DATA VISUALIZATION

Gerko Vink
g.vink@uu.nl
Methodology & Statistics @ Utrecht University

12 Jun 2025

DISCLAIMER

I owe a debt of gratitude to many people as the thoughts and code in these slides are the process of years-long development cycles and discussions with my team, friends, colleagues and peers. When someone has contributed to the content of the slides, I have credited their authorship.

Images are either directly linked, or generated with StableDiffusion or DALL-E. That said, there is no information in this presentation that exceeds legal use of copyright materials in academic settings, or that should not be part of the public domain.

> ⚠ **Warning**
>
> You **may use** any and all content in this presentation - including my name - and submit it as input to generative AI tools, with the following **exception**:
>
> - You must ensure that the content is not used for further training of the model

SLIDE MATERIALS AND SOURCE CODE

Materials

- lecture slides on Moodle
- course page: www.gerkovink.com/sur
- source: github.com/gerkovink/sur

RECAP

Gisteren hebben we deze onderwerpen behandeld:

- Ontbrekende waarden identificeren
- Synthetische imputaties maken

SOME ADDITION

I've showed you how to draw a *bootstrap* sample from a dataset in the following way:

```
which_rows <- sample(1:nrow(mice::boys), size = 100, replace = TRUE)
mice::boys[which_rows, ]
```

But, there is a much easier way:

```
mice::boys %>%
  slice_sample(n = 100, replace = TRUE) %>%
  glimpse()
```

```
Rows: 100
Columns: 9
$ age <dbl> 1.790, 1.086, 13.656, 13.300, 0.188, 17.911, 12.501, 2.502, 0.101,…
$ hgt <dbl> 90.2, 74.5, 146.9, 165.5, 58.5, 181.2, 170.5, 91.1, 55.8, 59.5, 17…
$ wgt <dbl> 14.50, 8.92, 34.70, 41.90, 6.03, 86.80, 53.40, 15.80, 5.06, 5.13, …
$ bmi <dbl> 17.82, 16.07, 16.07, 15.29, 17.61, 26.43, 18.36, 19.03, 16.25, 14.…
$ hc  <dbl> 51.2, 45.1, 55.1, 55.6, 41.5, 58.3, 56.4, 49.7, 38.5, 38.0, 57.0, …
$ gen <ord> NA, NA, G2, G2, NA, G5, G2, NA, NA, NA, G5, G2, NA, NA, NA, G4, NA…
$ phb <ord> NA, NA, P3, P2, NA, P5, P1, NA, NA, NA, P4, P1, NA, NA, NA, P5, NA…
$ tv  <int> NA, NA, NA, 5, NA, 15, 6, NA, NA, NA, 20, 2, NA, NA, NA, 25, NA, N…
$ reg <fct> east, west, city, east, east, north, south, south, east, west, sou…
```


TODAY

Vandaag behandelen we de volgende onderwerpen:

- Basisplots: histogrammen, scatterplots en boxplots
- Geavanceerde plots met ggplot2
- Aanpassen van grafieken voor publicatie
- Exporteren van grafieken en resultaten

WE USE THE FOLLOWING PACKAGES

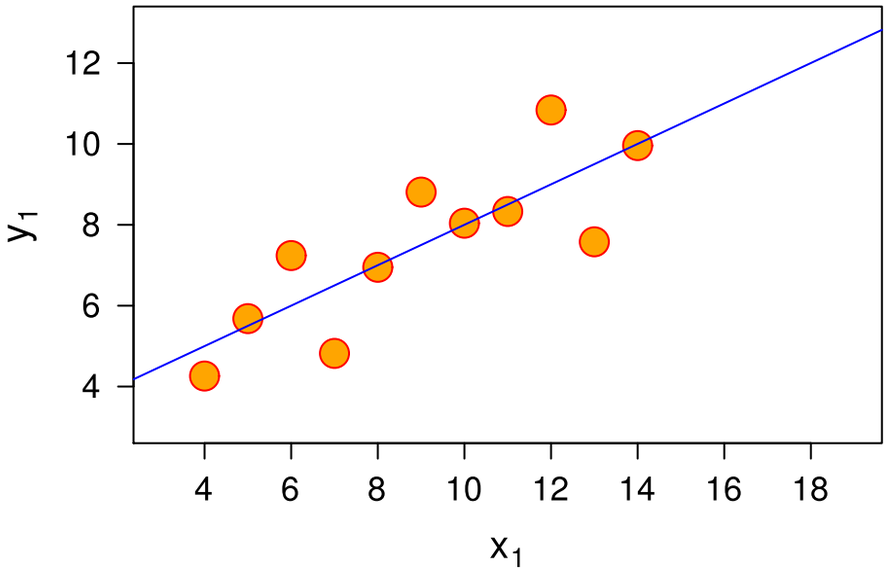
```
library(mice)     # Boys dataset
library(dplyr)    # Data manipulation
library(magrittr) # Pipes
library(ggplot2)  # Plotting suite
```

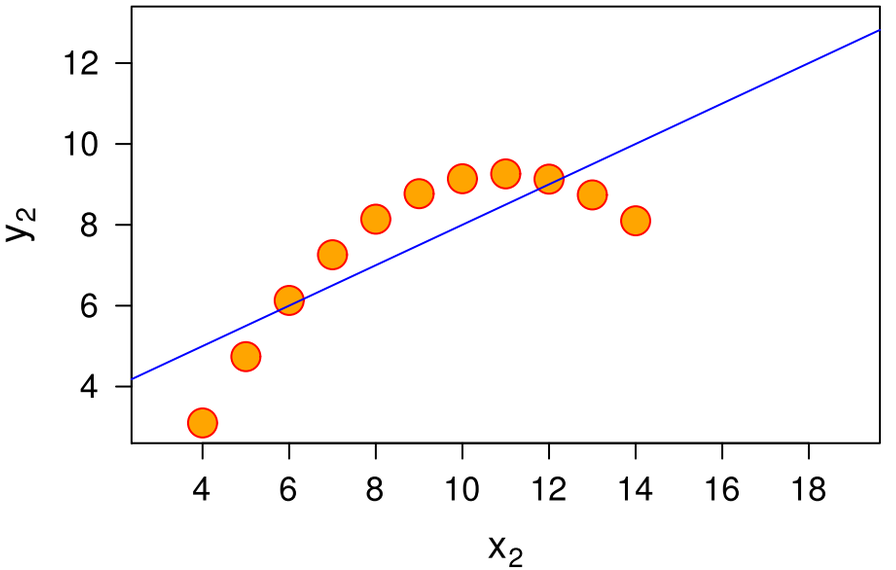
WHY VISUALISE?

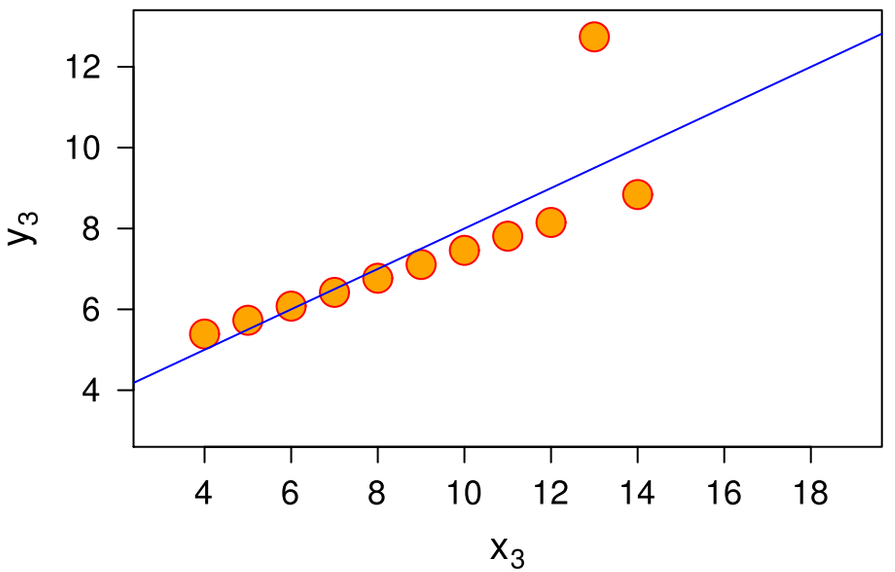
- We can process a lot of information quickly with our eyes
- Plots give us information about
  - Distribution / shape
  - Irregularities
  - Assumptions
  - Intuitions
- Summary statistics, correlations, parameters, model tests, *p*-values do not tell the whole story

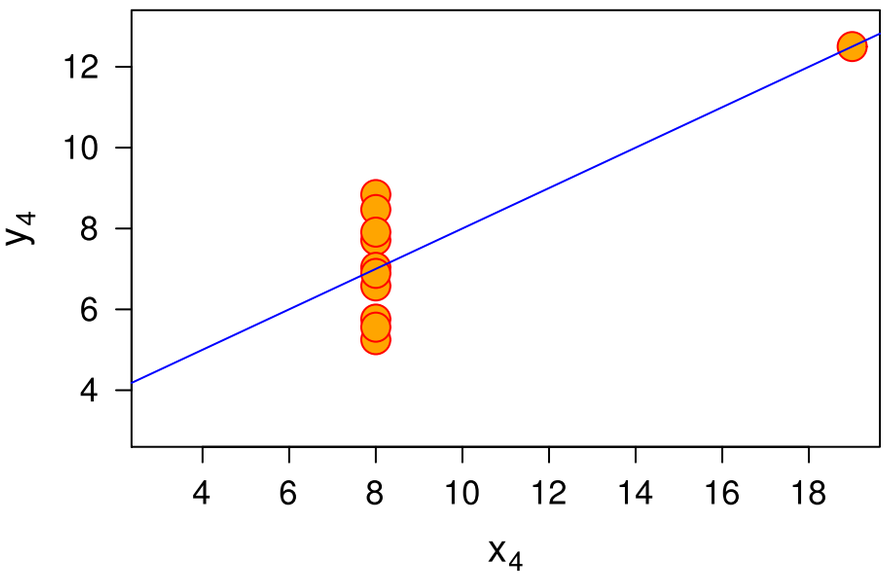
ALWAYS PLOT YOUR DATA!

WHY VISUALISE?



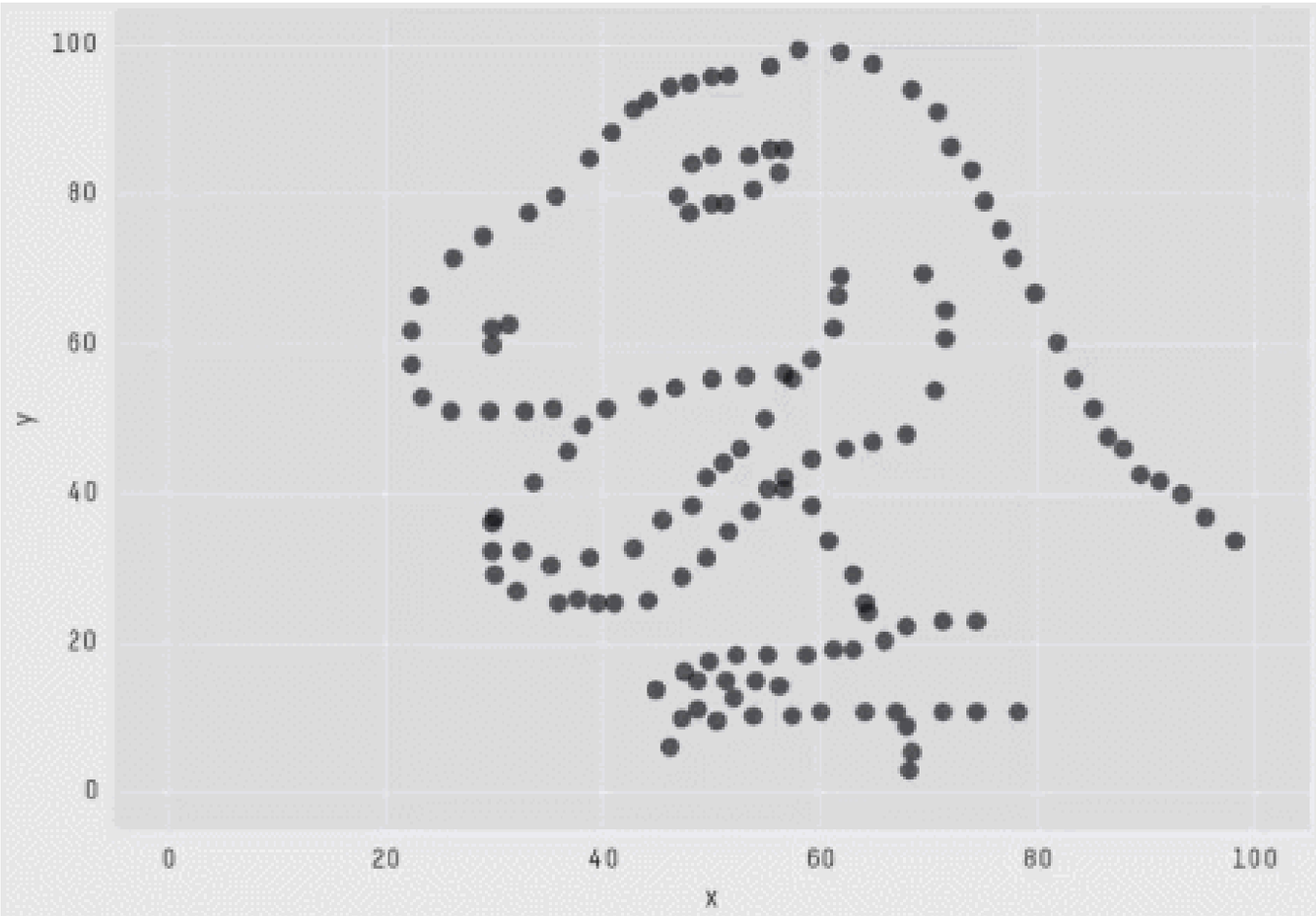






Source: Anscombe, F. J. (1973). "Graphs in Statistical Analysis". American Statistician. 27 (1): 17–21.

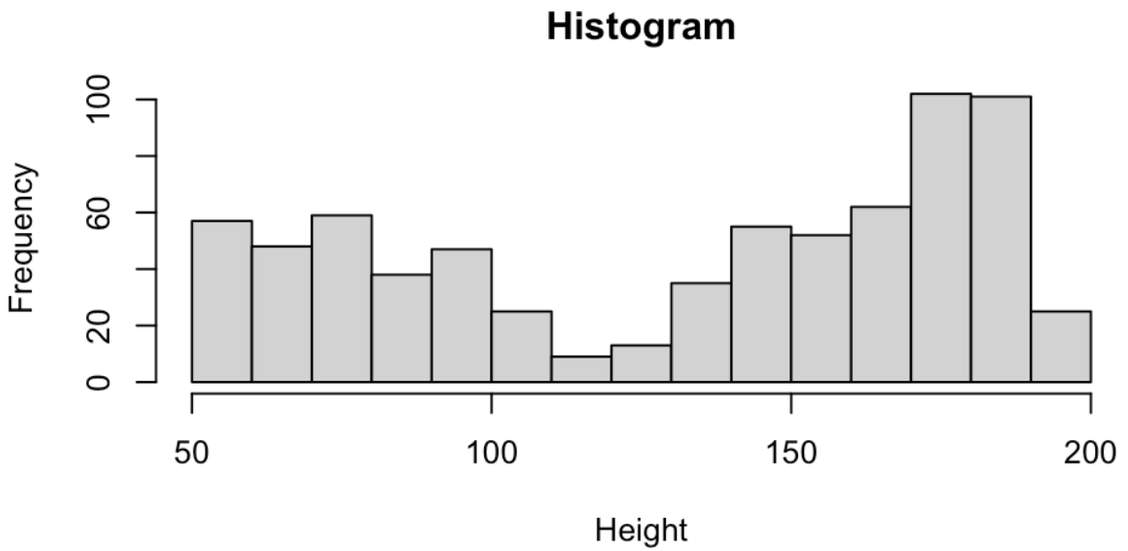
WHY VISUALISE?

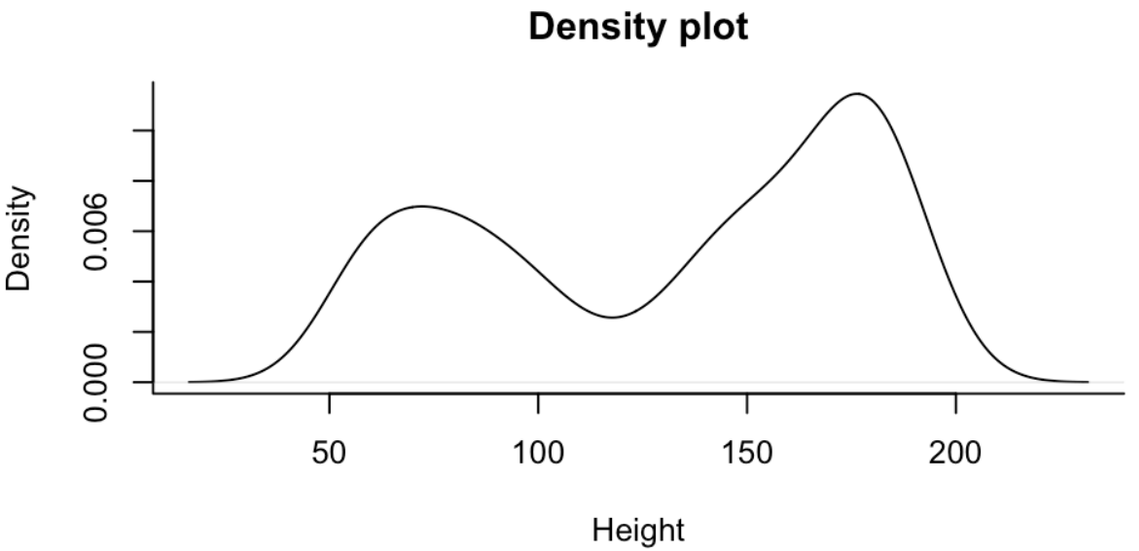


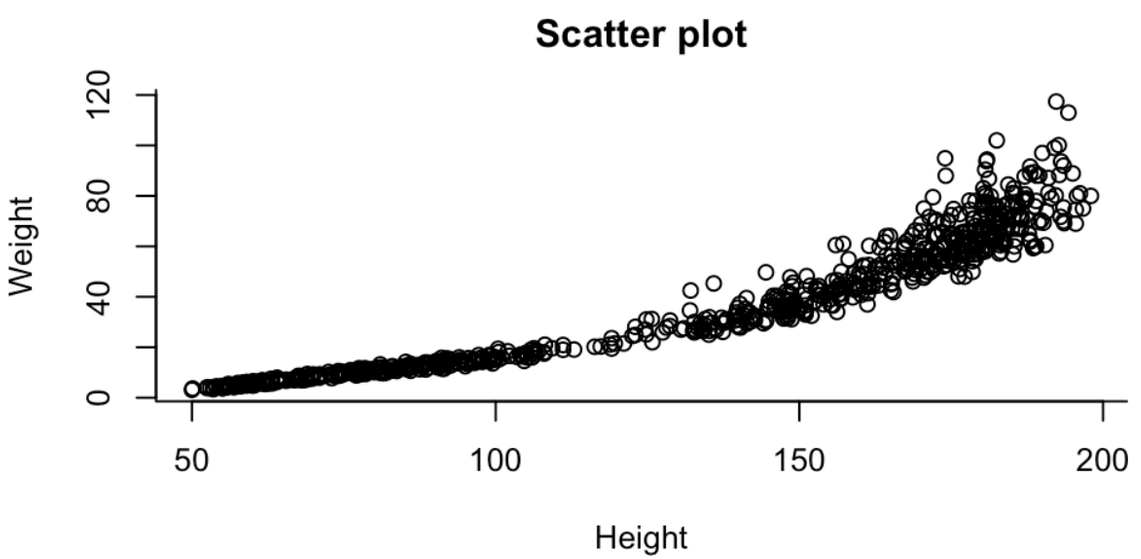
X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

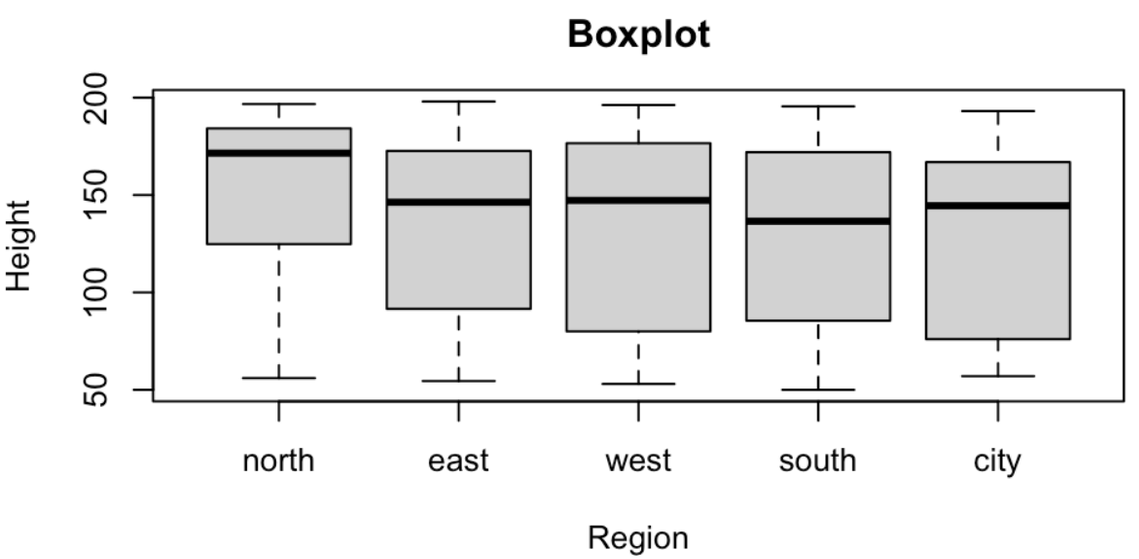
Source: https://www.autodeskresearch.com/publications/samestats

BASE R PLOTS


Histogram
Frequency
0
20
60
100
50
100
150
200
Height

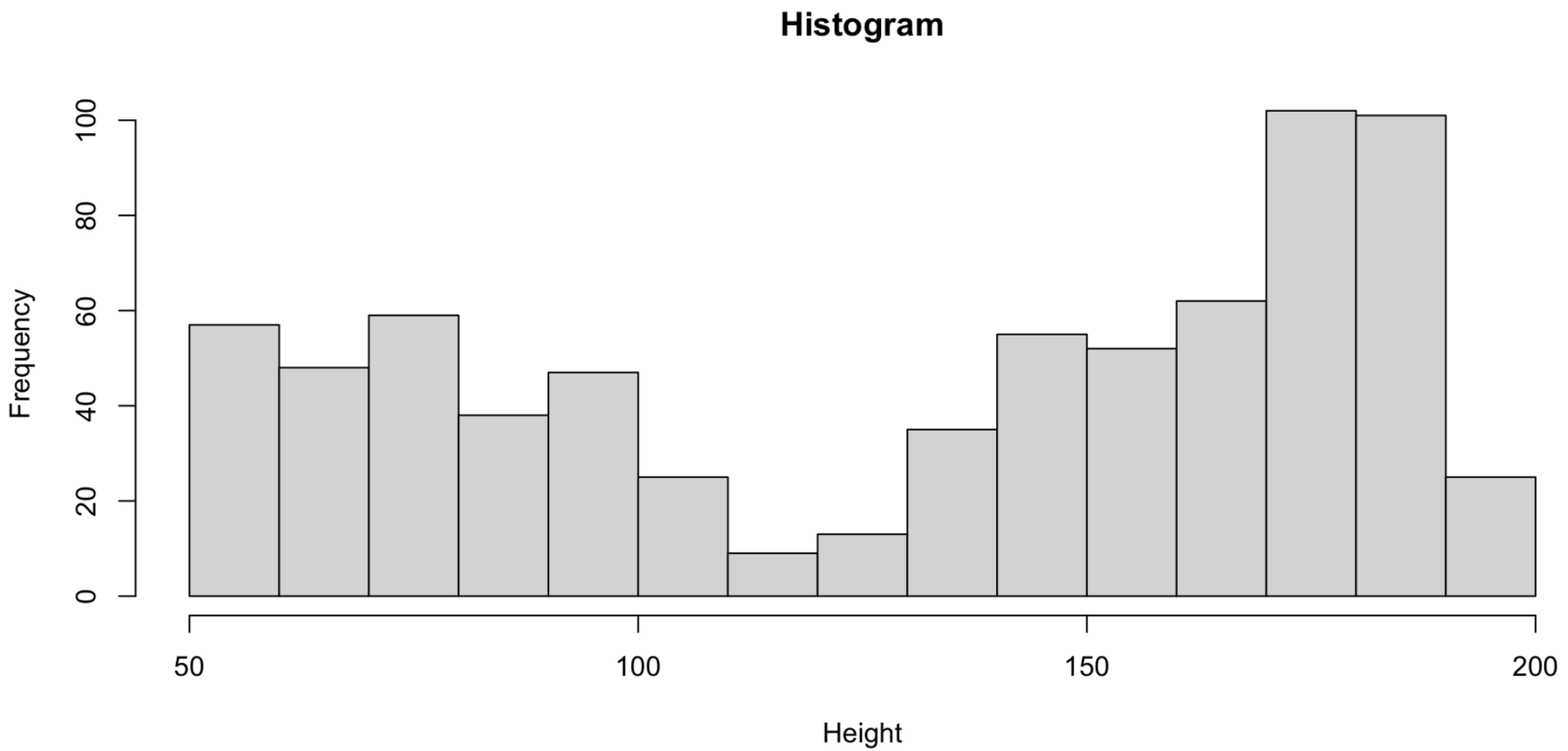

Density plot
Density
0.000
0.006
50
100
150
200
Height


Scatter plot
Weight
0
40
80
120
50
100
150
200
Height


Boxplot
Height
50
100
150
200
north
east
west
south
city
Region

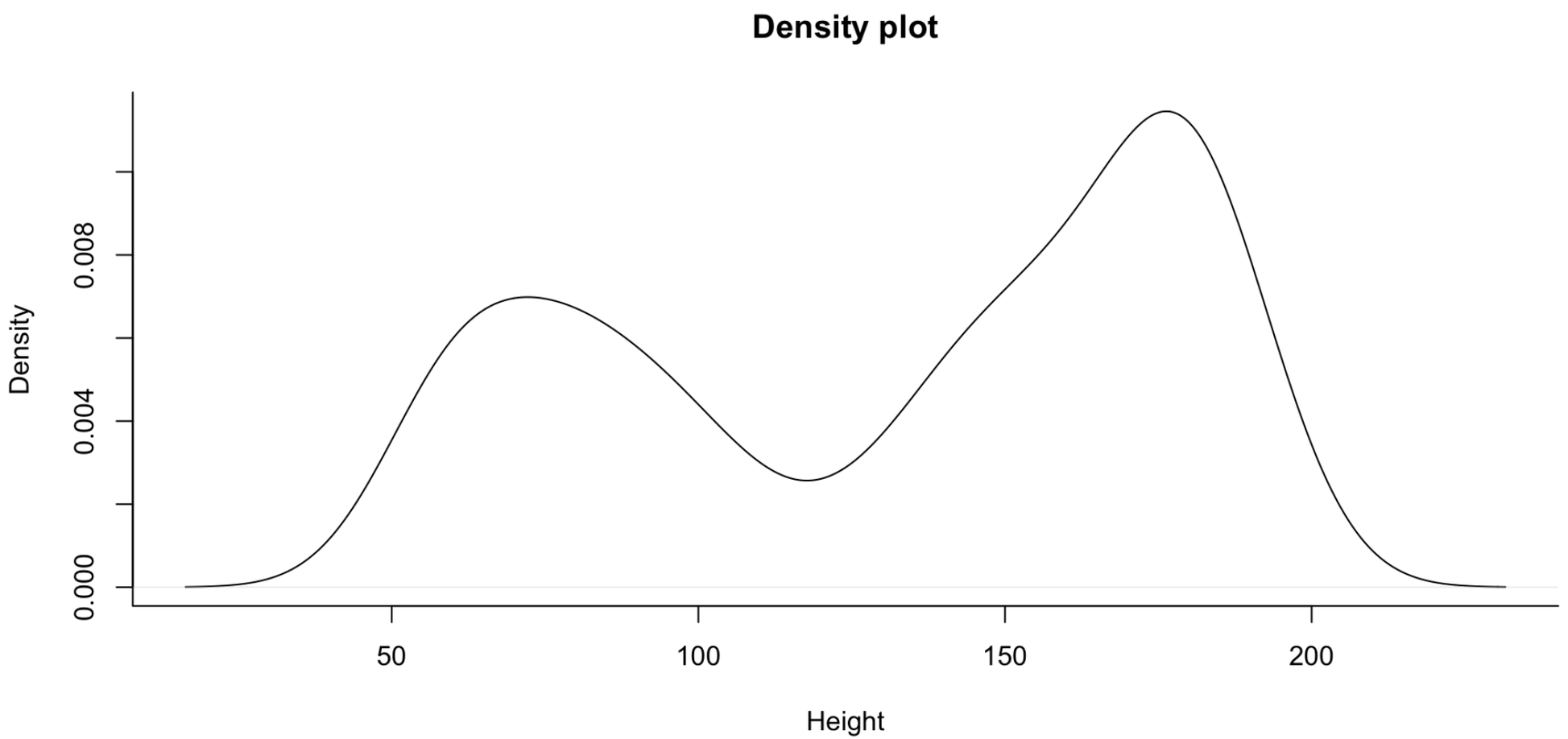
HISTOGRAM

```
1 hist(boys$hgt, main = "Histogram", xlab = "Height")
```



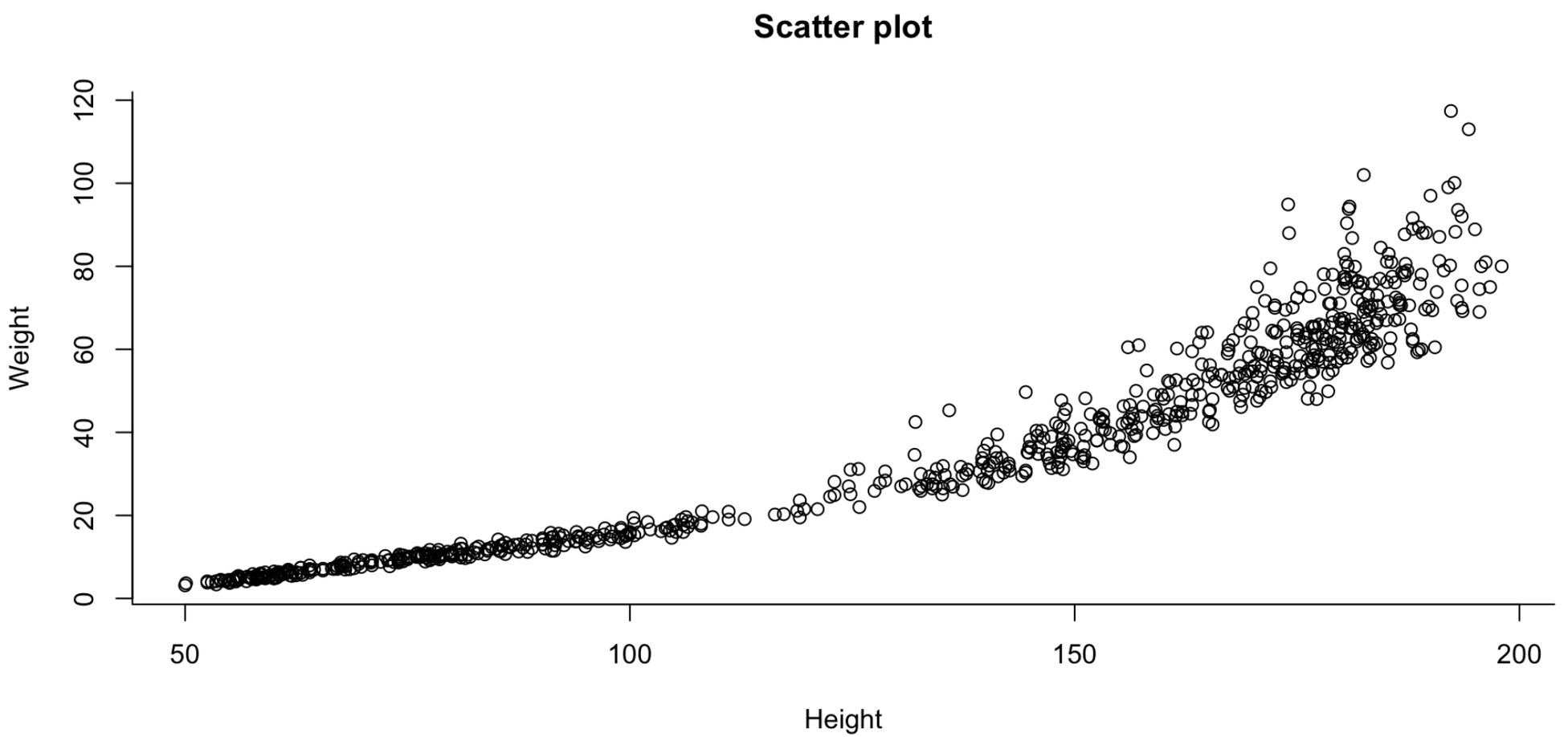
DENSITY

```
dens <- density(boys$hgt, na.rm = TRUE)
plot(dens, main = "Density plot", xlab = "Height", bty = "L")
```



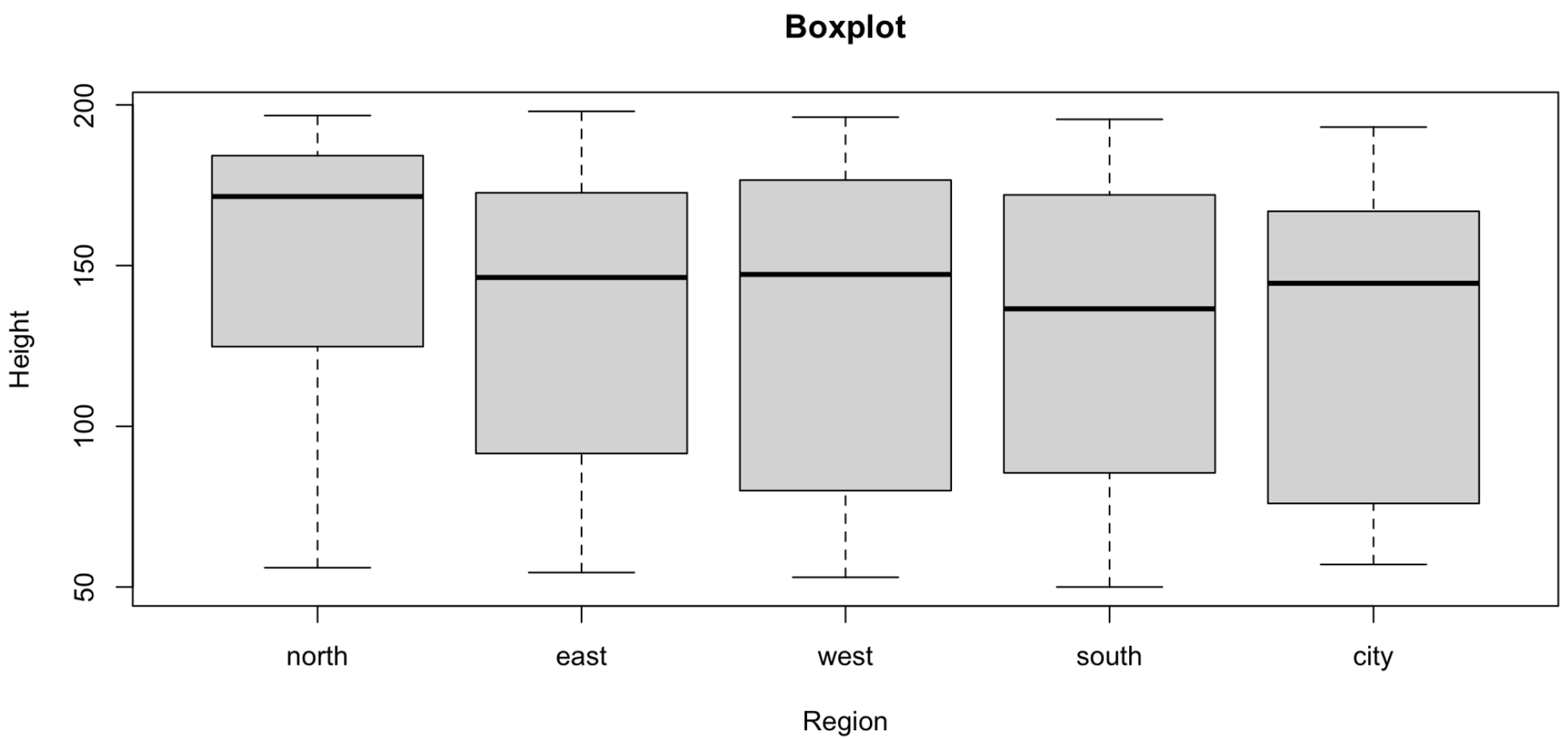
SCATTER PLOT

```
plot(x = boys$hgt, y = boys$wgt, main = "Scatter plot",
     xlab = "Height", ylab = "Weight", bty = "L")
```



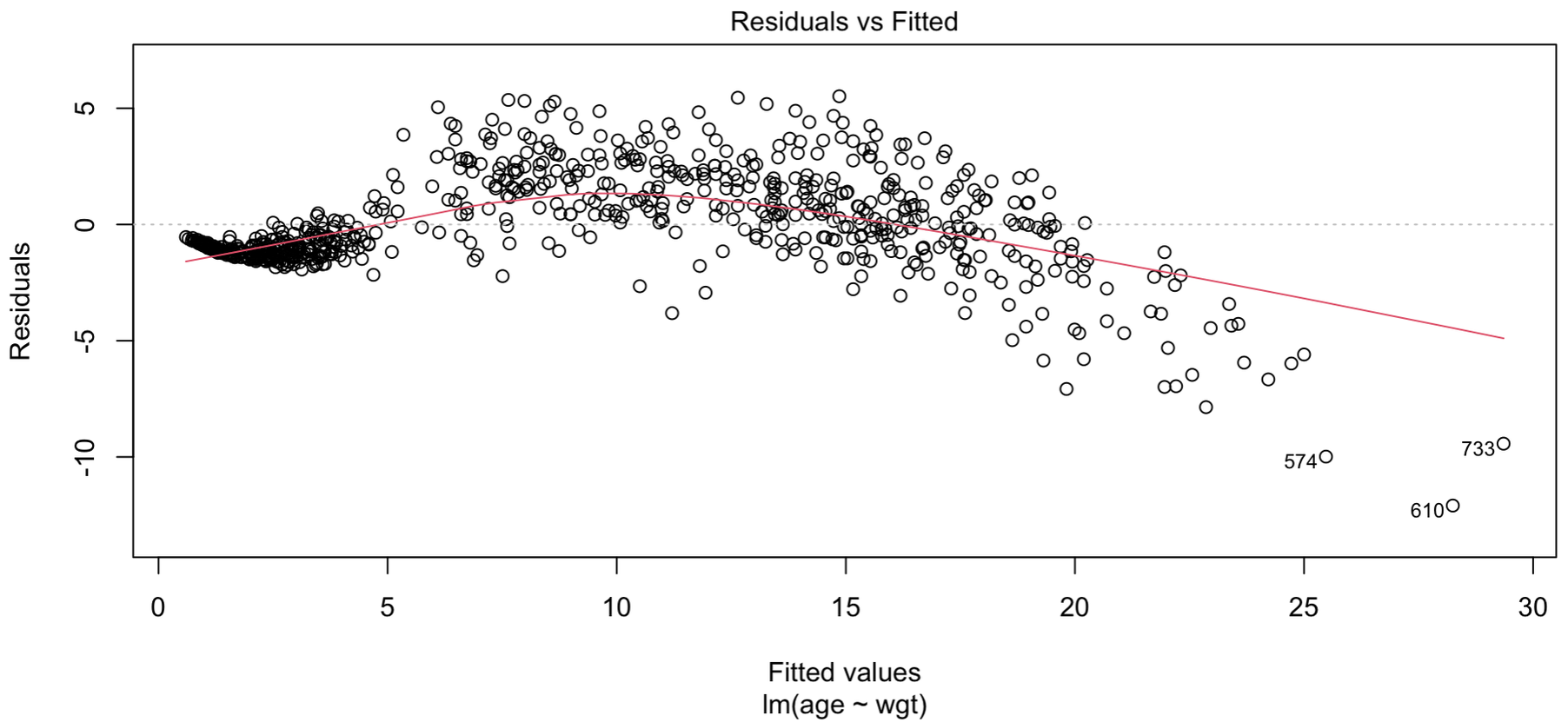
BOX PLOT

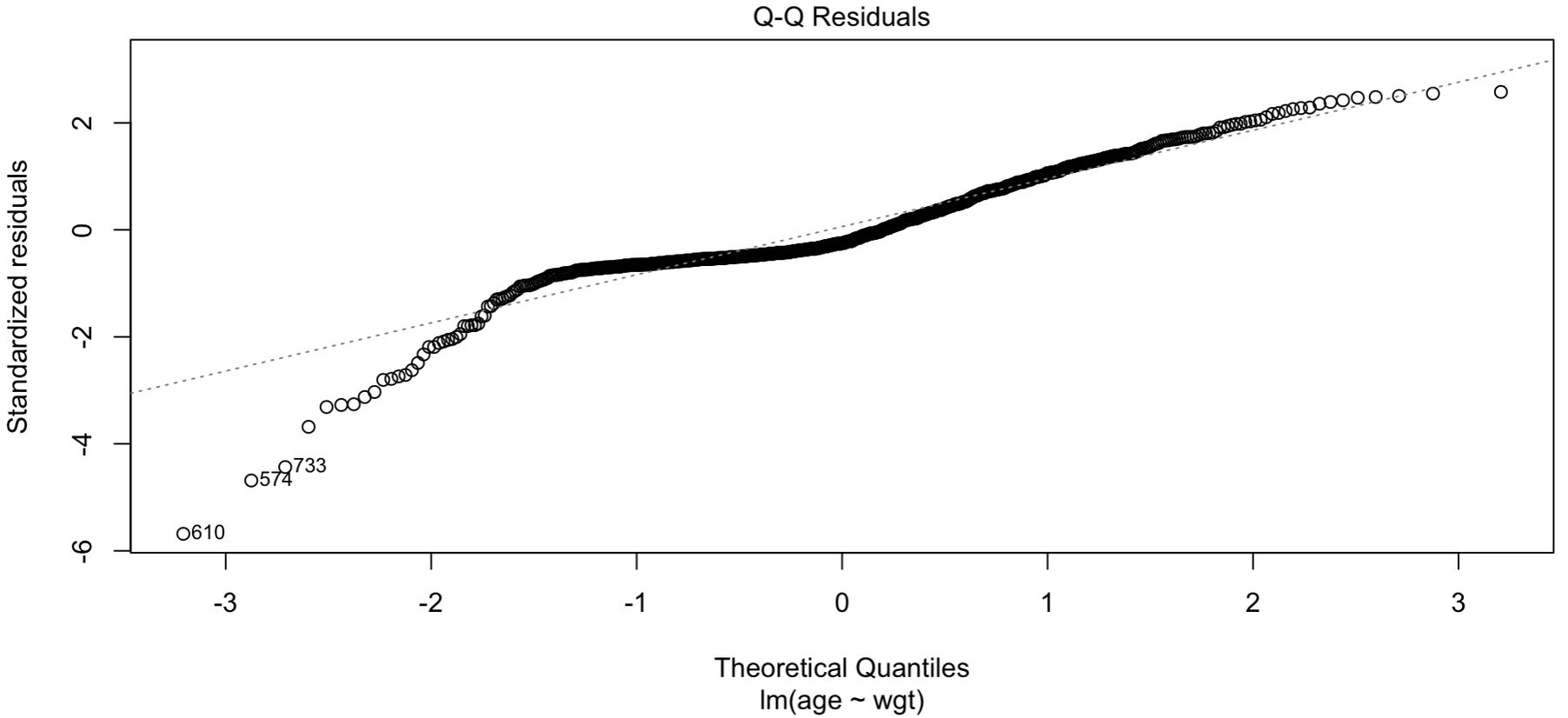
```
1 boxplot(boys$hgt ~ boys$reg, main = "Boxplot",
2         xlab = "Region", ylab = "Height")
```

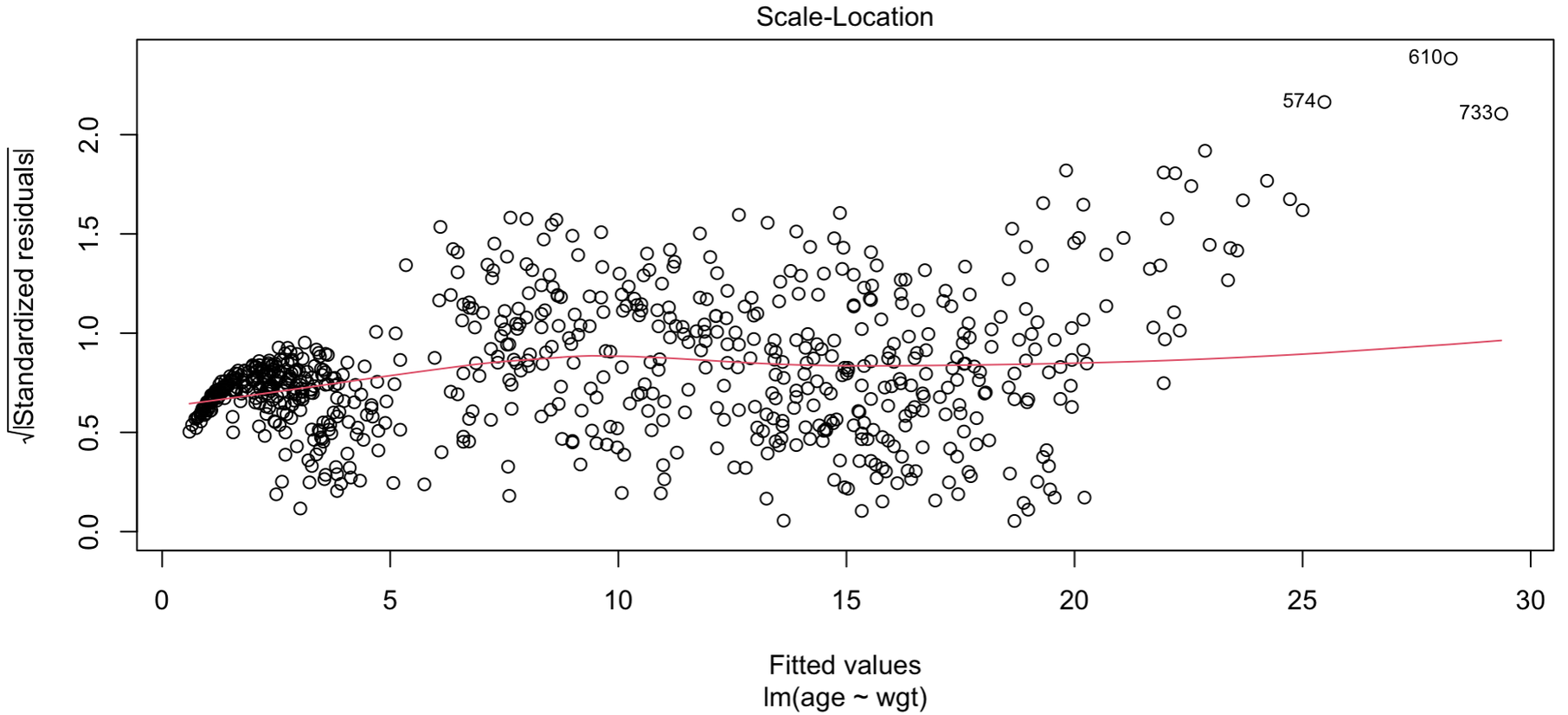


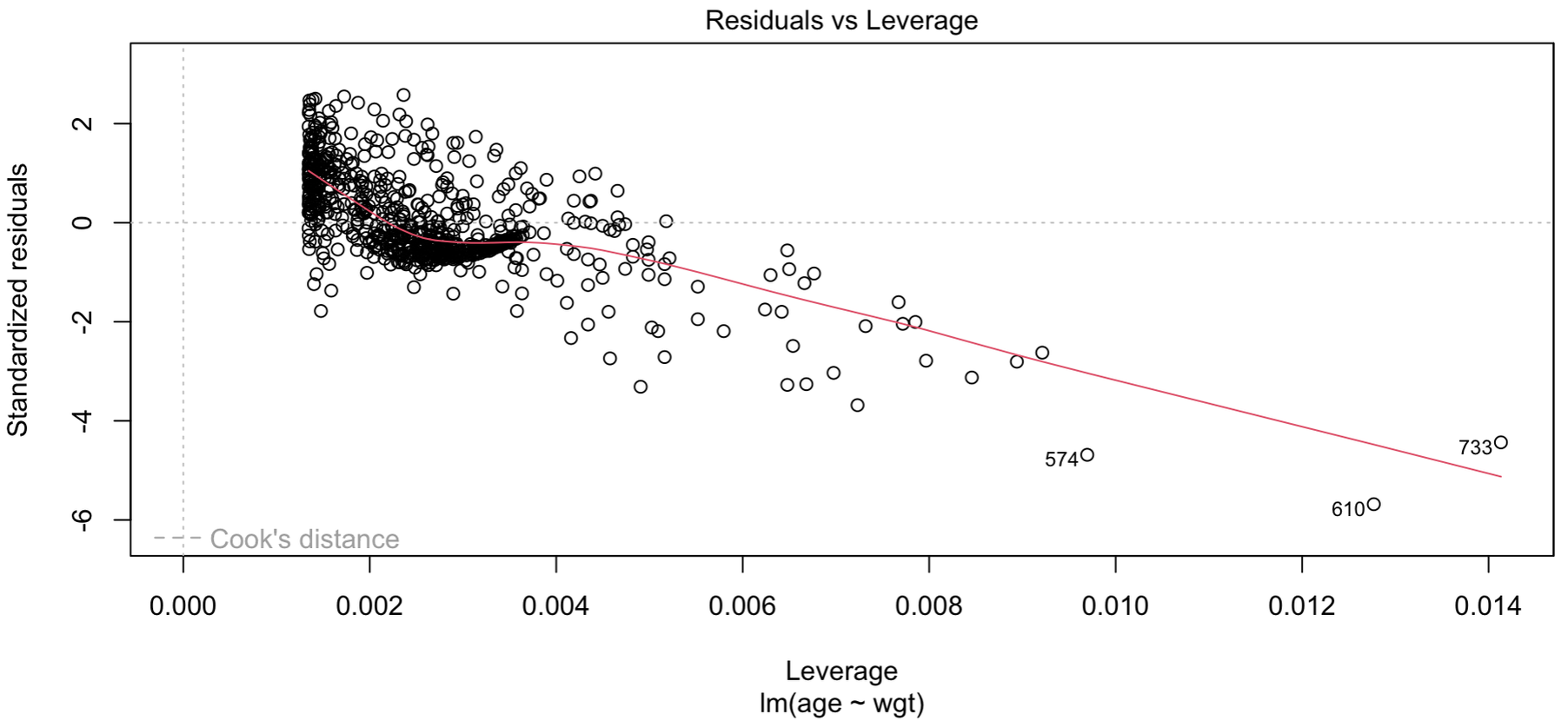
MANY R OBJECTS ALSO HAVE A plot() METHOD

```
1  boys %$% lm(age~wgt) %>% plot()
```




Q-Q Residuals
Standardized residuals
2
0
-2
-4
-6
610
574
733
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(age ~ wgt)


Scale-Location
610
574
733
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
5
10
15
20
25
30
Fitted values
lm(age ~ wgt)


Residuals vs Leverage
Standardized residuals
2
0
-2
-4
-6
Cook's distance
574
733
610
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
Leverage
lm(age ~ wgt)

NEAT! BUT WHAT IF WE WANT MORE CONTROL?

GGPLOT2

WHAT IS `ggplot2`?

Layered plotting based on the book **The Grammer of Graphics** by Leland Wilkinsons.

With `ggplot2` you

1. provide the *data*
2. define how to map variables to *aesthetics*
3. state which *geometric object* to display
4. (optional) edit the overall *theme* of the plot

`ggplot2` then takes care of the details

AN EXAMPLE: SCATTERPLOT

1: Provide the data

```
boys %>%
  ggplot()
```

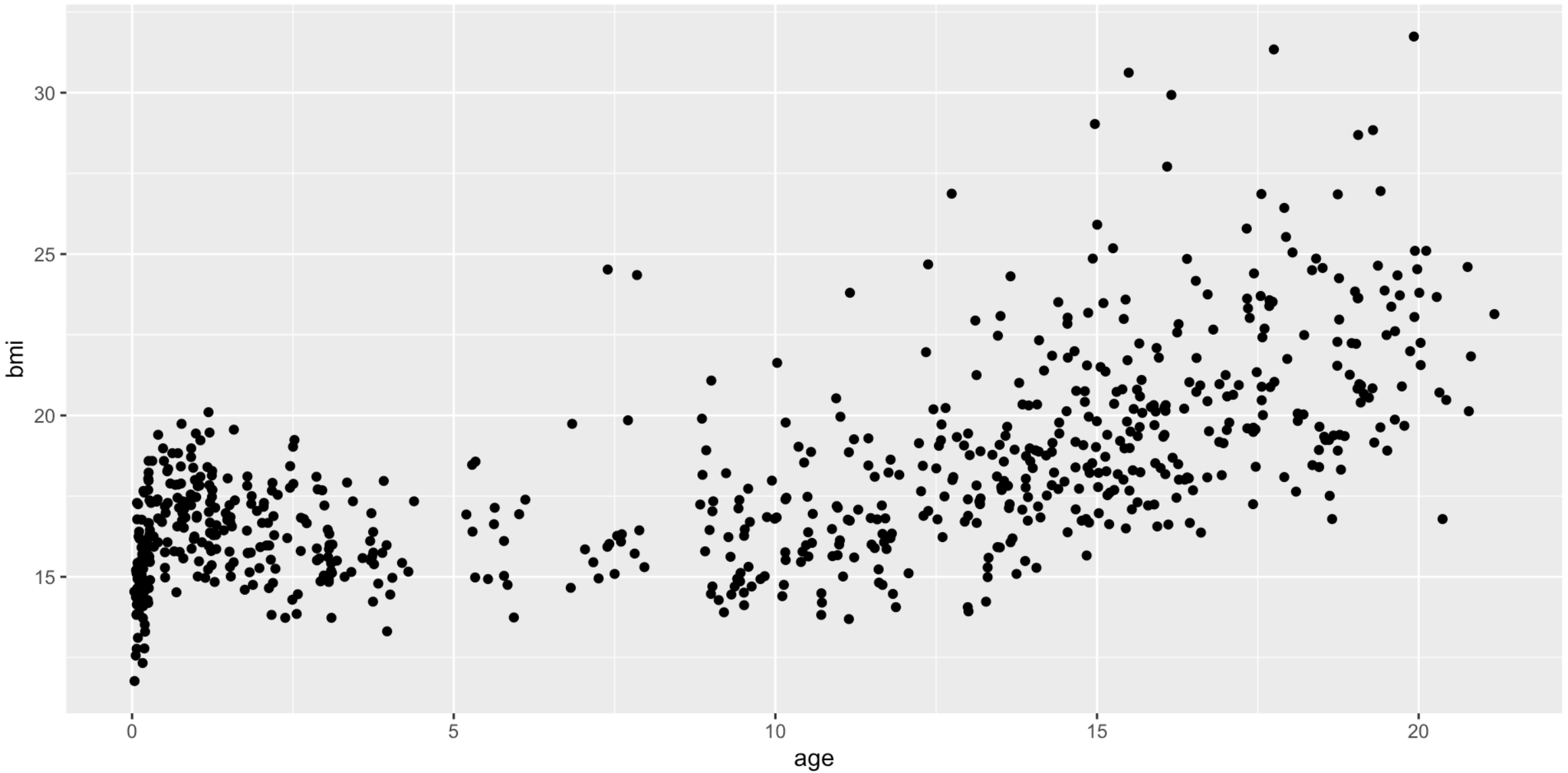
2: map variable to aesthetics

```
boys %>%
  ggplot(aes(x = age, y = bmi))
```

3: state which geometric object to display

```
boys %>%
  ggplot(aes(x = age, y = bmi)) +
  geom_point()
```

AN EXAMPLE: SCATTERPLOT



WHY THIS SYNTAX?

Create the plot

```
gg <-
  boys %>%
  ggplot(aes(x = age, y = bmi)) +
  geom_point(col = "dark green")
```

Add another layer (smooth fit line)

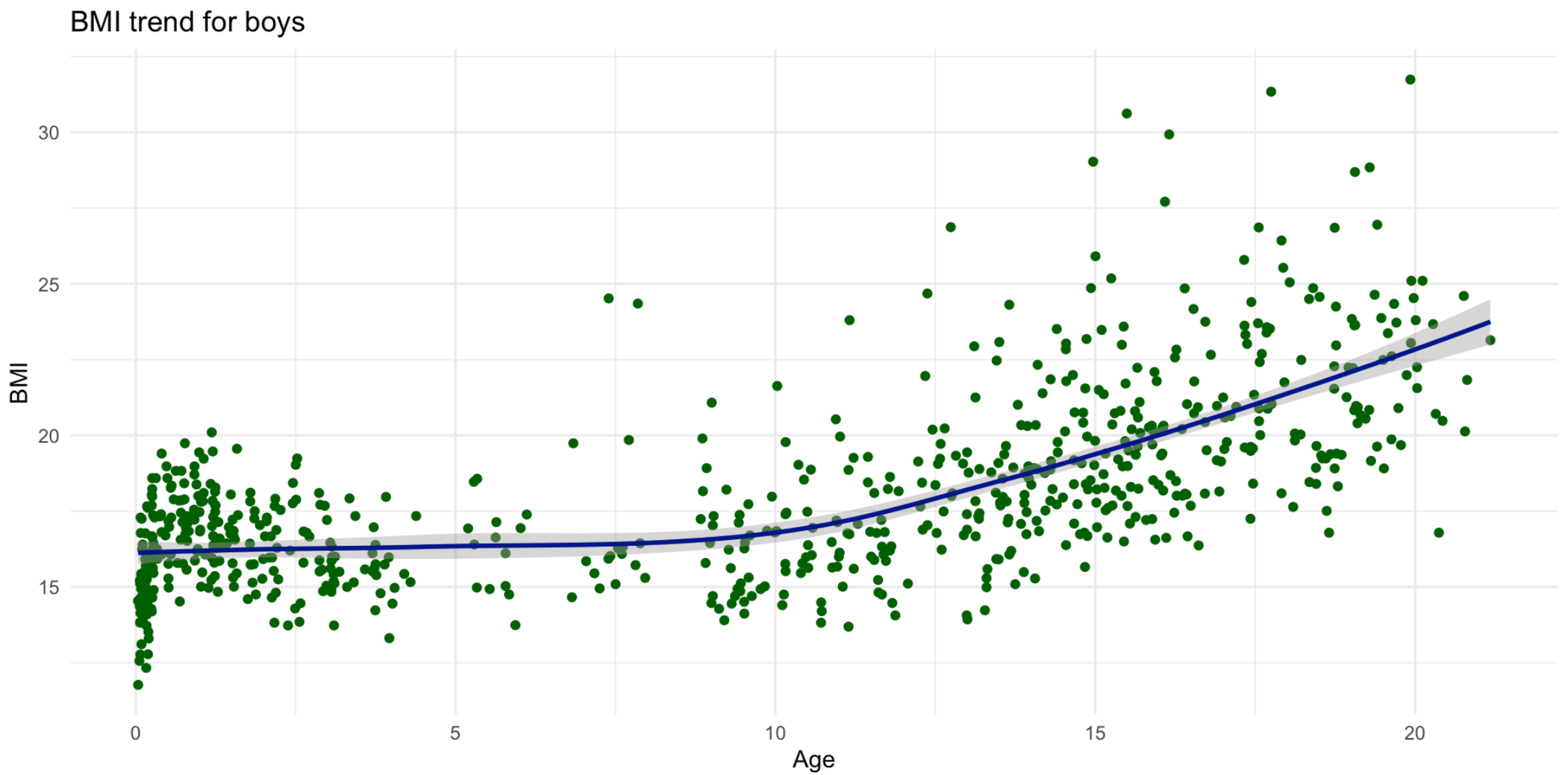
```
gg <- gg +
  geom_smooth(col = "dark blue")
```

Give it some labels and a nice look

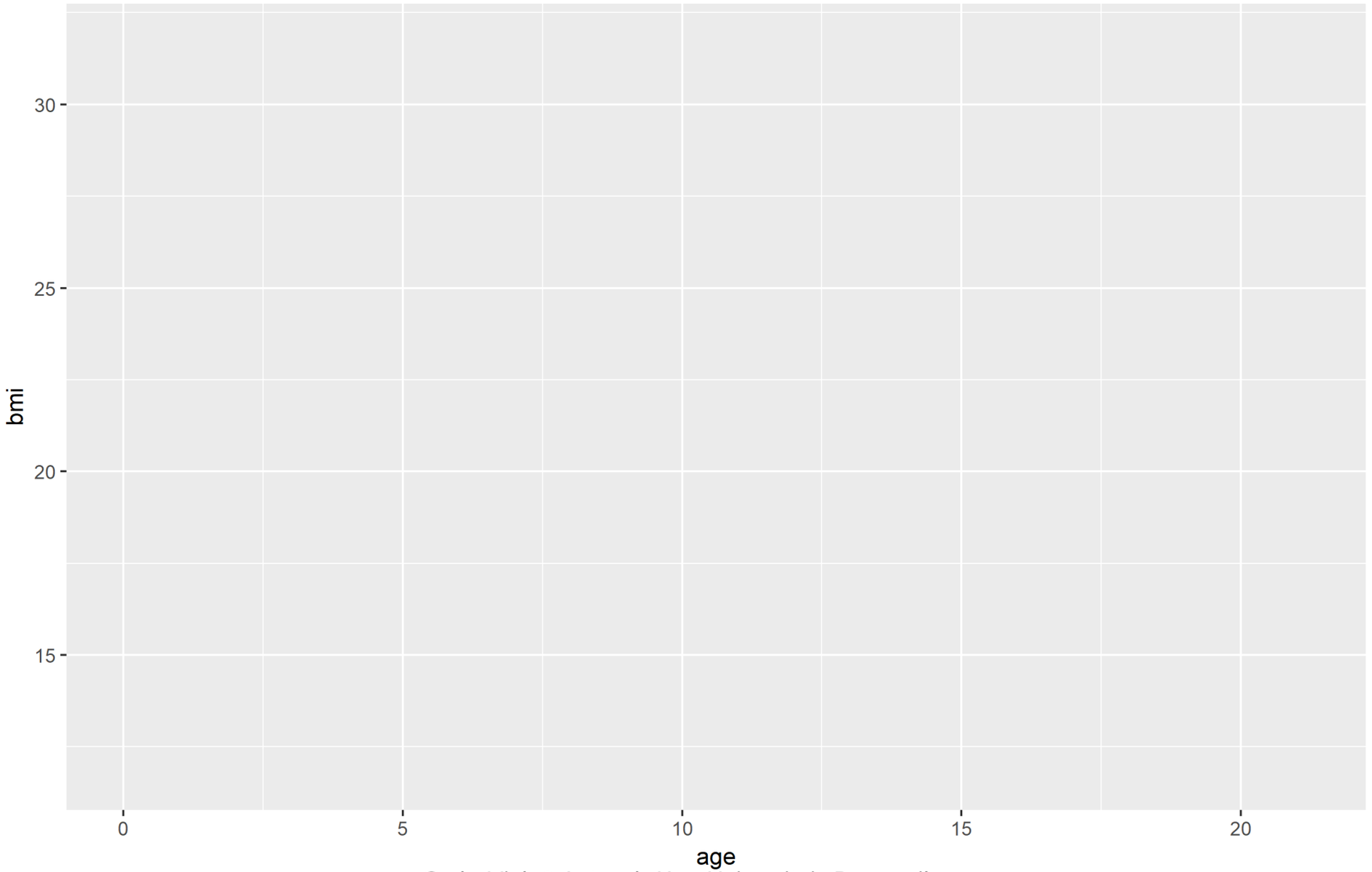
```
gg <- gg +
  labs(x = "Age", y = "BMI", title = "BMI trend for boys") +
  theme_minimal()
```

WHY THIS SYNTAX?

```
1 plot(gg)
```



WHY THIS SYNTAX?



AESTHETICS

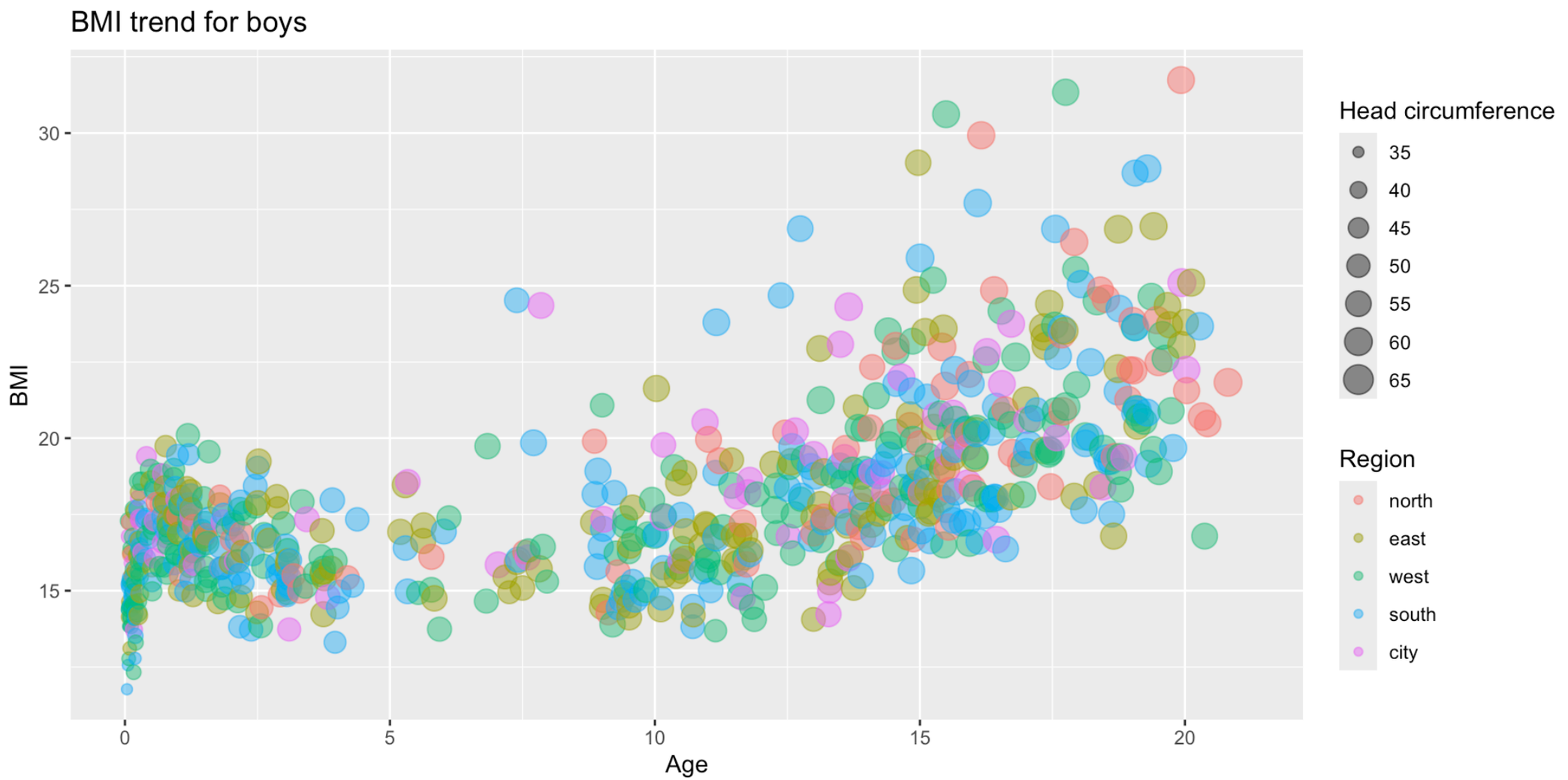
- x
- y
- size
- colour
- fill
- opacity (alpha)
- linetype
- …

AESTHETICS

```
gg <-
  boys %>%
  filter(!is.na(reg)) %>%

  ggplot(aes(x      = age,
             y      = bmi,
             size   = hc,
             colour = reg)) +

  geom_point(alpha = 0.5) +

  labs(title  = "BMI trend for boys",
       x      = "Age",
       y      = "BMI",
       size   = "Head circumference",
       colour = "Region")
```

AESTHETICS

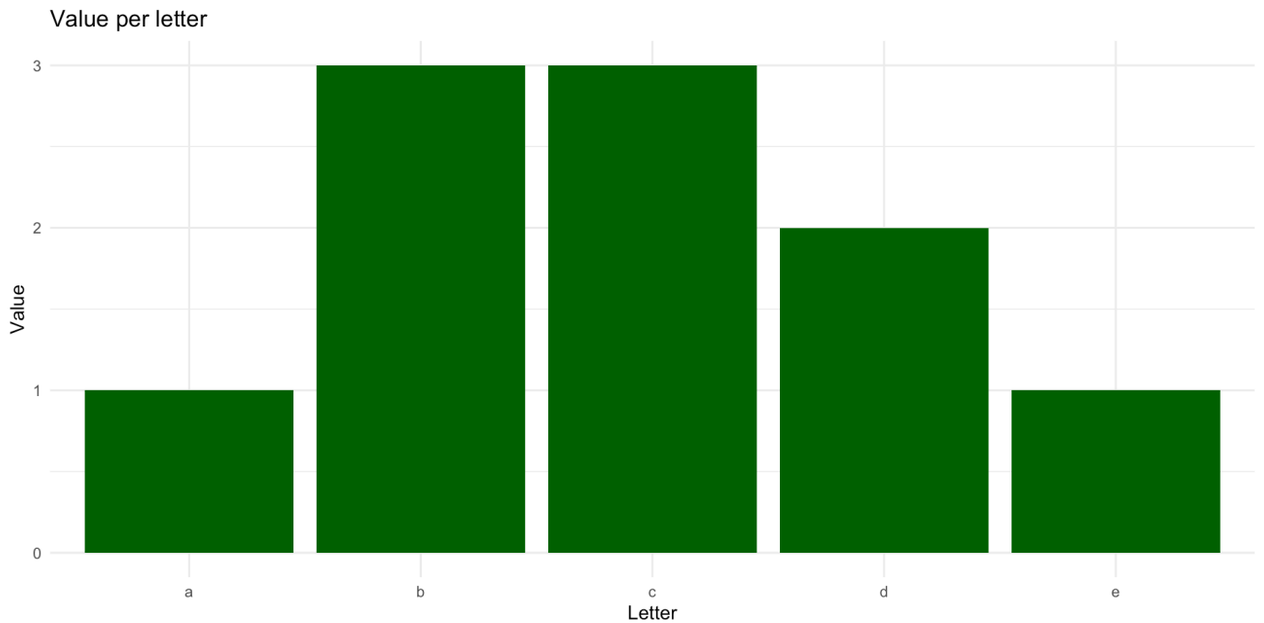
```
1 plot(gg)
```



GEOMS

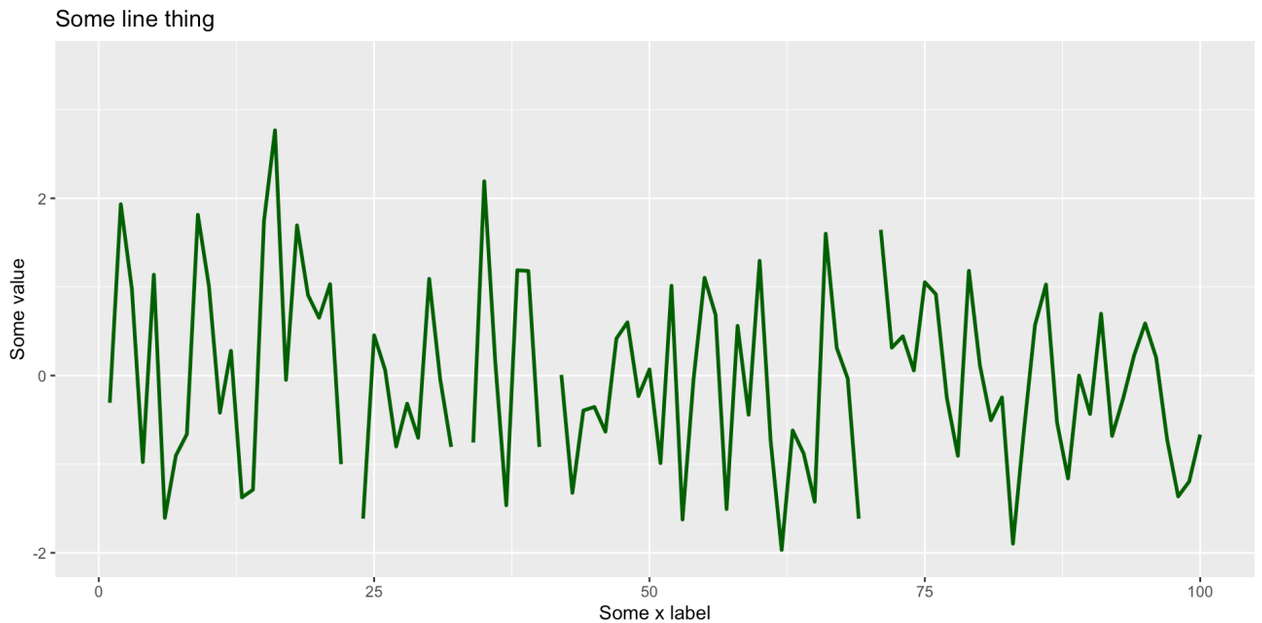
- geom_point
- geom_bar
- geom_line
- geom_smooth
- geom_histogram
- geom_boxplot
- geom_density

GEOMS: BAR



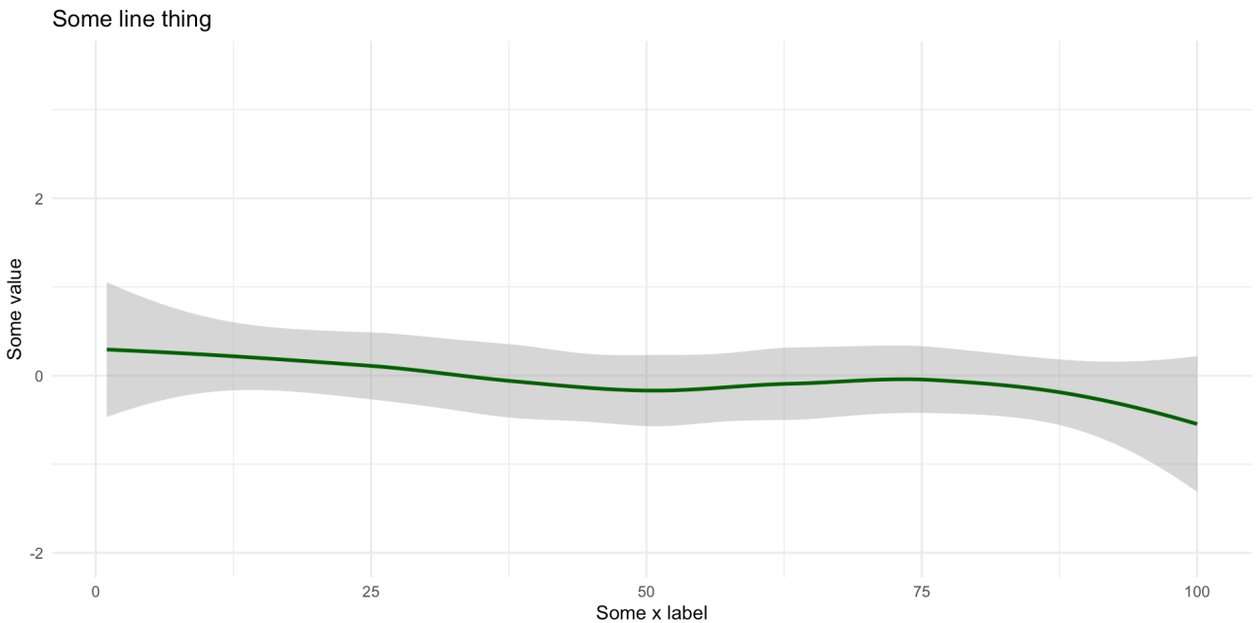

```
data.frame(x = letters[1:5],
           y = c(1, 3, 3, 2, 1)) %>%
  ggplot(aes(x = x, y = y)) +
  geom_bar(fill = "dark green",
           stat = "identity") +
  labs(title = "Value per letter",
       x     = "Letter",
       y     = "Value")
```

GEOMS: LINE



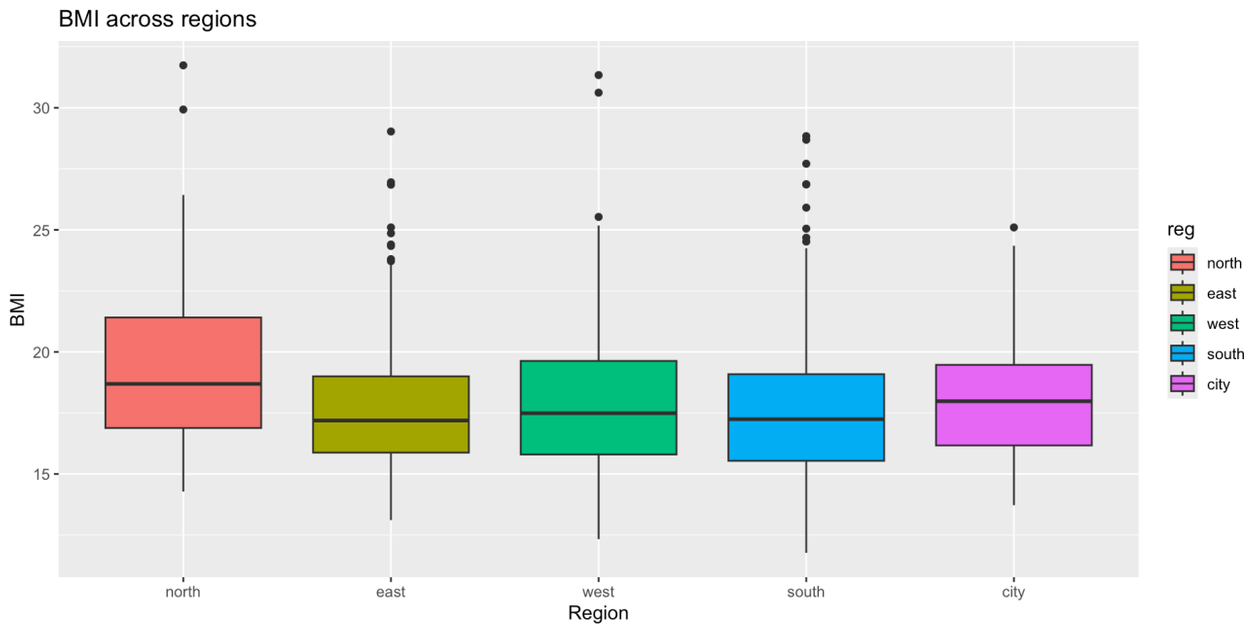

```r
ggdat <- data.frame(x = 1:100,
                    y = rnorm(100))
ggdat %>%
  ggplot(aes(x = x, y = y)) +
  geom_line(colour = "dark green",
            lwd = 1) +
  ylim(-2, 3.5) +
  labs(title = "Some line thing",
       x     = "Some x label",
       y     = "Some value")
```

GEOMS: SMOOTH



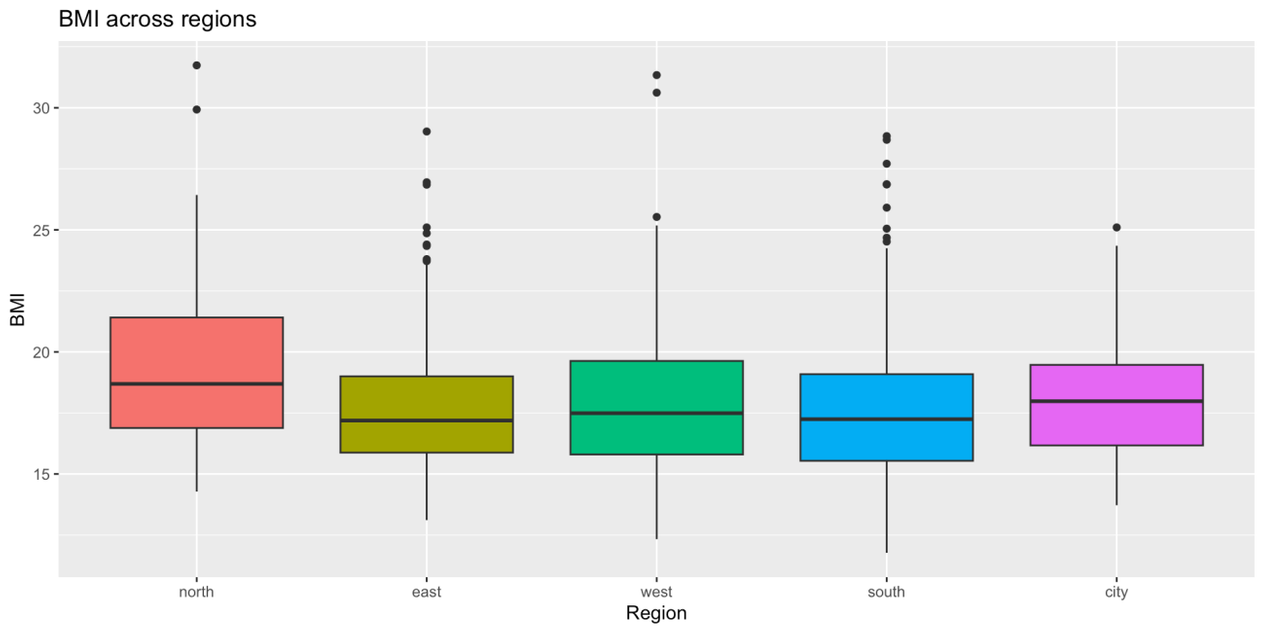

```
ggdat %>%
  ggplot(aes(x = x, y = y)) +
  geom_smooth(colour = "dark green",
              lwd = 1,
              se = TRUE) +
  ylim(-2, 3.5) +
  labs(title = "Some line thing",
       x     = "Some x label",
       y     = "Some value")
```

GEOMS: BOXPLOT



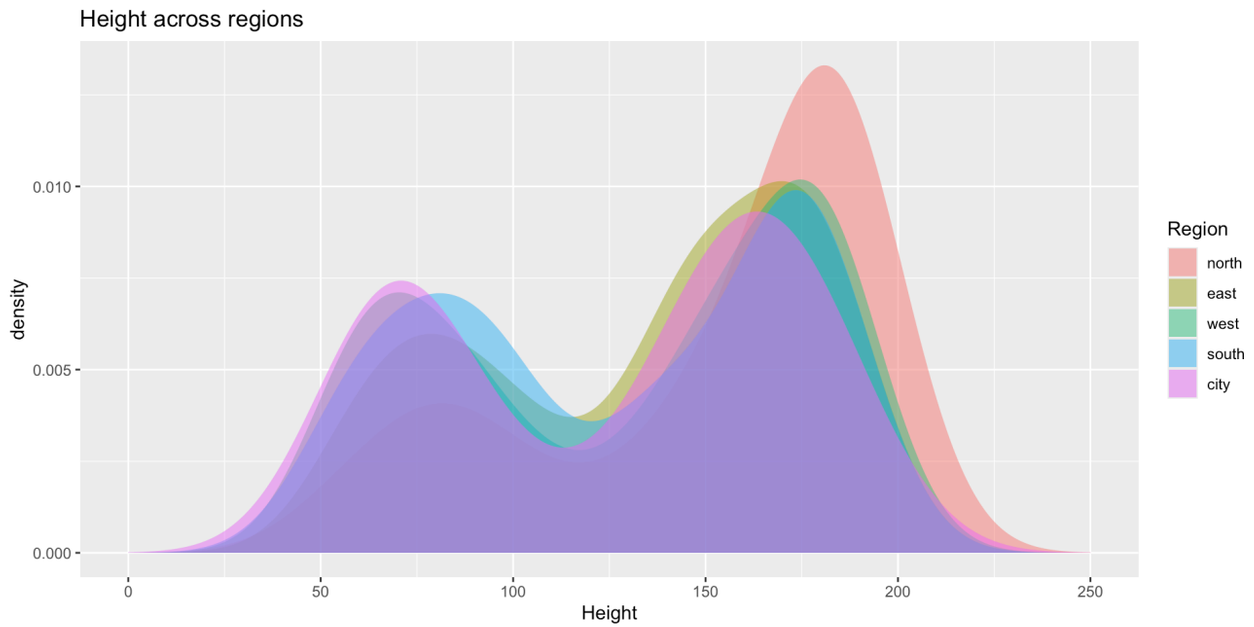

```
boys %>%
  filter(!is.na(reg)) %>%
  ggplot(aes(x = reg,
             y = bmi,
             fill = reg)) +
  geom_boxplot() +
  labs(title = "BMI across regions",
       x     = "Region",
       y     = "BMI")
```

GEOMS: BOXPLOT WITHOUT LEGEND




```
boys %>%
  filter(!is.na(reg)) %>%
  ggplot(aes(x = reg,
             y = bmi,
             fill = reg)) +
  geom_boxplot() +
  labs(title = "BMI across regions",
       x     = "Region",
       y     = "BMI") +
  theme(legend.position = "none")
```

GEOMS: DENSITY

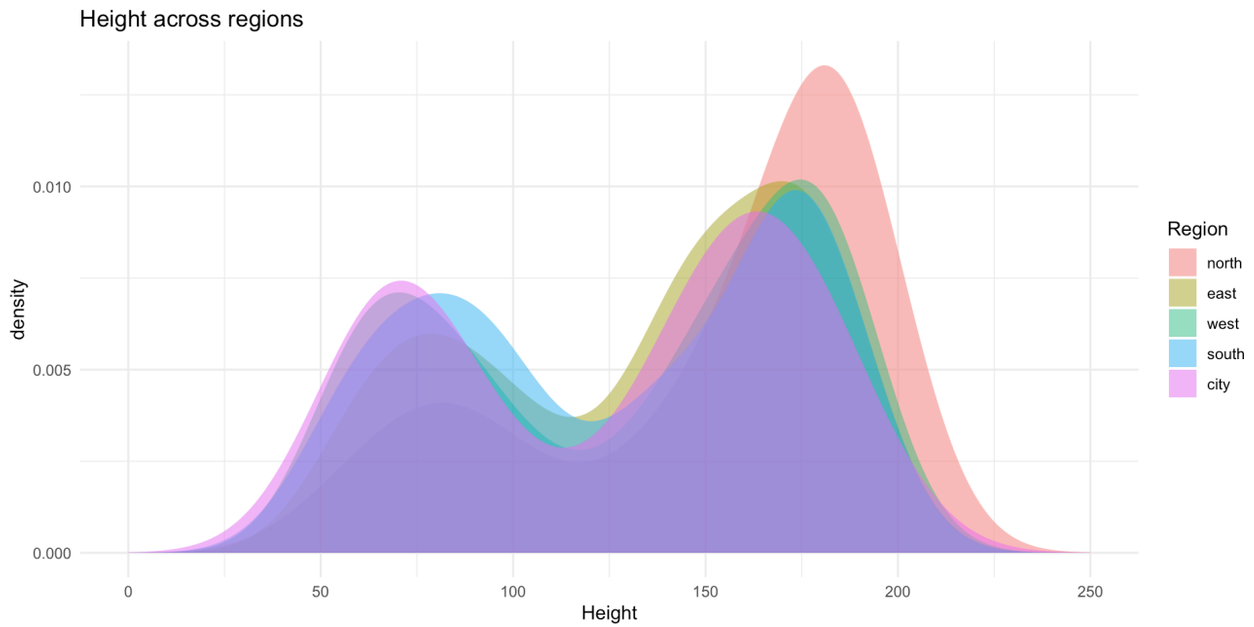



```
boys %>%
  filter(!is.na(reg)) %>%
  ggplot(aes(x = hgt, fill = reg)) +
  geom_density(alpha = 0.5,
               colour = "transparent") +
  xlim(0, 250) +
  labs(title = "Height across regions",
       x     = "Height",
       fill  = "Region")
```

CHANGING THE STYLE: THEMES

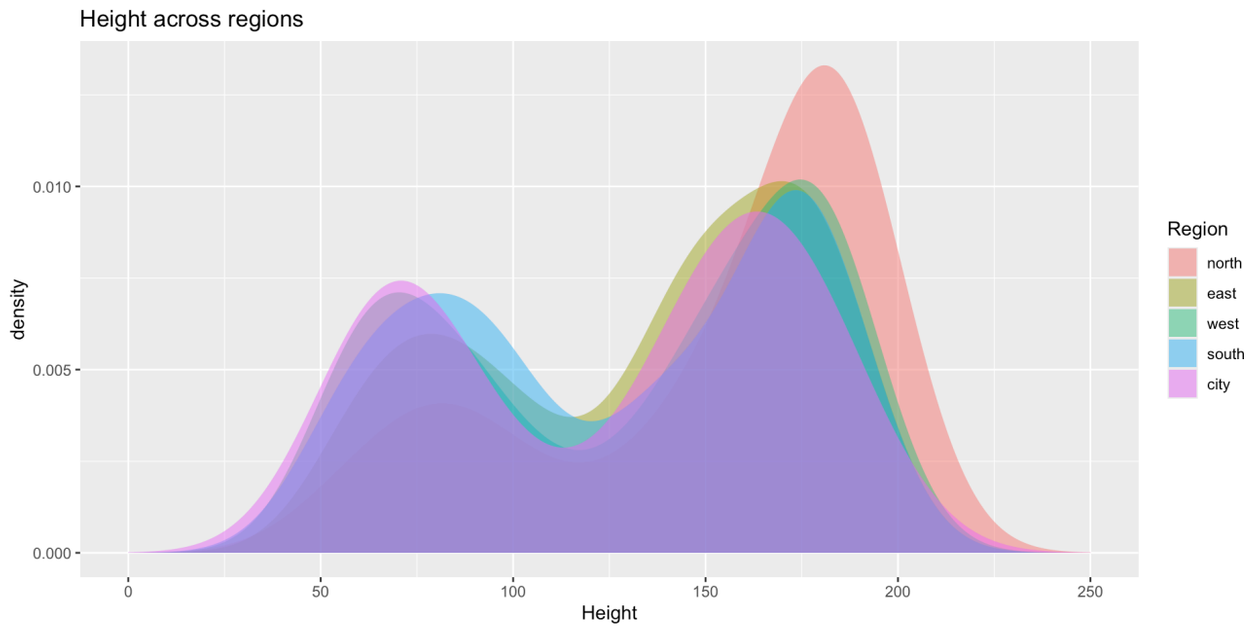
- Themes determine the overall appearance of your plot
- standard themes: e.g., `theme_minimal()`, `theme_classic()`, `theme_bw()`, ...
- extra libraries with additional themes: e.g., `ggthemes`
- customize own theme using options of `theme()`

CHANGING THE STYLE: THEMES



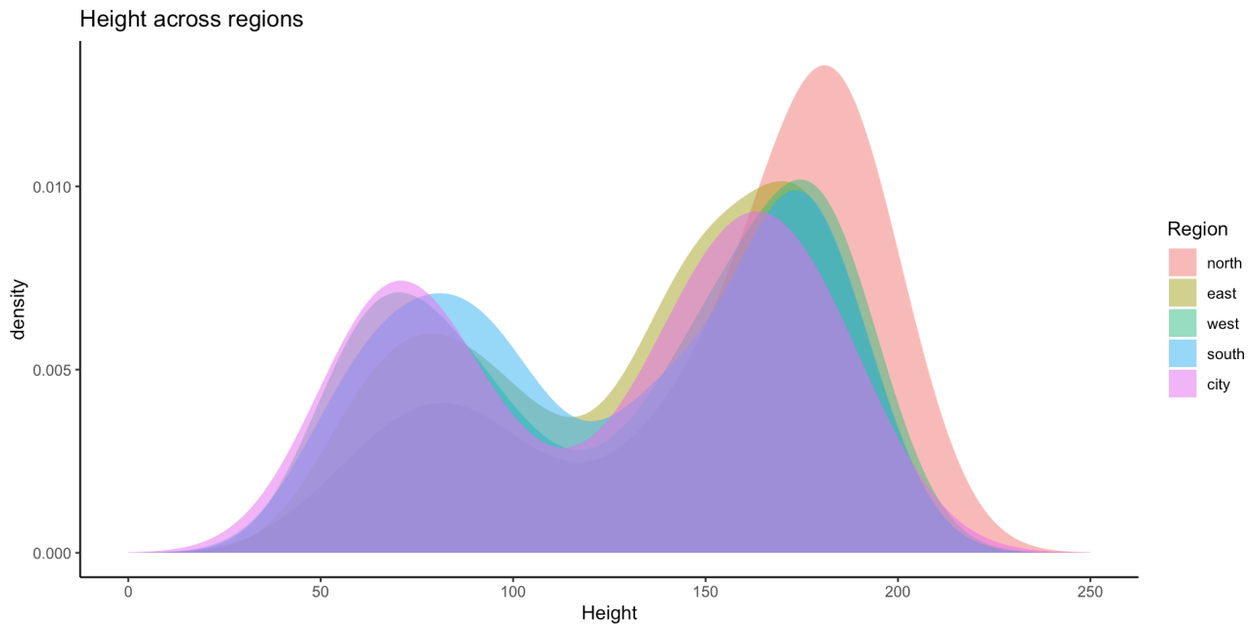

```
boys %>%
  filter(!is.na(reg)) %>%
  ggplot(aes(x = hgt, fill = reg)) +
  geom_density(alpha = 0.5,
               colour = "transparent") +
  xlim(0, 250) +
  labs(title = "Height across regions",
       x     = "Height",
       fill  = "Region") +
  theme_minimal()
```

CHANGING THE STYLE: THEMES



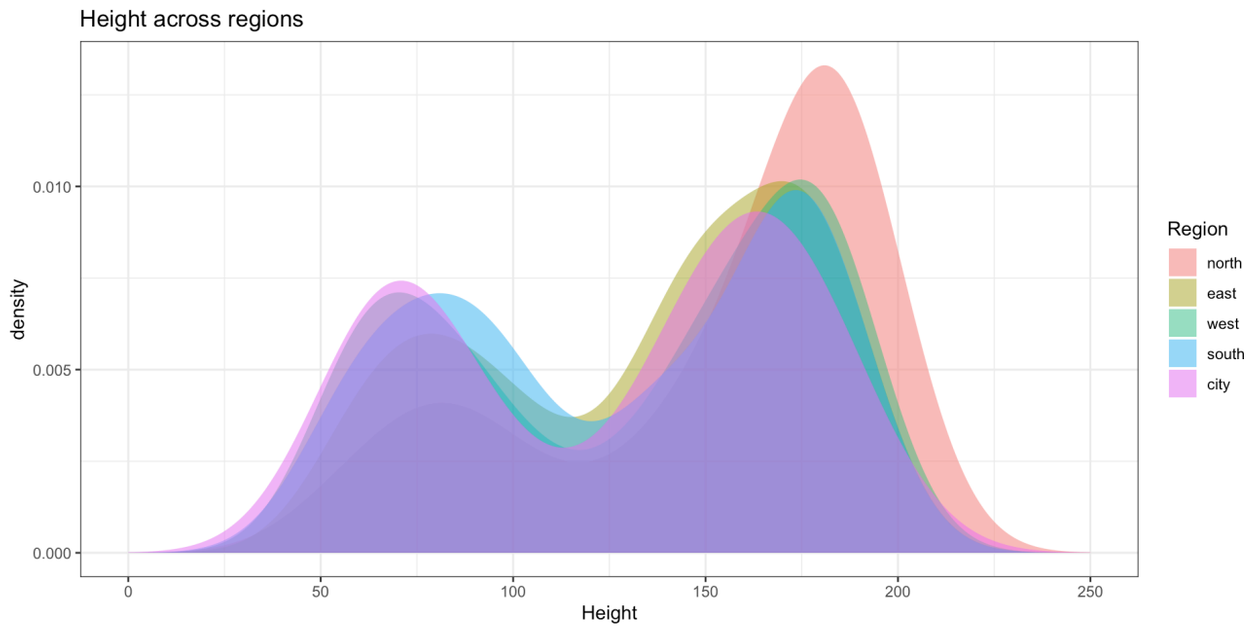

```
boys %>%
  filter(!is.na(reg)) %>%
  ggplot(aes(x = hgt, fill = reg)) +
  geom_density(alpha = 0.5,
               colour = "transparent") +
  xlim(0, 250) +
  labs(title = "Height across regions",
       x     = "Height",
       fill  = "Region") +
  theme_gray()
```

CHANGING THE STYLE: THEMES




```
boys %>%
  filter(!is.na(reg)) %>%
  ggplot(aes(x = hgt, fill = reg)) +
  geom_density(alpha = 0.5,
               colour = "transparent") +
  xlim(0, 250) +
  labs(title = "Height across regions",
       x     = "Height",
       fill  = "Region") +
  theme_classic()
```

CHANGING THE STYLE: THEMES




```
boys %>%
  filter(!is.na(reg)) %>%
  ggplot(aes(x = hgt, fill = reg)) +
  geom_density(alpha = 0.5,
               colour = "transparent") +
  xlim(0, 250) +
  labs(title = "Height across regions",
       x     = "Height",
       fill  = "Region") +
  theme_bw()
```

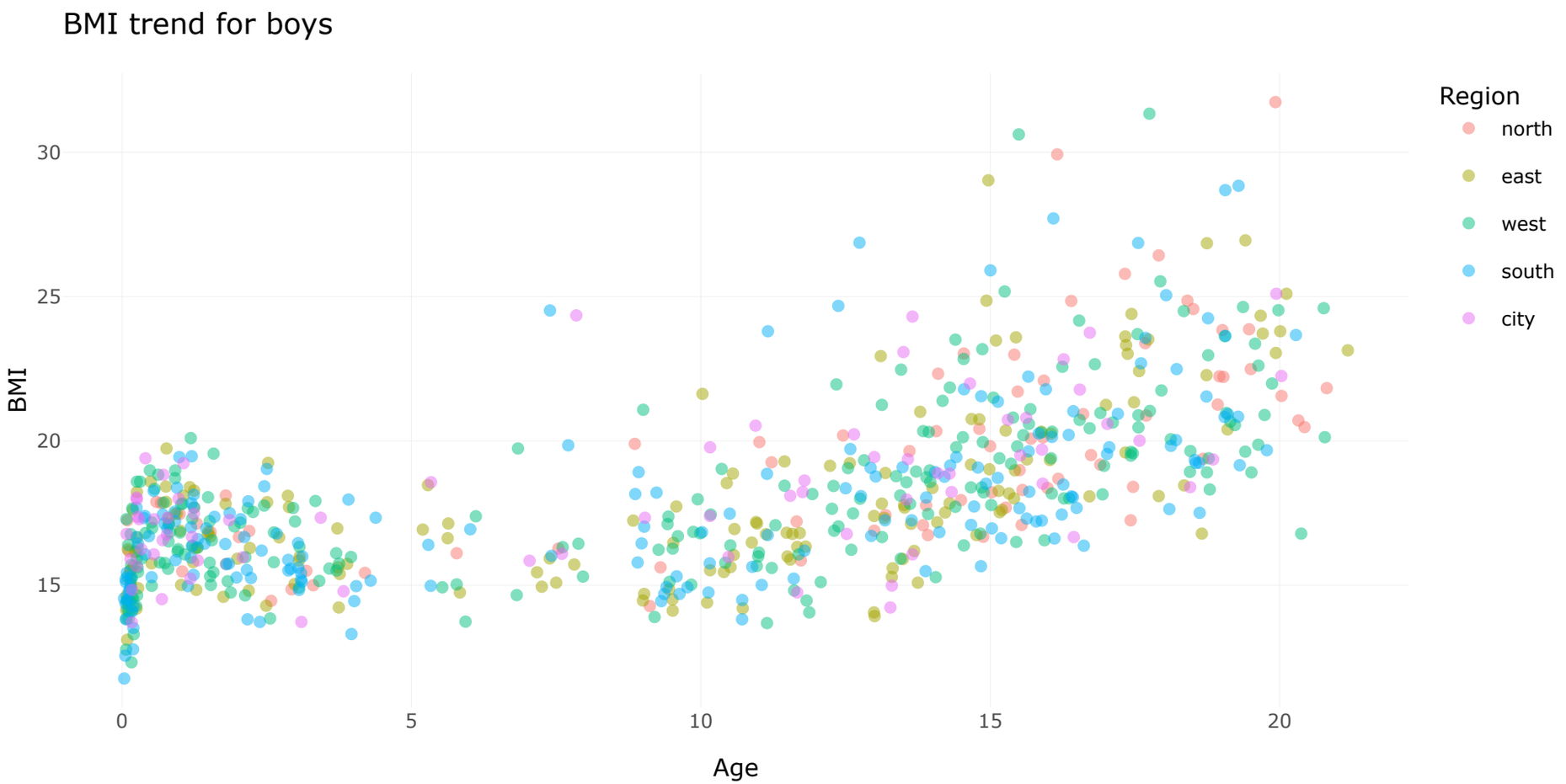
INTERACTIVE PLOTS

Use `plotly::ggplotly()` to make any ggplot interactive

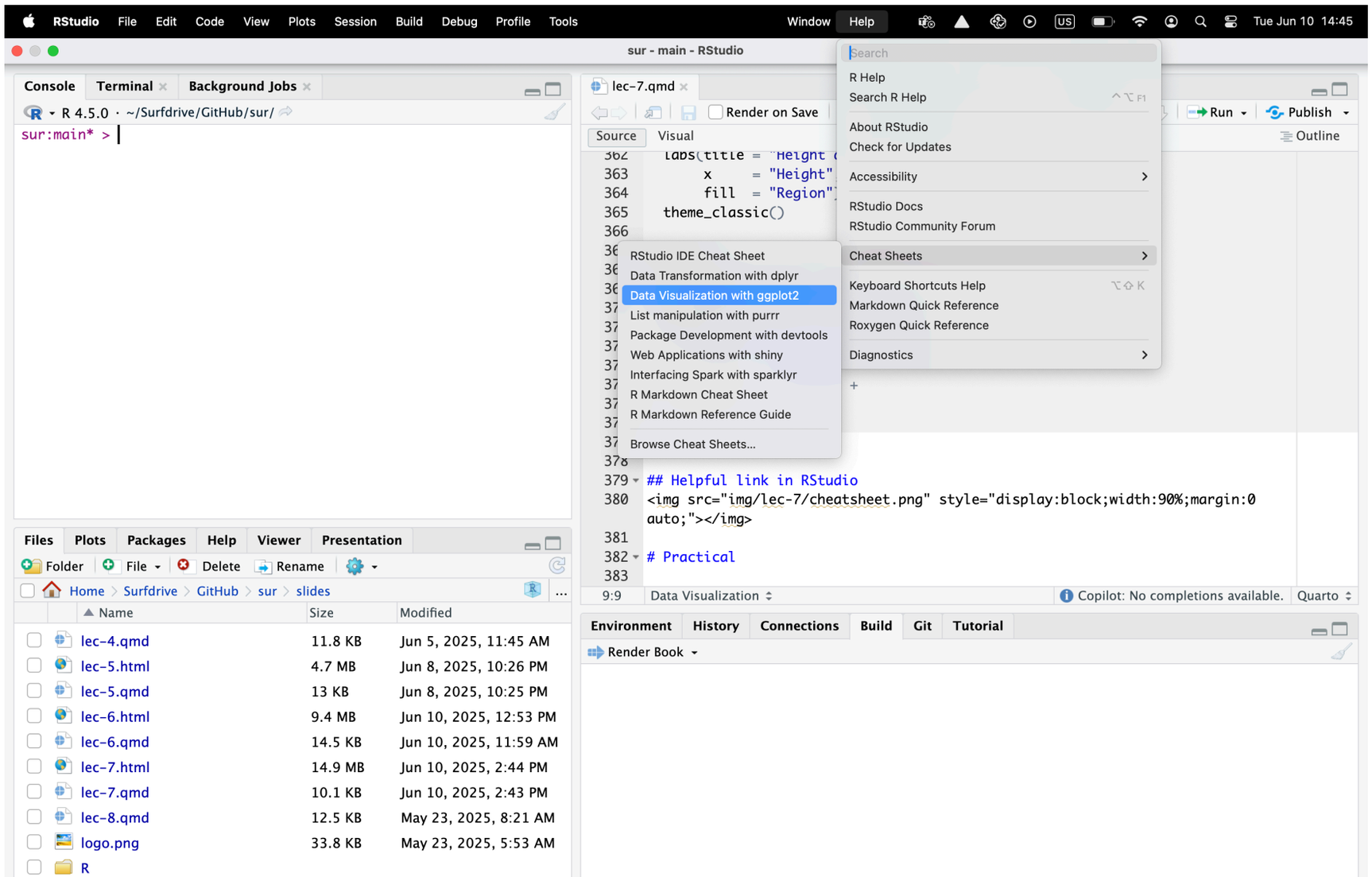
```
library(plotly)
gg <- boys %>%
  filter(!is.na(reg)) %>%

  ggplot(aes(x      = age,
             y      = bmi,
             colour = reg)) +

  geom_point(alpha = 0.5) +

  labs(title  = "BMI trend for boys",
       x      = "Age",
       y      = "BMI",
       colour = "Region") +
  theme_minimal()

ggplotly(gg)
```

INTERACTIVE PLOTS

Use `plotly::ggplotly()` to make any ggplot interactive



HELPFUL LINK IN RSTUDIO



PRACTICAL