

Data Analysis Skill Test

Francisco Jucelino Matos Junior

04/07/2020

Pacotes a serem utilizados

```
library(pwt8)
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo

library(tseries)
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric

library(lattice)
library(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union

library(ggplot2)
```

Case 1

Lendo o conjunto de dados

Análise Exploratória

Inicialmente, farei uma análise individual da série temporal de cada país.

USA

Dados apenas dos EUA

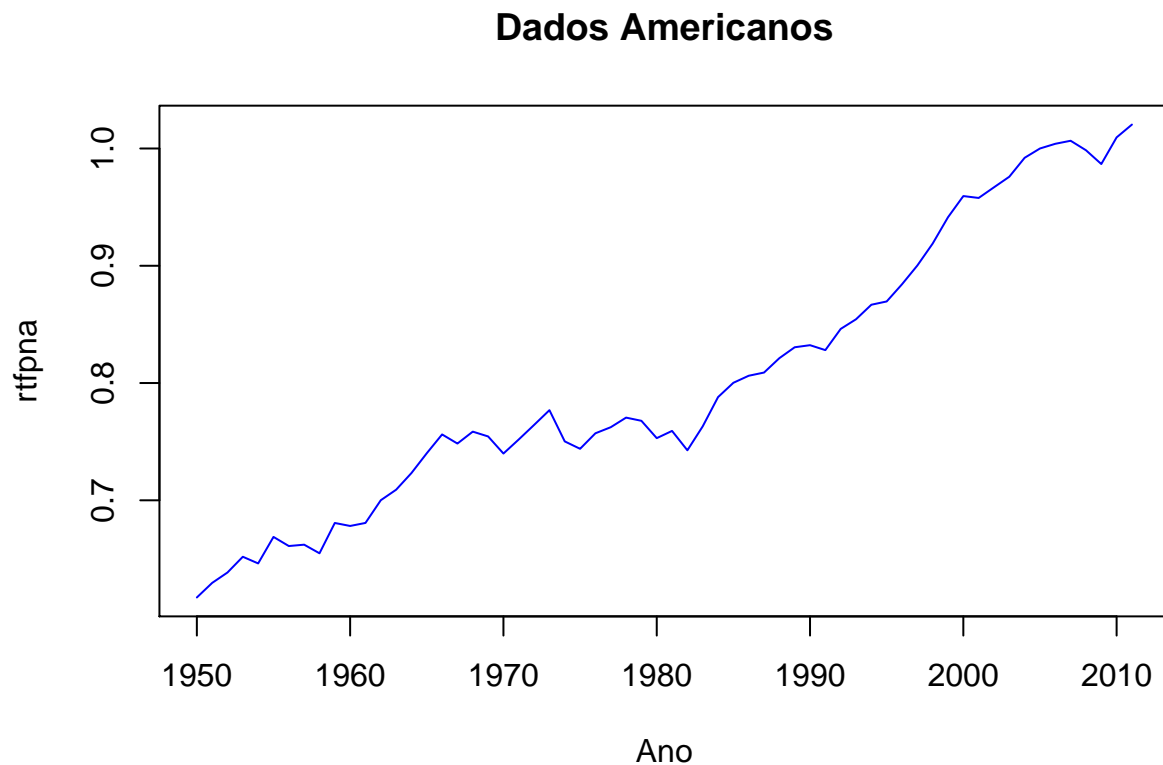
```
data_usa = subset(case1_data, case1_data$isocode == "USA")
```

Criamos uma série temporal da variável rtfpna apenas dos dados dos EUA

```
serie_usa = ts(data.frame(rtfpna = data_usa$rtfpna), start = 1950)
```

Gráfico da série temporal da variável rtfpna para os dados americanos

```
plot(serie_usa, col='blue',  
     main='Dados Americanos',  
     xlab='Ano')
```



Observando o gráfico, de modo geral, podemos ver um crescimento nos preços nacionais americanos no decorrer dos últimos 60 anos.

Canadá

Dados apenas do Canadá

```
data_canada = subset(case1_data, case1_data$isocode == "CAN")
```

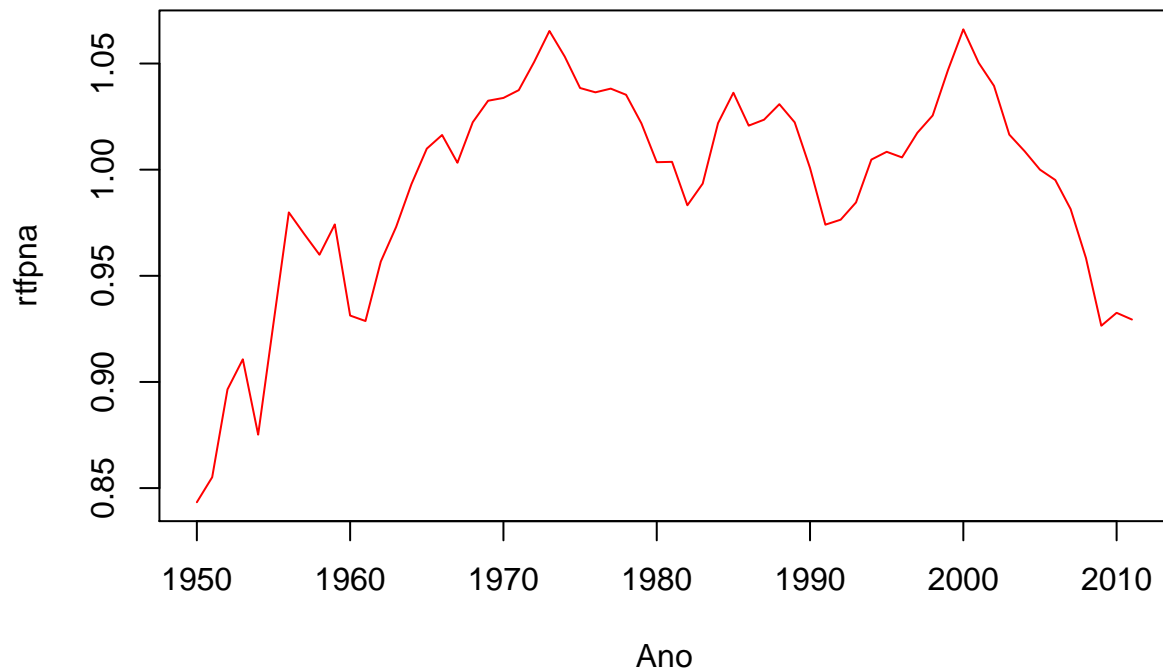
Criamos uma série temporal da variável rtfpna apenas dos dados dos Canadá

```
serie_canada = ts(data.frame(rtfpna = data_canada$rtfpna), start = 1950)
```

Gráfico da série temporal da variável rtfpna para os dados canadenses

```
plot(serie_canada, col='red',  
     main='Dados Canadenses',  
     xlab='Ano')
```

Dados Canadenses



Observando o gráfico, pode-se afirmar que houve um crescimento nos preços nacionais até os anos 70, em seguida houve, de modo geral, uma queda nos preços até o início dos anos 90. Seguimos com um alta dos preços nacionais até os anos 2000, seguido por uma queda até o ano de 2011.

México

Dados apenas dos México

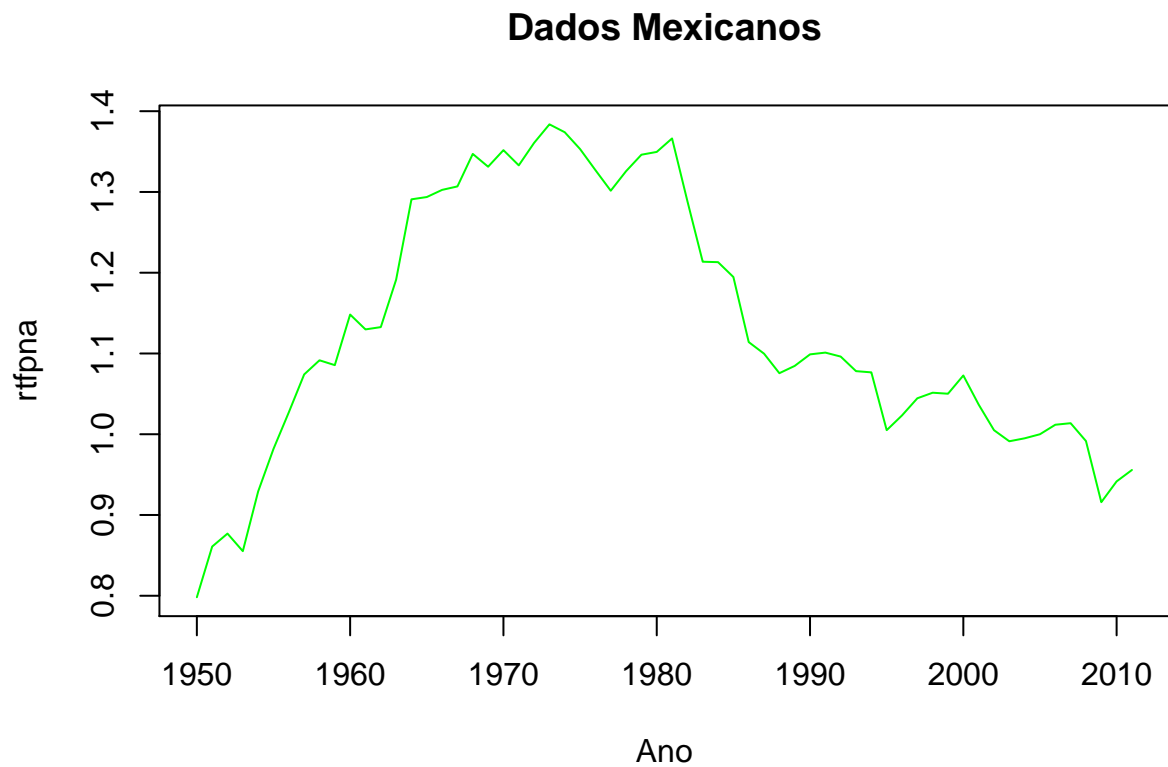
```
data_mexico = subset(case1_data, case1_data$isocode == "MEX")
```

Criamos uma série temporal da variável rtfpna apenas dos dados dos México

```
serie_mexico = ts(data.frame(rtfpna = data_mexico$rtfpna), start = 1950)
```

Gráfico da série temporal da variável rtfpna para os dados mexicanos

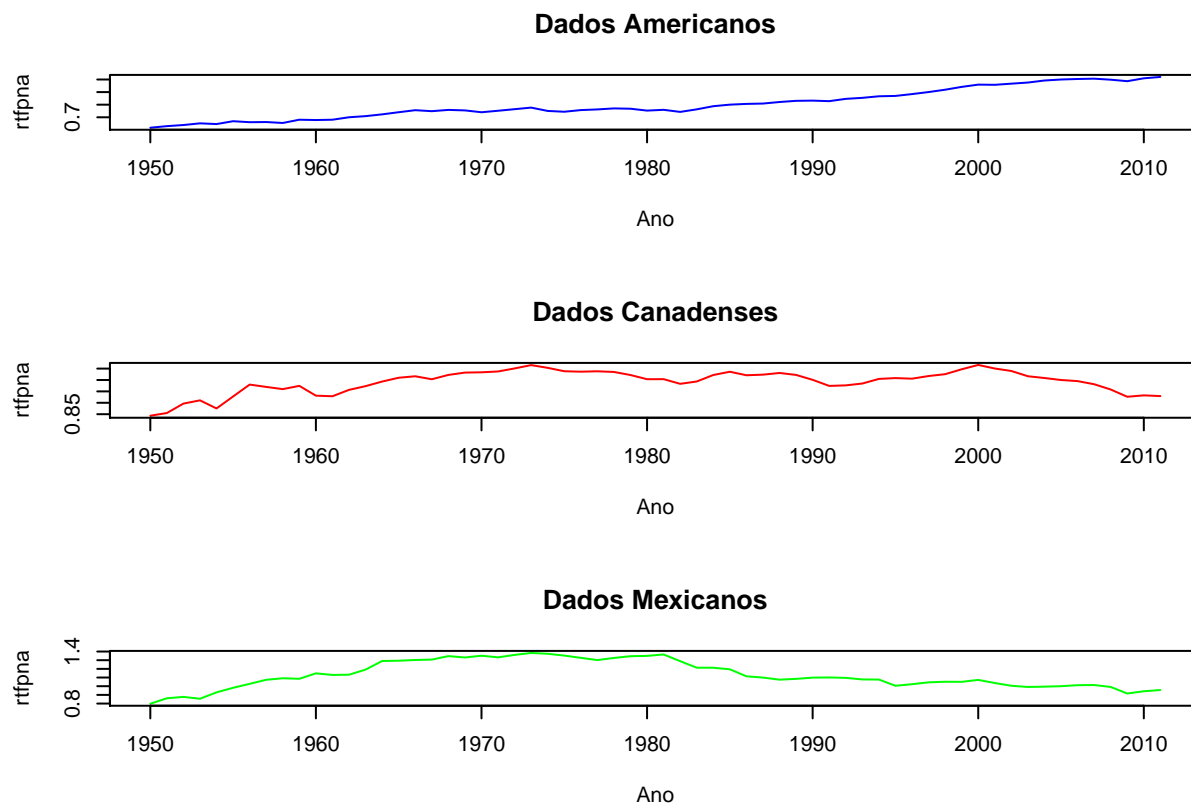
```
plot(serie_mexico, col='green',  
     main='Dados Mexicanos',  
     xlab='Ano')
```



Observando o gráfico, temos que os preços nacionais no México seguiram em alta até os anos 80, em que, em seguida, vem tendo um comportamento decrescente.

O Gráfico a seguir apresenta as séries temporais da variável rtfpna para os três países em estudo.

```
par(mfrow = c(3,1))
plot(serie_usa, col='blue',
     main='Dados Americanos',
     xlab='Ano')
plot(serie_canada, col='red',
     main='Dados Canadenses',
     xlab='Ano')
plot(serie_mexico, col='green',
     main='Dados Mexicanos',
     xlab='Ano')
```



Comparando as três séries dos preços nacionais dos EUA, Canadá e México, podemos observar que, os EUA vem seguindo com um crescimento gradual dos seus preços, em quanto que, o Canadá, apesar de possuir uma série de altas de seus preços, a partir dos anos 2000, observa-se uma queda gradual desses preços. No México, por outro lado, teve sua época de aumento de seu preço nacional até o início dos anos 80, para em seguida, iniciar uma queda gradual de seus preços.

Previsões para os próximos 10 anos

Inicialmente, precisaremos modelar os dados. Opto por uma modelagem individual da série temporal de cada país. Estimarei um modelo ARIMA para os dados do preço nacional usando a função `auto.arima()`.

USA

Inicialmente, ajusto o modelo e mostraremos as estimativas para o mesmo

```
modelo_usa <- auto.arima(data_usa$rtfpna)
summary(modelo_usa)
```

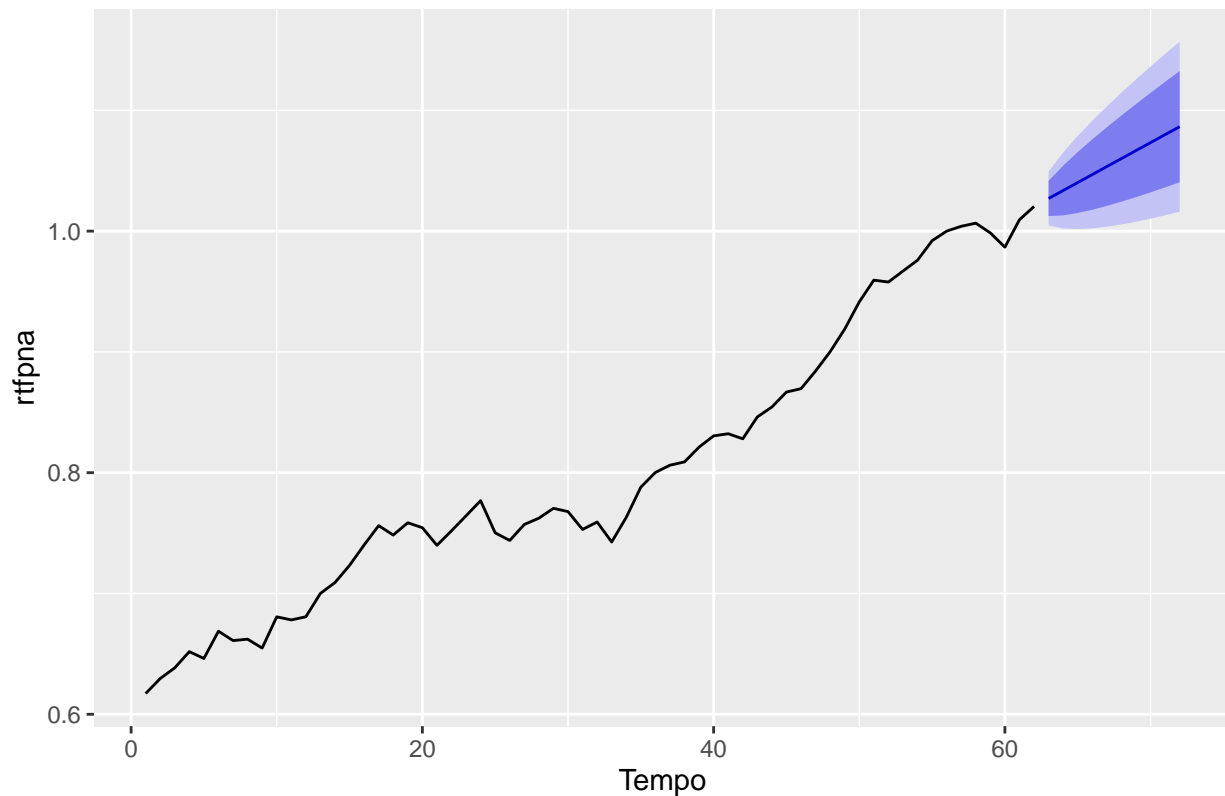
```
## Series: data_usa$rtfpna
## ARIMA(0,1,0) with drift
##
## Coefficients:
##      drift
##      0.0066
## s.e.  0.0014
##
## sigma^2 estimated as 0.0001291:  log likelihood=187.09
```

```
## AIC=-370.17   AICc=-369.97   BIC=-365.95
##
## Training set error measures:
##           ME           RMSE           MAE           MPE           MAPE           MASE
## Training set 9.847371e-06 0.01117543 0.009022366 -0.02282104 1.154826 0.799736
##           ACF1
## Training set 0.03225483
```

Neste caso, o modelo ajustado é um ARIMA(0,1,0). A seguir, apresenta-se o gráfico com a previsão e seu intervalo de confiança. A previsão informa que o preço nacional nos EUA continuará nos próximos 10 anos.

```
previsao_usa <- forecast(modelo_usa, h = 10) # previsão para os próximos 10 anos
autoplot(previsao_usa, ylab = "rtfpna", xlab = "Tempo")
```

Forecasts from ARIMA(0,1,0) with drift



Calculando o MASE do modelo. Podemos ver que o MASE para esse modelo é de 0,7997, indicando que modelo pode não estar muito bem ajustado.

```
accuracy(previsao_usa)[1,][6]
```

```
##      MASE
## 0.799736
```

Canadá

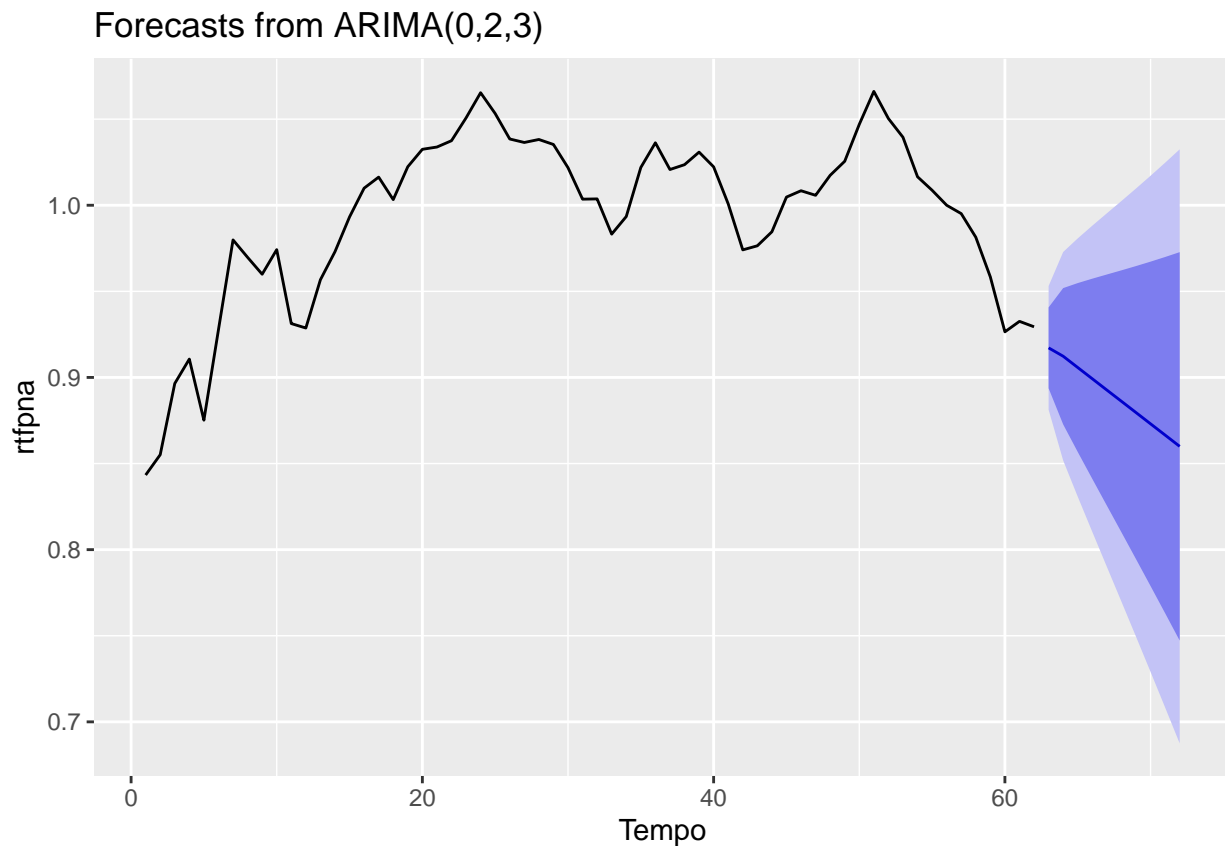
Inicialmente, ajusto o modelo e mostraremos as estimativas para o mesmo

```
modelo_canada <- auto.arima(data_canada$rtfpna)
summary(modelo_canada)
```

```
## Series: data_canada$rtfpna
## ARIMA(0,2,3)
##
## Coefficients:
##          ma1          ma2          ma3
##      -0.6432  -0.4813   0.2222
## s.e.   0.1267   0.1267   0.1163
##
## sigma^2 estimated as 0.0003356:  log likelihood=155.45
## AIC=-302.91  AICc=-302.18  BIC=-294.53
##
## Training set error measures:
##              ME          RMSE          MAE          MPE          MAPE          MASE
## Training set -0.003392639 0.01756456 0.01330863 -0.3506198 1.36017 0.8848314
##              ACF1
## Training set -0.03721486
```

Neste caso, o modelo ajustado é um ARIMA(0,2,3). A previsão informa uma queda no preço nacional do Canadá, entretanto, nota-se que o intervalo de confiança é bastante largo, para as previsões, indicando que o nosso modelo pode não estar bem adequado.

```
previsao_canada <- forecast(modelo_canada, h = 10) # previsão para os próximos 10 anos
autoplot(previsao_canada, ylab = "rtfpna", xlab = "Tempo")
```



Calculando o MASE do modelo. Podemos ver que o MASE para esse modelo é de 0,8848, indicando que modelo pode não estar muito bem ajustado

```
accuracy(previsao_canada)[1,][6]
```

```
##      MASE  
## 0.8848314
```

México

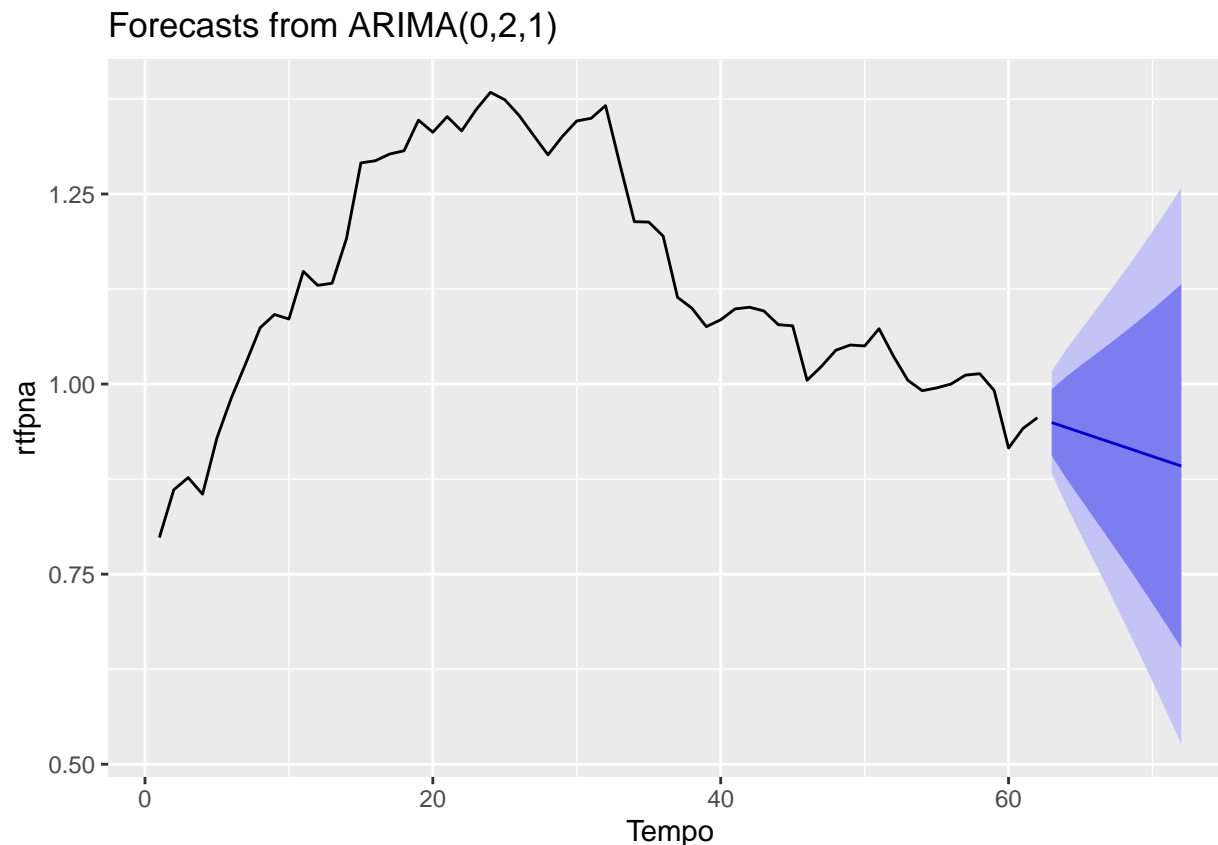
Inicialmente, ajusto o modelo e mostraremos as estimativas para o mesmo

```
modelo_mexico <- auto.arima(data_mexico$rtfpna)  
summary(modelo_mexico)
```

```
## Series: data_mexico$rtfpna  
## ARIMA(0,2,1)  
##  
## Coefficients:  
##          ma1  
##        -0.8502  
## s.e.    0.0617  
##  
## sigma^2 estimated as 0.001167:  log likelihood=117.33  
## AIC=-230.65   AICc=-230.44   BIC=-226.46  
##  
## Training set error measures:  
##              ME      RMSE      MAE      MPE      MAPE      MASE  
## Training set -0.004967013 0.03332467 0.02664148 -0.4162653 2.396274 1.004223  
##              ACF1  
## Training set 0.04940407
```

Neste caso, o modelo ajustado é um ARIMA(0,2,1). A previsão informa uma queda no preço nacional do México, entretanto, nota-se que o intervalo de confiança é bastante largo, para as previsões, indicando que o nosso modelo pode não estar bem adequado.

```
previsao_mexico <- forecast(modelo_mexico, h = 10) # previsão para os próximos 10 anos  
autoplot(previsao_mexico, ylab = "rtfpna", xlab = "Tempo")
```

Calculando o MASE do modelo. Podemos ver que o MASE para esse modelo é de 1,0042, indicando que o modelo pode não estar muito bem ajustado

```
accuracy(previsao_canada)[1,][6]
```

```
##      MASE
## 0.8848314
```

Como alternativa para melhorar o ajuste dos modelos, poderíamos inserir regressores, de tal forma a melhorar a explicação dos dados sobre o preço nacional. Poderíamos utilizar alguma ferramenta de mineração de dados afim de reduzir o conjunto de novas variáveis a serem adicionadas e, em seguida, adicioná-las ao modelo e, desta forma, torná-lo mais informativo em relação ao comportamento da variável *rtfpna*.

Case 2

Lendo o conjunto de dados

Criarei uma variável auxiliar *anos* com o objetivo de codificar a variável *date* em anos.

```
anos = year(as_date(case2_data$date))
```

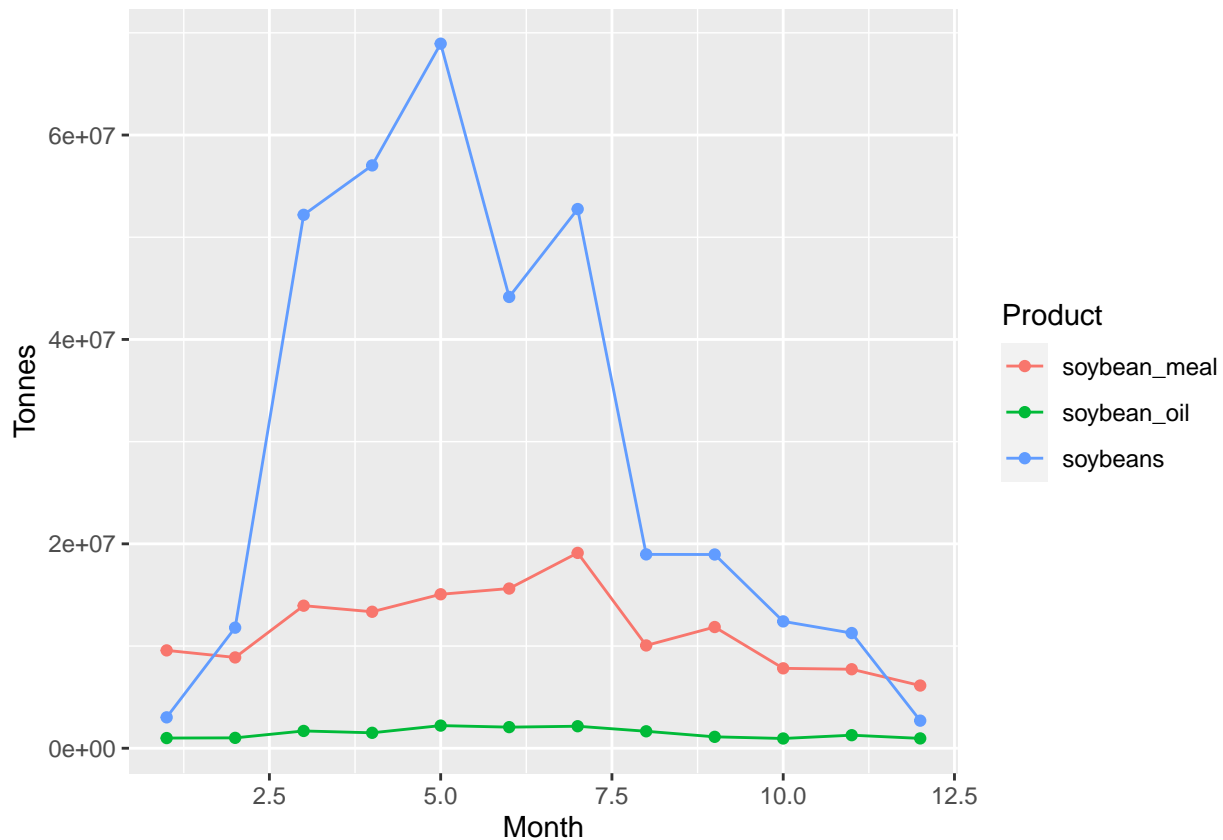
Análises

Primeiramente, analisaremos o comportamento das exportações totais mensais e anuais do Brasil de soja, óleo de soja e farelo de soja. O gráfico a seguir apresenta as toneladas totais mensais e anuais para cada um dos produtos acima.

```

produtos = c("soybean_meal", "soybean_oil", "soybeans")
total_meses = sapply(produtos, function(i){
  aux = subset(case2_data, product == i, select = c("tons","date"))
  aux$date = month(as_date(aux$date))
  tapply(aux$tons, as.factor(aux$date), function(x) sum(x, na.rm = T))
})
aux = data.frame(toneladas = c(total_meses), meses = rep(1:12,3), produto = rep(produtos,each = 12))
p = ggplot(data=aux, aes(x=meses, y=toneladas, colour=produto)) +
  geom_line(aes(color = produto))+
  geom_point(aes(color = produto))
p + labs(colour = "Product", y = "Tonnes", x = "Month")

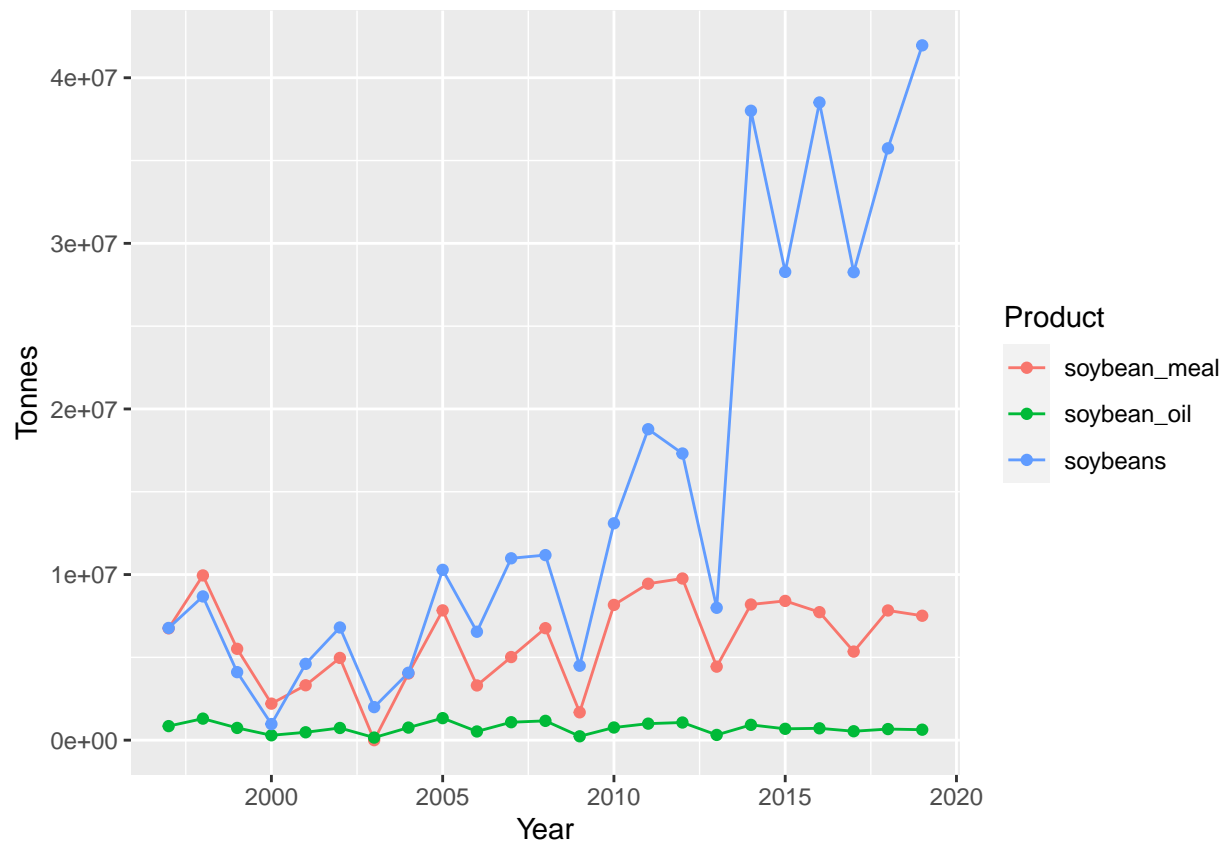
```



```

total_anos = sapply(produtos, function(j) sapply(levels(factor(anos)), function(i){
  aux = c(subset(case2_data, product == j & anos == as.numeric(i) &
    type == "Export", select = tons))$tons
  if(sum(aux) == 0){return(0)} else return(sum(aux, na.rm = T))}))
aux = data.frame(toneladas = c(total_anos), anos = rep(min(anos):max(anos),3), produto = rep(produtos,each = 3))
p = ggplot(data=aux, aes(x=anos, y=toneladas, colour=produto)) +
  geom_line(aes(color = produto))+
  geom_point(aes(color = produto))
p + labs(colour = "Product", y = "Tonnes", x = "Year")

```



Observando os gráficos acima, podemos afirmar ...

Dentre os produtos exportados pelo Brasil nos últimos 5 anos, podemos destacar, como os mais exportados, os seguintes produtos:

```
sort(media_product, decreasing = T)[1:3]
```

```
## soybeans      sugar      corn
## 172742735  58073228  54960041
```

Observando os dados de exportação de milho, temos que os principais tipos de rota de comércio que o Brasil tem adotado são:

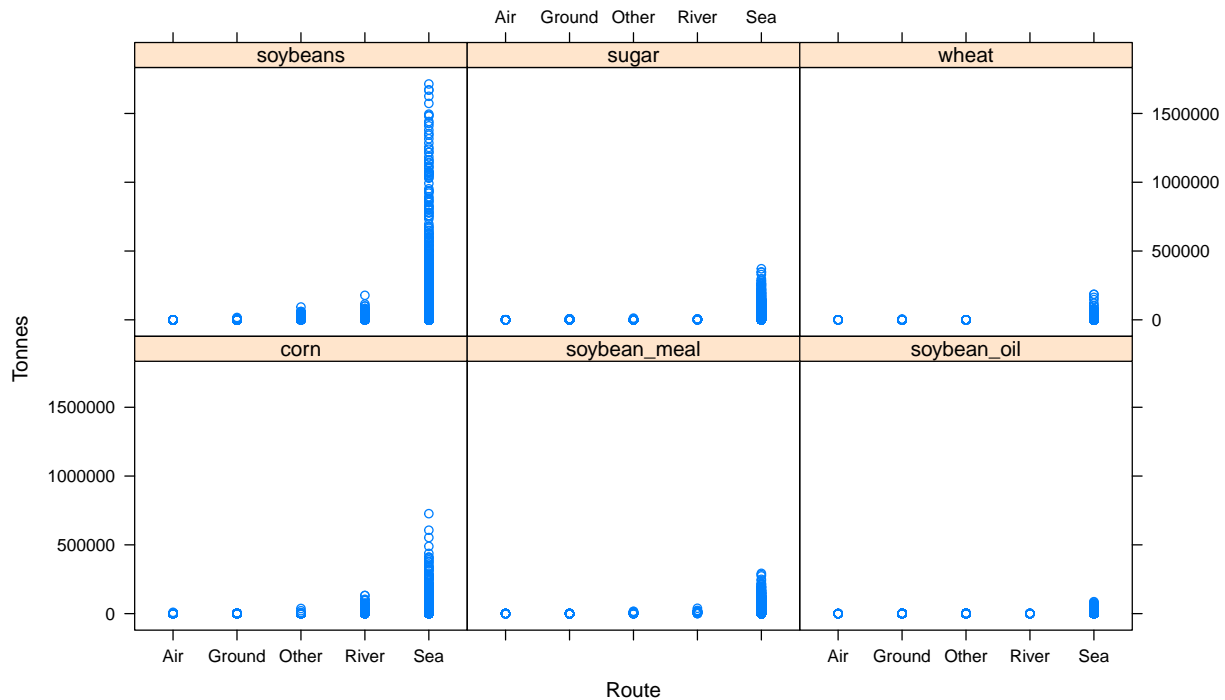
```
data_corn = subset(case2_data, product == "corn" & type == "Export", select = "route")$route
sort(table(data_corn), decreasing = T)
```

```
## data_corn
##   Sea Ground   Air  Other  River
##   5417   803   417   357   198
```

O Gráfico a seguir apresenta as toneladas exportadas de cada um dos produtos por cada uma das rotas mencionadas anteriormente. Observando o gráfico, podemos observar que, para todos os produtos, a prevalência das rotas de transporte se mantêm a mesma. Em particular, podemos destacar que, grande parte da toneladas exportadas dos produtos é relatada por rotas marítimas.

```
data_export = subset(case2_data, type == "Export", select = c("product", "route", "tons"))
produtos = levels(factor(data_export$product))
rotas = levels(factor(data_export$route))
```

```
xyplot(tons ~ factor(route) | factor(product), data = data_export, ylab = "Tonnes", xlab = "Route")
```



Avaliando os países que são parceiros comerciais do Brasil, no que diz respeito ao comércio de milho e açúcar nos últimos 3 anos, os países a seguir representam os países com maiores toneladas comercializadas de milho e açúcar, respectivamente.

```
produtos = c("corn", "sugar")
países = levels(factor(case2_data$country))
total_prod_países = sapply(produtos, function(i) sapply(países, function(j){
  aux = subset(case2_data, product == i & anos > 2016 & country == j, select = c("tons"))
  if(sum(aux$tons) == 0){return(0)} else{return(sum(aux$tons, na.rm = T))}
}))
sort(total_prod_países[,1], decreasing = T)[1:3]
```

```
## Iran Vietnam Egypt
## 7435073 3253610 2821014
```

```
sort(total_prod_países[,2], decreasing = T)[1:3]
```

```
## Algeria Bangladesh China
## 3949709 3875487 2006682
```

Agora, avaliaremos a quantidade de toneladas exportadas de cada produção em relação aos estados objetivando destacar os cinco estados líderes em exportação de cada um dos produtos. Desta forma, a Tabela a seguir apresenta os 5 estados com maiores toneladas de exportação para cada um dos produtos.

```
data_export = subset(case2_data, type == "Export", select = c("tons", "state", "product"))
produtos = levels(factor(case2_data$product))
estados_produtos = sapply(produtos, function(i){
  aux = subset(data_export, product == i, select = c("tons", "state"))
  return(names(sort(tapply(aux$tons, aux$state, mean), decreasing = T)[1:5]))
})
```

```

})
estados_produtos

##      corn soybean_meal soybean_oil soybeans sugar wheat
## [1,] "MT" "BA"          "PI"        "RS"      "AL"  "RS"
## [2,] "PR" "PR"          "RS"        "MT"      "PR"  "PR"
## [3,] "BA" "MT"          "PR"        "PR"      "SP"  "SC"
## [4,] "MA" "RS"          "MT"        "BA"      "MG"  "MS"
## [5,] "MS" "ES"          "MG"        "MA"      "PE"  "SP"

```

Modelagem

Para modelar os dados, optei por ajustar um modelo para cada um dos produtos em estudo e, em seguida, realizar a previsão da quantidade de exportações nos próximos 11 anos para cada um dos produtos. Os dados a serem utilizados são as quantidades totais de toneladas exportadas de cada produto por ano. Primeiramente, realizarei o ajuste do modelo sem as covariadas.

```

produtos = c("soybeans", "corn", "soybean_meal")
países = c("China", "Iran", "Netherlands", "Spain", "Thailand", "Egypt", "Japan", "Vietnam")
covariáveis = read.table("covariates.csv", head = T, sep = ";", dec = ",")

```

Note que, para todos os produtos, a medida de acurácia da previsão, MASE, são bastante elevadas, desta forma, temos indícios que o nosso modelo não está bem ajustado e, conseqüentemente, não nos fornecerá boas previsões. Como uma forma de aumentar o poder preditivo do modelo, inseri as covariáveis descritas em covariates (.xlsx) no modelo.

Milho Criando a série de dados e o ajustando o modelo ARIMA para os dados de exportação anual de milho e mostraremos as estimativas para o mesmo. Neste caso, o modelo ajustado é um ARIMA(0,1,1) e sua MASE 0,7477, indicando um modelo com pouco poder preditivo.

```

dados_export_corn = subset(case2_data, type = "Export", product == "corn", select = c("date", "tons"))
corn_year = ts(sapply(min(anos):max(anos), function(i){
  aux = subset(dados_export_corn, year(as_date(dados_export_corn$date)) == i, select = "tons")
  if(sum(aux$tons) == 0){return(NA)} else{return(sum(aux$tons, na.rm = T))}}
), start = 1997)
modelo_corn = auto.arima(corn_year, stepwise = T)
summary(modelo_corn)

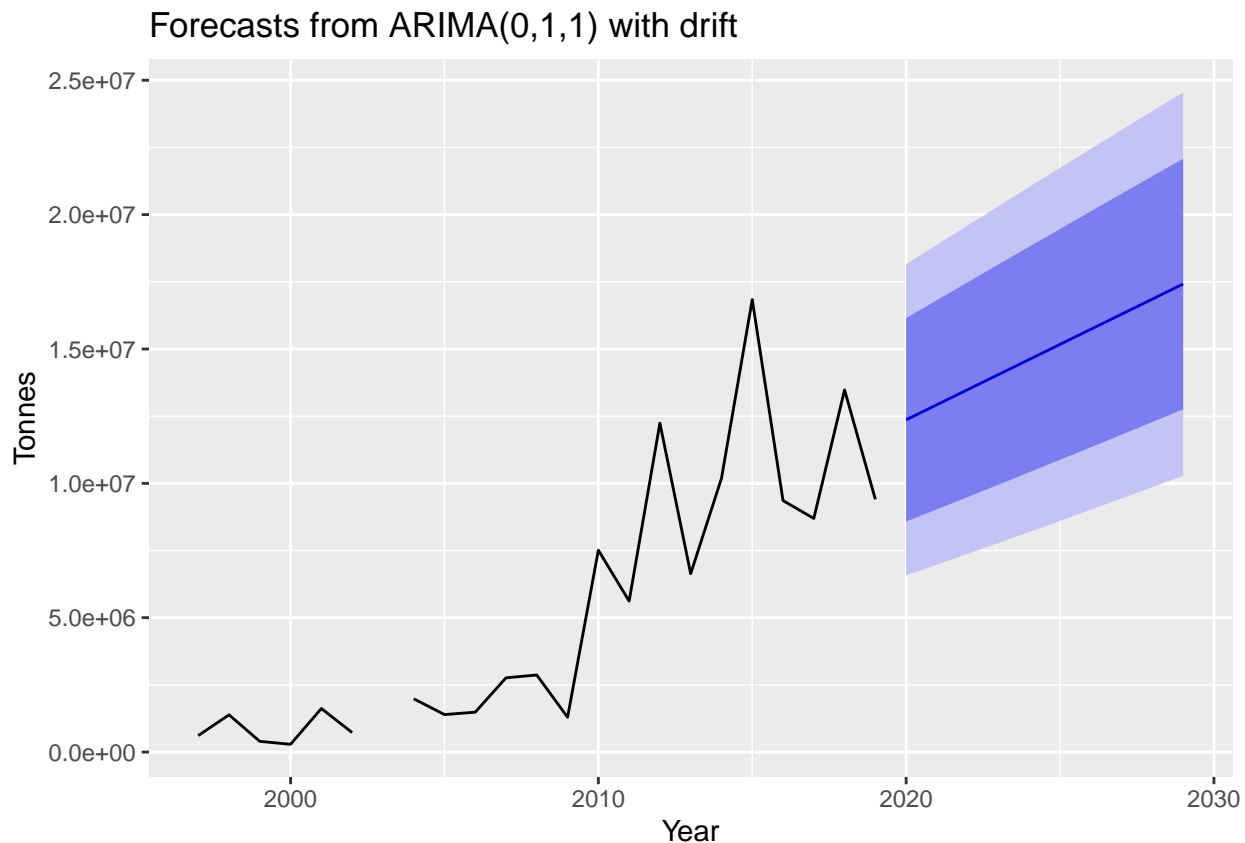
```

```

## Series: corn_year
## ARIMA(0,1,1) with drift
##
## Coefficients:
##          ma1      drift
##        -0.7598  561578.2
## s.e.    0.2882  193492.6
##
## sigma^2 estimated as 8.718e+12: log likelihood=-342.1
## AIC=690.2   AICc=691.54   BIC=693.48
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -122338.7 2743984 2066412 -73.67052 90.94422 0.7477483 -0.09429198

previsao_corn <- forecast(modelo_corn)
autoplot(previsao_corn, ylab = "Tonnes", xlab = "Year")

```

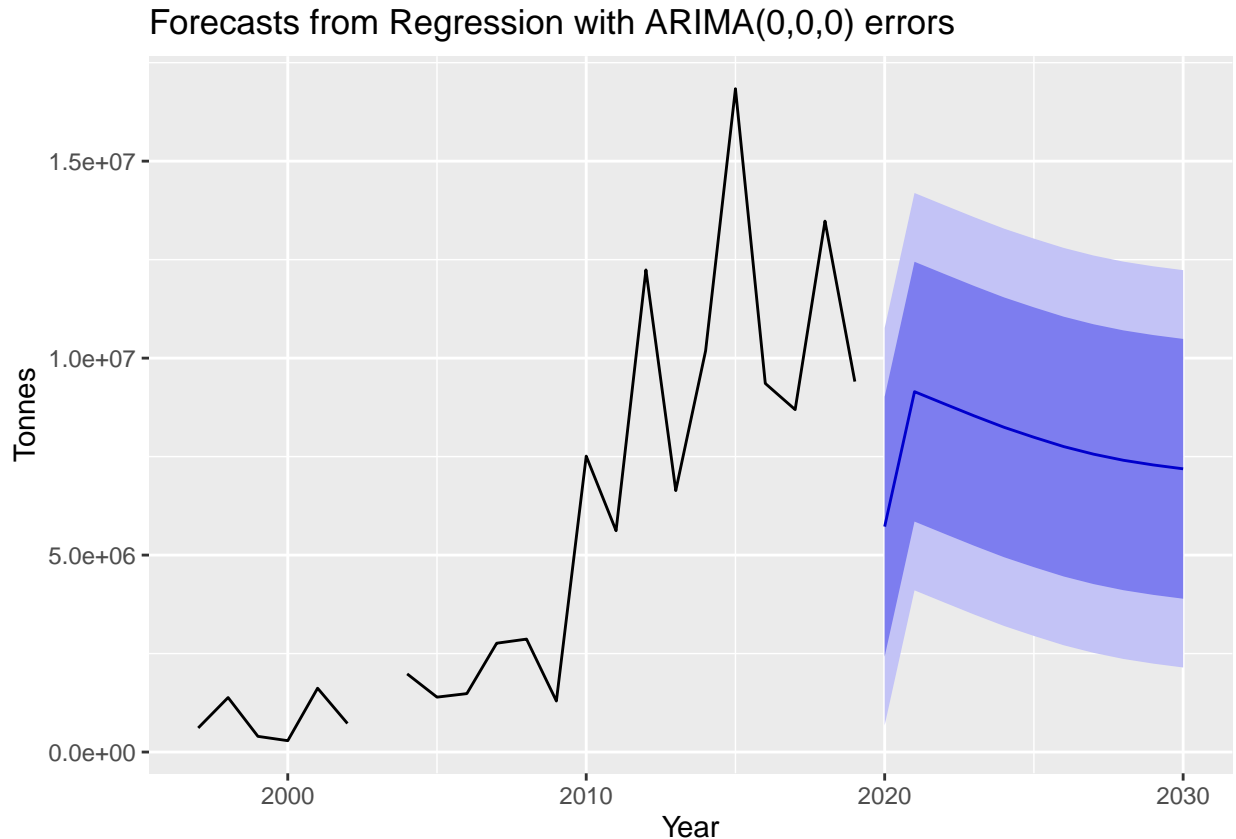


A seguir, apresento o modelo, suas estimativas e medidas, assim como um Gráfico apresentando as estimativas para os próximos 11 anos do total de toneladas de milho exportada. Por fim, ficamos com um modelo ARIMA(0,0,0), o que nos indicou que o modelo apenas com as covariadas é o melhor modelo, no que diz respeito as métricas de AIC, AICC, BIC e MASE.

```
X_reg = as.matrix(subset(covariaveis, year >= min(anos) & year <= max(anos))[, -c(1,2,4)])
X_previsao = as.matrix(subset(covariaveis, year > max(anos))[, -c(1,2,4)])
modelo_corn = auto.arima(corn_year, xreg = X_reg, stepwise = T)
summary(modelo_corn)
```

```
## Series: corn_year
## Regression with ARIMA(0,0,0) errors
##
## Coefficients:
##      price_corn  gdp_china  gdp_iran  gdp_netherlands  gdp_spain
##      -11200.85  21938.84  -136471.17      -61714.33    77118.1
## s.e.    18738.13  33230.02   78512.05    331685.30  249655.7
##      gdp_thailand  gdp_world  gdp_egypt  gdp_japan  gdp_vietnam
##      27145.55   241863.2   -43635.96  -49117.01  -103732.10
## s.e.    75070.41   313486.4    89177.43   191130.71    70571.01
##
## sigma^2 estimated as 6.62e+12:  log likelihood=-349.28
## AIC=720.56  AICC=744.56  BIC=733.05
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 1998.121 1900296 1525891 -25.98959 63.34356 0.5521564 -0.3061493
```

```
previsao_corn <- forecast(modelo_corn, xreg = X_previsao)
autoplot(previsao_corn, ylab = "Tonnes", xlab = "Year")
```



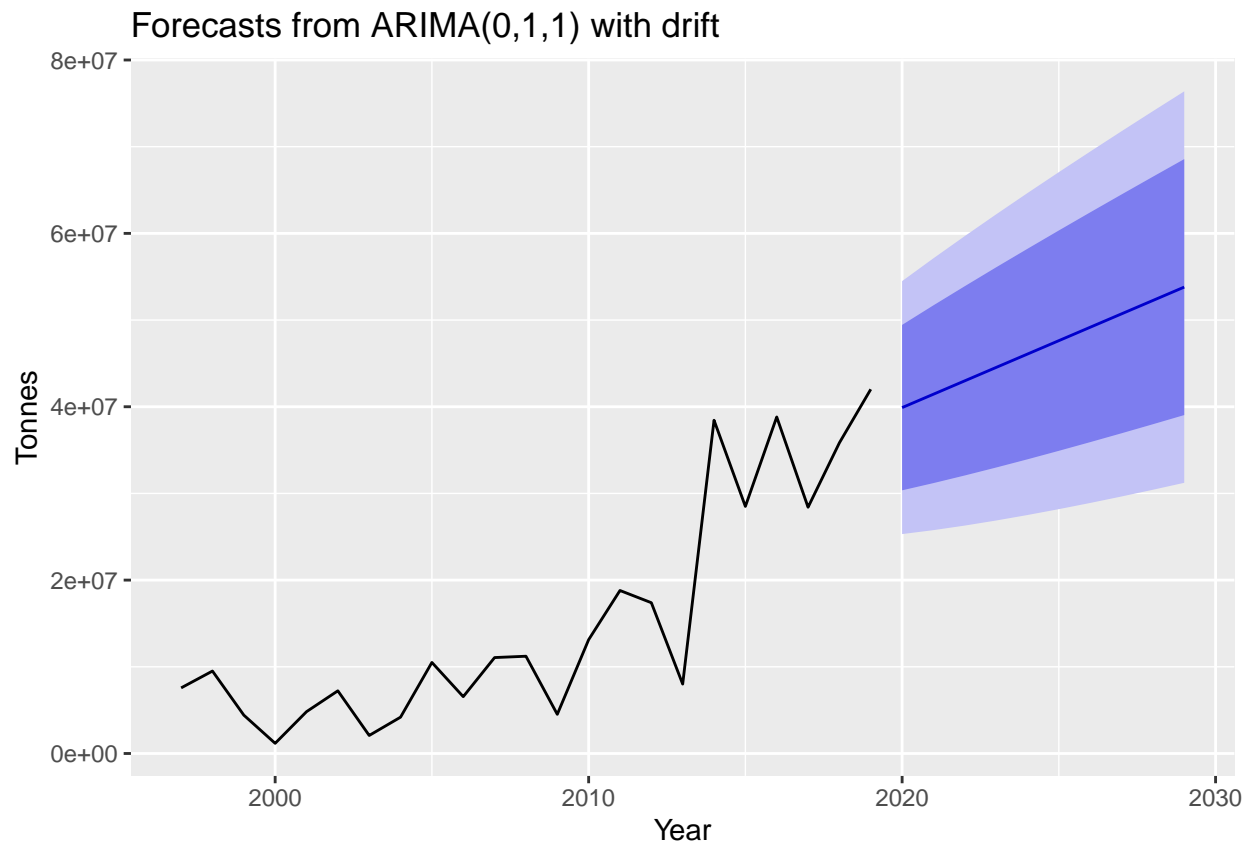
Soja Criando a série de dados e o ajustando o modelo ARIMA para os dados de exportação anual de soja e mostraremos as estimativas para o mesmo. Neste caso, o modelo ajustado é um ARIMA(0,1,1) e sua MASE 0,7433, indicando um modelo com pouco poder preditivo.

```
dados_export_soybean = subset(case2_data, type = "Export", product == "soybeans", select = c("date", "tons"))
soybean_year = ts(sapply(min(anos):max(anos), function(i){
  aux = subset(dados_export_soybean, year(as_date(dados_export_soybean$date)) == i, select = "tons")
  if(sum(aux$tons) == 0){return(NA)} else{return(sum(aux$tons, na.rm = T))}
}), start = 1997)
modelo_soybean = auto.arima(soybean_year, stepwise = T)
summary(modelo_soybean)
```

```
## Series: soybean_year
## ARIMA(0,1,1) with drift
##
## Coefficients:
##          ma1      drift
##        -0.6064  1544786
## s.e.    0.1564   643236
##
## sigma^2 estimated as 5.542e+13:  log likelihood=-378.5
## AIC=763   AICc=764.34   BIC=766.28
##
```

```
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -282405.6 6941812 4899875 -68.90002 84.9974 0.7433441 -0.136186
```

```
previsao_soybean <- forecast(modelo_soybean)
autoplot(previsao_soybean, ylab = "Tonnes", xlab = "Year")
```



A seguir, apresento o modelo, suas estimativas e medidas, assim como um Gráfico apresentando as estimativas para os próximos 11 anos do total de toneladas de soja exportada. Por fim, ficamos com um modelo ARIMA(0,0,0), o que nos indicou que o modelo apenas com as covariadas é o melhor modelo, no que diz respeito as métricas de AIC, AICC, BIC e MASE.

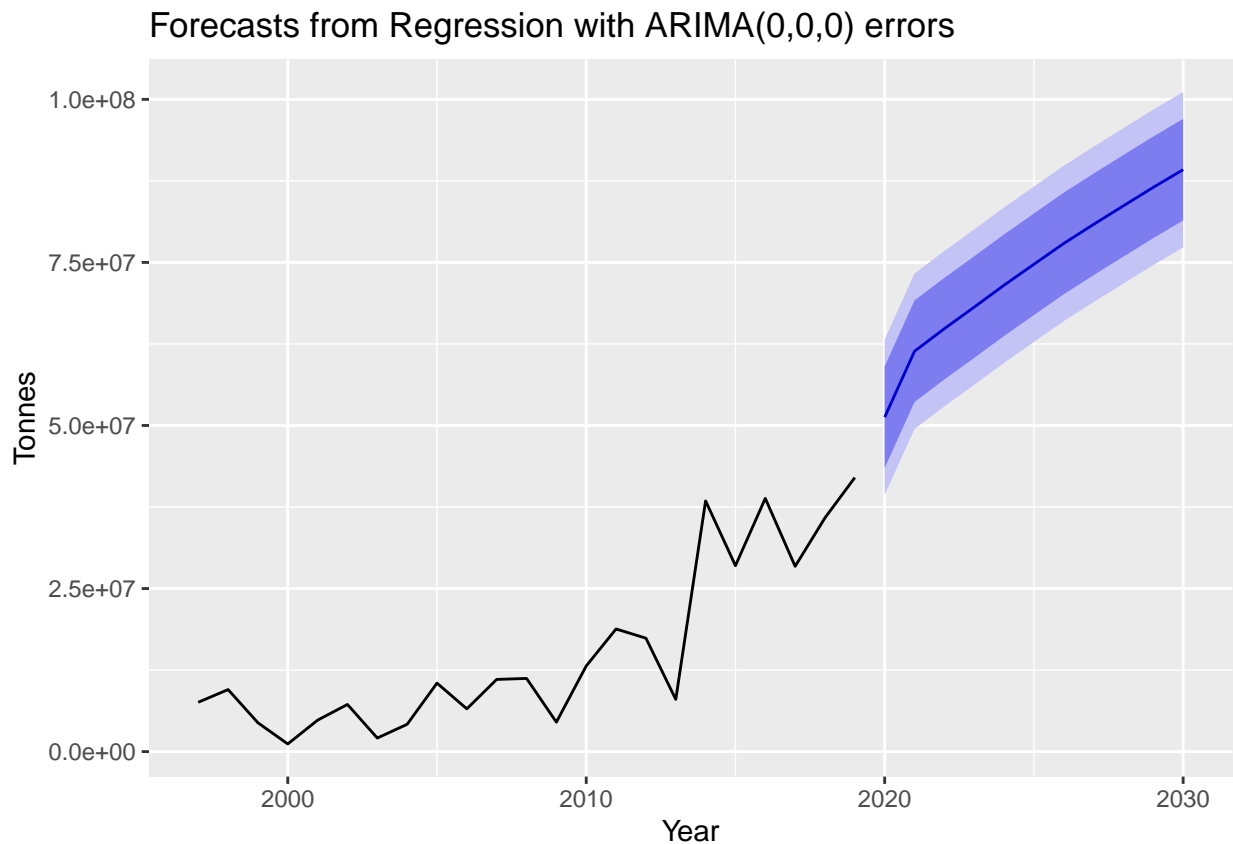
```
X_reg = as.matrix(subset(covariaveis, year >= min(anos) & year <= max(anos))[, -c(1,3,4)])
X_previsao = as.matrix(subset(covariaveis, year > max(anos))[, -c(1,3,4)])
modelo_soybean = auto.arima(soybean_year, xreg = X_reg, stepwise = T)
summary(modelo_soybean)
```

```
## Series: soybean_year
## Regression with ARIMA(0,0,0) errors
##
## Coefficients:
##      price_soybeans  gdp_china  gdp_iran  gdp_netherlands  gdp_spain
##           44157.84   30835.29  -40915.25        -904203.5    513894.1
## s.e.           65305.64   79191.54  185826.86         762449.6    587514.0
##      gdp_thailand  gdp_world  gdp_egypt  gdp_japan  gdp_vietnam
##        -373395.8   524657.3  -171591.5   597541.7    41234.88
## s.e.       183492.1   755138.8   219230.1   436843.9   169914.56
##
```



```
## sigma^2 estimated as 3.693e+13: log likelihood=-385.34
## AIC=792.67 AICc=816.67 BIC=805.16
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 2746.246 4568801 3349806 -22.45439 44.303 0.5081883 -0.5946722

previsao_soybean <- forecast(modelo_soybean, xreg = X_previsao)
autoplot(previsao_soybean, ylab = "Tonnes", xlab = "Year")
```



farelo de soja Criando a série de dados e o ajustando o modelo ARIMA para os dados de exportação anual de farelo de soja e mostraremos as estimativas para o mesmo. Neste caso, o modelo ajustado é um ARIMA(0,1,1) e sua MASE 0,7704, indicando um modelo com pouco poder preditivo.

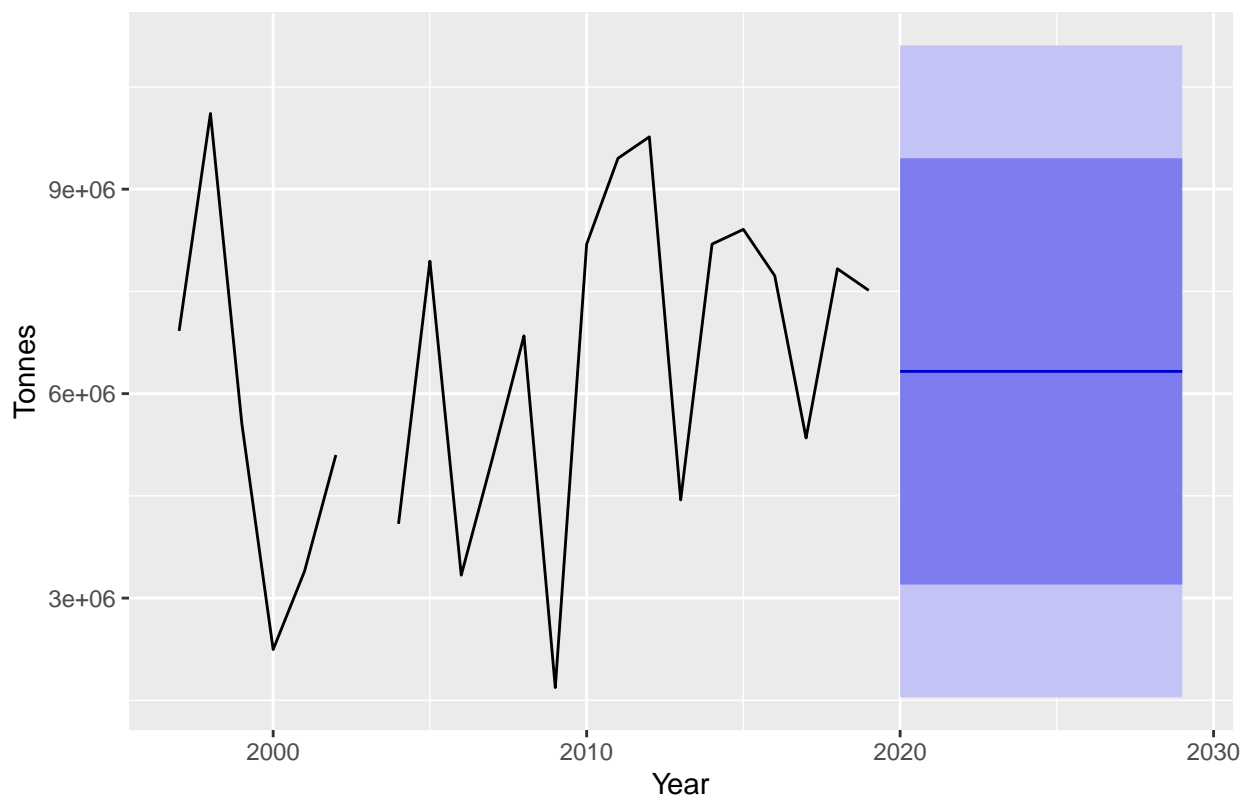
```
dados_export_soybean_meal = subset(case2_data, type = "Export", product == "soybean_meal", select = c("date", "tons"))
soybean_meal_year = ts(sapply(min(anos):max(anos), function(i){
  aux = subset(dados_export_soybean_meal, year(as_date(dados_export_soybean_meal$date)) == i, select = c("tons"))
  if(sum(aux$tons) == 0){return(NA)} else{return(sum(aux$tons, na.rm = T))}
}), start = 1997)
modelo_soybean_meal = auto.arima(soybean_meal_year, stepwise = T)
summary(modelo_soybean_meal)
```

```
## Series: soybean_meal_year
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##              mean
```

```
##          6326121.3
## s.e.    508449.5
##
## sigma^2 estimated as 5.956e+12: log likelihood=-354.27
## AIC=712.55   AICc=713.15   BIC=714.82
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 2.116756e-09 2384325 2091058 -24.78468 48.95445 0.7704626
##              ACF1
## Training set 0.1171636
```

```
previsao_soybean_meal <- forecast(modelo_soybean_meal)
autoplot(previsao_soybean_meal, ylab = "Tonnes", xlab = "Year")
```

Forecasts from ARIMA(0,0,0) with non-zero mean



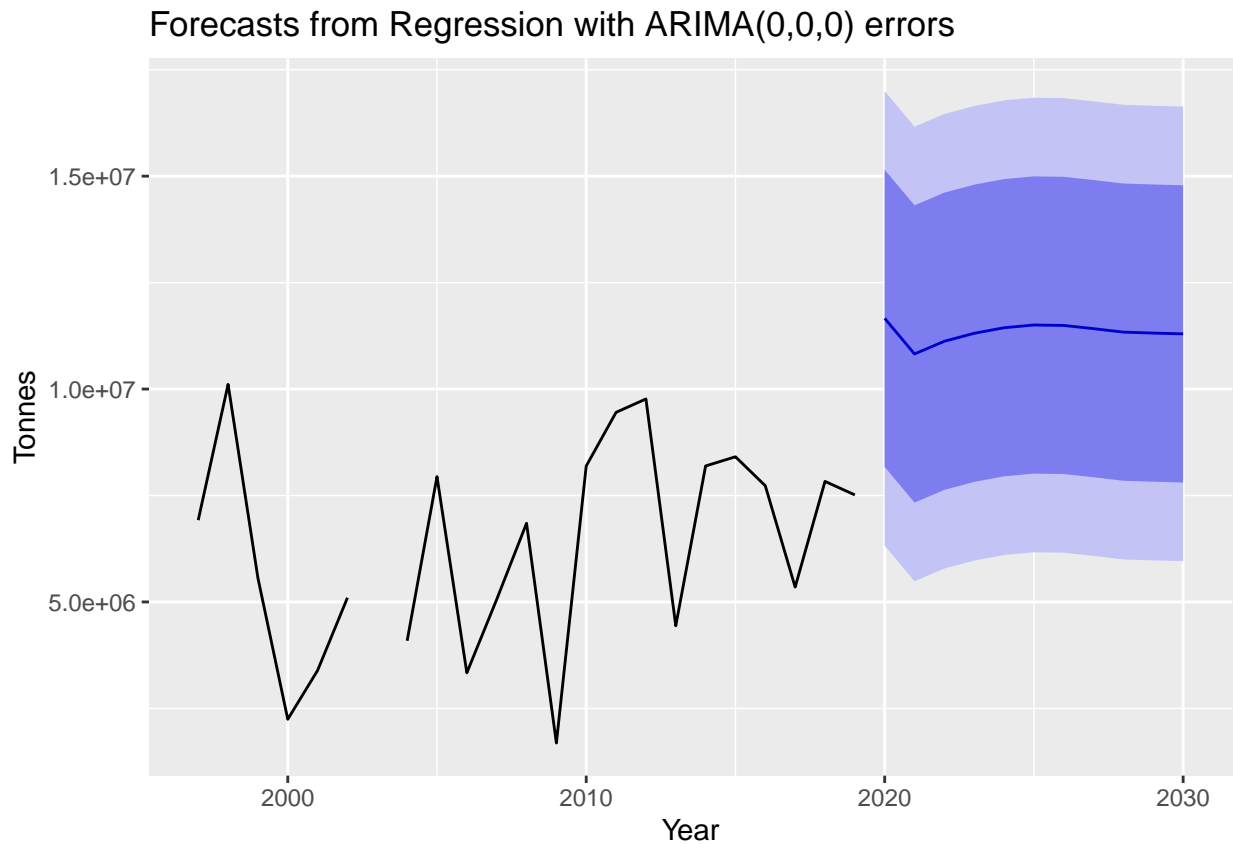
A seguir, apresento o modelo, suas estimativas e medidas, assim como um Gráfico apresentando as estimativas para os próximos 11 anos do total de toneladas de farelo de soja exportada. Por fim, ficamos com um modelo ARIMA(0,0,0), o que nos indicou que o modelo apenas com as covariadas é o melhor modelo, no que diz respeito as métricas de AIC, AICC, BIC e MASE.

```
X_reg = as.matrix(subset(covariaveis, year >= min(anos) & year <= max(anos))[, -c(1,2,3)])
X_previsao = as.matrix(subset(covariaveis, year > max(anos))[, -c(1,2,3)])
modelo_soybean_meal = auto.arima(soybean_meal_year, xreg = X_reg, stepwise = T)
summary(modelo_soybean_meal)
```

```
## Series: soybean_meal_year
## Regression with ARIMA(0,0,0) errors
##
```

```
## Coefficients:
##      price_soybean_meal  gdp_china  gdp_iran  gdp_netherlands  gdp_spain
##                5520.59   -4541.26  24272.59        -369438.4   31741.04
## s.e.                32856.75   35158.82  83036.31        344813.6  271956.59
##      gdp_thailand  gdp_world  gdp_egypt  gdp_japan  gdp_vietnam
##      -55856.46   107196.2   77095.16   236017.5   -2163.84
## s.e.        89369.19   332460.6   96009.50   195905.9   75770.27
##
## sigma^2 estimated as 7.416e+12:  log likelihood=-350.53
## AIC=723.06   AICc=747.06   BIC=735.55
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -42.7891 2011264 1622624 -18.25201 37.87853 0.5978655 -0.2340691

previsao_soybean_meal <- forecast(modelo_soybean_meal, xreg = X_previsao)
autoplot(previsao_soybean_meal, ylab = "Tonnes", xlab = "Year")
```



Pode-se notar que, com a inclusão das covariáveis no modelo, podemos obter melhores previsões para o total de toneladas exportadas dos produtos.