In [3]:
```
 # До початку роботи
import pandas as pd

df = pd.read_csv("survey_results_public.csv")
schema = pd.read_csv("survey_results_schema.csv")
```

```
C:\Users\Агрорен Ново\AppData\Local\Temp\ipykernel_6344\1186318207.py:4: DtypeWar
ning: Columns (56,74,92,97,98,105,109,110,132,162,165) have mixed types. Specify
dtype option on import or set low_memory=False.
  df = pd.read_csv("survey_results_public.csv")
```

In [2]:
```
df.shape
```

Out[2]:
```
(49191, 172)
```

In [3]:
```
schema.head()
```

Out[3]:

	qid	qname	question	type	sub	sq_id
0	QID18	TechEndorse_1	What attracts you to a technology or causes yo...	RO	AI integration or AI Agent capabilities	1.0
1	QID18	TechEndorse_2	What attracts you to a technology or causes yo...	RO	Easy-to-use API	2.0
2	QID18	TechEndorse_3	What attracts you to a technology or causes yo...	RO	Robust and complete API	3.0
3	QID18	TechEndorse_4	What attracts you to a technology or causes yo...	RO	Customizable and manageable codebase	4.0
4	QID18	TechEndorse_5	What attracts you to a technology or causes yo...	RO	Reputation for quality	5.0

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49191 entries, 0 to 49190
Columns: 172 entries, ResponseId to JobSat
dtypes: float64(52), int64(1), object(119)
memory usage: 64.6+ MB
```

In [5]:
```
df.isna().sum().sum()
```

Out[5]:
```
np.int64(4437549)
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
np.int64(0)
```

In [7]:
```
#Завдання 1. Підрахунок загальної кількості респондентів
df.shape[0]
```

```
Out[7]:  49191
```

```
In [8]:  #Завдання 2. Аналіз повноти відповідей респондентів
         schema_qnames = set(schema["qname"].dropna())

         len(schema_qnames)
```

```
Out[8]:  139
```

```
In [9]:  schema_qnames = schema["qname"].dropna().unique()
         schema_qnames
```

```
Out[9]:  array(['TechEndorse_1', 'TechEndorse_2', 'TechEndorse_3', 'TechEndorse_4',
                'TechEndorse_5', 'TechEndorse_6', 'TechEndorse_7', 'TechEndorse_8',
                'TechEndorse_9', 'TechEndorse_13', 'TechEndorse_13_TEXT',
                'TechOppose_1', 'TechOppose_2', 'TechOppose_3', 'TechOppose_5',
                'TechOppose_7', 'TechOppose_9', 'TechOppose_11', 'TechOppose_13',
                'TechOppose_16', 'TechOppose_15', 'TechOppose_15_TEXT',
                'JobSatPoints_1', 'JobSatPoints_2', 'JobSatPoints_16',
                'JobSatPoints_3', 'JobSatPoints_4', 'JobSatPoints_5',
                'JobSatPoints_6', 'JobSatPoints_7', 'JobSatPoints_8',
                'JobSatPoints_9', 'JobSatPoints_10', 'JobSatPoints_11',
                'JobSatPoints_13', 'JobSatPoints_14', 'JobSatPoints_15',
                'JobSatPoints_15_TEXT', 'SO_Actions_1', 'SO_Actions_16',
                'SO_Actions_3', 'SO_Actions_4', 'SO_Actions_5', 'SO_Actions_6',
                'SO_Actions_9', 'SO_Actions_7', 'SO_Actions_10', 'SO_Actions_15',
                'SO_Actions_15_TEXT', 'MainBranch', 'Age', 'EdLevel', 'Employment',
                'EmploymentAddl', 'WorkExp', 'LearnCodeChoose', 'LearnCode',
                'LearnCodeAI', 'AILearnHow', 'YearsCode', 'DevType', 'OrgSize',
                'ICorPM', 'RemoteWork', 'PurchaseInfluence', 'TechEndorseIntro',
                'Industry', 'JobSat', 'AIThreat', 'NewRole', 'ToolCountWork',
                'ToolCountPersonal', 'Country', 'Currency', 'CompTotal',
                'LanguageChoice', 'Language', 'LanguagesHaveEntry',
                'LanguagesWantEntry', 'DatabaseChoice', 'Database',
                'DatabaseHaveEntry', 'DatabaseWantEntry', 'PlatformChoice',
                'Platform', 'PlatformHaveEntry', 'PlatformWantEntry',
                'WebframeChoice', 'Webframe', 'WebframeHaveEntry',
                'WebframeWantEntry', 'DevEnvsChoice', 'DevEnvs', 'DevEnvHaveEntry',
                'DevEnvWantEntry', 'SOTags', 'SOTagsHaveEntry', 'SOTagsWant Entry',
                'OpSys', 'OfficeStackAsync', 'OfficeStackHaveEntry',
                'OfficeStackWantEntry', 'CommPlatform', 'CommPlatformHaveEntr',
                'CommPlatformWantEntr', 'AIModelsChoice', 'AIModels',
                'AIModelsHaveEntry', 'AIModelsWantEntry', 'SOAccount',
                'SOVisitFreq', 'SODuration', 'SOPartFreq', 'SO_Dev_Content',
                'SOComm', 'SOFriction', 'AISelect', 'AISent', 'AIAcc', 'AIComplex',
                'AITool', 'AIFrustration', 'AIExplain', 'AIAgents',
                'AIAgentChange', 'AIAgent_Uses', 'AgentUsesGeneral',
                'AIAgentImpact', 'AIAgentChallenges', 'AIAgentKnowledge',
                'AIAgentKnowWrite', 'AIAgentOrchestration', 'AIAgentOrchWrite',
                'AIAgentObserveSecure', 'AIAgentObsWrite', 'AIAgentExternal',
                'AIAgentExtWrite', 'AIHuman', 'AIOpen'], dtype=object)
```

```
In [4]:  df_columns = set(df.columns)

         df["ResponseId"].nunique()
```

```
Out[4]:  49191
```

In [11]:
```
schema_qnames = set(schema["qname"].dropna())
df_columns = set(df.columns)
```

In [12]:
```
survey_questions = schema_qnames.intersection(df_columns)

len(survey_questions)
```

Out[12]:
```
126
```

In [13]:
```
complete_respondents = df[list(survey_questions)].notna().all(axis=1)
complete_respondents.sum()
```

Out[13]:
```
np.int64(0)
```

In [14]:
```
#Завдання 3. Статистичний аналіз досвіду респондентів

df["WorkExp"].head()
```

Out[14]:
```
0     8.0
1     2.0
2    10.0
3     4.0
4    21.0
Name: WorkExp, dtype: float64
```

In [23]:
```
WorkExp = df["WorkExp"]

stats = pd.Series({
    "Mean (Середнє)": WorkExp.mean(),
    "Median (Медіана)": WorkExp.median(),
    "Mode (Мода)": WorkExp.mode().iloc[0]
})
stats.round(2)
```

Out[23]:
```
Mean (Середнє)      13.37
Median (Медіана)    10.00
Mode (Мода)         10.00
dtype: float64
```

In [ ]:
```
#Завдання 4. Аналіз віддаленої роботи
```

In [25]:
```
df["RemoteWork"].value_counts(dropna=False)
```

Out[25]:
```
RemoteWork
NaN
15411
Remote
10931
Hybrid (some remote, leans heavy to in-person)
6732
In-person
6042
Hybrid (some in-person, leans heavy to flexibility)
5831
Your choice (very flexible, you can come in when you want or just as needed)
4244
Name: count, dtype: int64
```

In [27]:
```
remote_count = (df["RemoteWork"] == "Remote").sum()
remote_count
```

Out[27]:
```
np.int64(10931)
```

In [36]:
```
#Завдання 5. Визначення популярності Python
[col for col in df.columns if "Language" in col]
```

Out[36]:
```
['LanguageChoice',
 'LanguageHaveWorkedWith',
 'LanguageWantToWorkWith',
 'LanguageAdmired',
 'LanguagesHaveEntry',
 'LanguagesWantEntry']
```

In [37]:
```
df["LanguageHaveWorkedWith"].head()
```

Out[37]:
```
0                      Bash/Shell (all shells);Dart;SQL
1                                                  Java
2                   Dart;HTML/CSS;JavaScript;TypeScript
3                                       Java;Kotlin;SQL
4     C;C#;C++;Delphi;HTML/CSS;Java;JavaScript;Lua;P...
Name: LanguageHaveWorkedWith, dtype: object
```

In [41]:
```
python_users = df["LanguageHaveWorkedWith"].str.contains("Python", na=False)
python_users.sum()
```

Out[41]:
```
np.int64(18466)
```

In [43]:
```
python_percentage = python_users.mean() * 100
round(python_percentage, 2)
f"{python_percentage:.2f}%"
```

Out[43]:
```
'37.54%'
```

In [46]:
```
#Завдання 6. Аналіз шляхів навчання програмуванню

[col for col in df.columns if "Learn" in col]
```

Out[46]:
```
['LearnCodeChoose', 'LearnCode', 'LearnCodeAI', 'AILearnHow']
```

In [48]:
```
df["LearnCode"].head()
```

Out[48]:
```
0    Online Courses or Certification (includes all ...
1    Online Courses or Certification (includes all ...
2    Online Courses or Certification (includes all ...
3    Other online resources (e.g. standard search, ...
4                                                  NaN
Name: LearnCode, dtype: object
```

In [49]:
```
online_courses = df["LearnCode"].str.contains("Online Courses", na=False)
online_courses.sum()
```

Out[49]:
```
np.int64(10973)
```

In [ ]:
```
#Завдання 7. Географічний аналіз компенсації Python-розробників
```

In [6]:
```
python_mask = df["LanguageHaveWorkedWith"].str.contains("Python", na=False)
python_df = df[python_mask]
```

In [7]:
```
python_df = python_df.dropna(subset=["ConvertedCompYearly"])
```

In [8]:
```
compensation_by_country = (
    python_df
    .groupby("Country")["ConvertedCompYearly"]
    .agg(
        mean_compensation="mean",
        median_compensation="median"
    )
    .reset_index()
    .sort_values(by="mean_compensation", ascending=False)
)
```

In [9]:
```
compensation_by_country.head(10)
```

Out[9]:

	Country	mean_compensation	median_compensation
102	Oman	390135.000000	390135.0
3	Andorra	226103.500000	226103.5
149	Viet Nam	218837.166667	8254.0
145	United States of America	173298.590211	150000.0
132	Switzerland	156456.600000	142592.0
121	Singapore	147515.121951	91391.0
65	Israel	135828.365385	142594.0
98	Nigeria	128773.276596	1808.0
63	Ireland	120523.918919	116015.0
99	Nomadic	120131.571429	139218.0

In [ ]:
```
#Завдання 8. Аналіз освіти найбільш оплачуваних спеціалістів
```

In [63]:
```
schema[schema["question"].str.contains("level", case=False, na=False)]
```

Out[63]:

	qid	qname	question	type	sub	sq_id
51	QID9	EdLevel	Which of the following best describes the high...	MC	NaN	NaN

In [64]:
```
schema[schema["question"].str.contains("education", case=False, na=False)]
```

Out[64]:

	qid	qname	question	type	sub	sq_id
51	QID9	EdLevel	Which of the following best describes the high...	MC	NaN	NaN
59	QID13	YearsCode	Including any education, how many years have y...	TE	NaN	NaN

In [65]:
```
df[["ConvertedCompYearly", "EdLevel"]].head()
```

Out[65]:

	ConvertedCompYearly	EdLevel
0	61256.0	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)
1	104413.0	Associate degree (A.A., A.S., etc.)
2	53061.0	Bachelor’s degree (B.A., B.S., B.Eng., etc.)
3	36197.0	Bachelor’s degree (B.A., B.S., B.Eng., etc.)
4	60000.0	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)

In [66]:
```
top_paid = df.dropna(subset=["ConvertedCompYearly", "EdLevel"])
```

In [67]:
```
top_paid = top_paid.sort_values(
    by="ConvertedCompYearly",
    ascending=False
)
```

In [68]:
```
top_5 = top_paid.head(5)
```

In [69]:
```
top_5[["ConvertedCompYearly", "EdLevel"]]
```

Out[69]:

	ConvertedCompYearly	EdLevel
34267	50000000.0	Associate degree (A.A., A.S., etc.)
28700	33552715.0	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)
43143	18387548.0	Associate degree (A.A., A.S., etc.)
35353	15430267.0	Bachelor’s degree (B.A., B.S., B.Eng., etc.)
45971	13921760.0	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)

In [ ]:
```
#*Завдання 9. Аналіз популярності Python по віковим категоріям
```

In [70]:
```
df["Age"].value_counts(dropna=False)
```

Out[70]:
```
Age
25-34 years old      16519
35-44 years old      13241
18-24 years old       9210
45-54 years old       6275
55-64 years old       2626
65 years or older      942
Prefer not to say      378
Name: count, dtype: int64
```

In [71]:
```
python_mask = df["LanguageHaveWorkedWith"].str.contains("Python", na=False)
```

In [72]:
```
age_python_df = df[["Age", "LanguageHaveWorkedWith"]].copy()
age_python_df["is_python"] = python_mask
```

In [73]:
```
result = (
    age_python_df
    .dropna(subset=["Age"])
    .groupby("Age")["is_python"]
    .mean()
    .reset_index()
)
```

In [74]:
```
result["python_percentage"] = (result["is_python"] * 100).round(2)
result = result[["Age", "python_percentage"]]
result
```

Out[74]:

	Age	python_percentage
0	18-24 years old	40.00
1	25-34 years old	36.94
2	35-44 years old	36.72
3	45-54 years old	38.63
4	55-64 years old	37.24
5	65 years or older	31.63
6	Prefer not to say	31.22

In [ ]:
```
#*Завдання 10. Аналіз індустрій серед високооплачуваних віддалених працівників
```

In [11]:
```
q75 = df["ConvertedCompYearly"].quantile(0.75)
```

In [12]:
```
high_paid_mask = df["ConvertedCompYearly"] >= q75
remote_mask = df["RemoteWork"] == "Remote"
```

In [16]:
```
filtered_df = df[high_paid_mask & remote_mask]
```

In [17]:
```
industry_counts = (
    filtered_df["Industry"]
    .dropna()
    .value_counts()
)

industry_counts.head(5)
```

Out[17]:
```
Industry
Software Development                          1186
Fintech                                        190
Healthcare                                     188
Other:                                         176
Internet, Telecomm or Information Services     138
Name: count, dtype: int64
```

In [ ]: