Package ‘workplanr’

February 5, 2019

Type Package

Title What the Package Does (Title Case)

Version 0.0.9

Author David Hammond

Maintainer David Hammond <anotherdavidhammond@gmail.com>

Description A package to allow project managers to assign resources to multiple projects, highlight bottlenecks and balance staff loads.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.4.0),
dplyr (>= 0.7.8)

Imports tidyr (>= 0.8.2),
lubridate (>= 1.7.4),
bizdays (>= 1.0.6),
padr (>= 0.4.1),
utils (>= 3.4.3),
RColorBrewer (>= 1.1.2),
ggplot2 (>= 3.0.0),
ggrepel (>= 0.0.8),
scales (>= 1.0.0),
grDevices (>= 3.4.3),
rio (>= 0.5.10),
methods (>= 3.4.3),
readxl (>= 1.1.0)

LazyData true

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`workplanr-package` *IEP charting functions for R*

Description

`workplanr` package

Details

The `iep.charts` package is a collection of functions I have written to help me with ggplot in R.

These functions are intended to produce pdfs that are importable into illustrator making the layout for comms much easier.

Installation

Note: If you have not set yourself up on Github yet you need to do this first, setup guide here `https://githubiep.github.io/Github.html#setup`. This needs to be done every time you get a new computer.

# Then you need devtools installed

```
install.packages('devtools')
```

# Now you can install from github

```
devtools::install_github('githubIEP/iep.charts',
auth_token=suppressWarnings(readLines(file.path(Sys.getenv('HOME'),
'authtoken.txt'))), type='source', INSTALL_opts=c('--no-lock'))
```

# Now you can load the package

```
library(iep.charts)
```

Limitations

Currently there is no way to add axis labels such as "Less Peaceful" "More Peaceful". I do this either in word of in illustrator. Its annoying but it is still a net benefit gain for the process from research to layout.

Use Cases

There are three use cases for `iep.charts`

- Producing charts for draft reports
- Producing finalised charts for layout
- Producing maps in IEP colours

iep.charts requires you to produce charts using `ggplot2`. Go here for any ggplot you will ever need:

`http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html`.

Draft chart reports using the theme_iep function

The process is

- load `library(ggplot2)`
- load `library(iep.charts)`
- Produce a ggplot chart in R (say its called `p` for example).
- Finish the plot by `p <- p + theme_iep()`
- run `iep.ggsave(p)` to save the chart into a png and pdf in the `graphs` directory.

Charts for Layout using the iep.chart.theme function

The process is

- load `library(ggplot2)`
- load `library(iep.charts)`
- Produce a ggplot chart in R (say its called `p` for example). **Note:** You need to finish your plot before you use iep.charts functions.
- This means it has a title, narrative title, source, x and y labels as per the example in `iep.chart.theme`
- run `p <- iep.chart.theme(p)` on your plot
- run `grid.draw(p)` to view it
- run `iep.ggsave(p)` to save the chart into a png and pdf in the `graphs` directory.

Mapping using the iep.map.data function

iep.map.data plots the maps for GPI, GTI and PPI. I STILL NEED TO DOCUMENT THESE - Watch this space.

Author(s)

David Hammond <anotherdavidhammond@gmail.com>

`build_sample_workplan` *Create a random workplan*

Description

This function creates an excel file that can be used to create a new project

Usage

```
build_sample_workplan(excel_file_name = "my-workplan.xlsx")
```

Arguments

`excel_file_name`
File name for project inputs

Examples

```
library(workplanr)
```

export_workplan *Create Excel file for project inputs*

Description

This function creates an excel file that can be used to create a new project

Usage

```
export_workplan(wp, excel_file_name = "my_workplan.xlsx")
```

Arguments

`wp` A workplan object

`excel_file_name`
File name for project inputs

Examples

```
library(workplanr)
wp <- build_sample_workplan()
## ----export_workplan, include = TRUE, results='hide', message=FALSE, warning=FALSE----
export_workplan(wp, excel_file_name = "my-workplan.xlsx")
## ----import_workplan, include = TRUE, results='hide', message=FALSE, warning=FALSE----
wp <- import_workplan(excel_file_name = "my-workplan.xlsx")
```

get_workplan *create a list of employees that are to be assigned to projects*

Description

create a list of employees that are to be assigned to projects

Usage

```
get_workplan(staff, staff_capacity, projects, project_probability,
  project_start, project_end, project_phases, project_time_estimates,
  staff_on_leave, leave_start, leave_end, leave_description,
  public_holidays_date, public_holidays_name,
  staff_project_assignment_capacity)
```

Arguments

`staff` Names of staff members

`staff_capacity`
Number of units of work per staff, for example 100 for full time equivalents, 40 for staff who work only 2 days per week

`projects` Names of projects

`project_probability`
Probability that the project will go ahead

`project_start`
Expected start date of project

`project_end` Expected end date of the project

`project_phases`
List of phases in any project in order of execution

`project_time_estimates`
Time estimates of how long each phase will take in relation

`staff_on_leave`
Names of staff that are going to be out of the office

`leave_start` Starting date for leave

`leave_end` Ending date for leave

`leave_description`
Type of leave, can be user defined but recommend "leave" or "work trip"

`public_holidays_date`
A data frame of dates of public holidays

`public_holidays_name`
A data frame of names of public holidays

`staff_project_assignment_capacity`
Amount of time each staff is expected to dedicate to each [project, phase]

Value

A reference table for staff

Examples

```
## ----library, include = TRUE, results='hide', message=FALSE, warning=FALSE----
library(workplanr)
## ----resources, include = TRUE, results='hide', message=FALSE, warning=FALSE----
staff <- c("Shelby", "Luis", "Taishawn", "Samantha", "Taylor", "unassigned")
staff_capacity <- c(40,60,100,100,100, 100)
## ----projects, include = TRUE, results='hide', message=FALSE, warning=FALSE----
projects <- LETTERS[1:3]
project_probability <- c(50, 100, 100)
project_start <- as.Date(c("2019-01-25", "2019-05-17", "2019-06-27"))
project_end <- as.Date(c("2019-06-03", "2019-06-16", "2019-09-27"))
project_phases <- c("research", "drafting", "editing", "design", "print", "events")
project_time_estimates  <- c(c(-10,-10,-10,-10,-10,-10), c(-10,-10,-10,-10,-10,-10), c(-1
## ----leave, include = TRUE, results='hide', message=FALSE, warning=FALSE----
staff_on_leave <- c("Luis", "Samantha")
leave_start <-  as.Date(c("2019-07-23", "2019-05-16"))
leave_end <- leave_start + c(20, 25)
leave_description <- c("leave", "work trip")
## ----holidays, include = TRUE, results='hide', message=FALSE, warning=FALSE----
url <- "https://data.gov.au/data/dataset/b1bc6077-dadd-4f61-9f8c-002ab2cdff10/resource/31
public_holidays <- utils::read.csv(url, stringsAsFactors = FALSE)
names(public_holidays) <- tolower(names(public_holidays))
public_holidays$date <- as.Date(lubridate::ymd(public_holidays$date))
public_holidays <- public_holidays %>% filter(jurisdiction == "nsw") %>%
  select(date, holiday.name) %>% rename(name = holiday.name)
public_holidays_date <- public_holidays$date
public_holidays_name = public_holidays$name
## ----assignments, include = TRUE, results='hide', message=FALSE, warning=FALSE----
```

```
staff_project_assignment_capacity <- sample(c(0,25,50,75,100), size = length(projects)*le
## ----get_workplan, include = TRUE, results='hide', message=FALSE, warning=FALSE----
wp <- get_workplan(staff = staff,
                   staff_capacity = staff_capacity,
                   projects = projects,
                   project_probability =  project_probability,
                   project_start = project_start,
                   project_end = project_end,
                   project_phases = project_phases,
                   project_time_estimates = project_time_estimates,
                   staff_on_leave = staff_on_leave,
                   leave_start = leave_start,
                   leave_end = leave_end,
                   leave_description = leave_description,
                   public_holidays_date = public_holidays_date,
                   public_holidays_name = public_holidays_name,
                   staff_project_assignment_capacity = staff_project_assignment_capacity)
print(wp)
```

import_workplan *Create Excel file for project inputs*

Description

This function creates an excel file that can be used to create a new project

Usage

```
import_workplan(excel_file_name = "my_workplan.xlsx")
```

Arguments

`excel_file_name`
File name for project inputs

Examples

```
library(workplanr)
wp <- build_sample_workplan()
## ----export_workplan, include = TRUE, results='hide', message=FALSE, warning=FALSE----
export_workplan(wp, excel_file_name = "my-workplan.xlsx")
## ----import_workplan, include = TRUE, results='hide', message=FALSE, warning=FALSE----
wp <- import_workplan(excel_file_name = "my-workplan.xlsx")
```

plot,staff_schedule-method

Coerce Object staff_schedule to a ggplot

Description

Coerce Object full_schedule to ggplot, avoiding using the "slot" notation.

Usage

```
## S4 method for signature 'staff_schedule'
plot(x)
```

Arguments

x	A `staff_schedule` object.

Examples

```
library(workplanr)
wp <- build_sample_workplan()
plot(wp@staff_schedule)
```

plot,team_schedule-method

Coerce Object team_schedule to a ggplot

Description

Coerce Object team_schedule to ggplot, avoiding using the "slot" notation.

Usage

```
## S4 method for signature 'team_schedule'
plot(x)
```

Arguments

x	A `team_schedule` object.

Examples

```
library(workplanr)
wp <- build_sample_workplan()
plot(wp@staff_schedule)
```

Index