

Estudo do comportamento do preço do milho no período de janeiro de 2005 a dezembro de 2019

Yago Matias (bolsista PUB), Cristian Villegas (supervisor: clobos@usp.br)

Escola Superior de Agricultura Luiz de Queiroz

Contents

1	Carregando os Pacotes	1
2	Gráfico da série temporal para verificar a estacionaridade	2
3	Transformando a serie em estacionária	4
4	Gráfico das difereças para verificar a estacionaridade	6
5	Gráficos do modelo ARIMA(1,1,1) e previsão	7

1 Carregando os Pacotes

```
rm(list = ls())
milho <- read_excel("milho.xlsx", col_types = c("numeric", "numeric", "numeric"))
```

A funcao TS transforma o banco de dados milho, que é um data.frame, em um formato ts (serie temporal)

```
Milho <- ts(milho, start= c(2005,1), end = c(2018,12), frequency= 12)
```

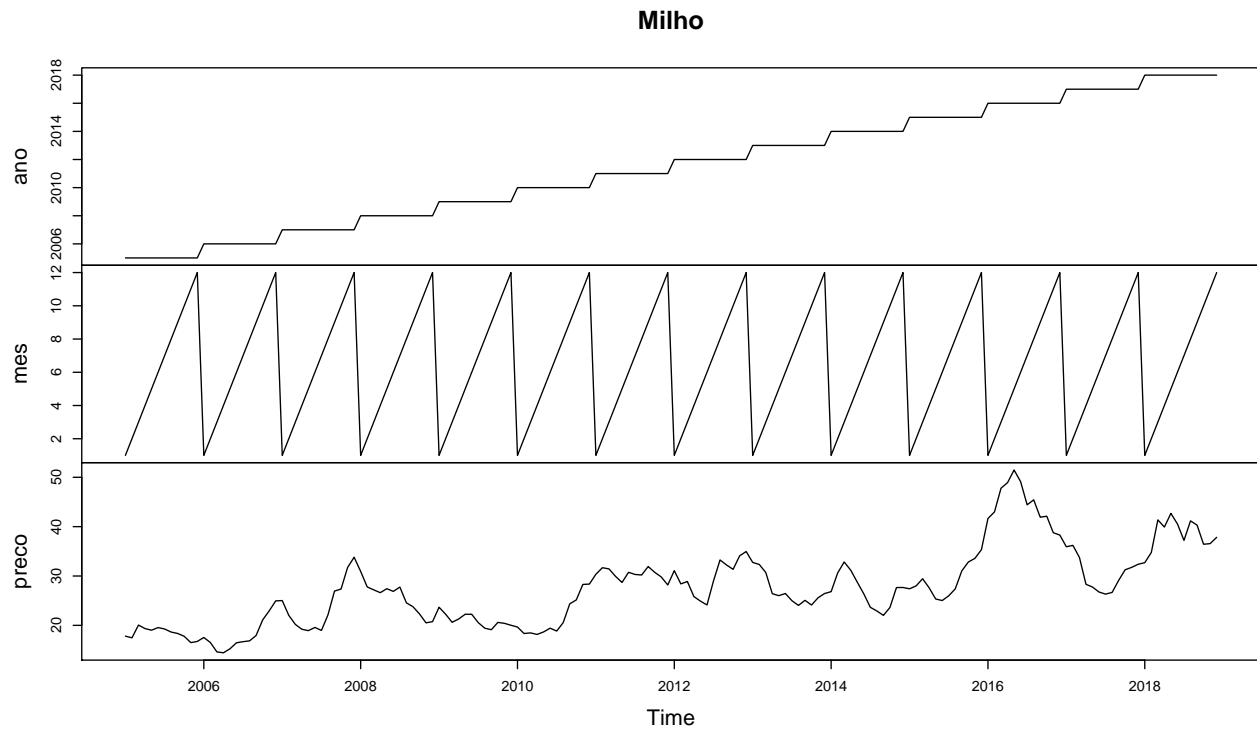
Pela funcao class, é possível saber como o R esta lendo o conjunto de dados

```
class(Milho)
```

```
[1] "mts"    "ts"     "matrix"
```

