

Enrichment of Dataset and Documentation

For the descriptive statistics, the data was filtered in Microsoft Excel in order to create a dataset without the unknown solos and unknown vagrant category types. This new dataset is in our dataset with the sheet name 'Data'. The pie charts were made using this data in our excel file with the sheet name as 'Sheet 1'. The regressions were run with the original dataset.

```
In [35]: #import pandas as pd
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression

import pandas as pd
import warnings

warnings.filterwarnings("ignore", category=UserWarning)
warnings.filterwarnings("ignore", category=FutureWarning)
pd.set_option("display.max_columns", None)
pd.set_option("display.max_rows", 100)
```

Overall

```
In [2]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
solo_male = 0
single_female = 0
dependent = 0
group_leader = 0
unknown_solo = 0
for i in range(0,n):
    if data.loc[i, 'Person Type'] == 'Solo Male':
        solo_male += 1
    if data.loc[i, 'Person Type'] == 'Single Female':
        single_female += 1
    if data.loc[i, 'Person Type'] == 'Dependent':
        dependent += 1
    if data.loc[i, 'Person Type'] == 'Group Leader':
        group_leader += 1
    if data.loc[i, 'Person Type'] == 'Unknown Solo':
        unknown_solo += 1
print('percentage of solo males:', solo_male / n)
print('percentage of single females:', single_female / n)
print('percentage of dependents:', dependent / n)
print('percentage of group leaders:', group_leader / n)
print('percentage of unknown solos:', unknown_solo / n)
solo_male, single_female, dependent, group_leader, unknown_solo, n

percentage of solo males: 0.3233996334261082
percentage of single females: 0.2716793286035775
percentage of dependents: 0.2558549966057973
percentage of group leaders: 0.14398207860973458
percentage of unknown solos: 0.0
(4764, 4077, 3769, 2121, 0, 14731)
```

City Vagrants

```
In [3]: n = len(data)
data_city_vagrants = data
for i in range(0,n):
    if data.loc[i, 'Vagrant Category'] != 'City Vagrant':
        data_city_vagrants = data_city_vagrants.drop(i)
data_city_vagrants = data_city_vagrants.reset_index(drop=True)
data_city_vagrants

Out[3]:
```

	Given Names	Surname	Gender of Lead Vagrant	Relationship to Lead Vagrant	Number of People in Group	Person Type	Vagrant Category	Magistrate Name	Taken From	Conveyed To
0	Mitchell	Bruce	M	[lead vagrant]	1	Solo Male	City Vagrant	John Hart	House	Cheshunt
1	John	Drive	M	[lead vagrant]	1	Solo Male	City Vagrant	John Hart	House	Cheshunt
2	Peter	Smith	M	[lead vagrant]	1	Solo Male	City Vagrant	P LeMesurier	House	Ridge
3	Thomas	Herry	M	[lead vagrant]	1	Solo Male	City Vagrant	Richard Clark	House	Cheshunt
4	James	Guttry	M	[lead vagrant]	1	Solo Male	City Vagrant	Nathaniel Newnam	House	Cheshunt
...
7130	Andrew	Brown	M	[lead vagrant]	1	Solo Male	City Vagrant	John Hart	House	Cheshunt
7131	Mary	McDaniel	F	[lead vagrant]	1	Single Female	City Vagrant	Thomas Woodridge	House	Cheshunt
7132	John	Jones	M	[lead vagrant]	1	Solo Male	City Vagrant	Thomas Wright	House	Cheshunt
7133	Elizabeth	Steward	F	[lead vagrant]	1	Single Female	City Vagrant	Brass Crosby	House	Egham
7134	John	Gifford	M	[lead vagrant]	1	Solo Male	City Vagrant	Richard Clark	House	Egham

