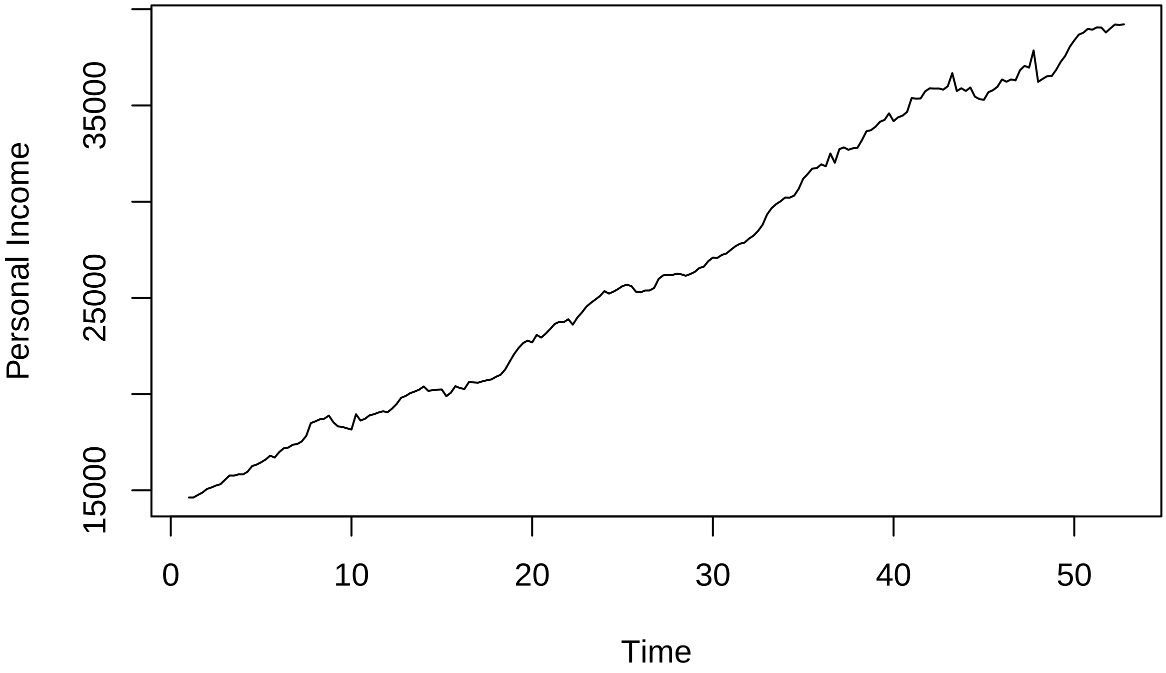
Assignment 2

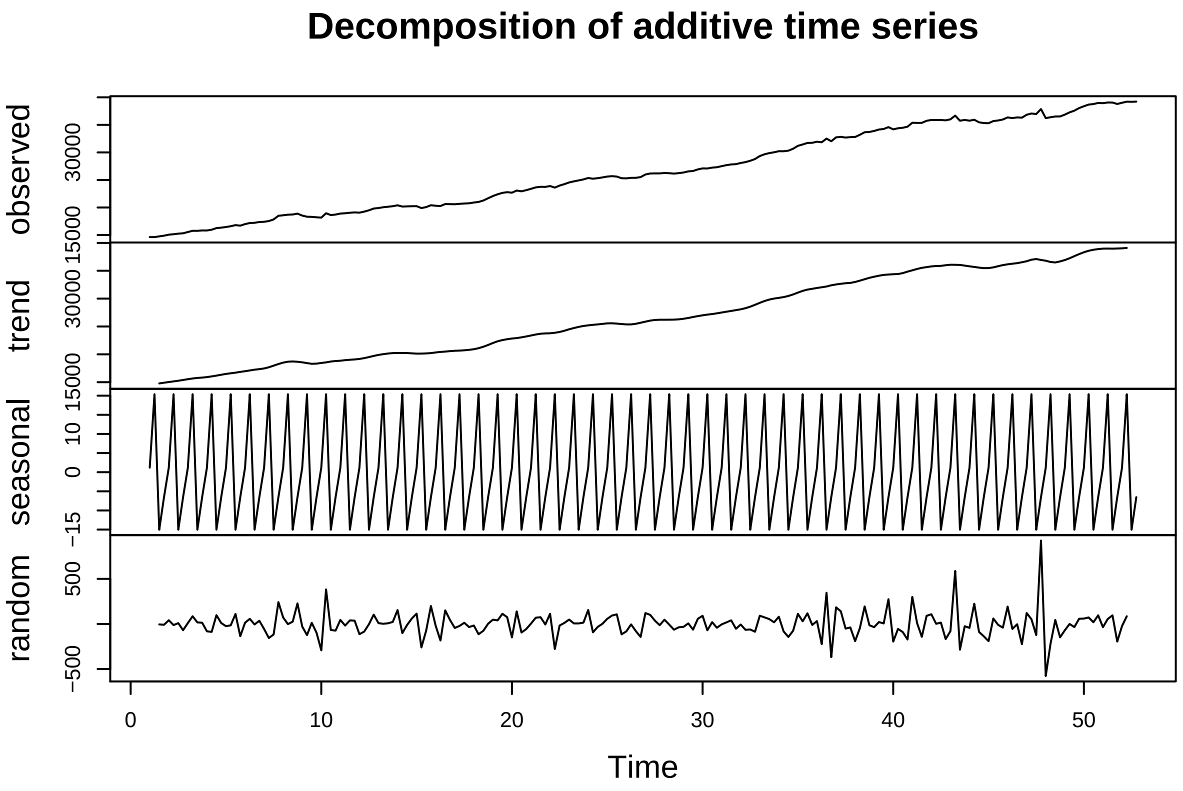
Question 1: Model selection 5.5 points

Plot the series and comment on how it evolves over time.



