In [2]:

```
import requests
from bs4 import BeautifulSoup as bs
import pandas as pd
import numpy as np
import csv
import seaborn as sns
import matplotlib.pyplot as plt
```

LOAD DATA FROM WEBSCRAPING ¶

In [44]:

```
df = pd.read_csv('./Data/jobsdf.csv')
```

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1807 entries, 0 to 1806
Data columns (total 15 columns):
Unnamed: 0     1807 non-null int64
category       1807 non-null object
company        1780 non-null object
jobclass       1807 non-null object
jobdate        1806 non-null object
jobid          1807 non-null int64
location       1807 non-null object
subcategory    1807 non-null object
suburb         1314 non-null object
title          1807 non-null object
worktype       1806 non-null object
period         1807 non-null object
uppersal       294 non-null float64
lowersal       393 non-null float64
finalsal       393 non-null float64
dtypes: float64(3), int64(2), object(10)
memory usage: 211.8+ KB
```

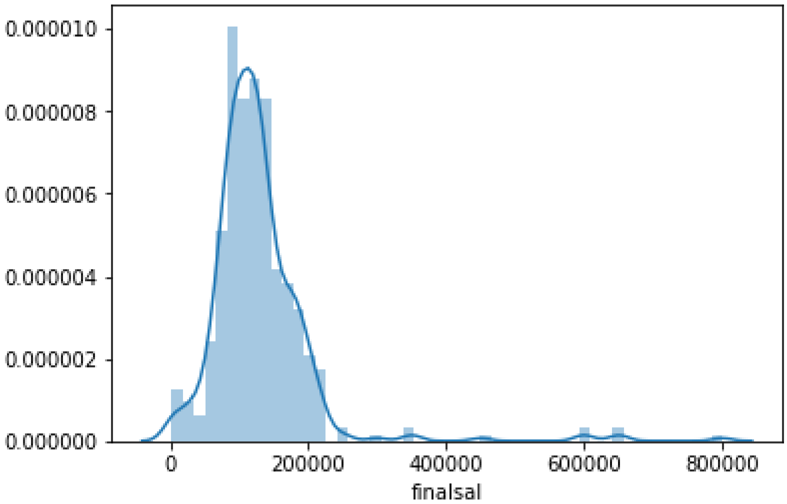
EDA VISUALISATION

In [196]:

```
# Check frequency distribution of salary data. Close to normal distribution with a few outl
x=df['finalsal'][df['finalsal']>0]
sns.distplot(x,bins=50)
```

Out[196]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2743972a5f8>
```

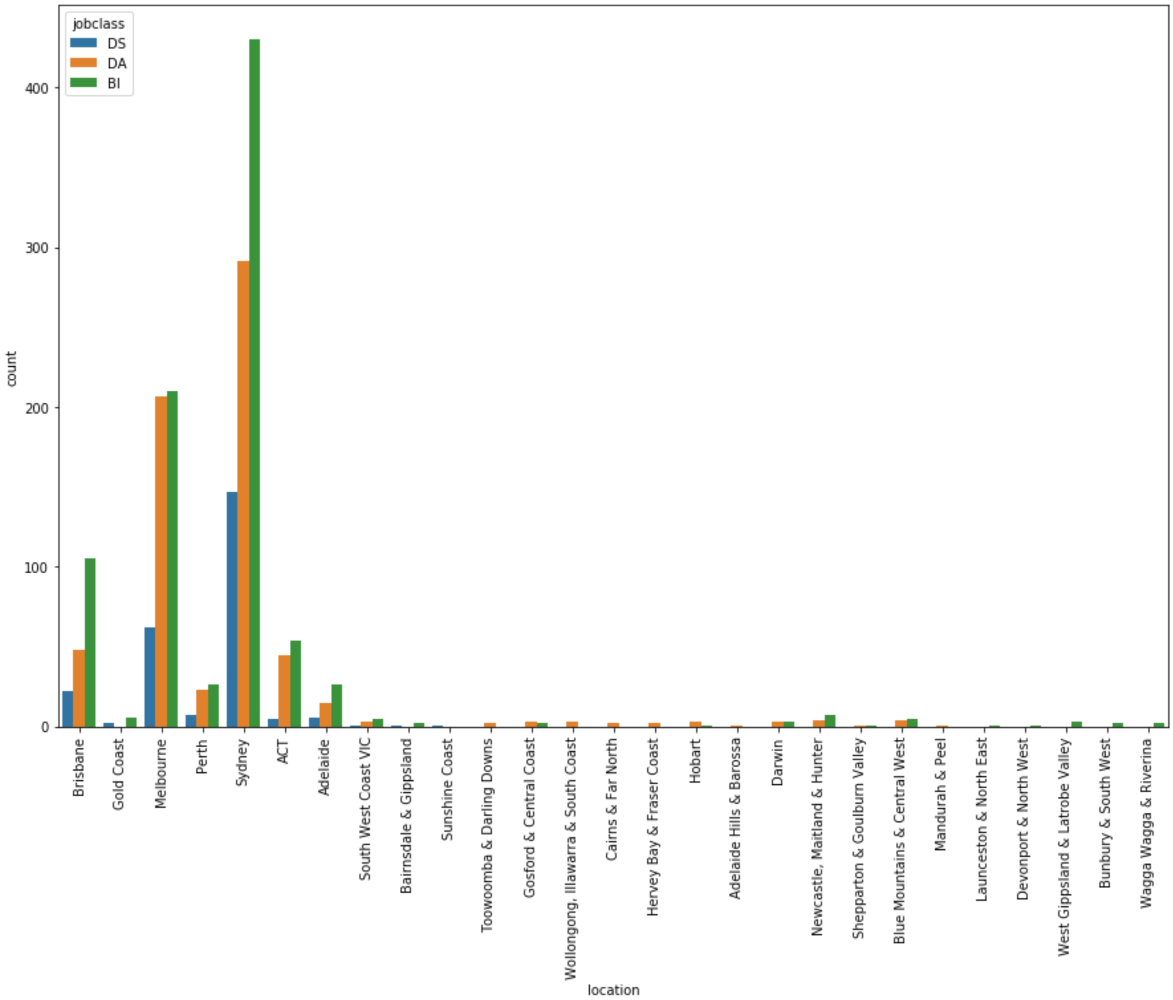


In [19]:

```
# Check dstribution of jobs by location. Most jobs are for capital cities only so include t
f, ax = plt.subplots(figsize=(15,10))
plt.xticks(rotation=90)
sns.countplot(data=df, x='location',hue='jobclass')
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x274388f31d0>
```

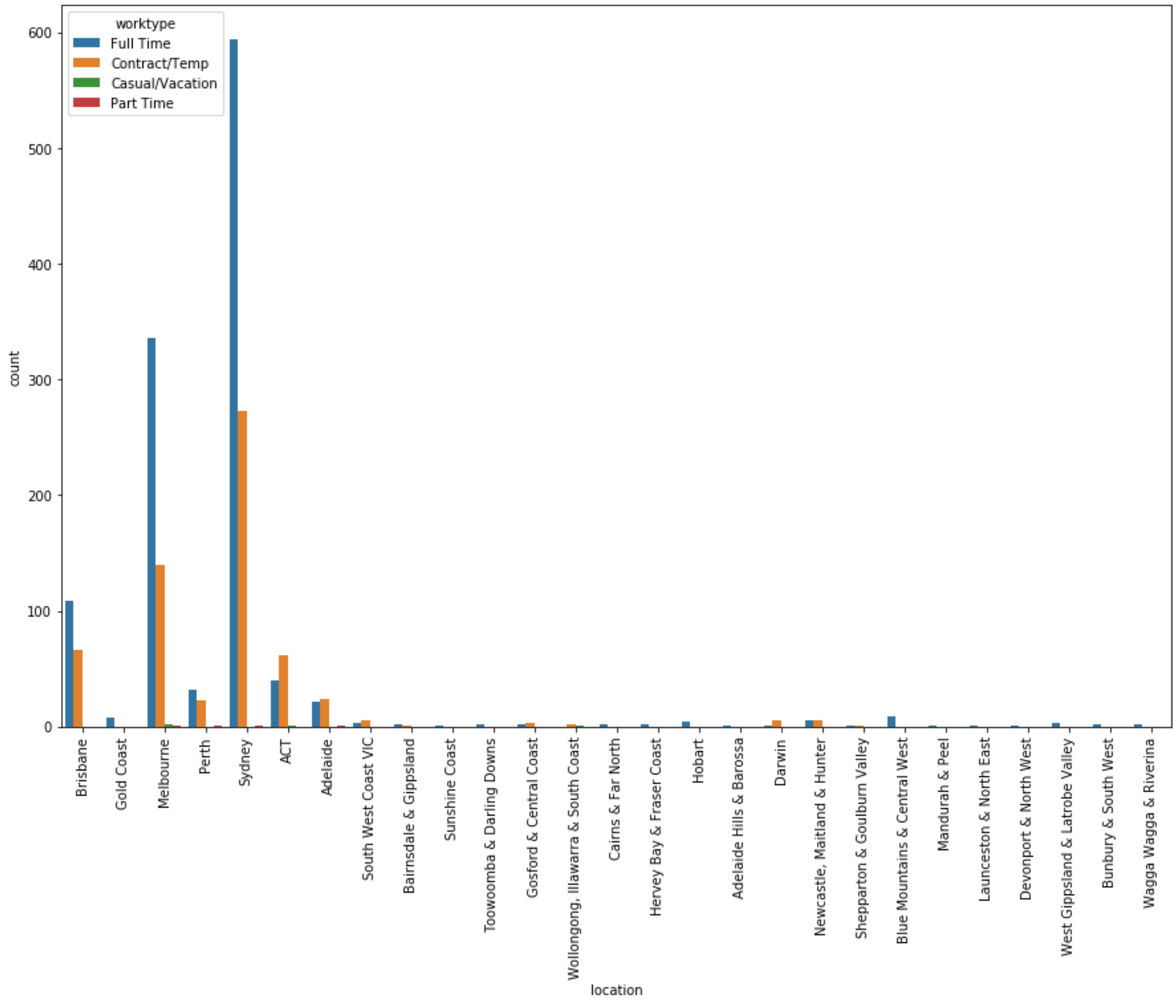


In [21]:

```
# Check dstribution of jobs by worktype and location. While more jobs are for full time, ke
f, ax = plt.subplots(figsize=(15,10))
plt.xticks(rotation=90)
sns.countplot(data=df,x='location',hue='worktype')
```

Out[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x274389ef400>
```

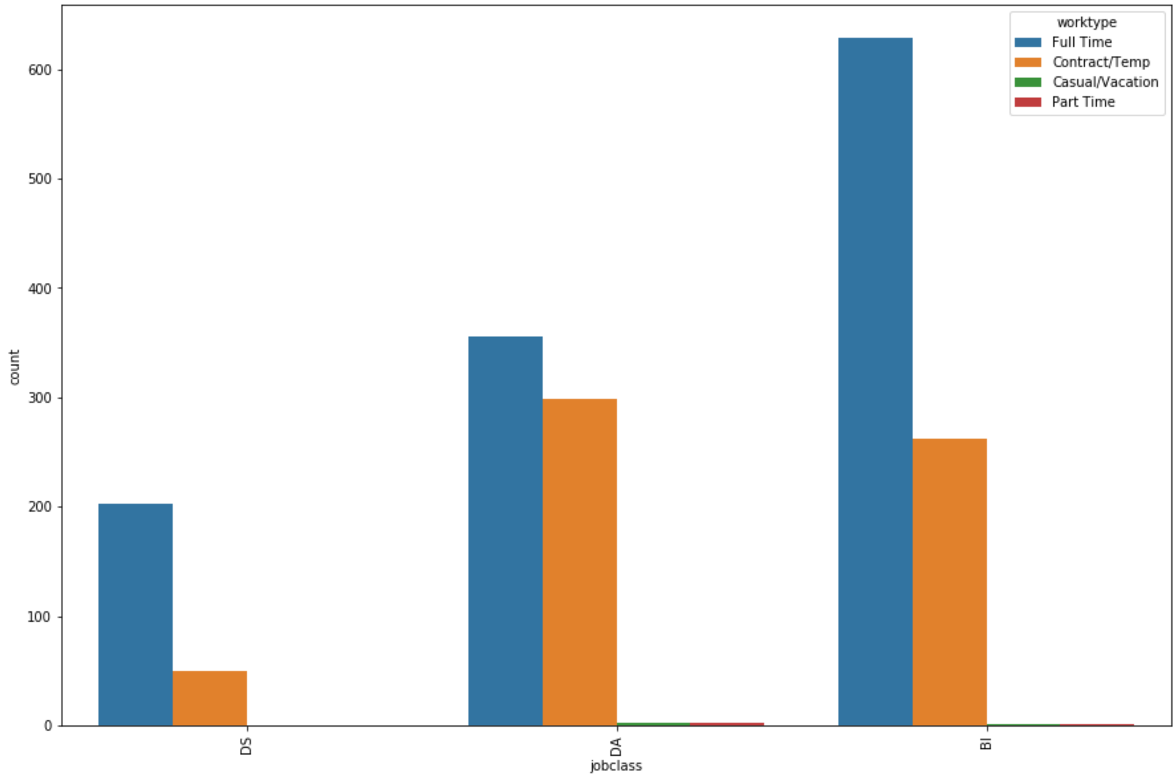


In [22]:

```
f, ax = plt.subplots(figsize=(15,10))
plt.xticks(rotation=90)
sns.countplot(data=df,x='jobclass',hue='worktype')
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x27438b08b70>
```



In [7]:

```
# Aggregate jobs and get Count, Mean and Median salaries by location and jobclass
x=df[['location','jobclass','finalsal','jobid']].groupby(['jobclass','location'])\
    .agg({'jobid':'count','finalsal':['mean','median']})
x.reset_index(inplace=True)
```

In [8]:

```
# Define list of capial cities
# Class - DATASCIENTIST Filter Count, Mean and Median salaries by location and jobclass
# There are no salaries given for Adelaide and Perth.
capc = ['Melbourne','Sydney','Brisbane','Adelaide','Perth','Hobart']
x[(x.jobclass=='DS')][ (x.location.isin(capc))]
```

```
C:\Users\Vinita Auplish\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
UserWarning: Boolean Series key will be reindexed to match DataFrame index.
```

Out[8]:

	jobclass	location	jobid	finalsal	
			count	mean	median
40	DS	Adelaide	6	NaN	NaN
42	DS	Brisbane	22	93723.812500	104895.5
44	DS	Melbourne	62	155833.277778	154999.5
45	DS	Perth	7	NaN	NaN
48	DS	Sydney	147	139905.430233	140000.0

In [80]:

```
# Class - DATAANALYST Filter Count, Mean and Median salaries by location and jobclass
# There are no salaries given for Adelaide
x[(x.jobclass=='DA')][ (x.location.isin(capc))]
```

```
C:\Users\Vinita Auplish\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

Out[80]:

	jobclass	location	jobid	finalsal	
			count	mean	median
21	DA	Adelaide	15	NaN	NaN
24	DA	Brisbane	48	123274.950000	124500.0
29	DA	Hobart	3	87024.500000	87024.5
31	DA	Melbourne	207	119510.612500	97500.0
33	DA	Perth	23	24500.000000	24500.0
36	DA	Sydney	291	146443.972973	125000.0

In [81]:

```
# Class - BUSINESSINTELLIGENCE Filter Count, Mean and Median salaries by location and jobcl
x[(x.jobclass=='BI')][ (x.location.isin(capc))]
```

```
C:\Users\Vinita Auplish\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

Out[81]:

	jobclass	location	jobid	finalsal	
			count	mean	median
1	BI	Adelaide	26	110000.000000	110000.0
4	BI	Brisbane	105	129923.076923	116000.0
10	BI	Hobart	1	NaN	NaN
12	BI	Melbourne	210	119856.478261	110000.0
14	BI	Perth	26	180000.000000	180000.0
17	BI	Sydney	430	126623.857143	120000.0

In [10]:

```
cats = ['Information & Communication Technology','Accounting Banking & Financial Services',
        'Government & Defence','Human Resources & Recruitment','Marketing & Communications'
y= df[['jobclass','category','finalsal','jobid']][df['category'].isin(cats)]\
   .groupby(['jobclass','category']).agg({'jobid':'count','finalsal':['mean','median']})
y.reset_index(inplace=True)
```

In [85]:

```
y[y.jobclass=='DS']
```

Out[85]:

	jobclass	category	jobid	finalsal	
			count	mean	median
10	DS	Human Resources & Recruitment	1	NaN	NaN
11	DS	Information & Communication Technology	151	122993.421875	122500.0
12	DS	Marketing & Communications	9	172500.000000	135000.0
13	DS	Sales	1	135000.000000	135000.0
14	DS	Science & Technology	49	168461.461538	175000.0

In [86]:

```
y[y.jobclass=='DA']
```

Out[86]:

	jobclass	category	jobid	finalsal	
			count	mean	median
5	DA	Human Resources & Recruitment	5	NaN	NaN
6	DA	Information & Communication Technology	419	144523.778481	120000.00
7	DA	Marketing & Communications	47	91411.954545	100000.00
8	DA	Sales	7	NaN	NaN
9	DA	Science & Technology	10	52509.750000	52509.75

In [88]:

```
y[y.jobclass=='BI']
```

Out[88]:

	jobclass	category	jobid	finalsal	
			count	mean	median
0	BI	Human Resources & Recruitment	24	63561.000000	56933.0
1	BI	Information & Communication Technology	604	133798.762500	120000.0
2	BI	Marketing & Communications	34	108700.000000	105000.0
3	BI	Sales	49	100267.821429	97500.0
4	BI	Science & Technology	4	95019.500000	95019.5

In [94]:

```
z= df[['jobclass','category','location','finalsal','jobid']][df['category'].isin(cats)][df[
   .groupby(['jobclass','location','category']).agg({'jobid':'count','finalsal':['mean','me
z.reset_index(inplace=True)
```

```
C:\Users\Vinita Auplish\Anaconda3\lib\site-packages\ipykernel_launcher.py:1:
UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  """Entry point for launching an IPython kernel.
```

In [95]:

```
z[z.jobclass=='DS']
```

Out[95]:

	jobclass	location	category	jobid	finalsal	
				count	mean	median
34	DS	Adelaide	Information & Communication Technology	6	NaN	NaN
35	DS	Brisbane	Information & Communication Technology	13	89255.785714	100000.0
36	DS	Brisbane	Marketing & Communications	1	NaN	NaN
37	DS	Brisbane	Science & Technology	1	NaN	NaN
38	DS	Melbourne	Human Resources & Recruitment	1	NaN	NaN
39	DS	Melbourne	Information & Communication Technology	33	155625.000000	155000.0
40	DS	Melbourne	Marketing & Communications	3	125000.000000	125000.0
41	DS	Melbourne	Science & Technology	16	176666.500000	175000.0
42	DS	Perth	Information & Communication Technology	6	NaN	NaN
43	DS	Sydney	Information & Communication Technology	86	123799.950000	127500.0
44	DS	Sydney	Marketing & Communications	4	220000.000000	220000.0
45	DS	Sydney	Sales	1	135000.000000	135000.0
46	DS	Sydney	Science & Technology	30	165999.950000	175000.0

In [96]:

```
z[z.jobclass=='DA']
```

Out[96]:

	jobclass	location	category	jobid	finalsal	
				count	mean	median
18	DA	Adelaide	Information & Communication Technology	10	NaN	NaN
19	DA	Adelaide	Marketing & Communications	1	NaN	NaN
20	DA	Brisbane	Information & Communication Technology	33	122968.687500	139375.00
21	DA	Brisbane	Marketing & Communications	4	100000.000000	100000.00
22	DA	Hobart	Information & Communication Technology	1	87024.500000	87024.50
23	DA	Melbourne	Information & Communication Technology	128	133753.565217	110000.00
24	DA	Melbourne	Marketing & Communications	13	82522.500000	82522.50
25	DA	Melbourne	Sales	3	NaN	NaN
26	DA	Melbourne	Science & Technology	5	NaN	NaN
27	DA	Perth	Information & Communication Technology	15	NaN	NaN
28	DA	Perth	Marketing & Communications	2	NaN	NaN
29	DA	Sydney	Human Resources & Recruitment	3	NaN	NaN
30	DA	Sydney	Information & Communication Technology	186	154779.714286	129999.75
31	DA	Sydney	Marketing & Communications	25	92560.812500	87499.75
32	DA	Sydney	Sales	4	NaN	NaN
33	DA	Sydney	Science & Technology	3	10000.000000	10000.00

In [97]:

```
z[z.jobclass=='BI']
```

Out[97]:

	jobclass	location	category	jobid	finalsal	
				count	mean	median
0	BI	Adelaide	Information & Communication Technology	23	110000.000000	110000.0
1	BI	Adelaide	Sales	1	NaN	NaN
2	BI	Brisbane	Human Resources & Recruitment	2	NaN	NaN
3	BI	Brisbane	Information & Communication Technology	84	137888.888889	116000.0
4	BI	Brisbane	Marketing & Communications	2	100000.000000	100000.0
5	BI	Brisbane	Sales	3	NaN	NaN
6	BI	Hobart	Information & Communication Technology	1	NaN	NaN
7	BI	Melbourne	Human Resources & Recruitment	7	102500.000000	102500.0
8	BI	Melbourne	Information & Communication Technology	132	125401.484848	110500.0
9	BI	Melbourne	Marketing & Communications	6	NaN	NaN
10	BI	Melbourne	Sales	15	115999.900000	30000.0
11	BI	Perth	Information & Communication Technology	22	180000.000000	180000.0
12	BI	Perth	Sales	3	NaN	NaN
13	BI	Sydney	Human Resources & Recruitment	13	35833.333333	45000.0
14	BI	Sydney	Information & Communication Technology	282	139899.612676	130000.0
15	BI	Sydney	Marketing & Communications	25	109666.666667	110000.0
16	BI	Sydney	Sales	26	91527.777778	120000.0
17	BI	Sydney	Science & Technology	2	NaN	NaN

CREATE DATAFRAME FOR PREDICTION

In [58]:

```
# Filter for Capital City locations and Selected job categories
# Features to be included - category, jobclass, location, finalsal
# Worktype could have been a determinant but the data is biased towards Full time
# Subcategory will be colinear with Category and Suburb will be colinear with Location
dfa = df[['jobid','category','location','jobclass','finalsal','worktype']][df.location.isir
dfn = df[['jobid','category','location','jobclass','finalsal','worktype']][df.location.isir
dfn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1286 entries, 0 to 1805
Data columns (total 6 columns):
jobid       1286 non-null int64
category    1286 non-null object
location    1286 non-null object
jobclass    1286 non-null object
finalsal    279 non-null float64
worktype    1286 non-null object
dtypes: float64(1), int64(1), object(4)
memory usage: 70.3+ KB

C:\Users\Vinita Auplish\Anaconda3\lib\site-packages\ipykernel_launcher.py:6:
UserWarning: Boolean Series key will be reindexed to match DataFrame index.
```

In [48]:

```
def mapcols(col,dic):
    print('Mapping',col,'with',dic)
    dfa[col].replace(dic, inplace=True)
    filenm = './Data/' + col+'.csv'
    w = csv.writer(open(filenm, "w"))
    for key, val in dic.items():
        w.writerow([key, val])
    dfa[col].value_counts()
```

In [34]:

```
def mapcolsnum(col,dic):
    print('Mapping',col,'with',dic)
    dfn[col].replace(dic, inplace=True)
    filenm = './Data/' + col+'num.csv'
    w = csv.writer(open(filenm, "w"))
    for key, val in dic.items():
        w.writerow([key, val])
    dfn[col].value_counts()
```

In [49]:

```
# Define dictionary of classifications to map
# Map category to dictionary
dcats = {'Information & Communication Technology':'IT','Accounting Banking & Financial Serv
         'Consulting & Strategy':'CS','Government & Defence':'GD','Human Resources & Recrui
         'Marketing & Communications':'MC','Sales':'SAL','Science & Technology':'ST'}
mapcols('category',dcats)
```

```
Mapping category with {'Information & Communication Technology': 'IT', 'Acco
unting Banking & Financial Services': 'ABFS', 'Consulting & Strategy': 'CS',
'Government & Defence': 'GD', 'Human Resources & Recruitment': 'HR', 'Market
ing & Communications': 'MC', 'Sales': 'SAL', 'Science & Technology': 'ST'}
```

In [50]:

```
# Define dictionary of locations to map
# Map location to dictionary
dcapc = {'Melbourne':'MEL','Sydney':'SYD','Brisbane':'BRI','Adelaide':'ADE','Perth':'PER','
mapcols('location',dcapc)
```

```
Mapping location with {'Melbourne': 'MEL', 'Sydney': 'SYD', 'Brisbane': 'BR
I', 'Adelaide': 'ADE', 'Perth': 'PER', 'Hobart': 'HOB'}
```

In [51]:

```
# Define dictionary of worktype to map
# Map worktype to dictionary
dwtype = {'Full Time':'FT', 'Contract/Temp':'CON', 'Casual/Vacation':'CAS', 'Part Time':'PT
mapcols('worktype',dwtype)
```

```
Mapping worktype with {'Full Time': 'FT', 'Contract/Temp': 'CON', 'Casual/Va
cation': 'CAS', 'Part Time': 'PT'}
```

In [52]:

```
dfa.worktype.value_counts()
```

Out[52]:

```
FT     835
CON    447
PT       3
CAS      1
Name: worktype, dtype: int64
```

In [53]:

```
# Create dummy variables
dfe = pd.get_dummies(dfa,columns= ['category','location','jobclass','worktype'])
dfe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1286 entries, 0 to 1805
Data columns (total 20 columns):
jobid           1286 non-null int64
finalsal        279 non-null float64
category_HR     1286 non-null uint8
category_IT     1286 non-null uint8
category_MC     1286 non-null uint8
category_SAL    1286 non-null uint8
category_ST     1286 non-null uint8
location_ADE    1286 non-null uint8
location_BRI    1286 non-null uint8
location_HOB    1286 non-null uint8
location_MEL    1286 non-null uint8
location_PER    1286 non-null uint8
location_SYD    1286 non-null uint8
jobclass_BI     1286 non-null uint8
jobclass_DA     1286 non-null uint8
jobclass_DS     1286 non-null uint8
worktype_CAS    1286 non-null uint8
worktype_CON    1286 non-null uint8
worktype_FT     1286 non-null uint8
worktype_PT     1286 non-null uint8
dtypes: float64(1), int64(1), uint8(18)
memory usage: 52.7 KB
```

In [59]:

```
# Write to csv files
dfe.to_csv('./Data/dfe.csv')
```

In [60]:

```
# Map column to numeric values for classifications
dcatsn = {'Information & Communication Technology':1,'Accounting Banking & Financial Servic
         'Consulting & Strategy':3,'Government & Defence':4,'Human Resources & Recruitment'
         'Marketing & Communications':6,'Sales':7,'Science & Technology':8}
mapcolsnum('category',dcatsn)
```

```
Mapping category with {'Information & Communication Technology': 1, 'Account
ing Banking & Financial Services': 2, 'Consulting & Strategy': 3, 'Governmen
t & Defence': 4, 'Human Resources & Recruitment': 5, 'Marketing & Communicat
ions': 6, 'Sales': 7, 'Science & Technology': 8}
```

In [61]:

```
# Map column to numeric values for location
dcapcn = {'Melbourne':1,'Sydney':2,'Brisbane':3,'Adelaide':4,'Perth':5,'Hobart':6}
mapcolsnum('location',dcapcn)
```

```
Mapping location with {'Melbourne': 1, 'Sydney': 2, 'Brisbane': 3, 'Adelaid
e': 4, 'Perth': 5, 'Hobart': 6}
```

In [62]:

```
# Map column to numeric values for job class
dclsn = {'DS':1,'DA':2,'BI':3}
mapcolsnum('jobclass',dclsn)
```

```
Mapping jobclass with {'DS': 1, 'DA': 2, 'BI': 3}
```

In [63]:

```
# Map column to numeric values for job type
dwtypen = {'Full Time':1, 'Contract/Temp':1, 'Casual/Vacation':3, 'Part Time':4}
mapcolsnum('worktype',dwtypen)
```

```
Mapping worktype with {'Full Time': 1, 'Contract/Temp': 1, 'Casual/Vacatio
n': 3, 'Part Time': 4}
```

In [64]:

```
dfn.head()
```

Out[64]:

	jobid	category	location	jobclass	finalsal	worktype
0	36129622	1	3	1	120000.0	1
2	36122943	1	1	1	NaN	1
3	36120661	1	1	1	NaN	1
4	36123336	8	1	1	200000.0	1
5	36117119	1	5	1	NaN	1

In [65]:

```
dfn.to_csv('./Data/dfn.csv')
```

In [ ]:

```

```