```
In [ ]:
```

About Dataset

This experiment data comes from a study that sought to understand the influence of race and gender on job application callback rates. The study monitored job postings in Boston and Chicago for several months during 2001 and 2002 and used this to build up a set of test cases. Over this time period, the researchers randomly generating resumes to go out to a job posting, such as years of experience and education details, to create a realistic-looking resume. They then randomly assigned a name to the resume that would communicate the applicant's gender and race. The first names chosen for the study were selected so that the names would predominantly be recognized as belonging to black or white individuals. For example, Lakisha was a name that their survey indicated would be interpreted as a black woman, while Greg was a name that would generally be interpreted to be associated with a white male.

- Column Description

- job_ad_id Unique ID associated with the advertisement.
- job_city City where the job was located.
- job_industry Industry of the job.
- job_type Type of role.
- job_fed_contractor Indicator for if the employer is a federal contractor.
- job_equal_opp_employer Indicator for if the employer is an Equal Opportunity Employer.
- job_ownership The type of company, e.g. a nonprofit or a private company.
- job_req_any Indicator for if any job requirements are listed. If so, the other job_req_* fields give more detail.
- job_req_communication Indicator for if communication skills are required.
- job_req_education Indicator for if some level of education is required.
- job_req_min_experience Amount of experience required.
- job_req_computer Indicator for if computer skills are required.
- job_req_organization Indicator for if organization skills are required.
- job_req_school Level of education required.
- received_callback Indicator for if there was a callback from the job posting for the person listed on this resume.
- firstname The first name used on the resume.
- race Inferred race associated with the first name on the resume.
- gender Inferred gender associated with the first name on the resume.
- years_college Years of college education listed on the resume.
- college_degree Indicator for if the resume listed a college degree.
- honors Indicator for if the resume listed that the candidate has beenawarded some honors.
- worked_during_school Indicator for if the resume listed working while in school.
- years_experience Years of experience listed on the resume.
- computer_skills Indicator for if computer skills were listed on the resume. These skills were adapted for listings, though the skills were assigned independently of other details on the resume.
- special_skills Indicator for if any special skills were listed on the resume.
- volunteer Indicator for if volunteering was listed on the resume.
- military Indicator for if military experience was listed on the resume.
- employment_holes Indicator for if there were holes in the person's employment history.

- has_email_address Indicator for if the resume lists an email address.

- resume_quality Each resume was generally classified as either lower or higher quality.
- Details

Because this is an experiment, where the race and gender attributes are being randomly assigned to the resumes, we can conclude that any statistically significant difference in callback rates is causally linked to these attributes.

Do you think it's reasonable to make a causal conclusion? You may have some health skepticism. However, do take care to appreciate that this was an experiment: the first name (and so the inferred race and gender) were randomly assigned to the resumes, and the quality and attributes of a resume were assigned independent of the race and gender. This means that any effects we observe are in fact causal, and the effects related to race are both statistically significant and very large: white applicants had about a 50\

Do you still have doubts lingering in the back of your mind about the validity of this study? Maybe a counterargument about why the standard conclusions from this study may not apply? The article summarizing the results was exceptionally well-written, and it addresses many potential concerns about the study's approach. So if you're feeling skeptical about the conclusions, please find the link below and explore!

Source Bertrand M, Mullainathan S. 2004. "Are Emily and Greg More Employable than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination". The American Economic Review 94:4 (991-1013). \Sexpr[results=rd,stage=build]{tools:::Rd_expr_doi("10.3386/w9873")}.

Import Liberaries

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import warnings
import plotly.express as px
warnings.filterwarnings('ignore')
```

In [2]:

```
# display all the columns of the dataset

pd.pandas .set_option('display.max_columns',None)
```

In [3]:

```
# import dataset to pandas dataframe

df = pd.read_csv(r'C:\Users\Mukul\Downloads\resume.csv')
```

In [4]:

```
# check the 5 row
df.head()
```

Out[4]:

	job_ad_id	job_city	job_industry	job_type	job_fed_contractor	job_equal_opp_employer
0	384	Chicago	manufacturing	supervisor	NaN	1
1	384	Chicago	manufacturing	supervisor	NaN	1
2	384	Chicago	manufacturing	supervisor	NaN	1
3	384	Chicago	manufacturing	supervisor	NaN	1
4	385	Chicago	other_service	secretary	0.0	1

In [5]:

```
# check the last 5 rows
df.tail()
```

Out[5]:

	job_ad_id	job_city	job_industry	job_type	job_fed_contractor	job_equa
4865	1344	Boston	finance_insurance_real_estate	secretary	0.0	
4866	382	Boston	other_service	manager	NaN	
4867	382	Boston	other_service	manager	NaN	
4868	382	Boston	other_service	manager	NaN	
4869	382	Boston	other_service	manager	NaN	

In [6]:

```
# check the columns
df.columns
```

Out[6]:

```
Index(['job_ad_id', 'job_city', 'job_industry', 'job_type',
       'job_fed_contractor', 'job_equal_opp_employer', 'job_ownership',
       'job_req_any', 'job_req_communication', 'job_req_education',
       'job_req_min_experience', 'job_req_computer', 'job_req_organizatio
n',
       'job_req_school', 'received_callback', 'firstname', 'race', 'gende
r',
       'years_college', 'college_degree', 'honors', 'worked_during_schoo
l',
       'years_experience', 'computer_skills', 'special_skills', 'voluntee
r',
       'military', 'employment_holes', 'has_email_address', 'resume_qualit
y'],
      dtype='object')
```

In [7]:

```
# check shape of dataframe
df.shape
```

Out[7]:

```
(4870, 30)
```

In [8]:

```
# check the information of dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4870 entries, 0 to 4869
Data columns (total 30 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   job_ad_id               4870 non-null   int64  
 1   job_city                4870 non-null   object 
 2   job_industry            4870 non-null   object 
 3   job_type                4870 non-null   object 
 4   job_fed_contractor      3102 non-null   float64
 5   job_equal_opp_employer  4870 non-null   int64  
 6   job_ownership           4870 non-null   object 
 7   job_req_any             4870 non-null   int64  
 8   job_req_communication   4870 non-null   int64  
 9   job_req_education       4870 non-null   int64  
 10  job_req_min_experience  2124 non-null   object 
 11  job_req_computer        4870 non-null   int64  
 12  job_req_organization    4870 non-null   int64  
 13  job_req_school          4870 non-null   object 
 14  received_callback       4870 non-null   int64  
 15  firstname               4870 non-null   object 
 16  race                    4870 non-null   object 
 17  gender                  4870 non-null   object 
 18  years_college           4870 non-null   int64  
 19  college_degree          4870 non-null   int64  
 20  honors                  4870 non-null   int64  
 21  worked_during_school    4870 non-null   int64  
 22  years_experience        4870 non-null   int64  
 23  computer_skills         4870 non-null   int64  
 24  special_skills          4870 non-null   int64  
 25  volunteer               4870 non-null   int64  
 26  military                4870 non-null   int64  
 27  employment_holes        4870 non-null   int64  
 28  has_email_address       4870 non-null   int64  
 29  resume_quality          4870 non-null   object 
dtypes: float64(1), int64(19), object(10)
memory usage: 1.1+ MB
```

- it is indicated the 1 float column , 19 integer column and 10 object columns

In [9]:

```
# check the data types
df.dtypes
```

Out[9]:

```
job_ad_id                     int64
job_city                     object
job_industry                 object
job_type                     object
job_fed_contractor          float64
job_equal_opp_employer        int64
job_ownership                object
job_req_any                   int64
job_req_communication         int64
job_req_education             int64
job_req_min_experience       object
job_req_computer              int64
job_req_organization          int64
job_req_school               object
received_callback             int64
firstname                    object
race                         object
gender                       object
years_college                 int64
college_degree                int64
honors                        int64
worked_during_school          int64
years_experience              int64
computer_skills               int64
special_skills                int64
volunteer                     int64
military                      int64
employment_holes              int64
has_email_address             int64
resume_quality               object
dtype: object
```

In [10]:

```
# check the missing values
df.isnull().sum()
```

Out[10]:

```
job_ad_id                   0
job_city                    0
job_industry                0
job_type                    0
job_fed_contractor       1768
job_equal_opp_employer      0
job_ownership               0
job_req_any                 0
job_req_communication       0
job_req_education           0
job_req_min_experience   2746
job_req_computer            0
job_req_organization        0
job_req_school              0
received_callback           0
firstname                   0
race                        0
gender                      0
years_college               0
college_degree              0
honors                      0
worked_during_school        0
years_experience            0
computer_skills             0
special_skills              0
volunteer                   0
military                    0
employment_holes            0
has_email_address           0
resume_quality              0
dtype: int64
```

- it is indicated the two columns are missing values 1.. job_fed_contracter 2.. job_req_min_experience

In [11]:

```
# check the total value of missing value
df.isnull().sum().sum()
```

Out[11]:

```
4514
```

In [12]:

```
# check the % of missing values

df.isnull().mean()*100
```

Out[12]:

```
job_ad_id                  0.000000
job_city                   0.000000
job_industry               0.000000
job_type                   0.000000
job_fed_contractor        36.303901
job_equal_opp_employer     0.000000
job_ownership              0.000000
job_req_any                0.000000
job_req_communication      0.000000
job_req_education          0.000000
job_req_min_experience    56.386037
job_req_computer           0.000000
job_req_organization       0.000000
job_req_school             0.000000
received_callback          0.000000
firstname                  0.000000
race                       0.000000
gender                     0.000000
years_college              0.000000
college_degree             0.000000
honors                     0.000000
worked_during_school       0.000000
years_experience           0.000000
computer_skills            0.000000
special_skills             0.000000
volunteer                  0.000000
military                   0.000000
employment_holes           0.000000
has_email_address          0.000000
resume_quality             0.000000
dtype: float64
```

In [13]:

```
# check the missing values in ascending order

(df.isnull().mean().sort_values(ascending=False)[0:6])*100
```

Out[13]:

```
job_req_min_experience    56.386037
job_fed_contractor        36.303901
job_ad_id                  0.000000
race                       0.000000
has_email_address          0.000000
employment_holes           0.000000
dtype: float64
```

In [14]:

```
# check the unique values
df.nunique()
```

Out[14]:

```
job_ad_id                    1323
job_city                        2
job_industry                    6
job_type                        6
job_fed_contractor              2
job_equal_opp_employer          2
job_ownership                   4
job_req_any                     2
job_req_communication           2
job_req_education               2
job_req_min_experience         12
job_req_computer                2
job_req_organization            2
job_req_school                  4
received_callback               2
firstname                      36
race                            2
gender                          2
years_college                   5
college_degree                  2
honors                          2
worked_during_school            2
years_experience               26
computer_skills                 2
special_skills                  2
volunteer                       2
military                        2
employment_holes                2
has_email_address               2
resume_quality                  2
dtype: int64
```

In [15]:

```
# check the sort value of unique value
df.nunique().sort_values(ascending=False)
```

Out[15]:

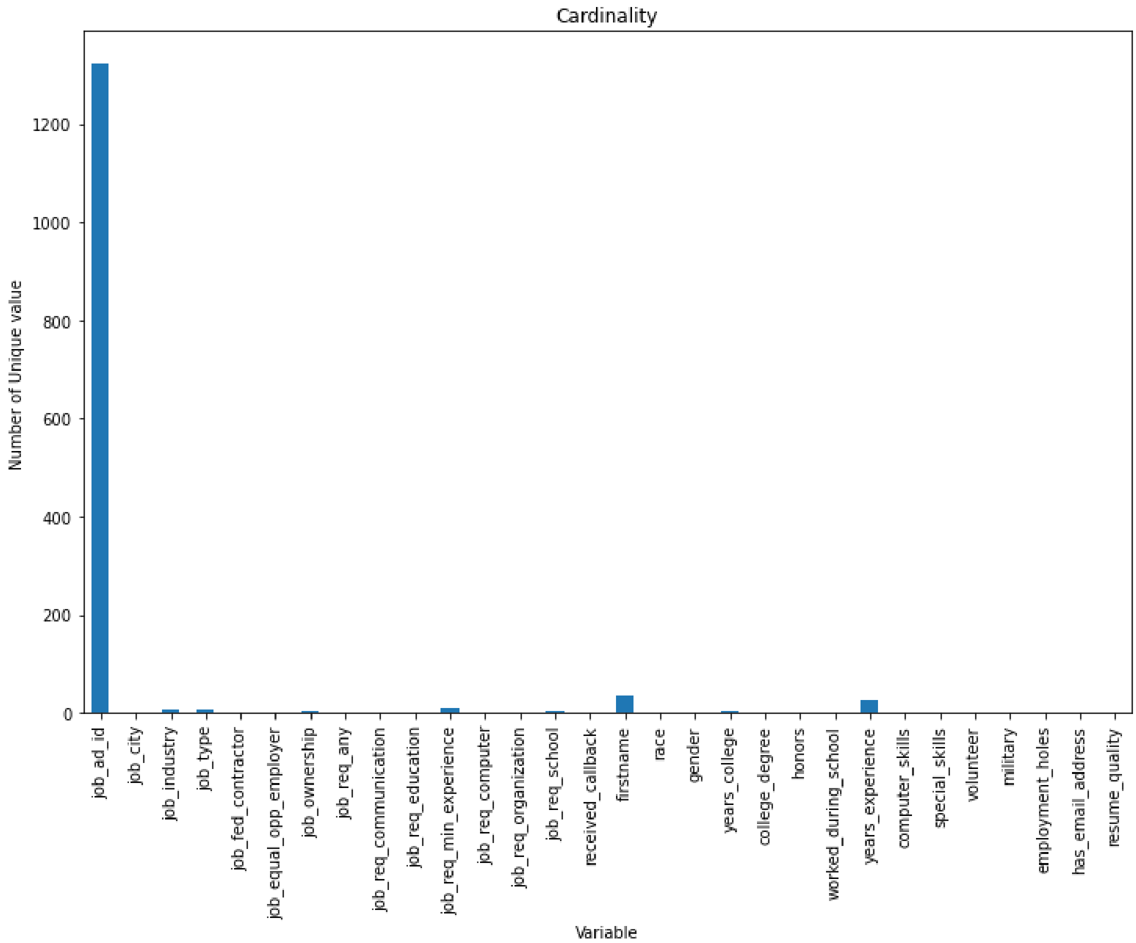
```
job_ad_id                 1323
firstname                   36
years_experience            26
job_req_min_experience      12
job_industry                 6
job_type                     6
years_college                5
job_ownership                4
job_req_school               4
has_email_address            2
employment_holes             2
military                     2
volunteer                    2
special_skills               2
computer_skills              2
worked_during_school         2
honors                       2
college_degree               2
race                         2
gender                       2
job_city                     2
received_callback            2
job_req_organization         2
job_req_computer             2
job_req_education            2
job_req_communication        2
job_req_any                  2
job_equal_opp_employer       2
job_fed_contractor           2
resume_quality               2
dtype: int64
```

In [16]:

```
#check the unique value in graph

df.nunique().plot.bar(figsize=(12,8))
plt.xlabel('Variable')
plt.ylabel('Number of Unique value')
plt.title ('Cardinality')
```

Out[16]:

```
Text(0.5, 1.0, 'Cardinality')
```



- Here is to many unique value

In [17]:

```
# check the duplicated values
df.duplicated().sum()
```

Out[17]:

```
0
```

- it is indicated the no duplictead values

In [18]:

```
# check the mempry_usage
df.memory_usage()
```

Out[18]:

```
Index                      128
job_ad_id                38960
job_city                 38960
job_industry             38960
job_type                 38960
job_fed_contractor       38960
job_equal_opp_employer   38960
job_ownership            38960
job_req_any              38960
job_req_communication    38960
job_req_education        38960
job_req_min_experience   38960
job_req_computer         38960
job_req_organization     38960
job_req_school           38960
received_callback        38960
firstname                38960
race                     38960
gender                   38960
years_college            38960
college_degree           38960
honors                   38960
worked_during_school     38960
years_experience         38960
computer_skills          38960
special_skills           38960
volunteer                38960
military                 38960
employment_holes         38960
has_email_address        38960
resume_quality           38960
dtype: int64
```

In [19]:

```
# check the sample of data
df.sample()
```

Out[19]:

	job_ad_id	job_city	job_industry	job_type	job_fed_contractor	job_equal_(
3553	70	Boston	wholesale_and_retail_trade	manager	0.0	

Handling the Null Values

In [20]:

```
df.isnull().sum()
```

Out[20]:

```
job_ad_id                    0
job_city                     0
job_industry                 0
job_type                     0
job_fed_contractor        1768
job_equal_opp_employer       0
job_ownership                0
job_req_any                  0
job_req_communication        0
job_req_education            0
job_req_min_experience    2746
job_req_computer             0
job_req_organization         0
job_req_school               0
received_callback            0
firstname                    0
race                         0
gender                       0
years_college                0
college_degree               0
honors                       0
worked_during_school         0
years_experience             0
computer_skills              0
special_skills               0
volunteer                    0
military                     0
employment_holes             0
has_email_address            0
resume_quality               0
dtype: int64
```

In [21]:

```
df['job_fed_contractor'].unique()
```

Out[21]:

```
array([nan,  0.,  1.])
```

In [22]:

```
df['job_fed_contractor'].value_counts()
```

Out[22]:

```
0.0    2746
1.0     356
Name: job_fed_contractor, dtype: int64
```

In [23]:

```
df['job_fed_contractor'].dtypes
```

Out[23]:

```
dtype('float64')
```

In [24]:

```
df['job_req_min_experience'].unique()
```

Out[24]:

```
array(['5', 'some', nan, '3', '2', '1', '8', '7', '0.5', '10', '0', '4',
       '6'], dtype=object)
```

In [25]:

```
df['job_req_min_experience'].dtypes
```

Out[25]:

```
dtype('O')
```

In [26]:

```
df['job_req_min_experience'].value_counts()
```

Out[26]:

```
some    1064
2        356
3        331
5        163
1        142
10        18
7         12
8         10
0.5        8
4          8
6          8
0          4
Name: job_req_min_experience, dtype: int64
```

- We are observed there are two feature are missing value 1 ... Job_fed_contractor 2... job_req_min_experiance

1... JOB_FED_CONTRACTOR ---> This feature are float dtypes.We are decided the filling missing values by ----- mean()

2.... JOB_REQ_MIN_EXPERIANCE----> This feature are indicated the object dtypes but we are observerd it is numerical feature and one data is "categorical"

In [27]:

```
# Handling the missinh values
df['job_fed_contractor'] = df['job_fed_contractor'].fillna(df['job_fed_contractor'].mean
```

In [28]:

```
df['job_req_min_experience'] = df['job_req_min_experience'].map({"some":9})
df['job_req_min_experience'] = df['job_req_min_experience'].fillna(df['job_req_min_exper
```

In [ ]:

```
```

In [ ]:

```
```

Drop the columns

In [29]:

```
df.drop(['job_ad_id'],axis=1,inplace=True)
```

In [30]:

```
df.shape
```

Out[30]:

```
(4870, 29)
```

- 4870 rows and 29 columns

In [31]:

```python
# getting the count of each categories from data

for feature in df.columns:
    print(df[feature].value_counts)
```

```
<bound method IndexOpsMixin.value_counts of 0       Chicago
1       Chicago
2       Chicago
3       Chicago
4       Chicago
         ...
4865     Boston
4866     Boston
4867     Boston
4868     Boston
4869     Boston
Name: job_city, Length: 4870, dtype: object>
<bound method IndexOpsMixin.value_counts of 0                      man
ufacturing
1                       manufacturing
2                       manufacturing
3                       manufacturing
4                       other_service
                    ...
```

In [32]:

```
# Getting the counts of each categories from unique values

for feature in df.columns:
    print(df[feature].unique())
```

```
['Chicago' 'Boston']
['manufacturing' 'other_service' 'wholesale_and_retail_trade'
 'business_and_personal_service' 'finance_insurance_real_estate'
 'transportation_communication']
['supervisor' 'secretary' 'sales_rep' 'retail_sales' 'manager' 'clerical']
[0.11476467 0.         1.        ]
[1 0]
['unknown' 'nonprofit' 'private' 'public']
[1 0]
[0 1]
[0 1]
[9.]
[1 0]
[0 1]
['none_listed' 'some_college' 'college' 'high_school_grad']
[0 1]
['Allison' 'Kristen' 'Lakisha' 'Latonya' 'Carrie' 'Jay' 'Jill' 'Kenya'
 'Tyrone' 'Aisha' 'Geoffrey' 'Matthew' 'Tamika' 'Leroy' 'Todd' 'Greg'
 'Keisha' 'Brad' 'Laurie' 'Meredith' 'Anne' 'Emily' 'Latoya' 'Ebony'
 'Brendan' 'Hakim' 'Jamal' 'Neil' 'Tremayne' 'Brett' 'Darnell' 'Sarah'
 'Jermaine' 'Tanisha' 'Rasheed' 'Kareem']
['white' 'black']
['f' 'm']
[4 3 1 2 0]
[1 0]
[0 1]
[0 1]
[ 6 22  5 21  3  8  4  2  7  9 13 19 12 11 10 23  1 14 18 26 15 25 16 20
 17 44]
[1 0]
[0 1]
[0 1]
[0 1]
[1 0]
[0 1]
['low' 'high']
```

Segregated the Numerical and Categorical Columns

In [33]:

```
[feature for feature in df.columns]
```

Out[33]:

```
['job_city',
 'job_industry',
 'job_type',
 'job_fed_contractor',
 'job_equal_opp_employer',
 'job_ownership',
 'job_req_any',
 'job_req_communication',
 'job_req_education',
 'job_req_min_experience',
 'job_req_computer',
 'job_req_organization',
 'job_req_school',
 'received_callback',
 'firstname',
 'race',
 'gender',
 'years_college',
 'college_degree',
 'honors',
 'worked_during_school',
 'years_experience',
 'computer_skills',
 'special_skills',
 'volunteer',
 'military',
 'employment_holes',
 'has_email_address',
 'resume_quality']
```

In [34]:

```
# Numerical Feature
num_feature = [feature for feature in df.columns if df[feature].dtypes!='object']
print("Number of Numerical Feature ",  len(num_feature))
print(num_feature)
```

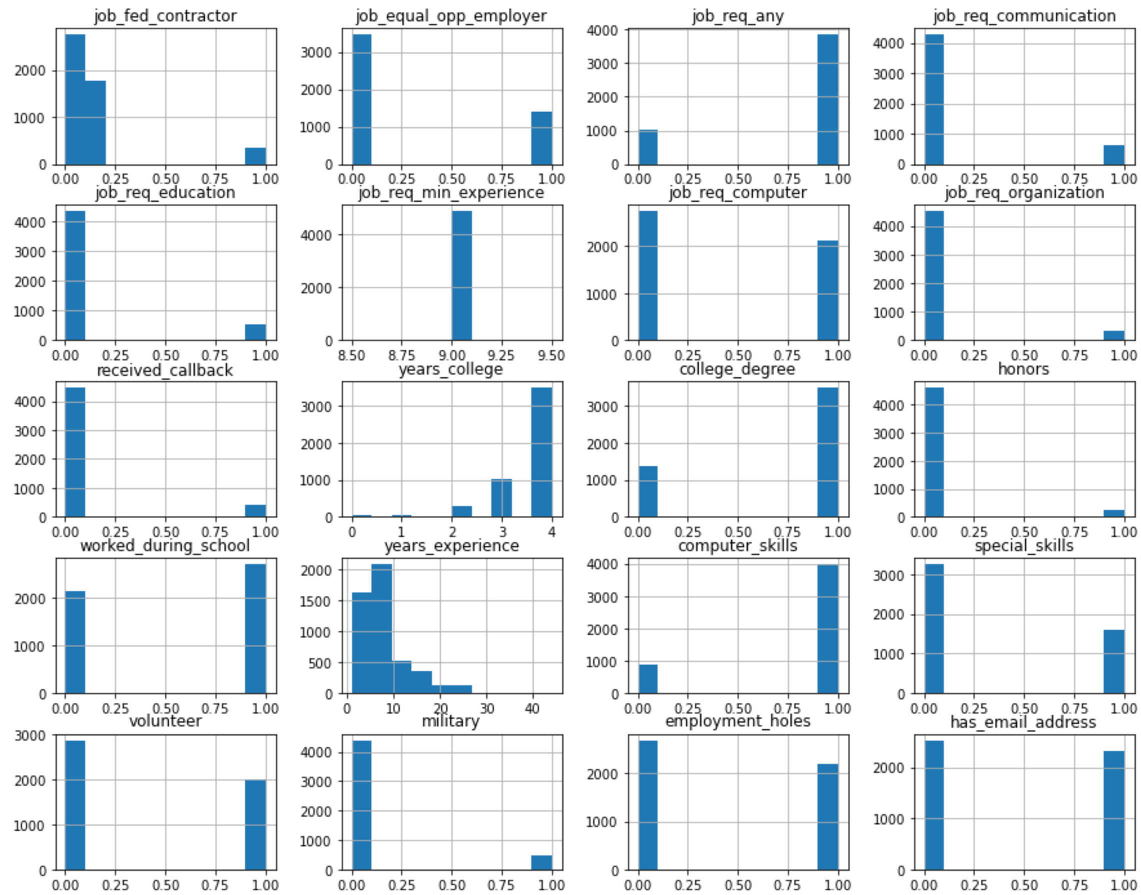
```
Number of Numerical Feature  20
['job_fed_contractor', 'job_equal_opp_employer', 'job_req_any', 'job_req_c
ommunication', 'job_req_education', 'job_req_min_experience', 'job_req_com
puter', 'job_req_organization', 'received_callback', 'years_college', 'col
lege_degree', 'honors', 'worked_during_school', 'years_experience', 'compu
ter_skills', 'special_skills', 'volunteer', 'military', 'employment_hole
s', 'has_email_address']
```

In [35]:

```
df.hist(figsize=(15,12))
```

Out[35]:

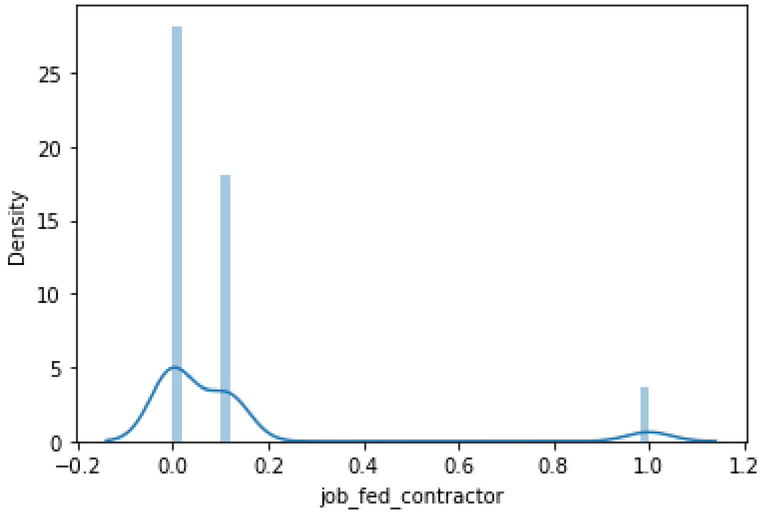
```
array([[<AxesSubplot:title={'center':'job_fed_contractor'}>,
        <AxesSubplot:title={'center':'job_equal_opp_employer'}>,
        <AxesSubplot:title={'center':'job_req_any'}>,
        <AxesSubplot:title={'center':'job_req_communication'}>],
       [<AxesSubplot:title={'center':'job_req_education'}>,
        <AxesSubplot:title={'center':'job_req_min_experience'}>,
        <AxesSubplot:title={'center':'job_req_computer'}>,
        <AxesSubplot:title={'center':'job_req_organization'}>],
       [<AxesSubplot:title={'center':'received_callback'}>,
        <AxesSubplot:title={'center':'years_college'}>,
        <AxesSubplot:title={'center':'college_degree'}>,
        <AxesSubplot:title={'center':'honors'}>],
       [<AxesSubplot:title={'center':'worked_during_school'}>,
        <AxesSubplot:title={'center':'years_experience'}>,
        <AxesSubplot:title={'center':'computer_skills'}>,
        <AxesSubplot:title={'center':'special_skills'}>],
       [<AxesSubplot:title={'center':'volunteer'}>,
        <AxesSubplot:title={'center':'military'}>,
        <AxesSubplot:title={'center':'employment_holes'}>,
        <AxesSubplot:title={'center':'has_email_address'}>]], dtype=objec
t)
```



check the distribution of numerical_columns

In [36]:

```
for i in num_feature:
    sns.distplot(df[i])
    plt.show()
```



In [37]:

```
# Categorical Columns
cat_feature = [feature for feature in df.columns if df[feature].dtypes=='object']
print("Number of Categorical Feature" , len(cat_feature))
print(cat_feature)
```

```
Number of Categorical Feature 9
['job_city', 'job_industry', 'job_type', 'job_ownership', 'job_req_schoo
l', 'firstname', 'race', 'gender', 'resume_quality']
```

In [38]:

```
# Fetch the Unique value of categorical data

for i in cat_feature:
    print(i , df[i].unique())
    print()
```

```
job_city ['Chicago' 'Boston']

job_industry ['manufacturing' 'other_service' 'wholesale_and_retail_trade'
 'business_and_personal_service' 'finance_insurance_real_estate'
 'transportation_communication']

job_type ['supervisor' 'secretary' 'sales_rep' 'retail_sales' 'manager' 'c
lerical']

job_ownership ['unknown' 'nonprofit' 'private' 'public']

job_req_school ['none_listed' 'some_college' 'college' 'high_school_grad']

firstname ['Allison' 'Kristen' 'Lakisha' 'Latonya' 'Carrie' 'Jay' 'Jill'
'Kenya'
 'Tyrone' 'Aisha' 'Geoffrey' 'Matthew' 'Tamika' 'Leroy' 'Todd' 'Greg'
 'Keisha' 'Brad' 'Laurie' 'Meredith' 'Anne' 'Emily' 'Latoya' 'Ebony'
 'Brendan' 'Hakim' 'Jamal' 'Neil' 'Tremayne' 'Brett' 'Darnell' 'Sarah'
 'Jermaine' 'Tanisha' 'Rasheed' 'Kareem']

race ['white' 'black']

gender ['f' 'm']

resume_quality ['low' 'high']
```

In [39]:

```
# fetch the only categorical columns
cat_df=df[cat_feature]
```

In [40]:

```
cat_df
```

Out[40]:

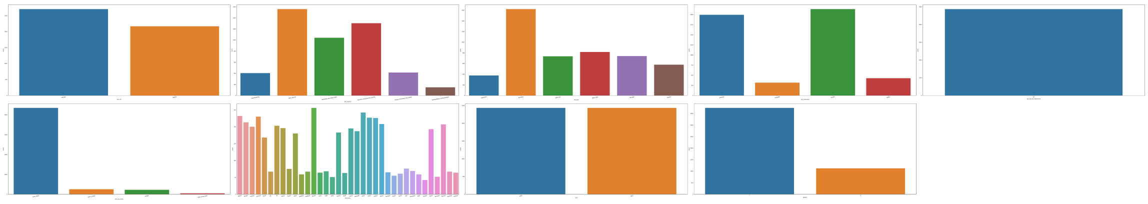
	job_city	job_industry	job_type	job_ownership	job_req_school	firstna
0	Chicago	manufacturing	supervisor	unknown	none_listed	Alli
1	Chicago	manufacturing	supervisor	unknown	none_listed	Kris
2	Chicago	manufacturing	supervisor	unknown	none_listed	Laki
3	Chicago	manufacturing	supervisor	unknown	none_listed	Lato
4	Chicago	other_service	secretary	nonprofit	none_listed	Ca
...	...	...	...	...	...	
4865	Boston	finance_insurance_real_estate	secretary	private	none_listed	Tan
4866	Boston	other_service	manager	unknown	none_listed	Eb
4867	Boston	other_service	manager	unknown	none_listed	
4868	Boston	other_service	manager	unknown	none_listed	Lato
4869	Boston	other_service	manager	unknown	none_listed	La

4870 rows × 9 columns

Data Visulization

In [41]:

```
plt.figure(figsize=(140,120))
#sns.set(rc={'figure.figsize':(15,6)})
cat_feature=['job_city','job_industry','job_type' ,'job_ownership','job_req_min_experien
for i in range(0 , len(cat_feature)):
    plt.subplot(10,5,i+1)
    sns.countplot(x=df[cat_feature[i]])
    plt.xlabel(cat_feature[i])
    plt.xticks(rotation=10)
    plt.tight_layout()
```



In [42]:

```
df['job_fed_contractor'].unique()
```

Out[42]:

```
array([0.11476467, 0.        , 1.        ])
```

In [43]:

```
df['job_fed_contractor'].value_counts()
```

Out[43]:

```
0.000000    2746
0.114765    1768
1.000000     356
Name: job_fed_contractor, dtype: int64
```

In [44]:

```
df['job_req_communication'].unique()
```

Out[44]:

```
array([0, 1], dtype=int64)
```

In [45]:

```
df['job_req_communication'].value_counts()
```

Out[45]:

```
0    4262
1     608
Name: job_req_communication, dtype: int64
```

In [46]:

```
df['computer_skills'].unique()
```

Out[46]:

```
array([1, 0], dtype=int64)
```

In [47]:

```
df['computer_skills'].value_counts()
```

Out[47]:

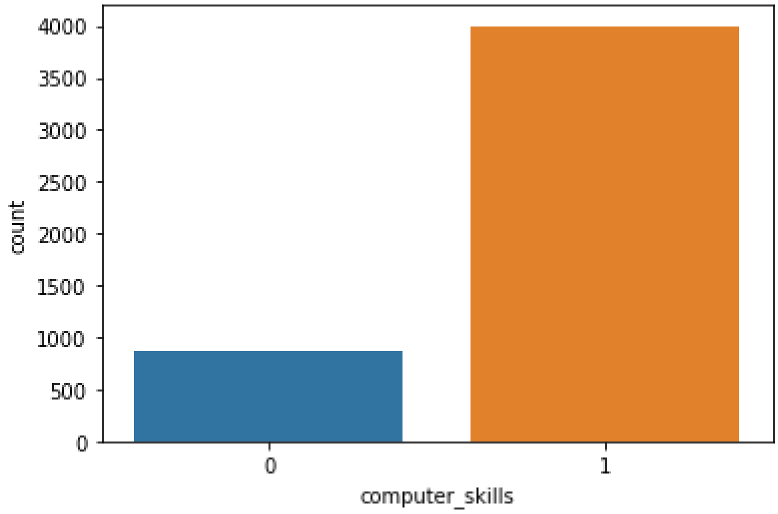
```
1    3996
0     874
Name: computer_skills, dtype: int64
```

In [48]:

```
sns.countplot(df['computer_skills'])
```

Out[48]:

```
<AxesSubplot:xlabel='computer_skills', ylabel='count'>
```



In [49]:

```
df['job_type'].unique()
```

Out[49]:

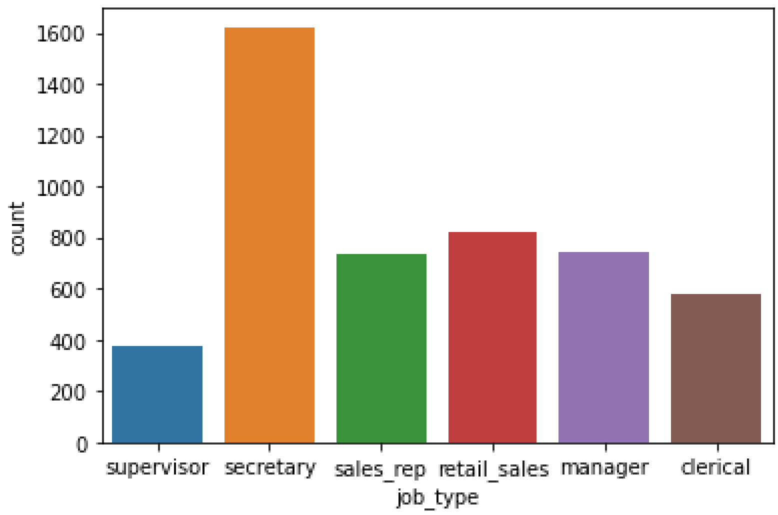
```
array(['supervisor', 'secretary', 'sales_rep', 'retail_sales', 'manager',
       'clerical'], dtype=object)
```

In [50]:

```
sns.countplot(df['job_type'])
```

Out[50]:

```
<AxesSubplot:xlabel='job_type', ylabel='count'>
```

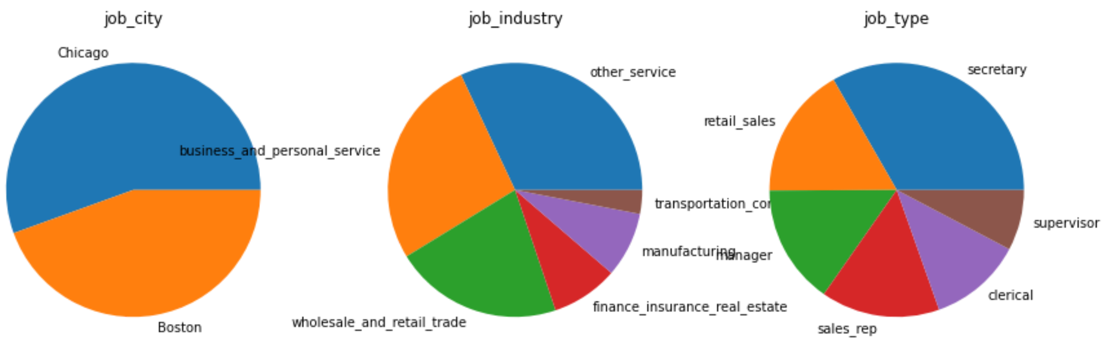


In [51]:

```
# job_city , job_industry , job_ytpes

# job_city
plt.figure(figsize=(15,12))
plt.subplot(1,3,1)
labels = df['job_city'].value_counts().index  # index is used for categorical_columns
size=df['job_city'].value_counts()
explode=None
plt.pie(size,labels=labels,explode=explode)
plt.title('job_city')

# job_industries
plt.subplot(1,3,2)
labels=df['job_industry'].value_counts().index
size=df['job_industry'].value_counts()
explode=None
plt.pie(size,labels=labels,explode=explode)
plt.title('job_industry')

# job_types
plt.subplot(1,3,3)
labels=df['job_type'].value_counts().index
size=df['job_type'].value_counts()
explode=None
plt.pie(size,labels=labels,explode=explode)
plt.title('job_type')
```

Out[51]:

```
Text(0.5, 1.0, 'job_type')
```



In [52]:

```
df.columns
```

Out[52]:

```
Index(['job_city', 'job_industry', 'job_type', 'job_fed_contractor',
       'job_equal_opp_employer', 'job_ownership', 'job_req_any',
       'job_req_communication', 'job_req_education', 'job_req_min_experien
ce',
       'job_req_computer', 'job_req_organization', 'job_req_school',
       'received_callback', 'firstname', 'race', 'gender', 'years_colleg
e',
       'college_degree', 'honors', 'worked_during_school', 'years_experien
ce',
       'computer_skills', 'special_skills', 'volunteer', 'military',
       'employment_holes', 'has_email_address', 'resume_quality'],
      dtype='object')
```

In [53]:

```
# create a dataframe race and total race data
race_data=df['race'].value_counts().reset_index()
race_data.columns=['race','Total_data']
race_data
```

Out[53]:

	race	Total_data
0	white	2435
1	black	2435

In [54]:

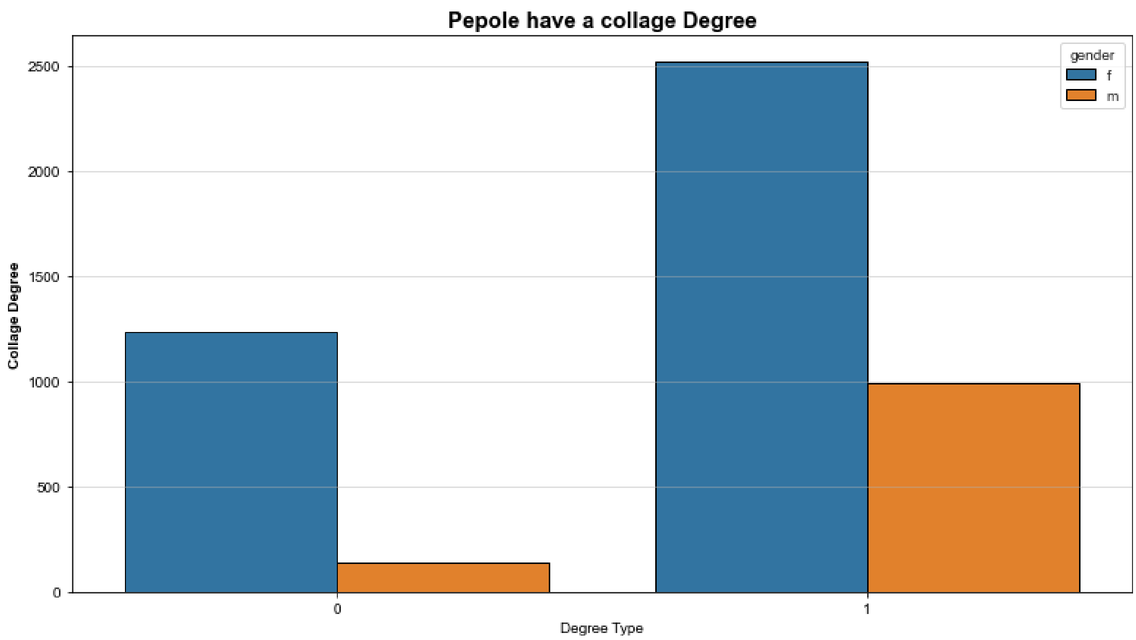
```
#
df.groupby(['gender'])['college_degree'].value_counts()
```

Out[54]:

```
gender  college_degree
f       1                 2515
        0                 1231
m       1                  989
        0                  135
Name: college_degree, dtype: int64
```

In [55]:

```
# plot barchart
plt.subplots(figsize=(13,7))
sns.set_style('whitegrid')
sns.countplot(x='college_degree',hue='gender',data=df,ec='black')
plt.title('Pepole have a collage Degree' ,weight='bold',fontsize=15 )
plt.ylabel('Collage Degree',weight='bold')
plt.xlabel('Degree Type')
plt.grid(alpha=0.5,axis='y')
plt.show()
```



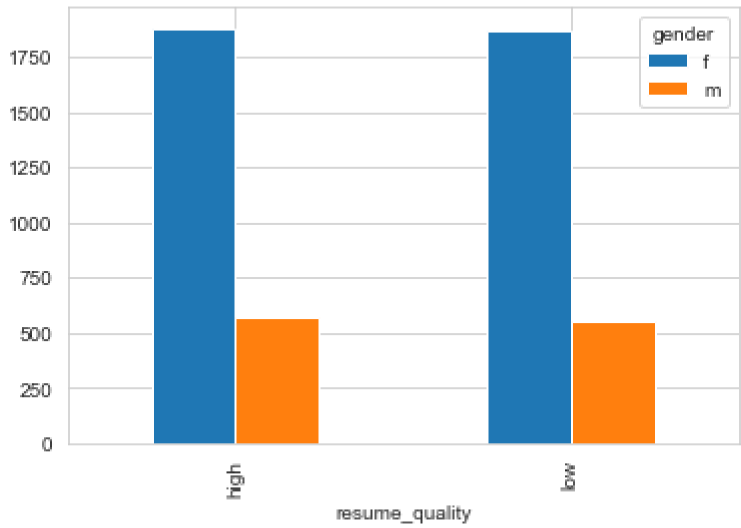
In [56]:

```
# check the resume quality
plt.figure(figsize=(15,12))
resume=pd.crosstab(df['resume_quality'],df['gender'])
resume.plot(kind='bar')
```

Out[56]:

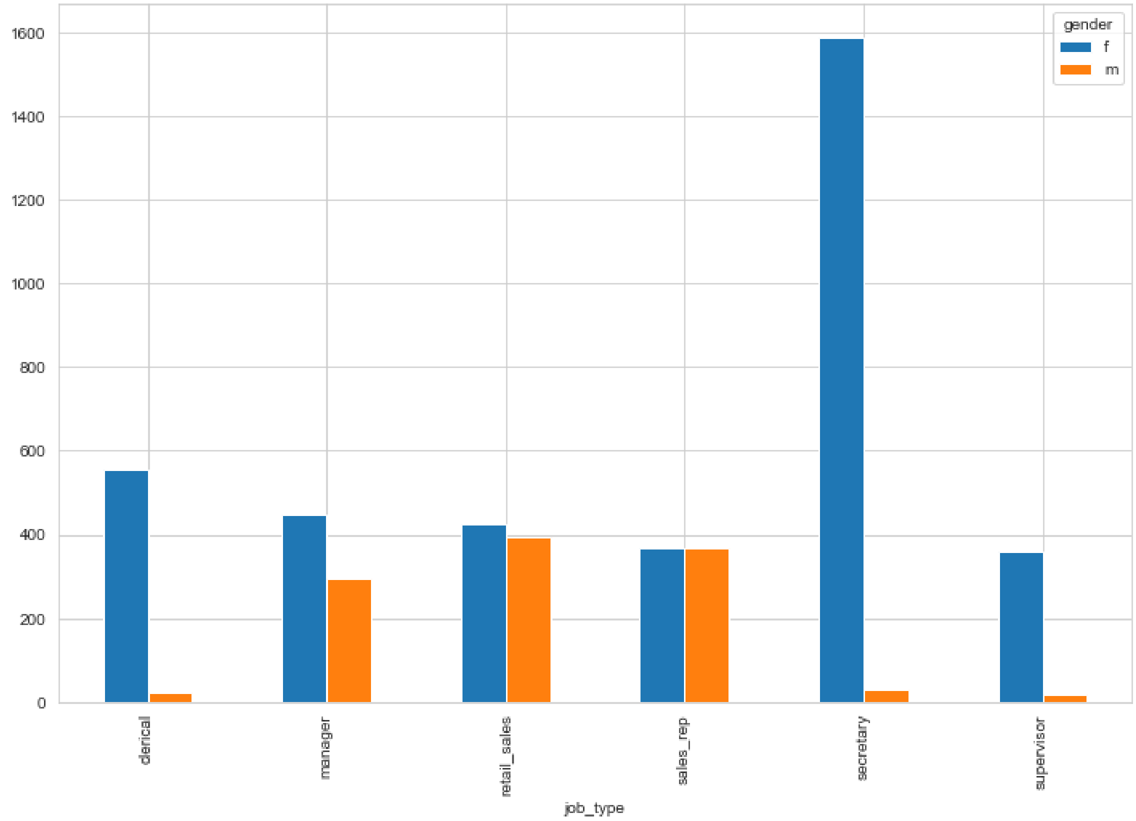
<AxesSubplot:xlabel='resume_quality'>

<Figure size 1080x864 with 0 Axes>



In [57]:

```
# check the gender for jobs types in categories
job_type_gender=pd.crosstab(df['job_type'],df['gender'])
job_type_gender.plot(kind='bar',figsize=(12,8))
plt.show()
```



In [ ]:

```

```

Resume quality-- low or high

In [58]:

```
percentage=df.resume_quality.value_counts(normalize=True)*100
pielabels=['high','low']
```

In [59]:

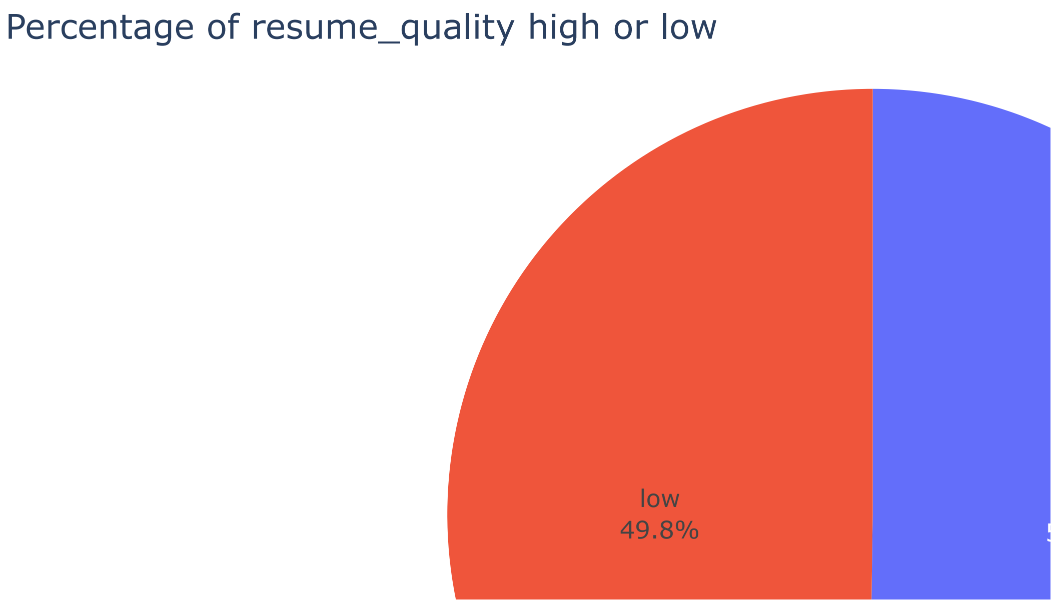
```
df['resume_quality'].unique
```

Out[59]:

```
<bound method Series.unique of 0        low
1       high
2        low
3       high
4       high
        ...
4865     low
4866     low
4867    high
4868    high
4869     low
Name: resume_quality, Length: 4870, dtype: object>
```

In [60]:

```
# plot pie chart with PLotly library

f1 = px.pie(values=percentage , names=pielabels , title='Percentage of resume_quality hi
f1.update_traces(textposition='inside',textinfo='percent+label')
f1.update_layout(margin={'r':50,'t':50 , 'b':50})
f1.show()
```



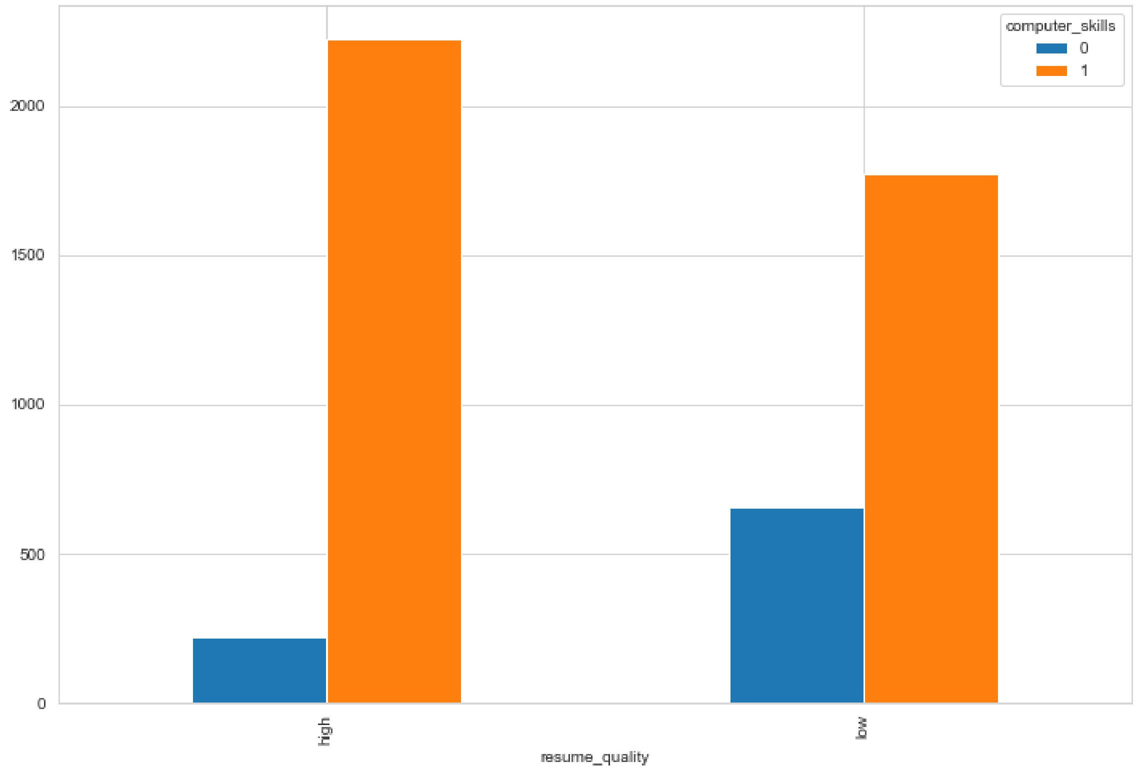
- it is indicated the resume_quality is low value is 49.8% and high quality is 50.2%

In [61]:

```
# ckeck the resume_quality for computer_skills
computer_skills = pd.crosstab(df['resume_quality'],df['computer_skills'])
computer_skills.plot(kind="bar",figsize=(12,8))
```

Out[61]:

```
<AxesSubplot:xlabel='resume_quality'>
```



In [62]:

```
df['job_industry'].unique()
```

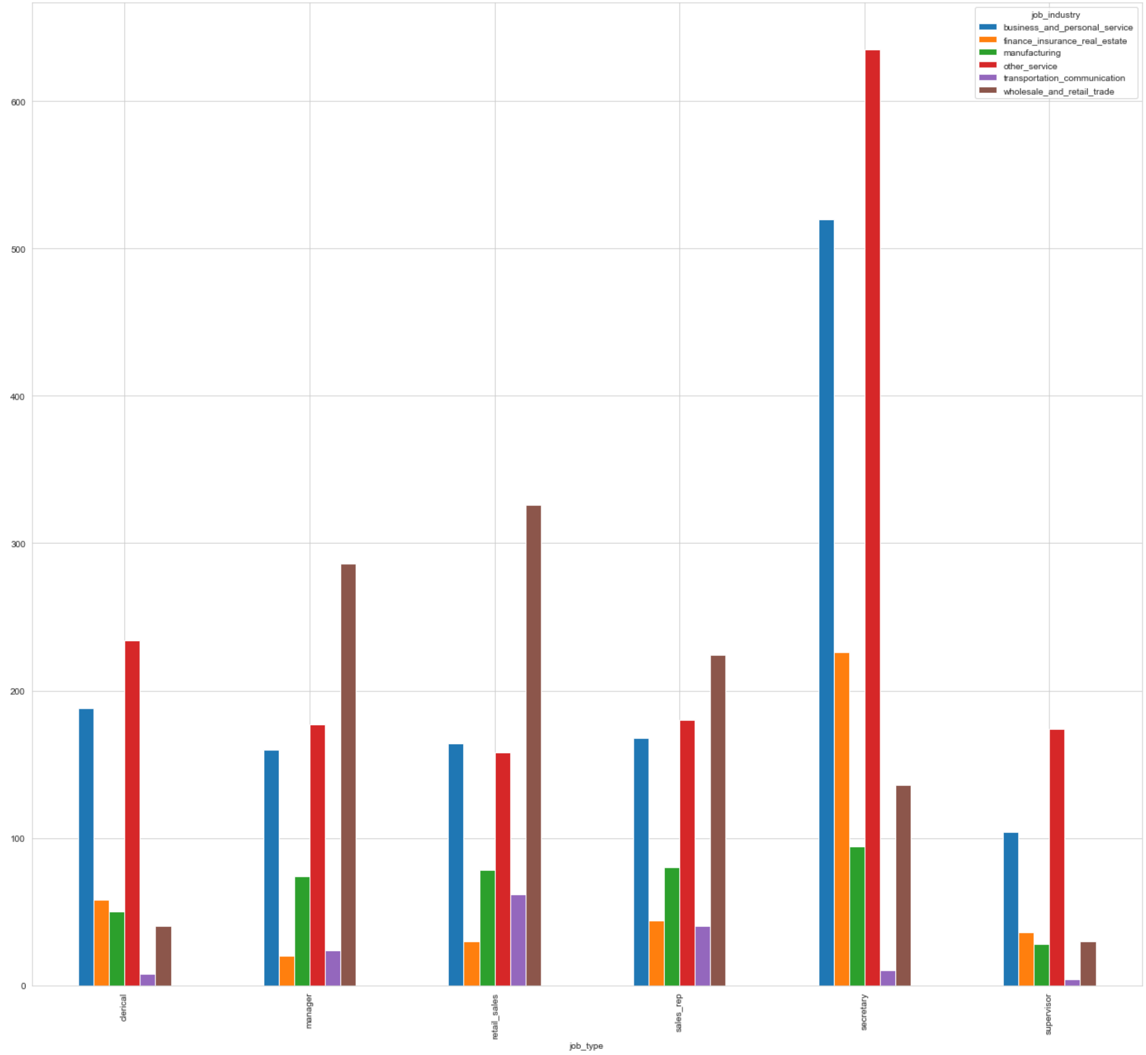
Out[62]:

```
array(['manufacturing', 'other_service', 'wholesale_and_retail_trade',
       'business_and_personal_service', 'finance_insurance_real_estate',
       'transportation_communication'], dtype=object)
```

which job industries has most job types?

In [63]:

```
most_job = pd.crosstab(df['job_type'],df['job_industry'])
most_job.plot(kind='bar',figsize=(22,20))
plt.show()
```



- it is indicated the too much demand for secretary in job_types and other_servies is job_industry
- low demand for supervisor job_types and the transportation_communication

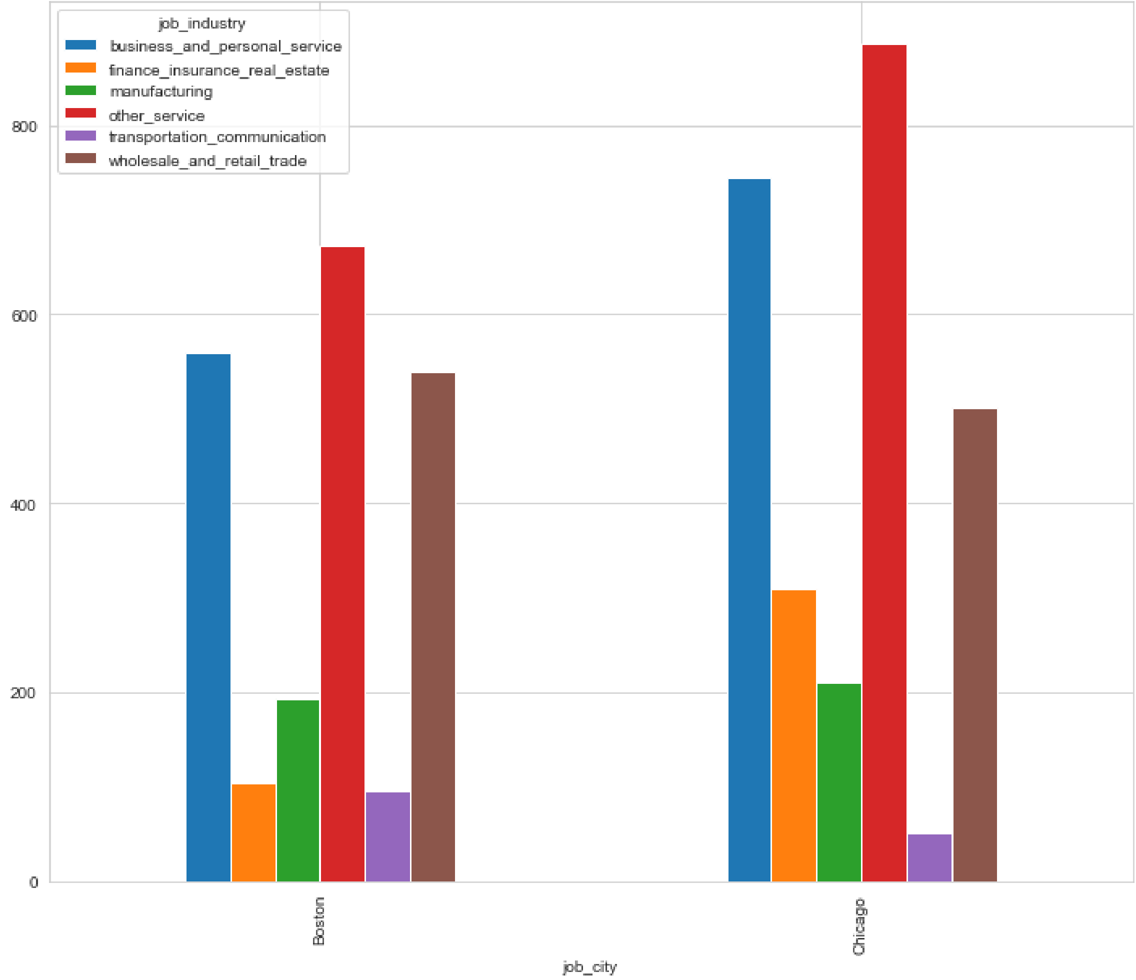
which city has most job industries?

In [64]:

```
most_job_city = pd.crosstab(df['job_city'],df['job_industry'])
most_job_city.plot(kind='bar',figsize=(12,10))
```

Out[64]:

```
<AxesSubplot:xlabel='job_city'>
```



- it is indicated the too much demand for Chicago job city is most other-service is job industries..
- low job demand for the Chicago job city in transportation-communication job industrie

In [65]:

```
df['job_ownership'].unique()
```

Out[65]:

```
array(['unknown', 'nonprofit', 'private', 'public'], dtype=object)
```

In [66]:

```
df['job_req_education'].unique()
```

Out[66]:

```
array([0, 1], dtype=int64)
```

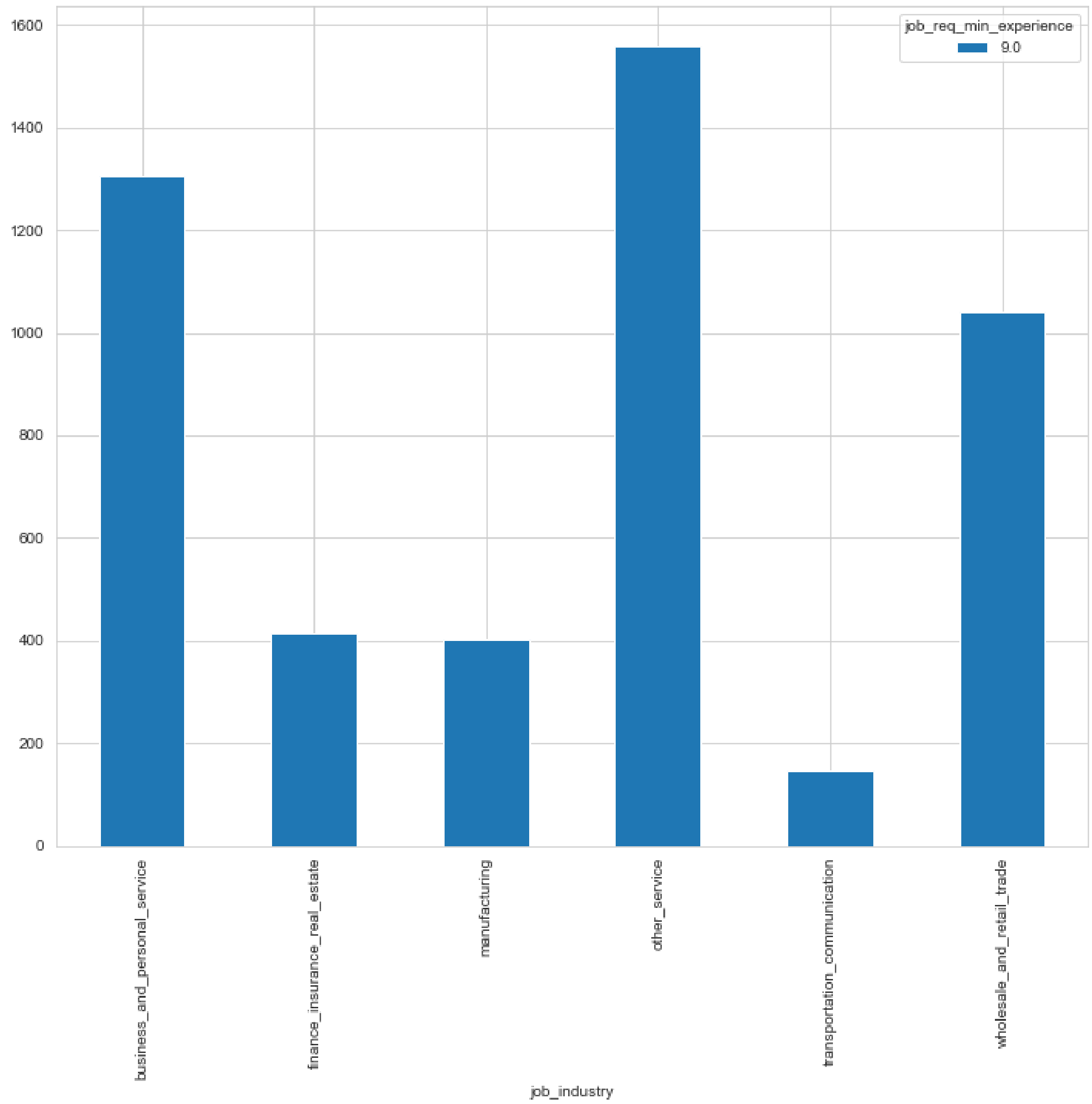
which job industries has required minimum job experience?

In [67]:

```
min_job_required = pd.crosstab(df['job_industry'],df['job_req_min_experience'])
min_job_required.plot(kind='bar',figsize=(12,10))
```

Out[67]:

```
<AxesSubplot:xlabel='job_industry'>
```



- we are observed finance_insurance_real_estate sector has 0 year of required experience

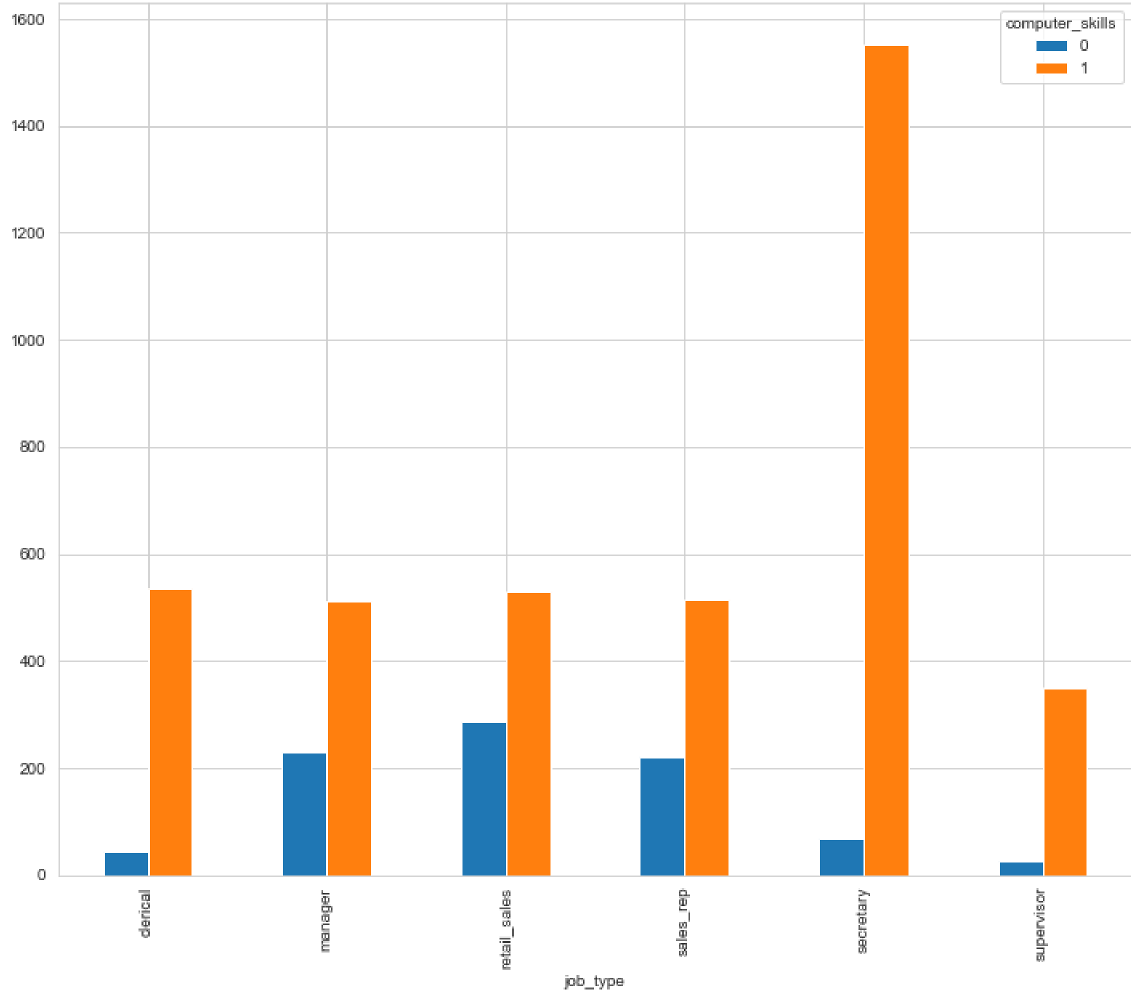
which job type has required the computer skilled

In [68]:

```
computer_skill = pd.crosstab(df['job_type'],df['computer_skills'])
computer_skill.plot(kind='bar',figsize=(12,10))
```

Out[68]:

```
<AxesSubplot:xlabel='job_type'>
```



which person firstname belong to the gender categories ?

In [69]:

```
df['firstname'].unique()
```

Out[69]:

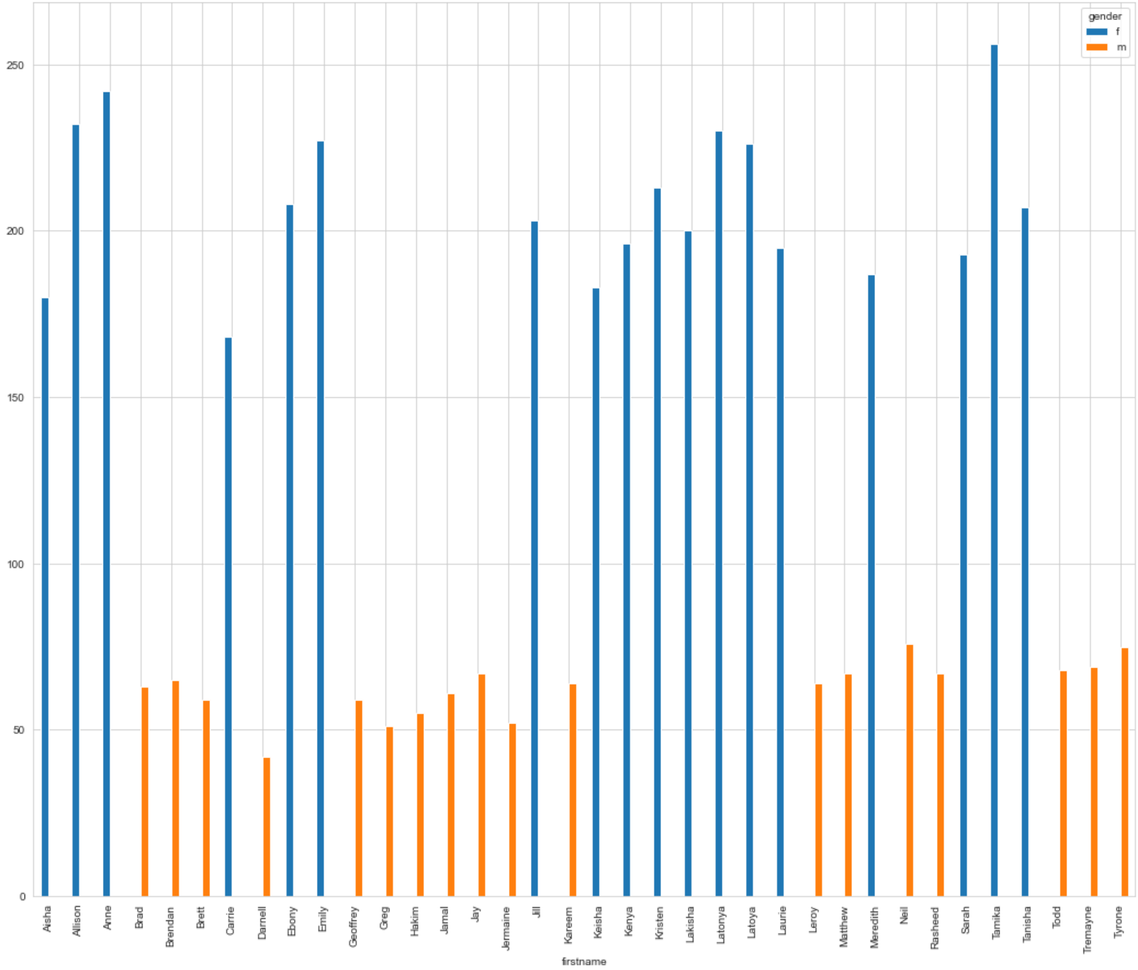
```
array(['Allison', 'Kristen', 'Lakisha', 'Latonya', 'Carrie', 'Jay',
       'Jill', 'Kenya', 'Tyrone', 'Aisha', 'Geoffrey', 'Matthew',
       'Tamika', 'Leroy', 'Todd', 'Greg', 'Keisha', 'Brad', 'Laurie',
       'Meredith', 'Anne', 'Emily', 'Latoya', 'Ebony', 'Brendan', 'Hakim',
       'Jamal', 'Neil', 'Tremayne', 'Brett', 'Darnell', 'Sarah',
       'Jermaine', 'Tanisha', 'Rasheed', 'Kareem'], dtype=object)
```

In [70]:

```
gender=pd.crosstab(df['firstname'],df['gender'])
gender.plot(kind='bar', figsize=(18,15))
```

Out[70]:

```
<AxesSubplot:xlabel='firstname'>
```



- 18 person firstname is belong to the male and 18 person firstname is belong to the female categories

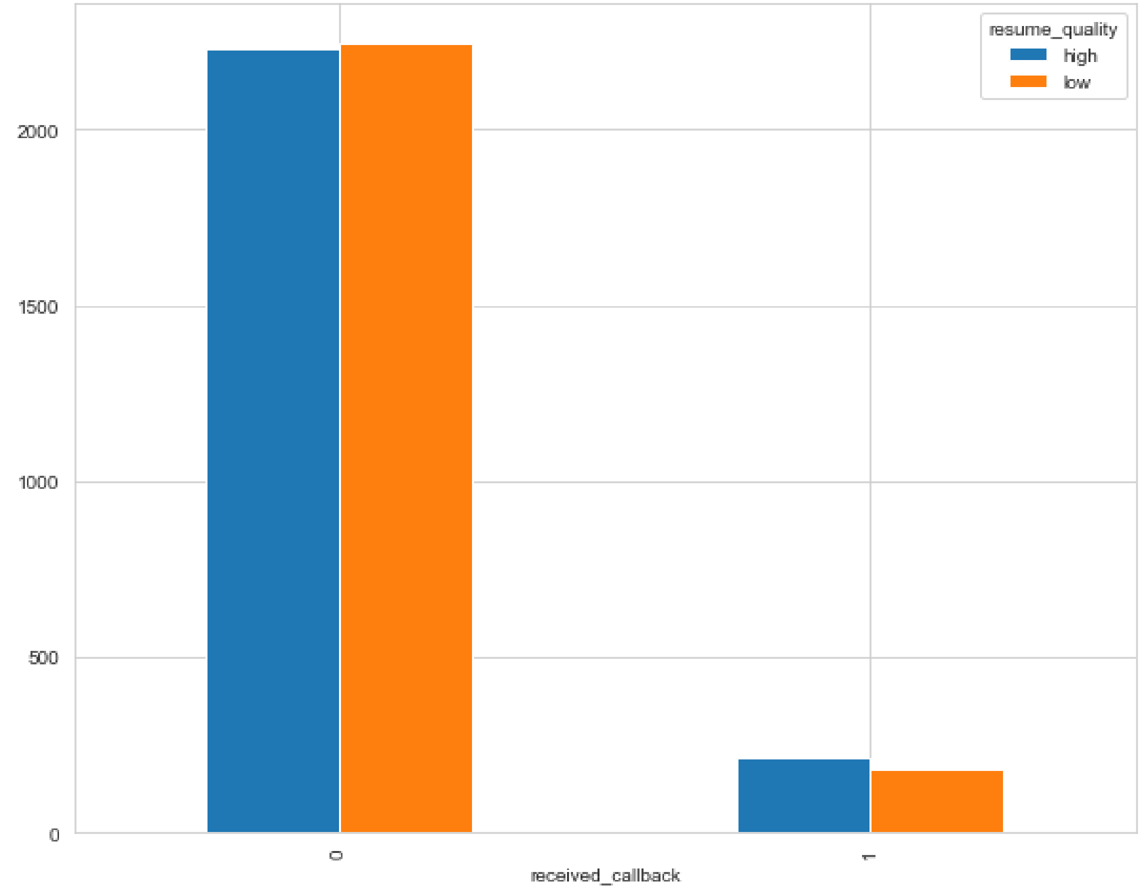
How many person recevie call back during the resume_quality ?

In [71]:

```
recevie_call = pd.crosstab(df['received_callback'],df['resume_quality'])
recevie_call.plot(kind='bar',figsize=(10,8))
```

Out[71]:

```
<AxesSubplot:xlabel='received_callback'>
```



In [ ]:

```
```

what are job industry is each distribution of the gender categories?

In [72]:

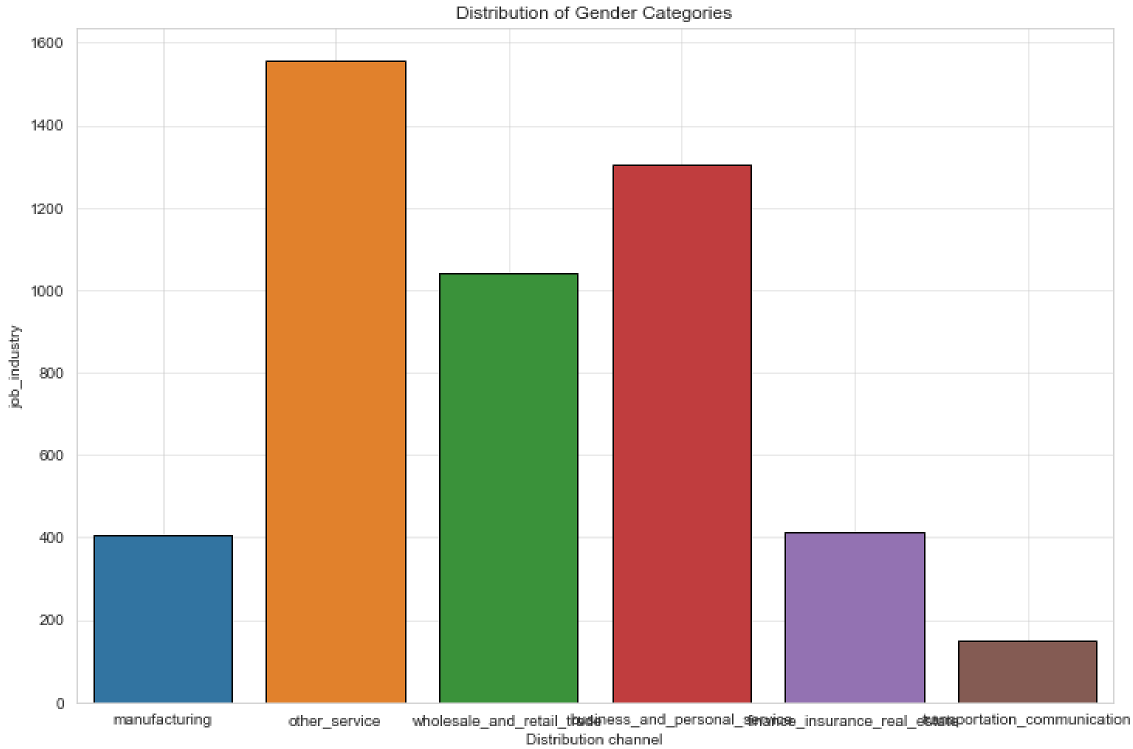
```
df.groupby(['job_industry'])['gender'].value_counts()
```

Out[72]:

```
job_industry                     gender
business_and_personal_service    f         1079
                                 m          225
finance_insurance_real_estate    f          375
                                 m           39
manufacturing                    f          284
                                 m          120
other_service                    f         1320
                                 m          238
transportation_communication     f           91
                                 m           57
wholesale_and_retail_trade       f          597
                                 m          445
Name: gender, dtype: int64
```

In [73]:

```
plt.subplots(figsize=(12,8))
sns.countplot(x='job_industry',data=df,ec='black')
plt.title('Distribution of Gender Categories')
plt.xlabel('Distribution channel')
plt.ylabel('job_industry')
plt.grid(alpha=0.5)
```



In [74]:

```
cat_feature
```

Out[74]:

```
['job_city',
 'job_industry',
 'job_type',
 'job_ownership',
 'job_req_min_experience',
 'job_req_school',
 'firstname',
 'race',
 'gender']
```

In [75]:

```
df['job_req_school'].unique()
```

Out[75]:

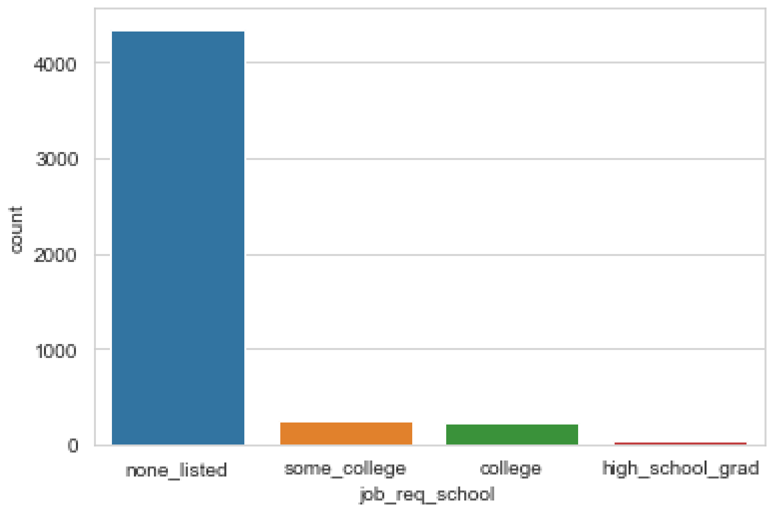
```
array(['none_listed', 'some_college', 'college', 'high_school_grad'],
      dtype=object)
```

In [76]:

```
sns.countplot(df['job_req_school'])
```

Out[76]:

```
<AxesSubplot:xlabel='job_req_school', ylabel='count'>
```



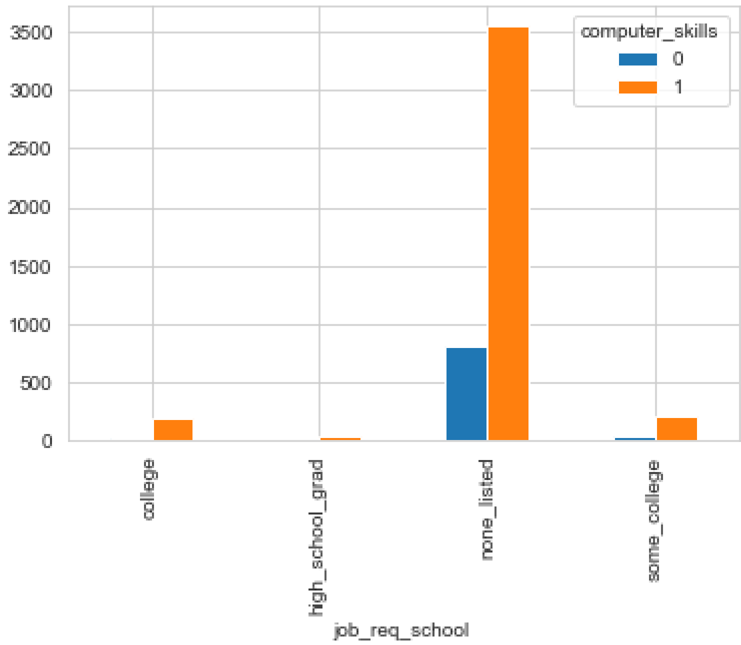
which job required school is needed for the computer school?

In [77]:

```
job_req_school = pd.crosstab(df['job_req_school'],df['computer_skills'])
job_req_school.plot(kind='bar')
```

Out[77]:

```
<AxesSubplot:xlabel='job_req_school'>
```



In [78]:

```
df.columns
```

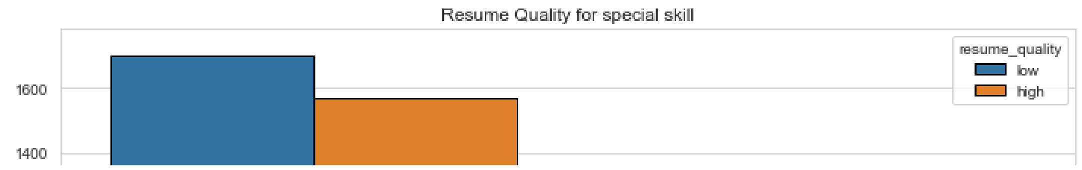
Out[78]:

```
Index(['job_city', 'job_industry', 'job_type', 'job_fed_contractor',
       'job_equal_opp_employer', 'job_ownership', 'job_req_any',
       'job_req_communication', 'job_req_education', 'job_req_min_experien
ce',
       'job_req_computer', 'job_req_organization', 'job_req_school',
       'received_callback', 'firstname', 'race', 'gender', 'years_colleg
e',
       'college_degree', 'honors', 'worked_during_school', 'years_experien
ce',
       'computer_skills', 'special_skills', 'volunteer', 'military',
       'employment_holes', 'has_email_address', 'resume_quality'],
      dtype='object')
```

In [79]:

```
plt.subplots(figsize=(12,7))
sns.countplot(x='special_skills',hue='resume_quality',data=df,ec='black')
plt.title('Resume Quality for special skill' , fontsize='bold',weight='bold')
plt.ylabel('')
plt.xlabel('special skills')
plt.legend(loc='upper right')
plt.grid(alpha=0.5)
```

```
File ~\anaconda3\lib\site-packages\matplotlib\font_manager.py:876, in F
ontProperties.set_size(self, size)
    874     scale = font_scalings[size]
    875 except KeyError as err:
--> 876     raise ValueError(
    877         "Size is invalid. Valid font size are "
    878         + ", ".join(map(str, font_scalings))) from err
    879 else:
    880     size = scale * FontManager.get_default_size()

ValueError: Size is invalid. Valid font size are xx-small, x-small, sma
ll, medium, large, x-large, xx-large, larger, smaller, None
```



In [80]:

```
df['years_experience']
```

Out[80]:

```
0        6
1        6
2        6
3        6
4       22
        ..
4865     1
4866     6
4867     8
4868     2
4869     7
Name: years_experience, Length: 4870, dtype: int64
```

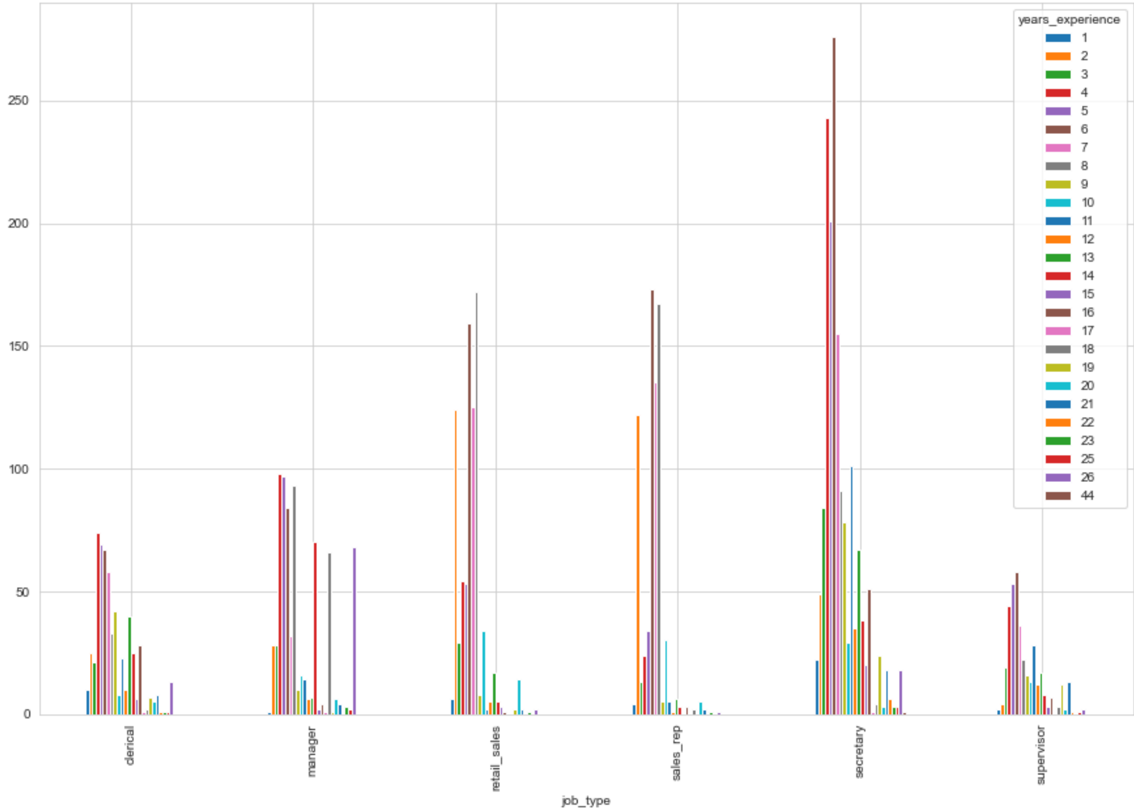
Check the total year of experience in job types industry

In [81]:

```
year_of_exper = pd.crosstab(df['job_type'],df['years_experience'])
year_of_exper.plot(kind='bar',figsize=(15,10))
```

Out[81]:

```
<AxesSubplot:xlabel='job_type'>
```



- Highest year of experience is 44 year and lowest year of experience is 1 year
- Secretary job type have highest year of experience in 44 year

In [82]:

```
cat_feature
```

Out[82]:

```
['job_city',
 'job_industry',
 'job_type',
 'job_ownership',
 'job_req_min_experience',
 'job_req_school',
 'firstname',
 'race',
 'gender']
```

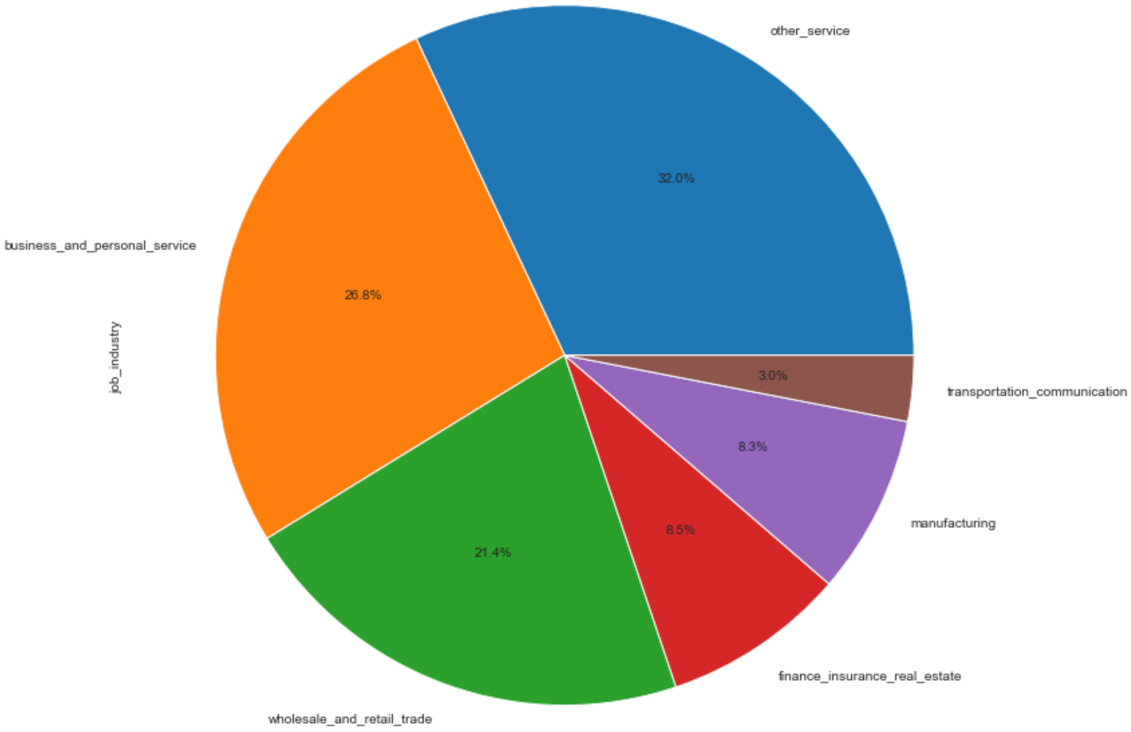
which is one of the most popular job industry

In [83]:

```
df['job_industry'].value_counts().plot.pie(figsize=(15,12),autopct="%1.1f%%")
```

Out[83]:

```
<AxesSubplot:ylabel='job_industry'>
```



- it is indicated the other_servies job industry has more job requiremnets-- 32 % is need

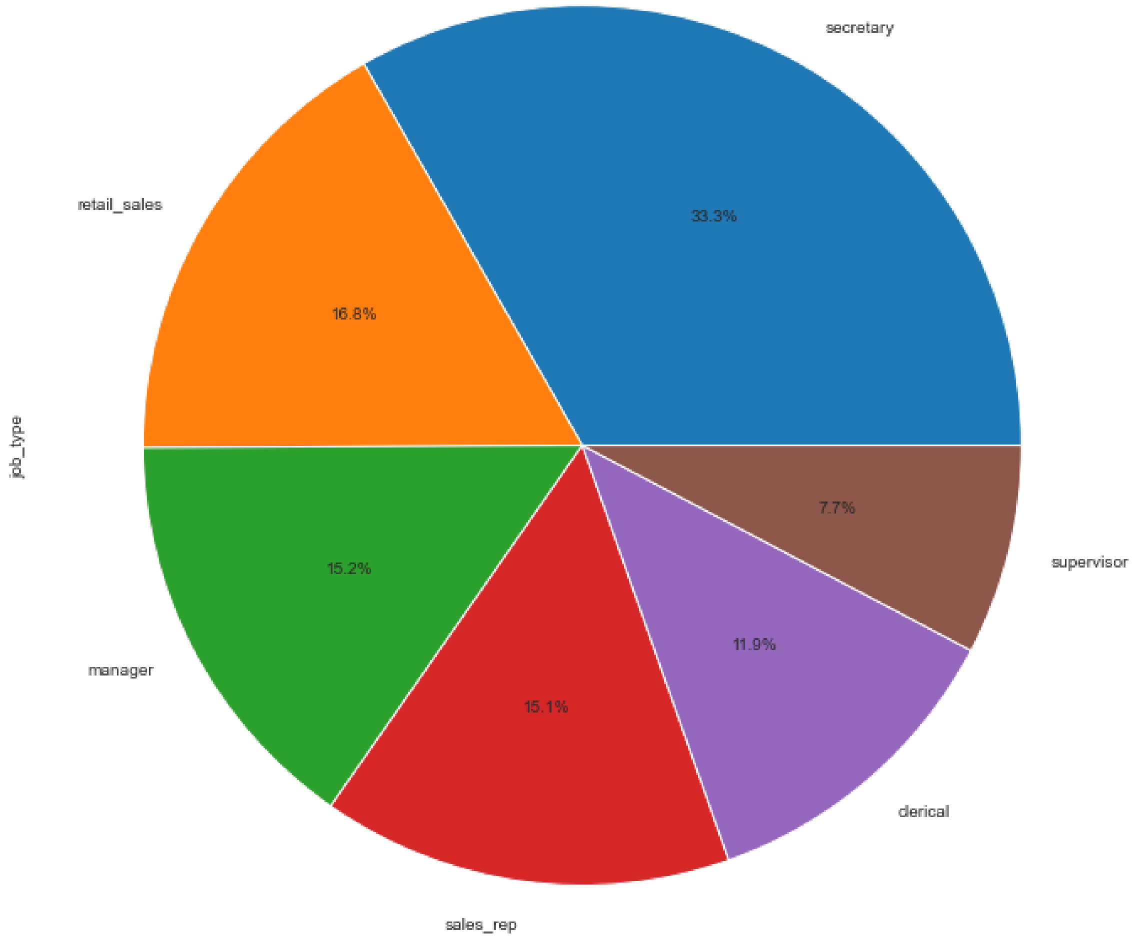
which is one of the most popular job types

In [84]:

```
df['job_type'].value_counts().plot.pie(figsize=(15,12),autopct="%1.1f%%")
```

Out[84]:

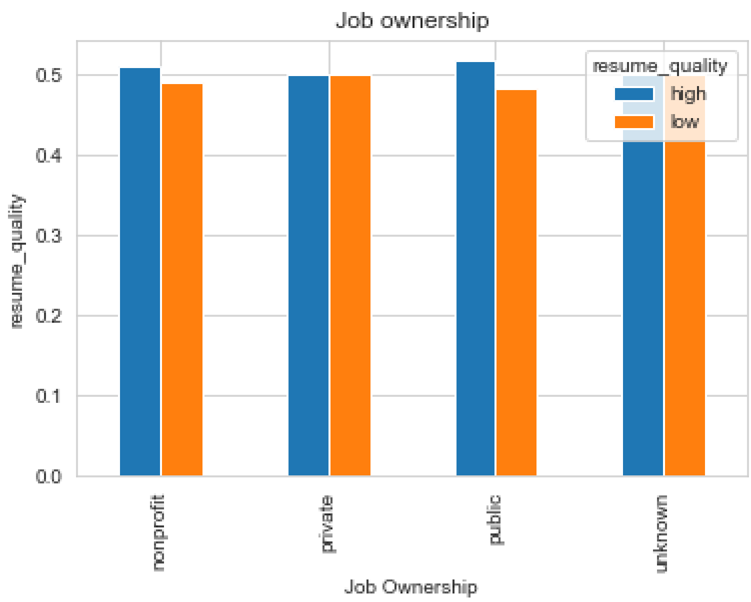
```
<AxesSubplot:ylabel='job_type'>
```



In [85]:

```
# job_ownership  , resume_quality
plt.figure(figsize=(18,15))
x=pd.crosstab(df['job_ownership'],df['resume_quality'])
x.div(x.sum(1).astype(float),axis=0).plot(kind='bar')
plt.title(' Job ownership ')
plt.xlabel('Job Ownership')
plt.ylabel('resume_quality')
plt.show()
```

<Figure size 1296x1080 with 0 Axes>



In [86]:

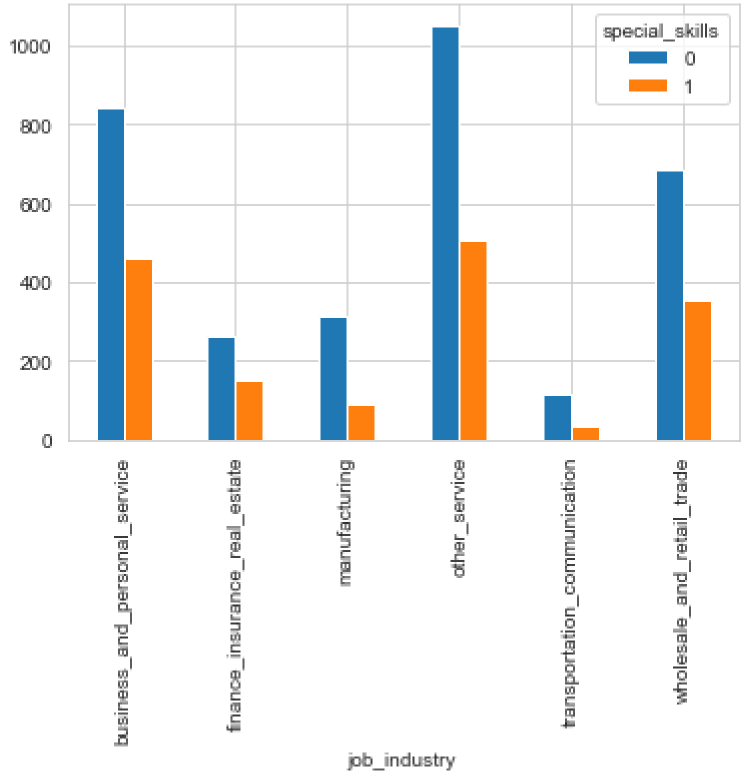
```
# job city , military

sns.barplot(df['job_city'],df['military'])
plt.title('')
plt.xlabel('Job City')
plt.ylabel('military')
plt.show()
```

In [87]:

```
# which job industry is required special skills

job_industry = pd.crosstab(df['job_industry'],df['special_skills'])
job_industry.plot(kind='bar')
```

Out[87]:

```
<AxesSubplot:xlabel='job_industry'>
```



In [88]:

```
df.columns
```

Out[88]:

```
Index(['job_city', 'job_industry', 'job_type', 'job_fed_contractor',
       'job_equal_opp_employer', 'job_ownership', 'job_req_any',
       'job_req_communication', 'job_req_education', 'job_req_min_experien
ce',
       'job_req_computer', 'job_req_organization', 'job_req_school',
       'received_callback', 'firstname', 'race', 'gender', 'years_colleg
e',
       'college_degree', 'honors', 'worked_during_school', 'years_experien
ce',
       'computer_skills', 'special_skills', 'volunteer', 'military',
       'employment_holes', 'has_email_address', 'resume_quality'],
      dtype='object')
```

In [89]:

```
df['job_req_school'].unique()
```

Out[89]:

```
array(['none_listed', 'some_college', 'college', 'high_school_grad'],
      dtype=object)
```

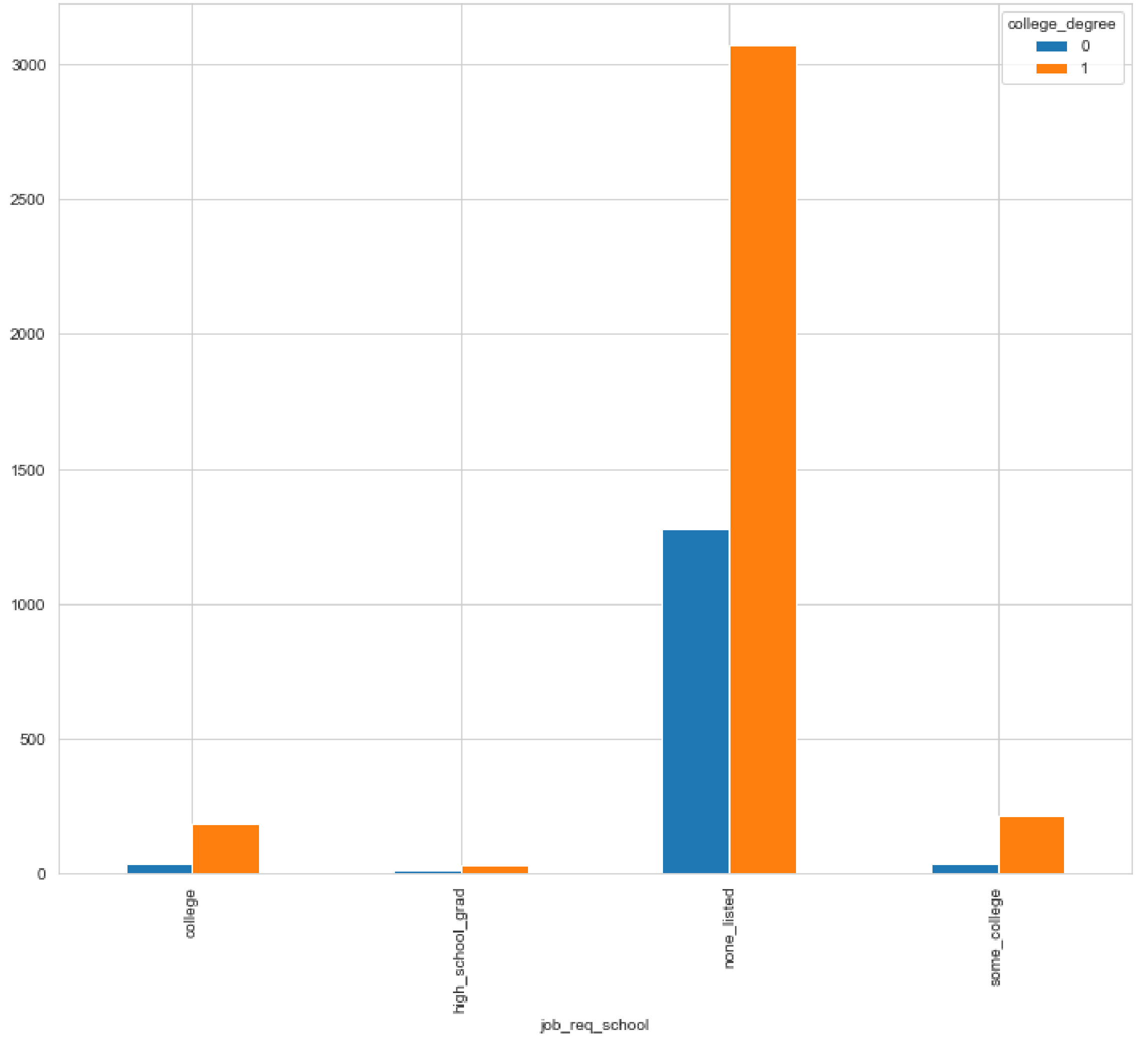
which job_req_school is need for college degree

In [90]:

```
job_req_school = pd.crosstab(df['job_req_school'], df['college_degree'])
job_req_school.plot(kind='bar', figsize=(12,10))
```

Out[90]:

```
<AxesSubplot:xlabel='job_req_school'>
```

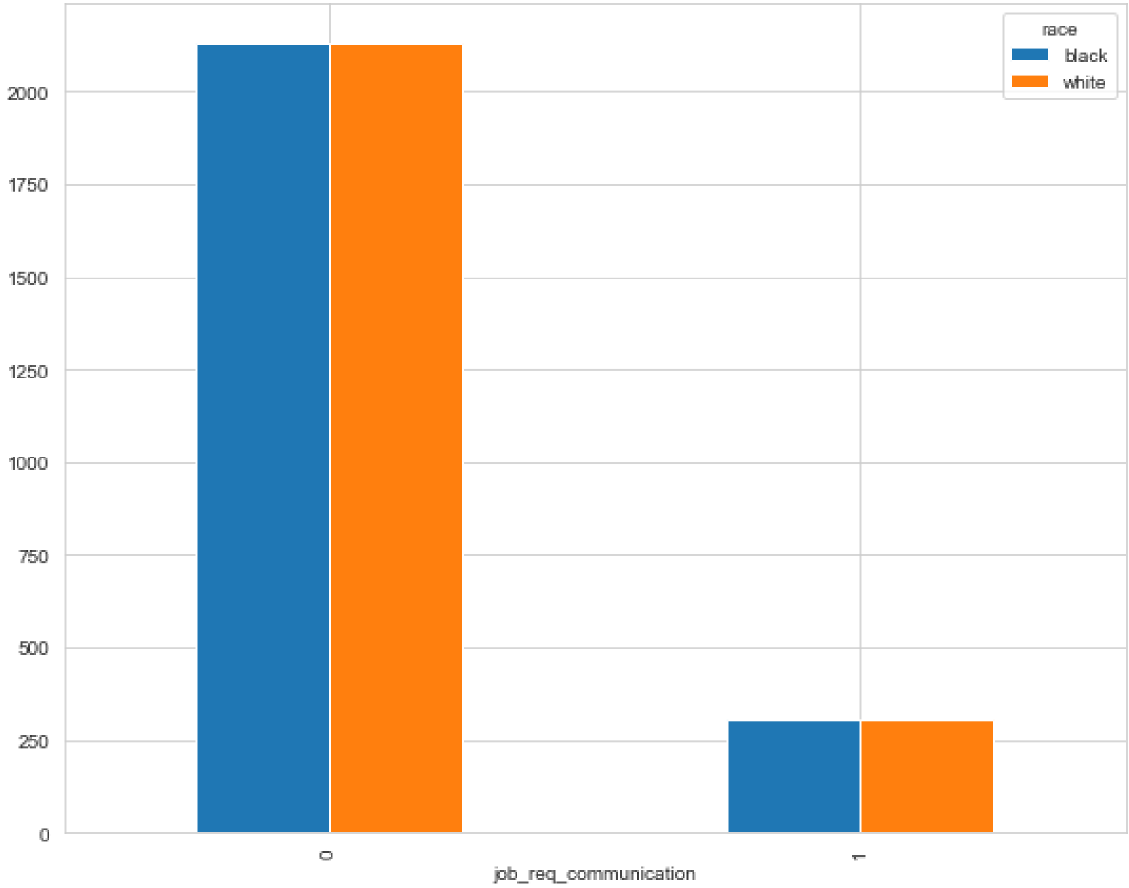


In [91]:

```
# which job_req_communication is need race
job_req_communication = pd.crosstab(df['job_req_communication'], df['race'])
job_req_communication.plot(kind='bar',figsize=(10,8))
```

Out[91]:

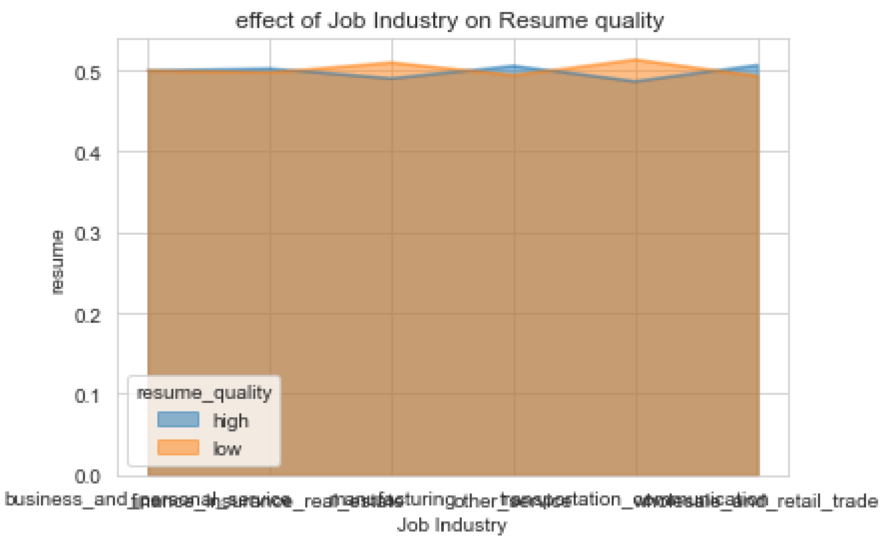
```
<AxesSubplot:xlabel='job_req_communication'>
```



In [92]:

```
# Department and resume_quality in Area chart
plt.figure(figsize=(30,25))
x=pd.crosstab(df['job_industry'],df['resume_quality'])
x.div(x.sum(1).astype(float),axis=0).plot(kind='area',stacked=False)
plt.title('effect of Job Industry on Resume quality ')
plt.xlabel('Job Industry')
plt.ylabel('resume')
plt.show()
```

<Figure size 2160x1800 with 0 Axes>



Target Columns

In [93]:

```
df['resume_quality'].nunique()
```

Out[93]:

```
2
```

In [94]:

```
df['resume_quality'].value_counts()
```

Out[94]:

```
high    2446
low     2424
Name: resume_quality, dtype: int64
```

In [95]:

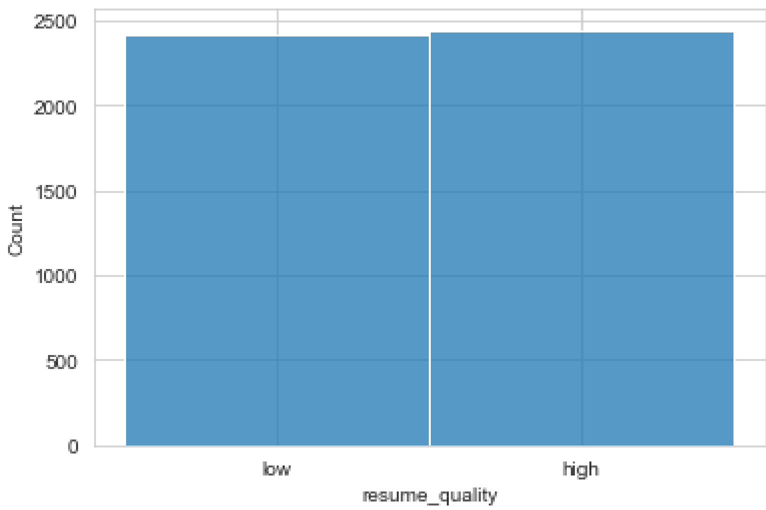
```
# Visulization of target quality

plt.subplots(figsize=(10,5))
sns.histplot(df['resume_quality'],ec='Black',color='blue',kde=True)
plt.title('resume_quality', weight='bold',fontsize=20,pad=20)
plt.ylabel('Count',weight='bold',fontsize=14)
plt.xlabel('resume_quality',weight='bold',fontsize=12)
plt.show()
```

In [96]:

```
sns.histplot(df['resume_quality'])
```

Out[96]:

```
<AxesSubplot:xlabel='resume_quality', ylabel='Count'>
```



Statistical Analysis

In [97]:

```
df.describe()
```

Out[97]:

	job_fed_contractor	job_equal_opp_employer	job_req_any	job_req_communication	jot
count	4870.000000	4870.000000	4870.000000	4870.000000	
mean	0.114765	0.291170	0.787269	0.124846	
std	0.254410	0.454349	0.409281	0.330578	
min	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	1.000000	0.000000	
50%	0.000000	0.000000	1.000000	0.000000	
75%	0.114765	1.000000	1.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	

In [98]:

```
# statistical

df.describe().T
```

Out[98]:

	count	mean	std	min	25%	50%	75%	max
job_fed_contractor	4870.0	0.114765	0.254410	0.0	0.0	0.0	0.114765	1.0
job_equal_opp_employer	4870.0	0.291170	0.454349	0.0	0.0	0.0	1.000000	1.0
job_req_any	4870.0	0.787269	0.409281	0.0	1.0	1.0	1.000000	1.0
job_req_communication	4870.0	0.124846	0.330578	0.0	0.0	0.0	0.000000	1.0
job_req_education	4870.0	0.106776	0.308860	0.0	0.0	0.0	0.000000	1.0
job_req_min_experience	4870.0	9.000000	0.000000	9.0	9.0	9.0	9.000000	9.0
job_req_computer	4870.0	0.437166	0.496087	0.0	0.0	0.0	1.000000	1.0
job_req_organization	4870.0	0.072690	0.259654	0.0	0.0	0.0	0.000000	1.0
received_callback	4870.0	0.080493	0.272083	0.0	0.0	0.0	0.000000	1.0
years_college	4870.0	3.618480	0.714997	0.0	3.0	4.0	4.000000	4.0
college_degree	4870.0	0.719507	0.449286	0.0	0.0	1.0	1.000000	1.0
honors	4870.0	0.052772	0.223601	0.0	0.0	0.0	0.000000	1.0
worked_during_school	4870.0	0.559548	0.496492	0.0	0.0	1.0	1.000000	1.0
years_experience	4870.0	7.842916	5.044612	1.0	5.0	6.0	9.000000	44.0
computer_skills	4870.0	0.820534	0.383782	0.0	1.0	1.0	1.000000	1.0
special_skills	4870.0	0.328747	0.469806	0.0	0.0	0.0	1.000000	1.0
volunteer	4870.0	0.411499	0.492156	0.0	0.0	0.0	1.000000	1.0
military	4870.0	0.097125	0.296159	0.0	0.0	0.0	0.000000	1.0
employment_holes	4870.0	0.448049	0.497345	0.0	0.0	0.0	1.000000	1.0
has_email_address	4870.0	0.479261	0.499621	0.0	0.0	0.0	1.000000	1.0

In [99]:

```
# statistical base on objective categories
df.describe(include='object')
```

Out[99]:

	job_city	job_industry	job_type	job_ownership	job_req_school	firstname	race	ge
count	4870	4870	4870	4870	4870	4870	4870	
unique	2	6	6	4	4	36	2	
top	Chicago	other_service	secretary	private	none_listed	Tamika	white	
freq	2704	1558	1621	2134	4350	256	2435	

In [100]:

```
cat_df = df[cat_feature]
```

In [101]:

```

cat_df
```

Out[101]:

	job_city	job_industry	job_type	job_ownership	job_req_min_experienc
0	Chicago	manufacturing	supervisor	unknown	9.(
1	Chicago	manufacturing	supervisor	unknown	9.(
2	Chicago	manufacturing	supervisor	unknown	9.(
3	Chicago	manufacturing	supervisor	unknown	9.(
4	Chicago	other_service	secretary	nonprofit	9.(
...	...	...	...	...	.
4865	Boston	finance_insurance_real_estate	secretary	private	9.(
4866	Boston	other_service	manager	unknown	9.(
4867	Boston	other_service	manager	unknown	9.(
4868	Boston	other_service	manager	unknown	9.(
4869	Boston	other_service	manager	unknown	9.(

4870 rows × 9 columns

In [102]:

```
# check the standard deviation
df.std()
```

Out[102]:

```
job_fed_contractor         0.254410
job_equal_opp_employer     0.454349
job_req_any                0.409281
job_req_communication      0.330578
job_req_education          0.308860
job_req_min_experience     0.000000
job_req_computer           0.496087
job_req_organization       0.259654
received_callback          0.272083
years_college              0.714997
college_degree             0.449286
honors                     0.223601
worked_during_school       0.496492
years_experience           5.044612
computer_skills            0.383782
special_skills             0.469806
volunteer                  0.492156
military                   0.296159
employment_holes           0.497345
has_email_address          0.499621
dtype: float64
```

In [103]:

```
# check the skewness
df.skew()
```

Out[103]:

```
job_fed_contractor         3.029703
job_equal_opp_employer     0.919626
job_req_any               -1.404351
job_req_communication      2.270617
job_req_education          2.547336
job_req_min_experience     0.000000
job_req_computer           0.253421
job_req_organization       3.292739
received_callback          3.084943
years_college             -2.306128
college_degree            -0.977539
honors                     4.001874
worked_during_school      -0.239974
years_experience           1.685523
computer_skills           -1.671084
special_skills             0.729334
volunteer                  0.359794
military                   2.721786
employment_holes           0.208998
has_email_address          0.083054
dtype: float64
```

In [104]:

```
# check the covariance
df.cov()
```

Out[104]:

	job_fed_contractor	job_equal_opp_employer	job_req_any	job_req_c
job_fed_contractor	0.064725	0.012152	-0.002807	
job_equal_opp_employer	0.012152	0.206433	0.024574	
job_req_any	-0.002807	0.024574	0.167511	
job_req_communication	0.000912	-0.001033	0.026564	
job_req_education	0.004538	0.030518	0.022719	
job_req_min_experience	0.000000	0.000000	0.000000	
job_req_computer	0.000761	0.013781	0.093018	
job_req_organization	-0.001334	-0.001864	0.015467	
received_callback	0.000781	0.000382	-0.004644	
years_college	0.003134	0.023207	0.006931	
college_degree	0.002116	0.017198	-0.000121	
honors	0.000424	-0.002635	0.000138	
worked_during_school	0.001269	0.005044	0.010000	
years_experience	-0.032210	0.044310	0.103566	
computer_skills	-0.001420	0.002153	0.027947	
special_skills	-0.001066	-0.011740	-0.012819	
volunteer	0.001620	0.004620	-0.009794	
military	-0.000570	-0.001792	-0.004801	
employment_holes	-0.001097	-0.019374	0.004966	
has_email_address	0.001482	-0.004229	0.000106	

In [105]:

```
# check the correlation
df.corr()
```

Out[105]:

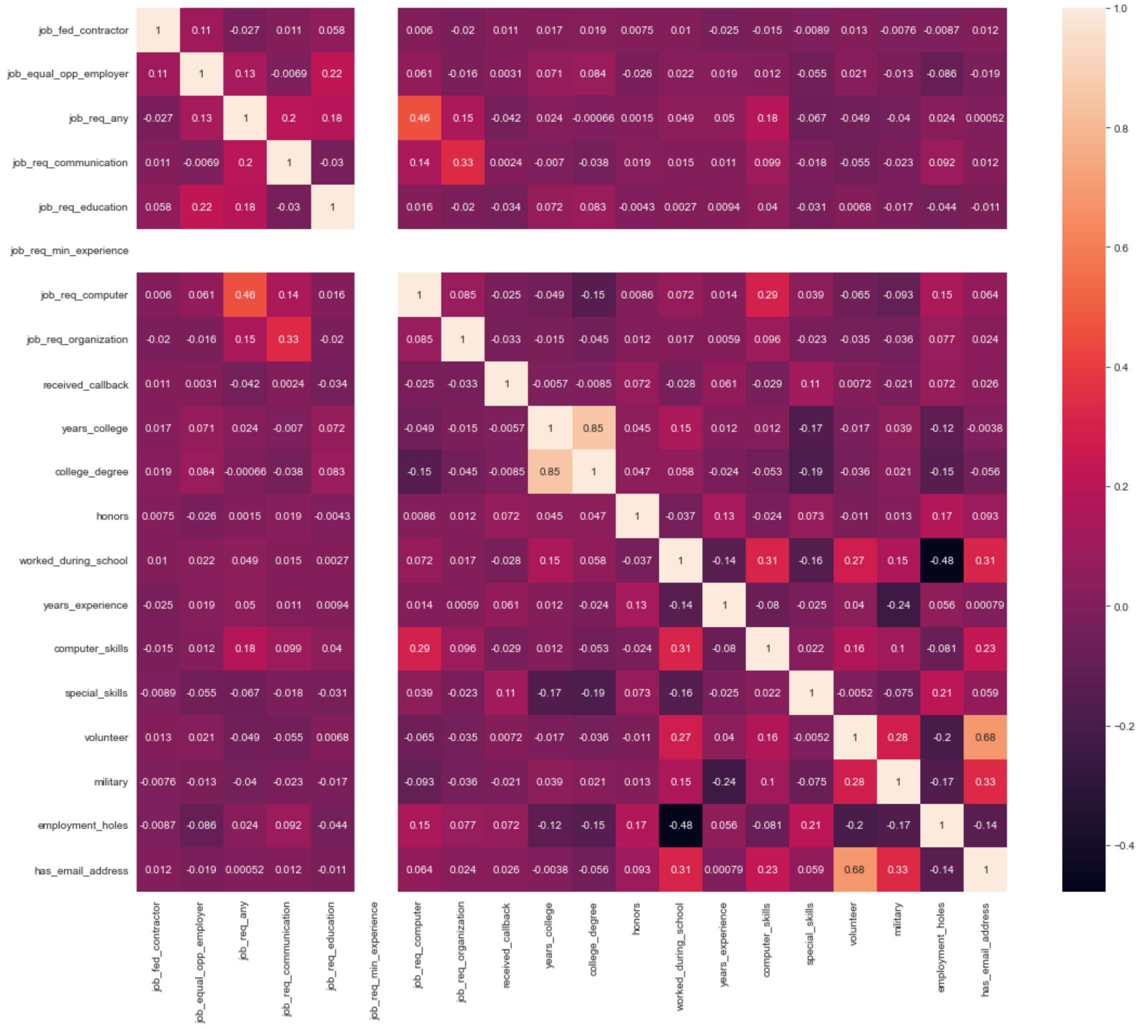
	job_fed_contractor	job_equal_opp_employer	job_req_any	job_req_c
job_fed_contractor	1.000000	0.105130	-0.026955	
job_equal_opp_employer	0.105130	1.000000	0.132151	
job_req_any	-0.026955	0.132151	1.000000	
job_req_communication	0.010839	-0.006880	0.196336	
job_req_education	0.057749	0.217472	0.179726	
job_req_min_experience	NaN	NaN	NaN	
job_req_computer	0.006033	0.061140	0.458128	
job_req_organization	-0.020190	-0.015798	0.145539	
received_callback	0.011277	0.003092	-0.041699	
years_college	0.017227	0.071437	0.023684	
college_degree	0.018514	0.084251	-0.000660	
honors	0.007451	-0.025939	0.001508	
worked_during_school	0.010046	0.022361	0.049213	
years_experience	-0.025097	0.019332	0.050161	
computer_skills	-0.014539	0.012347	0.177921	
special_skills	-0.008920	-0.055001	-0.066670	
volunteer	0.012938	0.020661	-0.048622	
military	-0.007561	-0.013315	-0.039612	
employment_holes	-0.008671	-0.085740	0.024396	
has_email_address	0.011656	-0.018630	0.000516	

In [106]:

```
# Heat Map
plt.figure(figsize=(18,15))
sns.heatmap(df.corr(), annot = True)
```

Out[106]:

```
<AxesSubplot:>
```



In [107]:

```
# check the mean
df.mean()
```

Out[107]:

```
job_fed_contractor        0.114765
job_equal_opp_employer    0.291170
job_req_any               0.787269
job_req_communication     0.124846
job_req_education         0.106776
job_req_min_experience    9.000000
job_req_computer          0.437166
job_req_organization      0.072690
received_callback         0.080493
years_college             3.618480
college_degree            0.719507
honors                    0.052772
worked_during_school      0.559548
years_experience          7.842916
computer_skills           0.820534
special_skills            0.328747
volunteer                 0.411499
military                  0.097125
employment_holes          0.448049
has_email_address         0.479261
dtype: float64
```

In [108]:

```
# check the median
df.median()
```

Out[108]:

```
job_fed_contractor        0.0
job_equal_opp_employer    0.0
job_req_any               1.0
job_req_communication     0.0
job_req_education         0.0
job_req_min_experience    9.0
job_req_computer          0.0
job_req_organization      0.0
received_callback         0.0
years_college             4.0
college_degree            1.0
honors                    0.0
worked_during_school      1.0
years_experience          6.0
computer_skills           1.0
special_skills            0.0
volunteer                 0.0
military                  0.0
employment_holes          0.0
has_email_address         0.0
dtype: float64
```

In [109]:

```
# check the quantile
df.quantile()
```

Out[109]:

```
job_fed_contractor        0.0
job_equal_opp_employer    0.0
job_req_any               1.0
job_req_communication     0.0
job_req_education         0.0
job_req_min_experience    9.0
job_req_computer          0.0
job_req_organization      0.0
received_callback         0.0
years_college             4.0
college_degree            1.0
honors                    0.0
worked_during_school      1.0
years_experience          6.0
computer_skills           1.0
special_skills            0.0
volunteer                 0.0
military                  0.0
employment_holes          0.0
has_email_address         0.0
Name: 0.5, dtype: float64
```

In [110]:

```
# check the minimum value
df.min()
```

Out[110]:

```
job_city                                      Boston
job_industry             business_and_personal_service
job_type                                    clerical
job_fed_contractor                               0.0
job_equal_opp_employer                             0
job_ownership                              nonprofit
job_req_any                                        0
job_req_communication                              0
job_req_education                                  0
job_req_min_experience                           9.0
job_req_computer                                   0
job_req_organization                               0
job_req_school                               college
received_callback                                  0
firstname                                      Aisha
race                                           black
gender                                             f
years_college                                      0
college_degree                                     0
honors                                             0
worked_during_school                               0
years_experience                                   1
computer_skills                                    0
special_skills                                     0
volunteer                                          0
military                                           0
employment_holes                                   0
has_email_address                                  0
resume_quality                                  high
dtype: object
```

In [111]:

```
# check the maximun value
df.max()
```

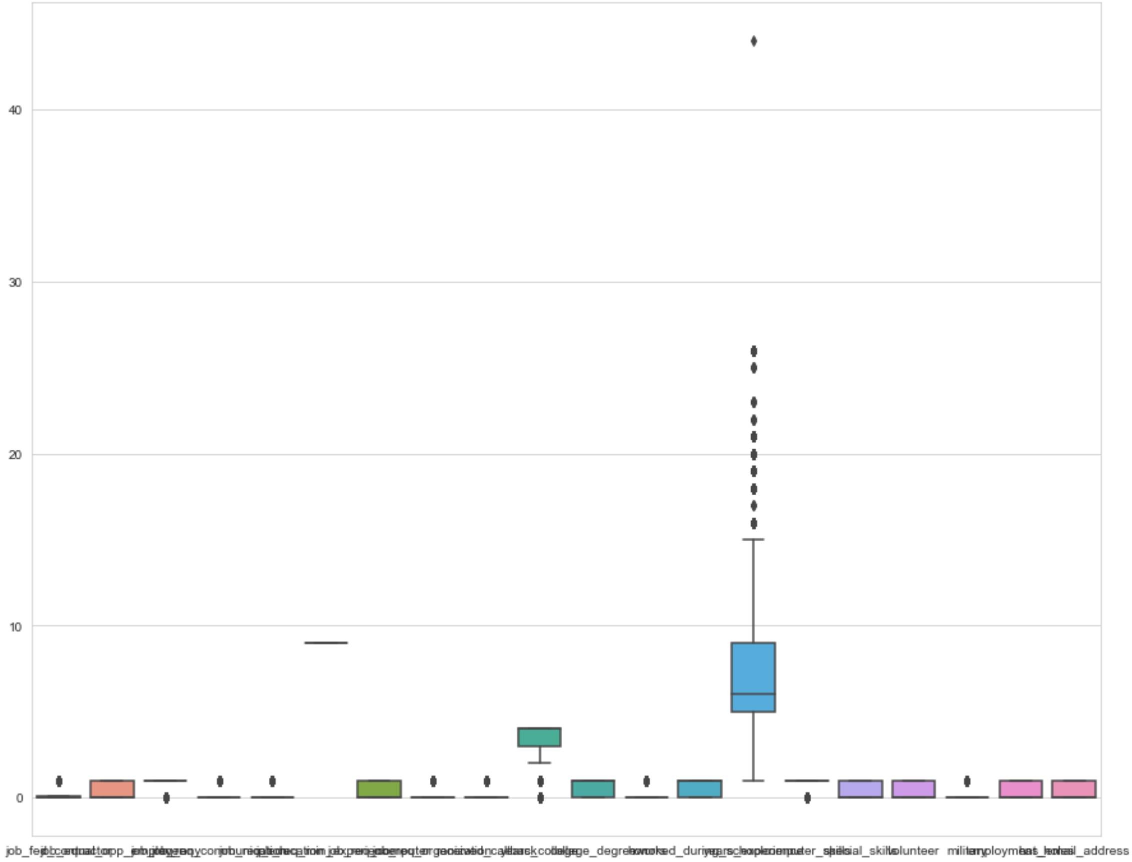
Out[111]:

```
job_city                                   Chicago
job_industry             wholesale_and_retail_trade
job_type                                supervisor
job_fed_contractor                             1.0
job_equal_opp_employer                           1
job_ownership                              unknown
job_req_any                                      1
job_req_communication                            1
job_req_education                                1
job_req_min_experience                         9.0
job_req_computer                                 1
job_req_organization                             1
job_req_school                        some_college
received_callback                                1
firstname                                   Tyrone
race                                         white
gender                                           m
years_college                                    4
college_degree                                   1
honors                                           1
worked_during_school                             1
years_experience                                44
computer_skills                                  1
special_skills                                   1
volunteer                                        1
military                                         1
employment_holes                                 1
has_email_address                                1
resume_quality                                 low
dtype: object
```

Box PLot

In [112]:

```
plt.figure(figsize=(15,12))
sns.boxplot(data=df , orient = 'v')
plt.show()
```



In [ ]:

```

```

In [ ]:

```

```

Binary Encoding

In [113]:

```
df['job_city'] = df['job_city'].replace({'Chicago':0 , 'Boston': 1})
df['race']  = df['race'].replace({'white':0,'black':1})
df['gender'] = df['gender'].replace({'f':0  , 'm': 1})

df['resume_quality'] = df['resume_quality'].replace({'low':0 , 'high':1})
```

Label Encoding

- After the binary encoding we have 6 columns are left Label Encoding job_industry , job_type ,

In [114]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [115]:

```
df['job_industry'] = le.fit_transform(df['job_industry'])
df['job_type'] = le.fit_transform(df['job_type'])
```

In [116]:

```
df.head()
```

Out[116]:

	job_city	job_industry	job_type	job_fed_contractor	job_equal_opp_employer	job_ownersh
0	0	2	5	0.114765	1	unknov
1	0	2	5	0.114765	1	unknov
2	0	2	5	0.114765	1	unknov
3	0	2	5	0.114765	1	unknov
4	0	3	4	0.000000	1	nonprc

In [ ]:

```
```

In [ ]:

```
```