Exercise: Analysis of taxi ride data

In this exercise you will use the the template notebook we went through at the lecture to make some basic computations and plots for a data set on taxi rides in New York.

Load the libraries you need

In [ ]:
```
import os
import sys
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from matplotlib_inline.backend_inline import set_matplotlib_formats
set_matplotlib_formats('retina', 'png')
sns.set()
sns.set_style("ticks")
# scale down size of default plots
sns.set_context("paper")
import matplotlib as mpl
scale = 0.8
d = dict([(k, v*scale) for (k, v) in sns.plotting_context('paper').items(
d['figure.figsize'] = [5.4, 3.5]
mpl.rcParams.update(d)
```

Load the data set

Pickup and dropoff are in a fancy date-time format so we exchange them for a simpler `duration` column for the number of minutes that reach taxi ride takes.

In [ ]:
```
import seaborn as sns
rides = sns.load_dataset('taxis')
rides['duration'] = [x.total_seconds()/60 for x in (rides.dropoff - rides
rides = rides[[x for x in rides.columns if x not in ['pickup', 'dropoff']]
rides.head()
```

Out[ ]:

	passengers	distance	fare	tip	tolls	total	color	payment	pickup_zone	d
0	1	1.60	7.0	2.15	0.0	12.95	yellow	credit card	Lenox Hill West	U
1	1	0.79	5.0	0.00	0.0	9.30	yellow	cash	Upper West Side South	
2	1	1.37	7.5	2.36	0.0	14.16	yellow	credit card	Alphabet City	
3	1	7.70	27.0	6.15	0.0	36.95	yellow	credit card	Hudson Sq	Y
4	3	2.16	9.0	1.10	0.0	13.40	yellow	credit card	Midtown East	Y

Exercise 1

Get the `pickup_zones` of the five taxi rides that took the longest time:

In [ ]:
```
rides.sort_values(by='duration', ascending=False).head(5)
```

Out[ ]:

	passengers	distance	fare	tip	tolls	total	color	payment	
6053	1	22.17	81.86	0.00	0.0	82.36	green	credit card	Univer
5567	1	25.51	93.50	0.00	0.0	94.80	green	credit card	
5648	2	33.46	150.00	0.00	18.9	169.70	green	cash	
5833	1	12.79	57.00	0.00	0.0	58.80	green	credit card	Queensb
4218	1	26.92	75.50	23.19	0.0	100.49	yellow	credit card	

Exercise 2

Get the mean fare for each pickup_borough:

In [ ]:
```
rides.groupby('pickup_borough').fare.mean()
```

Out[ ]:
```
pickup_borough
Bronx        20.999091
Brooklyn     16.520836
Manhattan    11.152889
Queens       24.934642
Name: fare, dtype: float64
```

Exercise 3

Are rides paid in cash shorter than those paid by credit card? (Note: Just find the mean duration for each payment type and compare them, no statistical test here, although as a bioinformatician you would probably want to do one.)

In [ ]:
```
rides.groupby('payment').duration.mean()
```

Out[ ]:
```
payment
cash           12.629323
credit card    15.067679
Name: duration, dtype: float64
```

Exercise 4

In which borrough do taxis most often pick up more than one passenger? (Note: The mean is not the answer here)

In [ ]:
```
x=rides.groupby('pickup_borough').passengers.count()
y=rides[rides.passengers > 1].groupby('pickup_borough').passengers.count(
y/x
```

Out[ ]:
```
pickup_borough
Bronx        0.101010
Brooklyn     0.148825
Manhattan    0.272589
Queens       0.231355
Name: passengers, dtype: float64
```

Exercise 5

Compute mean tip for all combinations of `pickup_borough` adn `dropoff_borough`. It should look like this:

		tip	fare
pickup_borough	dropoff_borough		
Bronx	Bronx	0.095758	14.539091
	Brooklyn	0.000000	54.062500
	Manhattan	0.335600	29.698000
	Queens	0.000000	40.157500
Brooklyn	Bronx	0.000000	58.124000
	Brooklyn	0.629362	11.877589
	Manhattan	2.331493	25.096567
	Queens	1.400769	34.842692
Manhattan	Bronx	0.891818	24.127273
	Brooklyn	3.402026	24.495098
	Manhattan	1.761994	9.727019
	Queens	5.904663	34.623804
	Staten Island	14.165000	44.500000
Queens	Bronx	1.577273	45.772727
	Brooklyn	4.145806	37.018871
	Manhattan	6.194330	36.923839
	Queens	0.843782	12.812178

In [ ]:
```
df = rides[['pickup_borough','dropoff_borough','tip', 'fare']].groupby(by
df
```

Out[ ]:

		tip	fare
pickup_borough	**dropoff_borough**		
Bronx	**Bronx**	0.095758	14.539091
	Brooklyn	0.000000	54.062500
	Manhattan	0.335600	29.698000
	Queens	0.000000	40.157500
Brooklyn	**Bronx**	0.000000	58.124000
	Brooklyn	0.629362	11.877589
	Manhattan	2.331493	25.096567
	Queens	1.400769	34.842692
Manhattan	**Bronx**	0.891818	24.127273
	Brooklyn	3.402026	24.495098
	Manhattan	1.761994	9.727019
	Queens	5.904663	34.623804
	Staten Island	14.165000	44.500000
Queens	**Bronx**	1.577273	45.772727
	Brooklyn	4.145806	37.018871
	Manhattan	6.194330	36.923839
	Queens	0.843782	12.812178

Exercise 6

Add a new column, `generosity`, to the dataframe you produced in the previous exercise. It should be `tip/fare`:

In [ ]:
```
df['generosity'] = df.tip/df.fare
df
```

Out[ ]:

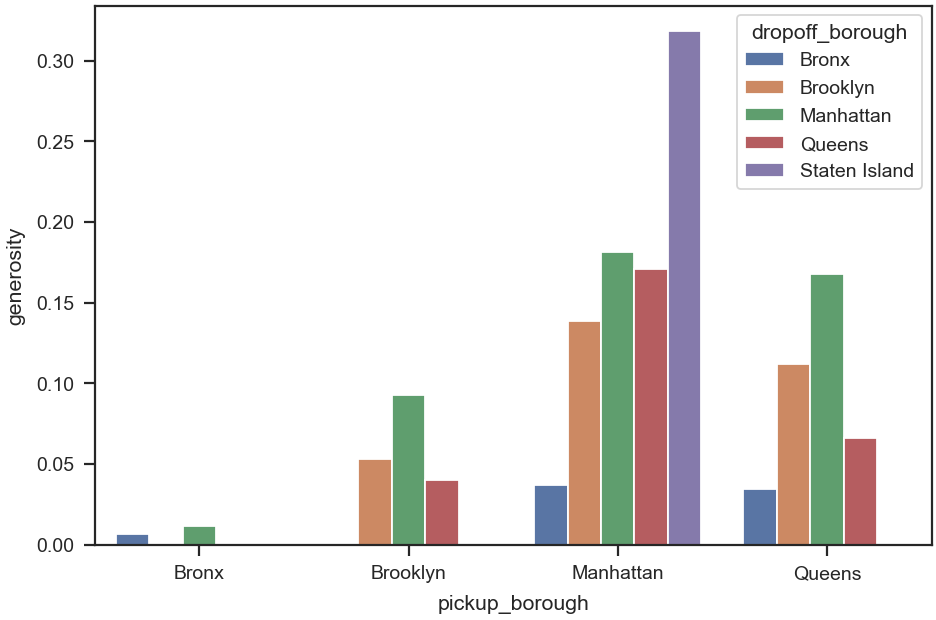
pickup_borough	dropoff_borough	tip	fare	generosity
Bronx	Bronx	0.095758	14.539091	0.006586
	Brooklyn	0.000000	54.062500	0.000000
	Manhattan	0.335600	29.698000	0.011300
	Queens	0.000000	40.157500	0.000000
Brooklyn	Bronx	0.000000	58.124000	0.000000
	Brooklyn	0.629362	11.877589	0.052987
	Manhattan	2.331493	25.096567	0.092901
	Queens	1.400769	34.842692	0.040203
Manhattan	Bronx	0.891818	24.127273	0.036963
	Brooklyn	3.402026	24.495098	0.138886
	Manhattan	1.761994	9.727019	0.181144
	Queens	5.904663	34.623804	0.170538
	Staten Island	14.165000	44.500000	0.318315
Queens	Bronx	1.577273	45.772727	0.034459
	Brooklyn	4.145806	37.018871	0.111992
	Manhattan	6.194330	36.923839	0.167760
	Queens	0.843782	12.812178	0.065858

Exercise 7

Make a barplot (`sns.barplot`) with these parameters: `data=df, x='pickup_borough', y='generosity', hue='dropoff_borough'`. Where do generous people live?

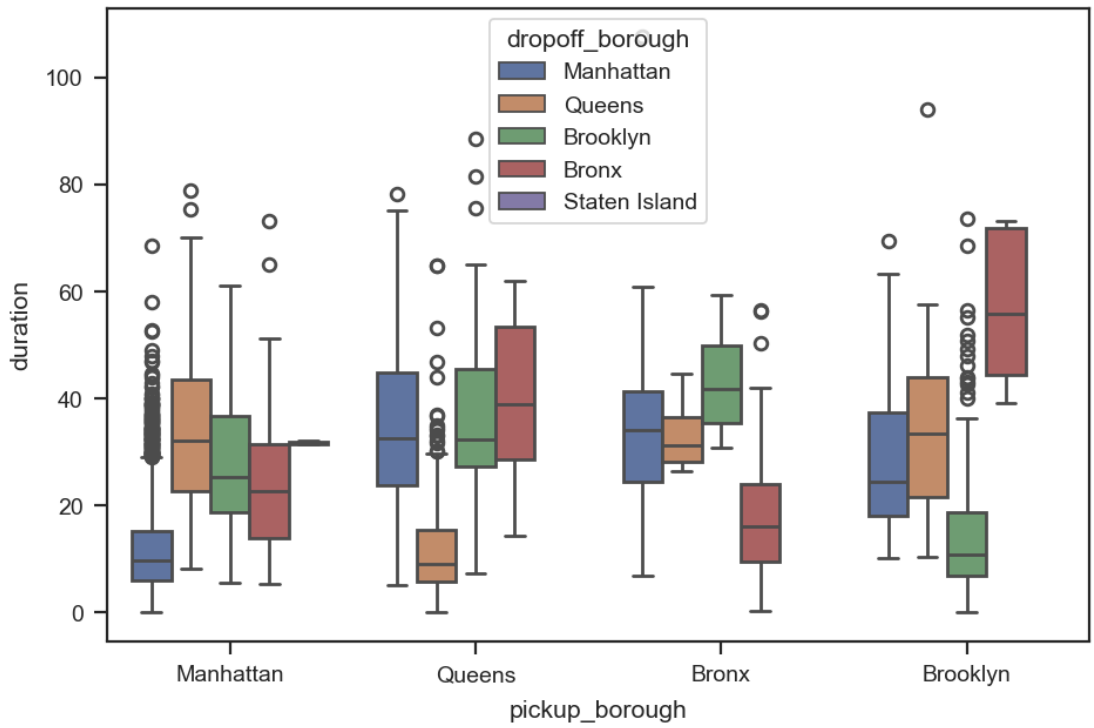
In [ ]:
```
sns.barplot(data=df, x='pickup_borough', y='generosity', hue='dropoff_bor
```

Out[ ]:
```
<Axes: xlabel='pickup_borough', ylabel='generosity'>
```



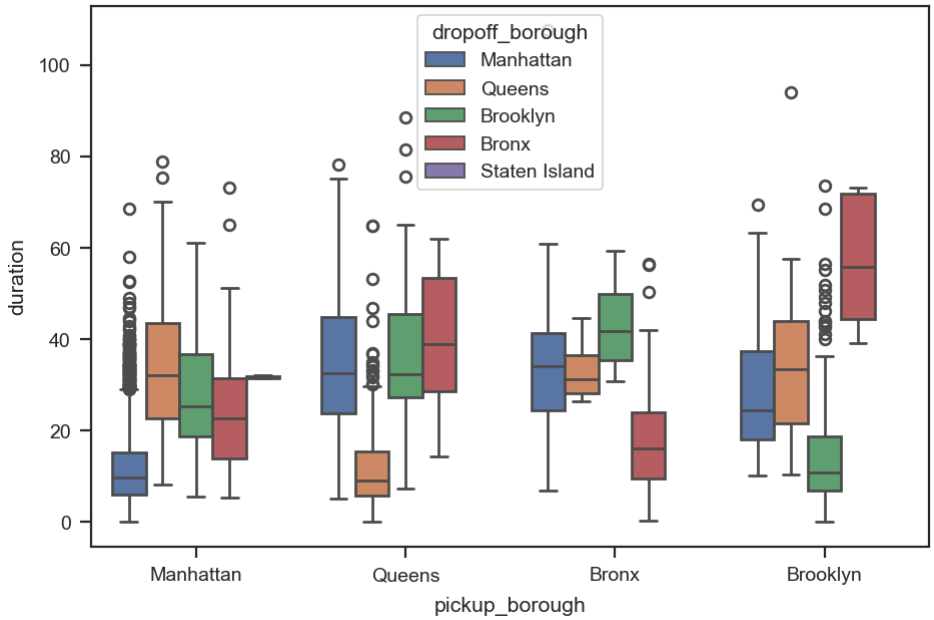
Exercise 8

Produce this plot (see the boxplot documentation):



In [ ]:
```
sns.boxplot(data=rides, x='pickup_borough', y='duration', hue='dropoff_bo
```

Out[ ]:
```
<Axes: xlabel='pickup_borough', ylabel='duration'>
```

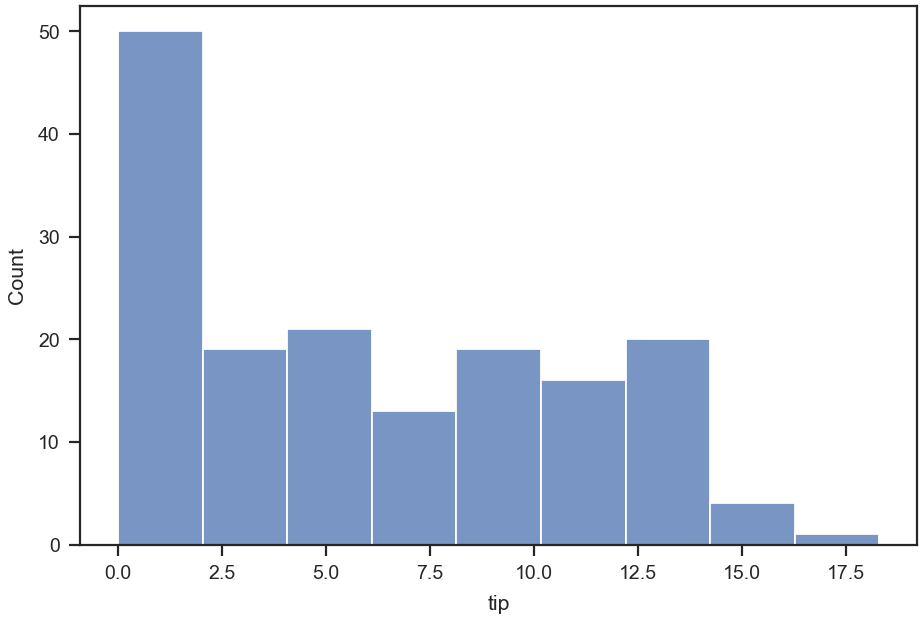


Exercise 9

Make a histogram of the tip for fares starting on Manhattan and ending in Queens:

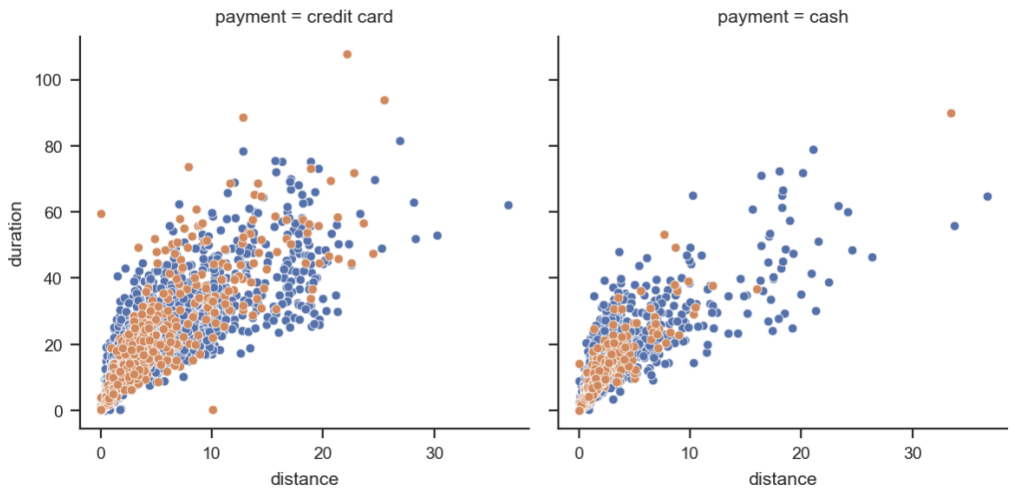
In [ ]:
```
df2 = rides.loc[(rides.dropoff_borough == 'Queens') & (rides.pickup_borou
sns.histplot(data=df2, x='tip')
```

Out[ ]:
```
<Axes: xlabel='tip', ylabel='Count'>
```



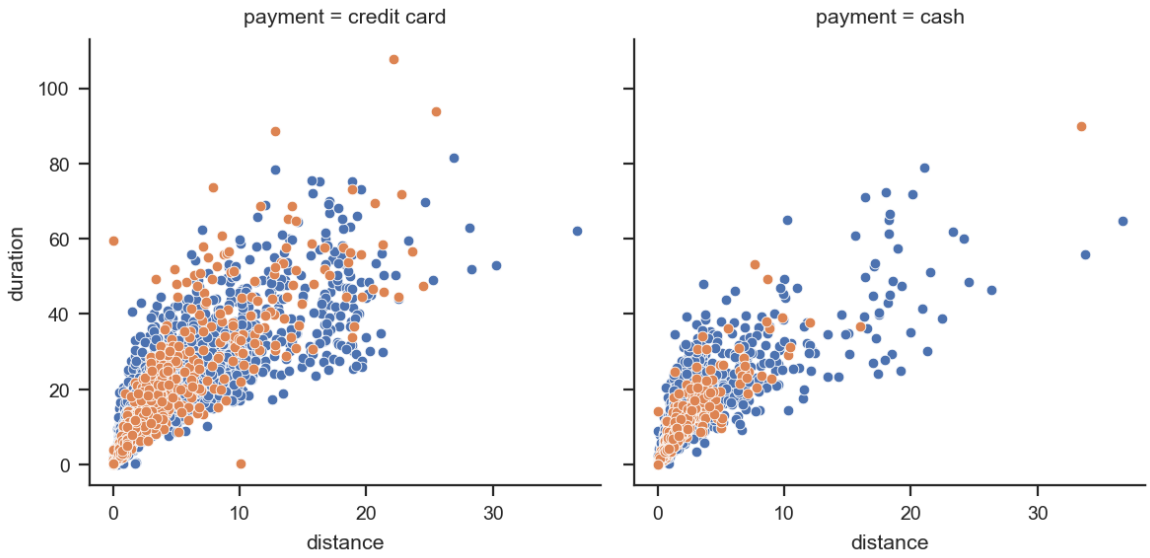
Exercise 10

Produce this plot, where datapoints are colored by the color of the taxi:



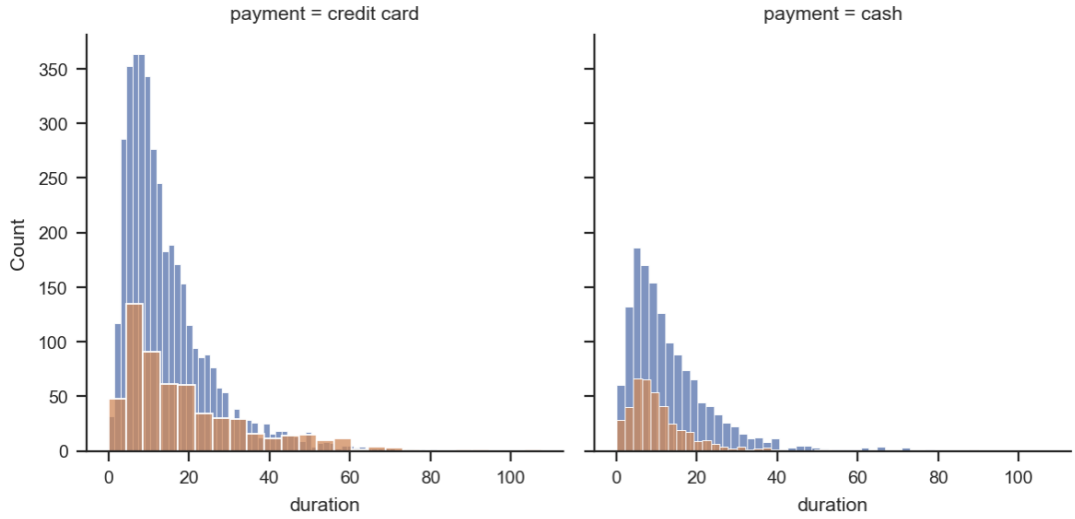
In [ ]:
```
g = sns.FacetGrid(data=rides, col="payment", hue='color')
g.map(sns.scatterplot, 'distance', 'duration')
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x14323ac40>
```



Exercise 11

Produce this plot where the color of the datapoints is also determined by the color of the taxi:



In [ ]:
```
g1 = sns.FacetGrid(rides, col='payment', hue='color')
g1.map(sns.histplot,'duration')
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x143948640>
```