

# 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 <- sselect(ub, 1:5)  
ub2 <- sfilter(ub1, origin == "Farm")
```

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

```
• Pipeline with {svFlow}  
ub %>%  
  sselect(., 1:5) %>%  
  sfilter(., origin == "Farm") %>%  
  ub2
```

`%>%` is an explicit pipe (dot must be specified). `%->%` is the alternate assignation to use at the end of the pipeline.

## 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()**

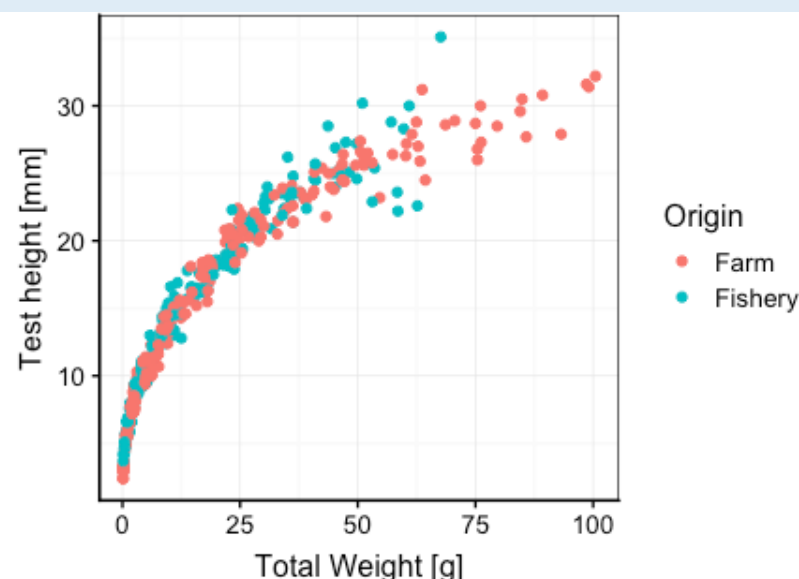
```
chart(data = ub, aes(x = weight, y = height, colour = origin)) +  
  geom_point()
```

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))
```

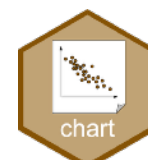
4. Use facets (<https://ggplot2-book.org/facet.html>) in the formula

```
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 directories, e.g.:
  - **data** sub-dir: all datasets (also **data-raw**)
  - **R** sub-dir: all R scripts
  - Main directory: all notebooks, reports, presentations, ...

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