ST565 Homework 3

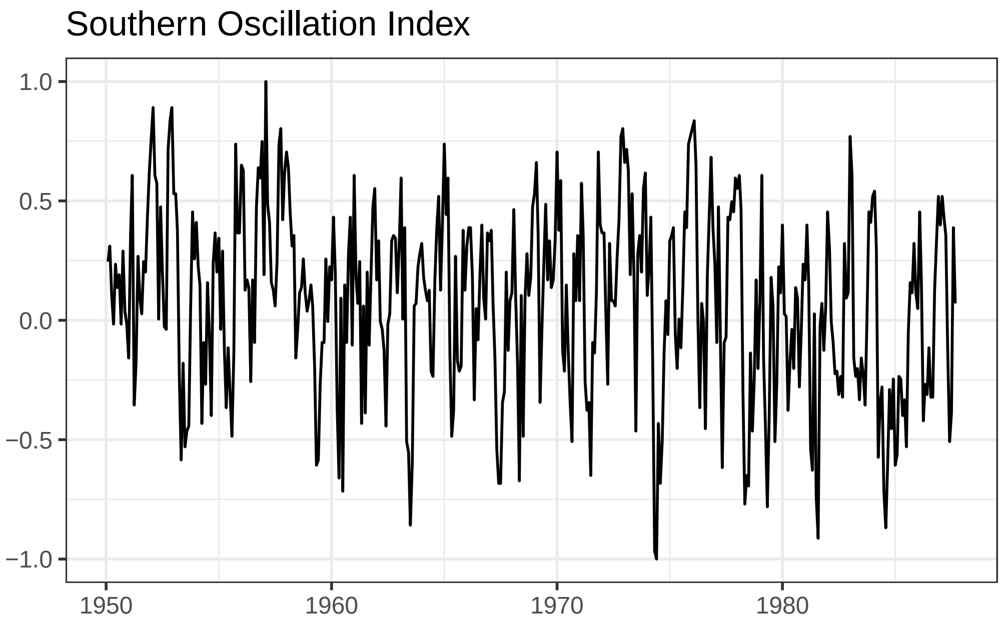
Brian Cervantes Alvarez
March 12, 2024

Problem 1

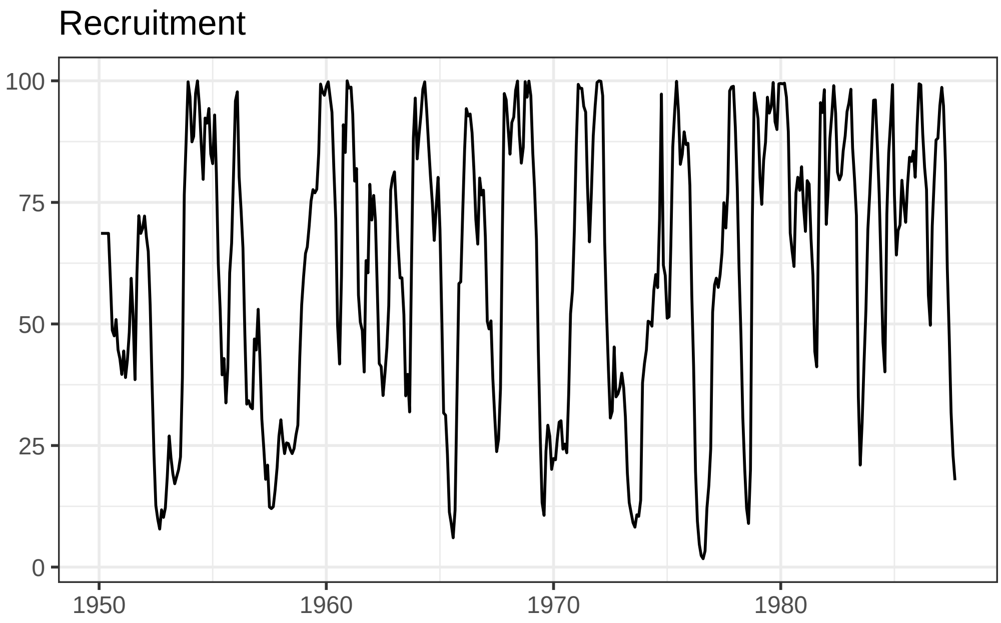
Part A

The recruitment time series exhibits a more distinct cycle and seasonality. In contrast, the southern index does not show a clear seasonal pattern. However, there appears to be a regular cycle occurring every five years according to the plot. Further analysis, including detrending and examining seasonality, is needed to obtain a complete understanding.

```r
library(ggplot2)
library(dplyr)
soiDs <- read.csv("soi_data.csv")
recDs <- read.csv("rec_data.csv")

soiDs <- soiDs %>%
  mutate(Year = floor(X1950),
         Month = round((X1950 - Year) * 12) + 1,
         Monthly = as.Date(paste(Year, Month, "01", sep = "-")),
         X = X0.377)  %>%
  dplyr::select(-X1950, -Year, -Month, -X0.377)


recDs <- recDs %>%
  mutate(Year = floor(X1950),
         Month = round((X1950 - Year) * 12) + 1,
         Monthly = as.Date(paste(Year, Month, "01", sep = "-")),
         X = X68.63)  %>%
  dplyr::select(-X1950, -Year, -Month, -X68.63)

# Plot SOI
ggplot(soiDs, aes(x = Monthly, y = X)) +
  geom_line() +
  labs(title = "Southern Oscillation Index", x = "", y = "") +
  theme_bw()
```



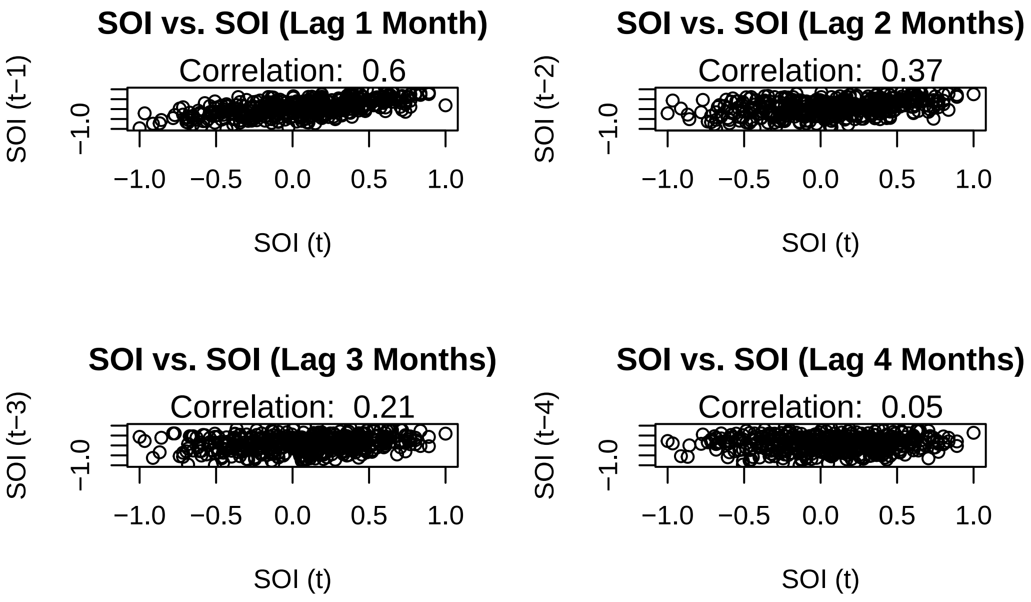
```
# Plot Recruitment
ggplot(recDs, aes(x = Monthly, y = X)) +
  geom_line() +
  labs(title = "Recruitment", x = "", y = "") +
  theme_bw()
```



This approach, analyzing autocorrelation in time series data, is useful for identifying trends or seasonality, building predictive models like ARIMA, transforming data to ensure stationarity, and detecting anomalies. Understanding the autocorrelation function helps us gain insights into the characteristics of the data and guide further analysis or modeling.

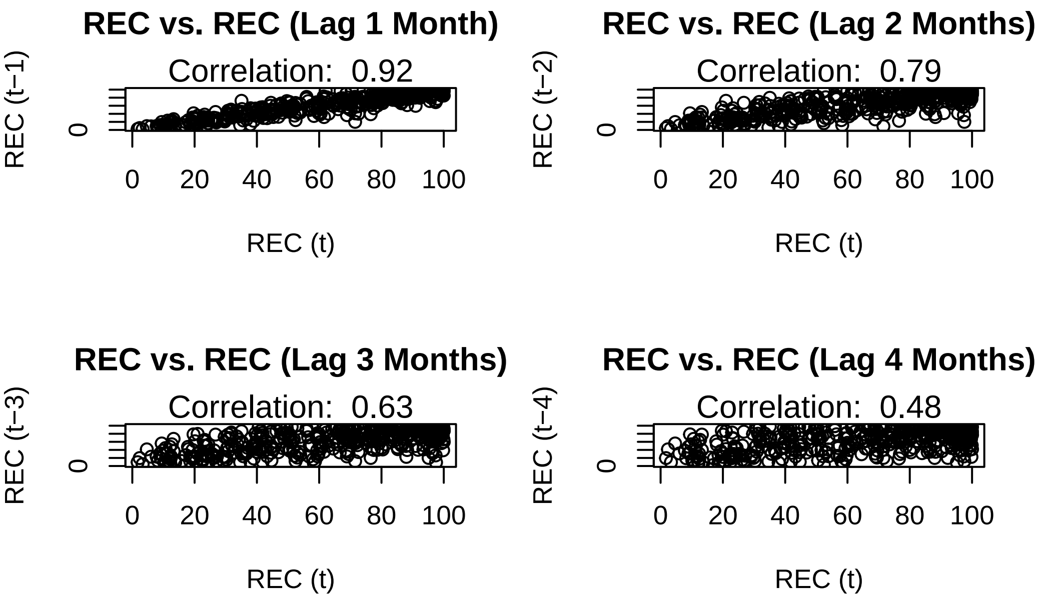
Part B

```
soi <- ts(soiDs$X, frequency = 12)
rec <- ts(recDs$X, frequency = 12)

# SOI plots
par(mfrow = c(2, 2))

# Lag 1
lag1_soi <- stats::lag(soi, -1)
plot(soi[-1], lag1_soi[-length(lag1_soi)], main="SOI vs. SOI (Lag 1 Month)",
     xlab="SOI (t)", ylab="SOI (t-1)")
cor1_soi <- cor(soi[-1], lag1_soi[-length(lag1_soi)])
mtext(paste("Correlation: ", round(cor1_soi, 2)), 3)

# Lag 2
lag2_soi <- stats::lag(soi, -2)
plot(soi[-(1:2)], lag2_soi[-(length(lag2_soi)-(0:1))],
     main="SOI vs. SOI (Lag 2 Months)", xlab="SOI (t)", ylab="SOI (t-2)")
cor2_soi <- cor(soi[-(1:2)], lag2_soi[-(length(lag2_soi)-(0:1))])
mtext(paste("Correlation: ", round(cor2_soi, 2)), 3)

# Lag 3
lag3_soi <- stats::lag(soi, -3)
plot(soi[-(1:3)], lag3_soi[-(length(lag3_soi)-(0:2))],
     main="SOI vs. SOI (Lag 3 Months)", xlab="SOI (t)", ylab="SOI (t-3)")
cor3_soi <- cor(soi[-(1:3)], lag3_soi[-(length(lag3_soi)-(0:2))])
mtext(paste("Correlation: ", round(cor3_soi, 2)), 3)

# Lag 4
lag4_soi <- stats::lag(soi, -4)
plot(soi[-(1:4)], lag4_soi[-(length(lag4_soi)-(0:3))],
     main="SOI vs. SOI (Lag 4 Months)", xlab="SOI (t)", ylab="SOI (t-4)")
cor4_soi <- cor(soi[-(1:4)], lag4_soi[-(length(lag4_soi)-(0:3))])
mtext(paste("Correlation: ", round(cor4_soi, 2)), 3)
```



```r
# REC plots
par(mfrow = c(2, 2))  # Resetting the plotting area to a 2x2 grid

# Lag 1
lag1_rec <- stats::lag(rec, -1)
plot(rec[-1], lag1_rec[-length(lag1_rec)],
     main="REC vs. REC (Lag 1 Month)", xlab="REC (t)", ylab="REC (t-1)")
cor1_rec <- cor(rec[-1], lag1_rec[-length(lag1_rec)])
mtext(paste("Correlation: ", round(cor1_rec, 2)), 3)

# Lag 2
lag2_rec <- stats::lag(rec, -2)
plot(rec[-(1:2)], lag2_rec[-(length(lag2_rec)-(0:1))],
     main="REC vs. REC (Lag 2 Months)", xlab="REC (t)", ylab="REC (t-2)")
cor2_rec <- cor(rec[-(1:2)], lag2_rec[-(length(lag2_rec)-(0:1))])
mtext(paste("Correlation: ", round(cor2_rec, 2)), 3)

# Lag 3
lag3_rec <- stats::lag(rec, -3)
plot(rec[-(1:3)], lag3_rec[-(length(lag3_rec)-(0:2))],
     main="REC vs. REC (Lag 3 Months)", xlab="REC (t)", ylab="REC (t-3)")
cor3_rec <- cor(rec[-(1:3)], lag3_rec[-(length(lag3_rec)-(0:2))])
mtext(paste("Correlation: ", round(cor3_rec, 2)), 3)

# Lag 4
lag4_rec <- stats::lag(rec, -4)
plot(rec[-(1:4)], lag4_rec[-(length(lag4_rec)-(0:3))],
     main="REC vs. REC (Lag 4 Months)", xlab="REC (t)", ylab="REC (t-4)")
```

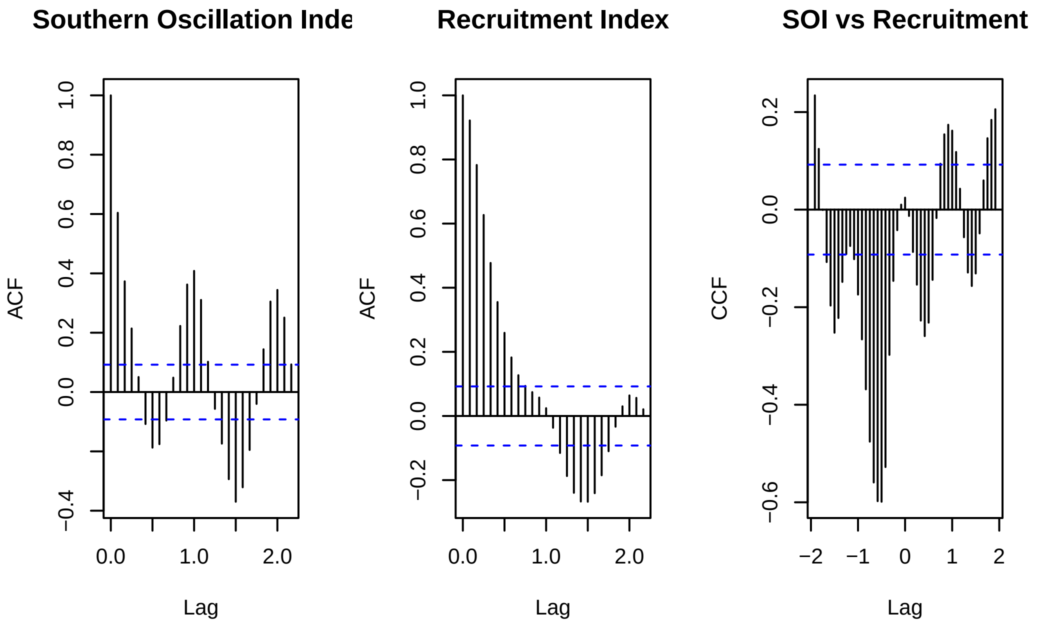
```
cor4_rec <- cor(rec[-(1:4)], lag4_rec[-(length(lag4_rec)-(0:3))])
mtext(paste("Correlation: ", round(cor4_rec, 2)), 3)
```



Part C

The ACF plots show autocorrelation within the SOI and Recruitment Index, indicating internal patterns, while the CCF plot reveals a significant relationship between the two series, suggesting interactions across time. Significant bars beyond confidence intervals in ACF suggest predictable internal patterns, whereas in CCF, they indicate predictive relationships between SOI and Recruitment. These identified patterns are essential for forecasting!

```r
# ACF Plots
par(mfrow = c(1, 3))

# Plot the ACF for the SOI
acf(soi, main="Southern Oscillation Index")

# Plot the ACF for the REC
acf(rec, main="Recruitment Index")

# Plot the CCF between SOI and REC
ccf(soi, rec, main="SOI vs Recruitment", ylab="CCF")
```



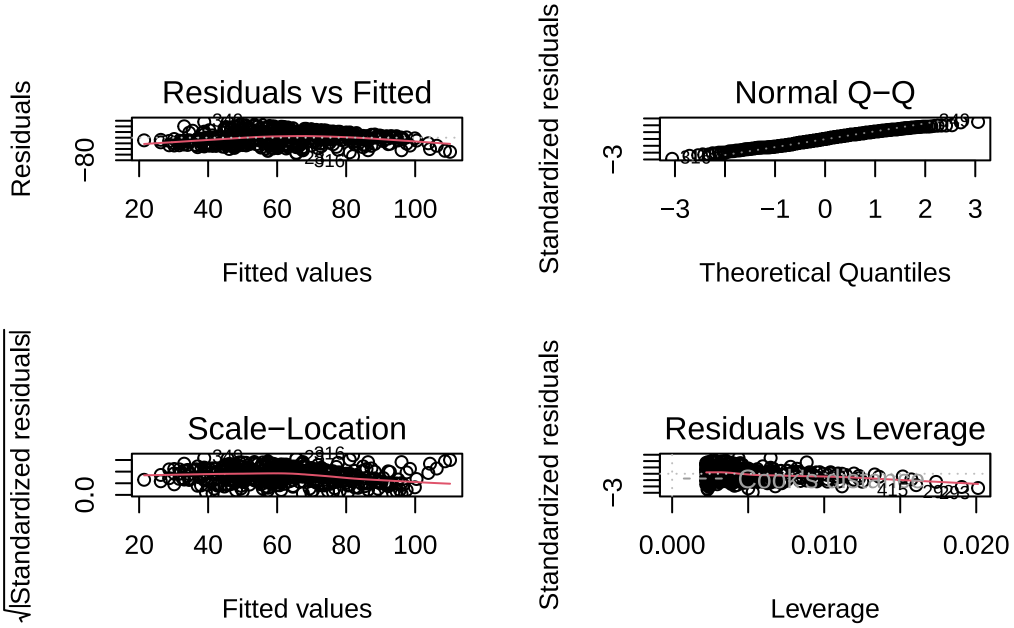
Part D

One of the issues result in using this method is that it assumes linearity and may ignore other influential factors, potentially oversimplifying the complex dynamics involved. That is why we use diagnostic plots to indicate possible non-linearity, heteroscedasticity, and the presence of influential outliers, suggesting the model's assumptions may not fully hold. To enhance model accuracy and reliability, considering non-linear relationships, additional variables, and addressing assumption violations could be necessary.

```r
soiL6 <- stats::lag(soi, -6)

# Aligning the lagged SOI series with the Recruitment series
fish <- ts.intersect(rec = rec, soiL6 = soiL6, dframe = TRUE)

# Fitting the linear regression model
fit1 <- lm(rec ~ soiL6, data = fish, na.action = NULL)

summary(fit1)
```

```
Call:
lm(formula = rec ~ soiL6, data = fish, na.action = NULL)

Residuals:
   Min     1Q Median     3Q    Max 
-65.19 -18.28   0.33  16.66  55.84 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   65.771      1.090   60.35   <2e-16 ***
soiL6        -44.328      2.785  -15.91   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.52 on 444 degrees of freedom
Multiple R-squared:  0.3632,	Adjusted R-squared:  0.3618 
F-statistic: 253.3 on 1 and 444 DF,  p-value: < 2.2e-16
```

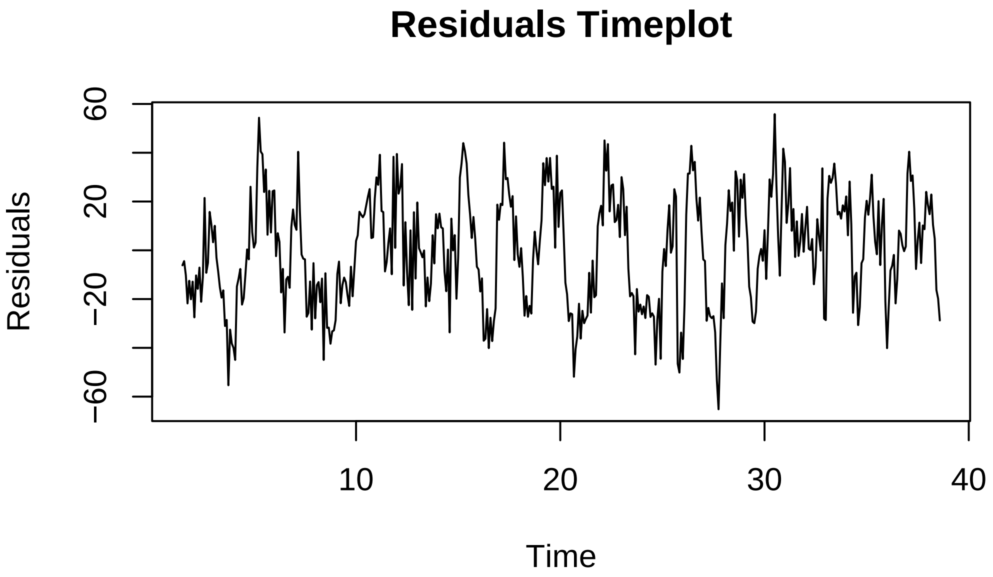
```r
par(mfrow = c(2, 2))
plot(fit1)
```


Residuals vs Fitted
Residuals
−80
20
40
60
80
100
Fitted values
349
316
Normal Q−Q
Standardized residuals
−3
−1
0
1
2
3
Theoretical Quantiles
349
316
Scale−Location
√|Standardized residuals|
0.0
20
40
60
80
100
Fitted values
349
316
Residuals vs Leverage
Standardized residuals
−3
0.000
0.010
0.020
Leverage
Cook's distance
415
293

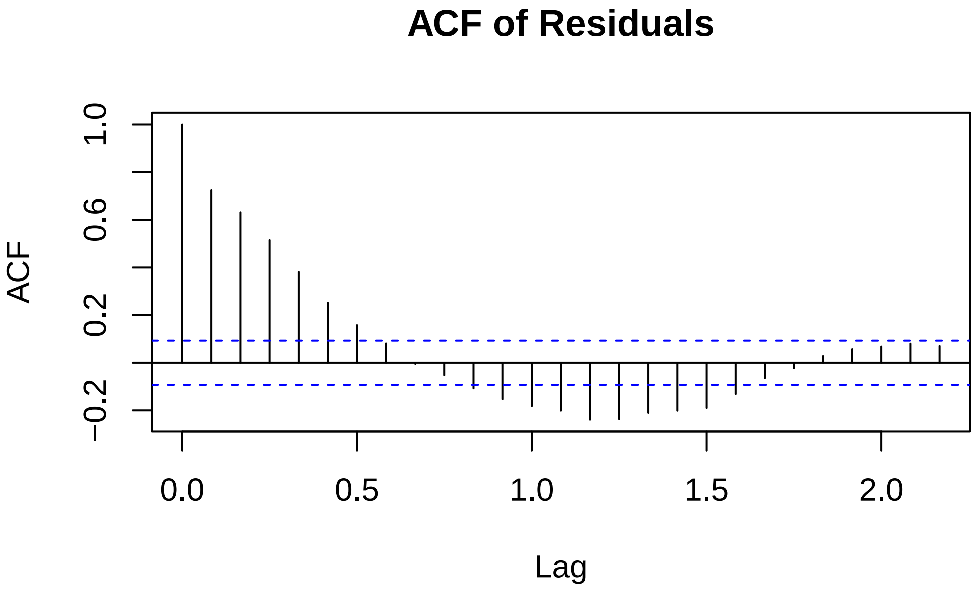
Part E

It’s reasonable to question the assumption that the residuals are white noise due to the evident autocorrelation at early lags in the ACF plot and the timeplot’s variance inconsistency. Instead, an AR model should be considered.

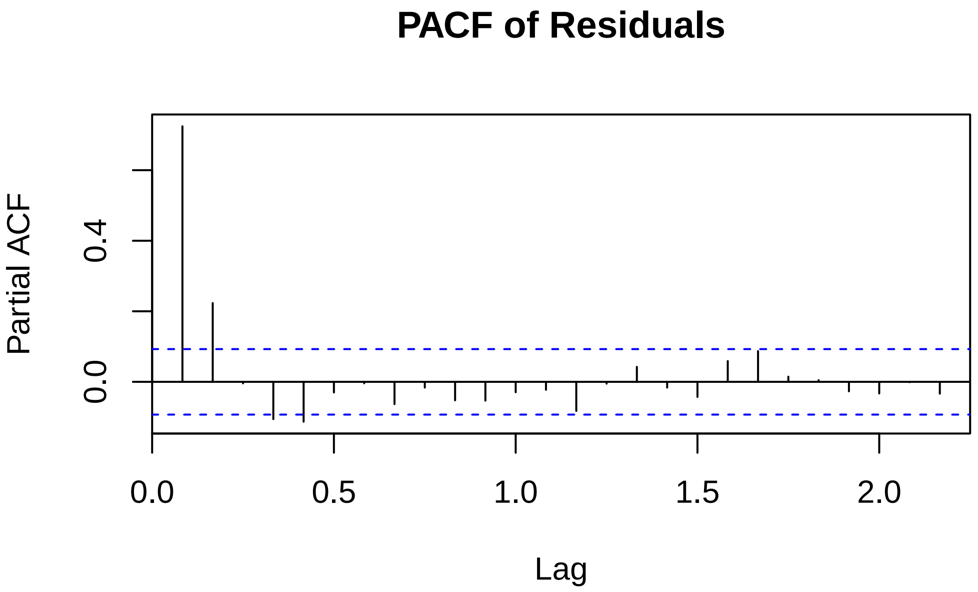
```
residuals <- residuals(fit1)
plot(residuals, type = 'l', main = "Residuals Timeplot", xlab = "Time", ylab = "Residuals")
```



```
acf(residuals, main = "ACF of Residuals")
```



```
pacf(residuals, main = "PACF of Residuals")
```


PACF of Residuals
Partial ACF
0.0
0.4
0.0
0.5
1.0
1.5
2.0
Lag

Part F

The model explains about 28.72% of the variance in recruitment, suggesting that additional variables or a more complex modeling approach could improve model fit and predictive power. I would consider applying an ARIMA model, which could provide a more accurate understanding of recruitment dynamics.

```r
soiL2 <- stats::lag(soi, -2)
soiL5 <- stats::lag(soi, -5)
fish_new <- ts.intersect(rec = rec, soiL2 = soiL2, soiL5 = soiL5, dframe = TRUE)
fit_new <- lm(rec ~ soiL2 + soiL5, data = fish_new)
summary(fit_new)
```

```

Call:
lm(formula = rec ~ soiL2 + soiL5, data = fish_new)

Residuals:
    Min      1Q  Median      3Q     Max 
-65.098 -18.756   2.793  16.175  57.083 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   65.025      1.166  55.759   <2e-16 ***
soiL2          5.548      3.005   1.846   0.0655 .  
soiL5        -40.223      3.017 -13.332   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.82 on 444 degrees of freedom
Multiple R-squared:  0.2872,	Adjusted R-squared:  0.284 
F-statistic: 89.44 on 2 and 444 DF,  p-value: < 2.2e-16
```