

## Case 2

Matheus Barreto Alves de Almeida

07/07/2020

Primeiramente, vamos iniciar a bibliotecas importantes

```
library(ggplot2)
library(dplyr)
library(readxl)
library(plotly)
library(purrr)
library(cluster)
library(forecast)
library(fpp2)
library(lattice)
```

Para responder cada questão, será criado dataframes que serão nomeados por “dfi” onde i é o numero da questão.

Lendo os dataframes

```
comex <- read.csv('data_comexstat.csv')
covar <- read_xlsx('covariates.xlsx')
```

```
head(comex)
```

##	date	product	state	country	type	route	tons	usd
## 1	1997-01-01	corn	ES	United States	Import	Sea	44.045	113029
## 2	1997-01-01	corn	GO	Argentina	Import	Ground	54.000	36720
## 3	1997-01-01	corn	GO	Bolivia	Export	Ground	0.200	180
## 4	1997-01-01	corn	GO	United States	Export	Sea	3.488	5688
## 5	1997-01-01	corn	MG	Argentina	Import	Ground	27.000	18630
## 6	1997-01-01	corn	MS	Paraguay	Export	Ground	40.000	38700

É visível que existem variáveis que precisam ser categorizadas:

```
comex$state <- factor(comex$state)
comex$product <- factor(comex$product)
comex$country <- factor(comex$country)
comex$route <- factor(comex$route)
comex$type <- factor(comex$type)
```

Além disso, precisamos transformar a coluna “date” para o formato Date:

```
comex$date <- as.Date(comex$date)
```

Para responder a primeira questão, vou fazer o uso de duas novas colunas : month e year, que irá usar a coluna “date” para extrair o mês e o ano.

```
comex <- comex %>% mutate(month = format(date,"%m"), year = format(date,"%Y")) #Esse código pega os ele
```

Nosso primeiro dataframe irá mostrar a evolução anual da exportação de todo o período para os três produtos pedidos ( total em toneladas ):

```
df <- comex %>% filter(type== "Export",) %>% group_by(year,product) %>%  
  filter(product %in% c('soybeans', 'soybean_oil', 'soybean_meal')) %>% summarize(total = sum(tons))
```

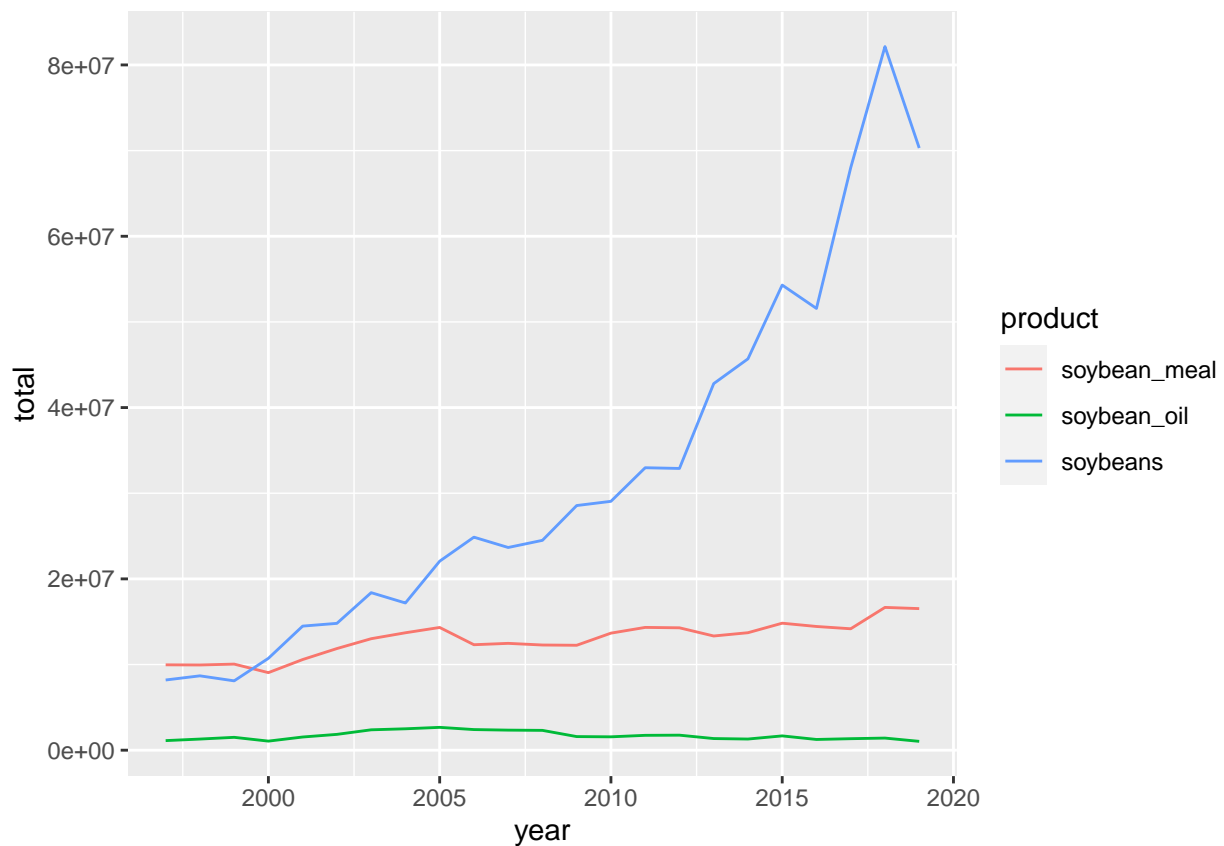
```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

```
df$year <- as.numeric(df$year)  
head(df)
```

```
## # A tibble: 6 x 3  
## # Groups:   year [2]  
##   year product      total  
##   <dbl> <fct>      <dbl>  
## 1 1997 soybean_meal 9965945.  
## 2 1997 soybean_oil 1117451.  
## 3 1997 soybeans   8195025.  
## 4 1998 soybean_meal 9947875.  
## 5 1998 soybean_oil 1300080.  
## 6 1998 soybeans   8679492.
```

Vamos visualizar melhor usando ggplot:

```
df %>% group_by(product) %>% ggplot(aes(year,total,color=product))+geom_line()
```



Para responder a segunda pergunta, precisamos criar um dataframe que agora, irá pegar apenas as observações a partir de 2014

```
df2 <- comex %>% filter(type == "Export") %>%
  filter(date >= "2014-01-01") %>% group_by(product) %>% summarize(total = sum(usd))
```

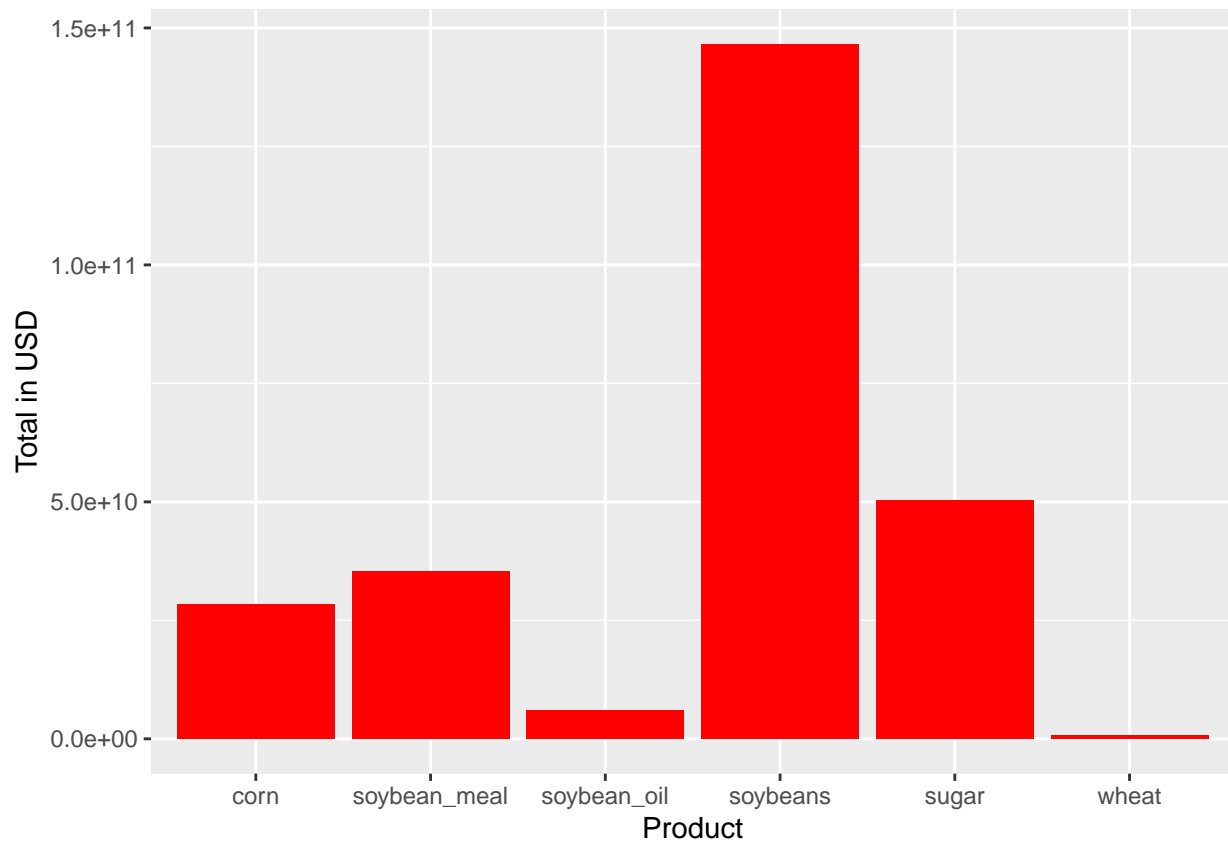
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(df2)
```

```
## # A tibble: 6 x 2
##   product      total
##   <fct>      <dbl>
## 1 corn      28373289898
## 2 soybean_meal 35408303281
## 3 soybean_oil  5924025117
## 4 soybeans   146600566283
## 5 sugar      50326838979
## 6 wheat       827671155
```

Agora, vamos criar gráfico para visualizar melhor o que está acontecendo:

```
ggplot(df2,aes(x=factor(product),y= total))+ geom_col(fill = "red")+xlab("Product") + ylab("Total in USD")
```



A partir deste gráfico, podemos ver que os produtos indicados pelo código 'soybeans', 'soybean\_meal' e 'sugar' são os produtos mais importantes em termos de valor em USD.

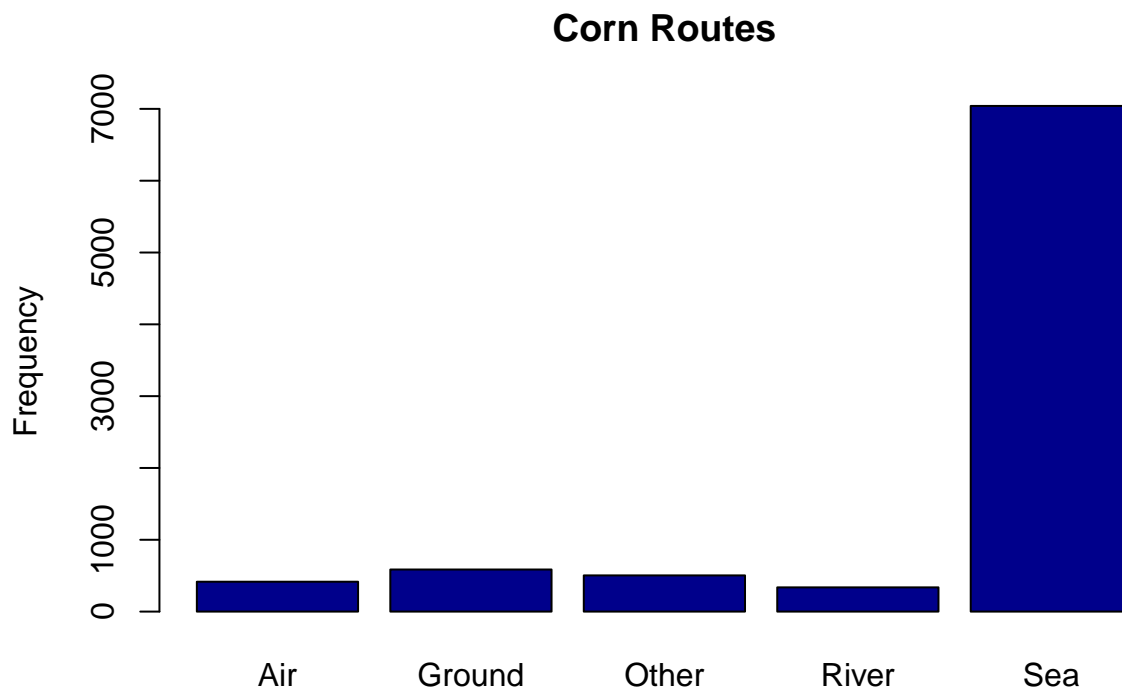
Na terceira pergunta, utilizou-se o mesmo período, filtrando-se apenas o produto pedido:

```
df3 <- comex %>% filter(type == "Export") %>% filter(date >= "2014-01-01" & product == "corn")
head(df3)
```

```
##      date product state  country  type route      tons      usd month year
## 1 2014-01-01   corn   BA  Malaysia Export   Sea    6.940    1388    01 2014
## 2 2014-01-01   corn   G0   Algeria Export   Sea 5361.021 1100618    01 2014
## 3 2014-01-01   corn   G0    Ecuador Export   Sea 2150.465 441921    01 2014
## 4 2014-01-01   corn   G0  Indonesia Export   Sea  220.000   40982    01 2014
## 5 2014-01-01   corn   G0  Indonesia Export   Sea  970.721 192407    01 2014
## 6 2014-01-01   corn   G0      Iran Export   Sea  147.116   29702    01 2014
```

Para visualizar as rotas mais usadas, vamos utilizar um gráfico de barras:

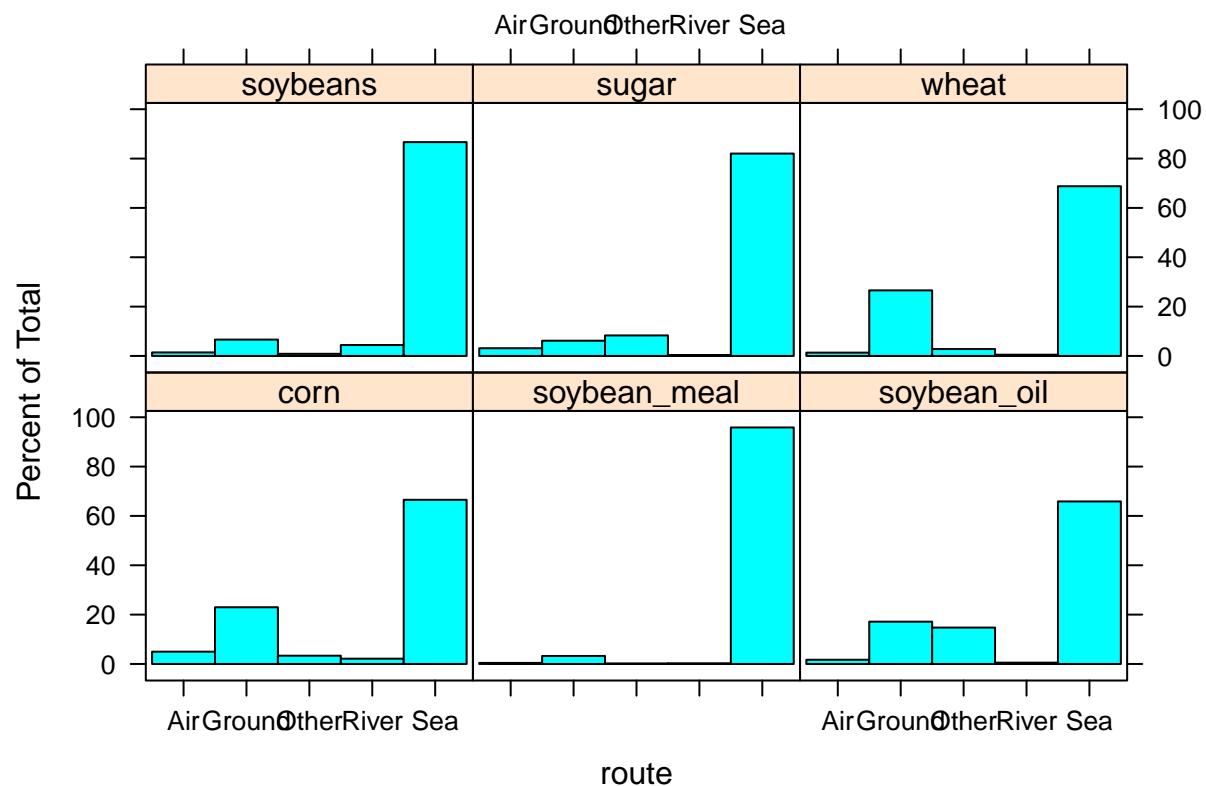
```
-barplot(table(df3$route), main = "Corn Routes", ylab = "Frequency", col = 'darkblue')
```



```
##      [,1]
## [1,] -0.7
## [2,] -1.9
## [3,] -3.1
## [4,] -4.3
## [5,] -5.5
```

Fica evidente que a rota mais usada é a marítima. Isso também se repete para os outros produtos:

```
with(comex, histogram(~route | product))
```



Agora, vamos verificar os parceiros mais importantes do Brasil nos ultimos 3 anos, respondendo a questão 4. Aqui, considera-se os parceiros mais importantes aqueles com o qual o Brasil importou e exportou mais.

```
df4 <- comex %>%filter(date >= "2017-01-01" & product %in% c("corn", 'sugar'))
head(df4)
```

```
##      date product state      country  type route   tons   usd
## 1 2017-01-01   corn    AL    Argentina Import   Sea 9759.000 1805225
## 2 2017-01-01   corn    CE    Argentina Import   Sea 29933.422 5586005
## 3 2017-01-01   corn    GO    Argentina Import Ground    5.000   20000
## 4 2017-01-01   corn    GO Dominican Republic Export   Sea  795.679 147710
## 5 2017-01-01   corn    GO      Ecuador Export   Air    0.122   1504
## 6 2017-01-01   corn    GO      Egypt Export   Sea  199.950 533884
##  month year
## 1     01 2017
## 2     01 2017
## 3     01 2017
## 4     01 2017
## 5     01 2017
## 6     01 2017
```

É útil utilizar a função `top_n` do `dplyr`, que verifica os `n` primeiros elementos com a maior contagem. Para o nosso caso, o número de contagens é o número de trades efetuados.

```
top6parceiros <- df4 %>% count(country, sort = TRUE) %>% top_n(6)
```

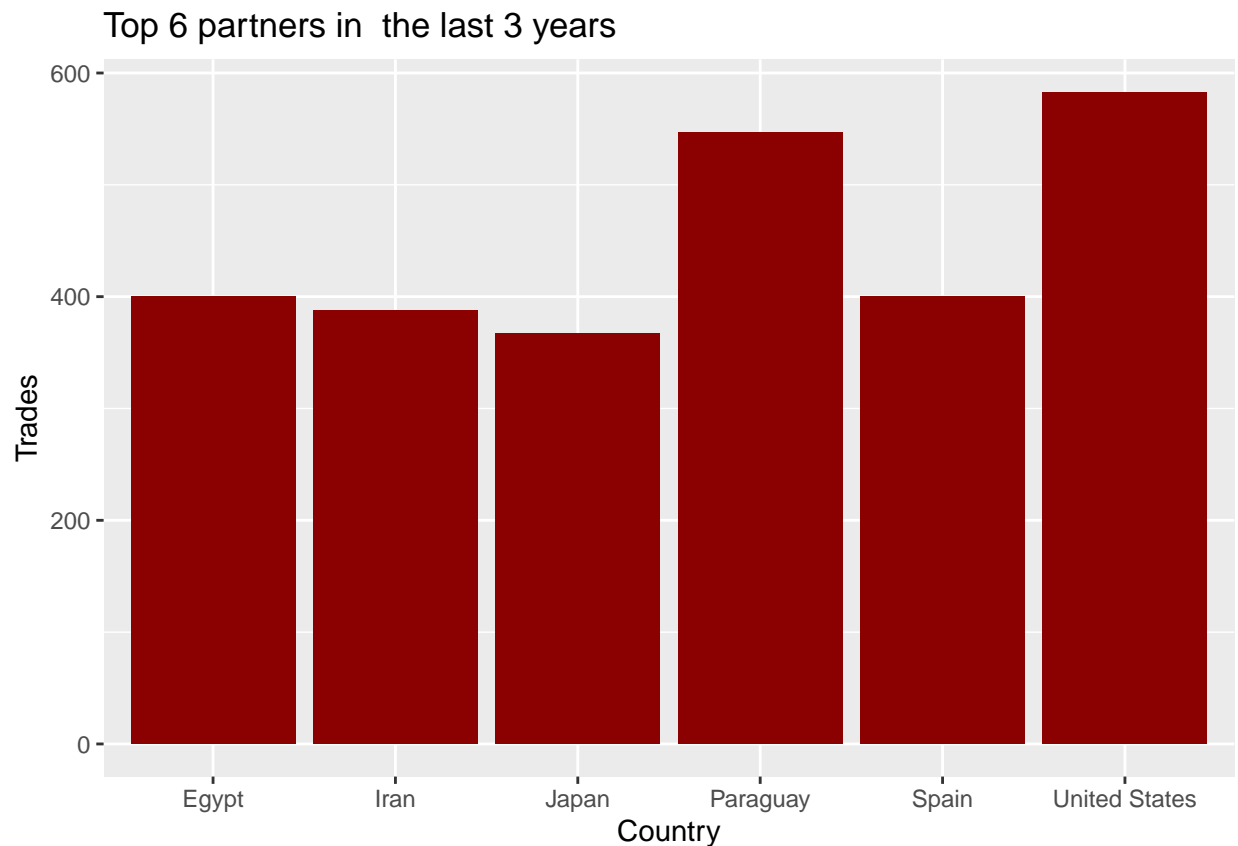
```
## Selecting by n
```

```
head(top6parceiros)
```

```
##      country  n
## 1 United States 583
## 2   Paraguay 547
## 3    Egypt 400
## 4    Spain 400
## 5    Iran 388
## 6    Japan 367
```

Vamos visualizar esses parceiros lado a lado:

```
ggplot(top6parceiros,aes(country,n))+ geom_col(fill="darkred") + xlab("Country")+ ylab("Trades") +labs(
```



Com a mesma função, podemos verificar os top 5 estados mais importantes em termos de exportação, levando em conta cada produto.

```
df5 <- comex %>% group_by(product) %>% count(state) %>% top_n(5)
```

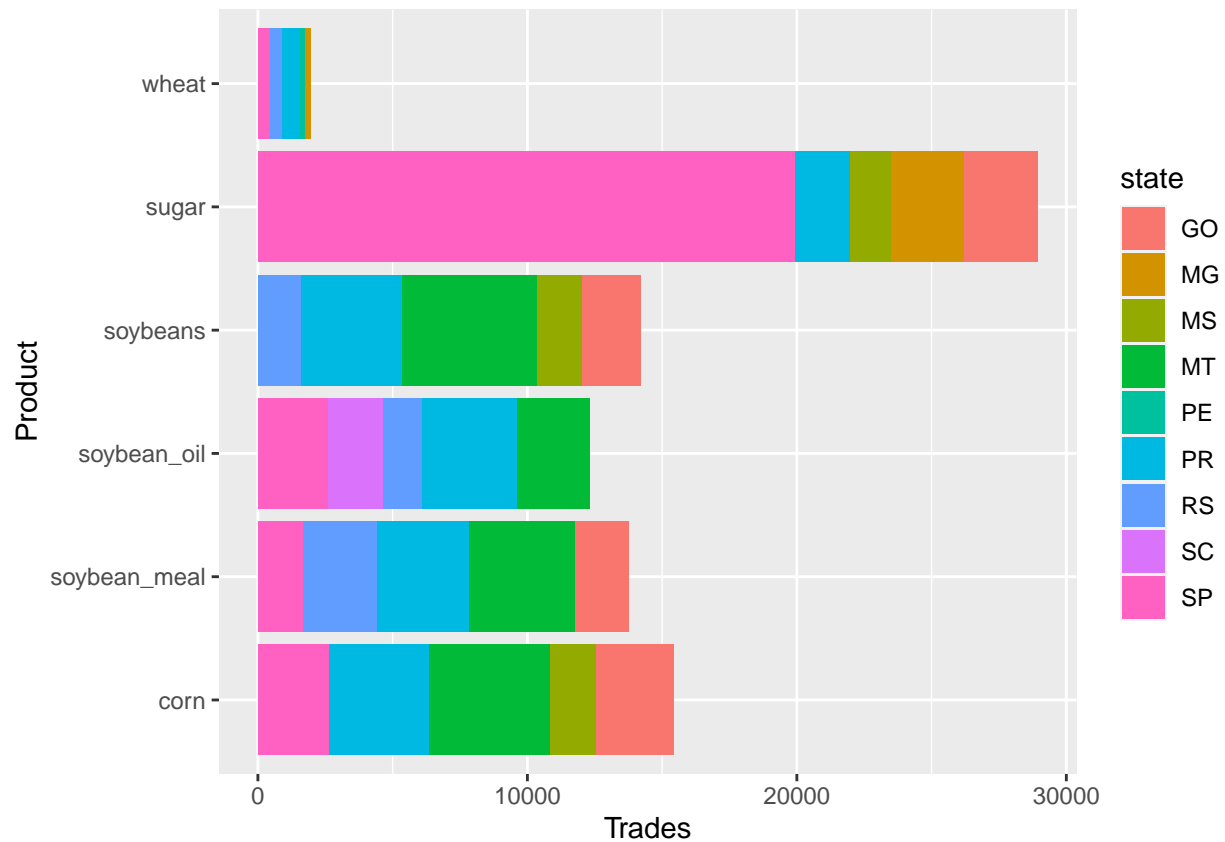
```
## Selecting by n
```

```
head(df5)
```

```
## # A tibble: 6 x 3
## # Groups:   product [2]
##   product      state      n
##   <fct>      <fct> <int>
## 1 corn       GO      2883
## 2 corn       MS      1711
```

```
## 3 corn      MT      4476
## 4 corn      PR      3744
## 5 corn      SP      2607
## 6 soybean_meal GO      1976
```

```
ggplot(df5,aes(n,product,fill = state))+ geom_col() + ylab('Product') + xlab("Trades")
```



Aqui, cada estado e representado por uma cor. O preenchimento da cor indica a contribuicao daquele estado.

Podemos agora tentar prever como sera a quantidade de toneladas da exportacao de Soybean, Corn e Soybean Meal usando o mesmo processo do Case 1.

```
df6 <- comex %>% filter(type== "Export",) %>% group_by(year,product) %>%
  filter(product %in% c('soybeans','corn','soybean_meal')) %>% summarize(total = sum(tons))
```

```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

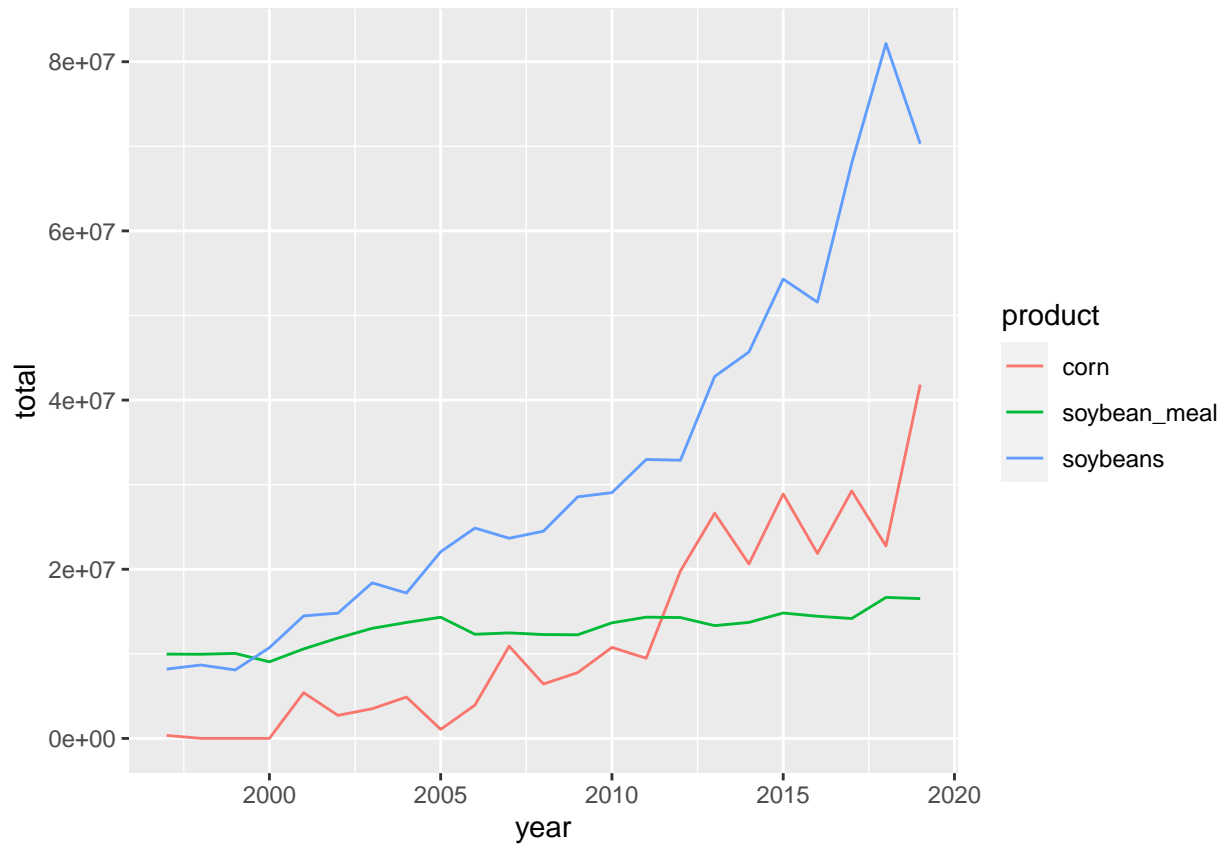
```
df6$year <- as.numeric(df6$year)
head(df6)
```

```
## # A tibble: 6 x 3
## # Groups:   year [2]
##   year product      total
##   <dbl> <fct>      <dbl>
## 1  1997 corn      356895.
## 2  1997 soybean_meal 9965945.
## 3  1997 soybeans    8195025.
## 4  1998 corn        6034.
## 5  1998 soybean_meal 9947875.
```

```
## 6 1998 soybeans      8679492.
```

Vamos visualizar a serie temporal dos dados:

```
df6 %>% group_by(product) %>% ggplot(aes(year,total,color=product))+geom_line()
```



Fazendo os subsets dos produtos:

```
soybean <- df6%>% filter(product=="soybeans")  
soybean_meal <- df6 %>% filter(product=="soybean_meal")  
corn <- df6 %>% filter(product=="corn")
```

Transformando os subsets and time series:

```
corn <- ts(corn[,3],start= 1997)  
soybean <- ts(soybean[,3],start = 1997)  
soybean_meal <- ts(soybean_meal[,3],start = 1997)
```

Aplicando forecast para cada um dos produtos:

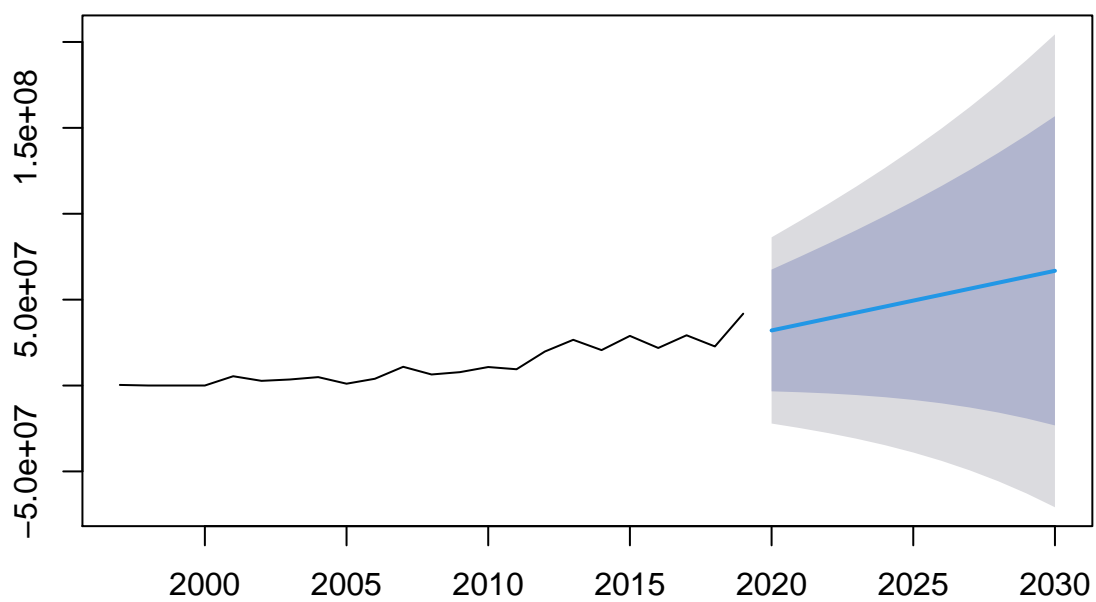
```
f.corn <- forecast(corn, h = 11)  
f.soybean <- forecast(soybean, h = 11)  
f.soybean_meal <- forecast(soybean_meal, h = 11)
```

Por fim, podemos ver uma previsao para os proximos 11 anos:

