In [2]:

```
## import all necessary libraries
import pandas as pd
import numpy as np
import os
from matplotlib import pyplot as plt
import seaborn as sns
```

In [4]:

```
import pandas
```

In [5]:

```
data=pandas.read_csv("insurance.csv")
```

In [11]:

```
data
```

Out[11]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [12]:

```
data.describe()
```

Out[12]:

	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

In [8]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

In [20]:

```
#print top 5 columns
data.head()
```

Out[20]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [15]:

```
data.columns
```

Out[15]:

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtyp
e='object')
```

In [16]:

```
print(data)
```

```
      age     sex     bmi  children smoker     region      charges
0      19  female  27.900         0    yes  southwest  16884.92400
1      18    male  33.770         1     no  southeast   1725.55230
2      28    male  33.000         3     no  southeast   4449.46200
3      33    male  22.705         0     no  northwest  21984.47061
4      32    male  28.880         0     no  northwest   3866.85520
...   ...     ...     ...       ...    ...        ...         ...
1333   50    male  30.970         3     no  northwest  10600.54830
1334   18  female  31.920         0     no  northeast   2205.98080
1335   18  female  36.850         0     no  southeast   1629.83350
1336   21  female  25.800         0     no  southwest   2007.94500
1337   61  female  29.070         0    yes  northwest  29141.36030

[1338 rows x 7 columns]
```

In [21]:

```
#print mean of the age data
print(data["age"].mean())
```

```
39.20702541106129
```

In [22]:

```
# charges column top 5 values
data["charges"].head()
```

Out[22]:

```
0    16884.92400
1     1725.55230
2     4449.46200
3    21984.47061
4     3866.85520
Name: charges, dtype: float64
```

In [23]:

```
# check shape of data set
data.shape
```

Out[23]:

```
(1338, 7)
```

In [ ]:

```
#Sorting data
```

In [24]:

```
#Sort charges values from low to high
data.sort_values("charges")
```

Out[24]:

	age	sex	bmi	children	smoker	region	charges
940	18	male	23.210	0	no	southeast	1121.87390
808	18	male	30.140	0	no	southeast	1131.50660
1244	18	male	33.330	0	no	southeast	1135.94070
663	18	male	33.660	0	no	southeast	1136.39940
22	18	male	34.100	0	no	southeast	1137.01100
...	...	...	...	...	...	...	...
819	33	female	35.530	0	yes	northwest	55135.40209
577	31	female	38.095	1	yes	northeast	58571.07448
1230	52	male	34.485	3	yes	northwest	60021.39897
1300	45	male	30.360	0	yes	southeast	62592.87309
543	54	female	47.410	0	yes	southeast	63770.42801

1338 rows × 7 columns

In [25]:

```
#sort charges values high to low
data.sort_values("charges" , ascending = False)
```

Out[25]:

	age	sex	bmi	children	smoker	region	charges
543	54	female	47.410	0	yes	southeast	63770.42801
1300	45	male	30.360	0	yes	southeast	62592.87309
1230	52	male	34.485	3	yes	northwest	60021.39897
577	31	female	38.095	1	yes	northeast	58571.07448
819	33	female	35.530	0	yes	northwest	55135.40209
...	...	...	...	...	...	...	...
22	18	male	34.100	0	no	southeast	1137.01100
663	18	male	33.660	0	no	southeast	1136.39940
1244	18	male	33.330	0	no	southeast	1135.94070
808	18	male	30.140	0	no	southeast	1131.50660
940	18	male	23.210	0	no	southeast	1121.87390

1338 rows × 7 columns

In [29]:

```
#filter only those values that are less than 20000
data[data["charges"]<=20000]
```

Out[29]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.90	0	yes	southwest	16884.9240
1	18	male	33.77	1	no	southeast	1725.5523
2	28	male	33.00	3	no	southeast	4449.4620
4	32	male	28.88	0	no	northwest	3866.8552
5	31	female	25.74	0	no	southeast	3756.6216
...	...	...	...	...	...	...	...
1332	52	female	44.70	3	no	southwest	11411.6850
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450

1065 rows × 7 columns

In [33]:

```
#filter any value that is greater than 20000 and less than 40000
data[(data["charges"] > 20000) & (data["charges"]<40000)]
```

Out[33]:

	age	sex	bmi	children	smoker	region	charges
3	33	male	22.705	0	no	northwest	21984.47061
9	60	female	25.840	0	no	northwest	28923.13692
11	62	female	26.290	0	yes	southeast	27808.72510
14	27	male	42.130	0	yes	southeast	39611.75770
19	30	male	35.300	0	yes	southwest	36837.46700
...	...	...	...	...	...	...	...
1308	25	female	30.200	0	yes	southwest	33900.65300
1313	19	female	34.700	2	yes	southwest	36397.57600
1321	62	male	26.695	0	yes	northeast	28101.33305
1328	23	female	24.225	2	no	northeast	22395.74424
1337	61	female	29.070	0	yes	northwest	29141.36030

194 rows × 7 columns

In [50]:

```
#show how many times age has appeared within the dataset
data["age"].value_counts()
```

Out[50]:

```
18    69
19    68
20    29
51    29
45    29
46    29
47    29
48    29
50    29
52    29
28    28
54    28
21    28
27    28
26    28
49    28
25    28
24    28
23    28
22    28
53    28
42    27
44    27
43    27
41    27
40    27
31    27
30    27
29    27
56    26
34    26
33    26
32    26
57    26
55    26
35    25
59    25
58    25
36    25
39    25
38    25
37    25
60    23
61    23
62    23
63    23
64    22
Name: age, dtype: int64
```

In [51]:

```
#show first 10
data["age"].value_counts().head()
```

Out[51]:

```
18    69
19    68
20    29
51    29
45    29
Name: age, dtype: int64
```

In [52]:

```
#Indexing , loc and iloc & isin
data.index
```

Out[52]:

```
RangeIndex(start=0, stop=1338, step=1)
```

In [53]:

```
#show multiple columns from dataset
list1 = ["age", "sex" , "bmi", "children", "smoker" , "region", "charges"]
data[list1].head()
```

Out[53]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [55]:

```
#show first 20 rows and 3 columns: age, sex & charges
data.loc[:19, ["age", "sex", "charges"]]
```

Out[55]:

	age	sex	charges
0	19	female	16884.92400
1	18	male	1725.55230
2	28	male	4449.46200
3	33	male	21984.47061
4	32	male	3866.85520
5	31	female	3756.62160
6	46	female	8240.58960
7	37	female	7281.50560
8	37	male	6406.41070
9	60	female	28923.13692
10	25	male	2721.32080
11	62	female	27808.72510
12	23	male	1826.84300
13	56	female	11090.71780
14	27	male	39611.75770
15	19	male	1837.23700
16	52	female	10797.33620
17	23	male	2395.17155
18	56	male	10602.38500
19	30	male	36837.46700

In [56]:

```
#show the 8th row
data.loc[7]
```

Out[56]:

```
age               37
sex           female
bmi            27.74
children           3
smoker            no
region     northwest
charges    7281.5056
Name: 7, dtype: object
```

In [58]:

```
#show 8th row with columns age and bmi
data.loc[7, ["age", "bmi"]]
```

Out[58]:

```
age       37
bmi    27.74
Name: 7, dtype: object
```

In [62]:

```
#use iloc to show rows 1-6 and age and charges column
data.iloc[[0,1,2,3,4,5], [0,6]]
```

Out[62]:

	age	charges
0	19	16884.92400
1	18	1725.55230
2	28	4449.46200
3	33	21984.47061
4	32	3866.85520
5	31	3756.62160

In [63]:

```
#Use index for isin to show regions northwest and southeast
data[data["region"].isin(["northwest", "southeast"])]
```

Out[63]:

	age	sex	bmi	children	smoker	region	charges
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
5	31	female	25.740	0	no	southeast	3756.62160
...	...	...	...	...	...	...	...
1327	51	male	30.030	1	no	southeast	9377.90470
1330	57	female	25.740	2	no	southeast	12629.16560
1333	50	male	30.970	3	no	northwest	10600.54830
1335	18	female	36.850	0	no	southeast	1629.83350
1337	61	female	29.070	0	yes	northwest	29141.36030

689 rows × 7 columns

In [ ]:

```
#Indexing
```

In [64]:

```
# set a column with index: age
data.set_index("age")
```

Out[64]:

age	sex	bmi	children	smoker	region	charges
19	female	27.900	0	yes	southwest	16884.92400
18	male	33.770	1	no	southeast	1725.55230
28	male	33.000	3	no	southeast	4449.46200
33	male	22.705	0	no	northwest	21984.47061
32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...
50	male	30.970	3	no	northwest	10600.54830
18	female	31.920	0	no	northeast	2205.98080
18	female	36.850	0	no	southeast	1629.83350
21	female	25.800	0	no	southwest	2007.94500
61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 6 columns

In [65]:

```
# Undo the above command
data.reset_index()
```

Out[65]:

	index	age	sex	bmi	children	smoker	region	charges
0	0	19	female	27.900	0	yes	southwest	16884.92400
1	1	18	male	33.770	1	no	southeast	1725.55230
2	2	28	male	33.000	3	no	southeast	4449.46200
3	3	33	male	22.705	0	no	northwest	21984.47061
4	4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...	...
1333	1333	50	male	30.970	3	no	northwest	10600.54830
1334	1334	18	female	31.920	0	no	northeast	2205.98080
1335	1335	18	female	36.850	0	no	southeast	1629.83350
1336	1336	21	female	25.800	0	no	southwest	2007.94500
1337	1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 8 columns

In [66]:

```
#  Drop Index
data.reset_index(drop = True)
```

Out[66]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [ ]:

```
#Grouping
```

In [68]:

```
#Show highest bmi
data["bmi"].max()
```

Out[68]:

```
53.13
```

In [69]:

```
data["charges"].mean()
```

Out[69]:

```
13270.422265141257
```

In [70]:

```
data["charges"].cumsum().head()
```

Out[70]:

```
0    16884.92400
1    18610.47630
2    23059.93830
3    45044.40891
4    48911.26411
Name: charges, dtype: float64
```

In [73]:

```
data.groupby("age")["charges"].mean()
data.groupby("age")["charges"].mean().head()
```

Out[73]:

```
age
18     7086.217556
19     9747.909335
20    10159.697736
21     4730.464330
22    10012.932802
Name: charges, dtype: float64
```

In [74]:

```
#Groupby multiple vairables - age, bmi & charges
data.groupby(["age", "bmi"])["charges"].mean()
#show first 10
data.groupby(["age", "bmi"])["charges"].mean()
```

Out[74]:

```
age  bmi
18   15.960     1694.79640
     17.290    12829.45510
     20.790     1607.51010
     21.470     1702.45530
     21.565    13747.87235
                  ...
64   39.050    16085.12750
     39.160    14418.28040
     39.330    14901.51670
     39.700    14319.03100
     40.480    13831.11520
Name: charges, Length: 1295, dtype: float64
```

In [75]:

```
#Replace any missing values & drop all duplicates
data.isnull()
```

Out[75]:

	age	sex	bmi	children	smoker	region	charges
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
1333	False	False	False	False	False	False	False
1334	False	False	False	False	False	False	False
1335	False	False	False	False	False	False	False
1336	False	False	False	False	False	False	False
1337	False	False	False	False	False	False	False

1338 rows × 7 columns

In [76]:

```
#end of code use sum to change from False to count of 0
data.isnull().sum()
```

Out[76]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [77]:

```
# No missing Values but if I had any misisng values I would use code - drop rows
data.dropna()
```

Out[77]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [78]:

```
#replace any missing values with 0
data.fillna(0)
```

Out[78]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [80]:

```
#replace any previous values with 0
data.fillna(method="bfill")
```

Out[80]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [81]:

```
data.dropna(axis=0)
```

Out[81]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

In [107]:

```
data = pd.DataFrame({
    "age" : ["22","22","22","21","25","23","22"],
    "bmi" : ["21.2","21.2", "23.4", "21.2", "26.4", "22.5", "19.5"],
})
print(data)
```

```
  age   bmi
0  22  21.2
1  22  21.2
2  22  23.4
3  21  21.2
4  25  26.4
5  23  22.5
6  22  19.5
```

In [108]:

```
data.drop_duplicates()
```

Out[108]:

	age	bmi
0	22	21.2
2	22	23.4
3	21	21.2
4	25	26.4
5	23	22.5
6	22	19.5

In [ ]:

```
#Looping
```

In [109]:

```
data.index
```

Out[109]:

```
RangeIndex(start=0, stop=7, step=1)
```

In [1]:

```
region_list= ["northwest", "northeast", "southwest", "southeast"]
for x in region_list:
    print(x)
```

```
northwest
northeast
southwest
southeast
```

In [ ]:

```
#Joining Data
```

In [110]:

```
data2 = pd.DataFrame({
    "sex" : ["male", "male", "female", "male", "female", "male", "female",],
    "hair_colour" : ["blonde", "black", "grey", "brown", "red", "pink", "purple"],
})
print(data2)
```

```
      sex hair_colour
0    male      blonde
1    male       black
2  female        grey
3    male       brown
4  female         red
5    male        pink
6  female      purple
```

In [111]:

```
left= data
right= data2
joined_frame = left.join(right)
print(joined_frame)
```

```
  age   bmi     sex hair_colour
0  22  21.2    male      blonde
1  22  21.2    male       black
2  22  23.4  female        grey
3  21  21.2    male       brown
4  25  26.4  female         red
5  23  22.5    male        pink
6  22  19.5  female      purple
```

In [ ]:

```
#Numpy
```

In [131]:

```
np_bmi = np.array( ["21.2","21.2", "23.4", "21.2", "26.4", "22.5", "19.5"])
```

In [6]:

```
np_data = np.array(data)
print(data)
```

```
      age     sex     bmi  children smoker     region      charges
0      19  female  27.900         0    yes  southwest  16884.92400
1      18    male  33.770         1     no  southeast   1725.55230
2      28    male  33.000         3     no  southeast   4449.46200
3      33    male  22.705         0     no  northwest  21984.47061
4      32    male  28.880         0     no  northwest   3866.85520
...   ...     ...     ...       ...    ...        ...         ...
1333   50    male  30.970         3     no  northwest  10600.54830
1334   18  female  31.920         0     no  northeast   2205.98080
1335   18  female  36.850         0     no  southeast   1629.83350
1336   21  female  25.800         0     no  southwest   2007.94500
1337   61  female  29.070         0    yes  northwest  29141.36030

[1338 rows x 7 columns]
```

In [7]:

```
np.std(np_data[:,2])
```

Out[7]:

```
6.095907641589428
```

In [9]:

```
np.mean(np_data[:,2])
```

Out[9]:

```
30.663396860986538
```

In [10]:

```
np_bmi = np.array(data["bmi"])
print(np_bmi)
```

```
[27.9  33.77 33.   ... 36.85 25.8  29.07]
```

In [13]:

```
average_bmi = np_bmi == 40
print(np_bmi.sum())
```

```
41027.625
```

In [ ]:

```
#Dictionaires & Lists
```

In [134]:

```
age_list= data[["age"]]
print(age_list)
```

```
      age
0      19
1      18
2      28
3      33
4      32
...   ...
1333   50
1334   18
1335   18
1336   21
1337   61

[1338 rows x 1 columns]
```

In [135]:

```
#slicing - Positive Indexing
print(age_list[:10])
```

```
   age
0   19
1   18
2   28
3   33
4   32
5   31
6   46
7   37
8   37
9   60
```

In [136]:

```
print(age_list [4:20])
```

```
    age
4    32
5    31
6    46
7    37
8    37
9    60
10   25
11   62
12   23
13   56
14   27
15   19
16   52
17   23
18   56
19   30
```

In [137]:

```
#Slicing - Negative Indexing
print(age_list[-20::1])
```

```
      age
1318   35
1319   39
1320   31
1321   62
1322   62
1323   42
1324   31
1325   61
1326   42
1327   51
1328   23
1329   52
1330   57
1331   23
1332   52
1333   50
1334   18
1335   18
1336   21
1337   61
```

In [138]:

```
#slice 2 lists evenly
print(age_list[0:669])
print(age_list[669:1338])
```

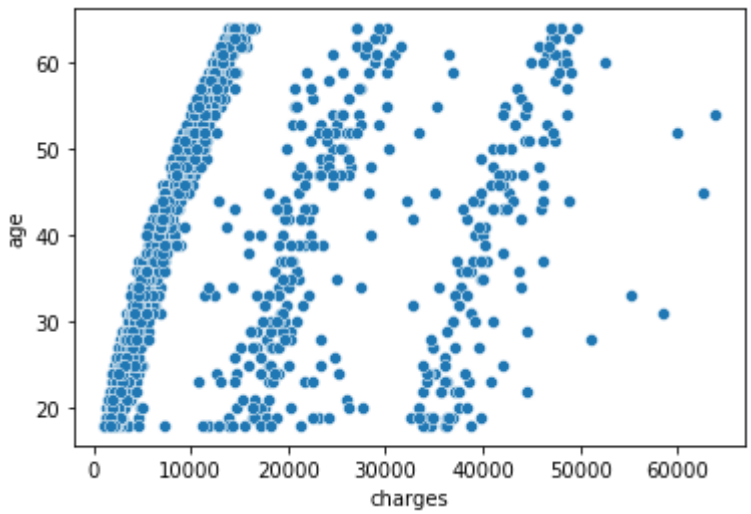
```
     age
0     19
1     18
2     28
3     33
4     32
..   ...
664   64
665   43
666   49
667   40
668   62

[669 rows x 1 columns]
      age
669    40
670    30
671    29
672    36
673    41
...   ...
1333   50
1334   18
1335   18
1336   21
1337   61

[669 rows x 1 columns]
```

In [ ]:

```
#Seaborn Plots
```

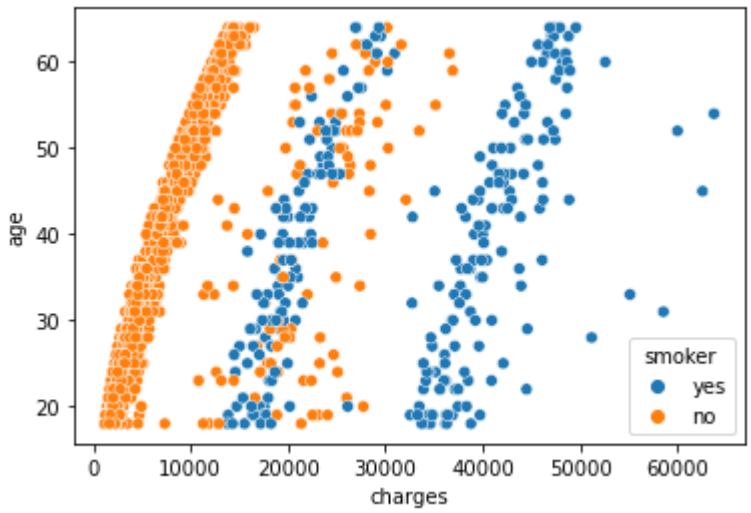
In [139]:

```
sns.scatterplot( x="charges", y = "age", data=data)
plt.show()
```



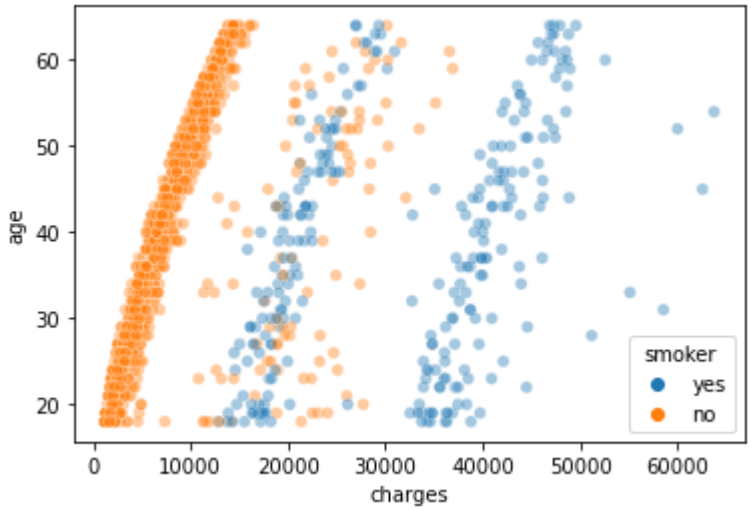
In [140]:

```
sns.scatterplot ( x="charges", y = "age", data=data, hue="smoker")
plt.show()
```



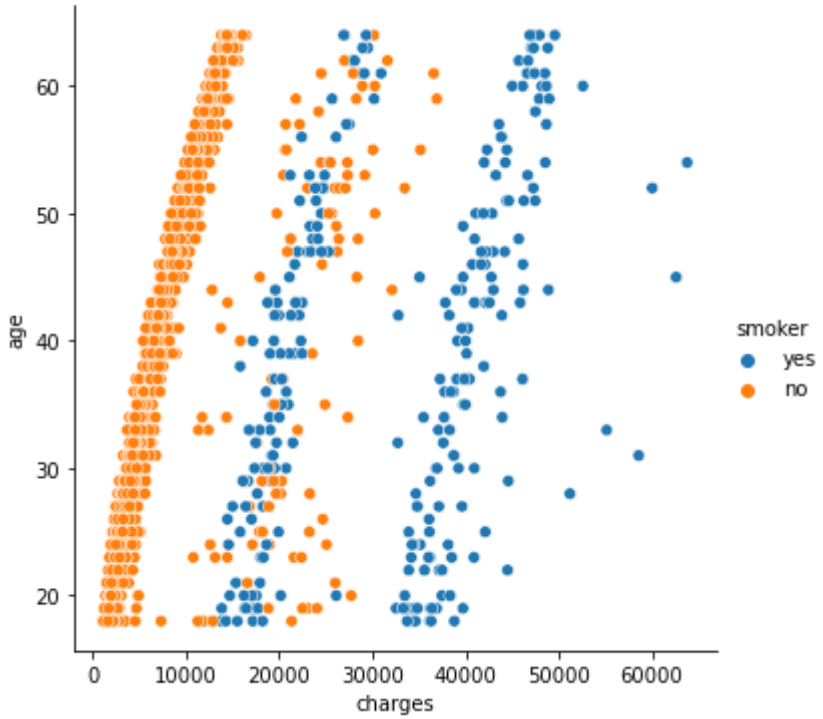
In [141]:

```
sns.scatterplot ( x="charges", y = "age", data=data, hue="smoker", hue_order=["yes", "no"],
plt.show()
```



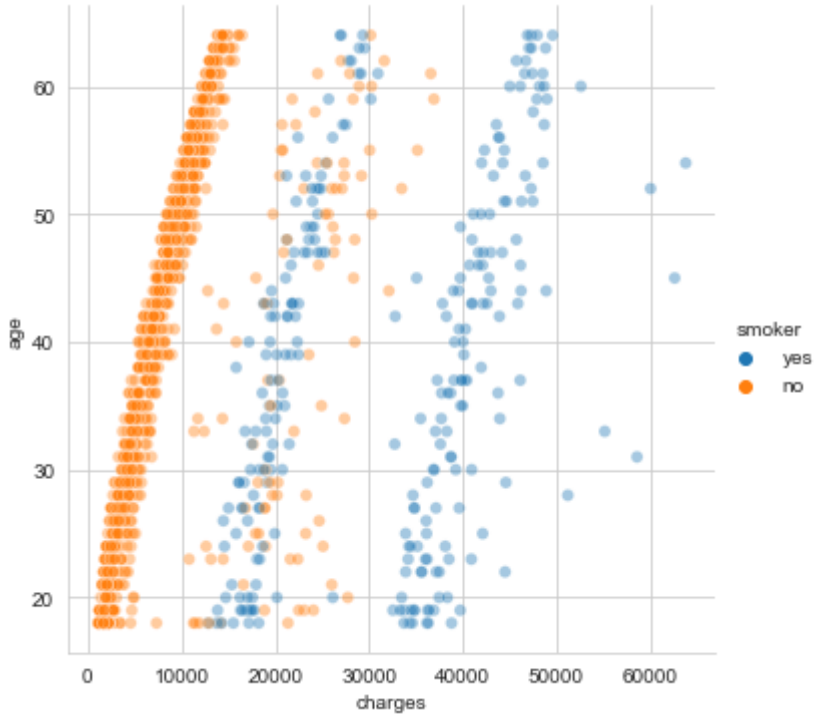
In [142]:

```
sns.relplot(x="charges", y = "age", data=data, kind="scatter", hue="smoker")
plt.show()
```



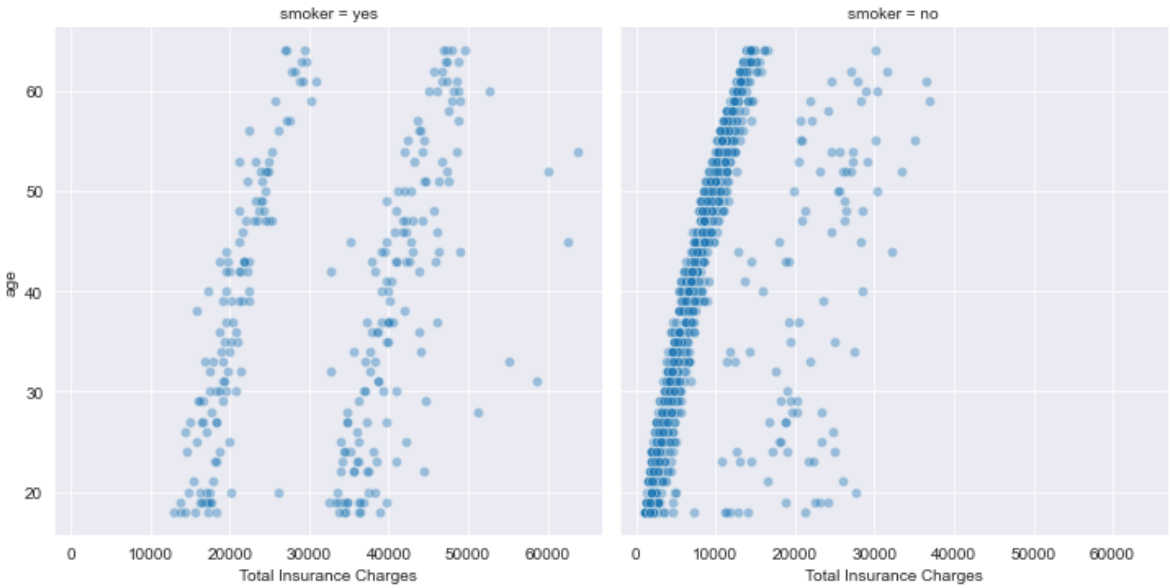
In [145]:

```
sns.relplot(x="charges", y = "age", data=data, kind="scatter", hue="smoker", alpha=0.4,)
sns.set_style("darkgrid")
plt.show()
```



In [147]:

```
g = sns.relplot(x="charges", y= "age", data=data, kind="scatter", col="smoker", alpha=0.4)
g.set(xlabel="Total Insurance Charges", ylabel="age")
sns.color_palette("tab10")
plt.show()
```

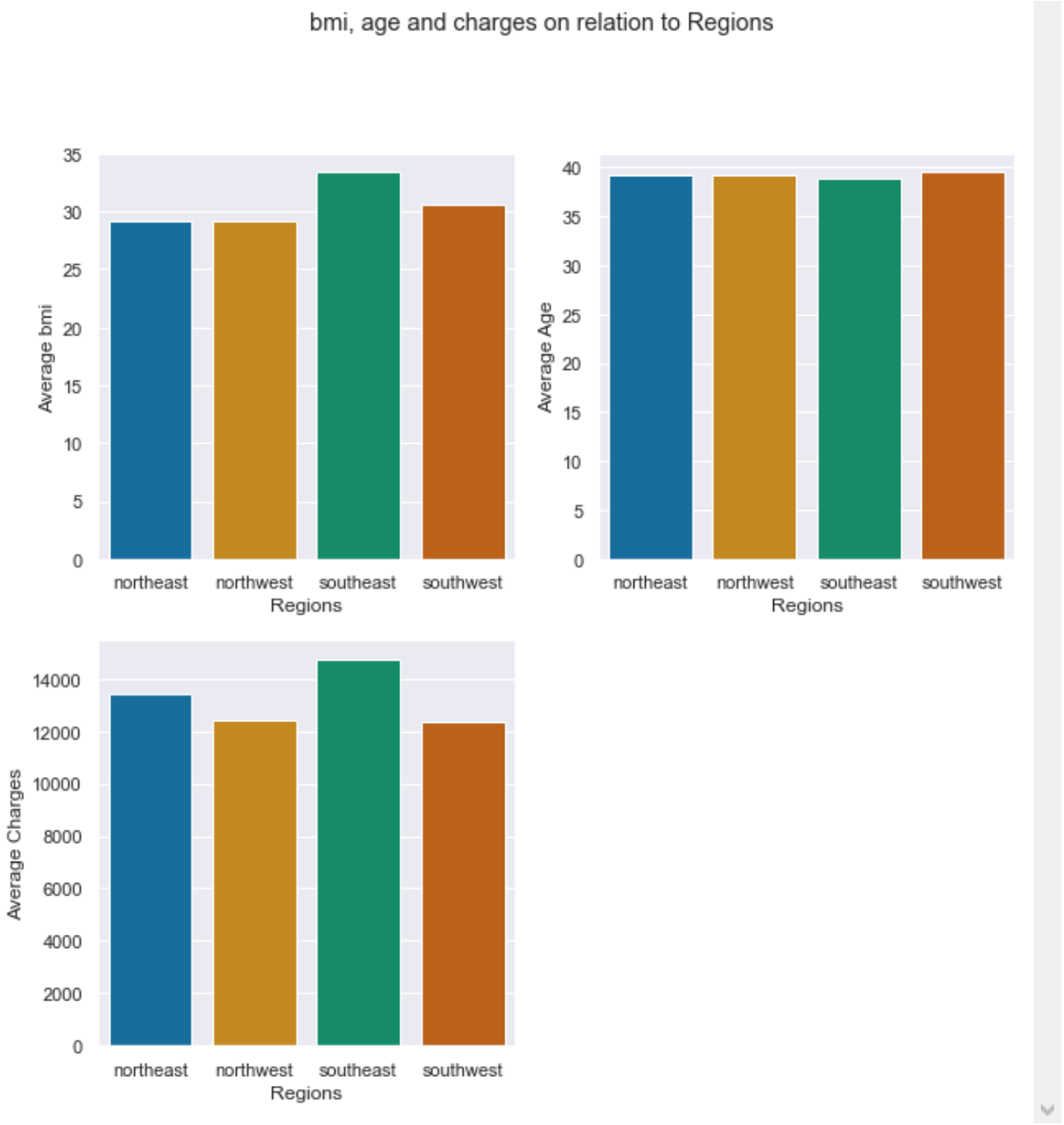


In [18]:

```
fig, ax = plt.subplots(2,2, figsize = (10,10))
sns.set_style("darkgrid")
sns.set_palette("colorblind")
sns.set_context("notebook")
fig1= sns.barplot(x="region", y="bmi",data=data_region, ax=ax[0,0])
fig2= sns.barplot(x="region", y="age",data=data_region, ax=ax[0,1])
fig3= sns.barplot(x="region", y="charges",data=data_region, ax=ax[1,0])
fig.delaxes(ax[1][1])

fig1.set_ylabel("Average bmi")
fig2.set_ylabel("Average Age")
fig3.set_ylabel("Average Charges")

fig1.set_xlabel("Regions")
fig2.set_xlabel("Regions")
fig3.set_xlabel("Regions")

fig.suptitle("bmi, age and charges on relation to Regions", y=1)
```

Out[18]:

```
Text(0.5, 1, 'bmi, age and charges on relation to Regions')
```



In [ ]:

```
#Comparing Data set based on Gender
```

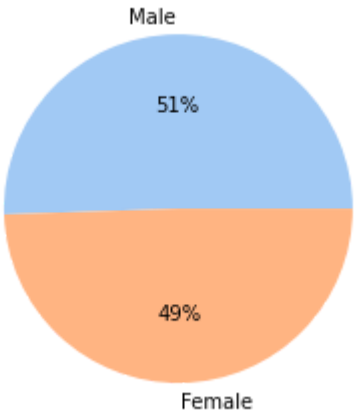
In [8]:

```
print(data["sex"].value_counts())
```

```
male      676
female    662
Name: sex, dtype: int64
```

In [9]:

```
colors = sns.color_palette('pastel')[0:5]
plt.pie(data["sex"].value_counts(), labels = ["Male", "Female"], colors = colors, autopct='
plt.show()
```



In [10]:

```
print(data["smoker"].value_counts())
```

```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [ ]:

```

```

In [ ]:

```
#Custom function
```

In [160]:

```
data["smoker"].count()
```

Out[160]:

```
1338
```

In [161]:

```
print(data["smoker"]=="yes")
```

```
0        True
1       False
2       False
3       False
4       False
        ...
1333    False
1334    False
1335    False
1336    False
1337     True
Name: smoker, Length: 1338, dtype: bool
```

In [162]:

```
smoker_yes = ["yes"]
total_smoker =data[data.smoker.isin(smoker_yes)]
print(total_smoker)
```

```
      age     sex     bmi  children smoker     region      charges
0      19  female  27.900         0    yes  southwest  16884.92400
11     62  female  26.290         0    yes  southeast  27808.72510
14     27    male  42.130         0    yes  southeast  39611.75770
19     30    male  35.300         0    yes  southwest  36837.46700
23     34  female  31.920         1    yes  northeast  37701.87680
...   ...     ...     ...       ...    ...        ...          ...
1313   19  female  34.700         2    yes  southwest  36397.57600
1314   30  female  23.655         3    yes  northwest  18765.87545
1321   62    male  26.695         0    yes  northeast  28101.33305
1323   42  female  40.370         2    yes  southeast  43896.37630
1337   61  female  29.070         0    yes  northwest  29141.36030

[274 rows x 7 columns]
```

In [ ]:

```
#Add 10 years to each age
```

In [163]:

```
def add_10years(x):
    return x+10
data["age"]=data["age"].transform(add_10years)
print(data)
```

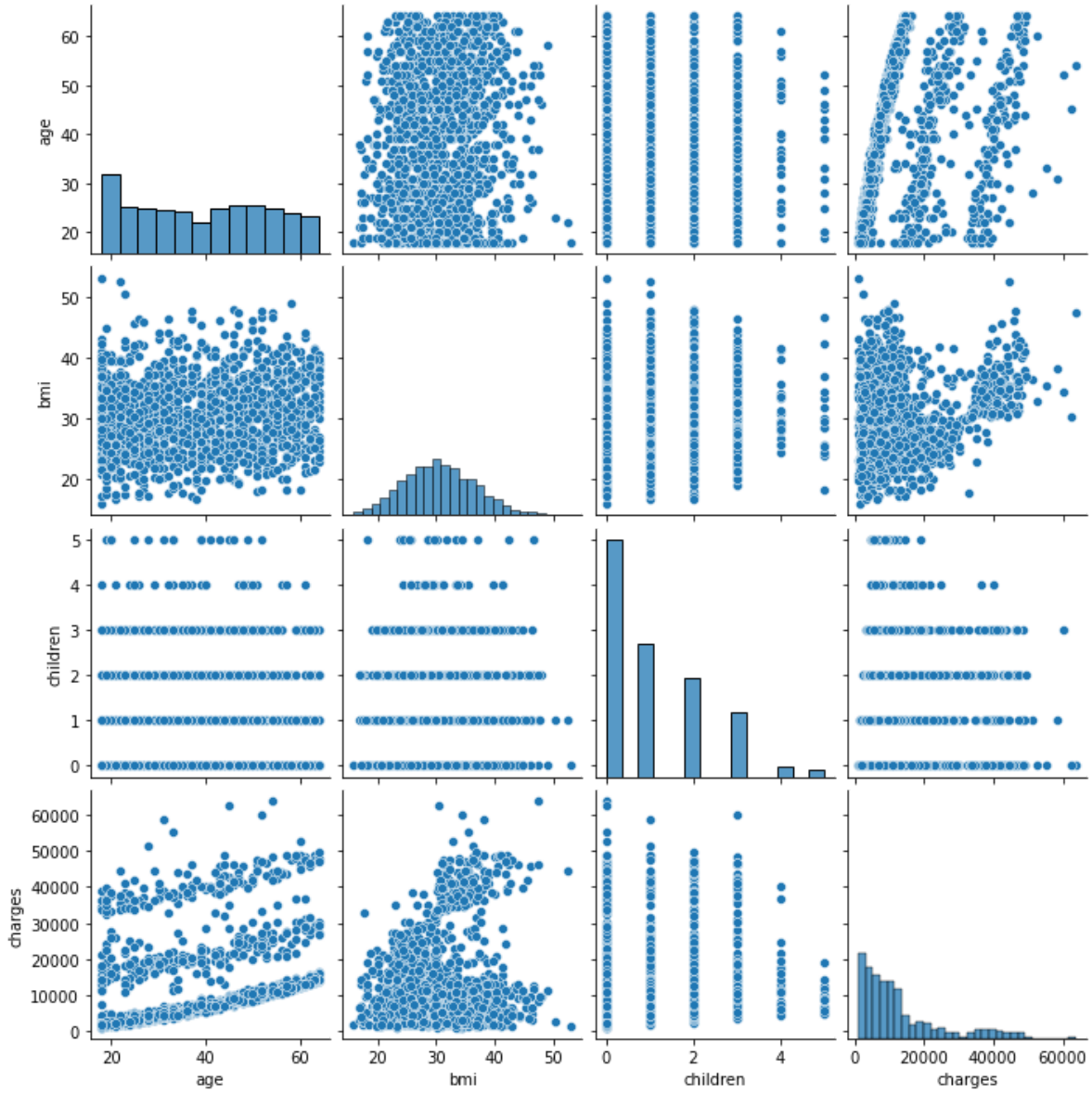
```
      age     sex     bmi  children smoker     region      charges
0      29  female  27.900         0    yes  southwest  16884.92400
1      28    male  33.770         1     no  southeast   1725.55230
2      38    male  33.000         3     no  southeast   4449.46200
3      43    male  22.705         0     no  northwest  21984.47061
4      42    male  28.880         0     no  northwest   3866.85520
...   ...     ...     ...       ...    ...        ...         ...
1333   60    male  30.970         3     no  northwest  10600.54830
1334   28  female  31.920         0     no  northeast   2205.98080
1335   28  female  36.850         0     no  southeast   1629.83350
1336   31  female  25.800         0     no  southwest   2007.94500
1337   71  female  29.070         0    yes  northwest  29141.36030

[1338 rows x 7 columns]
```

In [ ]:

```
#Visualiing the Data
```

In [7]:

```
sns.pairplot(data)
plt.show()
```



In [ ]:

```
# Retrieving Data using online APIs
```

In [34]:

```
import requests
import pandas as pd
import matplotlib.pyplot as plt
import time
plt.style.use('seaborn')
```

In [35]:

```
google = requests.get('http://google.com')
type(google)
```

Out[35]:

```
requests.models.Response
```

In [3]:

```
import requests
```

In [8]:

```
data=requests.get("https://www.weatherapi.com/docs/weather_conditions.json")
```

In [9]:

```
data
```

Out[9]:

```
<Response [200]>
```

In [10]:

```
data.json()
```

```
  'icon': 185},
 {'code': 1087,
  'day': 'Thundery outbreaks possible',
  'night': 'Thundery outbreaks possible',
  'icon': 200},
 {'code': 1114, 'day': 'Blowing snow', 'night': 'Blowing snow', 'icon': 22
7},
 {'code': 1117, 'day': 'Blizzard', 'night': 'Blizzard', 'icon': 230},
 {'code': 1135, 'day': 'Fog', 'night': 'Fog', 'icon': 248},

 {'code': 1147, 'day': 'Freezing fog', 'night': 'Freezing fog', 'icon': 26
0},
 {'code': 1150,
  'day': 'Patchy light drizzle',
  'night': 'Patchy light drizzle',
  'icon': 263},
 {'code': 1153, 'day': 'Light drizzle', 'night': 'Light drizzle', 'icon':
266},
 {'code': 1168,
  'day': 'Freezing drizzle',
```

In [12]:

```
type(data.json())
```

Out[12]:

```
list
```

In [23]:

```
iss = requests.get('https://www.weatherapi.com/docs/weather_conditions.json')
iss.json()
```

```
7},
 {'code': 1117, 'day': 'Blizzard', 'night': 'Blizzard', 'icon': 230},
 {'code': 1135, 'day': 'Fog', 'night': 'Fog', 'icon': 248},
 {'code': 1147, 'day': 'Freezing fog', 'night': 'Freezing fog', 'icon': 26
0},
 {'code': 1150,
  'day': 'Patchy light drizzle',
  'night': 'Patchy light drizzle',
  'icon': 263},
 {'code': 1153, 'day': 'Light drizzle', 'night': 'Light drizzle', 'icon':
266},
 {'code': 1168,
  'day': 'Freezing drizzle',
  'night': 'Freezing drizzle',
  'icon': 281},
 {'code': 1171,
  'day': 'Heavy freezing drizzle',
  'night': 'Heavy freezing drizzle',
  'icon': 284},
 {'code': 1180,
```

In [29]:

```
iss = requests.get('https://www.weatherapi.com/docs/weather_conditions.json')
iss_dict = iss.json()
```

In [30]:

```
data
```

Out[30]:

```
[{'code': 1000, 'day': 'Sunny', 'night': 'Clear', 'icon': 113},
 {'code': 1003, 'day': 'Partly cloudy', 'night': 'Partly cloudy', 'icon':
116},
 {'code': 1006, 'day': 'Cloudy', 'night': 'Cloudy', 'icon': 119},
 {'code': 1009, 'day': 'Overcast', 'night': 'Overcast', 'icon': 122},
 {'code': 1030, 'day': 'Mist', 'night': 'Mist', 'icon': 143},
 {'code': 1063,
  'day': 'Patchy rain possible',
  'night': 'Patchy rain possible',
  'icon': 176},
 {'code': 1066,
  'day': 'Patchy snow possible',
  'night': 'Patchy snow possible',
  'icon': 179},
 {'code': 1069,
  'day': 'Patchy sleet possible',
  'night': 'Patchy sleet possible',
  'icon': 182},
```

In [37]:

```
t = requests.get("https://www.weatherapi.com/docs/weather_conditions.json")
data = pd.DataFrame(t.json())
data.set_index('code').head()
```

Out[37]:

code	day	night	icon
1000	Sunny	Clear	113
1003	Partly cloudy	Partly cloudy	116
1006	Cloudy	Cloudy	119
1009	Overcast	Overcast	122
1030	Mist	Mist	143

In [38]:

```
import pandas as pd
pd.DataFrame(iss.json())
```

Out[38]:

	code	day	night	icon
0	1000	Sunny	Clear	113
1	1003	Partly cloudy	Partly cloudy	116
2	1006	Cloudy	Cloudy	119
3	1009	Overcast	Overcast	122
4	1030	Mist	Mist	143
5	1063	Patchy rain possible	Patchy rain possible	176
6	1066	Patchy snow possible	Patchy snow possible	179
7	1069	Patchy sleet possible	Patchy sleet possible	182
8	1072	Patchy freezing drizzle possible	Patchy freezing drizzle possible	185
9	1087	Thundery outbreaks possible	Thundery outbreaks possible	200
10	1114	Blowing snow	Blowing snow	227
11	1117	Blizzard	Blizzard	230
12	1135	Fog	Fog	248
13	1147	Freezing fog	Freezing fog	260
14	1150	Patchy light drizzle	Patchy light drizzle	263
15	1153	Light drizzle	Light drizzle	266
16	1168	Freezing drizzle	Freezing drizzle	281
17	1171	Heavy freezing drizzle	Heavy freezing drizzle	284
18	1180	Patchy light rain	Patchy light rain	293
19	1183	Light rain	Light rain	296
20	1186	Moderate rain at times	Moderate rain at times	299
21	1189	Moderate rain	Moderate rain	302
22	1192	Heavy rain at times	Heavy rain at times	305
23	1195	Heavy rain	Heavy rain	308
24	1198	Light freezing rain	Light freezing rain	311
25	1201	Moderate or heavy freezing rain	Moderate or heavy freezing rain	314
26	1204	Light sleet	Light sleet	317
27	1207	Moderate or heavy sleet	Moderate or heavy sleet	320
28	1210	Patchy light snow	Patchy light snow	323
29	1213	Light snow	Light snow	326
30	1216	Patchy moderate snow	Patchy moderate snow	329
31	1219	Moderate snow	Moderate snow	332
32	1222	Patchy heavy snow	Patchy heavy snow	335
33	1225	Heavy snow	Heavy snow	338

	code	day	night	icon
34	1237	Ice pellets	Ice pellets	350
35	1240	Light rain shower	Light rain shower	353
36	1243	Moderate or heavy rain shower	Moderate or heavy rain shower	356
37	1246	Torrential rain shower	Torrential rain shower	359
38	1249	Light sleet showers	Light sleet showers	362
39	1252	Moderate or heavy sleet showers	Moderate or heavy sleet showers	365
40	1255	Light snow showers	Light snow showers	368
41	1258	Moderate or heavy snow showers	Moderate or heavy snow showers	371
42	1261	Light showers of ice pellets	Light showers of ice pellets	374
43	1264	Moderate or heavy showers of ice pellets	Moderate or heavy showers of ice pellets	377
44	1273	Patchy light rain with thunder	Patchy light rain with thunder	386
45	1276	Moderate or heavy rain with thunder	Moderate or heavy rain with thunder	389
46	1279	Patchy light snow with thunder	Patchy light snow with thunder	392
47	1282	Moderate or heavy snow with thunder	Moderate or heavy snow with thunder	395

In [39]:

```
from datetime import date, datetime

datetime.now().strftime("%Y-%m-%d %H:%M:%S")
```

Out[39]:

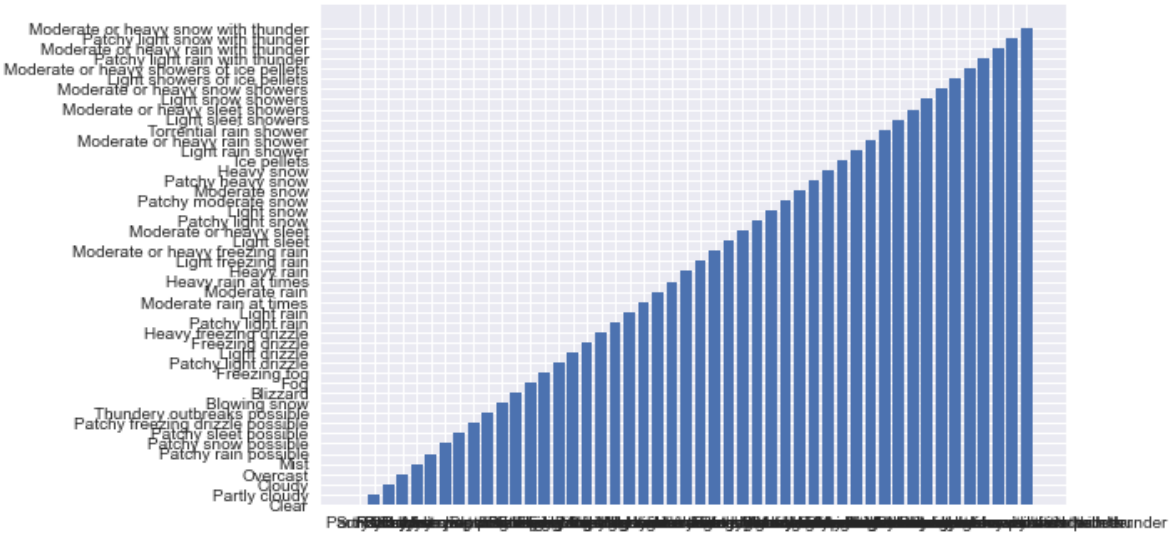
```
'2022-05-19 15:04:30'
```

In [40]:

```
plt.bar(data['day'], data['night'])
```

Out[40]:

```
<BarContainer object of 48 artists>
```



In [ ]:

```
```