future CHEAT SHEET

The purpose of the future package is to provide a very simple and uniform way of evaluating R expressions asynchronously using various resources available to the user.

Create a future

future(expr, envir, substitute, globals, packages, lazy**)**
Create a future.
Function can be substituted with '%<-%' or '%->%' expressions

expr
an R epxression

envir
The environment from where global objects should be identified

substitute
If TRUE, argument expr is substitute():ed, otherwise not

globals
logical how globals are handled

packages
which packages to be attached in the R environment evaluating the future

lazy
If FALSE (default), the future is resolved eagerly

```
fitB %<-% lm(y ~ x - 1, weights = w)      ## without offset
```

Specify how to resolve a future

plan(strategy, substitute**)**
This function allows the user to plan the future

strategy
The evaluation function to use for resolving a future. Available strategies:
- sequential
- transparent
- multisession
- multicore
- multiprocess
- cluster
- remote

substitute
If TRUE, the strategy expression is substituted

```
plan(sequential)
plan(multisession)
```

Check future status

resolved(x**)**
Check whether a future is resolved or not

x
A future, a list, or an environment

backtrace(future, envir**)**
Back trace the expressions evaluated when an error was caught

future
A future with a caught error

envir
The environment where to locate the future

Adjust future arguments

tweak(strategy**)**
Tweak a future function by adjusting its default arguments

strategy
An existing future function or the name of one