Taking R and shiny to production

Gao Maribe, PhD

Data Scientist at dotmodus

From Prototyping to Production

Gao Maribe, PhD

Data Scientist at dotmodus

Building a simple app and deploying it on prem

Gao Maribe, PhD

Data Scientist at dotmodus

Running a really simple shinyapp on a linux machine

Gao Maribe, PhD

Data Scientist at dotmodus

How to take over the world with R

Gao Maribe, PhD

Data Scientist at dotmodus

Demo

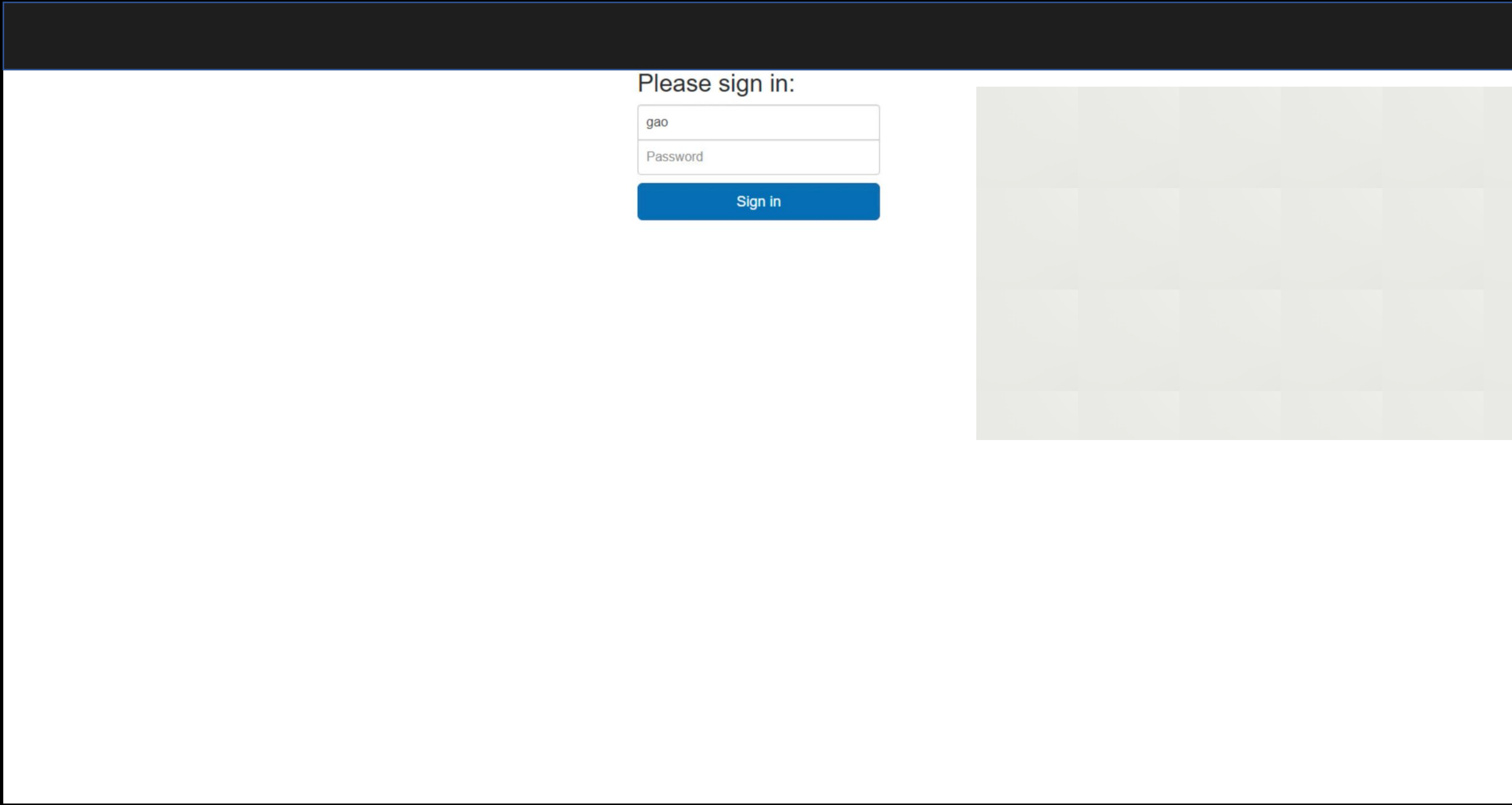
Problem statement

- Combine two data sets
  1. Attendance data
  2. Student Performance data
  3. Investigate impact using descriptives, t-test, F-test, correlation coefficient.

- **My Problem**:

How will they use this workflow when I no longer work there?

Tools

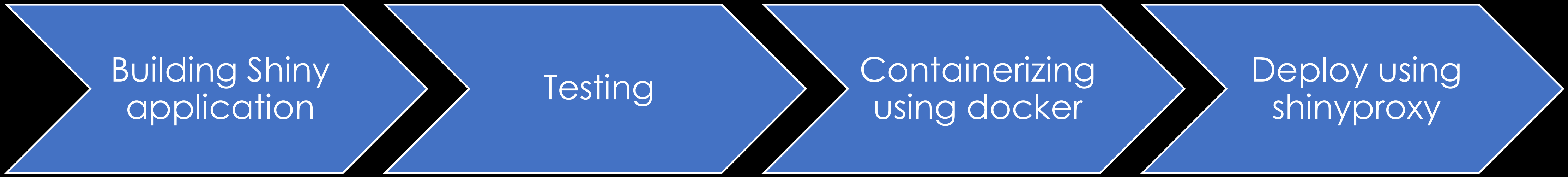


Please sign in:

gao

Password

Project structure



Building Shiny application

Testing

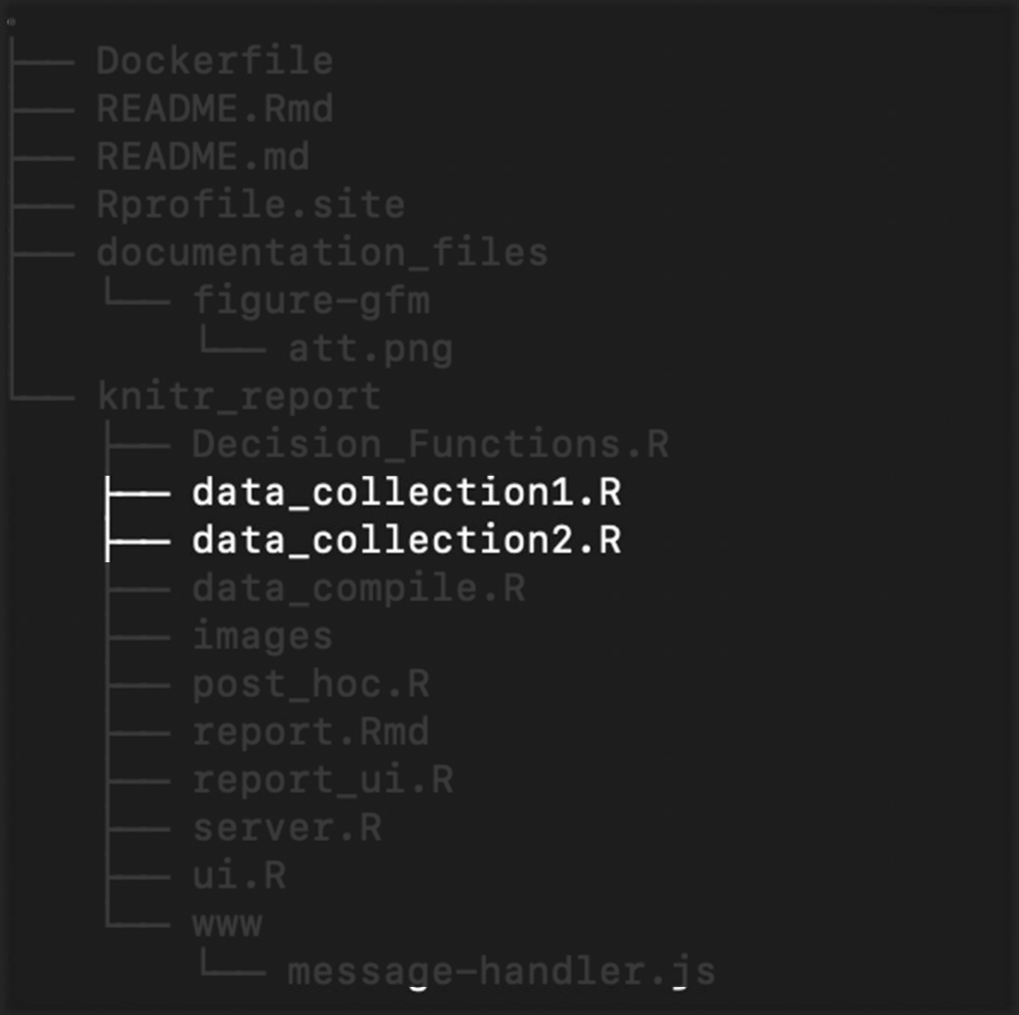
Containerizing using docker

Deploy using shinyproxy

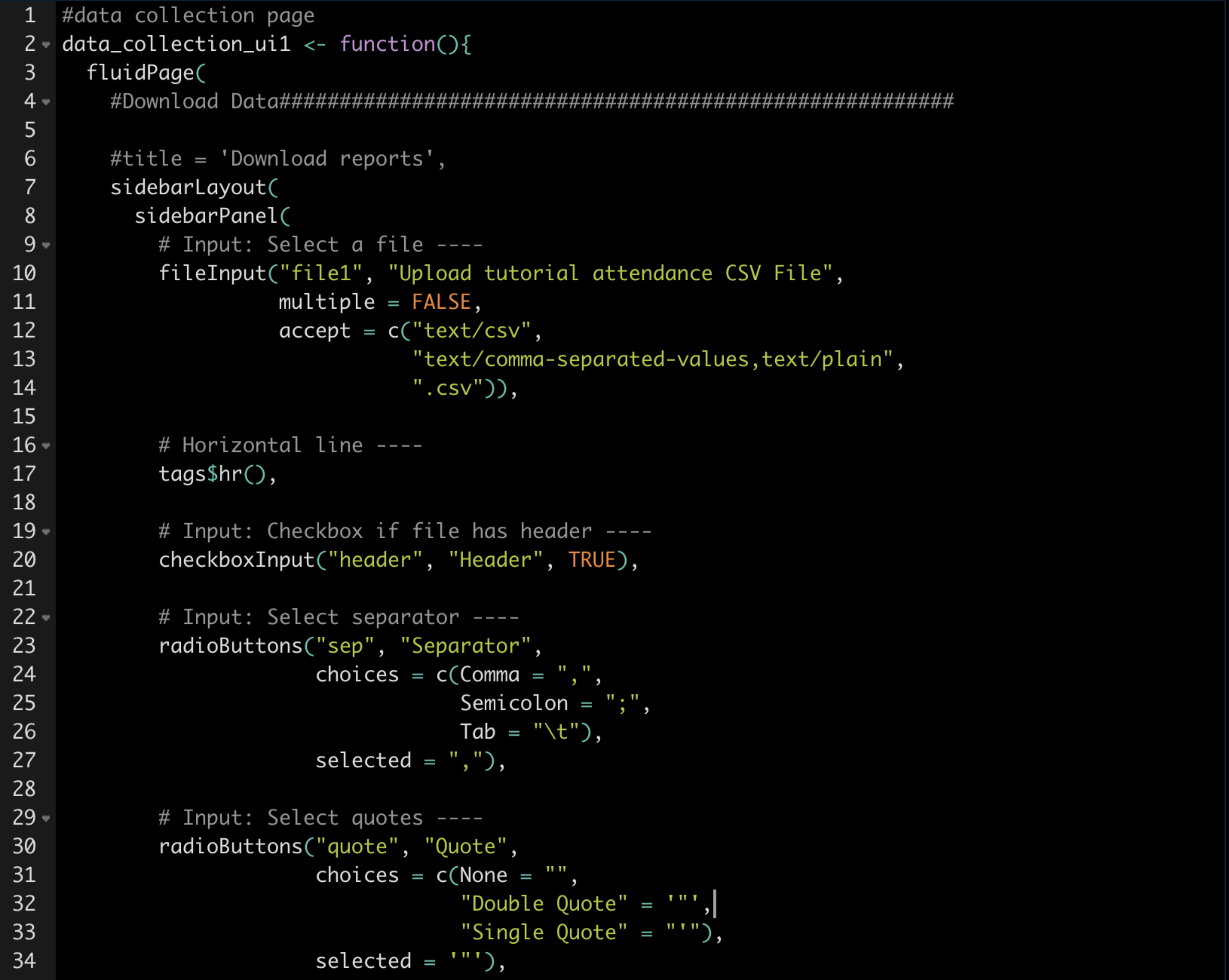
RStudio

Shiny

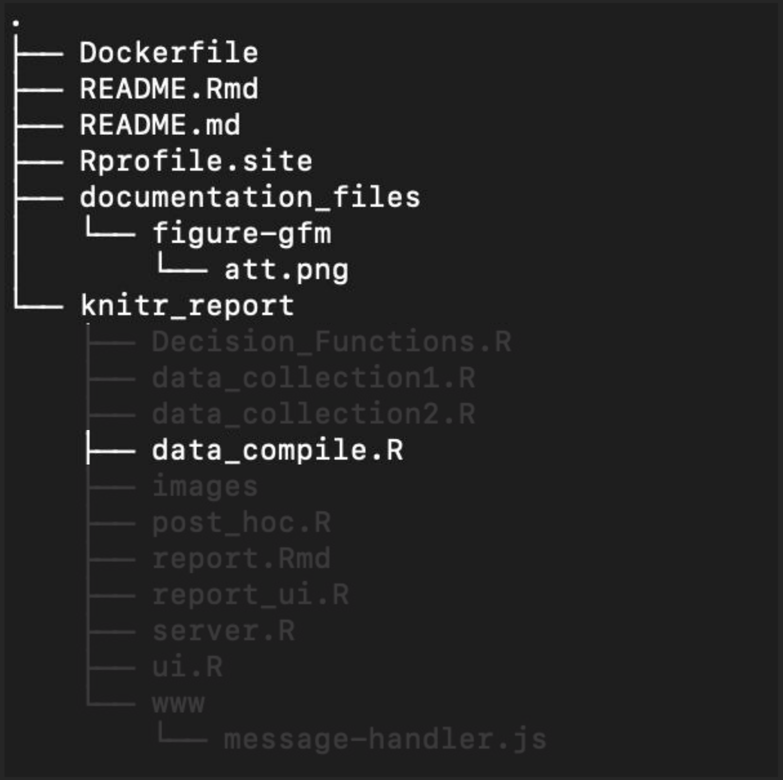
rmarkdown



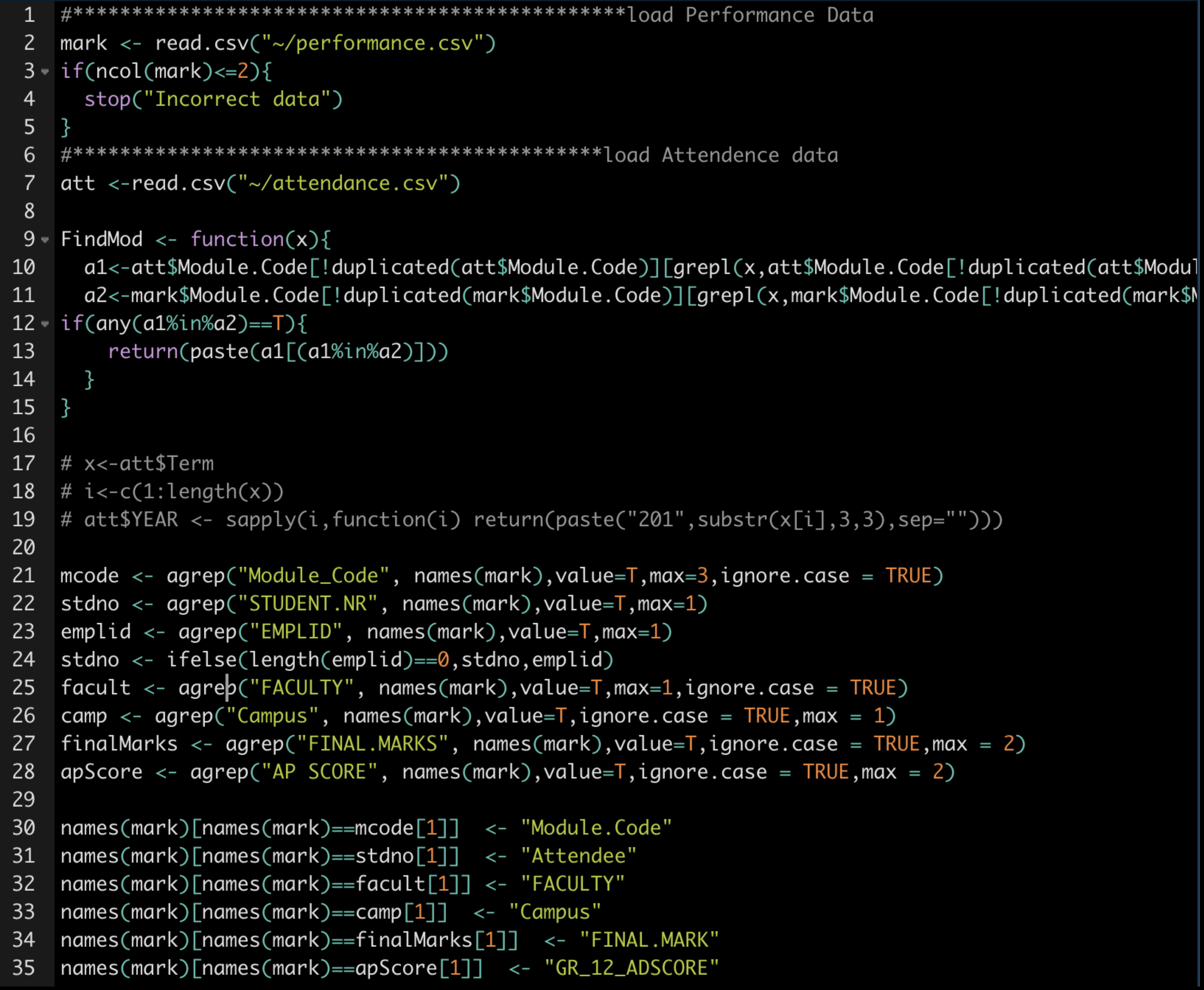
```
.
├── Dockerfile
├── README.Rmd
├── README.md
├── Rprofile.site
├── documentation_files
│   └── figure-gfm
│       └── att.png
└── knitr_report
    ├── Decision_Functions.R
    ├── data_collection1.R
    ├── data_collection2.R
    ├── data_compile.R
    ├── images
    ├── post_hoc.R
    ├── report.Rmd
    ├── report_ui.R
    ├── server.R
    ├── ui.R
    └── www
        └── message-handler.js
```



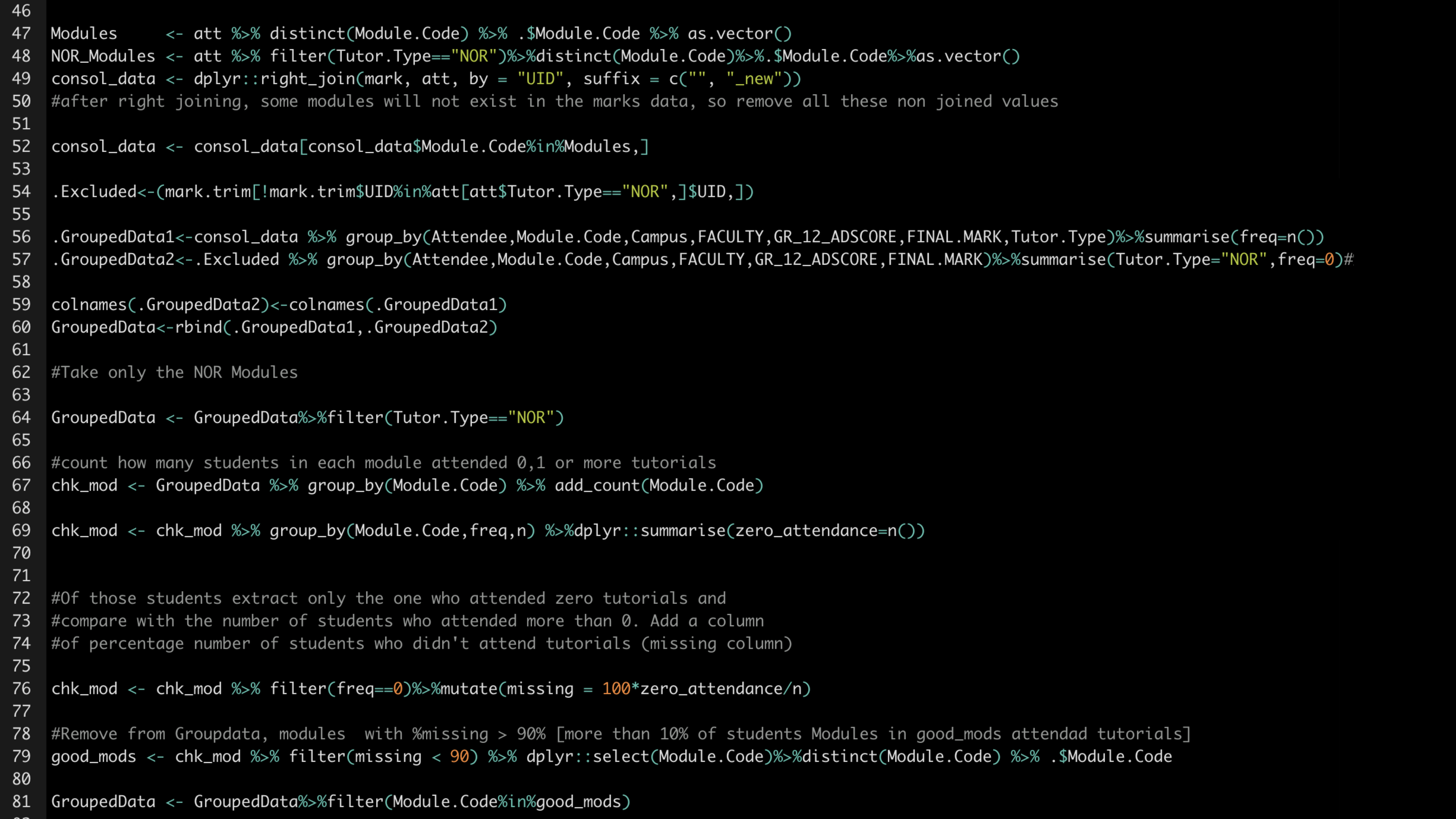
```r
#data collection page
data_collection_ui1 <- function(){
  fluidPage(
    #Download Data####################################################

    #title = 'Download reports',
    sidebarLayout(
      sidebarPanel(
        # Input: Select a file ----
        fileInput("file1", "Upload tutorial attendance CSV File",
                  multiple = FALSE,
                  accept = c("text/csv",
                             "text/comma-separated-values,text/plain",
                             ".csv")),

        # Horizontal line ----
        tags$hr(),

        # Input: Checkbox if file has header ----
        checkboxInput("header", "Header", TRUE),

        # Input: Select separator ----
        radioButtons("sep", "Separator",
                     choices = c(Comma = ",",
                                 Semicolon = ";",
                                 Tab = "\t"),
                     selected = ","),

        # Input: Select quotes ----
        radioButtons("quote", "Quote",
                     choices = c(None = "",
                                 "Double Quote" = '"',
                                 "Single Quote" = "'"),
                     selected = '"'),
```



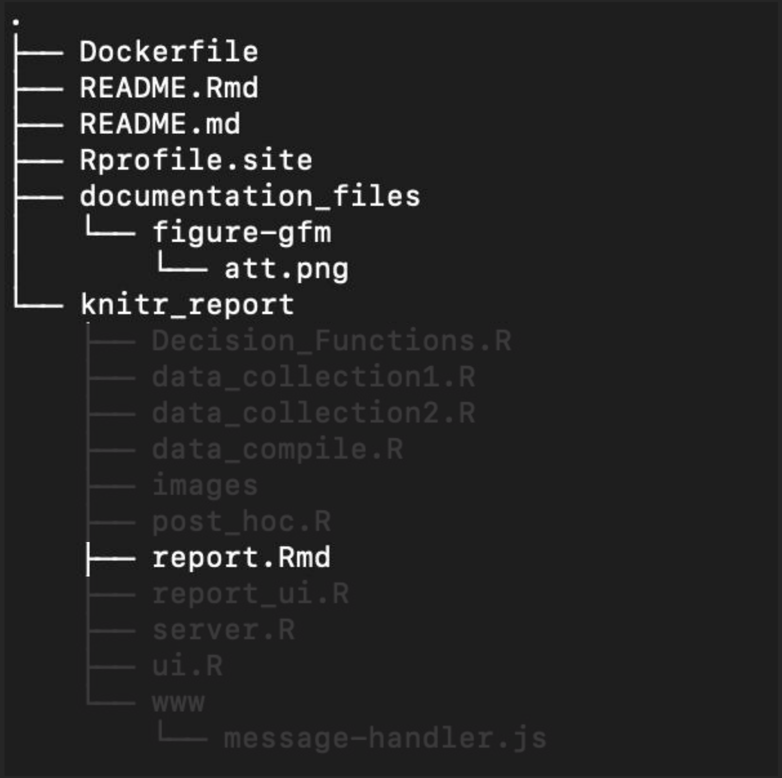
```
.
├── Dockerfile
├── README.Rmd
├── README.md
├── Rprofile.site
├── documentation_files
│   └── figure-gfm
│       └── att.png
└── knitr_report
    ├── Decision_Functions.R
    ├── data_collection1.R
    ├── data_collection2.R
    ├── data_compile.R
    ├── images
    ├── post_hoc.R
    ├── report.Rmd
    ├── report_ui.R
    ├── server.R
    ├── ui.R
    └── www
        └── message-handler.js
```



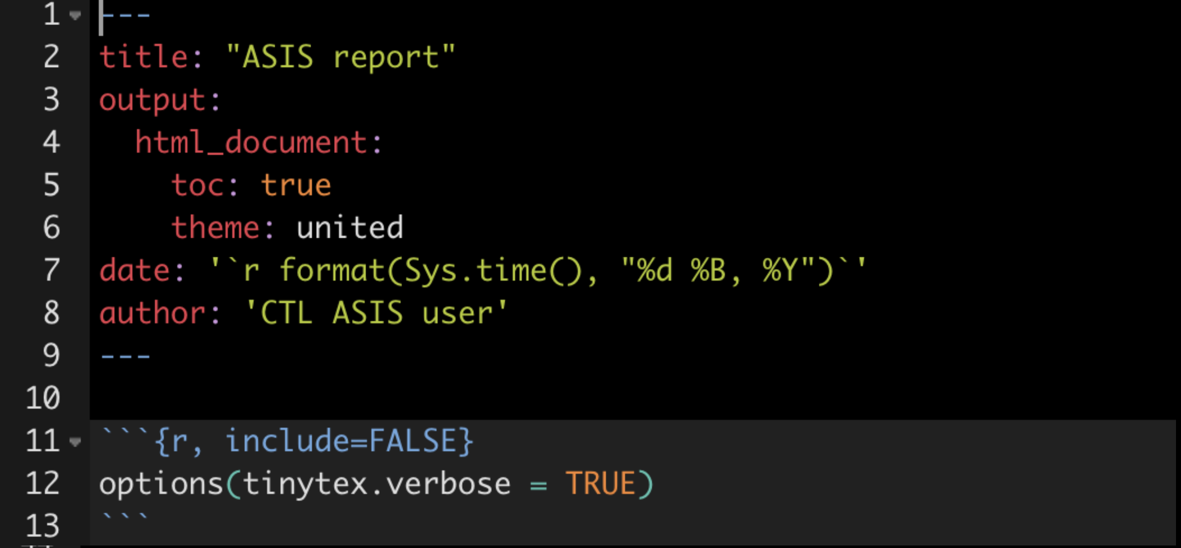
```
#*****************************************load Performance Data
mark <- read.csv("~/performance.csv")
if(ncol(mark)<=2){
  stop("Incorrect data")
}
#***************************************load Attendence data
att <-read.csv("~/attendance.csv")

FindMod <- function(x){
  a1<-att$Module.Code[!duplicated(att$Module.Code)][grepl(x,att$Module.Code[!duplicated(att$Modul
  a2<-mark$Module.Code[!duplicated(mark$Module.Code)][grepl(x,mark$Module.Code[!duplicated(mark$M
if(any(a1%in%a2)==T){
    return(paste(a1[(a1%in%a2)]))
  }
}

# x<-att$Term
# i<-c(1:length(x))
# att$YEAR <- sapply(i,function(i) return(paste("201",substr(x[i],3,3),sep="")))

mcode <- agrep("Module_Code", names(mark),value=T,max=3,ignore.case = TRUE)
stdno <- agrep("STUDENT.NR", names(mark),value=T,max=1)
emplid <- agrep("EMPLID", names(mark),value=T,max=1)
stdno <- ifelse(length(emplid)==0,stdno,emplid)
facult <- agrep("FACULTY", names(mark),value=T,max=1,ignore.case = TRUE)
camp <- agrep("Campus", names(mark),value=T,ignore.case = TRUE,max = 1)
finalMarks <- agrep("FINAL.MARKS", names(mark),value=T,ignore.case = TRUE,max = 2)
apScore <- agrep("AP SCORE", names(mark),value=T,ignore.case = TRUE,max = 2)

names(mark)[names(mark)==mcode[1]]  <- "Module.Code"
names(mark)[names(mark)==stdno[1]]  <- "Attendee"
names(mark)[names(mark)==facult[1]] <- "FACULTY"
names(mark)[names(mark)==camp[1]]  <- "Campus"
names(mark)[names(mark)==finalMarks[1]]  <- "FINAL.MARK"
names(mark)[names(mark)==apScore[1]]  <- "GR_12_ADSCORE"
```



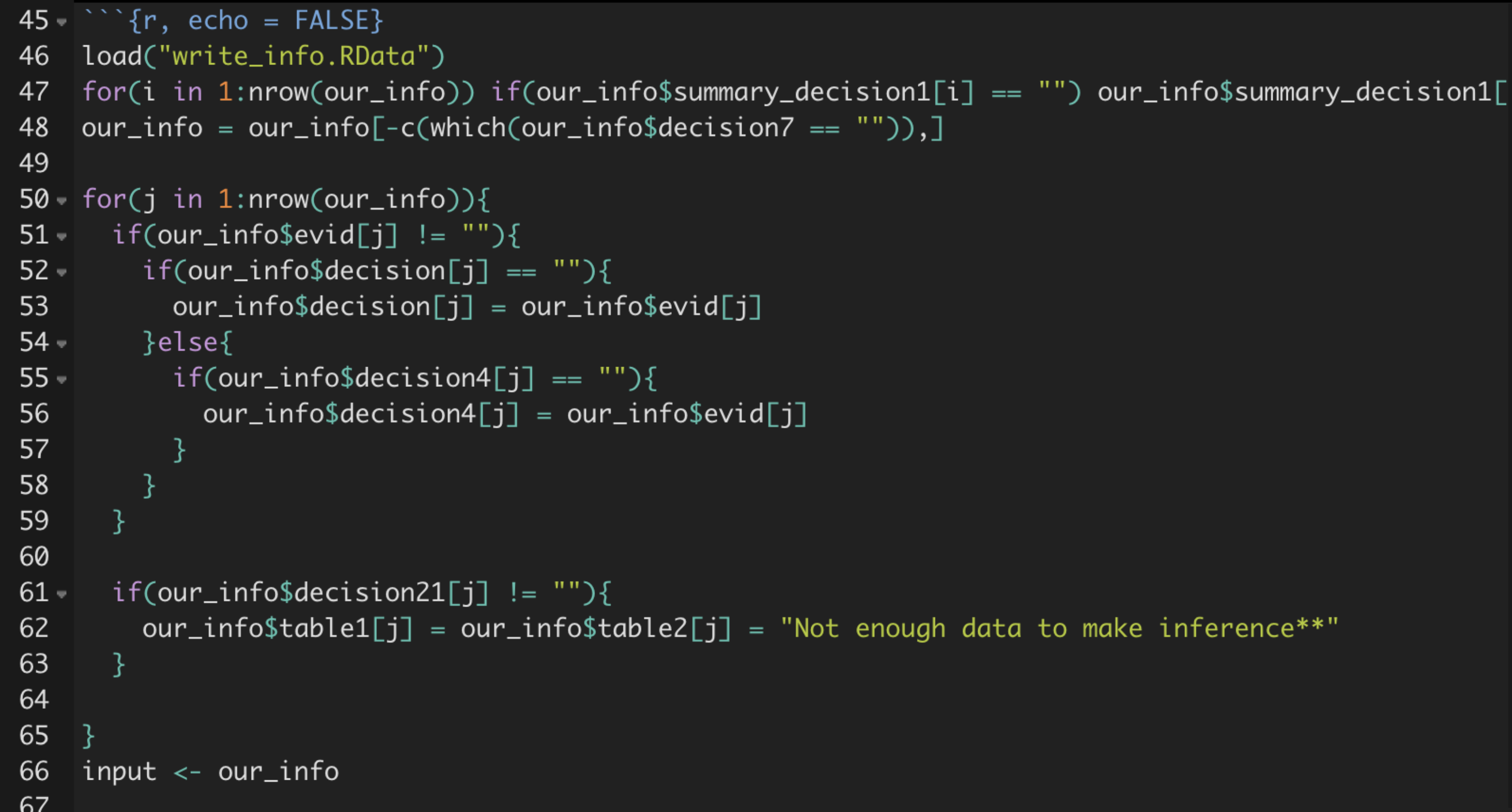
```

Modules     <- att %>% distinct(Module.Code) %>% .$Module.Code %>% as.vector()
NOR_Modules <- att %>% filter(Tutor.Type=="NOR")%>%distinct(Module.Code)%>%.$Module.Code%>%as.vector()
consol_data <- dplyr::right_join(mark, att, by = "UID", suffix = c("", "_new"))
#after right joining, some modules will not exist in the marks data, so remove all these non joined values

consol_data <- consol_data[consol_data$Module.Code%in%Modules,]

.Excluded<-(mark.trim[!mark.trim$UID%in%att[att$Tutor.Type=="NOR",]$UID,])

.GroupedData1<-consol_data %>% group_by(Attendee,Module.Code,Campus,FACULTY,GR_12_ADSCORE,FINAL.MARK,Tutor.Type)%>%summarise(freq=n())
.GroupedData2<-.Excluded %>% group_by(Attendee,Module.Code,Campus,FACULTY,GR_12_ADSCORE,FINAL.MARK)%>%summarise(Tutor.Type="NOR",freq=0)#

colnames(.GroupedData2)<-colnames(.GroupedData1)
GroupedData<-rbind(.GroupedData1,.GroupedData2)

#Take only the NOR Modules

GroupedData <- GroupedData%>%filter(Tutor.Type=="NOR")

#count how many students in each module attended 0,1 or more tutorials
chk_mod <- GroupedData %>% group_by(Module.Code) %>% add_count(Module.Code)

chk_mod <- chk_mod %>% group_by(Module.Code,freq,n) %>%dplyr::summarise(zero_attendance=n())


#Of those students extract only the one who attended zero tutorials and
#compare with the number of students who attended more than 0. Add a column
#of percentage number of students who didn't attend tutorials (missing column)

chk_mod <- chk_mod %>% filter(freq==0)%>%mutate(missing = 100*zero_attendance/n)

#Remove from Groupdata, modules  with %missing > 90% [more than 10% of students Modules in good_mods attendad tutorials]
good_mods <- chk_mod %>% filter(missing < 90) %>% dplyr::select(Module.Code)%>%distinct(Module.Code) %>% .$Module.Code

GroupedData <- GroupedData%>%filter(Module.Code%in%good_mods)
```



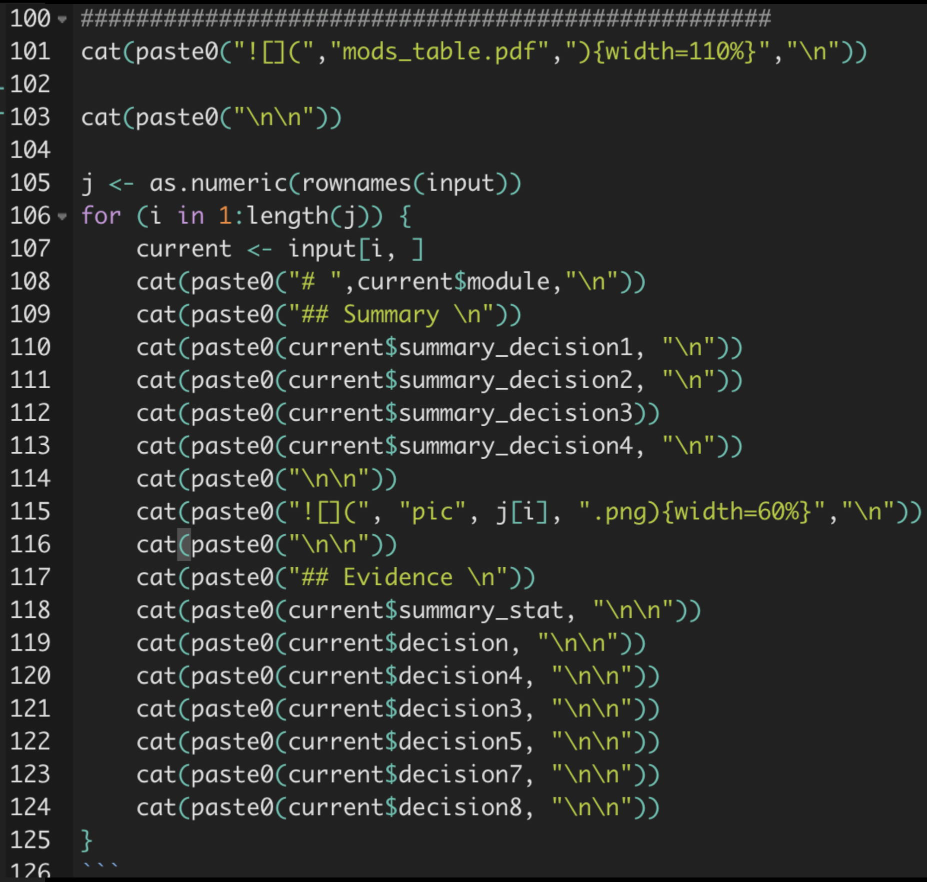
```
.
├── Dockerfile
├── README.Rmd
├── README.md
├── Rprofile.site
├── documentation_files
│   └── figure-gfm
│       └── att.png
└── knitr_report
    ├── Decision_Functions.R
    ├── data_collection1.R
    ├── data_collection2.R
    ├── data_compile.R
    ├── images
    ├── post_hoc.R
    ├── report.Rmd
    ├── report_ui.R
    ├── server.R
    ├── ui.R
    └── www
        └── message-handler.js
```



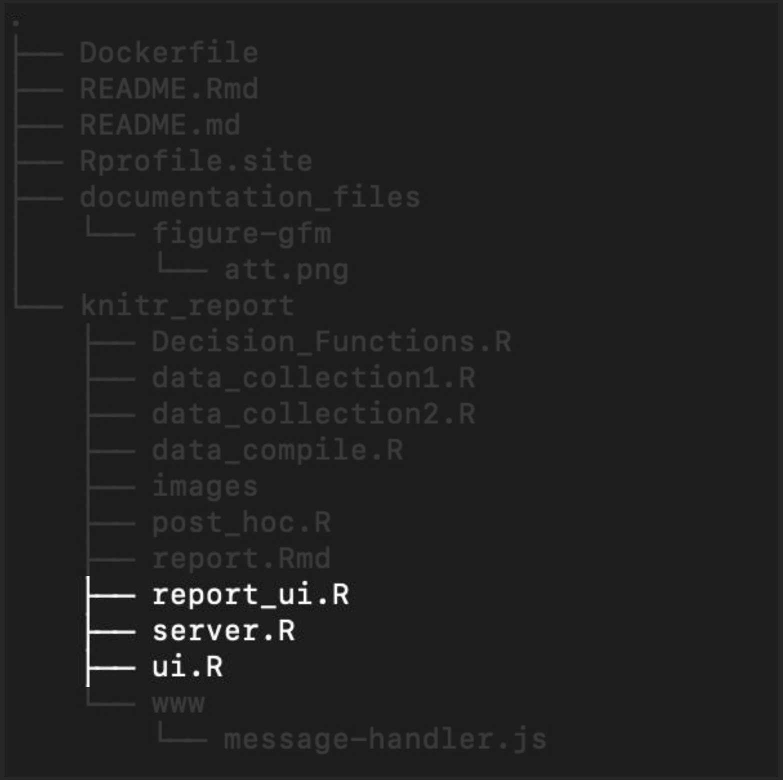
~~~
---
title: "ASIS report"
output:
  html_document:
    toc: true
    theme: united
date: '`r format(Sys.time(), "%d %B, %Y")`'
author: 'CTL ASIS user'
---

```{r, include=FALSE}
options(tinytex.verbose = TRUE)
```
~~~



~~~
```{r, echo = FALSE}
load("write_info.RData")
for(i in 1:nrow(our_info)) if(our_info$summary_decision1[i] == "") our_info$summary_decision1[
our_info = our_info[-c(which(our_info$decision7 == "")),]

for(j in 1:nrow(our_info)){
  if(our_info$evid[j] != ""){
    if(our_info$decision[j] == ""){
      our_info$decision[j] = our_info$evid[j]
    }else{
      if(our_info$decision4[j] == ""){
        our_info$decision4[j] = our_info$evid[j]
      }
    }
  }

  if(our_info$decision21[j] != ""){
    our_info$table1[j] = our_info$table2[j] = "Not enough data to make inference**"
  }

}
input <- our_info
~~~



```
##########################################
cat(paste0("![](","mods_table.pdf","){width=110%}","\n"))

cat(paste0("\n\n"))

j <- as.numeric(rownames(input))
for (i in 1:length(j)) {
    current <- input[i, ]
    cat(paste0("# ",current$module,"\n"))
    cat(paste0("## Summary \n"))
    cat(paste0(current$summary_decision1, "\n"))
    cat(paste0(current$summary_decision2, "\n"))
    cat(paste0(current$summary_decision3))
    cat(paste0(current$summary_decision4, "\n"))
    cat(paste0("\n\n"))
    cat(paste0("![](", "pic", j[i], ".png){width=60%}","\n"))
    cat(paste0("\n\n"))
    cat(paste0("## Evidence \n"))
    cat(paste0(current$summary_stat, "\n\n"))
    cat(paste0(current$decision, "\n\n"))
    cat(paste0(current$decision4, "\n\n"))
    cat(paste0(current$decision3, "\n\n"))
    cat(paste0(current$decision5, "\n\n"))
    cat(paste0(current$decision7, "\n\n"))
    cat(paste0(current$decision8, "\n\n"))
}
```
```
```



```
.
├── Dockerfile
├── README.Rmd
├── README.md
├── Rprofile.site
├── documentation_files
│   └── figure-gfm
│       └── att.png
└── knitr_report
    ├── Decision_Functions.R
    ├── data_collection1.R
    ├── data_collection2.R
    ├── data_compile.R
    ├── images
    ├── post_hoc.R
    ├── report.Rmd
    ├── report_ui.R
    ├── server.R
    ├── ui.R
    └── www
        └── message-handler.js
```

Testing

Containerization

We install Docker CE on our server.

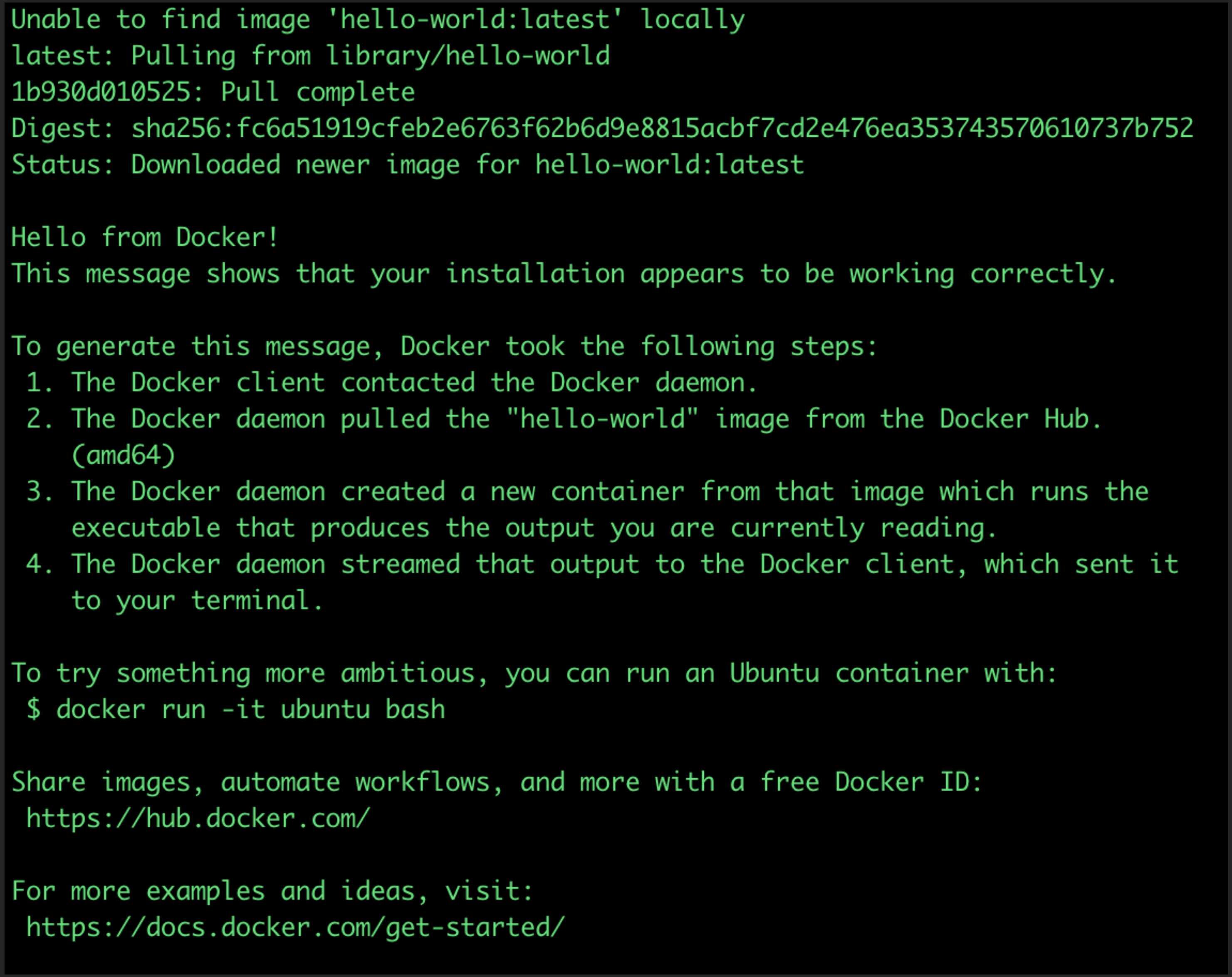
There is a need for some dependencies. It is also better to install the latest version of Docker CE using Docker servers as a third-party repository.

```
sudo apt-get install \ apt-transport-https \
ca-certificates \
curl \
gnupg2 \
software-properties-common
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-
key add - sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/debian \
$(lsb_release -cs) \
stable"
```

```
sudo apt-get install docker-ce
```

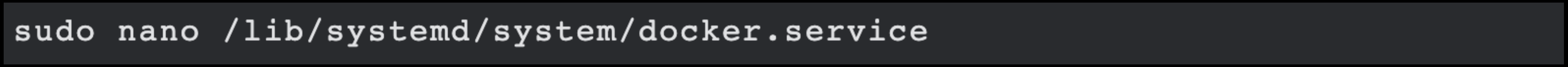
Test to see if it works

```
docker run hello-world
```

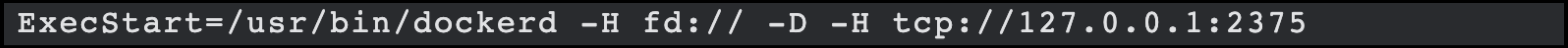


```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:fc6a51919cfeb2e6763f62b6d9e8815acbf7cd2e476ea353743570610737b752
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

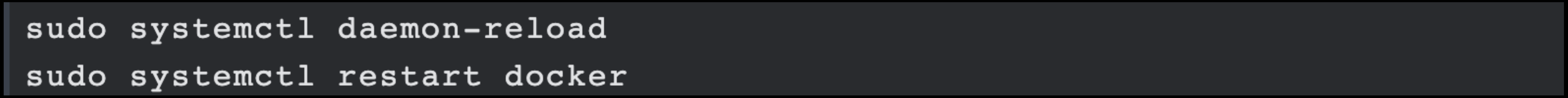
1. To ensure shinyproxy can hook docker instances we Modify the Docker file `/lib/systemd/systemd/system/docker.service` and restart Docker.



```
sudo nano /lib/systemd/system/docker.service
```



```
ExecStart=/usr/bin/dockerd -H fd:// -D -H tcp://127.0.0.1:2375
```



```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

2. Pull our knitr_report project from github and build an image.

```
git clone https://github.com/atwgao/knitr_report
```

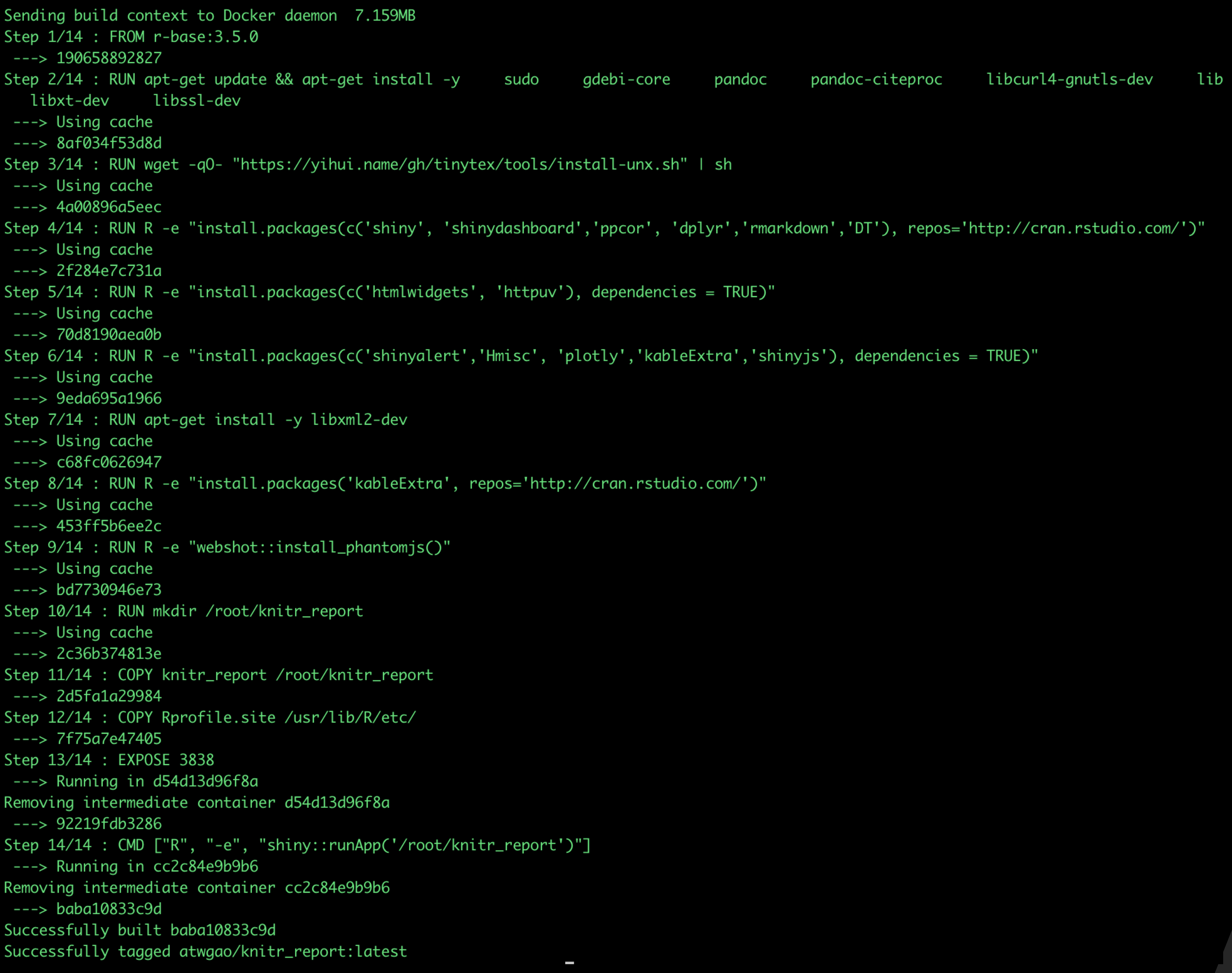
Navigate to where you placed the Dockerfile and run the following.

```
sudo docker build -t atwgao/knitr_report .
```

3. Push image to dockerhub

```
docker images
```

```
docker tag bb38******** atwgao/knitr_report:latest
```



```
Sending build context to Docker daemon  7.159MB
Step 1/14 : FROM r-base:3.5.0
 ---> 190658892827
Step 2/14 : RUN apt-get update && apt-get install -y     sudo     gdebi-core     pandoc     pandoc-citeproc     libcurl4-gnutls-dev     lib
   libxt-dev     libssl-dev
 ---> Using cache
 ---> 8af034f53d8d
Step 3/14 : RUN wget -qO- "https://yihui.name/gh/tinytex/tools/install-unx.sh" | sh
 ---> Using cache
 ---> 4a00896a5eec
Step 4/14 : RUN R -e "install.packages(c('shiny', 'shinydashboard','ppcor', 'dplyr','rmarkdown','DT'), repos='http://cran.rstudio.com/')"
 ---> Using cache
 ---> 2f284e7c731a
Step 5/14 : RUN R -e "install.packages(c('htmlwidgets', 'httpuv'), dependencies = TRUE)"
 ---> Using cache
 ---> 70d8190aea0b
Step 6/14 : RUN R -e "install.packages(c('shinyalert','Hmisc', 'plotly','kableExtra','shinyjs'), dependencies = TRUE)"
 ---> Using cache
 ---> 9eda695a1966
Step 7/14 : RUN apt-get install -y libxml2-dev
 ---> Using cache
 ---> c68fc0626947
Step 8/14 : RUN R -e "install.packages('kableExtra', repos='http://cran.rstudio.com/')"
 ---> Using cache
 ---> 453ff5b6ee2c
Step 9/14 : RUN R -e "webshot::install_phantomjs()"
 ---> Using cache
 ---> bd7730946e73
Step 10/14 : RUN mkdir /root/knitr_report
 ---> Using cache
 ---> 2c36b374813e
Step 11/14 : COPY knitr_report /root/knitr_report
 ---> 2d5fa1a29984
Step 12/14 : COPY Rprofile.site /usr/lib/R/etc/
 ---> 7f75a7e47405
Step 13/14 : EXPOSE 3838
 ---> Running in d54d13d96f8a
Removing intermediate container d54d13d96f8a
 ---> 92219fdb3286
Step 14/14 : CMD ["R", "-e", "shiny::runApp('/root/knitr_report')"]
 ---> Running in cc2c84e9b9b6
Removing intermediate container cc2c84e9b9b6
 ---> baba10833c9d
Successfully built baba10833c9d
Successfully tagged atwgao/knitr_report:latest
```

shinyproxy

1. Install Java

```
echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu xenial
main" | \ sudo tee /etc/apt/sources.list.d/webupd8team-java.list
echo "deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu
xenial main" | \ sudo tee -a /etc/apt/sources.list.d/webupd8team-
java.list sudo apt-key adv --keyserver
hkp://keyserver.ubuntu.com:80 --recv-keys EEA14886 sudo apt-get
install oracle-java8-installer
```

Check if it works:

```
java -version
```

2. Install shinyproxy which will be used to spin the containers:

```
wget https://www.shinyproxy.io/download/shinyproxy_2.2.3_amd64.deb
```

Inside the shinyproxy folder ensure there is a pom.xml file. From that directory. From this directory install shinyproxy (you may need to install maven)

```
sudo apt install maven
```

```
mvn -U clean install
```

Once shinyproxy is installed you can naviate to the target folder inside the shinyproxy folder and open the **`application.yml`** configuration file

```
proxy:
  title: Reporting dashboards
  logo-url:
https://www.________________________________
  landing-page: /
  heartbeat-rate: 10000
  heartbeat-timeout: 60000
  port: 8080
  authentication: simple
  admin-groups: tlcs
  # Example: 'simple' authentication configuration
  users:
  - name: gao
    password: statement
    groups: tlcs
  - name: providance
    password: password
    groups: tlcs
  # Example: 'ldap' authentication configuration
  ldap:
    url: ldap://ldap.forumsys.com:389/dc=example,dc=com
    user-dn-pattern: uid={0}
    group-search-base:
    group-search-filter: (uniqueMember={0})
    manager-dn: cn=read-only-admin,dc=example,dc=com
    manager-password: password
```

```
# Docker configuration
  docker:
    cert-path: /home/none
    url: http://localhost:2375
    port-range-start: 20000
  specs:
  - id: 01_asis
    display-name: ASIS
    description: This application generates
A_STEP impact reports
    container-image: atwgao/knitr_report:latest
    access-groups: [tlcs]

spring:
  servlet:
   multipart:
    max-file-size: 500MB
    max-request-size: 500MB

logging:
  file:
    shinyproxy.log
```

Deployment

Navigate to `~/shinyproxy/target` and run:

```
sudo nohup java –jar shinyproxy-2.2.3.jar
```

For debugging purposes it’s advisable to run without **`nohup`**