Logistic Regression Assignment Template

Use this notebook as a template to start your own analysis. Your analysis should be more in-depth than the limited analysis you see here (i.e. you should explore the relationships between more of the variables then what you see in this notebook). Think of this notebook as a jumping off point to get you started with the code. Customize for your own analysis!

You should use Tableau for your final visualizations for the video.

In [11]:
```
#start by importing the appropriate packages and data from the CSV file
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.formula.api import logit, probit, ols
import statsmodels as sm
import statsmodels.formula.api as sm
import statsmodels.api as sm2
data = pd.read_csv('Uber2.csv')
```

In [12]:
```
data.head()
```

Out[12]:

	wait_time	treatment	commute	trips_express	rider_cancellations	total_driver_payout	total_r
0	2	0	1	3245	256	34458.41163	
1	5	1	0	2363	203	29764.34982	
2	2	0	0	2184	118	27437.36736	
3	5	1	1	3584	355	44995.45299	
4	2	0	0	2580	181	27583.95530	

In [13]:
```
sumstats=data.describe()
sumstats.round(decimals=4)
```

Out[13]:

	wait_time	treatment	commute	trips_express	rider_cancellations	total_driver_payout	to
count	126.000	126.000	126.0000	126.0000	126.0000	126.0000	
mean	3.500	0.500	0.1587	2515.5238	177.7381	28237.2207	
std	1.506	0.502	0.3669	497.6936	51.6318	5450.9540	
min	2.000	0.000	0.0000	1638.0000	95.0000	18769.9930	
25%	2.000	0.000	0.0000	2225.0000	147.7500	24721.8883	
50%	3.500	0.500	0.0000	2427.5000	166.0000	27352.8406	
75%	5.000	1.000	0.0000	2661.7500	187.2500	30586.1075	
max	5.000	1.000	1.0000	4507.0000	355.0000	48600.4220	

In [14]:
```
#divide the data into 2 dataframes: one for the treatment group and one for the
df_treatment = data[data['treatment'] == 1]
df_control = data[data['treatment'] == 0]
```

In [15]:
```
#check to the dataframe to make sure that only the treatment group is in the tre
df_treatment
```

Out[15]:

	wait_time	treatment	commute	trips_express	rider_cancellations	total_driver_payout	total
1	5	1	0	2363	203	29764.34982	
3	5	1	1	3584	355	44995.45299	
5	5	1	0	2022	135	23888.11085	
7	5	1	0	2018	150	25794.86992	
9	5	1	1	2539	284	34047.47390	
...	...	...	...	...	...	...	...
117	5	1	0	2059	129	22995.15900	
119	5	1	0	1970	204	24339.58356	
121	5	1	0	2655	173	28288.52115	
123	5	1	0	2359	154	23525.11595	
125	5	1	0	2421	196	26017.52602	

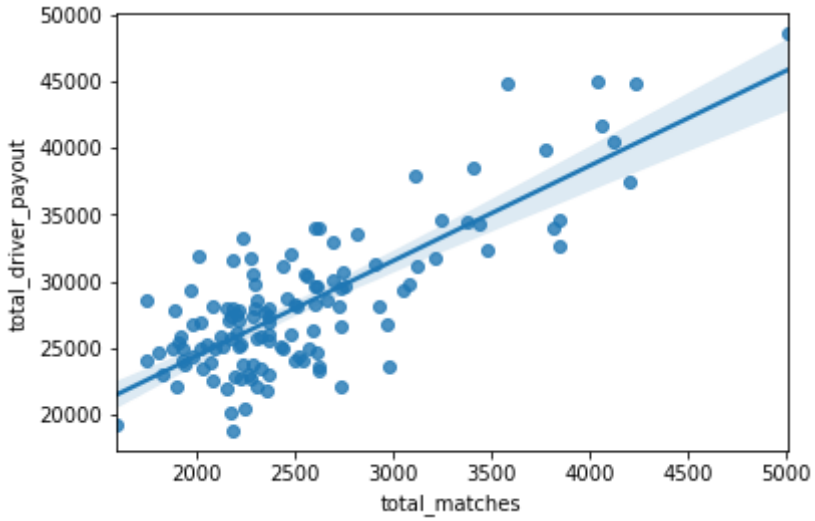
63 rows × 8 columns

In [16]:
```
#conduct a mann-whitney test to test whether a variable, in this example rider_c
#is the number of rider_cancellations statistically significantly different in t
u_statistic, p_value = stats.mannwhitneyu(df_treatment['wait_time'],df_control['
print('U-Statistic: ', u_statistic)
print('p-value: ', p_value)
```

```
U-Statistic:  0.0
p-value:  2.626795160290819e-29
```

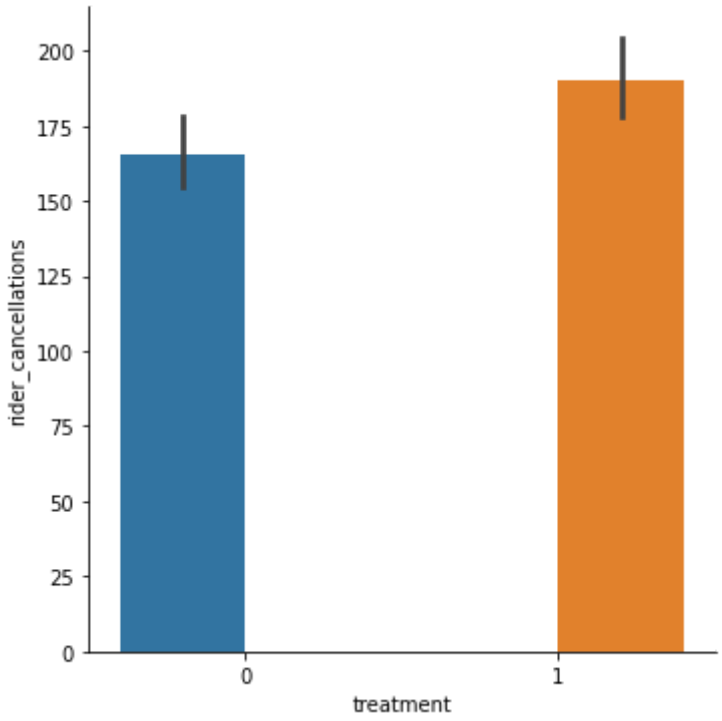
In [17]:
```
sns.regplot(x='total_matches', y='total_driver_payout', data=data)
```

Out[17]:
```
<AxesSubplot:xlabel='total_matches', ylabel='total_driver_payout'>
```



In [18]:
```
sns.catplot(x="treatment", y="rider_cancellations", hue="treatment", kind="bar",
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x7f92d87a43d0>
```



In [24]:
```
model1=ols("wait_time ~ rider_cancellations + total_driver_payout + trips_expres
model1.summary()
```

Out[24]:

OLS Regression Results

Dep. Variable:	wait_time	**R-squared:**	0.473
Model:	OLS	**Adj. R-squared:**	0.451
Method:	Least Squares	**F-statistic:**	21.56
Date:	Tue, 13 Apr 2021	**Prob (F-statistic):**	2.41e-15
Time:	18:29:51	**Log-Likelihood:**	-189.50
No. Observations:	126	**AIC:**	391.0
Df Residuals:	120	**BIC:**	408.0
Df Model:	5		
Covariance Type:	nonrobust		

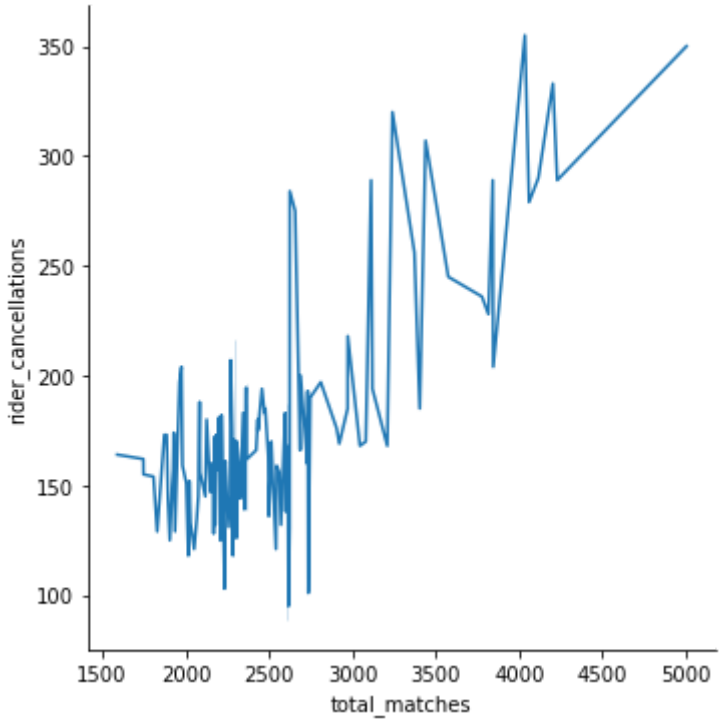
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	6.0128	0.622	9.673	0.000	4.782	7.244
rider_cancellations	0.0249	0.003	7.853	0.000	0.019	0.031
total_driver_payout	-0.0001	3.26e-05	-4.192	0.000	-0.000	-7.21e-05
trips_express	-0.0019	0.001	-2.395	0.018	-0.003	-0.000
total_matches	-0.0002	0.001	-0.361	0.718	-0.001	0.001
total_double_matches	0.0016	0.000	3.583	0.000	0.001	0.003

Omnibus:	6.820	**Durbin-Watson:**	2.894
Prob(Omnibus):	0.033	**Jarque-Bera (JB):**	3.109
Skew:	-0.010	**Prob(JB):**	0.211
Kurtosis:	2.231	**Cond. No.**	1.81e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.81e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

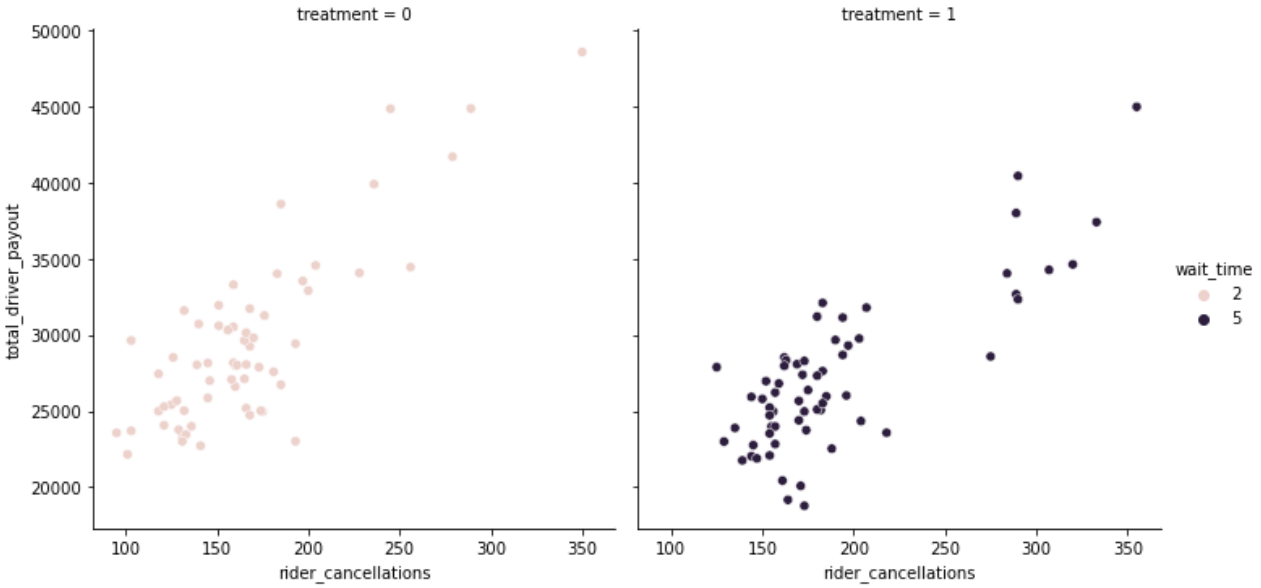
In [20]:
```
sns.relplot(x="total_matches", y="rider_cancellations", kind="line", ci="sd", da
```

Out[20]:
```
<seaborn.axisgrid.FacetGrid at 0x7f92d92f72b0>
```



In [21]:
```
sns.relplot(data=data, x="rider_cancellations", y="total_driver_payout", col="tr
            kind="scatter")
```

Out[21]:
```
<seaborn.axisgrid.FacetGrid at 0x7f92d7d05e80>
```


treatment = 0
treatment = 1
total_driver_payout
rider_cancellations
rider_cancellations
wait_time
2
5
50000
45000
40000
35000
30000
25000
20000
100
150
200
250
300
350

In [ ]: