Centro de Estatística Aplicada

Gustavo Kanno[1]
Rodrigo Marcel Araujo[2]
Victor Ribeiro Baião Decanini[3]

Junho de 2021

Sumário

[1] Número USP: 9795810
[2] Número USP: 9299208
[3] Número USP: 9790502

```r
library(forecast)
library(randtests)
```

```
##
## Attaching package: 'randtests'

## The following object is masked from 'package:tseries':
##
##     runs.test
```

```r
library(zoo)
library(TSA)
```

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

##
## Attaching package: 'TSA'

## The following object is masked from 'package:GeneCycle':
##
##     periodogram

## The following object is masked from 'package:readr':
##
##     spec

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```r
library(gridExtra)
library(FitAR)
```

```
## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:faraway':
##
##     melanoma
```

```
## Loading required package: leaps

## Loading required package: ltsa

## Loading required package: bestglm

##
## Attaching package: 'FitAR'

## The following object is masked from 'package:forecast':
##
##     BoxCox

## The following object is masked from 'package:car':
##
##     Boot
```

```r
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-1
```

```r
library(islasso)
library(astsa)
```

```
##
## Attaching package: 'astsa'

## The following objects are masked from 'package:fma':
##
##     chicken, sales

## The following object is masked from 'package:forecast':
##
##     gas

## The following object is masked from 'package:fpp2':
##
##     oil
```

```
## The following object is masked from 'package:faraway':
##
##     star
```

```
## The following object is masked from 'package:gamlss.data':
##
##     oil
```

```
library(lmtest)
library(forecast)
```

Análise das séries temporais mensais

Análise Descritiva

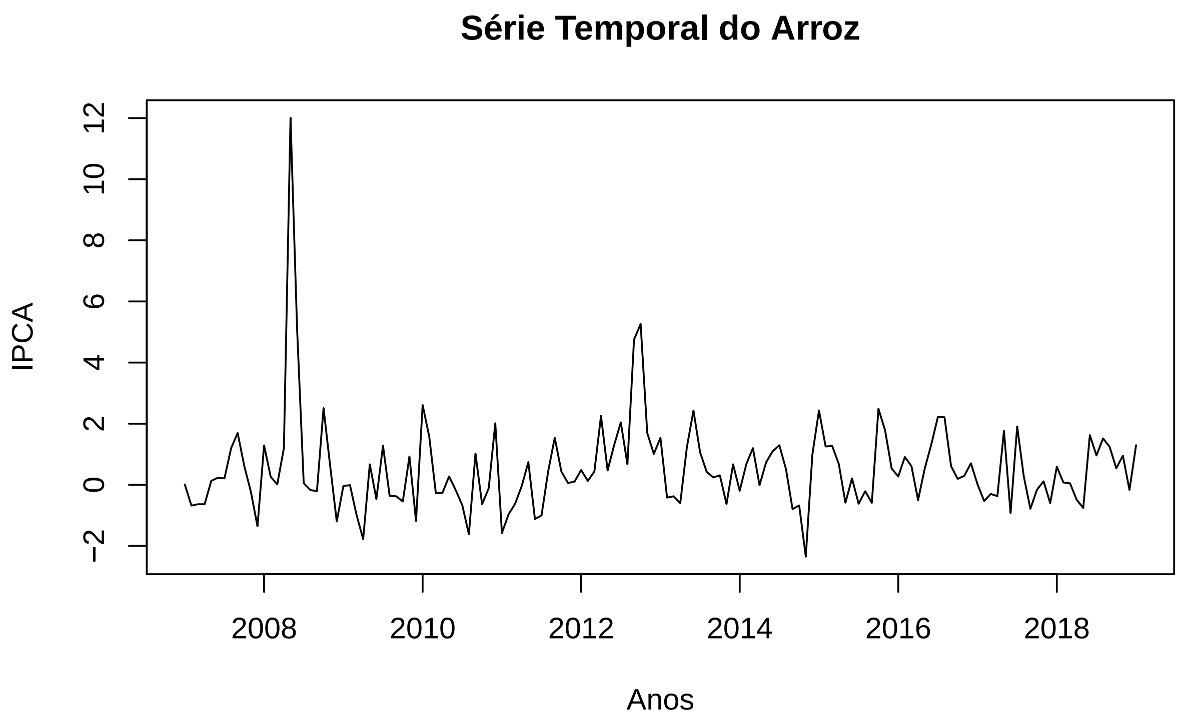
```
data$Data <- as.Date(data$Data)
head(data)
```

```
## # A tibble: 6 x 24
##   Data        Arroz `Avicultura de ~ `Avicultura de ~ Banana Batata
##   <date>      <dbl>            <dbl>            <dbl>  <dbl>  <dbl>
## 1 2007-01-01  0.01             0.295             3.43  -2.86   0.75
## 2 2007-02-01 -0.68             1.71              2.82  -1.62  -3.83
## 3 2007-03-01 -0.635            2.26             10.1    1.05   7.61
## 4 2007-04-01 -0.635           -0.56              1.31  -2.65  36.4
## 5 2007-05-01  0.13            -0.13             -1.11  -1.46  11.6
## 6 2007-06-01  0.230            0.27              4.93  -1.07  -5.17
## # ... with 18 more variables: Bovinocultura <dbl>, `Cacau e produtos` <dbl>,
## #   Café <dbl>, Cebola <dbl>, `Complexo soja` <dbl>, `Complexo
## #   sucroalc.` <dbl>, Feijão <dbl>, Frutas <dbl>, Hortícolas <dbl>,
## #   Indefinido <dbl>, `Laranja e citros` <dbl>, Lácteos <dbl>, Mandioca <dbl>,
## #   Milho <dbl>, Pescado <dbl>, Suinocultura <dbl>, Tomate <dbl>, Trigo <dbl>
```

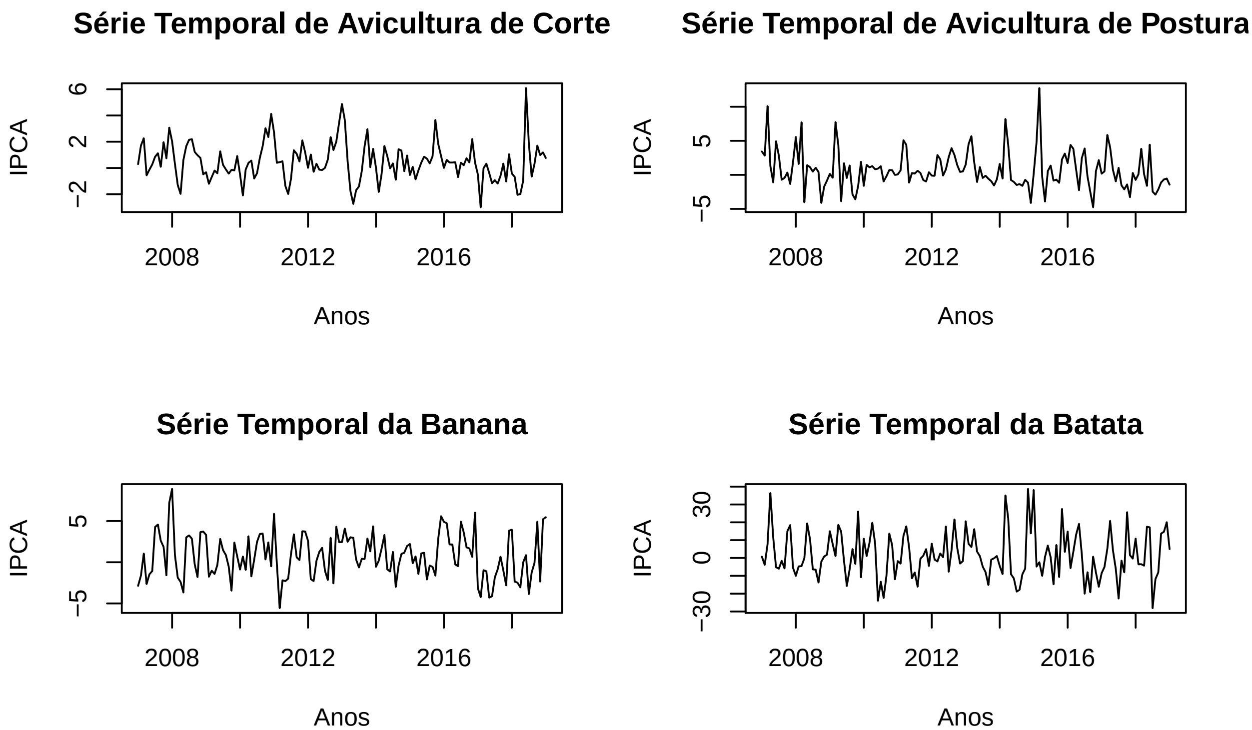
```
zt2 <- ts(data[,2], frequency = 12, start = 2007, end = 2019)
zt3 <- ts(data[,3], frequency = 12, start = 2007, end = 2019)
zt4 <- ts(data[,4], frequency = 12, start = 2007, end = 2019)
zt5 <- ts(data[,5], frequency = 12, start = 2007, end = 2019)
zt6 <- ts(data[,6], frequency = 12, start = 2007, end = 2019)
zt7 <- ts(data[,7], frequency = 12, start = 2007, end = 2019)
zt8 <- ts(data[,8], frequency = 12, start = 2007, end = 2019)
zt9 <- ts(data[,9], frequency = 12, start = 2007, end = 2019)
zt10 <- ts(data[,10], frequency = 12, start = 2007, end = 2019)
zt11 <- ts(data[,11], frequency = 12, start = 2007, end = 2019)

zt12 <- ts(data[,12], frequency = 12, start = 2007, end = 2019)
zt13 <- ts(data[,13], frequency = 12, start = 2007, end = 2019)
zt14 <- ts(data[,14], frequency = 12, start = 2007, end = 2019)
zt15 <- ts(data[,15], frequency = 12, start = 2007, end = 2019)
```

```r
zt16 <- ts(data[,16], frequency = 12, start = 2007, end = 2019)
zt17 <- ts(data[,17], frequency = 12, start = 2007, end = 2019)
zt18 <- ts(data[,18], frequency = 12, start = 2007, end = 2019)
zt19 <- ts(data[,19], frequency = 12, start = 2007, end = 2019)
zt20 <- ts(data[,20], frequency = 12, start = 2007, end = 2019)
zt21 <- ts(data[,21], frequency = 12, start = 2007, end = 2019)
zt22 <- ts(data[,22], frequency = 12, start = 2007, end = 2019)
zt23 <- ts(data[,23], frequency = 12, start = 2007, end = 2019)
zt24 <- ts(data[,24], frequency = 12, start = 2007, end = 2019)
```

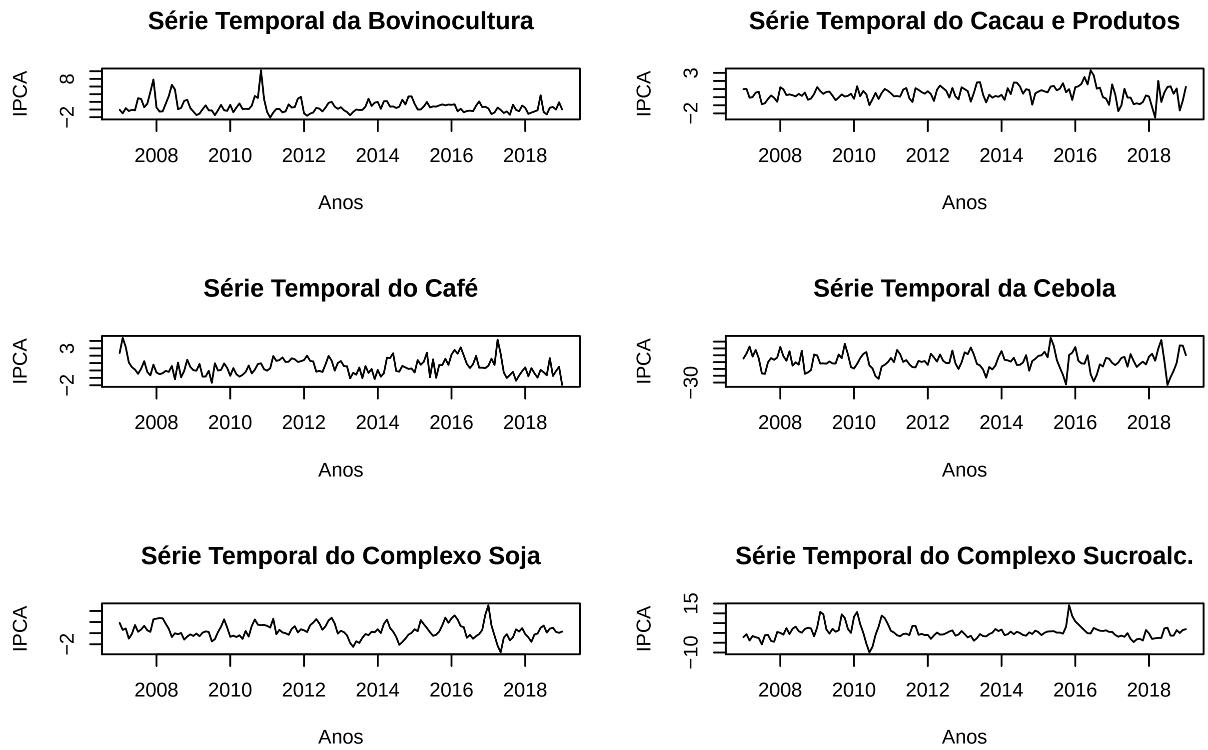
```r
plot(zt2,main="Série Temporal do Arroz", xlab= "Anos", ylab="IPCA")
```



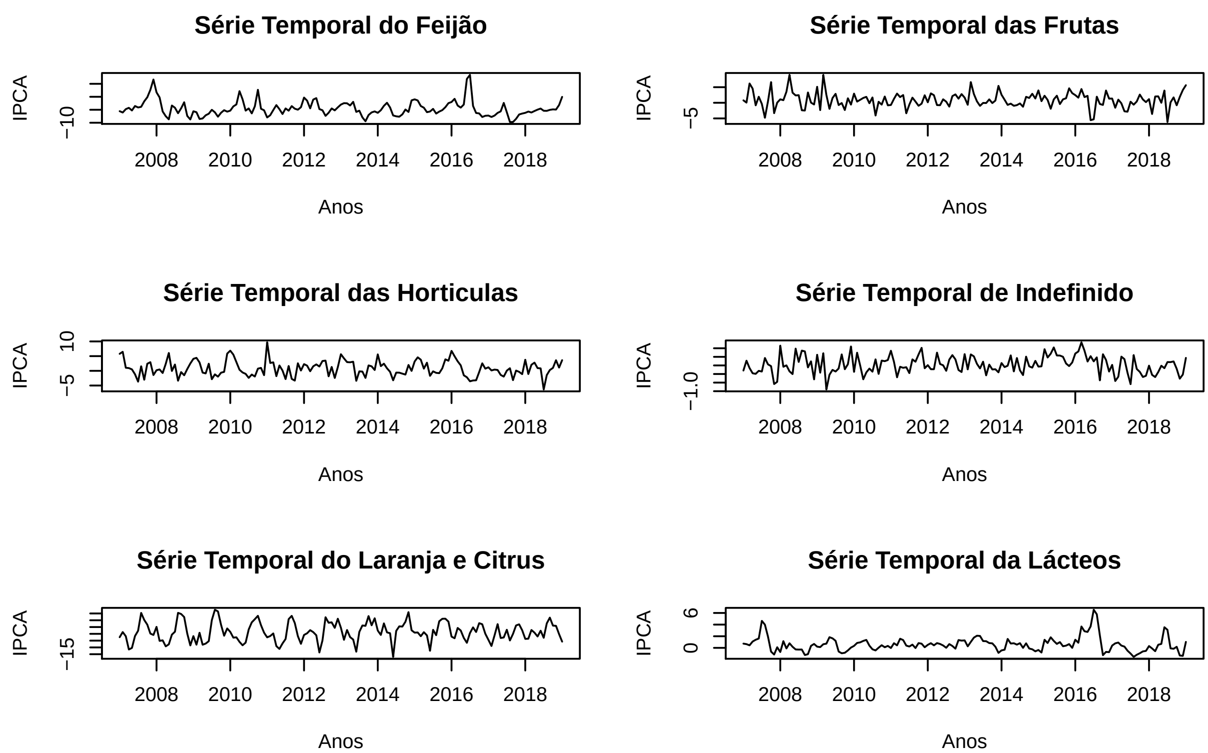

```r
par(mfrow = c(2, 2))
plot(zt3,main="Série Temporal de Avicultura de Corte", xlab= "Anos", ylab="IPCA")
plot(zt4,main="Série Temporal de Avicultura de Postura", xlab= "Anos", ylab="IPCA")
plot(zt5,main="Série Temporal da Banana", xlab= "Anos", ylab="IPCA")
plot(zt6,main="Série Temporal da Batata", xlab= "Anos", ylab="IPCA")
```



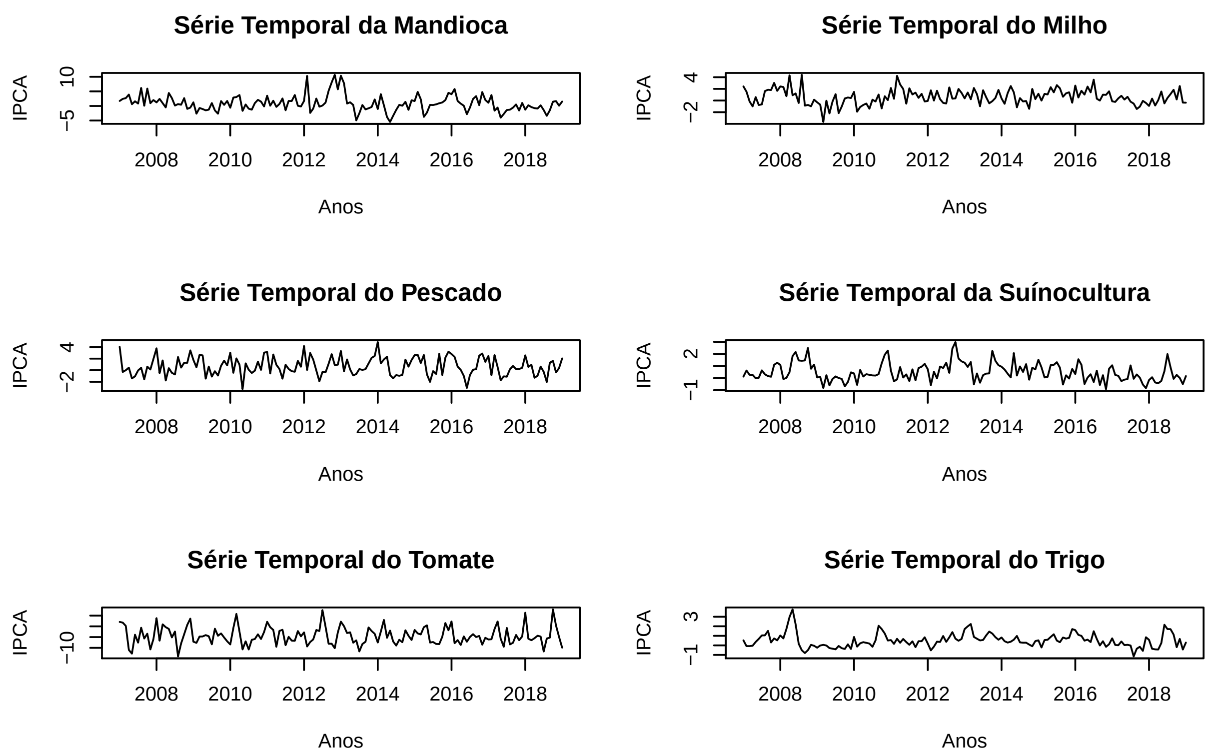

```r
par(mfrow = c(3, 2))


plot(zt7,main="Série Temporal da Bovinocultura", xlab= "Anos", ylab="IPCA")
plot(zt8,main="Série Temporal do Cacau e Produtos", xlab= "Anos", ylab="IPCA")
plot(zt9,main="Série Temporal do Café", xlab= "Anos", ylab="IPCA")
plot(zt10,main="Série Temporal da Cebola", xlab= "Anos", ylab="IPCA")
plot(zt11,main="Série Temporal do Complexo Soja", xlab= "Anos", ylab="IPCA")
plot(zt12,main="Série Temporal do Complexo Sucroalc.", xlab= "Anos", ylab="IPCA")
```



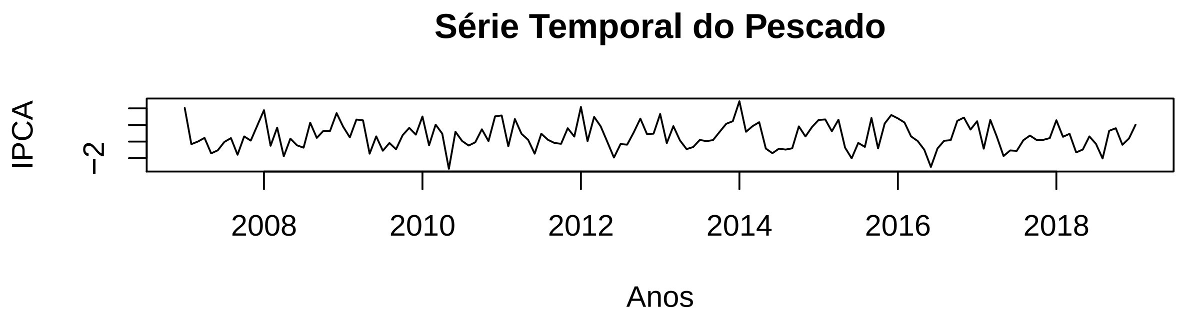

```r
par(mfrow = c(3, 2))


plot(zt13,main="Série Temporal do Feijão", xlab= "Anos", ylab="IPCA")
plot(zt14,main="Série Temporal das Frutas", xlab= "Anos", ylab="IPCA")
plot(zt15,main="Série Temporal das Horticulas", xlab= "Anos", ylab="IPCA")
plot(zt16,main="Série Temporal de Indefinido", xlab= "Anos", ylab="IPCA")
plot(zt17,main="Série Temporal do Laranja e Citrus", xlab= "Anos", ylab="IPCA")
plot(zt18,main="Série Temporal da Lácteos", xlab= "Anos", ylab="IPCA")
```

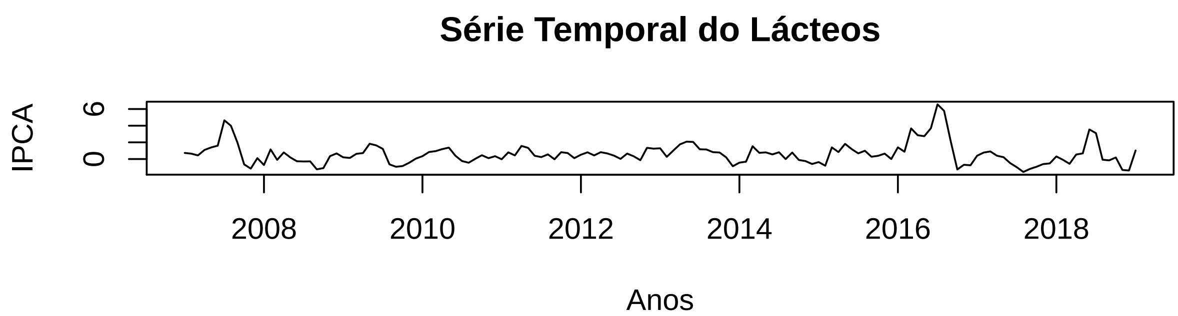



```r
par(mfrow = c(3, 2))


plot(zt19,main="Série Temporal da Mandioca", xlab= "Anos", ylab="IPCA")
plot(zt20,main="Série Temporal do Milho", xlab= "Anos", ylab="IPCA")
plot(zt21,main="Série Temporal do Pescado", xlab= "Anos", ylab="IPCA")
plot(zt22,main="Série Temporal da Suínocultura", xlab= "Anos", ylab="IPCA")
plot(zt23,main="Série Temporal do Tomate", xlab= "Anos", ylab="IPCA")
plot(zt24,main="Série Temporal do Trigo", xlab= "Anos", ylab="IPCA")
```



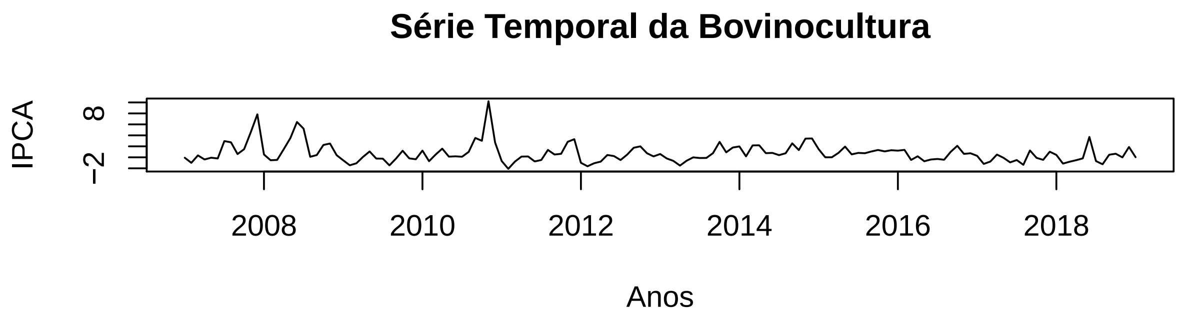

```r
par(mfrow = c(2, 1))
plot(zt21,main="Série Temporal do Pescado", xlab= "Anos", ylab="IPCA")
plot(zt18,main="Série Temporal do Lácteos", xlab= "Anos", ylab="IPCA")
```

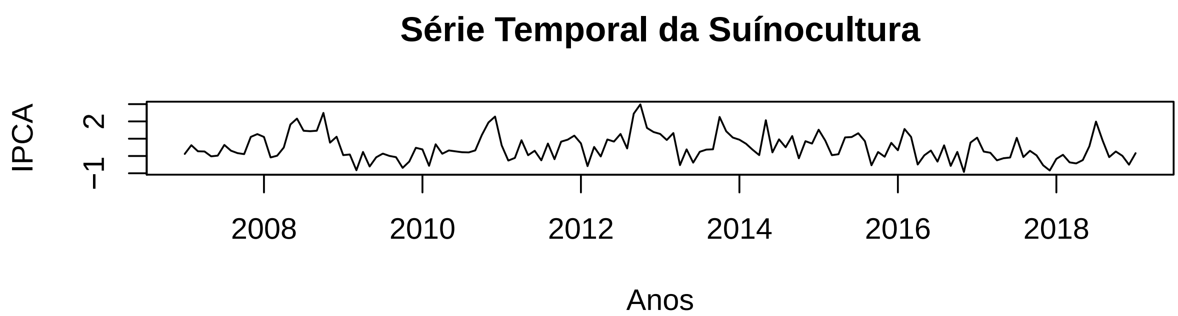




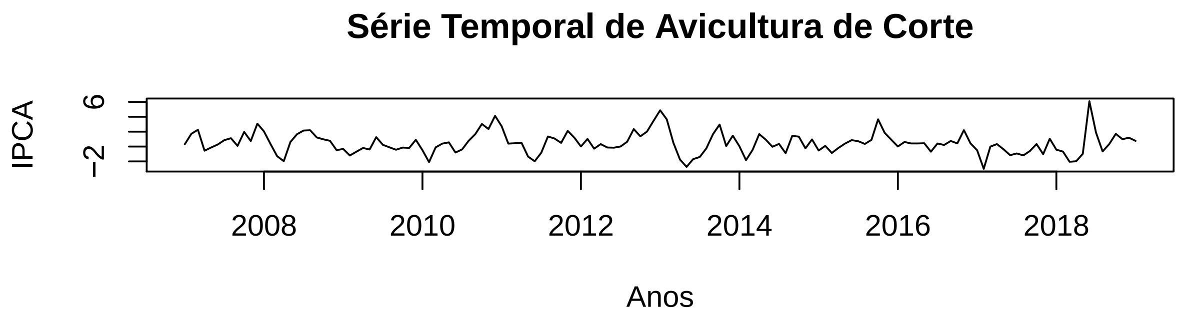

```
#900#650
```

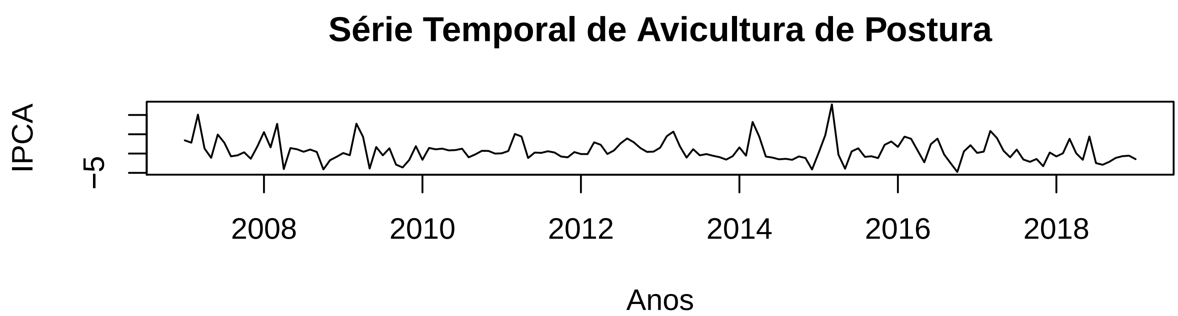
```
par(mfrow = c(2, 1))
plot(zt7,main="Série Temporal da Bovinocultura", xlab= "Anos", ylab="IPCA")
plot(zt22,main="Série Temporal da Suínocultura", xlab= "Anos", ylab="IPCA")
```






```r
par(mfrow = c(2, 1))
plot(zt3,main="Série Temporal de Avicultura de Corte", xlab= "Anos", ylab="IPCA")
plot(zt4,main="Série Temporal de Avicultura de Postura", xlab= "Anos", ylab="IPCA")
```

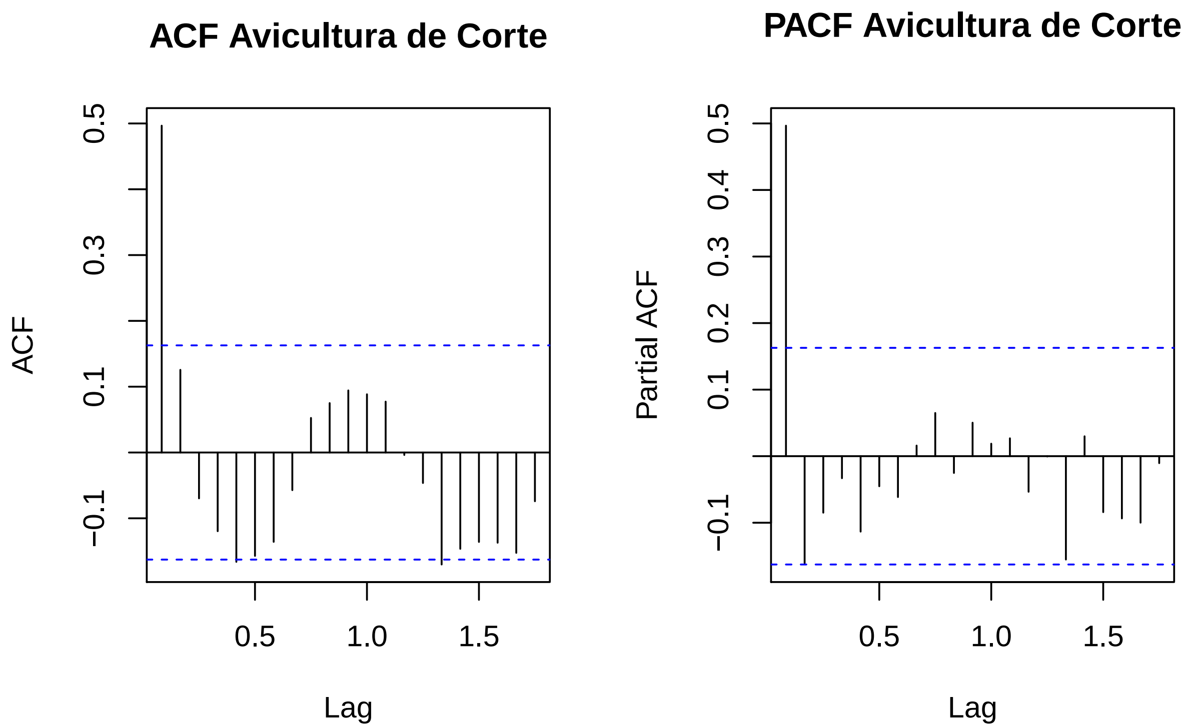




Funções de Autocorrelações

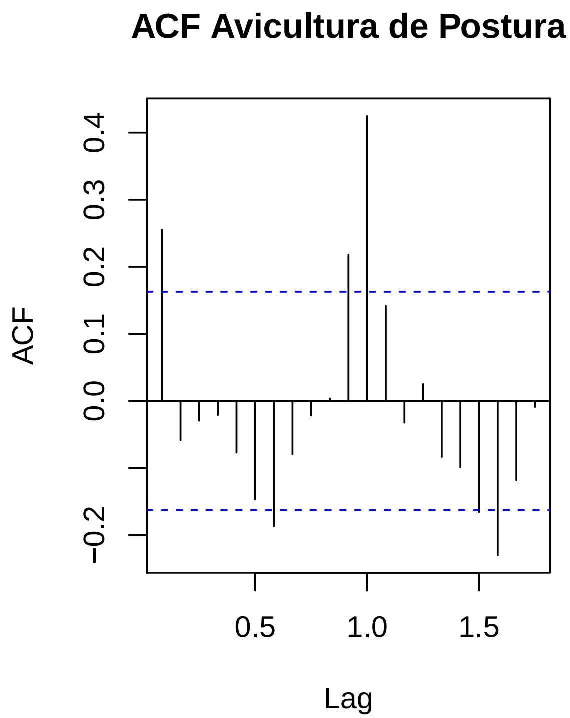
Funções de Autocorrelações para Avicultura de Corte

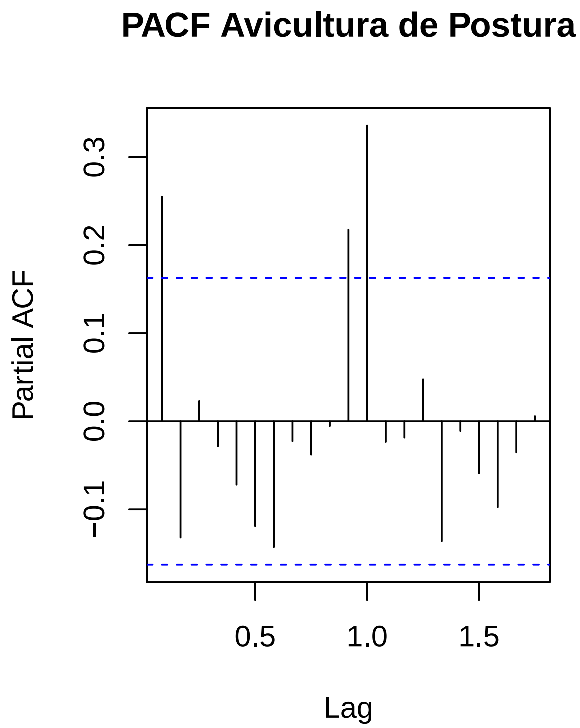
```
par(mfrow = c(1, 2))
acf(zt3, main="ACF Avicultura de Corte")
pacf(zt3, main="PACF Avicultura de Corte")
```



Funções de Autocorrelações para Avicultura de Postura

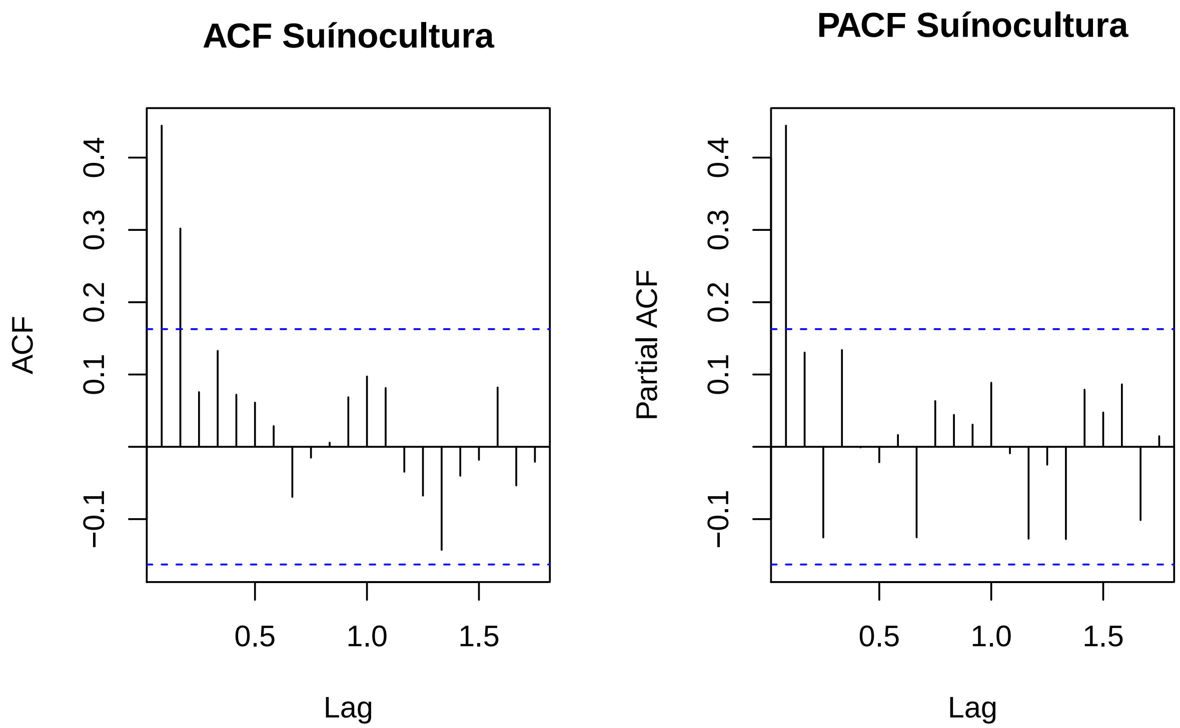
```
par(mfrow = c(1, 2))
acf(zt4, main="ACF Avicultura de Postura")
pacf(zt4, main="PACF Avicultura de Postura")
```





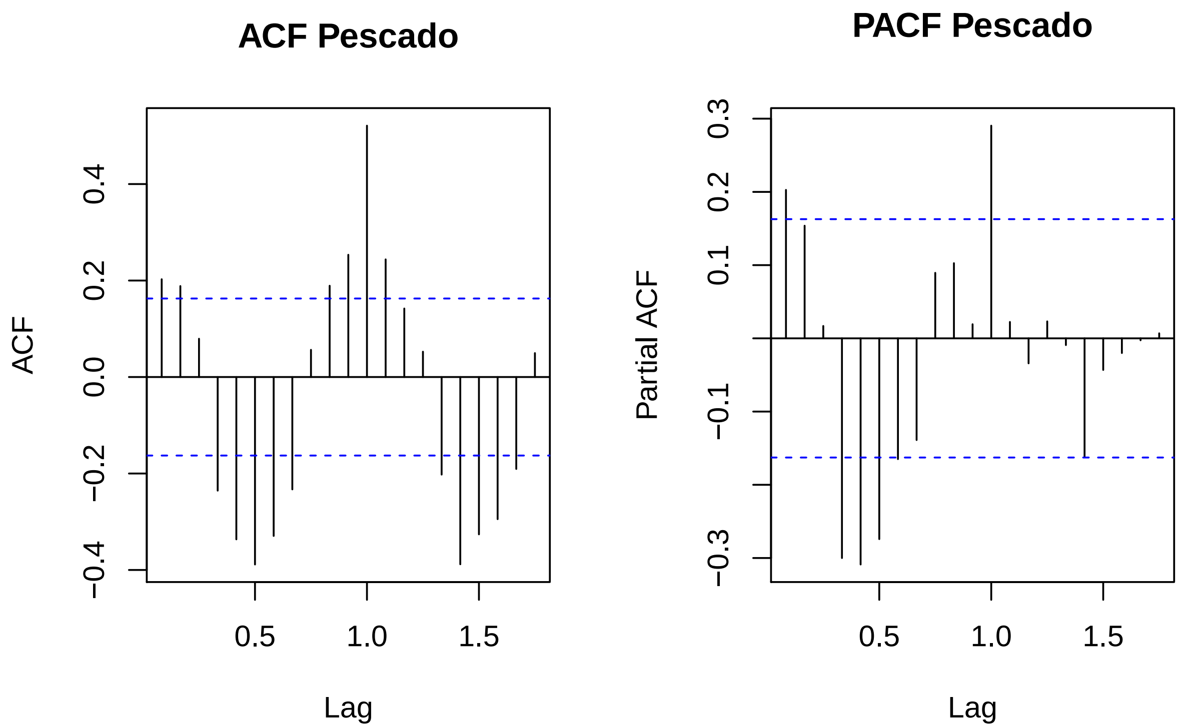
Funções de Autocorrelações para Suinocultura

```
par(mfrow = c(1, 2))
acf(zt22, main="ACF Suínocultura")
pacf(zt22, main="PACF Suínocultura")
```



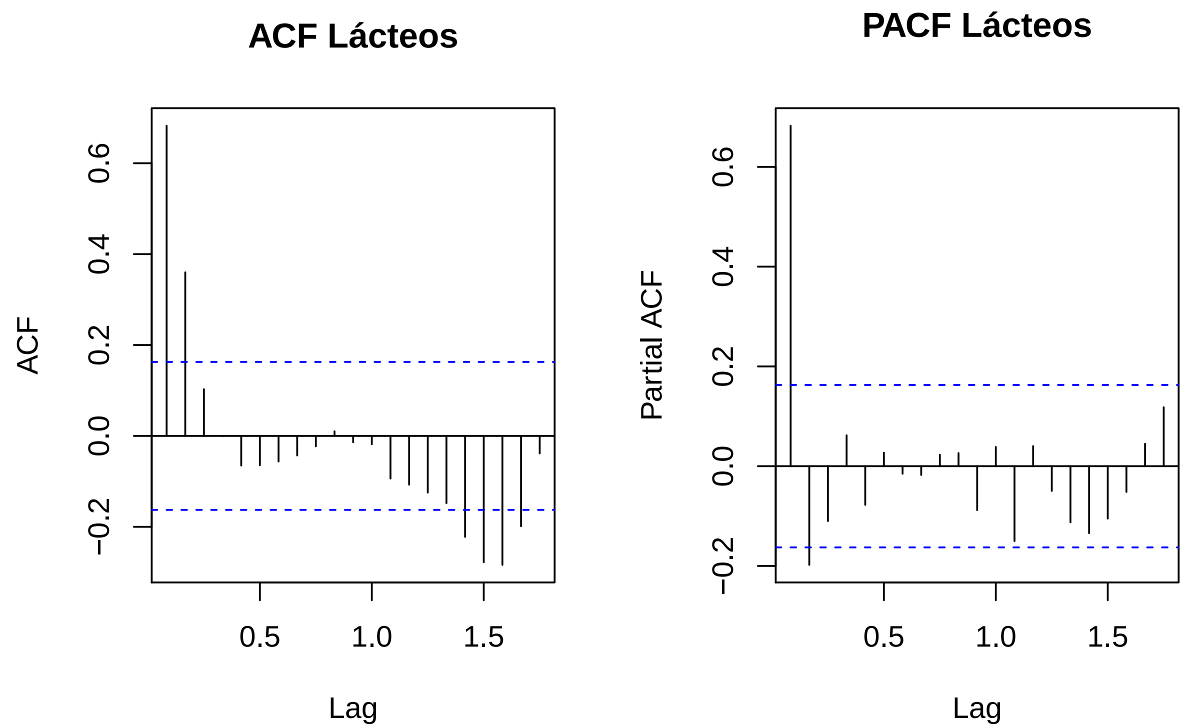
Funções de Autocorrelações para Pescado

```
par(mfrow = c(1, 2))
acf(zt21, main="ACF Pescado")
pacf(zt21, main="PACF Pescado")
```



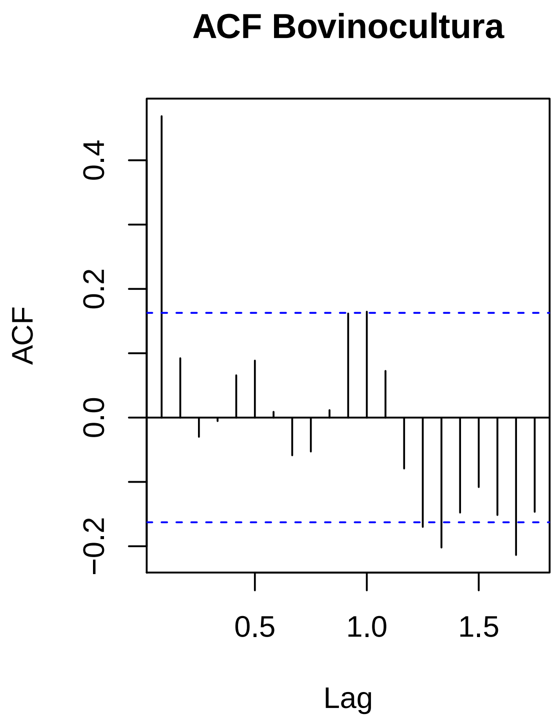
Funções de Autocorrelações para Lácteos

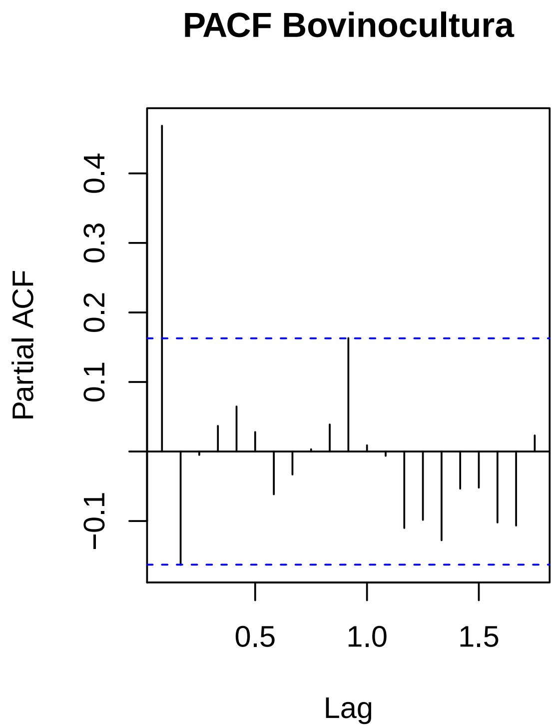
```
par(mfrow = c(1, 2))
acf(zt18, main="ACF Lácteos")
pacf(zt18, main="PACF Lácteos")
```



Funções de Autocorrelações para Bovinocultura

```
par(mfrow = c(1, 2))
acf(zt7, main="ACF Bovinocultura")
pacf(zt7, main="PACF Bovinocultura")
```

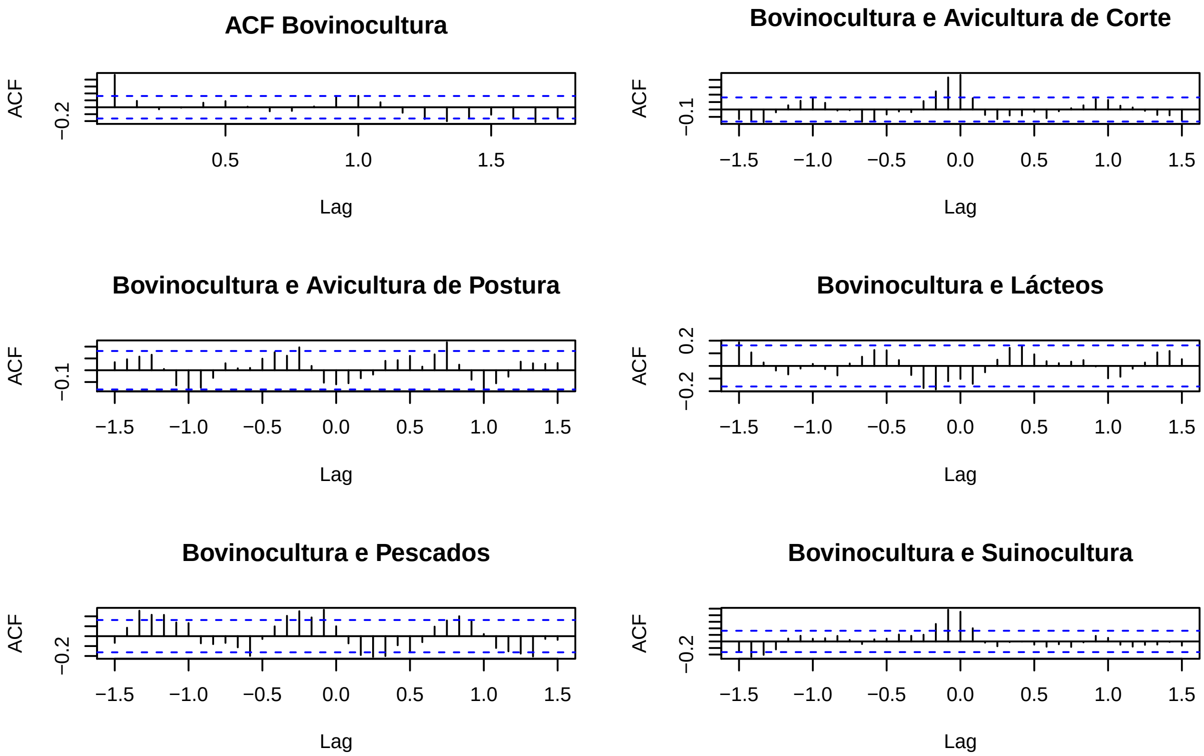




Análise Correlação Cruzada

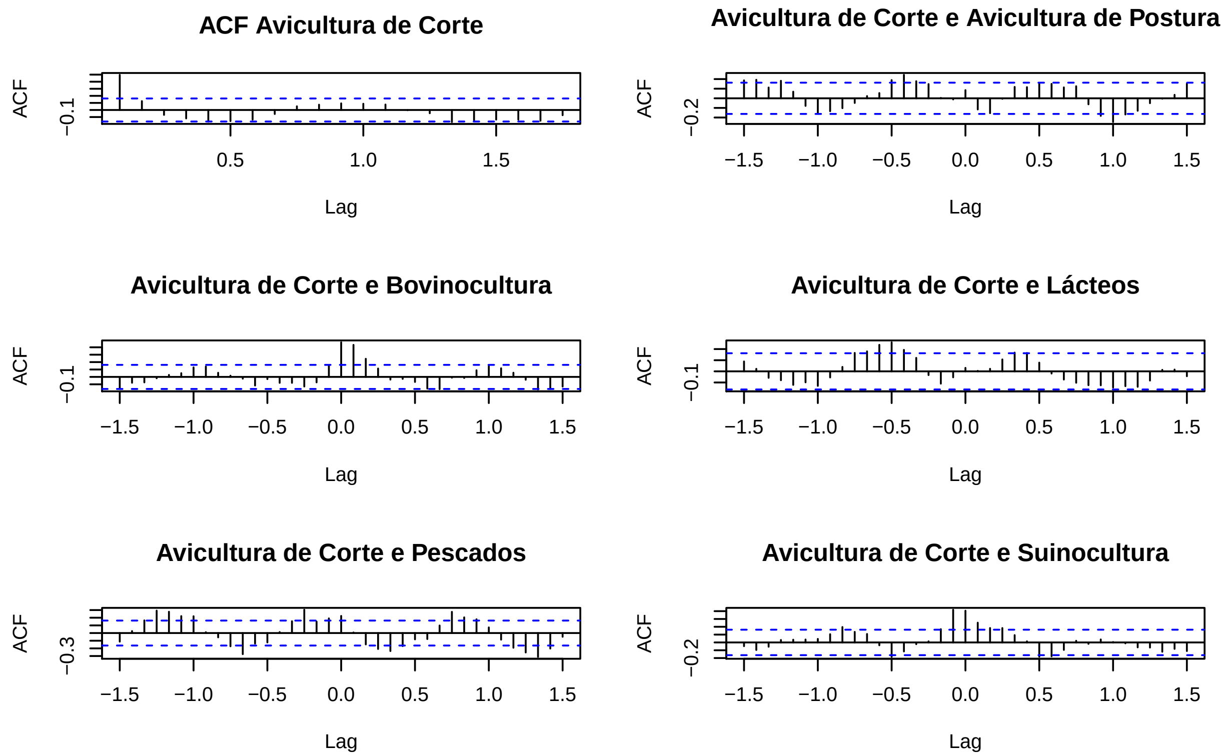
Correlaões cruzadas da Bovincultura

```
#Correlaões cruzadas da Bovincultura
par(mfrow = c(3,2))
acf(zt7,main="ACF Bovinocultura")
ccf(zt7,zt3,main="Bovinocultura e Avicultura de Corte")
ccf(zt7,zt4,main="Bovinocultura e Avicultura de Postura")
ccf(zt7,zt18,main="Bovinocultura e Lácteos")
ccf(zt7,zt21,main="Bovinocultura e Pescados")
ccf(zt7,zt22,main="Bovinocultura e Suinocultura")
```



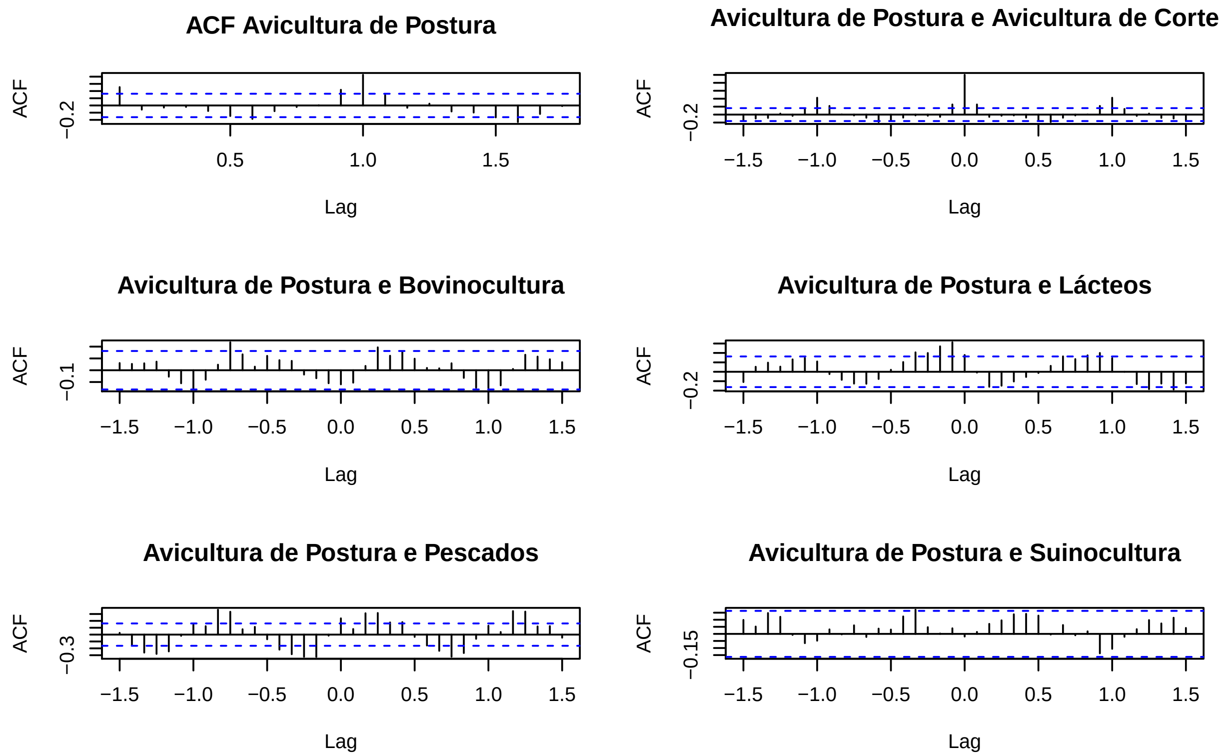
Correlações cruzadas da Avicultura de Corte

```
#Correlações cruzadas da Avicultura de Corte
par(mfrow = c(3,2))
acf(zt3,main="ACF Avicultura de Corte")
ccf(zt3,zt4,main="Avicultura de Corte e Avicultura de Postura")
ccf(zt3,zt7,main="Avicultura de Corte e Bovinocultura")
ccf(zt3,zt18,main="Avicultura de Corte e Lácteos")
ccf(zt3,zt21,main="Avicultura de Corte e Pescados")
ccf(zt3,zt22,main="Avicultura de Corte e Suinocultura")
```



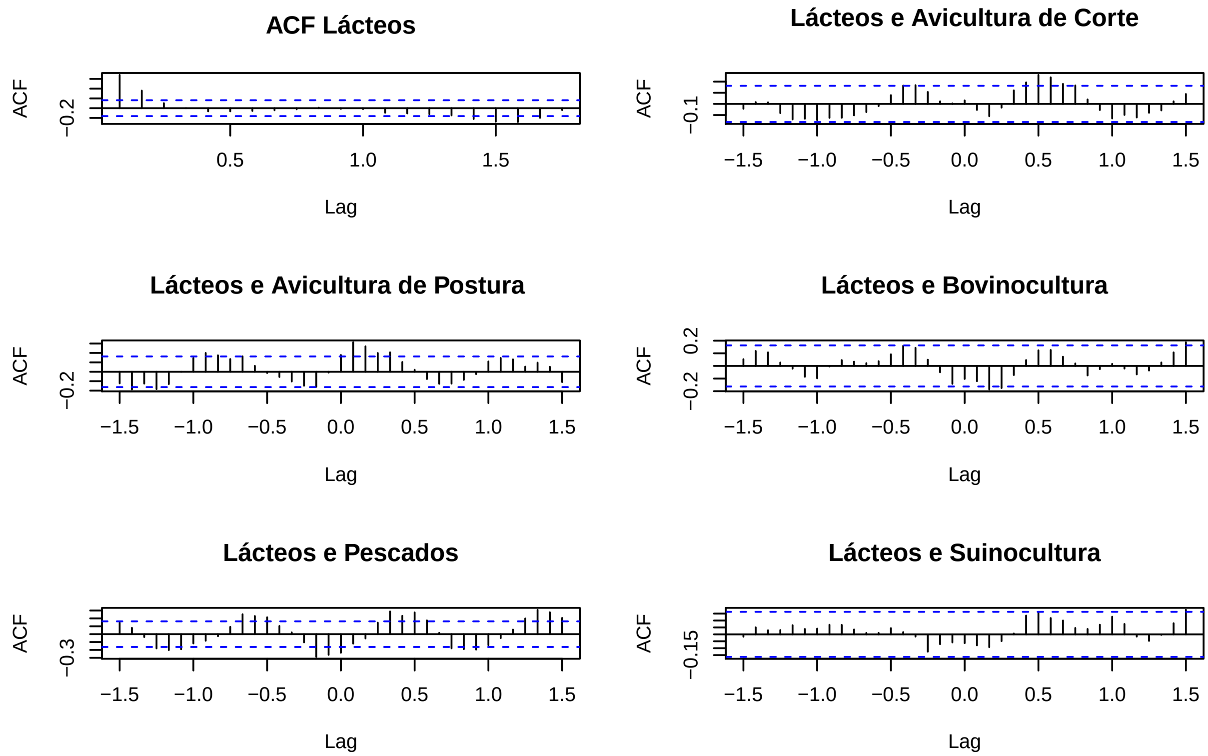
Correlações cruzadas da Avicultura de Postura

```
#Correlações cruzadas da Avicultura de Postura
par(mfrow = c(3,2))
acf(zt4,main="ACF Avicultura de Postura")
ccf(zt4,zt4,main="Avicultura de Postura e Avicultura de Corte")
ccf(zt4,zt7,main="Avicultura de Postura e Bovinocultura")
ccf(zt4,zt18,main="Avicultura de Postura e Lácteos")
ccf(zt4,zt21,main="Avicultura de Postura e Pescados")
ccf(zt4,zt22,main="Avicultura de Postura e Suinocultura")
```



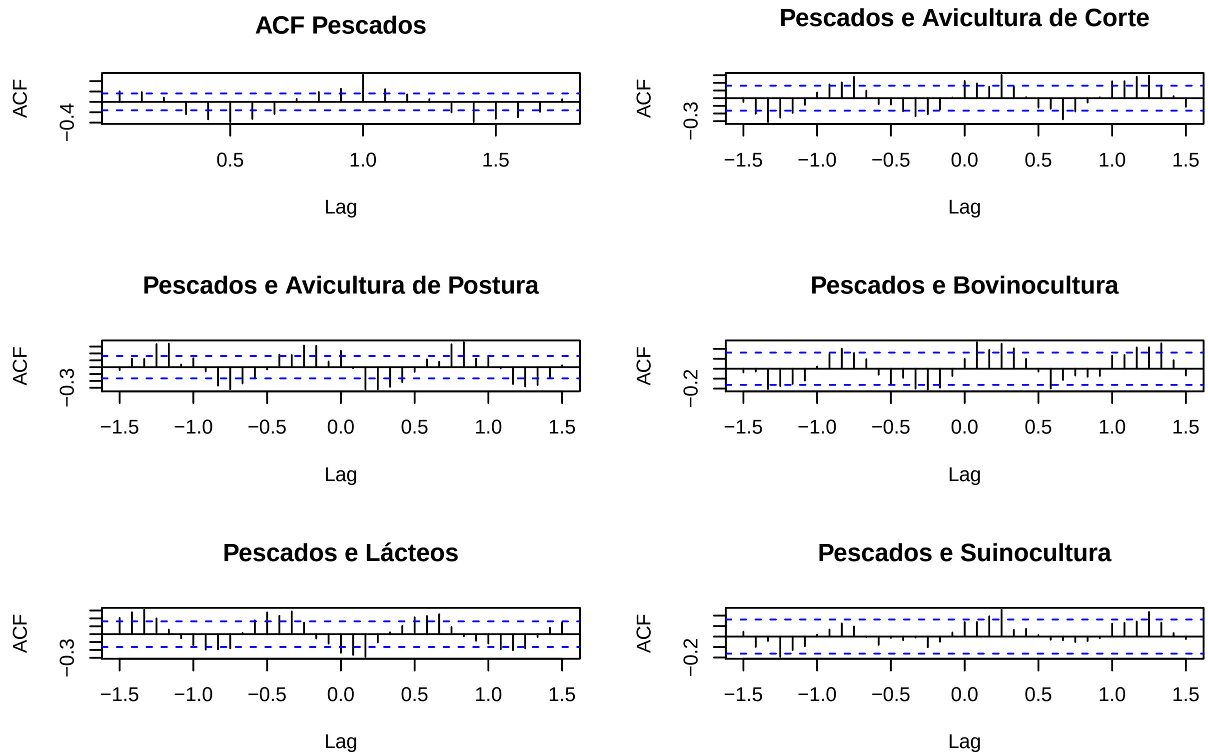
Correlações cruzadas dos Lácteos

```r
#Correlações cruzadas dos Lácteos
par(mfrow = c(3,2))
acf(zt18,main="ACF Lácteos")
ccf(zt18,zt3,main="Lácteos e Avicultura de Corte")
ccf(zt18,zt4,main="Lácteos e Avicultura de Postura ")
ccf(zt18,zt7,main="Lácteos e Bovinocultura")
ccf(zt18,zt21,main="Lácteos e Pescados")
ccf(zt18,zt22,main="Lácteos e Suinocultura")
```



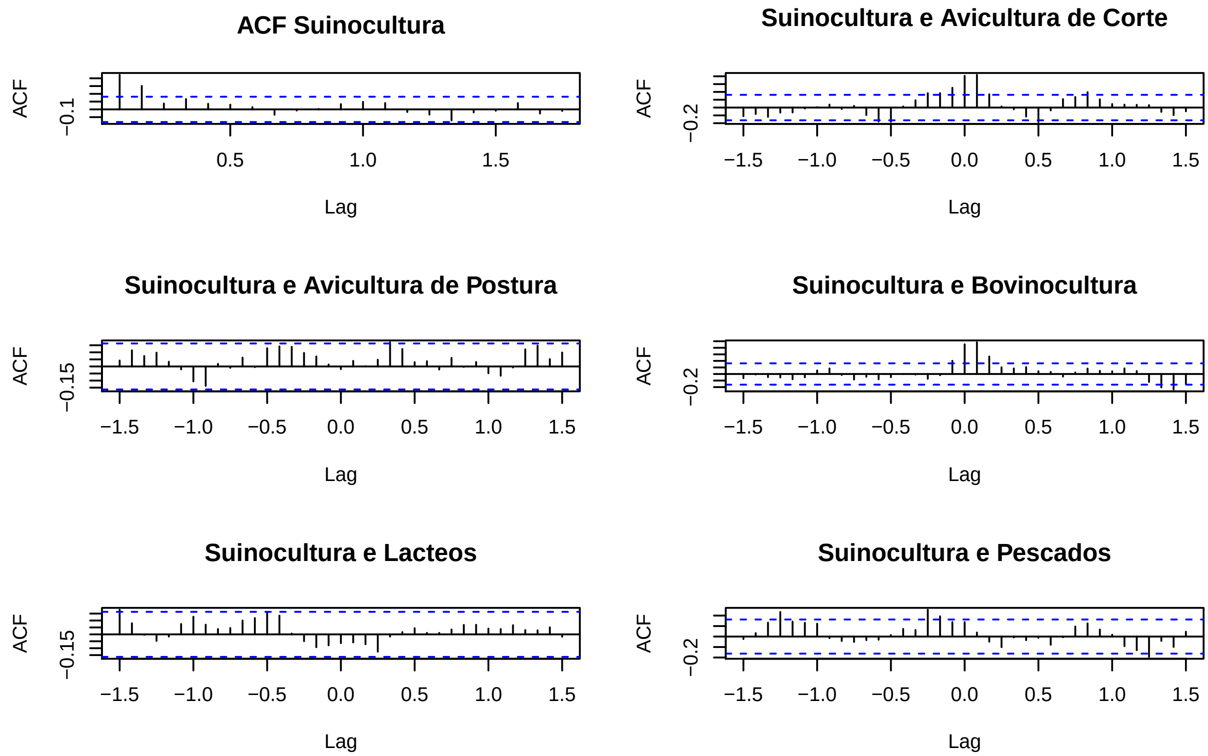
Correlaões cruzadas dos Pescados

```r
# Correlaões cruzadas dos Pescados
par(mfrow = c(3,2))
acf(zt21,main="ACF Pescados")
ccf(zt21,zt3,main="Pescados e Avicultura de Corte")
ccf(zt21,zt4,main="Pescados e Avicultura de Postura")
ccf(zt21,zt7,main="Pescados e Bovinocultura")
ccf(zt21,zt18,main="Pescados e Lácteos")
ccf(zt21,zt22,main="Pescados e Suinocultura")
```



Correlações cruzadas da Suinocultura

```r
#Correlações cruzadas da Suinocultura
par(mfrow = c(3,2))
acf(zt22,main="ACF Suinocultura")
ccf(zt22,zt3,main="Suinocultura e Avicultura de Corte")
ccf(zt22,zt4,main="Suinocultura e Avicultura de Postura")
ccf(zt22,zt7,main="Suinocultura e Bovinocultura")
ccf(zt22,zt18,main="Suinocultura e Lacteos")
ccf(zt22,zt21,main="Suinocultura e Pescados")
```



Selecionado as variáveis de interesse do estudo

Essa função retorna a coluna com a lag a ser considerada na análise

```
funcao_lags = function(df,coluna,nome,lag){
  n = nrow(df)
  pre = rep(NA,lag)
  newcol = c(pre,coluna)
  for (k in 1:lag){
    df = rbind(df,rep(NA,ncol(df)))
  }
  df[nome] = newcol
  return (df)
}
```

A seguir vamos selecionar apenas as variáveis de interesse para análise

```
data_cut = data[,c("Bovinocultura","Avicultura de Corte","Avicultura de Postura","Pescado","Lácteos","S
```

Modelo da Bovinocultura

Estruturando a base

```
data_cut = data[,c("Bovinocultura","Avicultura de Corte","Avicultura de Postura","Pescado","Lácteos","Su

df1<- funcao_lags(data_cut, data_cut$`Avicultura de Postura`, 'avp9', 9)
df1 <- funcao_lags(df1, df1$Pescado, 'p3', 3)
df1 <- funcao_lags(df1, df1$Pescado, 'p10', 10)
df1 <- funcao_lags(df1, df1$Bovinocultura, 'b1', 1)


df2 <- na.omit(df1)
```

Separando variável preditora e as covariáveis

```
x = model.matrix(Bovinocultura~.,df2)[,-1]
y = df2$Bovinocultura
```

Regressão LASSO

A seguir vamos utilizar a biblioteca "glmnet"

```
set.seed(123)
cv.lasso <- cv.glmnet(x, y, alpha = 1, family = "gaussian")
summary(cv.lasso)
```

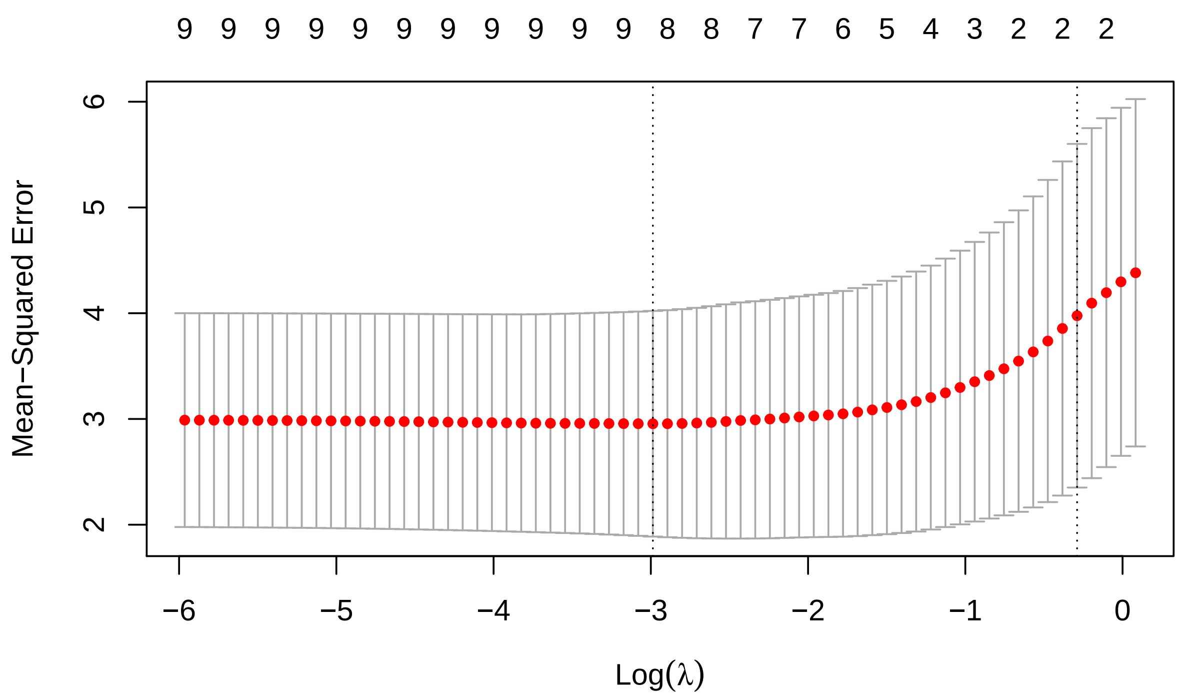
```
##            Length Class  Mode
## lambda     66     -none- numeric
## cvm        66     -none- numeric
## cvsd       66     -none- numeric
## cvup       66     -none- numeric
## cvlo       66     -none- numeric
## nzero      66     -none- numeric
## call        5     -none- call
## name        1     -none- character
## glmnet.fit 12     elnet  list
## lambda.min  1     -none- numeric
## lambda.1se  1     -none- numeric
## index       2     -none- numeric
```

```
print(cv.lasso)
```

```
##
## Call:  cv.glmnet(x = x, y = y, alpha = 1, family = "gaussian")
##
## Measure: Mean-Squared Error
##
```

```
##     Lambda Index Measure    SE Nonzero
## min 0.0504    34   2.955 1.067       8
## 1se 0.7489     5   3.976 1.625       2
```

```r
plot(cv.lasso)
```




```r
cv.lasso$lambda.min
```

```
## [1] 0.05043405
```

```r
cv.lasso$lambda.1se
```

```
## [1] 0.7489297
```

```r
coef(cv.lasso, cv.lasso$lambda.min)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                   1
## (Intercept)              0.31262107
## `Avicultura de Corte`    0.39843764
## `Avicultura de Postura`  .
## Pescado                 -0.15329263
## Lácteos                 -0.15094180
```

```
## Suinocultura             0.24605653
## avp9                     0.14932952
## p3                      -0.01311084
## p10                      0.01739267
## b1                       0.35784156
```

```r
coef(cv.lasso, cv.lasso$lambda.1se)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                   1
## (Intercept)              0.68566274
## `Avicultura de Corte`    0.08952851
## `Avicultura de Postura` .
## Pescado                 .
## Lácteos                 .
## Suinocultura            .
## avp9                    .
## p3                      .
## p10                     .
## b1                       0.15648256
```

A seguir vamos utilizar a biblioteca "islasso"

```r
model.islasso <- islasso(y ~ x, lambda = cv.lasso$lambda.min)
summary(model.islasso)
```

```
##
## Call:
## islasso(formula = y ~ x, lambda = cv.lasso$lambda.min)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5312 -0.9189 -0.0162  0.5589  8.5768
##
##                           Estimate Std. Error    Df z value Pr(>|z|)
## (Intercept)                0.28555    0.20359 1.000   1.403 0.160748
## x`Avicultura de Corte`     0.41319    0.11348 1.000   3.641 0.000271 ***
## x`Avicultura de Postura`   0.04527    0.06032 1.000   0.750 0.452995
## xPescado                  -0.25994    0.11192 1.000  -2.323 0.020202 *
## xLácteos                  -0.20752    0.12319 1.000  -1.685 0.092067 .
## xSuinocultura              0.27996    0.21147 0.999   1.324 0.185555
## xavp9                      0.17970    0.05358 1.000   3.354 0.000796 ***
## xp3                       -0.02206    0.10177 0.999  -0.217 0.828397
## xp10                       0.07148    0.10156 0.999   0.704 0.481548
## xb1                        0.37949    0.09756 1.000   3.890 0.000100 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.677741)
##
##     Null deviance: 630.47  on 145  degrees of freedom
## Residual deviance: 364.18  on 136  degrees of freedom
```

```
## AIC: 569.77
## Lambda: 0.050434
##
## Number of Newton-Raphson iterations: 4
```

Regressão classifica no contexto de Séries Temporais

Criando o modelo de Regressão Simples

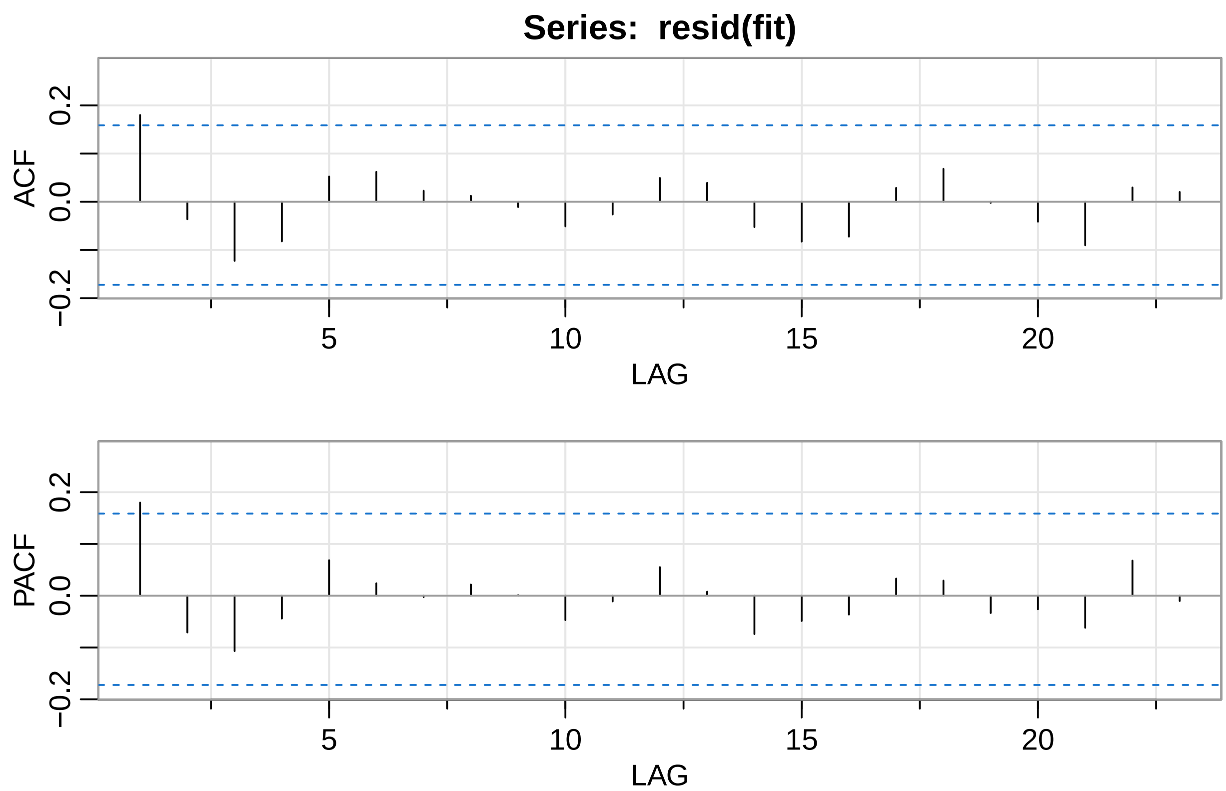
```r
set.seed(1234)

fit1 <- summary(fit <- lm(y~x))
fit1
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5314 -0.9189 -0.0157  0.5586  8.5757
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.28536    0.20364   1.401 0.163405
## x`Avicultura de Corte`     0.41328    0.11349   3.642 0.000384 ***
## x`Avicultura de Postura`   0.04542    0.06035   0.753 0.452982
## xPescado                  -0.26037    0.11194  -2.326 0.021498 *
## xLácteos                  -0.20785    0.12322  -1.687 0.093939 .
## xSuinocultura              0.28048    0.21162   1.325 0.187266
## xavp9                      0.17980    0.05358   3.356 0.001026 **
## xp3                       -0.02202    0.10186  -0.216 0.829147
## xp10                       0.07166    0.10163   0.705 0.481954
## xb1                        0.37950    0.09758   3.889 0.000157 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.636 on 136 degrees of freedom
## Multiple R-squared:  0.4224, Adjusted R-squared:  0.3841
## F-statistic: 11.05 on 9 and 136 DF,  p-value: 8.134e-13
```

```r
#write.csv(fit1$coefficients, file = 'tabela_reg.csv')
```

Análise dos Resíduos

```r
acf2(resid(fit))
```




```
##      [,1]  [,2]  [,3]  [,4] [,5] [,6] [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.18 -0.04 -0.12 -0.08 0.05 0.06 0.02 0.01 -0.01 -0.05 -0.03  0.05  0.04
## PACF 0.18 -0.07 -0.11 -0.04 0.07 0.02 0.00 0.02  0.00 -0.05 -0.01  0.06  0.01
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]
## ACF  -0.05 -0.08 -0.07  0.03  0.07  0.00 -0.04 -0.09  0.03  0.02
## PACF -0.07 -0.05 -0.04  0.03  0.03 -0.03 -0.03 -0.06  0.07 -0.01
```

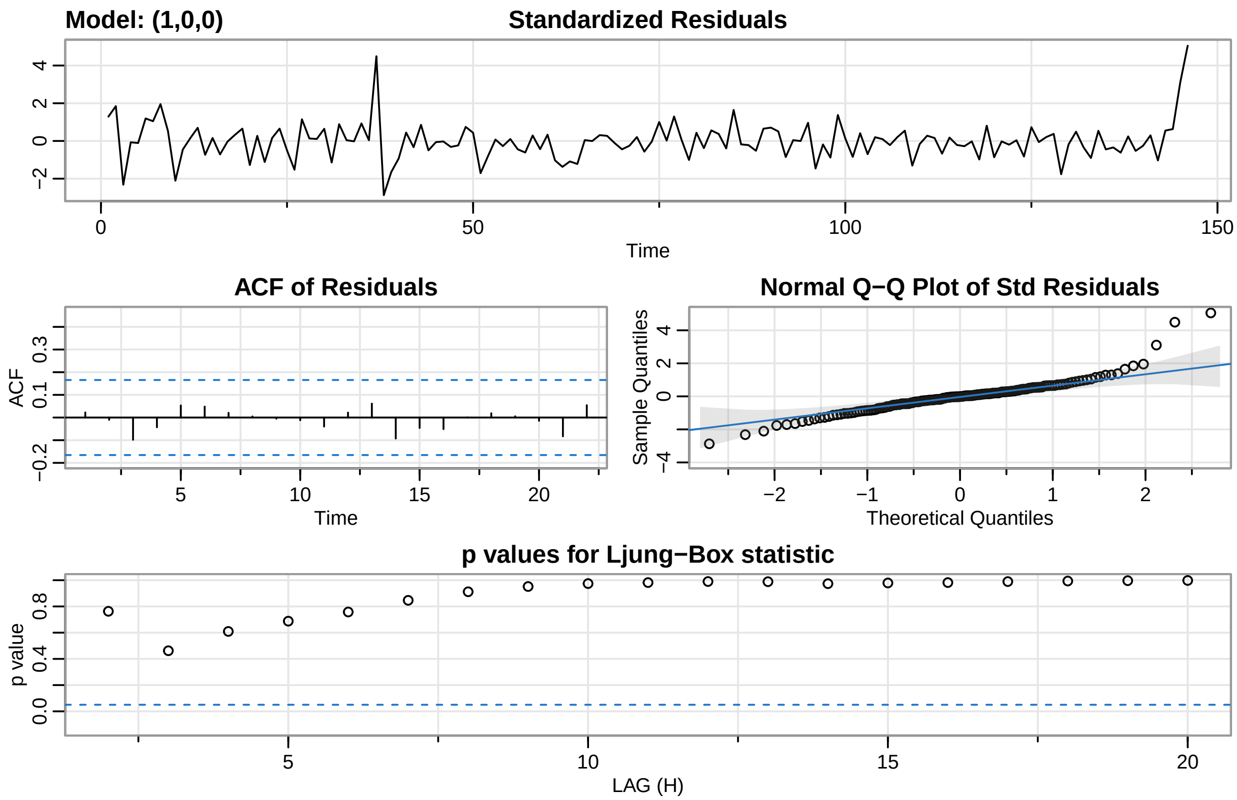
Regressão com erros autocorrelacionais

Criando o modelo de Regressão com erros autocorrelacionados

```
set.seed(12345)


fit2 <- sarima(y, 1,0,0, xreg= x)
```

```
## initial  value 0.453951
## iter   2 value 0.433468
## iter   3 value 0.420315
## iter   4 value 0.411120
## iter   5 value 0.410309
## iter   6 value 0.410165
## iter   7 value 0.410127
## iter   8 value 0.410120
## iter   9 value 0.410117
```

```
## iter  10 value 0.410117
## iter  11 value 0.410116
## iter  12 value 0.410116
## iter  12 value 0.410116
## iter  12 value 0.410116
## final  value 0.410116
## converged
## initial  value 0.414187
## iter   2 value 0.413832
## iter   3 value 0.413745
## iter   4 value 0.413714
## iter   5 value 0.413707
## iter   6 value 0.413704
## iter   7 value 0.413704
## iter   8 value 0.413704
## iter   9 value 0.413704
## iter  10 value 0.413704
## iter  10 value 0.413704
## iter  10 value 0.413704
## final  value 0.413704
## converged
```




```
fit2
```

```
## $fit
##
```

```
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1  intercept  ‘Avicultura de Corte‘  ‘Avicultura de Postura‘
##       0.4691     0.4196                 0.5589                   0.0076
## s.e.  0.1227     0.2895                 0.1131                   0.0540
##       Pescado  Lácteos  Suinocultura    avp9      p3     p10      b1
##       -0.1639  -0.1834        0.3054  0.1548  0.0282  0.1139  0.0712
## s.e.   0.0963   0.1454        0.2012  0.0492  0.0856  0.0844  0.1194
##
## sigma^2 estimated as 2.283:  log likelihood = -267.57,  aic = 559.13
##
## $degrees_of_freedom
## [1] 135
##
## $ttable
##                          Estimate     SE t.value p.value
## ar1                        0.4691 0.1227  3.8246  0.0002
## intercept                  0.4196 0.2895  1.4497  0.1495
## ‘Avicultura de Corte‘      0.5589 0.1131  4.9432  0.0000
## ‘Avicultura de Postura‘    0.0076 0.0540  0.1400  0.8889
## Pescado                   -0.1639 0.0963 -1.7023  0.0910
## Lácteos                   -0.1834 0.1454 -1.2614  0.2093
## Suinocultura               0.3054 0.2012  1.5182  0.1313
## avp9                       0.1548 0.0492  3.1468  0.0020
## p3                         0.0282 0.0856  0.3290  0.7427
## p10                        0.1139 0.0844  1.3496  0.1794
## b1                         0.0712 0.1194  0.5960  0.5521
##
## $AIC
## [1] 3.829668
##
## $AICc
## [1] 3.843162
##
## $BIC
## [1] 4.074896
```

Análise dos resíduos e seleção de variáveis de acordo com p-valor

```
fit3 = Arima(y,order=c(1,0,0),xreg=x)
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##                            Estimate Std. Error z value  Pr(>|z|)
## ar1                       0.4691103  0.1226563  3.8246  0.000131 ***
## intercept                 0.4196475  0.2894685  1.4497  0.147137
```

```
## `Avicultura de Corte`    0.5588534  0.1130554  4.9432 7.686e-07 ***
## `Avicultura de Postura`  0.0075602  0.0540004  0.1400  0.888658
## Pescado                 -0.1638517  0.0962515 -1.7023  0.088694 .
## Lácteos                 -0.1834013  0.1453893 -1.2614  0.207147
## Suinocultura             0.3054252  0.2011709  1.5182  0.128954
## avp9                     0.1547863  0.0491889  3.1468  0.001651 **
## p3                       0.0281560  0.0855908  0.3290  0.742185
## p10                      0.1139403  0.0844258  1.3496  0.177147
## b1                       0.0711886  0.1194340  0.5960  0.551142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#d = diag(fit3$var.coef)**(0.5)
#t = fit3$coef/d
#p = 2*pt(-abs(t),144)
#p
#max(p)

x = x[,-2]
fit3 = Arima(y,order=c(1,0,0),xreg=x)
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##                          Estimate Std. Error z value  Pr(>|z|)
## ar1                      0.472559   0.119873  3.9422 8.075e-05 ***
## intercept                0.424794   0.288533  1.4723  0.140952
## `Avicultura de Corte`  0.562057   0.110669  5.0787 3.800e-07 ***
## Pescado                 -0.159513   0.090943 -1.7540  0.079434 .
## Lácteos                 -0.180632   0.144222 -1.2525  0.210402
## Suinocultura             0.304150   0.200877  1.5141  0.129999
## avp9                     0.154093   0.048899  3.1512  0.001626 **
## p3                       0.029407   0.085013  0.3459  0.729412
## p10                      0.111887   0.083085  1.3466  0.178093
## b1                       0.067477   0.116181  0.5808  0.561382
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
x = x[,-6]
fit3 = Arima(y,order=c(1,0,0),xreg=x)
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##                          Estimate Std. Error z value  Pr(>|z|)
## ar1                      0.470714   0.120280  3.9135 9.097e-05 ***
## intercept                0.443624   0.282574  1.5699  0.116430
## `Avicultura de Corte`  0.560367   0.110747  5.0599 4.195e-07 ***
## Pescado                 -0.153605   0.089319 -1.7197  0.085480 .
## Lácteos                 -0.173315   0.142529 -1.2160  0.223984
```

```
## Suinocultura           0.298787   0.200426  1.4908  0.136024
## avp9                   0.153094   0.048831  3.1352  0.001717 **
## p10                    0.108064   0.082545  1.3091  0.190486
## b1                     0.065841   0.116363  0.5658  0.571513
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
x = x[,-7]
fit3 = Arima(y,order=c(1,0,0),xreg=x)
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##                         Estimate Std. Error z value  Pr(>|z|)
## ar1                     0.512849   0.091696  5.5929 2.233e-08 ***
## intercept               0.481325   0.293758  1.6385  0.101316
## `Avicultura de Corte`  0.575926   0.107139  5.3755 7.637e-08 ***
## Pescado               -0.141859   0.085662 -1.6560  0.097716 .
## Lácteos               -0.166660   0.144696 -1.1518  0.249406
## Suinocultura           0.321192   0.195518  1.6428  0.100430
## avp9                   0.149711   0.048098  3.1126  0.001854 **
## p10                    0.110441   0.081526  1.3547  0.175522
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
x = x[,-3]
fit3 = Arima(y,order=c(1,0,0),xreg=x)
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##                         Estimate Std. Error z value  Pr(>|z|)
## ar1                     0.535081   0.089654  5.9683 2.398e-09 ***
## intercept               0.408757   0.299850  1.3632  0.172818
## `Avicultura de Corte`  0.554877   0.106151  5.2272 1.721e-07 ***
## Pescado               -0.129387   0.084641 -1.5286  0.126353
## Suinocultura           0.322514   0.195422  1.6503  0.098873 .
## avp9                   0.151889   0.048036  3.1620  0.001567 **
## p10                    0.116779   0.081127  1.4395  0.150022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit3 = Arima(y,order=c(1,0,0),xreg=x,include.mean = FALSE)
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##                         Estimate Std. Error z value  Pr(>|z|)
```

```
## ar1                     0.536671   0.092785  5.7841 7.292e-09 ***
## `Avicultura de Corte`  0.566061   0.106554  5.3124 1.082e-07 ***
## Pescado               -0.117841   0.084716 -1.3910  0.164221
## Suinocultura           0.391642   0.193253  2.0266  0.042705 *
## avp9                   0.157512   0.048276  3.2628  0.001103 **
## p10                    0.135760   0.080474  1.6870  0.091605 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
x = x[,-2]
fit3 = Arima(y,order=c(1,0,0),xreg=x,include.mean = FALSE)
coeftest(fit3)
```

```
##
## z test of coefficients:
##
##                        Estimate Std. Error z value  Pr(>|z|)
## ar1                    0.530578   0.092219  5.7535 8.743e-09 ***
## `Avicultura de Corte` 0.537948   0.105359  5.1059 3.293e-07 ***
## Suinocultura           0.378967   0.193652  1.9570  0.050353 .
## avp9                   0.144710   0.047519  3.0453  0.002324 **
## p10                    0.121088   0.080385  1.5064  0.131976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
x = x[,-4]
fit3 = Arima(y,order=c(1,0,0),xreg=x,include.mean = FALSE)
coeftest(fit3)
```

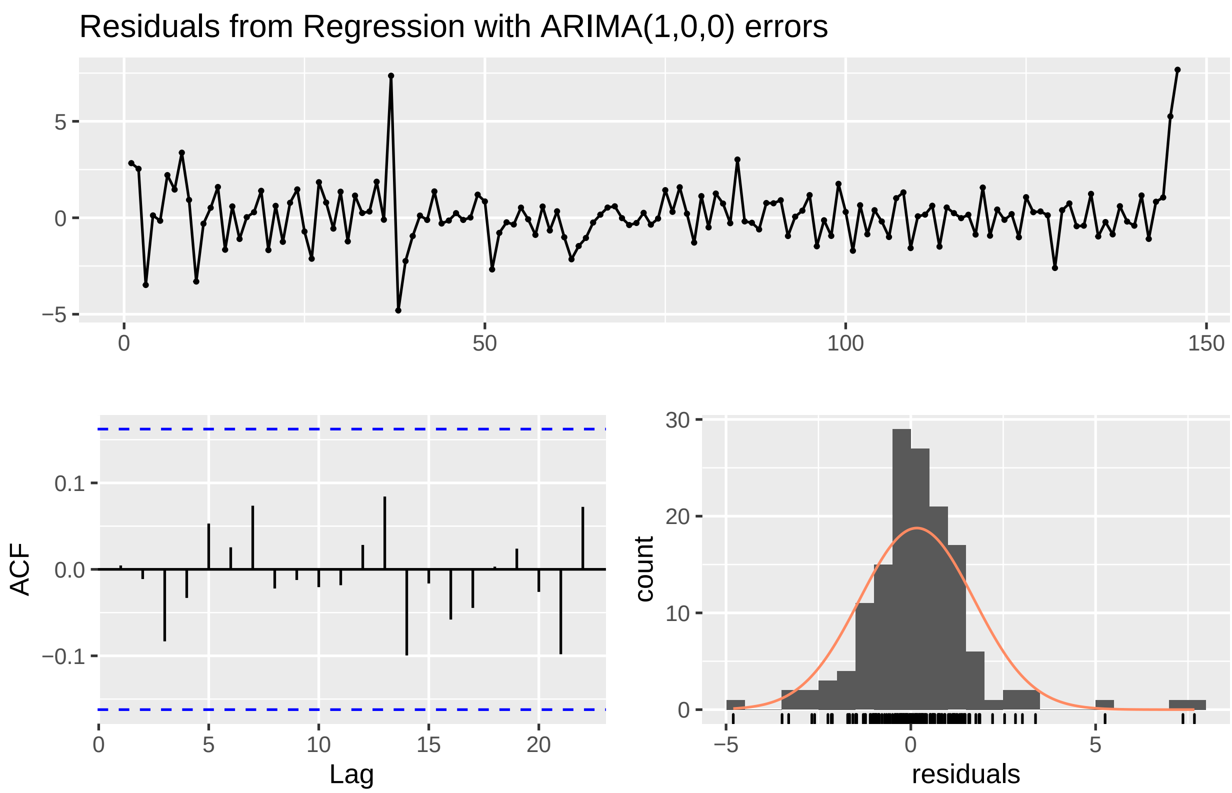
```
##
## z test of coefficients:
##
##                        Estimate Std. Error z value  Pr(>|z|)
## ar1                    0.526517   0.094724  5.5584 2.722e-08 ***
## `Avicultura de Corte` 0.541743   0.106566  5.0836 3.703e-07 ***
## Suinocultura           0.422422   0.194656  2.1701  0.029999 *
## avp9                   0.133101   0.047602  2.7961  0.005172 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(fit3)
```

```
## Series: y
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1  `Avicultura de Corte`  Suinocultura    avp9
##       0.5265                 0.5417        0.4224  0.1331
## s.e.  0.0947                 0.1066        0.1947  0.0476
##
## sigma^2 estimated as 2.477:  log likelihood=-271.5
```

```
## AIC=553   AICc=553.43   BIC=567.92
##
## Training set error measures:
##                     ME    RMSE     MAE       MPE     MAPE     MASE        ACF1
## Training set 0.1626874 1.55201 1.01952 -215.2519 535.9834 0.823407 0.004500127
```

```
checkresiduals(fit3)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,0) errors
## Q* = 2.7751, df = 6, p-value = 0.8365
##
## Model df: 4.   Total lags used: 10
```

Modelo da Avicultura de Corte

Estruturando a base

```
data_cut = data[,c("Bovinocultura","Avicultura de Corte","Avicultura de Postura","Pescado","Lácteos","Su
```

```r
df1<- funcao_lags(data_cut, data_cut$`Avicultura de Corte`, 'cort1', 1)
df1 <- funcao_lags(df1, df1$`Avicultura de Postura`, 'pos12', 12)
df1 <- funcao_lags(df1, df1$Bovinocultura, 'bov1', 1)
df1 <- funcao_lags(df1, df1$Pescado, 'pes4', 4)
df1 <- funcao_lags(df1, df1$Pescado, 'pes9', 9)
df1 <- funcao_lags(df1, df1$Suinocultura, 'sui1', 1)
df1 <- funcao_lags(df1, df1$Suinocultura, 'sui6', 6)

df2 <- na.omit(df1)
```

Separando variável preditora e as covariáveis

```r
x = model.matrix(`Avicultura de Corte`~.,df2)[,-1]
y = df2$`Avicultura de Corte`
```

Regressão LASSO

A seguir vamos utilizar a biblioteca "glmnet"

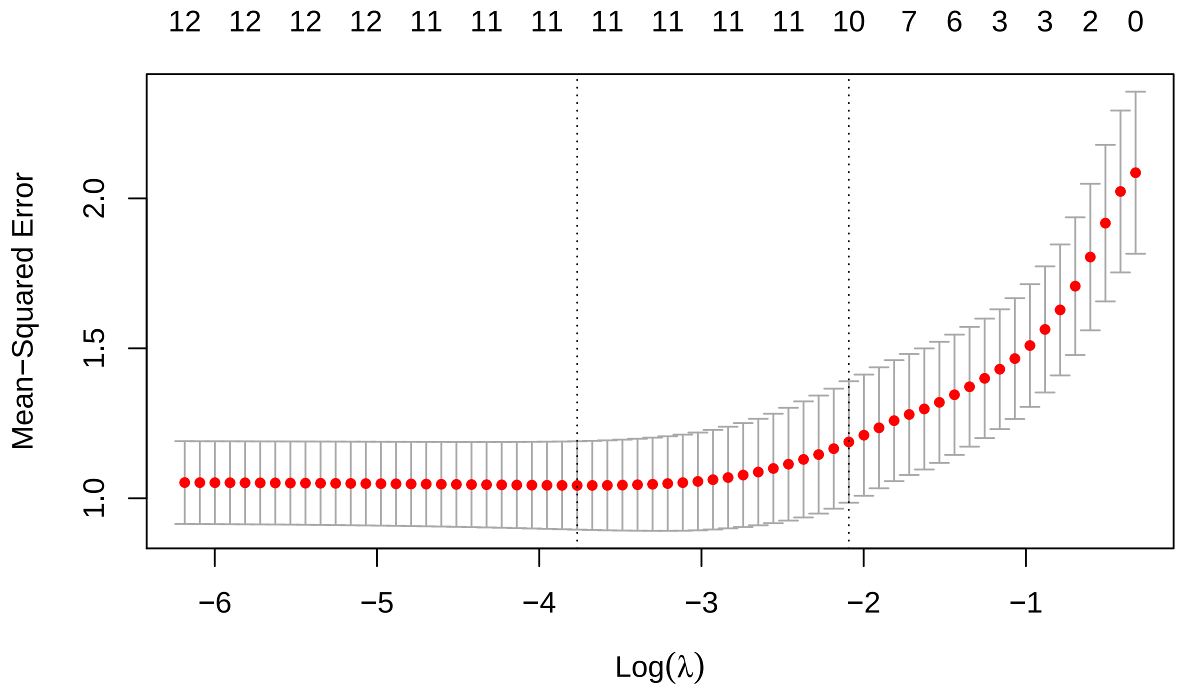
```r
set.seed(123)
cv.lasso <- cv.glmnet(x, y, alpha = 1, family = "gaussian")
summary(cv.lasso)
```

```
##            Length Class  Mode
## lambda     64     -none- numeric
## cvm        64     -none- numeric
## cvsd       64     -none- numeric
## cvup       64     -none- numeric
## cvlo       64     -none- numeric
## nzero      64     -none- numeric
## call        5     -none- call
## name        1     -none- character
## glmnet.fit 12     elnet  list
## lambda.min  1     -none- numeric
## lambda.1se  1     -none- numeric
## index       2     -none- numeric
```

```r
print(cv.lasso)
```

```
##
## Call:  cv.glmnet(x = x, y = y, alpha = 1, family = "gaussian")
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure     SE Nonzero
## min 0.02314    38   1.043 0.1473      11
## 1se 0.12349    20   1.188 0.2025      10
```

```r
plot(cv.lasso)
```




```r
cv.lasso$lambda.min
```

```
## [1] 0.0231396
```

```r
cv.lasso$lambda.1se
```

```
## [1] 0.1234891
```

```r
coef(cv.lasso, cv.lasso$lambda.min)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                                  1
## (Intercept)            -0.001182139
## Bovinocultura           0.220213134
## `Avicultura de Postura`  0.112923949
## Pescado                 0.061583582
## Lácteos                 0.206120348
## Suinocultura            0.167954058
## cort1                   0.337723751
## pos12                  -0.085647595
```

```
## bov1                        0.066240159
## pes4                       -0.044625127
## pes9                        0.147778932
## sui1                        .
## sui6                       -0.405537952
```

```r
coef(cv.lasso, cv.lasso$lambda.1se)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                                      1
## (Intercept)                 0.08401313
## Bovinocultura               0.21028331
## `Avicultura de Postura`     0.04114801
## Pescado                     0.01432741
## Lácteos                     0.05374680
## Suinocultura                0.07755948
## cort1                       0.33324689
## pos12                      -0.02031133
## bov1                        0.04772662
## pes4                        .
## pes9                        0.09622631
## sui1                        .
## sui6                       -0.22041914
```

A seguir vamos utilizar a biblioteca "islasso"

```r
model.islasso <- islasso(y ~ x, lambda = cv.lasso$lambda.min)
summary(model.islasso)
```

```
##
## Call:
## islasso(formula = y ~ x, lambda = cv.lasso$lambda.min)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8585 -0.5431 -0.0326  0.5124  3.4837
##
##                            Estimate Std. Error    Df z value Pr(>|z|)
## (Intercept)                -0.01215    0.13952 1.000  -0.087 0.930600
## xBovinocultura              0.22082    0.05196 1.000   4.250 2.14e-05 ***
## x`Avicultura de Postura`    0.12987    0.03793 1.000   3.424 0.000618 ***
## xPescado                    0.07100    0.06224 1.000   1.141 0.253913
## xLácteos                    0.24233    0.07548 1.000   3.210 0.001326 **
## xSuinocultura               0.19626    0.13940 0.999   1.408 0.159163
## xcort1                      0.33938    0.07224 1.000   4.698 2.63e-06 ***
## xpos12                     -0.10055    0.03611 1.000  -2.785 0.005359 **
## xbov1                       0.07237    0.06581 1.000   1.100 0.271503
## xpes4                      -0.06139    0.06409 1.000  -0.958 0.338119
## xpes9                       0.15780    0.06197 1.000   2.546 0.010887 *
## xsui1                      -0.01753    0.13014 0.999  -0.135 0.892838
## xsui6                      -0.44549    0.11305 1.000  -3.941 8.12e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9538159)
##
##     Null deviance: 300.04  on 143  degrees of freedom
## Residual deviance: 124.95  on 131  degrees of freedom
## AIC: 416.22
## Lambda: 0.02314
##
## Number of Newton-Raphson iterations: 4
```

Regressão classica no contexto de Séries Temporais

Criando o modelo de Regressão Simples

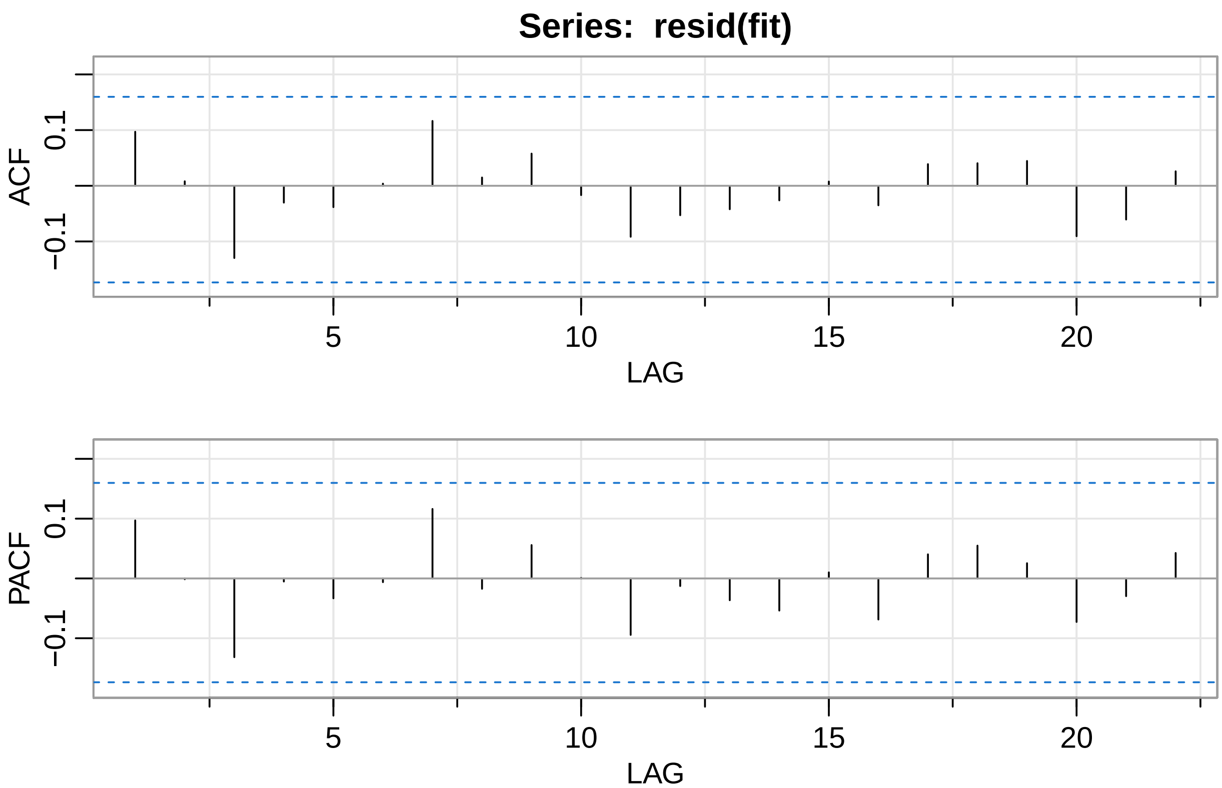
```r
set.seed(1234)

fit1 <- summary(fit <- lm(y~x))
fit1
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8583 -0.5435 -0.0324  0.5123  3.4823
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -0.01212    0.13955  -0.087 0.930894
## xBovinocultura            0.22080    0.05196   4.249 4.05e-05 ***
## x`Avicultura de Postura`  0.12991    0.03793   3.425 0.000822 ***
## xPescado                  0.07105    0.06226   1.141 0.255841
## xLácteos                  0.24253    0.07549   3.213 0.001655 **
## xSuinocultura             0.19667    0.13949   1.410 0.160939
## xcort1                    0.33941    0.07225   4.698 6.56e-06 ***
## xpos12                   -0.10059    0.03611  -2.785 0.006139 **
## xbov1                     0.07239    0.06584   1.099 0.273628
## xpes4                    -0.06147    0.06412  -0.959 0.339505
## xpes9                     0.15784    0.06198   2.547 0.012035 *
## xsui1                    -0.01789    0.13031  -0.137 0.891035
## xsui6                    -0.44583    0.11305  -3.944 0.000130 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9766 on 131 degrees of freedom
## Multiple R-squared:  0.5835, Adjusted R-squared:  0.5454
## F-statistic:  15.3 on 12 and 131 DF,  p-value: < 2.2e-16
```

```r
#write.csv(fit1$coefficients, file = 'tabela_reg.csv')
```

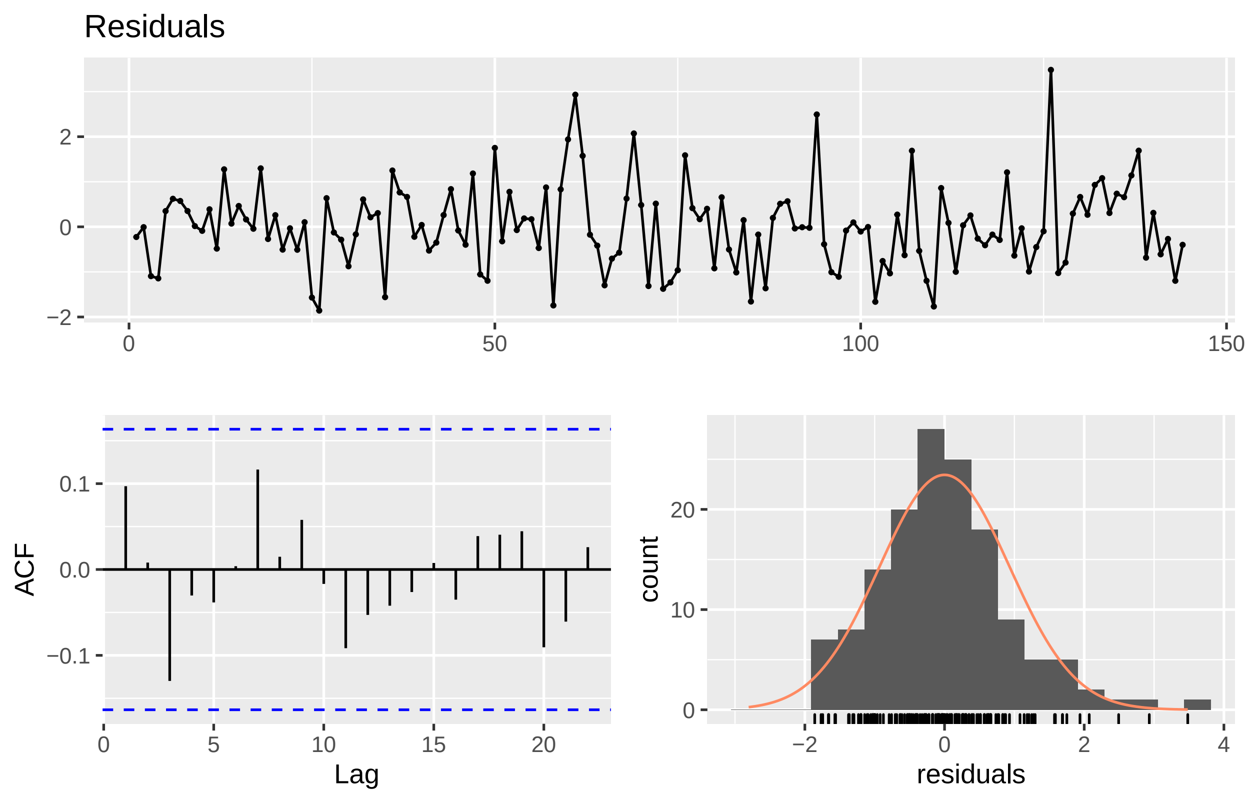
Análise dos Resíduos

```r
acf2(resid(fit))
```




```
##      [,1] [,2]  [,3]  [,4]  [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF   0.1 0.01 -0.13 -0.03 -0.04  0.00 0.12  0.01 0.06 -0.02 -0.09 -0.05 -0.04
## PACF  0.1 0.00 -0.13 -0.01 -0.03 -0.01 0.12 -0.02 0.06  0.00 -0.09 -0.01 -0.04
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22]
## ACF  -0.03  0.01 -0.04  0.04  0.04  0.04 -0.09 -0.06  0.03
## PACF -0.05  0.01 -0.07  0.04  0.05  0.03 -0.07 -0.03  0.04
```

```r
checkresiduals(fit)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 16
##
## data:  Residuals
## LM test = 12.064, df = 16, p-value = 0.7396
```

Seleção de variáveis

```
set.seed(123)

fit2 <- summary(fit <- lm(y~x-1))
fit2
```

```
##
## Call:
## lm(formula = y ~ x - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8723 -0.5489 -0.0374  0.5093  3.4770
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## xBovinocultura              0.21987    0.05066   4.340 2.81e-05 ***
## x`Avicultura de Postura`    0.12995    0.03779   3.439 0.000782 ***
## xPescado                    0.06999    0.06081   1.151 0.251842
```

```
## xLácteos                   0.24139    0.07407   3.259 0.001422 **
## xSuinocultura              0.19617    0.13885   1.413 0.160058
## xcort1                     0.33894    0.07177   4.722 5.87e-06 ***
## xpos12                    -0.10120    0.03528  -2.869 0.004801 **
## xbov1                      0.07263    0.06554   1.108 0.269802
## xpes4                     -0.06333    0.06022  -1.052 0.294866
## xpes9                      0.15575    0.05692   2.736 0.007073 **
## xsui1                     -0.02063    0.12596  -0.164 0.870154
## xsui6                     -0.44858    0.10813  -4.149 5.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.973 on 132 degrees of freedom
## Multiple R-squared:  0.6242, Adjusted R-squared:  0.5901
## F-statistic: 18.27 on 12 and 132 DF,  p-value: < 2.2e-16
```

```
x3 = x[, -11]
fit3 <- summary(fit <- lm(y~x3-1))
fit3
```

```
##
## Call:
## lm(formula = y ~ x3 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8830 -0.5400 -0.0451  0.5022  3.4787
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## x3Bovinocultura            0.22028    0.05041   4.370 2.49e-05 ***
## x3`Avicultura de Postura`  0.13011    0.03764   3.457 0.000734 ***
## x3Pescado                  0.06983    0.06058   1.153 0.251114
## x3Lácteos                  0.24110    0.07377   3.268 0.001378 **
## x3Suinocultura             0.18968    0.13259   1.431 0.154903
## x3cort1                    0.33634    0.06974   4.823 3.81e-06 ***
## x3pos12                   -0.10215    0.03467  -2.947 0.003795 **
## x3bov1                     0.07035    0.06382   1.102 0.272264
## x3pes4                    -0.06341    0.06000  -1.057 0.292466
## x3pes9                     0.15534    0.05666   2.742 0.006956 **
## x3sui6                    -0.45127    0.10648  -4.238 4.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9694 on 133 degrees of freedom
## Multiple R-squared:  0.6242, Adjusted R-squared:  0.5931
## F-statistic: 20.08 on 11 and 133 DF,  p-value: < 2.2e-16
```

```
x3 = x3[, -9]
fit3 <- summary(fit <- lm(y~x3-1))
fit3
```

```
##
```

```
## Call:
## lm(formula = y ~ x3 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8976 -0.5704 -0.0741  0.4571  3.5165
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## x3Bovinocultura              0.22857    0.04982   4.588 1.02e-05 ***
## x3`Avicultura de Postura`    0.12391    0.03719   3.331  0.00112 **
## x3Pescado                    0.07928    0.05994   1.323  0.18822
## x3Lácteos                    0.21898    0.07078   3.094  0.00240 **
## x3Suinocultura               0.15239    0.12787   1.192  0.23547
## x3cort1                      0.34792    0.06890   5.049 1.42e-06 ***
## x3pos12                     -0.10034    0.03464  -2.896  0.00441 **
## x3bov1                       0.07091    0.06384   1.111  0.26870
## x3pes9                       0.16315    0.05620   2.903  0.00432 **
## x3sui6                      -0.47912    0.10321  -4.642 8.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9698 on 134 degrees of freedom
## Multiple R-squared:  0.621,  Adjusted R-squared:  0.5927
## F-statistic: 21.96 on 10 and 134 DF,  p-value: < 2.2e-16
```

```r
x3 = x3[, -8]
fit3 <- summary(fit <- lm(y~x3-1))
fit3
```

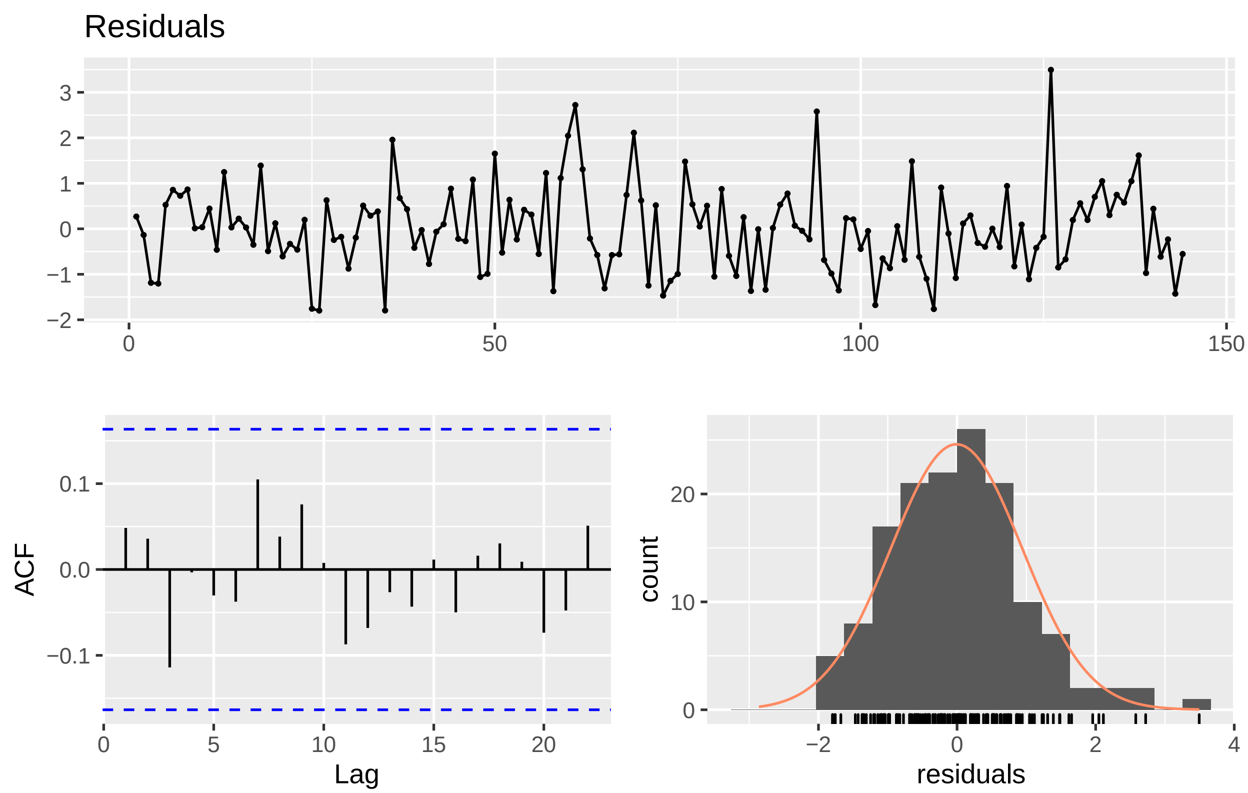
```
##
## Call:
## lm(formula = y ~ x3 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8176 -0.5785 -0.0705  0.4667  3.4514
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## x3Bovinocultura              0.25299    0.04475   5.653 8.98e-08 ***
## x3`Avicultura de Postura`    0.11915    0.03698   3.222  0.00160 **
## x3Pescado                    0.09276    0.05875   1.579  0.11671
## x3Lácteos                    0.21604    0.07079   3.052  0.00274 **
## x3Suinocultura               0.18318    0.12493   1.466  0.14491
## x3cort1                      0.37548    0.06433   5.836 3.77e-08 ***
## x3pos12                     -0.09830    0.03462  -2.839  0.00522 **
## x3pes9                       0.15961    0.05616   2.842  0.00518 **
## x3sui6                      -0.47677    0.10328  -4.616 8.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9707 on 135 degrees of freedom
```

```
## Multiple R-squared:  0.6175, Adjusted R-squared:  0.592
## F-statistic: 24.22 on 9 and 135 DF,  p-value: < 2.2e-16
```

```
x3 = x3[, -5]
fit3 <- summary(fit <- lm(y~x3-1))
fit3
```

```
##
## Call:
## lm(formula = y ~ x3 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7982 -0.6139 -0.0187  0.5330  3.4941
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## x3Bovinocultura            0.27948    0.04111   6.798 3.05e-10 ***
## x3`Avicultura de Postura`  0.11709    0.03711   3.156  0.00197 **
## x3Pescado                  0.09862    0.05886   1.675  0.09614 .
## x3Lácteos                  0.22468    0.07084   3.172  0.00187 **
## x3cort1                    0.41545    0.05852   7.099 6.33e-11 ***
## x3pos12                   -0.08983    0.03428  -2.620  0.00978 **
## x3pes9                     0.17105    0.05585   3.063  0.00264 **
## x3sui6                    -0.44790    0.10181  -4.399 2.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9748 on 136 degrees of freedom
## Multiple R-squared:  0.6114, Adjusted R-squared:  0.5886
## F-statistic: 26.75 on 8 and 136 DF,  p-value: < 2.2e-16
```

```
checkresiduals(fit3)
```



Análise das séries temporais anuais

Análise Descritiva

```
head(data_anual)
```

```
## # A tibble: 6 x 7
##    Anos `Avicultura de ~ `Avicultura Pos~ `Bovinocultura ~ Lácteos Pescado
##   <dbl>            <dbl>            <dbl>            <dbl>   <dbl>   <dbl>
## 1  2007            12.3             26.0             20.5    21.7     1.40
## 2  2008             8.33             8.27            23.7    -2.41    9.89
## 3  2009            -1.25             3.77            -3.75    4.55    7.12
## 4  2010             9.27             5.48            25.9     4.36    8.02
## 5  2011             6.21             9.15             3.67    7.51    6.61
## 6  2012            11.2             18.8              0.792   7.76   14.2
## # ... with 1 more variable: Suinocultura <dbl>
```

```
z_avc = data_anual$`Avicultura de Corte`
z_avc = ts(z_avc)

z_avp = data_anual$`Avicultura Postura`
z_avp = ts(z_avp)
```

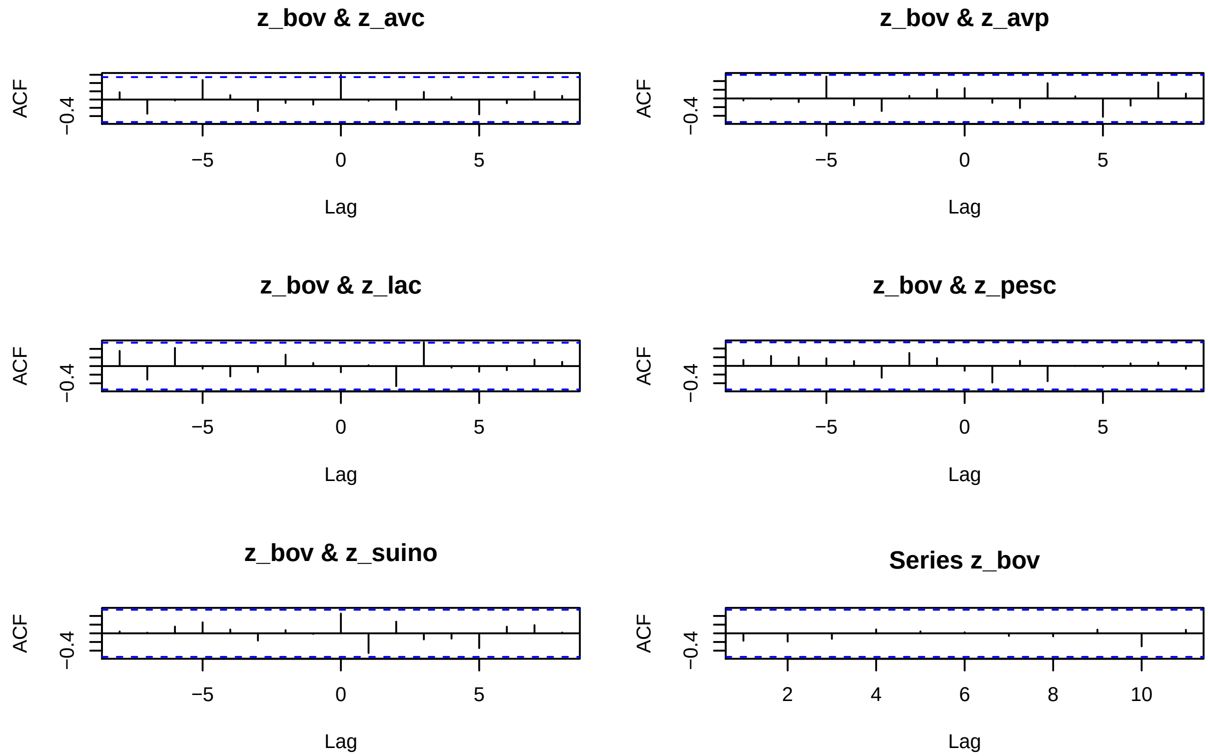
```
z_bov = data_anual$`Bovinocultura de corte`
z_bov = ts(z_bov)

z_lac = data_anual$`Lácteos`
z_lac = ts(z_lac)

z_pesc = data_anual$Pescado
z_pesc = ts(z_pesc)

z_suino = data_anual$Suinocultura
z_suino = ts(z_suino)
```

Regressão LASSO

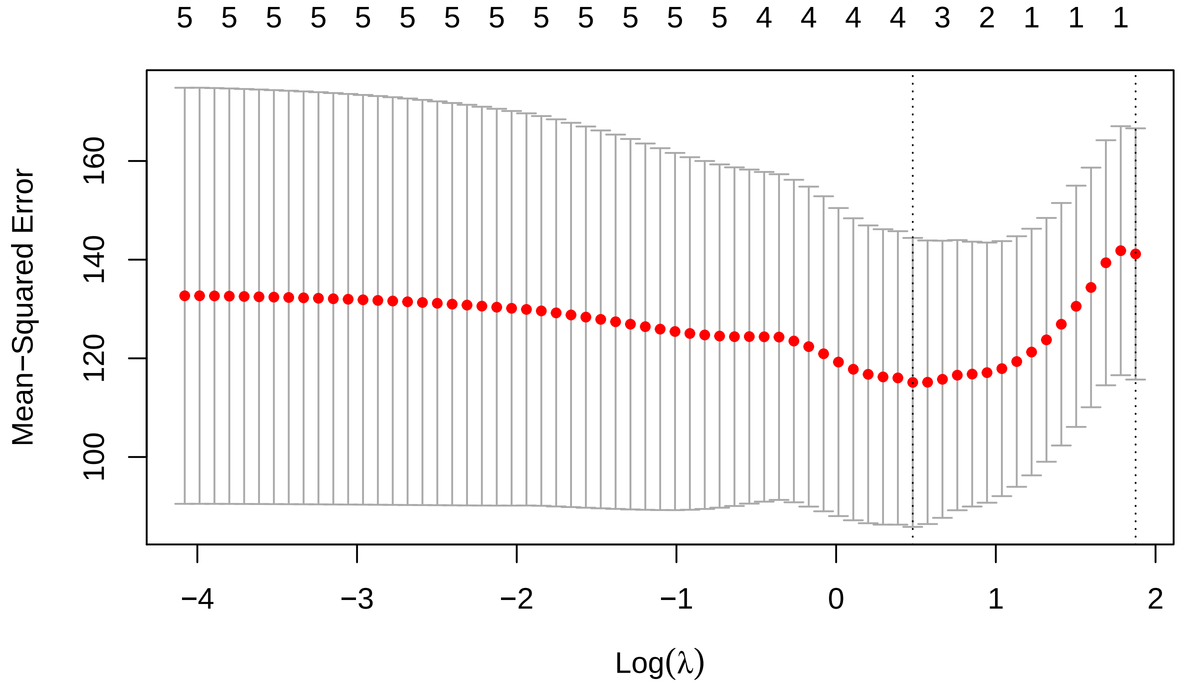
```
library(glmnet)

colnames(data_anual) = c("ANO", "AVC","AVP","BOV","LAC","PESC","SUIN")
data_anual = data_anual[,-1]
```

Modelo para Bovinocultura

```
# Bovinocultura

par(mfrow=c(3,2))
ccf(z_bov,z_avc)
ccf(z_bov,z_avp)
ccf(z_bov,z_lac)
ccf(z_bov,z_pesc)
ccf(z_bov,z_suino)
acf(z_bov)
```




```
x = model.matrix(BOV~ .,data=data_anual)[,-1]
y = data_anual$BOV

cv.model = cv.glmnet(x,y,alpha = 1)
par(mfrow=c(1,1))
plot(cv.model)
```

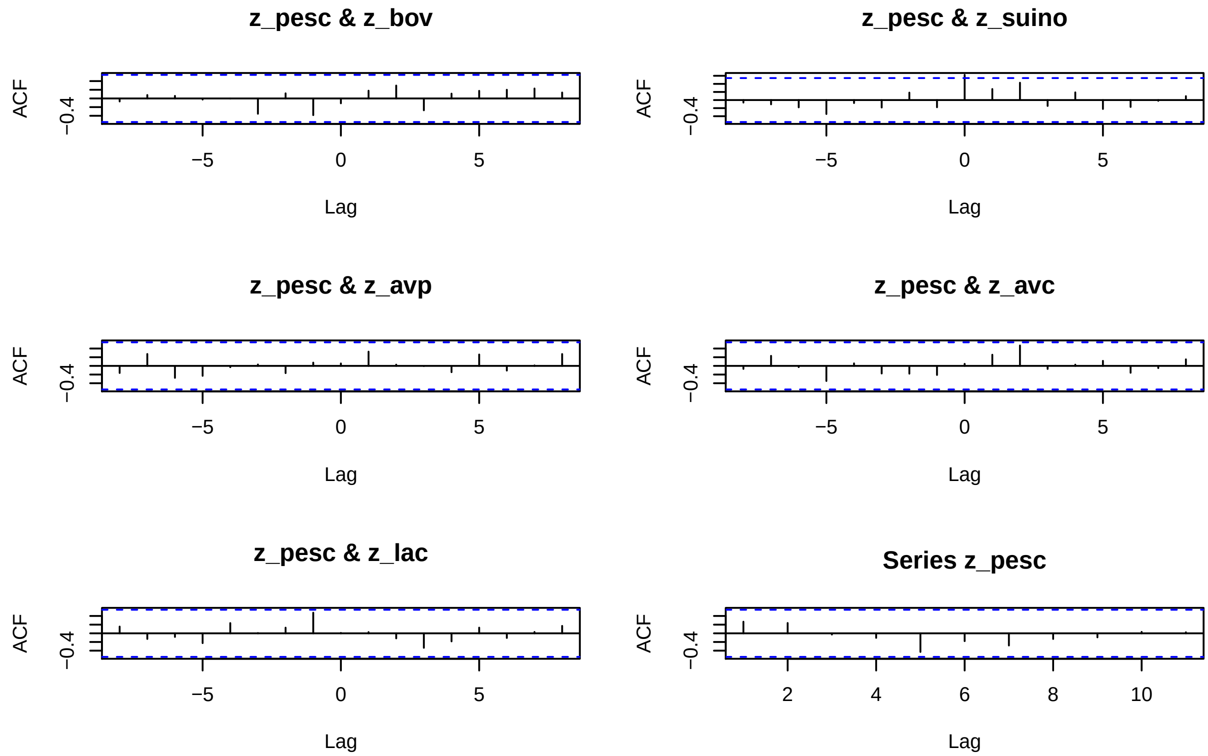



```
coef(cv.model,cv.model$lambda.min)
```

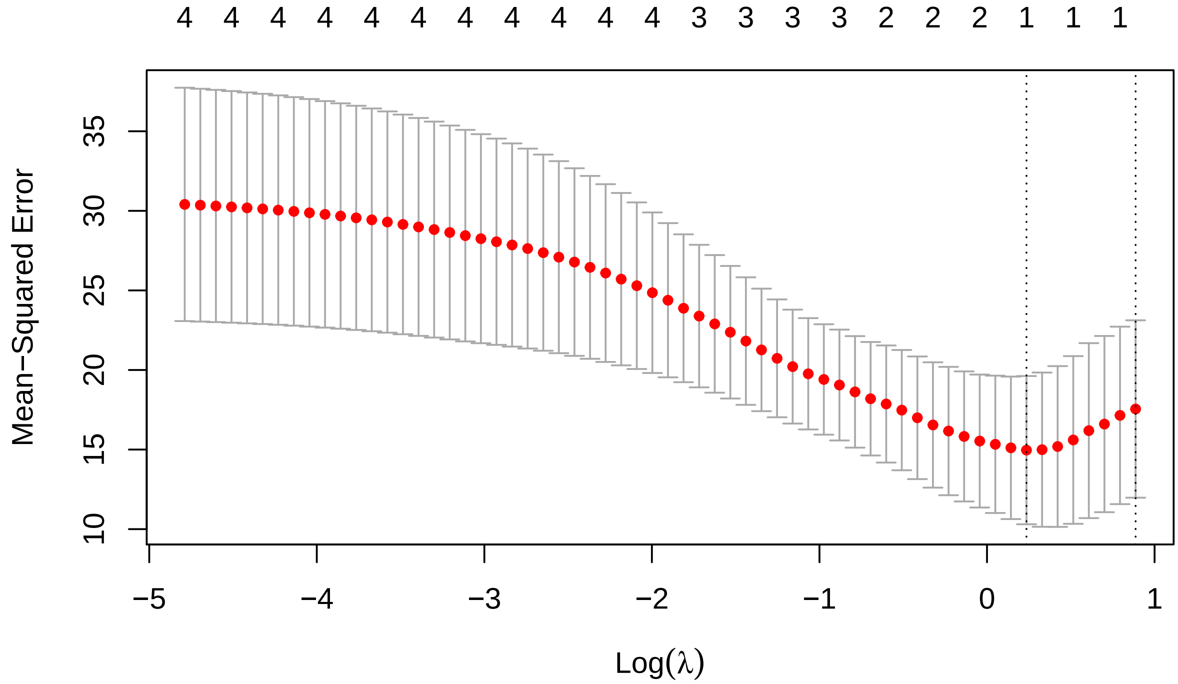
```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept)  5.49910654
## AVC          0.82215233
## AVP          .
## LAC         -0.07872937
## PESC        -0.14629328
## SUIN         0.27886874
```

Modelo para o Pescado

```
# Pescados

par(mfrow=c(3,2))
ccf(z_pesc,z_bov)
ccf(z_pesc,z_suino)
ccf(z_pesc,z_avp)
ccf(z_pesc,z_avc)
ccf(z_pesc,z_lac)
acf(z_pesc)
```




```
x = model.matrix(PESC~ .,data=data_anual)[,-1]
y = data_anual$PESC

cv.model = cv.glmnet(x,y,alpha = 1)
par(mfrow=c(1,1))
plot(cv.model)
```

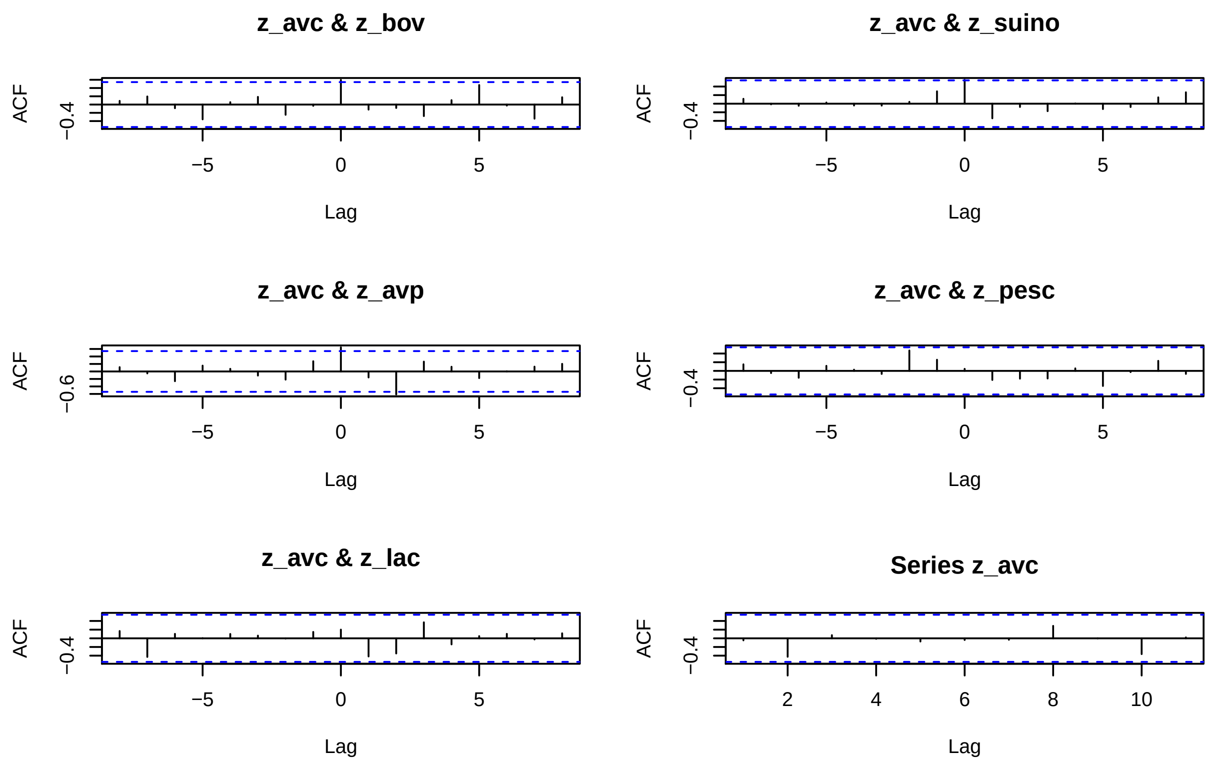



```r
coef(cv.model,cv.model$lambda.min)
```

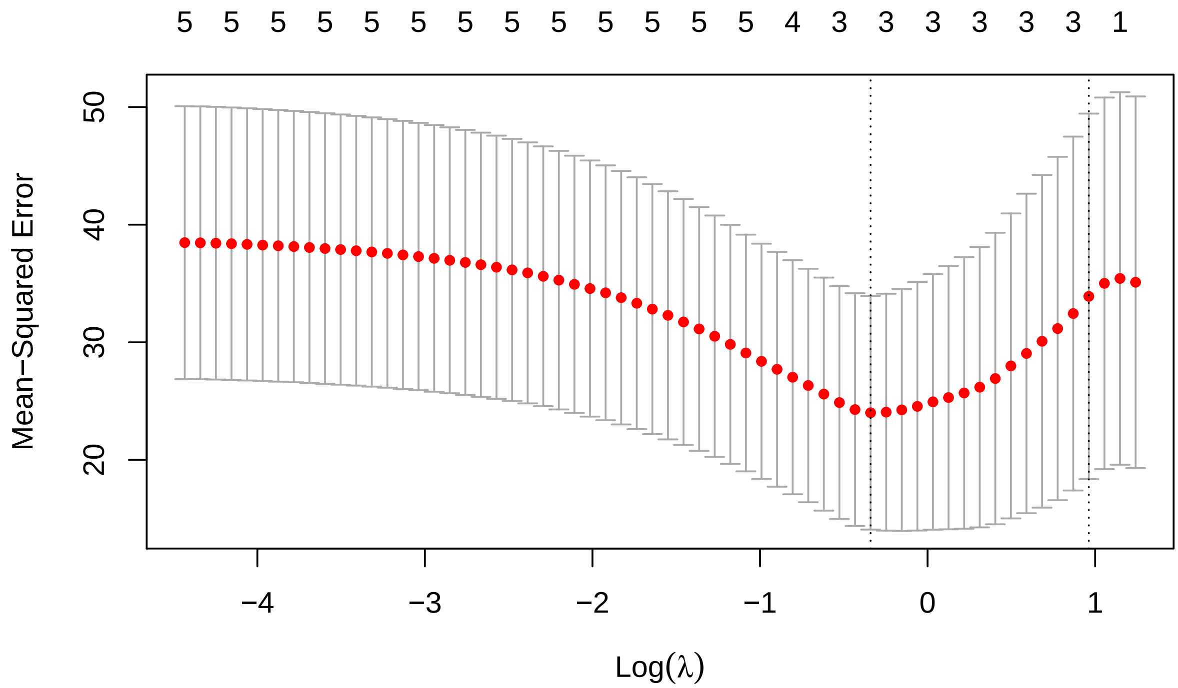
```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                    1
## (Intercept) 5.438190
## AVC         .
## AVP         .
## BOV         .
## LAC         .
## SUIN        0.266876
```

Modelo para a Avicultura de Corte

```r
# Avicultura de Corte

par(mfrow=c(3,2))
ccf(z_avc,z_bov)
ccf(z_avc,z_suino)
ccf(z_avc,z_avp)
ccf(z_avc,z_pesc)
ccf(z_avc,z_lac)
acf(z_avc)
```




```
x = model.matrix(AVC~ .,data=data_anual)[,-1]
y = data_anual$AVC

cv.model = cv.glmnet(x,y,alpha = 1)
par(mfrow=c(1,1))
plot(cv.model)
```

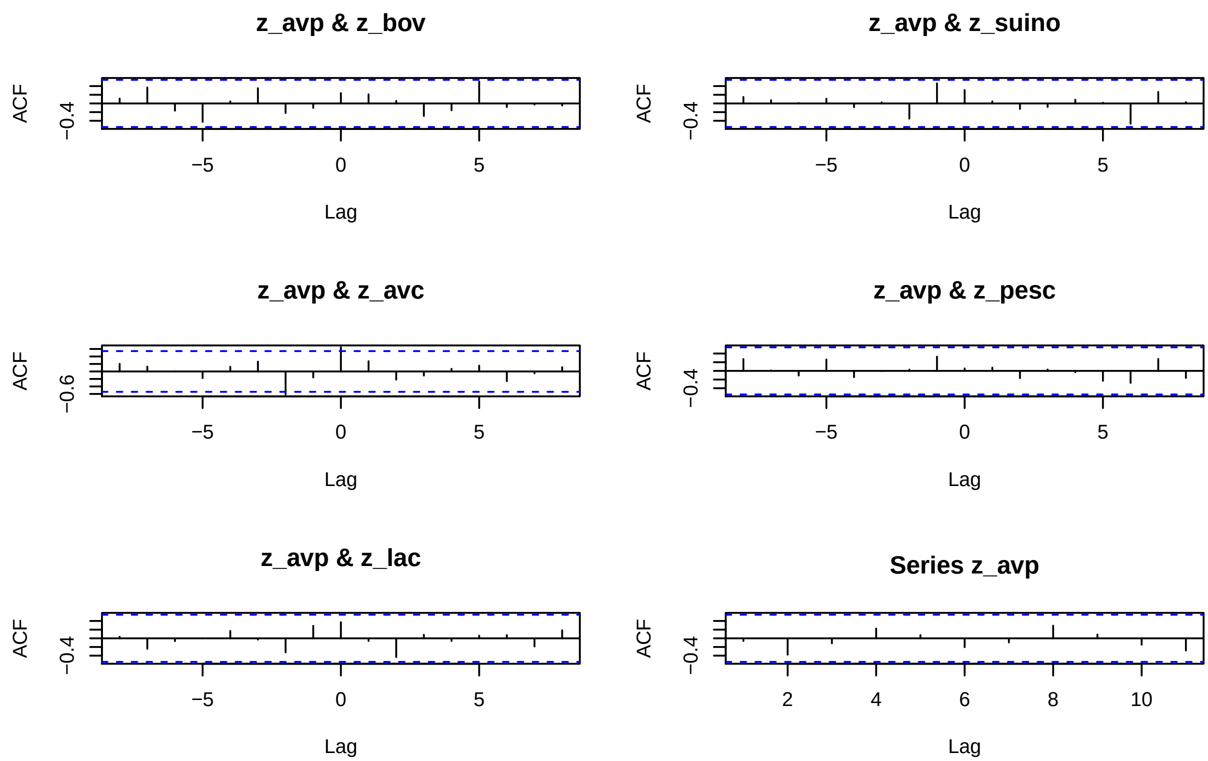



```
coef(cv.model,cv.model$lambda.min)
```

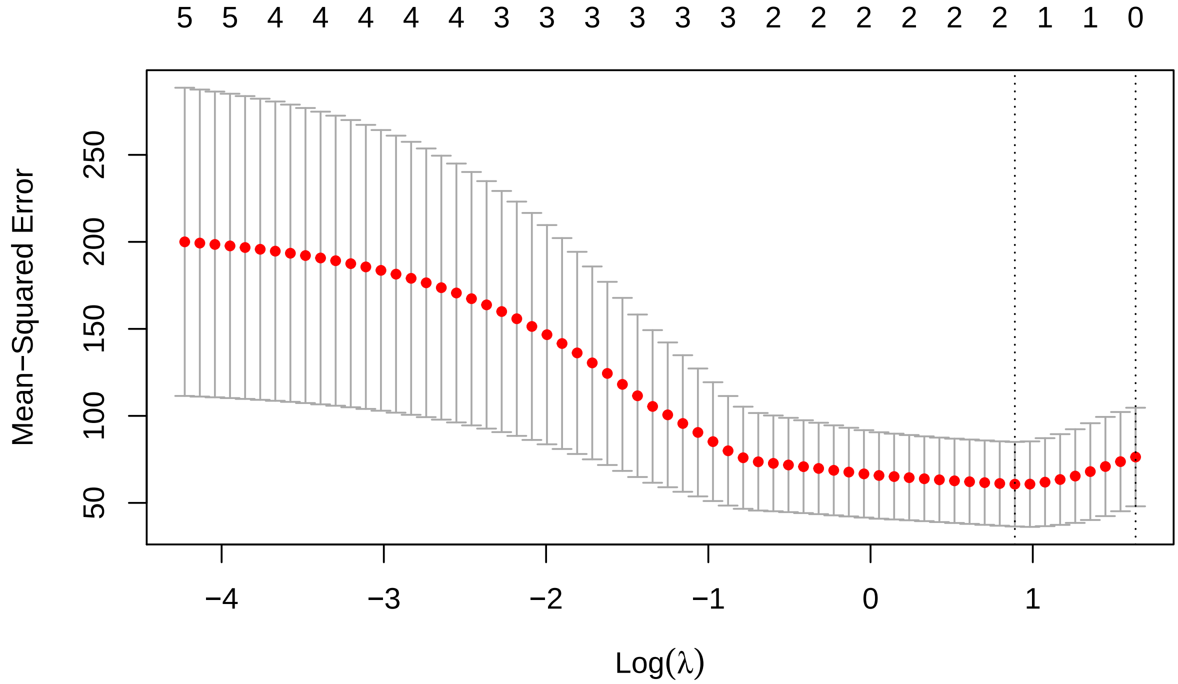
```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) 1.1062106
## AVP         0.2630295
## BOV         0.1454421
## LAC         .
## PESC        .
## SUIN        0.2064983
```

Modelo oara Avicultura de Postura

```
# Avicultura de Postura

par(mfrow=c(3,2))
ccf(z_avp,z_bov)
ccf(z_avp,z_suino)
ccf(z_avp,z_avc)
ccf(z_avp,z_pesc)
ccf(z_avp,z_lac)
acf(z_avp)
```




```
x = model.matrix(AVP~ .,data=data_anual)[,-1]
y = data_anual$AVP

par(mfrow=c(1,1))
cv.model = cv.glmnet(x,y,alpha = 1)
plot(cv.model)
```

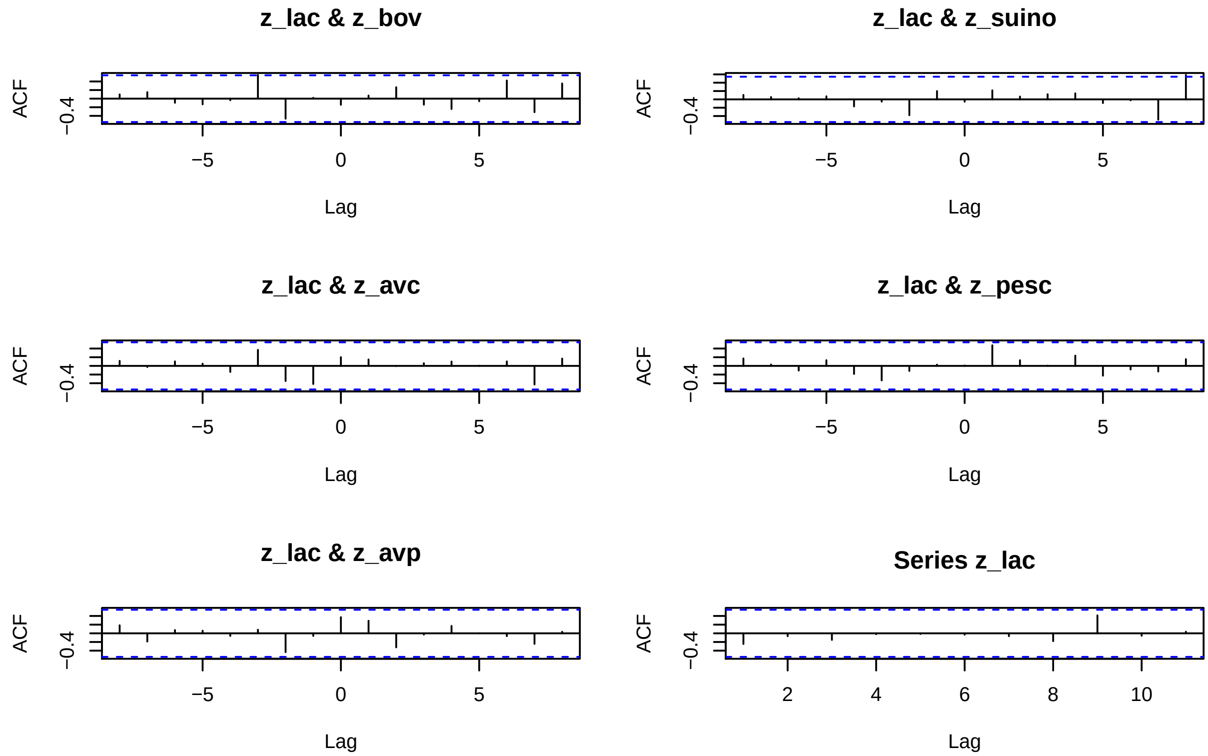



```r
coef(cv.model,cv.model$lambda.min)
```

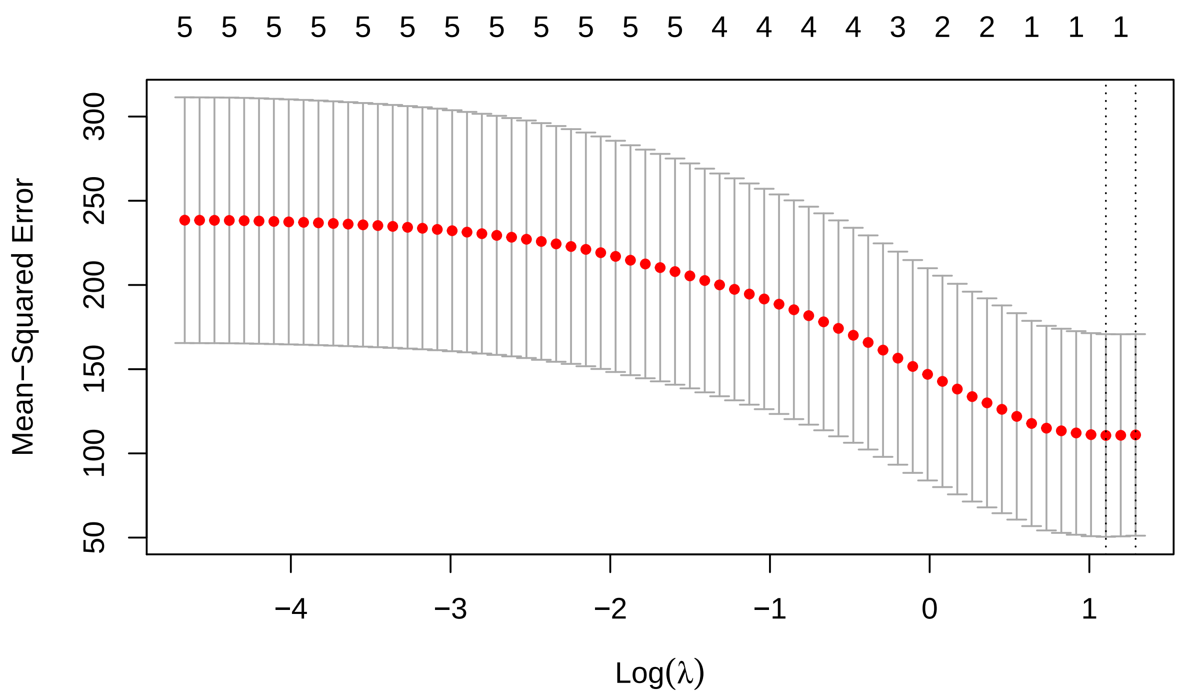
```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) 6.3392348
## AVC         0.5007544
## BOV         .
## LAC         .
## PESC        .
## SUIN        .
```

Modelo para o Lácteos

```r
# Lacteos

par(mfrow=c(3,2))
ccf(z_lac,z_bov)
ccf(z_lac,z_suino)
ccf(z_lac,z_avc)
ccf(z_lac,z_pesc)
ccf(z_lac,z_avp)
acf(z_lac)
```




```
x = model.matrix(LAC~ .,data=data_anual)[,-1]
y = data_anual$LAC

par(mfrow=c(1,1))
cv.model = cv.glmnet(x,y,alpha = 1)
plot(cv.model)
```

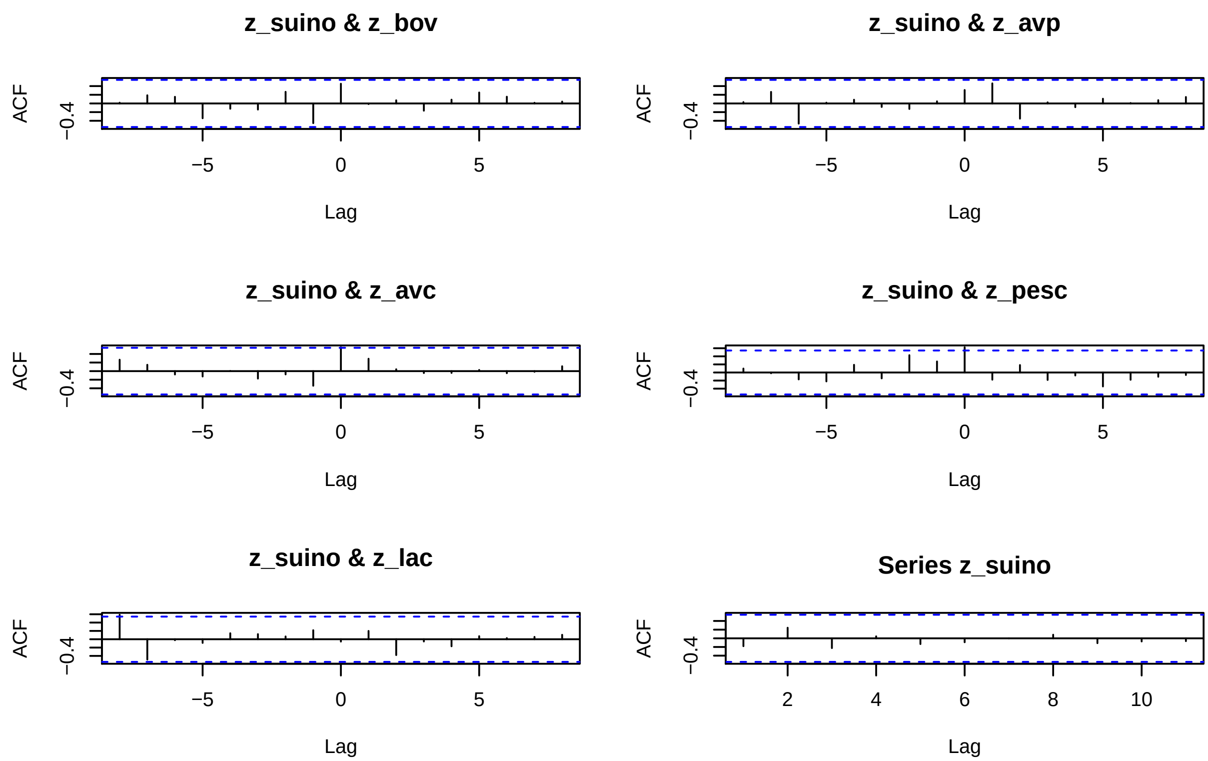



```
coef(cv.model,cv.model$lambda.min)
```

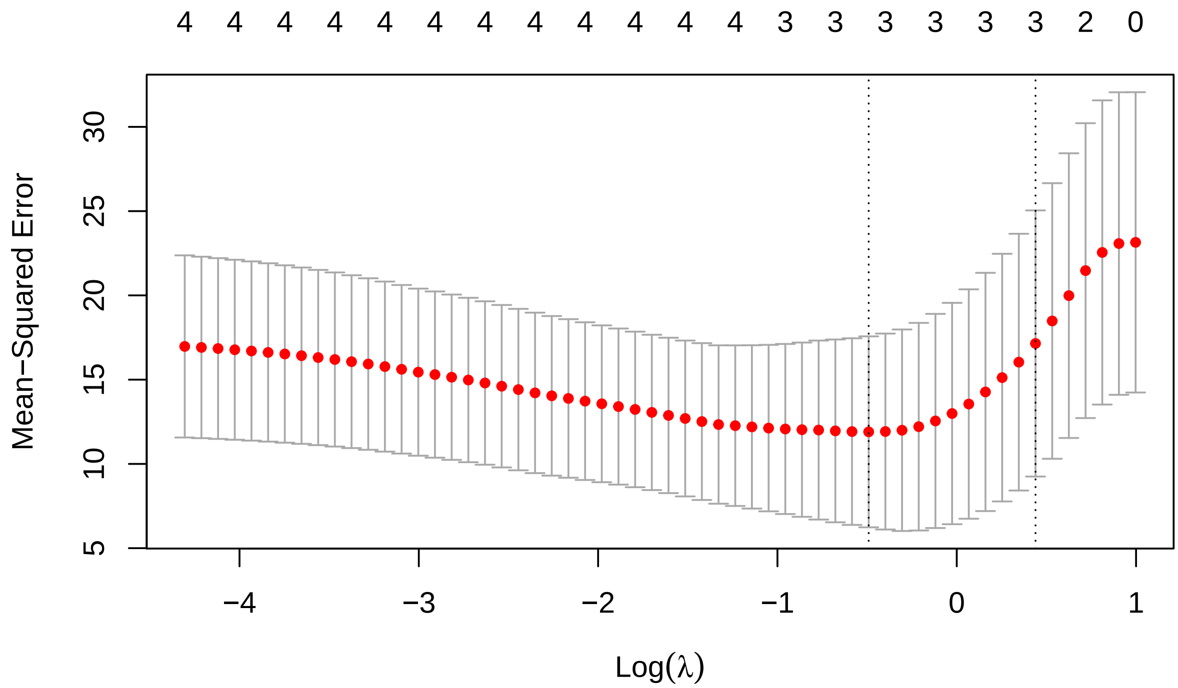
```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                      1
## (Intercept) 6.91010603
## AVC         .
## AVP         0.07756301
## BOV         .
## PESC        .
## SUIN        .
```

Modelo para Suinocultura

```
# Suinocultura

par(mfrow=c(3,2))
ccf(z_suino,z_bov)
ccf(z_suino,z_avp)
ccf(z_suino,z_avc)
ccf(z_suino,z_pesc)
ccf(z_suino,z_lac)
acf(z_suino)
```




```
x = model.matrix(SUIN~ .,data=data_anual)[,-1]
y = data_anual$SUIN

par(mfrow=c(1,1))
cv.model = cv.glmnet(x,y,alpha = 1)
plot(cv.model)
```




```r
coef(cv.model,cv.model$lambda.min)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) -0.33554640
## AVC          0.21438530
## AVP          .
## BOV          0.08463707
## LAC          .
## PESC         0.55149018
```