

Lab 6

Abdulla Almansoori, Alex Firth, Nathan Seto

2025-05-07

Ex 1

Figure 1 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.

- a. Explain the differences among these figures. Do they all indicate that the data are white noise?

As sample size increases, the variation decreases, and the autocorrelation value decreases. All these figures appear to be white noise.

- b. Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?

The values differ in distance because of the smaller variance and larger sample size. The expected value of white noise is zero, so the critical values will continue to approach zero as sample increases.

Ex 2

- a. Produce a time plot of the sheep population of England and Wales from 1867–1939 (data set **sheep**). Use the function `ndiffs()` to see how many times the time series needs to be differentiated to be stationary.

```
data(sheep)
ts.plot(sheep, main = "Sheep Time Series") # time-series plot
```

Sheep Time Series



```
ndiffs(sheep) # how many times the time series needs to be differentiated to become stationary
```

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

The time series needs to be differentiated one time to be stationary.

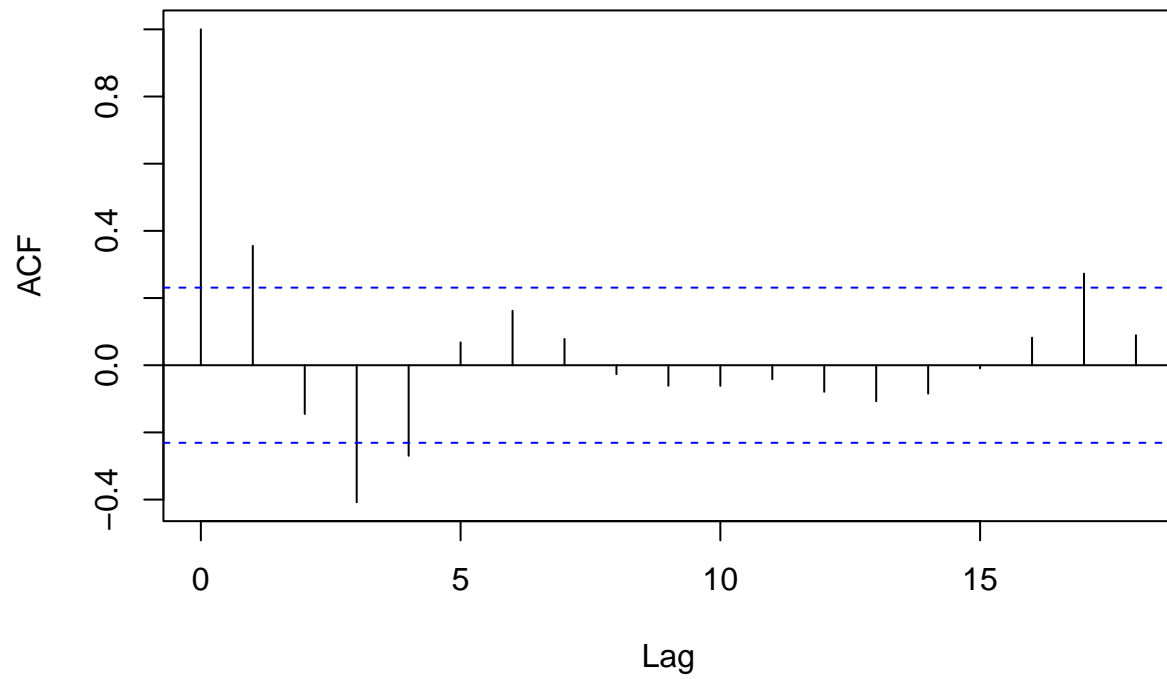
- b. Assume you decide to fit the following model: $y_t = y_{t-1} + \phi_1(y_{t-1} - y_{t-2}) + \phi_2(y_{t-2} - y_{t-3}) + \phi_3(y_{t-3} - y_{t-4}) + \epsilon_t$, where ϵ_t is a white noise series. What sort of ARIMA model is this (i.e., what are p, d, and q)?

This is an ARIMA(3, 1, 0) model.

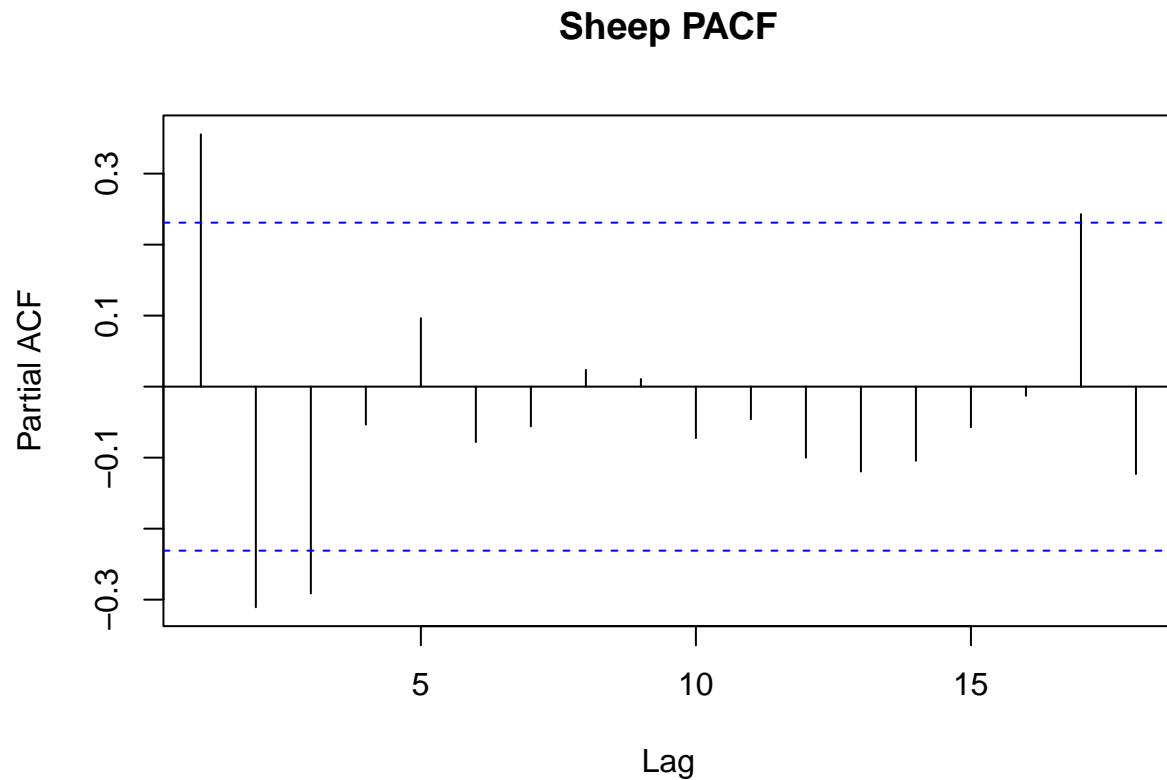
- c. By examining the ACF and PACF of the differenced data, explain why this model is appropriate.

```
sheep_diff <- diff(sheep) # differencing the series 1 time  
acf(sheep_diff, main = "Sheep ACF") # autocorrelation plot
```

Sheep ACF



```
pacf(sheep_diff, main = "Sheep PACF") # partial autocorrelation plot
```



The ACF yields an order of 3, and the data needs to be differenced one time.

```
## Series: sheep
## ARIMA(3,1,0)
##
## Coefficients:
##      ar1      ar2      ar3
##      0.421  -0.202  -0.304
## s.e.  0.119   0.136   0.124
##
## sigma^2 = 4991:  log likelihood = -407.56
## AIC=823.12  AICc=823.71  BIC=832.22
```

d. The last five values of the series are given below:

Year	1935	1936	1937	1938	1939
Millions of sheep	1648	1665	1627	1791	1797

The estimated parameters are $\phi_1=0.421$, $\phi_2=-0.202$, and $\phi_3=-0.304$. Without using the forecast function, calculate forecasts for the next three years (1940–1942).

```
# function to predict sheep
predictSheep <- function(y1, y2, y3, y4) {
  # calculate predictions with the provided coefficients
  c_pred <- y1 + ((y1 - y2) * (0.421)) + ((y2 - y3) * (-0.202)) + ((y3 - y4) *
    (-0.304))
}
```

```

    return(c_pred)
}

vec <- sheep[69:73]

# store predictions
c1 <- predictSheep(vec[5], vec[4], vec[3], vec[2])
c2 <- predictSheep(c1, vec[5], vec[4], vec[3])
c3 <- predictSheep(c2, c1, vec[5], vec[4])

# print results
cat("Predicted Number of Sheep in 1940:", round(c1), "\n")

## Predicted Number of Sheep in 1940: 1778
cat("Predicted Number of Sheep in 1941:", round(c2), "\n")

## Predicted Number of Sheep in 1941: 1719
cat("Predicted Number of Sheep in 1942:", round(c3), "\n")

## Predicted Number of Sheep in 1942: 1696

```

- e. Now fit the model in R and obtain the forecasts using the function `forecast()`. Are they different from yours?

```

forecast(fit, h = 3)

##      Point Forecast  Lo 80  Hi 80  Lo 95  Hi 95
## 1940          1778.0 1687.5 1868.5 1639.5 1916.5
## 1941          1718.9 1561.5 1876.2 1478.3 1959.5
## 1942          1696.0 1494.2 1897.8 1387.3 2004.7

```

They are the same.

Ex 3

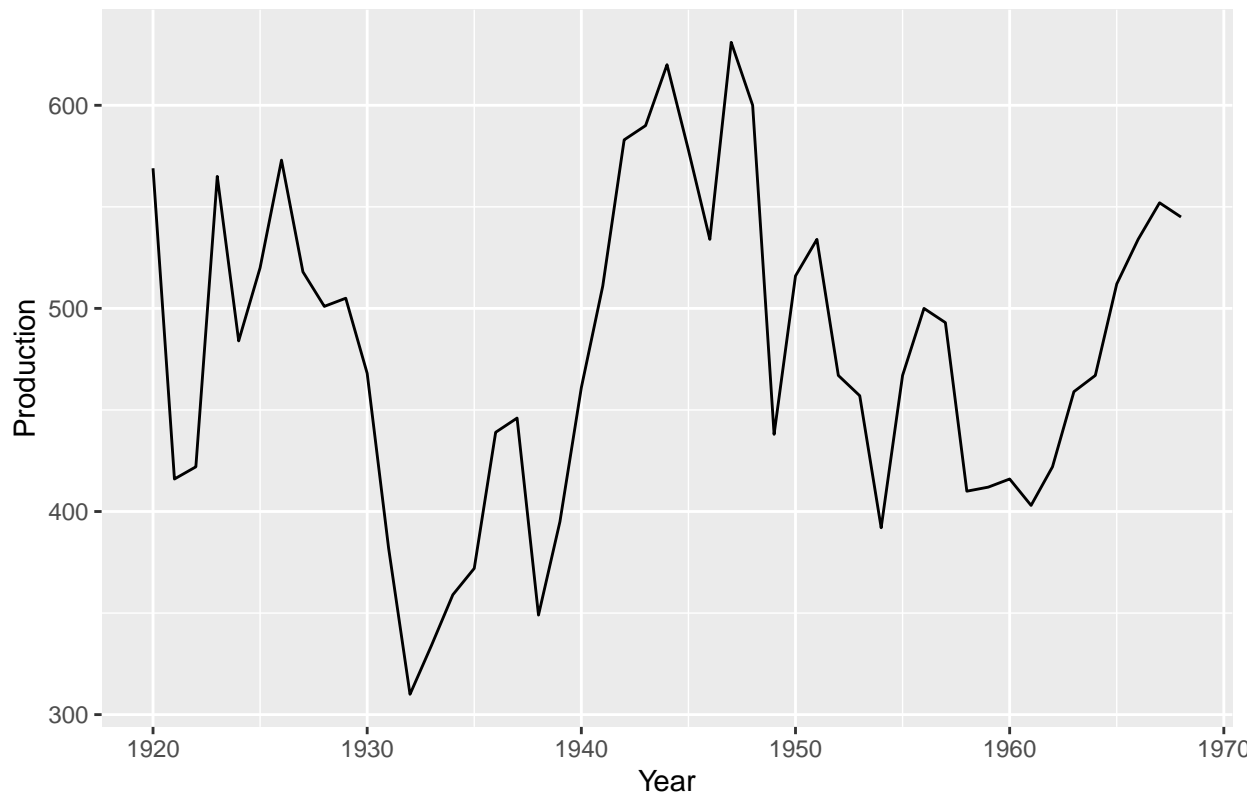
- a. Plot the annual bituminous coal production in the United States from 1920 to 1968 (data set `bicoal`). Use the function `ndiffs()` to see how many times the time series needs to be differentiated to be stationary.

```

autoplot(bicoal) + ggtitle("Annual Bituminous Coal Production in the U.S. (1920-1968)") +
  xlab("Year") + ylab("Production")

```

Annual Bituminous Coal Production in the U.S. (1920–1968)



```
dneeded <- ndiffs(bicoal)
print(dneeded)
```

```
## [1] 0
```

- b. You decide to fit the following model to the series: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \phi_4 y_{t-4} + \epsilon_t$ where y_t is the coal production in year t and ϵ_t is a white noise series. What sort of ARIMA model is this (i.e., what are p , d , and q)?

This is an ARIMA(4, 0, 0)/ar(4) model because it uses autoregressive terms up to 4 lags
 $p=4$ $d=0$ $q=0$

- c. Explain why this model was chosen using the ACF and PACF.

The PACF shows 4 significant lags before cutting off, while the ACF decayed gradually, indicating an autoregressive process rather than a moving average or mixed model.

- d.) Fit the model to the data using the Arima function. Note that you will need to set the parameter `include.constant=TRUE`. Check that $c=162.00$, $\phi_1=0.83$, $\phi_2=-0.34$, $\phi_3=0.55$, and $\phi_4=-0.38$. Note that the Arima function doesn't provide you with c directly. It provides you with the mean of the time series. You can calculate c by running: $c <- \mu * (1-\phi_1-\phi_2-\phi_3-\phi_4)$

```
## Series: bicoal
## ARIMA(4,0,0) with non-zero mean
##
## Coefficients:
##      ar1      ar2      ar3      ar4      mean
##      0.833 -0.344  0.552 -0.378  481.522
## s.e.  0.137   0.175  0.173   0.141   21.059
```

```
##
## sigma^2 = 2795: log likelihood = -262.05
## AIC=536.1 AICc=538.1 BIC=547.45

## intercept
## 162

## ar1
## 0.83

## ar2
## -0.34

## ar3
## 0.55

## ar4
## -0.38
```

e. The last five values of the series are given below.

Year	1964	1965	1966	1967	1968
Millions of tons	467	512	534	552	545

Without using the `forecast` function, calculate forecasts for the next three years (1969–1971) using R.

```
y <- as.numeric(bicoal)

# Manual forecast function
forecast_ar4 <- function(y, c, phi1, phi2, phi3, phi4, h) {
  forecasts <- numeric(h)
  for (t in 1:h) {
    forecasts[t] <- c + phi1 * y[length(y)] + phi2 * y[length(y) - 1] + phi3 *
      y[length(y) - 2] + phi4 * y[length(y) - 3]
    y <- c(y, forecasts[t]) # Append forecast to extend the series
  }
  return(forecasts)
}

# Generate forecasts
forecasts <- forecast_ar4(y, c, phi1, phi2, phi3, phi4, h = 3)
names(forecasts) <- 1969:1971
print(round(forecasts, 4))

## 1969 1970 1971
## 527.63 517.19 503.81
```

f. Now fit the model in R and obtain the forecasts from the same model. How are they different from yours? Why?

```
auto_forecasts <- forecast(fit, h = 3)
print(auto_forecasts)

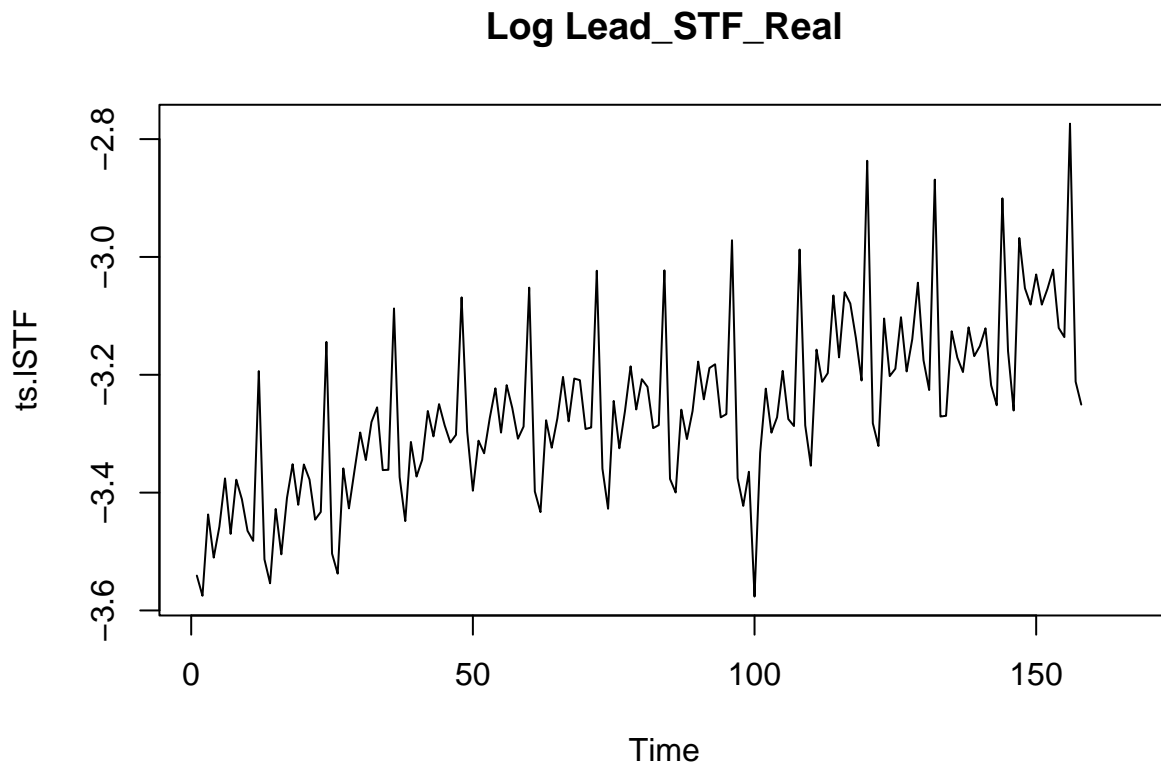
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 1969 527.63 459.88 595.38 424.02 631.24
## 1970 517.19 429.00 605.38 382.32 652.07
## 1971 503.81 412.48 595.13 364.13 643.48
```

They are the same as our forecasts as they use the same coefficients to forecast the data.

Extracredit

Load FinalData.xlsx and create a time series $ts.lSTF = \log(STF_Real)$. Plot $ts.lSTF$.

```
FinalData <- read_excel("~/ECON436_SalesTax/Data/FinalData.xlsx")
ts.lSTF <- ts(log(FinalData$Lead_STF_Real))
ts.plot(ts.lSTF, main = "Log Lead_STF_Real")
```



- a. Fit a seasonal ARIMA model using the option `seasonal=TRUE`. What model is identified as the best fit?

```
stf_fit <- auto.arima(FinalData$Lead_STF_Real, seasonal = TRUE)
stf_fit
```

```
## Series: FinalData$Lead_STF_Real
## ARIMA(2,1,2) with drift
##
## Coefficients:
##          ar1      ar2      ma1      ma2  drift
##        -0.615  -0.405  -0.416  -0.410     0
## s.e.    0.126   0.088   0.123   0.124     0
##
## sigma^2 = 1.85e-05: log likelihood = 634.05
## AIC=-1256.1   AICc=-1255.5   BIC=-1237.8
```


An ARIMA(2, 1, 2) model is the best fit.

- b. Predict the value of $\log(\text{STF_Real})$ for the first 11 months of 2025. Exponentiate the variables x, mean, lower, and upper and plot FC sales tax + the predictions for 2025.

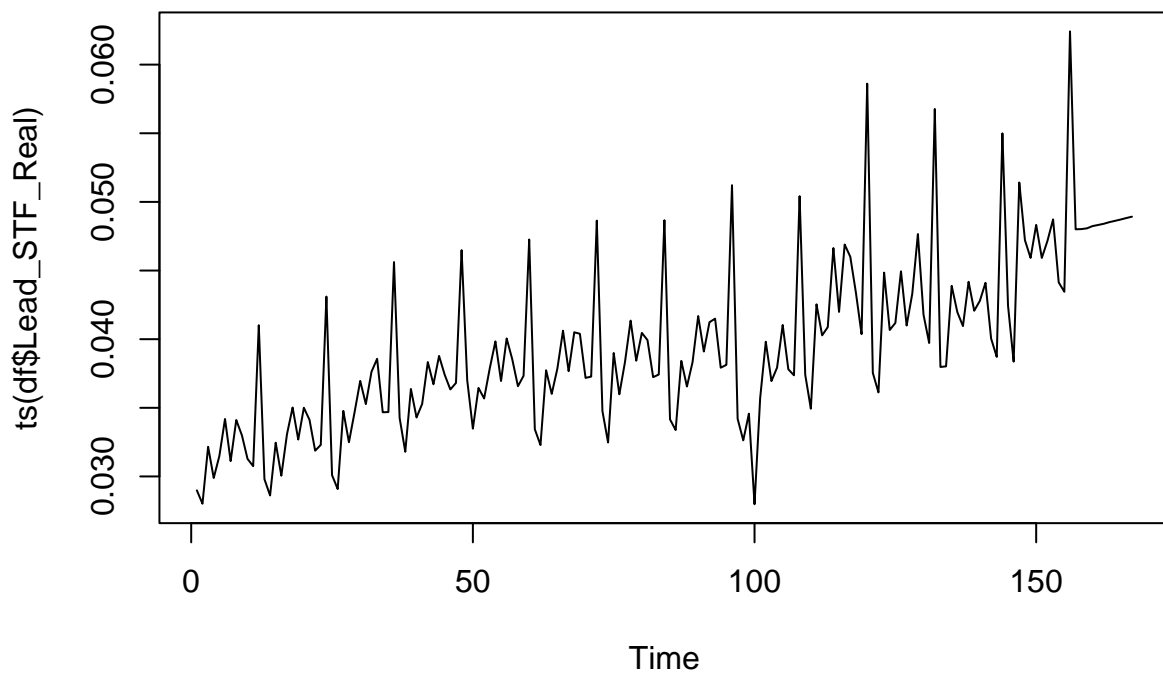
```
forecast <- forecast(stf_fit, h = 11)
forecast$mean <- exp(forecast$mean)
forecast$lower <- exp(forecast$lower)
forecast$upper <- exp(forecast$upper)

as.data.frame(forecast$mean)
```

```
##           x
## 1  1.0492
## 2  1.0492
## 3  1.0492
## 4  1.0494
## 5  1.0495
## 6  1.0496
## 7  1.0497
## 8  1.0498
## 9  1.0499
## 10 1.0500
## 11 1.0501
```

```
df <- FinalData |>
  select(Month, year, Lead_STF_Real)
df$year[is.na(df$year)] <- 2025
rows_2025 <- which(df$year == 2025)[1:11]
df$Lead_STF_Real[rows_2025] <- log(forecast$mean)

ts.plot(ts(df$Lead_STF_Real))
```



- c. They are a little different than the values presented in the final project, but they also have not been manipulated the same.