R visualization

Ho Thi Kim Cuong

Content

- Import data
- Reshape2 packages
- do.call () and reduce
- Basic plot and ggplot2
- Compare: histogram - boxplot / barplot - histogram

import data

read.csv(), read.table(), read.excel(), fread() *(package data.table),* load()

or choose.files(); load(files=)

```
data <- read.csv(
"/home/acer/Documents/DATA/DATA_VISUALIZATION/data/clinical.csv",
header=TRUE)
```

```
> data
   CaseNo Gender Age Survival.Months.
1       1      F  71                 1
2       2      M  70                13
3       3      F  73                 8
4       4      M  64                14
5       7      F  74                 4
6       8      F  84                 0
7       9      F  86                 1
8      12      M  49                15
9      13      M  76                 1
10     14      F  63                 7
11     N1      F  62               N/A
12     N2      M  64               N/A
13     N3      F  40               N/A
14     N4      F  56               N/A
```

install.packages(“data.table”)
library(data.table)

fread() function

It's simply works
Extremely fast
Improved read.table()

Example

```
data <- fread('/home/acer/Downloads/dowload/data.tsv', header=T)
```

https://www.youtube.com/watch?v=opai4Wmp6Zo

Reshape2

```
install.packages("reshape2")
library(reshape2)
```

```
FirstName <- c("Mary", "Mike", "Greg")
age <- c(44, 52, 46)
IQ <- c(160, 95, 110)
people <- data.frame(FirstName, age, IQ)
people
```

```
  FirstName age  IQ
1      Mary  44 160
2      Mike  52  95
3      Greg  46 110
```

```
> library(reshape2)
> melted_people <- melt(people, id = "FirstName")
> melted_people
```

```
  FirstName variable value
1      Mary      age    44
2      Mike      age    52
3      Greg      age    46
4      Mary       IQ   160
5      Mike       IQ    95
6      Greg       IQ   110
```

```
> dim(people)
[1] 3 3
```

```
> dim(melted_people)
[1] 6 3
```

Syntax:
melt(data, na.rm = FALSE, value.name = “value”)

Parameters:
data: represents dataset that has to be reshaped
na.rm: if TRUE, removes NA values from dataset
value.name: represents name of variable used to store values

A. **melt.data.frame** for data.frames

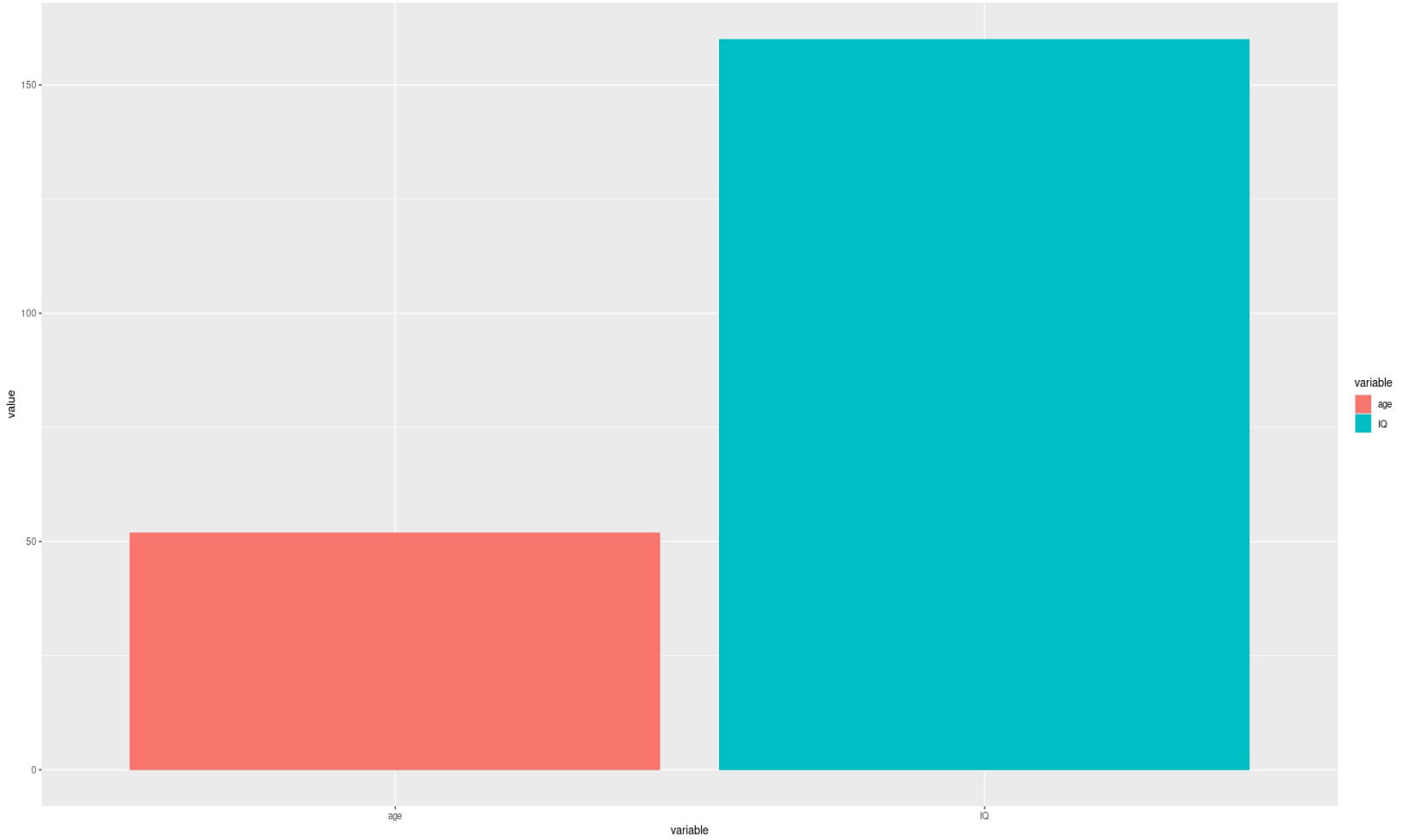
B. **melt.array** for arrays, matrices and tables

C. **melt.list** for lists

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> head(melt(iris))
Using Species as id variables
  Species     variable value
1  setosa Sepal.Length   5.1
2  setosa Sepal.Length   4.9
3  setosa Sepal.Length   4.7
4  setosa Sepal.Length   4.6
5  setosa Sepal.Length   5.0
6  setosa Sepal.Length   5.4
>
```

```
ggplot(data = melted_people , aes(x = variable, y = value, fill = variable)) +   geom_col(position = 'dodge')
```

	FirstName	variable	value
1	Mary	age	44
2	Mike	age	52
3	Greg	age	46
4	Mary	IQ	160
5	Mike	IQ	95
6	Greg	IQ	110



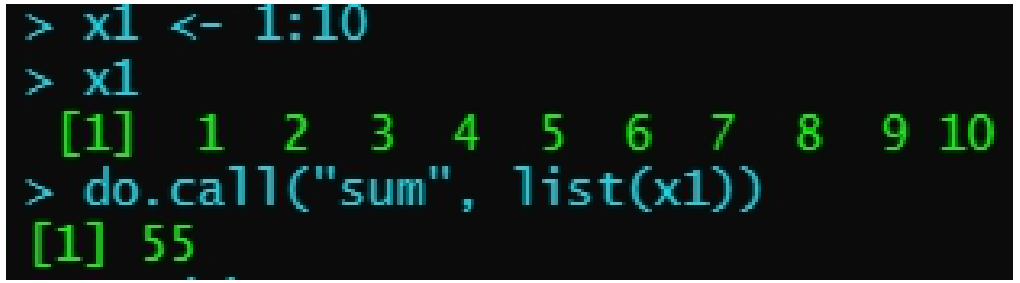
do.call ()

```
do.call ("any_function", arguments_list)
```

The do.call R function executes a function by its name and a list of corresponding arguments

This function allows you to call any R function, but instead of writing out the arguments one by one, you can use a list to hold the arguments of the function.

```
x1 <- 1:10
do.call(“sum”, list(x1))
```



Reduce ()

path="/home/acer/Documents/DATA/project_2/example"

setwd(path)

 # 5 files in list

list <- list.files(path, pattern = ".csv", recursive = T)

listdata <- lapply(list,function(x) read.csv(x))

#Reduce and apply function merge for 2 files for each

df <- Reduce (function(x,y) merge(x=x,y=y, by= c("MSHS","HoTên"),all=T),listdata)

df

```
path="D:/project_2/example"
setwd(path)
list <- list.files(path, pattern = ".csv", recursive = T)
listdata <- lapply(list,function(x) read.csv(x))

df <- Reduce (function(x,y) merge(x=x,y=y, by= c("MSHS","HoTên"),all=T),listdata)
df
```

Reduce vs do.call

```
x <- 1:10
Reduce(sum,x)
do.call(sum,list(x))
Reduce(function(A,B) sum(c(A,B)), x)
do.call(function(A,B) sum(c(A,B)), list(x))
Reduce(function(A,B,C) sum(c(A,B,C)), x)
```

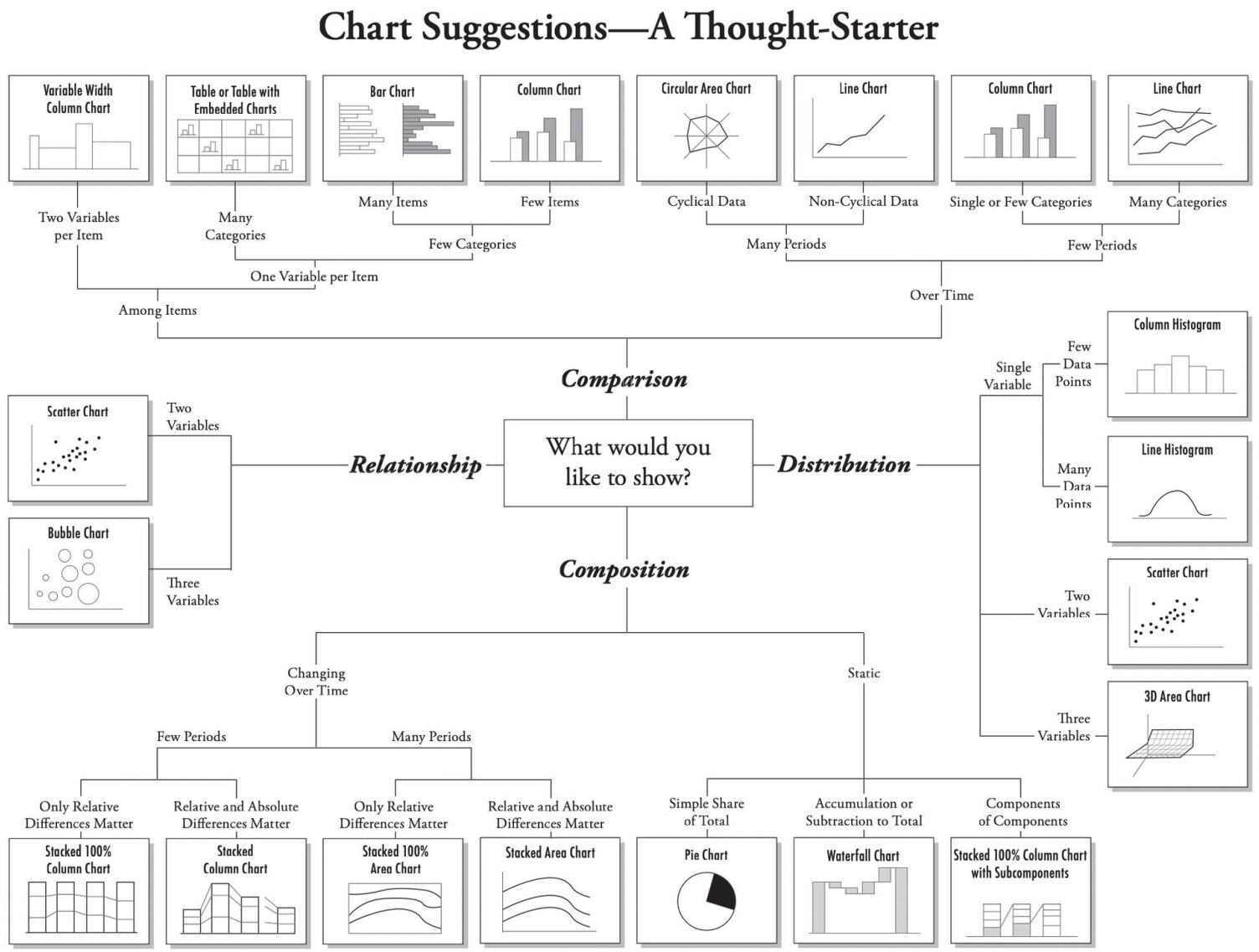
```
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> Reduce(sum,x)
[1] 55
> do.call(sum,list(x))
[1] 55
> Reduce(function(A,B) sum(c(A,B)), x)
[1] 55
> do.call(function(A,B) sum(c(A,B)), list(x))
Error in (function (A, B)  : argument "B" is missing, with no default
Called from: (function(A,B) sum(c(A,B)))(1:10)
Browse[1]> Q
> Reduce(function(A,B,C) sum(c(A,B,C)), x)
Error in f(init, x[[i]]) : argument "C" is missing, with no default
Called from: f(init, x[[i]])
Browse[1]> Q
>
```

Do.call	Reduce
apply function for **all elements** in list	Take **2 elements** to apply function for each (only 2 element)

Data Visualization Best Practices

There are four basic presentation types that you can use to present your data:

- **Comparison**
- **Composition**
- **Distribution**
- **Relationship**



Basic plot and ggplot2

- Univariate Graphs
- Bivariate Graphs

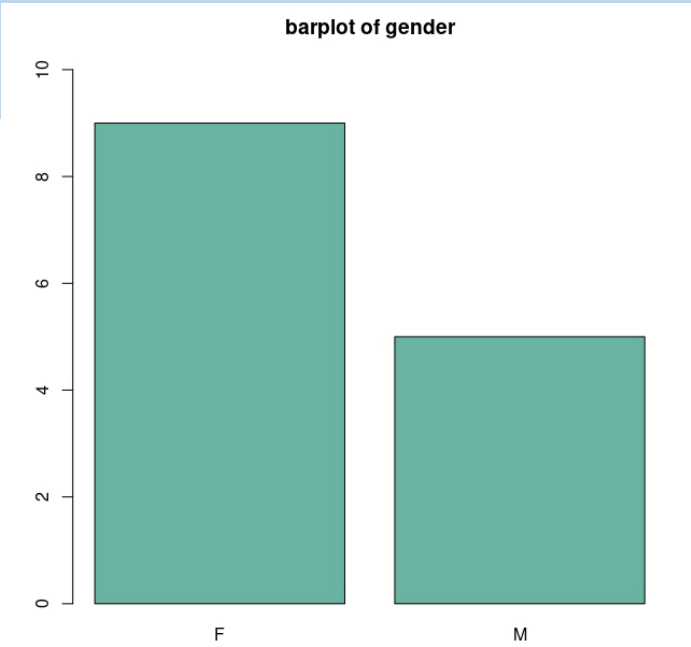
Univariate Graphs

- Univariate graphs plot the distribution of data from a single variable.
- The variable can be:
  - Categorical (e.g., race, sex): bar chart, pie chart
  - Quantitative (e.g., age, weight): histogram, density, dot chart

Univariate Graphs - Categorical

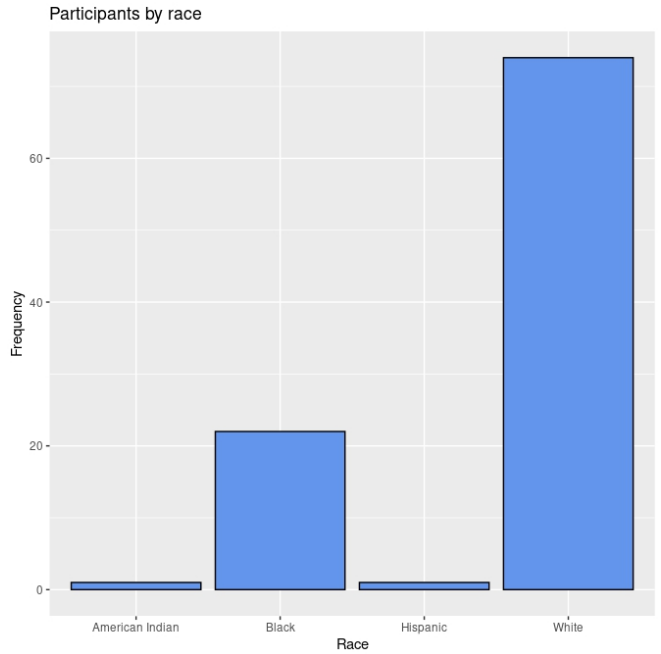
```
data <-
read.csv("/home/acer/Documents/DATA/DATA_VISUALIZATION/data/clinical.csv",header=TRUE)

b <- table(data$Gender)

barplot(b,ylim=c(0,10),col="#69b3a2",
main="barplot of gender")
```

```
> data
   CaseNo Gender Age Survival.Months.
1       1      F  71                1
2       2      M  70               13
3       3      F  73                8
4       4      M  64               14
5       7      F  74                4
6       8      F  84                0
7       9      F  86                1
8      12      M  49               15
9      13      M  76                1
10     14      F  63                7
11     N1      F  62              N/A
12     N2      M  64              N/A
13     N3      F  40              N/A
14     N4      F  56              N/A
> b <-table(data$Gender)
> b

F M
9 5
```




```
library(mosaicData)
library(ggplot2)
head(Marriage)

ggplot(Marriage, aes(x = race))
+
  geom_bar(fill = "cornflowerblue",
           color="black") +
  labs(x = "Race",
       y = "Frequency",
       title = "Participants by race")
```

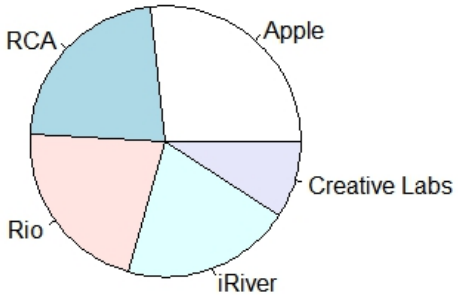
```
> head(Marriage)
  bookpageID    appdate ceremonydate delay     officialTitle person        dob
1   B230p539 1996-10-29   1996-11-09    11    CIRCUIT JUDGE   Groom 2064-04-11
2   B230p677 1996-11-12   1996-11-12     0 MARRIAGE OFFICIAL   Groom 2064-08-06
3   B230p766 1996-11-19   1996-11-27     8 MARRIAGE OFFICIAL   Groom 2062-02-20
4   B230p892 1996-12-02   1996-12-07     5          MINISTER   Groom 2056-05-20
5   B230p994 1996-12-09   1996-12-14     5          MINISTER   Groom 2066-12-14
6  B230p1209 1996-12-26   1996-12-26     0 MARRIAGE OFFICIAL   Groom 1970-02-21
       age     race prevcount prevconc hs college dayOfBirth        sign
1 32.60274    White         0     <NA> 12       7        102       Aries
2 32.29041    White         1  Divorce 12       0        219         Leo
3 34.79178 Hispanic         1  Divorce 12       3         51      Pisces
4 40.57808    Black         1  Divorce 12       4        141      Gemini
5 30.02192    White         0     <NA> 12       0        348 Saggitarius
6 26.86301    White         1     <NA> 12       0         52      Pisces
```

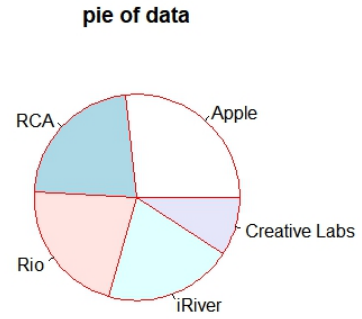


Univariate Graphs - Categorical

```
a <- c("Apple","RCA","Rio","iRiver","Creative Labs")
b <- c(0.18,0.15,0.144,0.135,0.062)
data <- data.frame(a,b)
data
pie(data$b, data$a)
pie(data$b, data$a, border ="red", main = "Pie of data")
```

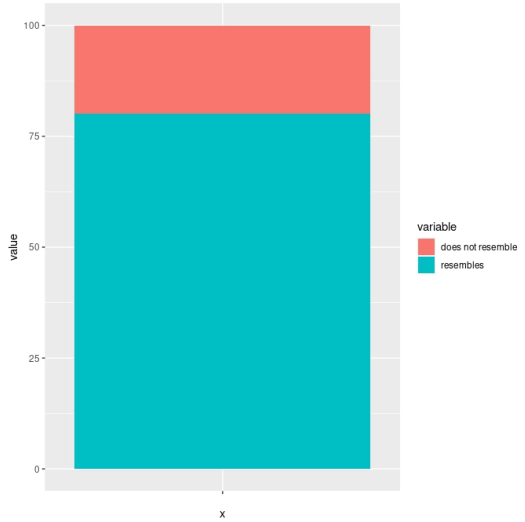
```
> data
             a     b
1        Apple 0.180
2          RCA 0.150
3          Rio 0.144
4       iRiver 0.135
5 Creative Labs 0.062
> pie(data$b,data$a)
```

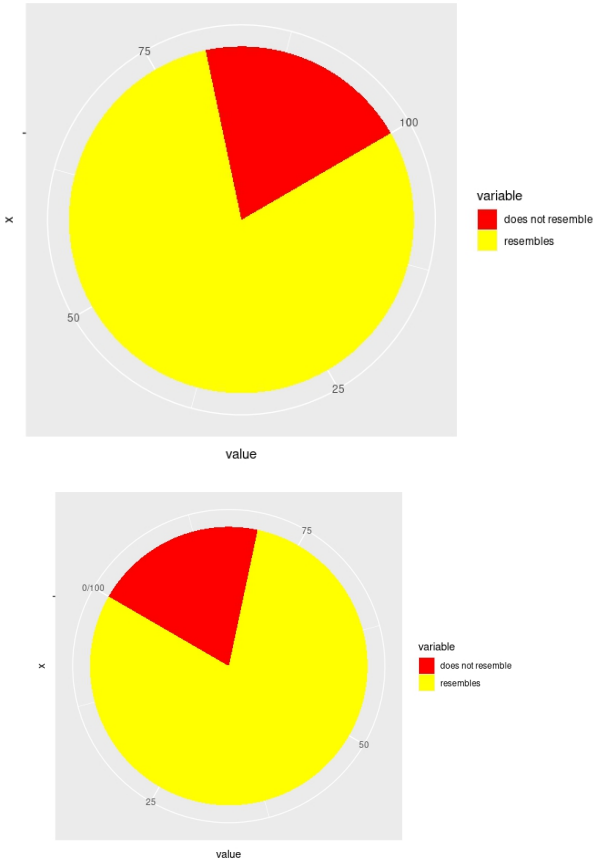




- A pie chart = stacked bar chart + polar coordinates

```
df <- data.frame(
  variable = c("does not resemble", "resembles"),
  value = c(20, 80))
ggplot(df, aes(x = "", y = value, fill = variable)) +
  geom_col(width = 1) +
  scale_fill_manual(values = c("red", "yellow")) +
  coord_polar("y", start = pi / 3, direction=-1)
```





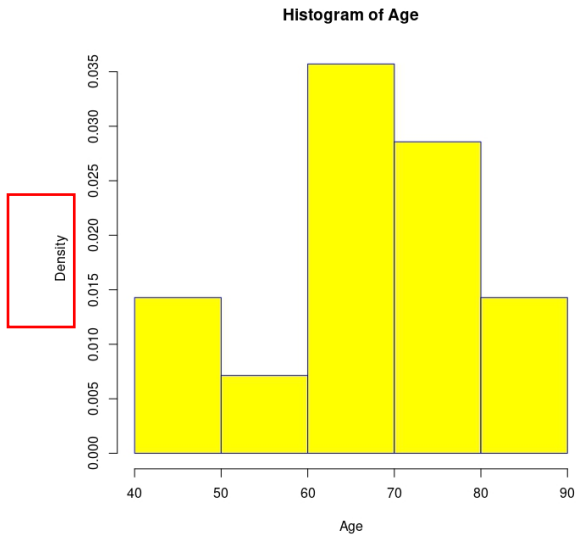
Univariate Graphs - Quantitative

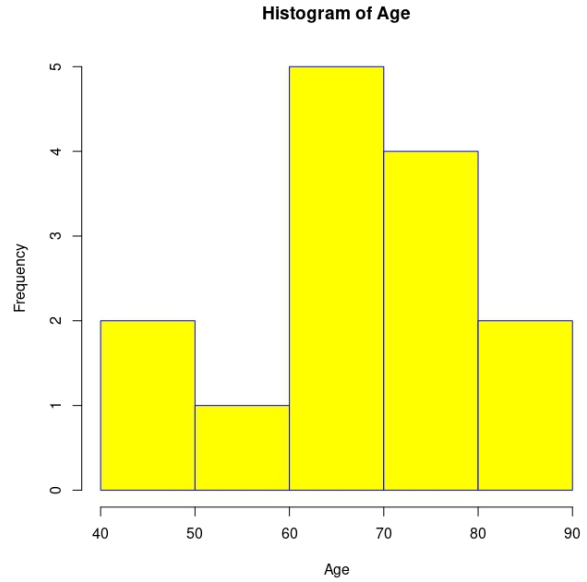
Histogram basic

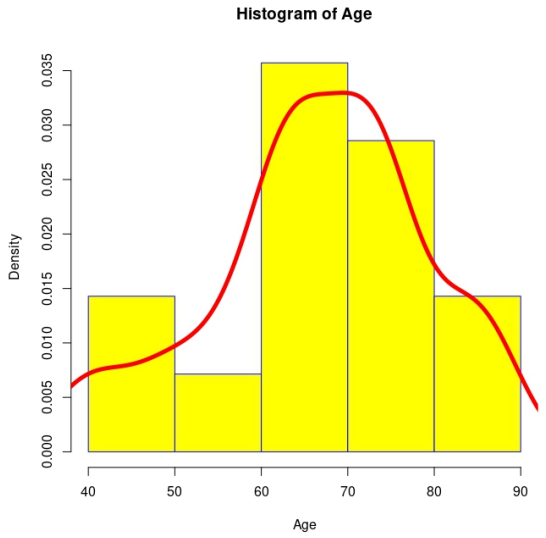
hist (**var**, xlab, ylab, main, xlim, ylim, col, border, prob)
lines (**density(var)**, col, lwd)

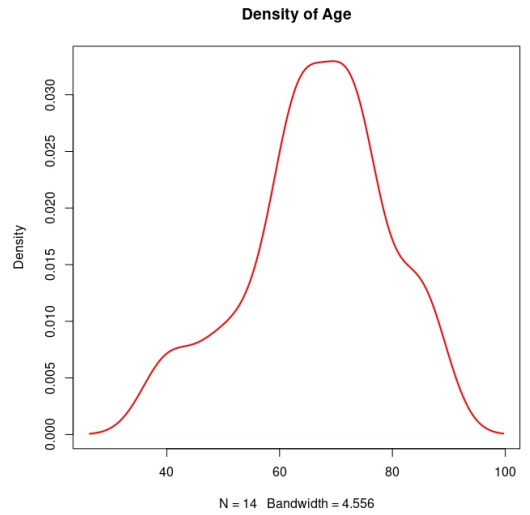
```
hist(data$Age, col = "yellow",  main= "Histogram of Age", xlab = "Age", border = "blue", probability = T)

 lines(density(data$Age),col="red",lwd=5)

dx <- density(data$Age)
plot(dx, lwd=2, col = "red", main = "Density of Age")
```

```
> density(data$Age)

Call:
        density.default(x = data$Age)

Data: data$Age (14 obs.);       Bandwidth 'bw' = 4.556

       x                y
 Min.   :26.33   Min.   :7.036e-05
 1st Qu.:44.67   1st Qu.:4.694e-03
 Median :63.00   Median :1.033e-02
 Mean   :63.00   Mean   :1.362e-02
 3rd Qu.:81.33   3rd Qu.:2.222e-02
 Max.   :99.67   Max.   :3.296e-02
```







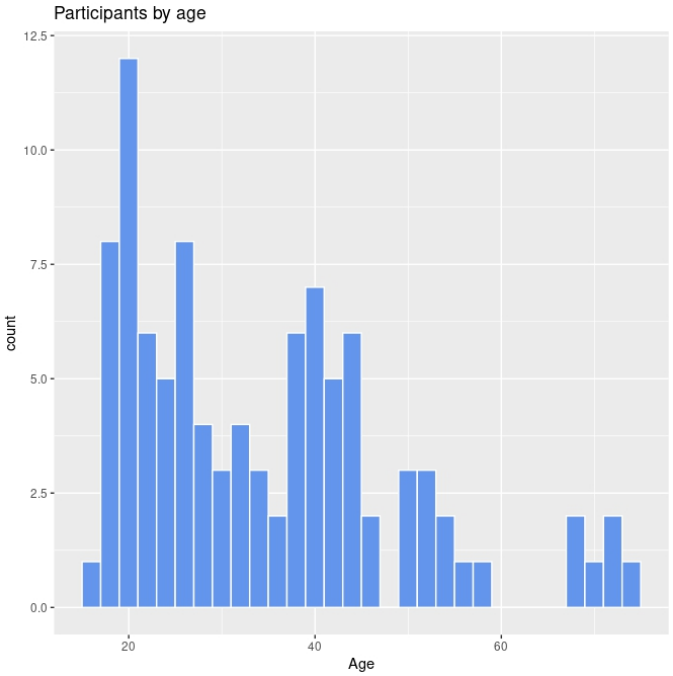


Univariate Graphs - Quantitative

Histogram with specified fill and border colors

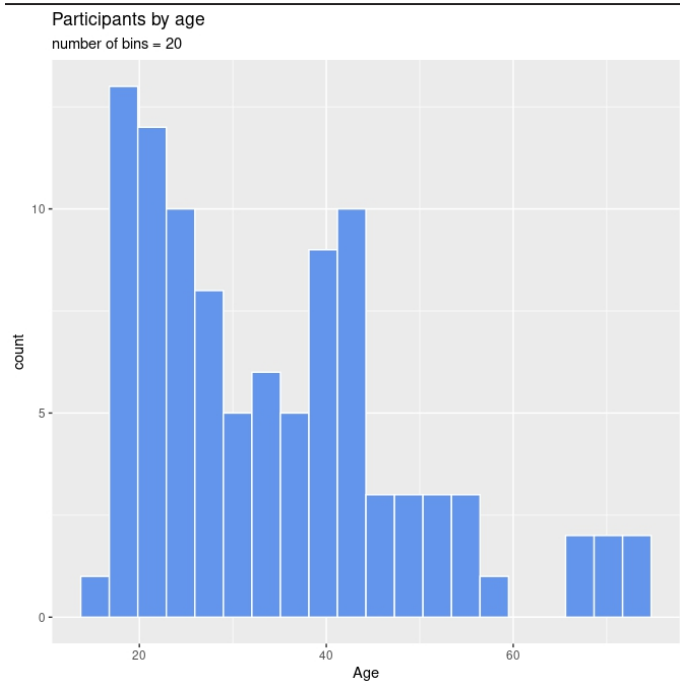
```
ggplot(Marriage, aes(x = age)) +
  geom_histogram(fill = "cornflowerblue",
                 color = "white") +
  labs(title="Participants by age",
       x = "Age")
```

```
> head(Marriage)
  bookpageID    appdate ceremonydate delay     officialTitle person        dob
1   B230p539 1996-10-29   1996-11-09    11    CIRCUIT JUDGE   Groom 2064-04-11
2   B230p677 1996-11-12   1996-11-12     0 MARRIAGE OFFICIAL   Groom 2064-08-06
3   B230p766 1996-11-19   1996-11-27     8 MARRIAGE OFFICIAL   Groom 2062-02-20
4   B230p892 1996-12-02   1996-12-07     5          MINISTER   Groom 2056-05-20
5   B230p994 1996-12-09   1996-12-14     5          MINISTER   Groom 2066-12-14
6  B230p1209 1996-12-26   1996-12-26     0 MARRIAGE OFFICIAL   Groom 1970-02-21
       age     race prevcount prevconc hs college dayOfBirth        sign
1 32.60274    White         0     <NA> 12       7        102       Aries
2 32.29041    White         1  Divorce 12       0        219         Leo
3 34.79178 Hispanic         1  Divorce 12       3         51      Pisces
4 40.57808    Black         1  Divorce 12       4        141      Gemini
5 30.02192    White         0     <NA> 12       0        348 Saggitarius
6 26.86301    White         1     <NA> 12       0         52      Pisces
```

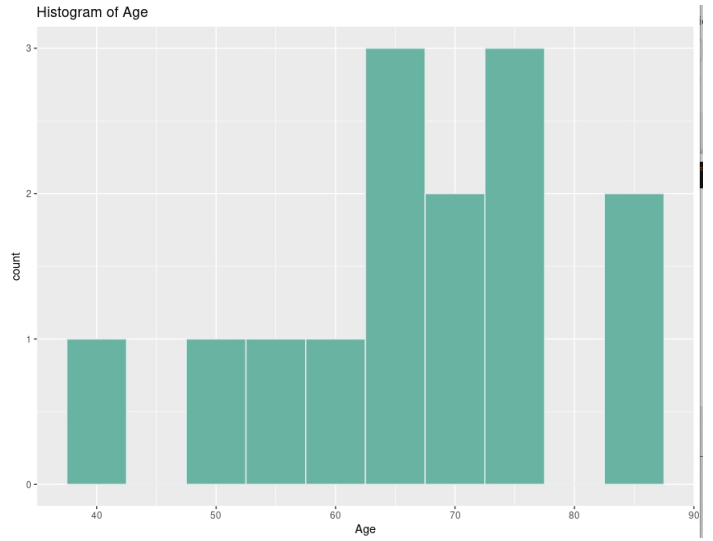


Histogram with a specified number of bins

```
ggplot(Marriage, aes(x = age)) +
  geom_histogram(fill = "cornflowerblue",
                 color = "white",
                 bins = 20) +
  labs(title="Participants by age",
       subtitle = "number of bins = 20",
       x = "Age")
```



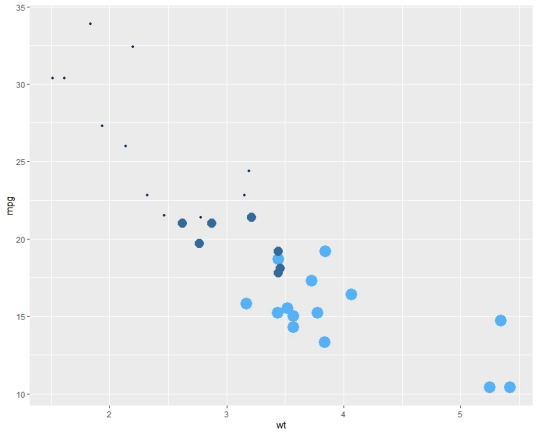
Univariate Graphs - Quantitative

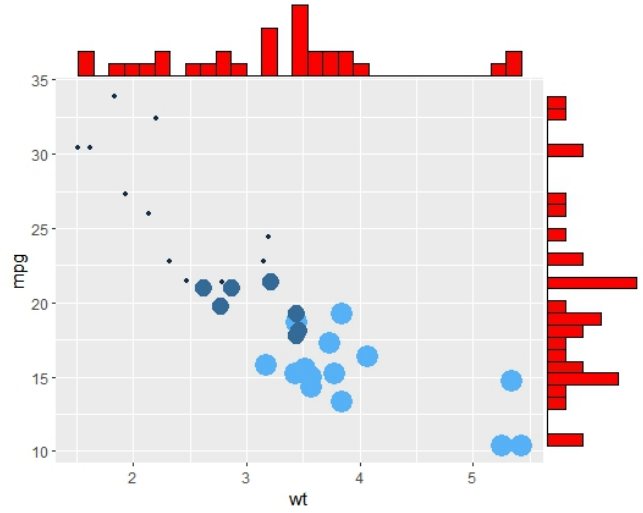


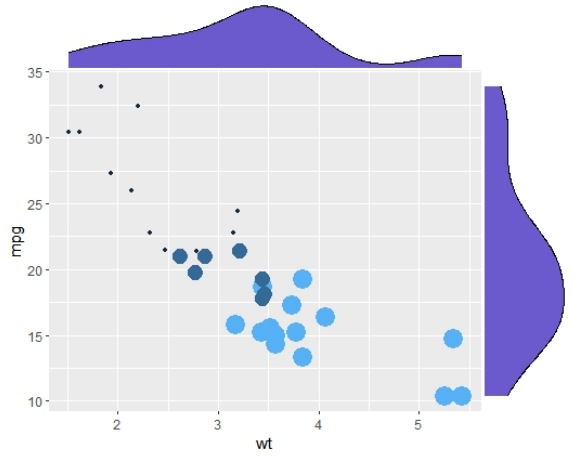
ggplot(data,aes(x=Age))+ geom_histogram(binwidth=5, fill="#69b3a2", color="#e9ecef") + ggtitle("Histogram of Age")

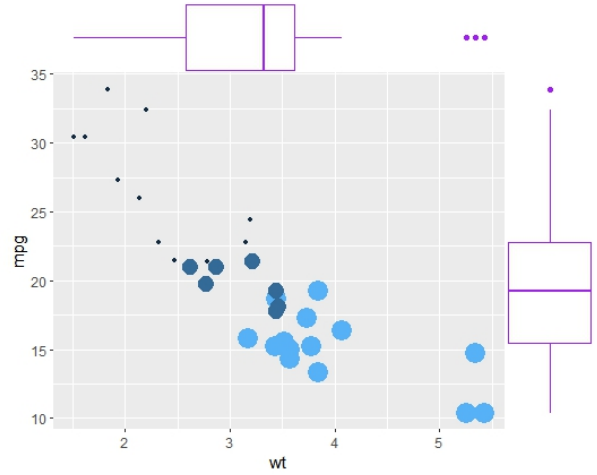
Histogram + scatterplot
library(ggExtra)
p <- ggplot(mtcars, aes(x=wt, y=mpg, color= cyl, size = cyl)) +
geom_point() + theme(legend.position = "none")
p1 <- ggMarginal(p, type="histogram", fill="red")
p2 <- ggMarginal(p, type="density", fill="slateblue")
p3 <- ggMarginal(p, type="boxplot", fill="purple")

```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```







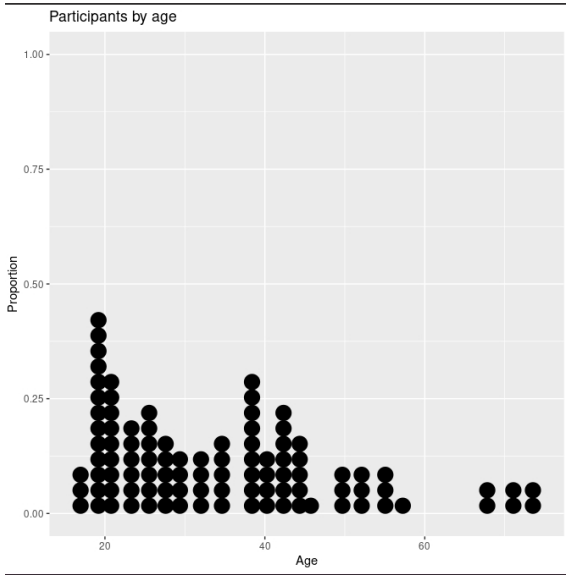


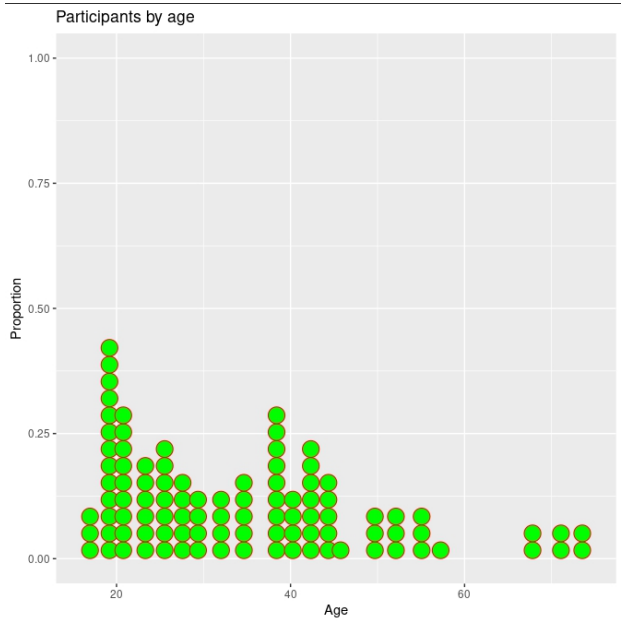
Univariate Graphs - Quantitative

Dotplot

- Each observation is represented by a dot
- By default, the width of a dot corresponds to the bin width, and dots are stacked, with each dot representing one observation.
- Best when the number of observations is small

```
ggplot(Marriage, aes(x = age)) + geom_dotplot() + labs(title =
"Participants by age", y = "Proportion", x = "Age")
#Dotplot with a specified color scheme
ggplot(Marriage, aes(x = age)) +
geom_dotplot(fill="green",color="red") + labs(title =
"Participants by age", y = "Proportion", x = "Age")
```

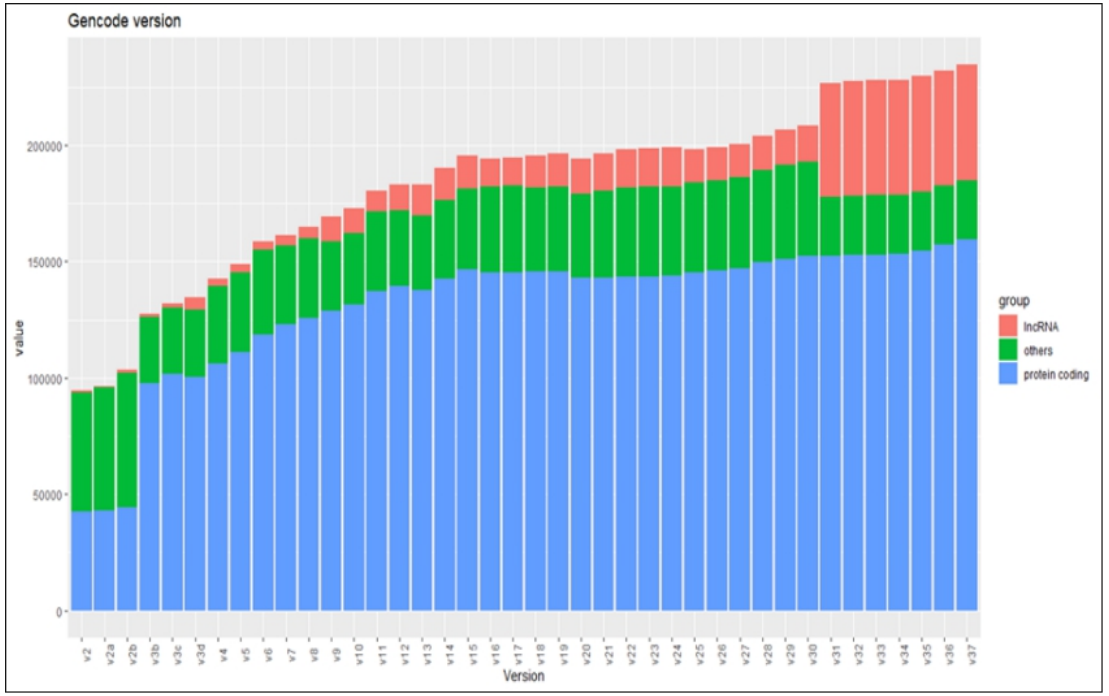




Bivariate Graphs - Categorical vs. Categorical

Stacked bar chart

```
#Download
for i in `seq 1 38`
do
wget http://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_${i}/gencode.v${i}.annotation.gtf.gz
echo ${i}
done
# edit
for i in `find *gtf.gz`
do
zcat $i | tr -d '";'| awk 'OFS="\t" {if ($3=="transcript") print $14}'|sort|uniq -c| awk 'OFS="\t" {print $1,$2}' > ${i}.txt
done
```




```
#R
path="D:/gencode2"
setwd(path)
list.files(path)
list=list.files(path,pattern=".gz.txt",recursive=T)
listdata=lapply(list,function(x) read.table(x,sep="\t",header=T,col.names=c(x,"count")))
head(listdata)
#merge
df <- Reduce(function(x,y) merge(x=x, y=y, by="count",all=T), listdata)
#rename
names(df) <- gsub("gencode.", "" ,names(df))
names(df) <- gsub(".annotation.gtf.*", "" ,names(df))
library(ggplot2)
library(reshape2)
#
data<- melt(df)
library(tidyverse)
data= data %>% mutate(group=ifelse(count=="lncRNA"|count=="lincRNA" | count=="miRNA","lncRNA",ifelse(count=="protein_coding","protein coding","others")))
data$variable <- factor(data$variable, levels=c("v2","v2a","v2b","v3b","v3c","v3d","v4","v5","v6","v7","v8","v9","v10","v11","v12","v13","v14","v15","v16","v17","v18","v19","v20","v21","v22","v23","v24","v25","v26","v27","v28","v29","v30","v31","v32","v33","v34","v35","v36","v37","v38"))
ggplot(data,aes(x=variable,y=value,fill=group))+
geom_bar(position="stack",stat="identity") + xlab("Version")+ ylab("value")+
ggtitle("Gencode version") +theme(axis.text.x = element_text(angle = 90))
```

Bivariate Graphs

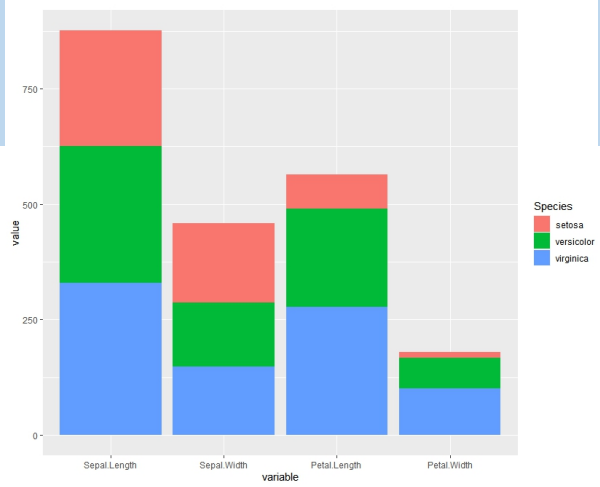
- Bivariate graphs display the relationship between two variables
  - Categorical vs. Categorical: Stacked bar chart, Grouped bar chart, Segmented bar chart
  - Quantitative vs. Quantitative: Scatterplot, Line plot
  - Categorical vs. Quantitative: barplot, boxplot, violin plot

Bivariate Graphs - Categorical vs. Categorical

Grouped bar chart

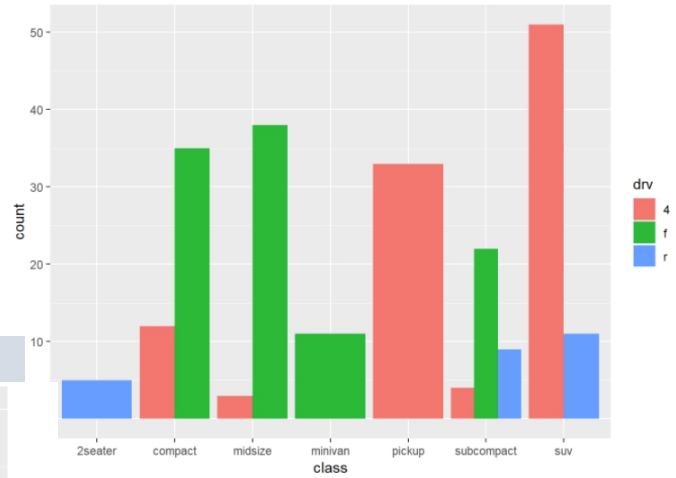
```
data <- melt(iris)
head(data)
ggplot(data,aes(x=variable,y=value,fill=Species))+
geom_bar(stat="identity")
```

```
> head(data)
  Species      variable value
1  setosa Sepal.Length   5.1
2  setosa Sepal.Length   4.9
3  setosa Sepal.Length   4.7
4  setosa Sepal.Length   4.6
5  setosa Sepal.Length   5.0
6  setosa Sepal.Length   5.4
```



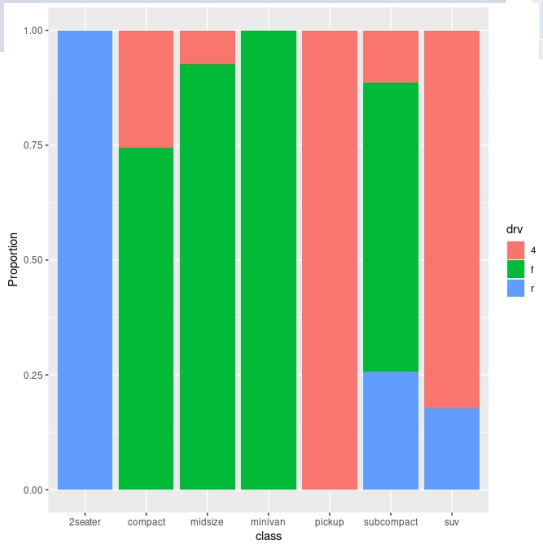

```
ggplot(mpg,
       aes(x = class,
           fill = drv)) +
  geom_bar(position = "dodge")
```

```
> mpg
# A tibble: 234 × 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
 2 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
 3 audi         a4           2    2008     4 manu… f        20    31 p     comp…
 4 audi         a4           2    2008     4 auto… f        21    30 p     comp…
 5 audi         a4           2.8  1999     6 auto… f        16    26 p     comp…
 6 audi         a4           2.8  1999     6 manu… f        18    26 p     comp…
 7 audi         a4           3.1  2008     6 auto… f        18    27 p     comp…
 8 audi         a4 quattro   1.8  1999     4 manu… 4        18    26 p     comp…
 9 audi         a4 quattro   1.8  1999     4 auto… 4        16    25 p     comp…
10 audi         a4 quattro   2    2008     4 manu… 4        20    28 p     comp…
# … with 224 more rows
> mpg$class
  [1] "compact"    "compact"    "compact"    "compact"    "compact"
  [6] "compact"    "compact"    "compact"    "compact"    "compact"
 [11] "compact"    "compact"    "compact"    "compact"    "compact"
 [16] "midsize"    "midsize"    "midsize"    "suv"        "suv"
 [21] "suv"        "suv"        "suv"        "2seater"    "2seater"
 [26] "2seater"    "2seater"    "2seater"    "suv"        "suv"
 [31] "suv"        "suv"        "midsize"    "midsize"    "midsize"
 [36] "midsize"    "minivan"    "minivan"    "minivan"    "minivan"
 [41] "minivan"    "minivan"    "minivan"    "minivan"    "minivan"
 [46] "minivan"    "minivan"    "minivan"    "pickup"     "pickup"
 [51] "pickup"     "pickup"     "pickup"     "pickup"     "pickup"
 [56] "pickup"     "pickup"     "suv"        "suv"        "suv"
 [61] "suv"        "suv"        "suv"        "suv"        "pickup"
 [66] "pickup"     "pickup"     "pickup"     "pickup"     "pickup"
 [71] "pickup"     "pickup"     "pickup"     "pickup"     "suv"
```



Segmented bar chart

```
ggplot(mpg, aes(x = class, fill = drv)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion")
```

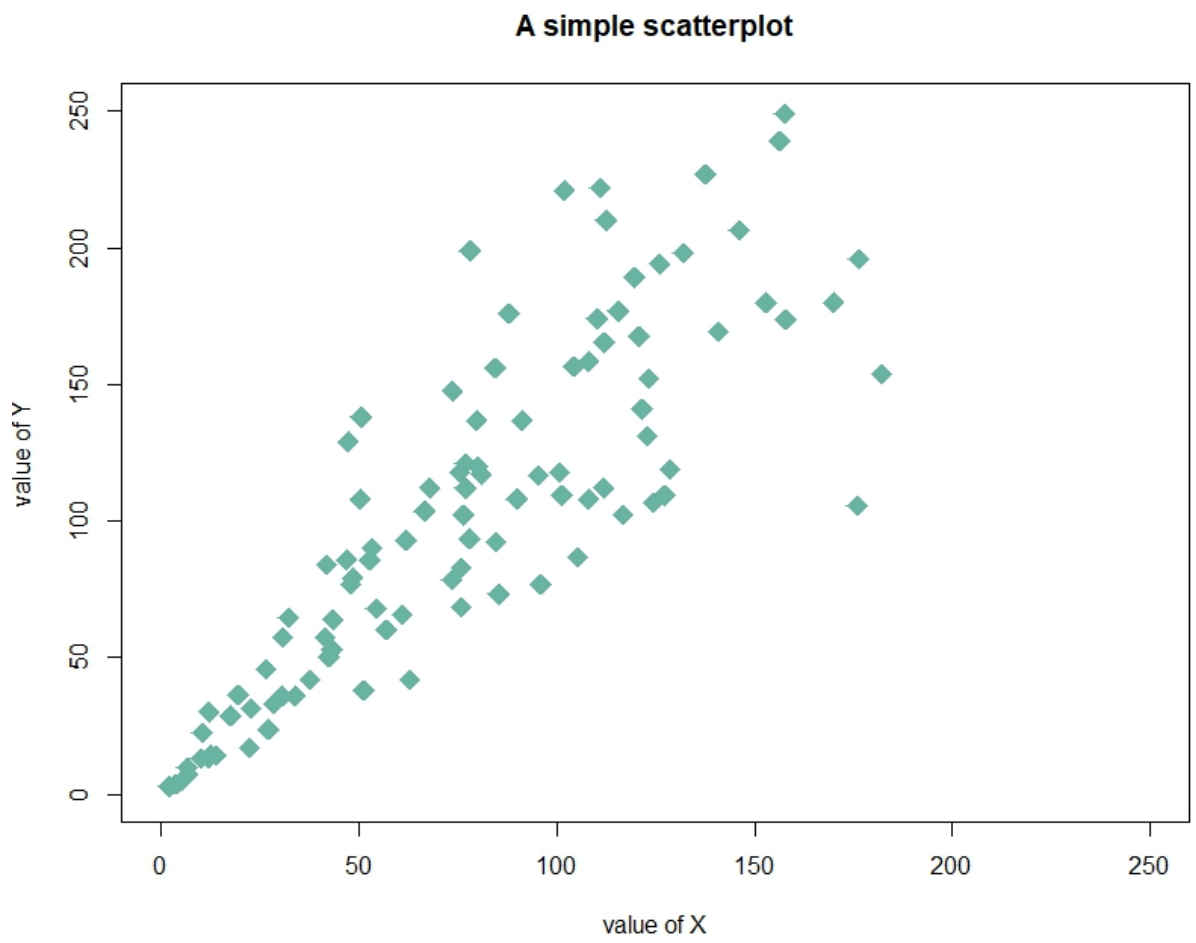


Bivariate Graphs - Quantitative vs. Quantitative

Scatter plot

plot (varX ~ varY, xlab, ylab,…)

```
data = data.frame(
  x=seq(1:100) + 0.1*seq(1:100)*sample(c(1:10) , 100 ,
replace=T),
  y=seq(1:100) + 0.2*seq(1:100)*sample(c(1:10) , 100 ,
replace=T)
)

# Basic scatterplot
plot(data$x, data$y,
     xlim=c(0,250) , ylim=c(0,250),
     pch=18,
     cex=2,
     col="#69b3a2",
     xlab="value of X", ylab="value of Y",
     main="A simple scatterplot"
     )
```



Bivariate Graphs - Quantitative vs. Quantitative

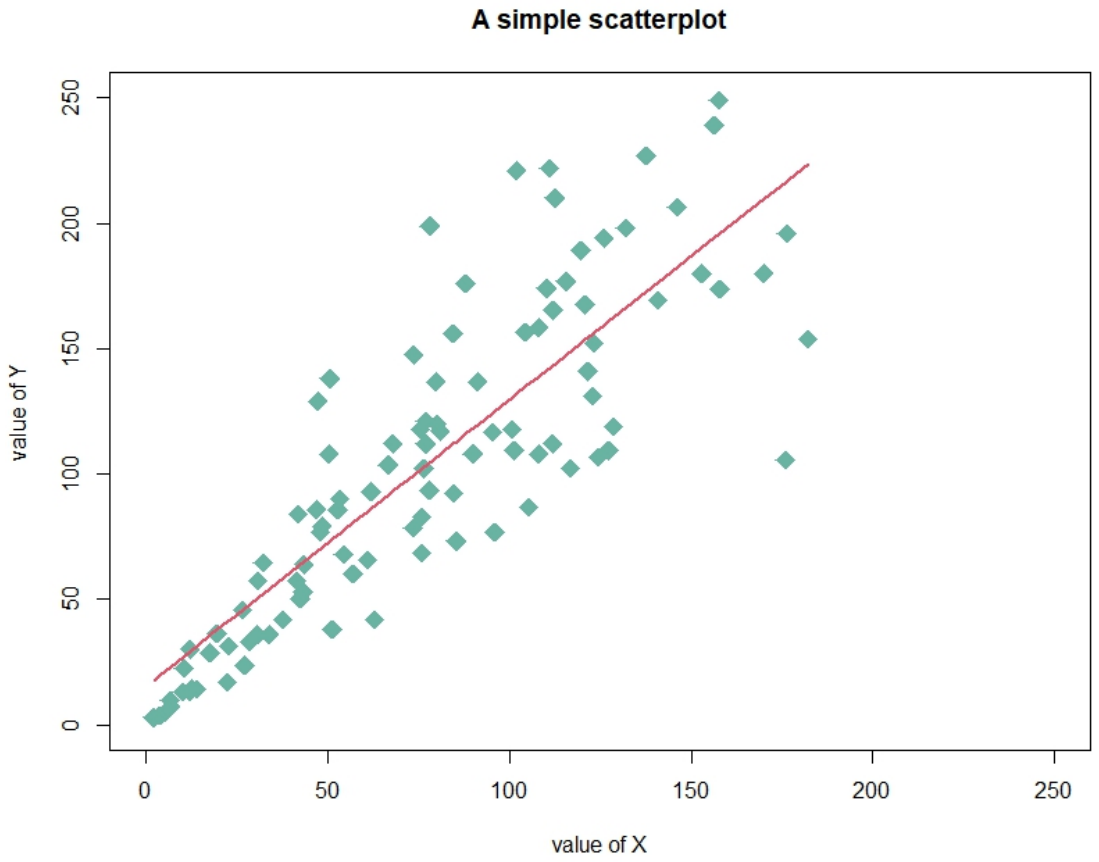
Scatter plot with trend line

plot (varX ~ varY, xlab, ylab,…)

model <- lm(y ~ x, data=data)

summary(model)

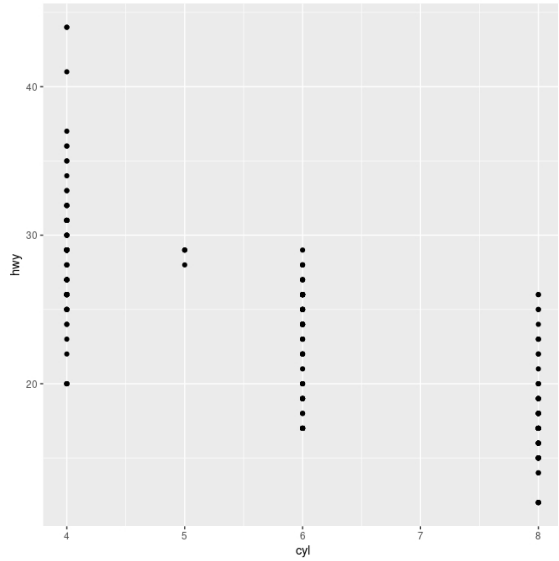
```
> model <- lm(y ~ x, data=data)
> summary(model)

Call:
lm(formula = y ~ x, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-110.889  -19.365   -7.025   17.922   94.145

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.43869    6.68064   2.311   0.0229 *
x            1.14233    0.07407  15.423   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.06 on 98 degrees of freedom
Multiple R-squared:  0.7082,    Adjusted R-squared:  0.7052
F-statistic: 237.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

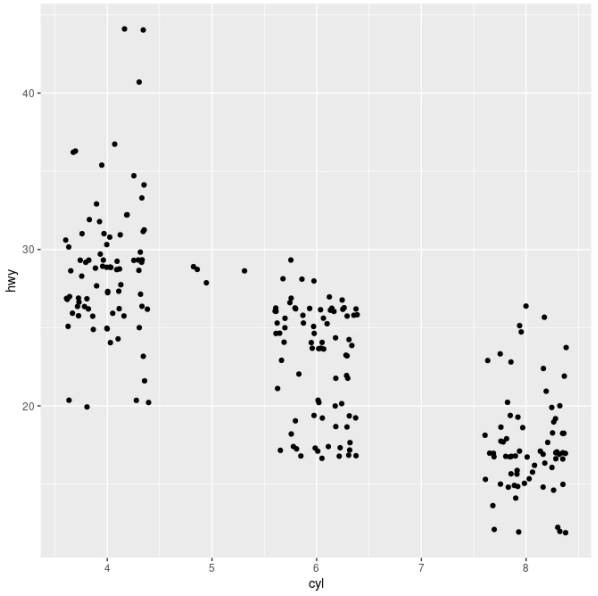


Y = 1.14x +15.4

geom_point () - jitter

Typically a **scatter plot** is **better** when showing the **relationship** between two variables which was not important in this specific case. A **jitter plot** is better to show the **distribution of data** using a randomized x-axis to disperse the points.





data avaible: mpg in R

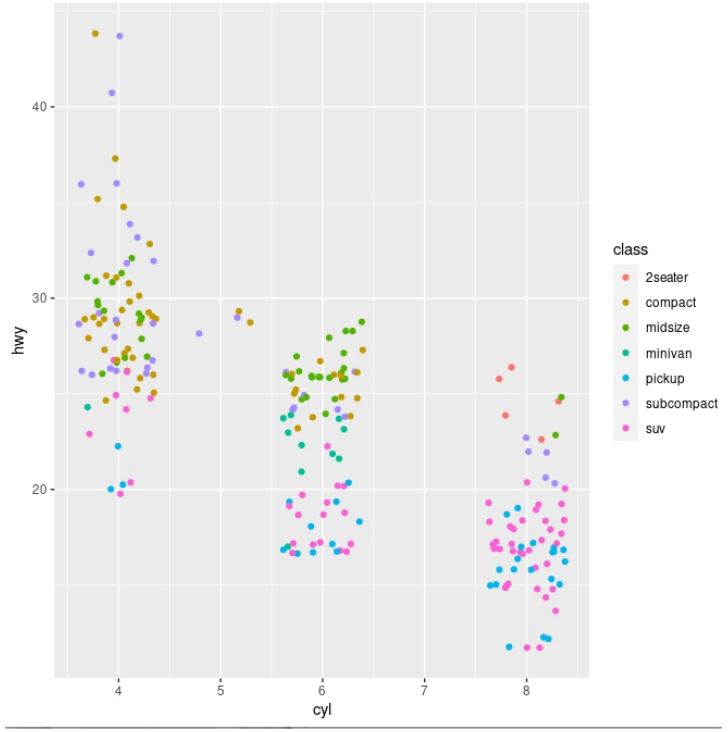
p <- ggplot(mpg, aes(cyl, hwy))

p + geom_point()

p + geom_jitter() [p + geom_point(position="jitter")]

p + geom_jitter(aes(colour = class))

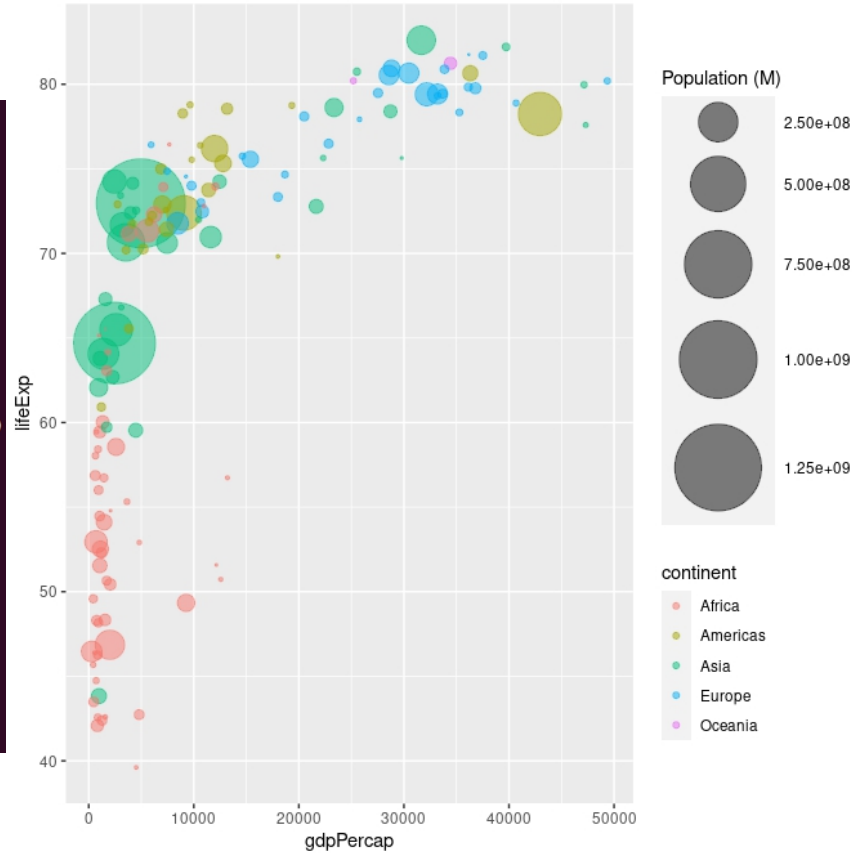
```
> mpg
# A tibble: 234 × 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
 2 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
 3 audi         a4           2    2008     4 manu… f        20    31 p     comp…
 4 audi         a4           2    2008     4 auto… f        21    30 p     comp…
 5 audi         a4           2.8  1999     6 auto… f        16    26 p     comp…
 6 audi         a4           2.8  1999     6 manu… f        18    26 p     comp…
 7 audi         a4           3.1  2008     6 auto… f        18    27 p     comp…
 8 audi         a4 quattro   1.8  1999     4 manu… 4        18    26 p     comp…
 9 audi         a4 quattro   1.8  1999     4 auto… 4        16    25 p     comp…
10 audi         a4 quattro   2    2008     4 manu… 4        20    28 p     comp…
# … with 224 more rows
```



geom_point () - Bubble

```
library(ggplot2)
library(dplyr)
library(gapminder)
data <- gapminder %>%
filter(year=="2007") %>% dplyr::select(-
year)
data %>%
  arrange(desc(pop)) %>%
  mutate(country = factor(country, country)) %>%
  ggplot(aes(x=gdpPercap, y=lifeExp, size=pop, color=continent)) +
    geom_point(alpha=0.5) +
    scale_size(range = c(.1, 24), name="Population (M)")
```

```
> gapminder
# A tibble: 1,704 × 6
   country     continent  year lifeExp      pop gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
 1 Afghanistan Asia       1952    28.8  8425333      779.
 2 Afghanistan Asia       1957    30.3  9240934      821.
 3 Afghanistan Asia       1962    32.0 10267083      853.
 4 Afghanistan Asia       1967    34.0 11537966      836.
 5 Afghanistan Asia       1972    36.1 13079460      740.
 6 Afghanistan Asia       1977    38.4 14880372      786.
 7 Afghanistan Asia       1982    39.9 12881816      978.
 8 Afghanistan Asia       1987    40.8 13867957      852.
 9 Afghanistan Asia       1992    41.7 16317921      649.
10 Afghanistan Asia       1997    41.8 22227415      635.
# … with 1,694 more rows
> data <- gapminder %>% filter(year=="2007") %>% dplyr::select(-year)
> data
# A tibble: 142 × 5
   country     continent lifeExp       pop gdpPercap
   <fct>       <fct>       <dbl>     <int>     <dbl>
 1 Afghanistan Asia         43.8  31889923      975.
 2 Albania     Europe       76.4   3600523     5937.
 3 Algeria     Africa       72.3  33333216     6223.
 4 Angola      Africa       42.7  12420476     4797.
 5 Argentina   Americas     75.3  40301927    12779.
 6 Australia   Oceania      81.2  20434176    34435.
 7 Austria     Europe       79.8   8199783    36126.
 8 Bahrain     Asia         75.6    708573    29796.
 9 Bangladesh  Asia         64.1 150448339     1391.
10 Belgium     Europe       79.4  10392226    33693.
# … with 132 more rows
```



- The first two dimensions for coordinates.
- the thrid dimension is for color and the fourth as size.

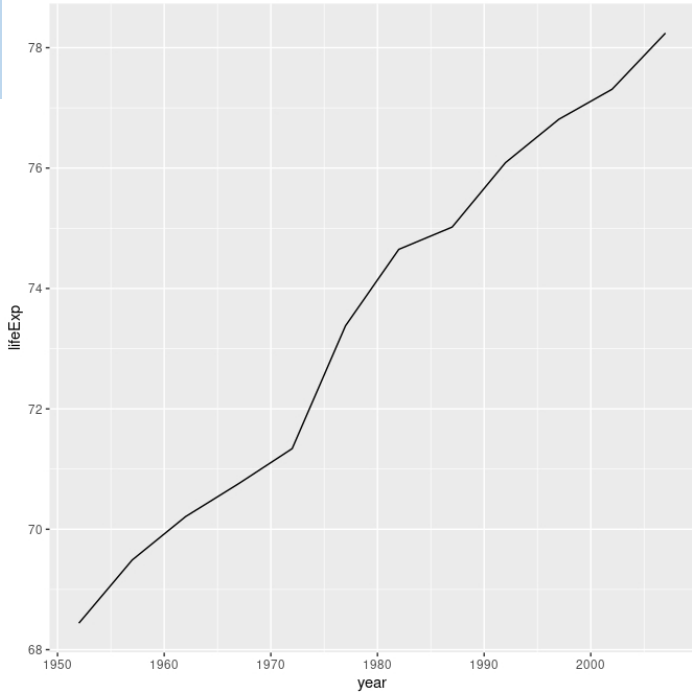
Bivariate Graphs - Quantitative vs. Quantitative

line plot

```
library(gapminder)
library(dplyr)
plotdata <- filter(gapminder,
    country == "United States")
```

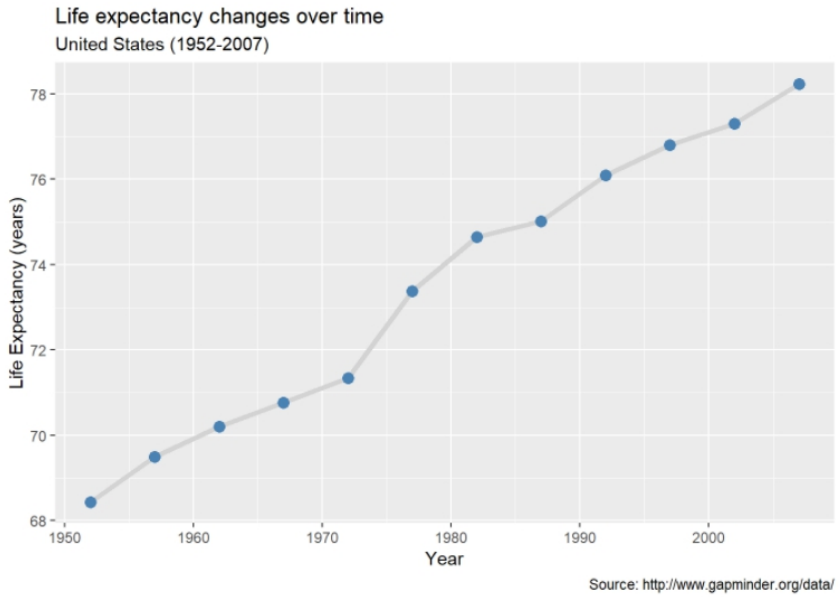
simple line plot

```
ggplot(plotdata,
       aes(x = year,
           y = lifeExp)) + geom_line()
```



line plot with points

```
ggplot(plotdata,
       aes(x = year,
           y = lifeExp)) +
  geom_line(size = 1.5,
            color = "lightgrey") +
  geom_point(size = 3,
             color = "steelblue") +
  labs(y = "Life Expectancy (years)",
       x = "Year",
       title = "Life expectancy changes
over time",
       subtitle = "United States (1952-
2007)",
       caption = "Source:
http://www.gapminder.org/data/")
```

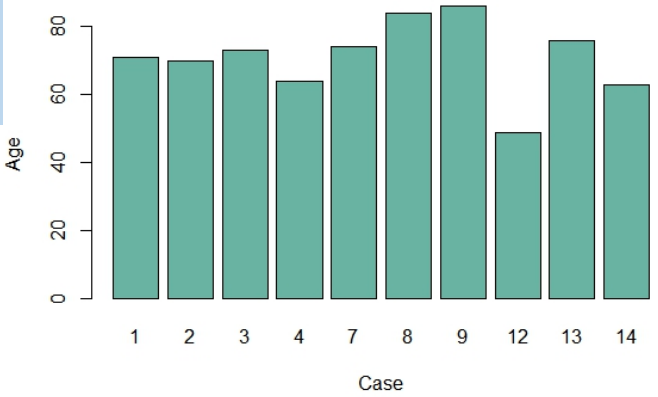


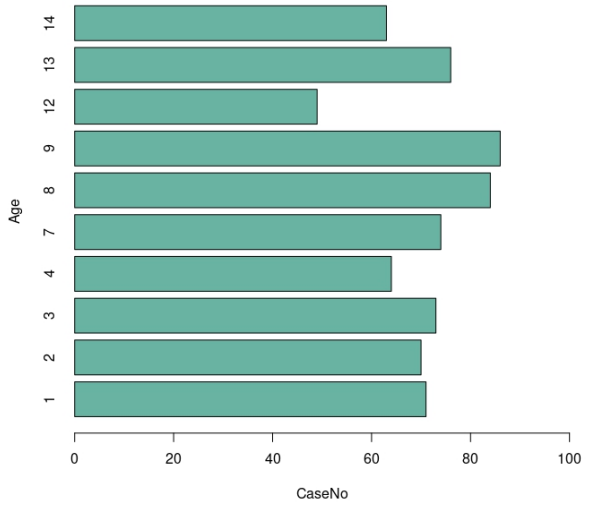
Bivariate Graphs - Categorical vs. Quantitative

barplot

```
data <-
read.csv("/home/acer/Documents/DATA/DATA_VISUALIZATION
/data/clinical.csv",header=TRUE)

b <- subset (data[1:10,])

barplot(height =b$Age, names = b$CaseNo,col="#69b3a2", xlab
= "CaseNo", ylab = "Age")

barplot(height =b$Age, names = b$CaseNo,col="#69b3a2", xlab
= "CaseNo", ylab = "Age",xlim=c(0,100),horiz=T)
```

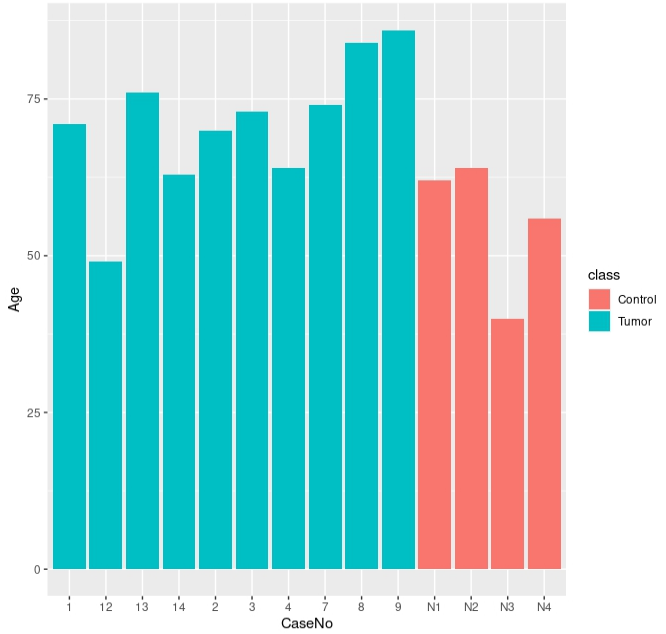
```
> data
   CaseNo Gender Age Survival.Months.
1       1      F  71                 1
2       2      M  70                13
3       3      F  73                 8
4       4      M  64                14
5       7      F  74                 4
6       8      F  84                 0
7       9      F  86                 1
8      12      M  49                15
9      13      M  76                 1
10     14      F  63                 7
11     N1      F  62               N/A
12     N2      M  64               N/A
13     N3      F  40               N/A
14     N4      F  56               N/A
> b <-table(data$Gender)
> b

F M
9 5
```





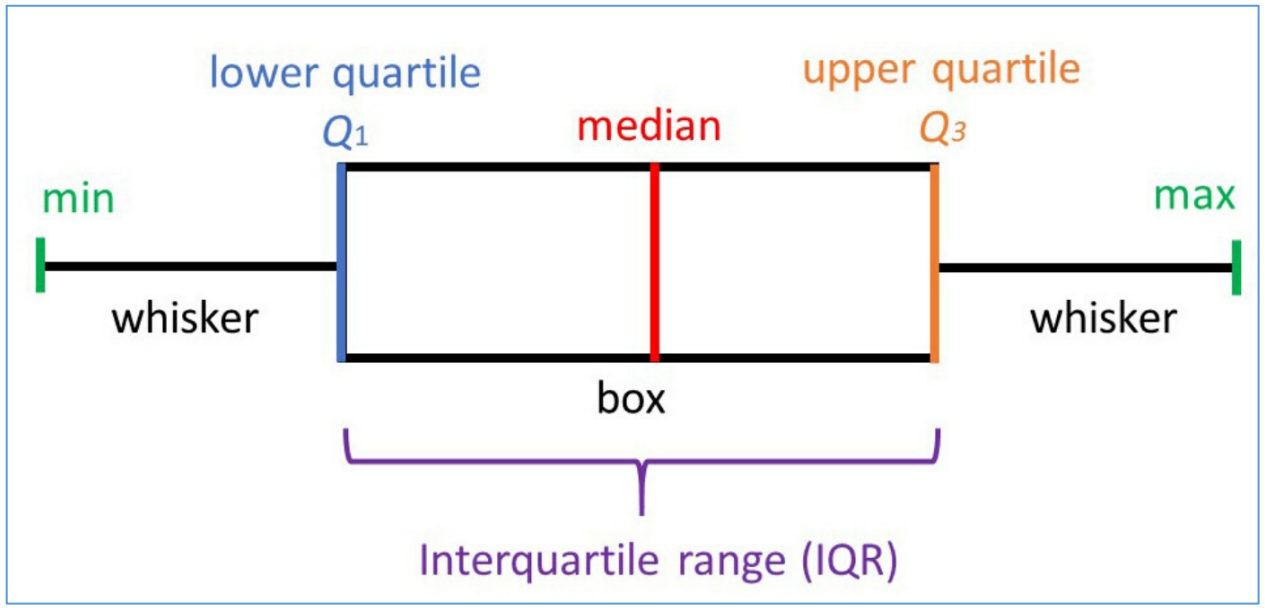
```
data$class <-
ifelse(data$CaseNo=="N1"|data$CaseNo=="N2"|data$CaseNo=
="N3"|data$CaseNo=="N4","Control",”Tumor”)

data$CaseNo <- factor (data$CaseNo,
levels=c(1,2,3,4,7,8,9,12,13,14,"N1","N2","N3","N4"))

ggplot(data,aes(y=Age,x=CaseNo,fill=class))+
geom_bar(stat="identity")
```

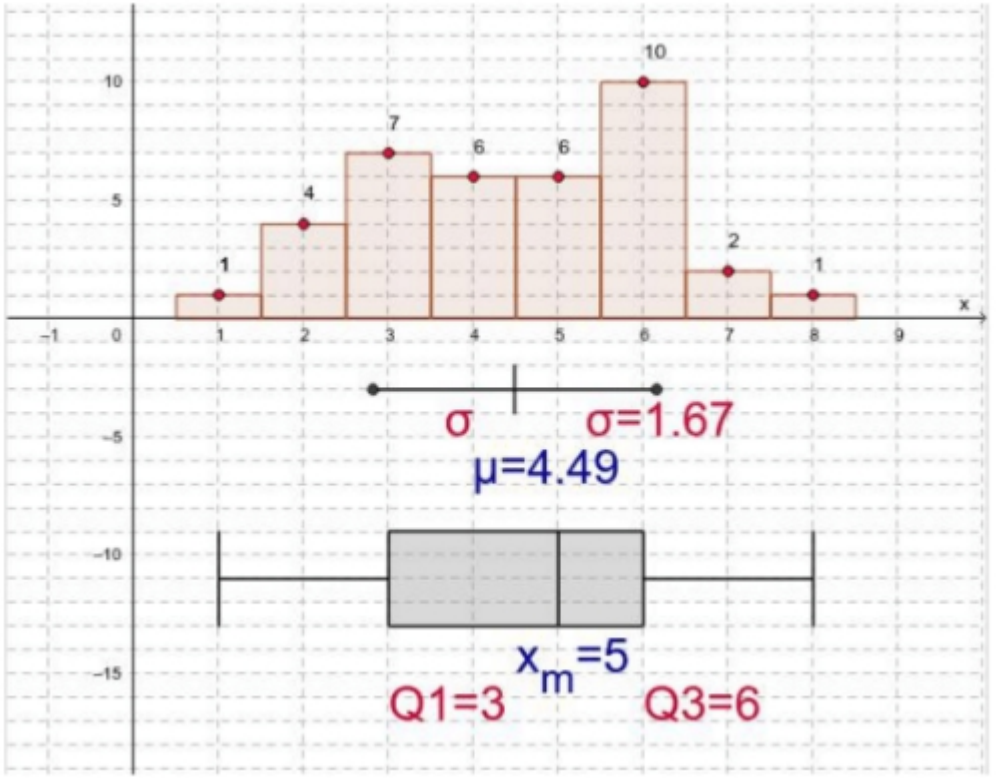
```
> data
   CaseNo Gender Age Survival.Months.   class
1       1      F  71                 1   Tumor
2       2      M  70                13   Tumor
3       3      F  73                 8   Tumor
4       4      M  64                14   Tumor
5       7      F  74                 4   Tumor
6       8      F  84                 0   Tumor
7       9      F  86                 1   Tumor
8      12      M  49                15   Tumor
9      13      M  76                 1   Tumor
10     14      F  63                 7   Tumor
11     N1      F  62               N/A Control
12     N2      M  64               N/A Control
13     N3      F  40               N/A Control
14     N4      F  56               N/A Control
```



Bivariate Graphs - Categorical vs. Quantitative

Boxplot



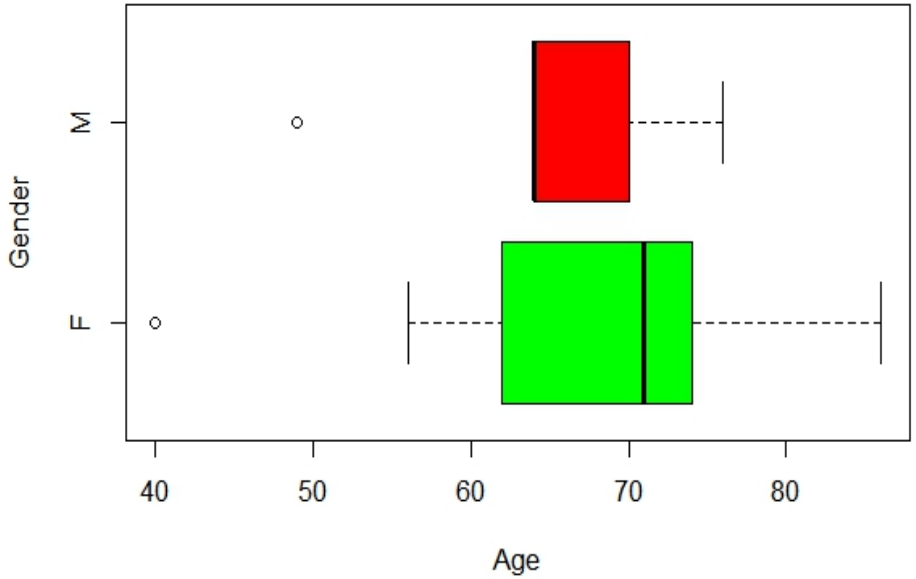


Bivariate Graphs - Categorical vs. Quantitative

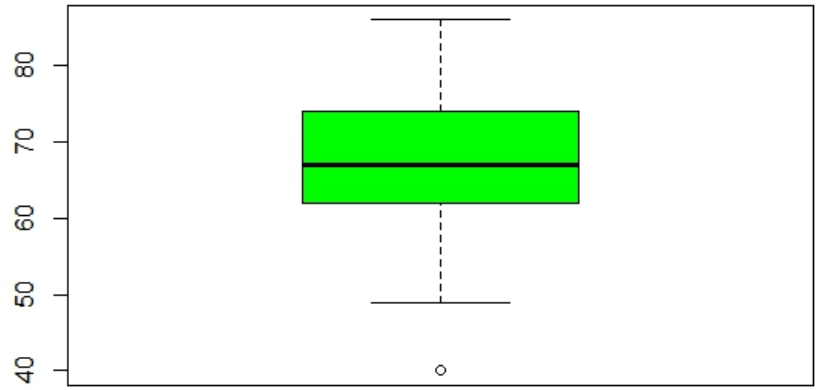
boxplot (**var**, xlab, ylab, main, border, col, xlim, ylim, horizontal)

boxplot (**var** ~ group, xlab, ylab,main, xlim, ylim, border, col, horizontal)

```
head(clinical)
boxplot(clinical$Age,col="green",xlab="Age")
boxplot(clinical$Age ~
clinical$Gender,col=c("green","red"),xlab = "Age", ylab =
"Gender",horizontal = T)
```



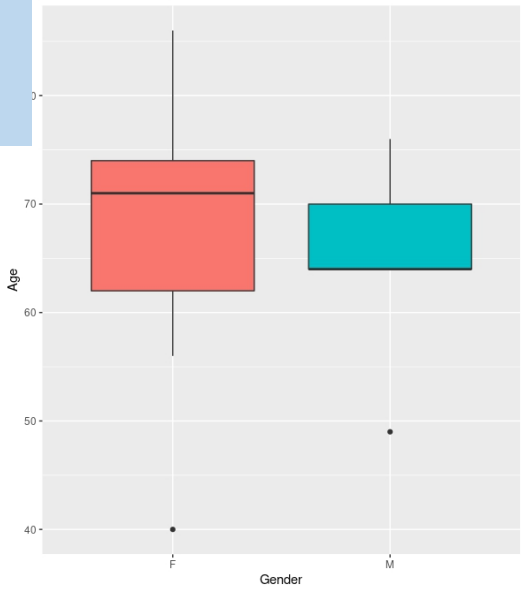

```
> head(clinical)
  CaseNo Gender Age Survival.Months.
1      1      F  71                1
2      2      M  70               13
3      3      F  73                8
4      4      M  64               14
5      7      F  74                4
6      8      F  84                0
```



Bivariate Graphs - Categorical vs. Quantitative

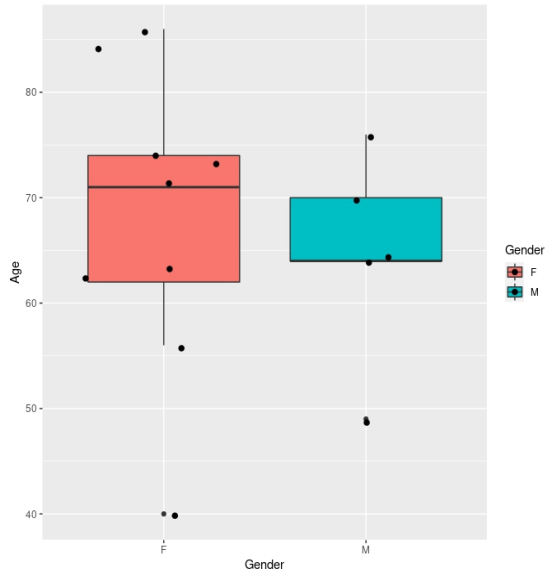
Basic boxplot

```
ggplot(data,aes(y=Age, x=Gender,fill=Gender))+geom_boxplot()
```



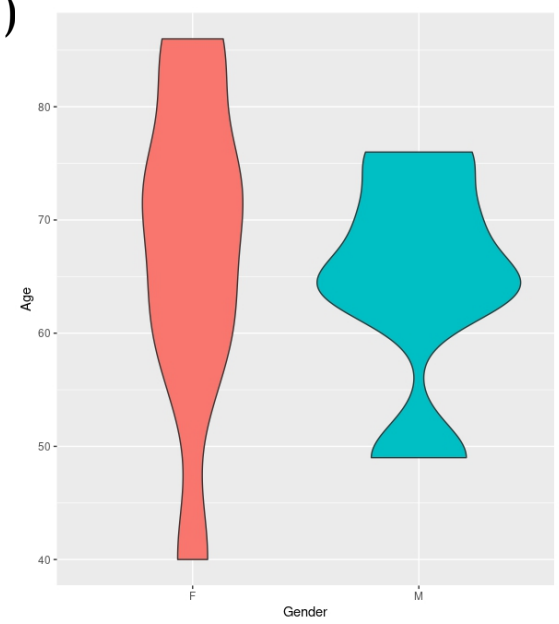
Boxplot with individual data points

```
ggplot(data,aes(y=Age,
x=Gender,fill=Gender))+geom_boxplot()+geom_jitter(size=1.95)
```



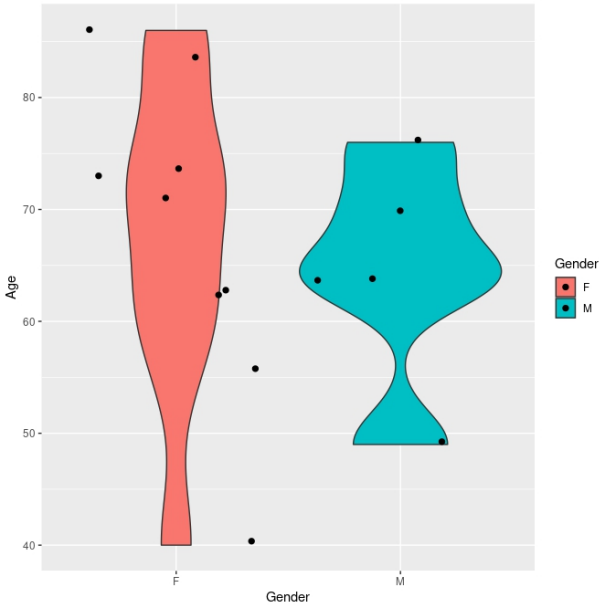
Basic Violint

```
ggplot(data,aes(y=Age, x=Gender,fill=Gender))+geom_violin()
```



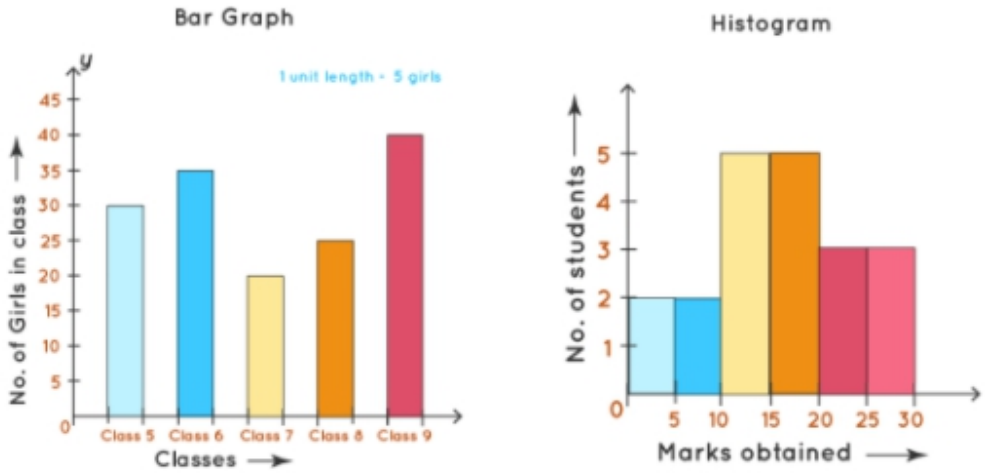
Violint with jitter

```
ggplot(data,aes(y=Age,
x=Gender,fill=Gender))+geom_violin()+geom_jitter(size=1.9)
```



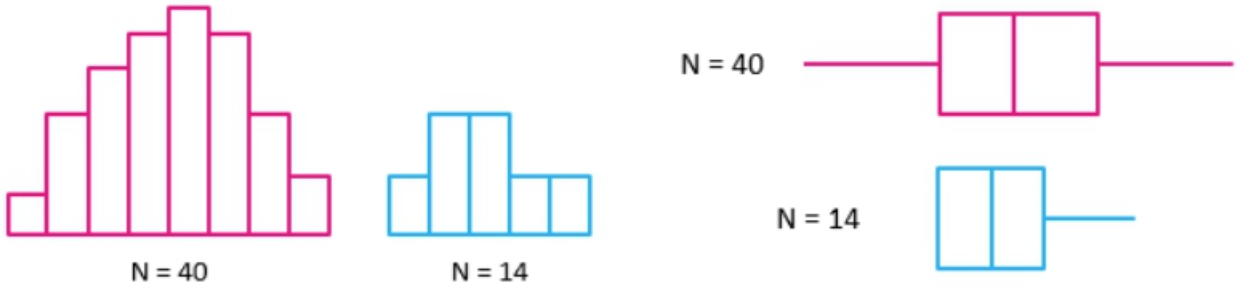
HISTOGRAM AND BARPLOT

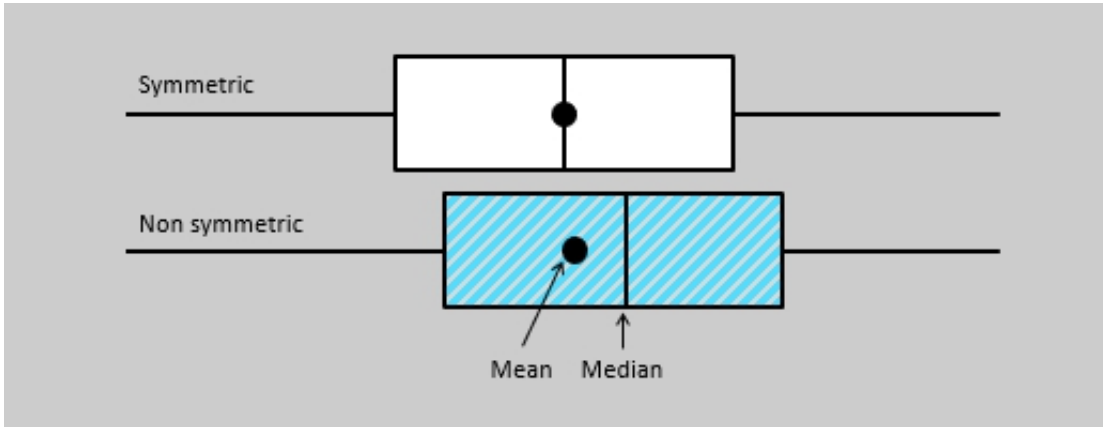
Histogram	Barplot
Show distributions of variables	Compare variables
Quantitative (numerical) data (continuous)	Categorical data
Not make sense to rearrange	The bars can be rearranged
No space between the columns	Have space between the columns



HISTOGRAM AND BOXPLOT

Histogram	Boxplot
probability distribution of a data	Comparing between several data sets. (distribution is symmetric or skewed)
Min, mean, max	Min, median, max and quartile





Although histograms are better in determining the underlying distribution of the data, box plots allow you to compare multiple data sets better than histograms as they are less detailed and take up less space.