



MiraiLabs

From prototype to production

Francesca Vitalini
Riccardo Porreca
Gabriel Foix

An initiative by
Mirai Solutions GmbH
labs@mirai-solutions.com
<https://mirai-solutions.ch>

Bringing ideas to life.

Smart.
Agile.
Personal.



MiraiLabs

MiraiLabs: What is it about?

- Are you a data scientist keen on trying new things?
- Do you use data analysis in your daily work? Do you want to expand your toolkit?
- Are you looking for a more guided hands-on introduction?

Data science workshops for professionals

Data science

Data analytics

New tools & techniques

Interactivity & visualization

For professionals

Based on real experience from the industry

Addressing relevant topics

Goals

Learn together

Establish a community

From Prototype to Production

Practical Infos:

References for this workshop:



- Slides and notes → <https://github.com/miraisolutions/MiraiLabs/tree/master/from-prototype-to-production>
- Final product → <https://github.com/miraisolutions/vbzdelays>
- Tmp workshop demo repository → <https://github.com/riccardoporreca/vbzdelays-demo>

Schedule:

4:15 pm - doors open

4:30 pm - welcome / workshop

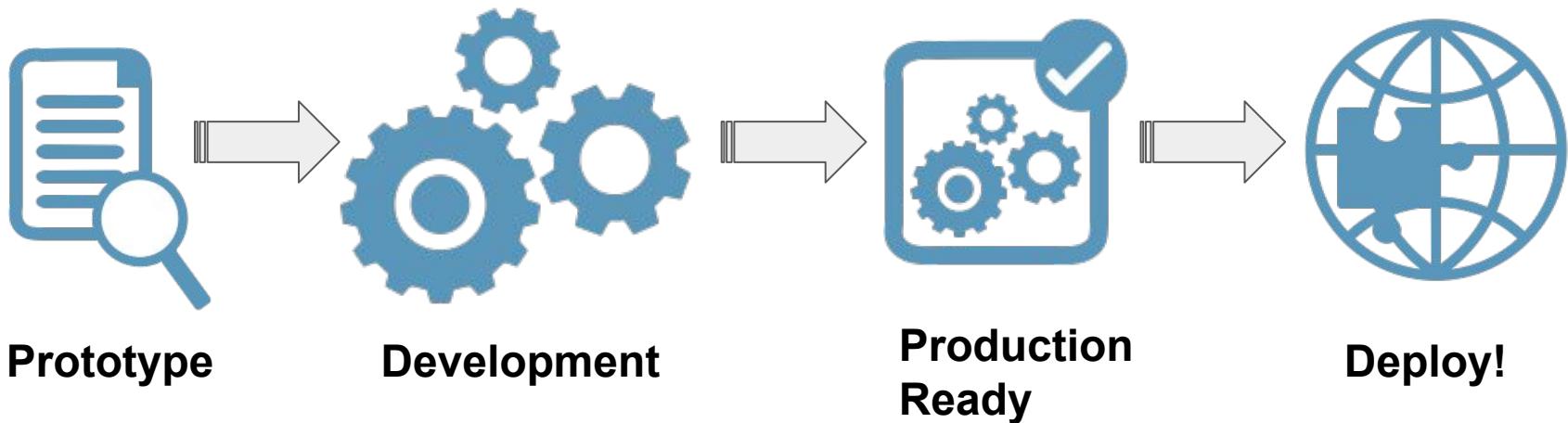
6:00 pm - short break

7:30 pm - end of workshop / closing remarks

7:40 - 9.00 pm - Apéro at the bar (Community Salon)

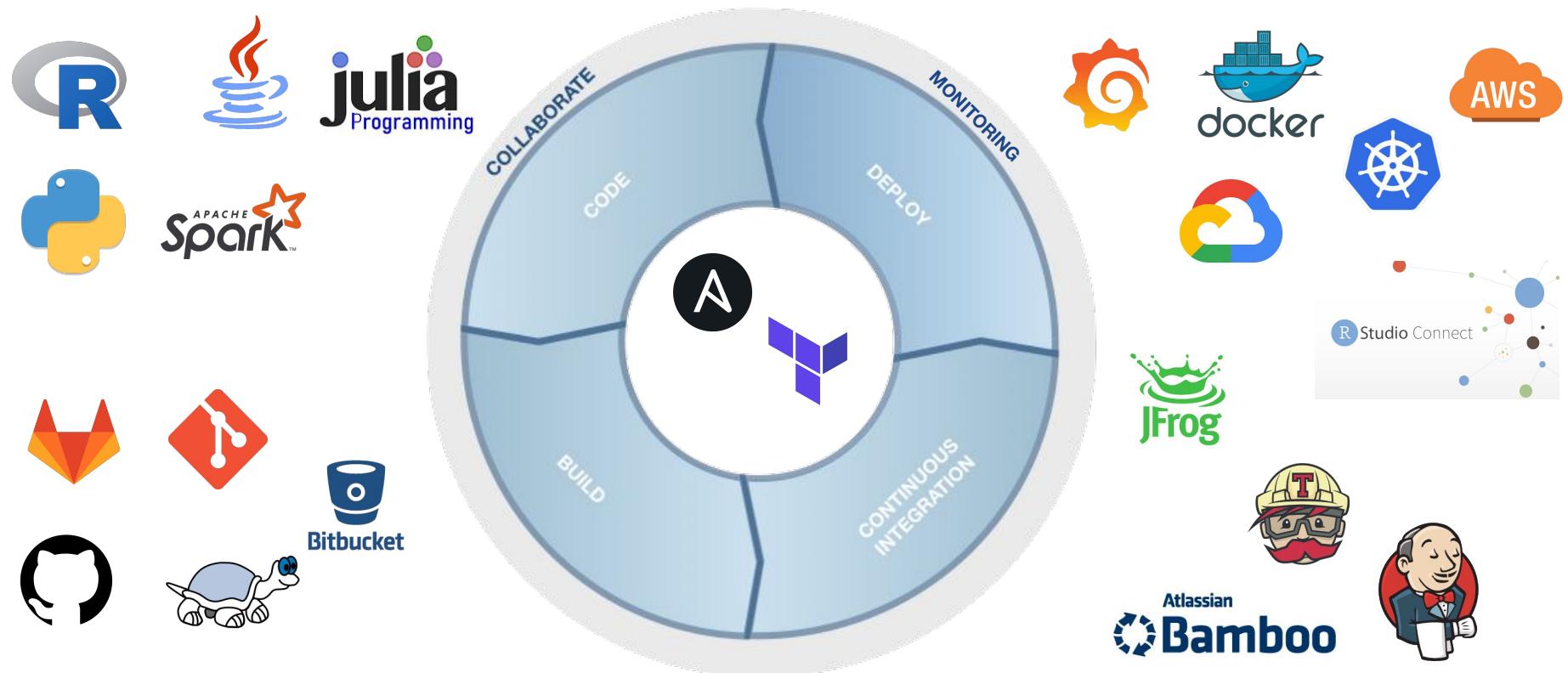
Introduction

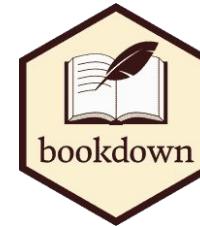
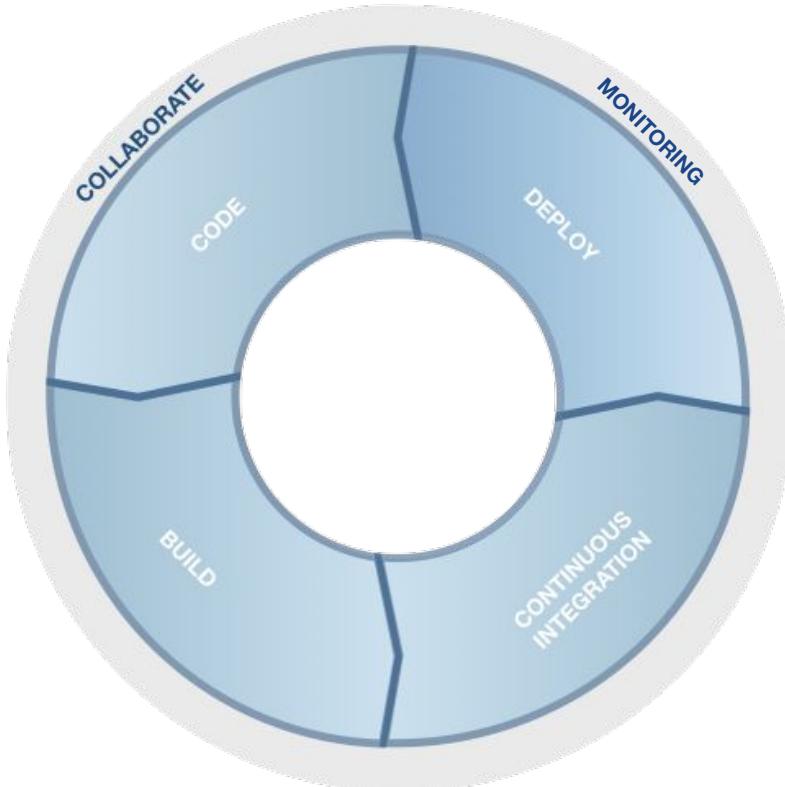
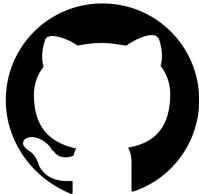
Workflow to Production



What does Production mean?

- The product-building process is an automated, streamlined pipeline, optimized for speed of delivery to the end-user.
- By automating building and testing processes, the pipeline to production ensures:
 - Security
 - Quality
 - Stability
- Production is not a fixed object, but a continuous improvement cycle.





The Input Data

Dataset: ZRH transport network travel times



For each date and line in the Zurich public network system, the expected and actual arrival and departure times for each track between two stations.

Data sets are provided on a weekly basis. In this workshop we will use the following weeks:

- July 14th - 20th
- July 21st - July 27th
- July 28th - August 3rd
- August 4th - August 10th

Data Description

The dataset contains the following information:

- `linie` - the Zurich public transport line number, including buses and trams.
- `betriebsdatum` **operation date**. `datum_von` and `datum_nach` indicate the beginning and at the end of the track respectively, as night tracks might end on the following day.
- `richtung` **is the direction**.
- `halt_id_von`, `halt_id_nach`, `halt_punkt_id_von`, `halt_punkt_id_nach` **contain information about the departure (von) and arrival (nach) stations**.
- `soll_an_von`, `soll_ab_von`, `soll_an_nach`, `soll_ab_nach` `ist_an_von`, `ist_ab_von`, `ist_an_nach1`, `ist_ab_nach` **are the expected (soll) and actual (ist) departure (ab) and arrival (an) times at the departure (von) and arrival (nach) stations - in seconds**.

Data Cleaning

To make the dataset more manageable, in this workshop we will:

- Focus on tram lines only (`line < 30`)
- Choose as `date` only the operation date (`betriebsdatum`), converted into day of the week (`weekday`), starting from Monday.
- Mutate the scheduled departure into the hour of the day, integer starting from 0. Operations between 1 and 4 are actually occurring on the day after and are set to after hour 24.
- Keep data for `actual_departure` (`ist_ab_von`), `scheduled_departure` (`soll_ab_von`), `actual_arrival` (`ist_an_nach1`) and `scheduled_arrival` (`soll_an_nach`).

You can find the script to download and clean the data in [section "Data handling" of the prototype](#).

Exploratory Analysis

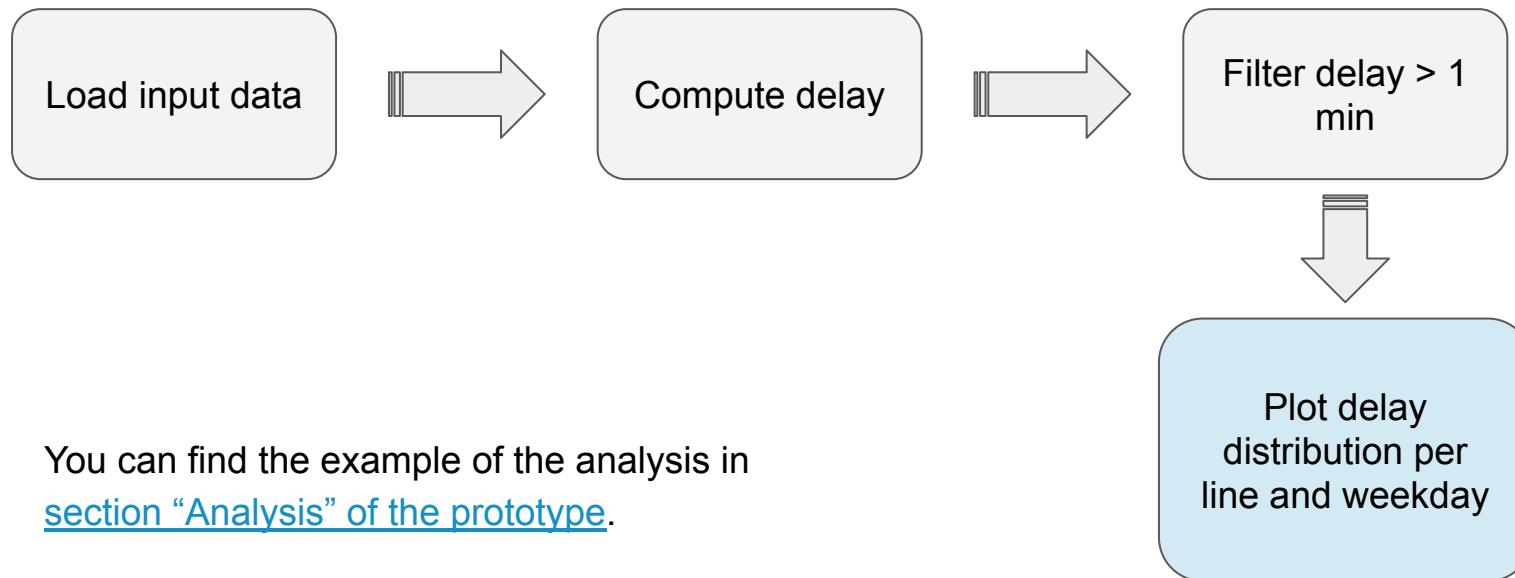
We calculate the delay for each track:

- We define track `delay` as the mean difference in expecting and actual departing from the original station and expecting and actual arriving at the final station (in minutes).

```
((actual_departure - scheduled_departure) +  
 (actual_arrival - scheduled_arrival)) / 2 / 60
```

- We want to report the distribution of delays by line and weekday.

Exploratory Analysis



You can find the example of the analysis in
[section “Analysis” of the prototype](#).

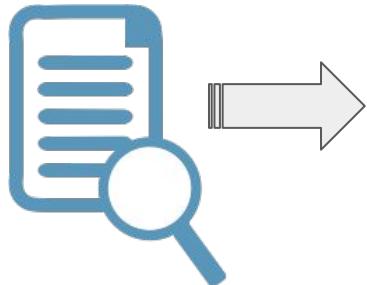
Rmarkdown



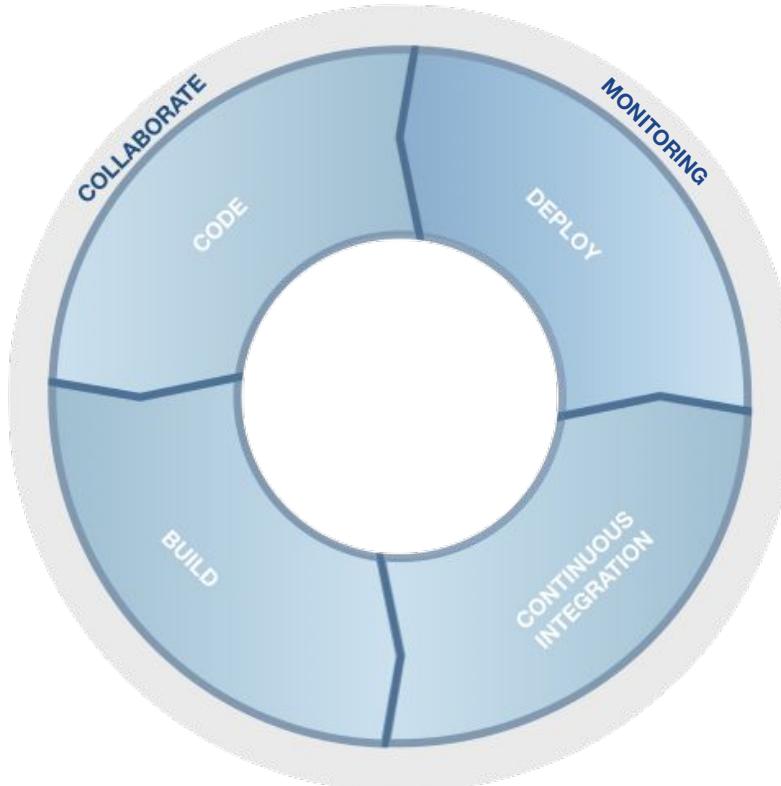
- Notebook interface, similar to Jupiter Notebooks, for high quality documents including executable code and text.
- Rmarkdown documents can be rendered in a variety of nicely formatted outputs, including html documents, pdf etc.
- They support multiple programming languages, in addition to R, e.g. SQL or python.

Reference:

<https://rmarkdown.rstudio.com/>



Prototype



Development

Best Practices

- Modularize your code
- Make your building blocks general and flexible
- Make your code robust with automated testing
- Trace your work with a version control system



Functions and
package structure





Modularisation: Package

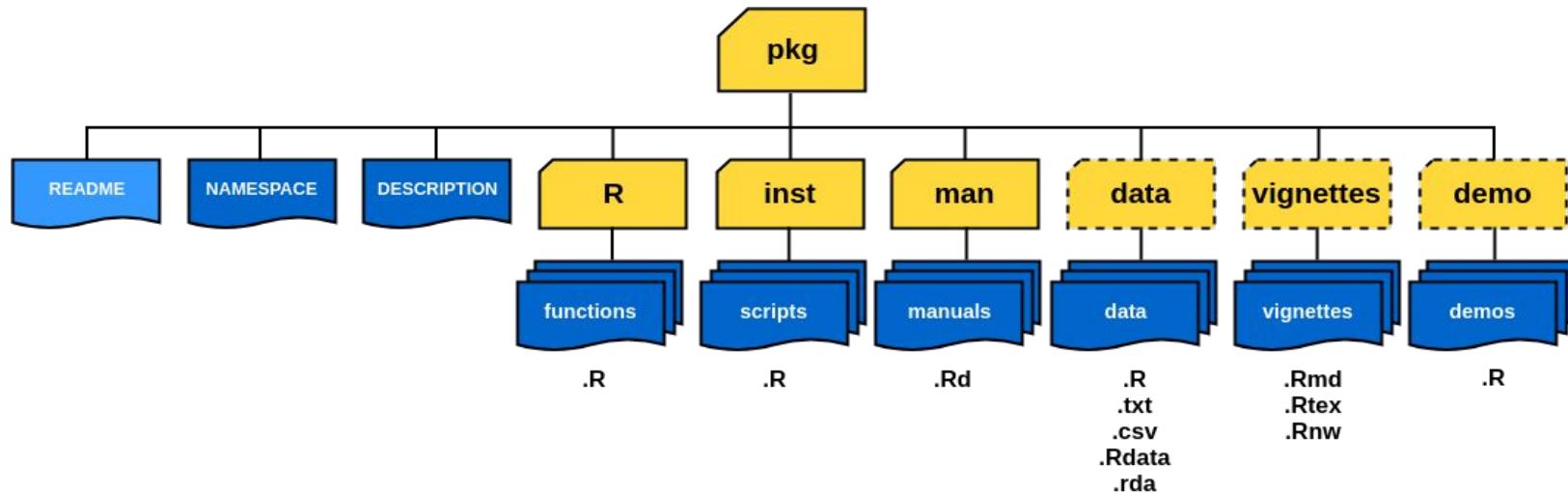
“In R, the fundamental unit of shareable code is the package. A package bundles together code, data, documentation, and tests, and is easy to share with others.”

[Introduction to R packages - Hadley Wickham](#)

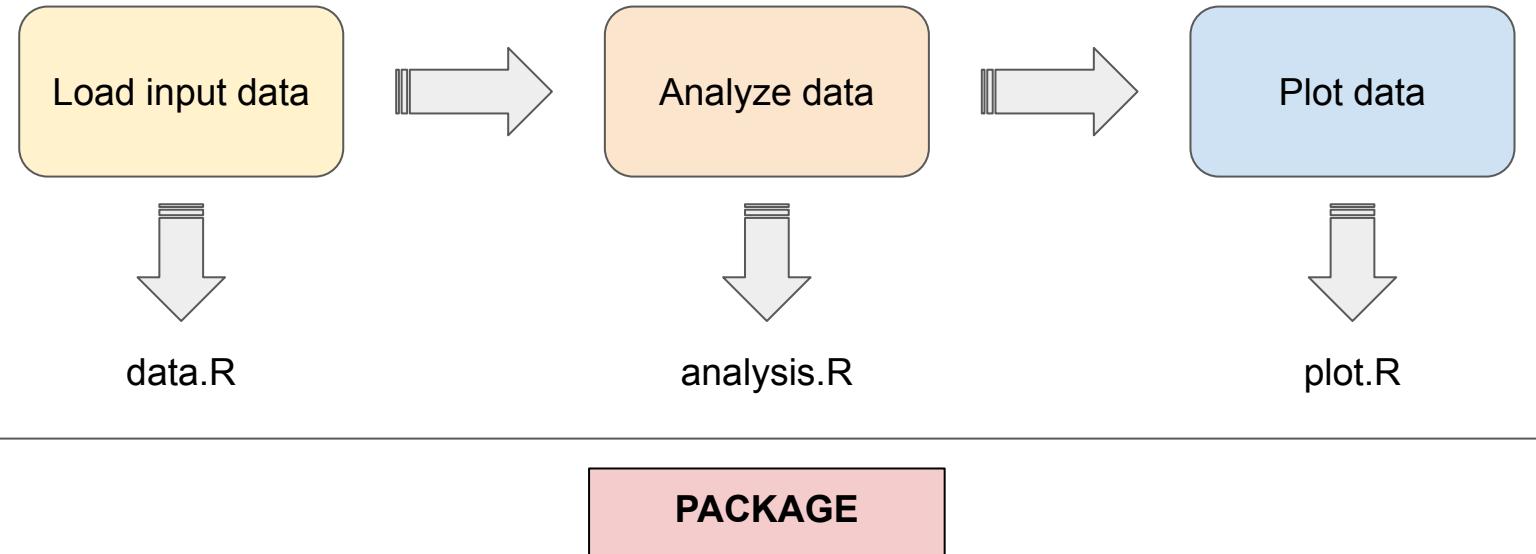
- Packages repository: CRAN - **Comprehensive R Archive Network**
 - Install package `x` from CRAN with `install.packages("pkg")` or from GitHub with `remotes::install_github("username/repo")`
 - Use it in R with `library(pkg)` or `pkg::fun()`
 - Access documentation with `help(package = "pkg")`

Modularisation: Package

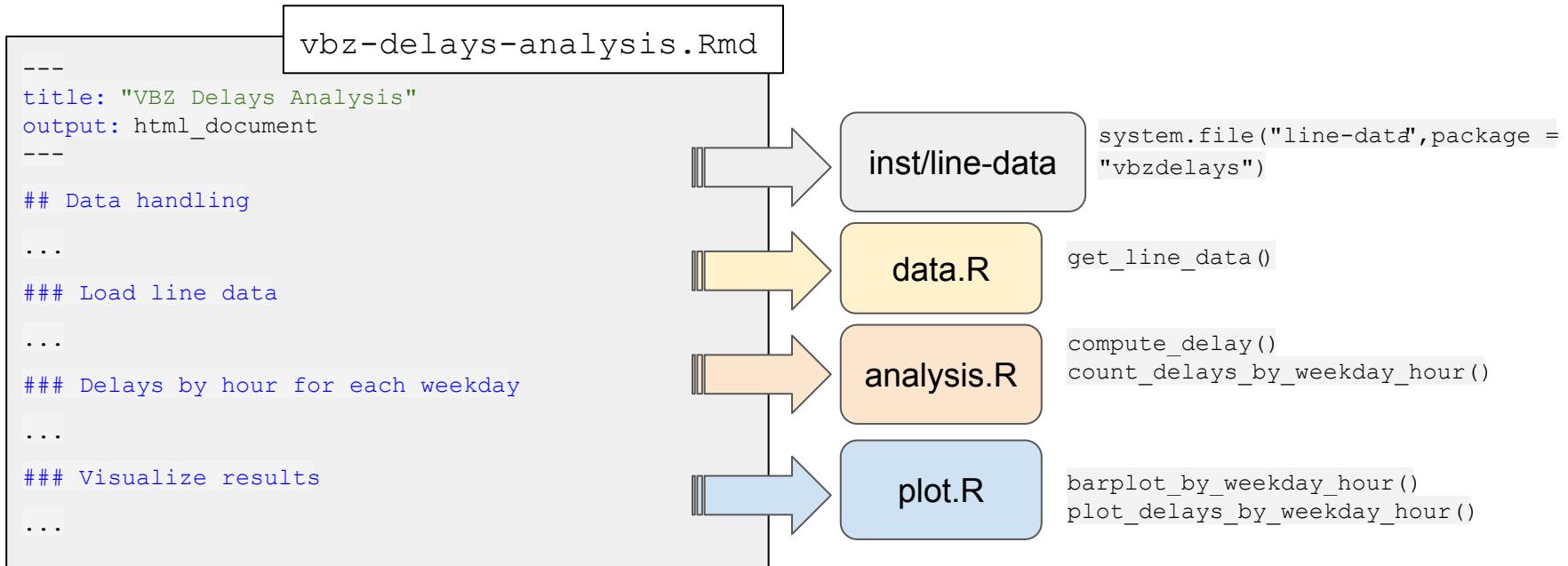
- `devtools::create ("pkg")`



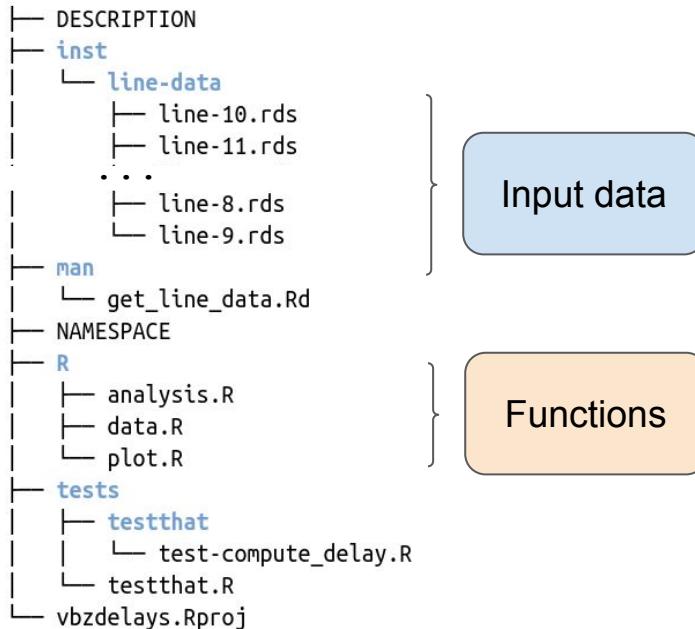
Modularisation



Modularisation: Package



Modularisation: vbzdelays Package



- `DESCRIPTION` shows the package dependencies and metadata
- `NAMESPACE` includes the functions exported and imported by the package
- `man` collects the documentation
- `tests` provides the unit tests
- `inst` files installed with the package and accessible with `system.file(package = "vbzdelays")`
- The package should also feature a `README.md` with general info for the user

Modularisation: Functions

A **function** is a set of statements to perform a specific action on the base of some input parameters (arguments).

```
function name <- function(arg_1, arg_2, ...) {  
  Function body  
}
```

Functions can be side-effecting: e.g., may write a file, print messages or create a plot.

Modularisation: Functions

Function to read in the data:

```
get_line_data <- function(line) {  
  line_file <- system.file("line-data", sprintf("line-%d.rds", line),  
                           package = "vbzdelays")  
  readRDS(line_file)  
}
```



Modularisation: Documentation

In R documentation is done via roxygen tags.

Package `roxygen2` uses tags to create help files and maintain the package NAMESPACE.

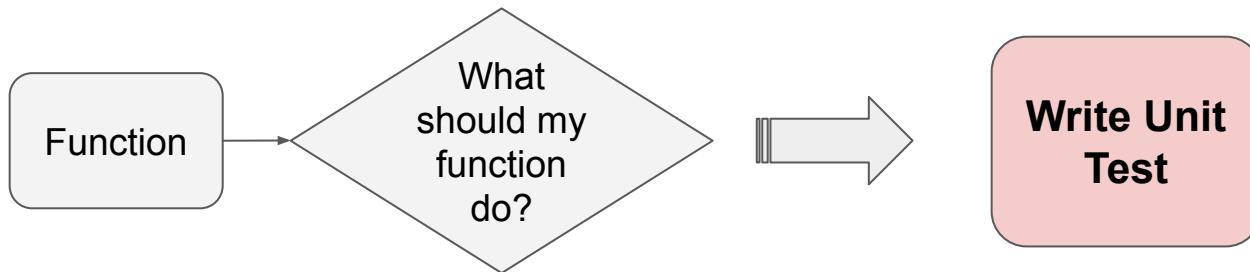
Documentation is essential for the package checks to pass

```
(devtools::check_man() )
```

```
#' Line data
#'
#' Returns departure and arrival data for a given `line`.
#'
#' @param line Scalar integer specifying the line.
#'
#' @return A `data.frame` with variables `line`, `date`,
#'   `weekday`, `hour`, `scheduled.departure`,
#'   `actual.departure`, `scheduled.arrival`,
#'   `actual.arrival`.
#'
#' @examples
#' data <- get_line_data(11)
#' head(data)
#' summary(data)
#'
#' @export
```



Modularisation: Unit Test





Modularisation: Unit Test

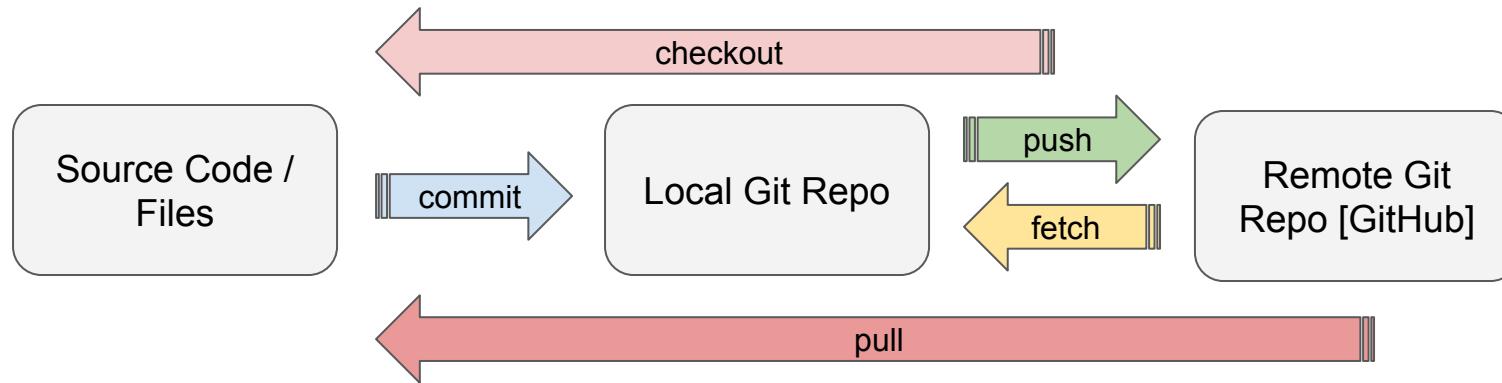
```
data <- data.frame(  
  scheduled_departure = c(1, 2) * 60,  
  actual_departure = c(3, 6) * 60,  
  scheduled_arrival = c(2, 3) * 60,  
  actual_arrival = c(6, 11) * 60  
)  
  
test_that("delays are computed correctly",  
{  
  res <- compute_delay(data)  
  expect_equal(res$delay, c(3, 6))  
})
```

