

Data Visualisation **geom** Encyclopedia

Thiyanga S. Talagala

2023-08-31

Contents

1	About	7
1.1	Usage	7
1.2	Render book	7
1.3	Preview book	8
2	Hello bookdown	9
2.1	A section	9
3	Graph types	11
3.1	Chapters and sub-chapters	11
4	R Packages	13
4.1	Data processing	13
4.2	Graph arrangement	13
4.3	Data packages	13
4.4	Large cross sectional dataset	14
4.5	Small cross sectional dataset	15
4.6	Time series dataset	15
4.7	Packages related to geom	16
4.8	geom Extensions	16
4.9	All available geom_... in the ggplot2 package	18
5	A: geom_a...	19
5.1	geom_abline	19
5.2	geom_area	19
6	B: geom_b...	23
6.1	geom_bar	23
6.2	geom_bin_2d	26
6.3	geom_bin2d	29
6.4	geom_blank	30
6.5	geom_boxplot	30

7 C: geom_c...	33
7.1 geom_col	33
8 D: geom_d...	35
8.1 geom_delaunay_tile	35
8.2 geom_density	36
8.3 geom_density_2d	36
8.4 geom_density_2d_filled	37
8.5 geom_density_ridges	38
8.6 geom_dl	39
9 E: geom_e...	41
9.1 geom_encircle	41
10 F: geom_f...	43
10.1 geom_function	43
11 A: geom_axxxxxx	45
12 H: geom_h...	47
12.1 geom_histogram	47
12.2 geom_hline	47
13 A: geom_axxxxxx	49
14 A: geom_axxxxxx	51
15 A: geom_axxxxxx	53
16 L: geom_l...	55
16.1 geom_label_repel	55
16.2 geom_lm	56
17 M: geom_m...	57
17.1 geom_mark_circle	57
18 A: geom_axxxxxx	59
19 A: geom_axxxxxx	61
20 P: geom_p...	63
20.1 geom_point	63
21 A: geom_axxxxxx	65
22 A: geom_axxxxxx	67

CONTENTS	5
23 S: geom_s...	69
23.1 geom_signif	69
23.2 geom_smooth	70
24 A: geom_axxxxxx	73
25 A: geom_axxxxxx	75
26 V: geom_v...	77
26.1 geom_vline	77
26.2 geom_violin	78
27 A: geom_axxxxxx	81
28 A: geom_axxxxxx	83
29 A: geom_axxxxxx	85
30 A: geom_axxxxxx	87
31 Others	89
31.1 ggpairs	89

Chapter 1

About

This is a *sample* book written in **Markdown**. You can use anything that Pandoc’s Markdown supports; for example, a math equation $a^2 + b^2 = c^2$.

1.1 Usage

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: `# A good chapter`, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: `## A short section` or `### An even shorter section`.

The `index.Rmd` file is required, and is also your first book chapter. It will be the homepage when you render the book.

1.2 Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select “All formats” if you’d like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you'll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.org/tinytex/>.

1.3 Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in “Preview book”, or from the R console:

```
bookdown::serve_book()
```

Chapter 2

Hello bookdown

All chapters start with a first-level heading followed by your chapter title, like the line above. There should be only one first-level heading (#) per .Rmd file.

2.1 A section

All chapter sections start with a second-level (##) or higher heading followed by your section title, like the sections above and below here. You can have as many as you want within a chapter.

An unnumbered section

Chapters and sections are numbered by default. To un-number a heading, add a {.unnumbered} or the shorter {-} at the end of the heading, like in this section.

Chapter 3

Graph types

3.1 Chapters and sub-chapters

Chapter 4

R Packages

This chapter lists all the packages necessary for plotting

4.1 Data processing

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.3     v purrr   1.0.2
## v tibble  3.2.1     v dplyr   1.1.2
## v tidyrr  1.3.0     v stringr 1.5.0
## v readr   2.1.3     vforcats 1.0.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

4.2 Graph arrangement

```
library(patchwork)
```

4.3 Data packages

4.3.1 Cross sectional data

```
#install_github("thiyanth/elephants")
library(elephants)
```

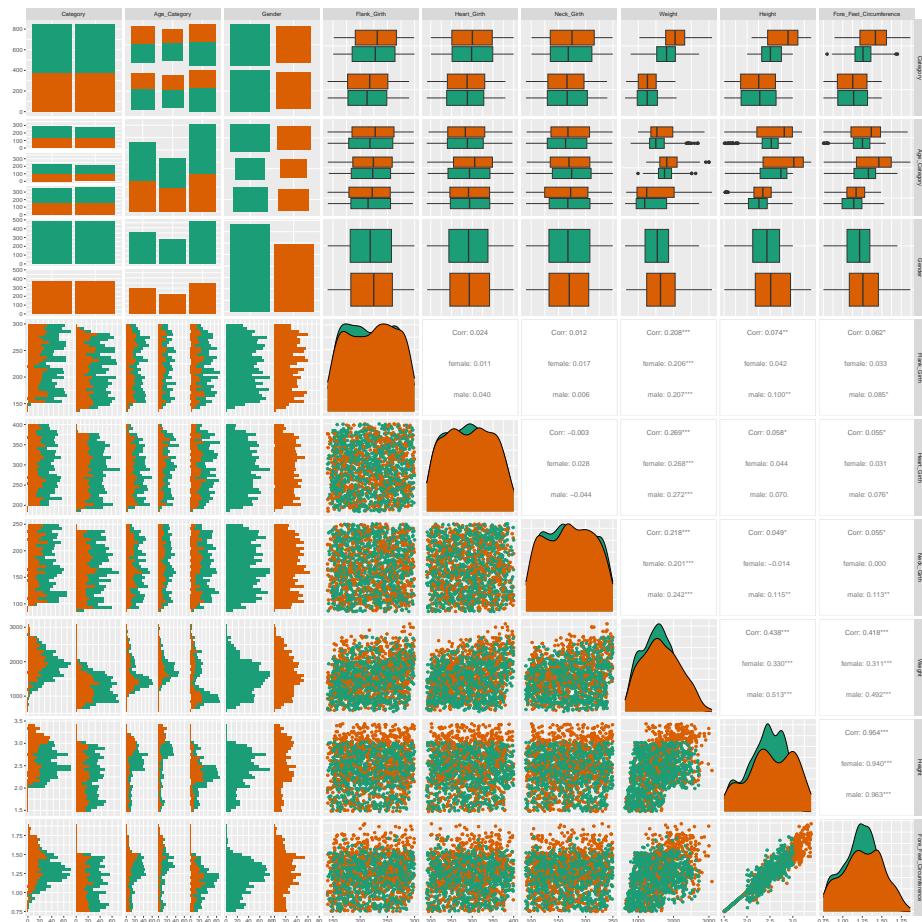
4.3.2 Time series data

```
#install_github("denguedatahub")
library(denguedatahub)
```

4.4 Large cross sectional dataset

```
data("elephants")
```

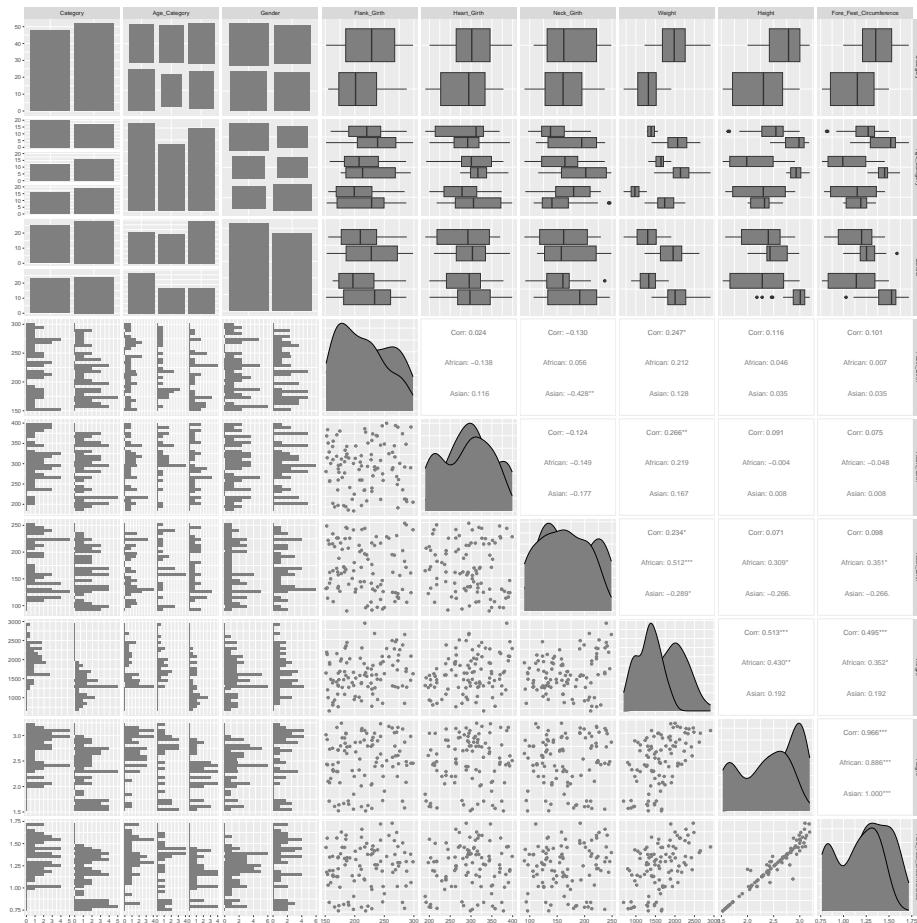
4.4.1 Glimpse of the large dataset



4.5 Small cross sectional dataset

```
set.seed(2023)
elephants_subset.100 <- elephants |> sample_n(100)
```

4.5.1 Glimpse of the small cross sectional dataset



4.6 Time series dataset

```
library(denguedatahub)
srilanka_weekly_data

## # A tibble: 21,934 x 6
##   year week start.date end.date   district cases
```

```

## * <dbl> <dbl> <date>      <date>      <chr>       <dbl>
## 1 2006     52 2006-12-23 2006-12-29 Colombo      71
## 2 2006     52 2006-12-23 2006-12-29 Gampaha      12
## 3 2006     52 2006-12-23 2006-12-29 Kalutara      12
## 4 2006     52 2006-12-23 2006-12-29 Kandy        20
## 5 2006     52 2006-12-23 2006-12-29 Matale         4
## 6 2006     52 2006-12-23 2006-12-29 NuwaraEliya    1
## 7 2006     52 2006-12-23 2006-12-29 Galle          1
## 8 2006     52 2006-12-23 2006-12-29 Hambanthota    1
## 9 2006     52 2006-12-23 2006-12-29 Matara        11
## 10 2006    52 2006-12-23 2006-12-29 Jaffna         0
## # i 21,924 more rows

```

```

library(ggplot2)
#devtools::install_github("EvaMaeRey/ggxmean")
#library(gxmean)
#devtools::install_github("davidsjoberg/ggsankey")
library(ggxmean)

```

```
library(GGally) # Matrix plots
```

4.8 geom Extensions

4.9 All available geom_... in the ggplot2 package

x
geom_abline
geom_area
geom_bar
geom_bin_2d
geom_bin2d
geom_blank
geom_boxplot
geom_col
geom_contour
geom_contour_filled
geom_count
geom_crossbar
geom_curve
geom_density
geom_density_2d
geom_density_2d_filled
geom_density2d
geom_density2d_filled
geom_dotplot
geom_errorbar
geom_errorbarh
geom_freqpoly
geom_function
geom_hex
geom_histogram
geom_hline
geom_jitter
geom_label
geom_line
geom_linerange
geom_map
geom_path
geom_point
geom_pointrange
geom_polygon
geom_qq
geom_qq_line
geom_quantile
geom_raster
geom_rect
geom_ribbon
geom_rug
geom_segment
geom_sf
geom_sf_label
geom_sf_text
geom_smooth
geom_spoke
geom_step
geom_text
geom_tile

Chapter 5

A: geom_abline

Package: ggplot2 [R-ggplot2]

Book:

Description: Draw a straight line ($Y = mX + c$) for a given slope (m) and intercept (c).

See also: geom_point, geom_vline, geom_hline

Example:

```
abline <- ggplot(elephants_subset.100, aes(y = Height, x=Fore_Feet_Circumference)) + geom_abline()
  labs(title="A: `geom_abline` only") +
  theme(aspect.ratio = 1)

pointabline <- ggplot(elephants_subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_point() +
  geom_abline(intercept = 0.15, slope = 1.9) +
  labs(title="B: `geom_point` + `geom_abline` both") +
  theme(aspect.ratio = 1)

library(patchwork)
abline | pointabline
```

5.2 geom_area

Package: ggplot2 [R-ggplot2]

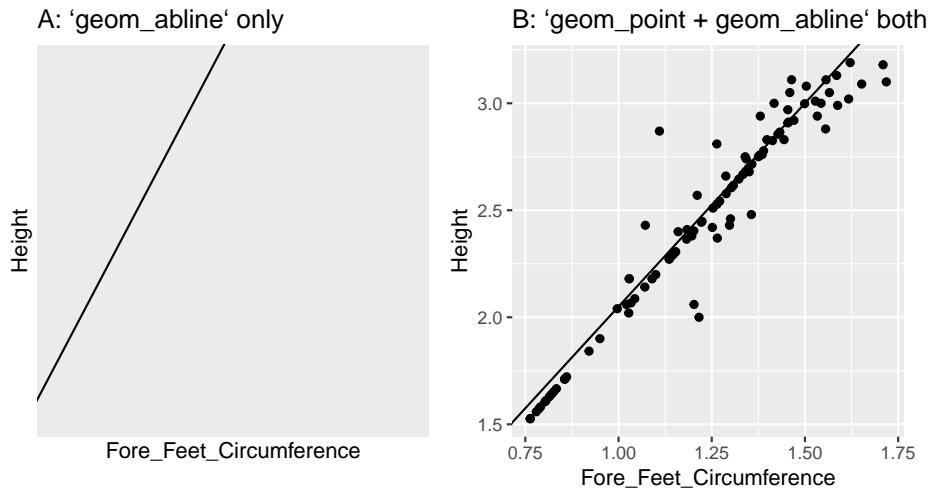


Figure 5.1: Illustration of (A) geom_abline and (B) use of geom_point and geom_abline both

Description: Create an area plot. This cover the space between x-axis and line that connects the data points.

See also: geom_line

```
colombo <- srilanka_weekly_data |>
  filter(district == "Colombo")
ggplot(data=colombo, aes(x=start.date, y=cases)) +
  geom_area()
```

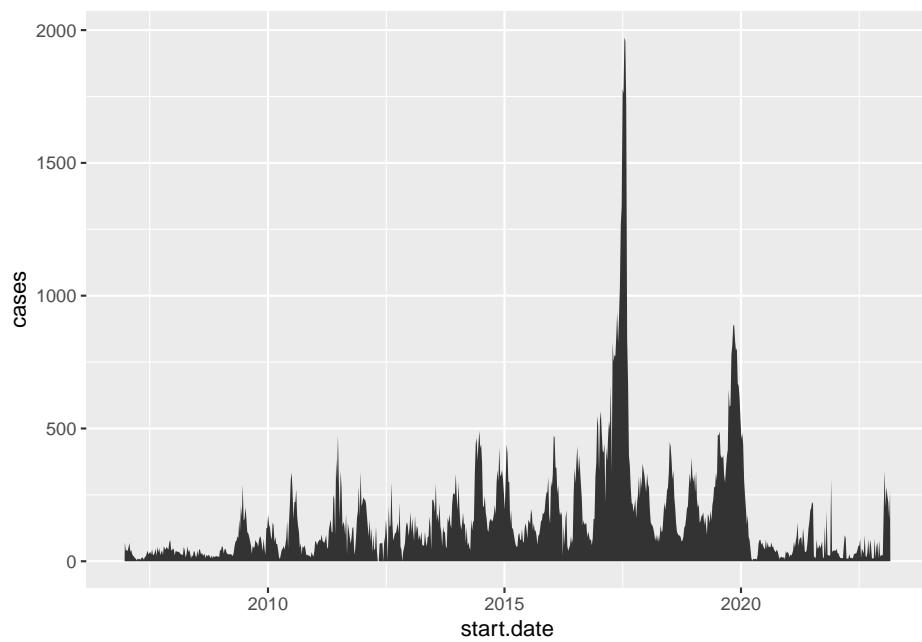


Figure 5.2: Illustration of geom_area

Chapter 6

B: geom_bar...

6.1 geom_bar

Package: ggplot2 [R-ggplot2]

Description: Draw a bar proportional to the specified number. For example, number of cases or user defined number.

Statistics layer(s):

stat_count - This is the default statistics layer. It counts number of cases in each group.

stat_identify - It plots the data as it is.

See also: geom_col

6.1.1 With count

```
ggplot(elephants, aes(y = Age_Category)) +  
  geom_bar()  
  
ggplot(elephants, aes(y = Age_Category, fill=Category)) +  
  geom_bar()  
  
ggplot(elephants, aes(y=Age_Category, fill = Category)) +  
  geom_bar(position = "dodge")  
  
ggplot(elephants, aes(y=Age_Category, fill = Category)) +  
  geom_bar(position = "dodge")
```

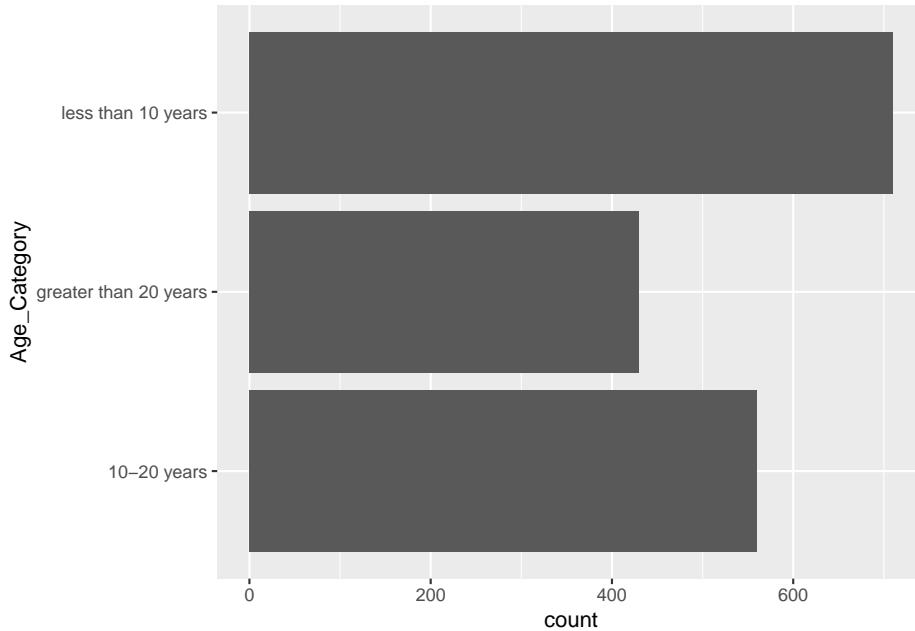


Figure 6.1: Illustration of geom_bar to create a bar chart

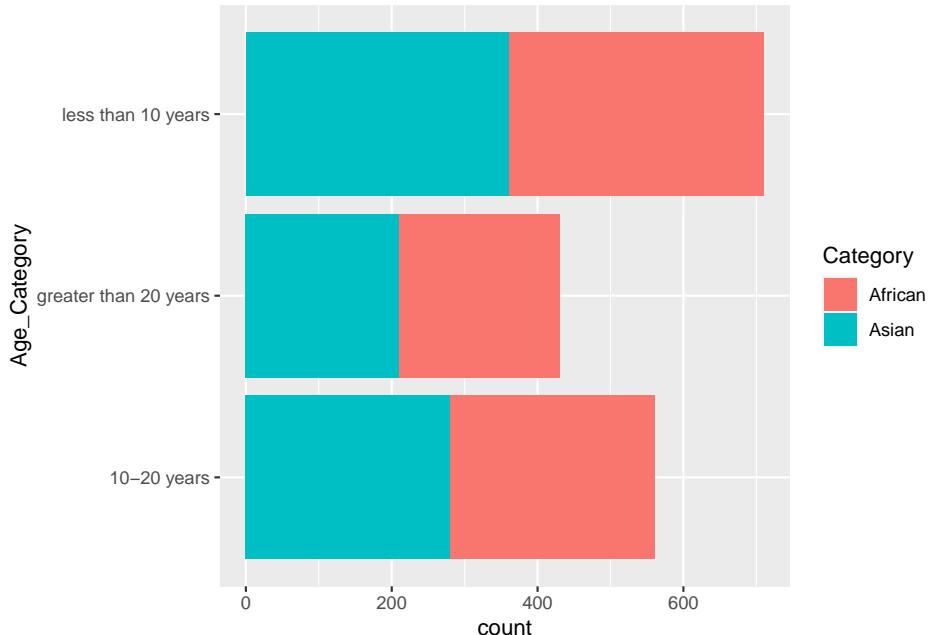


Figure 6.2: Illustration of geom_bar to create a stacked bar chart

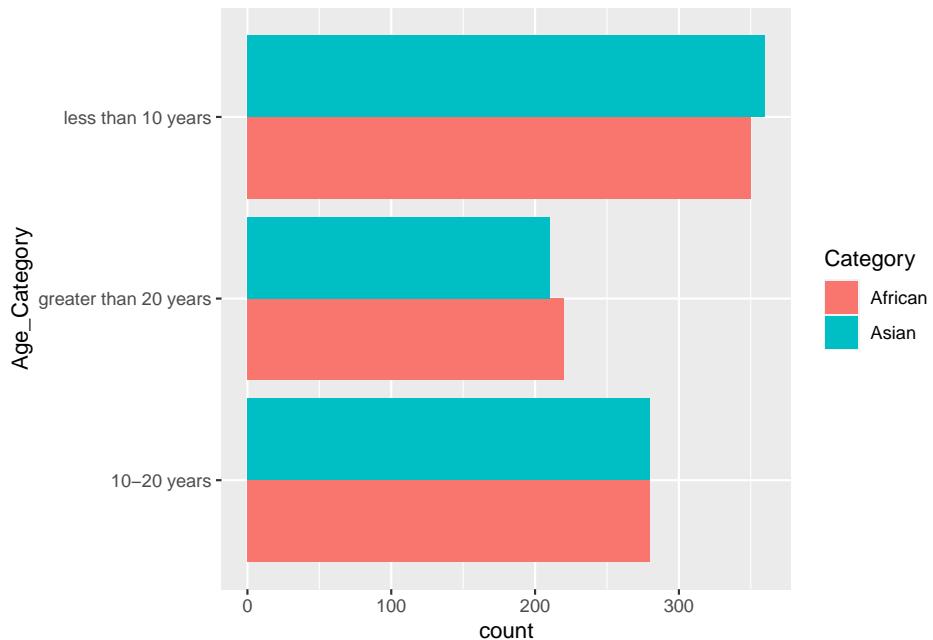


Figure 6.3: Illustration of geom_bar to create a cluster bar chart

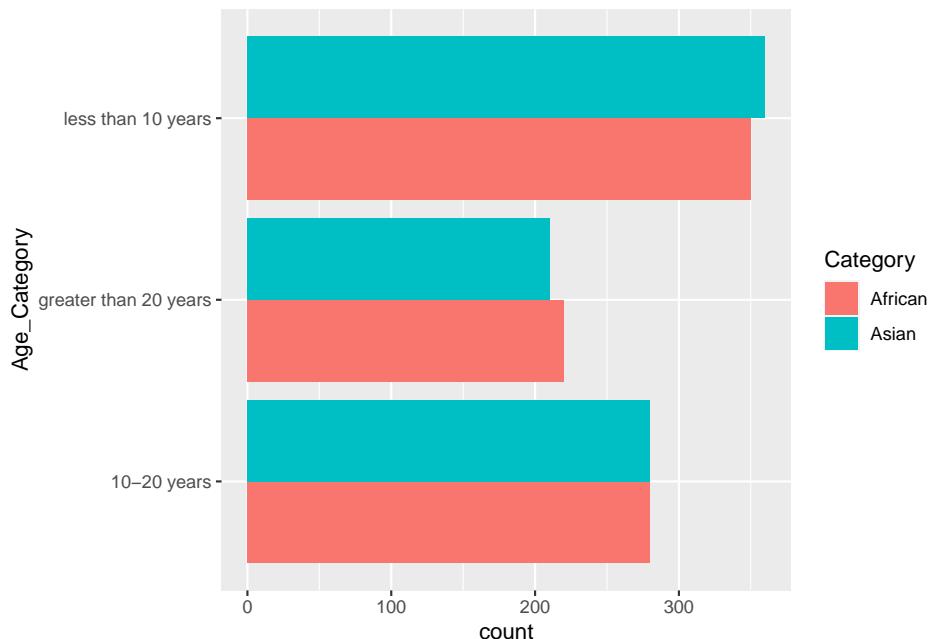
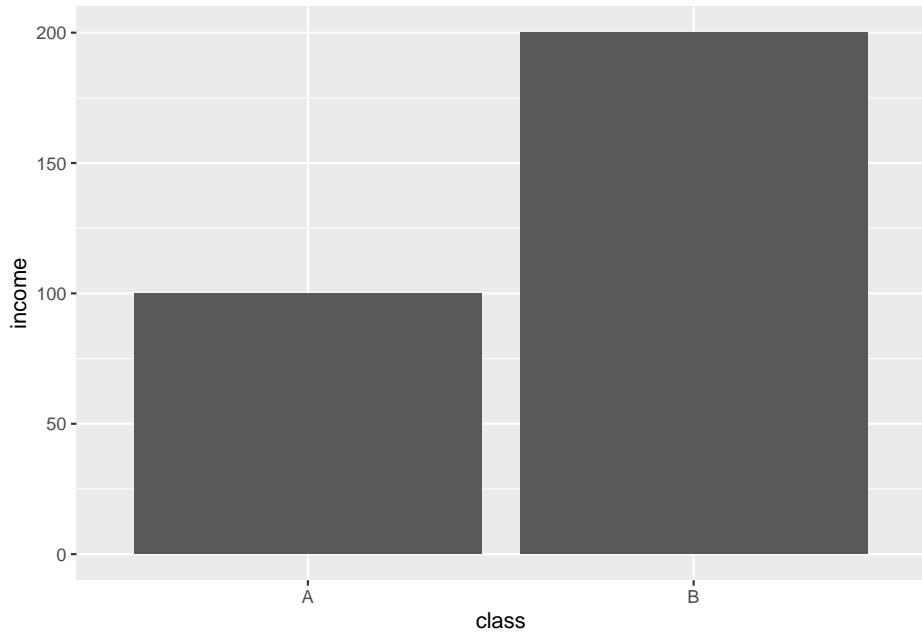


Figure 6.4: Illustration of geom_bar to create a cluster bar chart

6.1.2 With identity

```
dfbar <- data.frame(class=c("A", "B"), income = c(100, 200))
ggplot(dfbar, aes(class, income)) +
  geom_bar(stat="identity")
```



6.2 geom_bin_2d

Package: ggplot2 [R-ggplot2]

Description: Divides the Cartesian plane created by x-variable and y-variable into rectangles, counts the number of observations in each rectangle. Only the observations with rectangles are filled according to the number of observations.

Understandable aesthetics: x, y, fill, group

Statistics layer(s):

See also: geom_bin2d, geom_point

```
ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_bin_2d() +
  theme(aspect.ratio = 1)
```

```
ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_bin_2d(bins=20) +
  theme(aspect.ratio = 1)
```

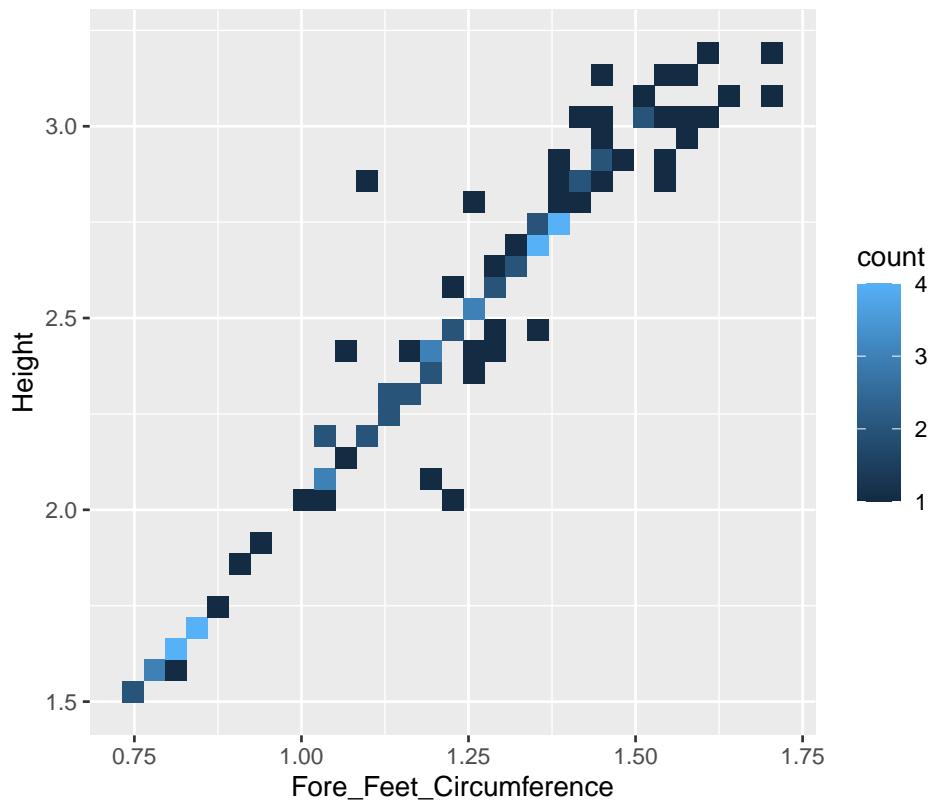


Figure 6.5: Illustration of using geom_bin_2d

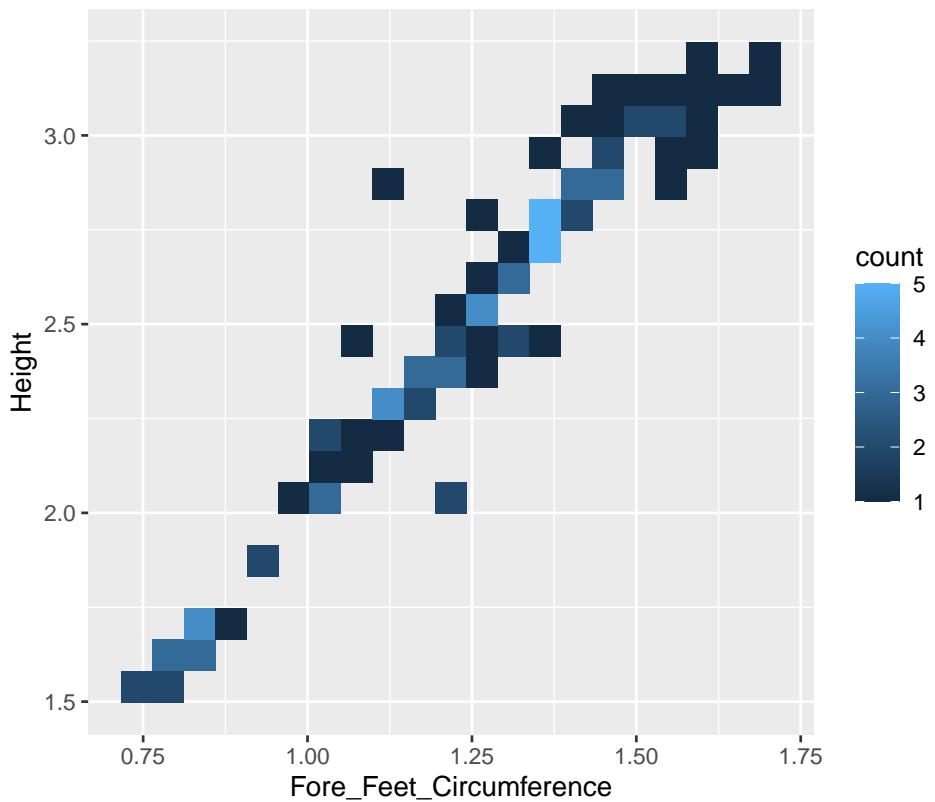


Figure 6.6: Illustration of changing bins in geom_bin_2d

6.3 geom_bin2d

Package: ggplot2 [R-ggplot2]

Description: Divides the Cartesian plane created by x-variable and y-variable into rectangles, counts the number of observations in each rectangle. Only the observations with rectangles are filled according to the number of observations.

Understandable aesthetics: x, y, fill, group

Statistics layer(s):

See also: geom_bin_2d, geom_point

Example:

```
ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_bin2d() +
  theme(aspect.ratio = 1)
```

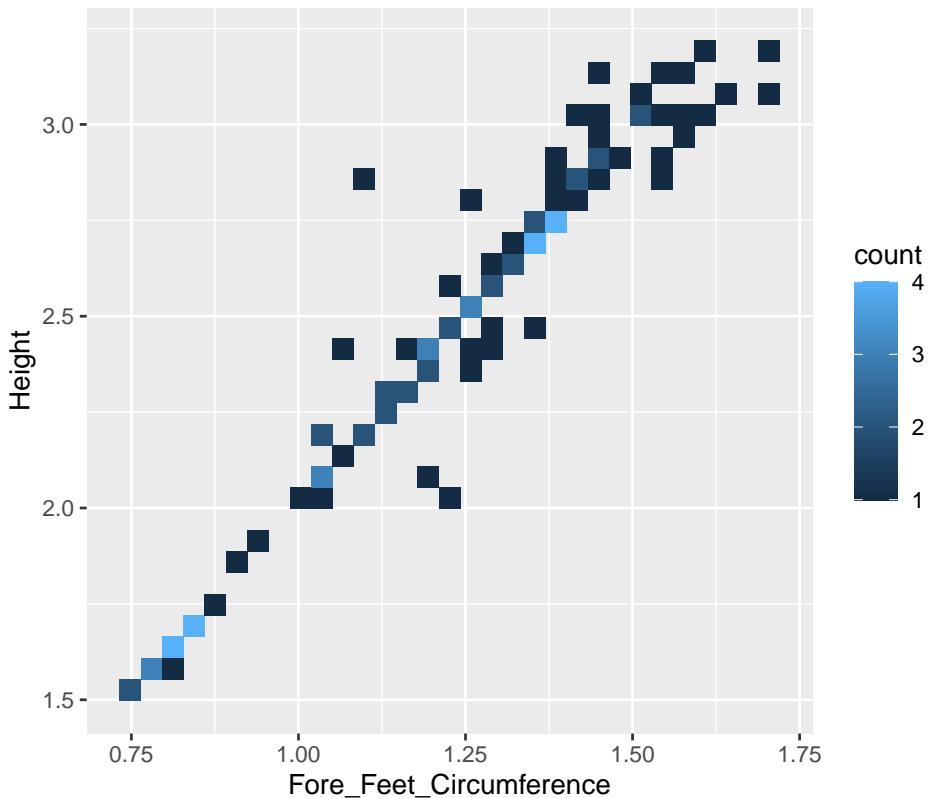


Figure 6.7: Illustration of using geom_bin_2d

6.4 geom_blank

Package: ggplot2 [R-ggplot2]

Description: Draws nothing.

6.5 geom_boxplot

Package: ggplot2 [R-ggplot2]

Description: Draw a bar proportional to the specified number. For example, number of cases or user defined number.

Statistics layer(s):

stat_boxplot - This the default statistics layer. This computes minimum, maximum, median, first quartile (Q_1), third quartile (Q_3), upper whisker extends up to $Q_3 + 1.5 \times IQR$ and lower whisker extends up to $Q_1 - 1.5 \times IQR$, where $IQR = Q_3 - Q_1$. In a notched box plot, it creates 95% confidence interval for mean.

See also: geom_col

```
ggplot(elephants, aes(y = Weight)) +
  geom_boxplot()
```

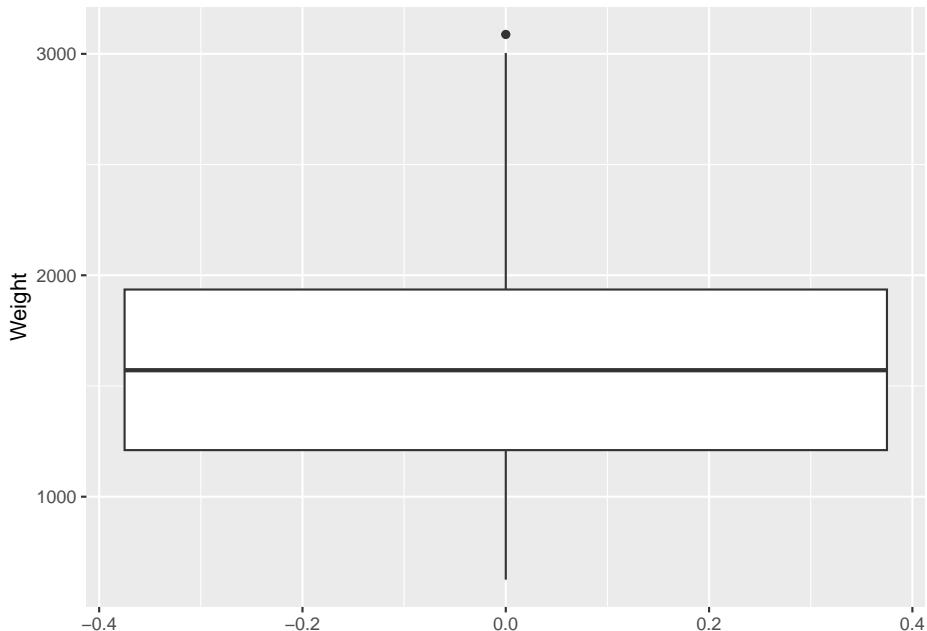


Figure 6.8: Illustration of using geom_boxplot

```
ggplot(elephants, aes(y = Weight)) +  
  geom_boxplot(outlier.colour="black", outlier.shape=16,  
               outlier.size=2, notch=TRUE)
```

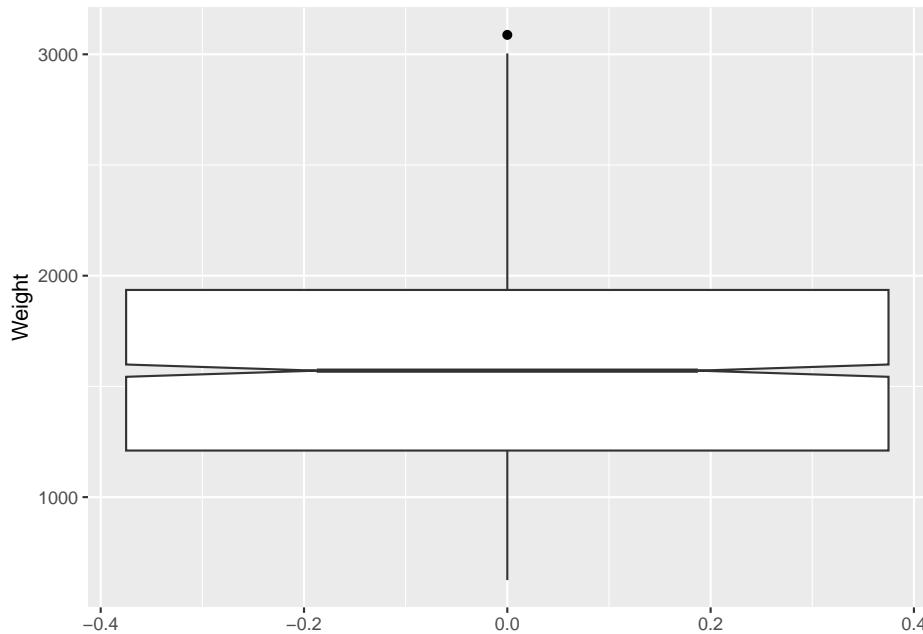


Figure 6.9: Illustration of using `geom_boxplot` with changing outliers and adding a notch to create notched box plot.

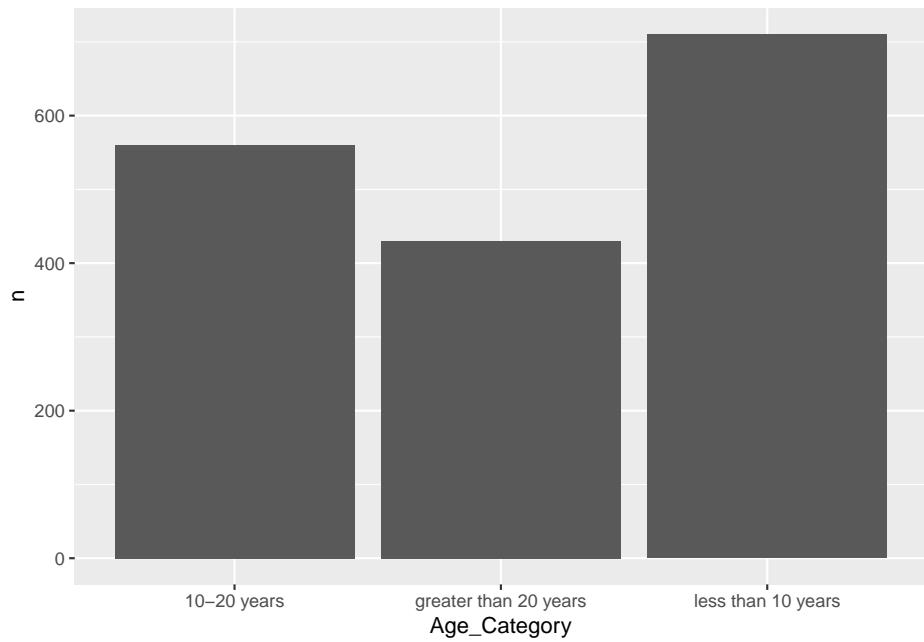
Chapter 7

C: geom_col...

7.1 geom_col

Before using `geom_col`, you need to create a summary table of counts or you can apply `geom_col` for a summary table already given.

```
## # A tibble: 3 x 2
##   Age_Category      n
##   <chr>              <int>
## 1 10-20 years        560
## 2 greater than 20 years  430
## 3 less than 10 years  710
```



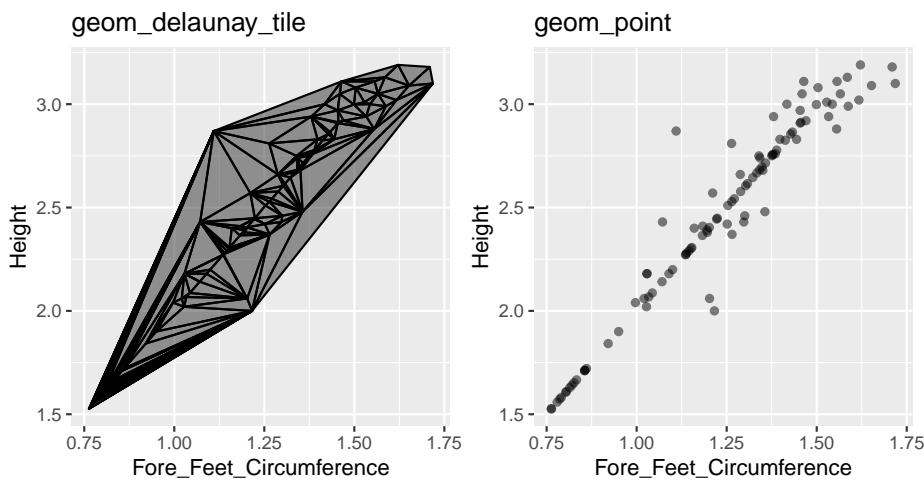
Chapter 8

D: geom_d...

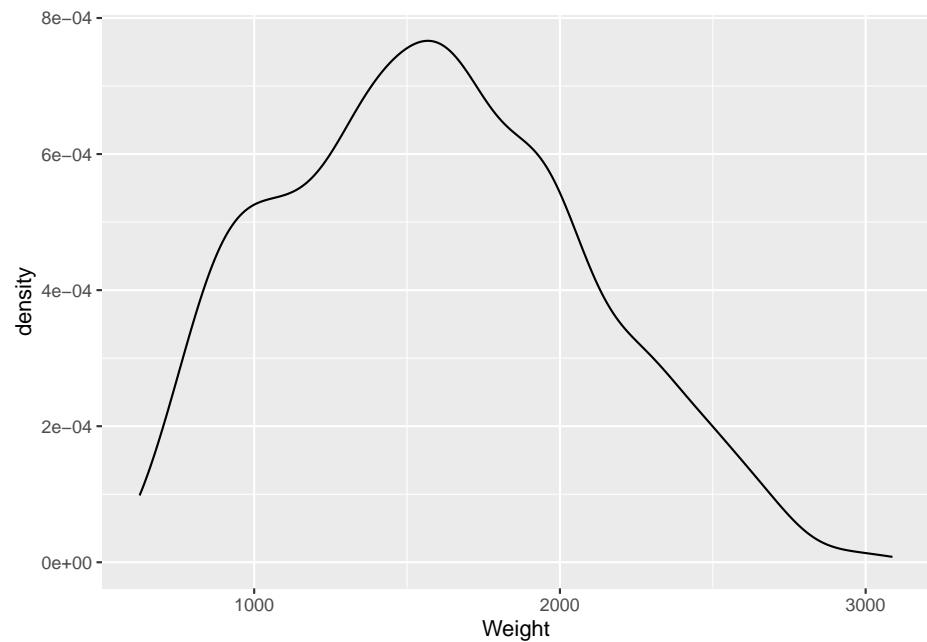
8.1 geom_delaunay_tile

Description: The Delaunay triangulation is used to create a planar graph.

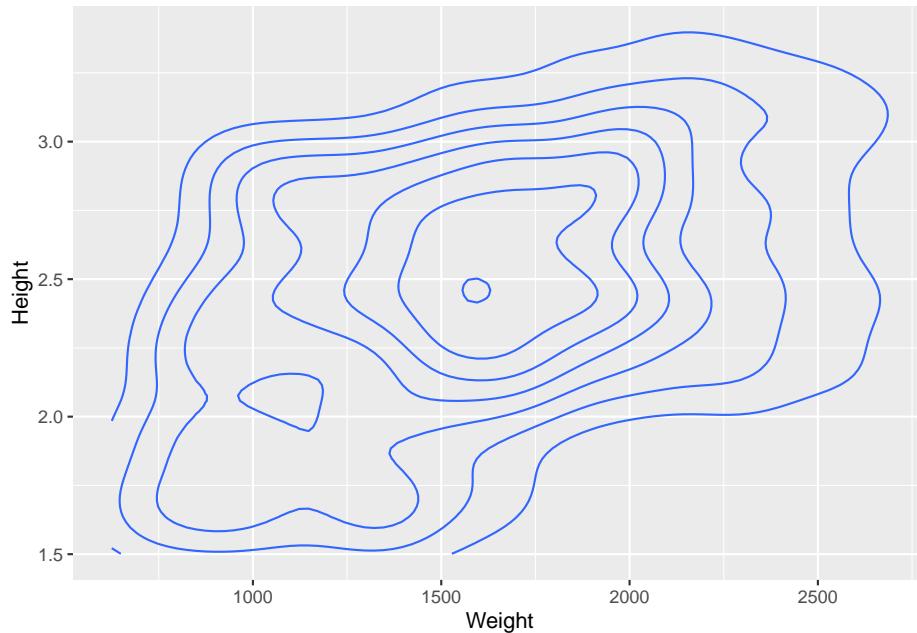
```
delaunay <- ggplot(elephants_subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +  
  ggforce::geom_delaunay_tile(alpha = 0.5, colour = 'black') + labs(title="geom_delaunay_tile") +  
  
# to compare with geom_point  
point <- ggplot(elephants_subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +  
  geom_point(alpha = 0.5, colour = 'black') + labs(title="geom_point") + theme(aspect.ratio = 1)  
  
library(patchwork)  
delaunay|point
```



```
ggplot(elephants, aes(x = Weight)) +  
  geom_density()  
⑧. ↵  geom_adensity
```

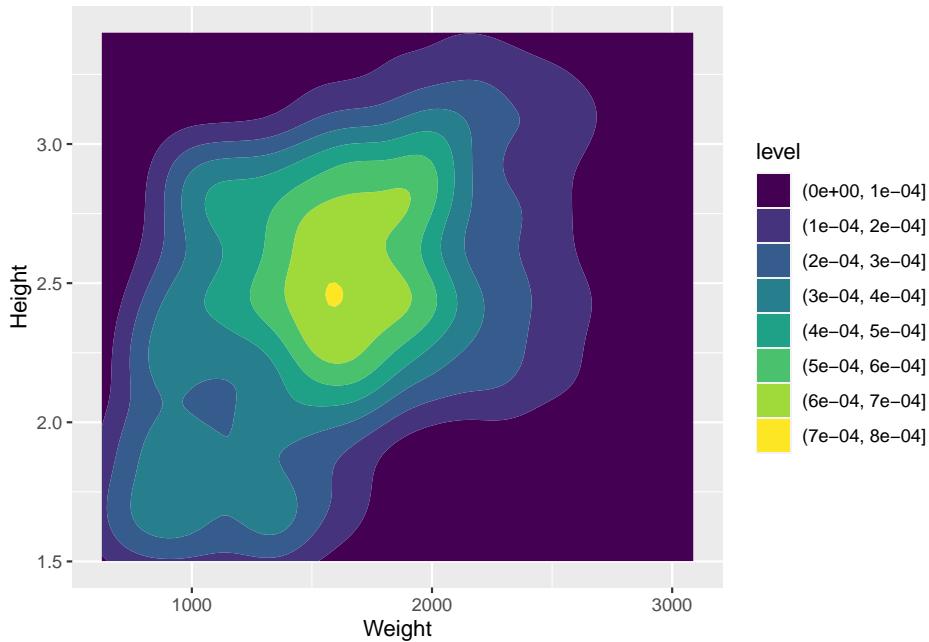


```
ggplot(elephants, aes(y = Height, x=Weight)) +  
  geom_density_2d()  
⑧. ↵  geom_density_2d
```



```
ggplot(elephants, aes(y = Height, x=Weight)) +  
  geom_density_2d_filled()
```

8.4 geom_density_2d_filled

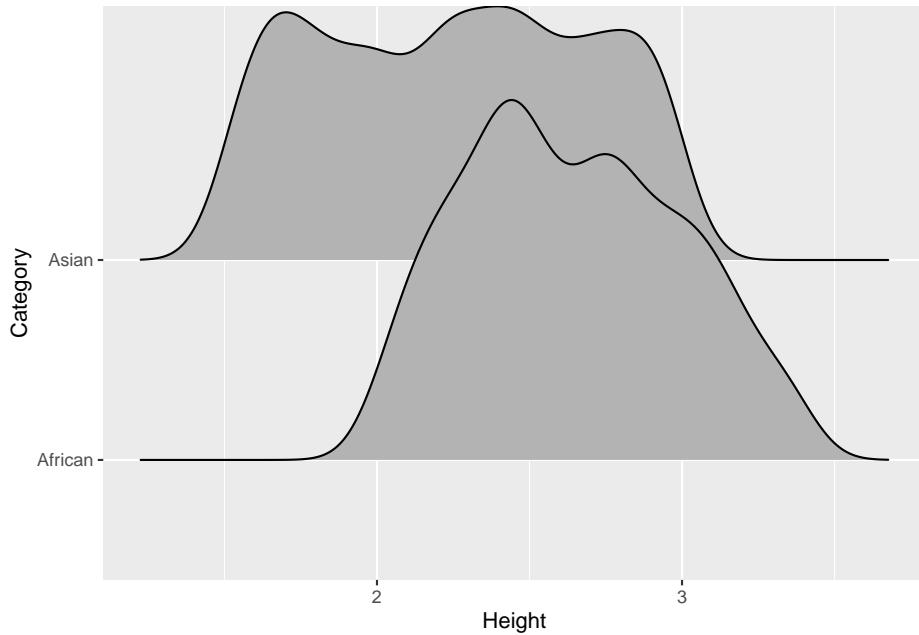


8.5 geom_density_ridges

Here the `y` variable should be qualitative and the `x` variable should be quantitative.

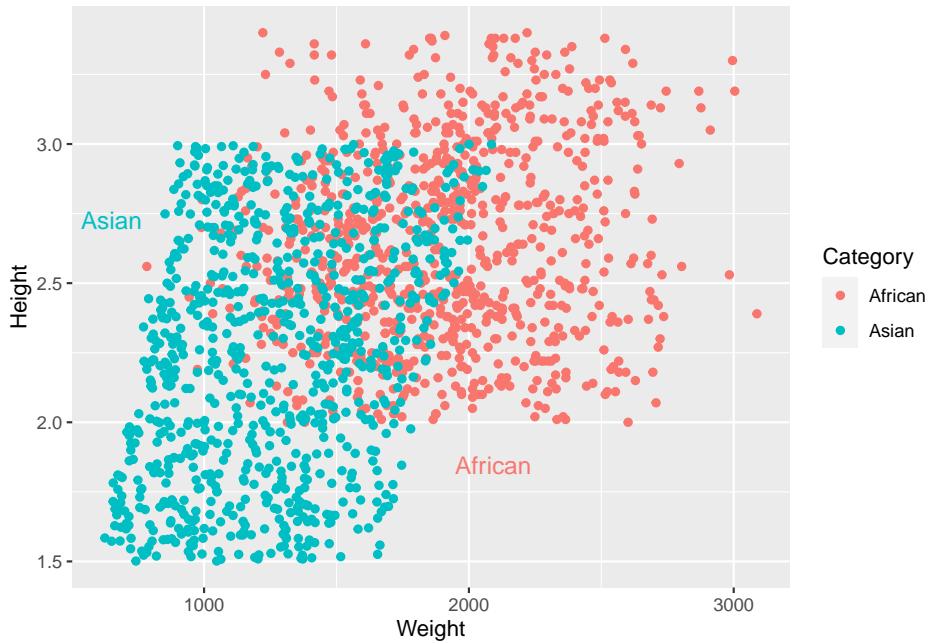
```
library(ggridges)
ggplot(elephants, aes(y=Category, x = Height)) +
  geom_density_ridges()
```

```
## Picking joint bandwidth of 0.092
```



```
library(directlabels)
ggplot(elephants, aes(y = Height, x=Weight, col=Category)) +
  geom_point() +
  geom_dl(aes(label=Category), method="smart.grid")
```

8.6 geom_dl



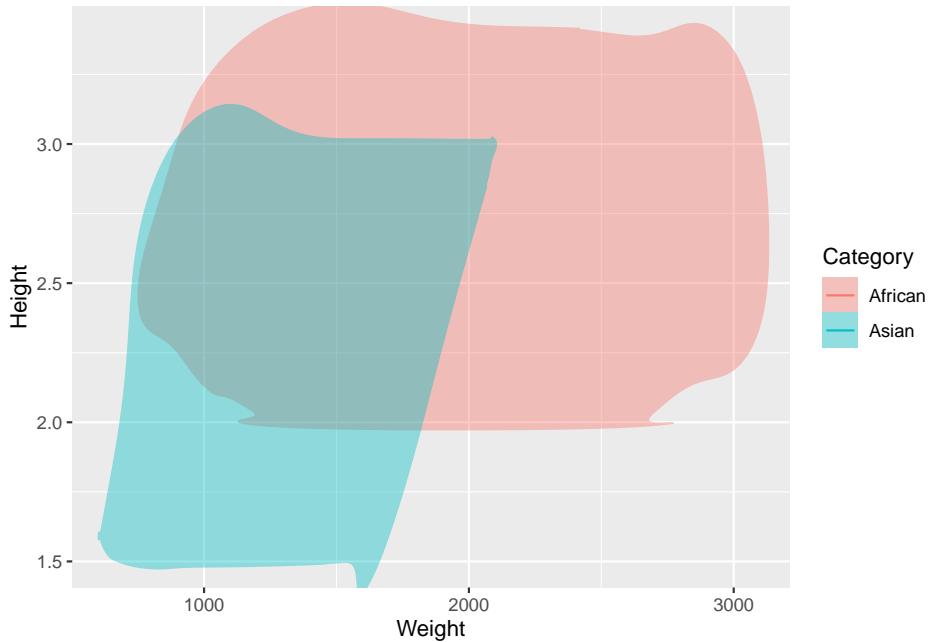
Chapter 9

E: geom_e...

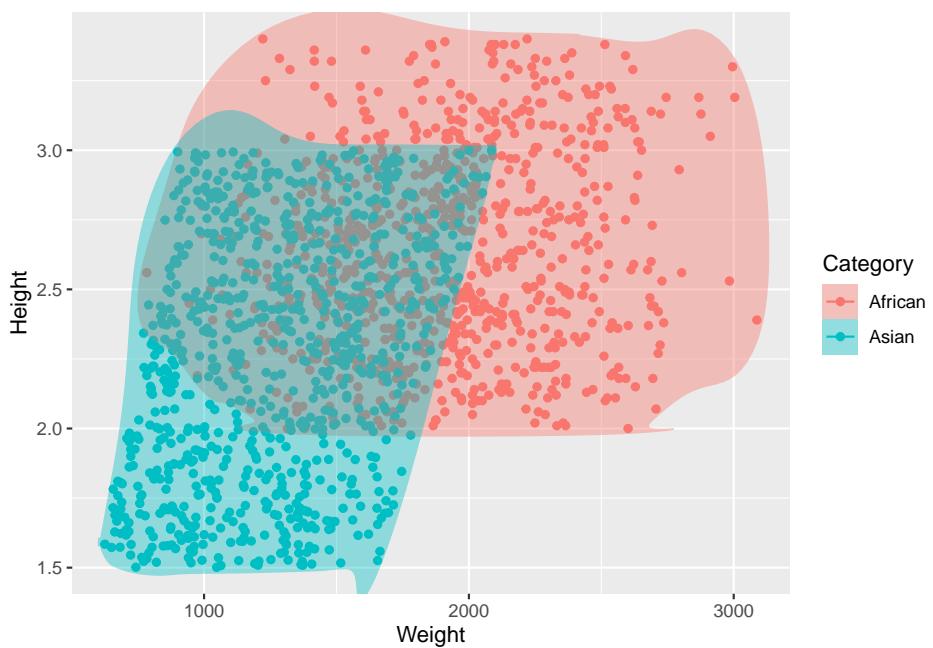
9.1 geom_encircle

Other related geoms: geom_mark_circle

```
## Registered S3 methods overwritten by 'ggalt':  
##   method           from  
##   grid.draw.absoluteGrob  ggplot2  
##   grobHeight.absoluteGrob ggplot2  
##   grobWidth.absoluteGrob ggplot2  
##   grobX.absoluteGrob     ggplot2  
##   grobY.absoluteGrob     ggplot2
```



`geom_encircle with geom_point`

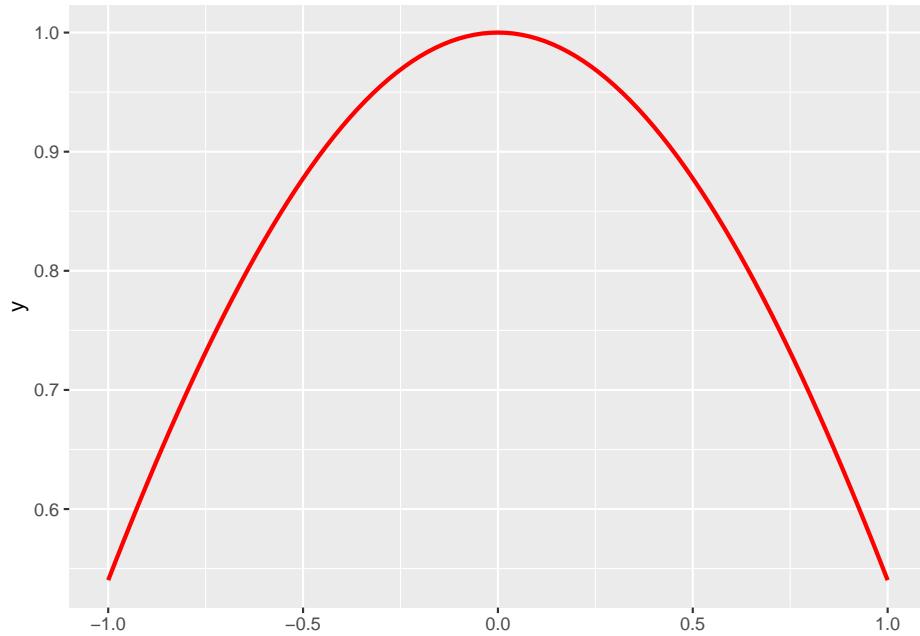


Chapter 10

F: geom_f...

10.1 geom_function

```
ggplot() + xlim(c(-1,1)) + geom_function(fun=cos, colour="red", lwd=1, linetype=1)
```



Chapter 11

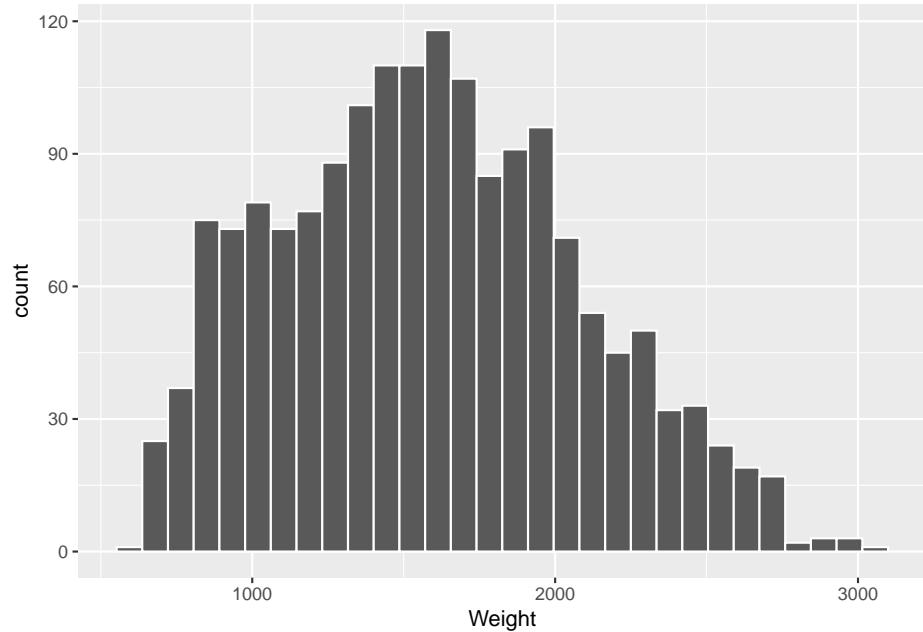
A: geom_axxxxx

Chapter 12

H: geom_h...

12.1 geom_histogram

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



12.2 geom_hline

Package: ggplot2 [R-ggplot2]

Book:

Description: Draw a horizontal line ($Y = c$) for a given value of c , which is known as `yintercept`.

See also: `geom_point`, `geom_vline`, `geom_hline`

Example:

```
hline <- ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) + geom_hline(yintercept = 2.5) + theme(aspect.ratio = 1)

pointhline <- ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_point() +
  geom_hline(yintercept = 2.5) +
  labs(title="B: `geom_point + geom_hline` both") +
  theme(aspect.ratio = 1)

hline | pointhline
```

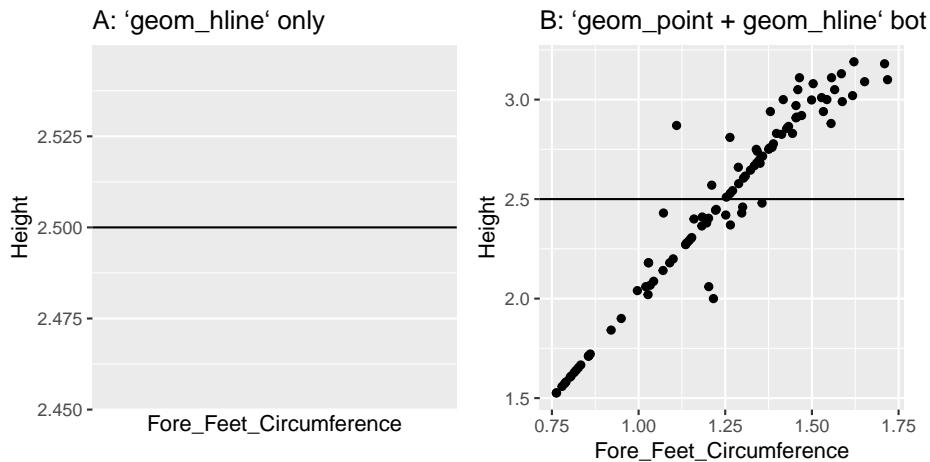


Figure 12.1: Illustration of (A) `geom_hline` and (B) use of `geom_point` and `geom_hline` both

Chapter 13

A: geom_axxxxx

Chapter 14

A: geom_axxxxx

Chapter 15

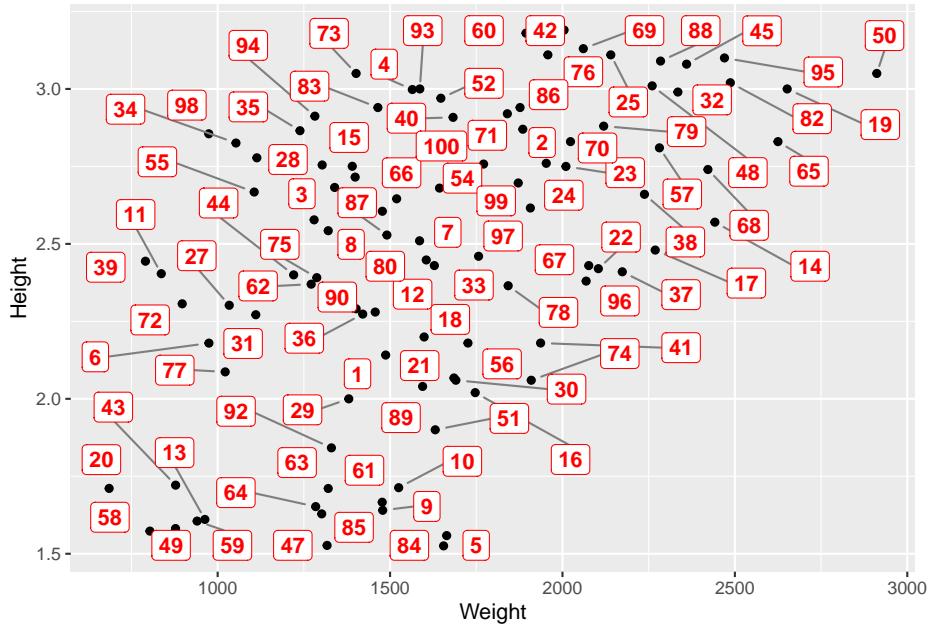
A: geom_axxxxx

Chapter 16

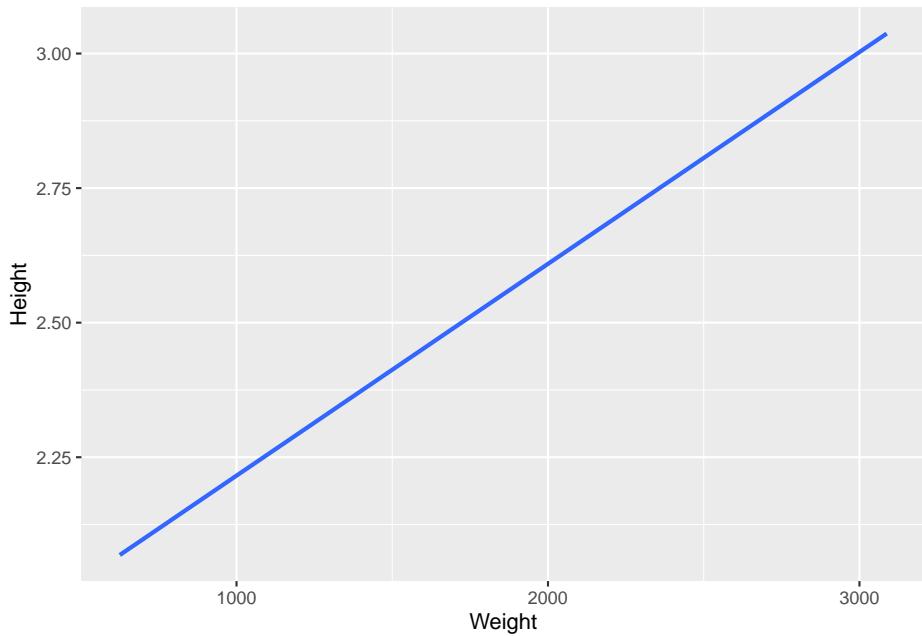
L: geom_l...

```
ggplot(elephants.subset.100, aes(y = Height, x=Weight)) +  
  geom_point() +  
  ggrepel::geom_label_repel(aes(y = Height, x=Weight,  
                                 label = rownames(elephants.subset.100)),  
                           fontface = 'bold', color = 'red',  
                           box.padding = unit(0.40, "lines"),  
                           point.padding = unit(0.6, "lines"),  
                           segment.color = 'grey50'  
)
```

```
## Warning: ggrepel: 5 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps
```



16.2 geom_lm

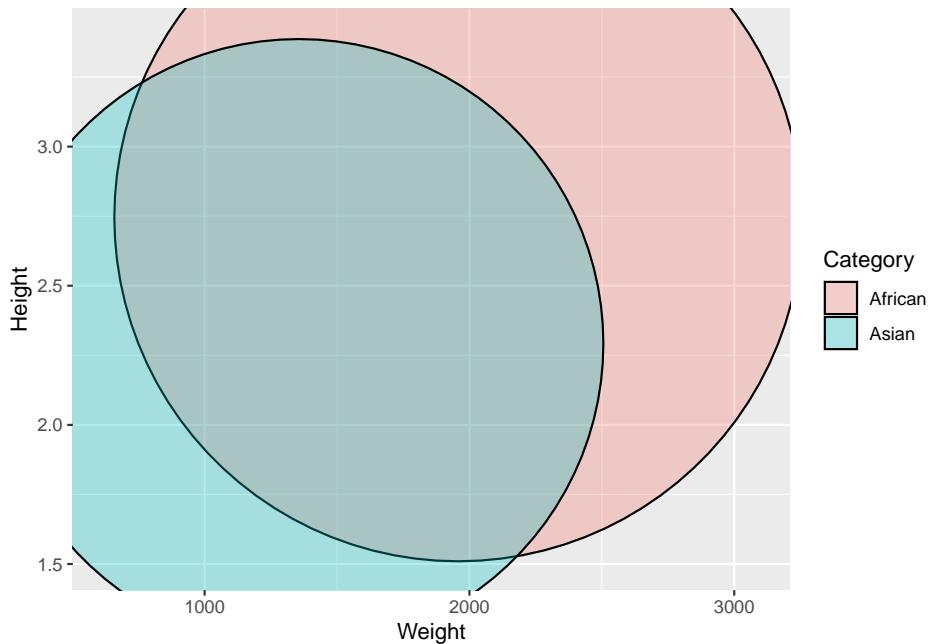


Chapter 17

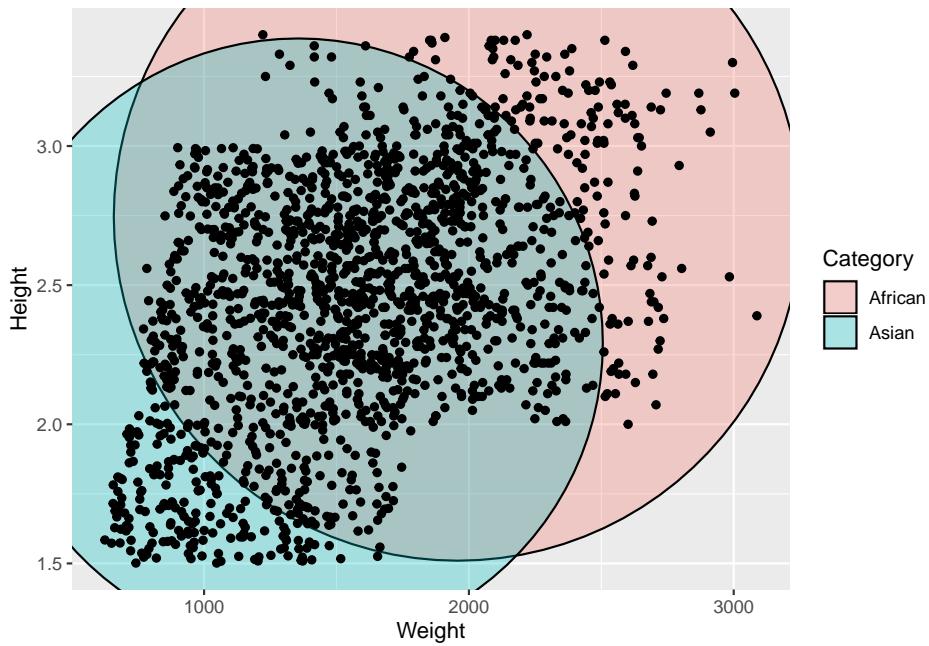
M: geom_mark...

17.1 geom_mark_circle

```
## Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2 3.4.0.  
## i Please use `linewidth` in the `default_aes` field and elsewhere instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```



With geom_point



Chapter 18

A: geom_axxxxx

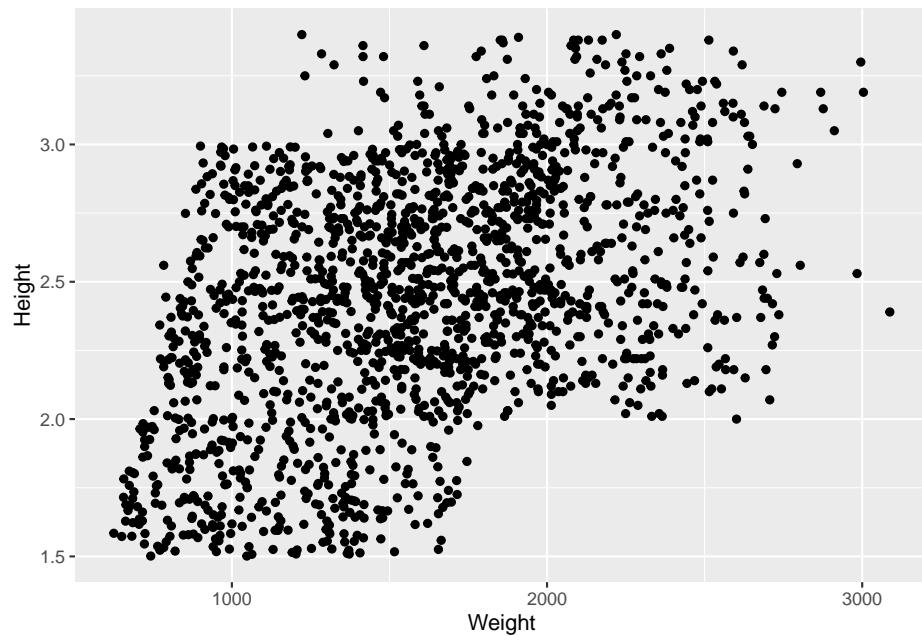
Chapter 19

A: geom_axxxxx

Chapter 20

P: geom_point

20.1 geom_point



Chapter 21

A: geom_axxxxx

Chapter 22

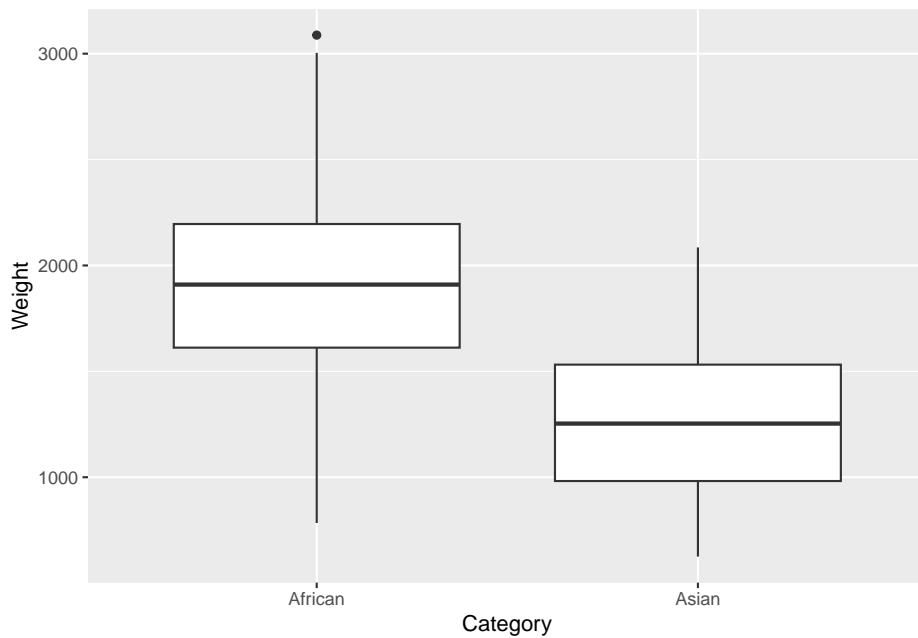
A: geom_axxxxx

Chapter 23

S: geom_s...

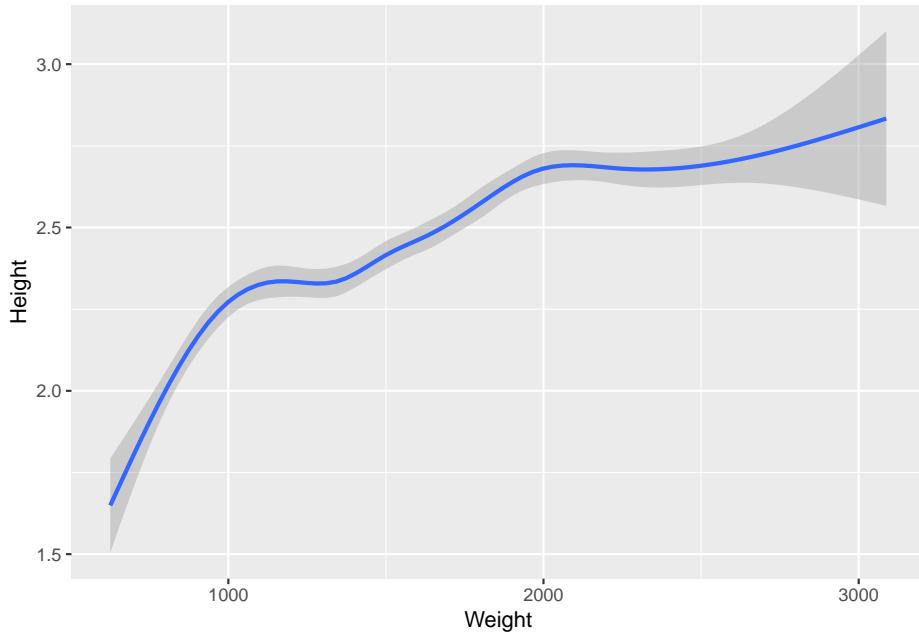
23.1 geom_signif

```
ggplot(elephants, aes(y = Weight, x=Category)) + geom_boxplot() + ggsignif::geom_signif()  
  
## Warning: Computation failed in `stat_signif()`  
## Caused by error in `$<- .data.frame`:  
## ! replacement has 1 row, data has 0
```

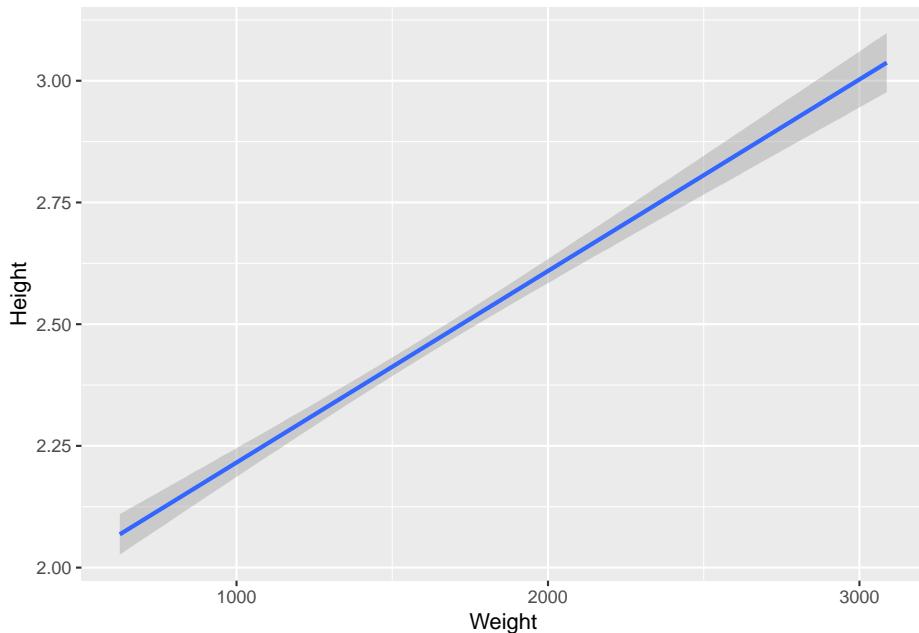


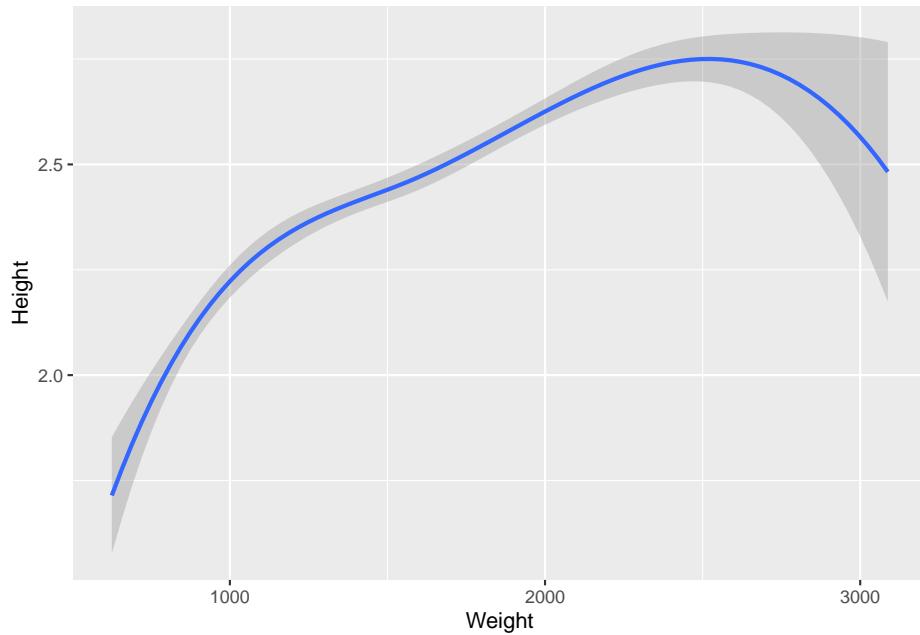
23.2 geom_smooth

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
## `geom_smooth()` using formula = 'y ~ x'
```





Chapter 24

A: geom_axxxxx

Chapter 25

A: geom_axxxxx

Chapter 26

V: geom_v...

26.1 geom_vline

Package: ggplot2 [R-ggplot2]

Book:

Description: Draw a vertical line ($X = c$) for a given value of c , which is known as `xintercept`.

See also: `geom_point`, `geom_vline`, `geom_hline`

Example:

```
vline <- ggplot(elephants_subset.100, aes(y = Height, x=Fore_Feet_Circumference)) + geom_vline(xintercept = 1.25)

pointvline <- ggplot(elephants_subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_point() +
  geom_vline(xintercept = 1.25) +
  labs(title="B: `geom_point + geom_vline` both") +
  theme(aspect.ratio = 1)

library(patchwork)
vline | pointvline
```

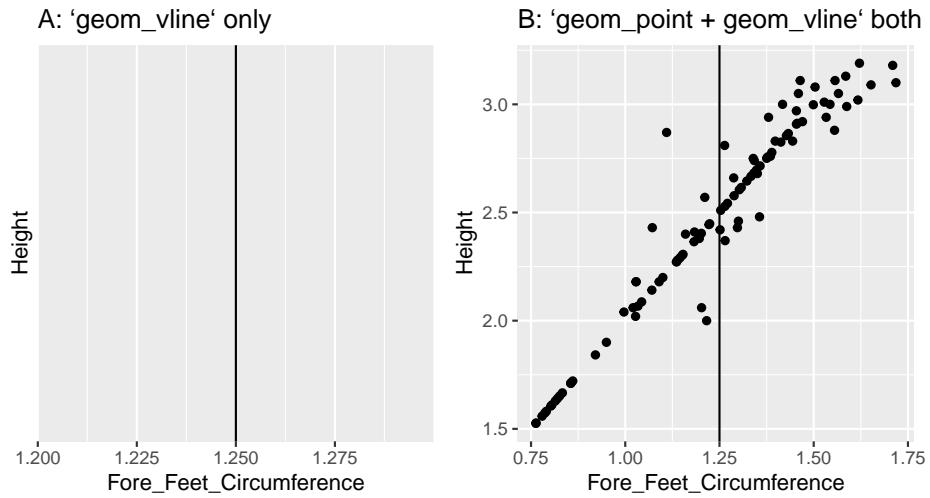
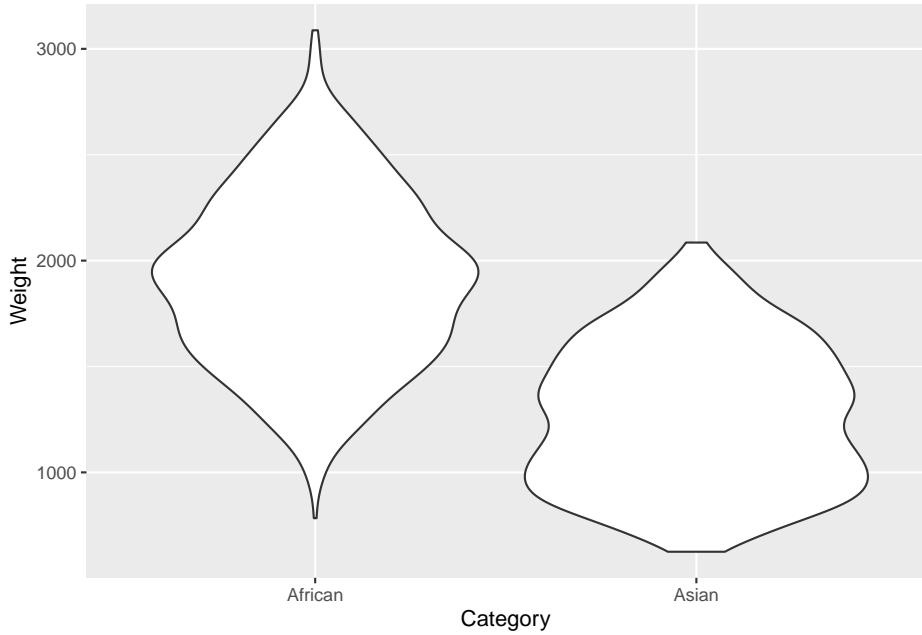
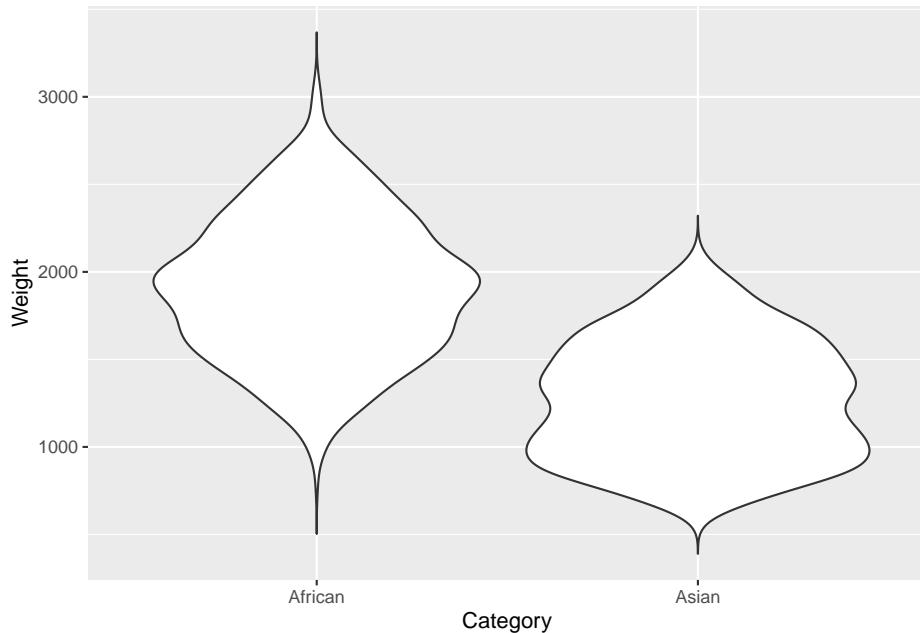


Figure 26.1: Illustration of (A) `geom_vline` and (B) use of `geom_point` and `geom_vline` both

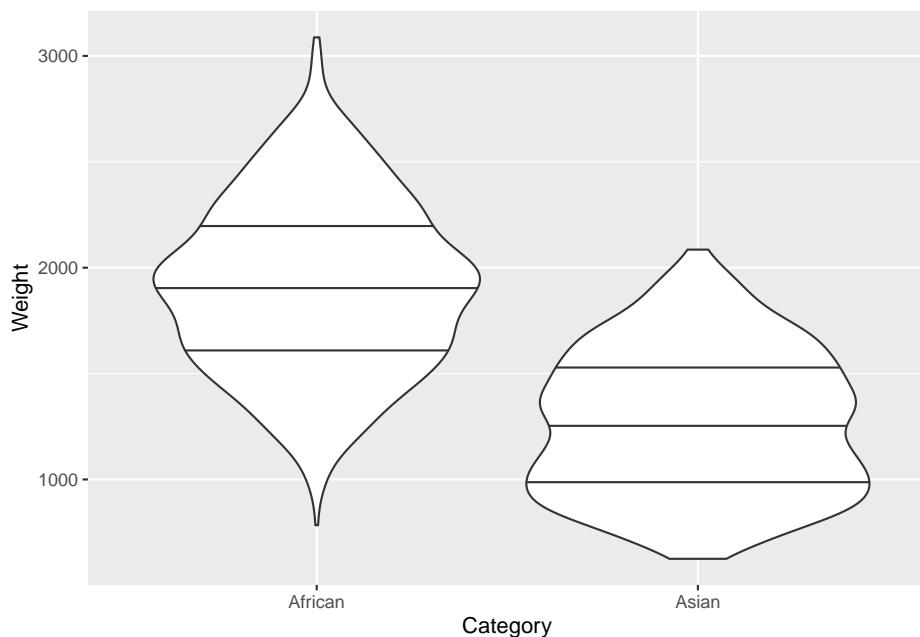
26.2 geom_violin



Without trimming



Draw quantiles



Chapter 27

A: geom_axxxxx

Chapter 28

A: geom_axxxxx

Chapter 29

A: geom_axxxxx

Chapter 30

A: geom_axxxxx

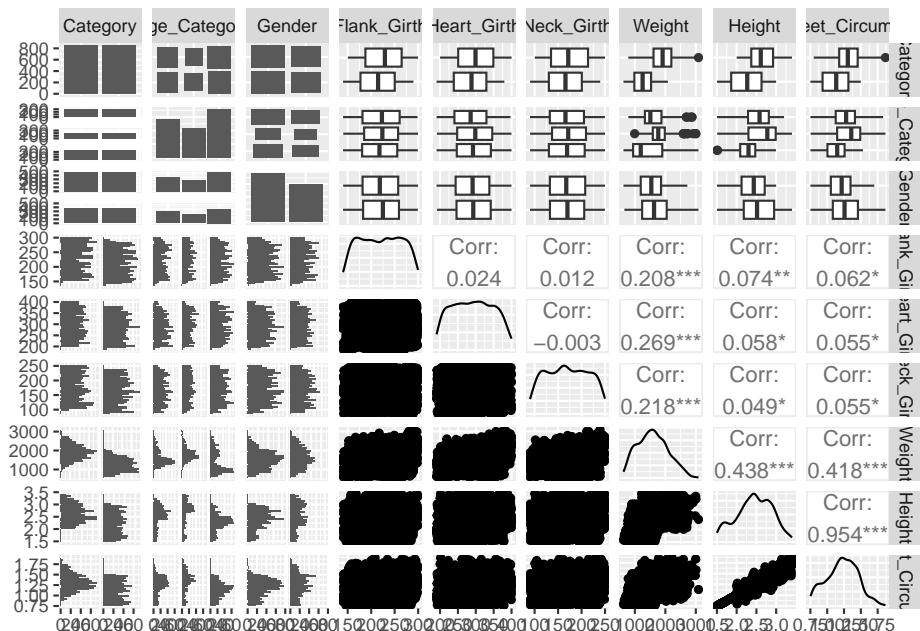
Chapter 31

Others

31.1 ggpairs

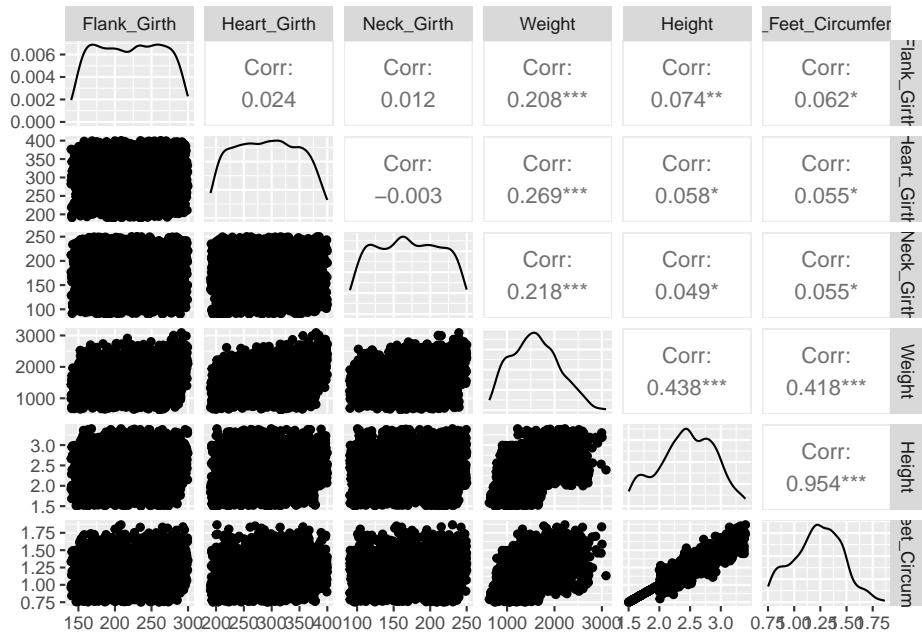
With all variables

```
GGally::ggpairs(elephants)
```



With only numeric variables

```
elephants.numeric <- elephants |> select_if(is.numeric)
GGally::ggpairs(elephants.numeric)
```



Colour the points according to category

```
elephants.numeric <- elephants |> select_if(is.numeric)
elephants.numeric$Category <- elephants$Category
GGally::ggpairs(elephants.numeric, aes(col=Category))
```

