STA3050 Assignment 5

Nzambuli Daniel

2024-07-28

QUESTION 1: Fitting an ARMA Model:

You are a data analyst tasked with modeling a time series using an ARMA model. Your objective is to understand the dynamics of the series and make future forecasts.

Packages: forecast and tseries

1. Simulate a time series dataset of length 500 from an ARMA(2,1) model with AR parameters 0.5 and 0.3, and an MA parameter 0.4. Ensure you set a seed for reproducibility

```
library(stats)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries)
```

ARMA in R

> Using `ARMA_SIM` from stats (r-project_org, 2024)

```
set.seed(2222)

q1_data = arima.sim(n = 500, model = list(ar = c(0.5, 0.3), ma = c(0.4)))
head(q1_data)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1]  0.5906134  0.8281733  0.4849216 -1.0740276 -1.1936571 -2.0064721
```

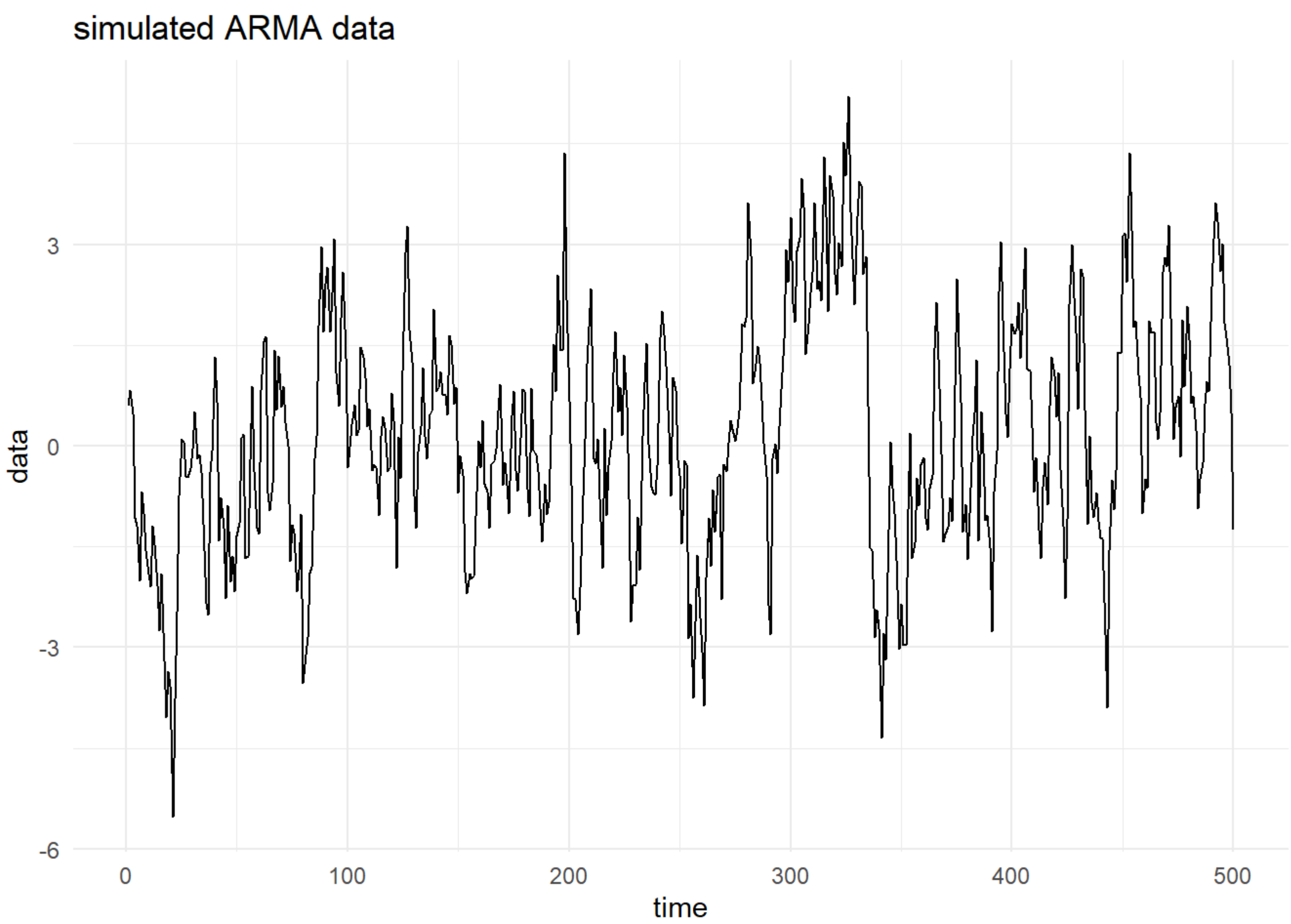
2. Plot the simulated time series data and describe any patterns or characteristics you observe

```
q1_arma_data = data.frame(
  time = seq(1,500),
  data = as.numeric(q1_data)
```

```
)
head(q1_arma_data)
```

```
##   time       data
## 1    1  0.5906134
## 2    2  0.8281733
## 3    3  0.4849216
## 4    4 -1.0740276
## 5    5 -1.1936571
## 6    6 -2.0064721
```

```
library(ggplot2)
ggplot(q1_arma_data, aes(x = time, y = data))+
  geom_line()+
  labs(
    title = "simulated ARMA data",
    xlab = "time",
    ylab = "Simulated"
  )+
  theme_minimal()
```



Observed Patterns and Characteristics

1. **Volatility** – the data exibits high volatility with steep rises and drops across the period of time
2. **Heteroskedasticity** – the amptitudes of the variance is not constant and varies across the time period
3. **Seasonality** – there is no observed begin and end. There is no observed pattern in the data

4. **Extreme values** – there are extreme values at around `-6` and `4.3`
5. **Mean** – the mean hovers about 0
6. **Trend** – there is no observed trend

3. Plot the ACF and PACF of the simulated ARMA data. Interpret the plots

Check stationarity

```
library(tseries)
adf.test(q1_arma_data$data, alternative = "stationary")
```

```
## Warning in adf.test(q1_arma_data$data, alternative = "stationary"): p-value
## smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  q1_arma_data$data
## Dickey-Fuller = -5.4436, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

- **Null Hypothesis** H_0: The data is non-stationary. This implies that the statistical properties of the series, such as the mean and variance, are dependent on time.
- **Alternative Hypothesis** H_1: The data is stationary. This means the statistical properties of the series, such as the mean and variance, are constant over time and do not depend on when the observations were taken.

> Because the p-value is < 0.05 we reject the null hypothesis and conclude that the data is stationary.
>
> We can run the ACF test without needing to *difference* the data to make it stationary

ACF plot

```
auto_corr_func = function(data, k){
  n = length(data)
  mu = mean(data)
  if(k == 0){
    return(1)
  }else{
  num = sum(((data[(k+1): n]) - mu)  *((data[1: (n-k)]) - mu))
  denom = sum((data -mu)^2)

  autocorr = num/denom

  return(autocorr)
  }
}
```

```
plot_data_acfs = data.frame(
  lag = seq(0,25)
```

```
)

data_acf = q1_arma_data$data
for(i in seq(0,25)){
  col_name = paste("ACF_k_", i)
  plot_data_acfs[[col_name]] = auto_corr_func(data_acf, i)
}


print(paste("There are:", ncol(plot_data_acfs)-1,"\nThe first 6 are:"))
```

```
## [1] "There are: 26 \nThe first 6 are:"
```

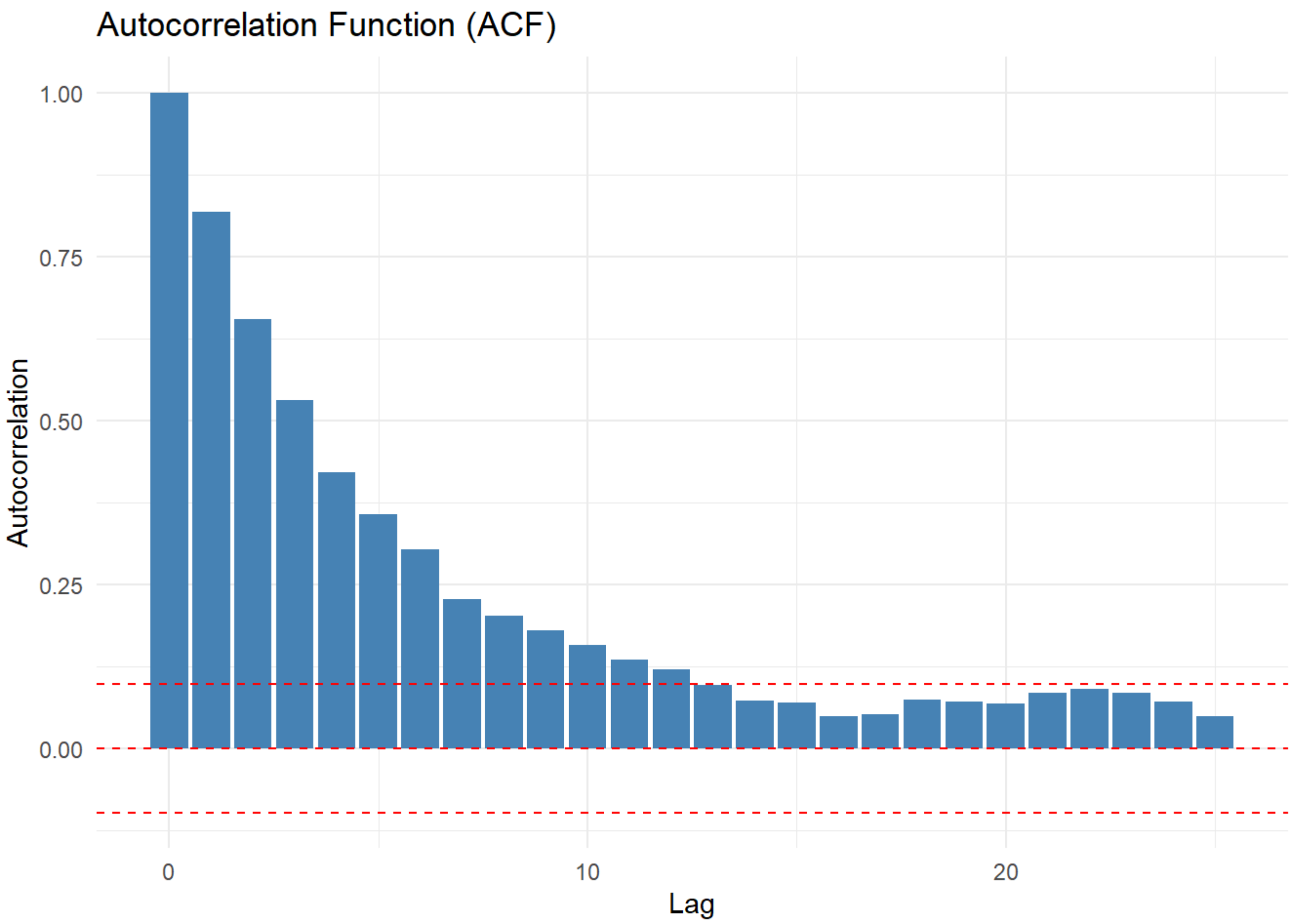
```
head(plot_data_acfs)
```

```
##   lag ACF_k_ 0  ACF_k_ 1  ACF_k_ 2  ACF_k_ 3  ACF_k_ 4 ACF_k_ 5  ACF_k_ 6
## 1   0        1 0.8181138 0.6548167 0.5302234 0.4200809    0.357 0.3025177
## 2   1        1 0.8181138 0.6548167 0.5302234 0.4200809    0.357 0.3025177
## 3   2        1 0.8181138 0.6548167 0.5302234 0.4200809    0.357 0.3025177
## 4   3        1 0.8181138 0.6548167 0.5302234 0.4200809    0.357 0.3025177
## 5   4        1 0.8181138 0.6548167 0.5302234 0.4200809    0.357 0.3025177
## 6   5        1 0.8181138 0.6548167 0.5302234 0.4200809    0.357 0.3025177
##    ACF_k_ 7 ACF_k_ 8  ACF_k_ 9 ACF_k_ 10 ACF_k_ 11 ACF_k_ 12 ACF_k_ 13
## 1 0.2278469  0.20136 0.1792468 0.1571164 0.1341661 0.1199883 0.0966758
## 2 0.2278469  0.20136 0.1792468 0.1571164 0.1341661 0.1199883 0.0966758
## 3 0.2278469  0.20136 0.1792468 0.1571164 0.1341661 0.1199883 0.0966758
## 4 0.2278469  0.20136 0.1792468 0.1571164 0.1341661 0.1199883 0.0966758
## 5 0.2278469  0.20136 0.1792468 0.1571164 0.1341661 0.1199883 0.0966758
## 6 0.2278469  0.20136 0.1792468 0.1571164 0.1341661 0.1199883 0.0966758
##    ACF_k_ 14  ACF_k_ 15  ACF_k_ 16  ACF_k_ 17  ACF_k_ 18  ACF_k_ 19  ACF_k_ 20
## 1 0.07232006 0.06929973 0.04905963 0.05183113 0.07451449 0.07087649 0.06838806
## 2 0.07232006 0.06929973 0.04905963 0.05183113 0.07451449 0.07087649 0.06838806
## 3 0.07232006 0.06929973 0.04905963 0.05183113 0.07451449 0.07087649 0.06838806
## 4 0.07232006 0.06929973 0.04905963 0.05183113 0.07451449 0.07087649 0.06838806
## 5 0.07232006 0.06929973 0.04905963 0.05183113 0.07451449 0.07087649 0.06838806
## 6 0.07232006 0.06929973 0.04905963 0.05183113 0.07451449 0.07087649 0.06838806
##    ACF_k_ 21  ACF_k_ 22  ACF_k_ 23  ACF_k_ 24  ACF_k_ 25
## 1 0.08475171 0.09064045 0.08467819 0.07113898 0.04822206
## 2 0.08475171 0.09064045 0.08467819 0.07113898 0.04822206
## 3 0.08475171 0.09064045 0.08467819 0.07113898 0.04822206
## 4 0.08475171 0.09064045 0.08467819 0.07113898 0.04822206
## 5 0.08475171 0.09064045 0.08467819 0.07113898 0.04822206
## 6 0.08475171 0.09064045 0.08467819 0.07113898 0.04822206
```

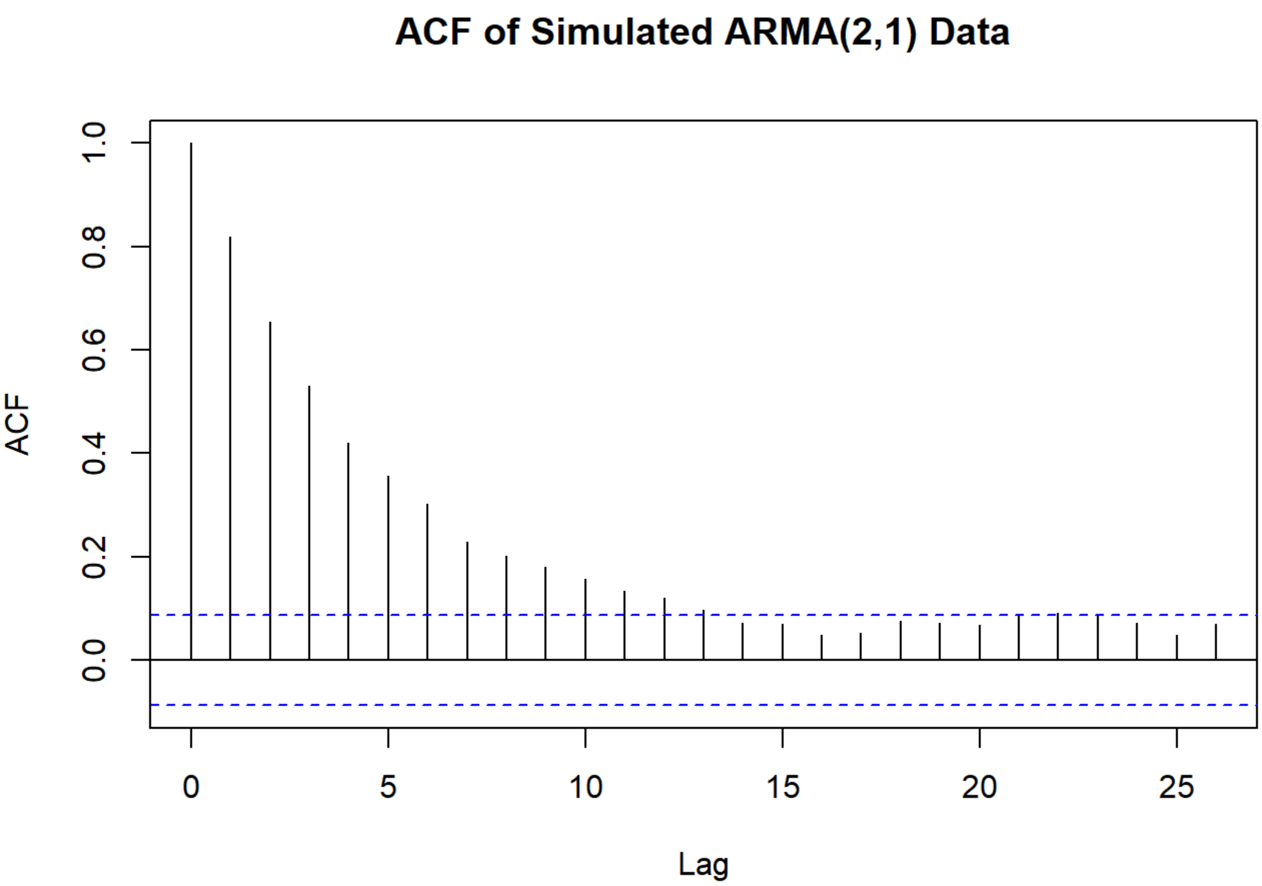
Comparing the calculated plot and the built-in plot

```
plot_data_acfs$acf = c(unlist(unname(plot_data_acfs[1, 2:27])))

N = length(plot_data_acfs$acf)
std_error = 0.5/sqrt(N)



ggplot(plot_data_acfs, aes(x = lag, y = acf)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_hline(yintercept = c(-std_error, std_error), linetype = "dashed", color = "red") +
  labs(
    title = "Autocorrelation Function (ACF)",
```

```
    x = "Lag",
    y = "Autocorrelation") +
  theme_minimal()
```




```
acf(q1_data, main = "ACF of Simulated ARMA(2,1) Data")
```



Observation

1. lag `0 - 12` are above the dashed line
2. There auto-correlations drop from 1 and keep dropping to close to 0

Interpretation

The near-zero auto-correlations after the initial drop indicate limited long-term predictable patterns.

The influence of the AR terms is strong initially and diminishes quickly

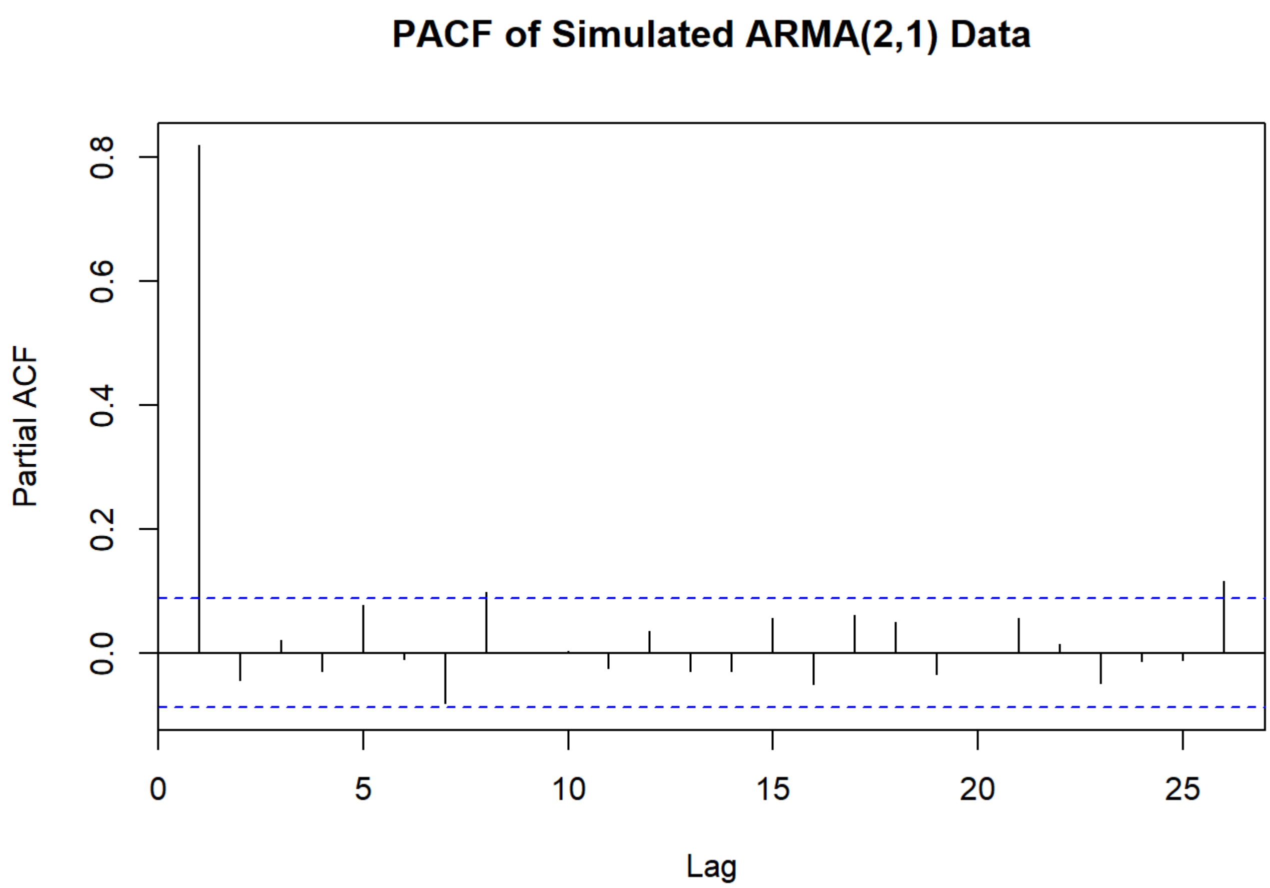
This follows where an ARMA(2,1) model moving averages impact the immediate lag and the auto-regressive parameters impact the first two lags.

When the lags move close to 0 it indicates that there is **little noise** with limited long-term predictable structure beyond what the model's parameters can explain

PACF plot

Based on r documentation `pacf` can be used to plot the partial autocorrelations.

```
pacf(q1_data, main = "PACF of Simulated ARMA(2,1) Data")
```



Observations

1. The **first lag(0)** and **second lag(1)** are the longest
2. After **lag 2** the values drop to `near 0` below the dashed line.

Interpretations

The dashed line is used to represents the a standard error of the data set at $\frac{\pm 2}{T}$ where T is the length of the time series data.

Because of this, the lags that fall below the dashed lines have no significant relationships with the past values.

The first two lags explain the unique information that is not explained by the other lag values.

Meaning: the `first two past values` have a significant linear relationship with the current value after accounting for all other lags.

Model Fit: The model fits for the an auto regressive(2). This is because the PACF **cuts off at the second lag**. An AR(2) is therefore the most appropriate for this auto-regressive moving average set at (2 AR parameters, 1 MA parameter).

4. Fit an ARMA(2,1) model to the simulated data. Summarize the model and interpret the key output components, including parameter estimates and their significance, standard error, and model fit statistics

```
arma_model_q1 = arima(q1_data, order= c(2, 0, 1))

summary(arma_model_q1)
```

```
##
## Call:
## arima(x = q1_data, order = c(2, 0, 1))
##
## Coefficients:
##          ar1     ar2     ma1  intercept
##       0.2997  0.4065  0.5626     0.1535
## s.e.  0.2869  0.2396  0.2757     0.2354
##
## sigma^2 estimated as 0.9957:  log likelihood = -708.95,  aic = 1427.89
##
## Training set error measures:
##                        ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.001029104 0.9978426 0.7937946 49.55445 156.8433 0.9480431
##                      ACF1
## Training set -0.001520979
```

Output	Type	Meaning
0.2997, 0.4065	auto-regressive parameters	• For ar1 the se is close to the ar1 parameter which indicates a challenge in the reliability of the estimate • ar2 is a more reliable estimate of the as the error has a lower magnitude than the estimate
0.5625	moving average	• MA adds to the prediction based on the error term from the previous time step.

0.1535 intercept

- the higher error than the estimate indicates slight uncertainty in the estimate

the **Variance** is close to `1` indicating that the model leaves alot of uncertainity unexplained

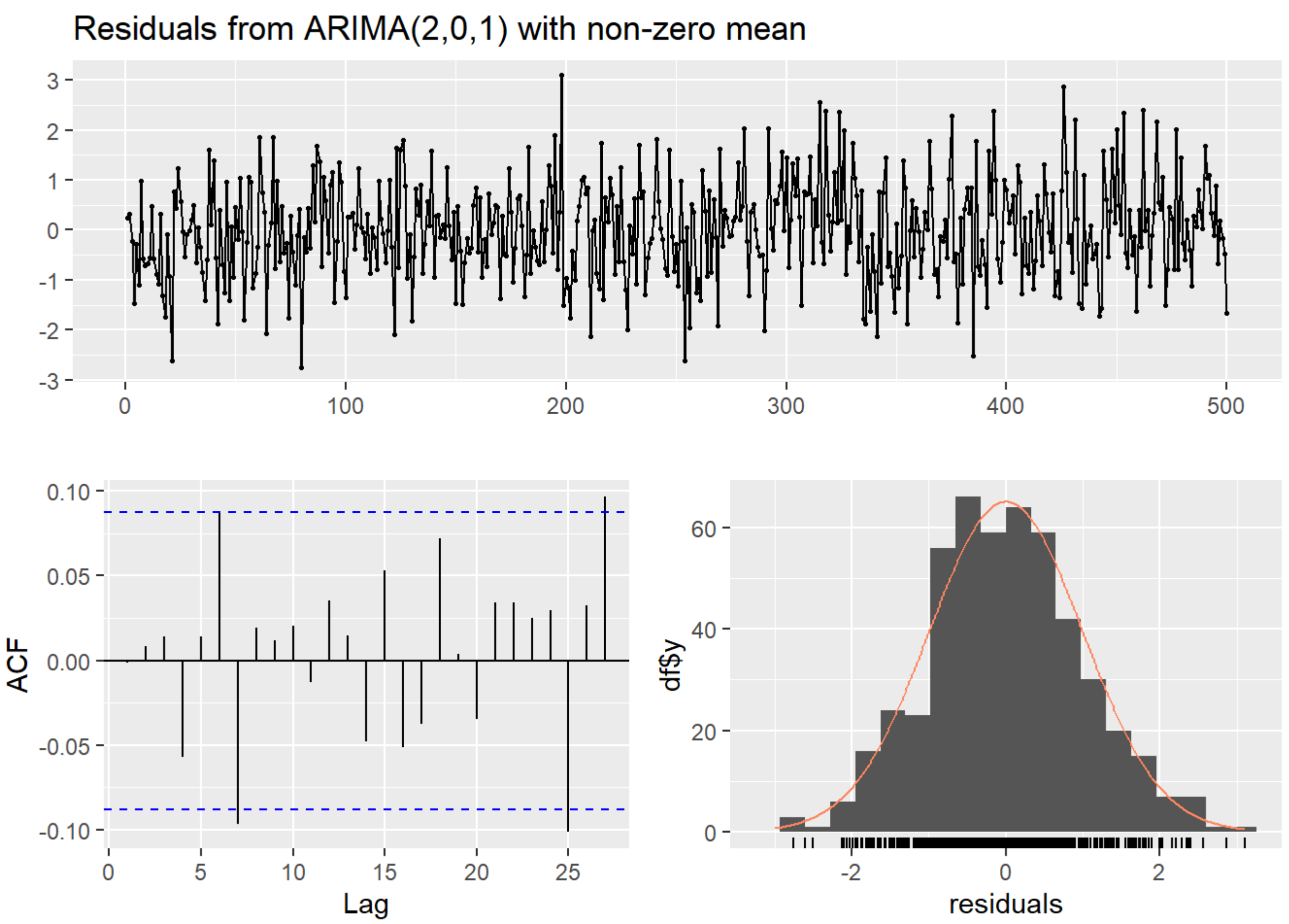
the **log likelihood** is a measure of how likely the model is to generate the provided data. so the target is a more positive number

Lower AIC are preferred.

5. Perform the diagnostic checks on the fitted ARMA model, including residual analysis and autocorrelation checks

residual analysis

```
checkresiduals(arma_model_q1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1) with non-zero mean
## Q* = 10.899, df = 7, p-value = 0.1431
##
## Model df: 3.   Total lags used: 10
```

Ljung-box Output

observation

- the data has an intercept, has a $\mu \neq 0$
- the sum of squared auto correlations is $Q = 10.899$
- the degrees of freedom are 7

$$\begin{aligned} Df &= Total\ lags\ used - model\ df \\ &= 10 - 3 \\ &= 7 \end{aligned}$$

- p-value 0.1431 which is greater than a significance value of 0.05
- there are 3 parameters used $model\ df$

interpretation

H_0 There is no significant evidence of autocorrelation in the residuals of ARMA

H_1 There is a statistically significant autocorrelation in the residuals of the ARMA

1. There is no significant evidence of autocorrelation in the residuals of your ARIMA(2,0,1) model at the lags tested up to lag 10
2. The model has adequately captured the auto-correlations in the data
3. The model however leaves some patterns unaccounted for

the plots

1. The residuals from the model were found to be normally distributed and did not show significant autocorrelation. This means that **Residuals are noise**
2. The model will be accurate as the residuals follow a normal distribution.
3. residuals appear as noice based on the line graph with volatile peaks and troughs that are angled sharply

Auto-correlation checks

```
Box.test(arma_model_q1$residuals, lag = 25, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  arma_model_q1$residuals
## X-squared = 27.189, df = 25, p-value = 0.3465
```

From the **P-value** of $0.3465 \geq 0.05$ the data residuals have no significant evidence of autocorrelation

Conclusion the residuals act as noise. The model captures the time based structure of the data.

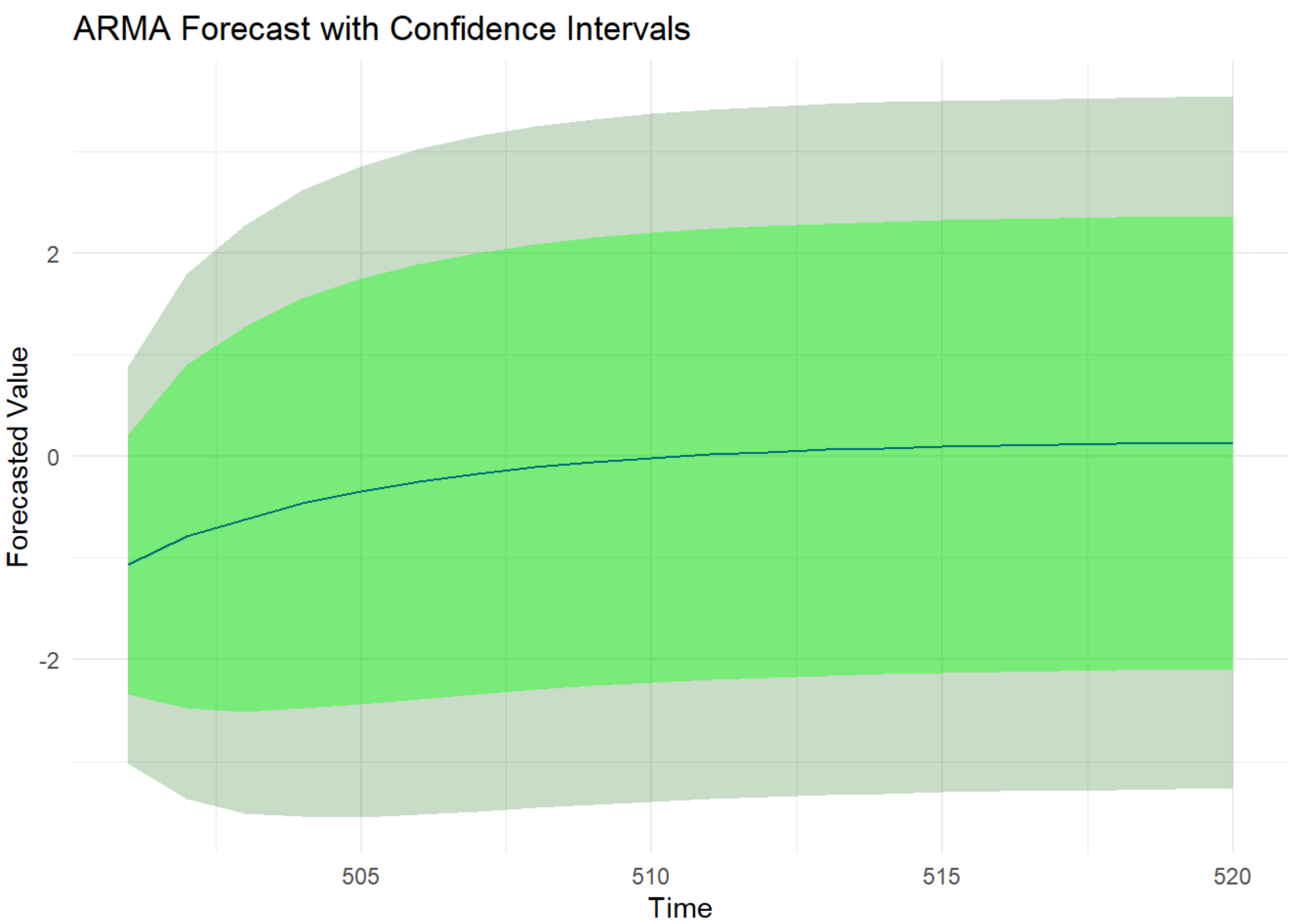
6. Using the fitted ARMA model, forecast the next 20 data points. Plot the forecasted values along with their

confidence intervals.

```
forcst = forecast(arma_model_q1, h = 20)
forcst
```

```
##     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 501    -1.06650228 -2.345289 0.2122844 -3.022238 0.8892332
## 502    -0.78300988 -2.471569 0.9055492 -3.365438 1.7994184
## 503    -0.62316229 -2.513755 1.2674300 -3.514574 2.2682491
## 504    -0.46000847 -2.477121 1.5571041 -3.544916 2.6248989
## 505    -0.34612548 -2.438579 1.7463282 -3.546257 2.8540062
## 506    -0.24566721 -2.386514 1.8951792 -3.519809 3.0284749
## 507    -0.16926164 -2.340475 2.0019516 -3.489846 3.1513225
## 508    -0.10552292 -2.296344 2.0852982 -3.456095 3.2450489
## 509    -0.05535851 -2.258722 2.1480047 -3.425112 3.3143947
## 510    -0.01441204 -2.225888 2.1970639 -3.396573 3.3677485
## 511     0.01825341 -2.198451 2.2349576 -3.371903 3.4084100
## 512     0.04468956 -2.175399 2.2647782 -3.350643 3.4400222
## 513     0.06589224 -2.156384 2.2881686 -3.332786 3.4645707
## 514     0.08299400 -2.140699 2.3066870 -3.317851 3.4838390
## 515     0.09673911 -2.127871 2.3213490 -3.305508 3.4989863
## 516     0.10781104 -2.117393 2.3330147 -3.295344 3.5109664
## 517     0.11671722 -2.108871 2.3423054 -3.287026 3.5204607
## 518     0.12388756 -2.101950 2.3497248 -3.280237 3.5280119
## 519     0.12965722 -2.096341 2.3556558 -3.274714 3.5340282
## 520     0.13430141 -2.091802 2.3604044 -3.270229 3.5388322
```

```
forecst_q1 = data.frame(
  time = seq(501, 520),
  PointForecast = as.numeric(forcst$mean),
  Lo80 = as.numeric(forcst$lower[,1]),
  Hi80 = as.numeric(forcst$upper[,1]),
  Lo95 = as.numeric(forcst$lower[,2]),
  Hi95 = as.numeric(forcst$upper[,2])
)

ggplot(forecst_q1, aes(x = time))+
  geom_line(aes(y = PointForecast), color = "blue") +
  geom_ribbon(aes(ymin = Lo95, ymax = Hi95), fill = "darkgreen", alpha = 0.2) +
  geom_ribbon(aes(ymin = Lo80, ymax = Hi80), fill = "green", alpha = 0.4) +
  labs(title = "ARMA Forecast with Confidence Intervals",
       x = "Time",
       y = "Forecasted Value") +
  theme_minimal()
```



7. Discuss the reliability of these forecasts based on the model diagnostics.

Observation

- narrow confidence interval at the beginning
- wider confidence interval at the end
- predicted values stabilize after time = 515

Interpretation

- There is decreasing accuracy in the forecast data as time increases. There is reduced reliability in long-term forecasts
- This model has values that fall within the threshold set at 95 % CL and even at 80% CL the predicted values still fall within the bounds indicating forecast values are reasonably reliable

Conclusion

The residuals from the model were found to be approximately normally distributed and did not show significant autocorrelation, as evidenced by ACF plots and Ljung-Box test results. **Residuals are noise**

The model is accurate as the residuals follow a normal distribution.

This is a reliable forecast based on a model that has effectively utilized available information in the historical data.

> The model is well fitted because of the AIC and BIC values provided earlier being relatively low, suggesting a good fit of the model to the data

QUESTION 2: Fitting an ARIMA Model:

You have another time series that appears to be non-stationary. Your task is to model this series using an ARIMA model to account for its integrated nature.

Packages: forecast and tseries

1. Simulate a time series dataset of length 500 from an ARIMA(1,1,1) model with AR parameters 0.65, and an MA parameter 0.4. Ensure you set a seed for reproducibility

```
library(forecast)
```

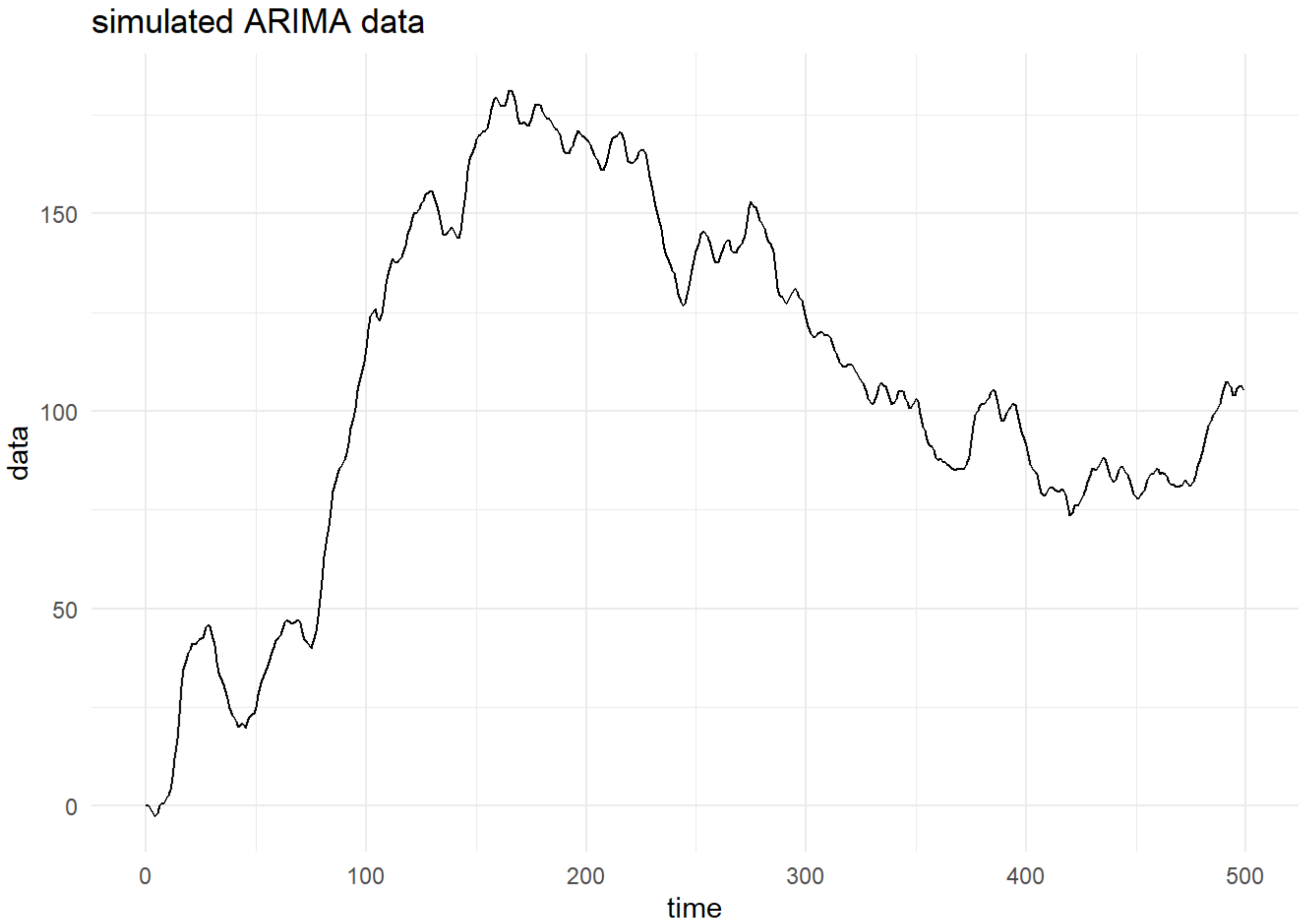
```
set.seed(1111)
sim_arima = arima.sim(n = 499, list(order = c(1, 1, 1), ar = (0.65), ma = c(0.4)))
head(sim_arima)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1]  0.0000000  0.0850389 -0.8543712 -1.7376689 -2.5245399 -1.6556405
```

```
data_q2 = as.numeric(sim_arima)
```

2. Plot the simulated time series data and describe any patterns or characteristics you observe

```
plot_data_q2 = data.frame(
  time = seq(0, 499),
  data = data_q2
)

ggplot(plot_data_q2, aes(x = time, y = data))+
  geom_line()+
  labs(
    title = "simulated ARIMA data",
    xlab = "time",
    ylab = "Simulated"
  )+
  theme_minimal()
```



3. Plot the ACF and PACF of the differenced simulated ARIMA data. Interpret the plots

before plotting the generated data must be made stationary

> ARIMA data is stationary after the first differencing

```
adf.test(sim_arima, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sim_arima
## Dickey-Fuller = -2.3838, Lag order = 7, p-value = 0.4158
## alternative hypothesis: stationary
```

> the p-value $0.4158 \geq 0.05$ we fail to reject H_0 for the Augmented Dickey-Fuller Test
>
> - **conclude** the data needs to be made stationary

```
stationary_arima = diff(sim_arima, differences =  1)
```

check the stationarity of the data

```
adf.test(stationary_arima, alternative = "stationary")
```

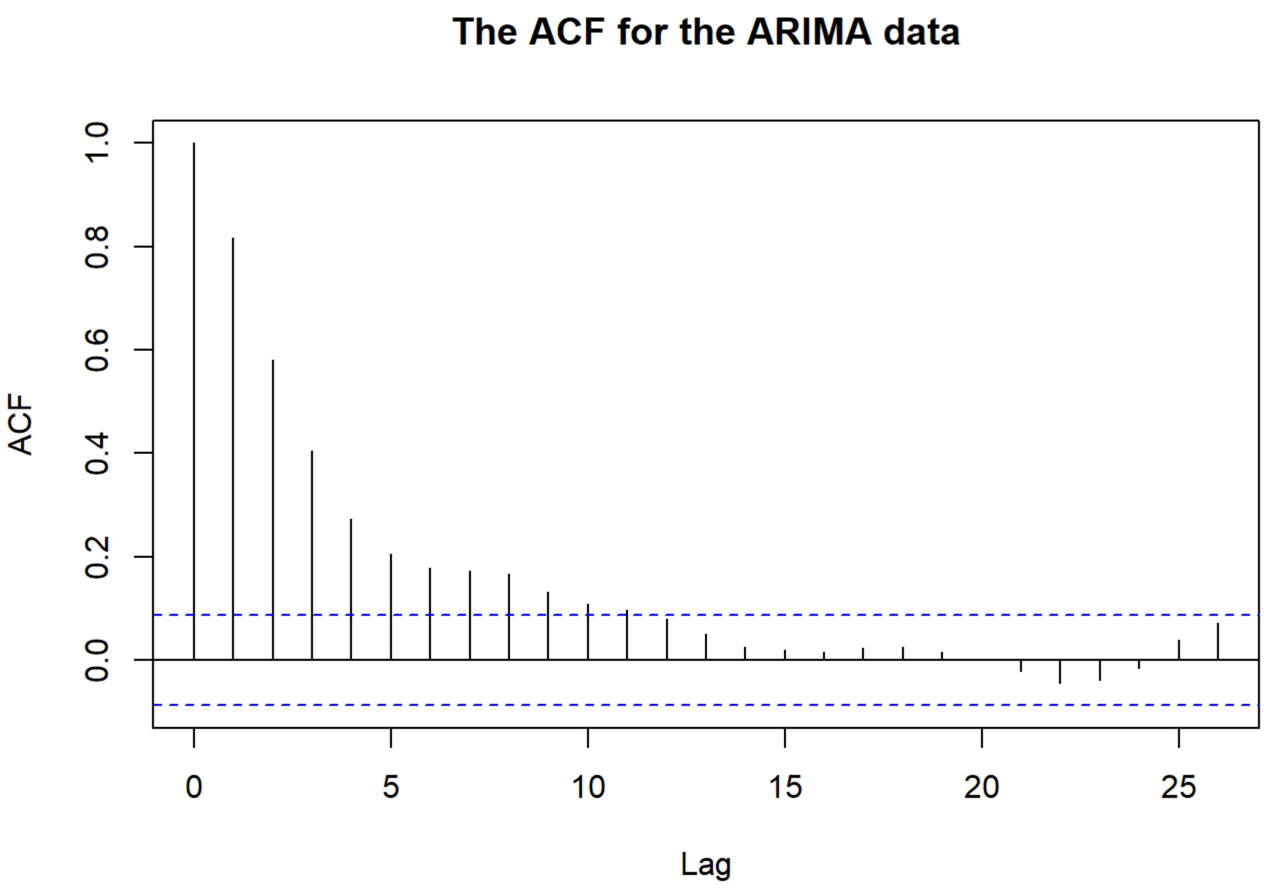
```
## Warning in adf.test(stationary_arima, alternative = "stationary"): p-value
## smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  stationary_arima
## Dickey-Fuller = -5.7167, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

The `Augmented Dickey-Fuller Test` has:

- the experiment p-value is now 0.01 and $0.01 < 0.05$ meaning that we can now reject the null hypothesis
- **conclude** the data is stationary; we can plot the ACF

```
acf(stationary_arima, main = "The ACF for the ARIMA data")
```



Observation

- `the first 10 lags` fall outside the 95% confidence level.
- There is a decrease in autocorrelation from the initial lag at lag 0 of 1
- There are no spikes after the initial 10 lags

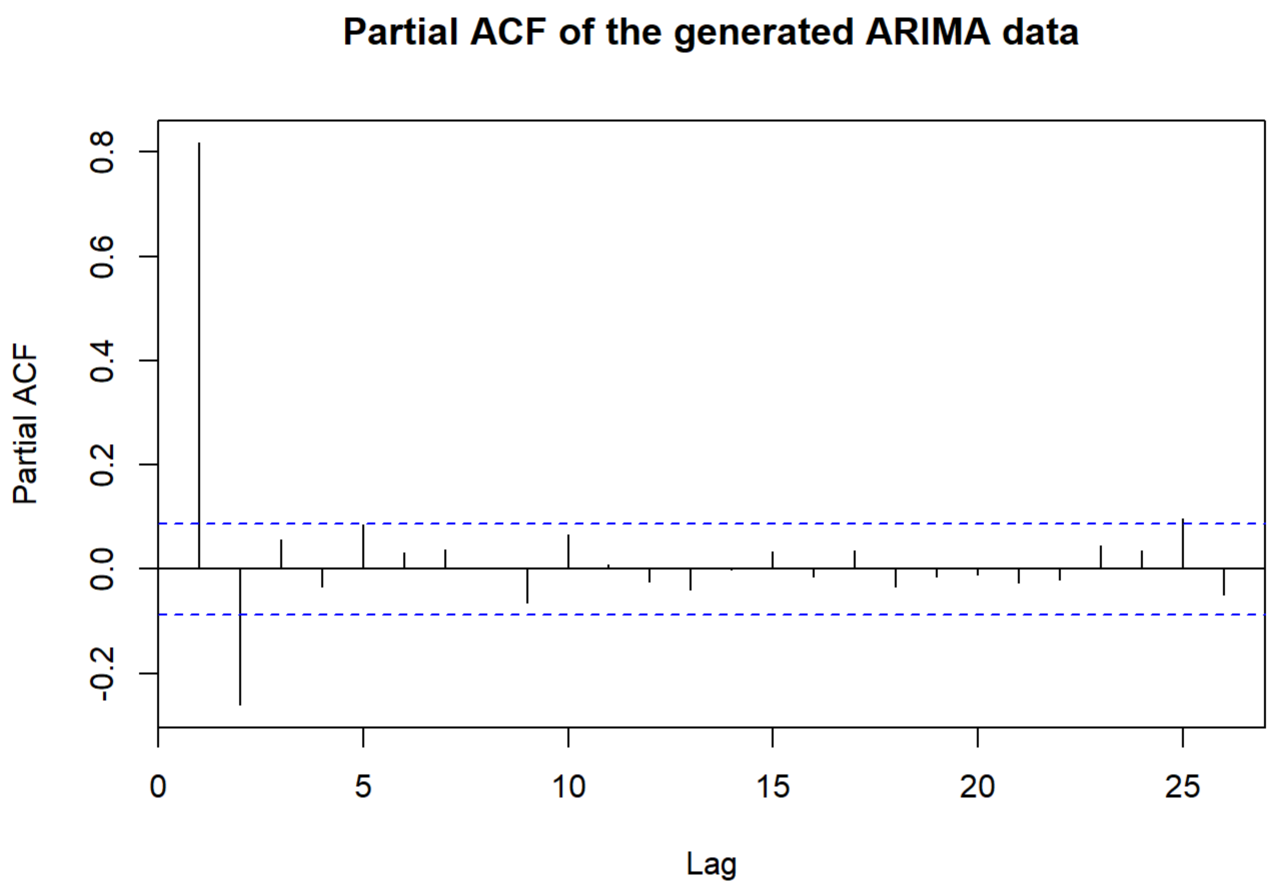
Interpretation

1. The influence of a given observation significantly diminishes as the time lag increases.
2. Past values have little to no influence on future values beyond the first 10 lags

3. The absence of spikes after lag 10 indicate that there are no additional seasonal patterns or longer-term autoregressive behaviors that are not already accounted for by the model.
4. The model accurately captures the auto-regression in the data

PACF

```
pacf(stationary_arima, main = "Partial ACF of the generated ARIMA data")
```



Observation

- The first lag and second lag have a partial auto correlation outside the bounds of the 95% confidence level blue horizontal lines
- After this the data drops and the auto correlation drops to below the confidence interval bounds $\frac{\pm 2}{T}$

Interpretation

1. The data at time t has a notable direct relationship with the data at time $t-1$ which drops significantly at $t-2$ and the relationship is no longer significant after
2. The model fits the AR(1) because of the lack of spikes after the lag 1
3.

4. Fit an ARMA(1,1,1) model to the simulated data. Summarize the model and interpret the key output

components, including parameter estimates and their significance, standard error, and model fit statistics

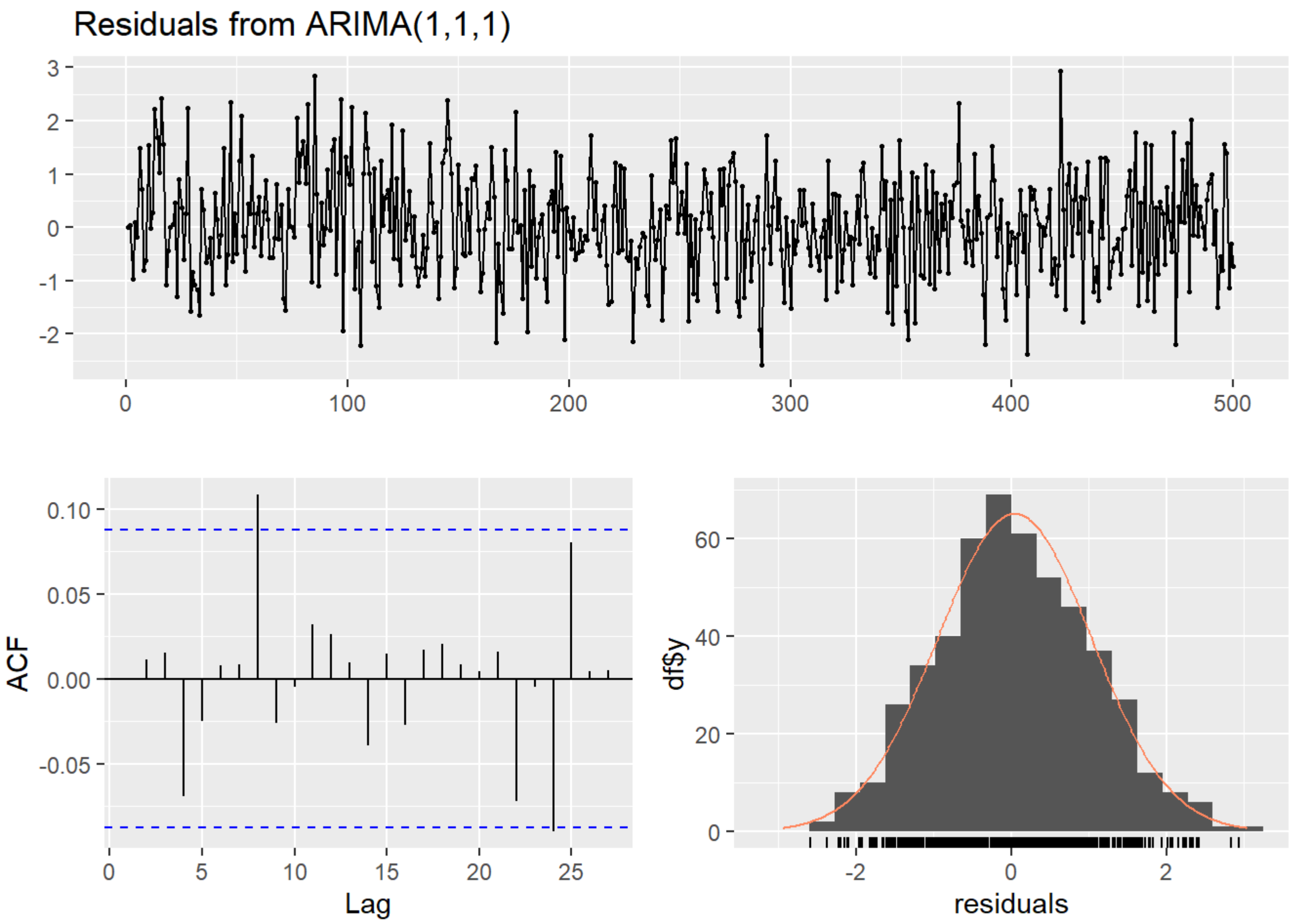
```
q2_model = Arima(sim_arima, order = c(1, 1, 1))
summary(q2_model)
```

```
## Series: sim_arima
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1     ma1
##       0.7171  0.3350
## s.e.  0.0368  0.0508
##
## sigma^2 = 0.9969:  log likelihood = -706.91
## AIC=1419.82   AICc=1419.86   BIC=1432.45
##
## Training set error measures:
##                      ME      RMSE       MAE       MPE     MAPE     MASE
## Training set 0.04315907 0.9954452 0.7950735 0.9438262 2.941211 0.560043
##                        ACF1
## Training set -0.0003234098
```

> **Observation**
>
> - The ar1 has s.e that is significantly lower than the actual estimate
> - the ma1 has a similarly lower s.e than its estimate
> - **Variance** the variance ≈ 1
>
> **Interpretation**
>
> the model leaves a lot of variance unexplained
>
> the model suggests a significant positive relationship between each value and the next in the series, meaning each value is strongly influenced by its immediate predecessor
>
> there is a moderate influence of the previous error term on the current prediction because of the high reliability of the MA
>
> the small se values indicate the model is reliable
>
> The model is a good fit for this particular data set.
>
> It is capable of capturing the main patterns and providing reliable forecasts

5. Perform the diagnostic checks on the fitted ARIMA model, including residual analysis and autocorrelation checks

```
checkresiduals(q2_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 9.3191, df = 8, p-value = 0.3161
##
## Model df: 2.   Total lags used: 10
```

> H_0 There is no significant evidence of autocorrelation in the residuals of ARIMA
>
> H_1 There is a statistically significant autocorrelation in the residuals of the ARIMA

because the pvalue is $0.3161 \geq 0.05$ we fail to reject H_0 and conclude that there is no significant evidence of autocorrelation in the residuals of the ARIMA model.

The plots

- The residuals appear to be noise indicating that the ARIMA(1,1,1) model has effectively extracted underlying patterns from the data leaving behind random noise which does not contain further information
- There is no evident autocorrelation or non-random pattern left in the residuals that could have been otherwise captured by the model.
- The forecasts of the model will therefore be reliable
- The residuals from the model were found to be normally distributed and did not show significant autocorrelation. This means that **Residuals are noise**
- The model will be accurate as the residuals follow a normal distribution.

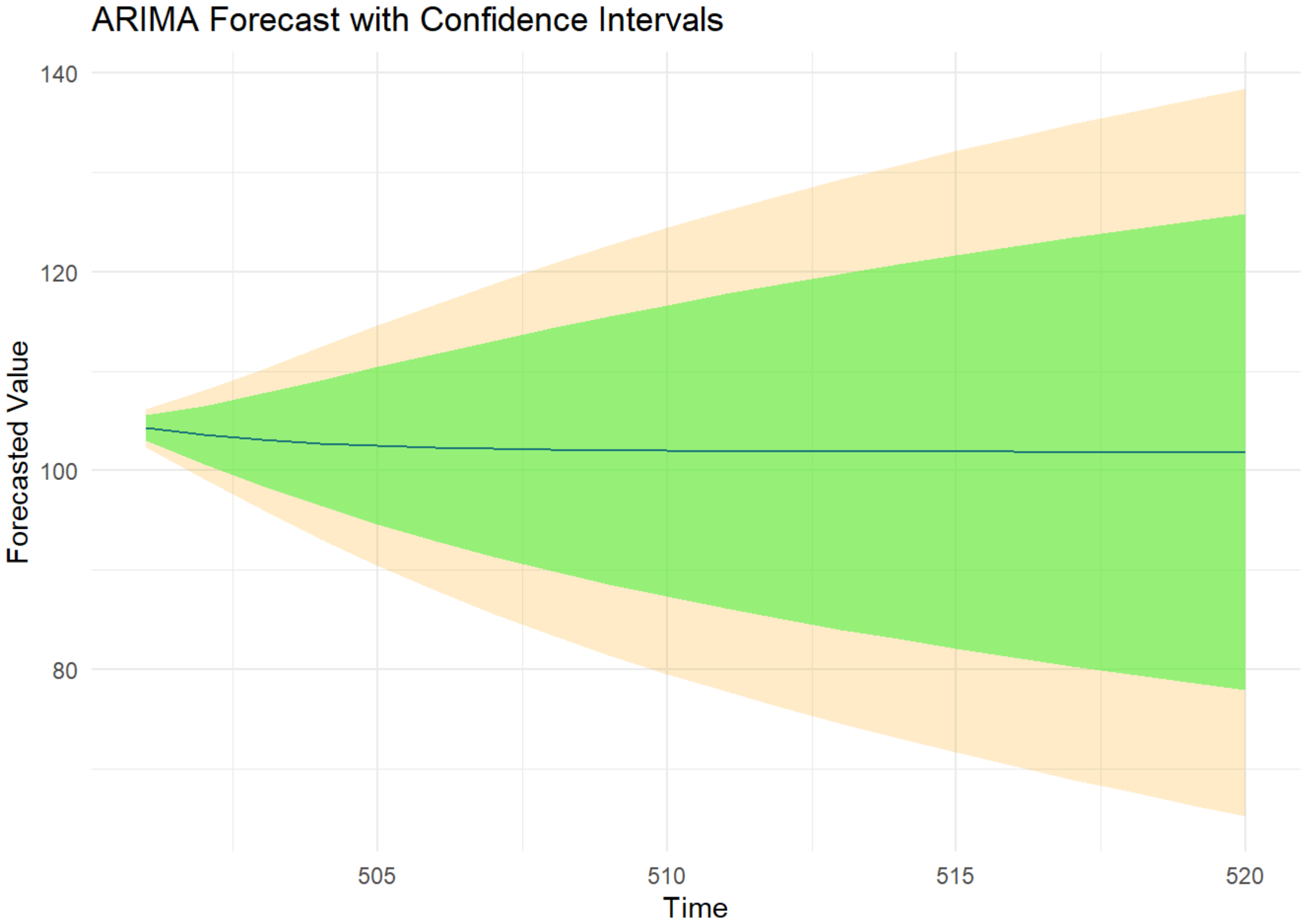
- residuals appear as noice based on the line graph with volatile sharp peaks and troughs

6. Using the fitted ARMA model, forecast the next 20 data points. Plot the forecasted values along with their confidence intervals.

```
forecst_data = forecast(q2_model, h = 20)
forecst_data
```

```
##     Point Forecast      Lo 80    Hi 80     Lo 95    Hi 95
## 501       104.2926 103.01300 105.5721 102.33565 106.2495
## 502       103.6078 100.68675 106.5288  99.14044 108.0751
## 503       103.1167  98.48741 107.7460  96.03681 110.1966
## 504       102.7645  96.45732 109.0718  93.11848 112.4106
## 505       102.5120  94.59792 110.4261  90.40847 114.6155
## 506       102.3309  92.89609 111.7657  87.90162 116.7601
## 507       102.2010  91.33424 113.0677  85.58173 118.8202
## 508       102.1078  89.89450 114.3212  83.42914 120.7865
## 509       102.0410  88.56043 115.5217  81.42423 122.6579
## 510       101.9931  87.31759 116.6687  79.54882 124.4375
## 511       101.9588  86.15357 117.7640  77.78680 126.1308
## 512       101.9341  85.05788 118.8104  76.12412 127.7442
## 513       101.9165  84.02166 119.8113  74.54872 129.2842
## 514       101.9038  83.03750 120.7701  73.05028 130.7573
## 515       101.8947  82.09916 121.6903  71.62002 132.1694
## 516       101.8882  81.20137 122.5750  70.25042 133.5260
## 517       101.8835  80.33971 123.4274  68.93510 134.8320
## 518       101.8802  79.51041 124.2500  67.66857 136.0918
## 519       101.8778  78.71026 125.0453  66.44611 137.3095
## 520       101.8761  77.93650 125.8156  65.26366 138.4885
```

```
forecst_q2 = data.frame(
  time = seq(501, 520),
  PointForecast = as.numeric(forecst_data$mean),
  Lo80 = as.numeric(forecst_data$lower[,1]),
  Hi80 = as.numeric(forecst_data$upper[,1]),
  Lo95 = as.numeric(forecst_data$lower[,2]),
  Hi95 = as.numeric(forecst_data$upper[,2])
)

ggplot(forecst_q2, aes(x = time))+
  geom_line(aes(y = PointForecast), color = "blue") +
  geom_ribbon(aes(ymin = Lo95, ymax = Hi95), fill = "orange", alpha = 0.2) +
  geom_ribbon(aes(ymin = Lo80, ymax = Hi80), fill = "green", alpha = 0.4) +
  labs(title = "ARIMA Forecast with Confidence Intervals",
       x = "Time",
       y = "Forecasted Value") +
  theme_minimal()
```



7. Discuss the reliability of these forecasts based on the model diagnostics

> The forecasts seems to show a generally stable forecast, with a slight downward trend as time progresses
>
> The precision decreased over time as the area under the 80% cf `green` widens. similar to the area under `95%` orange
>
> This means that there is increased uncertainity in the predictions over time
>
> The residuals from the model were found to be approximately normally distributed and did not show significant autocorrelation, as evidenced by ACF plots and Ljung-Box test results. **Residuals are noise**
>
> The residuals being normally distributed support the accuracy of the model. This follows the claim by Hyndman and Athanasopoulos (2018) that residuals for a good forecasting model should be normally distributed.
>
> This is a reliable forecast based on a model that has effectively utilized available information in the historical data.
>
> The model is well fitted because of the AIC and BIC values provided earlier being relatively low, suggesting a good fit of the model to the data