# End Trimester

Nzambuli Daniel

2024-08-08

# Question

**PACKAGES:** forecast and tseries

## 1. Load the Air Passengers data set in R. (2 marks)

```
library(forecast)
```

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

```
library(tseries)
```

```
data("AirPassengers")
AirPassengers
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

```
data = data.frame(
  Year = as.character(rep(seq(1949, 1960, 1), each  = 12)),
  Month = (rep(c("Jan", "Feb", "Mar", "Apr", "May",
              "Jun", "Jul", "Aug", "Sep", "Oct",
              "Nov", "Dec"), times = 12)),
  Passangers = as.numeric(AirPassengers),
  t = seq(1: length(AirPassengers))
)
head(data)
```

```
##   Year Month Passangers t
## 1 1949   Jan        112 1
## 2 1949   Feb        118 2
## 3 1949   Mar        132 3
## 4 1949   Apr        129 4
```

```
## 5 1949    May         121 5
## 6 1949    Jun         135 6
```

# 2. Plot the time series. Comment on any visible trends, seasonality, or anomalies that might affect your modeling strategy. (3 marks)

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

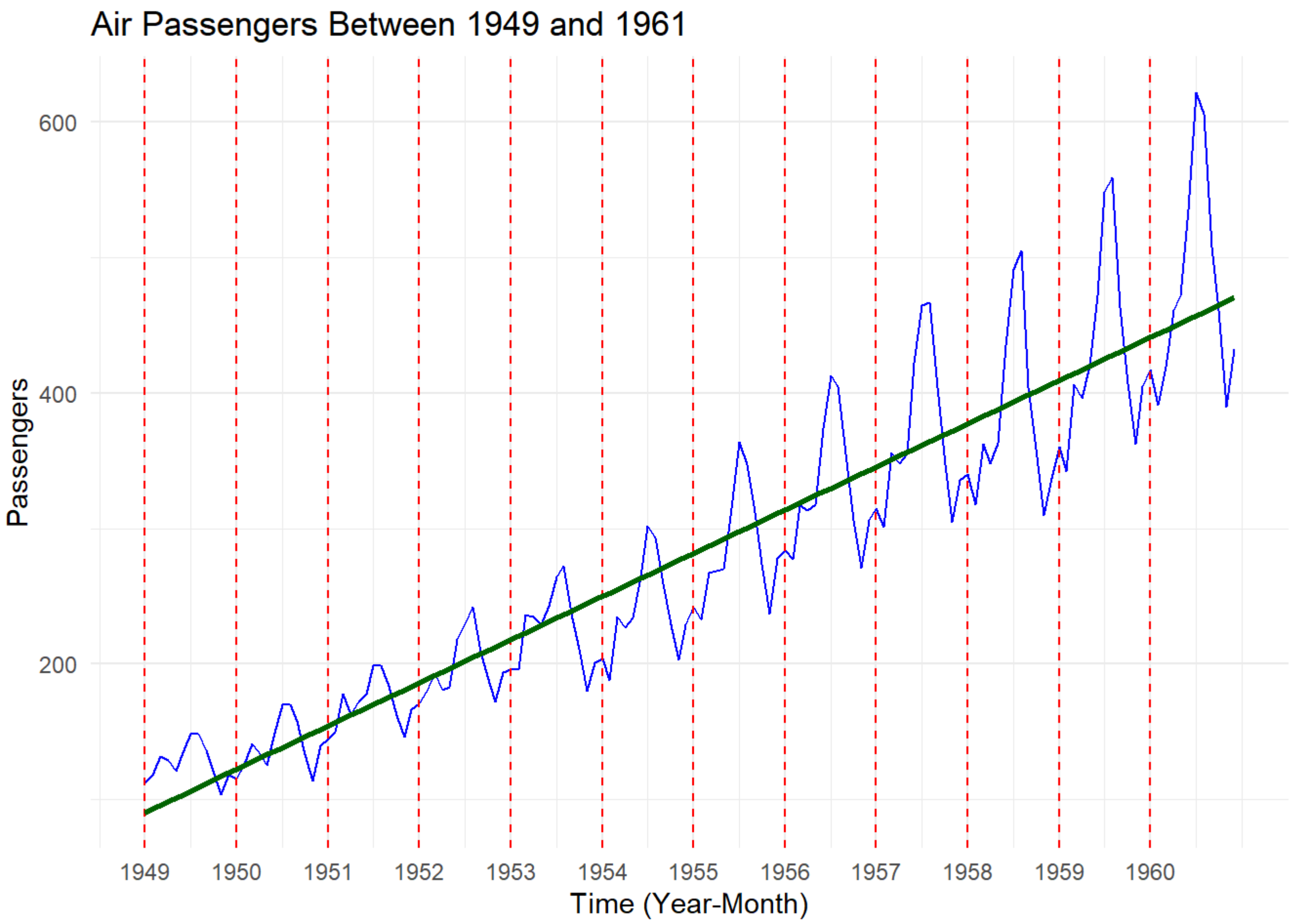
```
data <- data %>%
  mutate(
    YearMonth = paste(Year, Month)
  )

year_breaks <- data %>%
  group_by(Year) %>%
  summarize(start_t = min(t))
```

```
january = data %>% filter(Month == "Jan") %>% pull(t)
```

```
library(ggplot2)
ggplot(data, aes(x = t, y = Passangers)) +
  geom_line(color = "blue") +
  labs(
    title = "Air Passengers Between 1949 and 1961",
    y = "Passengers"
  ) +
  geom_vline(xintercept = january, color = "red", linetype = "dashed")+
  geom_smooth(method = lm, se = F, color = "darkgreen")+
  scale_x_continuous(name = "Time (Year-Month)",
                    breaks = year_breaks$start_t,
                    labels = year_breaks$Year)+
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




> **Observation**
>
> The data seems to be seasonal.
>
> - **January** has the lowest number of passengers. represented by the `vertical red line`
> - The middle of the year has the highest value within the year.
> - There is a spike at the beginning of the year. This spike drops down then rises up to the peak of the year.
> - The second drop of the year is the most severe leading to the next January having the lowest number of air passengers in the next year.
> - After January there is a slight drop before rising for the rest of the year.
>
> The data has an **observable upwards trend** represented by the green line

# 3. Check the stationary of the Air Passengers time series. (3 marks

```
adf.test(data$Passangers, alternative = "stationary")
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data$Passangers
```

```
## Dickey-Fuller = -7.3186, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

> because the parameter measure is **stationary**
>
> $H_0$ The data is non-stationary. This implies that the statistical properties of the series, such as the mean and variance, are dependent on time.
>
> $H_1$ 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.

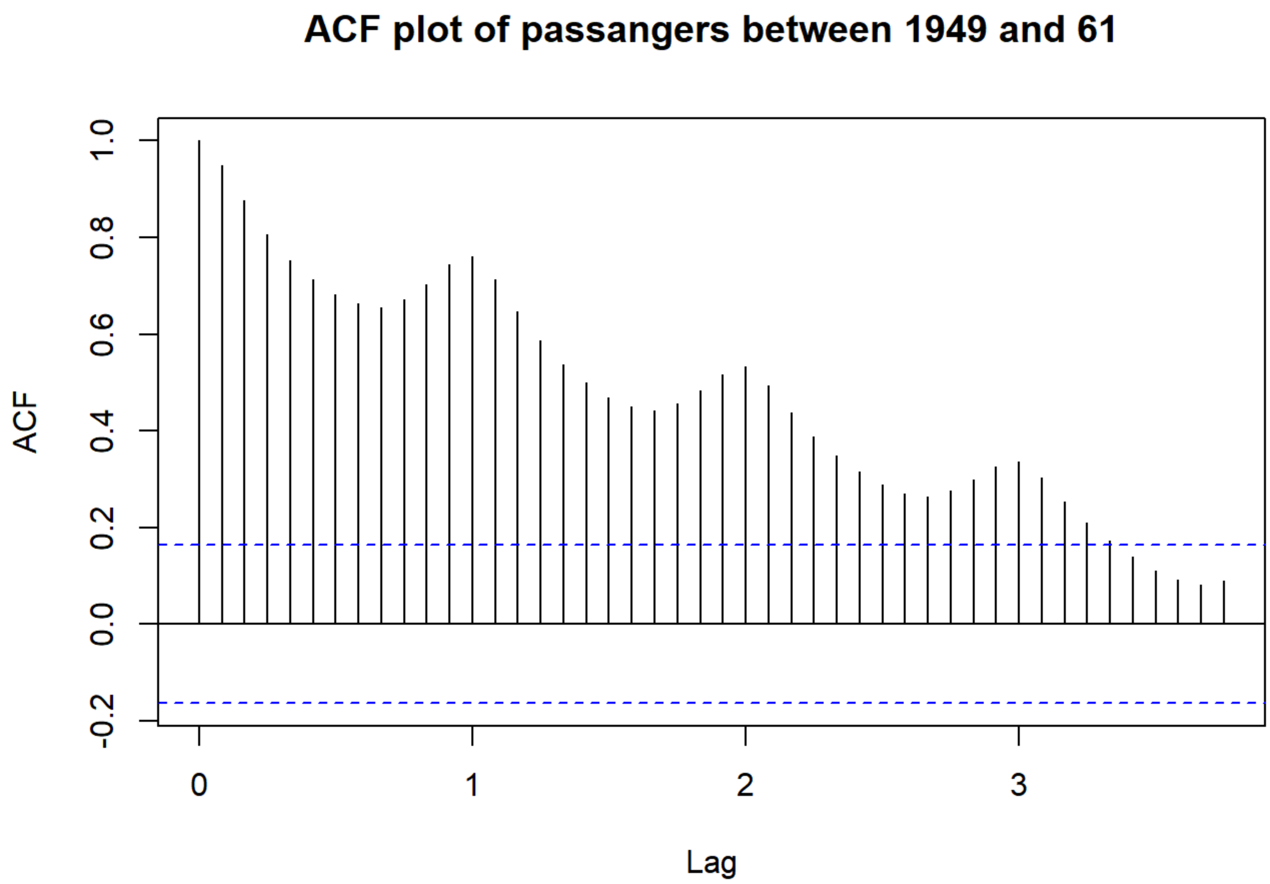
# 4. If the series is non-stationary, apply necessary transformations to make it stationary. Show the transformed series. (3 marks)

> The data is stationary
>
> because of this there is no need for transformation to make the data stationary

# 5. Use the ACF and PACF plots to suggest possible values of $p$ and $q$ for an ARMA model on the stationary series. (3 marks)

## ACF plot

```
acf(AirPassengers, lag.max = 45,main = "ACF plot of passangers between 1949 and 61")
```




Lags up to lag 40 are significant

**Expectations**

- **Positive spikes:** Indicate a positive correlation with past values.
- **Negative spikes:** Indicate a negative correlation with past values.
- **Quick decay to zero:** Suggests a short-term dependency.
- **Slow decay to zero:** Suggests a long-term dependency.

**Observations**

The spikes are becoming smaller and smaller.

- This means that there is a negative correlation between the past values and the predicted future values.
- There is reasonably long-term predictable patterns from the data

The first `40 lag` values. From this lags after 40 have little impact to prediction acting as noise.

**Interpretation**

- The ACF decays slowly to zero, indicating a potential AR component.
- There are no clear cuts off, suggesting a less prominent MA component.

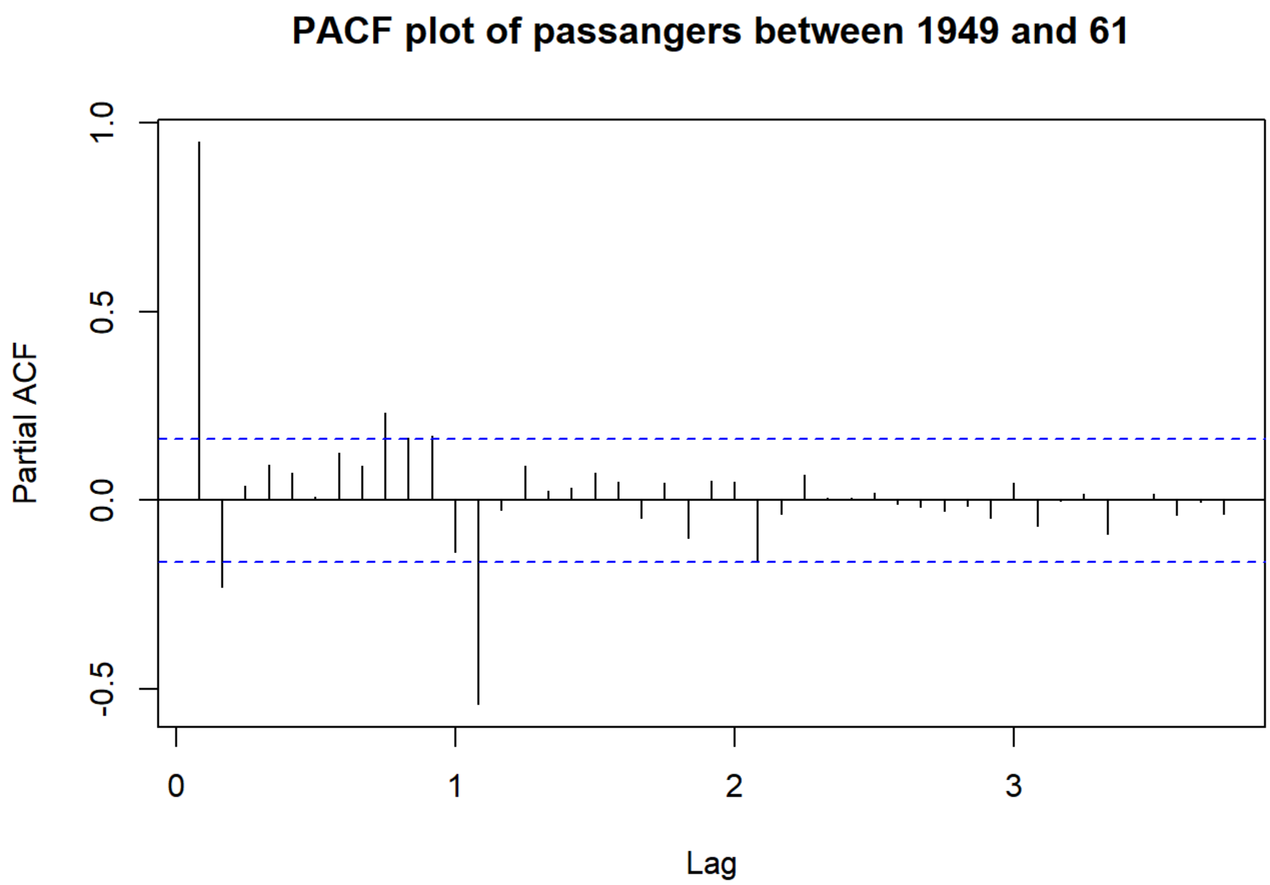
**Because of this**

- **AR order (p):** 2 or 3 (based on the slow decay of the ACF)
- **MA order (q):** 0 or 1 (due to the lack of a clear cutoff)

**Note** The ACF may need transformation because of the spikes as the lag spikes decrease

# PACF

```
pacf(AirPassengers, lag.max = 45,main = "PACF plot of passangers between 1949 and 61")
```




> **Observation**
>
> - **A spike at lag 1:** This indicates a strong direct relationship between the current value and the previous value (lag 1)
> - **No significant lag after lag 1** the relationship between the current value and past values is primarily explained by the direct relationship with the previous value.
>
> **Implications**
>
> - **AR order (p):** 1 (due to the significant spike at lag 1)
> - **MA order (q):** Likely 0, as there are no significant spikes in the ACF plot

The **ARMA** model is likely a ARMA(1, 0) but it may have have noise introduced by the seasonality.

# 6. Think about the seasonality in the original series. How might this influence your choice of p and q? (3marks)

- **Misleading ACF and PACF:** Seasonality can introduce artificial patterns into the ACF and PACF plots, making it difficult to accurately determine the orders `p` and `q` for the ARMA model.

- **Model Underfitting:** If seasonality is ignored, the ARMA model might not capture the underlying patterns in the data, leading to poor forecasting performance.

The predicted p and q need to be modified.

# 7. Based on your plots and seasonal considerations, fit an appropriate ARMA model. (4 marks)

## Stabilize variance by making

```
data$stab_var = log(data$Passangers)
```

## Remove annual seasonality

```
data$no_annual_sn = c(rep(NA, 12), diff(data$stab_var, lag = 12))
```

## Remove the seasons

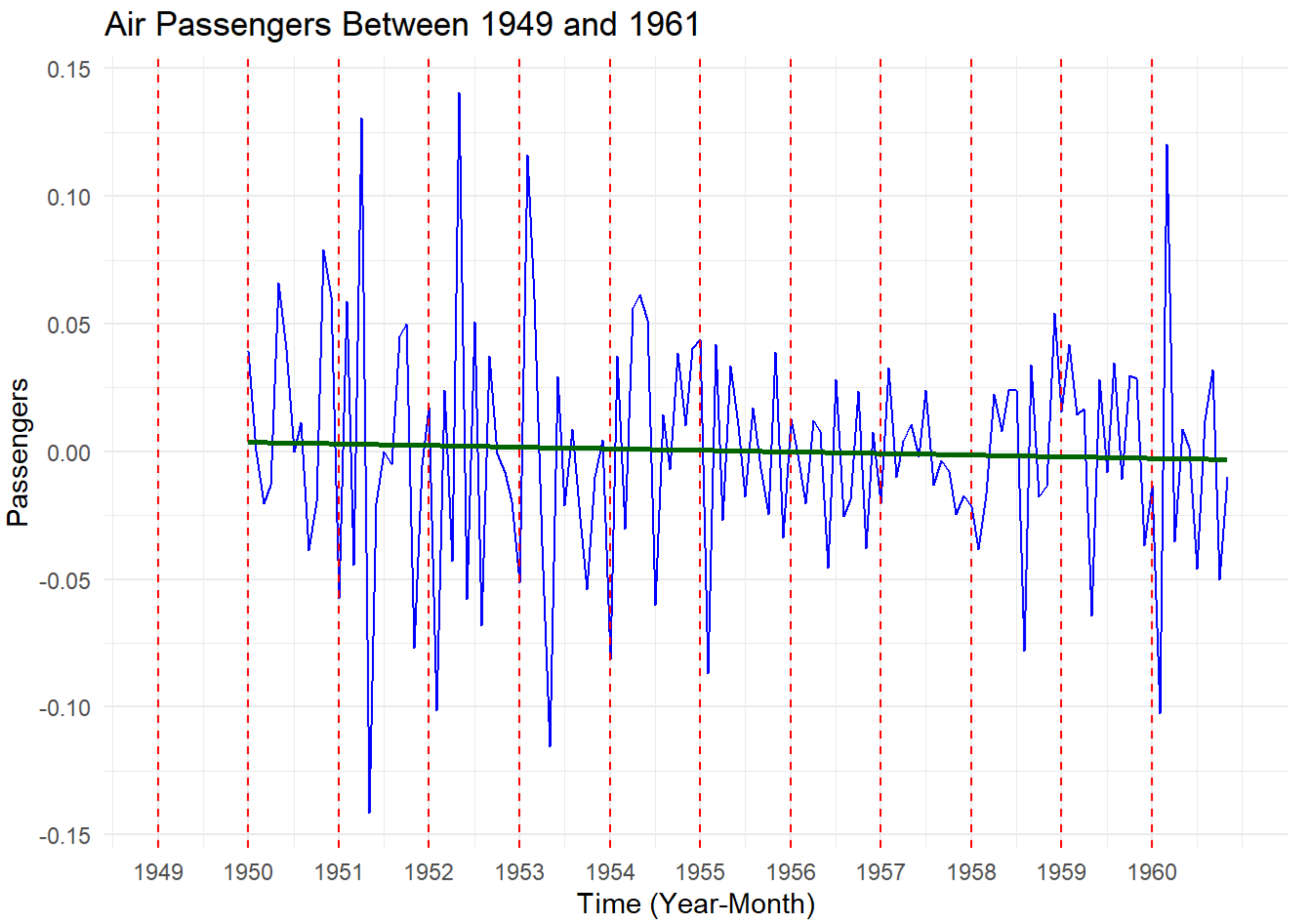
```
data$sec_dif = c(diff(data$no_annual_sn, lag = 1), NA)
```

```
ggplot(data, aes(x = t, y = sec_dif)) +
  geom_line(color = "blue") +
  labs(
    title = "Air Passengers Between 1949 and 1961",
    y = "Passengers"
  ) +
  geom_vline(xintercept = january, color = "red", linetype = "dashed")+
  geom_smooth(method = lm, se = F, color = "darkgreen")+
  scale_x_continuous(name = "Time (Year-Month)",
                     breaks = year_breaks$start_t,
                     labels = year_breaks$Year)+
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

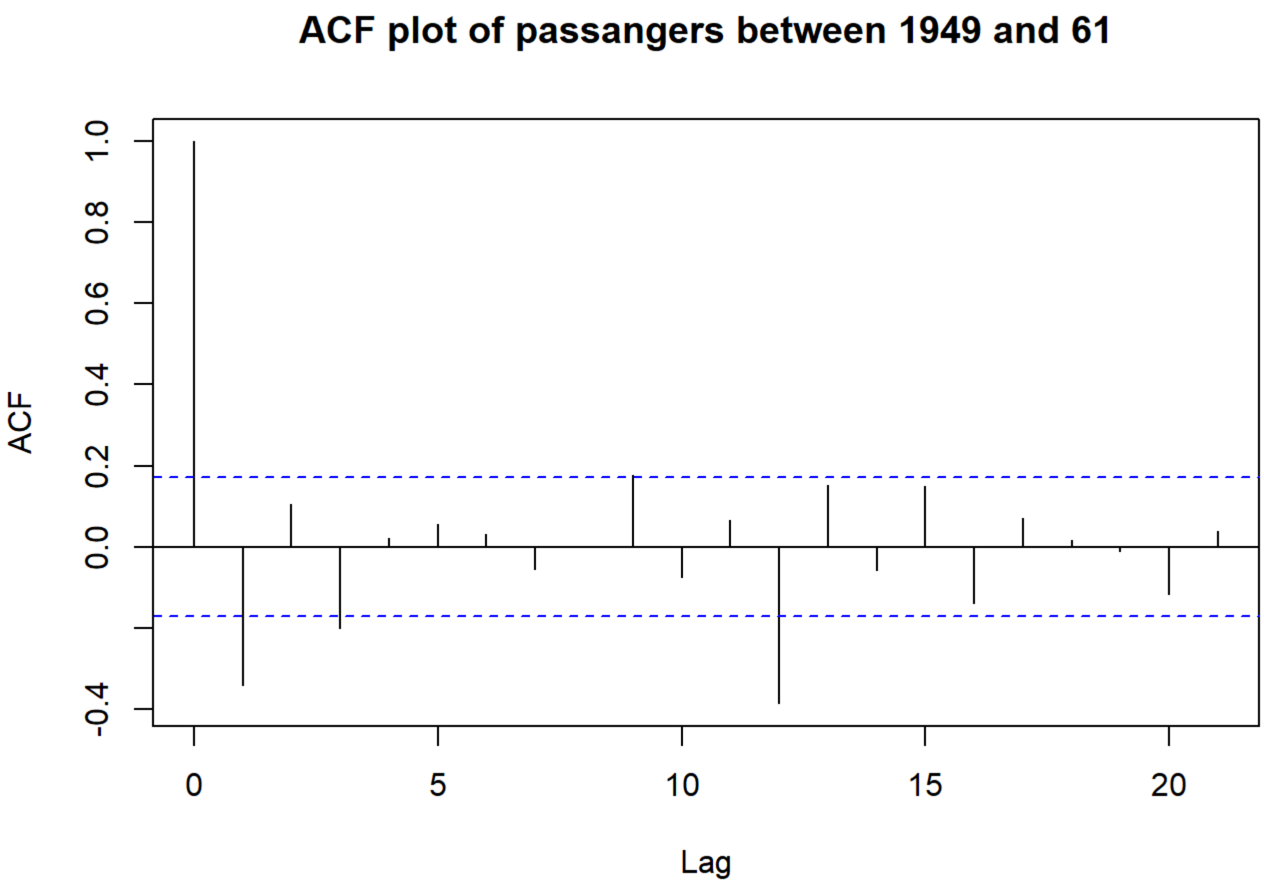
```
## Warning: Removed 13 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 13 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

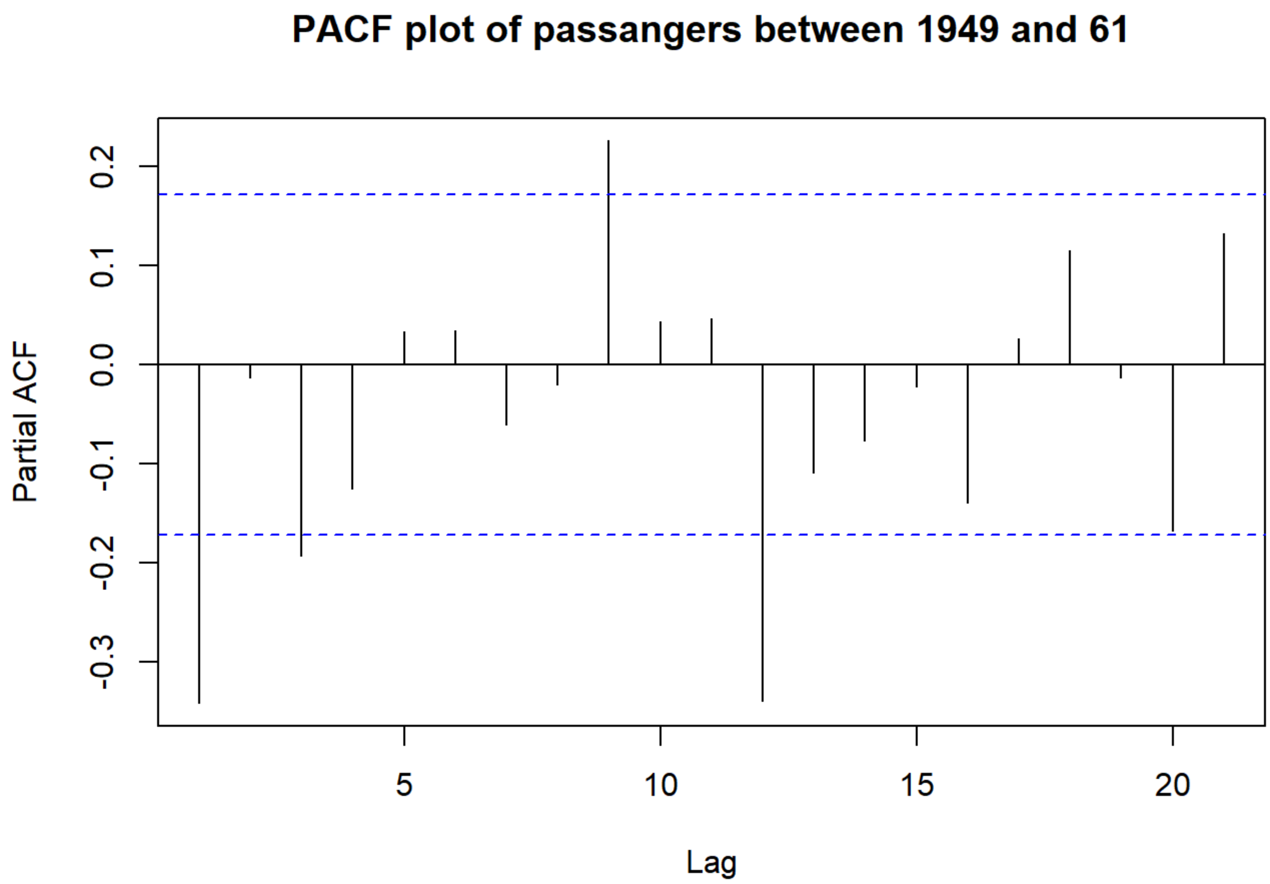



## ACF and PACF

```
acf(ts(na.omit(data$sec_dif)),main = "ACF plot of passangers between 1949 and 61")
```




```
pacf(ts(na.omit(data$sec_dif)),main = "PACF plot of passangers between 1949 and 61")
```




**Observation ACF**

**No clear pattern:** The ACF values fluctuate around zero without a clear pattern of decay or spikes. This suggests that there is little to no autocorrelation in the data.

This suggests that the data might be relatively random with little dependence on past values

**Observation PACF**

There are significant spikes at lags 1 and 13. This suggests a direct relationship between the current value and the previous value (lag 1), as well as a potential seasonal effect at lag 13.

The PACF values after lag 13 fluctuate around zero without a clear pattern, indicating that the relationship between the current value and past values is primarily explained by the direct relationship with the previous value and the seasonal component.

**Implication**

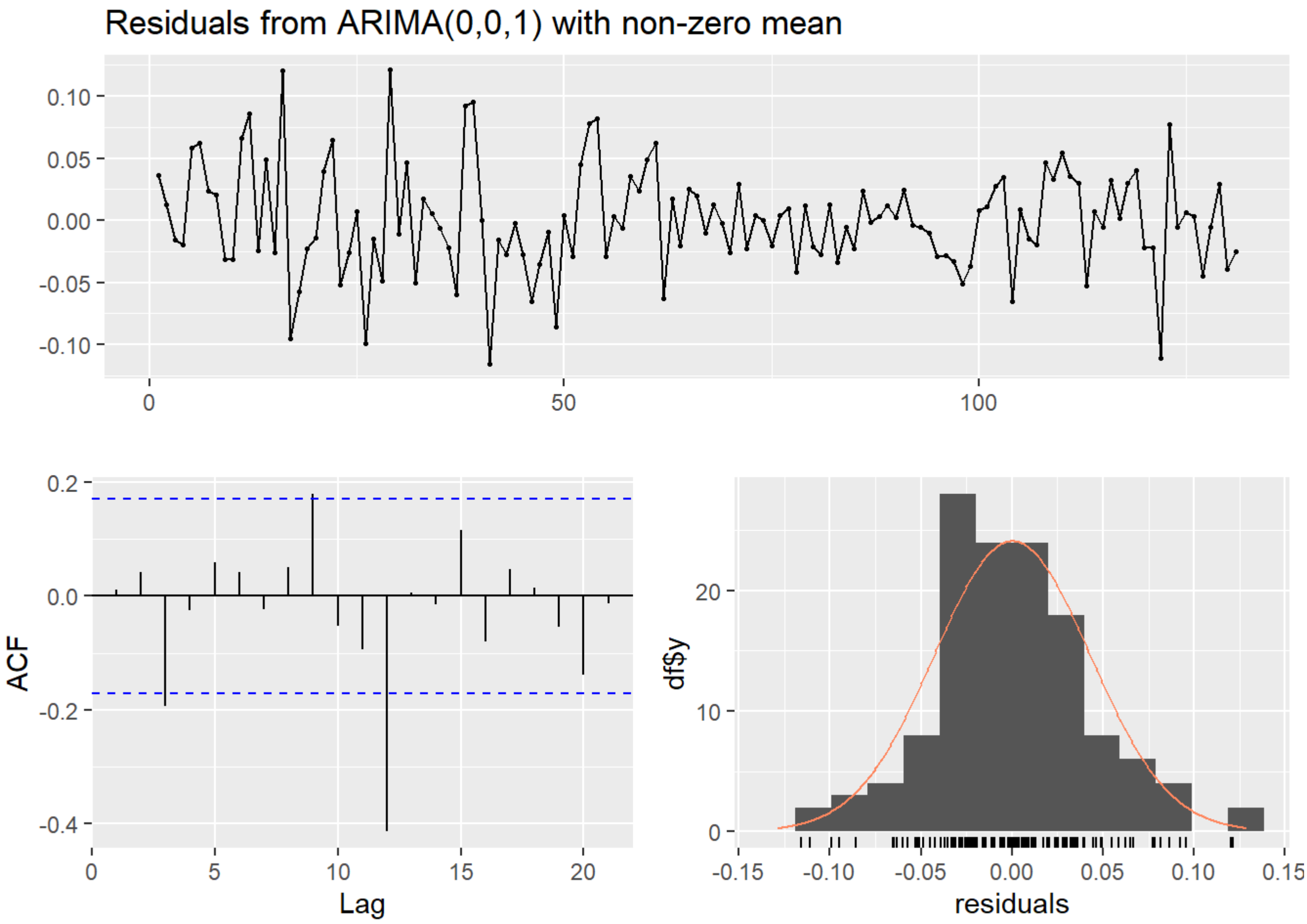
- **AR component:** The significant spike at lag 1 suggests an AR(0) component in the model.
- **Seasonal component:** The significant spike at lag 13 indicates a potential seasonal component with a period of 12 (monthly data).

**Conclusion**

my ARMA is AR(0), d(0), MA(1)

```
arma_model = arima(ts(na.omit(data$sec_dif)), order = c(0, 0, 1)) # ARMA seasonal , seasonal = list(order = c(0, 1, 0), period = 12)
```

```
checkresiduals(arma_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,1) with non-zero mean
## Q* = 11.594, df = 9, p-value = 0.2372
##
## Model df: 1.   Total lags used: 10
```

**Observation**

the $pvalue\ 0.2372 \geq 0.05$

**Inference**

Using ARMA AR(0) MA(1) d(0) the p-value is above 0.05

This means that for this model:

$H_0$ 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 ARMA(2,1) model
2. The model has adequately captured the auto-correlations in the data
3. The model however leaves some patterns unaccounted for

From the plots:

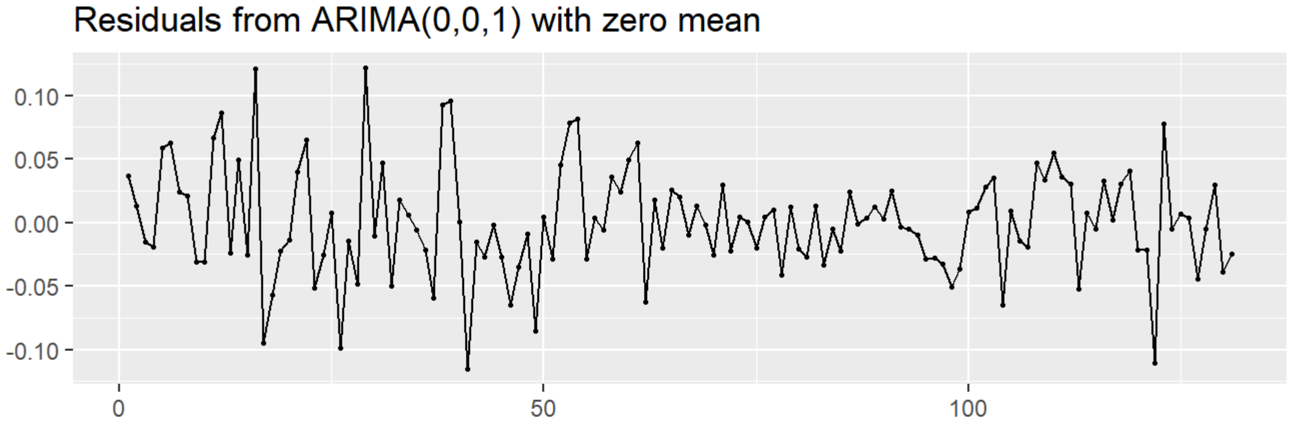
> The new frequency bar graph appears more dense(no gaps between stacks) and less skewed than the previous graph. This could indicate:
>
> 1. The residuals from the model were found to be normally distributed and did not show significant autocorrelation. This means that **Residuals are noise**
> 2. The 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

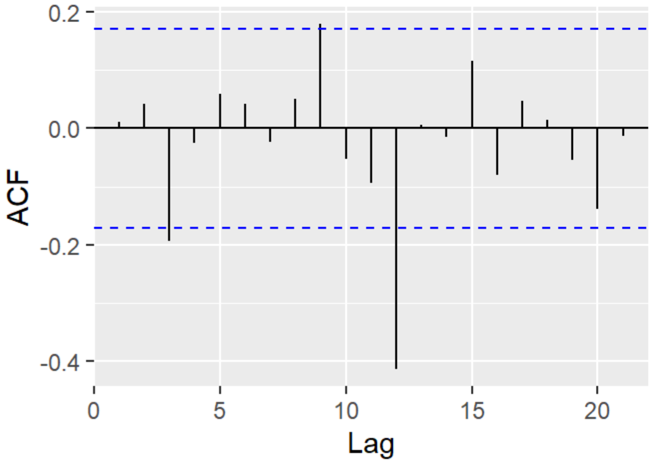
## Confirm my conclusion using built in functions

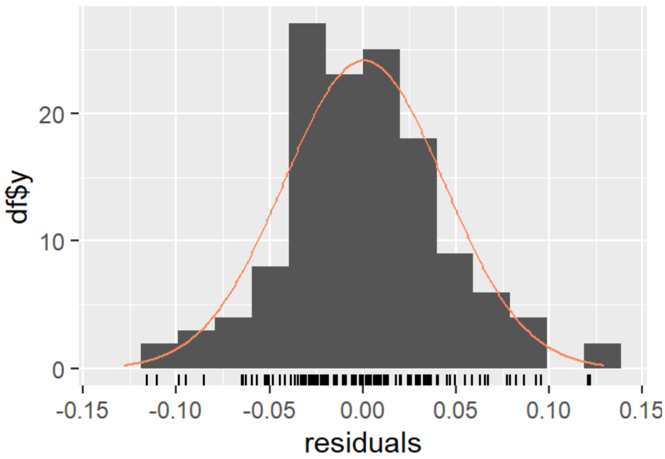
```
q7_auto_arma = auto.arima(ts(na.omit(data$sec_dif)), d = 0, max.d = 0, seasonal = FALSE)
summary(q7_auto_arma)
```

```
## Series: ts(na.omit(data$sec_dif))
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##           ma1
##       -0.3870
## s.e.   0.0887
##
## sigma^2 = 0.001842:  log likelihood = 226.99
## AIC=-449.98   AICc=-449.88   BIC=-444.23
##
## Training set error measures:
##                        ME      RMSE        MAE MPE MAPE      MASE       ACF1
## Training set 0.0005478462 0.0427537 0.03250592 NaN  Inf 0.5810385 0.01141653
```

```
checkresiduals(q7_auto_arma)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,1) with zero mean
## Q* = 11.598, df = 9, p-value = 0.2369
##
## Model df: 1.   Total lags used: 10
```

## conclusion

The new graphs match the ones generated by the manual method.

Based on the built-in function, the ideal ARMA is AR(0), MA(1), d(0)

This matches the approximation from the ACF and PACF in the initial inference

This means that the best arma is in-did `ARMA(0, 1)`

# 8. Fit an ARIMA model to the original Air Passengers series. Discuss your process to automatically select the best model. (4 marks)

*Auto-Regressive (p)* -> Number of autoregressive terms.

*Integrated (d)* -> Number of nonseasonal differences needed for stationarity.

*Moving Average (q)* -> Number of lagged forecast errors in the prediction equation.

```
auto.arima(AirPassengers, seasonal = F) # (RDocumentation, 2024)
```

```
## Series: AirPassengers
## ARIMA(4,1,2) with drift
##
## Coefficients:
##          ar1     ar2      ar3      ar4      ma1      ma2   drift
##       0.2243  0.3689  -0.2567  -0.2391  -0.0971  -0.8519  2.6809
## s.e.  0.1047  0.1147   0.0985   0.0919   0.0866   0.0877  0.1711
##
## sigma^2 = 706.3:  log likelihood = -670.07
## AIC=1356.15   AICc=1357.22   BIC=1379.85
```

> **Using the auto.arima**
>
> The resultant output has the following values
>
> 1. AR(4)
> 2. Integreated(d) 1
> 3. MA(2)
>
> This module was inspired by (Pulagam, 2020) and (Lingaraj Biradar, 2021). They proposed the inclusion of auto arima functions in python and R because:
>
> 1. In the basic ARIMA a person needs to perform differencing and plot ACF and PACF graphs to determine these values which are time-consuming
> 2. For the case of Air passangers the seasonality also makes the ACF and PACF output to be less accurate in predicting the P,Q, D and p, q, d for ARIMA

```
q8_arima = Arima(AirPassengers, order = c(4, 1, 2)) #, seasonal = list(order = c(0, 1,0), perio
d = 12)
```

# 9. Display the model summary and interpret the results. Think about the ARIMA specifications of the model and if you agree with the choice. (5 marks)

```
summary(q8_arima)
```

```
## Series: AirPassengers
## ARIMA(4,1,2)
##
## Coefficients:
##          ar1     ar2      ar3      ar4     ma1      ma2
##       0.2049  0.3601  -0.2507  -0.2229  0.0685  -0.7103
## s.e.  0.1087  0.1141   0.0980   0.0909  0.0838   0.0749
##
## sigma^2 = 814:  log likelihood = -679.78
## AIC=1373.56   AICc=1374.39   BIC=1394.3
##
## Training set error measures:
##                   ME     RMSE      MAE      MPE     MAPE      MASE        ACF1
## Training set 6.51499 27.82944 21.60696 1.904148 7.603494 0.6745787 -0.04509642
```

**Observation**

**Model Type**: ARIMA(4, 1, 2)

- **AR (4)**: The model includes four auto regressive terms. This suggests that the model predicts future values based on the values from the previous four time points.
- **I (1)**: The data has been differenced once to make it stationary, which is common for handling trends in time series data like Air Passengers which typically exhibits a non-stationary trend.
- **MA (2)**: The model includes two moving average terms, indicating that the prediction error is modeled using the errors from the two previous forecasts.

**Coefficients**

- **AR1 (0.2049), AR2 (0.3601), AR3 (-0.2507), AR4 (-0.2229)**: These coefficients represent the weights given to the past four observations. The mix of positive and negative values can indicate oscillating effects from previous periods.
- **MA1 (0.0685), MA2 (-0.7103)**: These indicate how previous forecast errors influence the current prediction. The notably larger negative coefficient for MA2 suggests a substantial adjustment in response to the error from two steps ago, which may be correcting overestimations or underestimations made by the first MA term.

The coefficients for both AR and MA terms are relatively small but significant given their standard errors. The signs of the coefficients suggest a complex interaction between past values and the predicted values

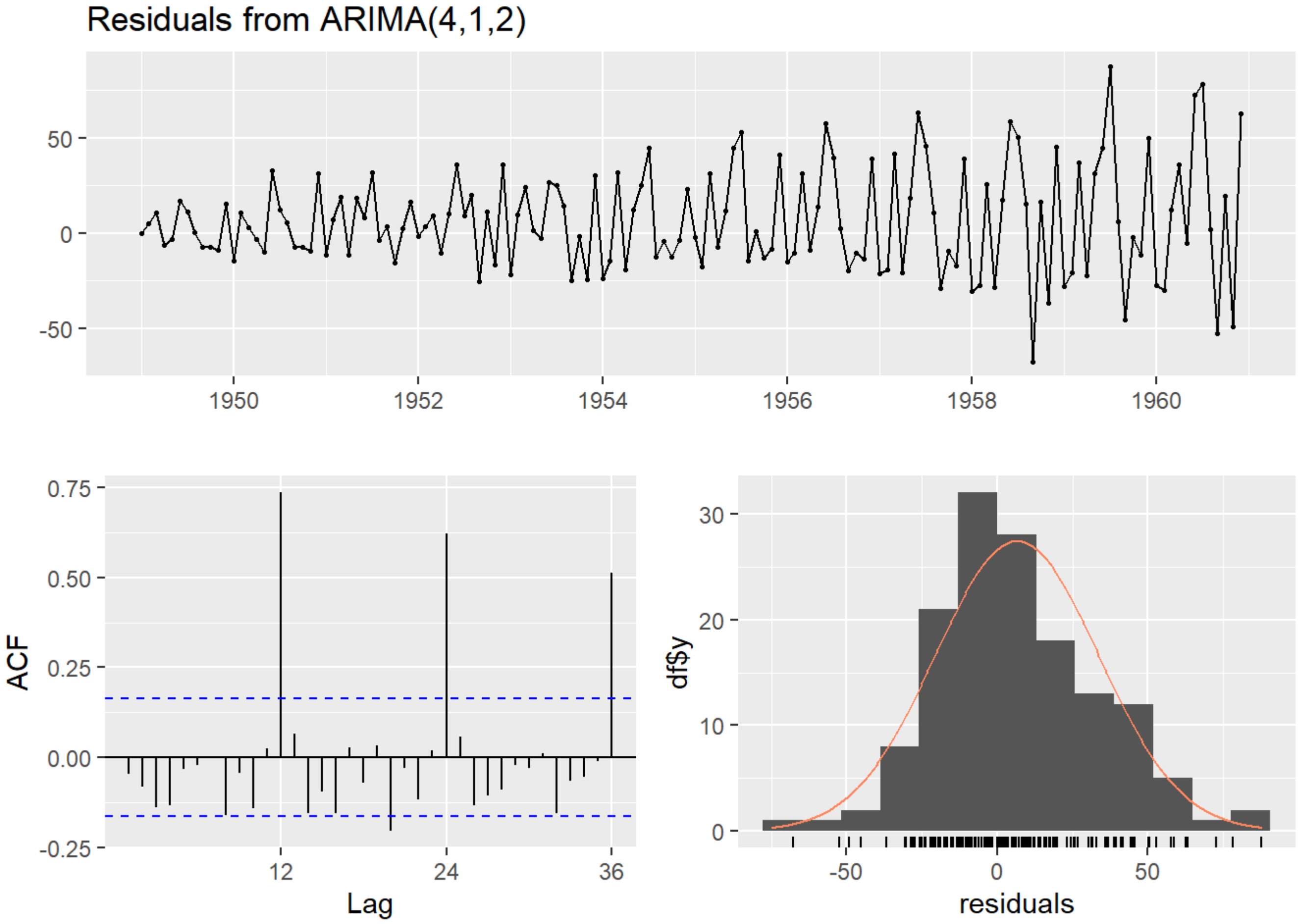
This is justified with the evidence of non-stationarity.

## However

The AIC value however indicates a potential for over-fitting

# 10. Perform diagnostic checks on your fitted ARIMA model. Are there any hidden patterns that might have been missed? (3 marks)

```
checkresiduals(q8_arima)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(4,1,2)
## Q* = 189.24, df = 18, p-value < 2.2e-16
##
## Model df: 6.   Total lags used: 24
```

**Ljung-Box test**

**P-value** the p-value is $2.2 * 10^{-16}$

**Plots**

- the frequency bar-graph for the **residuals** is skewed to the left.
- The ACF has spikes after lag 1 indicating that the residuals have no autocorrelation
- The residual data has an increasing variability meaning the residuals are not stationary

**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 **significant evidence of autocorrelation** in the residuals of your ARIMA(4,1, 2) model
2. The model **has not** adequately captured the auto-correlations in the data
3. The model leaves some patterns unaccounted for

> **Residuals are not noise** . The model does not adequately capture all the correlation and variation in the data

# 11. Discuss the results of your diagnostic checks. Are there any indications that your model is not adequate? How would you address these issues? (4 marks)

`Yes there are indicators that the model is not adequate`

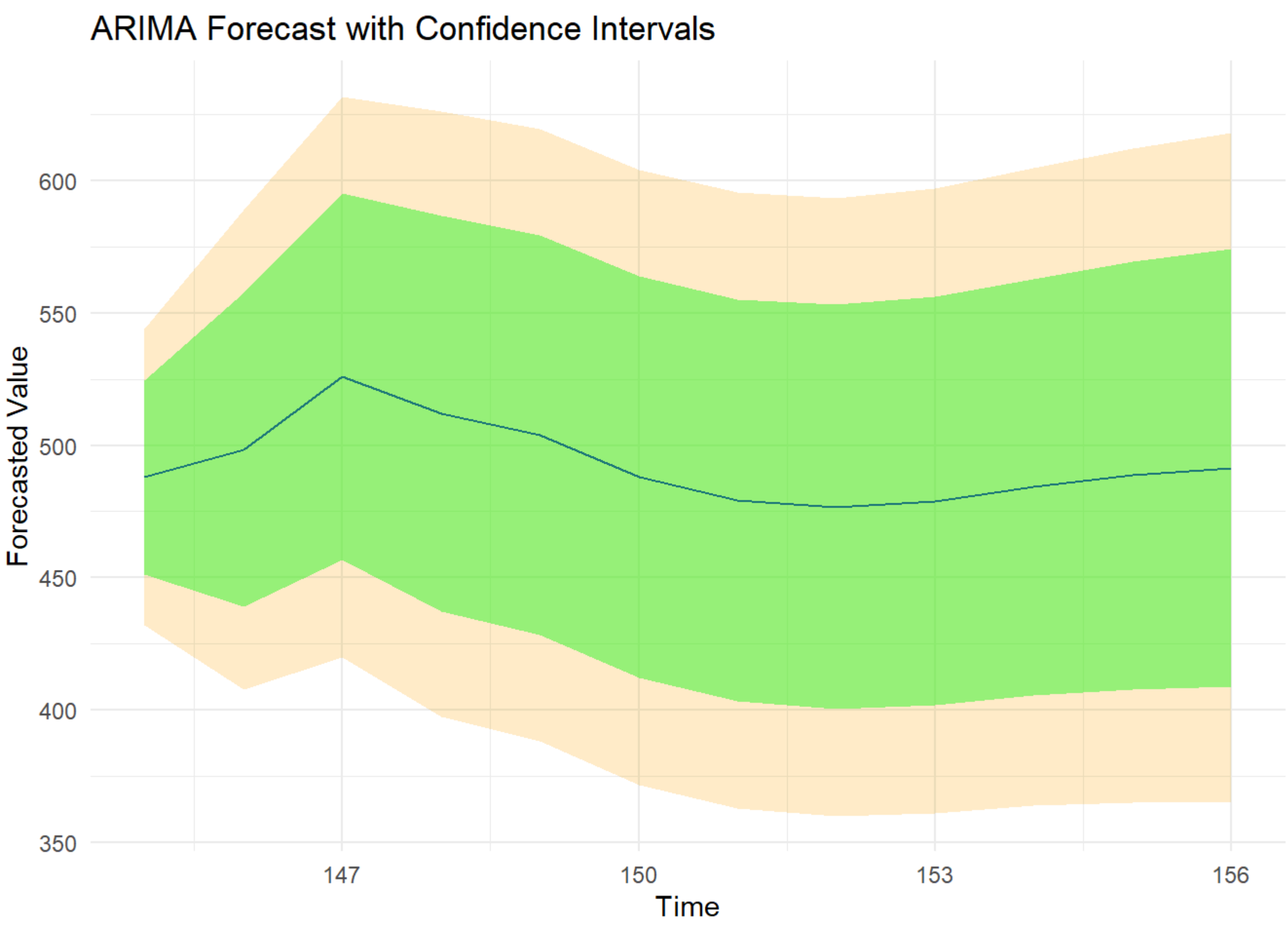
- I Could add a seasonal ARIMA to account for the potential lag spokes in the data
- There may be patterns in the residuals that are not accounted for
- Solve for autocorrelation in the data and handle the autocorrelation

# 12. Generate and plot a 12-month forecast using your fitted ARIMA model. Consider the uncertainty in your forecast. (5 marks)

```
forecst_Data = forecast(q8_arima, h=12)
```

```
forcst_plot = data.frame(
  time = seq(145, 156),
  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(forcst_plot, 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()
```




1. **Blue Line** represents the point forecast from the ARIMA model for each future time point
2. **Green Shaded Area** areas represent the 80% confidence intervals for the forecast
3. **Orange Shaded Area** 95% confidence intervals

**Interpretation**

The expansion of the confidence intervals over time, as seen from the widening bands, indicates increasing uncertainty in the predictions as you project further into the future.

# 13. Interpret the forecast results. How accurate are they, and what do they suggest about future values of the series? Discuss the limitations and potential improvements.(5 marks)

## Target

- A lower $\sigma^2$ indicates a better fit of the model to the data
- Higher (less negative) log likelihood values indicate a better model fit
- for **AIC** and **BIC**: Lower values generally indicate a better model. The relatively close values of AIC and BIC suggest a balance between model complexity and fit.
- A positive ME indicates a tendency of the model to overestimate the data.
- RMSE provides a measure of the average magnitude of the forecast errors. The value indicates how much the predictions deviate, on average, from the observed values.

- ACF1 if autocorrelation of residuals at lag 1 is close to zero, it suggesting that there is little to no autocorrelation left in the residuals, which is desirable.

## Observed Values

- **ACF1** is `-0.04509642`
- **ME** is `6.51499`

#### Limitations and Improvements

1. The model appears to fit the data reasonably well, as indicated by the low MASE and nearly zero ACF1
2. The positive ME and very slight MPE suggest a slight over-fitting tendency
3. There might be room for simplification or adjustment to reduce potential over-fitting or to capture more nuances in the underlying data pattern.

## A new model to try and improve the accuracy

To try and reduce the over-fit I have made a train and test set.

```
set.seed(123)
n = length(AirPassengers)
samp = sample(1:n, round(0.2 * n))

test_indices = rep(FALSE, n)
test_indices[samp] = TRUE
train_indices = !test_indices

test_data = AirPassengers[test_indices]
train_data = AirPassengers[train_indices]

test_len = length(test_data)
```

```
auto.arima(train_data, seasonal = F)
```

```
## Series: train_data
## ARIMA(2,1,2) with drift
##
## Coefficients:
##          ar1      ar2      ma1     ma2   drift
##       1.3627  -0.7564  -1.7131  0.7578  3.2659
## s.e.  0.0770   0.0690   0.1013  0.0938  0.3768
##
## sigma^2 = 1192:  log likelihood = -564.63
## AIC=1141.26   AICc=1142.05   BIC=1157.68
```

```
new_q8_arima = Arima(train_data, order = c(2, 1, 2))
summary(new_q8_arima)
```

```
## Series: train_data
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.5074  -0.8713  -1.7703  0.9258
## s.e.  0.0522   0.0546   0.0694  0.0615
##
## sigma^2 = 1280:  log likelihood = -569.36
```

```
## AIC=1148.73   AICc=1149.28   BIC=1162.41
##
## Training set error measures:
##                    ME    RMSE     MAE      MPE     MAPE      MASE      ACF1
## Training set 6.480609 34.9867 25.3986 1.774283 8.926172 0.8612256 0.0119795
```

> The reduction in the
>
> 1. ME
> 2. RMSE
> 3. ACF1
> 4. MASE
>
> indicate that this new model is more accurate with less over-fitting.

# 14. Fit a seasonal ARIMA model to the Air Passengers data set (3 marks)

```
seasonal_arima = auto.arima(AirPassengers)
```

```
seasonal_arima
```

```
## Series: AirPassengers
## ARIMA(2,1,1)(0,1,0)[12]
##
## Coefficients:
##          ar1     ar2      ma1
##       0.5960  0.2143  -0.9819
## s.e.  0.0888  0.0880   0.0292
##
## sigma^2 = 132.3:  log likelihood = -504.92
## AIC=1017.85   AICc=1018.17   BIC=1029.35
```

# 15. Compare the seasonal ARIMA model with the non-seasonal ARIMA model in terms of AIC/BIC values and forecast accuracy. Consider if seasonality is being captured adequately by the seasonal model.

| | ARIMA(4, 1, 2) | New ARIMA(2, 1, 2) | SEASONAL ARIMA(2, 1, 1)(0, 1, 0)[12] |
|---|---|---|---|
| **AIC** | 1373.56 | 1148.73 | 1017.85 |
| **BIC** | 1394.3 | 1162.41 | 1029.35 |
| **Forecast Accuracy** $\log likelihood$ | -679.78 | -569.36 | -504.92 |

Using the log likelihood to test the fit of the model to the data provided for training

**Models**

- **ARIMA(4, 1, 2)**: A non-seasonal model with four autoregressive terms, one differencing step, and two moving average terms.
- **New ARIMA(2, 1, 2)**: A simpler non-seasonal model with two autoregressive terms, one differencing step, and two moving average terms.

- **Seasonal ARIMA(2, 1, 1)(0, 1, 0)[12]**: Incorporates both non-seasonal and seasonal components, with two non-seasonal AR terms, one non-seasonal differencing step, one non-seasonal MA term, one seasonal differencing step, and an implicit seasonal pattern modeled over a period (likely 12, which could represent months in a year).

**Expectation**

- Higher (less negative) log likelihood values indicate a better model fit
- Lower AIC (Akaike Information Criterion) values are better. It balances model fit and complexity, with penalties for more parameters.
- BIC(Bayesian Information Criterion) adds more penalties to the model. But the lower it is the better the model.

> ### Observation
>
> 1. The Seasonal ARIMA model shows the lowest AIC, indicating it might be providing a better fit with fewer penalties for complexity compared to the others.
> 2. Again, the Seasonal ARIMA model has the lowest BIC, suggesting it is the most efficient model among the three in terms of balancing goodness of fit and model simplicity.
> 3. The Seasonal ARIMA model has the highest (least negative) log likelihood value, indicating it fits the data better than the two non-seasonal models.
>
> ### Interpretation
>
> - The seasonal ARIMA model appears to be the most effective model among the three, given its lowest AIC and BIC scores and the highest log likelihood
> - The shift from ARIMA(4, 1, 2) to New ARIMA(2, 1, 2) shows a decrease in AIC and BIC, which suggests that reducing the number of parameters `simplifying the model` does not overly compromise the model's ability to fit the data and actually results in a model that uses the least features to achieve similar results.
> - However, the Seasonal ARIMA model, while more complex with its seasonal components, provides a better fit and justifies its additional complexity with improved performance metrics
>
> ### Conclusion
>
> The data exhibits strong seasonal patterns, which the Seasonal ARIMA model is successfully capturing, as indicated by its superior performance metrics

# References


Hayes, S. (2022, August 31). *Finding seasonal trends in time-series data with python*. Medium. https://towardsdatascience.com/finding-seasonal-trends-in-time-series-data-with-python-ce10c37aa861

Lingaraj Biradar, A. (2021, June 16). *RPubs - time series analysis of airline passengers using R*. Rpubs.com. https://rpubs.com/Anilbiradar/time_series_sarima

Oreilly. (2021, April 21). *4. dealing with trends and seasonality - anomaly detection for monitoring [book]*. Www.oreilly.com. https://www.oreilly.com/library/view/anomaly-detection-for/9781492042341/ch04.html

Pulagam, S. (2020, June 27). *Time series forecasting using auto ARIMA in python*. Medium. https://towardsdatascience.com/time-series-forecasting-using-auto-arima-in-python-bb83e49210cd

RDocumentation. (2024). *Auto.arima function - rdocumentation*. Rdocumentation.org. https://www.rdocumentation.org/packages/forecast/versions/8.23.0/topics/auto.arima