Data Visualization Assignment 1

Hamed Ahmadinia

16/11/2020

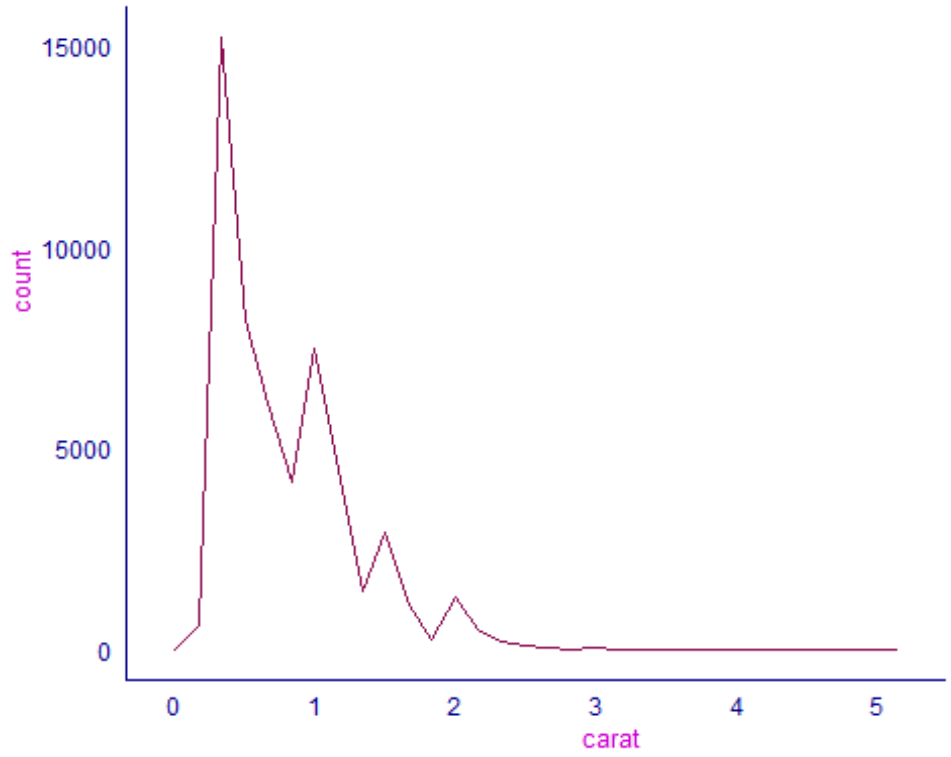
```
#Packages and libraries ###################################################
if(!'pacman'%in%installed.packages()){install.packages('pacman')}
pacman::p_load( ggplot2, ggthemes,nlme, gapminder, gganimate, ggExtra, psych,
reshape2, dplyr, nycflights13, ggcorrplot, waffle, tidyr, scales, ggalt,
data.table, extrafont, lubridate, DT, grid, gridExtra, prettydoc, devtools,
tidyverse, ggdark, here, gifski, forcats, tufte, colorspace, viridisLite,
formatR, DiagrammeR, xaringan, ggridges, GGally,corrplot, ggplot2movies,
ggpointdensity, rstat, ggstatsplot, ggbeeswarm,devtools)

devtools::install_github('Ather-Energy/ggTimeSeries')

devtools::install_github('erocoar/gghalves')

#Connect with the libraries
library(ggplot2)
library(ggthemes)
library(nlme)
library(gganimate)
library(gapminder)
library(ggExtra)
library(psych)
library(reshape2)
library(dplyr)
library(nycflights13)
library(ggcorrplot)
library(waffle)
library(tidyr)
library(scales)
library(ggalt)
library(data.table)
library(extrafont)
library(lubridate)
library(DT)
library(grid)
library(gridExtra)
library(prettydoc)
library(devtools)
library(tidyverse)
library(ggdark)
library(here)
library(gifski)
library(forcats)
```

```
library(tufte)
library(colorspace)
library(viridisLite)
library(formatR)
library(DiagrammeR)
library(xaringan)
library(ggridges)
library(GGally)
library(ggplot2movies)
library(corrplot)
library(ggpointdensity)
library(ggstatsplot)
library(ggTimeSeries)
library(ggbeeswarm)
library(gghalves)
```

#Exercise 01: Creating my own theme ######################################

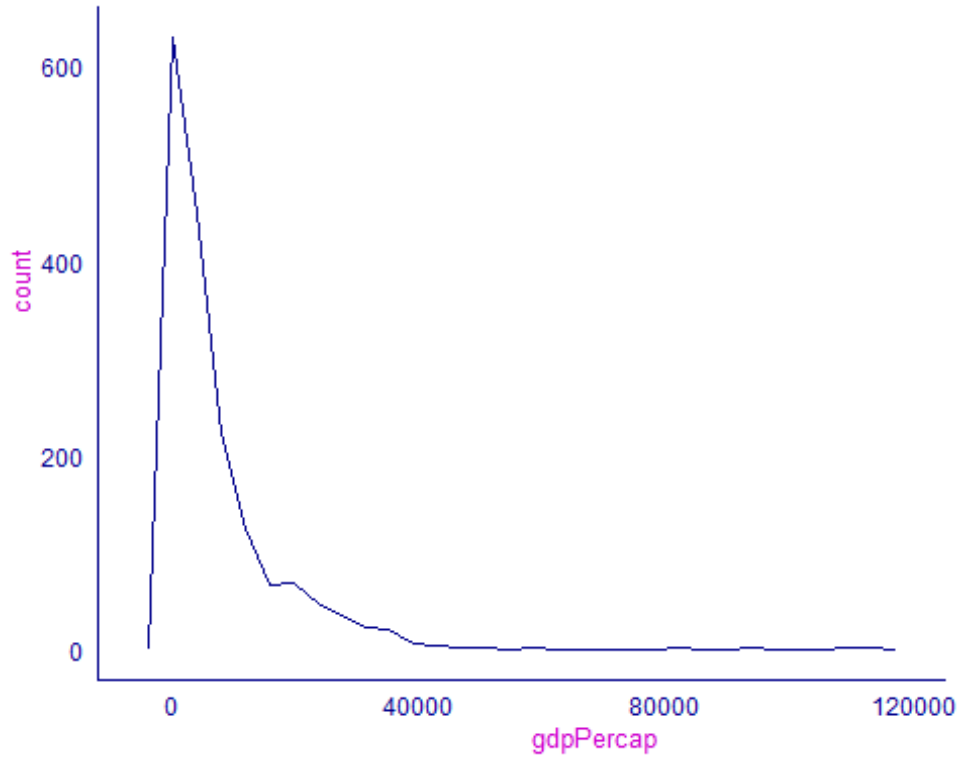
#Create a personalized theme

```
Hamed_theme<-theme_bw() + theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.title = element_text(size = 16, hjust = 0.4, color = "darkblue"),
  axis.title = element_text(size = 10, hjust = 0.6, color = "magenta3"),
  axis.text = element_text(colour = "darkblue", size = 9),
  axis.ticks = element_blank(),
  axis.line = element_line(colour = "darkblue", size=0.5, linetype =
"solid"),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  strip.text = element_text(size = 14, color = "darkblue"),
  panel.background = element_blank(),
  strip.background =element_blank(),
  plot.background = element_blank(),
  legend.text   = element_text(size = 12, hjust = 0.7, color = "darkblue"),
  legend.position = "top",
  legend.key = element_blank(),
  legend.title = element_blank()
)

#Now we set the new defined theme to the default option
theme_set(Hamed_theme)

#Simple chart
ggplot(diamonds, aes(carat)) +
  geom_freqpoly(colour = "deeppink4")
```

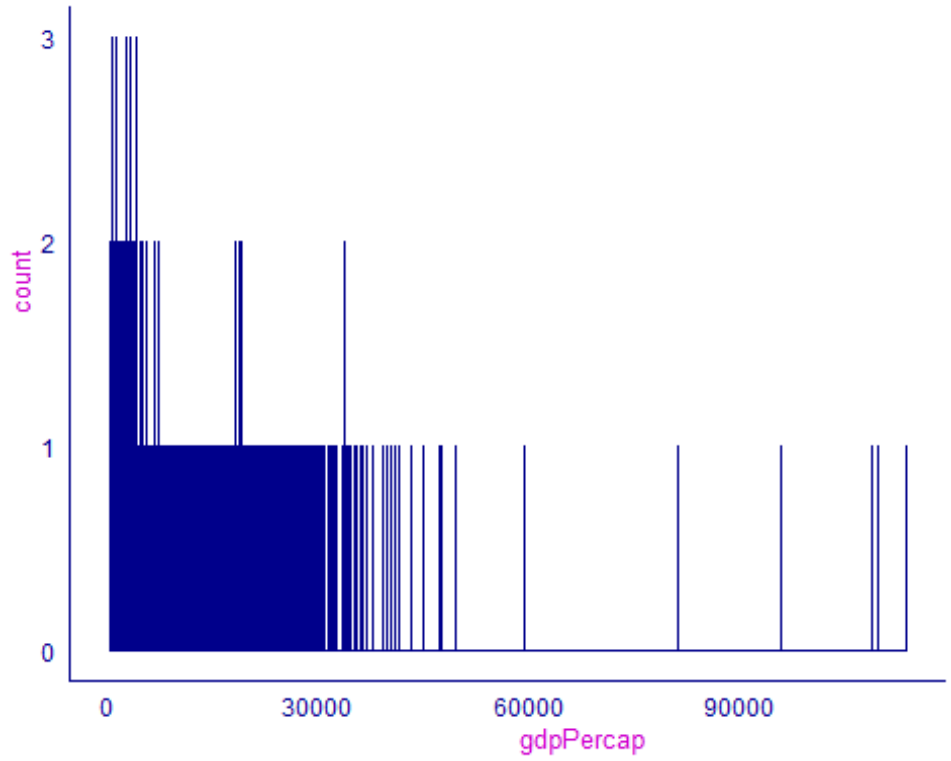


#Exercise 02: Frequency plot ##

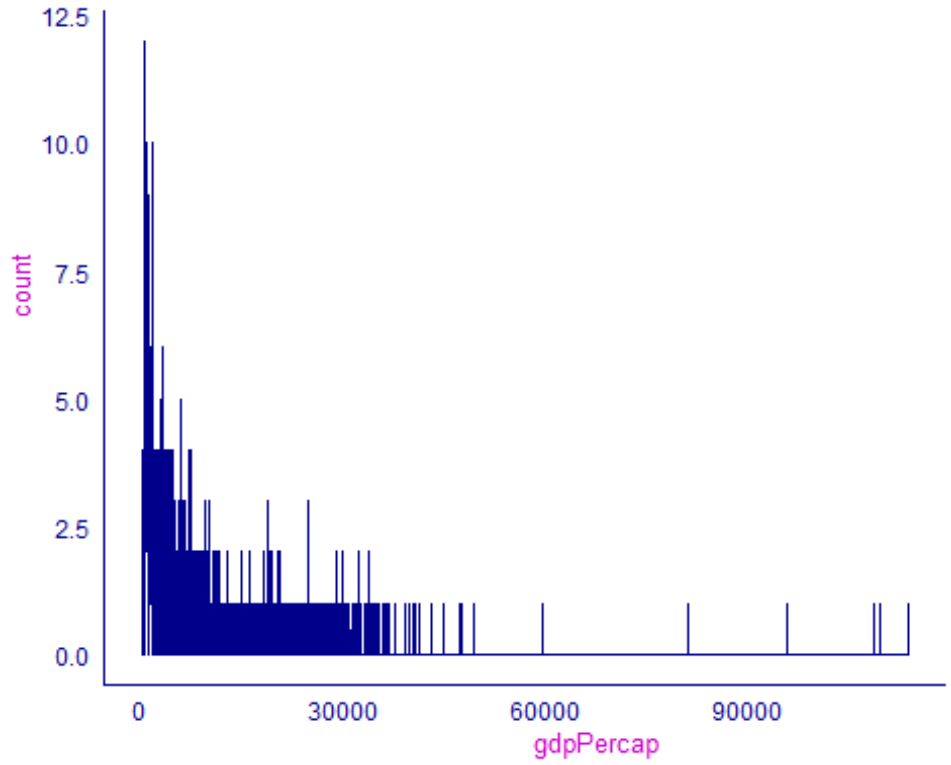
```
#gapminder dataset
#Simple chart
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour = "darkblue")
```



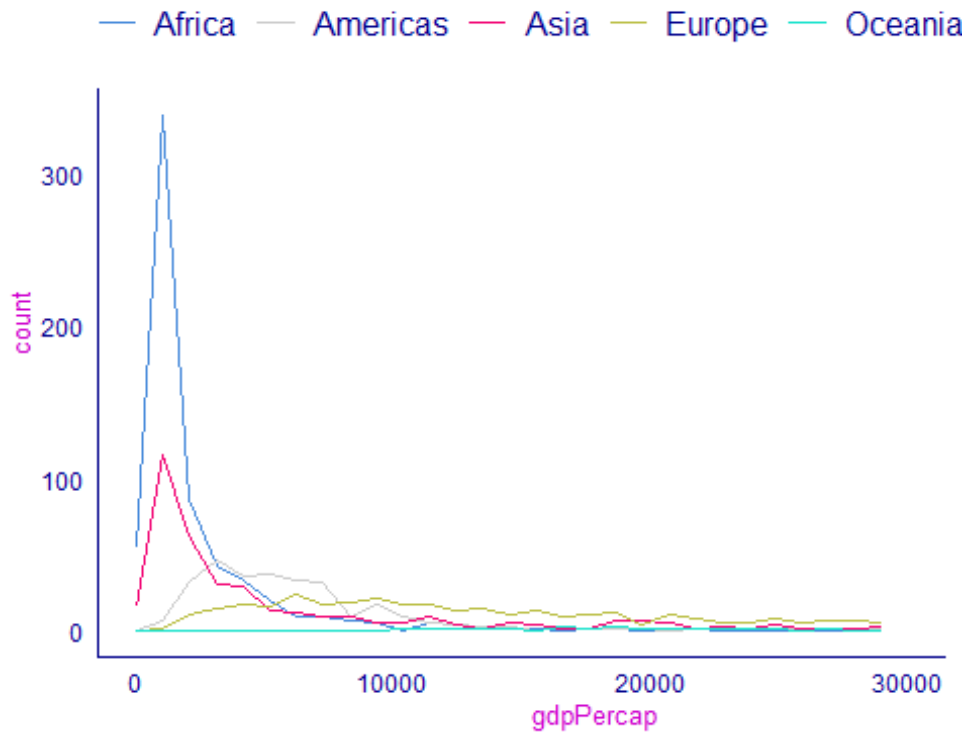

```
#Changing the bin width (more details)
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour = "darkblue", binwidth = 0.8)
```



```r
#Changing the bin width (less details)
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour ="darkblue", binwidth = 10)
```

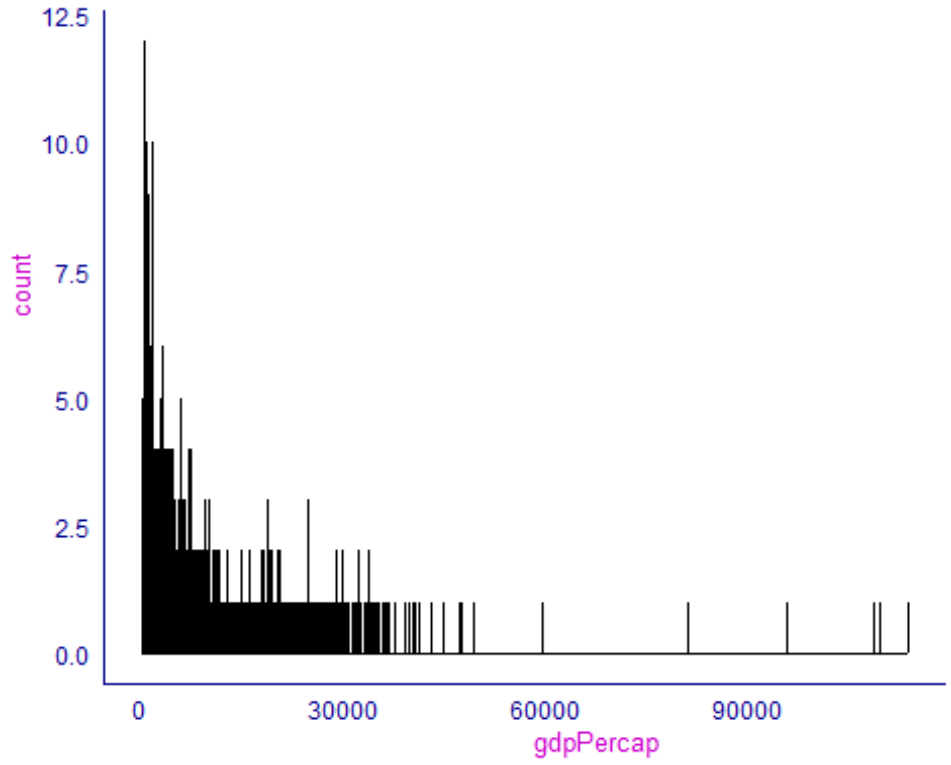



```r
# zoom by defining the limits for the x axis
ggplot(gapminder, aes(gdpPercap, colour = continent)) +
  geom_freqpoly() +
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048",
"#1ce3cd")) +
  xlim(0, 30000)
```

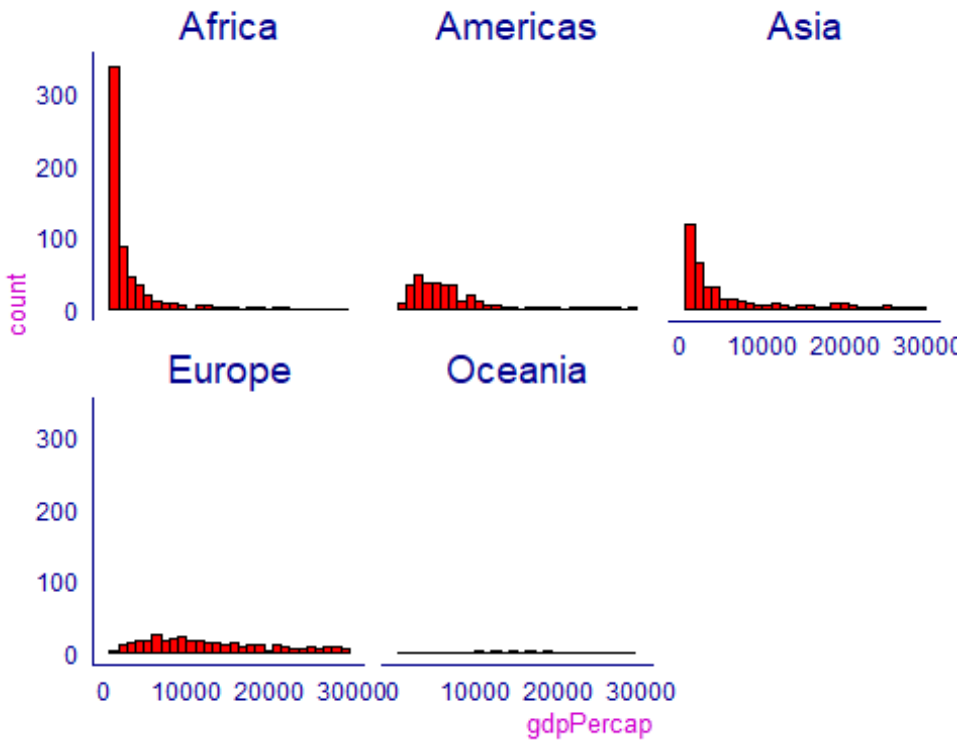


#Exercise 03: Histogram ##

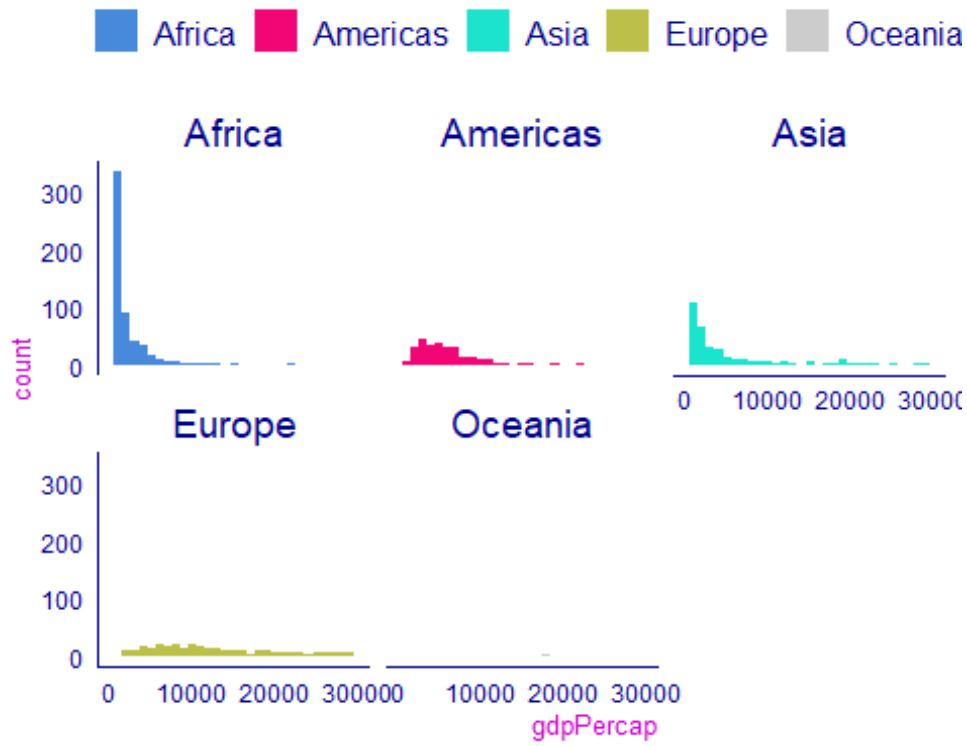
```
#gapminder dataset
# chart, the same with a histogram
ggplot(gapminder, aes(gdpPercap)) +
  geom_histogram(colour = "black", fill = "red1", binwidth = 10)
```



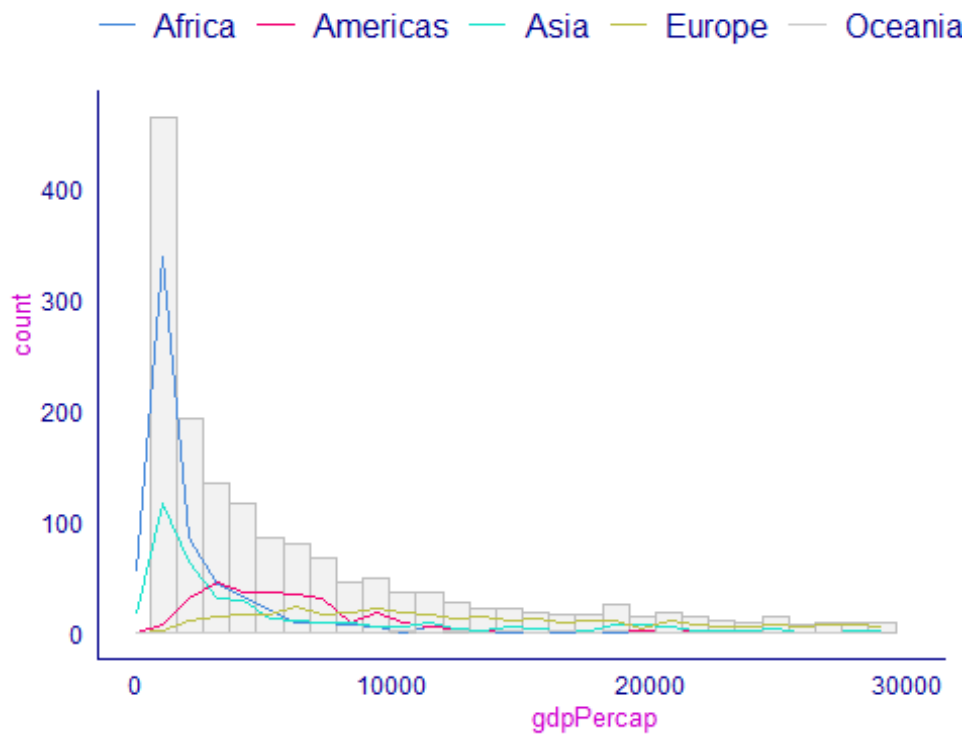

```
# zoom by defining the limits for the x axis+ facet
ggplot(gapminder, aes(gdpPercap)) +
  geom_histogram(colour = "black", fill = "red1") +
  xlim(0, 30000)+
  facet_wrap(. ~ continent)
```




```
#Changing the bin options+ zooming and facet
ggplot(gapminder, aes(gdpPercap, fill = continent)) +
  geom_histogram(position = "dodge", binwidth = 1000) +
  scale_fill_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048",
"#cccccc"))+
  xlim(0, 30000)+
  facet_wrap(. ~ continent)
```

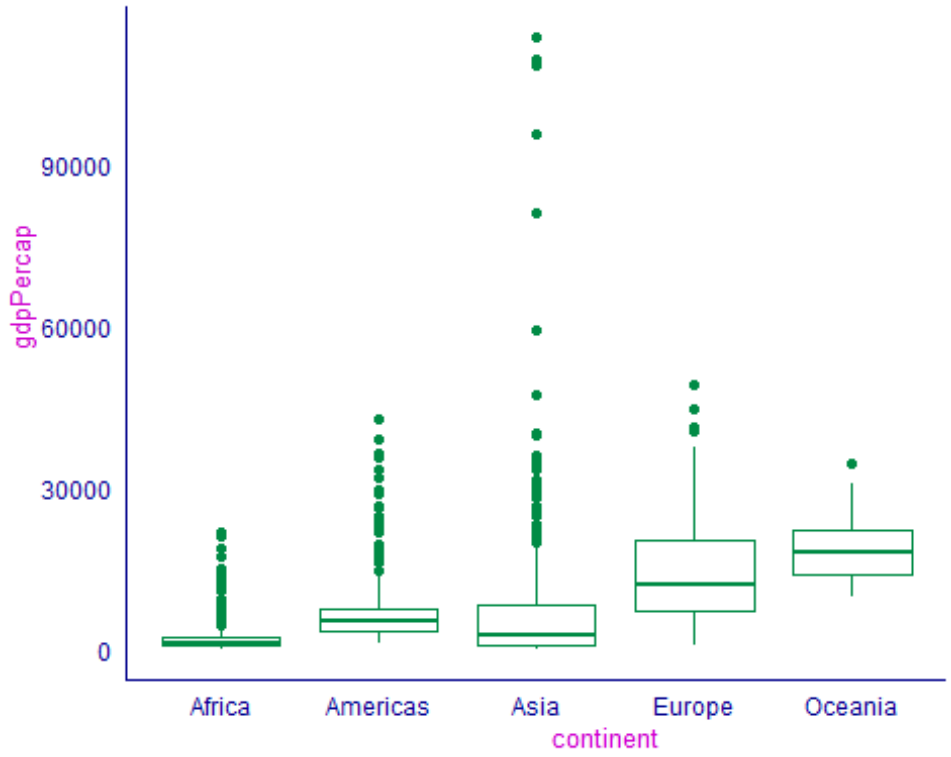



```
#The whole idea of the grammar of graphs+ zooming
ggplot(gapminder, aes(gdpPercap, color = continent)) +
  geom_histogram(colour="grey", fill = "grey", alpha = 0.2, size =0) +
  geom_freqpoly()+
  scale_colour_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048",
"#cccccc"))+ xlim(0, 30000)
```

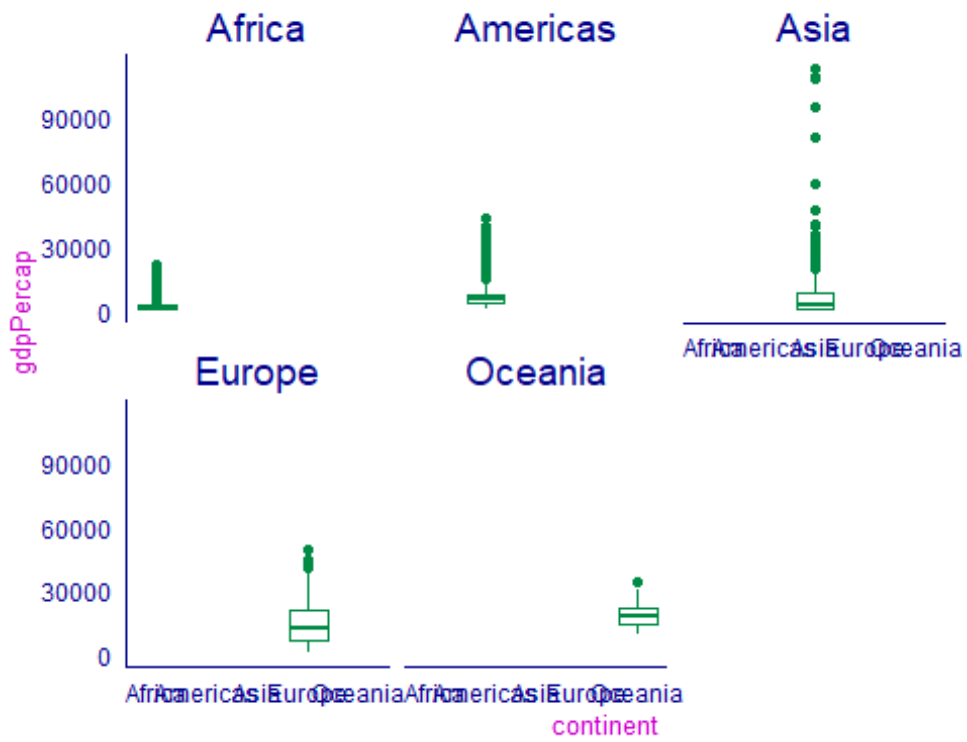


#Exercise 04: Boxplot ##

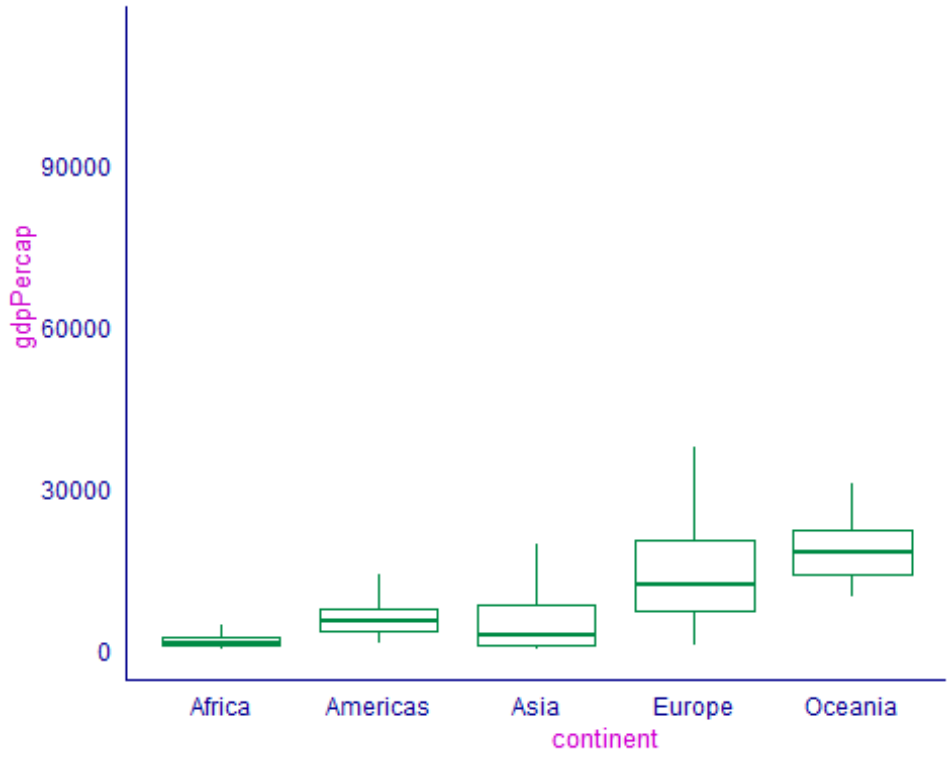
```
# gapminder dataset
# boxplot by category
ggplot(gapminder, aes(continent, gdpPercap)) +
  geom_boxplot(colour="springgreen4")
```



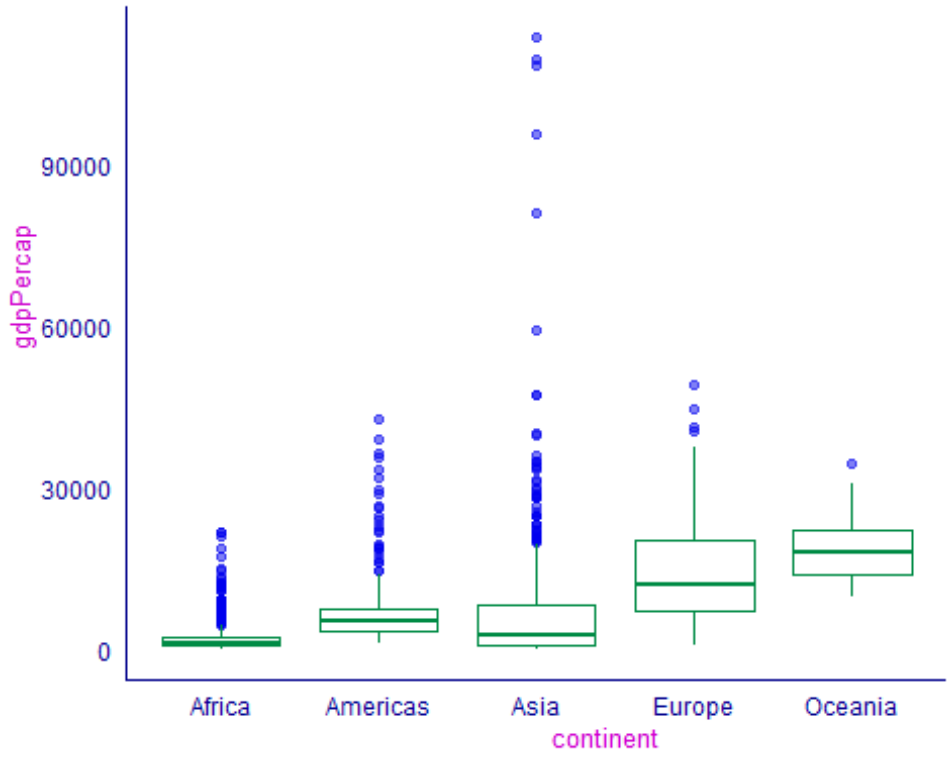

```
# boxplot by category + facet
ggplot(gapminder, aes(continent, gdpPercap)) +
  geom_boxplot(colour="springgreen4")+
  facet_wrap(. ~ continent)
```




```
# Without outliers
ggplot(gapminder, aes(continent, gdpPercap)) +
  geom_boxplot(aes(group = cut_width(continent, 1)), color="springgreen4",
               outlier.alpha=0)
```

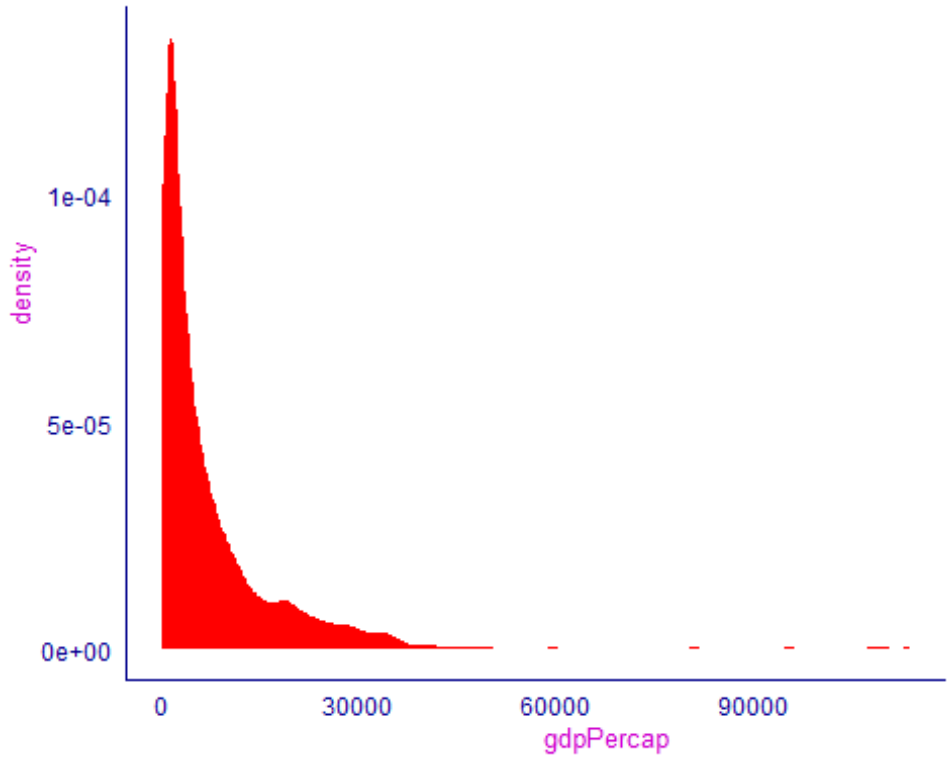



```
# Use a specific encoding for the outliers
ggplot(gapminder, aes(continent, gdpPercap)) +
  geom_boxplot(aes(group = cut_width(continent, 1)), color="springgreen4",
               color="springgreen4",
               outlier.alpha = 0.5,
               outlier.shape = 19,
               outlier.color="blue2")
```

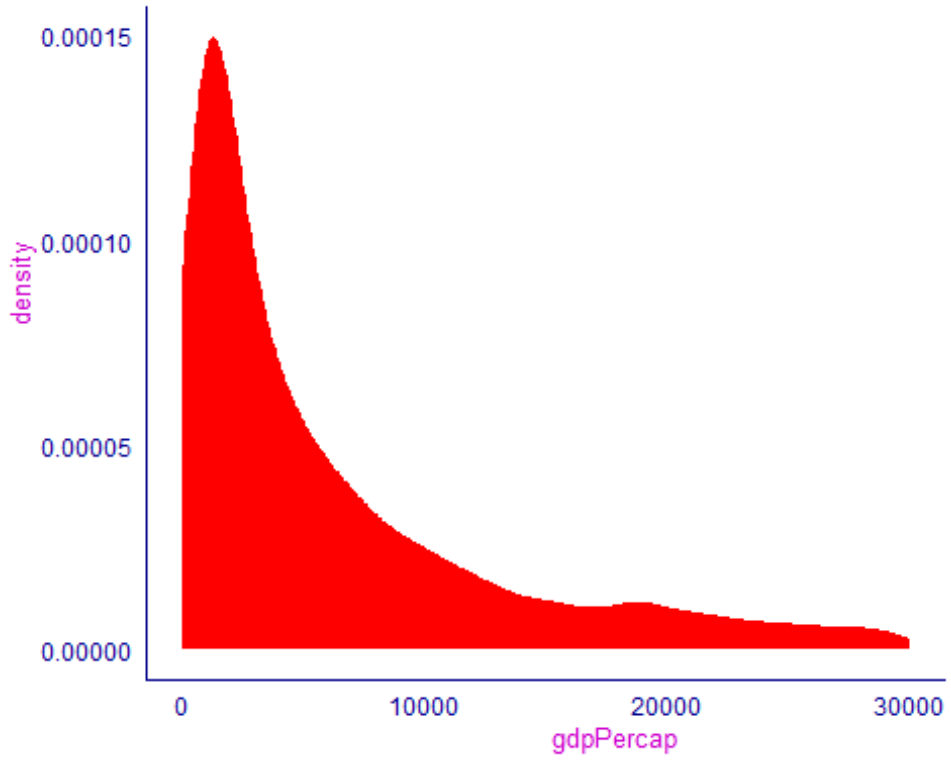


#Exercise 05: Density Plot ##

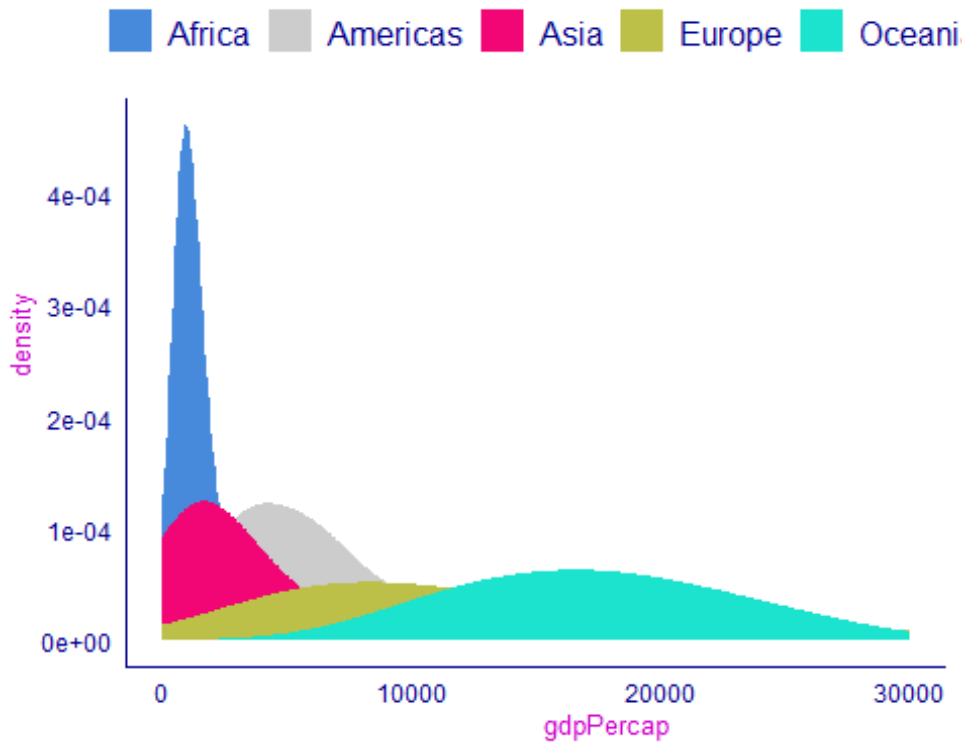
```
#gapminder dataset
#Simple chart
ggplot(gapminder, aes(gdpPercap)) +
  geom_density(fill = "red1", color = NA)
```



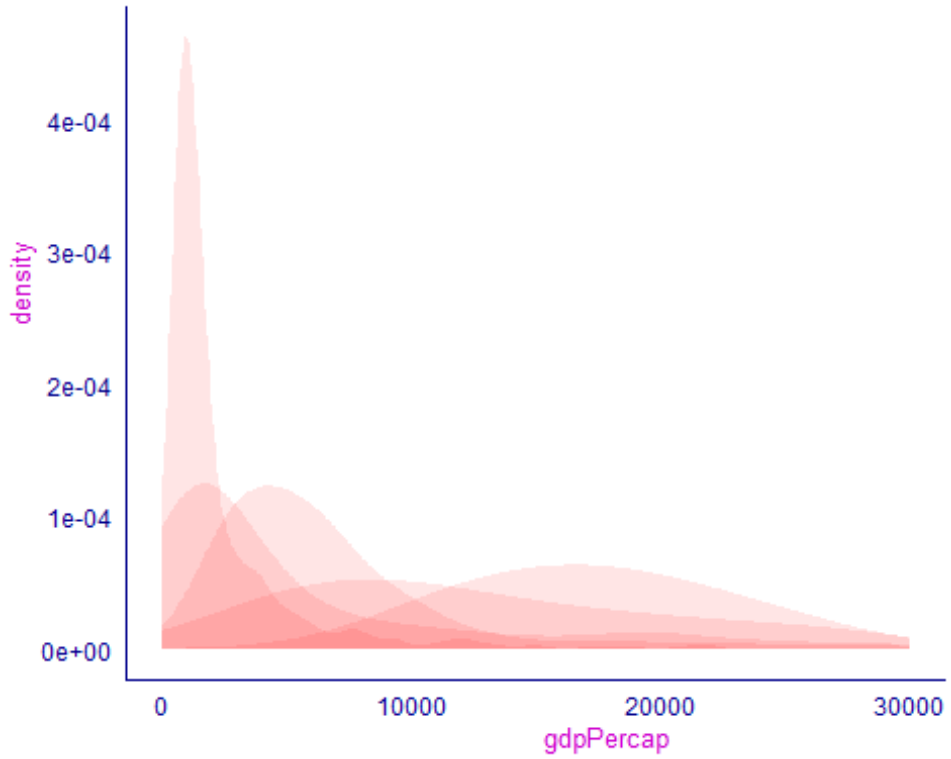

```
#Simple chart and zooming
ggplot(gapminder, aes(gdpPercap)) +
  geom_density(fill = "red1", color = NA)+
  xlim(0, 30000)
```



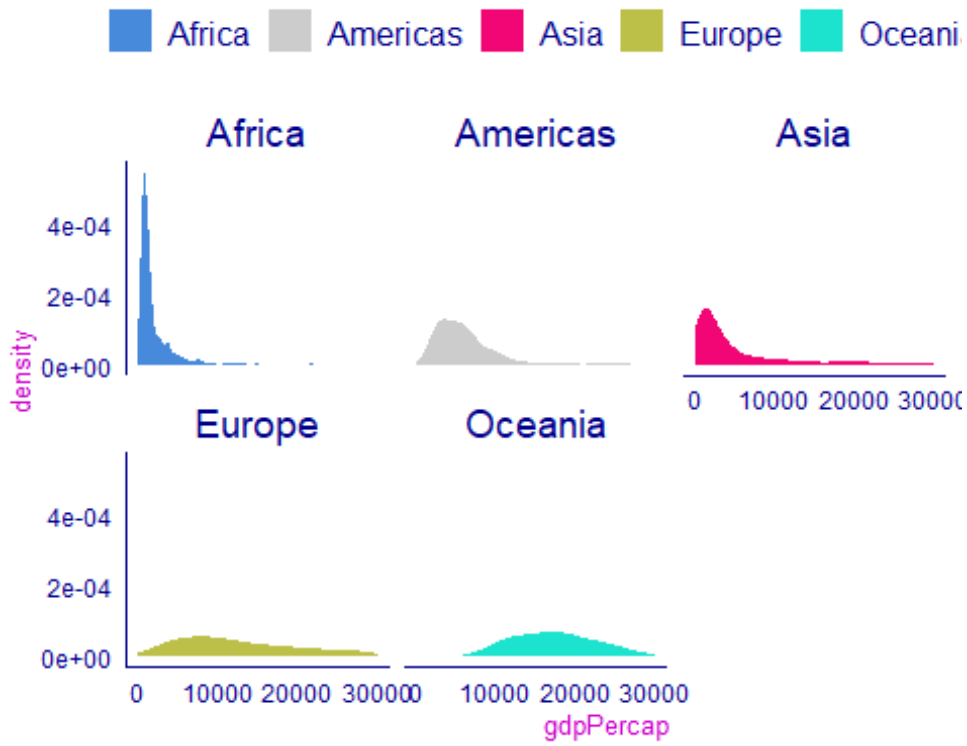

```
#Multiple density chart and zooming
ggplot(gapminder, aes(gdpPercap, group = continent, fill = continent)) +
  geom_density(adjust = 1.5 , color = NA) +
  scale_fill_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048",
"#1ce3cd"))+
  xlim(0, 30000)
```




```
#Multiple density chart and zooming
ggplot(gapminder, aes(gdpPercap, group=continent, fill=continent)) +
  geom_density(adjust=1.5 , color= NA, fill="red1", alpha =0.1)+
  xlim(0, 30000)
```

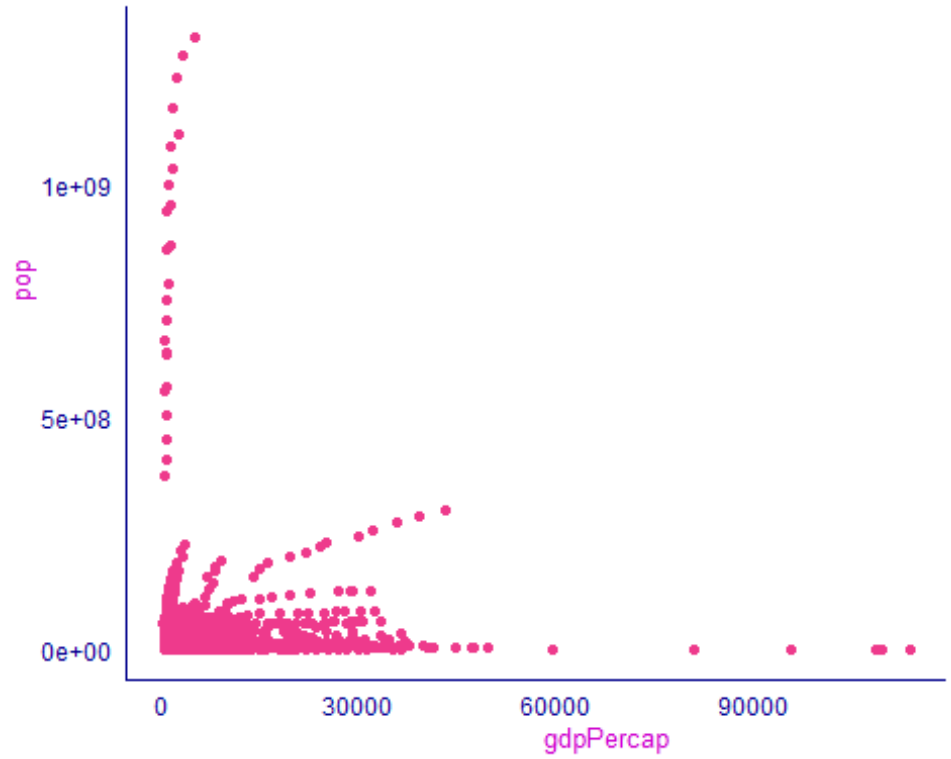



```
#Small multiple density by the different cuts and zooming
ggplot(gapminder, aes(gdpPercap, stat(density), fill=continent)) +
  geom_density(color = NA) +
  scale_fill_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048",
"#1ce3cd")) +
  facet_wrap(. ~ continent)+
  xlim(0, 30000)
```

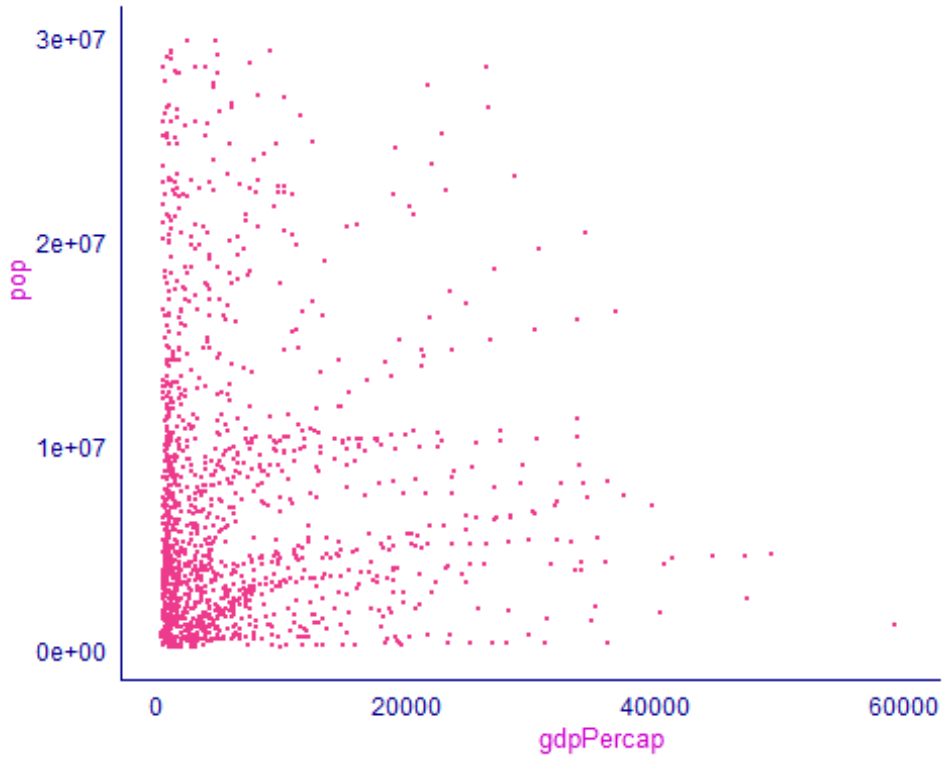


#Exercise 06: Scatter Plot##

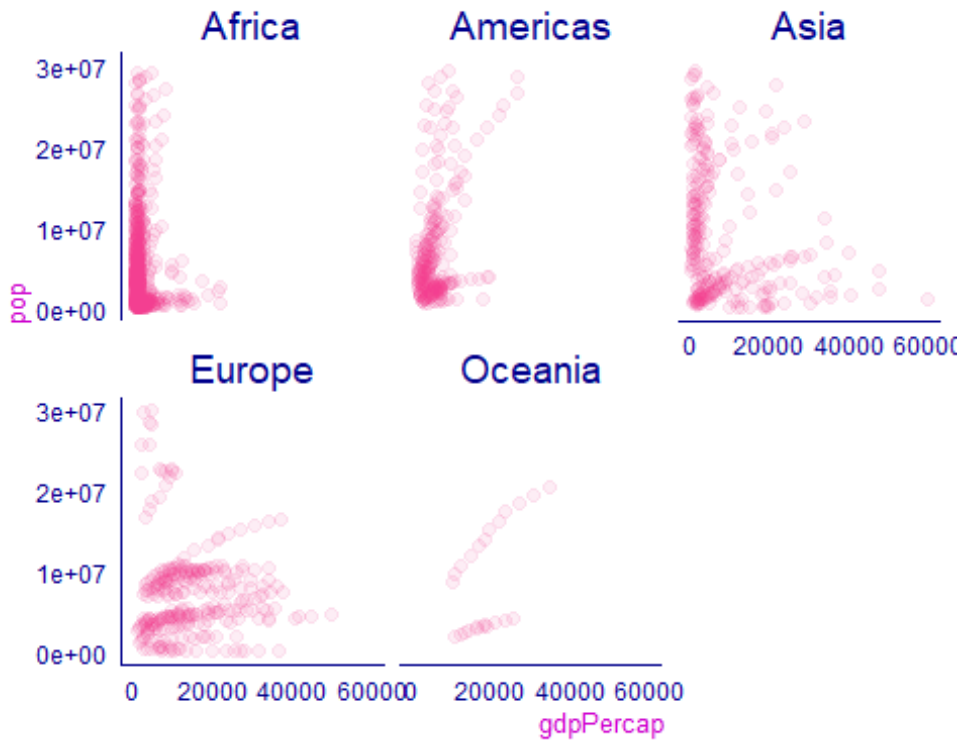
```
#gapminder dataset
#Basic scatter plot
ggplot(gapminder, aes(x=gdpPercap, y=pop)) +
  geom_point(color="violetred2")
```



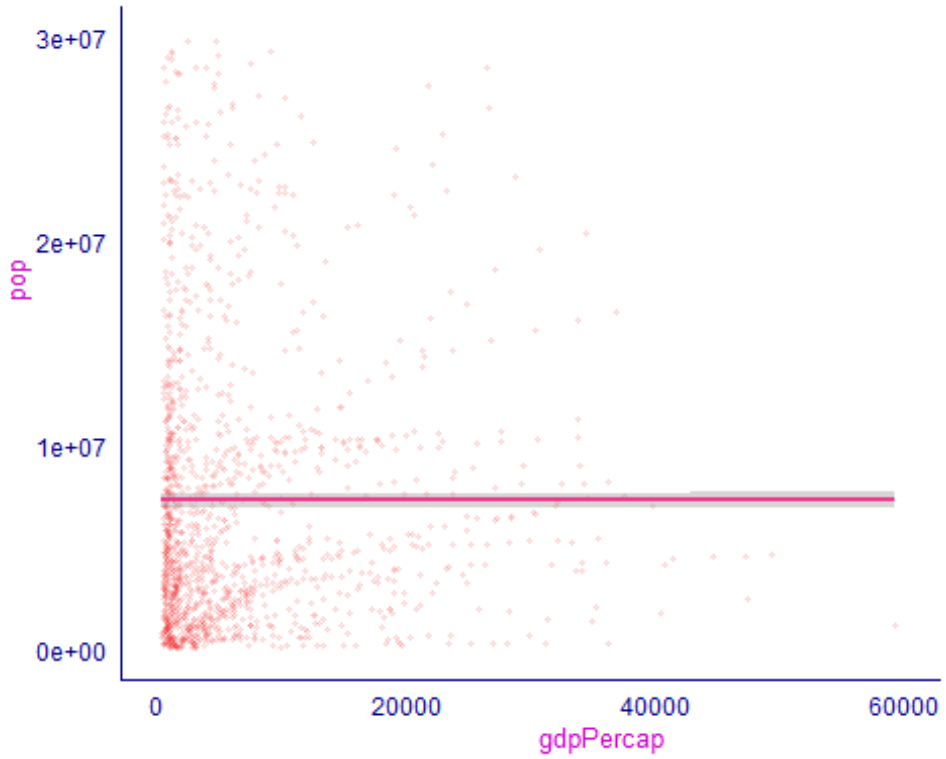

```
#Basic scatter plot - adjusting the size + change x and y limits
ggplot(gapminder, aes(x=gdpPercap, y=pop)) +
  geom_point(size=0.5, color="violetred2") +
  ylim(0, 30000000) +
  xlim(0, 60000)
```



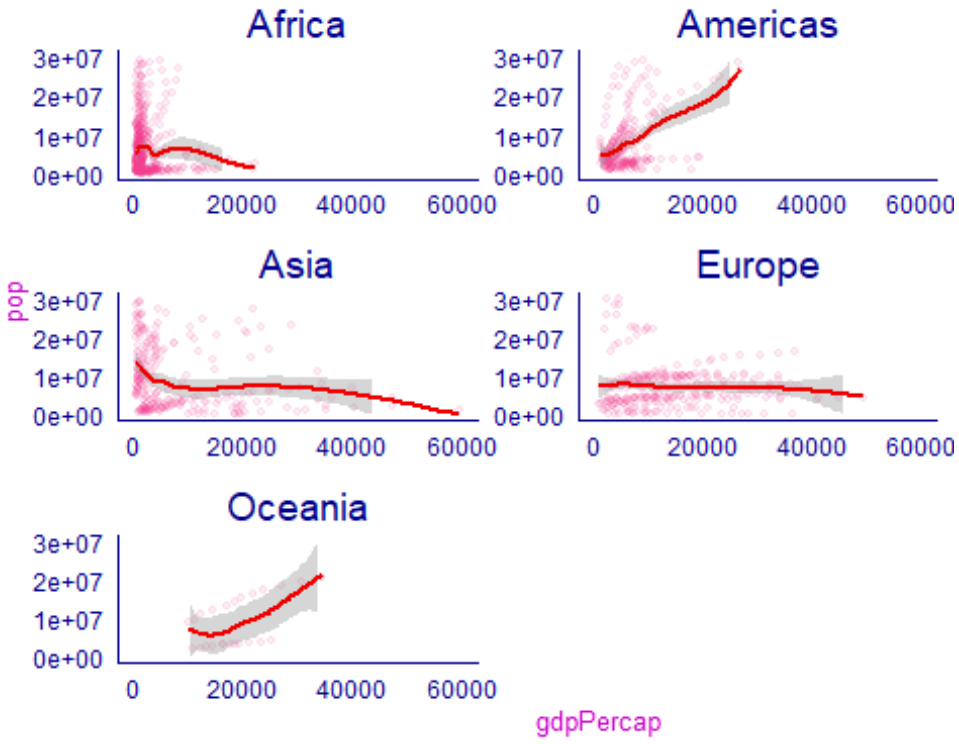

```
#Basic scatter plot - adjusting the opacity + change x and y limits
ggplot(gapminder, aes(x=gdpPercap, y=pop)) +
  geom_point(size=2, alpha=0.09, color="violetred2") +
  ylim(0, 30000000) +
  xlim(0, 60000)  +  facet_wrap(. ~ continent)
```




```
#Adding a trend line + change x and y limits
ggplot(gapminder, aes(x=gdpPercap, y=pop)) +
  geom_point(color="red2", size=0.8, alpha=0.09)+
  stat_smooth(color="violetred2")+
  ylim(0, 30000000) +
  xlim(0, 60000)
```

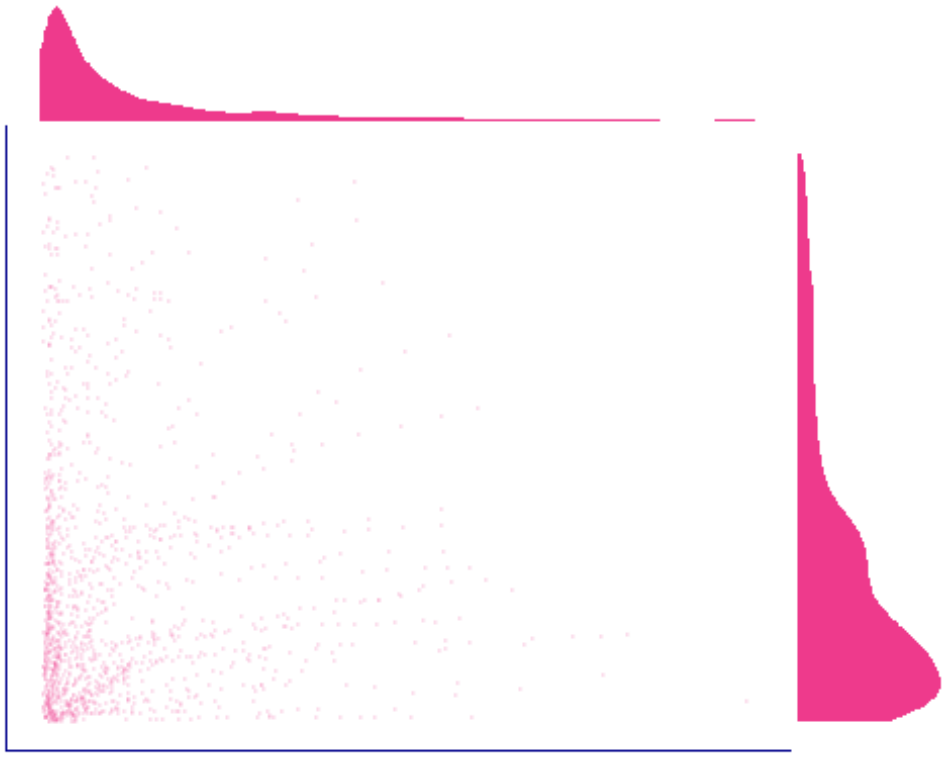



```
#Small multiples - one variable with free scale + change x and y limits
ggplot(gapminder, aes(x=gdpPercap, y=pop)) +
  geom_point(color="violetred2", size=1, alpha=0.09)+
  facet_wrap( ~ continent, ncol=2, scales = "free") +
  stat_smooth(color="red2") +
  ylim(0, 30000000) +
  xlim(0, 60000)
```

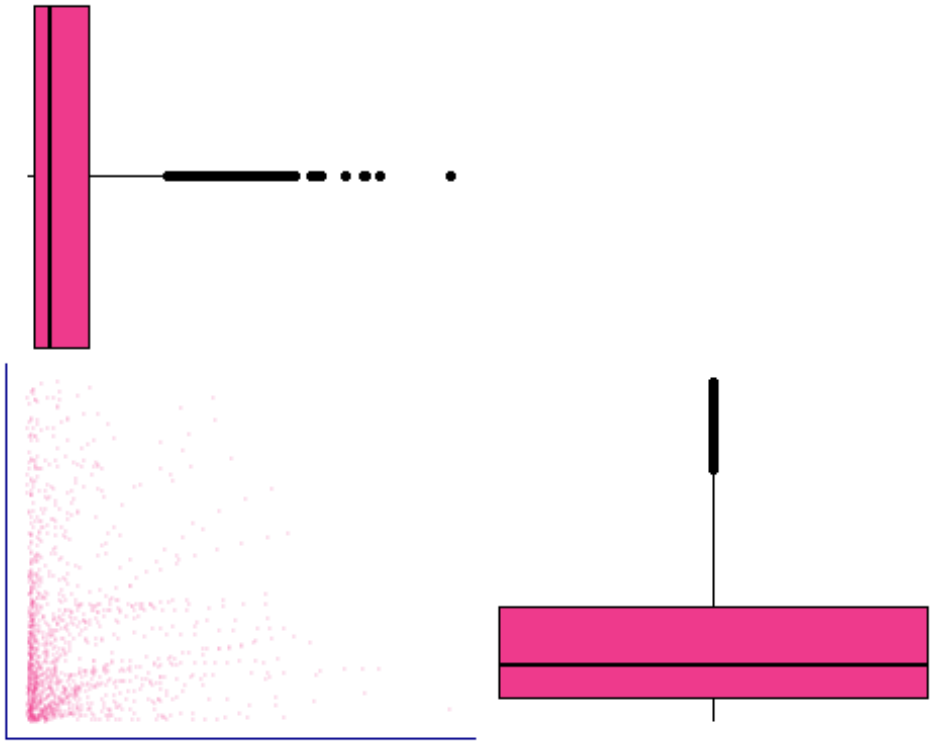


#Exercise 07: Marginal Scatter plot ##

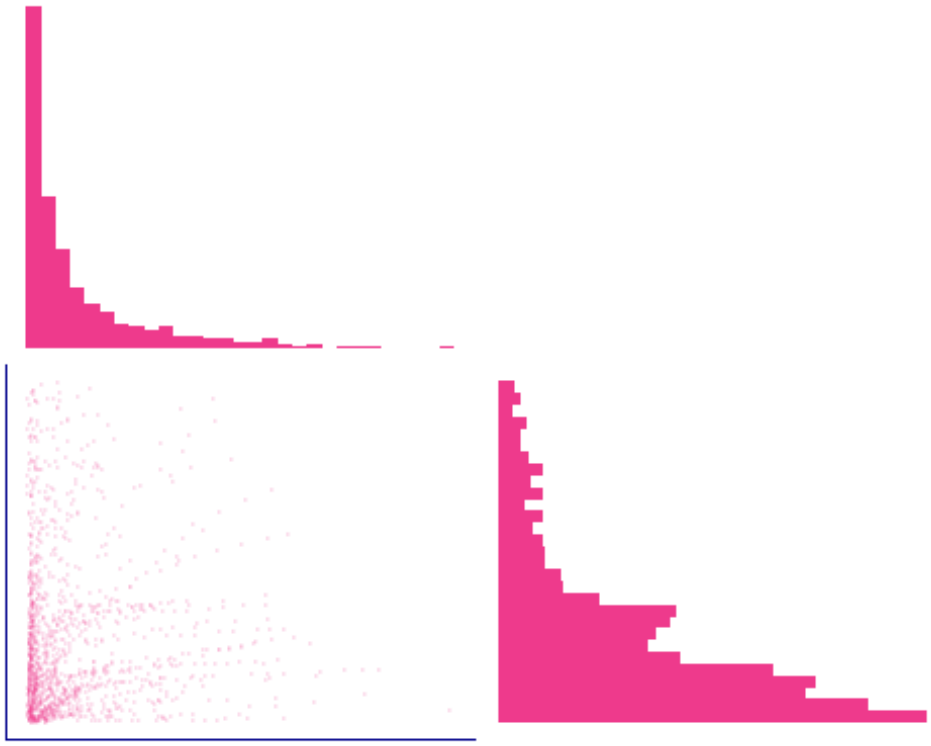
```
#gapminder dataset

#Density and change x and y limits
pp <- ggplot(gapminder, aes(x=gdpPercap, y=pop)) +
  geom_point(size=0.5, alpha=0.09, color="violetred2") +
  theme(axis.title=element_blank(), axis.text=element_blank())+
  ylim(0, 30000000) +
  xlim(0, 60000)

ggMarginal(pp, type = "density", fill="violetred2", alpha=1,
color='transparent')
```



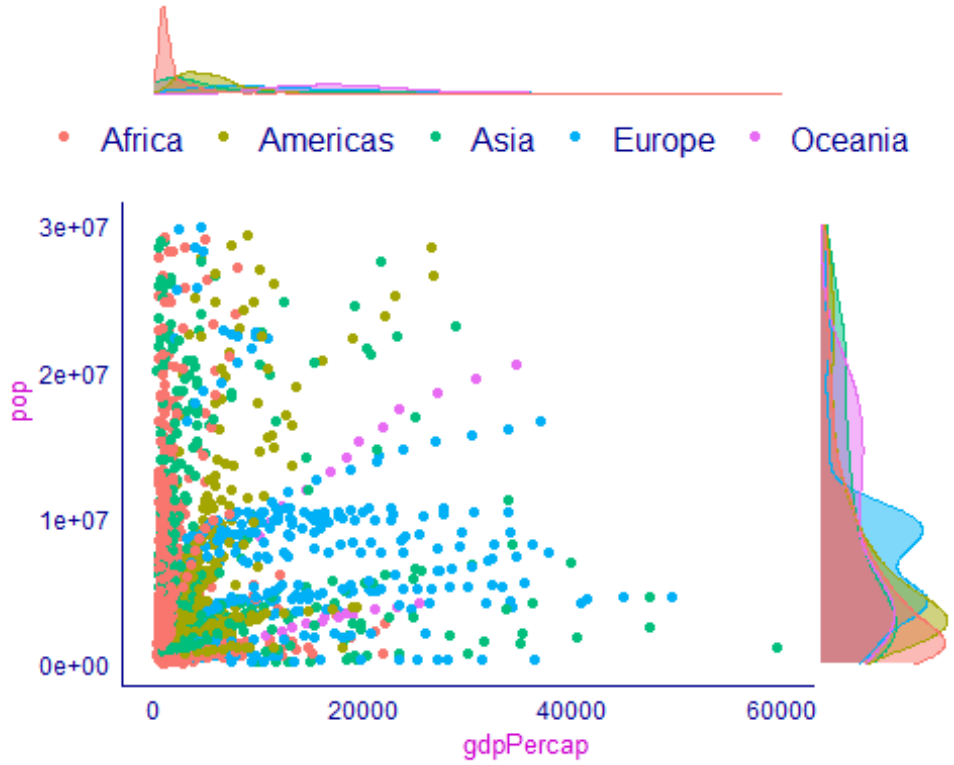
```
#Box-plot
ggMarginal(pp, type = "boxplot", size=1, fill="violetred2")
```



```r
#Histogram
ggMarginal(pp, type = "histogram", size=1, fill="violetred2", colour=NA)
```



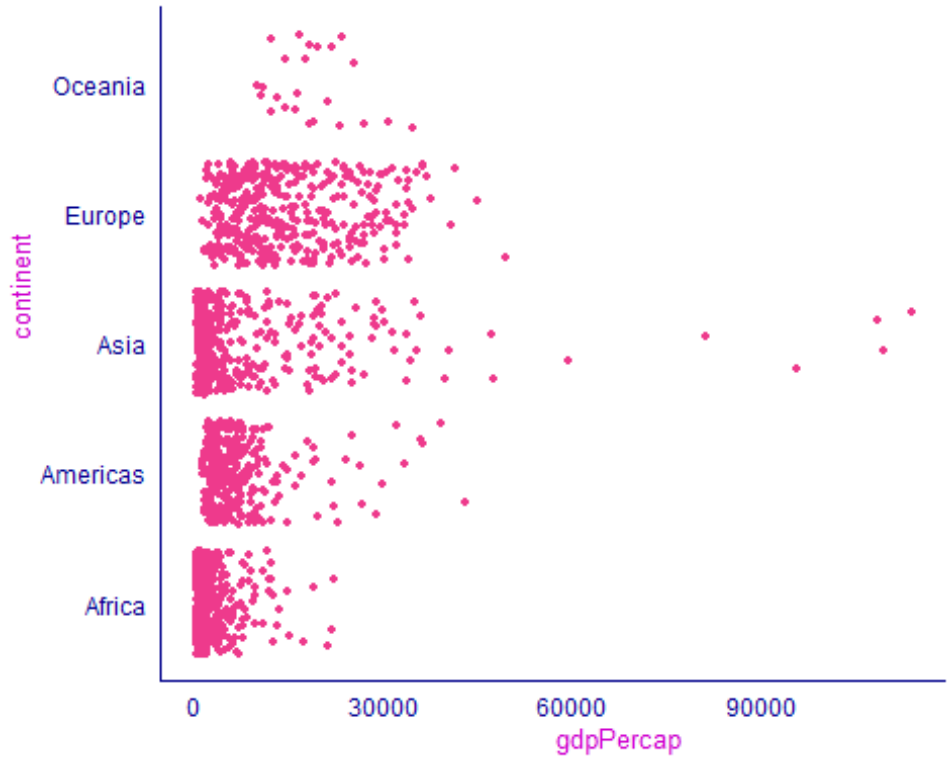
```r
#Using the gapminder data set + change x and y limits
pgapminder <- ggplot(gapminder, aes(x=gdpPercap, y=pop, colour = continent))
+
  geom_point() +
  ylim(0, 30000000) +
  xlim(0, 60000)

ggMarginal(pgapminder, groupColour = TRUE, groupFill = TRUE)
```



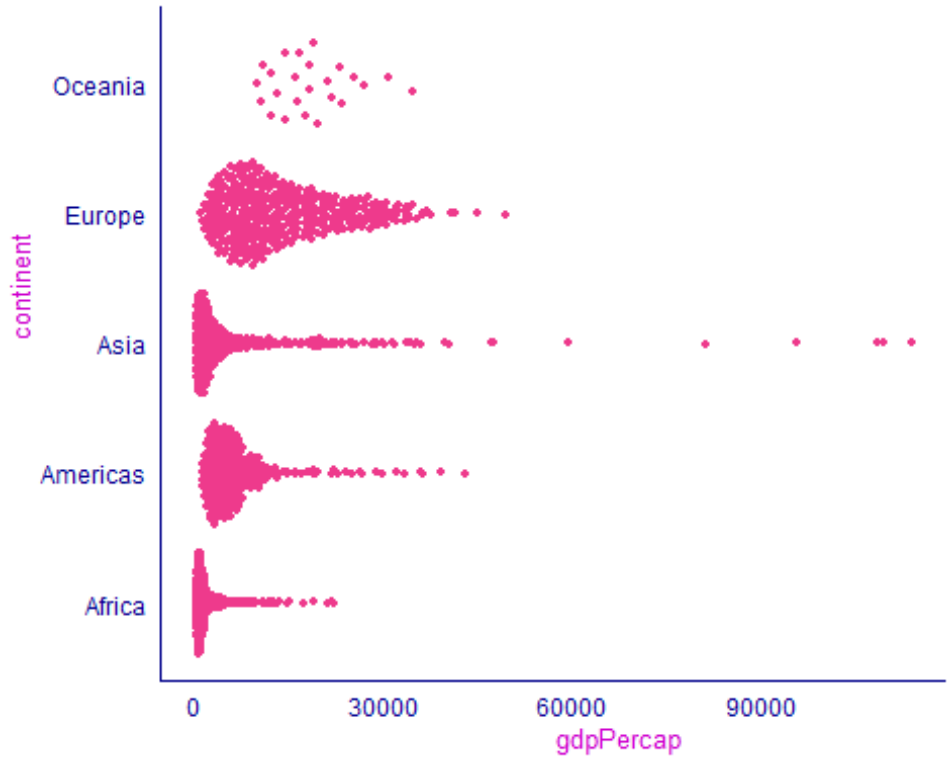
#Exercise 08: Beeswarm plot ##

```
#gapminder dataset
#Changing size and opacity
ggplot(gapminder, aes(x=continent, y=gdpPercap)) +
  geom_jitter(size=1, alpha=1, color="violetred2")
```

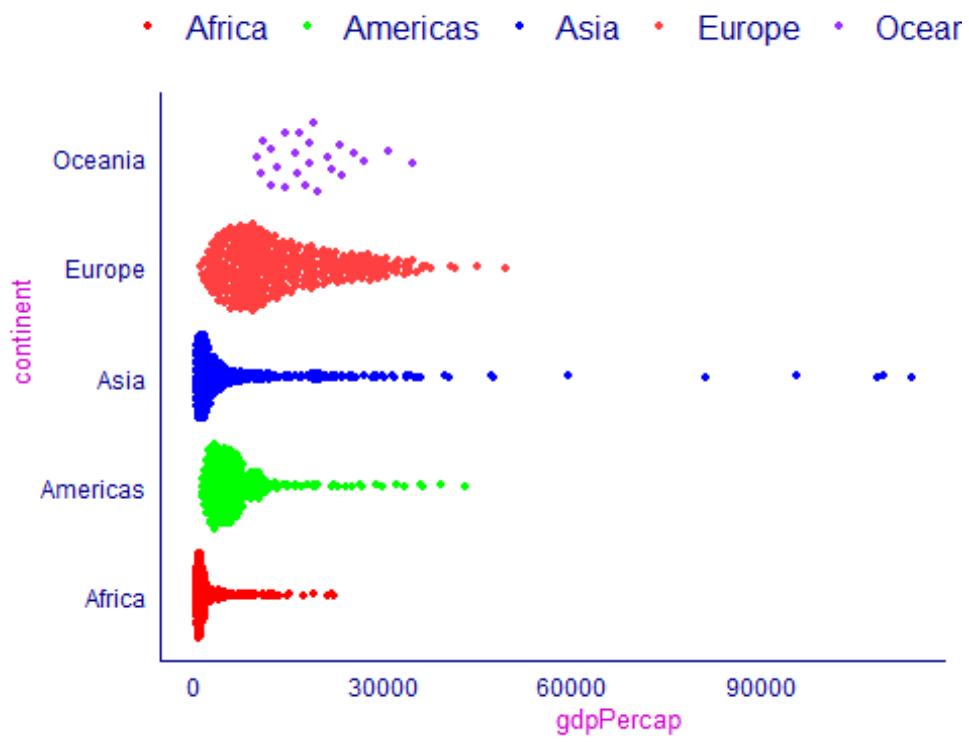

```
#Switching axis
ggplot(gapminder, aes(x=gdpPercap, y=continent)) +
  geom_jitter(size=1, alpha=1, color="violetred2")
```



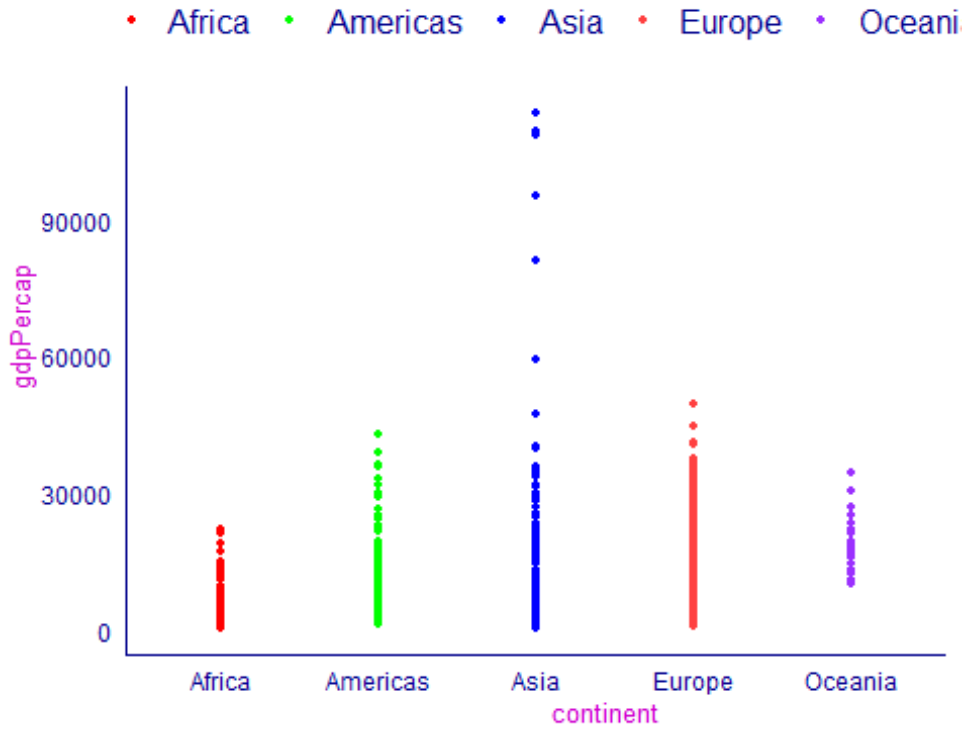
```r
#Simple beewswarm
ggplot(gapminder, aes(x=gdpPercap, y=continent)) +
  geom_quasirandom(size=1, alpha=1, color="violetred2", groupOnX=FALSE)
```




```r
#Adding color
ggplot(gapminder, aes(x=gdpPercap, y=continent, colour=continent)) +
  geom_quasirandom(size=1, alpha=1, groupOnX=FALSE) +
  scale_colour_manual(values=c("red1", "green1", "blue1", "brown1",
"purple1"))
```

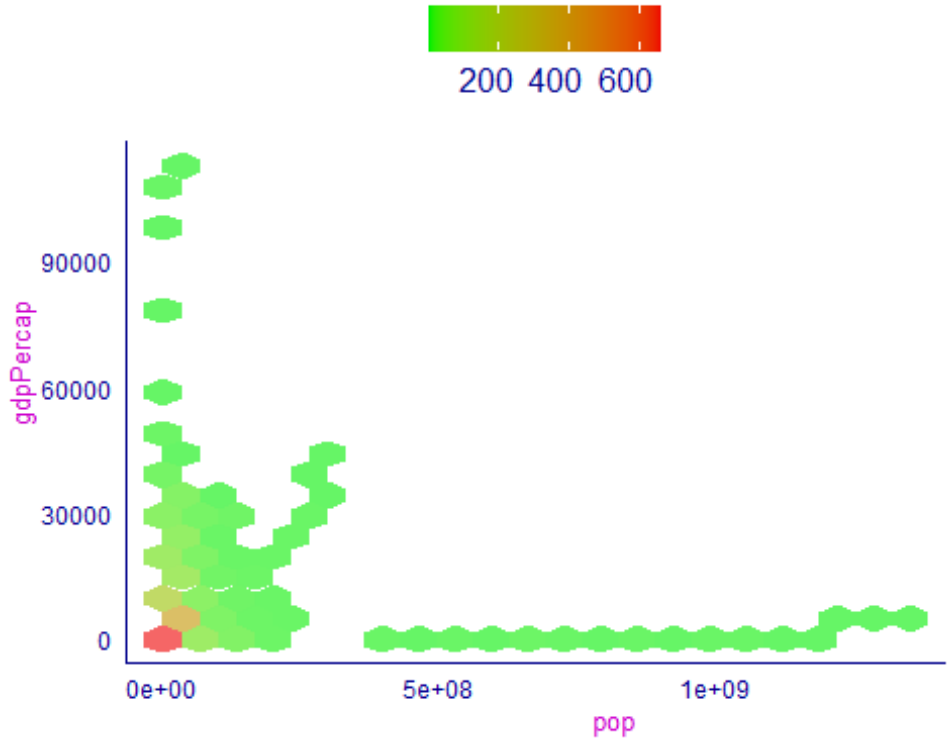



```
#Adding color
ggplot(gapminder, aes(x=continent, y=gdpPercap, colour=continent)) +
  geom_quasirandom(size=1, alpha=1, groupOnX=FALSE, method = "smiley") +
  scale_colour_manual(values=c("red1", "green1", "blue1", "brown1",
"purple1"))
```

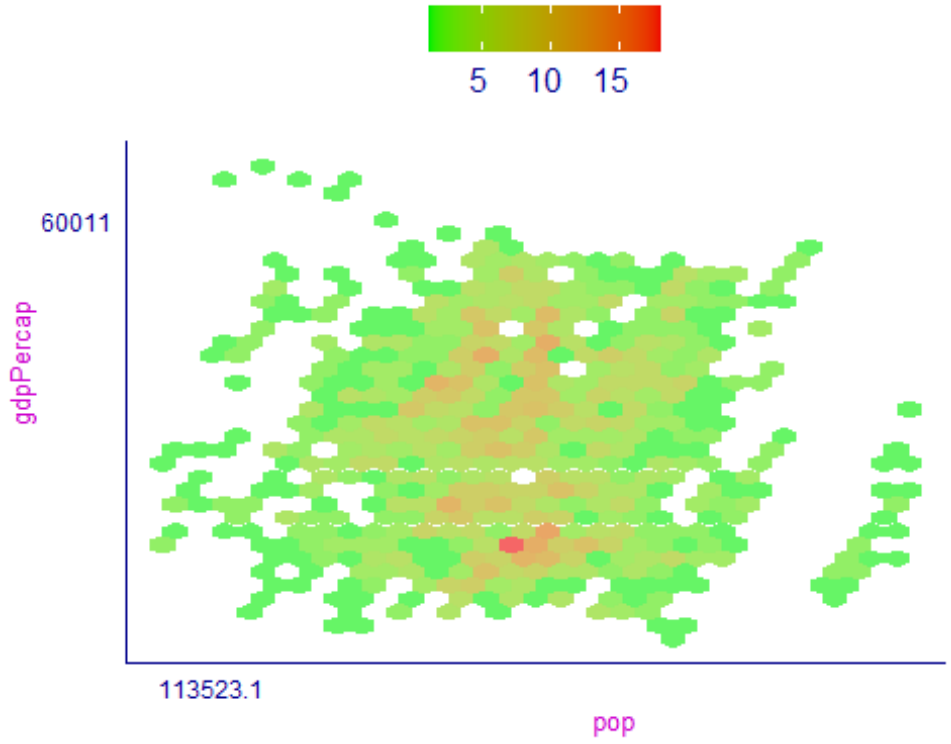


#Exercise 09: Hexagonal Binning ##

```
#gapminder dataset
#Aggregation through hexagonal binning - defining the number of bins
ggplot(gapminder, aes(x=pop, y=gdpPercap))+
  geom_hex(bins=20, alpha = 0.6)+
  scale_fill_gradient(low="green2", high="red2")
```

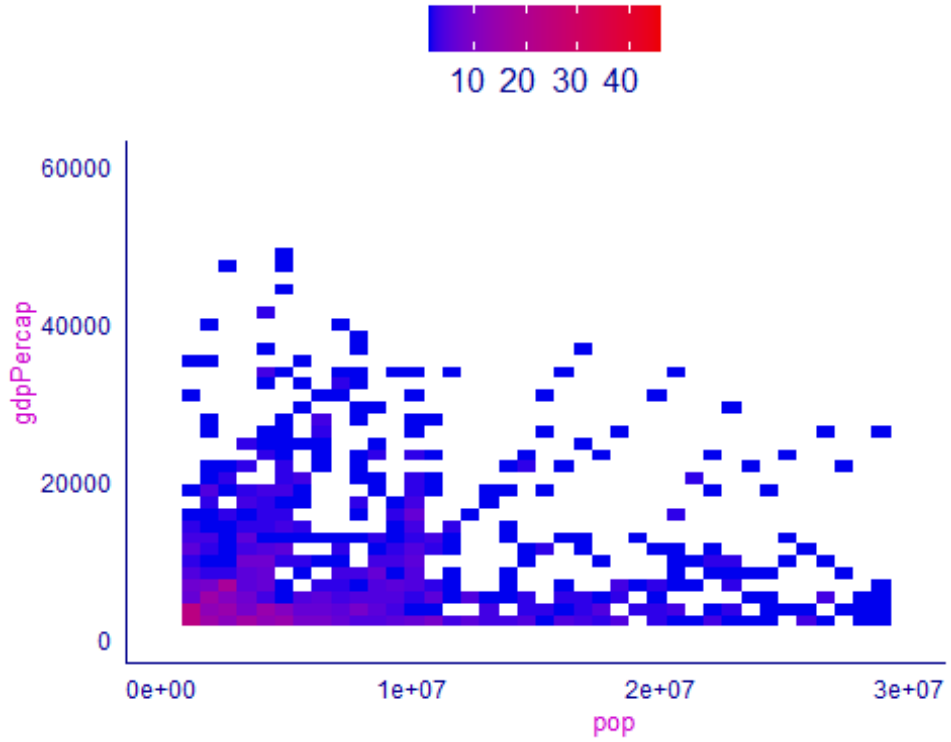



```
#Aggregation through hexagonal binning - logaritmic scaling
ggplot(gapminder, aes(x=pop, y=gdpPercap))  +
  geom_hex(alpha = 0.6) +
  scale_x_log10(breaks = round(as.vector(quantile(gapminder$gdpPercap)),
digits = 1))+
  scale_y_log10(breaks = round(as.vector(quantile(gapminder$pop)), digits =
1))+
  scale_fill_gradient(low="green2", high="red2")
```

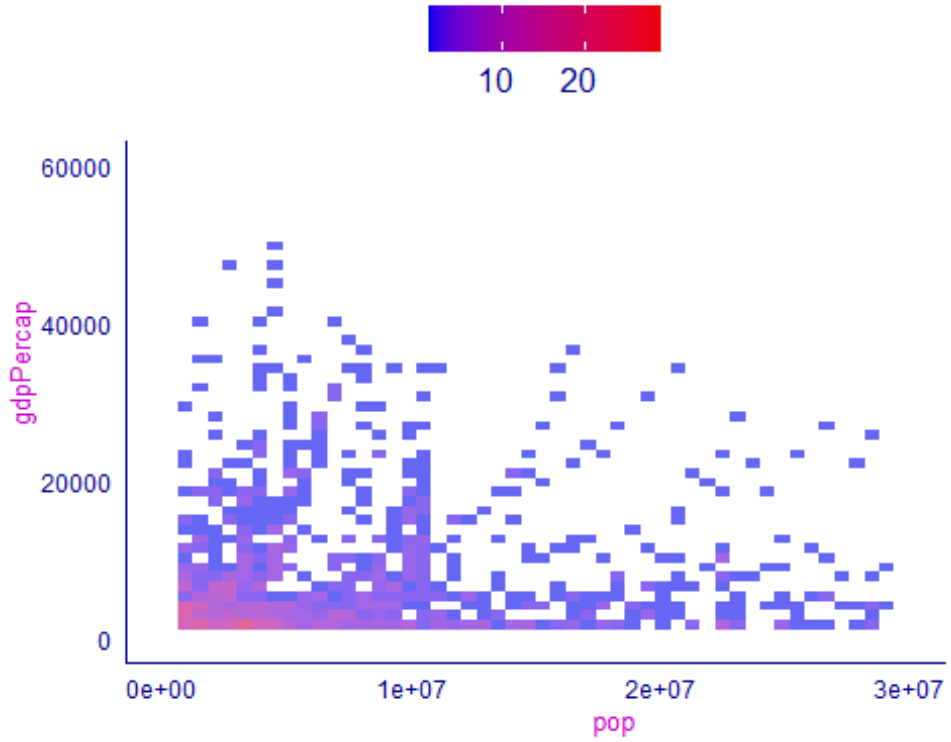


#Exercise 10: Scales and Axes ##

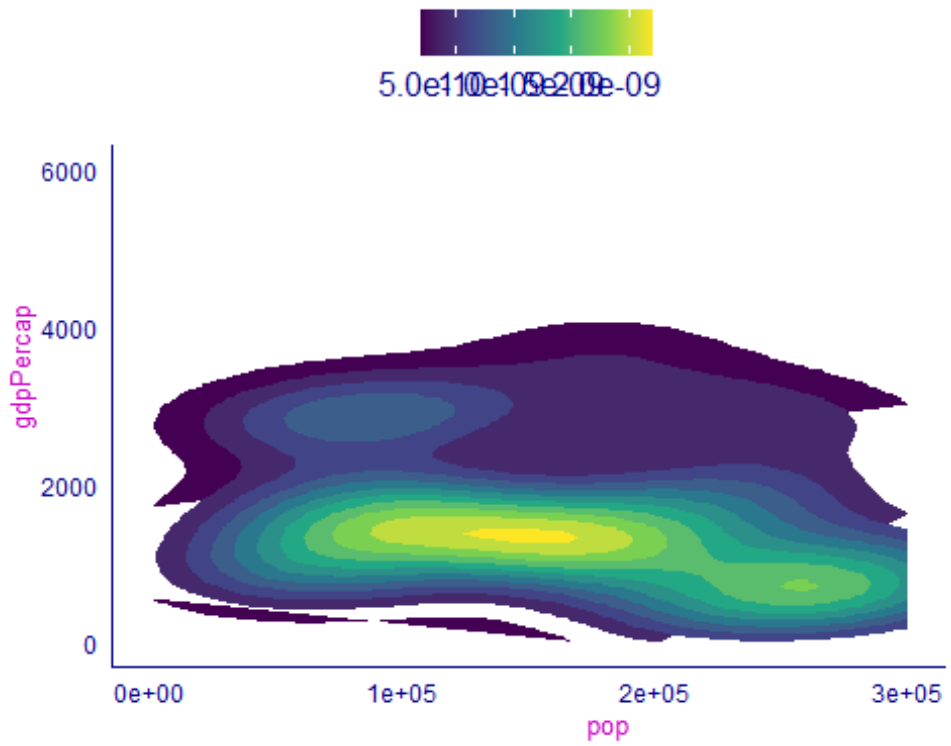
```
#gapminder dataset
#Heatmap based on rectangles and change x and y limits
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_bin2d(bins = 40) +
  scale_fill_gradient(low="blue2", high="red2")+
  xlim(0, 30000000) +
  ylim(0, 60000)
```



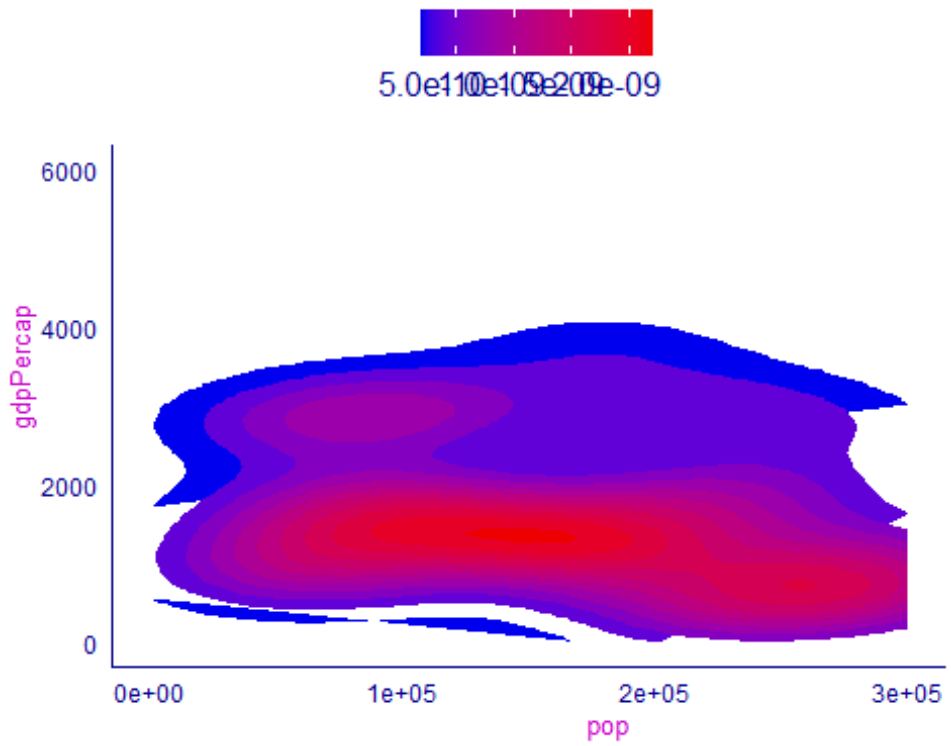

```
#Change opacity and change x and y limits
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_bin2d(bins = 50, alpha = 0.6)+
  scale_fill_gradient(low="blue2", high="red2") +
  xlim(0, 30000000) +
  ylim(0, 60000)
```



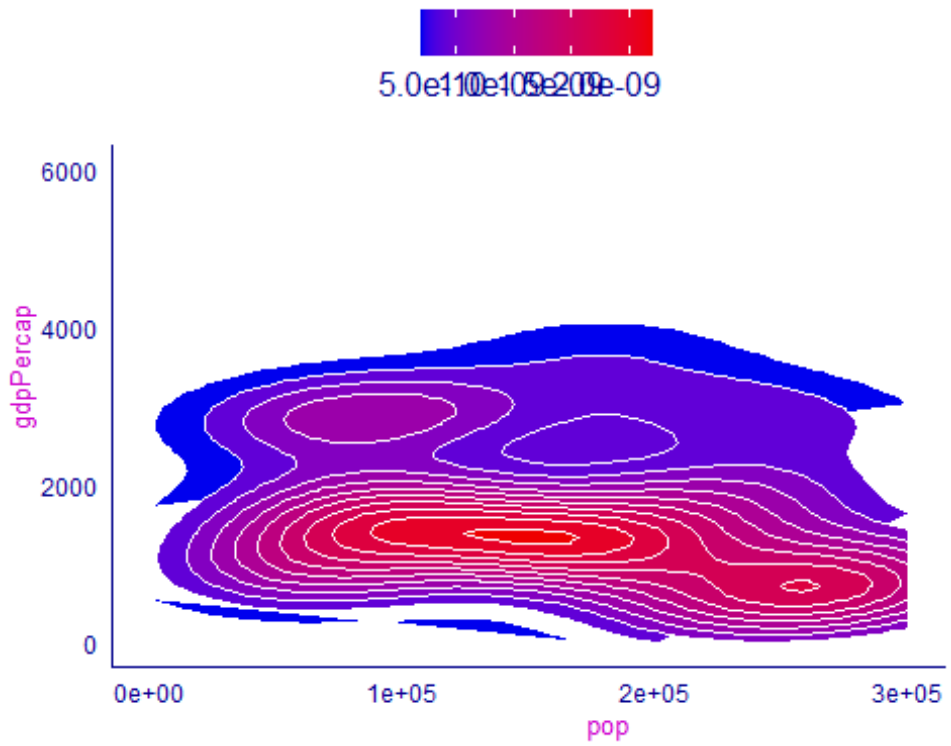

```
#Density estimation with contours and change x and y limits
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon")  +
  scale_fill_continuous(type = "viridis")+
  xlim(0, 300000) +
  ylim(0, 6000)
```




```
#Density estimation with contours and change x and y limits
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon") +
  scale_fill_gradient(low="blue2", high="red2")+
  xlim(0, 300000) +
  ylim(0, 6000)
```

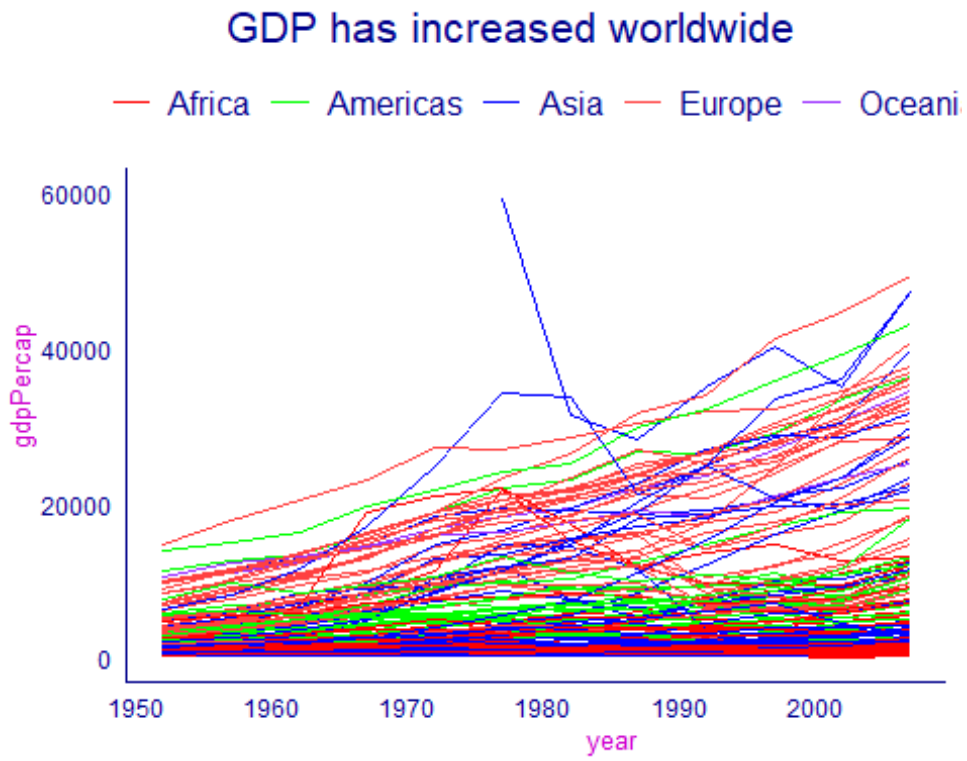



```
#Adding a stroke and change x and y limits
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour="white") +
  scale_fill_gradient(low="blue2", high="red2")+
  xlim(0, 300000) +
  ylim(0, 6000)
```

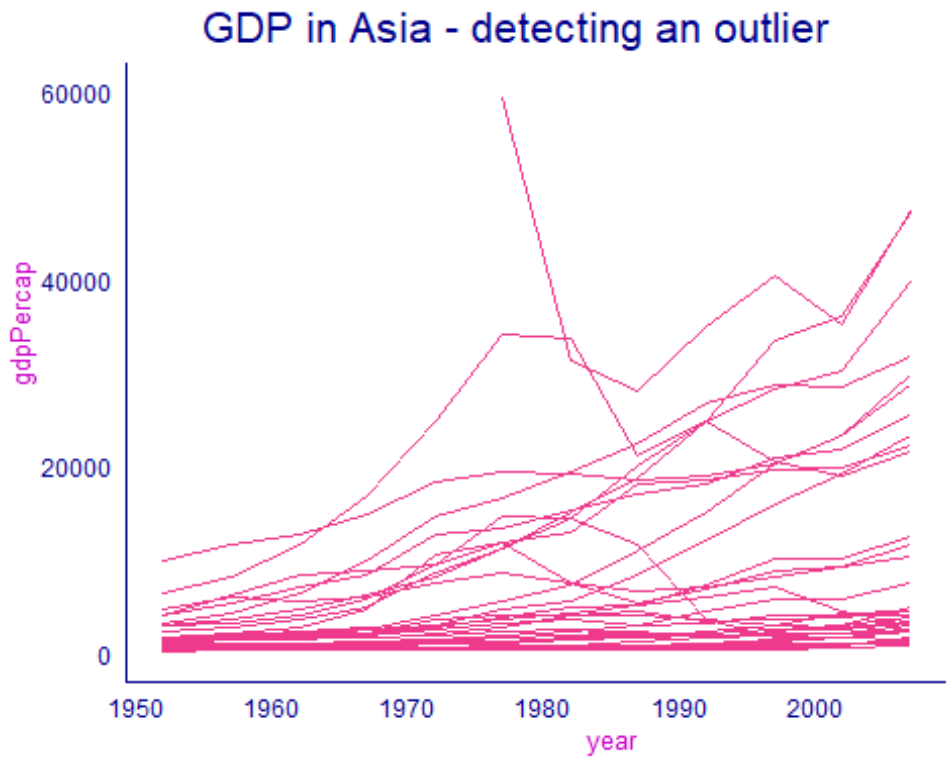


#Exercise 11: Flow Analysis ##

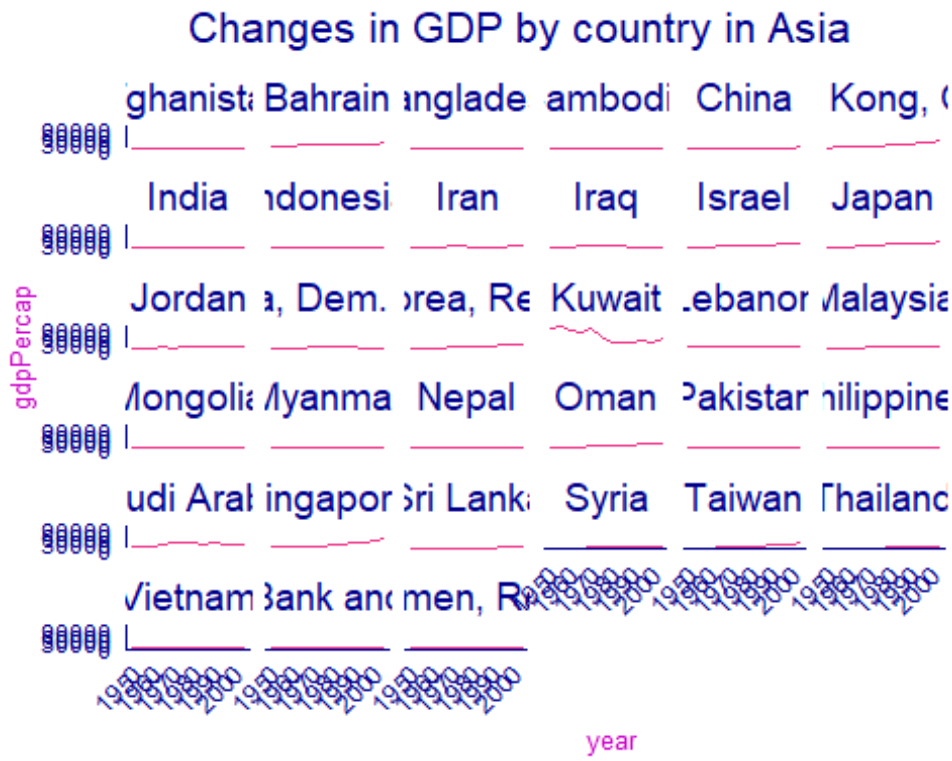
```
#gapminder dataset
#Checking on continents and y limits
ggplot(gapminder) +
  geom_line(aes (year, gdpPercap, group = country, color= continent), lwd =
0.3, show.legend = TRUE) +
  scale_color_manual(values=c("red1", "green1", "blue1", "brown1",
"purple1")) +
  labs(title = "GDP has increased worldwide") + ylim(0, 60000)
```



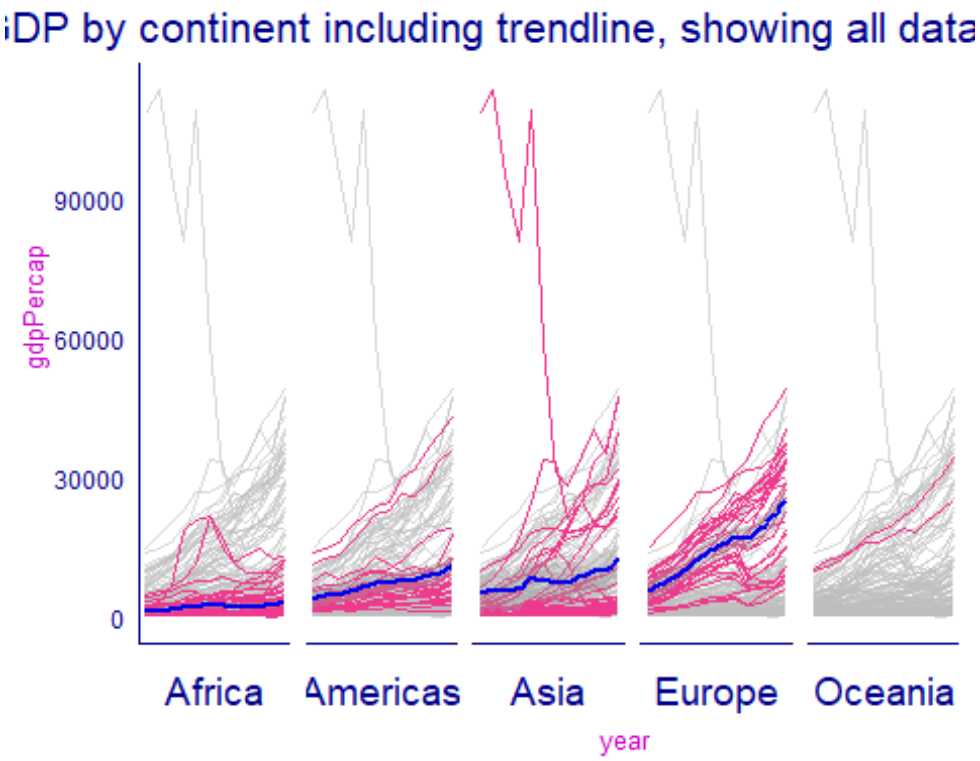

```
# Zooming in to see only Asia
ggplot(subset(gapminder, continent ==  "Asia")) +
  geom_line(aes(year, gdpPercap, group = country), color= "violetred2",
show.legend = FALSE) +
  labs(title = "GDP in Asia - detecting an outlier") + ylim(0, 60000)
```




```r
# Select only Asia in order to understand which country is the outlier
Asia <- dplyr::filter(gapminder, continent == "Asia")

ggplot(Asia, aes(year, gdpPercap)) +
  geom_line(color="violetred2") +
  facet_wrap(~country) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Changes in GDP by country in Asia")
```



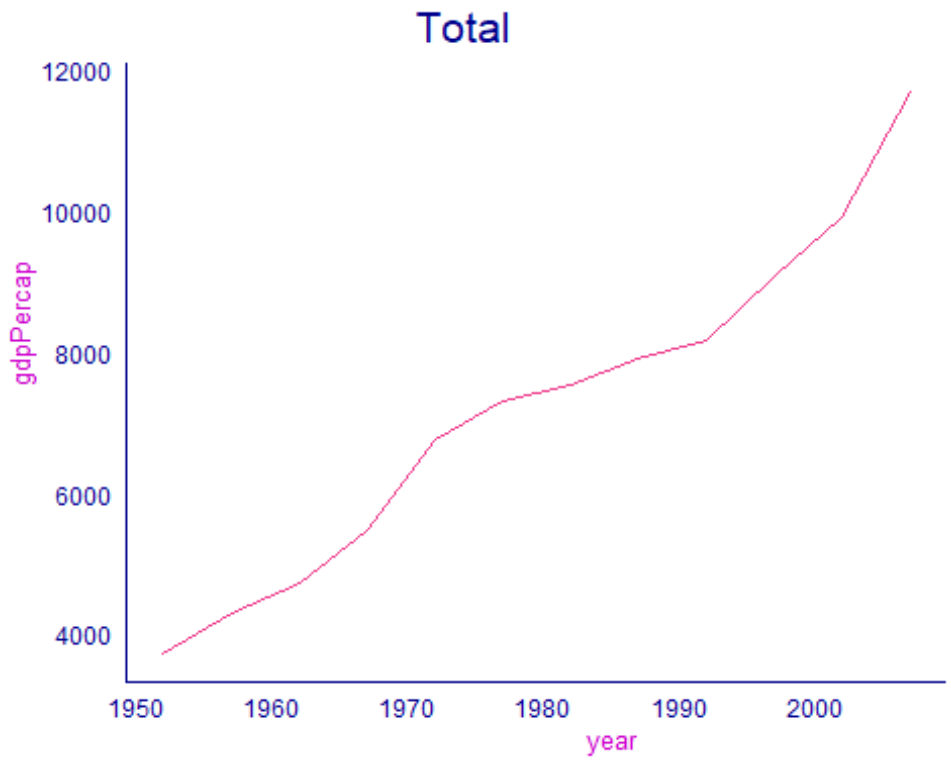

```
#We can even add all data in the background by setting the variable we do the facet with to zero
ggplot() +
  geom_line(data = transform(gapminder, continent = NULL), aes (year, gdpPercap, group = country), alpha = 0.6, lwd = 0.1, colour = "grey") +
  geom_line(data=gapminder, aes (year, gdpPercap, group = country), lwd = 0.3, show.legend = FALSE, color= "violetred2") +
  geom_smooth(data=gapminder, aes(year, gdpPercap, group = 1), lwd = 1, method = 'loess', span = 0.1, se = TRUE, color = "blue2") +
  facet_wrap(~ continent, ncol=5, strip.position = "bottom") +
  theme(strip.background = element_blank(), strip.placement = "outside") +
  theme(axis.text.x = element_blank()) +
  labs(title = "GDP by continent including trendline, showing all data in the back")
```




```
#Aggregating the data
gapminderavg<-aggregate(. ~year, data=gapminder, mean, na.rm=TRUE)
head(gapminderavg, n=10)
```

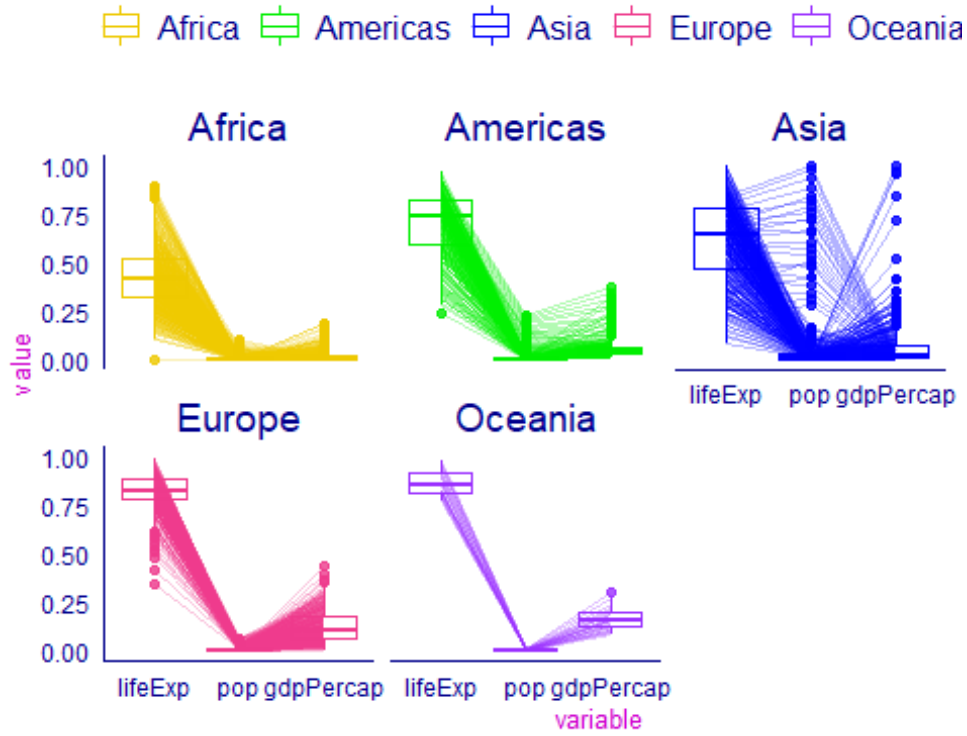
```
##    year country continent  lifeExp      pop gdpPercap
## 1  1952    71.5  2.330986 49.05762 16950402  3725.276
## 2  1957    71.5  2.330986 51.50740 18763413  4299.408
## 3  1962    71.5  2.330986 53.60925 20421007  4725.812
## 4  1967    71.5  2.330986 55.67829 22658298  5483.653
## 5  1972    71.5  2.330986 57.64739 25189980  6770.083
## 6  1977    71.5  2.330986 59.57016 27676379  7313.166
## 7  1982    71.5  2.330986 61.53320 30207302  7518.902
## 8  1987    71.5  2.330986 63.21261 33038573  7900.920
## 9  1992    71.5  2.330986 64.16034 35990917  8158.609
## 10 1997    71.5  2.330986 65.01468 38839468  9090.175
```

```
#Make a plot with the aggregated data
ggplot(gapminderavg) +
  geom_line(aes (year, gdpPercap), lwd = 0.3, show.legend = FALSE, color =
"violetred2") +
  labs(title = "Total")
```

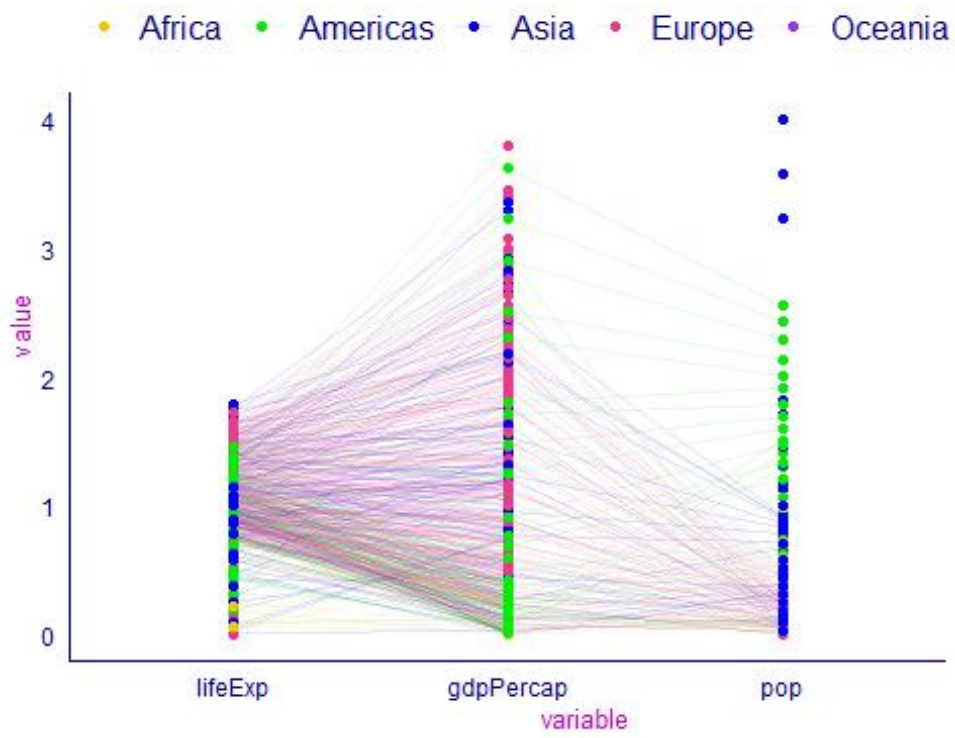


#Exercise 12: parallel coordinate plot ##################################

```
#gapminder dataset
#Simple chart, adding a color code
ggparcoord(gapminder, columns = 4:6, groupColumn = 2, alphaLines = 0.3,
scale="uniminmax", boxplot = TRUE) +
  scale_color_manual(values=c("gold2", "green2", "blue1", "violetred2",
"purple1")) +
  facet_wrap(. ~ continent)
```



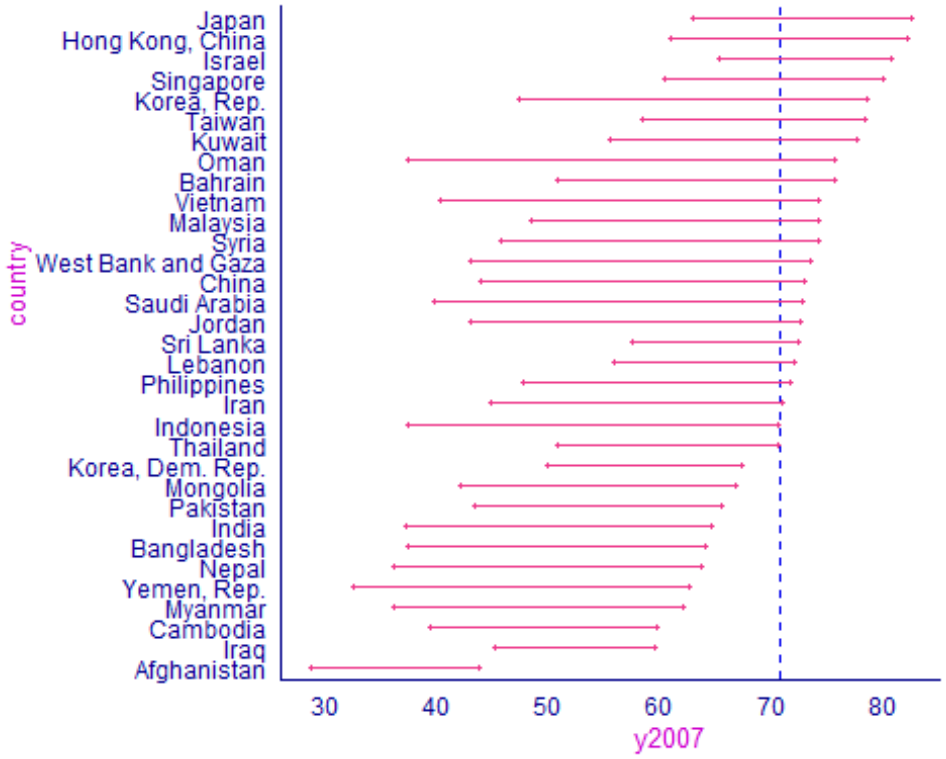

```
#Showing points, changing transparency and color
ggparcoord(gapminder, columns = 4:6, groupColumn = 2, order = "anyClass",
           showPoints = TRUE, alphaLines = 0.1)+
  scale_color_manual(values=c("gold2", "green2", "blue1", "violetred2",
"purple1"))+
  ylim(0, 4)
```



#Exercise 13: Dumbbell - Asia and 2007 ######################################

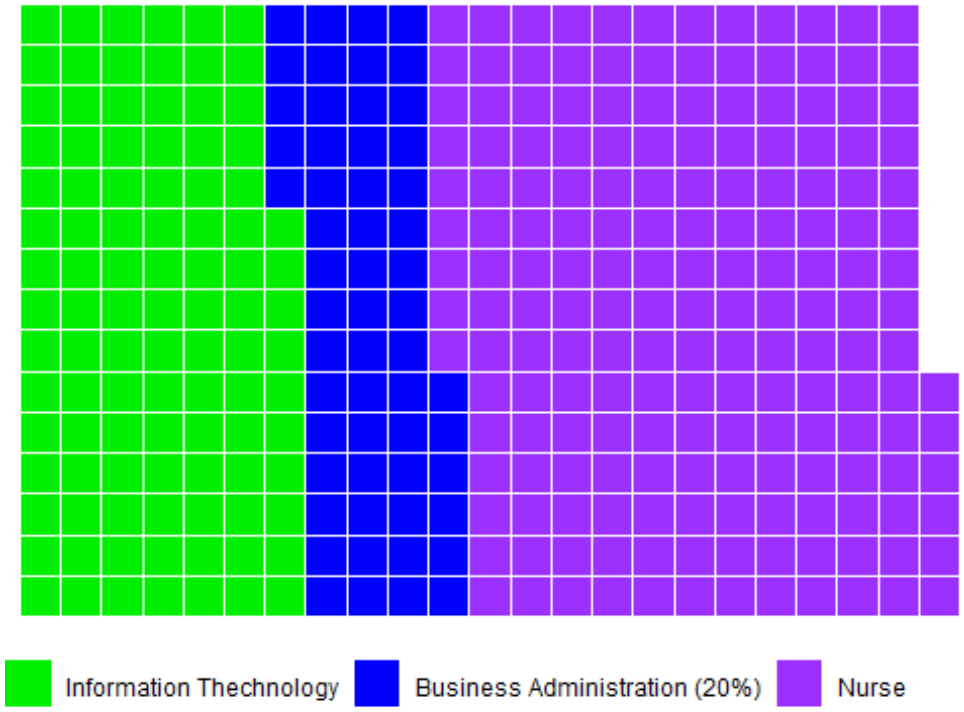
```
#gapminder dataset
#Creating a subsample
years <- filter(gapminder, year %in% c(1952, 2007)) %>% select(country,
continent, year, lifeExp)
#Convert data to wide format
years2 <- spread(years, year, lifeExp)
names(years2) <- c("country", "continent", "y1952", "y2007")
#Sorted by 2007
years3 <- arrange(years2, desc(y2007))
years3$country <- factor(years3$country, levels=rev(years3$country))

#Creating a subsample
asia2 <- filter(gapminder, continent == "Asia" & year %in% c(1952, 2007)) %>%
select(country, year, lifeExp)

#Convert data to wide format
asia3 <- spread(asia2, year, lifeExp)
names(asia3) <- c("country", "y1952", "y2007")

#Sorted by 2007
asia4 <- arrange(asia3, desc(y2007))
asia4$country <- factor(asia4$country, levels=rev(asia4$country))

#Create dumbbell plot now sorted
```

```
ggplot(asia4, aes(country, x = y2007, xend = y1952)) +
  geom_vline(xintercept=mean(asia4$y2007), color= "blue2", linetype =
"dashed") +
  geom_dumbbell(color="violetred2")
```

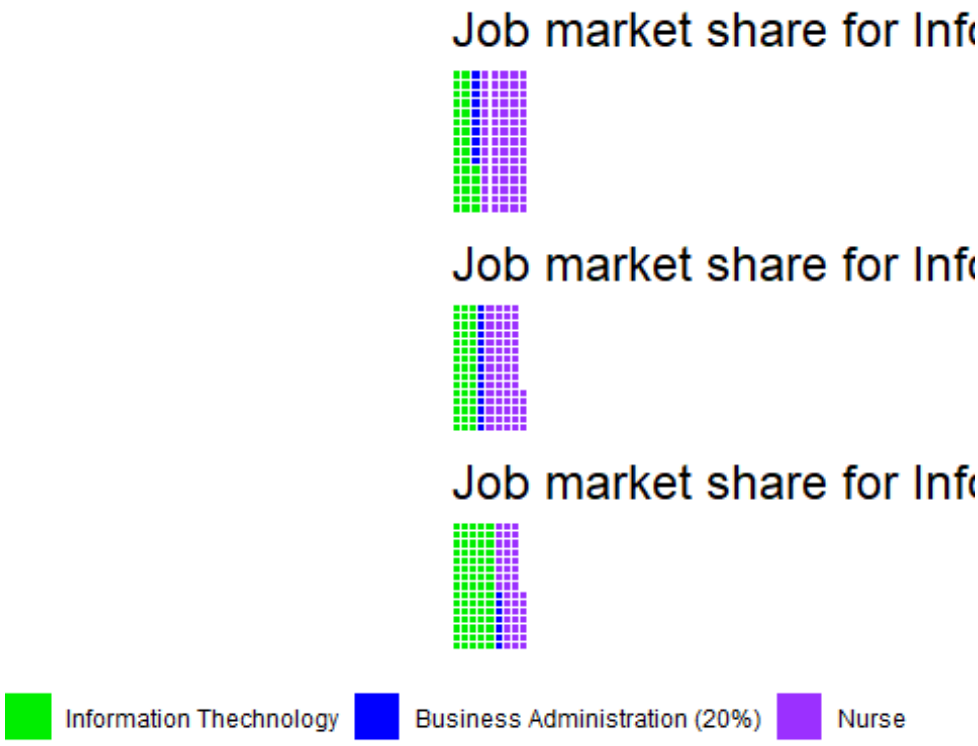


#Exercise 14: waffle - New data set and style ##################################

```
#Job market share as simple dataset created
Job <- c(`Information Thechnology`=100,`Business Administration (20%)`=
56,`Nurse`= 180)

#A simple waffle
waffle(Job, rows = 15, size = 0.2,
  colors = c("green2", "blue1", "purple1"), legend_pos = "bottom")
```

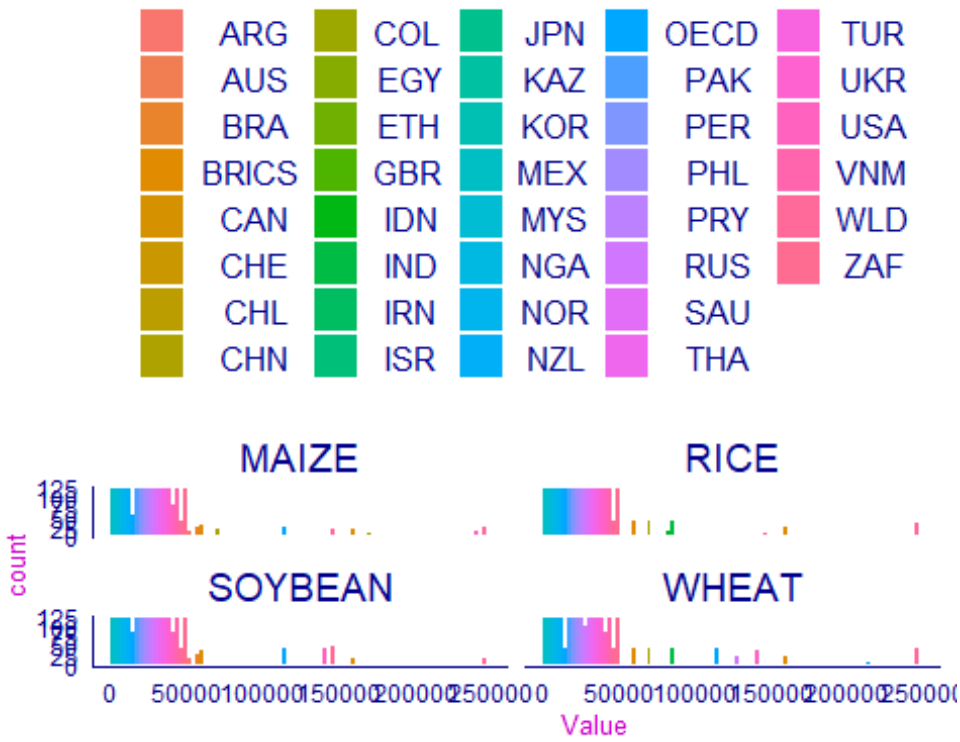



```
# Adding the legend only to one
iron(
  waffle(
    c(`Information Thechnology`=35,`Business Administration`= 10,`Nurse`=
75), rows = 15, size = 0.2,
    colors = c("green2", "blue1", "purple1"),
    keep = FALSE,
    title = "Job market share for Information Thechnology in 2018",
    legend='none'),

  waffle(
    c(`Information Thechnology`=45,`Business Administration (20%)`=
15,`Nurse`= 65), rows = 15, size = 0.2,
    colors = c("green2", "blue1", "purple1"),
    keep = FALSE,
    title = "Job market share for Information Thechnology in 2019",
    legend='none'),

  waffle(
    c(`Information Thechnology`=75,`Business Administration (20%)`=
7,`Nurse`= 45), rows = 15, size = 0.2,
    colors = c("green2", "blue1", "purple1"),
    keep = FALSE,
    title = "Job market share for Information Thechnology in 2020",
    legend_pos = "bottom"))
```

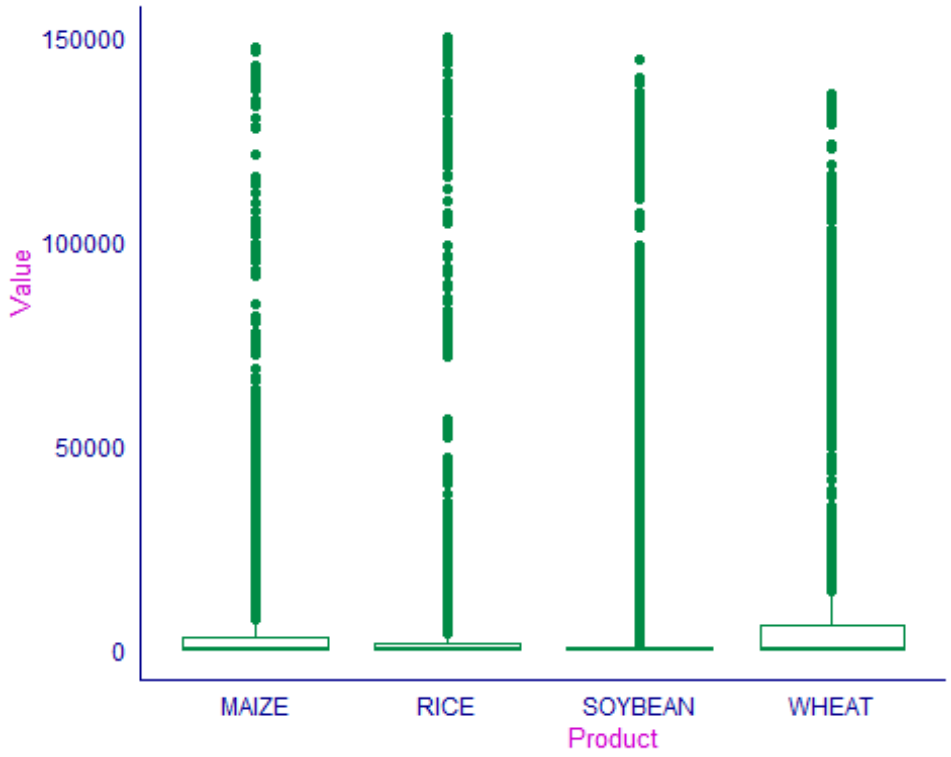

Job market share for Inf
Job market share for Inf
Job market share for Inf
Information Thechnology
Business Administration (20%)
Nurse

#Part 2: analyzing my selected data ####################################

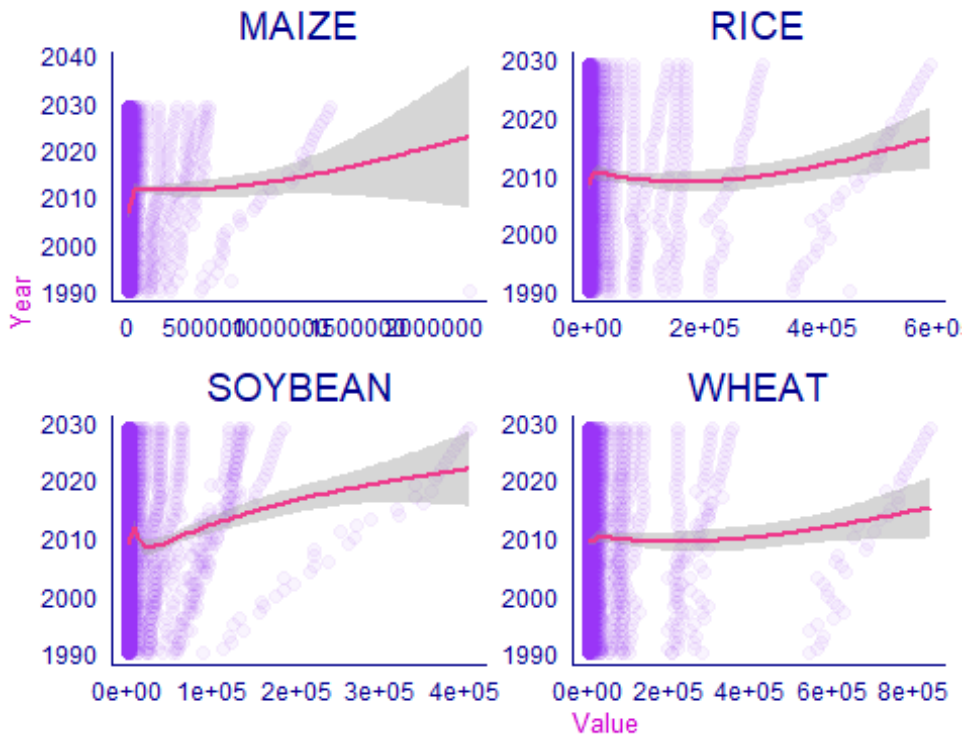
```
#read Agri Dataset
Agri <- read_csv("C:/Users/hamed/Desktop/Data visualization assignment
1/agriculture.csv")

##
## -- Column specification --------------------------------------------------
------
## cols(
##   LOCATION = col_character(),
##   INDICATOR = col_character(),
##   SUBJECT = col_character(),
##   MEASURE = col_character(),
##   FREQUENCY = col_character(),
##   TIME = col_double(),
##   Value = col_double(),
##   `Flag Codes` = col_logical()
## )

#Data preparation
Agri2 <- Agri %<>%
  as_tibble()%>%
  select(
    Country=matches("LOCATION"),
    Product=matches("SUBJECT"),
    Year=matches("TIME"),
    Value=matches("Value"),
  )%>%
  print()

## # A tibble: 18,240 x 4
##    Country Product  Year Value
##    <chr>   <chr>   <dbl> <dbl>
##  1 AUS     RICE     1990  8.32
##  2 AUS     RICE     1991  8.40
##  3 AUS     RICE     1992  8.09
##  4 AUS     RICE     1993  8.34
##  5 AUS     RICE     1994  8.54
##  6 AUS     RICE     1995  7.05
##  7 AUS     RICE     1996  8.26
##  8 AUS     RICE     1997  9.01
##  9 AUS     RICE     1998  9.20
## 10 AUS     RICE     1999  8.28
## # ... with 18,230 more rows

#Visualizing value of countries crop productions separated by products
ggplot(Agri2, aes(Value, fill = Country)) +
  geom_histogram(position = "dodge", binwidth = 100000) +
  facet_wrap(. ~ Product)+ xlim(0,250000)
```



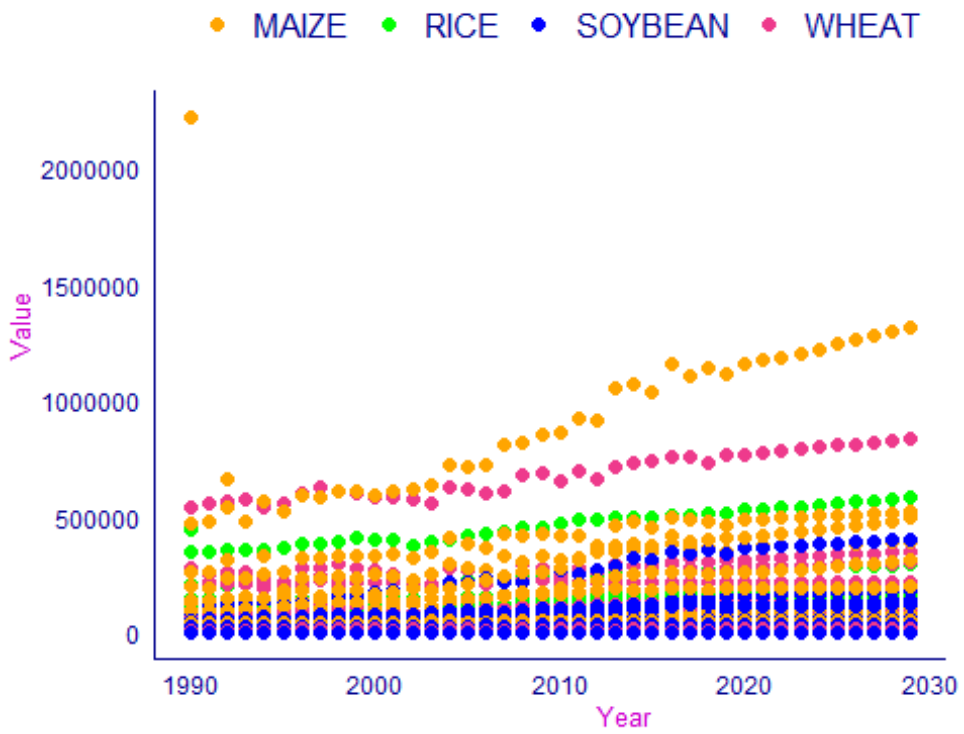

```
#Visualizing value of world crop productions for each different product
(boxplot)
ggplot(Agri2, aes(Product, Value)) +
  geom_boxplot(colour="springgreen4") + ylim(0,150000)
```



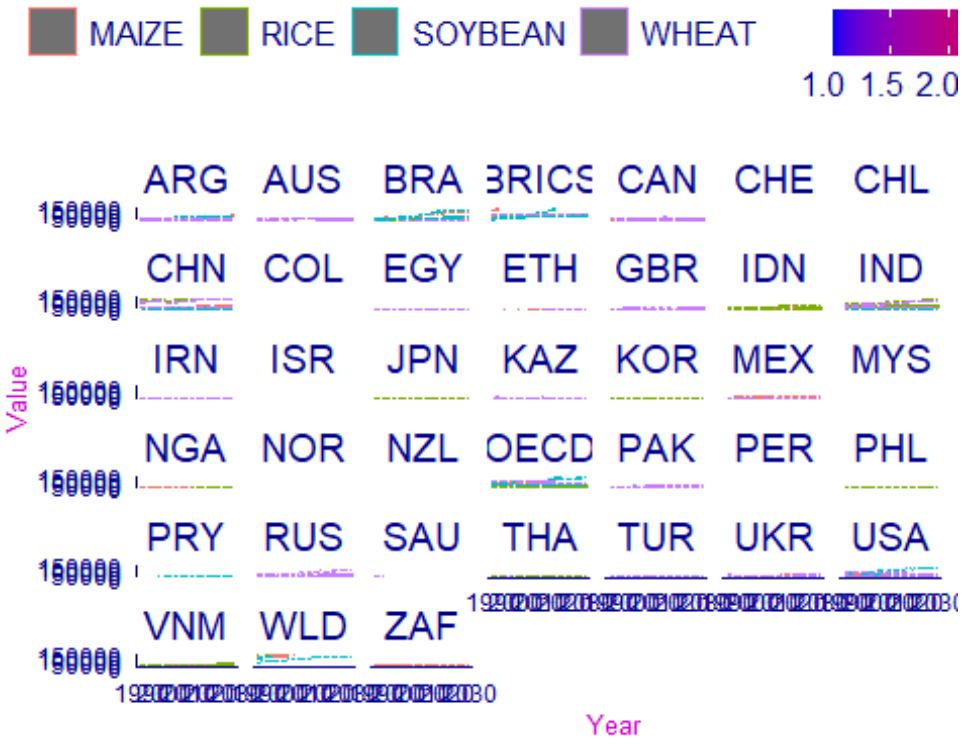

```
#Visualizing trend of world crop productions value by products
ggplot(Agri2, aes(x=Value, y=Year)) +
  geom_point(color="purple2", size=2, alpha=0.05)+
  facet_wrap( ~ Product, ncol=2, scales = "free") +
  stat_smooth(color="violetred2")
```




```r
#Visualizing of world crop productions value by products
ggplot(Agri2, aes(x=Year, y=Value, colour=Product)) +
  geom_quasirandom(size=2, alpha=1, groupOnX=FALSE) +
  scale_colour_manual(values=c("orange", "green1", "blue1", "violetred2"))
```




```
#Visualizing value of countries crop productions separated by countries
ggplot(Agri2, aes(x=Year, y=Value, colour=Product)) +
  geom_bin2d(bins = 50, alpha = 0.7)+
  scale_fill_gradient(low="blue2", high="red2")+ facet_wrap(. ~ Country) +
ylim(0, 150000)
```


MAIZE
RICE
SOYBEAN
WHEAT
1.0 1.5 2.0
ARG
AUS
BRA
BRICS
CAN
CHE
CHL
CHN
COL
EGY
ETH
GBR
IDN
IND
IRN
ISR
JPN
KAZ
KOR
MEX
MYS
NGA
NOR
NZL
OECD
PAK
PER
PHL
PRY
RUS
SAU
THA
TUR
UKR
USA
VNM
WLD
ZAF
Value
Year