ggpubr

庄闪闪

2020/10/1

目录

1 简介

Hadley Wickham 撰写的ggplot2是出色且灵活的软件包，可用于 R 中的优雅数据可视化。但是，默认生成的绘图需要先进行一些格式化，然后才

能发送它们进行发布。此外，要自定义 ggplot，语法是不透明的，这对没有高级 R 编程技能的研究人员增加了难度。

“ ggpubr” 软件包提供了一些易于使用的功能，用于创建和自定义基于“ ggplot2” 的可发布出版物的图表。

该文章来自ggpubr: Publication Ready Plots

2 安装

- 从 CRAN 安装如下：

```
#install.packages("ggpubr")
```

- 从 GitHub 安装最新版本，如下所示：

```
#安装
#if(!require(devtools))install.packages("devtools")
# devtools :: install_github("kassambara / ggpubr")
```

3 分布

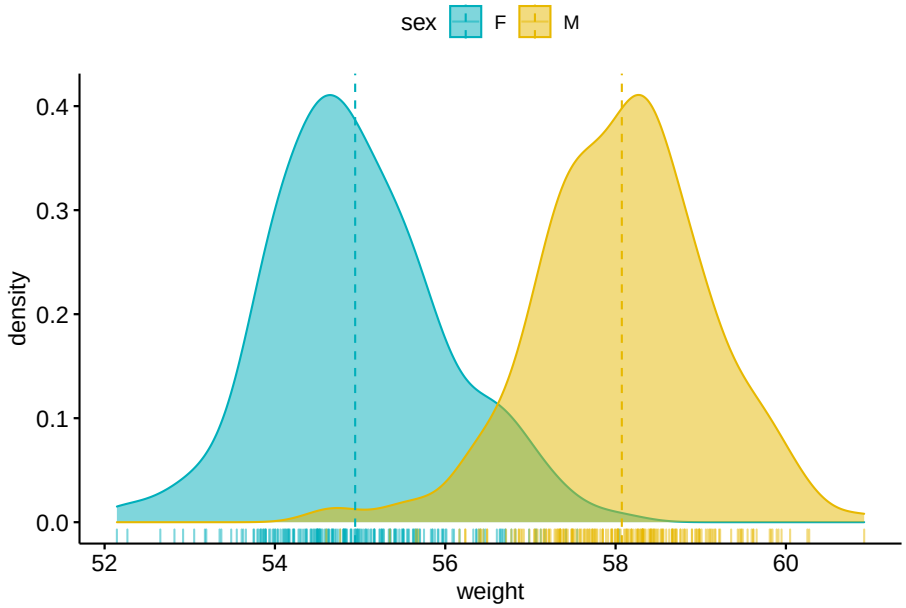
```
library(ggpubr)
#> Loading required package: ggplot2
#> Loading required package: magrittr
# Create some data format
# :::::::::::::::::::::::::::::::::::::::::::::::::
set.seed(1234)
wdata = data.frame(
   sex = factor(rep(c("F", "M"), each=200)),
   weight = c(rnorm(200, 55), rnorm(200, 58)))
head(wdata, 4)
```

```
##   sex weight
## 1   F  53.79
```

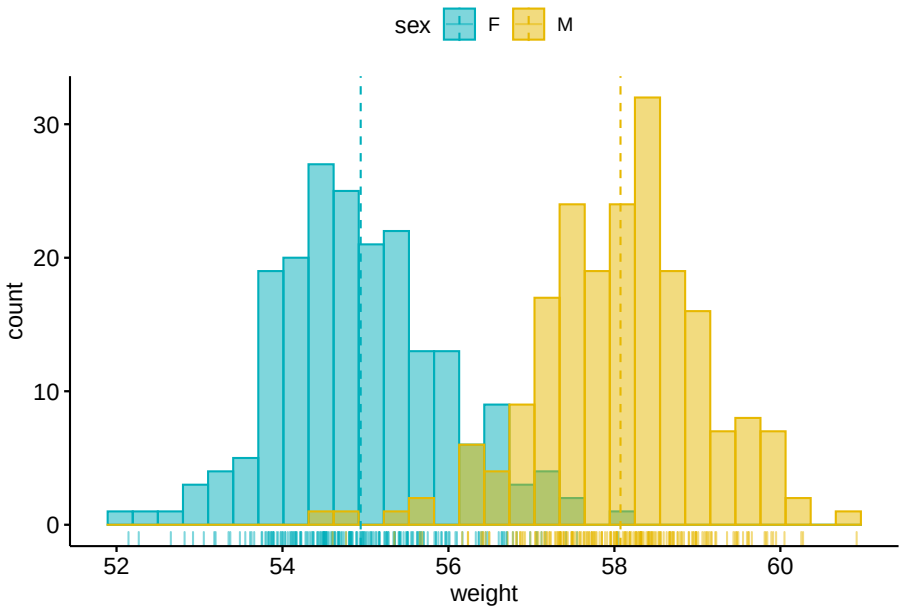
```
## 2   F  55.28
## 3   F  56.08
## 4   F  52.65
```

```
#>   sex   weight
#> 1   F 53.79293
#> 2   F 55.27743
#> 3   F 56.08444
#> 4   F 52.65430

# Density plot with mean lines and marginal rug
# :::::::::::::::::::::::::::::::::::::::::::::::::
# Change outline and fill colors by groups ("sex")
# Use custom palette
ggdensity(wdata, x = "weight",
   add = "mean", rug = TRUE,
   color = "sex", fill = "sex",
   palette = c("#00AFBB", "#E7B800"))
```



```
# Histogram plot with mean lines and marginal rug
# ::::::::::::::::::::::::::::::::::::::::::::::::
# Change outline and fill colors by groups ("sex")
# Use custom color palette
gghistogram(wdata, x = "weight",
   add = "mean", rug = TRUE,
   color = "sex", fill = "sex",
   palette = c("#00AFBB", "#E7B800"))
```



4 箱型图

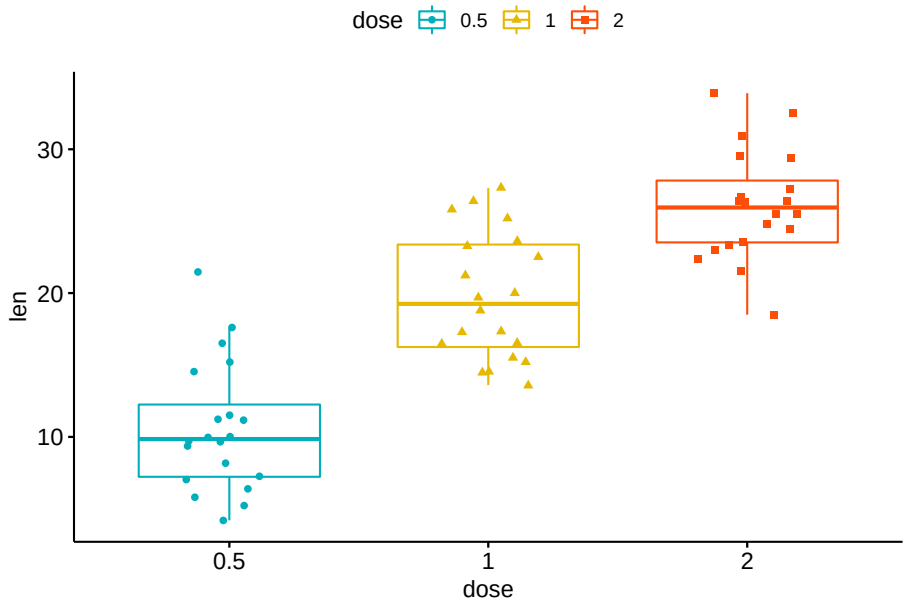
```
# Load data
data("ToothGrowth")
df <- ToothGrowth
head(df, 4)
```

```
##    len supp dose
## 1  4.2   VC  0.5
```

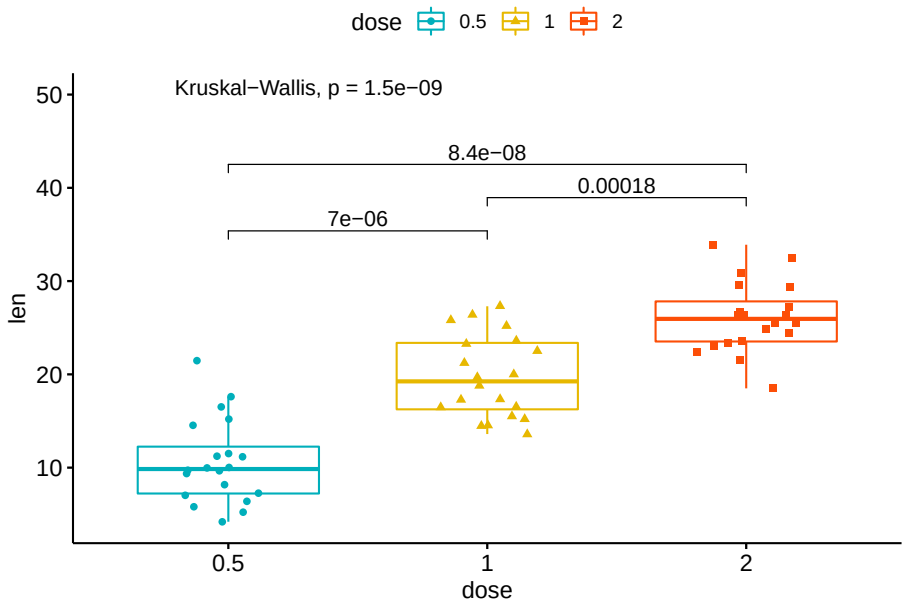
```
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
```

```
#>    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

# Box plots with jittered points
# ::::::::::::::::::::::::::::::::::::::::::::::::
# Change outline colors by groups: dose
# Use custom color palette
# Add jitter points and change the shape by groups
 p <- ggboxplot(df, x = "dose", y = "len",
                color = "dose", palette =c("#00AFBB", "#E7B800", "#FC4E07"),
                add = "jitter", shape = "dose")
 p
```

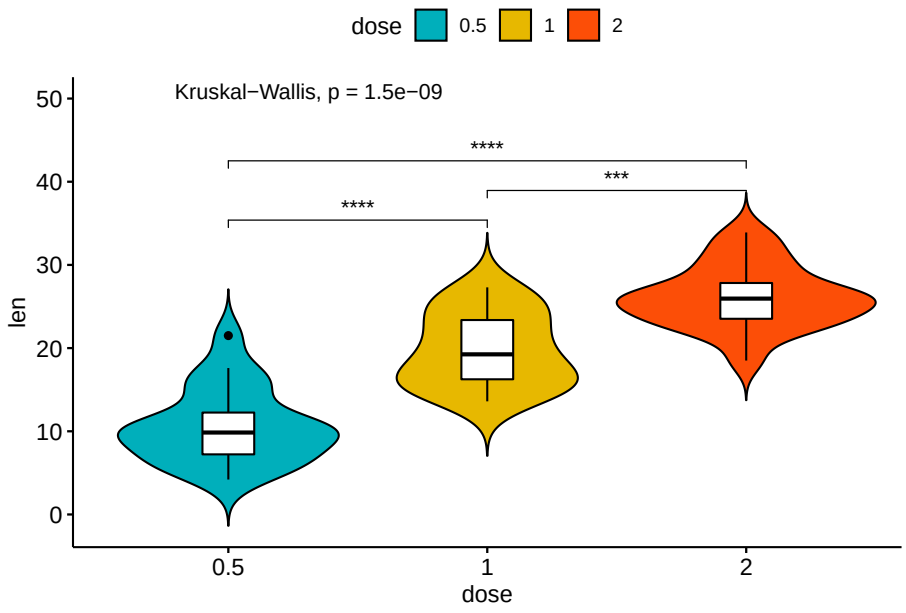


```
 # Add p-values comparing groups
 # Specify the comparisons you want
my_comparisons <- list( c("0.5", "1"), c("1", "2"), c("0.5", "2") )
p + stat_compare_means(comparisons = my_comparisons)+ # Add pairwise comparisons p-valu
  stat_compare_means(label.y = 50)                    # Add global p-value
```



5 小提琴图

```
# Violin plots with box plots inside
# :::::::::::::::::::::::::::::::::::::::::::::::::
# Change fill color by groups: dose
# add boxplot with white fill color
ggviolin(df, x = "dose", y = "len", fill = "dose",
         palette = c("#00AFBB", "#E7B800", "#FC4E07"),
         add = "boxplot", add.params = list(fill = "white"))+
  stat_compare_means(comparisons = my_comparisons, label = "p.signif")+ # Add significa
  stat_compare_means(label.y = 50)                                      # Add global th
```



6 条形图

6.1 数据集

```
# Load data
data("mtcars")
dfm <- mtcars
# Convert the cyl variable to a factor
dfm$cyl <- as.factor(dfm$cyl)
# Add the name colums
dfm$name <- rownames(dfm)
# Inspect the data
head(dfm[, c("name", "wt", "mpg", "cyl")])
```

```
##                                   name    wt  mpg cyl
## Mazda RX4                   Mazda RX4 2.620 21.0   6
## Mazda RX4 Wag           Mazda RX4 Wag 2.875 21.0   6
## Datsun 710                 Datsun 710 2.320 22.8   4
```

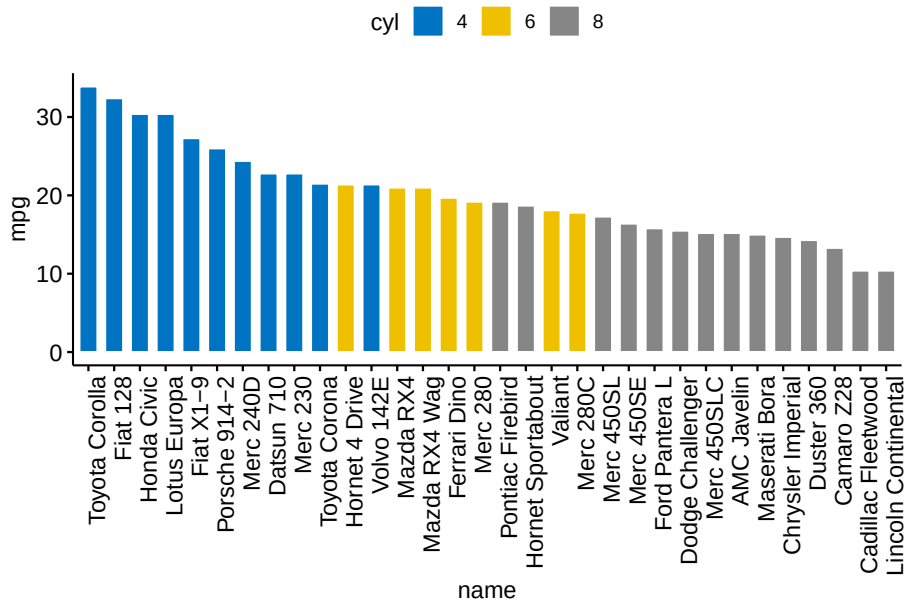
```
## Hornet 4 Drive       Hornet 4 Drive 3.215 21.4   6
## Hornet Sportabout Hornet Sportabout 3.440 18.7   8
## Valiant                     Valiant 3.460 18.1   6
```

```
#>                                name    wt  mpg cyl
#> Mazda RX4                 Mazda RX4 2.620 21.0   6
#> Mazda RX4 Wag         Mazda RX4 Wag 2.875 21.0   6
#> Datsun 710               Datsun 710 2.320 22.8   4
#> Hornet 4 Drive       Hornet 4 Drive 3.215 21.4   6
#> Hornet Sportabout Hornet Sportabout 3.440 18.7   8
#> Valiant                     Valiant 3.460 18.1   6
```

通过分组变量 “cyl” 改变填充颜色。排序将全局执行，而不是按组执行。

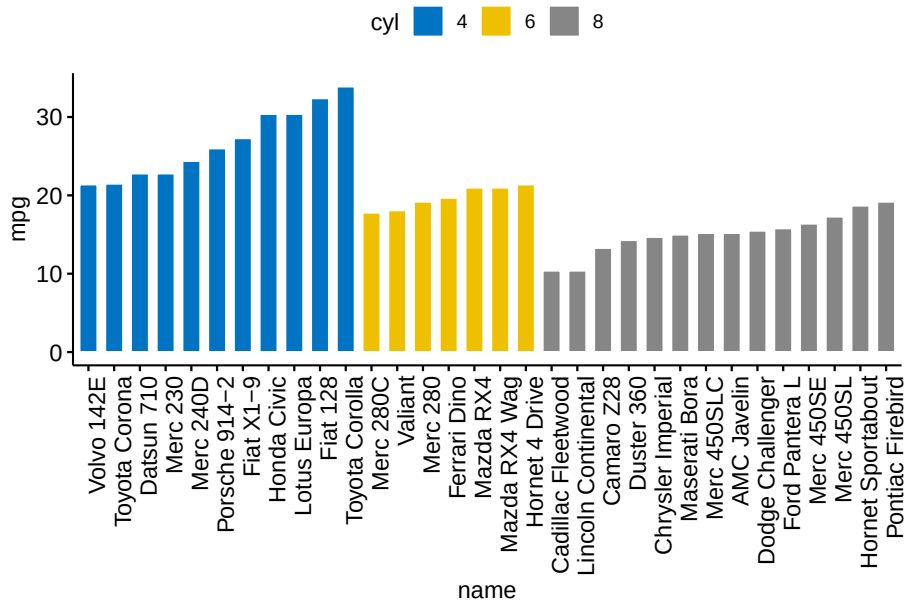
6.2 有序的条形图

```
ggbarplot(dfm, x = "name", y = "mpg",
          fill = "cyl",               # change fill color by cyl
          color = "white",            # Set bar border colors to white
          palette = "jco",            # jco journal color palett. see ?ggpar
          sort.val = "desc",          # Sort the value in dscending order
          sort.by.groups = FALSE,     # Don't sort inside each group
          x.text.angle = 90           # Rotate vertically x axis texts
          )
```



Sort bars inside each group. Use the argument sort.by.groups = TRUE.

```
ggbarplot(dfm, x = "name", y = "mpg",
          fill = "cyl",               # change fill color by cyl
          color = "white",            # Set bar border colors to white
          palette = "jco",            # jco journal color palett. see ?ggpar
          sort.val = "asc",           # Sort the value in dscending order
          sort.by.groups = TRUE,      # Sort inside each group
          x.text.angle = 90           # Rotate vertically x axis texts
          )
```



7 Deviation graphs

The deviation graph shows the deviation of quantitatives values to a reference value. In the R code below, we'll plot the mpg z-score from the mtcars dataset.

Calculate the z-score of the mpg data:

```
# Calculate the z-score of the mpg data
dfm$mpg_z <- (dfm$mpg -mean(dfm$mpg))/sd(dfm$mpg)
dfm$mpg_grp <- factor(ifelse(dfm$mpg_z < 0, "low", "high"),
                      levels = c("low", "high"))
# Inspect the data
head(dfm[, c("name", "wt", "mpg", "mpg_z", "mpg_grp", "cyl")])
```

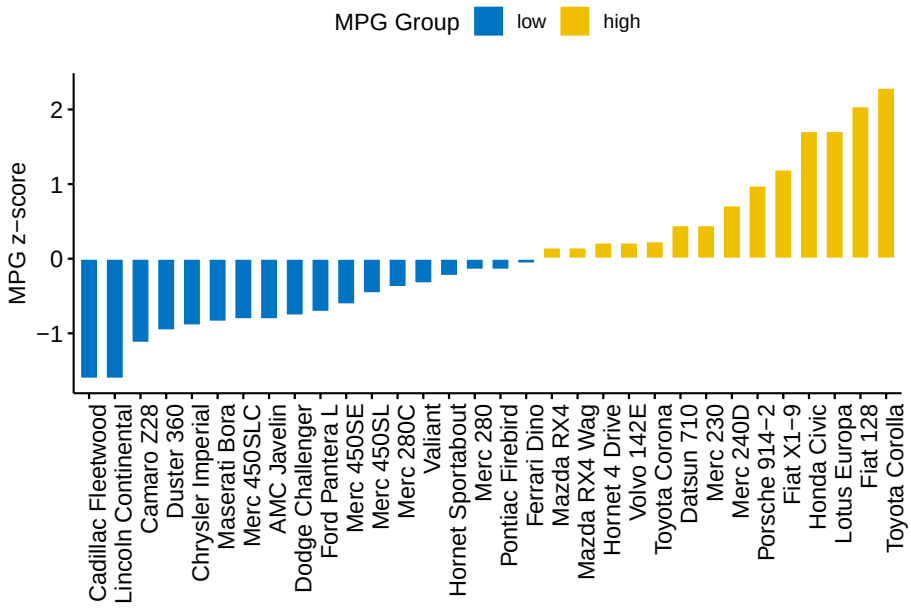
```
##                           name    wt  mpg   mpg_z mpg_grp cyl
## Mazda RX4                 Mazda RX4 2.620 21.0  0.1509    high   6
## Mazda RX4 Wag         Mazda RX4 Wag 2.875 21.0  0.1509    high   6
## Datsun 710               Datsun 710 2.320 22.8  0.4495    high   4
```

```
## Hornet 4 Drive       Hornet 4 Drive 3.215 21.4  0.2173    high   6
## Hornet Sportabout Hornet Sportabout 3.440 18.7 -0.2307     low   8
## Valiant                     Valiant 3.460 18.1 -0.3303     low   6
```

```
#>                                 name    wt  mpg      mpg_z mpg_grp cyl
#> Mazda RX4                 Mazda RX4 2.620 21.0  0.1508848    high   6
#> Mazda RX4 Wag         Mazda RX4 Wag 2.875 21.0  0.1508848    high   6
#> Datsun 710               Datsun 710 2.320 22.8  0.4495434    high   4
#> Hornet 4 Drive       Hornet 4 Drive 3.215 21.4  0.2172534    high   6
#> Hornet Sportabout Hornet Sportabout 3.440 18.7 -0.2307345     low   8
#> Valiant                     Valiant 3.460 18.1 -0.3302874     low   6
```

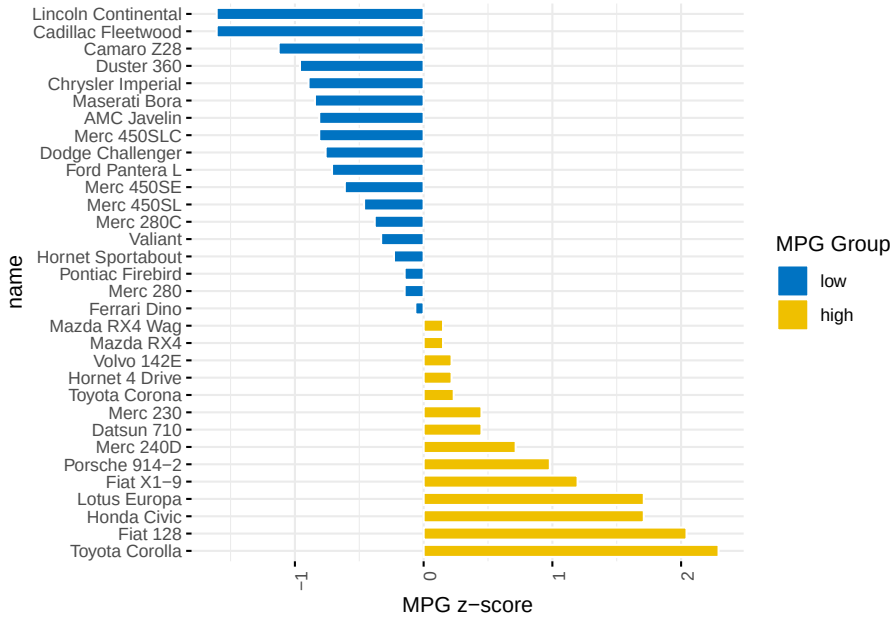
创建一个有序的 barplot，根据 mpg 级别着色:

```
ggbarplot(dfm, x = "name", y = "mpg_z",
          fill = "mpg_grp",           # change fill color by mpg_level
          color = "white",            # Set bar border colors to white
          palette = "jco",            # jco journal color palett. see ?ggpar
          sort.val = "asc",           # Sort the value in ascending order
          sort.by.groups = FALSE,     # Don't sort inside each group
          x.text.angle = 90,          # Rotate vertically x axis texts
          ylab = "MPG z-score",
          xlab = FALSE,
          legend.title = "MPG Group"
          )
```



Rotate the plot: use rotate = TRUE and sort.val = "desc"

```
ggbarplot(dfm, x = "name", y = "mpg_z",
          fill = "mpg_grp",           # change fill color by mpg_level
          color = "white",            # Set bar border colors to white
          palette = "jco",            # jco journal color palett. see ?ggpar
          sort.val = "desc",          # Sort the value in descending order
          sort.by.groups = FALSE,     # Don't sort inside each group
          x.text.angle = 90,          # Rotate vertically x axis texts
          ylab = "MPG z-score",
          legend.title = "MPG Group",
          rotate = TRUE,
          ggtheme = theme_minimal()
          )
```



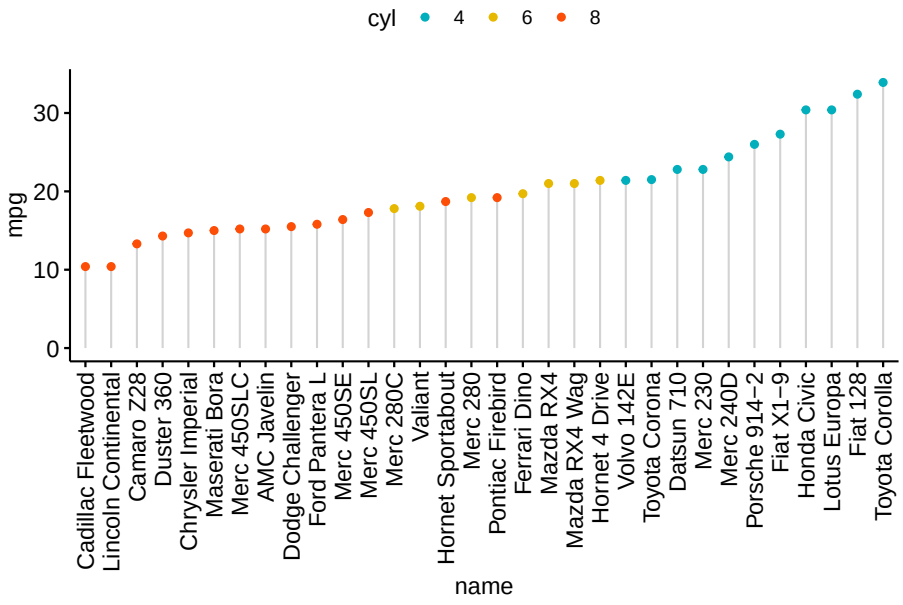
8 Dot charts

8.1 Lollipop chart

Lollipop chart is an alternative to bar plots, when you have a large set of values to visualize.

Lollipop chart colored by the grouping variable "cyl":

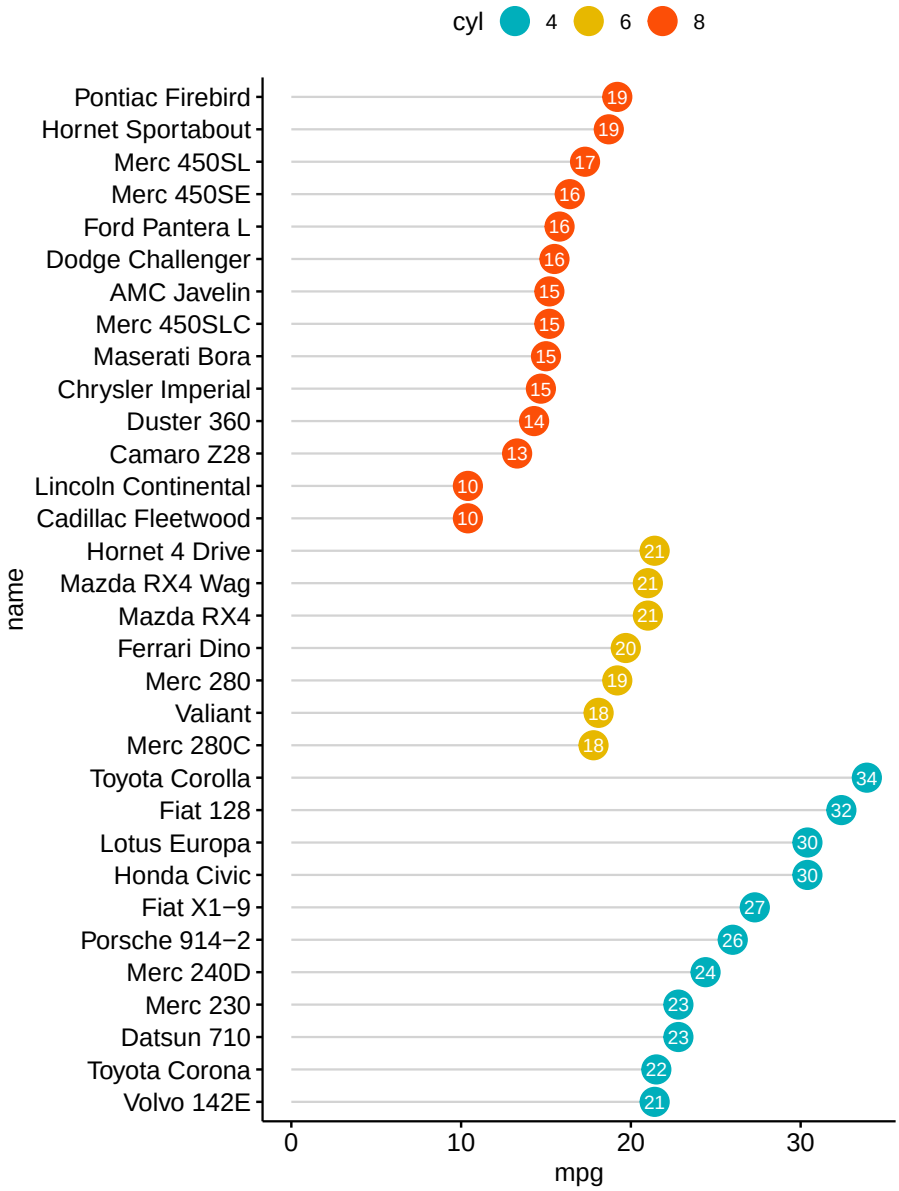
```
ggdotchart(dfm, x = "name", y = "mpg",
           color = "cyl",                                # Color by groups
           palette = c("#00AFBB", "#E7B800", "#FC4E07"), # Custom color palette
           sorting = "ascending",                        # Sort value in descending ord
           add = "segments",                             # Add segments from y = 0 to d
           ggtheme = theme_pubr()                        # ggplot2 theme
           )
```



- Sort in decending order. sorting = "descending".
- Rotate the plot vertically, using rotate = TRUE.
- Sort the mpg value inside each group by using group = "cyl".
- Set dot.size to 6.
- Add mpg values as label. label = "mpg" or label = round(dfm$mpg).

```
ggdotchart(dfm, x = "name", y = "mpg",
           color = "cyl",                                # Color by groups
           palette = c("#00AFBB", "#E7B800", "#FC4E07"), # Custom color palette
           sorting = "descending",                       # Sort value in descending ord
           add = "segments",                             # Add segments from y = 0 to d
           rotate = TRUE,                                # Rotate vertically
           group = "cyl",                                # Order by groups
           dot.size = 6,                                 # Large dot size
           label = round(dfm$mpg),                        # Add mpg values as dot label
           font.label = list(color = "white", size = 9,
```

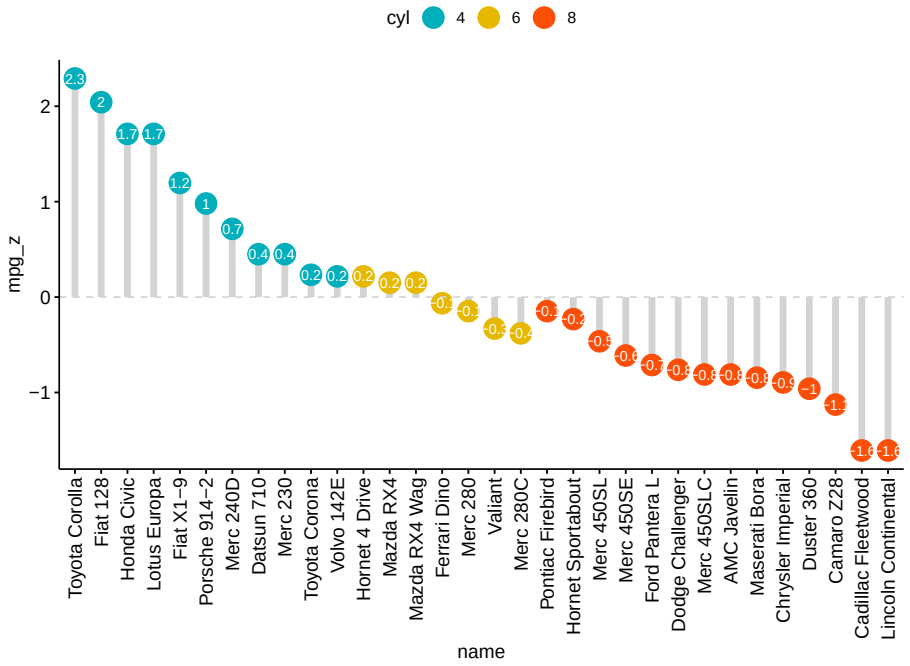
```
                                vjust = 0.5),                # Adjust label parameters
          ggtheme = theme_pubr()                             # ggplot2 theme
          )
```



Deviation graph:

- Use y = “mpg_z”
- Change segment color and size: add.params = list(color = “lightgray”, size = 2)

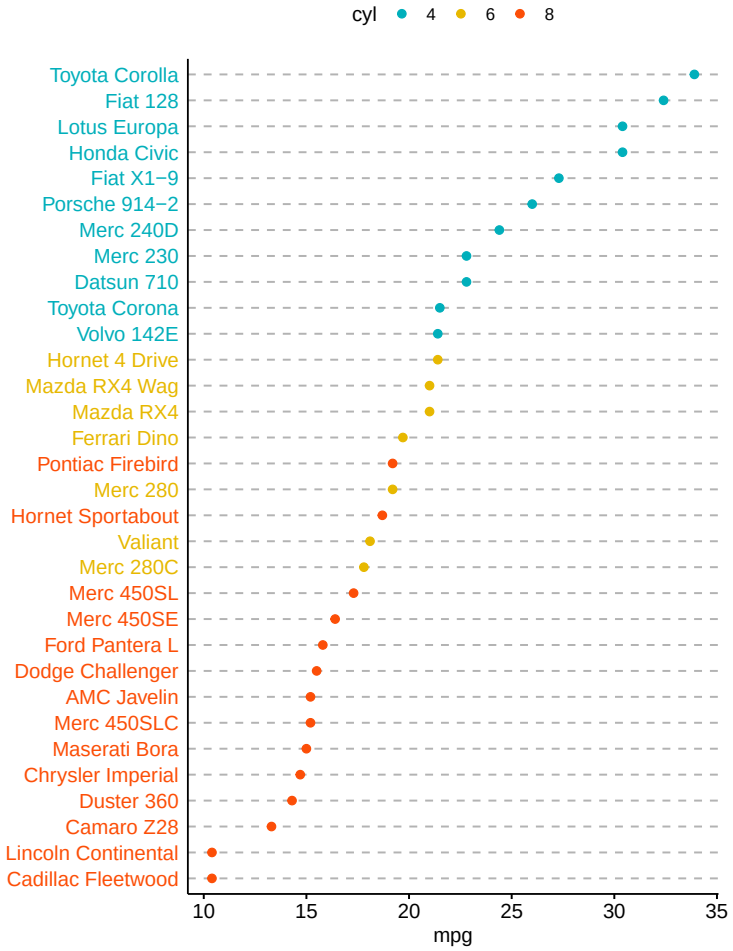
```
ggdotchart(dfm, x = "name", y = "mpg_z",
           color = "cyl",                                # Color by groups
           palette = c("#00AFBB", "#E7B800", "#FC4E07"), # Custom color palette
           sorting = "descending",                       # Sort value in descending ord
           add = "segments",                             # Add segments from y = 0 to d
           add.params = list(color = "lightgray", size = 2), # Change segment color and
           group = "cyl",                                # Order by groups
           dot.size = 6,                                 # Large dot size
           label = round(dfm$mpg_z,1),                        # Add mpg values as dot l
           font.label = list(color = "white", size = 9,
                             vjust = 0.5),               # Adjust label parameters
           ggtheme = theme_pubr()                        # ggplot2 theme
           )+
  geom_hline(yintercept = 0, linetype = 2, color = "lightgray")
```



9 Cleveland's dot plot

Color y text by groups. Use y.text.col = TRUE.

```
ggdotchart(dfm, x = "name", y = "mpg",
           color = "cyl",                                # Color by groups
           palette = c("#00AFBB", "#E7B800", "#FC4E07"), # Custom color palette
           sorting = "descending",                       # Sort value in descending ord
           rotate = TRUE,                                # Rotate vertically
           dot.size = 2,                                 # Large dot size
           y.text.col = TRUE,                            # Color y text by groups
           ggtheme = theme_pubr()                        # ggplot2 theme
           )+
  theme_cleveland()                                      # Add dashed grids
```


cyl 4 6 8
Toyota Corolla
Fiat 128
Lotus Europa
Honda Civic
Fiat X1−9
Porsche 914−2
Merc 240D
Merc 230
Datsun 710
Toyota Corona
Volvo 142E
Hornet 4 Drive
Mazda RX4 Wag
Mazda RX4
Ferrari Dino
Pontiac Firebird
Merc 280
Hornet Sportabout
Valiant
Merc 280C
Merc 450SL
Merc 450SE
Ford Pantera L
Dodge Challenger
AMC Javelin
Merc 450SLC
Maserati Bora
Chrysler Imperial
Duster 360
Camaro Z28
Lincoln Continental
Cadillac Fleetwood
10
15
20
25
30
35
mpg