```
library(ggplot2)

# 1. Plotting layers ----
## 1.1 Base layer ----
ggplot(mtcars)


## 1.2 Adding geom and scale layers ----
ggplot(mtcars) +
  geom_point(aes(x=wt, y=mpg))


## 1.3 Adding title annotations ----
ggplot(mtcars) +
  geom_point(aes(x=wt, y=mpg)) +
  labs(x='Car weight (1000 lbs)',
       y='Miles per (US) gallon',
       title = 'Is car weight related to fuel efficiency?',
       subtitle = 'Car weight is measured in 1000 lbs. Fuel efficiency is measured in
miles per (US) gallon.',
       caption = 'Data come from the mtcars dataset.')


## 1.4 Adjusting the theme layer ----
ggplot(mtcars) +
  geom_point(aes(x=wt, y=mpg)) +
  labs(x='Car weight (1000 lbs)',
       y='Miles per (US) gallon',
       title = 'Is car weight related to fuel efficiency?',
       subtitle = 'Car weight is measured in 1000 lbs. Fuel efficiency is measured in
miles per (US) gallon.',
       caption = 'Data come from the mtcars dataset.') +
  theme(axis.title = element_text(color = 'grey50'),
        plot.title = element_text(face = 'bold'),
        plot.subtitle = element_text(color = 'grey50'),
        plot.caption = element_text(face = 'italic'))


## 1.5 Adding another geom layer (line of best fit) ----
ggplot(mtcars) +
  geom_point(aes(x=wt, y=mpg)) +
  labs(x='Car weight (1000 lbs)',
       y='Miles per (US) gallon',
       title = 'Is car weight related to fuel efficiency?',
       subtitle = 'Car weight is measured in 1000 lbs. Fuel efficiency is measured in
miles per (US) gallon.',
       caption = 'Data come from the mtcars dataset.') +
  theme(axis.title = element_text(color = 'grey50'),
        plot.title = element_text(face = 'bold'),
        plot.subtitle = element_text(color = 'grey50'),
        plot.caption = element_text(face = 'italic')) +
  geom_smooth(aes(x=wt, y=mpg),
              method = 'lm')


## 1.6 Adding another geom layer (text annotation) ----
ggplot(mtcars) +
  geom_point(aes(x=wt, y=mpg)) +
  labs(x='Car weight (1000 lbs)',
       y='Miles per (US) gallon',
       title = 'Is car weight related to fuel efficiency?',
       subtitle = 'Car weight is measured in 1000 lbs. Fuel efficiency is measured in
miles per (US) gallon.',
       caption = 'Data come from the mtcars dataset.') +
```

```
  theme(axis.title = element_text(color = 'grey50'),
        plot.title = element_text(face = 'bold'),
        plot.subtitle = element_text(color = 'grey50'),
        plot.caption = element_text(face = 'italic')) +
  geom_smooth(aes(x=wt, y=mpg),
              method = 'lm') +
  annotate("text", x=4.5, y=25, label = 'NOTE')


# 2. Aesthetics ----
## 2.1 Example of Global Aesthetics ----
ggplot(mpg) +
  geom_point(aes(displ, hwy),
             color = 'red')


## 2.2 Example of Mapped Aesthetics ----
ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class))


# 3. Linetypes and shapes ----
## 3.1 Linetype as global aesthetic ----
ggplot(mtcars) +geom_smooth(aes(x=wt, y=mpg), method = 'lm', linetype = 2)


## 3.2 Linetype as mapped aesthetic ----
ggplot(mtcars) +geom_smooth(aes(x=wt, y=mpg, linetype=as.factor(cyl)), method = 'lm')


## 3.3 Shape as global aesthetic ----
ggplot(mtcars) +geom_point(aes(x=wt, y=mpg), shape = 2)


## 3.4 Shape as mapped aesthetic ----
ggplot(mtcars) +geom_point(aes(x=wt, y=mpg, shape = as.factor(cyl)))


# 4. Scales ----
# 4.1 Adjusting axis scales (continuous) ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Species)) +
  scale_x_continuous(limits=c(0,7), breaks = seq(1,7,1)) +
  scale_y_continuous(limits=c(0,7), breaks = seq(0,7,1))


# 4.2 Adjusting axis scales (discrete) ----
ggplot(iris) +
  geom_boxplot(aes(Species,Petal.Length)) +
  scale_x_discrete(limits = c('virginica','versicolor','setosa'),
                   labels = c('VIRGINICA','VERSICOLOR','SETOSA'))

# Caution: In many cases, the ordering of discrete values on the X axis
#          is better manipulated by factoring the variable while cleaning
#          prior to plotting.


# 4.3 Adjusting color scales (continuous) ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Species))


### 4.3a Low/High Gradient ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Petal.Width)) +
```

```
  scale_color_gradient(low = 'grey40', high = 'orange')


### 4.3b Low/High Steps ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Petal.Width)) +
  scale_color_steps(low = 'grey40', high = 'orange', n.breaks = 3)


### 4.3c Color Brewer Palettes ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Petal.Width)) +
  scale_color_fermenter(palette = 'Set2')

# Use scale_color_brewer() for gradient.
# Use scale_color_fermenter() for steps.
# See more palettes: https://r-graph-gallery.com/38-rcolorbrewers-palettes.html


### 4.3d Viridis Palettes ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Petal.Width)) +
  scale_color_viridis_c()

# Use scale_color_viridis_c() for gradient.
# Use scale_color_viridis_b() for steps.
# See more palettes: https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-
viridis.html


# 4.4 Adjusting color scales (discrete) ----
### 4.4a Manually ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Species)) +
  scale_color_manual(values = c('grey40','firebrick','orange'))

### 4.4b Greyscale ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Species)) +
  scale_color_grey()


### 4.4c Color Brewer Palettes ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Species)) +
  scale_color_brewer(palette = 'Set2')

# See more palettes: https://r-graph-gallery.com/38-rcolorbrewers-palettes.html


### 4.4d Viridis Palettes ----
ggplot(iris) +
  geom_point(aes(Petal.Length,Petal.Width, color=Species)) +
  scale_color_viridis_d(option = 'turbo')

# See more palettes: https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-
viridis.html


# 5. Faceting ----
## 5.1 Example of facet_wrap ----
ggplot(mtcars) +
  geom_point(aes(hp, mpg)) +
  facet_wrap(~as.factor(cyl))
```

```
## 5.2 Example of facet_grid ----
ggplot(mtcars) +
  geom_point(aes(hp, mpg)) +
  facet_grid(as.factor(cyl) ~ as.factor(gear))
```