In [196]:
```python
import warnings
warnings.filterwarnings('ignore')
import datetime
import string
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

In [197]:
```python
rawdata= pd.read_csv(r'C:\Users\krishna\Desktop\sem 6\Interview.csv', delimite
r=',')
```

In [198]:
```python
rawdata
```

Out[198]:

	Date of Interview	Client name	Industry	Location	Position to be closed	Nature of Skillset	Interview Type	Name(Ca I
0	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Candidate
1	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Candidate
2	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Candidate
3	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Candidate
4	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Candidate
...	...	...	...	...	...	...	...	
1229	07.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	Biosimiliars	Scheduled	Candida 12
1230	06.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	Biosimiliars	Scheduled	Candida 12
1231	06.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	generic drugs – RA	Scheduled	Candida 12
1232	06.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	generic drugs – RA	Scheduled	Candida 12
1233	NaN		NaN	NaN	NaN	NaN	NaN	Na

1234 rows × 28 columns

```
In [199]: print(rawdata.isnull().sum())
```

```
Date of Interview
1
Client name
0
Industry
1
Location
1
Position to be closed
1
Nature of Skillset
1
Interview Type
1
Name(Cand ID)
1
Gender
1
Candidate Current Location
1
Candidate Job Location
1
Interview Venue
1
Candidate Native location
1
Have you obtained the necessary permission to start at the required time
205
Hope there will be no unscheduled meetings
248
Can I Call you three hours before the interview and follow up on your attenda
nce for the interview     248
Can I have an alternative number/ desk number. I assure you that I will not t
rouble you too much       248
Have you taken a printout of your updated resume. Have you read the JD and un
derstood the same         249
Are you clear with the venue details and the landmark.
249
Has the call letter been shared
246
Expected Attendance
6
Observed Attendance
1
Marital Status
1
Unnamed: 23
1234
Unnamed: 24
1234
Unnamed: 25
1234
Unnamed: 26
1234
Unnamed: 27
1234
dtype: int64
```

In [200]:
```
#droping the last row as its all nan.
rawdata = rawdata.drop(rawdata.tail(1).index)
# drop irrelevant columns
rawdata.drop(['Unnamed: 23', 'Unnamed: 24', 'Unnamed: 25', 'Unnamed: 26', 'Unn
amed: 27','Name(Cand ID)'], axis=1, inplace=True)
```

In [201]:
```
rawdata
```

Out[201]:

	Date of Interview	Client name	Industry	Location	Position to be closed	Nature of Skillset	Interview Type	Gender
0	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Male
1	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Male
2	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Male
3	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Male
4	13.02.2015	Hospira	Pharmaceuticals	Chennai	Production-Sterile	Routine	Scheduled Walkin	Male
...	...	...	...	...	...	...	...	...
1228	07.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	generic drugs – RA	Scheduled	Male
1229	07.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	Biosimiliars	Scheduled	Male
1230	06.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	Biosimiliars	Scheduled	Male
1231	06.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	generic drugs – RA	Scheduled	Male
1232	06.05.2016	Pfizer	Pharmaceuticals	Chennai	Niche	generic drugs – RA	Scheduled	Female

1233 rows × 22 columns

In [202]:
```
rawdata['Date of Interview'].unique
```

Out[202]:
```
<bound method Series.unique of 0       13.02.2015
1       13.02.2015
2       13.02.2015
3       13.02.2015
4       13.02.2015
           ...
1228    07.05.2016
1229    07.05.2016
1230    06.05.2016
1231    06.05.2016
1232    06.05.2016
Name: Date of Interview, Length: 1233, dtype: object>
```

```
In [203]: rawdata.describe
```

```
Out[203]: <bound method NDFrame.describe of      Date of Interview Client name
          Industry Location  \
          0          13.02.2015     Hospira  Pharmaceuticals  Chennai
          1          13.02.2015     Hospira  Pharmaceuticals  Chennai
          2          13.02.2015     Hospira  Pharmaceuticals  Chennai
          3          13.02.2015     Hospira  Pharmaceuticals  Chennai
          4          13.02.2015     Hospira  Pharmaceuticals  Chennai
          ...               ...         ...              ...      ...
          1228       07.05.2016      Pfizer  Pharmaceuticals  Chennai
          1229       07.05.2016      Pfizer  Pharmaceuticals  Chennai
          1230       06.05.2016      Pfizer  Pharmaceuticals  Chennai
          1231       06.05.2016      Pfizer  Pharmaceuticals  Chennai
          1232       06.05.2016      Pfizer  Pharmaceuticals  Chennai

               Position to be closed  Nature of Skillset    Interview Type  Gender  \
          0      Production- Sterile             Routine  Scheduled Walkin    Male
          1      Production- Sterile             Routine  Scheduled Walkin    Male
          2      Production- Sterile             Routine  Scheduled Walkin    Male
          3      Production- Sterile             Routine  Scheduled Walkin    Male
          4      Production- Sterile             Routine  Scheduled Walkin    Male
          ...                    ...                 ...               ...     ...
          1228                 Niche  generic drugs – RA         Scheduled    Male
          1229                 Niche        Biosimiliars         Scheduled    Male
          1230                 Niche        Biosimiliars         Scheduled    Male
          1231                 Niche  generic drugs – RA         Scheduled    Male
          1232                 Niche  generic drugs – RA         Scheduled  Female

               Candidate Current Location Candidate Job Location  ...  \
          0                      Chennai                  Hosur  ...
          1                      Chennai              Bangalore  ...
          2                      Chennai                Chennai  ...
          3                      Chennai                Chennai  ...
          4                      Chennai              Bangalore  ...
          ...                        ...                    ...  ...
          1228                   Chennai                Chennai  ...
          1229                   Chennai                Chennai  ...
          1230                   Chennai                Chennai  ...
          1231                   Chennai                Chennai  ...
          1232                   Chennai                Chennai  ...

               Have you obtained the necessary permission to start at the required time
          \
          0                                                   Yes
          1                                                   Yes
          2                                                   NaN
          3                                                   Yes
          4                                                   Yes
          ...                                                 ...
          1228                                                Yes
          1229                                                Yes
          1230                                                Yes
          1231                                                Yes
          1232                                                NaN

               Hope there will be no unscheduled meetings  \
          0                                           Yes
          1                                           Yes
```

```
2                                            Na
3                                           Yes
4                                           Yes
...                                         ...
1228                                        Yes
1229                                        Yes
1230                                        Yes
1231                                        Yes
1232                                        NaN

     Can I Call you three hours before the interview and follow up on your at
tendance for the interview  \
0                                                   Yes
1                                                   Yes
2                                                   NaN
3                                                    No
4                                                   Yes
...                                                 ...
1228                                                Yes
1229                                                Yes
1230                                                Yes
1231                                                Yes
1232                                                NaN

     Can I have an alternative number/ desk number. I assure you that I will
not trouble you too much  \
0                                                   Yes
1                                                   Yes
2                                                   NaN
3                                                   Yes
4                                                    No
...                                                 ...
1228                                                Yes
1229                                                Yes
1230                                                Yes
1231                                                Yes
1232                                                NaN

     Have you taken a printout of your updated resume. Have you read the JD a
nd understood the same  \
0                                                   Yes
1                                                   Yes
2                                                   NaN
3                                                    No
4                                                   Yes
...                                                 ...
1228                                                Yes
1229                                                Yes
1230                                                Yes
1231                                                Yes
1232                                                NaN

     Are you clear with the venue details and the landmark.  \
0                                                   Yes
1                                                   Yes
2                                                   NaN
3                                                   Yes
```

```
4                                                    Yes
...                                                  ...
1228                                                 Yes
1229                                                 Yes
1230                                                 Yes
1231                                                 Yes
1232                                                 NaN

     Has the call letter been shared Expected Attendance Observed Attendance  \
0                                Yes                 Yes                  No
1                                Yes                 Yes                  No
2                                NaN           Uncertain                  No
3                                Yes           Uncertain                  No
4                                Yes           Uncertain                  No
...                              ...                 ...                 ...
1228                             Yes                 Yes                 Yes
1229                             Yes                 Yes                 Yes
1230                             Yes                 Yes                 Yes
1231                             Yes                 Yes                 Yes
1232                             NaN           Uncertain                 Yes

     Marital Status
0            Single
1            Single
2            Single
3            Single
4           Married
...             ...
1228        Married
1229         Single
1230        Married
1231         Single
1232         Single

