James Conlon
Trip Generation

Zone-based Work Trip Generation

To aggregate household data by zone number, I used python. This let me quickly get a zone total for each variable while minimizing human error.

```
import numpy as np
import pandas as pd

data_raw = pd.read_csv('data.csv')

zones = data_raw['zone'].values
low_zone = min(zones)
high_zone = max(zones)

zone_id = []
zone_ref = []
for i in range(low_zone,high_zone+1):
    ref_ = np.where(zones == i)[0]

    if(len(ref_)>0):
        temp_ = [i,ref_]
        zone_ref.append(temp_)

variables = list(data_raw.columns.values)[1:] #ignore 'zone' variable
count = [len(arr[1]) for arr in zone_ref ]

def aggregate(data,zone_array,var_array):
    return_array = []
    for zone_reference in zone_array:
        indexes_ = zone_reference[1]

        sum_array = []
        for var in var_array:
            data_ = data[var].values[indexes_]
            sum_values = data_.sum()
            sum_array.append(sum_values)

        return_array.append(sum_array)
    return(return_array)

agg = aggregate(data_raw,zone_ref,variables)
agg_df = pd.DataFrame(agg)
agg_df.columns=variables
agg_df['count'] = count
agg_df['zone'] = [arr[0] for arr in zone_ref]
agg_df = agg_df.set_index('zone')
out_csv = agg_df.to_csv('zone_totals.csv')
```

After I had the aggregated zone data, I selected a few variables that made sense to me as a starting point. I performed a regression for:

(number of persons), (number of vehicles), (number of full-time workers),(number of work at home),(number of students),(number of 65+)

The initial results had an adjusted R-square value of 0.9918, which seemed very high for an initial guess. Looking at the t-statistics, however, indicated that this regression was not acceptable.

	Coefficients	*Standard Error*	*t Stat*
Intercept	-0.76576	1.077832215	-0.71046
npers	0.140335	0.138860012	1.01062
nveh	0.002418	0.107633736	0.022467
nftw	0.87329	0.168654635	5.177975
nwah	-0.53457	0.487145635	-1.09736
nstud	-0.1719	0.225254015	-0.76313
n65+	-0.10683	0.290896499	-0.36725

James Conlon
Trip Generation

Only the t-statistic for nftw (full-time workers) would be considered acceptable. It was clear that the r-square value for the best fitting regression would need to be extremely close to one. Other measures should also be included (e.g. sum of squares, intercept)

There were twelve variables in total that could be included in the regression. The next steps were an iterative approach to determine model fit. For this, Python was used again. The Scikit-Learn library included a linear model that can fit data. Given a set of {**X**} data and **y** results, a model can be quickly fitted. From the fit, the predicted values can be obtained, which can be used to calculate sum of squares. Typically, regression attempts to minimize the sum of squares. Other outputs include R-square, coefficients, and intercept.

The three variables I considered most important were: Sum of Squares, adjusted R-Square, and intercept. *Note:* n = 49 since there are 49 unique Zones

Sum of squares was calculated as:

$$\sum_i \left(Y_{i,actual} - Y_{i,predicted}\right)^2$$

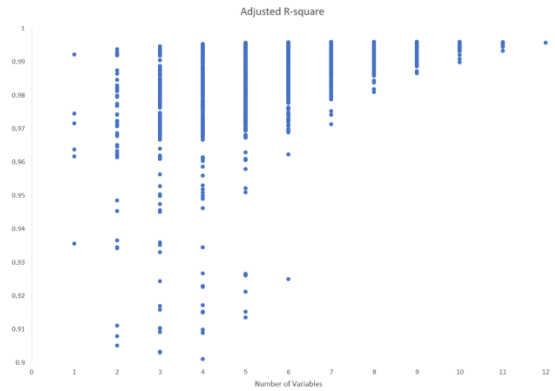
Adjusted R-Square as:

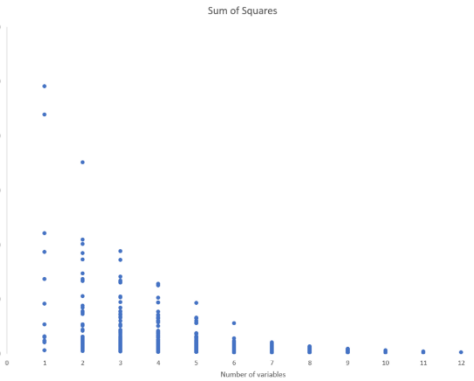
$R^2_{adj} = \left[R^2 - \left(\frac{k}{n-1}\right)\right]\left[\frac{n-1}{n-k-1}\right]$, where n = number of samples, k= number of regressors (variables)

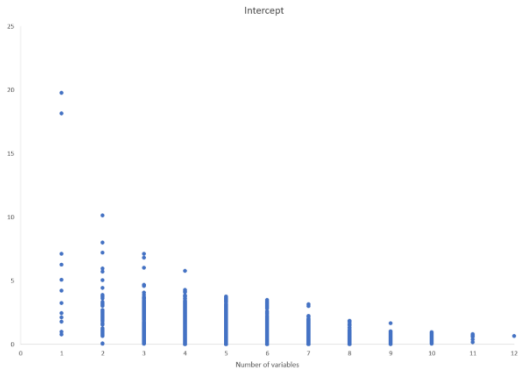
The intercept was taken as an absolute value. Its magnitude can represent the influence of unaccounted variables and should therefore be minimized.

Since there are twelve variables, there are $2^{12} - 1 = 4095$ combinations, or sets of variables that could possible be used for this regression. For each, I calculated SSQ, R-square (adjusted), and intercept.

These combinations include ones with many variables. With only 49 samples this can lead to overfitting (even using adjusted R-square). Below are how the measurements change with added variables.







James Conlon
Trip Generation

Even though there were 4095 values for each of the measurements, The first set of 12 (i.e. single variable regression for each individual variable) were useful (shown below)

n_variables	adjusted_r2	intercept	abs(intercept)	sum of squares	vars
1	0.974410719	4.218061538	4.218061538	4306.328477	('dwtype',)
1	0.963730479	-1.766224611	1.766224611	6103.667831	('npers',)
1	0.890670784	-2.439492131	2.439492131	18398.62204	('nveh',)
1	0.961637579	-2.127045006	2.127045006	6455.874446	('nlic',)
1	0.992026674	-0.995012446	0.995012446	1341.802511	('nftw',)
1	0.836223986	7.107281359	7.107281359	27561.27875	('nptw',)
1	0.415700176	19.77287384	19.77287384	98329.7241	('nwah',)
1	0.736222583	5.081612946	5.081612946	44390.15658	('nstud',)
1	0.935557557	0.766723115	0.766723115	10844.78794	('nfem',)
1	0.971455229	-3.233991203	3.233991203	4803.697278	('nmale',)
1	0.477934469	18.1516084	18.1516084	87856.53789	('nchild',)
1	0.777715141	6.245186136	6.245186136	37407.52262	('n65+',)

Both the adjusted R-square (which is just R-square in this case) and SSQ values were significantly better for **nftw (full-time workers)** than other variables. The intercept value was not the smallest, but this is alright at this point since we are only looking at one variable (there should be "room" for other variables to impact the final work trip estimate.

Using a single variable regression for full-time workers, the following was obtained

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-1.76622	2.179766	-0.81028	0.421862
nftw	0.584284	0.016354	35.72709	1.01E-35

Showing that nftw is statistically significant itself.

I then tried a regression with variables that would directly impact work trips. This included Full-time, part-time, and work at home (which should have a negative coefficient).

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-0.65472	0.962265	-0.68039	0.499742
FTW	0.984085	0.030513	32.25164	9.56E-33
PTW	0.533191	0.200526	2.658957	0.010817
Home	-0.62865	0.339796	-1.85009	0.070874

All t-statistics were acceptable in this case, except for the Intercept. There needs to be some non-direct work variables to account for that. I ran the iterative script again to obtain R-square, SSQ, and intercepts with FTW, PTW, and Work-from-home excluded. I chose to ignore adding more than 4 additional

variables to avoid overfitting. Both the lowest least square value and highest r-square included the dwelling type variable, so I added that to the regression

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	0.268	0.855	0.314	0.755
dwtype	0.133	0.032	4.144	0.000
nftw	0.730	0.067	10.971	0.000
nptw	0.581	0.172	3.368	0.002
nwah	-0.572	0.292	-1.961	0.056

The t-stats for the regressors were all acceptable, and the intercept dropped. The coefficients make sense: 0.75 daily trips for full time workers (close to the fraction 5/7ths, representing a work week). The coefficient for part time workers was lower. The coefficient for working at home was negative. Those who work at home actually produce *negative* work trips since they also belong to the full-time or part-time group.

$$T_{i,work} = 0.133 * DWTYPE + 0.730 * NFTW + 0.581 * NPTW - 0.572 * NWAH + 0.268$$