```
plot(f.corn, main = 'Corn 11 Years Forecast')
```

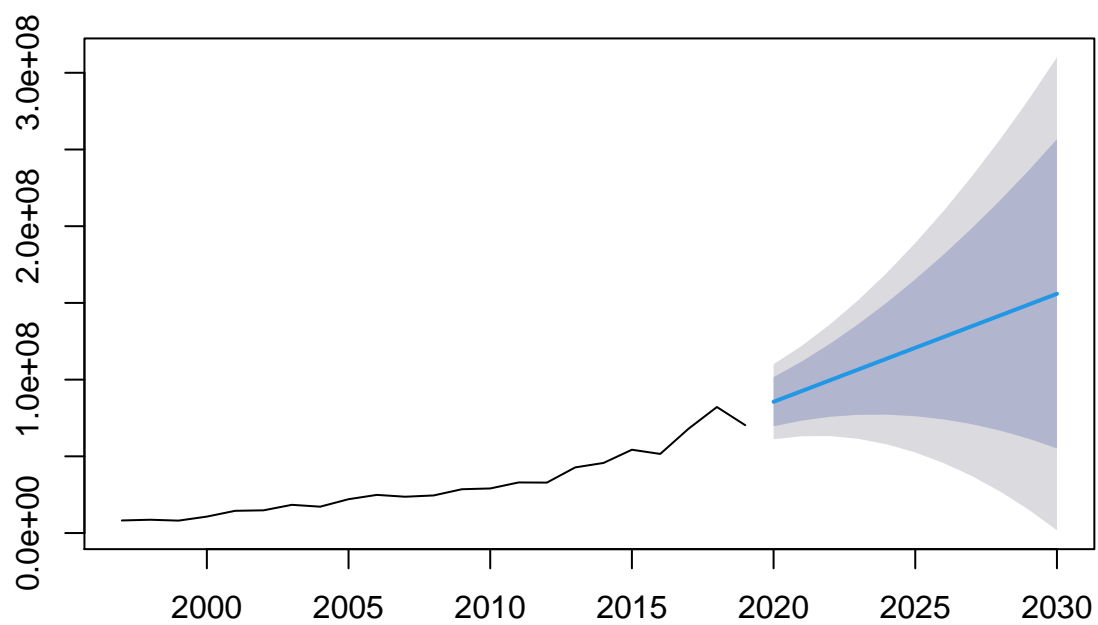


## Corn 11 Years Forecast



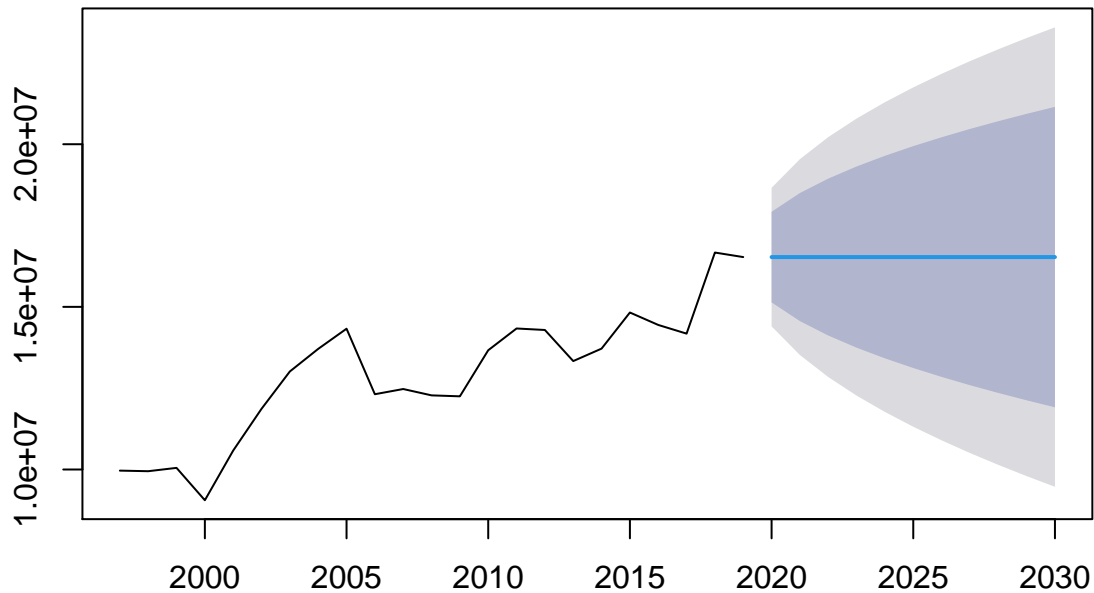
```
plot(f.soybean, main = 'Soybean 11 Years Forecast')
```

## Soybean 11 Years Forecast



```
plot(f.soybean_meal, main = 'Soybean Meal 11 Years Forecast')
```

## Soybean Meal 11 Years Forecast



Existem outras variaveis que sao importantes e devem ser analisadas, uma vez que a quantidade de exportacao depende tambem de outros fatores, como GDP Mundial e claro, o preco. Para isso, vamos analisar o comportamento da exportacao em relacao a essas variaveis.

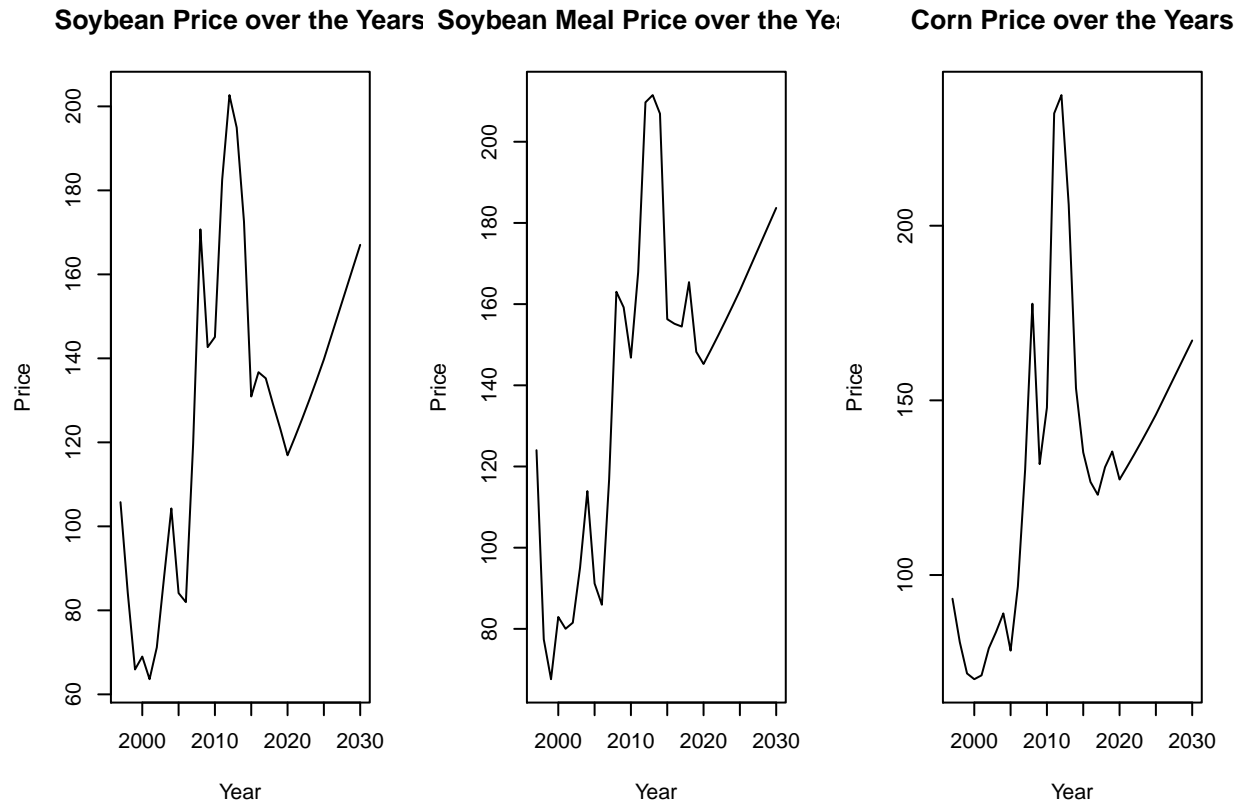
```
cov1 <- covar %>% filter(year >= 1997)
head(cov1)
```

```
## # A tibble: 6 x 13
##   year price_soybeans price_corn price_soybean_m~ gdp_china gdp_iran
##   <dbl>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1  1997           106.           93.2           124.           561.           107.
## 2  1998           84.2           80.8           77.4           604.           109.
## 3  1999           65.9           71.8           67.6           651.           109.
## 4  2000           69.0           70.2           82.9           706.           117.
## 5  2001           63.6           71.3           80.0           766.           118.
## 6  2002           71.2           79.0           81.5           835.           130.
## # ... with 7 more variables: gdp_netherlands <dbl>, gdp_spain <dbl>,
## #   gdp_thailand <dbl>, gdp_world <dbl>, gdp_egypt <dbl>, gdp_japan <dbl>,
## #   gdp_vietnam <dbl>
```

Vamos visualizar agora a evolucao temporal do preco dos produtos

```
par(mfrow= c(1,3))
with(cov1,plot(year,price_soybeans,type = 'l',xlab = 'Year',ylab= 'Price'))
title(main = "Soybean Price over the Years")
with(cov1,plot(year,price_soybean_meal,type = 'l',xlab = 'Year',ylab= 'Price'))
title(main = "Soybean Meal Price over the Years")
with(cov1,plot(year,price_corn,type = 'l',xlab = 'Year',ylab= 'Price'))
```

```
title(main = "Corn Price over the Years")
```



Os produtos possuem graficos parecidos. Portanto, em primeiro momento, dizer que suas variaveis correlatadas sao as mesmas.

Agora, vamos separar os produtos em subsets:

```
soybean <- df6 %>% filter(product=="soybeans")
soybean_meal <- df6 %>% filter(product=="soybean_meal")
corn <- df6 %>% filter(product=="corn")
head(soybean)
```

```
## # A tibble: 6 x 3
## # Groups:   year [6]
##   year product      total
##   <dbl> <fct>      <dbl>
## 1  1997 soybeans  8195025.
## 2  1998 soybeans  8679492.
## 3  1999 soybeans  8096035.
## 4  2000 soybeans 10725124.
## 5  2001 soybeans 14486902.
## 6  2002 soybeans 14806490.
```

uma vez que os dados foram separados, vamos junta-los a tabela cov1:

```
covcorn <- corn %>% inner_join(cov1, by = "year", suffix = c("_comex", "_cov"))
covsoybean <- soybean %>% inner_join(cov1, by = "year", suffix = c("_comex", "_cov"))
covsoybean_meal <- covsoybean <- soybean_meal %>% inner_join(cov1, by = "year", suffix = c("_comex", "_cov"))
```

```
head(covsoybean)
```

```
## # A tibble: 6 x 15
## # Groups:   year [6]
##   year product  total price_soybeans price_corn price_soybean_m~ gdp_china
##   <dbl> <fct>    <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1  1997 soybea~ 9.97e6          106.           93.2          124.          561.
## 2  1998 soybea~ 9.95e6           84.2           80.8           77.4          604.
## 3  1999 soybea~ 1.01e7           65.9           71.8           67.6          651.
## 4  2000 soybea~ 9.06e6           69.0           70.2           82.9          706.
## 5  2001 soybea~ 1.06e7           63.6           71.3           80.0          766.
## 6  2002 soybea~ 1.19e7           71.2           79.0           81.5          835.
## # ... with 8 more variables: gdp_iran <dbl>, gpd_netherlands <dbl>,
## #   gdp_spain <dbl>, gdp_thailand <dbl>, gdp_world <dbl>, gdp_egypt <dbl>,
## #   gdp_japan <dbl>, gdp_vietnam <dbl>
```

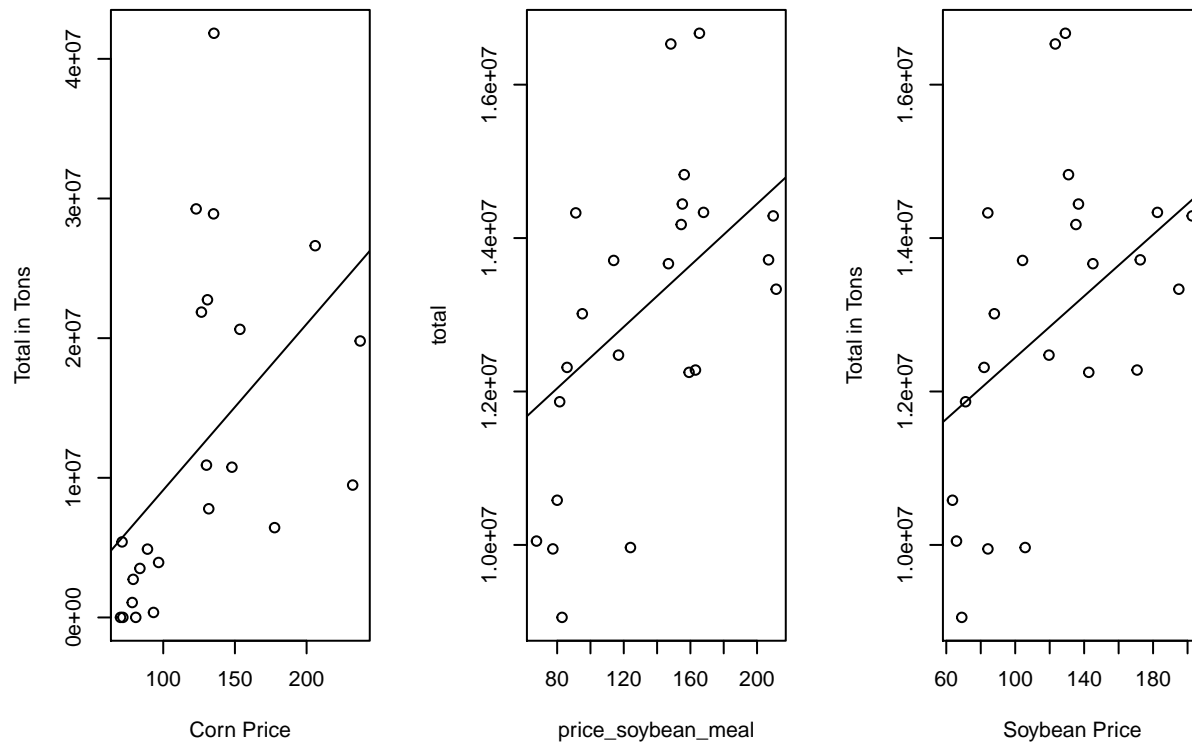
Agora que cada subset possui uma correspondencia com a tabela cov1, podemos fazer as analises com linhas de regressao. Vamos avaliar a correspondencia entra a quantidade de exportacao em toneladas com o GDP mundial e o preco de cada produto:

```
par(mfrow = c(1,3))
#Corn
with(covcorn,plot(price_corn,total, xlab="Corn Price", ylab = "Total in Tons"))
abline(lm(covcorn$total ~ covcorn$price_corn))

#Soybean Meal
with(covsoybean_meal,plot(price_soybean_meal,total))
abline(lm(covsoybean_meal$total ~ covsoybean_meal$price_corn, xlab="Soybean Meal Price", ylab = "Total in Tons"))

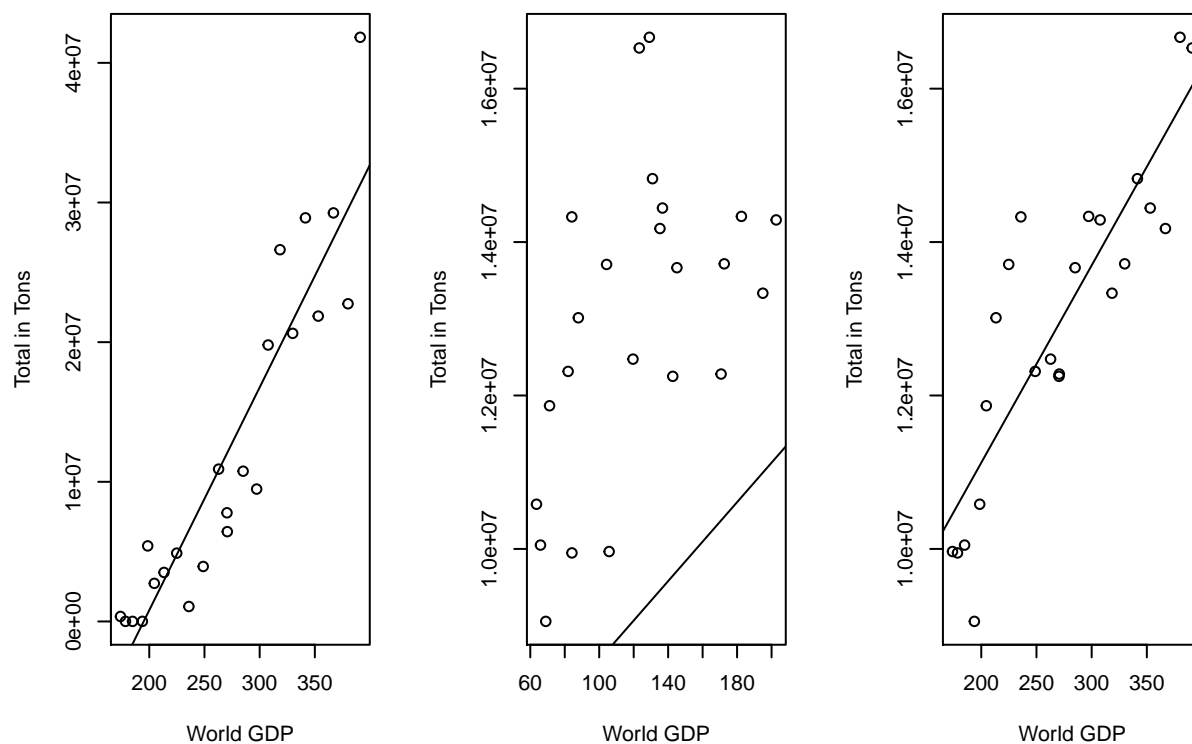
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##   extra arguments 'xlab', 'ylab' will be disregarded

#Soybeans
with(covsoybean,plot(price_soybeans,total, xlab="Soybean Price", ylab = "Total in Tons"))
abline(lm(covsoybean$total ~ covsoybean$price_corn))
```



para o GDP global:

```
par(mfrow = c(1,3))
#Corn
with(covcorn,plot(gdp_world,total, xlab= "World GDP",ylab = "Total in Tons"))
abline(lm(covcorn$total ~ covcorn$gdp_world))
#Soybeans
with(covsoybean,plot(price_soybeans,total, xlab= "World GDP",ylab = "Total in Tons"))
abline(lm(covsoybean$total ~ covsoybean$gdp_world))
#Soybean Meal
with(covsoybean_meal,plot(gdp_world,total, xlab= "World GDP",ylab = "Total in Tons"))
abline(lm(covsoybean_meal$total ~ covsoybean_meal$gdp_world))
```



Todas as projecoes indicam o crescimento da exportacao.