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]:
```
'D:\\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"),
    )
```

```
2020-06-04 13:39:42, INFO: No base transit network.
2020-06-04 13:39:42, INFO: No transit changes given or processed.
2020-06-04 13:39:42, INFO: Reading logfile: D:\lasso\examples\cube\st_paul_te
st.log
2020-06-04 13:39:42, INFO: Processed 7 Node lines and 1 Link lines
2020-06-04 13:39:42, INFO: Reading from following files:
-D:\lasso\examples\stpaul\link.json
-D:\lasso\examples\stpaul\node.geojson
-D:\lasso\examples\stpaul\shape.geojson.
2020-06-04 13:39:58, INFO: Read 1987590 links from D:\lasso\examples\stpaul\l
ink.json
2020-06-04 13:39:58, INFO: Read 223067 nodes from D:\lasso\examples\stpaul\no
de.geojson
2020-06-04 13:39:58, INFO: Read 331265 shapes from D:\lasso\examples\stpaul\s
hape.geojson
2020-06-04 13:40:35, INFO: Lasso base directory set as: D:\lasso
2020-06-04 13:40:35, INFO: Creating calculated roadway variables.
2020-06-04 13:40:35, INFO: Calculating Area Type from Spatial Data and adding
as roadway network variable: area_type
2020-06-04 13:40:36, DEBUG: Reading Area Type Shapefile D:\lasso\metcouncil_d
ata\area_type\ThriveMSP2040CommunityDesignation.shp

c:\anaconda3\envs\break_lasso_test\lib\site-packages\geopandas\tools\sjoin.p
y:61: UserWarning: CRS of frames being joined does not match!(+proj=longlat +
ellps=WGS84 +datum=WGS84 +no_defs +type=crs != epsg:4326)
  "(%s != %s)" % (left_df.crs, right_df.crs)

2020-06-04 13:40:42, DEBUG: Area Type Codes Used: {23: 4, 24: 3, 25: 2, 35:
2, 36: 1, 41: 1, 51: 1, 52: 1, 53: 1, 60: 1}
2020-06-04 13:40:42, INFO: Finished Calculating Area Type from Spatial Data i
nto variable: area_type
2020-06-04 13:40:42, INFO: Adding roadway network variable for county using a
spatial join with: D:\lasso\metcouncil_data\county\cb_2017_us_county_5m.shp

c:\anaconda3\envs\break_lasso_test\lib\site-packages\geopandas\tools\sjoin.p
y:61: UserWarning: CRS of frames being joined does not match!(+proj=longlat +
ellps=WGS84 +datum=WGS84 +no_defs +type=crs != epsg:4326)
  "(%s != %s)" % (left_df.crs, right_df.crs)
c:\anaconda3\envs\break_lasso_test\lib\site-packages\numpy\lib\function_base.
py:2167: RuntimeWarning: invalid value encountered in ? (vectorized)
  outputs = ufunc(*inputs)
```

```
2020-06-04 13:40:52, INFO: Finished Calculating county variable: county
2020-06-04 13:40:52, INFO: Calculating Centroid Connector and adding as roadw
ay network variable: centroidconnect
2020-06-04 13:40:52, DEBUG: Calculating Centroid Connectors using highest TAZ
number: 3100
2020-06-04 13:40:52, INFO: Finished calculating centroid connector variable:
centroidconnect
2020-06-04 13:40:52, INFO: Calculating MPO as roadway network variable: mpo
2020-06-04 13:40:52, DEBUG: MPO Counties: [,1,,, ,3,,, ,4,,, ,5,,, ,6,,, ,
7,,, ,2,]
2020-06-04 13:40:52, INFO: Finished calculating MPO variable: mpo
2020-06-04 13:40:52, INFO: Calculating Assignment Group as network variable:
assign_group
2020-06-04 13:40:52, DEBUG: Calculating Centroid Connectors
2020-06-04 13:40:52, INFO: Centroid Connector Variable 'centroidconnect' alre
ady in network. Returning without overwriting.
2020-06-04 13:40:52, DEBUG: Reading MRCC / Shared Streets Match CSV
2020-06-04 13:40:52, DEBUG: Reading MRCC Shapefile: D:\lasso\metcouncil_data
\mrcc\trans_mrcc_centerlines.shp
2020-06-04 13:40:59, DEBUG: MRCC GDF Columns
Index(['LINK_ID', 'ROUTE_SYS', 'ST_CONCAT', 'geometry'], dtype='object')
2020-06-04 13:41:00, DEBUG: mrcc shst ref df columns
Index(['shstReferenceId', 'shstGeometryId', 'pp_link_id', 'score'], dtype='ob
ject')
2020-06-04 13:41:11, DEBUG: WiDOT GDF Columns
Index(['OBJECTID', 'WISLR_OVLY', 'SURF_TYCD', 'TRLNS_SURF', 'TRLNS_SU_1',
       'LSHD_SURF_', 'LEFT_SHLD_', 'RSHD_SURF_', 'RGHT_SHLD_', 'RDWY_OWRST',
       'RW_INDC', 'RW_WD', 'MEDN_TYCD', 'LEFT_CURB_', 'RGHT_CURB_',
       'PRKG_TYCD', 'TRLNS_NB', 'WISLR_SFRT', 'SDWK_TYCD', 'HARST_TYCD',
       'VARST_TYCD', 'RDWY_MAGMT', 'RDWY_MAG_1', 'LOC_WISLR_', 'RWCRT_MILG',
       'RDWY_CTGY_', 'RDWY_SBCTG', 'AVG_DLY_TR', 'AVG_DLY__1', 'AVG_DLY__2',
       'RDWY_ACS_T', 'FED_URLC_T', 'FEDUA_TYCD', 'FED_CLSN_T', 'FNCT_CLS_T',
       'NHS_CLS_TY', 'NHS_RTE_NB', 'HPMS_SMPL_', 'HOV_LANE_T', 'PVMT_IRI_N',
       'PVMT_IRI_Y', 'ST_LABL_NM', 'ST_PRMY_SY', 'RDWY_LINK_', 'SHC_NTWK_N',
       'WISLR_PTY_', 'WISLR_PTY1', 'WISLR_CMTY', 'WISLR_CERT', 'DOT_CNTY_C',
       'LOC_FLKOS_', 'LOC_TLKOS_', 'LINK_OVLY_', 'DIR_INDC', 'OPOS_RDWY_',
       'INV_YR', 'LOC_WI_CMT', 'MEDN_WD_MS', 'RDWY_SBML_', 'RDWY_GTRST',
       'RDWY_HTRST', 'RDWY_MNTNR', 'RDWY_OWNR_', 'RDWY_SLRST', 'RDWY_WDRST',
       'RDWY_WTRST', 'RWCRT_PTY_', 'RDWY_CERT_', 'RWSR_SYS_T', 'RDWY_SURF_',
       'CSTMS_SFRT', 'RDWY_RTG_D', 'CNTY_PRMY_', 'DSGND_TRK_', 'DSGND_TRK1',
       'CTFR_INDC', 'FNCT_CLS_C', 'FNCT_CLS_G', 'HPMS_SMPL1', 'WISLR_PT_1',
       'SEG_WISLR_', 'RWCRT_RTE_', 'ASGD_RTE_I', 'ASGD_RTE_S', 'SFRTG_TYCD',
       'RTG_SURF_T', 'TRADS_ID', 'PA_WISLR_P', 'WISLR_PT_2', 'PMRC_TYCD',
       'DOT_RGN_NB', 'FED_SFGRP_', 'TRMT_TYID', 'MNTC_TRMT_', 'MNTC_TRMT1',
       'MNTC_TRM_1', 'SHAPE_Leng', 'geometry', 'LINK_ID'],
      dtype='object')
2020-06-04 13:41:33, DEBUG: widot shst ref df columns
2020-06-04 13:41:33, DEBUG: source ShSt rename_variables_for_dbf columns
Index(['shstReferenceId', 'shstGeometryId', 'pp_link_id', 'score'], dtype='ob
ject')
2020-06-04 13:41:33, DEBUG: source gdf columns
Index(['LINK_ID', 'ROUTE_SYS', 'ST_CONCAT', 'geometry'], dtype='object')
2020-06-04 13:41:35, DEBUG: source ShSt rename_variables_for_dbf columns
Index(['shstReferenceId', 'shstGeometryId', 'shstFromIntersectionId',
