

Histogram(L4)

```
hist(x, col = "lightblue", ylim = c(a,b), xlim=c(a,b), xlab = "Lab for x axis", right = TRUE, main="Title for the histogram", breaks = seq(m,n,p))
```

x: the vector to visualize

col=: change the color of the histogram

xlim/ylim=: define the range of x/y axis

xlab/ylab=: rename the label for x/y axis

right=TRUE/FALSE: "TRUE" stands for the right-closed (left-opened) interval. "FALSE" stands for the right-opened (left-closed) interval

main=: name the title for the histogram

breaks=: set up the value of x axis

Single Boxplot(L5)

```
boxplot(x, horizontal=TRUE, log="x")
```

x: the vector to visualize

horizontal=TRUE/FALSE: make the boxplot horizontally or vertically

log=: if the x value is in the log scale

Multiple Boxplot(L5)

```
ggplot(dataset, aes(x= ,y=)) + geom_boxplot() + labs() + theme(legend.position = "bottom")
```

dataset: the dataset to visualize

aes(x=,y=): plot by x & y

labs(): label the element in the boxplot

theme(legend.position): assign the position of the legend

Violin Plot(L5)

```
ggplot(dataset, aes(x= ,y= )) + geom_violin() + coord_flip() + labs() + theme()
```

dataset: the dataset to visualize

aes(x=,y=): plot by x & y

geom_violin: get the violin plot

coord_flip: flip the x and y coordinate

theme: customize the non-data component

Ridgeline Plot(L5)

```
ggplot(dataset, aes(x= ,y= )) + geom_density_ridges(fill = "blue", alpha= ,scale= )
```

dataset: the dataset to visualize

aes(x=,y=): plot by x & y

geom_density_ridges: get the Ridgeline plot

fill= : fill the Ridgeline with specific color

alpha= : set the transparency of the area under the Ridgeline

Types of data(L8)

Numerical	Categorical data
------------------	-------------------------

data

~Discrete ~Nominal - no fixed category order

~Continuous ~Ordinal - fixed category order

Tidy Data(L10)

pivot_longer(data, cols = , names_to = , values_to =): move selected columns' name to "name" column, and move values to a single "value" column

pivot_wider(data, names_from = , values_from =): use the name from a column as the column name, and use the value from select column to be the value in the final Dataframe

row.names_to_column(): add the column name to the rowname in the Dataframe

Parallel Coordinates(L13)

```
ggparcoord(dataset, columns = , scale = , alphaLines = , splineFactor = , groupColumn = )
```

dataset: the dataset to visualize

columns=: select columns of data that will include in the plot

scale= : method to scale the data (default is "std")

alphaLines= : value of alpha scalar for the lines of the parcoord plot or a column name of the data

splineFactor= : logical or numeric operator indicating whether spline interpolation should be used

groupColumn = : a single variable to group (color) by



By Icey_cey
cheatography.com/icey-cey/

Not published yet.

Last updated 1st April, 2022.

Page 1 of 2.

Sponsored by [ApolloPad.com](#)

Everyone has a novel in them. Finish Yours!

<https://apollopad.com>

Biplot (L14)

```
pca<- prcomp(dataset)
biplot (pca)
draw_b ipl ot( dat asset)
```

prcomp(): perform a principal components analysis on the given data matrix
draw_biplot(): perform PCA on a data frame and draw a biplot

Cleveland dot plot(L15)

```
ggplot(dataset, aes(x = , y =
fct_reorder()))
+geom_point( color = )
+theme_linedraw()
```

fct_reorder(): reorder factor levels by sorting along the variables
geom_point(): create scatterplots
theme_linedraw(): add black lines of various widths on white backgrounds

Multivariate Data(L15)

Stacked bar chart	Grouped bar chart	Mosaic plot (two variables)
ggplot(data, aes(x= , fill =))+geom_m_bar() +scale_fill_manual()	ggplot(data, aes(x= , fill=))+geom_bar(position = "dodge") +scale_fill_manual()	mosaic(x~y, direction = "row") + geom_text(aes(label = paste(after_start(stratum), "\n", after_start(count)))) + scale_x_discrete(limits =)

Multivariate Data(L15) (cont)

~plot x with different fill in color	~bar plot grouped with x filling with color	~direction stands for the direction of different variables. highlighting_fill used for distinguish different group color
--------------------------------------	---	--

Alluvial diagram(L16)

```
ggplot(dataset, aes(axis1 = ,
axis2 = , y = )) +
  geom_alluvium(color = )
+
  geom_s_stratum() +
  geom_text(stat = "stratum", aes(label = paste(after_start(stratum), "\n", after_start(count)))) +
  scale_x_discrete(limits = )
```

geom_alluvium(): plot both the nodes themselves, using geom_node(), and the flows between them, using geom_flow()
geom_stratum(): plot rectangles for these strata of a provided width
geom_text(): add only text to the plot
scale_x_discrete(): set the values for discrete x scale aesthetics

Heatmap(L17)

```
ggplot(dataset, aes(x= , y= )) +
  geom_tile(aes(fill = ), color = ) +
  coord_fixed()
```

geom_rect(): use the locations of the four corners (xmin, xmax, ymin and ymax)
geom_tile(): use the center of the tile and its size (x, y, width, height)
geom_raster(): a high performance special case for when all the tiles are the same size
coord_fixed(): a fixed scale coordinate system forces a specified ratio between data units on the axes

Time series(L20)

```
ggplot(dataset, aes(x= , y= , color= )) +
  geom_line() +
  geom_smooth(method= , span= )
```

ggplot(dataset, aes(x= , y= , color=)): plot multiple time series by different colors
geom_smooth(): add a smooth line according to the data

method= : smoothing method (function) to use
span= : control the amount of smoothing for the default loess smoother

Factor in R

fct_recode(): change the name of the factor
fct_inorder(): display by each factor in the original order

fct_relevel(x, "G1", "G2", after = 3): move the factor "G1", "G2" after the third item in factor x

fct_reorder(color, count, .desc=TRUE): order by decreasing frequency count

fct_infreq(): display by number of observations with each level (default is decreasing order of frequency)

fct_rev(): reverse the order of factor levels

fct_explicit_na(): turn NAs into a real factor level