DATA SCIENCE JOB SALARIES VERİ SETİ İNCELEMESİ

Veri seti hikayesi

https://www.kaggle.com/datasets/ruchi798/data-science-job-salaries?resource=download Veri seti veri bilimi alanında çalışan insanların yıl boyunca aldıkları maaş bilgisini içermektedir. Aşağıda veri setinde bulunan genel bilgiler verilmektedir.

- work_year (maaşın ödendiği yıl)
- experience_level (Deneyim seviyesi EN-Başlangıç seviyesi, MI-Orta seviye, SE-Kıdemli seviye, Uzman EX-yönetici seviyesi)
- employment_type (Rol içi istihdam türü. PT-yarı zamanlı, FT-Tam zamanlı, CT-sözleşmeli, FL serbest çalışan)
- job_title (Yıl boyunca çalışılan rol)
- salary (Ödenen toplam brüt maaş tutarı)
- salary_currency (Ödenen maaşın para birimi ISO 4217 para birimi kodudur.)
- salary_in_usd (ABD doları cinsinden maaş (Döviz kurunun ilgili yılın ortalama ABD doları kuruna bölünmesiyle elde edilen değer, fxdata.foorilla.com'dur).)
- employee_residence (Çalışanın çalışma yılı boyunca ikamet ettiği birincil ülke ISO 3166 ülke kodudur.)
- remote_radio (Uzaktan yapılan toplam iş miktarı, olası değerler şu şekildedir: 0 Uzaktan çalışma yok (%20'den az) 50 Kısmen uzaktan 100 Tamamen uzaktan (%80'den fazla))
- company_location (İşverenin merkez ofisinin veya sözleşmeli şubesinin bulunduğu ülke ISO 3166 ülke kodu olarak belirtilir)
- company_size (Yıl boyunca şirkette çalışan ortalama kişi sayısı: S 50'den az çalışan (küçük) M 50 ila 250 çalışan (orta) L 250'den fazla çalışan (büyük))

1.Veri seti hakkında genel bilgiler

Veri setini dosyadan okuma işlemi yapılır. Bu işlem için Python pandas kütüphanesi kullanılmaktadır. Pandas kütüphane olarak veri bilimi ve makine öğrenmesinde sıkça kullanılan bir araçtır.Pandas veri dosyası okuma (read_csv,read_excel), tablo yapısı oluşturmak(DataFrame), eksik-hatalı verileri temizlemek (dropna,fillna), veriyi gruplamak ve analiz etmek (groupby, agg), veriyi görselleştirmeye hazırlamak (sort,filter,pivot) gibi birçok alanda kullanılmaktadır.

In [2]:
```
import pandas as pd
ds_salaries=pd.read_csv("archive/ds_salaries.csv") #veri seti dosyadan okundu
```

In [3]:
```
ds_salaries
```

Out[3]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	sa
0	0	2020	MI	FT	Data Scientist	70000	EUR	
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	sa
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	
3	3	2020	MI	FT	Product Data Analyst	20000	USD	
4	4	2020	SE	FT	Machine Learning Engineer	150000	USD	
...	...	...	...	...	...	...	...	...
602	602	2022	SE	FT	Data Engineer	154000	USD	
603	603	2022	SE	FT	Data Engineer	126000	USD	
604	604	2022	SE	FT	Data Analyst	129000	USD	
605	605	2022	SE	FT	Data Analyst	150000	USD	
606	606	2022	MI	FT	AI Scientist	200000	USD	

607 rows × 12 columns

In [4]:
```
df=ds_salaries.copy() # orijinal veri setinin bozulmaması için bir kopya oluşturuldu
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Unnamed: 0          607 non-null    int64 
 1   work_year           607 non-null    int64 
 2   experience_level    607 non-null    object
 3   employment_type     607 non-null    object
 4   job_title           607 non-null    object
 5   salary              607 non-null    int64 
 6   salary_currency     607 non-null    object
 7   salary_in_usd       607 non-null    int64 
 8   employee_residence  607 non-null    object
 9   remote_ratio        607 non-null    int64 
 10  company_location    607 non-null    object
 11  company_size        607 non-null    object
dtypes: int64(5), object(7)
memory usage: 57.0+ KB
```

In [5]:
```
df.head()
```

Out[5]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	sala
0	0	2020	MI	FT	Data Scientist	70000	EUR	
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	
3	3	2020	MI	FT	Product Data Analyst	20000	USD	
4	4	2020	SE	FT	Machine Learning Engineer	150000	USD	

In [6]:
```
df.tail()
```

Out[6]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	sa
602	602	2022	SE	FT	Data Engineer	154000	USD	
603	603	2022	SE	FT	Data Engineer	126000	USD	
604	604	2022	SE	FT	Data Analyst	129000	USD	
605	605	2022	SE	FT	Data Analyst	150000	USD	
606	606	2022	MI	FT	AI Scientist	200000	USD	

1.1 Veri setindeki datatype'ı uygun bir veri tipine dönüştürme

Veri analizi yapılırken doğru veri tipleri ile çalışmak makine öğrenmesi ve filtreleme, gruplandırma ve görselleştirmenin doğru olması açısından çok önemlidir. Veri analizinde 5 temel veri tipi vardır:

- Numerical (int,float)
- Categorical (object,category)
- Text verileri (genellikle object türündedir.)
- Datetime (datetime64)
- Boolean (true,false,bool tipinde)

In [7]:
```
df.rename(columns={"Unnamed: 0":"unnamed"} , inplace=True)
```

In [8]:
```
df.head()
```

Out[8]:

	unnamed	work_year	experience_level	employment_type	job_title	salary	salary_currency	salar
0	0	2020	MI	FT	Data Scientist	70000	EUR	
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	
3	3	2020	MI	FT	Product Data Analyst	20000	USD	
4	4	2020	SE	FT	Machine Learning Engineer	150000	USD	

In [9]:
```
df.experience_level=pd.Categorical(df.experience_level)
df.employment_type=pd.Categorical(df.employment_type)
df.job_title=pd.Categorical(df.job_title)
df.salary_currency=pd.Categorical(df.salary_currency)
df.employee_residence=pd.Categorical(df.employee_residence)
df.company_location=pd.Categorical(df.company_location)
df.company_size=pd.Categorical(df.company_size)
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype   
---  ------              --------------  -----   
 0   unnamed             607 non-null    int64   
 1   work_year           607 non-null    int64   
 2   experience_level    607 non-null    category
 3   employment_type     607 non-null    category
 4   job_title           607 non-null    category
 5   salary              607 non-null    int64   
 6   salary_currency     607 non-null    category
 7   salary_in_usd       607 non-null    int64   
 8   employee_residence  607 non-null    category
 9   remote_ratio        607 non-null    int64   
 10  company_location    607 non-null    category
 11  company_size        607 non-null    category
dtypes: category(7), int64(5)
memory usage: 36.6 KB
```

2. Veri seti ile ilgili istatistikler

In [11]:
```
df.describe()
```

Out[11]:

	unnamed	work_year	salary	salary_in_usd	remote_ratio
count	607.000000	607.000000	6.070000e+02	607.000000	607.00000
mean	303.000000	2021.405272	3.240001e+05	112297.869852	70.92257
std	175.370085	0.692133	1.544357e+06	70957.259411	40.70913
min	0.000000	2020.000000	4.000000e+03	2859.000000	0.00000
25%	151.500000	2021.000000	7.000000e+04	62726.000000	50.00000
50%	303.000000	2022.000000	1.150000e+05	101570.000000	100.00000
75%	454.500000	2022.000000	1.650000e+05	150000.000000	100.00000
max	606.000000	2022.000000	3.040000e+07	600000.000000	100.00000

In [12]:
```
df.describe().T # Burada sayısal veriler ile ilgili genel bir istatistik çıkarıldı.
```

Out[12]:

	count	mean	std	min	25%	50%	75%	max
unnamed	607.0	303.000000	1.753701e+02	0.0	151.5	303.0	454.5	606.0
work_year	607.0	2021.405272	6.921330e-01	2020.0	2021.0	2022.0	2022.0	2022.0
salary	607.0	324000.062603	1.544357e+06	4000.0	70000.0	115000.0	165000.0	30400000.0
salary_in_usd	607.0	112297.869852	7.095726e+04	2859.0	62726.0	101570.0	150000.0	600000.0
remote_ratio	607.0	70.922570	4.070913e+01	0.0	50.0	100.0	100.0	100.0

3. Eksik değerlerin gözlemlenmesi

In [13]:
```
df.isnull().values.any()
```

Out[13]:
```
False
```

4. Kategorik değişkenlerin incelenmesi

In [14]:
```
katdf=df.select_dtypes(include=["category"]) # categorical değişkenlerin neler olduğ
```

In [15]:
```
katdf.head()
```

Out[15]:

	experience_level	employment_type	job_title	salary_currency	employee_residence	company_locati
0	MI	FT	Data Scientist	EUR	DE	
1	SE	FT	Machine Learning Scientist	USD	JP	
2	SE	FT	Big Data Engineer	GBP	GB	
3	MI	FT	Product Data Analyst	USD	HN	I

	experience_level	employment_type	job_title	salary_currency	employee_residence	company_locati
4	SE	FT	Machine Learning Engineer	USD	US	

In [16]:
```
katdf.experience_level.unique() # categorical değişkenlerin hangi kategorilerden olu
```

Out[16]:
```
['MI', 'SE', 'EN', 'EX']
Categories (4, object): ['EN', 'EX', 'MI', 'SE']
```

In [17]:
```
katdf.experience_level.value_counts() # kategorilerdeki dağılımlar gözlemlendi. katd
# bu kod ketogori sayisini verir.
```

Out[17]:
```
SE    280
MI    213
EN     88
EX     26
Name: experience_level, dtype: int64
```

In [18]:
```
katdf.groupby(['experience_level', 'job_title']).size() #iki değişken için arama yap
```

Out[18]:
```
experience_level  job_title
EN                3D Computer Vision Researcher         0
                  AI Scientist                          4
                  Analytics Engineer                    0
                  Applied Data Scientist                1
                  Applied Machine Learning Scientist    1
                                                      ..
SE                Principal Data Engineer               2
                  Principal Data Scientist              5
                  Product Data Analyst                  0
                  Research Scientist                    5
                  Staff Data Scientist                  1
Length: 200, dtype: int64
```

In [21]:
```
ds_df = katdf[katdf['job_title'] == 'Data Scientist'] # data scientist ler için expe
ds_df.experience_level.value_counts()
```

Out[21]:
```
SE    61
MI    60
EN    22
EX     0
Name: experience_level, dtype: int64
```

In [42]:
```
ds_df = katdf[(katdf['job_title'] == 'Data Engineer')&(katdf['company_location']=="C
ds_df.experience_level.value_counts()
```

Out[42]:
```
EN    0
EX    0
MI    0
SE    0
Name: experience_level, dtype: int64
```

In [19]:
```
katdf.employment_type.unique()
```

Out[19]:
```
['FT', 'CT', 'PT', 'FL']
Categories (4, object): ['CT', 'FL', 'FT', 'PT']
```

In [20]:
```
katdf.employment_type.value_counts()
```

Out[20]:
```
FT    588
PT     10
CT      5
FL      4
Name: employment_type, dtype: int64
```

In [24]:
```
ds_df = katdf[katdf['job_title'] == 'Data Engineer']
ds_df.employment_type.value_counts()
```

Out[24]:
```
FT    129
PT      2
FL      1
CT      0
Name: employment_type, dtype: int64
```

In [29]:
```
katdf.job_title.unique()
```

Out[29]:
```
['Data Scientist', 'Machine Learning Scientist', 'Big Data Engineer', 'Product Data A
nalyst', 'Machine Learning Engineer', ..., 'ETL Developer', 'Head of Machine Learnin
g', 'NLP Engineer', 'Lead Machine Learning Engineer', 'Data Analytics Lead']
Length: 50
Categories (50, object): ['3D Computer Vision Researcher', 'AI Scientist', 'Analytics
Engineer', 'Applied Data Scientist', ..., 'Principal Data Scientist', 'Product Data A
nalyst', 'Research Scientist', 'Staff Data Scientist']
```

In [30]:
```
list(df["job_title"].cat.categories) # 50 kategorinin ne olduğu gözlemlendi. cat pan
```

Out[30]:
```
['3D Computer Vision Researcher',
 'AI Scientist',
 'Analytics Engineer',
 'Applied Data Scientist',
 'Applied Machine Learning Scientist',
 'BI Data Analyst',
 'Big Data Architect',
 'Big Data Engineer',
 'Business Data Analyst',
 'Cloud Data Engineer',
 'Computer Vision Engineer',
 'Computer Vision Software Engineer',
 'Data Analyst',
 'Data Analytics Engineer',
 'Data Analytics Lead',
 'Data Analytics Manager',
 'Data Architect',
 'Data Engineer',
 'Data Engineering Manager',
 'Data Science Consultant',
 'Data Science Engineer',
 'Data Science Manager',
 'Data Scientist',
 'Data Specialist',
 'Director of Data Engineering',
 'Director of Data Science',
 'ETL Developer',
 'Finance Data Analyst',
 'Financial Data Analyst',
 'Head of Data',
 'Head of Data Science',
 'Head of Machine Learning',
```

```
 'Lead Data Analyst',
 'Lead Data Engineer',
 'Lead Data Scientist',
 'Lead Machine Learning Engineer',
 'ML Engineer',
 'Machine Learning Developer',
 'Machine Learning Engineer',
 'Machine Learning Infrastructure Engineer',
 'Machine Learning Manager',
 'Machine Learning Scientist',
 'Marketing Data Analyst',
 'NLP Engineer',
 'Principal Data Analyst',
 'Principal Data Engineer',
 'Principal Data Scientist',
 'Product Data Analyst',
 'Research Scientist',
 'Staff Data Scientist']
```

In [46]:
```
katdf.job_title.value_counts()
```

Out[46]:
```
Data Scientist                             143
Data Engineer                              132
Data Analyst                                97
Machine Learning Engineer                   41
Research Scientist                          16
Data Science Manager                        12
Data Architect                              11
Big Data Engineer                            8
Machine Learning Scientist                   8
Director of Data Science                     7
AI Scientist                                 7
Principal Data Scientist                     7
Data Science Consultant                      7
Data Analytics Manager                       7
Computer Vision Engineer                     6
BI Data Analyst                              6
ML Engineer                                  6
Lead Data Engineer                           6
Data Engineering Manager                     5
Business Data Analyst                        5
Applied Data Scientist                       5
Head of Data                                 5
Head of Data Science                         4
Data Analytics Engineer                      4
Applied Machine Learning Scientist           4
Analytics Engineer                           4
Machine Learning Developer                   3
Machine Learning Infrastructure Engineer     3
Lead Data Scientist                          3
Lead Data Analyst                            3
Data Science Engineer                        3
Principal Data Engineer                      3
Computer Vision Software Engineer            3
Principal Data Analyst                       2
Financial Data Analyst                       2
ETL Developer                                2
Director of Data Engineering                 2
Product Data Analyst                         2
Cloud Data Engineer                          2
NLP Engineer                                 1
Marketing Data Analyst                       1
3D Computer Vision Researcher                1
```

```
Machine Learning Manager                    1
Lead Machine Learning Engineer              1
Head of Machine Learning                    1
Finance Data Analyst                        1
Data Specialist                             1
Data Analytics Lead                         1
Big Data Architect                          1
Staff Data Scientist                        1
Name: job_title, dtype: int64
```

In [33]:
```
katdf.salary_currency.unique()
```

Out[33]:
```
['EUR', 'USD', 'GBP', 'HUF', 'INR', ..., 'CLP', 'BRL', 'TRY', 'AUD', 'CHF']
Length: 17
Categories (17, object): ['AUD', 'BRL', 'CAD', 'CHF', ..., 'PLN', 'SGD', 'TRY', 'US
D']
```

In [34]:
```
list(df["salary_currency"].cat.categories)
```

Out[34]:
```
['AUD',
 'BRL',
 'CAD',
 'CHF',
 'CLP',
 'CNY',
 'DKK',
 'EUR',
 'GBP',
 'HUF',
 'INR',
 'JPY',
 'MXN',
 'PLN',
 'SGD',
 'TRY',
 'USD']
```

In [48]:
```
katdf.salary_currency.value_counts()
```

Out[48]:
```
USD    398
EUR     95
GBP     44
INR     27
CAD     18
JPY      3
PLN      3
TRY      3
CNY      2
DKK      2
BRL      2
HUF      2
MXN      2
SGD      2
AUD      2
CHF      1
CLP      1
Name: salary_currency, dtype: int64
```

In [35]:
```
katdf.employee_residence.unique()
```

Out[35]:
```
['DE', 'JP', 'GB', 'HN', 'US', ..., 'EE', 'AU', 'BO', 'IE', 'CH']
Length: 57
Categories (57, object): ['AE', 'AR', 'AT', 'AU', ..., 'TR', 'UA', 'US', 'VN']
```

In [50]:
```
katdf.employee_residence.value_counts()
```

Out[50]:
```
US    332
GB     44
IN     30
CA     29
DE     25
FR     18
ES     15
GR     13
JP      7
PK      6
BR      6
PT      6
NL      5
IT      4
PL      4
RU      4
TR      3
AE      3
VN      3
AT      3
AU      3
BE      2
SI      2
MX      2
RO      2
SG      2
NG      2
HU      2
DK      2
TN      1
CL      1
RS      1
UA      1
BG      1
PR      1
BO      1
CH      1
PH      1
NZ      1
EE      1
MY      1
DZ      1
MT      1
MD      1
LU      1
KE      1
CN      1
JE      1
CO      1
IR      1
AR      1
CZ      1
IE      1
HR      1
HN      1
HK      1
```

```
IQ      1
Name: employee_residence, dtype: int64
```

In [37]:
```
katdf.company_location.unique()
```

Out[37]:
```
['DE', 'JP', 'GB', 'HN', 'US', ..., 'DZ', 'EE', 'MY', 'AU', 'IE']
Length: 50
Categories (50, object): ['AE', 'AS', 'AT', 'AU', ..., 'TR', 'UA', 'US', 'VN']
```

In [52]:
```
katdf.company_location.value_counts()
```

Out[52]:
```
US    355
GB     47
CA     30
DE     28
IN     24
FR     15
ES     14
GR     11
JP      6
PL      4
PT      4
NL      4
AT      4
MX      3
LU      3
TR      3
PK      3
AE      3
AU      3
BR      3
DK      3
CN      2
CZ      2
BE      2
SI      2
RU      2
NG      2
IT      2
CH      2
NZ      1
CL      1
EE      1
SG      1
UA      1
RO      1
CO      1
MY      1
DZ      1
MT      1
MD      1
KE      1
IR      1
IQ      1
AS      1
IL      1
IE      1
HU      1
HR      1
HN      1
VN      1
Name: company_location, dtype: int64
```

In [41]:
```
katdf.company_size.unique()
```

Out[41]:
```
['L', 'S', 'M']
Categories (3, object): ['L', 'M', 'S']
```

In [54]:
```
katdf.company_size.value_counts()
```

Out[54]:
```
M    326
L    198
S     83
Name: company_size, dtype: int64
```

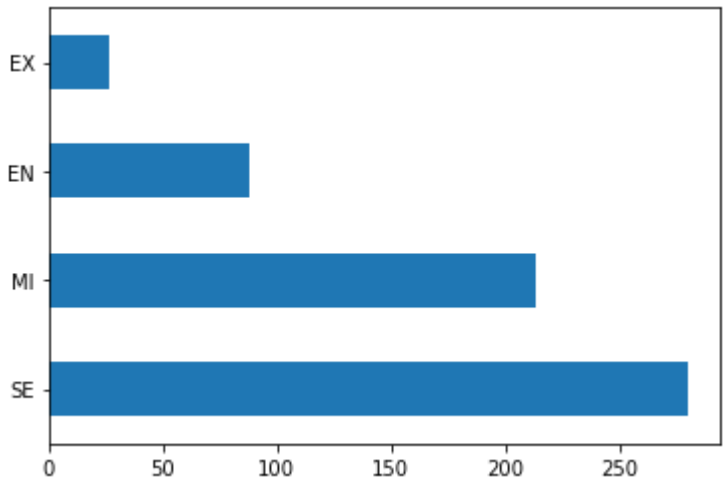
4.1. Kategorik değişkenlerin görselleştirilmesi

Veri görselleştirilmesinde birçok ara kullaılmaktadır bunlardan bazıları plot, bar, his ve scatter dır. bu türleirn açıklanması ve avantajları aşağıda açıklanmaktadır.

- plot() : çizgi grafiğidir. zamana bağlı değişimleri göstermek,grupların ortalama değerlerinin karşılaştırılması ve trend analizlerinde kullanılabilir. avantajı zamanla değişimii net göstermesidir.genellikle x ekseni zaman veya sıralı bir kategori iken y ekseni sayısal değerdir.
- bar() : sütun grafiğidir. kategorik değişkenlerin karşılaştırılmasında kullanılır. genelde x ekseni kategorik y ekseni sayısal olur.avantaj olarak sayısal karşılaştırmalarda güçlüdür.
- hist() : bir değişkenin frekans dağılımını göstermek,hangi aralıkta kaç kişi oldğunu görmek ,verilerin dağılımı normal veya çarpık mı diye kontrol etmek için kullanılır. x ekseni veri değerleri iken (aralık), y ekseni frekansı yani kaç kişinin o aralıkta olduğunu belirtir. avantajları dağılımı gözlemlerken aykırı değerlerin tepitinde yardımcı olur.
- scatter() : iki sayısal değer arasındaki ilişkiyi görselleştirmek, korelasyon analizi,regresyon modelleri öncesi görsel analiz yapmak için kullanılır.x ve y ekseni sayısaldır. her veri noktası grafikte bir nokta olarak gösterilir.avantaj olarak sayısal değişkenler arasındaki ilişkiyi açıkca göstrirken gruplar ve kümeler fark edilebilir.

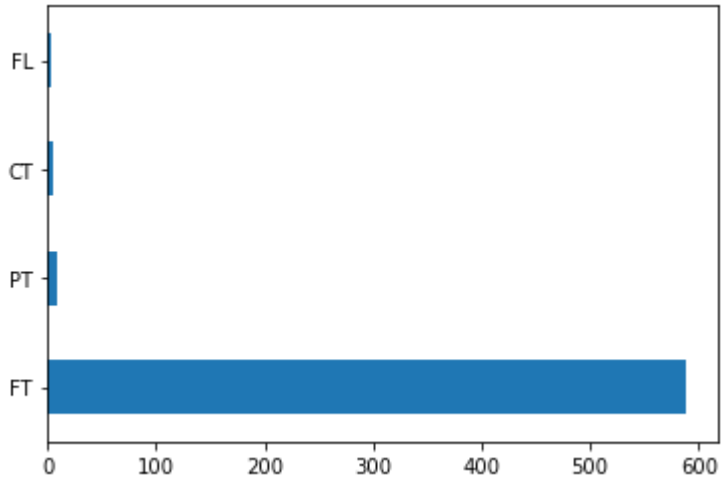
In [56]:
```
df["experience_level"].value_counts().plot.barh()
```

Out[56]:
```
<AxesSubplot:>
```



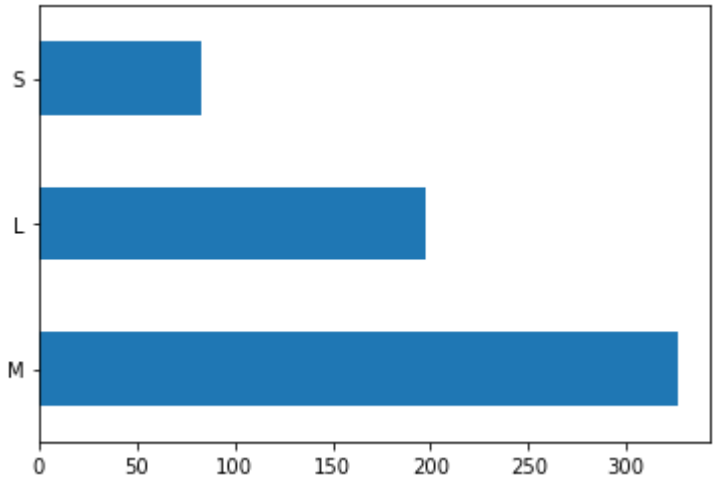
In [58]:
```
df["employment_type"].value_counts().plot.barh()
```

Out[58]:
```
<AxesSubplot:>
```



In [60]:
```
df["company_size"].value_counts().plot.barh()
```

Out[60]:
```
<AxesSubplot:>
```



5.Sürekli değişkenlerin incelenmesi

In [61]:
```
numdf=df.select_dtypes(include=["int64"])
```

In [62]:
```
numdf.head()
```

Out[62]:

	unnamed	work_year	salary	salary_in_usd	remote_ratio
0	0	2020	70000	79833	0
1	1	2020	260000	260000	0
2	2	2020	85000	109024	50
3	3	2020	20000	20000	0
4	4	2020	150000	150000	50

