ACTL30007 AM3 : ACTL90020 GIM

Practice Final Exam

Question 1

The attached dataset `practice_q1.xlsx` contains claims amounts in a travel insurance personal items cover:

- The column `claim` corresponds to the actual amount paid;
- The column `limit` corresponds to the limit applicable to that claim;
- The column `deductible` corresponds to the deductible applicable to that claim.

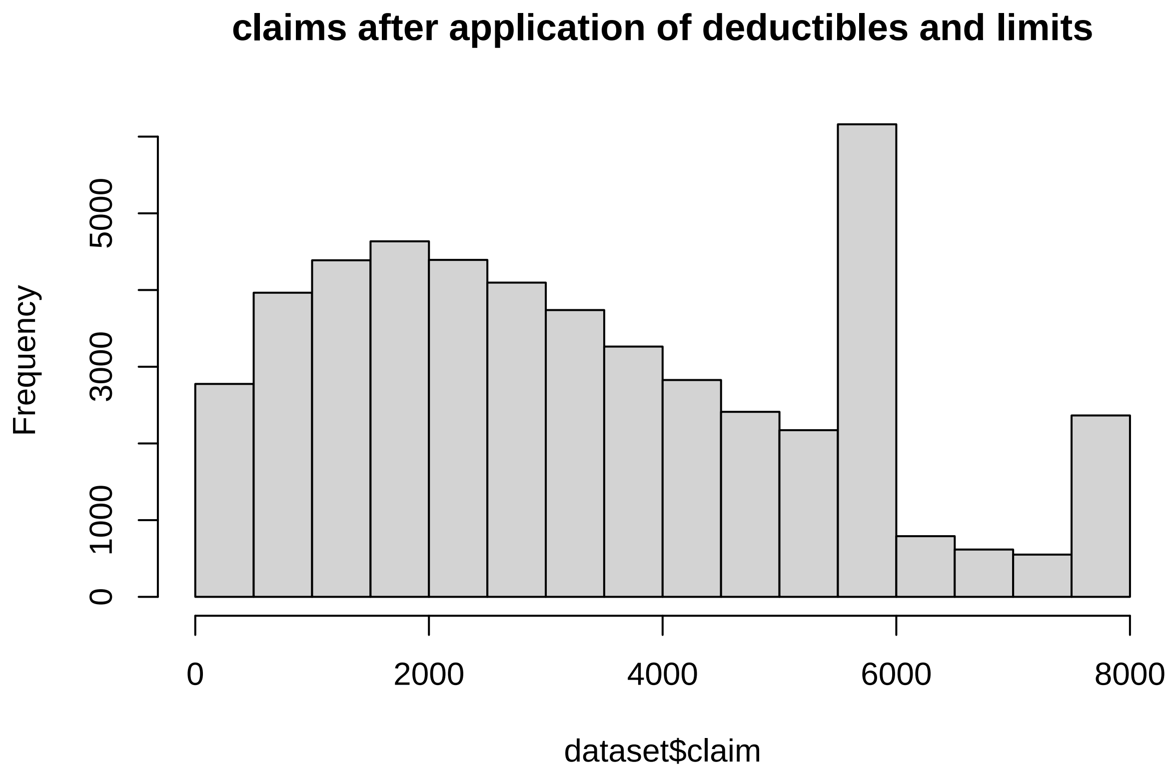
Note that if `claim`=`limit` then you can assume that the claim has been censored at that level, that is, the claim has reached its applicable limit.

Perform the following tasks:

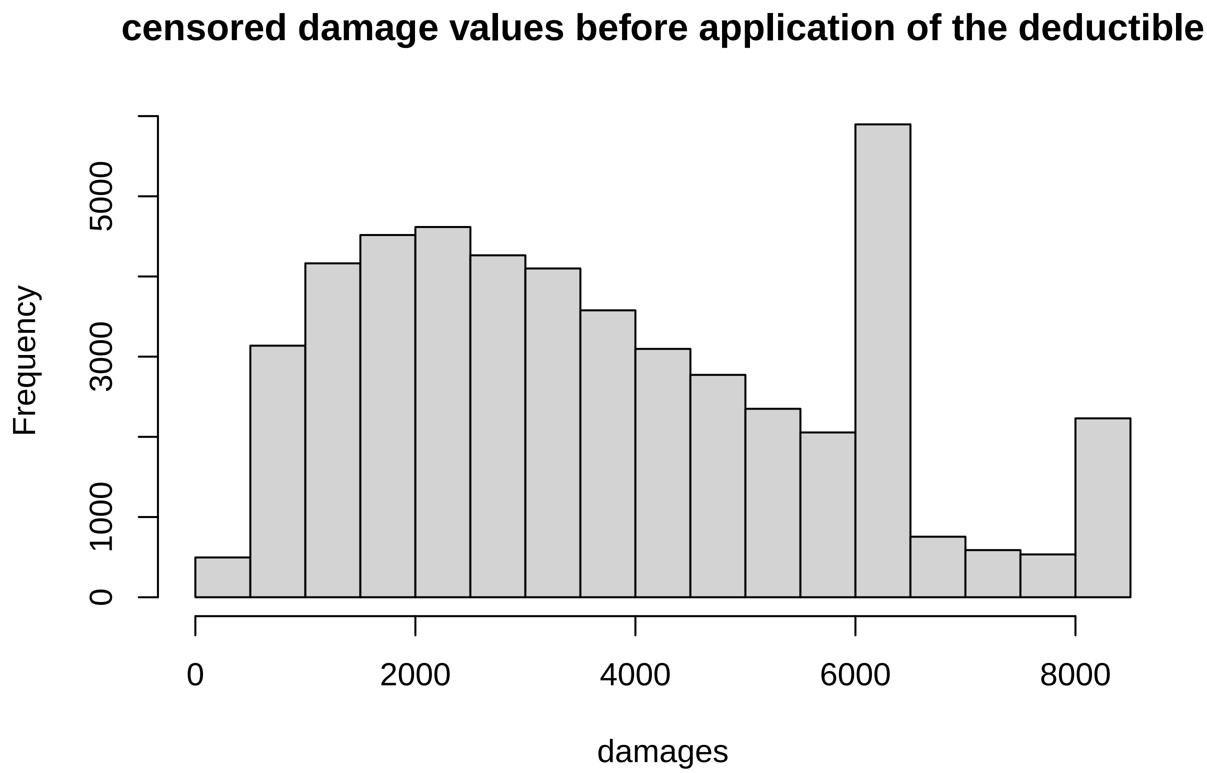
(a) Provide two histograms: one for the claims, and one for the raw (before the deductible is applied) damage costs, which could be censored. Explain which of their features are due to the existence of deductibles and limits. (2 marks)

The histograms are as follows:

```r
dataset <- readxl::read_xlsx("Practice_Final_Datasets/practice_q1_data.xlsx")

hist(dataset$claim, main = "claims after application of deductibles and limits")
```




```r
damages <- dataset$claim + dataset$deductible
hist(damages, main = "censored damage values before application of the deductible")
```

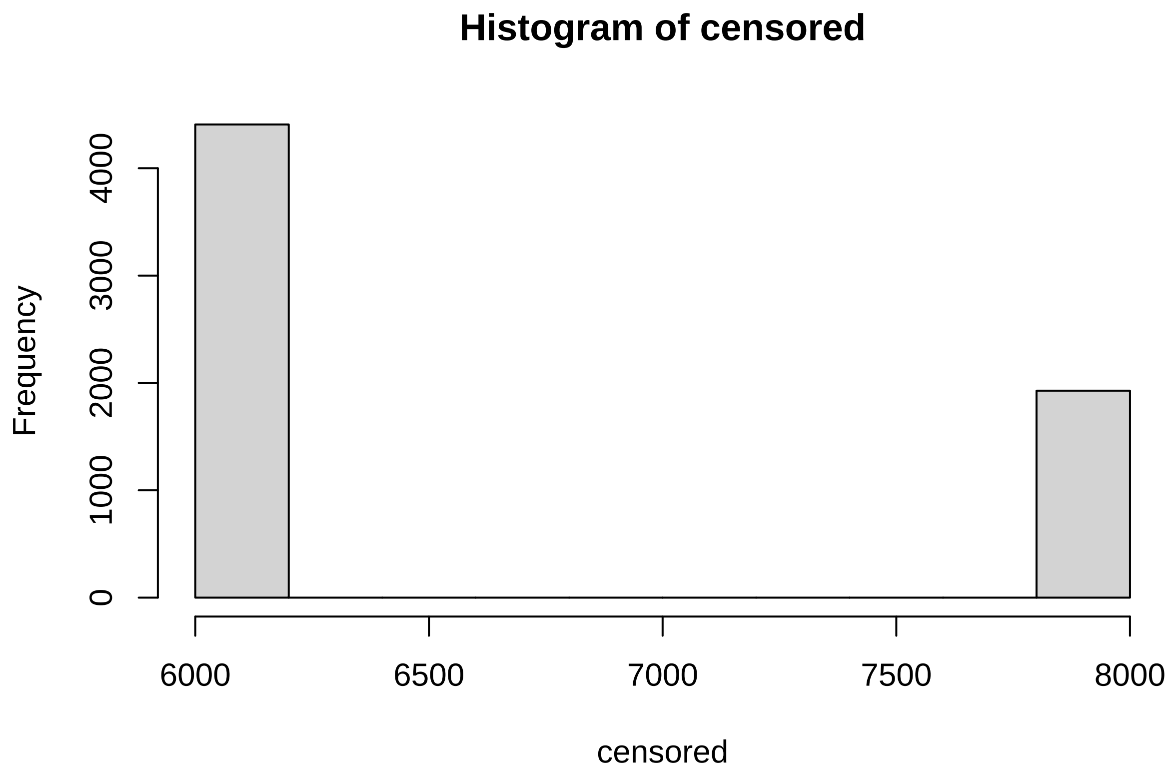


In both histograms there are two large spikes at around 6000 and 8000, which are typical of an aggregation of claim amounts due to the application of policy limits. The presence of deductibles is more subtle, and most obvious in the second histogram: there seems to be less claims in the first 500 than one would expect when looking at the shape of the distribution otherwise.

(b) Provide a histogram of those claims which have reached their limit. What are all the possible amounts for limits in this dataset? What is the number of claims that had reached the highest of those limits? (1 mark)

The histogram is obtained by filtering for those claims which are equal to their limit. The possible amounts are 6000 and 8000 (use, for instance, the R function `unique` on the filtered claims), and the number of censored claims at 8000 is 1928.

```
censored <- dataset$claim[dataset$claim == dataset$limit]
hist(censored)
```




```r
unique(censored)
```

```
## [1] 6000 8000
```

```r
length(censored[censored == 8000])
```

```
## [1] 1928
```

(c) We want to fit a distribution that describes the raw damage cost. For the 1st and 21st observations, write an expression for their contribution to the overall likelihood of this dataset, in terms of the pdf f, cdf F, and the relevant numerical values in the dataset (for instance, something like "$f(100) \times F(200)$"). (2 marks)

Let's extract the 1st and 21st claims

```r
# first claim:
dataset[1, ]
```

```
## # A tibble: 1 x 3
##   claim limit deductible
##   <dbl> <dbl>      <dbl>
## 1  2297  6000        500
```

```r
# 21st claim:
dataset[21, ]
```

```
## # A tibble: 1 x 3
##   claim limit deductible
##   <dbl> <dbl>      <dbl>
## 1  6000  6000        250
```

It turns out that the second claim is censored, but not the first, which will require a different treatment. Furthermore, one should add back the deductible to the claim, as we are modelling the damage variable. In the end we get:

- likelihood of the first claim: $f(2297+500)/(1-F(500))$
- likelihood of the 21st claim: $(1-F(6000+250))/(1-F(250))$ and the contribution to the overall likelihood is $f(2297+500)/(1-F(500)) \times (1-F(6000+250))/(1-F(250))$

(d) Ignoring truncation and censoring, fit a gamma distribution to the data using the method of moments. Provide the corresponding numerical estimates for the `shape` and `rate` parameters. (1 mark)

This is easily done using `fitdist` function or the sample mean and variance of the damages (note that the mme technique matches the population variance instead of sample variance, both approaches are acceptable):

```
gamma.mme = fitdistrplus::fitdist(damages, "gamma", method = "mme",
    order = 1:2)
gamma.mme$estimate
```

```
##         shape          rate
## 3.1456555180 0.0008381131
```

```
n <- length(damages)
shape0 <- mean(damages)^2/var(damages)
rate0 <- mean(damages)/var(damages)
shape0
```

```
## [1] 3.145592
```

```
rate0
```

```
## [1] 0.000838096
```

So that we have `shape`=3.1456555 and `rate`=8.3811306×10^{-4}.

(e) Calculate the overall negative loglikelihood of the whole dataset, if one assumes a gamma distribution with parameters `shape`=2, and `rate`=0.0005. Include the R code used for this specific part (and not earlier parts (a) to (d)). You are not required to optimise this negative loglikelihood, but only to calculate its value evaluated at `shape`=2, and `rate`=0.0005. (4 marks)

One can write the negative loglikelihood from scratch, or simply copy paste the function provided in the lecture slides:

```r
negloglik <- function(pdf, cdf, param, x, deduct, limitI) {
    # Function returns the negative log likelihood of the censored
    # and truncated dataset. Each data point's contribution to the
    # log likelihood depends on the theoretical distribution pdf and
    # cdf and also the deductible and limit values to adjust for
    # truncation and censoring
    PL <- do.call(cdf, c(list(q = deduct), param))
    PX <- do.call(cdf, c(list(q = x), param))
    fX <- do.call(pdf, c(list(x = x), param))
    lik.contr <- ifelse(limitI, log(1 - PX), log(fX)) - log(1 - PL)
    return(-sum(lik.contr))
}

answer <- negloglik(dgamma, pgamma, param = list(shape = 2,
    rate = 5e-04), damages, dataset$deductible, dataset$claim ==
    dataset$limit)
```

so that the answer is 394047.4.

Question 2

You are told that the following function

$$\phi_\theta(t) = e^{-t^{\frac{1}{\theta}}}, \quad \theta \in [1, \infty)$$

is either the generator function or the inverse generator function of an Archimedean copula.

(a) By examining the relevant property of an Archimedean copula, determine whether the function is a generator function or an inverse generator function. (1 mark)

The relevant property is $\psi(1) = 0$ if ψ is a generator function. Here $\phi_\theta(1) = e^{-1} \neq 0$, hence ϕ_θ is an inverse generator function $\phi_\theta \equiv \psi_\theta^{-1}$.

(b) By finding the inverse of $\phi_\theta(t)$ and following from part (a), express the corresponding bivariate Archimedean copula defined as

$$C_\theta(u, v) = \psi_\theta^{-1}\left(\psi_\theta(u) + \psi_\theta(v)\right).$$

in terms of u and v. (2 marks)

$$\begin{aligned} \phi_\theta^{-1}(t) \equiv \psi_\theta(t) &= (-\log t)^\theta \\ \Rightarrow C_\theta(u, v) &= \phi_\theta\left(\phi_\theta^{-1}(u) + \phi_\theta^{-1}(v)\right) \\ &= e^{-\left((-\log u)^\theta + (-\log v)^\theta\right)^{\frac{1}{\theta}}} \end{aligned}$$

(c) You are also told that the copula function is an increasing function of $\theta \in [1, \infty)$. Show that the copula does not contain the countermonotonicity copula. (2 marks)

For $\theta = 1$, we obtain

$$C_1(u, v) = e^{-((-\log u)+(-\log v))} = uv$$

Since $(u-1)(v-1) > 0 \Rightarrow uv > u + v - 1$ for $0 < u < 1, 0 < V < 1$, the fact that the copula is an increasing function of θ means that the independence copula uv is its lower bound, and so the copula does not contain the Frechet lower bound of $\max(u + v - 1, 0)$ defined by the countermonotonicity copula.

Question 3

The attached dataset `practice_q3_data.xlsx` corresponds to the `hurricanehist` dataset in the R package `CASdatasets`. It is described in the package documentation as "The dataset consists of 2010 observations for all tropical cyclones in the NHC best track record over the period 1899-2006. Each observation contains per cyclone maximum wind speeds and other relevant information."

The vignette goes on with describing: "`hurricanehist` contains 7 columns:

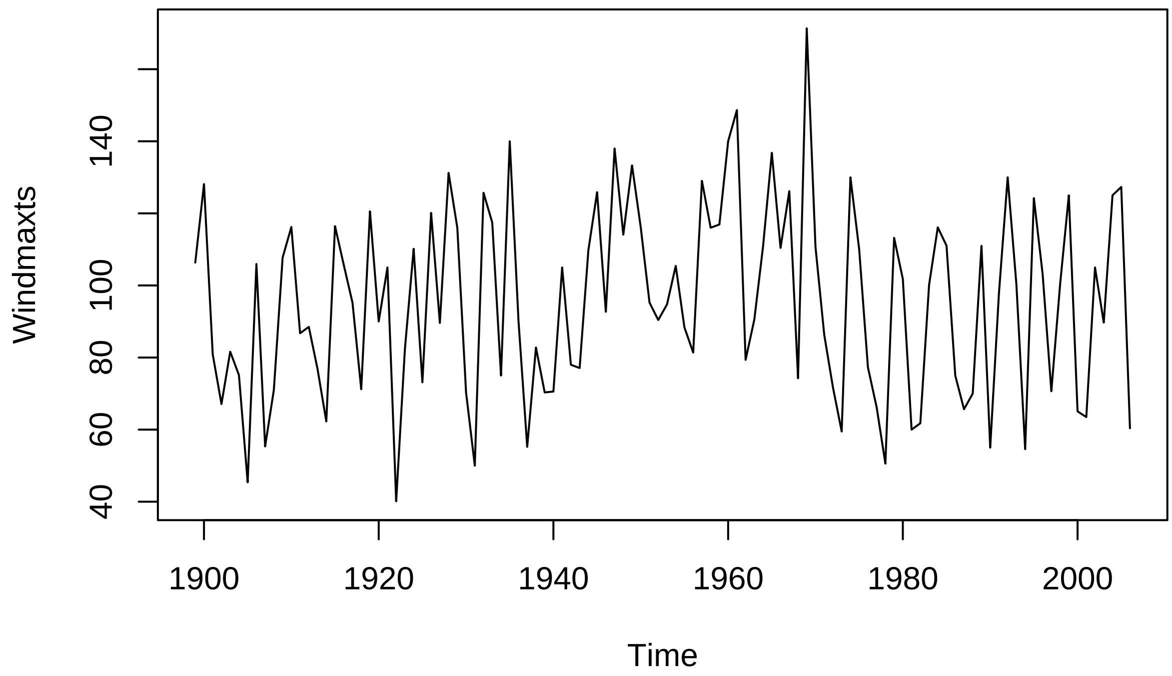
- `Year`: The Year.
- `Region`: The region among "Basin", "East Florida", "Gulf", "US".
- `Windmax`: The maximum windspeed in knot (1kt = 0.51 m/s).
- `NAO`: the North Atlantic Oscillation (NAO) index as an indicator of storm steering.
- `SOI`: the Southern Oscillation Index (SOI) as an indicator of El Nino-Southern Oscillation.
- `SST`: the Atlantic sea-surface temperature (SST) as an indicator of cyclone energy.
- `SSTmda`: the SST mda."

The original source of the data is http://myweb.fsu.edu/jelsner/_site/.

Perform the following tasks, which are focused on the "US" region only:

(a) Consider the maximum windspeeds in the region "US" only. Transform the data into a time series of yearly maxima, and include a time series plot (with the correct frequency and start date) in your answer, as well as the first 10 values. (3 marks)

```r
dataset <- readxl::read_xlsx("Practice_Final_Datasets/practice_q3_data.xlsx")

library(dplyr)

USmax <- dataset %>%
    filter(dataset$Region == "US") %>%
    group_by(Year) %>%
    summarise(Max_Windmax = max(Windmax))

Windmaxts <- ts(data = USmax$Max_Windmax, start = min(USmax$Year),
    frequency = 1)

plot(Windmaxts)
```




```r
USmax$Max_Windmax[1:10]
```

```
##  [1] 106.35318 128.12384  80.88609  67.09340  81.62150  75.12913  45.38930
##  [8] 105.94556  55.35019  71.06642
```

(b) Fit an appropriate Generalised Extreme Value (GEV) distribution for the ***yearly maxima*** of windspeed in the "US" region. Report the numerical values (and their standard errors) of the distribution parameters, and include the relevant diagnostic plots. State the type of the fitted GEV distribution and state the range of values of the yearly maxima. Discuss the goodness of fit of your results based on the diagnostic plots. (5 marks)

Because we already have a dataset of yearly maxima, there is no need to use the block maxima method. If we wanted a distribution of maxima per event, then we would need to use the block maxima method.

```r
fit.USmax <- extRemes::fevd(USmax$Max_Windmax)

fit.USmax
```

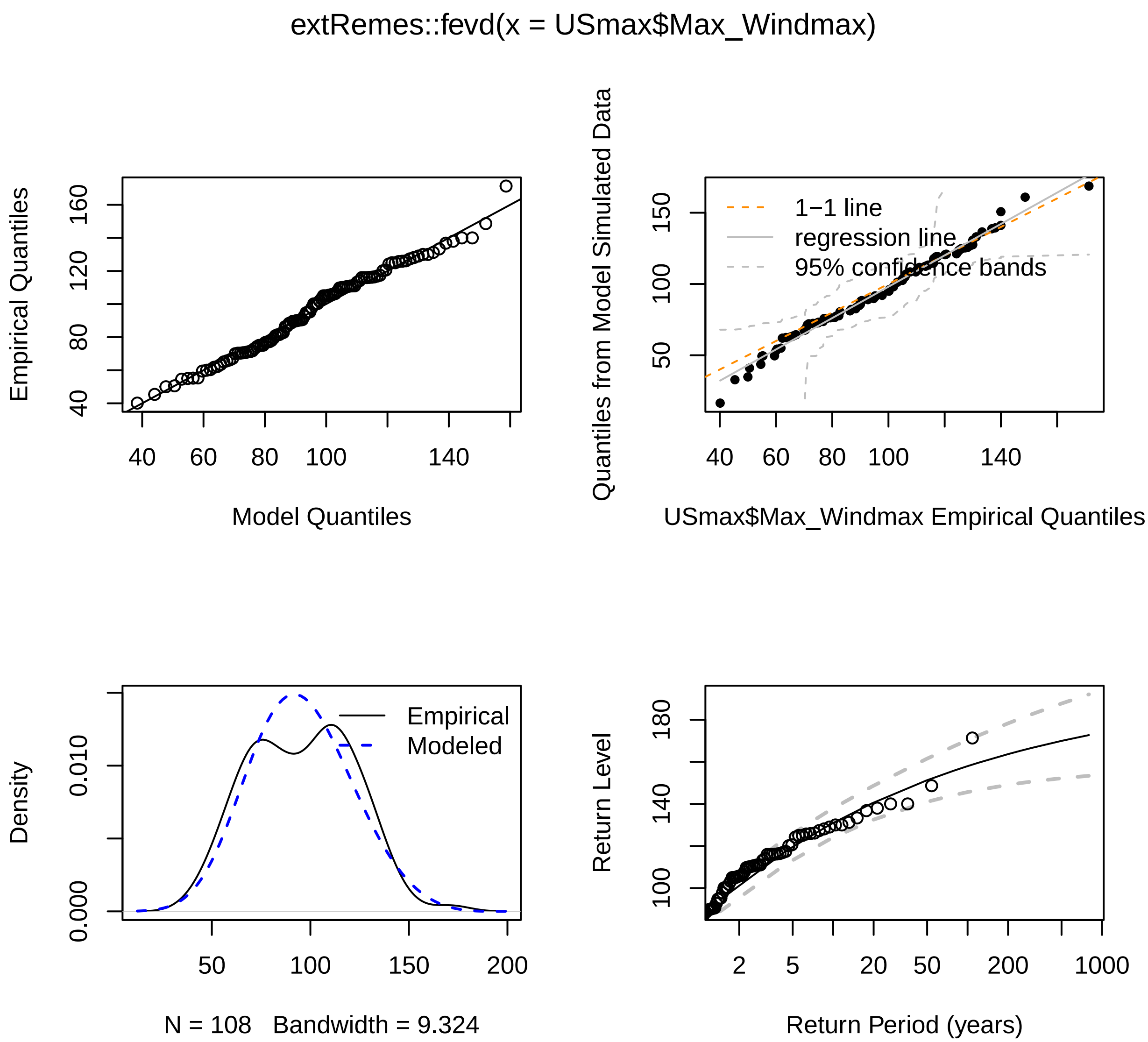
```
##
## extRemes::fevd(x = USmax$Max_Windmax)
##
## [1] "Estimation Method used: MLE"
```

```
##
##
##  Negative Log-Likelihood Value:  505.7602
##
##
##  Estimated parameters:
##   location      scale      shape
## 85.4656711 25.4042988 -0.2275732
##
##  Standard Error Estimates:
##   location      scale      shape
## 2.68824411 1.91816436 0.05739781
##
##  Estimated parameter covariance matrix.
##             location       scale        shape
## location  7.22665640  0.48676130 -0.056153148
## scale     0.48676130  3.67935453 -0.060548742
## shape    -0.05615315 -0.06054874  0.003294509
##
##  AIC = 1017.52
##
##  BIC = 1025.567
```

The estimated parameters of location, scale and shape and their standard errors are given above.

Since the shape parameters are negative and significant, it belongs to the upper bounded Weibull type with a support of $x \leq \mu - \frac{\sigma}{\gamma} = 197.097$.

```
plot(fit.USmax)
```



The diagnostic plots look reasonably good:

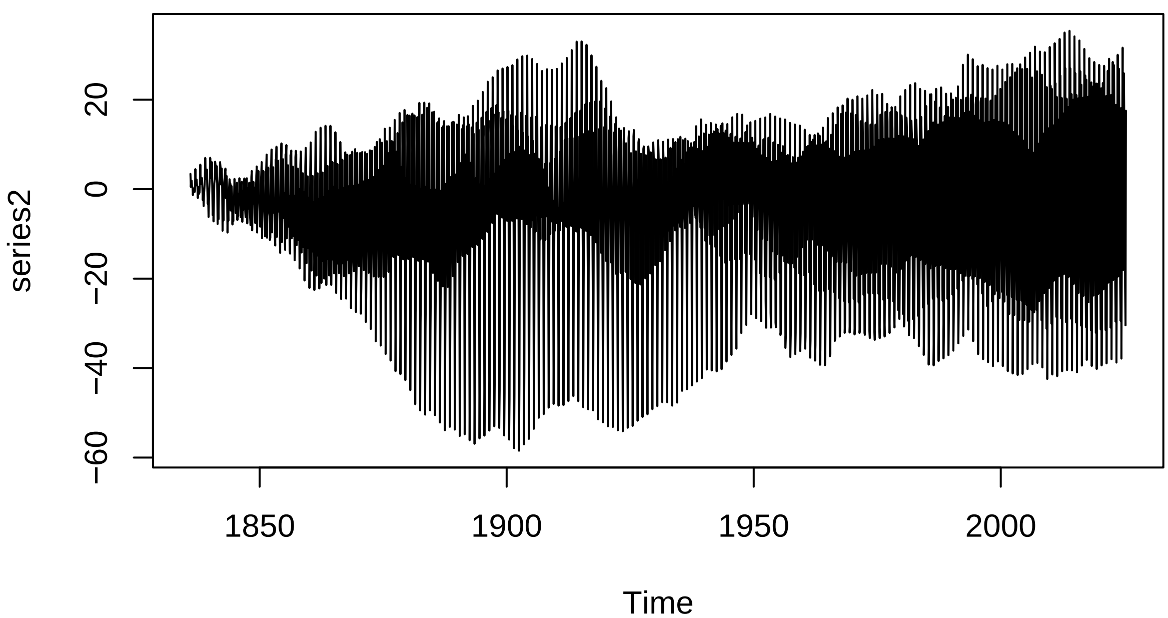
- the QQ plot (data vs model) is good, with most dots along the diagonal.
- the QQ plot (data vs simulated from model) is also quite good, with all dots well within the confidence bounds, and the regression line almost matching the diagonal.
- the density plot is well centered, and the bimodality of the data is simply smoothed into the unimodal EVD.

Question 4

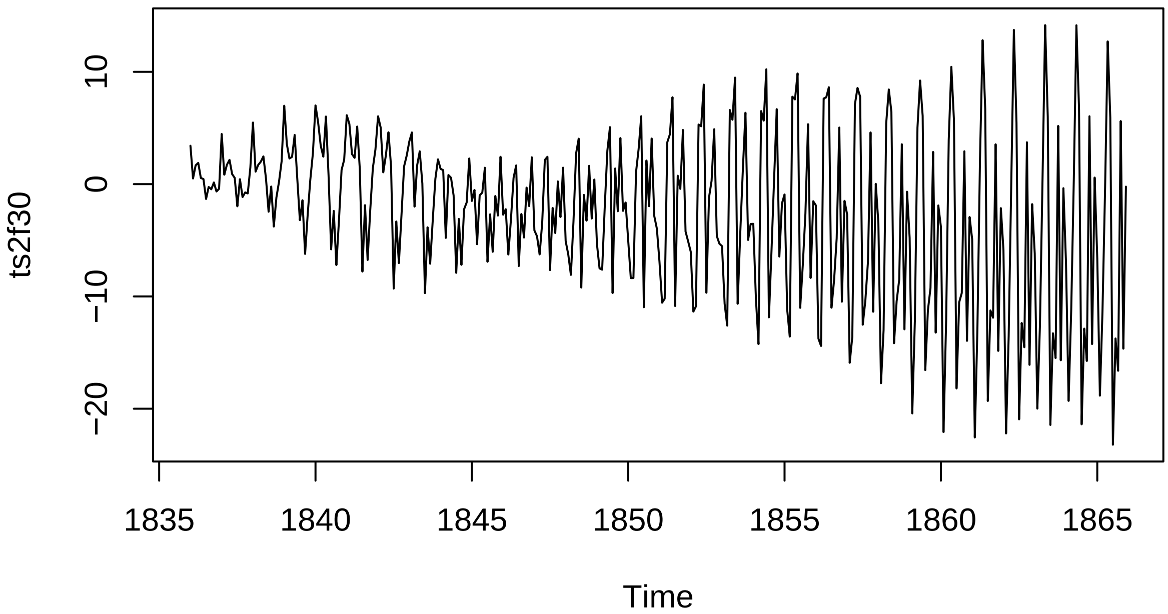
The dataset in `practice_q4_data.xlsx` contains monthly time series x_t starting from January 1836.

(a) Explore the time series x_t using relevant plots and discuss whether the series is likely to be stationary. Hint: you may want to plot the series for a specific sub-period and/or specify the maximum lag of your plot for better visualisation. (4 marks)

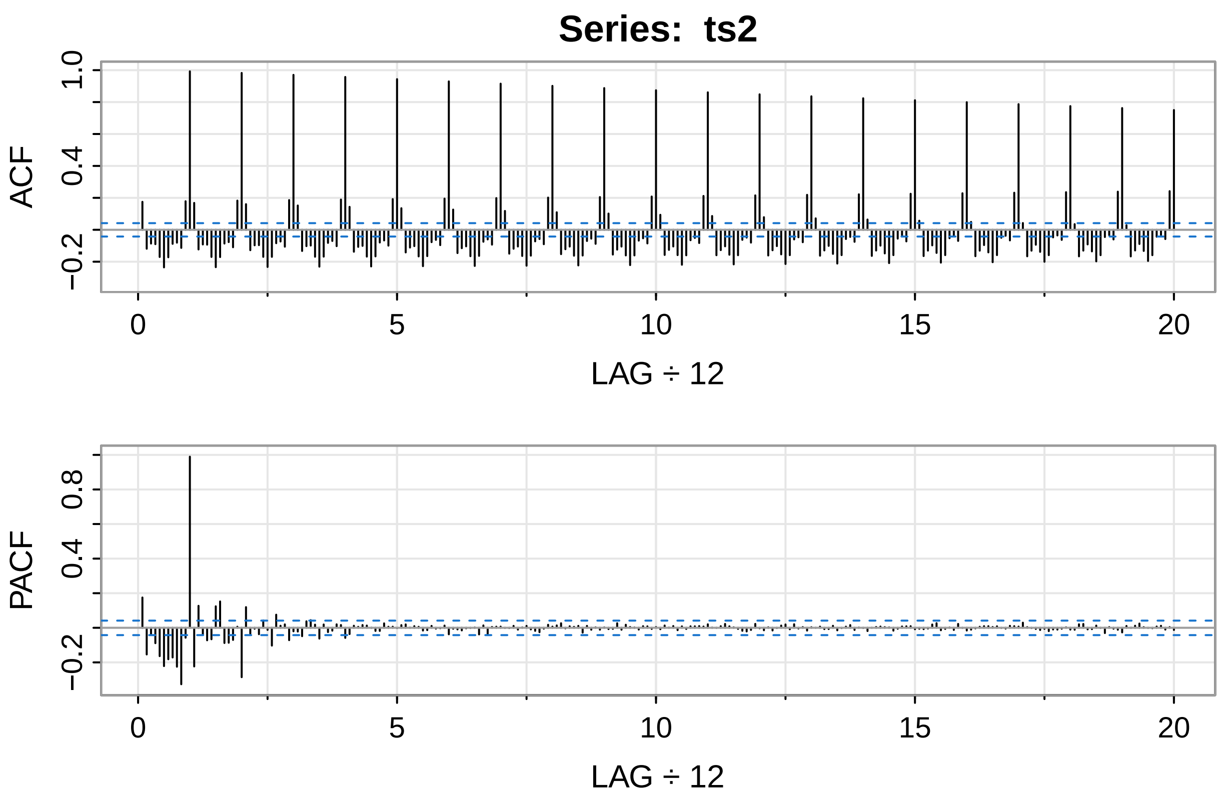
```
ts2 <- ts(data = read_xlsx("Practice_Final_Datasets/practice_q4_data.xlsx"),
    start = 1836, frequency = 12)

plot(ts2)
```




```
# plot the first 30 years of monthly data
ts2f30 <- ts(data = ts2[1:360], start = 1836, frequency = 12)
plot(ts2f30)
```




```r
# plot the acf and pacf for up to a maximum lag of 240
acf2(ts2, max.lag = 240)
```



There are obvious patterns of (i) non-constant mean, (ii) seasonal / cyclical component and (iii) increasing variance over time, suggesting that the series is not stationary.

We choose to plot the ACF and PACF up to a maximum lags of 20 years (240 months).

The extremely slow decaying speed of ACF at lag $12h$ also indicates that the series is not stationary.

(b) Using the plot in part (a), explain why de-trending via regression may not be a good idea for the series. (1 mark)

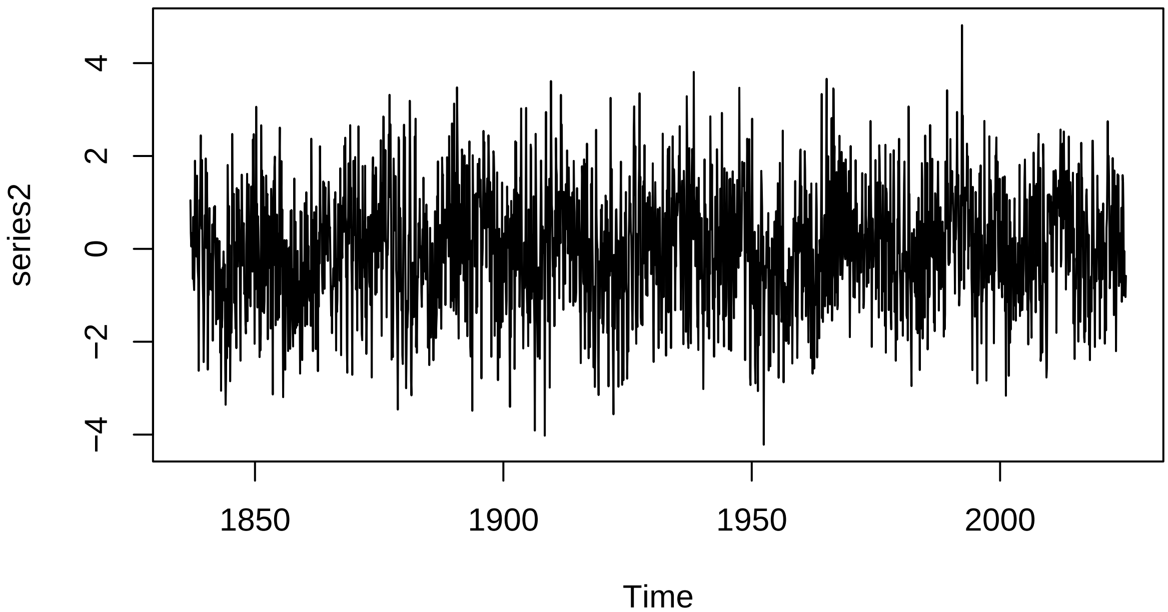
Looking at the data over close to 200 years, there is no obvious long-term deterministic trend underlying the data, due to the presence of seasonal effect.

(c) Perform the necessary next step in the modelling by considering potential integrated orders of

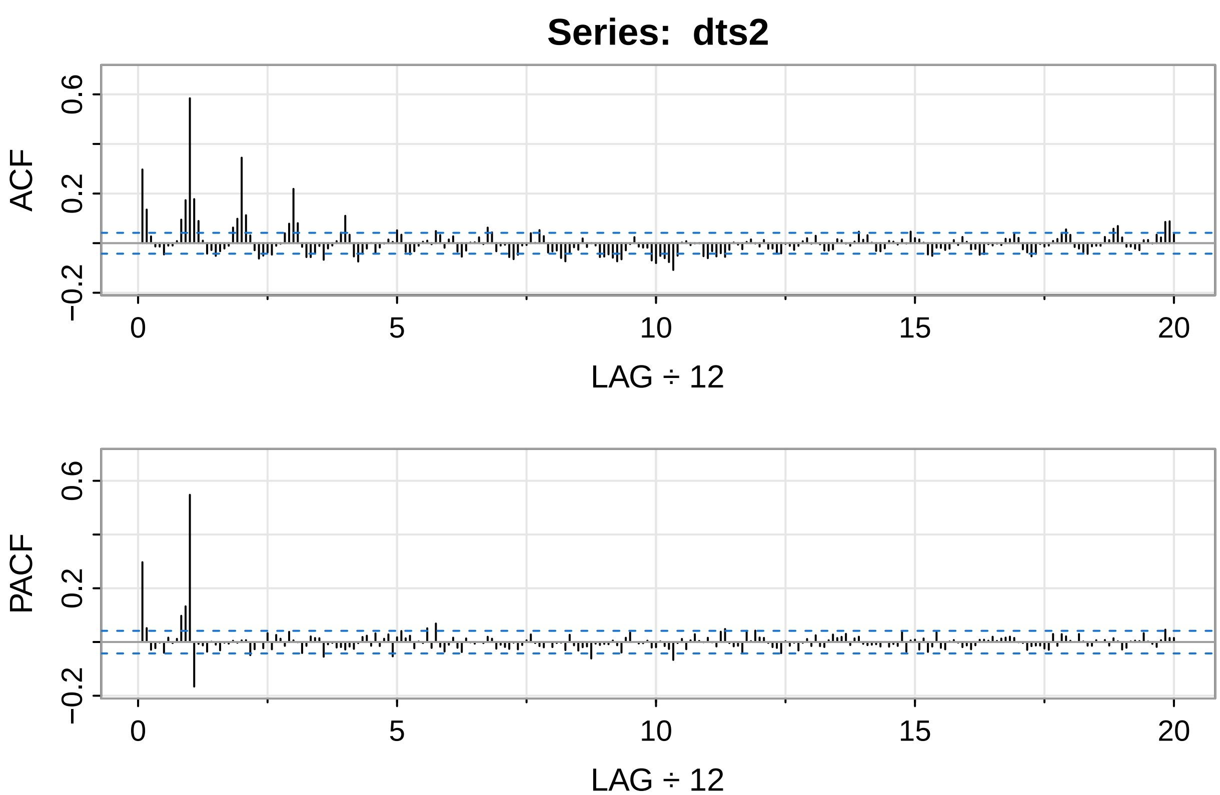
- $d = 0, D = 1$
- $d = 1, D = 0$

Comment on your findings, verify against your discussion in part (a) and determine the appropriate integrated orders. (4 marks)

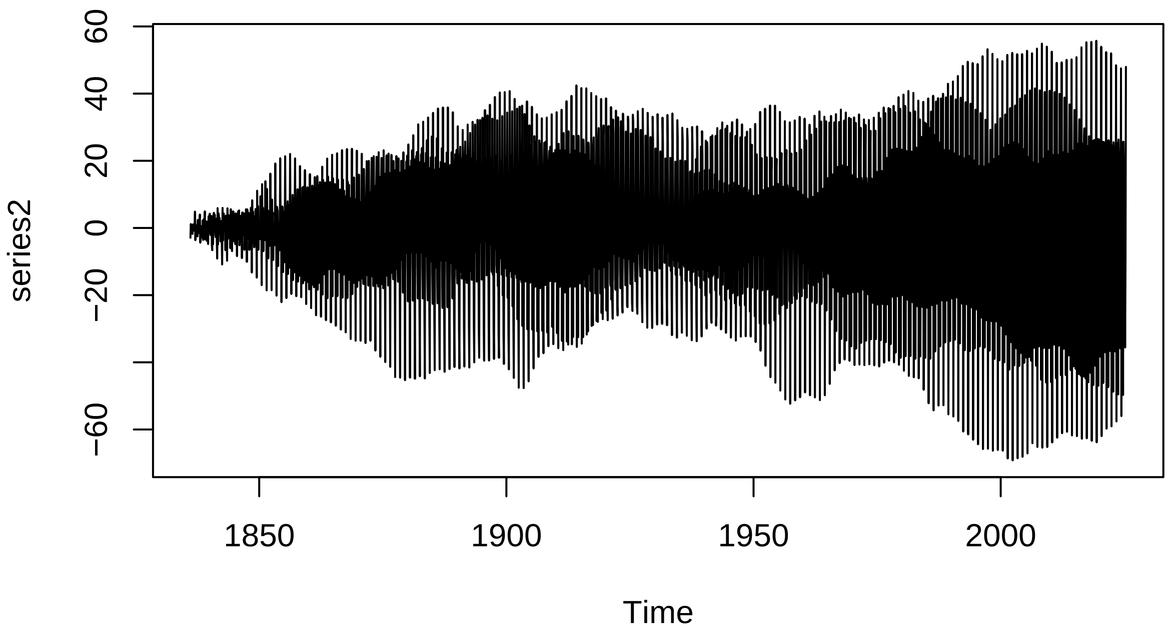
```
dts2 <- diff(ts2, lag = 12)
plot(dts2)
```



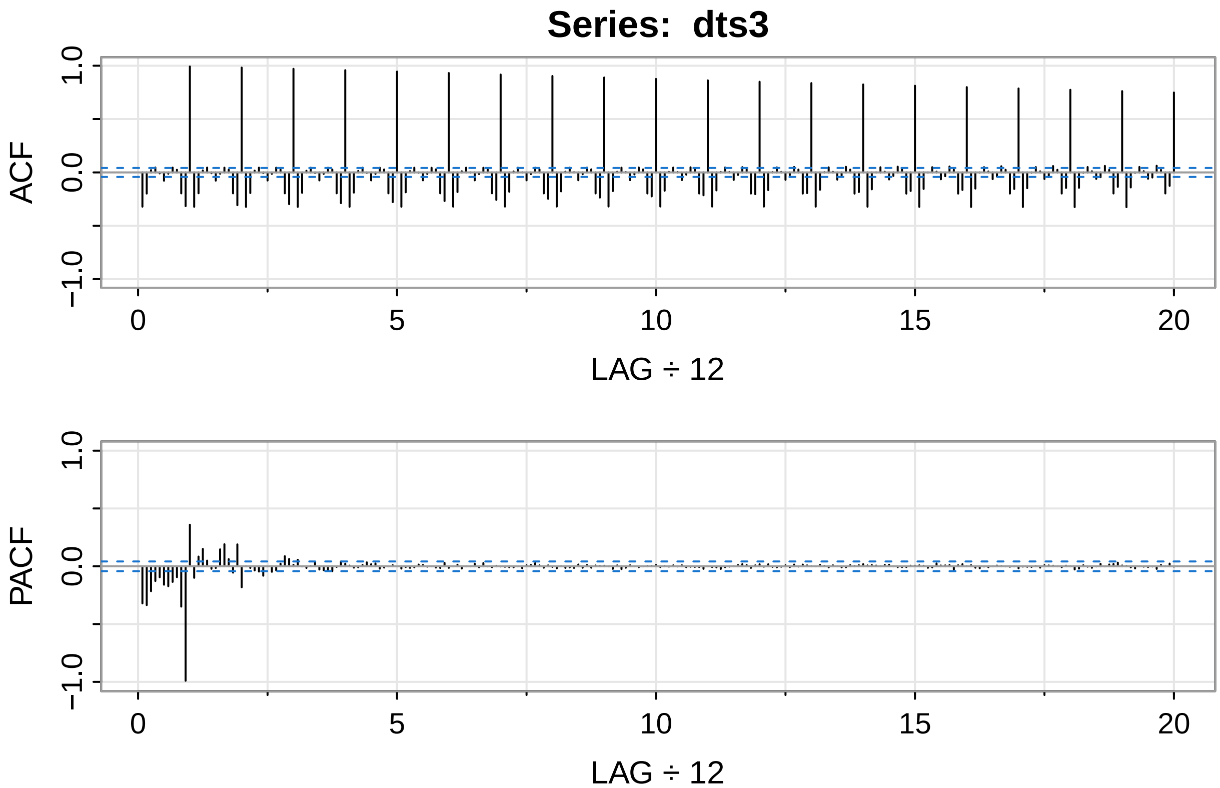

```r
acf2(dts2, max.lag = 240)
```



```
dts3 <- diff(ts2, lag = 1)
plot(dts3)
```




```
acf2(dts3, max.lag = 240)
```



The plot of the seasonally-differenced series ($D = 1$ with $s = 12$) appears to be stationary and the ACF at lags $12h$ appears to be tailing off, a marked improvement from the very slow decaying speed at lag $12h$ in the original series as shown in part (a).

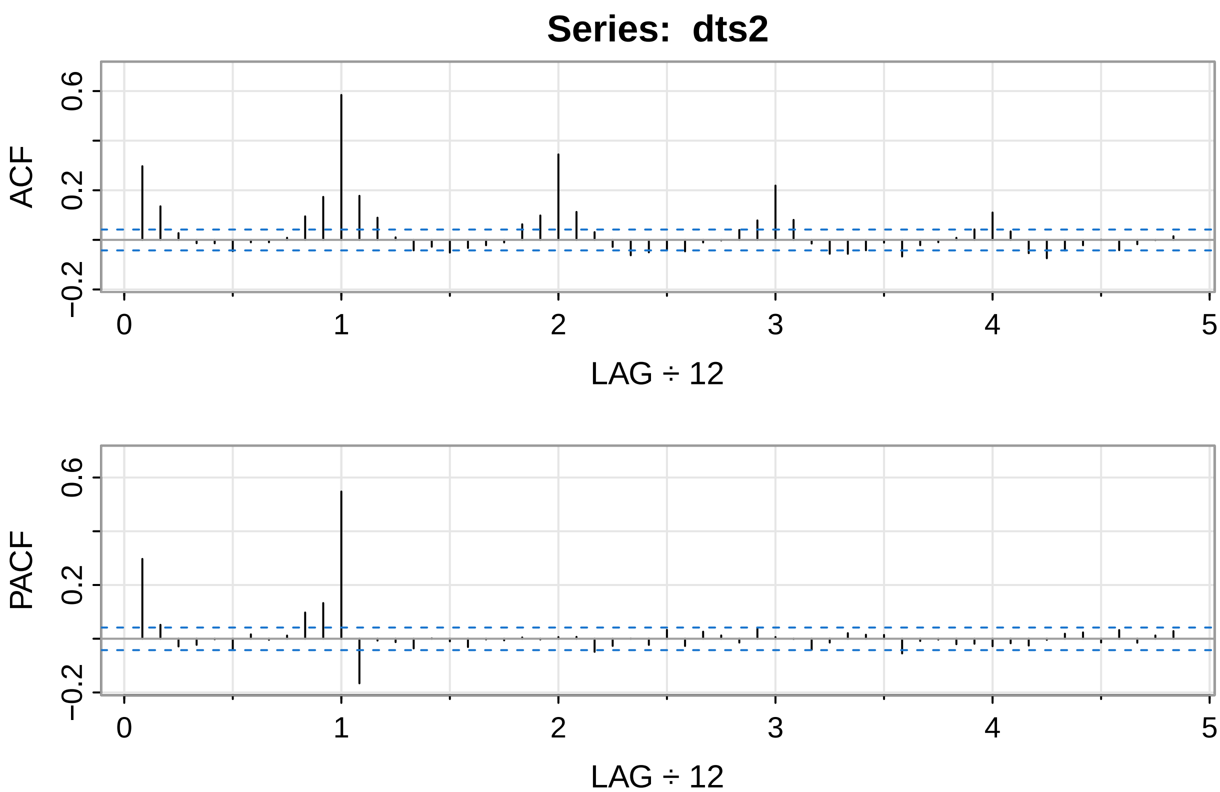
For the case of $d = 1$, the non-stationarity problem due to slow decaying of ACF remains and the plot of the non-seasonal-differenced series are similar to that of the original series.

Hence we select $d = 0, D = 1, s = 12$ to produce the desired stationary manipulated series.

(d) Using the ACF and PACF plots of the chosen manipulated series from part (c), identify at least three candidate time series models and discuss the rationales for these choices. (3 marks)

Zooming in the plot for a lag of up to 5 years

```
acf2(dts2)
```



The PACF cuts off at lag 12 and the ACF tails off at lag $12h$, indicating a $sAR(1)_{12}$ model with $P = 1, Q = 0, s = 12$.

At non-seasonal lags, the ACF appear to tail off on either side of the seasonal lags and the ACF appears to cut off at a lag of 1 on either side of lags 0 and 12, suggesting $ARMA(1, 0)$ with $p = 1, q = 0$. So the candidate model is $ARMA(1, 0) \times sAR(1)_{12}$ for the seasonally-differenced series.

We also consider another three candidate models of

- $ARMA(0, 2) \times sAR(1)_{12}$
- $ARMA(2, 0) \times sAR(1)_{12}$
- $ARMA(1, 2) \times sAR(1)_{12}$

since the patterns at the non-seasonal lags are numerically dependent on the relevant seasonal components and other time series models could lead to the same observed patterns.

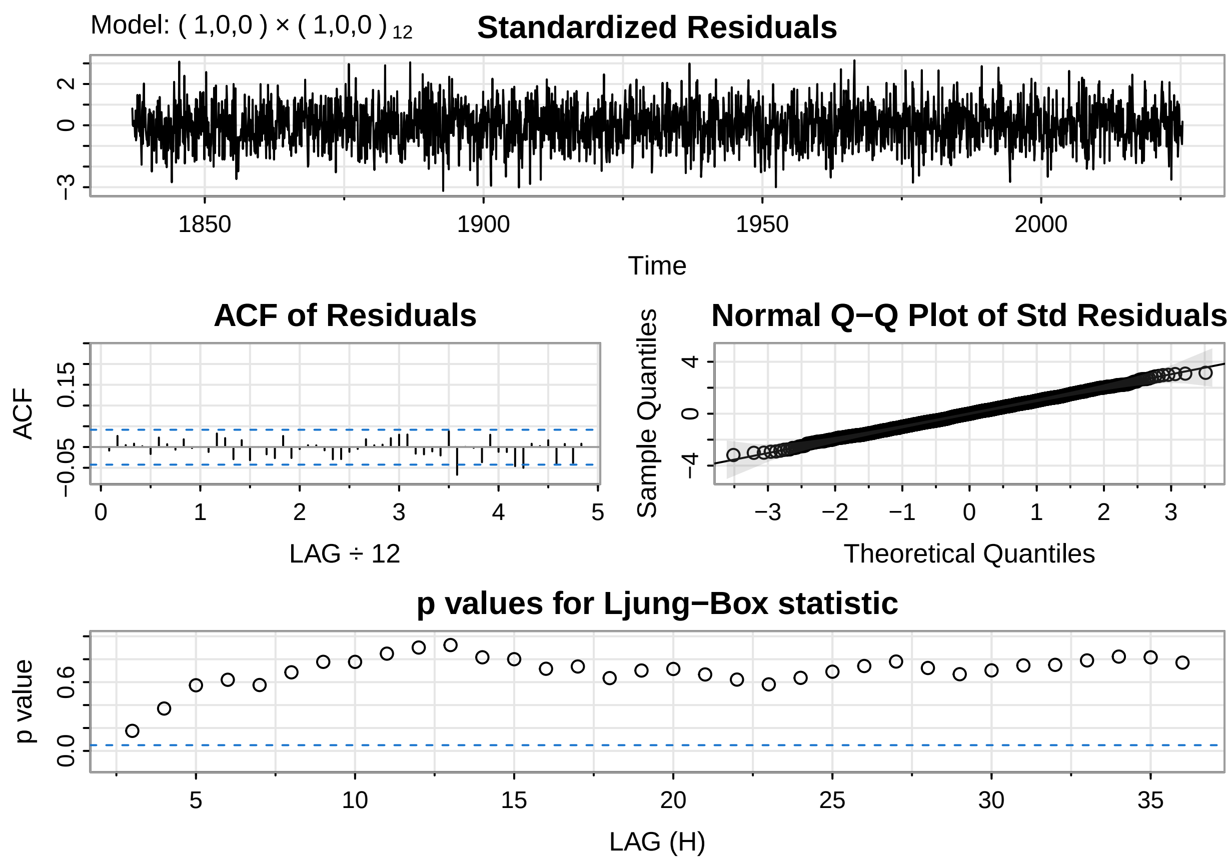
(e) By performing model estimations for the candidate models specified in part (d) and inspecting the resulting diagnostic plots, verify against your discussions in part (d) and justify the final selected model for the original series given in the dataset. Include the relevant parameters in the model and express the series x_t in the estimated model equation. (6 marks)

```
dts2.ar1 <- sarima(dts2, 1, 0, 0, 1, 0, 0, 12)
```

```
## initial  value 0.255972
```

```
## iter   2 value 0.000712
## iter   3 value 0.000707
## iter   4 value 0.000707
## iter   5 value 0.000707
## iter   6 value 0.000707
## iter   7 value 0.000707
## iter   8 value 0.000707
## iter   9 value 0.000707
## iter  10 value 0.000707
## iter  11 value 0.000707
## iter  12 value 0.000707
## iter  13 value 0.000707
## iter  14 value 0.000707
## iter  15 value 0.000706
## iter  16 value 0.000706
## iter  17 value 0.000706
## iter  17 value 0.000706
## iter  17 value 0.000706
## final  value 0.000706
## converged
## initial  value -0.000029
## iter   2 value -0.000031
## iter   3 value -0.000031
## iter   4 value -0.000031
## iter   5 value -0.000032
## iter   6 value -0.000032
## iter   7 value -0.000032
## iter   8 value -0.000032
## iter   9 value -0.000032
## iter  10 value -0.000032
## iter  11 value -0.000032
## iter  12 value -0.000032
## iter  13 value -0.000032
## iter  14 value -0.000032
## iter  15 value -0.000032
## iter  16 value -0.000032
## iter  17 value -0.000032
## iter  17 value -0.000032
## iter  17 value -0.000032
## final  value -0.000032
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
```

```
## ar1      0.2940 0.0201 14.6344  0.0000
## sar1     0.5830 0.0170 34.2889  0.0000
## xmean  -0.0018 0.0708 -0.0261  0.9792
##
## sigma^2 estimated as 0.9976934 on 2258 degrees of freedom
##
## AIC = 2.841351  AICc = 2.841356  BIC = 2.851477
##
```

