# Crop Statistics 1900-2017 dataset analasys

Jetze Luyten, Axel Van Gestel, David Silva Troya

## Setup the enviroment

The first thing to setup for the analysts is the environment with the required packages and settings.

### Install required packages and load required libraries

During this analysis we used the `tidyverse` package for reading, cleaning and plotting the data and the `ggcorrplot` package to visualize the correlation matrix into a heat map.

```
# install.packages("tidyverse")
# install.packages("ggcorrplot")
library(tidyverse) # Contains all tidyverse packages (ggplot2, dplyr, ...)
library(readxl) #  Need to load explicitly (not a core tidyverse package)
library(ggcorrplot) # Used for generating correlation heatmaps (uses ggplot2)
```

### Setup enviroment settings

In the following code block we set the language R uses for it's messages to English, clear all the global variables so that we always start with a clean slate and setup ggplot to center the plot titles by default.

```
Sys.setenv(LANG = "en") # Set language to English
rm(list = ls()) # Clears the Global Env
theme_update(plot.title = element_text(hjust = 0.5)) # Center all plot titles
```

## Read and import the data set

### Read the data set (uses readr)

```
column_types <- c(
  "numeric", # ...1
  "numeric", # Harvest_year
  "text",    # admin0
  "text",    # admin1
  "text",    # crop
  "numeric", # hectares (ha)
  "numeric", # production (tonnes)
  "numeric", # year
  "numeric", # yield(tonnes/ha)
  "text",    # admin2
  "text"     # notes
)
crops <- read_xlsx(
  path = "./crops/food-twentieth-century-crop-statistics-1900-2017-xlsx.xlsx",
  sheet = "CropStats",
  col_types = column_types)
```

```
## New names:
## * `` -> `...1`
```

### Drop not needed columns

```
crops <- select(crops, -c(...1, admin2, notes, Harvest_year))

crops <- crops %>% mutate(crop = factor(crop,
                                        levels = c("wheat", "winter wheat",
                                                   "spring wheat", "maize",
                                                   "cereals"),
                                        ordered = TRUE))

# Winter and Spring Wheat are only for a few countries.
# For that reason will be all united as one "wheat" column
crops$crop <- recode_factor(crops$crop,
                            "winter wheat" = "wheat",
                            "spring wheat"= "wheat",
                            "wheat"= "wheat",
                            "maize"= "maize",
                            "cereals"= "cereals",
                              .default = "Unknown", # NA -> Unknown
                              .ordered = TRUE)

crops <- crops %>% mutate(admin0 = as.factor(admin0))

crops <- crops %>% mutate(year = as.integer(year))
```

### Clear not needed variables

```
rm(column_types)
```

## Filtering and cleaning

#### Check for the number of NA's in each column

```
sanity_check <- function(my_df) {
  for (j in 1:ncol(my_df)) {
    print(paste(names(my_df[j]), ":", sum(is.na(my_df[, j]))))
  }
}

sanity_check(crops)
```

```
## [1] "admin0 : 0"
## [1] "admin1 : 2934"
## [1] "crop : 0"
## [1] "hectares (ha) : 1623"
## [1] "production (tonnes) : 1998"
## [1] "year : 0"
## [1] "yield(tonnes/ha) : 2013"
```

**View 'crop' tibble**

```
crops
```

```
## # A tibble: 36,707 x 7
##    admin0  admin1 crop  `hectares (ha)` `production (tonnes)`  year yield(tonn~1
##    <fct>   <chr>  <ord>           <dbl>                 <dbl> <int>        <dbl>
##  1 Austria <NA>   wheat              NA                    NA  1902         1.31
##  2 Austria <NA>   wheat              NA                    NA  1903         1.47
##  3 Austria <NA>   wheat              NA                    NA  1904         1.27
##  4 Austria <NA>   wheat              NA                    NA  1905         1.33
##  5 Austria <NA>   wheat              NA                    NA  1906         1.28
##  6 Austria <NA>   wheat              NA                    NA  1907         1.37
##  7 Austria <NA>   wheat              NA                    NA  1908         1.36
##  8 Austria <NA>   wheat              NA                    NA  1909         1.35
##  9 Austria <NA>   wheat              NA                    NA  1910         1.18
## 10 Austria <NA>   wheat              NA                    NA  1911         1.37
## # ... with 36,697 more rows, and abbreviated variable name
## #   1: `yield(tonnes/ha)`
```

```
tail(crops)
```

```
## # A tibble: 6 x 7
##   admin0 admin1   crop  `hectares (ha)` `production (tonnes)`  year yield(tonn~1
##   <fct>  <chr>    <ord>           <dbl>                 <dbl> <int>        <dbl>
## 1 China  zhejiang wheat           74490                271000  2012         3.64
## 2 China  zhejiang wheat           75520                278300  2013         3.69
## 3 China  zhejiang wheat           82120                309500  2014         3.77
## 4 China  zhejiang wheat           89800                351300  2015         3.91
## 5 China  zhejiang wheat           76590                253900  2016         3.32
## 6 China  zhejiang wheat          103670                419200  2017         4.04
## # ... with abbreviated variable name 1: `yield(tonnes/ha)`
```

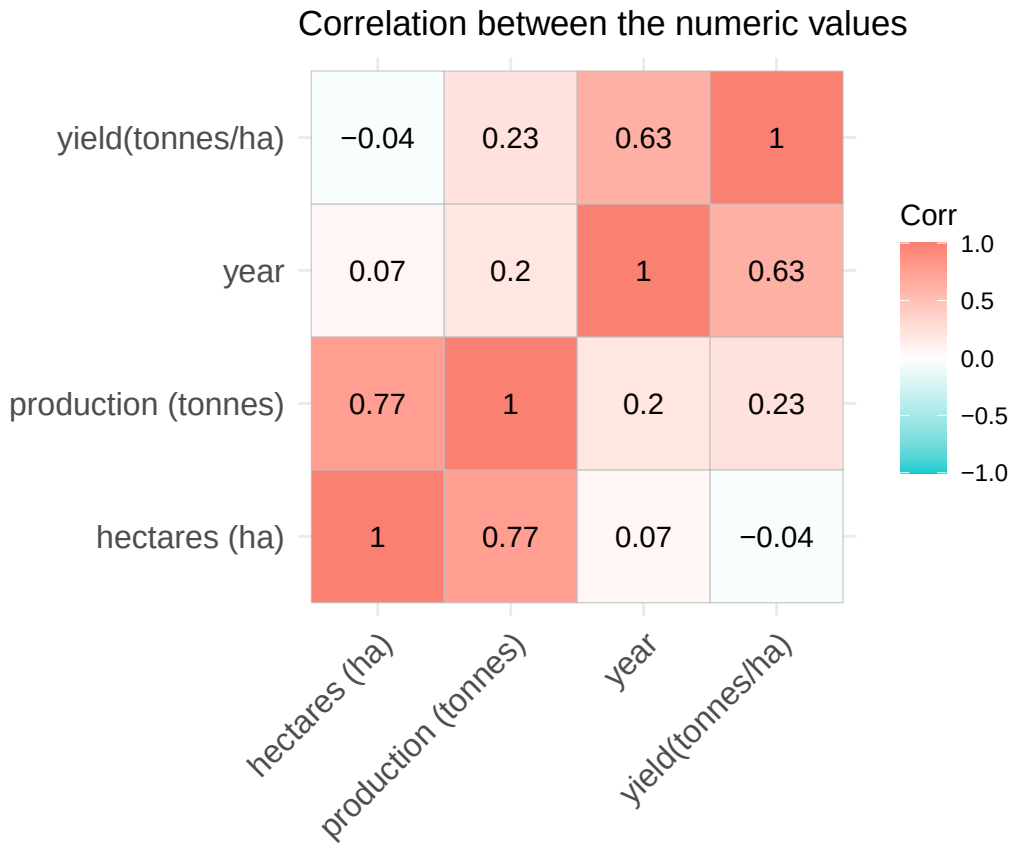
## Correlation heatmap (uses ggcorrplot)

**Generate a correlation heatmap of the numeric values**

```
crops_numeric <- select(crops,
                        `hectares (ha)`,
                        `production (tonnes)`,
                        year ,
                        `yield(tonnes/ha)`)

crops_numeric_corr <- cor(crops_numeric, use = "complete.obs") # Use only non NA

ggcorrplot::ggcorrplot(crops_numeric_corr,
                       lab = TRUE, # Show correlation coefficients
                       colors = c("darkturquoise", "white", "salmon"),
                       title = "Correlation between the numeric values")
```

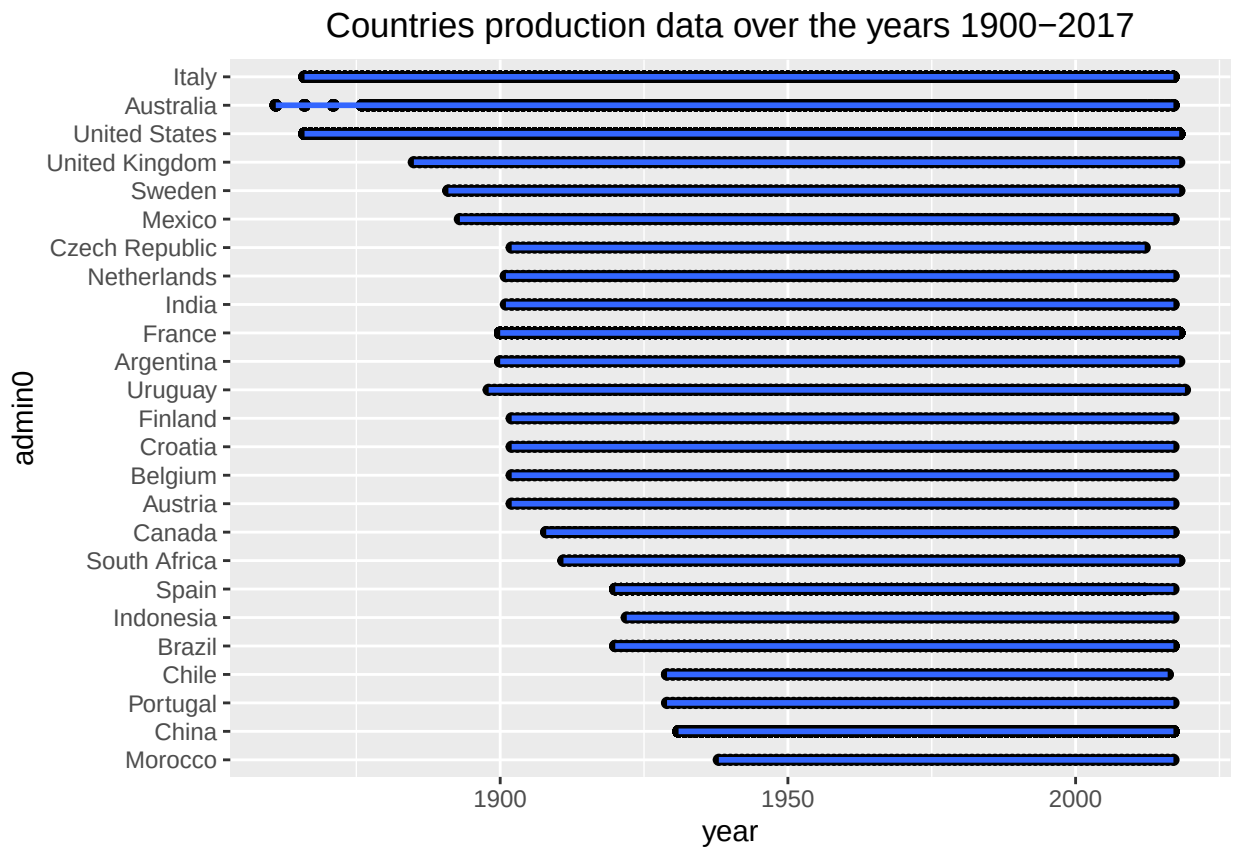



```
# Clear not needed variables
rm(crops_numeric, crops_numeric_corr)
```

## Plots and stuff (uses ggplot2)

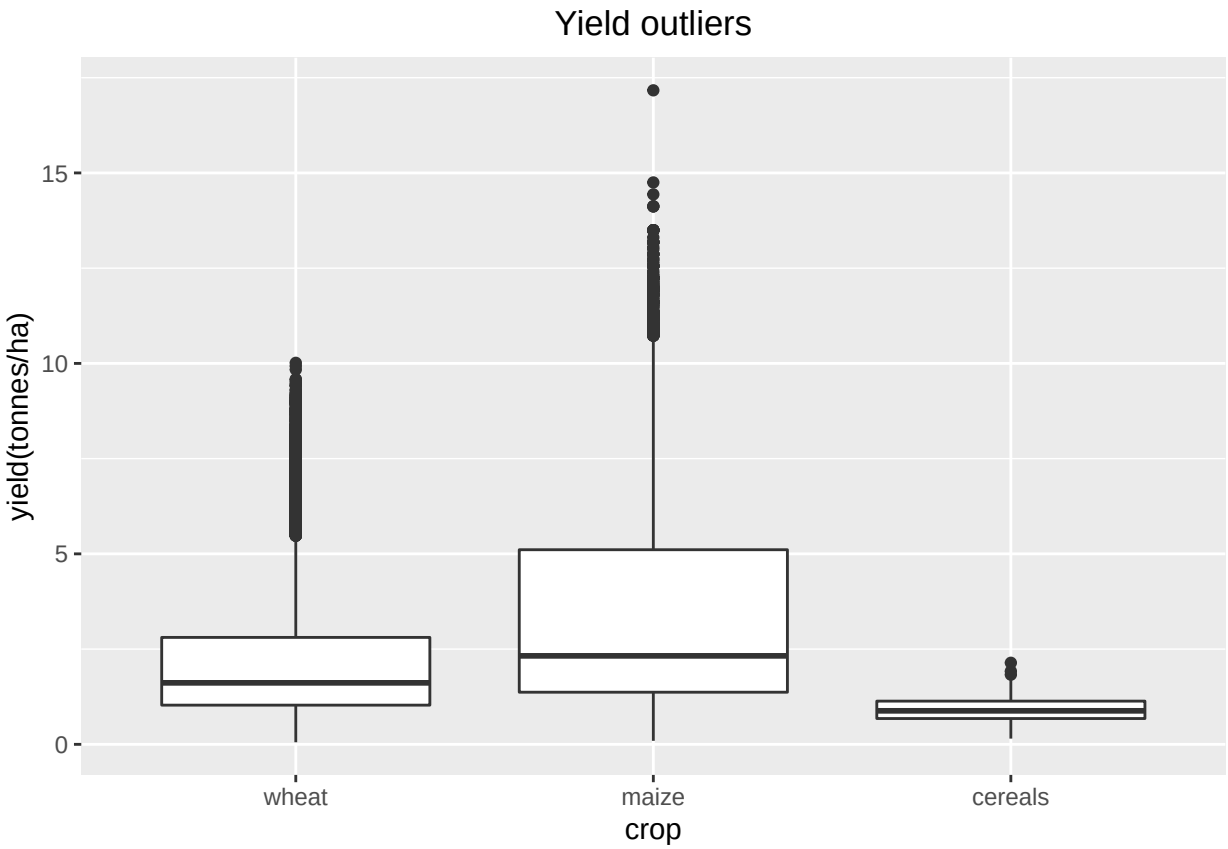
### Production x years

```
crops %>% mutate(admin0 = fct_reorder(admin0, desc(year) )) %>%
  ggplot(mapping = aes(x = year , y = admin0)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1)+
  ggtitle("Countries production data over the years 1900-2017")
```



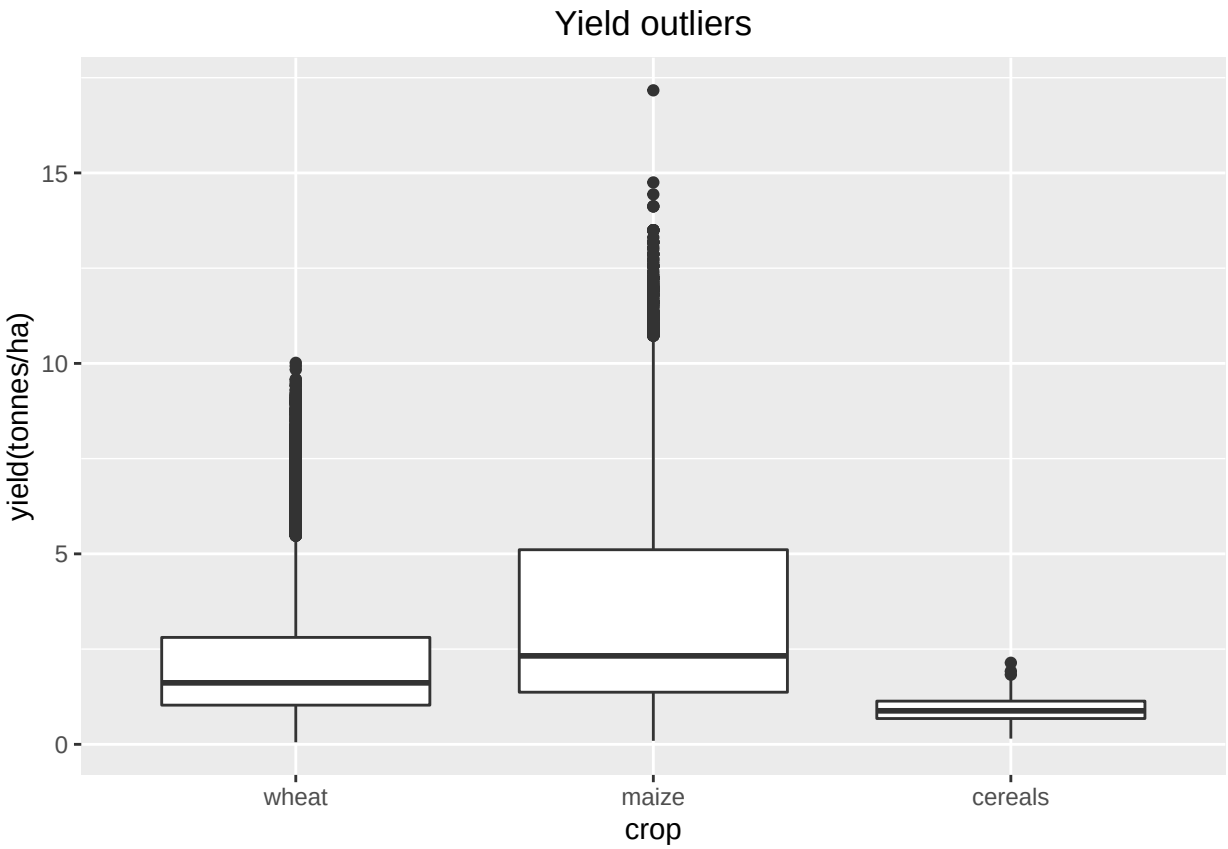

**Yield outliers**

```
ggplot(data = crops, mapping = aes(x = crop, y = `yield(tonnes/ha)`)) +
  geom_boxplot() +
  ggtitle("Yield outliers")
```



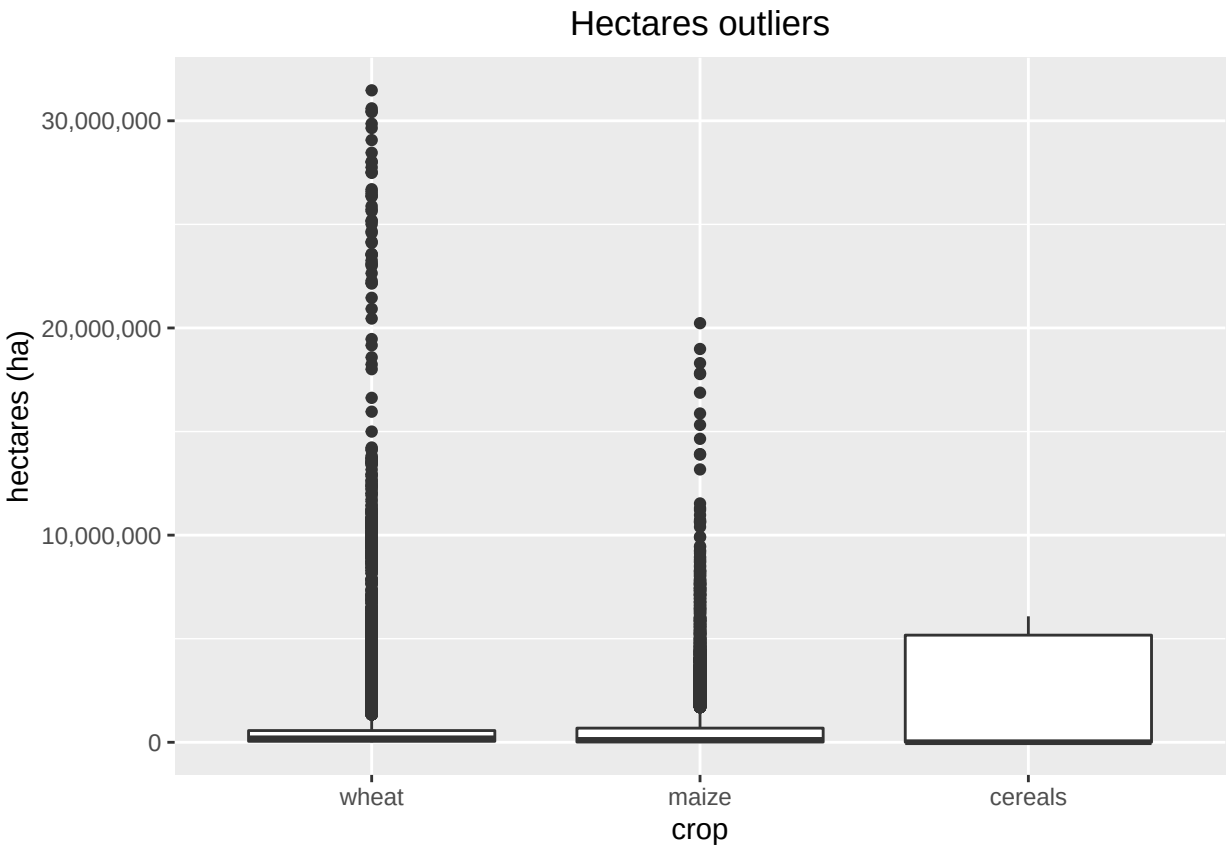

**Count of crops**

```
ggplot(data = crops, mapping = aes(x = crop, y = `yield(tonnes/ha)`)) +
  geom_boxplot() +
  ggtitle("Yield outliers")
```




**??? Hectares outliers ???**

```
ggplot(data = crops, mapping = aes(x = crop, y = `hectares (ha)`)) +
  geom_boxplot() +
  scale_y_continuous(labels = scales::comma) +
  ggtitle("Hectares outliers")
```




# Productions over the years (splitted in 3 sections)

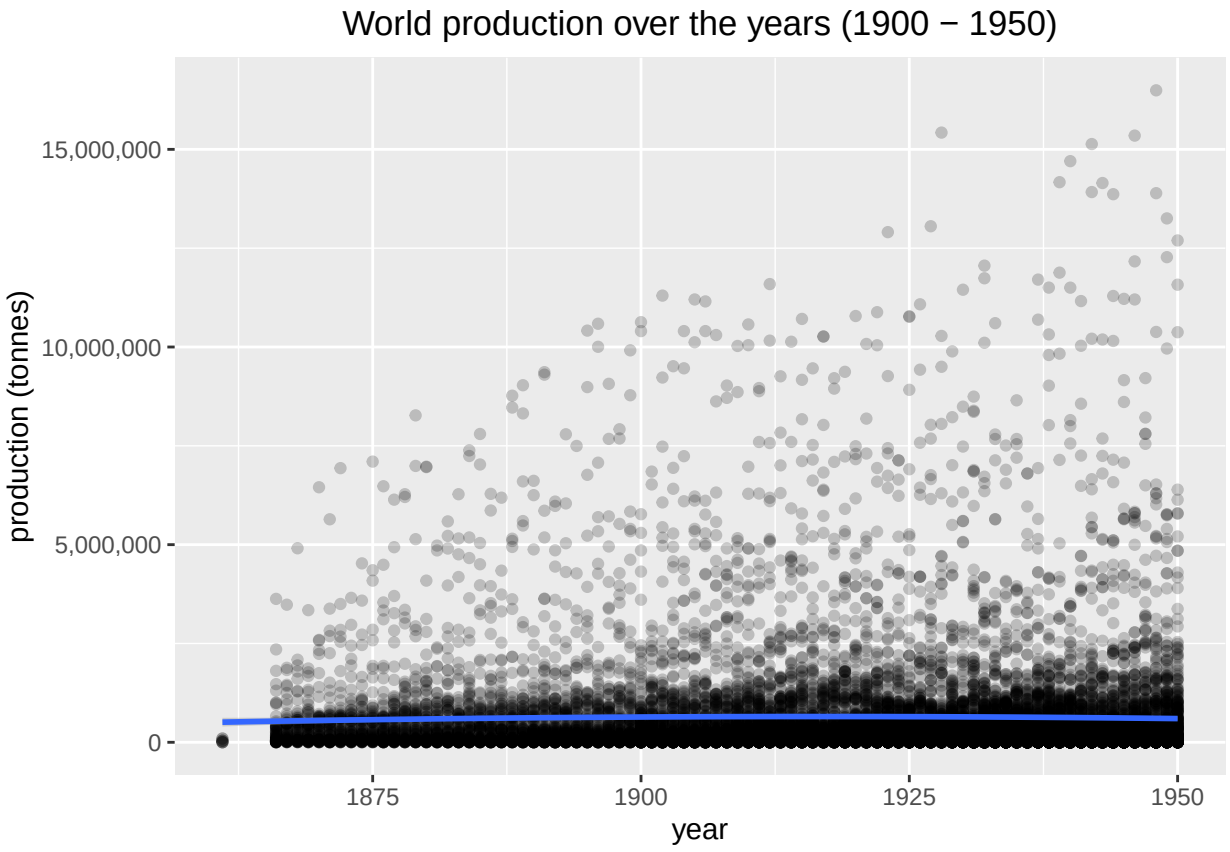
## Section 1 from 1900-1950

### To check the Production from the first 50 years (1900 - 1950)

```
Production1900to1950year <- filter(crops, year <= 1950) # Under the 1950 years
Production1900to1950year <- mutate(Production1900to1950year,
                                   admin0 = fct_reorder(admin0, desc(year) )) # Sorting the data
```

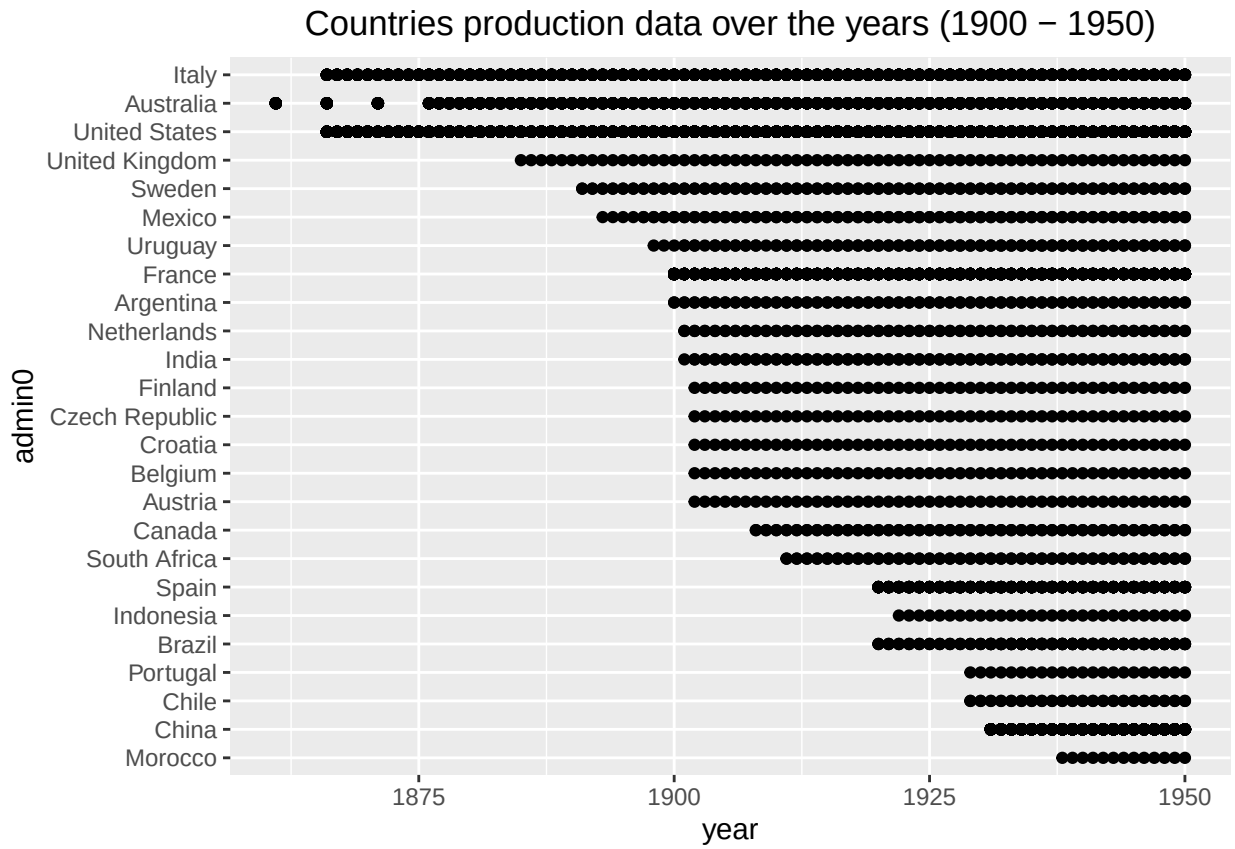
#### World Production(1900 - 1950)

```
ggplot(data = Production1900to1950year, mapping = aes(x = year, y = `production (tonnes)`)) +
  geom_point(alpha = 2/10) +
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1) +
  scale_y_continuous(labels = scales::comma) +
  ggtitle("World production over the years (1900 - 1950)")
```



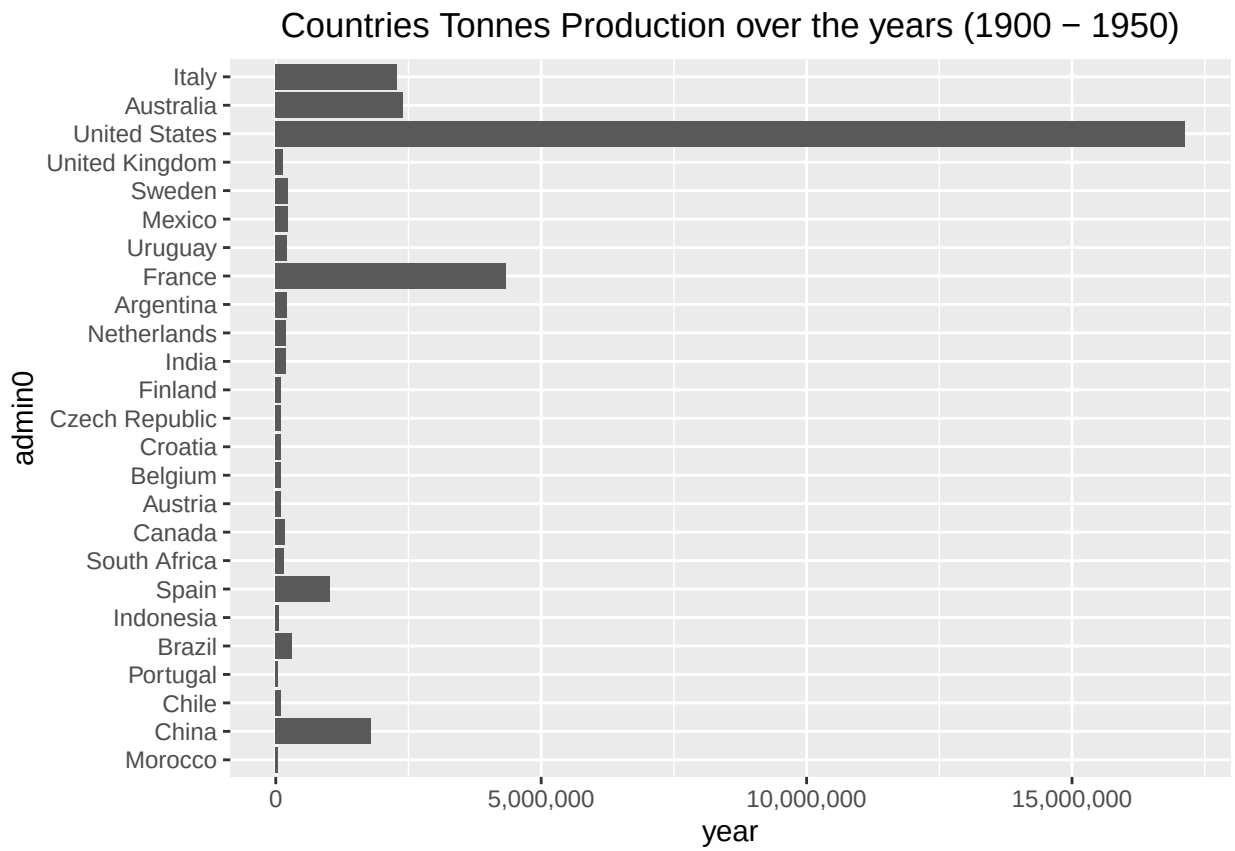

**Countries with more data of Production**

```
ggplot(data = Production1900to1950year, mapping = aes(x = year , y = admin0)) +
  geom_point() +
  ggtitle("Countries production data over the years (1900 - 1950)")
```




**Production of each country**

```
ggplot(data = Production1900to1950year, mapping = aes(x = year , y = admin0)) +
  geom_bar(stat = "identity") +
  scale_x_continuous(labels = scales::comma) +
  ggtitle("Countries Tonnes Production over the years (1900 - 1950)")
```



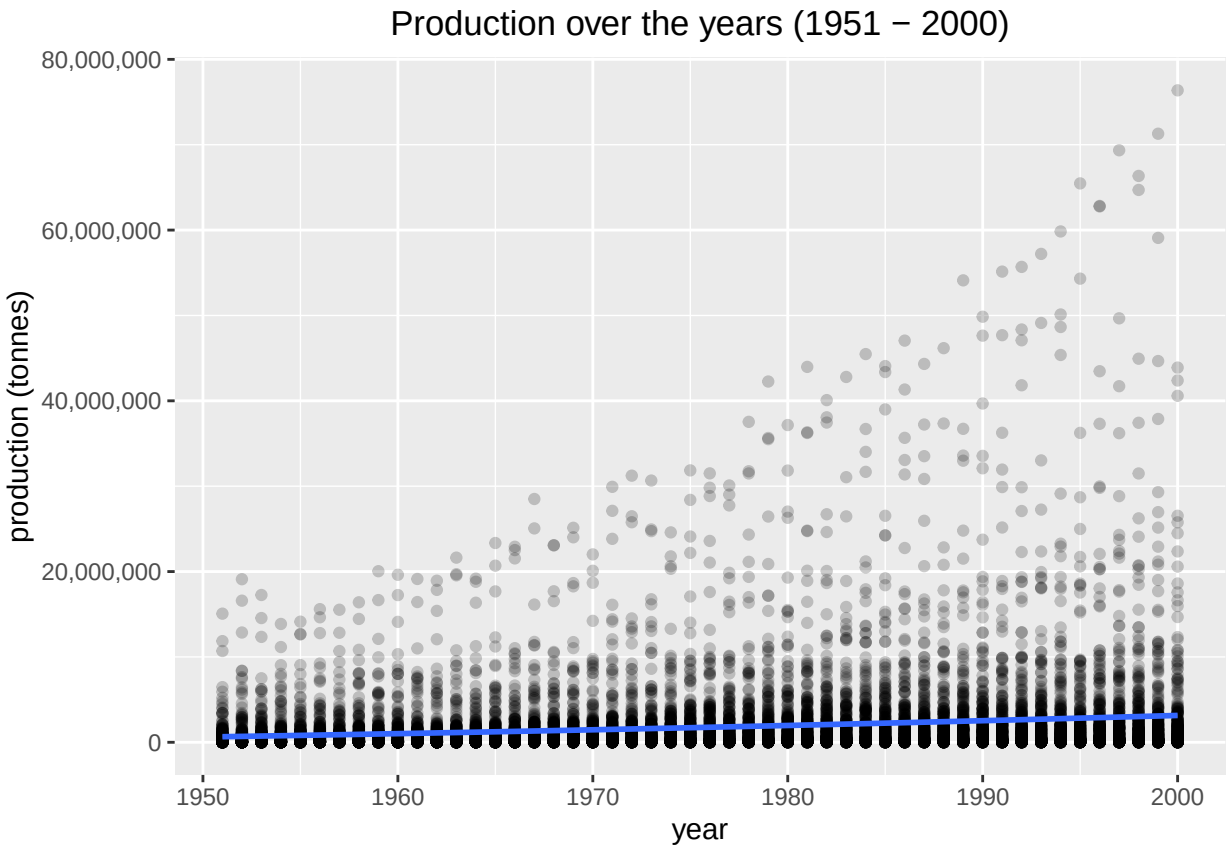

# Section 2 from 1951-2000

## To check the Production (1951 - 2000)

```
Production1950to2000year <- filter(crops, year > 1950 & year <= 2000) # Between the 1951-2000 years
Production1950to2000year <- mutate(Production1950to2000year,
                                   admin0 = fct_reorder(admin0, desc(year) )) # Sorting the data
```

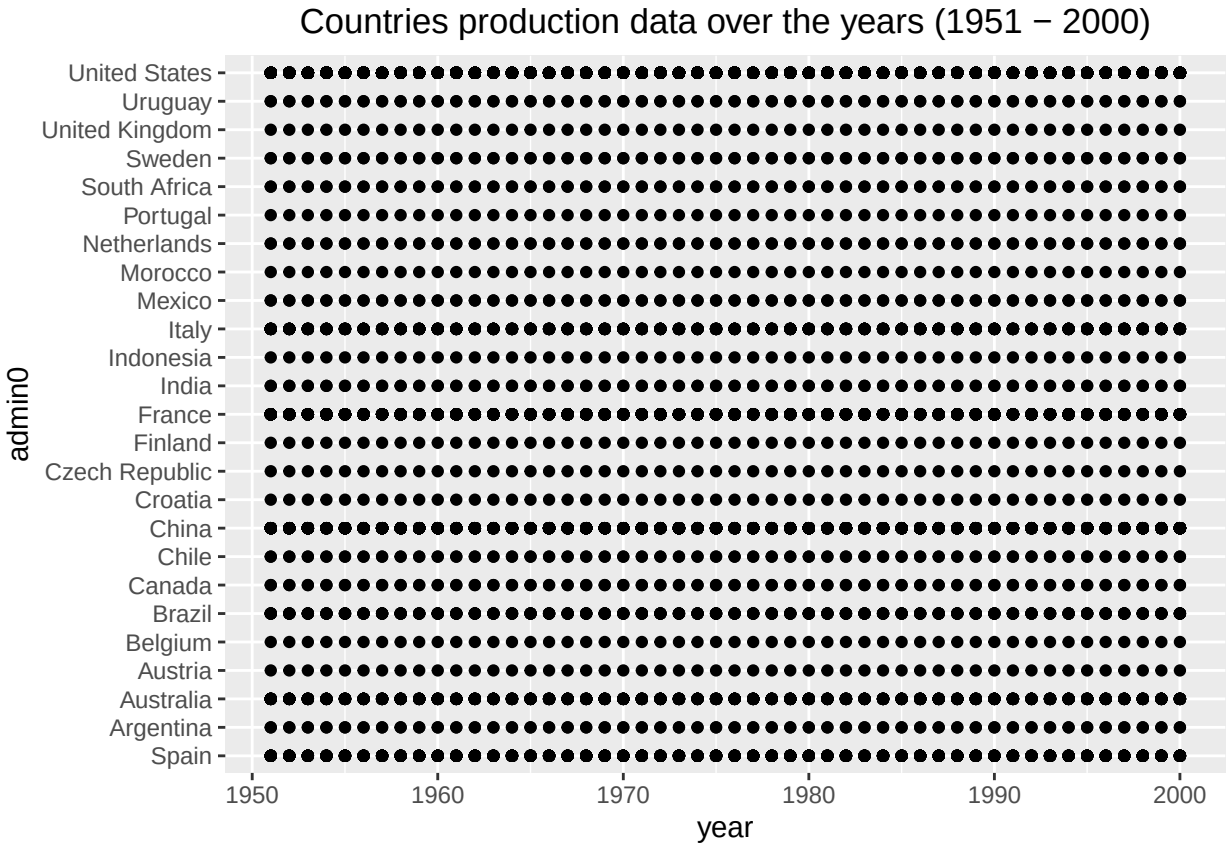
### World Production (1951 - 2000)

```
ggplot(data = Production1950to2000year, mapping = aes(x = year, y = `production (tonnes)`)) +
  geom_point(alpha = 2/10) +
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1) +
  scale_y_continuous(labels = scales::comma) +
  ggtitle("Production over the years (1951 - 2000)")
```



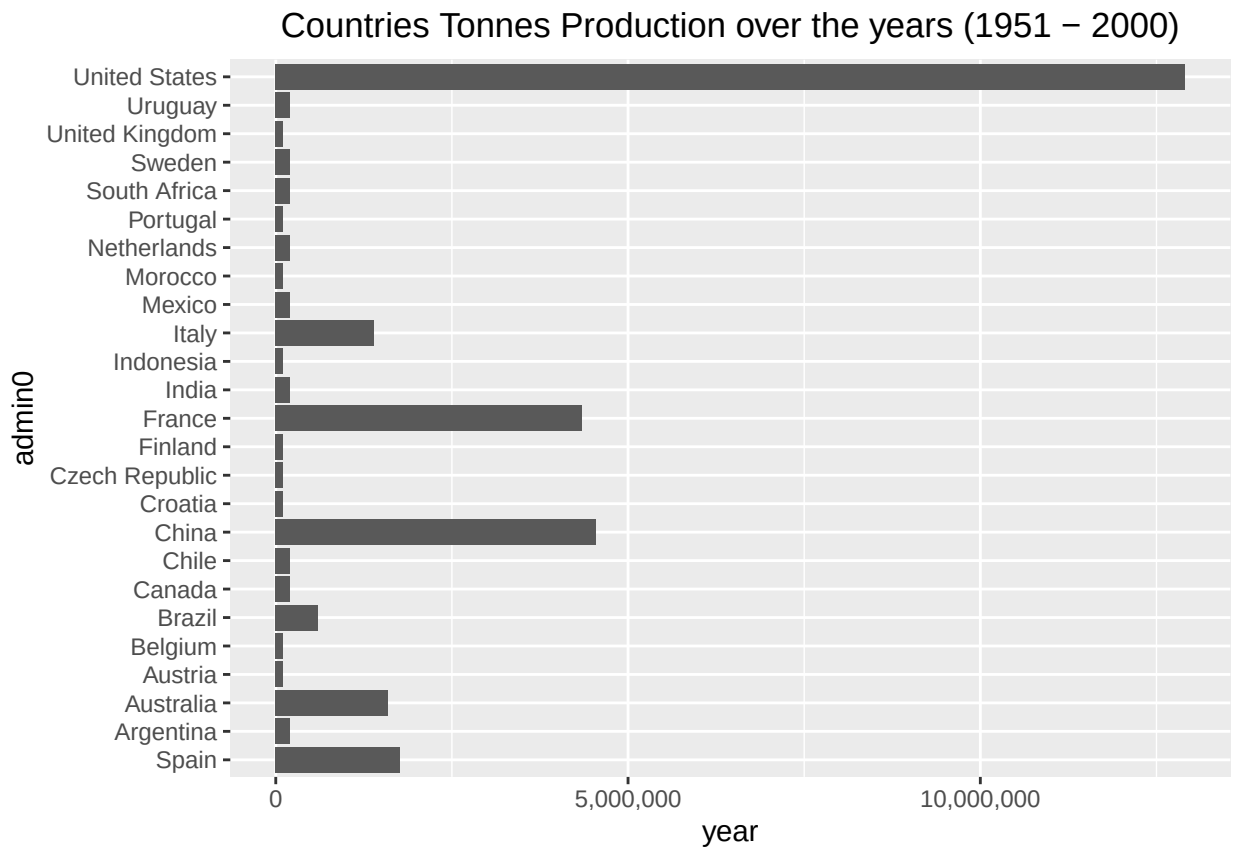

**Countries with more data of Production**

```
ggplot(data = Production1950to2000year, mapping = aes(x = year , y = admin0)) +
  geom_point() +
  ggtitle("Countries production data over the years (1951 - 2000)")
```




**Production of each country**

```
ggplot(data = Production1950to2000year, mapping = aes(x = year , y = admin0)) +
  geom_bar(stat = "identity") +
  scale_x_continuous(labels = scales::comma) +
  ggtitle("Countries Tonnes Production over the years (1951 - 2000)")
```



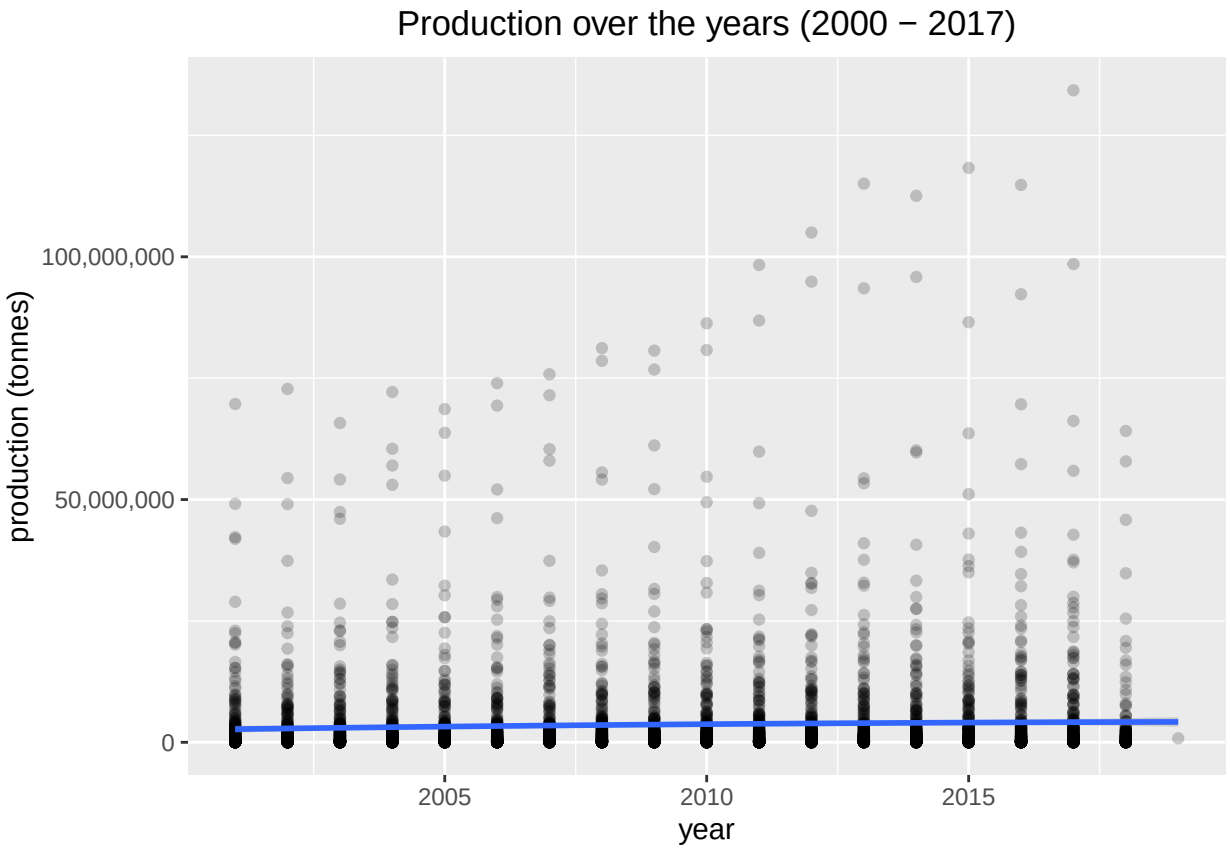

## Section 3 from 2000 - 2017

### To check the Production (2000 - 2017)

```
Production2000to2017year <- filter(crops, year > 2000) # Above the 2000 years
Production2000to2017year <- mutate(Production2000to2017year,
                                   admin0 = fct_reorder(admin0, desc(year) )) # Sorting the data
```

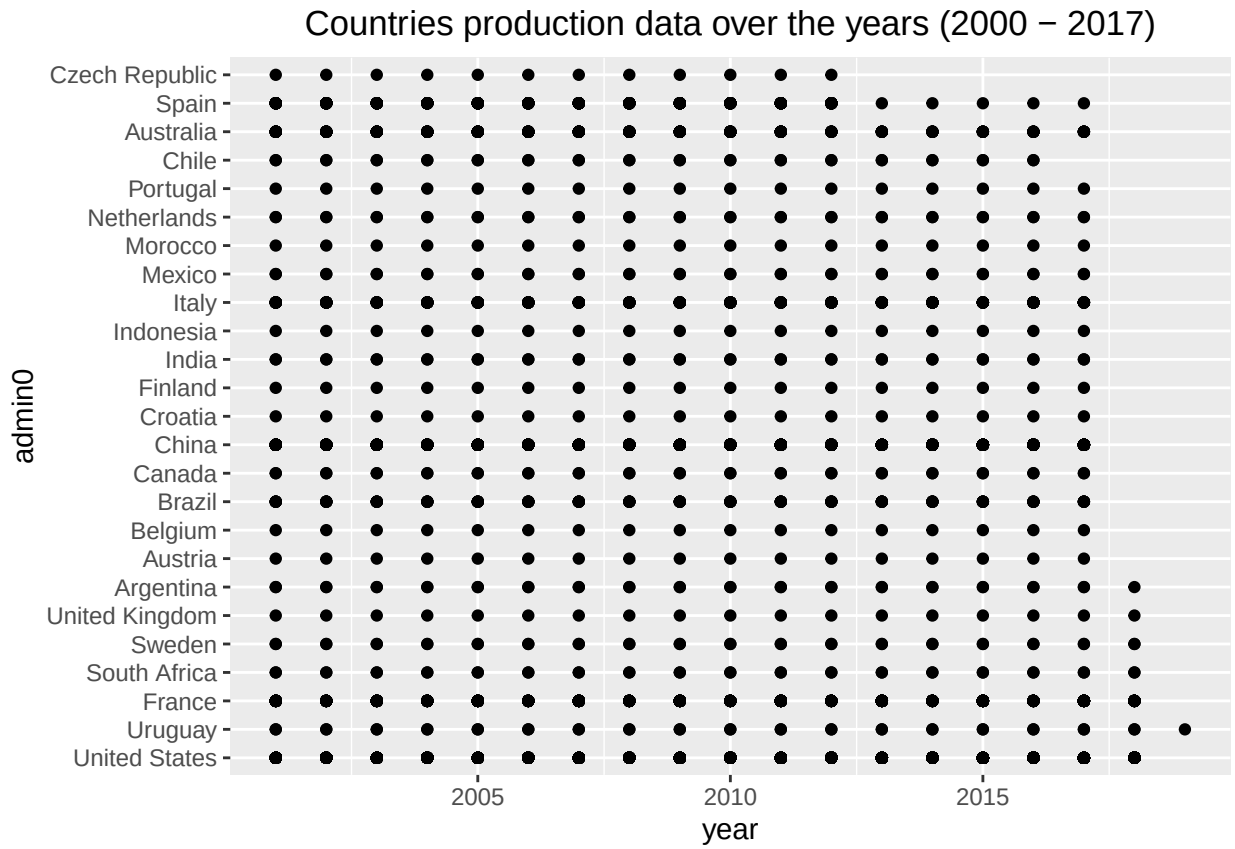
#### World Production(2000 - 2017)

```
ggplot(data = Production2000to2017year, mapping = aes(x = year, y = `production (tonnes)`)) +
  geom_point(alpha = 2/10) +
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1) +
  scale_y_continuous(labels = scales::comma) +
  ggtitle("Production over the years (2000 - 2017)")
```



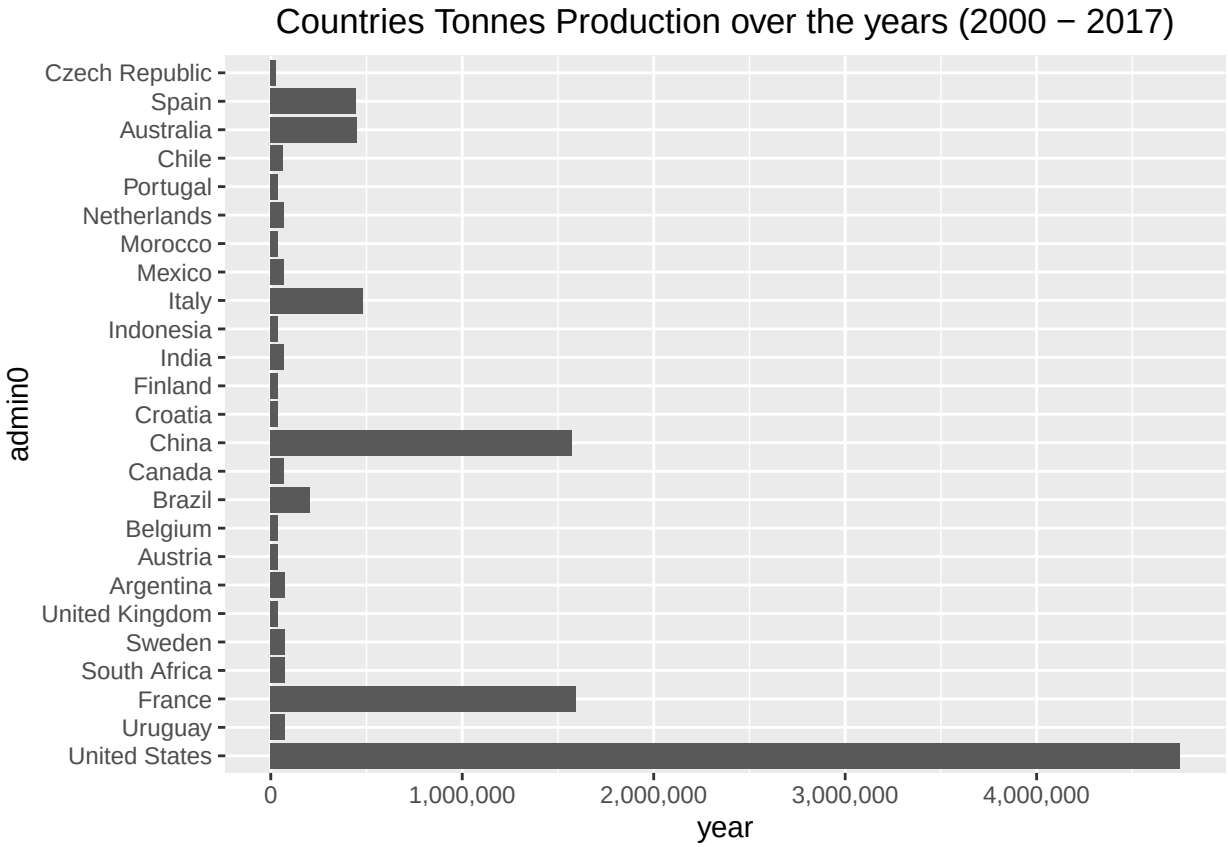

**Countries with more data of Production**

```
ggplot(data = Production2000to2017year, mapping = aes(x = year , y = admin0)) +
  geom_point() +
  ggtitle("Countries production data over the years (2000 - 2017)")
```




**Production of each country**

```
ggplot(data = Production2000to2017year, mapping = aes(x = year , y = admin0)) +
  geom_bar(stat = "identity") +
  scale_x_continuous(labels = scales::comma) +
  ggtitle("Countries Tonnes Production over the years (2000 - 2017)")
```




## References


- Correlation heatmap using ggplot2