Import libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

df = pd.read_csv("/content/insurance.csv")

df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 1338,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 18,\n
\"max\": 64,\n        \"num_unique_values\": 47,\n        \"samples\":
[\n          21,\n          45,\n          36\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"male\",\n          \"female\"\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"bmi\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 6.098382190003363,\n
\"min\": 16.0,\n        \"max\": 53.1,\n        \"num_unique_values\":
275,\n        \"samples\": [\n          28.6,\n          20.9\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"children\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 0,\n        \"max\": 5,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"smoker\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"no\",\n          \"yes\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"region\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
4,\n        \"samples\": [\n          \"southeast\",\n
\"northeast\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"expenses\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12110.011239706468,\n        \"min\":
1121.87,\n        \"max\": 63770.43,\n        \"num_unique_values\":
1337,\n        \"samples\": [\n          8688.86,\n          5708.87\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
df['sex'].value_counts()
```

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

```
df['region'].value_counts()
```

```
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

```
df['smoker'].value_counts()
```

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

```
df['children'].value_counts()
```

```
children
0    574
1    324
2    240
3    157
4     25
5     18
Name: count, dtype: int64
```

```
df.dtypes
```

```
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
expenses    float64
dtype: object
```

```
df.shape
```

```
(1338, 7)
```

```
df.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   expenses  1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

```
df.describe().T
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 4,\n  \"fields\": [\n
{\n      \"column\": \"count\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.0,\n        \"min\":
1338.0,\n        \"max\": 1338.0,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          1338.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"mean\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 6623.403434584269,\n
\"min\": 1.0949177877429,\n        \"max\": 13270.422414050823,\n
\"num_unique_values\": 4,\n        \"samples\": [\n
30.66547085201794\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"std\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 6051.4489621592675,\n        \"min\": 1.205492739781914,\n
\"max\": 12110.011239706468,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          6.098382190003363\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"min\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 555.3267604678048,\n
\"min\": 0.0,\n        \"max\": 1121.87,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          16.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"25%\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 2361.293853844906,\n
\"min\": 0.0,\n        \"max\": 4740.2875,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          26.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"50%\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 4679.309951074516,\n
\"min\": 1.0,\n        \"max\": 9382.029999999999,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          30.4\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"75%\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 8305.365823774588,\n
\"min\": 2.0,\n        \"max\": 16639.915,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          34.7\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"max\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 31864.875309890853,\
n        \"min\": 5.0,\n        \"max\": 63770.43,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          53.1\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

```
df.isnull().sum()
```

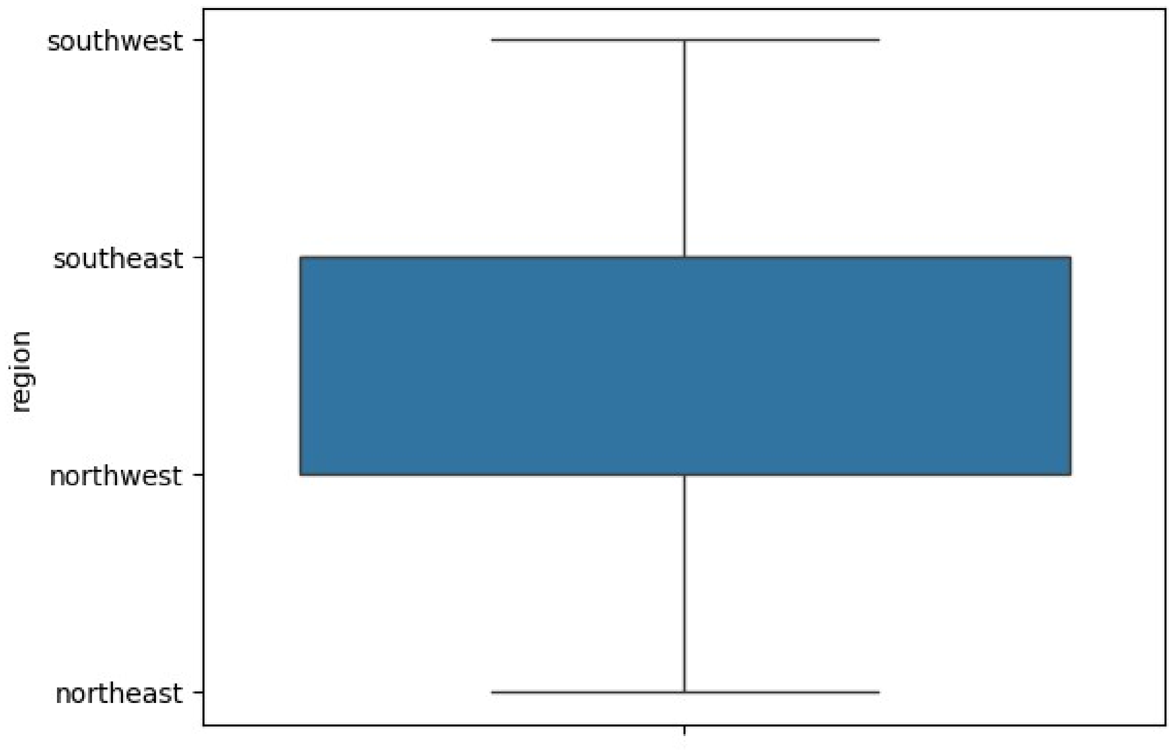
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
expenses    0
dtype: int64
```

```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()

df['sex']= label_encoder.fit_transform(df['sex'])
df['smoker']= label_encoder.fit_transform(df['smoker'])
```

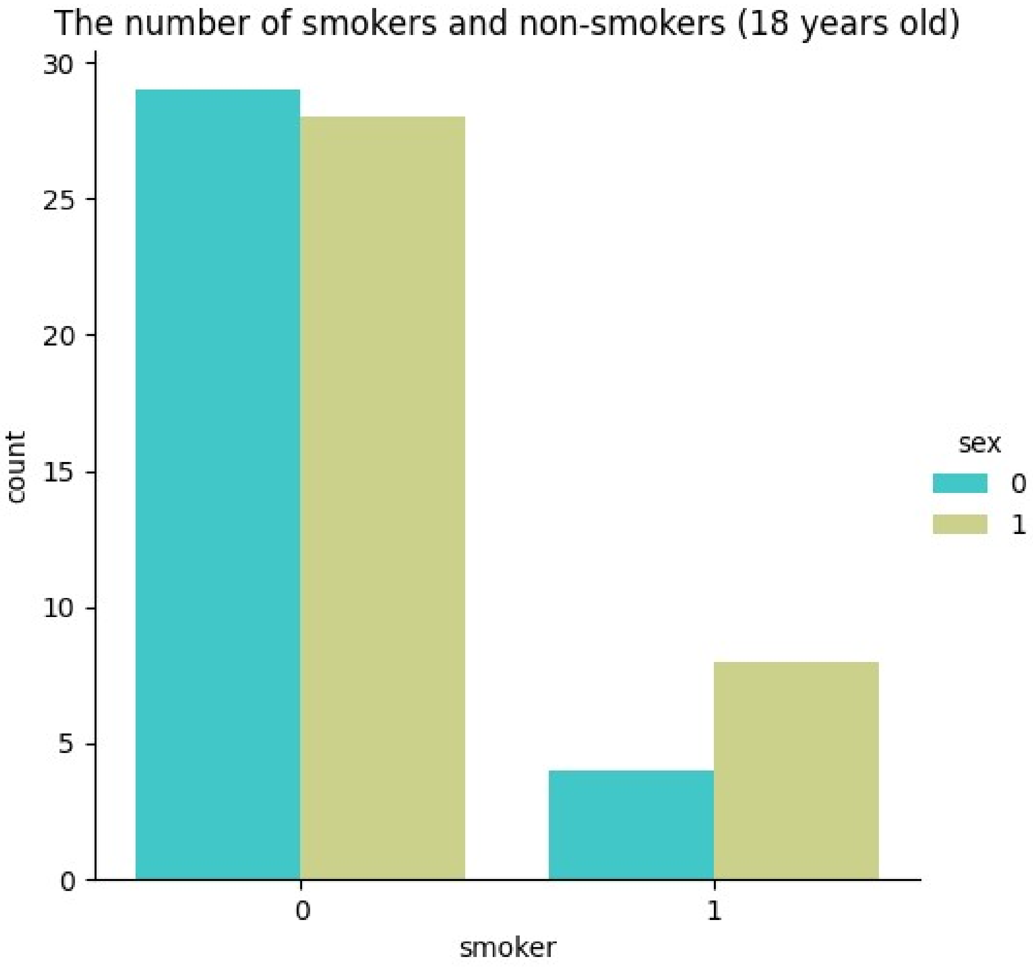
EDA and Visualizations

```
sns.boxplot(df['region'])
```

```
<Axes: ylabel='region'>
```

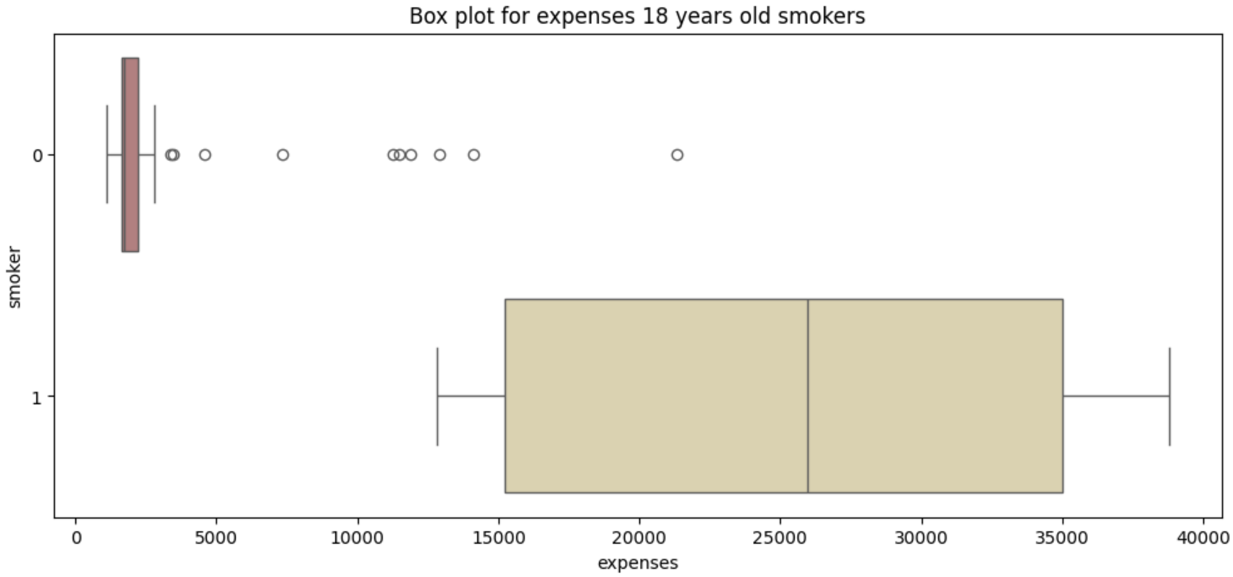


```
sns.catplot(x="smoker", kind="count",hue = 'sex', palette="rainbow",
data=df[(df.age == 18)])
plt.title("The number of smokers and non-smokers (18 years old)")

Text(0.5, 1.0, 'The number of smokers and non-smokers (18 years old)')
```



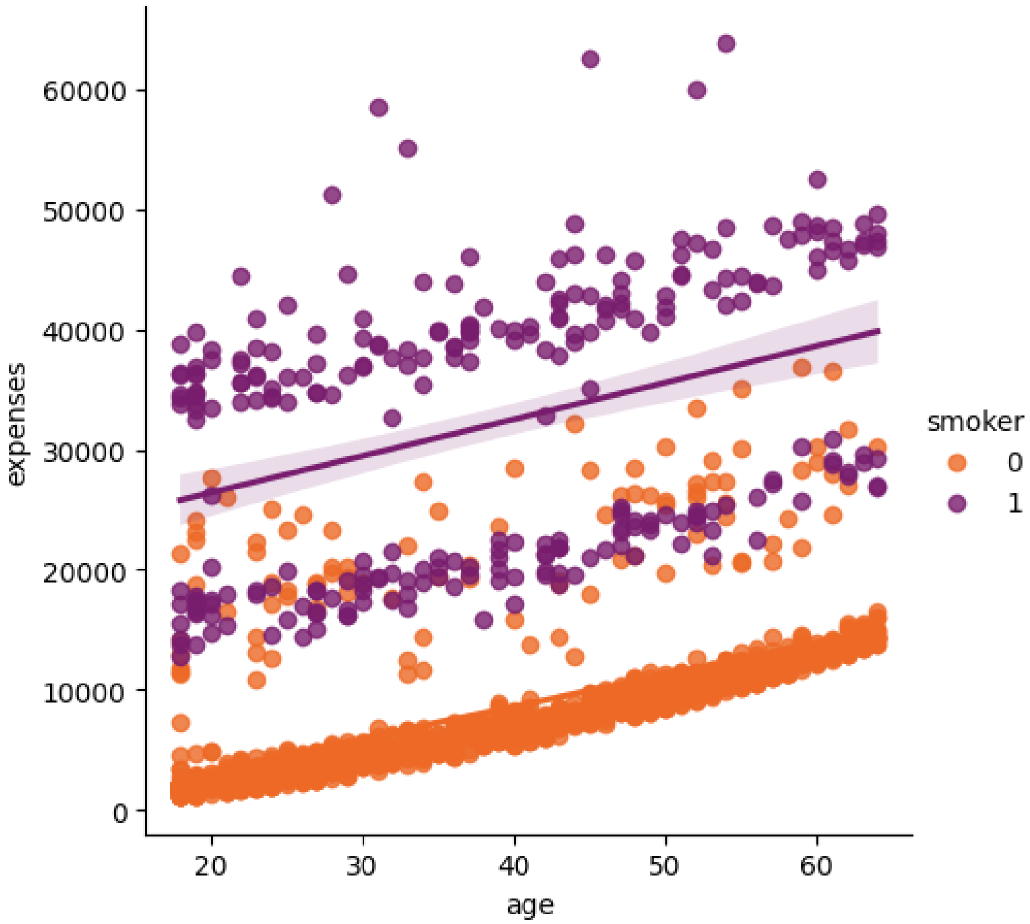
```
plt.figure(figsize=(12,5))
plt.title("Box plot for expenses 18 years old smokers")
sns.boxplot(y="smoker", x="expenses", data = df[(df.age == 18)] ,
orient="h", palette = 'pink')
```

```
<ipython-input-29-1b0f7b322340>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(y="smoker", x="expenses", data = df[(df.age == 18)] ,
orient="h", palette = 'pink')

<Axes: title={'center': 'Box plot for expenses 18 years old smokers'},
xlabel='expenses', ylabel='smoker'>
```



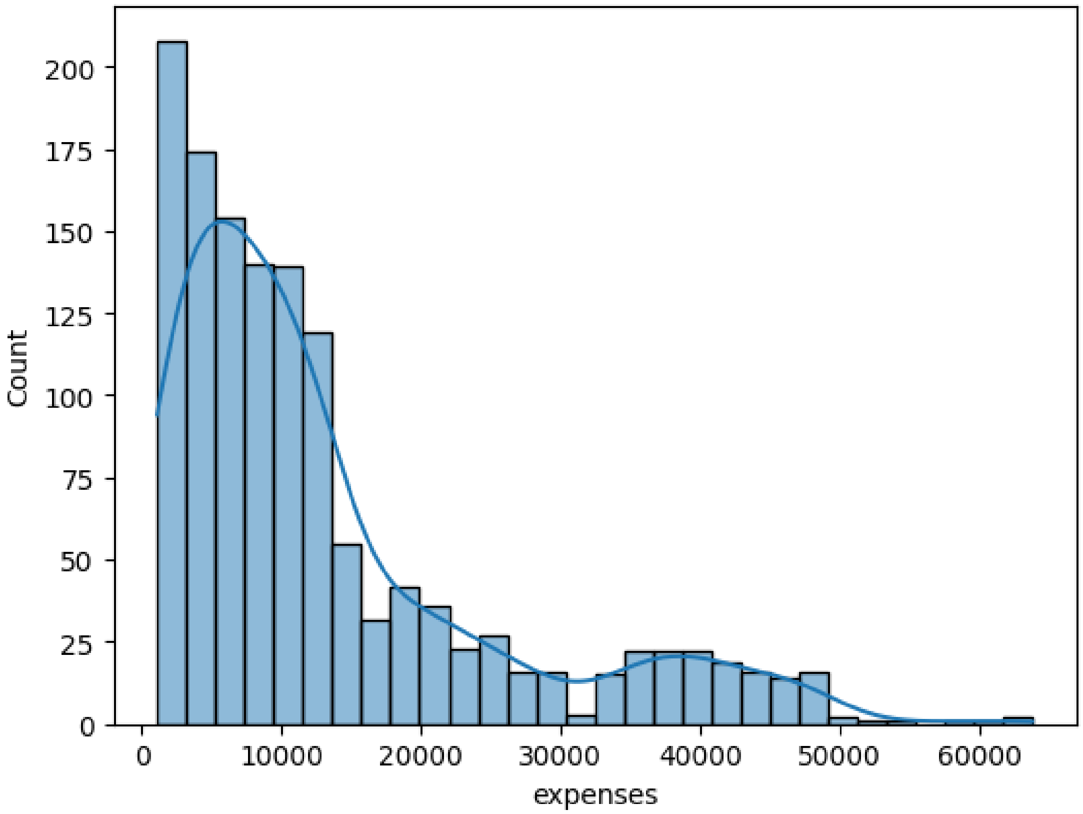
```
sns.lmplot(x="age", y="expenses", hue="smoker", data=df, palette =
'inferno_r')
```

```
<seaborn.axisgrid.FacetGrid at 0x7d4d135e2d10>
```

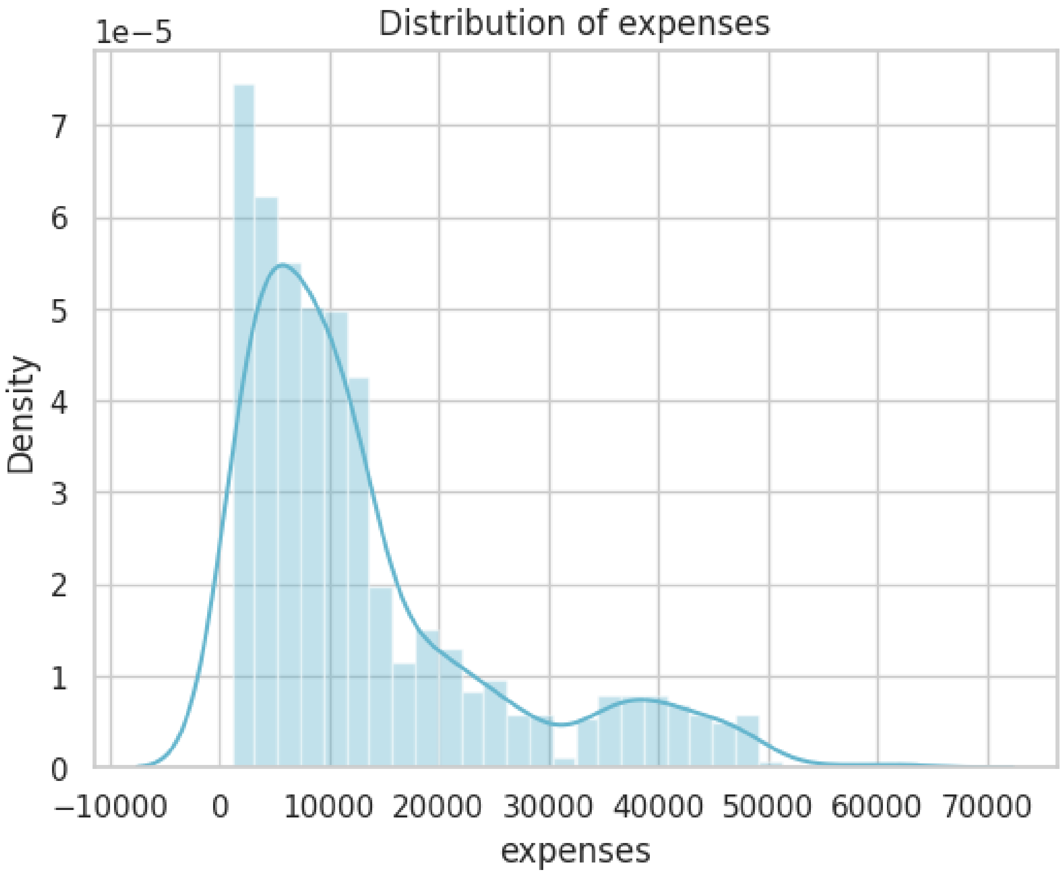


```
sns.histplot(data=df,x='expenses',kde=True)
```

```
<Axes: xlabel='expenses', ylabel='Count'>
```

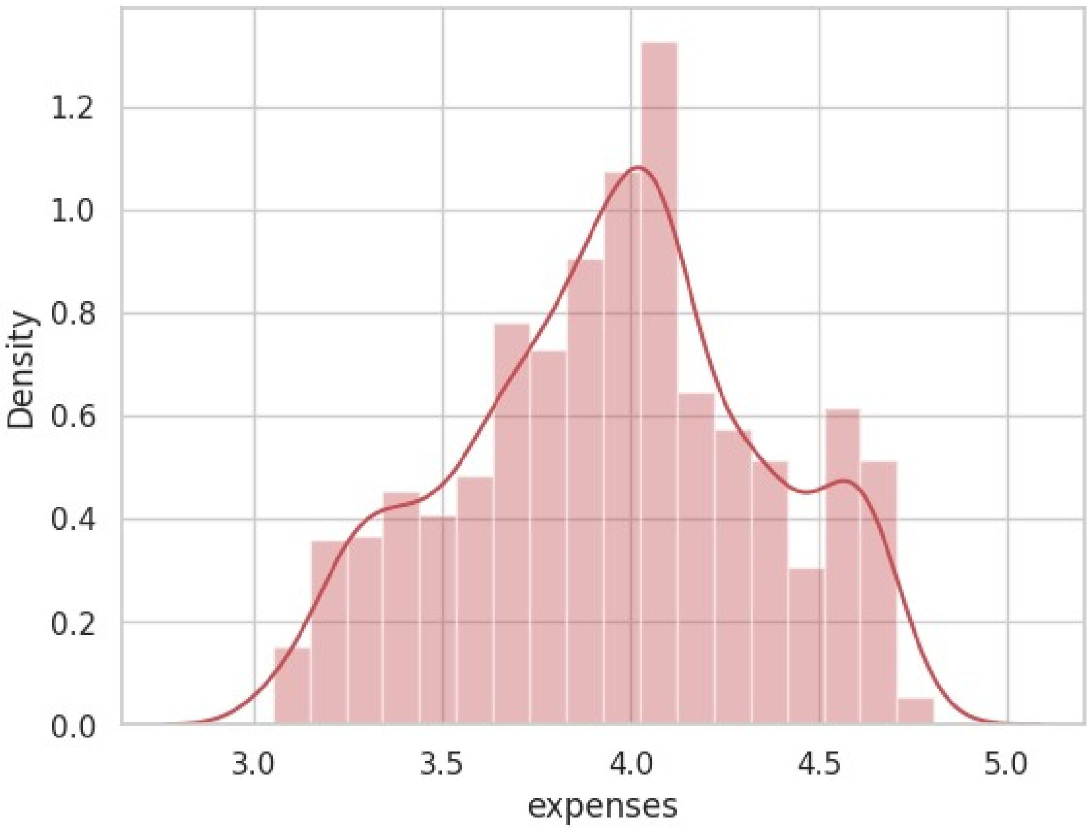


```
sns.set(style='whitegrid')
ax = sns.distplot(df['expenses'], kde = True, color = 'c')
plt.title('Distribution of expenses')

<ipython-input-31-8c979afc7dd3>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax = sns.distplot(df['expenses'], kde = True, color = 'c')

Text(0.5, 1.0, 'Distribution of expenses')
```



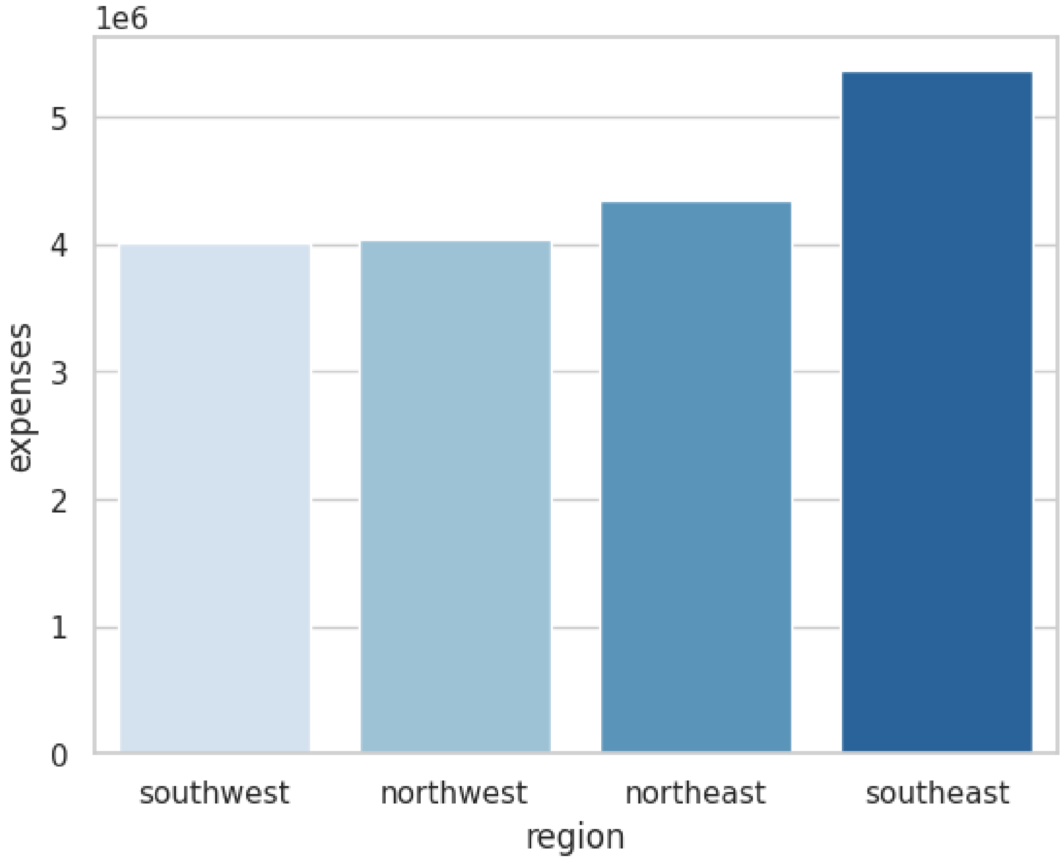
```
ax = sns.distplot(np.log10(df['expenses']), kde = True, color = 'r' )
```

```
<ipython-input-32-fa0e65f84bf2>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax = sns.distplot(np.log10(df['expenses']), kde = True, color =
'r' )
```

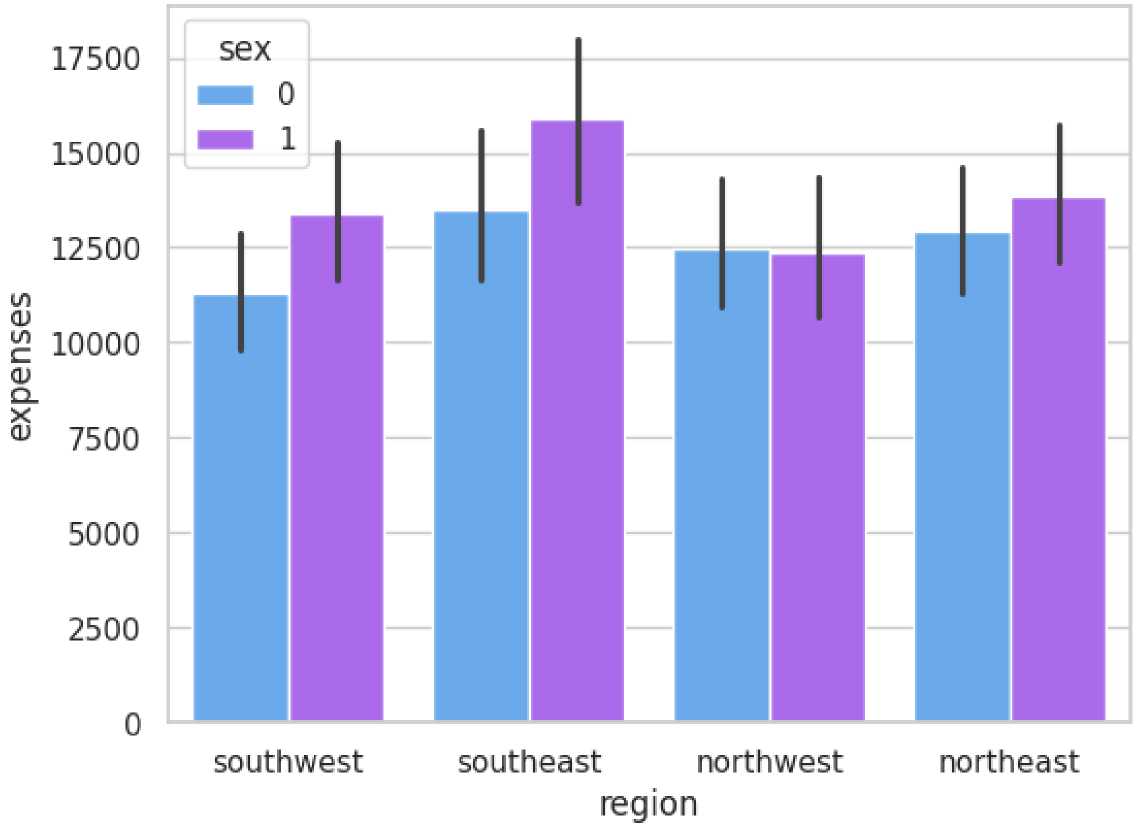


```
c = df['expenses'].groupby(df['region']).sum().sort_values(ascending =
True)
c = c.head()

sns.barplot(x=c.index, y=c, palette='Blues')
```

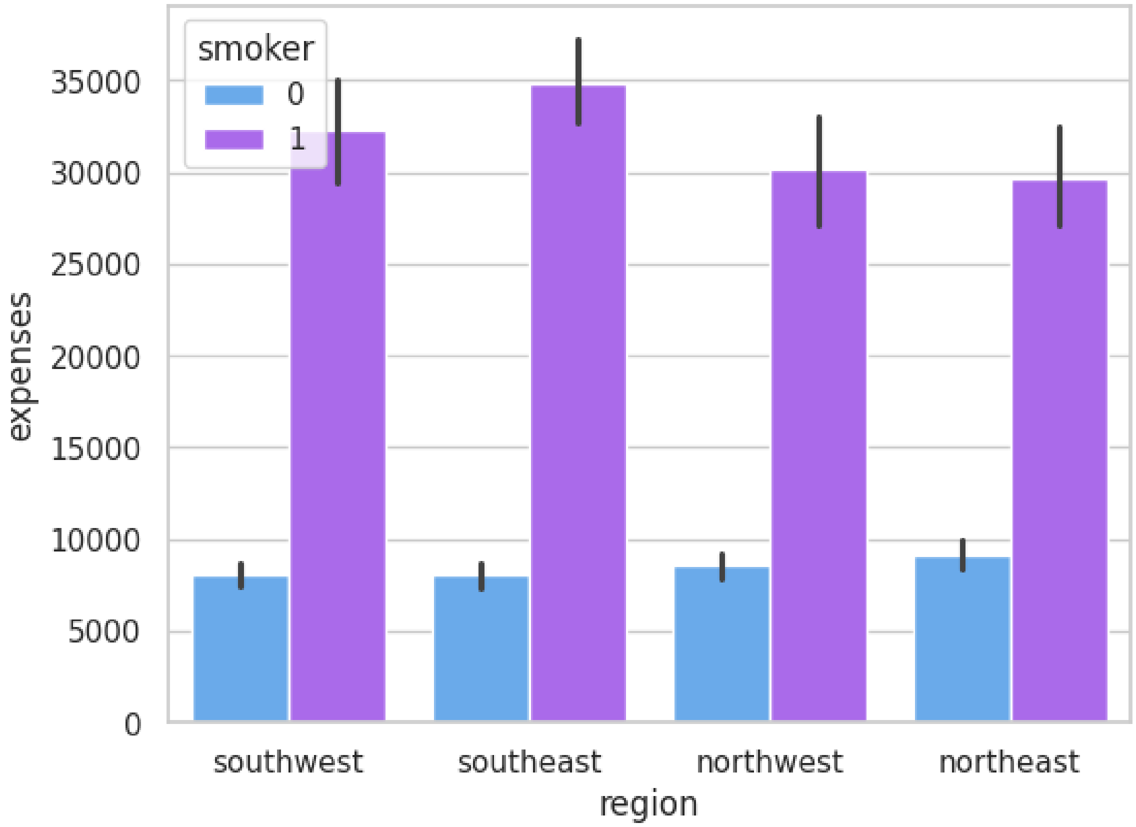
```
<ipython-input-35-aa3bb2842d4f>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=c.index, y=c, palette='Blues')

<Axes: xlabel='region', ylabel='expenses'>
```



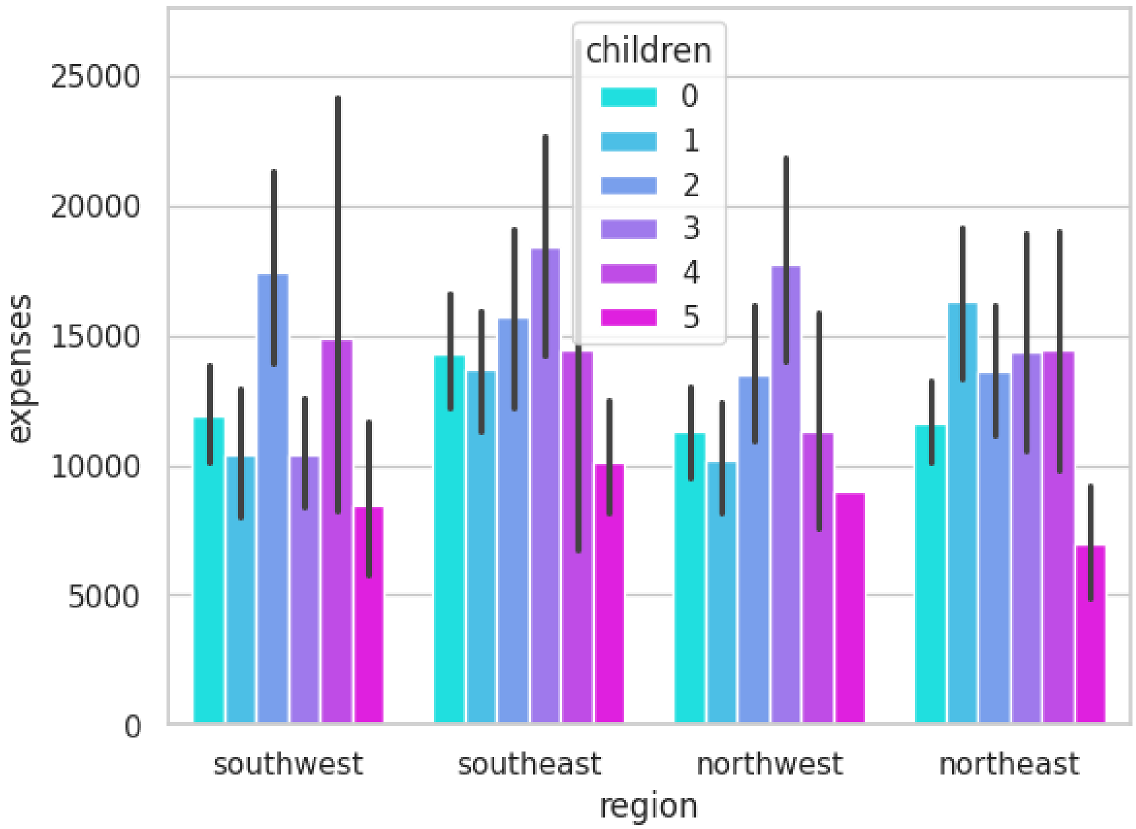
```
ax = sns.barplot(x='region', y='expenses', hue='sex', data=df,
palette='cool')
```



```
ax = sns.barplot(x='region', y='expenses', hue='smoker', data=df,
palette='cool')
```



```
ax = sns.barplot(x='region', y='expenses', hue='children', data=df,
palette='cool')
```

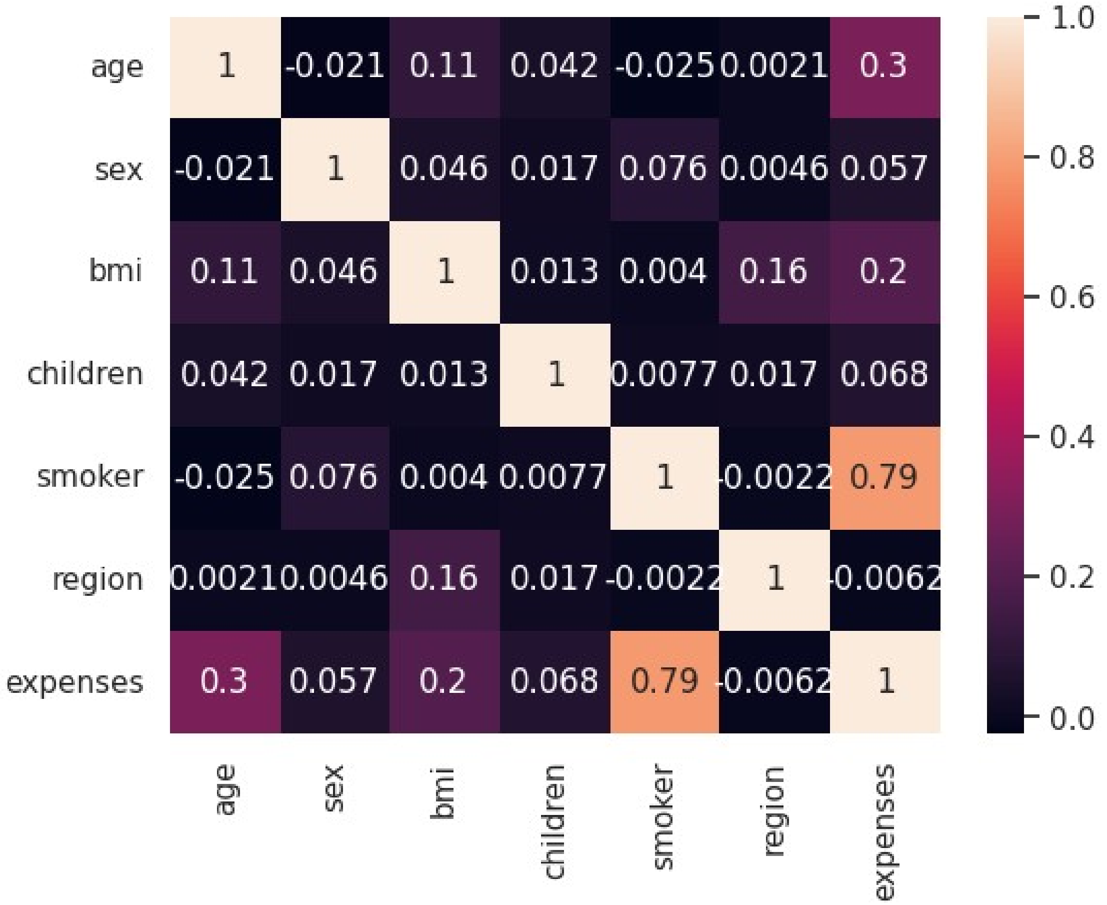


```
df['region']= label_encoder.fit_transform(df['region'])
df
```

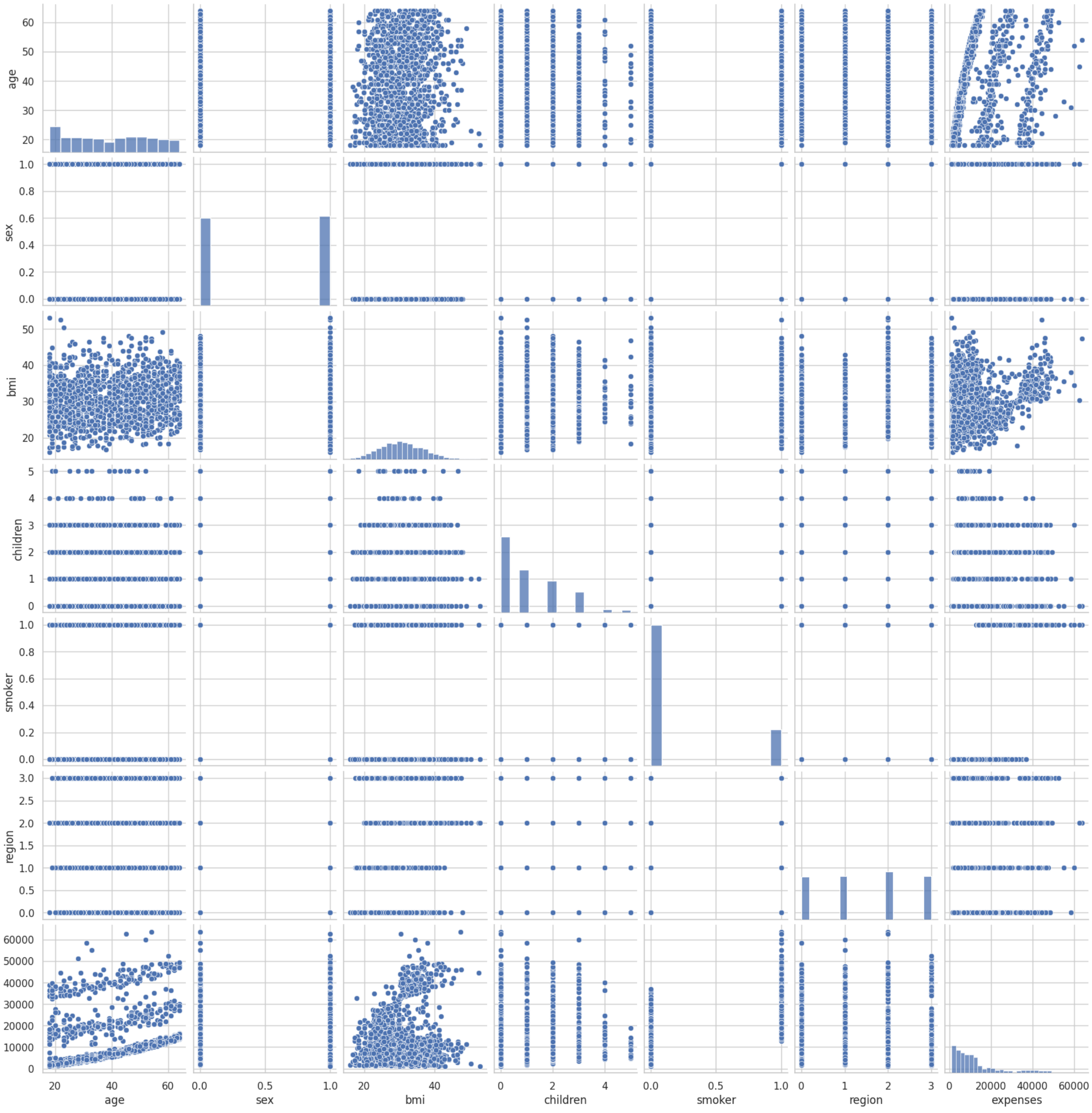
```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 1338,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 18,\n
\"max\": 64,\n        \"num_unique_values\": 47,\n        \"samples\":
[\n          21,\n          45,\n          36\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"sex\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"bmi\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 6.098382190003363,\n        \"min\":
16.0,\n        \"max\": 53.1,\n        \"num_unique_values\": 275,\n
\"samples\": [\n          28.6,\n          20.9\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"children\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 1,\n
\"min\": 0,\n        \"max\": 5,\n        \"num_unique_values\": 6,\n
\"samples\": [\n          0,\n          1\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"smoker\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          0,\n          1\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"region\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 1,\n
\"min\": 0,\n        \"max\": 3,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          2,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"expenses\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
12110.011239706468,\n        \"min\": 1121.87,\n        \"max\":
63770.43,\n        \"num_unique_values\": 1337,\n        \"samples\":
[\n          8688.86,\n          5708.87\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```



```
sns.pairplot(df)

<seaborn.axisgrid.PairGrid at 0x7d4d10896e90>
```


age
sex
bmi
children
smoker
region
expenses
age
sex
bmi
children
smoker
region
expenses