HW5

Farooq Mahmud

Problem 1

```r
library(forecast)
```

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

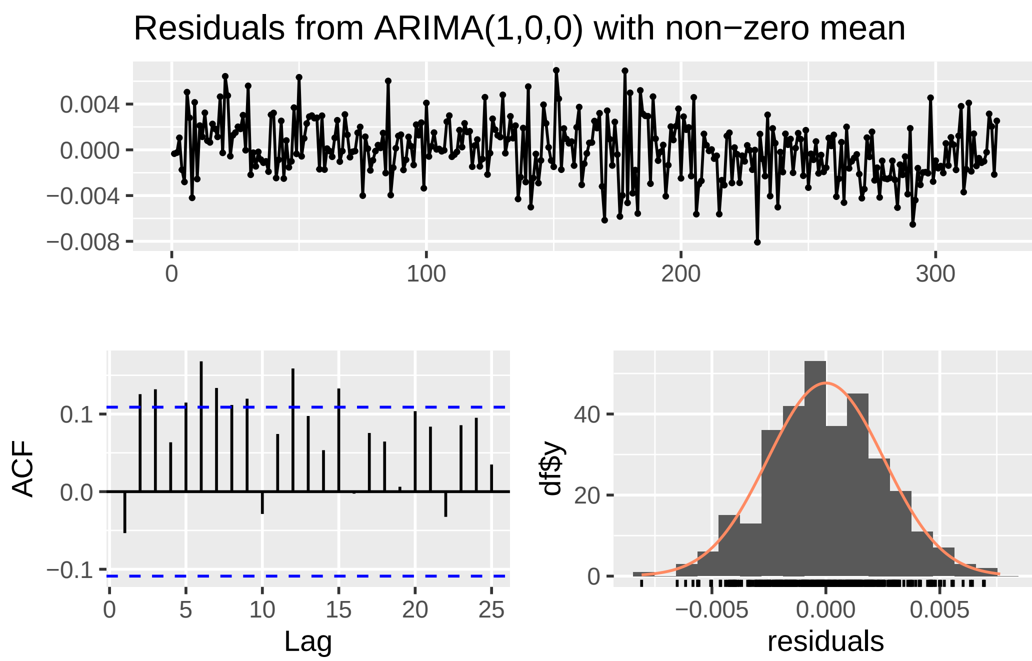
```r
library(TSA)
```

```
Registered S3 methods overwritten by 'TSA':
  method       from
  fitted.Arima forecast
  plot.Arima   forecast


Attaching package: 'TSA'


The following objects are masked from 'package:stats':

    acf, arima


The following object is masked from 'package:utils':

    tar
```

```r
library(ggplot2)
data(robot)

model_ar1 <- Arima(robot, order = c(1, 0, 0))
model_arima011 <- Arima(robot, order = c(0, 1, 1))

model_comparison <- data.frame(
  Model = c("AR(1)", "ARIMA(0,1,1)"),
  AIC = c(AIC(model_ar1), AIC(model_arima011)),
  BIC = c(BIC(model_ar1), BIC(model_arima011))
)

print(model_comparison)
```

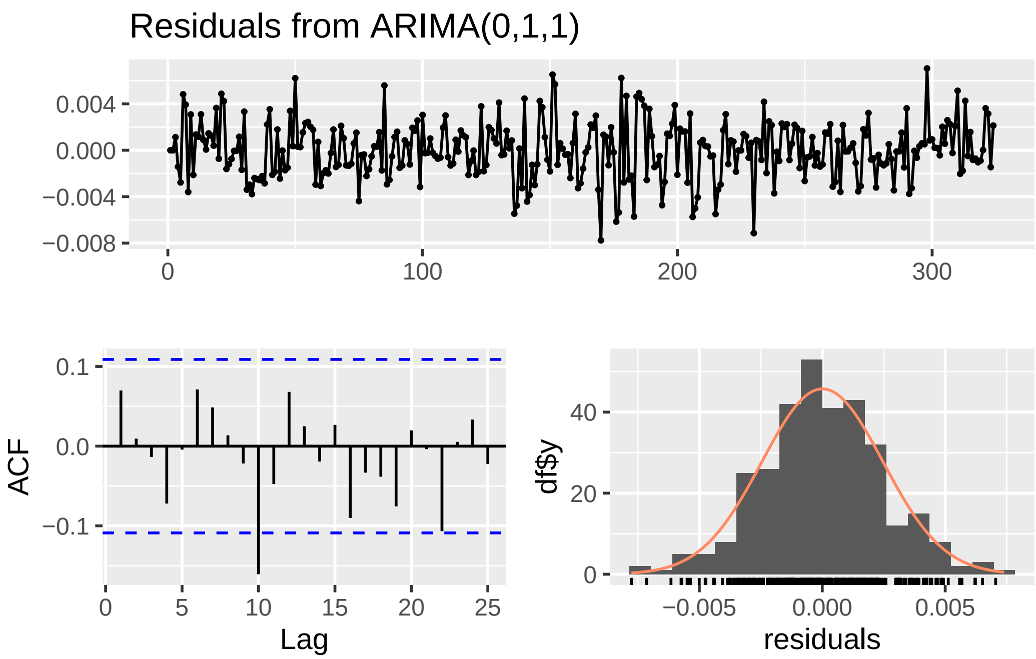
```
         Model       AIC       BIC
1        AR(1) -2945.078 -2933.735
2 ARIMA(0,1,1) -2957.901 -2950.346
```

```r
checkresiduals(model_ar1)
```



```
    Ljung-Box test

data:  Residuals from ARIMA(1,0,0) with non-zero mean
Q* = 42.118, df = 9, p-value = 3.127e-06

Model df: 1.   Total lags used: 10
```

```r
checkresiduals(model_arima011)
```

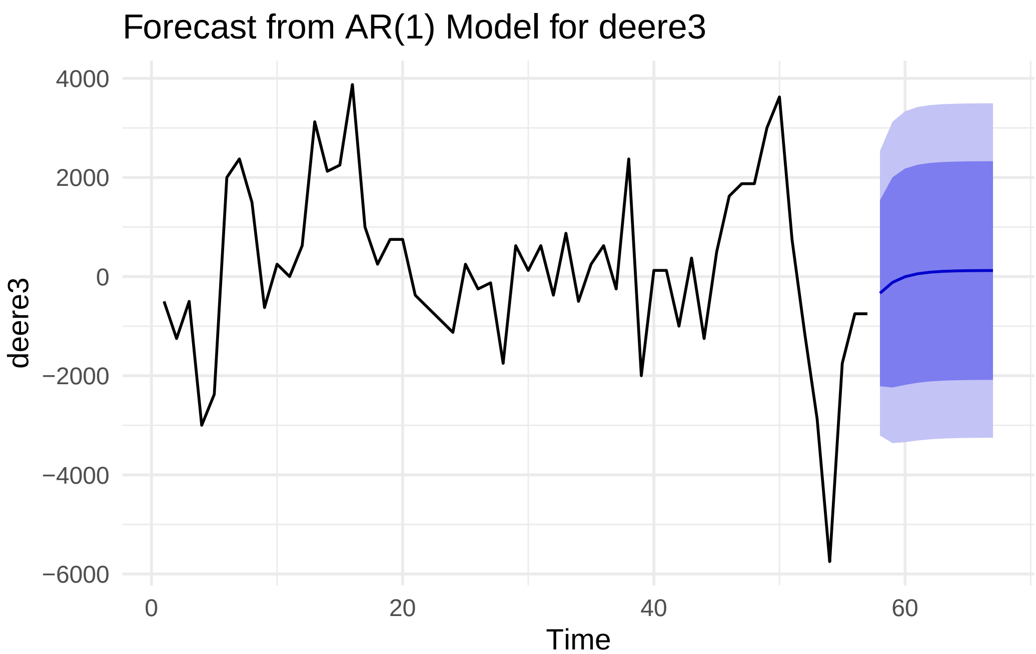



```
    Ljung-Box test

data:  Residuals from ARIMA(0,1,1)
Q* = 14.796, df = 9, p-value = 0.0967

Model df: 1.   Total lags used: 10
```

Based on lower AIC/BIC values and better residual diagnostics (especially the Ljung-Box test), ARIMA(0,1,1) is the preferred model.

Problem 2

```
data(deere3)
model_ar1 <- Arima(deere3, order = c(1, 0, 0))

forecast_ar1 <- forecast::forecast(model_ar1, h = 10)

autoplot(forecast_ar1) +
  labs(
    title = "Forecast from AR(1) Model for deere3"
  ) +
  theme_minimal()
```




```
forecast_table <- as.data.frame(forecast_ar1)

print(forecast_table)
```

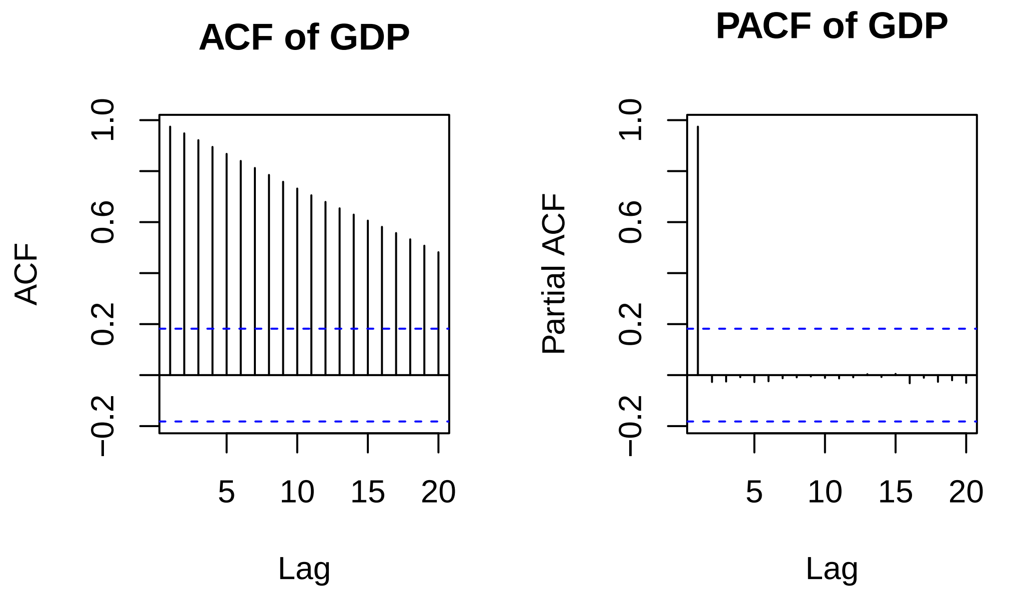
```
   Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
58    -335.145928 -2211.910 1541.618 -3205.409 2535.117
59    -117.120772 -2237.282 2003.041 -3359.628 3125.386
60      -2.538388 -2185.148 2180.071 -3340.551 3335.474
```

```
61      57.679997 -2141.865 2257.225 -3306.233 3421.593
62      89.327566 -2114.872 2293.527 -3281.705 3460.360
63     105.959839 -2099.523 2311.443 -3267.036 3478.955
64     114.700873 -2091.137 2320.539 -3258.837 3488.239
65     119.294695 -2086.641 2325.230 -3254.393 3492.982
66     121.708962 -2084.254 2327.672 -3252.020 3495.438
67     122.977772 -2082.992 2328.948 -3250.762 3496.718
```

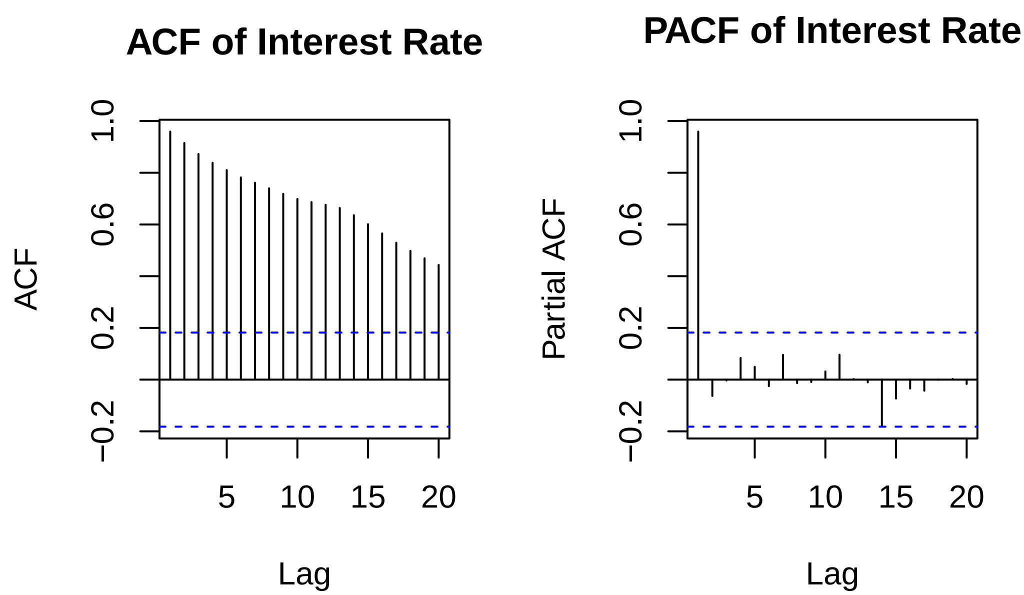
The forecast quickly reverts toward zero (the series mean), consistent with the nature of AR(1) models. The wide prediction intervals indicate substantial uncertainty, which is expected due to the variability observed in the original series.

Problem 3a

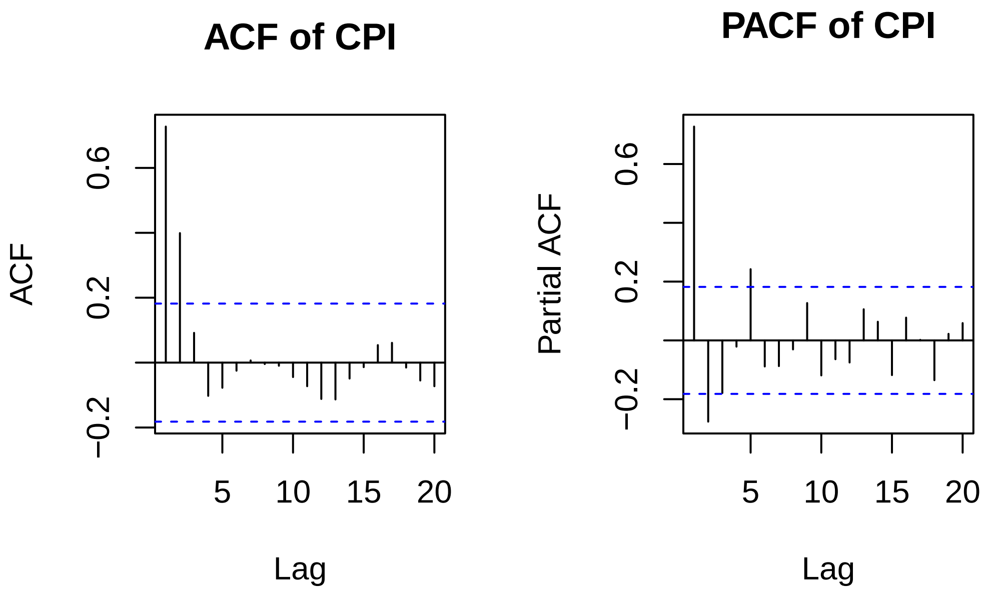
```r
library(tseries)

gdp <- read.csv("CAN_GDP.csv")
int <- read.csv("CAN_INT.csv")
cpi <- read.csv("CAN_CPI.csv")
pro <- read.csv("CAN_PRO.csv")

ts_gdp <- ts(gdp[, 7])
ts_int <- ts(int[, 7])
ts_cpi <- ts(cpi[, 7])
ts_pro <- ts(pro[, 7])

plot_acf_pacf <- function(ts_data, title) {
  par(mfrow = c(1, 2))
  acf(ts_data, main = paste("ACF of", title))
  pacf(ts_data, main = paste("PACF of", title))
  par(mfrow = c(1, 1))
}

plot_acf_pacf(ts_gdp, "GDP")
```



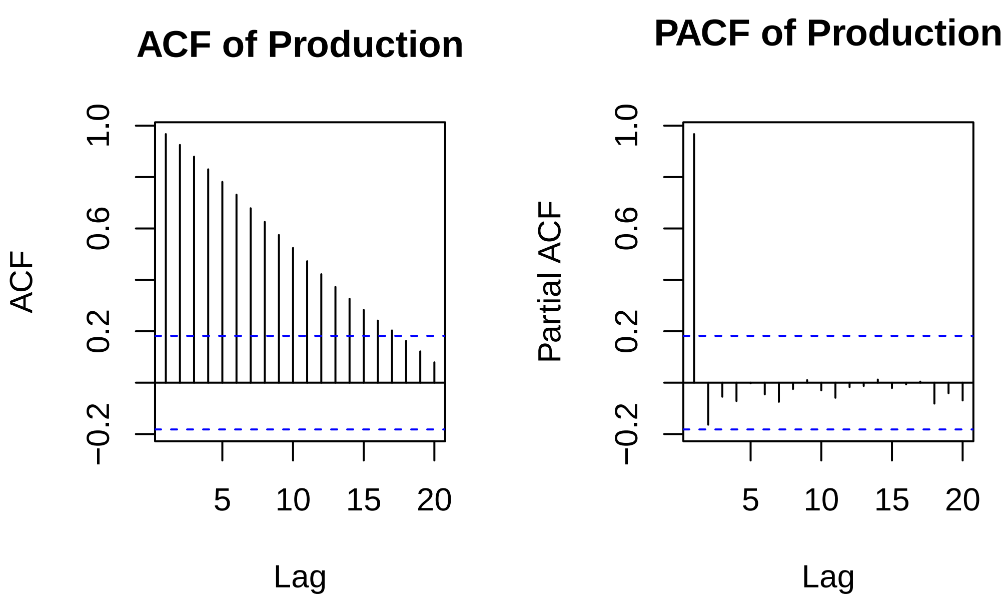

```
plot_acf_pacf(ts_int, "Interest Rate")
```



```r
plot_acf_pacf(ts_cpi, "CPI")
```




```r
plot_acf_pacf(ts_pro, "Production")
```

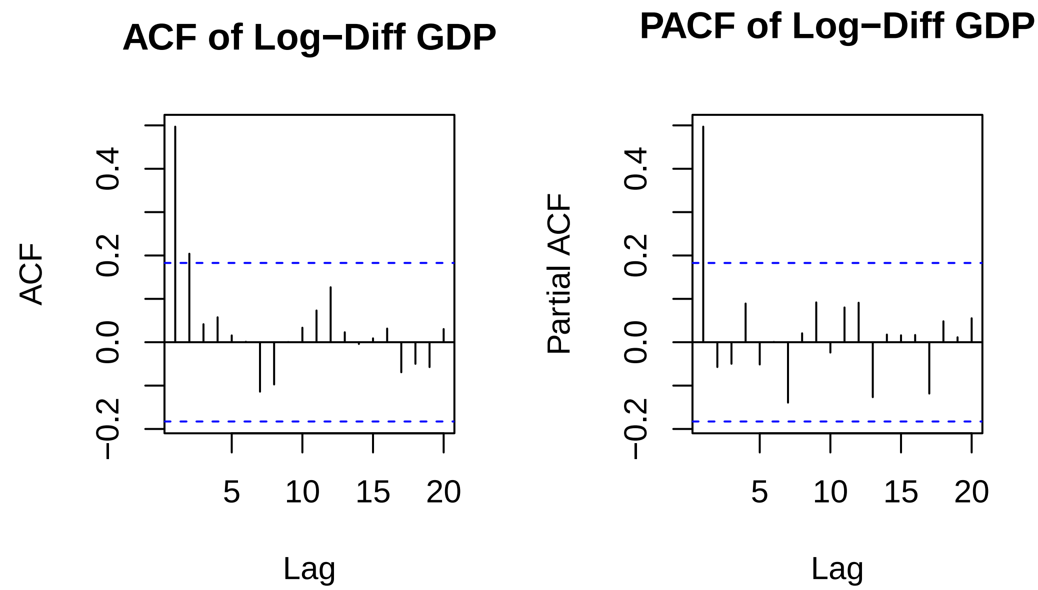


Stationarity Assessments

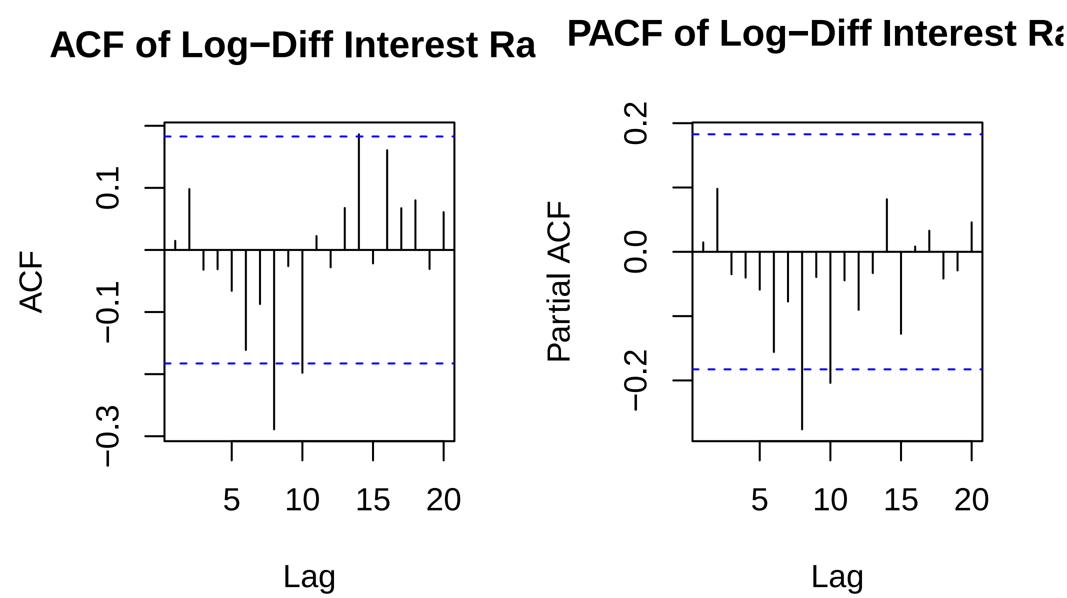
- GDP
  - The ACF shows a slow, exponential decay, typical of a non-stationary time series.
  - The PACF cuts off sharply after lag 1, typical of an AR(1) process after differencing, suggesting the underlying data could be modeled as an ARIMA(1,1,0) process.
- Interest Rate
  - Same assessment as GDP.
- CPI
  - ACF shows a large spike at lag 1, followed by quickly diminishing values which is a sign of stationarity.
  - The PACF cuts off sharply after lag 1, typical of an AR(1) process but differencing may not be required.
- Production
  - Same assessment as GDP and Interest Rate.

Problem 3b

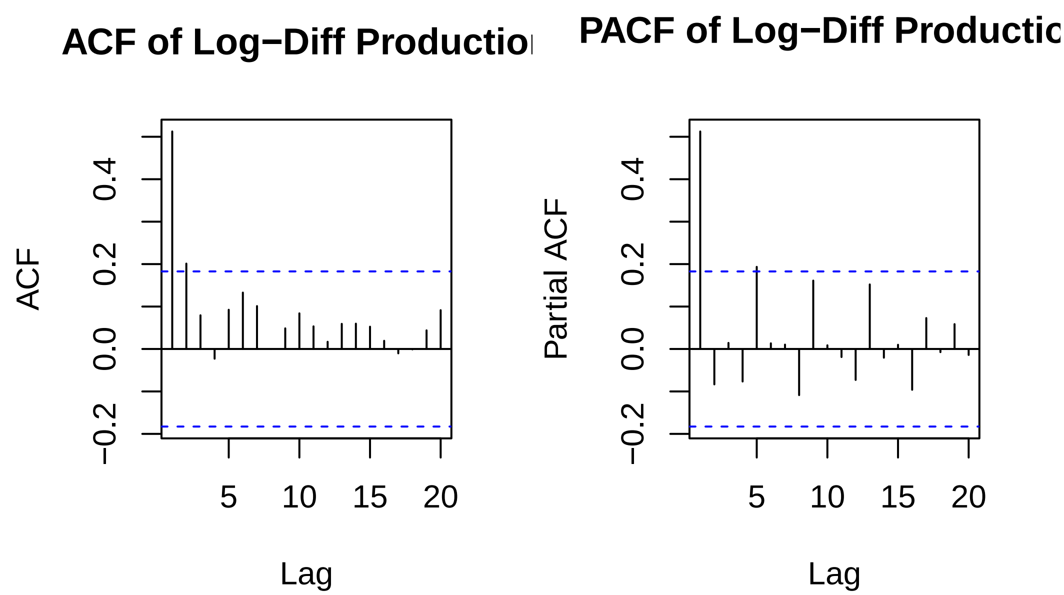
```r
log_diff <- function(series) {
  diff(log(series))
}

ts_gdp_diff <- log_diff(ts_gdp)
ts_int_diff <- log_diff(ts_int)
ts_pro_diff <- log_diff(ts_pro)

plot_acf_pacf(ts_gdp_diff, "Log-Diff GDP")
```




```r
plot_acf_pacf(ts_int_diff, "Log-Diff Interest Rate")
```



```r
plot_acf_pacf(ts_pro_diff, "Log-Diff Production")
```



Stationarity Assessments

- GDP
  - Only Lag 1 ACF is significant.
  - In the PACF, only the first lag is significant.
  - The transformed series appears stationary, possibly an AR(1) process.
- Interest Rate
  - ACF and PACF plots show most autocorrelations are within the confidence bounds.
  - The transformed series resembles white noise more closely.
- Production
  - Transformed series appears stationary for the same reasons as GDP.

Problem 3c

```r
library(vars)
```

```
Loading required package: MASS

Loading required package: strucchange

Loading required package: zoo


Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich

Loading required package: urca

Loading required package: lmtest
```

```r
clean_ts <- function(ts_data) {
  vec <- as.numeric(ts_data)
  vec[!is.finite(vec)] <- NA
  vec <- na.omit(vec)
  ts(vec, start = start(ts_data), frequency = frequency(ts_data))
}

ts_cpi_diff <- log_diff(ts_cpi)
ts_cpi_diff <- clean_ts(ts_cpi_diff) # Differenced CPI has INF, -INF. This gets rid of them.

combined_diff <- cbind(ts_gdp_diff, ts_int_diff, ts_cpi_diff, ts_pro_diff)
combined_diff <- na.omit(combined_diff) # Remove NA's.

VARselect(combined_diff)
```

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

$criteria
                   1             2             3             4             5
AIC(n) -2.567514e+01 -2.557521e+01 -2.550112e+01 -2.553925e+01 -2.554834e+01
HQ(n)  -2.546550e+01 -2.519786e+01 -2.495606e+01 -2.482647e+01 -2.466786e+01
SC(n)  -2.515729e+01 -2.464308e+01 -2.415472e+01 -2.377857e+01 -2.337339e+01
FPE(n)  7.072460e-12  7.828027e-12  8.462532e-12  8.204730e-12  8.225907e-12
                   6             7             8             9            10
AIC(n) -2.541890e+01 -2.528148e+01 -2.524889e+01 -2.512759e+01 -2.500693e+01
HQ(n)  -2.437071e+01 -2.406558e+01 -2.386527e+01 -2.357627e+01 -2.328789e+01
SC(n)  -2.282967e+01 -2.227798e+01 -2.183111e+01 -2.129553e+01 -2.076060e+01
FPE(n)  9.526864e-12  1.120079e-11  1.196037e-11  1.410061e-11  1.681873e-11
```

Based on the above output, VAR(1) is the most appropriate.

Problem 3d

```r
var_model <- VAR(combined_diff, p = 1)
summary(var_model)
```

```
VAR Estimation Results:
=========================
Endogenous variables: ts_gdp_diff, ts_int_diff, ts_cpi_diff, ts_pro_diff
Deterministic variables: const
Sample size: 110
Log Likelihood: 814.277
Roots of the characteristic polynomial:
0.5339 0.3401 0.1923 0.1062
Call:
VAR(y = combined_diff, p = 1)


Estimation results for equation ts_gdp_diff:
============================================
ts_gdp_diff = ts_gdp_diff.l1 + ts_int_diff.l1 + ts_cpi_diff.l1 + ts_pro_diff.l1 + const
```

```
                 Estimate Std. Error t value Pr(>|t|)
ts_gdp_diff.l1  0.3268818  0.1250852   2.613   0.0103 *
ts_int_diff.l1  0.0137291  0.0051736   2.654   0.0092 **
ts_cpi_diff.l1 -0.0020123  0.0009251  -2.175   0.0319 *
ts_pro_diff.l1  0.0635530  0.0425058   1.495   0.1379
const           0.0040070  0.0008061   4.971  2.6e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.004894 on 105 degrees of freedom
Multiple R-Squared: 0.3541, Adjusted R-squared: 0.3295
F-statistic: 14.39 on 4 and 105 DF,  p-value: 2.114e-09


Estimation results for equation ts_int_diff:
============================================
ts_int_diff = ts_gdp_diff.l1 + ts_int_diff.l1 + ts_cpi_diff.l1 + ts_pro_diff.l1 + const

                Estimate Std. Error t value Pr(>|t|)
ts_gdp_diff.l1  6.740654   2.309444   2.919  0.00430 **
ts_int_diff.l1  0.001691   0.095520   0.018  0.98591
ts_cpi_diff.l1  0.002474   0.017080   0.145  0.88513
ts_pro_diff.l1 -2.085685   0.784782  -2.658  0.00910 **
const          -0.046362   0.014882  -3.115  0.00237 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.09035 on 105 degrees of freedom
Multiple R-Squared: 0.0804, Adjusted R-squared: 0.04537
F-statistic: 2.295 on 4 and 105 DF,  p-value: 0.06408


Estimation results for equation ts_cpi_diff:
============================================
ts_cpi_diff = ts_gdp_diff.l1 + ts_int_diff.l1 + ts_cpi_diff.l1 + ts_pro_diff.l1 + const

               Estimate Std. Error t value Pr(>|t|)
ts_gdp_diff.l1  0.16196   12.71032   0.013  0.98986
ts_int_diff.l1  0.72471    0.52571   1.379  0.17097
ts_cpi_diff.l1 -0.28988    0.09400  -3.084  0.00261 **
ts_pro_diff.l1 -0.43724    4.31915  -0.101  0.91956
```

```
const          -0.01479    0.08191  -0.181  0.85702
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.4972 on 105 degrees of freedom
Multiple R-Squared: 0.09562,    Adjusted R-squared: 0.06117
F-statistic: 2.775 on 4 and 105 DF,  p-value: 0.03076


Estimation results for equation ts_pro_diff:
============================================
ts_pro_diff = ts_gdp_diff.l1 + ts_int_diff.l1 + ts_cpi_diff.l1 + ts_pro_diff.l1 + const

                Estimate Std. Error t value Pr(>|t|)
ts_gdp_diff.l1  0.912259   0.379862   2.402   0.0181 *
ts_int_diff.l1  0.025752   0.015711   1.639   0.1042
ts_cpi_diff.l1 -0.003897   0.002809  -1.387   0.1684
ts_pro_diff.l1  0.241205   0.129083   1.869   0.0645 .
const          -0.002635   0.002448  -1.076   0.2842
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.01486 on 105 degrees of freedom
Multiple R-Squared: 0.3271, Adjusted R-squared: 0.3015
F-statistic: 12.76 on 4 and 105 DF,  p-value: 1.683e-08



Covariance matrix of residuals:
            ts_gdp_diff ts_int_diff ts_cpi_diff ts_pro_diff
ts_gdp_diff   2.395e-05   7.631e-05  -0.0003236   4.887e-05
ts_int_diff   7.631e-05   8.163e-03   0.0033333   2.837e-04
ts_cpi_diff  -3.236e-04   3.333e-03   0.2472568  -1.531e-03
ts_pro_diff   4.887e-05   2.837e-04  -0.0015309   2.208e-04

Correlation matrix of residuals:
            ts_gdp_diff ts_int_diff ts_cpi_diff ts_pro_diff
ts_gdp_diff      1.0000     0.17259    -0.13298      0.6719
ts_int_diff      0.1726     1.00000     0.07419      0.2113
ts_cpi_diff     -0.1330     0.07419     1.00000     -0.2072
ts_pro_diff      0.6719     0.21130    -0.20717      1.0000
```

Problem 3e

```r
fit_ar_model <- function(ts_series, series_name) {
  model <- arima(ts_series, order = c(1, 0, 0))
  cat("\n===== AR(1) Model for", series_name, "=====\n")
  print(summary(model))
  return(model)
}

ar_gdp <- fit_ar_model(ts_gdp_diff, "GDP")
```

```
===== AR(1) Model for GDP =====

Call:
arima(x = ts_series, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.4943      6e-03
s.e.  0.0804      9e-04

sigma^2 estimated as 2.589e-05:  log likelihood = 443.98,  aic = -883.95

Training set error measures:

Warning in trainingaccuracy(object, test, d, D): test elements must be within
sample

              ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
```

```r
ar_int <- fit_ar_model(ts_int_diff, "Interest Rate")
```

```
===== AR(1) Model for Interest Rate =====

Call:
arima(x = ts_series, order = c(1, 0, 0))
```

```
Coefficients:
         ar1  intercept
      0.0148    -0.0162
s.e.  0.0935     0.0090

sigma^2 estimated as 0.008952:  log likelihood = 107.99,  aic = -211.97

Training set error measures:
```

```
Warning in trainingaccuracy(object, test, d, D): test elements must be within
sample
```

```
              ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
```

```r
ar_cpi <- fit_ar_model(ts_cpi_diff, "CPI")
```

```
===== AR(1) Model for CPI =====

Call:
arima(x = ts_series, order = c(1, 0, 0))

Coefficients:
          ar1  intercept
      -0.2788    -0.0195
s.e.   0.0906     0.0363

sigma^2 estimated as 0.2382:  log likelihood = -77.91,  aic = 159.82

Training set error measures:
```

```
Warning in trainingaccuracy(object, test, d, D): test elements must be within
sample
```

```
              ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
```

```r
ar_pro <- fit_ar_model(ts_pro_diff, "Production")
```

```
===== AR(1) Model for Production =====

Call:
arima(x = ts_series, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.5100     0.0033
s.e.  0.0795     0.0028

sigma^2 estimated as 0.0002231:  log likelihood = 320.11,  aic = -636.23

Training set error measures:
```

```
Warning in trainingaccuracy(object, test, d, D): test elements must be within
sample
```

```
              ME RMSE MAE MPE MAPE
Training set NaN  NaN NaN NaN  NaN
```

Problem 3f

```r
calculate_ar_mse <- function(models, data_list) {
  ar_mse <- sapply(seq_along(models), function(i) {
    resid <- residuals(models[[i]])
    mean(resid^2, na.rm = TRUE)
  })
  avg_mse <- mean(ar_mse)
  total_params <- length(models) * 2  # Each AR(1): 1 lag + 1 intercept
  list(avg_mse = avg_mse, total_params = total_params)
}

calculate_var_mse <- function(var_model) {
  resid <- residuals(var_model)

  # Remove rows with any NA/Inf across variables
```

```r
  resid_clean <- resid[apply(resid, 1, function(row) all(is.finite(row))), ]

  # MSE per series
  mse_per_series <- colMeans(resid_clean^2, na.rm = TRUE)
  avg_mse <- mean(mse_per_series)

  k <- ncol(resid)  # Number of variables
  p <- var_model$p  # VAR order
  total_params <- k^2 * p + k  # per standard VAR(p) parameter count

  list(avg_mse = avg_mse, total_params = total_params)
}

ar_models <- list(ar_gdp, ar_int, ar_cpi, ar_pro)
ar_results <- calculate_ar_mse(ar_models, list(ts_gdp_diff, ts_int_diff, ts_cpi_diff, ts_pro_
var_results <- calculate_var_mse(var_model)

table <- data.frame(
  Model = c("AR(1)", sprintf("VAR(%d)", var_model$p)),
  Avg_MSE = c(round(ar_results$avg_mse, 5), round(var_results$avg_mse, 5)),
  Num_Parameters = c(ar_results$total_params, var_results$total_params)
)

print(table)
```

```
   Model Avg_MSE Num_Parameters
1  AR(1) 0.06184              8
2 VAR(1) 0.06101             20
```

Problem 5g

Although the VAR(1) model has a slightly lower average MSE than the AR(1) models, the difference is small. The VAR(1) model uses more parameters than the AR(1) models which could lead to concerns about model complexity and overfitting.

Therefore, AR(1) is recommended for parsimony unless inter-variable relationships are essential to capture.