Employee Salaries for different job roles

Nombre Apellido1 Apellido2

Fecha

Introducción

En este proyecto se desarrolla en Python un análisis básico de datos sobre los sueldos que ganan distintos empleados según sus cometidos y experiencia, a lo largo de distintos negocios y zonas del mundo. La URL de referencia es la siguiente:

https://www.kaggle.com/datasets/inductiveanks/employee-salaries-for-different-job-roles (https://www.kaggle.com/datasets/inductiveanks/employee-salaries-for-different-job-roles)

En ella puede encontrarse información más detallada, así como una descripción precisa de cada columna. Seguidamente, te toca a ti hacer una breve introducción, completando el fragmento de letra en azul y desarrollándolo a tu antojo.

A partir de los datos proporcionados, he conseguido ... pero no he podido ...

Aunque al final de este notebook detallaré la calificación que calculo honestamente, globalmente, siguiendo las puntuaciones que se asigna a cada apartado, diría que he obtenido una nota de *** sobre 10.

Completa tus datos personales en la cabecera, bajo el rótulo inicial. Completa también el breve apartado anterior. Elimina este párrafo en verde. A partir de ahora, pon en azul los comentarios tuyos, dejando en negro los míos, del enunciado, y suprimiendo los fragmentos en verde, como éste, que son indicaciones pero que, una vez atendidas, deben suprimirse.

Librerías

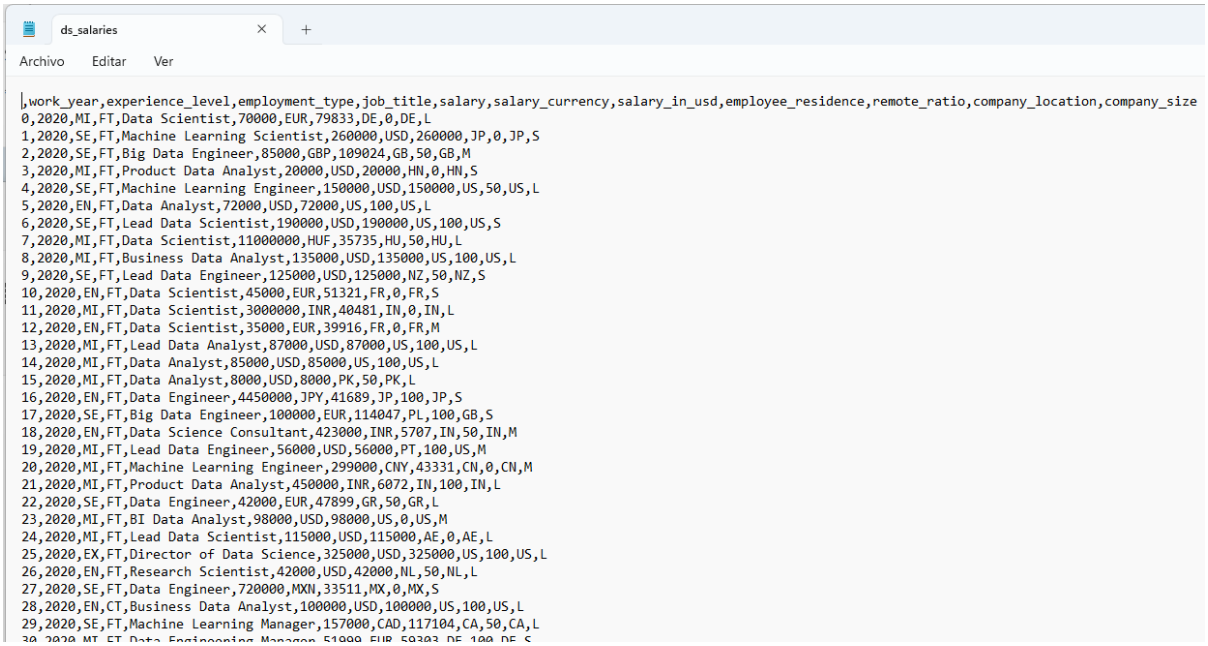
Pongamos todas las librerías necesarias al principio, tal como propone el estilo `pep-8`. Ej.: PEP 8 -- Style Guide for Python Code (https://www.python.org/dev/peps/pep-0008/).

In [1]:

```
# Esta celda debe ser completada por el estudiante.
```

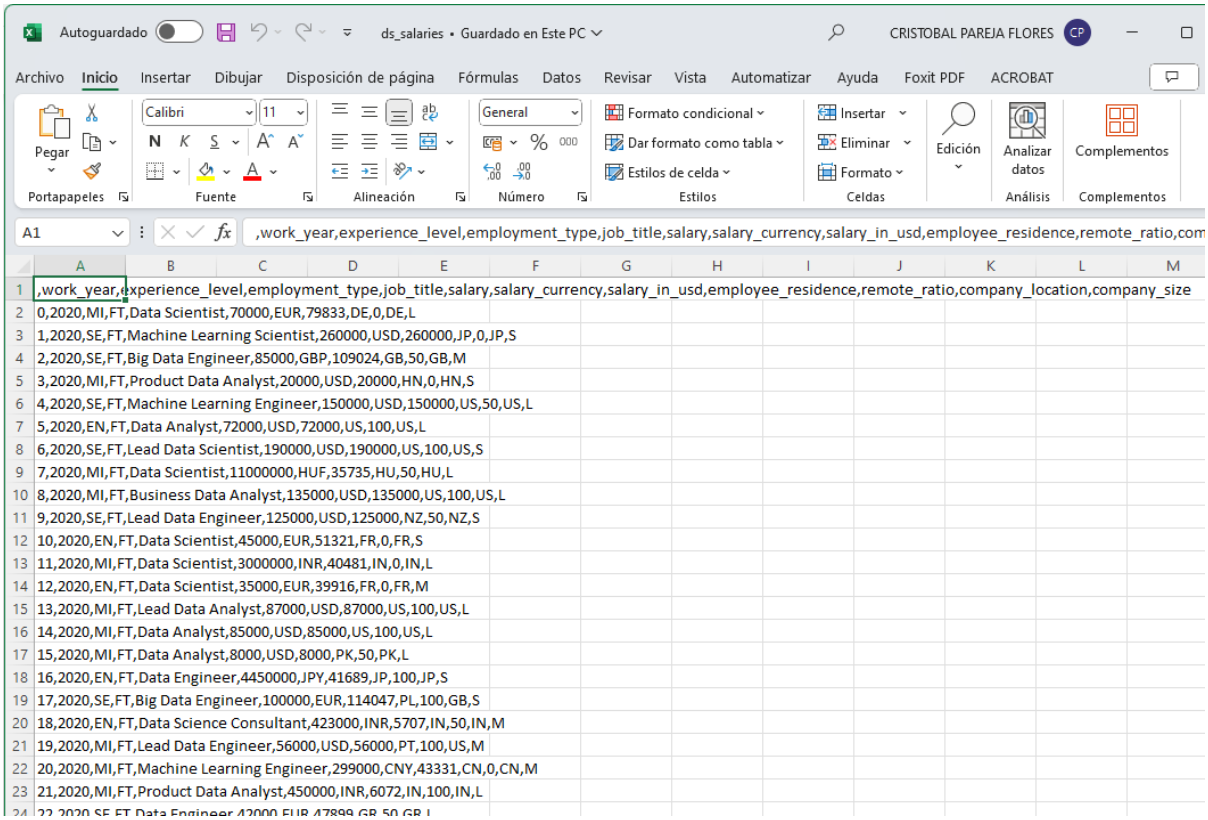
a) Algunas operaciones sencillas [3 puntos]

Nuestra tabla de datos es un archivo de texto (`ds_salaries.csv`) que puede verse así con cualquier editor:



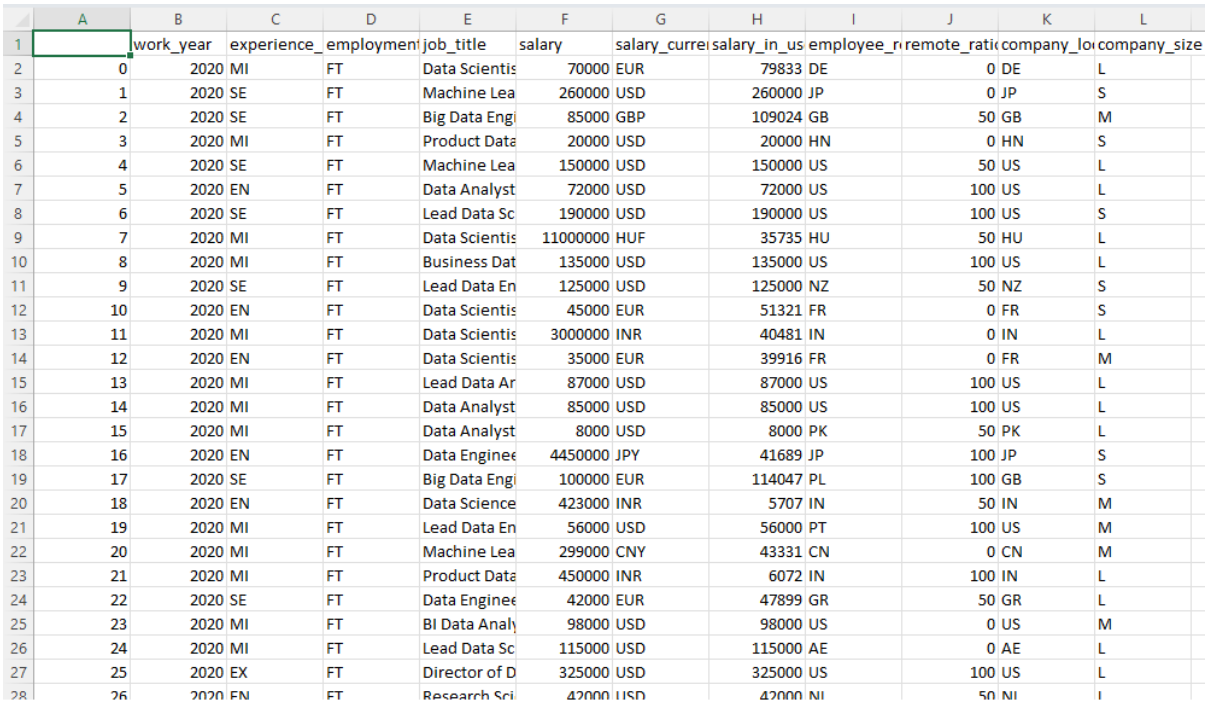
La primera columna es la cabecera, y contiene los nombres de los campos, separados por comas. Las demás, son los valores de dichos campos, consignando los datos de cada vehículo en una línea.

Si la abrimos con *excell*, vemos cada línea en una celda, sin separar los distintos campos:



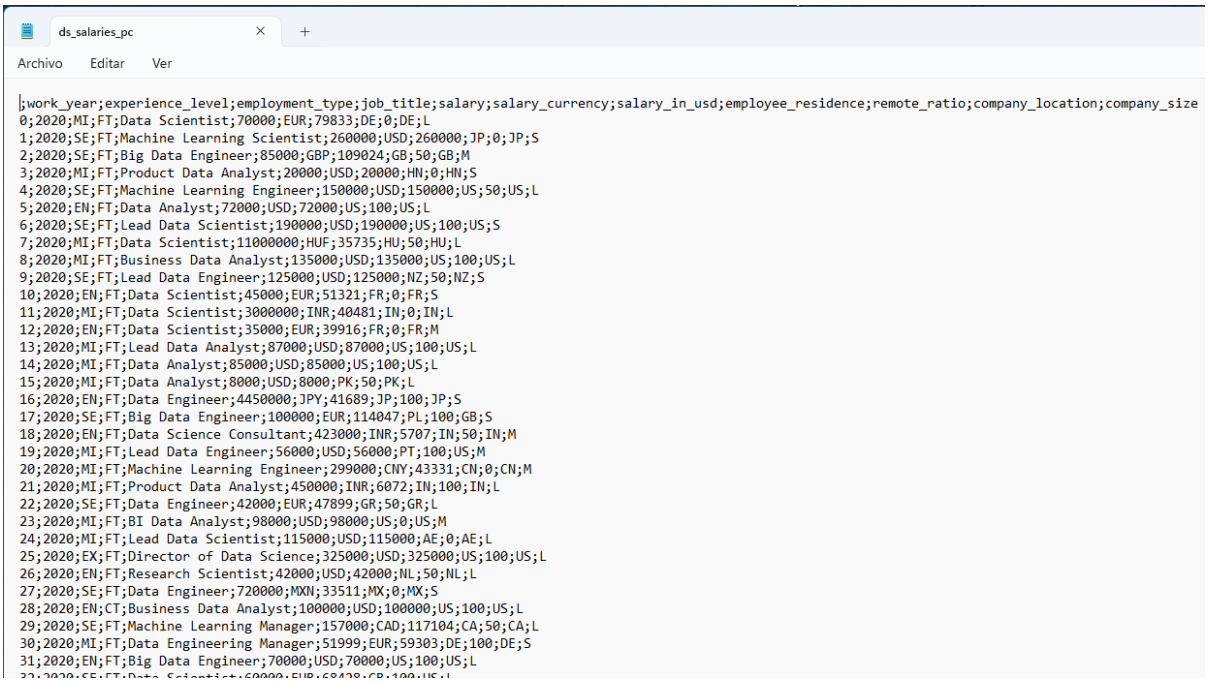
a.1) Cambiar el formato del archivo `csv` a "punto y coma"

Podemos importar la tabla de datos desde excell (pestaña `datos`), simplemente indicando que el separador es una coma:



	A	B	C	D	E	F	G	H	I	J	K	L
1		work_year	experience_	employment	job_title	salary	salary_curre	salary_in_us	employee_r	remote_ratio	company_lo	company_size
2	0	2020	MI	FT	Data Scientis	70000	EUR	79833	DE	0	DE	L
3	1	2020	SE	FT	Machine Lea	260000	USD	260000	JP	0	JP	S
4	2	2020	SE	FT	Big Data Engi	85000	GBP	109024	GB	50	GB	M
5	3	2020	MI	FT	Product Data	20000	USD	20000	HN	0	HN	S
6	4	2020	SE	FT	Machine Lea	150000	USD	150000	US	50	US	L
7	5	2020	EN	FT	Data Analyst	72000	USD	72000	US	100	US	L
8	6	2020	SE	FT	Lead Data Sc	190000	USD	190000	US	100	US	S
9	7	2020	MI	FT	Data Scientis	11000000	HUF	35735	HU	50	HU	L
10	8	2020	MI	FT	Business Dat	135000	USD	135000	US	100	US	L
11	9	2020	SE	FT	Lead Data En	125000	USD	125000	NZ	50	NZ	S
12	10	2020	EN	FT	Data Scientis	45000	EUR	51321	FR	0	FR	S
13	11	2020	MI	FT	Data Scientis	3000000	INR	40481	IN	0	IN	L
14	12	2020	EN	FT	Data Scientis	35000	EUR	39916	FR	0	FR	M
15	13	2020	MI	FT	Lead Data Ar	87000	USD	87000	US	100	US	L
16	14	2020	MI	FT	Data Analyst	85000	USD	85000	US	100	US	L
17	15	2020	MI	FT	Data Analyst	8000	USD	8000	PK	50	PK	L
18	16	2020	EN	FT	Data Enginee	4450000	JPY	41689	JP	100	JP	S
19	17	2020	SE	FT	Big Data Engi	100000	EUR	114047	PL	100	GB	S
20	18	2020	EN	FT	Data Science	423000	INR	5707	IN	50	IN	M
21	19	2020	MI	FT	Lead Data En	56000	USD	56000	PT	100	US	M
22	20	2020	MI	FT	Machine Lea	299000	CNY	43331	CN	0	CN	M
23	21	2020	MI	FT	Product Data	450000	INR	6072	IN	100	IN	L
24	22	2020	SE	FT	Data Enginee	42000	EUR	47899	GR	50	GR	L
25	23	2020	MI	FT	BI Data Analy	98000	USD	98000	US	0	US	M
26	24	2020	MI	FT	Lead Data Sc	115000	USD	115000	AE	0	AE	L
27	25	2020	EX	FT	Director of D	325000	USD	325000	US	100	US	L
28	26	2020	EN	FT	Research Sci	42000	USD	42000	NL	50	NL	L

Pero te propongo generar un archivo como el anterior, pero que use el punto y coma como separador, en vez de la coma:



ds_salaries_pc

Archivo Editar Ver

```
;work_year;experience_level;employment_type;job_title;salary;salary_currency;salary_in_usd;employee_residence;remote_ratio;company_location;company_size
0;2020;MI;FT;Data Scientist;70000;EUR;79833;DE;0;DE;L
1;2020;SE;FT;Machine Learning Scientist;260000;USD;260000;JP;0;JP;S
2;2020;SE;FT;Big Data Engineer;85000;GBP;109024;GB;50;GB;M
3;2020;MI;FT;Product Data Analyst;20000;USD;20000;HN;0;HN;S
4;2020;SE;FT;Machine Learning Engineer;150000;USD;150000;US;50;US;L
5;2020;EN;FT;Data Analyst;72000;USD;72000;US;100;US;L
6;2020;SE;FT;Lead Data Scientist;190000;USD;190000;US;100;US;S
7;2020;MI;FT;Data Scientist;11000000;HUF;35735;HU;50;HU;L
8;2020;MI;FT;Business Data Analyst;135000;USD;135000;US;100;US;L
9;2020;SE;FT;Lead Data Engineer;125000;USD;125000;NZ;50;NZ;S
10;2020;EN;FT;Data Scientist;45000;EUR;51321;FR;0;FR;S
11;2020;MI;FT;Data Scientist;3000000;INR;40481;IN;0;IN;L
12;2020;EN;FT;Data Scientist;35000;EUR;39916;FR;0;FR;M
13;2020;MI;FT;Lead Data Analyst;87000;USD;87000;US;100;US;L
14;2020;MI;FT;Data Analyst;85000;USD;85000;US;100;US;L
15;2020;MI;FT;Data Analyst;8000;USD;8000;PK;50;PK;L
16;2020;EN;FT;Data Engineer;4450000;JPY;41689;JP;100;JP;S
17;2020;SE;FT;Big Data Engineer;100000;EUR;114047;PL;100;GB;S
18;2020;EN;FT;Data Science Consultant;423000;INR;5707;IN;50;IN;M
19;2020;MI;FT;Lead Data Engineer;56000;USD;56000;PT;100;US;M
20;2020;MI;FT;Machine Learning Engineer;299000;CNY;43331;CN;0;CN;M
21;2020;MI;FT;Product Data Analyst;450000;INR;6072;IN;100;IN;L
22;2020;SE;FT;Data Engineer;42000;EUR;47899;GR;50;GR;L
23;2020;MI;FT;BI Data Analyst;98000;USD;98000;US;0;US;M
24;2020;MI;FT;Lead Data Scientist;115000;USD;115000;AE;0;AE;L
25;2020;EX;FT;Director of Data Science;325000;USD;325000;US;100;US;L
26;2020;EN;FT;Research Scientist;42000;USD;42000;NL;50;NL;L
27;2020;SE;FT;Data Engineer;720000;MXN;33511;MX;0;MX;S
28;2020;EN;CT;Business Data Analyst;100000;USD;100000;US;100;US;L
29;2020;SE;FT;Machine Learning Manager;157000;CAD;117104;CA;50;CA;L
30;2020;MI;FT;Data Engineering Manager;51999;EUR;59303;DE;100;DE;S
31;2020;EN;FT;Big Data Engineer;70000;USD;70000;US;100;US;L
```

Para ello, debes diseñar una función que tome con un archivo como el de partida que usa la coma como separador, y genere otro, con el punto y coma como separador.

In [2]:

```
# Esta celda debe ser completada por el estudiante.
```

In [3]:

```
# Ejecución de la función anterior:

DatosComas = "ds_salaries.csv"
DatosPunComas = "ds_salaries_pc.csv"
to_semicolon(DatosComas, DatosPunComas)
```

In [4]:

```
# Comprobamos que funciona como es debido, viendo las primeras cinco filas de ambos archi

with open(DatosComas, "r") as f:
    for _ in range(5):
        linea = f.readline()
        print(linea)

print(".....................................")

with open(DatosPunComas, "r") as f:
    for _ in range(5):
        linea = f.readline()
        print(linea)
```

```
,work_year,experience_level,employment_type,job_title,salary,salary_curren
cy,salary_in_usd,employee_residence,remote_ratio,company_location,company_
size

0,2020,MI,FT,Data Scientist,70000,EUR,79833,DE,0,DE,L

1,2020,SE,FT,Machine Learning Scientist,260000,USD,260000,JP,0,JP,S

2,2020,SE,FT,Big Data Engineer,85000,GBP,109024,GB,50,GB,M

3,2020,MI,FT,Product Data Analyst,20000,USD,20000,HN,0,HN,S

.....................................
;work_year;experience_level;employment_type;job_title;salary;salary_curren
cy;salary_in_usd;employee_residence;remote_ratio;company_location;company_
size

0;2020;MI;FT;Data Scientist;70000;EUR;79833;DE;0;DE;L

1;2020;SE;FT;Machine Learning Scientist;260000;USD;260000;JP;0;JP;S

2;2020;SE;FT;Big Data Engineer;85000;GBP;109024;GB;50;GB;M

3;2020;MI;FT;Product Data Analyst;20000;USD;20000;HN;0;HN;S
```

Nota. En la comprobación anterior, por cada línea que se imprime con la instrucción `print`, se realizan dos saltos de línea. Eso es porque las líneas anteriores se han cargado con la marca `\n`, como puedes ver a continuación, con la última línea. En las funciones que siguen deberás tener esto en cuenta para suprimir la marca `\n` cuando sea necesario.

In [5]:

```
#Observa la marca "\n" al final de la última línea leída:

linea
```

Out[5]:

```
'3;2020;MI;FT;Product Data Analyst;20000;USD;20000;HN;0;HN;S\n'
```

a.2) Selección de una línea, separando sus campos

Diseña ahora una función que selecciona una línea y nos da una lista con los valores de sus campos. Los ejemplares de funcionamiento te darán la información sobre cómo deseamos que funcione:

In [6]:

```
# Esta celda debe ser completada por el estudiante
```

In [7]:

```
# Comprobación del funcionamiento:

cabecera = select_line(DatosPunComas, 0)
print(cabecera)

linea_1 = select_line(DatosPunComas, 1)
print(linea_1)
```

```
['', 'work_year', 'experience_level', 'employment_type', 'job_title', 'sal
ary', 'salary_currency', 'salary_in_usd', 'employee_residence', 'remote_ra
tio', 'company_location', 'company_size']
['0', '2020', 'MI', 'FT', 'Data Scientist', '70000', 'EUR', '79833', 'DE',
'0', 'DE', 'L']
```

Nota: Observa que se suprime la marca de fin de línea, `\n`.

a.3) Ajustes en nuestro archivo de datos

En el archivo de datos, podemos prescindir de la primera fila, que es la cabecera, y de la primera columna, pues únicamente da un número de orden de las filas, de manera que vamos a suprimir ambas, la primera fila y la primera columna; también, la columna de la experiencia será más manejable si convertimos los código en números (así: "EN" -> 0, "MI" -> 1, "EX" -> 2, "SE" -> 3) y algo parecido haremos con el tamaño de las compañías ("S" -> 1, "EX: 0, "M" -> 2, "L" -> 2). Finalmente, para nuestros fines, preferimos manejar el salario en una moneda común, de manera que descartamos las columnas relativas al sueldo en las monedas de cada país y retenemos únicamente la que refleja el salario en dólares.

Realiza estos cambios y, con ellos, genera el archivo nuevo: `DatosSalariosNormalizados.csv` .

In [8]:

```
# Esta celda debe ser completada por el estudiante
```

In [9]:

```
DatosSalariosNormalizados = "ds_salaries.norm.csv"
normalize_data(DatosComas, DatosSalariosNormalizados)
```

b) extracción de algunos datos globales directamente de los archivos

b.1) Relación de puestos y su frecuencia

Con el archivo de datos normalizado, deseamos extraer algunos datos. Concretamente, para un año dado, deseamos conocer la relación de los cargos que aparecen en el archivo, así como la relación de países en que hay compañías con algún empleado residente en otro país, distinto del de la compañia, inicando cuántos empleados están en esta situación.

In [10]:

```
# Esta celda debe ser completada por el estudiante
```

In [11]:

```
puesto_y_frec = puesto_frec(DatosSalariosNormalizados)
puesto_y_frec
```

Out[11]:

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

b.2) Ídem, usando diccionarios por defecto

In [12]:

```
# Esta celda debe ser completada por el estudiante
```

In [13]:

```
puesto_y_frec = puesto_frec(DatosSalariosNormalizados)
puesto_y_frec
```

Out[13]:

```
defaultdict(int,
            {'Data Scientist': 143,
             'Machine Learning Scientist': 8,
             'Big Data Engineer': 8,
             'Product Data Analyst': 2,
             'Machine Learning Engineer': 41,
             'Data Analyst': 97,
             'Lead Data Scientist': 3,
             'Business Data Analyst': 5,
             'Lead Data Engineer': 6,
             'Lead Data Analyst': 3,
             'Data Engineer': 132,
             'Data Science Consultant': 7,
             'BI Data Analyst': 6,
             'Director of Data Science': 7,
             'Research Scientist': 16,
             'Machine Learning Manager': 1,
             'Data Engineering Manager': 5,
             'Machine Learning Infrastructure Engineer': 3,
             'ML Engineer': 6,
             'AI Scientist': 7,
             'Computer Vision Engineer': 6,
             'Principal Data Scientist': 7,
             'Data Science Manager': 12,
             'Head of Data': 5,
             '3D Computer Vision Researcher': 1,
             'Data Analytics Engineer': 4,
             'Applied Data Scientist': 5,
             'Marketing Data Analyst': 1,
             'Cloud Data Engineer': 2,
             'Financial Data Analyst': 2,
             'Computer Vision Software Engineer': 3,
             'Director of Data Engineering': 2,
             'Data Science Engineer': 3,
             'Principal Data Engineer': 3,
             'Machine Learning Developer': 3,
             'Applied Machine Learning Scientist': 4,
             'Data Analytics Manager': 7,
             'Head of Data Science': 4,
             'Data Specialist': 1,
             'Data Architect': 11,
             'Finance Data Analyst': 1,
             'Principal Data Analyst': 2,
             'Big Data Architect': 1,
             'Staff Data Scientist': 1,
             'Analytics Engineer': 4,
             'ETL Developer': 2,
             'Head of Machine Learning': 1,
             'NLP Engineer': 1,
             'Lead Machine Learning Engineer': 1,
             'Data Analytics Lead': 1})
```

b.3) Países con empleados residentes en el extranjero

In [14]:

```
# Esta celda debe ser completada por el estudiante
```

In [15]:

```
anno_cargos_paises_comps_empls(DatosSalariosNormalizados, 2021)
```

Out[15]:

```
({'3D Computer Vision Researcher',
  'AI Scientist',
  '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 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',
  'Finance Data Analyst',
  'Financial Data Analyst',
  'Head of Data',
  'Head of Data Science',
  'Lead Data Analyst',
  'Lead Data Engineer',
  'Lead Data Scientist',
  'ML Engineer',
  'Machine Learning Developer',
  'Machine Learning Engineer',
  'Machine Learning Infrastructure Engineer',
  'Machine Learning Scientist',
  'Marketing Data Analyst',
  'Principal Data Analyst',
  'Principal Data Engineer',
  'Principal Data Scientist',
  'Research Scientist',
  'Staff Data Scientist'},
 {('IN', 'US'): 3,
  ('GB', 'CA'): 1,
  ('IT', 'PL'): 1,
  ('BG', 'US'): 1,
  ('GR', 'DK'): 1,
  ('BR', 'US'): 2,
  ('DE', 'US'): 1,
  ('HU', 'US'): 1,
  ('PK', 'US'): 1,
  ('ES', 'RO'): 1,
  ('VN', 'US'): 1,
  ('SG', 'IL'): 1,
  ('RO', 'US'): 1,
  ('VN', 'GB'): 1,
  ('FR', 'ES'): 1,
  ('RO', 'GB'): 1,
  ('US', 'FR'): 1,
  ('DE', 'AT'): 1,
  ('FR', 'US'): 1,
```

b.4) Ídem, usando diccionarios por defecto

In [16]:

```
# Esta celda debe ser completada por el estudiante
```

```
  ('IT', 'US'): 1,
  ('HK', 'GB'): 1,
  ('IN', 'CH'): 1,
  ('US', 'CA'): 1,
  ('IN', 'AS'): 1,
  ('RS', 'DE'): 1,
  ('PR', 'US'): 1,
  ('NL', 'DE'): 1,
  ('JE', 'CN'): 1})
```

In [17]:

```
anno_cargos_paises_comps_empls(DatosSalariosNormalizados, 2021)
```

Out[17]:

```
({'3D Computer Vision Researcher',
  'AI Scientist',
  '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 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',
  'Finance Data Analyst',
  'Financial Data Analyst',
  'Head of Data',
  'Head of Data Science',
  'Lead Data Analyst',
  'Lead Data Engineer',
  'Lead Data Scientist',
  'ML Engineer',
  'Machine Learning Developer',
  'Machine Learning Engineer',
  'Machine Learning Infrastructure Engineer',
  'Machine Learning Scientist',
  'Marketing Data Analyst',
  'Principal Data Analyst',
  'Principal Data Engineer',
  'Principal Data Scientist',
  'Research Scientist',
  'Staff Data Scientist'},
 defaultdict(int,
             {('IN', 'US'): 3,
              ('GB', 'CA'): 1,
              ('IT', 'PL'): 1,
              ('BG', 'US'): 1,
              ('GR', 'DK'): 1,
              ('BR', 'US'): 2,
              ('DE', 'US'): 1,
              ('HU', 'US'): 1,
              ('PK', 'US'): 1,
              ('ES', 'RO'): 1,
              ('VN', 'US'): 1,
              ('SG', 'IL'): 1,
              ('RO', 'US'): 1,
              ('VN', 'GB'): 1,
              ('FR', 'ES'): 1,
              ('RO', 'GB'): 1,
              ('US', 'FR'): 1,
              ('DE', 'AT'): 1,
```

c) Un diccionario se parece a una tabla... [2 puntos]

c.1) Carga de todos los datos en un diccionario

Para cada tipo de puesto, año y país, deseamos tener la relación de salarios. Cargaremos esta información en un diccionario cuyas claves serán los puestos y cuyo valor, un nuevo diccionario con el año como clave y cuyo valor será un diccionario con el país como clave y la relación de salarios como valor. Aunque esto parece algo lioso, la idea es que podamos luego acceder a la información de la siguiente manera:

```
salarios["Data Scientist"][2021]["US"]
```

```
[73000, 100000, 80000, 82500, 150000, 147000, 160000, 135000, 165000, 115000,
 90000, 130000, 100000, 58000, 109000]
```

In [18]:

```
# Esta celda debe ser completada por el estudiante
```

In [19]:

```
# Comprobación:

Salarios_tabla = load_salaries(DatosSalariosNormalizados)
print(Salarios_tabla)
```

```
defaultdict(<class 'list'>, {('Data Scientist', 1, 2020, 'DE'): [79833],
('Machine Learning Scientist', 3, 2020, 'JP'): [260000], ('Big Data Engin
eer', 3, 2020, 'GB'): [109024, 114047], ('Product Data Analyst', 1, 2020,
'HN'): [20000], ('Machine Learning Engineer', 3, 2020, 'US'): [150000],
('Data Analyst', 0, 2020, 'US'): [72000, 91000], ('Lead Data Scientist',
3, 2020, 'US'): [190000], ('Data Scientist', 1, 2020, 'HU'): [35735], ('B
usiness Data Analyst', 1, 2020, 'US'): [135000], ('Lead Data Engineer',
3, 2020, 'NZ'): [125000], ('Data Scientist', 0, 2020, 'FR'): [51321, 3991
6], ('Data Scientist', 1, 2020, 'IN'): [40481], ('Lead Data Analyst', 1,
2020, 'US'): [87000], ('Data Analyst', 1, 2020, 'US'): [85000], ('Data An
alyst', 1, 2020, 'PK'): [8000], ('Data Engineer', 0, 2020, 'JP'): [4168
9], ('Data Science Consultant', 0, 2020, 'IN'): [5707], ('Lead Data Engin
eer', 1, 2020, 'US'): [56000], ('Machine Learning Engineer', 1, 2020, 'C
N'): [43331], ('Product Data Analyst', 1, 2020, 'IN'): [6072], ('Data Eng
ineer', 3, 2020, 'GR'): [47899], ('BI Data Analyst', 1, 2020, 'US'): [980
00], ('Lead Data Scientist', 1, 2020, 'AE'): [115000], ('Director of Data
Science', 2, 2020, 'US'): [325000], ('Research Scientist', 0, 2020, 'N
L'): [42000], ('Data Engineer', 3, 2020, 'MX'): [33511], ('Business Data
Analyst', 0, 2020, 'US'): [100000], ('Machine Learning Manager', 3, 2020,
('FR', 'US'): 1,
('IT', 'US'): 1,
('HK', 'GB'): 1,
('IN', 'CH'): 1,
('US', 'CA'): 1,
('IN', 'AS'): 1,
('RS', 'DE'): 1,
('PR', 'US'): 1,
('NL', 'DE'): 1,
('IE', 'GB'): 1})
```

In [20]:

```
# Esta celda debe ser completada por el estudiante
```

In [21]:

```
# Comprobación:

for cargo in ["Data Sci", "Machine", "Data Engi"]:
    print(cargo, sueldo_medio_agrupando(Salarios_tabla, cargo, 3, 2022))
```

```
Data Sci 161890.3
Machine 138693.6
Data Engi 140939.5
```

In [22]:

```
# Esta celda debe ser completada por el estudiante
```

In [23]:

```
# Comprobación de funcionamiento:

Salarios = load_salaries(DatosSalariosNormalizados)

print(Salarios)
```

```
defaultdict(<function load_salaries.<locals>.<lambda> at 0x00000278D1188A
F0>, {'Data Scientist': defaultdict(<function load_salaries.<locals>.<lam
bda>.<locals>.<lambda> at 0x00000278D1188CA0>, {2020: defaultdict(<class
'list'>, {'DE': [79833, 62726, 49268], 'HU': [35735], 'FR': [51321, 3991
6, 42197], 'IN': [40481], 'US': [68428, 45760, 105000, 118000, 120000, 13
8350, 412000, 105000], 'GB': [76958], 'ES': [38776], 'IT': [21669], 'AT':
[91237], 'LU': [62726]}), 2021: defaultdict(<class 'list'>, {'FR': [5319
2, 49646, 36643, 77684], 'IN': [29751, 9466, 33808, 28399, 16904], 'US':
[73000, 100000, 80000, 82500, 150000, 5679, 147000, 160000, 135000, 16500
0, 115000, 90000, 130000, 100000, 58000, 109000], 'NG': [50000], 'CA': [7
5774, 87738, 103691], 'UA': [13400], 'IL': [119059], 'MX': [2859], 'CL':
[40038], 'DE': [90734, 90734, 88654, 25532], 'AT': [61467], 'ES': [37825,
46809], 'BR': [12901], 'GB': [116914, 56256], 'VN': [4000], 'TR': [2017
1]}), 2022: defaultdict(<class 'list'>, {'US': [130000, 90000, 136620, 99
360, 146000, 123000, 165220, 120160, 180000, 120000, 95550, 167000, 12300
0, 150000, 211500, 138600, 170000, 123000, 215300, 158200, 180000, 26000
0, 180000, 80000, 140400, 215300, 104890, 140000, 220000, 140000, 185100,
200000, 120000, 230000, 100000, 100000, 165000, 48000, 135000, 78000, 141
300. 102100. 205300. 140400. 176000. 144000. 205300. 140400. 140000. 2100
```

c.2) Un print legible

En la comprobación anterior, puedes observar que yo he utilizado un diccionario por defecto dentro de otro diccionario por defecto. Pero la mezcla de información impide verla con claridad. Seguramente puedes tú mostrarla de manera más legible con unas pocas instrucciones:

In [24]:

```
# Esta celda debe ser completada por el estudiante
```

```
Data Scientist 2020 DE -> [79833, 62726, 49268]
Data Scientist 2020 HU -> [35735]
Data Scientist 2020 FR -> [51321, 39916, 42197]
Data Scientist 2020 IN -> [40481]
Data Scientist 2020 US -> [68428, 45760, 105000, 118000, 120000, 138350,
412000, 105000]
Data Scientist 2020 GB -> [76958]
Data Scientist 2020 ES -> [38776]
Data Scientist 2020 IT -> [21669]
Data Scientist 2020 AT -> [91237]
Data Scientist 2020 LU -> [62726]
Data Scientist 2021 FR -> [53192, 49646, 36643, 77684]
Data Scientist 2021 IN -> [29751, 9466, 33808, 28399, 16904]
Data Scientist 2021 US -> [73000, 100000, 80000, 82500, 150000, 5679, 147
000, 160000, 135000, 165000, 115000, 90000, 130000, 100000, 58000, 10900
0]
Data Scientist 2021 NG -> [50000]
Data Scientist 2021 CA -> [75774, 87738, 103691]
Data Scientist 2021 UA -> [13400]
```

d) Un cálculo y un gráfico con la tabla anterior

Cálculo del sueldo medio de un puesto de trabajo en un año y país dados. Para facilitar la lectura, redondeamos a dos decimales las medias. En esta función, debes tener cuidado con las situaciones posibles en que no existen salarios, pues la media se calcularçia erróneamente.

In [25]:

```
# Esta celda debe ser completada por el estudiante
```

In [26]:

```
# Comprobación de funcionamiento:

for anno in range(2020, 2024):
    print(anno, average_salary_with_dict(Salarios, "Data Scientist", anno, "US"))
```

```
2020 139067.25
2021 106261.19
2022 153483.33
2023 0
```

Nota. Observa que, si la tabla no contiene la información para un año (ej, 2023), la función da un cero, y no un error.

e) Algunas gráficas [1 punto]

e.1 Un modelo típico de gráfica

Vamos a diseñar un modelo de gráfica sencillo que nos sirva para las siguientes representaciones. Tomará como parámetro una lista de pares (x, y), y opcionalmente los tres rótulos explicativos que necesitamos incluir. Además, queremos que las etiquetas de las abcisas aparezcan inclinadas, para poder luego mostrar intervalos de edad.

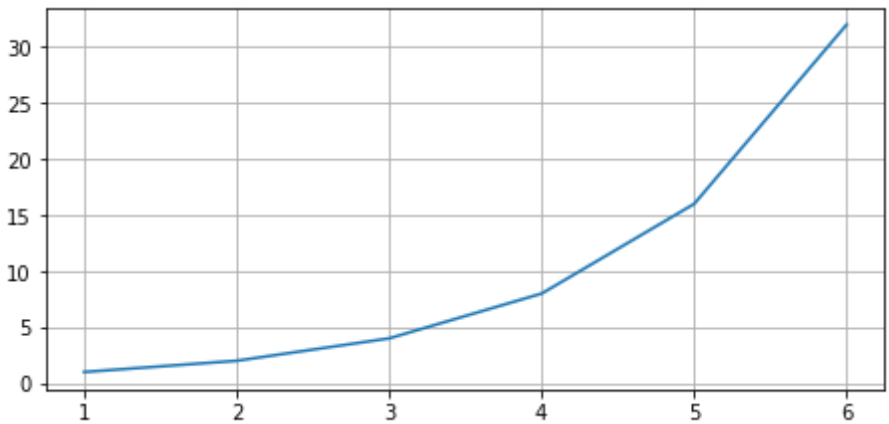
Las pruebas de funcionamiento te darán más información que las explicaciones que pueda yo dar aquí.

In [27]:

```
# Esta celda debe ser completada por el estudiante
```

In [28]:

```
# Pruebas de funcionamiento:

representar_xxx_yyy([(1, 8), (2, 4), (3, 2), (4, 1), (5, 0.5), (6, 0.25)], ["Serie descen

representar_xxx_yyy([(1, 1), (2, 2), (3, 4), (4, 8), (5, 16), (6, 32)])
```

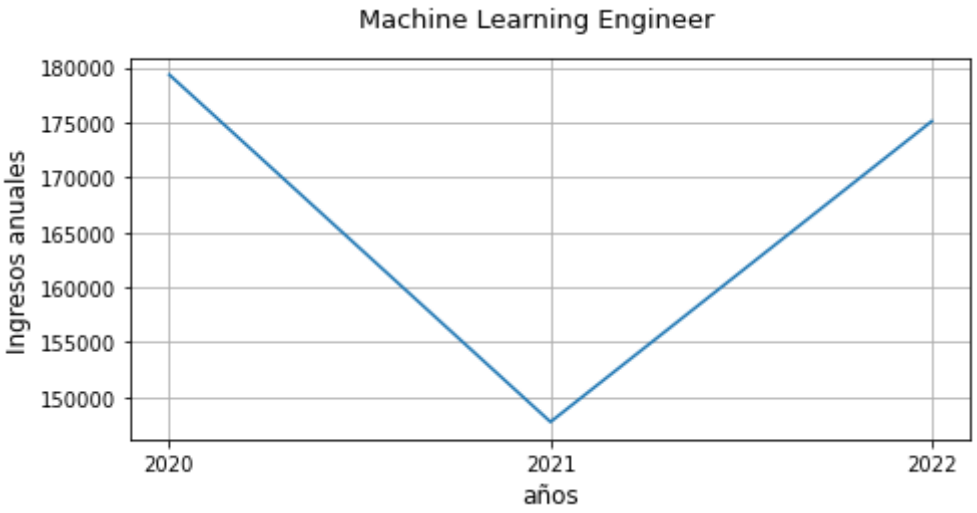


Lógicamente, hemos diseñado nuestro modelo para aplicarlo posteriormente a los datos que ya tenemos. Concretamente, a la representación de las autonomìas medias de los auotmóviles registrados en cada año.

In [29]:

```
# Pruebas de funcionamiento:

annos = range(2020, 2023)
annos_sueldos = [(anno, average_salary_with_dict(Salarios, "Machine Learning Engineer", a

print(annos_sueldos)

representar_xxx_yyy(annos_sueldos, ["Machine Learning Engineer", "Ingresos anuales", "año
```

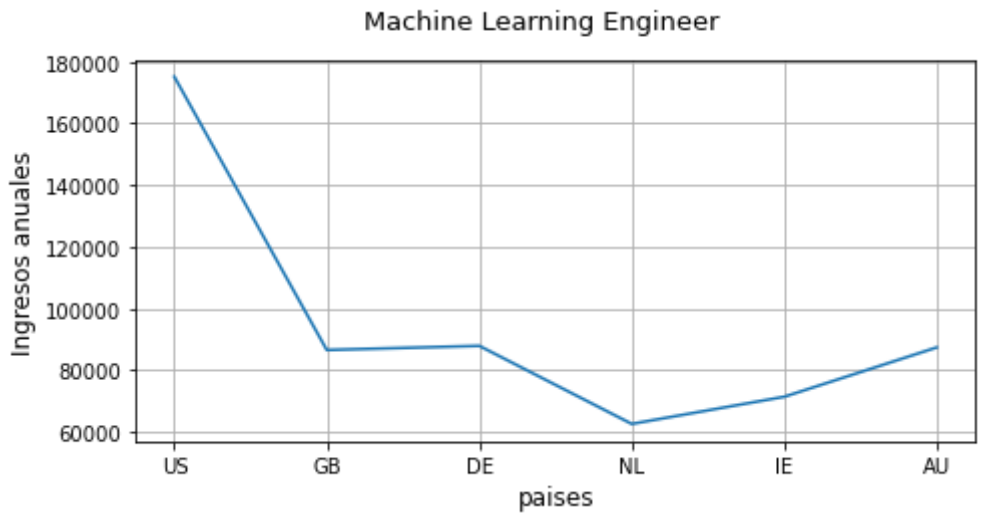
```
[(2020, 179333.33), (2021, 147750.0), (2022, 175099.11)]
```



In [30]:

```
# Pruebas de funcionamiento:

paises = ["US", "GB", "DE", "NL", "IE", "AU"]

paises_sueldos = [(pais, average_salary_with_dict(Salarios, "Machine Learning Engineer",

print(annos_sueldos)

representar_xxx_yyy(paises_sueldos, ["Machine Learning Engineer", "Ingresos anuales", "pa
```

```
[(2020, 179333.33), (2021, 147750.0), (2022, 175099.11)]
```



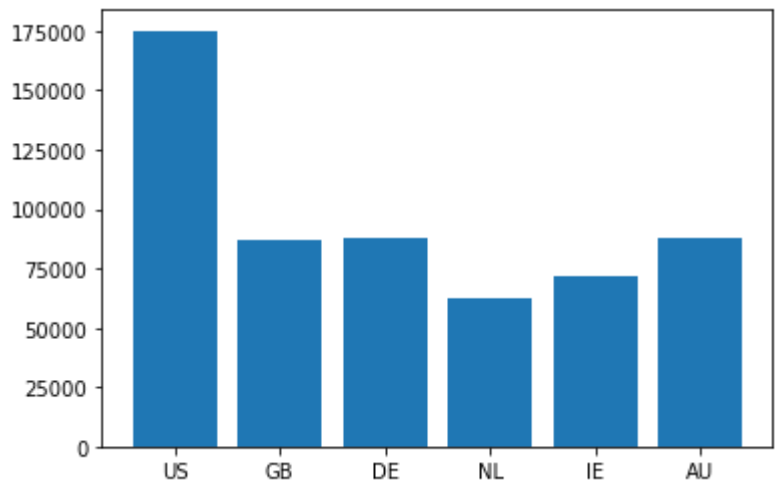
e.2 Histograma

Un gráfico más adecuado para este cometido es el histograma.

Las pruebas de funcionamiento te darán más información que las explicaciones que pueda yo dar aquí.

In [31]:

```
# Esta celda debe ser completada por el estudiante
```



Nota. Vemos que la curva se comporta de un modo extraño, pues sufre una caída en 2001: esto es lo que indican efectivamente los datos.

d) Operaciones con dataframes [3 puntos]

En este apartado, vamos a trabajar con tablas de la librería `pandas`, llamadas `dataframes`.

d.1) Carga del dataframe

La primera operación que necesitamos es cargar el archivo de datos en una tabla, como se ve en el siguiente ejemplo.

In [32]:

```
# Esta celda debe ser completada por el estudiante
```

In [33]:

```
# Comprobación

tabla_completa = load_dataframe(DatosPunComas)

tabla_completa
```

Out[33]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in
0	0	2020	MI	FT	Data Scientist	70000	EUR	7
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	26
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	10
3	3	2020	MI	FT	Product Data Analyst	20000	USD	2
4	4	2020	SE	FT	Machine Learning	150000	USD	15

d.2) Ajustes en nuestro archivo de datos

Deseamos ahora prescidir de la primera columna, pues únicamente da un número de orden de las filas, así como de las columnas relativas a la moneda local (`salary` y `salary currency`).

In [34]:

```
# Esta celda debe ser completada por el estudiante
```

In [35]:

```
# Comprobación

tabla_abreviada
```

Out[35]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_reside
0	2020	MI	FT	Data Scientist	79833	
1	2020	SE	FT	Machine Learning Scientist	260000	
2	2020	SE	FT	Big Data Engineer	109024	
3	2020	MI	FT	Product Data Analyst	20000	
4	2020	SE	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	
602	2022	SE	FT	Data Engineer	154000	
603	2022	SE	FT	Data Engineer	126000	
604	2022	SE	FT	Data Analyst	129000	
605	2022	SE	FT	Data Analyst	150000	
606	2022	MI	FT	AI Scientist	200000	

607 rows × 9 columns

Comprobamos también los tipos de datos de las columnas, para asegurarnos de que los datos numéricos se han cargado como tales; de lo contrario, deberíamos cambiar su tipo.

In [36]:

```
# Comprobación

tabla_abreviada.dtypes
```

Out[36]:

```
work_year              int64
experience_level      object
employment_type       object
job_title             object
salary_in_usd          int64
employee_residence    object
remote_ratio           int64
company_location      object
company_size          object
dtype: object
```

Aunque sólo sea a efectos didácticos, la columna de los porcentajes debería ser un real... Cambia esto, sólo para practicar.

In [37]:

```
# Esta celda debe ser completada por el estudiante


```

In [38]:

```
# Comprobación

tabla_abreviada.dtypes
```

Out[38]:

```
work_year              int64
experience_level      object
employment_type       object
job_title             object
salary_in_usd          int64
employee_residence    object
remote_ratio         float64
company_location      object
company_size          object
dtype: object
```

In [39]:

```
# Comprobación

tabla_abreviada
```

Out[39]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_reside
0	2020	MI	FT	Data Scientist	79833	
1	2020	SE	FT	Machine Learning Scientist	260000	
2	2020	SE	FT	Big Data Engineer	109024	
3	2020	MI	FT	Product Data Analyst	20000	
4	2020	SE	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	
602	2022	SE	FT	Data Engineer	154000	
603	2022	SE	FT	Data Engineer	126000	
604	2022	SE	FT	Data Analyst	129000	
605	2022	SE	FT	Data Analyst	150000	
606	2022	MI	FT	AI Scientist	200000	

607 rows × 9 columns

También, la columna de la experiencia será más manejable si convertimos los código en números (así: "EN" -> 0, "MI" -> 1, "EX" -> 2, "SE" -> 3) y algo parecido haremos con el tamaño de las compañías ("S" -> 1, "EX: 0, "M" -> 2, "L" -> 2).

In [40]:

```
# Esta celda debe ser completada por el estudiante
```

In [41]:

```
# Comprobación

tabla_abreviada
```

Out[41]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_reside
0	2020	1	FT	Data Scientist	79833	
1	2020	3	FT	Machine Learning Scientist	260000	
2	2020	3	FT	Big Data Engineer	109024	
3	2020	1	FT	Product Data Analyst	20000	
4	2020	3	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	
602	2022	3	FT	Data Engineer	154000	
603	2022	3	FT	Data Engineer	126000	
604	2022	3	FT	Data Analyst	129000	
605	2022	3	FT	Data Analyst	150000	
606	2022	1	FT	AI Scientist	200000	

607 rows × 9 columns

In [42]:

```
# Esta celda debe ser completada por el estudiante
```

In [43]:

```
print(average_salary_with_dataframe(tabla_abreviada, "Data Scientist", 2020, "US"))
```

```
139067.25
```

Comprobamos que el resultado es el mismo que el que definimos usando el diccionario:

In [44]:

```
print(average_salary_with_dict(Salarios, "Data Scientist", 2020, "US"))
```

```
139067.25
```

e) Un cálculo masivo con map-reduce [0.5 puntos]

En este apartado se ha de realizar un programa aparte que calcule, para cada país, el número de cada puesto de trabajo que tiene contratado, junto con el máximo sueldo de cada categoría pra dicho país con independencia de laño.

```
C:\...> python puestos_trabajo.py -q ds_salaries.norm.csv
```

El programa funcionará necesariamente con la técnica map-reduce, que podemos poner en juego con la librería `mrjob` .

El funcionamiento del mismo se puede activar también desde aquí:

In [45]:

```
# Hagamos una llamada al programa de consola desde aquí:

! python puestos_trabajo.py -q ds_salaries.norm.csv
```

```
["3D Computer Vision Researcher","IN"]  5409
["AI Scientist","AS"]   18053
["AI Scientist","DK"]   45896
["AI Scientist","ES"]   55000
["AI Scientist","US"]   200000
["Analytics Engineer","US"]     205300
["Applied Data Scientist","CA"] 54238
["Applied Data Scientist","GB"] 110037
["Applied Data Scientist","US"] 380000
["Applied Machine Learning Scientist","CZ"]     31875
["Applied Machine Learning Scientist","US"]     423000
["BI Data Analyst","KE"]        9272
["BI Data Analyst","US"]        150000
["Big Data Architect","CA"]     99703
["Big Data Engineer","CH"]      5882
["Big Data Engineer","GB"]      114047
["Big Data Engineer","IN"]      22611
["Big Data Engineer","MD"]      18000
["Big Data Engineer","RO"]      60000
```

In [46]:

```
# Para que el resultado se almacene en un archivo:

! python puestos_trabajo.py -q ds_salaries.norm.csv  > sueldos_maximos.txt
```

Para que pueda yo ver tu programa cómodamente desde aquí, también se puede mostrar con un comando de la consola, anteponiendo el símbolo `!` . Observaciones:

- La instrucción siguiente está comentada para ocultar una solución mía. Tú debes suprimir el símbolo `#` del comentario para mostrar tu solución aquí.

- Desde mac o linux, se ha de usar el comando `cat` , en vez de `type` .

In [47]:

```
# ! type puestos_trabajo.py
```

f) Un apartado libre [0.5 puntos]

Dejo este apartado a tu voluntad. Inventa tú mismo el enunciado y resuélvelo, mostrando algún aspecto de programación en Python no contemplado o alguna técnica o librería que no has puesto en juego en los apartados anteriores, relacionado con el análisis de datos y con este proyecto. He aquí dos o tres ejemplos posibles:

- Me he quedado un poco insatisfecho con el uso de pandas, que encuentro un poco escaso: este apartado puede poner en juego algunas algunas operaciones que no hemos visto en esta librería.
- El acabado de las figuras es algo rudimentario. en cambio, la librería Plotly me permite permitirte trazar figuras más profesionales, y una posibilidad sencilla es quizá importar los datos del archivo creado por el programa de map-reduce y representarlos gráficamente.
- La disponibilidad de datos de geolocalización puede permitirte alguna representación de la ubicación de los vehísulos registrados en su posición geográfica.

Estos ejemplos pueden servirte como pista, pero que no te limiten. Hay muchas otras posibilidades: geopandas, web scraping, etc.

En la evaluación, si este apartado está bien o muy bien, anota un 0,3 o 0,4. El 0,5 lo reservaremos para las situaciones en que se presente algo brillante, con alguna idea original o alguna técnica novedosa o complejidad especial o algún gráfico vistoso. Especialmente quien opta a un 9,5 o más, debe esmerarse en plantear este apartado a la altura de esa calificación.

g.1) Enunciado

Completa tú este enunciado, y pon el color de la fuente en azul oscuro.

In [48]:

```
# Este apartado debe ser completado por el estudiante
```

In [49]:

```
# Pruebas de funcionamiento, también tarea del estudiante:
```

Datos personales

- **Apellidos:** .....
- **Nombre:** .....
- **Email:** .....
- **Fecha:** .....

Ficha de autoevaluación

Aquí vienen comentarios del estudiante. Lo siguiente es un ejemplo posible obviamente ... elimina este párrafo y redacta el tuyo propio, en azul.

Apartado	Calificación	Comentario
a)	2.0 / 2.0	Completamente resuelto
b)	0.0 / 2.0	No lo he conseguido
c)	0.0 / 2.0	No he entendido el enunciado
d)	0.25 / 1.5	Sólo he conseguido una parte mínima
e)	0.0 / 1.5	No lo he conseguido
f)	0.5 / 0.5	No lo he conseguido más que mínimamente
g)	0.0 / 0.5	No he logrado el correcto funcionamiento
Total	**2.75** / 10.0	**Suspenso**

Ayuda recibida y fuentes utilizadas

... comentarios del estudiante ... Pon tú este párrafo con tus propias observaciones. Elimina este párrafo en verde.

Comentario adicional

... Este apartado es optativo. Si lo completas, ponlo en azul; si no, suprímelo con su título.

In [ ]:

```
# Esta celda se ha de respetar: está aquí para comprobar el funcionamiento de algunas fun
```