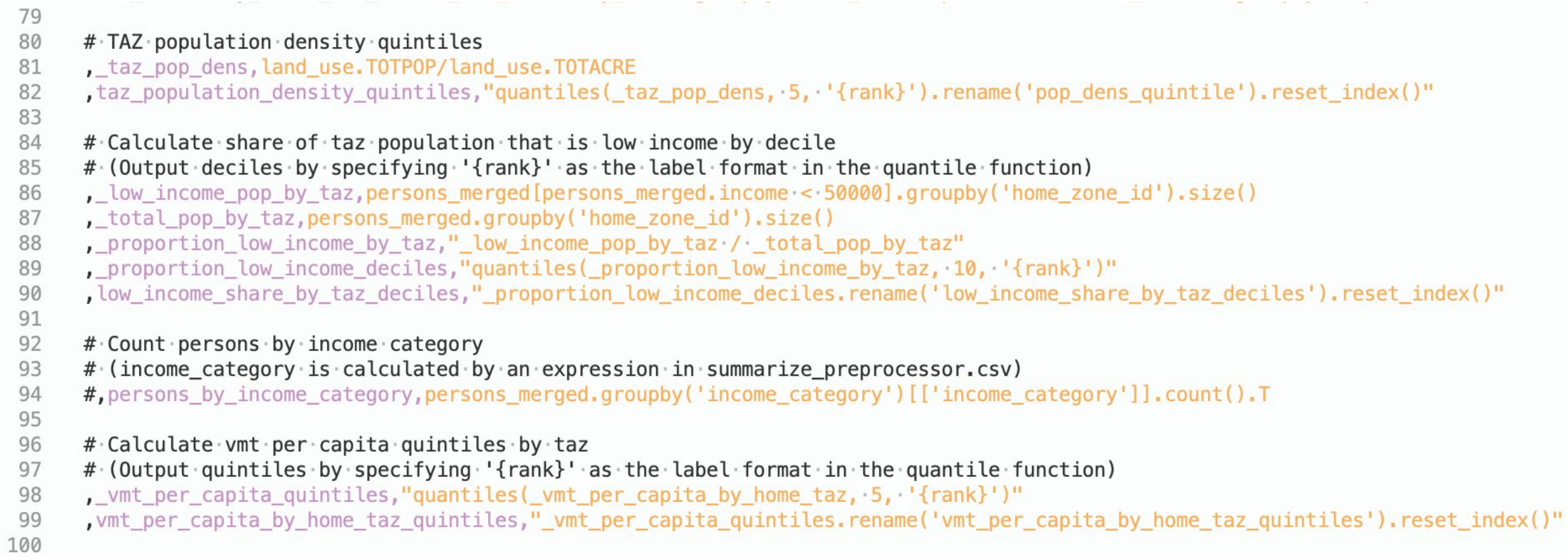
Visualization Update - March 1, 2022

Loose Ends Cleanup:

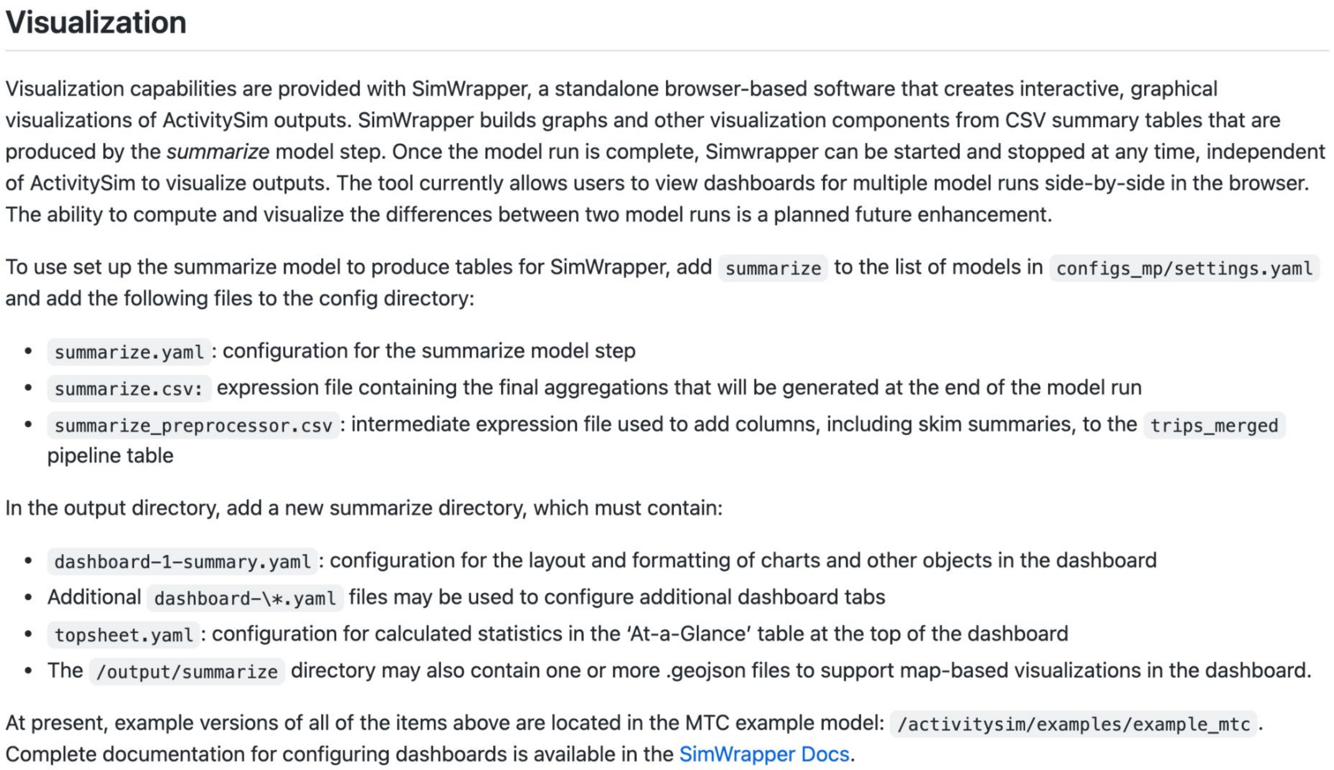
- Access all pipeline tables in summary expressions
- Allow temporary variables to be defined in summary expressions
- Flesh out documentation (esp. for summary expressions)
- Add summarize model to the MTC example
- Clean up sample expressions in MTC example
- Clean up sample dashboard in MTC example
- Make SimWrapper a formal dependency of ActivitySim

- Access all pipeline tables in summary expressions
- Allow temporary variables to be defined in summary expressions
- Clean up sample expressions in MTC example



```

# TAZ population density quintiles
,_taz_pop_dens,land_use.TOTPOP/land_use.TOTACRE
,taz_population_density_quintiles,"quantiles(_taz_pop_dens, 5, '{rank}').rename('pop_dens_quintile').reset_index()"

# Calculate share of taz population that is low income by decile
# (Output deciles by specifying '{rank}' as the label format in the quantile function)
,_low_income_pop_by_taz,persons_merged[persons_merged.income < 50000].groupby('home_zone_id').size()
,_total_pop_by_taz,persons_merged.groupby('home_zone_id').size()
,_proportion_low_income_by_taz,"_low_income_pop_by_taz / _total_pop_by_taz"
,_proportion_low_income_deciles,"quantiles(_proportion_low_income_by_taz, 10, '{rank}')"
,low_income_share_by_taz_deciles,"_proportion_low_income_deciles.rename('low_income_share_by_taz_deciles').reset_index()"

# Count persons by income category
# (income_category is calculated by an expression in summarize_preprocessor.csv)
#,persons_by_income_category,persons_merged.groupby('income_category')[['income_category']].count().T

# Calculate vmt per capita quintiles by taz
# (Output quintiles by specifying '{rank}' as the label format in the quantile function)
,_vmt_per_capita_quintiles,"quantiles(_vmt_per_capita_by_home_taz, 5, '{rank}')"
,vmt_per_capita_by_home_taz_quintiles,"_vmt_per_capita_quintiles.rename('vmt_per_capita_by_home_taz_quintiles').reset_index()"

```

- Flesh out documentation (esp. for summary expressions)



Visualization

Visualization capabilities are provided with SimWrapper, a standalone browser-based software that creates interactive, graphical visualizations of ActivitySim outputs. SimWrapper builds graphs and other visualization components from CSV summary tables that are produced by the *summarize* model step. Once the model run is complete, Simwrapper can be started and stopped at any time, independent of ActivitySim to visualize outputs. The tool currently allows users to view dashboards for multiple model runs side-by-side in the browser. The ability to compute and visualize the differences between two model runs is a planned future enhancement.

To use set up the summarize model to produce tables for SimWrapper, add `summarize` to the list of models in `configs_mp/settings.yaml` and add the following files to the config directory:

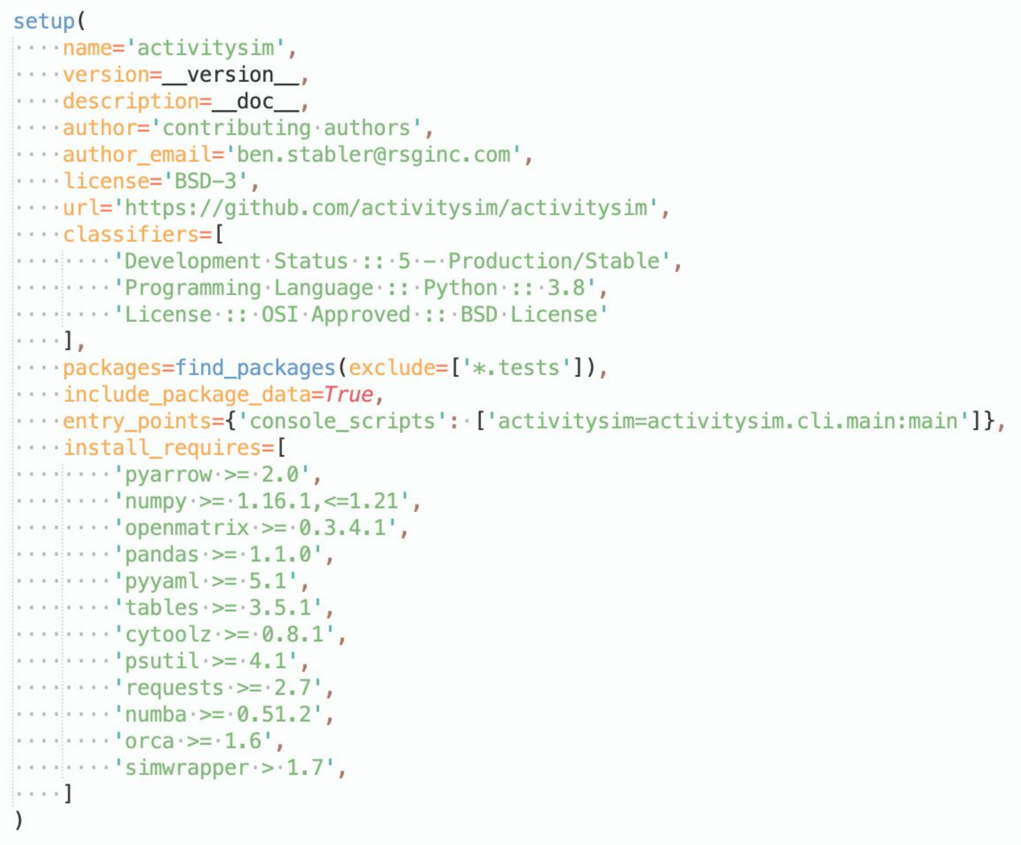
- `summarize.yaml`: configuration for the summarize model step
- `summarize.csv:` expression file containing the final aggregations that will be generated at the end of the model run
- `summarize_preprocessor.csv`: intermediate expression file used to add columns, including skim summaries, to the `trips_merged` pipeline table

In the output directory, add a new summarize directory, which must contain:

- `dashboard-1-summary.yaml`: configuration for the layout and formatting of charts and other objects in the dashboard
- Additional `dashboard-\*.yaml` files may be used to configure additional dashboard tabs
- `topsheet.yaml`: configuration for calculated statistics in the 'At-a-Glance' table at the top of the dashboard
- The `/output/summarize` directory may also contain one or more .geojson files to support map-based visualizations in the dashboard.

At present, example versions of all of the items above are located in the MTC example model: `/activitysim/examples/example_mtc`. Complete documentation for configuring dashboards is available in the SimWrapper Docs.

- Make SimWrapper a formal dependency of ActivitySim



```
setup(
    name='activitysim',
    version=__version__,
    description=__doc__,
    author='contributing authors',
    author_email='ben.stabler@rsginc.com',
    license='BSD-3',
    url='https://github.com/activitysim/activitysim',
    classifiers=[
        'Development Status :: 5 - Production/Stable',
        'Programming Language :: Python :: 3.8',
        'License :: OSI Approved :: BSD License'
    ],
    packages=find_packages(exclude=['*.tests']),
    include_package_data=True,
    entry_points={'console_scripts': ['activitysim=activitysim.cli.main:main']},
    install_requires=[
        'pyarrow >= 2.0',
        'numpy >= 1.16.1,<=1.21',
        'openmatrix >= 0.3.4.1',
        'pandas >= 1.1.0',
        'pyyaml >= 5.1',
        'tables >= 3.5.1',
        'cytoolz >= 0.8.1',
        'psutil >= 4.1',
        'requests >= 2.7',
        'numba >= 0.51.2',
        'orca >= 1.6',
        'simwrapper > 1.7',
    ]
)
```