[1233 rows x 22 columns]>
```

cleaning data

In [204]:
```
#begining with the Date

def get_cleaned_date(date):
    """
    Rteurn datetime object from a string
    """
    date = date.strip()

    if '&' in date:
        date = date.split('&')[0].strip()

    cleaned_date = None

    # Since there are a lot of formats in the data, need to handle all the pos
sible options
    date_formats = [
        '%d.%m.%Y', '%d.%m.%y', '%d.%m.%y', '%d-%m-%Y', '%d/%m/%y', '%d/%m/%Y'
, '%d %b %y', '%d-%b -%y',
        '%d – %b-%y', '%d -%b -%y'
    ]

    for date_format in date_formats:
        try:
            return datetime.datetime.strptime(date, date_format)
        except ValueError:
            pass
```

In [205]:
```
cleaned_interview_dates = rawdata['Date of Interview'].apply(get_cleaned_date)
```

In [206]:
```
# Check the min and max dates to see if the dates have been converted properly

print(cleaned_interview_dates.min())
print(cleaned_interview_dates.max())
```

```
2014-03-18 00:00:00
2023-04-12 00:00:00
```

In [207]:
```
#Looks like there's some incorrect data present. Inspect the dates sorted in r
everse order.

cleaned_interview_dates.sort_values(ascending=False)[:10]
```

Out[207]:
```
444   2023-04-12
443   2022-04-12
442   2021-04-12
441   2020-04-12
440   2019-04-12
439   2018-04-12
438   2017-04-12
228   2016-12-04
176   2016-12-04
170   2016-12-04
Name: Date of Interview, dtype: datetime64[ns]
```

In [208]:
```
#The date in indices 438-444 is incorrect, change the year to 2016 in each of
 them

rawdata.iloc[438:445, :]['Date of Interview'] = '12.04.2016'
```

In [209]:
```
# Since the data looks fine now, replace the column with this new Series
rawdata['Date of Interview'] = cleaned_interview_dates
```

In [210]:
```
#Cleaning the Client name column
rawdata['Client name'].value_counts()
```

Out[210]:
```
Standard Chartered Bank            904
Pfizer                              75
Hospira                             75
Aon Hewitt                          28
Flextronics                         23
ANZ                                 22
Hewitt                              20
UST                                 18
Prodapt                             17
Standard Chartered Bank Chennai     17
Astrazeneca                         15
Williams Lea                        11
Barclays                             5
Aon hewitt Gurgaon                   2
Woori Bank                           1
Name: Client name, dtype: int64
```

In [211]:
```
# Some clients are written in different names, so combine them
replace_dict = {
    'Standard Chartered Bank Chennai': 'Standard Chartered Bank',
    'Hewitt': 'Aon Hewitt',
    'Aon hewitt Gurgaon': 'Aon Hewitt'
}
rawdata['Client name'].replace(replace_dict, inplace=True)
```

In [212]:
```
#Club those client names which have count < 50 into a single category called
 "Others"

def merge_categories(column_name, threshold, merged_name='Others'):
    """
    Will merge those categories which have count below a certain threshold
    """
    column_counts = rawdata[column_name].value_counts()
    to_merge = column_counts[column_counts < threshold].index
    rawdata.loc[rawdata[column_name].isin(to_merge), column_name] = merged_nam
e
```

In [213]:
```
merge_categories('Client name', 50)
rawdata['Client name'].value_counts()
```

Out[213]:
```
Standard Chartered Bank    921
Others                     112
Pfizer                      75
Hospira                     75
Aon Hewitt                  50
Name: Client name, dtype: int64
```

In [214]:
```
#Cleaning the Industry column
rawdata['Industry'].value_counts()
```

Out[214]:
```
BFSI                         949
Pharmaceuticals              165
IT Products and Services      45
IT Services                   23
Electronics                   23
Telecom                       17
IT                            11
Name: Industry, dtype: int64
```

In [215]:
```
merge_categories('Industry', 50, 'IT')
rawdata['Industry'].value_counts()
```

Out[215]:
```
BFSI               949
Pharmaceuticals    165
IT                 119
Name: Industry, dtype: int64
```

In [216]:
```
#Cleaning the Position to be closed column
rawdata['Position to be closed'].value_counts()
```

Out[216]:
```
Routine                 1023
Niche                    163
Dot Net                   18
Trade Finance             11
AML                        8
Production- Sterile        5
Selenium testing           5
Name: Position to be closed, dtype: int64
```

In [217]:
```
replace_dict = {
    'Dot Net': 'Routine',
    'Trade Finance': 'Niche',
    'AML': 'Niche',
    'Selenium testing': 'Routine',
    'Production- Sterile': 'Routine'
}
rawdata['Position to be closed'].replace(replace_dict, inplace=True)
```

In [218]:
```
#Cleaning the Nature of Skillset column
rawdata['Nature of Skillset'].value_counts()
```

Out[218]:
```
JAVA/J2EE/Struts/Hibernate    220
Fresher                        86
Accounting Operations          86
AML/KYC/CDD                    84
CDD KYC                        52
                             ... 
SCCM – SQL                      1
Java, J2Ee                      1
sccm                            1
12.30 Pm                        1
SCCM – Sharepoint               1
Name: Nature of Skillset, Length: 92, dtype: int64
```

In [219]:
```
nature_of_skillset = rawdata['Nature of Skillset']

def clean_nature_of_skillset(x):
    x = x.lower()
    if 'java' in x:
        return 'java'
    elif 'oracle' in x:
        return 'oracle'
    elif 'testing' in x:
        return 'testing'
    elif 'aml' in x or 'kyc' in x or 'cdd' in x:
        return 'cdd'
    else:
        return x
cleaned_nature_of_skillset = nature_of_skillset.apply(clean_nature_of_skillset
)
cleaned_nature_of_skillset.value_counts()
```

```
Out[219]: java                                     459
          cdd                                      136
          accounting operations                     86
          fresher                                   86
          oracle                                    68
          routine                                   47
          testing                                   39
          sas                                       27
          lending and liabilities                   25
          banking operations                        24
          t-24 developer                            15
          sccm                                      15
          senior software engineer-mednet           15
          analytical r & d                          13
          cots developer                            13
          hadoop                                    12
          regulatory                                12
          publishing                                 9
          dot net                                    9
          ra publishing                              9
          etl                                        9
          tech lead-mednet                           8
          production                                 8
          biosimiliars                               6
          global labelling                           6
          emea                                       6
          senior analyst                             5
          product control                            5
          cots                                       4
          lending & liability                        4
          licensing – ra                             4
          generic drugs – ra                         4
          - sapbo, informatica                       4
          sccm- desktop support                      4
          11.30 am                                   3
          analytical r&d                             3
          biosimillar                                3
          tl                                         3
          ra label                                   2
          submission management                      2
          lcm -manager                               2
          production support - sccm                  2
          l & l                                      2
          lending&liablities                         2
          tech lead- mednet                          1
          12.30 pm                                   1
          basesas program/ reporting                 1
          10.00 am                                   1
          manager                                    1
          sccm- networking                           1
          technical lead                             1
          9.00 am                                    1
          biosimilars                                1
          9.30 am                                    1
          sccm-(network, sharepoint,ms exchange)     1
          sccm – sharepoint                          1
          sccm - sql                                    1
          Name: Nature of Skillset, dtype: int64
```

In [220]:
```
rawdata['Nature of Skillset'] = cleaned_nature_of_skillset
merge_categories('Nature of Skillset', 50)
rawdata['Nature of Skillset'].value_counts()
```

Out[220]:
```
java                         459
Others                       398
cdd                          136
fresher                       86
accounting operations         86
oracle                        68
Name: Nature of Skillset, dtype: int64
```

In [221]:
```
#We will re-classify them into walkin, scheduled and scheduled_walkin
```

In [222]:
```
replace_dict = {
    'Scheduled Walk In': 'Scheduled Walkin',
    'Sceduled walkin': 'Scheduled Walkin',
    'Walkin ': 'Walkin','scheduled_walkin':'scheduled Walkin',
}
rawdata['Interview Type'].replace(replace_dict, inplace=True)
```

In [223]:
```
rawdata['Interview Type'].unique()
```

Out[223]:
```
array(['Scheduled Walkin', 'Scheduled ', 'Walkin'], dtype=object)
```

In [224]:
```
#Cleaning the location columns
location_columns = [
    'Candidate Current Location', 'Candidate Job Location', 'Interview Venue',
    'Candidate Native location'
]

def clean_location(s):
    s = s.translate(str.maketrans({key: None for key in string.punctuation}))
# remove punctuations
    s = s.lower().strip()

    if 'delhi' in s or 'ncr' in s or 'gurgaon' in s or 'noida' in s:
        return 'ncr'
    else:
        return s