7135 rows x 10 columns

```
In [4]: n = len(data_city_vagrants)
solo_male = 0
single_female = 0
dependent = 0
group_leader = 0
unknown_solo = 0
for i in range(0,n):
    if data_city_vagrants.loc[i, 'Person Type'] == 'Solo Male':
        solo_male += 1
    if data_city_vagrants.loc[i, 'Person Type'] == 'Single Female':
        single_female += 1
    if data_city_vagrants.loc[i, 'Person Type'] == 'Dependent':
        dependent += 1
    if data_city_vagrants.loc[i, 'Person Type'] == 'Group Leader':
        group_leader += 1
    if data_city_vagrants.loc[i, 'Person Type'] == 'Unknown Solo':
        unknown_solo += 1
print('percentage of solo males:', solo_male / n)
print('percentage of single females:', single_female / n)
print('percentage of dependents:', dependent / n)
print('percentage of group leaders:', group_leader / n)
print('percentage of unknown solos:', unknown_solo / n)
solo_male, single_female, dependent, group_leader, unknown_solo, n

percentage of solo males: 0.3991590749824807
percentage of single females: 0.21625788367203924
percentage of dependents: 0.24176594253679046
percentage of group leaders: 0.14281709880688956
percentage of unknown solos: 0.0
(2848, 1543, 1725, 1019, 0, 7135)
```

Middlesex Vagrants and Westminster Vagrants

```
In [5]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
data_correctional_facilities = data
for i in range(0,n):
    if data_correctional_facilities.loc[i, 'Vagrant Category'] != 'Westminster Vagrant' and data_correctional_facilities.loc[i, 'Vagrant Category'] != 'Middlesex Vagrant':
        data_correctional_facilities = data_correctional_facilities.drop(i)
data_correctional_facilities = data_correctional_facilities.reset_index(drop=True)
data_correctional_facilities

Out[5]:
```

	Given Names	Surname	Gender of Lead Vagrant	Relationship to Lead Vagrant	Number of People in Group	Person Type	Vagrant Category	Magistrate Name	Taken From	Conveyed To
0	Laurence	Least	M	[lead vagrant]	1	Solo Male	Middlesex Vagrant	John Staples	Clerkenwell	Cheshunt
1	John	McFarland	M	[lead vagrant]	1	Solo Male	Middlesex Vagrant	William Hyde	Clerkenwell	Cheshunt
2	Christiana	Gray	F	[lead vagrant]	1	Single Female	Westminster Vagrant	William Hyde	Tothillfields	Cheshunt
3	Donald	Ross	M	[lead vagrant]	1	Solo Male	Westminster Vagrant	Charles Triquet	Tothillfields	Ridge
4	[Child]	McKenzie	[unknown]	[Child]	2	Dependent	Westminster Vagrant	Michael Downes	Tothillfields	Ridge
...
4319	Jane	Hardy	F	[lead vagrant]	1	Single Female	Middlesex Vagrant	William Hyde	Clerkenwell	Ran at Hounslow
4320	[Child]	Harry	[unknown]	[Child]	2	Dependent	Middlesex Vagrant	William Hyde	Clerkenwell	Egham
4321	Jane	Harry	F	[lead vagrant]	2	Group Leader	Middlesex Vagrant	William Hyde	Clerkenwell	Egham
4322	Elizabeth	Ramsey	F	[lead vagrant]	1	Single Female	Middlesex Vagrant	EB Green	Clerkenwell	Egham
4323	Elizabeth	Robertson	F	[lead vagrant]	1	Single Female	Middlesex Vagrant	EB Green	Clerkenwell	Egham

4324 rows x 10 columns

```
In [5]: n = len(data_correctional_facilities)
solo_male = 0
single_female = 0
dependent = 0
group_leader = 0
unknown_solo = 0
for i in range(0,n):
    if data_correctional_facilities.loc[i, 'Person Type'] == 'Solo Male':
        solo_male += 1
    if data_correctional_facilities.loc[i, 'Person Type'] == 'Single Female':
        single_female += 1
    if data_correctional_facilities.loc[i, 'Person Type'] == 'Dependent':
        dependent += 1
    if data_correctional_facilities.loc[i, 'Person Type'] == 'Group Leader':
        group_leader += 1
    if data_correctional_facilities.loc[i, 'Person Type'] == 'Unknown Solo':
        unknown_solo += 1
print('percentage of solo males:', solo_male / n)
print('percentage of single females:', single_female / n)
print('percentage of dependents:', dependent / n)
print('percentage of group leaders:', group_leader / n)
print('percentage of unknown solos:', unknown_solo / n)
solo_male, single_female, dependent, group_leader, unknown_solo, n

percentage of solo males: 0.27358926919518967
percentage of single females: 0.3404255319148936
percentage of dependents: 0.24768732654949122
percentage of group leaders: 0.13829787234042554
percentage of unknown solos: 0.0
(1183, 1472, 1071, 598, 0, 4324)
```

Passed To Middlesex Home

```
In [6]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
data_back_home = data
for i in range(0,n):
    if data_back_home.loc[i, 'Vagrant Category'] != 'Passed to Middlesex Home':
        data_back_home = data_back_home.drop(i)
data_back_home = data_back_home.reset_index(drop=True)
data_back_home

Out[6]:
```

	Given Names	Surname	Gender of Lead Vagrant	Relationship to Lead Vagrant	Number of People in Group	Person Type	Vagrant Category	Magistrate Name	Taken From	Conveyed To
0	William	Rockwell	M	[lead vagrant]	1	Solo Male	Passed to Middlesex Home	William Tyler	Tothillfields	Willesden
1	John	Hognot	M	[lead vagrant]	1	Solo Male	Passed to Middlesex Home	John Sherwood	Clerkenwell	Willesden
2	[Child]	Barr	[unknown]	[Child]	5	Dependent	Passed to Middlesex Home	Charles Oakes	South Mimms	Willesden
3	[Child]	Barr	[unknown]	[Child]	5	Dependent	Passed to Middlesex Home	Charles Oakes	South Mimms	Willesden
4	[Child]	Barr	[unknown]	[Child]	5	Dependent	Passed to Middlesex Home	Charles Oakes	South Mimms	Willesden
...
2176	Charles	Lee	M	[lead vagrant]	1	Solo Male	Passed to Middlesex Home	T Treslove	House	Action
2177	Elizabeth	Griffin	F	[lead vagrant]	1	Single Female	Passed to Middlesex Home	Thomas Collins	Clerkenwell	Action
2178	Elizabeth	Nixon	F	[lead vagrant]	1	Single Female	Passed to Middlesex Home	Philip Dyot	Clerkenwell	Action
2179	Eleanor	Giver	F	[lead vagrant]	1	Single Female	Passed to Middlesex Home	David Walker	Clerkenwell	Action
2180	Susannah	Little	F	[lead vagrant]	1	Single Female	Passed to Middlesex Home	Charles Martyn	Tothillfields	[none given]

2181 rows x 10 columns

```
In [8]: n = len(data_back_home)
solo_male = 0
single_female = 0
dependent = 0
group_leader = 0
unknown_solo = 0
for i in range(0,n):
    if data_back_home.loc[i, 'Person Type'] == 'Solo Male':
        solo_male += 1
    if data_back_home.loc[i, 'Person Type'] == 'Single Female':
        single_female += 1
    if data_back_home.loc[i, 'Person Type'] == 'Dependent':
        dependent += 1
    if data_back_home.loc[i, 'Person Type'] == 'Group Leader':
        group_leader += 1
    if data_back_home.loc[i, 'Person Type'] == 'Unknown Solo':
        unknown_solo += 1
print('percentage of solo males:', solo_male / n)
print('percentage of single females:', single_female / n)
print('percentage of dependents:', dependent / n)
print('percentage of group leaders:', group_leader / n)
print('percentage of unknown solos:', unknown_solo / n)
solo_male, single_female, dependent, group_leader, unknown_solo, n

percentage of solo males: 0.22054103622191656
percentage of single females: 0.3764328289775332
percentage of dependents: 0.26455734241173773
percentage of group leaders: 0.13846859238881248
percentage of unknown solos: 0.0
(481, 821, 577, 302, 0, 2181)
```

Passing Vagrant

```
In [13]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
data_passing_through = data
for i in range(0,n):
    if data_passing_through.loc[i, 'Vagrant Category'] != 'Passing Vagrant':
        data_passing_through = data_passing_through.drop(i)
data_passing_through = data_passing_through.reset_index(drop=True)
data_passing_through

Out[13]:
```

	Given Names	Surname	Gender of Lead Vagrant	Relationship to Lead Vagrant	Number of People in Group	Person Type	Vagrant Category	Magistrate Name	Taken From	Conveyed To
0	[Child]	Mackenzie	[unknown]	[Child]	2	Dependent	Passing Vagrant	John Bullock	Colnbrook	Cheshunt
1	Isabella	Mackenzie	F	[lead vagrant]	2	Group Leader	Passing Vagrant	John Bullock	Colnbrook	Cheshunt
2	George	Ferguson	M	[lead vagrant]	1	Solo Male	Passing Vagrant	William Parker	Colnbrook	Cheshunt
3	[Child]	Taylor	[unknown]	[Child]	4	Dependent	Passing Vagrant	Thomas Davies	Colnbrook	Cheshunt
4	[Child]	Taylor	[unknown]	[Child]	4	Dependent	Passing Vagrant	Thomas Davies	Colnbrook	Cheshunt
...
1086	Isaac	Curry	M	[lead vagrant]	1	Solo Male	Passing Vagrant	Thomas D'Avant	South Mimms	St Botolph
1087	Jacob	Gould	M	[lead vagrant]	1	Solo Male	Passing Vagrant	Thomas D'Avant	South Mimms	Sent on
1088	John	Davis	M	[lead vagrant]	1	Solo Male	Passing Vagrant	James Charles	South Mimms	Sent on
1089	Thomas	Vincent	M	[lead vagrant]	1	Solo Male	Passing Vagrant	B Spencer	South Mimms	Sent on
1090	Eleanor	Farkenson	F	[lead vagrant]	1	Single Female	Passing Vagrant	Andrew Wilkinson	Enfield	Egham

1091 rows x 10 columns

```
In [14]: n = len(data_passing_through)
solo_male = 0
single_female = 0
dependent = 0
group_leader = 0
unknown_solo = 0
for i in range(0,n):
    if data_passing_through.loc[i, 'Person Type'] == 'Solo Male':
        solo_male += 1
    if data_passing_through.loc[i, 'Person Type'] == 'Single Female':
        single_female += 1
    if data_passing_through.loc[i, 'Person Type'] == 'Dependent':
        dependent += 1
    if data_passing_through.loc[i, 'Person Type'] == 'Group Leader':
        group_leader += 1
    if data_passing_through.loc[i, 'Person Type'] == 'Unknown Solo':
        unknown_solo += 1
print('percentage of solo males:', solo_male / n)
print('percentage of single females:', single_female / n)
print('percentage of dependents:', dependent / n)
print('percentage of group leaders:', group_leader / n)
print('percentage of unknown solos:', unknown_solo / n)
solo_male, single_female, dependent, group_leader, unknown_solo, n

percentage of solo males: 0.230980751604033
percentage of single females: 0.220892584946012
percentage of dependents: 0.3629697525026233
percentage of group leaders: 0.185152373968836
percentage of unknown solos: 0.0
(252, 241, 396, 202, 0, 1091)
```

Unknown

```
In [5]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
data_unknown = data
for i in range(0,n):
    if data_unknown.loc[i, 'Vagrant Category'] != 'Unknown':
        data_unknown = data_unknown.drop(i)
data_unknown = data_unknown.reset_index(drop=True)
data_unknown

Out[5]:
```

	Given Names	Surname	Gender of Lead Vagrant	Relationship to Lead Vagrant	Number of People in Group	Person Type	Vagrant Category	Magistrate Name	Taken From	Conveyed To
0	Mary	Simpson	F	[lead vagrant]	1	Single Female	Unknown	Robertson	[illegible]	Ridge
1	William	A[illegible]	M	[lead vagrant]	1	Solo Male	Unknown	John Wilkes	[illegible]	Ridge
2	Constance	Woodburn	F	[lead vagrant]	1	Single Female	Unknown	[none]	[blacked out transcript-ac]	[blacked out transcript-ac]
3	James	Johnson	M	[lead vagrant]	1	Solo Male	Unknown	Richard Clark	[none given]	[none given]
4	John	Walker	M	[lead vagrant]	1	Solo Male	Unknown	[none]	[blacked out transcript-ac]	[blacked out transcript-ac]
5	Sarah	Williams	F	[lead vagrant]	1	Single Female	Unknown	[none]	[blacked out transcript-ac]	[blacked out transcript-ac]
6	William	Watkins	M	[lead vagrant]	1	Solo Male	Unknown	[none]	[blacked out transcript-ac]	[blacked out transcript-ac]
7	Sarah	Wilcock	F	[lead vagrant]	1	Single Female	Unknown	[none]	[blacked out transcript-ac]	[blacked out transcript-ac]
8	John	West	M	[lead vagrant]	1	Solo Male	Unknown	[none]	[blacked out transcript-ac]	[blacked out transcript-ac]
9	William	Williams	M	[lead vagrant]	1	Solo Male	Unknown	[none]	[blacked out transcript-ac]	[blacked out transcript-ac]
10	[Child]	Christie	[unknown]	[Child]	4	Dependent	Unknown	[none]	[none given]	St Botolph
11	[Child]	Christie	[unknown]	[Child]	4	Dependent	Unknown	[none]	[none given]	St Botolph
12	[Wife]	Christie	F	[Wife]	4	Dependent	Unknown	[none]	[none given]	St Botolph
13	Peter	Christie	M	[lead vagrant]	4	Group Leader	Unknown	[none]	[none given]	St Botolph
14	[none given]	Onion	[unknown]	[lead vagrant]	1	Unknown Solo	Unknown	[none]	[none given]	Ridge
15	[none given]	Roberts	[unknown]	[lead vagrant]	1	Unknown Solo	Unknown	[none]	[none given]	[none given]
16	Maria	Shepperd	F	[lead vagrant]	1	Single Female	Unknown	[none]	[none given]	[none given]

```
In [7]: n = len(data_unknown)
solo_male = 0
single_female = 0
dependent = 0
group_leader = 0
unknown_solo = 0
for i in range(0,n):
    if data_unknown.loc[i, 'Person Type'] == 'Solo Male':
        solo_male += 1
    if data_unknown.loc[i, 'Person Type'] == 'Single Female':
        single_female += 1
    if data_unknown.loc[i, 'Person Type'] == 'Dependent':
        dependent += 1
    if data_unknown.loc[i, 'Person Type'] == 'Group Leader':
        group_leader += 1
    if data_unknown.loc[i, 'Person Type'] == 'Unknown Solo':
        unknown_solo += 1
print('percentage of solo males:', solo_male / n)
print('percentage of single females:', single_female / n)
print('percentage of dependents:', dependent / n)
print('percentage of group leaders:', group_leader / n)
print('percentage of unknown solos:', unknown_solo / n)
solo_male, single_female, dependent, group_leader, unknown_solo, n

percentage of solo males: 0.35294117647058826
percentage of single females: 0.29411764705882354
percentage of dependents: 0.17647058823529413
percentage of group leaders: 0.05923529411764705
percentage of unknown solos: 0.11764705882352941
(6, 5, 3, 1, 2, 17)
```

Regression for 'Is City Vagrant'

```
In [36]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
data['is city vagrant'] = 0
for i in range(0,n):
    if data.loc[i, 'Vagrant Category'] == 'City Vagrant':
        data.loc[i, 'is city vagrant'] = 1
person_type = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx")
person_type_dummies = pd.get_dummies(person_type)
person_type_dummies = person_type_dummies.drop('Person Type_Unknown Solo', axis = 1)
person_type_dummies = sm.add_constant(person_type_dummies)
model = sm.OLS(data['is city_vagrant'], person_type_dummies)
results = model.fit()
results.summary()
```

```
Out[36]:
```

OLS Regression Results				
Dep. Variable:	is city_vagrant	R-squared:	0.012	
Model:	OLS	Adj. R-squared:	0.039	
Method:	Least Squares	F-statistic:	113.0	
Date:	Fri, 21 Oct 2022	Prob (F-statistic):	4.71e-95	
Time:	15:10:11	Log-Likelihood:	-10503.	
No. Observations:	14789	AIC:	2.102e+04	
Df Residuals:	14784	BIC:	2.105e+04	
Df Model:	4			
Covariance Type:	nonrobust			
	coef	std err	t	P> t
const	0.00930	0.040	2.338	0.019
Person Type,Dependent	-0.00311	0.075	-0.411	0.681
Person Type,Group Leader	0.0022	0.040	0.299	0.765
Person Type,Single Female	-0.0340	0.040	-0.850	0.396
Person Type,Solo Male	-0.0402	0.040	-1.006	0.315
Omnibus:	54864.387	Durbin-Watson:	0.903	
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2176.507	
Skew:	0.064	Prob(JB):	0.00	
Kurtosis:	1.125	Cond. No.	46.8	

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Regression for 'Is Middlesex/Westminster Vagrant'

```
In [37]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
data['is from correctional facility'] = 0
for i in range(0,n):
    if data.loc[i, 'Vagrant Category'] == 'Westminster Vagrant' or data.loc[i, 'Vagrant Category'] == 'Middlesex Vagrant':
        data.loc[i, 'is from correctional facility'] = 1
person_type = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx")
person_type_dummies = pd.get_dummies(person_type)
person_type_dummies = person_type_dummies.drop('Person Type_Unknown Solo', axis = 1)
person_type_dummies = sm.add_constant(person_type_dummies)
model = sm.OLS(data['is from correctional facility'], person_type_dummies)
results = model.fit()
results.summary()
```

```
Out[37]:
```

OLS Regression Results				
Dep. Variable:	is from correctional facility	R-squared:	0.010	
Model:	OLS	Adj. R-squared:	0.009	
Method:	Least Squares	F-statistic:	35.56	
Date:	Fri, 21 Oct 2022	Prob (F-statistic):	1.31e-29	
Time:	15:13:58	Log-Likelihood:	-9327.25	
No. Observations:	14789	AIC:	1.855e+04	
Df Residuals:	14784	BIC:	1.859e+04	
Df Model:	4			
Covariance Type:	nonrobust			
	coef	std err	t	P> t
const	0.02093	0.040	3.030	0.002
Person Type,Dependent	0.0746	0.069	1.074	0.283
Person Type,Group Leader	0.0725	0.070	1.039	0.299
Person Type,Single Female	0.1513	0.069	2.179	0.029
Person Type,Solo Male	0.0387	0.069	0.558	0.577
Omnibus:	9884.654	Durbin-Watson:	1.087	
Prob(Omnibus):	0.000	Jarque-Bera (JB):	69600.159	
Skew:	3.217	Prob(JB):	0.00	
Kurtosis:	11.459	Cond. No.	46.8	

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Regression for 'Is Passed to Middlesex Home'

```
In [41]: data = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx",
n = len(data)
data['is coming home'] = 0
for i in range(0,n):
    if data.loc[i, 'Vagrant Category'] == 'Passed to Middlesex Home':
        data.loc[i, 'is coming home'] = 1
person_type = pd.read_excel(r"C:\Users\Mufct\Desktop\Minor DH and SA\Intro to DH\MiddlesexVagrants1777-1786v1.1.xlsx")
person_type_dummies = pd.get_dummies(person_type)
person_type_dummies = person_type_dummies.drop('Person Type_Unknown Solo', axis = 1)
person_type_dummies = sm.add_constant(person_type_dummies)
model = sm.OLS(data['is coming home'], person_type_dummies)
results = model.fit()
results.summary()
```

```
Out[41]:
```

OLS Regression Results				
Dep. Variable:	is coming home	R-squared:	0.012	
Model:	OLS	Adj. R-squared:	0.012	
Method:	Least Squares	F-statistic:	44.75	
Date:	Fri, 21 Oct 2022	Prob (F-statistic):	2.07e-37	
Time:	16:40:38	Log-Likelihood:	-5581.	