R package development steps - main

1. Create Github Repository
2. Open RStudio project, create project, create R package with marking ☐ add Github Repo ☐ add Renv
3. Link project to Github repo with the following commands
4. In first command use SSH link of Github repo

```
git remote add origin git@github.com:berserkhmdvhb/aiinsurance.git
git add .
git commit -m "first commit"
git push --set-upstream origin main
git branch -M main
git push -u origin main
# if instead you put "server" instead of master, a "server" branch will be created and master
branch will have to pull request each time a push is made by "server"
```

1. Run renv::init()
2. Install the following packages: devtools, usethis, testthat, targets, shiny, shinycustomloader, rmarkdown
3. Remove ./NAMESPACE, ./R/Hello.R, ./man/Hello.Rd
4. Add dataset to package: First read.csv("dataset".csv)

```
dataset_name <- read.csv(dataset.csv)
usethis::use_data(dataset_name)
```

1. Write functions in ./R/ and their documentations
   Preferably don't put all functions in one file, and instead write each function in a separate .R file
2. Check and Install package for the first time

```
devtools::load_all(".")
devtools::check()
devtools::install()
```

1. Check documentation of package, and its functions with

```
help(package=mypackage)
?function
```

1. create vignette:
2. usethis::use_vignette("my-vignette")

3. If you want to add report Rmd file and any other file that don't intend to be built yet be stayed in package, put them all in ./inst/
4. If the Rmd file contain some packages that are not in the functions but yet needed to be loaded, put

them in "suggests" in ./DESCRIPTION

5. To create a pipeline, add _targets.R file to ./, add .functions.R that contain functions needed to be sourced and used in the _targets.R

```
library(targets)
targets::tar_make()
```

add /renv/
/inst/my_app/app-cache
to .gitignore

and add
^_targets.R$
^_targets$
to .Rbuildignore

1. To add shiny as part of package, create "./inst/my_app" and the files "server.R", "ui.R", "global.R"
2. Add a function for instance called shiny_run.R with the following contents

```
#' Display shiny app
#' @export
#' @return Returns fit object of glmnet function
#' @details
#' This function displays the interactive shiny app in which one can have an intuition about

shiny_run_hmd <- function() {
  appDir <- system.file("my_app", package = "mypackage")
  if (appDir == "") {
    stop("Could not find example directory. Try re-installing `aiinsurance`.", call. = FALSE)
  }
```