A funcao plot permite fazer um grafico de serie temporal.

```
plot(Milho)
```



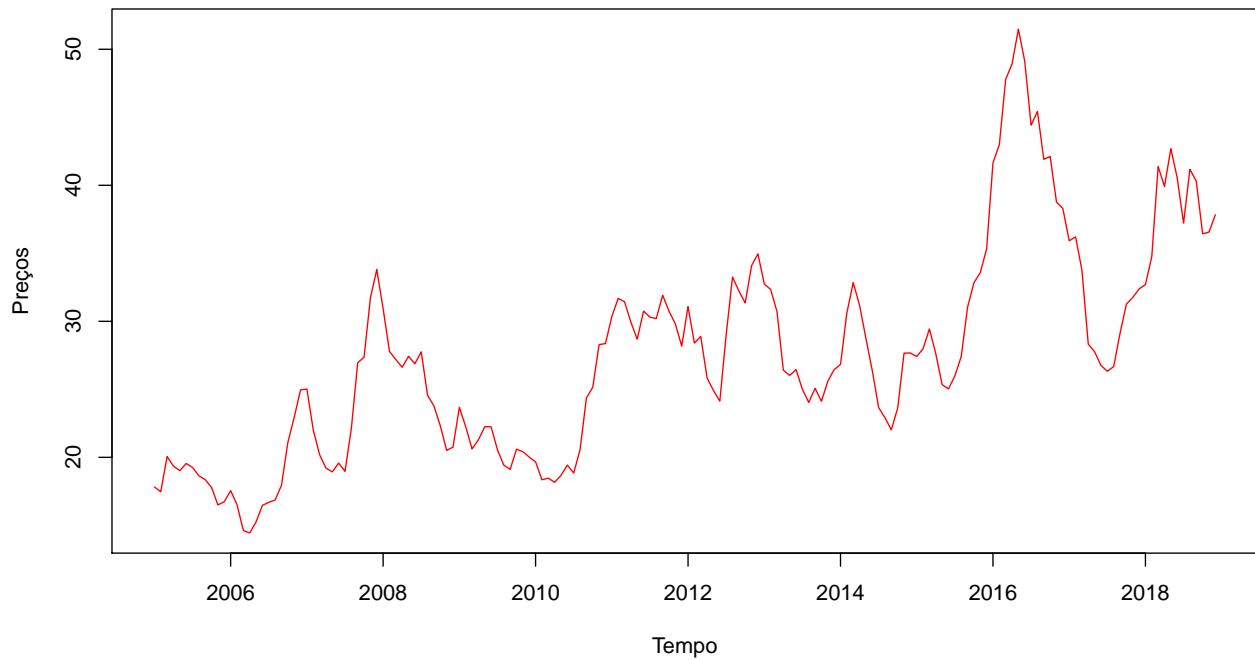
O plot feito acima fez um gráfico para as 3 variáveis do banco de dados, mas como eu quero apenas a variável preço, seleciona-se a coluna 3

```
pmilho <- Milho [,3]
```

2 Gráfico da série temporal para verificar a estacionaridade

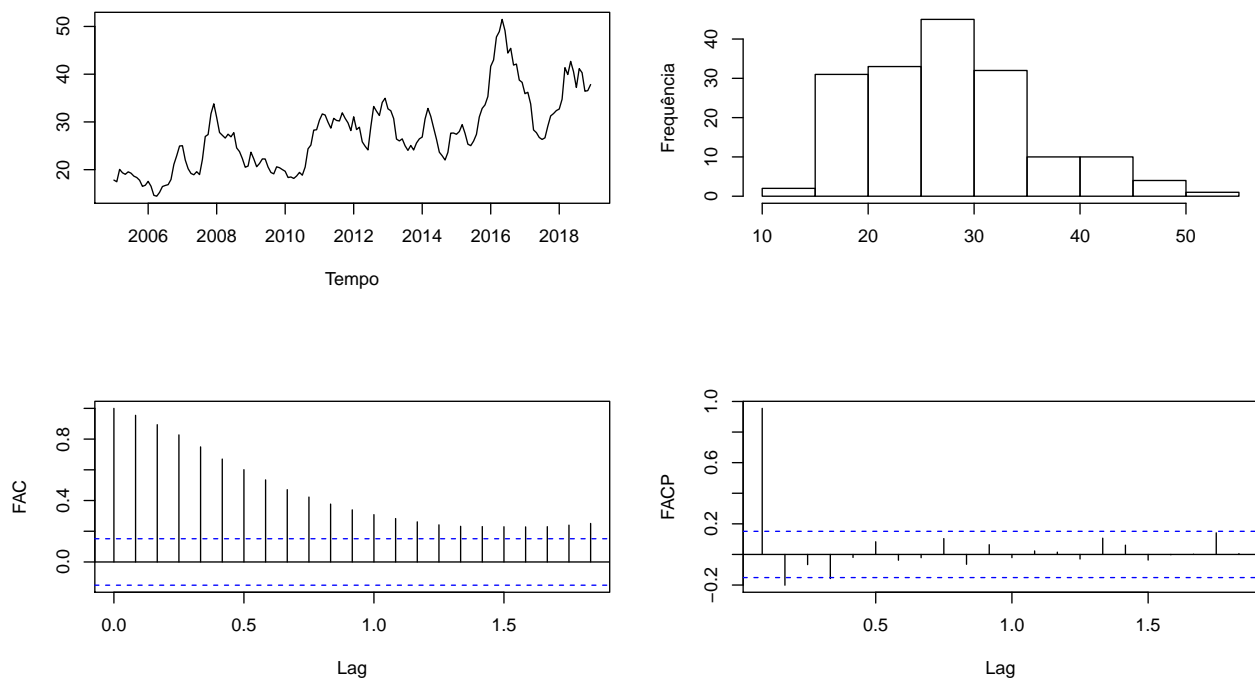
```
plot(pmilho,xlab='Tempo',ylab='Preços',
     main = "Série mensal de preços da saca de milho no período de janeiro de
     2005 a dezembro de 2019",
     type= "l", col= "red")
```

Série mensal de preços da saca de milho no período de janeiro de 2005 a dezembro de 2019



Com base no comportamento do gráfico os dados não são estacionários, então é preciso realizar uma transformação

```
par(mfrow=c(2,2))
plot(pmilho,xlab='Tempo',ylab='')
hist(pmilho,main='',ylab='Frequência',xlab='')
acf(pmilho, main='',ylab='FAC')
pacf(pmilho, main='',ylab='FACP')
```



3 Transformando a serie em estacionária

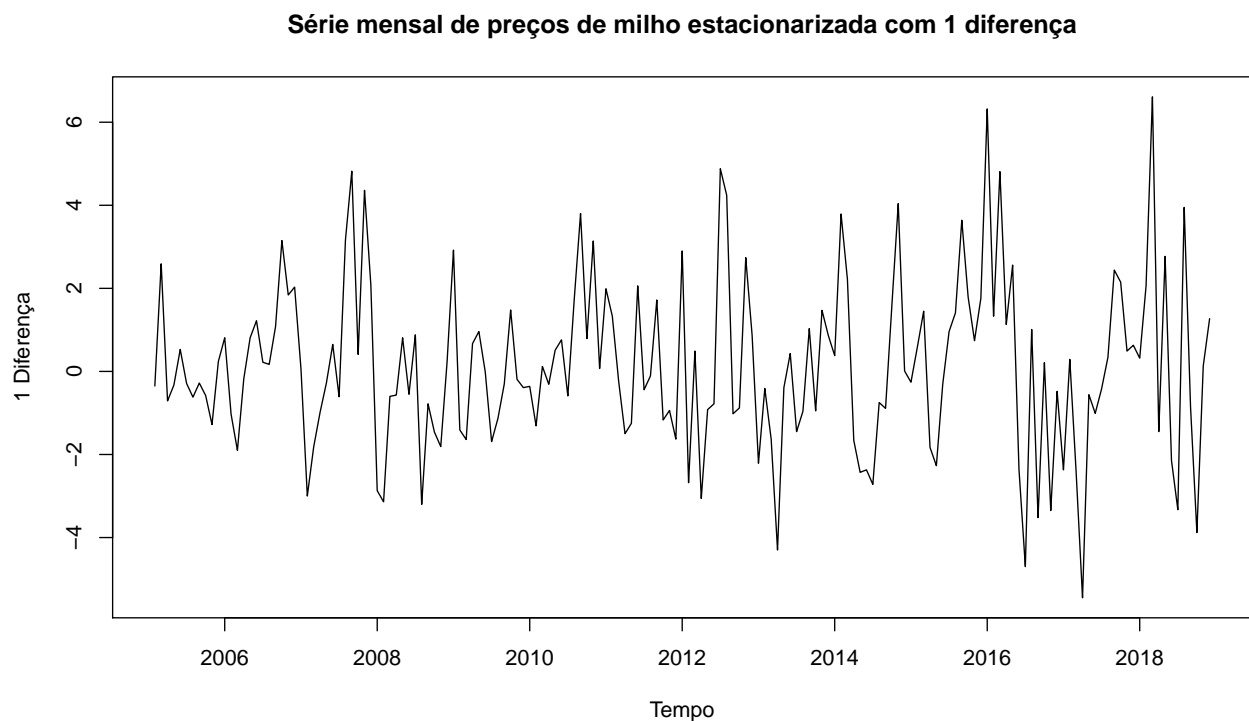
A função `ndiffs` usa um teste de raiz unitária para determinar o número de diferenças necessárias para que séries temporais sejam feitas estacionárias. A hipótese nula é que x tem uma raiz unitária contra uma alternativa de série estacionária. Em seguida, o teste retorna o menor número de diferenças necessárias para falhar o teste no nível α .

```
ndiffs(pmilho,alpha=0.05,test="adf")
```

