MATH5306 Assessment

201605543

December 17, 2025

Contents

1 Question 1

We are given a set of hard drive lifetimes to analyse. If we want to predict their lifetimes, the easiest method is to fit a common survival model like the exponential distribution. If we give R the data, we can very quickly visualise it using a histogram and box-plot. See Appendix B for the commands. We can visually see that the battery lifetimes follow a logarithmic-like distribution (Figure 1), and could easily be fitted by a standard survival analysis curve like the exponential distribution. We can also see a high length of outliers however, and this could pull the curve away from the bulkier half of the curve that die within 5 years. Observing the box-plot closer, we can see the median is quite low within the interquartile range, implying further that summary statistics are being dragged up by values above the mean ranging far higher above the mean than the values below the median range below it.

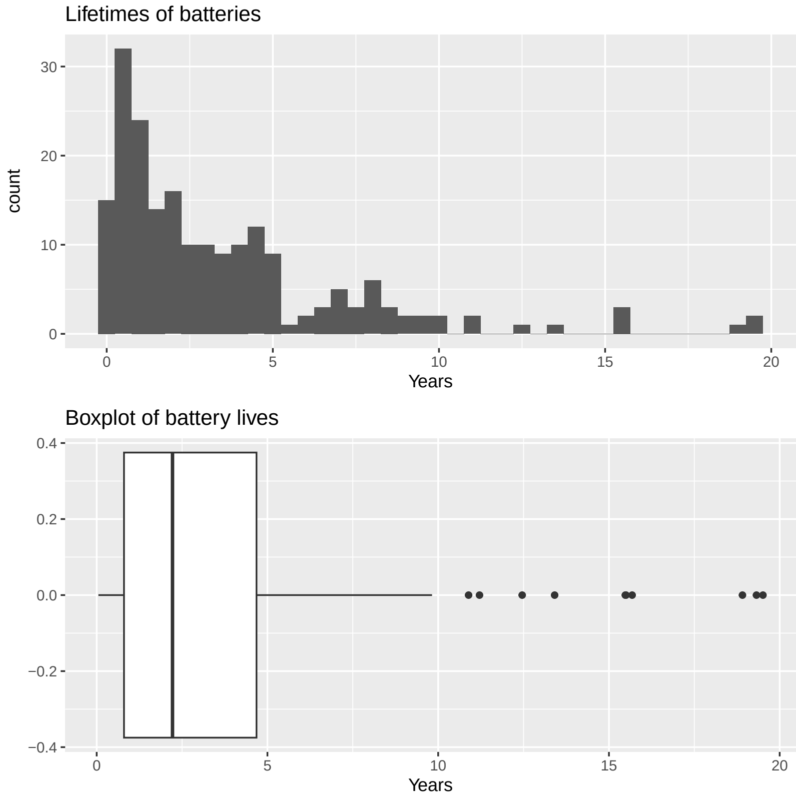


Figure 1: Histogram of battery lifetimes

If we want to fit a continuous distribution to data, we can call the R ”fitdistr” function (Appendix C). As we can see from Figure 2, both models fit roughly the same. While the gamma distribution appears to fit the lower values more closely, the exponential distribution follows the overall lifetimes more closely. We can formally test whether datasets follow known continuous distributions using the Kolmogorov-Smirnov test, but the test is inconclusive in this case (Appendix C). Therefore, we consider the AIC and BIC values of the models to choose between them.

Table 1: Information Criteria for models

Criterion	Exponential.Model	Gamma.Model
AIC	901.364261502602	902.57314739141
BIC	904.66257886915	909.169782124506

Looking at Table 1, both information criteria have lower values for the exponential model. The exponential model also has a higher log-likelihood than the gamma

(Appendix C). Therefore, the exponential model is likely a better fit. We can compare simulated versions of the plots to the original. Observing Figure 2, we can see that both have similarly distributed lifetimes. In Figure 3 however, we see the summary statistics like the median and interquartile range could be more accurate in the exponential simulation. Overall, it seems fair to argue that the exponential is the better pick in this situation, since it has lower information criteria, higher log-likelihood, and slightly closer looking distributions to the data-generating process.

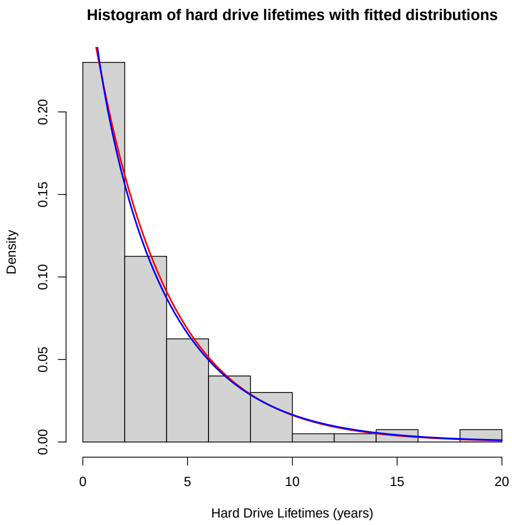


Figure 2: Fitted models to data distribution

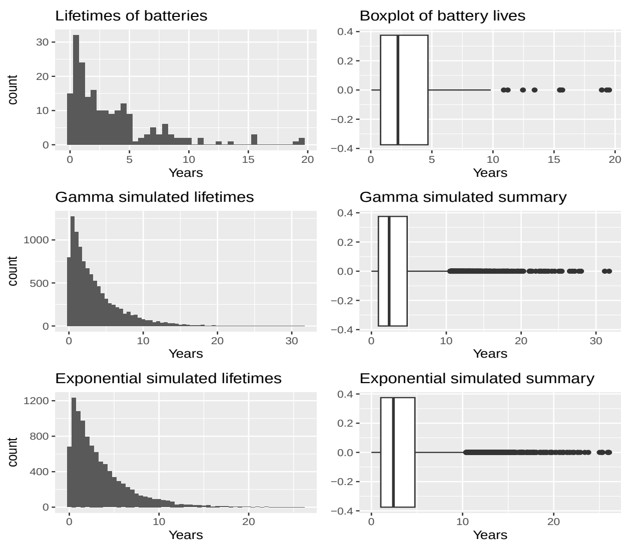


Figure 3: Simulations of fitted models

2 Question 2

We are given a set of European stock closing prices, and are asked to predict future trends. In order to fit time series models to the data, a common assumption is stationarity, or weakly a non-unit mean. Observing a plot of the dataset (Figure 4, made in Appendix D), we can see that the data visually is not stationary.. Firstly, we consider the specific series for the DAX closing prices. Observing Figure 5, we see that the ACFs for the DAX do not decrease in a geometric fashion, suggesting the series is not stationary. We can formally confirm this using an augmented Dickey-Fuller test, we have to take the first difference of the series in order to have stationarity (Appendix E). With stationary data, we can use the R forecast package to fit a time series model. We can fit an MA(p), AR(q), and ARMA(p, q) model to the dataset, but after fitting the optimal options of each, the ARIMA(0, 1, 1) model had the best AIC (Appendix E).

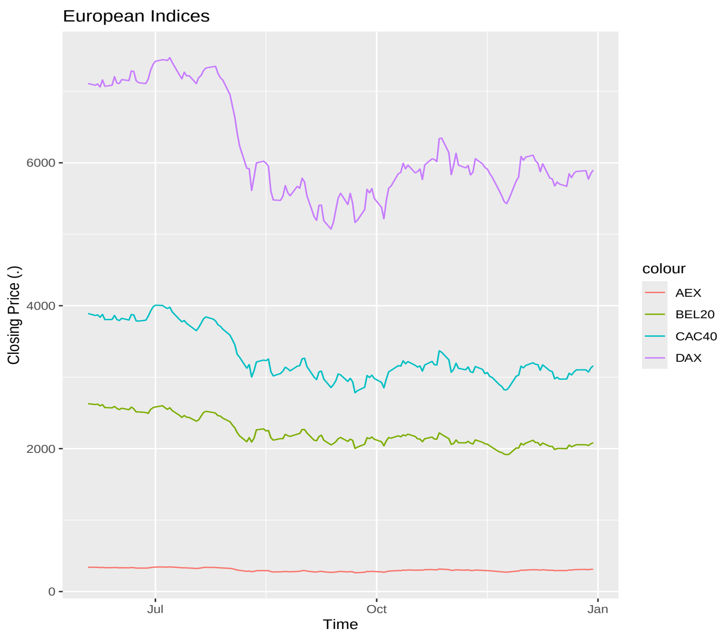


Figure 4: European Index Closing Prices

We can use this ARIMA model to predict future closing prices for the DAX, and plot them using R. Observing Figure 6, we can see that the model predicts general price stability in the future, with no visible upward or downward trends to follow. We can go beyond ARMA, and include the information from our entire dataset. For the VAR model, we first have to demonstrate stationarity and co-integration between the variables at the same integration order. This is done in Appendix E, since all the variables are stationary at first difference and co-integrated at second, we fit a VAR model with two lags. Since errors between variables can explain trends within the variables, the VAR model can provide impulse response functions for variable shifts. In Figure 7, we can see the VAR model predictions for the change in prices, which suggest a small decrease in the change in DAX closing prices before stabilisation over the next 10 days.

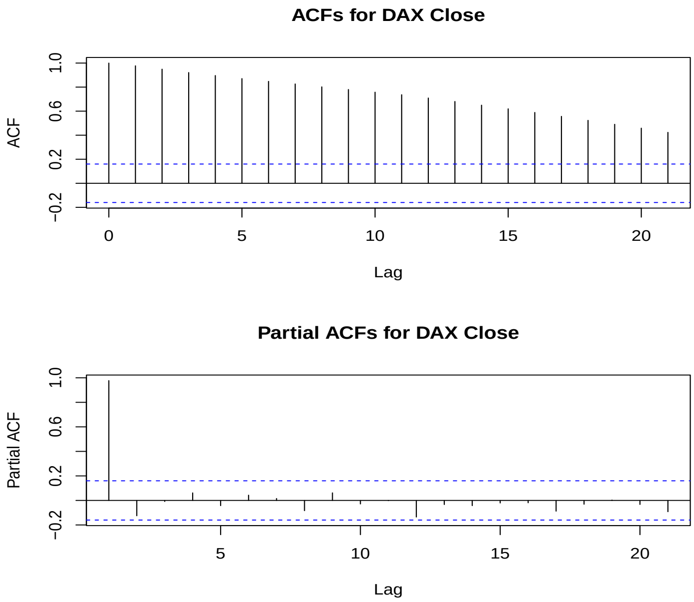


Figure 5: ACFs and partial ACFs for DAX closing price data

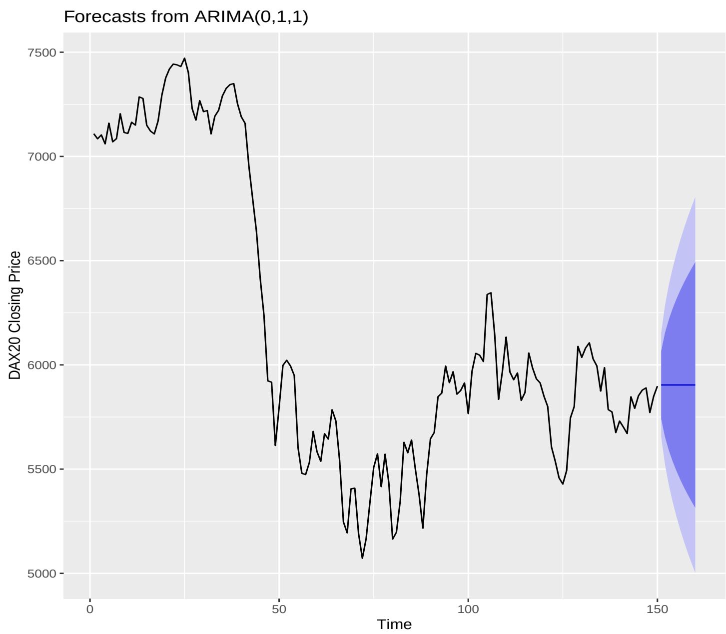


Figure 6: Predictions for DAX index from ARIMA model

Table 2: Engle-Granger test matrix (H1: cointegrated)

Indices	AEX	BEL20	CAC40	DAX
AEX		0.01	0.01	0.01
BEL20	0.01		0.01	0.01
CAC40	0.01	0.01		0.01
DAX	0.01	0.01	0.01	

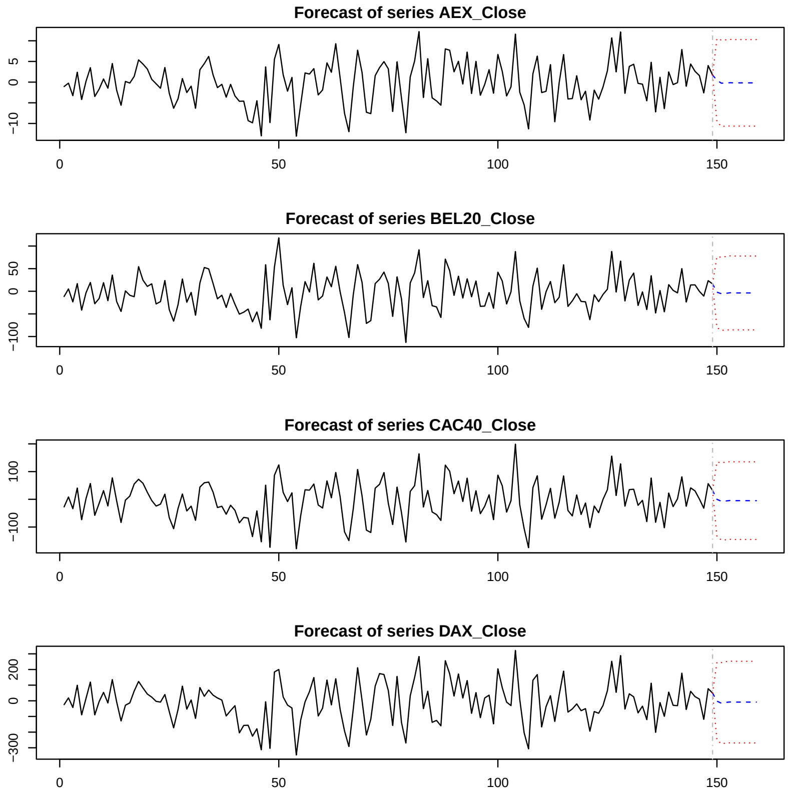


Figure 7: VAR Forecasting of dataset

After completing both models, an analysis of the residuals from the ARMA regression reveals that the residuals were not normal at all (Appendix E). This is likely reflective of non-constant variance, which ARIMA is not robust against. On the other hand, the VAR post-regression statistics showed no sign of serial correlation and had normal residuals (Appendix F), implying that the data satisfies all of the bias and efficiency assumptions typically placed on VAR modelling. Co-integration can be an incredibly powerful tool, and, in this case, offered a more robust model when all data was considered than just the one where only the DAX itself was considered.

3 Question 3

For this question, we are given a univariate time series of deaths in the United Kingdom due to COVID-19. We can make two models to predict this, one fitting an ARIMA to the raw data, and the other fitting an ARIMA model to the square rooted data. We can see from a visualisation that COVID deaths over time have two extreme spikes, constantly declining ACFs, and PACFs that do not completely drop after 2 lags (Appendix G). We can also test this formally with an augmented Dickey-Fuller test. Consequently, we have to take the first difference, which we can show are stationary (Appendix H).

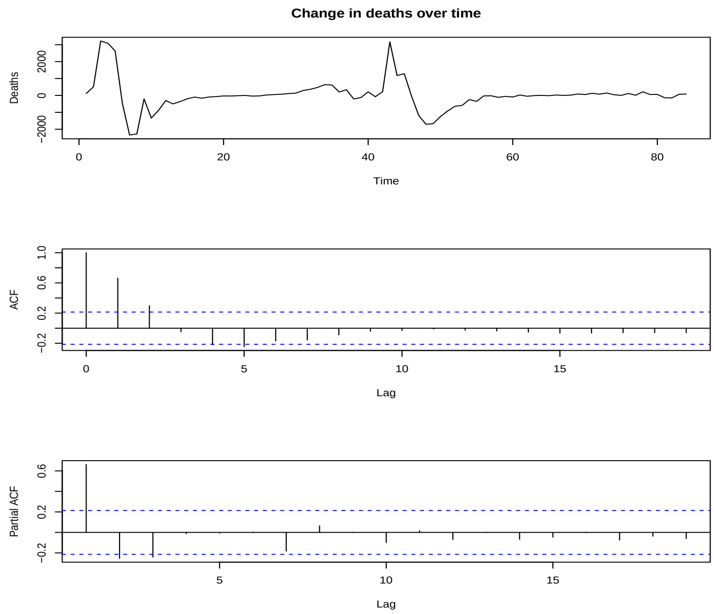


Figure 8: Change in deaths, ACF and PACFs for change in deaths

We can now fit our ARIMA models to the raw and square rooted data (Appendix G). Observing Figure 9, we can see that the raw data projections are far more conservative, and predict a shallower rise in deaths over time. On the other hand, the back-fitted projections suggest a far higher rise in deaths over the next two weeks (Figure 10). Hypothesis testing on residuals revealed no statistically significant issue with either model (Appendix H). Their residuals both followed a normal distribution based on the Shapiro-Wilk test, and neither showed evidence of serial-correlation under the Ljung-Box test. Observing Figures 11 and 12, we see that neither model has abnormal ACFs or PACFs (both rapidly decay below suggested p-values), and they do not exceed the white noise test statistic. The only potentially indicative statistic is that the back-fitted ARIMA model's residual quantiles visually fit better on a straight line than the base model. Unfortunately, the two models occupy different state spaces, and consequently information criteria are not available. Based on residual quantiles and the fact that it visually seems to fit the pre-existing trend better, the back-fitted model

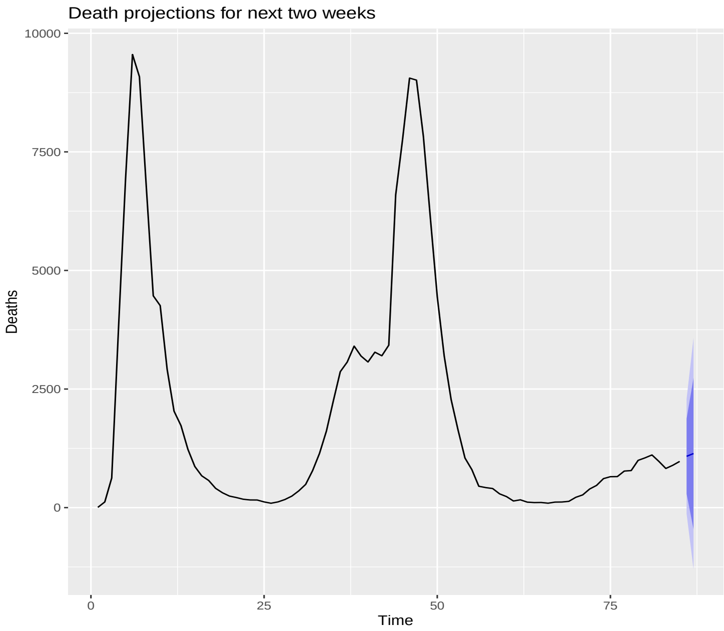


Figure 9: Base ARIMA Model death projections

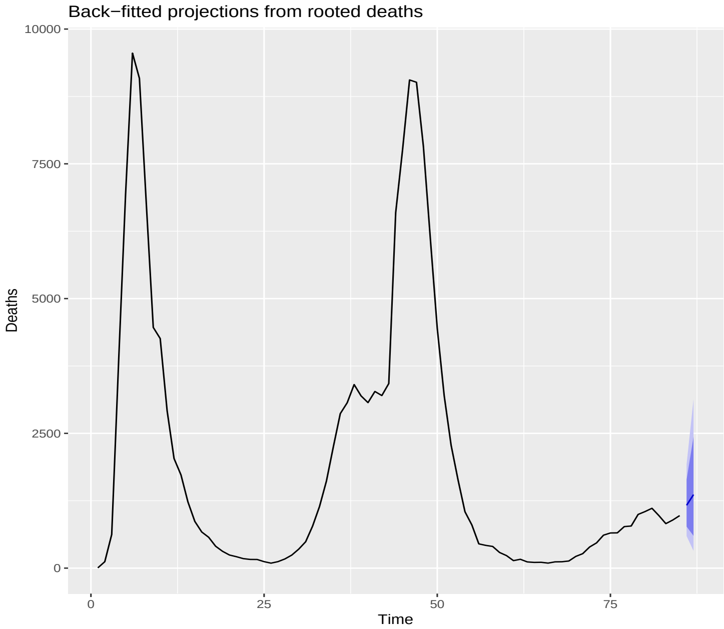


Figure 10: Back-fitted ARIMA model death projections

is likely the better model in this situation. Both predict an increase in deaths over the next 2 weeks, but the back-fitted one suggests 85 more deaths within the first week than the raw model, and 233 more deaths within the second over the raw model. The back-fitted model also follows an ARMA(2, 2) process, implying that an equal mix of previous values of the series and previous error terms of the series impact the current value, whereas the raw dataset model essentially follows an AR(3) process, suggesting previous error values are unimportant and lags predict current values far more. Preferring the back-fitted model over the raw one suggests that previous shocks and previous values matter. In the raw data model (Appendix H), the first auto-regression has a large positive coefficient, the second has a small negative auto-regression, and the third has a medium negative auto-regression, suggesting that the death auto-regressions start off in the same direction as the current value, but linearly move in an opposite direction up to the third past week.

```
Series: deaths$wk.deaths
ARIMA(3,1,0)

Coefficients:
         ar1      ar2      ar3
      0.7474  -0.0005  -0.3067
s.e.  0.1041   0.1375   0.1166

sigma^2 = 396218:  log likelihood = -659.53
AIC=1327.06   AICc=1327.57   BIC=1336.78
Series: sqdeaths$wk.deaths
ARIMA(2,1,2)

Coefficients:
         ar1      ar2      ma1     ma2
      1.1518  -0.6262  -0.4783  0.6201
s.e.  0.2000   0.1873   0.1948  0.1419

sigma^2 = 24.82:  log likelihood = -252.85
AIC=515.71   AICc=516.48   BIC=527.86
```

For the back-fitted model, the first auto-regression has a very large positive value, the second has a high negative value, the first moving average coefficient has a medium negative value, and the second moving average coefficient has a larger positive value. This suggests a similar auto-regressive pattern to the raw model, but that the error term tends to flip from period to period, with increasing volatility.

4 Question 4

We are given a univariate time series of exchange rates between the US Dollar and British Pound over 12 years. We want to understand changes in volatility before and after the credit crisis. If we want to stochastically model volatility, we can use the GARCH model. This model expects returns, so we can take the logged returns (Appendix I). Looking at Figure 13, the data visually looks stationary. We confirm this using a augmented Dickey-Fuller test in Appendix J.

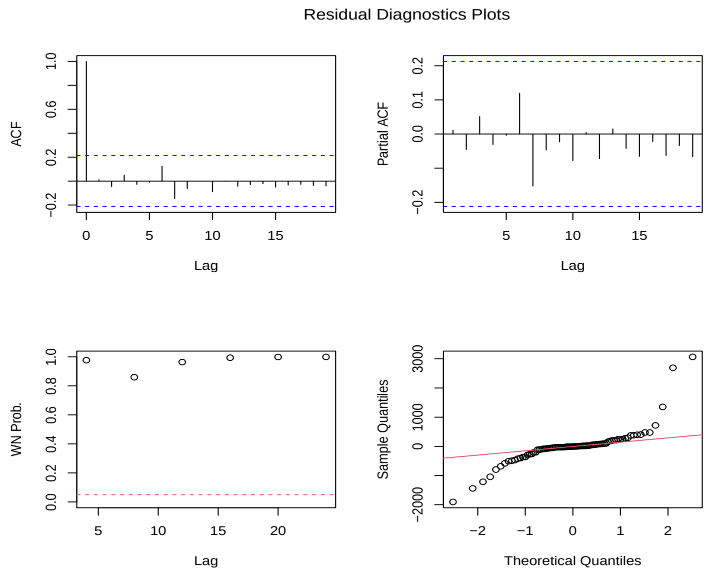


Figure 11: Visual residuals testing for the base ARIMA model

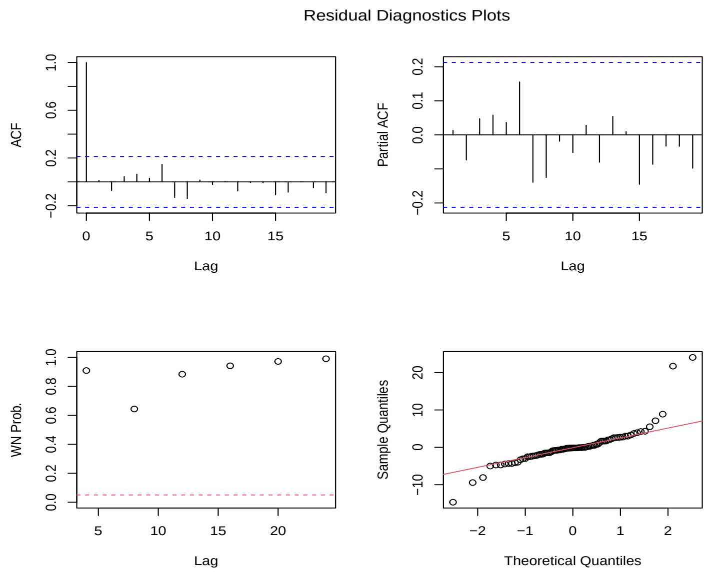


Figure 12: Visual residuals testing for the back-fitted ARIMA model

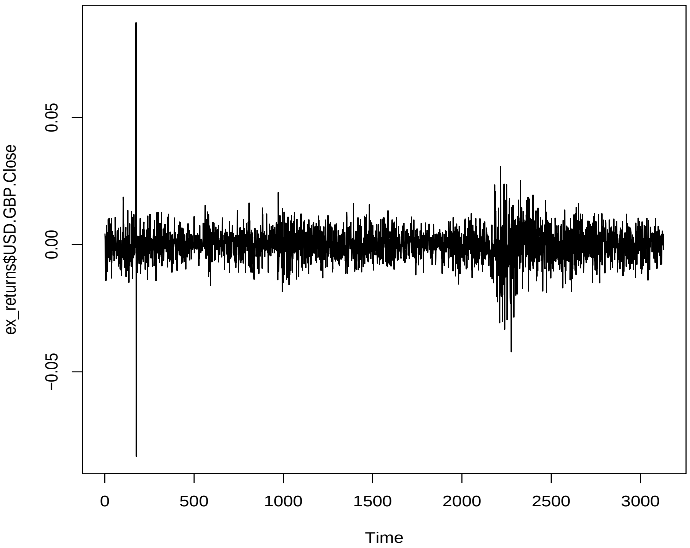


Figure 13: Logged returns for USD/GBP exchange rate

```
[1] "The pre-crash data ranges from 2000-05-02 to 2007-07-31, and"
[1] "the post-crash data ranges from 2007-08-01 to 2012-04-30."
```

For simplicity, we split the data assuming the crisis happened overnight at the start of August 2007. Firstly, direct residual testing finds no problems with the distribution of the residuals for a basic ARMA(0, 1) model on both datasets (Appendix J), but the residuals stop following a white noise process in the post-crash dataset (Appendix I). This could be due to stochastically changing variance, as ARIMA models are not designed around conditional variance. However, this does suggest that the mean can effectively be modelled by an ARMA distribution. We can now fit the GARCH models to the pre-crisis and post-crisis datasets in order to quantify changes to volatility in the time period. Looking at the models (Appendix J), we can see that they satisfy key model assumptions. GARCH models generally assume that:

1. The average value is already modelled / a white noise process
2. Stationarity, finite variance
3. Presence of ARCH prior to modelling

We have already shown that ARMA processes can model the mean of the data effectively. We have also shown that the data itself is stationary, and the α and β of the models are both far smaller than 1, implying that the variance can be modelled by GARCH. Additionally, we ran an ARCH test on the data before fitting the GARCH model, which was statistically significant at the 99.9% CI for both the pre-crisis and post-crisis datasets. The model also passes post regression statistical checks. Both models pass the Jarque-Bera and Shapiro-Wilk tests at the 95% CI, meaning that the residuals appear to be normal beyond a reasonable doubt. Neither model fails the

Ljung-Box test, so there is no evidence of serial-correlation. Neither model fails the post fit LM ARCH test, meaning that there is no evidence autoregressive conditional heteroscedasticity persists on the fitted data. Overall, the totality of evidence suggests the models satisfy assumptions, and they were chosen from a wide selection to minimise the AIC and BIC values as much as possible.

The pre-crisis GARCH model has an α of 4.64×10^{-2}, and a β of 9.50×10^{-1}, while the post-crisis GARCH model has an α of 4.422×10^{-2} and a β of 9.429×10^{-1} (Appendix J). The decrease in α implies that the speed at which volatility adjusts decreased in the exchange rate market, and the increase in β implies that the persistence of volatility increased. This could be related to higher risk-aversion in the market, where increases in volatility cause people to remain sceptical of the market for longer. Alternatively, there may have been structural alterations in the markets in response to the crisis that altered suggested or feasible trading approaches for the Forex markets. Overall, we find the GARCH model assumptions were well met, the speed at which volatility adjusts to new data decreased after the crisis, and the persistence of volatility after the crisis increased.

A Appendix

This document was compiled by lualatex (LuaHBTeX, Version 1.21.0) in Emacs (30.2) using org-babel (9.7.37). All source code blocks were evaluated in place, and are entirely functional without any additional code beyond specified libraries. The accompanying .rmd file uses the same code, but in the correct order. This appendix is laid out for block readability, while the .rmd has the overall logic and hopefully reproducible method.

B Question 1 Visualisations

```r
suppressMessages(library(tidyverse))
suppressMessages(library(here))
suppressMessages(library(patchwork))

hard_drives <- read.table(here("data", "hard_drive.txt"))

hard_drives <- tibble(hard_drives[1])
hard_drives$V1 <- as.numeric(hard_drives$V1)

plt <- ggplot(data = hard_drives, aes(x=V1))
plt <- plt + geom_histogram(binwidth = 0.5)
plt <- plt + labs(title = "Lifetimes of batteries", x = "Years")

plt2 <- ggplot (data = hard_drives, aes(x=V1))
plt2 <- plt2 + geom_boxplot()
plt2 <- plt2 + labs(title = "Boxplot of battery lives", x = "Years")

(plt / plt2)
```

Listing 1: Question 1: Visualisation of hard drive lifetimes

```r
hist(x=hard_drives$V1, freq = FALSE, breaks = 10,
     main = "Histogram of hard drive lifetimes with fitted
         distributions",
     xlab = "Hard Drive Lifetimes (years)")

curve(dexp(x, rate = exp_mod$estimate["rate"]),
      add = TRUE, col = "red", lwd = 2)

curve(dgamma(x, rate = gam_mod$estimate["rate"],
             shape = gam_mod$estimate["shape"]),
      add = TRUE, col = "blue", lwd= 2)
```

Listing 2: Question 1: fitted model

```r
set.seed(1234)
exp_sim <- rexp(n = 10000, rate = exp_rate)
gam_sim <- rgamma(n = 10000, shape = gam_shape, rate = gam_rate)
sims <- data.frame(exp_sim, gam_sim)

## gamma histogram
plt3 <- ggplot(data = sims, aes(x=gam_sim))
plt3 <- plt3 + geom_histogram(binwidth = 0.5)
plt3 <- plt3 + labs(title = "Gamma simulated lifetimes", x = "Years")

## gamma boxplot
```

```r
plt4 <- ggplot (data = sims, aes(x=gam_sim))
plt4 <- plt4 + geom_boxplot()
plt4 <- plt4 + labs(title = "Gamma simulated summary", x = "Years")

## exponential histogram
plt5 <- ggplot(data = sims, aes(x=exp_sim))
plt5 <- plt5 + geom_histogram(binwidth = 0.5)
plt5 <- plt5 + labs(title = "Exponential simulated lifetimes", x = "Years")

## gamma boxplot
plt6 <- ggplot (data = sims, aes(x=exp_sim))
plt6 <- plt6 + geom_boxplot()
plt6 <- plt6 + labs(title = "Exponential simulated summary", x = "Years")

## plt  plt2
## plt3 plt4
## plt5 plt6

(plt | plt2) / (plt3 | plt4) / (plt5 | plt6)
```

Listing 3: Question 1: Simulated models

C Question 1 Fitting

```r
suppressMessages(require(MASS))

exp_mod <- fitdistr(hard_drives$V1, "exponential")
gam_mod <- fitdistr(hard_drives$V1, "gamma")

exp_rate <- exp_mod$estimate['rate']
gam_shape <- gam_mod$estimate['shape']
gam_rate <- gam_mod$estimate['rate']
exp_lk <- exp_mod$loglik
gam_lk <- gam_mod$loglik

print("MASS stores key model fit information that we can access")
print("and read.")
sprintf("The exponential distribution rate is %.3f, and the gamma",
        exp_rate)
sprintf("distribution follows shape %.3f and rate %.3f",
        gam_shape, gam_rate)

print("The log-likelihood for the exponential distribution")
sprintf("is %f and the log-likelihood for the gamma",
        exp_lk)
sprintf("distribution is %f.", gam_mod$loglik)
```

Listing 4: Question 1: fitting gamma and exponential distributions

```
. + > [1] "MASS stores key model fit information that we can access"
[1] "and read."
[1] "The exponential distribution rate is 0.287, and the gamma"
[1] "distribution follows shape 0.926 and rate 0.266"
[1] "The log-likelihood for the exponential distribution"
[1] "is -449.682131 and the log-likelihood for the gamma"
[1] "distribution is -449.286574."
```

```r
ks.test(hard_drives, "pexp", exp_rate)
ks.test(hard_drives, "pgamma", shape = gam_shape, rate = gam_rate)
```

Listing 5: Question 1: Kolmogorov-Smirnov tests

```
 Asymptotic one-sample Kolmogorov-Smirnov test

data:  hard_drives
D = 0.057729, p-value = 0.5177
alternative hypothesis: two-sided


 Asymptotic one-sample Kolmogorov-Smirnov test

data:  hard_drives
D = 0.041533, p-value = 0.8805
alternative hypothesis: two-sided
```

```r
n <- dim(hard_drives)
AIC_exp <- 2 * 1 - 2 * exp_mod$loglik
AIC_gam <- 2 * 2 - 2 * gam_mod$loglik

BIC_exp <- 1 * log(n) - 2 * exp_mod$loglik
BIC_gam <- 2 * log(n) - 2 * gam_mod$loglik

fit_stats <- data.frame (
    Criterion = c("AIC", "BIC"),
    `Exponential Model` = c(AIC_exp, BIC_exp[1]),
    `Gamma Model` = c(AIC_gam, BIC_gam[1])
)

head(fit_stats)
```

Listing 6: AIC and BIC information for models

D Question 2 Visualisations

```r
dclose <- tibble(read.csv(here("data", "EuroIndices_Close.csv")))

## lubridate supplies dmy parsing
dclose$Date <- strptime(dclose$Date, "%b %d, %y")

plt <- ggplot(data = dclose, aes(x = as.POSIXct(Date)))
plt <- plt + geom_line(aes(y = AEX_Close, colour = "AEX")) +
    geom_line(aes(y = BEL20_Close, colour = "BEL20"))+
    geom_line(aes(y = CAC40_Close, colour = "CAC40")) +
    geom_line(aes(y = DAX_Close, colour = "DAX")) +
    labs(title = "European Indices",
         y = "Closing Price €()",
         x = "Time")
plt
```

Listing 7: Time series plotting of European Indices data

```
par(mfrow = c(2,1))

acf(dclose$DAX_Close, main = "ACFs for DAX Close")
pacf(dclose$DAX_Close, main = "Partial ACFs for DAX Close")

par(mfrow = c(1,1))
```

Listing 8: Visualisation of ACFs and PACFs for DAX closing price

```
Q2_arima_proj <- forecast(mod_arima_DAX, h = 10)

autoplot(Q2_arima_proj, ylab = "DAX20 Closing Price")
```

Listing 9: Predictions for DAX index from ARIMA model

E Question 2 Fitting

```
suppressMessages(library(aTSA))
## augmented dickey-fuller can test stationarity in the presence
## of serial-correlation
print("Dickey-Fuller test on raw dataset")
adf.test(x = dclose$DAX_Close, nlag = 2)

d1dclose <- dclose[-1, ]
d1dclose[,-1] <- lapply(dclose[,-1], diff)

print("Dickey-Fuller test on differenced dataset")
adf.test(x = d1dclose$DAX_Close, nlag = 2)
```

Listing 10: Stationarity testing DAX prices and first difference

```
[1] "Dickey-Fuller test on raw dataset"
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
     lag    ADF p.value
[1,]   0 -0.934   0.344
[2,]   1 -0.812   0.388
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -1.47   0.532
[2,]   1 -1.60   0.486
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -1.32   0.860
[2,]   1 -1.60   0.741
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
[1] "Dickey-Fuller test on differenced dataset"
Augmented Dickey-Fuller Test
alternative: stationary
```

```
Type 1: no drift no trend
     lag    ADF p.value
[1,]   0 -10.32    0.01
[2,]   1  -8.05    0.01
Type 2: with drift no trend
     lag    ADF p.value
[1,]   0 -10.31    0.01
[2,]   1  -8.06    0.01
Type 3: with drift and trend
     lag    ADF p.value
[1,]   0 -10.31    0.01
[2,]   1  -8.08    0.01
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
suppressMessages(library(forecast))

print("First we consider an AR(p) model.")
mod_AR_DAX <- arima(dclose$DAX_Close, c(1,0,0))

print("Second, we consider an MA(q) model.")
mod_MA_DAX <- arima(dclose$DAX_Close, c(0,0,1))

print("Finally, we select an ARIMA model.")
mod_arima_DAX <- auto.arima(dclose$DAX_Close)

AIC_AR <- mod_AR_DAX$aic
AIC_MA <- mod_MA_DAX$aic
AIC_ARIMA <- mod_arima_DAX$aic

sprintf("The AR model has an AIC of %.2f, the MA model has an AIC",
        AIC_AR)
sprintf("of %.2f, and the ARIMA model has an AIC of %.2f.",
        AIC_MA, AIC_ARIMA)
```

Listing 11: ARIMA fit on differenced DAX data

```
[1] "First we consider an AR(p) model."
[1] "Second, we consider an MA(q) model."
[1] "Finally, we select an ARIMA model."
[1] "The AR model has an AIC of 1890.03, the MA model has an AIC"
[1] "of 2221.60, and the ARIMA model has an AIC of 1869.57."
```

```r
Q2_arima_mod <- arima(dclose$DAX_Close, c(0,1,1))
Box.test(Q2_arima_mod$residuals, lag = 2, type = c("Ljung-Box"))
shapiro.test(Q2_arima_mod$residuals)
```

Listing 12: Question 2: Post-regression statistics for ARIMA model

```
 Box-Ljung test

data:  Q2_arima_mod$residuals
```

```
X-squared = 0.00021487, df = 2, p-value = 0.9999


 Shapiro-Wilk normality test

data:  Q2_arima_mod$residuals
W = 0.99529, p-value = 0.9134
```

F Question 2 VAR model

```
adf_checks <- (lapply(d1dclose[,-1], function(x)
    adf.test(x, nlag = 2)$p.value)) # all stationary at 99% CI

coint.test(d1dclose$DAX_Close, d1dclose$AEX_Close, nlag = 2)
coint.test(d1dclose$DAX_Close, d1dclose$BEL20_Close, nlag = 2)
coint.test(d1dclose$DAX_Close, d1dclose$CAC40_Close, nlag = 2)
coint.test(d1dclose$AEX_Close, d1dclose$BEL20_Close, nlag = 2)
coint.test(d1dclose$AEX_Close, d1dclose$CAC40_Close, nlag = 2)
coint.test(d1dclose$BEL20_Close, d1dclose$CAC40_Close, nlag = 2)

cointegration <- data.frame(Indices = c("AEX", "BEL20", "CAC40", "DAX")
    ,
                            AEX = c(NA, 0.01, 0.01, 0.01),
                            BEL20 = c(0.01, NA, 0.01, 0.01),
                            CAC40 = c(0.01, 0.01, NA, 0.01),
                            DAX = c(0.01, 0.01, 0.01, NA))

cointegration
```

Listing 13: Stationarity and Cointegration tests

```
suppressMessages(library(vars))

mod_var <- VARselect(d1dclose[,-1], lag.max = 5, type = c("const"))
mod_var
```

Listing 14: Question 2 VAR model initialisation

```
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     2      1      1      2

$criteria
                  1            2            3            4            5
AIC(n) 2.137585e+01 2.131110e+01 2.143224e+01 2.151343e+01 2.161659e+01
HQ(n)  2.154346e+01 2.161279e+01 2.186802e+01 2.208329e+01 2.232053e+01
SC(n)  2.178832e+01 2.205355e+01 2.250468e+01 2.291584e+01 2.334898e+01
FPE(n) 1.920711e+09 1.801253e+09 2.035923e+09 2.213536e+09 2.463781e+09
```

```
VAR_temp_mod <- VAR(d1dclose[,-1], p = 2)
var_proj <- predict(VAR_temp_mod, h = 10)
par(mar = c(4, 4, 2, 1))
plot(var_proj)
par(mfrow = c(1,1))
```

Listing 15: Question 2: VAR Forecasting

```
## serial correlation
serial.test(VAR_temp_mod)

## normality tests
normality.test(VAR_temp_mod)
```

Listing 16: Question 2: VAR post-regression statistics

```
 Portmanteau Test (asymptotic)

data:  Residuals of VAR object VAR_temp_mod
Chi-squared = 222.99, df = 224, p-value = 0.5064

$JB

 JB-Test (multivariate)

data:  Residuals of VAR object VAR_temp_mod
Chi-squared = 33.316, df = 8, p-value = 5.401e-05


$Skewness

 Skewness only (multivariate)

data:  Residuals of VAR object VAR_temp_mod
Chi-squared = 10.963, df = 4, p-value = 0.02698


$Kurtosis

 Kurtosis only (multivariate)

data:  Residuals of VAR object VAR_temp_mod
Chi-squared = 22.353, df = 4, p-value = 0.0001705
```

G Question 3 Visualisations

```
deaths <- tibble(read.csv(here("data", "deaths.csv")))

deaths$date <- strptime(deaths$date, "%Y-%m-%d")

par(mfrow = c(3,1))

ts.plot(deaths$wk.deaths, main = "Deaths over time",
        ylab = "Deaths")
acf(deaths$wk.deaths, main = "ACFs for deaths")
```

```
pacf(deaths$wk.deaths, main = "PACFs for deaths")

par(mfrow = c(1,1))
```

Listing 17: Question 3: Visualisation of deaths and ACFs

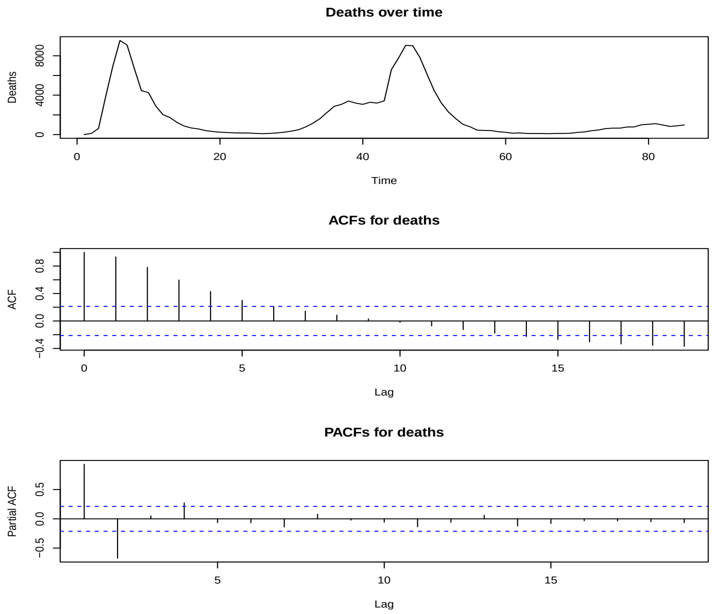


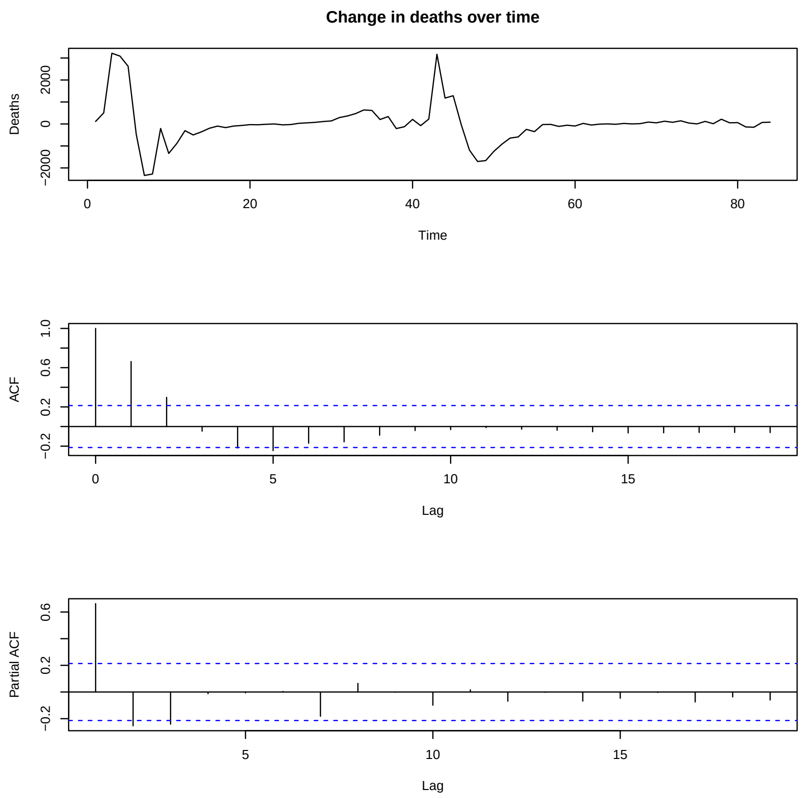
Figure 14: Question 3: Visualisation of deaths and ACFs

```
par(mfrow = c(3,1))

ts.plot(d.deaths$wk.deaths, main = "Change in deaths over time",
        ylab = "Deaths")
acf(d.deaths$wk.deaths, main = " ")
pacf(d.deaths$wk.deaths, main = " ")

par(mfrow = c(1,1))
```

Listing 18: Question 3: Visualisation of change in deaths and ACFs




```
death_mod <- auto.arima(deaths$wk.deaths, d = 1)
death_mod # fixed d = 1 or default auto.arima picks a worse model

proj_deaths <- forecast::forecast(death_mod, h = 2)
autoplot(proj_deaths, main = "Death projections for next two weeks",
         ylab = "Deaths")
```

Listing 19: Question 3: Projections of deaths using base ARIMA model

```
sqdeaths <- deaths
sqdeaths$wk.deaths <- sqrt(deaths$wk.deaths)

mod_sqd <- auto.arima(sqdeaths$wk.deaths, d = 1)

sqd_proj <- forecast::forecast(mod_sqd, h = 2)

sqd_proj$x <- sqd_proj$x^2
sqd_proj$fitted <- sqd_proj$fitted^2
sqd_proj$lower <- sqd_proj$lower^2
sqd_proj$upper <- sqd_proj$upper^2
sqd_proj$mean <- sqd_proj$mean^2
sqd_proj$residuals <- sqd_proj$residuals^2

autoplot(sqd_proj,
         main = "Back-fitted projections from rooted deaths",
         ylab = "Deaths")
```

Listing 20: Question 3: Projections of deaths using backfitted ARIMA model

```r
temp_death_mod <- arima(deaths$wk.deaths, c(3, 1, 0))

ts.diag(temp_death_mod)
```

Listing 21: Question 3: Visual residuals testing for base model

```r
temp_mod_sqd <- arima(sqdeaths$wk.deaths, c(2,1,2))

ts.diag(temp_mod_sqd)
```

Listing 22: Question 3: Visual residuals testing for back-fitted model

```r
death_mod
mod_sqd
```

Listing 23: Question 3: Model values

H Question 3 Fitting

```r
adf.test(deaths$wk.deaths, nlag = 3)

d.deaths <- deaths[-1,]
d.deaths$wk.deaths <- diff(deaths$wk.deaths)

adf.test(d.deaths$wk.deaths, nlag = 3)
```

Listing 24: Question 3: Stationarity testing

```
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
     lag   ADF p.value
[1,]   0 -1.27  0.2220
[2,]   1 -2.97  0.0100
[3,]   2 -2.35  0.0207
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -1.70  0.4430
[2,]   1 -3.95  0.0100
[3,]   2 -3.16  0.0273
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -1.95  0.5924
[2,]   1 -4.22  0.0100
[3,]   2 -3.42  0.0568
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
Augmented Dickey-Fuller Test
alternative: stationary
```

```
Type 1: no drift no trend
     lag   ADF p.value
[1,]   0 -4.08    0.01
[2,]   1 -4.78    0.01
[3,]   2 -6.35    0.01
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -4.05    0.01
[2,]   1 -4.75    0.01
[3,]   2 -6.31    0.01
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -4.03  0.0124
[2,]   1 -4.72  0.0100
[3,]   2 -6.22  0.0100
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
print("Testing for serial correlation:")
Box.test(death_mod$residuals, lag = 2, type = c("Ljung-Box"))
Box.test(mod_sqd$residuals, lag = 2, type = c("Ljung-Box"))

print("Testing for inappropriate residuals:")
shapiro.test(death_mod$residuals)
shapiro.test(mod_sqd$residuals)
```

Listing 25: Question 3: Post-regression residual testing

```
[1] "Testing for serial correlation:"

 Box-Ljung test

data:  death_mod$residuals
X-squared = 0.1935, df = 2, p-value = 0.9078


 Box-Ljung test

data:  mod_sqd$residuals
X-squared = 0.49706, df = 2, p-value = 0.7799

[1] "Testing for inappropriate residuals:"

 Shapiro-Wilk normality test

data:  death_mod$residuals
W = 0.72277, p-value = 2.252e-11


 Shapiro-Wilk normality test
```

```
data:  mod_sqd$residuals
W = 0.77624, p-value = 4.818e-10
```

I Question 4 Visualisations

```r
ex_rat <- tibble(read.csv(here("data", "USD-GBP.csv")))

ex_rat$Date <- strptime(ex_rat$Date, "%d/%m/%Y")

ex_returns <- ex_rat[-1,]
ex_returns$USD.GBP.Close <- diff(log(ex_rat$USD.GBP.Close))

ts.plot(ex_returns$USD.GBP.Close)
```

Listing 26: Question 4: Logged returns visualised

```r
ts.diag(arima_mod_post)
```

Listing 27: Visual residuals test for ARIMA model

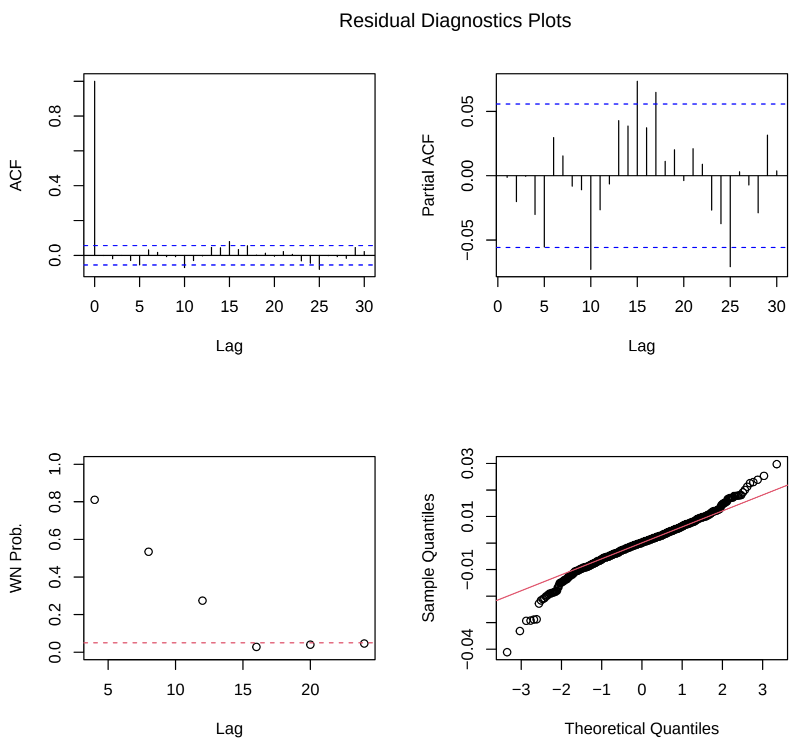


Figure 15: Visual residuals test for ARIMA model

J Question 4 Fitting

```
adf.test(ex_returns$USD.GBP.Close, nlag = 2)
```

Listing 28: Question 4: Stationarity testing

```
+ + > null device
          1
> . + > Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
     lag   ADF p.value
[1,]   0 -57.3    0.01
[2,]   1 -40.9    0.01
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -57.3    0.01
[2,]   1 -40.9    0.01
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -57.3    0.01
[2,]   1 -40.9    0.01
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```
## We can use dplyr to filter by criteria
## I think tidyverse masks filter but we call it explicitly
pre_crash <- ex_returns %>%
    dplyr::filter(Date < as.POSIXct("2007-08-01"))

post_crash <- ex_returns %>%
    dplyr::filter(Date >= as.POSIXct("2007-08-01"))

pre_dur <- range(pre_crash$Date)
post_dur <- range(post_crash$Date)

sprintf("The pre-crash data ranges from %s to %s, and",
        as.character(pre_dur[1,]), as.character(pre_dur[2]))
sprintf("the post-crash data ranges from %s to %s.",
        as.character(post_dur[1,]), as.character(post_dur[2]))
```

