In [165]:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression as lm
from sklearn.metrics import mean_squared_error, r2_score
```

In [194]:

```
somok = lambda cad: 2 if(cad=="yes") else 1
sexx = lambda cad:1 if(cad == "female") else 2
somok_lm = lambda cad: 1 if(cad=="yes") else 0
sexx_lm = lambda cad:0 if(cad == "female") else 1

Regions = {"northeast":1,"northwest":2,"southeast":3,"southwest":4}
regi = lambda cad: Regions[cad]

names_h=["age","sex","bmi","children","smoker","region","expenses"]
```

In [195]:

```
datas = pd.read_csv("insurance.csv",converters={"sex":sexx,"smoker":somok,"region":regi})
datas_lm = pd.read_csv("insurance.csv",converters={"sex":sexx_lm,"smoker":somok_lm,"region":regi})
```

In [196]:

```
print(str(datas))
print(str(datas_lm))
```

```
      age  sex     bmi  children  smoker  region      charges
0      19    1  27.900         0       2       4  16884.92400
1      18    2  33.770         1       1       3   1725.55230
2      28    2  33.000         3       1       3   4449.46200
3      33    2  22.705         0       1       2  21984.47061
4      32    2  28.880         0       1       2   3866.85520
...   ...  ...     ...       ...     ...     ...          ...
1333   50    2  30.970         3       1       2  10600.54830
1334   18    1  31.920         0       1       1   2205.98080
1335   18    1  36.850         0       1       3   1629.83350
1336   21    1  25.800         0       1       4   2007.94500
1337   61    1  29.070         0       2       2  29141.36030

[1338 rows x 7 columns]
      age  sex     bmi  children  smoker  region      charges
0      19    0  27.900         0       1       4  16884.92400
1      18    1  33.770         1       0       3   1725.55230
2      28    1  33.000         3       0       3   4449.46200
3      33    1  22.705         0       0       2  21984.47061
4      32    1  28.880         0       0       2   3866.85520
...   ...  ...     ...       ...     ...     ...          ...
1333   50    1  30.970         3       0       2  10600.54830
1334   18    0  31.920         0       0       1   2205.98080
1335   18    0  36.850         0       0       3   1629.83350
1336   21    0  25.800         0       0       4   2007.94500
1337   61    0  29.070         0       1       2  29141.36030

[1338 rows x 7 columns]
```

In [61]:

```
datas.agg({"charges":['min', 'max', 'median', 'skew',"mean"]})
```

Out[61]:

	charges
min	1121.873900
max	63770.428010
median	9382.033000
skew	1.515880
mean	13270.422265

In [73]:

```
datas["charges"].describe()
```

Out[73]:

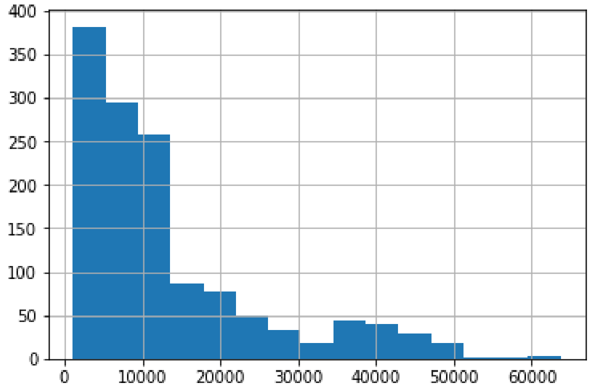
```
count     1338.000000
mean     13270.422265
std      12110.011237
min       1121.873900
25%       4740.287150
50%       9382.033000
75%      16639.912515
max      63770.428010
Name: charges, dtype: float64
```

In [99]:

```
datas["charges"].hist(bins=15)
```

Out[99]:

```
<AxesSubplot:>
```



In [100]:

```
datas["region"]
```

Out[100]:

```
0       4
1       3
2       3
3       2
4       2
       ..
1333    2
1334    1
1335    3
1336    4
1337    2
Name: region, Length: 1338, dtype: int64
```

In [102]:

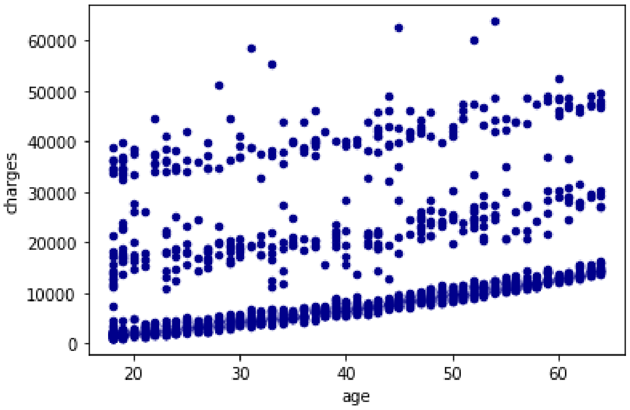
```
datas[["age", "bmi", "children", "charges"]].corr()
```

Out[102]:

	age	bmi	children	charges
age	1.000000	0.109272	0.042469	0.299008
bmi	0.109272	1.000000	0.012759	0.198341
children	0.042469	0.012759	1.000000	0.067998
charges	0.299008	0.198341	0.067998	1.000000

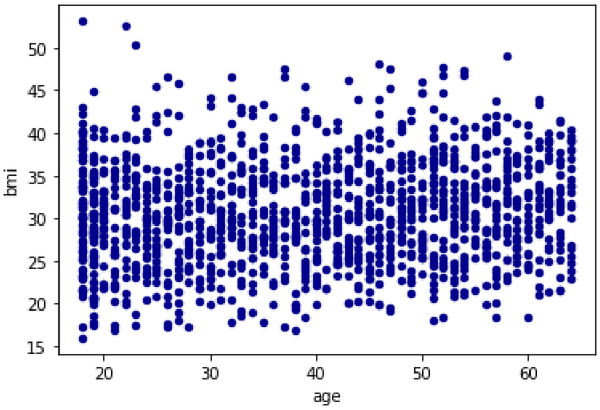
In [106]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="age",y="charges", c='DarkBlue')
```



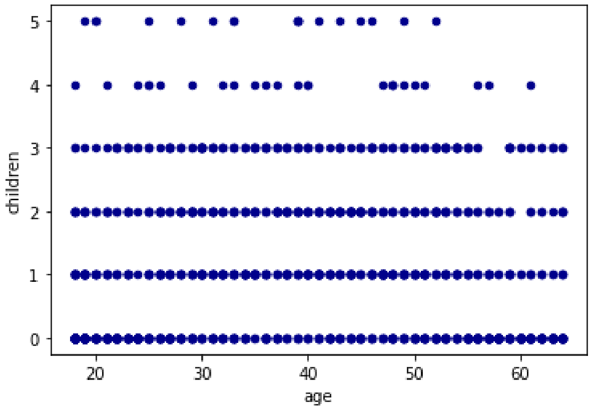
In [107]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="age",y="bmi", c='DarkBlue')
```



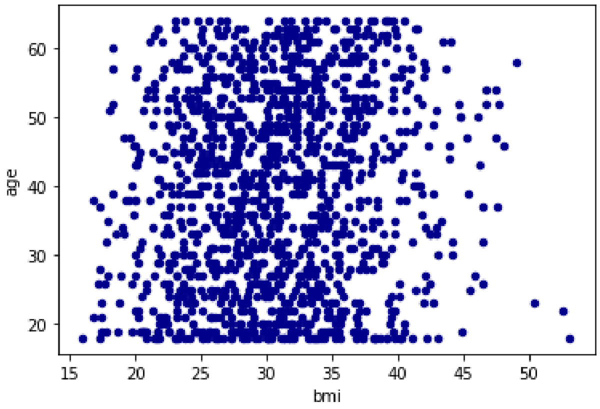
In [108]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="age",y="children", c='DarkBlue')
```



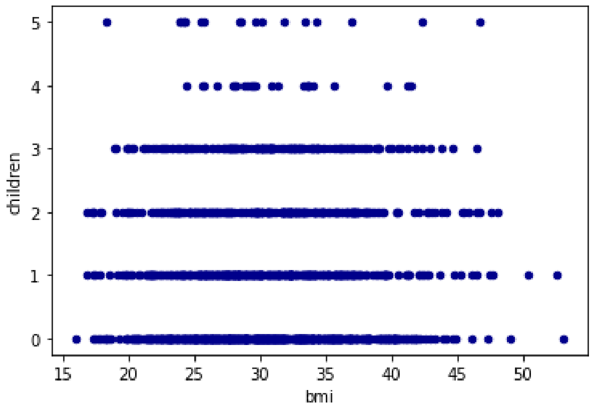
In [109]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="bmi",y="age", c='DarkBlue')
```



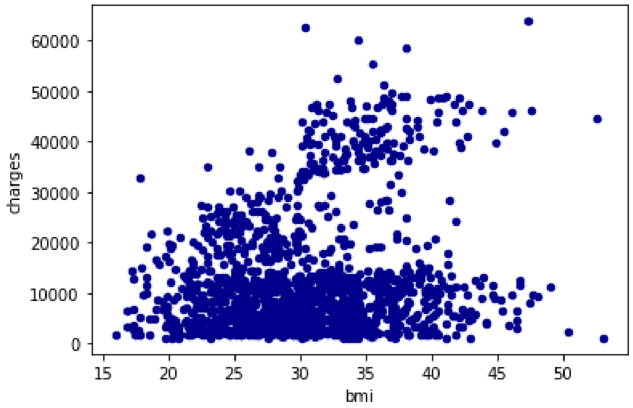
In [110]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="bmi",y="children", c='DarkBlue')
```



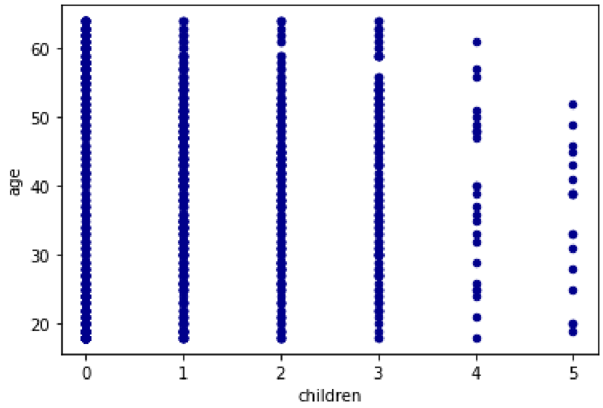
In [111]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="bmi",y="charges", c='DarkBlue')
```



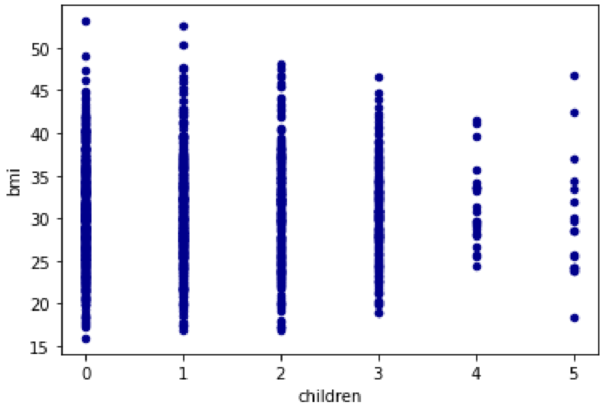
In [112]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="children",y="age", c='DarkBlue')
```



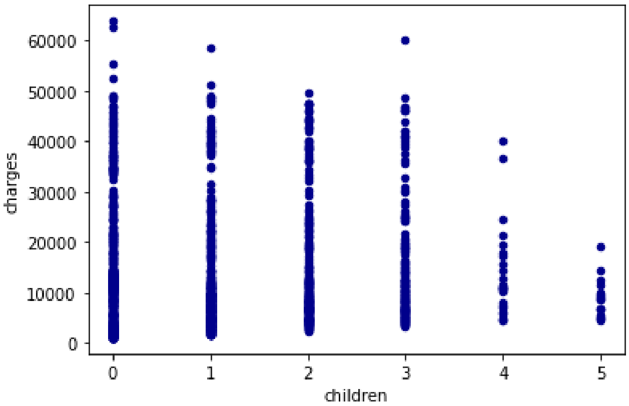
In [113]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="children",y="bmi", c='DarkBlue')
```



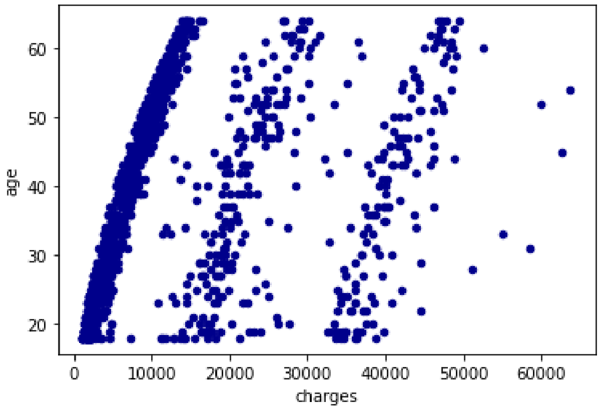
In [114]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="children",y="charges", c='DarkBlue')
```



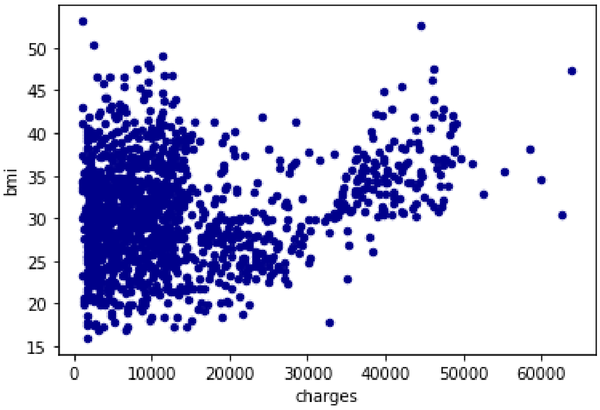
In [115]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="charges",y="age", c='DarkBlue')
```



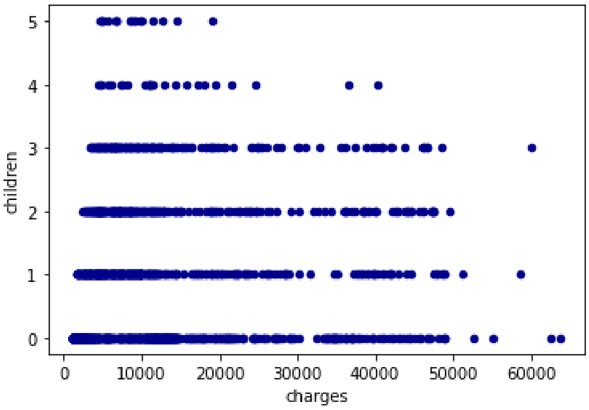
In [116]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="charges",y="bmi", c='DarkBlue')
```



In [117]:

```
ax1 = datas[["age", "bmi", "children", "charges"]].plot.scatter(x="charges",y="children", c='DarkBlue')
```



In [119]:

```
datas.shape
```

Out[119]:

```
(1338, 7)
```

In [198]:

```
datas.dtypes
```

Out[198]:

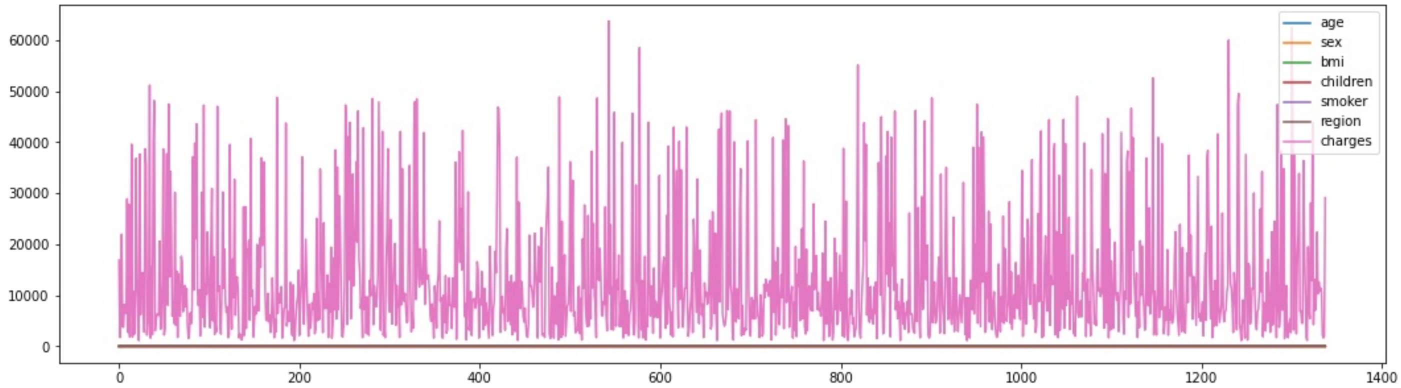
```
age          int64
sex          int64
bmi        float64
children     int64
smoker       int64
region       int64
charges    float64
dtype: object
```

In [121]:

```
datas.plot(figsize=(18,5))
```

Out[121]:

```
<AxesSubplot:>
```



In [122]:

```
datas.isnull().values.any()
```

Out[122]:

```
False
```

In [199]:

```
datanum= datas_lm.to_numpy()
```

In [200]:

```
np.set_printoptions(suppress=True)
print(datanum)
datanum.shape
```

```
[[   19.        0.       27.9    ...     1.        4.    16884.924 ]
 [   18.        1.       33.77   ...     0.        3.     1725.5523]
 [   28.        1.       33.     ...     0.        3.     4449.462 ]
 ...
 [   18.        0.       36.85   ...     0.        3.     1629.8335]
 [   21.        0.       25.8    ...     0.        4.     2007.945 ]
 [   61.        0.       29.07   ...     1.        2.    29141.3603]]
```

Out[200]:

```
(1338, 7)
```

In [201]:

```
print(datanum[:,-1:])
Y = datanum[:,-1:]
```

```
[[16884.924 ]
 [ 1725.5523]
 [ 4449.462 ]
 ...
 [ 1629.8335]
 [ 2007.945 ]
 [29141.3603]]
```

In [253]:

```
print(datanum[:,:-1])
X = datanum[:,:-1]
```

```
[[19.    0.   27.9   0.    1.    4.  ]
 [18.    1.   33.77  1.    0.    3.  ]
 [28.    1.   33.    3.    0.    3.  ]
 ...
 [18.    0.   36.85  0.    0.    3.  ]
 [21.    0.   25.8   0.    0.    4.  ]
 [61.    0.   29.07  0.    1.    2.  ]]
```

In [254]:

```
def AgregarCampo(datanum):
    location = datanum[:,-1:]

    lista = np.transpose(location).tolist()[0]
    regionnorthwest = list(map(lambda number:1 if(number == 2) else 0 , lista))
    regionsoutheast = list(map(lambda number:1 if(number == 3) else 0 , lista))
    regionsouthwest = list(map(lambda number:1 if(number == 4) else 0 , lista))

    regionnorthwest = np.array(regionnorthwest).reshape(len(regionnorthwest),1)
    regionsoutheast = np.array(regionsoutheast).reshape(len(regionnorthwest),1)
    regionsouthwest = np.array(regionsouthwest).reshape(len(regionnorthwest),1)
    return np.concatenate((datanum[:,:-1],regionnorthwest,regionsoutheast,regionsouthwest),1)

X = AgregarCampo(X)
print(X)
print(X.shape)
print(Y.shape)
```

```
[[19.    0.   27.9  ...  0.    0.    1.  ]
 [18.    1.   33.77 ...  0.    1.    0.  ]
 [28.    1.   33.   ...  0.    1.    0.  ]
 ...
 [18.    0.   36.85 ...  0.    1.    0.  ]
 [21.    0.   25.8  ...  0.    0.    1.  ]
 [61.    0.   29.07 ...  1.    0.    0.  ]]
(1338, 8)
(1338, 1)
```

In [252]:

```
print(X)
reg_mod = lm()
```

```
[[19.    0.   27.9  ...  0.    0.    1.  ]
 [18.    1.   33.77 ...  0.    1.    0.  ]
 [28.    1.   33.   ...  0.    1.    0.  ]
 ...
 [18.    0.   36.85 ...  0.    1.    0.  ]
 [21.    0.   25.8  ...  0.    0.    1.  ]
 [61.    0.   29.07 ...  1.    0.    0.  ]]
```

In [255]:

```
reg_mod.fit(X, Y)
```

Out[255]:

```
LinearRegression()
```

In [262]:

```
y_predict = reg_mod.predict(X)
```

In [263]:

```
reg_mod.coef_
```

Out[263]:

```
array([[  256.85635254,  -131.3143594 ,   339.19345361,   475.50054515,
        23848.53454191,  -352.96389942, -1035.02204939,  -960.0509913 ]])
```

In [266]:

```
rmse = mean_squared_error(Y, y_predict)
```

In [267]:

```
r2 = r2_score(Y, y_predict)
```

In [268]:

```
print('Slope:' ,reg_mod.coef_)
print('Intercept:', reg_mod.intercept_)
print('Root mean squared error: ', rmse)
print('R2 score: ', r2)
```

```
Slope: [[  256.85635254  -131.3143594    339.19345361   475.50054515
  23848.53454191  -352.96389942 -1035.02204939  -960.0509913 ]]
Intercept: [-11938.53857617]
Root mean squared error:  36501893.00741544
R2 score:  0.7509130345985207
```

In [ ]: