

Instructions to programs in R

Contents

0.1	Submission files	1
0.2	Software needed	1
0.3	Recreation of the R environment	2
0.4	Populating folders	15
0.5	Program modification	16
0.6	Program execution	18

0.1 Submission files

Novo Nordisk has used internally developed packages to conduct TFL and ADaM programming. Novo Nordisk has used the R package **renv** to manage R environments for each deliverable.

Novo Nordisk uses a package manager to serve both internal and external (CRAN) packages. As this server is not available externally, we have bundled our internal packages using the R package **pkglite** which represents packages as clear text files to pass the submission gateway. These text files needs to be unbundled and reconstructed into source R packages (.tar.gz) and stored in a common folder.

The **renv** package uses a so-called **renv.lock** file which specifies all needed packages and allows **renv** to “bootstrap” the R environment.

The files that accompany the recreation of the environments are:

- **buildrenviron.r**, which sets up a Novo Nordisk folder structure and unpacks **pkglite**-files and **renv.lock**-files.
- **pkglite-1234.txt**, which contains R packages re-packaged to a .txt file.
- **renvlock-1234.txt**, which contains an overview of all versions of packages used in this deliverable in JSON format as well as the R version.
- **activate.txt** which is the activation file for the functionality within the R package **renv**.

0.2 Software needed

Software	Download	Comment
R version 4.0.5	https://cran.r-project.org/bin/windows/base/old/4.0.5/	The trials were executed in R version 4.0.4 but the package devtools to rebuild the packages requires R version 4.0.5
Rstudio 2021.9.1.372	https://docs.posit.co/previous-versions/rstudio/#2021091	A later version of Rstudio can also be used
Rtools 4.0	https://cran.r-project.org/bin/windows/Rtools/rtools40.html	If rebuilding packages on a Windows machine Rtools is needed

To verify the R version you can execute the following:

```
R.version # This is the version this document was built under
```

```
##
## platform      x86_64-pc-linux-gnu
## arch          x86_64
## os            linux-gnu
## system        x86_64, linux-gnu
## status
## major         4
## minor         2.0
## year          2022
## month         04
## day           22
## svn rev       82229
## language      R
## version.string R version 4.2.0 (2022-04-22)
## nickname      Vigorous Calisthenics
```

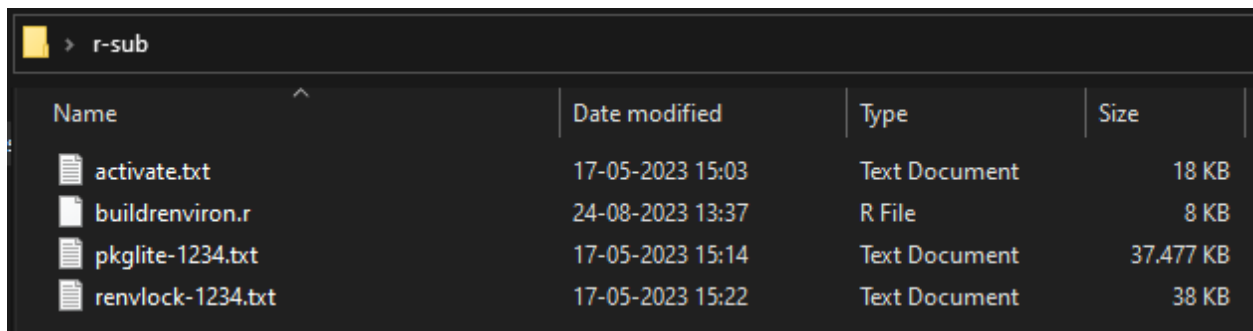
0.3 Recreation of the R environment

Please note that the `buildenvrion.r` file shown in this document is an updated version of the `buildenvrion.r` file submitted under module 5.3.5.1 in the “Analysis Program” folder under each trial. The `buildenvrion.r` file has been updated to:

1. circumvent hardcoded dependency in the package `renv` on MRAN (Microsoft R Application Network) which closed down June 2023
2. switch to install from public binaries rather than compiling from source to avoid C compiler issues related to Windows
3. upgrade/downgrade package versions in the `renv.lock` file to install from package binaries that fit the version of R used together with the Windows OS.

The updated `buildenvrion.r` file must be copied from here and replace the submitted files.

To set up the trial(s), the submission files mentioned above should be placed in a folder like so:



Name	Date modified	Type	Size
activate.txt	17-05-2023 15:03	Text Document	18 KB
buildenvrion.r	24-08-2023 13:37	R File	8 KB
pkglite-1234.txt	17-05-2023 15:14	Text Document	37.477 KB
renvlock-1234.txt	17-05-2023 15:22	Text Document	38 KB

When unpacking multiple trials only 1 `activate.txt` and 1 `buildenvrion.r` is needed.

If the needed software is installed and files copied, then open up `buildenvrion.R` in Rstudio. The first section of `buildenvrion.R` sets up all paths needed to unwrap and install code.

```
# Define necessary paths

# Define path for pkg lite files
pkglite_path <- "to-be-shared" # UPDATE TO YOUR PATH
```

```

# Define path to the unwrapped source code
pkglite_source <- "pkglite_source"

# Define path to the build tar.gz files that will be used when evaluating code
pkglite_cellar <- "pkglite_cellar"

# Define path to root renv folder structure
renv_root <- "~/renv-root"

# project_root
project_root <- pkglite_path

project <- "css123"

# Paths to public package managers with compiled binaries available
public_pkg_urls <- c("https://packagemanager.posit.co/cran/2021-06-09",
                    "https://packagemanager.posit.co/cran/2021-09-29")

```

In the following examples we will set the `pkglite_path` to a folder on the desktop ("C:/Users/aikp/Desktop/r-sub"). The top of `buildrenviro.R` should now look like this:

```

# Define necessary paths

# Define path for pkg lite files
pkglite_path <- "C:/Users/aikp/Desktop/r-sub"    # UPDATE TO YOUR PATH

# Define path to the unwrapped source code
pkglite_source <- "pkglite_source"

# Define path to the build tar.gz files that will be used when evaluating code
pkglite_cellar <- "pkglite_cellar"

# Define path to root renv folder structure
renv_root <- "~/renv-root"

# project_root
project_root <- pkglite_path

project <- "css123"

# Paths to public package managers with compiled binaries available
public_pkg_urls <- c("https://packagemanager.posit.co/cran/2021-06-09",
                    "https://packagemanager.posit.co/cran/2021-09-29")

```

No further modification of `buildrenviro.r` other than the path should be needed.

The second part of `buildrenviro.r` installs the required version of packages needed to unwrap and install source code:

```

## *****
## # install/load required packages      ----
## *****

# Install required libraries if they are not present

```

```

# Ensure installation of CRAN packages in the versions used in submission
rspm_repo = public_pkg_urls[1]
if (!requireNamespace("pkglite", quietly = TRUE))
  install.packages("pkglite", repos = rspm_repo)
if (!requireNamespace("devtools", quietly = TRUE))
  install.packages("devtools", repos = rspm_repo)
if (!requireNamespace("usethis", quietly = TRUE))
  install.packages("usethis", repos = rspm_repo)
if (!requireNamespace("jsonlite", quietly = TRUE))
  install.packages("jsonlite", repos = rspm_repo)

# Attach libraries
library(pkglite)
library(devtools)
library(usethis)
library(jsonlite)

```

The third part of the `buildrenviron.r` unpacks the R packages stored in the `pkglite-*.txt` files and builds them for installation later on.

```

## *****
## # Unpack pkglite files                                     ----
##
## The pkglite representation of the internal packages used in the different
## Parts of the submission are unpackaged and build as tar.gz files
## *****

pkglite_files <- list.files(pkglite_path, pattern = "^pkglite-.*\\.txt")
for (pkglite_file in pkglite_files) {

  # Make sure a folder exist for the unpacked source files
  cat("unpacking", pkglite_file, "\n")
  pkglite_output <- file.path(pkglite_source, tools::file_path_sans_ext(pkglite_file))
  dir.create(pkglite_output, showWarnings = FALSE, recursive = TRUE)

  # Unpack the pkglite representation
  unpack(file.path(pkglite_path, pkglite_file),
        output = pkglite_output,
        quiet = TRUE
  )

  # Get the names of the packages used
  pkgs <- list.files(pkglite_output, full.names = TRUE)
  names(pkgs) <- sapply(pkgs, basename)

  # Create R tar.gz files for installation
  lapply(file.path(pkglite_cellar, basename(pkgs)), dir.create,
        showWarnings = FALSE, recursive = TRUE)
  pkg_tar_list <-
    mapply(devtools::build, pkgs,
          vignettes = FALSE, path = file.path(pkglite_cellar, basename(pkgs)))
}

```

Upon executing the above you should see the following appear in the console showing that R CMD build is running:

```

+ # Get the names of the packages used
+ pkgs <- list.files(pkglite_output, full.names = TRUE)
+ names(pkgs) <- sapply(pkgs, basename)
+
+ # Create R tar.gz files for installation
+ lapply(file.path(pkglite_cellar, basename(pkgs)), dir.create, showWarnings = FALSE, recursive = TRUE)
+ pkg_tar_list <-
+   mapply(devtools::build, pkgs, vignettes = FALSE, path = file.path(pkglite_cellar, basename(pkgs)))
+ }
unpacking pkglite-1234.txt
-- R CMD build -----
v checking for file 'C:\Users\ai kp\OneDrive - Novo Nordisk\pkglite_source\pkglite-1234\NNaccess\DESCRIPTION' (781ms)
- preparing 'NNaccess':
v checking DESCRIPTION meta-information ...
- checking for LF line-endings in source and make files and shell scripts (397ms)
- checking for empty or unneeded directories
- building 'NNaccess_0.0.30.tar.gz'

-- R CMD build -----
v checking for file 'C:\Users\ai kp\OneDrive - Novo Nordisk\pkglite_source\pkglite-1234\NNadam\DESCRIPTION' (532ms)
- preparing 'NNadam':
v checking DESCRIPTION meta-information ...
- checking for LF line-endings in source and make files and shell scripts
- checking for empty or unneeded directories
- building 'NNadam_0.0.2.tar.gz'

-- R CMD build -----
v checking for file 'C:\Users\ai kp\OneDrive - Novo Nordisk\pkglite_source\pkglite-1234\NNbiostat\DESCRIPTION' (585ms)
- preparing 'NNbiostat':
v checking DESCRIPTION meta-information ...
- checking for LF line-endings in source and make files and shell scripts
- checking for empty or unneeded directories
- building 'NNbiostat_1.0.62.tar.gz'

```

The fourth and last part of `buildrenviron.r` sets up paths for `renv`, disables the MRAN dependency, Attaches the `Rtools` installation to the R environment and proceeds to build the folder structure and place all project files in their respective folders.

```

## *****
## # Create folders capable of bootstrapping -----
##
## Each study in the submission is run within slightly different environments.
## This part of the script creates folders capable of bootstrapping each study.
## *****

# Define the .Renviron file
renviron_content <- c(
  paste0("RENV_PATHS_ROOT=", normalizePath(renv_root), "\n"),
  paste0("RENV_PATHS_LOCAL=", normalizePath(pkglite_cellar), "\n"),
  paste0("RENV_CONFIG_MРАН_ENABLED = FALSE")
)

# Add Rtools 4.0 if on windows
if (.Platform$OS.type == "windows") {
  write('PATH="{RTOOLS40_HOME}\\usr\\bin;{PATH}"', file = "~/ .Renviron", append = TRUE)
  rtools <- 'PATH="{RTOOLS40_HOME}\\usr\\bin;{PATH}"'
} else {
  rtools <- ""
}

# Get the trial names
trials_01 <- list.files(pkglite_path, pattern = "^pkglite-.*\\.txt")
trials <- gsub(".txt$", "", gsub("^pkglite-", "", trials_01))

for (trial in trials) {

  # Create folders

```

```

new_trial_folder <- file.path(project_root, project, trial, "custom/stats/program")
dir.create(file.path(new_trial_folder, "renv"),
           showWarnings = FALSE, recursive = TRUE)
dir.create(file.path(new_trial_folder, "statprog"),
           showWarnings = FALSE, recursive = TRUE)
writeLines("Place the .R scripts in here",
           file.path(new_trial_folder, "statprog/README"))

# Create some convenience folders and helper README files
dir.create(file.path(new_trial_folder, "../data/sdtm"),
           showWarnings = FALSE, recursive = TRUE)
writeLines("Place your SDTM in this folder",
           file.path(new_trial_folder, "../data/sdtm/README"))
dir.create(file.path(new_trial_folder, "../data/adam"),
           showWarnings = FALSE, recursive = TRUE)
writeLines("Place your ADAM in this folder",
           file.path(new_trial_folder, "../data/adam/README"))
dir.create(file.path(new_trial_folder, "../data/metadata"),
           showWarnings = FALSE, recursive = TRUE)
writeLines("Place your metadata in this folder",
           file.path(new_trial_folder, "../data/metadata/README"))
dir.create(file.path(new_trial_folder, "../output/output_datasets"),
           showWarnings = FALSE, recursive = TRUE)
writeLines("The outputs will appear here",
           file.path(new_trial_folder, "../output/README"))

# Create needed files for renv
writeLines(renviron_content, file.path(new_trial_folder, ".Renviron"))
writeLines('source("renv/activate.R")', file.path(new_trial_folder, ".Rprofile"))
file.copy(file.path(pkglite_path, "activate.txt"),
          file.path(new_trial_folder, "renv", "activate.R"))
file.copy(file.path(pkglite_path, paste0("renvlock-", trial, ".txt")),
          file.path(new_trial_folder, "renv.lock"))

# Create R project file
create_project(new_trial_folder, open = FALSE)
unlink(file.path(new_trial_folder, "R"))

# Create lib calls to all R packages:
lock_contents <- readLines(file.path(new_trial_folder, "renv.lock"))
package_lines <- grep('"Package":', lock_contents, value = TRUE)
packages <- gsub('"', '?', "", gsub('.*: *\\n', "", package_lines))

writeLines(paste0("library(", packages, ")"),
           file.path(new_trial_folder, "_dependencies.R"))

# Update renv.lock to use compiled binaries from a public package manager
lock_json <- jsonlite::fromJSON(paste(lock_contents, collapse = "\n"))
lock_json$R$Repositories <- rbind(lock_json$R$Repositories, lock_json$R$Repositories)
lock_json$R$Repositories$URL <- public_pkg_urls

# On windows some packages need an updated compiled binary

```

```

if (!is.null(lock_json$Packages$Rcpp$Version))
  lock_json$Packages$Rcpp$Version <- "1.0.7"
if (!is.null(lock_json$Packages$sp$Version))
  lock_json$Packages$sp$Version <- "1.4-5"
if (!is.null(lock_json$Packages$xfun$Version))
  lock_json$Packages$xfun$Version <- "0.23"
if (!is.null(lock_json$Packages$rmarkdown$Version))
  lock_json$Packages$rmarkdown$Version <- "2.8"
lock_return <- jsonlite::toJSON(lock_json, pretty = TRUE, auto_unbox = TRUE)
writeLines(lock_return, con = file.path(new_trial_folder, "renv.lock"))
}

```

The following folder structure is being built:

```

.
|-- css123
|   |-- 1234
|       |-- custom
|           |-- stats
|               |-- data
|                   |-- adam
|                       |-- adadj.xpt
|                       |-- adeg.xpt
|                   |-- sdtm
|               |-- output
|                   |-- my_table.txt
|                   |-- my_table.xml
|                   |-- output_datasets
|                       |-- new_ds.xlsx
|                       |-- output_source
|                           |-- my_table.rds
|-- css123
|   |-- 5678
|       |-- custom
|           |-- stats
|               |-- data
|                   |-- adam
|                   |-- sdtm
|               |-- output
|                   |-- output_datasets
|                   |-- output_source

```

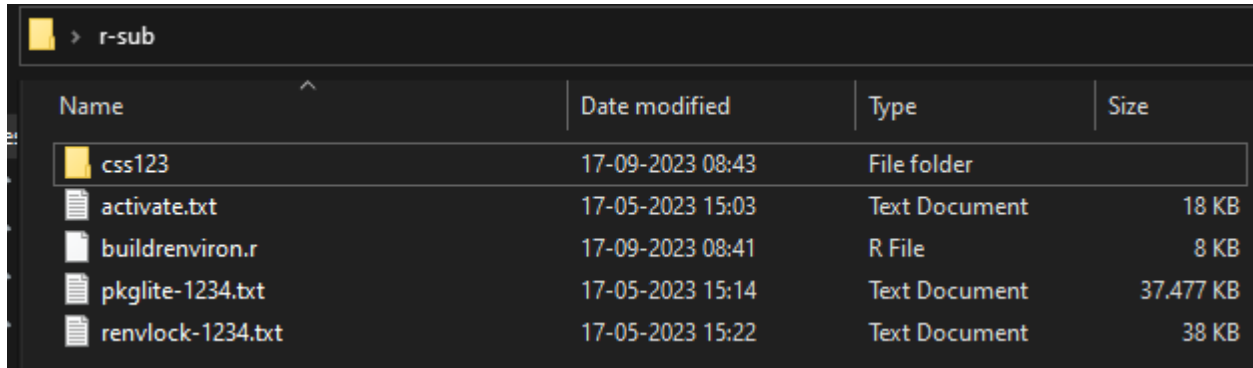
Upon executing the code above you should see the following in the console:

```

+ lock_json$Packages$rmarkdown$Version <- "2.8"
+ lock_return <- jsonlite::toJSON(lock_json, pretty = TRUE, auto_unbox = TRUE)
+ writeLines(lock_return, con = file.path(new_trial_folder, "renv.lock"))
+ }
v Setting active project to 'C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/program'
v Creating 'R/'
v Writing 'program.Rproj'
v Adding '.Rproj.user' to '.gitignore'
v Setting active project to '<no active project>'
>

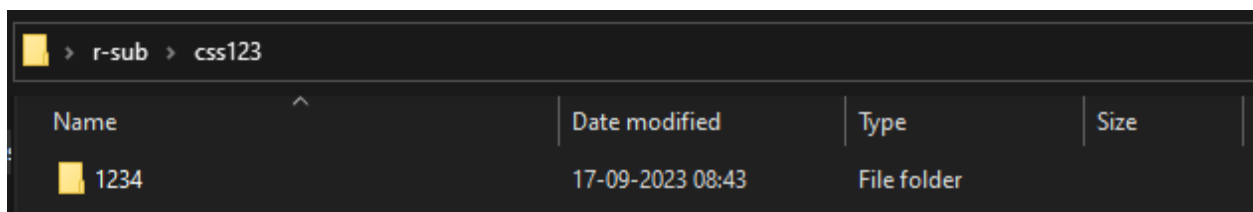
```

The folder specified in the `pkglite_path` now looks like this:



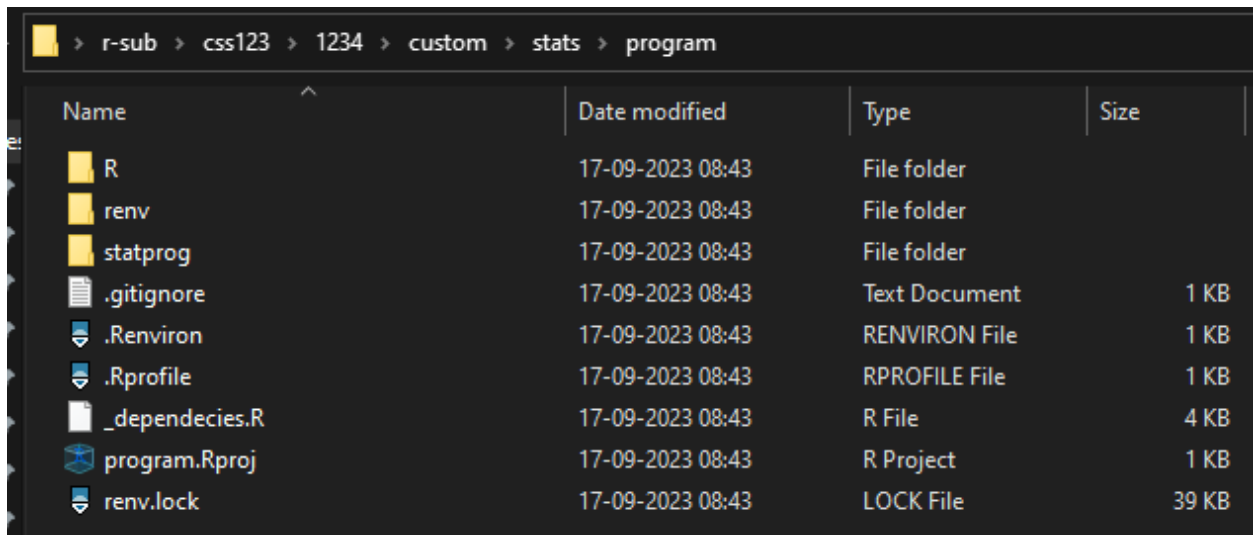
Name	Date modified	Type	Size
css123	17-09-2023 08:43	File folder	
activate.txt	17-05-2023 15:03	Text Document	18 KB
buildrenviron.r	17-09-2023 08:41	R File	8 KB
pkglite-1234.txt	17-05-2023 15:14	Text Document	37.477 KB
renvlock-1234.txt	17-05-2023 15:22	Text Document	38 KB

with the following subfolders - one for each trial:



Name	Date modified	Type	Size
1234	17-09-2023 08:43	File folder	

Open the program folder e.g. `css123/1234/custom/stats/program` and check that the project files are there:



Name	Date modified	Type	Size
R	17-09-2023 08:43	File folder	
renv	17-09-2023 08:43	File folder	
statprog	17-09-2023 08:43	File folder	
.gitignore	17-09-2023 08:43	Text Document	1 KB
.Renviron	17-09-2023 08:43	RENVIRON File	1 KB
.Rprofile	17-09-2023 08:43	RPROFILE File	1 KB
_dependencies.R	17-09-2023 08:43	R File	4 KB
program.Rproj	17-09-2023 08:43	R Project	1 KB
renv.lock	17-09-2023 08:43	LOCK File	39 KB

To further verify the setup you can check that:

1. The project `.Renviron` file contains something like:

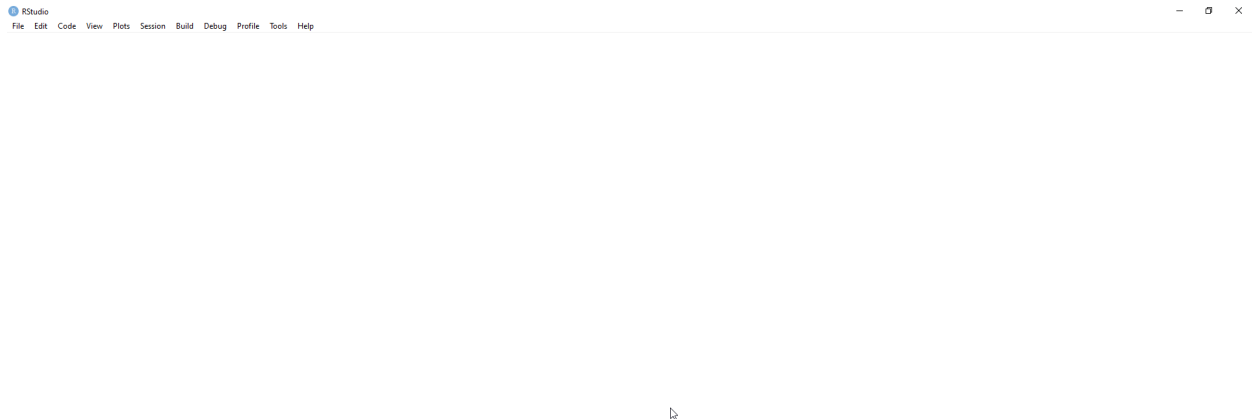
```
RENV_PATHS_ROOT='C:\Users\aikp\OneDrive - Novo Nordisk\renv-root'
RENV_PATHS_LOCAL='C:\Users\aikp\OneDrive - Novo Nordisk\pkglite_cellar'
RENV_CONFIG_MRAN_ENABLED = FALSE
```

2. the file `renv/activate.R` is present and The project `.Rprofile` file contains:

```
source("renv/activate.R")
```

3. Verify `renv.lock` exists.

To restore the environment first open the `program.Rproj` file. This should open up Rstudio. As `renv` is bootstrapping itself you might see this screen (blank screen) for ~10 seconds:



Followed by:

A screenshot of the R console window. The title bar shows 'Console', 'Terminal', and 'Jobs'. The console output shows the R version 4.0.5 startup message, followed by the renv 0.13.2 installation and loading process. The output is as follows:

```
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

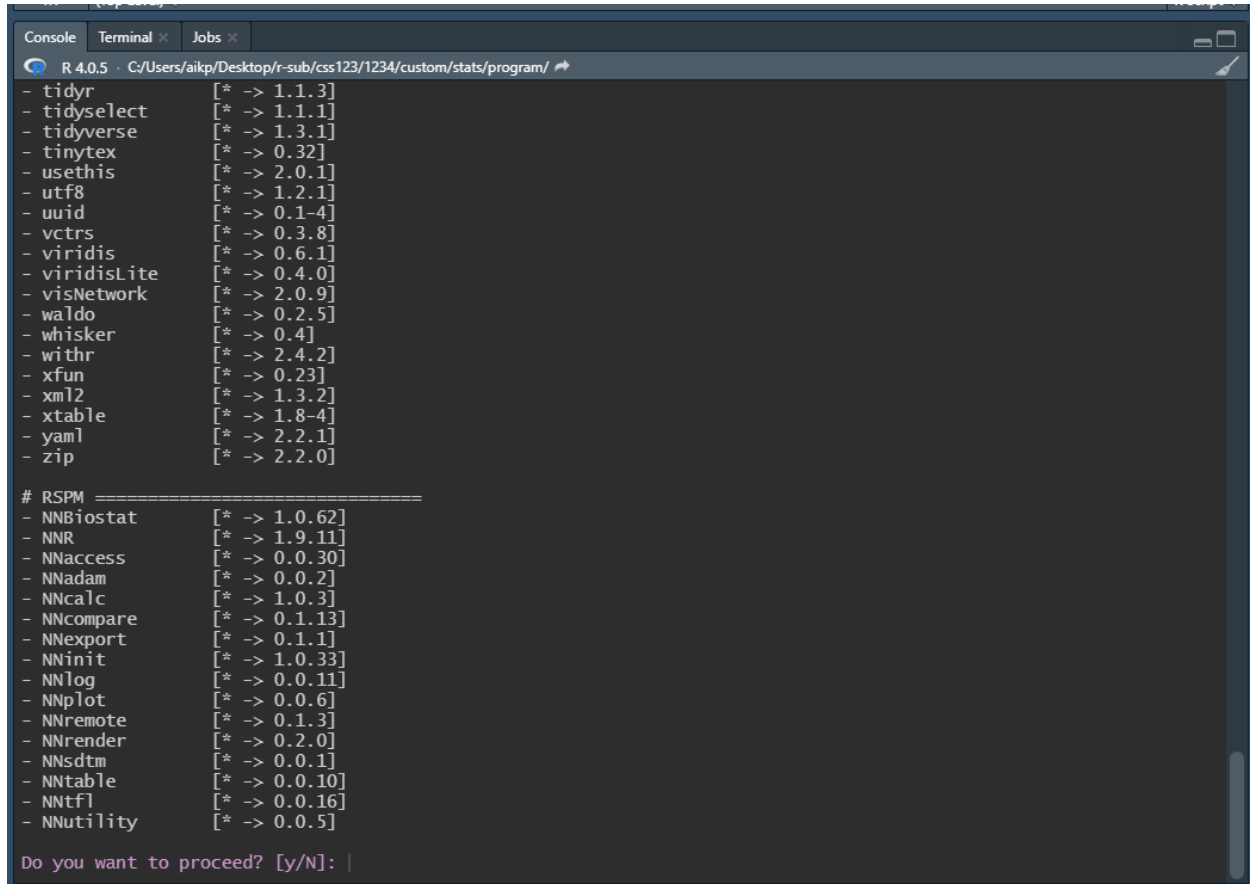
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

# Bootstrapping renv 0.13.2 -----
* Downloading renv 0.13.2 ... OK
* Installing renv 0.13.2 ... Done!
* Successfully installed and loaded renv 0.13.2.
* Project 'C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/program' loaded. [renv 0.13.2]
* The project library is out of sync with the lockfile.
* Use `renv::restore()` to install packages recorded in the lockfile.
> |
```

To restore the environment run `renv::restore()` in the console. Upon executing the command you should see the following in the console:



```

R 4.0.5 - C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/program/
- tidy [* -> 1.1.3]
- tidyselect [* -> 1.1.1]
- tidyverse [* -> 1.3.1]
- tinytex [* -> 0.32]
- usethis [* -> 2.0.1]
- utf8 [* -> 1.2.1]
- uuid [* -> 0.1-4]
- vctrs [* -> 0.3.8]
- viridis [* -> 0.6.1]
- viridisLite [* -> 0.4.0]
- visNetwork [* -> 2.0.9]
- waldo [* -> 0.2.5]
- whisker [* -> 0.4]
- withr [* -> 2.4.2]
- xfun [* -> 0.23]
- xml2 [* -> 1.3.2]
- xtable [* -> 1.8-4]
- yaml [* -> 2.2.1]
- zip [* -> 2.2.0]

# RSPM =====
- NNbiostat [* -> 1.0.62]
- NNR [* -> 1.9.11]
- NNaccess [* -> 0.0.30]
- NNadam [* -> 0.0.2]
- NNcalc [* -> 1.0.3]
- NNcompare [* -> 0.1.13]
- NNexport [* -> 0.1.1]
- NNinit [* -> 1.0.33]
- NNlog [* -> 0.0.11]
- NNplot [* -> 0.0.6]
- NNremote [* -> 0.1.3]
- NNrender [* -> 0.2.0]
- NNsdm [* -> 0.0.1]
- NNtable [* -> 0.0.10]
- NNtf1 [* -> 0.0.16]
- NNutility [* -> 0.0.5]

Do you want to proceed? [y/N]: |

```

Confirm with “y” and depending on whether you have any of the packages available you should see that R either start querying the package manager or using a cache. At the end of the command you should see something like this:



```

OK [linked cache]
Installing packrat [0.6.0] ...
OK [linked cache]
Installing parsedate [1.2.1] ...
OK [linked cache]
Installing reprex [2.0.0] ...
OK [linked cache]
Installing rlist [0.4.6.1] ...
OK [linked cache]
Installing rsconnect [0.8.18] ...
OK [linked cache]
Installing rsvg [2.1.2] ...
OK [linked cache]
Installing selectr [0.4-2] ...
OK [linked cache]
Installing rvest [1.0.0] ...
OK [linked cache]
Installing tidyverse [1.3.1] ...
OK [linked cache]
> |

```

To confirm that the environment is correctly set up execute `library(NNbiostat)` in the console. A correct environment will load the package and display:

```

R 4.0.5 - C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/program/
Installing rvest [1.0.0] ...
  OK [linked cache]
Installing tidyverse [1.3.1] ...
  OK [linked cache]
> library(NNbiostat)
-- Attaching packages ----- NNbiostat 1.0.62 --
v labelled 2.8.0      v ggplot2 3.3.3
v emmeans 1.6.1      v patchwork 1.1.1
v lme4 1.1.27       v tibble 3.1.2
v MASS 7.3.54       v tidyr 1.1.3
v Matrix 1.3.4      v readr 1.4.0
v survival 3.2.11   v purrr 0.3.4
v mice 3.13.0       v dplyr 1.0.6
v glue 1.4.2        v stringr 1.4.0
v data.table 1.14.0 v forcats 0.5.1

-- Attaching packages ----- NNR 1.9.11 --
v NNlog 0.0.11      v NNexport 0.1.1
v NNremote 0.1.3    v NNtfl 0.0.16
v NNaccess 0.0.30   v NNsdm 0.0.1
v NNcalc 1.0.3      v NNadam 0.0.2
v NNplot 0.0.6      v NNrender 0.2.0
v NNtable 0.0.10    v NNutility 0.0.5

-- Conflicts ----- NNbiostat_conflicts() --
x NNremote::%%() masks purrr::%%()
x patchwork::area() masks MASS::area()
x dplyr::between() masks data.table::between()
x methods::body<-() masks base::body<-()
x mice::cbind() masks base::cbind()
x dplyr::collapse() masks glue::collapse()
x tidyr::expand() masks Matrix::expand()
x dplyr::filter() masks mice::filter(), stats::filter()
x dplyr::first() masks data.table::first()
x methods::kronecker() masks base::kronecker()
x dplyr::lag() masks stats::lag()
x dplyr::last() masks data.table::last()
x tidyr::pack() masks Matrix::pack()
x mice::rbind() masks base::rbind()
x dplyr::select() masks MASS::select()
x purrr::transpose() masks data.table::transpose()
x tidyr::unpack() masks Matrix::unpack()
x Matrix::update() masks stats::update()
> |

```

0.3.1 Updated buildrenviron.r

Please note that the submitted buildrenviron.r files should be replaced with the latest version below.

```

## *****
## # Description:
##
## Build study containers and setup renv in each study
## Unpacks the internally developed in Novo Nordisk packages and prepares
## for download of externally needed packages.
##
## Programmer: sffl+aikp+abiu
## *****

# Define necessary paths

# Define path for pkg lite files
pkglite_path <- "to-be-shared" # UPDATE TO YOUR PATH

# Define path to the unwrapped source code
pkglite_source <- "pkglite_source"

```

```

# Define path to the build tar.gz files that will be used when evaluating code
pkglite_cellar <- "pkglite_cellar"

# Define path to root renv folder structure
renv_root <- "~/renv-root"

# project_root
project_root <- pkglite_path

project <- "css123"

# Paths to public package managers with compiled binaries available
public_pkg_urls <- c("https://packagemanager.posit.co/cran/2021-06-09",
                    "https://packagemanager.posit.co/cran/2021-09-29")

## *****
## # install/load required packages -----
## *****

# Install required libraries if they are not present

# Ensure installation of CRAN packages in the versions used in submission
rspm_repo = public_pkg_urls[1]
if (!requireNamespace("pkglite", quietly = TRUE))
  install.packages("pkglite", repos = rspm_repo)
if (!requireNamespace("devtools", quietly = TRUE))
  install.packages("devtools", repos = rspm_repo)
if (!requireNamespace("usethis", quietly = TRUE))
  install.packages("usethis", repos = rspm_repo)
if (!requireNamespace("jsonlite", quietly = TRUE))
  install.packages("jsonlite", repos = rspm_repo)

# Attach libraries
library(pkglite)
library(devtools)
library(usethis)
library(jsonlite)

## *****
## # Unpack pkglite files -----
##
## The pkglite representation of the internal packages used in the different
## Parts of the submission are unpackaged and build as tar.gz files
## *****

pkglite_files <- list.files(pkglite_path, pattern = "^pkglite-.*\\.txt")
for (pkglite_file in pkglite_files) {

  # Make sure a folder exist for the unpacked source files
  cat("unpacking", pkglite_file, "\n")
  pkglite_output <- file.path(pkglite_source, tools::file_path_sans_ext(pkglite_file))
  dir.create(pkglite_output, showWarnings = FALSE, recursive = TRUE)

```

```

# Unpack the pkglite representation
unpack(file.path(pkglite_path, pkglite_file),
       output = pkglite_output,
       quiet = TRUE
)

# Get the names of the packages used
pkgs <- list.files(pkglite_output, full.names = TRUE)
names(pkgs) <- sapply(pkgs, basename)

# Create R tar.gz files for installation
lapply(file.path(pkglite_cellar, basename(pkgs)), dir.create,
       showWarnings = FALSE, recursive = TRUE)
pkg_tar_list <-
  mapply(devtools::build, pkgs, vignettes = FALSE,
        path = file.path(pkglite_cellar, basename(pkgs)))
}

## *****
## # Create folders capable of bootstrapping          ----
##
## Each study in the submission is run within slightly different environments.
## This part of the script creates folders capable of bootstrapping each study.
## *****

# Define the .Renviron file
renviron_content <- c(
  paste0("RENV_PATHS_ROOT=", normalizePath(renv_root),""),
  paste0("RENV_PATHS_LOCAL=", normalizePath(pkglite_cellar),""),
  paste0("RENV_CONFIG_MRAN_ENABLED = FALSE")
)

# Add Rtools 4.0 if on windows
if (.Platform$OS.type == "windows") {
  write('PATH="%{RTOOLS40_HOME}\\usr\\bin;%{PATH}"', file = "~/Renviron", append = TRUE)
  rtools <- 'PATH="%{RTOOLS40_HOME}\\usr\\bin;%{PATH}"'
} else {
  rtools <- ""
}

# Get the trial names
trials_01 <- list.files(pkglite_path, pattern = "^pkglite-.*\\.txt")
trials <- gsub(".txt$", "", gsub("^pkglite-", "", trials_01))

for (trial in trials) {

  # Create folders
  new_trial_folder <- file.path(project_root, project, trial, "custom/stats/program")
  dir.create(file.path(new_trial_folder, "renv"),
            showWarnings = FALSE, recursive = TRUE)
  dir.create(file.path(new_trial_folder, "statprog"),
            showWarnings = FALSE, recursive = TRUE)
  writeLines("Place the .R scripts in here",

```

```

    file.path(new_trial_folder, "statprog/README"))

# Create some convenience folders and helper README files
dir.create(file.path(new_trial_folder, "../data/sdtm"),
            showWarnings = FALSE, recursive = TRUE)
writeLines("Place your SDTM in this folder",
            file.path(new_trial_folder, "../data/sdtm/README"))
dir.create(file.path(new_trial_folder, "../data/adam"),
            showWarnings = FALSE, recursive = TRUE)
writeLines("Place your ADAM in this folder",
            file.path(new_trial_folder, "../data/adam/README"))
dir.create(file.path(new_trial_folder, "../data/metadata"),
            showWarnings = FALSE, recursive = TRUE)
writeLines("Place your metadata in this folder",
            file.path(new_trial_folder, "../data/metadata/README"))
dir.create(file.path(new_trial_folder, "../output/output_datasets"),
            showWarnings = FALSE, recursive = TRUE)
writeLines("The outputs will appear here",
            file.path(new_trial_folder, "../output/README"))

# Create needed files for renv
writeLines(renvcontent, file.path(new_trial_folder, ".Renviron"))
writeLines('source("renv/activate.R")', file.path(new_trial_folder, ".Rprofile"))
file.copy(file.path(pkglite_path, "activate.txt"),
            file.path(new_trial_folder, "renv", "activate.R"))
file.copy(file.path(pkglite_path, paste0("renvlock-", trial, ".txt")),
            file.path(new_trial_folder, "renv.lock"))

# Create R project file
create_project(new_trial_folder, open = FALSE)
unlink(file.path(new_trial_folder, "R"))

# Create lib calls to all R packages:
lock_contents <- readLines(file.path(new_trial_folder, "renv.lock"))
package_lines <- grep('"Package":', lock_contents, value = TRUE)
packages <- gsub('"', '?', "", gsub('.*: *\\n', "", package_lines))

writeLines(paste0("library(", packages, ")"),
            file.path(new_trial_folder, "_dependencies.R"))

# Update renv.lock to use compiled binaries from a public package manager
lock_json <- jsonlite::fromJSON(paste(lock_contents, collapse = "\n"))
lock_json$R$Repositories <- rbind(lock_json$R$Repositories, lock_json$R$Repositories)
lock_json$R$Repositories$URL <- public_pkg_urls

# On windows some packages need an updated compiled binary
if (!is.null(lock_json$Packages$Rcpp$Version))
  lock_json$Packages$Rcpp$Version <- "1.0.7"
if (!is.null(lock_json$Packages$sp$Version))
  lock_json$Packages$sp$Version <- "1.4-5"
if (!is.null(lock_json$Packages$xfun$Version))
  lock_json$Packages$xfun$Version <- "0.23"

```

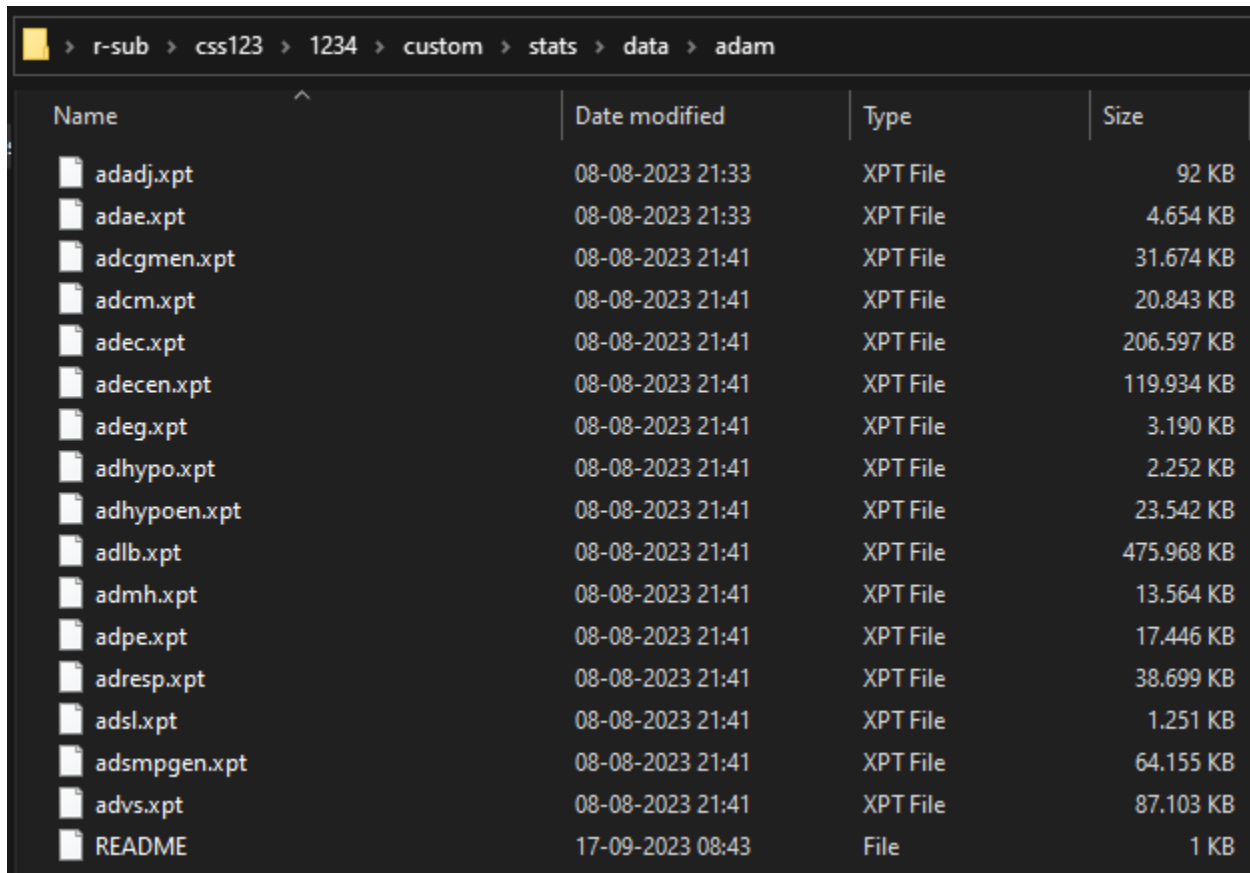
```

if (!is.null(lock_json$Packages$rmarkdown$Version))
  lock_json$Packages$rmarkdown$Version <- "2.8"
lock_return <- jsonlite::toJSON(lock_json, pretty = TRUE, auto_unbox = TRUE)
writeLines(lock_return, con = file.path(new_trial_folder, "renv.lock"))
}

```

0.4 Populating folders

Once the environment is set up data can be copied to the `stats/data` folder e.g. ADaM data in `css123/1234/custom/stats/data/adam` like so:



The screenshot shows a file explorer window with the breadcrumb path: `r-sub > css123 > 1234 > custom > stats > data > adam`. Below the path is a table listing the files and folders in the `adam` directory.

Name	Date modified	Type	Size
adadj.xpt	08-08-2023 21:33	XPT File	92 KB
adae.xpt	08-08-2023 21:33	XPT File	4.654 KB
adcgmen.xpt	08-08-2023 21:41	XPT File	31.674 KB
adcm.xpt	08-08-2023 21:41	XPT File	20.843 KB
adec.xpt	08-08-2023 21:41	XPT File	206.597 KB
adecen.xpt	08-08-2023 21:41	XPT File	119.934 KB
adeg.xpt	08-08-2023 21:41	XPT File	3.190 KB
adhypo.xpt	08-08-2023 21:41	XPT File	2.252 KB
adhypoen.xpt	08-08-2023 21:41	XPT File	23.542 KB
adlb.xpt	08-08-2023 21:41	XPT File	475.968 KB
admh.xpt	08-08-2023 21:41	XPT File	13.564 KB
adpe.xpt	08-08-2023 21:41	XPT File	17.446 KB
adresp.xpt	08-08-2023 21:41	XPT File	38.699 KB
adsl.xpt	08-08-2023 21:41	XPT File	1.251 KB
adsmpgen.xpt	08-08-2023 21:41	XPT File	64.155 KB
advs.xpt	08-08-2023 21:41	XPT File	87.103 KB
README	17-09-2023 08:43	File	1 KB

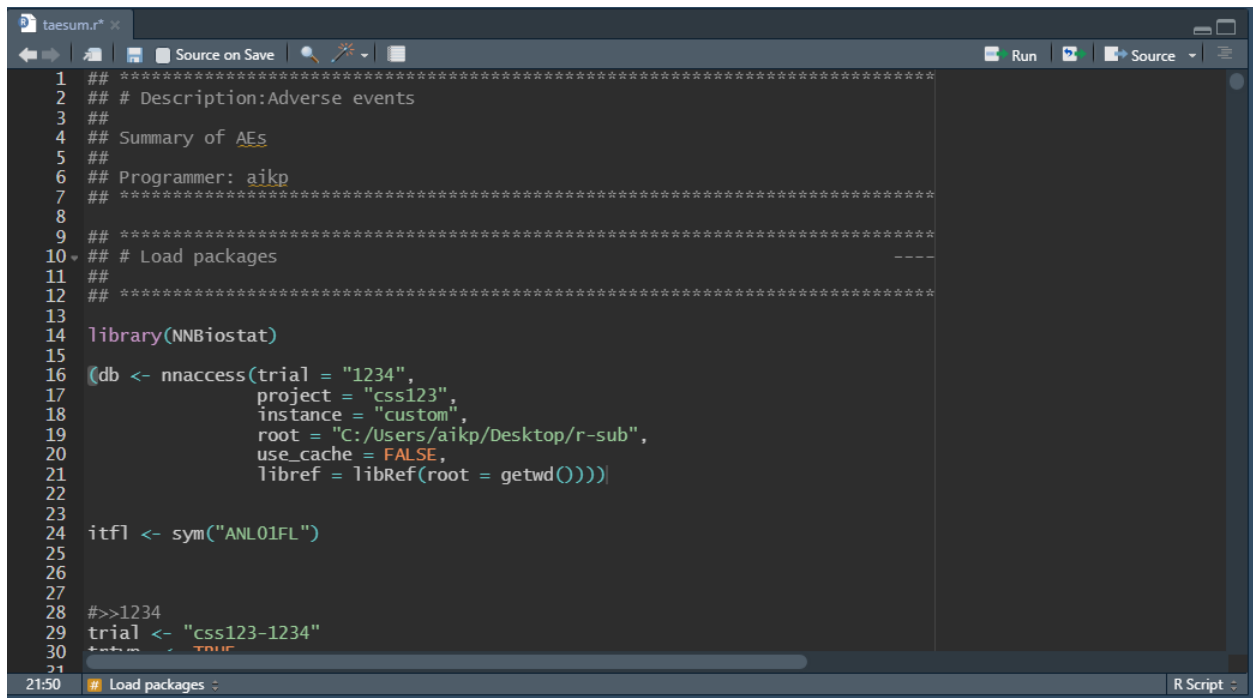
R programs for tables, listings and figures should be copied to the `stats/program/statprog` folder e.g. `css123/1234/custom/stats/program/statprog`. Note that if the programs copied into the instance have the `.txt` extension then this should be changed back to `.r/.R` in order for Rstudio to process it as a script.


```

project = "css123",
instance = "custom",
root = "C:/Users/aikp/Desktop/r-sub",
use_cache = FALSE,
libref = libRef(root = getwd()))

```

Such that program now looks like this:

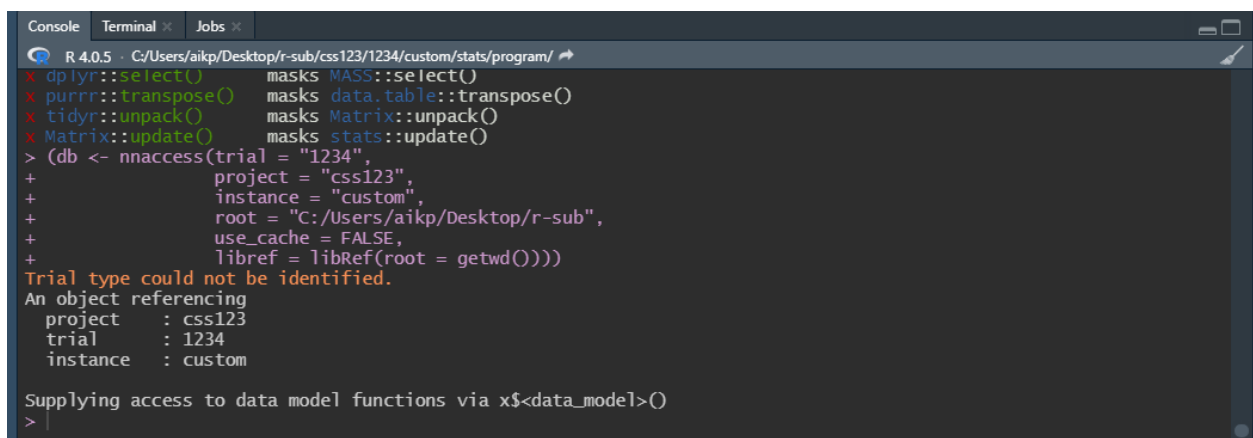


```

1 ## *****
2 ## # Description: Adverse events
3 ##
4 ## Summary of AEs
5 ##
6 ## Programmer: aikp
7 ## *****
8
9 ## *****
10 ## # Load packages
11 ##
12 ## *****
13
14 library(NNbiostat)
15
16 (db <- nnaccess(trial = "1234",
17                project = "css123",
18                instance = "custom",
19                root = "C:/Users/aikp/Desktop/r-sub",
20                use_cache = FALSE,
21                libref = libRef(root = getwd()))
22
23
24 itfl <- sym("ANL01FL")
25
26
27
28 #>>1234
29 trial <- "css123-1234"
30
31
21:50 # Load packages :

```

Upon running the code you will see the following in the console:



```

R 4.0.5 - C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/program/
x dplyr::select() masks MASS::select()
x purrr::transpose() masks data.table::transpose()
x tidyr::unpack() masks Matrix::unpack()
x Matrix::update() masks stats::update()
> (db <- nnaccess(trial = "1234",
+                project = "css123",
+                instance = "custom",
+                root = "C:/Users/aikp/Desktop/r-sub",
+                use_cache = FALSE,
+                libref = libRef(root = getwd()))
Trial type could not be identified.
An object referencing
 project : css123
 trial   : 1234
 instance : custom

Supplying access to data model functions via x$<data_model>()
>

```

The `NNaccess::nnaccess()` function sets up a consistent input/output object within a trial with agnostic file readers/writers, meaning that the following:

```
ads1 <- db$adam("ads1")
```

will always return the ADSL dataset regardless of the format (`rds`, `sas7bdat`, `xpt`, `csv`, `xlsx`) it is stored in within `stats/data/adam`.

0.6 Program execution

Once the `nnaccess()` function has been modified almost all code in the program will run.

Running the `aesum()` function within `taesum.R` and executing the first output you should see the following in the console:

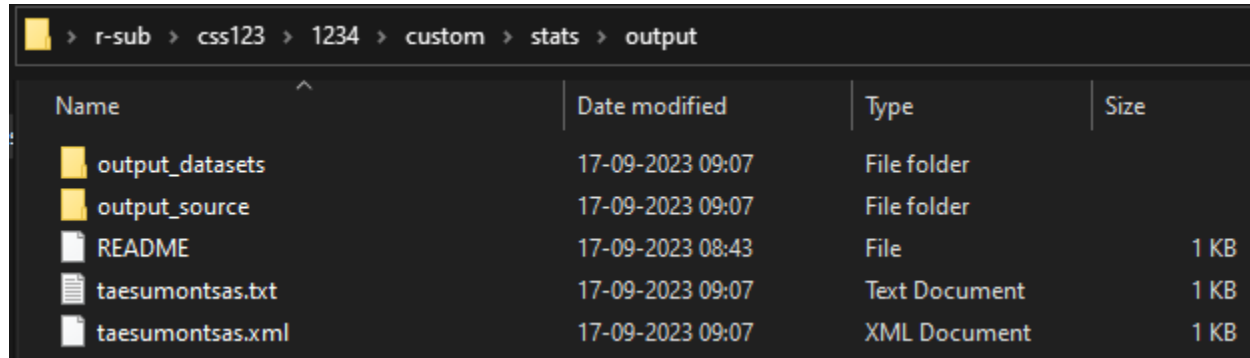
```

483
484 .NNTable <- final3 %>%
485   mutate(COL1 = str_to_sentence(COL1),
486          COL2 = ifelse(COL2 == "NA", "NA", str_to_sentence(COL2))) %>%
487   NNTable("TRTA", "COL1", "COL2", "N", "(%)", "(P)", "E", "R") %>%
488   addTransWide("TRTA" = c("N", "(%)", "E", "R"),
489               .remove_empty_columns = FALSE) %>%
490   addUnderScore() %>%
491   addGroupedColumns("COL1", "COL2", add_blank_row = FALSE) %>%
492   addExposure(tot2, format_alone = TRUE) %>%
493   addExposure(exposure = tot2,
494               format_alone = TRUE,
495               format_data = my_formats) %>%
496   addFormat(format_data = c(N = "%.0f", P = "%.1f", R = "%.2f")) %>%
497   addOrder(ORD = 1, SUBORD = 1) %>%
498   addTruncation(COL1 = 35)
499
500 footnoteEnv("meddrav", unique(adae_f$MEDDRAV))
501
502 ## *****
503 ## # Export -----
504 ##
505 ## *****
506 ## *****
507
508 # Export output -----
509 db$exportOutput(x = .NNTable, Name = outputname)
510 # Save data
511 db$output_datasets(outputname, .NNTable$data, ext = "rds")
512 # Save data as excel file
513 db$output_datasets(outputname, .NNTable$data, ext = "xlsx")
514
515 }
516
517
530:1 Define summary function
R Script

Console Terminal Jobs
R 4.0.5 C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/program/
> # Adverse events - on-treatment - summary - safety analysis set
> aesum(
+   outputname = "taesumontsas",
+   popfilter = SAFFL == "Y",
+   infilter = (!otfl) == "Y",
+   expvar = (!dury),
+   trtyp = trtyp_,
+   expontrt = TRUE
+ )
C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/output/output_datasets/taesumontsas.rds written to disk.
C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/output/output_datasets/taesumontsas.xlsx written to disk.
>

```

In the `stats/output` folder the following should now have appeared:



Name	Date modified	Type	Size
output_datasets	17-09-2023 09:07	File folder	
output_source	17-09-2023 09:07	File folder	
README	17-09-2023 08:43	File	1 KB
taesumontsas.txt	17-09-2023 09:07	Text Document	1 KB
taesumontsas.xml	17-09-2023 09:07	XML Document	1 KB

The `taesumontsas.txt` contains the table produced by the `taesum.R` program.

Due to the missing access to Novo Nordisk output metadata the table appears without Title and footnotes. Only the technical footnote can be seen. The name of the program that created the output is shown here:

```
css123/1234/custom
17SEP2023:09:07:14 - taesum/taesumontsas.txt
```

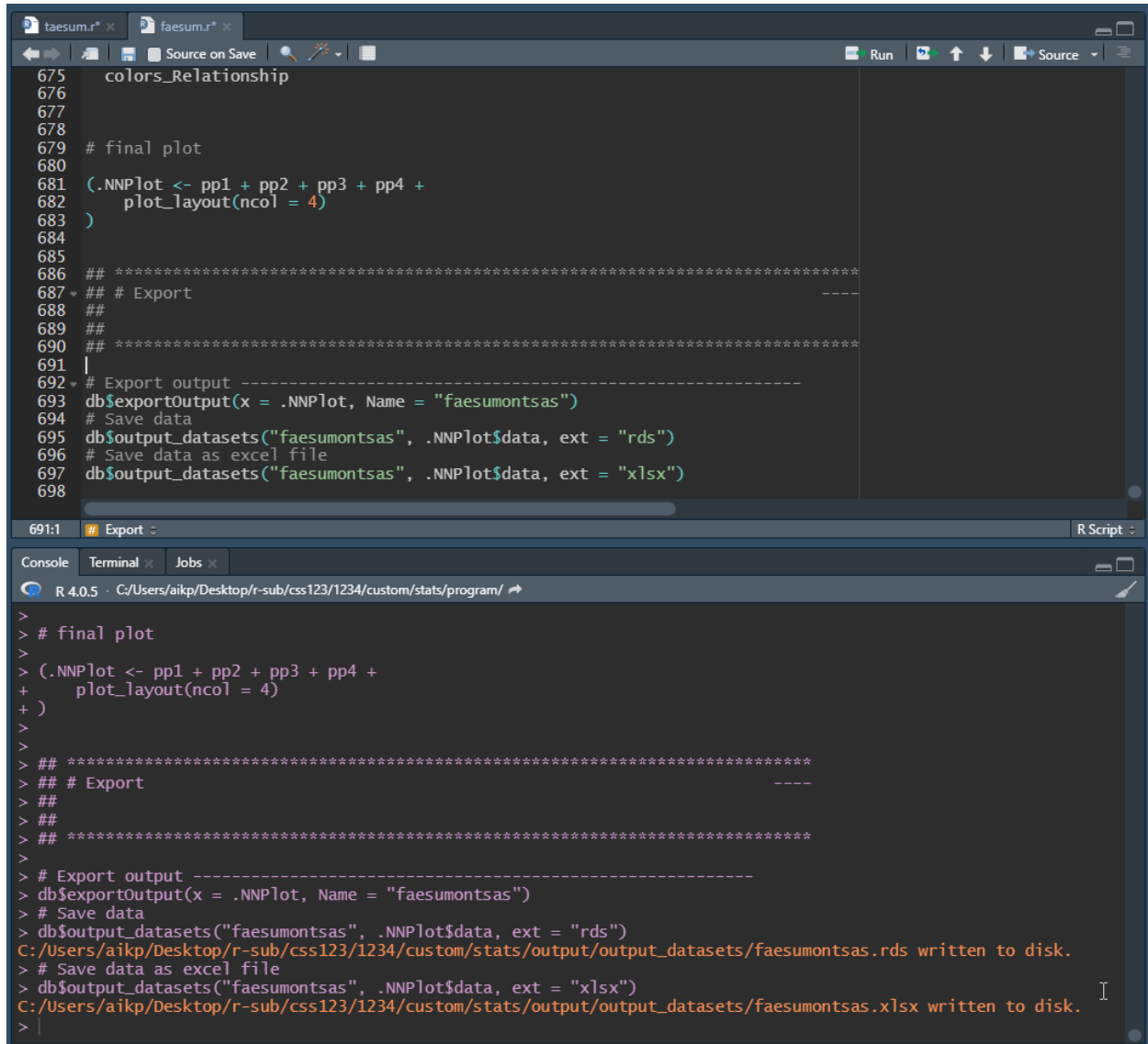
All outputs have their underlying data saved to the `stats/output/output_datasets` and `stats/output/output_source` folders in `.xlsx` and `.rds` format.

Applying the same `nnaccess()` update to the `stats/program/statprog/faesum.R` script:

```
library(NNBiostat)

(db <- nnaccess(trial = "1234",
               project = "css123",
               instance = "custom",
               root = "C:/Users/aikp/Desktop/r-sub",
               use_cache = FALSE,
               libref = libRef(root = getwd())))
```

and executing it, you will see the following:



```

675 colors_Relationship
676
677
678
679 # final plot
680
681 (.NNPlot <- pp1 + pp2 + pp3 + pp4 +
682   plot_layout(ncol = 4)
683 )
684
685
686 ## *****
687 ## # Export
688 ##
689 ##
690 ## *****
691 |
692 # Export output -----
693 db$exportOutput(x = .NNPlot, Name = "faesumontsas")
694 # Save data
695 db$output_datasets("faesumontsas", .NNPlot$data, ext = "rds")
696 # Save data as excel file
697 db$output_datasets("faesumontsas", .NNPlot$data, ext = "xlsx")
698

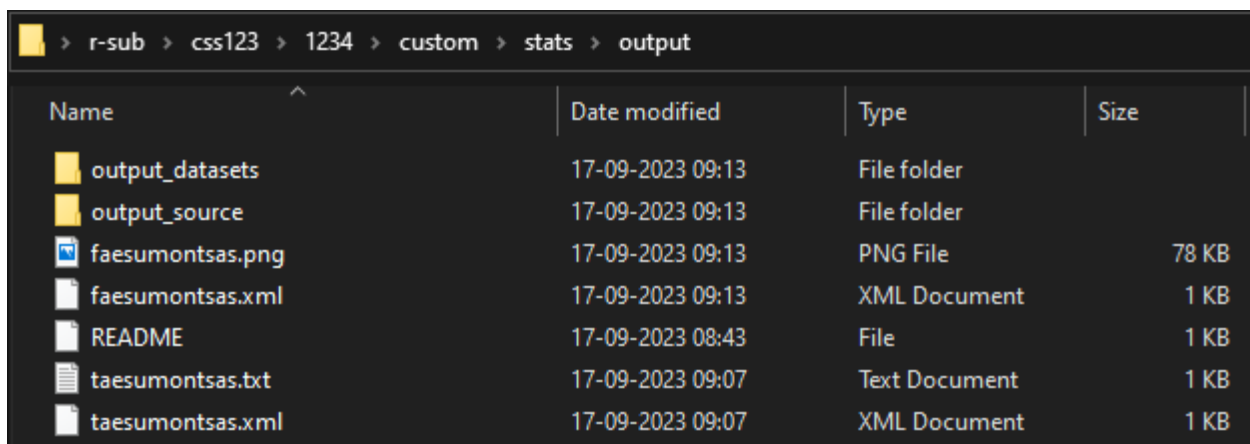
```

```

> # final plot
>
> (.NNPlot <- pp1 + pp2 + pp3 + pp4 +
+   plot_layout(ncol = 4)
+ )
>
>
> ## *****
> ## # Export
> ##
> ##
> ## *****
>
> # Export output -----
> db$exportOutput(x = .NNPlot, Name = "faesumontsas")
> # Save data
> db$output_datasets("faesumontsas", .NNPlot$data, ext = "rds")
C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/output/output_datasets/faesumontsas.rds written to disk.
> # Save data as excel file
> db$output_datasets("faesumontsas", .NNPlot$data, ext = "xlsx")
C:/Users/aikp/Desktop/r-sub/css123/1234/custom/stats/output/output_datasets/faesumontsas.xlsx written to disk.
>

```

In the stats/output folder a .png file will appear containing the plot:



Name	Date modified	Type	Size
output_datasets	17-09-2023 09:13	File folder	
output_source	17-09-2023 09:13	File folder	
faesumontsas.png	17-09-2023 09:13	PNG File	78 KB
faesumontsas.xml	17-09-2023 09:13	XML Document	1 KB
README	17-09-2023 08:43	File	1 KB
taesumontsas.txt	17-09-2023 09:07	Text Document	1 KB
taesumontsas.xml	17-09-2023 09:07	XML Document	1 KB