Data Visualisation `geom` Encyclopedia

Thiyanga S. Talagala

2023-08-31

About

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

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.

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/.

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()
```

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.

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.

Graph types

Chapters and sub-chapters

R Packages

This chapter lists all the packages necessary for plotting

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 tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Graph arrangement

```
library(patchwork)
```

Data packages

Cross sectional data

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

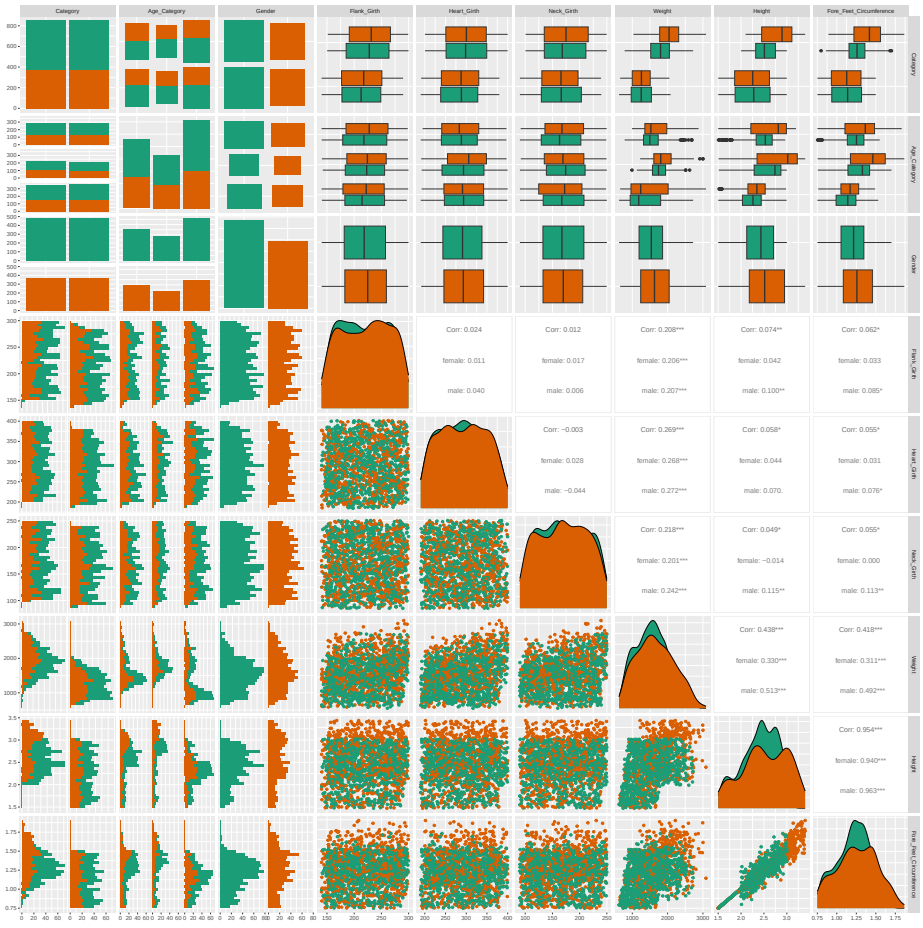
Time series data

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

Large cross sectional dataset

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

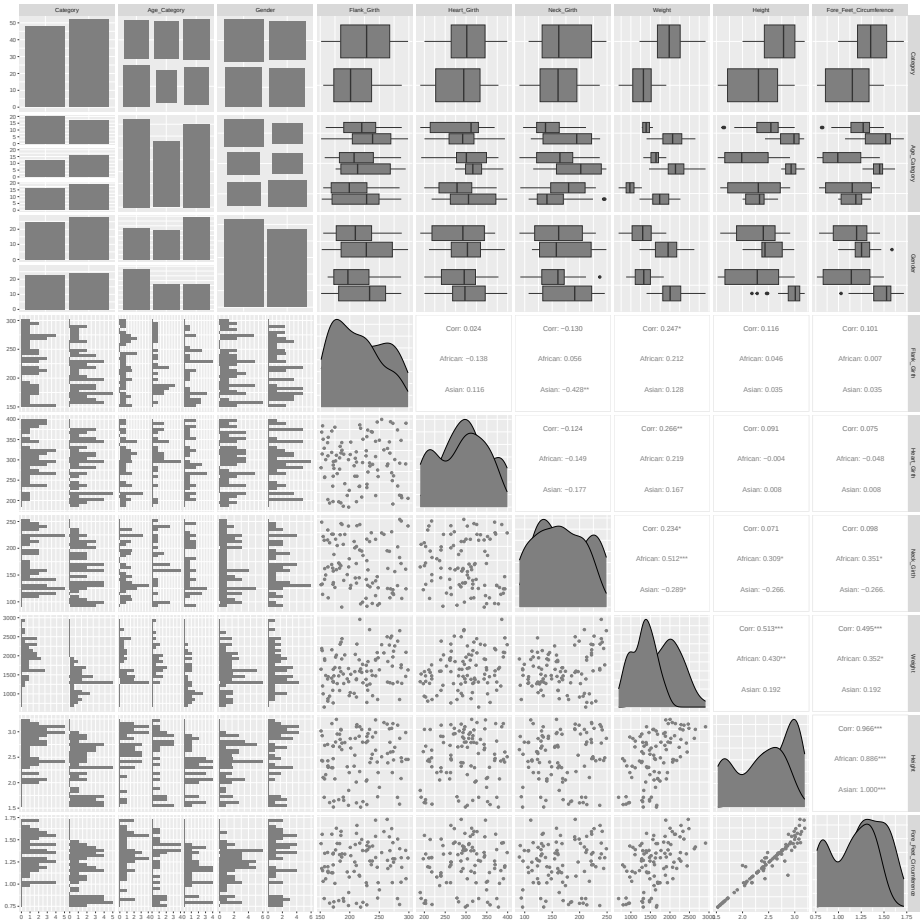
Glimpse of the large dataset



Small cross sectional dataset

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

Glimpse of the small cross sectional dataset



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
```

Packages related to geom

```
library(ggplot2)
#devtools::install_github("EvaMaeRey/ggxmean")
# library(gxmean)
# devtools::install_github("davidsjoberg/ggsankey")
library(ggxmean)
# install.packages("ggpattern")
library(ggpattern)
```

geom Extensions

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

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

x
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
geom_violin
geom_vline

A: geom_a...

geom_abline

Package: ggplot2 [Wickham, 2016]

Book:

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

Understandable aesthetics: alpha, colour, linetype, linewidth

Statistics layer(s):

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
```

geom_area

Package: ggplot2 [Wickham, 2016]

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

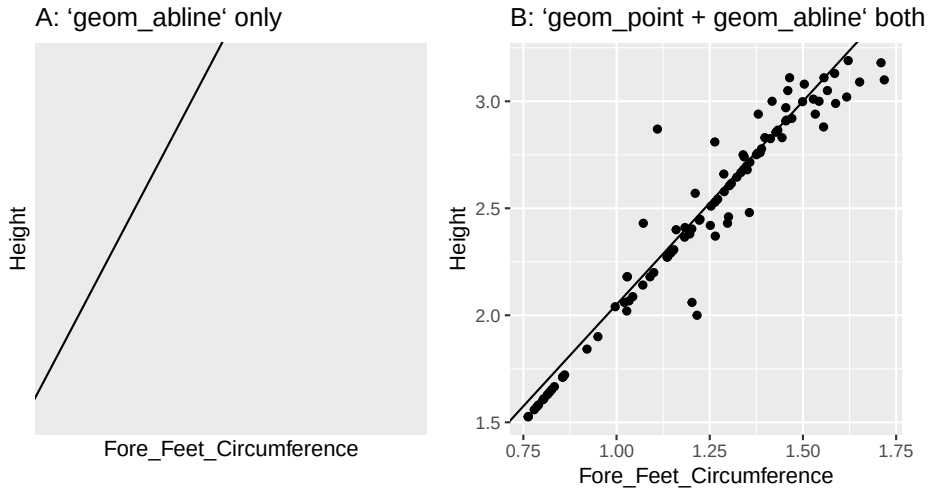


Figure 1: Illustration of (A) geom_abline and (B) use of geom_point and geom_abline both

Understandable aesthetics: alpha, colour, linetype, linewidth

Statistics layer(s):

See also: geom_line, geom_ribbon

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

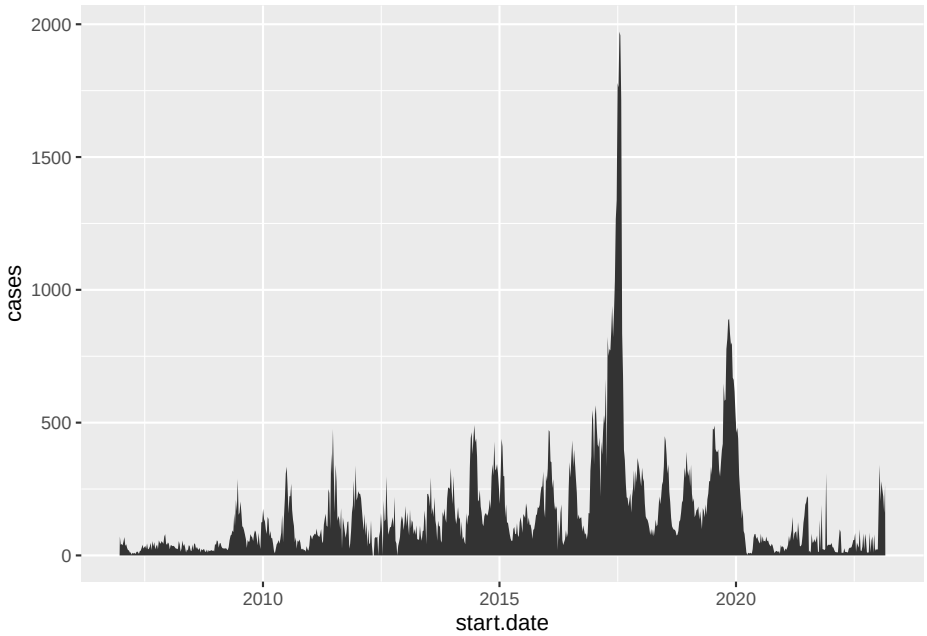


Figure 2: Illustration of geom_area

B: geom_b...

geom_bar

Package: ggplot2 [Wickham, 2016]

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

Statistics layer(s):

`stat_count` - This 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

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")
```

With identity

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

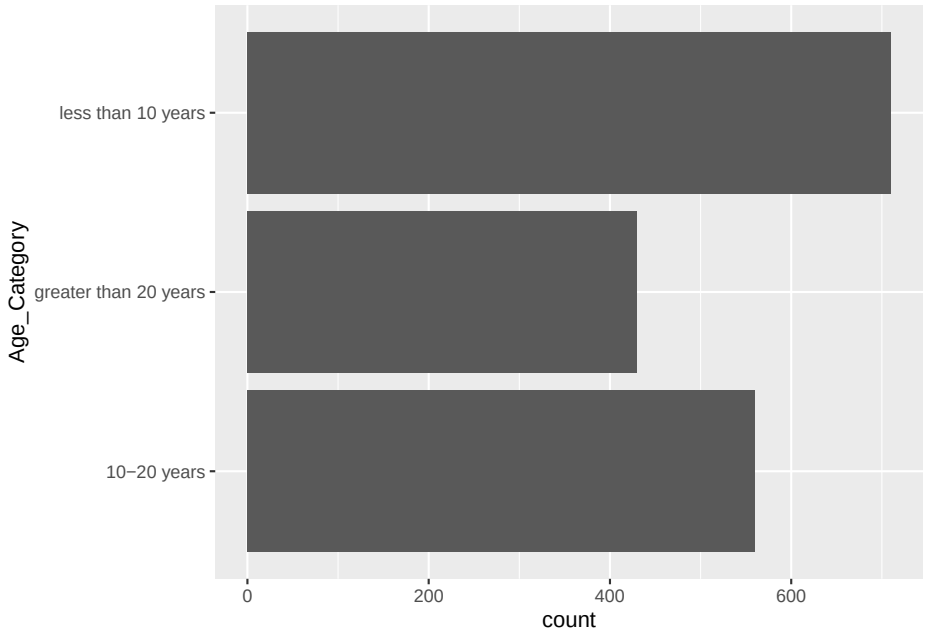


Figure 3: Illustration of geom_bar to create a bar chart

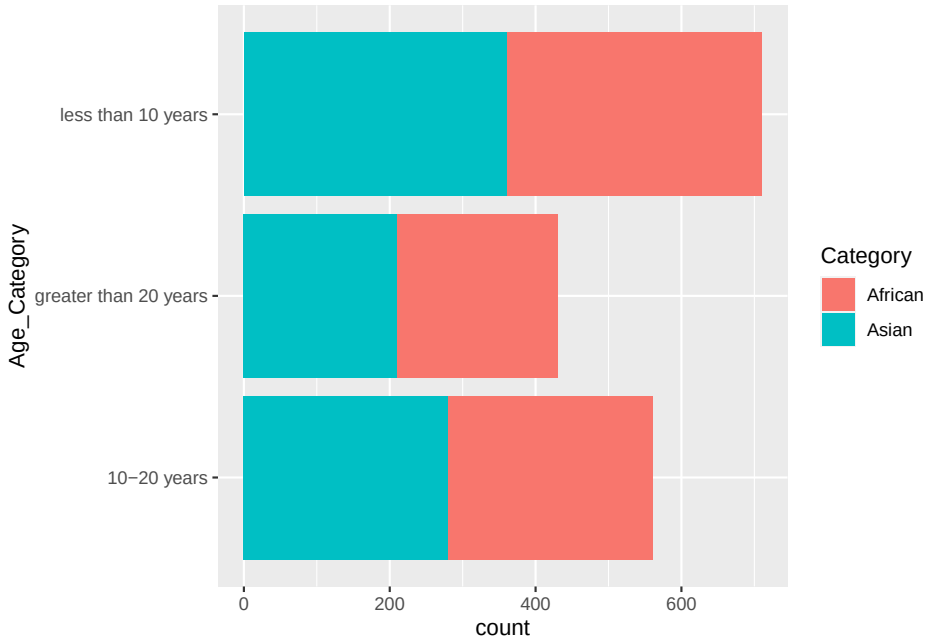


Figure 4: Illustration of geom_bar to create a stacked bar chart

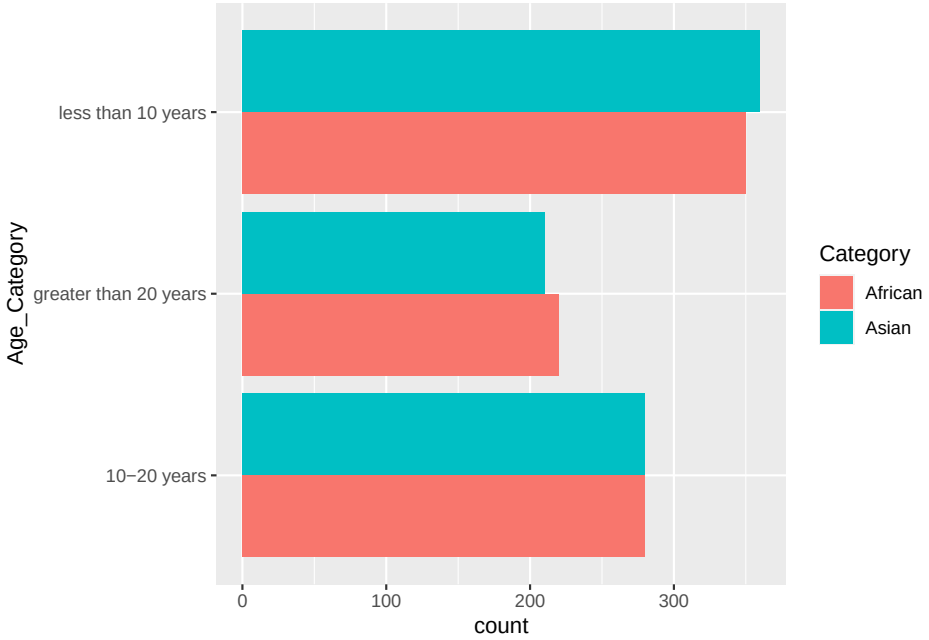


Figure 5: Illustration of geom_bar to create a cluster bar chart

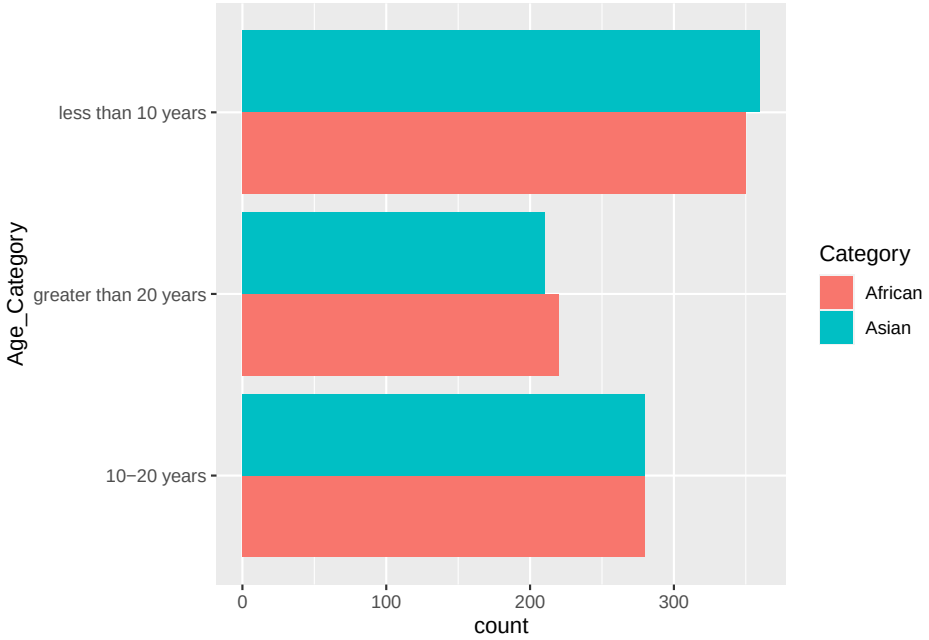
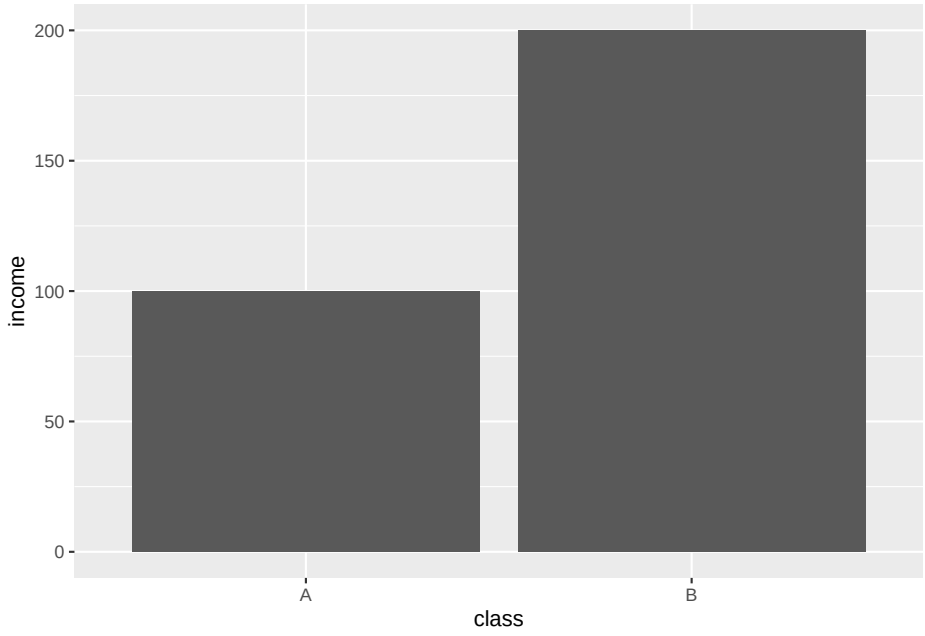


Figure 6: Illustration of geom_bar to create a cluster bar chart



geom_bin_2d

Package: ggplot2 [Wickham, 2016]

Description: Divides the Cartesian plane created by x-variable and y-variable into rectangles (2D histogram), counts the number of observations in each rectangle. Only the observations with rectangles are filled according to the number ofobservations.

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)
```

geom_bin2d_pattern

Package: ggpattern [FC et al., 2022]

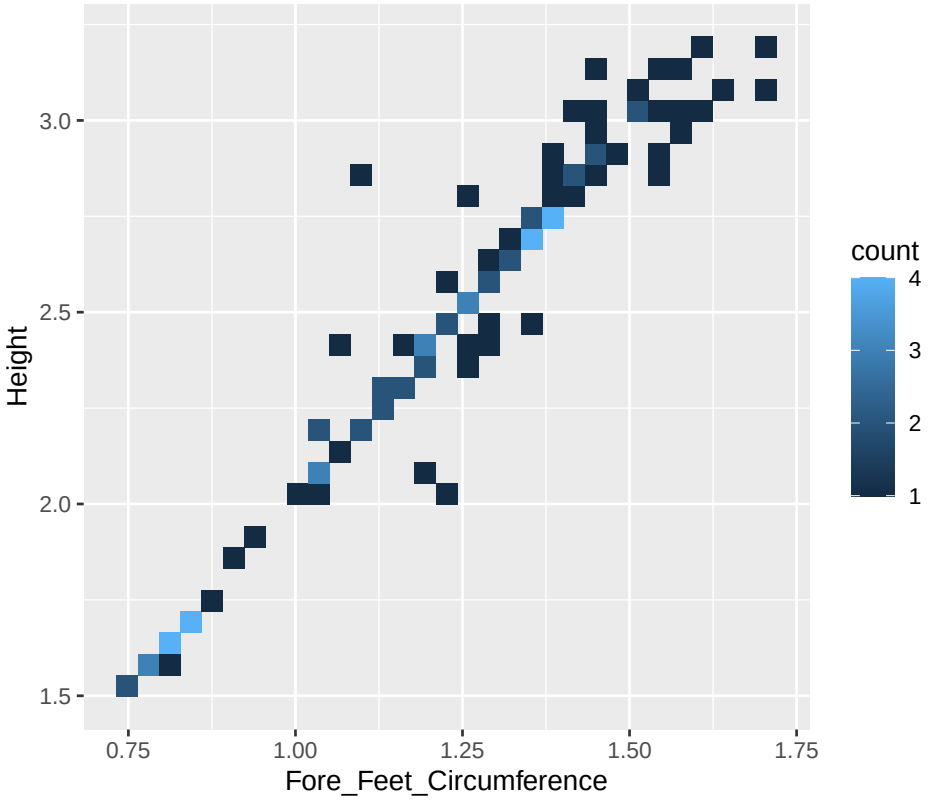


Figure 7: Illustration of using geom_bin_2d

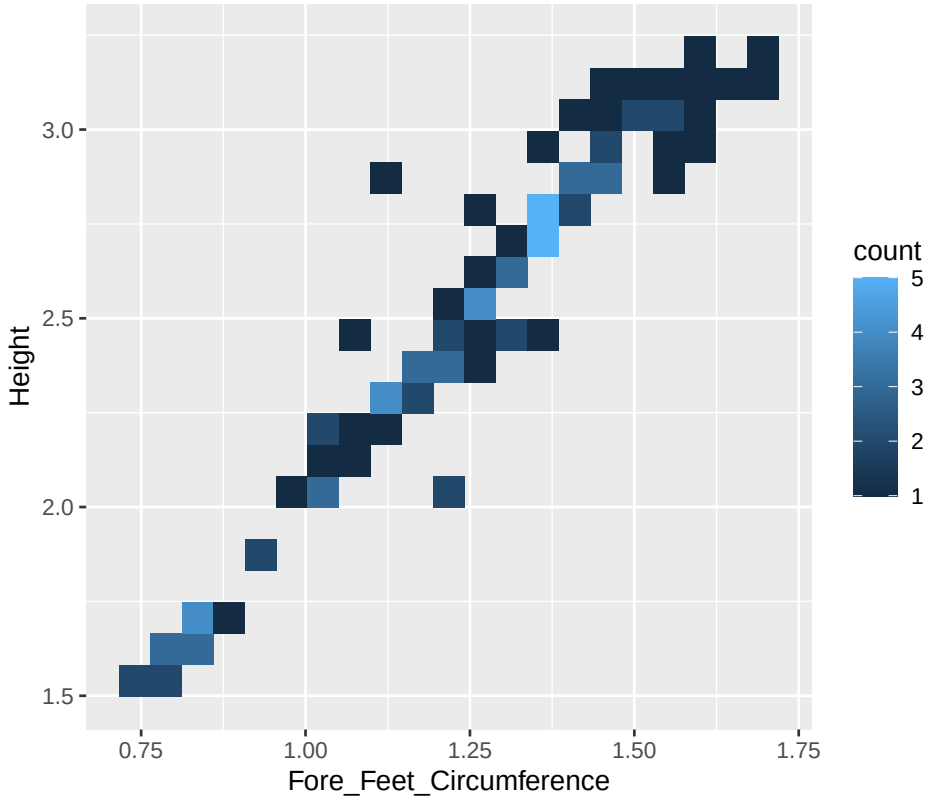


Figure 8: Illustration of changing bins in geom_bin_2d

Description: Divides the Cartesian plane created by x-variable and y-variable into rectangles (2D-Histogram), counts the number of observations in each rectangle. Only the observations with rectangles are filled with a pattern.

Understandable aesthetics:

pattern_fill (pattern_* - for mapping variables under aesthetics), pattern (to set a patten, for example pattern='stripe'), fill, colour

Statistics layer(s):

See also: geom_bin2d, geom_point

```
ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) +
  ggpattern::geom_bin2d_pattern(bins=5) +
  theme(aspect.ratio = 1)
```

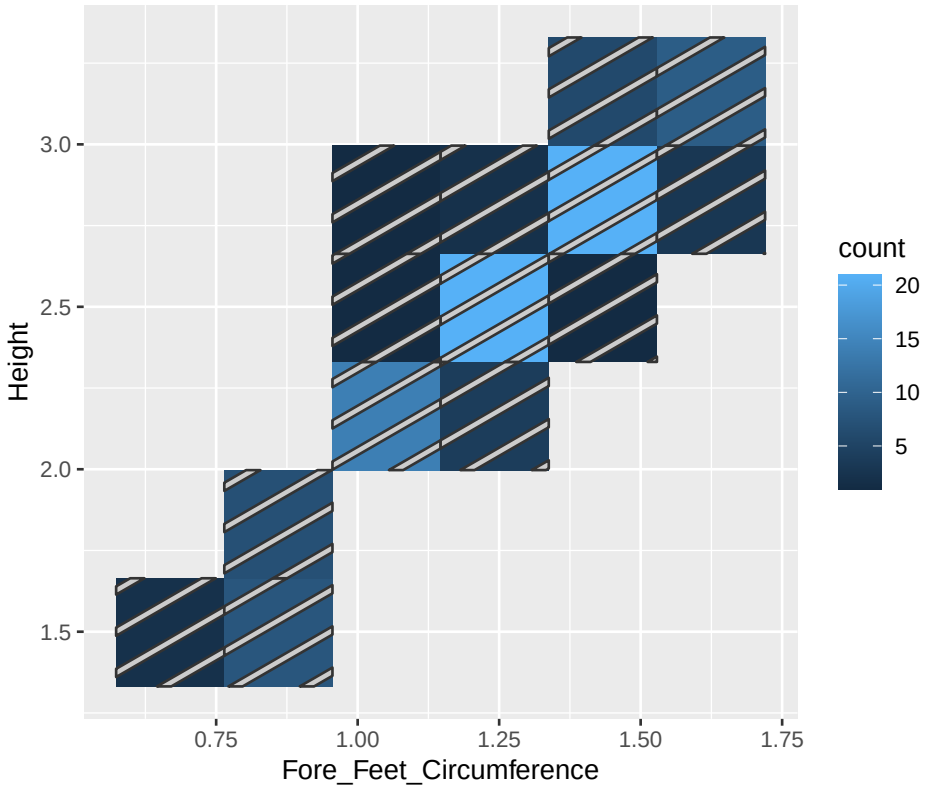


Figure 9: Illustration of using geom_bin_2d

geom_bin2d

Package: ggplot2 [Wickham, 2016]

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 ofobservations.

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)
```

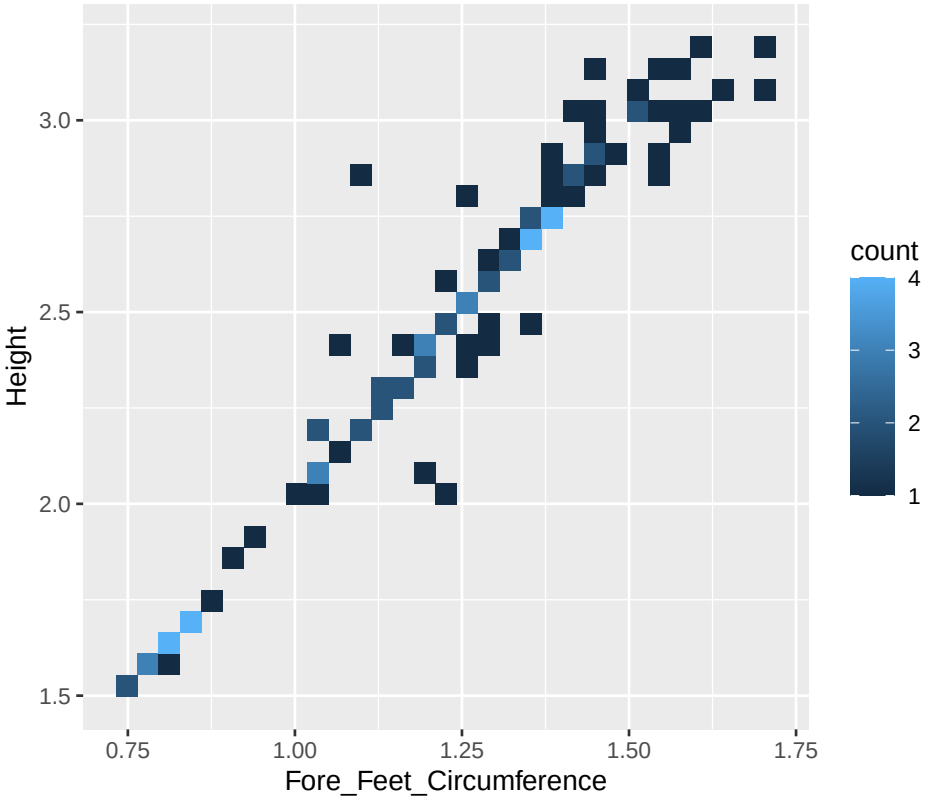


Figure 10: Illustration of using geom_bin_2d

geom_blank

Package: ggplot2 [Wickham, 2016]

Description: Draws nothing.

geom_boxplot

Package: ggplot2 [Wickham, 2016]

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 ($Q1$), third quartile ($Q3$), upper whisker extends up to $Q3 + 1.5 \times IQR$ and lower whisker extends up to $Q1 - 1.5 \times IQR$, where $IQR = Q3 - Q1$. In a notched box plot, it creates 95% confidence interval for mean.

See also: geom_col

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

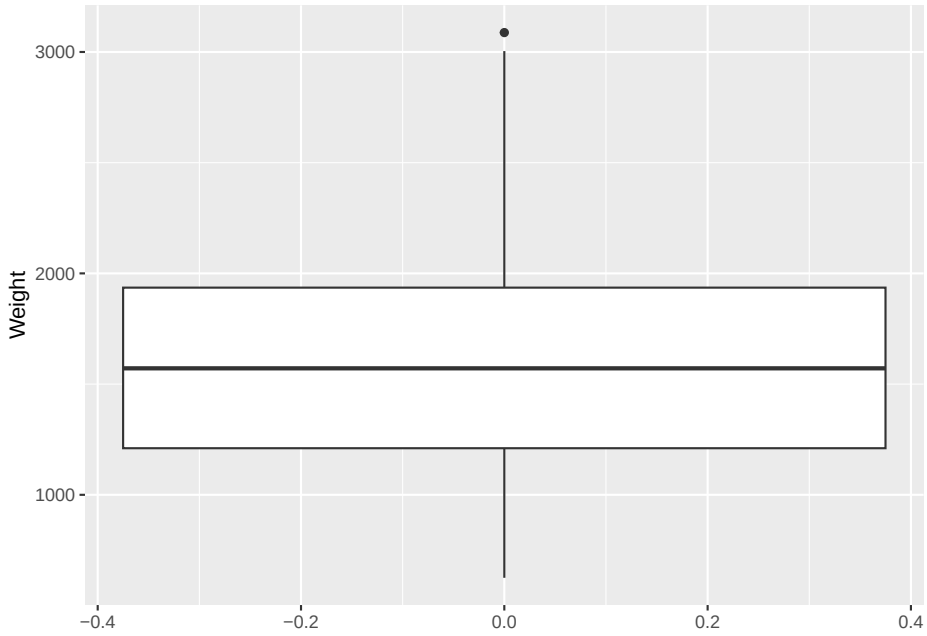


Figure 11: Illustration of using geom_boxplot

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

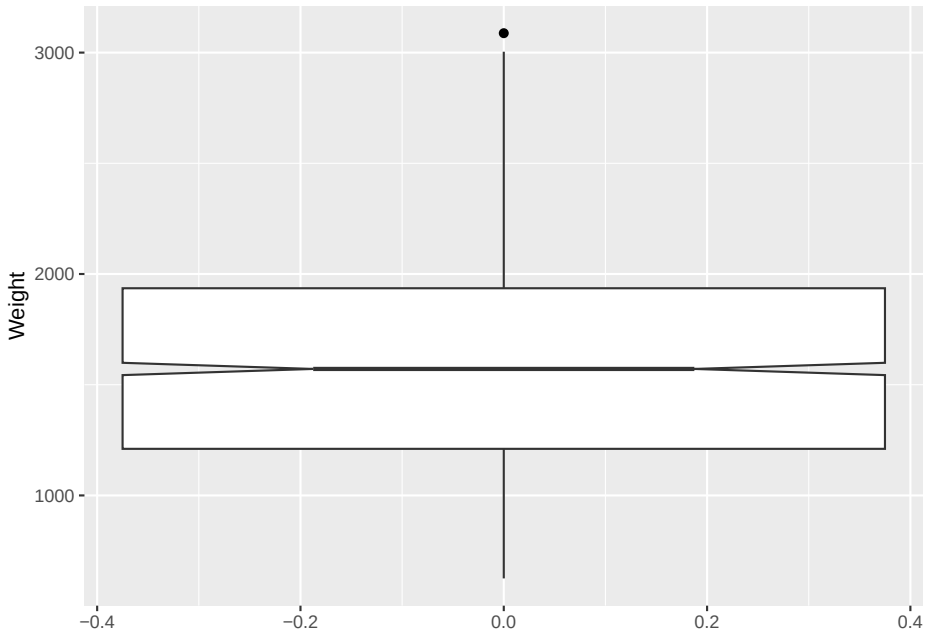


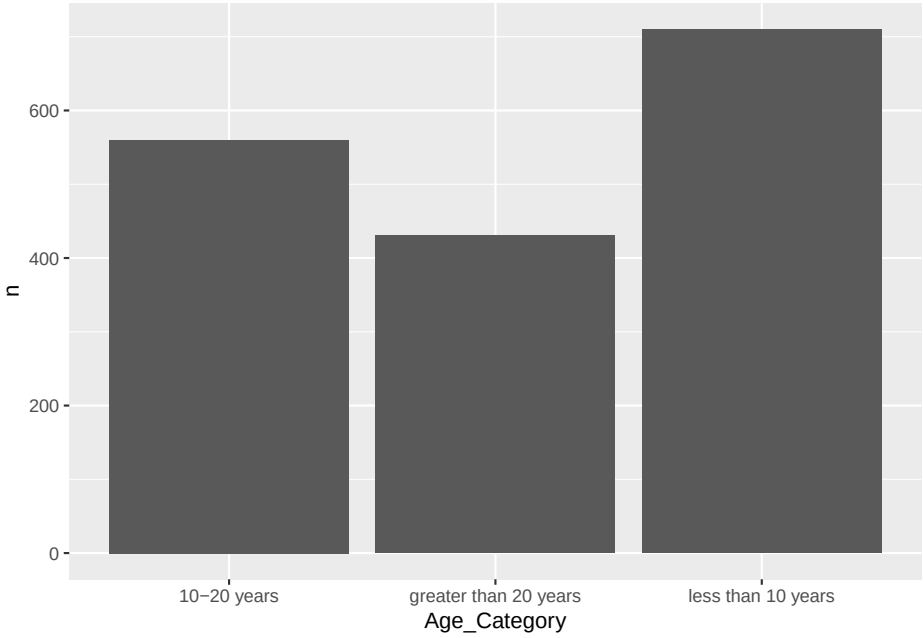
Figure 12: Illustration of using geom_boxplot with changing outliers and adding a notch to create notched box plot.

C: geom_c...

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
```

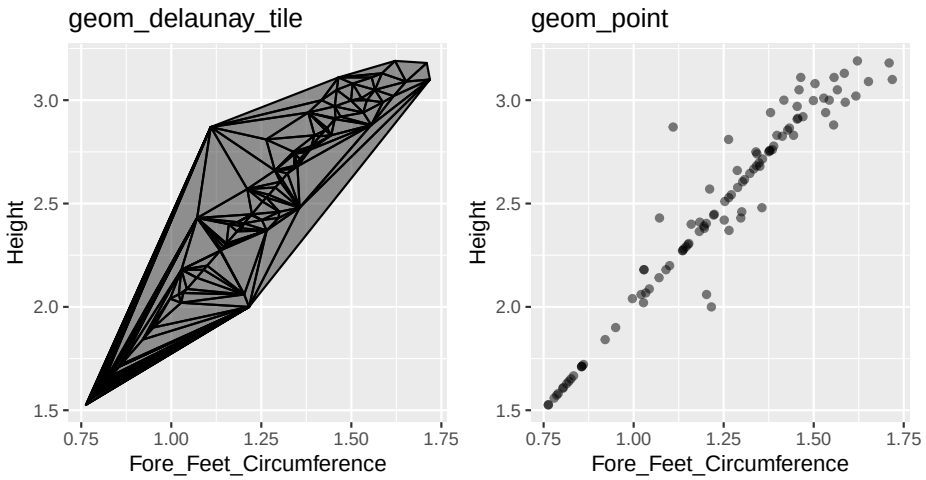


D: geom_d...

geom_delaunay_tile

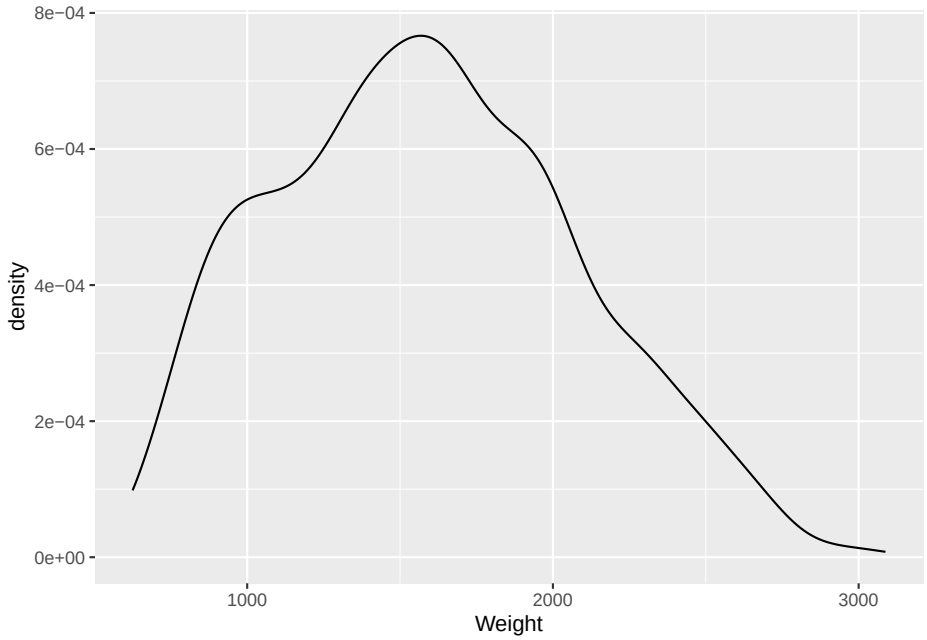
Description: The Delaunay triangulation is used to create a planar graph.

```
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
```



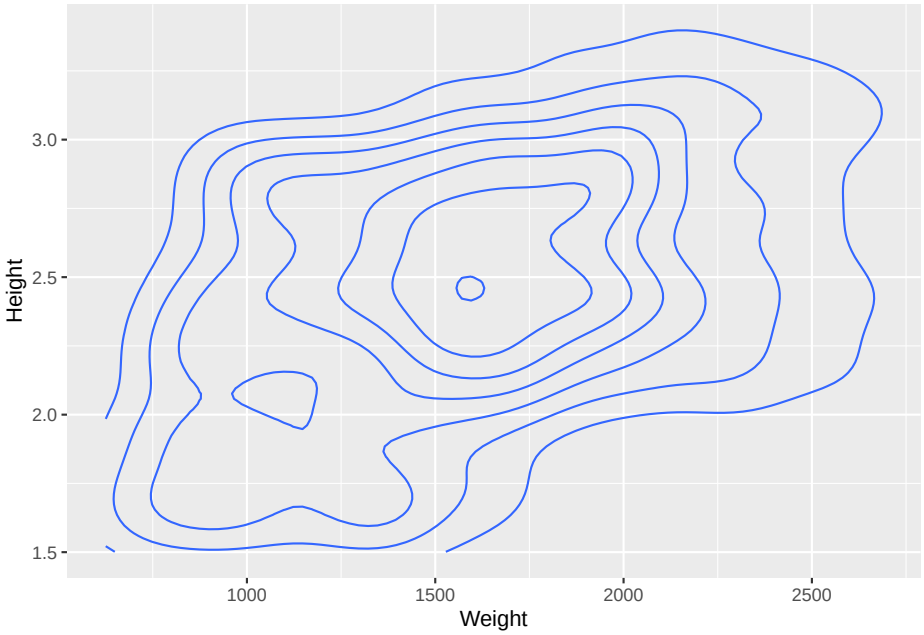
```
ggplot(elephants, aes(x = Weight)) +
  geom_density()
```

geom_density



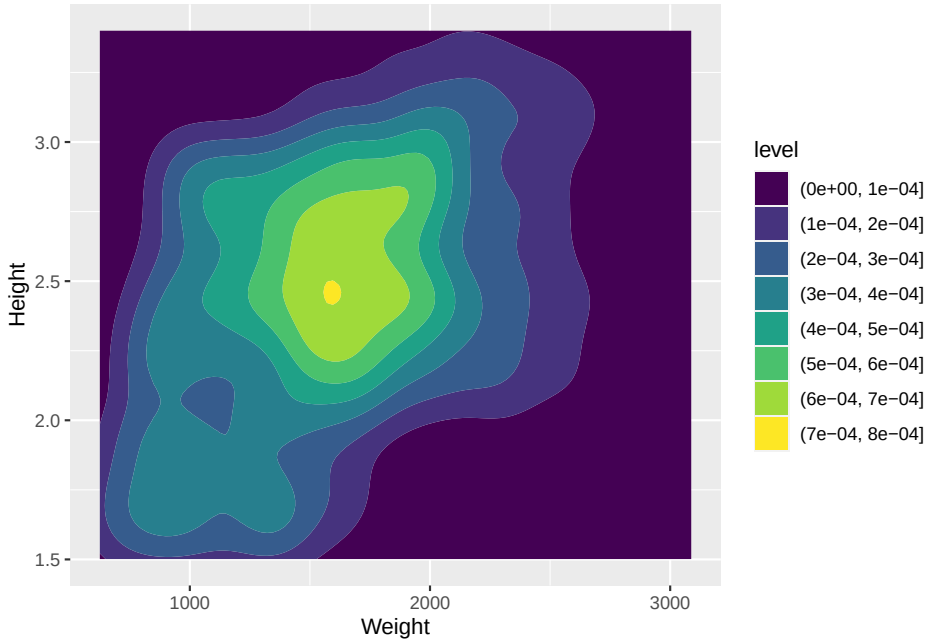

```
ggplot(elephants, aes(y = Height, x=Weight)) +
  geom_density_2d()
```

geom_density_2d




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

geom_density_2d_filled

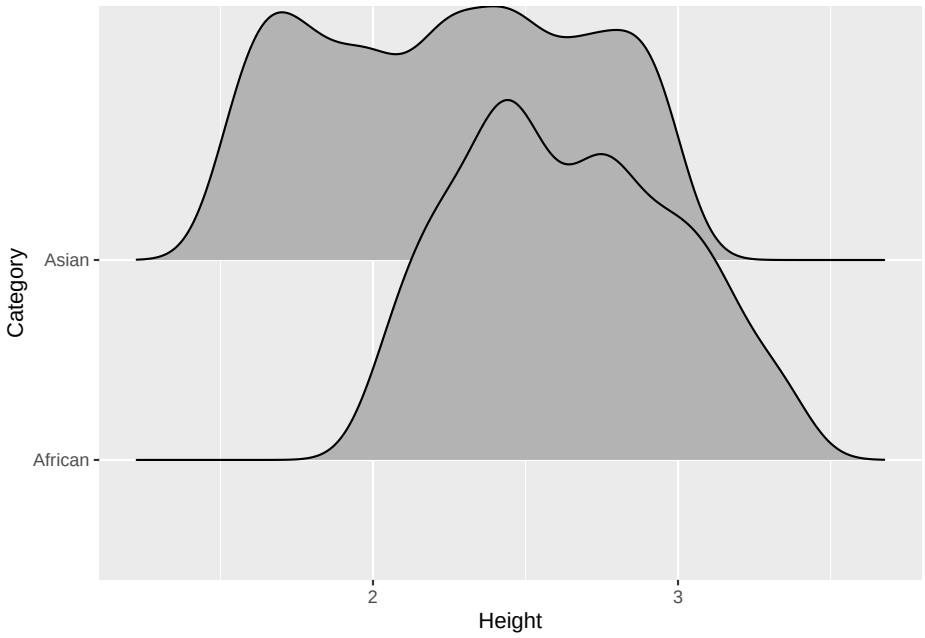


geom_density_ridges

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")
```

geom_dl


Asian
African
Height
Weight
3.0
2.5
2.0
1.5
1000
2000
3000
Category
African
Asian

E: geom_e...

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
```

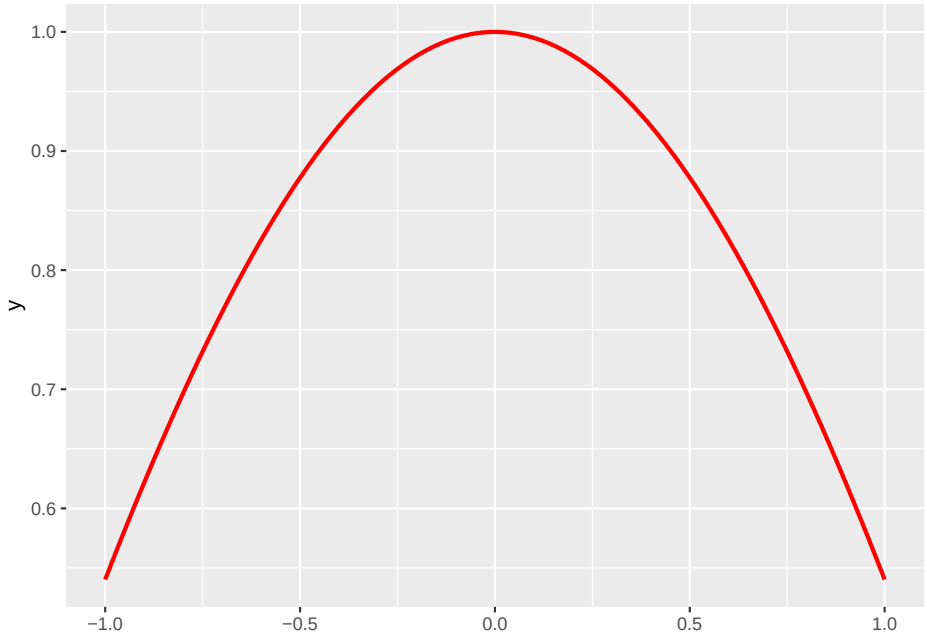
`geom_encircle` with `geom_point`


Height
3.0
2.5
2.0
1.5
1000
2000
3000
Weight
Category
African
Asian

F: geom_f...

geom_function

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

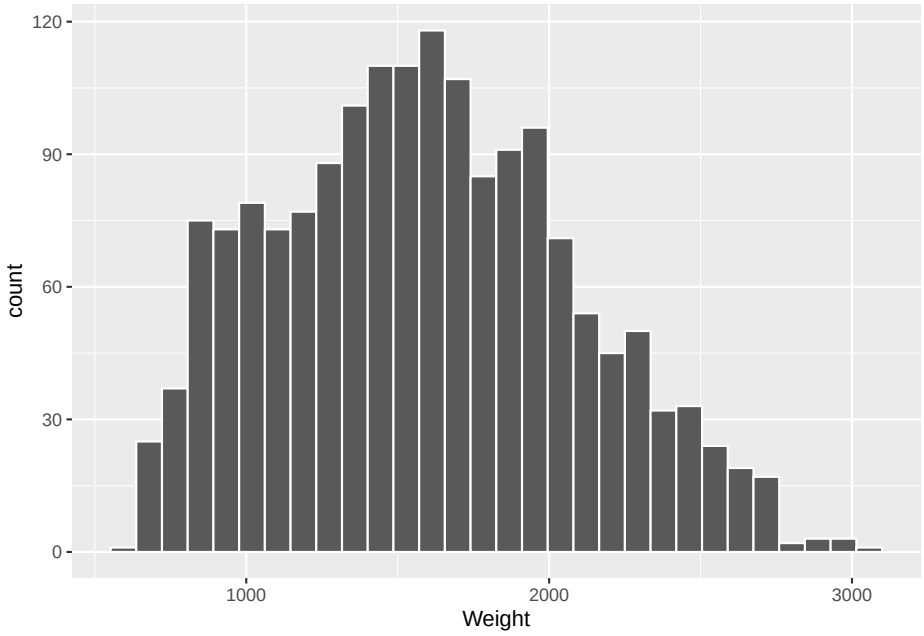


A: geom_axxxxx

H: geom_h...

geom_histogram

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



geom_hline

Package: ggplot2 [Wickham, 2016]

Book:

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

Understandable aesthetics: alpha, colour, linetype, linewidth

Statistics layer(s):

See also: geom_point, geom_vline, geom_hline

Example:

```
hline <- ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) + geo
  labs(title="A: `geom_hline` only") +
  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
```

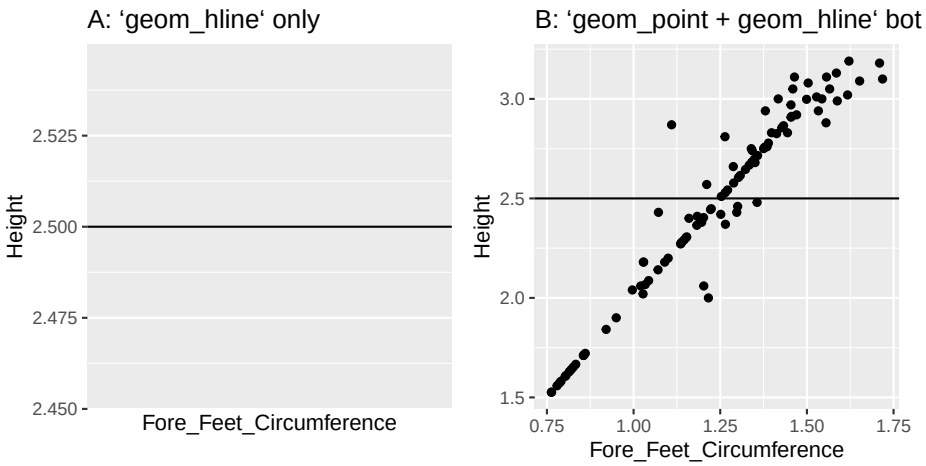


Figure 13: Illustration of (A) geom_hline and (B) use of geom_point and geom_hline both

A: geom_axxxxx

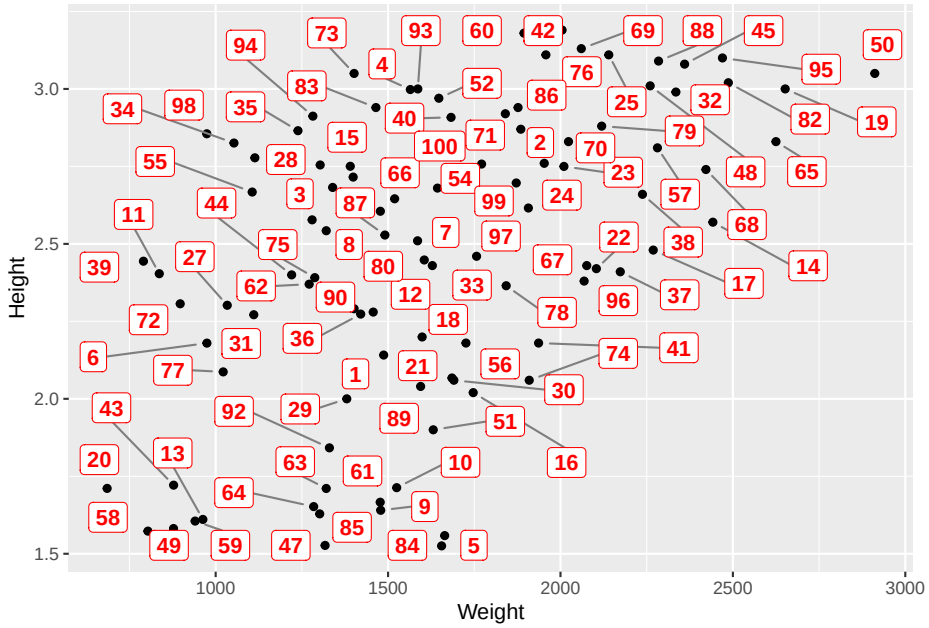
A: geom_axxxxx

A: geom_axxxxx

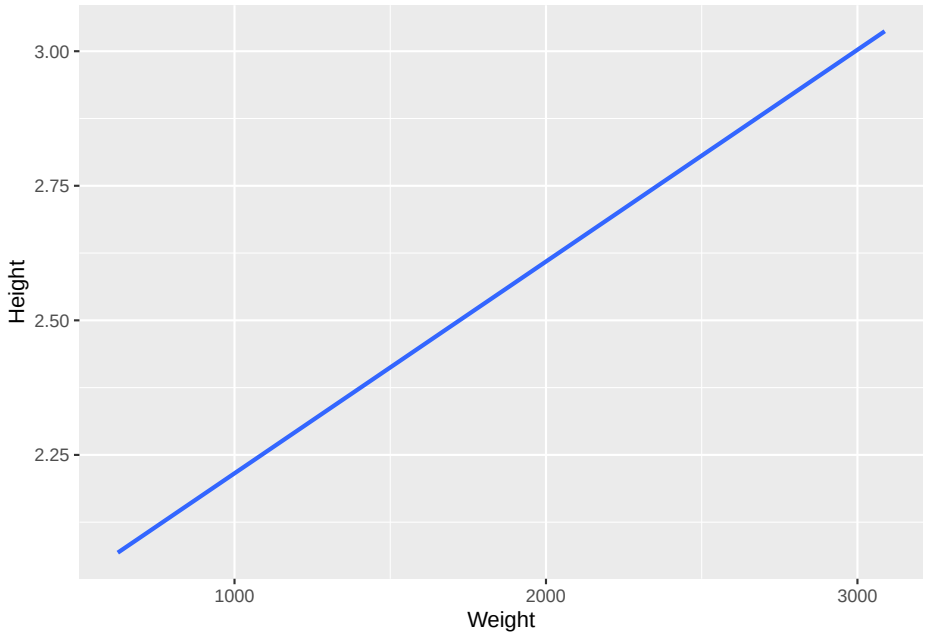
L: geom_l...

```
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
```


Height
Weight
3.0
2.5
2.0
1.5
1000
1500
2000
2500
3000

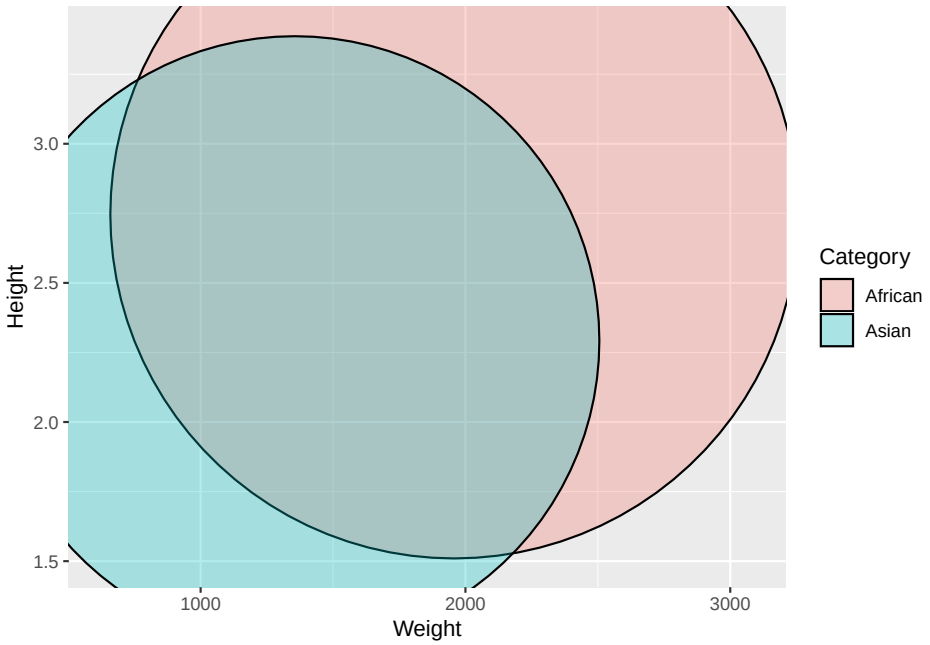
geom_lm


Height
Weight
3.00
2.75
2.50
2.25
1000
2000
3000

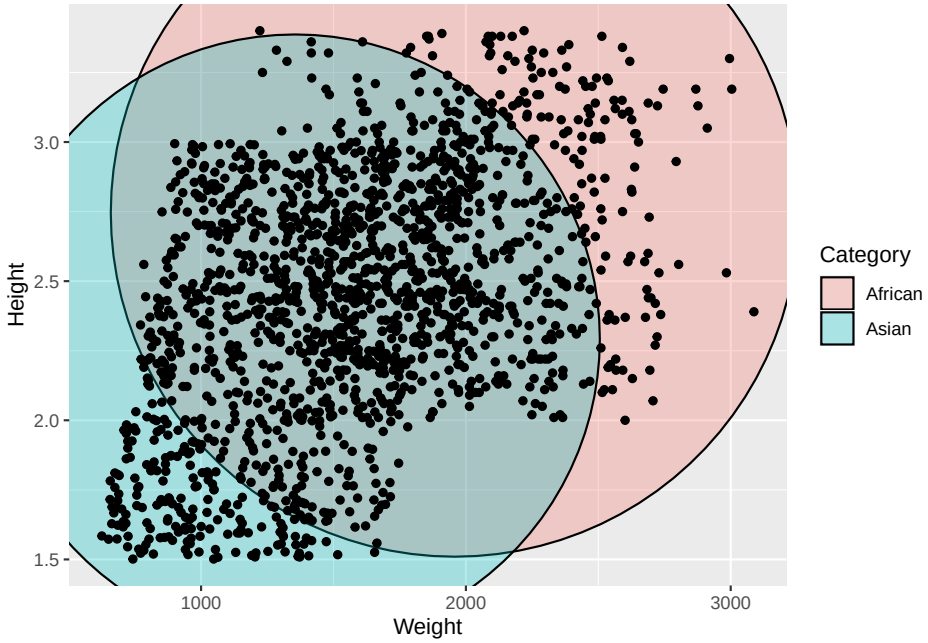
M: geom_m...

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

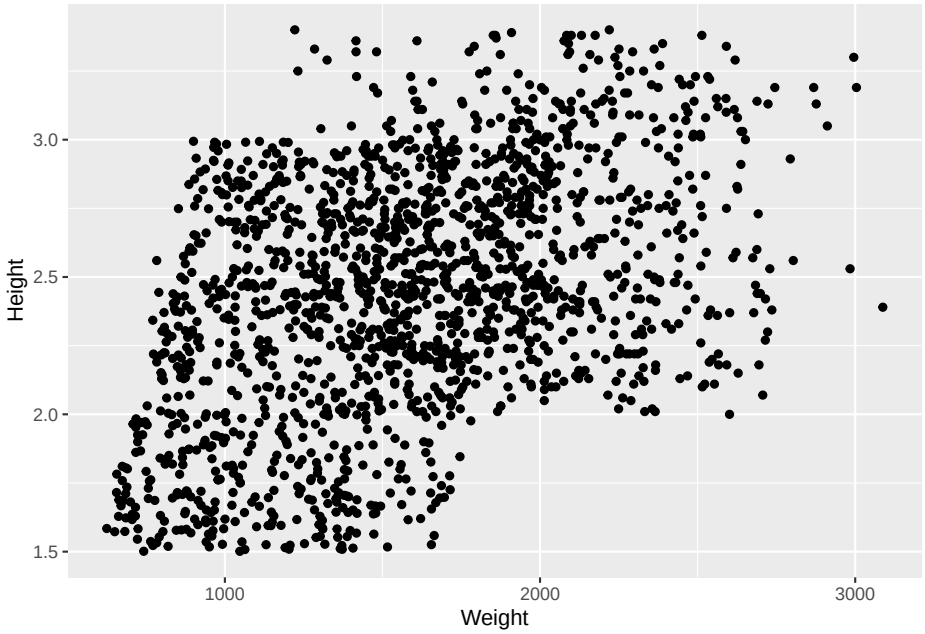

3.0
2.5
2.0
1.5
Height
1000
2000
3000
Weight
Category
African
Asian

A: geom_axxxxx

A: geom_axxxxx

P: geom_p...

geom_point



A: geom_axxxxx

R: geom_r...

geom_ribbon

Package: ggplot2 [Wickham, 2016]

Description: Displays a interval according given the upper interval boundary(ymax or xmax) and lower interval boundary (ymin or xmin).

Understandable aesthetics:

Required aesthetics: x or y, ymin or xmin, ymax or xmax

Other: alpha, colour, fill, group, linetype, linewidth

Statistics layer(s):

See also: geom_area, geom_line

```
colombo21 <- srilanka_weekly_data |>
  filter(district == "Colombo" & year==2021)
ribbon <- ggplot(data=colombo21, aes(x=start.date, y=cases)) +
  geom_ribbon(aes(ymin = cases - 50, ymax = cases + 50), fill = "grey70") + labs(title = "A: `geo

ribbonline <- ggplot(data=colombo21, aes(x=start.date, y=cases)) +
  geom_ribbon(aes(ymin = cases - 50, ymax = cases + 50), fill = "grey70") + geom_line() +
  labs(title = "B: `geom_ribbon + geom_line` both")

ribbon|ribbonline
```

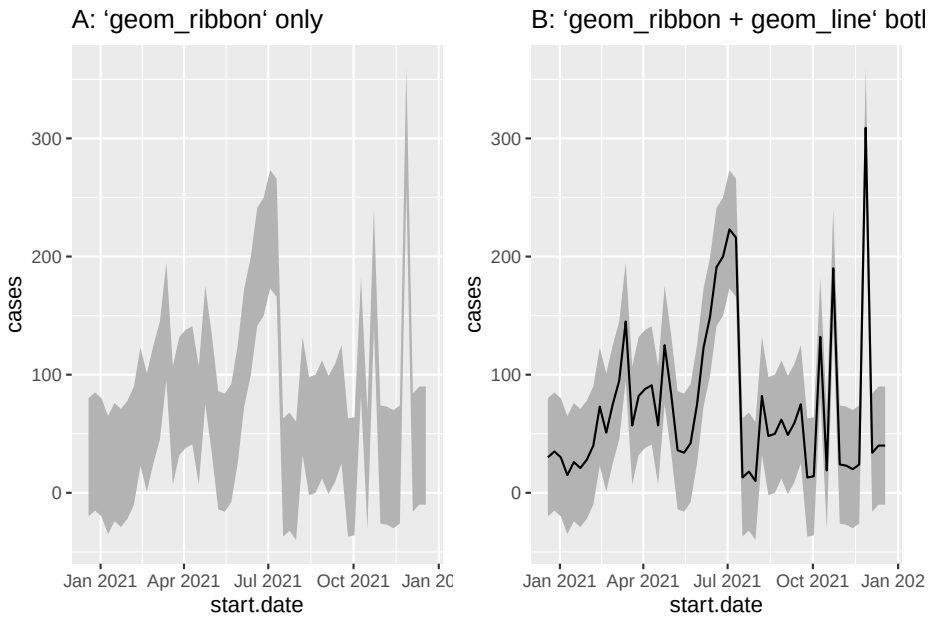


Figure 14: Illustration of geom_ribbon, A: geom_ribbon only and B: geom_ribbon + geom_line

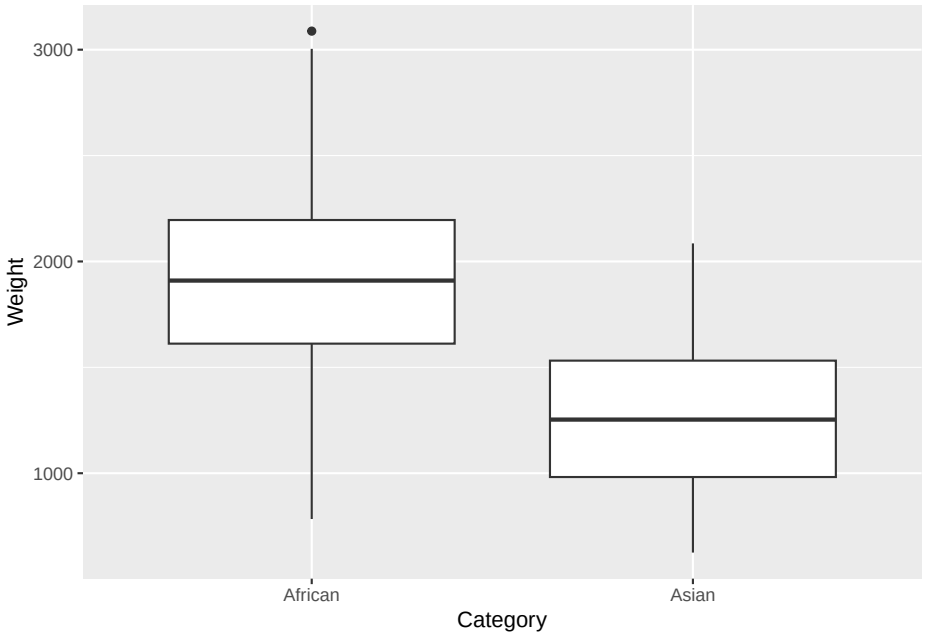
S: geom_s...

geom_segment

geom_signif

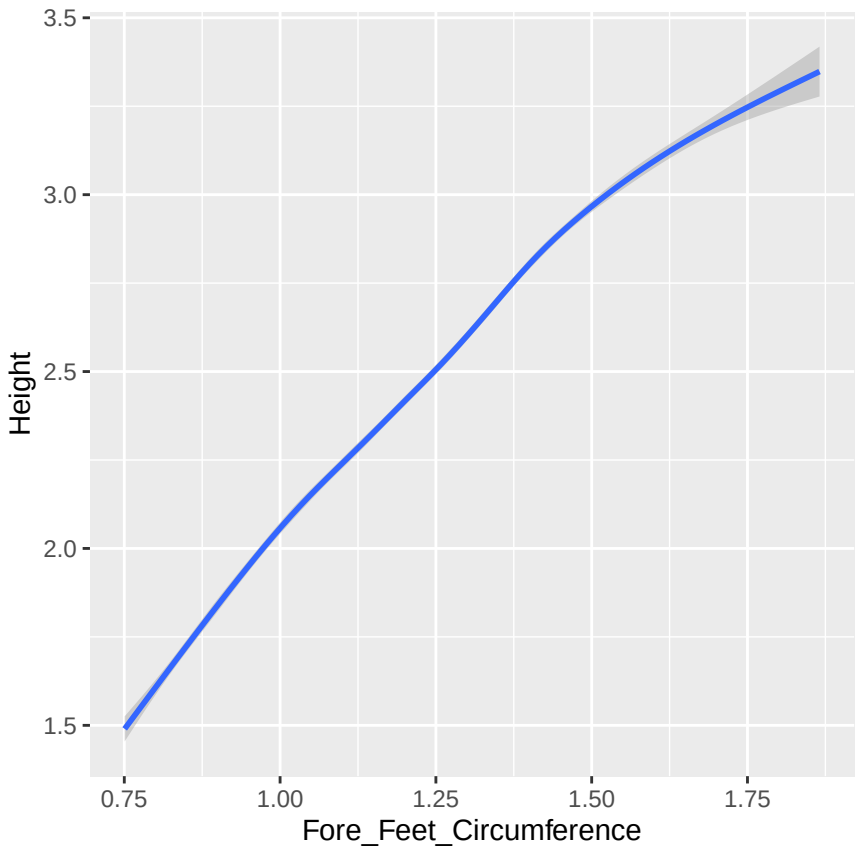
```
ggplot(elephants, aes(y = Weight, x=Category)) + geom_boxplot() + ggsignif::geom_signif()
```