Listing 29: Pre-Post Crisis split

```
auto.arima(pre_crash$USD.GBP.Close)
auto.arima(post_crash$USD.GBP.Close)

arima_mod_pre <- arima(pre_crash$USD.GBP.Close, c(0,0,1))
arima_mod_post <- arima(post_crash$USD.GBP.Close, c(0,0,1))

shapiro.test(arima_mod_pre$residuals)
shapiro.test(arima_mod_post$residuals)

Box.test(arima_mod_pre$residuals, lag = 2, type = c("Ljung-Box"))
Box.test(arima_mod_post$residuals, lag = 2, type = c("Ljung-Box"))
```

Listing 30: Question 4: ARIMA mean model fitting

```
Series: pre_crash$USD.GBP.Close
ARIMA(0,0,1) with zero mean

Coefficients:
          ma1
      -0.1134
s.e.   0.0232

sigma^2 = 3.293e-05:  log likelihood = 7076.01
AIC=-14148.01   AICc=-14148   BIC=-14136.92
Series: post_crash$USD.GBP.Close
ARIMA(0,0,1) with zero mean

Coefficients:
         ma1
      0.0630
s.e.  0.0289

sigma^2 = 4.968e-05:  log likelihood = 4381.63
AIC=-8759.26   AICc=-8759.25   BIC=-8749.02

 Shapiro-Wilk normality test

data:  arima_mod_pre$residuals
W = 0.87379, p-value < 2.2e-16


 Shapiro-Wilk normality test

data:  arima_mod_post$residuals
W = 0.97399, p-value = 3.848e-14


 Box-Ljung test

data:  arima_mod_pre$residuals
X-squared = 0.86386, df = 2, p-value = 0.6493


 Box-Ljung test

data:  arima_mod_post$residuals
X-squared = 0.50344, df = 2, p-value = 0.7775
```

```r
suppressMessages(library(FinTS))

```

```
ArchTest(pre_crash$USD.GBP.Close)
ArchTest(post_crash$USD.GBP.Close)
```

Listing 31: Question 4: ARCH test

```
 ARCH LM-test; Null hypothesis: no ARCH effects

data:  pre_crash$USD.GBP.Close
Chi-squared = 701.23, df = 12, p-value < 2.2e-16


 ARCH LM-test; Null hypothesis: no ARCH effects

data:  post_crash$USD.GBP.Close
Chi-squared = 164.29, df = 12, p-value < 2.2e-16
```

```
suppressMessages(library(fGarch))

## iterating over models for pre-crisis
grm_pre <- garchFit(formula = USD.GBP.Close ~ arma(1,0) + garch(1,1),
                data = pre_crash,
                trace = FALSE) # -6.692718 -6.525458, NaNs produced

grm_pre <- garchFit(formula = USD.GBP.Close ~ arma(0,0) + garch(1,1),
                data = pre_crash,
                trace = FALSE) # -5.956245 -5.822436, NaNs produced

grm_pre <- garchFit(formula = USD.GBP.Close ~  arma(1,2) + garch(1,1),
                data = pre_crash,
                trace = FALSE) # -7.667488 -7.433323


## iterating over models for post-crisis
grm_post <- garchFit(formula = USD.GBP.Close ~  + garch(1,1),
                data = post_crash,
                trace = FALSE) # -1.665750 -1.657883

grm_post <- garchFit(formula = USD.GBP.Close ~ arma(2,1)+ garch(1,1),
                data = post_crash,
                trace = FALSE) # -6.434903 -6.421137


summary(grm_pre)
summary(grm_post)
```

Listing 32: Question 4: GARCH Model Fitting

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = USD.GBP.Close ~ arma(1, 2) + garch(1, 1),
    data = pre_crash, trace = FALSE)
```

```
Mean and Variance Equation:
 USD.GBP.Close ~ arma(1, 2) + garch(1, 1)
 [data = pre_crash]

Conditional Distribution:
 norm

Coefficient(s):
         mu          ar1           ma1           ma2         omega        alpha1         beta1
 3.5594e-04  -9.9023e-01   9.9098e-01   7.8292e-03   2.5224e-07   4.6400e-
02   9.4962e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      3.559e-04   2.318e-04    1.535   0.1247
ar1    -9.902e-01   1.007e-02  -98.307  < 2e-16 ***
ma1     9.910e-01   2.773e-02   35.734  < 2e-16 ***
ma2     7.829e-03   2.543e-02    0.308   0.7582
omega   2.522e-07   1.167e-07    2.161   0.0307 *
alpha1  4.640e-02   6.961e-03    6.666 2.63e-11 ***
beta1   9.496e-01   6.850e-03  138.632  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 7190.182    normalized:  3.802317

Description:
 Wed Dec 17 16:58:12 2025 by user:


Standardised Residuals Tests:
                                   Statistic   p-Value
 Jarque-Bera Test   R    Chi^2  4.695102e+04 0.0000000
 Shapiro-Wilk Test  R    W      9.291171e-01 0.0000000
 Ljung-Box Test     R    Q(10)  3.098746e+00 0.9790037
 Ljung-Box Test     R    Q(15)  7.088517e+00 0.9551467
 Ljung-Box Test     R    Q(20)  1.292927e+01 0.8803982
 Ljung-Box Test     R^2  Q(10)  1.063699e+01 0.3864895
 Ljung-Box Test     R^2  Q(15)  1.090979e+01 0.7589627
 Ljung-Box Test     R^2  Q(20)  1.184517e+01 0.9213021
 LM Arch Test       R    TR^2   1.067555e+01 0.5569095

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-7.597231 -7.576705 -7.597258 -7.589673
```

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = USD.GBP.Close ~ arma(2, 1) + garch(1, 1),
    data = post_crash, trace = FALSE)

Mean and Variance Equation:
 USD.GBP.Close ~ arma(2, 1) + garch(1, 1)
 [data = post_crash]

Conditional Distribution:
 norm

Coefficient(s):
         mu           ar1           ar2           ma1         omega        alpha1          beta1
-4.9340e-05   1.9265e-01  -2.5909e-02  -1.8975e-01   2.9106e-07   4.4219e-
02   9.4923e-01

Std. Errors:
 based on Hessian

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu     -4.934e-05   1.409e-04   -0.350    0.726
ar1     1.927e-01   7.254e-01    0.266    0.791
ar2    -2.591e-02   2.901e-02   -0.893    0.372
ma1    -1.898e-01   7.260e-01   -0.261    0.794
omega   2.911e-07   1.772e-07    1.643    0.100
alpha1  4.422e-02   8.935e-03    4.949 7.47e-07 ***
beta1   9.492e-01   1.014e-02   93.625  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 4510.89    normalized:  3.640751

Description:
 Wed Dec 17 16:58:13 2025 by user:


Standardised Residuals Tests:
                                Statistic    p-Value
 Jarque-Bera Test   R    Chi^2   8.5047991 0.01423005
 Shapiro-Wilk Test  R    W       0.9968676 0.01431826
 Ljung-Box Test     R    Q(10)   3.8510716 0.95381538
 Ljung-Box Test     R    Q(15)   8.9034602 0.88251084
 Ljung-Box Test     R    Q(20)   9.5765574 0.97521655
```

```
 Ljung-Box Test     R^2  Q(10)   7.0702637 0.71879540
 Ljung-Box Test     R^2  Q(15)  19.3286245 0.19922658
 Ljung-Box Test     R^2  Q(20)  23.5741990 0.26148675
 LM Arch Test       R    TR^2    8.3446546 0.75765023

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-7.270202 -7.241264 -7.270266 -7.259319
```