ECON 144 Project #2

Deniz Gorgulu and Yüksel Polat Akbıyık

2/20/2022

Introduction

In this paper, we will analyze the relationship between Hires and Job Openings in retail trade in United States. Both of these datasets are from FRED and they are calculated by using the data from U.S. Bureau of Labor Statistics in their Job Openings and Labor Turnover Survey. Both of the datasets are monthly and not seasonally adjusted. Job Openings show the number of job opening in the retail sector in thousands and Hires show the number of hires in retail sector in thousands.

Our initial assumption is that there is a positive relationship between number of Job Openings and Hires and Job Openings affect hires. We will analyze whether this is a one way relationship or a bi-directional relationship.

```
# Get the data and set column names
jobs = read.csv("JOBS.csv", col.names = c("Date", "Hires", "Openings"))
```

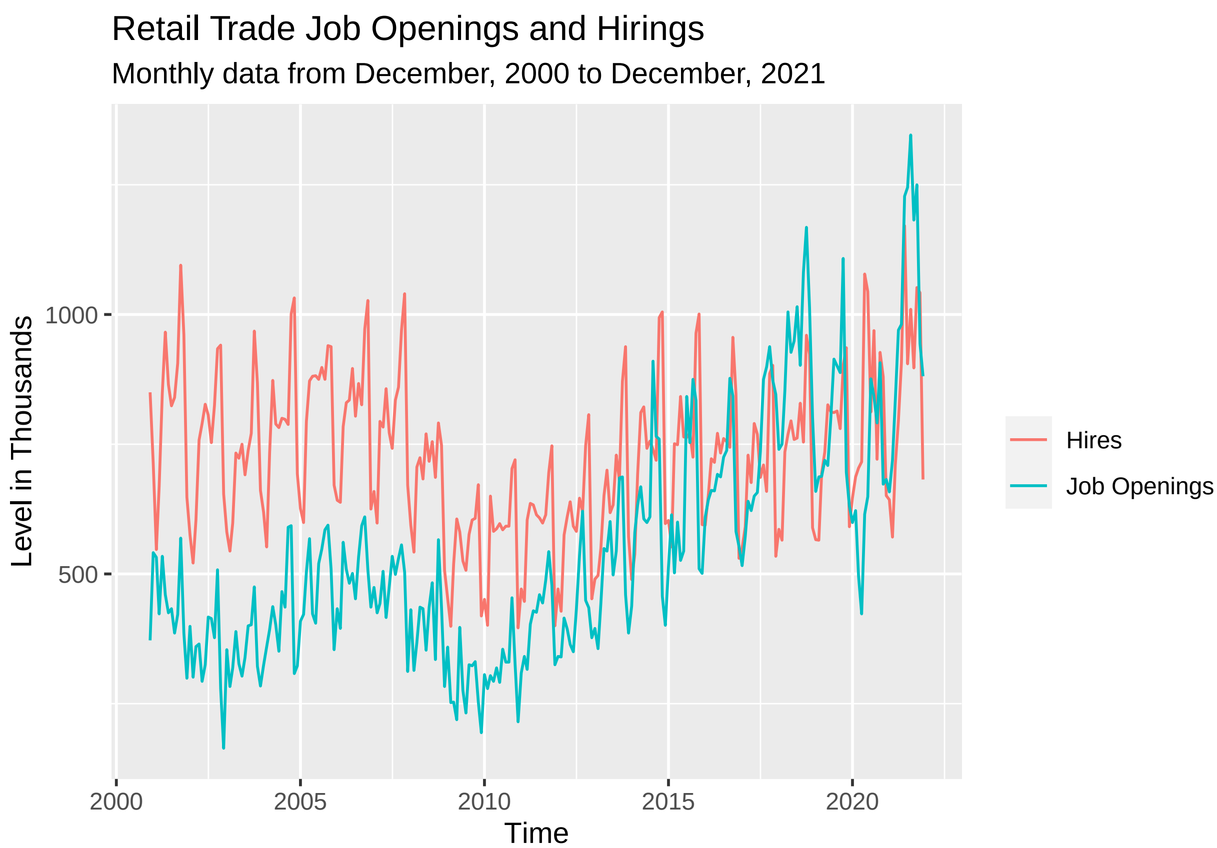
Results

(a) Produce a time-series plot of your data including the respective ACF and PACF plots.

```
# Convert the individual series into time series
hires_ts = ts(jobs[,2], start = c(2000,12), end = c(2021,12), frequency = 12)
open_ts = ts(jobs[,3], start = c(2000,12), end = c(2021,12), frequency = 12)
```

```
# Create a data frame containing both series
df = data.frame("hires" = hires_ts, "open" = open_ts,
                date = seq(as.Date("2000-12-01"), by = "1 month", length.out = 253) )
# Convert the data frame into a molten data frame indexing by date
df_long = melt(df, id="date")

# Plot the time series
ggplot(data=df_long,
       aes(x=date, y=value, colour=variable))+
       geom_line() +
  labs(x="Time", y="Level in Thousands",
       title = "Retail Trade Job Openings and Hirings",
       subtitle = "Monthly data from December, 2000 to December, 2021")+
  scale_color_discrete(name = "", breaks = c("hires", "open"),
                       labels = c("Hires", "Job Openings"))
```



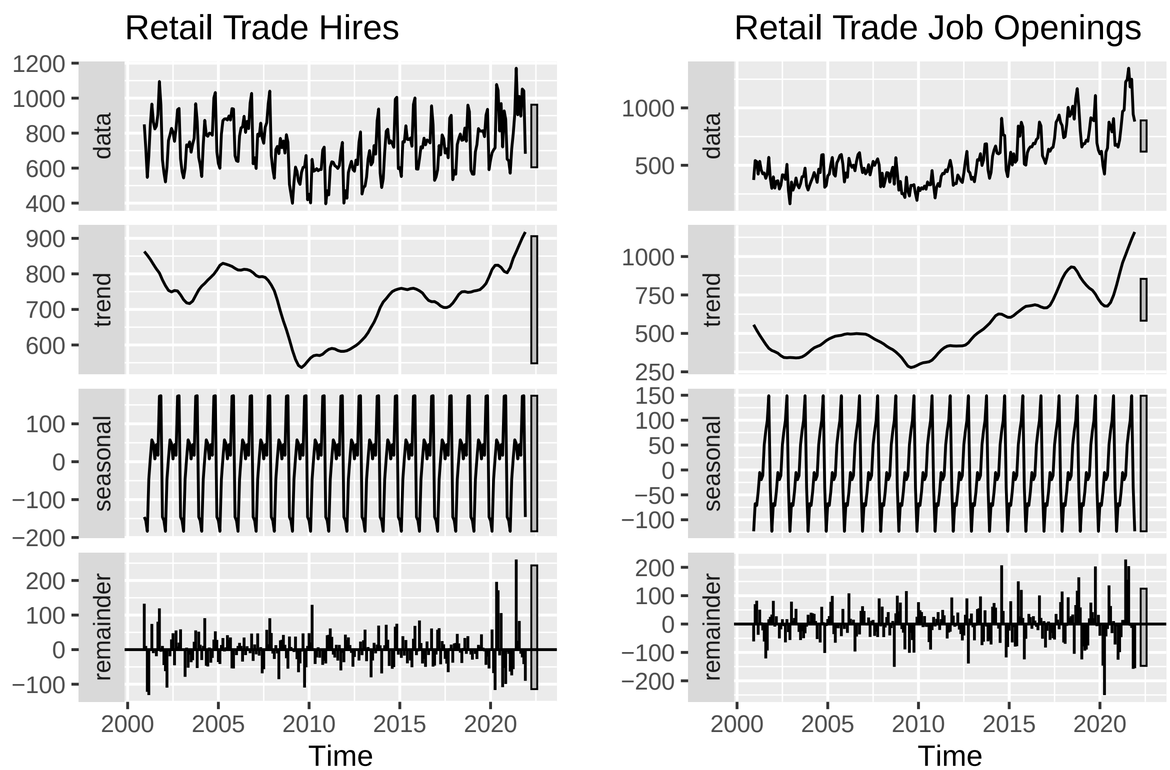

```
# Store ACF and PACF of both series into objects to plot them together
#-----------------HIRES--------------------
acf_hires = acf(hires_ts, plot = FALSE) %>% autoplot() +
  labs(title = "ACF of Hires")
pacf_hires = pacf(hires_ts, plot = FALSE) %>% autoplot() +
  labs(title = "PACF of Hires")
#-----------------JOB OPENINGS--------------------
acf_open = acf(open_ts, plot = FALSE) %>% autoplot() +
  labs(title = "ACF of Job Openings")
pacf_open = pacf(open_ts, plot = FALSE) %>% autoplot() +
  labs(title = "PACF of Job Openings")

# Plot the ACF and PACF plots of the two series together
(acf_hires + acf_open) / (pacf_hires + pacf_open) +
  plot_annotation(title = "ACF and PACF Plots of Retail Trade Hires and Job Openings")
```

(b) Plot the stl decomposition plot of your data, and discuss the results.

```
decomp_hires = stl(hires_ts, s.window = "periodic")
dcmplot_hires = autoplot(decomp_hires) +            # Perform an STL decomposition
  labs(title = "Retail Trade Hires")                # of both series and store the
decomp_open = stl(open_ts, s.window = "periodic")   # plots to display together
dcmplot_open = autoplot(decomp_open)  +
  labs(title = "Retail Trade Job Openings")
# Display the decomposition plots together
dcmplot_hires + dcmplot_open +
  plot_annotation(title = "STL Decompostion of Hires and Job Openings")
```

STL Decompostion of Hires and Job Openings



When we look at the STL decomposition of both series above, we can easily detect strong seasonality and some trend components for both series, though the trend of the hires series does not seem to be as deterministic. We also see that both series are non-stationary and display a high degree of persistence. Moreover, the remainder components of both decompositions show certain dynamics that can be fitted with a model for the cycles.

(c) Fit a model that includes, trend, seasonality and cyclical components. Make sure to discuss your model in detail.

```
# Fit a model with trend and seasonality to Hires and Job Openings using tslm
trend_hires = tslm(hires_ts~trend+I(trend^2))
trend_open = tslm(open_ts~trend+I(trend^2))
summary(trend_hires)
```

```
##
## Call:
## tslm(formula = hires_ts ~ trend + I(trend^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -304.29  -88.30    4.09   83.61  359.18
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
```

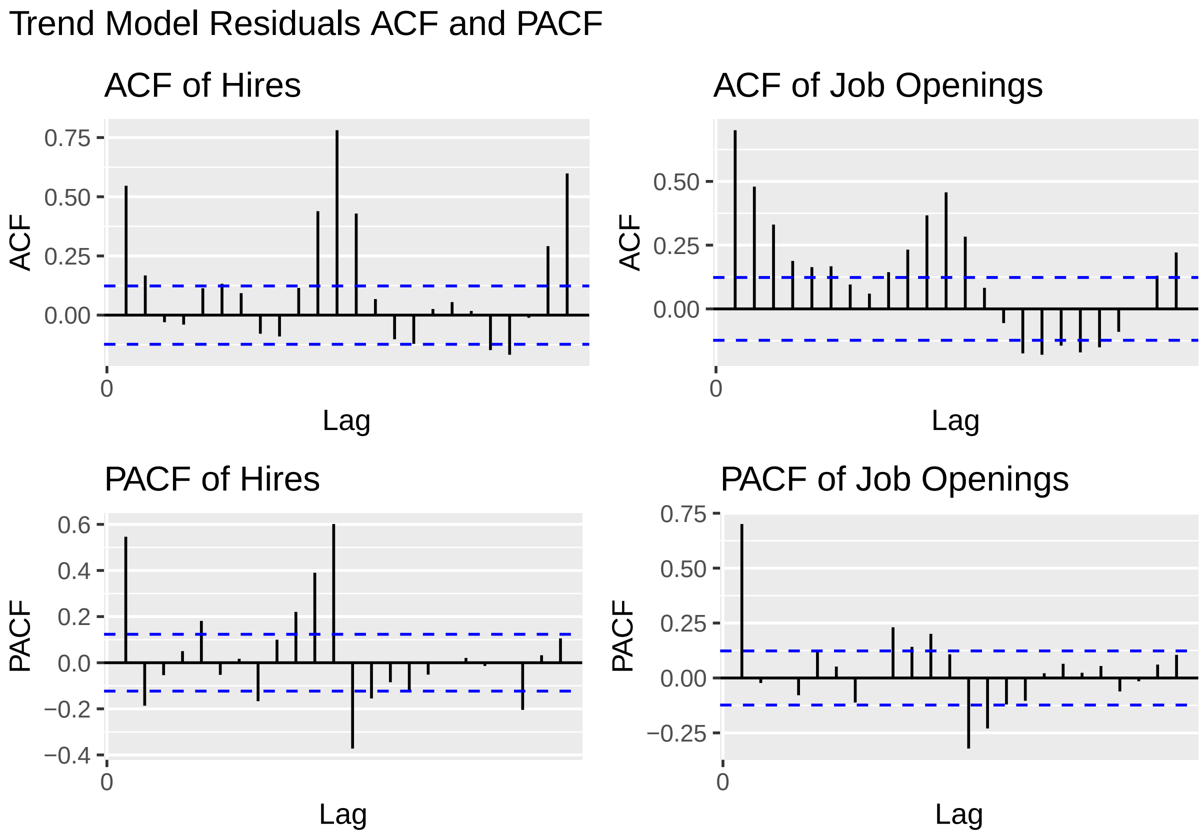
```
## (Intercept) 860.859323  26.612005  32.349  < 2e-16 ***
## trend        -3.228933   0.483801  -6.674 1.59e-10 ***
## I(trend^2)    0.012923   0.001845   7.006 2.27e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140 on 250 degrees of freedom
## Multiple R-squared:  0.1647, Adjusted R-squared:  0.158
## F-statistic: 24.64 on 2 and 250 DF,  p-value: 1.711e-10
```

```
summary(trend_open)
```

```
##
## Call:
## tslm(formula = open_ts ~ trend + I(trend^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -452.52 -101.91   -3.52   86.59  390.25
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 450.688913  25.856148  17.431  < 2e-16 ***
## trend        -2.086739   0.470060  -4.439 1.35e-05 ***
## I(trend^2)    0.016781   0.001792   9.363  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 136 on 250 degrees of freedom
## Multiple R-squared:  0.634,  Adjusted R-squared:  0.631
## F-statistic: 216.5 on 2 and 250 DF,  p-value: < 2.2e-16
```

After fitting a quadratic trend to both of the series, we can see that both models are jointly statistically significant at the less than 1% level. The trend component in both models are highly statistically significant, with the coefficients implying an exponentially decreasing trend. From the results of both of the models, we can conclude that the models are doing a fairly good job at fitting the trend component of the data, though we have not accounted for seasonality and cycles yet. To determine what kind of model to fit to the seasonal and remainder components, we will look at the residual ACF and PACF of our trend and seasonality models.

```
# Store the ACF and PACF plots of the residuals in objects for both series
#-------------HIRES-------------------------
acf_trend_hires = acf(resid(trend_hires), plot = FALSE) %>% autoplot() +
  labs(title = "ACF of Hires")
pacf_trend_hires = pacf(resid(trend_hires), plot = FALSE) %>% autoplot() +
  labs(title = "PACF of Hires")
#-------------JOB OPENINGS------------------
acf_trend_open = acf(resid(trend_open), plot = FALSE) %>% autoplot() +
  labs(title = "ACF of Job Openings")
pacf_trend_open = pacf(resid(trend_open), plot = FALSE) %>% autoplot() +
  labs(title = "PACF of Job Openings")
# Plot ACF and PACF plots together
(acf_trend_hires + acf_trend_open) / (pacf_trend_hires + pacf_trend_open) +
  plot_annotation(title = "Trend Model Residuals ACF and PACF")
```



The ACF and PACF plots for the trend and seasonality model residuals can be seen above. Since the data are clearly non-stationary, we will take the first seasonal difference to determine the order of our seasonal ARIMA model. Taking the first difference will consequently make I=1 in the S-ARIMA.

```
# Take seasonal difference and store the ACF and PACF plots
#--------------------HIRES---------------
sdres_hires = diff(resid(trend_hires),12)
acf_sd_hires = acf(sdres_hires, plot = FALSE) %>% autoplot() +
  labs(title = "ACF of Hires")
pacf_sd_hires = pacf(sdres_hires, plot = FALSE) %>% autoplot() +
  labs(title = "PACF of Hires")
#----------------JOB OPENINGS-------------
sdres_open = diff(resid(trend_open),12)
acf_sd_open = acf(sdres_open, plot = FALSE) %>% autoplot() +
  labs(title = "ACF of Job Openings")
pacf_sd_open = pacf(sdres_open, plot = FALSE) %>% autoplot() +
  labs(title = "PACF of Job Openings")

# Plot ACF and PACF plots together
(acf_sd_hires + acf_sd_open) / (pacf_sd_hires + pacf_sd_open) +
  plot_annotation(title = "Seasonally Differenced Trend Model Residuals ACF and PACF")
```

From the seasonally differenced trend model residuals ACF and PACF plots, we can determine the order of the ARIMA model for our cycles and the order of the S-ARIMA model for our seasonality component. The ACF plot of the hires residuals shows that the correlation coefficient exhibit a steady decay to zero, which indicates an AR(p) process for the cycles. We turn to the PACF of the hires residuals to determine the order. The PACF plot shows significant spikes at lags 12 and 13, which indicates that a S-AR(2) model would be appropriate to model the seasonality. We also see three significant spikes in the first three lags, indicating either an AR(3) model for the cycles or a combination ARMA(1,1) model.

The job opening residuals ACF plot clearly shows the behavior of an AR(p) process for the cycles, with the correlation coefficients decaying to zero. The significant spike at lag 12 implies that a S-MA(1) component might also be needed. The subsequent PACF plot shows significant spikes at the first three lags, though the most prominent one is the first spike. This could either indicate that an AR(3) process might appropriate for the cycles, or could indicate that a combination ARMA(1,1) process might be responsible for the behavior. Furthermore, the two significant spikes at lag 12 and 13 in the PACF plot implies that we should add a S-AR(2) component to the model.

From our analysis above, we determine that the models we are going to use to fit the trend, seasonality, and cycles in both series will be seasonal ARIMA models. Since there is a significant trend component in both of our series, we are going to difference the series through I=1 to account for the trend. Furthermore, since we previously took the seasonal difference of both series we are going to let I=1 in the seasonal ARIMA also. Since we have failed to choose between an AR(3) process and an ARMA(1,1) process for the cycles of both series, we will estimate both to determine which one gives lower AIC and BIC scores.

```r
# Fit 2 different ARIMA models to both series and compare
m1_hires = Arima(hires_ts, order = c(3,1,0), seasonal = list(order = c(2,1,0)))
m2_hires = Arima(hires_ts, order = c(1,1,1), seasonal = list(order = c(2,1,0)))
m1_open = Arima(open_ts, order = c(3,1,0), seasonal = list(order = c(2,1,1)))
```

```r
m2_open = Arima(open_ts, order = c(1,1,1), seasonal = list(order = c(2,1,1)))

# Look at the summaries
summary(m1_hires)
```

```
## Series: hires_ts
## ARIMA(3,1,0)(2,1,0)[12]
##
## Coefficients:
##           ar1      ar2      ar3     sar1     sar2
##       -0.5025  -0.3377  -0.1595  -0.5081  -0.2351
## s.e.   0.0672   0.0703   0.0646   0.0708   0.0745
##
## sigma^2 estimated as 3499:  log likelihood=-1319.23
## AIC=2650.47   AICc=2650.83   BIC=2671.35
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 2.783299 57.01252 41.60922 0.2687812 5.867684 0.6767325
##                      ACF1
## Training set -0.005244684
```

```r
summary(m2_hires)
```

```
## Series: hires_ts
## ARIMA(1,1,1)(2,1,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1     sar2
##       0.1718  -0.7053  -0.5055  -0.2244
## s.e.  0.1038   0.0719   0.0705   0.0745
##
## sigma^2 estimated as 3422:  log likelihood=-1317.07
## AIC=2644.13   AICc=2644.39   BIC=2661.54
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 3.664201 56.49942 41.30076 0.3754036 5.837346 0.6717157
##                     ACF1
## Training set 0.005796133
```

```r
summary(m1_open)
```

```
## Series: open_ts
## ARIMA(3,1,0)(2,1,1)[12]
##
## Coefficients:
##           ar1      ar2      ar3    sar1     sar2     sma1
##       -0.3866  -0.2155  -0.1135  0.1209  -0.0371  -0.7988
## s.e.   0.0655   0.0686   0.0660  0.0990   0.0872   0.0813
##
```

```
## sigma^2 estimated as 6467:  log likelihood=-1395.83
## AIC=2805.67   AICc=2806.15   BIC=2830.03
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 3.751137 77.33978 58.05942 -0.4111623 11.41014 0.5627541
##                     ACF1
## Training set -0.01570177
```

```
summary(m2_open)
```

```
## Series: open_ts
## ARIMA(1,1,1)(2,1,1)[12]
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1
##       0.2322  -0.6294  0.0926  -0.0500  -0.7913
## s.e.  0.1447   0.1164  0.1001   0.0878   0.0811
##
## sigma^2 estimated as 6368:  log likelihood=-1394.7
## AIC=2801.4   AICc=2801.76   BIC=2822.28
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 4.729094 76.91098 57.44201 -0.2757423 11.26551 0.5567698
##                     ACF1
## Training set -0.00577841
```

From the summaries above we can see that the ARIMA(1,1,1) has lower AIC values for both series. Therefore, our final model for the hires series will be an ARIMA(1,1,1)(2,1,0) and our final model for the job openings series will be an ARIMA(1,1,1)(2,1,1).

```
totmod_hires = Arima(hires_ts, order = c(1,1,1), seasonal = list(order = c(2,1,0)))
totmod_open = Arima(open_ts, order = c(1,1,1), seasonal = list(order = c(2,1,1)))
```

(e) Plot the respective residuals vs. fitted values and discuss your observations.

```
# Create objects for residuals and fitted values from the models
hires_res = residuals(totmod_hires)
hires_fit = fitted.values(totmod_hires)
open_res = residuals(totmod_open)
open_fit = fitted.values(totmod_open)

# Create data frames to be able to feed into ggplot function
resfit_hires = data.frame(x = hires_fit, y = hires_res)
resfit_open = data.frame(x = open_fit, y = open_res)

# Save the ggplots of the residuals vs fittted values to display together
p_resfit_hires = ggplot(resfit_hires, aes(x=x, y=y)) +
  geom_point(col = "indianred") +
  geom_hline(yintercept = 0, lty = 2) +
```

```r
  labs(x = "Fitted Values", y = "Residuals",
       title = "Hires Model")
p_resfit_open = ggplot(resfit_open, aes(x=x, y=y)) +
  geom_point(col = "slateblue3") +
  geom_hline(yintercept = 0, lty = 2) +
  labs(x = "Fitted Values", y = "Residuals",
       title = "Job Openings Model")

# Display plots together
p_resfit_hires + p_resfit_open +
  plot_annotation(title = "Residuals vs. Fitted Values Plots")
```

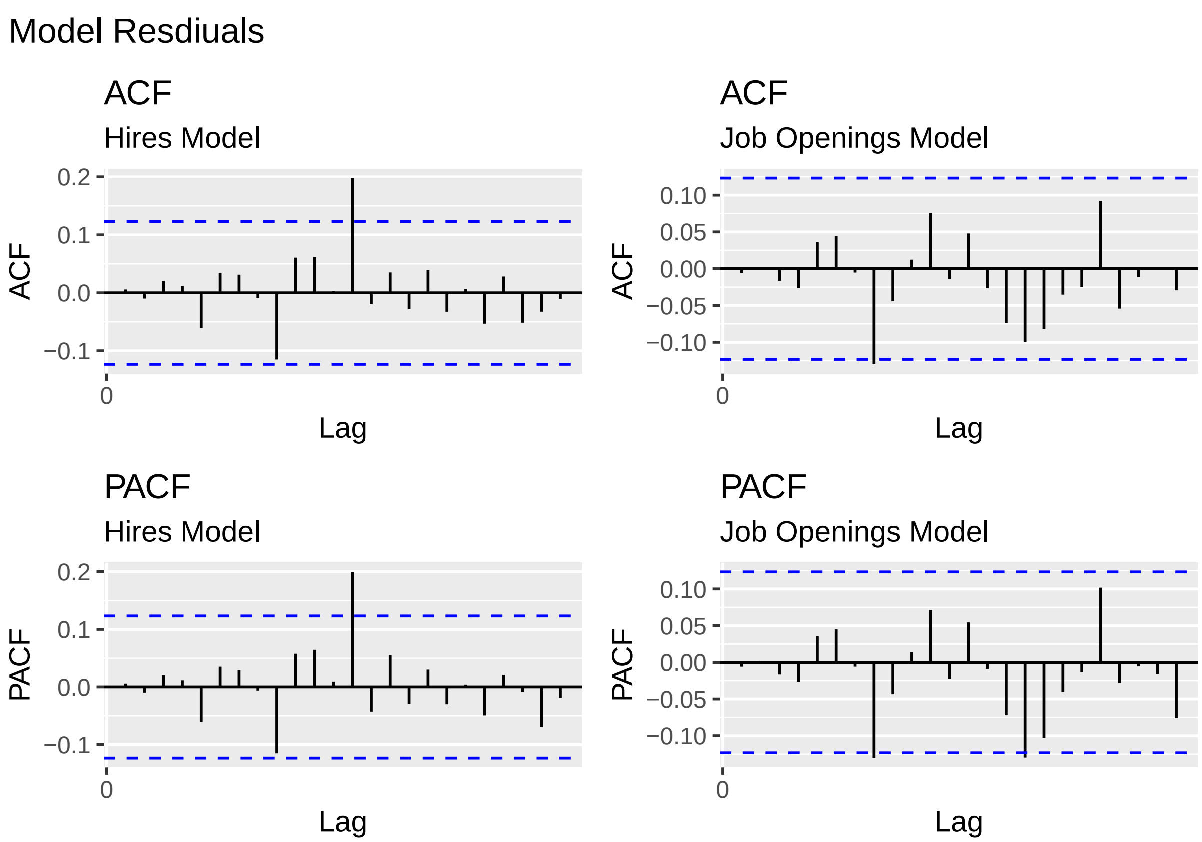
The residuals vs. fitted values plot shows that the residuals are somewhat random, though they still exhibit some variation with time. This implies that our model is doing a fairly good job at fitting the data, though it could definitely be further improved.

(f) Plot the ACF and PACF of the respective residuals and interpret the plots.

```r
# Store the ACF and PACF plots of the residuals in objects for both series
#-------------HIRES--------------------------
acf_hires_res = acf(hires_res, plot = FALSE) %>%
  autoplot() + labs(title = "ACF", subtitle = "Hires Model")
pacf_hires_res = pacf(hires_res, plot = FALSE) %>%
  autoplot() + labs(title = "PACF", subtitle = "Hires Model")
```

```
#---------------JOB OPENINGS-------------```
acf_open_res = acf(open_res, plot = FALSE) %>%
  autoplot() + labs(title = "ACF", subtitle = "Job Openings Model")
pacf_open_res = pacf(open_res, plot = FALSE) %>%
  autoplot() + labs(title = "PACF", subtitle = "Job Openings Model")

# Plot ACF and PACF plots together
(acf_hires_res + acf_open_res) / (pacf_hires_res + pacf_open_res) +
  plot_annotation("Model Resdiuals")
```



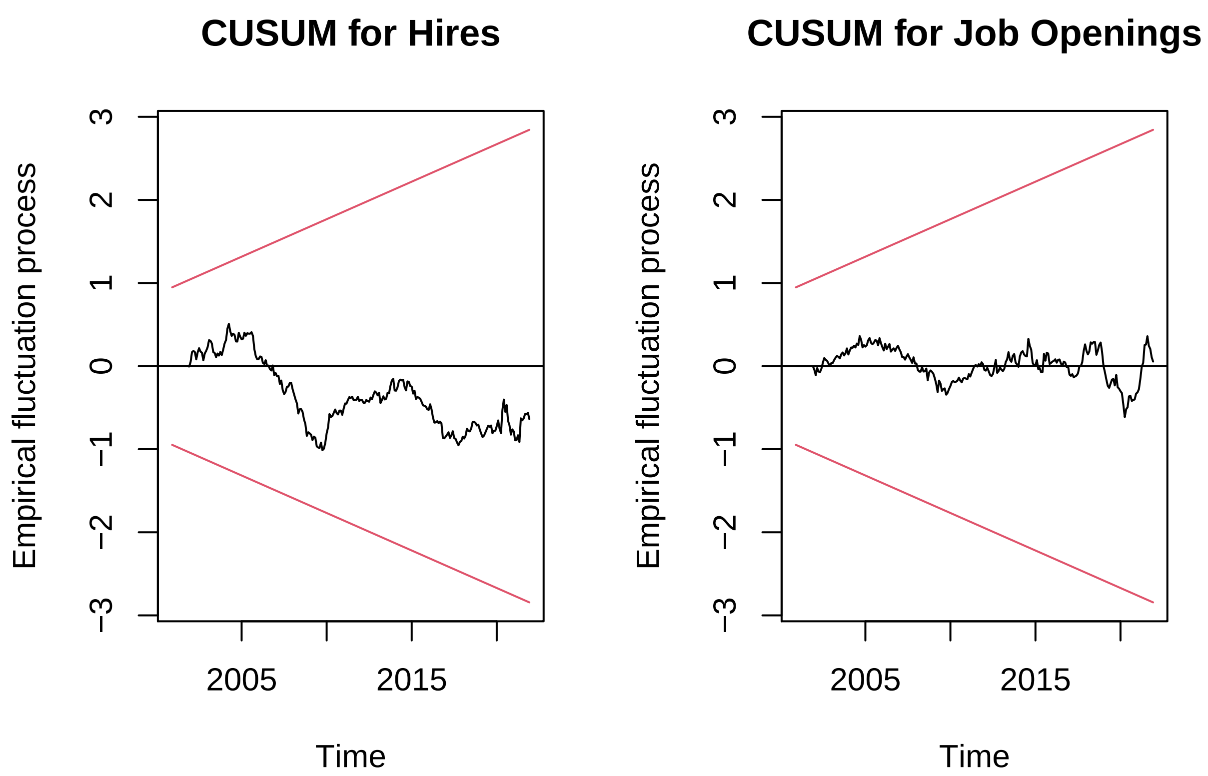
From the ACF and PACF plots of the hires model residuals, we can see that our model failed to capture all of the variation in the series since there is a strong significant spikes in the middle of both plots. There may be some dynamics left in the residuals that we can look into. The ACF and PACF plots of the job openings model residuals, however, shows that we did a good job at fitting the series, with the residuals being almost clear from any patterns and dynamics. Although a few spikes come close to the error bands, since we don't have strong significant peaks we conclude the errors are random.

(g) Plot the respective CUSUM and interpret the plot.

```
# Compute the recursive CUSUM for both models
cusum_hires = efp(totmod_hires$res ~ 1, type = "Rec-CUSUM")
cusum_open = efp(totmod_open$res ~ 1, type = "Rec-CUSUM")

# Plot the recursive CUSUMs together
```

```r
par(mfrow = c(1,2))
plot(cusum_hires, main = "CUSUM for Hires")
plot(cusum_open, main = "CUSUM for Job Openings")
```



The CUSUM plots for both models indicate that there aren't any structural breaks in either series. The coefficients do not fluctuate over time.

(h) For your model, discuss the associated diagnostic statistics.

```r
# Look at model summaries
summary(totmod_hires)
```

```
## Series: hires_ts
## ARIMA(1,1,1)(2,1,0)[12]
##
## Coefficients:
##          ar1      ma1     sar1     sar2
##       0.1718  -0.7053  -0.5055  -0.2244
## s.e.  0.1038   0.0719   0.0705   0.0745
##
## sigma^2 estimated as 3422:  log likelihood=-1317.07
## AIC=2644.13   AICc=2644.39   BIC=2661.54
##
```

```
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 3.664201 56.49942 41.30076 0.3754036 5.837346 0.6717157
##                     ACF1
## Training set 0.005796133
```

```r
summary(totmod_open)
```

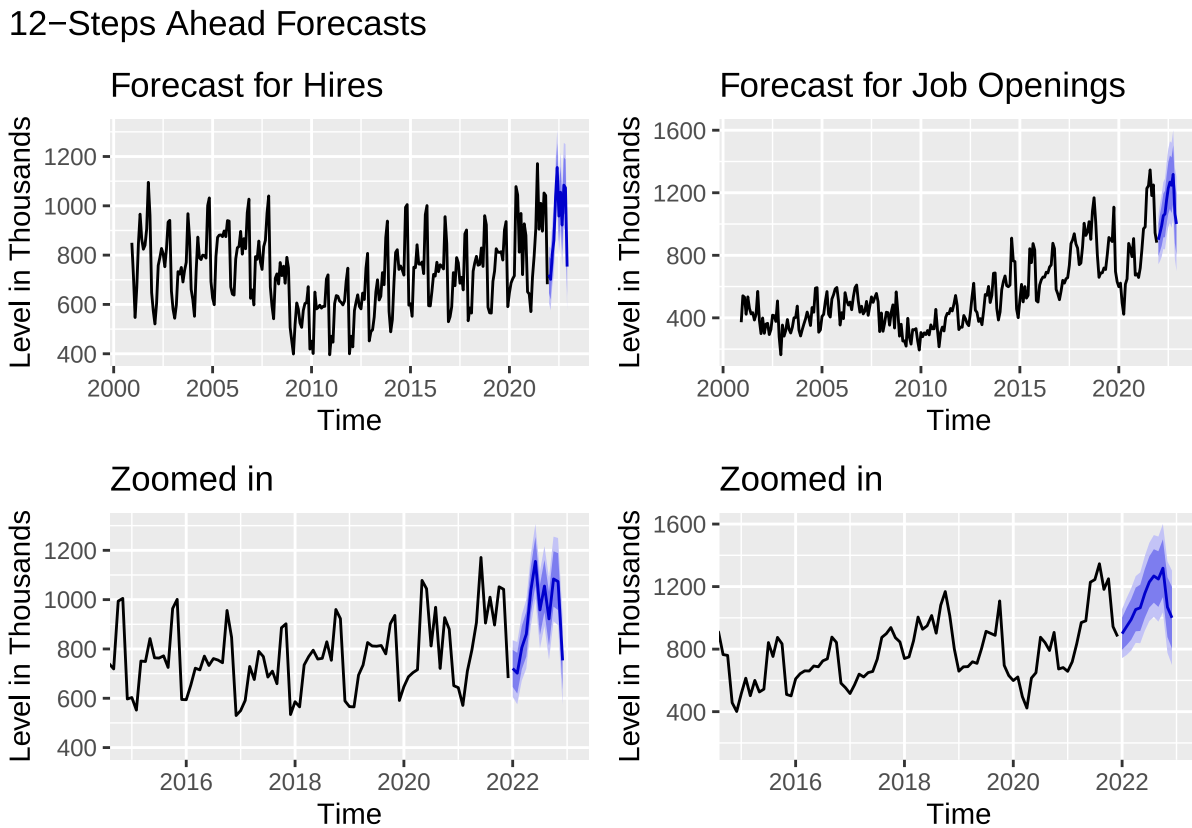
```
## Series: open_ts
## ARIMA(1,1,1)(2,1,1)[12]
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1
##       0.2322  -0.6294  0.0926  -0.0500  -0.7913
## s.e.  0.1447   0.1164  0.1001   0.0878   0.0811
##
## sigma^2 estimated as 6368:  log likelihood=-1394.7
## AIC=2801.4   AICc=2801.76   BIC=2822.28
##
## Training set error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 4.729094 76.91098 57.44201 -0.2757423 11.26551 0.5567698
##                     ACF1
## Training set -0.00577841
```

(i) Use your model to forecast 12-steps ahead. Your forecast should include the respective error bands.

```r
# Store the 12-step-ahead forecasts for both series
frc_hires = forecast(totmod_hires, h=12)
frc_open = forecast(totmod_open, h=12)

# Store the individual forecast plots and the zoomed in versions to plot together
frc_hires_p1 = autoplot(frc_hires) +
  labs(y = "Level in Thousands", title = "Forecast for Hires")
frc_hires_p2 = autoplot(frc_hires) +
  labs(y = "Level in Thousands", title = "Zoomed in") +
  coord_cartesian(xlim = c(2015, 2023))
frc_open_p1 = autoplot(frc_open) +
  labs(y = "Level in Thousands", title = "Forecast for Job Openings")
frc_open_p2 = autoplot(frc_open) +
  labs(y = "Level in Thousands", title = "Zoomed in") +
  coord_cartesian(xlim = c(2015, 2023))

# Display original forecast plots along with the zoomed in versions
(frc_hires_p1 + frc_open_p1) / (frc_hires_p2 + frc_open_p2) +
  plot_annotation(title = "12-Steps Ahead Forecasts")
```

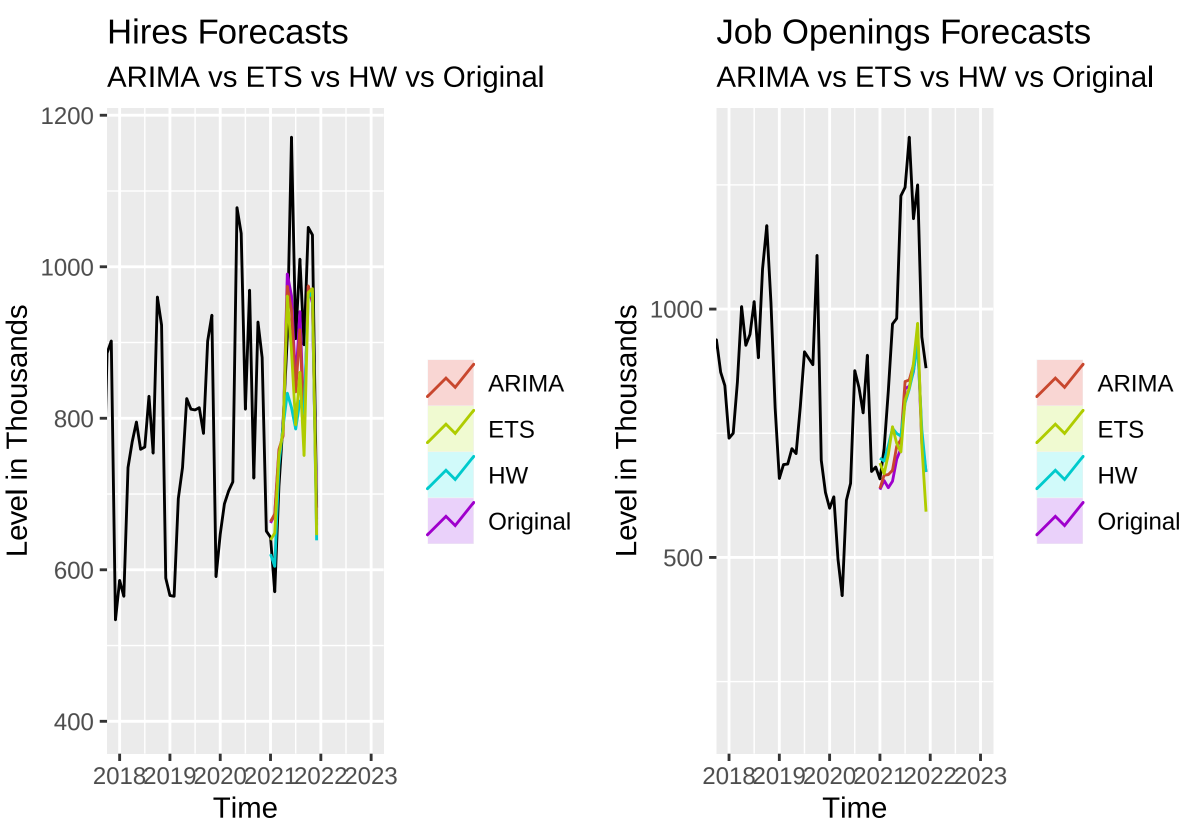


(j) Compare your forecast from (i) to the 12-steps ahead forecasts from ARIMA, Holt-Winters, and ETS models. Which model performs best in terms of MAPE?

```
train_hires = window(hires_ts, end=c(2020,12))
train_open = window(open_ts, end=c(2020,12))
test_hires = window(hires_ts, start = 2021)
test_open = window(open_ts, start = 2021)
#Build forecasts and save the plots
#------------------HIRES------------------
og_frc_hires = forecast(train_hires, h=12, model = totmod_hires)
arima_frc_hires = forecast(auto.arima(train_hires), h=12)
hw_frc_hires = hw(train_hires, h=12)
ets_frc_hires = forecast(ets(train_hires), h=12)
comp_hires_p = autoplot(hires_ts) +
  autolayer(og_frc_hires, series = "Original", PI = F) +
  autolayer(arima_frc_hires, series = "ARIMA", PI = F) +
  autolayer(hw_frc_hires, series = "HW", PI = F) +
  autolayer(ets_frc_hires, series = "ETS", PI = F) +
  coord_cartesian(xlim = c(2018, 2023)) +
  labs(y = "Level in Thousands", title = "Hires Forecasts",
       subtitle = "ARIMA vs ETS vs HW vs Original") +
  scale_color_discrete(name = "")
#-----------------JOB OPENINGS-------------
og_frc_open = forecast(train_open, h=12, model = totmod_open)
```

```r
arima_frc_open = forecast(auto.arima(train_open), h=12)
hw_frc_open = hw(train_open, h=12)
ets_frc_open = forecast(ets(train_open), h=12)
comp_open_p = autoplot(open_ts) +
  autolayer(og_frc_open, series = "Original", PI = F) +
  autolayer(arima_frc_open, series = "ARIMA", PI = F) +
  autolayer(hw_frc_open, series = "HW", PI = F) +
  autolayer(ets_frc_open, series = "ETS", PI = F) +
  coord_cartesian(xlim = c(2018, 2023)) +
  labs(y = "Level in Thousands", title = "Job Openings Forecasts",
       subtitle = "ARIMA vs ETS vs HW vs Original") +
  scale_color_discrete(name = "")

# Display the plots together
comp_hires_p + comp_open_p
```




```r
# Get the forecast diagnostics of the individual forecasts and combine them
#------------------HIRES----------------------------
acc_hires = rbind(accuracy(og_frc_hires, test_hires)["Test set", "MAPE"],
                  accuracy(arima_frc_hires, test_hires)["Test set", "MAPE"],
                  accuracy(hw_frc_hires, test_hires)["Test set", "MAPE"],
                  accuracy(ets_frc_hires, test_hires)["Test set", "MAPE"])
rownames(acc_hires) = c("Original", "ARIMA", "HW", "ETS")
#------------------JOB OPENINGS-----------------------
acc_open = rbind(accuracy(og_frc_open, test_open)["Test set", "MAPE"],
```

```
                 accuracy(arima_frc_open, test_open)["Test set", "MAPE"],
                 accuracy(hw_frc_open, test_open)["Test set", "MAPE"],
                 accuracy(ets_frc_open, test_open)["Test set", "MAPE"])
rownames(acc_open) = c("Original", "ARIMA", "HW", "ETS")

# Print the results in tables
kable(acc_hires, digits = 4, caption = "MAPE of Forecasts for Hires") %>%
  kable_styling(latex_options = "HOLD_position")  # Original has lower MAPE
```

Table 1: MAPE of Forecasts for Hires

Original	8.2842
ARIMA	8.8461
HW	9.7540
ETS	9.6310

```
kable(acc_open, digits = 4, caption = "MAPE of Forecasts for Job Openings") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 2: MAPE of Forecasts for Job Openings

Original	25.4589
ARIMA	24.0751
HW	22.9544
ETS	24.2387

```
  # HW has lower MAPE
```

According to the tables above the lowest MAPE belongs to the forecast using our original model for hires. The lowest MAPE for job openings belongs to the Holt-Winters forecast.

(k) Combine the four forecasts and comment on the MAPE from this forecasts vs., the individual ones.

```
# Combine forecasts by taking their average
combine_hires = (arima_frc_hires$mean + ets_frc_hires$mean +
                   hw_frc_hires$mean + og_frc_hires$mean)/4
combine_open = (arima_frc_open$mean + ets_frc_open$mean+
                  hw_frc_open$mean + og_frc_open$mean)/4
# Get the forecast diagnostics of the individual forecasts and combine them
#--------------------HIRES----------------------------
acc_hires2 = rbind(accuracy(og_frc_hires, test_hires)["Test set", "MAPE"],
                  accuracy(arima_frc_hires, test_hires)["Test set", "MAPE"],
                  accuracy(hw_frc_hires, test_hires)["Test set", "MAPE"],
                  accuracy(ets_frc_hires, test_hires)["Test set", "MAPE"],
                  accuracy(combine_hires, test_hires)["Test set", "MAPE"])
rownames(acc_hires2) = c("Original", "ARIMA", "HW", "ETS", "Combination")
```

```r
#------------------JOB OPENINGS-----------------------
acc_open2 = rbind(accuracy(og_frc_open, test_open)["Test set", "MAPE"],
                  accuracy(arima_frc_open, test_open)["Test set", "MAPE"],
                  accuracy(hw_frc_open, test_open)["Test set", "MAPE"],
                  accuracy(ets_frc_open, test_open)["Test set", "MAPE"],
                  accuracy(combine_open, test_open)["Test set", "MAPE"])
rownames(acc_open2) = c("Original", "ARIMA", "HW", "ETS", "Combination")

# Print the results in tables
kable(acc_hires2, digits = 4, caption = "MAPE of Forecasts for Hires") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 3: MAPE of Forecasts for Hires

Original	8.2842
ARIMA	8.8461
HW	9.7540
ETS	9.6310
Combination	8.6206

```r
kable(acc_open2, digits = 4, caption = "MAPE of Forecasts for Job Openings") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 4: MAPE of Forecasts for Job Openings

Original	25.4589
ARIMA	24.0751
HW	22.9544
ETS	24.2387
Combination	23.9252

From the table above, we conclude that the combination forecast for hires does not have a lower MAPE score than our forecast with the original model. Similarly, the lowest MAPE for job openings does not belong to our combination forecast, so we determine that the combination did not improve our forecasts.

(l) Fit an appropriate VAR model using your two variables. Make sure to show the relevant plots and discuss your results from the fit.

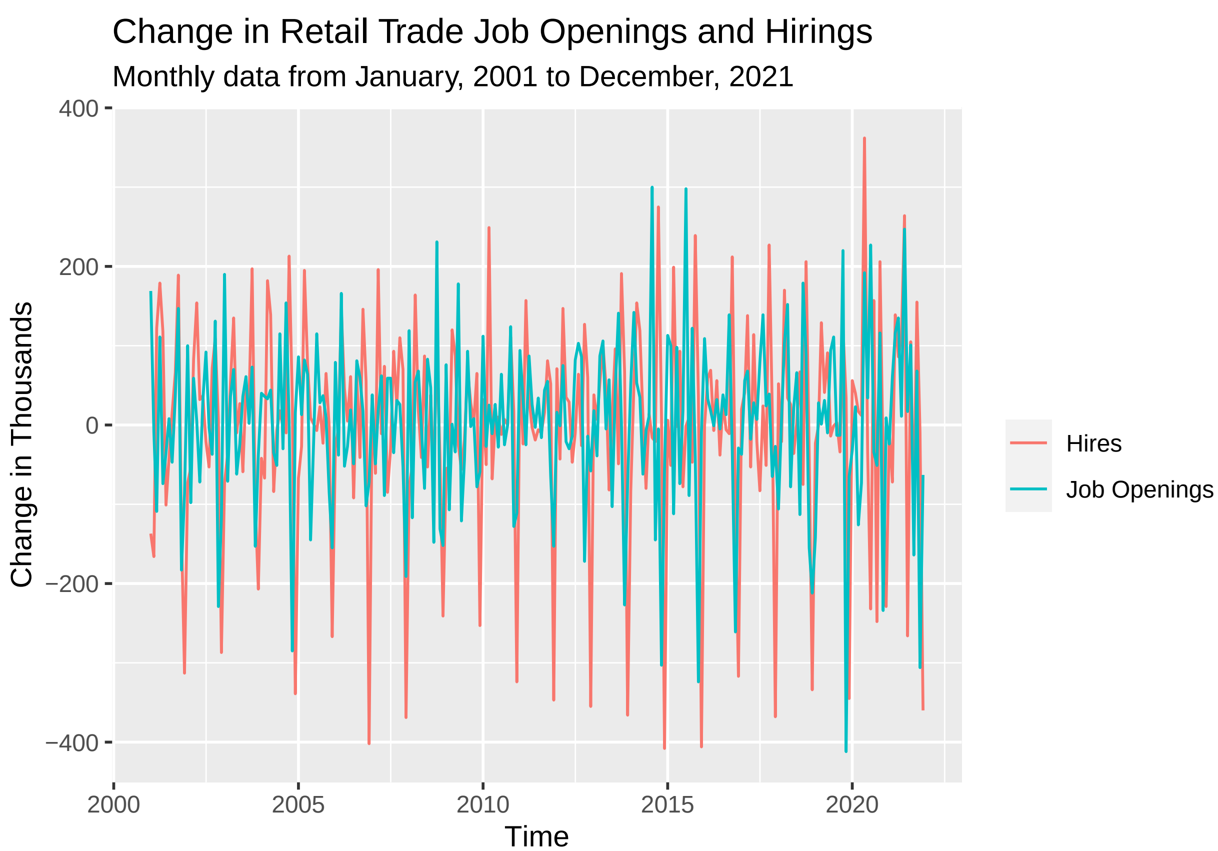
Since we know our series are non-stationary we need to take the first difference to continue with the VAR analysis.

```r
# Take the first difference of both series
d.hires_ts = diff(hires_ts)
d.open_ts = diff(open_ts)
```

```r
# Create a data frame containing both differenced series
df2 = data.frame("d.hires" = d.hires_ts, "d.open" = d.open_ts,
                date = seq(as.Date("2001-01-01"), by = "1 month", length.out = 252) )
```

```
# Convert the data frame into a molten data frame indexing by date
df_long2 = melt(df2, id="date")

# Plot the time series
ggplot(data=df_long2,
       aes(x=date, y=value, colour=variable))+
       geom_line() +
  labs(x="Time", y="Change in Thousands",
       title = "Change in Retail Trade Job Openings and Hirings",
       subtitle = "Monthly data from January, 2001 to December, 2021")+
  scale_color_discrete(name = "", breaks = c("d.hires", "d.open"),
                       labels = c("Hires", "Job Openings"))
```



From the plot above, we can see that taking the first difference of the series made them stationary. Now we can combine them in a VAR model. We will use VARselect to determine the order of the model.

```
# Combine two series in a data frame
y_ts = ts.union(d.hires_ts, d.open_ts, dframe = T)

# Use VARselect to determine the order of the model
VARselect(y_ts)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10     10     10     10
```

```
##
## $criteria
##                   1            2            3            4            5
## AIC(n) 1.872645e+01 1.871251e+01 1.859500e+01 1.847622e+01 1.840815e+01
## HQ(n)  1.876129e+01 1.877059e+01 1.867631e+01 1.858076e+01 1.853592e+01
## SC(n)  1.881295e+01 1.885668e+01 1.879684e+01 1.873573e+01 1.872533e+01
## FPE(n) 1.357669e+08 1.338894e+08 1.190472e+08 1.057181e+08 9.876746e+07
##                   6            7            8            9           10
## AIC(n) 1.837359e+01 1.835986e+01 1.830140e+01 1.823579e+01 1.803624e+01
## HQ(n)  1.852459e+01 1.853409e+01 1.849886e+01 1.845648e+01 1.828016e+01
## SC(n)  1.874843e+01 1.879237e+01 1.879158e+01 1.878364e+01 1.864176e+01
## FPE(n) 9.541948e+07 9.412879e+07 8.879681e+07 8.317323e+07 6.814265e+07
```

All of the selection criterion according to the VARselect choose the order to be 10 for the model.

```r
var_model = VAR(y_ts, p=10)
summary(var_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: d.hires_ts, d.open_ts
## Deterministic variables: const
## Sample size: 242
## Log Likelihood: -2827.151
## Roots of the characteristic polynomial:
## 0.9692 0.9692 0.968 0.968 0.9162 0.9162 0.9014 0.9014 0.8965 0.8965 0.8688 0.8688 0.8678 0.8678 0.84
## Call:
## VAR(y = y_ts, p = 10)
##
##
## Estimation results for equation d.hires_ts:
## ===========================================
## d.hires_ts = d.hires_ts.l1 + d.open_ts.l1 + d.hires_ts.l2 + d.open_ts.l2 + d.hires_ts.l3 + d.open_ts
##
##                Estimate Std. Error t value Pr(>|t|)
## d.hires_ts.l1  -0.590603   0.064550  -9.150  < 2e-16 ***
## d.open_ts.l1    0.421611   0.073284   5.753 2.90e-08 ***
## d.hires_ts.l2  -0.549701   0.071967  -7.638 6.60e-13 ***
## d.open_ts.l2    0.238513   0.083115   2.870 0.004508 **
## d.hires_ts.l3  -0.545508   0.077952  -6.998 3.06e-11 ***
## d.open_ts.l3    0.282376   0.083616   3.377 0.000866 ***
## d.hires_ts.l4  -0.543196   0.083815  -6.481 5.86e-10 ***
## d.open_ts.l4    0.111163   0.085178   1.305 0.193229
## d.hires_ts.l5  -0.375733   0.085540  -4.392 1.74e-05 ***
## d.open_ts.l5    0.049776   0.086626   0.575 0.566142
## d.hires_ts.l6  -0.438400   0.084081  -5.214 4.24e-07 ***
## d.open_ts.l6    0.005402   0.086499   0.062 0.950258
## d.hires_ts.l7  -0.286996   0.079882  -3.593 0.000403 ***
## d.open_ts.l7    0.048616   0.086266   0.564 0.573623
## d.hires_ts.l8  -0.336152   0.073654  -4.564 8.33e-06 ***
## d.open_ts.l8    0.057920   0.084903   0.682 0.495832
## d.hires_ts.l9  -0.388470   0.069703  -5.573 7.25e-08 ***
```

```
## d.open_ts.l9    0.064351   0.082746   0.778 0.437582
## d.hires_ts.l10 -0.380451   0.063016  -6.037 6.55e-09 ***
## d.open_ts.l10   0.216721   0.076551   2.831 0.005067 **
## const          -3.568824   6.053386  -0.590 0.556089
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 92.66 on 221 degrees of freedom
## Multiple R-Squared: 0.5533,  Adjusted R-squared: 0.5129
## F-statistic: 13.69 on 20 and 221 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation d.open_ts:
## =========================================
## d.open_ts = d.hires_ts.l1 + d.open_ts.l1 + d.hires_ts.l2 + d.open_ts.l2 + d.hires_ts.l3 + d.open_ts.l
##
##                Estimate Std. Error t value Pr(>|t|)
## d.hires_ts.l1  -0.20995    0.05860  -3.583 0.000418 ***
## d.open_ts.l1   -0.32967    0.06653  -4.955 1.44e-06 ***
## d.hires_ts.l2  -0.05681    0.06534  -0.869 0.385550
## d.open_ts.l2   -0.23237    0.07546  -3.080 0.002336 **
## d.hires_ts.l3  -0.16712    0.07077  -2.361 0.019072 *
## d.open_ts.l3   -0.14407    0.07591  -1.898 0.059022 .
## d.hires_ts.l4  -0.16232    0.07609  -2.133 0.034014 *
## d.open_ts.l4   -0.18898    0.07733  -2.444 0.015316 *
## d.hires_ts.l5  -0.21633    0.07766  -2.786 0.005806 **
## d.open_ts.l5   -0.17655    0.07864  -2.245 0.025762 *
## d.hires_ts.l6  -0.27705    0.07633  -3.629 0.000353 ***
## d.open_ts.l6   -0.03693    0.07853  -0.470 0.638652
## d.hires_ts.l7  -0.35210    0.07252  -4.855 2.27e-06 ***
## d.open_ts.l7   -0.09666    0.07832  -1.234 0.218445
## d.hires_ts.l8  -0.36420    0.06687  -5.447 1.36e-07 ***
## d.open_ts.l8   -0.17474    0.07708  -2.267 0.024356 *
## d.hires_ts.l9  -0.31595    0.06328  -4.993 1.21e-06 ***
## d.open_ts.l9   -0.07151    0.07512  -0.952 0.342174
## d.hires_ts.l10 -0.25883    0.05721  -4.524 9.89e-06 ***
## d.open_ts.l10  -0.07580    0.06950  -1.091 0.276574
## const           6.45776    5.49560   1.175 0.241228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 84.13 on 221 degrees of freedom
## Multiple R-Squared: 0.4128,  Adjusted R-squared: 0.3596
## F-statistic: 7.767 on 20 and 221 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##            d.hires_ts d.open_ts
## d.hires_ts       8587      1753
## d.open_ts        1753      7077
##
```

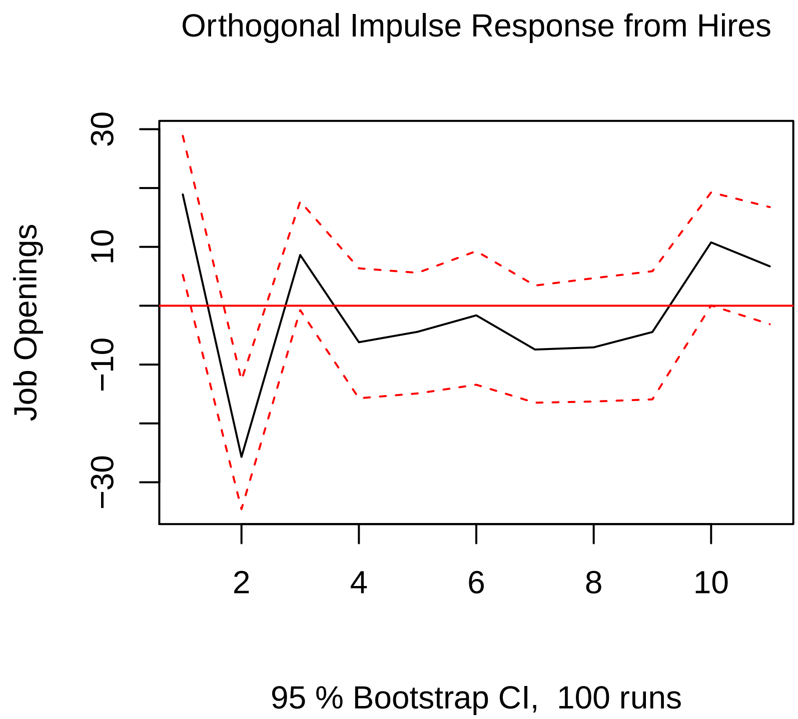
```
## Correlation matrix of residuals:
##            d.hires_ts d.open_ts
## d.hires_ts     1.0000    0.2249
## d.open_ts      0.2249    1.0000
```

In a VAR model we relate the current observations of the variable we are trying to predict to past observations of itself and past observations of other variables if our model is multivariate. For the hires model, we can see that all lags of hires and job openings from lag1 to lag3 including lag3 are significant. After lag3, lags of open are not significant while hires still are. This in a way indicates that we can use up to 3 lags of Job Openings data to predict Job Hires. This makes sense, if there are more job openings in time t-k, chances are more people will get hired in time k where k is the average time it takes for a company to fill a position. K can range from weeks to months and this is what the data suggest us. In the job openings model the significance of job openings coefficients are more complicated. Lag1 coefficients are significant for both models, many of the hire lags are also significant while opening lags are not. This suggests that we can use the lags of job hires data to predict new job openings. So, the way people get hired in the past effect the number of job openings today. This probably has something to do with the long term structures of the economies. If economies are doing well and more people get hired (fired) in time t-k chances are there will be more (less) job openings in time t because economy is doing well. However, I would not look beyond lag 6 because until lag 6 all values are significant in some way except lag 2 of hires. After lag 6 there are many insignificant values.

(m) Compute, plot, and interpret the respective impulse response functions.

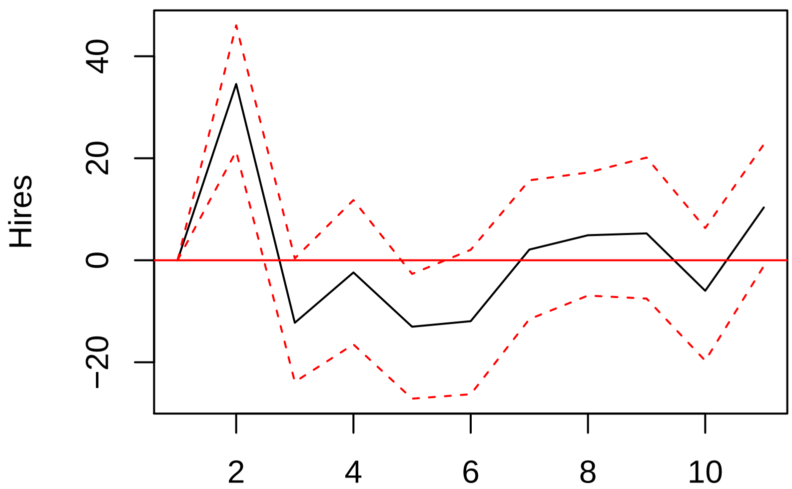
```r
# Compute the impulse response functions
IRF1 = irf(var_model, impulse = "d.hires_ts", response = "d.open_ts")
IRF2 = irf(var_model, impulse = "d.open_ts", response = "d.hires_ts")

# Plote the IRFs
plot(IRF1, main = "Orthogonal Impulse Response from Hires",
     ylab = "Job Openings")
```




```r
plot(IRF2, main = "Orthogonal Impulse Response from Job Openings",
     ylab = "Hires")
```

Orthogonal Impulse Response from Job Openings



95 % Bootstrap CI, 100 runs

From the plots above, we can see that a unit shock in hires has a direct effect on job openings, though it dies out quickly. This is supported from our var model where the results showed that the first lag was highly significant. On the other hand, a shock in job openings has a longer lasting effect on hires, which also supports the conclusions we made from our VAR model.

(n) Perform a Granger-Causality test on your variables and discuss your results from the test.

```
# Test for Job Openings granger-causes Hires
grangertest(d.hires_ts ~ d.open_ts, order = 10)
```

```
## Granger causality test
##
## Model 1: d.hires_ts ~ Lags(d.hires_ts, 1:10) + Lags(d.open_ts, 1:10)
## Model 2: d.hires_ts ~ Lags(d.hires_ts, 1:10)
##   Res.Df  Df      F    Pr(>F)
## 1    221
## 2    231 -10 5.1275 9.731e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

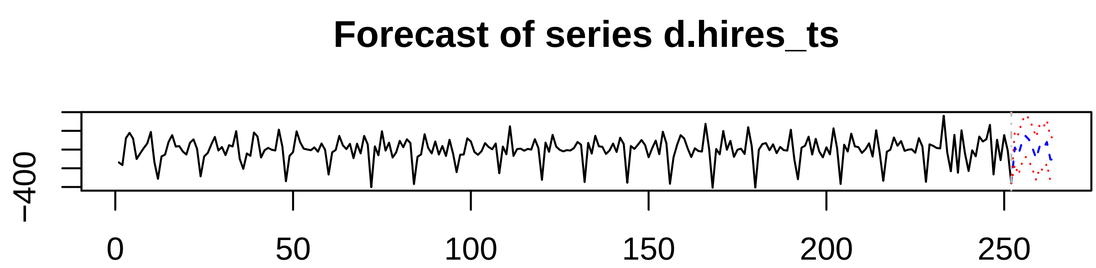
```
# Test for Hires granger-causes Job Openings
grangertest(d.open_ts ~ d.hires_ts, order = 10)
```

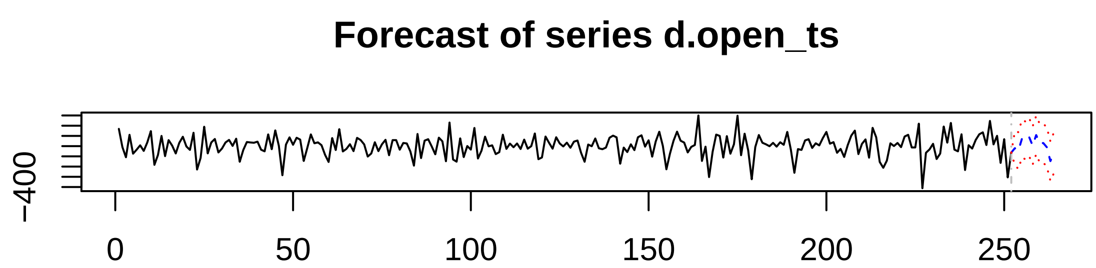
```
## Granger causality test
##
## Model 1: d.open_ts ~ Lags(d.open_ts, 1:10) + Lags(d.hires_ts, 1:10)
## Model 2: d.open_ts ~ Lags(d.open_ts, 1:10)
##   Res.Df  Df      F    Pr(>F)
## 1    221
## 2    231 -10 5.8265 8.733e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results of the Granger-Causality tests indicate that both hires and job openings granger cause one another, indicating a bi-directional explanatory relationship between the two variables.

(o) Use your VAR model to forecast 12-steps ahead. Your forecast should include the respective error bands. Comment on the differences between the VAR forecast and the other ones obtained using the different methods.

```
var.predict = predict(object = var_model, n.ahead = 12)
plot(var.predict)
```





The forecasts from the VAR model look completely different than the previous forecasts. This is because our VAR model was created through taking the first difference of the data, which made it stationary. The VAR forecasts therefore show forecasts of the change in the series. The respective error bands of the series forecasts seem to be really wide for both of them, which shows that our forecasts may not be so accurate.

Conclusions and Future Work

Our initial assumption was that Retail Trade Job Openings would cause more people to get hired. We assumed that this relationship would be uni-directional, with job openings affecting hires. This was based on the intuition that more job openings would lead to more hires, whereas more hires would not necessarily lead to more job openings. However, we ended up finding a bi-directional relationship between the two variables, which suggests that the effect goes both ways. It was interesting to see that a unit shock in hires leads to an immediate decrease in job openings, since we had assumed it would not have that much of an effect at all. The finding that a unit shock in job openings leads to a longer lasting decrease in hires was expected, however, as less job openings would intuitively lead to less hires.

Future work should focus on replicating the results for different industries to see whether this relationship holds. Different industries have different demand structure and its likely that they will exhibit different behavior when it comes to job openings and hires. We know that retail and tourism industries have highly seasonal pattern, but technology, or banking probably do not exhibit a similar seasonal pattern as their success lies more in long term trends and business cycles of the economy. In the future we can try to understand these differences better and come up with unique models for individual sectors of the economy, or even individual firms if there is enough data. Furthermore, we used an seasonal ARIMA model when fitting the series with a model that contained trend, seasonality, and cycles components. Other estimation procedures can be applied to see if they model the behaviors more accurately.

References

U.S. Bureau of Labor Statistics, Hires: Retail Trade [JTU4400HIL], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/JTU4400HIL, February 21, 2022.

U.S. Bureau of Labor Statistics, Job Openings: Retail Trade [JTU4400JOL], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/JTU4400JOL, February 21, 2022.