

LISTA 4

ANÁLISE DE SÉRIES TEMPORAIS

Tailine J. S. Nonato

2024-04-27

Descrição da atividade

Incluido as seguintes informações referentes a março de 2024, Energia = 419 e Dias = 33, utilizando a técnica de previsão com base na função de autocorrelação amostral, obtenha a previsão do consumo a ser faturado em abril de 2024, reproduzindo todos os elementos apresentados na aula 9.

Respostas

Carregando os pacotes necessários

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(tidyverse, readxl, knitr, latex2exp, pracma)
```

Leitura e manipulação dos dados

Os dados novos foram inseridos de forma mecânica no arquivo original da base de dados.

```
energia <- read_excel("ConsumoEnergiaEAgua.xlsx")
energia <- energia[,c(1,3,4)]
energia$consumo <- energia$Energia/energia$Dias
n.size <- length(energia$consumo)
kable(tail(energia), align = "c",
      caption = "Últimos registros da base de dados")
```

Table 1: Últimos registros da base de dados

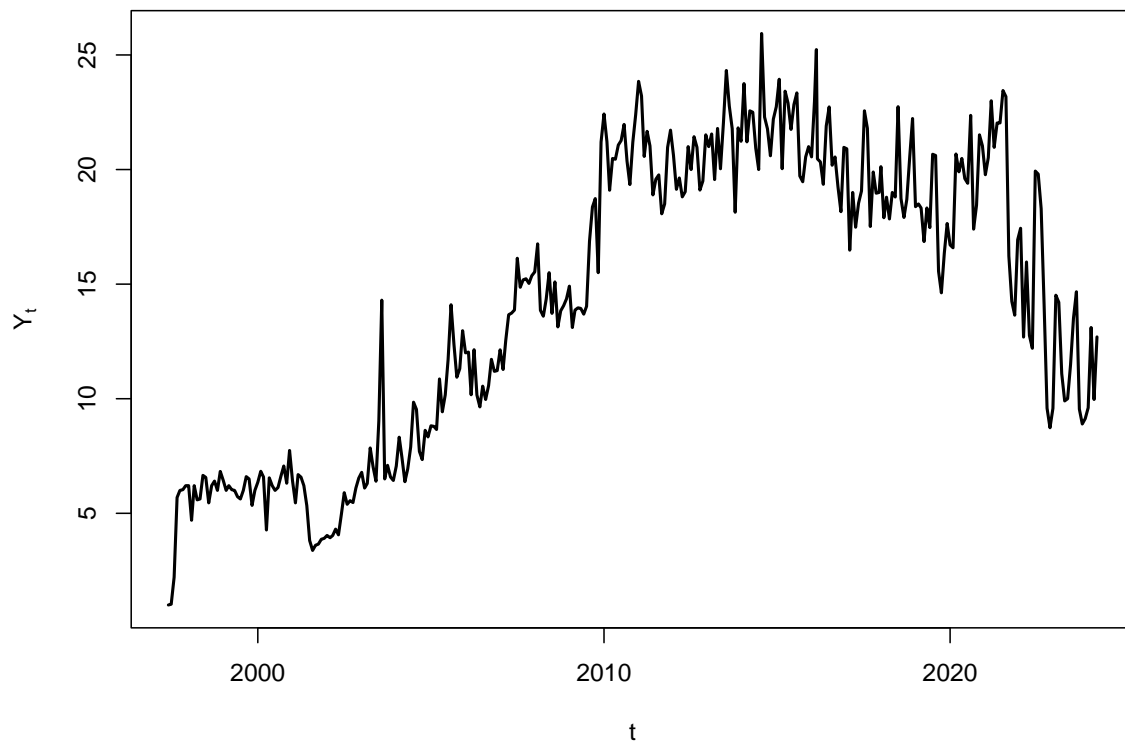
mes	Energia	Dias	consumo
2023-10-27	258	29	8.896552
2023-11-27	274	30	9.133333
2023-12-28	317	33	9.606061
2024-01-28	367	28	13.107143
2024-02-28	299	30	9.966667
2024-03-30	419	33	12.696970

Replicação dos itens da aula 9

1. Série temporal de jun/1997 a mar/2024

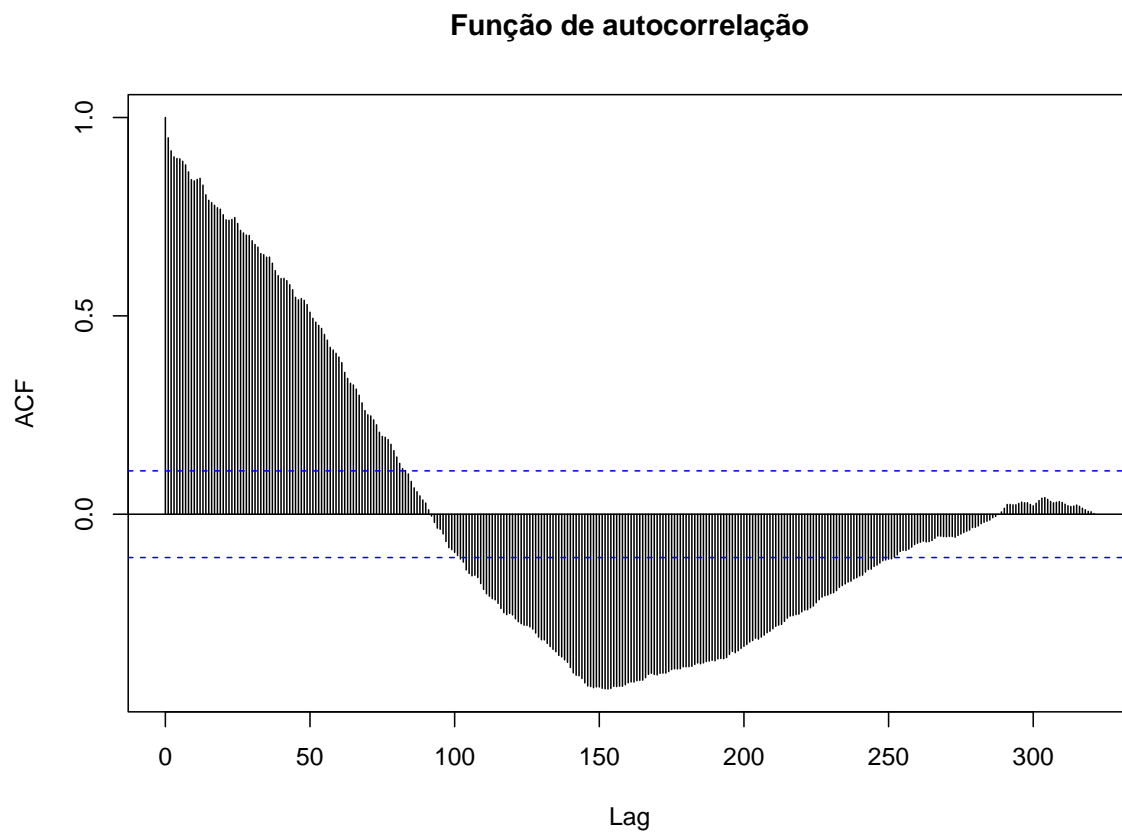
```
plot(energia$mes, energia$consumo, type = "l", lwd = 2,
     main = "Série temporal de jun/1997 a mar/2024",
     xlab = TeX(r'($t$)'),
     ylab = TeX(r'($Y_t$)'))
```

Série temporal de jun/1997 a mar/2024



2. Função de autocorrelação

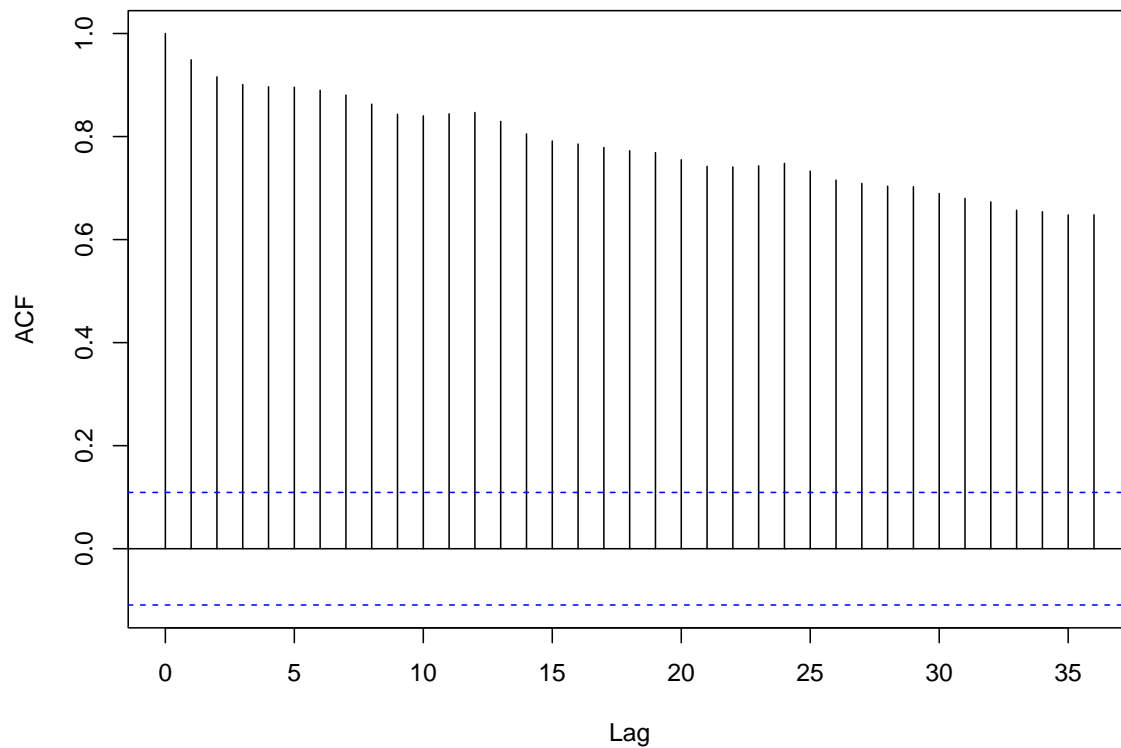
```
rho <- acf(energia$consumo, lag = n.size, plot = FALSE)
plot(rho, main = "Função de autocorrelação", xlab = "Lag", ylab = "ACF")
```



3. Função de autocorrelação

```
rho <- acf(energia$consumo, lag = 36, plot = FALSE)
plot(rho, main = "Função de autocorrelação com lag=36", xlab = "Lag", ylab = "ACF")
```

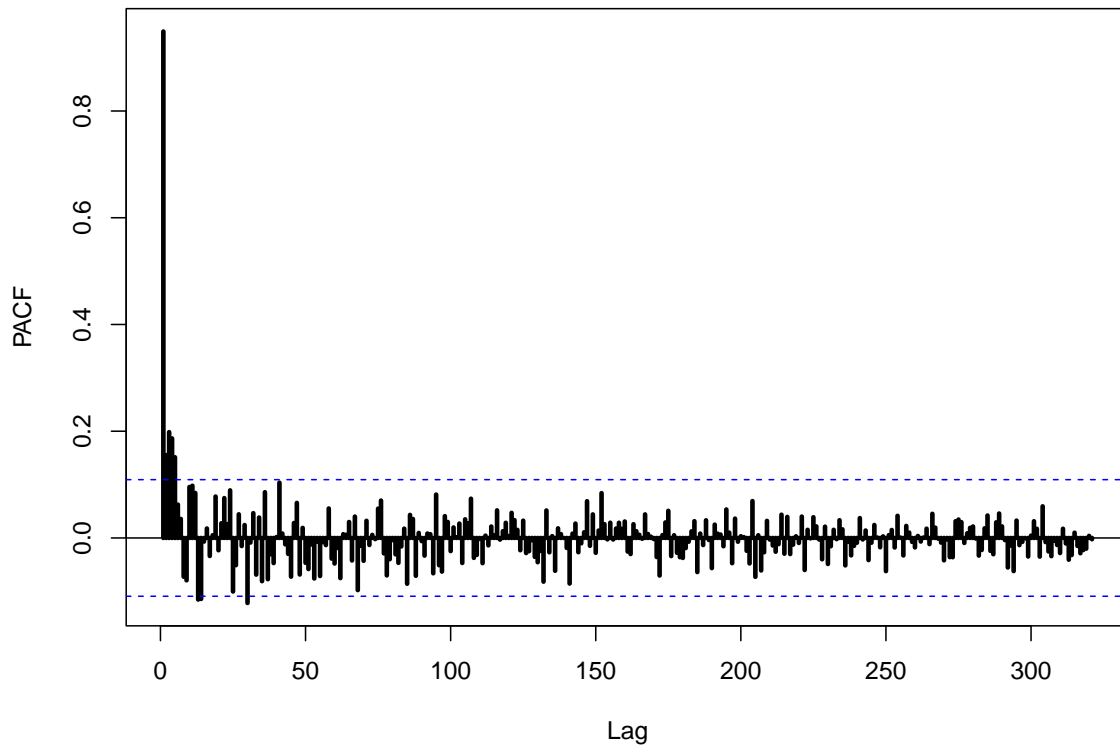
Função de autocorrelação com lag=36



4. Função de autocorrelação parcial

```
phi <- pacf(energia$consumo, lag = n.size, plot = FALSE)
plot(phi, main = "Função de autocorrelação parcial", xlab = "Lag", ylab = "PACF", lwd = 3)
```

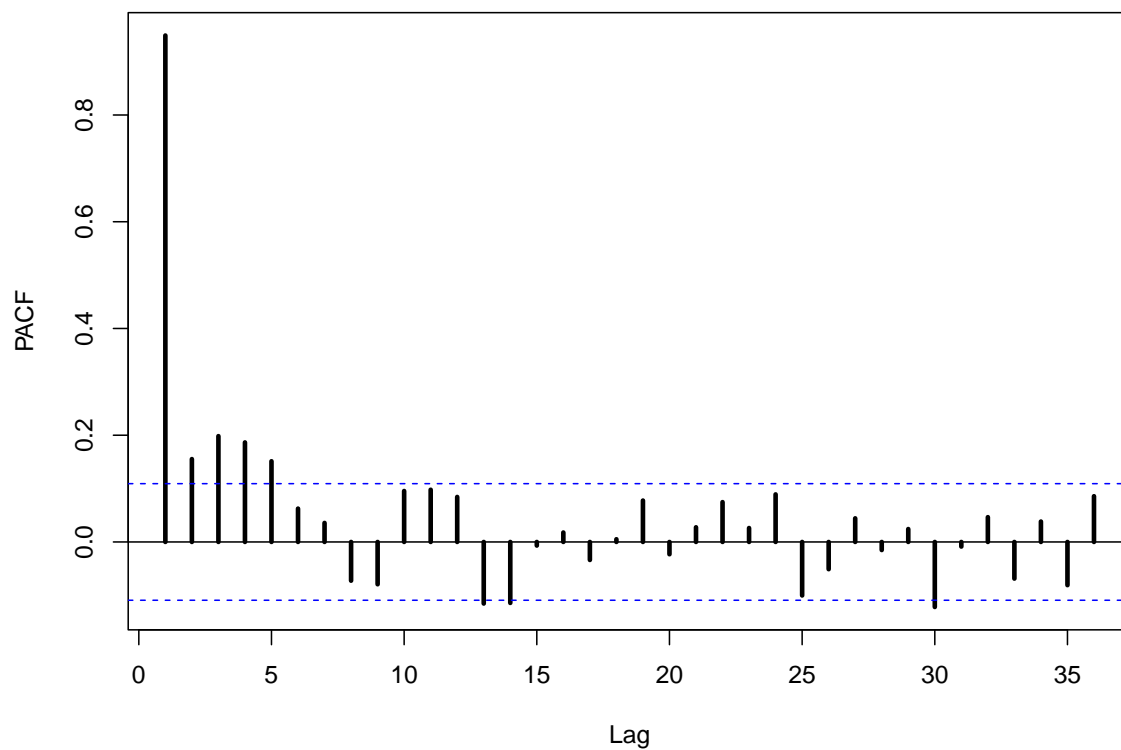
Função de autocorrelação parcial



5. Função de autocorrelação parcial

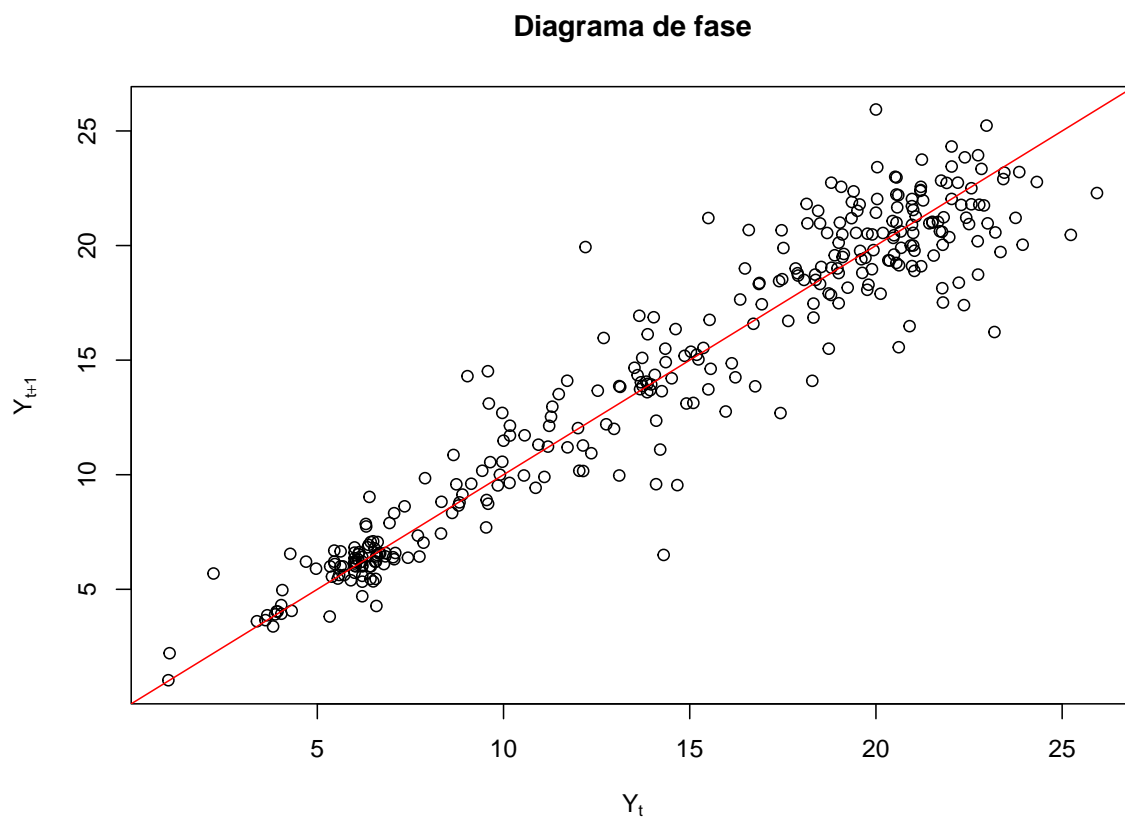
```
phi <- pacf(energia$consumo, lag = 36, plot = FALSE)
plot(phi, main = "Função de autocorrelação parcial com lag=36", xlab = "Lag", ylab = "PACF")
```

Função de autocorrelação parcial com lag=36



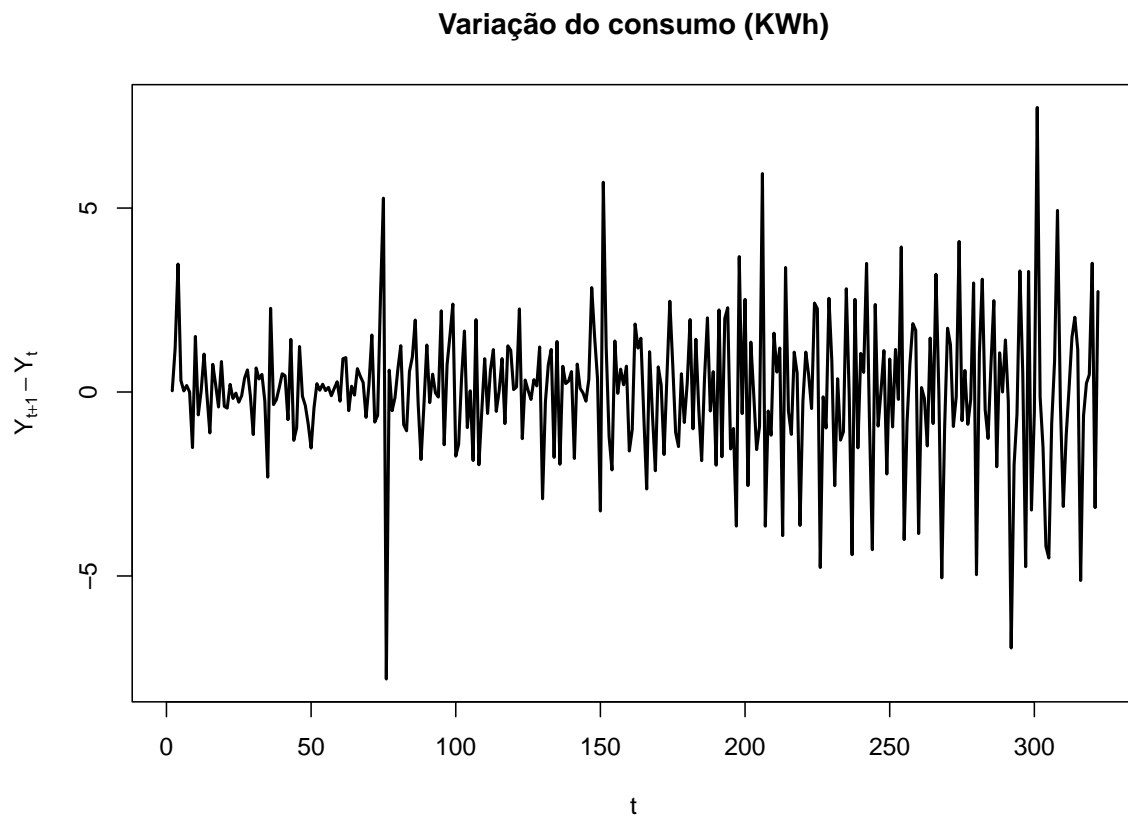
6. Diagrama de fase

```
energia$consumo_1 <- c(NA, energia$consumo[1:(n.size-1)])  
plot(energia$consumo_1, energia$consumo,  
     main = "Diagrama de fase",  
     xlab = TeX(r'($Y_t$)'),  
     ylab = TeX(r'($Y_{t+1}$)'),  
     abline(0, 1, col = "red"))
```



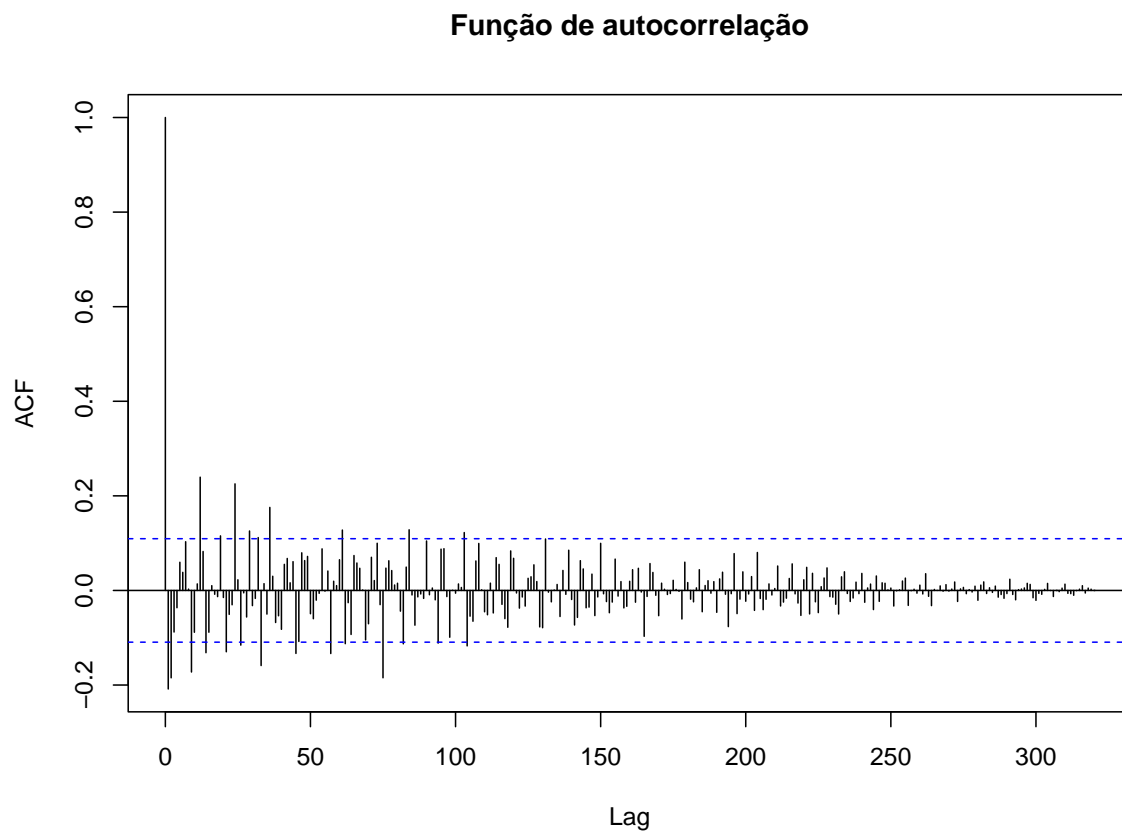
7. Variação do consumo (KWh)

```
energia$diff <- c(NA, diff(energia$consumo))
plot(energia$diff, type = "l", lwd = 2,
     main = "Variação do consumo (KWh)",
     xlab = TeX(r'($t$)'),
     ylab = TeX(r'($Y_{t+1} - Y_t$)'))
```

8. Função de autocorrelação da variação do consumo

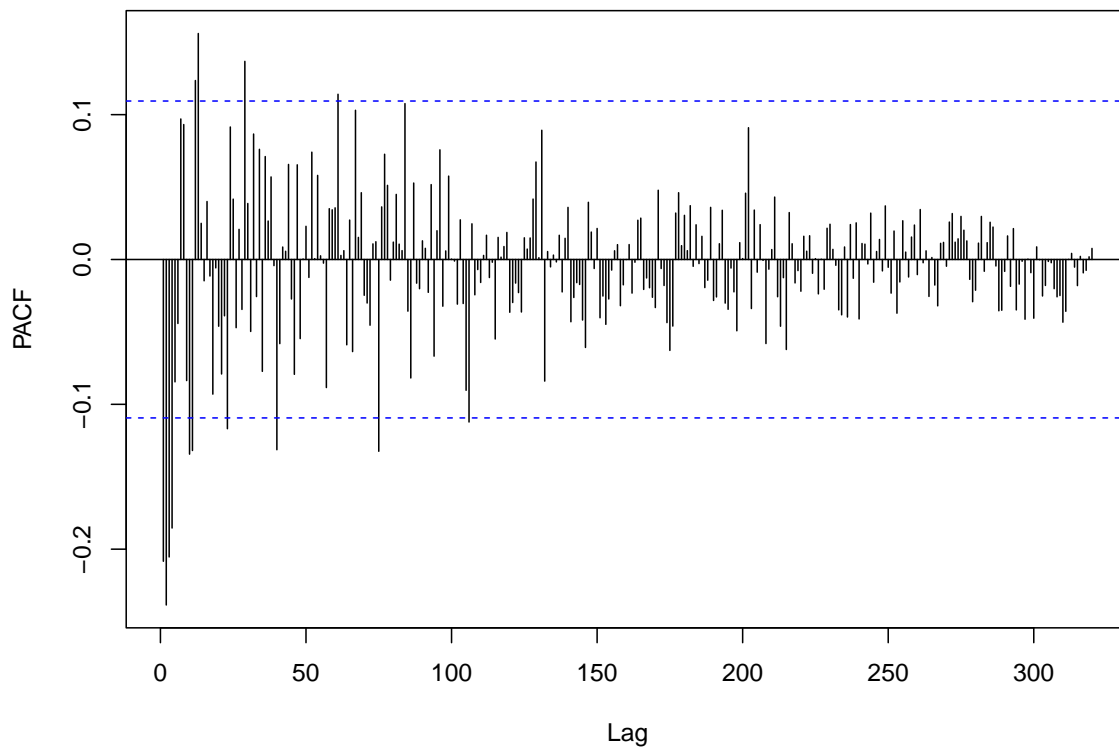
```
x <- na.omit(energia$diff)
rho <- acf(x, lag = length(x), plot = FALSE)
plot(rho, main = "Função de autocorrelação", xlab = "Lag", ylab = "ACF")
```



9. Função de autocorrelação parcial da variação do consumo

```
phi <- pacf(x, lag = length(x), plot = FALSE)
plot(phi, main = "Função de autocorrelação parcial", xlab = "Lag", ylab = "PACF")
```

Função de autocorrelação parcial



Modelos de Previsão

Utilizando a funções ar e predict

```
modelo <- ar(energia$consumo, aic = TRUE)
consumo <- c(energia$consumo, rep(NA, 1))
consumo[n.size+1] <- predict(modelo, n.ahead = 1)$pred
consumo[n.size+1]
```

```
[1] 11.00617
```

```
energia_abr_ar <- consumo[n.size+1] * 30
energia_abr_ar
```

```
[1] 330.1852
```

Assim, utilizando esse modelo, a energia a ser faturada em abril de 2024 prevista é de 330.19 KWh. Com consumo diário de 11.01 KWh.

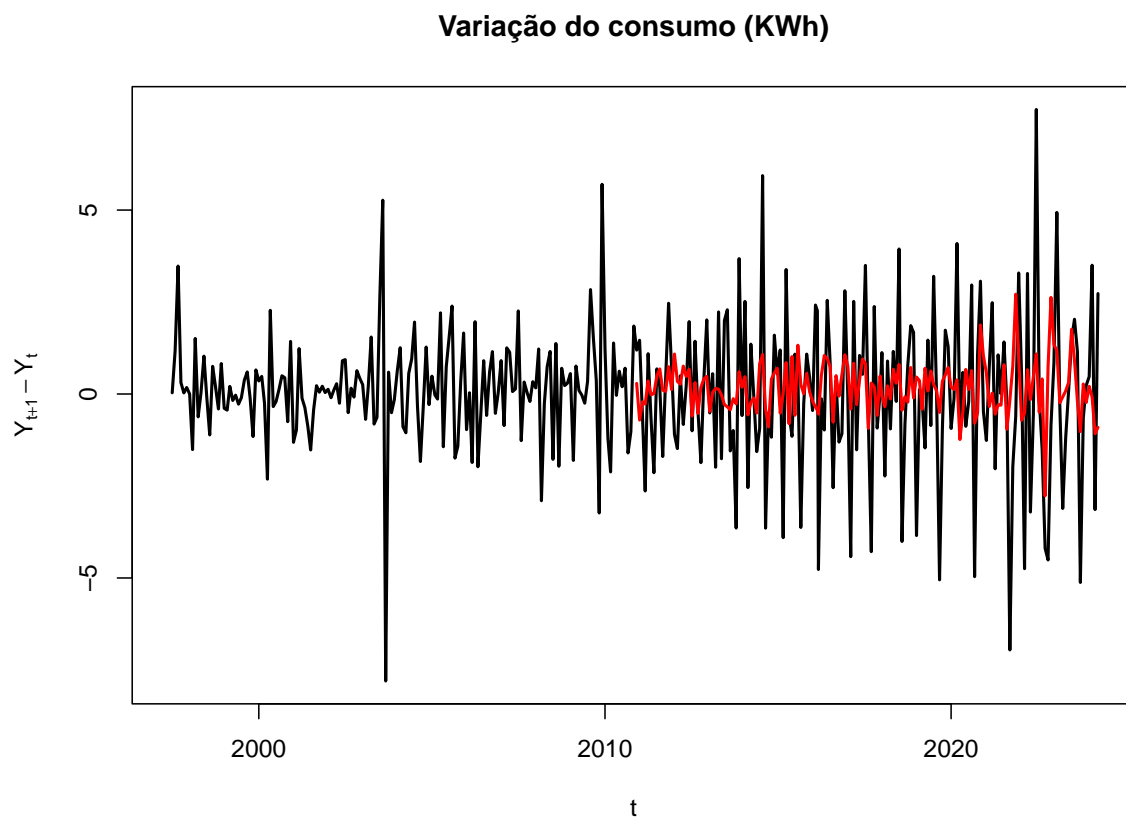
Reproduzindo aula 9

```
x <- na.omit(energia$diff)
n.size <- length(x)
n.training <- ceiling(n.size/2)
observed <- NULL
predicted <- NULL

for (t in (n.training+1):n.size) {
  x.training <- x[1:(t-1)]
  rho <- acf(x.training, lag = (t-1), plot = FALSE)
  last.lag <- length(rho$acf)
  Rho <- rho$acf
  Omega <- toeplitz(Rho[-last.lag])
  beta <- inv(Omega) %*% Rho[-1]
  beta.0 <- mean(x.training) * (1 - sum(beta))
  predicted[t] <- beta.0 + sum(rev(beta) * x.training[-1])
  observed[t] <- x[t]
}

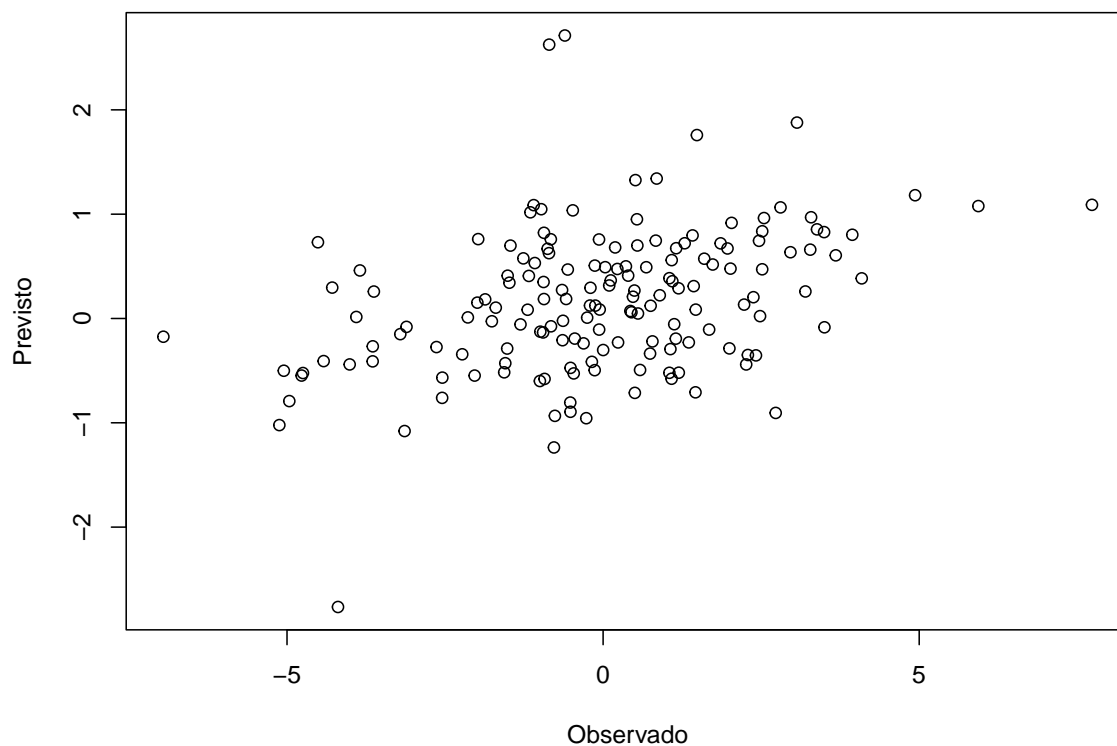
energia$predicted <- c(NA, predicted)

plot(energia$mes, energia$diff, type = "l", lwd = 2,
     main = "Variação do consumo (KWh)",
     xlab = TeX(r'($t$)'),
     ylab = TeX(r'($Y_{t+1} - Y_t$)'))
lines(energia$mes, energia$predicted, col = "red", lwd = 2)
```



```
plot(observed, predicted, main = "Linearidade entre observado e previsto",  
      xlab = "Observado", ylab = "Previsto")
```

Linearidade entre observado e previsto



```
# correlation  
cor(observed, predicted, use = "complete.obs")
```

```
[1] 0.3829436
```

```
# mean absolute error  
mean(abs(na.omit(observed) - na.omit(predicted)))
```

```
[1] 1.657283
```

Tendo como base os exemplos em sala, busca-se aprimorar essas métricas com \hat{Y}

```

Y.hat <- NULL
Y.t <- NULL

for (h in (n.training+1):n.size){
  Y.t[h+1] <- energia$consumo[h] + energia$diff[h+1]
  Y.hat[h+1] <- energia$consumo[h] + energia$predicted[h+1]
}

# correlation
cor(Y.t, Y.hat, use = "complete.obs")

```

```
[1] 0.829377
```

```

# mean absolute error
mean(abs(na.omit(Y.t) - na.omit(Y.hat)))

```

```
[1] 1.657283
```

```

# mean absolute percentage error
mean(abs(na.omit(Y.t) - Y.hat)/na.omit(Y.t))

```

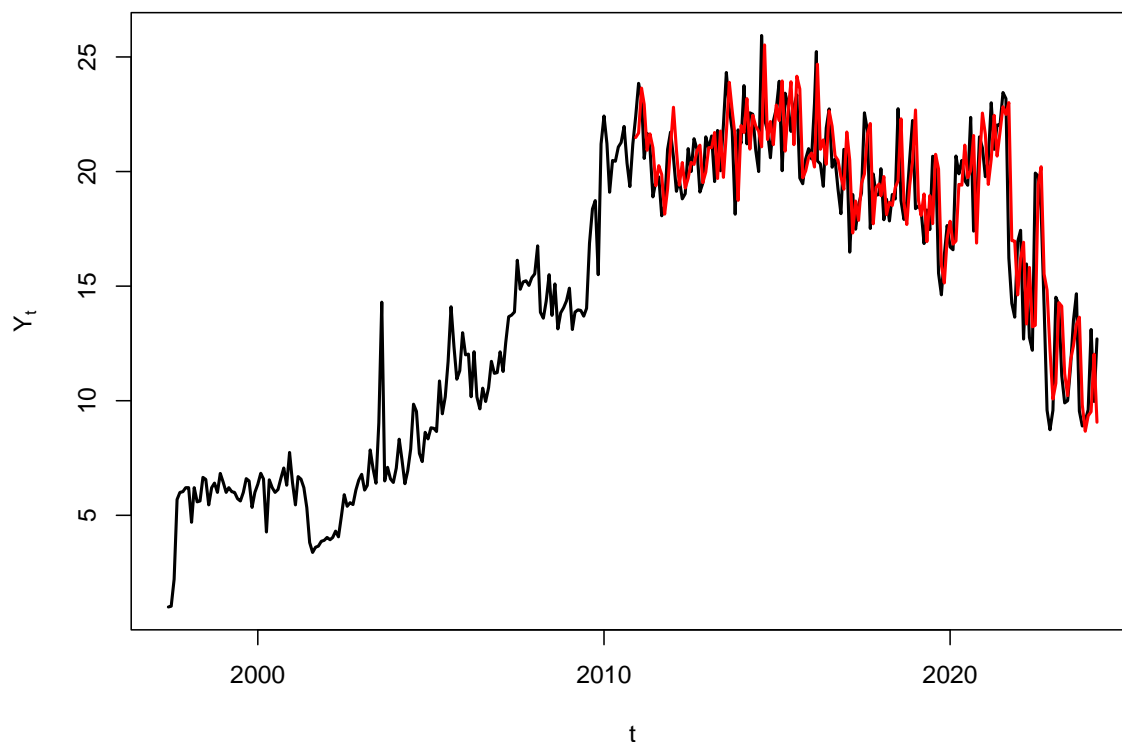
```
[1] 0.09530472
```

```

plot(energia$mes, energia$consumo, type = "l", lwd = 2,
     main = "Série temporal de fev/1997 a mar/2024 com valores previstos",
     xlab = TeX(r'($t$)'),
     ylab = TeX(r'($Y_t$)'))
lines(energia$mes, Y.hat, col = "red", lwd = 2)

```

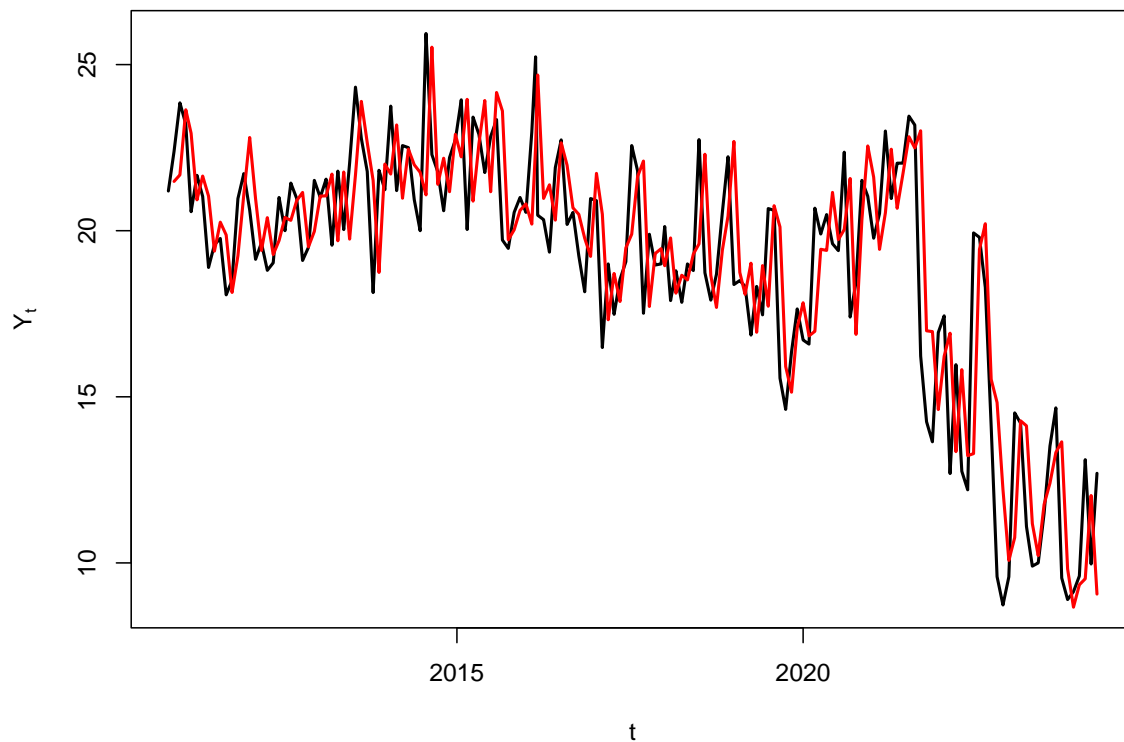
Série temporal de fev/1997 a mar/2024 com valores previstos



Fazendo um “zoom” para os dados de teste:

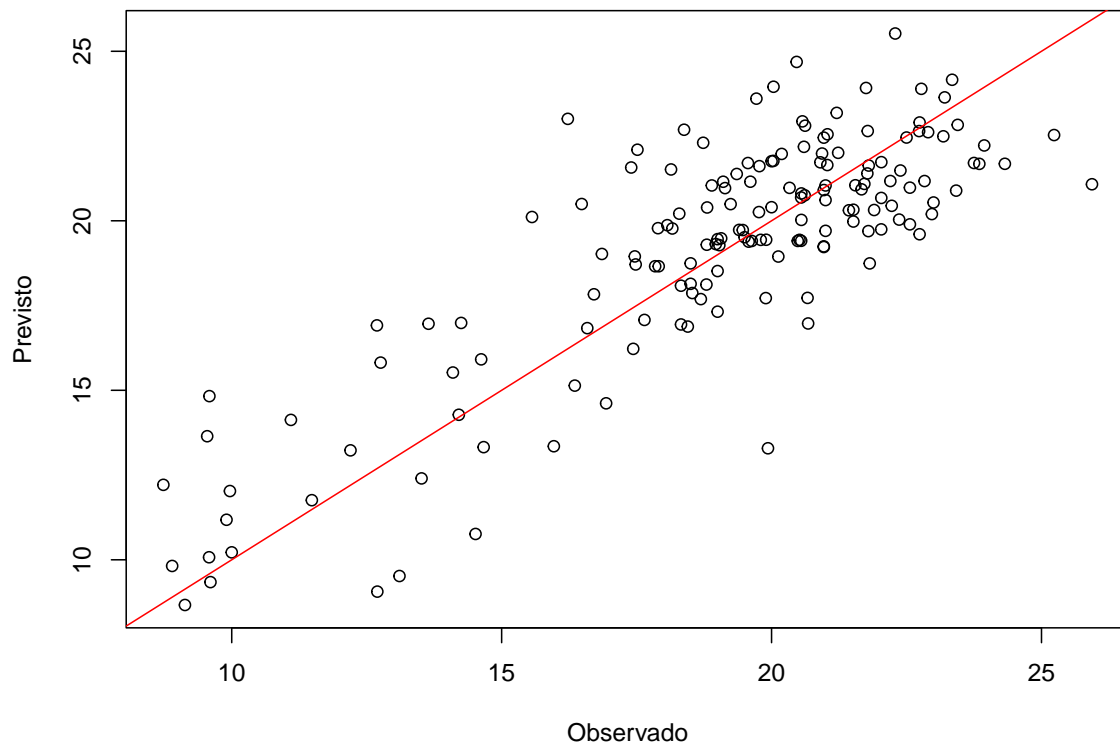
```
plot(energia$mes[162:322], energia$consumo[162:322], type = "l", lwd = 2,
     main = "Série temporal de nov/2010 a mar/2024 com valores previstos",
     xlab = TeX(r'($t$)'),
     ylab = TeX(r'($Y_t$)'))
lines(energia$mes[162:322], Y.hat[162:322], col = "red", lwd = 2)
```


Série temporal de nov/2010 a mar/2024 com valores previstos



```
plot(Y.t, Y.hat, main = "Linearidade entre observado e previsto",  
      xlab = "Observado", ylab = "Previsto")  
abline(0, 1, col = "red")
```

Linearidade entre observado e previsto



```
cor(Y.t, Y.hat, use = "complete.obs")
```

```
[1] 0.829377
```

```
shapiro.test(na.omit(Y.t - Y.hat))
```

Shapiro-Wilk normality test

```
data:  na.omit(Y.t - Y.hat)  
W = 0.99205, p-value = 0.5203
```

```
sd(na.omit(Y.t - Y.hat))
```

```
[1] 2.131004
```

```
Y.next <- Y.hat[n.size] + 2*sd(na.omit(Y.t - Y.hat))
Y.next
```

```
[1] 16.2901
```

```
energia_abr_acf <- Y.next * 30
energia_abr_acf
```

```
[1] 488.7029
```

Assim, utilizando esse modelo, a energia a ser faturada em abril de 2024 prevista é de 488.7 KWh. Com consumo diário de 16.29 KWh.

Comparando os modelos

Se colocarmos em uma tabela:

```
comp_modelos <- data.frame(Função = c("ar", "acf"),
                           Consumo = c(round(consumo[n.size+1],2), round(Y.next,2)),
                           Energia = c(round(energia_abr_ar,2), round(energia_abr_acf,2)))
kable(comp_modelos, align = "c",
      caption = "Previsões de consumo diário e do mês para abril de 2024")
```

Table 2: Previsões de consumo diário e do mês para abril de 2024

Função	Consumo	Energia
ar	12.70	330.19
acf	16.29	488.70

Conclusão

Adicionando o valor previsto para abril utilizando o modelo com a função ACF, é possível fazer a série temporal novamente:

```
energia$consumo[n.size+1] <- Y.next
energia$mes[n.size+1] <- "2024-04-30"
plot(energia$mes, energia$consumo, type = "l", lwd = 2,
     main = "Série temporal de jun/1997 ao que é previsto para abr/2024",
     xlab = TeX(r'($t$)'),
     ylab = TeX(r'($Y_t$)'))
```

Série temporal de jun/1997 ao que é previsto para abr/2024

