Project Template & Kaggle Competition

R-ladies - August 2014

What is ProjectTemplate?

Provides functions to automatically build a directory structure for a new R project.

Automates data loading, preprocessing, library importing and unit testing.

Installation

From CRAN:

```
install.packages('ProjectTemplate')
```

From the Github (to install ProjectTemplate from source):

```
devtools::install_github('johnmyleswhite/ProjectTemplate')
```

Installation

From CRAN:

```
install.packages('ProjectTemplate')
```

From the Github (to install ProjectTemplate from source):

```
devtools::install_github('johnmyleswhite/ProjectTemplate')
```

Create a project

1. Set the work direcory
   a. setwd("*<Path to where you want to create the project>*")
      i. Eg: setwd("/Users/gqueiroz/Dropbox/Rladies/Meetups/Kaggle.Competition/")
2. Load the library
   a. library(ProjectTemplate)
3. Create the project
   a. create.project("Titanic")

Create a project

Source on Save

```
setwd("/Users/gqueiroz/Dropbox/Rladies/Meetups/Kaggle.Competition/")
library(ProjectTemplate)
create.project("Titanic")
```

Files | Plots | Packages | Help | Viewer

New Folder | Delete | Rename | More

Home > Dropbox > Rladies > Meetups > Kaggle.Competition > Titanic

Name	Size	Modified
..		
cache		
config		
data		
diagnostics		
doc		
graphs		
lib		
logs		
munge		
profiling		
README	1.5 KB	Aug 28, 2014, 3:21 PM
reports		
src		
tests		
TODO	0 B	Aug 28, 2014, 3:21 PM

Open terminal

```
Last login: Thu Aug 28 15:22:24 on ttys028
gqueiroz@alpine:~ $ cd Dropbox/Rladies/Meetups/Kaggle.Competition/Titanic/
```

```
Last login: Thu Aug 28 15:22:24 on ttys028
gqueiroz@alpine:~ $ cd Dropbox/Rladies/Meetups/Kaggle.Competition/Titanic/
gqueiroz@alpine:~/Dropbox/Rladies/Meetups/Kaggle.Competition/Titanic $ ls
README      cache       data        doc         lib         munge       reports     tests
TODO        config      diagnostics graphs      logs        profiling   src
gqueiroz@alpine:~/Dropbox/Rladies/Meetups/Kaggle.Competition/Titanic $
```

Directories and Files

Each of these serves a specific purpose:

- **cache:** Here you'll store any data sets that
  - (a) are generated during a preprocessing step and
  - (b) don't need to be regenerated every single time you analyze your data.

You can use the cache() function to store data to this directory automatically. Any data set found in both the cache and data directories will be drawn from cache instead of data based on ProjectTemplate's priority rules.

Directories and Files

- **config:** Here you'll store any configurations settings for your project. Use the DCF format that the read.dcf() function parses.

- **data:** Here you'll store your raw data files. If they are encoded in a supported file format, they'll automatically be loaded when you call **load.project()**.

Directories and Files

- **diagnostics:** Here you can store any scripts you use to diagnose your data sets for corruption or problematic data points.

- **doc:** Here you can store any documentation that you’ ve written about your analysis.

Directories and Files

- **graphs:** Here you can store any graphs that you produce.
- **lib:** (not talk for now)
- **logs:** (not talk for now)

Directories and Files

- **munge:** Here you can store any preprocessing or data munging code for your project. The preprocessing scripts stored in munge will be executed sequentially when you call **load.project()**, so you should append numbers to the filenames to indicate their sequential order.

Directories and Files

- **profiling:** (not talk for now)

- **reports:** Here you can store any output reports, such as HTML or LaTeX versions of tables, that you produce. Sweave or brew documents should also go in the reports directory.

Directories and Files

- **src:** Here you'll store your final statistical analysis scripts.
  - You should add the following piece of code to the start of each analysis script:
    - library('ProjectTemplate)
    - load.project()

Directories and Files

- **tests:** Here you can store any test cases for the functions you've written. Your test files should use testthat style tests so that you can call the test. project() function to automatically execute all of your test code.

Directories and Files

- **README:** In this file, you should write some notes to help orient any newcomers to your project.

- **TODO:** In this file, you should write a list of future improvements and bug fixes that you plan to make to your analyses.

Download the Titanic dataset

www.kaggle.com/c/titanic-gettingStarted/data

(You need to register)

Knowledge • 2,456 teams

Titanic: Machine Learning from Disaster

Fri 28 Sep 2012 — Wed 31 Dec 2014 (4 months to go)

Competition Details » Get the Data » Make a submission

Data Files

File Name	Available Formats
train	.csv (59.76 kb)
gendermodel	.csv (3.18 kb)
genderclassmodel	.csv (3.18 kb)
test	.csv (27.96 kb)
gendermodel	.py (3.58 kb)
genderclassmodel	.py (5.63 kb)
myfirstforest	.py (3.99 kb)

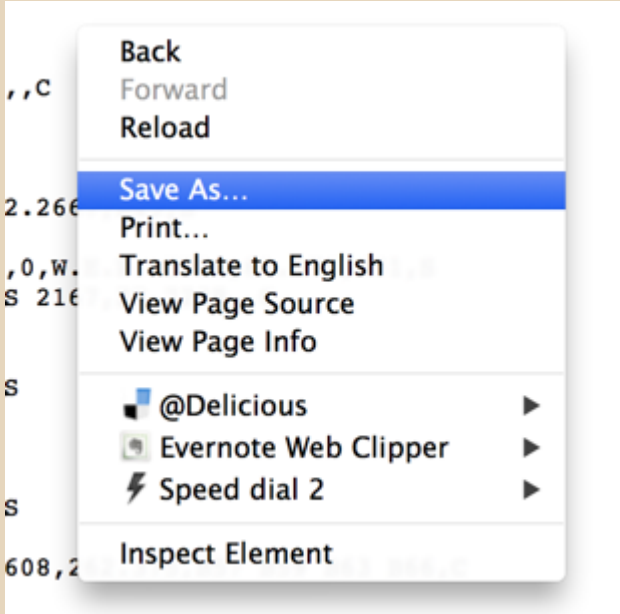
Download the Titanic dataset

Train Dataset

http://bit.ly/titanic_train

Test Dataset

http://bit.ly/titanic_test



Open up RStudio and type:

- setwd("*<Path to Titanic directory>*")
- library('ProjectTemplate')
- load.project()

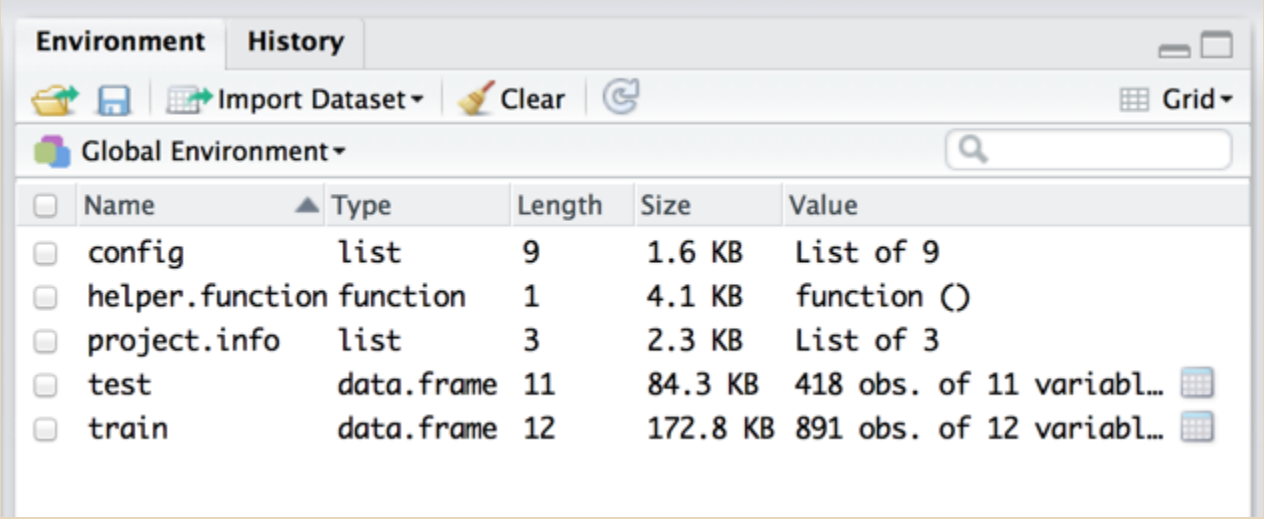
```
> setwd("/Users/gqueiroz/Dropbox/Rladies/Meetups/Kaggle.Competition/Titanic/")
> library(ProjectTemplate)
> load.project()
Loading project configuration
Autoloading helper functions
 Running helper script: helpers.R
Autoloading cache
Autoloading data
 Loading data set: test
 Loading data set: train
Munging data
 Running preprocessing script: 01-A.R
```

notice that the data sets were automatically loaded into memory

RStudio

- ls()

```
> ls()
[1] "config"          "helper.function" "project.info"    "test"            "train"
```



RStudio

- head(train)

```
> head(train)
  PassengerId Survived Pclass                                                Name    Sex Age SibSp Parch           Ticket    Fare Cabin Embarked
1           1        0      3                                   Braund, Mr. Owen Harris   male  22     1     0        A/5 21171  7.2500              S
2           2        1      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0         PC 17599 71.2833   C85        C
3           3        1      3                                    Heikkinen, Miss. Laina female  26     0     0 STON/O2. 3101282  7.9250              S
4           4        1      1            Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0           113803 53.1000  C123        S
5           5        0      3                                  Allen, Mr. William Henry   male  35     0     0           373450  8.0500              S
6           6        0      3                                          Moran, Mr. James   male  NA     0     0           330877  8.4583              Q
>
```

Let's get started!

Goal: predict whether a passenger survived the Titanic crash. You are given two datasets (Train & Test) each of which include predictor variables such as Age, Passenger Class, Sex, etc.

STEPS

1. Create a model which will predict whether a passenger survived using only the **Train** data set.
2. Predict whether the passengers survived in the **Test** data set based on the model we created.

RESULT

- Spreadsheet with **predictions** for which passengers in the **Test** data set survived.

- It will have only **2 columns**:
  - the Passenger ID
  - indicates whether they survived (0 for death, 1 for survival).

Data Exploration

Before actually building a model, we need to explore the data: