classic() +
    theme(
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 9,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 9,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696")
    )

unm_plot <- ggplot(
  unemployment,
  aes(
    x = time_index,
    y = overall / 100
    )
  ) +
  geom_line(
    size = 0.8,
    color = "cornflowerblue"
    ) +
  labs(
    title = "USA - Unemployment Rate",
    subtitle = "2000 - 2023 (16 years and over)",
    x = "Date",
    y = "Unemployment (Rate)"
       ) +
    theme_classic() +
    theme(
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
```

```
                                size = 9,
                                face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 9,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696"),
    ) + scale_y_continuous(labels = scales::percent)

unm_acf <- ACF(
      fill_gaps(unemployment, .full = TRUE),
      overall
    ) %>% autoplot() + labs(
      x = "lag [Unit = 1M]",
      y = "Autocorrelation",
      title = "Autocorrelation Function",
      subtitle = "Unemployment Rate - Monthly Lag"
    ) +
    theme_classic() +
    theme(
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 9,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 9,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696"),
    ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))

unm_pacf <- PACF(
      fill_gaps(unemployment, .full = TRUE),
      overall
    ) %>% autoplot() + labs(
      x = "lag [Unit = 1M]",
      y = "Partial Autocorrelation",
      title = "Partial Autocorrelation Function",
      subtitle = "Unemployment Rate - Monthly Lag"
    ) +
    theme_classic() +
    theme(
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 9,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
```

```
                                size = 9,
                                face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696"),
    ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))

(unm_hist | unm_plot) /(unm_acf | unm_pacf)
```

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



By looking at the histogram we notice that it's right skewed, meaning that the lower values have higher frequencies. Looking at the series plot it seems like we might have a non-stationary series. Finally, it seems that the ACF is slowly decaying while the PCF drops dramatically after lag 1.

(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?

We wrote a couple of things from the ACF and PACF which are signs that **we could have an AR(1) series**. Although as we said, there also seems to be some signs of non-stationarity (like certain trend and the variance apparently increases as time passes), so a pure AR(1) model might not be enough.

(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.

Since we're only analyzing ARMA models, it seems that the **best model would be an AR(1)**, because of the slowly decaying ACF and the abrupt decrease in the PACF (right after lag 1).

So, the model is the following

```
mod.unm.ar1 <- unemployment %>%
model(ARIMA(overall ~ 1 + pdq(1,0,0) , stepwise=F, greedy=F))

mod.unm.ar1 %>%
report()
```

```
## Series: overall
## Model: ARIMA(1,0,0) w/ mean
##
## Coefficients:
##          ar1  constant
##       0.9420    0.3245
## s.e.  0.0193    0.0371
##
## sigma^2 estimated as 0.4365:  log likelihood=-286.34
## AIC=578.69   AICc=578.77   BIC=589.65
```

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

We also have some fluctuation in the PACF that's decaying to zero so perhaps we might have some MA interaction as well. So we will try to choose the best MA interaction with our AR(1) model as follows:

```
# Using AIC
mod.unm.a1mq.a <- unemployment %>%
model(ARIMA(overall ~ 1 + pdq(1,0,1:10), ic="aic", stepwise=F, greedy=F))

mod.unm.a1mq.a %>%
report()
```

```
## Series: overall
## Model: ARIMA(1,0,2) w/ mean
##
## Coefficients:
##          ar1     ma1      ma2  constant
##       0.9478  0.0537  -0.1063    0.2911
## s.e.  0.0212  0.0642   0.0710    0.0349
##
## sigma^2 estimated as 0.4344:  log likelihood=-284.68
## AIC=579.37   AICc=579.58   BIC=597.63
```

```
# Using BIC
mod.unm.a1mq.b <- unemployment %>%
model(ARIMA(overall ~ 1 + pdq(1,0,1:10), ic="bic", stepwise=F, greedy=F))

mod.unm.a1mq.b %>%
report()
```

```
## Series: overall
## Model: ARIMA(1,0,1) w/ mean
##
## Coefficients:
##          ar1     ma1  constant
##       0.9339  0.0720    0.3713
## s.e.  0.0222  0.0687    0.0400
##
## sigma^2 estimated as 0.4363:  log likelihood=-285.8
## AIC=579.6   AICc=579.74   BIC=594.21
```

We notice that forcing the AR(1) model to also have an MA component, we can get different results on what the best model is depending on the criterion we use:

- Using AIC the best model would be an ARMA(1,2)
- Using BIC the best model would be an ARMA(1,1)

Although, both of them have higher AICs and BICs than the pure AR(1) model, so we would still choose the model without the MA component.

We could also argue that the order of the AR model could be higher than 1, although the PACF seems clearly as an AR(1) model, but we can still try to find the best order as follows:

```
# Using AIC
mod.unm.arqmaq.a <- unemployment %>%
model(ARIMA(overall ~ 1 + pdq(0:10,0,0:10), ic="aic", stepwise=F, greedy=F))

mod.unm.arqmaq.a %>%
report()
```

```
## Series: overall
## Model: ARIMA(1,0,0) w/ mean
##
## Coefficients:
##          ar1  constant
##       0.9420    0.3245
## s.e.  0.0193    0.0371
##
## sigma^2 estimated as 0.4365:  log likelihood=-286.34
## AIC=578.69   AICc=578.77   BIC=589.65
```

```
# Using BIC
mod.unm.arqmaq.b <- unemployment %>%
model(ARIMA(overall ~ 1 + pdq(0:10,0,0:10), ic="aic", stepwise=F, greedy=F))

mod.unm.arqmaq.b %>%
report()
```

```
## Series: overall
## Model: ARIMA(1,0,0) w/ mean
##
## Coefficients:
##          ar1  constant
##       0.9420    0.3245
## s.e.  0.0193    0.0371
##
## sigma^2 estimated as 0.4365:  log likelihood=-286.34
## AIC=578.69   AICc=578.77   BIC=589.65
```

And we find that the best model (using both criterions), is the proposed AR(1) model we stated previously

- (2 points) Interpret the coefficients of the model that you have estimated.

Therefore, the chosen model will have the following form:

$$x_t = c + \alpha_1 x_{t-1} + w_t$$

Where c would be the intercept, α_1 is the model parameter for lag 1 and w_t is white noise

```
mod.unm.ar1 %>%
report()
```

```
## Series: overall
## Model: ARIMA(1,0,0) w/ mean
##
## Coefficients:
##          ar1  constant
##       0.9420    0.3245
## s.e.  0.0193    0.0371
##
## sigma^2 estimated as 0.4365:  log likelihood=-286.34
## AIC=578.69   AICc=578.77   BIC=589.65
```

Meaning we have the following model:

$$x_t = 0.3245 + 0.9420 * x_{t-1} + w_t$$

- (2 points) Produce and interpret the model diagnostic plots to evaluate how well your best-fitting model is performing.

We can also run a Ljung Box Test to test if these residuals are randomly distributed:

```
resid.ar1 <- mod.unm.ar1 %>%
  augment() %>%
  select(.resid) %>%
  as.ts()

Box.test(resid.ar1, lag = 1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.ar1
## X-squared = 0.83771, df = 1, p-value = 0.3601
```

```
Box.test(resid.ar1, lag = 2, type = "Ljung-Box")
```

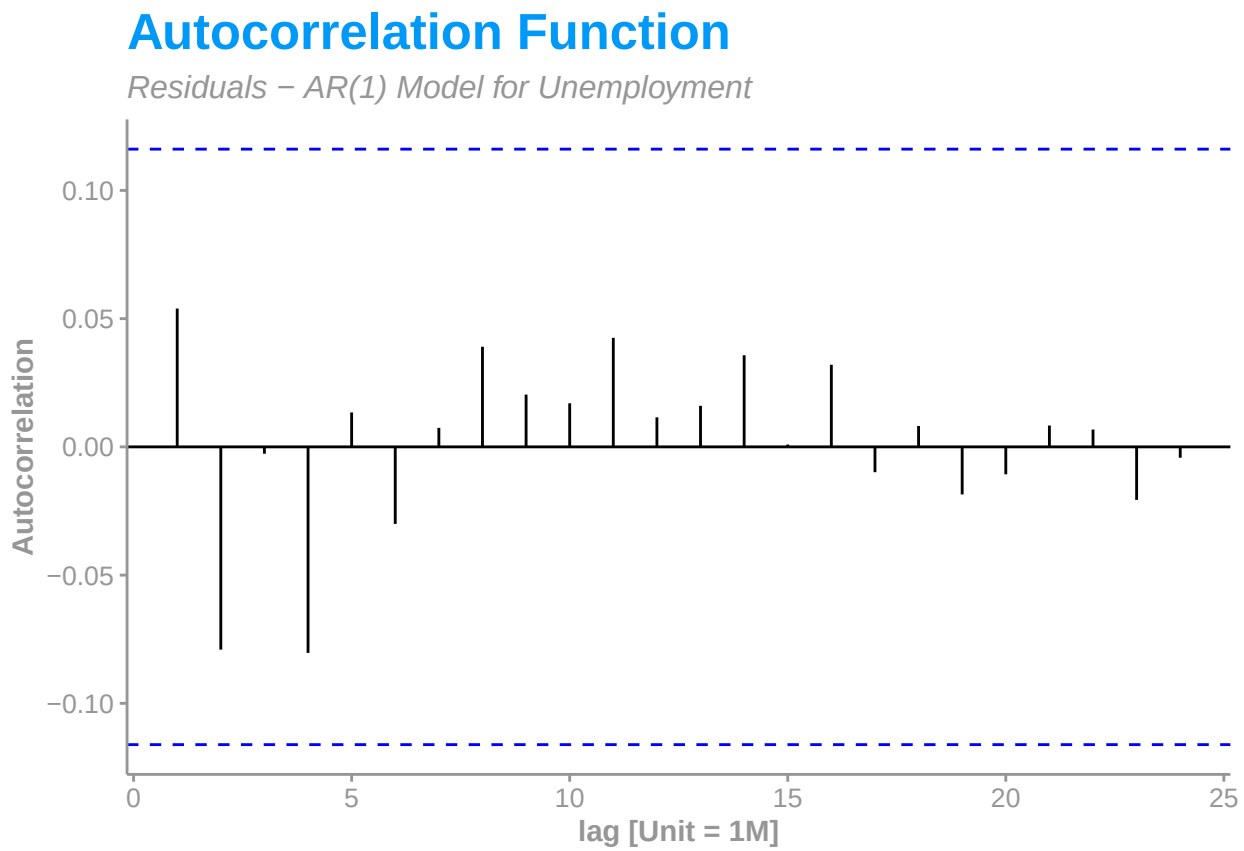
```
##
##  Box-Ljung test
##
## data:  resid.ar1
## X-squared = 2.6441, df = 2, p-value = 0.2666
```

```
Box.test(resid.ar1, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.ar1
## X-squared = 5.5178, df = 10, p-value = 0.854
```

With lags $k = 1$, $k = 2$ and $k = 10$ we fail to reject the null hypothesis (H_0 : data are independently distributed), although the p-value for the small lags is not as high as with other series, so we'll proceed to analyze the residuals with an ACF plot

```
mod.unm.ar1 %>%
  augment() %>%
  ACF(.resid) %>%
  autoplot()+ labs(
    x = "lag [Unit = 1M]",
    y = "Autocorrelation",
    title = "Autocorrelation Function",
    subtitle = "Residuals - AR(1) Model for Unemployment"
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
  axis.ticks = element_line(color = "#969696"),
  ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))
```



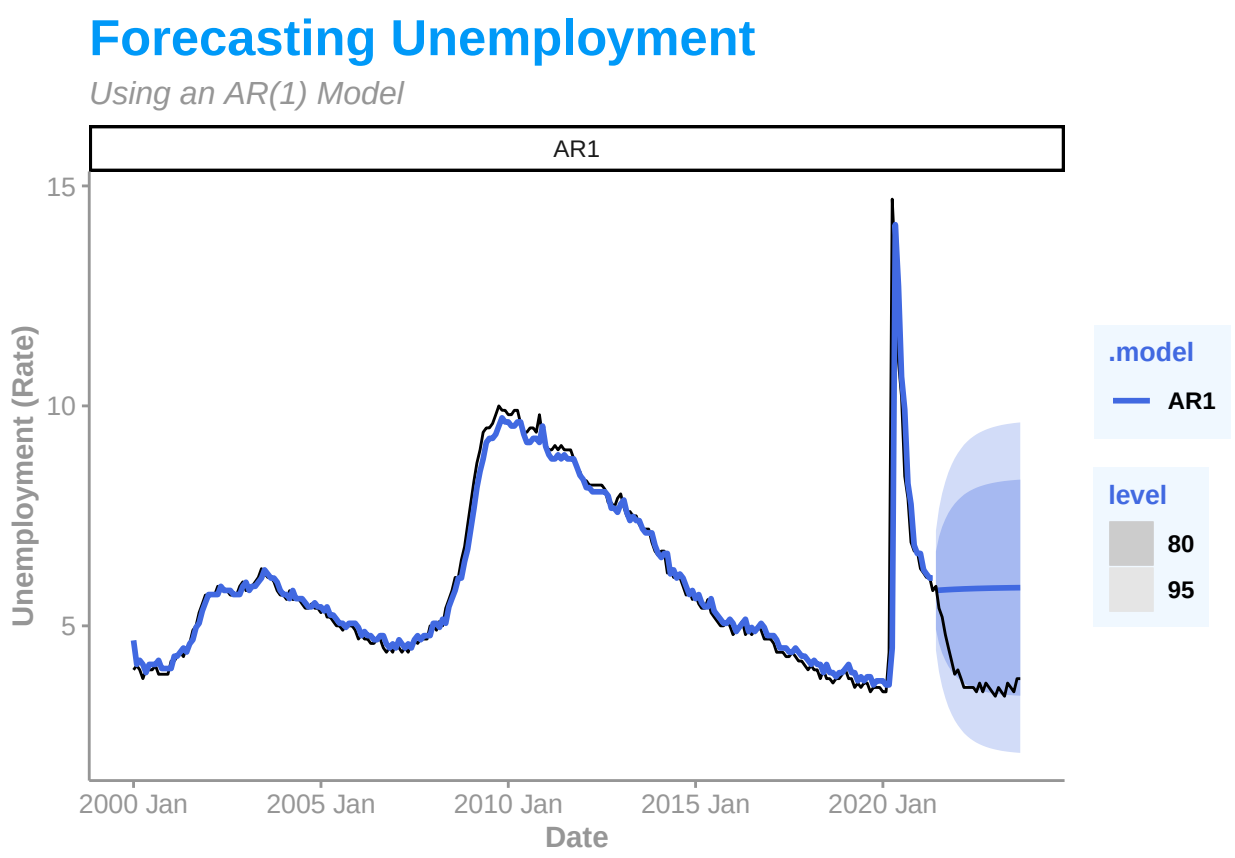
And this still doesn't look quite like white noise even when it doesn't have significant ACF values.

Finally we'll split our data in training (90%) and testing (10%) data to produce a forecast with the model as follows:

```
test.size.unm <- trunc(dim(unemployment)[1] *.1)

unm.train <- unemployment %>%
  slice(1:(n()-test.size.unm))

mod.trn.unm <- unm.train %>%
  model(AR1 = ARIMA(overall ~ 1 + pdq(1,0,0), stepwise=F, greedy=F))

mod.unm.for <- forecast(mod.trn.unm, h=test.size.unm)

mod.unm.for %>%
  autoplot(colour="royalblue", size = 1) +
  autolayer(unm.short, colour="black") +
  geom_line(data = mod.trn.unm %>% augment(),
            size = 1,
            aes(time_index, .fitted, color = .model)) +
  facet_wrap(~.model, ncol=1, nrow=2) +
  scale_color_manual(values = c("AR1" = "royalblue")) +
  labs(
    x = "Date",
    y = "Unemployment (Rate)",
    title = "Forecasting Unemployment",
```

```
    subtitle = "Using an AR(1) Model"
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
  axis.ticks = element_line(color = "#969696"),
  legend.title = element_text(color = "royalblue",
                              size = 10,
                              face = "bold"),
  legend.background = element_rect(fill ="aliceblue"),
  legend.text=element_text(size=9, face = "bold")
  )
```

```
## Plot variable not specified, automatically selected `.vars = overall`
```



And we can see that even when the real values are still within the confidence interval, perhaps we can still derive in a better model.

- (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.

As we said we seem to have a non stationary series, so perhaps we should treat it as one and use the difference in order to achieve stationarity as follows:

```
mod.unm.diff <- unemployment %>%
  model(ARIMA(overall ~ 1 + pdq(1:10,0:2,0:10), ic="aic", stepwise=F, greedy=F))

mod.unm.diff %>%
report()
```

```
## Series: overall
## Model: ARIMA(1,1,2) w/ drift
##
## Coefficients:
##          ar1      ma1      ma2  constant
##       0.6386  -0.6232  -0.1357   -0.0004
## s.e.  0.2100   0.2158   0.0702    0.0095
##
## sigma^2 estimated as 0.4413:  log likelihood=-284.83
## AIC=579.67   AICc=579.89   BIC=597.91
```

```
resid.diff <- mod.unm.diff %>%
  augment() %>%
  select(.resid) %>%
  as.ts()

Box.test(resid.diff, lag = 1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.diff
## X-squared = 0.0016095, df = 1, p-value = 0.968
```

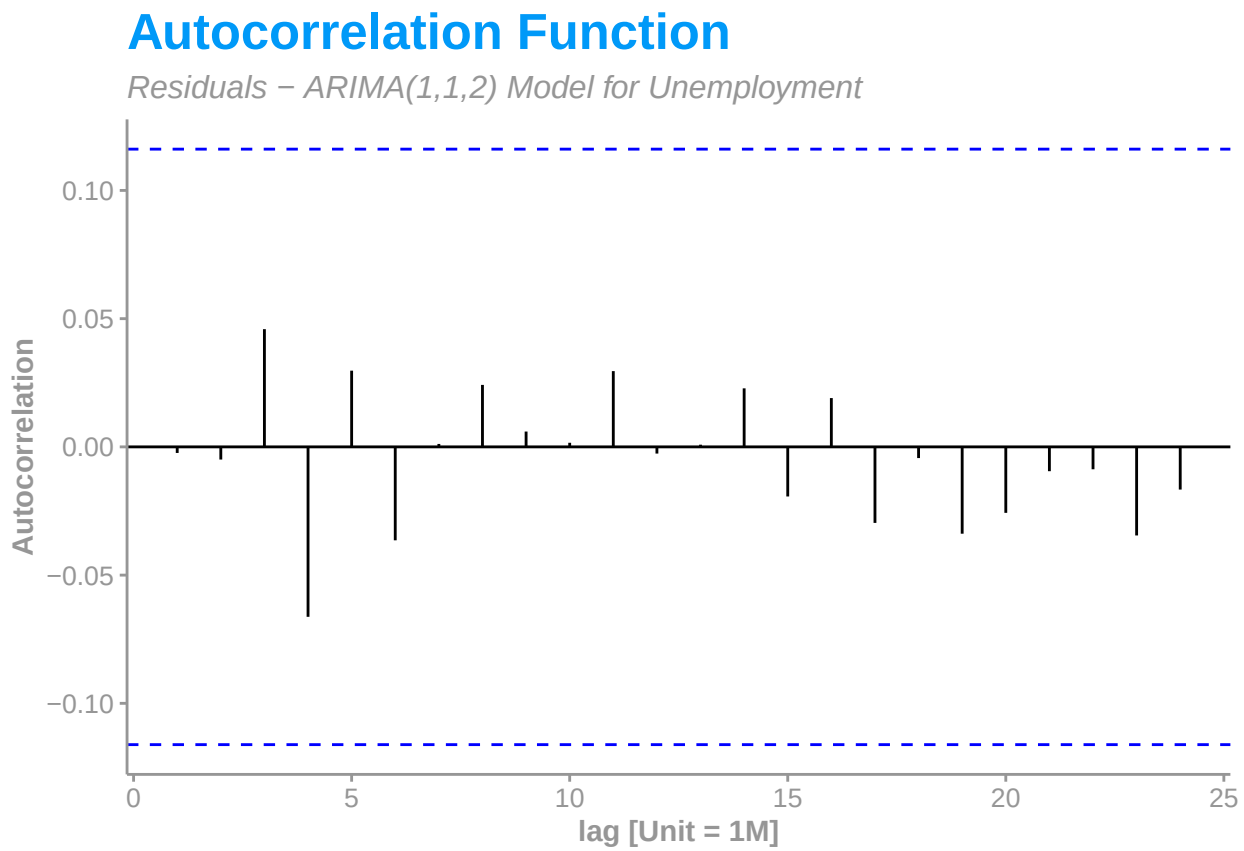
```
Box.test(resid.diff, lag = 2, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.diff
## X-squared = 0.0086346, df = 2, p-value = 0.9957
```

```
Box.test(resid.diff, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.diff
## X-squared = 2.7293, df = 10, p-value = 0.9871
```

```
mod.unm.diff %>%
  augment() %>%
  ACF(.resid) %>%
  autoplot()+ labs(
    x = "lag [Unit = 1M]",
    y = "Autocorrelation",
    title = "Autocorrelation Function",
    subtitle = "Residuals - ARIMA(1,1,2) Model for Unemployment"
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
  axis.ticks = element_line(color = "#969696"),
  ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))
```



We seem to have better Ljung Box tests, that we can also see in the residuals. Although there still seems to be room for improvement perhaps by using a linear model to extract the deterministic component first.

(14 Points Total) Question-2: Forecasting Inflation

The Federal Reserve tracks inflation data across countries on a monthly level.

(1 point) Part-1: Load and Clean Data

Load the CSV provided and store the data in a useful dataframe. Check for missing values and outliers in the data. Perform any cleaning that is necessary.

Also create lagged columns for GBR and CAN and training and test datasets based on pre and post Jan 1, 2022.

```
infl_df <- read.csv("./data/inflation_country.csv", header=T,na.strings=c("","NA"))

# Adding lagged Columns
CAN_LAG <- append(infl_df$CAN,infl_df$CAN[1],0)[1:(length(infl_df$CAN))]
GBR_LAG <- append(infl_df$GBR,infl_df$GBR[1],0)[1:(length(infl_df$CAN))]

infl_df$CAN_LAG <- CAN_LAG
infl_df$GBR_LAG <- GBR_LAG

# Creating tsibble
```

```
infl_ts <- infl_df %>%
  mutate(time_index = yearmonth(as.Date(date))) %>%
  as_tsibble(index = time_index)

sapply(infl_ts, function(x) sum(is.na(x)))
```

```
##       date        AUT        BEL        CAN        CHE        CHL        DEU
##          0          0          0          0          0          0          0
##        DNK        ESP        FIN        FRA        GBR        GRC        ISL
##          0          0          0          0          0          0          0
##        ISR        ITA        JPN        KOR        LUX        MEX        NLD
##          0          0          0          0          0          0          0
##        NOR        PRT        SWE        TUR        USA    CAN_LAG    GBR_LAG
##          0          0          0          0          0          0          0
## time_index
##          0
```

(5 points) Part-2: Produce a Model on the Training Data

1. Select inflation data for the US.

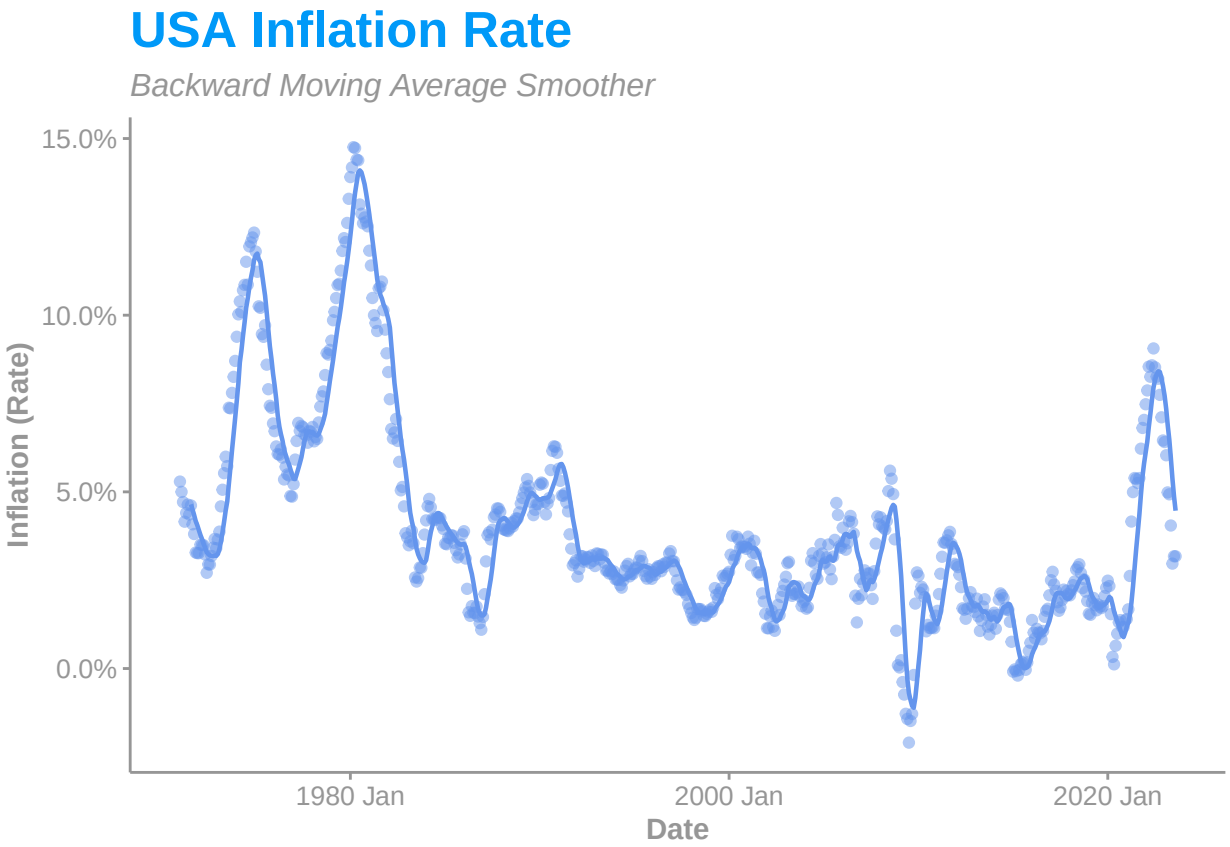
```
infl_usa_raw <- infl_ts[c("USA", "time_index")]
```

2. Produce a 7-day, backward smoother of inflation and plot the original time series with its smoothed trend.

```
infl_usa <- infl_usa_raw %>%
   mutate(
    back_smth = slider::slide_dbl(USA, mean,
                                  .before = 7,
                                  .after = 0,
                                  .complete = TRUE)
  )

usa_plot <- ggplot(
  infl_usa,
  aes(
    x = time_index
    )
  ) +
  geom_point(
    aes(y = USA),
    size = 1.5,
    alpha = 0.5,
    color = "cornflowerblue"
    ) +
  geom_line(
    aes(y = back_smth),
    size = 0.8,
    color = "cornflowerblue"
    ) +
```

```
  labs(
    title = "USA Inflation Rate",
    subtitle = "Backward Moving Average Smoother",
    x = "Date",
    y = "Inflation (Rate)"
    ) +
  theme_classic() +
  theme(
    plot.title = element_text(color = "#0099F8",
                              size = 19,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 12,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 11,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 10),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696"),
    legend.title = element_text(color = "royalblue",
                                size = 7,
                                face = "bold"),
    legend.background = element_rect(fill ="aliceblue"),
    legend.text=element_text(size=7),
    legend.position="bottom"
    ) + scale_y_continuous(labels = scales::percent)

usa_plot
```

```
## Warning: Removed 7 row(s) containing missing values (geom_path).
```



3. Produce a time series model of inflation. This should include:

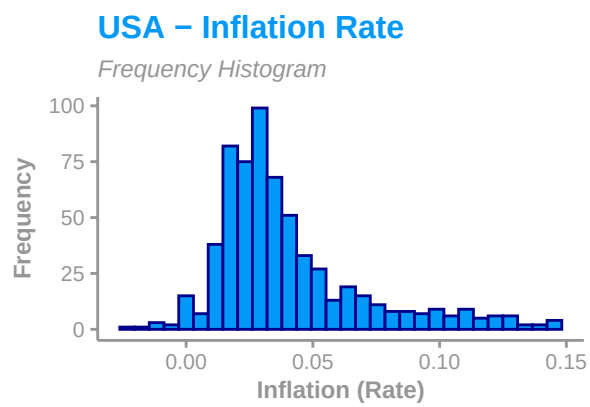
- Conducting a full EDA and description of the data that you observe.

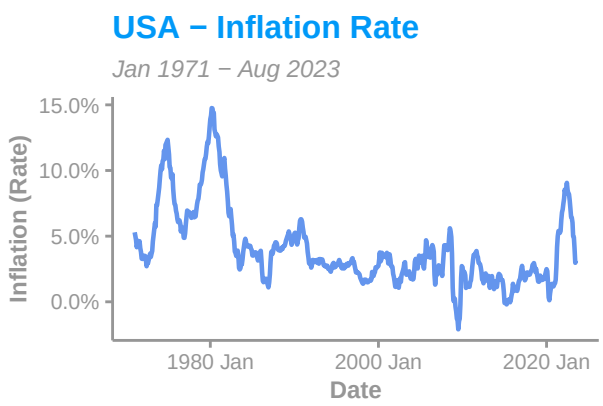
```
inf_hist <- ggplot(infl_usa,
  aes(
    x = USA
    )) +
    geom_histogram(
      aes(y = ..count..),
      col = I("darkblue"),
      fill = "#0099F8") +
    labs(
      title = "USA - Inflation Rate",
      subtitle = "Frequency Histogram",
      x = "Inflation (Rate)",
      y = "Frequency"
    ) +
    theme_classic() +
    theme(
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 9,
```

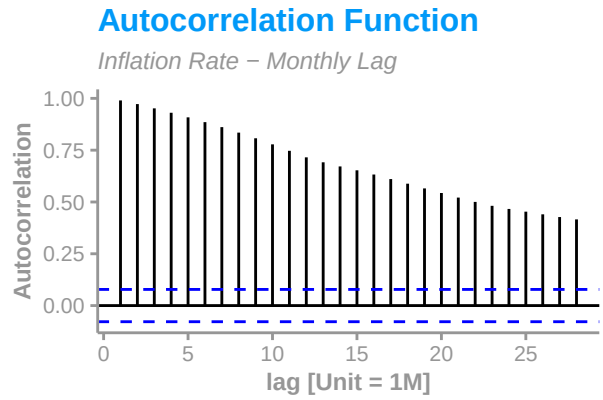
```
                                face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 9,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696")
    )

inf_plot <- ggplot(
  infl_usa,
  aes(
    x = time_index,
    y = USA
    )
  ) +
  geom_line(
    size = 0.8,
    color = "cornflowerblue"
    ) +
  labs(
    title = "USA - Inflation Rate",
    subtitle = "Jan 1971 - Aug 2023",
    x = "Date",
    y = "Inflation (Rate)"
       ) +
    theme_classic() +
    theme(
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 9,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 9,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696"),
    ) + scale_y_continuous(labels = scales::percent)

inf_acf <- ACF(
      fill_gaps(infl_usa, .full = TRUE),
      USA
    ) %>% autoplot() + labs(
      x = "lag [Unit = 1M]",
      y = "Autocorrelation",
      title = "Autocorrelation Function",
      subtitle = "Inflation Rate - Monthly Lag"
    ) +
    theme_classic() +
    theme(
```

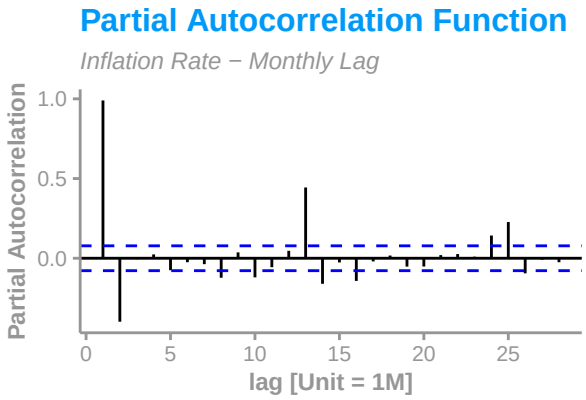
```
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 9,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 9,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696"),
    ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))

inf_pacf <- PACF(
      fill_gaps(infl_usa, .full = TRUE),
      USA
    ) %>% autoplot() + labs(
      x = "lag [Unit = 1M]",
      y = "Partial Autocorrelation",
      title = "Partial Autocorrelation Function",
      subtitle = "Inflation Rate - Monthly Lag"
    ) +
    theme_classic() +
    theme(
    plot.title = element_text(color = "#0099F8",
                              size = 12,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 9,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 9,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 8),
    axis.line = element_line(color = "#969696"),
    axis.ticks = element_line(color = "#969696"),
    ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))

(inf_hist | inf_plot) /(inf_acf | inf_pacf)
```

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









We can observe that the series might not be stationary and we have once again a slowly decaying ACF with a PACF that decays abruptly after lag 2 but still has some significant values after certain lags which could be derived of yearly seasonal behavior, and it also fluctuates between positive and negative numbers.

- Estimating a model that you believe is appropriate after conducting your EDA.

Using ARIMA to estimate the model we have the following:

```
inf.train <- infl_usa %>%
  filter_index(~ "2021-12-31")

test.size.inf <- dim(infl_usa)[1] - dim(inf.train)[1]

mod.inf.arpmaq <- inf.train %>%
  model(ARIMA(USA ~ 1 + pdq(1:10,0:2,0:10) + PDQ(0,0,0), ic="bic", stepwise=F, greedy=F))

mod.inf.arpmaq %>% report()
```

```
## Series: USA
## Model: ARIMA(5,1,1) w/ drift
##
## Coefficients:
##           ar1     ar2      ar3      ar4     ar5     ma1  constant
##       -0.5106  0.3865  -0.0819  -0.0368  0.1415  0.9841     0e+00
## s.e.   0.0405  0.0450   0.0477   0.0452  0.0406  0.0081     3e-04
```

```
##
## sigma^2 estimated as 1.312e-05:  log likelihood=2570.04
## AIC=-5124.07   AICc=-5123.83   BIC=-5088.75
```

- Evaluating the model performance through diagnostic plots and making any necessary adjustments to satisfy key assumptions.

We'll start by analyzing the residuals:

```
resid.inf <- mod.inf.arpmaq %>%
  augment() %>%
  select(.resid) %>%
  as.ts()

Box.test(resid.inf, lag = 1, type = "Ljung-Box")
```

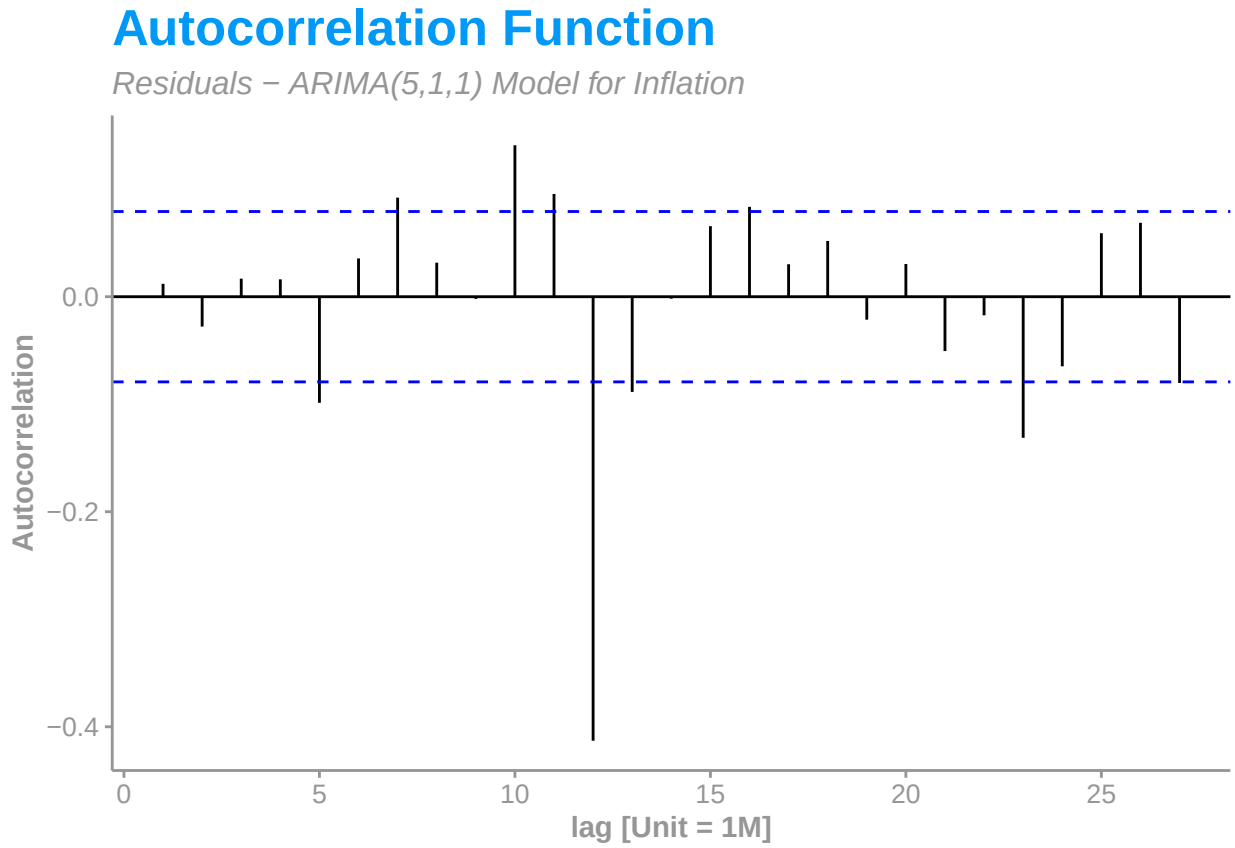
```
##
##  Box-Ljung test
##
## data:  resid.inf
## X-squared = 0.088476, df = 1, p-value = 0.7661
```

```
Box.test(resid.inf, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.inf
## X-squared = 25.996, df = 10, p-value = 0.003745
```

```
mod.inf.arpmaq %>%
  augment() %>%
  ACF(.resid) %>%
  autoplot() + labs(
    x = "lag [Unit = 1M]",
    y = "Autocorrelation",
    title = "Autocorrelation Function",
    subtitle = "Residuals - ARIMA(5,1,1) Model for Inflation"
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
```

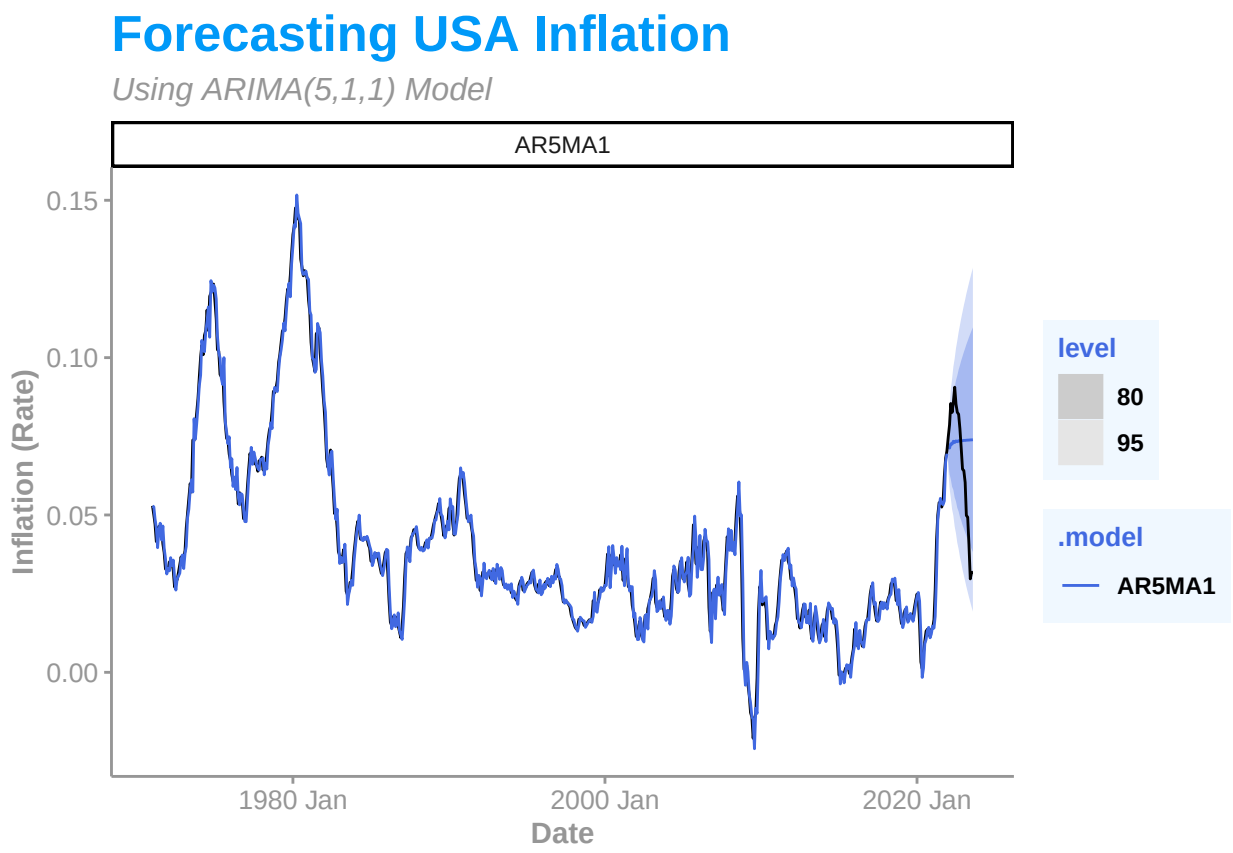
```
  axis.ticks = element_line(color = "#969696"),
  ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))
```




```
mod.trn.inf <- inf.train %>%
  model(AR5MA1 = ARIMA(USA ~ 1 + pdq(5,1,1) + PDQ(0,0,0), ic="bic", stepwise=F, greedy=F))

model.forecasts <- forecast(mod.trn.inf, h = test.size.inf)

model.forecasts %>%
  autoplot(colour="royalblue") +
  autolayer(infl_usa, colour="black") +
  geom_line(data = mod.trn.inf %>% augment(), aes(time_index,.fitted, color = .model)) +
  facet_wrap(~.model, ncol=1, nrow=2) +
  scale_color_manual(values = c("AR5MA1" = "royalblue")) +
  labs(
    x = "Date",
    y = "Inflation (Rate)",
    title = "Forecasting USA Inflation",
    subtitle = "Using ARIMA(5,1,1) Model"
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
```

```
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
  axis.ticks = element_line(color = "#969696"),
  legend.title = element_text(color = "royalblue",
                              size = 10,
                              face = "bold"),
  legend.background = element_rect(fill ="aliceblue"),
  legend.text=element_text(size=9, face = "bold")
  )
```

```
## Plot variable not specified, automatically selected `.vars = USA`
```



While the prediction remains inside of the confidence intervals, we can see that the residuals still have a lot of stationarity and especially when analyzing certaing lags we hve a Ljung Box test that rejects the null hypothesis (H_0 : data are independently distributed) for lag $k = 10$. Perhaps due to a seasonal factor that we still need to analyze.

(5 points) Part-3: Leveraging Other Countries

1. Examine how correlated (lagged) GBR and CAN inflation is with the US. Do you think prior inflation data in these countries can help forecast inflation in the US?

```
inf_ctry <- infl_ts[c("USA", "CAN_LAG", "GBR_LAG","time_index")]
inf_ctry <- na.omit(inf_ctry)
ctr_cor <- cor(inf_ctry[,1:3])
ctr_cor
```

```
##                USA   CAN_LAG   GBR_LAG
## USA     1.0000000 0.8662467 0.8104065
## CAN_LAG 0.8662467 1.0000000 0.8293122
## GBR_LAG 0.8104065 0.8293122 1.0000000
```

The values for lagged GBR and CAN inflation seem highly correlated with USA's values, so it seems like they might be useful in our model.

2. Build an appropriate time series model for US inflation, leveraging prior data in the selected countries (with a wide tsibble with each country as a column, we can add additional predictors to `ARIMA()` by name like we would in `lm()`):

- This is known as adding exogenous variables to `ARIMA()` and behind the scenes the function will estimate a linear time series model with exogenous variables as predictors, using an `ARIMA()` model for the residuals
- Estimate a model that you believe is appropriate after conducting your EDA.
- Evaluate the model performance through diagnostic plots and making any necessary adjustments to satisfy key assumptions.

```
inf.train <- inf_ctry %>%
  filter_index(~ "2021-12-31")

mod.inf.ctr <- inf.train %>%
  model(ARIMA(USA ~ 1 + pdq(1:10,0:2,0:10) + PDQ(0,0,0) + GBR_LAG + CAN_LAG, ic="bic", stepwise=F, gree

mod.inf.ctr %>% report()
```

```
## Series: USA
## Model: LM w/ ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2  GBR_LAG  CAN_LAG  intercept
##       1.3536  -0.3723   0.1070   0.0373     0.0334
## s.e.  0.0413   0.0407   0.0322   0.0338     0.0079
##
## sigma^2 estimated as 1.367e-05:  log likelihood=2559.54
## AIC=-5107.07   AICc=-5106.93   BIC=-5080.57
```

So we have a model of the form:

$$x_t = 0.325 + 1.3536 * x_{t-1} - 0.3723 * x_{t-2} + 0.1070 * GBR_{t-1} + 0.0373 * CAN_{t-1} + w_t$$

With the following residual analysis:

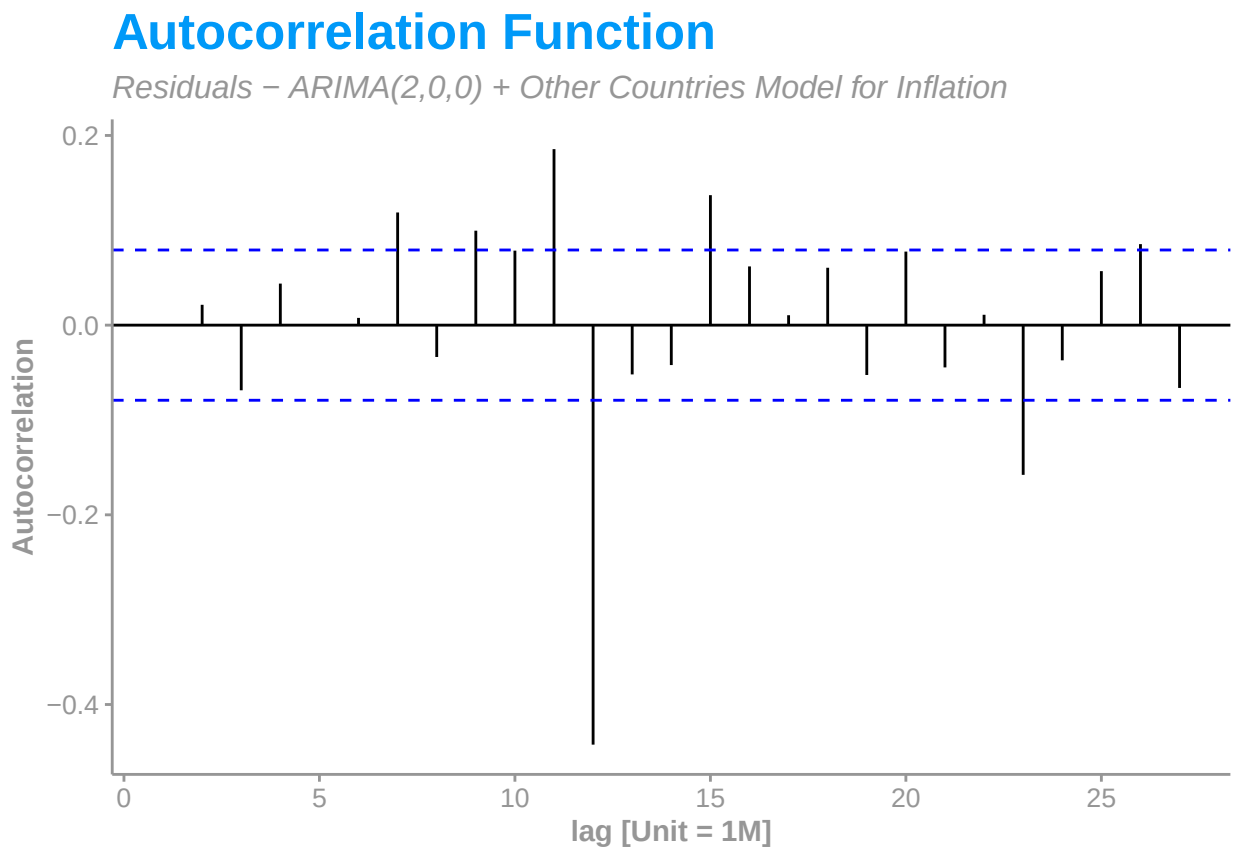
```
resid.ctr <- mod.inf.ctr %>%
  augment() %>%
  select(.resid) %>%
  as.ts()

Box.test(resid.ctr, lag = 1, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.ctr
## X-squared = 0.00029309, df = 1, p-value = 0.9863
```

```
Box.test(resid.ctr, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resid.ctr
## X-squared = 23.916, df = 10, p-value = 0.007825
```

```
mod.inf.ctr %>%
  augment() %>%
  ACF(.resid) %>%
  autoplot() + labs(
    x = "lag [Unit = 1M]",
    y = "Autocorrelation",
    title = "Autocorrelation Function",
    subtitle = "Residuals - ARIMA(2,0,0) + Other Countries Model for Inflation"
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
  axis.ticks = element_line(color = "#969696"),
  ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))
```



While Ljung Box Test gives better results and certain lags became less significant, we still have some patterns (perhaps seasonal), that we need to solve.

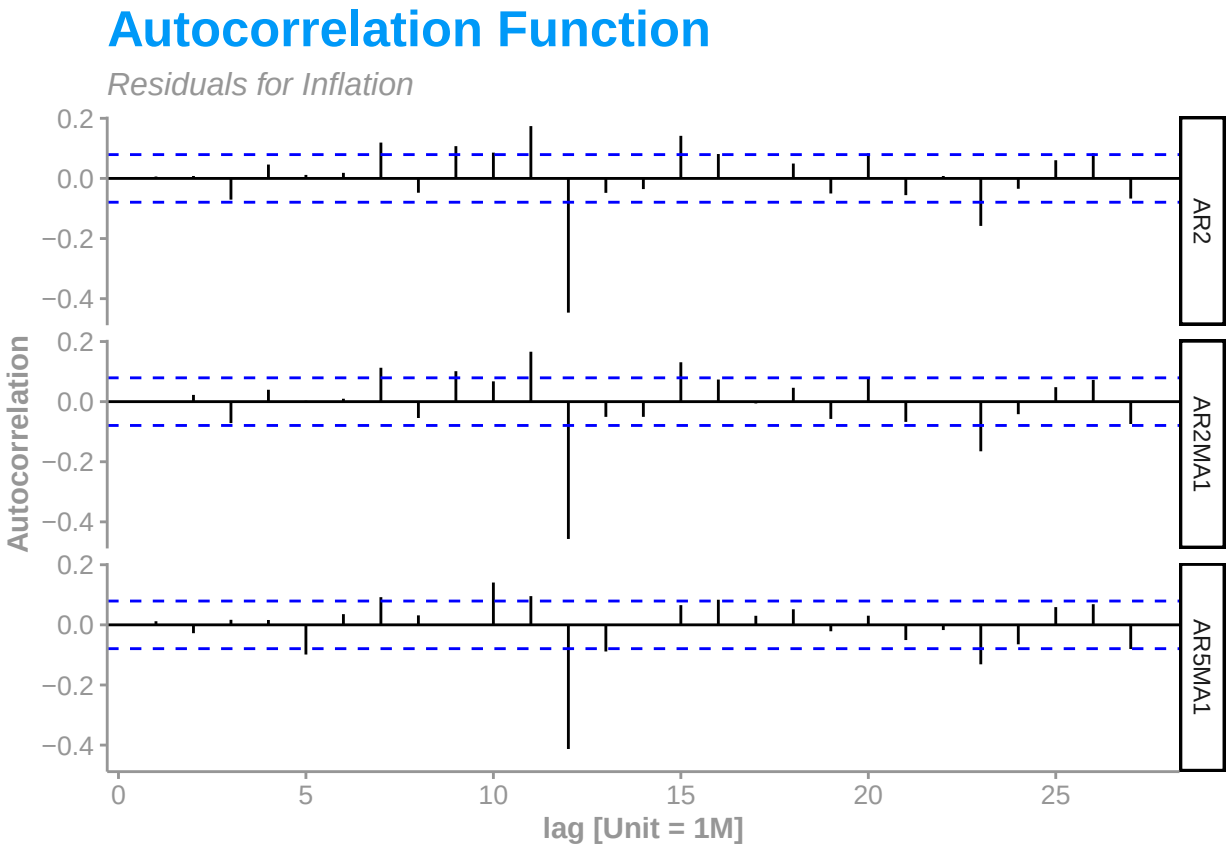
(3 points) Part-4: Forecasting and Model Comparison

Forecast inflation in 2022 and beyond (i.e. the test data) in the US using the various models you created. Include any models that did not necessarily satisfy the assumption of white noise in the residuals that you might have originally tried.

```
test.size.inf <- dim(inf_ctry)[1] - dim(inf.train)[1]

model.comp <- inf.train %>%
  model(
    AR2 = ARIMA(USA ~ 1 + pdq(2,0,0) + PDQ(0,0,0), ic="bic", stepwise=F, greedy=F),
    AR2MA1 = ARIMA(USA ~ 1 + pdq(2,1,0) + PDQ(0,0,0), ic="bic", stepwise=F, greedy=F),
    AR5MA1 = ARIMA(USA ~ 1 + pdq(5,1,1) + PDQ(0,0,0), ic="bic", stepwise=F, greedy=F)
    )
```

```
model.comp %>%
  augment() %>%
  ACF(.resid) %>%
  autoplot()+ labs(
    x = "lag [Unit = 1M]",
    y = "Autocorrelation",
    title = "Autocorrelation Function",
    subtitle = "Residuals for Inflation"
```

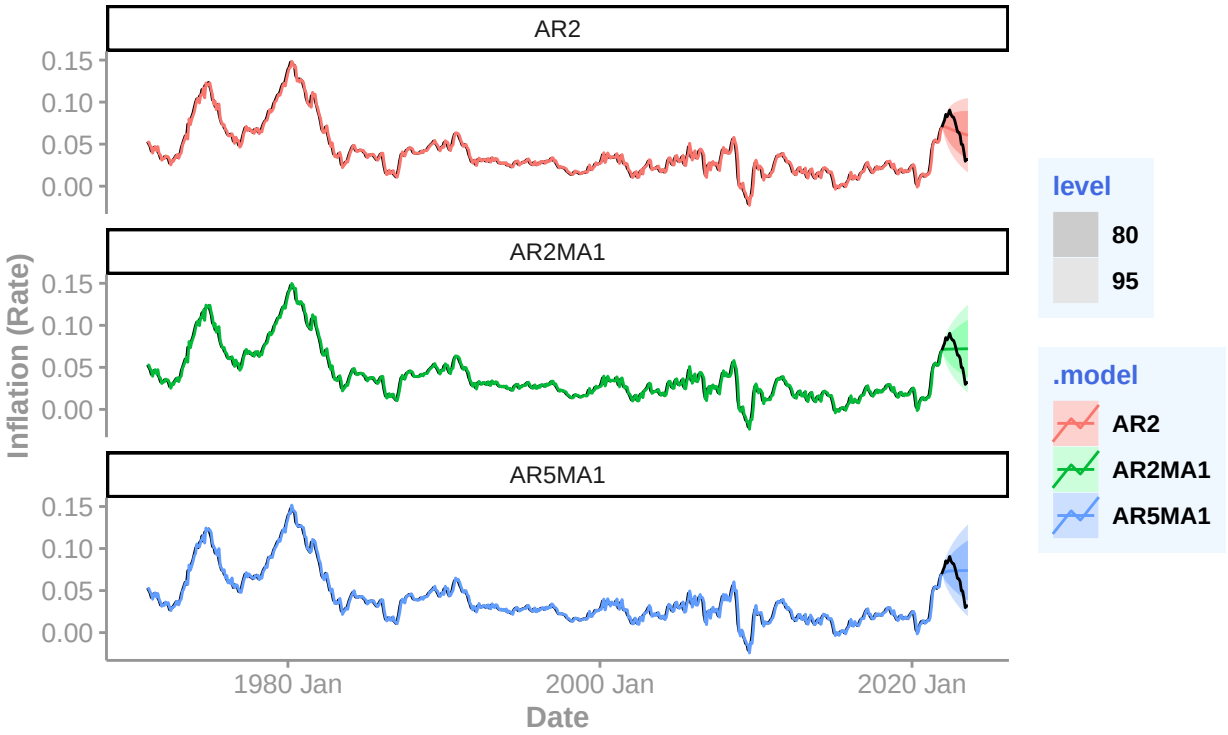
```
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
  axis.ticks = element_line(color = "#969696"),
  ) + scale_x_continuous(breaks = c(0,5,10,15,20,25))
```




```
model.forecasts<-forecast(model.comp, h=test.size.inf)

model.forecasts %>%
  autoplot(colour="cornflowerblue") +
  autolayer(inf_ctry, colour="black") +
  geom_line(data = model.comp %>% augment(), aes(time_index,.fitted,color=.model)) +
  facet_wrap(~.model, ncol=1, nrow=3) +
  labs(
```

```
    x = "Date",
    y = "Inflation (Rate)",
    title = "Forecasting Inflation",
    subtitle = "Using different ARIMA models"
  ) +
  theme_classic() +
  theme(
  plot.title = element_text(color = "#0099F8",
                            size = 19,
                            face = "bold"),
  plot.subtitle = element_text(color="#969696",
                               size = 12,
                               face = "italic"),
  axis.title = element_text(color = "#969696",
                            size = 11,
                            face = "bold"),
  axis.text = element_text(color = "#969696", size = 10),
  axis.line = element_line(color = "#969696"),
  axis.ticks = element_line(color = "#969696"),
  legend.title = element_text(color = "royalblue",
                              size = 10,
                              face = "bold"),
  legend.background = element_rect(fill ="aliceblue"),
  legend.text=element_text(size=9, face = "bold")
  )
```

```
## Plot variable not specified, automatically selected `.vars = USA`
```




```
accuracy(model.forecasts, inf_ctry)
```

```
## # A tibble: 3 x 10
##   .model .type        ME   RMSE    MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr>     <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 AR2    Test   0.000187 0.0170 0.0143 -11.4  28.9 0.903 0.789 0.880
## 2 AR2MA1 Test  -0.00592  0.0208 0.0168 -24.0  36.9 1.06  0.969 0.884
## 3 AR5MA1 Test  -0.00738  0.0215 0.0171 -26.7  38.2 1.08  1.00  0.885
```

Which model is the best? Explain your results.

For these models we have that the one with the lowest AIC, AICc and BIC is an ARIMA(5,1,1), although when predicting the new values we have that the one with the lowest RMSE and closer to the actual values is a plain AR(2) model. So perhaps we should use this model which is also easier to explain.