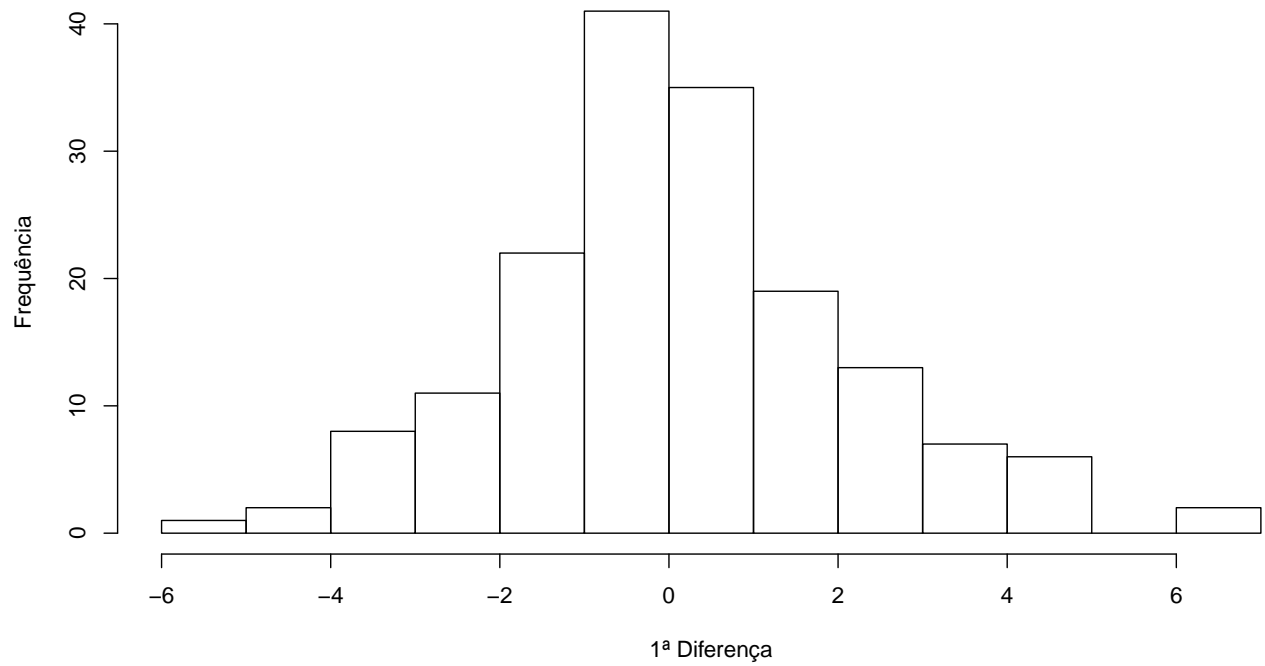
```
[1] 1
```

```
d1s <- diff(pmilho, differences = 1)
```

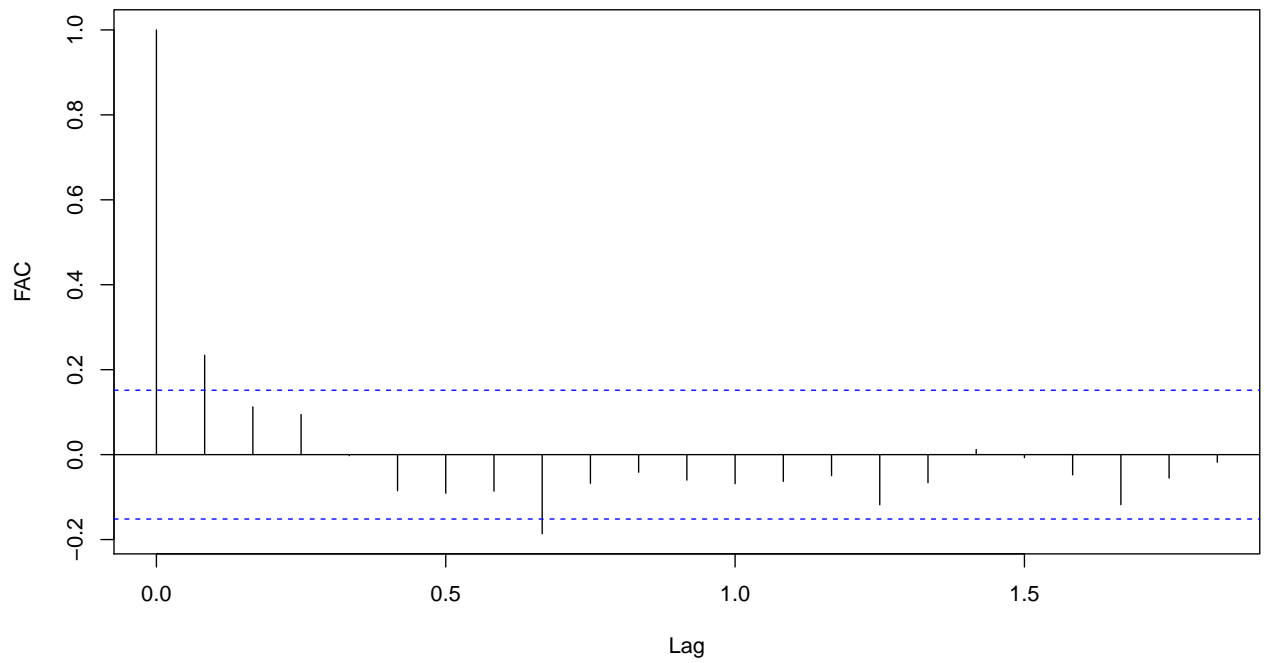
```
par(mfrow=c(1,1))  
plot(d1s,xlab='Tempo',ylab='1 Diferença',  
     main= "Série mensal de preços de milho estacionarizada com 1 diferença" )
```



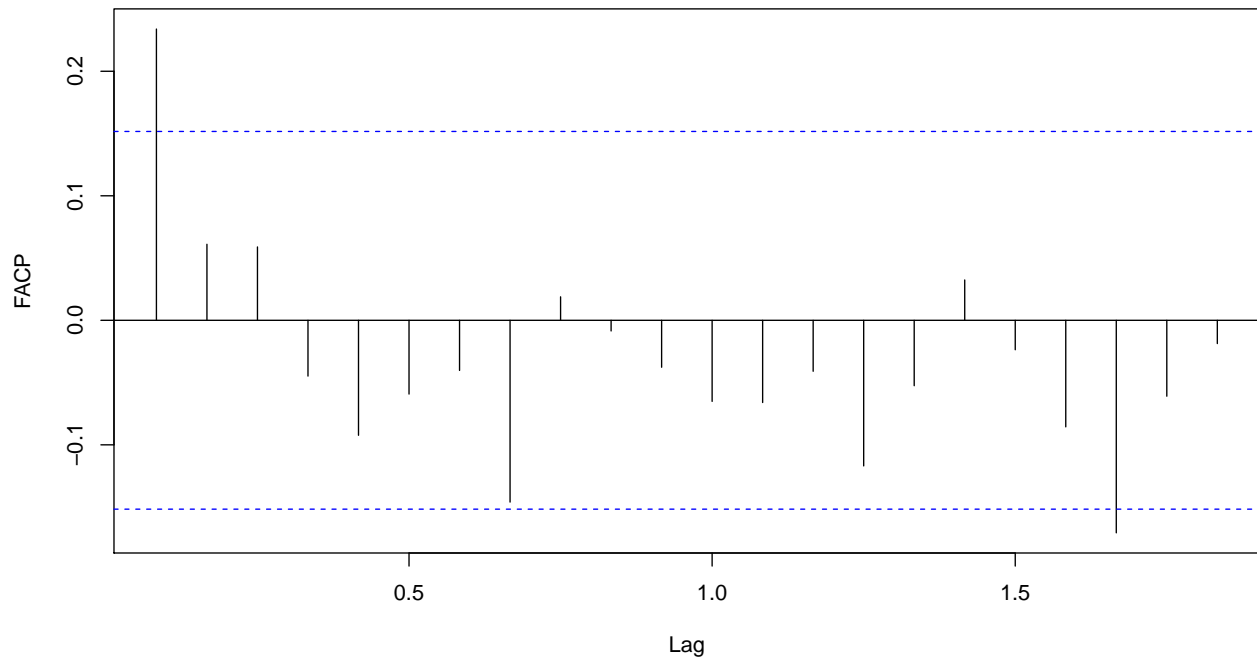
```
hist(d1s,main='',ylab='Frequência',xlab='1ª Diferença')
```



```
acf(d1s,main='',ylab='FAC') #Média móvel
```



