First I will load the required packages.

library(tidyverse) library(gutenbergr) library(tidytext)

Since Pride and Prejudice is currently the most downloaded book on gutenberg over the past 30 days and I happen to like it myself, I’ll use that as my example book. It is easy to download the full text with the gutenbergr package.

book\_text <- gutenberg\_download(1342)

data("stop\_words") # load stop words book\_text %>%

unnest\_tokens(word, text) %>% # turn the text into a single column of words

mutate(word = str\_extract(string = word, pattern = "[[:alpha:]]+"))

%>% # remove any non-alphanumeric characters select(word) %>% # get rid of the extra columns unique() %>% # get rid of duplicate words

anti\_join(stop\_words, by = "word") %>% # get rid of boring "stop" words

drop\_na() %>% # drop anything that didn't make it through cleanly unlist() %>% # turn the column into a vector that sample() knows what

to do with

sample(4) # chose four words at random

## word273 word4600 word632 word4122 ## "choose" "content" "circumstance" "mingled"

Well that was super easy, wasn’t it? The only thing that isn’t easy with this setup is looking up a book to use. But that can readily be solved with a shiny app. You can see the code below

library(shiny) library(shinythemes) library(gutenbergr) library(dplyr) library(tidyr) library(tidytext) library(stringr) library(rclipboard)

# get all the titles for the drop-down menu

titles <- gutenberg\_works(only\_text = TRUE, distinct = TRUE) %>% select(title) %>%

drop\_na()

# load the stop words so that we don't have to reload it later data("stop\_words")

ui <- fluidPage(theme = shinytheme("cerulean"),

rclipboardSetup(), # what it sounds like verticalLayout(

fluidRow(

column(width = 8, offset = 1,

titlePanel(title = "XKCD-Inspired, Gutenberg-Sourced

Passwords"), by ",

Gutenberg"),

p("This web-app lets you generate passwords inspired

a(href = "https://xkcd.com/936/", "this xkcd comic."), br(),

"First select a book from ",

a(href = "https://[www.gutenberg.org/](http://www.gutenberg.org/)", "Project

" and then chose the number of words you want to

use from that book for your password.")

),

),

fluidRow(

column(width = 6, offset = 1, selectizeInput(inputId = "book\_title",

label = "Book Title", choices = c("Chose one" = "",

titles), # removes the default selection, but needs error handling for the down-stream items

selected = NULL), #choices = titles,

#selected = "Pride and Prejudice"), p(textOutput("book\_length")), sliderInput("number\_of\_words",

"Number of words to chose", min = 1,

max = 10,

value = 4))

),

# Show the password fluidRow(

column(width = 6, offset = 1, tags$hr(),

textOutput("password", container = tags$strong)

),

),

# Show the password without spaces fluidRow(

column(width = 6, offset = 1, uiOutput("password\_no\_spaces"))

)

)

)

server <- function(input, output) { # get the book

gutenberg\_book <- reactive({ validate(

need(input$book\_title != "", "Please chose a book.")

)

gutenberg\_works(title == input$book\_title) %>% # get the gutenberg id

gutenberg\_download() %>%

unnest\_tokens(word, text) %>% # turn the text into a single column of words

mutate(word = str\_extract(string = word, pattern = "[[:alpha:]]+")) %>% # remove any non-alphanumeric characters.

select(word) %>% # get rid of the extra columns unique() %>% # get rid of duplicate words

anti\_join(stop\_words, by = "word") %>% # get rid of boring,

"stop" words cleanly

})

drop\_na() %>% # drop anything that didn't make it through unlist()

# Report the number of unique words in the book output$book\_length <- renderText({

length <- gutenberg\_book() %>% length() %>%

format(big.mark = ",") # Add some nice formating

paste0("There are ", length, " unique words in this book (including diffent forms of the same word).")

})

# Generate the actual password from the book password <- reactive({

validate(

need(input$book\_title != "", "")

)

gutenberg\_book() %>%

sample(input$number\_of\_words) %>% # chose words at random paste0() # drop the names

})

# Output the password for the UI output$password <- renderText({

password()

})

# Make the button to copy the password to the clipboard output$password\_no\_spaces <- renderUI({

rclipButton("clip\_button", paste0("Copy \"", str\_flatten(password()), "\""), str\_flatten(password()))

})

}

# Run the application

shinyApp(ui = ui, server = server)