James Conlon
Trip Generation

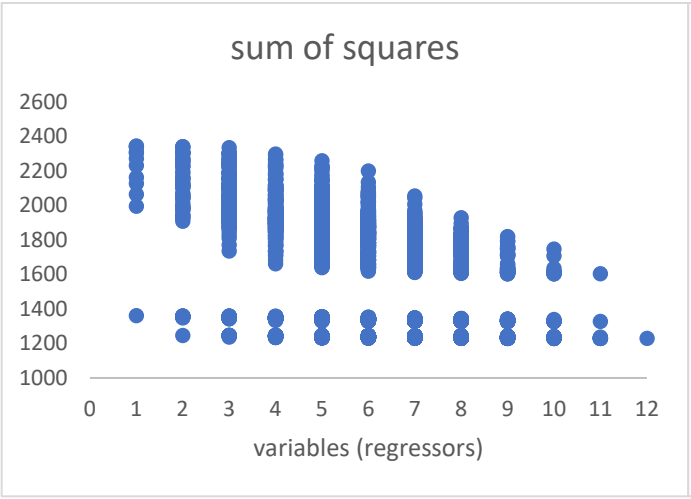
Household Work Trip Generation

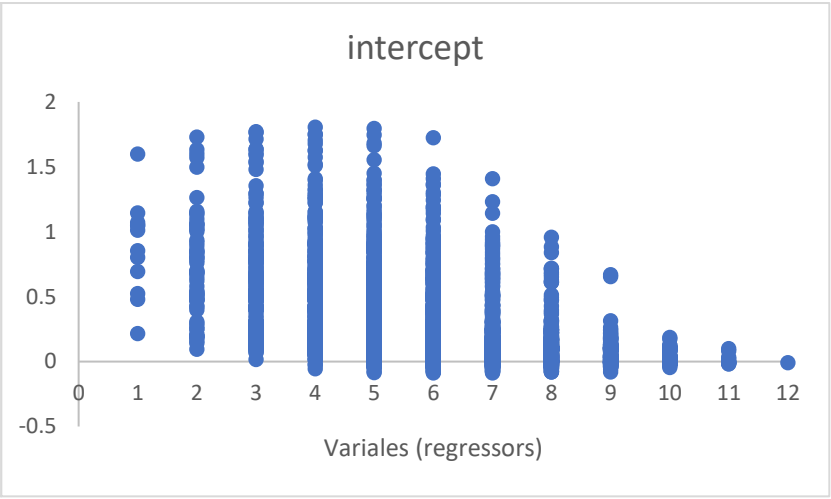
First, a comparison to the zone-based model was performed. Regression results were:

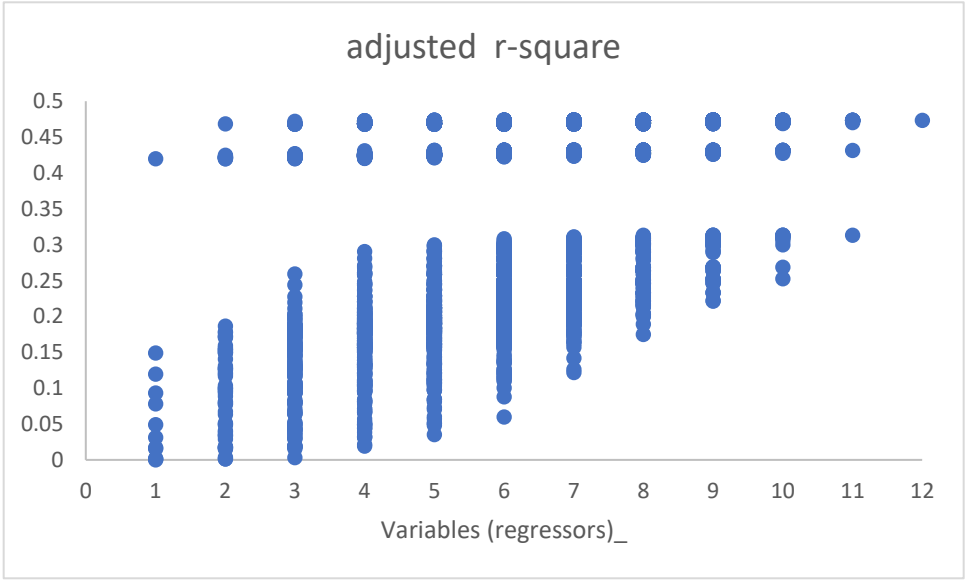
	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	-0.054	0.077	-0.708	0.479
dwtype	0.063	0.037	1.690	0.091
nftw	0.867	0.020	43.927	0.000
nptw	0.584	0.039	14.890	0.000
nwah	0.292	0.066	4.402	0.000

Showing that this initial guess is good, considering that t-stats for all regressors are > 1.65. The Work-at-home coefficient was positive, so it should be adjusted

The same iterative script was ran for household data. Plots of SSQ, R-square, and intercepts are below:







James Conlon
Trip Generation

This was interesting since more regressors increases accuracy, but there are optimum values for r-square and SSQ with a low number of regressors. Finding these points was easy since the values were already calculated and in excel.

I filtered for R-square above 0.47 and SSQ below 1250 (roughly where the flat part in the plots were located). The results for 4 variables were

r-square	intercept	abs(intercept)	SSQ	Regressors
0.4724	-0.0543	0.0543	1235.7353	('dwtype', 'nftw', 'nptw', 'nwah')
0.4728	0.1004	0.1004	1234.8156	('npers', 'nftw', 'nptw', 'nwah')
0.4721	0.0603	0.0603	1236.5590	('nveh', 'nftw', 'nptw', 'nwah')
0.4719	0.0591	0.0591	1236.9453	('nlic', 'nftw', 'nptw', 'nwah')
0.4731	0.0789	0.0789	1234.2223	('nftw', 'nptw', 'nwah', 'nstud')
0.4728	0.0904	0.0904	1234.8093	('nftw', 'nptw', 'nwah', 'nfem')
0.4719	0.0720	0.0720	1237.0864	('nftw', 'nptw', 'nwah', 'nmale')
0.4721	0.0697	0.0697	1236.5408	('nftw', 'nptw', 'nwah', 'nchild')
0.4724	0.0876	0.0876	1235.8894	('nftw', 'nptw', 'nwah', 'n65+')

Interestingly, the same combination that I tried before was in this.

Three of these have t-stats over 2.06, indicating 99% confidence

I chose the following:

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	0.0789	0.0275	2.8685	0.0042
nftw	0.8607	0.0193	44.6491	0.0000
nptw	0.6050	0.0409	14.8053	0.0000
nwah	0.2832	0.0658	4.3008	0.0000
nstud	-0.0549	0.0230	-2.3842	0.0172

This makes sense since the number of students would decrease the total number of work trips. The positive coefficient for work-at-home may not be unreasonable, since home-based businesses can still travel (e.g. handyman, consultant). Every statistic is over 99% confident.

Equation:

$$T_{hh,work} = 0.8607 * NFTW + 0.605 * NPTW + 0.283 * NWAH - 0.0549 * NSTUD + 0.0789$$

James Conlon
Trip Generation

Question 2: Household-based non-work.

R-square, SSQ, and intercepts were calculated, but with dependent variable '**nnwk**'.

Here, it is reasonable to assume that the non-work-related (i.e. not NFTW, NPTW) will be significant, and may need more regressors. For 5 regressors, the intercept (0.17) was minimized for the following:

('npers', 'nveh', 'nlic', 'nwah', 'nfem')

The R-square was 0.32, which was reasonable, considering an average of 0.29 for the iterations. The set with highest R-square included 'nstud', so I added that to the regression. This could account for students driving to school. In fact, switching the number of females with the number of students resulted in an acceptable regression:

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	0.5840	0.0865	6.7491	0.0000
npers	0.5096	0.0507	10.0477	0.0000
nveh	0.4797	0.0692	6.9343	0.0000
nlic	0.3702	0.0638	5.8039	0.0000
nwah	0.3959	0.1725	2.2958	0.0218
nstud	1.2327	0.0741	16.6345	0.0000

This is still not accounting for young children and older (65+) people, who theoretically can only make non-work related trips. Added those and took out licensed drivers since it is likely a proxy for number of people and number of vehicles for a given household.

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	0.634	0.090	7.042	0.000
npers	0.714	0.055	13.094	0.000
nveh	0.631	0.061	10.334	0.000
nchild	-0.319	0.100	-3.188	0.001
n65+	-0.317	0.088	-3.612	0.000
nwah	0.455	0.174	2.620	0.009
nstud	1.221	0.079	15.494	0.000

For a child or elderly person in a household, the expected number of trips should decrease. This means that parents need to stay home to watch the children, or that the older people are incapable or don't need to drive anywhere. This following is over 99% confident:

$$T_{hh,non-work} = .714 * NPERS + .631 * NVEH - .319 * NCHILD - .317 * N65^{+} + .455 * NWAH + 1.221 * NSTUD + .634$$

James Conlon
Trip Generation

Question 3: cross-classification.

To get the household sized normalization, the 'nnwk' column can be divided by 'npers', or number of household members

Comparing rows (HH size) and columns (# vehicles)

Count of nnwk	Column Labels							
Row Labels	**0**	**1**	**2**	**3**	**4**	**5**	**(blank)**	**Grand Total**
1	678	380	19	5	1			1083
2	272	376	115	3				766
3	78	113	36	12		1		240
4	31	82	24	4	1			142
5	17	23	9		1			50
6	2	10	5	2	1			20
7	2	2		1				5
8		2						2
9		1						1
10	1							1
(blank)								
Grand Total	**1081**	**989**	**208**	**27**	**4**	**1**		**2310**

Many single person households do not have a vehicle, so this group of zero vehicles should be a classification. There is a significant drop-off past one vehicle, so the classes will be: 0,1,2+ for vehicles. A similar classification can be used for HH size, except starting at one: 1,2,3+. This can be considered as single residents, couples, and families.
(count)

		#vehicles		
		0	**1**	**2+**
HHSize	**1**	678	380	25
	2	272	376	118
	3+	131	233	97

(SUM)

		# vehicles		
		0	**1**	**2+**
	1	924	769	63
	2	692	1160	390
HH size	**3+**	520	1147	557

James Conlon
Trip Generation

Using these, the expected ratios can be found: This is the cross-classification model

		# vehicles		
		0	**1**	**2+**
	1	1.36	2.02	2.52
	2	2.54	3.09	3.31
HH size	**3+**	3.97	4.92	5.74

Using VLookups, these ratios can be assigned to classifications of n_veh and HHsize. This multiplied by npers can get the predicted value. Here are the regression results:

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	0.0032	0.1034	0.0308	0.9754
X Variable 1	0.9993	0.0347	28.7874	0.0000

The X value has a very high t-stat and the p-value for the intercept indicates it is 97% confident. This model is more simple to use than the regression model in question 2, while still maintaining accuracy. I believe that households are better described qualitatively, which closer represents classification. Similar households are grouped based on their size and vehicle behavior. This model also prevents possible over-fitting that can occur in multivariable linear regression. This model is better for non-work, however for work trip forecasting, I would still use a standard regression.

James Conlon
Trip Generation

This is the iterative script for getting measurements for regressors:

```
import numpy as np
import pandas as pd
import itertools
from sklearn import linear_model

zone_data = pd.read_csv('zone_totals.csv')

variables = list(zone_data.columns.values)[1:-4]

it_arr = []
for i in range(1,len(variables)+1):
    it_ = itertools.combinations(variables,i)
    it_arr.extend(it_)


def r_adjust(R2,k,n):
    if(n==1):
        return(R2)
    r_adj = (R2-(k/(n-1)))*((n-1)/(n-k-1))
    return(r_adj)

def sum_squares(predict,y):
    stack_ = np.column_stack((predict,y))
    squares = [(row[0] - row[1])**2 for row in stack_]

    return(sum(squares))


def reg(data,var,y_str):
    X = data[var].values
    y = data[y_str].values
    reg = linear_model.LinearRegression()
    model = reg.fit(X,y)
    r_square = model.score(X,y)
    k = len(var)
    n = len(X)
    r_square_adjust = r_adjust(r_square,k,n)
    coef = model.coef_
    intercept = model.intercept_
    predict = model.predict(X)
    ssq = sum_squares(predict,y)

    return(predict,r_square,r_square_adjust,coef,intercept,ssq)

predict = []
r2 = []
r2_adj = []
coef = []
intercept = []
ssq = []
n_var = []
for it_ in it_arr:

    predict,r_square_,r_square_adjust_,coef_,intercept_,ssq_ = reg(zone_data,list(it_),'nwork')
    n_var.append(len(list(it_)))
    r2.append(r_square_)
    r2_adj.append(r_square_adjust_)
    coef.append(coef_)
    intercept.append(intercept_)
    ssq.append(ssq_)

stack = np.column_stack((n_var,r2_adj,intercept,ssq))
out_df = pd.DataFrame(stack)
out_df.columns = ['n_variables','adjusted_r2','intercept','sum of squares']
out_df['vars']=it_arr
out_df.to_csv('iterative_data.csv')
```