Case Study 4 - Global Fishing

Loading Libraries & Data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

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

```
library(readr)
library(skimr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(SimDesign)
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(RSQLite)
library(knitr)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(RColorBrewer)
library(treemap)
library(readxl)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

```
library(viridis)
```

```
## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':
##
##     viridis_pal
```

```
library(grid)
library(lattice)

farmed <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/20
```

```
## Rows: 11657 Columns: 4

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Aquaculture production (metric tons)

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
captured_vs_farmed <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/mas
```

```
## Rows: 14674 Columns: 5
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (3): Year, Aquaculture production (metric tons), Capture fisheries produ...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
captured <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/1
```

```
## Rows: 14516 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Capture fisheries production (metric tons)
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
consumption <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/da
```

```
## Rows: 11028 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (2): Year, Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
stock <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/202
```

```
## Rows: 51 Columns: 5
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (3): Year, Share of fish stocks within biologically sustainable levels (...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
fishery <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2
```

```
## Rows: 61 Columns: 8
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (6): Year, Artisanal (small-scale commercial), Discards, Industrial (lar...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
production <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data
```

```
## Rows: 10326 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Entity, Code
## dbl (8): Year, Commodity Balances - Livestock and Fish Primary Equivalent - ...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Cross checking the "captured" and the "farmed" tables with the "captured_vs_farmed" table

```
#Checking if all "captured" data is found in the captured_vs_farmed table
captured_vs_farmed_filtered <- captured_vs_farmed%>%
  filter(`Capture fisheries production (metric tons)` != "NA")

sum(captured$`Capture fisheries production (metric tons)`)
```

```
## [1] 41264851330
```

```
sum(captured_vs_farmed_filtered$`Capture fisheries production (metric tons)`)
```

```
## [1] 41264851330
```

```
#Checking if all "farmed" data is found in the captured_vs_farmed table
captured_vs_farmed_filtered2 <- captured_vs_farmed%>%
  filter(`Aquaculture production (metric tons)` != "NA")

sum(farmed$`Aquaculture production (metric tons)`)
```

```
## [1] 19842737675
```

```r
sum(captured_vs_farmed_filtered2$`Aquaculture production (metric tons)`)
```

```
## [1] 19842737675
```

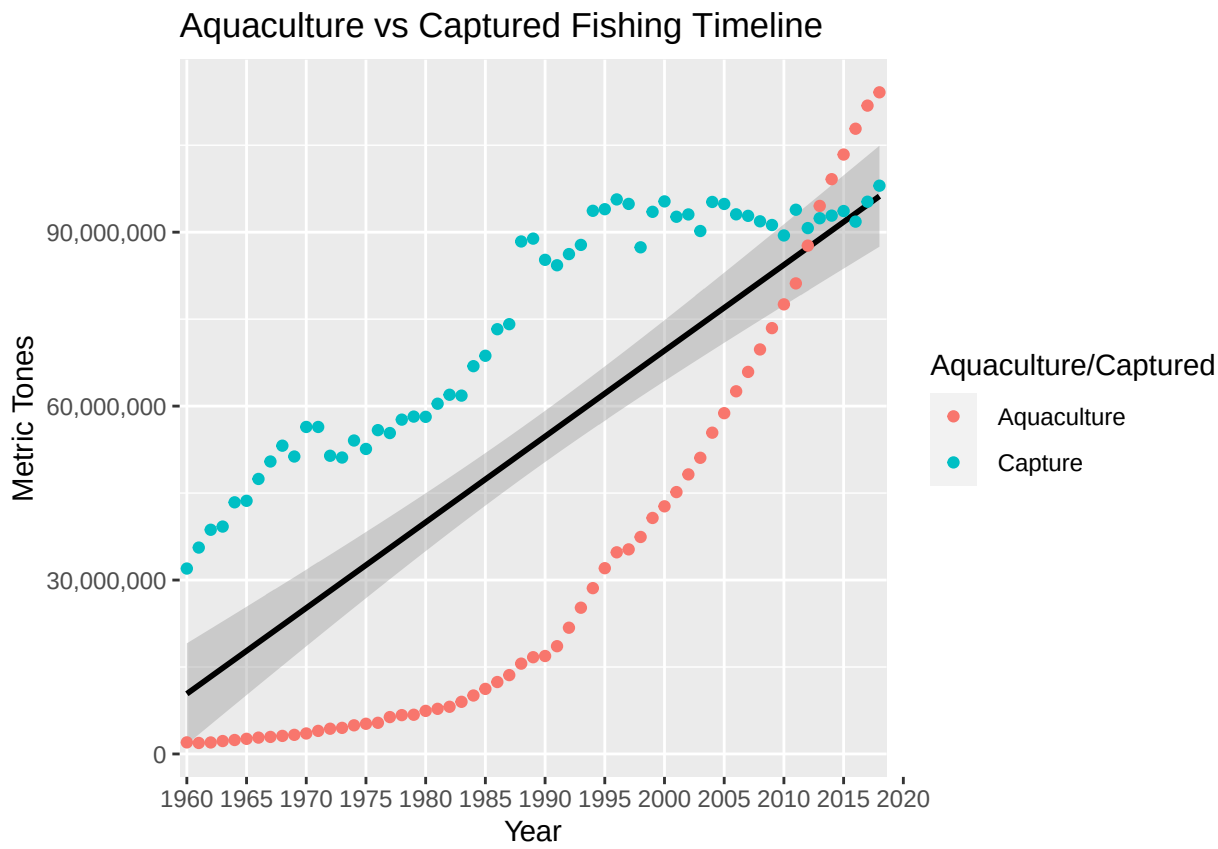
Removing the regions that are groups from the captured_vs_farmed and from consumption

```r
#Gather and exclude all the entities that are regions and not countries
ToExclude <- c("Africa", "Americas", "Africa Eastern and Southern", "Africa Western and Central", "Arab

captured_vs_farmed <- captured_vs_farmed%>%
  filter(!Entity %in% ToExclude)

consumption <- consumption%>%
  filter(!Entity %in% ToExclude)
```

Visualizing the Timeline graph of the captured_vs_farmed

```r
cap_vs_farm_time <- select(captured_vs_farmed, c("Year", "Aquaculture production (metric tons)","Capture
cap_vs_farm_time[is.na(cap_vs_farm_time)] <- 0
cap_vs_farm_time <- cap_vs_farm_time %>%
  pivot_longer(!Year, names_to = "Aquaculture/Captured", values_to = "Metric Tones")
cap_vs_farm_time$`Aquaculture/Captured`[cap_vs_farm_time$`Aquaculture/Captured` == "Aquaculture producti
cap_vs_farm_time$`Aquaculture/Captured`[cap_vs_farm_time$`Aquaculture/Captured` == "Capture fisheries pr

cap_vs_farm_time %>%
  group_by(Year, `Aquaculture/Captured`)%>%
  summarize(Metric_Sum = sum(`Metric Tones`))%>%
  ggplot(aes(x = Year, y = Metric_Sum, colour = `Aquaculture/Captured`)) +
  geom_smooth(method= lm, color = "black") +
  geom_point() +
  scale_x_discrete(limits = c(1960, 1965, 1970,1975,1980,1985,1990,1995,2000,2005,2010,2015,2020)) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Aquaculture vs Captured Fishing Timeline") +
  ylab("Metric Tones")
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.

## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?

## `geom_smooth()` using formula 'y ~ x'
```

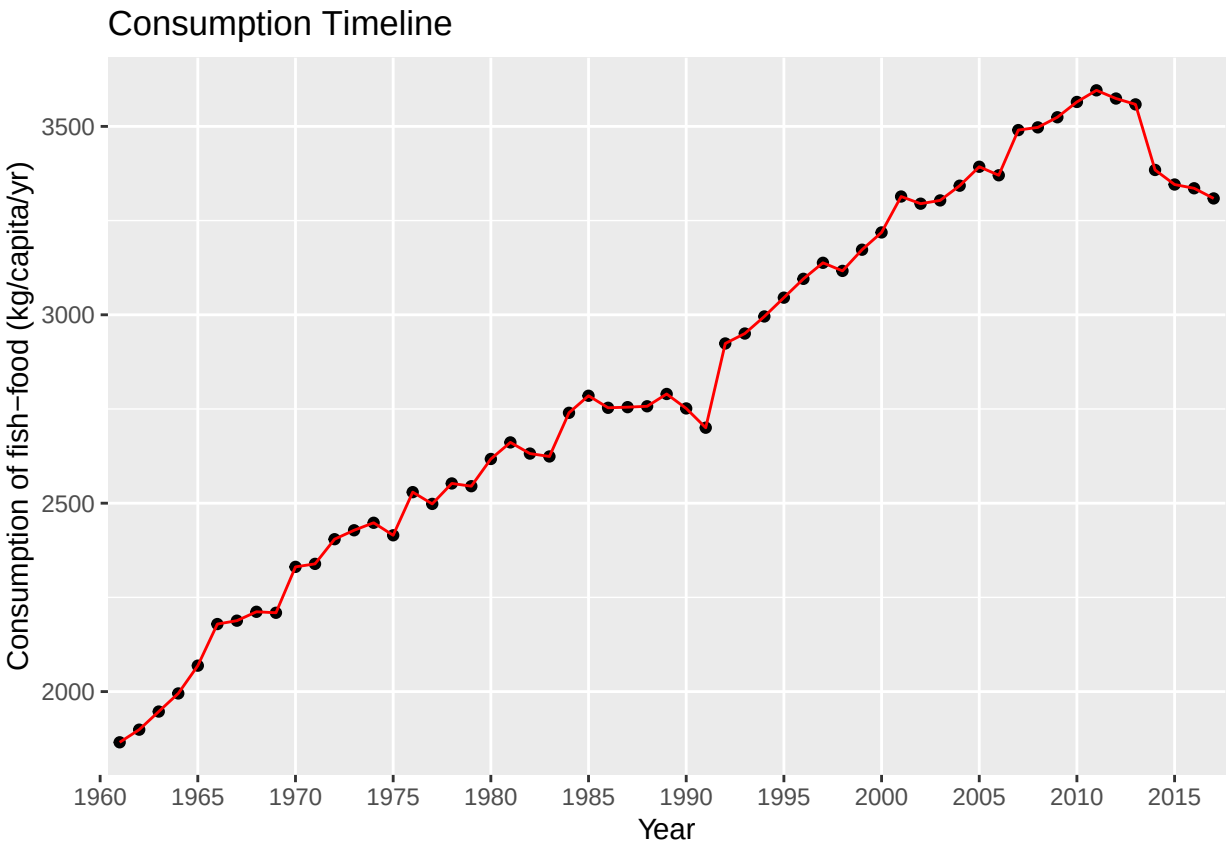


As we see here Aquaculture has evolved throughout the years. It's starting to incline more steeply arou

Consumption of seafood related data

```
#Timeline viz of the total consumption over the years
consumption%>%
  group_by(Year)%>%
  summarize(Cons_sum = sum(`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`))%>%
  ggplot(aes(x = Year, y = Cons_sum)) + geom_point() + geom_line(linetype = 1, color = "red") +
  scale_x_discrete(limits = c(1960, 1965, 1970,1975,1980,1985,1990,1995,2000,2005,2010,2015,2020)) +
  labs(title = "Consumption Timeline") +
  ylab("Consumption of fish-food (kg/capita/yr)")
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```



The consumption of seafood is increasing over time and of course this relates to the worldwide populati

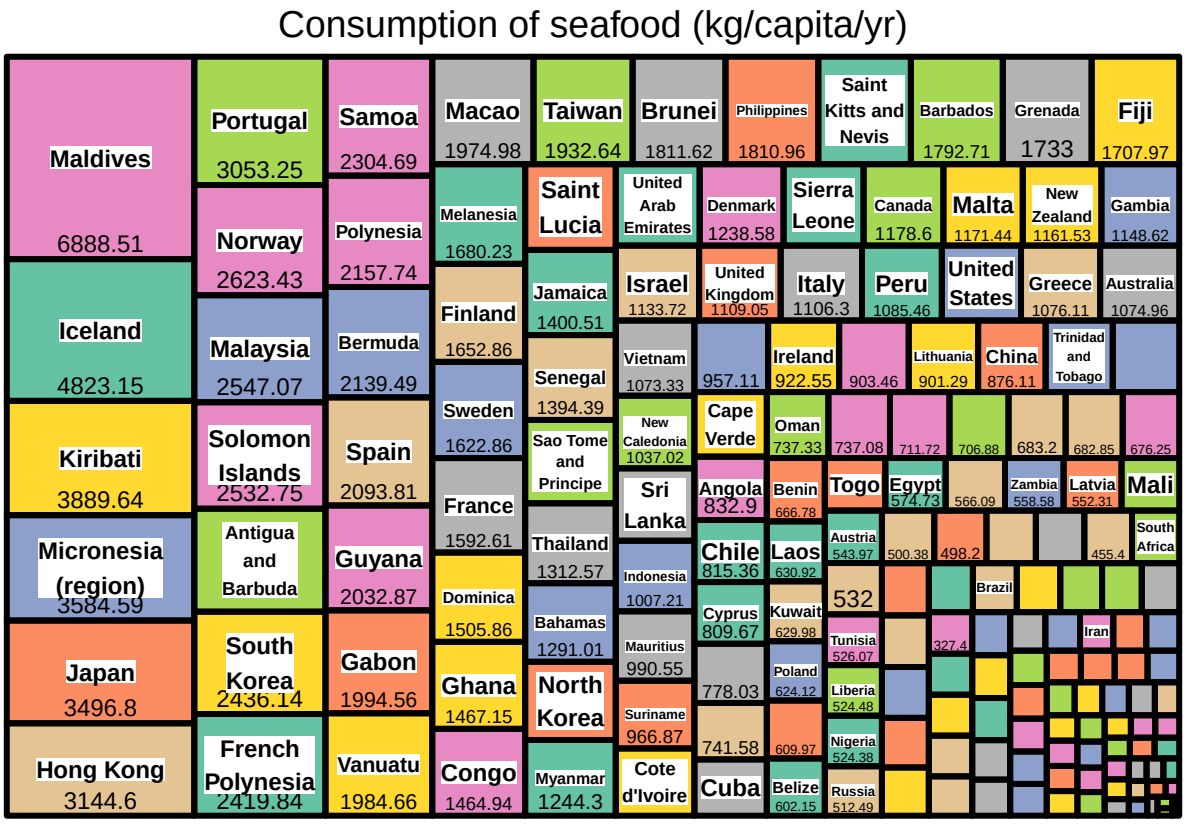
In 2014 though we can see a very sharp decrease in the consumption from 4000-4100 to 3300-3400 kg/capit

Consumption of seafood related data (Full TreeMap)

```
consumption_per_country_all <- consumption%>%
  group_by(Entity)%>%
  summarize(Cons_sum = sum(`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`))%>%
  treemap(index=c("Entity", "Cons_sum"),
            vSize="Cons_sum",
            type="index",
            palette = "Set2",
            bg.labels=c("white"),
            fontsize.labels = c(9,9),
            title = "Consumption of seafood (kg/capita/yr)",
            align.labels=list(
              c("center", "center"),
              c("center", "bottom")
            ),
          )
```

```
## Warning in if (class(try(col2rgb(bg.labels), silent = TRUE)) == "try-error")
```

```
## stop("Invalid bg.labels"): the condition has length > 1 and only the first
## element will be used
```



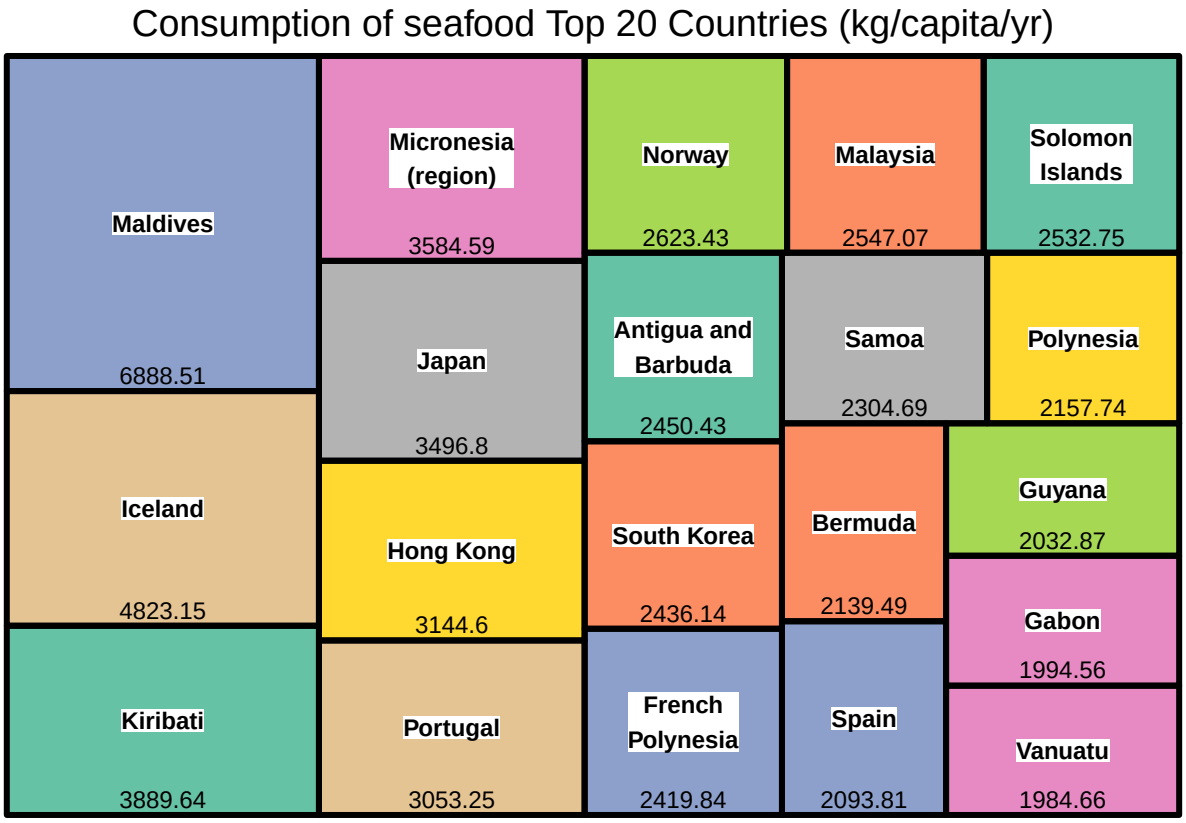

```
The Maldives hold the most consumpton of seafood by 6,888.51 kg/capita/yr followed by Iceland and Kiriba
```

Consumption of seafood related data (Top 20 Countries TreeMap)

```
consumption_per_country_top20Tree <- consumption%>%
  group_by(Entity)%>%
  summarize(Cons_sum = sum(`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`))%>%
  top_n(20)%>%
  treemap(index=c("Entity", "Cons_sum"),
            vSize="Cons_sum",
            type="index",
            palette = "Set2",
            bg.labels=c("white"),
            fontsize.labels = c(9,9),
            title = "Consumption of seafood Top 20 Countries (kg/capita/yr)",
            align.labels=list(
              c("center", "center"),
              c("center", "bottom")
            ),
          )
```

```
## Selecting by Cons_sum
```

```
## Warning in if (class(try(col2rgb(bg.labels), silent = TRUE)) == "try-error")
## stop("Invalid bg.labels"): the condition has length > 1 and only the first
## element will be used
```

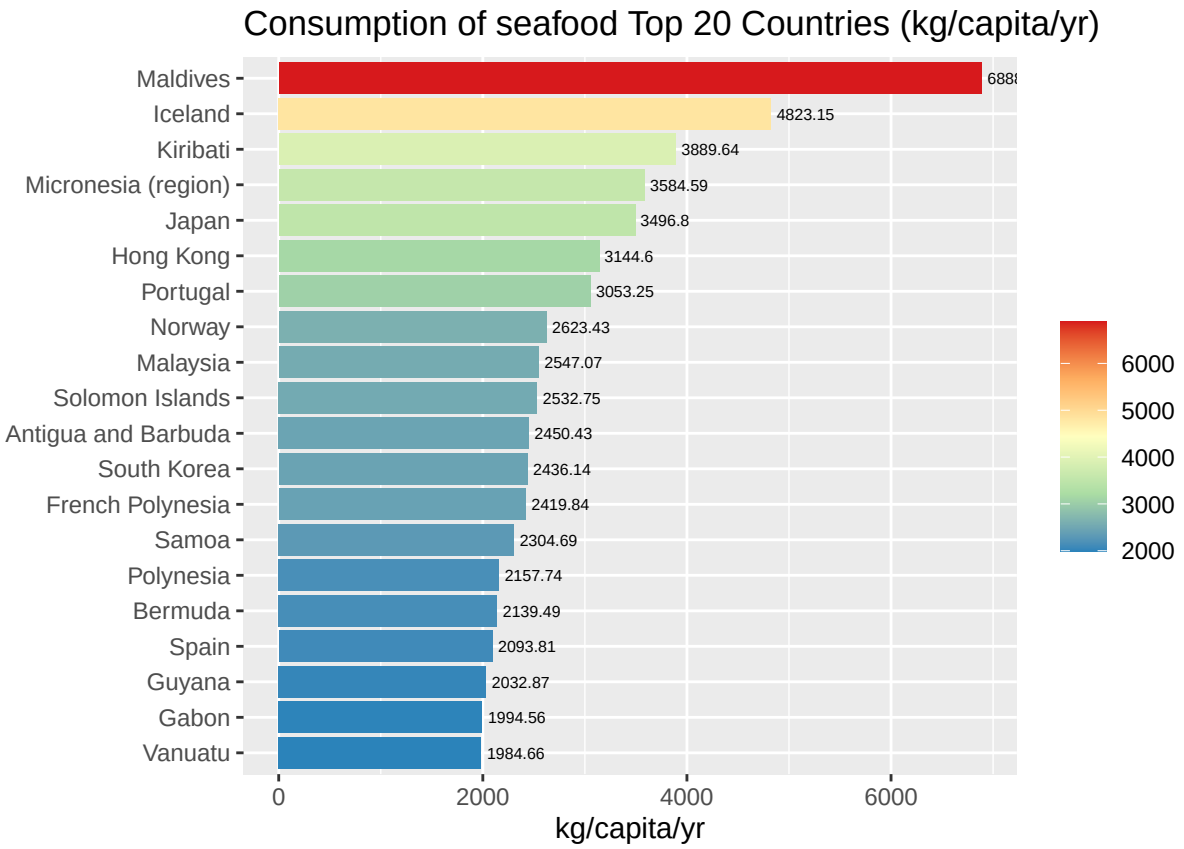


Consumption of seafood related data (BarChart)

```
consumption_per_country_top20Bar <- consumption%>%
  group_by(Entity)%>%
  summarise(Conss_Sum = sum(`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`))%>%
  top_n(20)%>%
  ggplot(aes(x = Conss_Sum,y = reorder(factor(Entity),Conss_Sum), fill = Conss_Sum)) +
  geom_bar(stat = "identity") +
  scale_fill_gradientn(name = '',colours = rev(brewer.pal(5,'Spectral'))) +
  geom_text(aes(label = Conss_Sum), hjust = -0.1, size = 2) +
  labs(title = "Consumption of seafood Top 20 Countries (kg/capita/yr)") +
  xlab("kg/capita/yr") +
  ylab("")
```

```
## Selecting by Conss_Sum
```

```
consumption_per_country_top20Bar
```



Checking the Percentage of the top 20 countries over the total number of consumption

```
Total_consumption <- sum(consumption$`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`)

consumption_per_country_top20 <- consumption%>%
  group_by(Entity)%>%
  summarise(Conss_Sum = sum(`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`))%>%
  top_n(20)
```

```
## Selecting by Conss_Sum
```

```
Total_top20_consumption <- sum(consumption_per_country_top20$Conss_Sum)

Percentage_Total <- Total_top20_consumption/Total_consumption*100
Percentage_Total
```

```
## [1] 36.28955
```

```
#The top 20 countries have %36 of the consumption comparing to the total consumption in the world
```

Tracking the difference of the total production and total consumption

```
#Removing the NA values and Adding a total production column
captured_vs_farmed$`Aquaculture production (metric tons)`[is.na(captured_vs_farmed$`Aquaculture producti
captured_vs_farmed$`Capture fisheries production (metric tons)`[is.na(captured_vs_farmed$`Capture fishe
captured_vs_farmed$Total_Production <- captured_vs_farmed$`Aquaculture production (metric tons)`+captur


#Grouping the captured/farmed table by entity
captured_vs_farmed_group <- captured_vs_farmed%>%
  group_by(Entity)%>%
  summarise(Total_Production = sum(Total_Production))

#Grouping the consumption table by entity
consumption_group <- consumption%>%
  group_by(Entity)%>%
  summarise(Total_Consumption = sum(`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`))

##Converting the Kg to Metric Tonnes
consumption_group$Total_Consumption <- consumption_group$Total_Consumption/1000

##In order to be able to get and analyze the difference between the production and the consumption we n

#Importing the Population data
population_data <- read_excel("~/Desktop/Data Analysis Case Studies/Case Study 4/API_SP.POP.TOTL_DS2_en
```

TOTALS WITH NO TIMLINE

```
## New names:
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
## * `` -> ...7
## * ...
```

```
#Manipulation of the population data
population_data <- population_data[-c(1,2),]
names(population_data) <- population_data[1,]
```

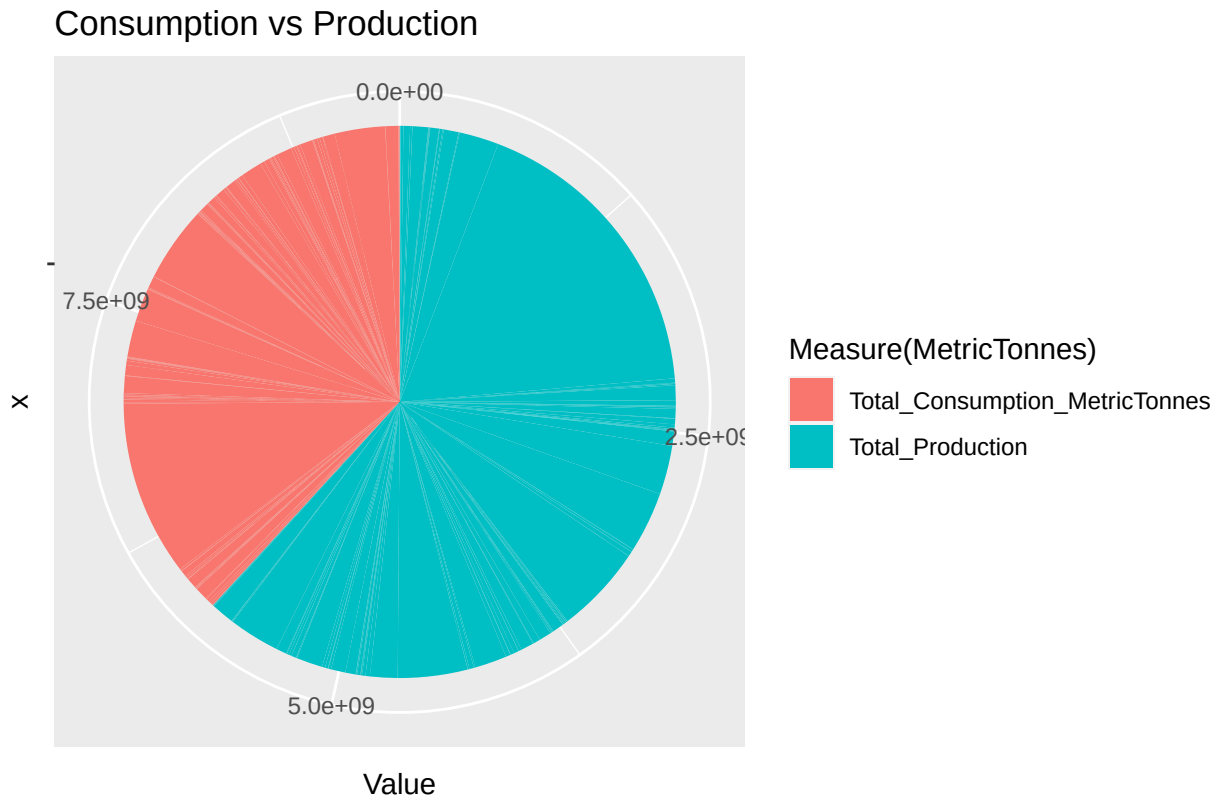
```
## Warning: The `value` argument of `names<-` must be a character vector as of
## tibble 3.0.0.
```

```
population_data <- population_data[-c(1),]
population_data <- population_data[-c(3,4)]
population_data <- population_data%>%
  pivot_longer(!`Country Name` & !`Country Code`,names_to = "Year", values_to = "Population_count")
```

```r
population_data$Population_count <- as.numeric(population_data$Population_count)

#Group the population table by Country Name
population_data <- population_data %>%
  rename(Entity = `Country Name`)

population_data_Group <- population_data%>%
  group_by(Entity)%>%
  summarise(Average_Population = mean(Population_count))

#Merging the Population_data with the Consumption_group
Consumption_population_data <- merge(x = population_data_Group, y = consumption_group, by = "Entity" )
Consumption_population_data$Total_Consumption_MetricTonnes <- Consumption_population_data$Total_Consump
Consumption_population_data <-Consumption_population_data[-c(2,3)]

#Merging the Consumption data with the captured/Farmed data ("Production" data)
Cons_Prod_Diff <- merge(x = Consumption_population_data, y = captured_vs_farmed_group, by = "Entity")
Cons_Prod_Diff$Total_Production <- round(Cons_Prod_Diff$Total_Production, digits = 0)

#Creating a difference column which measures the difference between the production and the consumption
Cons_Prod_Diff$Difference <- Cons_Prod_Diff$Total_Production-Cons_Prod_Diff$Total_Consumption_MetricTonn

#Visualizing and Analyzing the Difference to see if there is a surplus or a deficit in the seafood prod
Cons_Prod_Diff <- Cons_Prod_Diff%>%
  pivot_longer(!Entity, names_to = "Measure(MetricTonnes)", values_to = "Value")
Cons_Prod_Diff %>%
  filter(Cons_Prod_Diff$`Measure(MetricTonnes)` != "Difference")%>%
  ggplot(aes(x = "", y = Value, fill = `Measure(MetricTonnes)`, fill = `Measure(MetricTonnes)`)) +
  geom_bar(stat = "identity", width = 6) +
  coord_polar("y", start = 0) +
  labs(title = "Consumption vs Production")
```

```
## Warning: Duplicated aesthetics after name standardisation: fill
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```



This shows clearly that there is a good amount of surpluss between what is produced vs what is consumed

```
#Grouping the captured/farmed table by entity and year
captured_vs_farmed_TimeGroup <- captured_vs_farmed%>%
  group_by(Entity,Year)%>%
  summarise(Total_Production = sum(Total_Production))
```

TIMELINE DIFFERENCE OVER THE YEARS

```
## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.
```

```
#Grouping the consumption table by entity and year
consumption_TimeGroup <- consumption%>%
  group_by(Entity,Year)%>%
  summarise(Total_Consumption = sum(`Fish, Seafood- Food supply quantity (kg/capita/yr) (FAO, 2020)`))
```

```
## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.
```

```
consumption_TimeGroup$Total_Consumption <- consumption_TimeGroup$Total_Consumption/1000

#Group the population table by Country Name
```

```r
population_data_TimeGroup <- population_data[-c(2)]

#Merging the Population_data with the Consumption_group
Consumption_population_TimeData <- merge(x = population_data_TimeGroup, y = consumption_TimeGroup, by =
Consumption_population_TimeData <- Consumption_population_TimeData%>%
  filter(Total_Consumption != 0)
Consumption_population_TimeData$Total_Consumption_MetricTonnes <- Consumption_population_TimeData$Total_
Consumption_population_TimeData <- Consumption_population_TimeData[-c(4)]

#Merging the Consumption data with the captured/Farmed data ("Production" data)
Cons_Prod_TimeDiff <- merge(x = Consumption_population_TimeData, y = captured_vs_farmed_TimeGroup, by =
Cons_Prod_TimeDiff$Total_Production <- round(Cons_Prod_TimeDiff$Total_Production, digits = 0)
Cons_Prod_TimeDiff$Total_Consumption_MetricTonnes <- round(Cons_Prod_TimeDiff$Total_Consumption_MetricT

#Creating a difference column which measures the difference between the production and the consumption
Cons_Prod_TimeDiff$Difference <- Cons_Prod_TimeDiff$Total_Production-Cons_Prod_TimeDiff$Total_Consumptio

#Converting the table into a long structure
Cons_Prod_TimeDiff <- Cons_Prod_TimeDiff%>%
  pivot_longer(!Entity & !Year, names_to = "Description", values_to = "Value")

#Visualizing and Analyzing the Difference to see if there is a surplus or a deficit in the seafood prod
Cons_Prod_TimeDiff$Description[Cons_Prod_TimeDiff$Description == "Total_Consumption_MetricTonnes"] <- "

Cons_Prod_TimeDiff$Year <- as.character(Cons_Prod_TimeDiff$Year)
p1 <- Cons_Prod_TimeDiff %>%
  filter(Description != "Population_count")%>%
  group_by(Year, Description)%>%
  summarise(Metric_Tonnes = sum(Value))%>%
  ggplot(aes(x =Year, y = Metric_Tonnes, color = Description, group = Description)) +
  geom_line() +
  geom_point() +
  scale_y_discrete(limits =  c(25000000, 50000000, 75000000, 100000000,125000000,150000000, 175000000))
  scale_x_discrete(limits = c("1961", "1965", "1970", "1975", "1980", "1985", "1990", "1995", "2000", "
  scale_y_discrete(limits =  c(25000000, 50000000, 75000000, 100000000,125000000,150000000, 175000000),
  labs(title = "Difference Between Consumption & Production") +
  ylab("Metric Tones")
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.

## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?

## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?

## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

```r
p2 <- Cons_Prod_TimeDiff %>%
  filter(Description == "Population_count")%>%
```

```
  group_by(Year, Description)%>%
  summarise(Count = sum(Value))%>%
  ggplot(aes(x =Year, y = Count, color = Description, group = Description)) +
  geom_line() +
  geom_point() +
  scale_x_discrete(limits = c("1961", "1965", "1970", "1975", "1980", "1985", "1990", "1995", "2000", "
  labs(title = "Population Timeline") +
  scale_y_continuous(labels = scales::comma)
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```

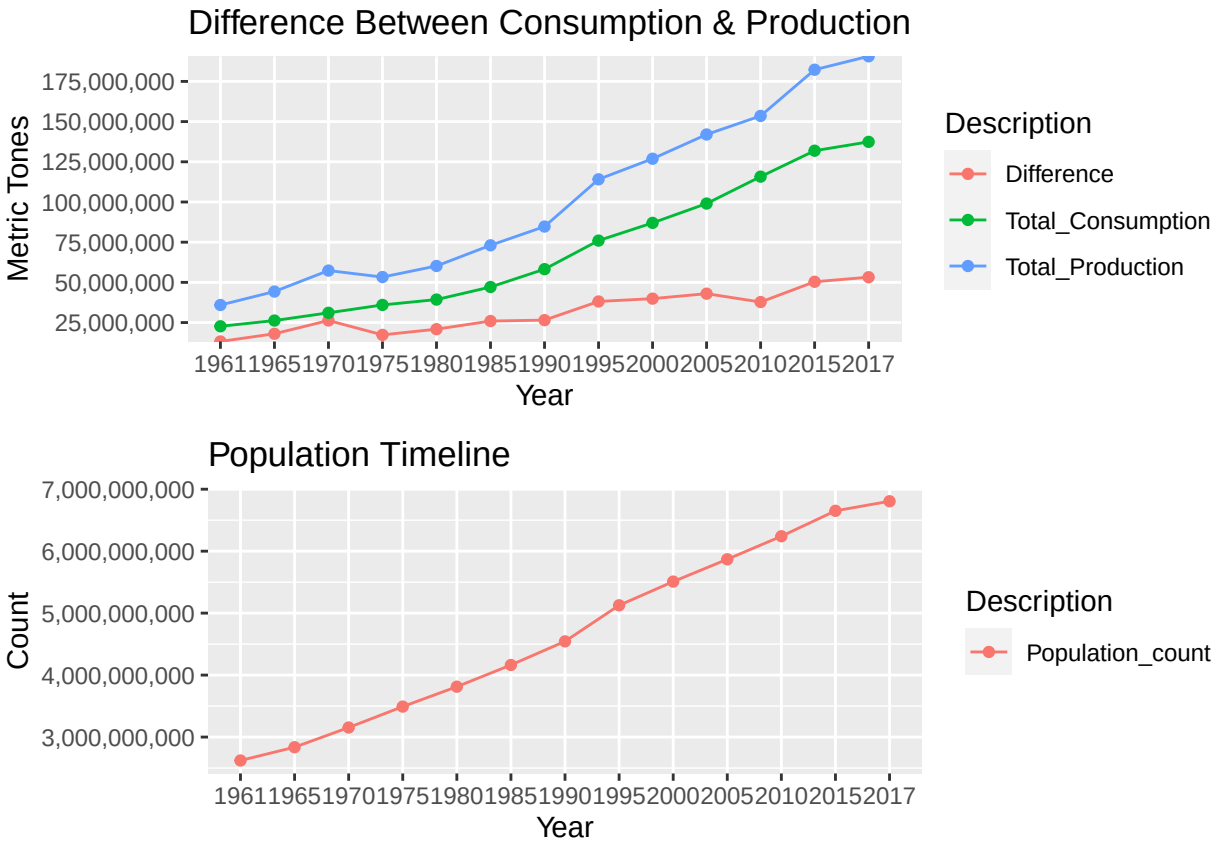
```
grid.arrange(p1,p2, nrow = 2, ncol = 1)
```

```
## Warning: Removed 132 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 132 rows containing missing values (geom_point).
```

```
## Warning: Removed 44 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 44 rows containing missing values (geom_point).
```



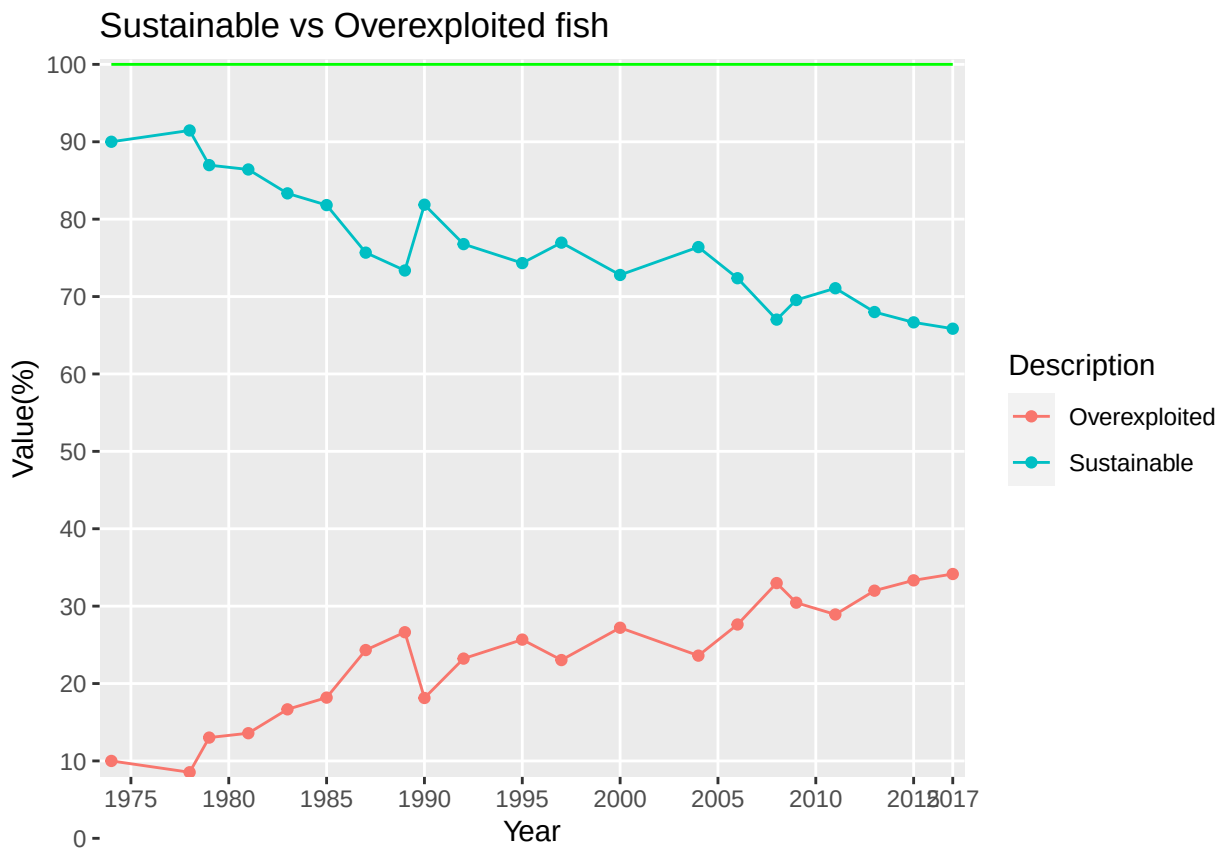
All the numbers in the above graphs show a significant increase from the 1960s till 2017, by all we are
It seems that as the population increases along with the consumption, naturally, of seafood related proc

Visualizing the Sustainable vs Overexploited fish throughout the years

```
stock_Pivoted <- select(stock,c(!"Code"))
stock_Pivoted <- stock_Pivoted%>%
  rename(Sustainable = `Share of fish stocks within biologically sustainable levels (FAO, 2020)`) %>%
  rename(Overexploited = `Share of fish stocks that are overexploited`)

stock_Pivoted <- pivot_longer(stock_Pivoted, !`Year` & !`Entity`, names_to = 'Description', values_to =



p3 <- stock_Pivoted %>%
  filter(Entity == "World")%>%
  group_by(Year, Description)%>%
  summarize(`Value(%)` = sum(Value))%>%
  ggplot(aes(x = Year, y = `Value(%)`, color = Description)) +
  geom_line(linetype = 1) +
  #geom_smooth(method= lm, color = "black") +
  geom_point() +
  stat_summary(fun.y = sum, na.rm = TRUE, group = 3, color = 'green', geom ='line' ) +
  scale_x_discrete(limits = c(1975,1980,1985,1990,1995,2000,2005,2010,2015,2017)) +
  scale_y_discrete(limits = c(0,10,20,30,40,50,60,70,80,90,100)) +
  theme(legend.position = "right",legend.direction = "vertical") +
  labs(title = "Sustainable vs Overexploited fish")
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.

## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?

## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

```
p3
```



Starting from 1978 the sustainable Fish % is on almost a constant decrease starting from %91.46 till it

Visualizing the type of fisheries throughout the years

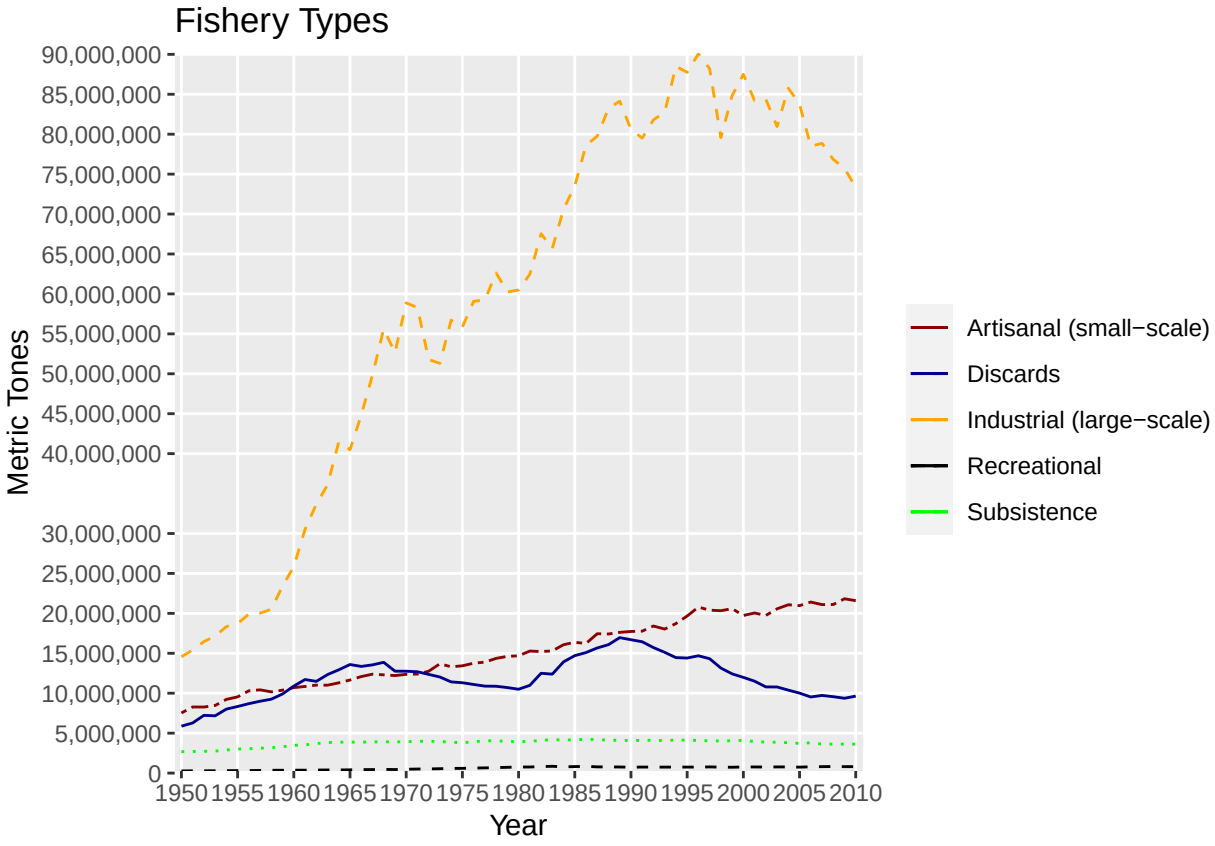
```
fishery <- fishery %>%
  rename(`Artisanal (small-scale)` = "Artisanal (small-scale commercial)") %>%
  rename(`Industrial (large-scale)` = "Industrial (large-scale commercial)")


p5 <- fishery%>%
  ggplot(aes(x = Year)) +
  geom_line(aes(y = fishery$`Artisanal (small-scale)`, color = "Artisanal (small-scale)"), linetype = "
  geom_line(aes(y = fishery$Discards, color = "Discards"), linetype = "solid") +
  geom_line(aes(y = fishery$`Industrial (large-scale)`, color = "Industrial (large-scale)"), linetype =
  geom_line(aes(y = fishery$Recreational, color = "Recreational"), linetype = "dashed") +
  geom_line(aes(y = fishery$Subsistence, color = "Subsistence"), linetype = "dotted") +
  scale_color_manual(values = c(
    "Artisanal (small-scale)" = 'darkred',
    "Discards" = "darkblue",
    "Industrial (large-scale)" = "orange",
    "Recreational" = "black",
    "Subsistence" = "green"
  )) +
  scale_x_discrete(limits = c(1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2
```

```
  scale_y_discrete(limits = c(0, 5000000, 10000000,15000000,20000000,25000000,30000000,40000000,45000000
  #theme(legend.title = element_blank(),legend.position = "bottom",legend.direction = "horizontal") +
  theme(legend.title = element_blank(),legend.position = "right",legend.direction = "vertical") +
  labs(title = "Fishery Types",x = "Year", y = "Metric Tones")
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean 'limits = factor(...)' or 'scale_*_continuous()'?
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean 'limits = factor(...)' or 'scale_*_continuous()'?
```

```
p5
```



The graph is showing us here that the industrial (large scale) fisheries, even though are the largest, a

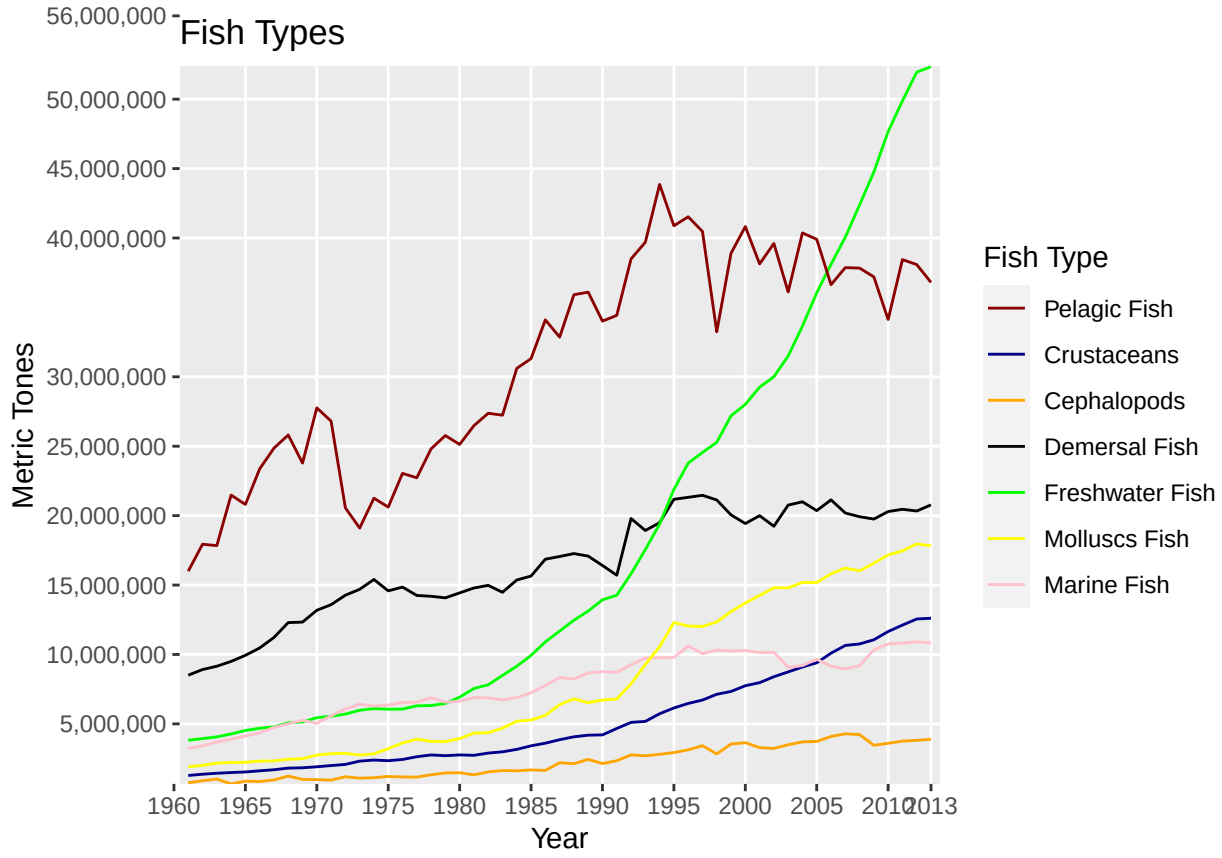
Visualizing the types of fish production over the years

Fish type Examples & Definition:
- Pelagic Fish: Fish that inhabit not near the bottom or the shore of coasts, open oceans, and lakes. E
- Crustaceans: Crustaceans form a large, diverse arthropod taxon which includes such animals as crabs,
- Cephalopods: Are members of a class of marine animals that includes octopuses, squid, cuttlefish and
- Demersal Fish: Also known as groundfish, live and feed on or near the bottom of seas or lakes. They o
- Freshwater fish: Are those that spend some or all of their lives in fresh water, such as rivers and l
- Molluscs Fish: Most molluscs have a hinged two-part shell and include clams, mussels, oysters, and sc
- Marine Fish: Saltwater fish, also called marine fish, are fish that live in ocean water. Saltwater fi

```r
# Removing the Regions and World from Data to avoid double counting
production <- production%>%
  filter(!Entity %in% ToExclude)
# Renaming column names into proper headers
production <- production%>%
  rename("Pelagic Fish" = `Commodity Balances - Livestock and Fish Primary Equivalent - Pelagic Fish - 2
  rename("Crustaceans" = `Commodity Balances - Livestock and Fish Primary Equivalent - Crustaceans - 27
  rename("Cephalopods" = `Commodity Balances - Livestock and Fish Primary Equivalent - Cephalopods - 27
  rename("Demersal Fish" = `Commodity Balances - Livestock and Fish Primary Equivalent - Demersal Fish -
  rename("Freshwater Fish" = `Commodity Balances - Livestock and Fish Primary Equivalent - Freshwater Fi
  rename("Molluscs Fish" = `Commodity Balances - Livestock and Fish Primary Equivalent - Molluscs, Other
  rename("Marine Fish" = `Commodity Balances - Livestock and Fish Primary Equivalent - Marine Fish, Oth


# Visualizing the different types per year
production_year <- production%>%
  pivot_longer(!`Entity` & !`Code` & !`Year`, names_to = "Fish Type", values_to = "Values (Metric Tones)


production_year$`Values (Metric Tones)` <- replace_na(production_year$`Values (Metric Tones)`,0)

p6 <- production_year%>%
  #filter(Entity == "World") %>%
  group_by(Year, `Fish Type`)%>%
  summarize(Value_Sum = sum(`Values (Metric Tones)`))%>%
  ggplot(aes(x = Year, y = Value_Sum, color = `Fish Type`, group = `Fish Type`)) +
  #geom_point() +
  geom_line(linetype = 1) +
  scale_color_manual(values = c(
    "Pelagic Fish" = 'darkred',
    "Crustaceans" = "darkblue",
    "Cephalopods" = "orange",
    "Demersal Fish" = "black",
    "Freshwater Fish" = "green",
    "Molluscs Fish" = "yellow",
    "Marine Fish" = "pink")) +
  scale_x_discrete(limits = c(1960, 1965, 1970, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010,2013)) +
  scale_y_discrete(limits = c(5000000, 10000000,15000000,20000000,25000000,30000000,40000000,45000000,50
  theme(legend.position = "right",legend.direction = "vertical") +
  labs(title = "Fish Types",x = "Year", y = "Metric Tones")
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.

## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?

## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

```r
p6
```



This graph shows us clearly that Freshwater Fish roduction has become a long way from being at medium pr
This concludes that people are aiming more to produce either natuarally or by aquaculture more Freshwat

Visualizing the types of fish production per country (Top 20)

```
# Sorting the top 20 Countries according to their Production
production_year_Top20_Entities <- production_year%>%
  group_by(Entity)%>%
  summarise(Fish_Sum = sum(`Values (Metric Tones)`))%>%
  top_n(20)
```

```
## Selecting by Fish_Sum
```

```
# Filtering out all other countries
production_year_Top20 <- production_year
production_year_Top20 <- production_year_Top20%>%
  filter(production_year_Top20$Entity %in% production_year_Top20_Entities$Entity)

# Scaling the numbers in millions for the visuals to be neater
production_year_Top20$`Values (Metric Tones)` <- production_year_Top20$`Values (Metric Tones)`/1000000

#Visualization grid
p7 <- production_year_Top20%>%
```

```
  group_by(Entity, Year)%>%
  summarize(Value_Sum = sum(`Values (Metric Tones)`))%>%
  ggplot(aes(x = Year, y = Value_Sum, color = Entity, group = reorder(factor(Entity),Value_Sum), fill =
  geom_area(color = "black") +
  scale_fill_viridis(option = "turbo", discrete = "T") +
  scale_x_discrete(limits = c(1960, 1965, 1970, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010,2013)) +
  scale_y_discrete(limits = c(0,25,50,75,100,125)) +
  labs(x = element_blank(), y = element_blank()) +
  theme(
    legend.position="bottom",
    legend.direction = "horizontal",
    legend.title = element_blank(),
    legend.key.size = unit(0.2, 'cm'),
    legend.key.height = unit(0.2, 'cm'),
    legend.key.width = unit(0.2, 'cm'),
    legend.text = element_text(size=7))
```

```
## `summarise()` has grouped output by 'Entity'. You can override using the `.groups` argument.
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

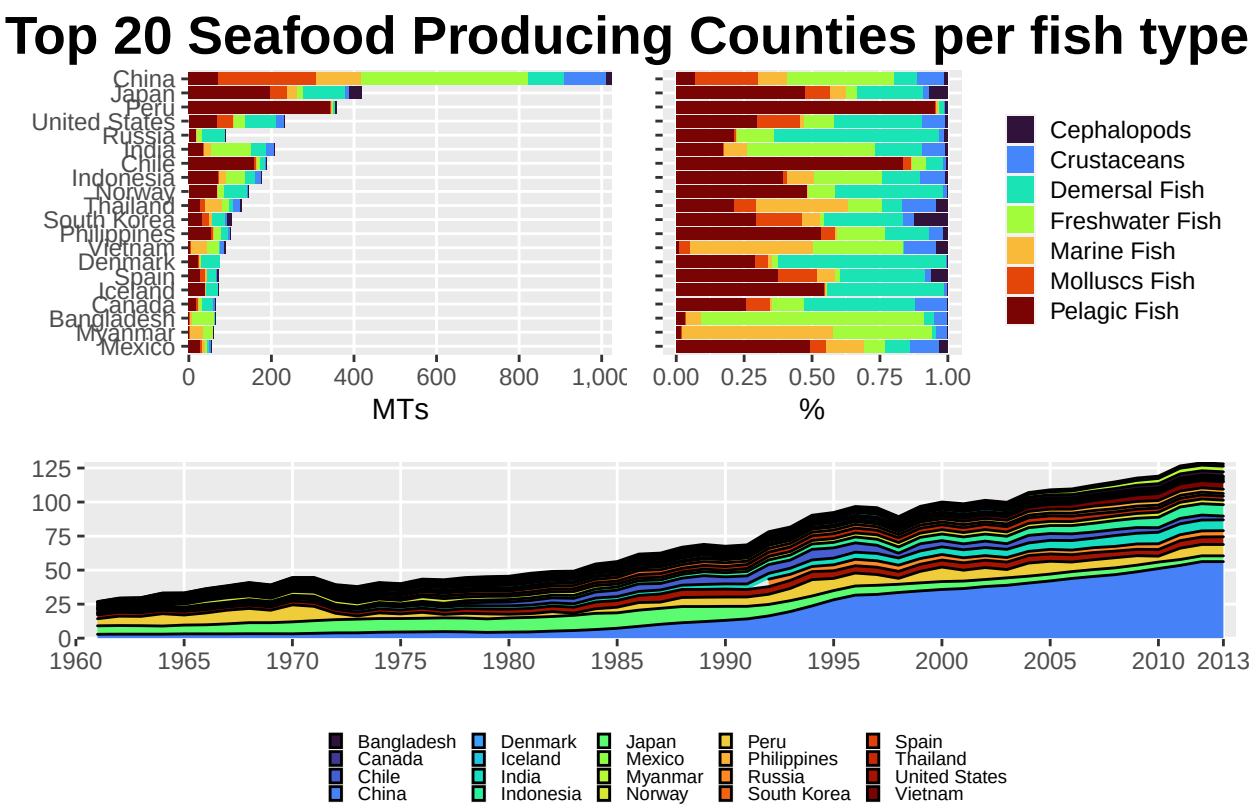
```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

```
p8 <- production_year_Top20 %>%
  ggplot(aes(x = reorder(factor(Entity),`Values (Metric Tones)`), y = `Values (Metric Tones)`, fill = `
  geom_bar( stat="identity") +
  coord_flip() +
  scale_fill_viridis(option = "turbo",discrete = T) +
  labs(x = element_blank(), y = "MTs") +
  scale_y_discrete(limits = c(0,200,400,600,800,1000),labels = scales::comma) +
  theme(legend.position="none")
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

```
p9 <- production_year_Top20 %>%
  ggplot(aes(x = reorder(factor(Entity),`Values (Metric Tones)`), y = `Values (Metric Tones)`, fill = `
  geom_bar(position="Fill", stat="identity") +
  coord_flip() +
  scale_fill_viridis(option = "turbo",discrete = T) +
  labs(x = element_blank(), y = "%") +
  theme(
    legend.position="right",
    legend.direction = "vertical",
    legend.title = element_blank(),
    legend.key.size = unit(0.4, 'cm'),
    legend.key.height = unit(0.4, 'cm'),
    legend.key.width = unit(0.4, 'cm'),
    legend.text = element_text(size=9),
    axis.text.y = element_blank())
```

```
grid.arrange(arrangeGrob(p8,p9, ncol=2, heights=unit(c(2,2), c("in", "mm"))), p7,
             top = textGrob("Top 20 Seafood Producing Counties per fish type", gp=gpar(fontsize=20,font=
             bottom = textGrob("All Values Are in Metric Tones (Millions)", gp=gpar(fontsize=10,font=4))
```



The obvious country that is leading the seafood market in production is China with more than 1 Billion m