In [ ]:

```
```

Prepared By 크리스나

201873001

source helped From youtube, Kagle

==========

안녕하세요 교수님 kaggle 및 Youtube 보면서 했습니다

Machine Learning 대해서 아직 부족 하는것 같아요

앞으로 열심히 배울게요

Thank You Very Much

Have a great days ahead

In [27]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os
```

In [28]:

```
insurance_df=pd.read_csv("insurance.csv")  #To Load The Data
                                #인수레느스 insurance_df data frame 변수 에 저장
```

In [3]:

```
insurance_df.shape    # Size Of The Data,Row = 1338 and Column is 7
```

Out[3]:

```
(1338, 7)
```

In [29]:

```
insurance_df.head() #Head Info Of insurance_df
```

Out[29]:

	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

To chek NA data,Not availliable data is no there

In [5]:

```
insurance_df.isna().sum() #To chek NA data,Not availliable data is no there
```

Out[5]:

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

In [30]:

```
insurance_df.index  #To Find Index detail start fro 0 to 1338 gap is 1
```

Out[30]:

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

Import Metaplot for Graphical Representation

In [7]:

```
import matplotlib.pyplot as plot
%matplotlib inline
import seaborn as sns
```

Most important and UseFul information of Data Insurance

In [31]:

```
insurance_df.describe()
```

Out[31]:

	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

To find the correlation among

The columns using pearson method

In [32]:

```
insur_corr=insurance_df.corr()
insur_corr
```

Out[32]:

	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

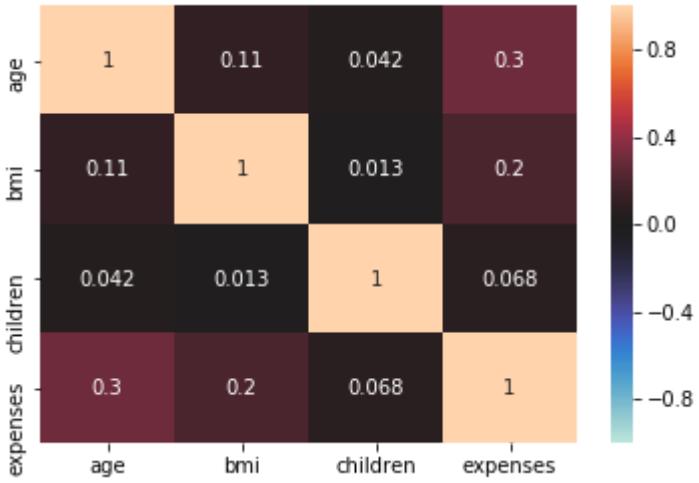
Heatmap of Correlation Data of Insurance

In [33]:

```
s.heatmap(insur_corr,vmin=-1,vmax=1,center=0,annot=True)    #Heatmap of Correlation Data of Insurance
```

Out[33]:

<matplotlib.axes._subplots.AxesSubplot at 0x115ad240>



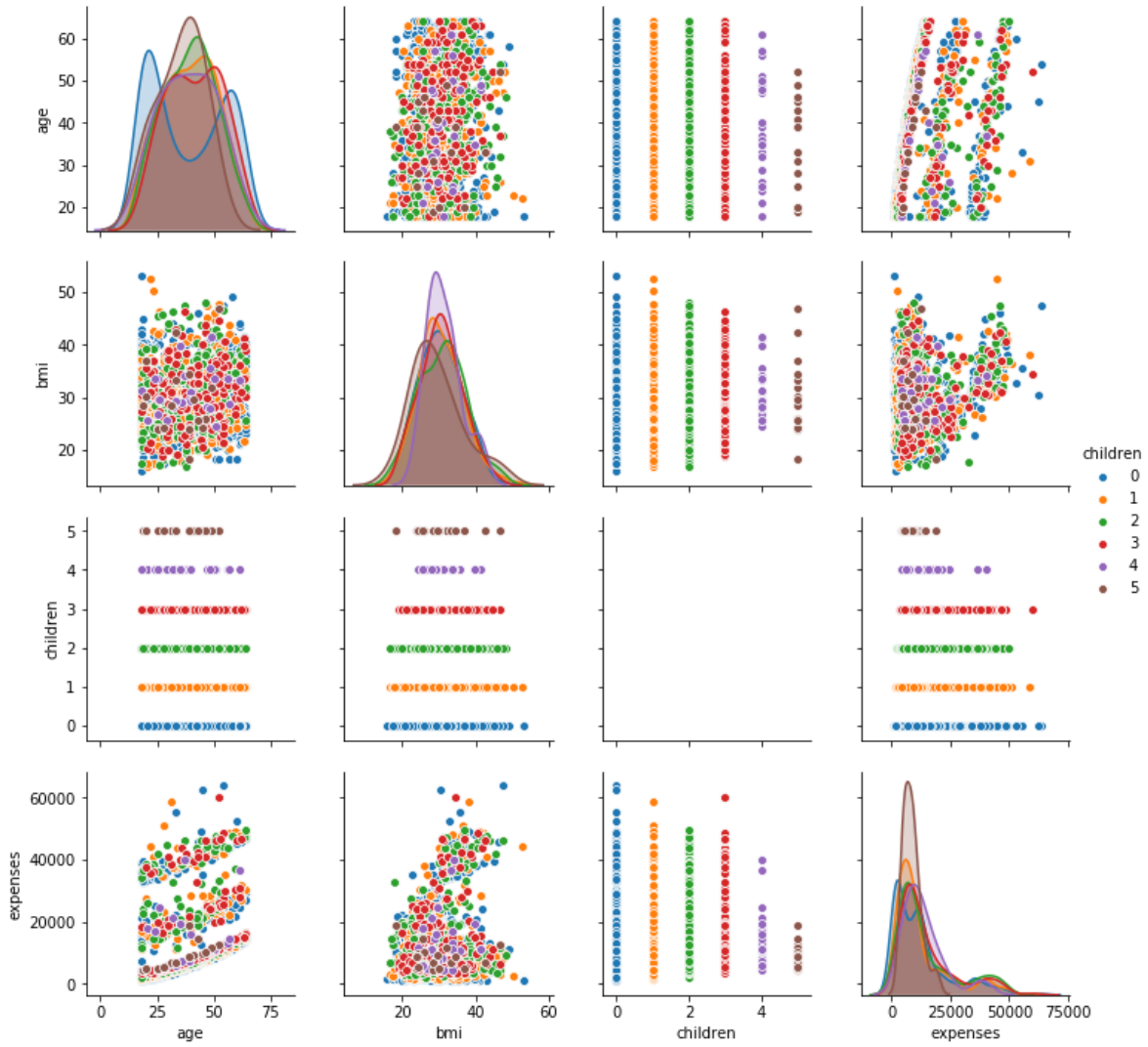
Various Diagrams of children relation

In [36]:

```
sns.pairplot(data=insurance_df,hue='children')
```

Out[36]:

<seaborn.axisgrid.PairGrid at 0x12d93eb8>



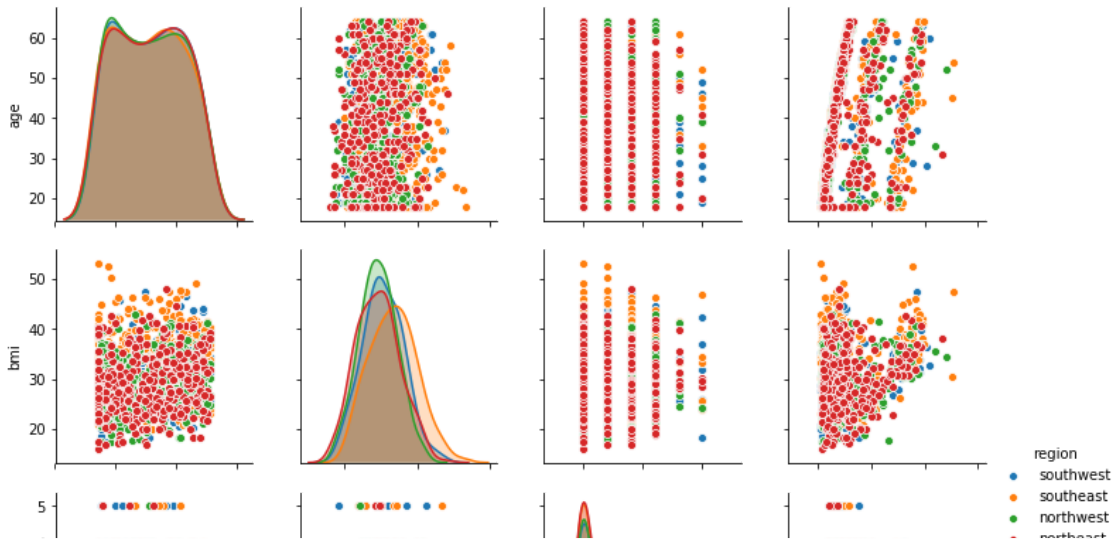
Various Relation with Region

In [12]:

```
sns.pairplot(data=insurance_df,hue='region')
```

Out[12]:

<seaborn.axisgrid.PairGrid at 0xca39128>



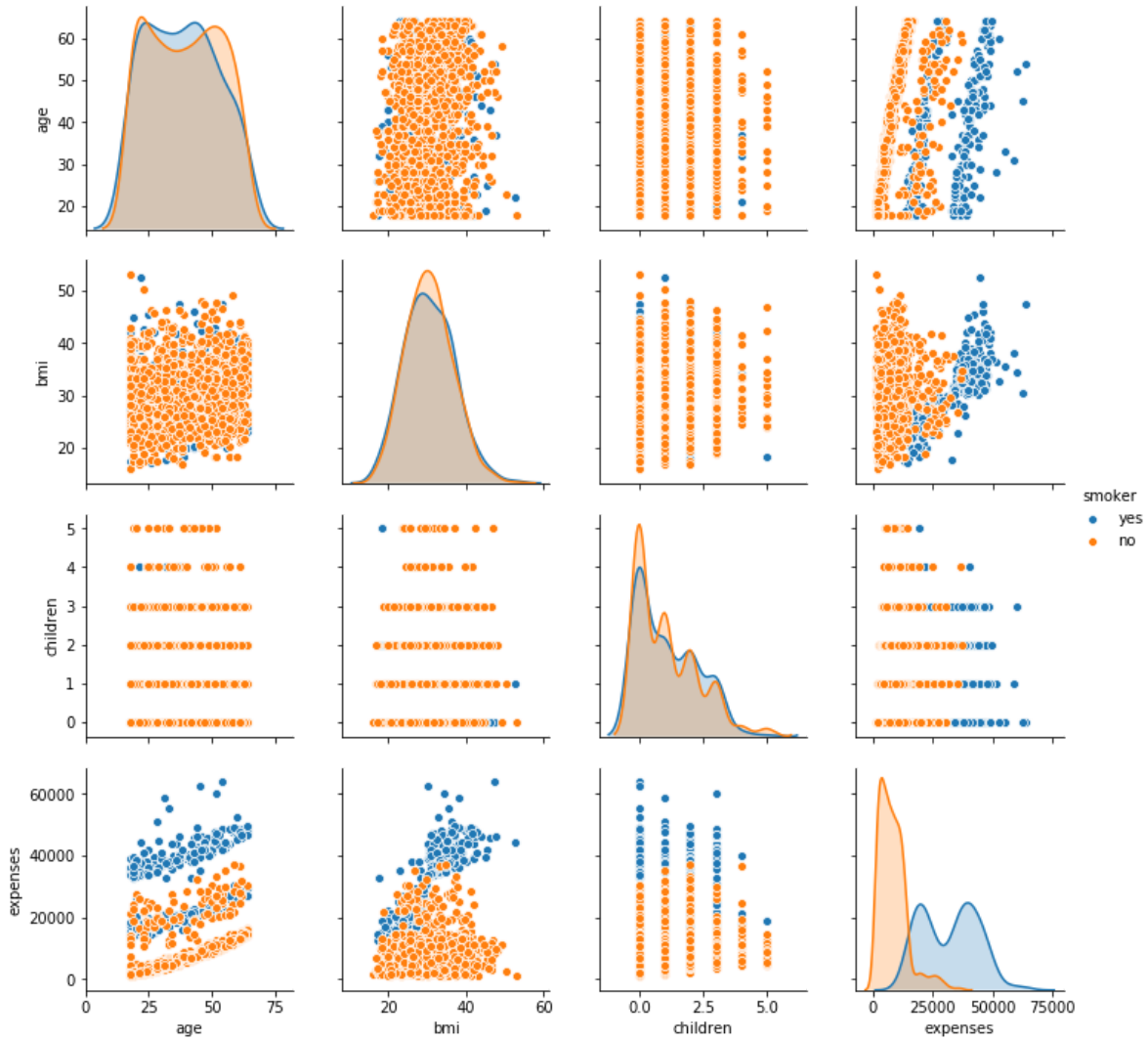
Relation with smoker

In [13]:

```
sns.pairplot(data=insurance_df,hue='smoker')
```

Out[13]:

<seaborn.axisgrid.PairGrid at 0xdfee1d0>



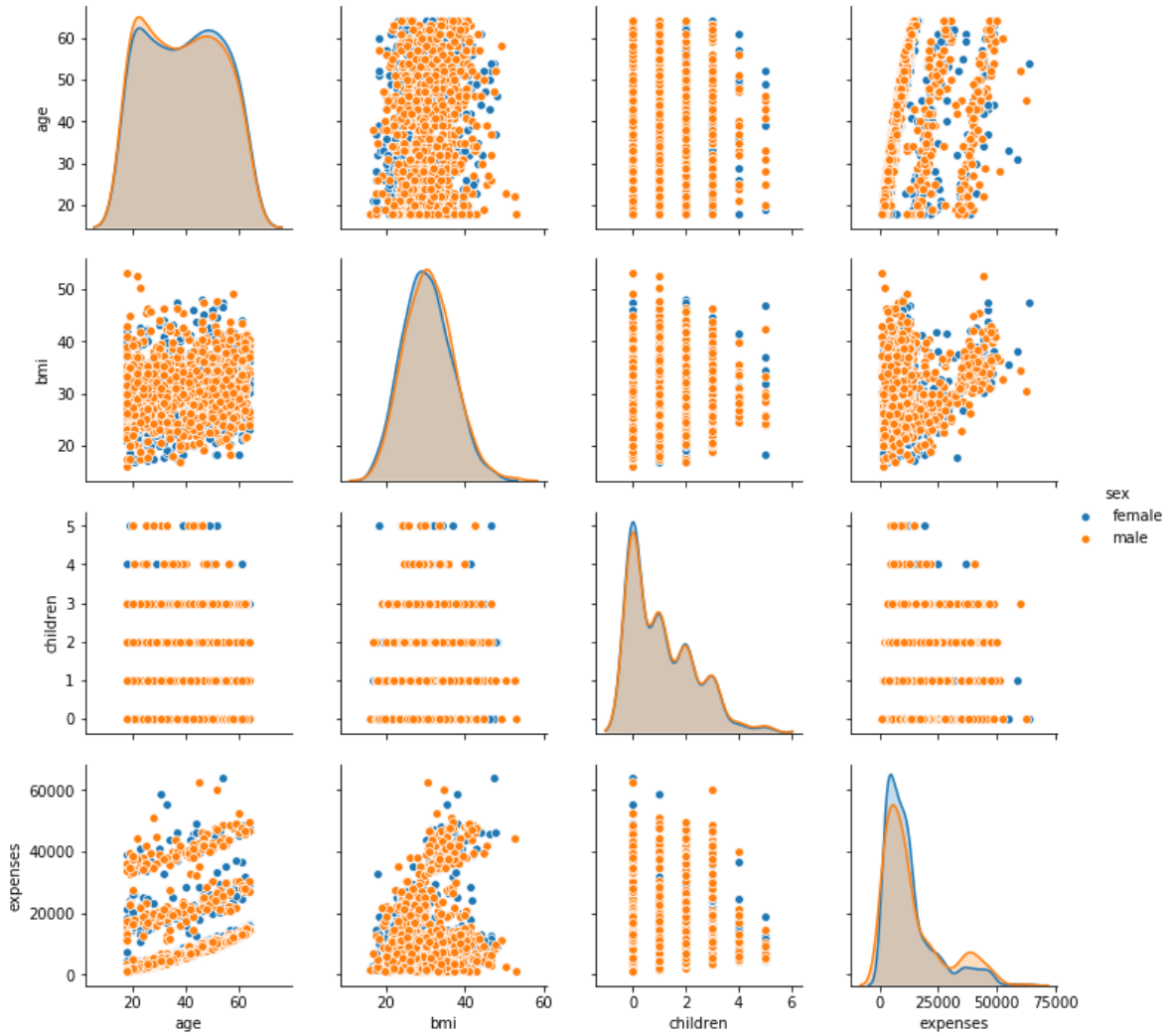
Relation with sex

In [14]:

```
sns.pairplot(data=insurance_df,hue='sex')
```

Out[14]:

<seaborn.axisgrid.PairGrid at 0xe72e550>



In [38]:

```
cat_col=['smoker','region','sex']
num_col=[i for i in insurance_df.columns if i not in cat_col]
#all except smoker region & sex
num_col
```

Out[38]:

```
['age', 'bmi', 'children', 'expenses']
```

In [39]:

```
# one-hot encoding
one_hot=pd.get_dummies(insurance_df[cat_col])
insur_procsd_df=pd.concat([insurance_df[num_col],one_hot],axis=1)
insur_procsd_df.head(10)
```

Out[39]:

	age	bmi	children	expenses	smoker_no	smoker_yes	region_northeast	region_northwest
0	19	27.9	0	16884.92	0	1	0	0
1	18	33.8	1	1725.55	1	0	0	0
2	28	33.0	3	4449.46	1	0	0	0
3	33	22.7	0	21984.47	1	0	0	1
4	32	28.9	0	3866.86	1	0	0	1
5	31	25.7	0	3756.62	1	0	0	0
6	46	33.4	1	8240.59	1	0	0	0
7	37	27.7	3	7281.51	1	0	0	1
8	37	29.8	2	6406.41	1	0	1	0
9	60	25.8	0	28923.14	1	0	0	1

In [40]:

```
#label encoding
insr_procsd_df_label=insurance_df
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for i in cat_col:
    insr_procsd_df_label[i] = label_encoder.fit_transform(insr_procsd_df_label[i])
insr_procsd_df_label.head(10)
```

Out[40]:

	age	sex	bmi	children	smoker	region	expenses
0	19	0	27.9	0	1	3	16884.92
1	18	1	33.8	1	0	2	1725.55
2	28	1	33.0	3	0	2	4449.46
3	33	1	22.7	0	0	1	21984.47
4	32	1	28.9	0	0	1	3866.86
5	31	0	25.7	0	0	2	3756.62
6	46	0	33.4	1	0	2	8240.59
7	37	0	27.7	3	0	1	7281.51
8	37	1	29.8	2	0	0	6406.41
9	60	0	25.8	0	0	1	28923.14

In [41]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error,r2_score
```

In [42]:

```
X=insur_procsd_df.drop(columns='expenses')
y=insurance_df[['expenses']]
```

In [43]:

```
train_X, test_X, train_y, test_y = train_test_split(X,y,test_size=0.3,random_state=1234)
```

In [44]:

```
model = LinearRegression()

model.fit(train_X,train_y)
```

Out[44]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [45]:

```
model.intercept_
```

Out[45]:

```
array([724.25311371])
```

In [46]:

```
model.coef_
```

Out[46]:

```
array([[   248.4673171 ,    319.2767806 ,    421.13556486,
        -12335.81112837,  12335.81112837,    680.16937687,
           -61.90676651,   -383.06551251,   -235.19709785,
           252.32908757,   -252.32908757]])
```

In [47]:

```
cdf = pd.DataFrame(data=model.coef_.T, index=X.columns, columns=["Coefficients"])
cdf
```

Out[47]:

	Coefficients
age	248.467317
bmi	319.276781
children	421.135565
smoker_no	-12335.811128
smoker_yes	12335.811128
region_northeast	680.169377
region_northwest	-61.906767
region_southeast	-383.065513
region_southwest	-235.197098
sex_female	252.329088
sex_male	-252.329088

In [48]:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error,r2_score

print("The train 데이터 예측")
train_predict = model.predict(train_X)
print("The test 데이터 예측")
test_predict = model.predict(test_X)
print("MAE")
print("Train : ",mean_absolute_error(train_y,train_predict))
print("Test  : ",mean_absolute_error(test_y,test_predict))
print("====================================")
print("MSE")
print("Train : ",mean_squared_error(train_y,train_predict))
print("Test  : ",mean_squared_error(test_y,test_predict))
print("====================================")
import numpy as np
print("RMSE")
print("Train : ",np.sqrt(mean_squared_error(train_y,train_predict)))
print("Test  : ",np.sqrt(mean_squared_error(test_y,test_predict)))
print("====================================")
print("R^2")
print("Train : ",r2_score(train_y,train_predict))
print("Test  : ",r2_score(test_y,test_predict))
```

```
The train 데이터 예측
The test 데이터 예측
MAE
Train :  4094.4715056018945
Test  :  4190.452194935136
====================================
MSE
Train :  36502855.45666693
Test  :  37242969.996799946
====================================
RMSE
Train :  6041.759301450773
Test  :  6102.701860389376
====================================
R^2
Train :  0.7544304473973156
Test  :  0.7369995351189047
```

In [56]:

```
plot.figure(figsize=(10,7))
plot.title("Actual vs. predicted expenses",fontsize=25)
plot.xlabel("Actual expenses",fontsize=18,color = "green")
plot.ylabel("Predicted expenses", fontsize=18,color = "red")
plot.scatter(x=test_y,y=test_predict,color= "green")
# How to show different colar
```

Out[56]:

```
<matplotlib.collections.PathCollection at 0x1547d7b8>
```

In [ ]: