Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

| **import** pandas **as** pd *#Data manipulation*  **import** numpy **as** np *#Data manipulation*  **import** matplotlib.pyplot **as** plt *# Visualization*  **import** seaborn **as** sns *#Visualization*  plt**.**rcParams['figure.figsize'] **=** [8,5]  plt**.**rcParams['font.size'] **=**14  plt**.**rcParams['font.weight']**=** 'bold'  plt**.**style**.**use('seaborn-whitegrid') |
| --- |

In [1]: In [3]:

Out[3]:

| *# Import dataset*  *#path ='dataset/'*  path **=** '../input/'  df **=** pd**.**read\_csv("dataset\_3\_insurance.csv")  print('\nNumber of rows and columns in the data set: ',df**.**shape)  print('')  df**.**head() |
| --- |

Number of rows and columns in the data set: (1338, 7)

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

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 1 of 18

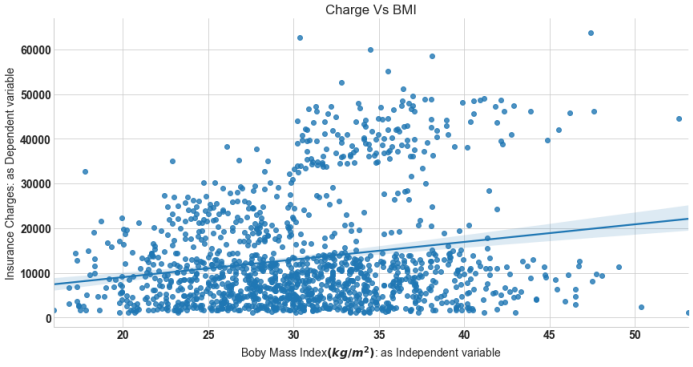
Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

In [4]:

for our visualization purpose will fit line using seaborn library only for bmi as independent variable and charges as dependent variable

| sns**.**lmplot(x**=**'bmi',y**=**'charges',data**=**df,aspect**=**2,height**=**6)  plt**.**xlabel('Boby Mass Index$(kg/m^2)$: as Independent variable')  plt**.**ylabel('Insurance Charges: as Dependent variable')  plt**.**title('Charge Vs BMI'); |
| --- |

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 2 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| df**.**describe() |
| --- |

In [5]:

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 3 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

Out[5]:

In [9]: In [11]:

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

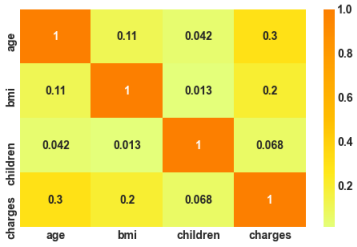
max 64.000000 53.130000 5.000000 63770.428010

| plt**.**figure(figsize**=**(12,2))  sns**.**heatmap(df**.**isnull(),cbar**=False**,cmap**=**'viridis',yticklabels**=False**)  plt**.**title('Missing value in the dataset'); |
| --- |



| *# correlation plot*  corr **=** df**.**corr()  sns**.**heatmap(corr, cmap **=** 'Wistia', annot**= True**); |
| --- |

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 4 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| f**=** plt**.**figure(figsize**=**(12,4))  ax**=**f**.**add\_subplot(121)  sns**.**distplot(df['charges'],bins**=**50,color**=**'r',ax**=**ax)  ax**.**set\_title('Distribution of insurance charges')  ax**=**f**.**add\_subplot(122)  sns**.**distplot(np**.**log10(df['charges']),bins**=**40,color**=**'b',ax**=**ax)  ax**.**set\_title('Distribution of insurance charges in $log$ sacle')  ax**.**set\_xscale('log'); |
| --- |

In [12]:

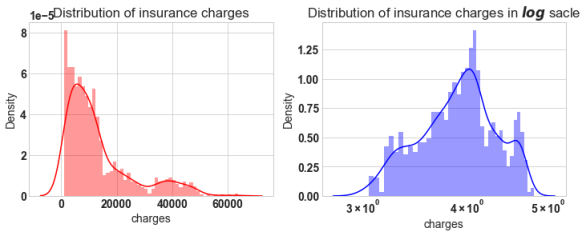
http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 5 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

In [13]:

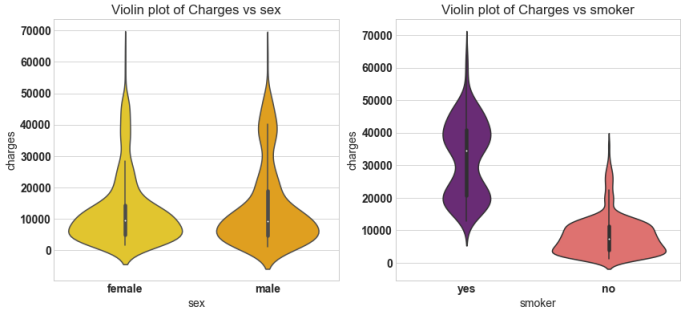
/Users/dr.renukar.patil/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2557: FutureWarning: `d istplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `d isplot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms). warnings.warn(msg, FutureWarning)

/Users/dr.renukar.patil/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2557: FutureWarning: `d istplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `d isplot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms). warnings.warn(msg, FutureWarning)



| f **=** plt**.**figure(figsize**=**(14,6))  ax **=** f**.**add\_subplot(121)  sns**.**violinplot(x**=**'sex', y**=**'charges',data**=**df,palette**=**'Wistia',ax**=**ax)  ax**.**set\_title('Violin plot of Charges vs sex')  ax **=** f**.**add\_subplot(122)  sns**.**violinplot(x**=**'smoker', y**=**'charges',data**=**df,palette**=**'magma',ax**=**ax)  ax**.**set\_title('Violin plot of Charges vs smoker'); |
| --- |

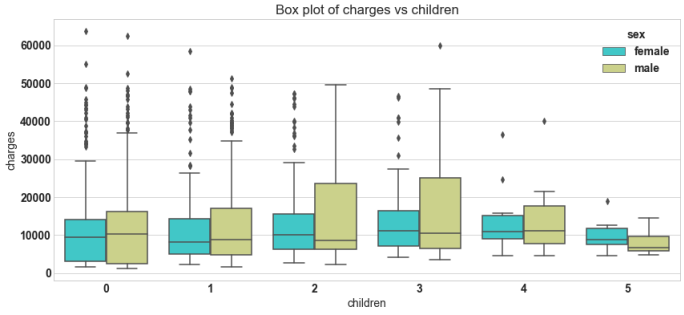
http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 6 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| plt**.**figure(figsize**=**(14,6))  sns**.**boxplot(x**=**'children', y**=**'charges',hue**=**'sex',data**=**df,palette**=**'rainbow')  plt**.**title('Box plot of charges vs children'); |
| --- |

In [14]:

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 7 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| df**.**groupby('children')**.**agg(['mean','min','max'])['charges'] |
| --- |

In [15]:

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 8 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

Out[15]: In [16]:

mean min max

children

0 12365.975602 1121.8739 63770.42801

1 12731.171832 1711.0268 58571.07448

2 15073.563734 2304.0022 49577.66240

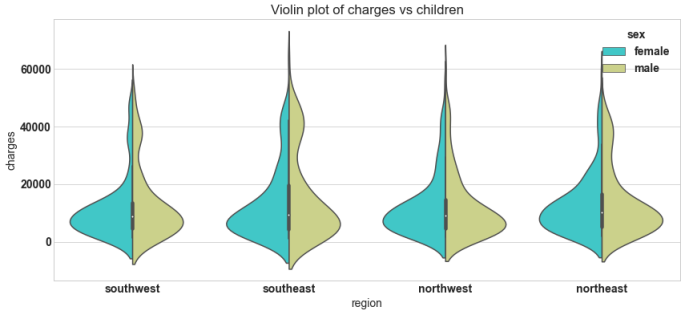
3 15355.318367 3443.0640 60021.39897

4 13850.656311 4504.6624 40182.24600

5 8786.035247 4687.7970 19023.26000

| plt**.**figure(figsize**=**(14,6))  sns**.**violinplot(x**=**'region', y**=**'charges',hue**=**'sex',data**=**df,palette**=**'rainbow',split**=True**)  plt**.**title('Violin plot of charges vs children'); |
| --- |

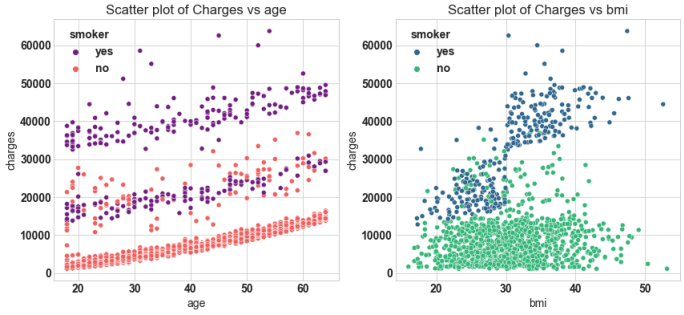
http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 9 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| f **=** plt**.**figure(figsize**=**(14,6))  ax **=** f**.**add\_subplot(121)  sns**.**scatterplot(x**=**'age',y**=**'charges',data**=**df,palette**=**'magma',hue**=**'smoker',ax**=**ax)  ax**.**set\_title('Scatter plot of Charges vs age')  ax **=** f**.**add\_subplot(122)  sns**.**scatterplot(x**=**'bmi',y**=**'charges',data**=**df,palette**=**'viridis',hue**=**'smoker')  ax**.**set\_title('Scatter plot of Charges vs bmi')  plt**.**savefig('sc.png'); |
| --- |

In [17]:

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 10 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| *# Dummy variable*  categorical\_columns **=** ['sex','children', 'smoker', 'region']  df\_encode **=** pd**.**get\_dummies(data **=** df, prefix **=** 'OHE', prefix\_sep**=**'\_',  columns **=** categorical\_columns,  drop\_first **=True**,  dtype**=**'int8') |
| --- |

