Exploratry_Data_Analysis_IN_Jupyter_NoteBook

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv("D:\\datasets\\insurance.csv")
df
```

Out[2]:

	age	sex	bmi	children	smoker	region	expenses
0	19	female	27.9	0	yes	southwest	16884.92
1	18	male	33.8	1	no	southeast	1725.55
2	28	male	33.0	3	no	southeast	4449.46
3	33	male	22.7	0	no	northwest	21984.47
4	32	male	28.9	0	no	northwest	3866.86
...	...	...	...	...	...	...	...
1333	50	male	31.0	3	no	northwest	10600.55
1334	18	female	31.9	0	no	northeast	2205.98
1335	18	female	36.9	0	no	southeast	1629.83
1336	21	female	25.8	0	no	southwest	2007.95
1337	61	female	29.1	0	yes	northwest	29141.36

1338 rows × 7 columns

_univariate statistics

In [3]:
```
df.describe()
```

Out[3]:

	age	bmi	children	expenses
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.665471	1.094918	13270.422414
std	14.049960	6.098382	1.205493	12110.011240
min	18.000000	16.000000	0.000000	1121.870000
25%	27.000000	26.300000	0.000000	4740.287500
50%	39.000000	30.400000	1.000000	9382.030000
75%	51.000000	34.700000	2.000000	16639.915000
max	64.000000	53.100000	5.000000	63770.430000

In [4]:
```
#culoums & rows
df.shape
```

Out[4]:
```
(1338, 7)
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'expenses'], dtype='object')
```

In [6]:
```
#number of students
df.age.count()
```

Out[6]:
```
1338
```

In [7]:
```
print(f'age: {df.age.count()}')
print(f'sex: {df.sex.count()}')
print(f'bmi: {df.bmi.count()}')
print(f'children: {df.children.count()}')
print(f'smoker: {df.smoker.count()}')
print(f'region: {df.region.count()}')
print(f'expenses: {df.expenses.count()}')
```

```
age: 1338
sex: 1338
bmi: 1338
children: 1338
smoker: 1338
region: 1338
expenses: 1338
```

In [8]:

```
print(f'age: {df.age.dtypes}')
print(f'sex: {df.sex.dtypes}')
print(f'bmi: {df.bmi.dtypes}')
print(f'children: {df.children.dtypes}')
print(f'smoker: {df.smoker.dtypes}')
print(f'region: {df.region.dtypes}')
print(f'expenses: {df.expenses.dtypes}')
```

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

In [9]:

```
print(f'age: {pd.api.types.is_numeric_dtype(df.age)}')
print(f'sex: {pd.api.types.is_numeric_dtype(df.sex)}')
print(f'bmi: {pd.api.types.is_numeric_dtype(df.bmi)}')
print(f'children: {pd.api.types.is_numeric_dtype(df.children)}')
print(f'smoker: {pd.api.types.is_numeric_dtype(df.smoker)}')
print(f'region: {pd.api.types.is_numeric_dtype(df.region)}')
print(f'expenses: {pd.api.types.is_numeric_dtype(df.expenses)}')
```

```
age: True
sex: False
bmi: True
children: True
smoker: False
region: False
expenses: True
```

In [10]:

```
print(f'age: {df.age.isnull().sum()}')
print(f'sex: {df.sex.isnull().sum()}')
print(f'bmi: {df.bmi.isnull().sum()}')
```

```
print(f'children: {df.children.isnull().sum()}')
print(f'smoker: {df.smoker.isnull().sum()}')
print(f'region: {df.region.isnull().sum()}')
print(f'expenses: {df.expenses.isnull().sum()}')
```

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

In [11]:

```
print(df.expenses.min())
print(df.expenses.max())
print(df.expenses.quantile(.25))
print(df.expenses.quantile(.50))
print(df.expenses.quantile(.75))
print(df.expenses.mean())
print(df.expenses.median())
print(df.expenses.mode())
```

```
1121.87
63770.43
4740.2875
9382.029999999999
16639.915
13270.422414050803
9382.029999999999
0    1639.56
dtype: float64
```

In [12]:

```
df.expenses.std()
```

Out[12]:

```
12110.011239706457
```

In [13]:

```
np.std(df.expenses, ddof=1)
```

Out[13]:

```
12110.011239706457
```

In [14]:

```
df.expenses.var()
```

Out[14]:
```
146652372.22581673
```

In [15]:
```
from scipy.stats import stats
```

In [16]:
```
print(df.expenses.kurtosis())
print(df.expenses.skew())
```

```
1.6062986577747589
1.51587966289798
```

_Correlation and P-value

In [17]:
```
df=pd.read_csv("D:\\datasets\\insurance.csv")
df.corr()
```

Out[17]:

	age	bmi	children	expenses
age	1.000000	0.109341	0.042469	0.299008
bmi	0.109341	1.000000	0.012645	0.198576
children	0.042469	0.012645	1.000000	0.067998
expenses	0.299008	0.198576	0.067998	1.000000

In [18]:
```
df.expenses.corr(df.bmi)
```

Out[18]:
```
0.198576255018932
```

corr & p value

In [19]:
```
from scipy.stats import stats
r, p =stats.pearsonr(df.expenses, df.age)
print(round(r,5))
print(round(p,5))
```

```
0.29901
0.0
```

In [20]:

```
corr_df=pd.DataFrame(columns=['r', 'p'])

for col in df:
    print(col)
    if pd.api.types.is_numeric_dtype(df[col]):
        r,p = stats.pearsonr(df.expenses, df[col])
        corr_df.loc[col]=[round(r, 3), round(p, 3)]
corr_df
```

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

Out[20]:

	r	p
age	0.299	0.000
bmi	0.199	0.000
children	0.068	0.013
expenses	1.000	0.000

In [21]:

```
corr_df=pd.DataFrame(columns=['r', 'p'])

for col in df:
    print(col)
    if pd.api.types.is_numeric_dtype(df[col]) and col  != 'expenses':
        r,p = stats.pearsonr(df.expenses, df[col])
        corr_df.loc[col]=[round(r, 3), round(p, 3)]
corr_df
```

```
age
sex
bmi
children
smoker
```

```
region
expenses
```

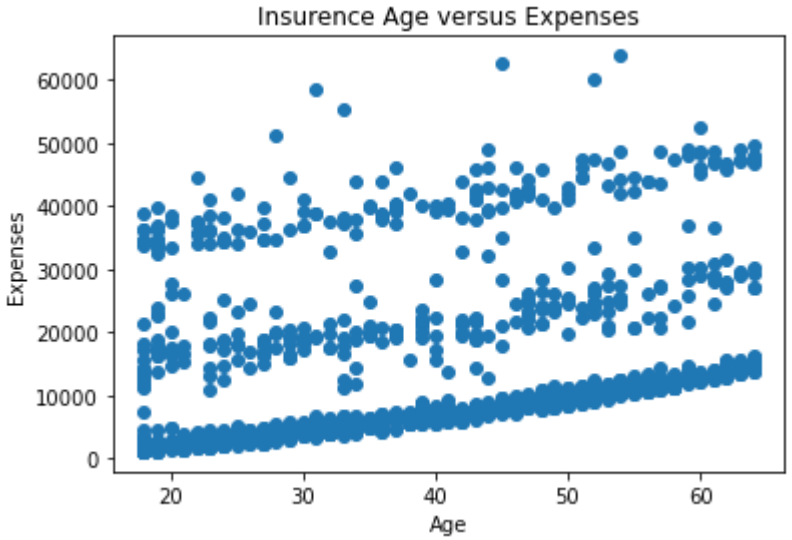
Out[21]:

	r	p
age	0.299	0.000
bmi	0.199	0.000
children	0.068	0.013

Scatterplots Linear Regression_ Heteroscedasticity

In [22]:

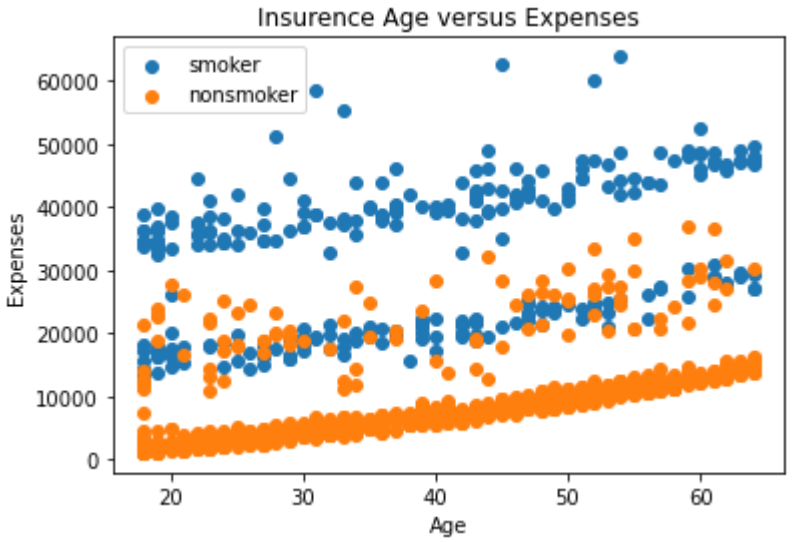
```
import matplotlib.pyplot as plt
plt.scatter(df.age, df.expenses)
plt.title("Insurence Age versus Expenses")
plt.xlabel("Age")
plt.ylabel("Expenses")
plt.show()
```



In [23]:

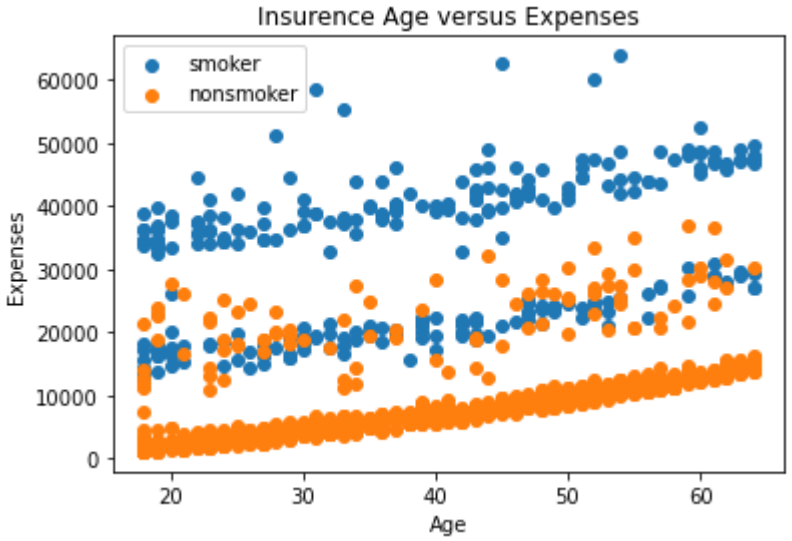
```
df_smoker=df[df['smoker']=='yes']
df_nonsmoker=df[df['smoker']=='no']
plt.scatter(df_smoker.age, df_smoker.expenses, label='smoker')
plt.scatter(df_nonsmoker.age, df_nonsmoker.expenses, label='nonsmoker')
```

```
plt.title("Insurence Age versus Expenses")
plt.xlabel("Age")
plt.ylabel("Expenses")
plt.legend()
plt.show()
```



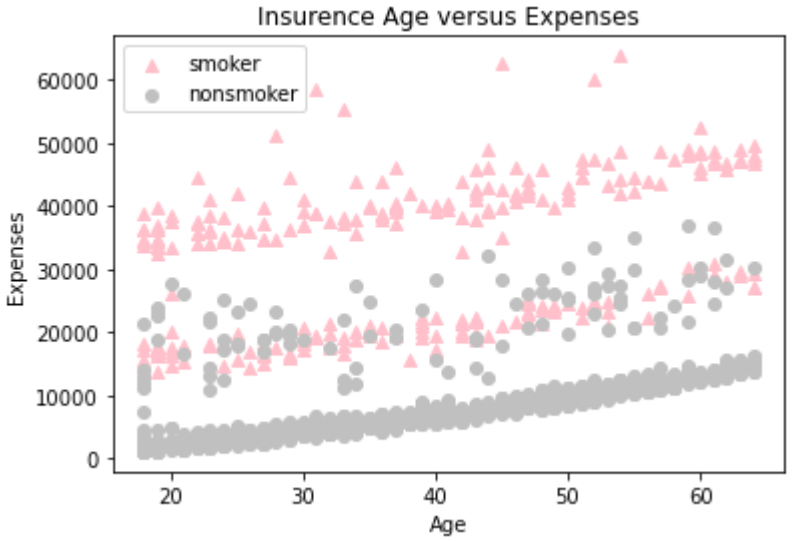
In [24]:

```
df_smoker_reduced= df_smoker.sample(50)
df_nonsmoker_reduced= df_nonsmoker.sample(50)

plt.scatter(df_smoker.age, df_smoker.expenses, label='smoker')
plt.scatter(df_nonsmoker.age, df_nonsmoker.expenses, label='nonsmoker')

plt.title("Insurence Age versus Expenses")
plt.xlabel("Age")
plt.ylabel("Expenses")
plt.legend()
plt.show()
```



In [25]:

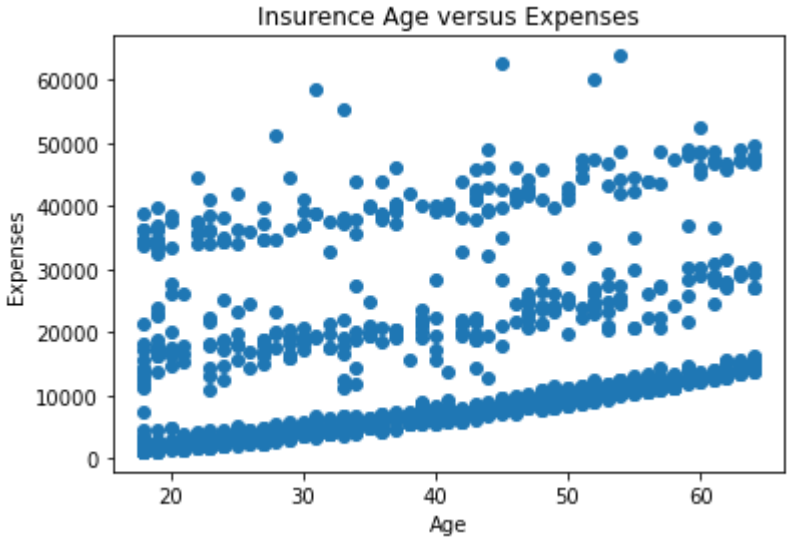
```
df_smoker_reduced= df_smoker.sample(25)
df_nonsmoker_reduced= df_nonsmoker.sample(25)

plt.scatter(df_smoker.age, df_smoker.expenses, label='smoker', color='pink', marker='^')
plt.scatter(df_nonsmoker.age, df_nonsmoker.expenses, label='nonsmoker', color='silver', marker='o')

plt.title("Insurence Age versus Expenses")
plt.xlabel("Age")
plt.ylabel("Expenses")
plt.legend()
plt.show()
```



In [26]:

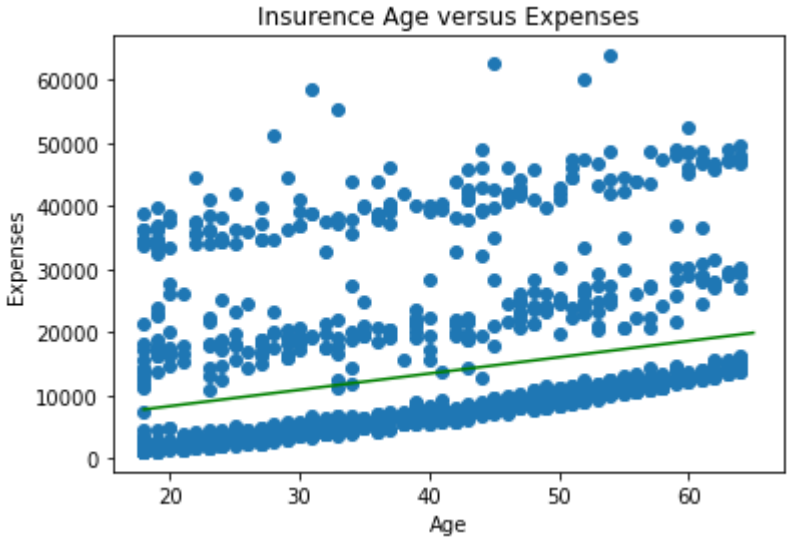
```
from scipy import stats
print(stats.linregress(df.age, df.expenses))

plt.scatter(df.age, df.expenses)
plt.title("Insurence Age versus Expenses")
plt.xlabel("Age")
plt.ylabel("Expenses")

plt.show()
```

```
LinregressResult(slope=257.72261782980775, intercept=3165.8851877923134, rvalue=0.2990081922850828, pvalue=4.886695589990
2494e-29, stderr=22.50238930009366, intercept_stderr=937.1494656252503)
```



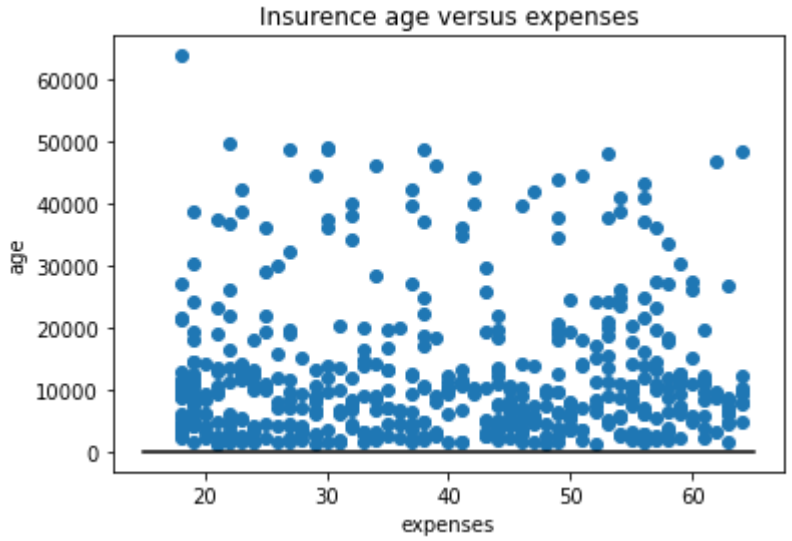
In [27]:

```
# y=mx+b
# y=slop(x)+intersept

from scipy import stats
m, b, r, p, err=stats.linregress(df.age, df.expenses)
x=range(18, df.age.max()+2)
y=m*x+b
plt.plot(x, y, color='green')
plt.scatter(df.age, df.expenses)
plt.title("Insurence Age versus Expenses")
plt.xlabel("Age")
plt.ylabel("Expenses")

plt.show()
```



In [28]:

```
# y=mx+b
# y=slop(x)+intersept
from scipy import stats
m, b, r, p, err=stats.linregress(df.expenses, df.age)
x=range(15, df.age.max()+2)
y=m*x+b
plt.plot(x, y, color='black')
plt.scatter(df.age.sample(500), df.expenses.sample(500))
plt.title("Insurence age versus expenses")
plt.xlabel("expenses")
plt.ylabel("age")
plt.show()
```



In [29]:

```
from statsmodels.stats.diagnostic import het_breuschpagan
from statsmodels.stats.diagnostic import het_white
from statsmodels.formula.api import ols

model=ols(formula='age~expenses', data=df).fit()

white_test= het_white(model.resid, model.model.exog)
breuschpagan_test= het_breuschpagan(model.resid, model.model.exog)

output_df=pd.DataFrame(columns=['LM stat', 'LM P', 'F stat', 'F stat p'])
output_df.loc['white']=white_test
output_df.loc['Breusch_pagan']=breuschpagan_test

output_df
```

Out[29]:

	LM stat	LM P	F stat	F stat p
white	113.205742	2.616288e-25	61.695940	2.358801e-26
Breusch_pagan	48.227291	3.795686e-12	49.955826	2.525659e-12

In [30]:

```
from statsmodels.stats.diagnostic import het_breuschpagan
from statsmodels.stats.diagnostic import het_white
from statsmodels.formula.api import ols
```

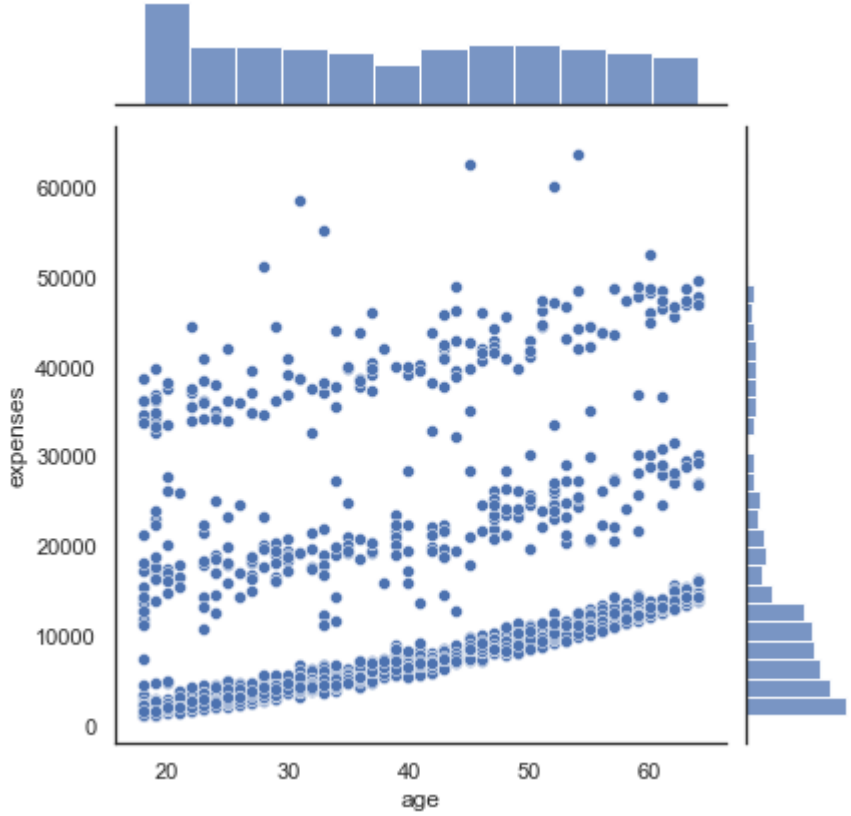
```
model=ols(formula='expenses~age', data=df).fit()

white_test= het_white(model.resid, model.model.exog)
breuschpagan_test= het_breuschpagan(model.resid, model.model.exog)

output_df=pd.DataFrame(columns=['LM stat', 'LM P', 'F stat', 'F stat p'])
output_df.loc['white']=white_test
output_df.loc['Breusch_pagan']=breuschpagan_test

output_df
```

Out[30]:

	LM stat	LM P	F stat	F stat p
white	0.002713	0.998645	0.001353	0.998648
Breusch_pagan	0.000414	0.983776	0.000413	0.983791

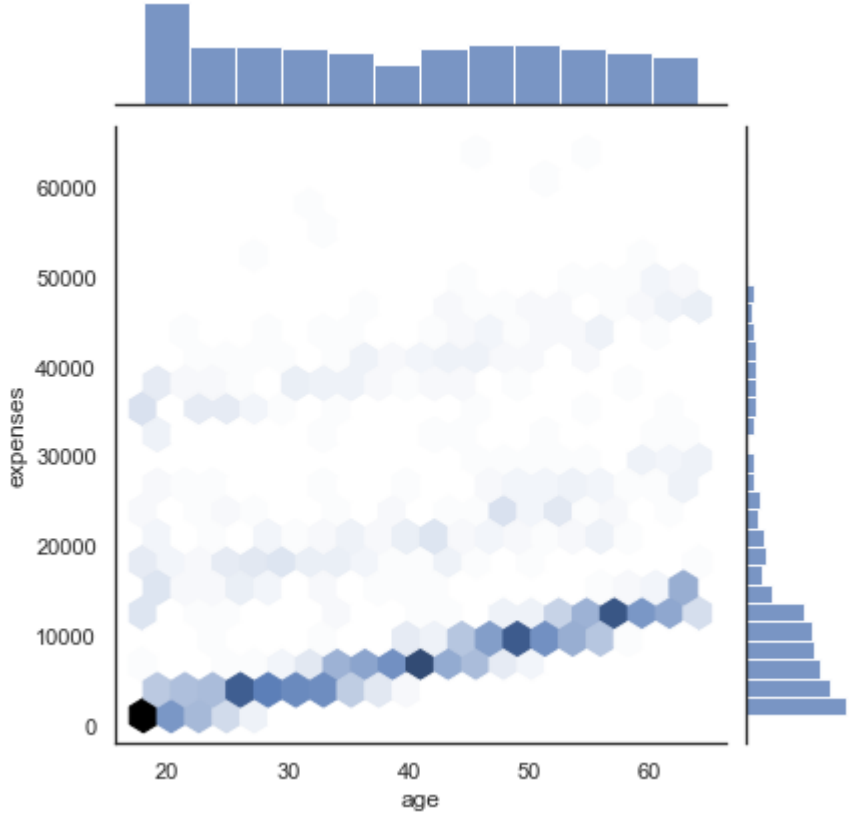
In [31]:

```
import seaborn as sns

sns.set(color_codes='True')
sns.set_style('white')
sns.jointplot(x='age', y='expenses', data=df)
plt.show()
```



In [32]:

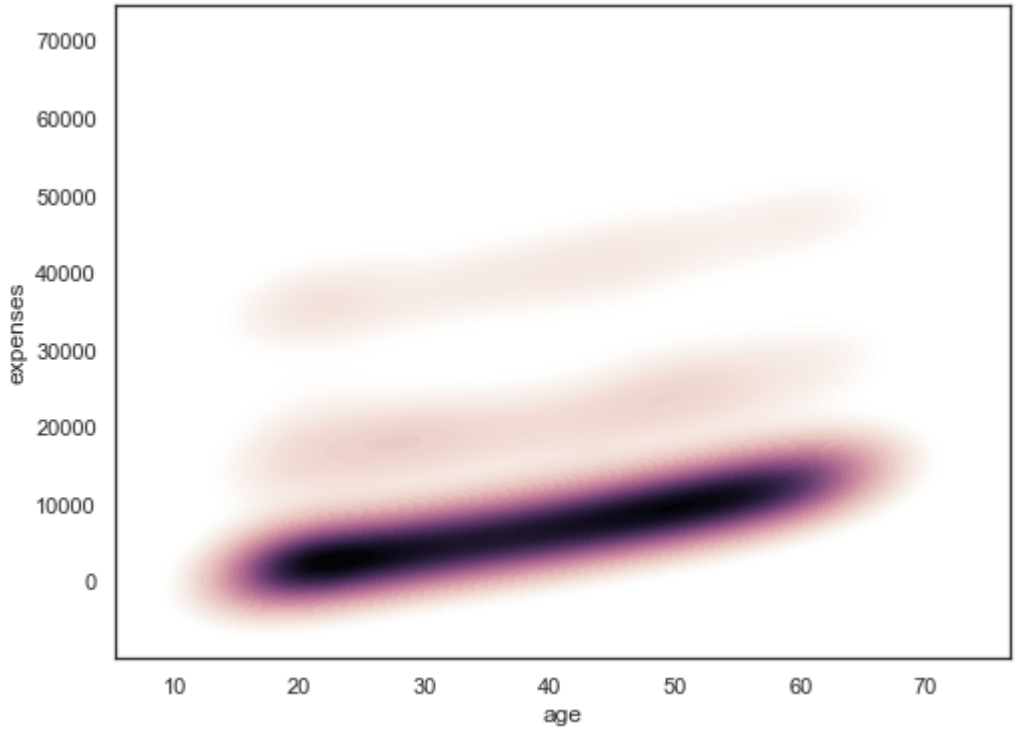
```
sns.set_style('white')
sns.jointplot(x='age', y='expenses', data=df, kind='hex');
```



In [33]:
```
f, ax =plt.subplots(figsize=(8, 6))
cmap= sns.cubehelix_palette(as_cmap=True, dark=0, light=1, reverse=False)
sns.kdeplot(df.age, df.expenses, cmap=cmap, n_levels=60, shade=True)
```

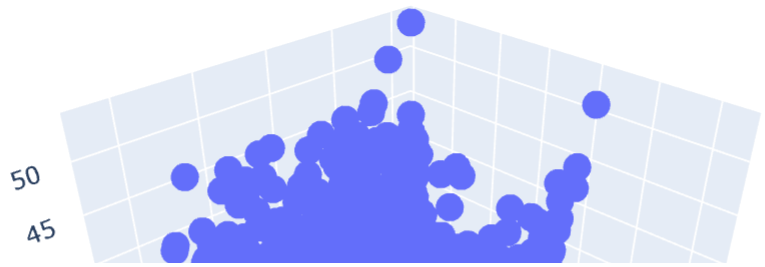
```
C:\Users\TAWAB COMPUTERS\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

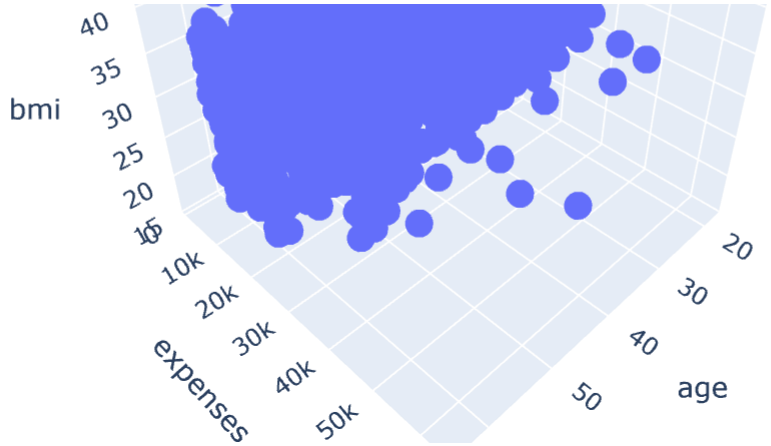
Out[33]:
```
<AxesSubplot:xlabel='age', ylabel='expenses'>
```



In [34]:

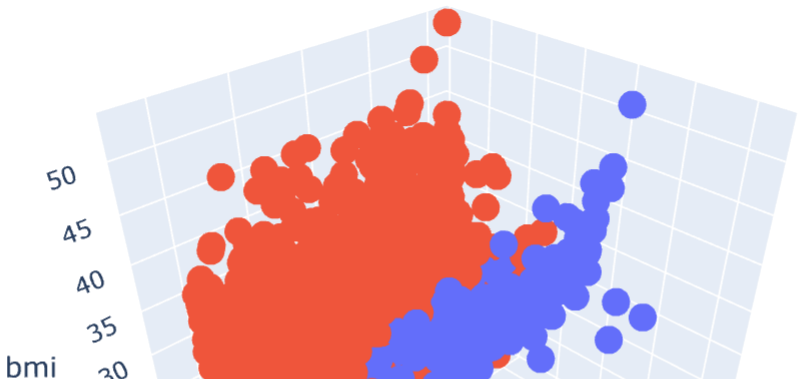
```
import pandas as pd
import plotly.express as px

fig= px.scatter_3d(df, x="age", y="expenses", z="bmi")
fig.show()
```

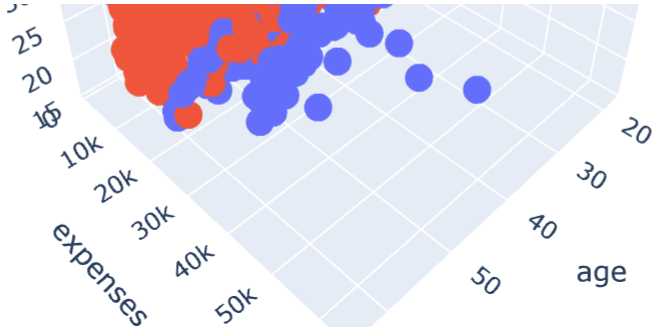




In [35]:

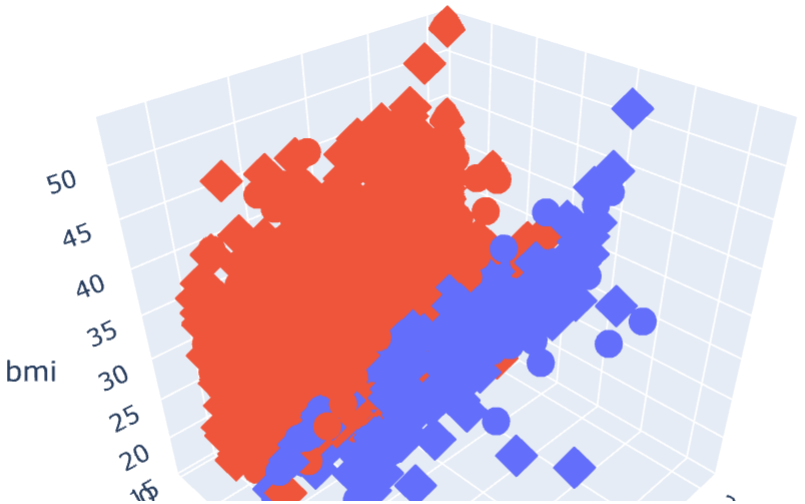
```
import pandas as pd
import plotly.express as px

fig= px.scatter_3d(df, x="age", y="expenses", z="bmi", color='smoker')
fig.show()
```





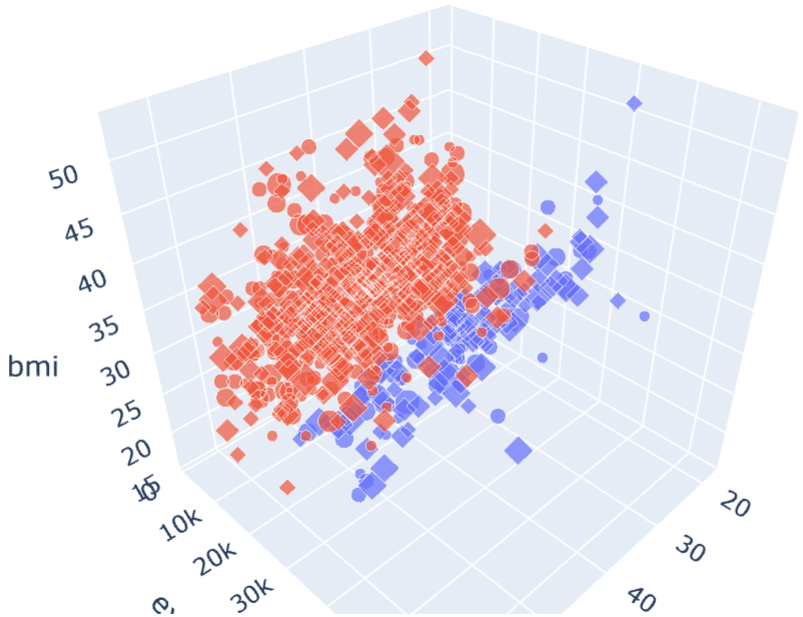
In [36]:

```
import pandas as pd
import plotly.express as px

fig= px.scatter_3d(df, x="age", y="expenses", z="bmi", color='smoker', symbol='sex')
fig.show()
```



In [37]:

```
import pandas as pd
import plotly.express as px

fig= px.scatter_3d(df, x="age", y="expenses", z="bmi", color='smoker', symbol='sex', size='children')
fig.show()
```




xpenses
40k
50k
50
age