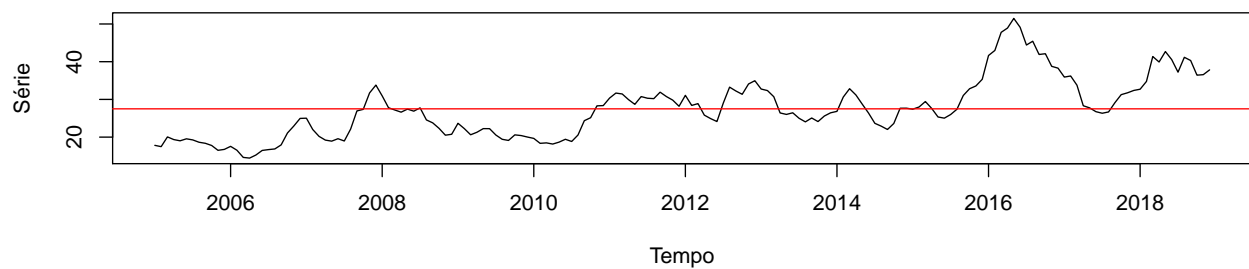
```
pacf(d1s,main='',ylab='FACP') #Lag
```



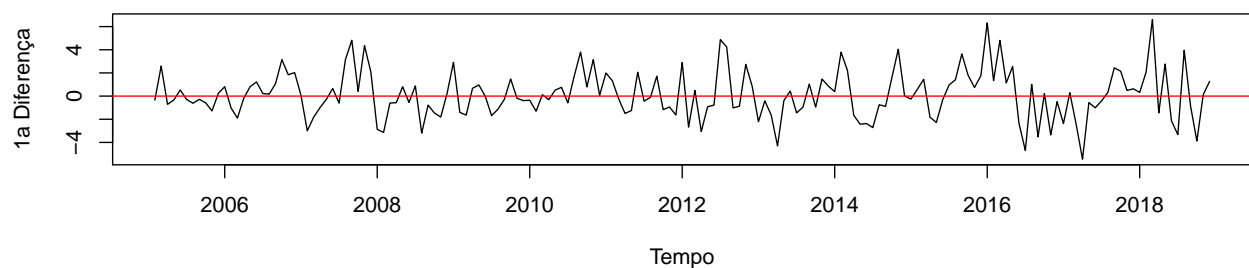
4 Gráfico das diferenças para verificar a estacionaridade

```
par(mfrow=c(2,1))
plot(pmilho,xlab='Tempo',ylab='Série', main='(a)')
abline(h=mean(pmilho),col='red')
plot(d1s,xlab='Tempo',ylab='1a Diferença', main='(b)')
abline(h=0,col='red')
```

(a)



(b)



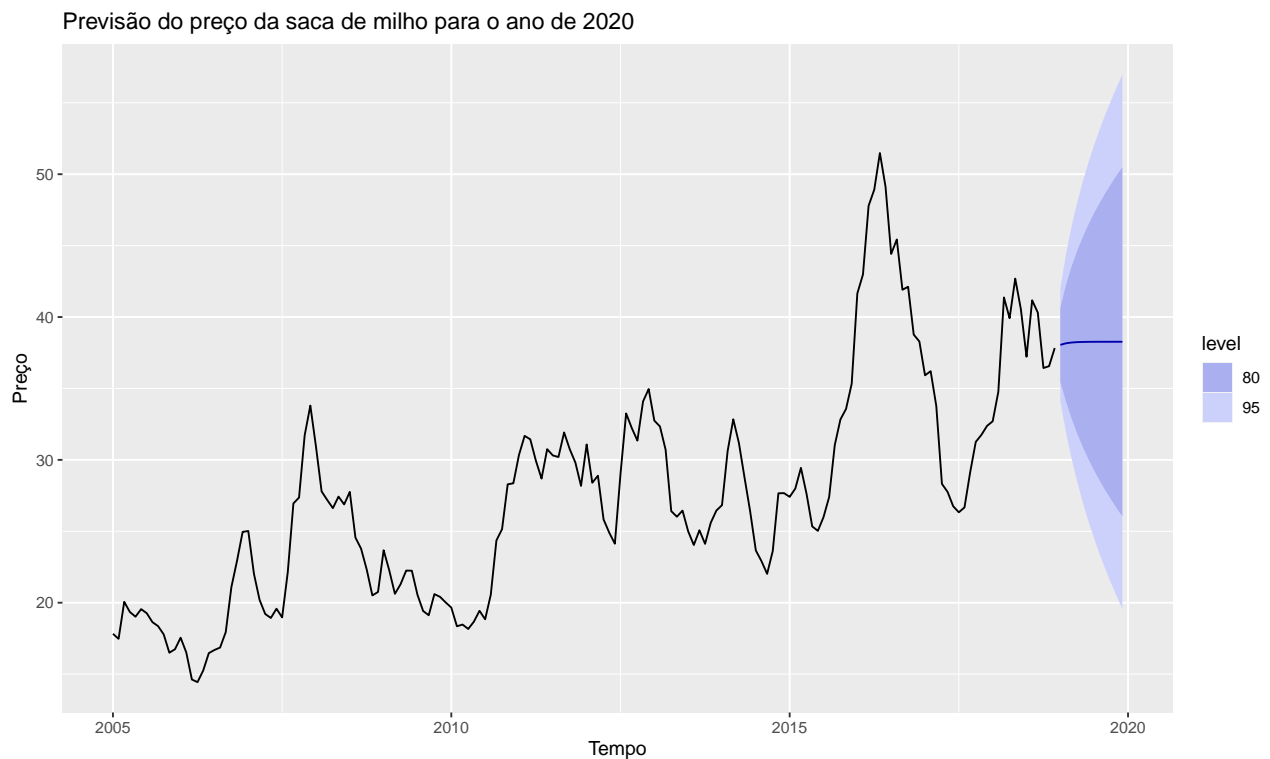
5 Gráficos do modelo ARIMA(1,1,1) e previsão

```
milho2 <- read_excel("milho2.xlsx",
                     col_types = c("numeric", "numeric", "numeric"))

ano2 <- ts(milho2, start= c(2019,1),
           end = c(2019,12), frequency= 12)

a <- milho2 [, 3]

fit6<- forecast::Arima(pmilho, order=c(1,1,1))
fcast_cons6 <-forecast(fit6, h= 12)
autoplot(fcast_cons6, ylab="Preço",
         xlab= "Tempo",
         main= "Previsão do preço da saca de milho para o ano de 2020")
```



```
data.frame(a, fcast_cons6)
```

	preco	Point.Forecast	Lo.80	Hi.80	Lo.95	Hi.95
Jan 2019	38.91	38.05284	35.47148	40.63419	34.10499	42.00068
Feb 2019	40.89	38.16264	34.09147	42.23381	31.93632	44.38896
Mar 2019	39.82	38.21675	32.89490	43.53859	30.07768	46.35581
Apr 2019	36.42	38.24341	31.83778	44.64903	28.44685	48.03997
May 2019	34.84	38.25654	30.89243	45.62066	26.99410	49.51899
Jun 2019	38.04	38.26302	30.03670	46.48933	25.68196	50.84408
Jul 2019	37.10	38.26621	29.25314	47.27927	24.48192	52.05050
Aug 2019	36.41	38.26778	28.52826	48.00730	23.37247	53.16309

Sep 2019	37.64	38.26855	27.85171	48.68540	22.33737	54.19974
Oct 2019	41.51	38.26894	27.21554	49.32233	21.36422	55.17365
Nov 2019	44.54	38.26912	26.61358	49.92467	20.44350	56.09474
Dec 2019	48.16	38.26922	26.04099	50.49744	19.56776	56.97067

```
teste3<- data.frame(x=milho$preco,
                    y=c(fcast_cons6$fitted,fcast_cons6$mean))
matplot(teste3, type="l",col=1:2, lwd=2)
```

