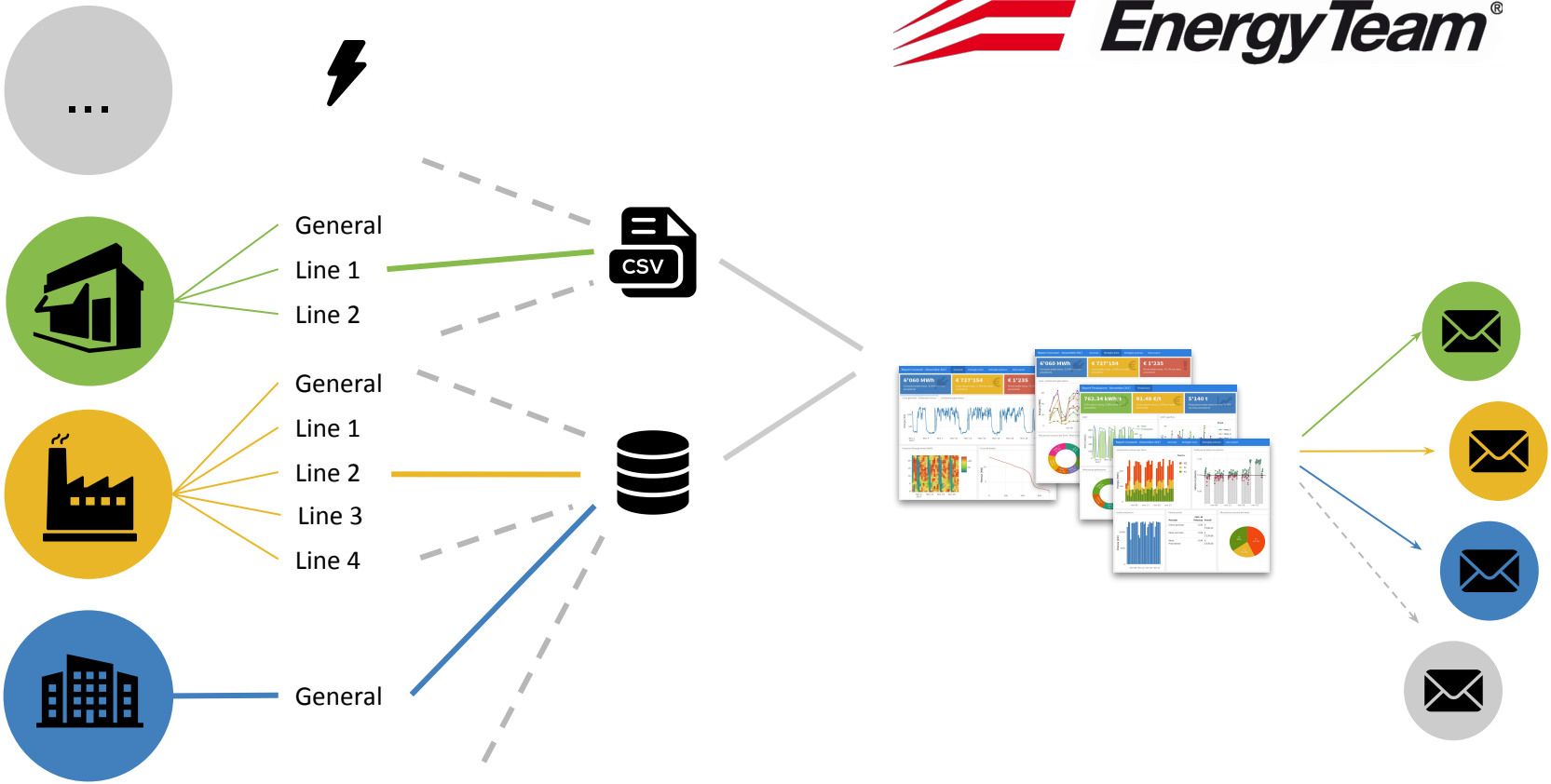
Complete automation

Project & Code design
for a massive reporting system with R

Outline

1. **Challenge**
   a. Energy team
   b. Interactive and Serverless
   c. Fully automated
2. **Method**
   a. Project in a Package
   b. Dashboard design
3. **Data**
   a. Different sources and data checks
   b. Preprocessing data pipeline
   c. Handling data content variability
4. **Interface**
   a. Granular output (and readable code)
   b. Appearance and readability
   c. Data export
5. **Robustness**
   a. Tests
6. **Demo!**
7. **Contact us**

EnergyTeam®


...
General
Line 1
Line 2
General
Line 1
Line 2
Line 3
Line 4
General
CSV

Fully automated

Run by itself

No person required

It can be scheduled

Robust to data variability (i.e.: the number of line changes)

Parametrized

Same report for different data or time period

Connected with database with real time data

Customizable for specific usage

Interactive & Serverless

Serverless:
Single independent HTML file
Shareable by email
Opened by a simple web browser

Interactive:
Zoom In/Out
Tooltip in plots
Export data as xls or png images

Interactive & Serverless

Serverless:

Single independent HTML file

Shareable by email

Opened by a simple web browser

`Flexdashboard`

Interactive:

Zoom In/Out

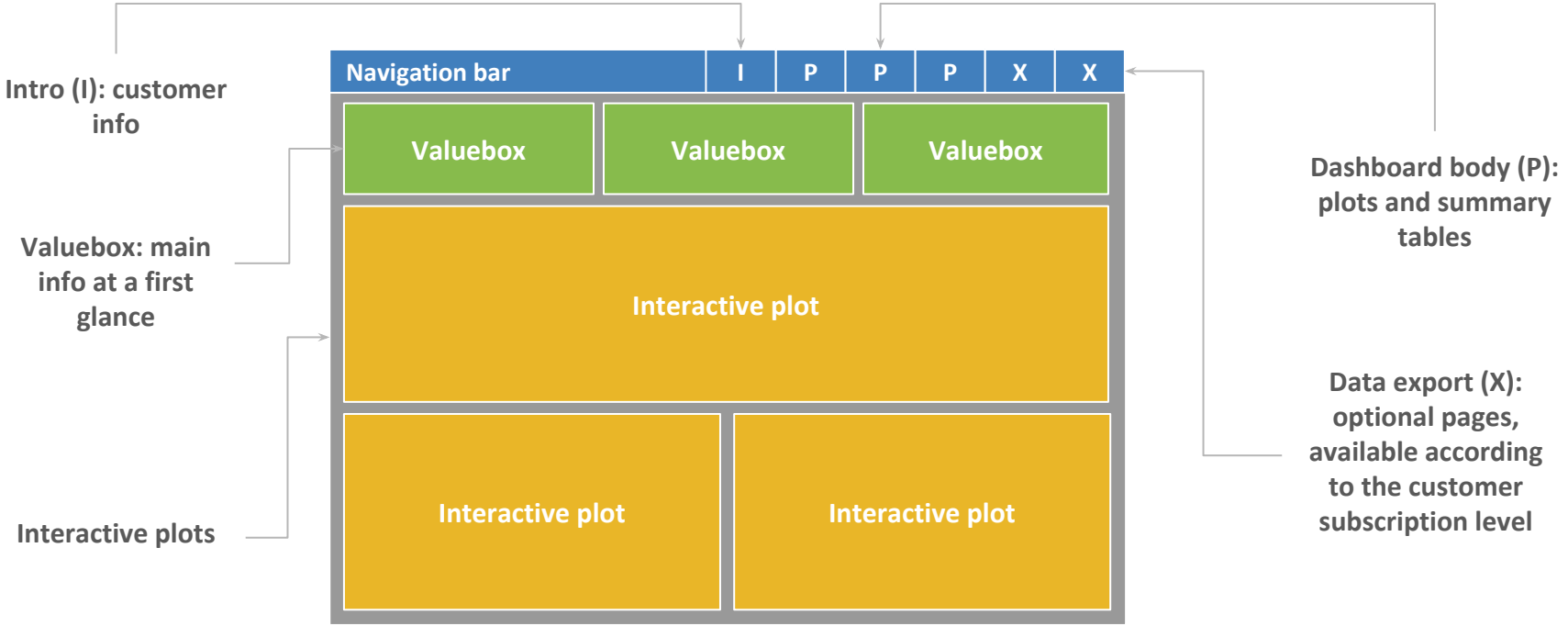
Tooltip in plots

Export data as xls or png images

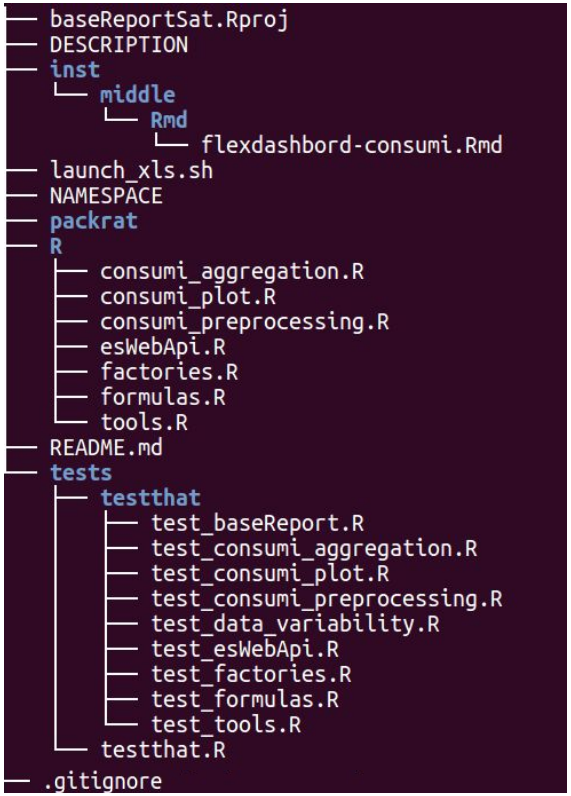
`Plotly`

`DT`

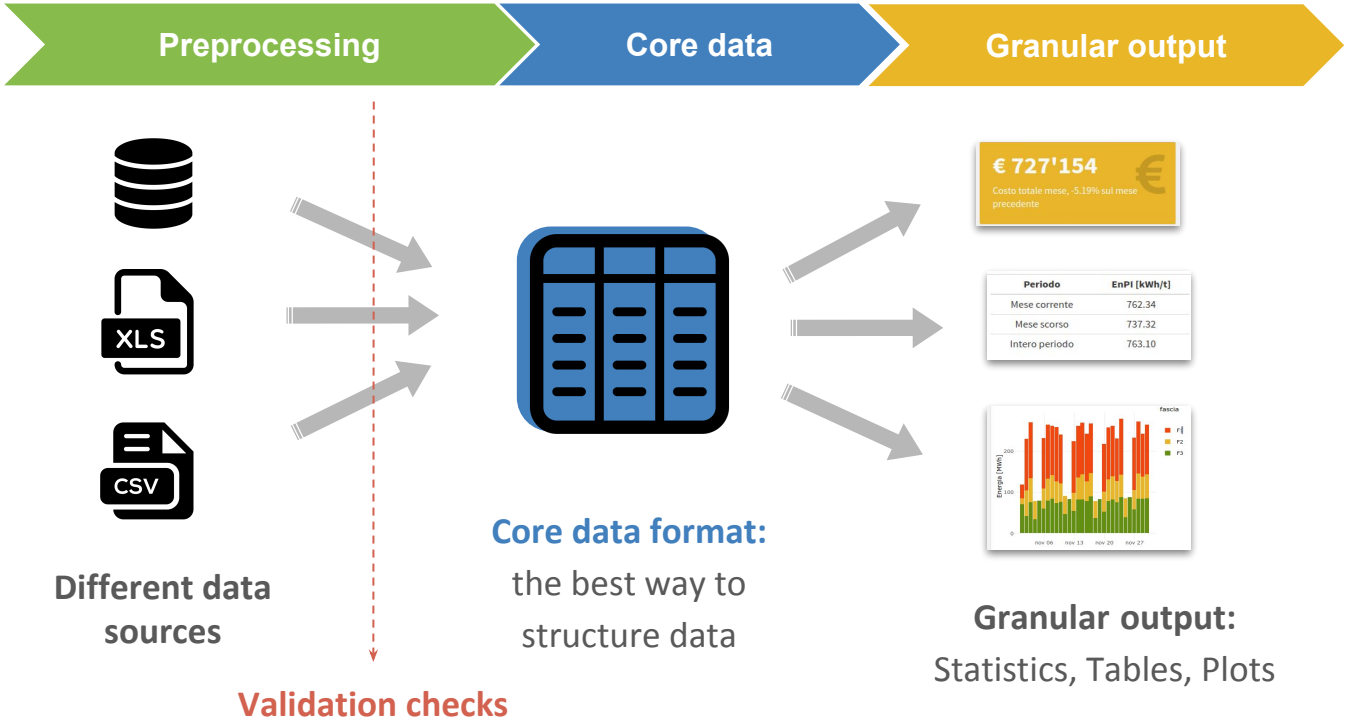
Dashboard design



Project in a Package



Different sources and data checks



Preprocessing: data pipeline

Core data format: the best way to structure data

- All the relevant information
- Sources of variability to factors (ex. lines)
- Time variable at the finest used level

```
consumption_tbl <- consumi_read(file) %>%
    consumption_clean() %>%
    consumption_manipulate() %>%
    consumption_reshape()
```

```
consumption_read <- function(file) {
  stopifnot(file.exists(file))
  switch(tools::file_ext(file),
     "xls"  = consumi_xls_read(file),
     "csv"  = consumi_csv_read(file),
  stop("File extension unknown"))}
```

consumption_reshape()

general	linea1	linea2
...	...	...

linea	power
general	...
linea1	...
linea2	...

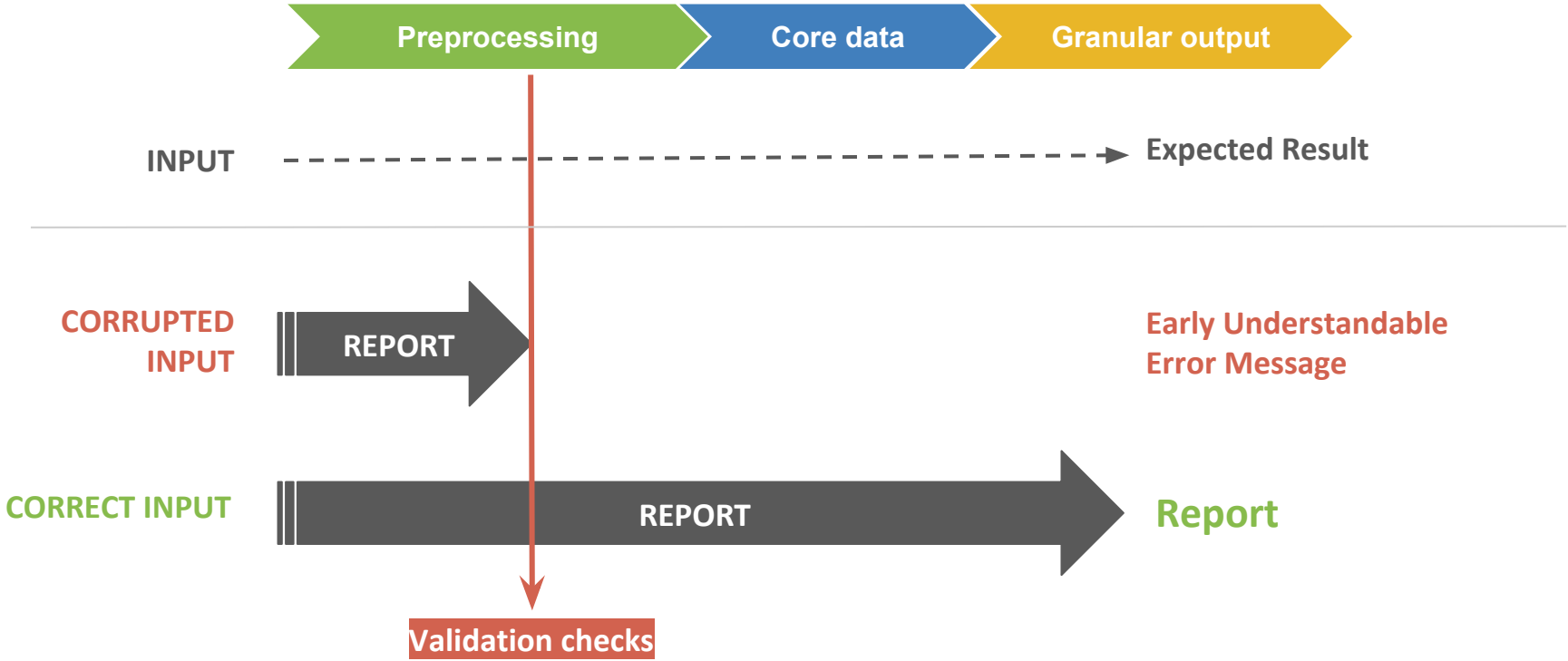
Handling data variability

CORRUPTED INPUT

Handling data variability

CORRUPTED INPUT

Handling data variability



Handling data variability

Example: `sqrt()` requires non negative numbers to work as expected

RANDOMIC DATA

```
num <- runif(1)
```

CORRUPTED INPUT:

FILTER
`num[ num < 0 ]`

sqrt(num)

```
expect_error(
  sqrt(num))
```

CORRECT INPUT:

COUNTER FILTER
`num[ ! num < 0 ]`

sqrt(num)

```
expect_equal(
  sqrt(num^2)==num)
```

for 1 to 100

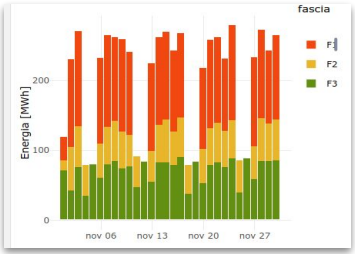
TEST

Granular output (and readable code)

Core data → Granular output

consumi-flexdashboard.Rmd

```
core_data %>%
 hist_daily_setup() %>%
 hist_daily_plot()
```

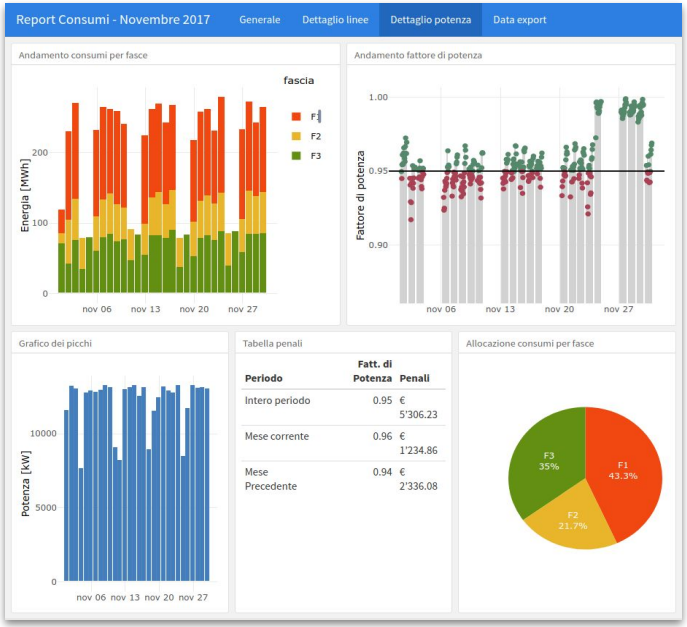


consumi-plot.R

```
hist_daily_setup <- function(df, month, type="F123"){
 df %>%
  filter_month(month) %>%
  ...
  summarize(energy = sum(energy))}

hist_daily_plot <- function(hist_daily_setup){
  gg_hist_day <- ggplot(
     data = hist_daily_setup,
     aes(..., text=text) +
     ...
     scale_fill_manual(values = shift_palette()
  )
  ggplotly(gg_hist_day, tooltip = "text") }
```

Appearance and readability



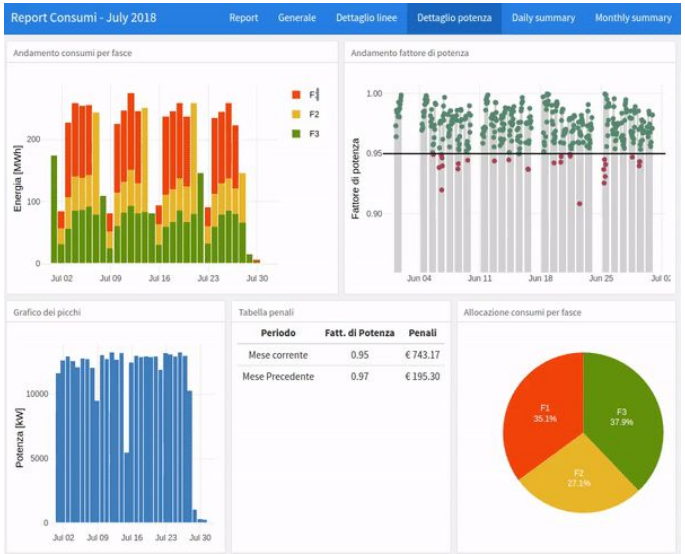
Interactivity

```
gg_hist_day <- ggplot(
      data = hist_daily_setup,
      aes(..., text=text) +
      ...
      scale_fill_manual(values = shift_palette()
  )
```

```
ggplotly(gg_hist_day, tooltip = "text")
```

```
install.package("plotly")
```

Appearance and readability

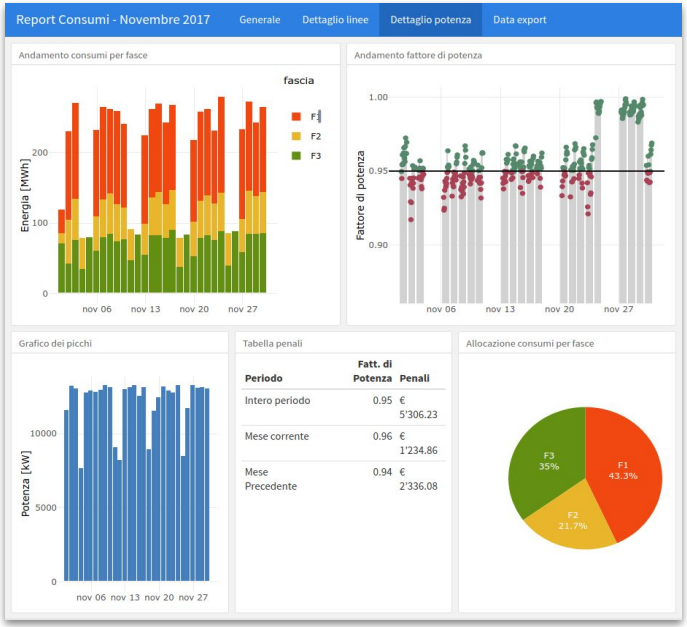


Tooltip

```
gg_hist_day <- ggplot(
     data = hist_daily_setup,
     aes(..., text=text) +
     ...
     scale_fill_manual(values = shift_palette()
  )

  ggplotly(gg_hist_day, tooltip = "text")
```

Appearance and readability



Cross-plot palette

```
gg_hist_day <- ggplot(
     data = hist_daily_setup,
     aes(..., text=text) +
     ...
     scale_fill_manual(values = shift_palette()
  )

  ggplotly(gg_hist_day, tooltip = "text")
```

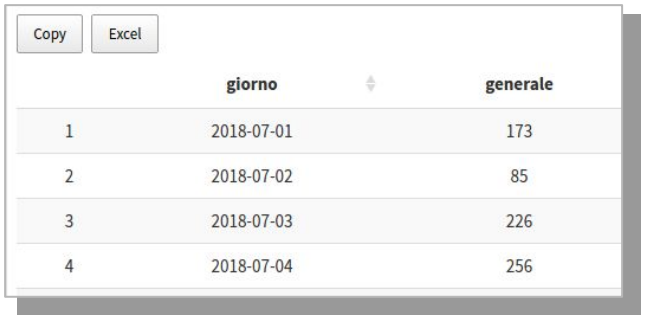
```
shift_palette <- function()
     {c("#F2440C", "#E9B628", "#63910C")}
```

Data export and print

consumi-flexdashboard.Rmd

```
```{r, eval = params$if_export}

core_data %>%
  export_table_setup() %>%
  export_table()
```
```



consumi-plot.R

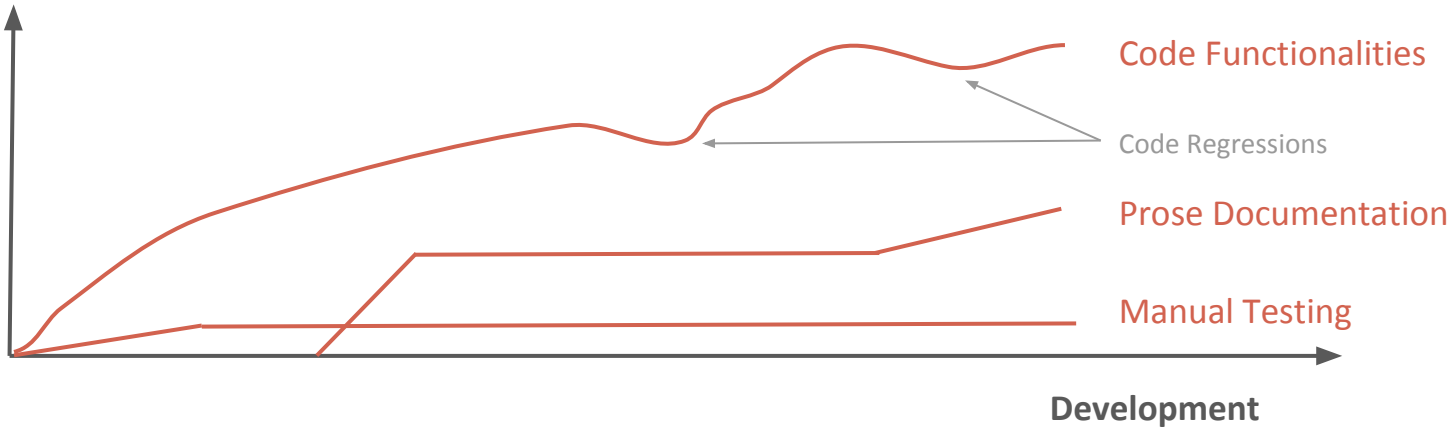
```
export_table <- function(df,
                         buttons = c("copy",
                                     "excel")){
    datatable(df,
      extensions = 'Buttons')}
```

```
install.package("DT")
```

Tests

PRODUCT

- Valid
- Documented
- Improve without regression



Tests

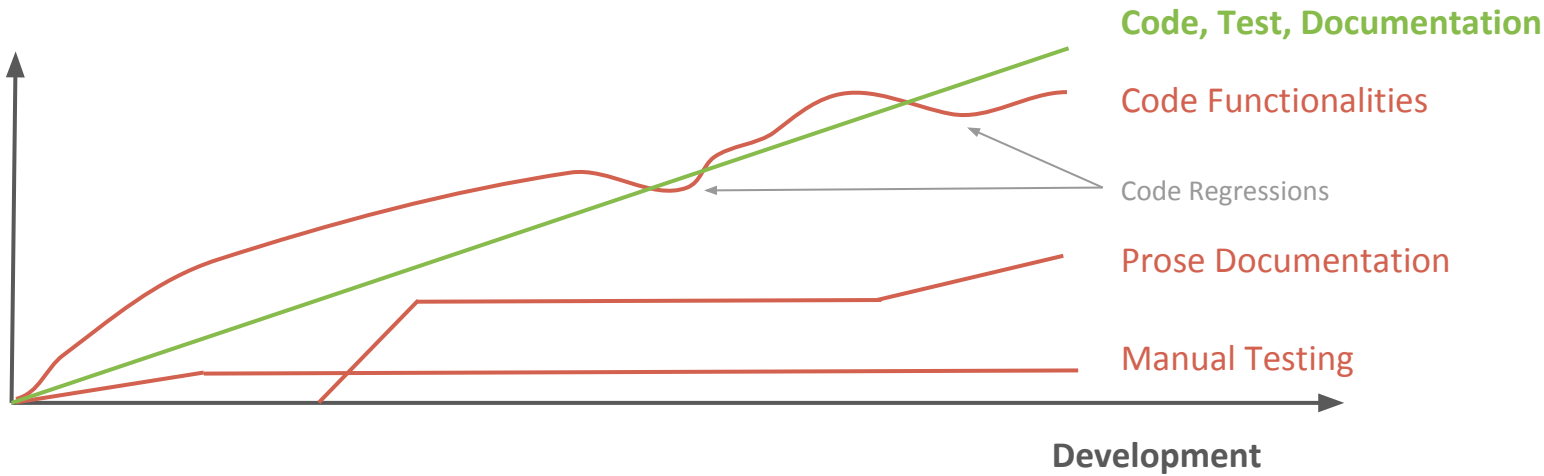
PRODUCT

Valid

Documented

Improve without regression

AUTOMATED TESTS



Contact

QUANTIDE
knowledge from data

www.quantide.com
info@quantide.com

Milano R

www.milanor.net
facebook.com/MilanoRcommunity/

Mariachiara Fortuna
mariachiara.fortuna1@gmail.com

@maryclaryf
Git: mariachiarafortuna

Andrea Melloncelli
andrea.melloncelli@gmail.com

@akirocode
Git: akirocode