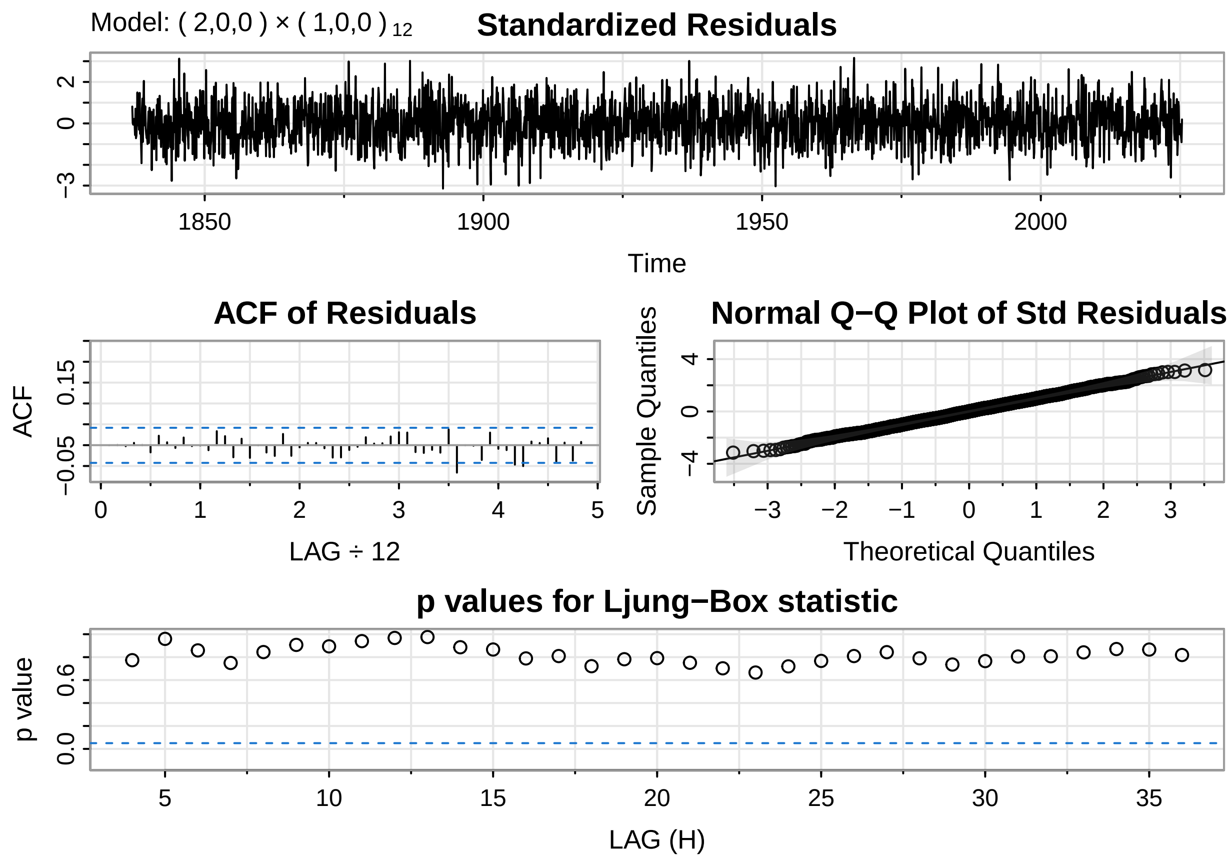


Both AR and sAR coefficients are significant. The ACF of residuals and the p-values of Ljung-Box statistic suggest white noise residuals. The constant term is not significant.

```
dts2.ar2 <- sarima(dts2, 2, 0, 0, 1, 0, 0, 12)
```

```
## initial  value 0.256184
## iter   2 value 0.007167
## iter   3 value 0.000902
## iter   4 value 0.000527
## iter   5 value 0.000502
## iter   6 value 0.000500
## iter   7 value 0.000500
## iter   8 value 0.000499
## iter   9 value 0.000499
```

```
## iter  10 value 0.000499
## iter  11 value 0.000499
## iter  12 value 0.000499
## iter  13 value 0.000499
## iter  14 value 0.000499
## iter  15 value 0.000499
## iter  16 value 0.000499
## iter  17 value 0.000499
## iter  18 value 0.000499
## iter  19 value 0.000499
## iter  20 value 0.000499
## iter  21 value 0.000499
## iter  21 value 0.000499
## iter  21 value 0.000499
## final  value 0.000499
## converged
## initial  value -0.000443
## iter   2 value -0.000445
## iter   3 value -0.000445
## iter   4 value -0.000445
## iter   5 value -0.000446
## iter   6 value -0.000446
## iter   7 value -0.000446
## iter   8 value -0.000446
## iter   9 value -0.000446
## iter  10 value -0.000446
## iter  10 value -0.000446
## final  value -0.000446
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1     0.2855 0.0210 13.5896  0.0000
## ar2     0.0288 0.0210  1.3687  0.1712
## sar1    0.5823 0.0170 34.2049  0.0000
## xmean  -0.0018 0.0728 -0.0254  0.9797
##
## sigma^2 estimated as 0.9968736 on 2257 degrees of freedom
##
## AIC = 2.841407  AICc = 2.841415  BIC = 2.854064
##
```

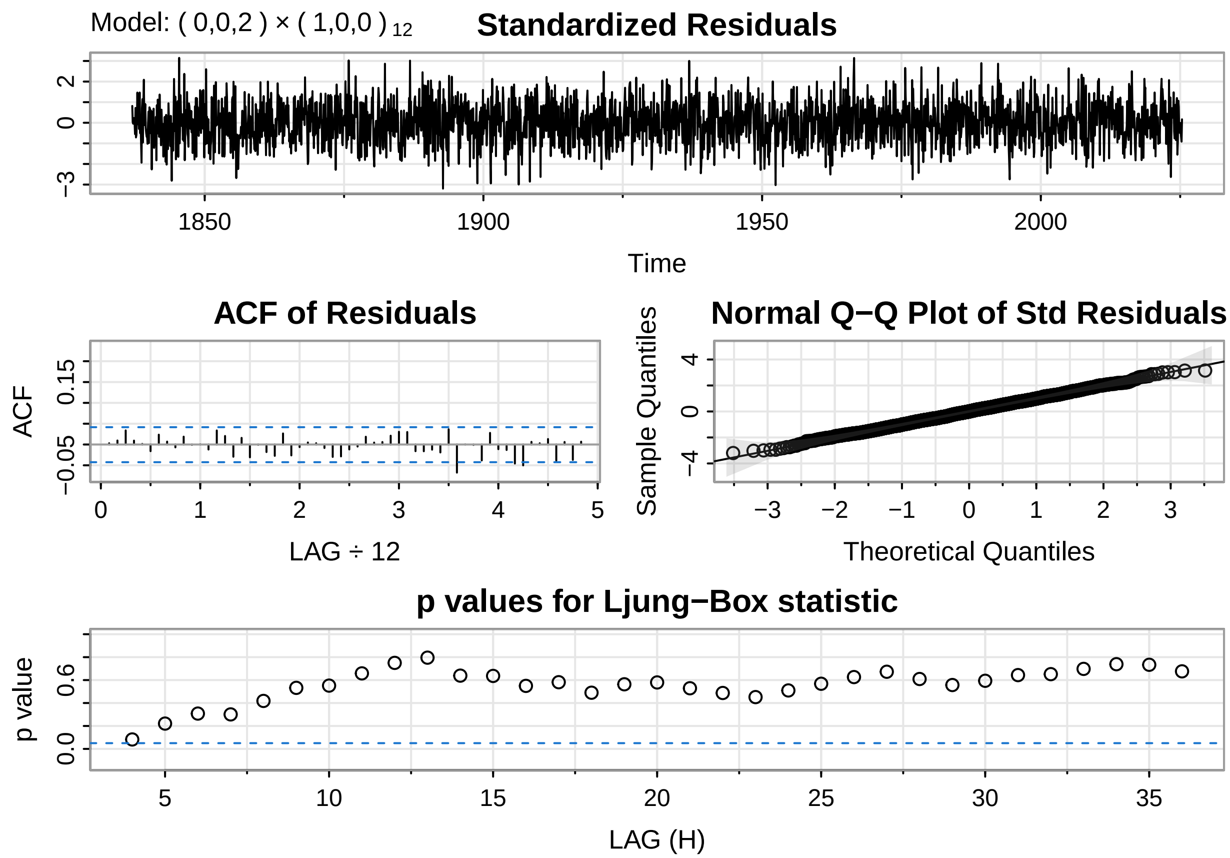


Compared to AR(1) model, the second AR coefficient is not significant, so we drop this model.

```
dts2.ma2 <- sarima(dts2, 0, 0, 2, 1, 0, 0, 12)
```

```
## initial  value 0.255889
## iter   2 value 0.001432
## iter   3 value 0.000785
## iter   4 value 0.000748
## iter   5 value 0.000747
## iter   6 value 0.000747
## iter   7 value 0.000747
## iter   8 value 0.000747
## iter   9 value 0.000747
## iter  10 value 0.000747
## iter  11 value 0.000747
## iter  12 value 0.000747
## iter  13 value 0.000747
## iter  14 value 0.000747
## iter  15 value 0.000747
## iter  16 value 0.000747
## iter  17 value 0.000747
```

```
## iter  18 value 0.000747
## iter  19 value 0.000747
## iter  20 value 0.000747
## iter  20 value 0.000747
## iter  20 value 0.000747
## final  value 0.000747
## converged
## initial  value 0.000191
## iter   2 value 0.000188
## iter   3 value 0.000188
## iter   4 value 0.000188
## iter   5 value 0.000188
## iter   6 value 0.000188
## iter   7 value 0.000188
## iter   8 value 0.000187
## iter   9 value 0.000187
## iter  10 value 0.000187
## iter  11 value 0.000187
## iter  12 value 0.000187
## iter  13 value 0.000187
## iter  14 value 0.000187
## iter  15 value 0.000187
## iter  16 value 0.000187
## iter  17 value 0.000187
## iter  18 value 0.000187
## iter  19 value 0.000187
## iter  20 value 0.000187
## iter  20 value 0.000187
## iter  20 value 0.000187
## final  value 0.000187
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ma1     0.2832 0.0208 13.5951  0.0000
## ma2     0.1005 0.0205  4.9022  0.0000
## sar1    0.5820 0.0170 34.1591  0.0000
## xmean  -0.0018 0.0690 -0.0266  0.9788
##
## sigma^2 estimated as 0.9981416 on 2257 degrees of freedom
##
## AIC = 2.842674  AICc = 2.842682  BIC = 2.855331
##
```

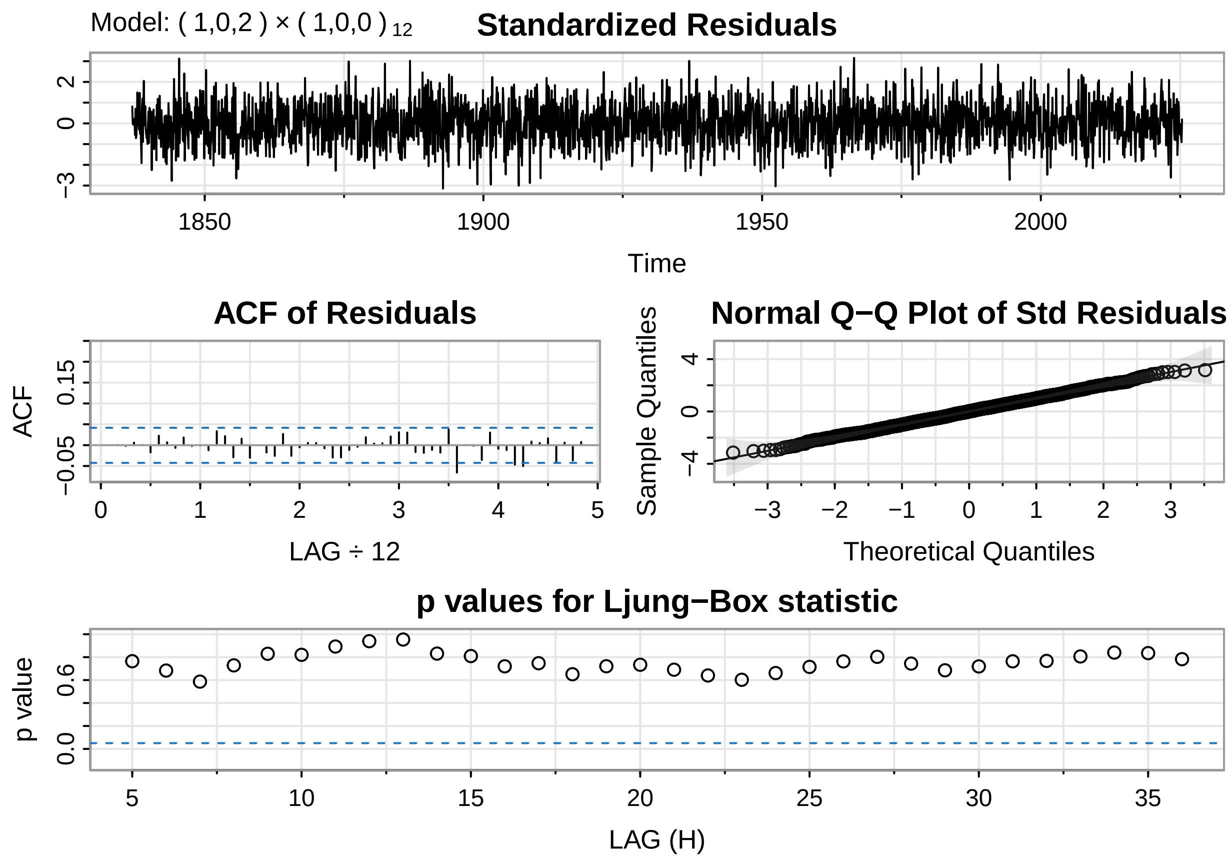


The MA and sAR coefficients are significant. The ACF of residuals looks good but one of the p-values of the Ljung-Box statistic do not suggest white noise residuals. The constant term is not significant.

```
dts2.ar1ma2 <- sarima(dts2, 1, 0, 2, 1, 0, 0, 12)
```

```
## initial  value 0.255972
## iter   2 value 0.047151
## iter   3 value 0.005813
## iter   4 value 0.001753
## iter   5 value 0.000579
## iter   6 value 0.000561
## iter   7 value 0.000559
## iter   8 value 0.000527
## iter   9 value 0.000476
## iter  10 value 0.000388
## iter  11 value 0.000318
## iter  12 value 0.000282
## iter  13 value 0.000280
## iter  14 value 0.000279
## iter  15 value 0.000279
## iter  16 value 0.000279
```

```
## iter  17 value 0.000279
## iter  18 value 0.000279
## iter  19 value 0.000279
## iter  20 value 0.000279
## iter  21 value 0.000279
## iter  22 value 0.000279
## iter  23 value 0.000279
## iter  24 value 0.000279
## iter  25 value 0.000279
## iter  26 value 0.000279
## iter  27 value 0.000279
## iter  28 value 0.000279
## iter  28 value 0.000279
## iter  28 value 0.000279
## final  value 0.000279
## converged
## initial  value -0.000442
## iter   2 value -0.000444
## iter   3 value -0.000444
## iter   4 value -0.000444
## iter   5 value -0.000444
## iter   6 value -0.000445
## iter   7 value -0.000445
## iter   8 value -0.000445
## iter   9 value -0.000445
## iter  10 value -0.000445
## iter  11 value -0.000445
## iter  12 value -0.000445
## iter  12 value -0.000445
## final  value -0.000445
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1     0.3555 0.1786  1.9901  0.0467
## ma1    -0.0699 0.1800 -0.3885  0.6977
## ma2     0.0089 0.0563  0.1579  0.8746
## sar1    0.5823 0.0170 34.1696  0.0000
## xmean  -0.0019 0.0727 -0.0255  0.9797
##
## sigma^2 estimated as 0.9968755 on 2256 degrees of freedom
##
## AIC = 2.842294  AICc = 2.842305  BIC = 2.857482
##
```



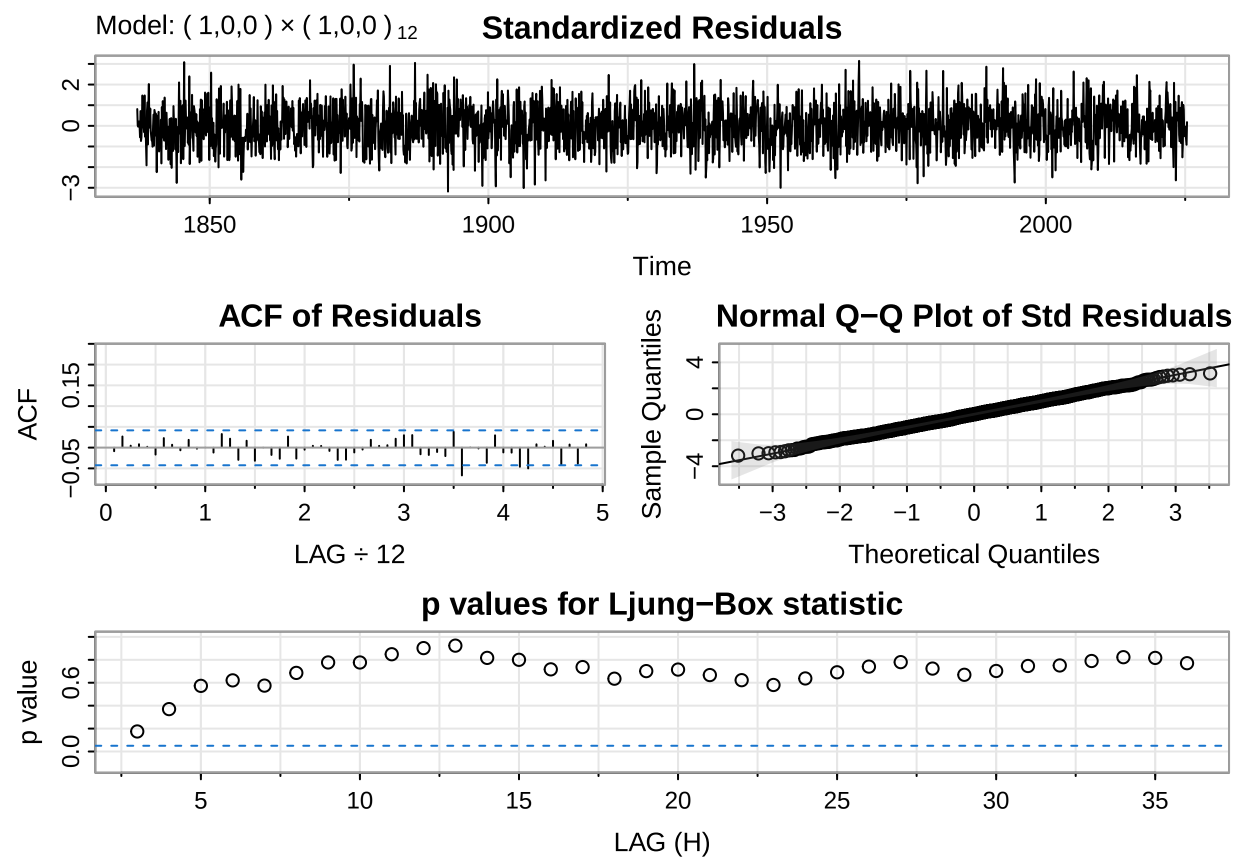
The MA coefficients θ_1 and θ_2 are not significant and the constant term is not significant, so we drop this model.

We refit the two remaining candidate models by setting `no.constant=TRUE`.

```
dts2.ar1 <- sarima(dts2, 1, 0, 0, 1, 0, 0, 12, no.constant = TRUE)
```

```
## initial  value 0.255976
## iter   2 value 0.000714
## iter   3 value 0.000708
## iter   4 value 0.000708
## iter   5 value 0.000708
## iter   6 value 0.000708
## iter   6 value 0.000708
## final  value 0.000708
## converged
## initial  value -0.000029
## iter   2 value -0.000032
## iter   3 value -0.000032
## iter   4 value -0.000032
## iter   4 value -0.000032
## iter   4 value -0.000032
## final  value -0.000032
```

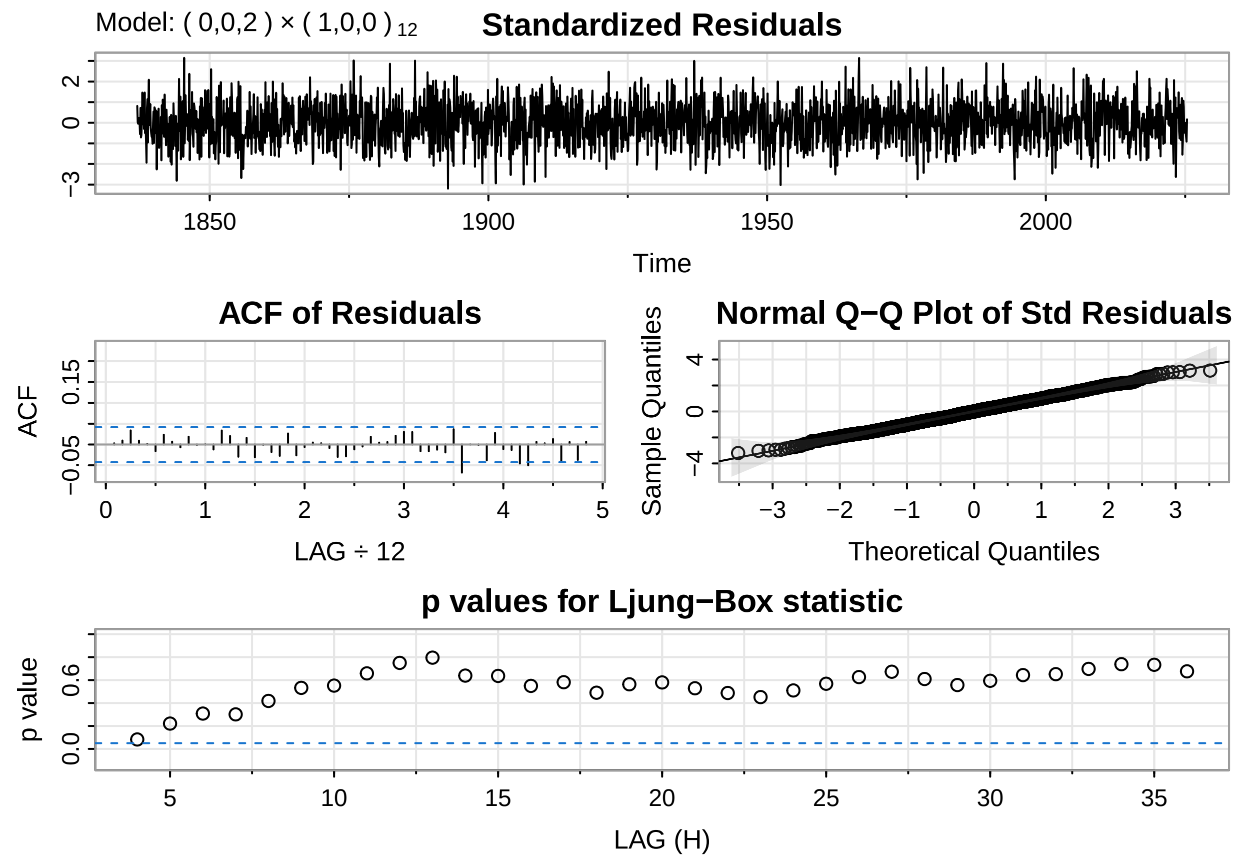
```
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate     SE t.value p.value
## ar1     0.294 0.0201 14.6345       0
## sar1    0.583 0.0170 34.2890       0
##
## sigma^2 estimated as 0.9976937 on 2259 degrees of freedom
##
## AIC = 2.840467  AICc = 2.840469  BIC = 2.848061
##
```




```
dts2.ma2 <- sarima(dts2, 0, 0, 2, 1, 0, 0, 12, no.constant = TRUE)
```

```
## initial  value 0.255893
## iter   2 value 0.001433
## iter   3 value 0.000787
## iter   4 value 0.000749
## iter   5 value 0.000749
## iter   6 value 0.000748
## iter   7 value 0.000748
```

```
## iter   7 value 0.000748
## iter   7 value 0.000748
## final  value 0.000748
## converged
## initial  value 0.000190
## iter   2 value 0.000188
## iter   3 value 0.000187
## iter   4 value 0.000187
## iter   5 value 0.000187
## iter   5 value 0.000187
## iter   5 value 0.000187
## final  value 0.000187
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate     SE t.value p.value
## ma1    0.2832 0.0208 13.5951       0
## ma2    0.1005 0.0205  4.9022       0
## sar1   0.5820 0.0170 34.1591       0
##
## sigma^2 estimated as 0.9981419 on 2258 degrees of freedom
##
## AIC = 2.84179  AICc = 2.841795  BIC = 2.851916
##
```



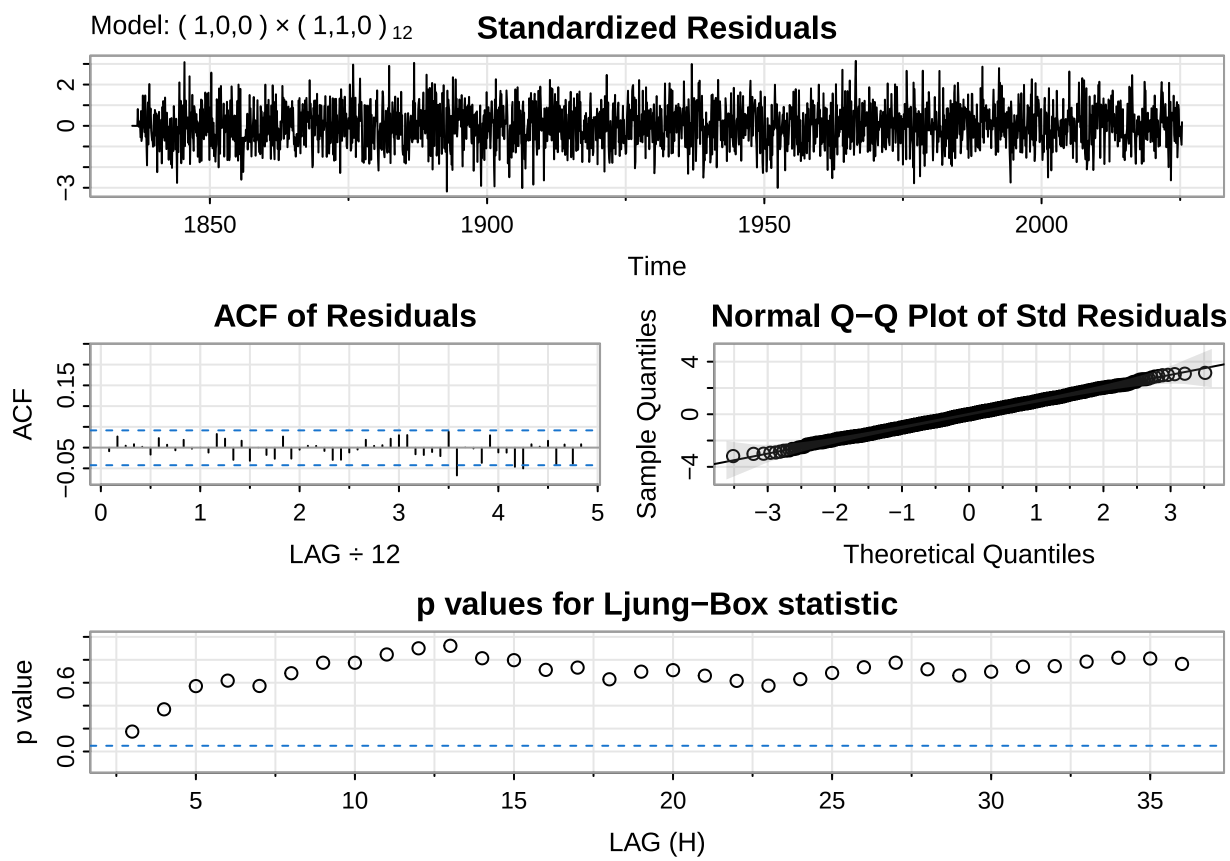
The diagnostics of both models are equally good. We select the first model $ARMA(1,0) \times sAR(1)_{12}$ because of the lower BIC, AIC and AICc in favour of model parsimony.

Note that we obtain the same fit below if we change the arguments of sarima accordingly.

```
dts2.ar1b <- sarima(ts2, 1, 0, 0, 1, 1, 0, 12, no.constant = TRUE)
```

```
## initial  value 0.255976
## iter   2 value 0.000714
## iter   3 value 0.000708
## iter   4 value 0.000708
## iter   5 value 0.000708
## iter   6 value 0.000708
## iter   6 value 0.000708
## final  value 0.000708
## converged
## initial  value -0.000029
## iter   2 value -0.000032
## iter   3 value -0.000032
## iter   4 value -0.000032
## iter   4 value -0.000032
## iter   4 value -0.000032
## final  value -0.000032
```

```
## converged
## <><><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate     SE t.value p.value
## ar1     0.294 0.0201 14.6344       0
## sar1    0.583 0.0170 34.2890       0
##
## sigma^2 estimated as 0.9976937 on 2259 degrees of freedom
##
## AIC = 2.840467  AICc = 2.840469  BIC = 2.848061
##
```



The estimated model can be expressed as

$$\begin{aligned}(1 - 0.3258B)(1 - B^{12})(1 - 0.5823B^{12})x_t &= w_t \\ (1 - 0.3258B - 0.5823B^{12} + 0.5155B^{13} + 0.5823B^{24} - 0.1897B^{25})x_t &= w_t\end{aligned}$$

$$\Rightarrow x_t = 0.3258x_{t-1} + 0.5823x_{t-12} - 0.5155x_{t-13} - 0.5823x_{t-24} + 0.1897x_{t-25} + w_t$$

with $\hat{\sigma}^2 = 0.9987$.

Question 5

Consider a time series x_t specified in the following model

$$x_t = -0.7x_{t-1} + 0.6x_{t-2} + w_t - 0.2w_{t-1}$$

where w_t is uncorrelated white noise with constant variance σ^2.

(a) Determine whether the model is invertible. (1 mark)

The MA polynomial is given by

$$\begin{aligned}\theta(B) &= 1 - 0.2B \\ \theta(z) &= 1 - 0.2z = 0 \\ z &= 5 > 1\end{aligned}$$

that is, the root is outside the unit circle, so the model is invertible.

(b) Determine whether the model is causal. (2 marks)

The AR polynomial is given by

$$\begin{aligned}\phi(B) &= 1 + 0.7B - 0.6B^2 \\ \phi(z) &= 1 + 0.7z - 0.6z^2 = (1 + 1.2z)(1 - 0.5z) = 0 \\ |z| &= |-\frac{5}{6}| < 1 \text{ and } z = 2 > 1\end{aligned}$$

that is, one of the roots is inside the unit circle, so the model is not causal.

(c) Simulate 300 values of the time series using a suitable R function and verify against your answer in part (b). (2 marks)

```
x <- arima.sim(list(order = c(2, 0, 1), ar = c(-0.7, 0.6), ma = -0.2),
    n = 300)
## Error in arima.sim(list(order = c(2, 0, 1), ar = c(-0.7, 0.6), ma = -0.2), : 'ar' p
```

```
## [1] "'ar' part of model is not stationary"
```

Consider the AR polynomial of

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2$$

The required condition for causality is that both the roots of the characteristic equation

$$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 = 0$$

denoted by z_1 and z_2 are outside the unit circle.

Suppose that we aim to determine the equivalent required condition for causality in terms of ϕ_1 and ϕ_2.

(d) Let us divide the characteristic equation by z^2 and obtain

$$\frac{1}{z^2} - \frac{\phi_1}{z} - \phi_2 = 0$$

If we define $y = \frac{1}{z}$, then the resulting equation can be expressed as

$$\alpha(y) = y^2 - \phi_1 y - \phi_2 = 0$$

Write down the equivalent required condition for causality in terms of the characteristic roots y_1 and y_2 and express ϕ_1 and ϕ_2 in terms of y_1 and y_2. (3 marks)

$$|y_1| = |\frac{1}{z_1}| < 1, \quad |y_2| = |\frac{1}{z_2}| < 1$$

$$\begin{aligned}
&\alpha(y) = (y - y_1)(y - y_2) = y^2 - (y_1 + y_2)y + y_1 y_2 = y^2 - \phi_1 y - \phi_2 = 0 \\
&\Rightarrow \phi_1 = y_1 + y_2, \quad \phi_2 = -y_1 y_2
\end{aligned}$$

(e) Following from part (d), determine the equivalent required condition for causality in terms of ϕ_1 and ϕ_2. Hint: consider (i) the **sign** of $(y_1 + a)(y_2 + b)$, (ii) the **sign** of $(y_1 + c)(y_2 + d)$ and (iii) the magnitude of $y_1 y_2$ for suitably chosen constants a, b, c and d. (3 marks)

$$\begin{aligned}
(y_1 - 1)(y_2 - 1) &= y_1 y_2 - (y_1 + y_2) + 1 > 0 \\
y_1 + y_2 - y_1 y_2 &< 1 \\
\Rightarrow \phi_1 + \phi_2 &< 1 \\
(y_1 + 1)(y_2 + 1) &= y_1 y_2 + (y_1 + y_2) + 1 > 0 \\
-(y_1 + y_2) - y_1 y_2 &< 1 \\
\Rightarrow -\phi_1 + \phi_2 &< 1 \\
|y_1 y_2| &< 1 \\
\Rightarrow |\phi_2| < 1
\end{aligned}$$

(f) Following from part (e), determine whether the model given above

$$x_t = -0.7x_{t-1} + 0.6x_{t-2} + w_t - 0.2w_{t-1}$$

is causal without finding the characteristic roots. (1 mark)

We have $\phi_1 = -0.7, \phi_2 = 0.6$ and so $|\phi_2| < 1, \phi_2 + \phi_1 = -0.1 < 1$ but $\phi_2 - \phi_1 = 1.3 > 1$, so the model is not causal.

Question 6

Consider the following two time series:

$$x_t = w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2},$$

and

$$y_t = w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2},$$

where w_t is white noise with variance σ_w^2.

Furthermore, we have that

$$\begin{aligned} \alpha_1 &= 0.8465; \\ \alpha_2 &= 0.2476; \\ \beta_1 &= 0.8028; \\ \beta_2 &= 0.2498; \\ \sigma_w^2 &= 1.3647. \end{aligned}$$

Explain whether each of the following statements is true or false.

(a) Both series are stationary and causal. (1 mark)

(b) The variance of y_t is 2.3294. (2 marks)

(c) The autocorrelation of x_t at lag 1 is 0.6152. (2 marks)

(d) The cross-covariance $\gamma_{xy}(t, t+1) = 1.4265$. (2 marks)

(a) True - because both are pure MA model with an AR polynomial of $\phi(B) = 1$.

(b) True - because we have that $\gamma_y(t,t) = cov(y_t, y_t) = cov(w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2}, w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2}) = \sigma_w^2 (1 + \beta_1^2 + \beta_2^2)$

(c) False - because we have $\gamma_x(t,t) = cov(x_t, x_t) = cov(w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}, w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}) = \sigma_w^2 (1 + \alpha_1^2 + \alpha_2^2)$ and $\gamma_x(t+1, t) = Cov(x_{t+1}, x_t) = Cov(w_{t+1} + \alpha_1 w_t + \alpha_2 w_{t-1}, w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}) = (\alpha_1 + \alpha_2 \alpha_1)\sigma_w^2 \Rightarrow \rho_x(1) = \frac{\gamma_x(t,t+1)}{\gamma_x(t,t)} = 0.594$

(d) False - because we have that $\gamma_{xy}(t, t+1) = Cov(x_t, y_{t+1}) = Cov(w_t + \alpha_1 w_{t-1} + \alpha_2 w_{t-2}, w_{t+1} + \beta_1 w_t + \beta_2 w_{t-1}) = \sigma_w^2 (\beta_1 + \alpha_1 \beta_2) = 1.3841$