Time Series Analysis Homework Explaination

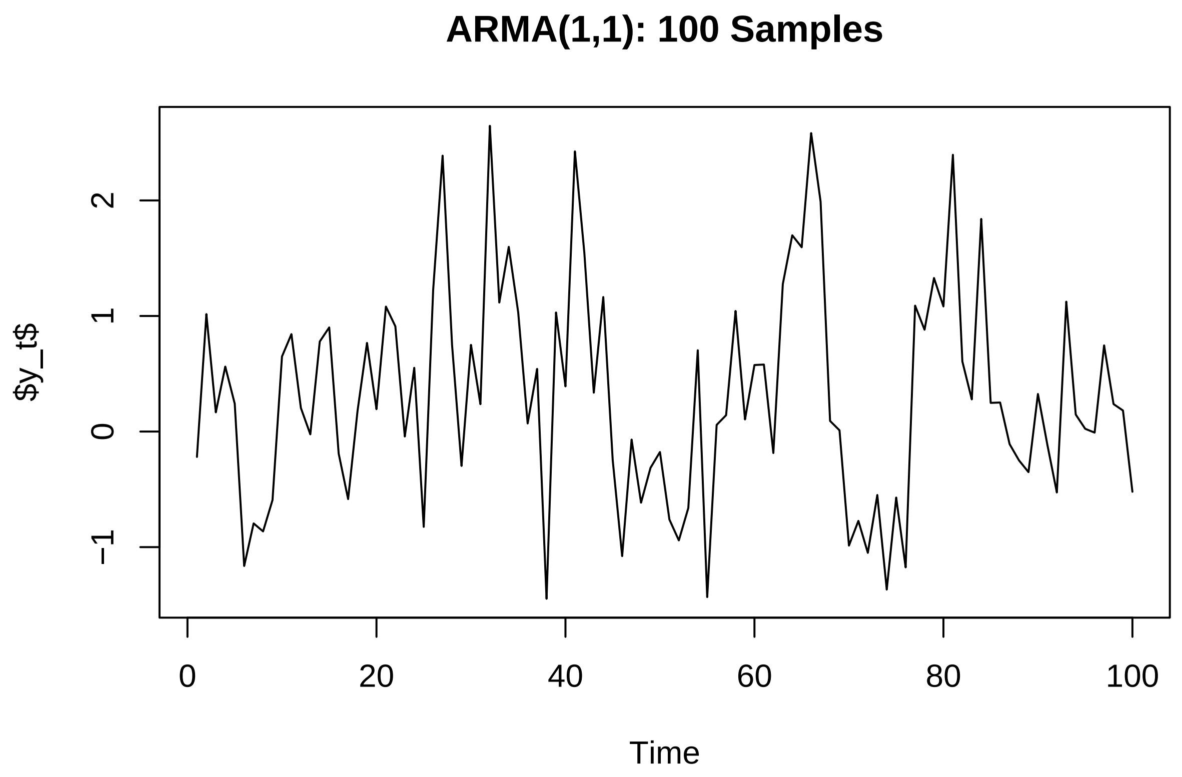
Hung-Ta, Wang (R13546017)

November 21, 2024

1. Simulate a time series $y_t$ of length $n = 100$ following an ARMA(1,1) model with $\phi = 0.8$ and $\theta = 0.4$.

Explanation:
We use {arima.sim} to simulate $y_t$ by $\phi = 0.8$ and $\theta = -0.4$
Generating function of ARMA(1,1)in R is $y_t = \phi y_{t-1} + a_t + \theta a_{t-1}$
Finally, we set $n = 100$ to generate 100 samples of $y_t$

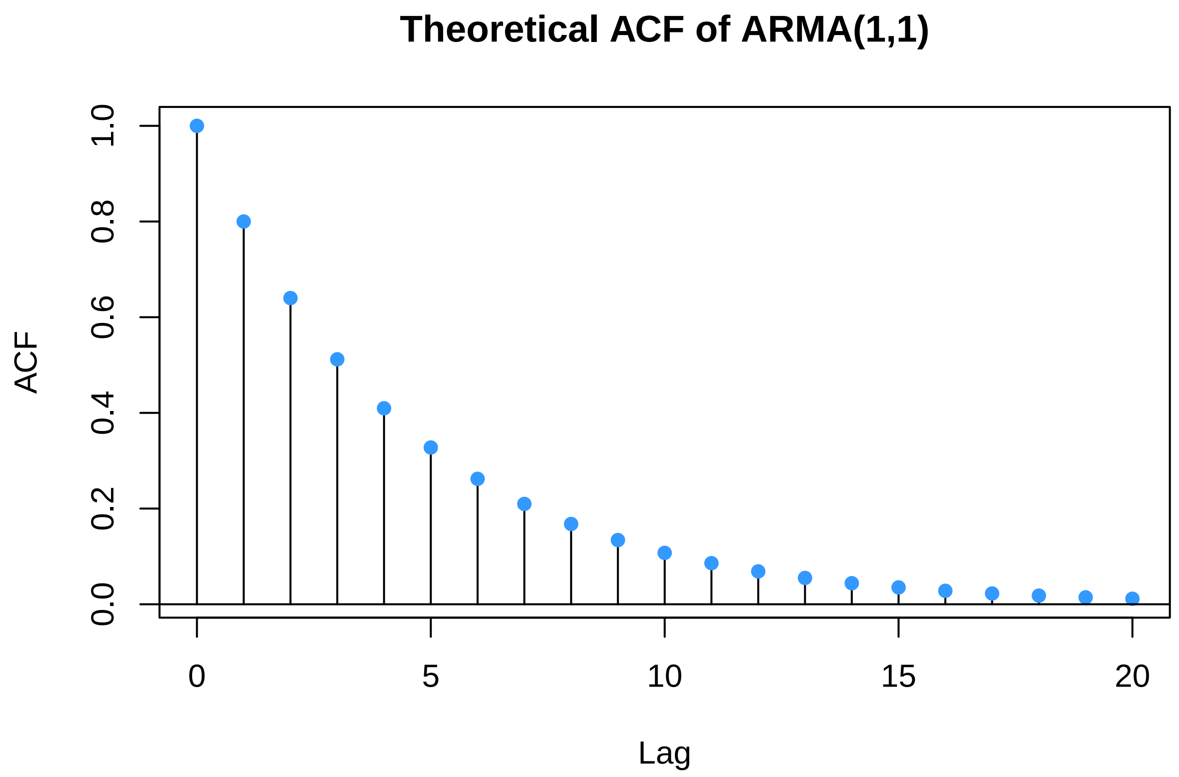
```r
library(forecast)
library(TSA)
library(tseries)

n <- 100
phi <- 0.8
theta <- -0.4

set.seed(1)
arma_model <- arima.sim(model = list(ar = phi, ma = theta), n = n)
plot(arma_model, main = "ARMA(1,1): 100 Samples", ylab = "$y_t$", xlab = "Time")
```



1.a. Calculate and plot the theoretical autocorrelation function for this model. Plot sufficient lags until the correlations are negligible.

Explanation:
We take 0.2 * length of y_t as the sufficient lags until the correlations are negligible
By Yule-Walker Equations we get: $\rho_{k+1} = \phi\rho_k$
Since ρ_0 is always 1, we use it to calculate theoretical autocorrelation ρ_{k+1}

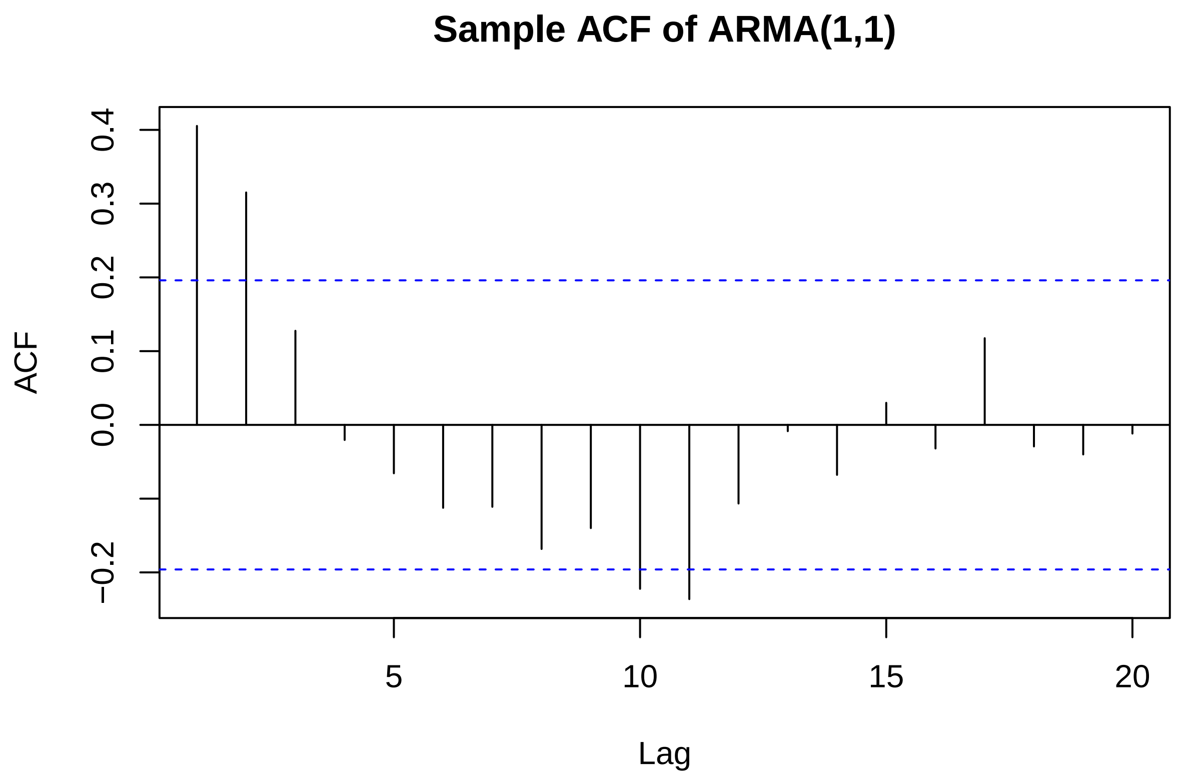
```r
max_lag <- 20
rho <- numeric(max_lag + 1)
rho[1] <- 1

for (k in 1:max_lag) {
  rho[k + 1] <- phi * rho[k]
}

lags <- 0:max_lag

plot(lags, rho, type = "h", xlab = "Lag", ylab = "ACF",
     main = "Theoretical ACF of ARMA(1,1)", col = "black", lwd = 1)
points(lags, rho, pch = 16, col = "#3399FF")
abline(h = 0, col = "black")
```



1.b. Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?

Explanation:
We use acf function in R to calculate sample ACF of generated y_t
In early periods, Sample ACF decreases as Theoretical ACF
However, Sample ACF performs different trends from Theoretical ACF
The reason is the cumulated effect of white noises becomes larger as the time flows
It would finally converge to 0 as the trend of Theoretical ACF of ARMA(1,1)

```
acf(arma_model, lag.max = max_lag, main = "Sample ACF of ARMA(1,1)")
```



1.c. Calculate and interpret the sample EACF for this series. Does the EACF help you specify the correct orders for the model?

Explanation:
We use eacf function in R to calculate sample EACF of generated y_t
It can be observed that the left-top corner is AR/MA = 1/1
Thus, EACF help us specify the correct orders for the model which is ARMA(1,1)

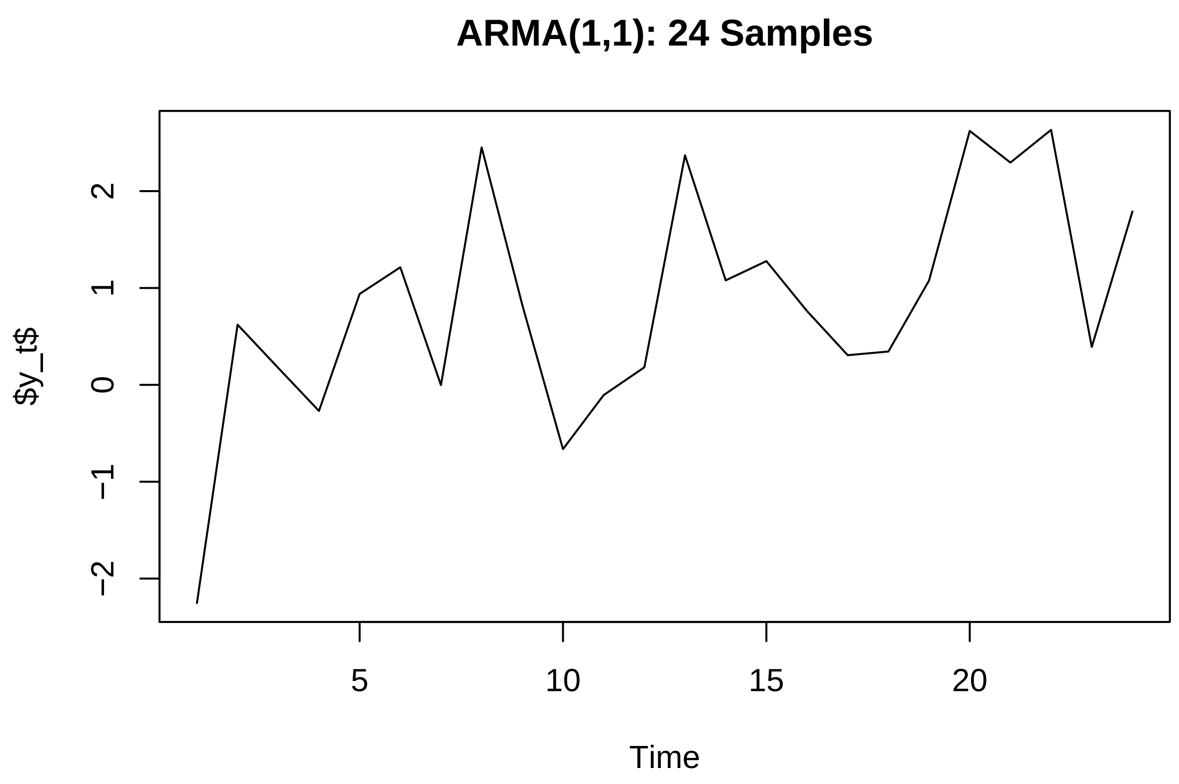
```
eacf(arma_model, ar.max = 9, ma.max = 9)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9
## 0 x x o o o o o o o x
## 1 x o o o o o o o o o
## 2 x x o o o o o o o o
## 3 x x o o o o o o o o
## 4 x o o o o o o o o o
## 5 x o o o o o o o o o
## 6 x o o x o o o o o o
## 7 o x x o o o o o o o
## 8 x x x o o x o o o o
## 9 x x o o o x o o o o
```

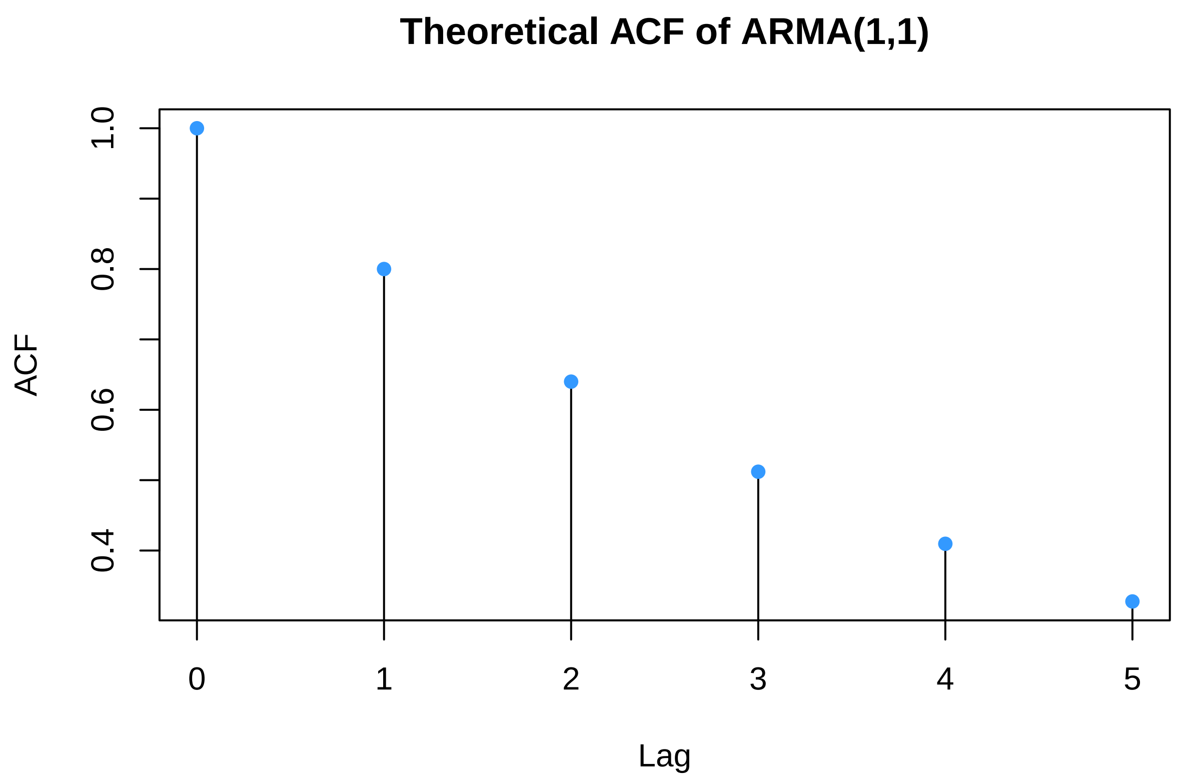
1.d. Repeat parts (b) and (c) with a new simulation using the same parameter values but sample size $n = 24$.

1.d.b. Explanation:
We use the same method to generate and analyze data as part (a) with the number of sample is changed into 24
We can observe that theoretical ACF is still the same whereas sample ACF does not obviously follow the rule of theoretical ACF
The reason is sample is too less to construct an obvious pattern

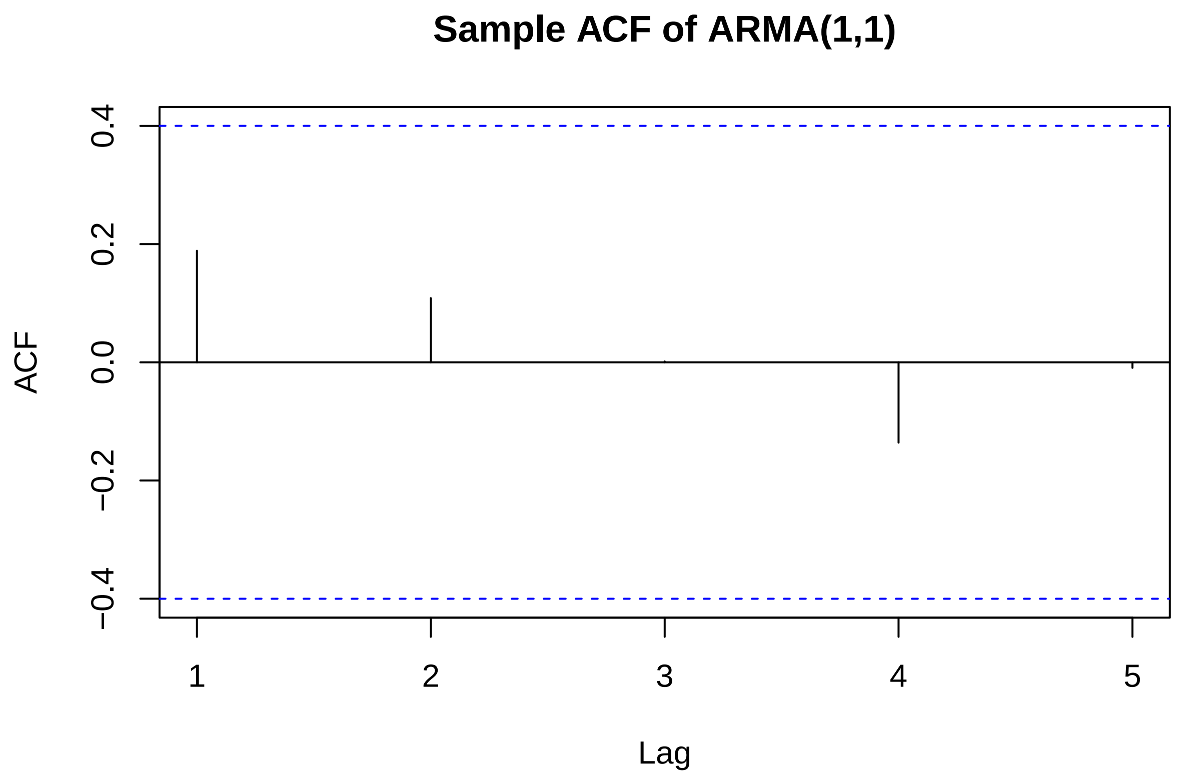
```r
n <- 24
max_lag <- 5
rho <- numeric(max_lag + 1)
rho[1] <- 1

arma_model <- arima.sim(model = list(ar = phi, ma = theta), n = n)
plot(arma_model, main = "ARMA(1,1): 24 Samples", ylab = "$y_t$", xlab = "Time")
```



```r
for (k in 1:max_lag) {
  rho[k + 1] <- phi * rho[k]
}

lags <- 0:max_lag

plot(lags, rho, type = "h", xlab = "Lag", ylab = "ACF",
     main = "Theoretical ACF of ARMA(1,1)", col = "black", lwd = 1)
points(lags, rho, pch = 16, col = "#3399FF")
abline(h = 0, col = "black")
```



```
acf(arma_model, lag.max = max_lag, main = "Sample ACF of ARMA(1,1)")
```



1.d.c. Explanation:
We can use all models to analyze this data
However, this result is biased since the sample is too less to detect which model fit it
EACF cannot help us specify the correct orders for the model

```
eacf_matrix <- eacf(arma_model, ar.max = 5, ma.max = 5)
```

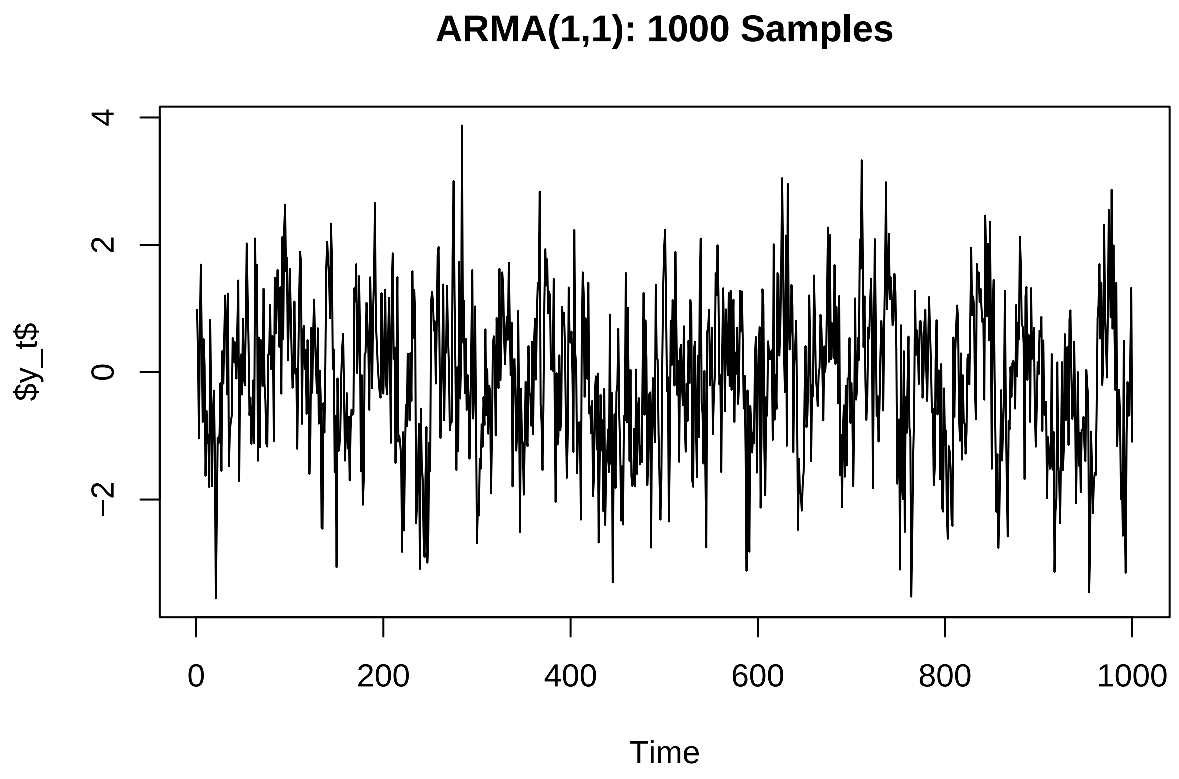
```
## AR/MA
##   0 1 2 3 4 5
## 0 o o o o o o
## 1 o o o o o o
## 2 o o o o o o
## 3 o o o o o o
## 4 o o o o o o
## 5 o o o o o o
```

1.e. Repeat parts (b) and (c) with a new simulation using the same parameter values but sample size $n = 1000$.

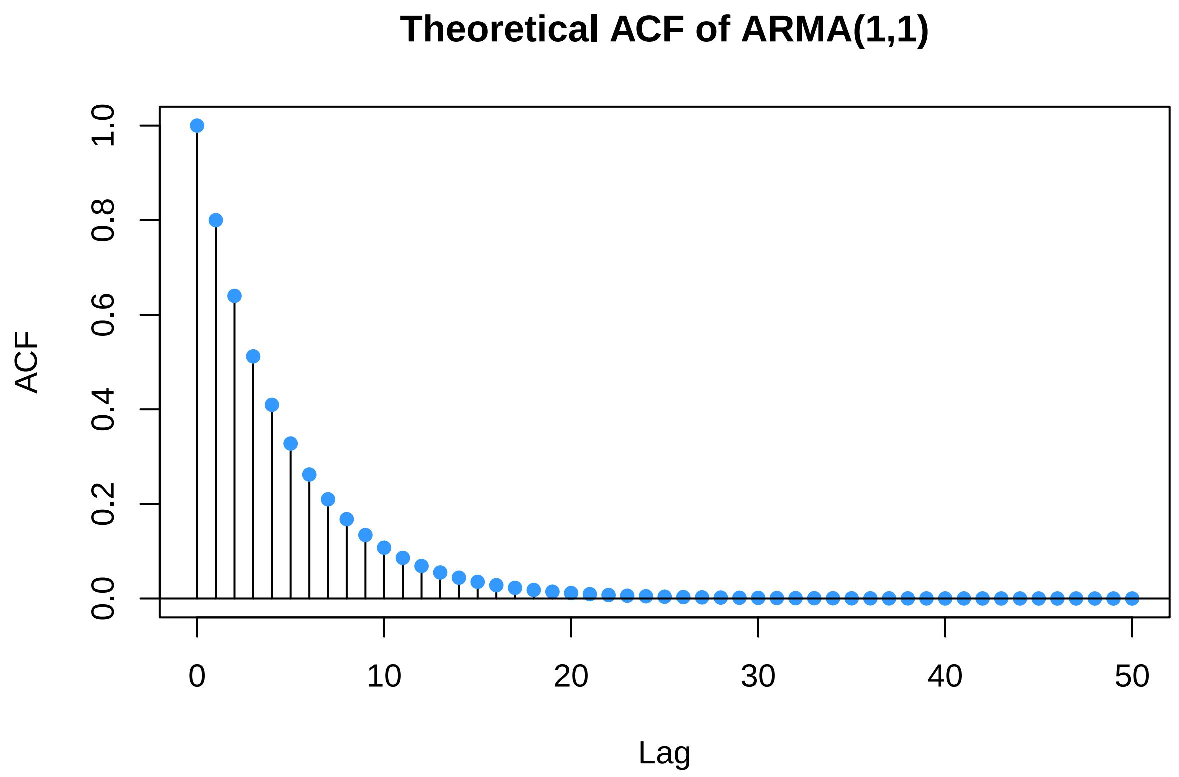
1.e.b. Explanation:
By using the same method to generate and analyze data as part (a) with the number of sample is changed into 1000
We set the maximum lags into 50 because $0.2 * 1000$ is still to large to identify the pattern
We can observe that theoretical ACF performs stronger rule as part (a) whereas sample ACF does not

perfectly follow the rule of theoretical ACF
The reason is also the cumulated effect of white noises
It will not be eliminated as the time flows

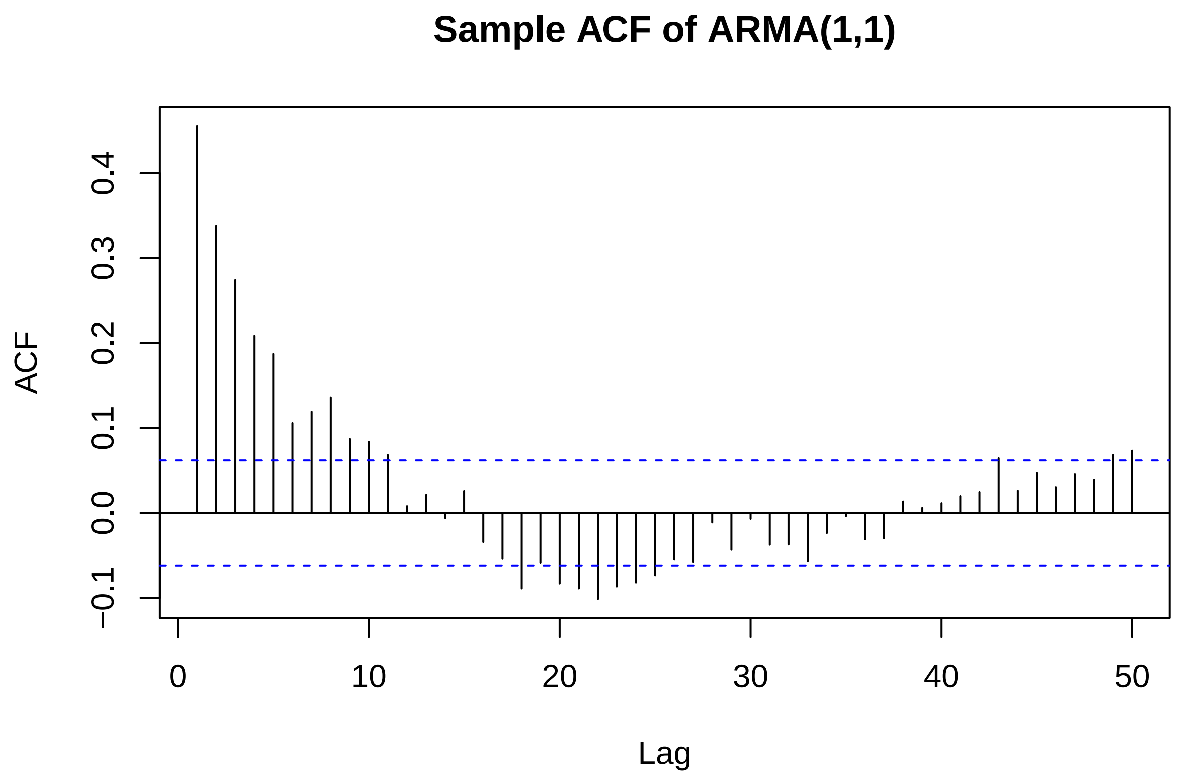
```r
n <- 1000
max_lag <- 50
rho <- numeric(max_lag + 1)
rho[1] <- 1

arma_model <- arima.sim(model = list(ar = phi, ma = theta), n = n)
plot(arma_model, main = "ARMA(1,1): 1000 Samples", ylab = "$y_t$", xlab = "Time")
```



```r
for (k in 1:max_lag) {
  rho[k + 1] <- phi * rho[k]
}

lags <- 0:max_lag

plot(lags, rho, type = "h", xlab = "Lag", ylab = "ACF",
     main = "Theoretical ACF of ARMA(1,1)", col = "black", lwd = 1)
points(lags, rho, pch = 16, col = "#3399FF")
abline(h = 0, col = "black")
```



```r
acf(arma_model, lag.max = max_lag, main = "Sample ACF of ARMA(1,1)")
```



1.e.c Explanation:
It can be observed that the left-top corner is AR/MA = 1/1
Thus, EACF help us specify the correct orders for the model which is ARMA(1,1)
We can also see that EACF shows more explicit specification since we have sufficient data

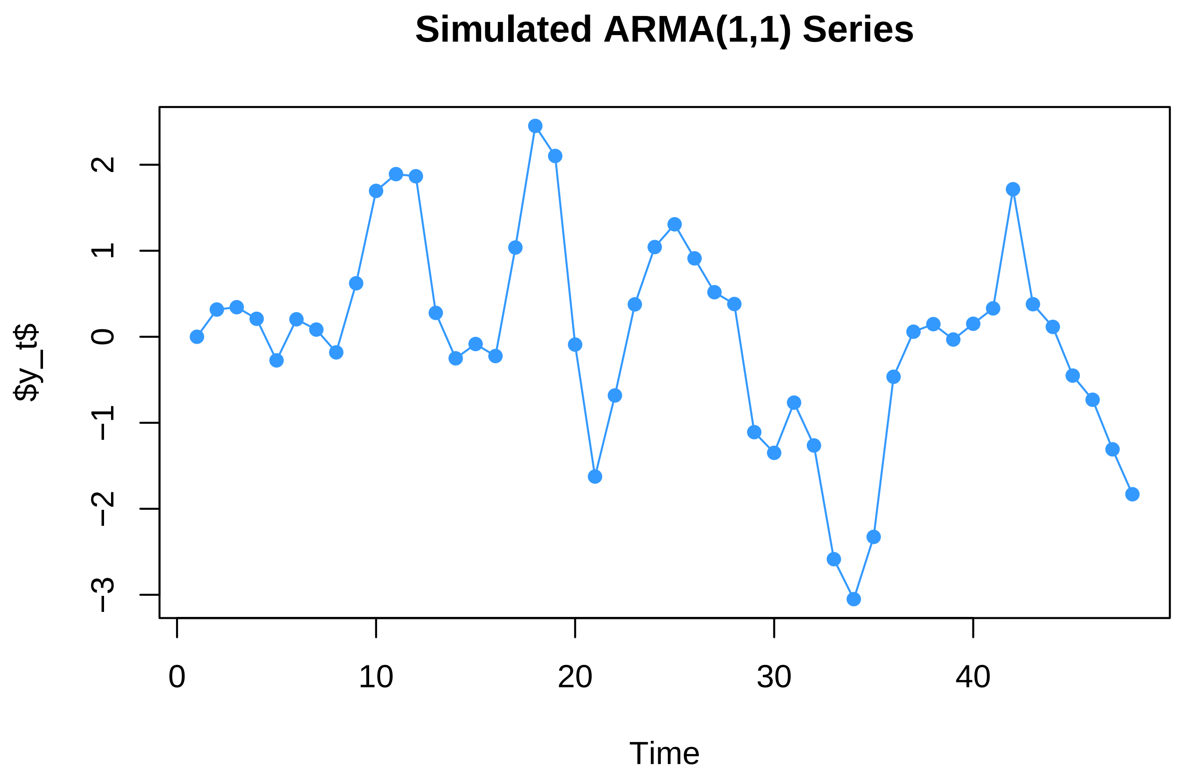
```r
eacf_matrix <- eacf(arma_model, ar.max = 9, ma.max = 9)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9
## 0 x x x x x x x x x x
## 1 x o o o x x o x o o
## 2 x x o o o x o x o o
## 3 x x o o o x o o o o
## 4 x x o o o x o o o o
## 5 x o x o x x o o o o
## 6 x o x x x x o o o o
## 7 x x x x o x o o o o
## 8 x o x x o x o o o o
## 9 x x o x x x o o x o
```

2. Simulate an ARMA(1,1) series with $\phi = 0.7$, $\theta = -0.6$, $n = 48$ but with error terms from a centered t-distribution with degrees of freedom 6.

Explanation:
We use the formula to simulate $y_t$ by $y_t = 0.7 y_{t-1} + a_t + 0.6 a_{t-1}
And we generate white noises which follow t-distribution then minus the mean of it to make it be centered
Finally, we set $n = 48$ to generate 48 samples of $y_t$

```r
phi <- 0.7
theta <- 0.6
n <- 48
df <- 6

set.seed(0)
errors <- rt(n, df = df)
errors <- errors - mean(errors)


arma_model <- numeric(n)
arma_model[1] <- 0

for (t in 2:n) {
  arma_model[t] <- phi * arma_model[t - 1] + errors[t] + theta * errors[t - 1]
}

plot(arma_model, type = "o", main = "Simulated ARMA(1,1) Series",
     xlab = "Time", ylab = "$y_t$", col = "#3399FF", pch = 16)
```



2.a. Display the sample EACF of the series. Is an ARMA(1,1) model suggested?

Explanation:
We can observe that the left-top corner is AR/MA = 1/1
Thus, an ARMA(1,1) model is suggested

```
eacf_matrix <- eacf(arma_model, ar.max = 9, ma.max = 9)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9
## 0 x x o o o o o o o o
## 1 x o o o o o o o o o
## 2 o o o o o o o o o o
## 3 x o o o o o o o o o
## 4 x o o o o o o o o o
## 5 o o o o o o o o o o
## 6 o x o o o o o o o o
## 7 o o o o o o o o o o
## 8 o o o o o o o o o o
## 9 x o o o o o o o o o
```

2.b. Estimate ϕ and θ from the series and comment on the results.

Explanation:
We use {summary(arma_fit)} to estimate ϕ and θ of ARIMA(1,0,1) model which is equivalent to ARMA(1,1)

Estimated $\phi = 0.5927$
Estimated $\theta = 0.4976$
We can see that estimated ϕ and estimated θ are all less than 1
It means the model is stationary

```
arma_fit <- arima(arma_model, order = c(1, 0, 1), method = "ML")
summary(arma_fit)
```

```
##
## Call:
## arima(x = arma_model, order = c(1, 0, 1), method = "ML")
##
## Coefficients:
##          ar1     ma1  intercept
##       0.5927  0.4976    -0.0758
## s.e.  0.1300  0.1149     0.3578
##
## sigma^2 estimated as 0.4852:  log likelihood = -51.37,  aic = 108.74
##
## Training set error measures:
##                ME RMSE MAE MPE MAPE
## Training set NaN  NaN NaN NaN  NaN
```

```
phi_hat <- arma_fit$coef["ar1"]
theta_hat <- arma_fit$coef["ma1"]

cat(sprintf("Estimated phi: %.4f\n", phi_hat))
```

```
## Estimated phi: 0.5927
```

```
cat(sprintf("Estimated theta: %.4f\n", theta_hat))
```

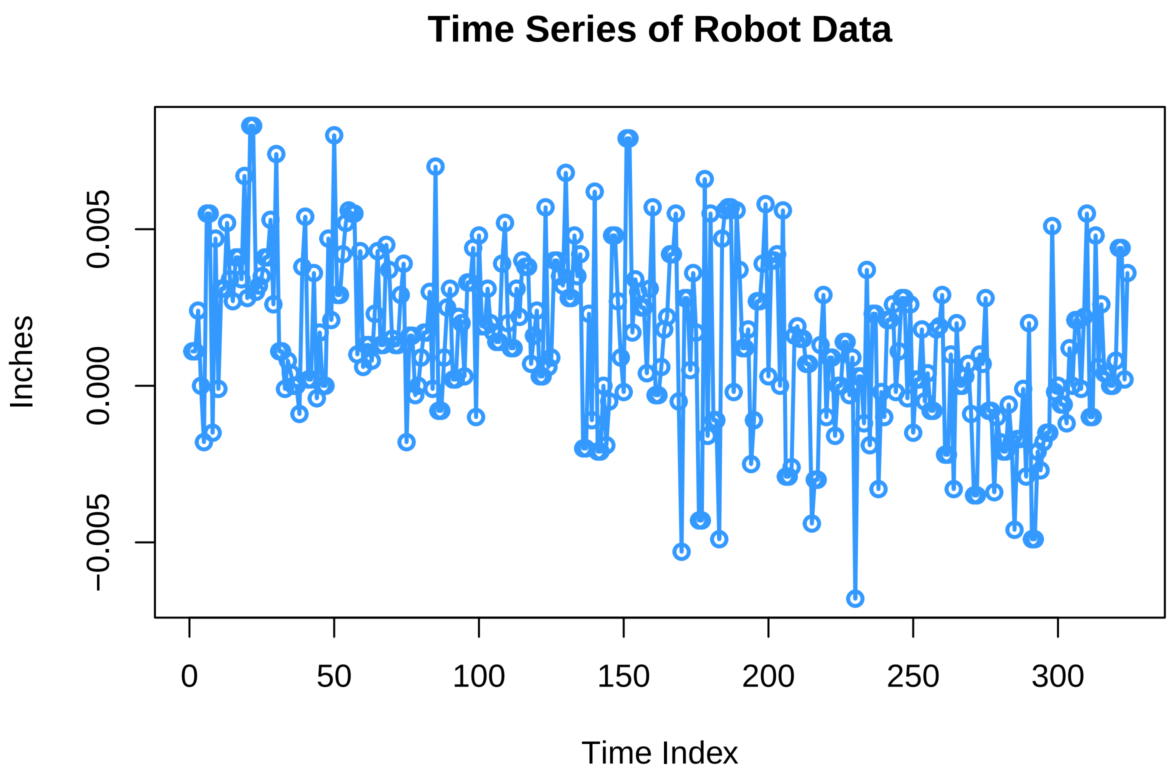
```
## Estimated theta: 0.4976
```

3. The data file named robot contains a time series obtained from an industrial robot. The robot was put through a sequence of maneuvers, and the distance from a desired ending point was recorded in inches.

3.a. Display the time series plot of the data. Based on the chart, do these data appear to come from a stationary or nonstationary process?

Explanation:
By Augmented Dickey-Fuller Test, we can see that p-value is less than 0.05
We reject null hypothesis and suppose that these data does not have unit root, which means it is stationary

```
file_path <- "C:/Git_Code/Some-practice/TSA HW06.robot.csv"
robot_data <- read.csv(file_path)
robot_ts <- ts(robot_data$robot)
plot(robot_ts, type = "o", col = "#3399FF", main = "Time Series of Robot Data",
     xlab = "Time Index", ylab = "Inches", lwd = 2)
```



```r
adf_test <- adf.test(robot_ts)
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  robot_ts
## Dickey-Fuller = -4.6522, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

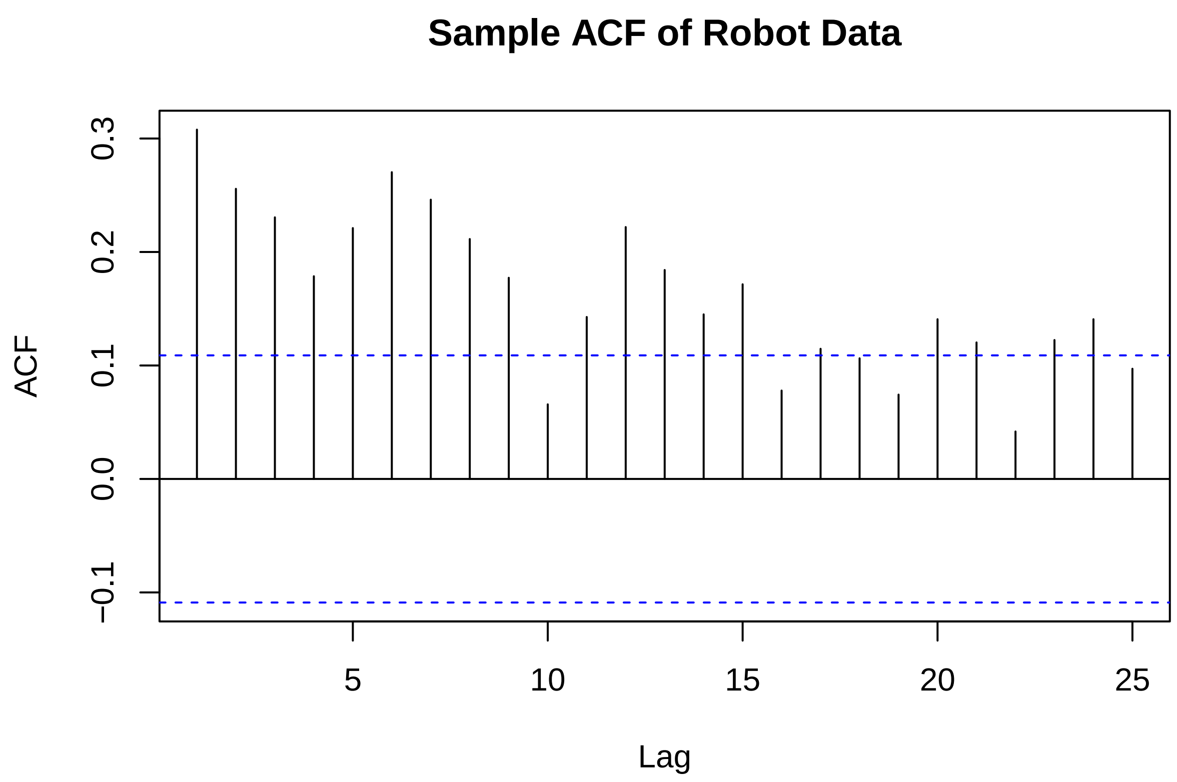
```r
if (adf_test$p.value < 0.05) {
  cat("The time series is stationary (reject null hypothesis of unit root).\n")
} else {
  cat("The time series is non-stationary (fail to reject null hypothesis of unit root).\n")
}
```

```
## The time series is stationary (reject null hypothesis of unit root).
```

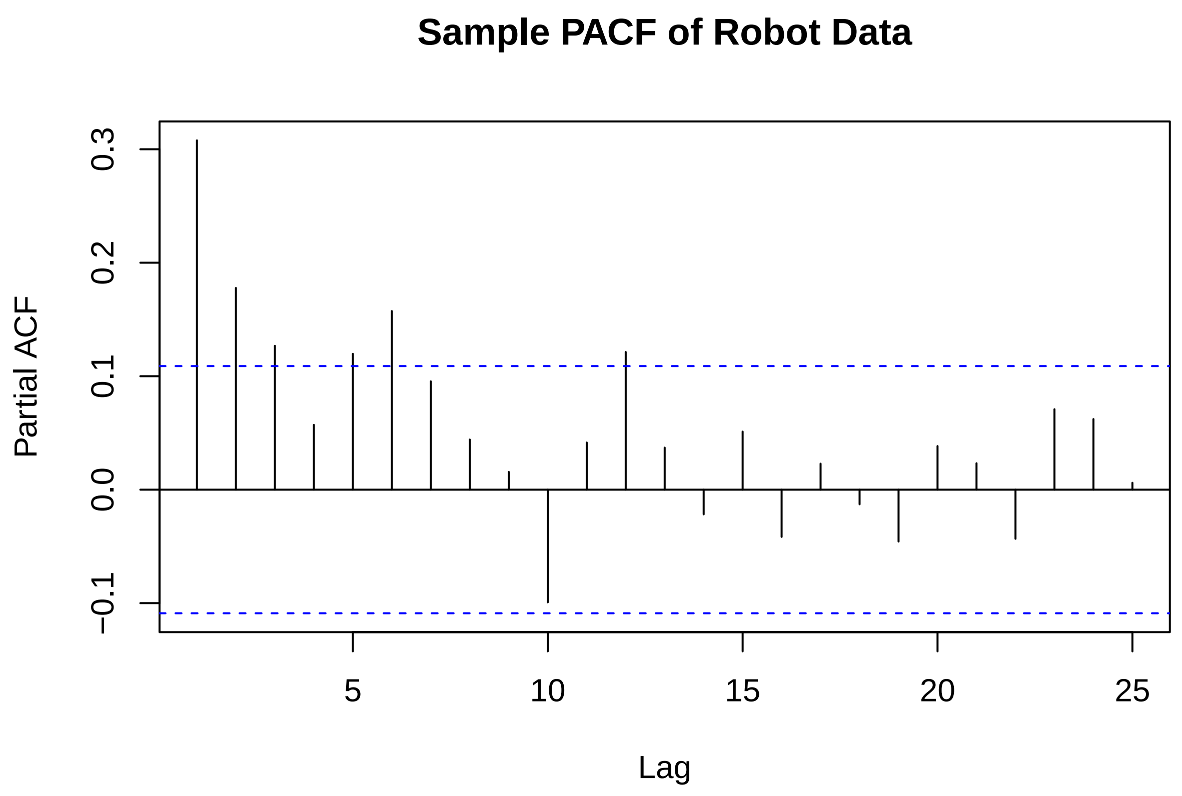
3.b. Calculate and plot the sample ACF and PACF for these data. Based on this additional information, do these data appear to come from a stationary or nonstationary process?

Explanation:
We can see that sample ACF still have some decreasing trends.
In addition, it seems to be converged into zero only when t gets extremely larger.
Thus, these data may not be stationary process even though we reject null hypothesis in ADF Test

```r
acf(robot_ts, main = "Sample ACF of Robot Data")
```



```r
pacf(robot_ts, main = "Sample PACF of Robot Data")
```



3.c. Calculate and interpret the sample EACF

Explanation:
We can observe that EACF recommend ARMA(1,1) for this model.

```
eacf_result <- eacf(robot_ts,  ar.max = 9, ma.max = 9)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9
## 0 x x x x x x x x x o
## 1 x o o o o o o o o o
## 2 x x o o o o o o o o
## 3 x x o o o o o o o o
## 4 x x x x o o o o o o
## 5 x x x o o o o o o o
## 6 x o o o o x o o o o
## 7 x o o x o x x o o o
## 8 x o o o o x o o o o
## 9 x x o o x x x o o o
```

3.d. Estimate the parameters of an AR(1) model and IMA(1, 1) for these data, respectively.

Explanation:
We use {ar1_fit} and {ima_fit} to estimate ϕ and θ of the robot data
The output below us the estimated ϕ and estimated θ if we want to use AR(1) or IMA(1,1)

```r
ar1_fit <- Arima(robot_ts, order = c(1, 0, 0))
phi_hat <- arma_fit$coef["ar1"]
cat(sprintf("Estimated phi: %.4f\n", phi_hat))
```

```
## Estimated phi: 0.5927
```

```r
ima_fit <- Arima(robot_ts, order = c(0, 1, 1))
theta_hat <- ima_fit$coef["ma1"]
cat(sprintf("Estimated theta: %.4f\n", theta_hat))
```

```
## Estimated theta: -0.8713
```

3.e. Compare the results from parts (d) in terms of AIC and discuss the residual tests.

Explanation:
1. From the aspect of AIC
We can see that AR(1) has a lower AIC than IMA(1,1)
It means AR(1) is more fittable to the data.

2. From the aspect of the residual tests
We can see the p-value of Ljung-Box test
It also tell us that p-value of AR(1) is 0.000003127 whereas p-value of IMA(1,1) is 0.0967
Thus, we can see that the white noises of AR(1) may not be random, however, IMA(1,1) may be white noise process

```r
ar1_aic <- AIC(ar1_fit)
ima_aic <- AIC(ima_fit)
cat(sprintf("AIC of AR(1): %.2f\n", ar1_aic))
```

```
## AIC of AR(1): -2945.08
```

```r
cat(sprintf("AIC of IMA(1,1): %.2f\n", ima_aic))
```

```
## AIC of IMA(1,1): -2957.90
```

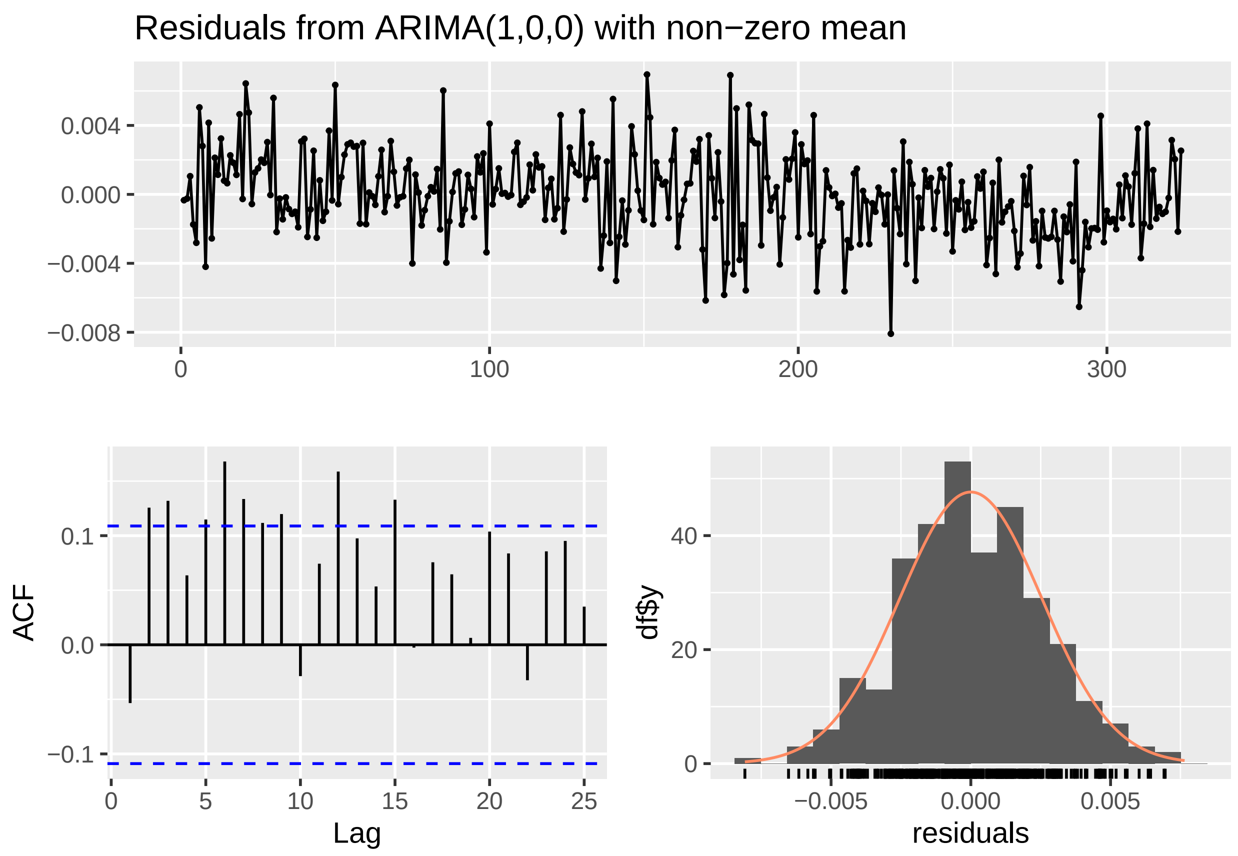
```r
if (ar1_aic < ima_aic) {
  cat("The AR(1) model has a lower AIC and is preferred.\n")
} else {
  cat("The IMA(1,1) model has a lower AIC and is preferred.\n")
}
```

```
## The IMA(1,1) model has a lower AIC and is preferred.
```

```r
cat("\nResidual Diagnostics for AR(1):\n")
```

```
##
## Residual Diagnostics for AR(1):
```

```r
checkresiduals(ar1_fit)
```

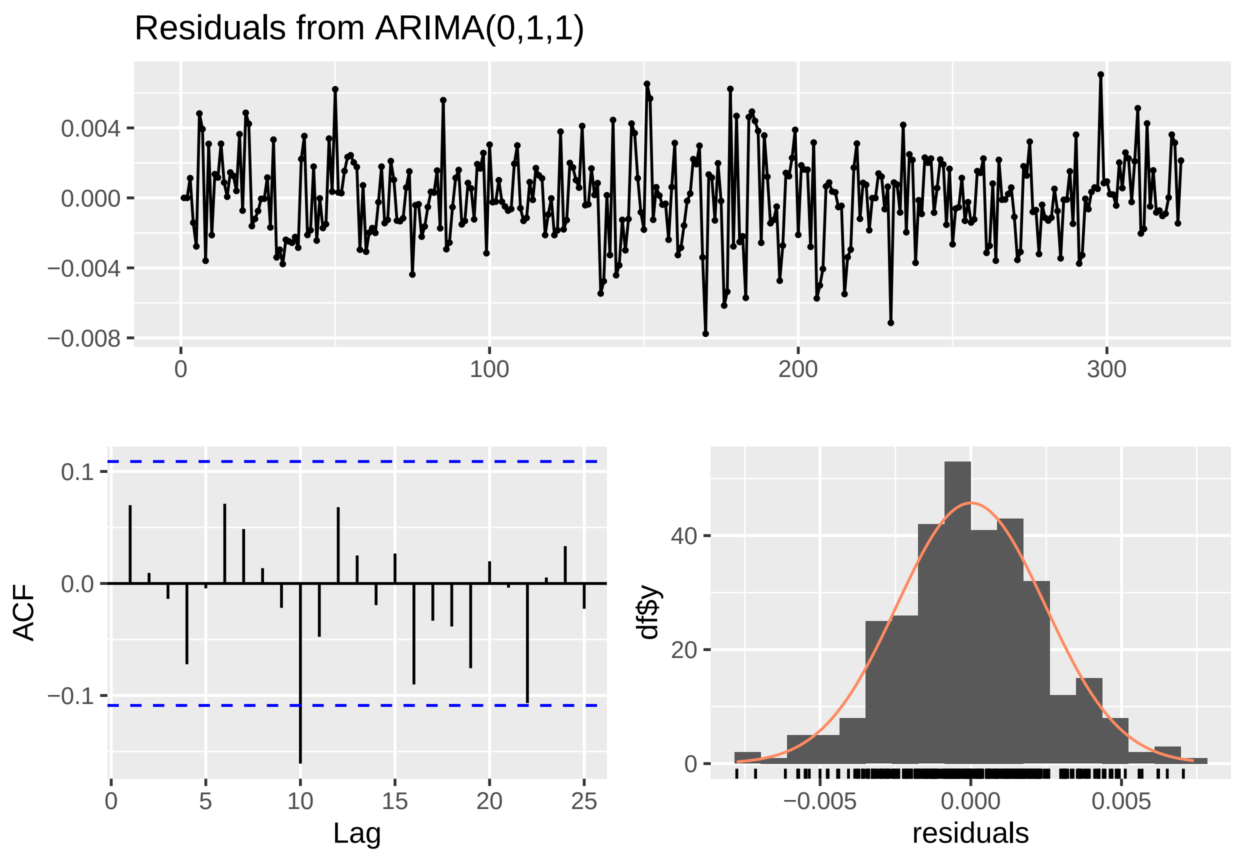


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 42.118, df = 9, p-value = 3.127e-06
##
## Model df: 1.   Total lags used: 10
```

```r
cat("\nResidual Diagnostics for IMA(1,1):\n")
```

```
##
## Residual Diagnostics for IMA(1,1):
```

```
checkresiduals(ima_fit)
```



```
##
##  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
```