Data Science and R

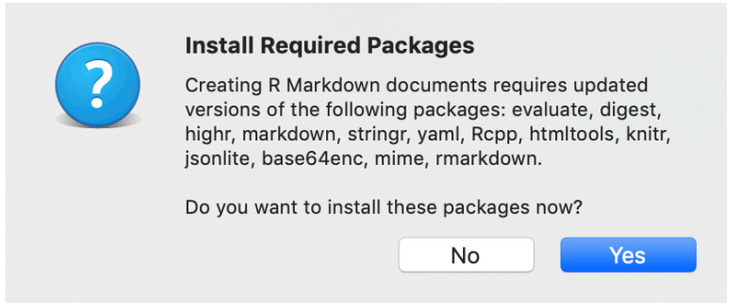
5.2. R Markdowns, Basic Plotting

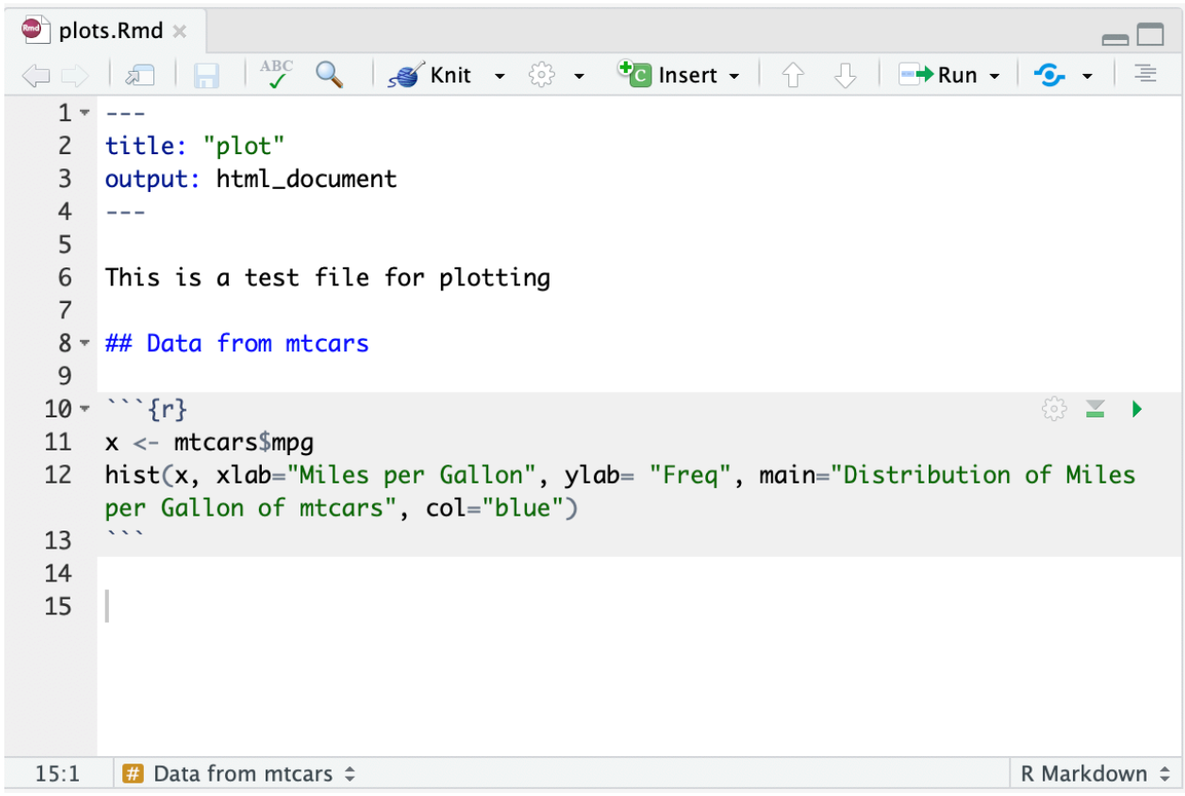
Recap Packages, Iterations, Conditionals

- Installing and loading packages
  - `install.packages("MASS")`
  - `library(MASS)  # load MASS into memory`
- Typing `data()` now should give an additional bunch of datasets in the 'MASS' package. To load one of the datasets, e.g. `geyser`, the command is:
  - `data(geyser)`
- Iterations in the form of for loops lets us define the sequence (number range, membership), body (what happens at each iteration), and any output
- Conditionals
  - `if(test-statement) { #do something } else { #do something else}`
  - `ifelse(condition, action_if_true, action_if_false)`

R Markdowns

- File that have extension `.Rmd`
- Can use it to generate report in html, pdf or Word
- Can write code chunks and execute multiple lines of code at once
- `File->New File->R Markdown`
- Choose HTML for now
- If R Markdown is not available, install it
- `install.packages("rmarkdown")`
- The file has three parts:
  - A header (surrounded by – – –)
  - R code chunks (surrounded by ```)
  - Text
- Run using ONE of these options
  - Pressing `Play` at code chunk
  - `Run -> Run Current Chunk`
  - `Shift + Ctrl + Enter`



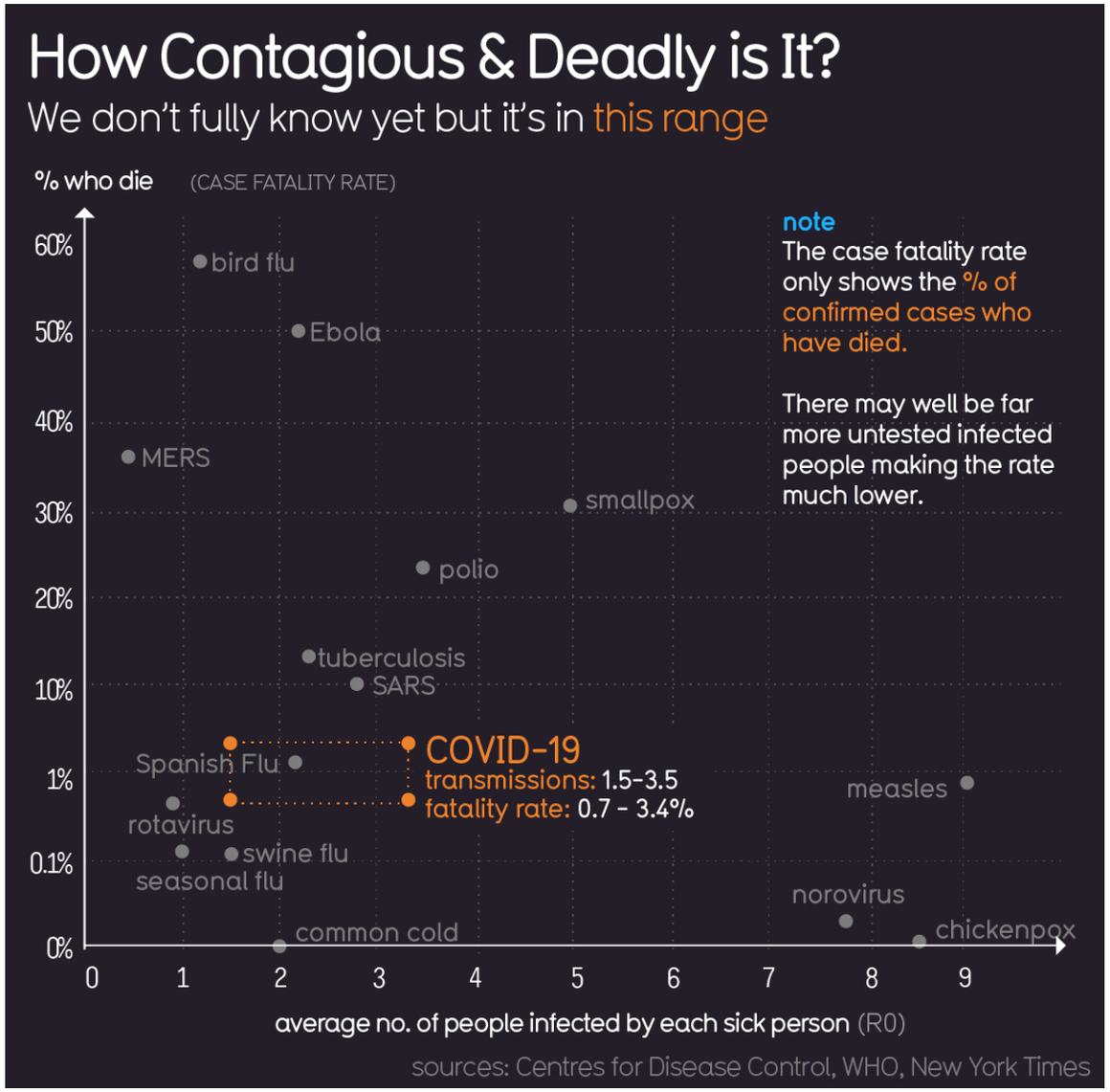


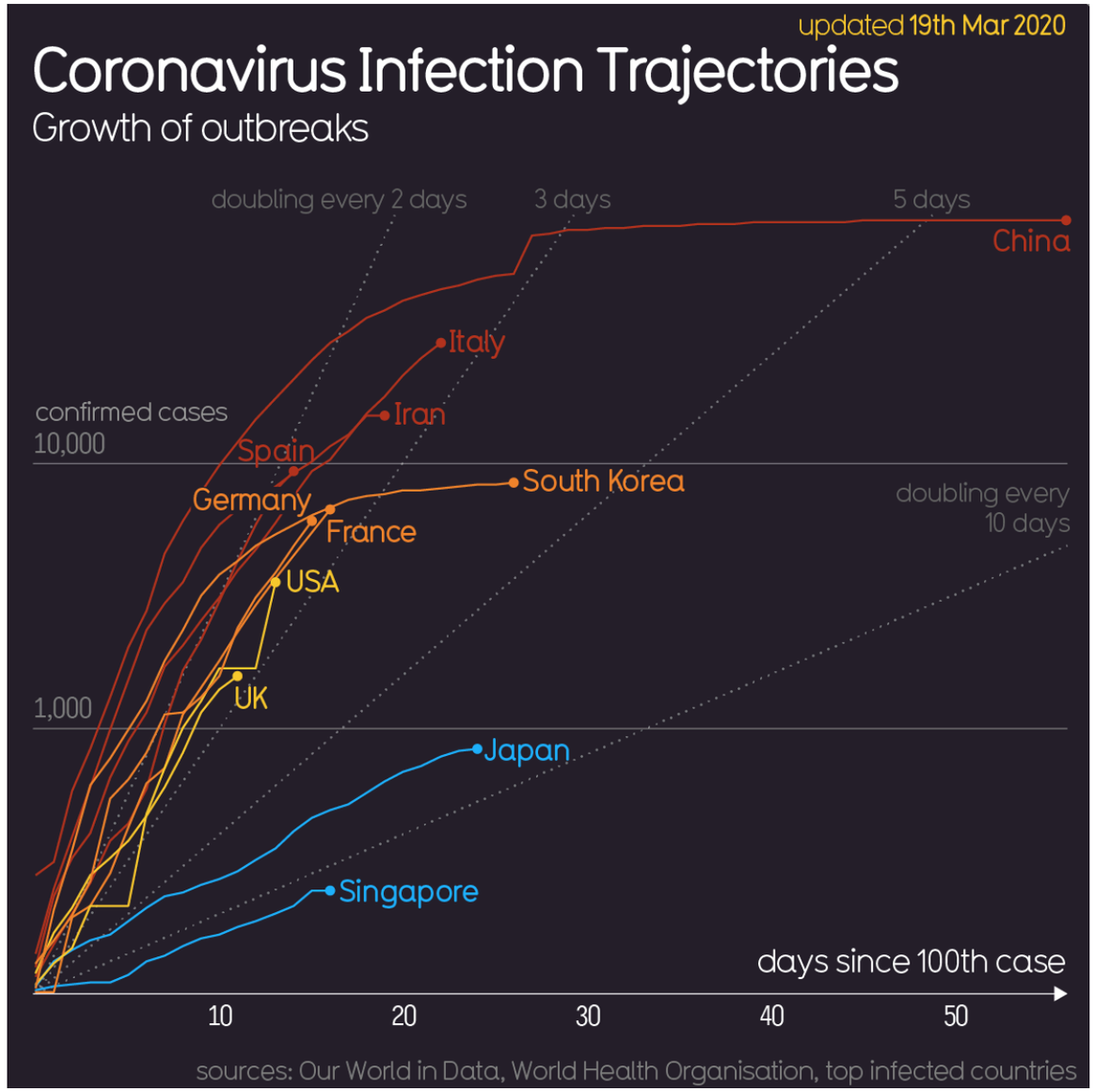
Data is just (Structured) Text

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.32					
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.21					
Hornet Sportabout	18.7	8	360.0	175	3.15	3.44					
Valiant	18.1	6	225.0	105	2.76	3.46					
Duster 360	14.3	8	360.0	245	3.21	3.57					
Merc 240D	24.4	4	146.7	62	3.69	3.19					
Merc 230	22.8	4	140.8	95	3.92	3.15					
Merc 280	19.2	6	167.6	123	3.92	3.44					
Merc 280C	17.8	6	167.6	123	3.92	3.44					
Merc 450SE	16.4	8	275.8	180	3.07	4.07					
Merc 450SL	17.3	8	275.8	180	3.07	3.73					
Merc 450SLC	15.2	8	275.8	180	3.07	3.78					
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.25					
Lincoln Continental	10.4	8	460.0	215	3.00	5.42					

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa

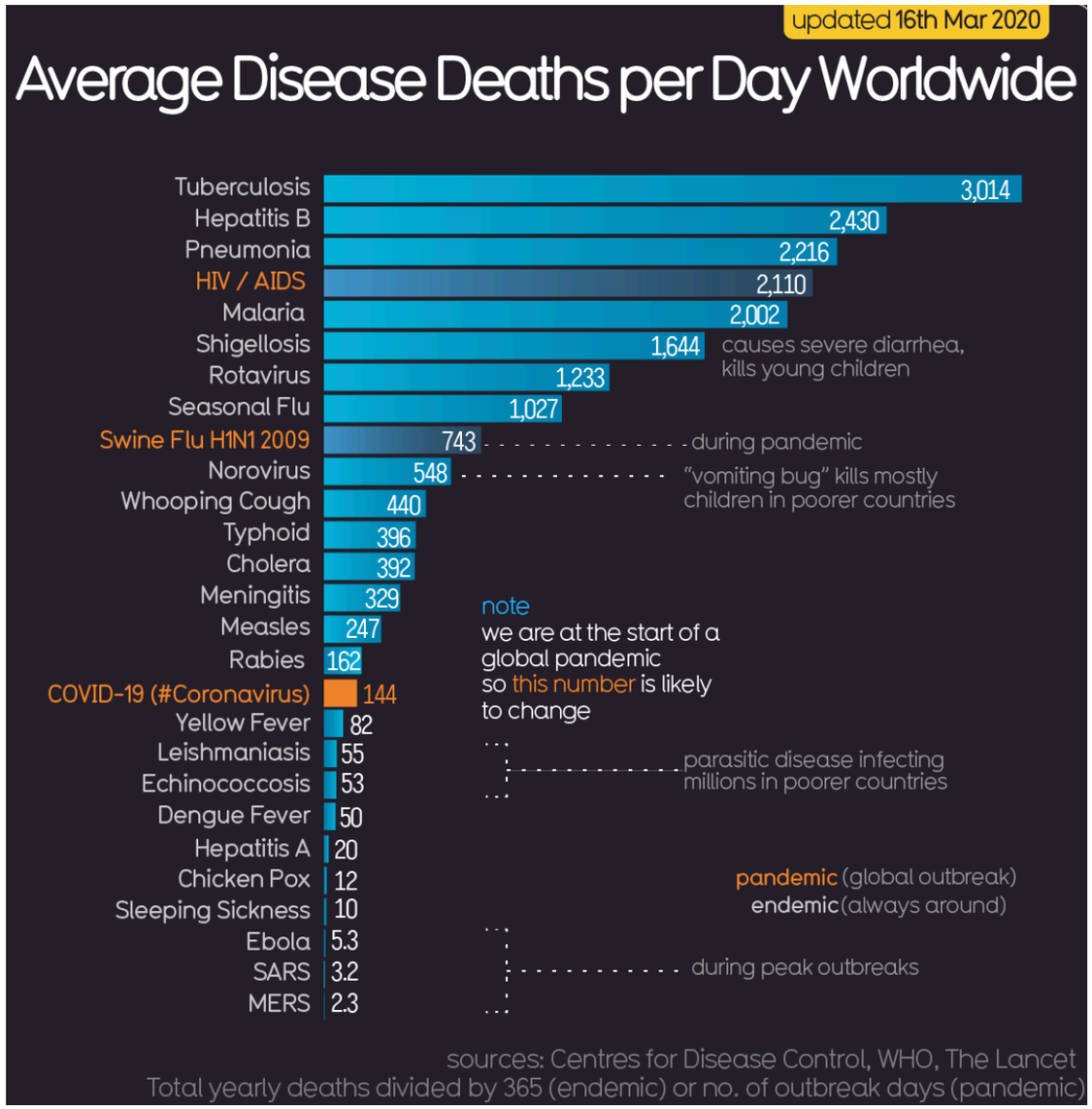
Visualisation of Textual Data Makes it More Readable (e.g. COVID-19)

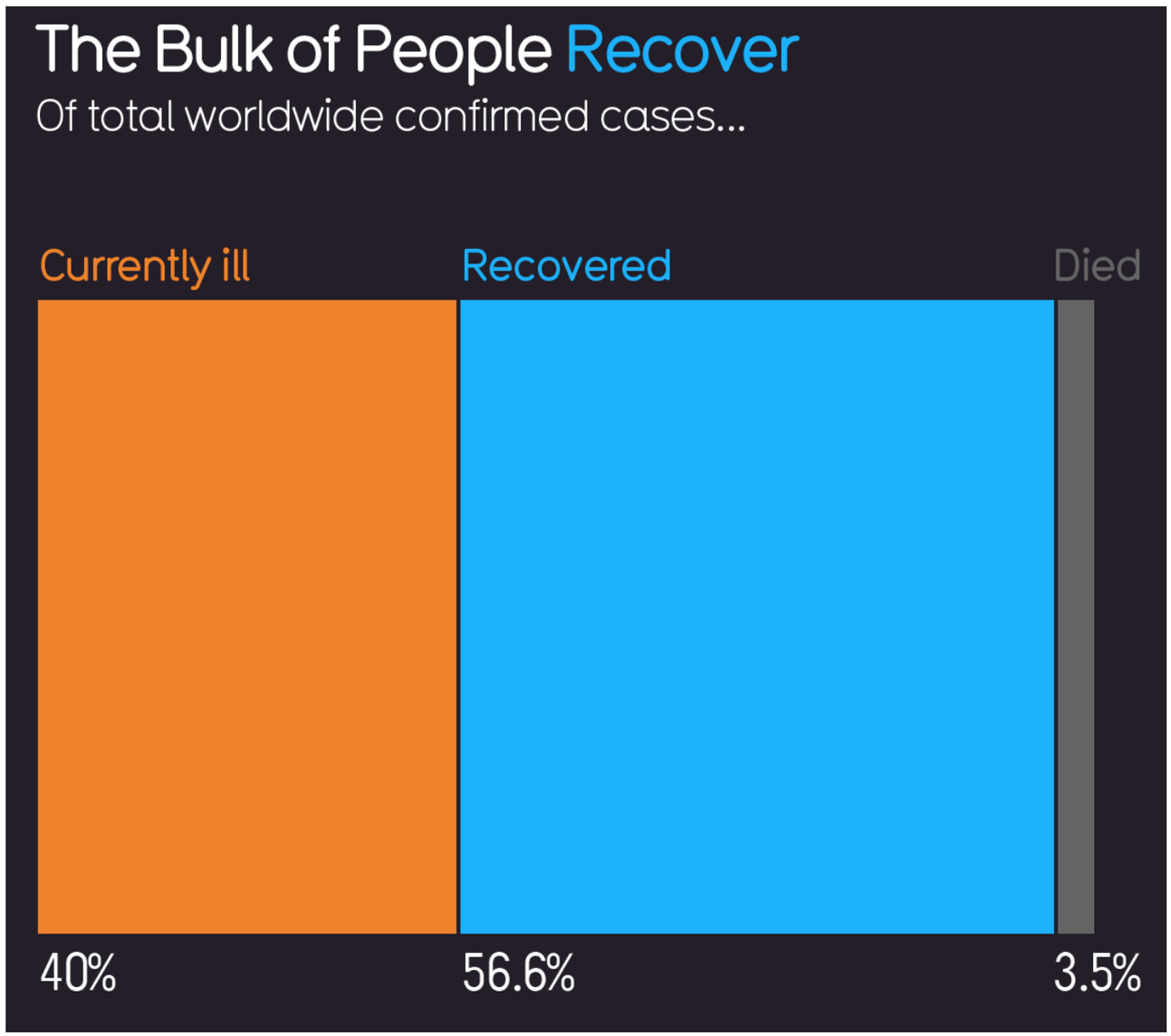




Source: informationisbeautiful.net

Covid-19 Visualisations





Source: informationisbeautiful.net

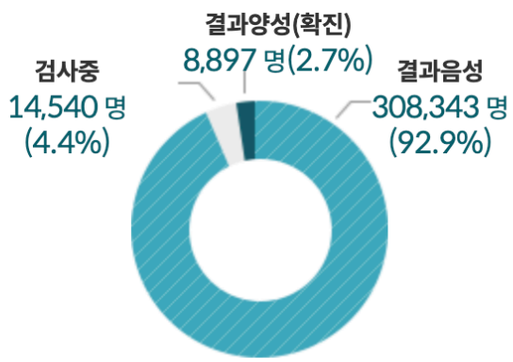
Covid-19 in Korea

확진환자	완치 (격리해제)	치료 중 (격리 중)	사망
(누적)8,897	2,909	5,884	104
전일대비 (+ 98)	(+ 297)	(- 201)	(+ 2)

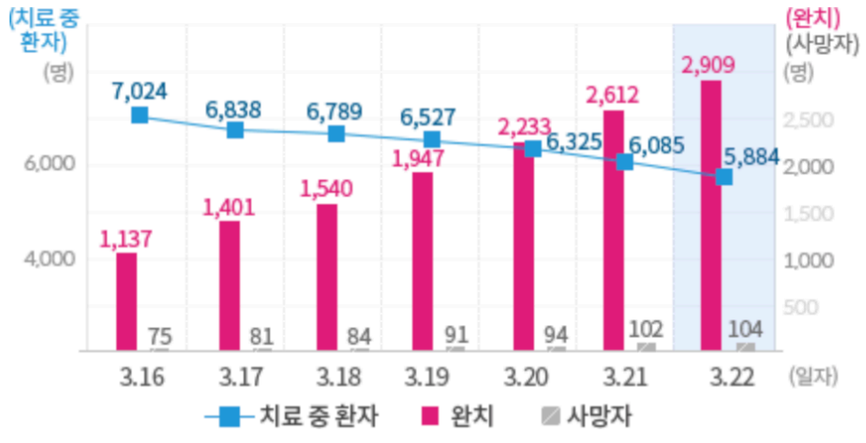
검사현황

누적 검사수	331,780 건
누적 검사완료수	317,240 건
누적 확진률	2.8 %

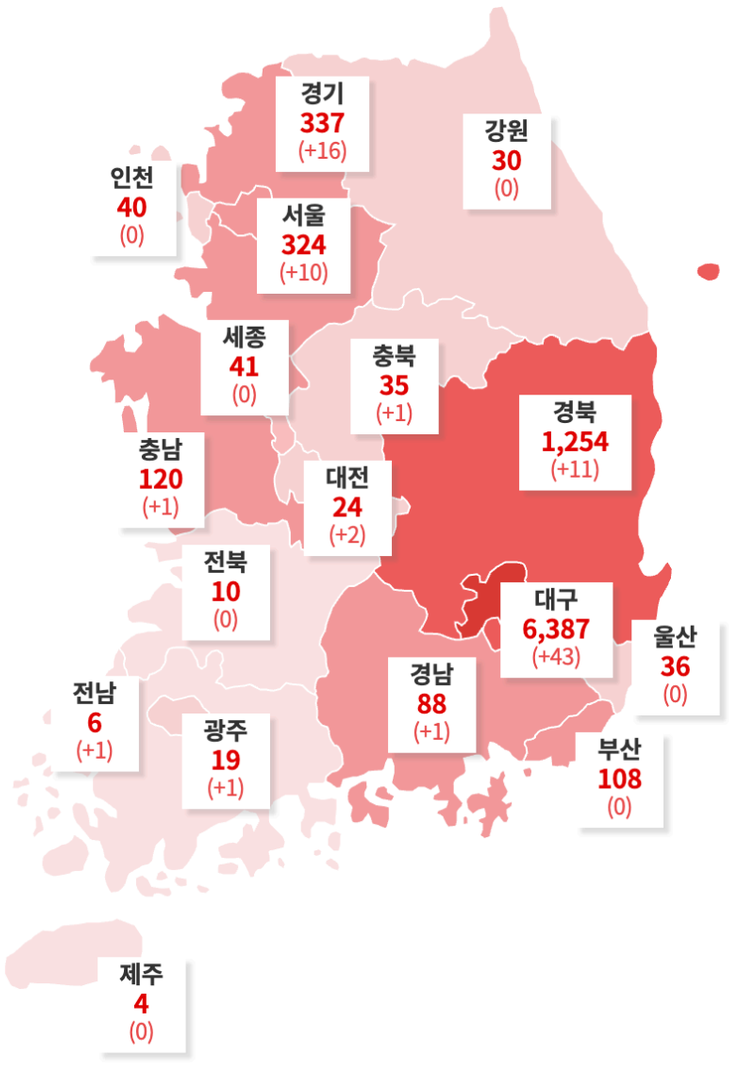
(결과양성 / 총 검사완료수 * 100%)



주간 환자동향 (치료 중 환자, 완치, 사망자)

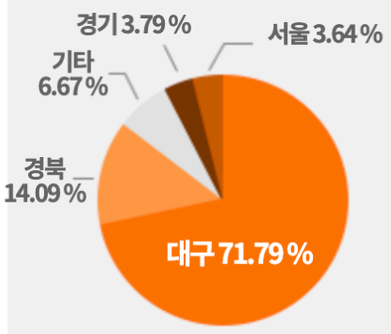


시도를 클릭하시면 상세 현황을 확인할 수 있습니다.
( ) 숫자는 전일대비 증감수치



전국

지역 발생비율



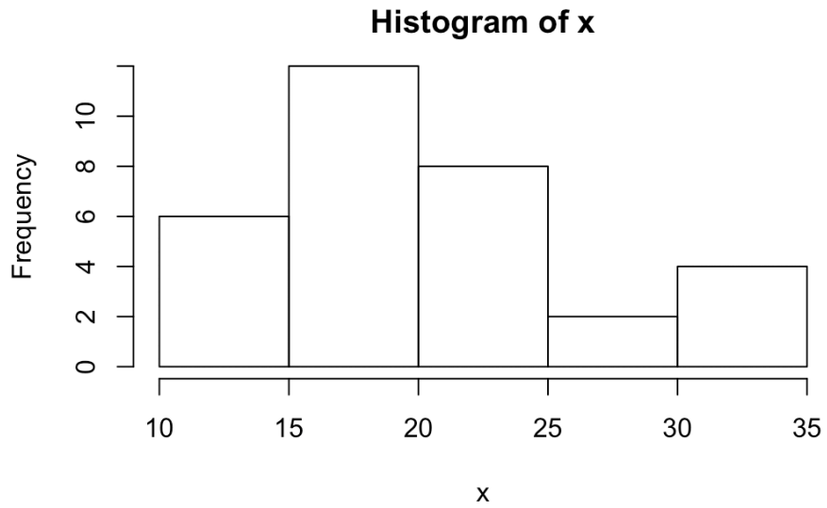
누적 확진환자	8,897 명
전일 대비 증감	(+98)
사망자	104 명
누적 격리해제	2,909 명
10만명당 발생률	17.16 명

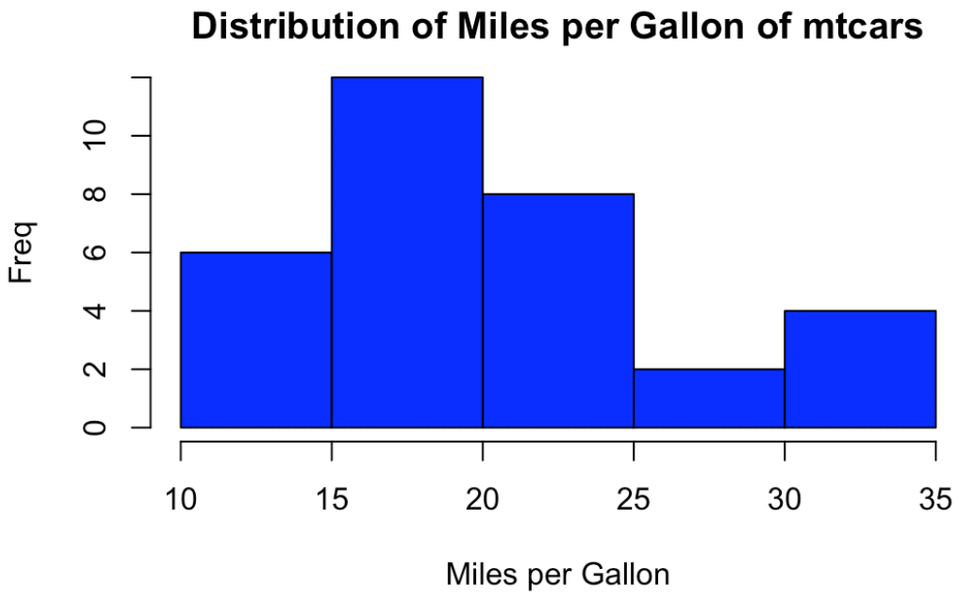
① **발생률 :** 지역별 인구 출처 - 행정안전부, 주민등록인구현황 ('20.1월 기준)

Source: http://ncov.mohw.go.kr/

Basic Plotting: `hist()`

- To plot a histogram of a set of numerical data, use `hist()`
  - **`hist(rnorm(100))`** `# 100 random numbers for a standard normal distribution with mean 0 and standard deviation 1`
  - `x <- sample(1:10, 100, replace=T)`
  - `hist(x)`
- Adding labels and colours
- `x <- mtcars$mpg`
- `hist(x, xlab="Miles per Gallon", ylab="Freq", main="Distribution of Mile per Gallon from mtcars", col="blue")`

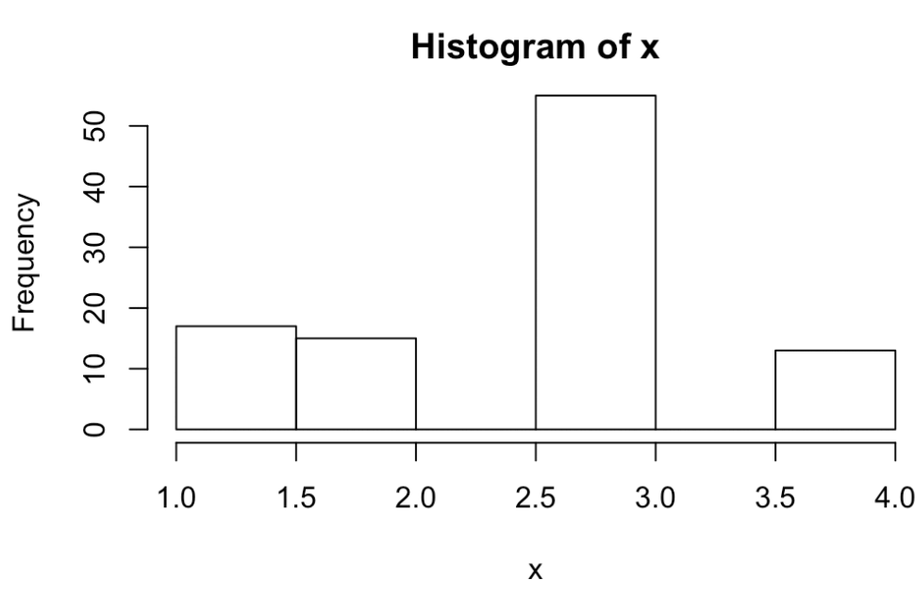




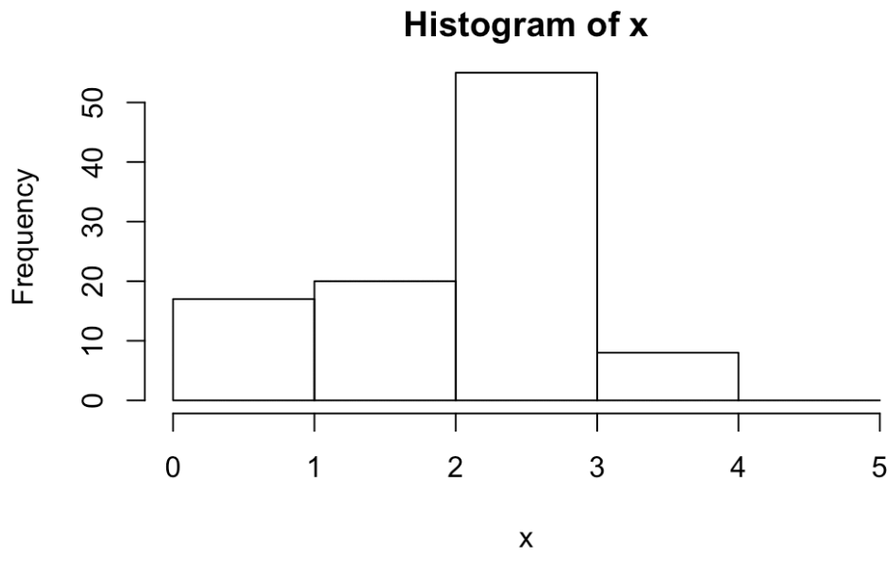
Histogram: `breaks`

- Specify the bin width using the `breaks` parameter
- Can also rename the values on the x or y axes (Lab sheet)

```{r}
x <- sample(1:5, 100, replace=T, prob = c(0.2, 0.2, 0.5, 0.1, 0.0))
hist(x)
```




```{r}
x <- sample(1:5, 100, replace=T, prob = c(0.2, 0.2, 0.5, 0.1, 0.0))
hist(x, breaks = c(0,1,2,3,4,5))
```



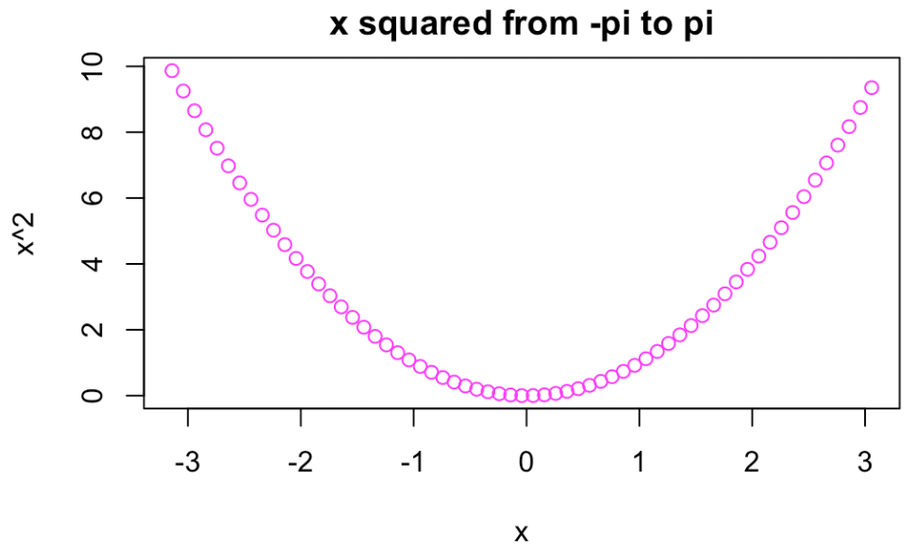
- Many ways to specify breaks, could also pass in a single number to specify the number of breaks needed
- Here breaks are specified at specific values of x

Basic Plotting: `plot()`

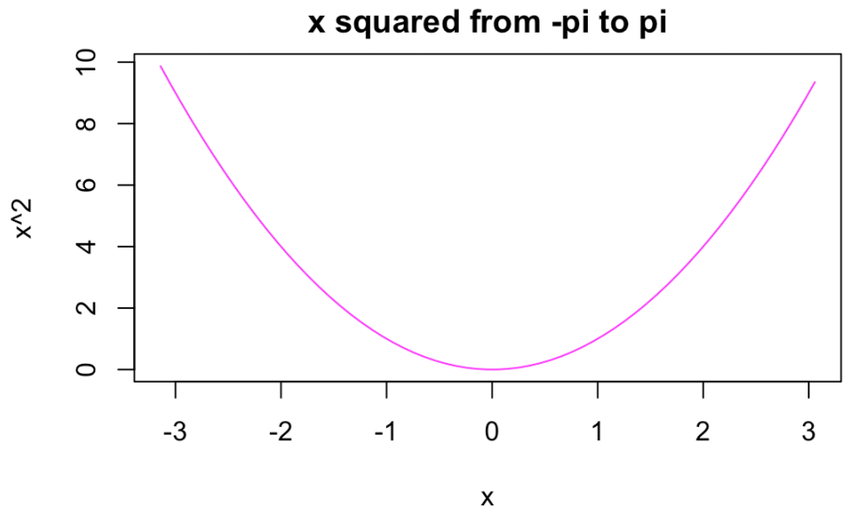
- The `plot()` function is the main plotting function to draw all kinds of graphs in R
- Can change the plot type by using the `type` parameter
- Simplest case is passing one vector, but normally we pass in two vectors, for the x and y axis
- Options for `type`:
  - “p” points

```
"p" - points
"l" - lines
"b" - both points and lines
"c" - empty points joined by lines
"o" - overplotted points and lines
"s" and "S" - stair steps
"h" - histogram-like vertical lines
"n" - does not produce any points or lines
```

```{r}
x <- seq(-pi,pi,0.1)
plot(x, x^2, main = "x squared from -pi to pi", col="magenta")
```



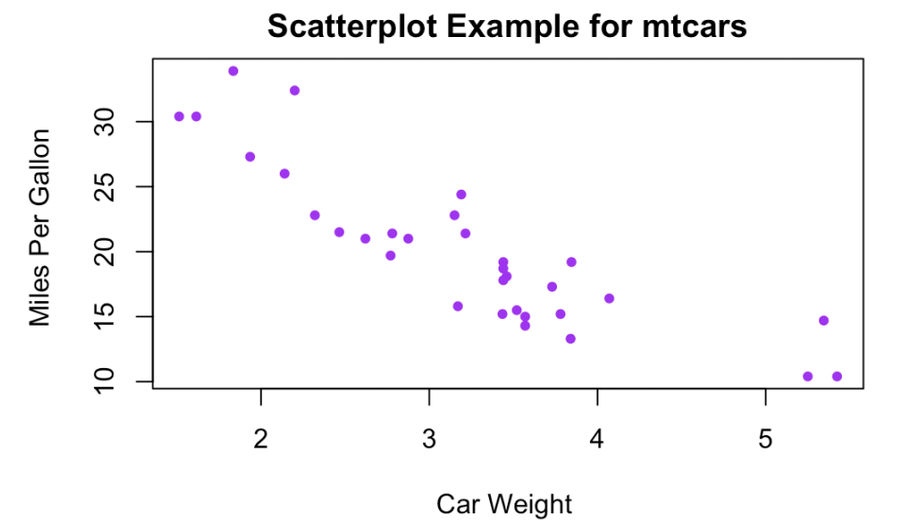

```{r}
x <- seq(-pi,pi,0.1)
plot(x, x^2, main = "x squared from -pi to pi", type = "l", col="magenta")
```



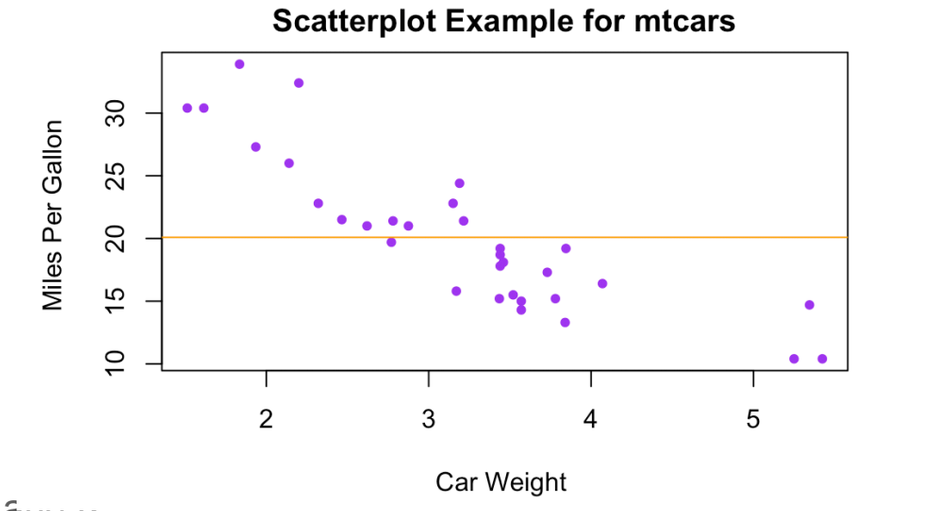
Example: `mtcars`

- The weight of the car (wt) versus the miles per gallon (mpg) are plotted
- With a few added details
  - Symbols for points can be specified using the plotting character `pch` parameter
  - Added a line for mean of mpg using `abline()` function
  - Can also change line type in `abline` by adding `lty` (1-6) parameter

```r
with(mtcars, plot(wt, mpg, main="Scatterplot Example for mtcars",
   xlab="Car Weight ", ylab="Miles Per Gallon ", pch=20, col="purple"))
```




```r
with(mtcars, plot(wt, mpg, main="Scatterplot Example for mtcars",
   xlab="Car Weight ", ylab="Miles Per Gallon ", pch=20, col="purple"))
abline(h=mean(mtcars$mpg), col="orange")
```



Tips & Links

- R Markdown: https://rmarkdown.rstudio.com/lesson-1.html
- You could also include packages by going to the “Package” tab in the “Help” panel (bottom right) and ticking on the packages that you want OR
- Using file menu `Tools -> Install Packages` and typing the package name and click on `Install dependencies`
- List of graphical parameters (`pch,lty,col`) [Link]
- `colors()` OR `colours()` returns all the colours available
- `example(plot)  # shows a few example plots`
- `example(point)  # shows a few example scatter plots`