



Michael Gaunt mike.gaunt@wsp.com

2020-11-11

Table of Contents

Overview	1
Files and Directories.....	2
Root Level Files.....	2
Sub Folders	4
Application Build Process.....	6
data_source_list.xlsx Description	7
Example: Adding Data	11
General.....	12
Notes	12
Locally Viewing the Application	12
Redeploying the Application	13































Overview

This document serves as a guide to the UHSGT dashboard back-end . This guide is not intended to teach how to use R but to describe the main components, files, directories, update methods, and deployment strategies regarding the application

The UHSGT dashboard was made using the R language and Shiny dashboard framework. Both R and Shiny require specific domain knowledge to implement; however, this application was built in a way to minimize

Files and Directories

Please see the below image that details the dashboards files. The ensuing bullet list describes the items and folders as seen in the image.

Name	Date modified	Type	Size
 Maintenance-Manual	2020-12-11 9:34	Microsoft Word D...	58 KB
 Maintenance Manual	2020-12-11 9:34	RMD File	7 KB
 global	2020-11-29 22:54	R File	7 KB
 ui	2020-11-29 22:51	R File	18 KB
 server	2020-11-29 22:51	R File	38 KB
 data_mapper	2020-11-29 22:50	RHISTORY File	18 KB
 data_mapper	2020-11-29 22:49	R File	11 KB
 file52901af5fba.Rprof	2020-11-25 15:21	R Workspace	20,073 KB
 file52901af5fba.Rprof	2020-11-23 11:49	RPROF File	109 KB
 data_scripts	2020-11-23 10:28	R File	16 KB
 Operation_Manual	2020-11-23 9:54	Adobe Acrobat D...	289 KB
 Operation_Manual	2020-11-23 9:54	Microsoft Word D...	746 KB
 Operation_Manual	2020-11-23 9:54	RMD File	17 KB
 Operation_Manual	2020-11-23 9:53	Microsoft Edge H...	1,899 KB
 data_source_list	2020-11-22 17:44	Microsoft Excel W...	35 KB
 MPO_resource_table	2020-11-16 16:41	Microsoft Excel W...	19 KB
 ca_parkride_raw	2020-11-16 10:41	Microsoft Excel W...	12 KB
	2020-11-11 15:43	Text Document	1 KB
 US_House_Reps	2020-09-28 21:04	Microsoft Excel W...	61 KB
 docs	2020-12-09 15:58	File folder	
 R	2020-11-29 22:50	File folder	
 mapedit_tmp	2020-11-23 21:47	File folder	
 handlers	2020-11-23 10:11	File folder	
 map_ready	2020-11-22 17:52	File folder	
 application_shapefiles	2020-11-17 14:25	File folder	
 corrdior_buffer	2020-11-17 13:57	File folder	
 shapefiles_raw_cache	2020-11-16 15:08	File folder	
 www	2020-11-11 15:33	File folder	
 ca_census_cache	2020-11-11 10:50	File folder	
 rsconnect	2020-08-26 16:49	File folder	

Root Level Files

- Root Level *.R files
 - global.R
 - Initializes the R environment
 - Loads the required packages
 - If you are doing any maintenance on the application or redeploying the application you will need to make sure all of

these libraries are installed on your machine and loaded for your session

- Loads custom functions needed for the application
 - It calls two files at the end of the script after initializing the environment
 - data_mapper.R and data_scripts.R
 - data_mapper.R processes/preps the data when the application starts and data_scripts.R contains static strings that are called by the application modals and information buttons
- ui.r
 - This is the R script that contains the UI logic for the application
- server.r
 - This is the R script that contains the server logic for the application
- Data_mapper.R
 - Called by the global.R script
 - It preps the shapefile data in the map_ready folder for the shiny app
 - It makes objects (maps, plots, tables) used in the shiny application
- Data_scripts.R
 - Contains static text that is called by the shiny-file
 - Contains items like modal text or text for in app help buttons
- Root Level Excel Spreadsheets
 - General: These spreadsheets are used to build certain objects that are needed to run the application
 - data_source_list.xlsx
 - This spreadsheet defines all data layers and map processing operations that are performed when the application is ran
 - Augmenting this spreadsheet will create data new or edit existing data layers
 - This sheet is called by the data processing scripts (the scripts located in the *handlers* folder) and the application scripts (data_mapper.R and data_scripts.R)
 - MPO_resource_table.xlsx
 - Contains all the information for US MPOs and Vancouver Metro
 - This tabular data is merged spatial boundaries for the regional planning jurisdictions
 - This sheet is called by the data processing scripts (the scripts located in the *handlers* folder) and the application scripts (data_mapper.R and data_scripts.R)
 - ca_parknride_raw
 - Contains data depicting the park and ride locations in Vancouver Metro area
 - This data was gathered by hand and its shape-file was made by hand
 - US_House_Reps.xlsx

- Contains tabular data for US state house and senate representatives
 - This tabular data is merged with spatial boundaries for US house and senate districts
 - This data was gathered by hand
 - This sheet is called by the data processing scripts (the scripts located in the *handlers* folder)
- Root Level Maintenance-Manual.* and Operation_Manual.* Files
 - The .Rmd files are the editable manuals that create word documents when knitted
 - Is is a Markdown language that can also contain executable R code
 - Please see this site for any additional information:
<https://rmarkdown.rstudio.com/>

Sub Folders

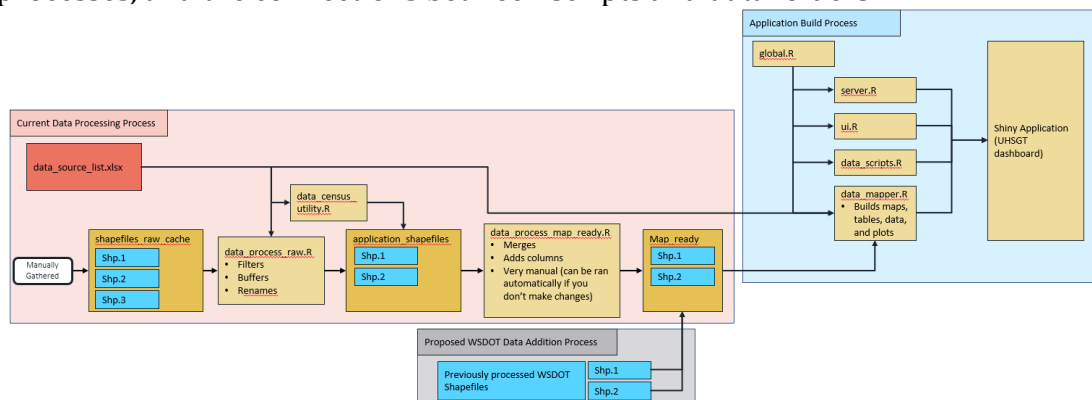
- application_shapefiles folder
 - Contains all the shape-files that have been initially processes
 - These files have been buffered and filtered
 - The files located in this folder are further filtered by data_process_map_ready.R
 - This folder is specific to the data processing process executed by the developer (mike.gaunt.404@gmail.com)
 - If you add data to the application you should process it elsewhere and add it to the map_ready folder
- corridor_buffer folder
 - Contains a singular shapefile that is used to buffer the raw shape-files that are contained in shapefiles_raw_cache
 - The buffer was created manually by tracing I5 from Eugene, OR to Vancouver, BC.
- docs folder
 - Contains auxiliary files
 - The application does not use anything in this folder to build or run the application
- handlers folder
 - General: Contains R scripts used to obtain or process raw shape-files and to prepare them for the application - they are not directly called by the application
 - Each R script contains a description of what it does in the code itself
 - data_census_utility.R
 - Performs all operations related to acquiring and processing US and CA census data from their respective APIs
 - This script uses data_source_list.xlsx to define the queries
 - The US data query information comes from the “tidycensus” tab
 - Information regarding this package (tidycensus) and the API can be found here: <https://github.com/walkerke/tidycensus>

- You will need an API key to run or perform any development of your own
- The CA data query information comes from the “ca_census” tab
 - Information regarding this package (tidycensus) and the API can be found here: <https://cran.r-project.org/web/packages/cancensus/vignettes/cancensus.html>
 - You will need an API key to run or perform any development of your own
- data_process_raw.R
 - Uses the information contained in data_source_list.xlsx to process shape-files located in shapefiles_raw_cache
 - Primarily buffers the raw shape-files and removes unneeded data attributes
 - After processing, the shape-files are written to the application_shapefiles folder
 - This is not called by the shiny file and is ran manually
 - Note: running this file is only required if you want to reprocess files that were included in the initial release, you don not need to this script to process files that have already been processed outside of R – already processed files can be dropped into the map_ready folder to avoid processing
- data_data_process_map_ready.R
 - This script further processes the shape-files that are located in the application_shapefiles folder
 - This script performs any and all bespoke processing required by each data
- Example: Adding attributes based of existing attributes, modify existing attributes, merging data to existing shape-files, merging shape-files to create new data layers, etc .
 - This script performs the final processing, once processed the shape-files are ready to be included in the map and written to the map_ready folder
 - This is not called by the shiny file and is ran manually
 - Note: running this file is only required if you want to reprocess files that were included in the initial release, you don not need to this script to process files that have already been processed outside of R – already processed files can be dropped into the map_ready folder to avoid processing
- scratch.R
 - This script was used for development
 - It is not used in the shiny script
- map_ready folder
 - This folder contains all the shape-files that have gone through final processing and that will be used in the dashboard
 - Any shape-files that have been processed outside of R and that are to be included in the shiny application should be placed here

- **mapedit_tmp folder**
 - Contains dummy shape-files that are rewritten when the user generates new features and would like to download them
 - When they are downloaded the shiny app writes those features here and zips this folder for the user
- **Rsconnect folder**
 - This contains information regarding the deployment of the app on shiny-apps.io
 - It is currently linked to XXXXXXXXXXXXXXXXXXXXX
 - You will need to update this folder if it is deployed elsewhere
- **shape-files_raw_cache folder**
 - Contains all the raw shape-files that are used and were potentially going to be used for the dashboard
 - `data_process_raw.R` uses the information contained in `data_source_list.xlsx` to pull data from this location and perform the initial processing
 - The excel spreadsheet `data_source_list.xlsx` defines which shape-files are taken from this folder and processed
 - This folder is specific to the data processing process executed by the developer (mike.gaunt.404@gmail.com)
 - You will likely never have to perform the data processing process that queries data from this folder
- **www folder**
 - This is a standard folder that is used to hold images, css, or java script files that will be used by the shiny application
 - This folder currently contains images and html scripts used by the application

Application Build Process

Please see the below image that details the dashboards build process. This image provides a high-level overview of the overall process flow used to build the application, the sub-processes, and the connections between scripts and data folders.



data_source_list.xlsx Description

- General Notes:
 - There are two major processes that make up the current application build process - they are the *data processing* and *application build* processes
 - The former is the process that takes raw shape-files and processes them to be included into the application
 - The latter takes the processed shape-files (deemed *map ready*) and uses them to build the application along with the server and UI logic
 - An additional process has been included in this image to represent potential future data layer additions to the map
 - The data processing process shown above is specific to the files located in *shapefiles_raw_cache* and their particular data cleaning needs
 - Data should be processed using other software or a new r script and then placed in the *map_ready* folder
 - data_source_list.xlsx is the major driver for many of the data processing processes and is queued by many of the scripts that both process the data and that build the application
 - The format for each spreadsheet and the cell values are very important
 - The contents of the cell values will trigger different processes throughout the code
- Detailed look at data_source_list.xlsx
 - This spreadsheet has three tabs - tidycensus, ca_census, and manual
 - tidycensus tab
 - You will only need to make changes to this spreadsheet if you want to include/exclude more/less variables in the US census map layer
 - Each row is a different census variable query
 - There are many columns that record different types of meta-data for each census variable (row)
 - The columns with the green accented headers are the only meta-data that is required to obtain and build the shapefiles, they are:
 - boundary: tells the API at what geographic level to obtain the data at
 - variable: unique ID for each census variable
 - var_name: a unique name to call the variable other than it's ID
 - name: the category that the variable falls under
 - processed_name: the name of the folder that the shape-file will be written, the layers' names, and the base map layer name it will have in the map
 - Variables with the same processed_name will be grouped into the same shape-file

- notes: additional information that is associated with a particular layer
- group: a variable used for indexing in the code, should either be Political or Census (case sensitive)
- src_url: indicates the source for the data, this items in this spreadsheet should point to the same *tidycensus* source
- selection: contains a string of the raw column names that are written when the shape-file is first written to *application_shapefiles*
 - The esri format has a limit on column name lengths and these column names are a result process
 - The string needs to contain all the column names included in the shapefile
- selected_new: A string of the corrected *selection* names that are human readable

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	boundary	src	year	state	variable	var_name	geometry	name	processed_name	estimate	notes	processed_map	acsr	group	src_url	selection		
1	congressional district	acs	2018	NULL	801003_001	Constituents	Y	US_Congressional_Districts	US_Congressional_Districts	estimate	This data is Y	X	Party	Political	https://www.NAME.Congress			
2	state legislative district (lower chamber)	acs	2018	OR	801003_001	Constituents	Y	US_State_Legislature_Lower	US_State_Legislature_Lower	estimate	This data is Y	X	Party	Political	https://www.NAME.Congress			
3	state legislative district (upper chamber)	acs	2018	OR	801003_003	Constituents	Y	US_State_Legislature_Upper	US_State_Legislature_Upper	estimate	This data is Y	X	Party	Political	https://www.NAME.Congress			
4	tract	acs	2018	S OR	51901_C01_001	Total Population (20-64yrs)	Y	US_Employment	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
5	tract	acs	2018	S OR	51901_C01_008	Population at or Below Poverty	Y	US_Employment	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
6	tract	acs	2018	S OR	51901_C02_001	Participation Rate	Y	US_Employment	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
7	tract	acs	2018	S OR	51901_C04_001	Unemployment Rate	Y	US_Employment	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
8	tract	acs	2018	S OR	51901_C03_011	Employment/Pop. Ratio	Y	US_Employment	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
9	tract	acs	2018	S OR	51901_C01_012	Household Median Income	Y	US_Employment	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
10	tract	acs	2018	S OR	802001_001	Total Population	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
11	tract	acs	2018	S OR	802001_002	White alone	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
12	tract	acs	2018	S OR	802001_003	African American alone	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
13	tract	acs	2018	S OR	802001_004	American Indian and Alaska Native alone	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
14	tract	acs	2018	S OR	802001_005	Asian alone	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
15	tract	acs	2018	S OR	802001_006	Native Hawaiian and Other Pacific Islander alone	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
16	tract	acs	2018	S OR	802001_007	Some other race alone	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
17	tract	acs	2018	S OR	802001_008	Two or more races	Y	US_Race	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
18	tract	acs	2018	S OR	51901_C01_001	Number of Households	Y	US_Income	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					
19	tract	acs	2018	S OR	51901_C01_002	Less than \$10,000	Y	US_Income	US_Census	estimate	This data was acquired using	Total Pop Census	https://www.Tract.County.Sta					

- ca_census tab
 - This spreadsheet is very similar to the tidycensus spreadsheet with some minor executive
 - You will only need to make changes to this spreadsheet if you want to include/exclude more/less variables in the CA census map layer
 - Each row is a different census variable query
 - There a many columns that record different types of meta-data for each census variable (row)
 - The columns with the green accented headers are the only meta-data that is required to obtain and build the shapefiles, they are:
 - variable: unique ID for each census variable
 - var_name: a unique name to call the variable other than it's ID - based on the actual name of the variable provided for by the
 - name: the category that the variable falls under
 - processed_name: the name of the folder that the shape-file will be written, the layers' names, and the base map layer name it will have in the map
 - Variables with the same processed_name will be grouped into the same shape-file

- selection: contains a string of the raw column names that are written when the shape-file is first written to *application_shapefiles*
 - The esri format has a limit on column name lengths and these column names are a result process
 - The string needs to contain all the column names included in the shapefile
- selected_new: A string of the corrected *selection* names that are human readable

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	boundary	src	year	variable	var_name	name	zool	selected							
2	CT	ca	16	v_CA16_	Population, 2016	General	Population (2016)	Name, Area, Population (201							
3	CT	ca	16	v_CA16_	Population percentage change, 2011 to 2016	General	Population (2016)	Name, Area, Population (201							
4	CT	ca	16	v_CA16_	Population density per square kilometre	General	Population (2016)	Name, Area, Population (201							
5	CT	ca	16	v_CA16_	Participation rate	Employment	Population (2016)	Name, Area, Population (201							
6	CT	ca	16	v_CA16_	Employment rate	Employment	Population (2016)	Name, Area, Population (201							
7	CT	ca	16	v_CA16_	Unemployment rate	Employment	Population (2016)	Name, Area, Population (201							
8	CT	ca	16	v_CA16_	Total Population for all income groups	Income	Population (2016)	Name, Area, Population (201							
9	CT	ca	16	v_CA16_	Without total income	Income	Population (2016)	Name, Area, Population (201							
10	CT	ca	16	v_CA16_	With total income	Income	Population (2016)	Name, Area, Population (201							
11	CT	ca	16	v_CA16_	Under \$10,000 (including loss)	Income	Population (2016)	Name, Area, Population (201							
12	CT	ca	16	v_CA16_	\$10,000 to \$19,999	Income	Population (2016)	Name, Area, Population (201							
13	CT	ca	16	v_CA16_	\$20,000 to \$29,999	Income	Population (2016)	Name, Area, Population (201							
14	CT	ca	16	v_CA16_	\$30,000 to \$39,999	Income	Population (2016)	Name, Area, Population (201							
15	CT	ca	16	v_CA16_	\$40,000 to \$49,999	Income	Population (2016)	Name, Area, Population (201							
16	CT	ca	16	v_CA16_	\$50,000 to \$59,999	Income	Population (2016)	Name, Area, Population (201							
17	CT	ca	16	v_CA16_	\$60,000 to \$69,999	Income	Population (2016)	Name, Area, Population (201							
18	CT	ca	16	v_CA16_	\$70,000 to \$79,999	Income	Population (2016)	Name, Area, Population (201							
19	CT	ca	16	v_CA16_	\$80,000 to \$89,999	Income	Population (2016)	Name, Area, Population (201							
20	CT	ca	16	v_CA16_	\$90,000 to \$99,999	Income	Population (2016)	Name, Area, Population (201							
21	CT	ca	16	v_CA16_	\$100,000 and over	Income	Population (2016)	Name, Area, Population (201							
22	CT	ca	16	v_CA16_	\$100,000 to \$149,999	Income	Population (2016)	Name, Area, Population (201							
23	CT	ca	16	v_CA16_	\$150,000 and over	Income	Population (2016)	Name, Area, Population (201							

- manual tab
 - This sheet differs from the ones above - it is only used to process data that has already been obtained and is not used to acquire new data
 - This data is used in two ways - to determine which shape-files to process and how to process them and how to process them (name changes, etc)
 - Each row is a singular shape-file that could be included in the final application
 - Each column is a unique attribute given a shape-file or a trigger for a process to be ran during the data processing or application building process
 - Trigger Columns:
 - to_application_shp
 - Indicates (Y) weather a shapefile should be initially processed (spatially filtered and unneeded columns removed)
 - These shape-files are taken from *shapefiles_raw_cache* to *application_shapefiles*
 - Only relevant if you are rerunning the developers processing workflow

- to_map_ready
 - Indicates (Y) whether a shapefile should be fully processed and ready to be seen in and that it should be included in the application
 - These shape-files either will undergo manual (actually automated code written specifically for different shape-files) processing, be merged with other shape-files, and/or not be processed at all
 - This column is relevant for all shape-files that are intended to be seen in the application
- to_application_shp/to_map_ready combinations and what they mean
 - Y/Y is a file that is in *shapefiles_raw_cache*, processed by *data_process_raw.R*, put into *application_shapefiles*, processed by *data_process_map_ready.R*, put into *map_ready*, and will be in the application
 - N/Y is a file that does not undergo initial processing but will be used in the application
 - Examples: Shape-files that were manually created (either by merging or from excel data) via *data_process_map_ready.R*, any shape-file that doesn't need to be processed from its raw form via the current workflow
 - Y/N is a file that can be processed via the developers workflow but was decided not to be used for the application
 - N/N is a file that will not be processed and will not be in the application
- Attribute Columns:
 - raw_name and raw_layer_name: raw folder and layer name for a shape-file, only required if it will be processed (if to_application_shp == Y)
 - processed_name: the name of the folder that the shape-file will be written to and the layers' names
 - These files **MUST** be shape-files and must have the same name for the layers as the folder it is in
 - E.g. path = `"./folder_name/layer_name.shp"` where (folder_name == layer_name)
 - group: a variable used for indexing in the code, should either be Political, Transportation, or Landmarks (case sensitive)
 - dl_date: deprecated

- selection: contains a string of the raw column names that are written when the shape-file is first written to *application_shapefiles*
 - Note: esri format has a limit on column name lengths and these column names are a result process
 - The string needs to contain all the column names included in the shape-file
- zcol: the variable (human-readable form of column, should be in selected_new column) that should be used for layer coloring
 - *plain* is used to indicate no coloring
- selected_new: A string of the corrected *selection* names that are human readable - should be the same length as *selection* column strings
- notes: additional information that is associated with a particular layer
- src_url: indicates the source for the data - *NA* if there is no source or if was made manually

to_application_shp	to_map_ready	raw_name	raw_layer_name	processed_name	group	dl_date	selection	zcol	selected_new	notes	src_url
Y	Y	Airports_Aeropot_2014_en_shaj	Airport-Aeropot_2014	CA_Airports	Transport	NA	AIRPORT, CITY_NAME, PROVINCE		Name, City, Prov	Canadian airports served by	https://open.canada.ca/data/en/da
Y	Y	ABMS_MUNICIPALITIES_SP	ABMS_MUNI_polygon	CA_Cities	Political	NA	AA_NAME, ABRVN, AA_PARENT	plain	Name, Short Name	Legally defined Municipalities	https://catalogue.data.gov.bc.ca/da
N	N	CA_District_Municipality	CA_District_Municipality	CA_District	Municipal	NA	RGN_NM, RGN_TYP, NCRPTN	plain	Name, Type, Stat	This layer was built from a	https://catalogue.data.gov.bc.ca/da
N	N	Premiere_Nation_First_Nation	Premiere_Nation_First_Nation	CA_First_Nations	Political	NA	BAND_NBR, BAND_NAME	plain	Number, Name	The First Nations geographical	https://open.canada.ca/data/en/da
Y	Y	CA_First_Peoples	CA_First_Peoples	CA_First_Peoples	Political	NA	RGN_NM, RGN_TYP, NCRPTN	plain	Name, Type, Stat	This layer was built from a	https://catalogue.data.gov.bc.ca/da
N	N	EBC_LOCAL_GOVERNMENTS_SP	EBC_LOC_GO_polygon	CA_Government	Political	NA	NA	plain	NA	This layer was built from a	https://catalogue.data.gov.bc.ca/da
Y	Y	GM_MRN	GM_MRN	CA_Major_Roadways	Transport	NA	RoadName, Municipality	plain	Name, City	The Major Road Network (M)	https://catalogue.data.gov.bc.ca/da
Y	Y	CA_Park_and_Ride	CA_Park_and_Ride	CA_Park_and_Ride	Transport	NA	Station, Spaces, Location, Hoi	plain	Station, Spaces	This layer was manually created	https://new.translink.ca/ride-guide
Y	Y	GM_PortofVancouverFacilities	GM_PortofVancouverFacilities	CA_Ports	Transport	NA	Facility, Category, Operator	plain	NA	This data set shows the local	https://catalogue.data.gov.bc.ca/da
Y	Y	GBA_RAILWAY_TRACKS_SP	RW_TRACK_line	CA_Rail	Transport	NA	TRACK_NM, TRNSPT_TE, TRCI	Owner	Name, Type, Trci	Published by the Ministry of	https://catalogue.data.gov.bc.ca/da
Y	N	GBA_RAILWAY_STATIONS_SP	RW_STN_point	CA_Rail_Stations	Transport	NA	STTN_NGLSM, STTN_TYP, STTI	System	Name, Type, Sys	Published by the Ministry of	https://catalogue.data.gov.bc.ca/da
Y	N	CA_Regional_District	CA_Regional_District	CA_Regional_District	Political	NA	RGN_NM, RGN_TYP, NCRPTN	plain	Name, Type, Stat	This layer was built from a	https://catalogue.data.gov.bc.ca/da
N	N	NA	NA	Regional_Planning	Political	NA	Name, ORG, SRC, Acronym, Web	plain	Name, Website	This layer was constructed	NA
Y	Y	Intermodal_Passenger_Connecti	Intermodal_Passenger_Connecti	Multimodal_Stations	Transport	44089	metro_area, fac_name, addre	Modes Served	Metro Area, Name	The IPCD is as of July 03, 2010	https://hifid-geoplatform.opendata
N	N	NA	NA	NA	NA	NA	NA	NA	in download	metro boundary Published by the Crown Corp	https://catalogue.data.gov.bc.ca/da
N	N	NA	NA	NA	NA	NA	NA	not yet	MOT Regional B	Published by the Ministry of	https://catalogue.data.gov.bc.ca/da
Y	N	OR_city_limits_shape	citylim_2018	OR_Cities	Political	NA	CITY_NAME	plain	NA	NA	NA
N	N	OR_Roadway_shape	OR_Roadway	OR_Roadway	Transport	NA	NA	plain	NA	NA	NA

Example: Adding Data

This is a brief overview of how data would be added to the map.

Example Data:

Assumption - the shape-file was manually constructed from an excel table

Assumption - the data is made from lat/long points and projected with WGS84

(<https://spatialreference.org/ref/epsg/wgs-84/>)

Assumption - the data has already been processed (spatially filtered for relevancy and only displays)

Assumption - the .shp files are contained in a folder named - *UHSR_Stations (studied)*

Assumption - the .shp files are in a folder named - *UHSR_Stations (studied)*

Assumption - the .shp files are named - *UHSR_Stations (studied)*

Assumption - the data has the attributes "City, StationLoc, Status, State"

Assumption - the data attributes should be displayed in the application as "City, Location, Status, State"

Steps to add data:

- 1) Place folder containing .shp files in *map_ready* folder
- 2) Open *data_source_list.xlsx*
- 3) Begin new entry on manual tab
- 4) Input “N” in *to_application_shp* column since the data has been processed already
- 5) Input “Y” in *to_map_ready* column since the will be used in the application
- 6) Input “NA” in *raw_name* and *raw_layer_name* columns because it is not being processed as a raw file
- 7) Input “UHSR_Stations (studied)” in *processed_name* column because it’s folder and layer names are that
- 8) Input “Transpiration” in the *group* column since the layer is relevant to transportation
- 9) Input “NA” in the *dl_date* column
- 10) Input “City, StationLoc, Status, State” in the *selection* column - the script that builds the application will look to make sure that it selects/inputs the specified columns so this column string must be accurate to what is in the .shp files
- 11) Input “plain” in the *zcol* column if you do not want to display distinct colors for the shape-files
 - Alternatively, the features could be colored by “status” indicating if have just been studied or if they are apart of existing infrastructure
- 12) Input notes that provide ample information to the user regarding this layer
- 13) Input “NA” for *src_url* column since the data was made manually and does not have a direct source link
- 14) Run the *data_mapper.R* to check whether the map is included in the application objects
 - Note: Make sure packages are loaded (run *global.R*)
 - Note: Make sure the working directory is set to root

General

Notes

- All paths are relative
 - Only relevant during dev
 - For dev set working directory to root

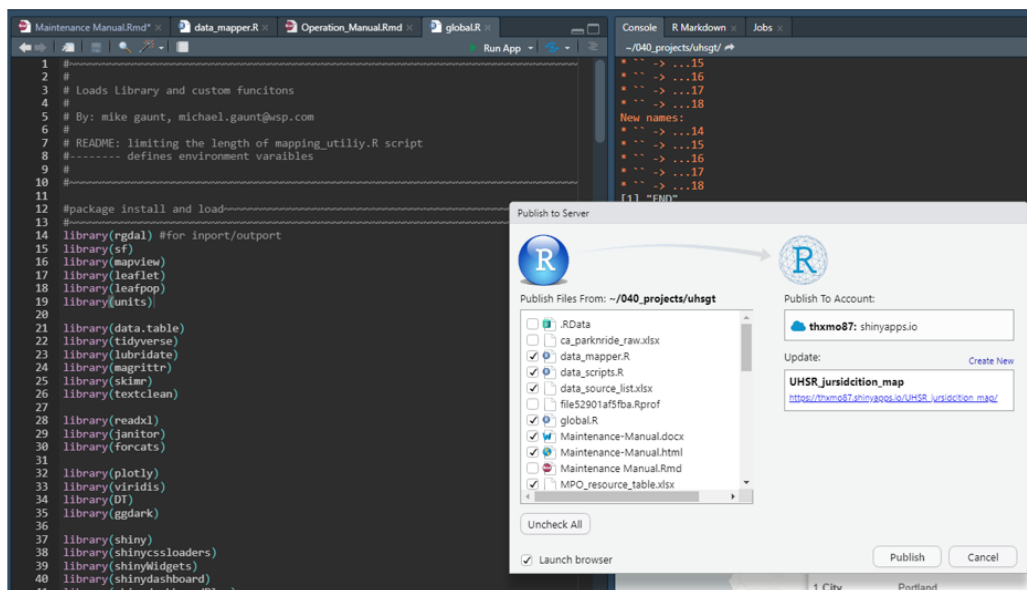
Locally Viewing the Application

The application can be ran locally either in an R window or in a local host via a browser. This can be done by running either the *global.R*, *ui.R*, or *server.R* scripts. It doesn’t matter which one you run since R runs all three of these scripts as a “team” with the execution of one of them - “global.R” will always run which will define the environment and the call the *data_mapper.R* and *data_scripts.R* files.

Redeploying the Application

Note: the process below assumes the application is deployed on shinyapps.io (<https://www.shinyapps.io/>)

After changes have been made to the application and it is stable, the application can be redeployed to the hosting service. Once the application is redeployed, users will have access to the changes that were made since the last deployment. Before the application is deployed, the R environment needs to be cleaned - remove all variables that are in the environment and detach all packages that are currently loaded. To do this, close R and do not save any the environment or the .RData file, this ensures that no hidden dependencies are in the global environment the next time R is opened. Once closed, reopen R and run either the either the *global.R*, *ui.R*, or *server.R* scripts to run the application locally. Following this process ensures that application are capable of building the application and does not depend on any variables that had been created during earlier development. Once the application is running, the app can be pushed to the shinyapps.io.



Click on the “blue circular icon” to the right of the “run-app” button on the navbar (seen in picture above). A pop-up window will appear, indicating which account the application will be pushed to, which project the application will be pushed to/update, and a list of items to publish. Only the items that are required to build the application need to be pushed, these items are as follows:

- global.R
- ui.R
- server.R
- data_mapper.R
- data_scripts.R
- corridor_buffer items

- map_ready items
- mapedit_tmp items
- www items