Project Card Creation Quickstart

In this notebook we will run through:

1. creating a highway project card from a cube log file
2. creating a transit project card from two cube line files

In [1]:
```
import os
import sys

from lasso import Project
from lasso import CubeTransit
from network_wrangler import WranglerLogger
```

In [2]:
```
%load_ext autoreload
%autoreload 2
```

In [3]:
```
import logging
logger = logging.getLogger("WranglerLogger")
logger.handlers[0].stream = sys.stdout
# if you don't want to see so much detail, set to logging.INFO or DEBUG
logger.setLevel(logging.DEBUG)
```

In [4]:
```
#set examples directory
EX_DIR = os.path.join(os.path.dirname(os.getcwd()), "examples")
EX_DIR
```

Out[4]:
```
'C:\\Lasso\\examples'
```

Roadway Project Card

Roadway project cards are built by reading in a base network in the standard network format and a then processing cube log files.

The log file information is stored in a dataframe called `roadway_changes`

The project card data is created when comparing log file to the base network and is stored in the variable `card_data`

In [5]:
```
test_roadway_project = Project.create_project(
        base_roadway_dir=os.path.join(EX_DIR,"stpaul"),
        roadway_log_file=os.path.join(EX_DIR,"cube","st_paul_test.log"),
    )
```

```
2020-06-05 14:28:56, INFO: No base transit network.
2020-06-05 14:28:56, INFO: No transit changes given or processed.
2020-06-05 14:28:56, INFO: Reading logfile: C:\Lasso\examples\cube\st_paul_tes
t.log
2020-06-05 14:28:56, INFO: Processed 1 Node lines and 7 Link lines
2020-06-05 14:28:56, INFO: Reading from following files:
-C:\Lasso\examples\stpaul\link.json
-C:\Lasso\examples\stpaul\node.geojson
-C:\Lasso\examples\stpaul\shape.geojson.
2020-06-05 14:29:04, INFO: Read 66253 links from C:\Lasso\examples\stpaul\link.
json
2020-06-05 14:29:04, INFO: Read 17159 nodes from C:\Lasso\examples\stpaul\node.
geojson
2020-06-05 14:29:04, INFO: Read 66253 shapes from C:\Lasso\examples\stpaul\shap
e.geojson
2020-06-05 14:29:27, INFO: Lasso base directory set as: C:\Lasso
2020-06-05 14:29:27, INFO: Creating calculated roadway variables.
2020-06-05 14:29:27, INFO: Calculating Area Type from Spatial Data and adding a
s roadway network variable: area_type
2020-06-05 14:29:27, DEBUG: Reading Area Type Shapefile C:\Lasso\metcouncil_dat
a\area_type\ThriveMSP2040CommunityDesignation.shp


---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-5-9e09c472f64d> in <module>
      1 test_roadway_project = Project.create_project(
      2         base_roadway_dir=os.path.join(EX_DIR,"stpaul"),
----> 3         roadway_log_file=os.path.join(EX_DIR,"cube","st_paul_test.log")
,
      4     )

c:\lasso\lasso\project.py in create_project(roadway_log_file, roadway_shp_file,
roadway_csv_file, base_roadway_dir, base_transit_source, build_transit_source,
 roadway_changes, transit_changes, base_roadway_network, base_transit_network,
 build_transit_network, project_name, parameters)
    229                 True,
    230             )
--> 231             base_roadway_network.create_calculated_variables()
    232             base_roadway_network.calculate_distance(overwrite = True)
    233             base_roadway_network.fill_na()

c:\lasso\lasso\roadway.py in create_calculated_variables(self)
    165         """
    166         WranglerLogger.info("Creating calculated roadway variables.")
--> 167         self.calculate_area_type()
    168         self.calculate_county()
    169         self.calculate_centroidconnect()

c:\lasso\lasso\roadway.py in calculate_area_type(self, area_type_shape, area_ty
pe_shape_variable, network_variable, area_type_codes_dict, overwrite)
    359         WranglerLogger.debug("Reading Area Type Shapefile {}".format(ar
```

```
ea_type_shape))
    360         area_type_gdf = gpd.read_file(area_type_shape)
--> 361         area_type_gdf = area_type_gdf.to_crs(epsg=RoadwayNetwork.CRS)
    362
    363         joined_gdf = gpd.sjoin(

c:\programdata\miniconda3\envs\lasso_env3\lib\site-packages\geopandas\geodatafr
ame.py in to_crs(self, crs, epsg, inplace)
    562         else:
    563             df = self.copy()
--> 564         geom = df.geometry.to_crs(crs=crs, epsg=epsg)
    565         df.geometry = geom
    566         df.crs = geom.crs

c:\programdata\miniconda3\envs\lasso_env3\lib\site-packages\geopandas\geoserie
s.py in to_crs(self, crs, epsg)
    420         if self.crs is None:
    421             raise ValueError(
--> 422                 "Cannot transform naive geometries.  "
    423                 "Please set a crs on the object first."
    424             )

ValueError: Cannot transform naive geometries.  Please set a crs on the object
 first.
```

In [ ]:
```
test_roadway_project.roadway_changes[0:10]
```

In [ ]:
```
test_roadway_project.roadway_changes.columns
```

In [ ]:
```
test_roadway_project.card_data
```

In [ ]:
```
test_roadway_project.write_project_card(
        "roadway_test.yml"
    )
```

Transit Project Card

Transit project cards are built by taking the differences between two cube transit line files.

In [ ]:
```
test_transit_project = Project.create_project(
        base_transit_source=os.path.join(EX_DIR,"cube","transit.LIN"),
        build_transit_source=os.path.join(EX_DIR,"cube","single_transit_route_att
    )

test_transit_project.write_project_card(
        "transit_test.yml"
    )
```

Roadway Project Card - difference from csv files or shapefiles

For functionality of taking difference from csv files, or shapefiles

In [ ]:
```
test_roadway_project = Project.create_project(
        roadway_csv_file=os.path.join(EX_DIR,"cube","example_csv_roadway_change.c
        base_roadway_dir=os.path.join(EX_DIR,"stpaul")
    )


test_roadway_project.write_project_card(
           os.path.join(
            SCRATCH_DIR,
            "t_" + "example_csv_roadway_change" + ".yml",
        )
    )
```

In [ ]:
```
test_roadway_project = Project.create_project(
        roadway_shp_file=os.path.join(EX_DIR,"cube","example_shapefile_roadway_ch
        base_roadway_dir=os.path.join(EX_DIR,"stpaul")
    )


test_roadway_project.write_project_card(
           os.path.join(
            SCRATCH_DIR,
            "t_" + "example_shapefile_roadway_change" + ".yml",
        )
    )
```