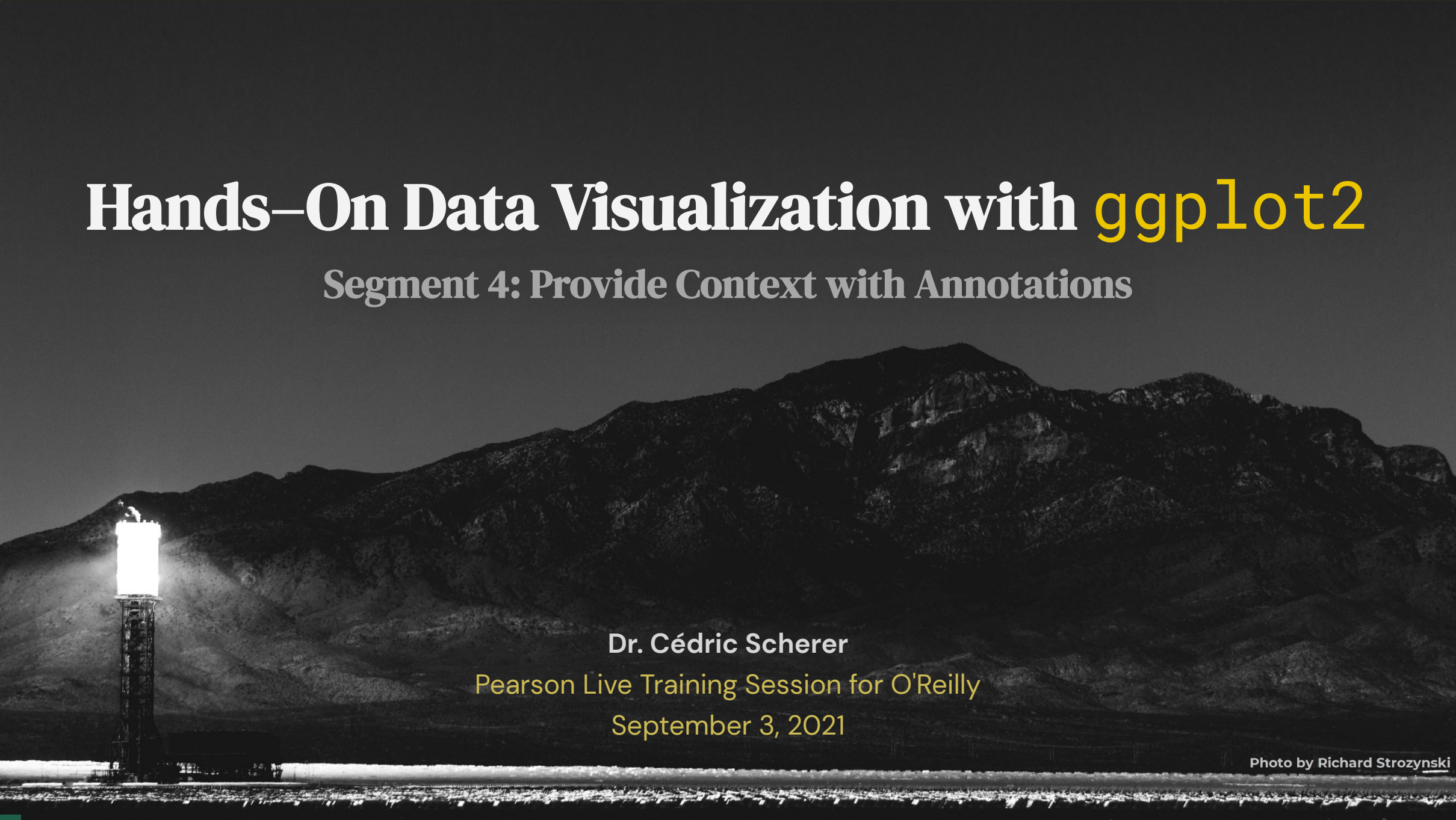
Hands–On Data Visualization with ggplot2

Segment 4: Provide Context with Annotations



Dr. Cédric Scherer

Pearson Live Training Session for O'Reilly

September 3, 2021

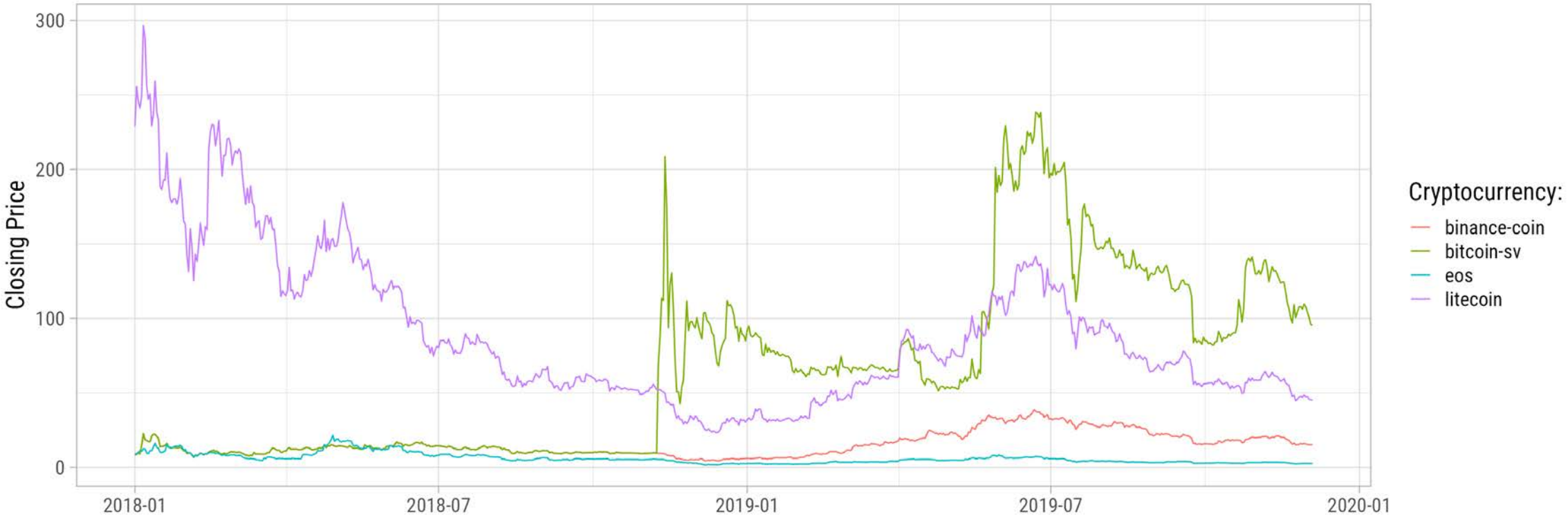
Photo by Richard Strozynski

Titles, Labels & Co

Labels: `labs()`

To change the labels and add a title, a subtitle, a caption and/or a tag, use `labs()`:

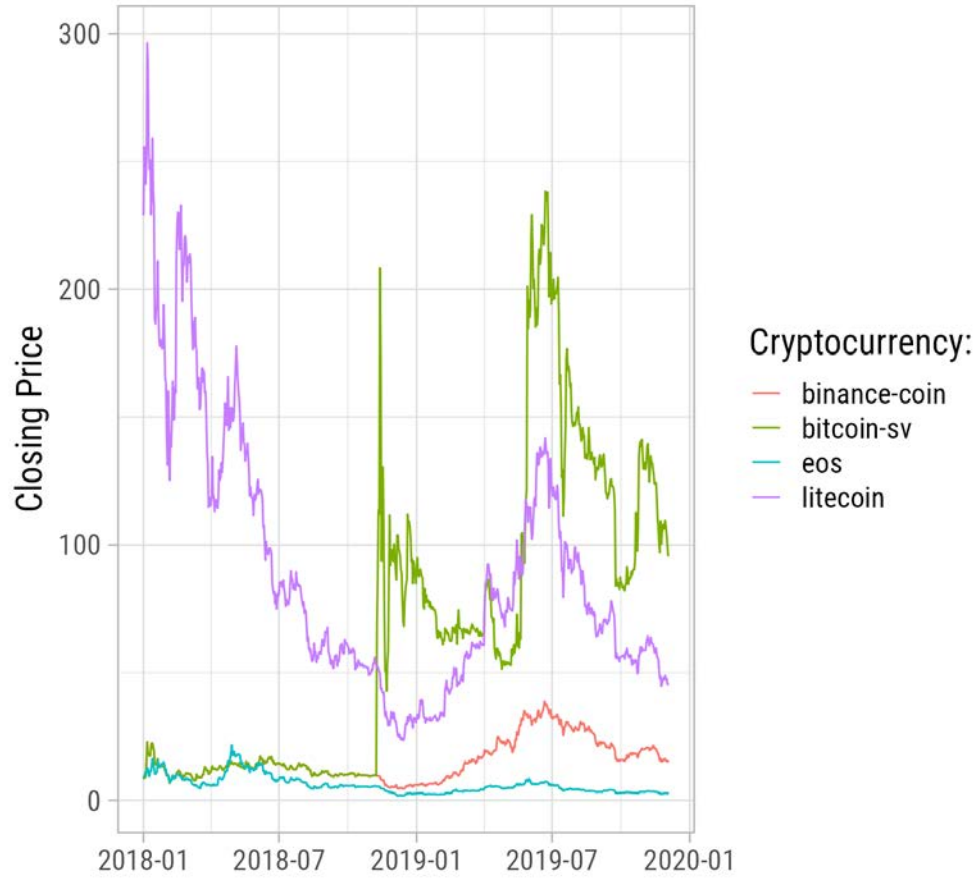
```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(x = NULL, y = "Closing Price", color = "Cryptocurrency:")
```



Labels: `labs()`

There are multiple ways to add labels:

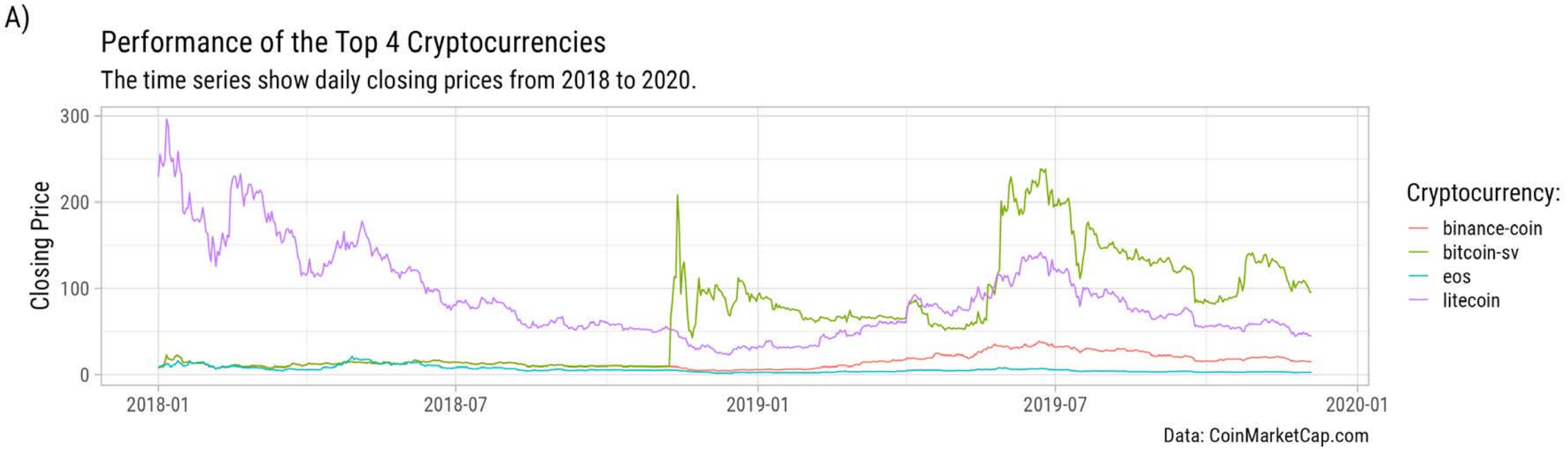
```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  xlab(NULL) +
  # scale_x_date(name = NULL) +
  ylab("Closing Price") +
  # scale_y_continuous(name = "Closing Price") +
  scale_color_discrete(name = "Cryptocurrency:")
```



Labels: `labs()`

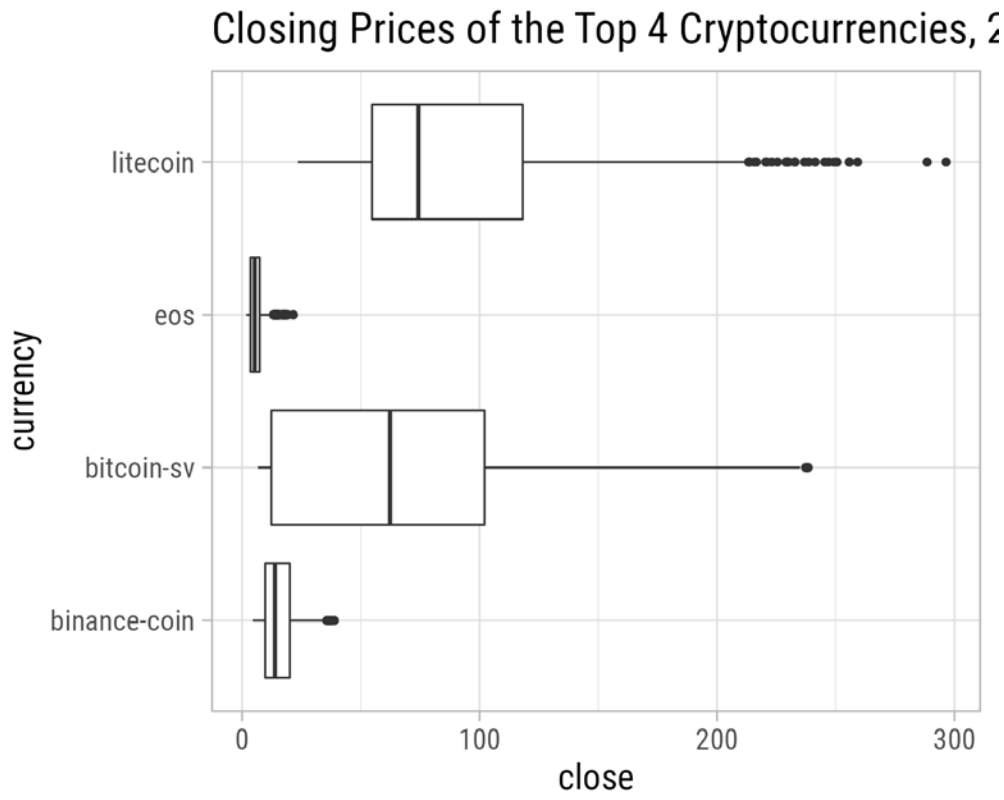
To change the labels and add a title, a subtitle, a caption and/or a tag, use `labs()`:

```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(x = NULL, y = "Closing Price", color = "Cryptocurrency:",
       title = "Performance of the Top 4 Cryptocurrencies", caption = "Data: CoinMarketCap.com",
       subtitle = "The time series show daily closing prices from 2018 to 2020.", tag = "A)")
```

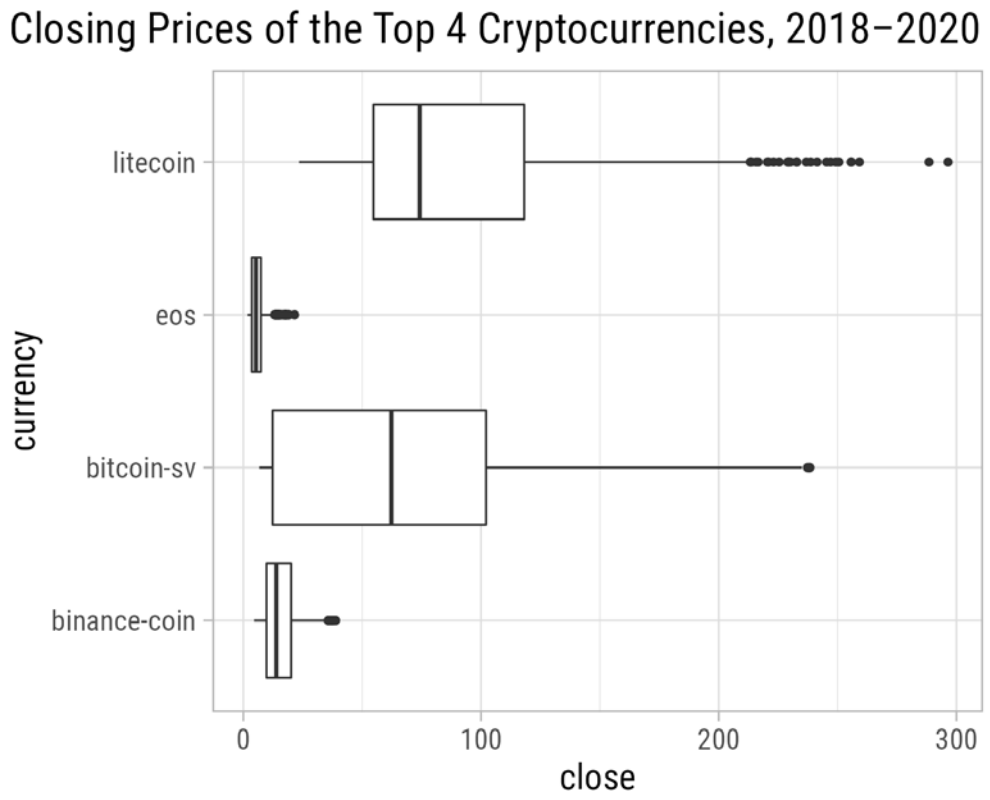


Plot Position

```
ggplot(data, aes(close, currency)) +
  geom_boxplot() +
  ggtitle("Closing Prices of the Top 4 Cryptocurrenc
  theme(plot.title.position = "panel")
```




```
ggplot(data, aes(close, currency)) +
  geom_boxplot() +
  ggtitle("Closing Prices of the Top 4 Cryptocurrenc
  theme(plot.title.position = "plot")
```



Text Rendering with {ggtext}

The {ggtext} package provides simple Markdown and HTML rendering for {ggplot2}.

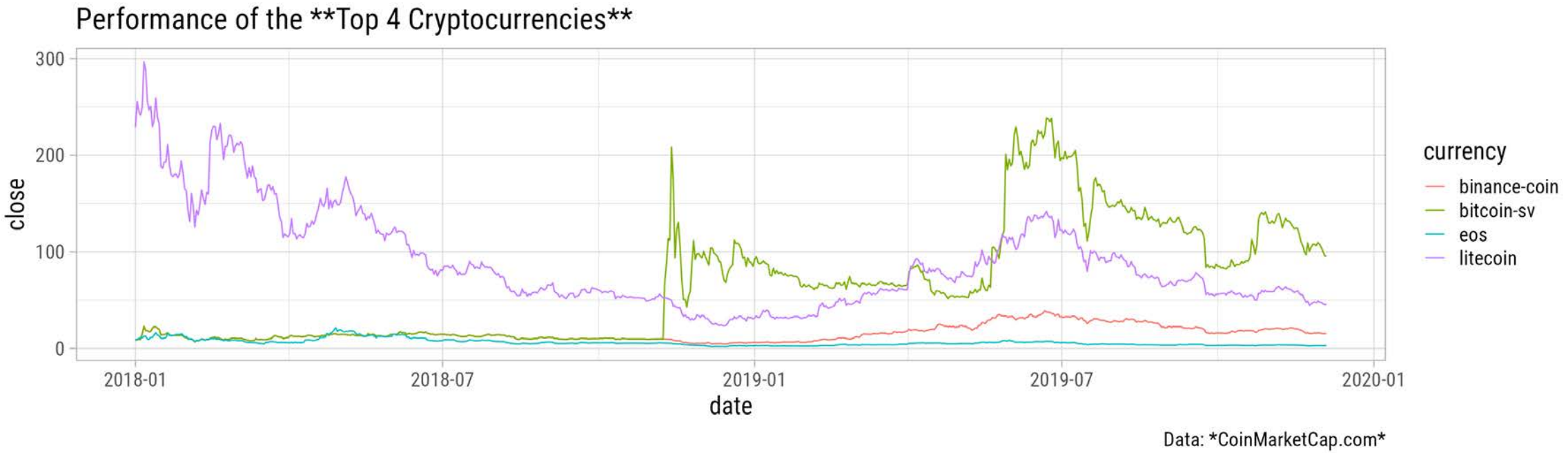
POLL: What is your level of expertise?

- I know neither Markdown nor HTML.
- I use Markdown but not HTML.
- I use HTML (basic) but not Markdown.
- I use HTML (extensively) but not Markdown.
- I use HTML (basic) and Markdown.
- I use HTML (extensively) and Markdown.

Text Rendering with {ggtext}

The {ggtext} package provides simple Markdown and HTML rendering for {ggplot2}.

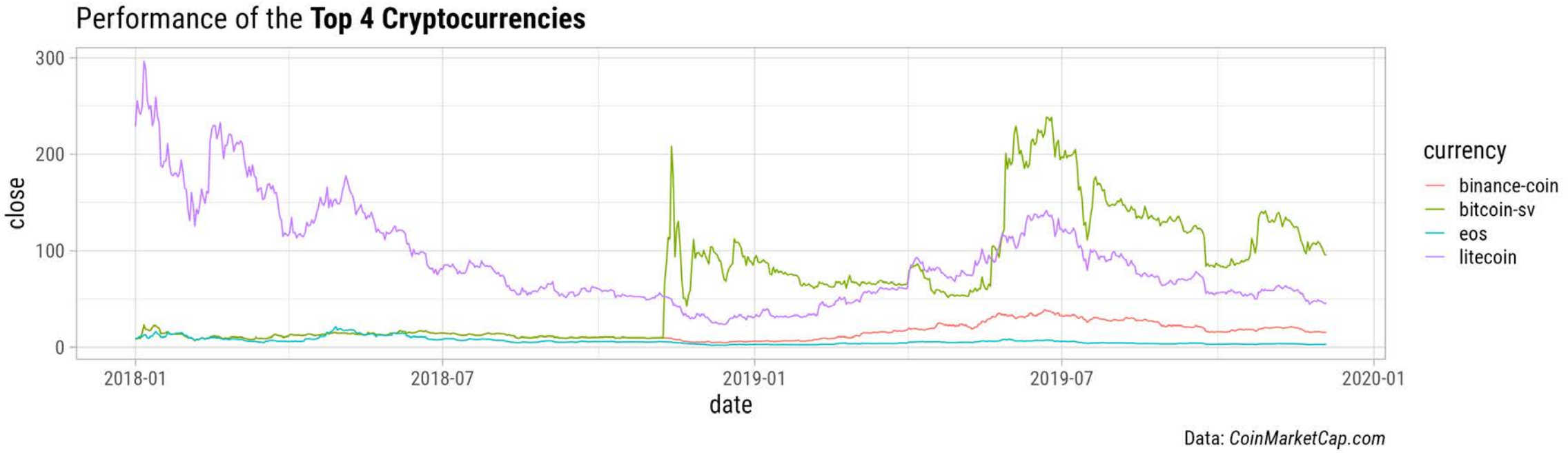
```
#install.packages("ggtext")
library(ggtext)
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(title = "Performance of the **Top 4 Cryptocurrencies**", caption = "Data: *CoinMarketCap.com*")
```



Text Rendering with {ggtext}

The {ggtext} package provides simple Markdown and HTML rendering for {ggplot2}.

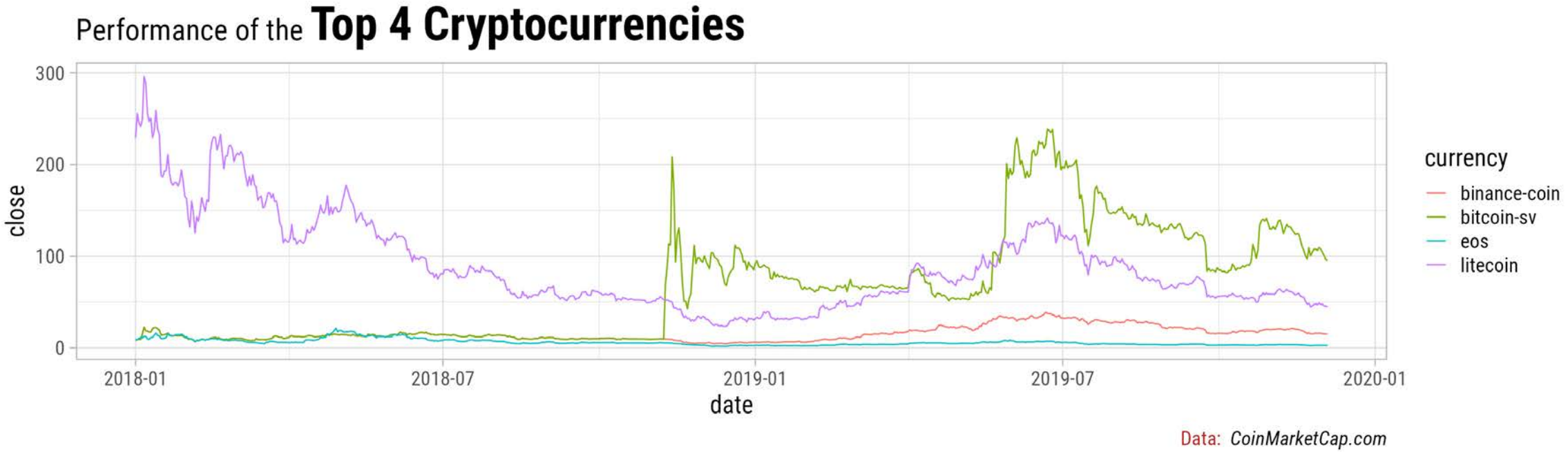
```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(title = "Performance of the **Top 4 Cryptocurrencies**", caption = "Data: *CoinMarketCap.com*") +
  theme(plot.title = element_markdown(), plot.caption = element_markdown())
```



Text Rendering with {ggtext}

The {ggtext} package provides simple Markdown and HTML rendering for {ggplot2}.

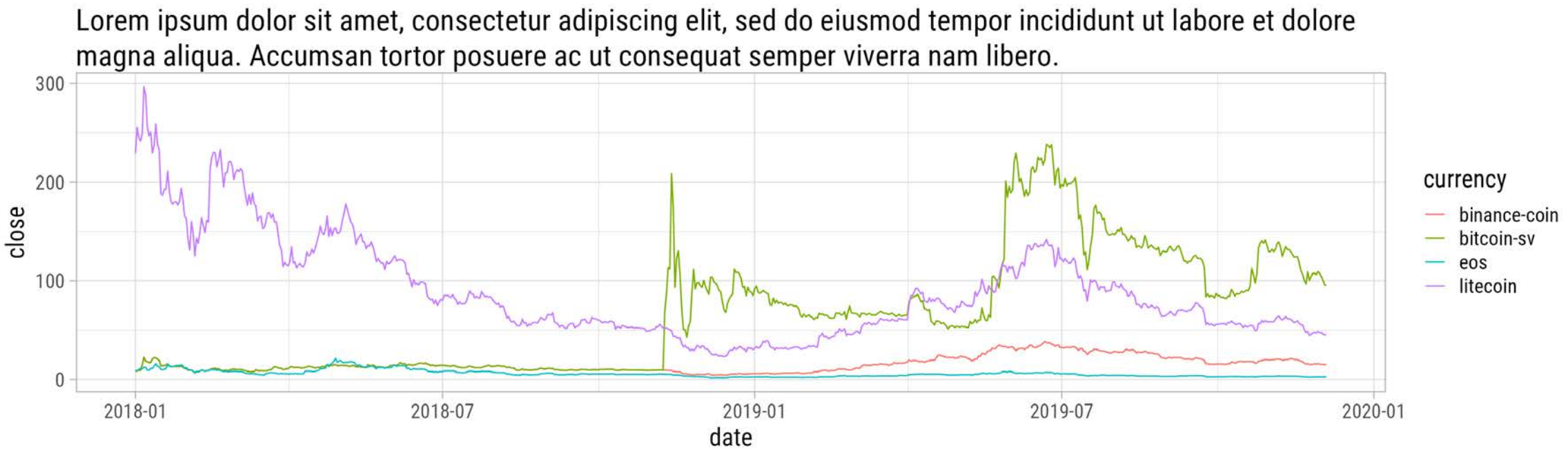
```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(title = "Performance of the <b style='font-size:34pt;'>Top 4 Cryptocurrencies</b>",
       caption = "<span style='color:firebrick;'>Data:</span> <i>CoinMarketCap.com</i>") +
  theme(plot.title = element_markdown(), plot.caption = element_markdown())
```



Text Rendering with {ggtext}

`element_textbox` and `element_textbox_simple` automatically wrap long text:

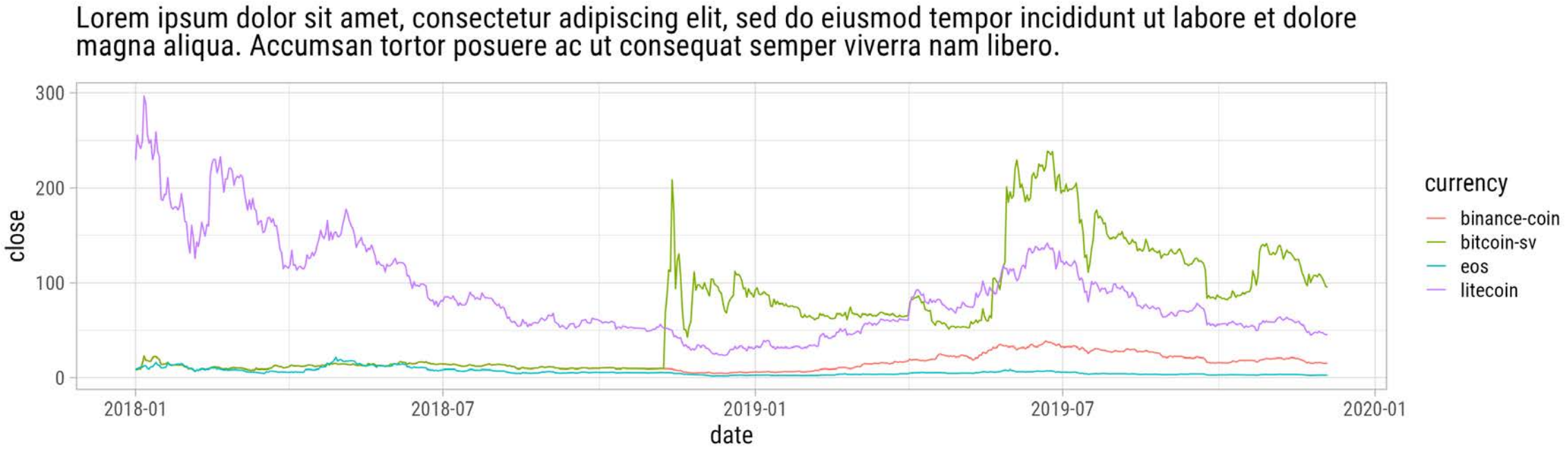
```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(title = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
  theme(plot.title = element_textbox_simple())
```



Text Rendering with {ggtext}

`element_textbox` and `element_textbox_simple` automatically wrap long text:

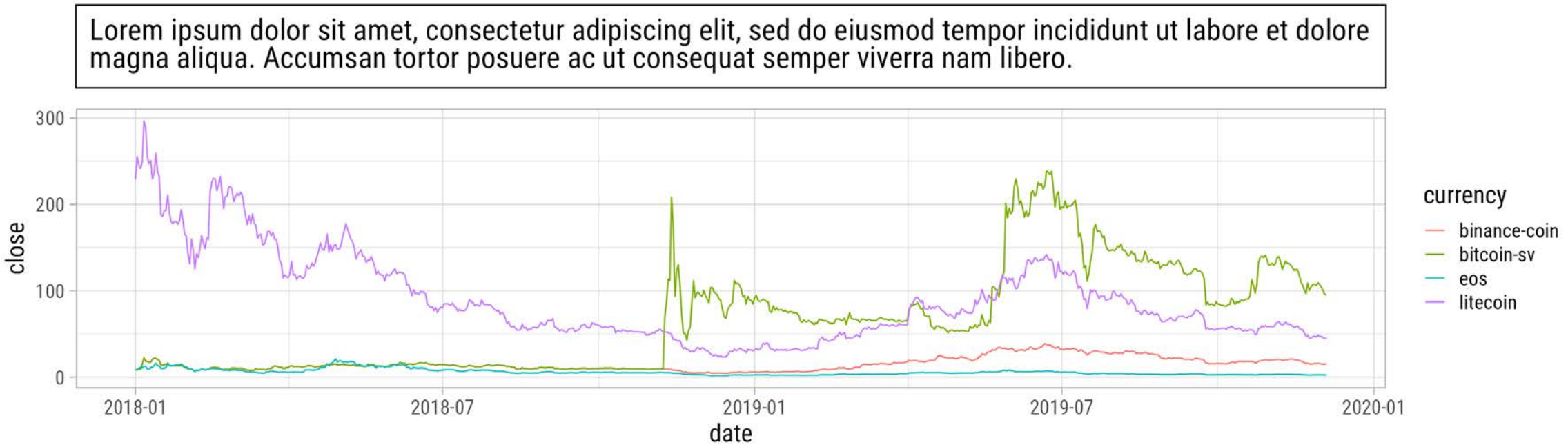
```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(title = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
  theme(plot.title = element_textbox_simple(margin = margin(b = 15), lineheight = .9))
```



Text Rendering with {ggtext}

`element_textbox` and `element_textbox_simple` automatically wrap long text:

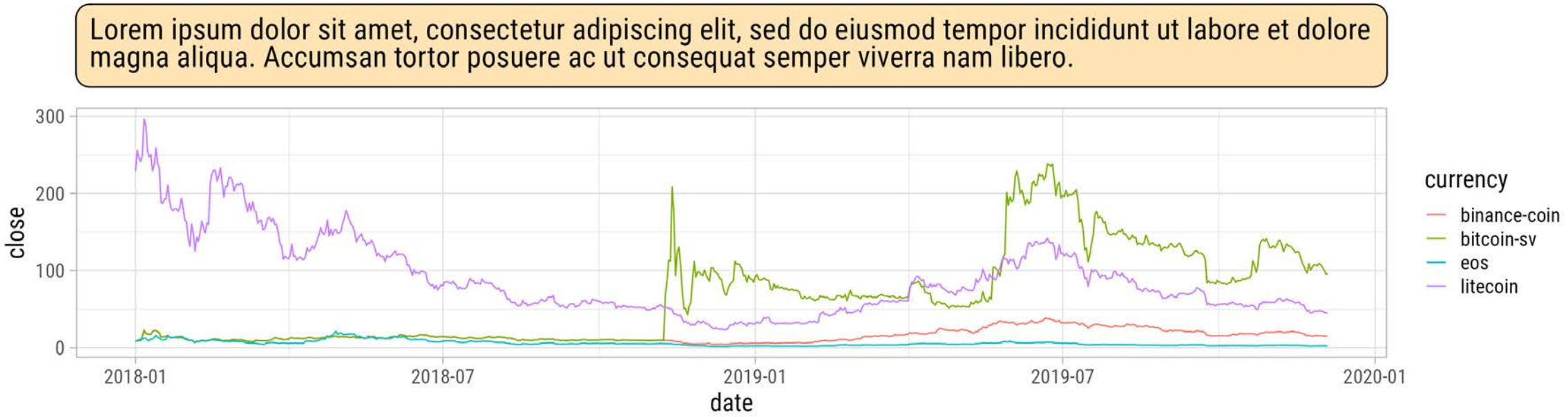
```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(title = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
  theme(plot.title = element_textbox_simple(margin = margin(b = 15), lineheight = .9,
                                            linetype = 1, padding = margin(rep(10, 4))))
```



Text Rendering with {ggtext}

`element_textbox` and `element_textbox_simple` automatically wrap long text:

```
ggplot(data, aes(date, close, color = currency)) +
  geom_line() +
  labs(title = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
  theme(plot.title = element_textbox_simple(margin = margin(b = 15), lineheight = .9,
                                            linetype = 1, padding = margin(rep(10, 4)),
                                            r = unit(10, "pt"), fill = "moccasin"))
```

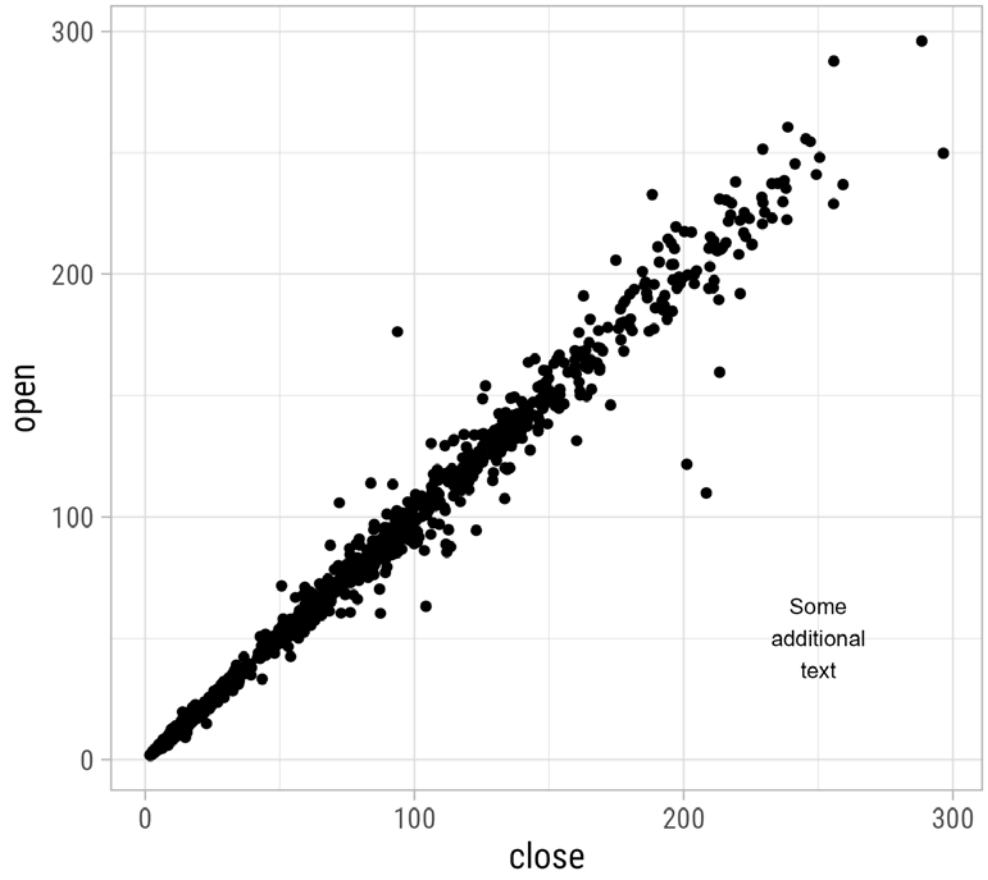


Basic Text Labelling

Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

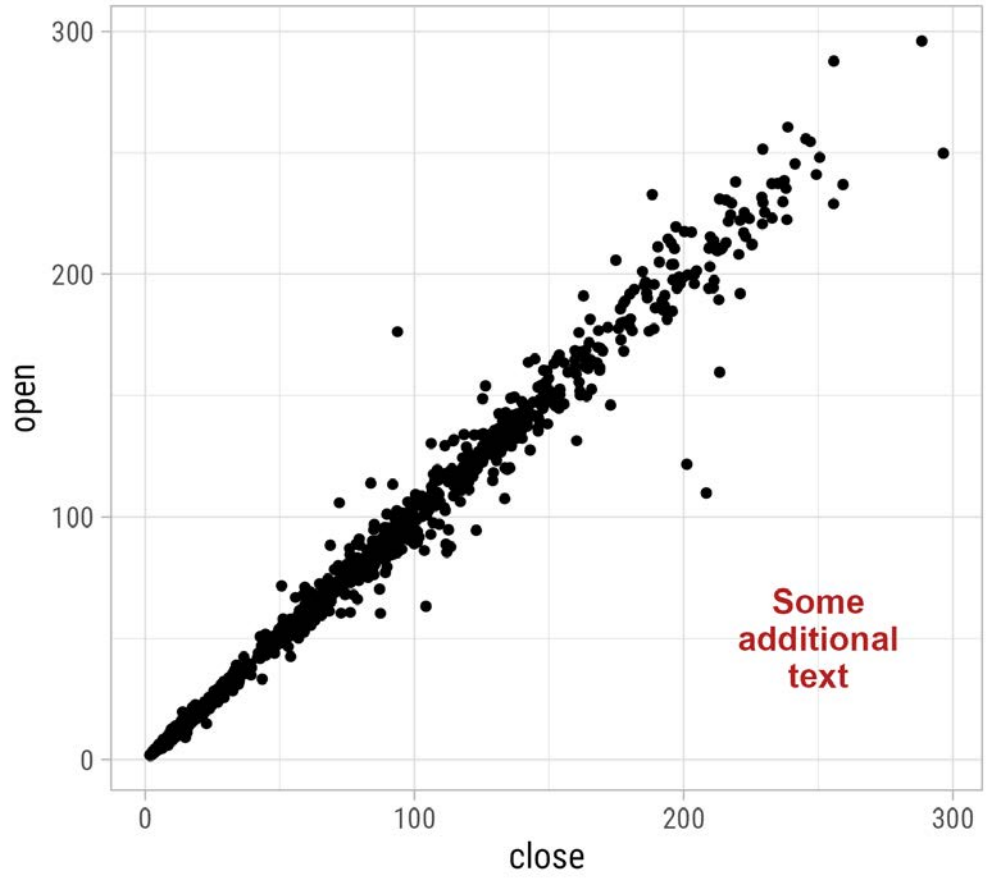
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "text",
    x = 250,
    y = 50,
    label = "Some\nadditional\ntext"
  )
```



Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

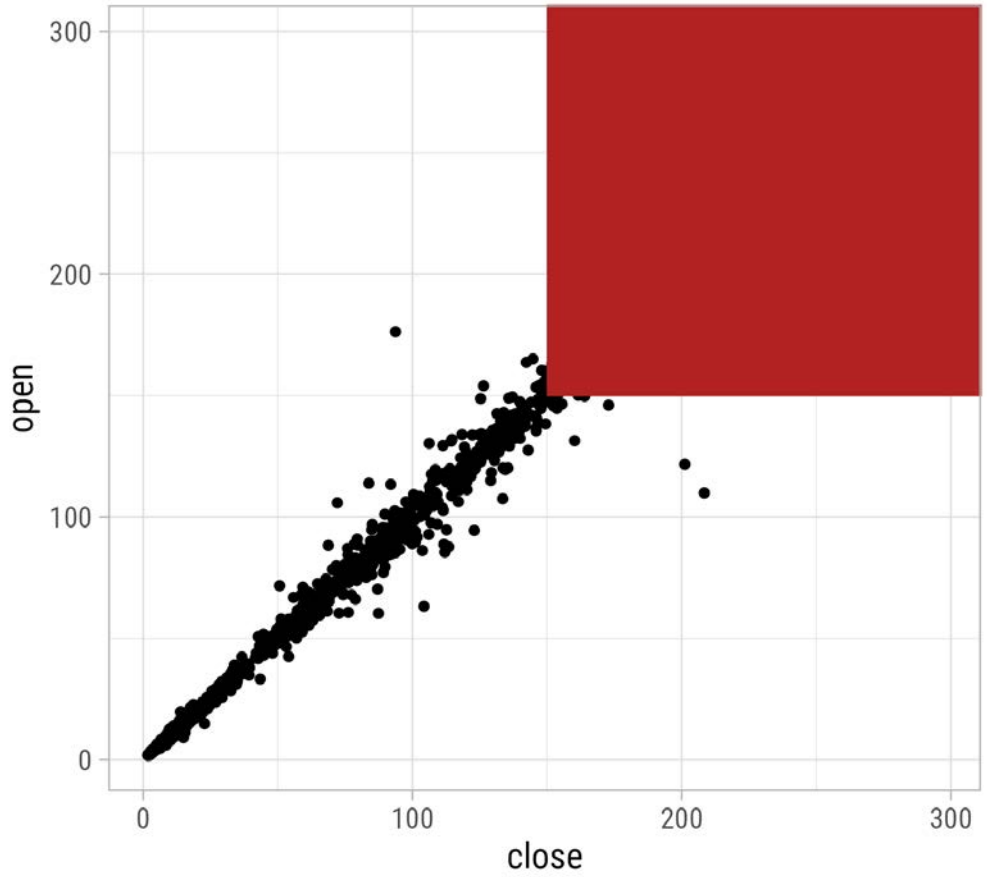
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "text",
    x = 250,
    y = 50,
    label = "Some\nadditional\ntext",
    size = 6,
    color = "firebrick",
    fontface = "bold",
    lineheight = .9
  )
```



Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

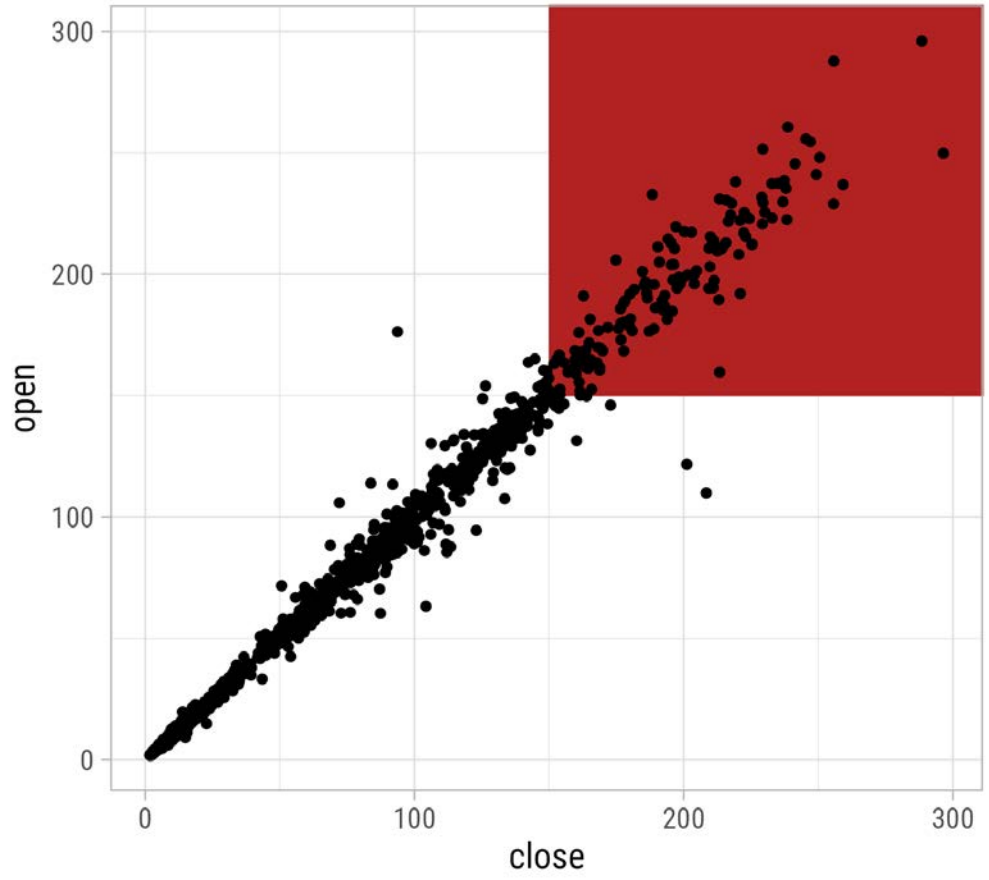
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "rect",
    xmin = 150,
    xmax = Inf,
    ymin = 150,
    ymax = Inf,
    fill = "firebrick"
  )
```



Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

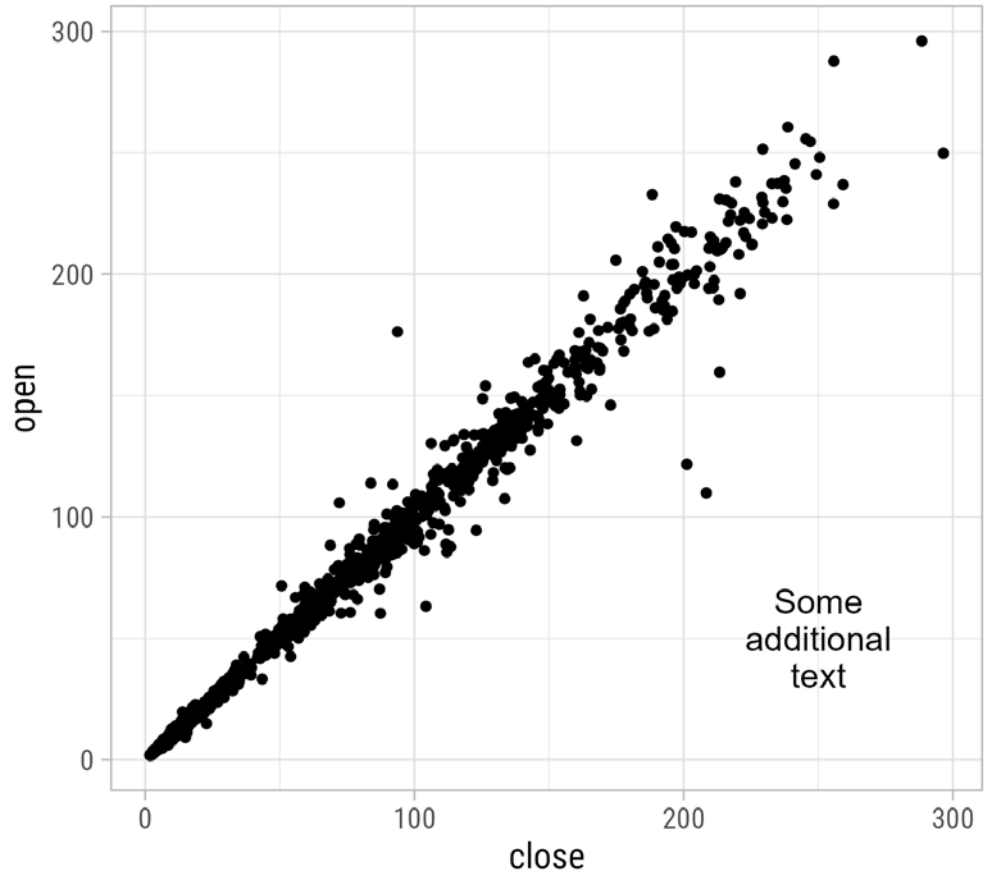
```
ggplot(data, aes(close, open)) +
  annotate(
    geom = "rect",
    xmin = 150,
    xmax = Inf,
    ymin = 150,
    ymax = Inf,
    fill = "firebrick"
  ) +
  geom_point(size = 2)
```



Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

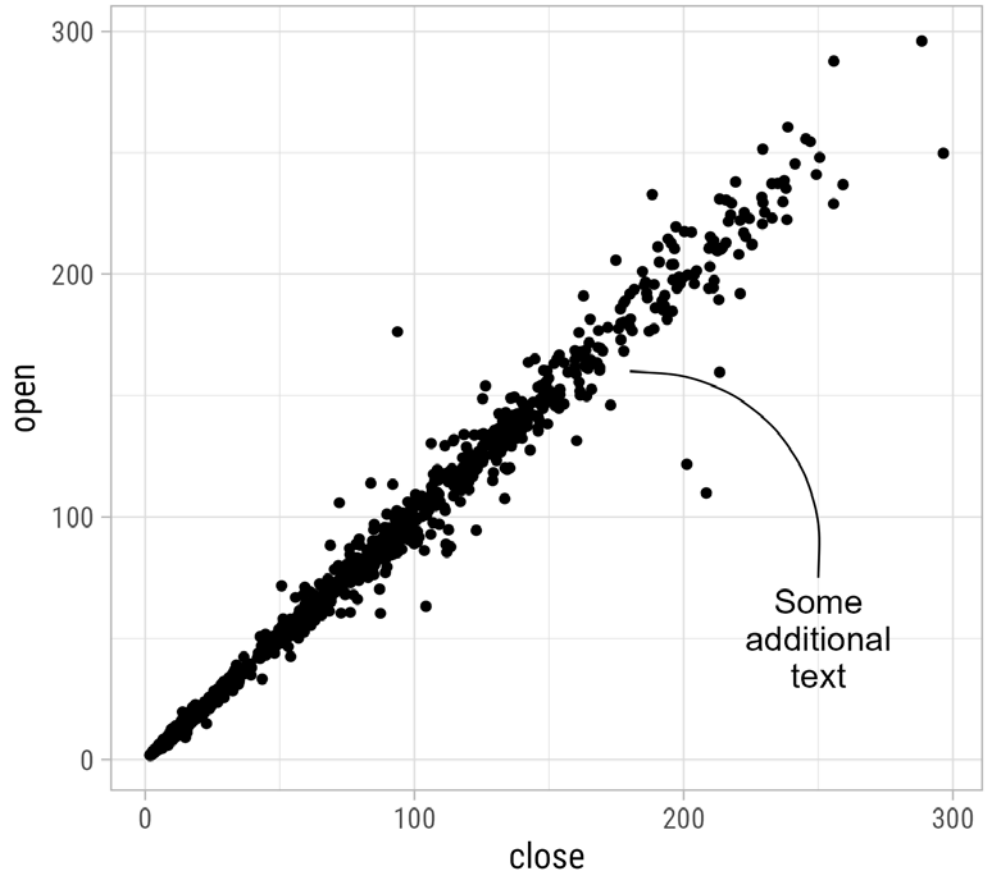
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "text",
    x = 250, y = 50,
    label = "Some\nadditional\ntext",
    size = 6,
    lineheight = .9
  ) +
  annotate(
    geom = "line",
    x = 250, y = 75,
    xend = 180, yend = 160
  )
```



Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

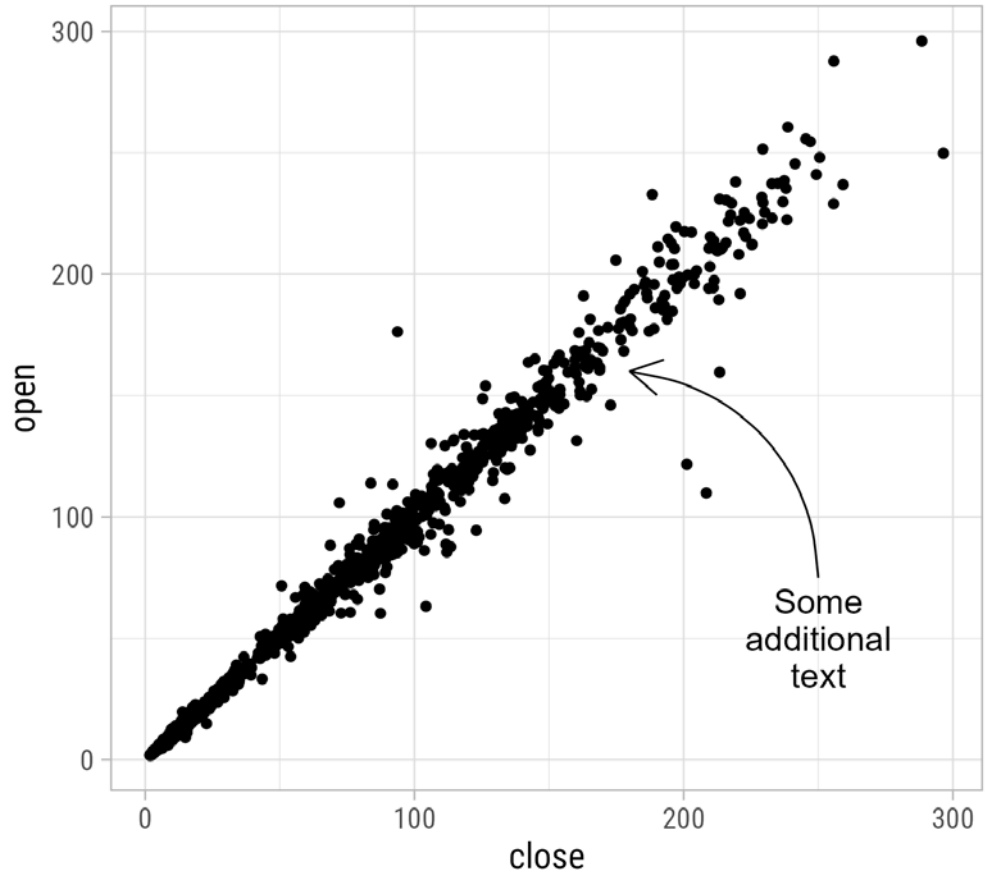
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "text",
    x = 250, y = 50,
    label = "Some\nadditional\ntext",
    size = 6,
    lineheight = .9
  ) +
  annotate(
    geom = "curve",
    x = 250, y = 75,
    xend = 180, yend = 160
  )
```



Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

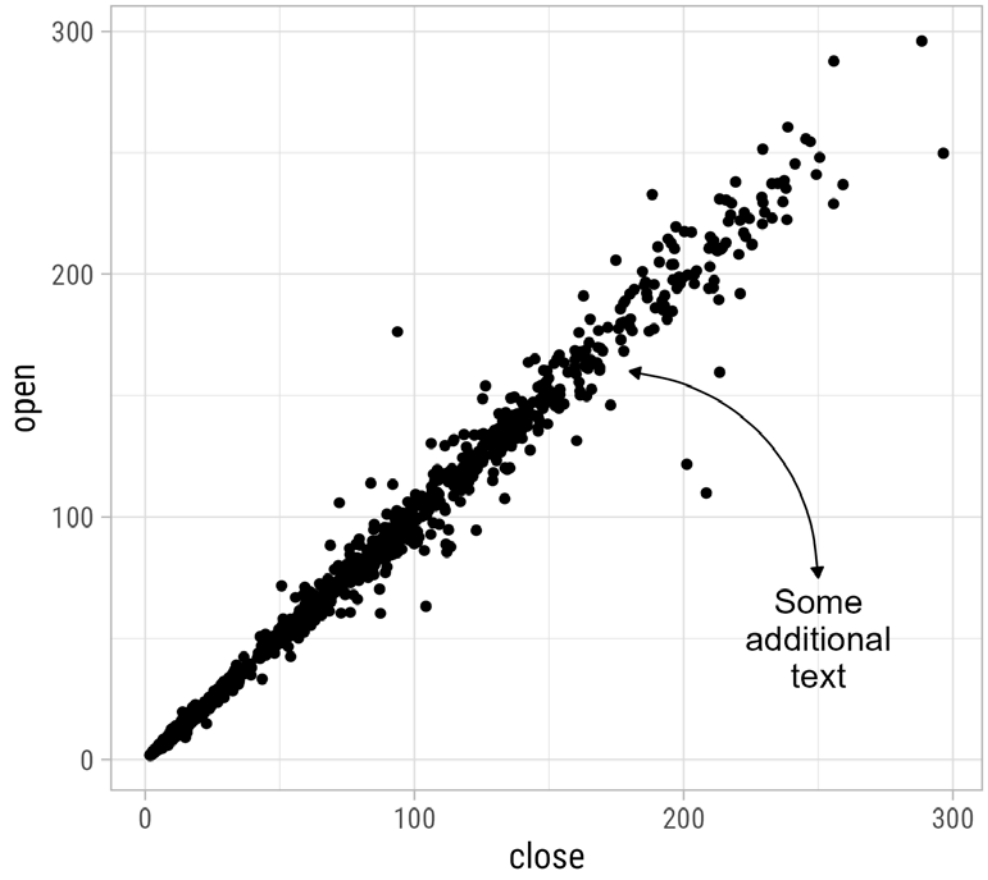
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "text",
    x = 250, y = 50,
    label = "Some\nadditional\ntext",
    size = 6,
    lineheight = .9
  ) +
  annotate(
    geom = "curve",
    x = 250, y = 75,
    xend = 180, yend = 160,
    curvature = .4,
    arrow = arrow()
  )
```



Annotations via `annotate()`

The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

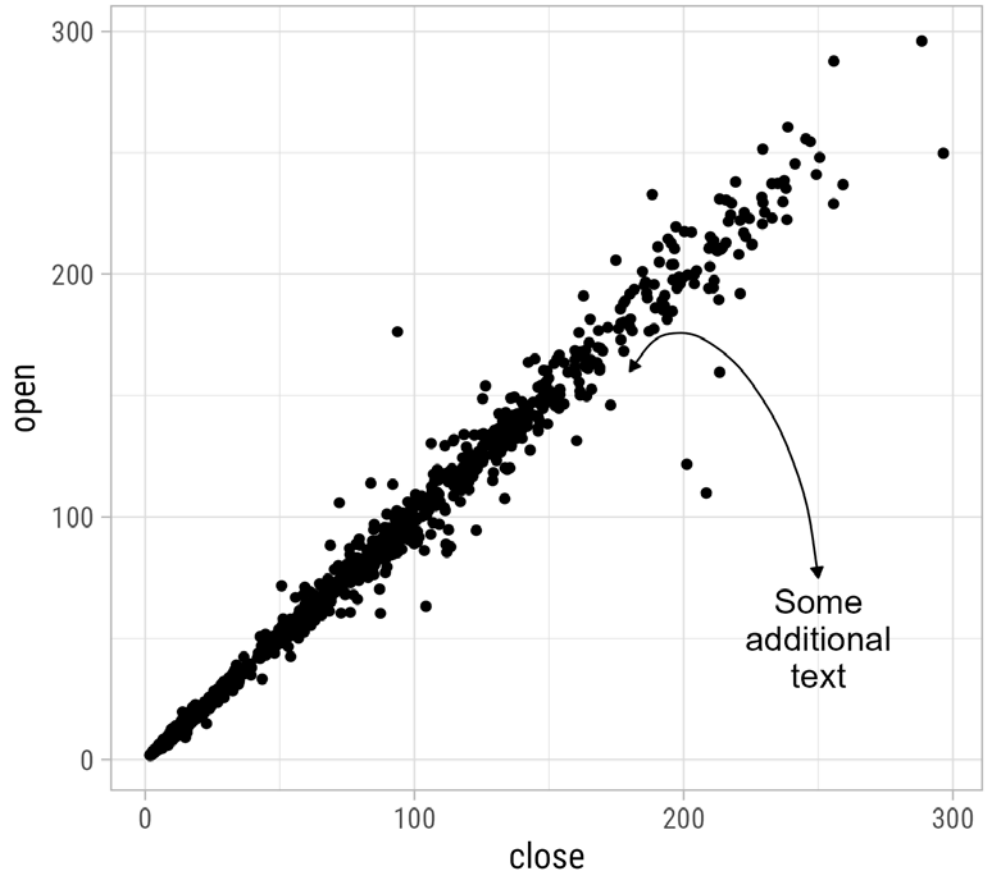
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "text",
    x = 250, y = 50,
    label = "Some\nadditional\ntext",
    size = 6,
    lineheight = .9
  ) +
  annotate(
    geom = "curve",
    x = 250, y = 75,
    xend = 180, yend = 160,
    curvature = .4,
    arrow = arrow(length = unit(0.4, "lines"),
                  type = "closed",
                  ends = "both")
  )
```



Annotations via `annotate()`

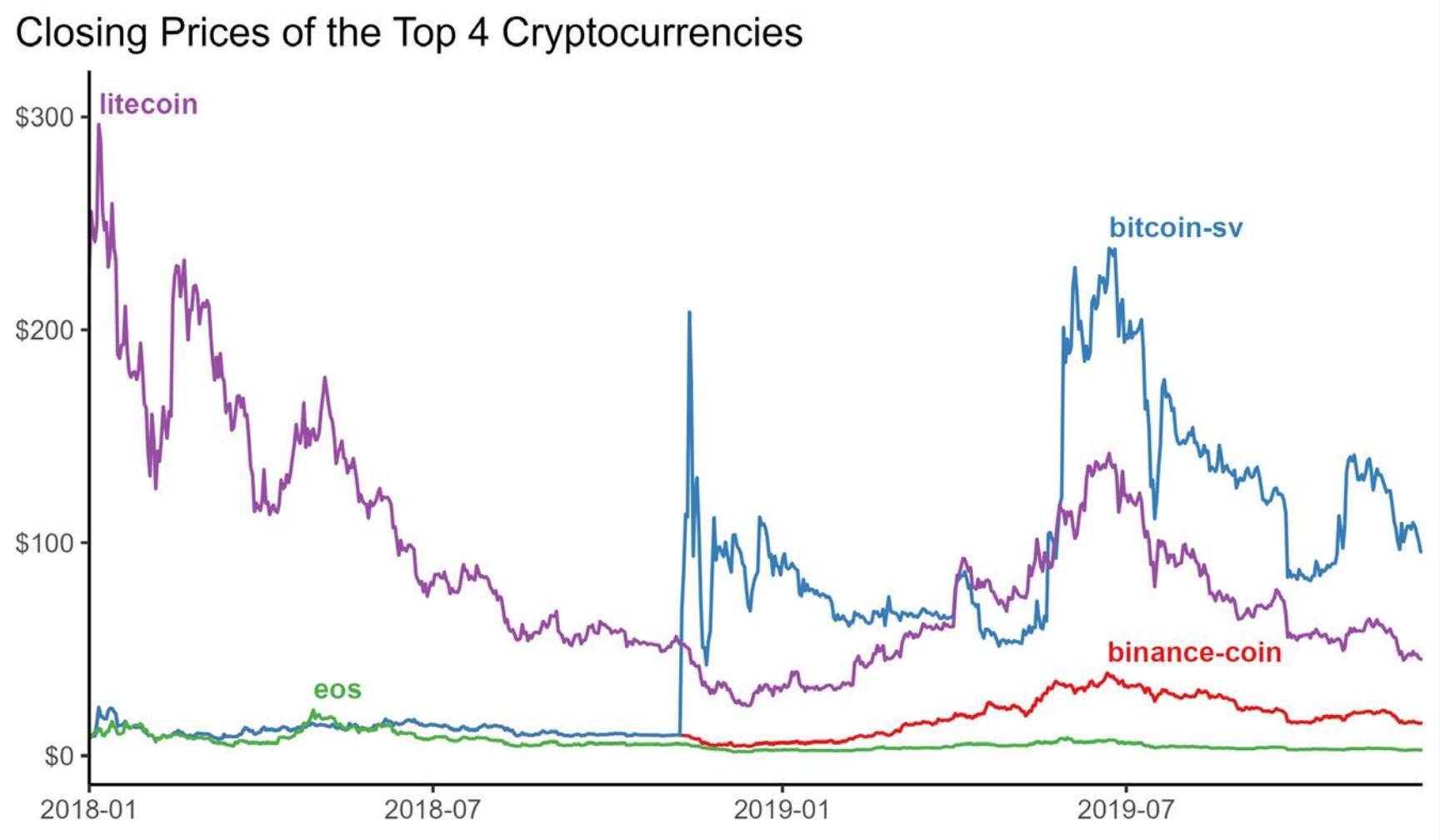
The `annotate()` function allows to add geom's to a plot without mapping to variables to aesthetics:

```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  annotate(
    geom = "text",
    x = 250, y = 50,
    label = "Some\nadditional\ntext",
    size = 6,
    lineheight = .9
  ) +
  annotate(
    geom = "curve",
    x = 250, y = 75,
    xend = 180, yend = 160,
    curvature = .8,
    angle = 130,
    arrow = arrow(length = unit(0.4, "lines"),
                  type = "closed",
                  ends = "both")
  )
```



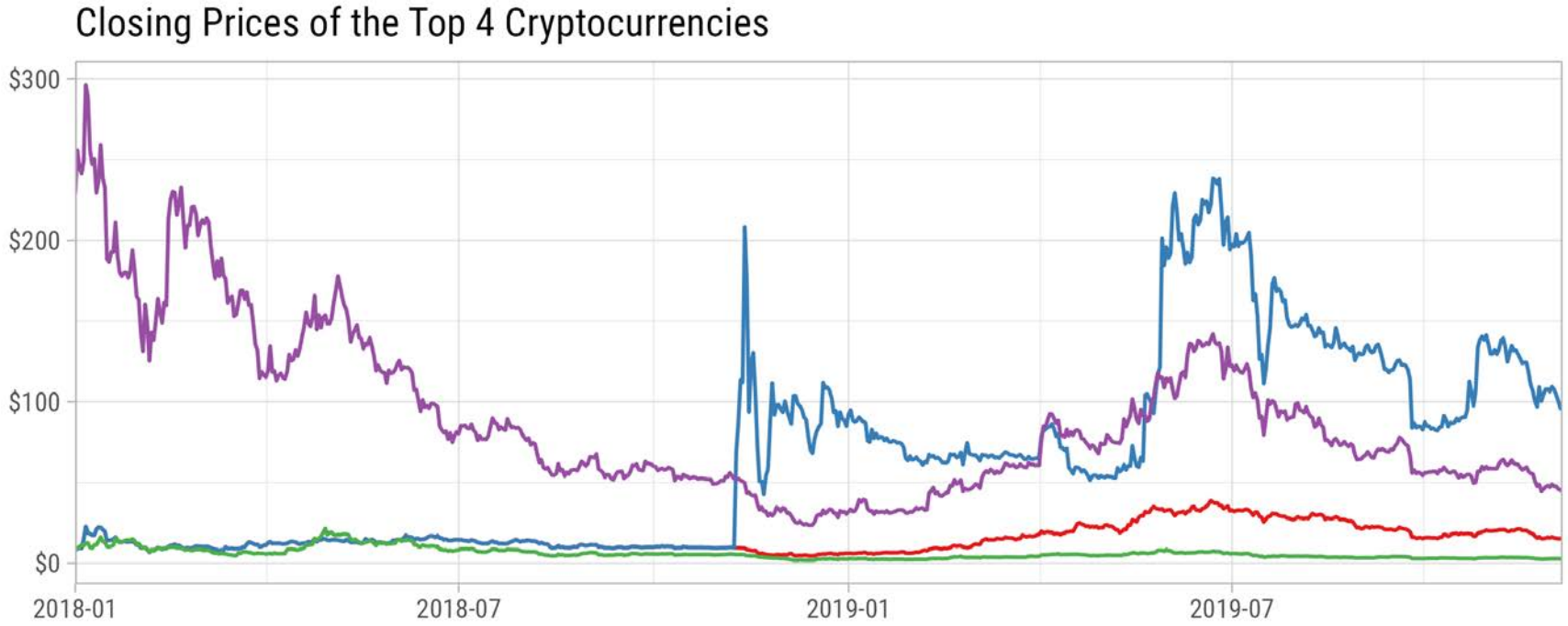
Exercise 1:

- Create the following visualization:



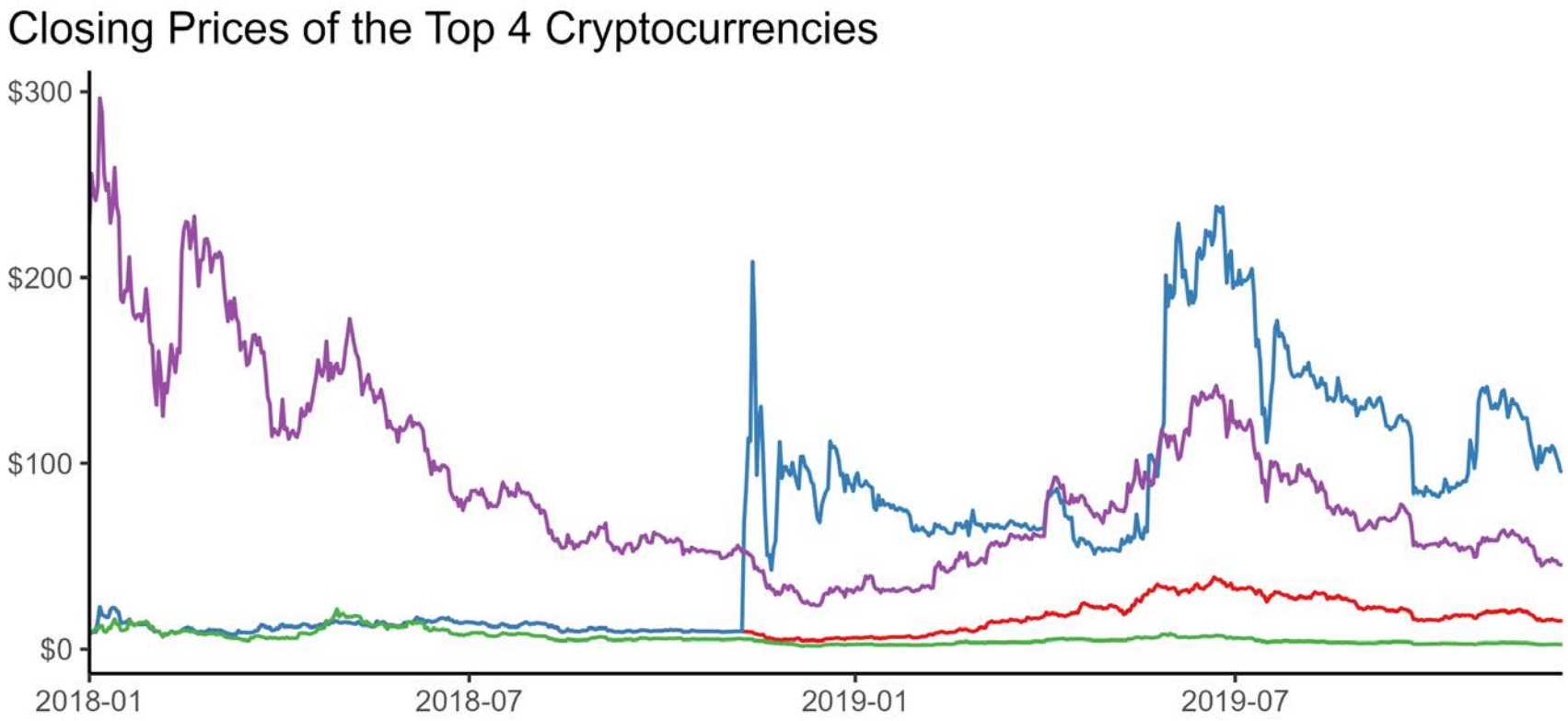
Exercise 1: Create Line Plot

```
(g <- ggplot(data, aes(date, close, color = currency)) +
  geom_line(size = 1) +
  scale_x_date(expand = c(0, 0)) +
  scale_y_continuous(labels = scales::dollar_format()) +
  scale_color_brewer(palette = "Set1", guide = "none") +
  labs(x = NULL, y = NULL, title = "Closing Prices of the Top 4 Cryptocurrencies"))
```



Exercise 1: Modify Theme

```
(g <- g +
  theme_classic(base_size = 20) +
  theme(plot.title.position = "plot"))
```

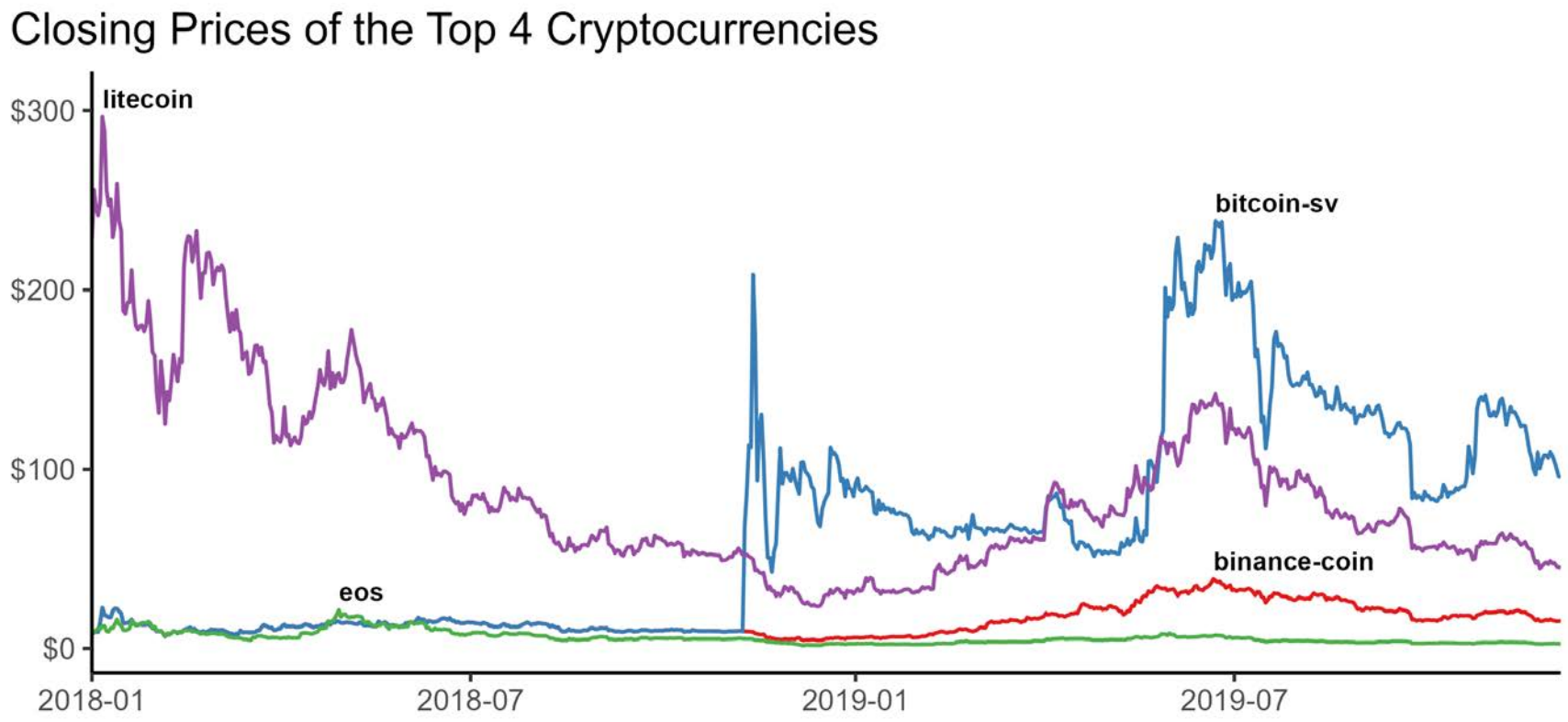


Exercise 1: Add Annotations with `annotate()`

```
annotate <- data %>% group_by(currency) %>% filter(close == max(close))

annotate
## # A tibble: 4 x 9
## # Groups:   currency [4]
##   currency     date        open  high   low close  year month  yday
##   <chr>        <date>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 binance-coin 2019-06-21  36.7  38.9  36.3  38.8  2019     6   172
## 2 eos          2018-04-29  18.9  22.9  18.9  21.5  2018     4   119
## 3 litecoin     2018-01-06 250.  323.  249.  296.   2018     1     6
## 4 bitcoin-sv   2019-06-22 222.  256.  222.  238.   2019     6   173
```

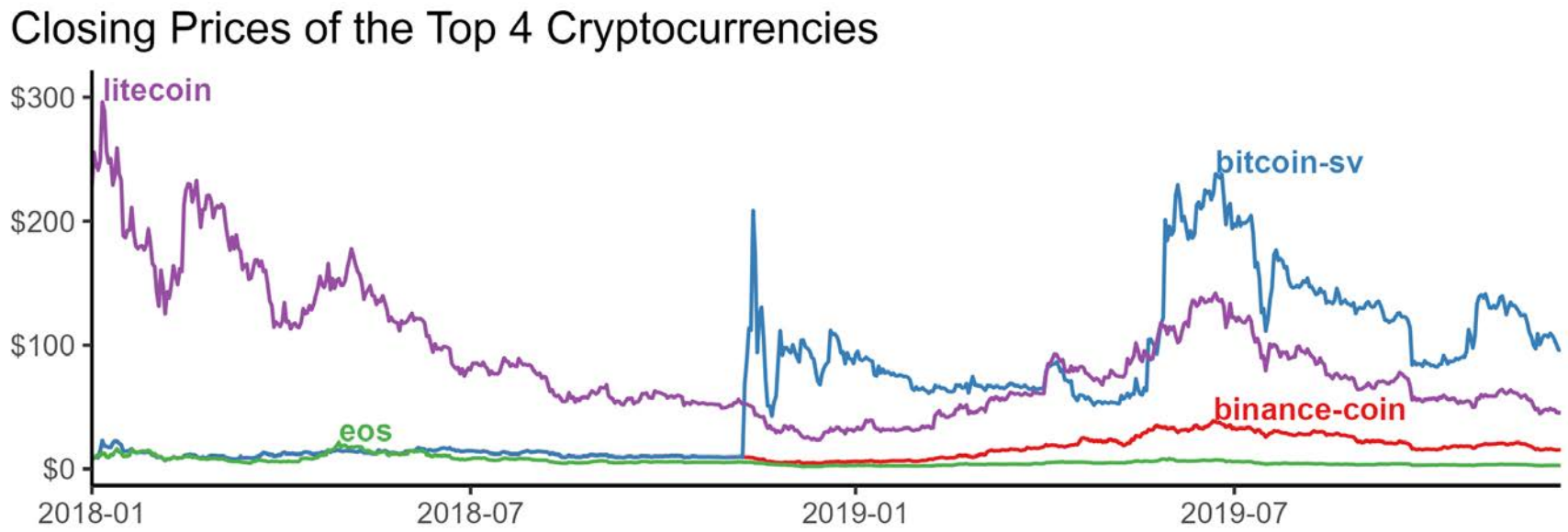
Exercise 1: Add Annotations with `annotate()`

```
g +
  annotate(geom = "text", x = annotate$date, y = annotate$close + 10,
           label = annotate$currency, hjust = 0, size = 5, fontface = "bold")
```



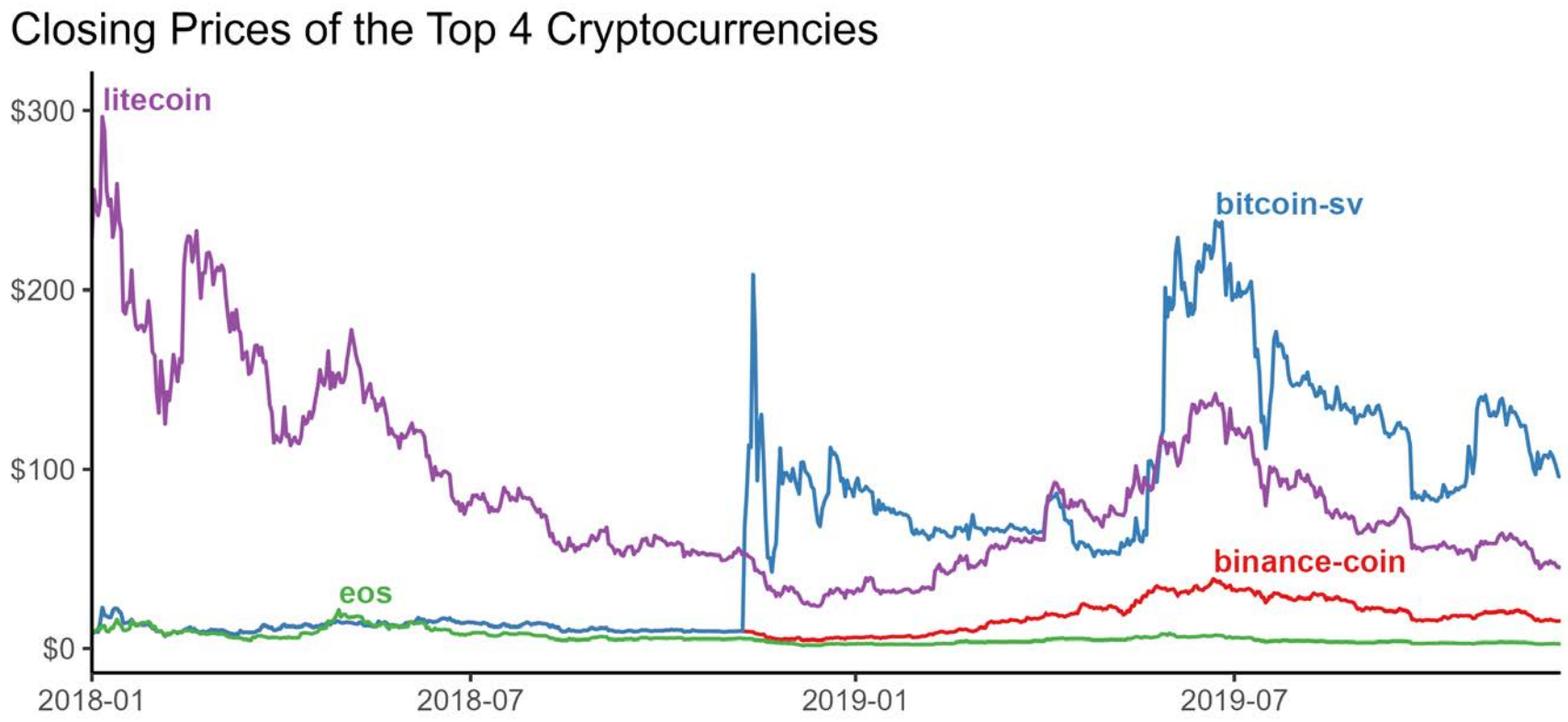
Exercise 1: Add Annotations with `annotate()`

```
g +
  annotate(geom = "text", x = annotate$date[1], y = annotate$close[1] + 10,
           label = annotate$currency[1], hjust = 0, size = 6, fontface = "bold", color = "#E41A1C") +
  annotate(geom = "text", x = annotate$date[2], y = annotate$close[2] + 10,
           label = annotate$currency[2], hjust = 0, size = 6, fontface = "bold", color = "#4DAF4A") +
  annotate(geom = "text", x = annotate$date[3], y = annotate$close[3] + 10,
           label = annotate$currency[3], hjust = 0, size = 6, fontface = "bold", color = "#984EA3") +
  annotate(geom = "text", x = annotate$date[4], y = annotate$close[4] + 10,
           label = annotate$currency[4], hjust = 0, size = 6, fontface = "bold", color = "#377EB8")
```



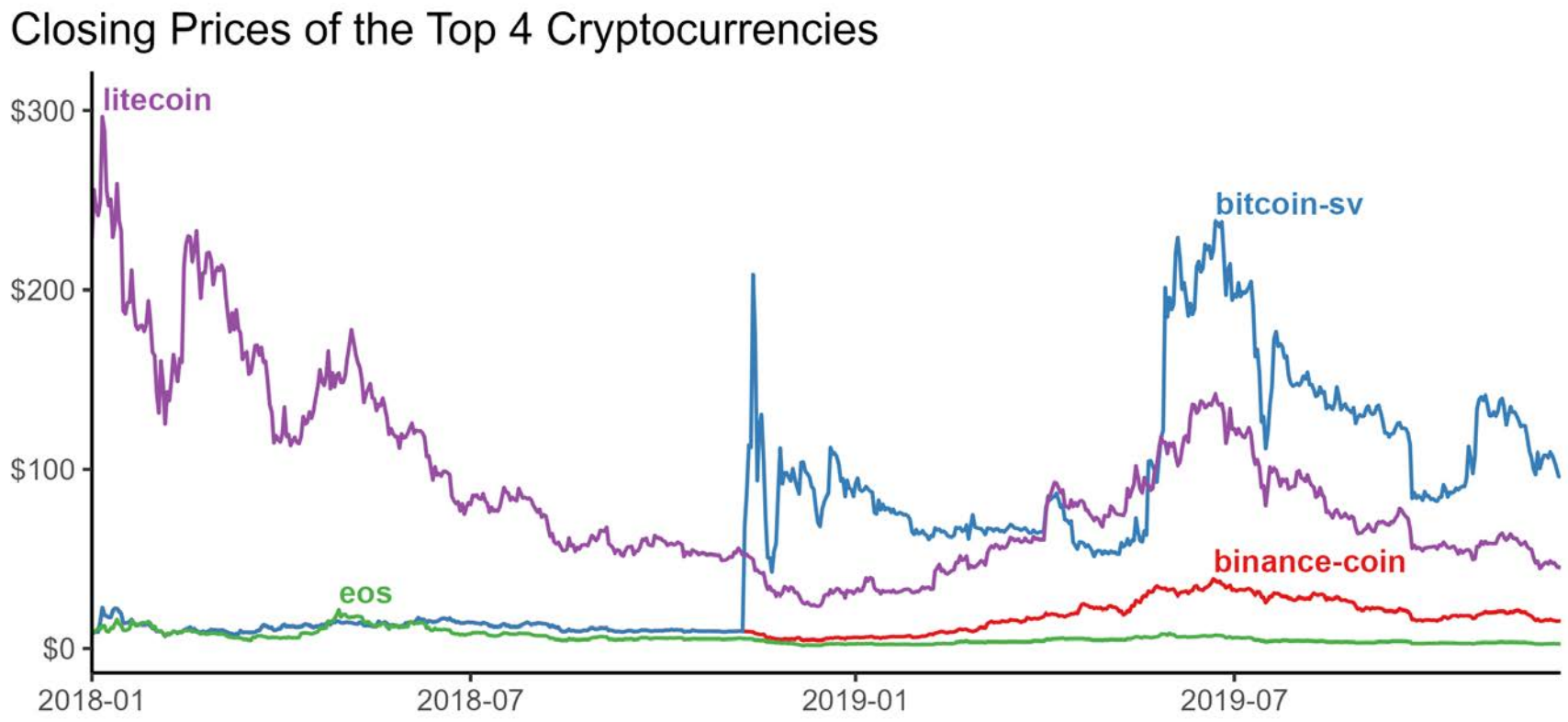
Exercise 1: Add Annotations with `geom_text|label()`

```
g +
  geom_text(data = annotate,
            aes(label = currency), size = 6, nudge_y = 10, hjust = 0, fontface = "bold")
```



Exercise 1: Add Annotations with `geom_text|label()`

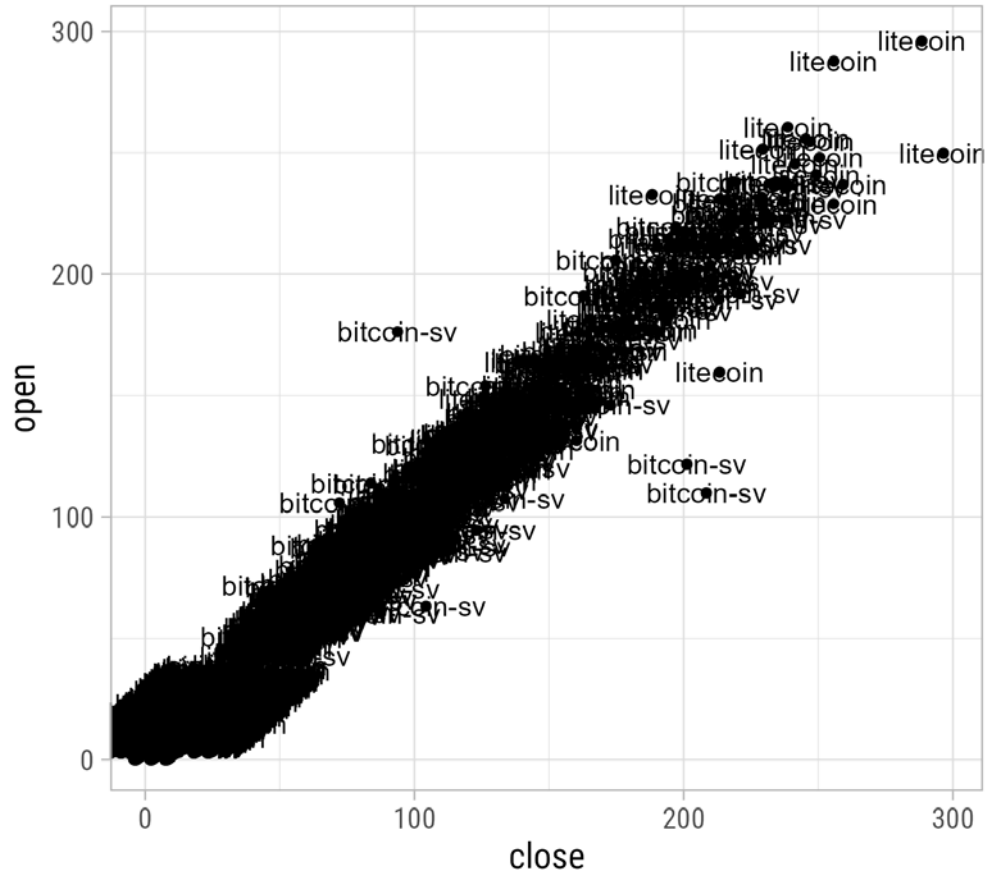
```
g +
  geom_text(data = data %>% group_by(currency) %>% filter(close == max(close)),
            aes(label = currency), size = 6, nudge_y = 10, hjust = 0, fontface = "bold")
```



Annotations via `geom_text|label()`

You already know `geom_text()`:

```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  geom_text(
    aes(label = currency),
    size = 5
  )
```



Annotations via `geom_text|label()`

Let's tag only those days that are considerably far from the diagonal:

```
outliers <- data %>%
  mutate(dist = abs(close - open) / sqrt(2)) %>%
  filter(dist > 20)
```

Annotations via `geom_text|label()`

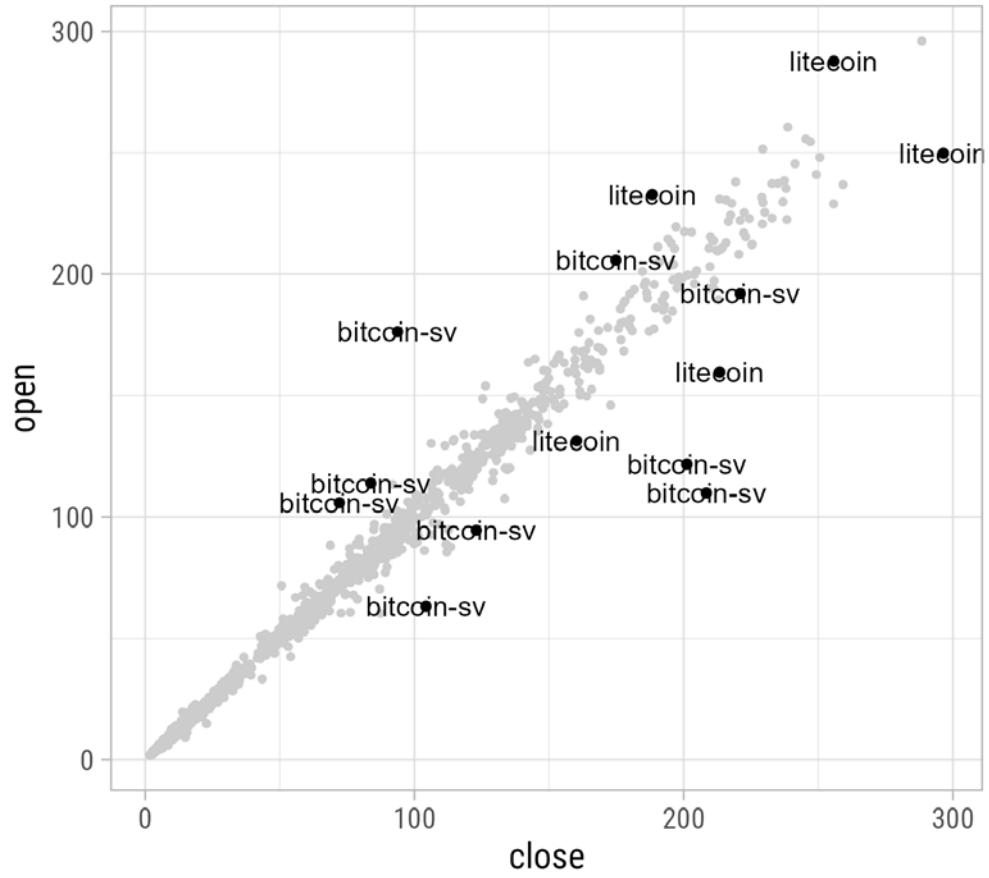
Let's tag only those days that are considerably far from the diagonal:

```
outliers <- data %>%
  mutate(dist = abs(close - open) / sqrt(2)) %>%
  filter(dist > 20)

outliers
## # A tibble: 14 x 10
##    currency   date        open  high   low close  year month  yday  dist
##    <chr>      <date>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 litecoin   2018-02-14 160.   218. 160.  213.   2018     2    45  38.0
##  2 litecoin   2018-02-03 131.   166. 121.  160.   2018     2    34  20.4
##  3 litecoin   2018-01-16 233.   233. 152.  188.   2018     1    16  31.5
##  4 litecoin   2018-01-08 288.   289. 238.  256.   2018     1     8  22.6
##  5 litecoin   2018-01-06 250.   323. 249.  296.   2018     1     6  33.0
##  6 bitcoin-sv 2019-09-24 114.   115.  71.0  83.8  2019     9   267  21.3
##  7 bitcoin-sv 2019-06-03 192.   236. 192.  221    2019     6   154  20.5
##  8 bitcoin-sv 2019-05-29 122.   201. 115.  201.   2019     5   149  56.2
##  9 bitcoin-sv 2019-05-21  63.2  139.  61.7 104.   2019     5   141  29.1
## 10 bitcoin-sv 2018-11-19 106.   108.  69.7  72.1  2018    11   323  23.9
## 11 bitcoin-sv 2018-11-16  94.5  134.  76.8 123.   2018    11   320  20.2
## 12 bitcoin-sv 2018-11-15 176.   180.  71.0  93.7  2018    11   319  58.4
## 13 bitcoin-sv 2018-11-14 206.   244. 155.  175.   2018    11   318  21.9
```

Annotations via `geom_text|label()`

We can use both data sets in the same ggplot:

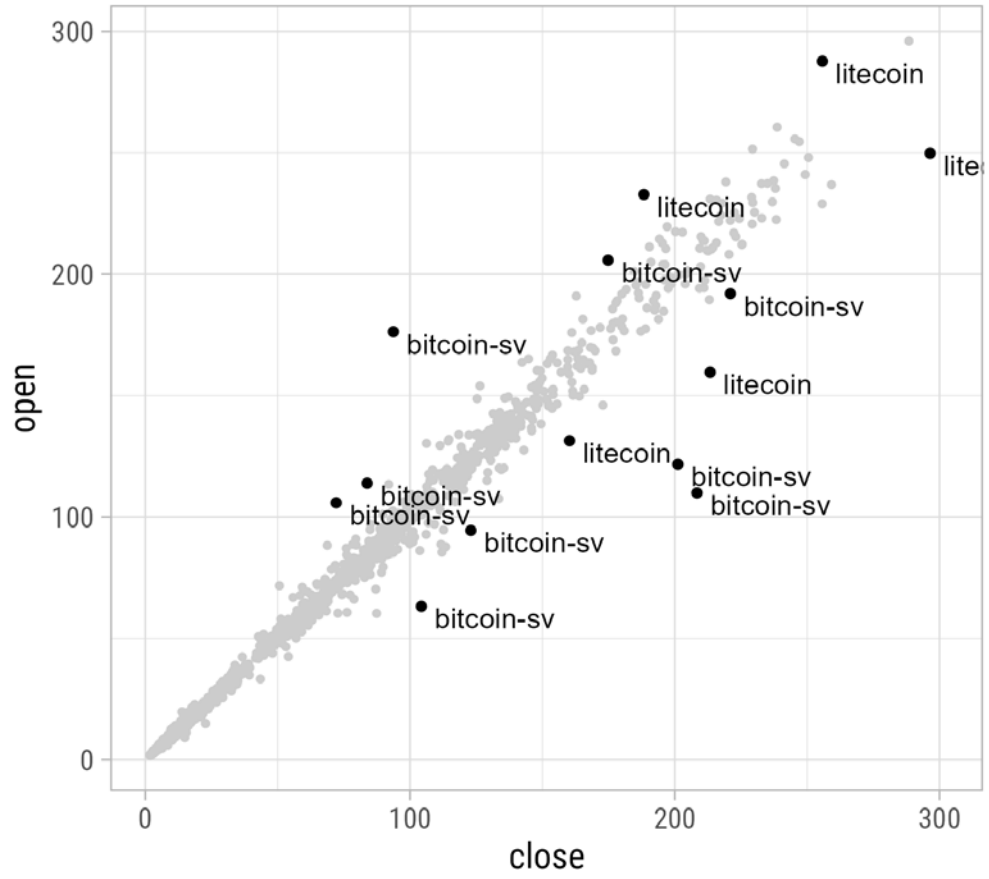
```
ggplot(outliers, aes(close, open)) +
  geom_point(data = data, color = "grey80") +
  geom_point(size = 2) +
  geom_text(
    aes(label = currency),
    size = 5
  )
```



Annotations via `geom_text|label()`

We can offset the labels with the help of the nudge arguments:

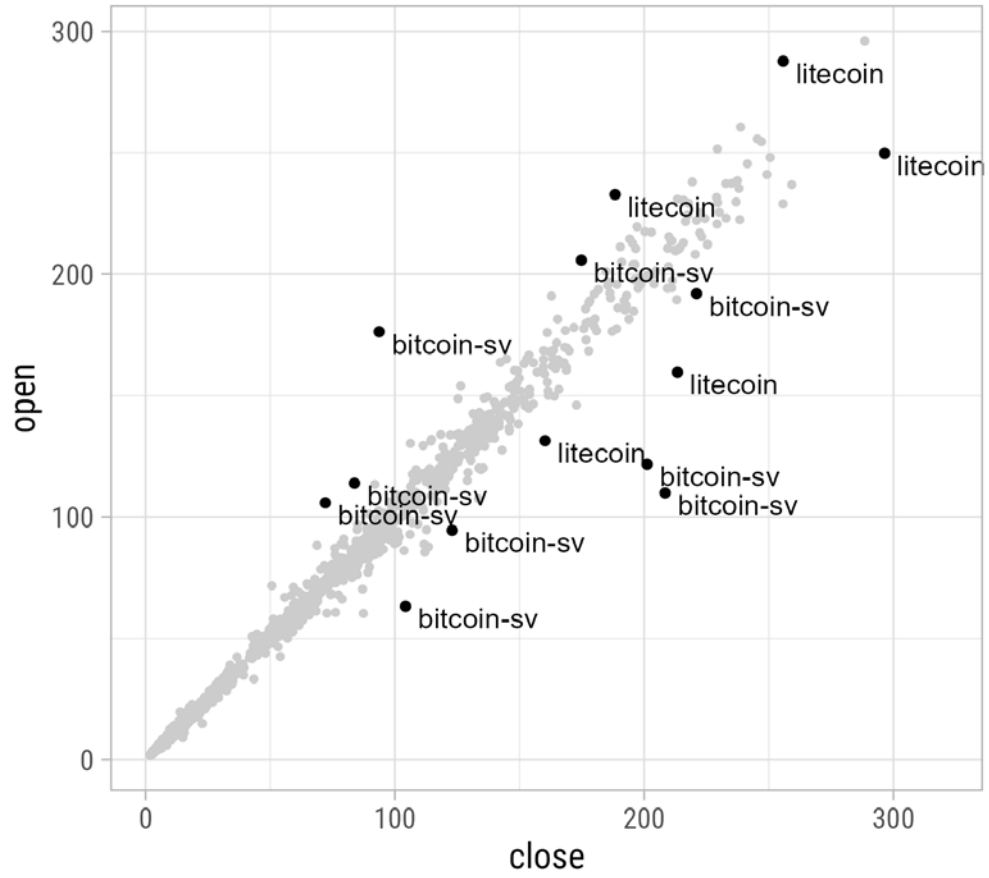
```
ggplot(outliers, aes(close, open)) +
  geom_point(data = data, color = "grey80") +
  geom_point(size = 2) +
  geom_text(
    aes(label = currency),
    size = 5,
    hjust = 0,
    nudge_x = 5,
    nudge_y = -5
  )
```



Annotations via `geom_text|label()`

We can offset the labels with the help of the nudge arguments:

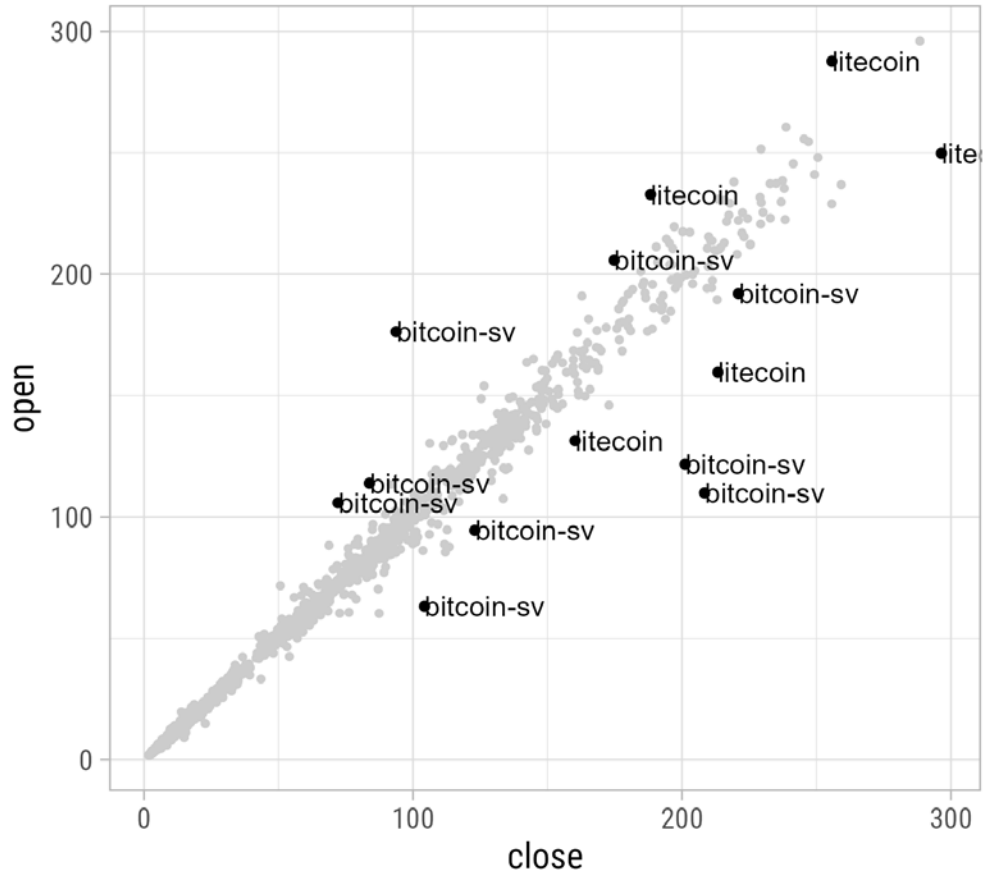
```
ggplot(outliers, aes(close, open)) +
  geom_point(data = data, color = "grey80") +
  geom_point(size = 2) +
  geom_text(
    aes(label = currency),
    size = 5,
    hjust = 0,
    nudge_x = 5,
    nudge_y = -5
  ) +
  scale_x_continuous(
    limits = c(NA, 320)
  )
```

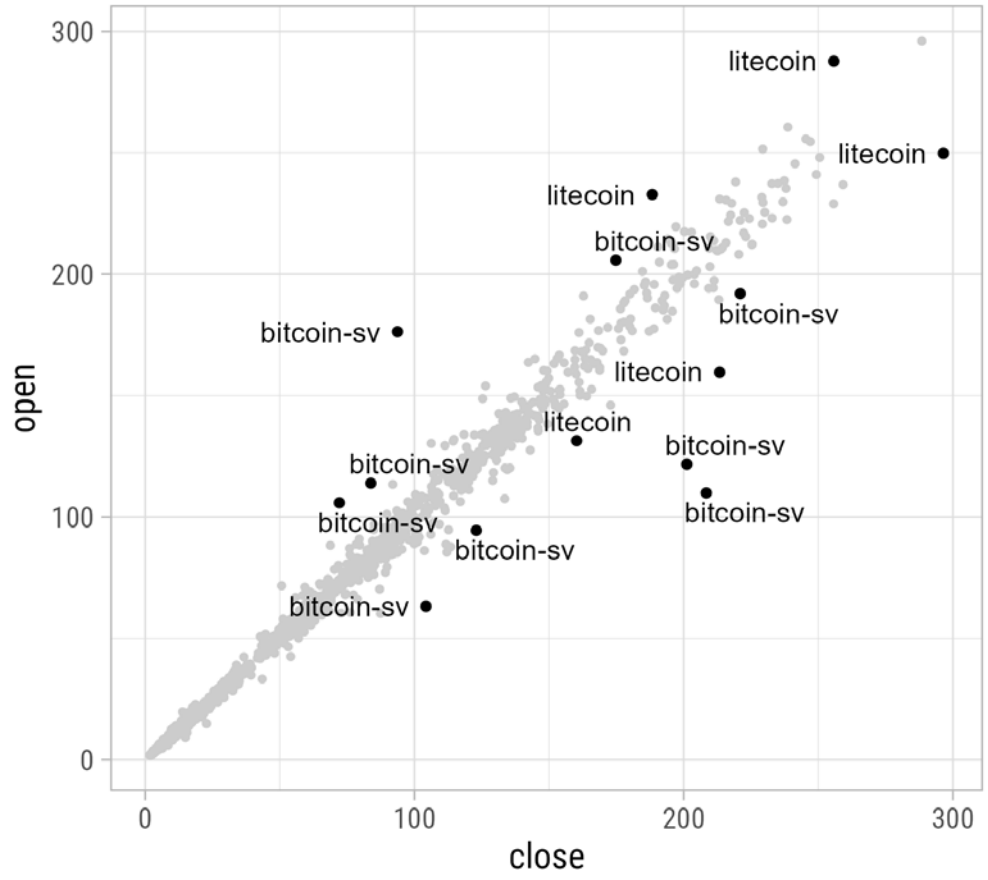


Advanced Text Labelling

Annotations via `geom_text|label_repel()`

The `{ggrepel}` package provides two geom's to repel overlapping text labels:

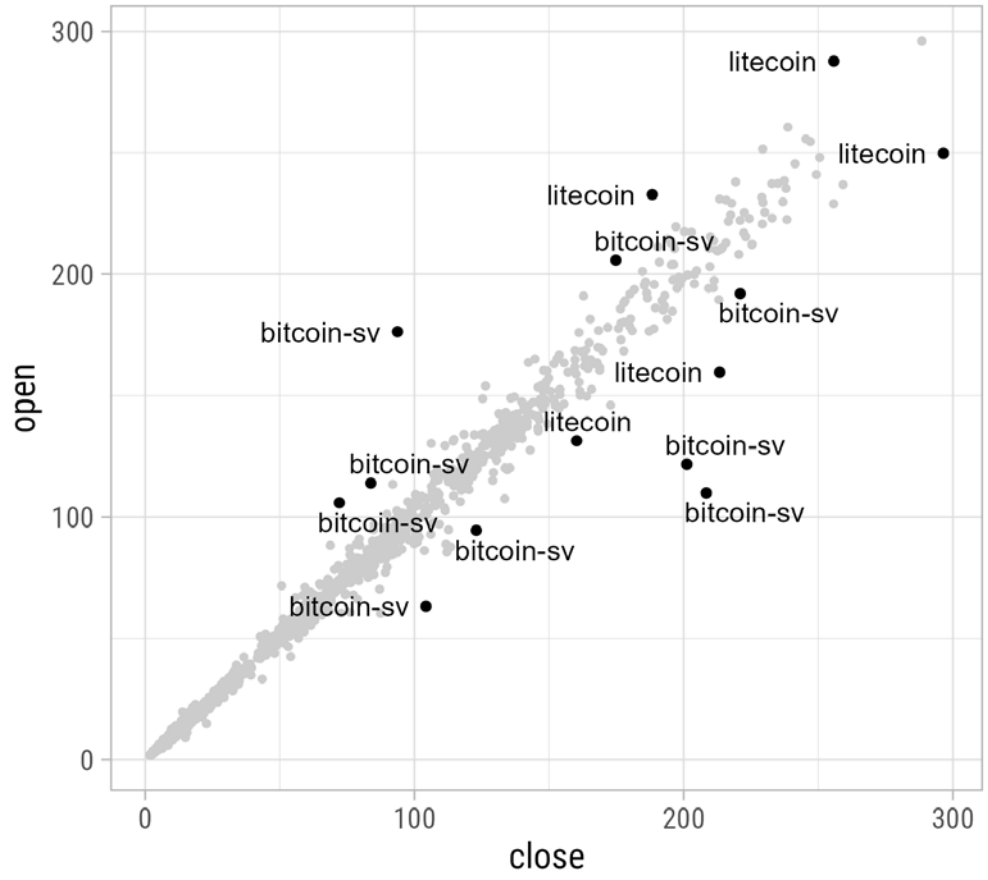




Annotations via `geom_text|label_repel()`

The {ggrepel} package provides two geom's to repel overlapping text labels:

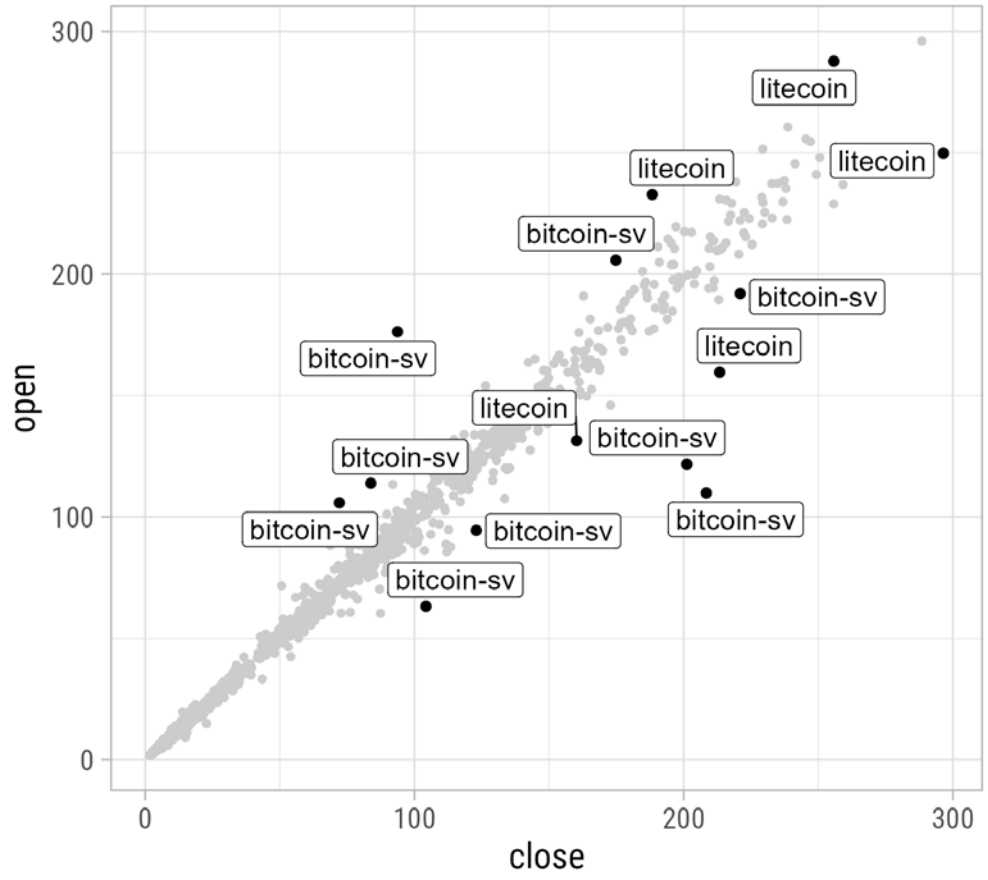
```
#install.packages("ggrepel")

ggplot(outliers, aes(close, open)) +
  geom_point(data = data, color = "grey80") +
  geom_point(size = 2) +
  ggrepel::geom_text_repel(
    aes(label = currency),
    size = 5,
    hjust = 0
  )
```



Annotations via `geom_text|label_repel()`

The {ggrepel} package provides two geom's to repel overlapping text labels:

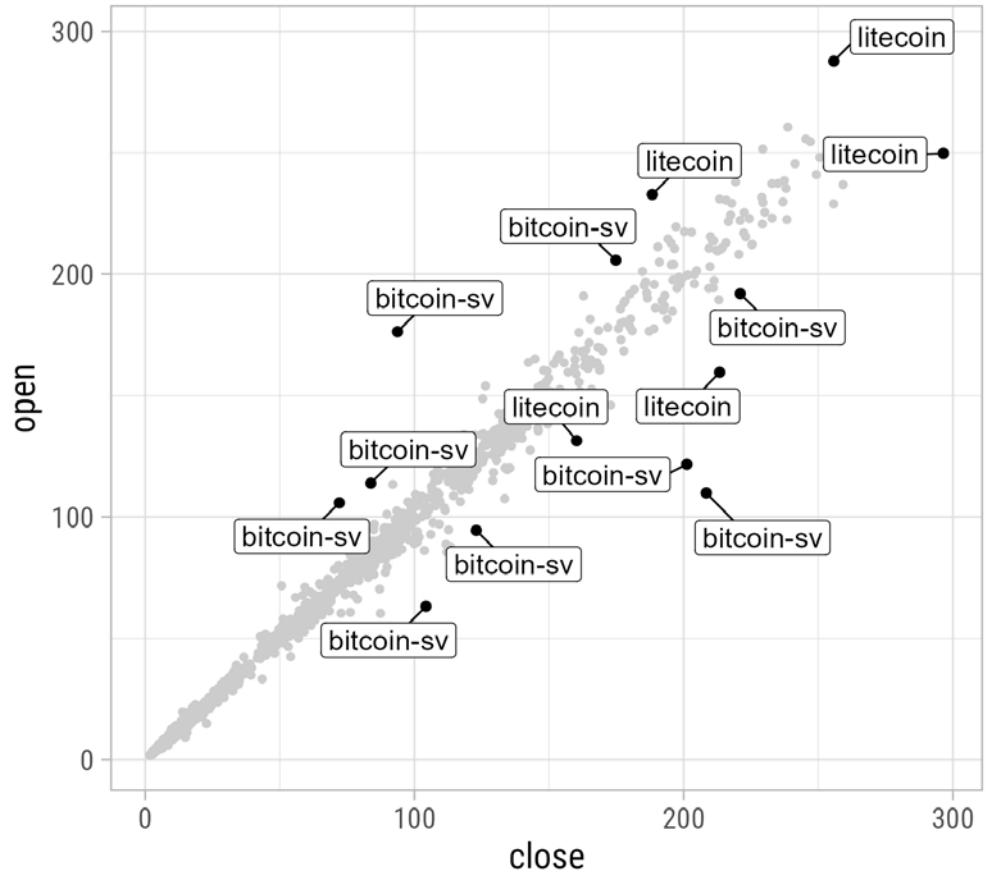
```
#install.packages("ggrepel")

ggplot(outliers, aes(close, open)) +
  geom_point(data = data, color = "grey80") +
  geom_point(size = 2) +
  ggrepel::geom_label_repel(
    aes(label = currency),
    size = 5
  )
```



Annotations via `geom_text|label_repel()`

The `{ggrepel}` package provides two geom's to repel overlapping text labels:

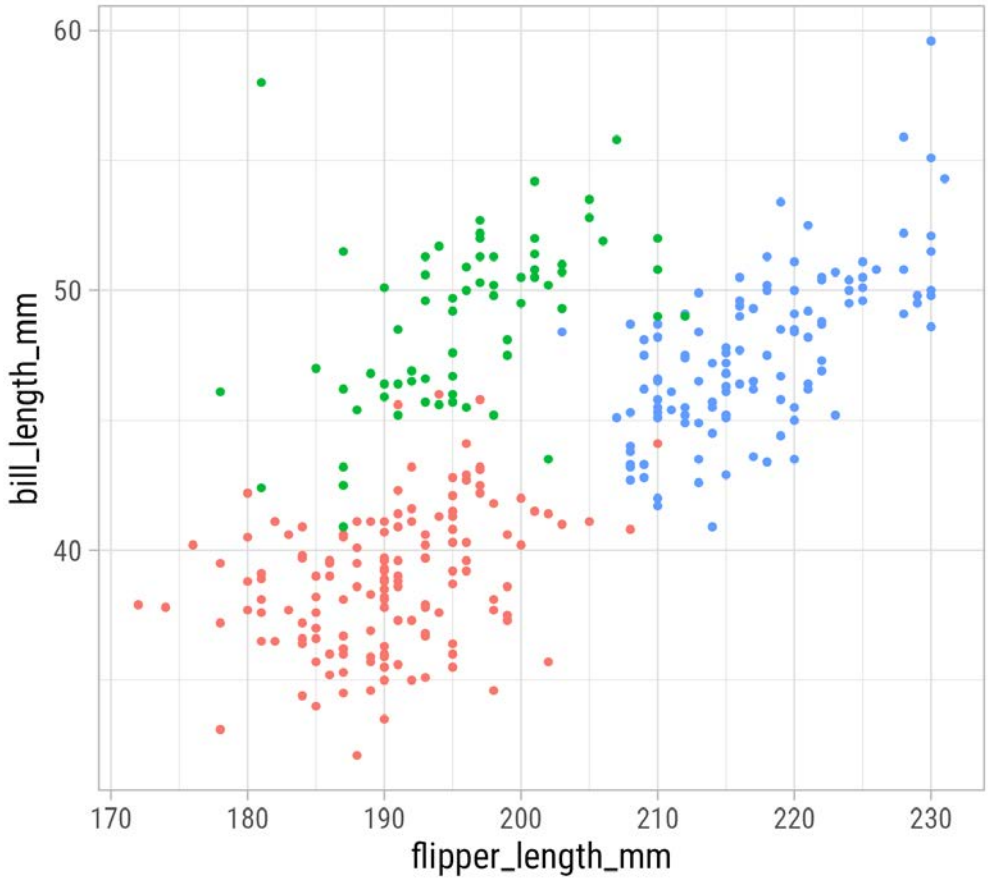
```
#install.packages("ggrepel")

ggplot(outliers, aes(close, open)) +
  geom_point(data = data, color = "grey80") +
  geom_point(size = 2) +
  ggrepel::geom_label_repel(
    aes(label = currency),
    size = 5,
    ## space between points + labels
    box.padding = .5,
    ## always draw segments
    min.segment.length = 0
  )
```



Annotations via `geom_mark_*()`

For illustration purposes we use a data set about Palmer penguins (Horst, Hill & Gorman 2020).

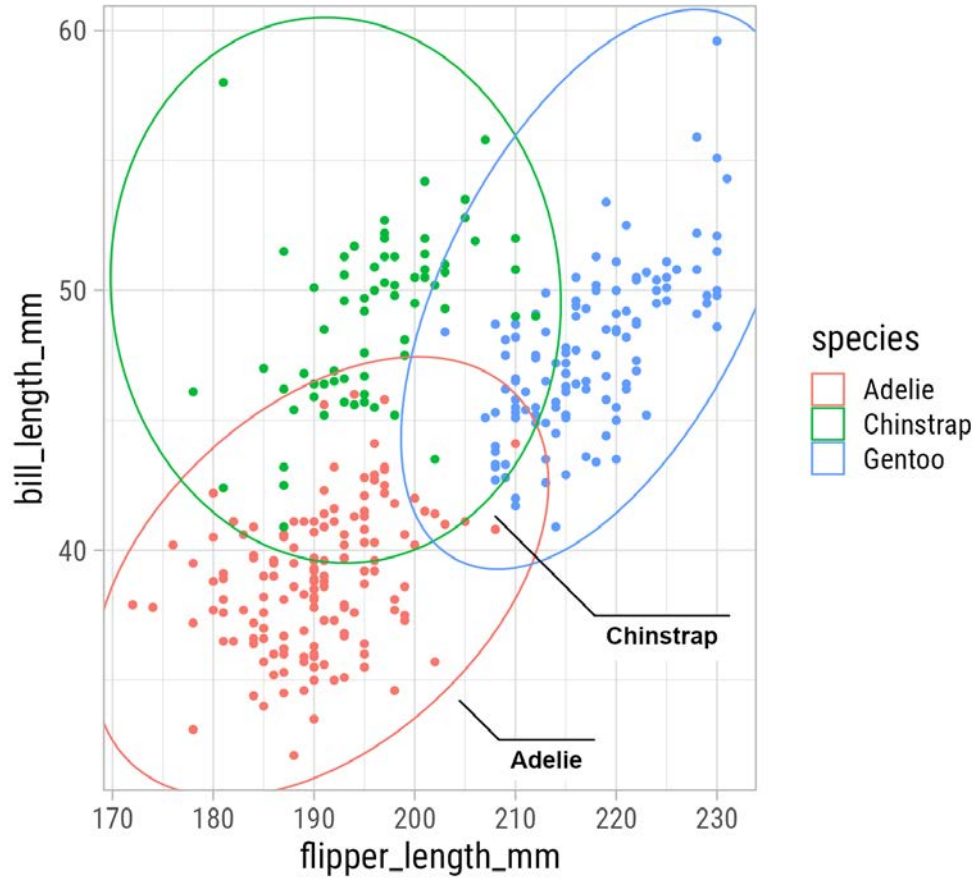
```
penguins <- read_csv(
  here::here("data", "penguins.csv")
)

(g <-
  ggplot(
    penguins,
    aes(flipper_length_mm, bill_length_mm,
        color = species)
    ) +
    geom_point(show.legend = FALSE)
)
```



Annotations via `geom_mark_*()`

The `{ggforce}` package provides functionality to highlight groups:

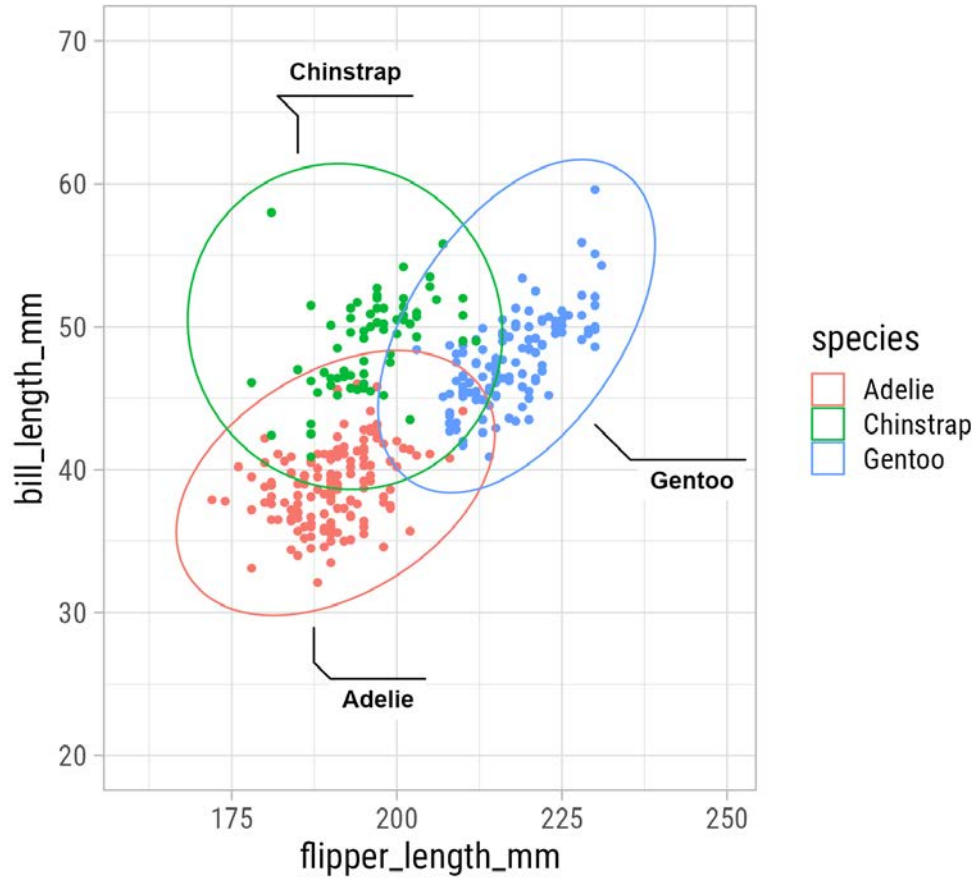
```
#install.packages("ggforce")

g +
  ggforce::geom_mark_ellipse(
    aes(label = species)
  )
```



Annotations via `geom_mark_*()`

The `{ggforce}` package provides functionality to highlight groups:

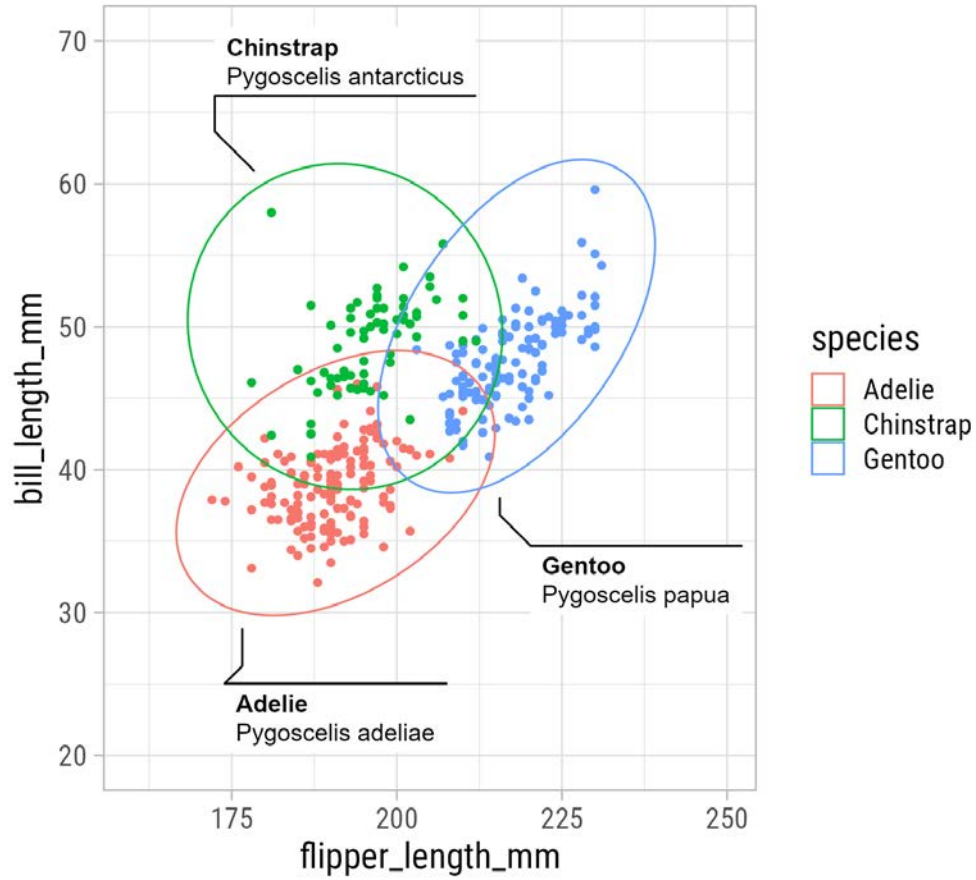
```
g +
  ggforce::geom_mark_ellipse(
    aes(label = species)
  ) +
  coord_cartesian(
    xlim = c(160, 250),
    ylim = c(20, 70)
  )
```



Annotations via `geom_mark_*()`

The `{ggforce}` package provides functionality to highlight groups:

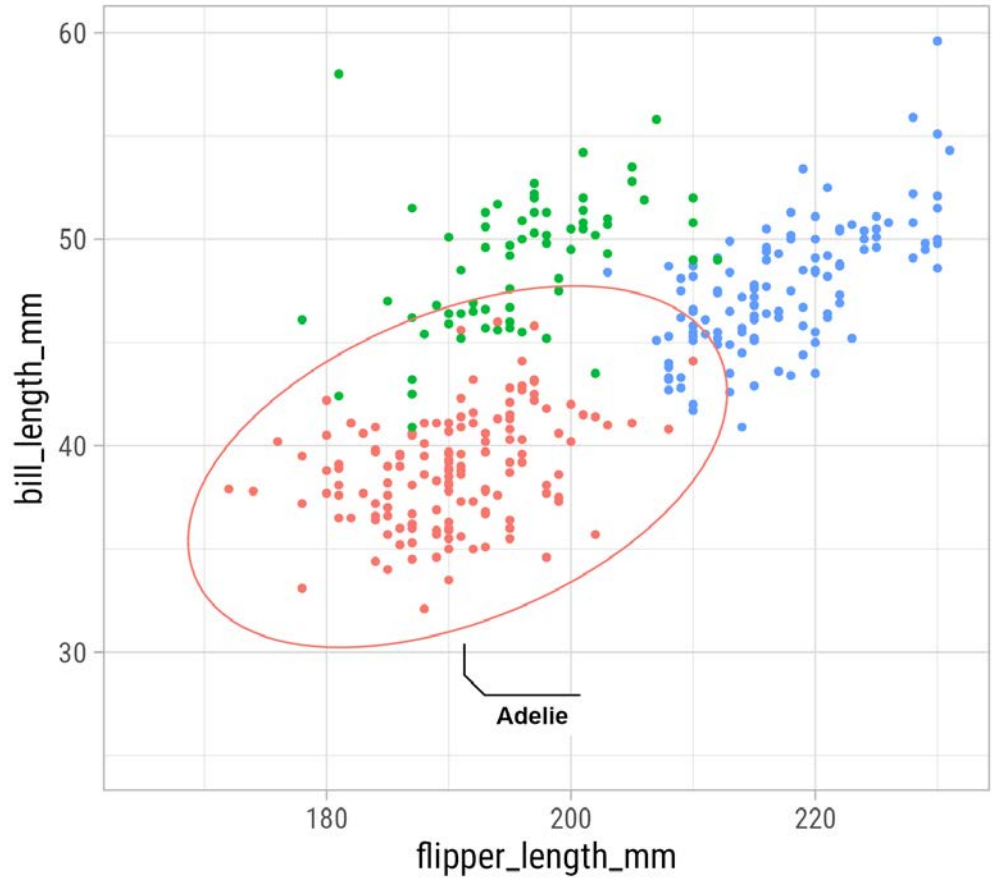
```
g +
  ggforce::geom_mark_ellipse(
    aes(label = species,
        description = latin_name)
  ) +
  coord_cartesian(
    xlim = c(160, 250),
    ylim = c(20, 70)
  )
```



Annotations via `geom_mark_*()`

The {ggforce} package provides functionality to highlight groups:

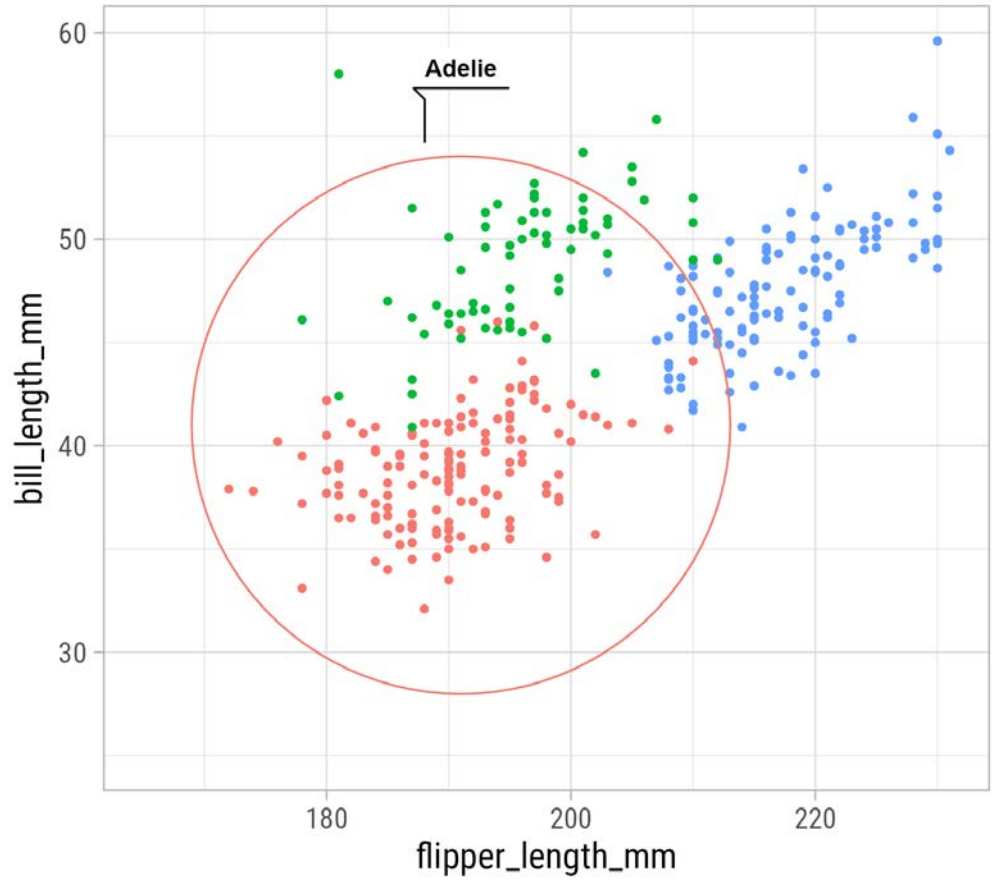
```
g +
  ggforce::geom_mark_ellipse(
    aes(label = species,
        filter = species == "Adelie")
  ) +
  coord_cartesian(
    xlim = c(165, NA),
    ylim = c(25, NA)
  ) +
  theme(legend.position = "none")
```



Annotations via `geom_mark_*()`

The {ggforce} package provides functionality to highlight groups:

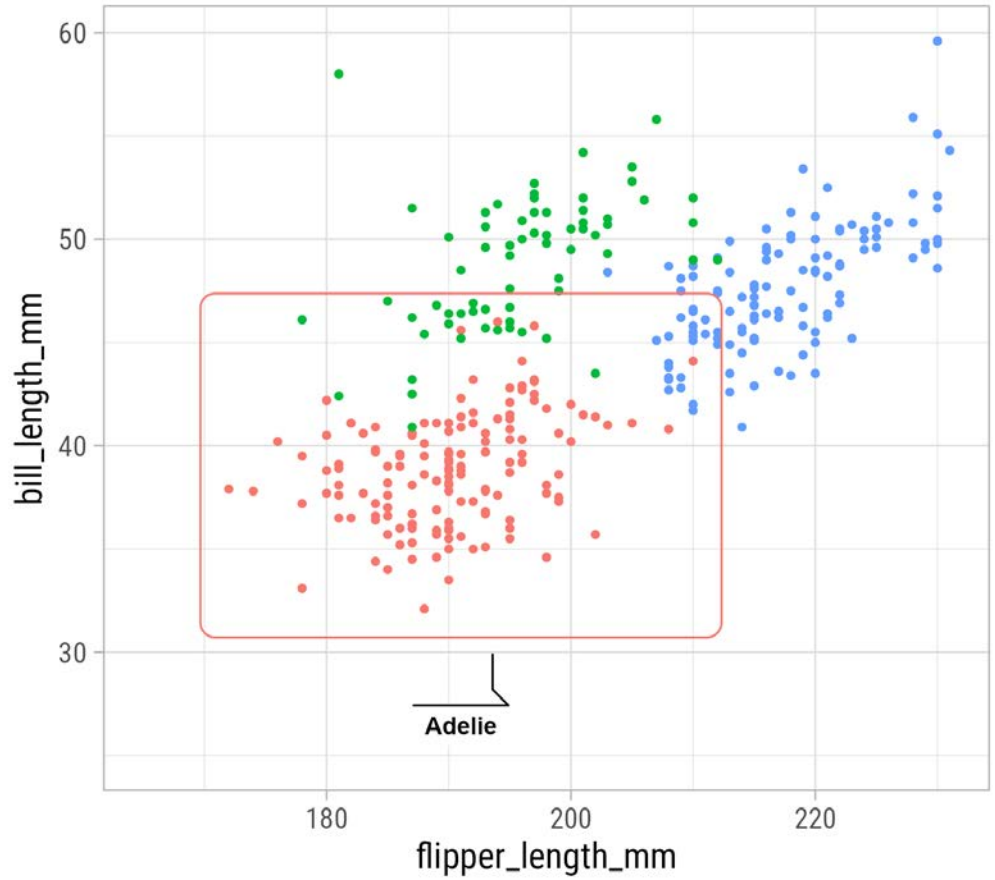
```
g +
  ggforce::geom_mark_circle(
    aes(label = species,
        filter = species == "Adelie")
  ) +
  coord_cartesian(
    xlim = c(165, NA),
    ylim = c(25, NA)
  ) +
  theme(legend.position = "none")
```



Annotations via `geom_mark_*()`

The {ggforce} package provides functionality to highlight groups:

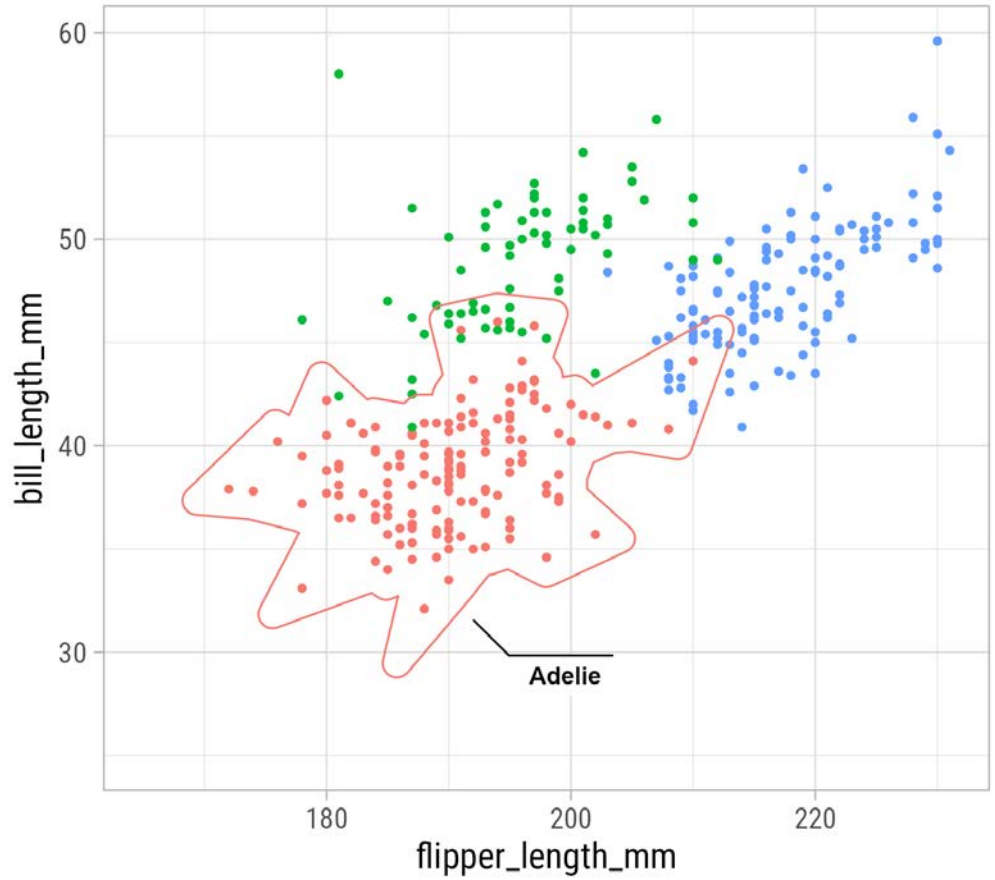
```
g +
  ggforce::geom_mark_rect(
    aes(label = species,
        filter = species == "Adelie")
  ) +
  coord_cartesian(
    xlim = c(165, NA),
    ylim = c(25, NA)
  ) +
  theme(legend.position = "none")
```



Annotations via `geom_mark_*()`

The {ggforce} package provides functionality to highlight groups:

```
g +
  ggforce::geom_mark_hull(
    aes(label = species,
        filter = species == "Adelie")
  ) +
  coord_cartesian(
    xlim = c(165, NA),
    ylim = c(25, NA)
  ) +
  theme(legend.position = "none")
```



Add Images

Add Images via `annotation_custom()`

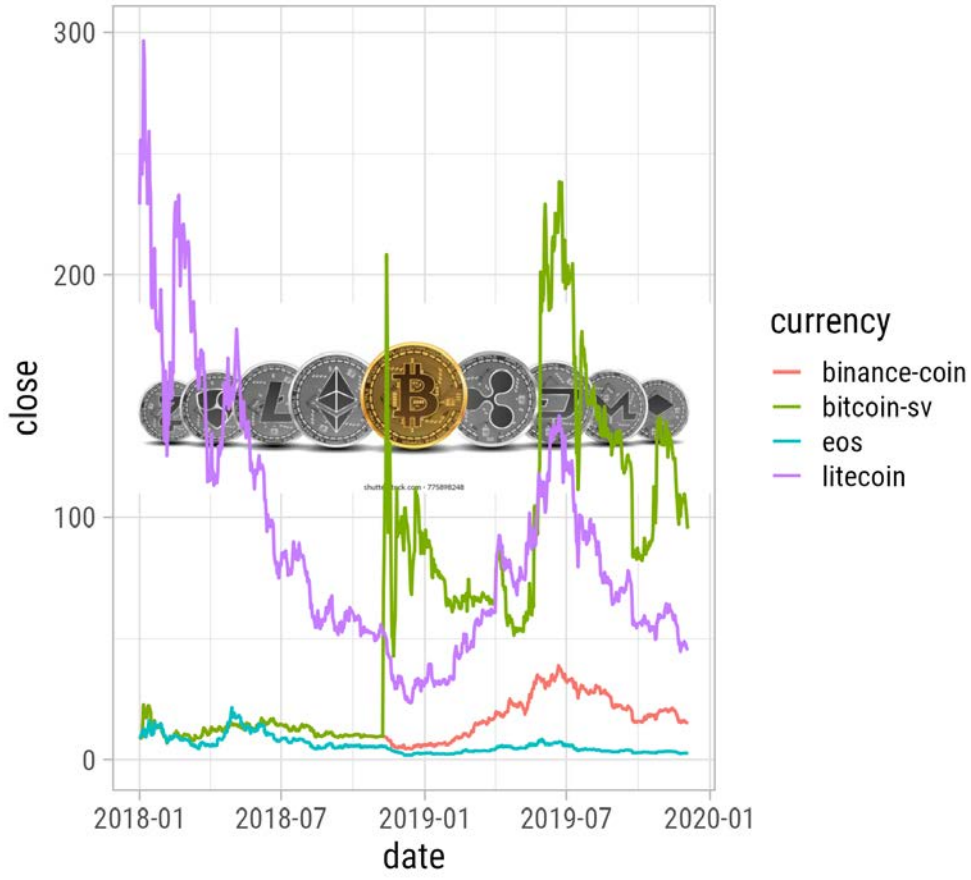
`magick::image_read()` allows to directly store images from the web in R:

```
#install.packages("magick")
url <- "https://image.shutterstock.com/image-vector/set-gold-silver-crypto-currencies-260nw-775898248.jpg"
img <- magick::image_read(url)

img
```

Add Images via `annotation_custom()`

`annotation_custom()` in combination with `grid::rasterGrob()` allows to add images:

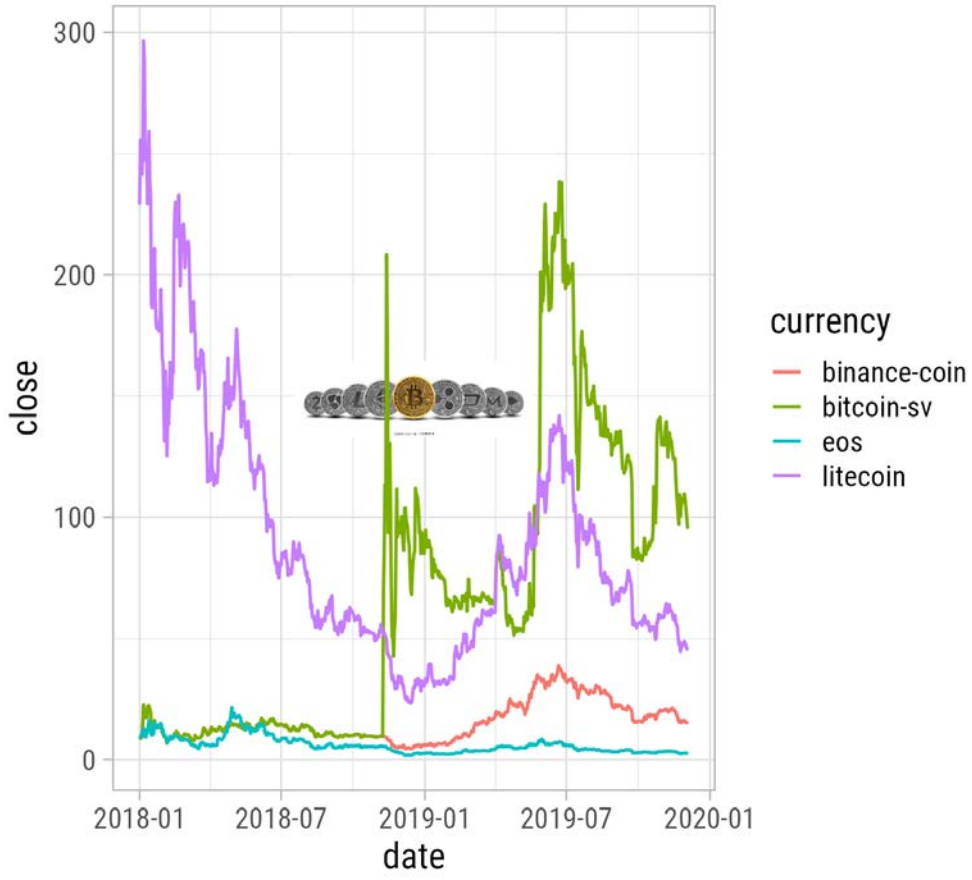
```
ggplot(data, aes(date, close, color = currency)) +
  annotation_custom(
    grid::rasterGrob(
      image = img
    )
  ) +
  geom_line(size = .8)
```



Add Images via `annotation_custom()`

`annotation_custom()` in combination with `grid::rasterGrob()` allows to add images:

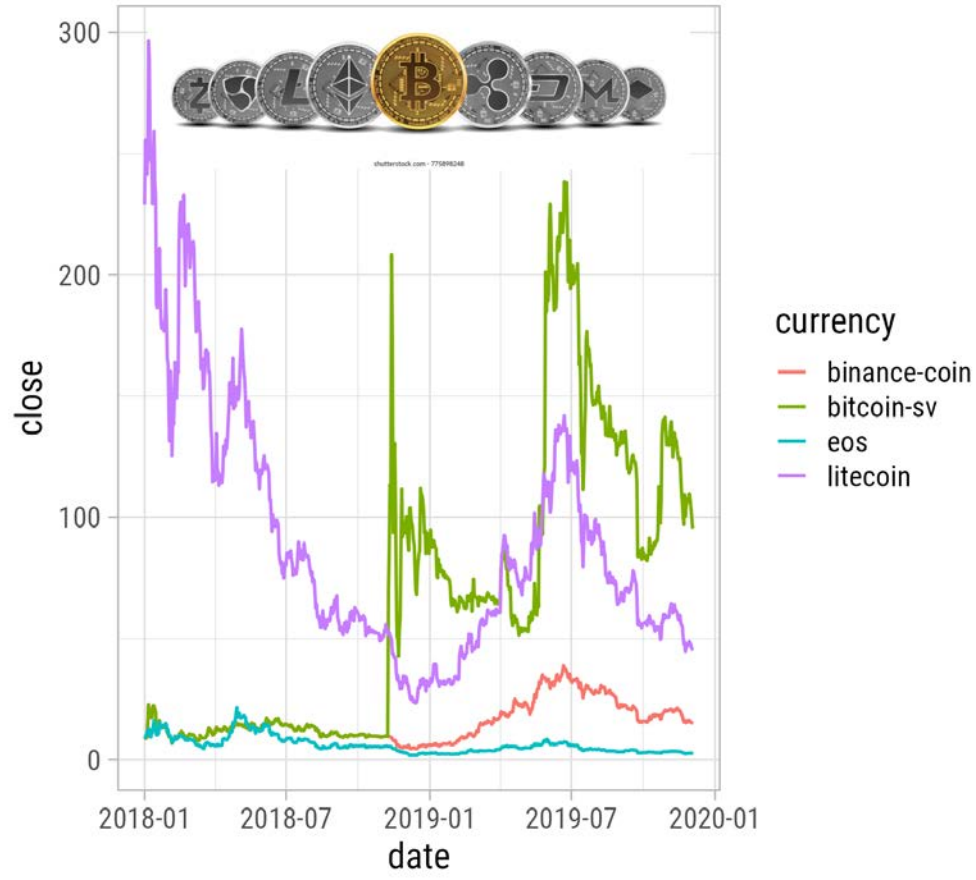
```
ggplot(data, aes(date, close, color = currency)) +
  annotation_custom(
    grid::rasterGrob(
      image = img,
      width = unit(.4, "npc")
    )
  ) +
  geom_line(size = .8)
```



Add Images via `annotation_custom()`

`annotation_custom()` in combination with `grid::rasterGrob()` allows to add images:

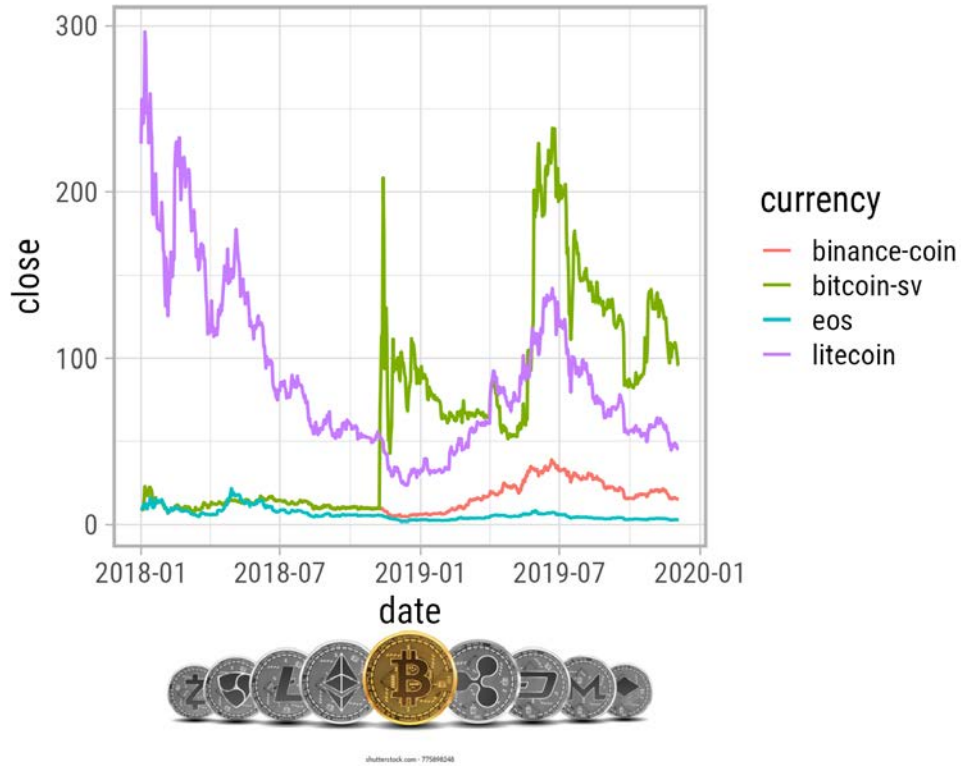
```
ggplot(data, aes(date, close, color = currency)) +
  annotation_custom(
    grid::rasterGrob(
      image = img,
      x = .5,
      y = .9,
      width = .9
    )
  ) +
  geom_line(size = .8)
```



Add Images via `annotation_custom()`

`annotation_custom()` in combination with `grid::rasterGrob()` allows to add images:

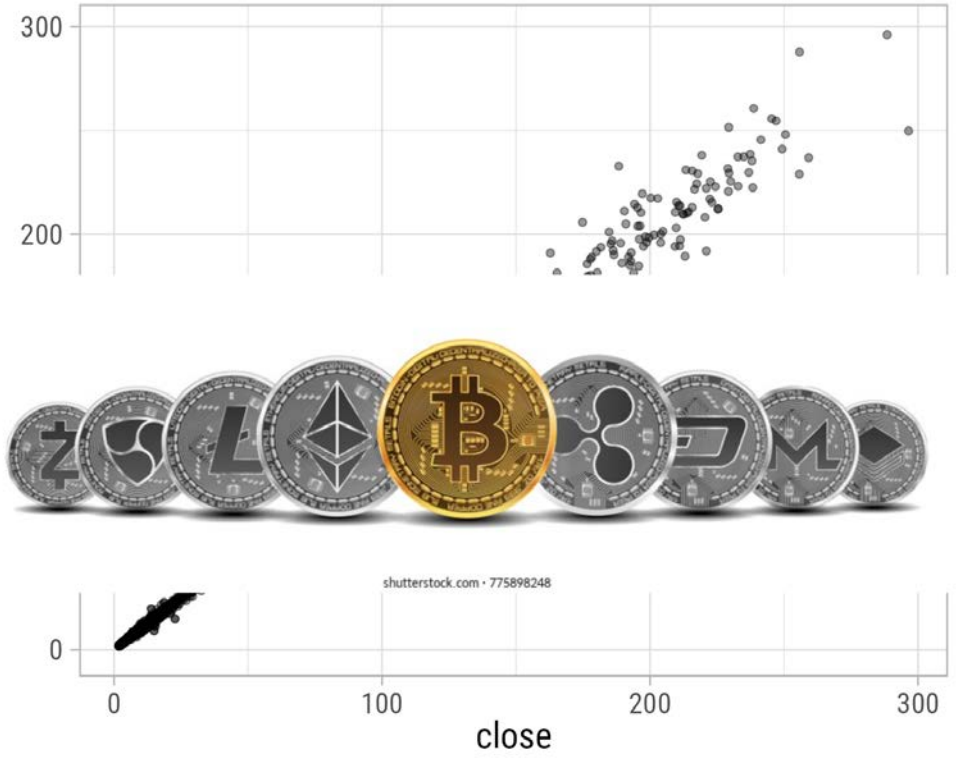
```
ggplot(data, aes(date, close, color = currency)) +
  annotation_custom(
    grid::rasterGrob(
        image = img,
        x = .5,
        y = -.25,
        width = .9
    )
  ) +
  geom_line(size = .8) +
  coord_cartesian(clip = "off") +
  theme(plot.margin = margin(12, 12, 130, 12))
```



Add Images with the {cowplot} Package

The cowplot package can also be used to add images:

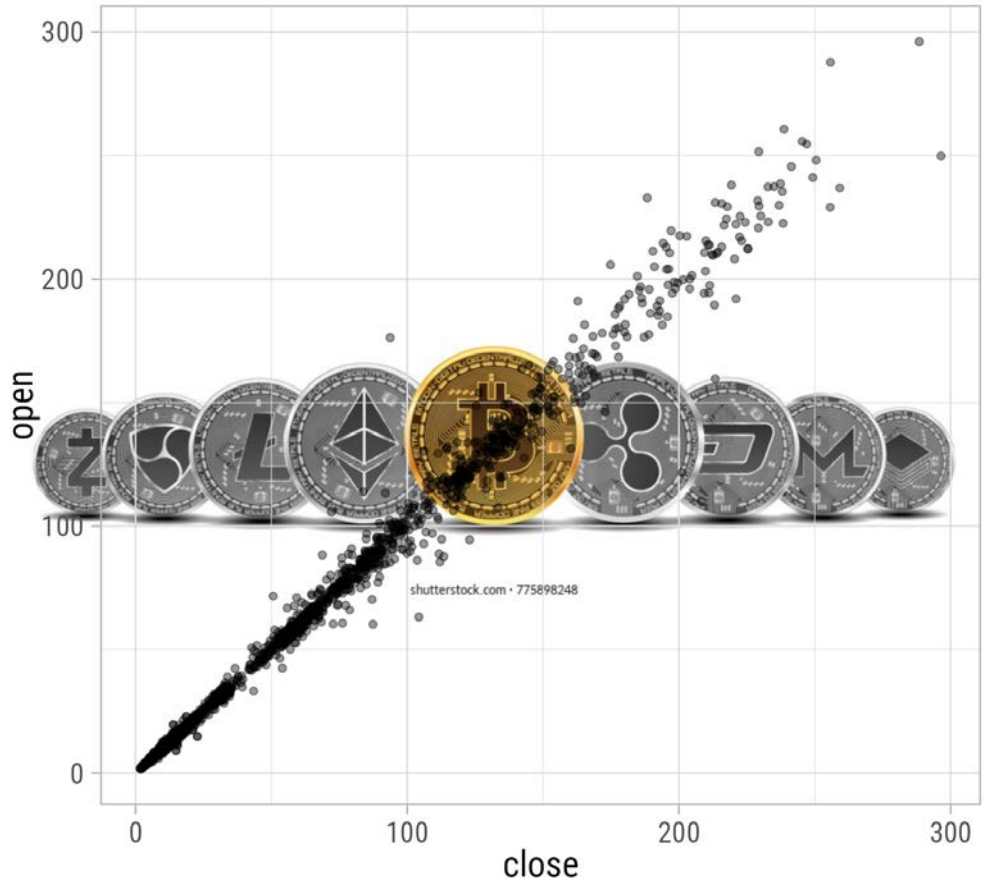
```
#install.packages("cowplot")
g <- ggplot(data, aes(close, open)) +
  geom_point(alpha = .4) +
  theme(plot.margin = margin(12, 12, 70, 12))

cowplot::ggdraw(g) +
  cowplot::draw_image(
    img,
    scale = 1
  )
```



Add Images with the {cowplot} Package

The cowplot package can not only add plots to other plots but also images:

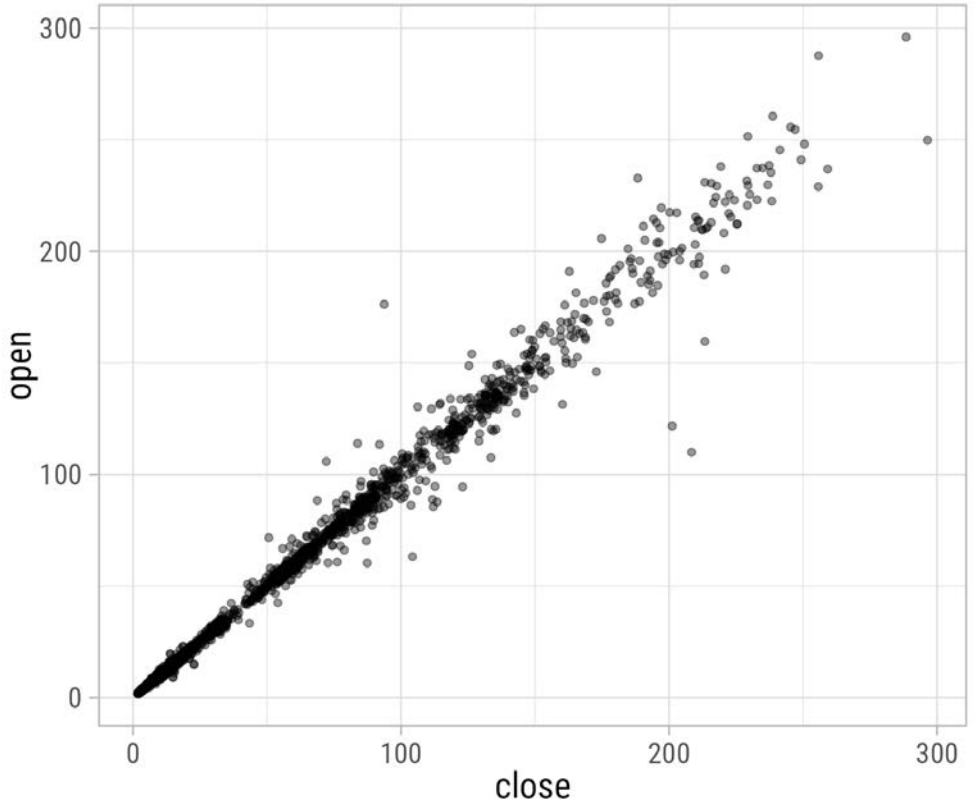
```
g <- ggplot(data, aes(close, open)) +
  geom_point(alpha = .4) +
  theme(panel.background = element_blank(),
        plot.background = element_blank())

cowplot::ggdraw() +
  cowplot::draw_image(
    img,
    scale = 1
  ) +
  cowplot::draw_plot(g)
```



Add Images with the {cowplot} Package

The cowplot package can also be used to add images:

```
g <- ggplot(data, aes(close, open)) +
  geom_point(alpha = .4) +
  theme(plot.margin = margin(12, 12, 45, 12))

url <- "https://upload.wikimedia.org/wikipedia/en/t
logo <- magick::image_read(url)

cowplot::ggdraw(g) +
  cowplot::draw_image(
    logo,
    scale = .2,
    x = 1,
    hjust = 1,
    halign = 1,
    valign = 0
  )
```



Pearson

Exercise 2:

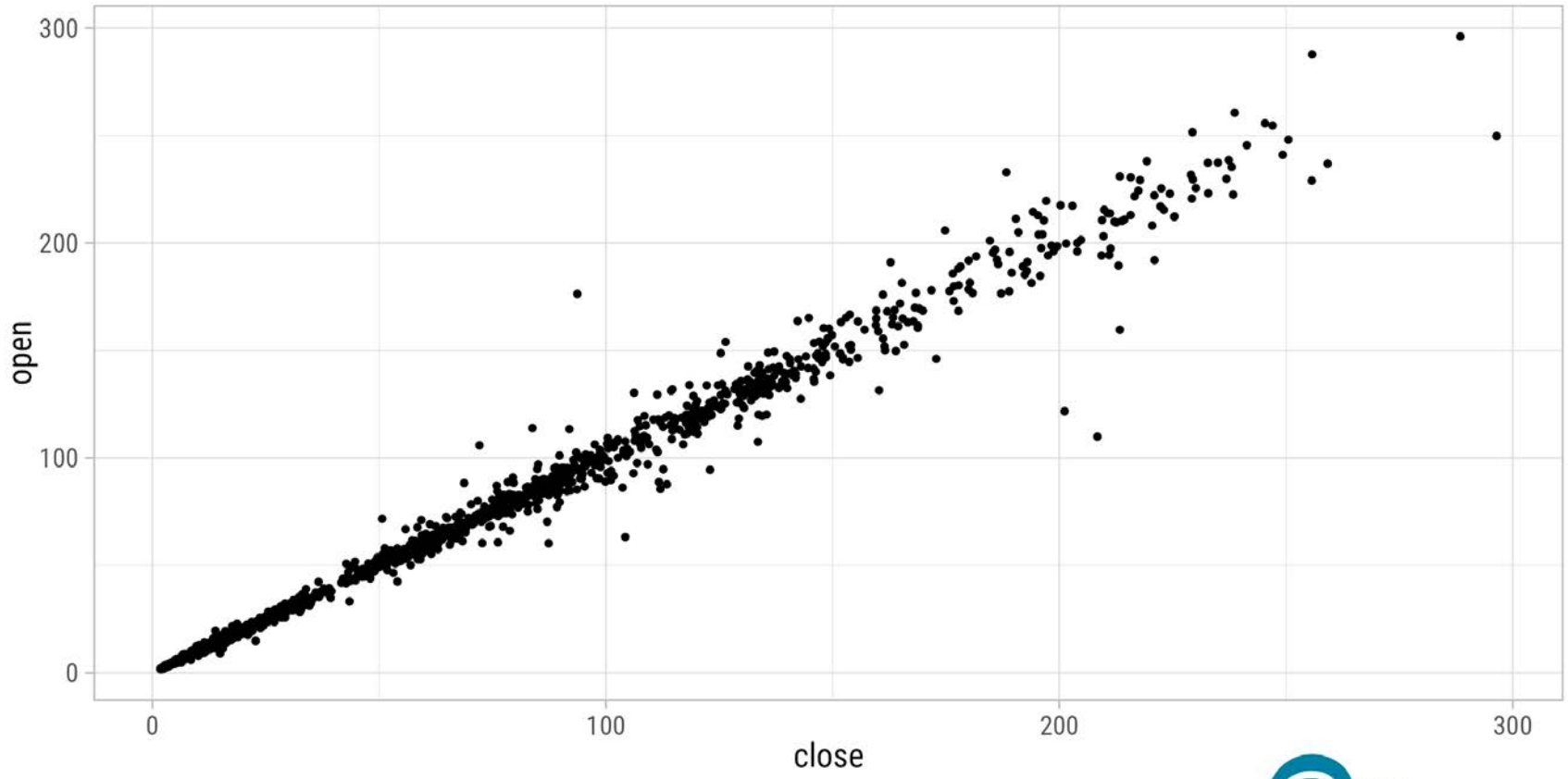
- Play around with the two different approaches and how they differ in placing and scaling the images.
- **Bonus:** Write a function that lets you add your company's logo to a ggplot object.

Exercise 2:

```
add_logo <- function(g) {
  url <- "https://upload.wikimedia.org/wikipedia/en/thumb/3/35/Pearson_logo.svg/1280px-Pearson_logo.svg.pn
  logo <- magick::image_read(url)

  g <- g +
    theme(plot.margin = margin(12, 12, 45, 12))

  plot <-
    cowplot::ggdraw(g) +
    cowplot::draw_image(
      logo,
      scale = .2,
      x = 1,
      hjust = 1,
      halign = 1,
      valign = 0
    )

  return(plot)
}
```

Exercise 2:

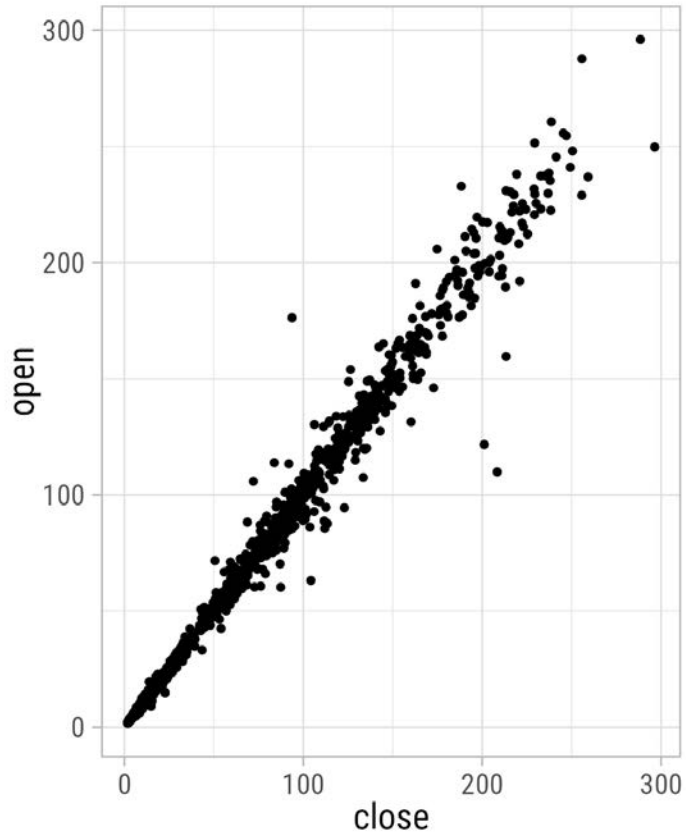
```
my_plot <- ggplot(data, aes(close, open)) + geom_point()
add_logo(my_plot)
```



Pearson

Exercise 2:

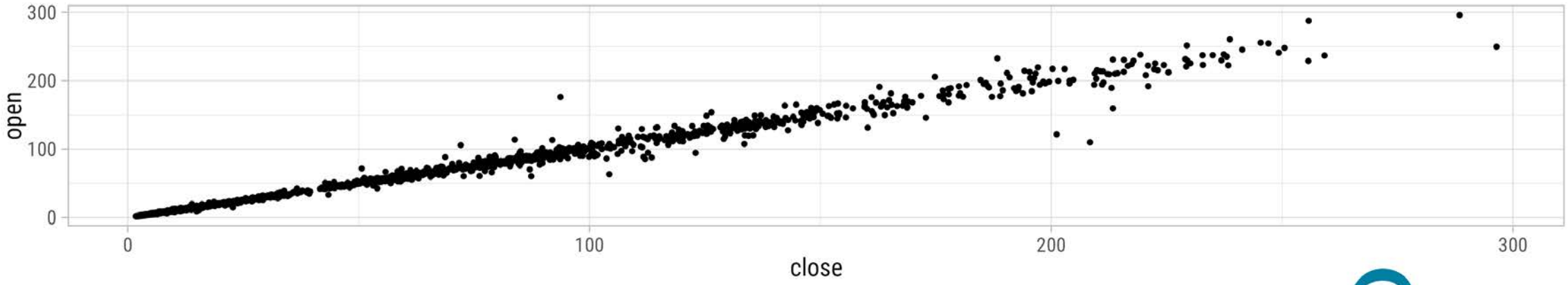
```
my_plot <- ggplot(data, aes(close, open)) + geom_point()
add_logo(my_plot)
```



Pearson

Exercise 2:

```
my_plot <- ggplot(data, aes(close, open)) + geom_point()
add_logo(my_plot)
```



Pearson

Resources

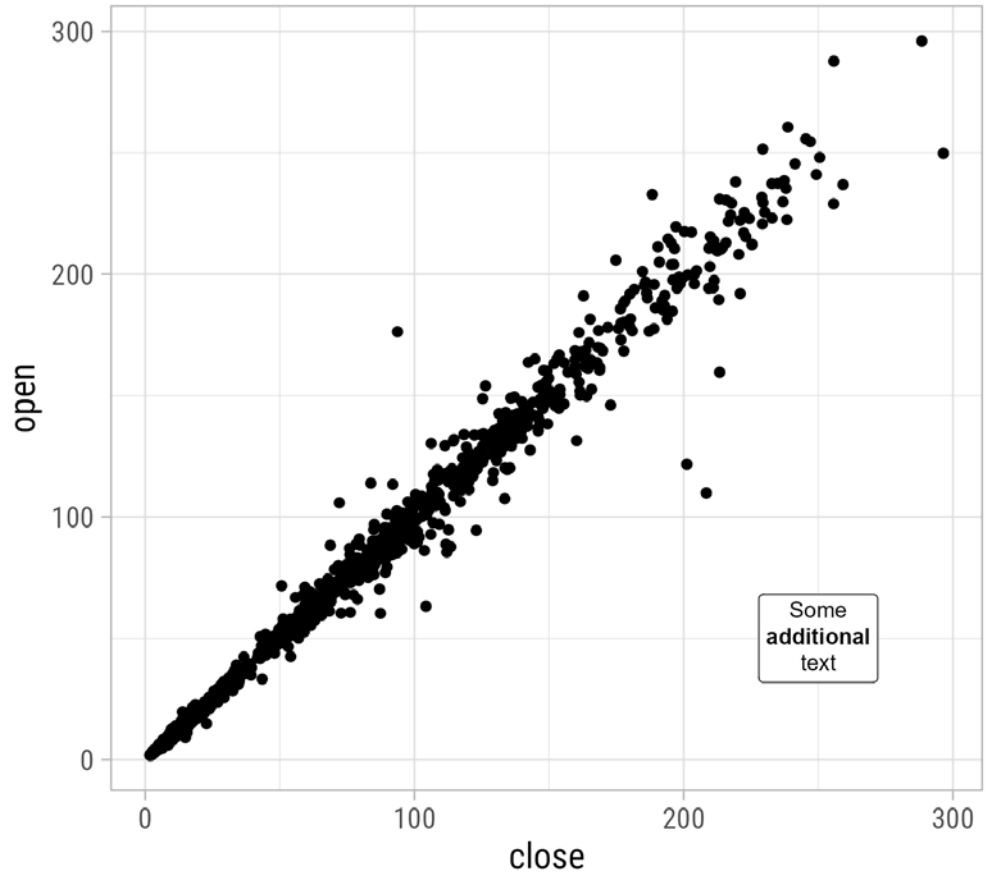
- Chapter 8 `Annotations` of the “ggplot2” book by Hadley Wickham et al.
- Chapter 7 `Annotations` of the “R Graphics Cookbook” book by Winston Chang
- `“Add a Logo to Your Plot”`, blog post by Thomas Mock
- `“How to Add a Logo to ggplot by Magick”`, blog post by Daniel Hadley
- `“A {ggplot2} Tutorial for Beautiful Plotting in R”`, my extensive "how to"-tutorial

APPENDIX

Annotations with {ggtext}

The {ggtext} package also comes with two geom's: geom_richtext() and geom_textbox():

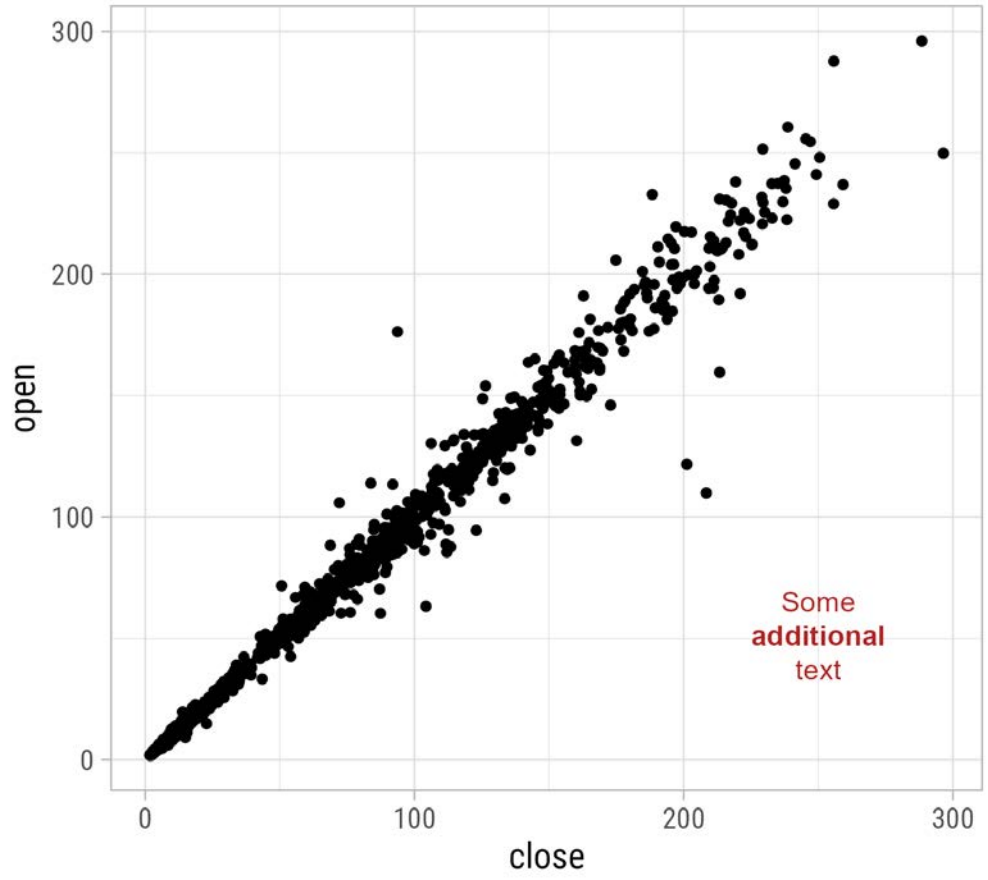
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  geom_richtext(
    aes(
      x = 250, y = 50,
      label = "Some<br>**additional**<br>text"
    ),
    stat = "unique"
  )
```



Annotations with {ggtext}

The {ggtext} package also comes with two geom's: geom_richtext() and geom_textbox():

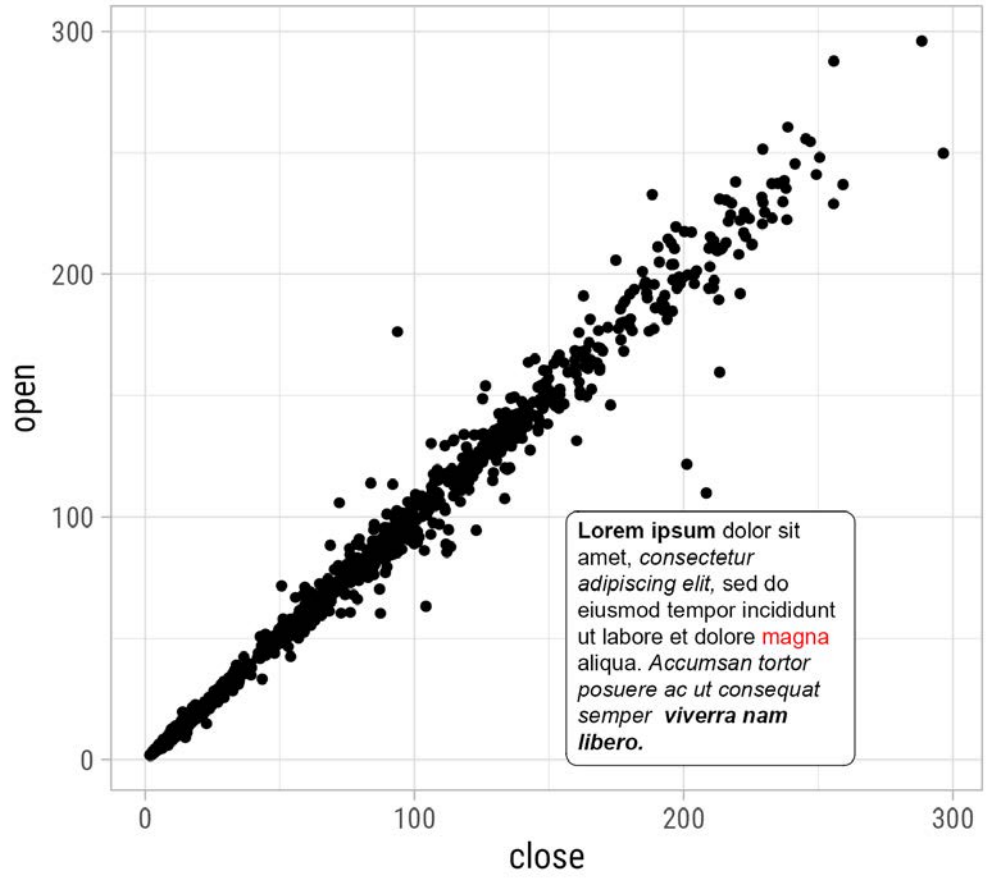
```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  geom_richtext(
    aes(
      x = 250, y = 50,
        label = "Some<br>**additional**<br>text"
    ),
    stat = "unique",
    color = "firebrick",
    size = 5,
    fill = NA,
    label.color = NA
  )
```



Annotations with {ggtext}

The {ggtext} package also comes with two geom's: geom_richtext() and geom_textbox():

```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  geom_textbox(
    aes(
      x = 210, y = 50,
      label =  "**Lorem ipsum** dolor sit amet, *co
    ),
   stat = "unique"
  )
```



Annotations with {ggtext}

The {ggtext} package also comes with two geom's: geom_richtext() and geom_textbox():

```
ggplot(data, aes(close, open)) +
  geom_point(size = 2) +
  geom_textbox(
    aes(
      x = 210, y = 50,
      label =  "**Lorem ipsum** dolor sit amet, *cor
    ),
    stat = "unique",
   width = unit(15, "lines")
  )
```