Projeto - Final

Nome: José Mauricio Nunes de Oliveira Junior
E-mail: jose.mauricio@aluno.ufabc.edu.br
Nome: Marcela Akemi Yamashita
E-mail: marcela.a@aluno.ufabc.edu.br

09 December, 2020

Contexto

A produção global de grãos abastece os mercados e casas do mundo inteiro. Entender a tendência da produção em diversos países pode trazer insights tanto sobre a economia global desses produtos, como o mercado interno de país quando relacionado a indicadores sociais e também até relações internacionais quando se leva outros dados em conta. Disponibilizado pelo "Our World in Data" e dentro tidytuesdayR, este conjunto de dados temos informações da produção de grãos para diversos países, mas também obtemos outros dados importantes como uso de fertilizantes, hectares de solo arado, tratores por metros quadrados e a população da região. Desta forma seria interessante explorar os dados e construir um modelo para predição de anos futuros a evolução de produção de grãos.

Importância do projeto

Este projeto se destaca pela construção de um modelo preditivo para produção anual de grãos para cada país ou região.

Carregando os dados e produzindo os datasets

- Para conjunto inicial de dados precisamos realizar alguns tratamentos.
  - Ao invés de tenta predizer cada grão individualmente vamos transformar varias colunas em uma coluna categórica apenas para definir o grão, *crop*, e uma variável numérica que diz a colheita numérica de toneladas por hectare referente ao grão produzido, *crop production*.
  - Decidimos que o nosso "target", nossa variável resposta, para o modelo seria o crop production, que é a *eficiência de plantio* de um país.
  - Para construção de variáveis de nosso modelo atrasamos todas as variáveis em um ano, então para o ano 1999, por exemplo, todas as variáveis serão referente ao ano 1998. Isto pois a finalidade do modelo é predizer informações, e não faria sentido variáveis do mesmo tempo de nossa variável resposta.
    * No entanto essa construção foi a respeito apenas de *nossa modelagem* toda exploração, todos os gráficos terão a correspodênncia correta em anos

```
## Warning: package 'tidytuesdayR' was built under R version 3.6.2
```

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'dplyr' was built under R version 3.6.2

## -- Conflicts ------------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## --- Compiling #TidyTuesday Information for 2020-09-01 ----

## --- There are 5 files available ---

## --- Starting Download ---

##
##  Downloading file 1 of 5: `arable_land_pin.csv`
##  Downloading file 2 of 5: `cereal_crop_yield_vs_fertilizer_application.csv`
##  Downloading file 3 of 5: `cereal_yields_vs_tractor_inputs_in_agriculture.csv`
##  Downloading file 4 of 5: `key_crop_yields.csv`
##  Downloading file 5 of 5: `land_use_vs_yield_change_in_cereal_production.csv`

## --- Download complete ---

## --- Compiling #TidyTuesday Information for 2020-09-01 ----

## --- There are 5 files available ---

## --- Starting Download ---

##
##  Downloading file 1 of 5: `arable_land_pin.csv`
##  Downloading file 2 of 5: `cereal_crop_yield_vs_fertilizer_application.csv`
##  Downloading file 3 of 5: `cereal_yields_vs_tractor_inputs_in_agriculture.csv`
##  Downloading file 4 of 5: `key_crop_yields.csv`
##  Downloading file 5 of 5: `land_use_vs_yield_change_in_cereal_production.csv`

## --- Download complete ---

## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

## Warning: Problem with `mutate()` input `Year`.
## i NAs introduced by coercion
## i Input `Year` is `as.integer(Year)`.

## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
## Warning: Problem with `mutate()` input `Year`.
## i NAs introduced by coercion
## i Input `Year` is `as.integer(Year)`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
## Warning: Problem with `mutate()` input `Year`.
## i NAs introduced by coercion
## i Input `Year` is `as.integer(Year)`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
## Warning: Problem with `mutate()` input `Year`.
## i NAs introduced by coercion
## i Input `Year` is `as.integer(Year)`.
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

Exploração inicial dos dados

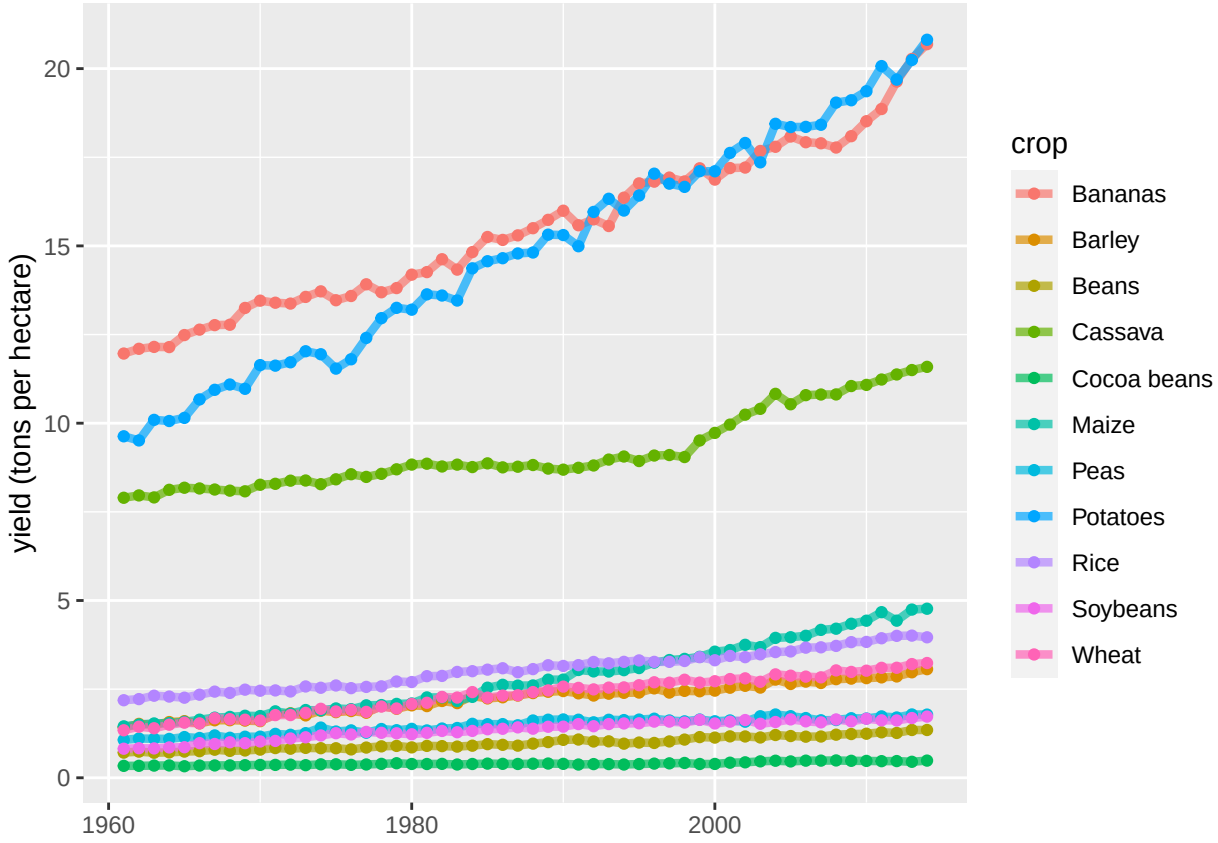
Vamos começar explorando a produção de cada. Vamos começar analisando globalmente

```
## [1] 52
```

```
## [1] 11
```

```
## [1] 572
```

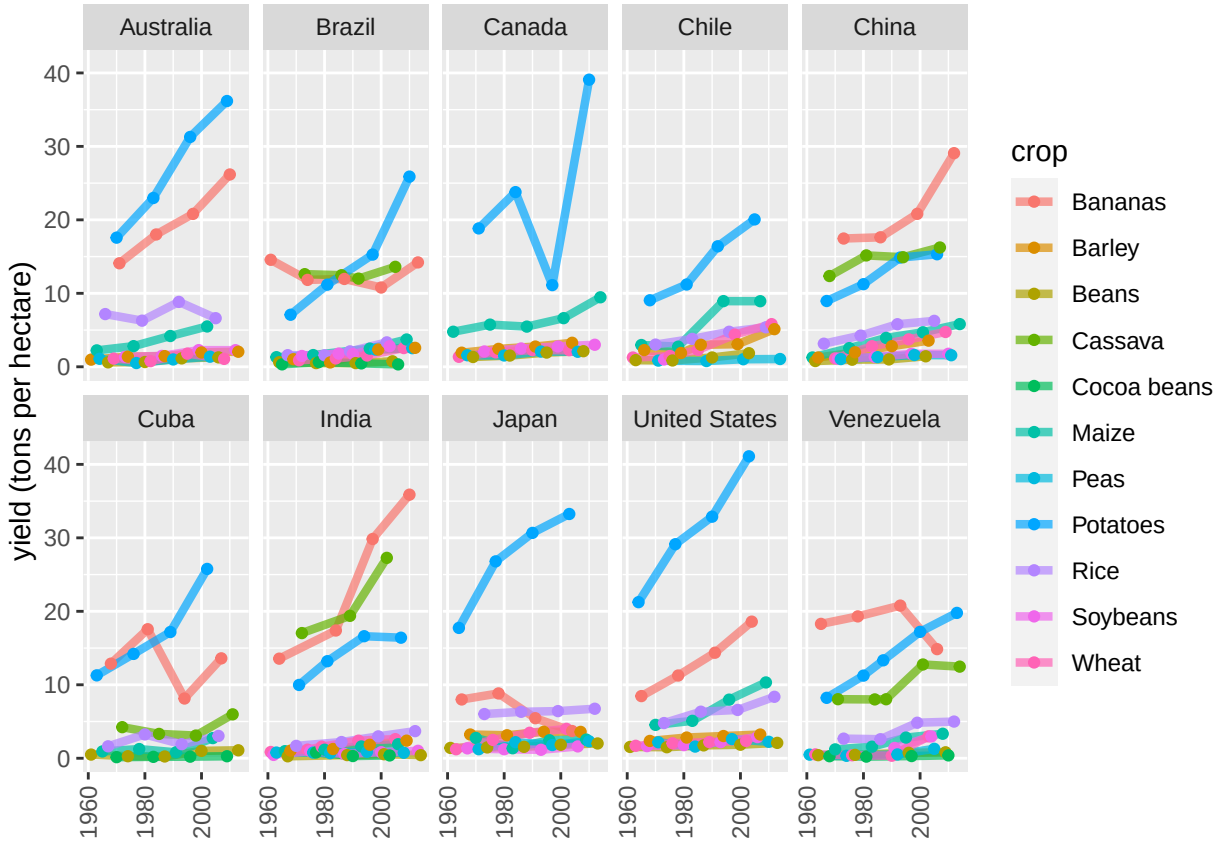
```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```



Observamos que todas as plantações tiveram um acréscimo de efieciência de plantio, sendo as batatas e bananas as produções que mais se desenvolveram furante os anos. Vamos observar agora alguns países que, ou

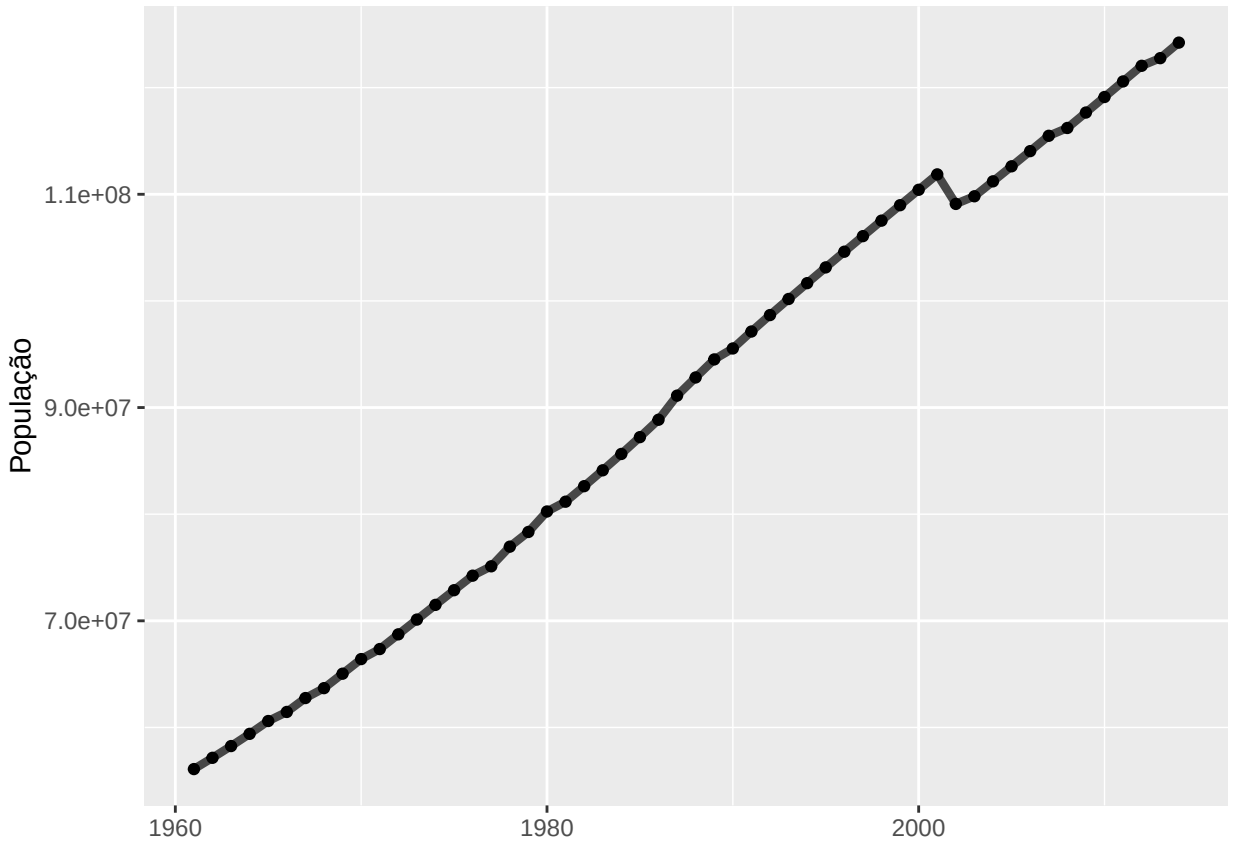
são relevantes políticamente de alguma forma, ou são grandes produtores de plantio. Também adicionamos alguns países da américa latina para comparação.

```
## Warning in entity == c("China", "Canada", "Australia", "Brazil", "United
## States", : longer object length is not a multiple of shorter object length
```

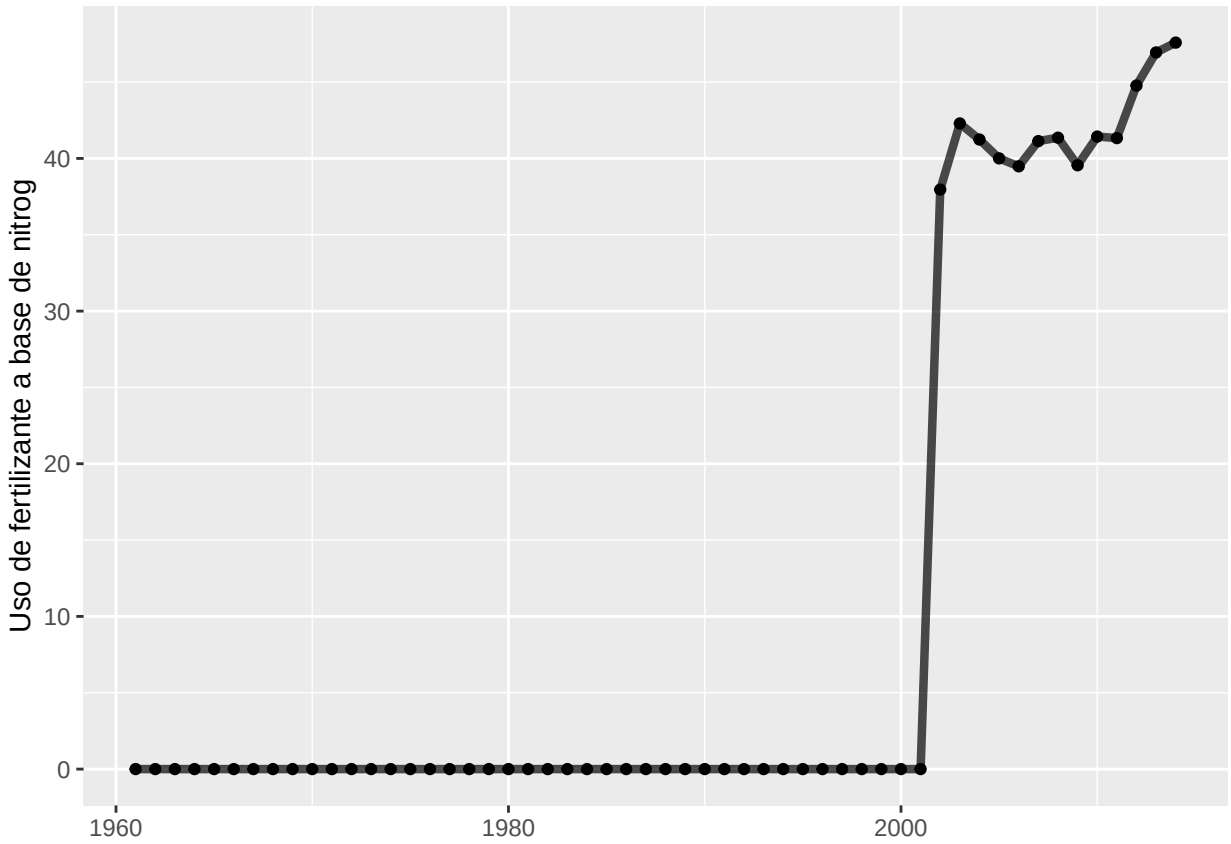


Existem algumas observações interessantes, países escolhidos possuem grande volume de bata sendo produzido, junto a banana, tendência mundial. Em alguns países "cassava" em alguns momentos é a primeira ou segunda maior produção. Agora vamos observar o crescimento populacional médio de cada país.

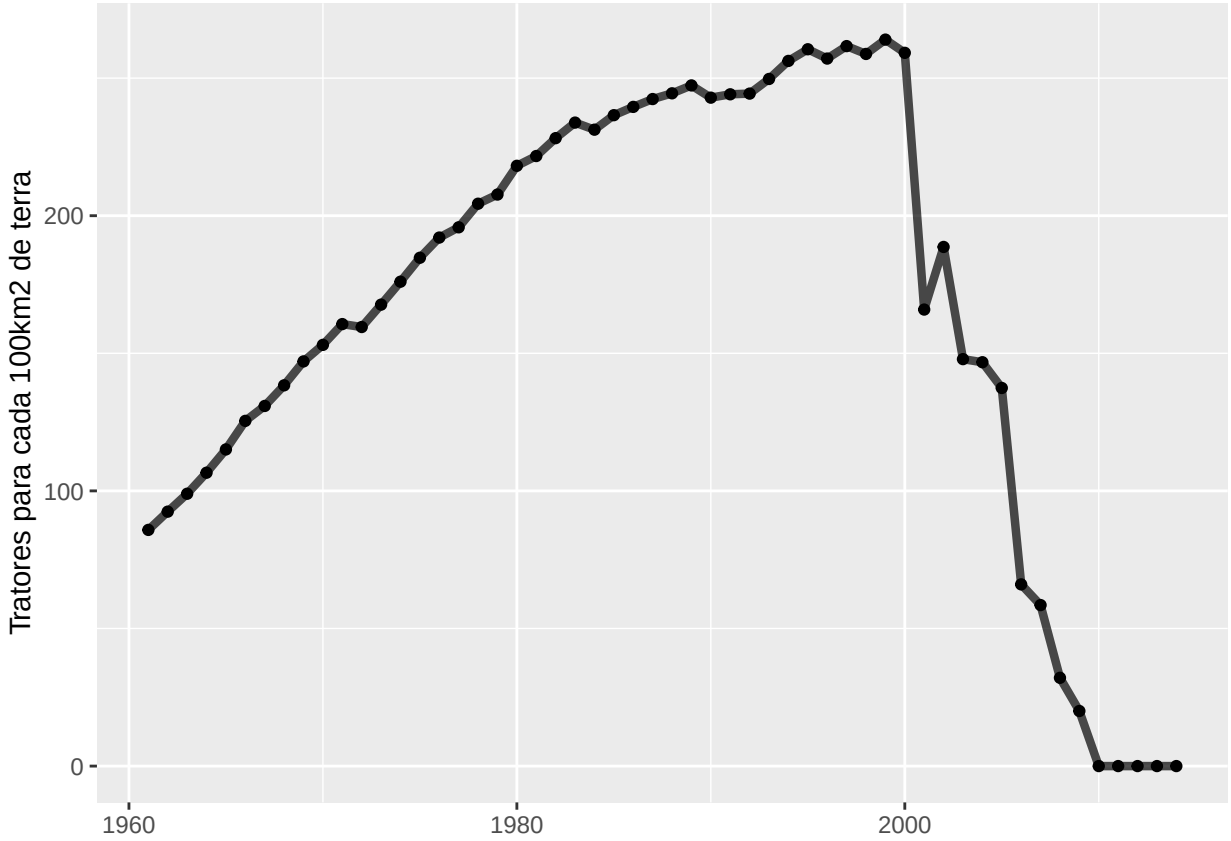
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



O crescimento populacional desde 1960 tem uma tendência positiva, com um crescimento linear (na média). Existem outras variáveis interessantes para explorar: O usode fertilizantes a base de nitrôgenio e uso de tratores por m2 na média para todos os países.



Para o uso de fertilizantes a base de nitrogênio é possível observar que é um tendência atuala partir dos anos 2000, apesar de apresentar um quadro inconsistente no começo dos anos 2000, apresenta um crescimento para os anos seguintes.



O uso de tratores vemos algo interessante: a partir de 2000 começa uma queda brusca até zerar. Essa variável parece um tanto quanto inconsistente, no entanto ainda materemos ela ainda no modelo, é possível que a quantia de tratores seja relacionada ao surgimento de novas tecnologias.

Modelagem

Um passo importante da nossa amostra é entender que não podemos amostrar aleatoriamente: se o nosso modelo sempre pretende "prever" o futuro não é válido usar um conjunto de testes que esteja num tempo anterior ao de treino. É importante perceber que num conjunto de dados temporal, as observações são depedentes entre si, não sendo válido a utilização de métodos de validação cruzada. Desta forma iremos calcular um tamanho de amostra desejavel para teste tendo certeza que o conjunto de treino sempre esteja no passado, e nosso conjunto de validação em um período seguinte. O conjunto de dados começa em 1961, como usamos até 3 anos de atraso como novas variáveis (produção anual no ano passado, retrasado e anterior a este) nosso tibble de dados começa em 1964 totalizando 89463 linhas, vamos usar aproximadamente os anos anterior a 2005, totalizando 78694 linhas (~80%) para treino, até o ano 2014 como teste para parametrização de modelos e finalmente o ano 2015 como um última validação, ou seja, uma simulação de aplicação "real" do nosso modelo.

```
## Warning: package 'tidymodels' was built under R version 3.6.2
```

```
## -- Attaching packages ---------------------------------------------------------- tidymodels 0.1.1 --
```

```
## v broom     0.7.1      v recipes   0.1.13
## v dials     0.0.9      v rsample   0.0.8
## v infer     0.5.3      v tune      0.1.1
## v modeldata 0.0.2      v workflows 0.2.1
## v parsnip   0.1.3      v yardstick 0.0.7
```

```
## Warning: package 'broom' was built under R version 3.6.2
```

```
## Warning: package 'dials' was built under R version 3.6.2
## Warning: package 'scales' was built under R version 3.6.2
## Warning: package 'infer' was built under R version 3.6.2
## Warning: package 'modeldata' was built under R version 3.6.2
## Warning: package 'parsnip' was built under R version 3.6.2
## Warning: package 'recipes' was built under R version 3.6.2
## Warning: package 'rsample' was built under R version 3.6.2
## Warning: package 'tune' was built under R version 3.6.2
## Warning: package 'workflows' was built under R version 3.6.2
## Warning: package 'yardstick' was built under R version 3.6.2
## -- Conflicts ------------------------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## Warning in predict.lm(object = object$fit, newdata = new_data, type =
## "response"): prediction from a rank-deficient fit may be misleading
## Warning in predict.lm(object = object$fit, newdata = new_data, type =
## "response"): prediction from a rank-deficient fit may be misleading
## Warning in predict.lm(object = object$fit, newdata = new_data, type =
## "response"): prediction from a rank-deficient fit may be misleading
```

Modelo Linear

Pelo vamor dos RMSE, podemos ver que o erro do conjunto teve um resultado razoavel se comparado ao resultado do conjunto de testes e abaixo do conjunto de validação, o que nos leva a conclusão de que o modelo obteve um resultado satisfatório na predição dos dados.

```
cat("RMSE para Treino: ", treino_rmse, "\n")
```

```
## RMSE para Treino:  1.069008
```

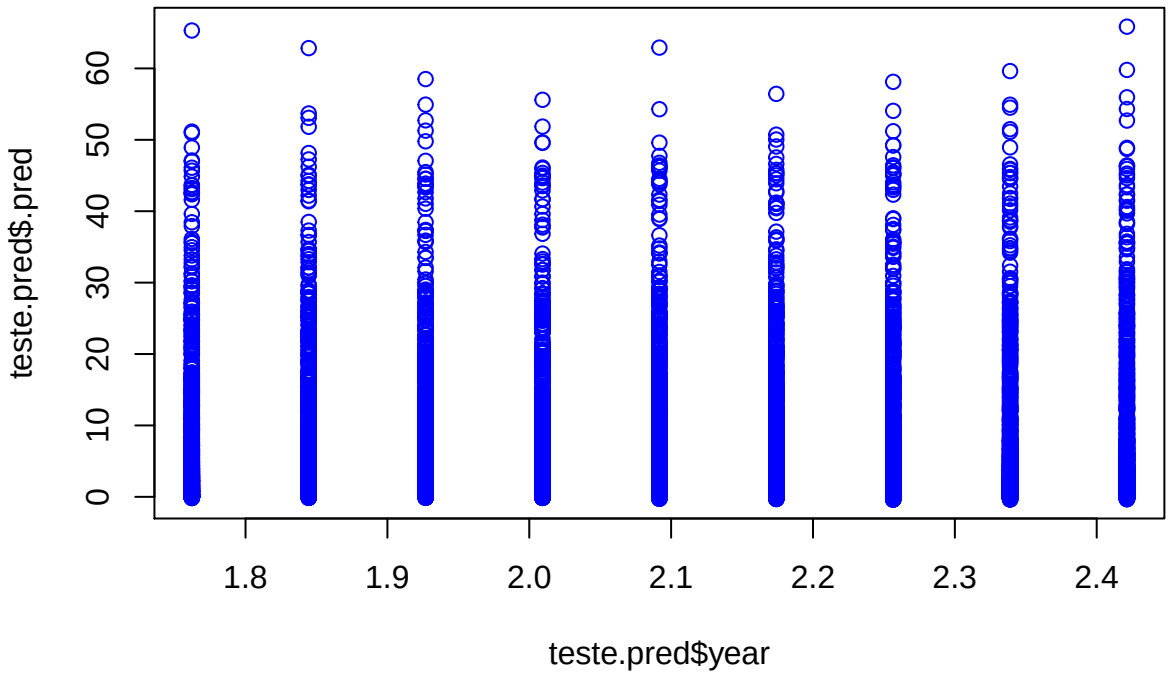
```
cat("RMSE para Teste: ", teste_rmse, "\n")
```

```
## RMSE para Teste:  1.277265
```

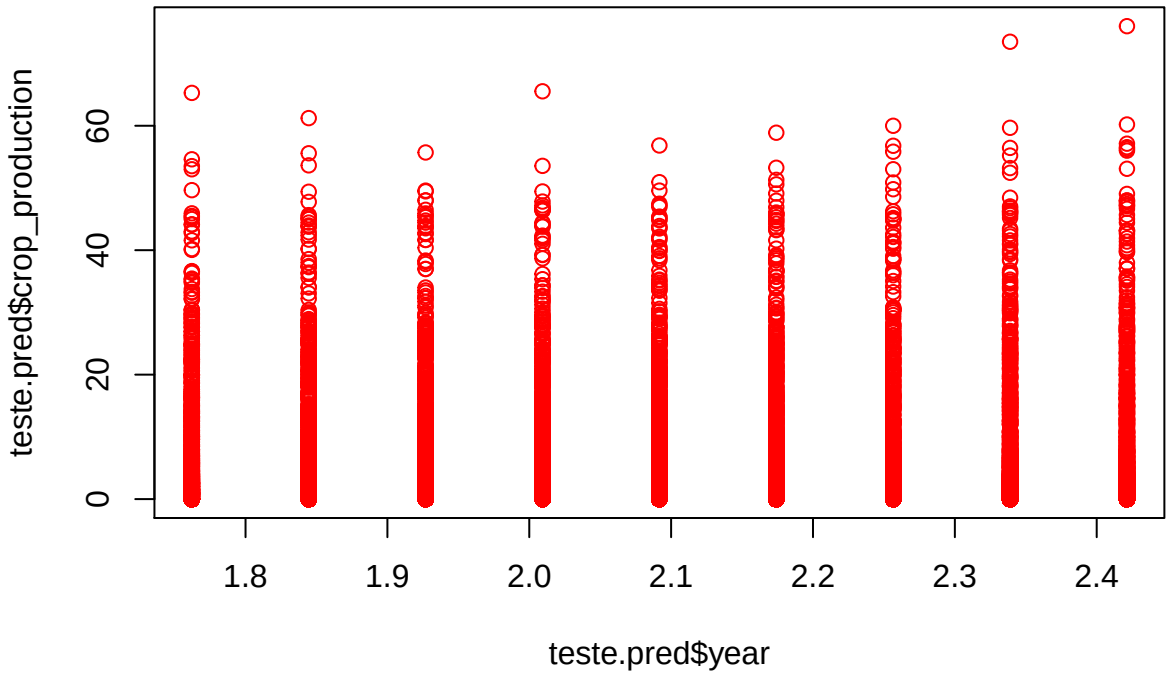
```
cat("RMSE para Valida: ", valida_rmse)
```

```
## RMSE para Valida:  1.783481
```

```
plot(teste.pred$year, teste.pred$.pred, type = "p", col="blue")
```




```
plot(teste.pred$year, teste.pred$crop_production, type = "p", col="red")
```




```
teste
```

```
## # A tibble: 15,609 x 17
##    entity code   year crop  crop_production last_year_produ~ last_2year_prod~
##    <fct>  <fct> <dbl> <fct>           <dbl>            <dbl>            <dbl>
##  1 Afgha~ AFG    2006 Wheat            1.38             1.82             1.27
##  2 Afgha~ AFG    2006 Rice             3.38             3.03             2.37
##  3 Afgha~ AFG    2006 Maize            2.62             1.21             1.6
##  4 Afgha~ AFG    2006 Soyb~            0                0                0
##  5 Afgha~ AFG    2006 Pota~           15               15               17.6
```

```
##  6 Afgha~ AFG    2006 Beans            0                0              0
##  7 Afgha~ AFG    2006 Peas             0                0              0
##  8 Afgha~ AFG    2006 Cass~            0                0              0
##  9 Afgha~ AFG    2006 Barl~            1.54             1.40           0.921
## 10 Afgha~ AFG    2006 Coco~            0                0              0
## # ... with 15,599 more rows, and 10 more variables:
## #   last_3year_production <dbl>, `Cereal yield (tonnes per hectare)` <dbl>,
## #   `Nitrogen fertilizer use (kilograms per hectare)` <dbl>, `Tractors per 100
## #   sq km arable land` <dbl>, `Cereal yield (kilograms per hectare) (kg per
## #   hectare)` <dbl>, `Total population (Gapminder).x` <dbl>, `Cereal yield
## #   index` <dbl>, `Change to land area used for cereal production since
## #   1961` <dbl>, `Total population (Gapminder).y` <dbl>, `Arable land needed to
## #   produce a fixed quantity of crops ((1.0 = 1961))` <dbl>
```

Modelo Xgboost (Gradient Boosting com árvore)

Agora iremos testar nossa base em um modelo de árvores de gradiente usando o algoritmo de xgboost.

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

Observamos o valor do RMSE para esse algoritmo:

```
cat("RMSE para Treino: ", treino_rmse, "\n")
```

```
## RMSE para Treino:  1.179947
```

```
cat("RMSE para Teste: ", teste_rmse, "\n")
```

```
## RMSE para Teste:  1.581164
```

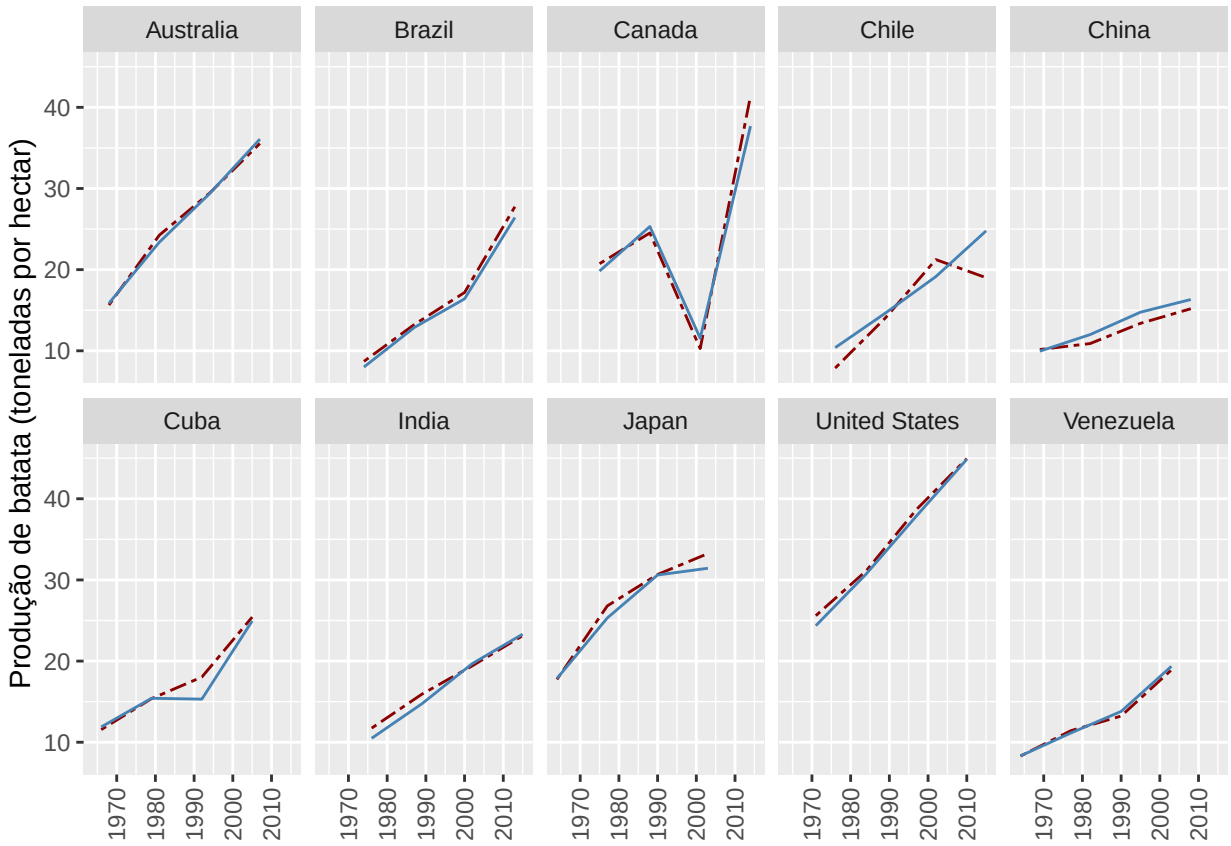
```
cat("RMSE para Valida: ", valida_rmse)
```

```
## RMSE para Valida:  1.9428
```

Vamos visualizar a predição para a produção de batatas em certos países (usaremos a batata por ser um dos grãos com maior produção nos países visualizados anteriormente):

```
## Warning in `==.default`(entity, c("China", "Canada", "Australia", "Brazil", :
## longer object length is not a multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```



Random Forest

```
tree_randforest <- rand_forest( min_n = 5,
        trees = 500,
        mtry = sqrt(ncolunas)
        ) %>%
        set_engine("randomForest") %>%
        set_mode("regression")
rf<-tree_randforest%>% fit(crop_production ~.,treino.prep)

treino.pred <- treino.prep %>%
    bind_cols(rf %>% predict(new_data = treino.prep))

teste.pred <- teste.prep %>%
    bind_cols(rf %>% predict(new_data = teste.prep))

valida.pred <- valida.prep %>%
    bind_cols(rf %>% predict(new_data = valida.prep))

teste_rmse <- RMSE(teste.pred$crop_production, teste.pred$.pred)
treino_rmse <- RMSE(treino.pred$crop_production, treino.pred$.pred)
valida_rmse <- RMSE(valida.pred$crop_production, valida.pred$.pred)

cat("RMSE para Treino: ", treino_rmse, "\n")
```

```
## RMSE para Treino:  1.077061
```

```
cat("RMSE para Teste: ", teste_rmse, "\n")
```

```
## RMSE para Teste:  1.796408
```

```
cat("RMSE para Valida: ", valida_rmse)
```

```
## RMSE para Valida:  2.247694
```

```
## Warning in `==.default`(entity, c("China", "Canada", "Australia", "Brazil", :
## longer object length is not a multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```