NANYANG TECHNOLOGICAL UNIVERSITY
AI6123-TIME SERIES ANALYSIS
Assignment 1 Individual
NTAMBARA Etienne G2304253K
May 3, 2024

NTAMBARA Etienne G2304253K [1 2 3] May 3, 2024

1. Intoduction

An excellent example of using the Autoregressive Integrated Moving Average ARIMA model is the wwwusage time series data, which is the total number of users connected to the internet via a server and is gathered at one-minute intervals across a 100 observations.The purpose of this assignment is to investigate the use of ARIMA modeling to predict patterns of internet usage. This analysis attempts to determine the best model for predicting future internet usage by fitting appropriate ARIMA models to the data, performing diagnostic checks to verify the model's assumptions, and comparing model performance using metrics like the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Hannan-Quinn Information Criterion (HQC). The conducted results are presented in the next sections of this report.

2. Presenting Conducted Results Step by Step

2.1. Data Loading

The first step in order to work on our wwwusage dataset is to load data, this is the result of loaded data of 100 observations:

```
Loaded Dataset
----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
  [1]  88  84  85  85  84  85  83  85  88  89  91  99 104 112 126 138 146 151
 [19] 150 148 147 149 143 132 131 139 147 150 148 145 140 134 131 131 129 126
 [37] 126 132 137 140 142 150 159 167 170 171 172 172 174 175 172 172 174 174
 [55] 169 165 156 142 131 121 112 104 102  99  99  95  88  84  84  87  89  88
 [73]  85  86  89  91  91  94 101 110 121 135 145 149 156 165 171 175 177 182
 [91] 193 204 208 210 215 222 228 226 222 220
```

Figure 1. Loaded dataset of 100 observations.

2.2. Check Stationary or Nan-Stationary using Plot

The time series plot of WWW usage shows a non-stationary pattern due to the presence of trends and possible seasonality, as proved by the systematic changes in the mean over time. To fit an ARIMA model, differencing is required to remove these non-stationary components and achieve stationary, figure(2) for visualization and checking in order to take decision.

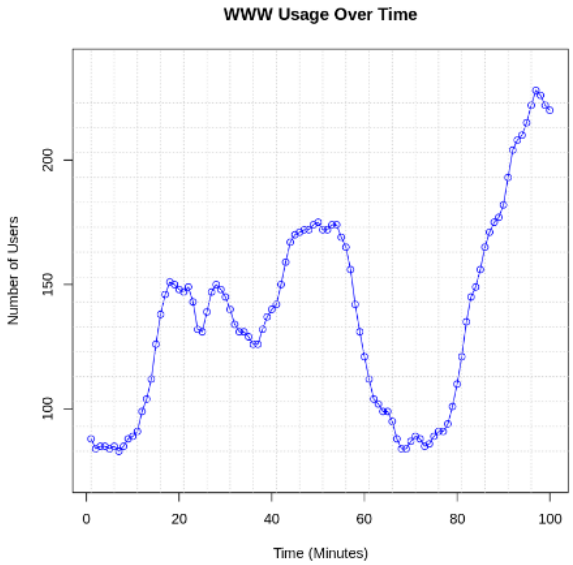


Figure 2. Original Plot of dataset.

Above figure 2 shows that our time series is not stationary because between time 20-40 and 60-80 the variance is very high that's means the change in variance over time. Also, the mean is not constant because of upward trending.

2.3. ACF and PACF Plot for Original Dataset

To determine the order of AR (AutoRegressive) and MA (Moving Average) terms in ARIMA model, which is important for modeling and forecasting the time series data we need to plot ACF for **MA order** and PACF for **AR order**. plotted results are presented below:

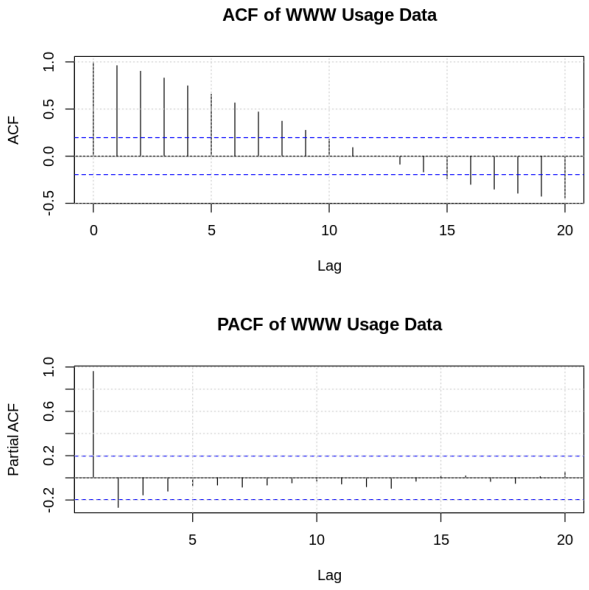


Figure 3. ACF and PACF Orginal data plot.

The ACF and PACF plots (figure: 3) proves non-stationarity in the original dataset, as indicated by the slow decay of the ACF plot. This typically presents the presence of a trend or seasonal pattern. The strong initial correlation that tails off over time as shown in the ACF, alongside the sharp drop after the 1st lag in the PACF, often suggests that differencing (at least once) may be necessary to achieve stationarity for appropriate ARIMA modeling. Stationarity is a key assumption for ARIMA models and these plots guide the shortlisting of the differencing order 'd' and the autoregressive (**AR**) 'p' and moving average (**MA**) 'q' parameters. I implemented codes to choose the best model fit for 1st and 2st order differencing instead of looking on simple model I tried to look for complex model which contain also the simple models, the codes will be shown in the appendix section.

3. Differencing

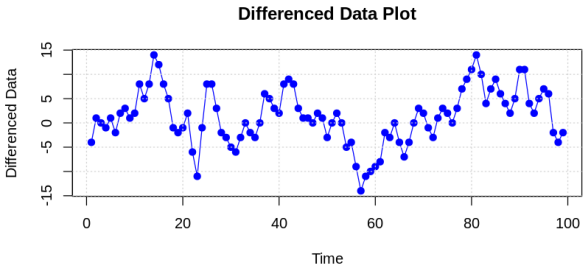
Plot (figure: 4) the differenced "wwwusage" time series data indicates that the mean may have stabilized through differencing, since the data appears **stationary** with mean fluctuations near zero. This might support the use of an ARIMA model with at least one differencing level (d ¿= 1). The process of selecting a suitable ARIMA model would require analyzing several combinations of ARIMA(p,d,q) models according to AIC, BIC, and HQC values in addition to performing diagnostic tests for residual randomness and autocorrelation. The fitted model should pass diagnostic tests and minimize AIC, BIC, and HQC.

Differencing formulas: It is easily seen that $\nabla Z_t = Z_t - Z_{t-1}$ is a MA(1) for d= 1, and $\nabla^2 Z_t = Z_t - 2Z_{t-1} + Z_{t-2}$ is a MA(2) for d=2. Similarly one may see that $\nabla^k Z_t$ is a MA(k) model. Therefore it is stationary.

First Order Differencing d=1

We neeed to use this fomula: $\nabla Z_t = Z_t - Z_{t-1}$

Figure 4. One Time Differencing .



3.1. ACF AND PACF after one time differencing

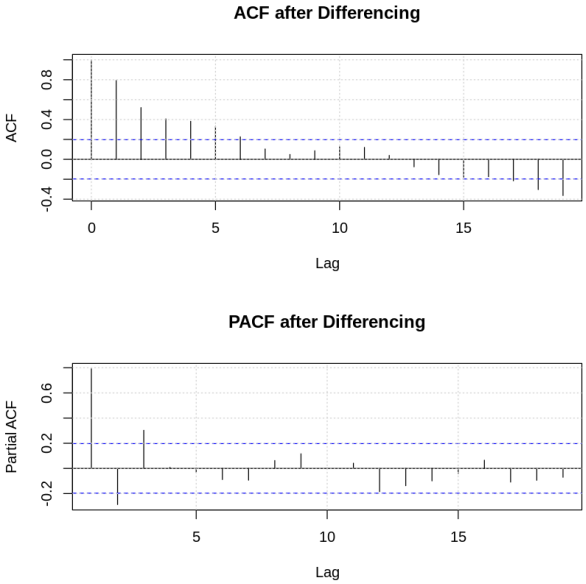


Figure 5. ACF AND PACF after one time Differencing

Plotting the differenced 'wwwusage' time series data reveals the ACF and PACF (see Figure: 5). The observed sharp decline in correlation after differencing, as suggested by the ACF plot, indicates that differencing may have successfully removed seasonality or a trend from the data. The PACF plot, which exhibits prominent spikes at specific lags, aids in determining the order of the autoregressive (AR) component for an ARIMA model. The absence of a clear decay pattern or distinct cut-off in these plots suggests the potential applicability of a mixed ARIMA model. In such cases, the number of significant lags identified in the PACF plot influences the selection of the 'p' parameter, which defines the order of the AR component.

Currently, I have considered relatively simple ARIMA models such as ARIMA(1,1,1) and ARIMA(3,1,0). To accurately determine the best fitting model, I have explored a more simpler and complex model for both 1st and 2st order differencing, evaluating them across three different criteria: In machine learning, we need to make trial and error. I did this to get other fitted models to our dataset. The codes does these tasks are also available in the submitted codebase. AIC, BIC), and HQC. The analysis identified the best-fit models, which are detailed in (table 1).

Criterion	Differencing 1	Differencing 2
AIC	ARIMA(5,1,4)	ARIMA(5,2,5)
BIC	ARIMA(1,1,1)	ARIMA(2,2,0)
HQC	ARIMA(3,1,0)	ARIMA(2,2,0)

Table 1. Best ARIMA Models Based on AIC, BIC, and HQC for Differencing 1 and 2

The outcome of the ARIMA(1,1,1) model displays one **MA** term (ma1 = 0.5256) and one **AR** term (ar1 = 0.6504), along with a list of standard errors suggesting that the coefficients are probably significant. The model's AIC of 514.3 and estimated variance of the residuals (sigma^2) of 9.793 indicate its relative quality in comparison to other models. A practically zero initial autocorrelation of residuals (ACF1) indicates no substantial autocorrelation in the model's residuals. Error metrics on the training set, such as the mean error (ME) and root mean square error (RMSE), and others from (table 2), show the model's predictive ability.

Table 2. ARIMA(1,1,1) Model Output

Call:

```
arima(x = data_set, order =
arima_111_order)
```

Coefficients:

$$\begin{aligned} \text{ar1} &= 0.6504 & \text{ma1} &= 0.5256 \\ \text{s.e.} &= 0.0842 & \text{s.e.} &= 0.0896 \end{aligned}$$

σ^2 estimated as 9.793: log likelihood = -254.15, aic = 514.3

Training set error measures:

$$\begin{aligned} \text{ME} &= 0.3035616 \\ \text{RMSE} &= 3.113754 \\ \text{MAE} &= 2.405275 \\ \text{MPE} &= 0.2805566 \\ \text{MAPE} &= 1.917463 \\ \text{MASE} &= 0.5315228 \\ \text{ACF1} &= -0.01715517 \end{aligned}$$

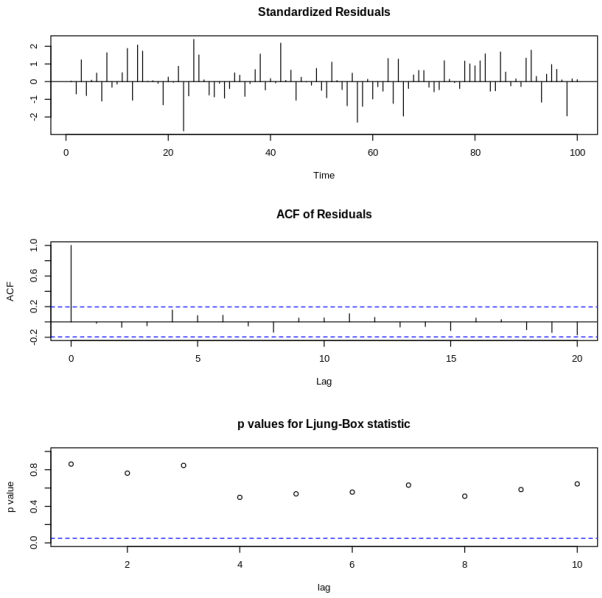


Figure 6. ARIMA(1,1,1) Dignostic

ARIMA(1,1,1): This model's diagnostic plots (figure: 6) similarly showed a satisfactory fit, with the residuals showing little autocorrelation and the Ljung-Box test p-values being non-significant(exceeding the common threshold of 0.05). A little less efficient fit but still a strong model is indicated by the AIC = 514.2995, and HQC = 514.4082 being slightly higher than those for the ARIMA(3,1,0) model.

3.2. ARIMA(3,1,0) Dignostic one time differencing

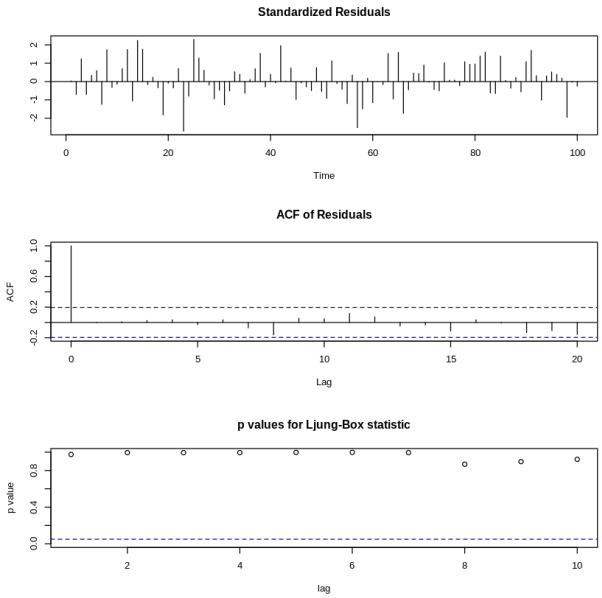


Figure 7. ARIMA(3,1,0) Dignostic

Table 3. ARIMA(3,1,0) Model Output

Call:

```
arima(x = data_set, order =
arima_310_order)
```

Coefficients:

$$\begin{aligned} \text{ar1} &= 1.1513 & \text{s.e.} &= 0.0950 \\ \text{ar2} &= -0.6612 & \text{s.e.} &= 0.1353 \\ \text{ar3} &= 0.3407 & \text{s.e.} &= 0.0941 \end{aligned}$$

σ^2 estimated as 9.363: log likelihood = -252, aic = 511.99

Training set error measures:

$$\begin{aligned} \text{ME} &= 0.230588 \\ \text{RMSE} &= 3.044632 \\ \text{MAE} &= 2.367157 \\ \text{MPE} &= 0.2748377 \\ \text{MAPE} &= 1.890528 \\ \text{MASE} &= 0.5230995 \\ \text{ACF1} &= -0.003095065 \end{aligned}$$

Three autoregressive terms with coefficients of 1.1513, -0.6612, and 0.3407 in the ARIMA(3,1,0) model each have comparatively small standard errors, indicating that the estimations of these coefficients are accurate. The model has a log likelihood of -252 and an AIC of 511.99, which is less than that of the ARIMA(1,1,1) model, possibly indicating a better fit. The estimated variance of the residuals in the model is 9.363. The model's residuals appear to be well-distributed and lack considerable autocorrelation, as indicated by the respectable error metrics on the training set and the very low autocorrelation of the first lag of residuals (ACF1) (see table 3).

The ARIMA(3,1,0) model shows a good fit, as evidenced by the plot (figure: 8) and further explained by (table: 3), The standardized residuals appear to be free of any discernible patterns, and the ACF of the residuals largely remains within the confidence bounds, indicating that the residuals are well-behaved and the model adequately captures the data's structure. Moreover, the model strikes an effective balance between goodness of fit and simplicity, as reflected by its relatively low AIC of 511.994, BIC of 522.3745, and HQC of 513.1571.

Table 4. ARIMA(5,1,4) Model Output

Call:

```
arima(x = data_set, order = arima_514_
```

Coefficients:

ar1 = 0.4088	s.e. = 0.0847
ar2 = −0.4828	s.e. = 0.0969
ar3 = 0.0475	s.e. = 0.1165
ar4 = −0.2487	s.e. = 0.1004
ar5 = 0.5967	s.e. = 0.0848
ma1 = 0.7168	s.e. = 0.0610
ma2 = 0.7826	s.e. = 0.0616
ma3 = 0.7168	s.e. = 0.0725
ma4 = 0.9999	s.e. = 0.0782

σ^2 estimated as 7.472: log likelihood = -245.57, aic = 511.14

Training set error measures:

ME	0.2249542
RMSE	2.719761
MAE	2.105426
MPE	0.2510075
MAPE	1.654741
MASE	0.4652614
ACF1	0.04183073

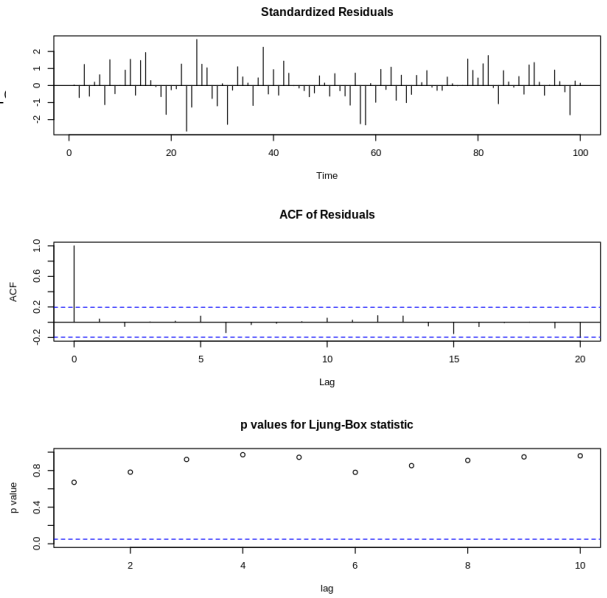


Figure 8. ARIMA(5,1,4) Dignostic

The result of the ARIMA(5,1,4) model reveals a somewhat complex model with four MA and five AR parameters that demonstrates a strong fit to the data, as indicated by the relatively modest standard errors and significant coefficients. The model's log likelihood of -245.57 and AIC of 511.14 indicate that model complexity and data fit are balanced. The training set's error metrics (ME, RMSE, MAE, MPE, MAPE, MASE, ACF1) show that the model is successful in capturing the underlying process of the data because its predictions are mostly accurate and the residuals show little autocorrelation.

Second Oeder Differencing

(Hyndman & Athanasopoulos, 2018), Sometimes the differenced data won't seem to be stationary, and in order to get a stationary series, the data may need to be differenced twice:

$$y''_t = y'_t - y'_{t-1}$$

$$= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$$

$$= y_t - 2y_{t-1} + y_{t-2}.$$

above formula is similar to

$\nabla^2 Z_t = Z_t - 2Z_{t-1} + Z_{t-2}$, from lecture 3 notes (Stationary time series)

Table 5. ARIMA(5,2,5) Model Output

Call:

```
arima(x = data_set, order = best_aic_o
```

Coefficients:

ar1 = 0.4157	ma1 = −0.2850
ar2 = −0.4783	ma2 = 0.0654
ar3 = 0.0535	ma3 = −0.0651
ar4 = −0.2449	ma4 = 0.2859
ar5 = 0.6034	ma5 = −0.9993
s.e.(ar1-5)	s.e.(ma1-5)

σ^2 estimated as 7.521: log likelihood = -243.91, aic = 509.82

Training set error measures:

$$\begin{aligned} \text{ME} &= 0.1330769 \\ \text{RMSE} &= 2.714929 \\ \text{MAE} &= 2.097323 \\ \text{MPE} &= 0.2171532 \\ \text{MAPE} &= 1.640865 \\ \text{MASE} &= 0.463471 \\ \text{ACF1} &= 0.05077024 \end{aligned}$$

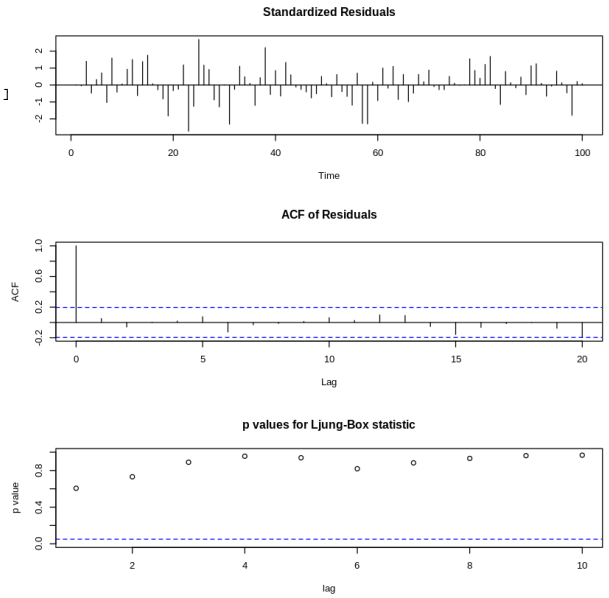


Figure 9. ARIMA(5,2,5) Diagnostic

3.3. ARIMA(5,2,5) MODEL DIAGNOSTICS TWO TIME DIFFERENCING

Table 5, With 5 **ar** and 5 **ma** components with differencing of 2, the ARIMA(5,2,5) model has a complex structure that suggests it can capture both short- and long-term dependencies in the data set. The model exhibits a low estimated variance of residuals (sigma^2 = 7.521) and the lowest AIC (509.82) among the models considered, indicating a strong fit to the data despite its complexity. Despite encouraging error metrics and a reasonably low first lag autocorrelation in the residuals (ACF1 = 0.05077024), suggesting negligible residual autocorrelation, the complexity of the model—reflected by the large number of parameters raises worries about possible overfitting.

ARIMA(5,2,5): The low AIC = 509.8191 value of this model (figure: 9) suggests that it a good fit of data; but, when compared to other models, the high BIC = 538.2538 and HQC= 518.3627 values point to a more complex model that might not be as frugal as the others.

3.4. ARIMA(2,2,0) MODEL DIAGNOSTICS two time differencing

From (table: 6), the best BIC indicates that the ARIMA(2,2,0) model balances complexity and fit, and it includes 2 **ar** terms to capture relationships up to 2 time delays. In comparison to the ARIMA(5,2,5) model, it indicates a moderate fit to the data with an estimated variance of residuals (sigma^2 = 10.13) and an AIC of 511.46. Despite being simple, the model has a good predictive performance with minimal residual autocorrelation, as evidenced by its error metrics and a slight negative first lag autocorrelation in the residuals (ACF1 = -0.0235521).This demonstrates the model's potential for forecasting success.

Table 6. ARIMA(2,2,0) Model Output

Call:

```
arima(x = data_set, order =
best_bic_order)
```

Coefficients:

$$\text{ar1} = 0.2579 \qquad \text{s.e.} = 0.0915$$
$$\text{ar2} = -0.4407 \qquad \text{s.e.} = 0.0906$$

σ^2 estimated as 10.13: log likelihood = -252.73, aic = 511.46

Training set error measures:

$$\text{ME} = 0.02797758$$
$$\text{RMSE} = 3.150308$$
$$\text{MAE} = 2.511921$$
$$\text{MPE} = 0.206235$$
$$\text{MAPE} = 1.994727$$
$$\text{MASE} = 0.5550897$$
$$\text{ACF1} = -0.0235521$$

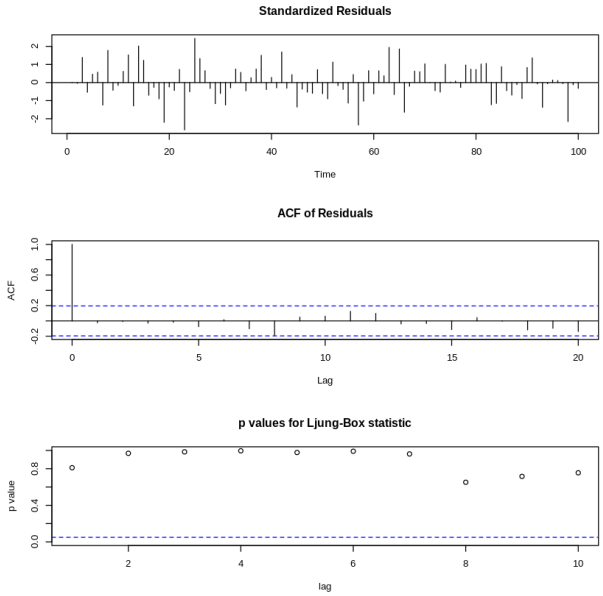


Figure 10. ARIMA(2,2,0) Diagnostic

ARIMA(2,2,0): The model's plot (figure: 10) showed well-behaved residuals with low autocorrelation, which is in line with the low AIC = 511.4645 and BIC =519.2194, HQC = 511.5733 of the model, suggesting a decent fit with a reasonably straightforward model structure.

3.5. AIC,BIC and HQC Fitted Values and forecasted values analysis

Table 7. AIC, BIC, and HQC of the fitted models

Model	ARIMA	AIC	BIC	HQC
1	(3,1,0)	511.994	522.3745	513.1571
2	(1,1,1)	514.2995	522.0848	514.4082
3	(5,1,4)	511.1394	537.0906	518.6286
4	(2,2,0)	511.4645	519.2194	511.5733
5	(5,2,5)	509.8191	538.2538	518.3627

The AIC, BIC, and HQC values for each of the five ARIMA models (table: 7) fitted to the wwwusage time series data reveal varied preferences depending on the criterion used. The ARIMA(5,2,5) model demonstrates the best fit with the lowest AIC value among all models, suggesting its superiority in capturing the data's nuances despite its complexity. However, when considering the BIC, which penalizes model complexity more heavily, the ARIMA(2,2,0) model emerges as the preferred choice, offering a more balanced approach between simplicity and fit. Interestingly, the ARIMA(5,1,4) model presents a competitive AIC value close to that of ARIMA(5,2,5), indicating a similarly good fit, but its significantly higher BIC value reflects its complexity, potentially making it less favorable under criteria that penalize for additional parameters. This model, despite its detailed capturing of the series' dynamics as evidenced by a favorable AIC, may not be the optimal choice when considering the parsimony principle as emphasized by its BIC and HQC values.

3.6. Plot all above fitted models

Table (7) presents the results of all fitted models; the plots below will now visualize these findings

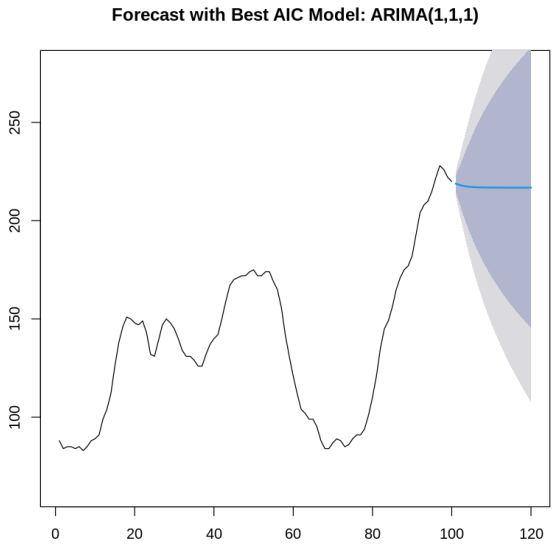


Figure 11. Forecast fitted for ARIMA(1,1,1)

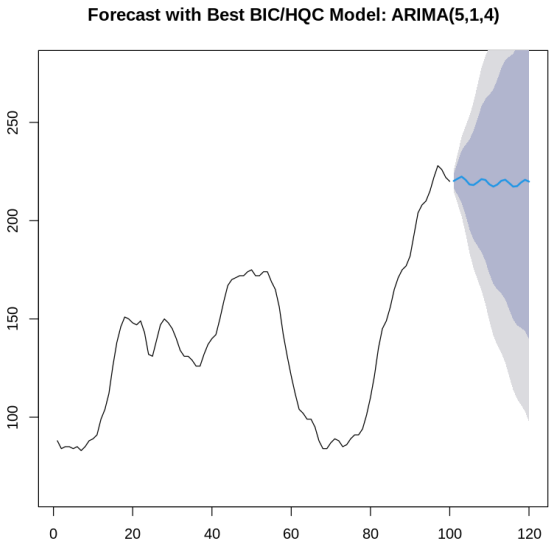


Figure 13. Forecast fitted for ARIMA(5,1,4)

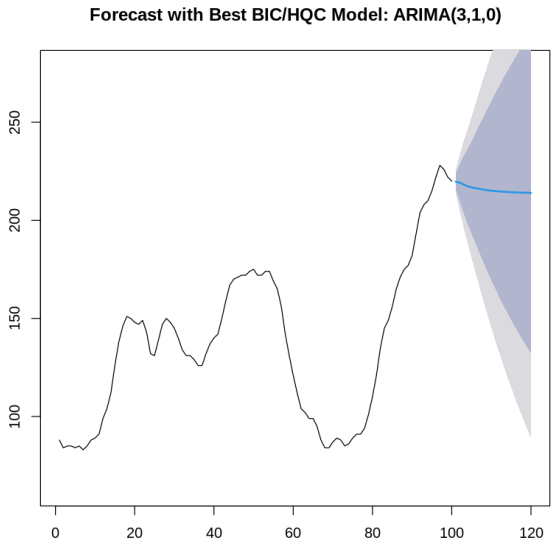


Figure 12. Forecast fitted for ARIMA(3,1,0)

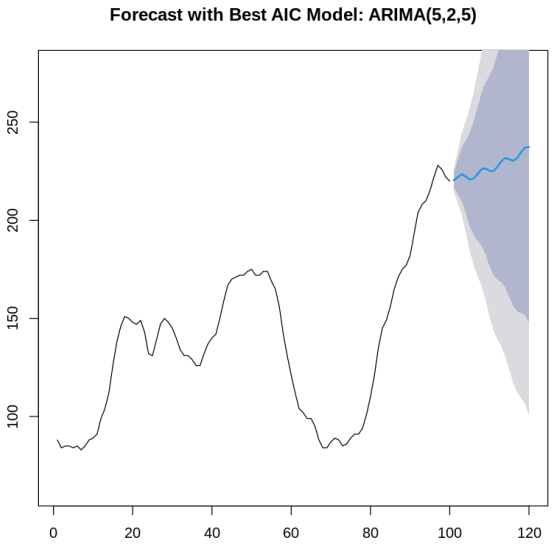


Figure 14. Forecast fitted for ARIMA(5,2,5)

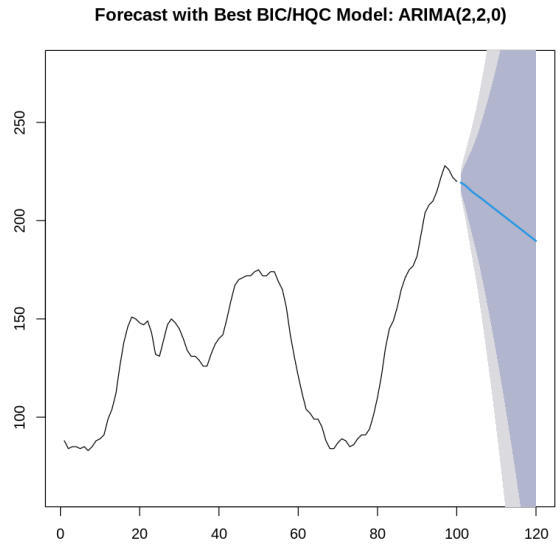


Figure 15. Forecast fitted for ARIMA(2,2,0)

Table 8. Accuracy taken for various ARIMA models.

Model	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
ARIMA(5,2,5) Training							
Train	0.1331	2.7149	2.0973	0.2172	1.6409	0.4635	0.0508
Test	-8.3453	12.5214	9.4015	-4.1107	4.5757	2.0776	0.6442
ARIMA(2,2,0) Training							
Train	0.0280	3.1503	2.5119	0.2062	1.9947	0.5551	-0.0236
Test	2.3926	14.7508	13.1187	0.7787	6.1556	2.8990	0.6809
ARIMA(1,1,1) Training							
Train	0.3036	3.1138	2.4053	0.2806	1.9175	0.5315	-0.0172
Test	-2.5856	11.3514	9.2651	-1.4666	4.4373	2.0474	0.6411
ARIMA(3,1,0) Training							
Train	0.2306	3.0446	2.3672	0.2748	1.8905	0.5231	-0.0031
Test	-2.1689	12.0719	10.0869	-1.2911	4.8168	2.2290	0.6598
ARIMA(5,1,4) Training							
Train	0.2250	2.7198	2.1054	0.2510	1.6547	0.4653	0.0418
Test	-5.3271	12.2213	9.3202	-2.7428	4.5148	2.0596	0.6695

The forecast plots for various ARIMA models, assessed according to AIC, BIC, and HQC criteria, showcase the projected internet user connection trends to a server (figures (11, 12, 14, 15,, 13) plot. The ARIMA(1,1,1) (figure: 11) and ARIMA(2,2,0) (figure: 15) models project a stable trend, while the ARIMA(5,2,5) (figure: 15) and ARIMA(3,1,0) (figure: 12) anticipate a modest increase, each within their confidence intervals. The ARIMA(5,1,4) forecast, although not depicted in these forecast plots, also demonstrates a close fit during the training phase, but exhibits larger forecast errors in the test phase, suggesting its susceptibility to overfitting—similar to the ARIMA(5,2,5) model. Despite their complexity, both the ARIMA(5,2,5) and ARIMA(5,1,4) models provide valuable insights into the future trend, aligning well with the observed data within the forecast horizon and falling within the predicted confidence ranges. These models underscore the importance of carefully balancing model complexity against predictive performance to avoid overfitting, especially when extrapolating beyond the range of observed data.

3.7. Model Accuracy's Report

The (table: 8) represents an evaluation of five ARIMA models on both training and test datasets for the "wwwusage" time series. The ARIMA(5,2,5) model stands out for having the smallest errors in the training set, yet it incurs the largest Mean Error on the test set, implying a likelihood of overfitting. Similarly, the ARIMA(5,1,4) model exhibits a low Mean Error in the training set, suggesting a good in-sample fit, but displays a substantial increase in Mean Error when applied to the test set, further reinforcing the overfitting trend seen in complex models. ARIMA(2,2,0) and ARIMA(1,1,1) show a more balanced performance across training and test sets, but with a noticeable uptick in errors on the test set, pointing to better generalization but not necessarily optimal performance. The ARIMA(3,1,0) model achieves comparable errors in training and testing, which might indicate a consistent but not necessarily superior model performance due to elevated errors on the test set. The near-zero ACF1 values across all models suggest minimal residual autocorrelation. In contrast to ARIMA(5,2,5), the ARIMA(5,1,4) does not achieve the lowest Theil's U statistic, which, combined with its test set performance, could imply it is less predictive relative to simpler models and a naive model, despite its apparent good fit to the training data.

3.8. Residual Results Analysis for each Model

Ljung-Box Test, figure (16)

Data: Residuals from ARIMA(1,1,1)
$Q^* = 7.8338$, df $= 8$, p-value $= 0.4499$

Model df: 2. Total lags used: 10

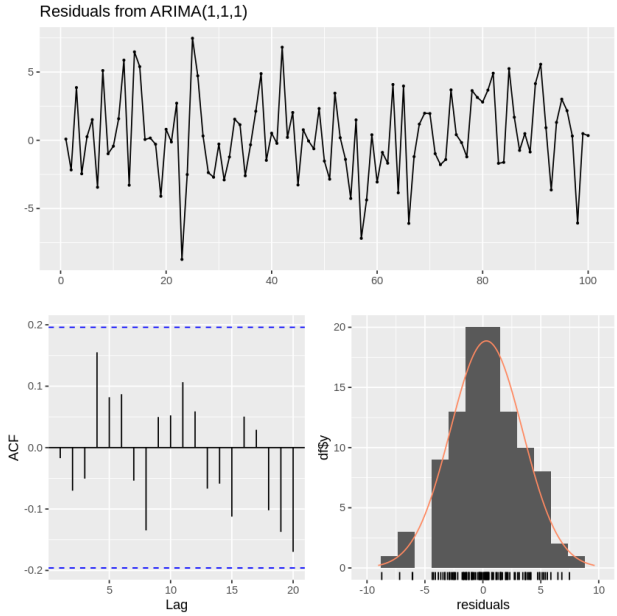


Figure 16. Residual ARIMA(1,1,1)

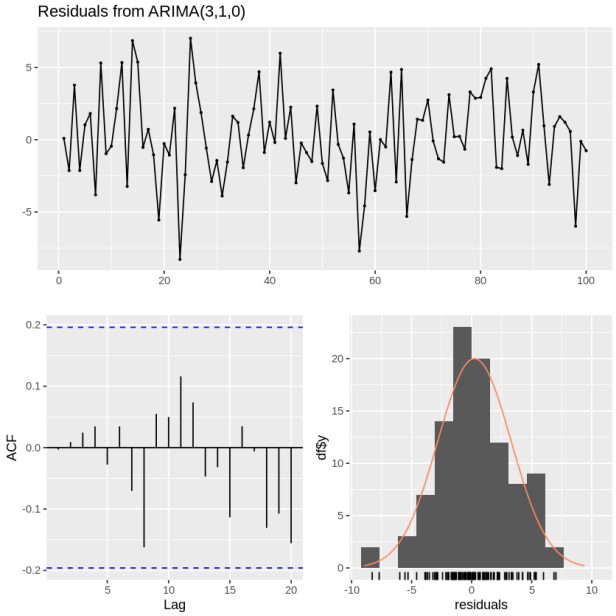


Figure 17. Residual ARIMA(3,1,0)

In (figure 16), the diagnostic plots for the ARIMA(1,1,1) model indicate a strong fit; the residuals appear as white noise, devoid of any discernible patterns or systematic structure. The Ljung-Box test, with a high p-value of 0.4499 for lags up to 10 and Q* = 7.8338, suggests the lack of significant autocorrelation in the residuals, as corroborated by the ACF plot. Moreover, the distribution of residuals closely approximates a normal distribution, further validating the model's assumptions.

Ljung-Box test, figure (17)

Data: Residuals from ARIMA(3,1,0)
$Q^* = 4.4913$, df $= 7$, p-value $= 0.7218$

Model df: 3. Total lags used: 10

In (figure 17),The residuals for an ARIMA(3,1,0) model are randomly distributed around zero, with no discernible trend or seasonality, according to the diagnostic plots. The autocorrelations are within the confidence bounds, as indicated by the ACF plot, indicating a well-fitting model. The Ljung-Box test findings validate the validity of the model by confirming that there is no substantial autocorrelation in the residuals, with a high p-value of 0.7218 and total lags of 10, Q* = 4.4913.

Ljung-Box Test Results for ARIMA(5,1,4)

The Ljung-Box test on the residuals from the ARIMA(5,1,4) model yields $Q^* = 4.6369$ with degrees of freedom $df = 3$ and a p-value of 0.2004. The model uses 9 degrees of freedom, with 12 total lags considered.

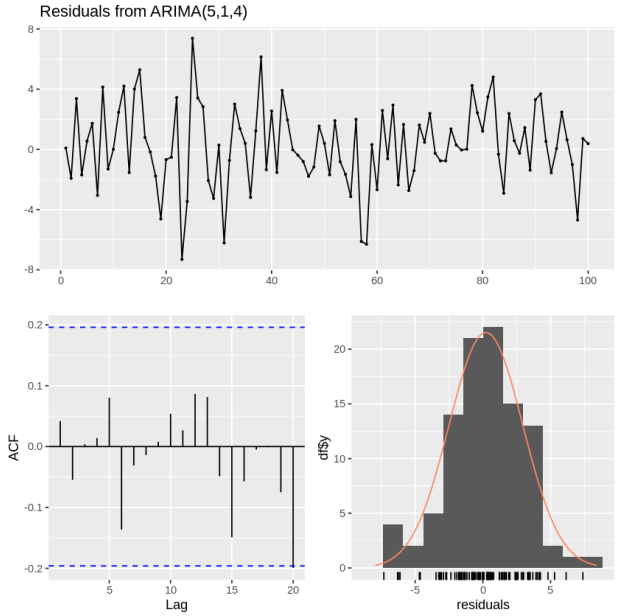


Figure 18. Residual ARIMA(5,1,4)

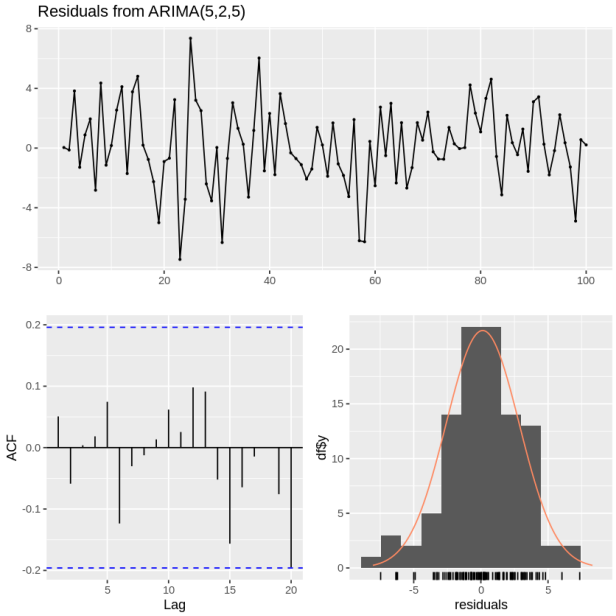


Figure 19. Residual ARIMA(5,2,5)

The diagnostic plots for the ARIMA(5,1,4) model, as presented in (figure 18), indicate that the residuals are randomly distributed over time, suggesting a good fit with no apparent missed patterns. The lack of significant autocorrelation in the ACF plot implies that the residuals behave like white noise, a finding corroborated by the Ljung-Box test, which detects no significant autocorrelation up to 12 lags (p-value of 0.2004). Given these observations, the model appears to be well-specified, meeting key assumptions for residual analysis.

Ljung-Box Test, figure (19)

Data: Residuals from ARIMA(5,2,5)
$Q^* = 5.6546$, df $= 3$, p-value $= 0.1297$

Model df: 10. Total lags used: 13

The diagnostics for the ARIMA(5,2,5) model, as shown in (figure 19), feature a histogram that closely approximates a normal distribution, residuals that behave randomly over time, and ACF values lying within the confidence bounds, indicating a lack of autocorrelation. Furthermore, the Ljung-Box test, with a p-value of 0.1297, confirms the lack of significant autocorrelation in the residuals. These diagnostics collectively suggest that the ARIMA(5,2,5) model adequately fits the data, adhering to critical assumptions about residual analysis

Ljung-Box Test, figure (20)

Data: Residuals from ARIMA(2,2,0)
$Q^* = 6.6784$, df $= 8$, p-value $= 0.5717$

Model df: 2. Total lags used: 10

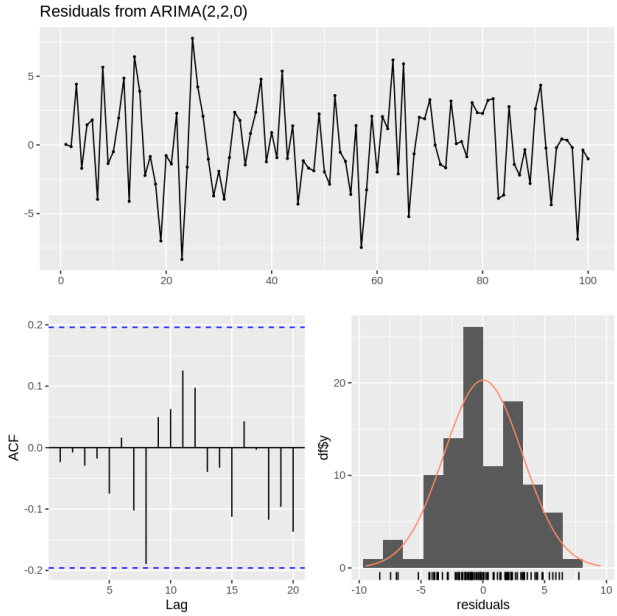


Figure 20. Residual ARIMA(2,2,0)

In (figure 20), the ARIMA(2,2,0) model's residuals randomly oscillate around zero, demonstrating how well the model fits the data. Since all of the spikes in the ACF plot are contained within the confidence intervals, there is no discernible autocorrelation. The appropriateness of the model is confirmed by the Ljung-Box test, which yielded a p-value of 0.5717,Q* = 6.6784, indicating the absence of significant autocorrelation in the residuals.

List of Abbreviations Used

ARIMA:	Autoregressive Integrated Moving Average
ACF:	Autocorrelation Function
PACF:	Partial Autocorrelation Function
AIC:	Akaike Information Criterion
BIC:	Bayesian Information Criterion
HQC:	Hannan-Quinn Information Criterion
MA:	Moving Average
AR:	Autoregressive

4. Conclusion

Model Complexity vs. Fit: The model that best fits the data, ARIMA(5,2,5), has the lowest AIC value and captures both short- and long-term dependencies. Its intricacy, as evidenced by high HQC and BIC values, prompts worries about overfitting, which would reduce its ability to predict outcomes from unobserved data.

Balancing Complexity and Fit: A more balanced choice is the ARIMA(2,2,0) model, which has a lower BIC value and a moderate AIC value, indicating less complexity and a moderate fit to the data. It is proposed that this model has minimum residual autocorrelation and strong predictive ability, which may make it more appropriate for forecasting. **Overfitting Considerations:** AIC, BIC, and HQC values indicate that while complex models such as ARIMA(5,2,5) and ARIMA(5,1,4) exhibit good in-sample fit, they are less suitable for predicting future trends than simpler models due to their complexity and associated risk of overfitting.

In summary, the simpler ARIMA(2,2,0) model may be more useful for predicting than the more complex ARIMA(5,2,5) model, despite the latter's high fit to the historical data and tendency toward overfitting. In order to improve forecast accuracy and dependability, this analysis emphasizes how crucial it is to strike a balance between model fit and complexity.

References

Hyndman, R. J. and Athanasopoulos, G. Forecasting: Principles and practice. Monash University, Australia, 2018. URL https://otexts.com/fpp2/stationarity.html. Available from: https://otexts.com/fpp2/stationarity.html.

5. Appendix

Codes used to develop above assignment presented in this report appendix and available to this colab link: https://shorturl.at/hIV14

```
# Install Libraries used
install.packages("tseries")
install.packages("forecast")
# DATA LOADING AND FIND BEST PARAMETERS
library(forecast)
# Load your time series data
data_set <- scan("/content/wwwusage.txt",
    skip = 1)
# Load the dataset of WWW usage (number of
    users connected to the Internet)
cat("Loaded Dataset\n")
cat(replicate(13, "-----"))
cat('\n')
print(data_set)

# Basic statistical exploration
minimum_value <- min(data_set)
maximum_value <- max(data_set)
average_value <- mean(data_set)

# Plotting the initial time series
plot(data_set, type = "o", main = "WWW
    Usage Over Time", xlab = "Time (Minutes
    )", ylab = "Number of Users", ylim = c(
    minimum_value - 10, maximum_value + 10)
    , col = "blue")
# Adding a grid
abline(h = seq(from = floor(minimum_value),
     to = ceiling(maximum_value), by = 10),
     col = "lightgray", lty = "dotted")
abline(v = seq(from = 1, to = length(data_
    set), by = 5), col = "lightgray", lty =
     "dotted")
```

Table 9. R code for computing Assignment.

```
# Plot ACF && PACF
# Autocorrelation and Partial
    Autocorrelation Plots
par(mfrow=c(2,1)) # Set up a 2x1 grid for
    plots
acf(data_set, main = "ACF of WWW Usage Data
    ")
grid()
pacf(data_set, main = "PACF of WWW Usage
    Data")
grid()

# Differencing the series for stationarity
par(mfrow=c(2,1)) # Set up a 2x1 grid for
    plots
differenced_data <- diff(data_set)
# Plot differenced data
plot(differenced_data, type = "o", col = "
    blue",pch =19,
      xlab = "Time", ylab = "Differenced
          Data",
      main = "Differenced Data Plot")
grid()
# Plot ACF and PACF after one time
    Differencing
par(mfrow=c(2,1)) # Set up a 2x1 grid for
    plots
# Autocorrelation and Partial
    Autocorrelation of Differenced Data
acf(differenced_data, main = "ACF after
    Differencing")
grid()
pacf(differenced_data, main = "PACF after
    Differencing")
grid()
# best fit ar model for Simple model d=1
differenced_data_1 = diff(data_set,
    differences = 1)
ar.yw(differenced_data_1, max=9)
# best fit ar model for Simple model d=2
differenced_data_2 = diff(data_set,
    differences = 2)
ar.yw(differenced_data_2, max=9)
```

Table 10. R code for computing Assignment.

[1]Nanyang Technological University (NTU), Singapore [2]School of Computer Science and Engineering (SCSE) [3]Master of Science in Artificial Intelligence. Correspondence to: Ntambara Etienne <ntam0001@e.ntu.edu.sg>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

```r
# iteratively fits ARIMA models across a
    range of specified orders
# for the autoregressive (p), differencing
    (d), and moving average (q) parameters
n <- length(data_set)

# Initialize lists to store models and
    their evaluations
models_list <- list()
aic_values <- setNames(numeric(), character
    ())
bic_values <- setNames(numeric(), character
    ())
hqc_values <- setNames(numeric(), character
    ())

# Define ranges for p, d, q
p_range <- 0:9
d_range <- 0:3
q_range <- 0:9

# Fit models
for (p in p_range) {
    for (d in d_range) {
        for (q in q_range) {
            model_id <- paste("ARIMA(", p,
                ",", d, ",", q, ")", sep =
                "")
            model <- suppressWarnings(
                tryCatch({
                arima(data_set, order = c(p
                    , d, q))
            }, error=function(e) NULL))
            if (!is.null(model)) {
                models_list[[model_id]] <-
                    model
                aic_values[model_id] <- AIC
                    (model)
                bic_values[model_id] <- BIC
                    (model)
                hqc_values[model_id] <- -2
                    * logLik(model) + 2 *
                    length(coef(model)) *
                    log(log(n))
            }
        }
    }
}

# Function to print best models for
    differencing 1 and 2
print_best_models <- function(d) {
    d_models <- names(aic_values)[grepl(
        sprintf(",%d,", d), names(aic_
        values))]
    if (length(d_models) == 0) {
        cat(sprintf("No models fitted for
            differencing %d\n\n", d))
        return()
    }

    aic_best <- d_models[which.min(aic_
        values[d_models])]
    bic_best <- d_models[which.min(bic_
        values[d_models])]
    hqc_best <- d_models[which.min(hqc_
        values[d_models])]

# Extract the model orders from the model
    ids
    aic_order <- strsplit(gsub("ARIMA
        \\(|\\)", "", aic_best), ",")[[1]]
    bic_order <- strsplit(gsub("ARIMA
        \\(|\\)", "", bic_best), ",")[[1]]
    hqc_order <- strsplit(gsub("ARIMA
        \\(|\\)", "", hqc_best), ",")[[1]]

    cat("For differencing", d, ":\n")
    cat("AIC Best Model: ARIMA(", paste(aic
        _order, collapse = ","), ")\n")
    cat("BIC Best Model: ARIMA(", paste(bic
        _order, collapse = ","), ")\n")
    cat("HQC Best Model: ARIMA(", paste(hqc
        _order, collapse = ","), ")\n\n")
}

# Print best models for d=1 and d=2
cat("Size:", n, "\n")
print_best_models(1)
print_best_models(2)
# BEST MODELS FOUND ABOVE ITERATIONS
# Sample size
n <- length(data_set)
# Define the models to fit found from above
    section of codes
models_to_fit <- list(
  c(1, 1, 1),
  c(3, 1, 0),
  c(5, 1, 4),
  c(5, 2, 5),
  c(2, 2, 0)
)

# Fit the models and store their AIC, BIC,
    and HQC values
for (model_params in models_to_fit) {
  p <- model_params[1]
  d <- model_params[2]
  q <- model_params[3]

  model <- suppressWarnings(tryCatch({
    arima(data_set, order = c(p, d, q))
  }, error=function(e) NULL))

  if (!is.null(model)) {
    model_id <- paste(p, d, q, sep = "-")
    models_list[[model_id]] <- model
    aic_values <- c(aic_values, AIC(model))
    bic_values <- c(bic_values, BIC(model))
    # Calculate HQC
    k <- length(coef(model))
    hqc <- -2 * logLik(model) + 2 * k * log
        (log(n))
    hqc_values <- c(hqc_values, hqc)
  }
}
```

Table 11. R code for computing Assignment.

Table 12. R code for computing Assignment.

```
# Print the AIC, BIC, and HQC values for
    the specified models
for (i in 1:length(models_to_fit)) {
  model_id <- paste(models_to_fit[[i]],
      collapse = "-")
  cat("Model(", model_id, ")\n")
  cat("AIC:", aic_values[which(names(models
      _list) == model_id)], "\n")
  cat("BIC:", bic_values[which(names(models
      _list) == model_id)], "\n")
  cat("HQC:", hqc_values[which(names(models
      _list) == model_id)], "\n\n")
}
# Best Model Generated by complex Algorithm
arima_310_order <-  c(3, 1, 0)
arima_111_order <-  c(1, 1, 1)
arima_514_order <-  c(5, 1, 4)

best_aic_order <- c(5, 2, 5)  # For AIC
best_bic_order <- c(2, 2, 0)  # For BIC
best_hqc_order <- c(2, 2, 0)  # For HQC,
    which is same as BIC in your results

# Model fitting with ARIMA based on prior
    analysis
# our best AIC model is ARIMA(5,2,5) and
    the best BIC/HQC model is ARIMA(2,2,0)
    ARIMA(1,1,1), ARIMA(3,1,0), ARIMA
    (5,1,4)
# Fitting these models
arima_111_model <- arima(data_set, order =
    arima_111_order)
arima_310_model <- arima(data_set, order =
    arima_310_order)
arima_514_model <- arima(data_set, order =
    arima_514_order)
best_aic_model <- arima(data_set, order =
    best_aic_order)
best_bic_hqc_model <- arima(data_set, order
    = best_bic_order)
# This is also the best model by HQC as per
    your results
# Model diagnostics
# Display results for ARIMA(1,1,1)
cat("ARIMA(1,1,1) Model Output\n")
print(summary(arima_111_model))
cat("\n")
tsdiag(arima_111_model)
```

Table 13. R code for computing Assignment.

```
# Display results for ARIMA(3,1,0)
cat("ARIMA(3,1,0) Model Output\n")
print(summary(arima_310_model))
cat("\n")
tsdiag(arima_310_model)
# Model diagnostics
# Display results for ARIMA(5,1,4)
cat("ARIMA(5,1,4) Model Output\n")
print(summary(arima_514_model))
cat("\n")
tsdiag(arima_514_model)
# Display results for Best AIC Model: ARIMA
    (5,2,5)
cat("ARIMA(5,2,5) Output\n")
print(summary(best_aic_model))
cat("\n")
# Model diagnostics
tsdiag(best_aic_model)

# Display results for Best BIC/HQC Model:
    ARIMA(2,2,0)
cat("Model: ARIMA(2,2,0) Output\n")
print(summary(best_bic_hqc_model))
cat("\n")
tsdiag(best_bic_hqc_model)

# Forecasting and plotting future values
forecast_aic <- forecast(best_aic_model, h
    = 20)
forecast_bic_hqc <- forecast(best_bic_hqc_
    model, h = 20)
forecast_arima_111 <- forecast(arima_111_
    model, h = 20)
forecast_arima_310 <- forecast(arima_310_
    model, h = 20)
forecast_arima_310 <- forecast(arima_514_
    model, h = 20)

plot(forecast_arima_111, main = "Forecast
    with Best AIC Model: ARIMA(1,1,1)",
    ylim = c(min(data_set) - 20, max(data_
    set) + 50))
plot(forecast_arima_310, main = "Forecast
    with Best BIC/HQC Model: ARIMA(3,1,0)",
     ylim = c(min(data_set) - 20, max(data_
    set) + 50))
plot(forecast_arima_310, main = "Forecast
    with Best BIC/HQC Model: ARIMA(5,1,4)",
     ylim = c(min(data_set) - 20, max(data_
    set) + 50))
plot(forecast_aic, main = "Forecast with
    Best AIC Model: ARIMA(5,2,5)", ylim = c
    (min(data_set) - 20, max(data_set) +
    50))
plot(forecast_bic_hqc, main = "Forecast
    with Best BIC/HQC Model: ARIMA(2,2,0)",
     ylim = c(min(data_set) - 20, max(data_
    set) + 50))
```

Table 14. R code for computing Assignment.

```
# Model accuracy assessment on a test set
test_set <- window(data_set, start = length
    (data_set) - 9)

# Forecast using ARIMA(5,2,5)
forecast_aic <- forecast(models_list[["
    5-2-5"]], h = 10)
# Add a title
cat("\nAccuracy Assessment for Model ARIMA
    (5,2,5):\n")
# Assess accuracy
round(accuracy(forecast_aic, test_set),4)

# Forecast using ARIMA(2,2,0)
forecast_bic_hqc <- forecast(models_list[["
    2-2-0"]], h = 10)
# Add a title
cat("\nAccuracy Assessment for Model ARIMA
    (2,2,0):\n")
# Assess accuracy
round(accuracy(forecast_bic_hqc, test_set)
    ,4)

# Forecast using ARIMA(1,1,1)
forecast_arima_111 <- forecast(models_list
    [["1-1-1"]], h = 10)
# Add a title
cat("\nAccuracy Assessment for Model ARIMA
    (1,1,1):\n")
# Assess accuracy
round(accuracy(forecast_arima_111, test_set
    ),4)

# Forecast using ARIMA(3,1,0)
forecast_arima_310 <- forecast(models_list
    [["3-1-0"]], h = 10)
# Add a title
cat("\nAccuracy Assessment for Model ARIMA
    (3,1,0):\n")
# Assess accuracy
round(accuracy(forecast_arima_310, test_set
    ),4)

# Forecast using ARIMA(3,1,0)
forecast_arima_514 <- forecast(models_list
    [["5-1-4"]], h = 10)
# Add a title
cat("\nAccuracy Assessment for Model ARIMA
    (5,1,4):\n")
# Assess accuracy
round(accuracy(forecast_arima_514, test_set
    ),4)
# check Residuals
checkresiduals(arima_111_model)
checkresiduals(arima_310_model)
checkresiduals(arima_514_model)

checkresiduals(best_aic_model) #ARIMA
    (5,2,5)
checkresiduals(best_bic_hqc_model) #ARIMA
    (2,2,0)
```

Table 15. R code for computing Assignment.