

# SciViews :: CHEAT SHEET



## SciViews :: R

**SciViews::R** offers additional functions on top of <https://tidyverse.org>. To use them just type:

```
SciViews::R
```

```
# For a better help, replace ?topic by  
?.?topic or about("topic")
```

## Read datasets

**read()** unifies the data importation methods and also loads datasets from R packages.

```
ub <- read("urchin_bio", package = "data.io",  
lang = "fr") - Load data from a package  
ub1 <- read("file.csv") - Import local data  
ub1 <- read$csv2("file.csv") or  
ub1 <- read("file.csv", type = "csv2") - Import  
local data with explicit format specification
```

**write()** for data exportation (always explicit).

```
write$csv(x, file = "filename.csv")
```

**read()** and **write()** support many formats: .txt, .rds, .xls(x), .sas, ...

```
data_types() - List supported data formats
```

## Workflow


Functions are building blocks. They can be *nested*, *piped* (`%>%` operator), or used in *successive statements*. A pipeline is usually more readable.

```
ub <- read("urchin_bio", package = "data.io")
```

```
• Successive statements: select then filter data  
ub1 <- select(ub, 1:5)  
ub2 <- filter(ub1, origin == "Farm")
```

```
• Nesting functions  
ub2 <- filter(select(ub, 1:5),  
origin == "Farm")
```

```
• Pipeline with flow  
ub %>%  
  select(., 1:5) %>%  
  filter(., origin == "Farm") -> ub2
```

 `%>%` is an explicit pipe (dot must be specified). Less ambiguous than tidyverse's pipe `%>%`.

## Data visualisation

**chart()** uses four rules against **ggplot()**

```
ub <- read("urchin_bio", package = "data.io")  
ggplot(data = ub, mapping = aes(x = weight, y = height, colour =  
origin)) +  
  geom_point()
```

1. Replace **ggplot()** by **chart()**

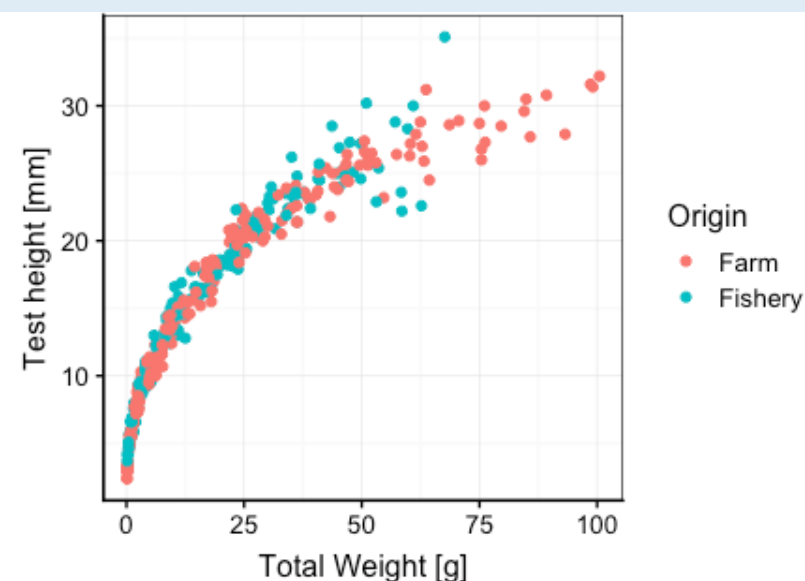
```
chart(data = ub, aes(x = weight, y = height, colour = origin)) +  
  geom_point()
```

2. Replace argument **aes()** by **f\_aes()** to use a formula instead

```
ggplot(data = ub, f_aes(height ~ weight %col=% origin)) +  
  geom_point()
```

3. Even better: use **chart()** with formula syntax directly

```
chart(data = ub, height ~ weight %col=% origin) +  
  geom_point()
```



In addition **chart()** uses associated metadata (labels and units) to provide a plot close to publication ready.

```
ggplot(data = ub, mapping = aes(x = weight, y = height)) +  
  geom_point() +  
  facet_grid(~ origin)
```

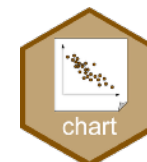
4. Use facets (<https://ggplot2-book.org/facet.html>) in the formula

```
chart(data = ub, height ~ weight | origin) +  
  geom_point()
```

**chart()** provides a unified interface for base plots, lattice and ggplot2 with argument **type =** or with **\$**

```
chart(data = ub, height ~ weight | origin, type =  
"geom_point") - type = e.g. "xyplot" for lattice plot
```

```
chart$xyplot(data = ub, height ~ weight | origin)
```



## Reproducible research

Respect the three rules below for reproducible works:

1. Organise your analyses in **(RStudio) Project** with a README(.md) file and all other files in dedicated sub-directories, e.g.:
  - **data** sub-dir: all datasets (also **data-raw**)
  - **R** sub-dir: all R scripts
  - **docs** sub-dir: all reports, presentations, ...
  - **notes** sub-dir: your notebooks
2. Make your project **portable**:
  - Use only *relative* paths, or use **here::here()**
3. Use a **version control** system like git (and GitHub, Gitlab, ...)

## SciViews snippets

The **SciViews Snippets** in RStudio are organised in a succession of drop-down menus. Use them to recall code syntax.



...	First level	Second level	Description
→	..d dataframes	.dm management	Import and export data
		.dr reshape	Rename and arrange columns
		.do observations	Extract rows
		.dv variables	Extract and compute new columns
		.ds summarise	Summarise a dataset
		.dg group data	Group data by factor
		.dc combine	Combine several datasets
→	..e exploratory stats	.es summary	Summarise a dataset
		.ec contingency	Create and visualise a contingency table
→	..c charts	.cu univariate	plot using a single variable
		.cb bivariate	plot using two variables
		.cm multivariate	plot using several variables
→	..h hypothesis test	.hc contingency	Chi2 test
		.hd distribution	Distribution-related like Shapiro-Wilk test
		.hm means	t-tests and ANOVAs
		.hn nonparametric	Wilcoxon, Kruskal-Wallis tests, ...
		.hp proportions	Proportion test
		.hv variances	Bartlett, Levene tests, ...
		.hc correlation	Correlation test
→	..i (d)istributions	.iu uniform	Uniform distribution (calculations and plots)
		.in normal	Normal distribution
		.il log-normal	Log-Normal distribution
		.it t (Student)	Student t distribution
		.ib binomial	Binomial distribution
		.ip poissons	Poissons distribution
		.ic chi2	Chi2 distribution
		.if F	Fischer F distribution
		.ia annotations	Annotate plots for distributions
→	..m models	.ml linear	Fit and examine a linear model
		.mt tools	Model tools (e.g., export as dataframe)
→	..t tools	.tm memory	Inspect R object size