



OS06 - R Shiny from Code to Cloud

PHUSE EU Connect 2024

Abu Zafar Mohammad Sami, *Principal Statistical Programmer*

Yann Féat, *Statistician*



www.mainanalytics.de



Agenda



- Overview
- Challenges
- Development Workflow
 - Shiny Modules
 - Package Development
- Shiny App in Cloud
 - Continuous Integration
 - Continuous Delivery and Deployment
- App Structure and Examples
- Conclusion

R Shiny Overview

```
ui <- fluidPage(
```

```
  titlePanel(<your title>)
```

```
  sidebarLayout
```

```
    sidebarPanel(<input widgets>)
```

```
    mainpanel (<output>)
```

```
server <- function(input, output,server) {
```

```
  <calculations>
```

```
}
```

```
shinyApp(ui=ui, server=server)
```

mainanalytics GmbH

Select Types of Analysis

KAPLAN-MEIER SURVIVAL ANALYSIS

ANCOVA ANALYSIS

SUMMARY STATISTICS

MORE ▾

Km Type

☒ Plot

☐ Summary

☐ Details

Analysis Datasets*

adtte.xpt ▾

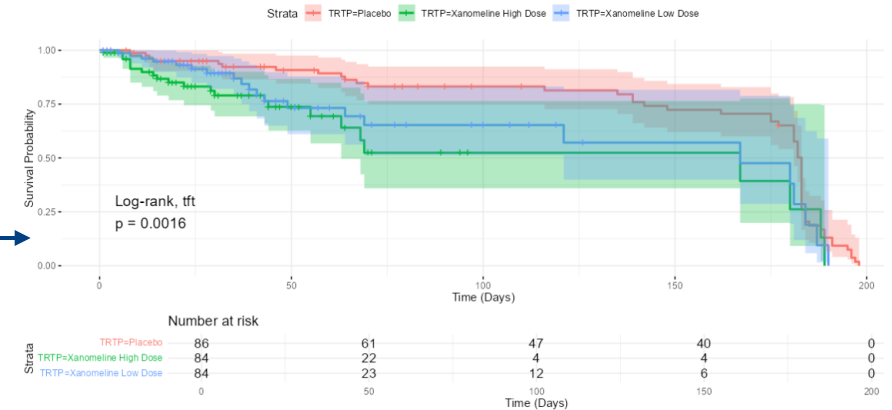
Choose Parameter*

PARAM ▾

Choose Parameter Value*

Time to First Dermatologic Event

PLOT KAPLAN-MEIER



```
fit <- survfit(<extract parameters from user selections>)  
ggsurvplot <- (fit, <draw plot>)
```

R Shiny Overview

- Web App Framework
 - Compiles your code into HTML, CSS and JavaScript
 - needed to display your application on web
- Good tutorial: mastering-shiny.org

R Shiny Overview

Possible use cases in clinical research

- Data exploration (pre-clinical, clinical), also in laboratory analyses
- Clinical trial monitoring (safety analyses, data checks...)
- FDA Submission

R Shiny Overview

Existing usage in pharma industry

- Safety review of overall and patient-specific information from ongoing trials
- Exploratory statistical analysis in studies
- Patient profiles (lab, AEs, ...)

R Shiny Challenges

- Code reaching cliff of complexity
- Documentation and testing of features
- Imposing good software engineering practices
- Complexity of continuous integration, development and deployment

R Shiny Development Flow – Shiny Modules

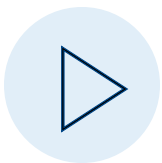
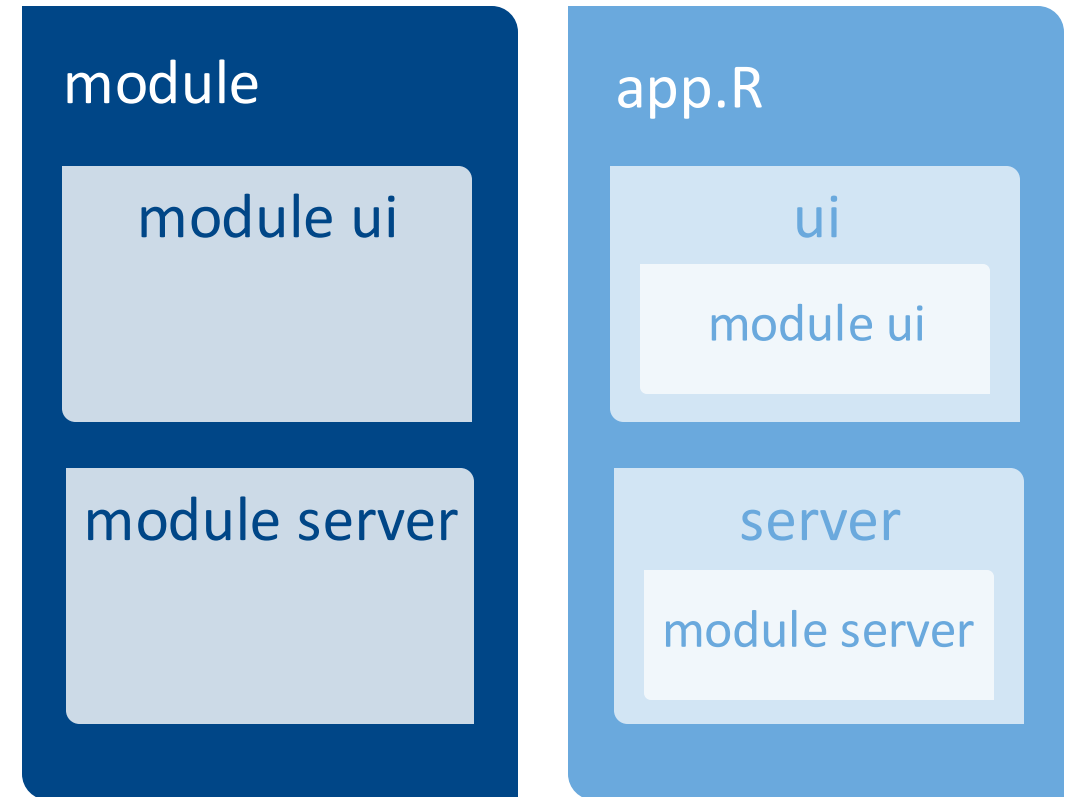
- Write scripts for producing each form (UI/SERVER)
- Code complexity triggered as more features introduced

The screenshot displays the 'main analytics Shiny portfolio' interface. On the left, the 'Analysis Dataset' section includes dropdowns for 'advs', 'SAFFL', 'PARAMCD', 'AVISITN', 'ANL01FL', and 'ATPT', along with checkboxes for 'Y', 'DIABP', '0', 'Y', and 'AFTER LYING I'. Below this, a 'Data' section shows '253 rows' and a table with columns 'Endpoint', 'Covariate', and 'Order'. The 'ANOVA' option is highlighted in blue. The main panel, titled 'Parameters', contains a 'Method' dropdown set to 'Type III ANCOVA'. Below this are sections for 'Endpoint' (AVAL), 'Treatments' (TRTP), 'Subgroups' (AGEGR1), and 'Covariate' (AVAL). Each section has a 'Label' field and an 'Order' dropdown. The 'Interaction with treatment' dropdown is set to 'No'. A 'Submit' button is at the bottom. The 'Output' section at the bottom displays the text: 'Type III ANCOVA of Diastolic Blood Pressure (mmHg) by Planned Treatment adjusted for Diastolic Blood Pressure (mmHg)'.

R Shiny Development Flow – Shiny Modules

- An app is divided up into pieces for individual features and each piece has a unique pair name.
- A module is a piece of a Shiny app reusable inside a more complex app.