for col in location_columns:
    rawdata[col] = rawdata[col].apply(clean_location)
```

In [225]:
```
rawdata['interview_venue_same_as_current_location'] = rawdata['Candidate Curre
nt Location'] == rawdata['Interview Venue']
rawdata['interview_venue_same_as_native_location'] = rawdata['Candidate Native
location'] == rawdata['Interview Venue']
```

In [226]:
```
merge_categories('Candidate Current Location', 35)
merge_categories('Interview Venue', 35)
merge_categories('Candidate Native location', 40)
```

In [227]:
```
# Rename the long question columns

columns_rename_dict = {
    'Have you obtained the necessary permission to start at the required time'
: 'question_obtained_necessary_permission',
    'Hope there will be no unscheduled meetings': 'question_no_unscheduled_mee
tings',
    'Can I Call you three hours before the interview and follow up on your att
endance for the interview': 'question_can_follow_up',
    'Can I have an alternative number/ desk number. I assure you that I will n
ot trouble you too much': 'question_alternate_number',
    'Have you taken a printout of your updated resume. Have you read the JD an
d understood the same': 'question_taken_printout',
    'Are you clear with the venue details and the landmark.': 'question_clear_
with_venue_details',
    'Has the call letter been shared': 'question_call_letter_shared'
}
rawdata.rename(columns=columns_rename_dict, inplace=True)
```

In [228]:
```
rawdata
```

Out[228]:

	Date of Interview	Client name	Industry	Location	Position to be closed	Nature of Skillset	Interview Type	Gender	Candic Cur Locat
0	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
1	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
2	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
3	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
4	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
...	...	...	...	...	...	...	...	...	
1228	2016-05-07	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1229	2016-05-07	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1230	2016-05-06	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1231	2016-05-06	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1232	2016-05-06	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Female	cher

1233 rows × 24 columns

In [229]:
```
question_columns = [col for col in rawdata.columns if col.startswith('questio
n')]
```

In [230]:
```
question_columns
```

Out[230]:
```
['question_obtained_necessary_permission',
 'question_no_unscheduled_meetings',
 'question_can_follow_up',
 'question_alternate_number',
 'question_taken_printout',
 'question_clear_with_venue_details',
 'question_call_letter_shared']
```

In [231]:
```
def clean_question_answers(a):
    yes_answers = ['yes']
    not_known_answers = ['cant say', 'yet to confirm', 'need to check', 'na',
'not sure']
    no_answers = [
        'no', 'no- i need to check', 'not yet', 'no i have only thi number',
'no dont', 'havent checked',
        'yet to check', 'no- will take it soon'
    ]

    if pd.isna(a):
        return 'not_known'

    a = a.lower().strip()
    if a in yes_answers:
        return 'yes'
    elif a in not_known_answers:
        return 'not_known'
    elif a in no_answers:
        return 'no'

for col in question_columns:
    rawdata[col] = rawdata[col].apply(clean_question_answers)
```

In [232]:
```
rawdata
```

Out[232]:

	Date of Interview	Client name	Industry	Location	Position to be closed	Nature of Skillset	Interview Type	Gender	Candic Curr Locat
0	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
1	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
2	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
3	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
4	2015-02-13	Hospira	Pharmaceuticals	Chennai	Routine	Others	Scheduled Walkin	Male	cher
...	...	...	...	...	...	...	...	...	
1228	2016-05-07	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1229	2016-05-07	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1230	2016-05-06	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1231	2016-05-06	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Male	cher
1232	2016-05-06	Pfizer	Pharmaceuticals	Chennai	Niche	Others	Scheduled	Female	cher

1233 rows × 24 columns

In [233]:
```
#Cleaning the Attendance columns
rawdata['Expected Attendance'].value_counts()
```

Out[233]:
```
Yes          882
Uncertain    250
No            59
NO            34
10.30 Am       1
11:00 AM       1
yes            1
Name: Expected Attendance, dtype: int64
```

In [234]:
```
def clean_expected_attendance(x):
    yes_list = ['yes', '11:00 am', '10.30 am']
    not_known_list = ['uncertain']
    no_list = ['no']

    if pd.isna(x):
        return 'not_known'

    x = x.lower().strip()
    if x in yes_list:
        return 'yes'
    elif x in not_known_list:
        return 'not_known'
    elif x in no_list:
        return 'no'

rawdata['Expected Attendance'] = rawdata['Expected Attendance'].apply(clean_ex
pected_attendance)
```

In [235]:
```
rawdata['Observed Attendance'] = rawdata['Observed Attendance'].apply(lambda x
: x.lower().strip())
rawdata['Observed Attendance'].value_counts()
```

Out[235]:
```
yes    783
no     450
Name: Observed Attendance, dtype: int64
```

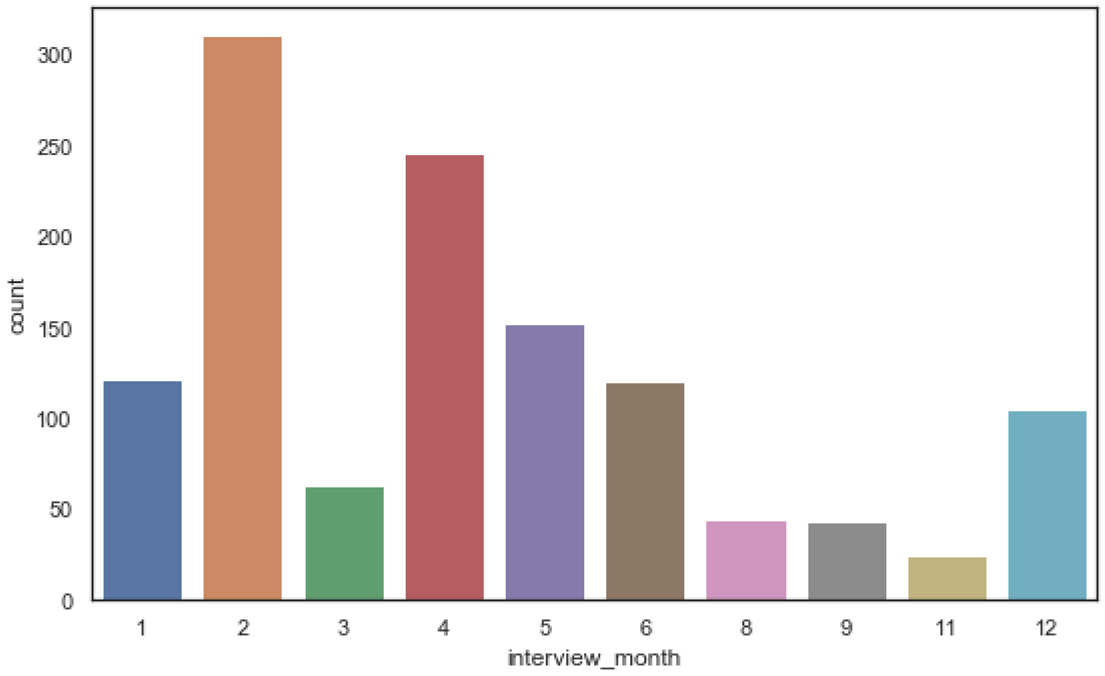
In [236]:
```
# Create new columns for interview date, month and day of week

rawdata['interview_date'] = rawdata['Date of Interview'].apply(lambda x: x.day
)
rawdata['interview_month'] =rawdata['Date of Interview'].apply(lambda x: x.mon
th)
rawdata['interview_day'] = rawdata['Date of Interview'].apply(lambda x: x.dayo
fweek)
```

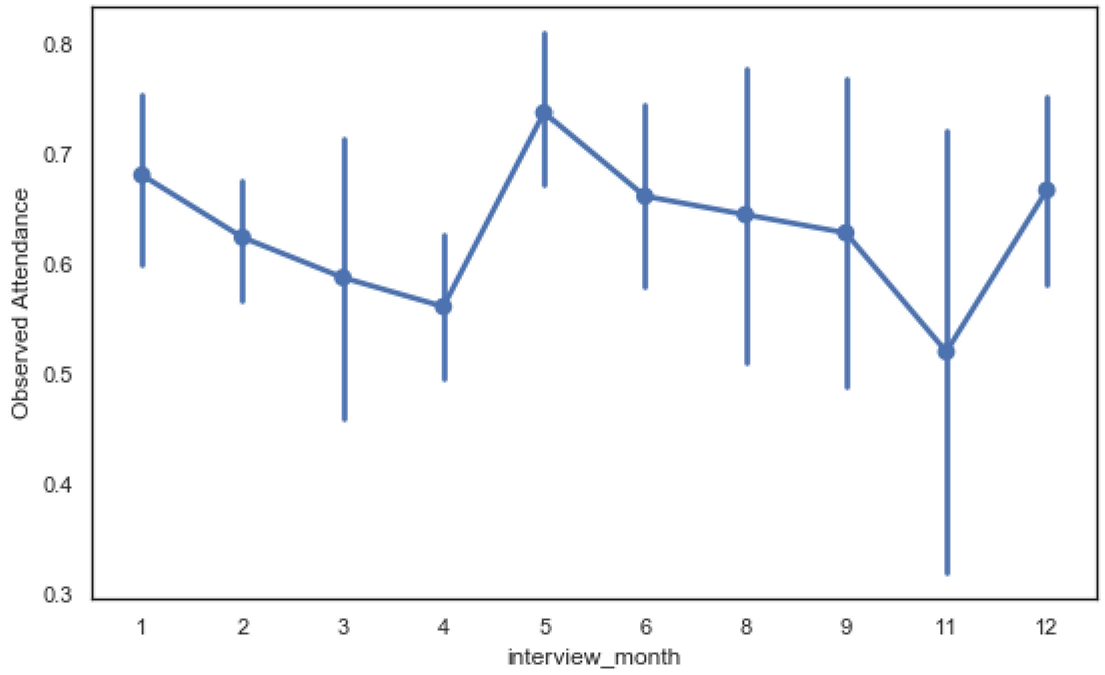
In [237]:
```
sns.set(rc={'figure.figsize': (9, 6)})
sns.set_style('white')
```

In [238]:
```
#giving observed attendance binary value
rawdata['Observed Attendance'].replace({'no': 0, 'yes': 1}, inplace=True)
```

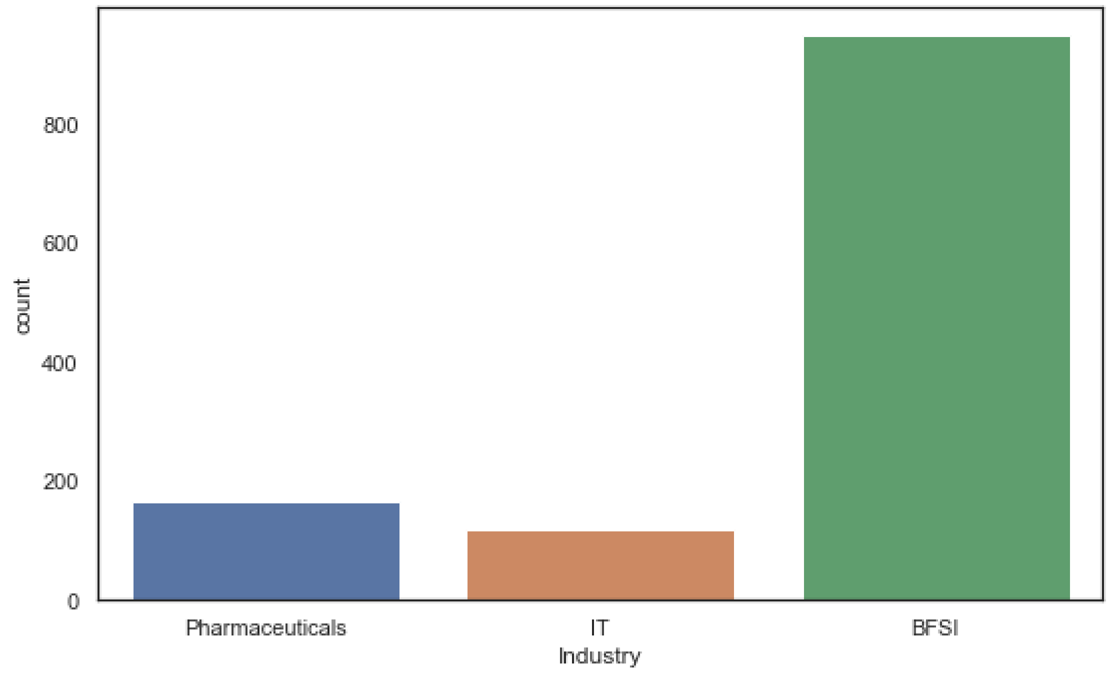
In [239]:
```
def plot_categorical_column(column_name):
    f, (ax1, ax2) = plt.subplots(2, figsize=(9, 12))
    sns.countplot(x=column_name, data=rawdata, ax=ax1)
    sns.pointplot(x=column_name, y='Observed Attendance', data=rawdata, ax=ax2
)
```

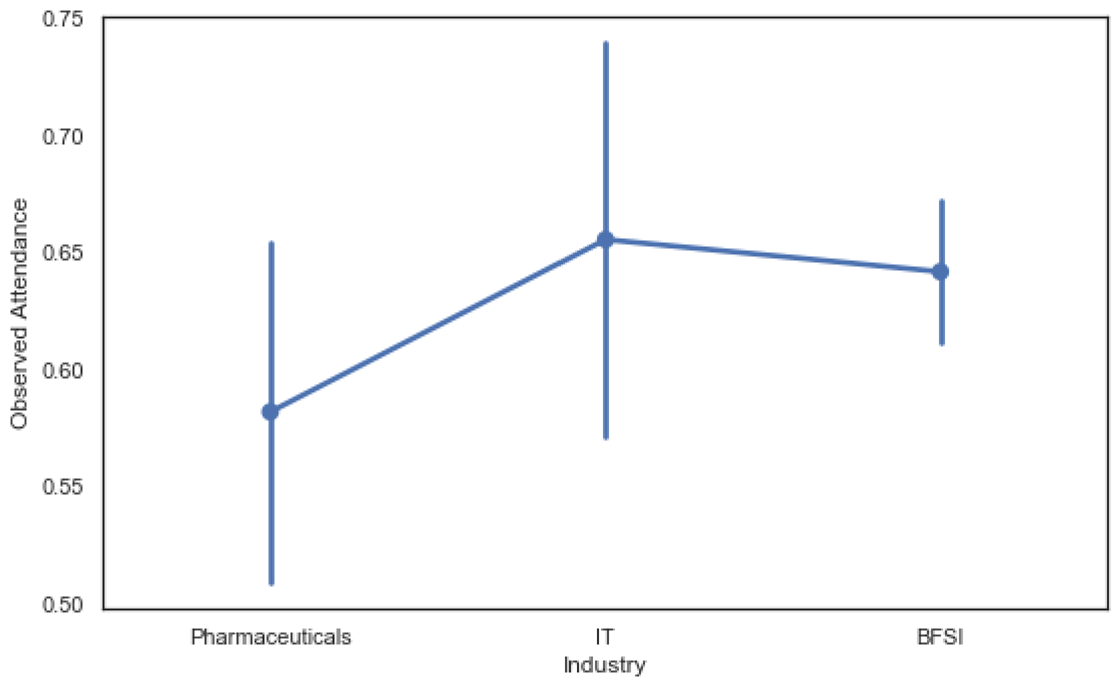
In [240]:
```
plot_categorical_column('interview_month')
```



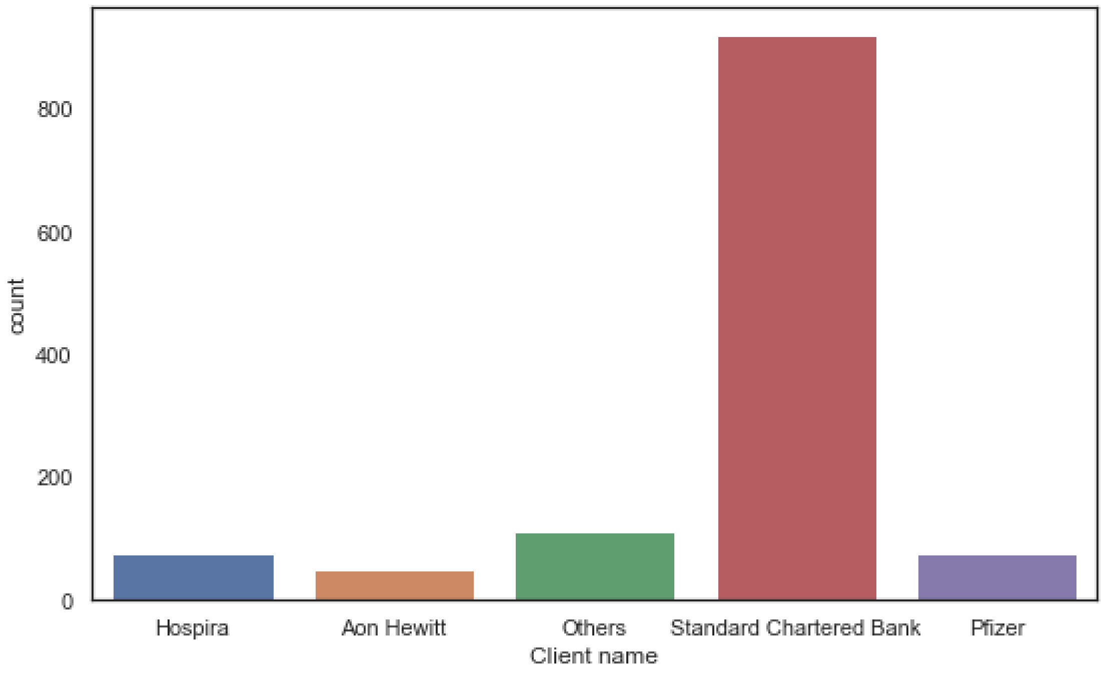


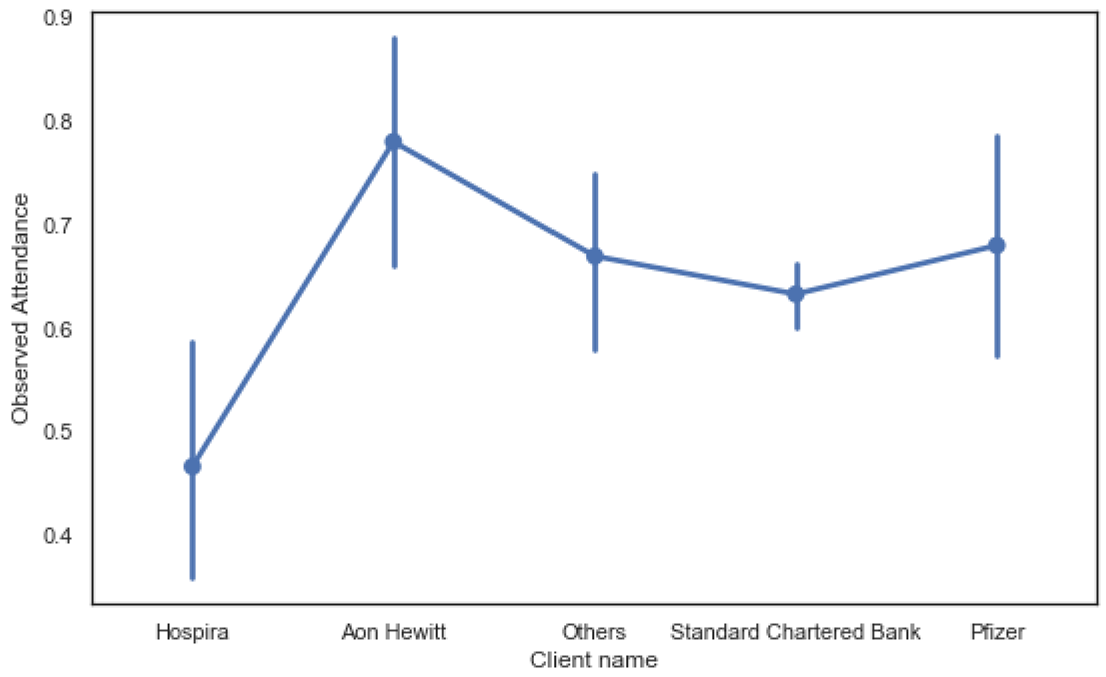
In [241]:
```
plot_categorical_column('Industry')
```



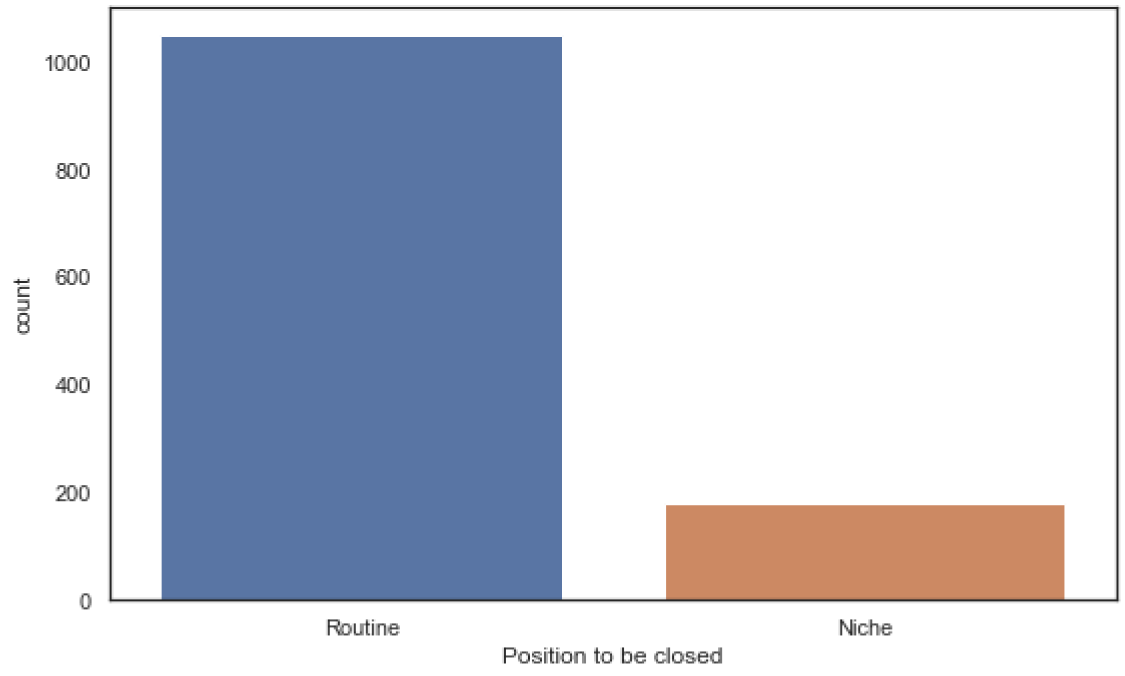


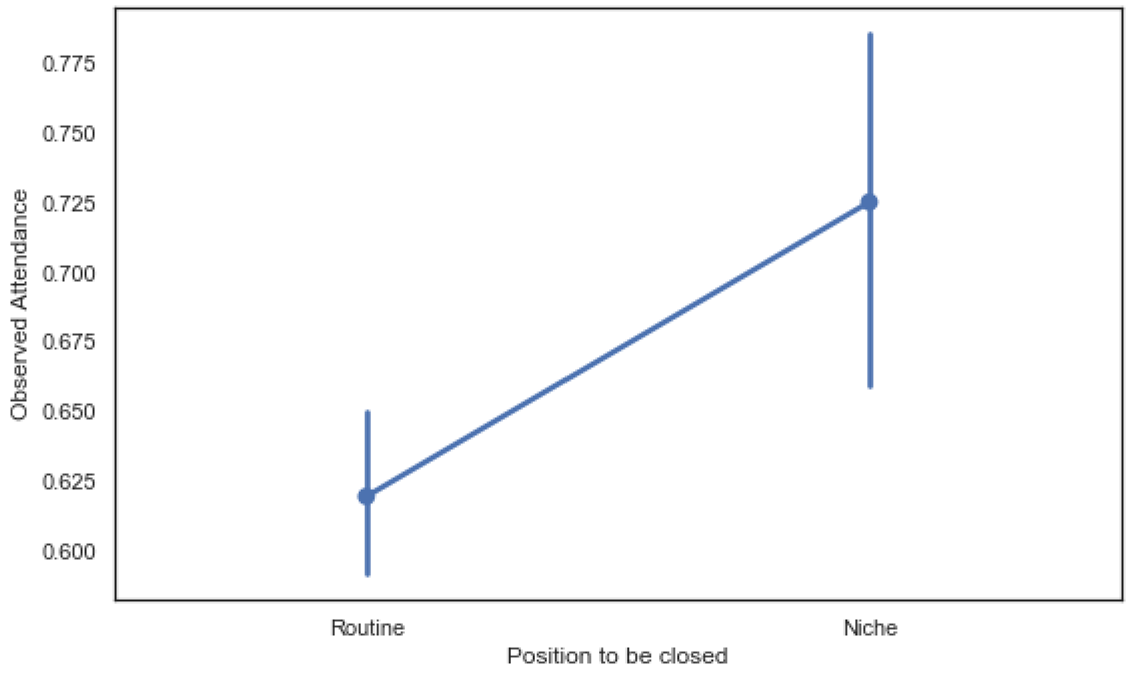
```
In [242]: plot_categorical_column('Client name')
```



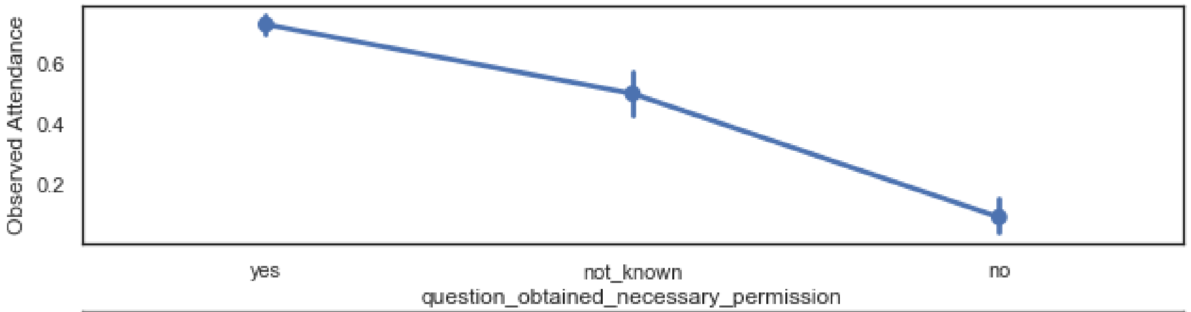


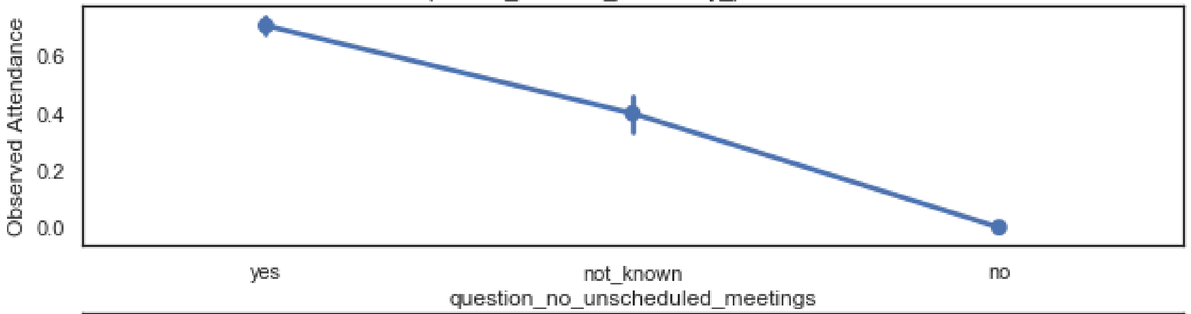
In [243]:
```
plot_categorical_column('Position to be closed')
```

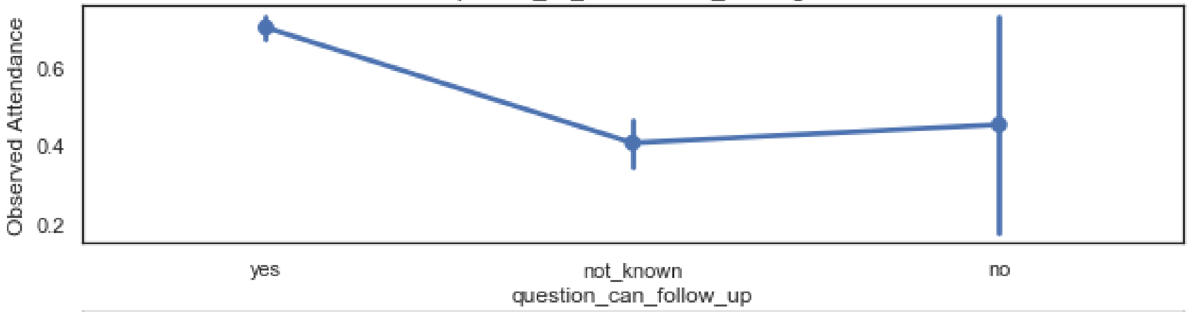


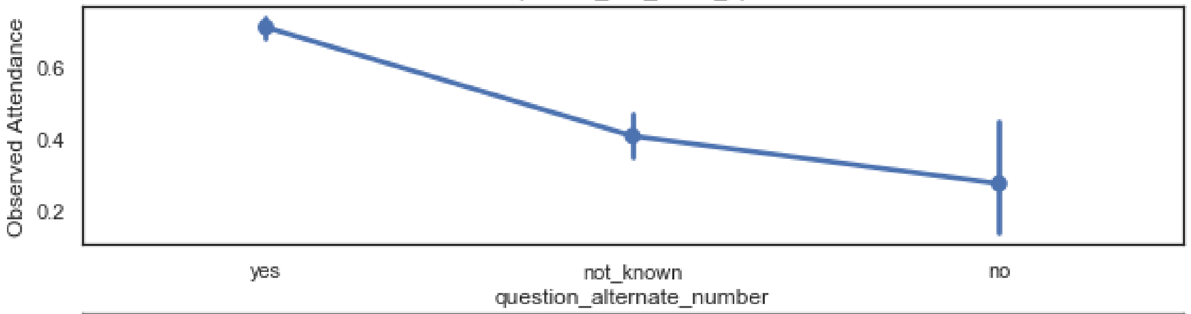


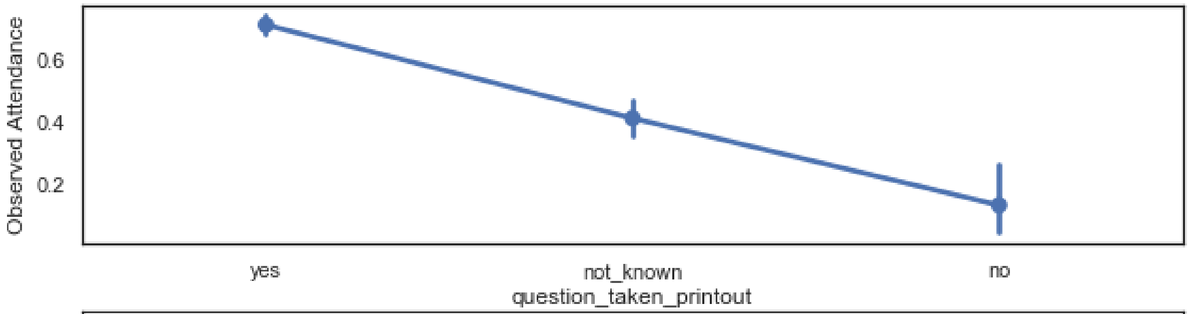
In [244]:
```
f, (ax1, ax2, ax3, ax4, ax5, ax6, ax7) = plt.subplots(7, figsize=(9, 16))
f.tight_layout()
axes = {
    'ax1': ax1,
    'ax2': ax2,
    'ax3': ax3,
    'ax4': ax4,
    'ax5': ax5,
    'ax6': ax6,
    'ax7': ax7
}
for ctr, col in enumerate(question_columns):
    ax_number = 'ax%s' % (ctr+1)
    sns.pointplot(x=col, y='Observed Attendance', data=rawdata, ax=axes[ax_num
ber])
```

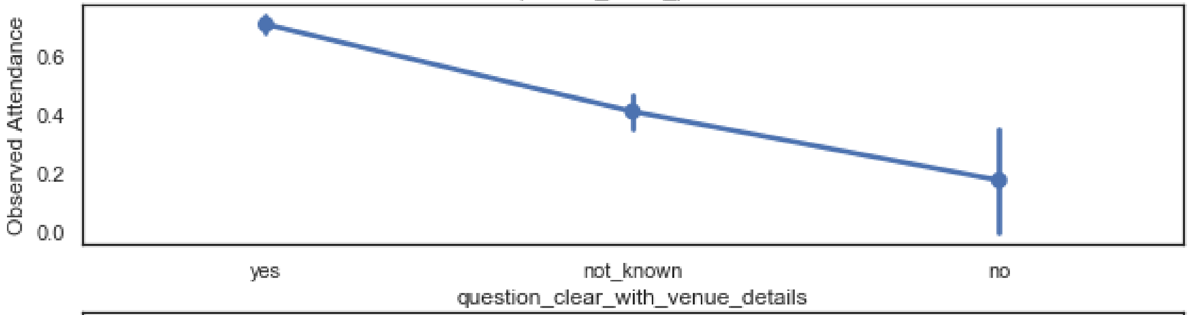

Observed Attendance
0.6
0.4
0.2
yes
not_known
no
question_obtained_necessary_permission

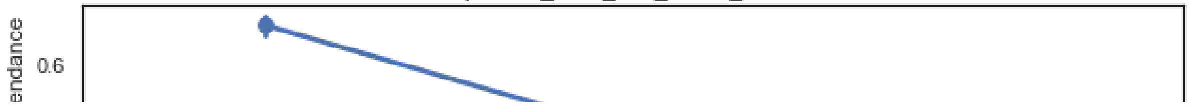

Observed Attendance
0.6
0.4
0.2
0.0
yes
not_known
no
question_no_unscheduled_meetings

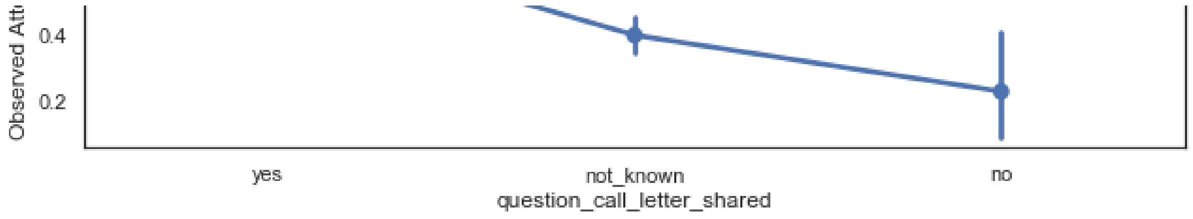

Observed Attendance
0.6
0.4
0.2
yes
not_known
no
question_can_follow_up


Observed Attendance
0.6
0.4
0.2
yes
not_known
no
question_alternate_number


Observed Attendance
0.6
0.4
0.2
yes
not_known
no
question_taken_printout

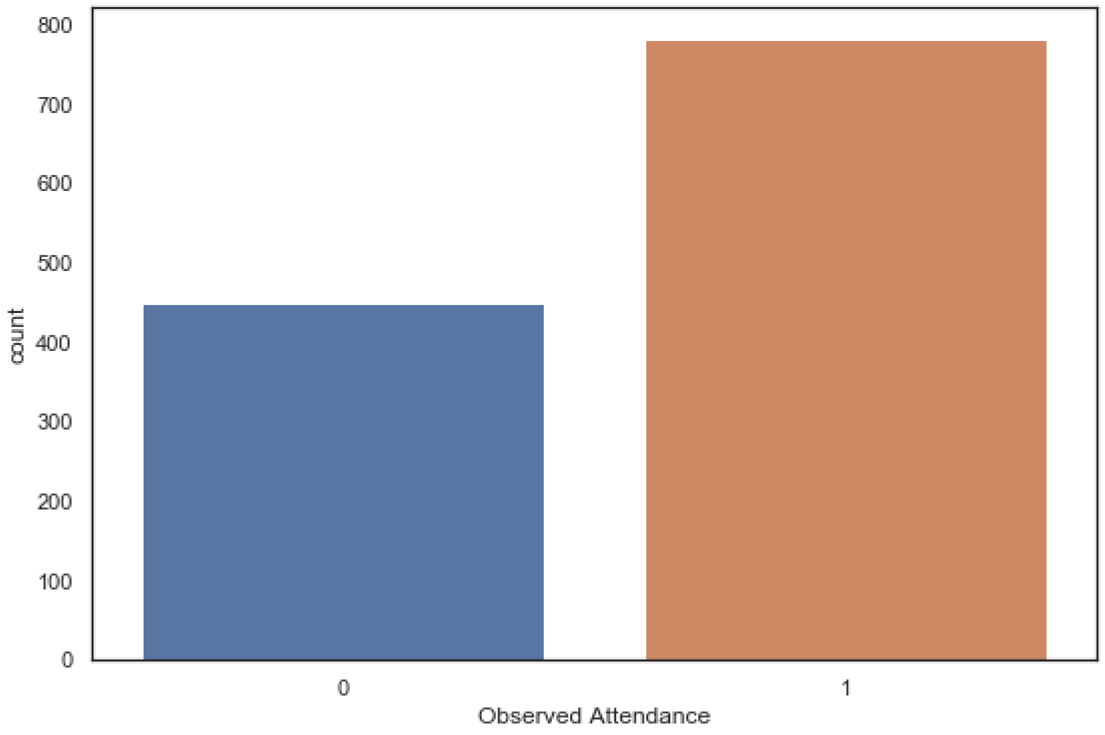

Observed Attendance
0.6
0.4
0.2
0.0
yes
not_known
no
question_clear_with_venue_details


0.6



```
In [245]: sns.countplot(x='Observed Attendance', data=rawdata)
```

```
Out[245]: <matplotlib.axes._subplots.AxesSubplot at 0x25e24597308>
```



```
In [246]: unrequired_columns = [
              'Date of Interview', 'Location',  'Candidate Job Location',
          ]
          rawdata.drop(unrequired_columns, axis=1, inplace=True)
```

```
In [247]: # Create dummy variables for the categorical columns

          categorical_columns = [
              'Client name', 'Industry', 'Nature of Skillset', 'Interview Type', 'Candid
          ate Current Location',
              'Interview Venue', 'Candidate Native location'
          ]
          categorical_columns += question_columns
          for categorical_column in categorical_columns:
              dummy_df = pd.get_dummies(rawdata[categorical_column], prefix=categorical_
          column)
              rawdata = pd.concat([rawdata, dummy_df], axis=1)
              rawdata.drop([categorical_column], axis=1, inplace=True)
```

In [248]:
```
rawdata
```

Out[248]:

	Position to be closed	Gender	Expected Attendance	Observed Attendance	Marital Status	interview_venue_same_as_current_locati
0	Routine	Male	yes	0	Single	Fa
1	Routine	Male	yes	0	Single	Fa
2	Routine	Male	not_known	0	Single	Fa
3	Routine	Male	not_known	0	Single	Fa
4	Routine	Male	not_known	0	Married	Fa
...	...	...	...	...	...	
1228	Niche	Male	yes	1	Married	T
1229	Niche	Male	yes	1	Single	T
1230	Niche	Male	yes	1	Married	T
1231	Niche	Male	yes	1	Single	T
1232	Niche	Female	not_known	1	Single	T

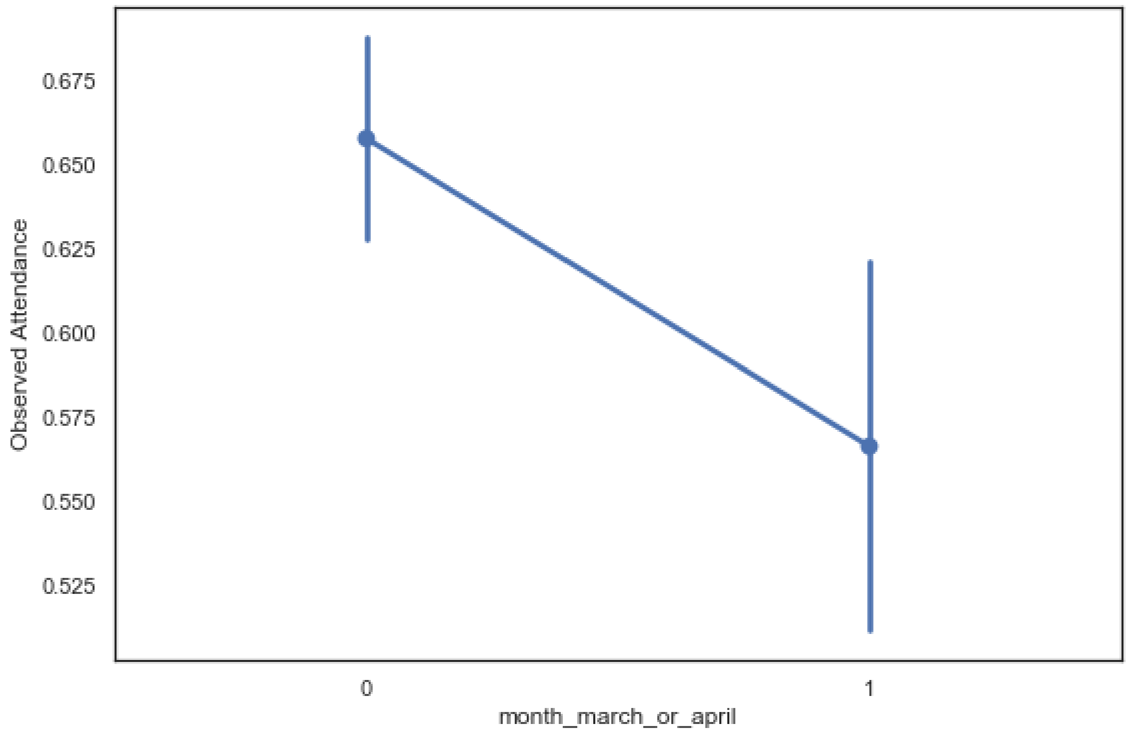
1233 rows × 63 columns

In [249]:
```
# replace binary text values with numbers

binary_columns_replace_dict = {
    'Position to be closed': {
        'Routine': 0,
        'Niche': 1
    },
    'Gender': {
        'Female': 0,
        'Male': 1
    },
    'interview_venue_same_as_current_location': {
        False: 0,
        True: 1
    },
    'interview_venue_same_as_native_location': {
        False: 0,
        True: 1
    },
    'Marital Status': {
        'Single': 0,
        'Married': 1
    },
}

binary_columns = [
    'Position to be closed', 'Gender', 'interview_venue_same_as_current_locati
on',
    'interview_venue_same_as_native_location', 'Marital Status'
]
for binary_col in binary_columns_replace_dict:
    rawdata[binary_col].replace(binary_columns_replace_dict[binary_col], inpla
ce=True)
```

In [250]:
```
# We can see in the pointplots that, observed attendance is low in months of M
arch and April and high in May.
# This could be because the salary hike takes place in these months, so people
tend not to take leaves from
# current company during this period and switch after taking increment.

rawdata['month_march_or_april'] = rawdata['interview_month'].apply(lambda x: 1
if x in [3, 4] else 0)
rawdata['month_may'] = rawdata['interview_month'].apply(lambda x: 1 if x == 5
else 0)
```

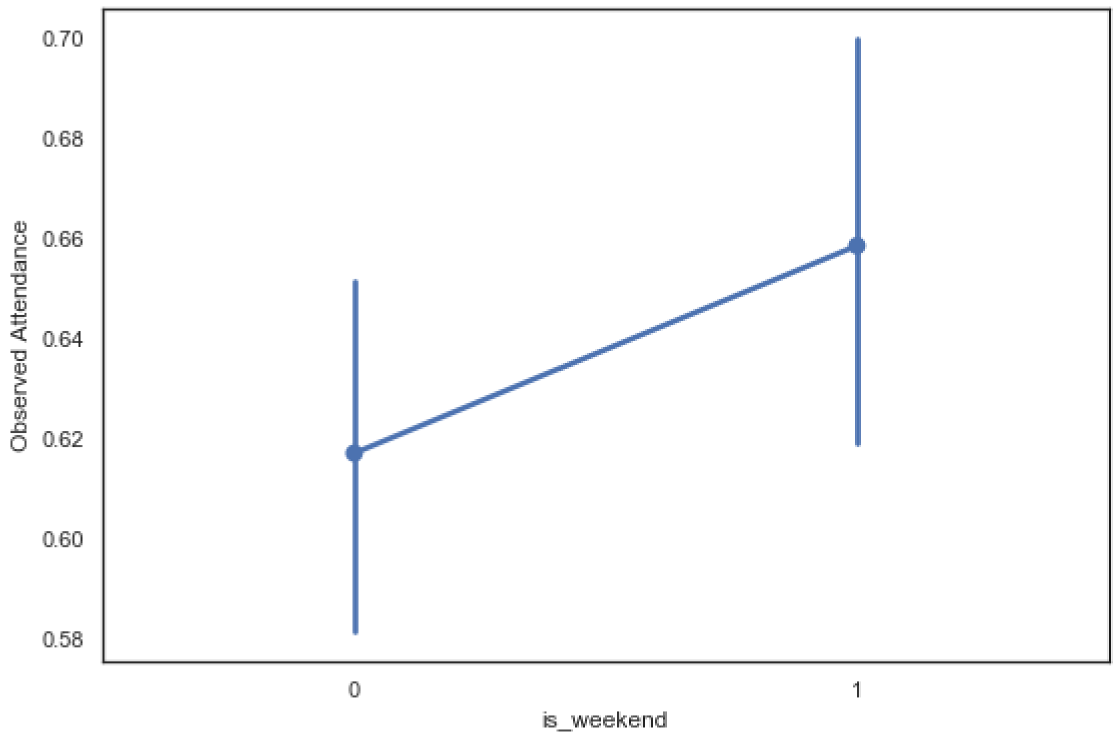
```
In [251]: sns.pointplot(x='month_march_or_april', y='Observed Attendance', data=rawdata)
```

```
Out[251]: <matplotlib.axes._subplots.AxesSubplot at 0x25e2298d588>
```



In [252]:
```
# People prefer going for interviews on weekends, as they don't have to take a
leave from office.

rawdata['is_weekend'] = rawdata['interview_day'].apply(lambda x: 1 if x in [5,
6] else 0)
# Also, we can see from plots that attendance is quite low on fridays, so crea
te a feature for that also
rawdata['day_friday'] = rawdata['interview_day'].apply(lambda x: 1 if x == 4 e
lse 0)
sns.pointplot(x='is_weekend', y='Observed Attendance', data=rawdata)
```

Out[252]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25e21e19908>
```



In [253]:
```
rawdata
```

Out[253]:

	Position to be closed	Gender	Expected Attendance	Observed Attendance	Marital Status	interview_venue_same_as_current_locati
0	0	1	yes	0	0	
1	0	1	yes	0	0	
2	0	1	not_known	0	0	
3	0	1	not_known	0	0	
4	0	1	not_known	0	1	
...	...	...	...	...	...	
1228	1	1	yes	1	1	
1229	1	1	yes	1	0	
1230	1	1	yes	1	1	
1231	1	1	yes	1	0	
1232	1	0	not_known	1	0	

1233 rows × 67 columns

In [256]:
```
rawdata['Expected Attendance'].unique()
```

Out[256]:
```
array(['yes', 'not_known', 'no'], dtype=object)
```

In [259]:
```
rawdata['Expected Attendance']=rawdata['Expected Attendance'].replace({'yes':1
,'not_known':0,'no':1})
```

In [260]:
```
from sklearn.model_selection import train_test_split
rawdata
```

Out[260]:

	Position to be closed	Gender	Expected Attendance	Observed Attendance	Marital Status	interview_venue_same_as_current_locati
0	0	1	1	0	0	
1	0	1	1	0	0	
2	0	1	0	0	0	
3	0	1	0	0	0	
4	0	1	0	0	1	
...	...	...	...	...	...	
1228	1	1	1	1	1	
1229	1	1	1	1	0	
1230	1	1	1	1	1	
1231	1	1	1	1	0	
1232	1	0	0	1	0	

1233 rows × 67 columns

In [261]:
```
y=rawdata.pop("Observed Attendance")
# Split Data into training and test set
X_train1, X_test, y_train1, y_test = train_test_split(rawdata , y, test_size =
0.3, random_state = 100)

# Further split the Training set into training and validation set
X_train, X_val, y_train, y_val = train_test_split( X_train1, y_train1, test_si
ze = 0.3, random_state = 100)
```

In [263]:
```
print("Test set",rawdata.shape, y.shape)
print("Training set",X_train.shape, y_train.shape)
print("Val set", X_val.shape, y_val.shape)
print("Test set",X_test.shape, y_test.shape)
```

```
Test set (1233, 66) (1233,)
Training set (604, 66) (604,)
Val set (259, 66) (259,)
Test set (370, 66) (370,)
```

In [264]:
```
clf_entropy = DecisionTreeClassifier(criterion = "entropy", random_state = 100
,
 max_depth=None, min_samples_leaf=5)
clf_entropy.fit(X_train1, y_train1)
clf_entropy
```

Out[264]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=Non
e,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=5, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=100, splitter='best')
```

In [282]:
```
y_pred_en = clf_entropy.predict(X_test)
print("Accuracy is using information gain ", accuracy_score(y_test,y_pred_en)*
100)
```

```
Accuracy is using information gain  63.78378378378379
```

In [285]:
```
clf_gini = DecisionTreeClassifier(criterion = "gini", random_state = 100,
max_depth=None, min_samples_leaf=4)
clf_gini=clf_gini.fit(X_train1,y_train1)
clf_gini
```

Out[285]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=4, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=100, splitter='best')
```

In [286]:
```
y_pred = clf_gini.predict(X_test)
#y_pred
print("Accuracy is uning gini index", accuracy_score(y_test,y_pred)*100)
```

```
Accuracy is uning gini index 64.05405405405405
```

In [280]:
```
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_absolute_error
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
```

```
Accuracy is using information gain  63.78378378378379
```

In [ ]:
```

```

In [268]:
```
from sklearn.ensemble import RandomForestRegressor
```

In [270]:
```
model=RandomForestRegressor(random_state=1)
```

In [271]:
```
model.fit(X_train,y_train)
```

Out[271]:
```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
                      max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=2,
                      min_weight_fraction_leaf=0.0, n_estimators=10,
                      n_jobs=None, oob_score=False, random_state=1, verbose=
0,
                      warm_start=False)
```

In [276]:
```
p=model.predict(X_val)
```

In [281]:
```
print(mean_absolute_error(y_val,p))
```

```
0.36160492171107383
```

In [ ]: