# INSTALL SOFTWARE, CREATE ENVIRONMENTS, RUN REFRESH PIPELINES, AND MAKE FRIENDS..... AUTOMATICALLY!

## JUST DOUBLE CLICK ON THE BATCH SCRIPT

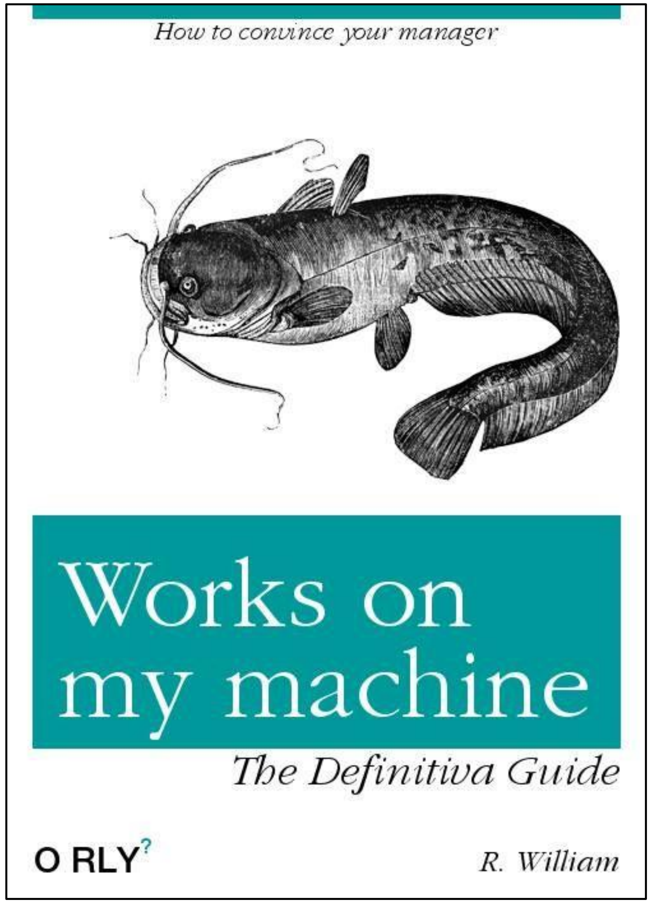
Washington State Department of HEALTH

Russell Shean,
Visualization Section, Center for Analytics, Informatics and Modernization (AIM)

# Have you ever written a really cool R script??

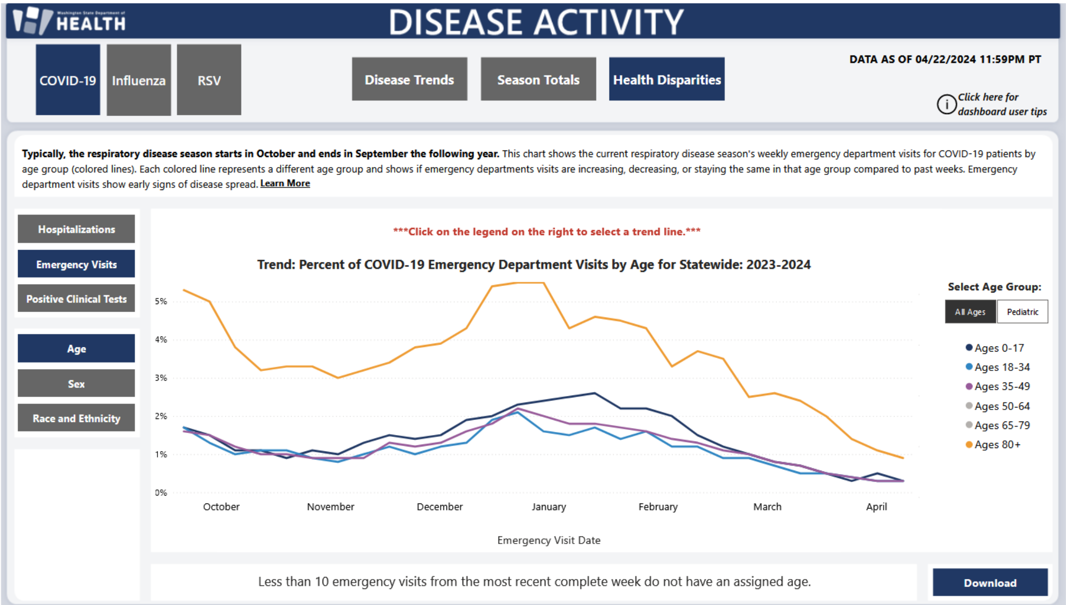
- Only to be told “But I don’t have R installed on my computer”
- “I have to open Rstudio and then click on the project and then click the pull button and then what?”….
- “Sounds complicated”

- **What if they could run a script that installs all the things they need to run R  and then just runs the script??**



# Data Visualization Section

- **Visualization Products:** We specialize in developing fully automatized data dashboards that integrate various data sources for quick insights.
- **Enterprise GIS Solutions:** We promote, support, and expand the use of Enterprise Geographic Information Systems (GIS) throughout the agency.
- **Customized Training:** We can support you with training and knowledge transfers to design and develop dashboards and visualizations using various tools.
- **Technical Support and Troubleshooting:** Have difficulties with your dashboard? Reach out to us and we will work with you to find a solution.



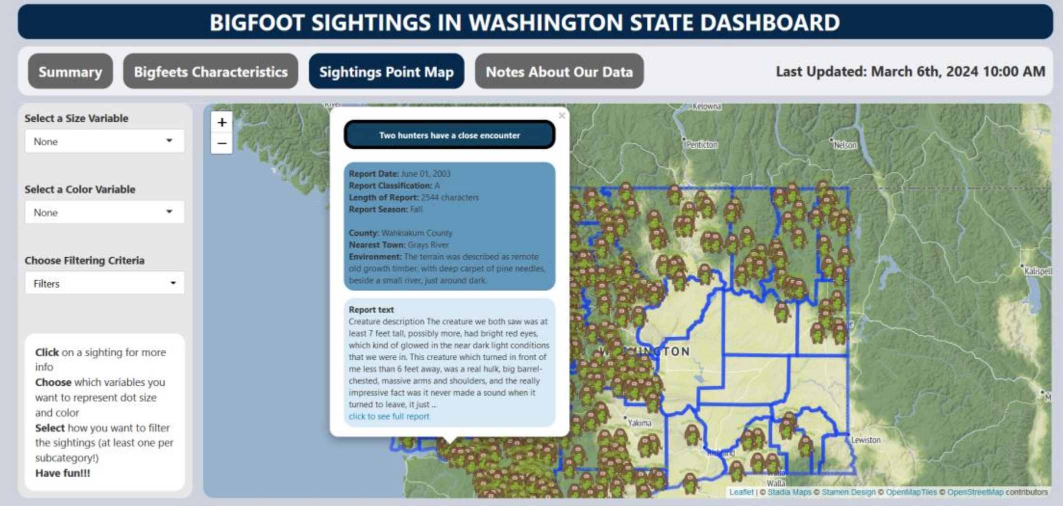
DATA AS OF 04/22/2024 11:59PM PT

**Typically, the respiratory disease season starts in October and ends in September the following year.** This chart shows the current respiratory disease season's weekly emergency department visits for COVID-19 patients by age group (colored lines). Each colored line represents a different age group and shows if emergency departments visits are increasing, decreasing, or staying the same in that age group compared to past weeks. Emergency department visits show early signs of disease spread. Learn More

***Click on the legend on the right to select a trend line.***

Trend: Percent of COVID-19 Emergency Department Visits by Age for Statewide: 2023-2024

Less than 10 emergency visits from the most recent complete week do not have an assigned age.



Last Updated: March 6th, 2024 10:00 AM

Two hunters have a close encounter

**Report Date:** June 01, 2003
**Report Classification:** A
**Length of Report:** 2544 characters
**Report Season:** Fall

**County:** Wahkiakum County
**Nearest Town:** Grays River
**Environment:** The terrain was described as remote old growth timber, with deep carpet of pine needles, beside a small river, just around dark.

**Report text**
Creature description The creature we both saw was at least 7 feet tall, possibly more, had bright red eyes, which kind of glowed in the near dark light conditions that we were in. This creature which turned in front of me less than 6 feet away, was a real hulk, big barrel-chested, massive arms and shoulders, and the really impressive fact was it never made a sound when it turned to leave, it just ...
click to see full report

**Click** on a sighting for more info
**Choose** which variables you want to represent dot size and color
**Select** how you want to filter the sightings (at least one per subcategory!)
**Have fun!!!**

# Power BI Dashboard Lifecycle

We build dashboards for partners using their data and design specifications

## Development

- Step 1: Decide on dashboard type, metrics, design, timeline, etc
- Step 2: Build pipelines to process data **(R)**
- Step 3: Build dashboard visualizations **(Power BI)**
- Step 4: Publish dashboard

## Post- Development

- Step 1: Re-run data pipelines to reprocess new data as it arrives ← Today's talk
- Step 2: Update visualizations to use newly processed data
- Step 3: Republish dashboard

# Refresh handovers

- After the data pipeline code is written we give it back to our partners so they can refresh the data themselves
- Pipelines are written in R
- Challenges for other teams:
  - Install required software
  - Clone our code from Github
  - Make sure code is up-to-date
  - Set up environment and install R packages
  - Run the code

- Time intensive: Over an hour for initial training
- Error prone: mismatched packages, objects in the environment, git pull, wrong file, missing software

# Wouldn't it be great if that was all automatic?

- Yes! That's why we wrote batch scripts

- Batch scripts use a (windows) shell language to send instructions to the (windows) operating system
- We use batch scripts to **automatically**:
  - Check for and install R, Rtools, Rstudio, Pandoc, and git
  - Create a file structure
  - Clone remote GitHub repositories containing all our pipeline code
  - Pull in changes from github and switch branches
  - Run the R scripts!

# Check for software

- First step to automatically installing software is seeing if it already exists!

- Two different methods
  - Check for software in a specific location

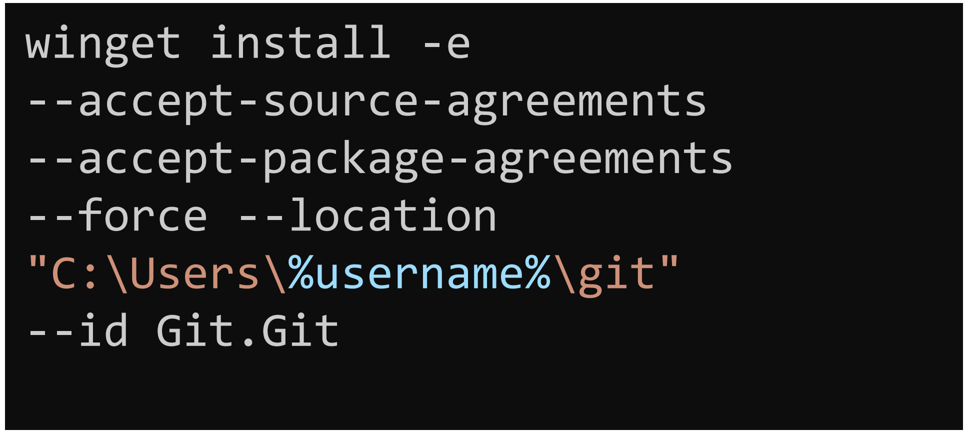
    ```
    if exist "C:/Program Files/R" (
         echo R is installed on your computer
    ```

  - Check for software in any location on C drive

    ```
    where /q git
      IF ERRORLEVEL 1 ( git is not installed! )
    ```

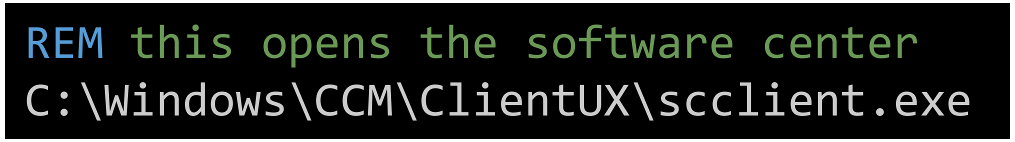
# Install software

- First choice: winget
  - Windows package manager… (sort of similar to apt on linux)
  - Comes on all our computers ……(so far at least!)
  - Automatically installs software
  - Additional arguments can be passed
  - No need to provide list of websites to visit and software to install!



```
winget install -e
--accept-source-agreements
--accept-package-agreements
--force --location
"C:\Users\%username%\git"
--id Git.Git
```

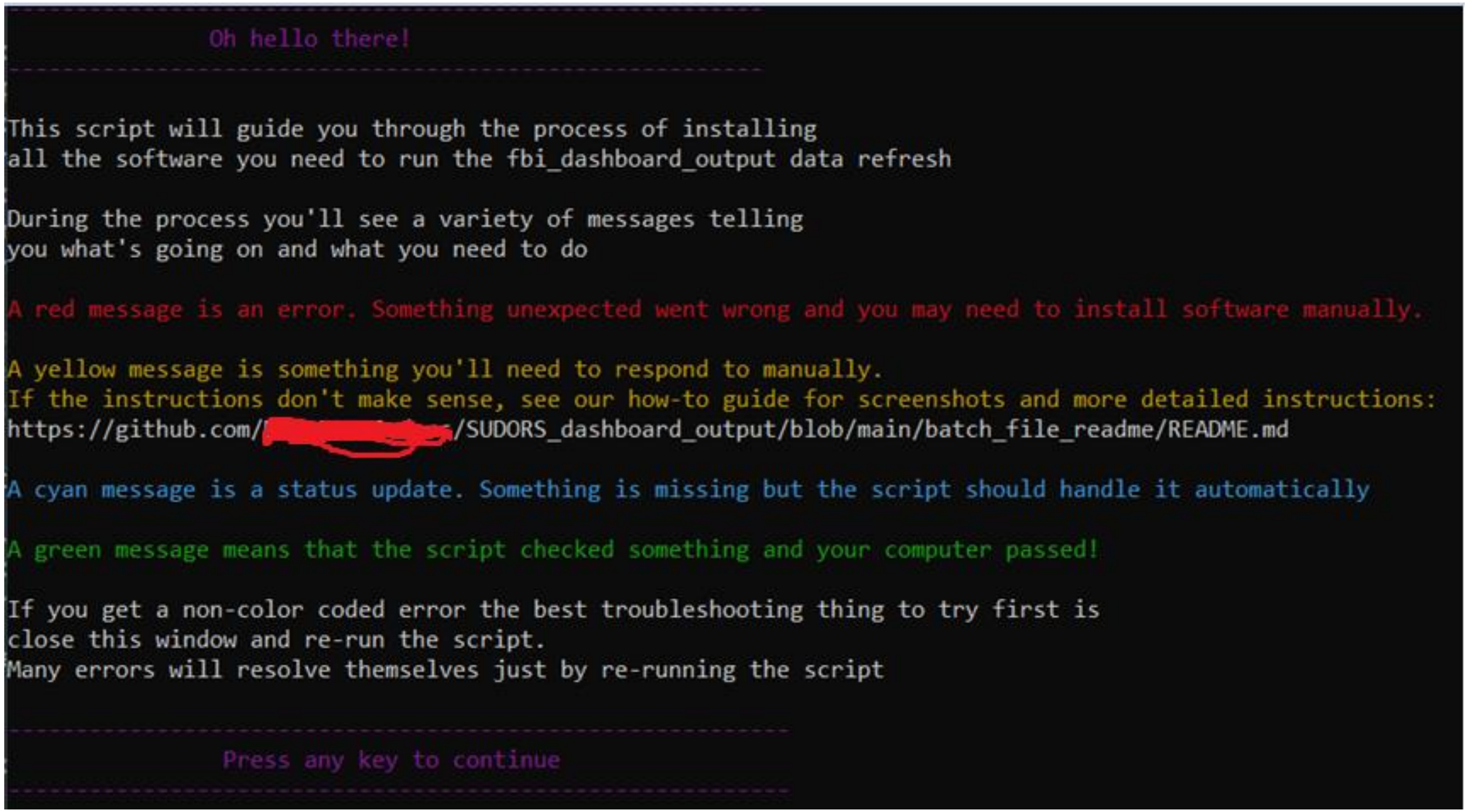
- Second choice: launch software center, company portal, or provide link
  - Installation blocked by IT
  - Winget not found



```
REM this opens the software center
C:\Windows\CCM\ClientUX\scclient.exe
```

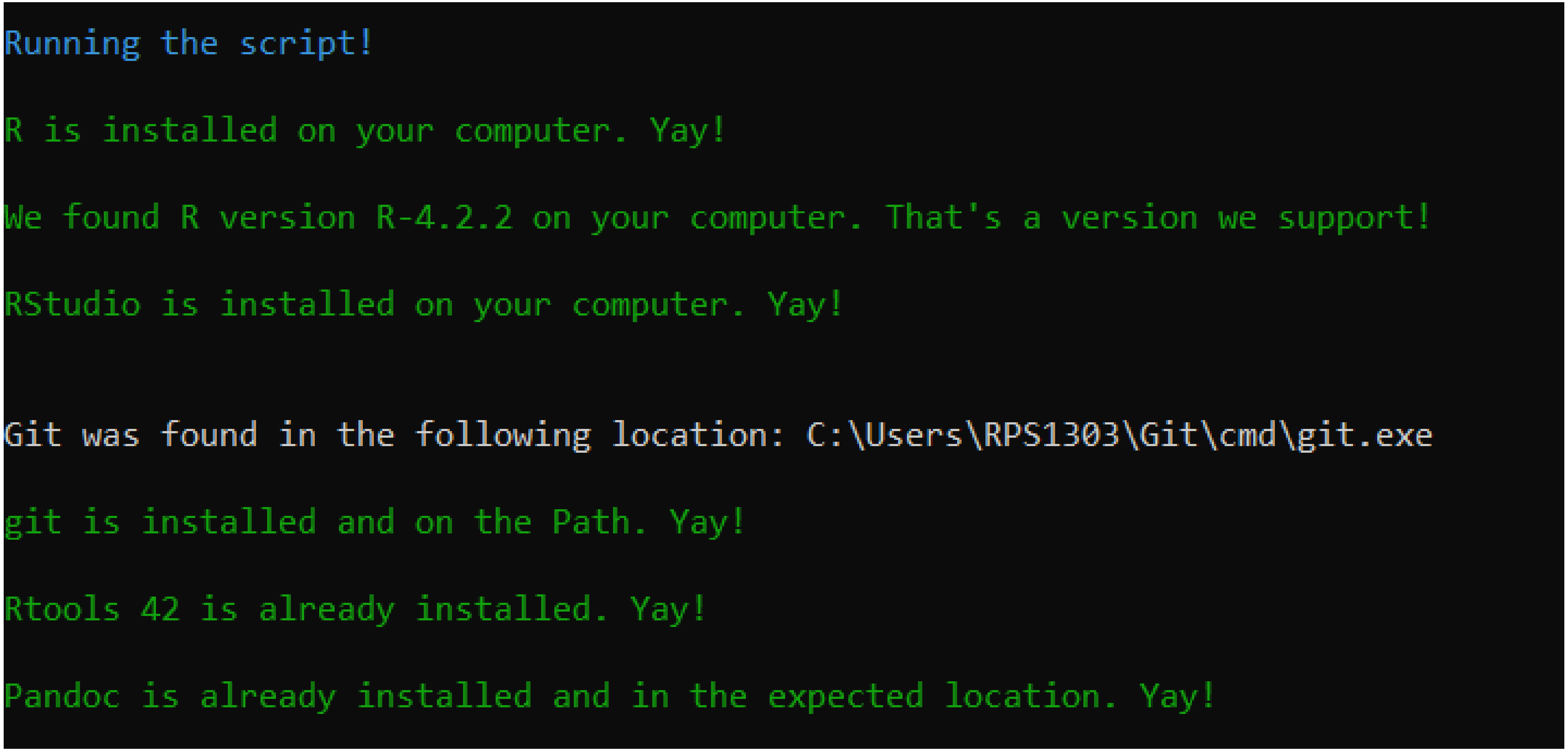
# Messages to user

- Not everything can (or should!) be automatic
- We provide color coded messages to guide users
- Provide updates
- Explain manual steps they need to take
- Explain choices they need to make



```
-----------------------------------------------------------------
               Oh hello there!
-----------------------------------------------------------------


This script will guide you through the process of installing
all the software you need to run the fbi_dashboard_output data refresh

During the process you'll see a variety of messages telling
you what's going on and what you need to do

A red message is an error. Something unexpected went wrong and you may need to install software manually.

A yellow message is something you'll need to respond to manually.
If the instructions don't make sense, see our how-to guide for screenshots and more detailed instructions:
https://github.com/[redacted]/SUDORS_dashboard_output/blob/main/batch_file_readme/README.md

A cyan message is a status update. Something is missing but the script should handle it automatically

A green message means that the script checked something and your computer passed!

If you get a non-color coded error the best troubleshooting thing to try first is
close this window and re-run the script.
Many errors will resolve themselves just by re-running the script


-------------------------------------------------------------------
                Press any key to continue
-------------------------------------------------------------------
```

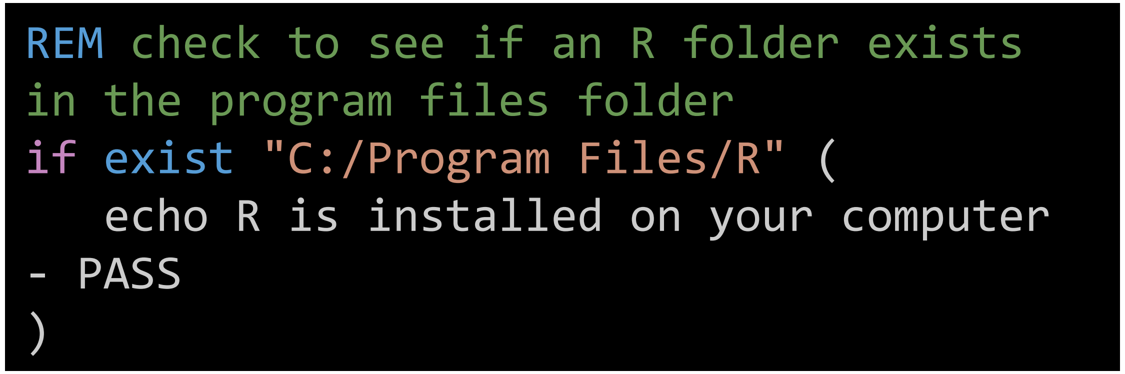
# Messages to user



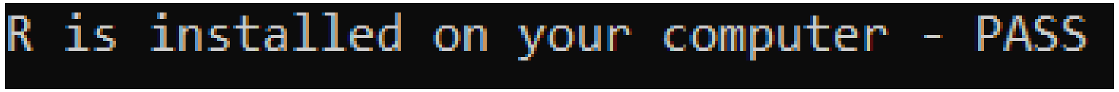
```
Running the script!

R is installed on your computer. Yay!

We found R version R-4.2.2 on your computer. That's a version we support!

RStudio is installed on your computer. Yay!


Git was found in the following location: C:\Users\RPS1303\Git\cmd\git.exe

git is installed and on the Path. Yay!

Rtools 42 is already installed. Yay!

Pandoc is already installed and in the expected location. Yay!
```

# Pretty colors

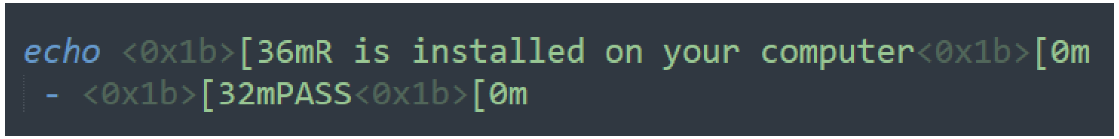
- Text is displayed to the user using the echo command
- Colors are created using ANSI escape characters: https://stackoverflow.com/a/38617204
- You can't use some text editors because they strip out the escape characters
  - Sublime Text ✓
  - Visual Studio Code ✓
  - Notepad++ ✓
  - Rstudio ✗



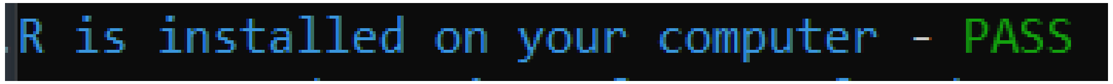
```
REM check to see if an R folder exists
in the program files folder
if exist "C:/Program Files/R" (
   echo R is installed on your computer
- PASS
)
```



```
R is installed on your computer - PASS
```

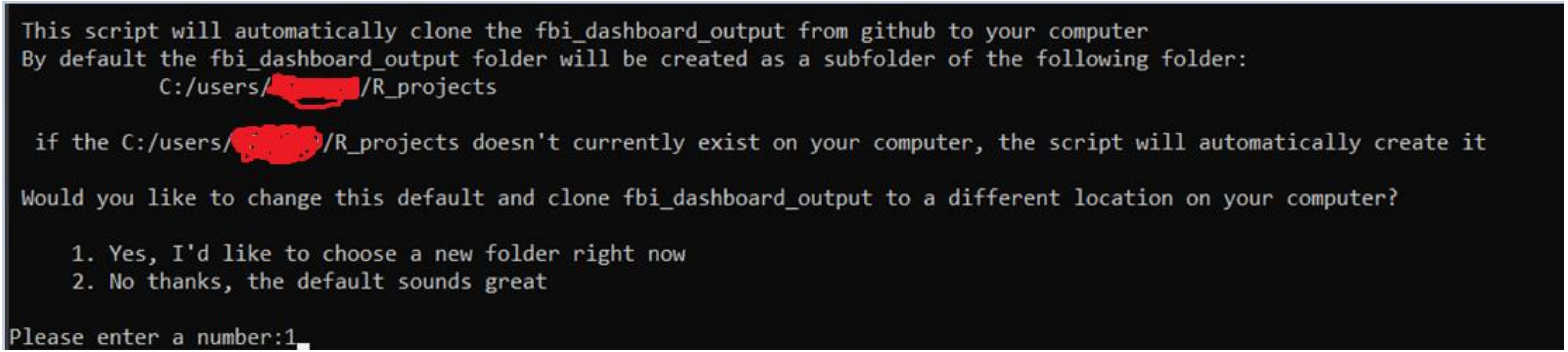


```
echo <0x1b>[36mR is installed on your computer<0x1b>[0m
 - <0x1b>[32mPASS<0x1b>[0m
```



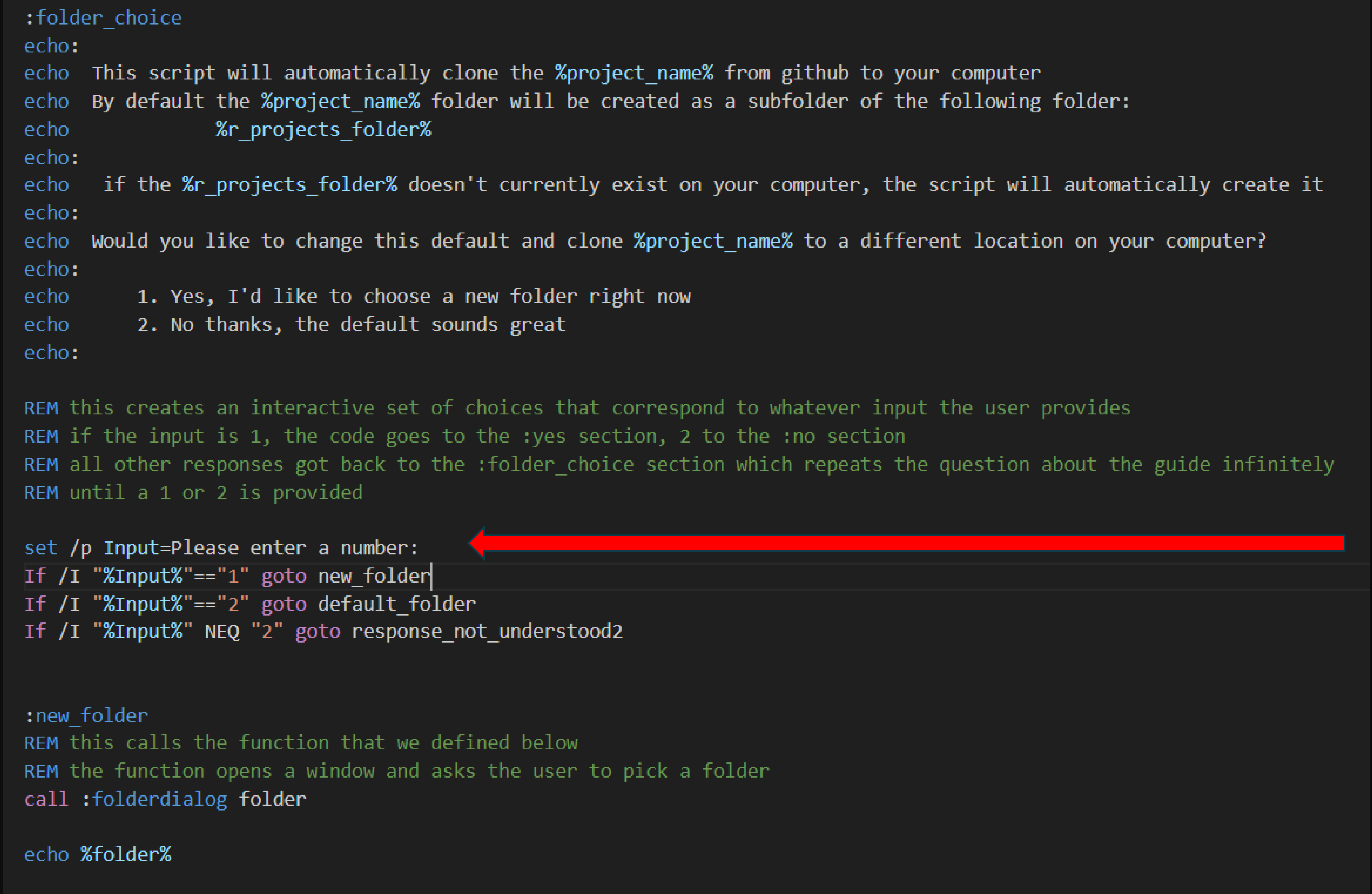
```
R is installed on your computer - PASS
```

# User input



```
This script will automatically clone the fbi_dashboard_output from github to your computer
By default the fbi_dashboard_output folder will be created as a subfolder of the following folder:
          C:/users/[redacted]/R_projects

 if the C:/users/[redacted]/R_projects doesn't currently exist on your computer, the script will automatically create it

Would you like to change this default and clone fbi_dashboard_output to a different location on your computer?

    1. Yes, I'd like to choose a new folder right now
    2. No thanks, the default sounds great

Please enter a number:1
```

- Users can make choice
  - View a how-to-use guide
  - Accept default folder location or choose a new one
  - Run the refresh or exit after software is installed
- Users use a dialogue box to manually choose new folder

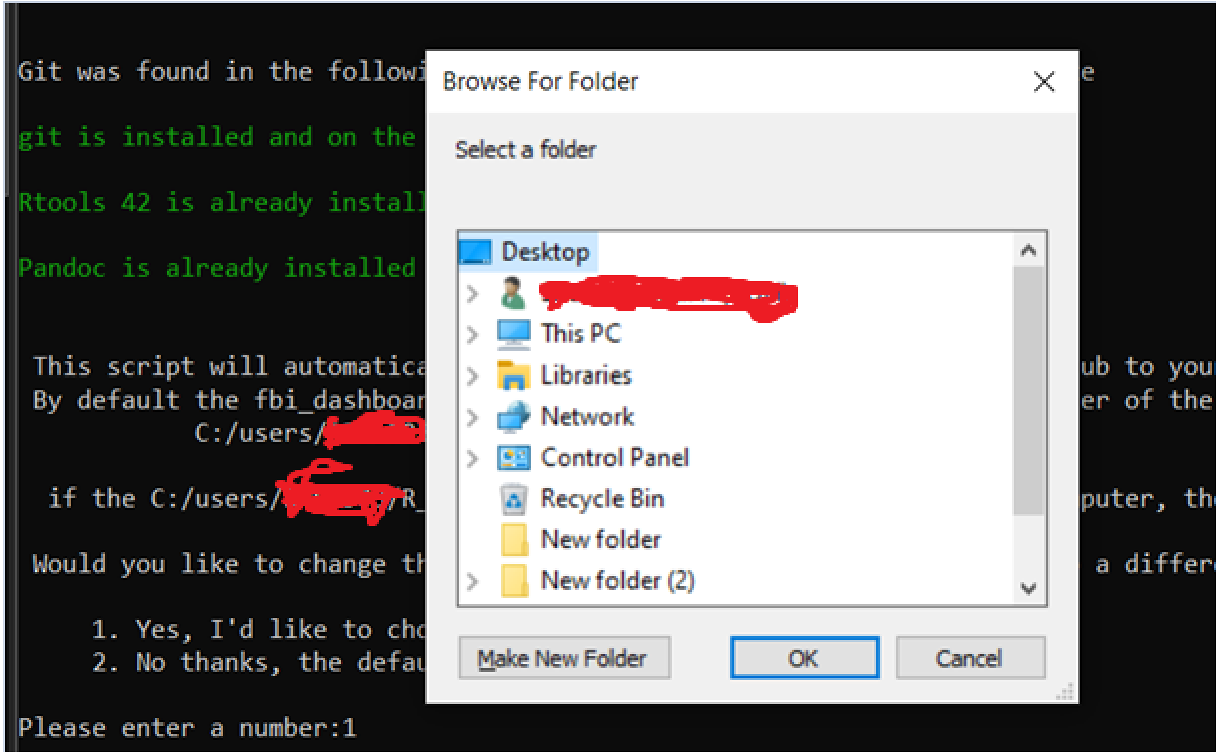


```
:folder_choice
echo:
echo  This script will automatically clone the %project_name% from github to your computer
echo  By default the %project_name% folder will be created as a subfolder of the following folder:
echo             %r_projects_folder%
echo:
echo   if the %r_projects_folder% doesn't currently exist on your computer, the script will automatically create it
echo:
echo  Would you like to change this default and clone %project_name% to a different location on your computer?
echo:
echo      1. Yes, I'd like to choose a new folder right now
echo      2. No thanks, the default sounds great
echo:

REM this creates an interactive set of choices that correspond to whatever input the user provides
REM if the input is 1, the code goes to the :yes section, 2 to the :no section
REM all other responses got back to the :folder_choice section which repeats the question about the guide infinitely
REM until a 1 or 2 is provided

set /p Input=Please enter a number:
If /I "%Input%"=="1" goto new_folder
If /I "%Input%"=="2" goto default_folder
If /I "%Input%" NEQ "2" goto response_not_understood2


:new_folder
REM this calls the function that we defined below
REM the function opens a window and asks the user to pick a folder
call :folderdialog folder

echo %folder%
```

# Choose your own folder!

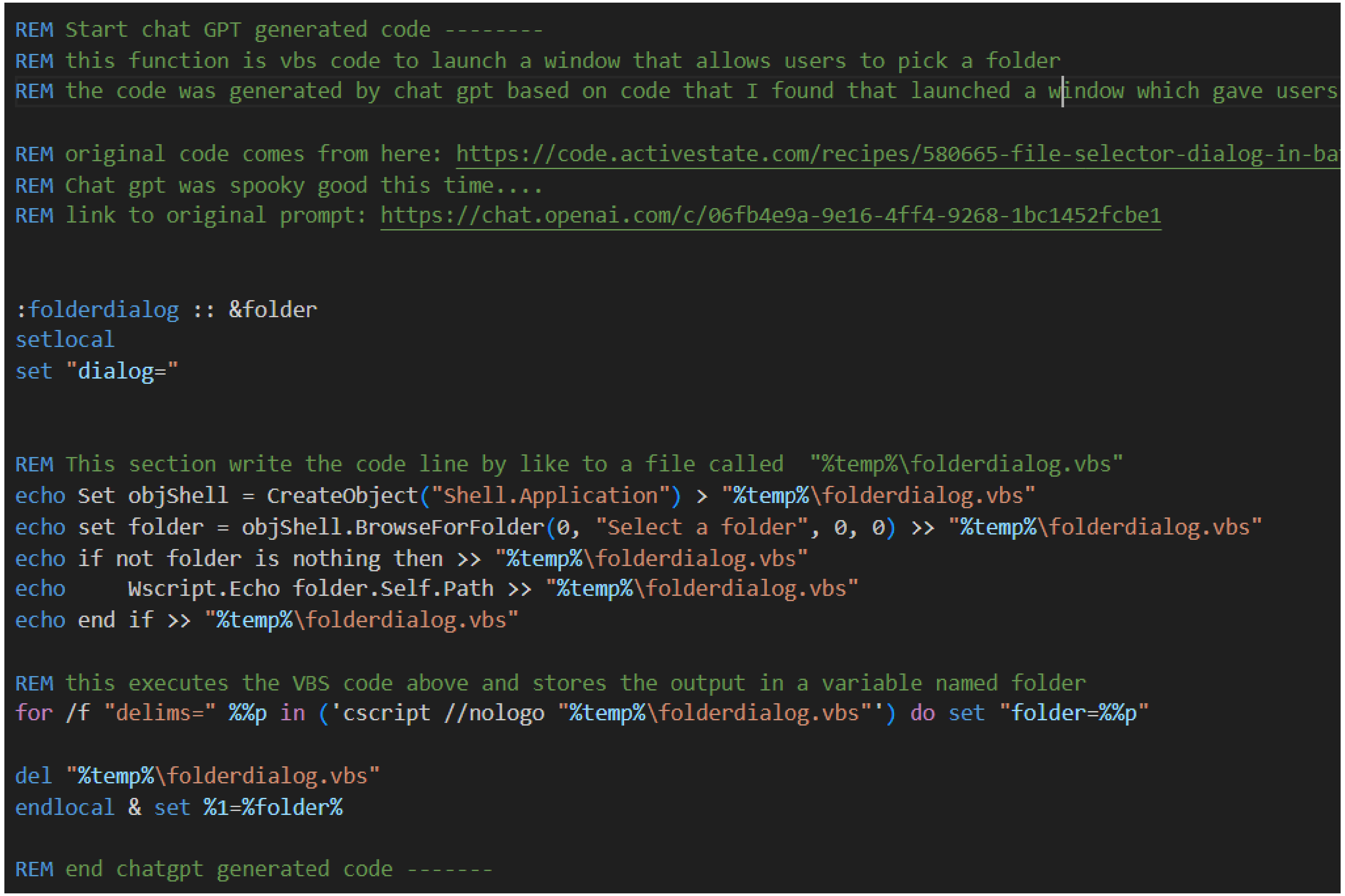
- Manually chosen folder is stored as a variable…….overwriting default
- Git repository is cloned into the folder the user chose OR the default
- If the folder doesn’t exist, it’s created

```
if not exist "%r_projects_folder%"
mkdir "%r_projects_folder%“


pushd "%r_projects_folder%"

git clone
https://github.com/Russell-
Shean/%project_name%
```



```
Git was found in the followi                                              e

git is installed and on the

Rtools 42 is already install

Pandoc is already installed


 This script will automatica                                              ub to you
 By default the fbi_dashboar                                              er of the
            C:/users/[redacted]

  if the C:/users/[redacted]/R_                                           puter, th

 Would you like to change th                                                a differ

     1. Yes, I'd like to cho
     2. No thanks, the defau

Please enter a number:1
```

# Folder choice



```
REM Start chat GPT generated code --------
REM this function is vbs code to launch a window that allows users to pick a folder
REM the code was generated by chat gpt based on code that I found that launched a window which gave users

REM original code comes from here: https://code.activestate.com/recipes/580665-file-selector-dialog-in-ba
REM Chat gpt was spooky good this time....
REM link to original prompt: https://chat.openai.com/c/06fb4e9a-9e16-4ff4-9268-1bc1452fcbe1


:folderdialog :: &folder
setlocal
set "dialog="


REM This section write the code line by like to a file called  "%temp%\folderdialog.vbs"
echo Set objShell = CreateObject("Shell.Application") > "%temp%\folderdialog.vbs"
echo set folder = objShell.BrowseForFolder(0, "Select a folder", 0, 0) >> "%temp%\folderdialog.vbs"
echo if not folder is nothing then >> "%temp%\folderdialog.vbs"
echo     Wscript.Echo folder.Self.Path >> "%temp%\folderdialog.vbs"
echo end if >> "%temp%\folderdialog.vbs"

REM this executes the VBS code above and stores the output in a variable named folder
for /f "delims=" %%p in ('cscript //nologo "%temp%\folderdialog.vbs"') do set "folder=%%p"

del "%temp%\folderdialog.vbs"
endlocal & set %1=%folder%

REM end chatgpt generated code -------
```

- Launches an interactive window that lets users choose their own folder
- Clicked on folder stored as a variable
- ChatGPT generated code ‼️👀👀 💥
- Reach out for more details!

# Git operations

- Get the code

```
pushd "%r_projects_folder%"
git clone https://github.com/DOH-EPI-Coders/%project_name%
```

- Keep the code up-to-date

```
REM this switches the working directory
REM it's equivalent to setwd() in R
cd "%r_projects_folder%/%project_name%"

echo [36mpulling in changes from github![0m
git pull
```

- Switch to correct branch

```
git switch %branch_name%
```

# Run R script

- Determine R version

```
if exist "C:/Program Files/R/R-4.2.3" ( set "r_version=R-4.2.3")
if exist "C:/Program Files/R/R-4.2.2" ( set "r_version=R-4.2.2")
if exist "C:/Program Files/R/R-4.2.1" ( set "r_version=R-4.2.1")
```

- Use correct R version to run script

```
"C:/Program Files/R/%r_version%/bin/Rscript.exe" "R/refresh_data_fbi.R"
```

- First argument is location of R executable
- Second argument is location of R file we want to execute
- R script sources other R scripts

# What does the R script do??

- Installs packages and correct versions of each package
  - `if(!require("renv")){install.packages("renv")}`
  - `renv::restore()`
- Checks that VPN is connected
  - Use cmd prompt to get information about network connections
  - Process output in R
  - `shell("ipconfig", intern = TRUE)`
- Process the data!
  - ELT
  - Aggregation
  - Calculate rates and new fields
  - Small number suppression
  - Save output

# Things I glossed over

- Switching slash directions for different windows commands
- variables inside of if statements https://stackoverflow.com/a/9102569
- Loops
- text manipulation
- Delayed expansion
- Network drives vs hard drive
- Functions

- Not core concepts, but important for getting edge cases to work
- Happy to answer questions,  share code, or connect offline

# Why not PowerShell?

- PowerShell is incredibly slow on my computer
  - Speed  can be probably be configured
  - But for this to be portable,  all computers need to be configured
- We want something that users can just double click to run
- This script started out small
- Tried to put as much as possible into R script NOT batch script

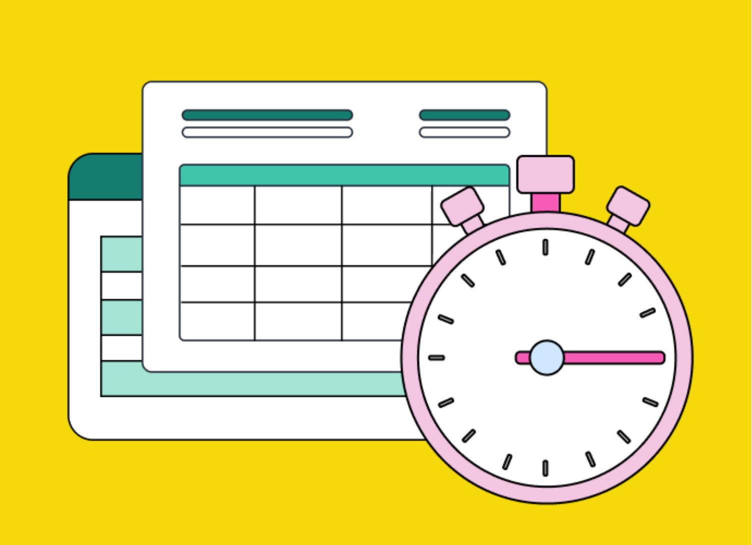
- If you have PowerShell ideas, please reach out!

# Future Fun Possibilities

- Task scheduler!!!
- These scripts could be scheduled to automatically run refreshes on a schedule
  - Manual set up in task scheduler

OR

  - Task could be created with a batch script using `schtask` module
  - [Microsoft learn - schtask]()



- Python
- Databricks command line
- PowerBI version control + data backup

# Get in touch!

- Where’s the code???
  - https://github.com/Russell-Shean/batch_scripts

- For specific questions/ideas about batch scripts:
  - Russell.shean@doh.wa.gov

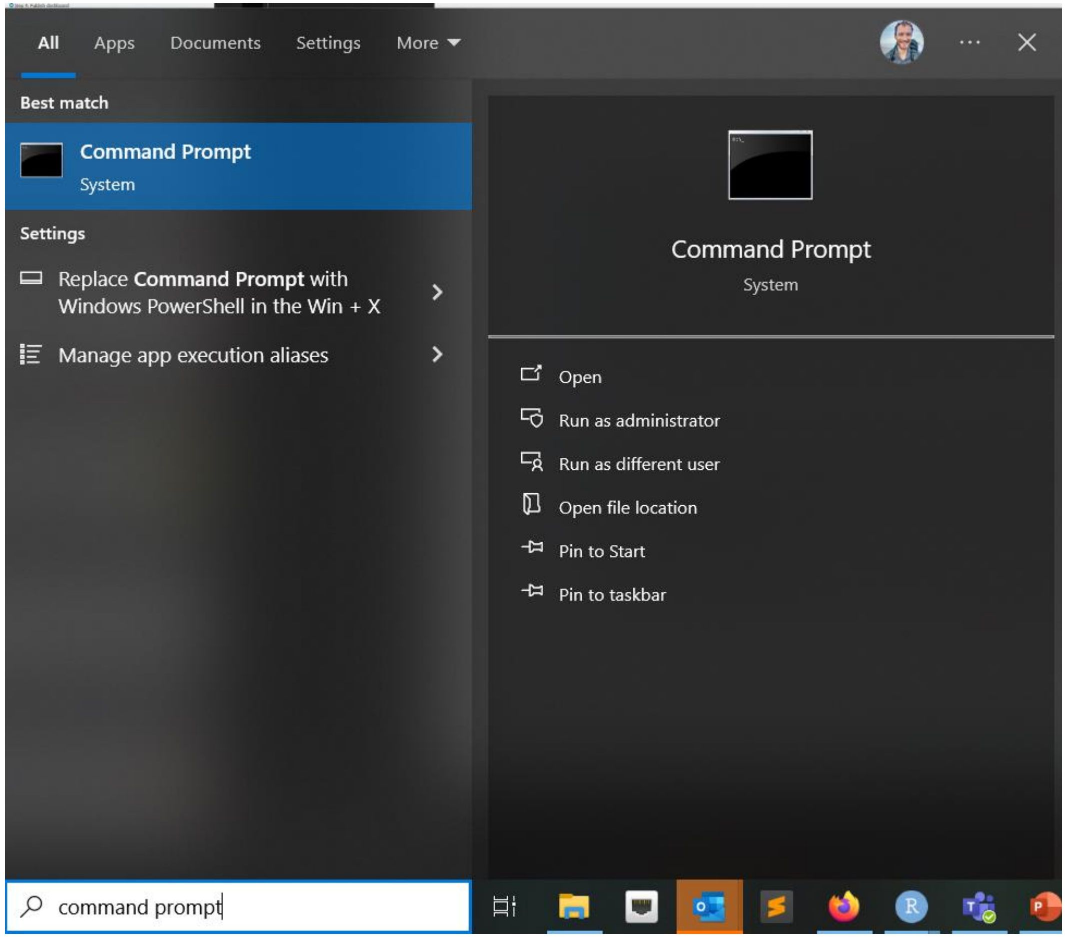
- For general questions about the visualization section or to learn how we can help YOU build beautiful, easy-to-update dashboards 🧑‍💻⚙️🧑‍🏭🤖 = 💪💯💯💯💅😎
  - CDS-Dashboards@doh.wa.gov

Washington State Department of
HEALTH

To request this document in another format, call 1-800-525-0127. Deaf or hard of hearing customers, please call 711 (Washington Relay) or email civil.rights@doh.wa.gov.

# Command prompt demo
# (You *can* try this at home!)

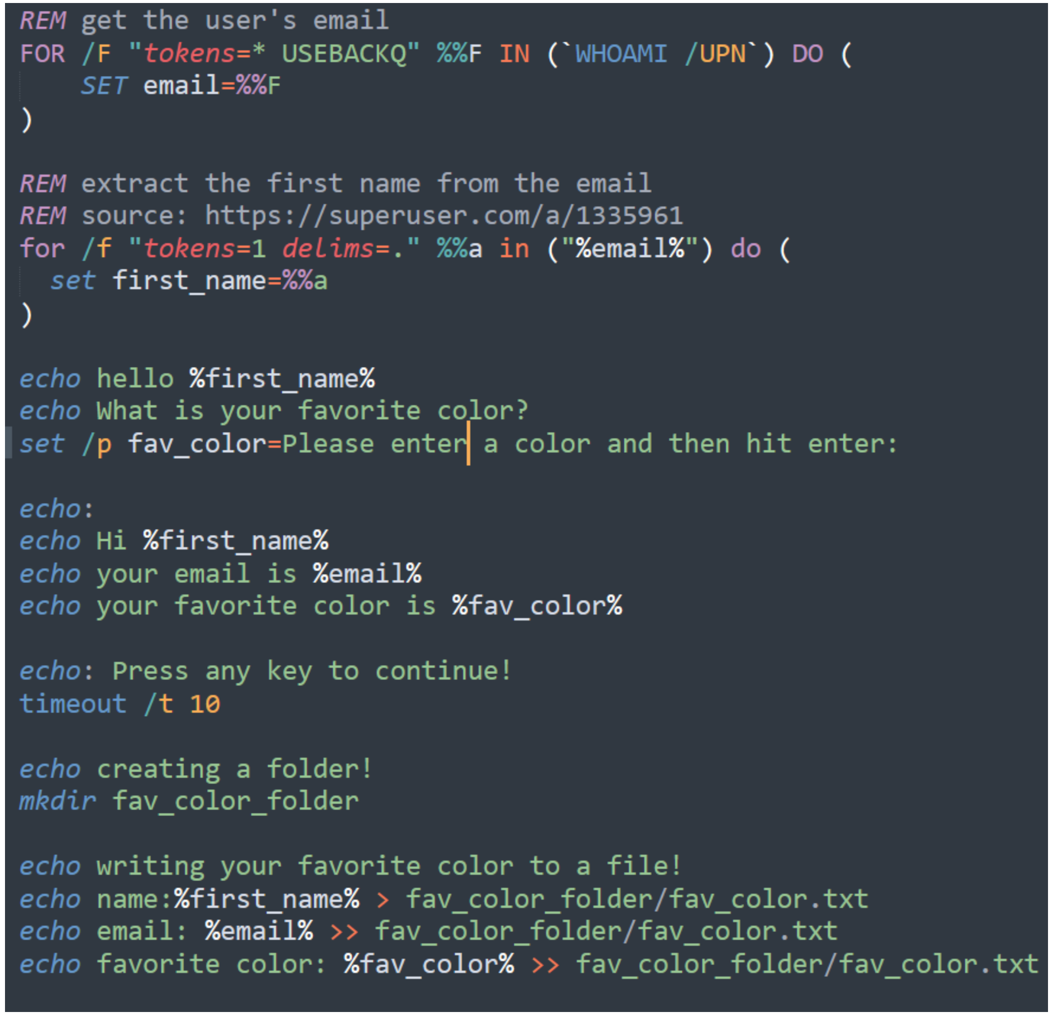
```
Echo Hello world!
Echo Hello %username%!
whoami
whoami /?
WHOAMI /UPN
set fav_color=green
echo your favorite color is %fav_color%!
dir
cd R_projects
git status
mkdir super_special_folder
rmdir super_special_folder
start "" "https://www.google.com/search?q=puppies&udm=2"
```



command prompt

# Batch Script Demo

- This can also be tried at home!
- This demo shows how the commands from the previous slide can be put together in a script
- To run the script just double click on it
- To see what’s in the script, right click then click edit
- **Be careful about running batch scripts other people wrote** (view in text editor or right click + edit first)



```
REM get the user's email
FOR /F "tokens=* USEBACKQ" %%F IN (`WHOAMI /UPN`) DO (
    SET email=%%F
)

REM extract the first name from the email
REM source: https://superuser.com/a/1335961
for /f "tokens=1 delims=." %%a in ("%email%") do (
  set first_name=%%a
)

echo hello %first_name%
echo What is your favorite color?
set /p fav_color=Please enter a color and then hit enter:

echo:
echo Hi %first_name%
echo your email is %email%
echo your favorite color is %fav_color%

echo: Press any key to continue!
timeout /t 10

echo creating a folder!
mkdir fav_color_folder

echo writing your favorite color to a file!
echo name:%first_name% > fav_color_folder/fav_color.txt
echo email: %email% >> fav_color_folder/fav_color.txt
echo favorite color: %fav_color% >> fav_color_folder/fav_color.txt
```