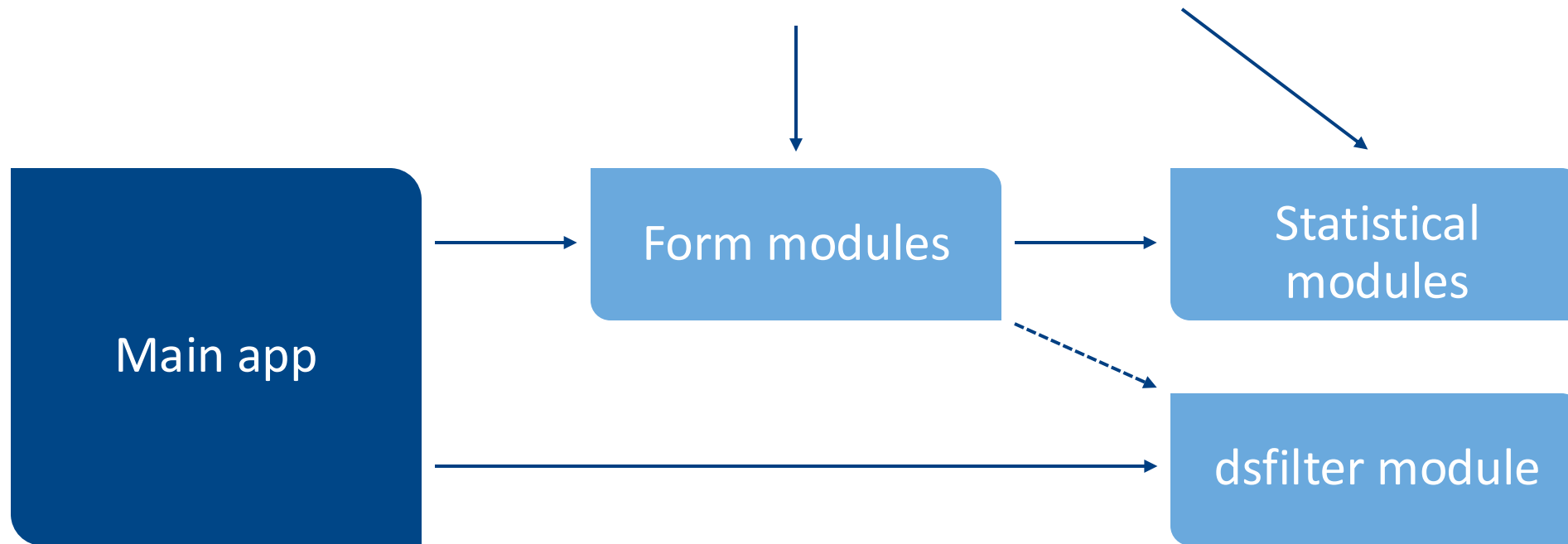
Modules don't just help organize your code; they help you organize your thinking.



[R Consortium - Structure your app:
introduction to Shiny modules](#)

R Shiny Development Flow – Shiny Modules

Automatically detected by the engine:
no need to update main app when adding a module



R Shiny Development Flow – Shiny Modules

Example

- module-form-anova.R
- module-form-numbycat.R
- module-form-numbynum.R
- module-form-surv.R
- module-stat-anova-ancova.R**
- module-stat-data-dataview.R
- module-stat-numbycat-boxpl.R
- module-stat-numbynum-linepl.R
- module-stat-numbynum-scatter.R
- module-stat-surv-cif.R
- module-stat-surv-km.R

```
lm_model <- lm(ancova_formula, data = ds_adam)
ancova <- Anova(lm_model,
               type = 3,
               contrasts = list(topic = contr.sum, sys = contr.sum)
)

# Extract F-tests p-values
pval_format <- label_pvalue(accuracy = 0.001)
covar_pval <- pval_format(ancova[2, "Pr(>F)"])
trt_pval <- pval_format(ancova[3, "Pr(>F)"])
if (subgrp_anal) {
  subgrp_pval <- pval_format(ancova[4, "Pr(>F)"])
  if (subgroup_intertrt) {
    trt_x_subgrp_pval <- pval_format(ancova[5, "Pr(>F)"])
  }
}

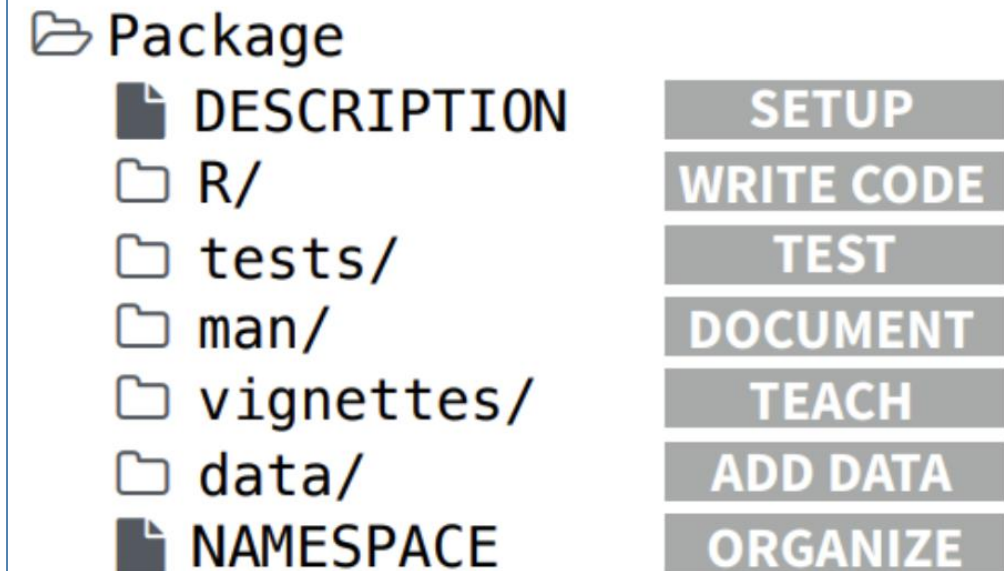
# Adjust means and ICs: LS Means and weight proportionately to match
# OM option on PROC GLM in SAS (as per the R Consortium R submission
# Pilot 2 Project)
ancova_adjust <- lsmeans(lm_model, cat_vars,
                        by = by_cat, weights = "proportional"
) %>%
  update(calc = c(n = ".wgt."))
```

R Shiny Development Flow – Package Development

A package is a convention for organizing files into directories.

There are multiple packages useful for package development, incl. ***usethis*** (handily automates repetitive tasks).
Installing and loading devtools: wraps packages to access everything in one step.

The 7 common parts of an R Package



[Posit's package development cheatsheet](#)

R Shiny Development Flow – Package Development

Directory: DESCRIPTION

Imports

Packages that are needed by users at runtime

Suggests

Packages that are either needed for development tasks or might unlock optional functionality for your users

Remotes

Packages that are not in CRAN

VignetteBuilder

Lists any package that your package needs as a vignette engine

Documentation for package ‘shiny.portfolio’ version 0.1

- [DESCRIPTION file](#).

```
Package: shiny.portfolio
Title: mainanalytics' R Shiny Portfolio
Version: 0.1
Authors@R: c(
  person("Sami", "Abu", , "abu.sami@mainanalytics.de",
    role = c("cre", "aut")),
  person("Yann", "Féat", , "yann.feat@mainanalytics.de",
    role = c("cre", "aut")),
)
Description: This is a Shiny app displaying a portfolio of Shiny modules.
License: file LICENSE
Depends:
  R (>= 4.2.0)
Imports:
  admiral (>= 0.11.0),
  dplyr (>= 1.1.2),|
  DT (>= 0.28),
  ggplot2 (>= 3.4.4),
  haven (>= 2.5.3),
  magrittr (>= 2.0.3),
  rlang (>= 1.1.0),
  shiny (>= 1.7.4),
  shinyjs (>= 2.1.0),
  stringr (>= 1.5.0),
  survival (>= 3.4.0),
  survminer (>= 0.4.9),
Remotes:
  rdboyes/forester@a9ad7c4
Suggests:
  knitr (>= 1.45),
  testthat (>= 3.2.1)
Encoding: UTF-8
Roxygen: list(markdown = TRUE)
RoxygenNote: 7.2.3
VignetteBuilder: knitr
```