In [64]:
```
numdf.describe()
```

Out[64]:

	unnamed	work_year	salary	salary_in_usd	remote_ratio
count	607.000000	607.000000	6.070000e+02	607.000000	607.00000
mean	303.000000	2021.405272	3.240001e+05	112297.869852	70.92257
std	175.370085	0.692133	1.544357e+06	70957.259411	40.70913
min	0.000000	2020.000000	4.000000e+03	2859.000000	0.00000
25%	151.500000	2021.000000	7.000000e+04	62726.000000	50.00000
50%	303.000000	2022.000000	1.150000e+05	101570.000000	100.00000
75%	454.500000	2022.000000	1.650000e+05	150000.000000	100.00000
max	606.000000	2022.000000	3.040000e+07	600000.000000	100.00000

In [65]:
```
numdf.describe().T
```

Out[65]:

	count	mean	std	min	25%	50%	75%	max
unnamed	607.0	303.000000	1.753701e+02	0.0	151.5	303.0	454.5	606.0
work_year	607.0	2021.405272	6.921330e-01	2020.0	2021.0	2022.0	2022.0	2022.0
salary	607.0	324000.062603	1.544357e+06	4000.0	70000.0	115000.0	165000.0	30400000.0
salary_in_usd	607.0	112297.869852	7.095726e+04	2859.0	62726.0	101570.0	150000.0	600000.0
remote_ratio	607.0	70.922570	4.070913e+01	0.0	50.0	100.0	100.0	100.0

6. Veri seti üzerinde yapılan incelemeler

Deneyimin maaşlar üzerindeki etkisi

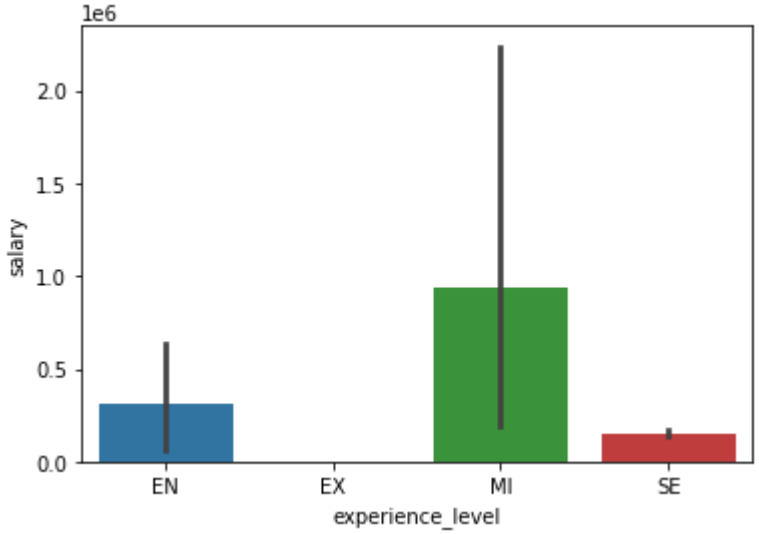
In [70]:
```
df[df["job_title"]=="Data Scientist"].pivot_table("salary",index=("experience_level"
# veriler experience level'a göre gruplandırıldı.
```

Out[70]:

	salary
experience_level	
EN	311231.818182
MI	939987.166667
SE	154874.098361

In [71]:
```
import seaborn as sns
sns.barplot(x="experience_level",y="salary",data=df[df["job_title"]=="Data Scientist
```

Out[71]:
```
<AxesSubplot:xlabel='experience_level', ylabel='salary'>
```



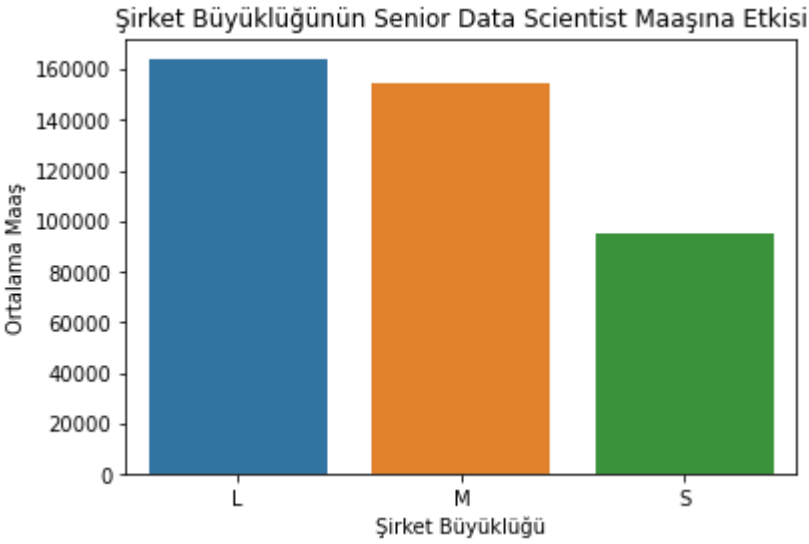
Şirket büyüklüğünün senior level çalışanlan data scienistlerin maaşı üzerindeki etkisi

In [74]:
```
senior_ds = df[(df["job_title"] == "Data Scientist") & (df["experience_level"] == "S
```

In [75]:
```
senior_ds.groupby("company_size")["salary"].mean()
```

Out[75]:
```
company_size
L    163868.750000
M    154312.093023
S     95000.000000
Name: salary, dtype: float64
```

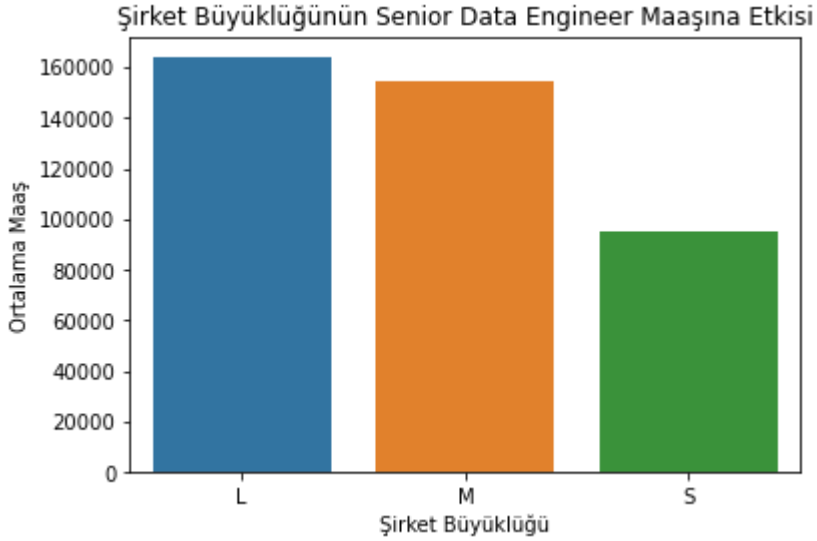
In [76]:
```
import matplotlib.pyplot as plt
sns.barplot(x="company_size",y="salary",data=senior_ds,ci=None,)
plt.title("Şirket Büyüklüğünün Senior Data Scientist Maaşına Etkisi")
plt.xlabel("Şirket Büyüklüğü")
plt.ylabel("Ortalama Maaş")
plt.show()
```



Şirket büyüklüğünün senior level çalışanlan data engineerların maaşı üzerindeki etkisi

In [78]:
```
senior_de=df[(df["job_title"]=="Data Engineer") &(df["experience_level"]=="SE")]
```

In [79]:
```
import matplotlib.pyplot as plt
sns.barplot(x="company_size",y="salary",data=senior_ds,ci=None,)
plt.title("Şirket Büyüklüğünün Senior Data Engineer Maaşına Etkisi")
plt.xlabel("Şirket Büyüklüğü")
plt.ylabel("Ortalama Maaş")
plt.show()
```

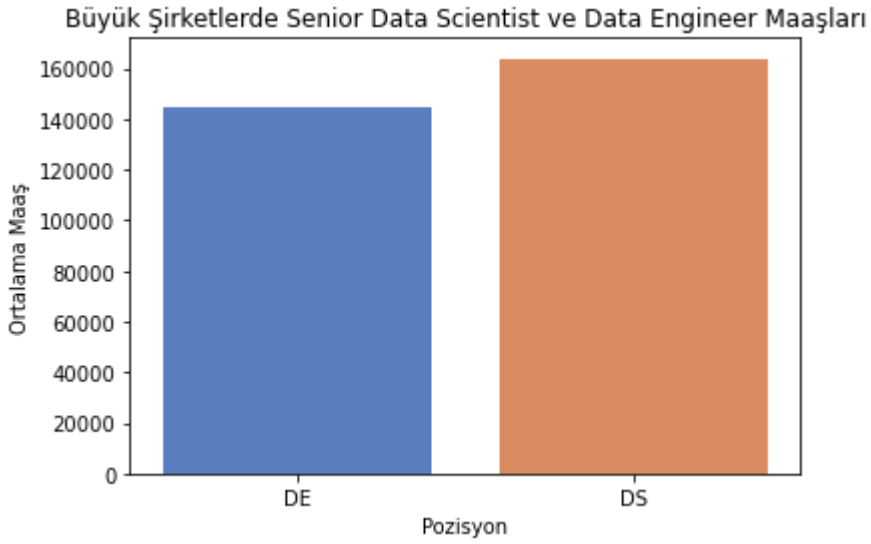


Büyük şirketlerde çalışan senior data scientist ve data engineer ların maaş karşılaştırması

In [96]:
```
senior_large = df[
    (df["company_size"] == "L") &
    (df["experience_level"] == "SE") &
    (df["job_title"].isin(["Data Scientist", "Data Engineer"]))
] # burada istediğimiz veri türlerini seçtik.
```

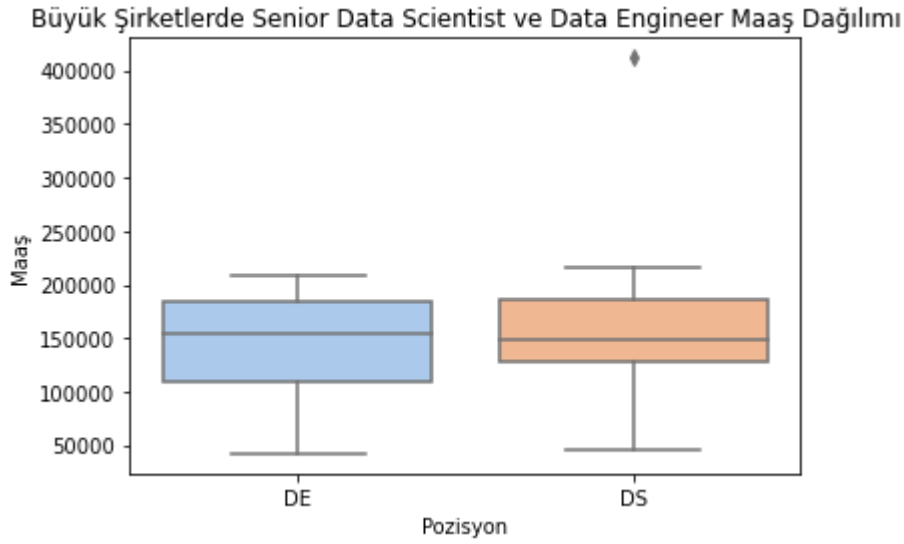
In [97]:
```
senior_large = senior_large.copy()
senior_large["job_title_short"] = senior_large["job_title"].map({
    "Data Scientist": "DS",
    "Data Engineer": "DE"
})
```

In [100…
```
sns.barplot(
    x="job_title_short",
    y="salary",
    data=senior_large,
    ci=None,
    palette="muted"
)

plt.title("Büyük Şirketlerde Senior Data Scientist ve Data Engineer Maaşları")
plt.xlabel("Pozisyon")
plt.ylabel("Ortalama Maaş")
plt.show()
```



In [99]:

```
sns.boxplot(
    x="job_title_short",
    y="salary",
    data=senior_large,
    palette="pastel"
)

plt.title("Büyük Şirketlerde Senior Data Scientist ve Data Engineer Maaş Dağılımı")
plt.xlabel("Pozisyon")
plt.ylabel("Maaş")
plt.show()
```

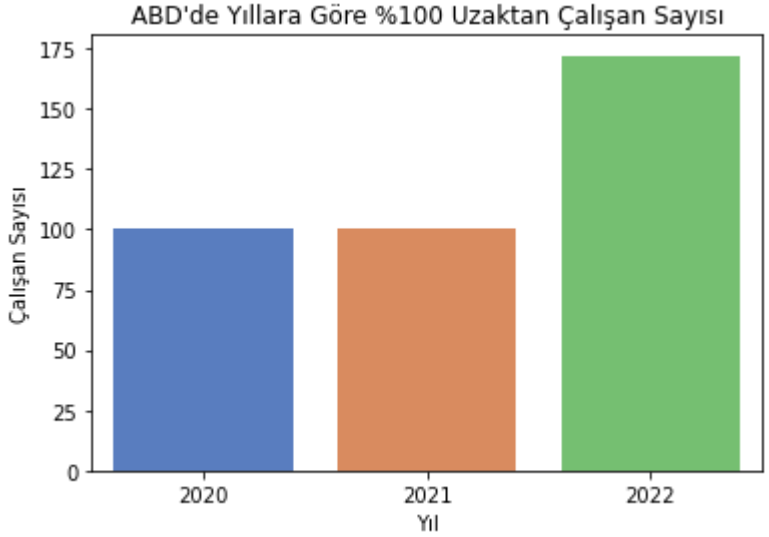


Amerikada yıllara göre uzaktan çalışma oranının ddeğişimi

In [46]:

```
company_us = df[(df["company_location"] == "US") & (df["remote_ratio"]== 100)]
sns.barplot(
    x="work_year",
    y="remote_ratio",
    data=company_us,
    ci=None,
    palette="muted"
)
remote_count_by_year = company_us["work_year"].value_counts().sort_index() # filtrel

sns.barplot(x=remote_count_by_year.index, y=remote_count_by_year.values, palette="mu
```

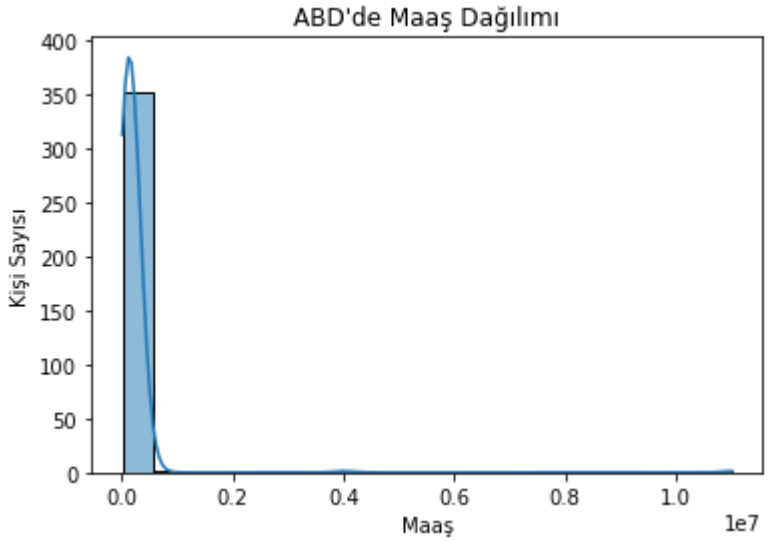
```
plt.title("ABD'de Yıllara Göre %100 Uzaktan Çalışan Sayısı")
plt.xlabel("Yıl")
plt.ylabel("Çalışan Sayısı")
plt.show()
```



Histogram grafiğinin kullanımı

In [56]:

```
sns.histplot(df[df["company_location"] == "US"]["salary"], bins=20, kde=True) #bins
#yani verinin eğrisel dağılım tahminini çizer
plt.title("ABD'de Maaş Dağılımı")
plt.xlabel("Maaş")
plt.ylabel("Kişi Sayısı")
plt.show()
```

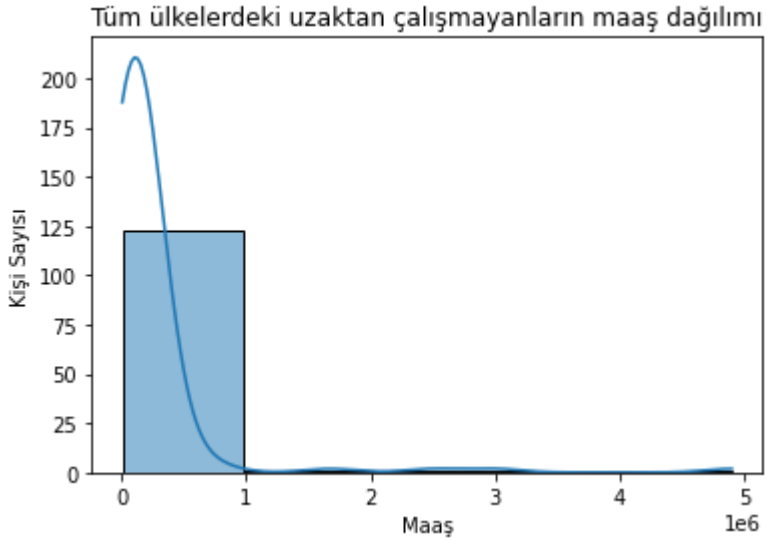


In [60]:

```
sns.histplot(df[df["remote_ratio"] == 100]["salary"], bins=5, kde=True)
plt.title("Tüm ülkelerdeki %100 uzaktan çalışanların maaş dağılımı")
plt.xlabel("Maaş")
plt.ylabel("Kişi Sayısı")
plt.show()
```

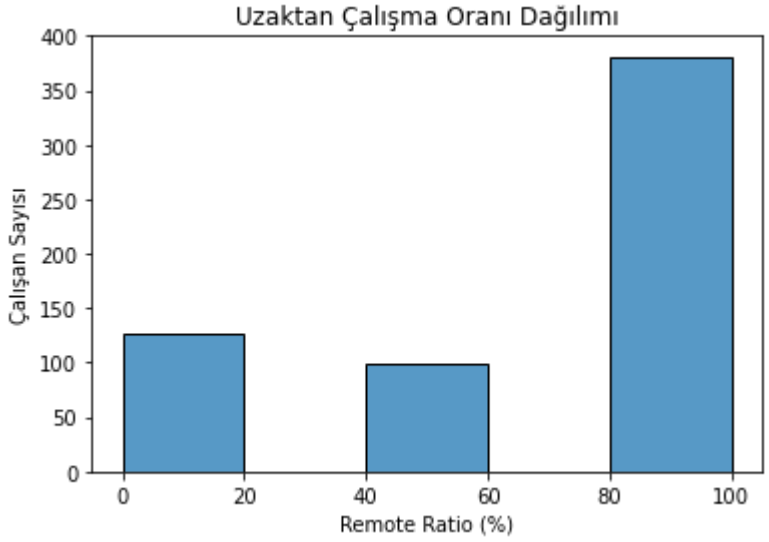
In [62]:

```
sns.histplot(df[df["remote_ratio"] == 0]["salary"], bins=5, kde=True)
plt.title("Tüm ülkelerdeki uzaktan çalışmayanların maaş dağılımı")
plt.xlabel("Maaş")
plt.ylabel("Kişi Sayısı")
plt.show()
```



In [63]:

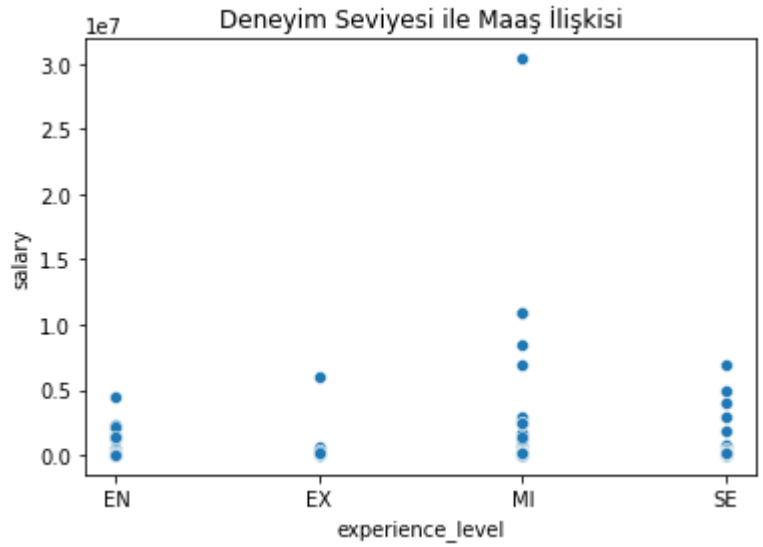
```
sns.histplot(df["remote_ratio"], bins=5)
plt.title("Uzaktan Çalışma Oranı Dağılımı")
plt.xlabel("Remote Ratio (%)")
plt.ylabel("Çalışan Sayısı")
plt.show()
```



Scatter kullanımı

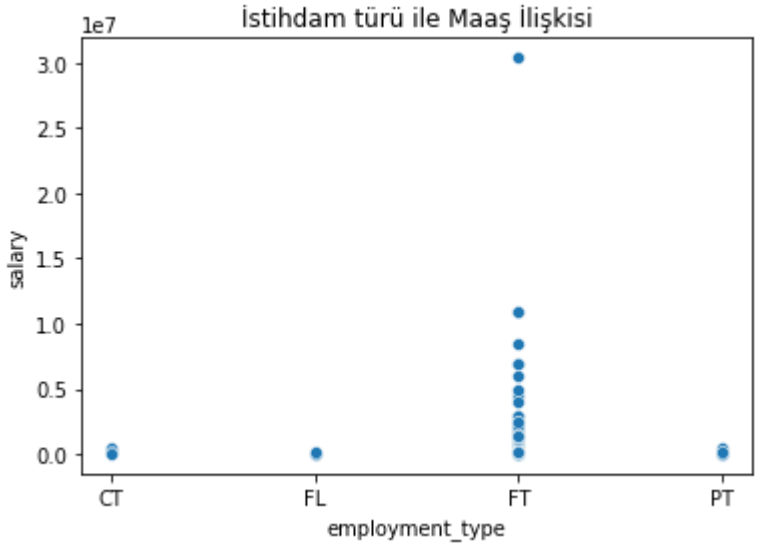
In [65]:
```
experience_map = {"EN": 1, "MI": 2, "SE": 3, "EX": 4} #kategorik verinin sayısal dön
df["experience_numeric"] = df["experience_level"].map(experience_map)

sns.scatterplot(x="experience_level", y="salary", data=df)
plt.title("Deneyim Seviyesi ile Maaş İlişkisi")
```

Out[65]:
```
Text(0.5, 1.0, 'Deneyim Seviyesi ile Maaş İlişkisi')
```



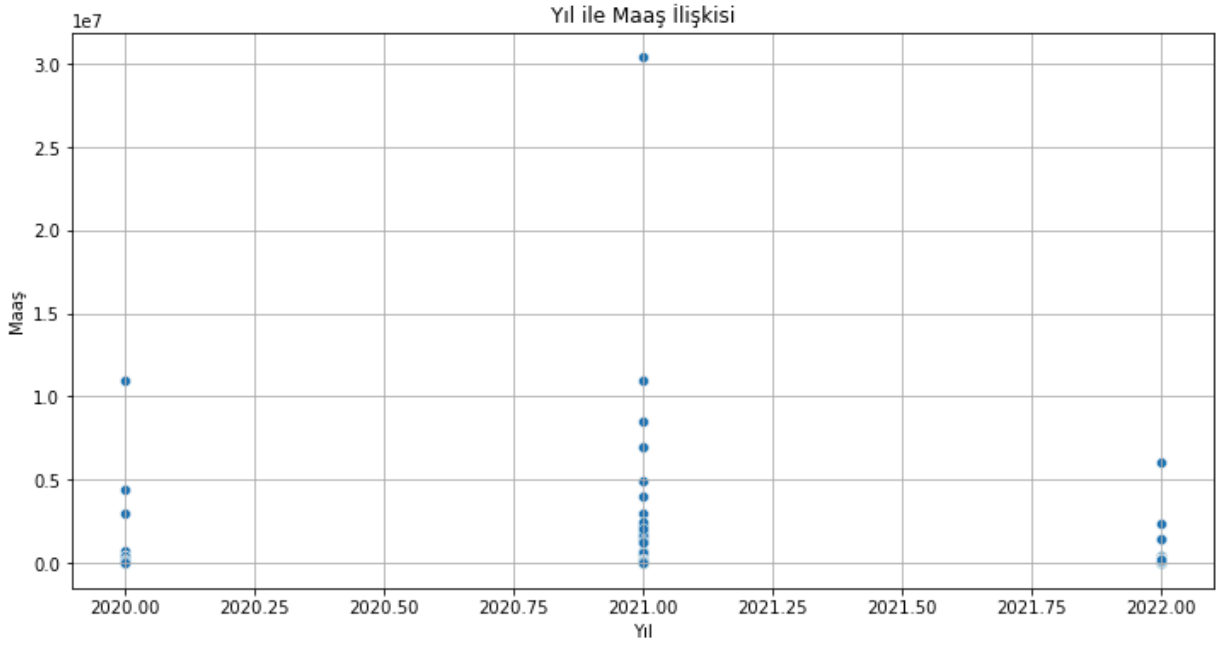
In [67]:
```
employmenttype_map = {"FT": 1, "PT": 2, "CT": 3, "FL": 4}
df["employmenttype_numeric"] = df["employment_type"].map(employmenttype_map)

sns.scatterplot(x="employment_type", y="salary", data=df)
plt.title("İstihdam türü ile Maaş İlişkisi")
```

Out[67]:
```
Text(0.5, 1.0, 'İstihdam türü ile Maaş İlişkisi')
```



In [73]:

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))  # Grafiği daha geniş yapar

sns.scatterplot(x="work_year", y="salary", data=df)
plt.title("Yıl ile Maaş İlişkisi")
plt.xlabel("Yıl")
plt.ylabel("Maaş")
plt.grid(True)
plt.show()
```

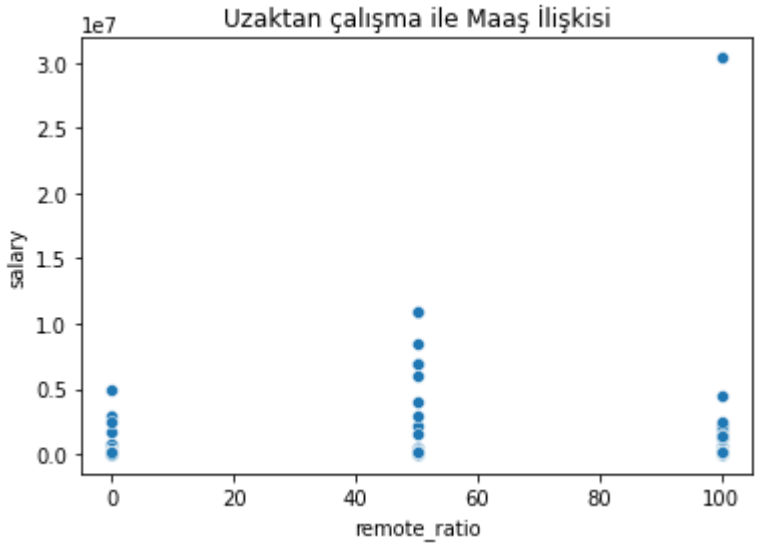


In [70]:

```
sns.scatterplot(x="remote_ratio", y="salary", data=df)
plt.title("Uzaktan çalışma ile Maaş İlişkisi")
```

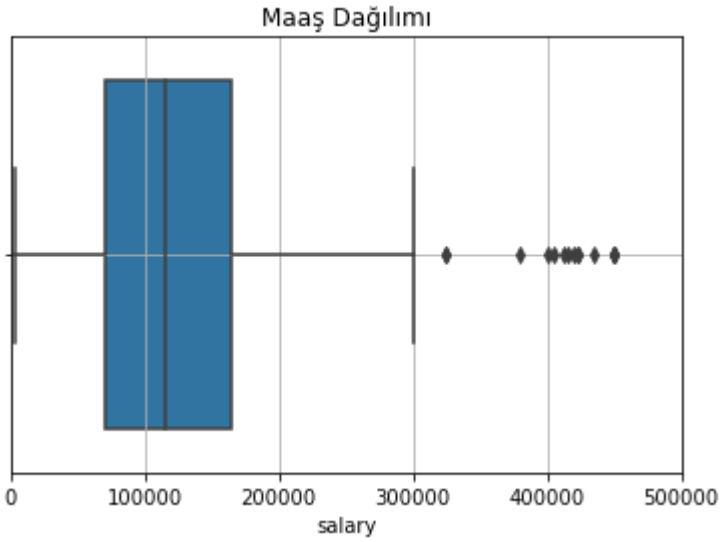
Out[70]:

```
Text(0.5, 1.0, 'Uzaktan çalışma ile Maaş İlişkisi')
```

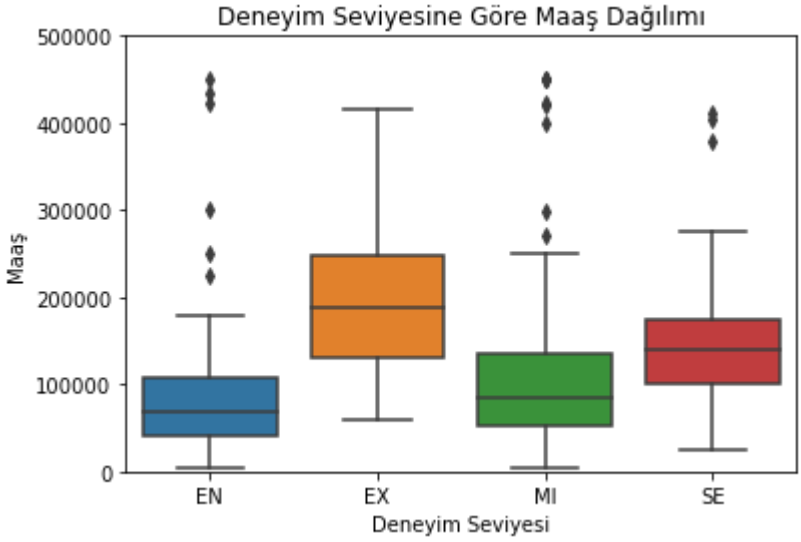


Boxplot kullanımı

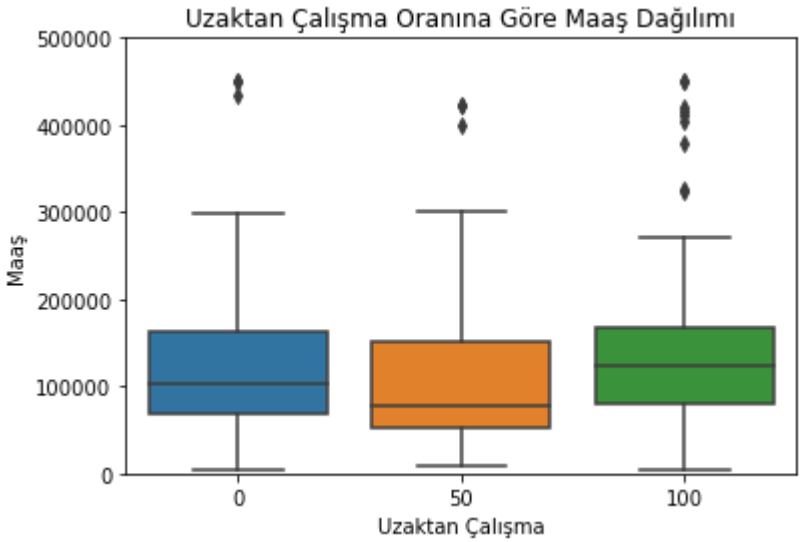
In [82]:
```
sns.boxplot(x=df["salary"])
plt.xlim(0, 500000)
plt.title("Maaş Dağılımı")
plt.grid(True)
plt.show()
```



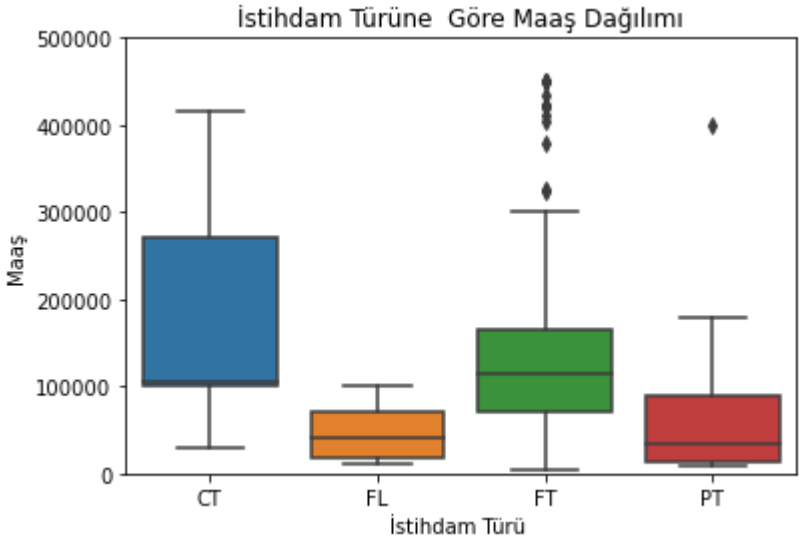
In [81]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x="experience_level", y="salary", data=df)
plt.ylim(0, 500000)
plt.title("Deneyim Seviyesine Göre Maaş Dağılımı")
plt.xlabel("Deneyim Seviyesi")
plt.ylabel("Maaş")
plt.show()
```



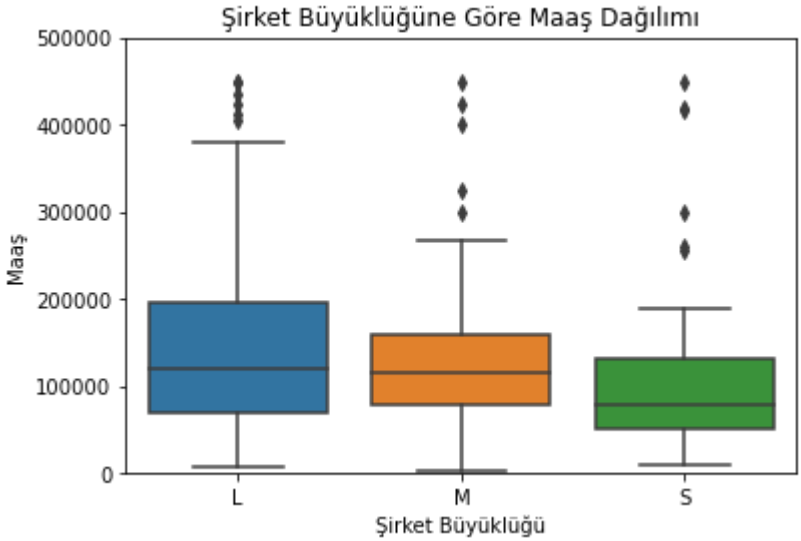
In [84]:
```
sns.boxplot(x="remote_ratio", y="salary", data=df)
plt.ylim(0, 500000)
plt.title("Uzaktan Çalışma Oranına Göre Maaş Dağılımı")
plt.xlabel("Uzaktan Çalışma")
plt.ylabel("Maaş")
plt.show()
```



In [87]:
```
sns.boxplot(x="employment_type", y="salary", data=df)
plt.ylim(0, 500000)
plt.title("İstihdam Türüne  Göre Maaş Dağılımı")
plt.xlabel("İstihdam Türü ")
plt.ylabel("Maaş")
plt.show()
```



In [88]:
```
sns.boxplot(x="company_size", y="salary", data=df)
plt.ylim(0, 500000)
plt.title("Şirket Büyüklüğüne Göre Maaş Dağılımı")
plt.xlabel("Şirket Büyüklüğü")
plt.ylabel("Maaş")
plt.show()
```



Aykırı gözlem analizi

In [91]:
```
df_say= df.select_dtypes(include="int64")
```

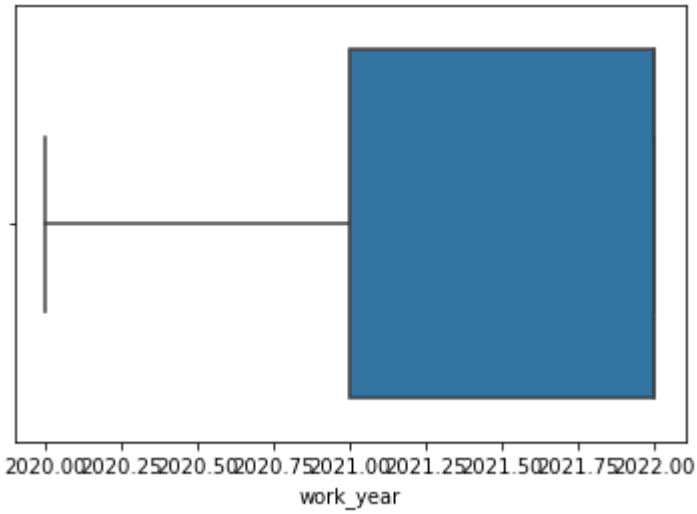
In [92]:
```
df_say.head()
```

Out[92]:

	unnamed	work_year	salary	salary_in_usd	remote_ratio
0	0	2020	70000	79833	0
1	1	2020	260000	260000	0
2	2	2020	85000	109024	50
3	3	2020	20000	20000	0
4	4	2020	150000	150000	50

In [95]:
```
df_work=df_say["work_year"]
sns.boxplot(x=df_work)
```

Out[95]:
```
<AxesSubplot:xlabel='work_year'>
```



In [97]:
```
Q1 = df_work.quantile(0.25)
Q3 = df_work.quantile(0.75)
IQR = Q3-Q1
print(Q1)
print(Q3)
print(IQR)
```

```
2021.0
2022.0
1.0
```

In [99]:
```
alt_sinir = Q1- 1.5*IQR
ust_sinir = Q3 + 1.5*IQR
print(alt_sinir)
print(ust_sinir)
```

```
2019.5
2023.5
```

In [100…
```
(df_work < alt_sinir) | (df_work > ust_sinir)
```

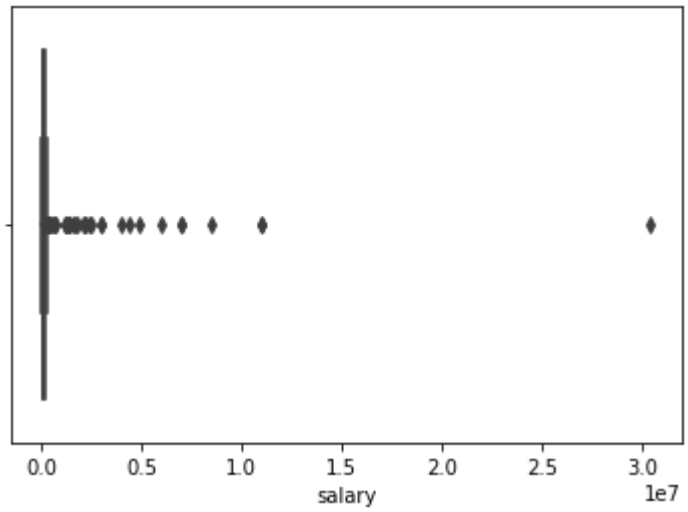
Out[100…
```
0      False
1      False
2      False
3      False
4      False
       ...
602    False
603    False
604    False
605    False
606    False
Name: work_year, Length: 607, dtype: bool
```

In [102…
```
aykiri_tf = (df_work < alt_sinir) | (df_work > ust_sinir)
df_work[aykiri_tf] #aykırı değerler gözlemlendi.
```

Out[102…
```
Series([], Name: work_year, dtype: int64)
```

In [103…
```
df_sl=df_say["salary"]
sns.boxplot(x=df_sl)
```

Out[103…
```
<AxesSubplot:xlabel='salary'>
```



In [105…
```
Q1 = df_sl.quantile(0.25)
Q3 = df_sl.quantile(0.75)
IQR = Q3-Q1
print(Q1)
print(Q3)
print(IQR)
```

```
70000.0
165000.0
95000.0
```

In [106…
```
alt_sinir = Q1- 1.5*IQR
ust_sinir = Q3 + 1.5*IQR
print(alt_sinir)
print(ust_sinir)
```

```
-72500.0
307500.0
```

In [107…
```
(df_sl < alt_sinir) | (df_sl > ust_sinir)
```

Out[107…
```
0      False
1      False
2      False
3      False
4      False
       ...
602    False
603    False
604    False
605    False
606    False
Name: work_year, Length: 607, dtype: bool
```

In [108…
```
aykiri_tf = (df_sl < alt_sinir) | (df_sl > ust_sinir)
df_sl[aykiri_tf]
```

Out[108…

```
7      11000000
11      3000000
16      4450000
18       423000
21       450000
25       325000
27       720000
33       450000
50       450000
63       412000
77       400000
92      1450000
94      2200000
97       450000
102    11000000
109     2250000
127      700000
129     3000000
136     7000000
137     8500000
157      423000
177    30400000
179      420000
180     1672000
197     1799997
198     4000000
213      435000
222     2500000
225      416000
230     1200000
239     1600000
244     1335000
252      600000
253     2100000
262     1250000
263     4900000
285     7000000
384     6000000
458     1400000
459     2400000
463     1400000
482      324000
519      380000
523      405000
Name: salary, dtype: int64
```