Tests are useful for:

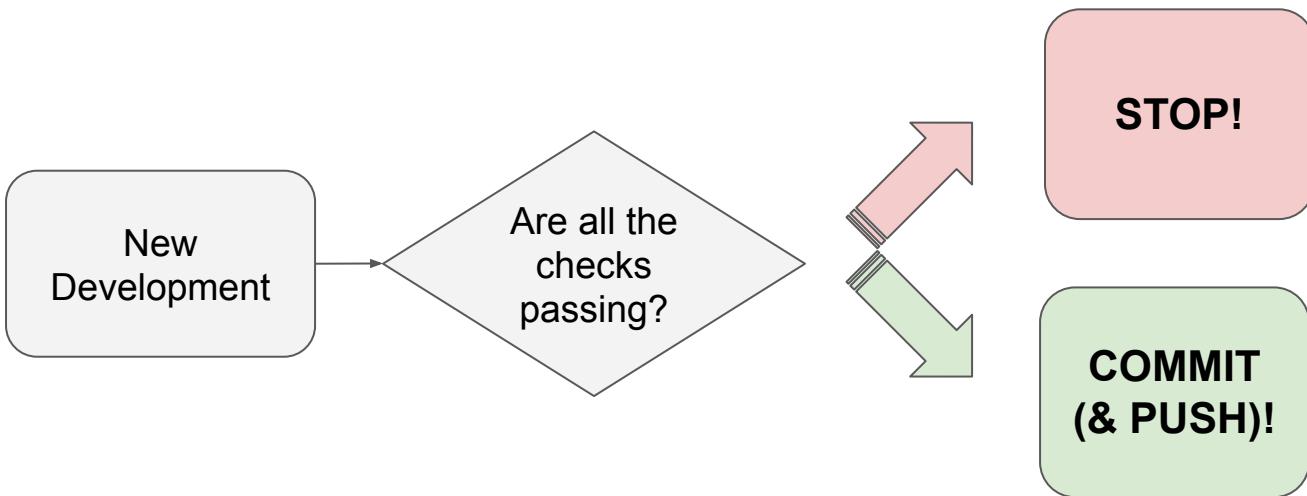
- Think what the function should do
- Make sure the functionality is not broken
- Provide minimal example
- Test expectations while developing



Version Control: GitHub

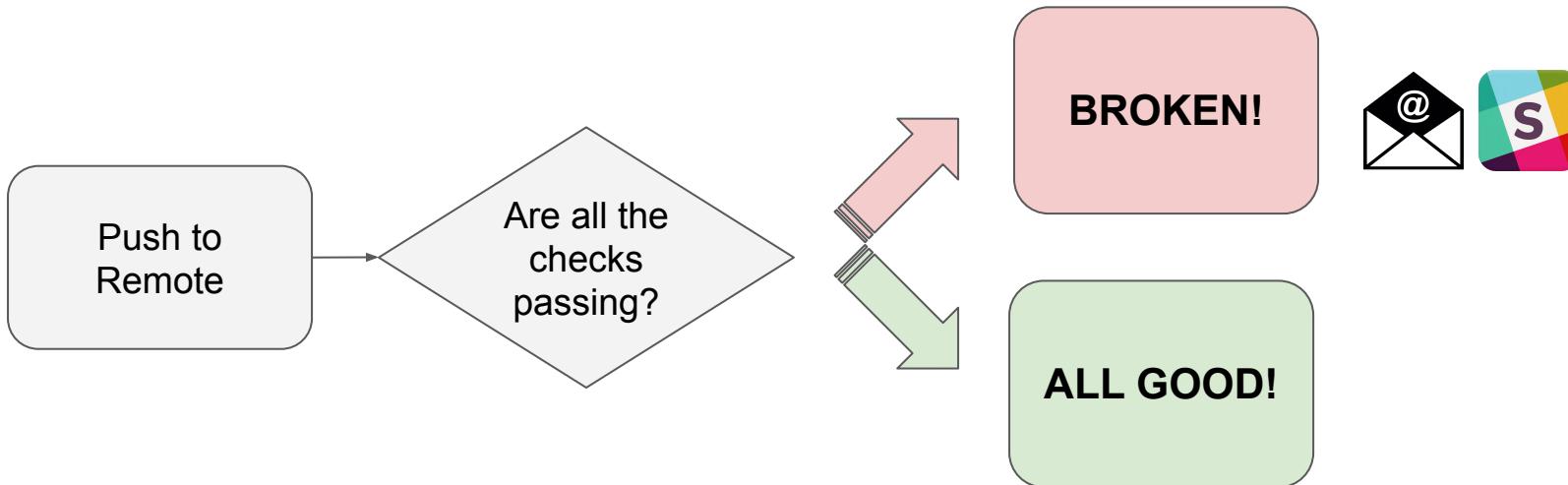


Local Package-level Checks



R CMD check

Automated Testing: Continuous Integration (CI)



R CMD check



TravisCI



“In [software engineering](#), continuous integration (CI) is the practice of merging all developers’ working copies to a shared [mainline](#) several times a day”

[wikipedia](#)

Travis CI is an open-source continuous integration service that can be used to build and test software projects hosted on GitHub.

- `usethis::use_travis()` → creates a Travis CI configuration file (`.travis.yml`)

```
language: r           .travis.yml
cache: packages
```

TravisCI



- To enable Travis CI go to Travis-ci.com and [Sign up with GitHub](#).
- On Travis CI the user needs to select the repositories on which to activate the continuous integration.
- Then GitHub notifies Travis CI of the occurrence of new commits / pull requests, triggering actions on Travis CI as specified in `.travis.yml`



Mirai Solutions GmbH

@miraisolutions

Repositories

We're only showing your public repositories. You can find your private projects on travis-ci.com.

Legacy Services Integration



vzb



vbzdelays



TravisCI



miraisolutions / vbzdelays

Unwatch ▾ 4 ⭐ Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Branch: master ▾

Commits on Sep 20, 2019

Enable Travis CI.
riccardoporreca committed 26 seconds ago

Delay calculation and analysis.
riccardoporreca committed 7 minutes ago

Added data loading functionality.
riccardoporreca committed 17 minutes ago

Some checks haven't completed yet

1 pending check

continuous-integration/travis-ci/push Pending — ... Details

b1c6b2f b6ce3fa 29f5d13

README.md

The README.md contains the basic information about a repository, including installation and usage examples.

vbzdelays

build passing

The goal of vbzdelays is to analyze VBZ delays.

Installation

You can install vbzdelays with:

```
remotes::install_github("miraisolutions/vbzdelays")
```

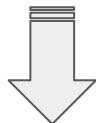
Example

Load line data

```
vbzdelays::get_line_data(line = 11)
```

Mean Delay

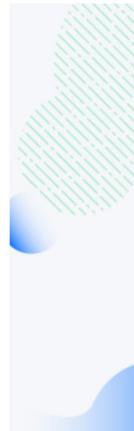
```
((actual_departure - scheduled_departure)
 + (actual_arrival - scheduled_arrival))
 / 2 / 60
```



```
mean(c(
  actual_departure - scheduled_departure,
  actual_arrival - scheduled_arrival
)) / 60
```

Broken: miraisolutions/vbzdelays#4 (master - 1b60f53) [Travis CI](#)

Travis CI <builds@travis-ci.org> [Unsubscribe](#)
to me ▾ 3:06 PM (0 minutes ago) ★

 miraisolutions / vbzdelays

⌚ master

⚠ Build #4 was broken > ⌂ 1 min and 46 secs

⚡ riccardoporreca [1b60f53 CHANGESET →](#)

Use mean() for delay calculation.

 miraisolutions / vbzdelays  build failing

**Packaged piece of software that
has all the functionality to
perform our analysis!**

Produce Analysis Report

```
render_line_report <- function(line, output_dir = ".", ...)  
{  
  report_file <- system.file("reports", "line-report.Rmd",  
                            package = "vbzdelays")  
  params <- structure(list(line = line),  
                        class = "knit_param")  
  rmarkdown::render(report_file, output_dir = output_dir,  
                   params = params, ...)  
}
```

Analysis Report

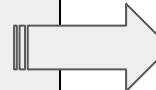
```
---
```

```
title: "Analysis of delays"
author: "Mirai Solutions"
date: "`r Sys.time()`"
output: html_document
params:
  line: -1
---

## Analysis of delays for line `r params$line`
### Delays by hour for each weekday

The following plot shows the distribution of delays by hour for each weekday.

```{r echo=FALSE, fig.asp=1.5}
data <- vbzdelays::get_line_data(params$line)
vbzdelays::plot_delays_by_weekday_hour(data)
````
```



Analysis of delays

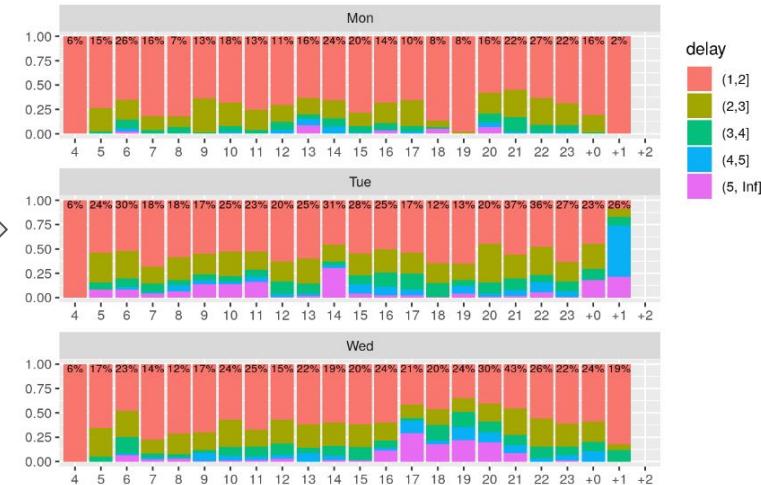
Mirai Solutions

2019-09-19 11:15:56

Analysis of delays for line 11

Delays by hour for each weekday

The following plot shows the distribution of delays by hour for each weekday.



**Packaged piece of software that
can be installed and run
anywhere!**

```
remotes::github_install("miraisolutions/vbzdelays"
)
vzbdelays::render_line_report(13)
```

Break!

See you back in 15 mins

Production Ready



Product: Website

```
render site <- function(output dir = "site", ...) {  
  # absolute path of output_dir (which must exist)  
  dir.create(output dir)  
  output dir <- normalizePath(output dir)  
  input dir <- system.file("site", package = "vzbdelays")  
  restore dir <- setwd(input dir) # returns the wd before the call  
  on.exit(setwd(restore dir), add = TRUE)  
  bookdown:::render book(".", output dir = output dir,  
                      clean_envir = FALSE, ...)  
}
```

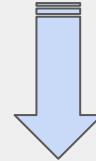
Reference bookdown

- <https://bookdown.org/yihui/bookdown/github.html>

Product: Website

One page report per line

```
for (line in vbzdelays::available_lines()) {  
  params <- list(line = line)  
  cat(knitr::knit child(  
    system.file("reports", "line-report.Rmd", package =  
"vbzdelays"),  
    quiet = TRUE  
  ))  
}
```



```
data <- vbzdelays::get_line_data(params$line)  
vbzdelays::plot_delays_by_weekday_hour( data)
```

Product

1 Introduction

2 Results by line

 2.1 Analysis of delays for line 2

 2.2 Analysis of delays for line 3

 2.3 Analysis of delays for line 4

 2.4 Analysis of delays for line 5

 2.5 Analysis of delays for line 6

 2.6 Analysis of delays for line 7

 2.7 Analysis of delays for line 8

 2.8 Analysis of delays for line 9

 2.9 Analysis of delays for line 10

 2.10 Analysis of delays for line 11

 2.10.1 Delays by hour for each weekday

 2.11 Analysis of delays for line 12

 2.12 Analysis of delays for line 13

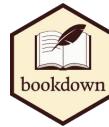
 2.13 Analysis of delays for line 14

 2.14 Analysis of delays for line 15

 2.15 Analysis of delays for line 17



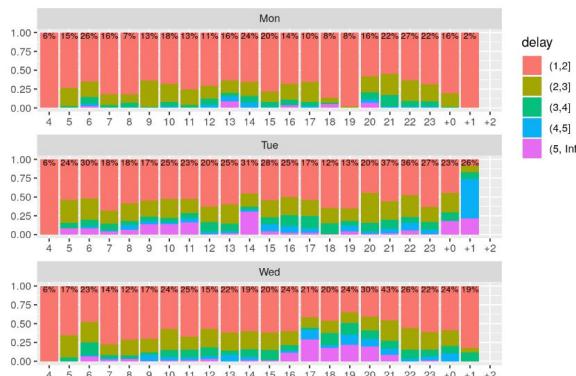
<http://miraisolutions.github.io/vbzdelays/>



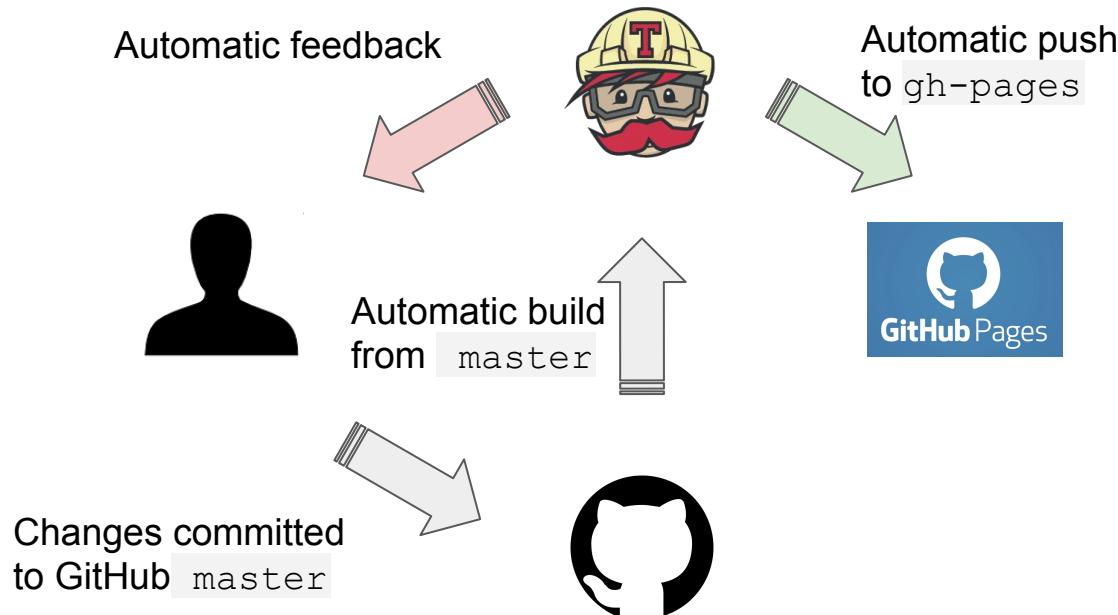
2.10 Analysis of delays for line 11

2.10.1 Delays by hour for each weekday

The following plot shows the distribution of delays by hour for each weekday.



Continuous Deployment (CD)





GitHub Pages: good-to-know

Reference GitHub Pages

- <https://help.github.com/en/articles/configuring-a-publishing-source-for-github-pages>
- <https://help.github.com/en/articles/user-organization-and-project-pages>

GitHub Pages requires a branch `gh-pages` with website html at top level repository to render the website.

GitHub Pages sites are subject to the following usage limits:

- Source repositories have a recommended limit of 1GB.
- Published GitHub Pages sites may be no larger than 1 GB.
- GitHub Pages sites have a soft bandwidth limit of 100GB per month.
- GitHub Pages sites have a soft limit of 10 builds per hour.



GitHub Pages: create gh-pages branch

```
git checkout --orphan gh-pages  
git rm -rf .  
echo "# Hello World" > index.md  
git add index.md  
git commit -m "Initial commit"  
git push origin gh-pages
```



Continuous Deployment

`$PKG_TARBALL` is defined at build time in Travis CI.

`$GITHUB_PAT` is a secure variable defined in the Travis CI settings for the repository.

How to create GitHub Personal Access Tokens (PAT):
<https://help.github.com/en/articles/creating-a-personal-access-token-for-the-command-line>



```
language: r
cache: packages

before deploy:
- R CMD INSTALL $PKG_TARBALL
- Rscript -e "vbzdelays::render_site()"

deploy:
  provider: pages
  skip cleanup: true
  github token: $GITHUB_PAT
  local_dir: _site
  on:
    branch: master
```

travis.yml



GITHUB_PAT

- GitHub Apps
- OAuth Apps
- Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

vbzdelays

What's this token for?

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

- | | |
|---|--------------------------------------|
| <input checked="" type="checkbox"/> repo | Full control of private repositories |
| <input checked="" type="checkbox"/> repo:status | Access commit status |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status |
| <input checked="" type="checkbox"/> public_repo | Access public repositories |
| <input checked="" type="checkbox"/> repo:invite | Access repository invitations |

GITHUB_PAT



miraisolutions / vbzdelays  build passing

Current Branches Build History Pull Requests **Settings**

More options 

Environment Variables

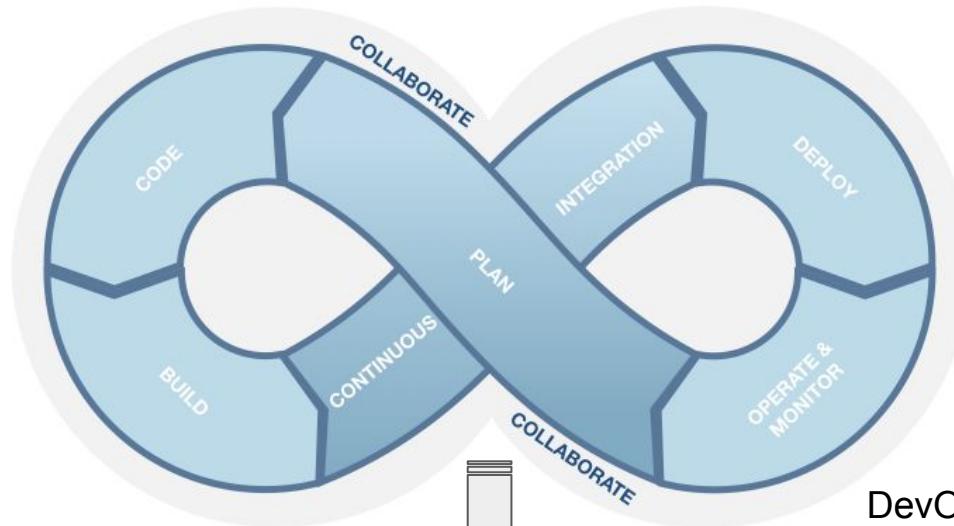
Customize your build using environment variables. For secure tips on generating private keys [read our documentation](#)

 If your secret variable has special characters like `&`, escape them by adding `\` in front of each special character. For example, `ma&w!doc` would be entered as `ma\&\w\!doc`.

| NAME | VALUE | BRANCH | DISPLAY VALUE IN
BUILD LOG | Add |
|---|---|--|-------------------------------|------------------------------------|
| <input type="text" value="GITHUB_PAT"/> | <input data-bbox="456 864 744 907" type="text" value="Value"/>
<small>This field is required</small> | <input data-bbox="768 864 1056 907" type="text" value="All branches"/>
<small>▼</small> | <input type="checkbox"/> | <input type="button" value="Add"/> |

Benefits of CI/CD

DevOps



Agile

DevOps in 3 sentences:
<https://dev.to/ashokisaac/devops-in-3-sentences-17c4>

DevOps + Agile:
<https://resources.collab.net/devops-101/what-is-devops>

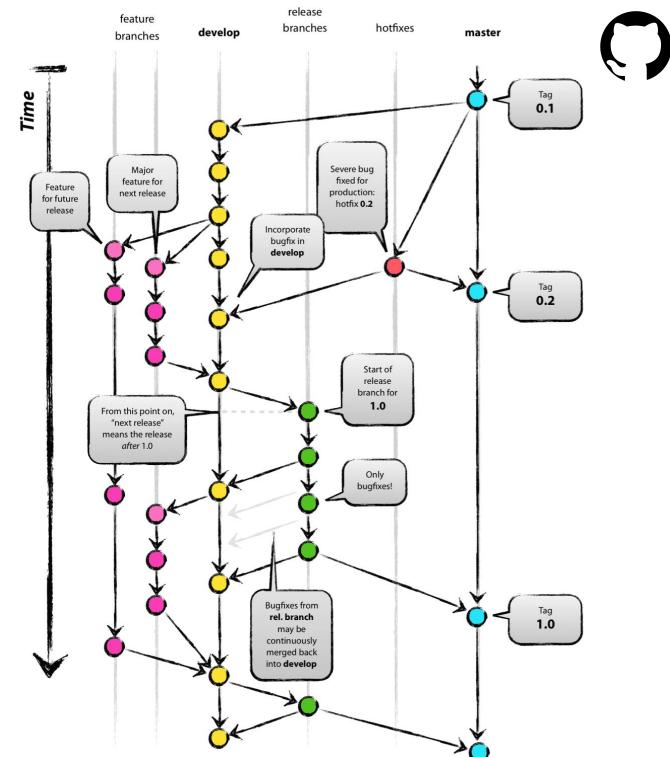
Gitflow

Gitflow is a standard setup that allows a collaborative development, while assuring the safety of the released product.

According to the Gitflow approach, the central repo holds two main branches with an infinite lifetime:

- `master`: branch where source code of HEAD always reflects a production-ready state.
- `develop`: branch where source code of HEAD always reflects a state with the latest delivered developments for the next release.

Each feature branch provides an additional incremental value to the product, potentially ready for deployment.

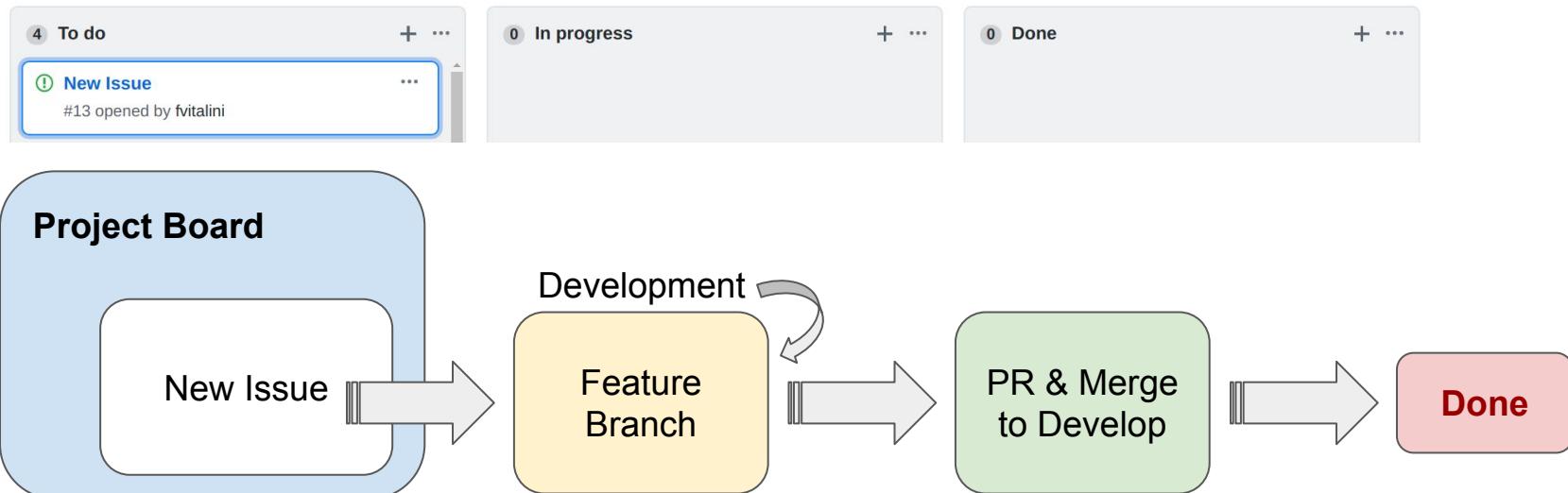


<https://nvie.com/posts/a-successful-git-branching-model/>



Collaboration

Gitflow is an **agile** framework by nature:



Control Branches



Set restrictions on branches: require review and status check (travis-ci) before merging to `master` or `develop`.

A screenshot of a GitHub repository page for "miraisolutions / vbzdelays". The page shows various navigation links: Code (selected), Issues (1), Pull requests (0), Projects (1), Wiki, Security, Insights, and Settings. A red box highlights the "Settings" button. To the right, a sidebar menu is open, showing options like Options, Collaborators & teams, and a expanded section for Branches. A large gray arrow points from the "Settings" button towards the "Branches" section of the sidebar. The sidebar also includes sections for Webhooks, Notifications, Integrations & services, Deploy keys, and Security alerts.



Control Branches

Rule settings

Protect matching branches
Disables force-pushes to all matching branches and prevents them from being deleted.

Require pull request reviews before merging
When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.
Required approving reviews: 1 ▾

Dismiss stale pull request approvals when new commits are pushed
New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

Require review from Code Owners
Require an approved review in pull requests including files with a designated code owner.

Restrict who can dismiss pull request reviews
Specify people or teams allowed to dismiss pull request reviews.

Require status checks to pass before merging
Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

Require branches to be up to date before merging
This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

Status checks found in the last week for this repository

| | |
|--|----------|
| <input checked="" type="checkbox"/> continuous-integration/travis-ci | Required |
| <input type="checkbox"/> github/pages | |

Require signed commits
Commits pushed to matching branches must have verified signatures.

Include administrators
Enforce all configured restrictions for administrators.

Merge to
Develop





New Issue

Add coverage label

Related Issues [Beta](#). Try it.

Write Preview AA B i “ “ <> @ ↵

Extend the plots so that for every hour the percentage of delayed lines is displayed as a label.

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

[Submit new issue](#)

| | |
|------------------------|--------------------|
| Assignees | ⚙️ |
| No one—assign yourself | ⚙️ |
| Labels | ⚙️ |
| None yet | ⚙️ |
| Projects | ⚙️ |
| MiraiLabs/From_prot... | ⚙️ |
| Milestone | ⚙️ |
| No milestone | ⚙️ |

Create a new branch,

`feature/1-add-coverage-label`

Develop the feature within the branch

Once done, make a pull request back to `develop`



Pull Request

Show tests on pull request level

Review required
At least 1 approving review is required by reviewers with write access. [Learn more.](#)

Some checks haven't completed yet Hide all checks
2 pending checks

- continuous-integration/travis-ci/pr Pending — The Travis CI build is in progress. Required Details
- continuous-integration/travis-ci/push Pending — The Travis CI build is in progress. Required Details

Merging is blocked
Merging can be performed automatically with 1 approving review

[Merge pull request](#)

Review required
At least 1 approving review is required by reviewers with write access. [Learn more.](#)

All checks have failed Hide all checks
2 failing checks

- continuous-integration/travis-ci/pr — The Travis CI build failed Required Details
- continuous-integration/travis-ci/push — The Travis CI build failed Required Details

Merging is blocked
Merging can be performed automatically with 1 approving review

[Merge pull request](#)

4 In progress

Use mean() for delay calculation ...
#4 opened by riccardoporreca

Review required

Release Concept



According to the Gitflow approach, no work should be pushed directly to `master`, which is updated as part of a release with a pull request from a stable `develop` branch or a `release` branch.

- Create a `release` branch (e.g. `release/v2.0.0`) from `develop`
- Consolidate the changelog (`NEWS.md` in R, see e.g. [recommendations](#) and [style guide](#))
- Update package version (`NEWS.md` and `DESCRIPTION` files in R)
- Create and merge a new pull request `release/v2.0.0` → `master`
- Create a new release tag on `master` (on GitHub [Code > releases > Draft new release](#))
- Create a new pull request and merge `release/v2.0.0` → `develop`
- Prepare for next version on `develop` (by changing the version `NEWS.md` and `DESCRIPTION` files)



Continuous Deployment

Release v1.1.0 #5
riccardoporreca wants to merge 4 commits into `master` from `release/v1.1.0`

Review required Show all reviewers
At least 1 approving review is required by reviewers with write access. [Learn more.](#)

1 pending reviewer

Some checks haven't completed yet Hide all checks
1 pending and 1 successful checks

- continuous-integration/travis-ci/pr** Pending — The Travis CI build is in prog... **Required** [Details](#)
- continuous-integration/travis-ci/push** — The Travis CI build passed **Required** [Details](#)

Merging is blocked
Merging can be performed automatically with 1 approving review.

[Merge pull request](#) or view [command line instructions](#).

```
563 Setting environment variables from repository settings
564 $ export GITHUB_PAT=[secure]
565
▶ 566 Setting up build cache
572
▶ 573 Setting up package cache
▶ 576 R session information
▶ 599 Installing package dependencies
▶ 696 Building package
▶ 706 Checking package
799 The command "grep -q -R "WARNING" "${RCHECK_DIR}/00check.log"" exited with 0.
▶ 800 store build cache
827
▶ 828 $ R CMD INSTALL $PKG_TARBALL
▶ 842 $ Rscript -e "vbdelay::render_site()"
▶ 873 $ rvm $(travis_internal_ruby) --fuzzy do ruby -S gem install dpl
876
▶ 877 Installing deploy dependencies
887 Logged in as @riccardoporreca (Riccardo Porreca)
888 cd /tmp/d20190921-12884-1dxo8cq/work
▶ 889 Preparing deploy
▶ 891 Deploying application
916 Done. Your build exited with 0.
```



Release Tag

The screenshot shows the GitHub interface for creating a new release tag. The repository is `miraisolutions / vbzdelays`. The **Releases** tab is selected. A new tag `v1.1.0` is being created, targeting the `master` branch. The message for the tag is `vbzdelays 1.1.0`. The release notes in the preview pane mention the addition of a percentage of delayed connections to the `plot_delays_by_weekday_hour()` function. The release is shown as the latest release, version `v1.1.0`, commit hash `086c132`, and is verified.

miraisolutions / vbzdelays

Code Issues 1 Pull requests 1 Projects 1 Wiki Security

Releases Tags

v1.1.0 @ Target: master

Excellent! This tag will be created from the target when you publish this release.

vbzdelays 1.1.0

Write Preview

• `plot_delays_by_weekday_hour()` extended with the percentage of delayed connections (#1).

Latest release

v1.1.0
086c132
Verified

vbzdelays 1.1.0

fvitolini released this now

- `plot_delays_by_weekday_hour()` extended with the percentage of delayed connections (#1).

Why DevOps

Goals

- Achieve faster time to production
- Enable more frequent deployments
- Lower failure rate of new releases
- Shorten lead time between fixes
- Enhance collaboration between groups

Challenges

- Mindset and Culture
- Security
- Responsibility

Benefits

- Development via small work items increases the speed of development and improves its quality.
- Slow, repetitive and human made actions are error-prone. Automation in testing and deployment increases the stability of the production process.
- Version controlled development ensures traceability and repeatability of the development.
- Collaboration and an agile approach are keys to a functional production pipeline.
- Building, testing and deploying discrete development components improve confidence in the final product.

Note:

- Infrastructure can also follow a continuous delivery philosophy (Infrastructure as Code).

DevOps is a culture. Tools can enable it.

A well-structured
development
approach allows for
a variety of
deployment targets



Q&A

Closing Remarks

MiraiLabs - Next Steps

Feedback welcome:

- Which are the topics you are interested in?
- Format of the workshop: Facilities, materials.

Get in touch!

labs@mirai-solutions.com

Thank you!
Apero Time