How to make plots in R using ggpubr

Michail Belias

December 6, 2018

Introduction

The ggpubr is a R package that helps you create basic beautiful ggplot2-based graphs.

What is ggpubr:

- Wrapper around the ggplot2 package for beginners in R programming.
- Helps researchers, with basic R programming skills, to create easily publication-ready plots.
- Gives the possibility to add p-values and significance levels to plots.
- Makes it easy to arrange and annotate multiple plots on the same page.
- Makes it easy to change grahical parameters such as colors and labels.
- Is still a ggplot2 object...
  - Therefore, it can be further manipulated as a ggplot object

Generate some data for descriptive statistics

```
Sigma= matrix(c(20,15,15,20), 2)
Sigma2= matrix(c(15,10,10,15), 2)



df =  data.frame(mvrnorm(1000,c(180,80) , Sigma = Sigma) ,
df2 =  data.frame(mvrnorm(400,c(170,65) , Sigma = Sigma2) ,

df = rbind(df, df2)
names(df) = c("Height","Weight","Gender")
```

1. Distribution plots

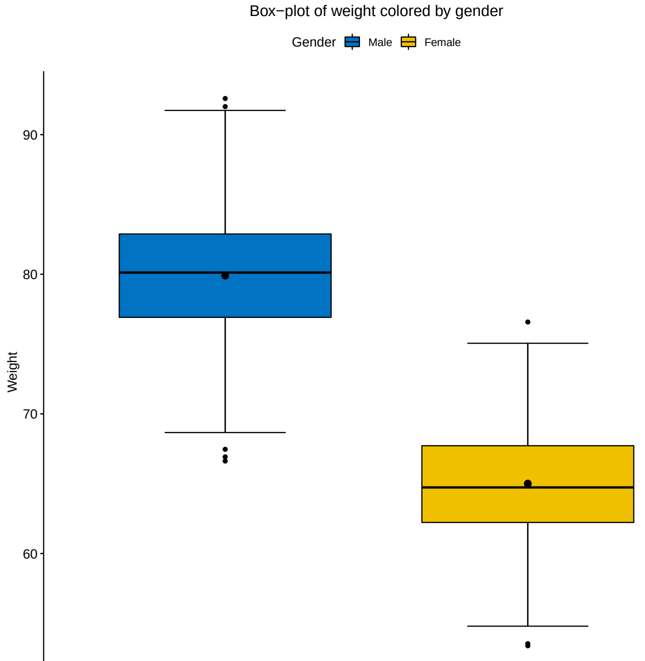
Under this section we consider

- Box Plots
- Violin + Boxplot
- Dot + Box Plot
- Histograms
- Density Plots

Boxplot code

```
gg<- ggboxplot(df ,
            x = "Gender",
            y = "Weight", # variable to be plotted
            color = "black", # paint the borders by Gender
            fill = "Gender", # fill the boxes with color
            title = "Box-plot of weight colored by gender",
            palette = "jco", # use the jco palette
            add = "mean", # or median
            bxp.errorbar = T  # adds the error bars of boxp
            ) +
  theme(plot.title = element_text(hjust = 0.5))
```

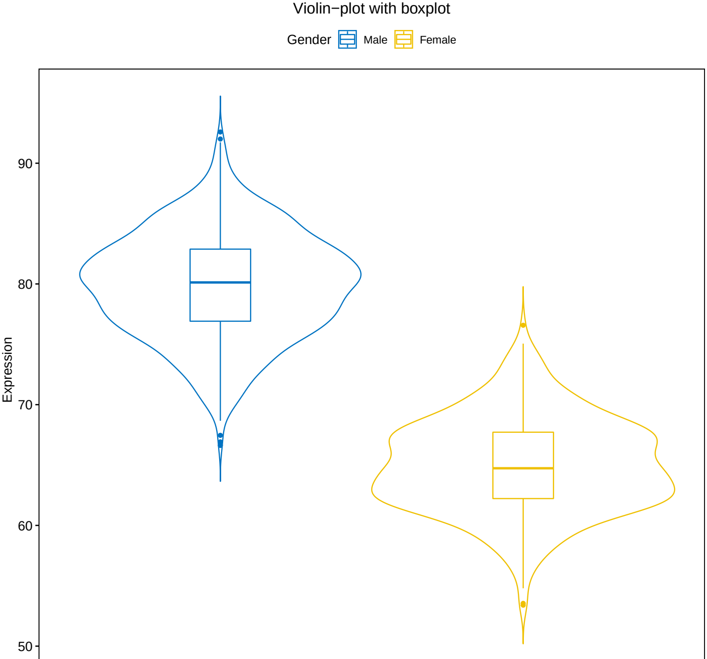
Boxplot


Box−plot of weight colored by gender
Gender
Male
Female
Weight
90
80
70
60

Violin plot with boxplot code

```r
gg <-ggviolin(df ,
            x = "Gender",
            y = "Weight", # variable to be plotted
          combine = TRUE, title="Violin-plot with boxplot",
          color = "Gender", palette = "jco",
          ylab = "Expression",
          add = "boxplot")+
  theme(plot.title = element_text(hjust = 0.5))
```

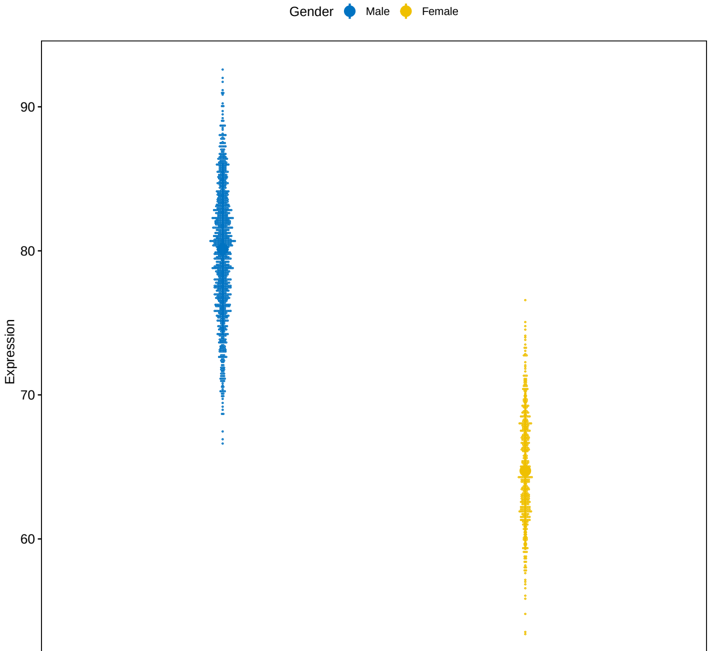
Violin plot with boxplot



Dotplot code

```
gg<- ggdotplot(df ,
            x = "Gender",
            y = "Weight", # variable to be plotted
            combine = TRUE,
            color = "Gender",
            palette = "jco",
            fill = "white",
            binwidth = 0.1,
            ylab = "Expression",
            add = "median_iqr",
            add.params = list(size = 0.9)
          )
```

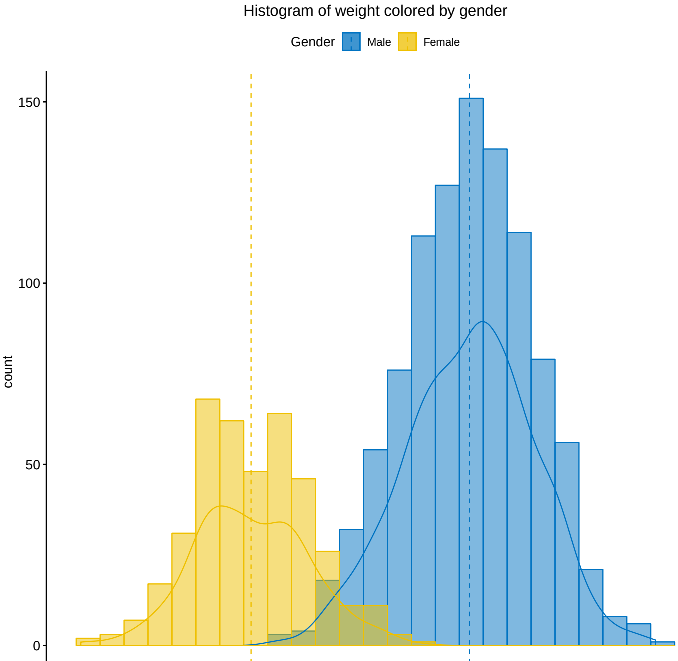
Dotplot



Histogram code

```
gg<- gghistogram(df,
            x= "Weight" ,   # variable to be plotted
            y= "..count..", # or "..density.."
            color = "Gender", # paint the borders by Gende
            fill = "Gender", # fill the bars with color
            bins = 25 , # control how many bars will the h
            title = "Histogram of weight colored by gender"
            palette = "jco", # use the jco palette
            add = "mean",
            add_density = T
            ) +   theme(plot.title = element_text(hjust =
```

Histogram plot

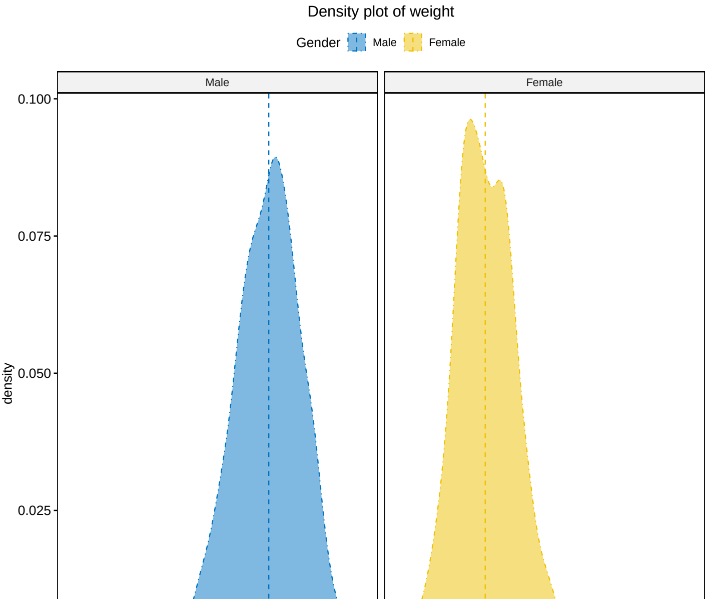


Density plot code

```r
gg <-  ggdensity(df,
          x = "Weight",
          fill = "Gender",
          palette = "jco",
          adjust = 3,
          title = "Density plot of weight",
          linetype = "dotdash",color = "Gender",
          facet.by = "Gender",add = "mean" ) +
  theme(plot.title = element_text(hjust = 0.5))
```

Density plot

```
plot(gg)
```



2.Correlation plots

Under this section we consider

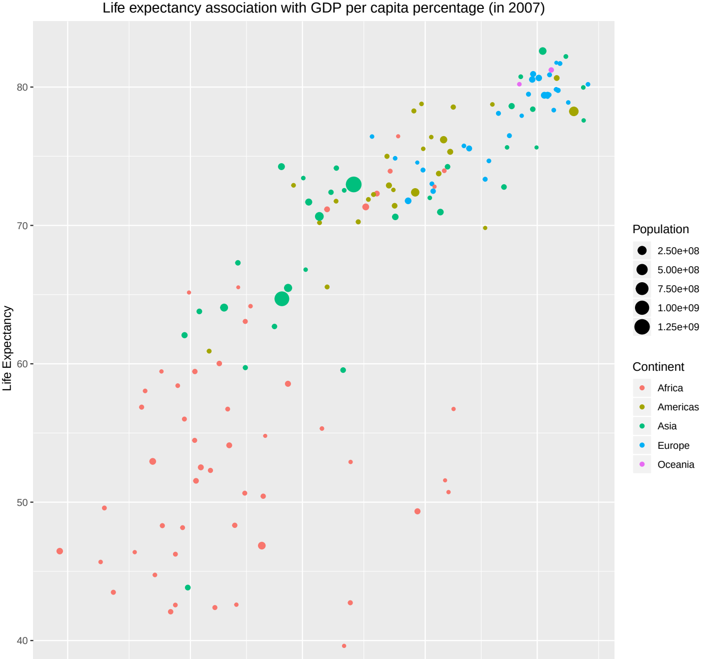
- Scatterplots
- Jitter Plots
- Counts Chart
- Bubble Plot
- Marginal Histograms / Boxplot

Scatterplots Bubble plot code

```
gapminder = gapminder
# Scatterplot
names(gapminder) =  c("Country","Continent","Year","Life_Ex
                      "Population","GDP_per_capita_percenta

gg = gapminder%>%
    filter(Year %in% "2007")%>%
ggplot( aes(GDP_per_capita_percentage, Life_Expectancy,size
            color = Continent)) + # This is then main plot
  ggtitle("Life expectancy association with GDP per capita
  # a new title
  theme(plot.title = element_text(hjust = 0.5))+
    geom_point() + # insert the points of the parameters us
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("GDP per capita") + # Change the label of X-axis
  ylab("Life Expectancy")+ # Change the label of Y-axis
  scale_x_log10() # log-Scale X values
```

Scatterplots Bubble plot

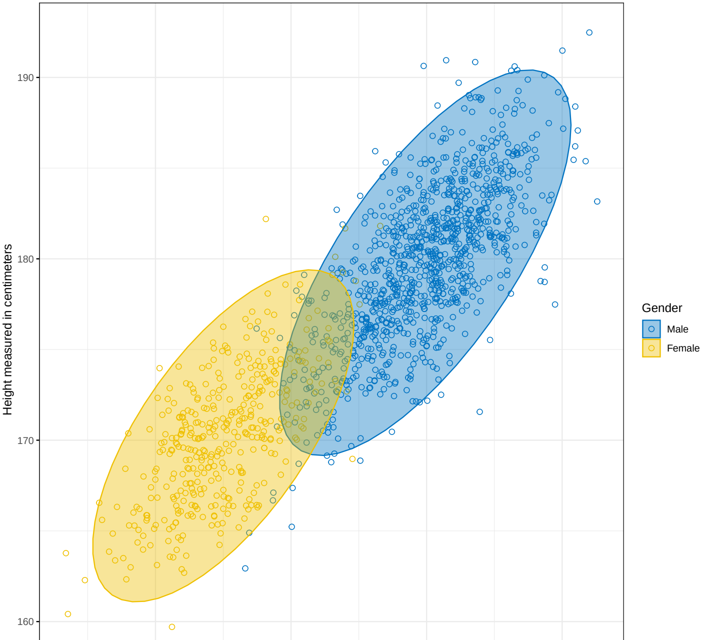


Scatterplot code

```
# Scatterplot


g = ggscatter(df , x = "Weight",y = "Height",
              color = "Gender", palette = "jco",
              title = "Association of weight and height in
              xlab = "Weight measured in kilograms",
              ylab = "Height measured in centimeters",
              shape = 1,
              ggtheme = theme_bw(),
              ellipse = T,
              ellipse.alpha = 0.4,
              fill = "white"
              )
```

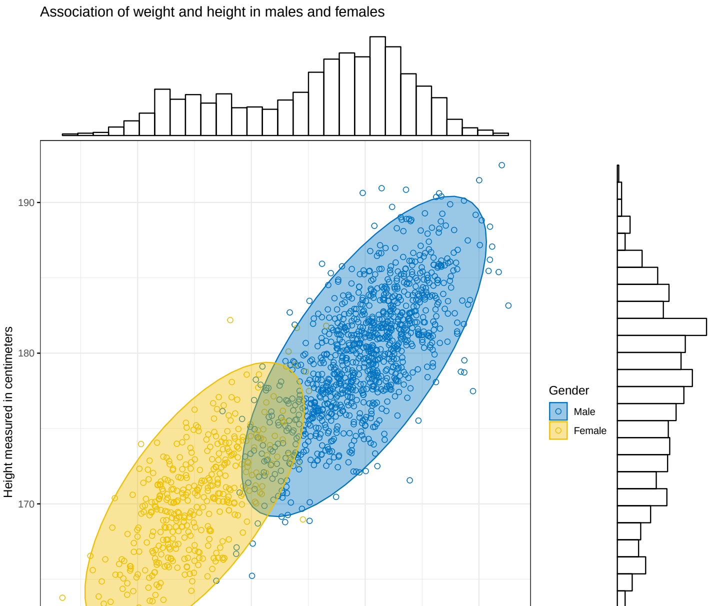
Scatterplot with ellipsis plot



Association of weight and height in males and females

Scatterplot with marginal histogram plot

```
ggMarginal(g, type = "histogram", fill="transparent")
```



Error plots

```
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
```

```
# Change error plot type and add mean points
ggerrorplot(ToothGrowth, x = "dose", y = "len",
            desc_stat = "mean_sd",
            error.plot = "errorbar",            # Change e
            add = "mean"                        # Add mean
            )
```

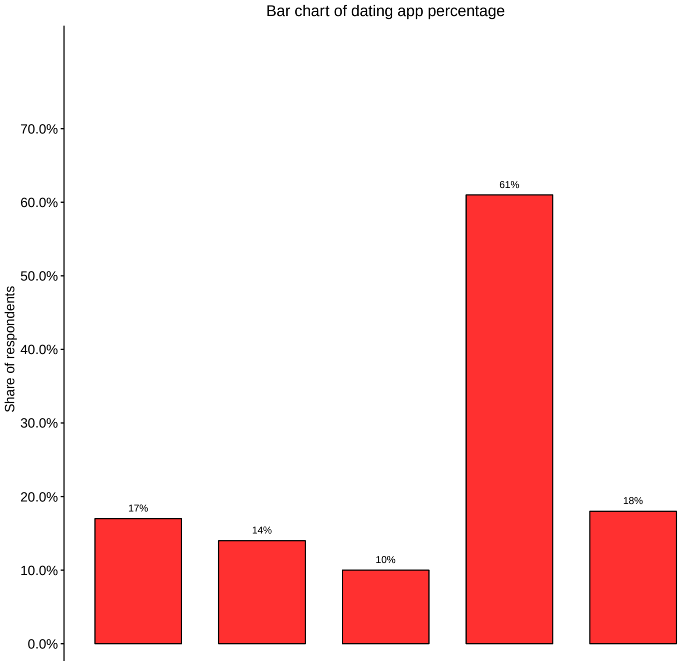
Ranking plots

- (Ordered) Bar Chart
- Lollipop Chart
- Dot Plot
- Slope Chart
- Dumbbell Plot

Bar Chart code

```
gg= ggbarplot(data = data1, #import  Data
          x="App",          # The X-value
          y="Proportion",   # The percentages
          fill="firebrick1",# The color of the bars
          xlab = "",        # label of X-axis
          title = "Bar chart of dating app percentage", #
          ylim = c(0,0.80), # Increase the ylimits
          ylab = "Share of respondents",
          order = c("Happn","Lexa","Paiq","Tinder","Badoo"
  theme(plot.title = element_text(hjust = 0.5))+
  scale_y_continuous(breaks=seq(0,0.7,0.1),
                     labels=scales::percent)+
  geom_text(aes(label=paste(Proportion*100,"%",sep = "")),
```

Bar Chart



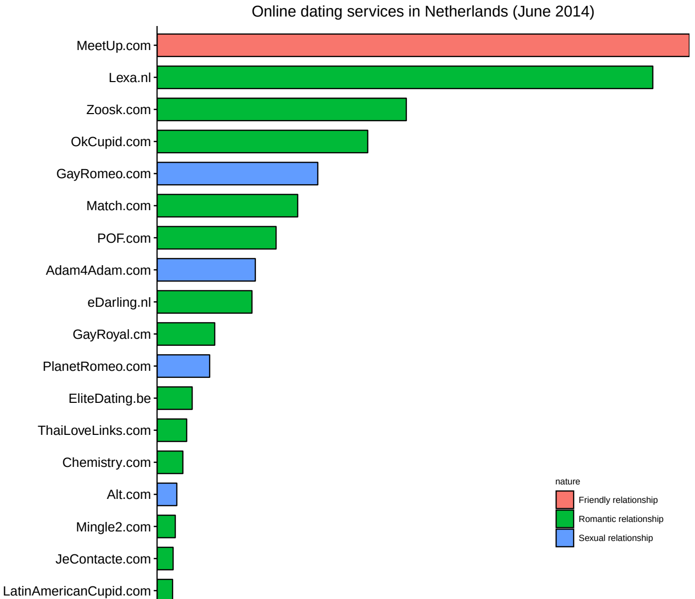
Bar chart of dating app percentage
Share of respondents
70.0%
60.0%
50.0%
40.0%
30.0%
20.0%
10.0%
0.0%
17%
14%
10%
61%
18%

Ordered bar chart code

```
gg= data2%>%
  arrange(desc(-freq))%>%
  ggbarplot( #import  Data
          x="website",           # The X-value
          y="freq",   # The percentages
          fill="nature",# The color of the bars
          xlab = "",        # label of X-axis
          title = "Online dating services in Netherlands (J
          ylim = c(0,300000), # Increase the ylimits
          ylab = "Share of respondents")+ rotate() +
  scale_y_continuous(breaks=seq(0,300000,50000),expand=c(0
    theme(legend.position =c(0.85,0.25),
          legend.title=element_text(size=8),
          legend.text=element_text(size=8),
          plot.title = element_text(hjust = 0.5))
```

Ordered bar chart

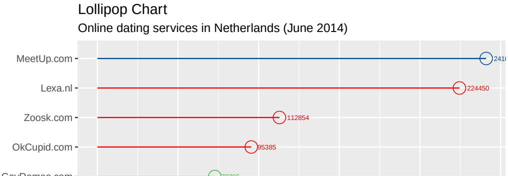
```
plot(gg)
```



Lollipop Chart code

```
# Plot
gg= ggplot(data2,aes(x=reorder(website,freq),
                     y=freq, color=nature,
                     fill = nature)) +
  geom_point(size=5,shape = 1) +
  geom_segment(aes(x=website, xend=website, y=0, yend=freq
  labs(title="Lollipop Chart",
       subtitle="Online dating services in Netherlands (Ju
       caption="source: Alexa.com") + scale_color_lancet()
  theme(axis.text.x = element_text(angle=65, vjust=0.6))+
  geom_text(aes(label= freq),size=2,hjust=-0.35) + rotate(
```

```
plot(gg)
```


Lollipop Chart
Online dating services in Netherlands (June 2014)
MeetUp.com
Lexa.nl
Zoosk.com
OkCupid.com
GayRomeo.com