Snapchat Political Ads

This project uses political ads data from Snapchat, a popular social media app. Interesting questions to consider include:

- What are the most prevalent organizations, advertisers, and ballot candidates in the data? Do you recognize any?
- What are the characteristics of ads with a large reach, i.e., many views? What may a campaign consider when maximizing an ad's reach?
- What are the characteristics of ads with a smaller reach, i.e., less views? Aside from funding constraints, why might a campaign want to produce an ad with a smaller but more targeted reach?
- What are the characteristics of the most expensive ads? If a campaign is limited on advertising funds, what type of ad may the campaign consider?
- What groups or regions are targeted frequently? (For example, for single-gender campaigns, are men or women targeted more frequently?) What groups or regions are targeted less frequently? Why? Does this depend on the type of campaign?
- Have the characteristics of ads changed over time (e.g. over the past year)?
- When is the most common local time of day for an ad's start date? What about the most common day of week? (Make sure to account for time zones for both questions.)

Getting the Data

The data and its corresponding data dictionary is downloadable here (https://www.snap.com/en-US/political-ads/). Download both the 2018 CSV and the 2019 CSV.

The CSVs have the same filename; rename the CSVs as needed.

Note that the CSVs have the exact same columns and the exact same data dictionaries (`readme.txt`).

Cleaning and EDA

- Concatenate the 2018 CSV and the 2019 CSV into one DataFrame so that we have data from both years.
- Clean the data.
  - Convert `StartDate` and `EndDate` into datetime. Make sure the datetimes are in the correct time zone.
- Understand the data in ways relevant to your question using univariate and bivariate analysis of the data as well as aggregations.

Hint 1: What is the "Z" at the end of each timestamp?

Hint 2: `pd.to_datetime` will be useful here. `Series.dt.tz_convert` will be useful if a change in time zone is needed.

Tip: To visualize geospatial data, consider Folium (https://python-visualization.github.io/folium/) or another geospatial plotting library.

Assessment of Missingness

Many columns which have `NaN` values may not actually have missing data. How come? In some cases, a null or empty value corresponds to an actual, meaningful value. For example, `readme.txt` states the following about `Gender` :

> Gender - Gender targeting criteria used in the Ad. If empty, then it is targeting all genders

In this scenario, an empty `Gender` value (which is read in as `NaN` in pandas) corresponds to "all genders".

- Refer to the data dictionary to determine which columns do **not** belong to the scenario above. Assess the missingness of one of these columns.

Hypothesis Test / Permutation Test

Find a hypothesis test or permutation test to perform. You can use the questions at the top of the notebook for inspiration.

Summary of Findings

Introduction

The question we pick:

> 'What are the characteristics of ads with a large reach, i.e., many views? What may a campaign consider when maximizing an ad's reach?'

The 2018 and 2019 datasets contained in https://www.snap.com/en-US/political-ads/ (https://www.snap.com/en-US/political-ads/) are two datasets about political advertisemet in 2018 and 2019, and they give details about all political and advocacy advertising running on the Snapchat platform.

These datasets are related to the question we are investigating, because these datasets contains the number of views('Impression') for each advertisement, and many other potential characteristics that could be related to the number of views.

Cleaning and EDA

- **Clean the data**
  - Concatenate two dataframes 2018 and 2019 into one called Political_Ads.
  - Convert StartDate and EndDate into datetime.
  - Add column 'Year' to Political_Ads.
  - Add column TimeDuration in unit of hours.
- **Univariate Analysis**
  Look at the statistics of Spend, StartDate, Startdate Hour separately to understand the dataset.
- **Bivariate Analysis & Interesting Aggregates**
  We Choose columns to group-by and examine aggregate statistics. Then look at the statistics of pairs of columns to identify possible associations:
  - Spend vs Impressions
  - TimeDuration vs Impressions
  - CandidateBallotInformation vs Impressions
  - Gender vs Impression

Assessment of Missingness

- **Differentiate trivial missingness and non-trivial missingness**

Trivial
Gender
AgeBracket
RegionID
ElectoralDistrictID
LatLongRad MetroID OsType
Targeting Geo-Postal Code

Non-Trivial

Non-Trivial
Interests
Language
AdvancedDemographics
Target Connection Type
Targeting Carrier (ISP)
EndDate
CandidateBallotInformation
Segments
LocationType
CreativeProperties

- **Assess whether data is NMAR**

Column	Is NMAR	Reason	Additional data
Interests	No	The audiences' interest is random, we don't think itself will affect the missingness	~
Language	No	The audiences' language is random, we don't think itself will affect the missingness	~
AdvancedDemographics	Yes	We think some specific 3rd party don't want their information to be collected	The name of 3rd party
Target Connection Type	Yes	Some specific internet connection type may be encrypted(e.g used for military), which make it hard to be known	User type
Targeting Carrier (ISP)	Yes	The carrier used by audiences is privacy	Is there a protection of user privacy in the carrier's contract
EndDate	No	The end date is random, we don't think itself will affect the missingness	~
CandidateBallotInformation	No	The name is random, we don't think itself will affect the missingness	~
Segments	Yes	Some specific segment is encrypted(e.g used for military), which make it hard to be known	User type
LocationType	No	The audiences' location type is random, we don't think itself will affect the missingness	~
CreativeProperties	No	The URL attachments is random, we don't think itself will affect the missingness	~

- **Analyses of one non-trivial column**
  We do the analyses on the column **Interests**, want to see whether this column depends on column **Spend** and **Impressions**.
- **Result**
  We get the P-value:
  - P-value(Interests ~ Impressions): 0.003
  - P-value(Interests ~ Spend): 0.107

  We set the **significant level** as 0.1. The p-value for the Interests' missingness and Impression is lower than significant level, which means that the missingness of Interests depends on the column

Impression.

And the p-value for the Interests' missingness and Spend is higher than significant value, which means that the missingness of Interests doesn't depends on the column Spend.

The missingness of Interests depends on the Impression, so maybe the feature Interests have some effect on the number of views.

Hypothesis Test

Formulate a hypothesis and perform a hypothesis test.

- **First characteristic: 'Spend'**
  - **Null hypothesis:** 'Spend' of ads with large number of impressions has the same distribution as that with small number of impressions.
  - **Alternative hypothesis:** 'Spend' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions.
  - **The test_statistic:** KS statistic
  - **The significant level:** 0.1
  - **The result p_value:** 0.0
  - **Result:** 'Spend' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions. So the 'Spend' is a significant feature affecting whether ads have a large reach.

- **Second characteristic: 'TimeDuration'**
  - **Null hypothesis:** 'TimeDuration' of ads with large number of impressions has the same distribution as that with small number of impressions.
  - **Alternative hypothesis:** 'TimeDuration' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions.
  - **The test_statistic:** KS statistic
  - **The significant level:** 0.1
  - **The result p_value:** 0.085
  - **Result:** 'TimeDuration' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions. So the 'TimeDuration' is also a significant feature affecting whether ads have a large reach.

Answer to the question

The characteristic of **Spend** and **TimeDuration** both have a significant effect on whether ads have a large reach.

A campaign should consider both the **spend** and the **time duration** of ads when they want to maximize an ad's reach.

Code

In [2]:

```
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import seaborn as sns
from scipy import stats
%matplotlib inline
%config InlineBackend.figure_format = 'retina'  # Higher resolution figures
```

Cleaning and EDA

1. Concatenate the 2018 CSV and the 2019 CSV into one DataFrame so that we have data from both years.

In [3]:

```
fp_2019 = os.path.join('data', 'PoliticalAds2019.csv')
political_ads_2019 = pd.read_csv(fp_2019, encoding = "utf-8")
political_ads_2019['Year'] = 2019
fp_2018 = os.path.join('data', 'PoliticalAds2018.csv')
political_ads_2018 = pd.read_csv(fp_2018, encoding = "utf-8")
political_ads_2018['Year'] = 2018
political_ads = pd.concat([political_ads_2018, political_ads_2019], ignore_index
=True)
political_ads.head()
```

Out[3]:

	ADID	CreativeUrl	Spend
0	2ac103bc69cce2d24b198e6a6d052dbff2c25ae9b6bb9e...	https://www.snap.com/political-ads/asset/69afd...	165
1	40ee7e900be9357ae88181f5c8a56baf6d5aab0e8d0f51...	https://www.snap.com/political-ads/asset/0885d...	17
2	c80ca50681d552551ceaf625981c0202589ca710d51925...	https://www.snap.com/political-ads/asset/a36b7...	60
3	a3106af2289b62f57f63f4fb89753bdf94e2fadede0478...	https://www.snap.com/political-ads/asset/46819...	2492
4	7afda4224482eb70315797966b4dcdeb856df916df5bdc...	https://www.snap.com/political-ads/asset/ee833...	5795

5 rows × 28 columns

2. Clean the data.

Convert StartDate and EndDate into datetime. Make sure the datetimes are in the correct time zone.

In [4]:

```
political_ads.StartDate = pd.to_datetime(political_ads.StartDate)
political_ads.EndDate = pd.to_datetime(political_ads.EndDate)
political_ads['TimeDuration'] = (political_ads.EndDate -  political_ads.StartDat
e).astype('timedelta64[h]')
political_ads['Impressions_per_H'] = political_ads['TimeDuration']/political_ads
['Impressions']
political_ads.head()
```

Out[4]:

	ADID	CreativeUrl	Spend	I
0	2ac103bc69cce2d24b198e6a6d052dbff2c25ae9b6bb9e...	https://www.snap.com/political-ads/asset/69afd...	165	
1	40ee7e900be9357ae88181f5c8a56baf6d5aab0e8d0f51...	https://www.snap.com/political-ads/asset/0885d...	17	
2	c80ca50681d552551ceaf625981c0202589ca710d51925...	https://www.snap.com/political-ads/asset/a36b7...	60	
3	a3106af2289b62f57f63f4fb89753bdf94e2fadede0478...	https://www.snap.com/political-ads/asset/46819...	2492	
4	7afda4224482eb70315797966b4dcdeb856df916df5bdc...	https://www.snap.com/political-ads/asset/ee833...	5795	

5 rows × 30 columns

3. Understand the data in ways relevant to your question using univariate and bivariate analysis of the data as well as aggregations.

a. Univariate

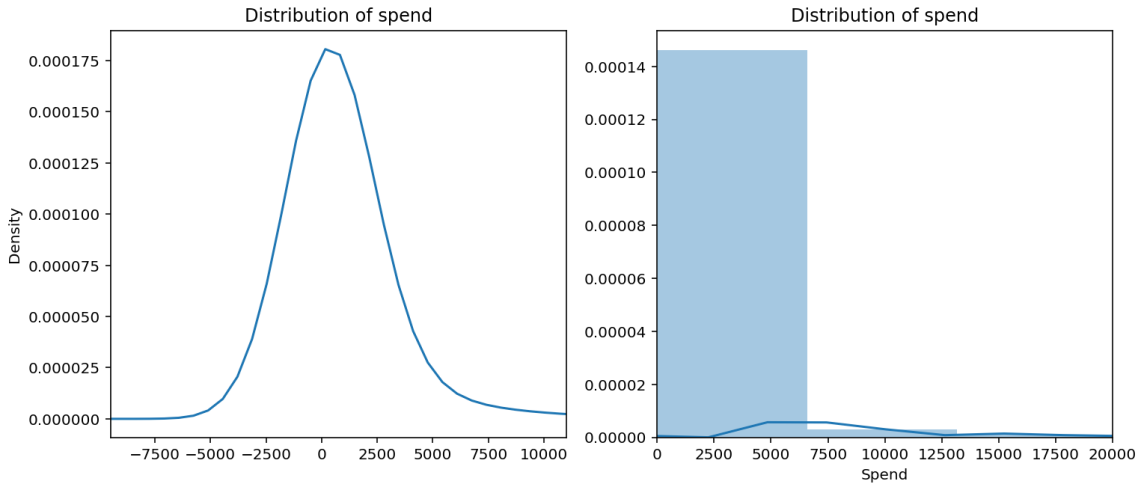
1. Visualization of Spend

Explanation of this visualization:

We can find that the spend is Gaussian distribution and the number of spend focuses on around 6000

In [197]:

```
fig = plt.subplots(1, 2, figsize=(12, 5))

# The plot of distribution of variate spend
plt.subplot(121)
political_ads['Spend'].plot(kind='kde')
plt.xlim(xmin=-9500, xmax = 11000)
plt.title('Distribution of spend')


# The plot of spend number
plt.subplot(122)
sns.distplot(political_ads['Spend'])
plt.xlim(xmin = 0, xmax = 20000)
plt.title('Distribution of spend')

plt.show()
```



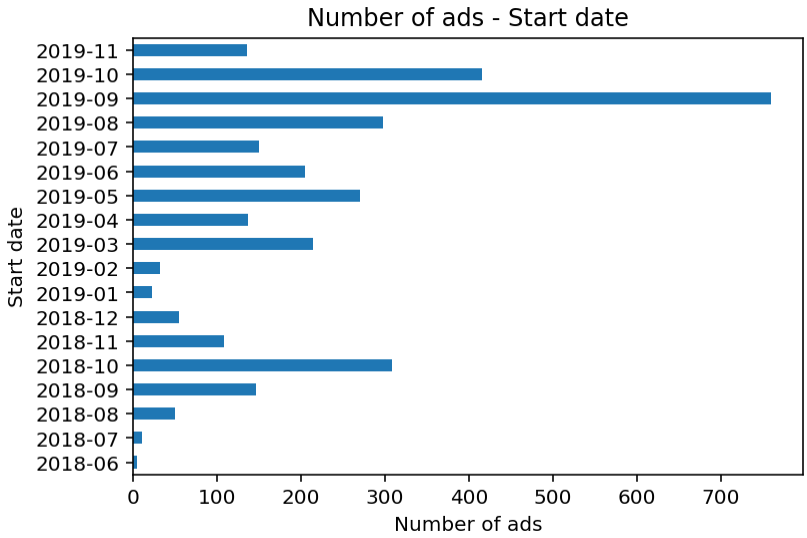
2. Visualization of start date

Explanation of this visualization:

We can find that when the start date is around every year's Sep. and Oct., the number of ads will have a peak.

In [199]:

```
# The plot of distribution of variate start date
s = pd.to_datetime(political_ads['StartDate']).apply(lambda x: str(x.year) + '-'
+ str(x.month) if len(str(x.month)) == 2 else str(x.year) + '-0' + str(x.month))
s.value_counts().sort_index().plot(kind='barh')
plt.title('Number of ads - Start date')
plt.xlabel('Number of ads')
plt.ylabel('Start date')
plt.show()
```



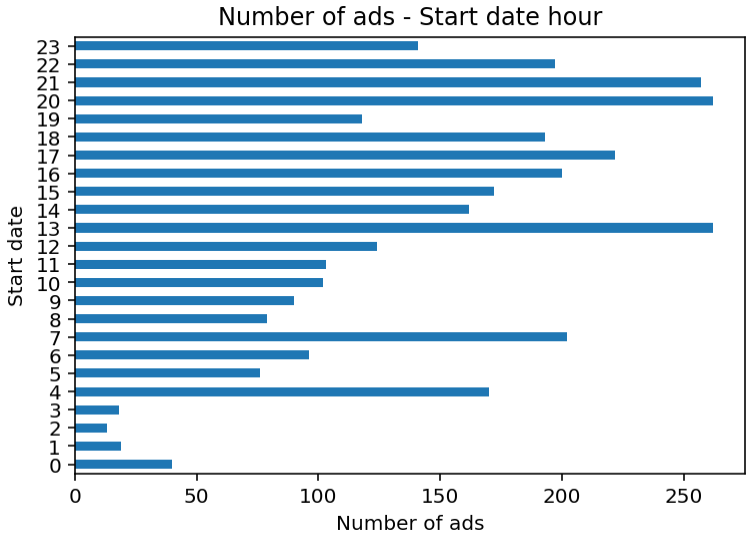
3. Visualization of start date hour

Explanation of this visualization:

We can find that when the hour of the start date is around the day's 7:00am, 13:00pm, 17:00pm and 20:00pm, the number of ads will have a peak.

In [200]:

```
# The plot of distribution of variate start date
s = pd.to_datetime(political_ads['StartDate']).apply(lambda x: x.hour)
s.value_counts().sort_index().plot(kind='barh')
plt.title('Number of ads - Start date hour')
plt.xlabel('Number of ads')
plt.ylabel('Start date')
plt.show()
```



b. Bivariate & Interesting Aggregates

1. Visualization of spend and impressions

Explanation of this visualization:

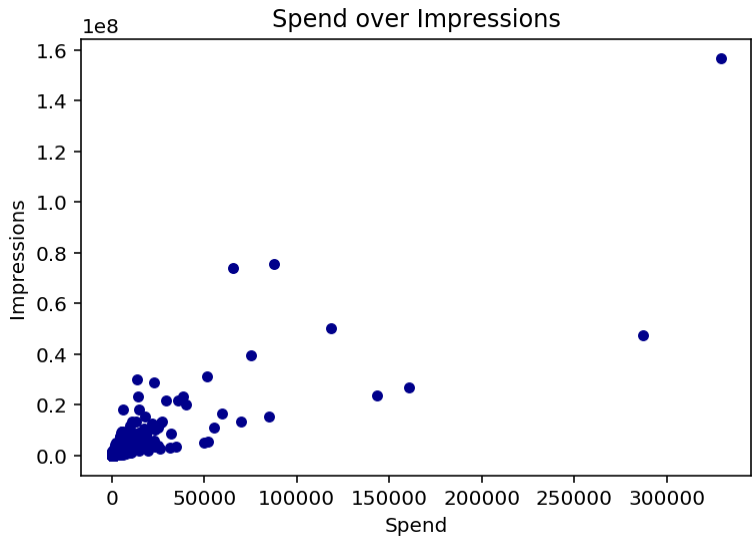
We can find that spend is correlated with impression since their correlation is 08 and the graph shows a generally positive linear graph.

In [196]:

```
# Spend vs Impressions, good characteristic
political_ads.plot.scatter(x='Spend',
                           y='Impressions',
                           c='DarkBlue',
                           title='Spend over Impressions'
                          )
correlation = political_ads['Impressions'].corr(political_ads['Spend'])
correlation
```

Out[196]:

```
0.8393838492481681
```



2. Visualization of TImeduration and impressions

Explanation of this visualization:

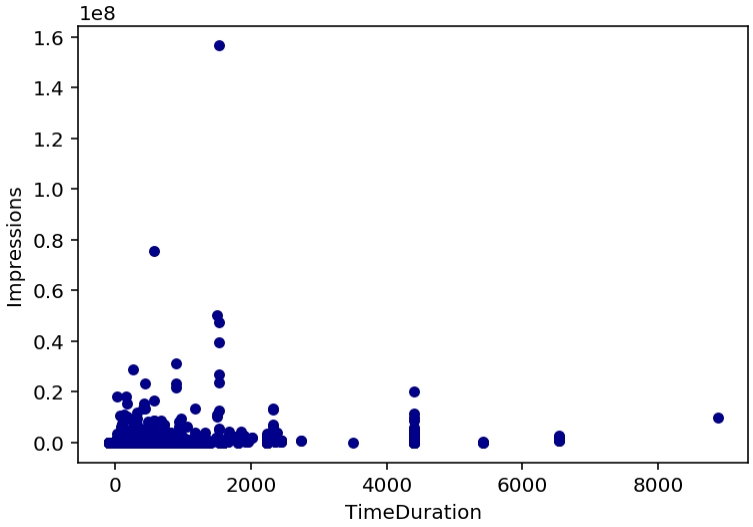
We can find that when the there is no clear relationship between timeduration and impressions

In [142]:

```
# TimeDuration vs Impressions, not a characteristic
political_ads.plot.scatter(x='TimeDuration',
                           y='Impressions',
                           c='DarkBlue')
```

Out[142]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c05d408c88>
```



3. Visualization of language and impressions

Explanation of this visualization:

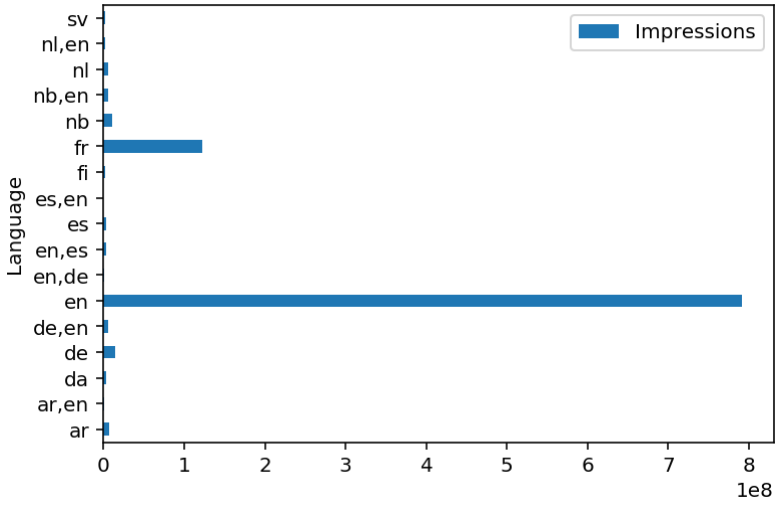
We can find that when the there is no clear relationship between timeduration and impressions

In [42]:

```
# Language vs Impressions, languages that are english or french tend to have mor
e views
political_ads.pivot_table(
    index='Language',
    values='Impressions',
    aggfunc='sum'
).plot(kind='barh')
```

Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c05aeb84e0>
```



4. Visualization of CandidateBallotInformation and impressions

Explanation of this visualization:

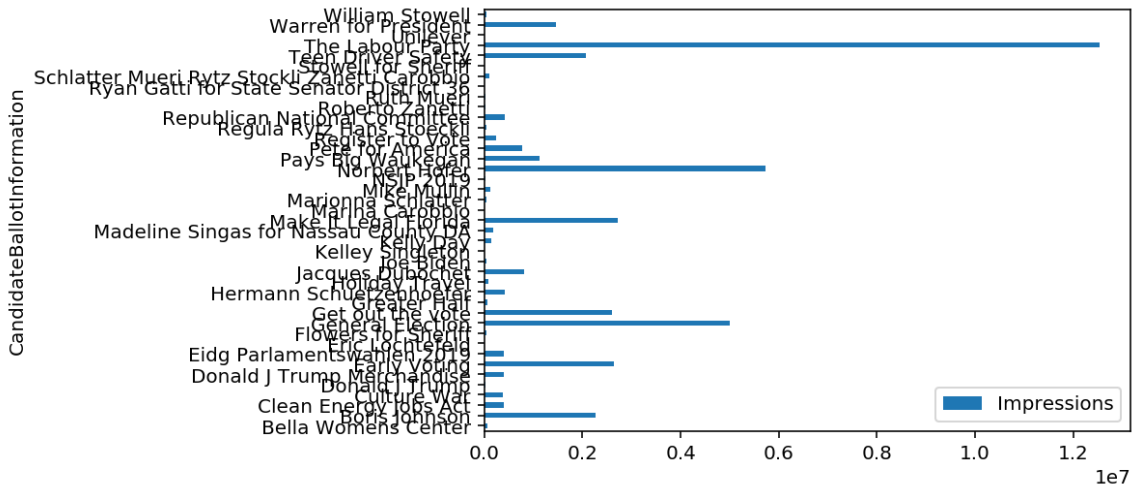
We can find that ads with certain candidate ballot information have more views.

In [56]:

```
# CandidateBallotInformation vs Impressions, as seen in the gragh, there are cou
ple companies tend to have large views
df = political_ads.pivot_table(
    index='CandidateBallotInformation',
    values='Impressions',
    aggfunc='sum'
)
df.plot(kind='barh')
df.sort_values('Impressions', ascending=False).head(10)
```

Out[56]:

CandidateBallotInformation	Impressions
The Labour Party	12549837
Norbert Hofer	5729008
General Election	5003669
Make It Legal Florida	2729241
Early Voting	2637803
Get out the vote	2602403
Boris Johnson	2260530
Teen Driver Safety	2067227
Warren for President	1456274
Pays Big Waukegan	1134746



5. Visualization of CandidateBallotInformation and impressions

Explanation of this visualization:

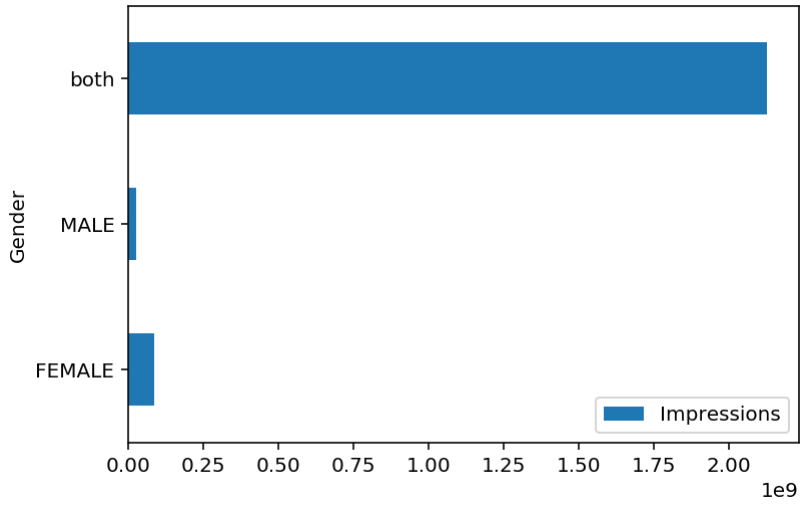
We can find that ads targetting all genders have more views, and female comes second, male comes last. However, gender might not be a good characteristic for the second question because the majority of the gender in this dataset is both

In [49]:

```
# Gender vs Impressions, as seen in the gragh,
#ads that are target to both genders tend to have large views, and female comes
 second, male comes last
political_ads.Gender = political_ads.Gender.fillna('both')
political_ads.pivot_table(
    index='Gender',
    values='Impressions',
    aggfunc='sum'
).plot(kind='barh')
```

Out[49]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c05bb93cf8>
```



Assessment of Missingness

1. Differentiate trivial missingness and non-trivial missingness

Columns with missingness:

In [203]:

```
pd.Series(political_ads.columns[political_ads.isnull().mean() != 0], name='columnName').to_frame()
```

Out[203]:

	columnName
0	EndDate
1	CandidateBallotInformation
2	AgeBracket
3	RegionID
4	ElectoralDistrictID
5	LatLongRad
6	MetroID
7	Interests
8	OsType
9	Segments
10	LocationType
11	Language
12	AdvancedDemographics
13	Targeting Connection Type
14	Targeting Carrier (ISP)
15	Targeting Geo - Postal Code
16	CreativeProperties
17	TimeDuration
18	Impressions_per_H

Trivial columns:

- Gender
- AgeBracket
- RegionID
- ElectoralDistrictID
- LatLongRad MetroID OsType
- Targeting Geo-Postal Code

Non-trivial columns:

- Interests
- Language
- AdvancedDemographics
- Target Connection Type
- Targeting Carrier (ISP)
- EndDate
- CandidateBallotInformation
- Segments
- LocationType
- CreativeProperties

2. Assess whether data is NMAR

Column	Is NMAR	Reason	Additional data
Interests	No	The audiences' interest is random, we don't think itself will affect the missingness	~
Language	No	The audiences' language is random, we don't think itself will affect the missingness	~
AdvancedDemographics	Yes	We think some specific 3rd party don't want their information to be collected	The name of 3rd party
Target Connection Type	Yes	Some specific internet connection type may be encrypted(e.g used for military), which make it hard to be known	User type
Targeting Carrier (ISP)	Yes	The carrier used by audiences is privacy	Is there a protection of user privacy in the carrier's contract
EndDate	No	The end date is random, we don't think itself will affect the missingness	~
CandidateBallotInformation	No	The name is random, we don't think itself will affect the missingness	~
Segments	Yes	Some specific segment is encrypted(e.g used for military), which make it hard to be known	User type
LocationType	No	The audiences' location type is random, we don't think itself will affect the missingness	~
CreativeProperties	No	The URL attachments is random, we don't think itself will affect the missingness	~

3. Analyses of one non-trivial column

We choose the column **Interests** to analyze, and set the **significant level** as **0.1**:

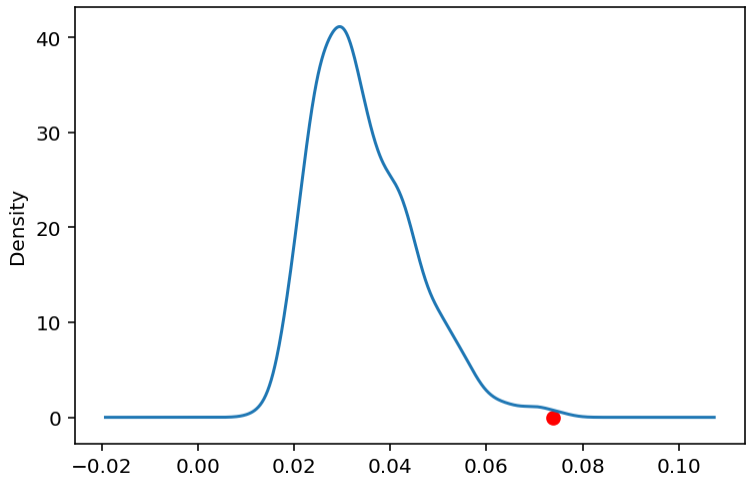
In [206]:

```
def permutation(col, dep_test, N=100):
    # Select the needed columns
    part = political_ads[[col, dep_test]]
    part = part.assign(
        is_missing=part[col].isnull()
    )

    # Calculate the observation value
    grps = part.groupby('is_missing')[dep_test]
    obv = stats.ks_2samp(grps.get_group(True), grps.get_group(False)).statistic

    ks_ls = []
    for _ in range(N):
        # Sample the tested column
        shuffled_col = (
            part[dep_test]
            .sample(frac=1, replace = False)
            .reset_index(drop=True)
        )
        shuffled_df = part.assign(
            shuffled=shuffled_col
        )

        # Calculate sampled column value
        grps = shuffled_df.groupby('is_missing')['shuffled']
        ks = stats.ks_2samp(grps.get_group(True), grps.get_group(False)).statist
ic
        ks_ls.append(ks)

    # Plot the distribution and observation value
    pd.Series(ks_ls).plot(kind='kde')
    plt.scatter(obv, 0, s=40, c='r')

    # Calculate the P-value
    p_value = np.count_nonzero(np.array(ks_ls) > obv) / N
    return p_value
```

In [207]:

```
# P-value(Interests ~ Impressions)
print('P-value(Interests ~ Impressions): ', permutation('Interests', 'Impressions', 1000))
```

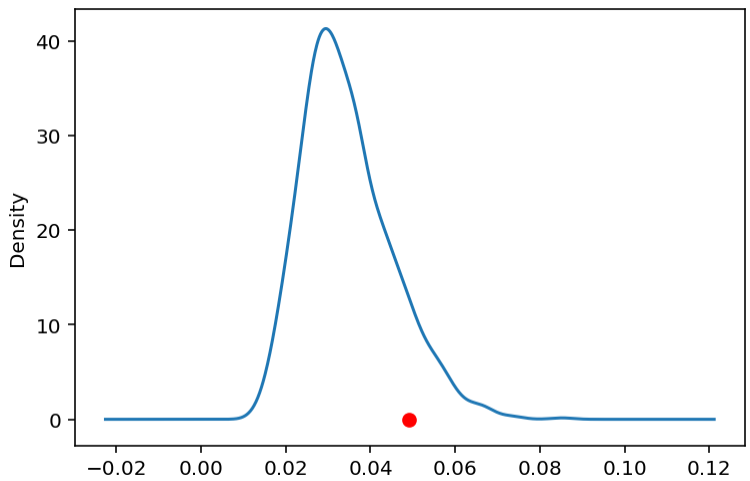
```
P-value(Interests ~ Impressions):  0.002
```



In [208]:

```
# P-value(Interests ~ Spend)
print('P-value(Interests ~ Spend): ', permutation('Interests', 'Spend', 1000))
```

```
P-value(Interests ~ Spend):  0.098
```



According to the above analyses:

- P-value(Interests ~ Impressions): 0.003
- P-value(Interests ~ Spend): 0.107

So, the missingness of column Interests **depends** on the column Impression, and does **not depends** on the column Spend.

4. Interpretation of the result

The p-value for the Interests' missingness and Impression is lower than significant value, which means that the missingness of Interests depends on the column Impression.

And the p-value for the Interests' missingness and Spend is higher than significant value, which means that the missingness of Interests doesn't depends on the column Spend.

The missingness of Interests depends on the Impression, so maybe the feature Interests have some effect on the number of views. To answer our question, we will confirm this in the next section.

Hypothesis Test

To simplify the further analysis, we set a scaling limit at (mean + str) for the impressions.

Any impressions >= this limit, we set it to 1, 0 otherwise.

This way, we could easily define whether a certain ads has large reach or not.

In [218]:

```
limit = political_ads['Impressions'].mean() + political_ads['Impressions'].std()
political_ads['ImpressionsB'] = political_ads['Impressions'].apply(lambda x: 1 if x >= limit else 0)
```

We created a function called permutation_ that can do permutation, plot the permutation graph and p-value.

It requires two parameters: the characteristic we want to investigate, the number of trials we want to shuffle the values.

In [219]:

```
def permutation_(dep, N_trials=1000):
    differences = []

    # Calculate observation values
    grps = political_ads[[dep, 'ImpressionsB']].groupby('ImpressionsB')[dep]
    obv = stats.ks_2samp(grps.get_group(1), grps.get_group(0)).statistic
    for i in np.arange(N_trials):
        # Do sampling
        shuffled = (
            political_ads[dep]
            .sample(replace=False, frac=1)
            .reset_index(drop=True)
        )

        shuffled = (
            political_ads
            .assign(**{'Shuffled': shuffled})
        )

        # Calculate KS statistic
        grps = shuffled[['Shuffled', 'ImpressionsB']].groupby('ImpressionsB')['S
huffled']
        difference = stats.ks_2samp(grps.get_group(1), grps.get_group(0)).statis
tic

        differences.append(difference)
    pd.Series(differences).plot(kind='hist', density=True, alpha=0.8)
    plt.scatter(obv, 0, color='red', s=40)
    p_val = np.count_nonzero(differences >= obv) / N_trials
    return p_val
```

1. First characteristic: 'TimeDuration'

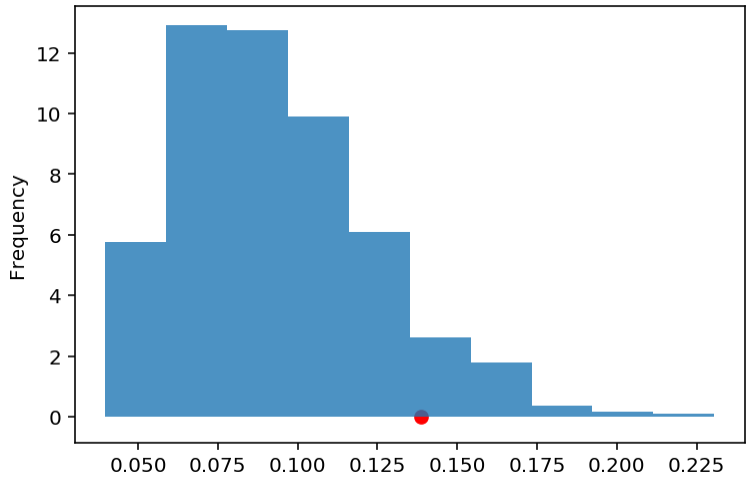
- **Null hypothesis:** 'TimeDuration' of ads with large number of impressions has the same distribution as that with small number of impressions.
- **Alternative hypothesis:** 'TimeDuration' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions.
- **The test_statistic:** KS statistic
- **The significant level:** 0.1
- **The result p_value:** 0.085
- **Result:** 'TimeDuration' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions. So the 'TimeDuration' is also a significant feature affecting whether ads have a large reach.

In [214]:

```
permutation_('TimeDuration', N_trials=1000)
```

Out[214]:

0.085



Reason for picking 'TimeDuration':

'TimeDuration' is a good choice for answering the second question. Ads with longer TimeDuration have more views in general. Although in the previous section, Bivariate, the graph between TimeDuration and impressions is not clear, through permutations, we found that impressions do relate to timedurations, and its p_val is below our significance level.

2. Second characteristic: 'Spend'

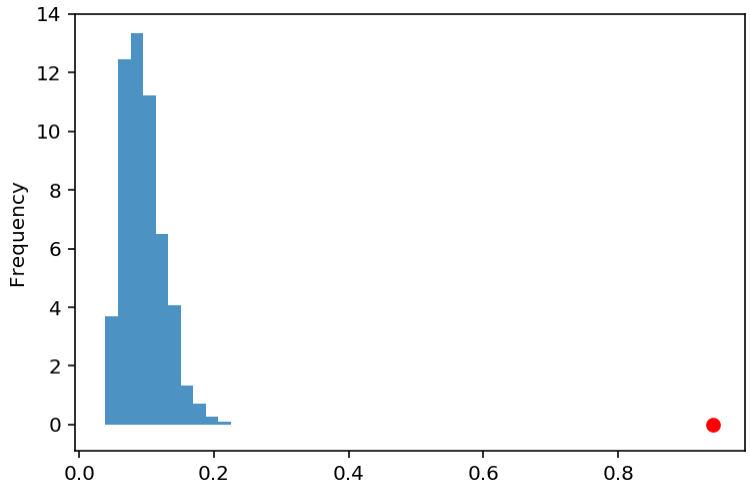
- **Null hypothesis:** 'Spend' of ads with large number of impressions has the same distribution as that with small number of impressions.
- **Alternative hypothesis:** 'Spend' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions.
- **The test_statistic:** KS statistic
- **The significant level:** 0.1
- **The result p_value:** 0.0
- **Result:** 'TimeDuration' of ads with large number of impressions doesn't have the same distribution as that with small number of impressions. So the 'TimeDuration' is also a significant feature affecting whether ads have a large reach.

In [215]:

```
permutation_('Spend', N_trials=1000)
```

Out[215]:

0.0



Reason for picking 'Spend':

'Spend' is a good choice for answering the second question. Ads with more spend have more view in general. As shown in the previous section, Bivariate, the graph between spend and impressions are positive and their correlation is 0.8. Thus, it is a good choice for answering the second question.