imports

In [1]:
```
#import neccessary packages
import pandas as pd
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import seaborn as sns
import re
```

In [2]:
```
#set some of the display options
pd.set_option('display.max_columns', 250)
pd.set_option('display.max_colwidth', None)
```

Read and Clean the data

In [3]:
```
#read clean_kaggle_data file
kaggle_data = pd.read_csv('clean_kaggle_data.csv')

# check top 5 rows
kaggle_data.head()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3
063: DtypeWarning: Columns (74,78,115,147,154,172,176,213,225,229,232) have m
ixed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[3]:

	Unnamed: 0	Time from Start to Finish (seconds)	Q1	Q2	Q2_OTHER_TEXT	Q3	Q4	Q5	Q5_OTHE
0	0	510	22-24	Male	-1	France	Master’s degree	Software Engineer	
1	1	423	40-44	Male	-1	India	Professional degree	Software Engineer	
2	3	391	40-44	Male	-1	Australia	Master’s degree	Other	
3	4	392	22-24	Male	-1	India	Bachelor’s degree	Other	
4	5	470	50-54	Male	-1	France	Master’s degree	Data Scientist	

Rename the columns

In [4]:
```
#rename first column to id
kaggle_data.rename(columns={'Unnamed: 0':'id'}, inplace = True)

#set the id as index
kaggle_data.set_index('id', inplace = True)
```

In [5]:
```
#map the question prompt to keywords
question_mapping = {'Time from Start to Finish (seconds)': 'survey_duration',
                    'Q1':'age',
                    'Q2':'gender',
                    'Q3':'country',
                    'Q4':'education',
                    'Q5':'job_title',
                    'Q6':'company_size',
                    'Q7':'DS_team_size',
                    'Q8':'is_ML_used',
                    'Q9':'job_activities',
                    'Q10':'salary_USD',
                    'Q11':'money_spent_ML_cloud',
                    'Q12':'DS_info_source',
                    'Q13':'DS_learning_platform',
                    'Q14':'DS_primary_tool',
                    'Q15':'DS_coding_experience',
                    'Q16':'IDE',
                    'Q17':'notebook',
                    'Q18':'programming_language',
                    'Q19':'recommended_programming_language',
                    'Q20':'visulization_library',
                    'Q21':'hardware',
                    'Q22':'has_used_TPU',
                    'Q23':'ML_experience',
                    'Q24':'ML_algorithm',
                    'Q25':'ML_tool_category',
                    'Q26':'CV_method',
                    'Q27':'NLP_method',
                    'Q28':'ML_framework',
                    'Q29':'cloud_computing_platform',
                    'Q30':'cloud_computing_product',
                    'Q31':'big_data_product',
                    'Q32':'ML_product',
                    'Q33':'AutoML_tool',
                    'Q34':'relational_DB_product'
                   }

question_prompt  = pd.DataFrame.from_records([question_mapping])
question_prompt
```

Out[5]:

	Time from Start to Finish (seconds)	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
0	survey_duration	age	gender	country	education	job_title	company_size	DS_team_size	is_ML_us

In [6]:
```
#We may want to read questions_only file.
#And create appropriate sort abbreviations to rename columns in our dataset fo
r convenience.
#Because it is not allowed to read files except mentioned in the assignment sh
eet, I am creating a df using a dictionary above.

#read the file with question prompts
#question_prompt = pd.read_csv('other_data/questions_only.csv')

#add the question keyword to question_prompt df
#question_prompt = question_prompt.append(question_mapping, ignore_index = Tru
e)
```

In [7]:
```
#rename the column names using keywords mapped above for better readability an
d easy preprocessing
kaggle_data.rename(columns={'Time from Start to Finish (seconds)':'survey_dura
tion'}, inplace = True)
for Q_no in question_prompt.columns:

    for col_name in kaggle_data.columns:
        #check for matching question number
        if re.search(r'\b'+ Q_no + r'\b', col_name):
            #read the new name from the question_prompt row 1
            new_name = col_name.replace(Q_no, question_prompt.loc[0:, Q_no].va
lues[0])
            #rename the col in kaggle_data df with new_name
            kaggle_data.rename(columns={col_name:new_name}, inplace = True)

        if Q_no + '_' in col_name:
            #read the new name from the question_prompt row 1
            new_name = col_name.replace(Q_no, question_prompt.loc[0:, Q_no].va
lues[0])
            #rename the col in kaggle_data df with new_name
            kaggle_data.rename(columns={col_name:new_name}, inplace = True)
```

In [8]:
```
#check whether new column names are as intended
kaggle_data.head()
```

Out[8]:

	survey_duration	age	gender	gender_OTHER_TEXT	country	education	job_title	job_title
id								
0	510	22-24	Male	-1	France	Master's degree	Software Engineer	
1	423	40-44	Male	-1	India	Professional degree	Software Engineer	
3	391	40-44	Male	-1	Australia	Master's degree	Other	
4	392	22-24	Male	-1	India	Bachelor's degree	Other	
5	470	50-54	Male	-1	France	Master's degree	Data Scientist	

Filter the columns based on questions of interest

In [9]:
```
#questions of interest for this assignment
ques_of_interest = {'Q1':'age',
                    'Q2':'gender',
                    'Q3':'country',
                    'Q4':'education',
                    'Q5':'job_title',
                    'Q6':'company_size',
                    'Q7':'DS_team_size',
                    'Q8':'is_ML_used',
                    'Q9':'job_activities',
                    'Q10':'salary_USD',
                    'Q15':'DS_coding_experience',
                    'Q18':'programming_language',
                    'Q23':'ML_experience',
                   }
```

In [10]:
```
#add columns that contain data related to above questions
col_of_interest = []

for value in ques_of_interest.values():
    for col_name in kaggle_data.columns:
        #check for matching question number
        if re.search(r'\b'+ value, col_name):
            col_of_interest.append(col_name)

print('columns of interest for this assignment:\n', col_of_interest)
```

```
columns of interest for this assignment:
 ['age', 'gender', 'gender_OTHER_TEXT', 'country', 'education', 'job_title',
'job_title_OTHER_TEXT', 'company_size', 'DS_team_size', 'is_ML_used', 'job_ac
tivities_Part_1', 'job_activities_Part_2', 'job_activities_Part_3', 'job_acti
vities_Part_4', 'job_activities_Part_5', 'job_activities_Part_6', 'job_activi
ties_Part_7', 'job_activities_Part_8', 'job_activities_OTHER_TEXT', 'salary_U
SD', 'DS_coding_experience', 'programming_language_Part_1', 'programming_lang
uage_Part_2', 'programming_language_Part_3', 'programming_language_Part_4',
'programming_language_Part_5', 'programming_language_Part_6', 'programming_la
nguage_Part_7', 'programming_language_Part_8', 'programming_language_Part_9',
'programming_language_Part_10', 'programming_language_Part_11', 'programming_
language_Part_12', 'programming_language_OTHER_TEXT', 'ML_experience']
```

In [11]:
```
#this is the dataframe that we will use for the rest of the tasks.
kaggle_data = kaggle_data.loc[:, col_of_interest]
```

In [12]:
```
kaggle_data.head()
```

Out[12]:

	age	gender	gender_OTHER_TEXT	country	education	job_title	job_title_OTHER_TEXT	c
id								
0	22-24	Male	-1	France	Master's degree	Software Engineer	-1	
1	40-44	Male	-1	India	Professional degree	Software Engineer	-1	
3	40-44	Male	-1	Australia	Master's degree	Other	0	
4	22-24	Male	-1	India	Bachelor's degree	Other	1	
5	50-54	Male	-1	France	Master's degree	Data Scientist	-1	

Exploratory Data Analysis

Q1: Perform exploratory data analysis to analyze the survey dataset and to summarize its main characteristics. Present 3 graphical figures that represent different trends in the data. For your explanatory data analysis, you can consider Country, Age, Education, Professional Experience, and Salary.

info

In [13]:
```
print(kaggle_data.info(verbose=True))
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12497 entries, 0 to 19716
Data columns (total 35 columns):
 #   Column                           Non-Null Count  Dtype 
---  ------                           --------------  ----- 
 0   age                              12497 non-null  object
 1   gender                           12497 non-null  object
 2   gender_OTHER_TEXT                12497 non-null  int64 
 3   country                          12497 non-null  object
 4   education                        12497 non-null  object
 5   job_title                        12497 non-null  object
 6   job_title_OTHER_TEXT             12497 non-null  int64 
 7   company_size                     12497 non-null  object
 8   DS_team_size                     12497 non-null  object
 9   is_ML_used                       12497 non-null  object
 10  job_activities_Part_1            5979 non-null   object
 11  job_activities_Part_2            3507 non-null   object
 12  job_activities_Part_3            4890 non-null   object
 13  job_activities_Part_4            3285 non-null   object
 14  job_activities_Part_5            3634 non-null   object
 15  job_activities_Part_6            2303 non-null   object
 16  job_activities_Part_7            515 non-null    object
 17  job_activities_Part_8            239 non-null    object
 18  job_activities_OTHER_TEXT        12497 non-null  int64 
 19  salary_USD                       12497 non-null  int64 
 20  DS_coding_experience             11422 non-null  object
 21  programming_language_Part_1      9363 non-null   object
 22  programming_language_Part_2      3652 non-null   object
 23  programming_language_Part_3      5428 non-null   object
 24  programming_language_Part_4      949 non-null    object
 25  programming_language_Part_5      1356 non-null   object
 26  programming_language_Part_6      1598 non-null   object
 27  programming_language_Part_7      1720 non-null   object
 28  programming_language_Part_8      353 non-null    object
 29  programming_language_Part_9      1763 non-null   object
 30  programming_language_Part_10     962 non-null    object
 31  programming_language_Part_11     69 non-null     object
 32  programming_language_Part_12     1016 non-null   object
 33  programming_language_OTHER_TEXT  12497 non-null  int64 
 34  ML_experience                    10541 non-null  object
dtypes: int64(5), object(30)
memory usage: 3.4+ MB
None
```

We can see that all the columns, except ML_experience, have no null values. The questions that have more than one columns (corrosponding to multiple choice answers) may have different non-null values for different choices. However, column name ending in 'OTHER_TEXT' can be used to infer the completeness (no missing values).

Summary

In [14]:
```
kaggle_data.describe()['salary_USD']
```

Out[14]:
```
count     12497.000000
mean      57124.189806
std       73710.709307
min        1000.000000
25%        7500.000000
50%       30000.000000
75%       80000.000000
max      500000.000000
Name: salary_USD, dtype: float64
```

As you can see, describe method doesn't provide much information due to nature of data. So, we will look at the unique values and some charts to observe the trends below.

check unique values for each column

In [15]:
```
for col in kaggle_data:
    print(col,':', kaggle_data[col].unique(), '\n')
```

```
age : ['22-24' '40-44' '50-54' '55-59' '30-34' '18-21' '35-39' '25-29' '45-4
9'
 '60-69' '70+']

gender : ['Male' 'Female' 'Prefer to self-describe' 'Prefer not to say']

gender_OTHER_TEXT : [-1  0  1  2  3  4  5  6  7  8  9 11 13 14 15 16 18 22 27
30 31 33 34 35
 37 39]

country : ['France' 'India' 'Australia' 'United States of America' 'Netherlan
ds'
 'Germany' 'Ireland' 'Russia' 'Greece' 'Ukraine' 'Pakistan' 'Japan'
 'Other' 'Brazil' 'South Korea' 'Belarus' 'Nigeria'
 'United Kingdom of Great Britain and Northern Ireland' 'Sweden' 'Mexico'
 'Canada' 'Portugal' 'Poland' 'Indonesia' 'Italy' 'Czech Republic' 'Spain'
 'Chile' 'Hong Kong (S.A.R.)' 'South Africa' 'Argentina' 'Turkey' 'Israel'
 'Taiwan' 'Egypt' 'Morocco' 'Hungary' 'Colombia' 'Norway' 'Thailand'
 'Switzerland' 'Viet Nam' 'Singapore' 'Bangladesh'
 'Iran, Islamic Republic of...' 'Peru' 'Kenya' 'Romania' 'China' 'Belgium'
 'Austria' 'Algeria' 'New Zealand' 'Tunisia' 'Philippines' 'Malaysia'
 'Republic of Korea' 'Denmark' 'Saudi Arabia']

education : ['Master’s degree' 'Professional degree' 'Bachelor’s degree'
 'Doctoral degree'
 'Some college/university study without earning a bachelor’s degree'
 'I prefer not to answer' 'No formal education past high school']

job_title : ['Software Engineer' 'Other' 'Data Scientist' 'Statistician'
 'Product/Project Manager' 'Data Analyst' 'Research Scientist'
 'Business Analyst' 'Data Engineer' 'DBA/Database Engineer']

job_title_OTHER_TEXT : [ -1   0   1   2   4   5   6   7   8   9  10  11  12
13  14  15  16  17
  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305
 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323
 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341
 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359
 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377
 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395
 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413
 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431
```

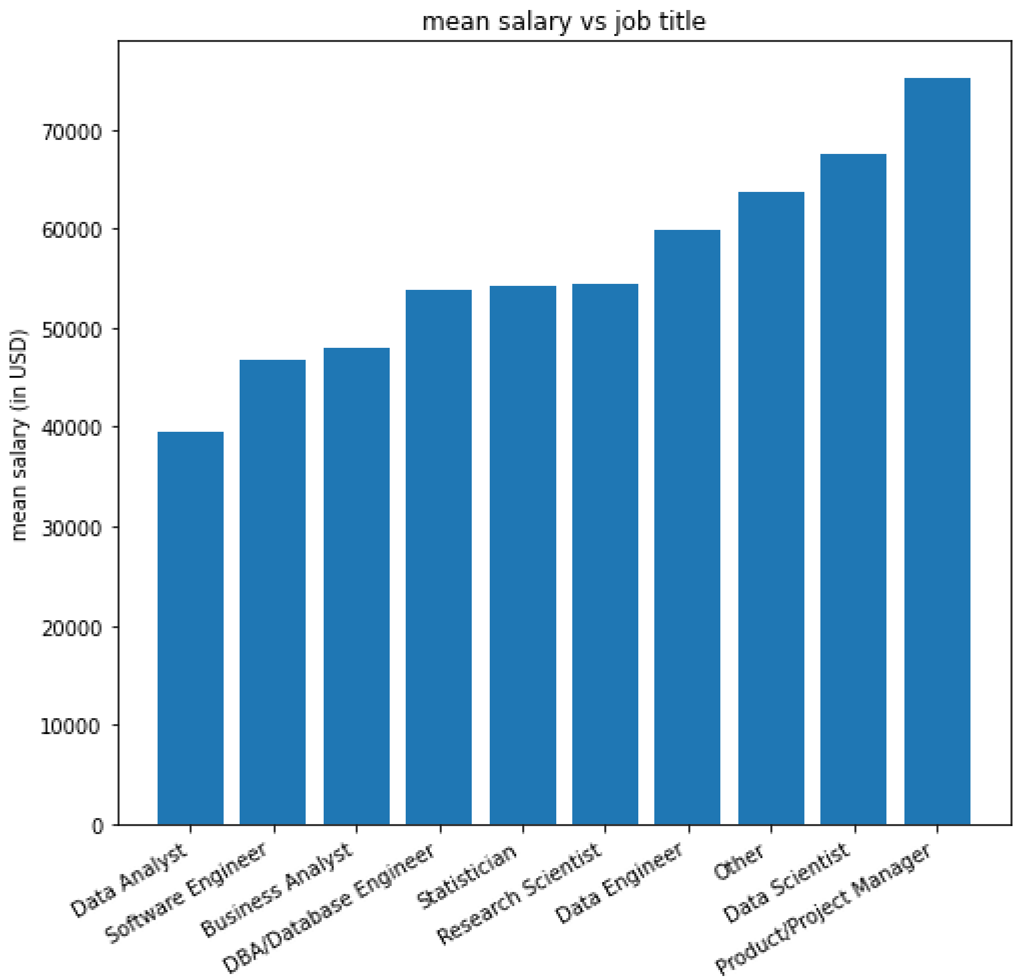
```
 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449
 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467
 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485
 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503
 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521
 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539
 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557
 559 560 561 562 563 564 565 566 567 568 569 570 571 573 574 575 576 577
 578 579 580 581 582 584 586 589 590 591 594 595 596 597 598 599 601 602
 604 606 607 608 609 611 613 617 618 619 621 622 623 624 626 632 633 635
 636 637 638 640 642 643 644 645 646 648 649 652 653 655 657 658 660 661
 662 663 664 665 666 670 671 672 674 675 676 677   3 680 681 682 683 684
 685 687 690 691 692 693 694 695 696 698 699 700 701 702 703 704 705 706
 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724
 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 741 742 743
 744 745 746 747 749 750 751 752 754 756 757 760 761 762 763 766 771 773
 775 776 780 781 782 783 785 786 787 789 790 791 794 795 796 797 798 799
 800 801 802 804 805 806 808 809 810 813 815 816 817 818 819 820 821 822
 823 824 825 827 828 829 830 656 831 833 834 836 837 838 839 840 841 842
 843 846 847 848 852 853 856 857 859 860 861 863 864 865 866 868 869 871
 873 874 877]

company_size : ['1000-9,999 employees' '> 10,000 employees' '0-49 employees'
 '50-249 employees' '250-999 employees']

DS_team_size : ['0' '20+' '3-4' '1-2' '5-9' '10-14' '15-19']

is_ML_used : ['I do not know'
 'We have well established ML methods (i.e., models in production for more th
an 2 years)'
 'No (we do not use ML methods)'
 'We are exploring ML methods (and may one day put a model into production)'
 'We recently started using ML methods (i.e., models in production for less t
han 2 years)'
 'We use ML methods for generating insights (but do not put working models in
to production)']

job_activities_Part_1 : [nan
 'Analyze and understand data to influence product or business decisions']

job_activities_Part_2 : [nan
 'Build and/or run the data infrastructure that my business uses for storing,
analyzing, and operationalizing data']

job_activities_Part_3 : [nan 'Build prototypes to explore applying machine le
arning to new areas']

job_activities_Part_4 : [nan
 'Build and/or run a machine learning service that operationally improves my
product or workflows']

job_activities_Part_5 : [nan 'Experimentation and iteration to improve existi
ng ML models']

job_activities_Part_6 : [nan 'Do research that advances the state of the art
of machine learning']
```

```
job_activities_Part_7 : [nan 'None of these activities are an important part
of my role at work']

job_activities_Part_8 : [nan 'Other']

job_activities_OTHER_TEXT : [ -1   0   1   2   3   4   5   6   7   8   9  10
11  12  13  14  15  16
  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52
  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70
  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88
  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 123 124 125 126
 127 128 129 130 131 132 133 134 135 136 137]

salary_USD : [ 40000   7500 300000   5000  70000  15000  90000   1000   3000
80000
 100000 150000  50000  25000  20000 125000  10000 200000  30000   4000
   2000 250000  60000 500000 400000]

DS_coding_experience : ['1-2 years' 'I have never written code' '< 1 years'
'20+ years'
 '3-5 years' '5-10 years' '10-20 years' nan]

programming_language_Part_1 : ['Python' nan]

programming_language_Part_2 : ['R' nan]

programming_language_Part_3 : ['SQL' nan]

programming_language_Part_4 : [nan 'C']

programming_language_Part_5 : [nan 'C++']

programming_language_Part_6 : ['Java' nan]

programming_language_Part_7 : ['Javascript' nan]

programming_language_Part_8 : [nan 'TypeScript']

programming_language_Part_9 : [nan 'Bash']

programming_language_Part_10 : ['MATLAB' nan]

programming_language_Part_11 : [nan 'None']

programming_language_Part_12 : [nan 'Other']

programming_language_OTHER_TEXT : [ -1   0   1   2   3   4   5   6   7   9  1
0  11  12  14  15  16  17  18
  19  20  21  22  24  25  26  27  28  30  31  32  33  34  35  36  37  38
  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  29  54  55
  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73
  74  75  76  77  78   8  79  80  81  82  83  84  86  87  88  89  90  91
  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111
 112 113 114 115 118 119 120  23 122 123 124 126 127 128 129 131 132 133
 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151
```

```
  13 152 153 154 156 157 158 159 160 161 162 163 164 165 166 167 168 169
 170 172 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
 190 191 193 194 195 198 199 200 201 202 204 205 206 207 208 209 210 211
 212 213 214 215 217 218 219 220 221 223 224 225 226 227 228 229 230 231
 232 222 233 234 235 236 237 239 240 241 242 243 244 246 247 248 250 251
 252 253 254 256 257 258 259 261 262 263 264 265 266  85 267 268 269 270
 271 272 273 274 276 277 278 279 280 282 283 285 286 287 289 290 292 293
 294 295 296 297 298 299 300 301 302 305]

ML_experience : ['1-2 years' nan '2-3 years' '< 1 years' '10-15 years' '3-4 y
ears'
 '4-5 years' '5-10 years' '20+ years']
```

Trend 1: Job Title vs Mean Salary

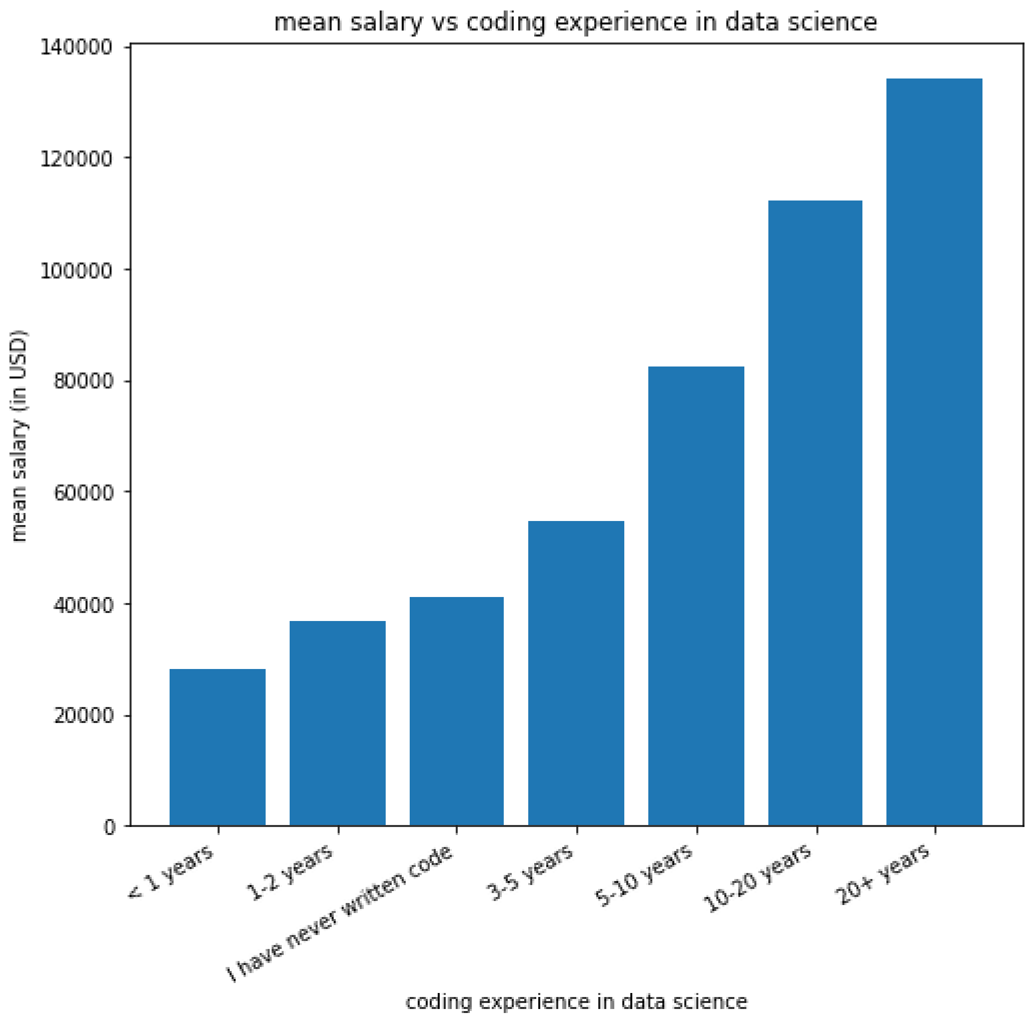
In [16]:
```
#new figure and associated axes
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()

#group by job title and find mean salary for each group
data = kaggle_data.groupby(['job_title'])['salary_USD'].mean().sort_values(asc
ending=True)

#create bar plot
ax.bar(data.index, data);

#set labels and titles
ax.set_ylabel('mean salary (in USD)')
ax.set_title('mean salary vs job title')

#format x_ticks and y_ticks appropriately
fig.autofmt_xdate()
```



As we can see in the graph above, the product/project managers and data scientists command highest salary on average among repondents in our dataset.

Trend 2: Coding Expereince (DS) VS Mean Salary

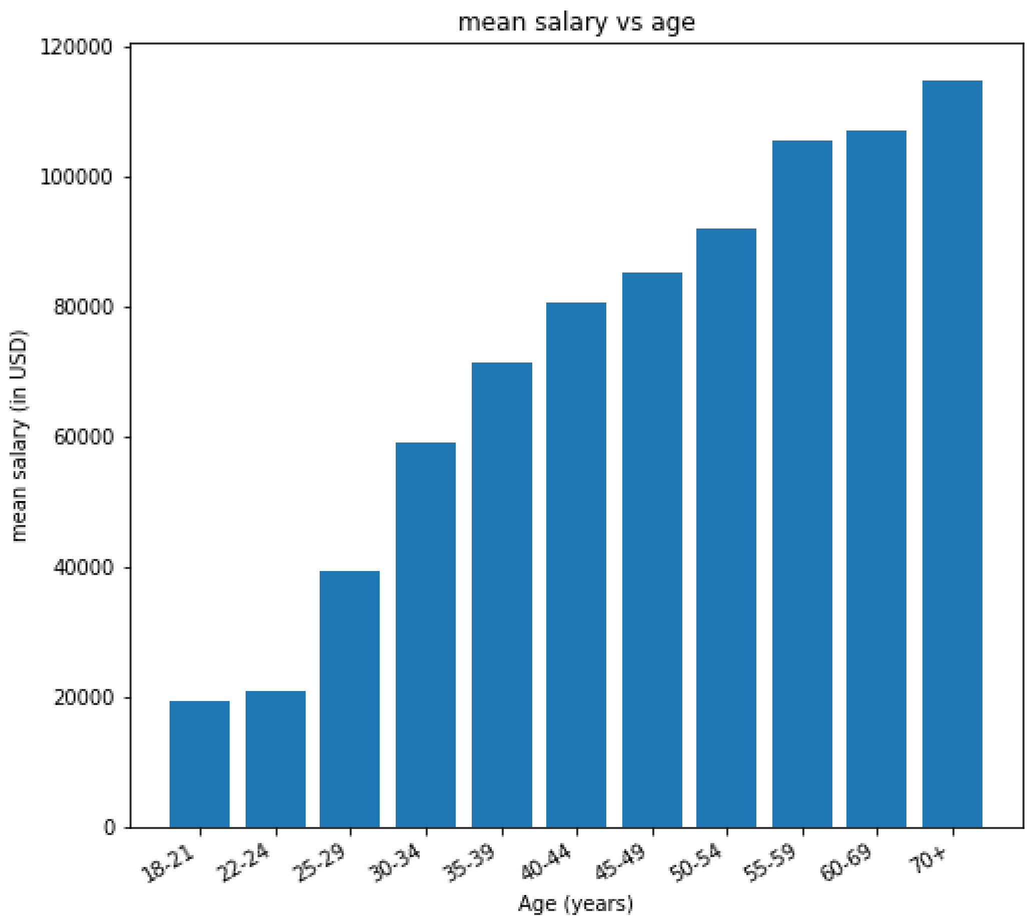
In [17]:
```
#new figure and associated axes
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()

#group by DS coding experience and find mean salary for each group
data = kaggle_data.groupby(['DS_coding_experience'])['salary_USD'].mean().sort
_values(ascending=True)

#create bar plot
ax.bar(data.index, data);

#set labels and titles
ax.set_xlabel('coding experience in data science')
ax.set_ylabel('mean salary (in USD)')
ax.set_title('mean salary vs coding experience in data science')

#format x_ticks and y_ticks appropriately
fig.autofmt_xdate()
```



As we can see in the graph above, the salary increases with experience in general. It seems that the increase is exponential. However, it may be because the respondents with experience less than 1 year could be students, and they are not working full time yet.

Trend 3: Age VS Mean Salary

In [18]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
data = kaggle_data.groupby(['age'])['salary_USD'].mean().sort_values(ascending
=True)
ax.bar(data.index, data);
ax.set_xlabel('Age (years)')
ax.set_ylabel('mean salary (in USD)')
ax.set_title('mean salary vs age')
fig.autofmt_xdate()
```



The averge salary increases with increase in age. This directly relates to trend we observed in previous graph. However, in this graph, the increase doesn't seem exponential.

Q2 Estimating the difference between average salary (Q10) of males vs. females.

Q2a: Compute and report descriptive statistics for each group (remove missing data, if necessary).

Gender column has four distinct values in our dataset. We will only consider two groups: Males and Females.

In [19]:
```
#filter df based on gender
filter1 = kaggle_data['gender'].isin(['Male','Female'])
```

In [20]:
```
#df with column of interest for question 2
df_q2 = kaggle_data.loc[filter1, ['gender', 'salary_USD']]
```

descriptive statistics of male and female group for salary

In [21]:
```
df_q2.groupby(['gender']).describe()
```

Out[21]:

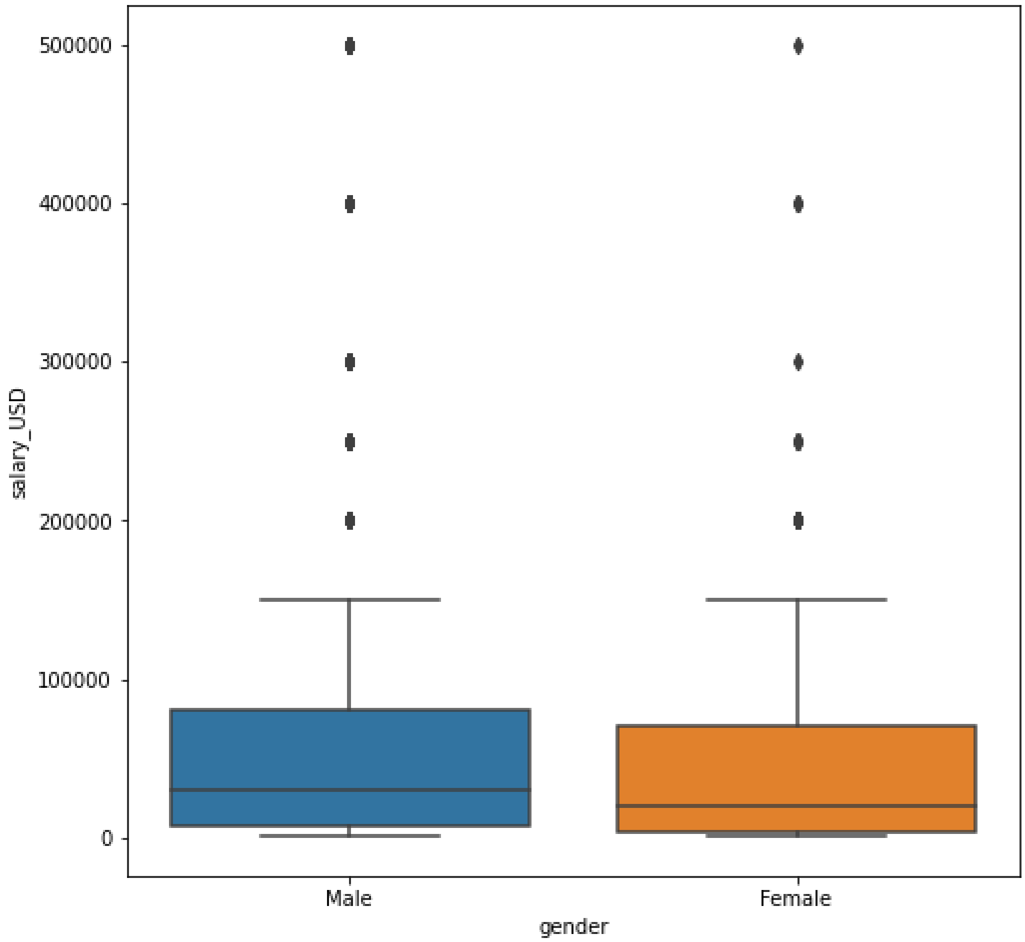
	salary_USD							
	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
gender								
Female	1827.0	45933.771210	60253.789591	1000.0	3000.0	20000.0	70000.0	500000.0
Male	10473.0	58709.586556	74920.620048	1000.0	7500.0	30000.0	80000.0	500000.0

We can see that the mean salary for females is comparatively less than males. Also, the number of female respondents in survey are very less in comparision to males. This supports the fact that few females are working in STEM.

The numbers we obtained above doesn't provide full picture. Let's look at the box plot for both groups.

Distribution of salary among females and males

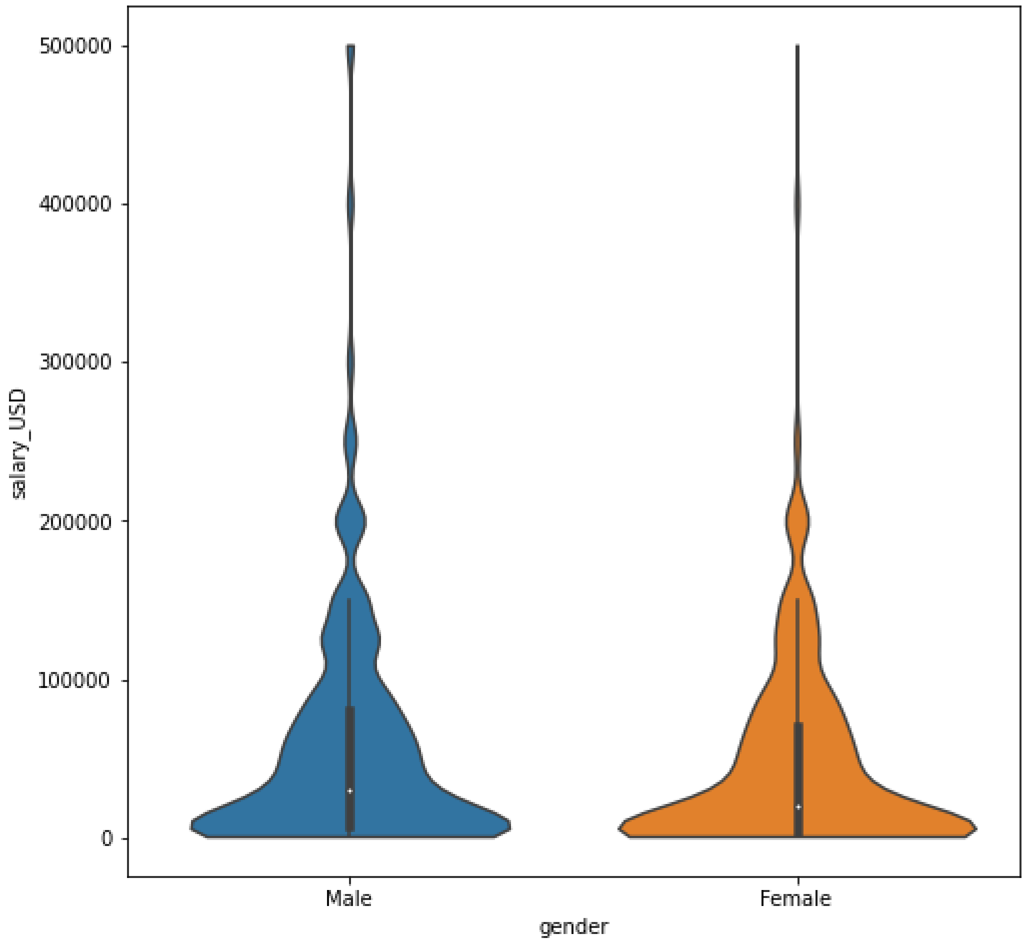
In [22]:
```
#new figure
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()

#create box plots for males and females
ax = sns.boxplot(x="gender", y="salary_USD", data=df_q2)
```



We can see that the distributions have long tails. Though, this graph is not sufficient to understand how salary is distrbuted. Let's look at violin plot.

Violin plot

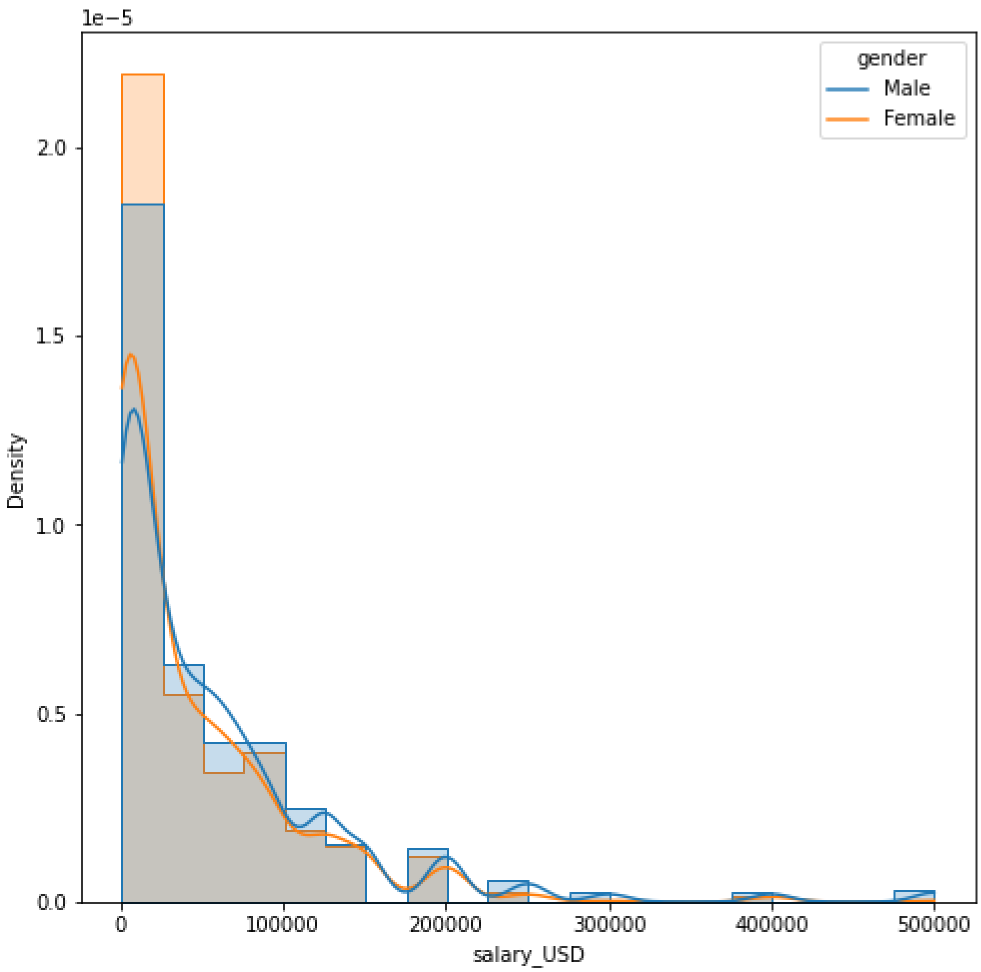
In [23]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
ax = sns.violinplot(x="gender", y="salary_USD", data=df_q2, cut = 0)
```



The majority of both males and females have very less salary, less than mean values.

Histogram

In [24]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
sns.histplot(ax = ax, data=df_q2, x="salary_USD", hue="gender", element="step"
, bins=20, stat="density", common_norm=False);
sns.kdeplot(ax = ax, data=df_q2, x="salary_USD", hue="gender", common_norm=Fal
se, cut=0);
```



Finally, we can see that the distributions for both males and females are right skewed with long tails. Overall, the shape of distribution overlaps with each other. It is hard to conclude anything about difference in mean salary by looking at these histograms. For that, we will perform hypothesis test below.

Note: The densities are calculated by accounting for the disparity in the number of respondents between two groups. This is done by passing extra parameters to seaborn plot methods used above.

Q2b: If suitable, perform a two-sample t-test with 0.05 threshold. Explain your rationale.

If a population from which the data is collected violates any of the t-test assumptions, the result of analysis may be incorrect or misleading. In our case, the assumption of 'normality' is violated. The outliers are present in the sample, which is also representative of a population. We know that the salary can't be less than zero. And hence, the distribution will always be skewed.

For these reasons, it doesn't seem appropriate to perform a two-sample t-test.

To overcome these challenges, we may need to transform data in some way. Here, we are using bootstraping as demonstarted below.

Q2c: Bootstrap your data for comparing the mean of salary (Q10) for the two groups. Note that the number of instances you sample from each group should be relative to its size. Use 1000 replications. Plot two bootstrapped distributions (for males and females) and the distribution of the difference in means.

Bootstrapping

In bootstrapping, we resample data from our original sample with replacement. Ideally, we want to resample as many points as we have in our original sample.

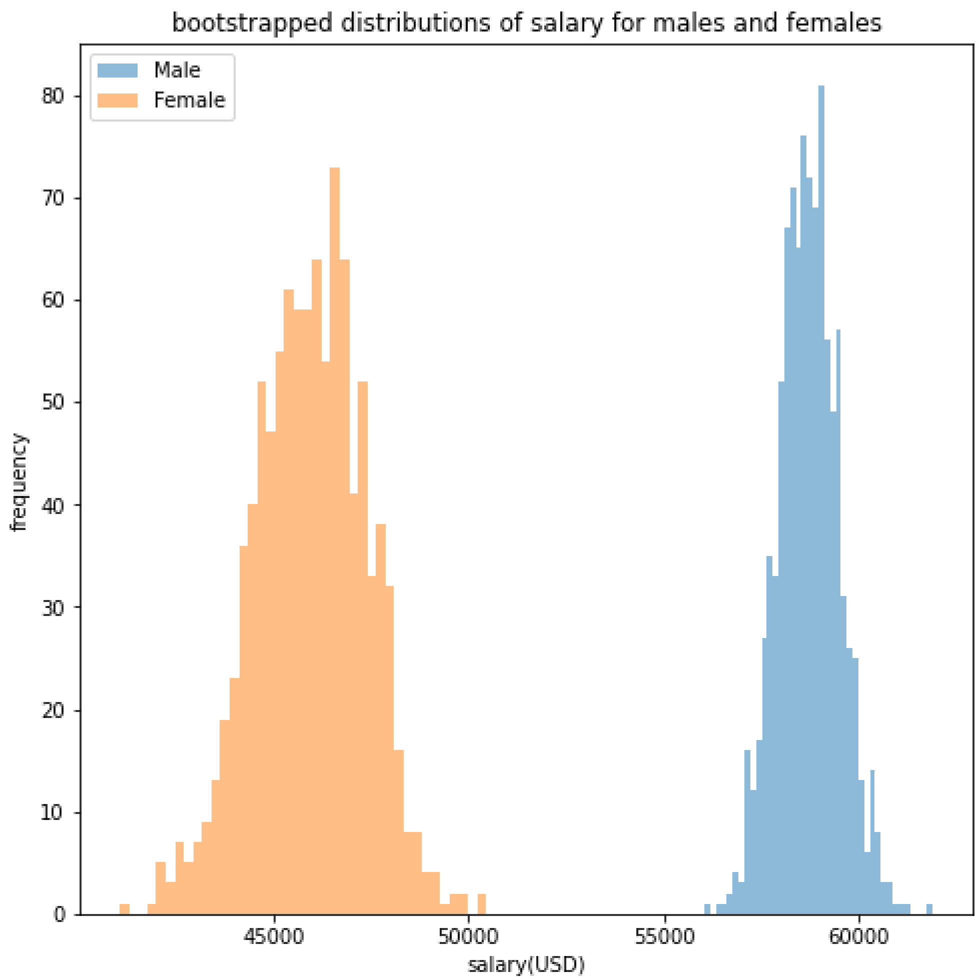
Here, we are creating two bootstrapped samples--one for male and another for female. we are resampling for the same number of points as we have in our original dataset for both groups (maintaining the resampling instances relative to each group's size). In total we create 1000 replications. For each, replication the mean salary is calculated.

In [25]:
```
#create seperated df for males and females to use in bootstraping
df_male = df_q2[df_q2['gender'] == 'Male']
df_female = df_q2[df_q2['gender'] == 'Female']
```

In [26]:
```
male_bootstrap = []
female_bootstrap = []
for i in range(1000):
    #sample from df_male with replacement. Frac = 1 indicates that n is equal
 to observation in original df
    mean_male = df_male['salary_USD'].sample(frac=1, replace=True).mean()
    #sample from df_female with replacement. Frac = 1 indicates that n is equa
l to observation in original df
    mean_female = df_female['salary_USD'].sample(frac=1, replace=True).mean()
    male_bootstrap.append(mean_male)
    female_bootstrap.append(mean_female)
```

Bootstrapped distribution of means for males and females

In [27]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
plt.hist(male_bootstrap, bins = 40, label = 'Male', alpha = 0.5)
plt.hist(female_bootstrap, bins = 40, label = 'Female', alpha = 0.5)
plt.legend()
plt.xlabel('salary(USD)')
plt.ylabel('frequency')
plt.title('bootstrapped distributions of salary for males and females');
```

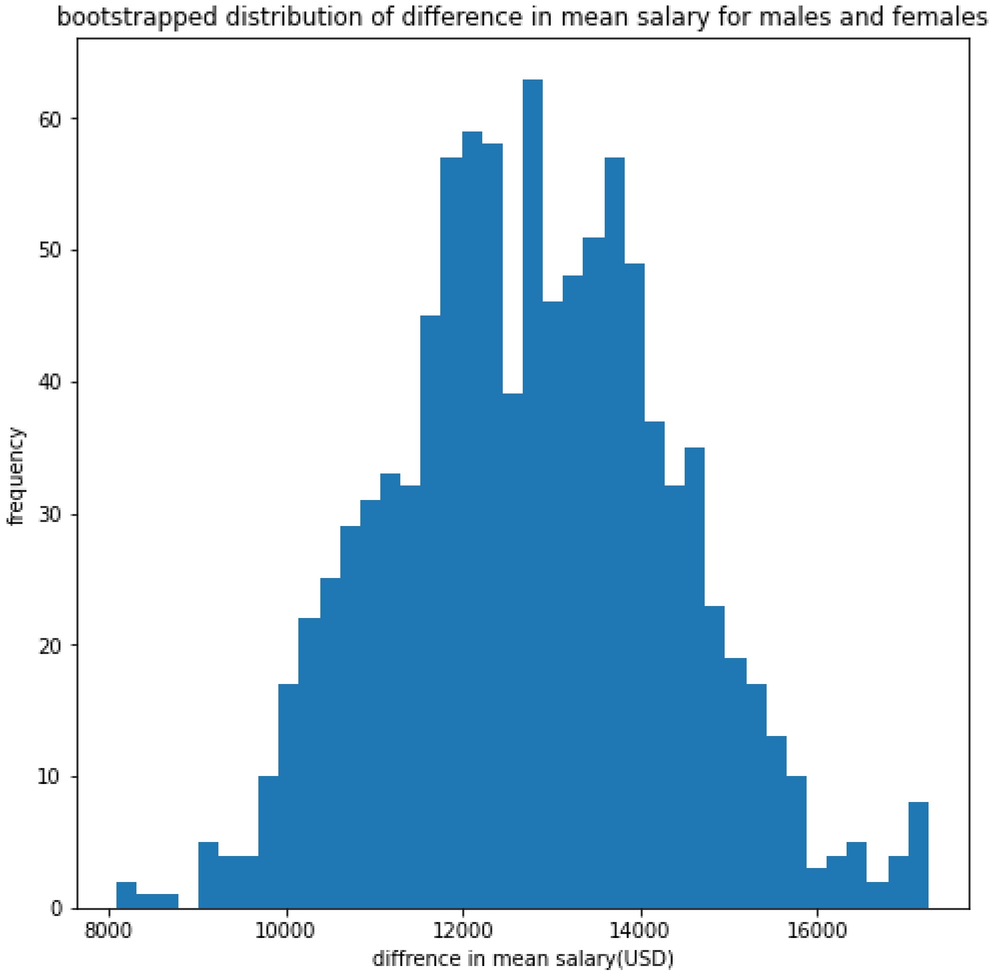


We can observe that bootstrapping removes the skew and yields distributions identical to normal distributions. We can also observe that both distribution are significantly apart from each other.

Bootstrapped distribution of difference in mean

In [28]:
```
#find the diff in mean salary for each bootstrapped sample
diff_mean = np.array(male_bootstrap) - np.array(female_bootstrap)
```

In [29]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
plt.hist(diff_mean, bins = 40)
plt.xlabel('diffrence in mean salary(USD)')
plt.ylabel('frequency')
plt.title('bootstrapped distribution of difference in mean salary for males an
d females');
```



The differences of mean salary between two groups for bootstrapped samples are plotted above. It's distribution is identical to normal distribution, and mean is approxiamtly same as the difference between the mean of two distributions we obtained above.

Q2d: If suitable, perform a two-sample t-test with 0.05 threshold on the bootstrapped data. Explain your rationale.

Rationale to perform t-test:

We can perform t-test on bootstrapped samples as it doesn't severly violate the assumptions of t-test. Given that the distributions are identical to normal distribution, we can be confident in the outcome of test.

Hypothesis testing

Null hypothesis

H_0: There's no difference between the mean of female salary and male salary.

Alternate hypothesis

H_a: There's a difference between the mean of female salary and male salary. (bootstrap)

level of significance

α = 5%

In [30]:
```
ttest, pval= stats.ttest_ind(female_bootstrap, male_bootstrap)

print('t-statistics:', ttest)
print('p-value:', pval)
if pval < 0.05:
    print('p-values is less than 0.05; reject null hypothesis at 5% level of s
ignificance.')
else:
    print('p-values is greater than 0.05; hence, do not reject null hypothesis
at 5% level of significance.')
```

```
t-statistics: -250.8287732373092
p-value: 0.0
p-values is less than 0.05; reject null hypothesis at 5% level of significanc
e.
```

Q2e: Comment on your findings.

We have found enough evidence to reject null hypothesis at 5% level of significance. Which suggests that there is a significant difference between mean salary of male and female.

Q3 Select “highest level of formal education” (Q4) from the dataset and repeat steps a to e, this time use analysis of variance (ANOVA) instead of t test for hypothesis testing to compare the means of salary for three groups (Bachelor’s degree, Doctoral degree, and Master’s degree).

Q3a: Compute and report descriptive statistics for each group (remove missing data, if necessary).

Education column has seven distinct values in our dataset. We will only consider three groups: Doctoral degree, Master’s degree, Bachelor’s degree.

In [31]:
```
#filter df based on education level
filter2 = kaggle_data['education'].isin(['Master’s degree', 'Bachelor’s degre
e', 'Doctoral degree'])
```

In [32]:
```
#df with column of interest for question 3
df_q3 = kaggle_data.loc[filter2, ['education', 'salary_USD']]
```

In [33]:
```
df_q3.head()
```

Out[33]:

	education	salary_USD
id		
0	Master’s degree	40000
3	Master’s degree	300000
4	Bachelor’s degree	5000
5	Master’s degree	70000
6	Master’s degree	15000

descriptive statistics of three groups for salary

In [34]:
```
df_q3.groupby(['education']).describe()
```

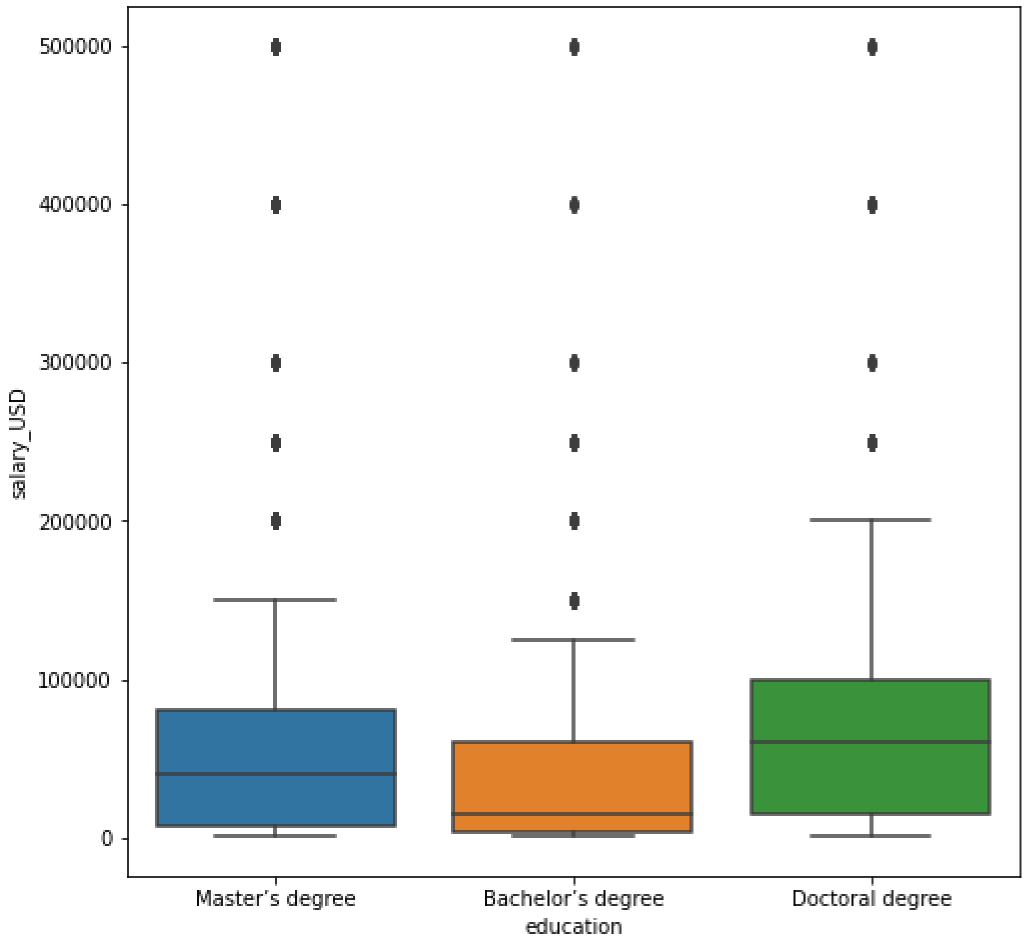
Out[34]:

	salary_USD							
	count	mean	std	min	25%	50%	75%	max
education								
Bachelor’s degree	3361.0	44999.256174	67923.680798	1000.0	3000.0	15000.0	60000.0	500000.0
Doctoral degree	2083.0	75761.401824	83376.717093	1000.0	15000.0	60000.0	100000.0	500000.0
Master’s degree	5868.0	58778.629857	70265.728605	1000.0	7500.0	40000.0	80000.0	500000.0

We can see that mean salary for people with doctoral degrees are high compare to other groups.

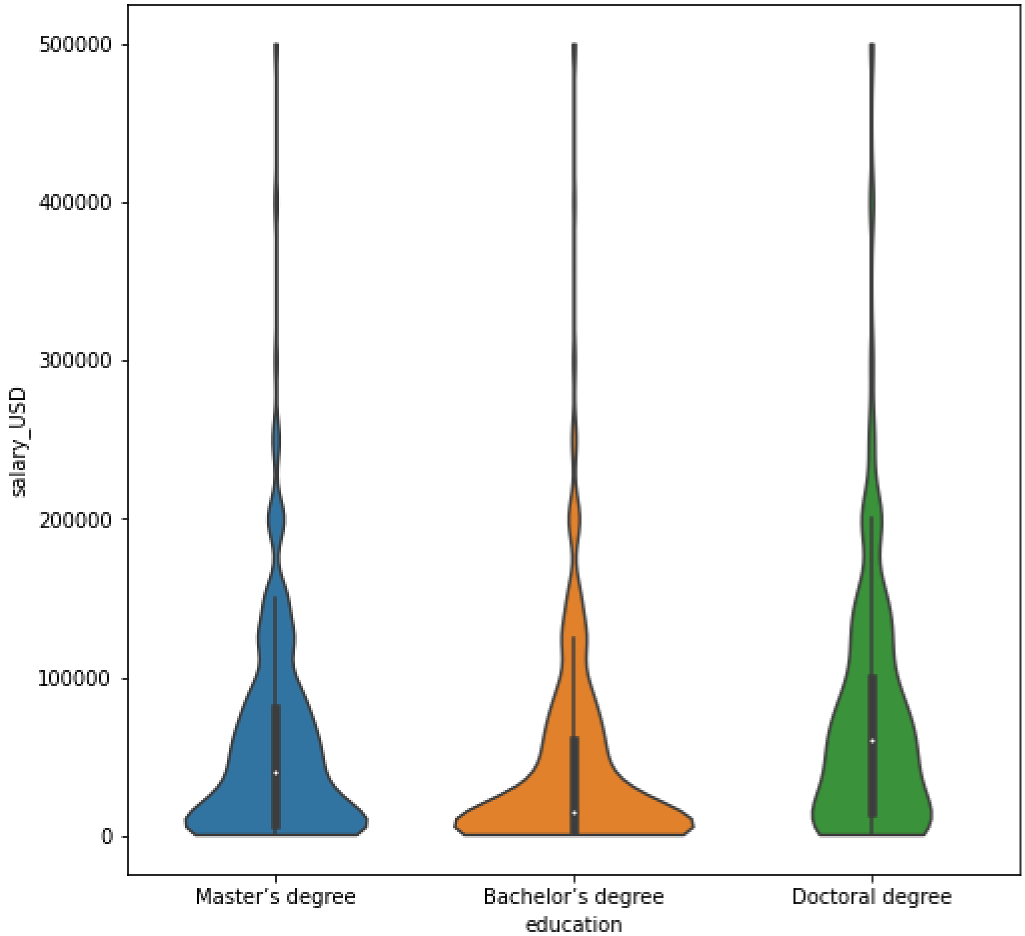
box plot

In [35]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
ax = sns.boxplot(x="education", y="salary_USD", data=df_q3)
```



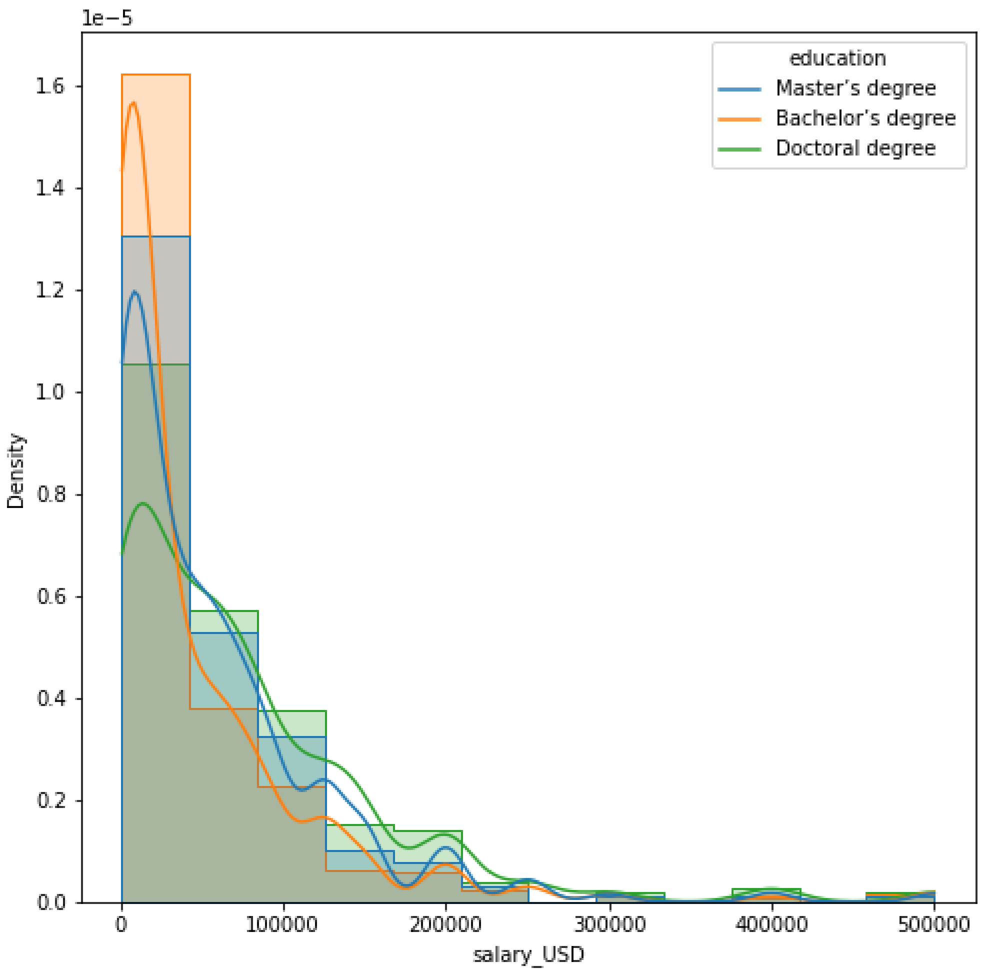
violin plot

In [36]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
ax = sns.violinplot(x="education", y="salary_USD", data=df_q3, cut = 0)
```



Histogram

In [37]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
sns.histplot(ax = ax, data=df_q3, x="salary_USD", hue="education", element="st
ep", bins=12, stat="density", common_norm=False);
sns.kdeplot(ax = ax, data=df_q3, x="salary_USD", hue="education", common_norm=
False, cut=0);
```



All three distributions follow all the characterstic that we mentioned in the answer of question 2a.

Q3b: If suitable, perform a ANOVA test with 0.05 threshold. Explain your rationale.

For similar reasons to those given in answer of question 2b, we can not use ANVOA directly. The reason being skewness, outliers, and violation of normality.

Q2c: Bootstrap your data for comparing the mean of salary (Q10) for the three groups. Note that the number of instances you sample from each group should be relative to its size. Use 1000 replications. Plot three bootstrapped distributions and the distribution of the difference in means.

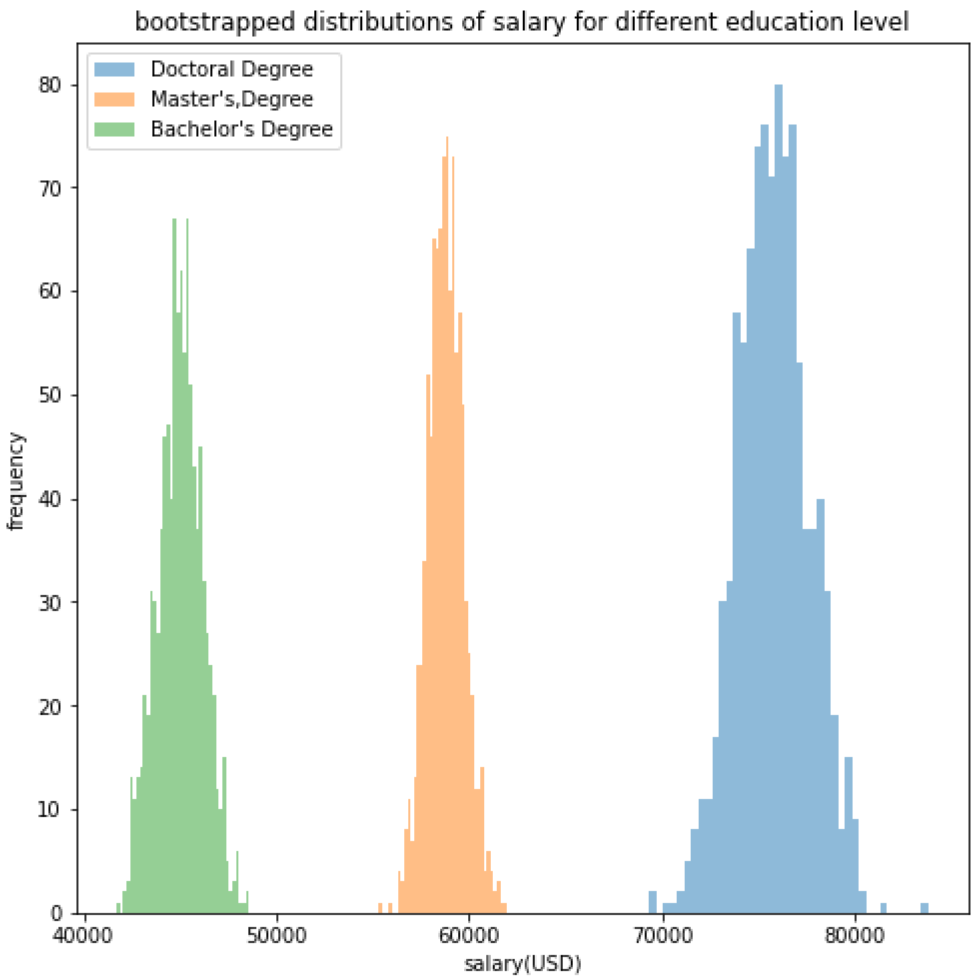
Bootstrapping

In [38]:
```
#create seperated df for males and females to use in bootstraping
df_doctoral = df_q3[df_q3['education'] == 'Doctoral degree']
df_masters = df_q3[df_q3['education'] == 'Master’s degree']
df_bachelor = df_q3[df_q3['education'] == 'Bachelor’s degree']
```

In [39]:
```
doctoral_bootstrap = []
masters_bootstrap = []
bachelor_bootstrap = []
for i in range(1000):
    #sample from df_male with replacement. Frac = 1 indicates that n is equal
 to observation in original df
    mean_doctoral = df_doctoral['salary_USD'].sample(frac=1, replace=True).mea
n()
    #sample from df_female with replacement. Frac = 1 indicates that n is equa
l to observation in original df
    mean_masters = df_masters['salary_USD'].sample(frac=1, replace=True).mean
()
    #sample from df_female with replacement. Frac = 1 indicates that n is equa
l to observation in original df
    mean_bachelor = df_bachelor['salary_USD'].sample(frac=1, replace=True).mea
n()
    doctoral_bootstrap.append(mean_doctoral)
    masters_bootstrap.append(mean_masters)
    bachelor_bootstrap.append(mean_bachelor)
```

Bootstrapped distribution of means for different education levels

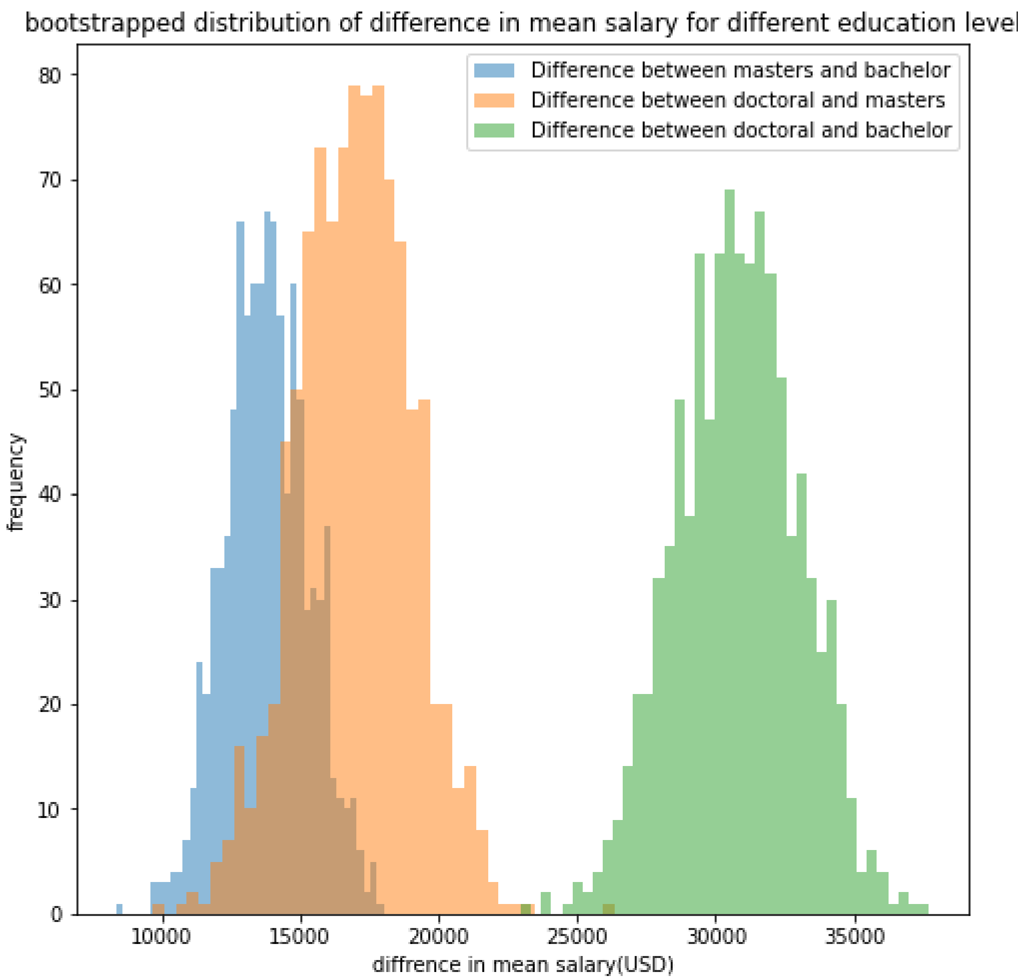
In [40]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
plt.hist(doctoral_bootstrap, bins = 40, label = "Doctoral Degree", alpha = 0.5
)
plt.hist(masters_bootstrap, bins = 40, label = "Master's,Degree", alpha = 0.5)
plt.hist(bachelor_bootstrap, bins = 40, label = "Bachelor's Degree", alpha =
0.5)
plt.legend()
plt.xlabel('salary(USD)')
plt.ylabel('frequency')
plt.title('bootstrapped distributions of salary for different education level'
);
```



Bootstrapped distribution of difference in means

In [41]:
```
#find the diff in mean salary for each bootstrapped sample
diff_masters_bachelor = np.array(masters_bootstrap) - np.array(bachelor_bootst
rap)
diff_doctoral_masters = np.array(doctoral_bootstrap) - np.array(masters_bootst
rap)
diff_doctoral_bachelor = np.array(doctoral_bootstrap) - np.array(bachelor_boot
strap)
```

In [42]:
```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot()
plt.hist(diff_masters_bachelor, bins = 40, label = "Difference between masters
and bachelor", alpha = 0.5)
plt.hist(diff_doctoral_masters, bins = 40, label = "Difference between doctora
l and masters", alpha = 0.5)
plt.hist(diff_doctoral_bachelor, bins = 40, label = "Difference between doctor
al and bachelor", alpha = 0.5)
plt.legend()
plt.xlabel('diffrence in mean salary(USD)')
plt.ylabel('frequency')
plt.title('bootstrapped distribution of difference in mean salary for differen
t education level');
```



Q3d: If suitable, perform a ANOVA with 0.05 threshold on the bootstrapped data. Explain your rationale.

Rationale to perform ANOVA:

The ditributions obtained after bootstraping are identical to normal distribution. No outliers and skewness. Hence, we can use ANOVA for hypothesis testing

Hypothesis testing

Null hypothesis

H_0: There's no difference between the mean of salaries for people with Doctoral, Masters, or Bachelor degrees.

Alternate hypothesis

H_a: There's a difference between the mean of salaries among peole with Doctoral, Masters, and Bachelor degrees. (bootstrap)

level of significance

α = 5%

In [43]:
```
#calculate the test statistics and p-value
ftest, pval = stats.f_oneway(masters_bootstrap, doctoral_bootstrap, bachelor_b
ootstrap)

#print the values
print('f-statistics:', ftest)
print('p-value:', pval)

#test the hypothesis
if pval < 0.05:
    print('p-values is less than 0.05; reject null hypothesis at 5% level of s
ignificance.')
else:
    print('p-values is greater than 0.05; hence, do not reject null hypothesis
at 5% level of significance.')
```

```
f-statistics: 124455.29931401365
p-value: 0.0
p-values is less than 0.05; reject null hypothesis at 5% level of significanc
e.
```

Q3e: Comment on your findings.

We have found enough evidence to reject null hypothesis at 5% level of significance. Which suggests that there is a significant difference between mean salary among different education levels under consideration--bachelor, masters, and doctoral degrees.