```
## Warning: Computation failed in `stat_signif()`
## Caused by error in `$<-.data.frame`:
## ! replacement has 1 row, data has 0
```



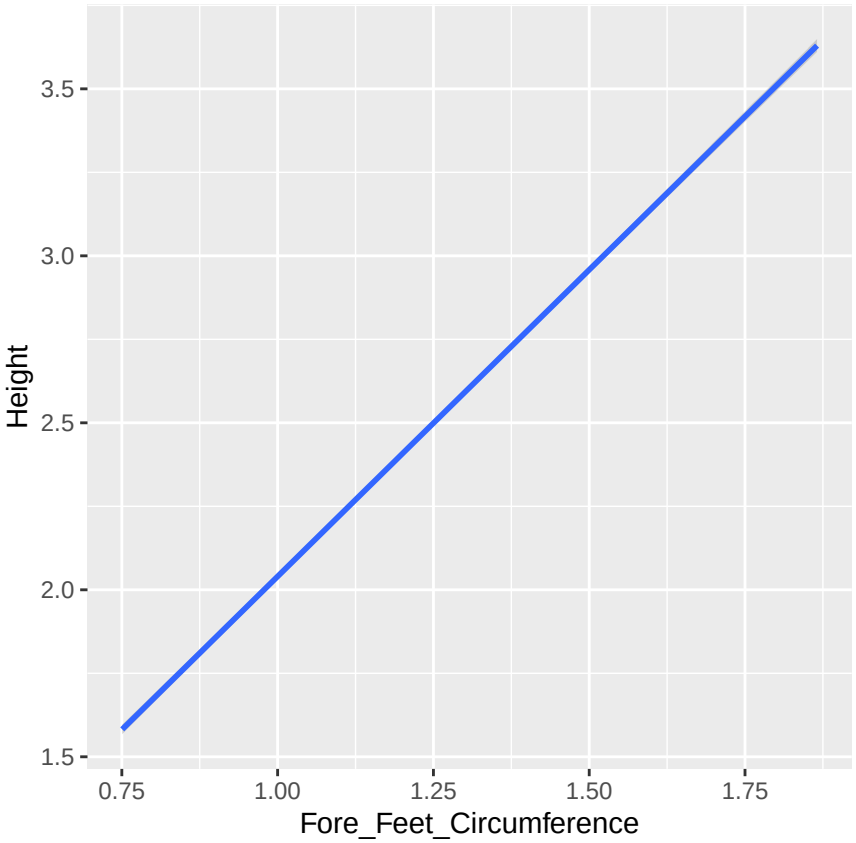
```
ggplot(elephants, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_smooth() + theme(aspect.ratio=1)
```

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

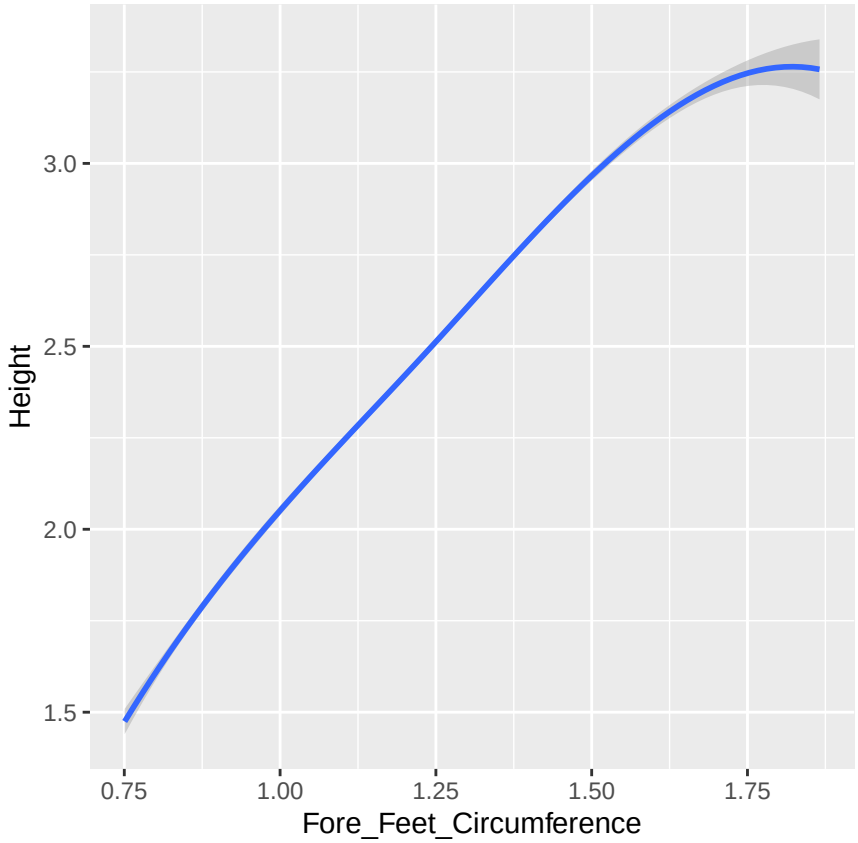



```
ggplot(elephants, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_smooth(method=lm) +
  theme(aspect.ratio=1)
```

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




```
ggplot(elephants, aes(y = Height, x=Fore_Feet_Circumference)) +
  geom_smooth(method = lm, formula = y ~ splines::bs(x, 4)) + theme(aspect.ratio=1)
```


Height
3.0
2.5
2.0
1.5
0.75
1.00
1.25
1.50
1.75
Fore_Feet_Circumference

A: geom_axxxxx

A: geom_axxxxx

V: geom_v...

geom_vline

Package: ggplot2 [Wickham, 2016]

Book:

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

Understandable aesthetics: alpha, colour, linetype, linewidth

Statistics layer(s):

See also: geom_point, , geom_hline, geom_smooth, geom_vline

Example:

```
vline <- ggplot(elephants.subset.100, aes(y = Height, x=Fore_Feet_Circumference)) + geom_vline(xi
  labs(title="A: `geom_vline` only") +
  theme(aspect.ratio = 1)

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
```

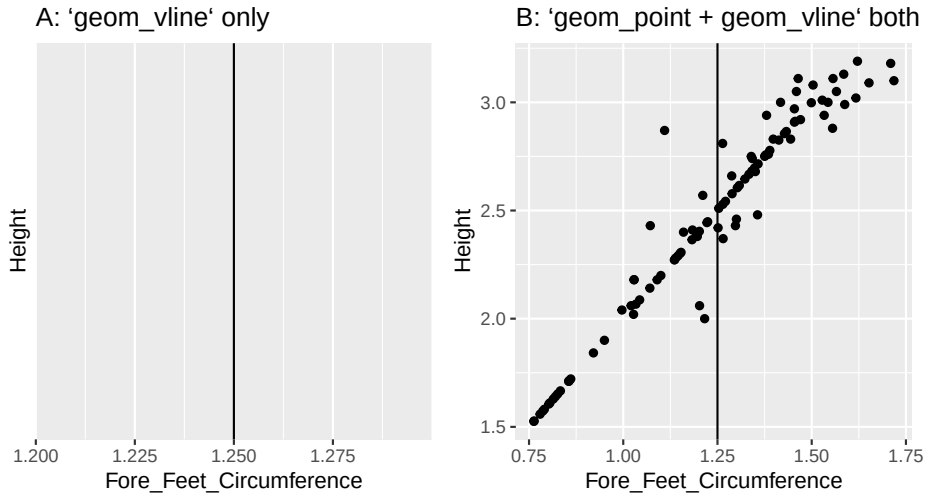
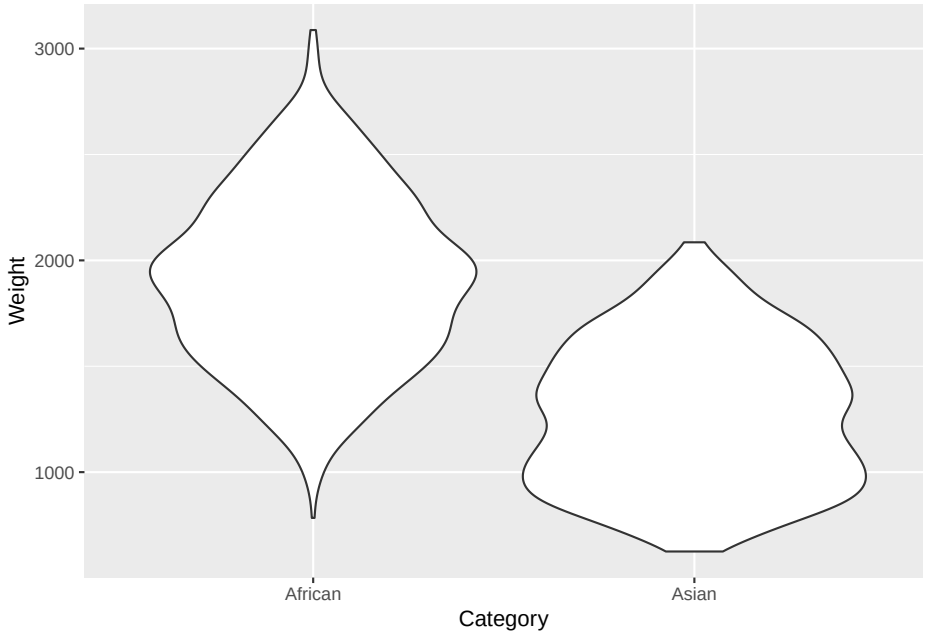
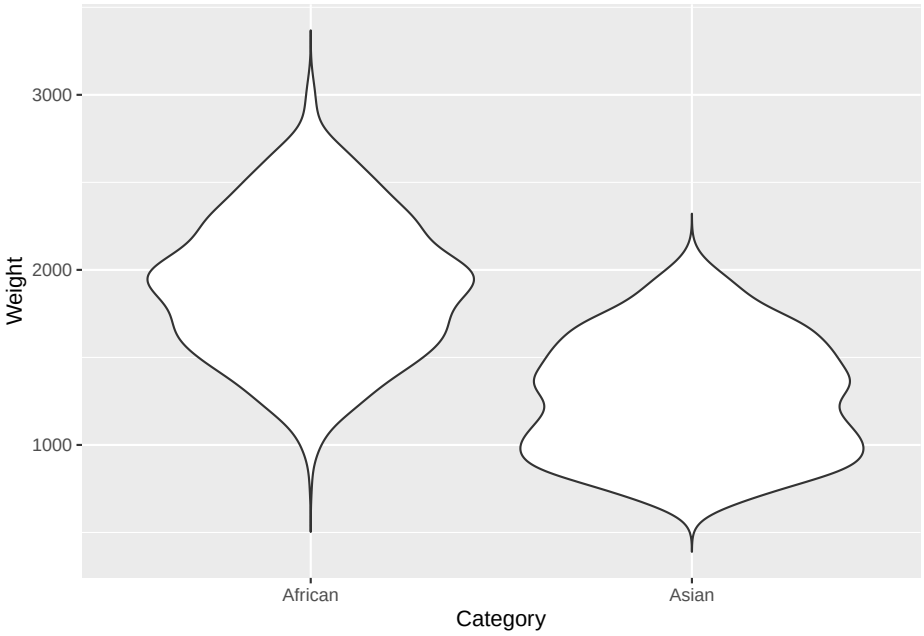


Figure 15: Illustration of (A) geom_vline and (B) use of geom_point and geom_vline both

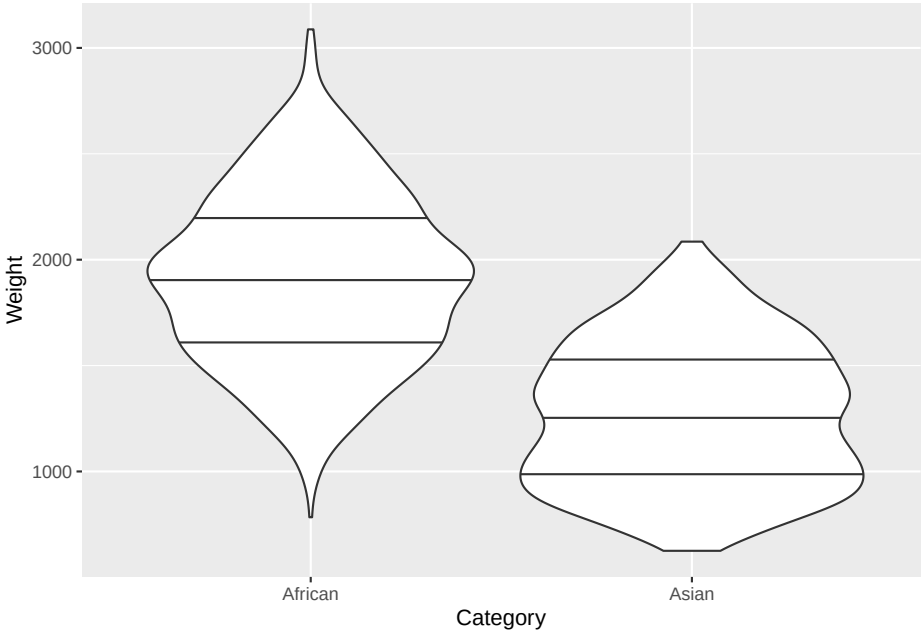
geom_violin



Without trimming



Draw quantiles



A: geom_axxxxx

A: geom_axxxxx

A: geom_axxxxx

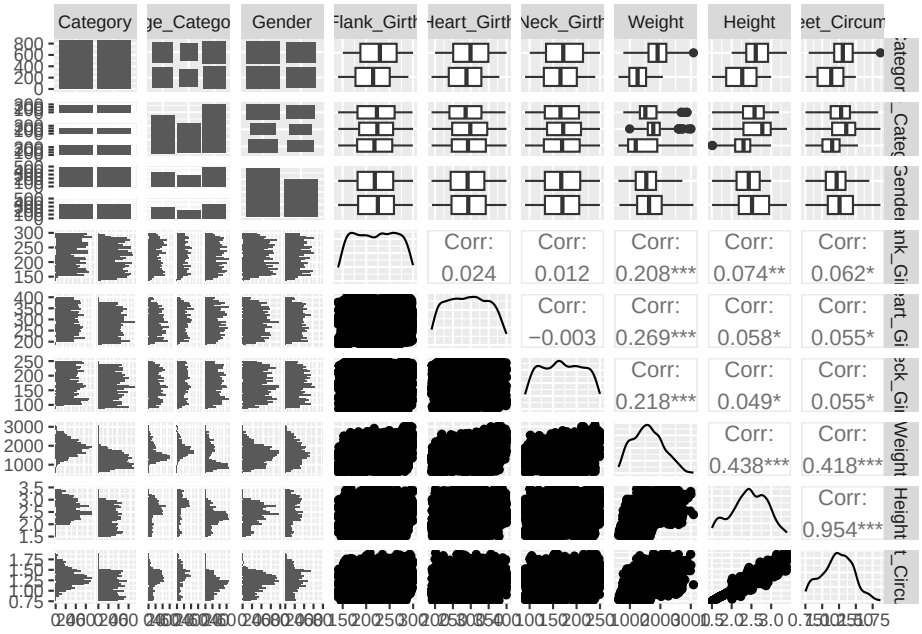
A: geom_axxxxx

Others

ggpairs

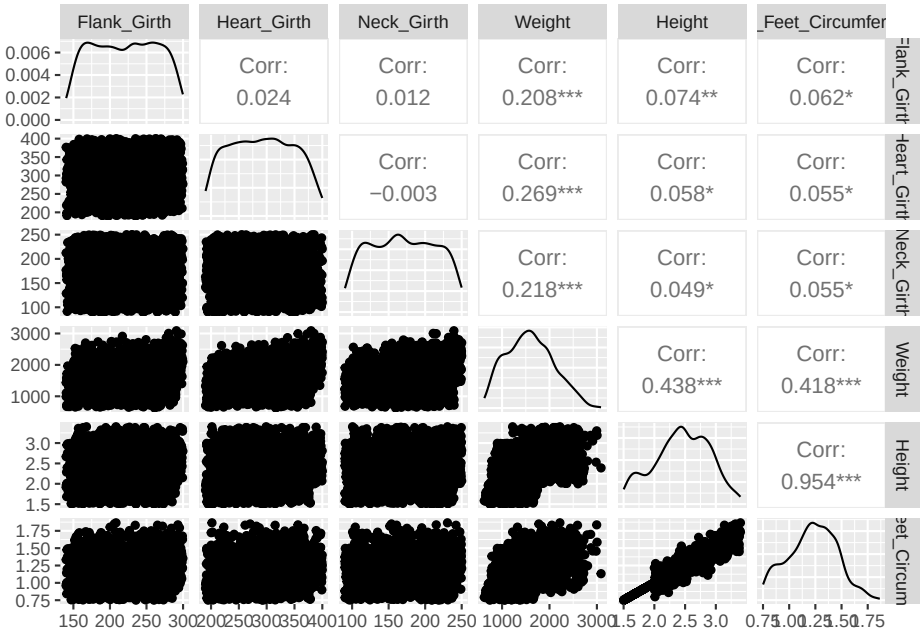
With all variables

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



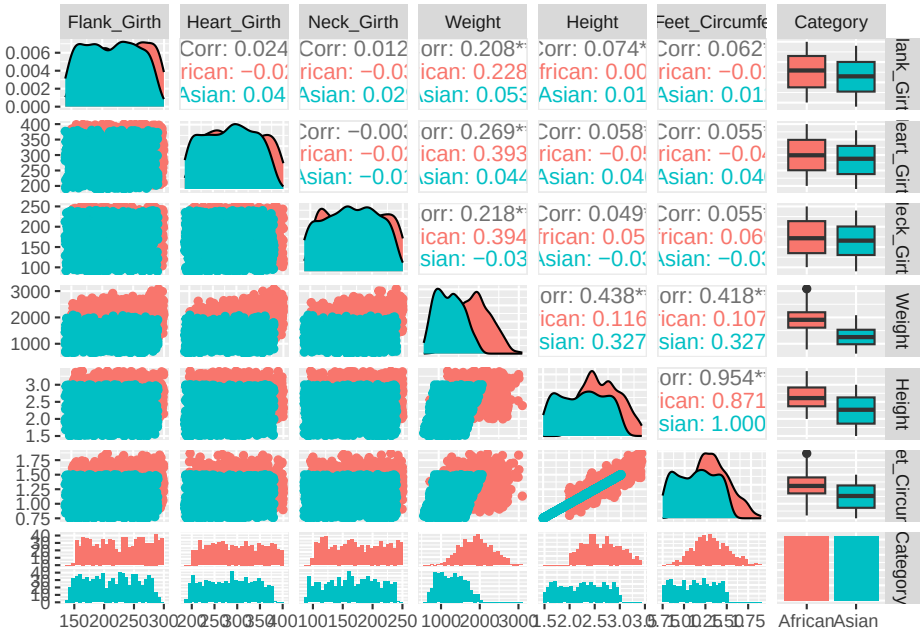
With only numeric variables

```
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))
```



Bibliography

Mike FC, Trevor L Davis, and ggplot2 authors. *ggpattern: 'ggplot2' Pattern Geoms*, 2022. URL https://CRAN.R-project.org/package=ggpattern. R package version 1.0.1.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.