DATA VISUALIZATION WITH GGPLOT2

Graphics of Large Data

Defining largeness

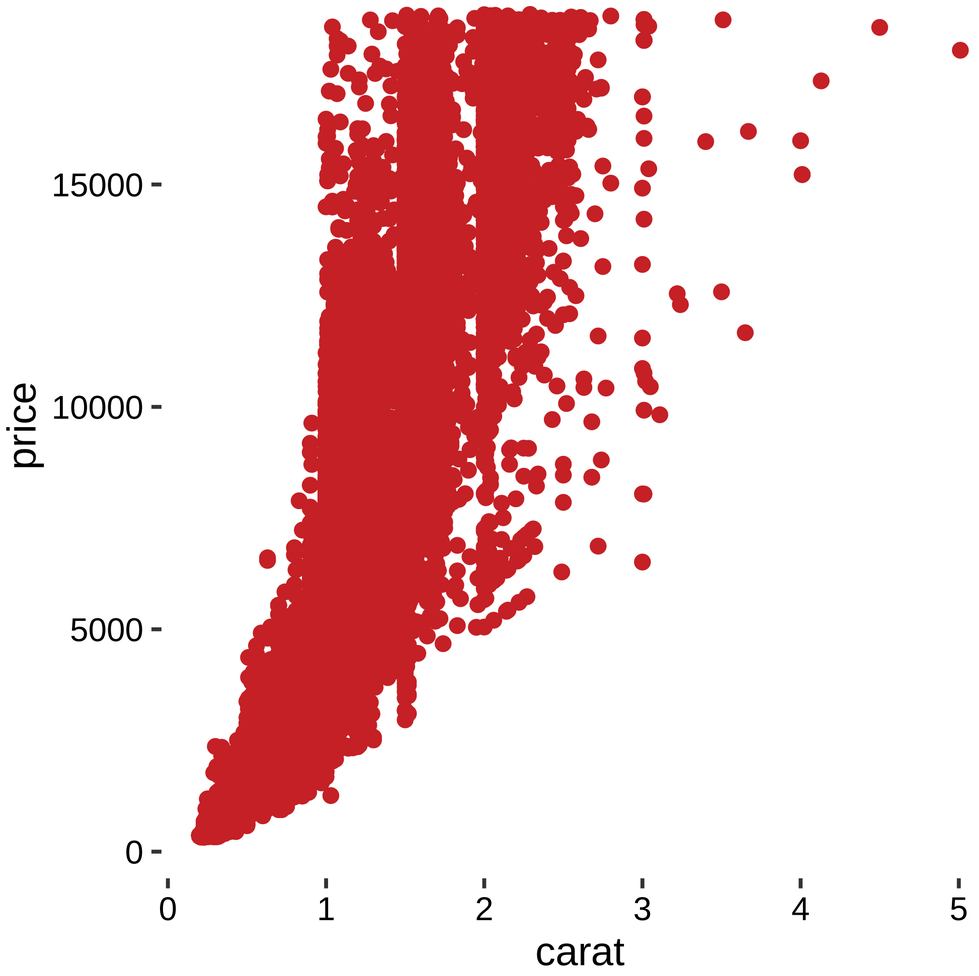
- Many observations
  - High-resolution time series
  - Large surveys
  - Website analytics
- Many variables
  - Multidimensional data
- Combination

Many observations

```
> dim(diamonds)
[1] 53940    10

> head(diamonds)
  carat       cut color clarity depth table price    x    y    z
1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

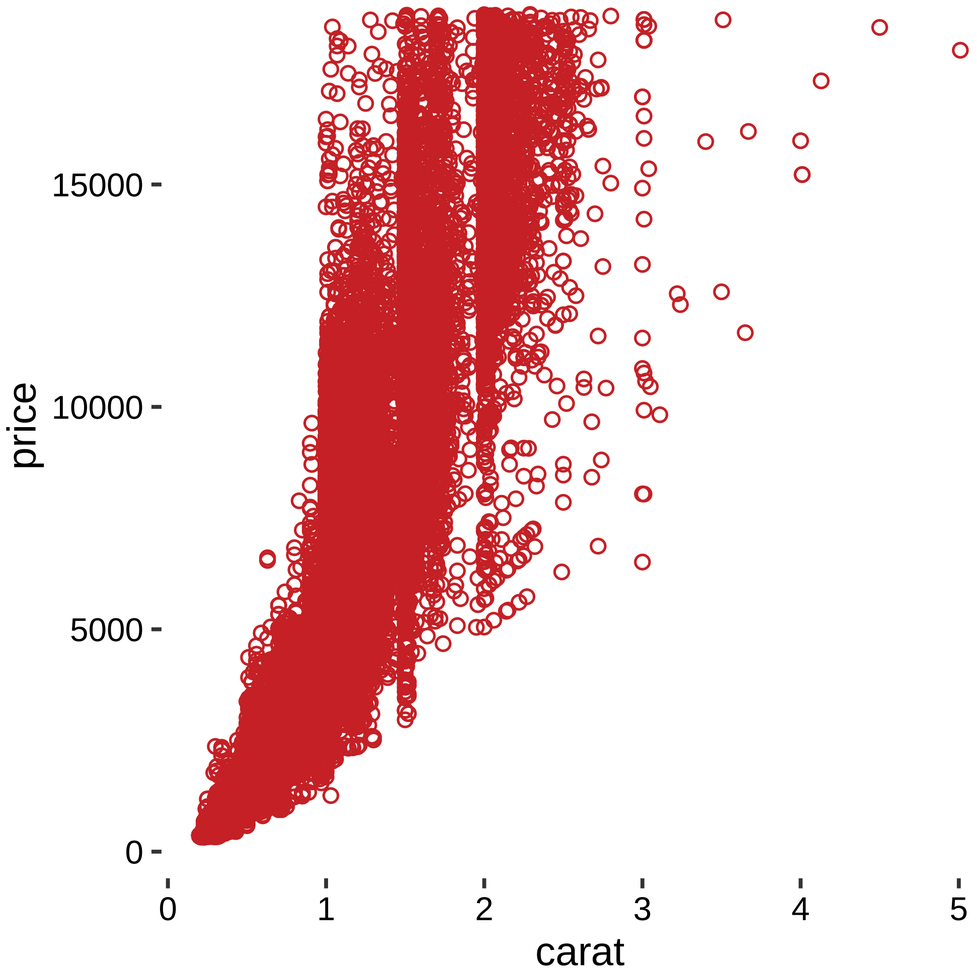
Scatter plot

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point() # simplified
```



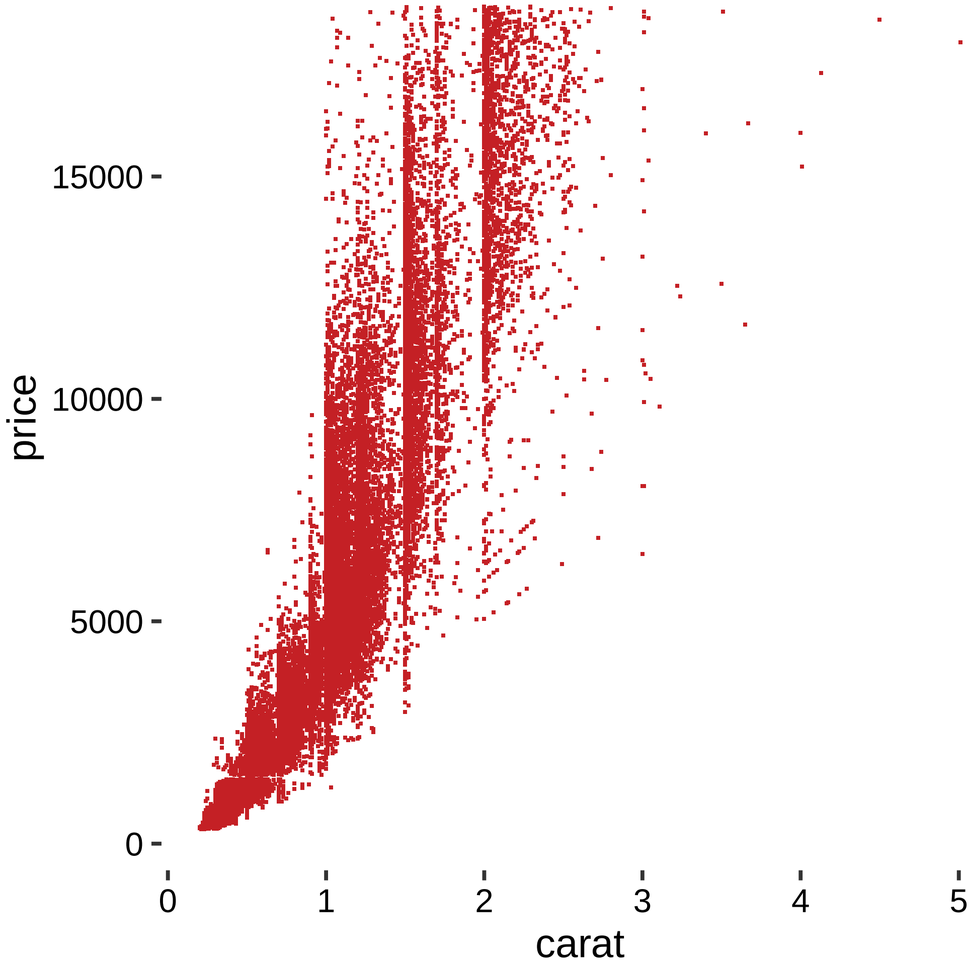
Adjust symbol

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(shape = 1)
```



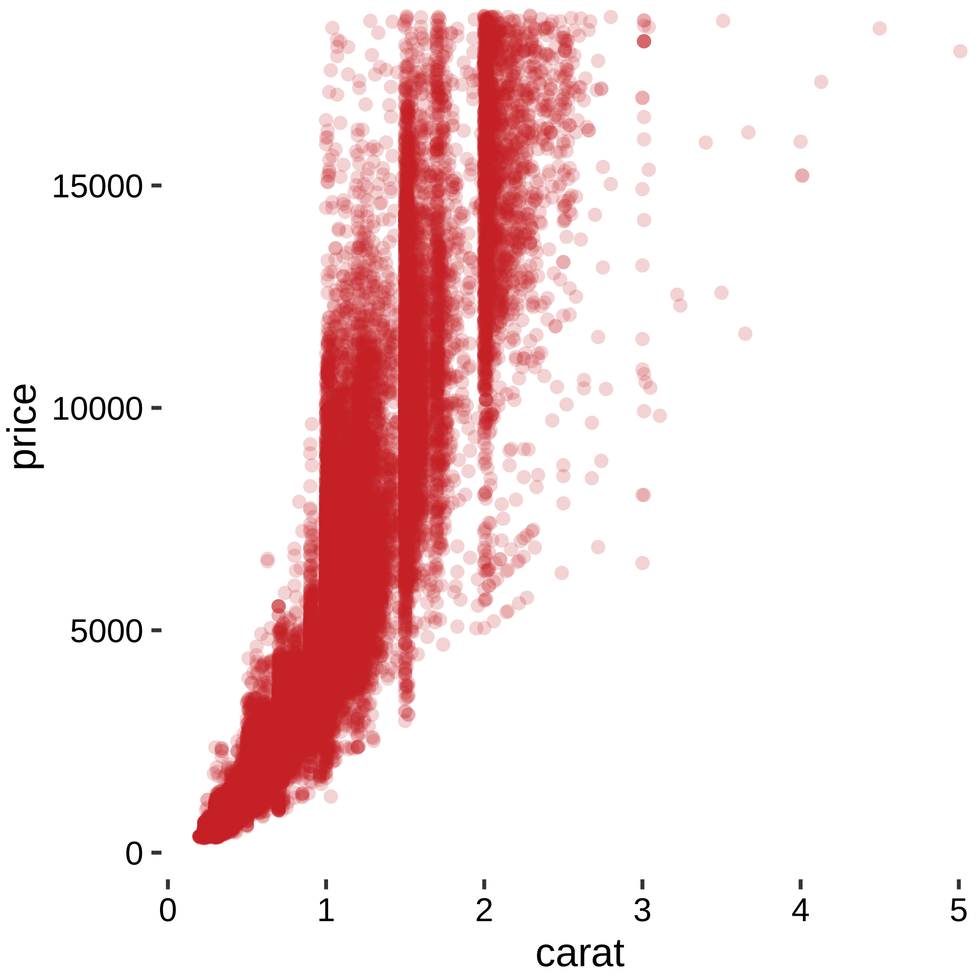
Adjust size

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(shape = ".")
```



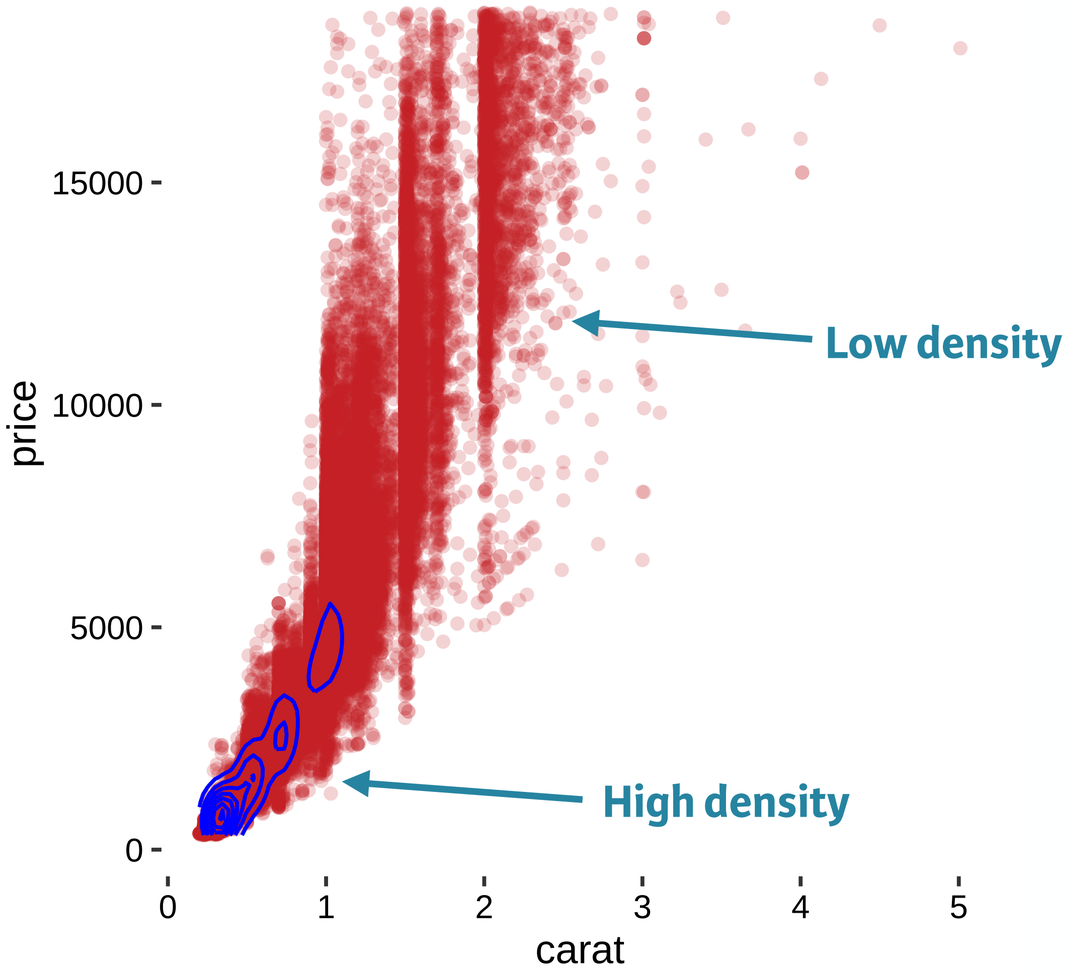
Adjust opacity

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(shape = 16, alpha = 0.2)
```

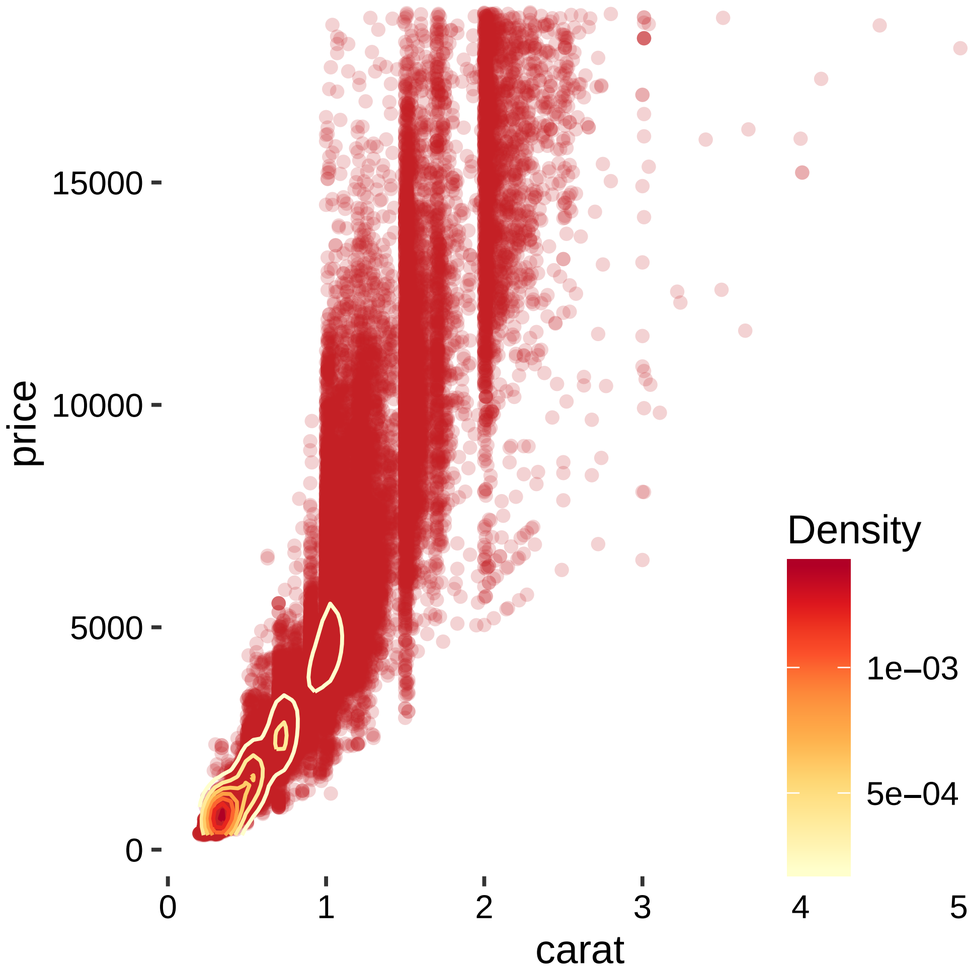


2D density

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(shape = 16, alpha = 0.2) +
    stat_density2d(col = "blue")
```

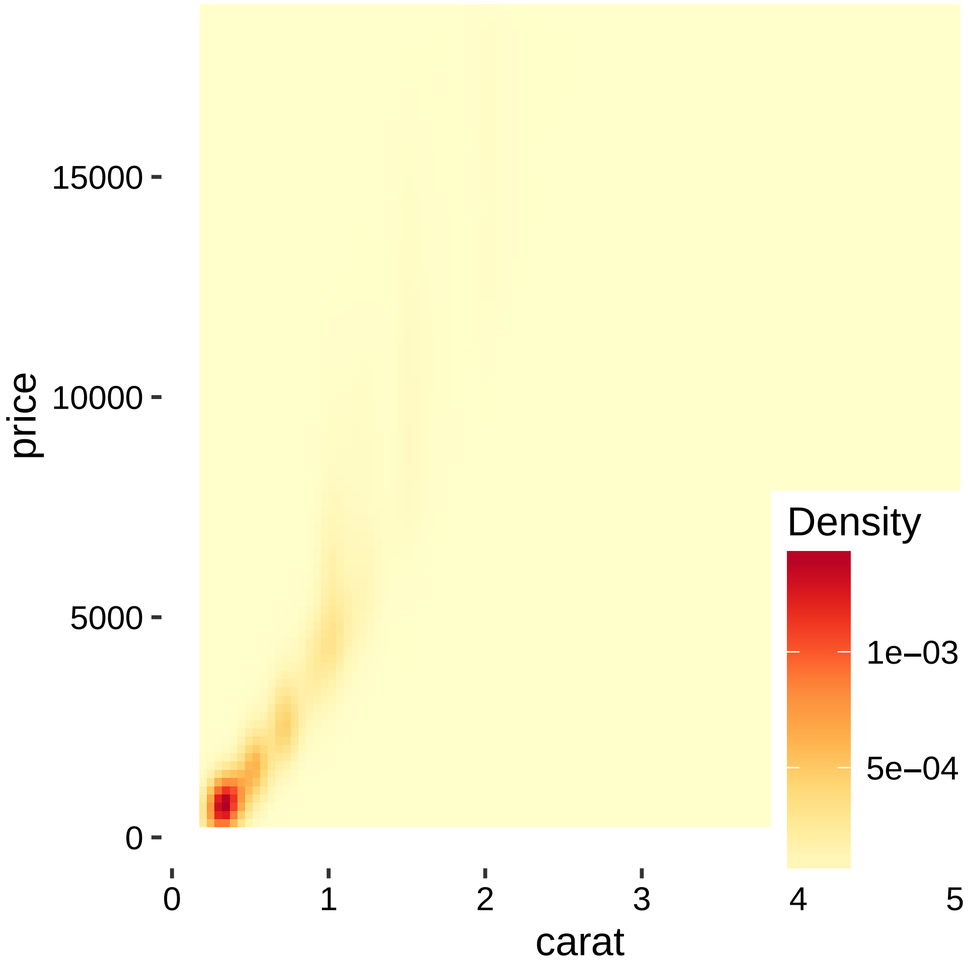


2D density (2)



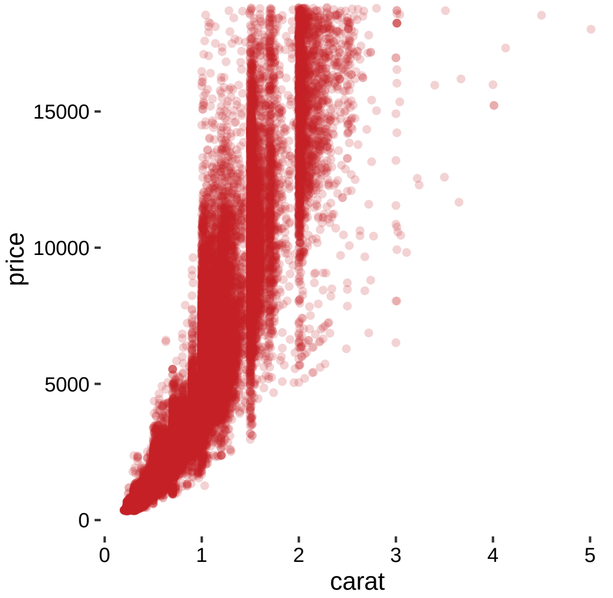

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_point(shape = 16, alpha = 0.2) +
    stat_density2d(aes(colour=..level..))
```

2D density (3)

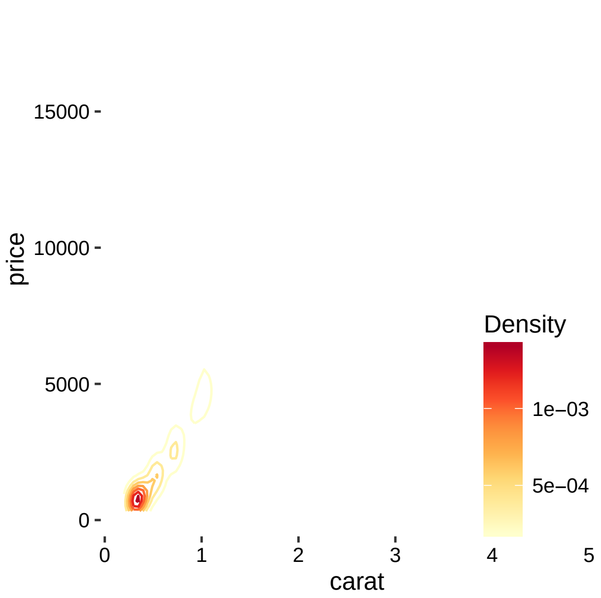



```
> ggplot(diamonds, aes(x = carat, y = price)) +
    stat_density2d(geom = "tile",
                   aes(fill = ..density..),
                   contour = FALSE)
```

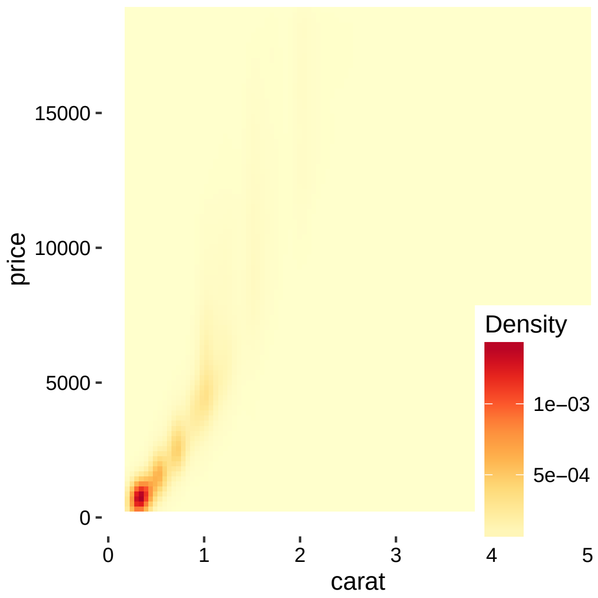
Comparison



Opacity

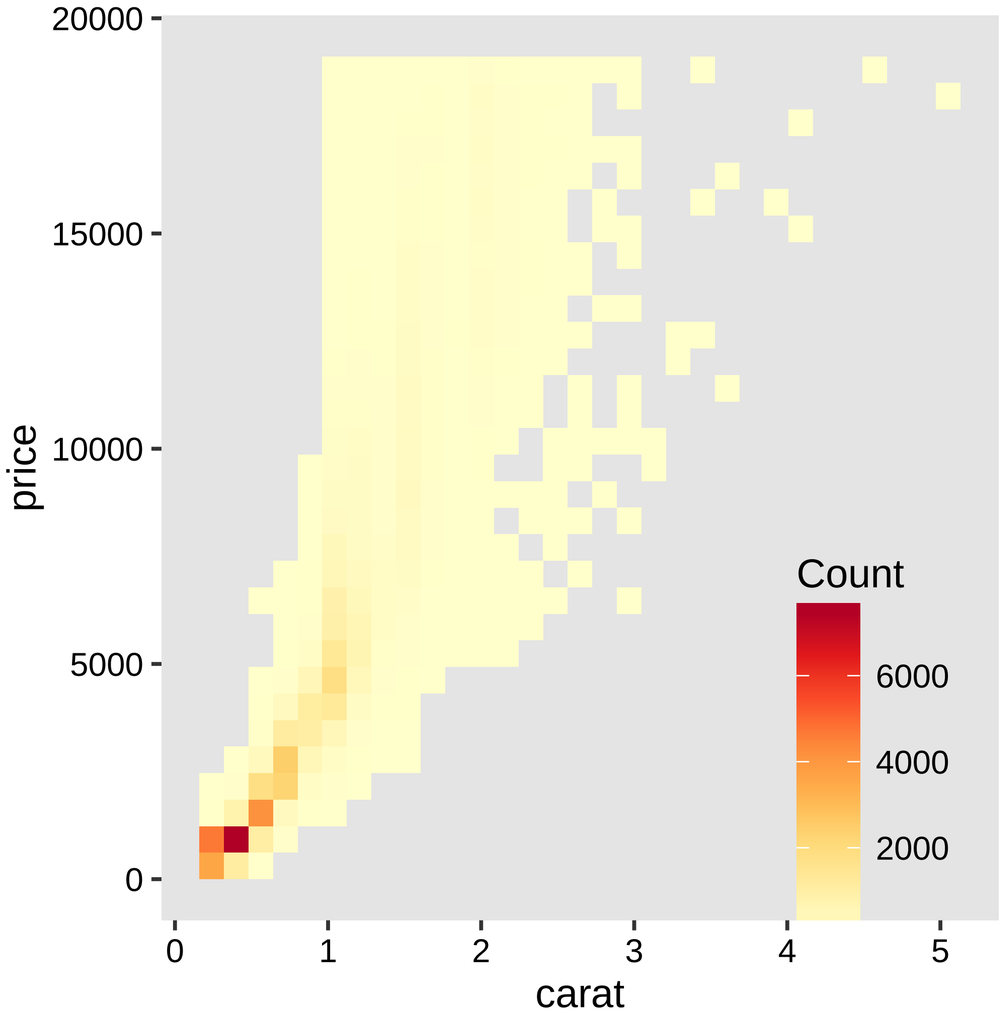


colored contours



heatmap

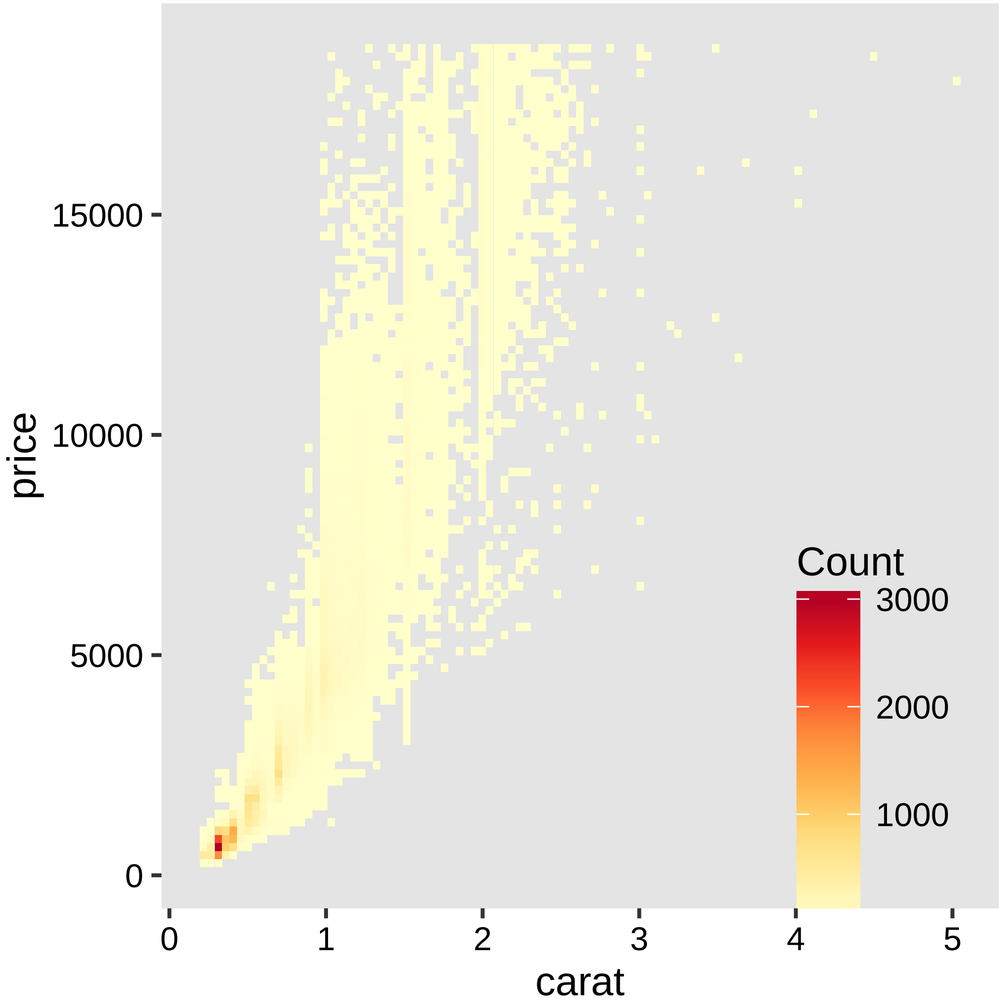
Binning (1)




```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_bin2d()
```

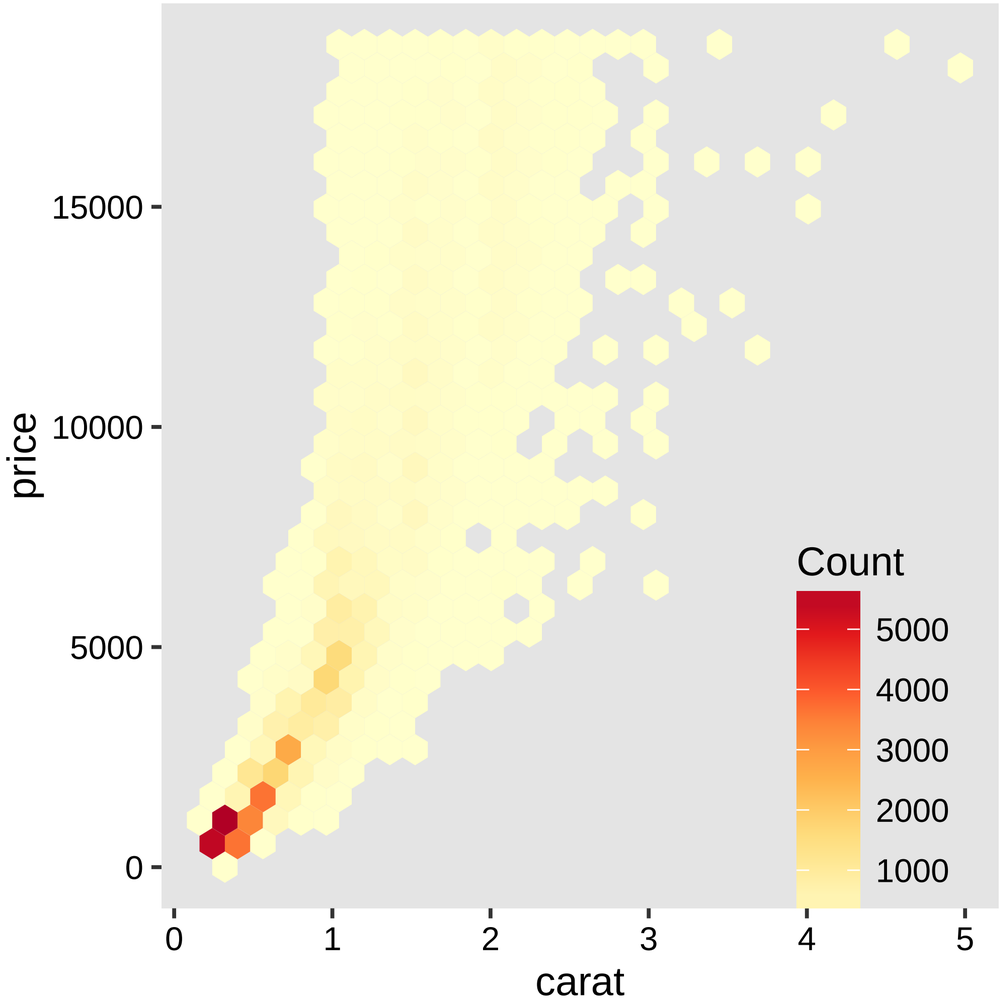
Binning (2)

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_bin2d(bins = 100)
```



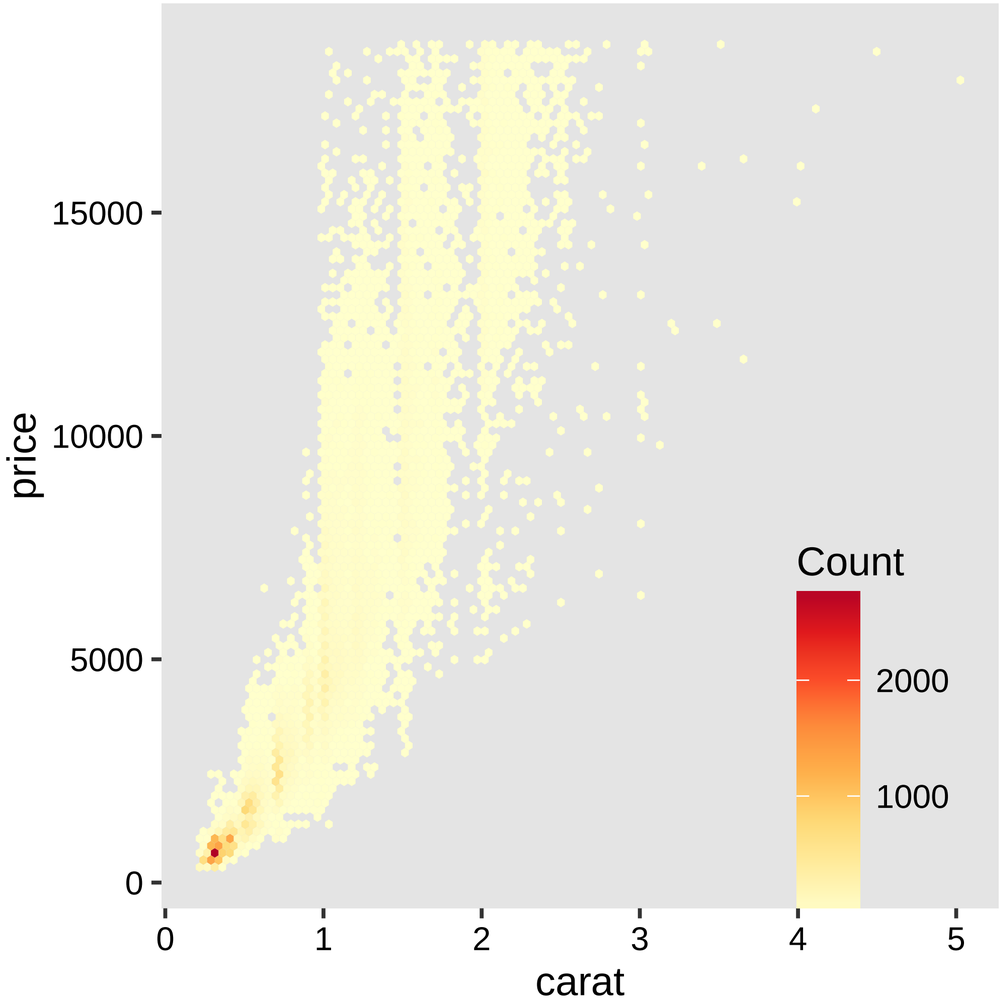
Hex-binning (1)

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_hex()
```



Hex-binning (2)

```
> ggplot(diamonds, aes(x = carat, y = price)) +
    geom_hex(bins = 100)
```



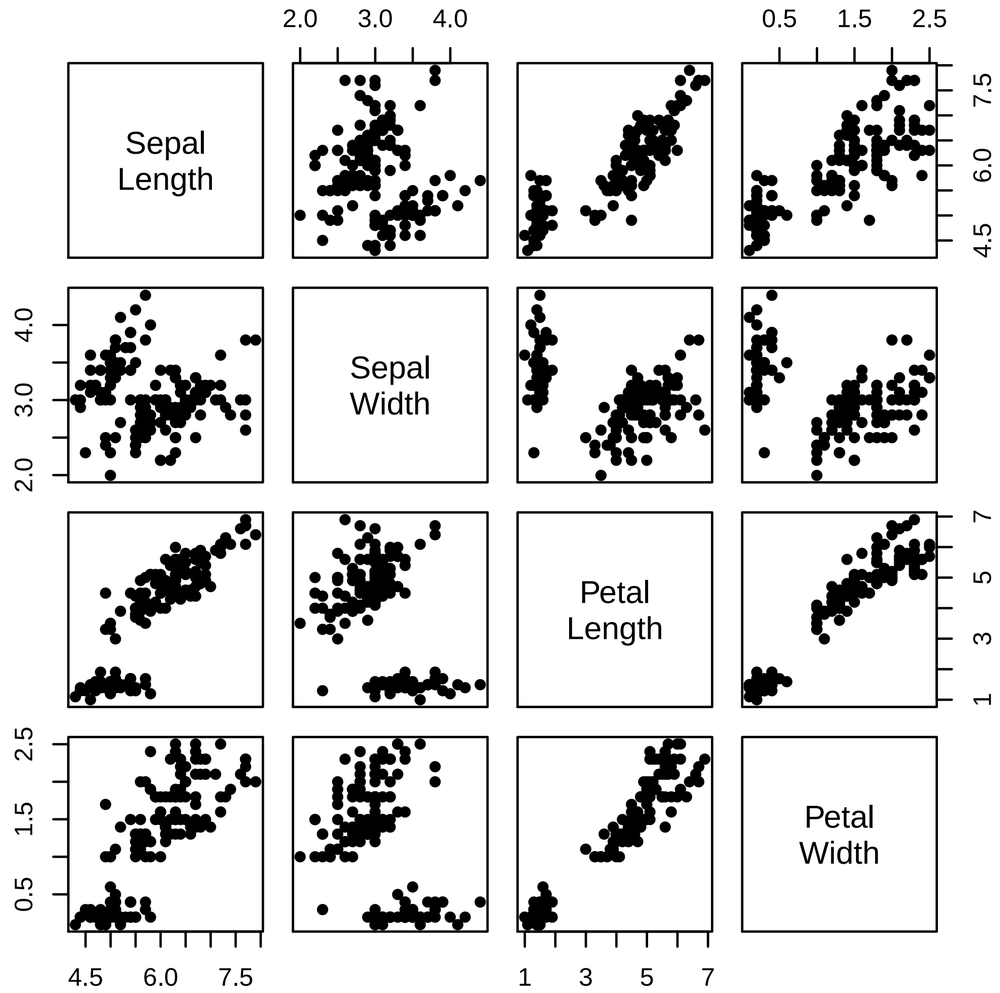
Many observations

- Reducing over-plotting
- Reducing amount of information that is plotted
- Aggregates

Many variables

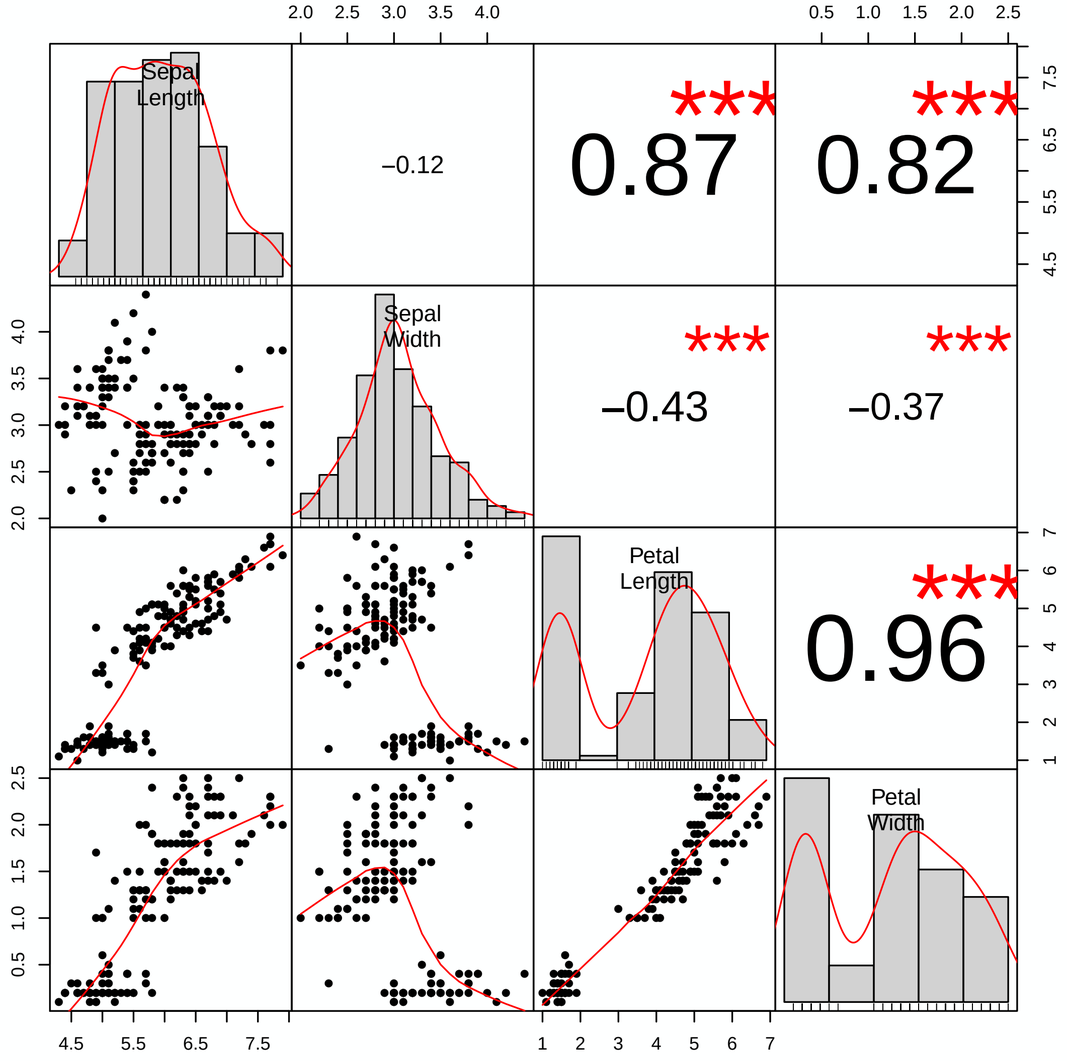
- Multi-variate or high-dimensional data
- Data reduction methods
- Previously: facets
- Two plot types

SPLOM - Scatter PLOt Matrix



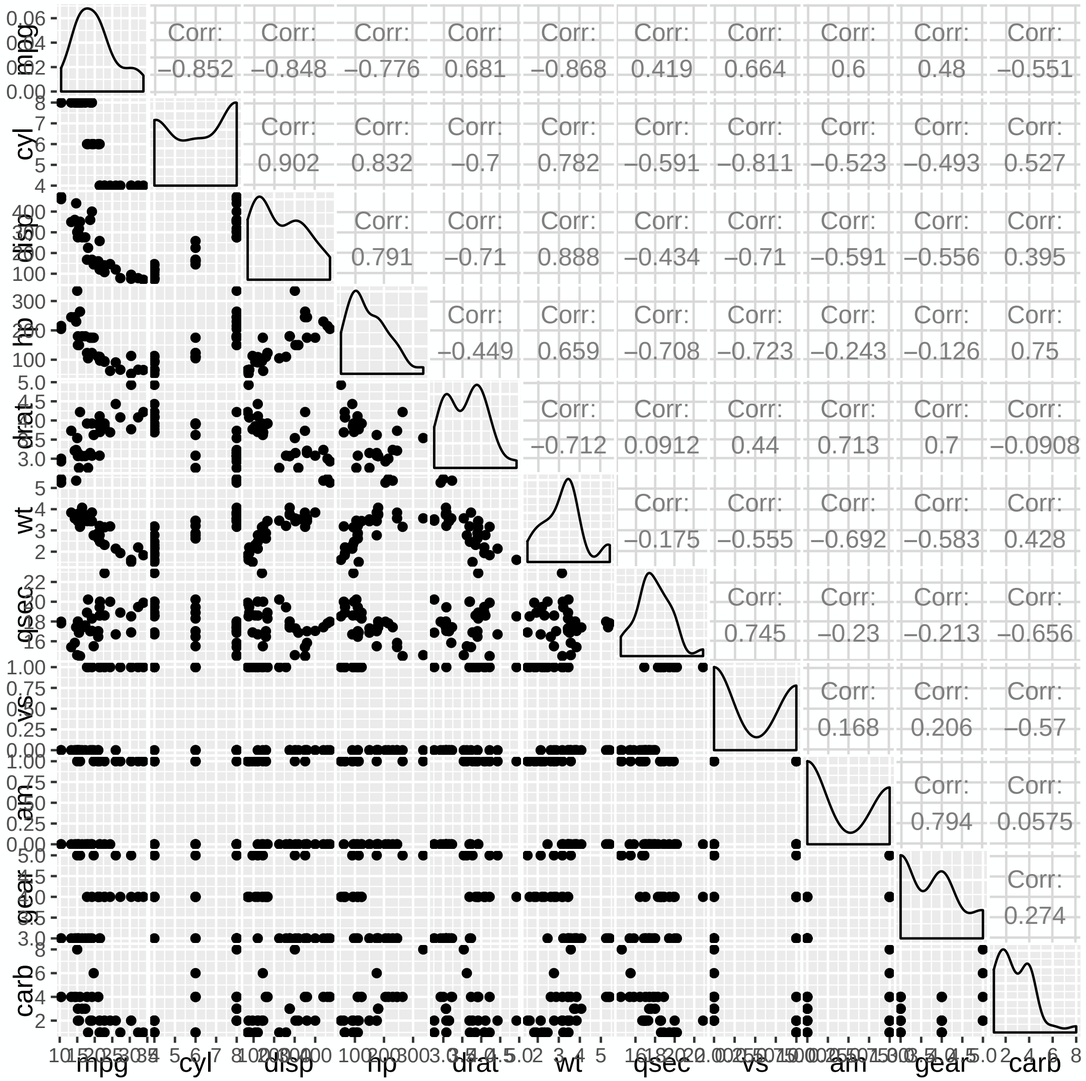

```
> pairs(iris[-5])
```

Correlation matrix



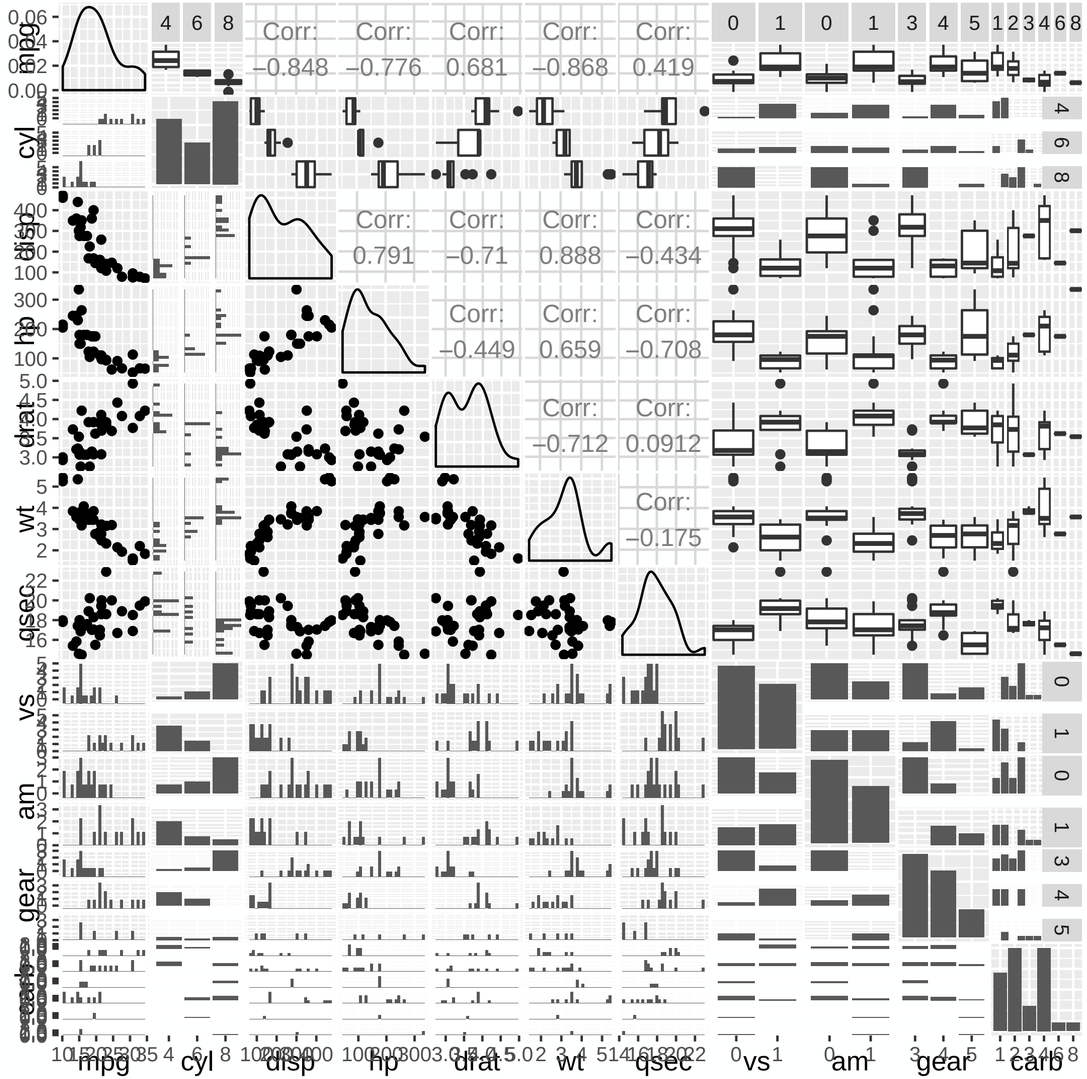

```
> library(PerformanceAnalytics)
> chart.Correlation(iris[-5])
```

ggAlly::ggpairs()



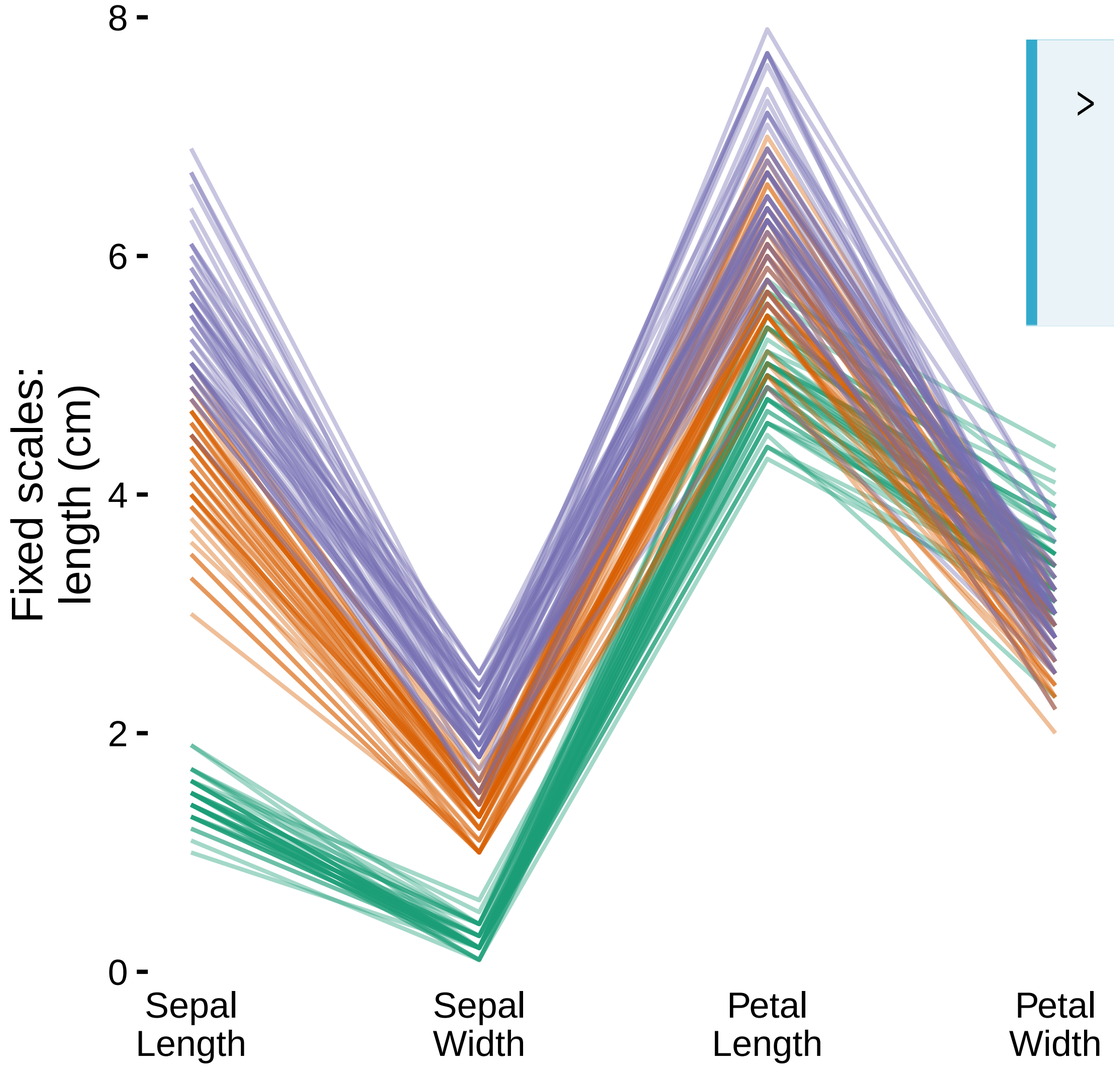

```
> library(GGally)
> ggpairs(mtcars)
```

ggAlly::ggpairs()




```
> library(GGally)

> # make columns categorical
> ... # omitted

> ggpairs(mtcars)
```

Parallel coordinate plot




```
> ggparcoord(iris, columns = 1:4,
             groupColumn = 5,
             scale = "globalminmax",
             order = "anyClass", alphaLines = 0.4)
```

DATA VISUALIZATION WITH GGPLOT2

Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

Ternary Plots

Ternary plot

- Triangle plot
- Compositional trivariate data
- Add up to 100%
- Typical example: soil composition
  - Sand
  - Silt
  - Clay

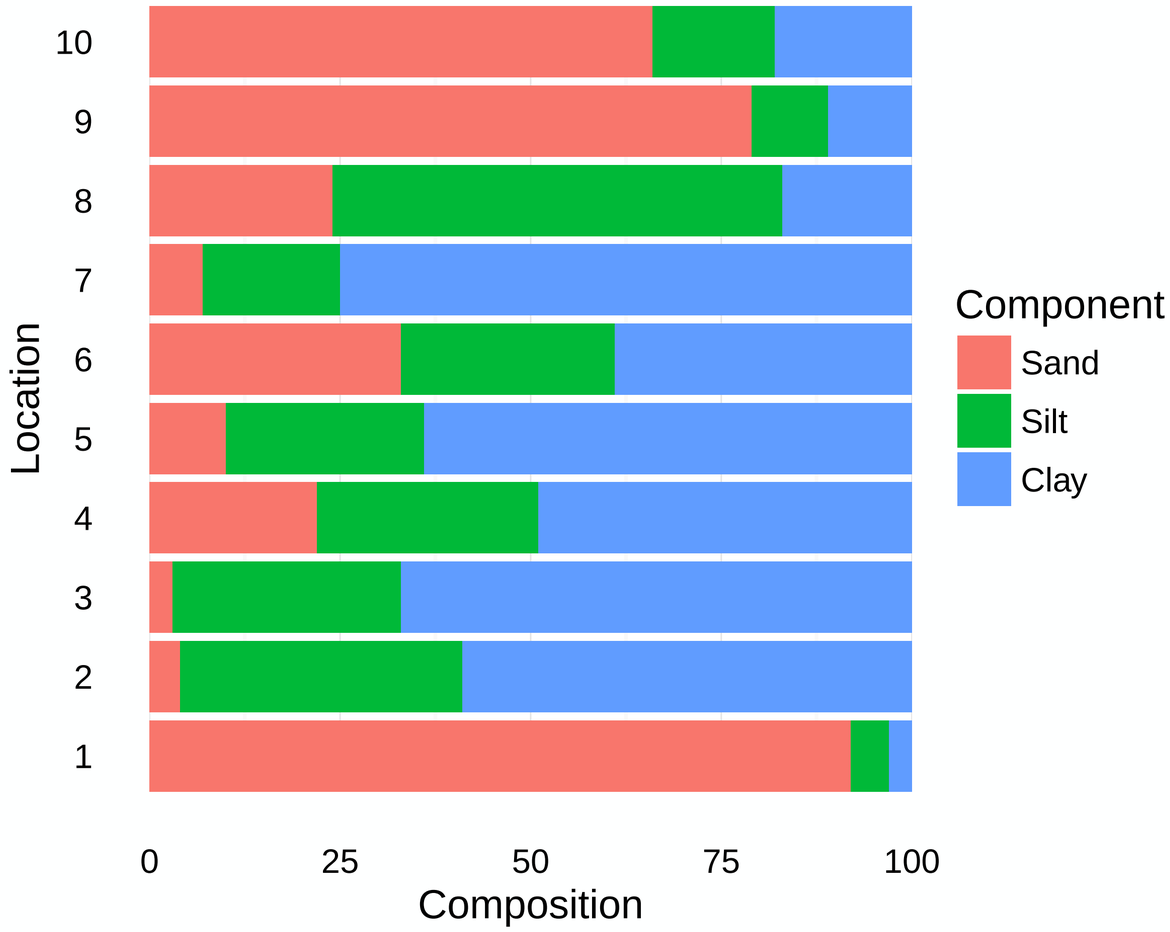
africa

```
> dim(africa)
[1] 40093     3

> head(africa)
      Sand Silt Clay
18100   24   12   64
18103   36   14   50
18113   56   18   26
18115   52   21   27
18142   65    3   32
18158   43   14   43
```

africa.sample

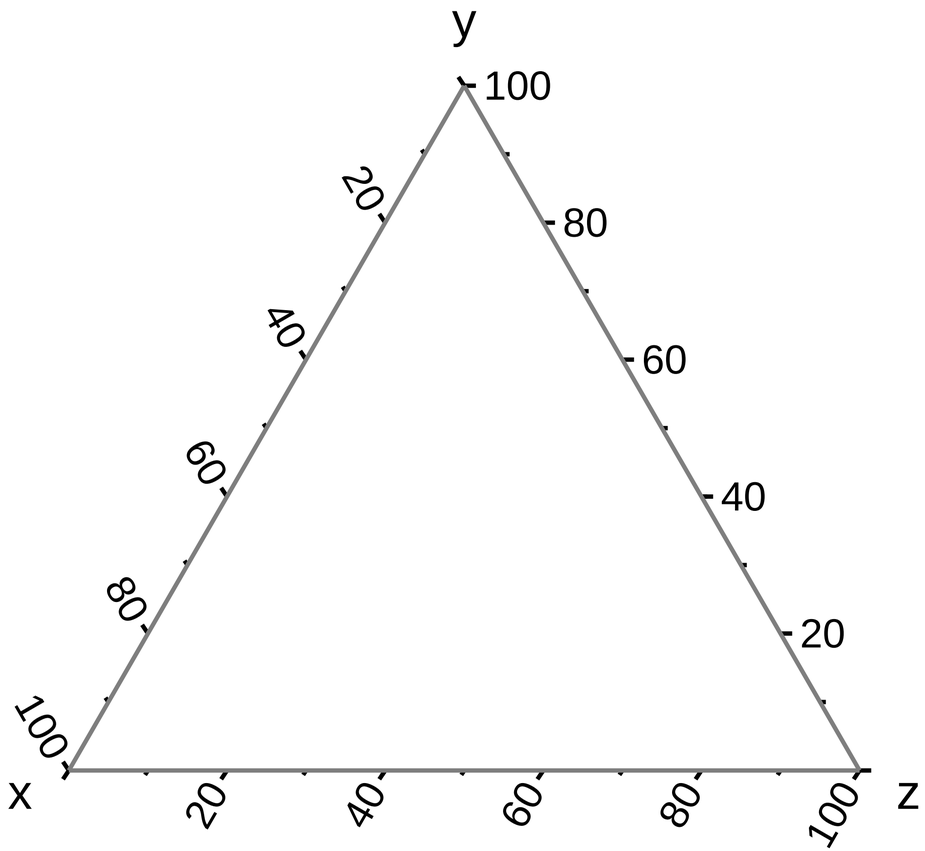
```
> africa.sample
       Sand Silt Clay ID
23678    92    5    3  1
27995     4   37   59  2
35263     3   30   67  3
41380    22   29   49  4
47814    10   26   64  5
62640    33   28   39  6
67707     7   18   75  7
80337    24   59   17  8
84911    79   10   11  9
85014    66   16   18 10
```

Stacked bar plot

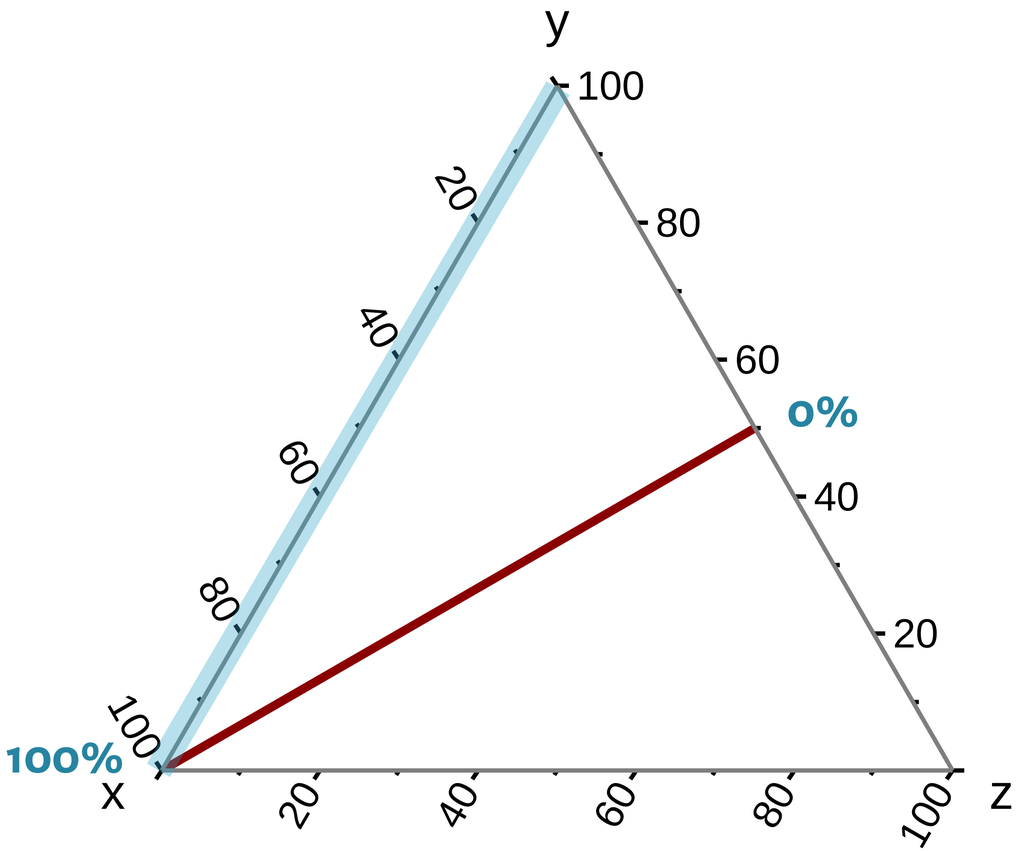



```
> africa.sample
      Sand Silt Clay ID
23678   92    5    3  1
27995    4   37   59  2
35263    3   30   67  3
41380   22   29   49  4
...
```

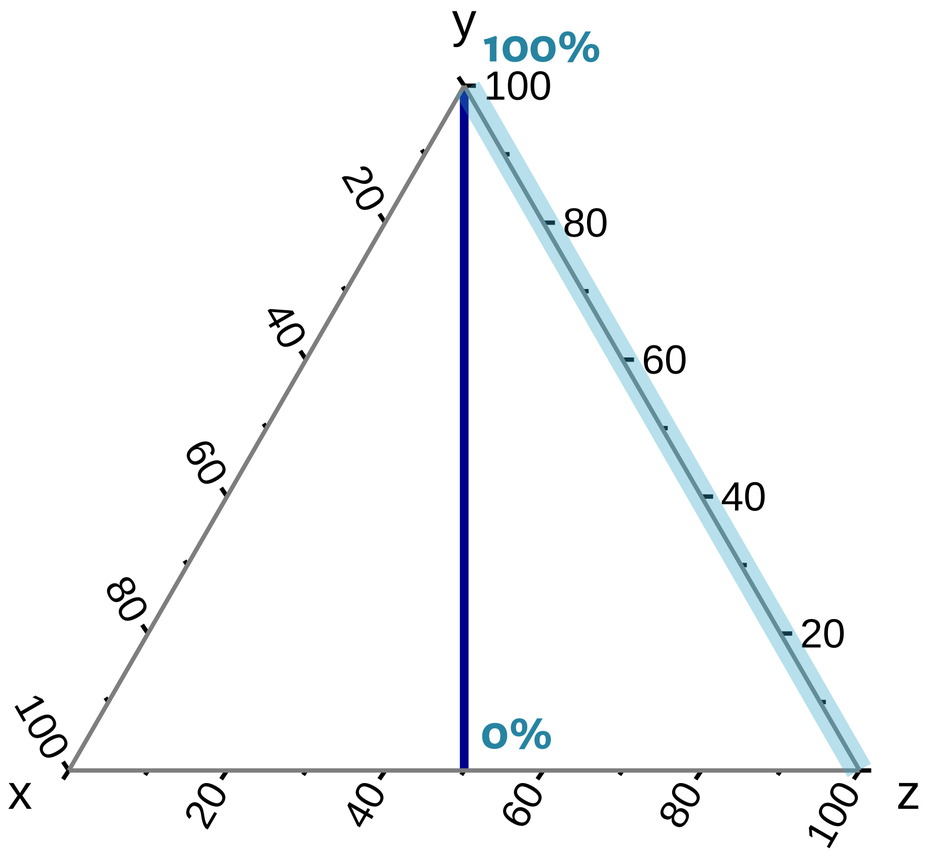
Ternary plot



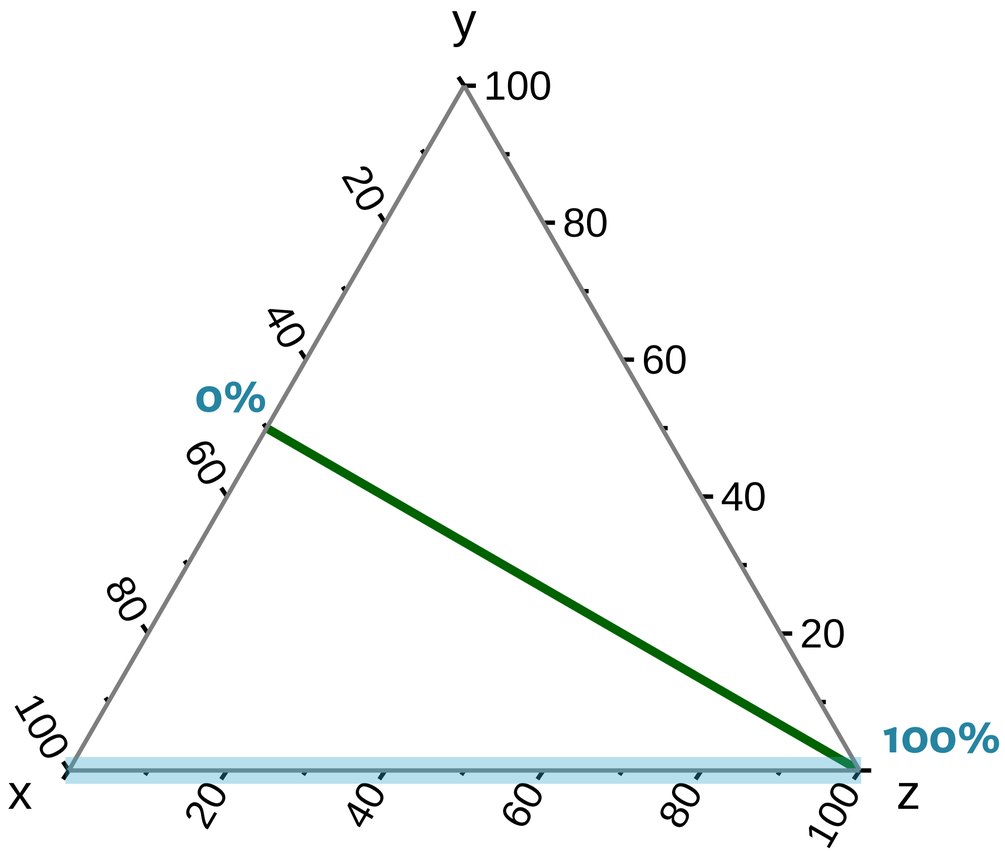
Ternary plot - X



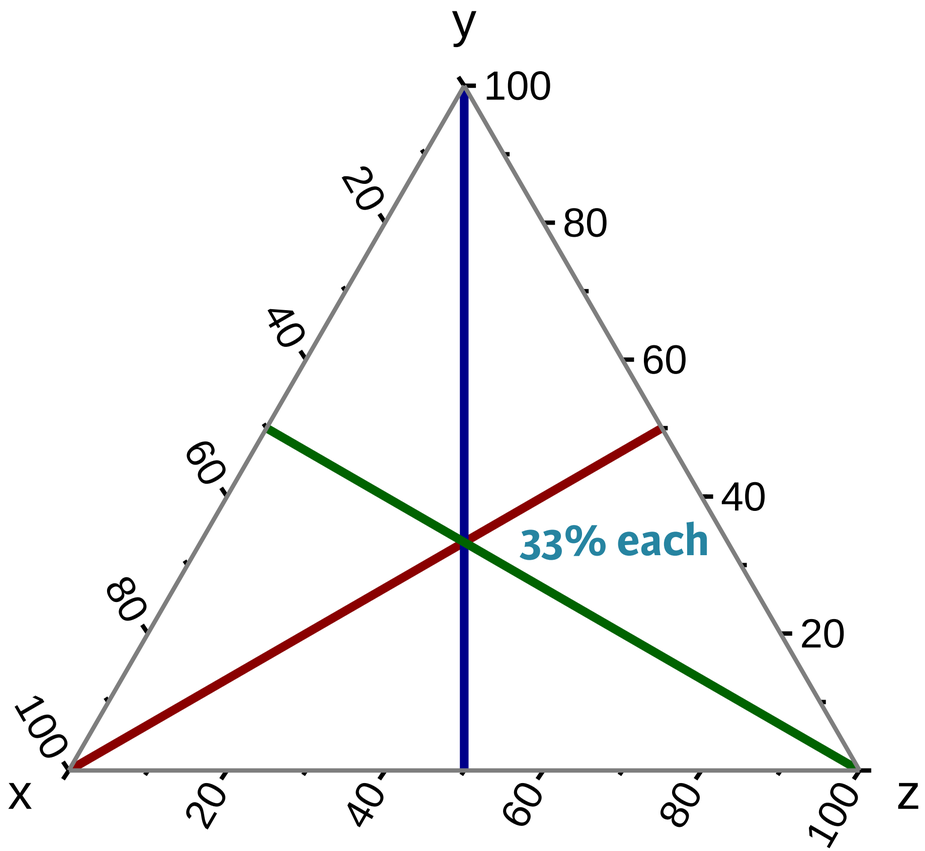
Ternary plot - Y



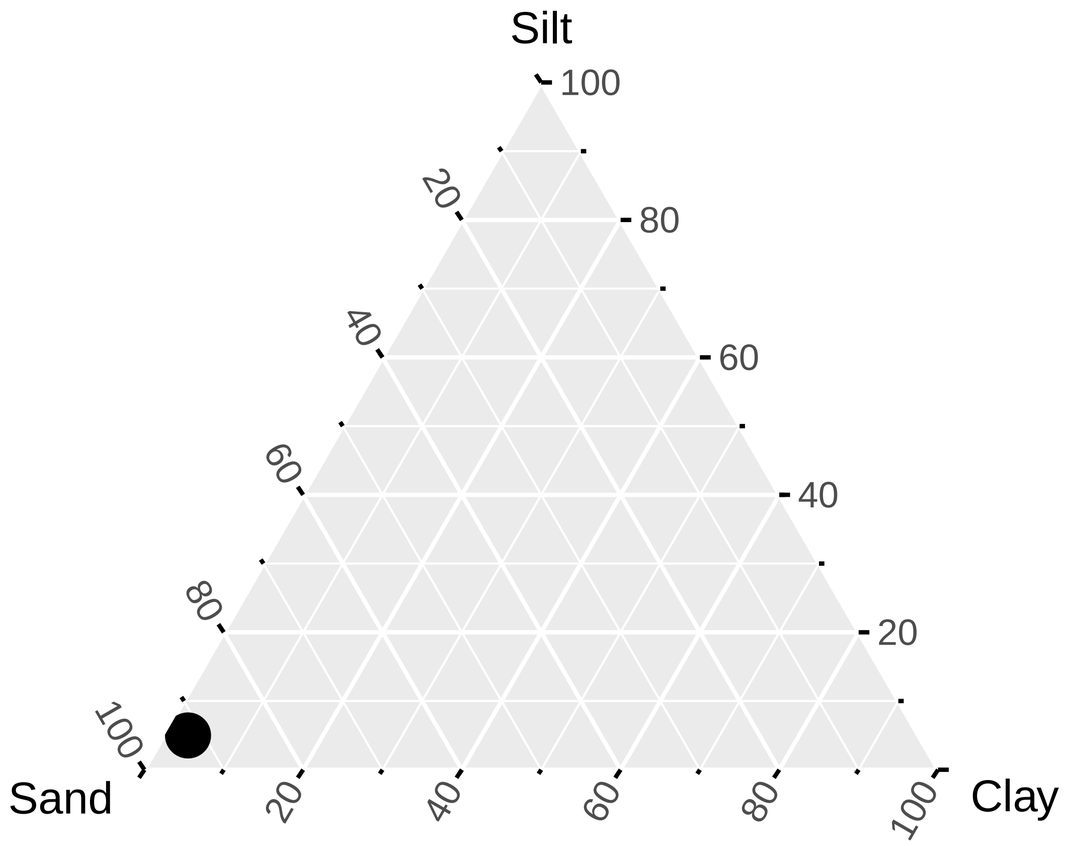
Ternary plot - Z



Ternary plot - 3 axes

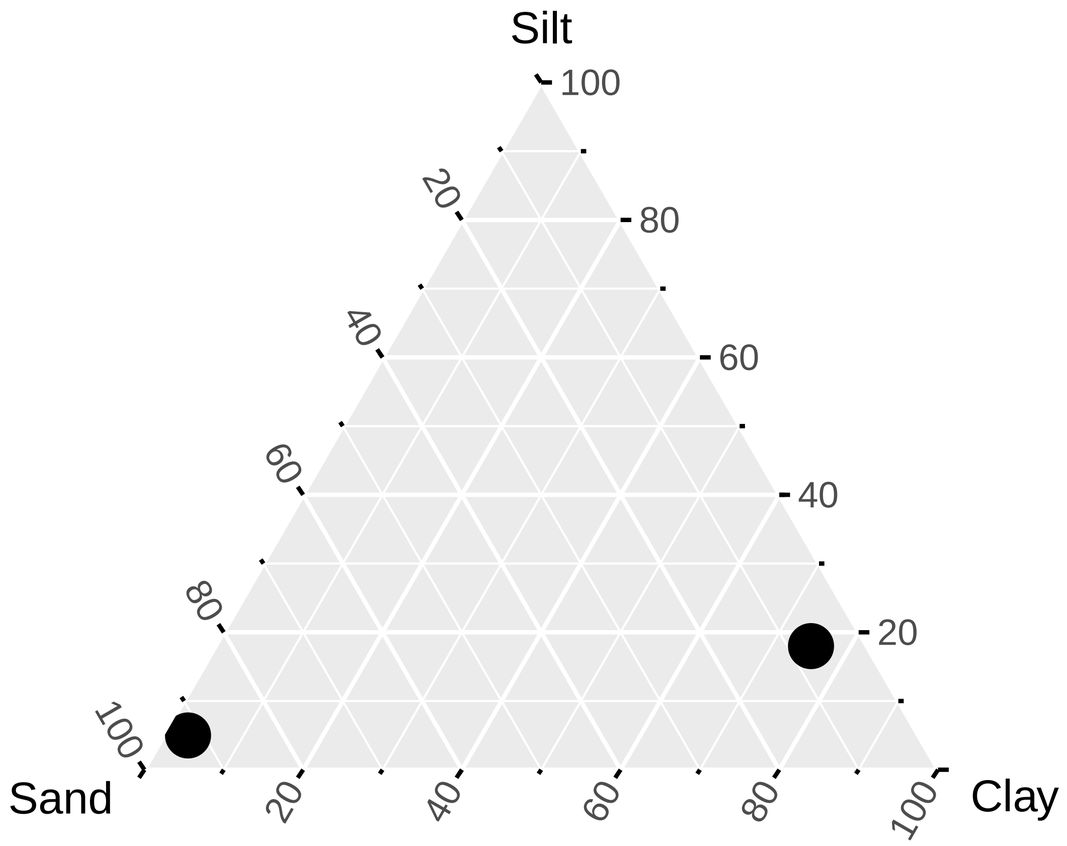


Location 1



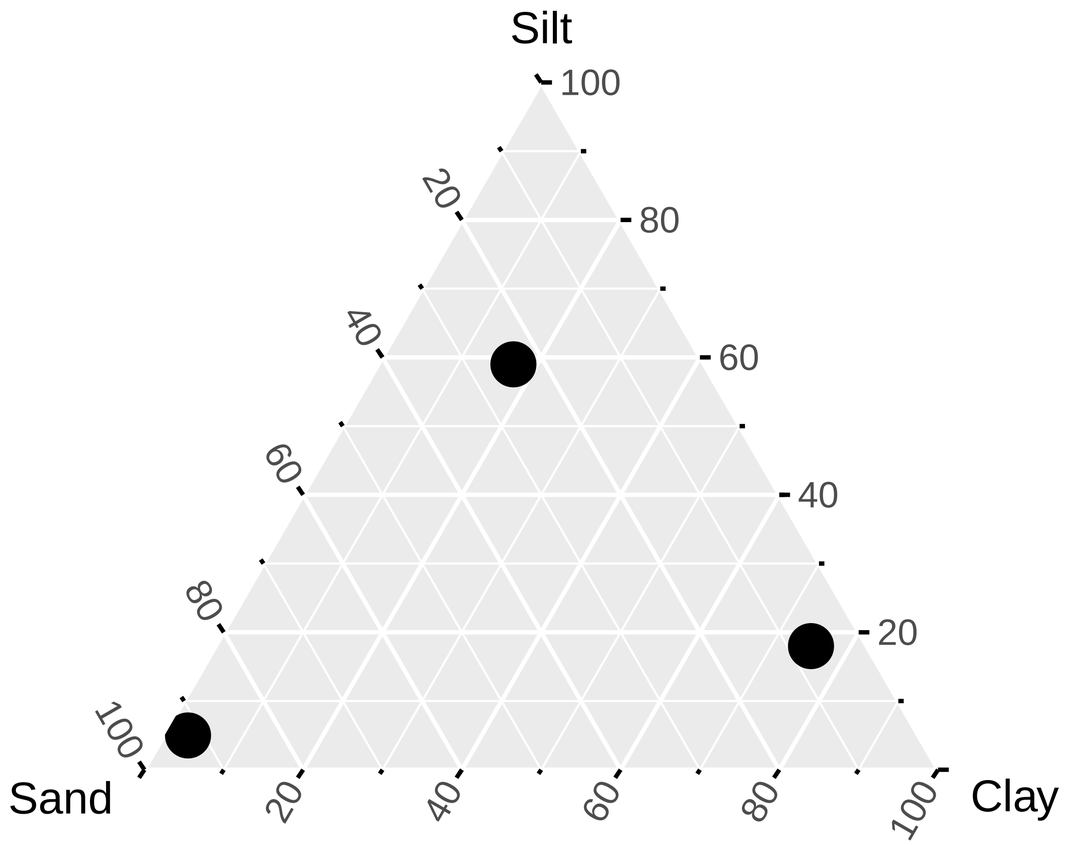

```
> africa.sample
       Sand Silt Clay ID
23678    92    5    3  1
27995     4   37   59  2
35263     3   30   67  3
41380    22   29   49  4
...
```

Location 7




```
> africa.sample
      Sand Silt Clay ID
...
62640   33   28   39  6
67707    7   18   75  7
80337   24   59   17  8
...
```

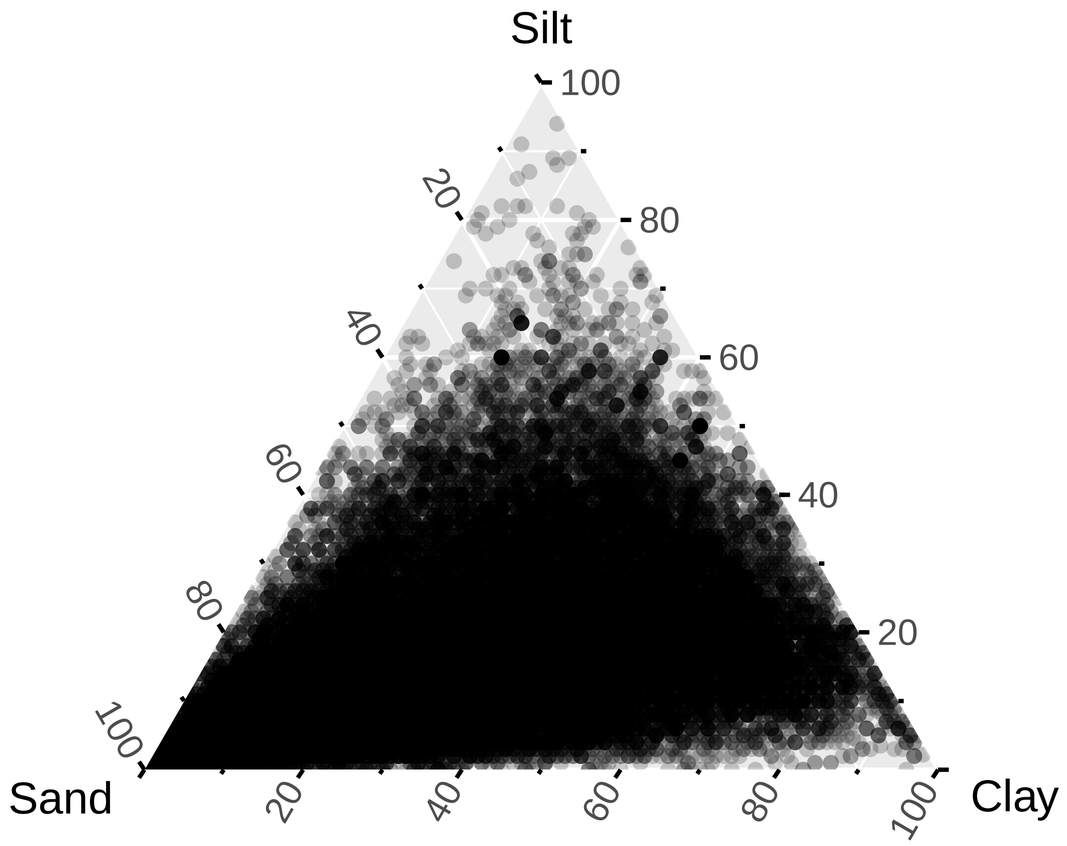
Location 8




```
> africa.sample
      Sand Silt Clay ID
...
62640   33   28   39  6
67707    7   18   75  7
80337   24   59   17  8
...
```

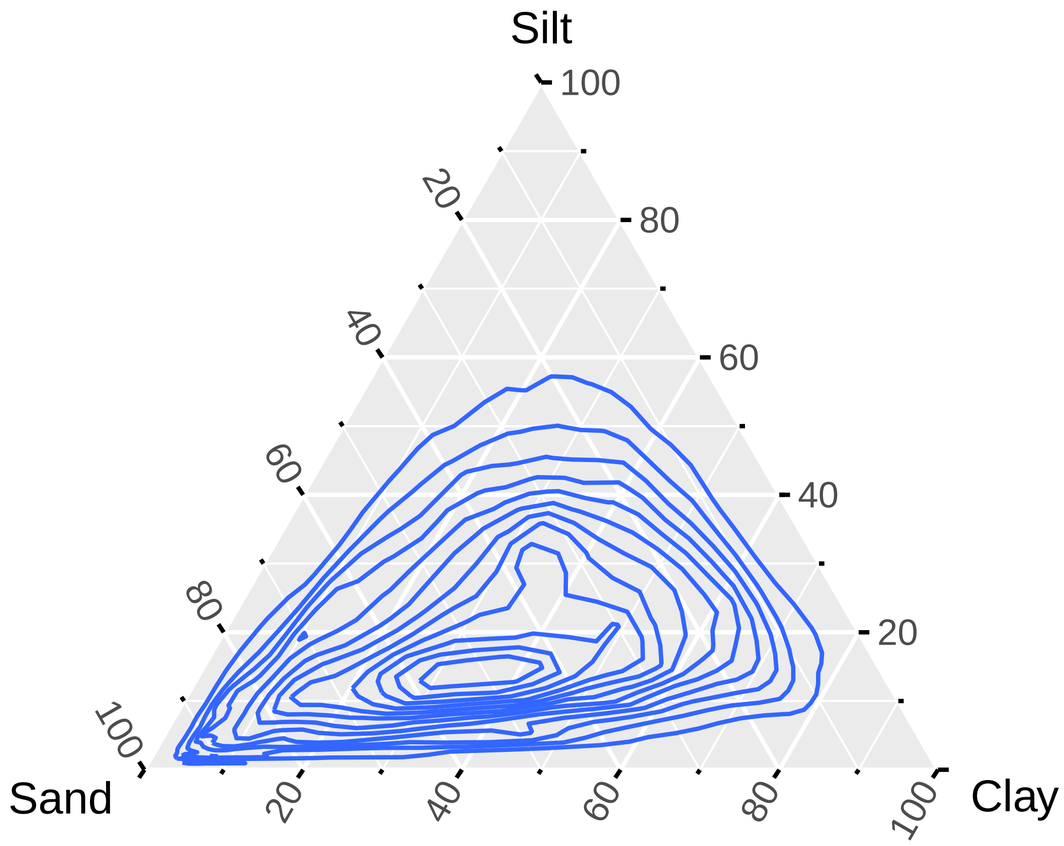
All data (1)

```
> library(ggtern)
> ggtern(africa, aes(Sand, Silt, Clay)) +
    geom_point(alpha = 0.2, shape = 16)
```



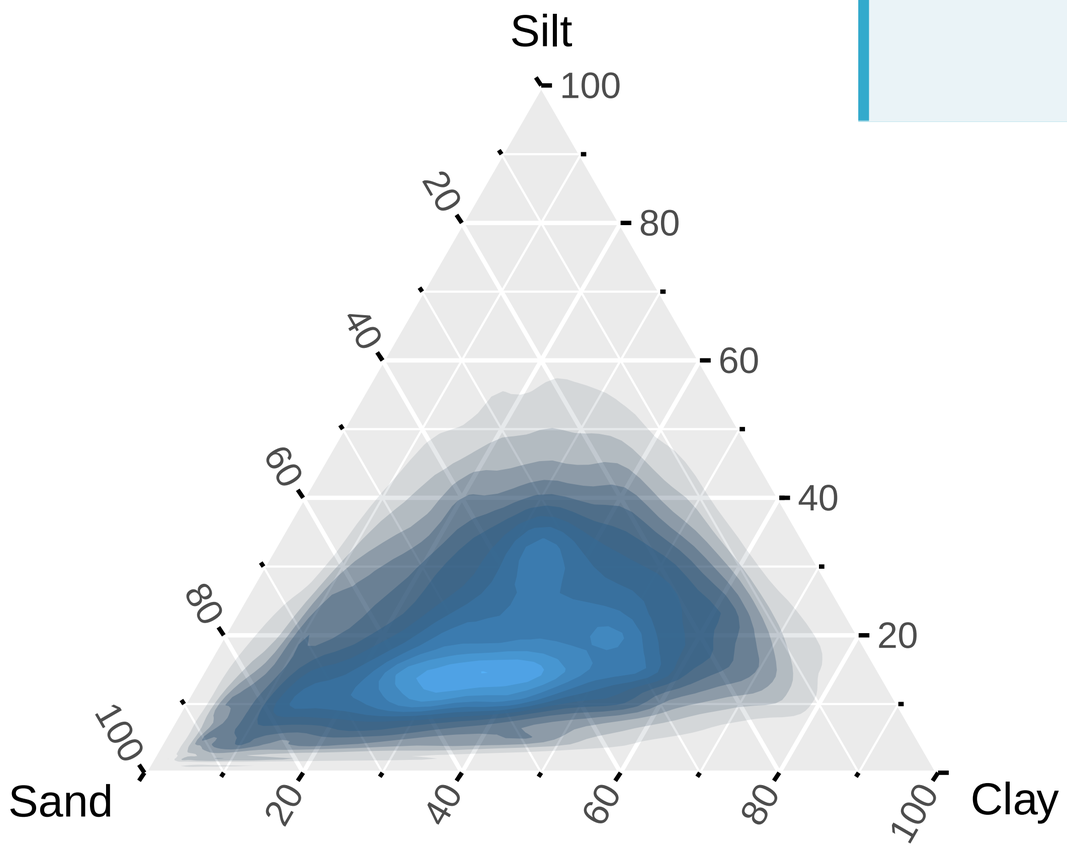
All data (2)

```
> library(ggtern)
> ggtern(africa, aes(Sand, Silt, Clay)) +
    geom_density_tern()
```



All data (3)

```
> library(ggtern)
> ggtern(africa, aes(Sand, Silt, Clay)) +
    stat_density_tern(geom = "polygon",
                      n = 200,
                      aes(fill = ..level..,
                          alpha = ..level..))
```



DATA VISUALIZATION WITH GGPLOT2

Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

Network Plots

Blood types

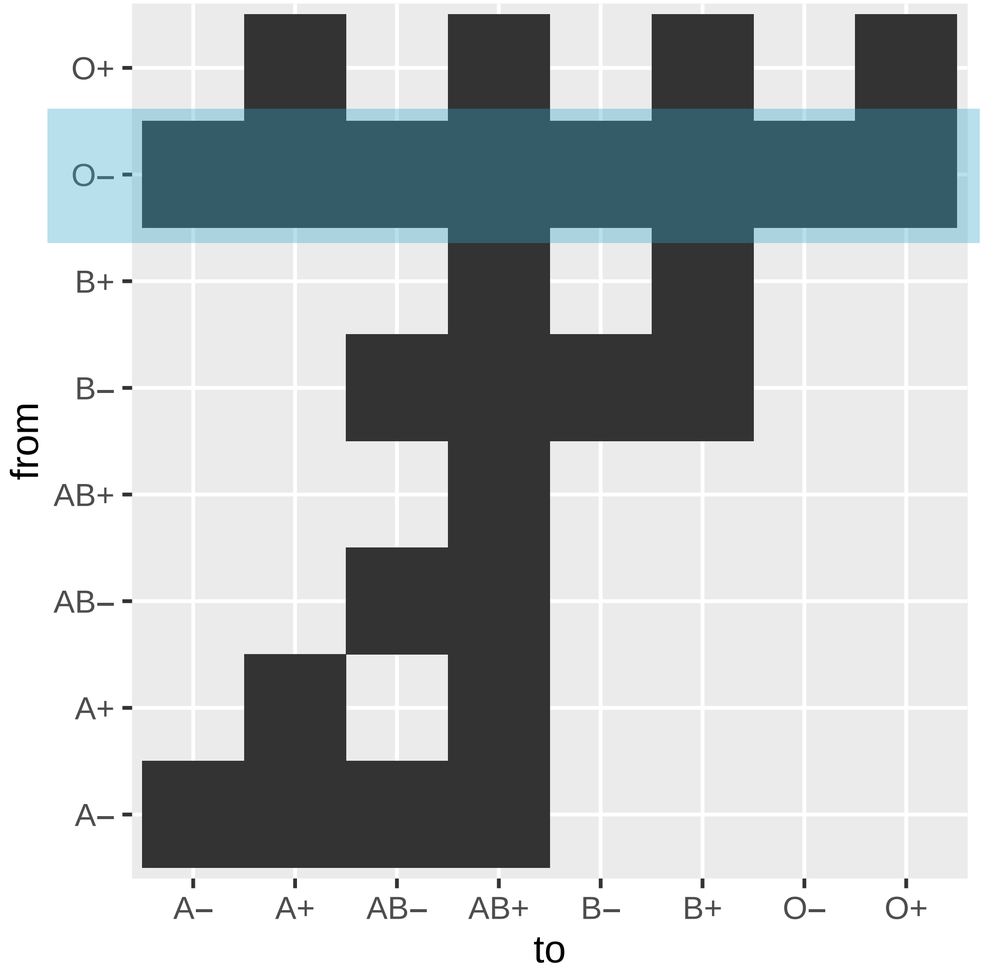
- Four main types
  - A, B, AB, O
- Rho negative or positive
- 8 different types
- Not all compatible
- Which blood type can be used as a donor?

blood_donors

```
> dim(blood_donors)
[1] 27  2

> head(blood_donors)
  from  to
1  AB- AB+
2  AB- AB-
3  AB+ AB+
4   A- AB+
5   A- AB-
6   A-  A+

> levels(blood_donors$from)
[1] "A-"  "A+"  "AB-" "AB+" "B-"  "B+"  "O-"  "O+"
```

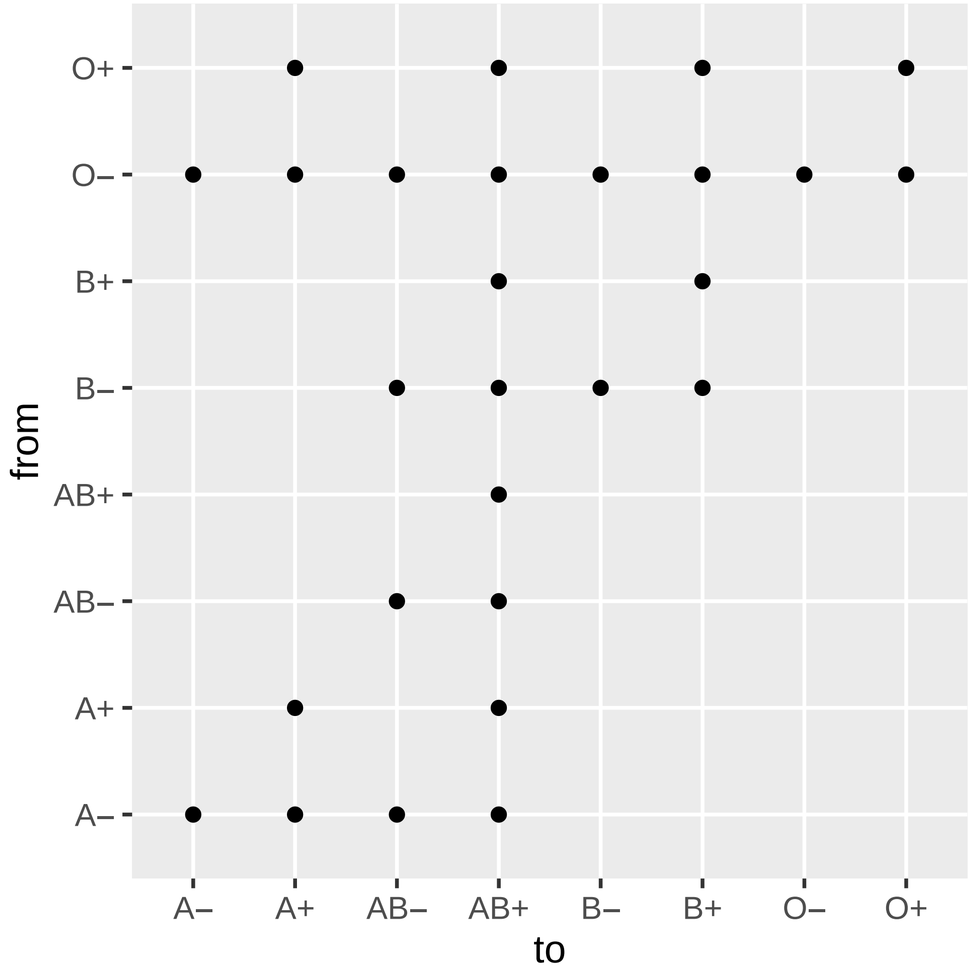
Heatmap

```
> ggplot(Blood_donors, aes(x = to, y = from)) +
    geom_tile()
```

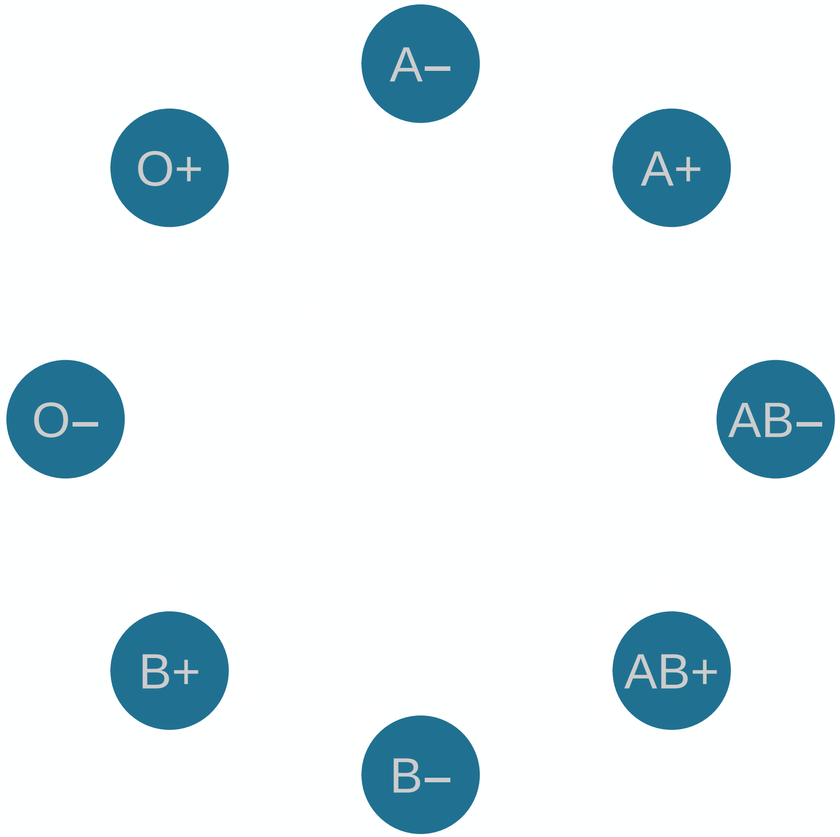


Scatter plot

```
> ggplot(Blood_donors, aes(x = to, y = from)) +
    geom_point()
```

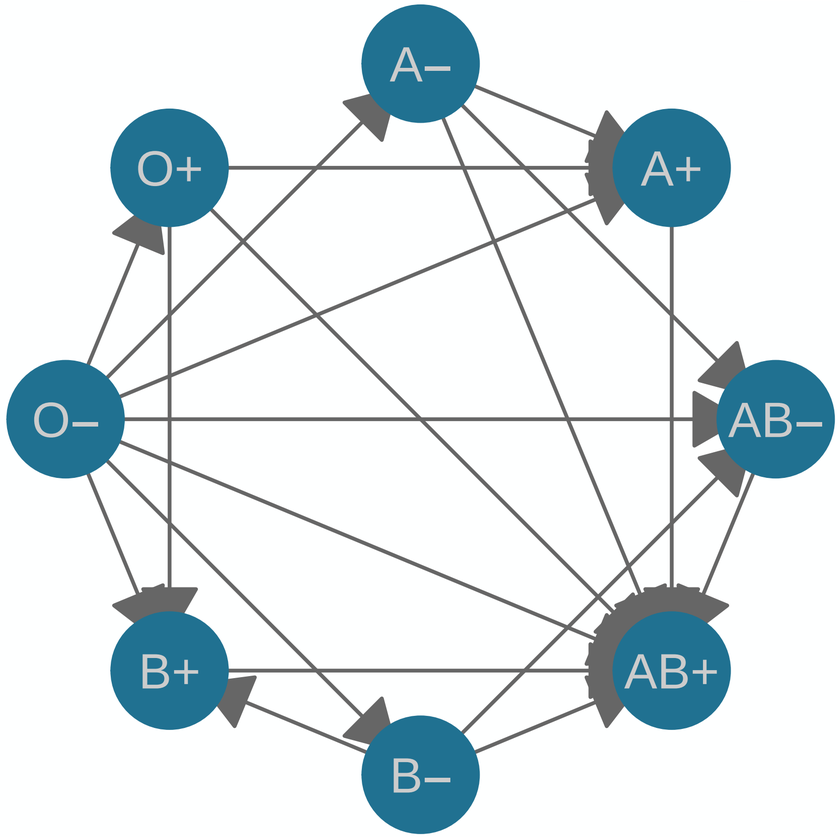


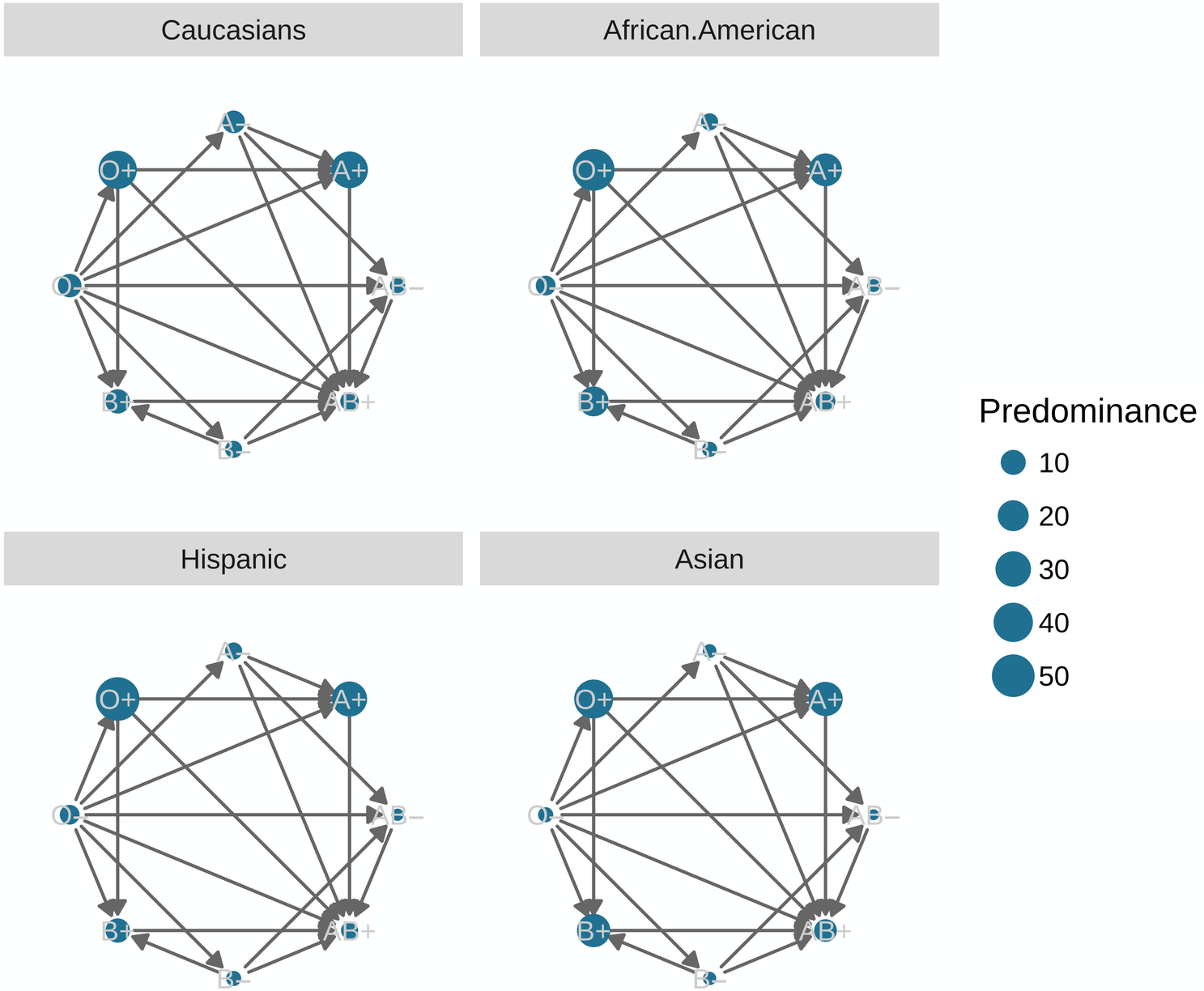
Network



Network

```
> library(geomnet)
> ggplot(blood_donors, aes(from_id = from, to_id = to)) +
    geom_net(layout = "circle",
             label = TRUE, directed = TRUE)
```




Caucasians
African.American
Hispanic
Asian
O+
A+
B+
AB+
Predominance
10
20
30
40
50

DATA VISUALIZATION WITH GGPLOT2

Let’s practice!

DATA VISUALIZATION WITH GGPLOT2

Diagnostic Plots

Diagnostic Plots

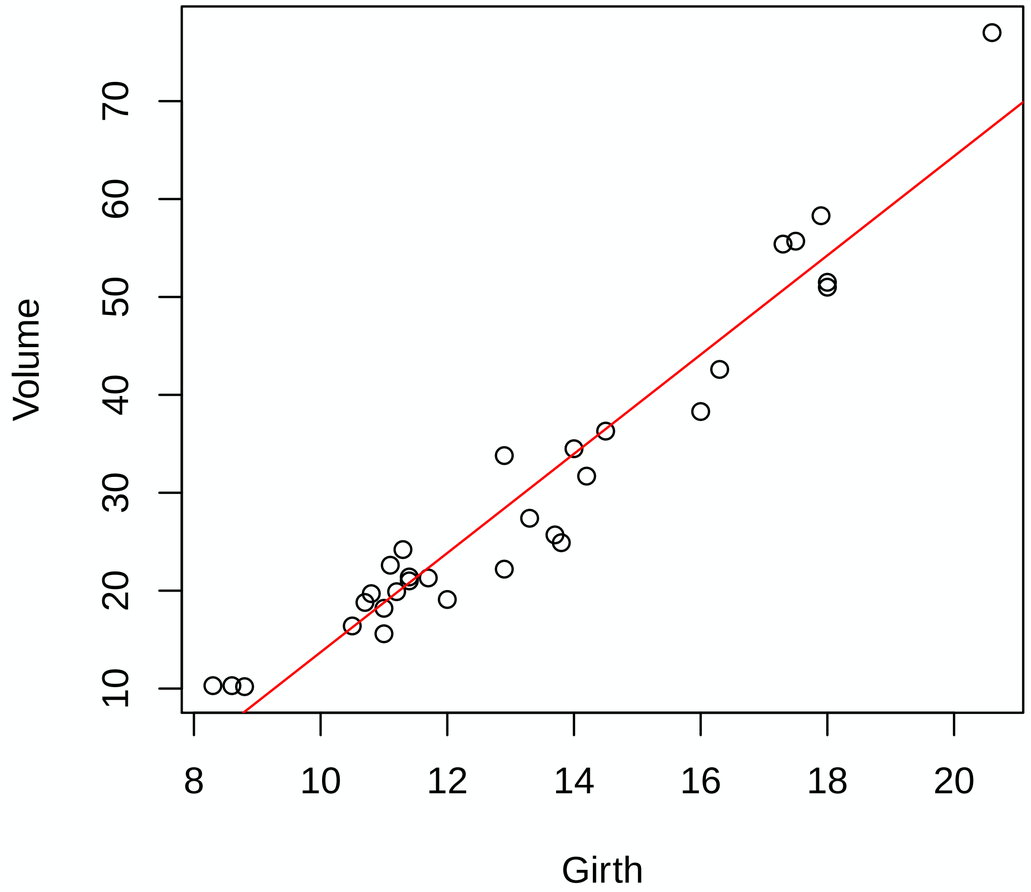
- Quantitative data
- Ordinary least squares model
- Assess how the model fits the data

trees

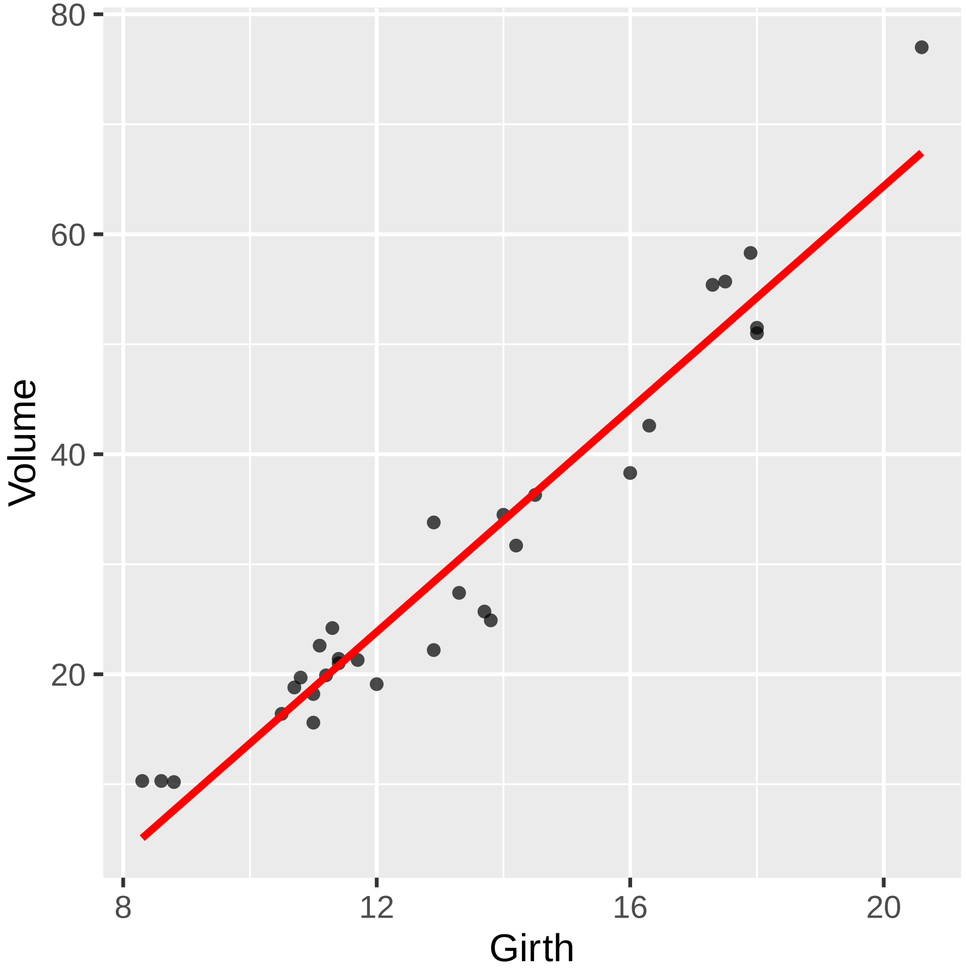
```
> dim(trees)
[1] 31  3

> head(trees)
  Girth Height Volume
1   8.3     70   10.3
2   8.6     65   10.3
3   8.8     63   10.2
4  10.5     72   16.4
5  10.7     81   18.8
6  10.8     83   19.7
```

Linear model

```
> res <- lm(Volume ~ Girth, data = trees)
> plot(Volume ~ Girth, data = trees)
> abline(res, col = "red")
```



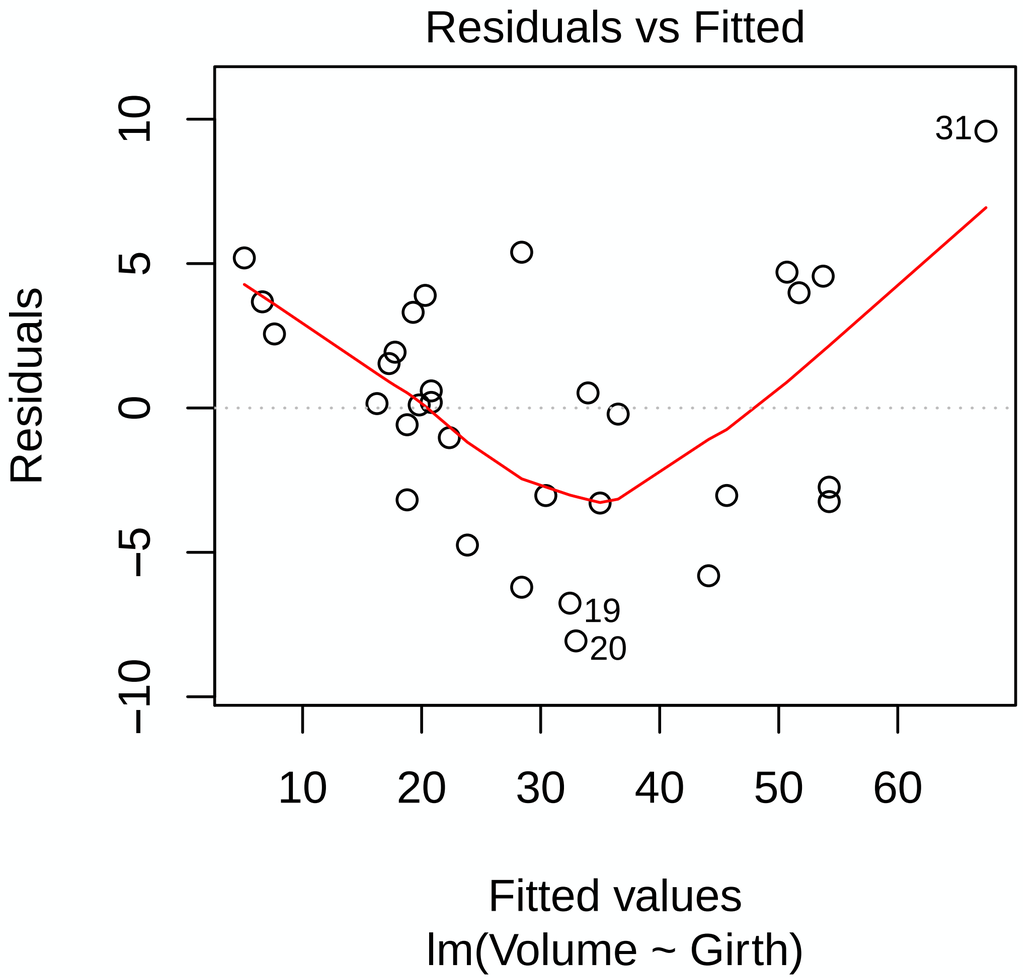
Linear model




```
> ggplot(trees, aes(Girth, Volume)) +
    geom_point(shape = 16, alpha = 0.7) +
    stat_smooth(method = "lm", se = F, col = "red")
```

plot model (1)

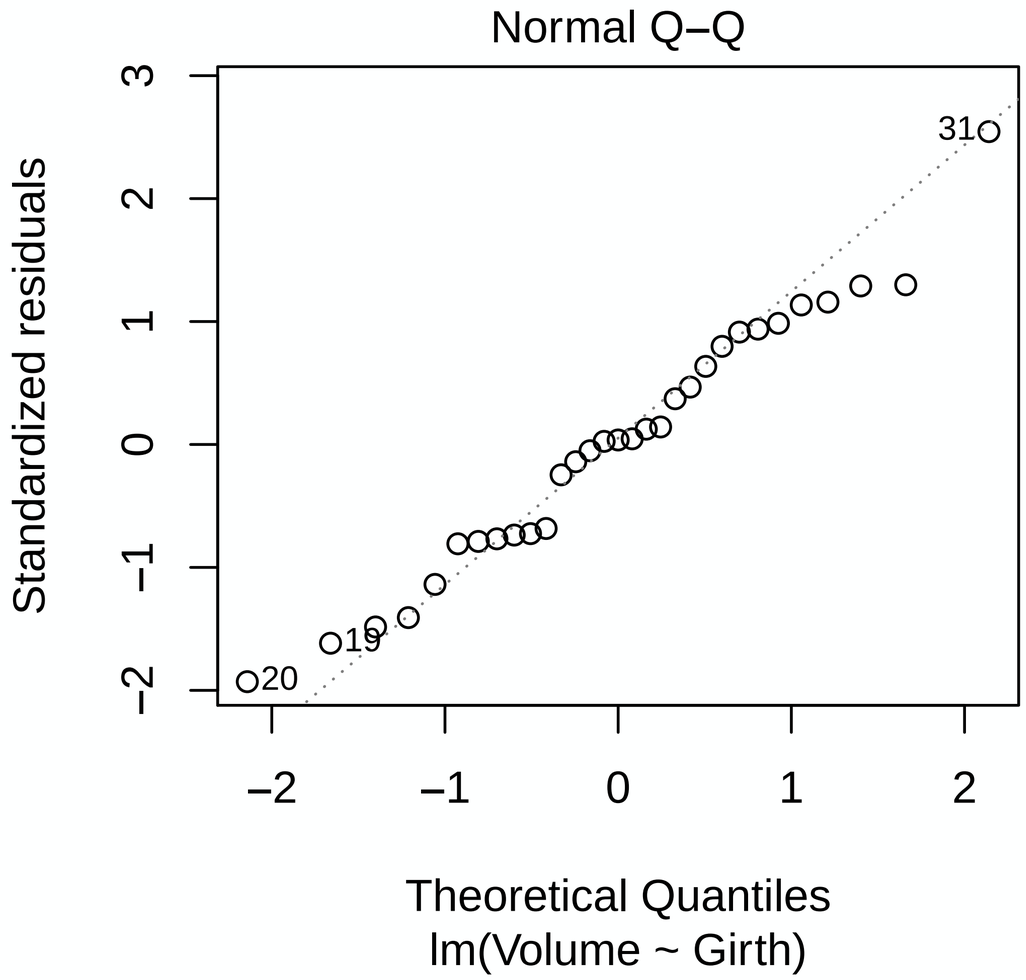
```
> res <- lm(Volume ~ Girth, data = trees)
> plot(res)
```



Ideally no clear trend

plot model (2)

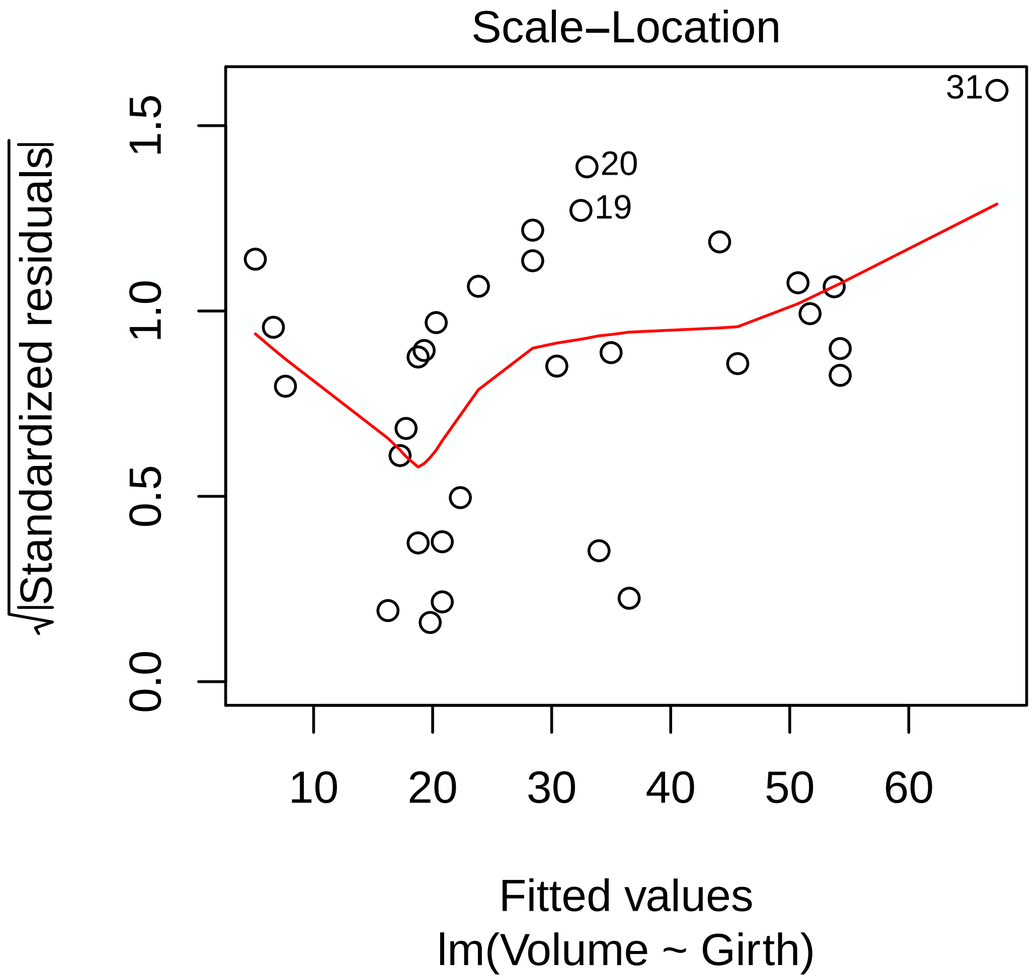
```
> res <- lm(Volume ~ Girth, data = trees)
> plot(res)
```



Are residuals normally distributed?

plot model (3)

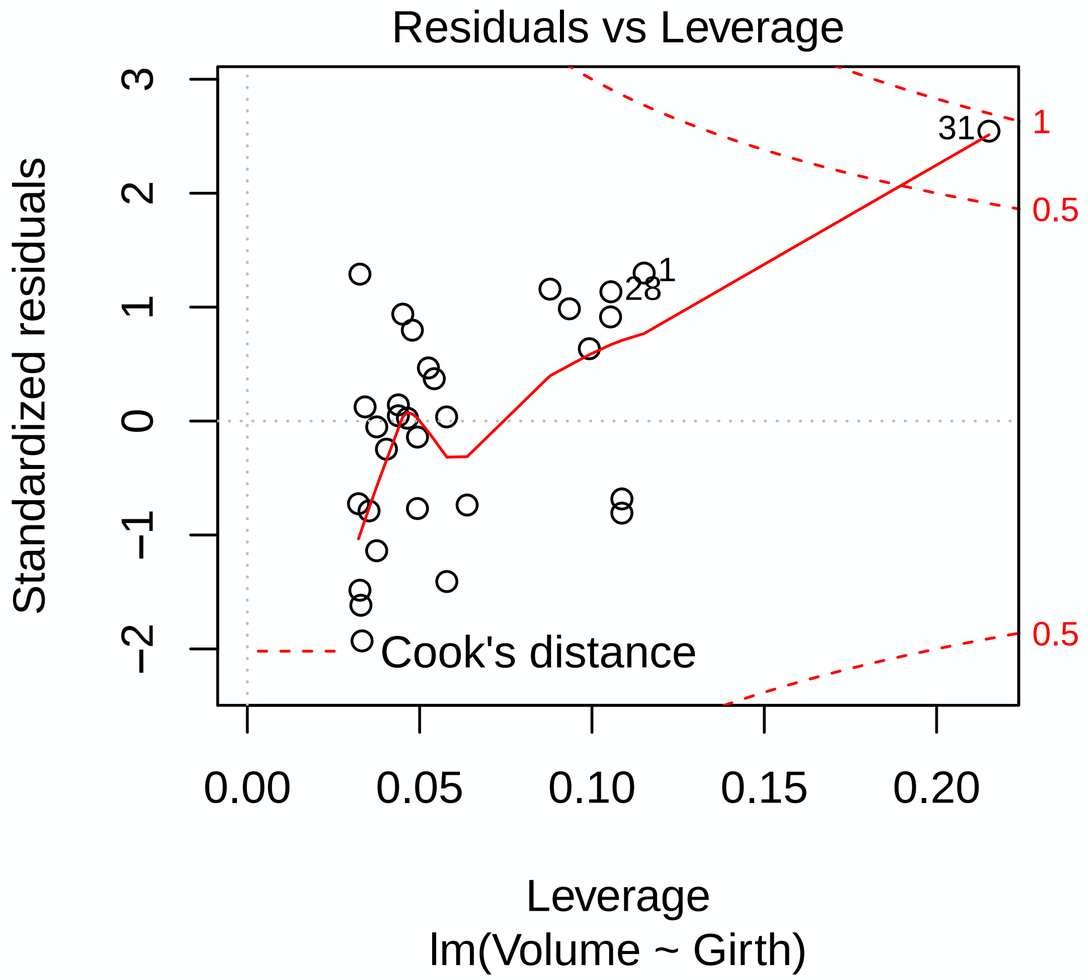
```
> res <- lm(Volume ~ Girth, data = trees)
> plot(res)
```

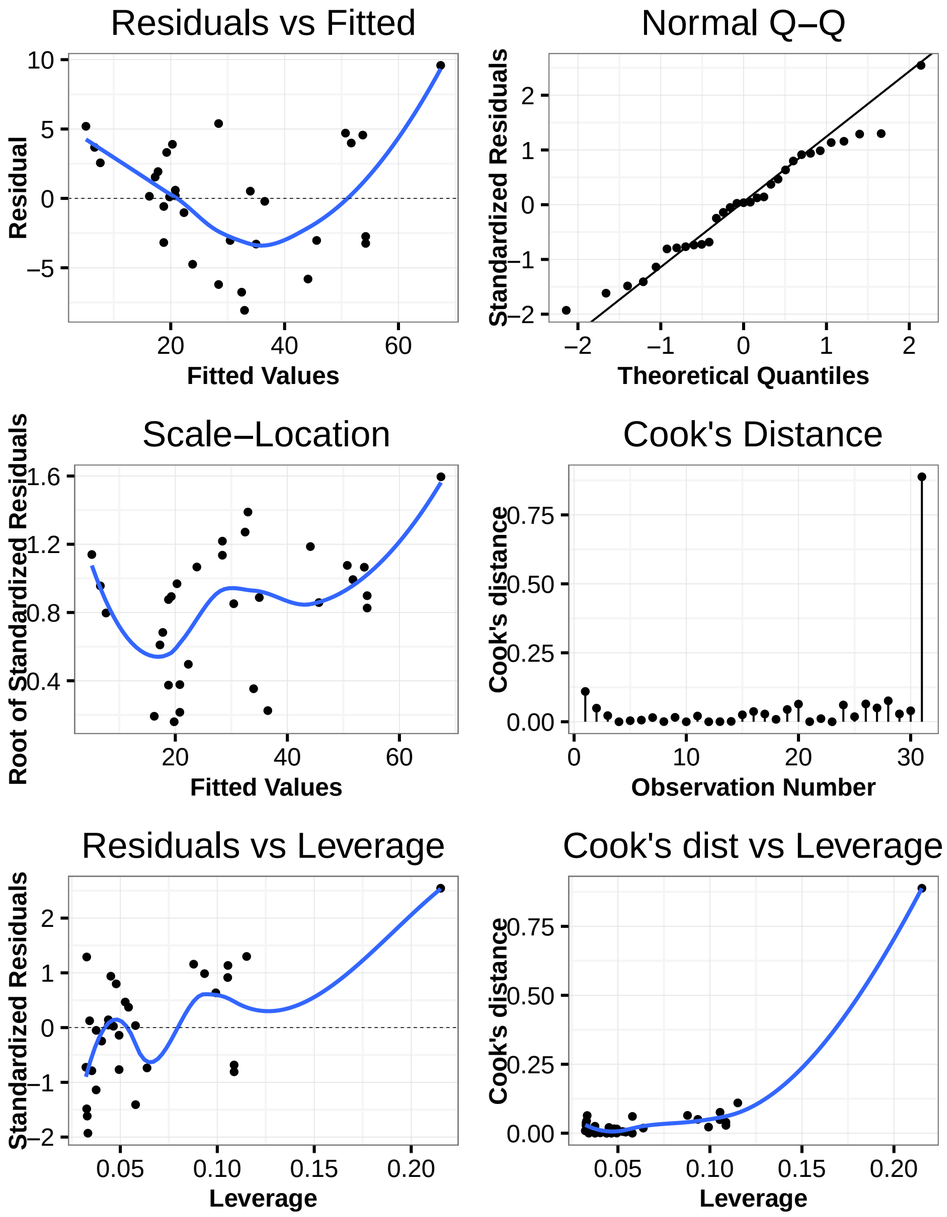


Assess homoscedasticity

plot model (4)

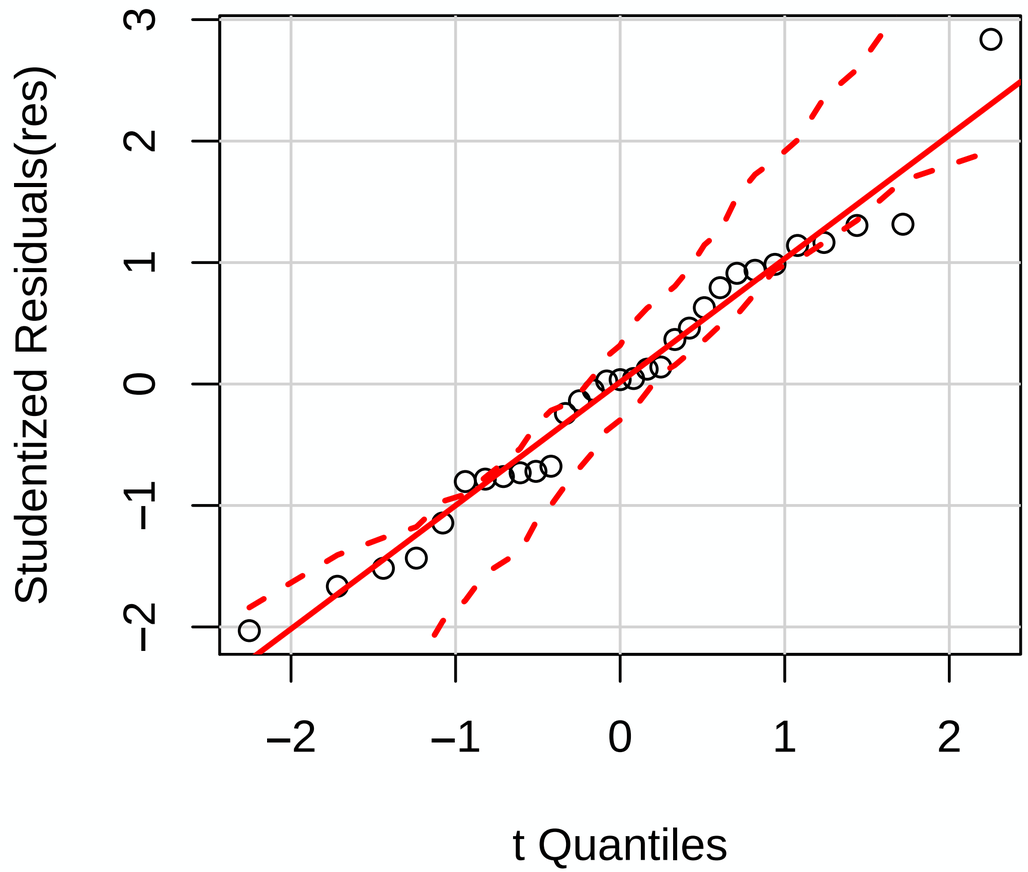
```
> res <- lm(Volume ~ Girth, data = trees)
> plot(res)
```




Residuals vs Fitted
Residual
Fitted Values
Normal Q–Q
Standardized Residuals
Theoretical Quantiles
Scale–Location
Root of Standardized Residuals
Fitted Values
Cook's Distance
Cook's distance
Observation Number
Residuals vs Leverage
Standardized Residuals
Leverage
Cook's dist vs Leverage
Cook's distance
Leverage

car package

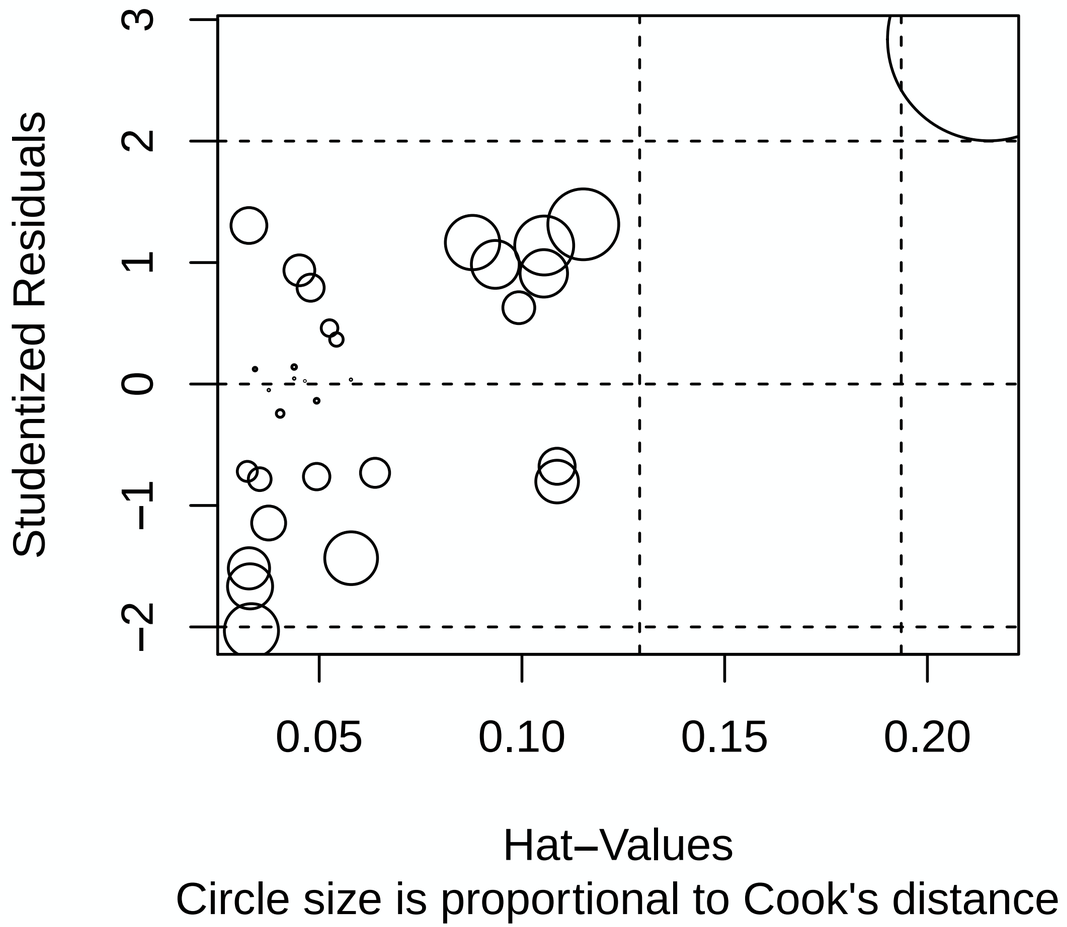
```
> library(car)
> qqPlot(res, id.method = "identify",
         simulate = TRUE, main = "Q-Q Plot")
```



John Fox & Sanford Weisberg, An R Companion to Applied Regression

car package

```
> library(car)
> influencePlot(res, id.method = "identify",
                    main = "Influence Plot")
```



Recent development

- John Fox
- Michael Friendly
- `matlib`

DATA VISUALIZATION WITH GGPLOT2

Let’s practice!