OSS Report Documentation

*Last Updated: December 12, 2022*

## Purpose

The purpose of this document is to allow a DVRPC staff member to understand the infrastructure needed to produce, maintain, and potentially further build out the Yearly Municipal Crash Report for the Office of Safe Streets.

## A High Level Overview

What this report processing pipeline does is….

1. Retrieves crash and population data from the DVRPC Data Catalog, in preparation of generating a new report
2. Checks this newly-retrieved data against a cache .parquet file that was saved as part of the last report-building process
3. If the newly-retrieved data is “better”, as in it has newer year data, more rows of data, or was simply modified more recently than the cache version of the data, then the newly-retrieved version of the data is used to generate a new report.
   1. As referenced above, this new data becomes the new “cache” version for future checking.
4. If the newly-retrieved data is identical to the cache version of the data, then nothing happens.

## Necessary Files

All files necessary for this report are located in the folder “2022\_exercise”. In addition to the actual files themselves, it will be useful to have the latest version of R (the report was originally built under R 4.2.2), and R-Studio (this report was built using RStudio 2022.07.2+576 “Spotted Wakerobin”, 2022-09-06). You will need to install any and all packages required for this report build; this can be accomplished via the code below:

install.packages("pacman")  
pacman::p\_load(tidyverse, janitor, lubridate, ckanr, here, arrow, gt, glue, ggplot2, quarto)

The files needed are:

1. report\_functions\_data.R - This file loads all packages, defines several handy functions, and does most, if not all of the data pre-processing that is needed for the report.
2. construct\_report.R - This is the “main” rendering script that does the actual report generation. It first sources report\_functions\_data.R, and then compares the newly-retrieved data against the existing cache file. If running a new report is appropriate, you guessed it, it generates new tables and runs a new report! Otherwise, it will give you a message and then not do anything further.
3. sample\_report.qmd - This is the skeleton of the actual report document that is outputted by this entire pipeline. As much as possible, steps were made to ensure it needs minimal maintenance and updating over time. For example, there is a lot of use of in-line R-code so that the “Key Findings” section is generated automatically and with no user input.

## Rendering a Report

In order to render a report, the user can source the script, either within an interactive R session (e.g. source("construct\_report.R")), or by running it via command line (e.g. Rscript construct\_report.R). The latter is useful for task-scheduling or other automated processes. The way that the script is written, it should either produce a new report or not, depending on the state of the latest data.

Knowing that these data sets represent yearly data, it may be prudent to only run these reports on, say, January 15th of each year, to give enough time for end-of-year data to be vetted and pushed out to the DVRPC Data Catalog.

## Troubleshooting

Errors in the tidyverse-era of R are generally very well-defined and helpful. Read the error carefully to understand where and why the error occurred. Simply identifying where in the pipeline the error occurred will hopefully be enough to address the issue and get things back up and running.

The most likely sources of future errors are:

1. The DVRPC Data Catalog has changed in some way, and the current code resource IDs or specific variables fail to pull the right data or can’t find the right variables. This might be an easy fix such as exploring the Data Catalog and updating the resource pointer, or updating the code from TOTAL CRASH to Total Crashes. The complications may be more nuanced; the CRASH YEAR might be a character one time and then numeric another.
2. Functions or syntax in R-packages that are used here have become deprecated or have changed slightly, leading to messages at best to errors at worst. Try to keep up with the latest versions of packages and read the package release notes, etc. Stackoverflow is your friend.

Testing your code in interactive or staging-like environments is preferred before the code goes into production.

## Future To-Do List

Perhaps one of the most obvious ongoing improvements to be made with this report is to talk to the Office of Safe Streets and get their take on what data analyses and statistics they would find useful and actionable.

One known limitation is how gt() tables are rendered in current .docx output files. Hopefully this will be solved in future versions, but the current workaround in this version of the report-building is to save off the tables via gtsave(), and then re-insert them as appropriate in the body of the sample\_report.qmd.

The current outcome variables and the current report format were a very quick first-pass of generating something interesting and hopefully at least a little bit useful. Of course, there are always more variables to explore, better plots to make, maps to create, etc…