|  |  |
| --- | --- |
| Shared Parking Analysis Tool | |
|  |  |
| A picture containing diagram  Description automatically generated | |
| **2025** | **Chittenden County Regional Planning Commission** |

|  |
| --- |
| Chittenden County Regional Planning Commission |
| **Report Title**:  Shared Parking Analysis Tool |
| **Report Prepared by**: RSG  V1: 2023  V2.1: April 2025 |
| **Report Prepared for**:  Chittenden County Regional Planning Commission |
| **For additional information regarding this report, or for questions about permissions or use of findings contained therein, please contact**:  RSG (Headquarters) 55 Railroad Row White River Junction, VT 05001 (802) 295-4999 www.rsginc.com |

© 2023 RSG

[1.0 Introduction 1](#_Toc196850011)

[1.1 Background 1](#_Toc196850012)

[1.2 How To Use This Guide 1](#_Toc196850013)

[2.0 Using SPA 2](#_Toc196850014)

[2.1 Background Software 2](#_Toc196850015)

[2.2 Download the SPA code 3](#_Toc196850016)

[2.3 Installation of the SPA Tool 4](#_Toc196850017)

[2.4 Input Files 6](#_Toc196850018)

[Generators – GIS File 6](#_Toc196850019)

[Parking Supply – GIS File 7](#_Toc196850020)

[Parking Demand and Adjustments.xlsx 8](#_Toc196850021)

[Restrict List (restrict\_list.csv) 9](#_Toc196850022)

[Configuration.YAML 9](#_Toc196850023)

[2.5 Running the Tool 10](#_Toc196850024)

[3.0 Output Data and Results 11](#_Toc196850025)

[3.1 CSV outputs 11](#_Toc196850026)

[3.2 Post-processing 11](#_Toc196850027)

[3.3 post-processing output 15](#_Toc196850028)

[4.0 Calibration and Refining Input Data 19](#_Toc196850029)

[5.0 Tool Methodology 20](#_Toc196850030)

**List of Figures**

[Figure 1: Anaconda Prompt 8](#_Toc140436497)

[Figure 2: Conda Environment Install 9](#_Toc140436498)

[Figure 3: Conda Python Environment (after) 9](#_Toc140436499)

[Figure 4: post process step 1 - load packages 12](#_Toc140436500)

[Figure 5: post process step 2 - define project & loading inputs 12](#_Toc140436501)

[Figure 6: post process step 2 - define project, setting directories 13](#_Toc140436502)

**List of Tables**

**No table of figures entries found.**

# Introduction

## Background

This user guide provides instructions for the Shared Parking Analysis (SPA) tool developed by the Chittenden County Regional Planning Commission (CCRPC) with support from RSG. The SPA tool models shared parking demand for specific geographic areas based on land use and adjustment factors by month, day of week, and time of day.

The SPA uses parking demand factors from Shared Parking, Third Edition by the Urban Land Institute*[[1]](#footnote-1)* (ULI) are incorporated by default, although the tool is flexible to accept user specific inputs along with other parking demand factors.

The SPA uses open-source software in both Python and R software languages complemented by any GIS program that can generate a Shapefile with latitude and longitude location data for both the source of parking demand (i.e., land uses) and the parking lots (parking supply).

The SPA is a unique and powerful tool that expands the methodology developed by the ULI. Rather than the typical ULI Shared Parking analysis as a many demand to one lot analysis, The SPA tool evaluates multiple parking demands across multiple possible parking lots.

## How To Use This Guide

This guide sets out the process for developing the inputs for the SPA, installing and running the SPA tool, and using the post process visualization summary in Excel.

This guide provides chapters on:

* Preparing SPA inputs
* Running the SPA tool
* Analyzing the results
* Testing the effect of adding a new generator to an existing population of parking lots and generators
* Refining the input data with real world observations

File names are in *italics*.

Tab names in an Excel file are in **bold**.

# Using SPA

This chapter sets out the process for installing the necessary software to run the SPA tool.

## Background Software

Several pieces of software are necessary to develop the inputs and run the SPA tool. These include the following:

* **conda via**[**Anaconda**](https://www.anaconda.com/products/individual)**or [Miniconda](https://docs.conda.io/en/latest/miniconda.html)**: Conda is an open-source package management and environment management system that simplifies the installation and management of software packages and dependencies across multiple programming languages. It allows users to create isolated environments to run different projects with specific package versions and configurations.

A screenshot of a computer

Description automatically generated

* **R**. R is a free and open-source programming language and software environment primarily used for statistical computing and graphics. It provides a wide range of statistical and graphical techniques and has a large community of users and developers contributing to its extensive collection of packages.
* **R Studio**. RStudio is an integrated development environment (IDE) for the R programming language. It provides a user-friendly interface and various tools to enhance productivity, making it easier to write, run, and debug R code while offering features like code editing, data visualization, and package management.

A screenshot of a computer

Description automatically generated

* **GIS (qGIS, ArcGIS, ArcMap)**. GIS software (Geographic Information System) is a type of software designed for capturing, analyzing, managing, and presenting geographic data. It enables users to visualize spatial information, perform spatial analysis, create maps, and make informed decisions based on geographical data. QGIS is an open source GIS program while ArcGIS and ArcMap are proprietary ESRI products.

## Download the SPA code

The SPA Source code is stored in a GitHub repository. In order to use the SPA tool the user is required to download the code by cloning the repository or downloading the zip file of the repository from GitHub.

The GitHub repository has the following files:

A screenshot of a computer

Description automatically generated

The SPA ‘Shared\_Parking’ folder should be moved to the location on the local machine where it will be called. Typically, a location near to the root C: drive or in a My Documents location is used.

## Installation of the SPA Tool

The installation of the SPA tool requires the use of setting up a python environment using the Anaconda python interface.

Use the Start menu to find the Anaconda Prompt (see Figure 1). A command window opens. Using commands you need to navigate to the shared\_parking folder.

Figure : Anaconda Prompt

A screenshot of a computer

Description automatically generated

For an example stored on the C drive in a GitHub folder

* cd c:\GitHub\shared\_parking

Once the command window is setup to point to the shared\_parking folder, install the Shared\_Parking environment.

* conda env create -f environment.yml

Figure : Conda Environment Install

A screen shot of a computer

Description automatically generated

The environment loads the necessary supporting software packages into the Python environment so that the analysis can be completed.

Figure : Conda Python Environment (after)

A screen shot of a computer

Description automatically generated

## Input Files

The next chapter sets out the process for setting up the inputs. The model input files are comprised of three components:

* Generators: the land uses that generate the demand for parking
* Parking: the supply of parking spaces
* Demand and Adjustment file: this is the ULI rates of parking demand by land use.
* Python Configuration file

### Generators – GIS File

The parking generator data are spatial datasets that are used to specify the land uses which are associated with a level of parking demand. The tool requires GIS shapefiles that include the specific land uses and the land use characteristics for any scenarios to be tested in the SPA tool. The shapefile is created in a GIS program.

The generator shapefile has the following data fields:

* Name – Name of the parking generator. This column is not directly used in the tool and is there only to help the user identify the generator. *optional*
* Address – Physical address of the land use. *optional*
* Gen\_ID – Unique ID associated with each generator. ***mandatory*.**
* Long & Lat – are the longitude and latitude of the centroid of the land use. ***mandatory***
* NAICS – code for distinguishing the land use that generates parking demand. This code and the description are useful for the user to select the most appropriate land use code for aligning the land use to the Shared Parking methodology. *optional*
* NAICS Desc – the description for the NAICS code. *optional*
* Land Use – general description of the land use, either zoning, or some other local definition. *optional*
* LUC – Land Use Code of a particular generator’s land use. This number must match a land use code in the *Land Use Demand* file and *Adjustment Factors* file. ***mandatory***
* Type – Description of the type of land use. Similar, if not identical to, the Land Use field. *optional*
* Unit – The unit in which size is measured. The unit type must match the Land Use’s unit type in the *Land Use Demand* file. ***mandatory***
* Size – The size of the land use in the units specified. For example, number of housing units or the square footage of a specific nonresidential use. ***mandatory***
* DUType – SF single family & MF multifamily. ***mandatory***
* DUCount – the number of households living in the structure. ***mandatory***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GEN\_NAME | GEN\_ADDR | GEN\_UID | GEN\_LON | GEN\_LAT | NAICS | NAICS\_DESC | LAND\_USE | LUC | TYPE | UNIT | SIZE | DUTYPE | DUCOUNT |
| **BAKER'S DOZEN INC** | **70 Roosevelt Hwy Ste 2** | **100** | **-73.1816388468** | **44.5032888286** | **31** | **Manufacturing** | **Industrial** | **60** | **Industrial** | **ksf GLA** | **6.756** |  | **0** |

### Parking Supply – GIS File

The parking supply data are spatial datasets that are used to specify the characteristics of the parking lots. The tool requires GIS shapefiles that includes information pertaining to the parking lot, who may have access to the lot, and other aspects. The shapefile is created in a GIS program.

Sample parking generator lot:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LOT\_NAME | SPACE\_TOT | LOT\_UID | LOT\_LON | LOT\_LAT | RESTRICT | LOT\_GEN\_ID | Shape\_Leng | Shape\_Area | NOTES | Category | Street |
| Lapointe Street | 21 | 1 | -73.1746875862 | 44.4972150788 | 0 | 0 | 407.050748091 | 611.072401597 |  | On Street | Lapointe St |

The parking supply shapefile has the following data fields:

* Name – Name of the parking location. *optional*
* Space\_Tot – the number of parking spaces in the parking lot. *optional*
* Lot\_UID - Unique ID associated with each generator. ***mandatory***
* Long & Lat – are the longitude and latitude of the centroid of the parking supply. ***mandatory***
* Restrict – this code specifies who can park in the parking spaces. ***mandatory***
* Lot\_Gen\_ID: if there is a specific land use that it is connected with.
* Shape\_length – information on the parking polygon. *optional*
* Shape\_area – information on the parking polygon. *optional*
* Category – on-street, off-street, residential, etc. *optional*
* Street – the name of the street it is most closely associated with. *optional*

### Parking Demand and Adjustments.xlsx

This Excel file contains parking demand ratios for the land uses in the *Generators* folder. The SPA tool uses demand factors and adjustments from the second edition of Shared Parking, but different data can be used as long as it conforms to the same format. For example, if any local data is collected, the parking rates can be adjusted in this Excel file to reflect local conditions. The Parking Demand and Adjustment file includes all hours, months, and seasons of the year. The following fields are used within the Excel file to define the parking generation demand for the land uses.

* LUC – Land Use Codes. These numbers can be arbitrary, but they must match the land use codes in the *Generators* file (LUC column).
* Land Use – Description of the type of land use. This column is not used in the tool.
* User – Either “Visitor/Customer” or “Employee.” These two types of users have different parking demand ratios. In the case of housing, residents’ parking demand appears under “Employee.”
* Weekday – Weekday demand ratio, the number of parking spaces per unit required at peak weekday times.
* Weekend – Weekend demand ratio, the number of parking spaces per unit required at peak weekend times.
* Unit – The units which correspond to the demand ratios.

Sample data is shown below:

A white background with black text

Description automatically generated

The Excel file includes the Land Use, Monthly parking demand factors, and a Time of Day (TOD) worksheet.

### Restrict List (restrict\_list.csv)

The restrict list is a file designed to limit the land uses who are allowed to park in specific parking areas. The CSV includes two columns of data which specifies for any specific lots, whether that demand can park there or not. The demand (GEN\_UID) is attempted to be first allocated to these parking lots (LOT\_UID), but is allowed to park elsewhere. But if the LOT\_UID is specified in this file, then only those Generators listed will be able to be allocated there. The table shows a sample restrict file.

A screenshot of a computer

Description automatically generated

The Restrict Codes are used to inform which land uses can park in the specific parking lots.

A computer screen with text

AI-generated content may be incorrect.

### Configuration.YAML

Python uses a configuration file to guide the collection of the input files and direct outputs to specific locations. This ‘*Configuration.yaml*’ file is the necessary link between the data and the analysis. The file has several sections and areas for the user to input specific file names and folders for the parking analysis.

The file includes information on months to analyze, days of the week, hours of the day, etc.

* *data\_dir: data* #this is the folder where the data inputs are stored. These data include:
  + parking generation shapefile
  + parking supply shapefile
  + Parking demand and adjustments file
  + Restricted parking lot file
* *output\_dir: baseoutput* #this is the name of the folder where the output from the SPA will go.
* *factors\_file:* ***'Parking Demand and Adjustments.xlsx'*** #excel file with worksheets
* *monthly\_sheetname: Monthly* #worksheet name with monthly parking factors
* *daily\_sheetname: LandUse* #worksheet name with daily parking rates
* *hourly\_sheetname: TOD* #worksheet name with hourly parking rates

Other data is specified in the configuration file but shouldn’t need to be changed such as the name of the land use columns, and setting other variable names in the Parking Demand and Adjustments Excel file.

The configuration file requires the name of the specific demand (i.e., Generator) file and the supply file.

* demand\_shapefile: **WinCity\_Calibratedbased.shp**
* supply\_shapefile: **WinCity\_BaseCalibration\_Supply.shp**

Other data in the file shouldn’t need to be altered, but the YAML does provide flexibility on other key model parameters and inputs. Several of which are used in the calibration process to adjust parking rates and the attractiveness of certain lots.

## Running the Tool

Once the conda environment has been activated then the tool is ready for running. Enter the following command to run the tool:

* Run “*python run\_model.py -c tests/winooski\_example/configuration.yaml*” to use the example configuration. This will generate output files for use in the post-processing R scripts. Note, the later part of the python code points to the location of the Configuration YAML file.

# Output Data and Results

## CSV outputs

The Shared Parking Analysis Tool’s Post-Processing R Scripts are designed to extract data from the CSV output files are created after running the Python script. These include: This section describes the format of the output files generated by the Python SPA:

* Demand: Demand for individual lots.
* Factors: Hourly adjustment factors for weekdays and weekends.
* Gen\_lots: Parking generators with generator and lot IDs.
* Preference: Generates information for parking preferences.
* Timeseries: A timeseries showing utilization by lot.

## Post-processing

The post-process Excel visualization file will be created by running the R script, “*Post Process.R*” in the Post Process folder. This is the main file that can run all other supporting scripts to analyze the outputs from the SPA. Note, that required libraries will be called in the R script but may require installation before running the script. These libraries are similar to the packages that were installed during the building of the Shared\_Parking environment in the Python setup.

The supporting R scripts called by the Post Process.R script are located within the subfolder titled “Source” in the R project directory. The Source directory contains the following supporting files:

* *read\_shapefiles.R*: Open’s shapefiles of generated parking lots.
* *specific\_gens\_lots.R*: Gets demand information for one or more generators or parking lots using generator and lot ID variables.
* *constraint check.R*: Generates summary space constraints on parking lots and demand.
* *counts\_analysis.R*: Generates parking counts for on and off-street parking during weekdays and weekends.
* *demand\_check.R*: Estimates total parking unconstrained demand versus demand in the parking model.

Figure : post process step 1 - load packages

Text

Description automatically generated

Running the post process will require defining the project type and editing directory selections within the script to correctly call and store files. The projects available for analysis include “Winooski\_city” and “Winooski\_ave.” The directory called from for “model\_dir” should contain the output from Python scripts used in the SPA. The directory called from “dir” should contain a folder titled “Outputs” for storing post processing results.

Figure : post process step 2 - define project & loading inputs

Text

Description automatically generated

Figure : post process step 2 - define project, setting directories

Text

Description automatically generated

Step 3 reads data outputs from the Python model and shapefile inputs used for the SPA model run using the “*read\_shapefiles.R*” script. The folder “Source” must be in the same directory as the R Project in order these scripts.

Figure : post process step 3 – read data

Text

Description automatically generated

Step 4 analyzes the parking data to project capacity versus demand stored in usable output files. Run the constraint check section to plot demand vs capacity for both the model and unconstrained demand given the SPA outputs.

Figure : post process step 4 – constraint check

A screenshot of a computer

Description automatically generated

Running the script file “counts\_analysis.R” will develop counts for weekday and weekend to project demand by hour, day, and month. Land use codes can be adjusted for on-street shared parking as well.

Figure : post process step 4 – counts analysis

Text

Description automatically generated

Running the last step will export an Excel output to the “Outputs” folder under the output directory.

Figure : post process step 5 – write output

Text

Description automatically generated

## post-processing output

Post-processing will store an Excel file titled “Visualization.XLSX” to the chosen directory. The file contains the following tabs:

* “Overall summaries” contains pivot tables showing high-level summaries with changeable filters.
* “OnStreet 3 period” shows on-street parking utilization for hours 8:00, 13:00, and 18:00 by street.
* “Single Street Pivot” shows parking demand and utilization parking over a 1-day period for a given street.
* “parking\_formatted” shows geographic information of parking lots.
* “timeseries” contains raw data for parking utilization by street, time, and land use.

The first table in the “Overall summaries” worksheet shows the following data points for a for a given month, day (weekday vs weekend), and hour:

* The total number of spaces (“Sum of SPACE\_TOT”).
* Total utilized spaces by land use type (“Sum of demand”).

Graphical user interface, table

Description automatically generated

figure 10: overall summaries output 1

In contrast, the second table, “OnStreet 3 Period,” can also be filtered by land use category and is analyzed at a lot-specific level. This table includes the same analyzed variables and percent utilization for each lot (“Sum of PctFull”).

Table

Description automatically generated

figure 11: overall summaries output 2

The table in this worksheet shows utilization rates for each parking lot (“Lot\_UID”) at hours 8:00, 13:00, and 16:00, organized by street. This also shows a grand total utilization rate for a 24-hour period. These can be filtered by month, day, and land use category.

Table

Description automatically generated

figure 12: onstreet 3 period output

This worksheet shows aggregated parking utilization for a given street by hours 0:00 and 6:00-23:00. The table includes:

* Total parking spaces (“Sum of SPACE\_TOT”)
* Utilized spaces/demand (“Sum of demand”)
* Percent utilization (“Sum of PctFull”)

These observations can be filtered by month, day, and street.

Graphical user interface, table

Description automatically generated

figure 13: single street pivot output

The user should check these results for reasonableness. Do they generally agree with the user’s expectation? Do they agree with anecdotal data on this location? If not, the user should double check the inputs and consider if there are special cases in this area that do not conform well with the default demand and adjustment factors.

# Calibration and Refining Input Data

The SPA tool uses a generic dataset of national data from Shared Parking. It should be used as a planning tool to understand the effects of shared parking, both where excess capacity may exist and where a new generator may require more parking than is currently available. Like all planning data, the demand and adjustment factors used here are not perfect, and the user should be careful when demand is shown to be close to supply.

There are a variety of reasons a user may want to change the demand and adjustments factors. A user may decide to use local data for time adjustment factors or use a higher generation rate for a particularly popular generator. The available land use codes may not cover a desired land use type. Shared Parking explains its methods for data collection and how to collect local data.

A good first step is to perform field counts at the times the SPA tool indicates peak demand occurs. It may also be helpful to compare anecdotal data for particular times with what the tool’s output indicates. These observations may show that the tool is generally accurate, or over- or under-estimating peak demand. It is also possible that some stores are not open when the default factors are showing they have demand, e.g. restaurants that are not open after midnight.

If the user determines that the demand and adjustment factors need to be refined, the user should perform parking lot counts in accordance with the Shared Parking methodology. It may be possible to perform counts at only the times of highest demand and adjust the factors accordingly and thus avoid counting all 26 days of factors. Changing factors to reflect store hour hours will also help calibrate a particular area.

# Tool Methodology

The premise of the SPA tool is that for each hour of the analysis it goes through an iterative process of:

* Estimating the parking demand generated by each of the land uses in the model shapefile
* Allocates that demand based on a utility function subject to size constraints and restrictions on which land uses can part in which lots.

The SPA tool uses the parking demand estimates for each land use from the Urban Land Institute’s Shared Parkingguide.

The SPA tool calculates each hour of demand independently, using the hourly factors from Shared Parking. This aggregate approach avoids the need to estimate the individual parking behaviors of each individual vehicle in the network and rather model the overall demand at each time period separately.

The analysis considers:

* Time of Day: 6 AM – Midnight
* Day: Weekday or Weekend
* Time of Year: all 12 months and December after Christmas (termed “Late December”)

1. Smith, Mary S. *Shared Parking*, Third Edition. Washington, D.C.: ULI-the Urban Land Institute and the International Council of Shopping Centers, 2020. [↑](#footnote-ref-1)