R Shiny Development Flow – Package Development

Directory: NAMESPACE

export

Functions of a package that makes it available for others to use

importFrom

Imports functions from other packages

```
export("%>%")
export(all_distinct_assert)
export(dsfilter_server)
export(dsfilter_ui)
export(form_anova_desc)
export(form_anova_server)
export(form_anova_ui)
export(stat_anova_ancova_desc)
export(stat_anova_ancova_server)
export(stat_anova_ancova_ui)
export(stat_surv_cif_desc)
export(stat_surv_cif_server)
export(stat_surv_cif_ui)
export(stat_surv_km_desc)
export(stat_surv_km_server)
export(stat_surv_km_ui)
importFrom(DT, dataTableOutput)
importFrom(DT, datatable)
importFrom(DT, formatStyle)
importFrom(DT, renderDataTable)
importFrom(DiagrammeR, grViz)
importFrom(admiral, convert_blanks_to_na)
```

R Shiny Development Flow – Package Development

Directory: R/man

Documentation of all exported functions of a package

- Require .Rd files written in Latex-style notation in the man/ directory
- Roxygen2 comments to the .R files and then run `devtools::document()` to convert the roxygen comments to .Rd file

ROXYGEN2

The **roxygen2** package lets you write documentation inline in your .R files with shorthand syntax.

- Add roxygen documentation as comments beginning with **#'**.
- Place a roxygen **@** tag (right) after **#'** to supply a specific section of documentation.
- Untagged paragraphs will be used to generate a title, description, and details section (in that order).



```
#' Add together two numbers
#'
#' @param x A number.
#' @param y A number.
#' @returns The sum of `x` and `y`.
#' @export
#' @examples
#' add(1, 1)
add <- function(x, y) {
  x + y
}
```

COMMON ROXYGEN TAGS

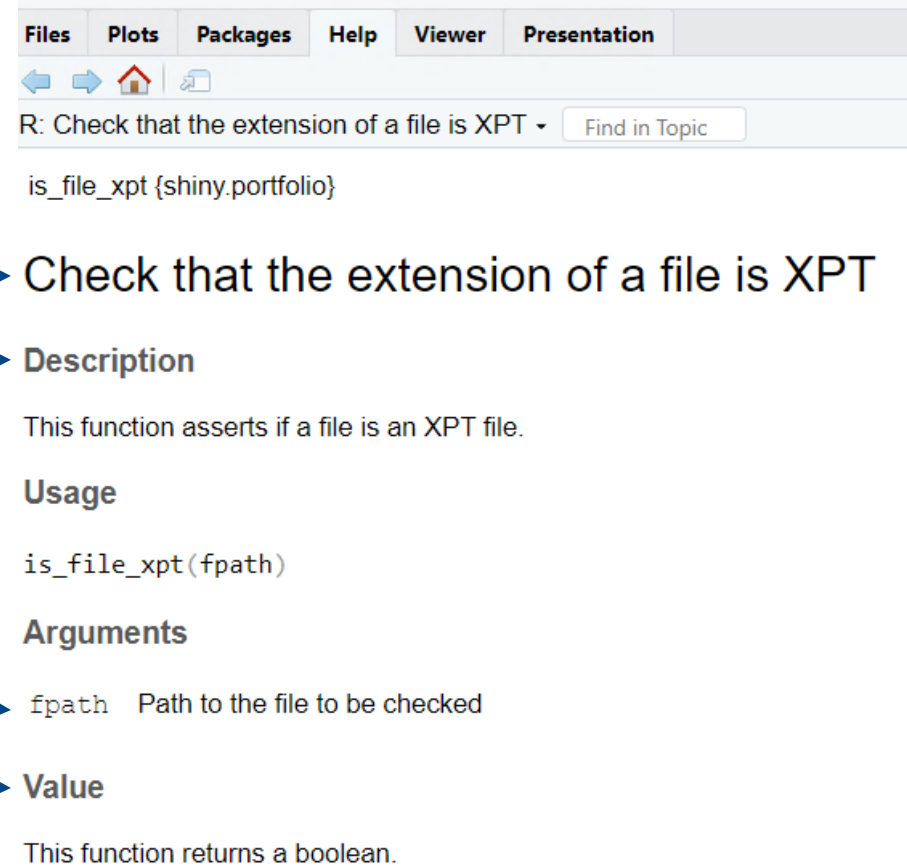
@description	@family	@returns
@examples	@inheritParams	@seealso
@examplesIf	@param	
@export	@rdname	

R Shiny Development Flow – Package Development

Directory: R/man

```
#' Check that the extension of a file is XPT
#'
#' This function asserts if a file is an XPT file.
#'
#' @param fpath Path to the file to be checked
#'
#' @return This function returns a boolean.
#'
is_file_xpt <- function(fpath) {
  fext <- unlist(strsplit(basename(fpath), "\\.")) [2]

  return(tolower(fext) == "xpt")
}
```



Files Plots Packages Help Viewer Presentation

R: Check that the extension of a file is XPT ▾ Find in Topic

is_file_xpt {shiny.portfolio}

Check that the extension of a file is XPT

Description

This function asserts if a file is an XPT file.

Usage

is_file_xpt(fpath)

Arguments

fpath Path to the file to be checked

Value

This function returns a boolean.

R Shiny Development Flow – Package Development

Directory: Vignettes

An instructive tutorial demonstrating practical uses of the software with discussion of the interpretation of the results.

- Create a vignette of package with `use_vignette()`
- Write the body of vignettes in R Markdown

R Shiny Development Flow – Package Development

Directory: Tests

Add automated unit tests of functionality

- Leads to test driven development
- Enhances code robustness

 tests/



- ✓ Set up test infrastructure with **use_testthat()**.
- ✓ Create a test file with **use_test()**.
- ✓ Write tests with **test_that()** and **expect_()**.
- ✓ Run all tests with **test()** and run tests for current file with **test_active_file()**.
- ✓ See coverage of all files with **test_coverage()** and see coverage of current file with **test_coverage_active_file()**.

R Shiny Development Flow – Package Development

Directory: Test

Exception handling of functions

- expect_equal(): Does code return the expected value?
- expect_type(): Does code return an object inheriting from the expected base type?
- expect_gt(): Does code return a number greater than the expected value?
- expect_length(): Does code return a vector with the specified length?

```
datasets_unfiltered <- import_folder_adam(adam_folder_test)
datasets_filtered <- import_folder_adam(adam_folder_test, "ads1")

test_that("output type is list", {
  expect_type(datasets_unfiltered, "list")
  expect_type(datasets_filtered, "list")
})

test_that("output length is correct", {
  expect_length(datasets_unfiltered, 10)
  expect_length(datasets_filtered, 1)
})

test_that("output elements have all character columns converted to factors", {
  d_ufltr <- datasets_unfiltered[["ads1"]]
  d_fltr <- datasets_filtered[["ads1"]]
  expect_equal(dim(d_ufltr[, sapply(d_ufltr, class) == "character"])[2], 0)
  expect_equal(dim(d_fltr[, sapply(d_fltr, class) == "character"])[2], 0)
  expect_gt(dim(d_ufltr[, sapply(d_ufltr, class) == "factor"])[2], 0)
  expect_gt(dim(d_fltr[, sapply(d_fltr, class) == "factor"])[2], 0)
})
```

<https://testthat.r-lib.org/reference/index.html>

R Shiny Development Flow – Package Development

Package: renv (Reproducible Environments)

System library

Contains all installed packages

Project library

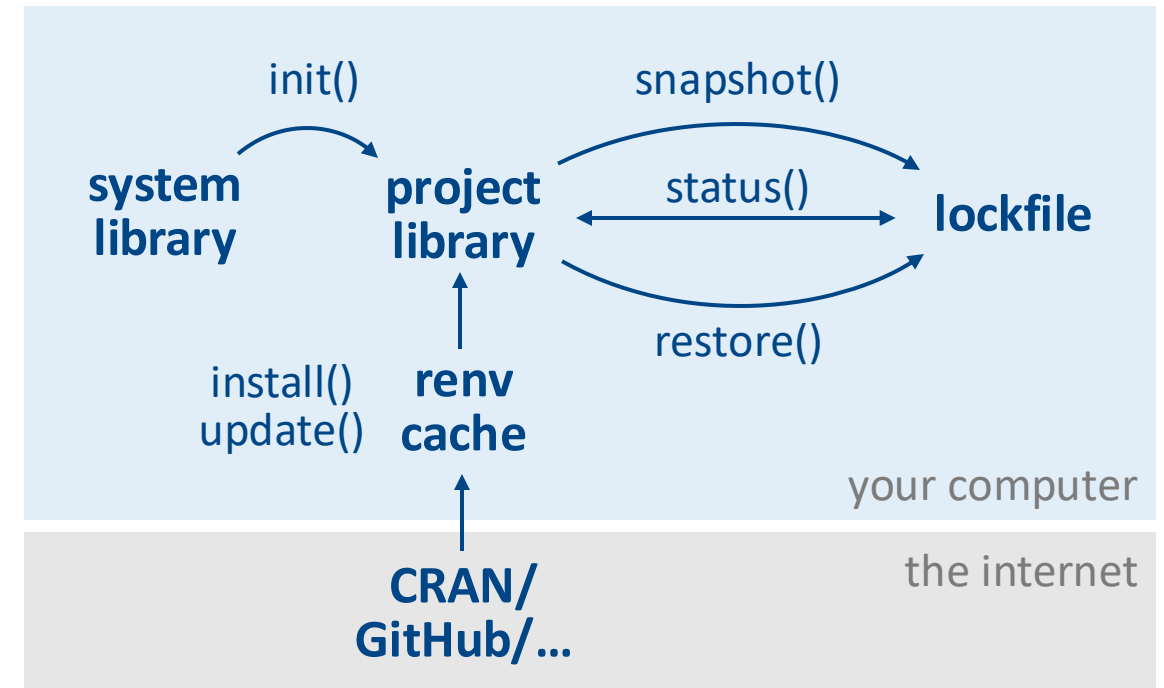
Gives each project its own independent collection of packages

Lockfile

Records metadata about packages to re-install them on a new machine

renv cache

Uses package cache, is shared across all projects



[Introduction to renv • renv \(rstudio.github.io\)](https://rstudio.github.io/renv/)

R Shiny Development Flow – Package Development

Package: renv (Reproducible Environments)

Advantages

- Resolves version control problem
- Installing and restoring packages is much faster, as renv can find and re-use previously installed packages from the cache
- Projects take up less disk space, because each project doesn't need to contain its own copy of every package

Example: Lockfile

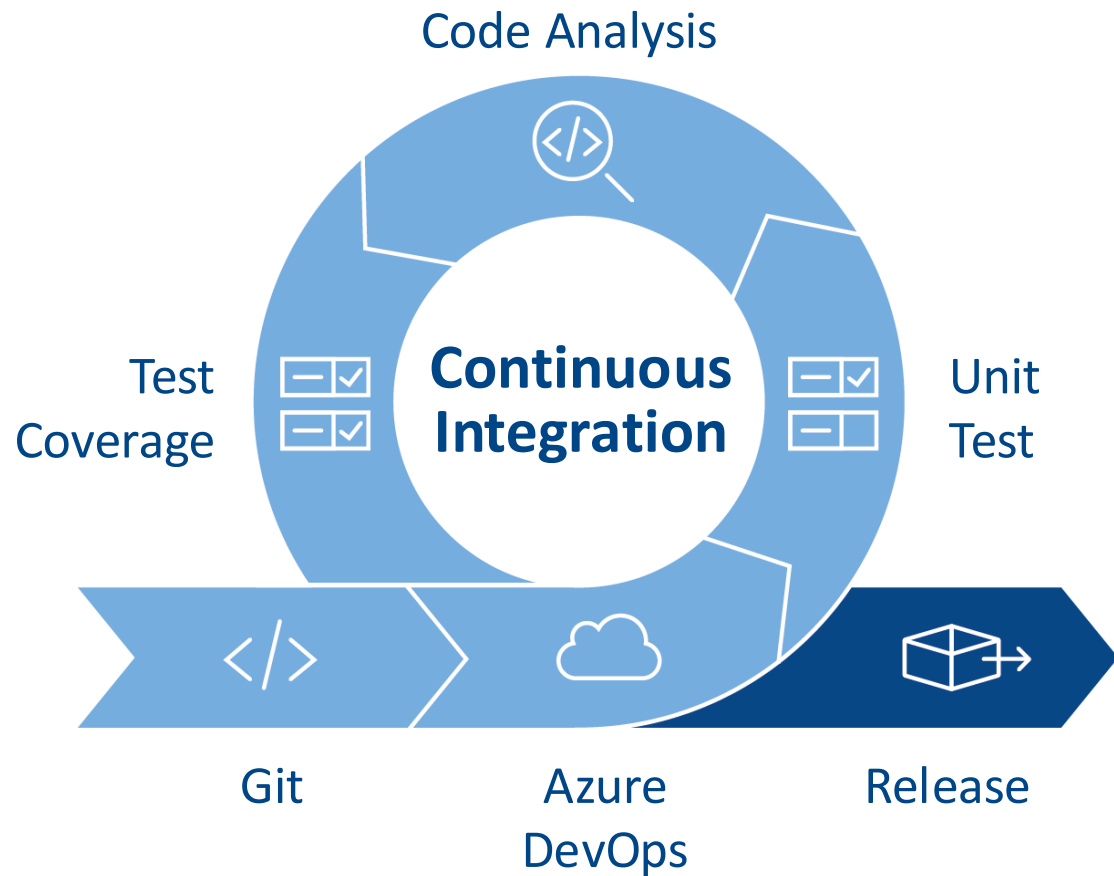
```
{
  "R": {
    "Version": "4.2.2",
    "Repositories": [
      {
        "Name": "CRAN",
        "URL": "https://cran.rstudio.com"
      }
    ]
  },
  "admiral": {
    "Package": "admiral",
    "Version": "1.1.1",
    "Source": "Repository",
    "Repository": "CRAN",
    "Requirements": [
      "R",
      "admiraldev",
      "cli",
      "dplyr",
      "hms",
      "lifecycle",
      "lubridate",
      "magrittr",
      "purrr",
      "rlang",
      "stringr",
      "tidyr",
      "tidyselect"
    ],
    "Hash": "a06f94a38ce50c4745ac7102f161bea4"
  },
}
```

Shiny App in Cloud

Continuous Integration

- Merged code changes into a central repository
- Automatically builds and runs unit tests on the new code changes to immediately surface any errors

Find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.



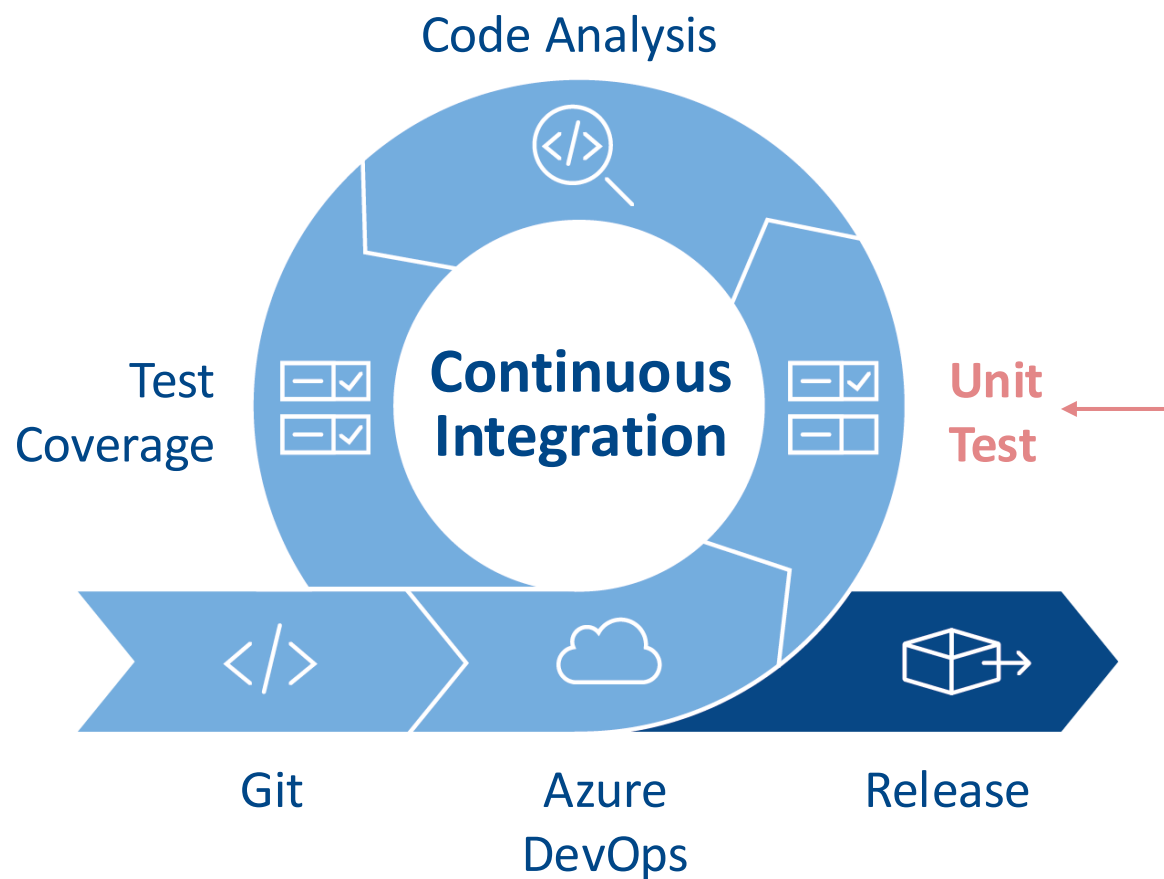
Shiny App in Cloud

Continuous Integration

Run as part of R CMD Checks

- Package structure
- R code
- Documentation
- Test: each file in tests is run
- Vignettes

Details: <https://r-pkgs.org/r-cmd-check.html>



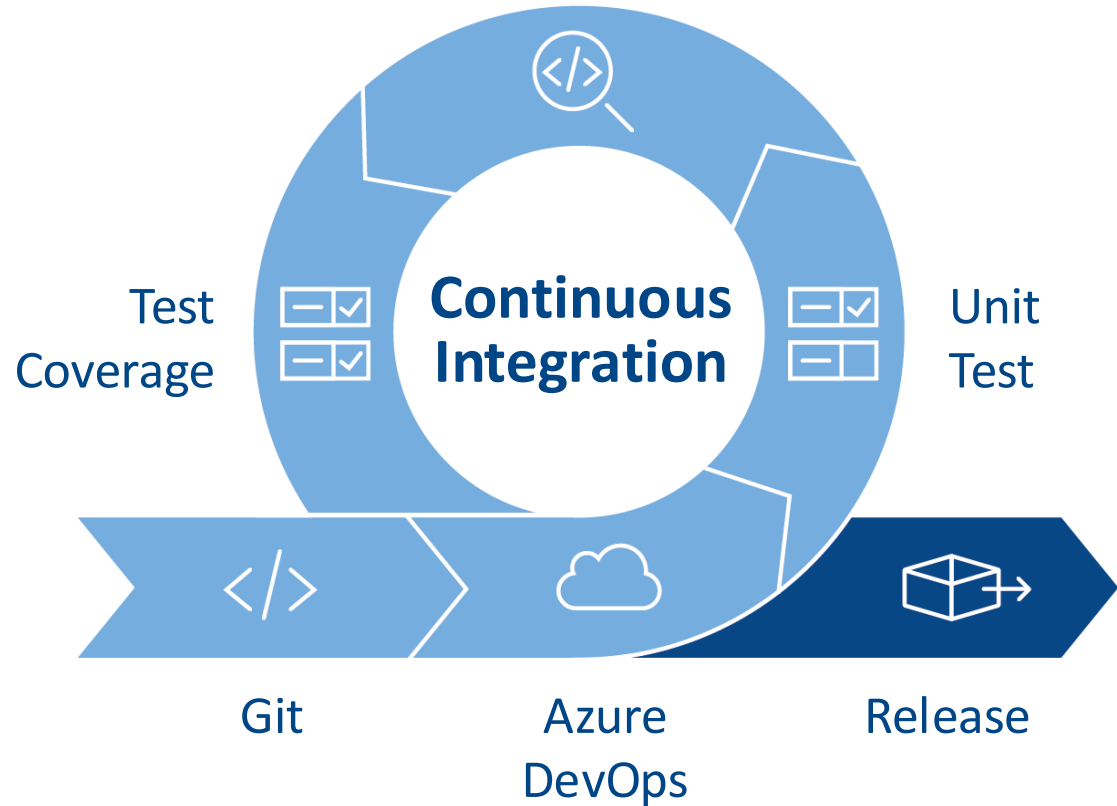
Shiny App in Cloud

Continuous Integration

[oxsecurity/megalinter](#) (lintr package) → Code Analysis

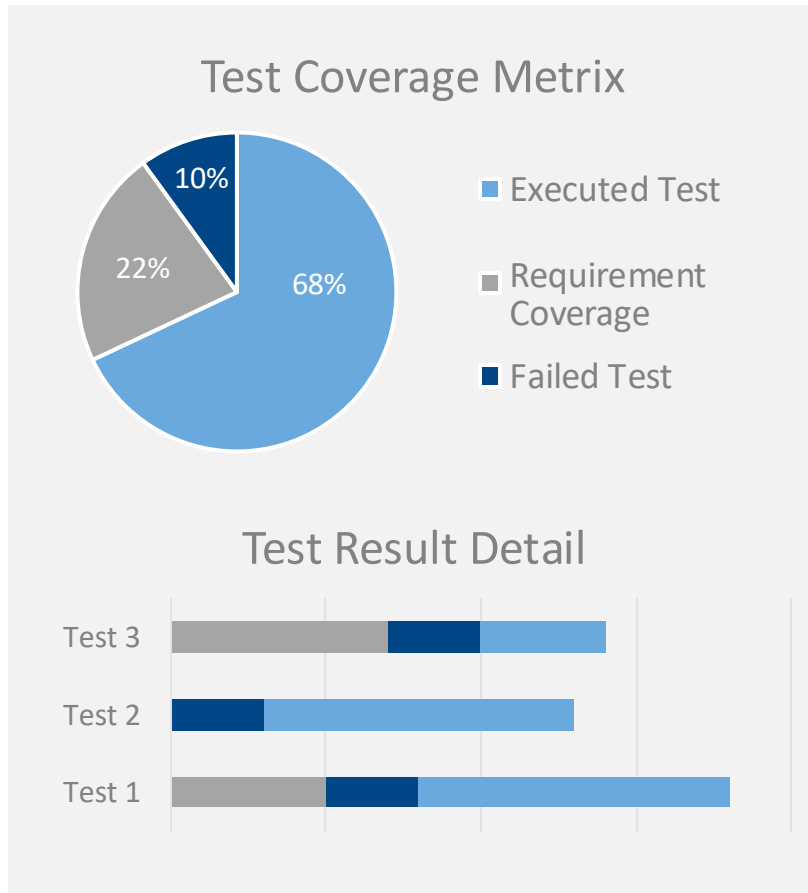
At each pull request, automatically analysis all code:

- Syntax Error
- Semantic Error



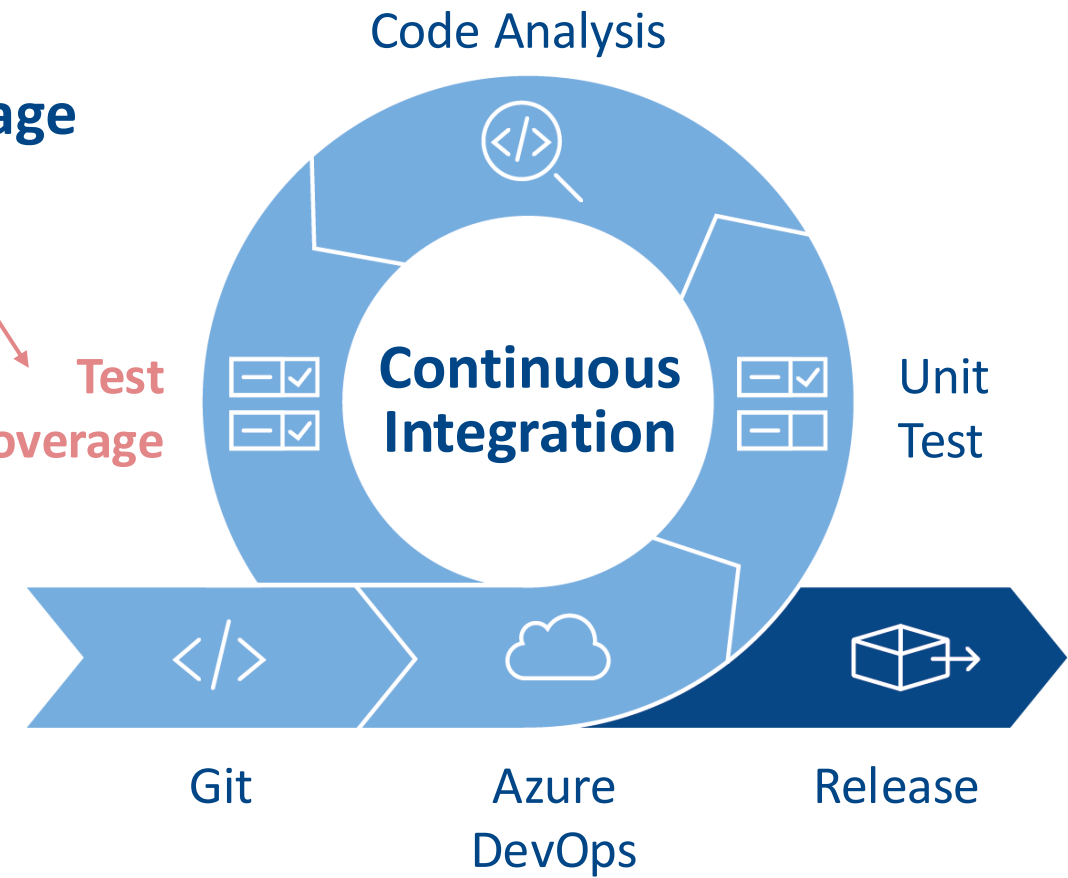
Shiny App in Cloud

Continuous Integration



covr Package

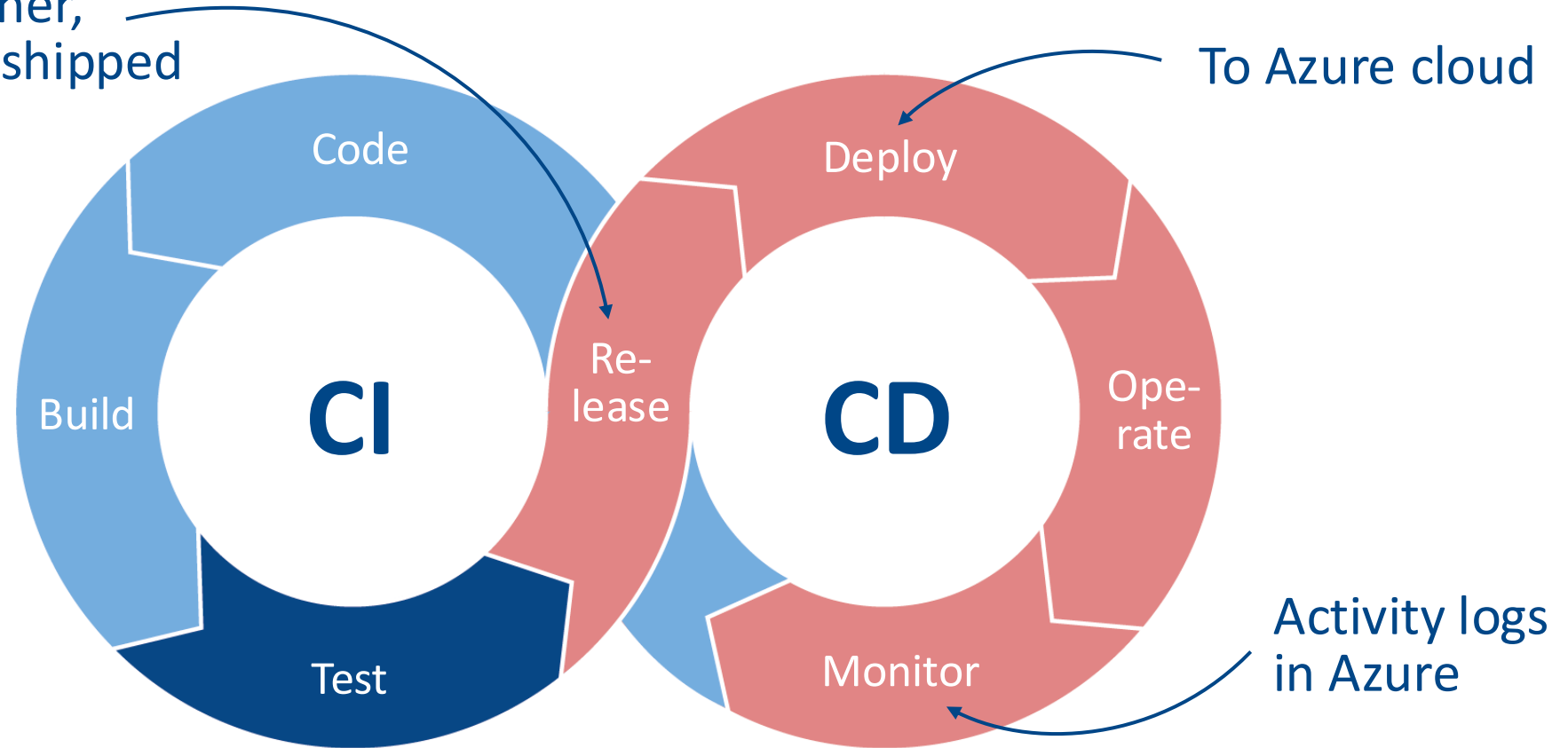
Test Coverage



Shiny App in Cloud

Continuous Delivery and Deployment (CD)

Build container,
ready to be shipped



Shiny App Structure and Examples

inst		Arbitrary files to be copied to the top level of the package's structure
└─ extdata		
└─ ...	XPT files use as source data for this demo app, come from CDISC's sample submission to the FDA	
└─ www		
└─ tmp		
└─ ...	Temporary files made visible to the client of the app, e.g., graph images, and files for download	
└─ ...	Static files made visible to the client of the app, e.g., logo, banner, favicon	Technical documentation
man		
└─ ...	Documentation files generated by Roxygen using tags in the code	
renv		Generated and used by the renv package
└─ ...	Renv R scripts	
renv.lock	renv snapshot: contains the full list of dependencies with versions	
tests		Unit testing
└─ testthat		
└─ ...	Unit tests	
└─ testthat.R	Script for initialisation of the unit tests	
vignettes		Package vignettes, automatically converted to a user-readable format upon building
└─ contrib.Rmd	Package contributor guide	

Shiny App Structure and Examples

R		R code
→ app-main.R	Defines the main function for starting the app using the UI and server functions	
→ form-engine.R	Engine to process form modules	
→ import.R	Functions to import the source data	
→ module-dsfilter.R	Dataset filtering module	
→ module-form-anova.R	Form module defining inputs for ANOVA/ANCOVA analyses	
→ module-form-numbycat.R	Form module defining inputs for analyses with numeric outcomes and categorical covariates	
→ module-form-numbynum.R	Form module defining inputs for analyses with numeric outcomes and numeric covariates	
→ module-form-surv.R	Form module for survival analyses	
→ module-stat-anova-ancova.R	Statistical module for the ANCOVA analysis	
→ module-stat-data-dataview.R	"Statistical" module for data viewing	
→ module-stat-numbynum-scatter.R	Statistical module for scatter plots	
→ module-stat-surv-cif.R	Statistical module for visualization of cumulative incidence functions	
→ module-stat-surv-km.R	Statistical module for Kaplan-Meier plots	
→ server.R	Shiny server component	
→ shiny.portfolio-package.R	For package-level documentation and some other package-wide housekeeping tasks	
→ stat-engine.R	Engine to process statistical modules	
→ ui.R	Shiny UI component	
→ utils-form.R	Utility functions for form modules	
→ utils-pipe.R	Defines magrittr's pipe operator (good practice for use with Roxygen)	
→ utils-stat.R	Utility functions for statistical modules	
→ utils-vignettes.R	Utility functions for vignettes	
→ utils.R	Global utility functions	
→ zzz.R	Side effects: code to be executed upon loading or attaching the package	

Shiny App Structure and Examples

 .dockerignore	List of files to be excluded when building the Docker image	Configuration files for the various utilities
 .gitignore	List of files to be excluded by default from git operations	
 .lintr	lintr configuration file (for code linting in the CI pipeline)	
 .mega-linter.yml	Mega Linter configuration file (for general linting in the CI pipeline)	
 .Rbuildignore	List of files to be excluded when the package is built	
 jscpd.json	JSCPD configuration file (for copy-paste detection in the CI pipeline)	
 azure-pipelines.yml	Azure DevOps configuration file for the CI pipeline	Configuration files for CI/CD and containerization
 azure-pipelines-1.yml	Azure DevOps configuration file for the CD pipeline	
 Dockerfile	Docker image configuration	
 sys_deps.txt	List of system dependencies to be installed in CI pipeline and Docker image	
 DESCRIPTION	Package description	Files for packaging
 LICENSE	Package license	
 NAMESPACE	Package namespace: defines exported functions, and imported functions from dependencies	
 README.md	Package instructions for developers and users	
 README.Rmd	R Markdown document used to generate README.md	
 .Rprofile	R code to be executed when an R session starts	Other
 shiny.portfolio.Rproj	Rproj file for easy project setup in R Studio	

Shiny App Structure and Examples

main analytics Shiny portfolio

Analysis Dataset

advs

Population

Parameter

Visit

Analysis flag

Other filter

32139 rows

Data

ANOVA

Continuous endpoint, categorical covariate

Continuous endpoint, continuous covariate

New widget pops up
with values of selected
variable

main analytics Shiny portfolio

Analysis Dataset

advs

SAFFL

Y

PARAM

Diastolic Bloo

AVISIT

Week 8

ANL01FL

Y

Other filter

567 rows

Data

ANOVA

Continuous endpoint, categorical covariate

Continuous endpoint, continuous covariate

Shiny App Structure and Examples

main analytics Shiny portfolio

Analysis Dataset

advs

Population

Parameter

Visit

Analysis flag

Other filter

32139 rows

Data

ANOVA

Continuous endpoint, categorical covariate

Continuous endpoint, continuous covariate

New widget pops up
with metadata values
of selected variables

main analytics Shiny portfolio

Analysis Dataset

advs

SAFFL

Y

PARAM

Diastolic Bloo

AVISIT

Week 8

ANL01FL

Y

Other filter

567 rows

Data

ANOVA

Continuous endpoint, categorical covariate

Continuous endpoint, continuous covariate

Shiny App Structure and Examples

ANCOVA Module: Selection

The screenshot displays the 'ANCOVA Module: Selection' interface. It features several dropdown menus and text input fields. The 'Method' dropdown is set to 'Type III ANCOVA'. The 'Endpoint' dropdown is set to 'CHG'. The 'Treatments' dropdown is set to 'TRTP'. The 'Subgroups' dropdown is set to 'Nothing selected'. The 'Covariate' dropdown is set to 'BASE'. Below these, there are three 'Label' input fields. The first two are 'Change from Baseline in Diastolic Blood' and 'Planned Treatment'. The third is 'Baseline Diastolic Blood Pressure (mmHg)'. Below the first two labels is an 'Order' dropdown set to 'TRTPN'. A red box highlights the first two labels and the 'Order' dropdown. An arrow points from this box to the text 'Labels and Order populated dynamically'. Another red box highlights the third label, with an arrow pointing down to the same text. A 'Submit' button is located at the bottom left.

Method
Type III ANCOVA

Endpoint
CHG

Treatments
TRTP

Subgroups
Nothing selected

Covariate
BASE

Label
Change from Baseline in Diastolic Blood

Label
Planned Treatment

Order
TRTPN

Label
Baseline Diastolic Blood Pressure (mmHg)

Submit

Labels and Order populated dynamically

Shiny App Structure and Examples

ANCOVA Module Output

Type III ANCOVA of Diastolic Blood Pressure (mmHg) by Planned Treatment

adjusted for Baseline Diastolic Blood Pressure (mmHg)

	n	Mean	SE	95% CI	p-value
Placebo					
Model Estimates	72	75	0.88	[74, 77]	
Xanomeline Low Dose					
Model Estimates	60	75	0.96	[73, 77]	
Contrast / Placebo		-0.5	1.3	[-3.1, 2.1]	0.698
Xanomeline High Dose					
Model Estimates	56	76	0.99	[74, 78]	
Contrast / Placebo		0.41	1.3	[-2.2, 3]	0.755
Contrast / Xanomeline Low Dose		0.92	1.4	[-1.8, 3.6]	0.507

Treatment p-value: 0.800

Covariate p-value: <0.001

Subgroup	Control	Comparator	Estimate
Xanomeline High Dose / Placebo			
All subjects	72	56	0.4 (-2.2 to 3.0)
Xanomeline High Dose / Xanomeline Low Dose			
All subjects	60	56	0.9 (-1.8 to 3.6)
Xanomeline Low Dose / Placebo			
All subjects	72	60	-0.5 (-3.1 to 2.1)

Shiny App Structure and Examples

ANCOVA Module: Subgroup

Method			
Type III ANCOVA			
Endpoint	Treatments	Subgroups	Covariate
CHG	TRTP	AGEGR1	BASE
Label	Label	Label	Label
Change from Baseline in Diastolic Blood	Planned Treatment	Pooled Age Group 1 (N)	Baseline Diastolic Blood Pressure (mmHg)
	Order	Order	
	TRTPN	AGEGR1N	
		Interaction with treatment	
		Yes	
Submit			

Shiny App Structure and Examples

ANCOVA Module Output: Subgroup

Type III ANCOVA of Diastolic Blood Pressure (mmHg) by Planned Treatment					
adjusted for Baseline Diastolic Blood Pressure (mmHg)					
	n	Mean	SE	95% CI	p-value
<65					
Placebo					
Model Estimates	12	75	2.2	[70, 79]	
Xanomeline Low Dose					
Model Estimates	5	77	3.4	[70, 83]	
Contrast / Placebo		2	4	[-5.9, 9.9]	0.613
Xanomeline High Dose					
Model Estimates	7	76	2.8	[71, 82]	
Contrast / Placebo		1.5	3.6	[-5.6, 8.5]	0.678
Contrast / Xanomeline Low Dose		-0.54	4.4	[-9.2, 8.1]	0.902
65-80					
Placebo					
Model Estimates	37	76	1.2	[73, 78]	
Xanomeline Low Dose					
Model Estimates	35	74	1.3	[71, 76]	
Contrast / Placebo		-1.7	1.8	[-5.2, 1.8]	0.327
Xanomeline High Dose					
Model Estimates	37	76	1.2	[74, 79]	
Contrast / Placebo		0.41	1.7	[-3, 3.9]	0.815
Contrast / Xanomeline Low Dose		2.1	1.8	[-1.3, 5.6]	0.227
>80					
Placebo					
Model Estimates	23	76	1.6	[73, 79]	
Xanomeline Low Dose					
Model Estimates	20	76	1.7	[73, 80]	
Contrast / Placebo		0.79	2.3	[-3.7, 5.3]	0.733
Xanomeline High Dose					
Model Estimates	12	75	2.2	[71, 80]	
Contrast / Placebo		-0.38	2.7	[-5.7, 4.9]	0.886
Contrast / Xanomeline Low Dose		-1.2	2.7	[-6.6, 4.2]	0.670
Treatment p-value: 0.946					
Subgroup: Age, p-value: 0.878					
Subgroup interaction with treatment p-value: 0.794					
Covariate p-value: <0.001					

Shiny App Structure and Examples

DevOps

phuseeuconnect2024 +

Overview
Boards
Repos
Files
Commits
Pushes
Branches
Tags
Pull requests
Advanced Security
Pipelines
Test Plans
Artifacts

master

Pushes

8/27/2024 (4 updates)

>	AS	Updated to 1f40f794: Updated renv.lock	1f40f794	Abu Sami	Tue at 3:40 PM	failed
>	AS	Updated to edf83680: Detect Error	edf83680	Abu Sami	Tue at 3:38 PM	
>	AS	Updated to fb6a07c2: Update azure-pipelines.yml for Azure Pipelines	fb6a07c2	Abu Sami	Tue at 1:36 PM	
>	AS	Created at ccc7a4af: firstcommit	ccc7a4af	Abu Sami	Tue at 1:25 PM	failed

Results of each pushes to Cloud

Shiny App Structure and Examples

DevOps

The screenshot shows a Shiny app build status page. At the top, there are four sections: 'Repository and version' (phuseeuconnect2024, master, 1f40f794), 'Time started and elapsed' (Tue at 3:40 PM, 42m 29s), 'Related' (0 work items, 4 published), and 'Tests and coverage' (75% passed, 0.96% covered). The 'Tests and coverage' section is highlighted with a red box. Below this, there are two tabs: 'Errors' (2) and 'Warnings' (4). The 'Errors' tab is highlighted with a red box. Below the tabs, there are two error messages: 'Bash exited with code '1'. Check R Package • Install and check package' and 'There are one or more test failures detected in result files. Detailed summary of published test results can be viewed in the Tests tab. Linter • Publish MegaLinter Results'. Below the error messages, there is a link 'View documentation for troubleshooting failed runs'. At the bottom, there is a table titled 'Jobs' with columns 'Name', 'Status', and 'Duration'. The table has two rows: 'Check R Package' (Failed, 37m 22s) and 'Linter' (Failed, 4m 46s). The 'Status' and 'Duration' columns are highlighted with a red box. A blue box on the right contains the text: '100% passed is required' and 'Covered should be improved'. An arrow points from the 'Tests and coverage' section to the blue box.

Repository and version
◆ phuseeuconnect2024
🔗 master 1f40f794

Time started and elapsed
📅 Tue at 3:40 PM
🕒 42m 29s

Related
🔗 0 work items
📄 4 published

Tests and coverage
🕒 75% passed
📊 0.96% covered

Errors 2 Warnings 4

✖ Bash exited with code '1'.
Check R Package • Install and check package

✖ There are one or more test failures detected in result files. Detailed summary of published test results can be viewed in the Tests tab.
Linter • Publish MegaLinter Results

[View documentation for troubleshooting failed runs](#)

Jobs

Name	Status	Duration
✖ Check R Package	Failed	🕒 37m 22s
✖ Linter	Failed	🕒 4m 46s

- 100% passed is required
- Covered should be improved

Shiny App Structure and Examples

DevOps

Windows PowerShell

```
PS C:\Users\abu.sami\Desktop\phuseeuconnect2024> git init
Reinitialized existing Git repository in C:/Users/abu.sami/Desktop/phuseeuconnect2024/.git/
PS C:\Users\abu.sami\Desktop\phuseeuconnect2024> git add .
PS C:\Users\abu.sami\Desktop\phuseeuconnect2024> git commit -m "Resolved CI Error"
PS C:\Users\abu.sami\Desktop\phuseeuconnect2024> git config http.postBuffer 524288000
PS C:\Users\abu.sami\Desktop\phuseeuconnect2024> git push -u origin1 main
```

#20240831.6 • Resolved CI Error

Cancel

Summary Code Coverage

Triggered by  Abu Sami

View change

Repository and version

◆ phuseeuconnect2024

🔗 main 🔗 c7465964

Time started and elapsed

🕒 Just now

⌚ 10s

Related

📁 0 work items

📁 0 artifacts

Tests and coverage

🔗 Get started

Jobs

Name

Status

Duration

🕒 Check R Package

Queued

🕒 Linter

Queued

Shiny App Structure and Examples

DevOps

Repository and version
phuseeuconnect2024
main 5c19393d

Time started and elapsed
Today at 6:39 AM
1h 0m 7s

Related
0 work items
4 published

Tests and coverage
100% passed
0.96% covered

Warnings 3

Task '[To be deprecated] Publish code coverage' version 1 (PublishCodeCoverageResults@1) is deprecated.
Check R Package • Initialize job

The PublishCodeCoverageResults@1 is deprecated. Users are recommended to switch to task version 2. For more details, see <https://devblogs.micros...>
Check R Package • Initialize job

New V2 version of task publishing code coverage results is available to all our customers now. We highly recommend to stop using the V1 version an...
Check R Package • Publish code coverage

Jobs

Name	Status	Duration
Check R Package	Success	55m 2s
Lint	Success	4m 23s

100% passed is achieved

Conclusion

Strengths

- Minimal changes are required in order to add/update modules (+ tests)
- Developed workflow in place for QA
- App with high interactivity

Challenges

- More unit tests needed
- Modules should be statistically reviewed
- User acceptance testing of modules' UI



Thank you. Any questions?

abu.sami@mainanalytics.de



www.mainanalytics.de

