library(tidyverse) library(ggplot2)

install.packages("gridExtra")

library(gridExtra) library(ggplot2) library(dplyr) library(plotly) library(hrbrthemes)

data()

head(diamonds) str(diamonds)

qplot(carat, price, data = diamonds)

qplot(log(carat), log(price), data = diamonds)

qplot(carat, price, data = diamonds[1:50,], colour = color)

qplot(carat, price, data = diamonds[1:50,], shape = cut)

qplot(carat, price, data = diamonds[1:50,], size = price)

library(scales) qplot(carat, price, data = diamonds, colour = I(alpha("black", 1/200)))

qplot(carat, price, data = diamonds, geom = c("point", "smooth"))

qplot(carat, data = diamonds, geom = "histogram") qplot(carat, data = diamonds, geom = "density") qplot(carat, data = diamonds, geom = "histogram", fill = color) qplot(carat, data = diamonds, geom = "density", colour = color)

str(diamonds)

points-scatterplot

p<-ggplot(diamonds, aes(x=carat, y=price, color=cut))+ geom_point() p ggsave("myggplot.png", plot=p, width = 10,height=20, dpi = 300) # save a stored ggplot

bar

p<- ggplot(diamonds, aes(cut)) + geom_bar() p

p<-ggplot(diamonds, aes(cut)) + geom_bar(aes(fill = clarity),position = "stack") p

ggplot(diamonds, aes(cut)) + geom_bar(aes(fill = clarity), position = "fill")

ggplot(diamonds, aes(cut)) + geom_bar(aes(fill = clarity), position = "dodge")

"

marker

p<-ggplot(diamonds, aes(x=carat, y=price)) + geom_point(aes(size=carat, shape=clarity, alpha=price)) p

"

Layers

Overlay a smoothing line on top of the scatter plot using geom_smooth

Adding scatterplot geom (layer1) and smoothing geom (layer2).

p<-ggplot(diamonds) + geom_point(aes(x=carat, y=price, color=cut)) + geom_smooth(aes(x=carat, y=price, color=cut)) p

facet-faceting that allows the user to split one plot into

multiple plots based on a factor included in the dataset.

columns defined by 'cut'

facet_wrap(~ factor1 + factor2 + ... + factorn)

p<- p+facet_wrap(~ cut, nrow=3,ncol=2)
p

facet_grid(row_variable ~ column_variable)

p<-p + facet_grid(color ~ cut) p

p <- ggplotly(p) p

ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point() + facet_wrap(~ clarity)

plot <- ggplot(diamonds, aes(x=carat,y=price))+ geom_density(aes(fill=cut),alpha=0.5)

plot<-plot+facet_wrap(~cut) plot

the main title, x and y axis labels

ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point() + geom_smooth() + ggtitle("Scatter") + xlab("carat") + ylab("price")+ coord_cartesian(ylim=c(0, 10000))

Change the appearance of the main title &axis labels -- theme() & element_text()

main title -p + theme(plot.title = element_text(family, face, colour, size))

x/y axis title -p + theme(axis.title.x/y = element_text(family, face, colour, size))

"

family : font family face : font face. Possible values are plain, italic, bold and bold.italic colour : text color size : text size in pts hjust : horizontal justification (in [0, 1]) vjust : vertical justification (in [0, 1]) lineheight : line height. In multi-line text, the lineheight argument is used to change the spacing between lines. color : an alias for colour "

Legend - Deleting and Changing Position

remove legend

p1 <- ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point() + geom_smooth() +
theme(legend.position="none", axis.title.x = element_text(color="blue", size=14, face="bold"), axis.title.y = element_text(color="#993333", size=14, face="bold")) +
labs(title="legend.position='none'")

legend at top

p2 <- ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point() + geom_smooth() + theme(legend.position="top") + labs(title="legend.position='top'")

legend inside the plot.

p3 <- ggplot(diamonds, aes(x=carat, y=price, color=cut)) + geom_point() + geom_smooth() + labs(title="legend.position='coords inside plot'") +
theme(legend.justification=c(1,0), legend.position=c(1,0))

arrange

grid.arrange(p1, p2, p3,nrow=1,ncol=3)