FIT5145 - Introduction to Data Science

Summer Semester B 2020

Assignment 1

This assesment aims to guide you in exploring a data set through the process of exploratory data analysis (EDA), primarily through visualisation of that data using various data science tools.

You will need to draw on what you have learnt and will continue to learn, in class. You are also encouraged to seek out alternative information from reputable sources. If you use or are 'inspired' by any source code from one of these sources, you must reference this.

Learning outcomes You will learn the following through completing this assessment:

1. Read in files and extract data from them into a data frame.
2. Wrangle and process data.
3. Use graphical and non-graphical tools to perform EDA.
4. Use basic tools for managing and processing big data.
5. Determine information
6. Communicate your findings in your report.

Submission details The Python code as a Jupyter notebook file (.ipyn). A PDF print of your Jupyter notebook containing the code, figures and answers to all the questions. Hint: Wrap your code using the Jupyter magics or pythonic standard.

Please note: Marks will be assigned based on their correctness and clarity of your answers and code. The PDF should be concise and not take up an excessive number of pages. You should not print the data frames in your PDF (comment out the code that prints those).

Zip file submissions attract a penalty of 10%. Submit two separate files requested above together. You will need to submit your PDF to Turnitin.

Task

In this course, you have learned about the definitions, skill sets, tools, applications and knowledge domains attributed to data science. However, these are extremely diverse and make data science challenging to define precisely. By completing the EDA, we hope you can get a clearer understanding of how a career in data science compares to others in the IT industry.

The Data

In late 2018, a survey was conducted for a large Australian collective of IT professionals. The survey, which received 7000 responses, aimed to gather information about IT professionals. The dataset was made public, and many insights have emerged since. We have taken the data set and heavily modified the data. Both to clean the data, a significant component of data science and to ensure original assignment submission.

The data set is called *assignment1_dataset.csv*, and contains respondents answers to survey questions. Each column contains the answers of one respondent to a specific question. Do not alter this dataset.

How to complete this assesment

The following notebook has been constructed to provide you with directions (blue), questions (yellow) and background information. Responses to both blue directions and yellow questions are assessed.

Underneath the blue direction boxes, there are empty cells with the comment #Your code. Place your code in these. You should not need to but may insert new cells under this cell if required.

To respond to questions you should double click on the cell beneath each question with the comment Answer. Write your answer under these.

Please note, your commenting and adherence to Python code standards will be marked. This notebook has been designed to give you a template for the layout of future notebooks you might create. If you require further information on Python standards, please visit https://www.python.org/dev/peps/pep-0008/ (https://www.python.org/dev/peps/pep-0008/)

Do not change any of the directions or answer boxes, the order of questions, order of code entry cells or the name of the input files.

Table of contents

Enter your information in the following cell. Please make sure you specify what version of python you are using as your tutor may not be using the same version and will adjust your code accordingly.

Student Information

Please enter your details here.

Name: Gayatri Aniruddha

Student number: 30945305

Tutorial number. : 02 P1 : 1 PM to 3 PM

Tutor: Tooba Jalalidil

Environment: Python 3.7 and Anaconda 5.3.0 (64-bit)

Load your libraries and files

This assesment will be conducted using pandas. You will also be required to create visualisations. We recommend Seaborn, which is more visually appealing than matplotlib. However, you may choose either. For further information on Seaborn visit https://seaborn.pydata.org/ (https://seaborn.pydata.org/)

Hint: Remember to comment on what each library does.

In [506]:

```
# Your code
# Loading libraries
import pandas as pd # for loading CSV file and reading the data
import seaborn as sns # for visualisation and creating graphs
import numpy as np # for mathematical and statistical operations
import matplotlib.pyplot as plt #for visualisation and creating graphs

# # loading and reading assignment1_dataset file as a1_dataset
a1_dataset = pd.read_csv('assignment1_dataset.csv')
```

1. Demographic Analysis

Who are the survey participants?

Let's get a general understanding of the characteristics of the survey participants. Demographic overviews are a standard way to start an exploration of survey data. The types of participants can heavily affect survey responses.

1.1 Age

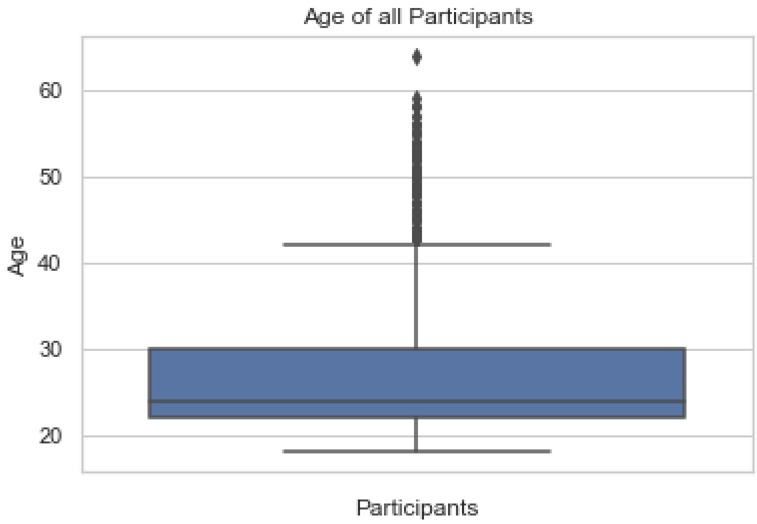
Visualisation is a quick and easy way to gain an overview of the data. One method is through a boxplot. Boxplots are a way to show the distribution of numerical data and display the five descriptive statistics: minimum, first quartile, median, third quartile, and maximum. Outliers should also be shown.

1. Create a box plot showing the age of all the participants.

Your plot must have labels for each axis, a title, numerical points for the age axis and also show the outliers.

In [507]:

```
# your code

# box plot showing age of all participants
sns.boxplot(y = a1_dataset['Age'])

plt.title('Age of all Participants')
plt.ylabel('Age')
plt.xlabel('Participants')
plt.show()
```



2. Calculate the five descriptive statistics as shown on the boxplot, as well as the mean.
Round your answer to the nearest whole number.

In [508]:

```
# Your code
# Descriptive statistics are:
# 1) minimum 2) first quartile 3) median,
# 4) third quartile 5) maximum

# Performing the calculations
minimum_age = a1_dataset['Age'].min()
maximum_age = a1_dataset['Age'].max()
median_age = a1_dataset['Age'].median()
mean_age = a1_dataset['Age'].mean()
first_quartile = np.percentile(a1_dataset['Age'],25)
third_quartile = np.percentile(a1_dataset['Age'],75)

# Printing out the descriptive statistics
print("Five descriptive statistics from the boxplot are:")
print("Minimum age :", minimum_age)
print("Maximum age :", maximum_age)
print("Median age :", round(median_age))
print("Mean age :", round(mean_age))
print("First Quartile :", round(first_quartile))
print("Third Quartile :", round(third_quartile))
```

```
Five descriptive statistics from the boxplot are:
Minimum age : 18
Maximum age : 64
Median age : 24
Mean age : 27
First Quartile : 22.0
Third Quartile : 30.0
```

Answer

> 3.i. Looking at the boxplot, what general conclusion can you make about the age of the participants? You must explain your answer with reference to all five descriptive statistics. Simply listing will not suffice. You must discuss the conclusions drawn based on these descriptive statistics' relationship to each other. You must also make mention of the outliers if there are any.
>
> 3.ii. Would the mode be greater or lower than the mean? Why?

Answer i : From the boxplot, we understand that the minimum age is below 20 years and the maximum age is above 60 years. Most of the respondents are young i.e their ages are between 20 and 30 years of age. Hence, the first quartile lies at 22 and the third quartile lies at 30. This explains why the median age is 24 and the mean age is 27 years. Thus, the median indicates that, from the survey conducted, ages of most respondents revolve around 24 years. There are a few respondents above 40 years and few above 60 years. These are the outliers in the boxplot. Furthermore, we can also deduce that the data in the box plot is right skewed.

Answer ii : From the above data, we know that the boxplot is right skewed. In such a boxplot, we know that the mode is always lesser than the mean. This is clear from the descriptive statistics provided as well. From them, we understand that the median is lesser than the mean. This means that the distribution has more values closer and around the median. Hence, mode i.e 21 years has to be lower than the mean i.e 27 years.

4. Regardless of the errors that the data show, we are interested in working-age IT professionals, aged between 20 and 65.

Calculate how many respondents were under 20 or over 65?

In [459]:

```
# Your code

# Respondents under 20 and over 65 years of age
respondents = 0
for item in a1_dataset.Age:
    if item < 20 or item > 65:
        respondents += 1

print(" Total Number of respondents under 20 or over 65 years of age are :", respondents)
```

```
 Total Number of respondents under 20 or over 65 years of age are : 90
```

1.2 Gender

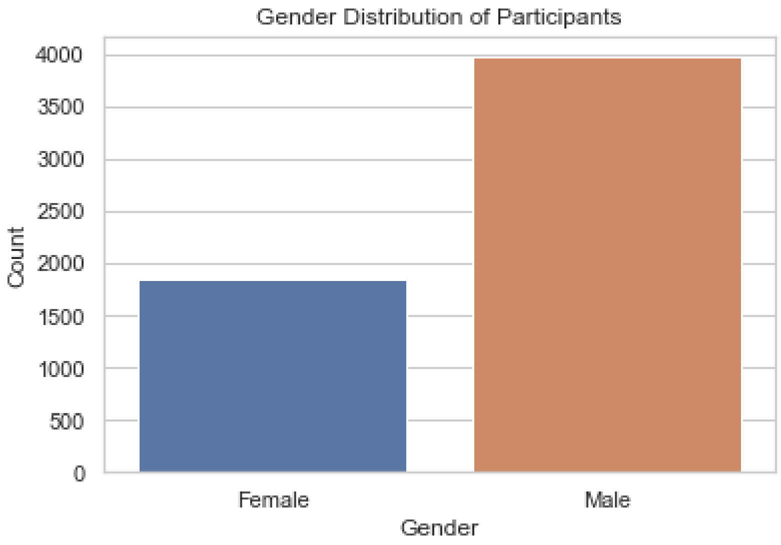
We are interested in the gender of respondents. Within the STEM fields, there are more males than females or other genders. In 2016 the Office of the chief scientist found that women held only 25% of jobs in STEM. Let's see how that compares to our participants.

5. Plot the gender distribution of survey participants.

In [509]:

```
# Your code

# Taking a count of respondents of each gender
fun = {'Gender' : {'Count' : lambda x: x.count()}}

# Grouping the count of respondents by gender
groupbygender = a1_dataset.groupby('Gender').agg(fun)
groupbygender = groupbygender.reset_index()
groupbygender.columns = groupbygender.columns.droplevel(0)
groupbygender.rename(columns = {'':'Gender'},inplace = True)

# Plotting the bar plot
sns.barplot(x = 'Gender', y = 'Count', data = groupbygender)

plt.title('Gender Distribution of Participants')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\Gayatri Aniruddha\Anaconda3\lib\site-packages\pandas\core\groupby\g
eneric.py:1315: FutureWarning: using a dict with renaming is deprecated and
will be removed in a future version
  return super(DataFrameGroupBy, self).aggregate(arg, *args, **kwargs)
```



6. Calculate what percentage of respondents were men and what percentage were women.

In [493]:

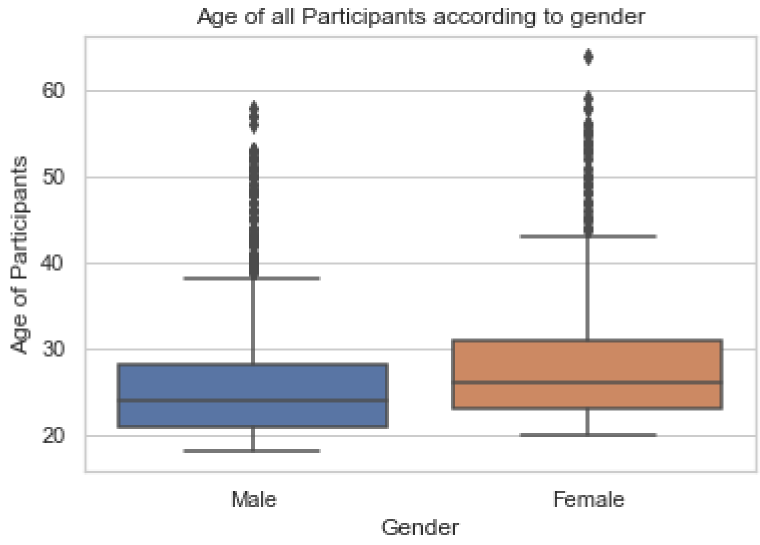
```
# Your code

# Total number of respondents
total = groupbygender["Count"].sum()

# Percentage of male respondents
male = round( (groupbygender["Count"][1] / total) * 100, 2)

# Percentage of female respondents
female = round( (groupbygender["Count"][0] / total) * 100, 2)

print("Percentage of Male respondents:", male, "%")
print("Percentage of Female respondents:", female, "%")
```

```
Percentage of Male respondents: 68.35 %
Percentage of Female respondents: 31.65 %
```

7. Let's see if there is any relationship between age and gender.
Create a box plot showing the age of all the participants according to gender.

In [510]:

```
# Your code

# box plot for age of all the participants according to gender
sns.boxplot(x = 'Gender', y = 'Age', data = a1_dataset)

plt.title('Age of all Participants according to gender')
plt.xlabel('Gender')
plt.ylabel('Age of Participants')
plt.show()
```



8. What comments can you make about the relationship between the age and gender of the respondents?

Hint: You need to determine the descriptive statistics.

In [511]:

```
# Your code

# Calculating the Descrptive statistics
fun = {'Minimum Age': 'min', 'Maximum Age': 'max',
       'First Quartile': lambda x: round(np.percentile(x,25)),
       'Third Quartile' : lambda x: round(np.percentile(x,75)),
       'Mean':'mean','Count':'count','Standard Deviation':'std'}

# Grouping by Gender
groupbygender = a1_dataset.groupby('Gender')['Age'].agg(fun).reset_index()
groupbygender
```

```
C:\Users\Gayatri Aniruddha\Anaconda3\lib\site-packages\ipykernel_launcher.p
y:10: FutureWarning: using a dict on a Series for aggregation
is deprecated and will be removed in a future version
  # Remove the CWD from sys.path while we load stuff.
```

Out[511]:

	Gender	Minimum Age	Maximum Age	First Quartile	Third Quartile	Mean	Count	Standard Deviation
0	Female	20	64	23	31	28.285870	1840	7.646899
1	Male	18	58	21	28	26.058128	3974	6.513043

* From the data, we can see that the average age of female employees, which is around 28 years, is slightly greather than that of male employees, which is around 26 years.

* The maximum age of female employees, which is around 64 years is greater than the maximum age of male employees, which is around 58 years.

* The minimum age found in male employees, which is 18 years, is slightly lesser than that found in female employees, which is 20 years. Thus, we can say that male employees start working at an earlier age when compared to female employees. Similarly, female employees retire comparatively later when compared to male employees.

* Thus, we can say that male employees start working at an earlier age when compared to female employees. Similary, female employees retire comparatively later when compared to male employees.

1.3 Country

We know that people practice IT all over the world. The United States is thought of as a central 'hub' for commercial IT services as well as research followed by the United Kingdom and Germany.

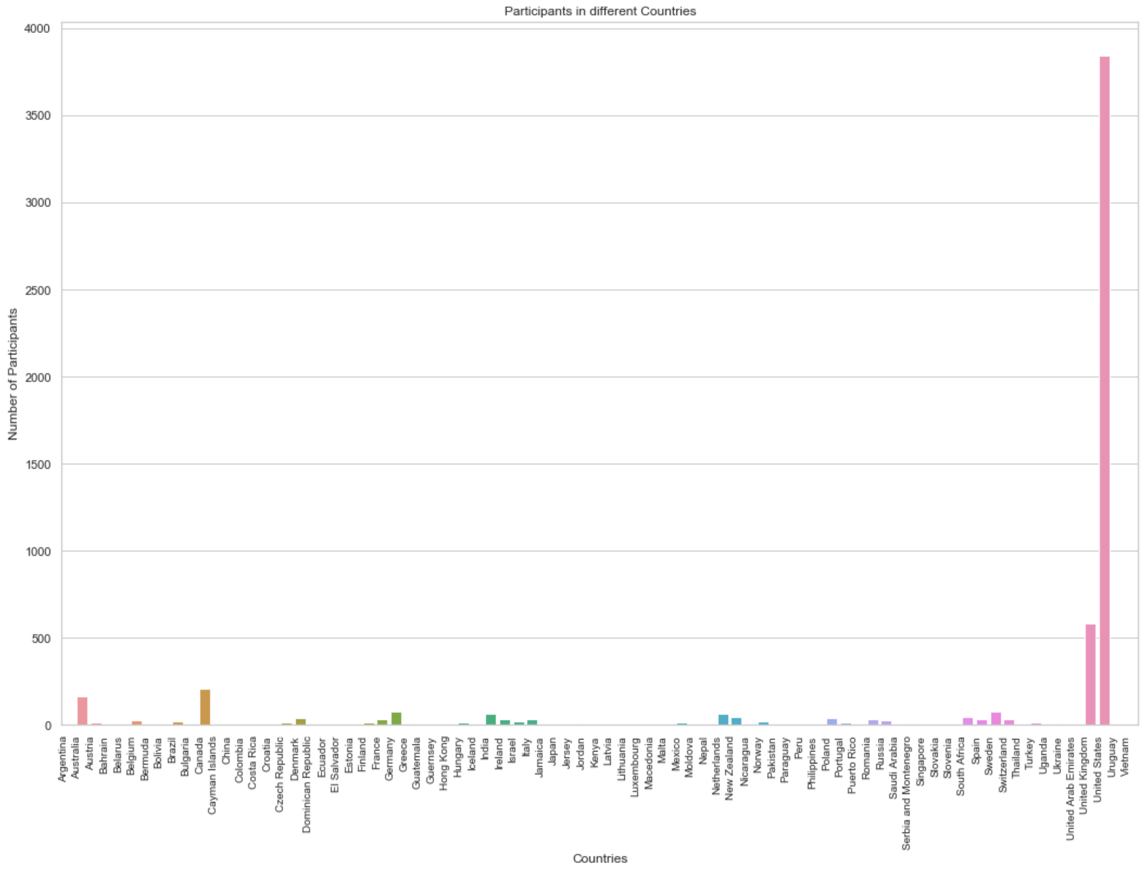
Because the field is evolving so quickly, and it may be that these perceptions, formed in the late 2000's are now inaccurate. So let's find out where IT professionals live.

9. Create a bar graph of the respondents according to which country they are from.
Find the percentage of respondents from the top 5 countries.
Print your display rounding to two decimal places before writing out your answer.

In [464]:

```
# Your code
# Part 1 : bar graph of the respondents according to their country

# Calculating the country count
fun = {'Country' : {'Count' : lambda x: x.count()}}

# Grouping the data by country
groupbycountry = a1_dataset.groupby('Country').agg(fun)
groupbycountry = groupbycountry.reset_index()
groupbycountry.columns = groupbycountry.columns.droplevel(0)
groupbycountry.rename(columns = {'':'Country'},inplace = True)

# Total count of all the countries
total = groupbycountry["Count"].sum()

# Plotting the bar plot
plt.figure( figsize = (15,10) )
bar_plot = sns.barplot(x = 'Country', y = 'Count',
                       data = groupbycountry)

plt.title('Participants in different Countries')
plt.xlabel('Countries')
plt.ylabel('Number of Participants')
plt.tight_layout()

bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation=90,
                         ha="right",fontsize=10)
plt.show()

# Part 2 : Percentage of respondents from the top 5 countries
# Sorting the values based on the 'Count' value
groupbycountry.sort_values(["Count"],axis=0, ascending=False, inplace=True)
groupbycountry = groupbycountry.reset_index()

print("The percentage of respondants of top 5 countries are :")
for i in range(5):
    print(groupbycountry.Country[i],":",
          round((groupbycountry.Count[i]/total)*100 ,2),"%")
```

```
C:\Users\Gayatri Aniruddha\Anaconda3\lib\site-packages\pandas\core\groupby\g
eneric.py:1315: FutureWarning: using a dict with renaming is deprecated and
will be removed in a future version
  return super(DataFrameGroupBy, self).aggregate(arg, *args, **kwargs)
```



```
The percentage of respondants of top 5 countries are :
United States : 66.15 %
United Kingdom : 10.04 %
Canada : 3.61 %
Australia : 2.87 %
Sweden : 1.32 %
```

10. Find the percentage of respondents from the top 5 countries.
Print your display rounding to two decimal places before writing out your answer.

In [465]:

```
# Your code

# percentages of respondents from the top 5 countries
print("The percentage of respondants of top 5 countries are :")
for i in range(5):
    print( groupbycountry.Country[i],":",
          round((groupbycountry.Count[i]/total)*100 , 2), "%" )


# For the next question
print("**********************************")
germany = round( (groupbycountry.Count[5]/total)*100 , 2 )
print( "Calculations for the next question:")
print("Percentage of respondents from Germany :", germany, "%")
```

```
The percentage of respondants of top 5 countries are :
United States : 66.15 %
United Kingdom : 10.04 %
Canada : 3.61 %
Australia : 2.87 %
Sweden : 1.32 %
**********************************
Calculations for the next question:
Percentage of respondents from Germany : 1.29 %
```

Answer

11. What comments can you make about the United States, the United Kingdom and Germany? Are these results consistent with what you expected?
Explain why.

From the above graph, we can deduce that United States has the most percentage of respondents around 66 %.

United kingdom has the second largest percentage of respondents around 10 %.

Germany stands 6th in the list with just 1.29 % of respondents.

The United States is thought of as a central 'hub' for commercial IT services as well as research followed by the United Kingdom and Germany.

Thus, these results are consistent with what we expected.

12. Now that we have another demographic variable let's see if there is any relationship between country, age and gender. We are specifically interested in the top 5 countries.
Calculate the mean, median and count for the ages of each gender for each of these countries.

Hint: You may need to create a copy or slice.

In [466]:

```
# Your Code
# Our Top 5 Countries : United States, United Kingdom, Canada, Australia, Sweden

# Filtering out the required data
countries = ['United States','United Kingdom', 'Canada', 'Australia','Sweden']
country_df = a1_dataset[a1_dataset.Country.isin(countries)].reset_index()

# Calculating the mean, median and count of ages
fun = {'Age':{'Mean Age':'mean','Median Age':'median'},
     'Gender' : {'Count' : lambda x: x.count()}}

# Grouping the data by country
statistics_df = country_df.groupby(['Country','Gender']).agg(fun)
statistics_df = statistics_df.reset_index()
statistics_df.columns = statistics_df.columns.droplevel(0)
statistics_df.columns = ['Country','Gender','Mean Age',
                         'Median Age','Gender Count']

statistics_df
```

```
C:\Users\Gayatri Aniruddha\Anaconda3\lib\site-packages\pandas\core\groupby\g
eneric.py:1315: FutureWarning: using a dict with renaming is deprecated and
will be removed in a future version
  return super(DataFrameGroupBy, self).aggregate(arg, *args, **kwargs)
```

Out[466]:

	Country	Gender	Mean Age	Median Age	Gender Count
0	Australia	Female	27.863636	26.5	44
1	Australia	Male	26.902439	25.0	123
2	Canada	Female	26.658537	25.0	41
3	Canada	Male	26.869822	25.0	169
4	Sweden	Female	27.050000	25.0	20
5	Sweden	Male	26.912281	25.0	57
6	United Kingdom	Female	25.963415	24.0	164
7	United Kingdom	Male	24.754762	22.0	420
8	United States	Female	28.709538	26.0	1384
9	United States	Male	26.310317	23.5	2462

13. What Pattern do you notice about the relationship between age, gender for each of these countries? (if any).

In Australia, Sweden, United Kingdom and United States, the average working age of women is greater than the average working age of men.

In Canada, the average working age of men and women is nearly the same.

Similarly, in all the countries, there are more number of working men than working women.

We can also notice that the median age for the respondents is near mid-twenties irrespective of their gender.

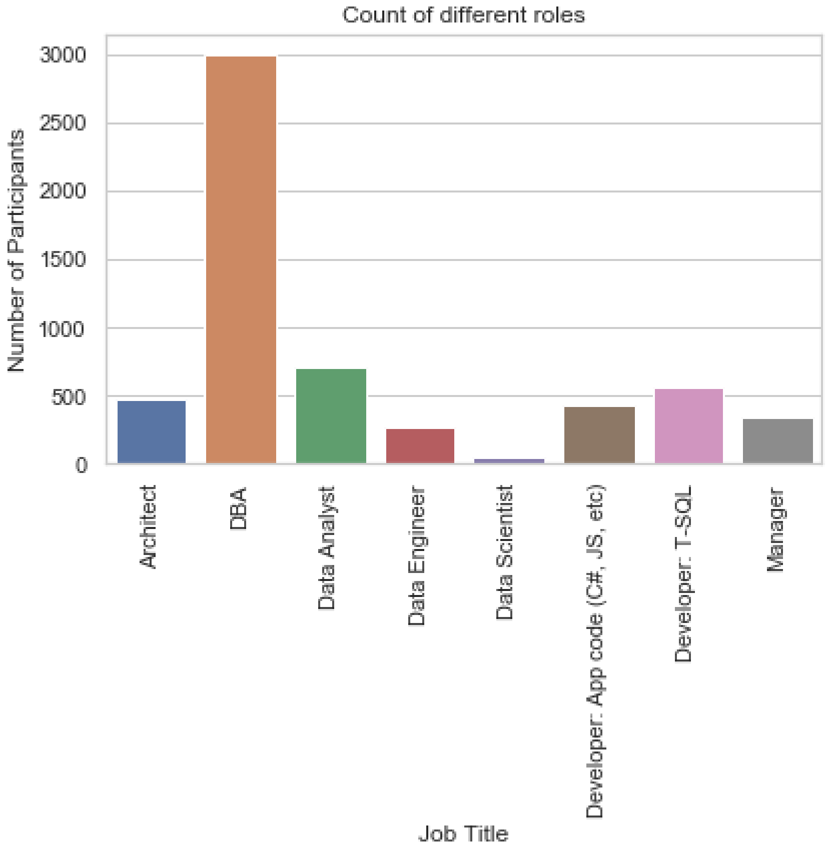
1.4 Roles

Now let's investigate the different roles assumed by IT professionals and how they are distributed. Since we are specifically interested in data science, we will also create a flag for each of the participants to indicate whether his/her role is data-science related.

14. Plot a bar graph depicting the counts of different roles (each bar should represent the count of participants assuming a certain job role).

In [467]:

```
# Your code
# Bar graph depicting the counts of different roles

fun = {'JobTitle' : {'Count' : lambda x: x.count()}}

groupbyrole = a1_dataset.groupby('JobTitle').agg(fun)

groupbyrole = groupbyrole.reset_index()
groupbyrole.columns = groupbyrole.columns.droplevel(0)
groupbyrole.rename(columns = {'':'JobTitle'},inplace = True)

bar_plot = sns.barplot(x = 'JobTitle', y = 'Count', data = groupbyrole)

plt.title('Count of different roles')
plt.xlabel('Job Title', fontsize=12)
plt.ylabel('Number of Participants', fontsize=12)
plt.tight_layout()
bar_plot.set_xticklabels(bar_plot.get_xticklabels(), rotation=90)
plt.show()
```



15. What is the percentage of Data Scientists among the survey respondents?

In [468]:

```
# Your code
# Percentage of Data Scientists

fun = {'JobTitle' : {'Count' : lambda x: x.count()}}

groupbyrole=a1_dataset.groupby('JobTitle').agg(fun)

groupbyrole=groupbyrole.reset_index()
groupbyrole.columns = groupbyrole.columns.droplevel(0)
groupbyrole.rename(columns = {'':'JobTitle'},inplace = True)

# Total number of Data Scientists
total = groupbyrole["Count"].sum()

# Total number of participants
ds_count = 0
for each in a1_dataset['JobTitle']:
    if each == 'Data Scientist':
        ds_count += 1

answer = round ((ds_count/total)*(100), 2)

print("Percentage of Data Scientists : ", answer ,"%")
```

```
Percentage of Data Scientists :  0.83 %
```

Answer

16. Data Scientists usually work closely with specific functions in organisations. Data Analysts and Data Engineers are among the top collaborators with Data Scientists. Since our analysis will now focus on data science roles.

Create a boolean column "DataScienceRelated" which holds if a participant has a job title among "Data Scientist, Data Analyst or Data Engineer."

In [512]:

```
# Your code
# Creating a new column "DataScienceRelated"

a1_dataset['DataScienceRelated'] = ( (a1_dataset['JobTitle'] == 'Data Analyst') |
                                    (a1_dataset['JobTitle'] == 'Data Engineer') |
                                    (a1_dataset['JobTitle'] == 'Data Scientist') )
```

17. What is the percentage of Data Science related roles among the survey participants?

In [513]:

```
# Your code
# Percentage of Data Science related roles

# Number of participants working in Data Science related roles
ds_count = len( a1_dataset[ ( a1_dataset['DataScienceRelated'] == True) ] )

# Total number of participants
total_count = a1_dataset['DataScienceRelated'].count()

answer = round( (ds_count/total_count)*(100), 2)

print("Percentage of Data Science related roles : ", answer, "%")
```

```
Percentage of Data Science related roles :  17.56 %
```

Answer

2. Education

So far, we have seen that there may be some relationships between age, gender and the country that the respondents are from. Next, we should look at what their education is like.

2.1 Formal education

We saw in a recent activity that a significant number of data scientists job advertisements call for a masters degree or a PhD. Let's see if this is a reasonable ask based on the respondent's formal education.

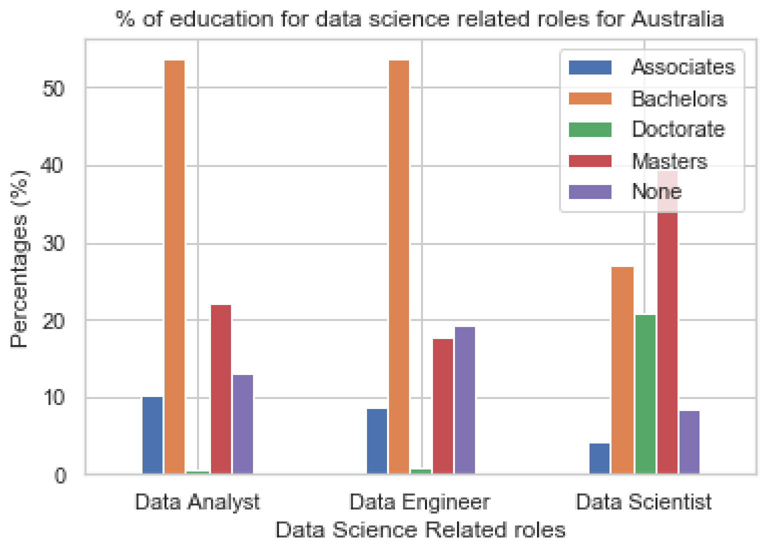
1. Plot a bar chart showing the percentage of each type of education for the three data science related roles.

Hint: You should appropriately label your axes with a legend and a title

In [471]:

```
# Your code

# New Dataset with the Job and Education column
groupbyEdu = a1_dataset[['JobTitle','Education','DataScienceRelated']]
groupbyEdu = groupbyEdu[groupbyEdu.DataScienceRelated == True].reset_index()

# Total number of Data Analysts
da = len(groupbyEdu[(groupbyEdu['JobTitle'] == 'Data Analyst')])
# Total number of Data Engineers
de = len(groupbyEdu[(groupbyEdu['JobTitle'] == 'Data Engineer')])
# Total number of Data Scientists
ds = len(groupbyEdu[(groupbyEdu['JobTitle'] == 'Data Scientist')])

# For Data Analyst roles :
# Number of respondents for each type of education
da_Associates = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Associates (2 years)')])
da_Bachelors = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Bachelors (4 years)')])
da_Doctorate = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Doctorate/PhD')])
da_Masters = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Masters')])
da_None = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'None (no degree completed)')
                         ])

# For Data Engineer roles :
# Number of respondents for each type of education
de_Associates = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Associates (2 years)')])
de_Bachelors = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Bachelors (4 years)')])
de_Doctorate = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Doctorate/PhD')])
de_Masters = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Masters')])
de_None = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'None (no degree completed)')
                         ])

# For Data Science roles :
# Number of respondents for each type of education
ds_Associates = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Associates (2 years)')])
ds_Bachelors = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Bachelors (4 years)')])
ds_Doctorate = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Doctorate/PhD')])
ds_Masters = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Masters')])
ds_None = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'None (no degree completed)')
                         ])
# For Data Analyst roles :
# Percentage of respondents for each type of education
p_da_Associates = round( (da_Associates/da)*100, 2)
p_da_Bachelors = round( (da_Bachelors/da)*100, 2)
p_da_Doctorate = round( (da_Doctorate/da)*100, 2)
```

```
p_da_Masters = round( (da_Masters/da)*100, 2)
p_da_None = round((da_None/da)*100, 2)

# For Data Engineer roles :
# Percentage of respondents for each type of education
p_de_Associates = round( (de_Associates /de)*100, 2)
p_de_Bachelors = round( (de_Bachelors/de)*100, 2)
p_de_Doctorate = round( (de_Doctorate/de)*100, 2)
p_de_Masters = round( (de_Masters/de)*100, 2)
p_de_None = round((de_None/de)*100, 2)

# For Data Scientist roles :
# Percentage of respondents for each type of education
p_ds_Associates = round( (ds_Associates /ds)*100, 2)
p_ds_Bachelors = round( (ds_Bachelors/ds)*100, 2)
p_ds_Doctorate = round( (ds_Doctorate/ds)*100, 2)
p_ds_Masters = round( (ds_Masters/ds)*100, 2)
p_ds_None = round((ds_None/ds)*100, 2)

# Index for the dataframe
roles = ['Data Analyst','Data Engineer','Data Scientist']

# Data for the new dataframe
data = {'Associates': [ p_da_Associates, p_de_Associates, p_ds_Associates ],
        'Bachelors' : [ p_de_Bachelors, p_de_Bachelors, p_ds_Bachelors ],
        'Doctorate' : [ p_da_Doctorate, p_de_Doctorate, p_ds_Doctorate ],
        'Masters' : [ p_da_Masters, p_de_Masters, p_ds_Masters ],
        'None' : [ p_da_None, p_de_None, p_ds_None] }

# creating a new data frame
df = pd.DataFrame(data, index = roles)

# @meseeksmachine. (July 26, 2019). Answer to question: pandas.DataFrame.plot.bar
# Retrieved from: https://pandas.pydata.org/pandas-docs/stable/reference/api
# /pandas.DataFrame.plot.bar.html
# Date accessed: Jan 21, 2020

ax = df.plot.bar(rot=0)

plt.title('% of education for data science related roles for Australia')
plt.ylabel('Percentages (%)',fontsize = 12)
plt.xlabel('Data Science Related roles',fontsize = 12)
plt.show()
```



2. Based on what you have seen, do you think that a Master's or Doctoral degree is too unrealistic for job advertisers looking for someone with data science skills or is it job-dependent?

From the plot, we can see that almost 3% of Data Analysts and 1% of Data Engineers have Masters Degree. However, less than 0.5 % of respondents have a Doctorate/PhD degree. Further, I think that a Masters Degree gives sufficient data science knowledge which is needed for the industry. Moreover, a Doctoral Degree is for those interested in further research, development and teaching. According to me, for jobs in the industry, a candidate will a Masters Degree will be able to meet all the job requirements. Thus, job advertisers looking for someone with data science skills to fill a technical position in the industry can expect a Masters Degree and it's a reasonable ask. Similarly, for jobs in the education sector, along the lines of teaching, the applicants should have a more deeper understanding and knowledge of the subjects. Hence, for recruiting for these sectors, the job advertisers can expect a Doctoral degree.

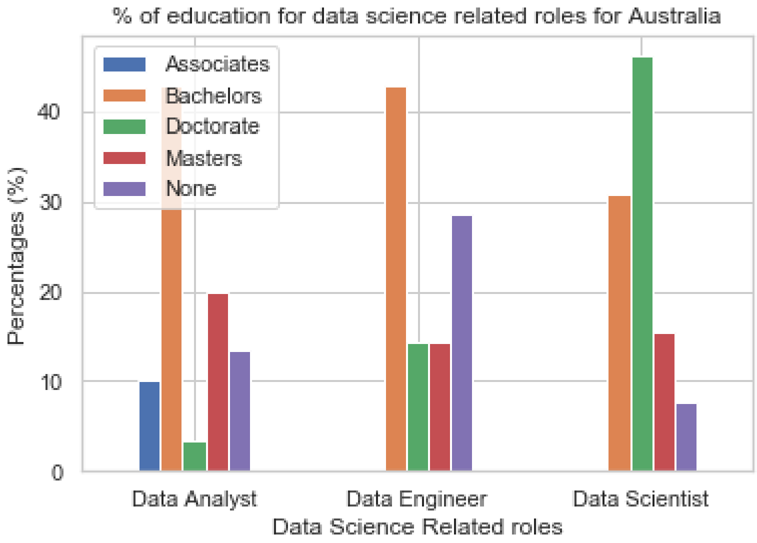
Expecting a Doctoral degree from a candidate applying for a role in the industry is too unrealistic. On the other hand, expecting a Doctoral degree from a candidate applying for a teaching position at University is a reasonable ask. In conclusion, whether the educational qualifications of the candidate will suffice is job dependent.

3. Let's see if the trend is reflected in the Australian respondents.
Plot a bar chart like above but only for Australia, and display the counts of the number of Australian respondents holding a Doctoral degree for each of the three job roles as text output.

In [472]:

```
# your code

total = len(a1_dataset)

# Filtering out the data for Australia
new_ds = a1_dataset[ a1_dataset['Country'] == 'Australia' ]

groupbyEdu = new_ds[['JobTitle','Education','DataScienceRelated']]
groupbyEdu = groupbyEdu[groupbyEdu.DataScienceRelated == True].reset_index()

# Total number of Data Analysts
da = len(groupbyEdu[(groupbyEdu['JobTitle'] == 'Data Analyst')])

# Total number of Data Engineers
de = len(groupbyEdu[(groupbyEdu['JobTitle'] == 'Data Engineer')])

# Total number of Data Scientists
ds = len(groupbyEdu[(groupbyEdu['JobTitle'] == 'Data Scientist')])

# For Data Analysts
# Count of each type of education
da_Associates = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Associates (2 years)')])
da_Bachelors = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Bachelors (4 years)')])
da_Doctorate = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Doctorate/PhD')])
da_Masters = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'Masters')])
da_None = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Analyst') &
                                (groupbyEdu['Education'] == 'None (no degree completed)')
                         ])

# For Data Engineers
# Count of each type of education
de_Associates = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Associates (2 years)')])
de_Bachelors = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Bachelors (4 years)')])
de_Doctorate = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Doctorate/PhD')])
de_Masters = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'Masters')])
de_None = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Engineer') &
                                (groupbyEdu['Education'] == 'None (no degree completed)')
                         ])

# For Data Scientists
# Count of each type of education
ds_Associates = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Associates (2 years)')])
ds_Bachelors = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Bachelors (4 years)')])
ds_Doctorate = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Doctorate/PhD')])
ds_Masters = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'Masters')])
ds_None = len( groupbyEdu[ (groupbyEdu['JobTitle'] == 'Data Scientist') &
                                (groupbyEdu['Education'] == 'None (no degree completed)')
                         ])

# For Data Analysts
# Percentage of each type of education
p_da_Associates = round( (da_Associates/da)*100, 2)
p_da_Bachelors = round( (da_Bachelors/da)*100, 2)
p_da_Doctorate = round( (da_Doctorate/da)*100, 2)
p_da_Masters = round( (da_Masters/da)*100, 2)
p_da_None = round((da_None/da)*100, 2)

# For Data Engineers
# Percentage of each type of education
p_de_Associates = round( (de_Associates /de)*100, 2)
p_de_Bachelors = round( (de_Bachelors/de)*100, 2)
p_de_Doctorate = round( (de_Doctorate/de)*100, 2)
p_de_Masters = round( (de_Masters/de)*100, 2)
p_de_None = round((de_None/de)*100, 2)

# For Data Scientists
# Percentage of each type of education
p_ds_Associates = round( (ds_Associates /ds)*100, 2)
p_ds_Bachelors = round( (ds_Bachelors/ds)*100, 2)
p_ds_Doctorate = round( (ds_Doctorate/ds)*100, 2)
p_ds_Masters = round( (ds_Masters/ds)*100, 2)
p_ds_None = round((ds_None/ds)*100, 2)


# Index for the new dataframe
roles = ['Data Analyst','Data Engineer','Data Scientist']

# Data for the new dataframe
data = {'Associates': [ p_da_Associates, p_de_Associates, p_ds_Associates ],
        'Bachelors' : [ p_de_Bachelors, p_de_Bachelors, p_ds_Bachelors ],
        'Doctorate' : [ p_da_Doctorate, p_de_Doctorate, p_ds_Doctorate ],
        'Masters' : [ p_da_Masters, p_de_Masters, p_ds_Masters ],
        'None' : [ p_da_None, p_de_None, p_ds_None] }

# Creating a new dataframe
df = pd.DataFrame(data, index = roles)

# @meseeksmachine. (July 26, 2019). Answer to question: pandas.DataFrame.plot.bar
# Retrieved from: https://pandas.pydata.org/pandas-docs/stable/reference/api
# /pandas.DataFrame.plot.bar.html
# Date accessed: Jan 21, 2020

ax = df.plot.bar(rot=0)

plt.title('% of education for data science related roles for Australia')
plt.ylabel('Percentages (%)',fontsize=12)
plt.xlabel('Data Science Related roles',fontsize=12)
plt.show()

new_ds = new_ds1[[ 'JobTitle','Education','DataScienceRelated'] ]
new_ds = new_ds[ new_ds['DataScienceRelated'] == True ]

new_ds = new_ds.groupby( ['Education','JobTitle']).count()
new_ds = new_ds.reset_index()
new_ds = new_ds.rename( columns = {'DataScienceRelated':'Count'})


#Australian respondents holding a Doctoral degree for the three job roles
```

```
# @Anton Protopopov (April 18, 2016). Answer to question:
# Extract column value based on another column pandas dataframe.
# Retrieved from: https://stackoverflow.com/questions/36684013/
# extract-column-value-based-on-another-column-pandas-dataframe
# Date accessed: Jan 17, 2020.

new_ds3 = new_ds[new_ds['Education'] == 'Doctorate/PhD']

da = new_ds3.loc[ new_ds3['JobTitle'] == 'Data Analyst','Count'].iloc[0]
de = new_ds3.loc[ new_ds3['JobTitle'] == 'Data Engineer','Count'].iloc[0]
ds = new_ds3.loc[ new_ds3['JobTitle'] == 'Data Scientist','Count'].iloc[0]

print("Australian respondents holding a doctoral degree for :")
print("Data Analyst role :", da)
print("Data Engineer role :", de)
print("Data Scientist role :", ds)
```



```
Australian respondents holding a doctoral degree for :
Data Analyst role : 1
Data Engineer role : 1
Data Scientist role : 6
```

4. Display as text output the mean and median age of ALL respondents according to each degree type.

In [473]:

```
# Your code

mean_age = a1_dataset.groupby('Education').mean()
median_age = a1_dataset.groupby('Education').median()

print(" \n The mean ages of each type of degree:")
print(" For :-")
print(" Associates is     : " , round(mean_age.Age.iloc[0],2))
print(" Bachelors is      : " , round(mean_age.Age.iloc[1],2))
print(" Doctrates/PhD is  : " , round(mean_age.Age.iloc[2],2))
print(" Masters is        : " , round(mean_age.Age.iloc[3],2))
print(" None is           : " , round(mean_age.Age.iloc[4],2))

print("\n The Median ages of each type of degree:")
print(" For :-")
print(" Associates is     : " , round(median_age.Age.iloc[0],2))
print(" Bachelors is      : " , round(median_age.Age.iloc[1],2))
print(" Doctrates/PhD is  : " , round(median_age.Age.iloc[2],2))
print(" Masters is        : " , round(median_age.Age.iloc[3],2))
print(" None is           : " , round(median_age.Age.iloc[4],2))
```

```
 
 The mean ages of each type of degree:
 For :-
 Associates is     :  26.45
 Bachelors is      :  26.68
 Doctrates/PhD is  :  31.36
 Masters is        :  27.52
 None is           :  26.16

 The Median ages of each type of degree:
 For :-
 Associates is     :  24
 Bachelors is      :  24
 Doctrates/PhD is  :  29
 Masters is        :  25
 None is           :  24
```

3. Employment

Many of you will be seeking work after your degree. Let's have a look at the state of the employment market for the respondents of the survey.

Let's have a look at the data.

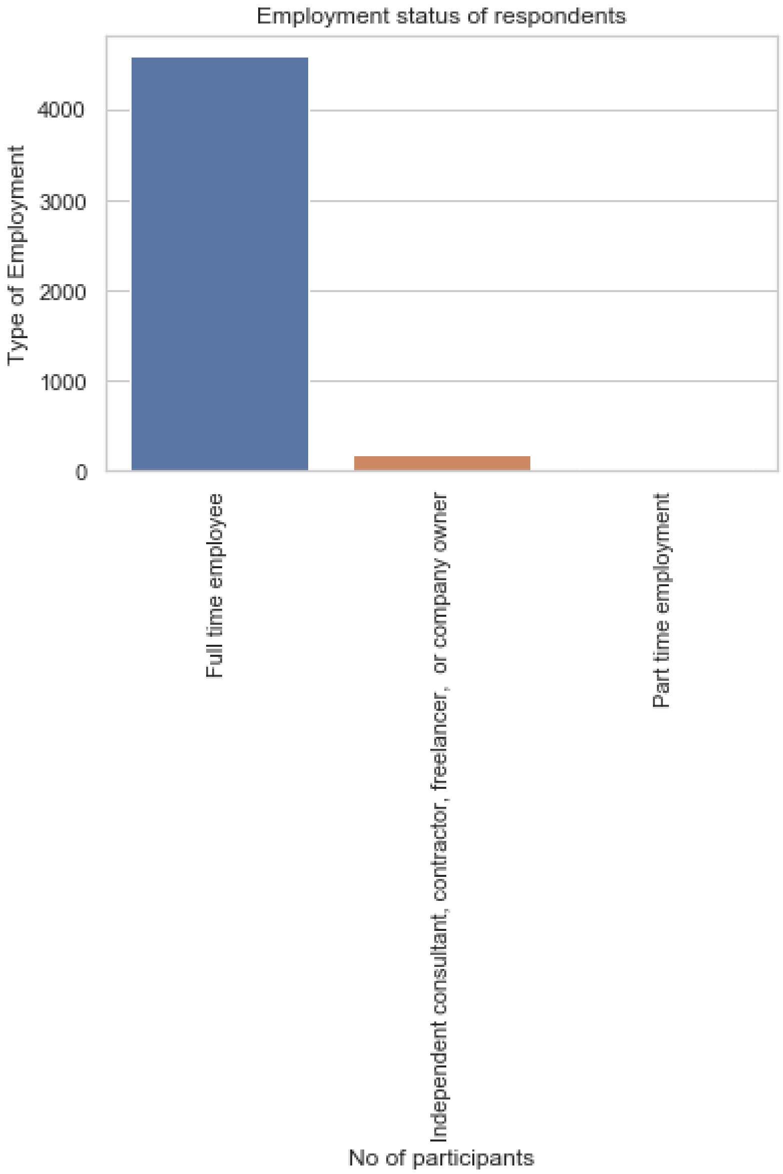
3.1 Employment status

The type of employment will affect the salary of a worker. Those employed part-time will likely earn less than those who work full time.

1. Plot the type of employment the respondents have on a bar chart for respondents who do not assume data science related roles.

In [514]:

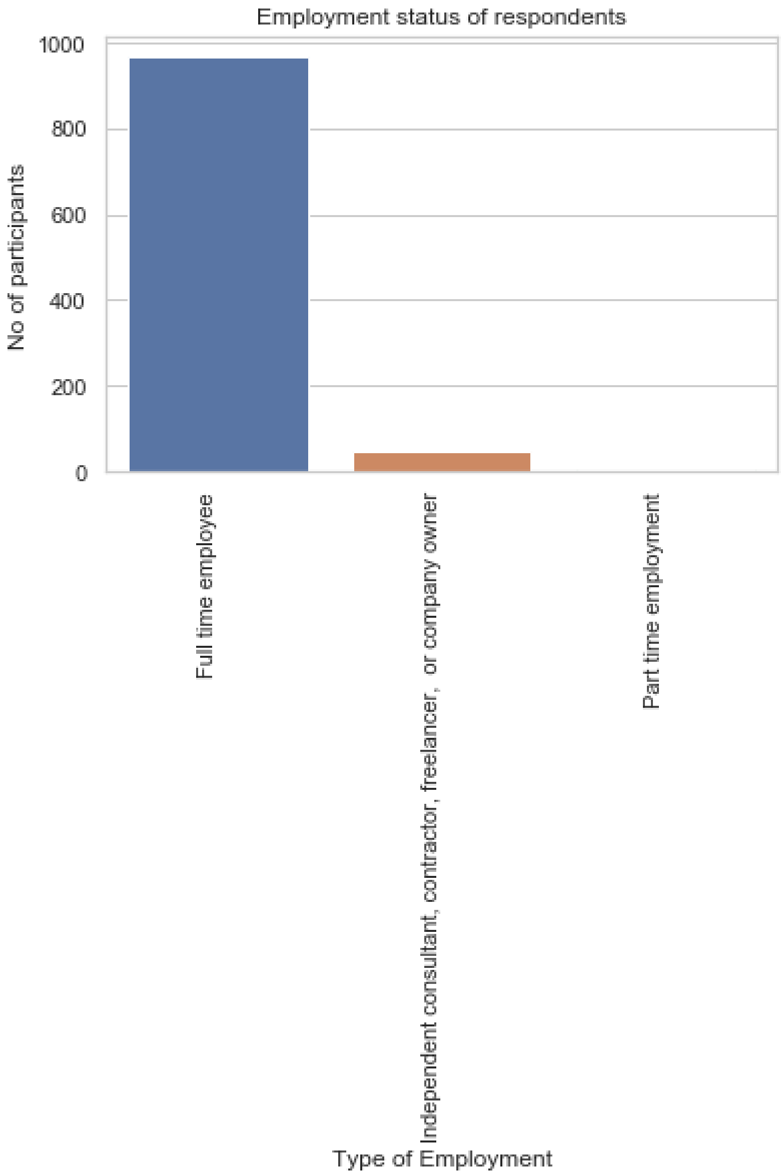
```
# Your code

# Filtering out Non Data Science related roles
non_ds_dataset = a1_dataset[a1_dataset['DataScienceRelated'] == False]

# Creating a new dataframe
new_df = non_ds_dataset.groupby('EmploymentStatus').count().reset_index()

# Plotting the barchart
ax = sns.barplot(y = new_df.Gender, x = new_df.EmploymentStatus)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90)

plt.title('Employment status of respondents')
plt.ylabel('Type of Employment')
plt.xlabel('No of participants')
plt.show()
```



2. Now plot the type of employment the respondents have on a bar chart only for those assuming data science related roles

In [475]:

```
# Your code

# Filtering out only Data Science related roles
ds_dataset = a1_dataset[a1_dataset['DataScienceRelated'] == True]

# Creating a new dataframe
new_df = ds_dataset.groupby('EmploymentStatus').count().reset_index()

ax = sns.barplot(x = new_df.EmploymentStatus,y = new_df.Gender)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90, ha = "right")

plt.title('Employment status of respondents')
plt.xlabel('Type of Employment')
plt.ylabel('No of participants')
plt.show()
```



3. Comparing the two graphs, would you say that the data science roles differ in the type of employment as opposed to non-data science roles?
Explain your answers.

Comparing the two graphs, we understand that the data science roles do not differ in the type of employment as opposed to non data science roles.

In both cases, majority of the respondents are full time employed, followed by a few independent contractors and the number of respondents involved in part time jobs are nearly negligible when compared to the full time employees.

However, though the type of employment is the same, the number in each category differs.

*) While considering the data science related roles, there are more than 4000 employees employed full time, less than 1000 independent contractors and negligible part time workers.

*) While considering the non data science roles, there are nearly 900 employees employed in the full time sector, less than 200 employees working as independent contractors and negligible employees in the part time field.

4. Let's investigate whether the type of employment is country dependent.
Print out the percentages of all respondents who are employed full time in Australia, United Kingdom and the United States.

In [476]:

```
# Your code

# @Roman. (May 24, 2013). Answer to question:
# How to get a value from a cell of a dataframe?.
# Retrieved from: https://stackoverflow.com/questions/16729574/
# how-to-get-a-value-from-a-cell-of-a-dataframe.
# Date accessed: Jan 14,2020.

total = a1_dataset['Country'].count()

countries = ['United States', 'United Kingdom','Australia']

country_df = a1_dataset[a1_dataset.Country.isin(countries)]

emp_df = country_df[country_df.EmploymentStatus == 'Full time employee']
emp_df = emp_df.groupby('Country').agg({'EmploymentStatus':{'emp_count':'count'}})
emp_df = emp_df.reset_index()
emp_df.columns = emp_df.columns.droplevel(0)
emp_df.columns = ['Country','emp_count']

# Calculating the percentages
australia = round (( (emp_df.iat[0,1])/ (total) * 100 ), 2)
uk = round( ( (emp_df.iat[1,1])/ (total) * 100 ), 2)
usa = round( ( (emp_df.iat[2,1])/ (total) * 100 ), 2)

print('Percentage of respondents employed full time in Australia :',australia,"%")
print('Percentage of respondents employed full time in United Kingdom :',uk,"%")
print('Percentage of respondents employed full time in United States :',usa,"%")
```

```
Percentage of respondents employed full time in Australia : 2.51 %
Percentage of respondents employed full time in United Kingdom : 9.31 %
Percentage of respondents employed full time in United States : 64.59 %
```

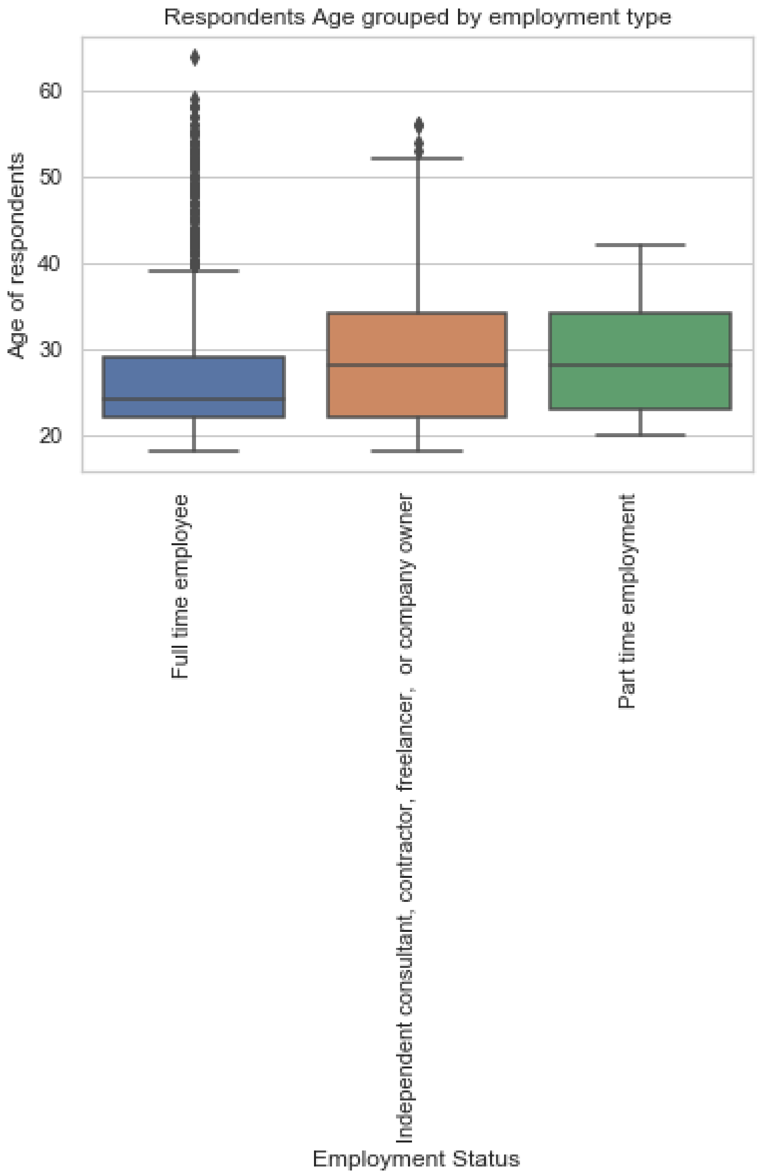
Remember earlier, we saw that age seemed to have some interesting characteristics when plotted with other variables.

Let's find out the median age of employees by type of employment.

5. Plot a boxplot of the respondents age, grouped by employment type.

In [477]:

```
# Your code

ax = sns.boxplot(x = 'EmploymentStatus', y = 'Age', data = a1_dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90, ha = "right")

plt.title('Respondents Age grouped by employment type')
plt.xlabel('Employment Status')
plt.ylabel('Age of respondents')
plt.show()
```



6. What are your observations?

From the boxplot, we can deduce that the age of most of the full time employees is between 20 to 30 years of age.

Similarly, most of the respondents doing part time jobs and working as independent contractors are aged between 25 to 35 years.

The minimum age of full time employees and independent contractors is less than 20 years while that of part time employees is around 20 years.

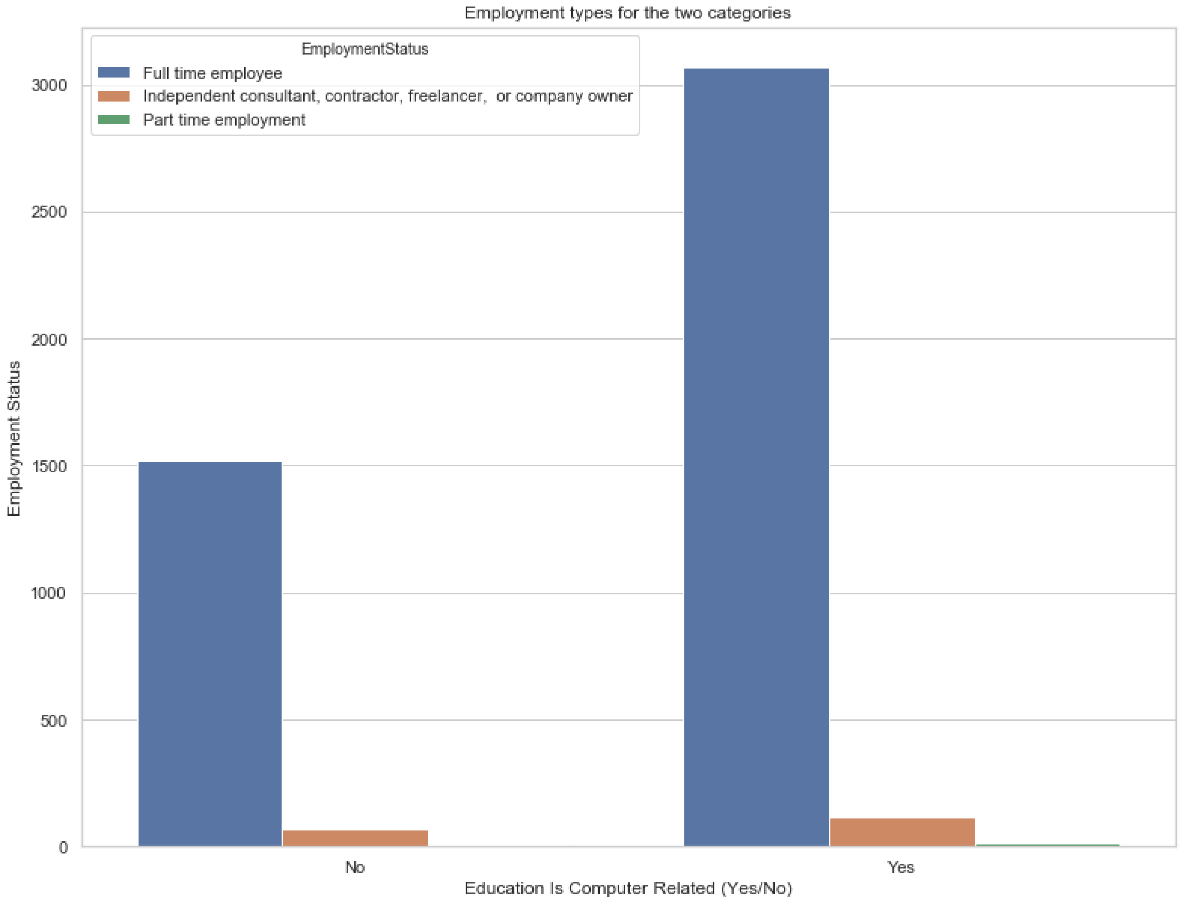
There are many outliers present in the box plot of full time employees, some in that of independent contractors and zero in that of part time employees.

This explains why the maximum age of full time employees, independent contractors and part time employees is above 60, 50 and 40 years of age respectively.

7. You may be wondering if a relevant Computer degree is necessary to help gain full-time employment after graduation.
Plot the respondents' employment types (for all respondents) for each of the two categories of "EducationIsComputerRelated".

In [479]:

```
# Your code

# Michael Waskom, (2012 - 2018). Answer to question: seaborn.countplot
# Retrieved from: https://seaborn.pydata.org/generated/seaborn.countplot.html
# Date accessed: Jan 17, 2020.

plt.figure(figsize=(13,10))

# Generating the count plot
ax = sns.countplot(x = 'EducationIsComputerRelated', data = a1_dataset,
                   hue = 'EmploymentStatus')

plt.title('Employment types for the two categories')
plt.ylabel('Employment Status')
plt.xlabel('Education Is Computer Related (Yes/No)')
plt.show()
```



8. Looking at the graph, does holding a computer-related degree improves your chances of securing a full-time job?
Explain your answers.

Yes, holding a computer related degree significantly improves the chances of securing a full time job.

Reason :

**) From the above graph, nealy 3000 respondents having a computer related degree are employed in a full time job.

**) Whereas, the number of respondents with a non computer related degree employed in a full time job is around 1500.

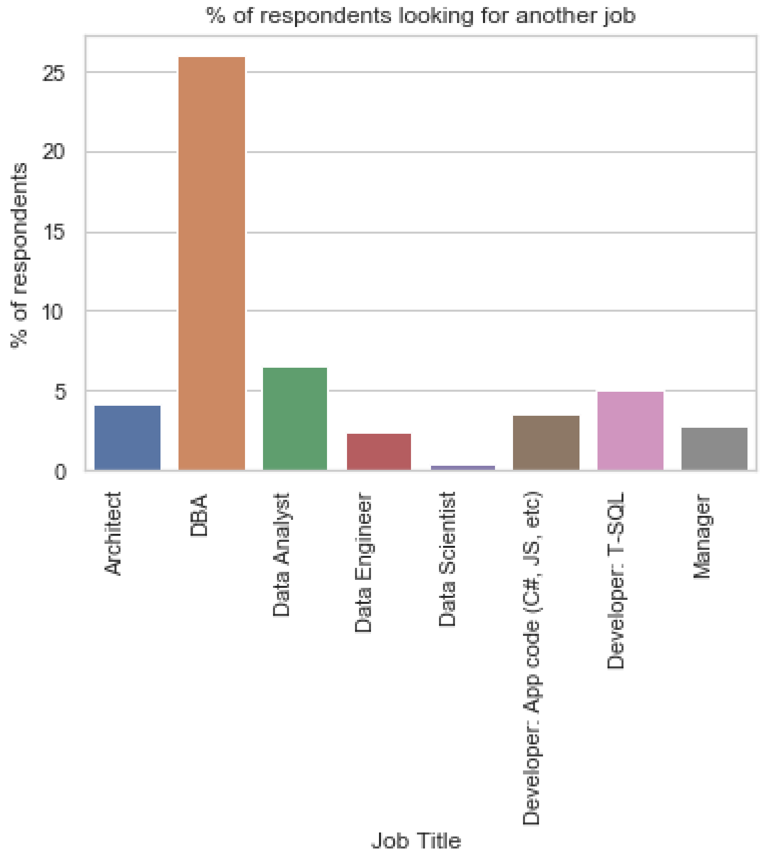
3.2 Job Satisfaction

Let's now investigate how happy IT professionals are about their jobs. It is also relevant to look at the years of experience to see whether the job gets boring after a while.

9. Create a bar chart for the percentage of respondents who are looking for another job grouped by the different job titles.

In [504]:

```
# Your code

total = a1_dataset['JobTitle'].count()

another_job = a1_dataset[ a1_dataset['LookingForAnotherJob'] == 'Yes' ]

fun = {'JobTitle':{'emp_count':'count'}}
groupbyJobTitle = another_job.groupby('JobTitle').agg(fun)
groupbyJobTitle = groupbyJobTitle.reset_index()
groupbyJobTitle.columns = groupbyJobTitle.columns.droplevel(0)
groupbyJobTitle.columns = ['JobTitle','emp_count']

# Plotting the percentage bar chart
ax = sns.barplot( x = groupbyJobTitle.JobTitle,
                  y = (groupbyJobTitle.emp_count/total) * 100 )
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90, ha = "right")

plt.title('% of respondents looking for another job')
plt.xlabel('Job Title ')
plt.ylabel('% of respondents')
plt.show()
```



10. What are the two roles that have the highest and lowest percentage of employees looking for other jobs?

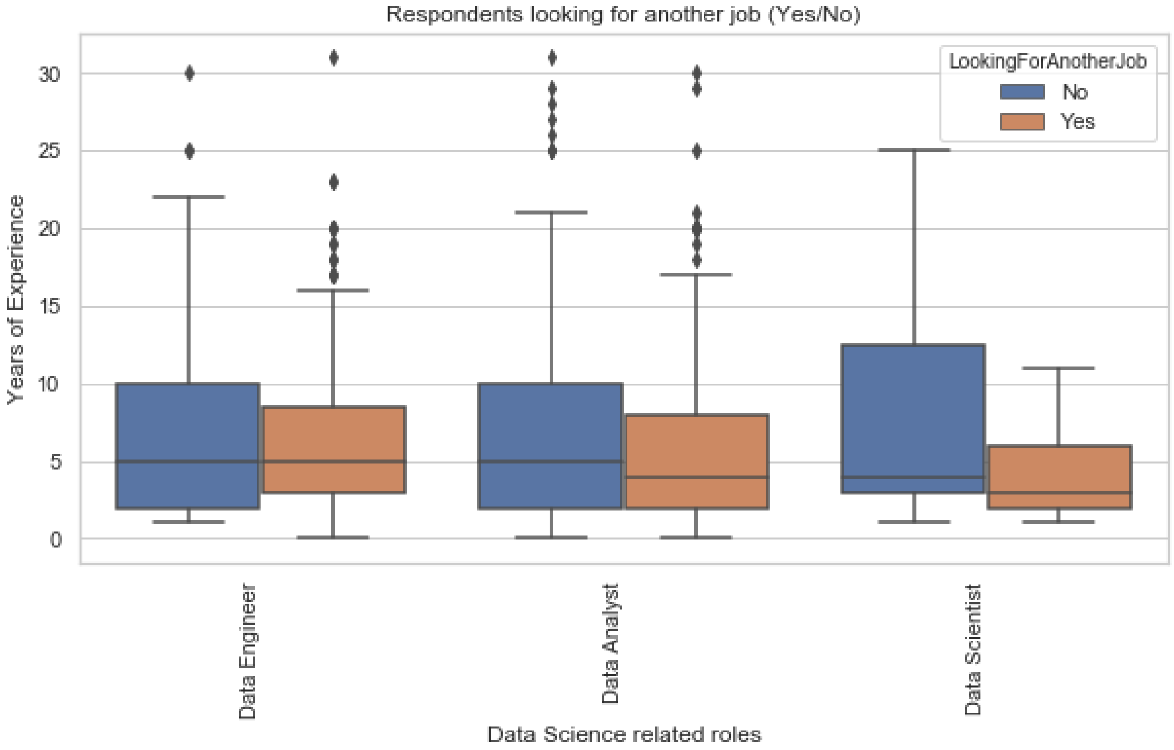
Roles which have the highest percentage of employees : DBA (Greater than 25%)

Roles which have the lowest percentage of employees : Data Scientist (Less than 1%)

11. Let's focus on data science-related roles. Plot a box plot depicting the distribution of years-of-experience of those respondents who are looking for another job versus those who are not for each of the three roles.

In [481]:

```
# Your code

# Getting the required values
ds_dataset = a1_dataset[ a1_dataset['DataScienceRelated'] == True]

# Plotting the boxplot
plt.figure(figsize=(10,5))
ax = sns.boxplot(x ='JobTitle', y = 'YearsofExperience',
                 hue = 'LookingForAnotherJob', data = ds_dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90, ha = "right")

plt.title('Respondents looking for another job (Yes/No)')
plt.xlabel('Data Science related roles')
plt.ylabel('Years of Experience')
plt.show()
```



12. What can you say about the years of experience as to whether it impacts happiness?

From the above three boxplots, we understand that the boxplots follow a similar pattern for all the three roles. For all the three Data Science related role, we can say that the work experience of respondents not looking for another job is greater than the respondents looking for another job.

For respondents working as Data Engineers, respondents not looking for an another job have around 2.5 to 10 years of work experience. On the other hand, respondents looking for an another job have less than 10 years of work experience.

The same pattern follows for respondents working as a Data Analysts. Likewise, for respondents working as Data Scientists, the respondents not looking for another jobs have a work experience of 2.5 to 12.5 years and the respondents looking for another jobs have a work experience of around 2.5 to slightly above 5 years. Thus, only in the role of Data Scientists, we could see a significant change in the number of years of work experience.

Thus, in conclusion, respondents having more years of experience are satisfied and happy with their existing job . These respondents are not looking for an another jobs. Similarly, the respondents with comparatively lesser number of experience seem to be not happy and satisfied with their jobs. These respondents are looking for another jobs.

4. Salary

Data science is considered a very well paying role and was named 'best job of the year' for 2019.

We would like to investigate in this section the different salary ranges for the different job roles in the IT industry and compare it to those of Data Science roles.

4.1 Salary overview

Note that the salaries given in the dataset is in USD. If we are to investigate the salaries in AUD, we need to consider the currency conversion.

You can use the following rate of conversion:

```
1 USD = 1.47 AUD
```

Let's have a look at the data.

1. Create a derived column "SalaryAUD" containing the converted salary data into Australian Dollars (AUD).
Print out the maximum and median salary in AUD for each of the job roles in our dataset.

In [516]:

```
# Creating a derived column "SalaryAUD"
a1_dataset['SalaryAUD'] = a1_dataset['SalaryUSD']*(0.68027211)

# Max and Median salary grouped by job title

max = a1_dataset.groupby('JobTitle', as_index = True)['SalaryAUD'].max()
median = a1_dataset.groupby('JobTitle', as_index = True)['SalaryAUD'].median()

print("Maximum salary in AUD for each of the job roles :")
print(round(max,2))

print("***********************************************")

print("Median salary in AUD for each of the job roles :")
print(round(median,2))
```

```
Maximum salary in AUD for each of the job roles :
JobTitle
Architect                          238095.24
DBA                                653061.23
Data Analyst                       289115.65
Data Engineer                      442176.87
Data Scientist                     108843.54
Developer: App code (C#, JS, etc)    131972.79
Developer: T-SQL                   479591.84
Manager                            427793.88
Name: SalaryAUD, dtype: float64
***********************************************
Median salary in AUD for each of the job roles :
JobTitle
Architect                          81632.65
DBA                                61224.49
Data Analyst                       52380.95
Data Engineer                      64625.85
Data Scientist                     75510.20
Developer: App code (C#, JS, etc)    54421.77
Developer: T-SQL                   57823.13
Manager                            74829.93
Name: SalaryAUD, dtype: float64
```

2. Do those figures confirm that data scientists are well paid?

Yes, these figures confirm that data scientists are well paid.

From the above data, we can clearly see that the median salary for Data Scientists is around 75,000 AUD and the maximum salary is around 108,843 AUD.Thus, most of the salaries are nearing the median values - around 75,000 AUD.

Also, while comparing the maximum salaries of all the role, though the maximum salary is around 653,061 AUD for the DBA role, the median salary of Data Scientist is greater than that of the DBA role.

Thus, these figures clearly confirm that Data Scientits are well paid.

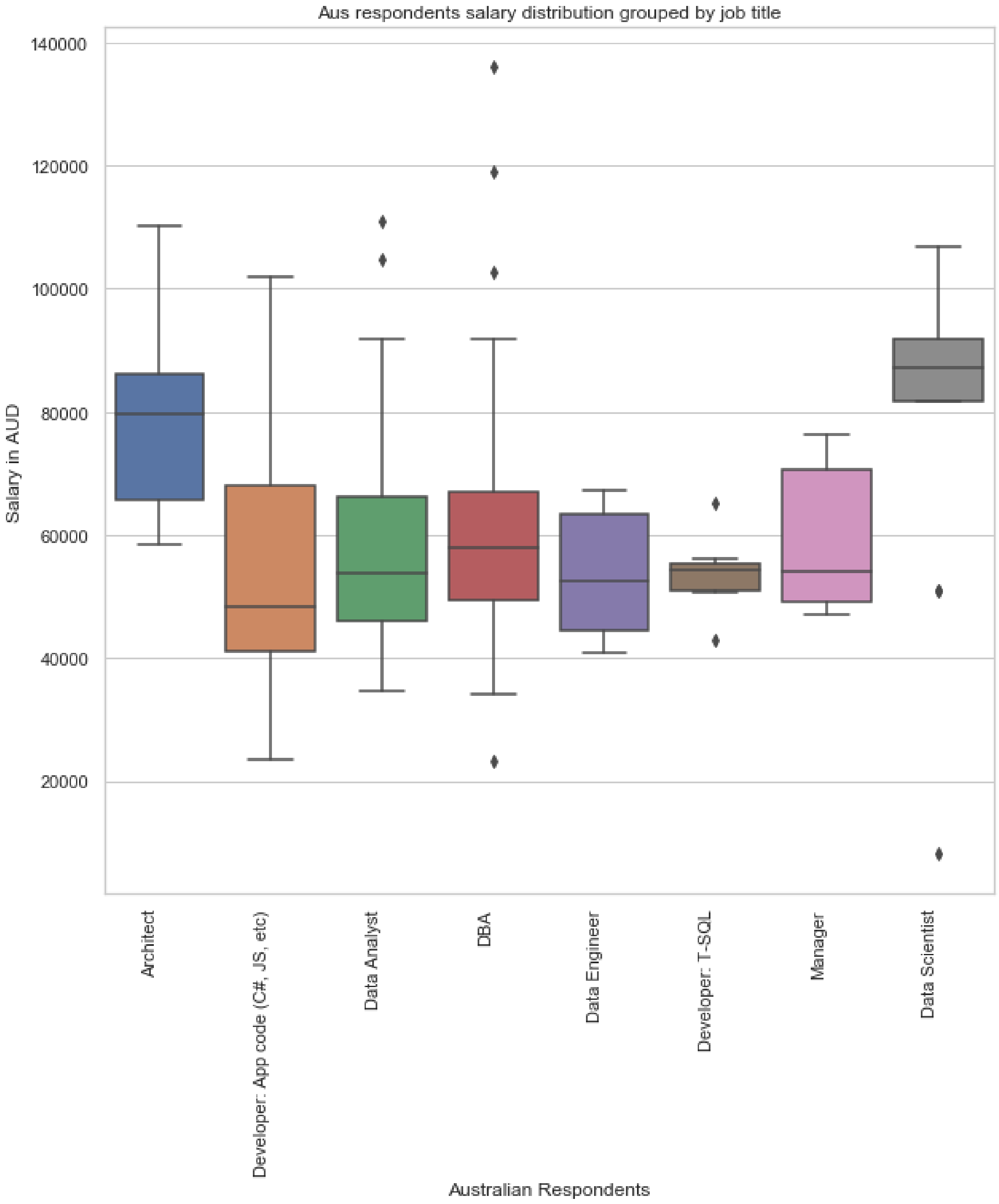
4.2 Salary by country

Since each country has different cost of living and pay indexes, we want to compare these jobs only in Australia.

3. Plot boxplot chart of the Australian respondents salary distribution grouped by the different job titles.

In [517]:

```
# Your code

# How to sort Australian respondents Country = Australia
new_dataset = a1_dataset[a1_dataset.Country == 'Australia']

# Plotting the box plot
sns.set(style = "whitegrid")
plt.figure(figsize=(10,10))
ax = sns.boxplot(x = 'JobTitle', y = 'SalaryAUD', data = new_dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90, ha = "right")


plt.title('Aus respondents salary distribution grouped by job title')
plt.xlabel('Australian Respondents',fontsize = 12)
plt.ylabel('Salary in AUD',fontsize = 12)
plt.show()
```



4. How are data scientists paid in comparison to other roles in Australia?

Yes, Data Scientists are indeed well paid in comparison to other roles in Australia.

1) From the boxplot, it is clear that the salaries of most Data Scientists are between 80,000 to 100,000 AUD with the median value at around 90,000 AUD.

2) Though the maximum salaries of some roles like DBA go much greater than the maximum salary of Data Scientists, most of the salaries for DBA are between 40,000 to 70,000 AUD.

3) Also, when we compare the majority salary ranges of all the roles, they are the most for Architect roles, around 65,000 AUD to 85,000 AUD which is still lesser than the majority salary range of Data Scientists.

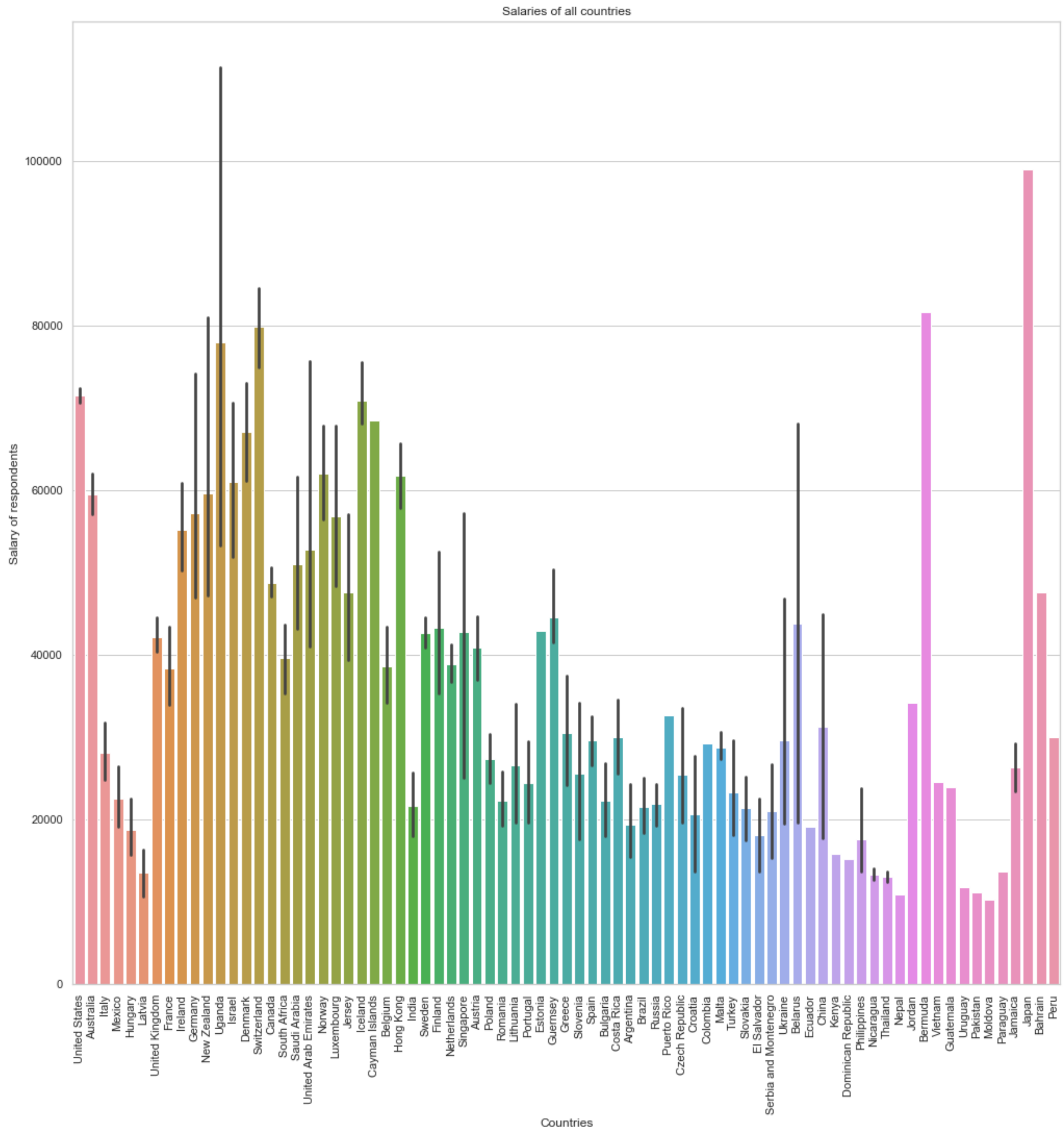
Thus, Data Scientists are paid significantly quite higher when compared to other roles.

5. Australia's salaries look pretty good in general. Is that the case for all other countries?
Plot the salaries of all countries on a bar chart (with error bars).

Hint: Consider all job titles and filter for full-time employees only

In [518]:

```
# Your code

# Filter Full time employees
new_dataset = a1_dataset[a1_dataset.EmploymentStatus == 'Full time employee']

plt.figure(figsize=(17,17))

ax = sns.barplot(y = 'SalaryAUD', x = 'Country', data = new_dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90)

plt.title('Salaries of all countries')
plt.xlabel('Countries')
plt.ylabel('Salary of respondents')
plt.show()
```



6. What do you notice about the distributions? What do you think is the cause of this?

We notice the following things from the distributiion :

1) The distribution of salary is uneven and varied.

2) Salaries in developed countries like United States, United Kingdom, Australia and Germany is much higher salaries than salaries in developing countries like India.

3) Though India is also among the top employers of the world, salaries in India are way below than that present in most developed countries.

Cause for this uneven and varied distribution is :

* Low currency rate

* Increasing population

* Poor infrastructure

* Inefficient use of human resources

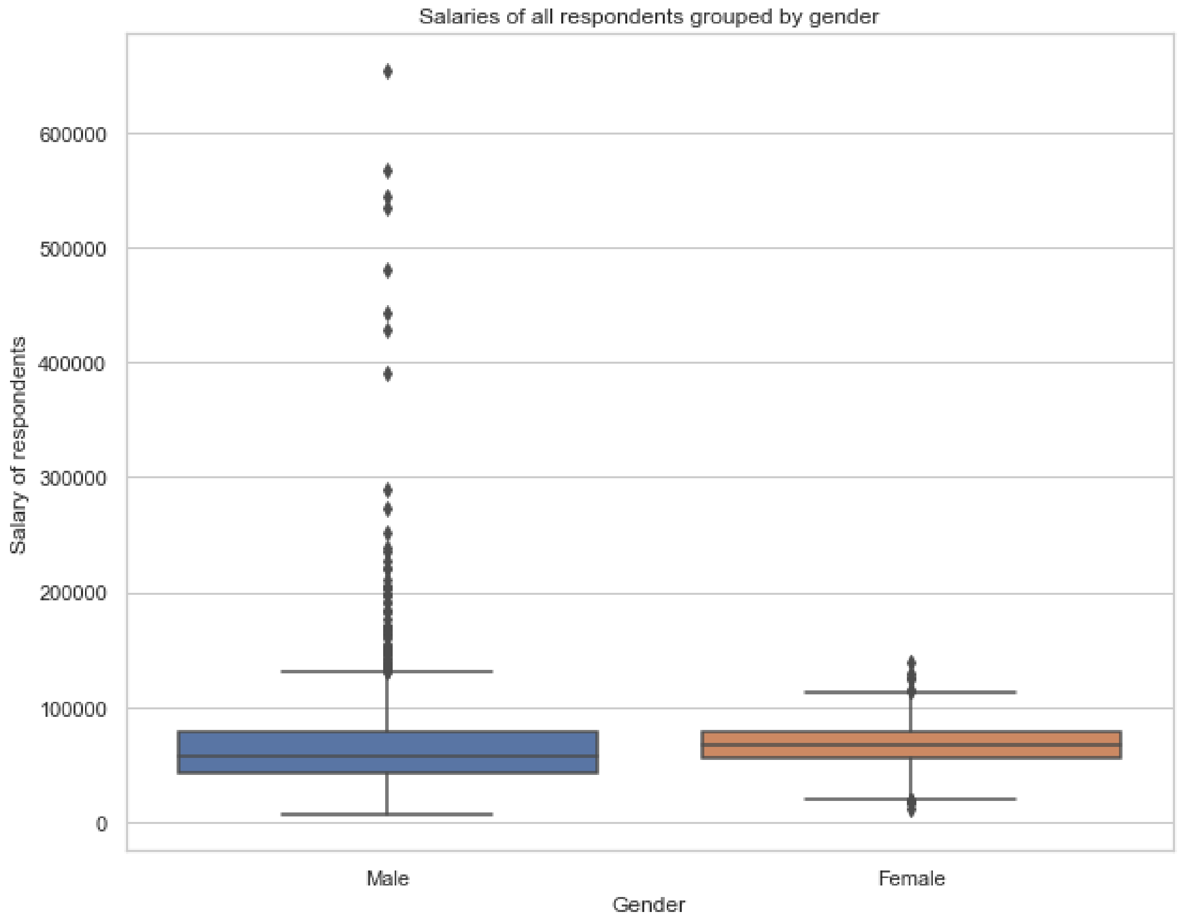
4.3 Salary and Gender

The gender pay gap in the tech industry is a big talking point. Let's see if the respondents are noticing the effect.

7. Plot the salaries of all respondents grouped by gender on a boxplot.

In [519]:

```
# Your code
plt.figure(figsize=(10,8))

ax = sns.boxplot(x = 'Gender', y= 'SalaryAUD', data=a1_dataset)

plt.title('Salaries of all respondents grouped by gender')
plt.xlabel('Gender')
plt.ylabel('Salary of respondents')
plt.show()
```



Salaries of all respondents grouped by gender

8. What do you notice about the distributions?

We notice the following things from the distribution :

1) There are more number of male respondents who have salaries in the range of 0 to 10000 AUD than female respondents.

2) The highest salary of male respondents is slightly higher than the highest salary of female respondents.

3) The lowest salary of female respondents is greater than the lowest salary of male respondents.

4) The median salary of female respondents is greater than the median salary of male respondents.

5) Finally, the boxplot for male employees has a larger number of outliers than the boxplot for female employees.

9. The salaries may be affected by the country the respondent is from. In Australia, the weekly difference in pay between men and women is 17.7%, and in the United States it is 26%.
Print the median salaries of Australia, United States and India grouped by gender.

In [486]:

```
# Your code

countries = ['Australia','United States','India']
df = a1_dataset[a1_dataset.Country.isin(countries)]

fun = {'SalaryAUD':{'Median Salary':'median'}}
df = df.groupby(['Country','Gender']).agg(fun).reset_index()
df.columns = df.columns.droplevel(0)
df.columns =['Country','Gender','Median Salary']
df
```

```
C:\Users\Gayatri Aniruddha\Anaconda3\lib\site-packages\pandas\core\groupby\g
eneric.py:1315: FutureWarning: using a dict with renaming is deprecated and
will be removed in a future version
  return super(DataFrameGroupBy, self).aggregate(arg, *args, **kwargs)
```

Out[486]:

	Country	Gender	Median Salary
0	Australia	Female	64625.850450
1	Australia	Male	56462.585130
2	India	Female	22278.911603
3	India	Male	15850.340163
4	United States	Female	68306.122565
5	United States	Male	71428.571550

4.4 Salary and formal education

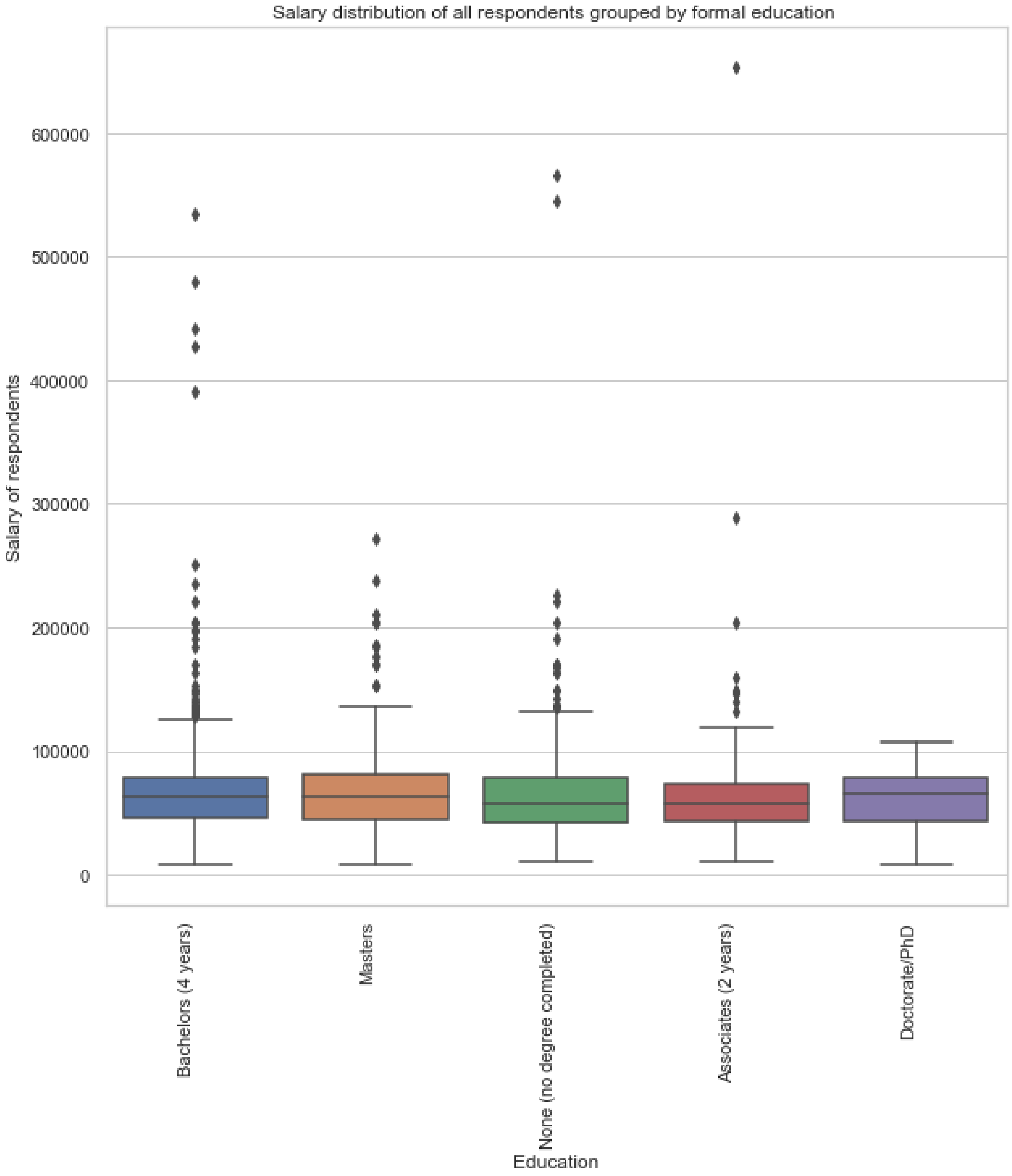
Is getting your master's really worth it ? Do PhDs get more money?

Let's see.

10. Plot the salary distribution of all respondants and group by formal education type on a boxplot.

In [505]:

```
# Your code

plt.figure(figsize=(10,10))
ax = sns.boxplot(x = 'Education', y = 'SalaryAUD', data = a1_dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90, ha = 'right')

plt.title('Salary distribution of all respondents grouped by formal education ')
plt.xlabel('Education')
plt.ylabel('Salary of respondents')
plt.show()
```



11. Is it better to get your Masters or PhD?
Explain your answer.

It is better to get Masters done - for better salaries!

Reason :

1) From the boxplot, we can see that the salaries for employees with Masters and salaries for employees with PhDs lie between 50,000 to 100,000 AUD.

2) Though the median salary for employees with Masters is lesser than the median salary for employees with PhDs, the whisker for Masters is longer than the whisker for PhDs.

3) A longer whisker indicates that there are more employees with Masters having salaries higher than employees with PhDs.

Thus, it's better to do Masters as more number of employees who have done Masters have higher salaries than employees with PhDs.

4.5 Salary and Employment Sector

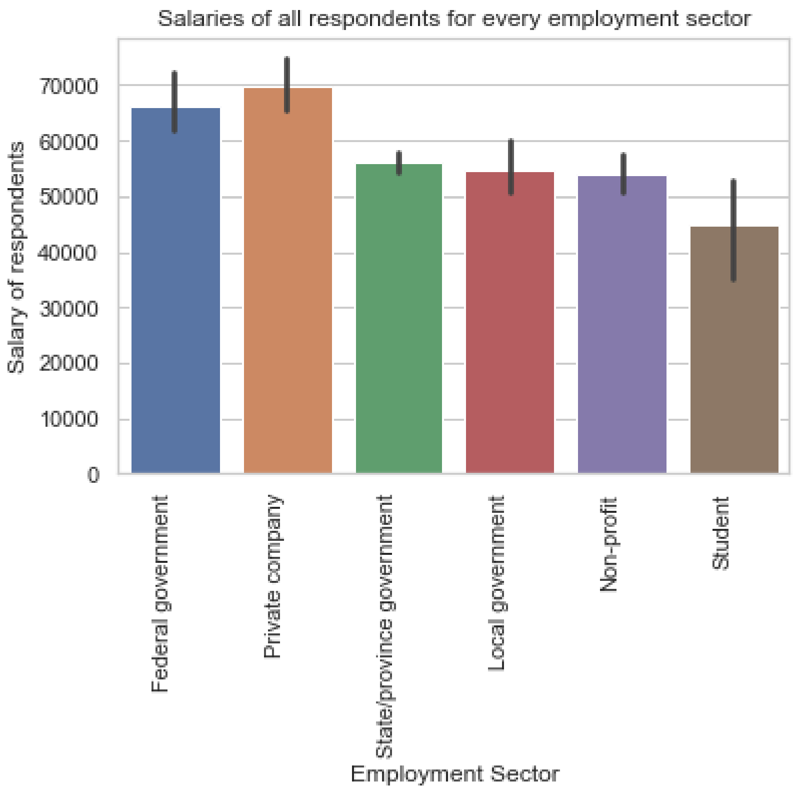
Do government jobs pay better than private sector? Does it differ based on the country?

Let's see.

12. Plot a bar chart (with error bars) of the salaries of respondents for each of the employment sectors.

In [488]:

```
# Your code

ax = sns.barplot(y = 'SalaryAUD', x = 'EmploymentSector', data = a1_dataset)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90, ha = 'right')

plt.title('Salaries of all respondents for every employment sector')
plt.xlabel('Employment Sector')
plt.ylabel('Salary of respondents')
plt.show()
```



13. Which seems to be the highest paying sector overall?
Do you think it would differ based on the country?
Propose a method to find out and explain your answer.

From the barplot, private companies seem to be the highest paying sector overall.

Yes, the highest paying sector would differ based on the country. The highest paying sector in each country is dependent on each country's resources. It is also dependent on how each country effectively utilizes it's human resources. It will also depend on the popularity and opportunities provided by each of these sectors. The country's economy and development would also affect it's highest paying sector.

Thus, in order to find out the highest paying sector in each country, we will have to plot a graph of salaries and sectors for each of the countries separately. This will also throw light on how each employment sector is thriving in each of the different countries.

5. Predicting salary

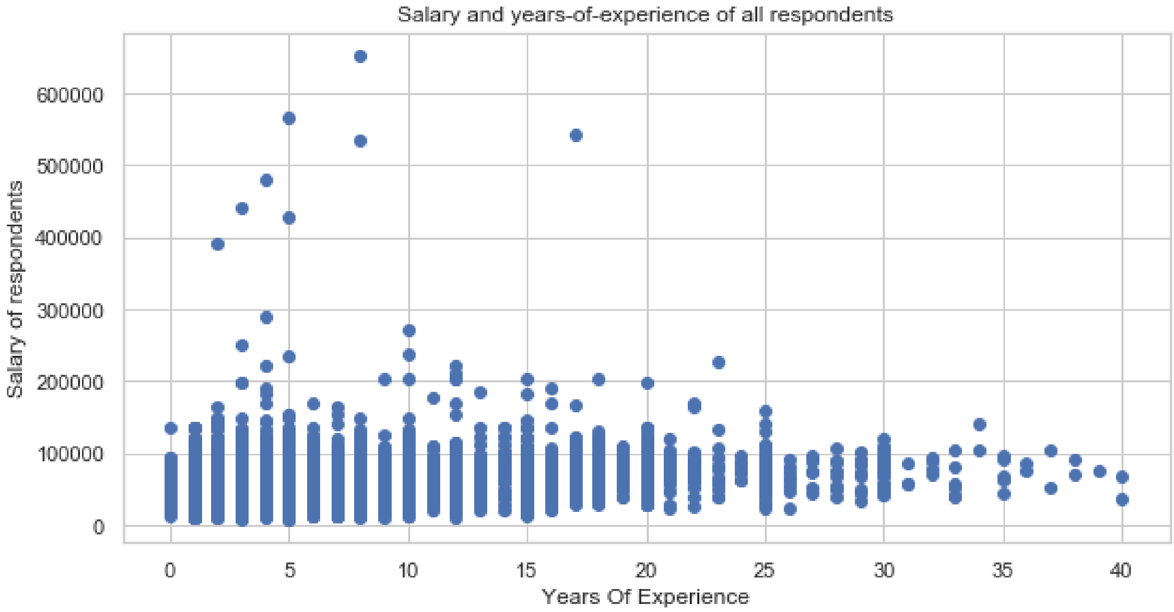
We have looked at many variables and seen that there are a lot of factors that could affect your salary.

Let's say we wanted to reduce it; one method we could use is a linear regression. This is a basic but powerful model that can give us some insights. Note though, there are more robust ways to predict salary based on categorical variables. But this exercise will give you a taste of predictive modelling.

1. Plot the salary and years-of-experience of respondants on a scatterplot.

In [489]:

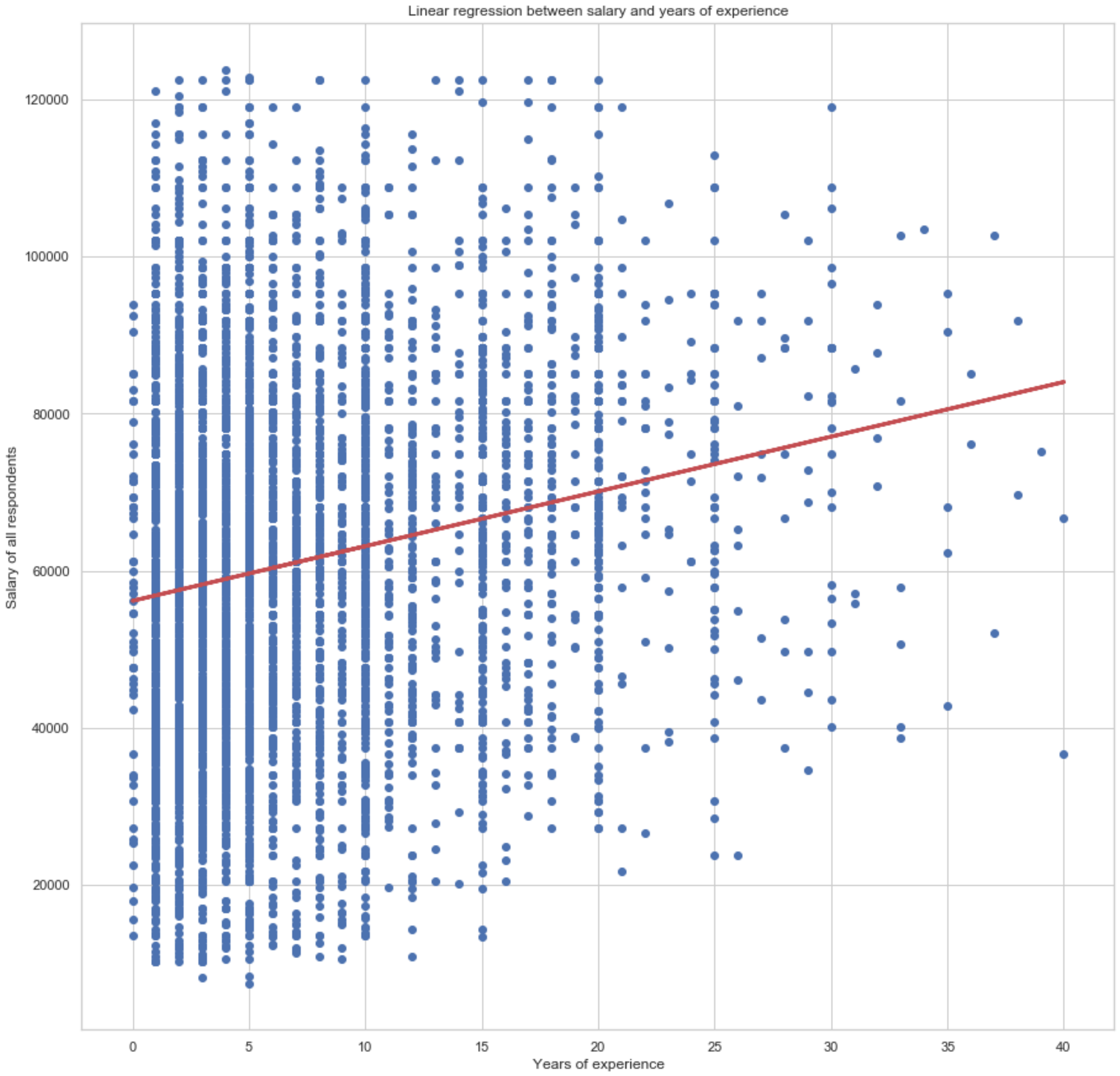
```
# Your code
plt.figure(figsize = (10,5))
sns.set(style = "whitegrid")
plt.scatter(a1_dataset['YearsofExperience'], a1_dataset['SalaryAUD'])

#plt.plot(a1_dataset.YearsOfExperience,a1_dataset.SalaryAUD,'.b')
plt.title('Salary and years-of-experience of all respondents')
plt.xlabel('Years Of Experience')
plt.ylabel('Salary of respondents')
plt.show()
```



2. Let's refine this.
Remove Salary outliers using 2-sigma rule and then create a linear regression between the salary and years-of experience of full-time respondents.
Plot the linear fit over the scatterplot.

In [490]:

```
#Your code

fulltime = a1_dataset[a1_dataset['EmploymentStatus'] == 'Full time employee']
salary = a1_dataset['SalaryAUD']
mean = a1_dataset['SalaryAUD'].mean()
sd = a1_dataset['SalaryAUD'].std()

# @Punit Jajodia(2020). Answer to question:
# Remove Outliers Using Normal Distribution and S.D.
# Retrieved from: https://www.kdnuggets.com/2017/02/
# removing-outliers-standard-deviation-python.html
# Date accessed: Jan 21, 2020.

# Removing outliers
a1_dataset['outliers'] = ( (a1_dataset['SalaryAUD'] > (mean - 2 * sd))
                          & (a1_dataset['SalaryAUD'] < (mean + 2 *sd)) )

final_outliers = a1_dataset[a1_dataset['outliers'] == True]
plt.figure(figsize = (15,15))

# Linear Regression between salary and years-of-experience
from scipy.stats import linregress
slope,intercept,r_value,p_value,std_err = linregress(final_outliers['YearsofExperience'],
                                                     final_outliers['SalaryAUD'])

line = [slope*x + intercept for x in final_outliers['YearsofExperience']]

# Plotting the line
plt.plot(final_outliers['YearsofExperience'], line, 'r-',linewidth = 3)

# Plotting the Scatter Plot
plt.scatter(final_outliers['YearsofExperience'],final_outliers['SalaryAUD'])

plt.title('Linear regression between salary and years of experience')
plt.xlabel('Years of experience')
plt.ylabel('Salary of all respondents')
plt.show()
```



3. Do You think that this is a good way to predict salaries?
Explain your answer.

Yes, this is a good way to predict salaries.

Reason: From the plot, it is clear that the linear line is covering most of the scattered dots. Thus, we are getting a good indication of all the possible values. This is the line of best fit and we are able to minimise the errors using this method of linear regression. As a result, we will be able to see the future values in the best possible manner.

6. Tasks and tools

You might be wondering (or not) what different tasks you will be assigned in a data science role and what kind of tools would you be using the most?

In this section, we perform necessary text processing to investigate such aspects.

6.1 Data science common tasks

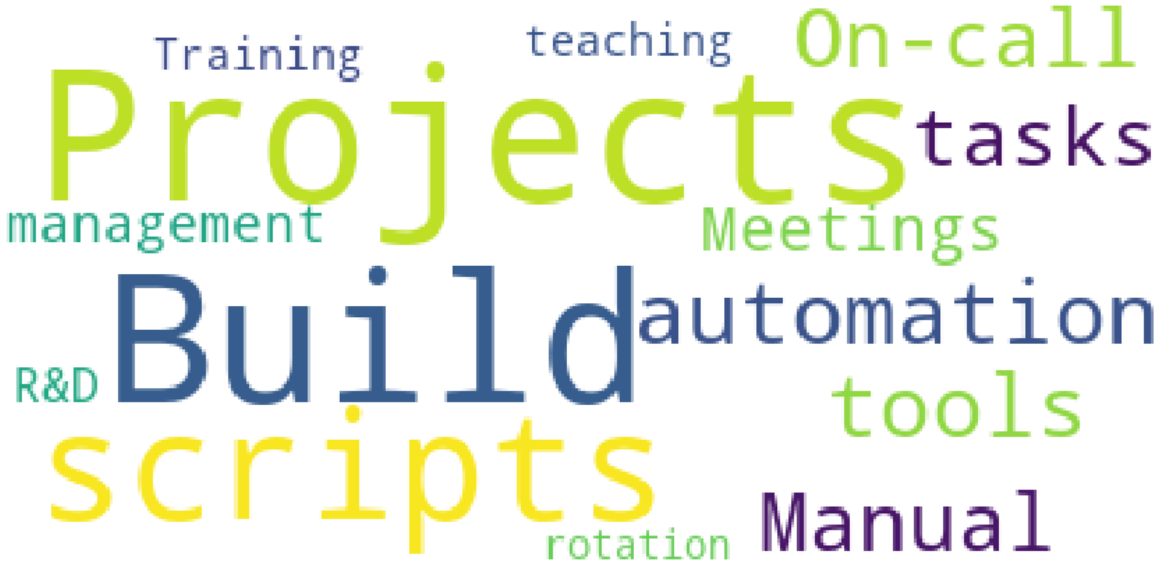
We focus here on the three data science job roles and investigate the tasks usually carried out in such roles.

1. Investigate the 'KindsOfTasksPerformed' column and perform the required text processing to enable you to plot a word cloud depicting the frequency of the different tasks.

In [491]:

```
# Your code

from wordcloud import WordCloud, STOPWORDS

tasks_performed = str( list( a1_dataset['KindsOfTasksPerformed'].values))
text = tasks_performed

# Cleaning the text for better presentation of the word cloud
text = text.replace("["," ")
text = text.replace("]"," ")
text = text.replace("nan"," ")
text = text.replace("\n"," ")
text = text.replace(","," ")
text = text.replace("'"," ")
text = text.replace("&","")
text = text.replace("RD","R&D")

text = text.split(",")
text = str(text)

# Cleaning the text for better presentation of the word cloud
text = text.replace("Training/teaching", "Training teaching")
text = text.replace("']", "")
text = text.replace("['", "")
text = text.replace("24/7/365", " ")

# @HackerRank. Answer to question: collections.Counter()
# Generate word cloud from single-column Pandas dataframe.
# Retrieved from: https://www.hackerrank.com/challenges/
# collections-counter/problem
# Date accessed: Jan 21, 2020.

# Counter to count the number of occurences
from collections import Counter
dict_of_words = Counter(text.split(' '))

# Further cleaning the text
# This is for deleting : On-call "as a part of"
del dict_of_words['of']
del dict_of_words['part']
del dict_of_words['a']
del dict_of_words['as']
del dict_of_words[' ']

stopwords = list(STOPWORDS)

# @cmc_carlos. (March 27, 2017). Answer to question:
# Wordcloud Python with generate_from_frequencies
# Retrieved from: https://stackoverflow.com/questions/43043437/
# wordcloud-python-with-generate-from-frequencies
# Date accessed: Jan 21, 2020.

# @Languitar. (April 25, 2015). Answer to question:
# Generate word cloud from single-column Pandas dataframe.
# Retrieved from: https://stackoverflow.com/questions/43606339/
# generate-word-cloud-from-single-column-pandas-dataframe
# Date accessed: Jan 15, 2020.

# Generating the word cloud
```

```
wordcloud = WordCloud(stopwords = stopwords,
                      background_color='white').generate_from_frequencies(dict_of_words)

# Generating the plot
plt.figure(figsize=(15,10))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```



6.2 Data Science Common Tools

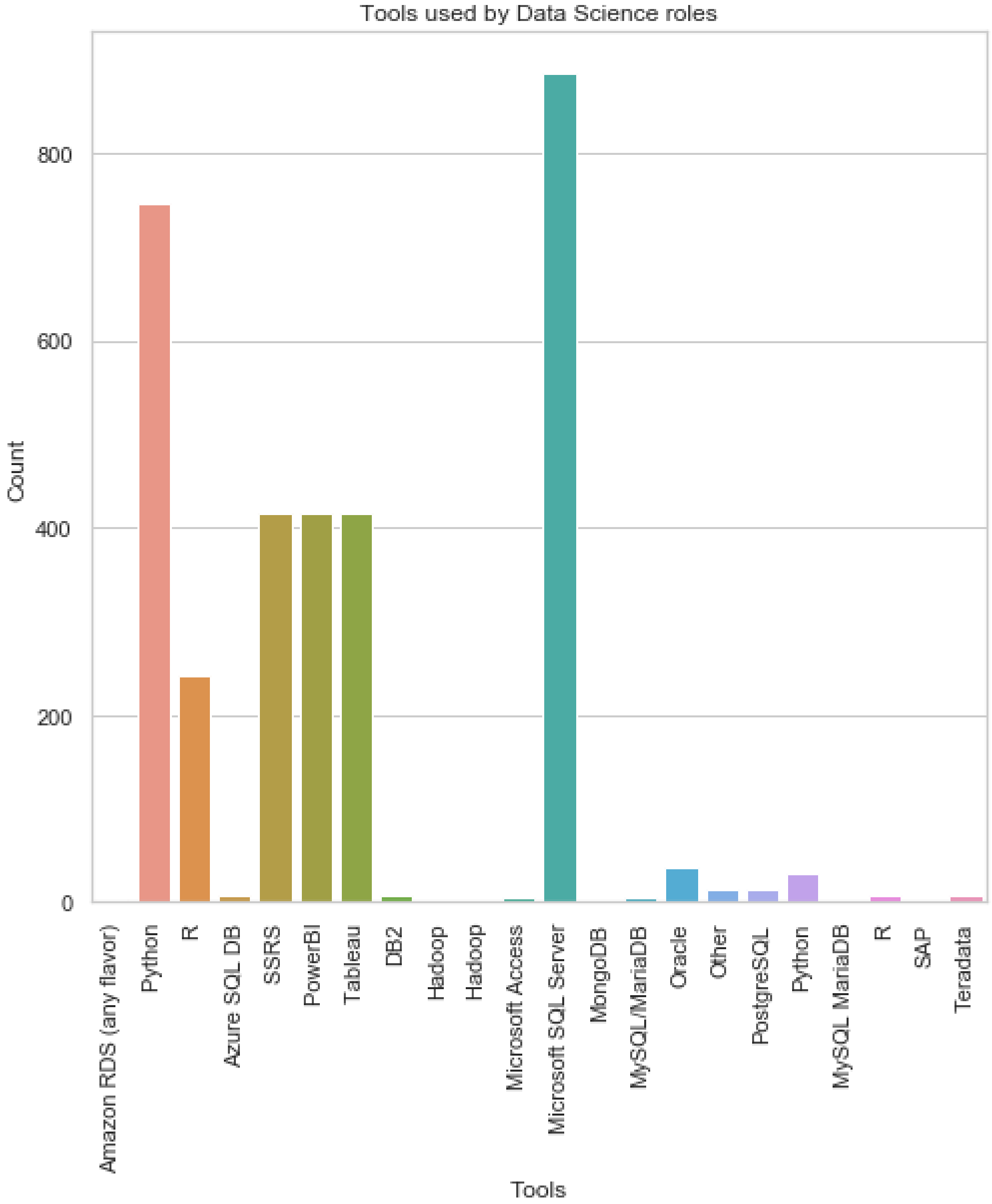
Now we compare the skillset required by data science roles and other IT roles.

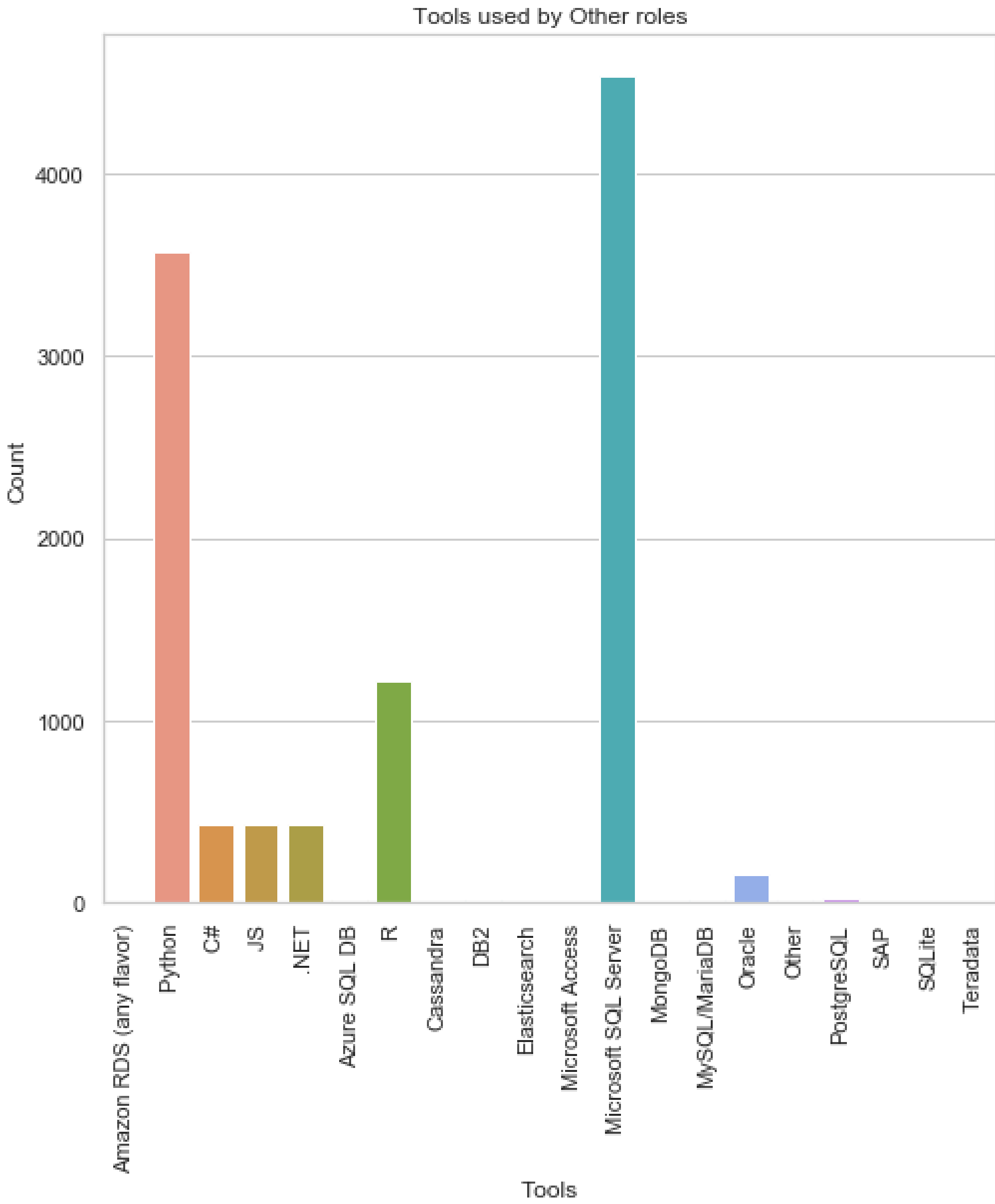
2. Filter your respondents based on DataScienceRelated flag and plot two seperate bar charts depicting the tools used by data science roles versus other roles.

Hint: You will need to do similar text processing to the previous task.

In [492]:

```
# Your code

# Filtering  Dataframe based on DS Flag
# ds_roles
ds_roles = a1_dataset[ ( a1_dataset['DataScienceRelated'] == True) ]

# other_roles
other_roles = a1_dataset[ ( a1_dataset['DataScienceRelated'] == False) ]

ds = {}
other = {}

tools = list(a1_dataset['ToolsUsed'])
ds_tools = list(ds_roles['ToolsUsed'])
other_tools = list(other_roles['ToolsUsed'])


# Creating a dictionary of all tools used by data science roles
for i in ds_tools:
    for each in i.split(","):
        if each in ds.keys():
            ds[each] += 1
        else:
            ds[each] = 1

# Creating a dictionary of all tools used by other roles
for i in other_tools:
    for each in i.split(","):
        if each in other.keys():
            other[each] += 1
        else:
            other[each] = 1

# Creating a new dataframe for count of tools used in ds and non ds roles
ds_frame = pd.DataFrame(list(ds.items()), columns = ['Tools', 'Count'])
others_frame = pd.DataFrame(list(other.items()), columns = ['Tools', 'Count'])

# Barplots for the Data Science Roles
plt.figure(figsize = (8,8))
ax = sns.barplot(y = 'Count', x = 'Tools', data = ds_frame)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90)
plt.title('Tools used by Data Science roles')
plt.xlabel('Tools')
plt.ylabel('Count')
plt.show()

# Barplots for the Other Roles
plt.figure( figsize = (8,8) )
ax = sns.barplot(y = 'Count', x = 'Tools', data = others_frame)
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90)
plt.title('Tools used by Other roles')
plt.xlabel('Tools')
plt.ylabel('Count')
plt.show()
```


Tools used by Data Science roles
Count
0
200
400
600
800
Tools
Amazon RDS (any flavor)
Python
R
Azure SQL DB
SSRS
PowerBI
Tableau
DB2
Hadoop
Hadoop
Microsoft Access
Microsoft SQL Server
MongoDB
MySQL/MariaDB
Oracle
Other
PostgreSQL
Python
MySQL MariaDB
R
SAP
Teradata



3. What do you think are the most commonly used tools for a data science role?

Most commonly used tools for a data science role :

```
1. Microsoft SQL Server
2. Python
3. SSRRS, PowerBI and Tableau
```

From the bar chart, it is clear that the most commonly used tools for a data science role are Microsoft SQL Server, followed by Python. Microsoft SQL Server is used by more than 900 respondents. Python is the next most used tool with almost more than 650 employees using it. The next in the line are SSRRS, PowerBI and Tableau with more than a 400 respondents using it. Other tool used is R with just above 200 respondents using it.

7. Data quality assessment

' Garbage in, garbage out'.

The saying means that poor quality data will return unreliable and often conflicting results. In this task, you need to assess your data set critically and understand not just what its use means for the outcome of your analysis, but also how those insights inform decisions which lead to broader effects.

1. Now that you have analysed the data. Go into the data set file and determine two anomalies. These could be parts of the data that don't seem quite right or logically can't co-exist. Write a paragraph about these explaining what part of your analysis alerted you to them, why they are anomalies, why they may exist, and what could be done to fix them.

While analyzing the data set, I was keen on the Data Science related roles. Hence, I filtered out all the Data Science related roles. Then, I was interested in the educational backgrounds and the tasks performed by respondents with a non computer science background. That's where I observed my first anomaly.The first data anomaly is that even for the respondents with a non computer background, the tools used are computer related like Python, R, Tableau etc. Also, some of the tasks performed by respondents without a computer related education like building scripts and automation tools is computer related. This anomaly exists because in most companies, job related training is given to employees. As a result, employees with a non computer science related background use computer related tools and are able to perform computer related tasks. In order to fix this anomaly, we can create one more column named - "ComputerRelatedTrainingTaken". This will give us a rough idea of respondents with a non computer science background doing computer related tasks because of the training provided.

Another data anomaly that I observed was between Age and YearsofExperience. Respondents aged 22 years have 2 years of work experience with a Bachelors Degree. This does'nt seem logically right to co-exist. As, we complete our Bachelors Degree at the age of 22. Hence, how can someone completing his degree at 22 years have 2 years years of work experience. This can exist if the work experience is part-time or if the respondent has skipped grades. However, here it still is an anomaly to me because the category of work experience is not specifically mentioned. Thus, in order to overcome this anomaly, we can add another column - "YearsofFullTimeExperience". This will give us the full time experience of respondents which will explain the "actual" full time work experience at a required age.

Well done! You have completed the assignment!

For reassurance, the Australian 2019 Graduate Outcomes Survey found the median salary for Masters graduates in Computer Science and Information Systems for was AUD 92,900 for full-time employment.