

Instructions on How to Run the Interactive Report

Prepared by: Brayden Tang, UBC-MDS Capstone 2020 (email: brayden.tang1@gmail.com)
Date: June 21, 2020

The easiest and fastest way to view the interactive report is to use the prepared Docker container and Git. Nothing else needs to be installed besides these two programs and the repository itself. Furthermore, the report will be viewable forever (providing you have access to the Docker image, and the repository).

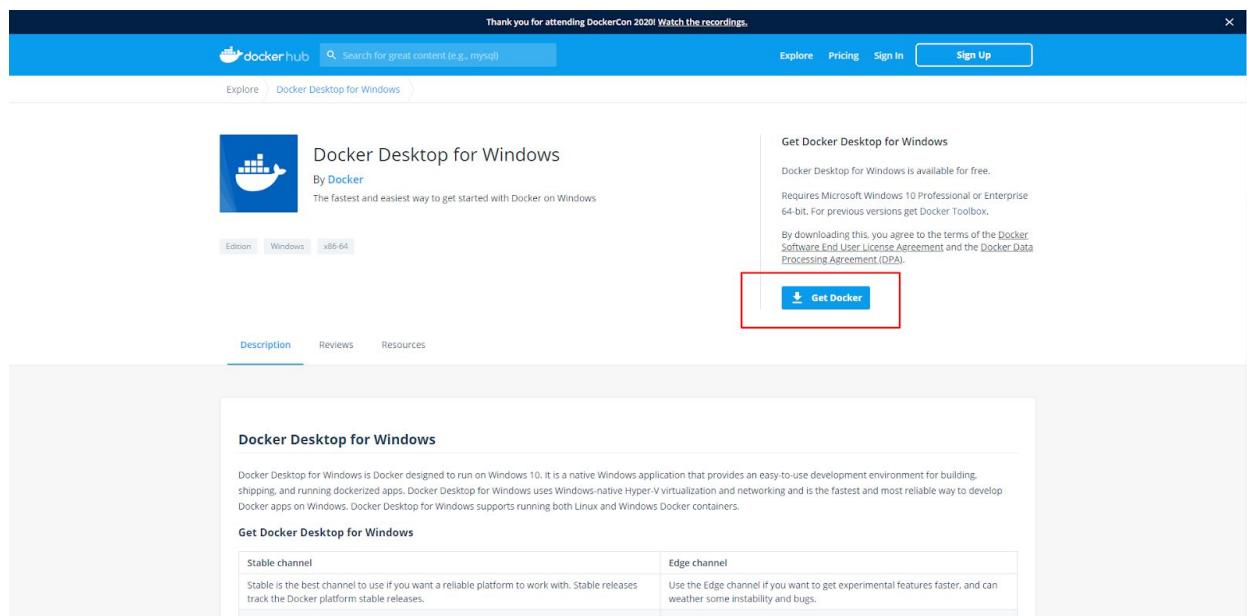
This was tested on a Windows 10 Pro Computer with no previous software installed (no Python or R).

If you already have Docker/Git, please skip to step 3).

- 1) Install Docker. For Windows, please visit this link here:

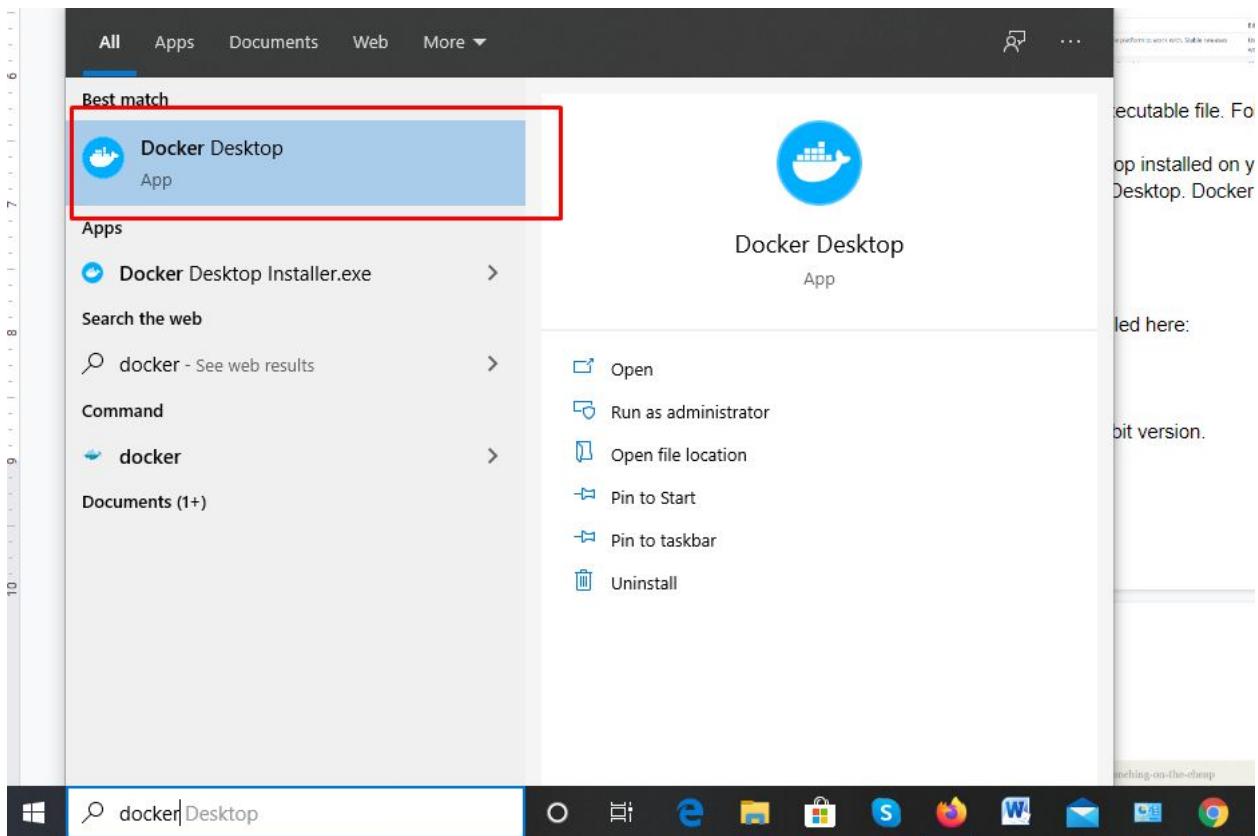
<https://hub.docker.com/editions/community/docker-ce-desktop-windows/>

Click on the blue button here (marked in red):



This will start the download of an executable file. Run this executable file and follow the resulting instructions to complete the install. Note that Docker requires Windows 10 64 bit (Enterprise, Pro, or Student editions) and Hyper-V must be enabled. Hyper-V should automatically become enabled if you follow the instructions provided when you install Docker.

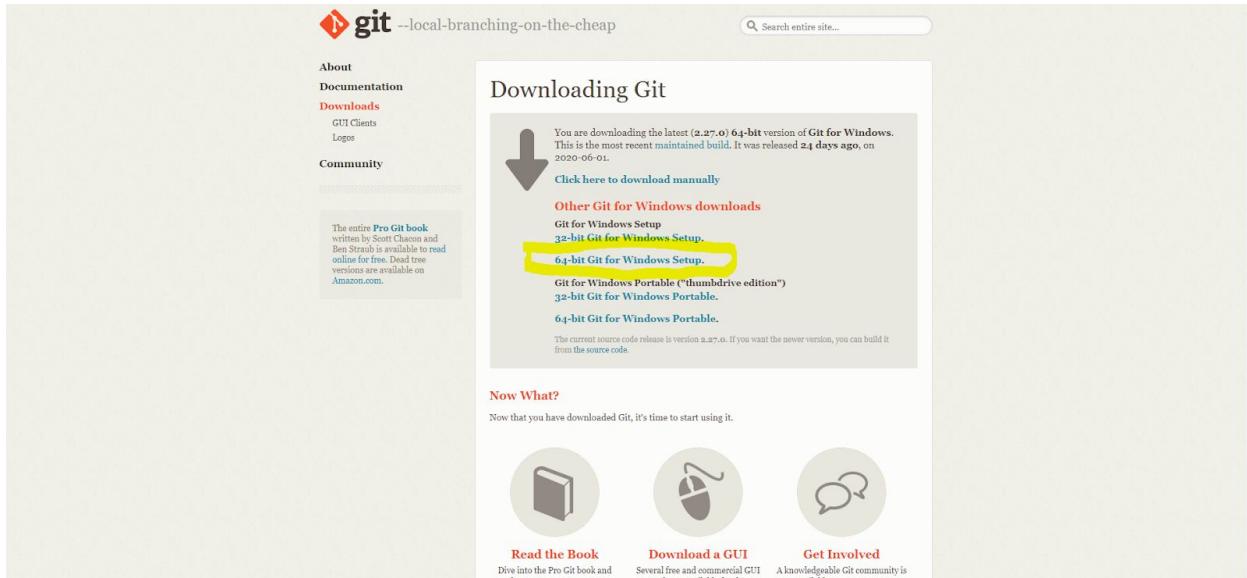
You should now have Docker Desktop installed on your computer. To verify, open the start menu in Windows and search for Docker Desktop. Docker Desktop should appear as the top match (with a blue whale as an icon):



2) Install Git. This can be installed here:

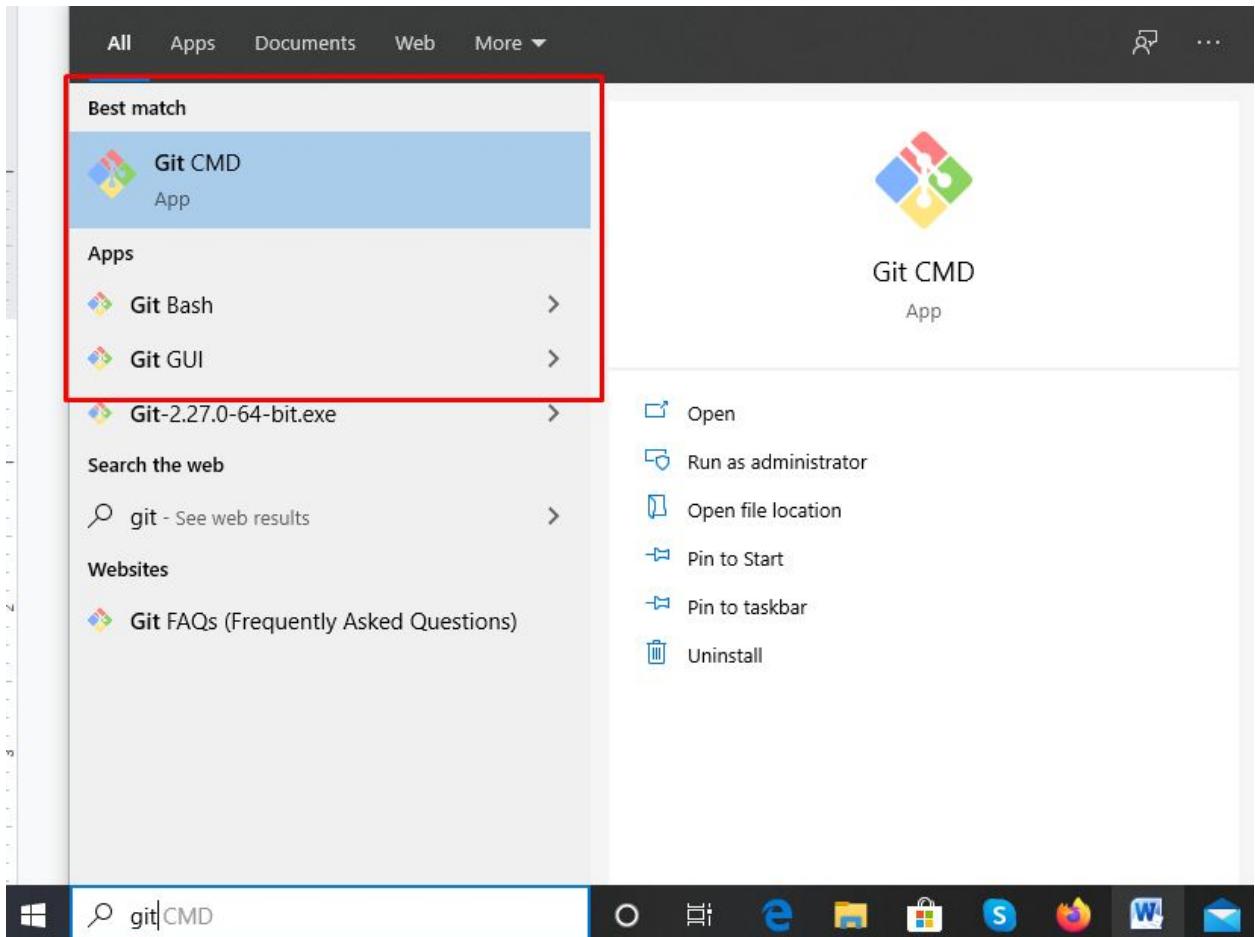
<https://git-scm.com/download/win>

You will likely have to install the 64-bit version.

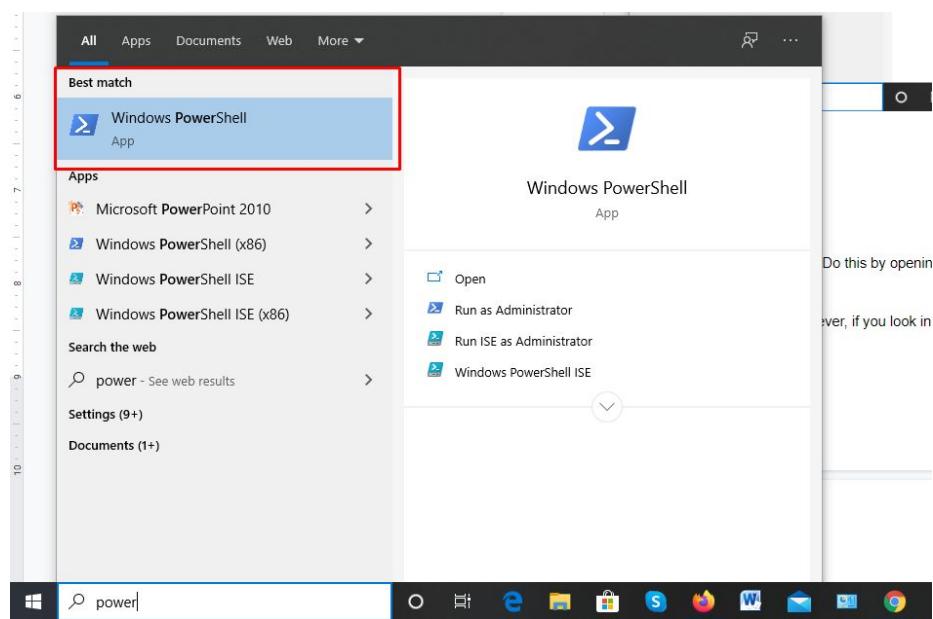


This will start the download of an executable file. Run the executable file and follow the resulting instructions to complete the install.

You should have Git installed on your computer. To verify, open the start menu in Windows and search for git. Git GUI, Git CMD, and Git Bash should all return as matches:

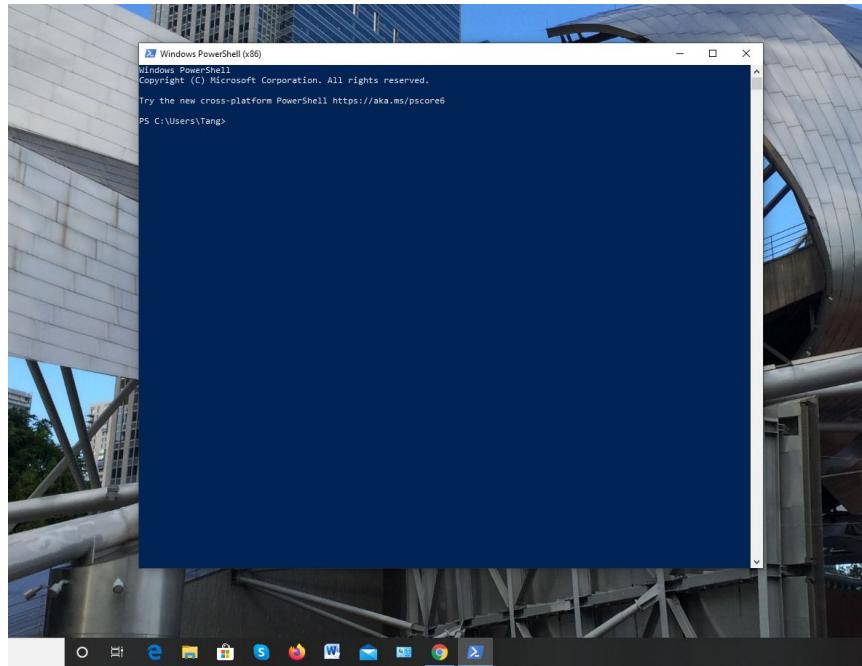


3) Next, open up Windows PowerShell:



Make sure to use Windows PowerShell and **not** Windows PowerShell ISE.

If you were successful, a terminal should have opened up:



- 4) Navigate to a folder where you wish to store the GitHub repository. For example, if you wish to store the repository on the Desktop, you have to run the command: "cd Desktop" (with no quotations) as shown below:

A screenshot of a Windows PowerShell window, identical in appearance to the previous one, but with a red rectangular box highlighting the command "cd Desktop" in the command line area. This indicates that the user has just typed this command and is awaiting its execution.

Notice that the path has changed from C:\Users\Tang to C:\Users\Tang\Desktop.

If you have a specific file path you wish to store the repository in, then you must give the full file path. For example, if you wish to store the repo in a folder inside Documents called wow, I will need to first change the directory with the following command:

```
cd C:\Users\Tang\Documents\wow
```

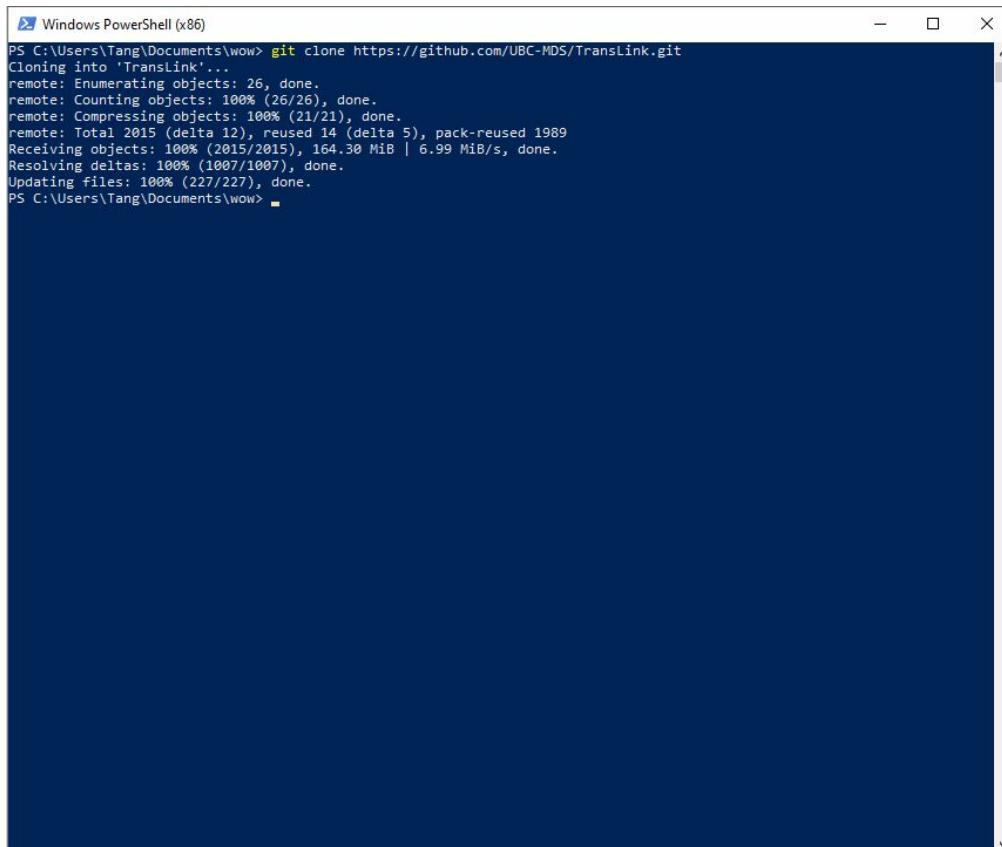
5) Next, run this command in Powershell:

```
git clone https://github.com/UBC-MDS/TransLink.git
```

This command downloads the repository to whatever directory you are currently in.

You may get a popup window asking for your GitHub credentials. This is to verify that you actually have access to the repository. Enter your GitHub username and password to proceed. Make sure you have access to the repository (consult brayden.tang1@gmail.com if you do not have access!)

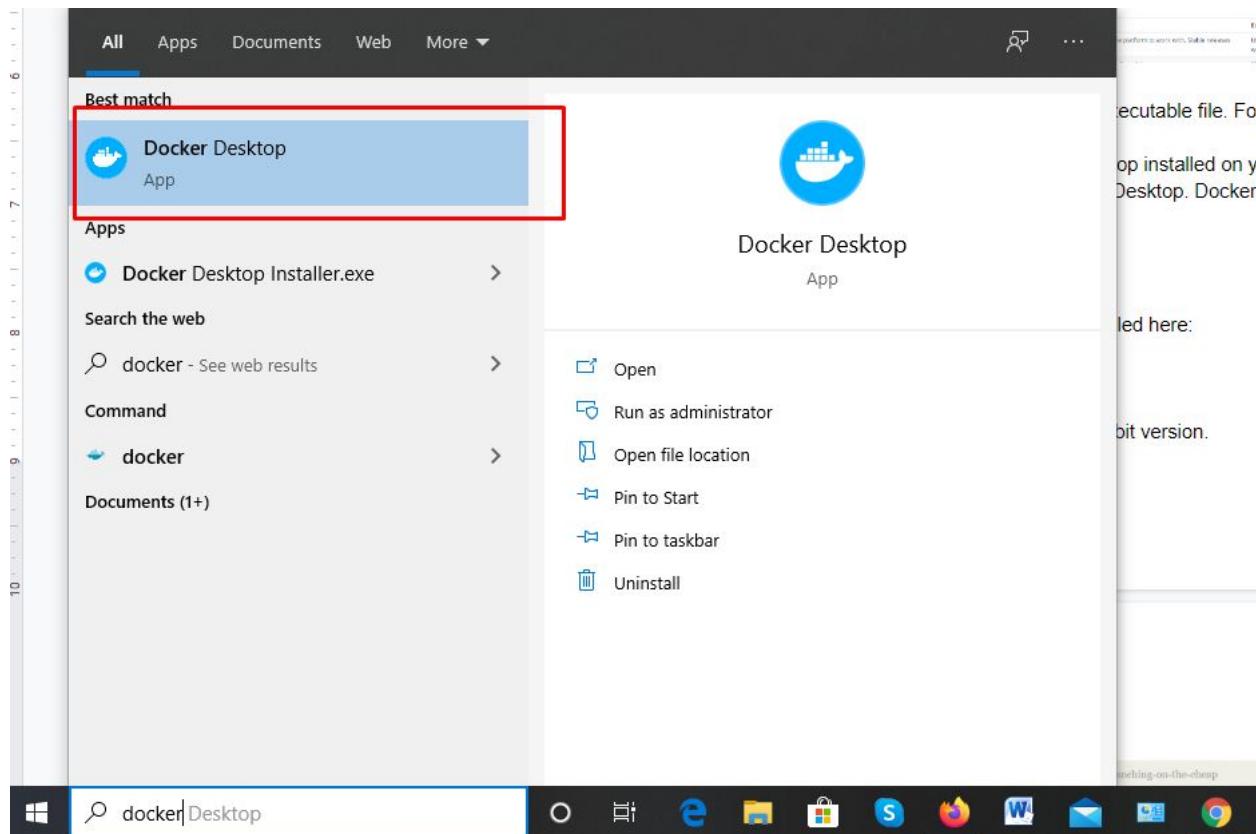
If you have successfully downloaded the repository, you should see this in your terminal:



A screenshot of a Windows PowerShell window titled "Windows PowerShell (x86)". The window shows the command "git clone https://github.com/UBC-MDS/TransLink.git" being run in the directory "C:\Users\Tang\Documents\wow". The output of the command is displayed, showing the progress of cloning the repository, including object enumeration, counting, compressing, and receiving objects, as well as resolving deltas and updating files. The process is shown to be 100% complete.

```
PS C:\Users\Tang\Documents\wow> git clone https://github.com/UBC-MDS/TransLink.git
Cloning into 'TransLink'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 2015 (delta 12), reused 14 (delta 5), pack-reused 1989
Receiving objects: 100% (2015/2015), 164.30 MiB | 6.99 MiB/s, done.
Resolving deltas: 100% (1007/1007), done.
Updating files: 100% (227/227), done.
PS C:\Users\Tang\Documents\wow>
```

- 6) Before we actually run any commands, we need to ensure that Docker has access to the local files in the repository you just downloaded. To do this, first start Docker Desktop:

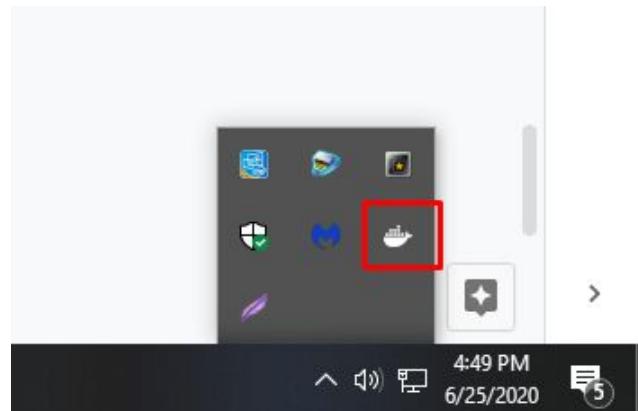


No application will actually open but this will start Docker Desktop in the background.

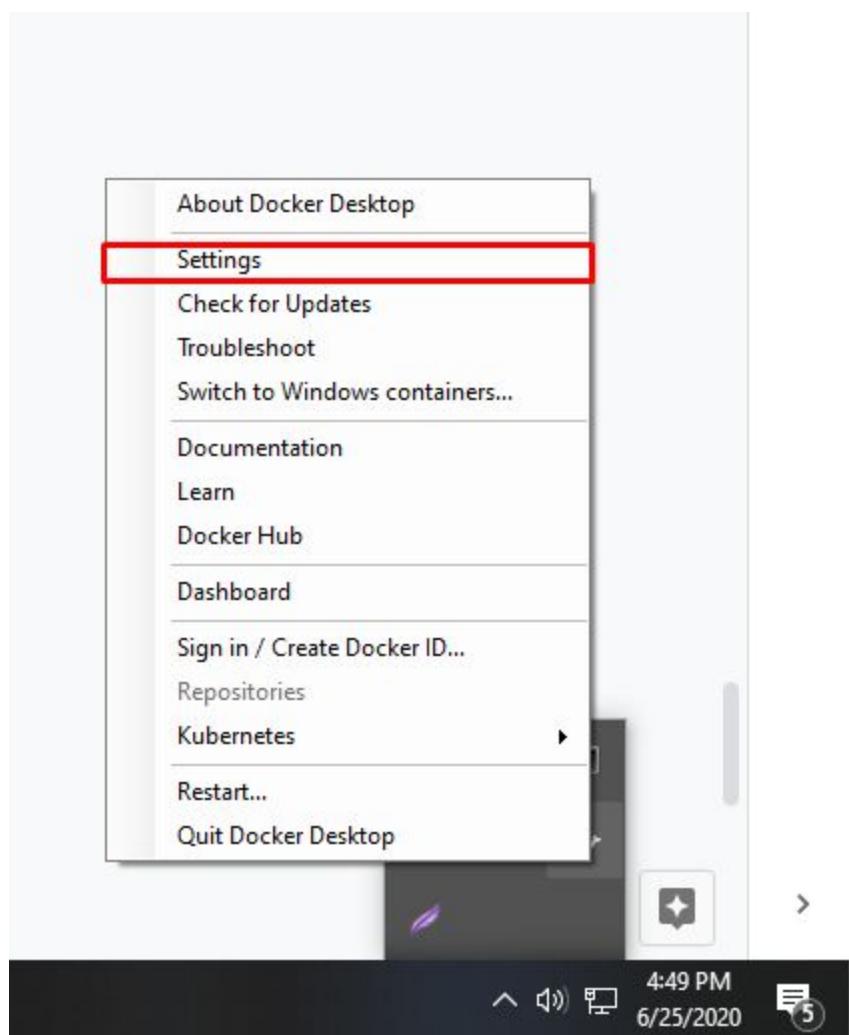
To open up the application, click on the “^” on the taskbar in the bottom right corner near the time (marked in red):



Click on the Docker symbol that appears:



Next, click on Settings:



An application should now have opened (see below). Click on Resources → FILE SHARING.

The screenshot shows the Docker Desktop settings window. The left sidebar has tabs for General, Resources (which is selected and highlighted with a red box), Docker Engine, Command Line, and Kubernetes. The main area is titled 'General' with the sub-section 'Adjust how Docker Desktop behaves according to your preferences.' It contains several configuration options with checkboxes:

- Start Docker Desktop when you log in
- Automatically check for updates
- Expose daemon on tcp://localhost:2375 without TLS

Exposing daemon on TCP without TLS helps legacy clients connect to the daemon. It also makes yourself vulnerable to remote code execution attacks. Use with caution.
- Use the WSL 2 based engine (requires Win 10 build 19018+)

WSL 2 provides better performance than the legacy Hyper-V backend. [Learn more](#).
- Send usage statistics

Send error reports, system version and language as well as Docker Desktop lifecycle information (e.g., starts, stops, resets).

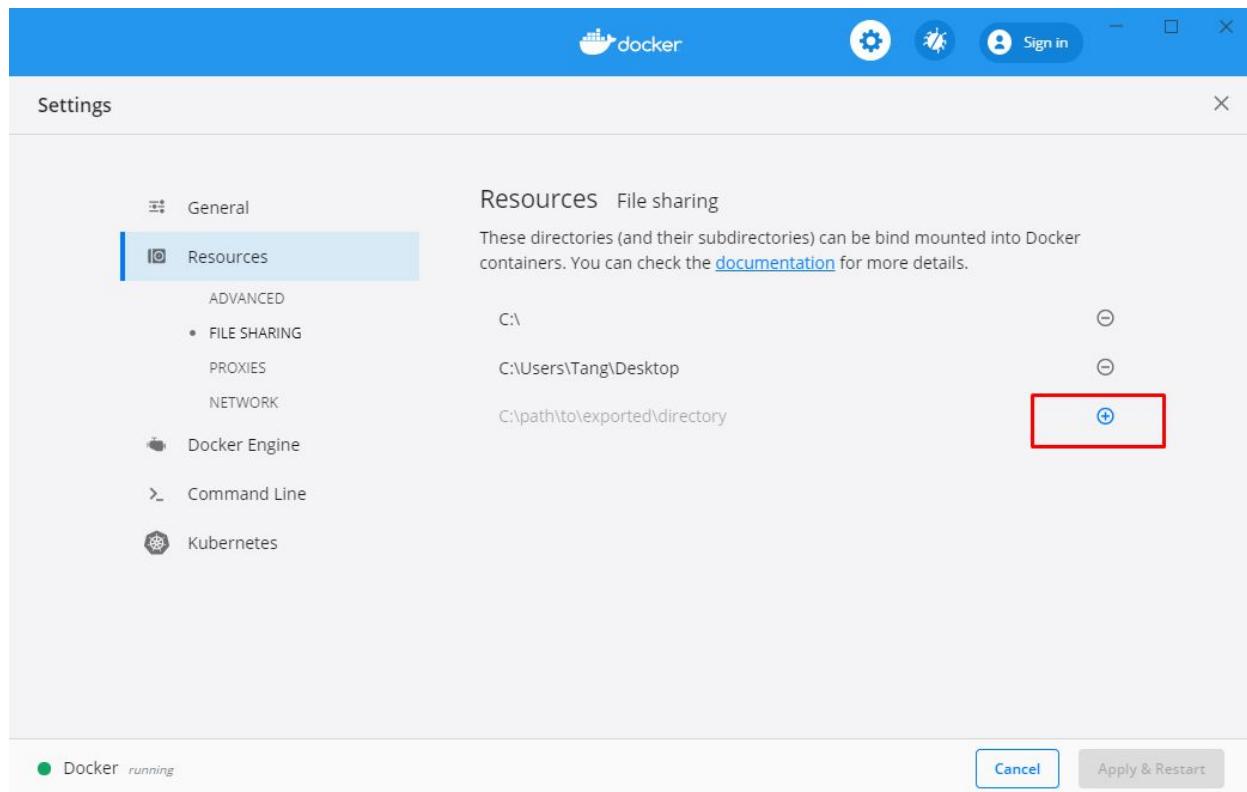
At the bottom, there is a note: "Discover experimental features. [Switch to the Edge version](#)." The status bar at the bottom left says "Docker running".

This screenshot shows the same Docker Desktop settings window, but the 'Resources' tab is selected (highlighted with a red box). The left sidebar includes ADVANCED (with FILE SHARING selected and highlighted with a red box), PROXIES, NETWORK, Docker Engine, Command Line, and Kubernetes. The main area is titled 'Resources' with the sub-section 'File sharing'. It displays a list of bind mounted directories:

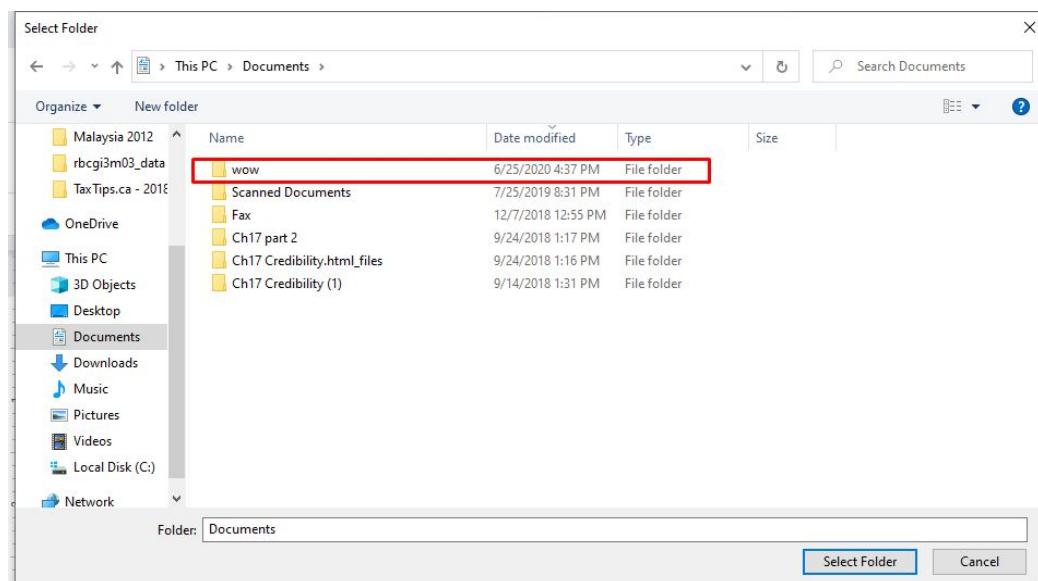
Path	Action
C:\	⊖
C:\Users\Tang\Desktop	⊖
C:\path\to\exported\directory	⊕

The status bar at the bottom left says "Docker running".

- 7) We need to allow file sharing between local files stored on your computer and the virtual Docker container. To do this, click on the blue “+” sign as shown below:

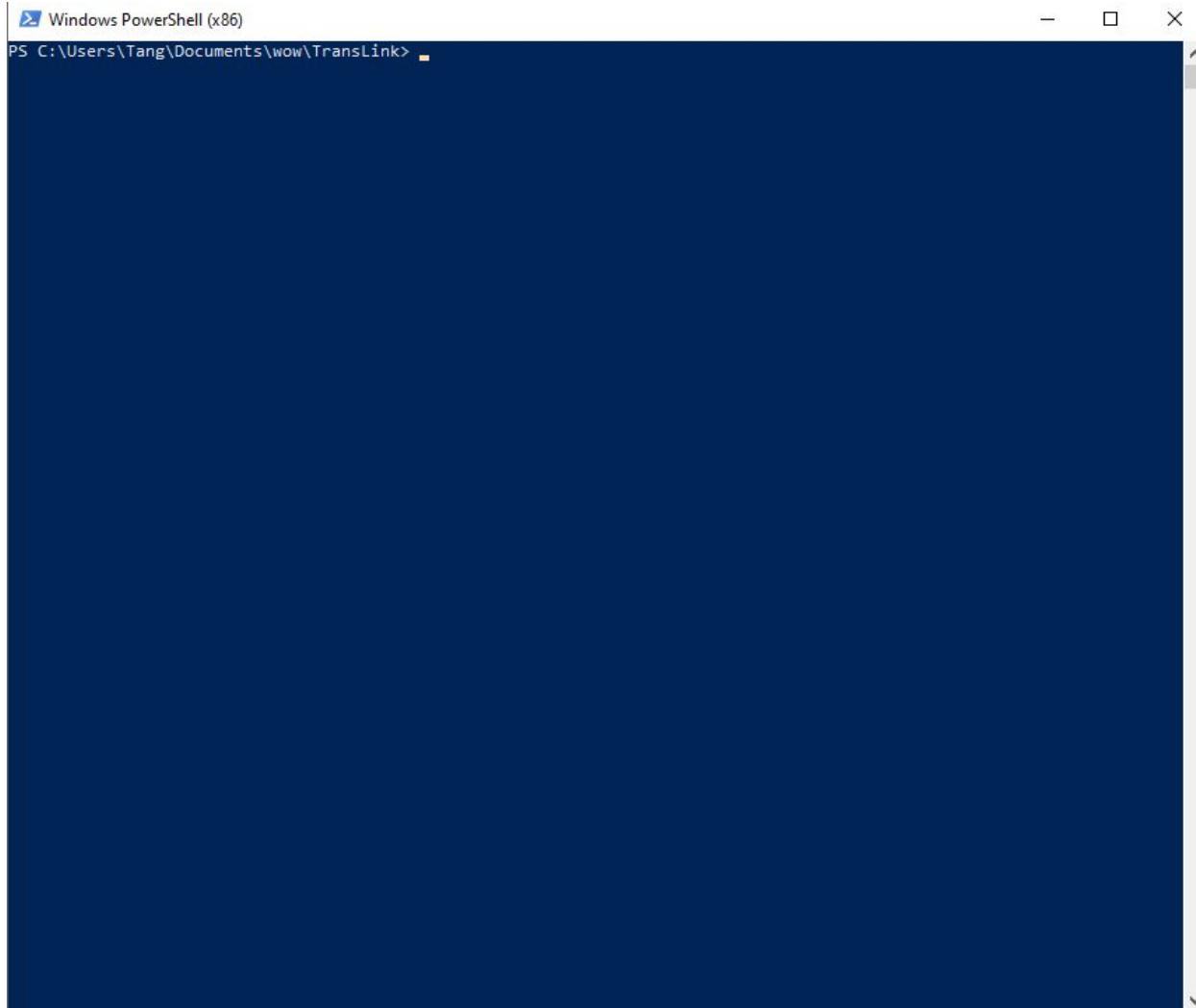


This will open a file directory where you now need to select the folder in which you stored the repository in step 4). For example, I cloned the repository in the folder "C:\Users\Tang\Documents/wow" and therefore, I must select this folder:



Selecting the most general path will also work. For example, while I stored the repository in “C:\Users\Tang\Documents\wow\TransLink”, I can simply select the “Documents” folder or the folder “wow” and not specifically the “TransLink” folder. Docker will know to trust all subfolders within the more general folder - it doesn’t matter.

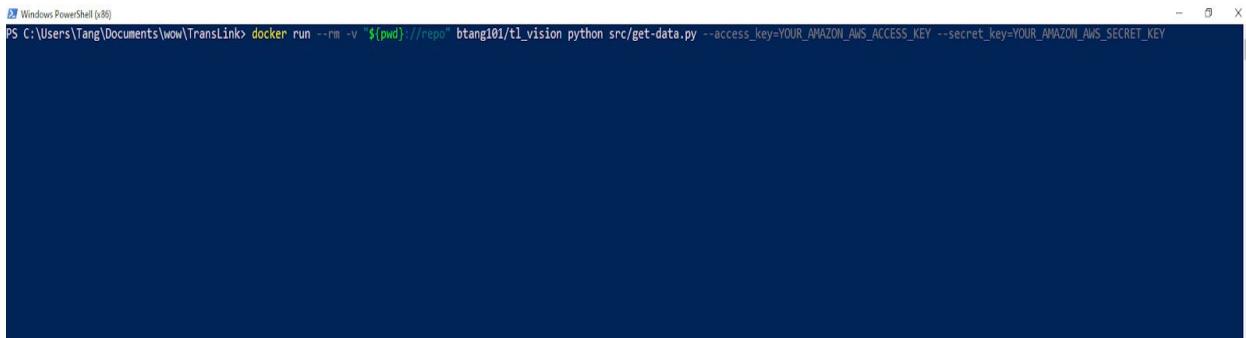
- 8) Switch back to PowerShell, and navigate to the directory where you stored the repository (if you aren’t already there!). For example, I stored the repository in “C:\Users\Tang\Documents\wow\TransLink”:



A screenshot of a Windows PowerShell window titled "Windows PowerShell (x86)". The window shows the command prompt "PS C:\Users\Tang\Documents\wow\TransLink>" followed by a blank line for input. The window has a standard blue title bar and a white body area.

- 9) In PowerShell, **while you are in the specific repository folder “TransLink” as in the above photo (this is “the root” of the repository)**, run the following command (all on one single line as shown below!):

```
docker run --rm -v "${pwd}://repo" btang101/tl_vision python src/get-data.py  
--access_key=YOUR_AMAZON_AWS_ACCESS_KEY  
--secret_key=YOUR_AMAZON_AWS_SECRET_KEY
```

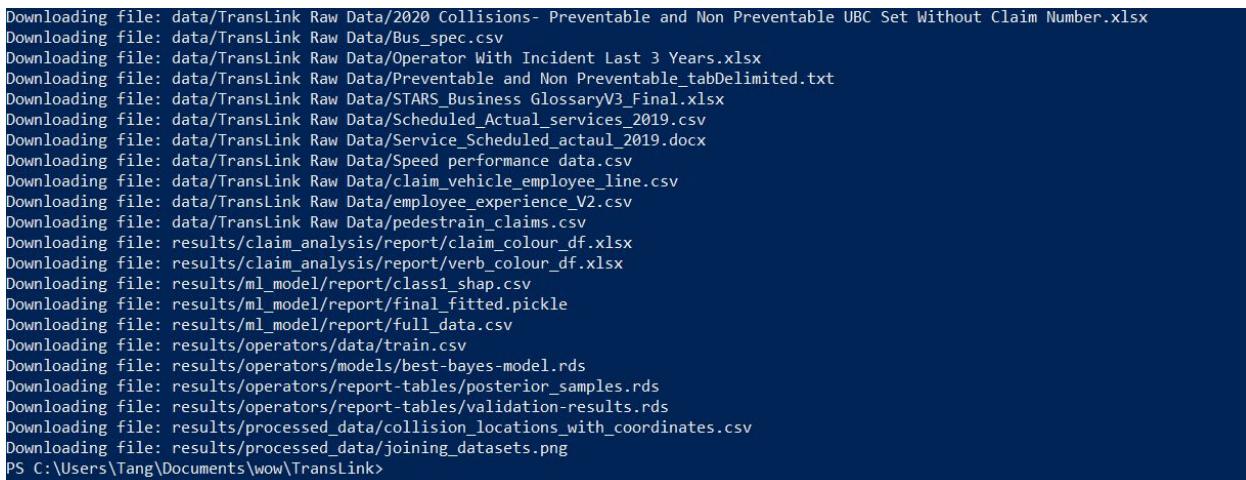
A screenshot of a Windows PowerShell window titled "Windows PowerShell (x86)". The command being run is: PS C:\Users\Tang\Documents\wow\TransLink> docker run --rm -v "\${pwd}://repo" btang101/tl_vision python src/get-data.py --access_key=YOUR_AMAZON_AWS_ACCESS_KEY --secret_key=YOUR_AMAZON_AWS_SECRET_KEY

Note that YOUR_AMAZON_AWS_ACCESS_KEY and YOUR_AMAZON_AWS_SECRET_KEY must be the access key and secret key that give you access to the Amazon S3 bucket. Saeed probably knows what keys you should use since he is the one who has been uploading all the data for us to S3.

This command downloads all of the contents in the S3 bucket required to run the interactive report. This may take a couple minutes. If this is your first time ever running this command, your computer will also download the Docker container automatically which is about 2 GB.

Note that once you run this command once, you do not need to rerun this command ever again (unless you delete these downloaded files from your computer, or if you want to download the files again for some reason).

10) Once the command has finished, you should see this:

A screenshot of a terminal window showing the output of the command execution. The output lists numerous files being downloaded from an S3 bucket, including various CSV, XLSX, and pickle files related to collision data, vehicle performance, and machine learning models.

Next, run the command (again all on one line like shown below!):

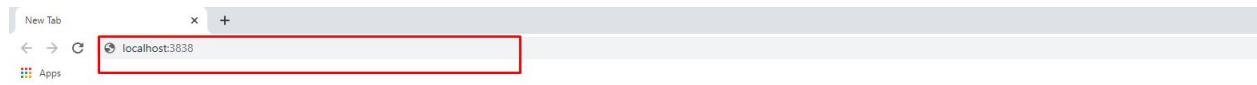
```
docker run --rm -p 3838:3838 -v "${pwd}://repo" btang101/tl_vision Rscript -e  
"rmarkdown::run('doc/interactive-report/interactive-report.rmd', shiny_args = list(port = 3838,  
host = '0.0.0.0'))"
```

If successful, you should see this message:

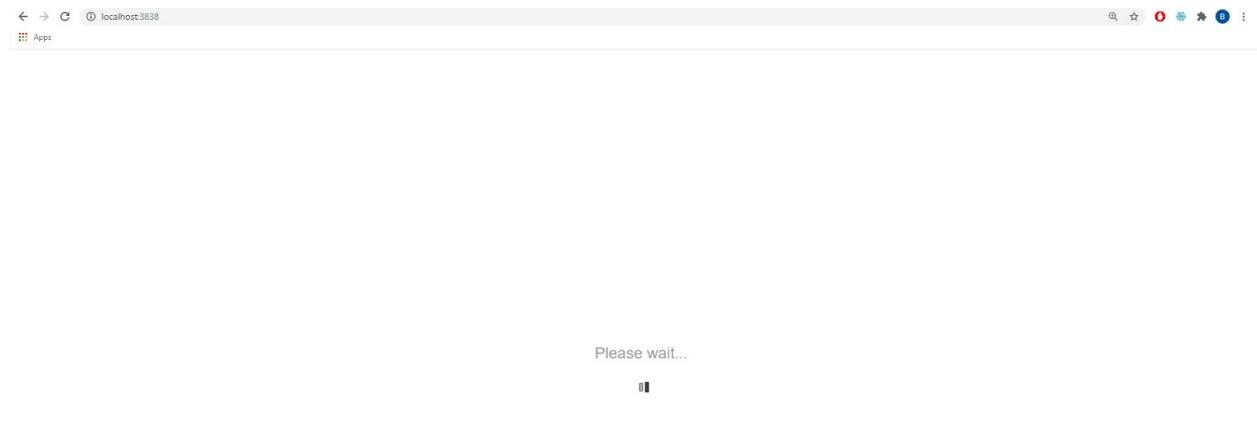
```
PS C:\Users\Yang\Documents\wow\transLink> docker run --rm -p 3838:3838 -v "${pwd}://repo" btang101/tl_vision Rscript -e rmarkdown::run('doc/interactive-report/interactive-report.rmd', shiny_args = list(port = 3838, host = '0.0.0.0'))  
Loading required package: shiny  
Listening on http://0.0.0.0:3838
```

11) In any web browser, navigate to this address like in the figure below:

localhost:3838



You should now see this screen here:



The report should take around 45 seconds to start.

If successful, you should see this:

Vision Over Incidents and Claims - An Analysis

Brayden Tang, Merve Sahin, Simardeep Kaur, Xugang Zhong

29/06/2020

Introduction

With the largest transit service area in Canada, TransLink is operating more than 245 bus routes and 79 kilometers of rapid transit to meet the transportation needs of 2.5 million people in Metro Vancouver as of the end of 2018 (TransLink 2018). Legislation requires TransLink to carry a \$1 million per occurrence liability policy on each of its revenue vehicles and a \$200,000 per occurrence liability policy on each of its non-revenue vehicles. Since 2014/2015, the premium paid to ICBC has increased by over 200% to cover onboard passenger injuries, cyclist injuries, pedestrian injuries, and losses from collisions with third party vehicles. For at-fault physical damage losses to its vehicles, the premium paid to its own captive insurance company has increased by 33%.

In response to soaring insurance costs and road safety concerns, TransLink has asked us to analyze key variables of interest that may be predictive of bus incidents. These variables include an analysis of the operators involved, the impact of bus characteristics (if any), and the effect of weather, time, location, and line number on the likelihood of an incident occurring. Finally, TransLink has also asked us to analyze the types of claims that are occurring - in particular, if there are common types of claims per location and if particular locations yield large paid costs.

A variety of statistical and machine learning methods are employed in this report to address these questions. As a result, the report is divided into six sections:

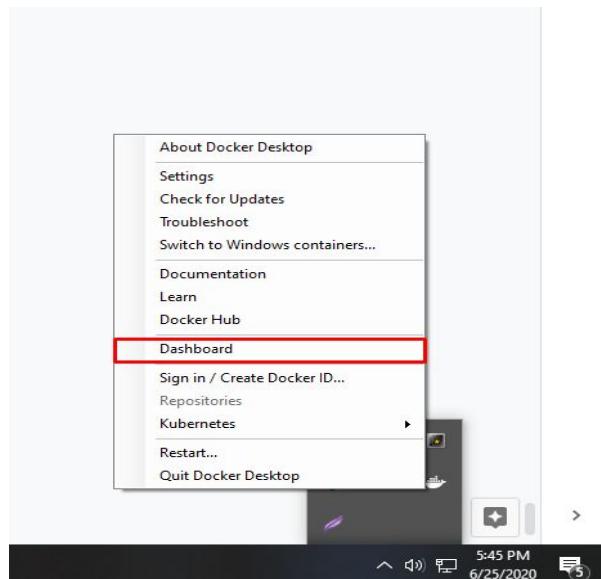
- 1. Executive Summary
- 2. Predictive Power of Location and Operator Experience
- 3. A Combined Analysis of All Factors Using Machine Learning
- 4. Common Types of Claims by Location
- 5. Assessment of Claims Costs by Location
- 6. Future Analysis

Executive Summary Predictive Power of Location and Operator Experience A Combined Analysis of All Factors Using Machine Learning

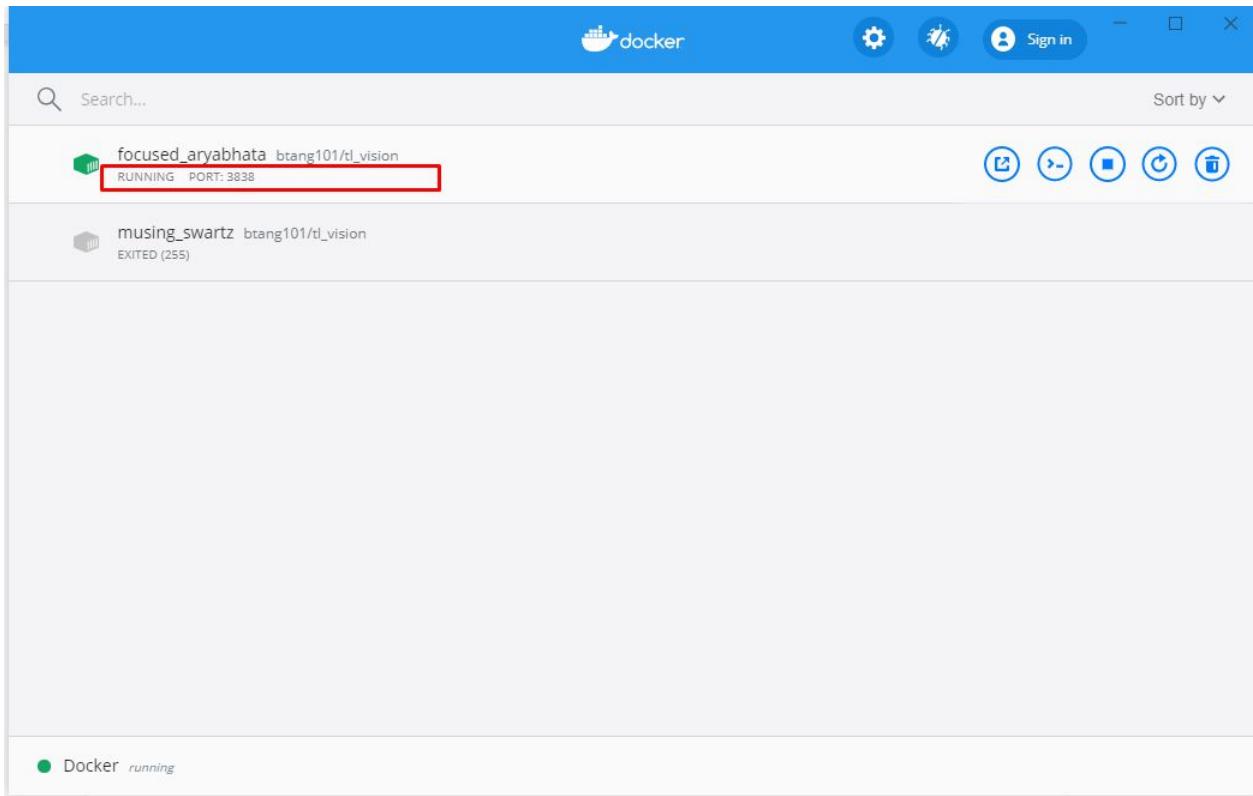
Common Types of Claims by Location Assessment of Claims Costs by Location Future Analysis References

We recommend viewing the report in full screen (press F11). Furthermore, zooming in/out by holding Ctrl and pressing -/+ , depending on your screen, might be useful.

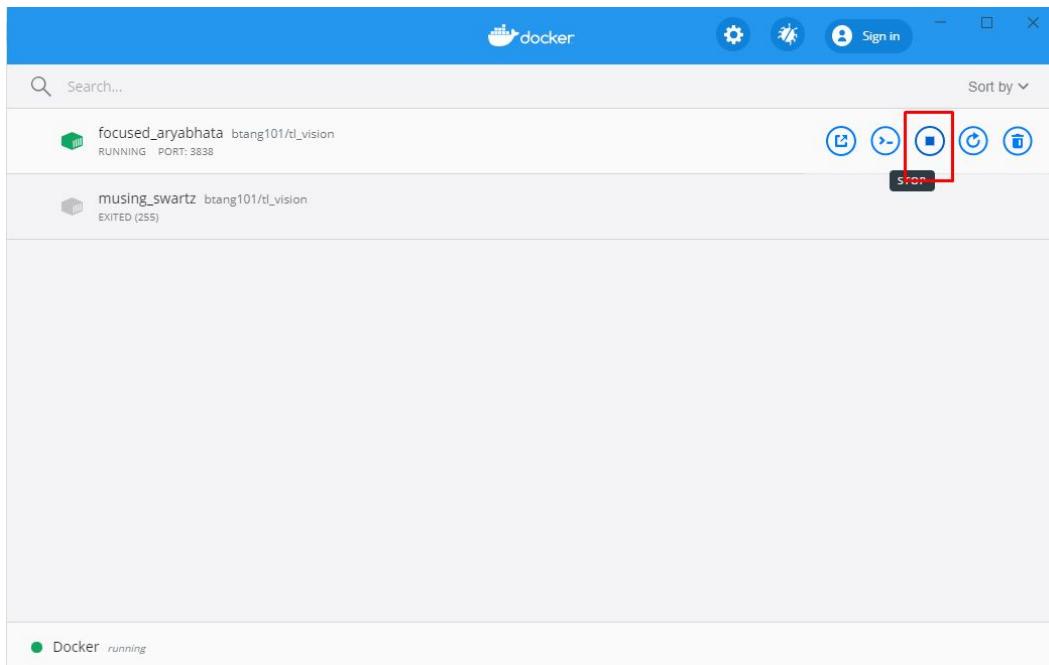
- 12)** Once you are done with the report, in PowerShell (where the report is being run) press Ctrl + C to terminate the process.
- 13)** Next, to stop the Docker container from using CPU resources, reopen Docker Desktop as in step 6), but this time navigate to the Dashboard:



You should see Docker Desktop open once again. Next, find the container that is running on port 3838. In my case:



Next, click on the stop button to stop the container:



- 14) If you wish to view the report again, there is no need to rerun all steps. Instead, repeat from step 10) and onwards.