As we can see from the above graph, the personal income seems to increase over time suggesting that there might be some sort of trend that is happening. To verify this, I decide to decompose the data to see if there is indeed any trend or seasonailty factor that is at play on the data.

```
personalincomedecomposed <- decompose(personalincome)
plot(personalincomedecomposed)
```



According to the above graph, our time series data is affected by an upward trend which I believe will only require one difference to make the time series stationary. To verify my assumption I use the ndiff command that estimates the number of differences required to make a given time series stationary depending on the test that I select. In this case I use the "adf" to specify that I want to test against the Augmented Dickey-Fuller test.

```
forecast::ndiffs(personalincome, test = "adf")
```

```
## [1] 1
```

```
DF <- tseries::adf.test(personalincome, k = 0)
ADF2 <- tseries::adf.test(personalincome, k = 2)
ADF <- tseries::adf.test(personalincome)
DF
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  personalincome
## Dickey-Fuller = -2.3392, Lag order = 0, p-value = 0.4339
## alternative hypothesis: stationary
```

```
ADF2
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  personalincome
## Dickey-Fuller = -2.1312, Lag order = 2, p-value = 0.521
## alternative hypothesis: stationary
```

```
ADF
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  personalincome
## Dickey-Fuller = -2.111, Lag order = 5, p-value = 0.5295
## alternative hypothesis: stationary
```

In the above block of code, I perform both an Dickey-Fuller (k=0) test, as for k = 0 the standard Dickey-Fuller test is computed, and the Augmented Dickey-Fuller test, at both k=2 and at k = $trunc((length(x)-1)^(1/3))$ the suggested upper bound on the number of lags that should be used. In each case, we get a p-value that is greater than 5%, meaning that we fail to reject the null hypothesis that there is a unit root.

As I have hinted previously, the way that I would continue to work with the series is by taking the difference of the series as many times as required to ensure that the time series becomes stationary. As shown above, if we were to use the ADF testing method, I would only need to take one difference of the time series in order to make it stationary. To verify this claim, I run another series of DF and ADF tests, with the same k-values to see if we reject the null hypothesis that there is a unit root.

```
diff1dat <- diff(personalincome)
DF <- tseries::adf.test(diff1dat, k = 0)
```

```
## Warning in tseries::adf.test(diff1dat, k = 0): p-value smaller than printed p-
## value
```

```
ADF2 <- tseries::adf.test(diff1dat, k = 2)
```

```
## Warning in tseries::adf.test(diff1dat, k = 2): p-value smaller than printed p-
## value
```

```
ADF <- tseries::adf.test(diff1dat)
```

```
## Warning in tseries::adf.test(diff1dat): p-value smaller than printed p-value
```

```
DF
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff1dat
## Dickey-Fuller = -17.096, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

```
ADF2
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff1dat
## Dickey-Fuller = -7.4854, Lag order = 2, p-value = 0.01
## alternative hypothesis: stationary
```

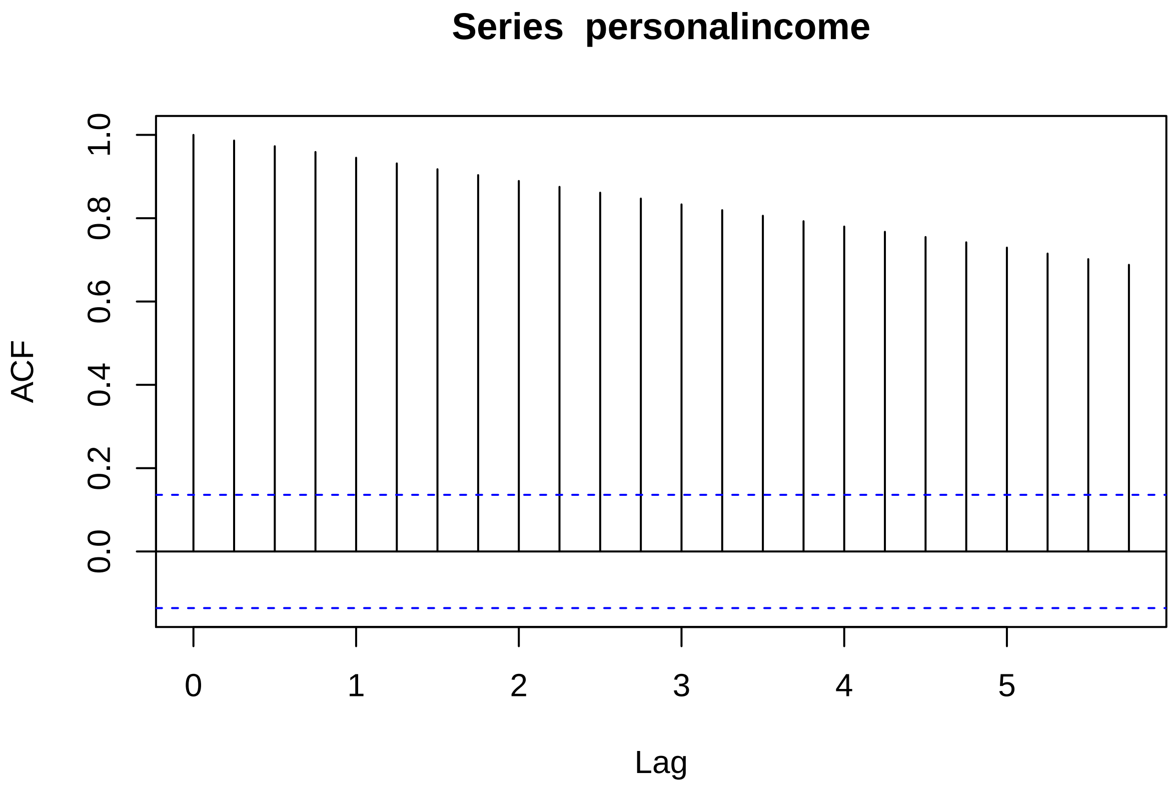
```
ADF
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff1dat
## Dickey-Fuller = -6.2674, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

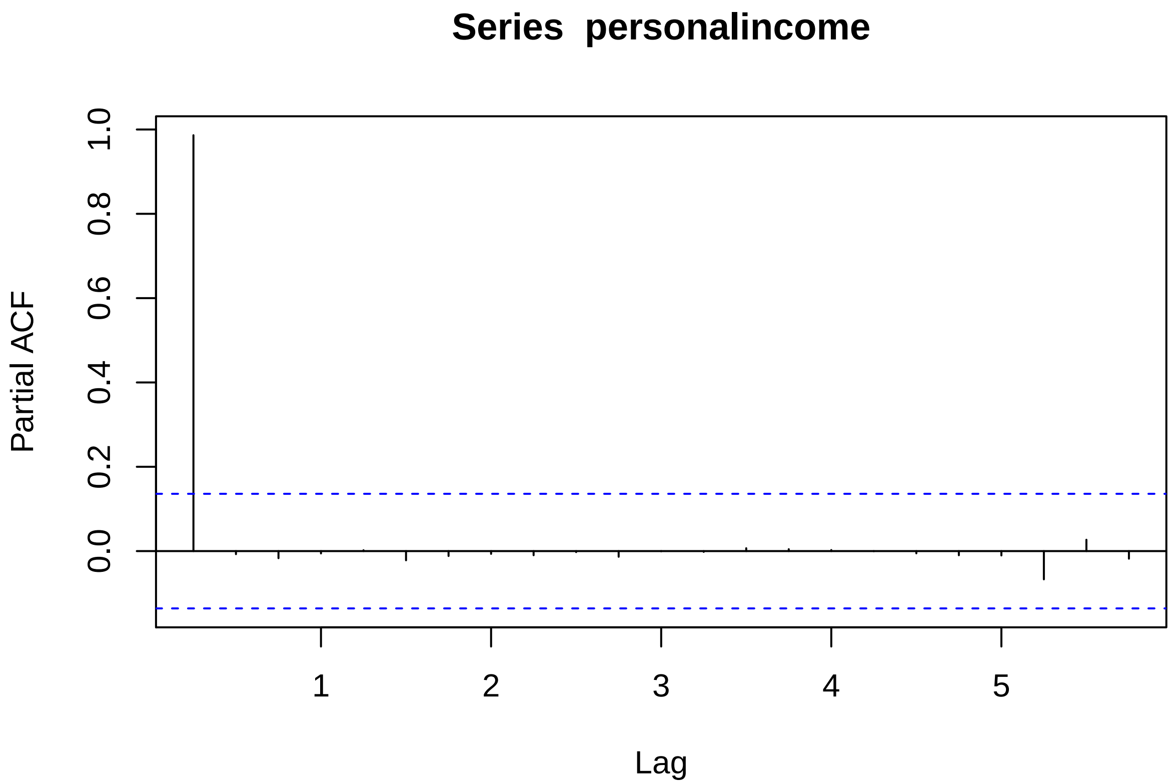
We have verified that the p-value indicates that we reject the null hypothesis that there is a unit root in our transformed series.

In the graphs below, I plotting both the ACF and PACF on the original and transformed series.

```
acforiginal <- acf(personalincome)
```

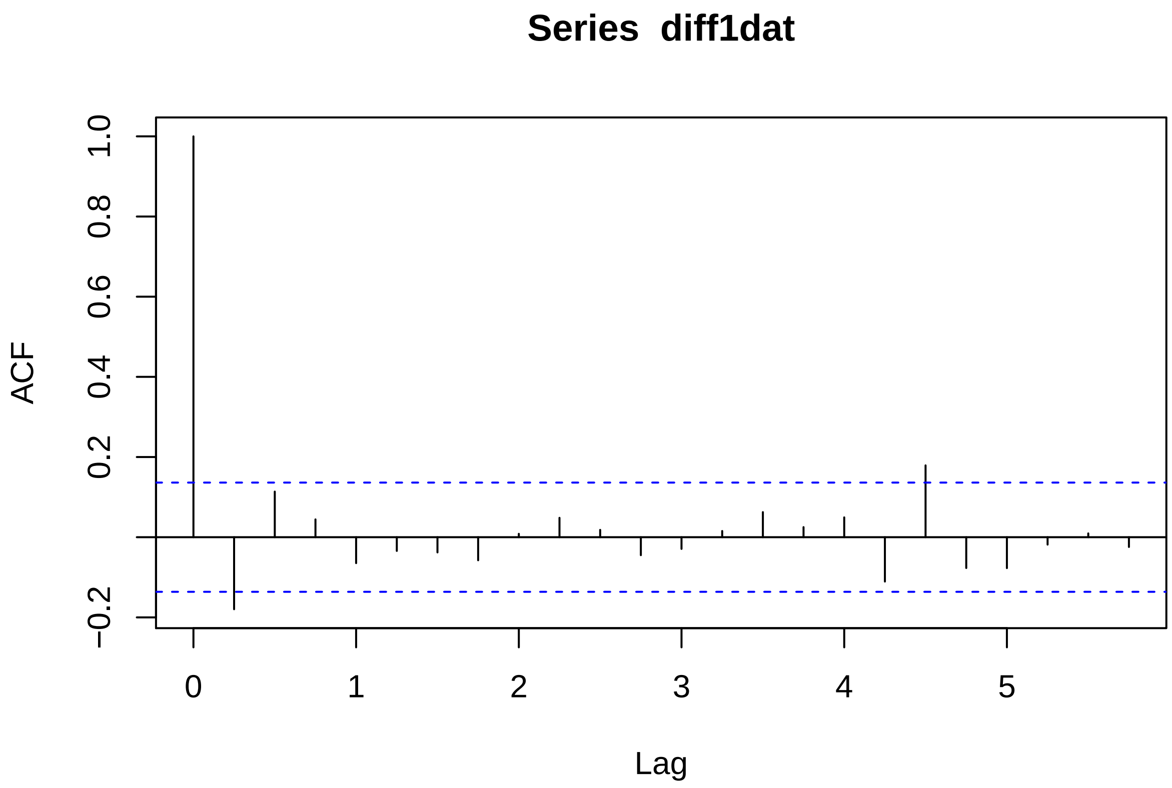


```
paforiginal <- pacf(personalincome)
```

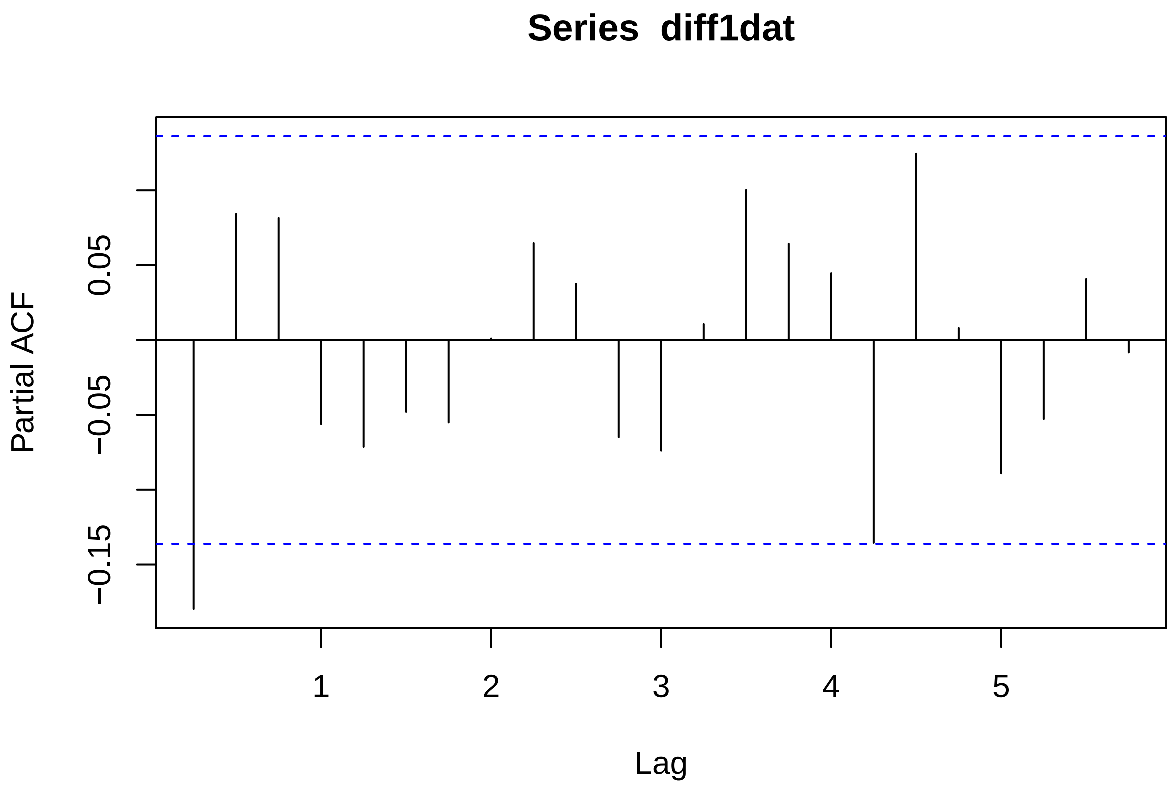


According to the ACF and PACF plots, when it come to the original series, it seems that the model that would best fit this series is an AR(1) model as the PACF seems to die off rather quickly (although not at the first lag). On the other hand, when I plot the ACF and PACF of the the transformed series, the model that I would choose is perhaps an MA(2) model because the ACF seems to die off after the second lag (although the PACF is statistically insignificant and the ACF dies off before the first lag as well).

```
acftransformed <- acf(diff1dat)
```



```
paftransformed <- pacf(diff1dat)
```



In the following block of code, and its output, I am computing the AIC for various models using the auto.arima function which gives us the best ARIMA model according to either AIC, AICc or BIC values. In this part, I have specified the model to show us the various models with their corresponding AIC and giving us the model with the lowest AIC.

```
ARMAAIC <-forecast::auto.arima(diff1dat, ic = c("aic"), trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2)(1,0,1)[4] with non-zero mean : 2885.535
##  ARIMA(0,0,0)           with non-zero mean : 2881.771
##  ARIMA(1,0,0)(1,0,0)[4] with non-zero mean : 2882.766
##  ARIMA(0,0,1)(0,0,1)[4] with non-zero mean : 2879.298
##  ARIMA(0,0,0)           with zero mean     : 2921.086
##  ARIMA(0,0,1)           with non-zero mean : 2878.171
##  ARIMA(0,0,1)(1,0,0)[4] with non-zero mean : 2883.042
##  ARIMA(0,0,1)(1,0,1)[4] with non-zero mean : 2885.011
##  ARIMA(1,0,1)           with non-zero mean : 2879.04
##  ARIMA(0,0,2)           with non-zero mean : 2876.137
##  ARIMA(0,0,2)(1,0,0)[4] with non-zero mean : 2881.132
##  ARIMA(0,0,2)(0,0,1)[4] with non-zero mean : 2877.439
##  ARIMA(0,0,2)(1,0,1)[4] with non-zero mean : 2883.109
##  ARIMA(1,0,2)           with non-zero mean : 2878.731
##  ARIMA(0,0,3)           with non-zero mean : 2877.81
##  ARIMA(1,0,3)           with non-zero mean : 2880.283
##  ARIMA(0,0,2)           with zero mean     : 2910.089
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,0,2)           with non-zero mean : 2876.195
##
##  Best model: ARIMA(0,0,2)           with non-zero mean
```

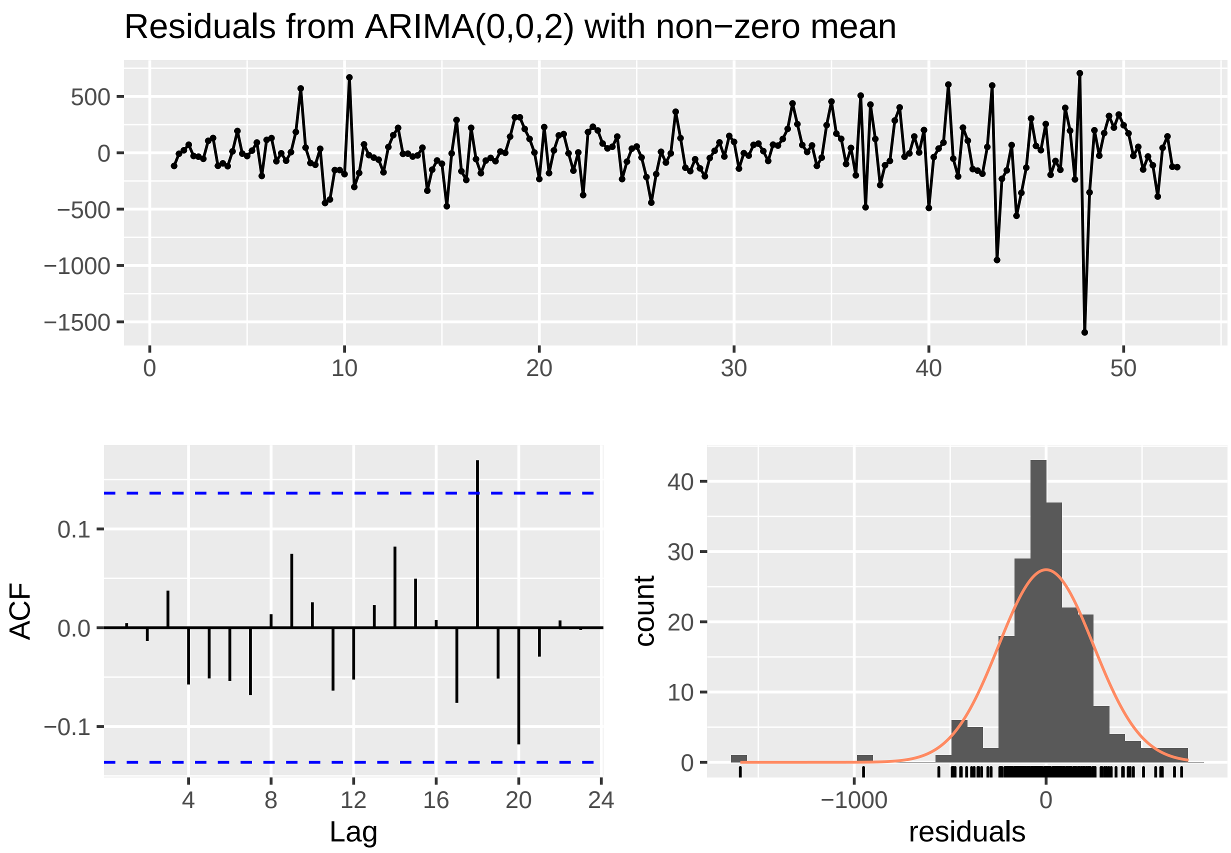
As we can see from the lines of code, the model with the lowest AIC is the MA(2) mode. However, when we use the function specifying the BIC values we get that the AR(1) model would be better, this is to be expected because the BIC tends to prefer models with less lags.

```
ARMABIC <-forecast::auto.arima(diff1dat, ic = c("bic"), trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2)(1,0,1)[4] with non-zero mean : 2912.197
##  ARIMA(0,0,0)           with non-zero mean : 2888.437
##  ARIMA(1,0,0)(1,0,0)[4] with non-zero mean : 2896.097
##  ARIMA(0,0,1)(0,0,1)[4] with non-zero mean : 2892.629
##  ARIMA(0,0,0)           with zero mean     : 2924.418
##  ARIMA(0,0,0)(1,0,0)[4] with non-zero mean : 2896.638
##  ARIMA(0,0,0)(0,0,1)[4] with non-zero mean : 2892.905
##  ARIMA(0,0,0)(1,0,1)[4] with non-zero mean : 2901.956
##  ARIMA(1,0,0)           with non-zero mean : 2887.738
##  ARIMA(1,0,0)(0,0,1)[4] with non-zero mean : 2892.13
##  ARIMA(1,0,0)(1,0,1)[4] with non-zero mean : 2901.429
##  ARIMA(2,0,0)           with non-zero mean : 2892.606
##  ARIMA(1,0,1)           with non-zero mean : 2892.371
##  ARIMA(0,0,1)           with non-zero mean : 2888.17
##  ARIMA(2,0,1)           with non-zero mean : 2897.47
##  ARIMA(1,0,0)           with zero mean     : 2930.509
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,0,0)           with non-zero mean : 2886.995
##
##  Best model: ARIMA(1,0,0)           with non-zero mean
```

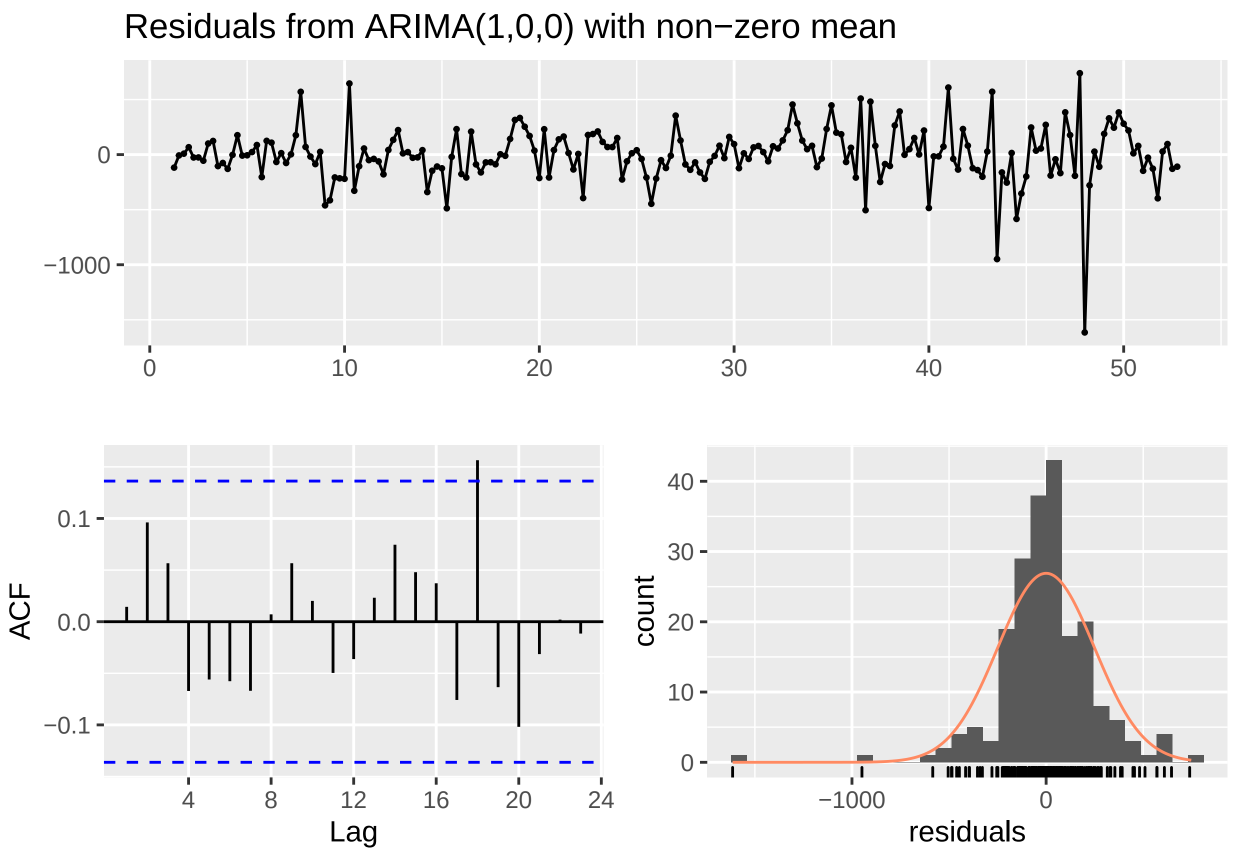
Consequently, I decided to test the MA(2) and the AR(1) for residuals autocorrelation.

```
forecast::checkresiduals(ARMAAIC)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,2) with non-zero mean
## Q* = 3.282, df = 5, p-value = 0.6566
##
## Model df: 3.   Total lags used: 8
```

```
forecast::checkresiduals(ARMABIC)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 6.0047, df = 6, p-value = 0.4227
##
## Model df: 2.   Total lags used: 8
```

In the case of checking the residuals, since we have p-values greater than 0.05, we can say that we fail to reject the null hypothesis that the autocorrelation in the series is zero.

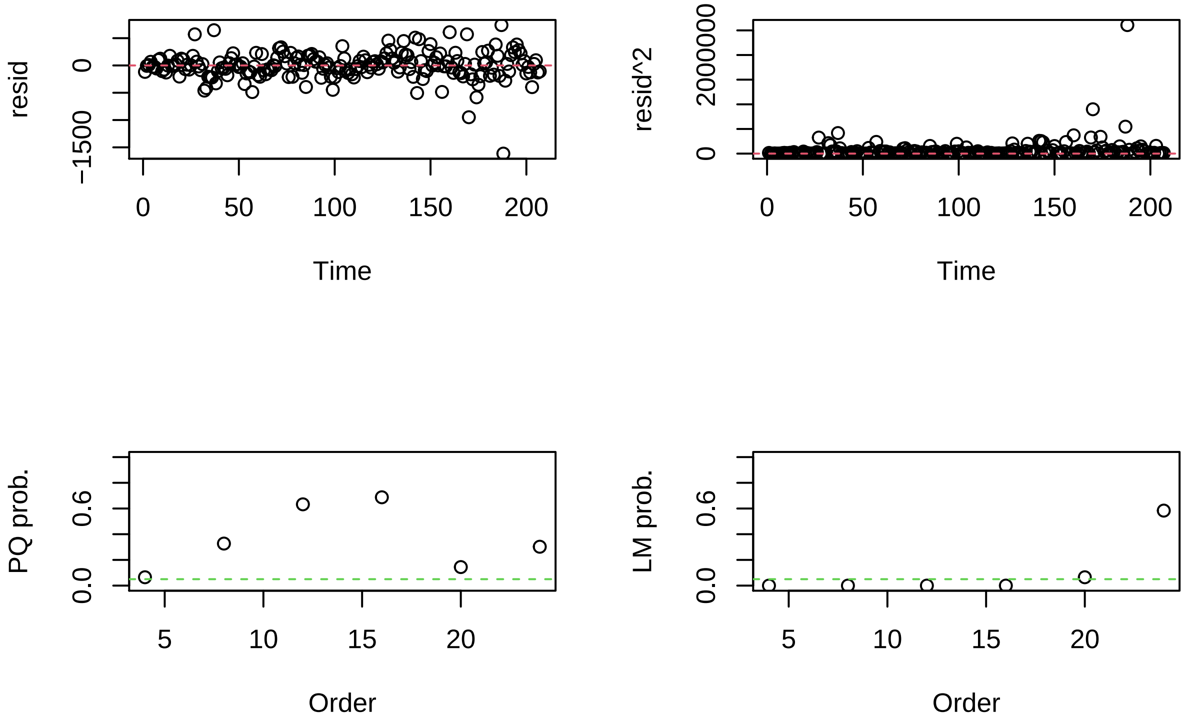
```
ArchTest(diff1dat, lags = 1, demean = FALSE)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  diff1dat
## Chi-squared = 10.802, df = 1, p-value = 0.001014
```

```
arch.test(arima(diff1dat,order=c(1,0,0)),output=TRUE)
```

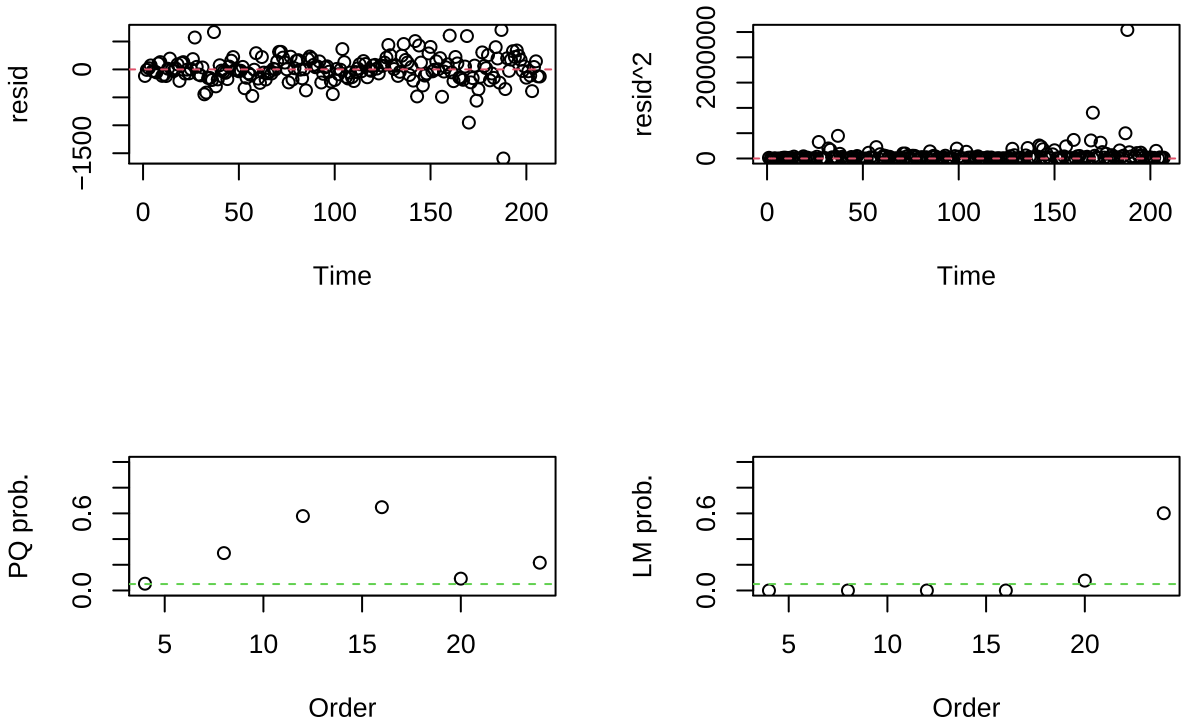
```
## ARCH heteroscedasticity test for residuals
## alternative: heteroscedastic
##
```

```
## Portmanteau-Q test:
##      order    PQ p.value
## [1,]     4  8.85  0.0649
## [2,]     8  9.18  0.3270
## [3,]    12  9.81  0.6325
## [4,]    16 12.80  0.6874
## [5,]    20 26.69  0.1442
## [6,]    24 27.04  0.3024
## Lagrange-Multiplier test:
##      order    LM  p.value
## [1,]     4 312.4 0.00e+00
## [2,]     8 142.7 0.00e+00
## [3,]    12  87.7 4.80e-14
## [4,]    16  56.3 1.09e-06
## [5,]    20  29.1 6.48e-02
## [6,]    24  21.0 5.84e-01
```



```
arch.test(arima(diff1dat,order=c(0,0,2)),output=TRUE)
```

```
## ARCH heteroscedasticity test for residuals
## alternative: heteroscedastic
##
## Portmanteau-Q test:
##      order    PQ p.value
## [1,]     4  9.36  0.0527
## [2,]      8  9.64  0.2909
## [3,]     12 10.42  0.5789
## [4,]     16 13.34  0.6481
## [5,]     20 28.78  0.0921
## [6,]     24 29.11  0.2161
## Lagrange-Multiplier test:
##      order    LM  p.value
## [1,]     4 304.4 0.00e+00
## [2,]     8 138.8 0.00e+00
## [3,]    12  85.0 1.56e-13
## [4,]    16  54.7 2.02e-06
## [5,]    20  28.4 7.68e-02
## [6,]    24  20.7 6.01e-01
```



In the above code, I run two separate ARCH test, one testing on the difference data that we have shown to not contain a unit root, and then running another ARCH test on the 2 models that I have chose in order to reinforce my first finding. From the p-values, we can conclude that we reject the null hypothesis that there is no heteroscedsaticity and that we have an ARCH model of order 1, when looking at the Lagrange-Multiplier test.

Given all this statiscal evidence, my two favorite models that I would choose for the transformed series are the MA(2) and the AR(1) models with the addition of the ARCH(1). As shown multiple times, the MA(2) and the AR(1) models have shown to be the best fit for the data. The economic reason for me choosing the these two models is due to the nature of the our observed data, personal income. The autoregressive models uses observations from the past in order to predict the value in the next period, and since we are dealing with personal income, it is conceivable that the individual will see an increase in his income because of his career choices/promotions or due to shocks that take longer to recover, such as a trade war. On the other hand, the MA model is valid because the individual's income could be dependent on some economic shocks that only last for a relatively "short time" such as a reform of the tax code.

Question 2:

Part a)

Look at Lecture 4

Heteroscedasticity is when the variance of errors are changing over time, usually when they become higher and higher the further we move to the right on the x axis. It might be important for the specific series that we are using because, when the variance of errors is changing, it would be harder to fit a model due to increase variation from the trend relationship. For this, I refer to the hand drawn graph.

Before testing for heteroscedasticity we should first test for serial correlation. After the serial correlation has been corrected for, then we test for heteroscedasticity. We do two tests depending on the type of heteroscedasticity that we suspect. If there is autocorrelation or not. If there is then we add lags until it disappears, if not we continue on to the regular tests.

To test for ARCH/AGRCH we may use the Breusch-Pagan Test for Heteroskedasticity or the Lagrange Multiplier test. Once it has been shown that heteroscedasticity is present then we look that the ACF and the PACF to tell us which Model to use. Once we have determined the model, we can then use the maximum likelihood estimation method to estimate parameters of ARCH or GARCH model. Generally we can assume first that we need to model an ARCH model and then we test ARCH with numerous long lags to see if they are significant. If the long term lags are significant, then we can suspect a GARCH model of variance.

Part b)

Autocorrelation occurs when the errors are serially correlated. The assumtion that is violated when it comes to autocorrelation is the TS 5 assumption of no serial correlation, however for the finite sample, TS.5 is not needed for having unbiased OLS estimator but the OLS estimators will not be BLUE. The "no autocorrelation" assumption is not necessary for Consistency of OLS estimators in general with the exception of when we are dealing with lagged dependent variables in our model(s). The reason is that contemporaneous exogeneity would not hold making the OLS yield inconsistent estimates.

To overcome the issue, we could use the MLE, remove it or do inference with "corrected" tests. The reason is that the OLS standard errors overstate the statistical significance as there is less independent variation. Therefore, we need to correct the standard errors after OLS using the Newey-West correction for autocorrelation.

Part c)

As can be seen by the page attached to his document, if we were to simply have an AR(1) model and we assume that there is a serial correlation in the error term. What ends up happening, through substitution, is the the OLS is giving us coefficients that are consistent with an AR(1) model even though the correct model for y is the AR(2) model. Consequently, this acts like an omission of an explanatory variable thus causing bias and inconsistency in the OLS estimates.

$AR(1) = y_t = \beta_0 + \beta_1 y_{T-1} + u_t \qquad u_t = \rho u_{t-1} + e_t$

so auto correlation

$y_t = \beta_0 + \beta_1 y_{T-1} + \rho(y_{t-1} - \beta_0 - \beta_1 y_{t-2}) \quad u_{t-1} = y_{t-1} - \beta_0 - \beta_1 y_{T-2}$

$= \beta_0(1-\rho) + (\beta_1 + \rho) y_{t-1} + \rho\beta_1 y_{T-2} + e_t$

Have an AR(1) but we really have an AR(2) so OLS is inconsistent because it uses the wrong model.

Question 3 AR(1)

AR(1): $y_t = \beta_0 + \beta_1 y_{t-1} + u_t$

Assume $\beta_0 = 0$ and u_t = white noise

given $I_T = \{y_1, y_2, \dots, y_T; u_1, u_2, \dots, u_T\}$

One Period Ahead Forecast optimal Forecast

$x_{T+1,T} = \beta x_T + u_{T+1}$

$$x_{T+1,T} = E[x_{T+1} | I_T] = E[\beta y_T + u_{T+1} | I_T]$$
$$= E[\beta y_T | I_T] + E[u_{T+1} | I_T]$$
$$= \beta y_T + 0$$
$$= \boxed{\beta y_T}$$

Two Period Ahead Forecast

$$y_{T+2} = \beta y_{T+1} + u_{T+2}$$
$$= \beta(\beta y_T + u_{T+1}) + u_{T+2}$$
$$= \beta^2 y_T + \beta u_{T+1} + u_{T+2}$$

$$y_{T+2,T} = E(x_{T+2} | I_T) = \beta^2 y_T + \beta 0 + 0$$
$$= \boxed{\beta^2 y_T}$$

Three Period Forecasts

$$y_{T+3} = \beta y_{T+2} + u_{T+3}$$
$$= \beta(\beta^2 y_T + \beta u_{T+1} + u_{T+2}) + u_{T+3}$$
$$= \beta^3 y_T + \beta^2 u_{T+1} + \beta u_{T+2} + u_{T+3}$$

$$y_{T+3,T} = \boxed{\beta^3 y_T}$$

K period Forecasts

$$y_{T+K,T} = \boxed{\beta^K y_T}$$

Forecast error Variance

Period(1) = $Var(e_{T+1|T}) = \boxed{\sigma^2}$

Period(2) = $Var(e_{T+2|T}) = \boxed{(1+\beta^2)\sigma^2}$

Period(3) = $Var(e_{T+3|T}) = Var(\beta^2 u_{T+1} + \beta u_{T+2} + u_{T+3})$

$$= \boxed{(\beta^4 + \beta^2 + 1)\sigma^2}$$

Forecast Error (1 period)

$$e_{T+1,T} = x_{T+1} - x_{T+1,T} = \boxed{u_{T+1}}$$

Forecast Error (2 period)

$$e_{T+2,T} = x_{T+2} - x_{T+2,T}$$
$$= \boxed{\beta u_{T+1} + u_{T+2}}$$

Forecast error 3 period

$$e_{T+3,T} = x_{T+3} - x_{T+2,T}$$
$$= \boxed{\beta^2 u_{T+1} + \beta u_{T+2} + u_{T+3}}$$

Forecast error k periods

$$e_{T+K,T} = \boxed{\beta^{K-1} u_{T+1} + \beta^{K-2} u_{T+2} + \dots + u_{T+K}}$$

AR(2)

$AR(2) = y_T \pm \beta_1 y_{T-1} + \beta_2 y_{T-2} + u_t$

One period ahead: Forecast

$y_{T+1} = \beta_1 y_T + \beta_2 y_{T-1} + u_{t+1}$

$$\begin{aligned} y_{T+1,T} &= E[y_{T+1} | I] \\ &= E[\beta_1 y_T] + E[\beta_2 y_{T-1}] + E[u_{t+1}] \\ &= \beta_1 y_T + \beta_2 y_{T-2} + 0 \\ &= \boxed{\beta_1 y_T + \beta_2 y_{T-1}} \end{aligned}$$

2 period ahead Forecast

$y_{T+2} = \beta_1 y_{T+1} + \beta_2 y_T + u_{t+2}$

$$\begin{aligned} y_{T+2,T} &= E[\beta_1 y_{T+1}] + E[\beta_2 y_T] + E[u_{t+2}] \\ &= \beta_1(\beta_1 y_T + \beta_2 y_{T-1}) + \beta_2 y_T + 0 \\ &= \beta_1^2 y_T + \beta_1 \beta y_{T-1} + \beta_2 y_T \\ &= \boxed{y_T(\beta_1^2 + \beta_2) + \beta_1 \beta_2 y_{T-1} +} \end{aligned}$$

3 period ahead: Forecast

$y_{t+3} = \beta_1 y_{T+2} + \beta_2 y_{T+1} + u_{t+3}$

$$\begin{aligned} y_{t+3,T} &= \beta_1 E[y_{T+2}] + \beta_2 E[y_{T+1}] + 0 \\ &= [\beta_1^3 + \beta_1 \beta_2] y_t + \beta_1^2 \beta_2 y_{T-1} + \beta_1 \beta_2 y_T + \beta_2^2 y_{T-1} \\ &= \boxed{[\beta_1^3 + 2\beta_1 \beta_2] y_t + [\beta_1^2 * \beta_2 + \beta_2^2] y_{T-1}} \end{aligned}$$

Forecast error 1 period :

$$\begin{aligned} e_{T+1,T} &= y_{T+1} - y_{T+1,T} \\ &= \boxed{u_{t+1}} \end{aligned}$$

Forecast error 2 period ahead

$$\begin{aligned} e_{T+2,T} &= y_{T+2} - y_{T+2,T} \\ &= \boxed{\beta_1 u_{T+1} + u_{t+2}} \end{aligned}$$

Forecast error 3 period

$$\begin{aligned} e_{T+3,t} &= y_{T+3} - y_{T+3,T} \\ &= \boxed{\beta_1 u_{T+2} + \beta_1 \beta_2 u_{T+1} + u_{t+3}} \end{aligned}$$

Forecast Error Variance:

$$\boxed{\begin{aligned} Var(e_{T+1,T}) &= \sigma^2 \\ Var(e_{T+2,T}) &= (1 + \beta_1^2)\sigma^2 \\ Var(e_{T+3,T}) &= (\beta^4 + \beta^2 + 1)\sigma^2 \end{aligned}}$$