       'shstToIntersectionId', 'gisReferenceId', 'gisGeometryId',
       'gisTotalSegments', 'gisSegmentIndex', 'gisFromIntersectionId',
       'gisToIntersectionId', 'startSideOfStreet', 'endSideOfStreet',
```

```
       'sideOfStreet', 'score', 'matchType', 'pp_link_id', 'geometry'],
      dtype='object')
2020-06-04 13:41:35, DEBUG: source gdf columns
Index(['OBJECTID', 'WISLR_OVLY', 'SURF_TYCD', 'TRLNS_SURF', 'TRLNS_SU_1',
       'LSHD_SURF_', 'LEFT_SHLD_', 'RSHD_SURF_', 'RGHT_SHLD_', 'RDWY_OWRST',
       'RW_INDC', 'RW_WD', 'MEDN_TYCD', 'LEFT_CURB_', 'RGHT_CURB_',
       'PRKG_TYCD', 'TRLNS_NB', 'WISLR_SFRT', 'SDWK_TYCD', 'HARST_TYCD',
       'VARST_TYCD', 'RDWY_MAGMT', 'RDWY_MAG_1', 'LOC_WISLR_', 'RWCRT_MILG',
       'RDWY_CTGY_', 'RDWY_SBCTG', 'AVG_DLY_TR', 'AVG_DLY__1', 'AVG_DLY__2',
       'RDWY_ACS_T', 'FED_URLC_T', 'FEDUA_TYCD', 'FED_CLSN_T', 'FNCT_CLS_T',
       'NHS_CLS_TY', 'NHS_RTE_NB', 'HPMS_SMPL_', 'HOV_LANE_T', 'PVMT_IRI_N',
       'PVMT_IRI_Y', 'ST_LABL_NM', 'ST_PRMY_SY', 'RDWY_LINK_', 'SHC_NTWK_N',
       'WISLR_PTY_', 'WISLR_PTY1', 'WISLR_CMTY', 'WISLR_CERT', 'DOT_CNTY_C',
       'LOC_FLKOS_', 'LOC_TLKOS_', 'LINK_OVLY_', 'DIR_INDC', 'OPOS_RDWY_',
       'INV_YR', 'LOC_WI_CMT', 'MEDN_WD_MS', 'RDWY_SBML_', 'RDWY_GTRST',
       'RDWY_HTRST', 'RDWY_MNTNR', 'RDWY_OWNR_', 'RDWY_SLRST', 'RDWY_WDRST',
       'RDWY_WTRST', 'RWCRT_PTY_', 'RDWY_CERT_', 'RWSR_SYS_T', 'RDWY_SURF_',
       'CSTMS_SFRT', 'RDWY_RTG_D', 'CNTY_PRMY_', 'DSGND_TRK_', 'DSGND_TRK1',
       'CTFR_INDC', 'FNCT_CLS_C', 'FNCT_CLS_G', 'HPMS_SMPL1', 'WISLR_PT_1',
       'SEG_WISLR_', 'RWCRT_RTE_', 'ASGD_RTE_I', 'ASGD_RTE_S', 'SFRTG_TYCD',
       'RTG_SURF_T', 'TRADS_ID', 'PA_WISLR_P', 'WISLR_PT_2', 'PMRC_TYCD',
       'DOT_RGN_NB', 'FED_SFGRP_', 'TRMT_TYID', 'MNTC_TRMT_', 'MNTC_TRMT1',
       'MNTC_TRM_1', 'SHAPE_Leng', 'geometry', 'LINK_ID'],
      dtype='object')
2020-06-04 13:41:42, INFO: Finished calculating assignment group variable: as
sign_group
2020-06-04 13:41:43, INFO: Calculating Roadway Class
2020-06-04 13:41:43, INFO: Finished calculating roadway class variable: roadw
ay_class
2020-06-04 13:41:43, INFO: Adding Counts
2020-06-04 13:41:43, DEBUG: Adding MNDOT Counts using
- shst file: D:\lasso\metcouncil_data\count_mn\mn_count_ShSt_API_match.csv
- shp file: AADT_mn
- as network variable: AADT
2020-06-04 13:41:43, INFO: Adding Variable AADT using Shared Streets Referenc
e from D:\lasso\metcouncil_data\count_mn\mn_count_ShSt_API_match.csv
2020-06-04 13:41:43, INFO: Added variable: AADT using Shared Streets Referenc
e
2020-06-04 13:41:43, DEBUG: Adding WiDot Counts using
- shst file: D:\lasso\metcouncil_data\Wisconsin_Lanes_Counts_Median\wi_count_
ShSt_API_match.csv
- shp file: AADT_wi
- as network variable: AADT
2020-06-04 13:41:43, INFO: Adding Variable AADT using Shared Streets Referenc
e from D:\lasso\metcouncil_data\Wisconsin_Lanes_Counts_Median\wi_count_ShSt_A
PI_match.csv
2020-06-04 13:41:44, INFO: Added variable: AADT using Shared Streets Referenc
e
2020-06-04 13:41:44, INFO: Finished adding counts variable: AADT
2020-06-04 13:41:44, INFO: Finished creating ML lanes variable: ML_lanes
2020-06-04 13:41:44, INFO: Finished creating hov corridor variable: segment_i
d
2020-06-04 13:41:44, INFO: Overwriting existing distance Variable 'distance'
already in network
```

```
c:\anaconda3\envs\break_lasso_test\lib\site-packages\lasso\roadway.py:933: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  ].astype(network_var_type)

2020-06-04 13:41:48, INFO: Calculating distance for centroid connectors
2020-06-04 13:41:48, INFO: Filling nan for network from network wrangler
2020-06-04 13:41:50, INFO: Converting variable type to MetCouncil standard
2020-06-04 13:41:52, INFO: Lasso base directory set as: D:\lasso
2020-06-04 13:41:52, INFO: Evaluating compatibility between roadway network c
hanges and base network. Not evaluating deletions.
2020-06-04 13:41:53, INFO: Evaluating project changes.
2020-06-04 13:41:53, DEBUG: Processing link deletions
2020-06-04 13:41:53, DEBUG: 2 Links Deleted.
2020-06-04 13:41:53, DEBUG: Processing link additions
2020-06-04 13:41:53, DEBUG: 2 Links Added
2020-06-04 13:41:53, DEBUG: 1 Nodes Added
2020-06-04 13:41:53, DEBUG: Processing changes
2020-06-04 13:41:53, DEBUG: 2 Changes Processed
```

In [6]:
```
test_roadway_project.roadway_changes[0:10]
```

Out[6]:

	OBJECT	OPERATION	GROUP	A	B	model_link_id	county	trn_priority	area_typ
0	L	C	0	3230	52771	224	5	0	
1	L	C	0	3261	3262	280	5	0	
2	L	C	0	3261	68075	282	5	0	
3	L	A	0	3230	3262	999998	5	0	
4	L	A	0	3262	3230	999997	5	0	
5	L	D	0	3261	131209	281	5	0	
6	L	D	0	178775	42542	477533	5	0	
7	N	A	0	NaN	NaN	NaN	NaN	NaN	Nal

8 rows × 35 columns

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_attribute_change","transit.LIN"),
    )

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.csv"),
        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_change.shp"),
        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",
        )
    )
```