true

2021-01-23

Contents

5

1 → 7

1.1 (Kittler, 2019) 7

1.2 8

1.3 10

1.4 12

1.5 14

2 17

2.1 17

3 25

3.1 25

3.2 26

3.3 (circle) 27

3.4 (line) 28

3.5 29

4 Methods 31

5 Applications 33

5.1 Example one 33

5.2 Example two 33

6 Final Words 35

.

KCD 2020 .

Chapter 1

$\rightarrow$ 1

1.1 (Kittler, 2019)

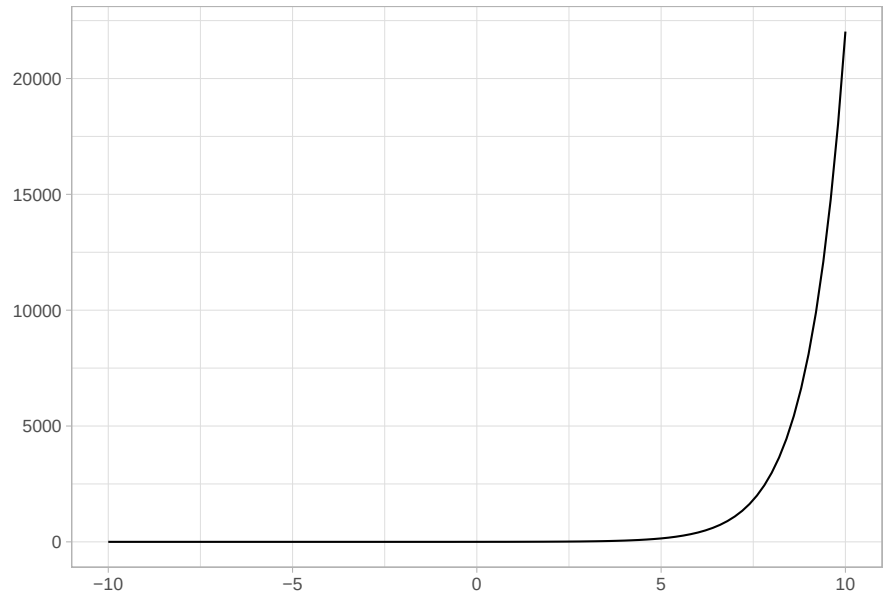
-10 10 .

$$y = \exp(x)$$

`geom_function()` `scale_x_continuous()` `-10 ~ +10`
.

```
library(tidyverse)

# exp_tbl <- expand_grid(x_range = -10:10)
#
# exp_tbl %>%
#   ggplot(aes(x_range)) +
#     geom_density() +
#     geom_function(fun = function(x) exp(x))

ggplot() +
    geom_density() +
    scale_x_continuous(limits = c(-10,10)) +
    geom_function(fun = function(x) exp(x) ) +
    theme_light() +
    labs(x="", y="")
```

[1] Draw a function as a continuous curve



1.2

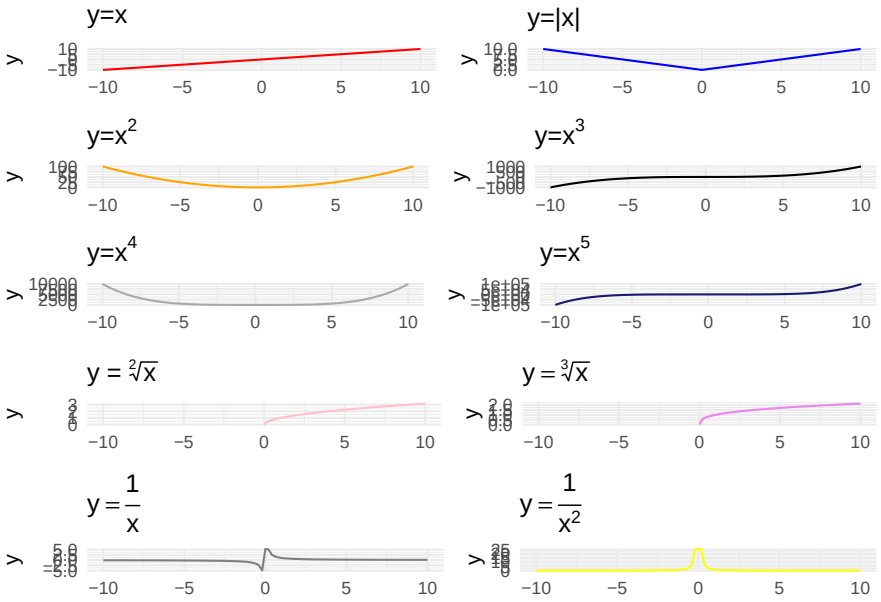
10 ggplot . geom_function . , latex2exp $LaTeX$ ggplot . 10 patchwork .

- $y = x$
- $y = |x|$
- $y = x^2$
- $y = x^3$
- $y = x^4$
- $y = x^5$
- $y = \sqrt{x}$
- $y = \sqrt[3]{x}$
- $y = \frac{1}{x}$
- $y = \frac{1}{x^2}$

```
library(patchwork)
library(latex2exp)
theme_set(theme_minimal())

simple_eq_01 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
```

```
  geom_function(fun = function(x) x, color="red" ) +
  labs(title = TeX(c("y=x")))

simple_eq_02 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) abs(x), color="blue" ) +
  labs(title = TeX(c("y=|x|")))

simple_eq_03 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) x^2,    color="orange" ) +
  labs(title = TeX(c("y=x^2")))

simple_eq_04 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) x^3,    color="black" ) +
  labs(title = TeX(c("y=x^3")))

simple_eq_05 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) x^4,    color="darkgray" ) +
  labs(title = TeX(c("y=x^4")))

simple_eq_06 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) x^5,    color="midnightblue" ) +
  labs(title = TeX(c("y=x^5")))

simple_eq_07 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) x^(1/2),    color="pink" ) +
  labs(title = TeX(c("y = \\sqrt[2]_{x}")))

simple_eq_08 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) x^(1/3),    color="violet" ) +
  labs(title = TeX(c("$y = \\sqrt[3]_{x}$")))

simple_eq_09 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) 1 / x,    color="gray50" ) +
  labs(title = TeX(c("$y = \\frac{1}{x}$")))

simple_eq_10 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
```

```
  geom_function(fun = function(x) 1 / (x)^2,     color="yellow" ) +
  labs(title = TeX(c("$y = \\frac{1}{x^2}$")))

( simple_eq_01 + simple_eq_02 ) /
( simple_eq_03 + simple_eq_04 ) /
( simple_eq_05 + simple_eq_06 ) /
( simple_eq_07 + simple_eq_08 ) /
( simple_eq_09 + simple_eq_10 )
```



1.3

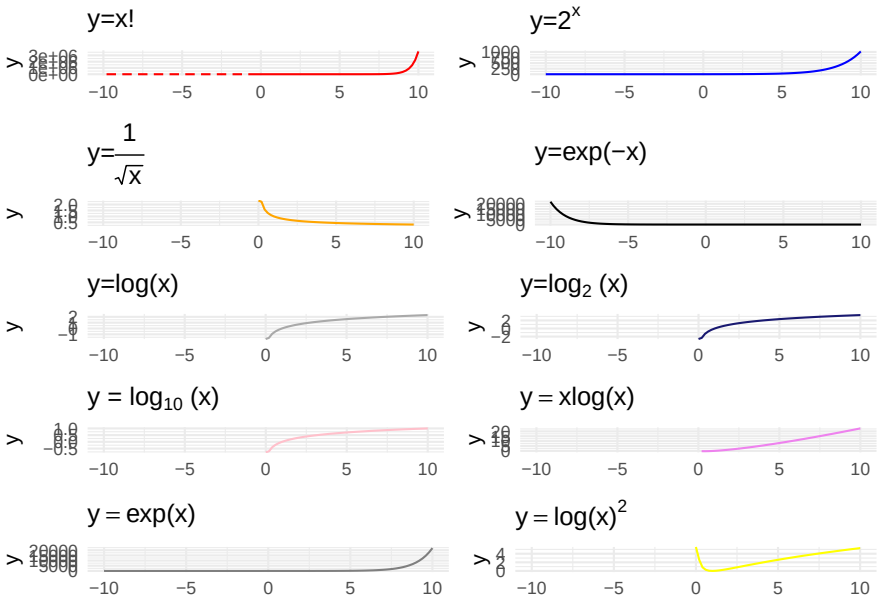
.

- $y = x!$
- $y = 2^x$
- $y = \frac{1}{\sqrt{x}}$
- $y = exp(-x)$
- $y = log(x)$
- $y = log_2(x)$
- $y = log_{10}(x)$
- $y = x \times log(x)$
- $y = exp(x)$

- $y = log(x)^2$

```
simple_eq_11 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) factorial(x), color="red" ) +
  labs(title = TeX(c("y=x!")))

simple_eq_12 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) 2^x, color="blue" ) +
  labs(title = TeX(c("y=2^x")))

simple_eq_13 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) 1 / sqrt(x),    color="orange" ) +
  labs(title = TeX(c("y=\\frac{1}{\\sqrt{x}}")))

simple_eq_14 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) exp(-x),    color="black" ) +
  labs(title = TeX(c("y=exp(-x)")))

simple_eq_15 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) log(x),    color="darkgray" ) +
  labs(title = TeX(c("y=log(x)")))

simple_eq_16 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) log2(x),    color="midnightblue" ) +
  labs(title = TeX(c("y=log_2 (x)")))

simple_eq_17 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) log10(x),    color="pink" ) +
  labs(title = TeX(c("y = log_{10} (x)")))

simple_eq_18 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) x * log(x),    color="violet" ) +
  labs(title = TeX(c("$y = x log(x)")))

simple_eq_19 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) exp(x),    color="gray50" ) +
```

```
  labs(title = TeX(c("$y = exp(x)$")))

simple_eq_10 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) log(x)^2,    color="yellow" ) +
  labs(title = TeX(c("$y = log(x)^2")))

( simple_eq_11 + simple_eq_12 ) /
( simple_eq_13 + simple_eq_14 ) /
( simple_eq_15 + simple_eq_16 ) /
( simple_eq_17 + simple_eq_18 ) /
( simple_eq_19 + simple_eq_10 )
```



1.4

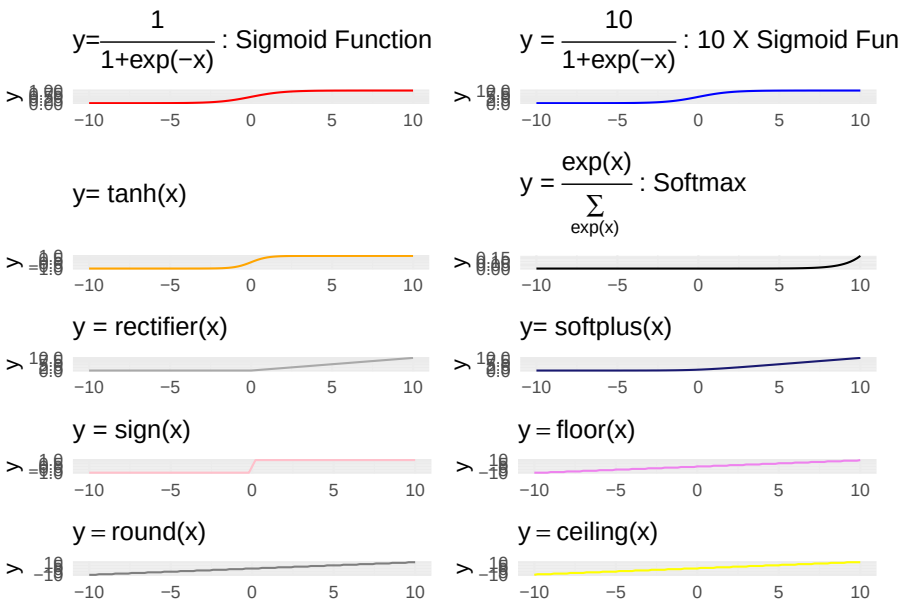
(activation) .

1. $y = \frac{1}{1+exp(-x)} : SigmoidFunction$
2. $y = \frac{10}{1+exp(-x)} : 10XSigmoidFunction$
3. $y = tanh(x)$
4. $y = \frac{exp(x)}{\sum exp(x)} : Softmax$

5. $y = rectifier(x)$
6. $y = softplus(x)$
7. $y = sign(x)$
8. $y = floor(x)$
9. $y = round(x)$
10. $y = ceiling(x)$

```
simple_eq_21 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) 1 / (1 + exp(-x)), color="red" ) +
  labs(title = TeX(c("y=\\frac{1}{1+exp(-x)} : Sigmoid Function")))

simple_eq_22 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) 10 / (1 + exp(-x)), color="blue" ) +
  labs(title = TeX(c("y = \\frac{10}{1+exp(-x)} : 10 X Sigmoid Function")))

simple_eq_23 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) tanh(x),    color="orange" ) +
  labs(title = TeX(c("y= tanh(x)")))

simple_eq_24 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) exp(x) / sum(exp(x)),    color="black" ) +
  labs(title = TeX(c("y = \\frac{exp(x)}{\\sum{exp(x)}} : Softmax")))

simple_eq_25 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) pmax(0, x),    color="darkgray" ) +
  labs(title = TeX(c("y = rectifier(x)")))

simple_eq_26 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) log(1 + exp(x)),    color="midnightblue" ) +
  labs(title = TeX(c("y= softplus(x)")))

simple_eq_27 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) sign(x),    color="pink" ) +
  labs(title = TeX(c("y = sign(x)")))

simple_eq_28 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) floor(x),    color="violet" ) +
```

```
  labs(title = TeX(c("$y = floor(x)")))

simple_eq_29 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) round(x),    color="gray50" ) +
  labs(title = TeX(c("$y = round(x)$")))

simple_eq_30 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) ceiling(x),    color="yellow" ) +
  labs(title = TeX(c("$y = ceiling(x)")))

( simple_eq_21 + simple_eq_22 ) /
( simple_eq_23 + simple_eq_24 ) /
( simple_eq_25 + simple_eq_26 ) /
( simple_eq_27 + simple_eq_28 ) /
( simple_eq_29 + simple_eq_30 )
```



1.5

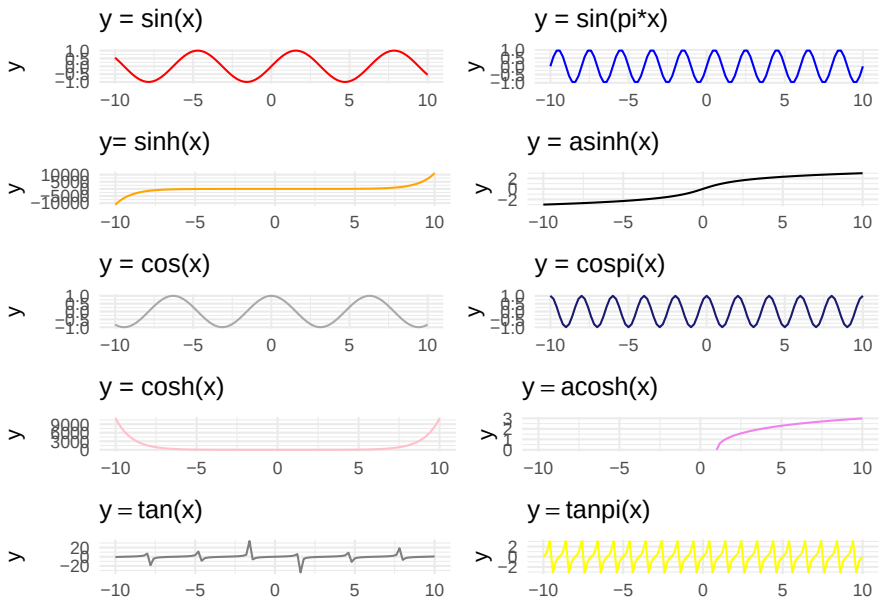
(trigonometric) .

1. $y = sin(x)$

2. $y = sin(pi * x)$
3. $y = sinh(x)$
4. $y = asinh(x)$
5. $y = cos(x)$
6. $y = cospi(x)$
7. $y = cosh(x)$
8. $y = acosh(x)$
9. $y = tan(x)$
10. $y = tanpi(x)$

```
simple_eq_31 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) sin(x), color="red" ) +
  labs(title = TeX(c("y = sin(x)")))

simple_eq_32 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) sinpi(x), color="blue" ) +
  labs(title = TeX(c("y = sin(pi*x)")))

simple_eq_33 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) sinh(x),    color="orange" ) +
  labs(title = TeX(c("y= sinh(x)")))

simple_eq_34 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) asinh(x),    color="black" ) +
  labs(title = TeX(c("y = asinh(x)")))

simple_eq_35 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) cos(x),    color="darkgray" ) +
  labs(title = TeX(c("y = cos(x)")))

simple_eq_36 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) cospi(x),    color="midnightblue" ) +
  labs(title = TeX(c("y = cospi(x)")))

simple_eq_37 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) cosh(x),    color="pink" ) +
  labs(title = TeX(c("y = cosh(x)")))
```

```
simple_eq_38 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) acosh(x),    color="violet" ) +
  labs(title = TeX(c("$y = acosh(x)")))

simple_eq_39 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) tan(x),    color="gray50" ) +
  labs(title = TeX(c("$y = tan(x)$")))

simple_eq_40 <- ggplot() +
  scale_x_continuous(limits = c(-10,10)) +
  geom_function(fun = function(x) tanpi(x),    color="yellow" ) +
  labs(title = TeX(c("$y = tanpi(x)")))

( simple_eq_31 + simple_eq_32 ) /
( simple_eq_33 + simple_eq_34 ) /
( simple_eq_35 + simple_eq_36 ) /
( simple_eq_37 + simple_eq_38 ) /
( simple_eq_39 + simple_eq_40 )
```



Chapter 2

- (Prime Number): 1 , 1 ( )
- (Composite Number):

  − 1
- (Prime Factor):
- (Coprime, Relatively Prime): 1

2.1

, N . N p, q .
N p q . N .(,
2020)

2.1.1

Prime number function in R . N $2 \sim N-1$
. %% . , 10 3 1 . 10
%% 3 1 .

```
is_prime <- function(num) {
   if (num == 2) {
      TRUE
   } else if (any(num %% 2:(num-1) == 0)) {
      FALSE
   } else {
```

```
      TRUE
   }
}

is_prime(3)
```

```
[1] TRUE
```

$1 \sim 100$.

```
library(tidyverse)

natural_number <- 1:100

prime_number_decision <- map_lgl(natural_number, is_prime)

tibble(     = natural_number,
           = prime_number_decision) %>%
  reactable::reactable()
```

자연수	소수판정
1	false
2	true
3	true
4	false
5	true
6	false
7	true
8	false
9	false
10	false

1–10 of 100 rows Previous 1 2 3 4 5 ... 10 Next

N N . reactable .

```
library(reactable)
calculate_primes <- function(number) {

  natural_number <- 1:number

  prime_number_decision <- map_lgl(natural_number, is_prime)

  tibble(     = natural_number,
              = prime_number_decision) %>%
    reactable::reactable(columns = list(
            = colDef(
          cell = function(value) {
            if (value == TRUE ) paste0("", value) else value
          },
```

```
          style = function(value) {
            color <- if (value == TRUE) {
              "#008000"
            } else{
              "#e00000"
            }
            list(fontWeight = 600, color = color)
          }
        )
      ))
}

calculate_primes(10)
```

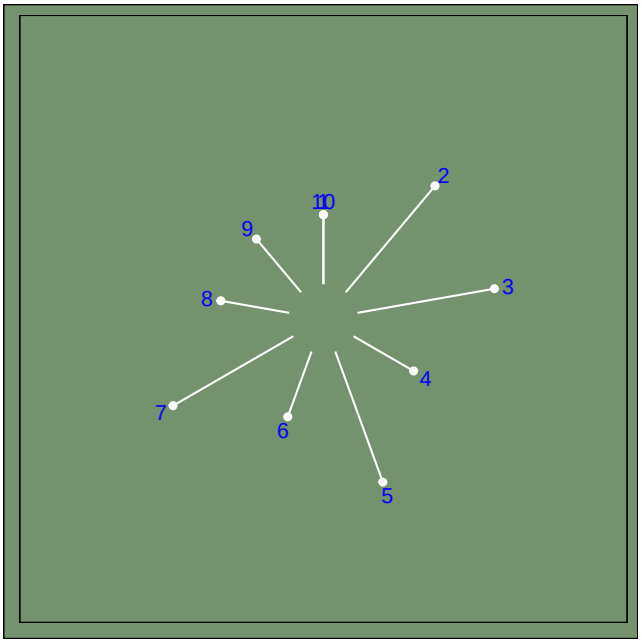
자연수	소수판정
1	FALSE
2	TRUE
3	TRUE
4	FALSE
5	TRUE
6	FALSE
7	TRUE
8	FALSE
9	FALSE
10	FALSE

2.1.2

.

```
##
generate_primes <- function(number) {

  natural_number <- 1:number
  prime_number_decision <- map_lgl(natural_number, is_prime)

  decide_prime_tbl <- tibble(     = natural_number,
                                  = prime_number_decision %>% as.integer + 1)

  return(decide_prime_tbl)
}

prime_tbl <- generate_primes(10)

##
lines <- tibble(number = prime_tbl %>% pull(`  `),
                x      = seq(1, 10, by = 1),
                xend   = x,
                y      = rep(0, 10),
                yend   = prime_tbl %>% pull(`  `))

dots <- lines %>%
  select(x, yend)

##    ----
lines %>% ggplot() +
  geom_segment(aes(x = x, xend = xend,
                   y = y, yend = yend),
               color = "white") +
  geom_text(aes(x = x, y = yend + 0.2, label = number),
            color = "blue") +
  geom_point(data = dots,
             aes(x = x, y = yend),
             color = "white") +
  coord_polar() +
  ylim(-0.5, 3) +
  coord_polar() +
  theme(
    plot.background = element_rect(
      fill = "#75926f"),
    panel.background = element_rect(
      fill = "#75926f"),
```

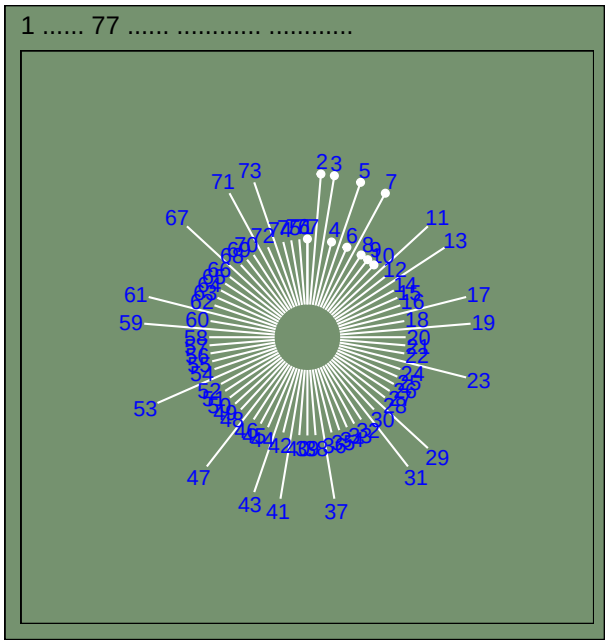
```
    panel.grid = element_blank(),
    plot.caption = element_text(
      family = "Open Sans",
      size = 6,
      color = "white"),
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank()
  )
```



. , .

```
generate_primes <- function(number) {

  natural_number <- 1:number
  prime_number_decision <- map_lgl(natural_number, is_prime)

  decide_prime_tbl <- tibble(     = natural_number,
                                  = prime_number_decision %>% as.integer + 1)
  return(decide_prime_tbl)
}

visualize_prime <- function(number) {
```

```
  #

  prime_tbl <- generate_primes(number)

  graph_tbl <- tibble( natural_number = prime_tbl %>% pull(`  `),
                       x              = seq(1, number, by = 1),
                       xend           = x,
                       y              = rep(0, number),
                       yend           = prime_tbl %>% pull(`  `)
  )

  dots <- lines %>%
    select(x, yend)

  ##    ----
  graph_tbl %>% ggplot() +
    geom_segment(aes(x = x, xend = xend,
                     y = y, yend = yend),
                 color = "white") +
    geom_text(aes(x = x, y = yend + 0.2, label = natural_number),
              color = "blue") +
    geom_point(data = dots,
               aes(x = x, y = yend),
               color = "white") +
    coord_polar() +
    ylim(-0.5, 3) +
    coord_polar() +
    theme(
      plot.background = element_rect(
        fill = "#75926f"),
      panel.background = element_rect(
        fill = "#75926f"),
      panel.grid = element_blank(),
      plot.caption = element_text(
        family = "Open Sans",
        size = 6,
        color = "white"),
      axis.title = element_blank(),
      axis.text = element_blank(),
      axis.ticks = element_blank()
    ) +
    labs(title = glue::glue("1   {number}        "))
}

visualize_prime(77)
```

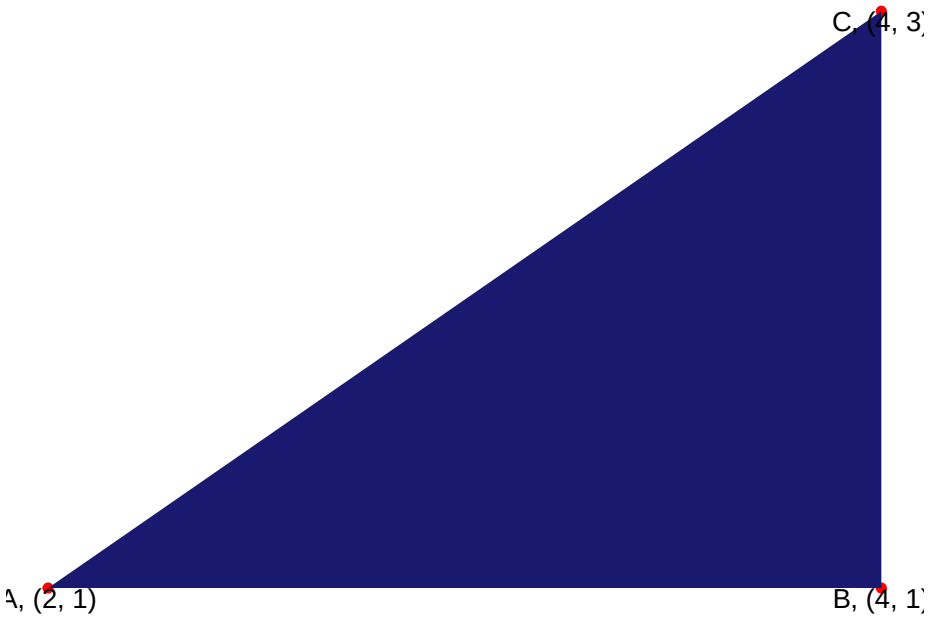

1 ..... 77 ....

Chapter 3

3.1

. ggplot . geom_polygon()
point geom_text() .

```
library(tidyverse)
triangle_tbl <- tibble(coord_x   = c(2,4,4),
                       coord_y   = c(1,1,3),
                       group     = c(1,1,1),
                       point     = c("A", "B", "C")) %>%
  mutate(point = glue::glue("{point}, ({coord_x}, {coord_y})"))

triangle_tbl %>%
  ggplot(aes(x     = coord_x,
             y     = coord_y,
             group = group,
             label = point)) +
    geom_point(size = 2, color = "red") +
    geom_polygon(fill ="midnightblue") +
    geom_text(vjust = 1, hjust= 0.5, size = 5) +
    theme_void()
```



3.2

ggplot 4 geom_rect() .

```
rect_tbl <- tibble(bottom_left  = 1,
                   bottom_right = 3,
                   top_left     = 1,
                   top_right    = 3)


rect_point_tbl <- rect_tbl %>%
  select(contains("left")) %>%
  rename(x=bottom_left, y=top_left) %>%
  bind_rows(
    rect_tbl %>%
      select(contains("right")) %>%
      rename(x=bottom_right, y=top_right)
  ) %>%
  mutate(point = c("A", "D")) %>%
  mutate(point = glue::glue("{point}:({x},{y})"))

ggplot() +
    geom_rect(data = rect_tbl, aes(xmin=bottom_left, xmax=bottom_right, ymin=top_left,
```

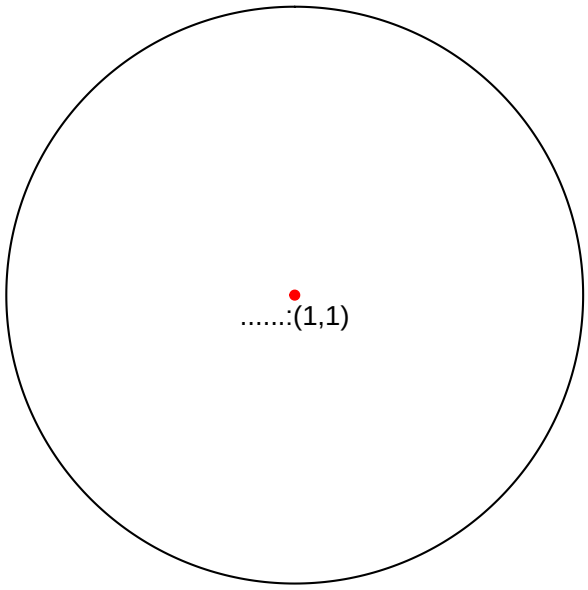
```
    geom_point(data = rect_point_tbl, mapping=aes(x=x, y=y), size = 2, color = "red") +
    geom_polygon(fill ="midnightblue") +
    geom_text(data = rect_point_tbl, mapping=aes(x=x, y=y, label = point), vjust = 1, hjust= 0.5,
    theme_void() +
    theme(legend.position = "none")
```

3.3 (circle)

ggforce geom_circle() (x, y) r .

```
library(ggforce)

circle_tbl <- tibble(x_pos = 1,
                     y_pos = 1) %>%
  mutate(center = glue::glue(" :({x_pos},{y_pos})"))

circle_tbl %>%
  ggplot(aes(x_pos, y_pos)) +
  geom_circle(aes(x0 = 1, y0 = 1, r = 1), inherit.aes = FALSE) +
  geom_point(size = 2, color = "red") +
  geom_text(aes(label = center), vjust = 1.5, hjust= 0.5, size = 5) +
  coord_fixed(ratio = 1) +
```

```
  theme_void() +
  theme(legend.position = "none")
```

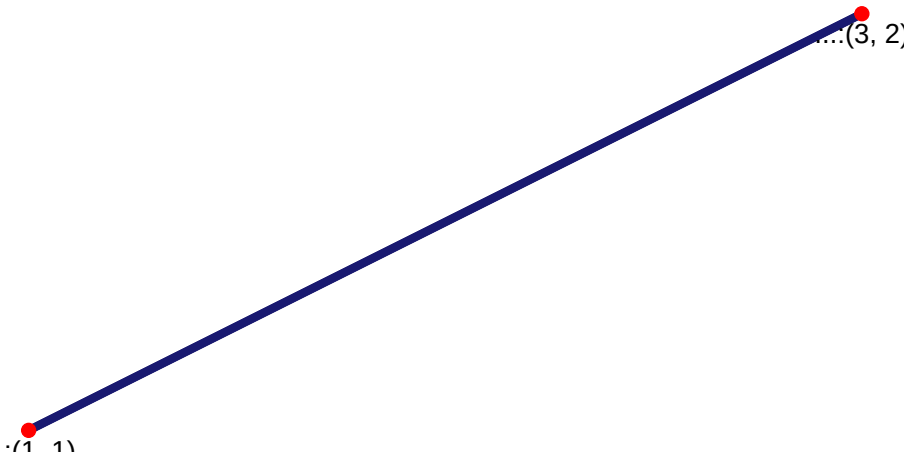


3.4 (line)

geom_segment() ggplot .

```
line_tbl <- tibble(x = c(1, 3),
                   y = c(1, 2)) %>%
  mutate(point = glue::glue(" :({x}, {y})"))

line_tbl %>%
  ggplot(aes(x, y, label = point)) +
    geom_segment(aes(x = line_tbl %>% select(x) %>% slice(1) %>% pull,
                     y = line_tbl %>% select(y) %>% slice(1) %>% pull,
                     xend = line_tbl %>% select(x) %>% slice(2) %>% pull,
                     yend = line_tbl %>% select(y) %>% slice(2) %>% pull),
                 size = 2,
                 linetype = 1,
                 color = "midnightblue") +
    geom_point(size = 3, color = "red")   +
    geom_text(vjust = 1.5, hjust= 0.5, size = 5) +
```

```
    coord_fixed(ratio = 1) +
    theme_void() +
    theme(legend.position = "none")
```



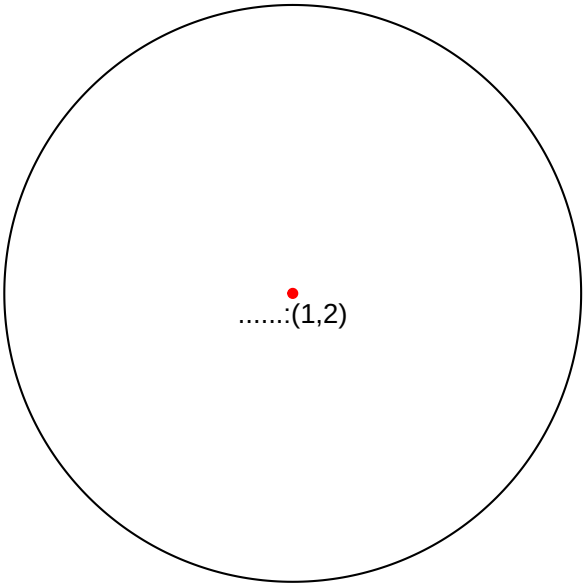
3.5

.

x, y r .

```
draw_circle <- function(x_pos, y_pos, radius) {

  circle_tbl <- tibble(x_pos = x_pos,
                       y_pos = y_pos) %>%
  mutate(center = glue::glue(" :({x_pos},{y_pos})"),
         radius = radius)

  circle_tbl %>%
    ggplot(aes(x_pos, y_pos)) +
    geom_circle(aes(x0 = x_pos, y0 = y_pos, r = radius), inherit.aes = FALSE) +
    geom_point(size = 2, color = "red") +
    geom_text(aes(label = center), vjust = 1.5, hjust= 0.5, size = 5) +
```

```
    coord_fixed(ratio = 1) +
    theme_void() +
    theme(legend.position = "none")
}

draw_circle(1,2,2)
```



, , .

Chapter 4

Methods

We describe our methods in this chapter.

Chapter 5

Applications

Some *significant* applications are demonstrated in this chapter.

5.1 Example one

5.2 Example two

Chapter 6

Final Words

We have finished a nice book.

Bibliography

Kittler, J. (2019). Overview of 40 mathematical functions in r.

(2020). : .