In [18]:

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 11 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

| *# Lets verify the dummay variable process*  print('Columns in original data frame:\n',df**.**columns**.**values)  print('\nNumber of rows and columns in the dataset:',df**.**shape)  print('\nColumns in data frame after encoding dummy variable:\n',df\_encode**.**columns**.**values) print('\nNumber of rows and columns in the dataset:',df\_encode**.**shape) |
| --- |

In [19]:

In [20]:

Out[20]: In [21]:

Columns in original data frame:

['age' 'sex' 'bmi' 'children' 'smoker' 'region' 'charges']

Number of rows and columns in the dataset: (1338, 7)

Columns in data frame after encoding dummy variable:

['age' 'bmi' 'charges' 'OHE\_male' 'OHE\_1' 'OHE\_2' 'OHE\_3' 'OHE\_4' 'OHE\_5'

'OHE\_yes' 'OHE\_northwest' 'OHE\_southeast' 'OHE\_southwest']

Number of rows and columns in the dataset: (1338, 13)

| **from** scipy.stats **import** boxcox  y\_bc,lam, ci**=** boxcox(df\_encode['charges'],alpha**=**0.05)  *#df['charges'] = y\_bc*  *# it did not perform better for this model, so log transform is used*  ci,lam |
| --- |

((-0.01140290617294196, 0.0988096859767545), 0.043649053770664956)

| *## Log transform*  df\_encode['charges'] **=** np**.**log(df\_encode['charges']) |
| --- |

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 12 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

| **from** sklearn.model\_selection **import** train\_test\_split  X **=** df\_encode**.**drop('charges',axis**=**1) *# Independet variable*  y **=** df\_encode['charges'] *# dependent variable*  X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X,y,test\_size**=**0.3,random\_state**=**23) |
| --- |

In [22]: In [23]:

In [24]: In [26]:

| *# Step 1: add x0 =1 to dataset*  X\_train\_0 **=** np**.**c\_[np**.**ones((X\_train**.**shape[0],1)),X\_train]  X\_test\_0 **=** np**.**c\_[np**.**ones((X\_test**.**shape[0],1)),X\_test]  *# Step2: build model*  theta **=** np**.**matmul(np**.**linalg**.**inv( np**.**matmul(X\_train\_0**.**T,X\_train\_0) ), np**.**matmul(X\_train\_0**.**T,y\_train)) |
| --- |

| *# The parameters for linear regression model*  parameter **=** ['theta\_'**+**str(i) **for** i **in** range(X\_train\_0**.**shape[1])]  columns **=** ['intersect:x\_0=1'] **+** list(X**.**columns**.**values)  parameter\_df **=** pd**.**DataFrame({'Parameter':parameter,'Columns':columns,'theta':theta}) |
| --- |

| *# Scikit Learn module*  **from** sklearn.linear\_model **import** LinearRegression  lin\_reg **=** LinearRegression()  lin\_reg**.**fit(X\_train,y\_train) *# Note: x\_0 =1 is no need to add, sklearn will take care of it. #Parameter*  sk\_theta **=** [lin\_reg**.**intercept\_]**+**list(lin\_reg**.**coef\_)  parameter\_df **=** parameter\_df**.**join(pd**.**Series(sk\_theta, name**=**'Sklearn\_theta'))  parameter\_df |
| --- |

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 13 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

Out[26]:

Parameter Columns theta Sklearn\_theta 0 theta\_0 intersect:x\_0=1 7.059171 7.059171 1 theta\_1 age 0.033134 0.033134 2 theta\_2 bmi 0.013517 0.013517 3 theta\_3 OHE\_male -0.067767 -0.067767 4 theta\_4 OHE\_1 0.149457 0.149457 5 theta\_5 OHE\_2 0.272919 0.272919 6 theta\_6 OHE\_3 0.244095 0.244095 7 theta\_7 OHE\_4 0.523339 0.523339 8 theta\_8 OHE\_5 0.466030 0.466030 9 theta\_9 OHE\_yes 1.550481 1.550481

10 theta\_10 OHE\_northwest -0.055845 -0.055845 11 theta\_11 OHE\_southeast -0.146578 -0.146578 12 theta\_12 OHE\_southwest -0.133508 -0.133508

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 14 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

| *# Normal equation*  y\_pred\_norm **=** np**.**matmul(X\_test\_0,theta)  *#Evaluvation: MSE*  J\_mse **=** np**.**sum((y\_pred\_norm **-** y\_test)**\*\***2)**/** X\_test\_0**.**shape[0]  *# R\_square*  sse **=** np**.**sum((y\_pred\_norm **-** y\_test)**\*\***2)  sst **=** np**.**sum((y\_test **-** y\_test**.**mean())**\*\***2)  R\_square **=** 1 **-** (sse**/**sst)  print('The Mean Square Error(MSE) or J(theta) is: ',J\_mse)  print('R square obtain for normal equation method is :',R\_square) |
| --- |

In [27]: In [28]:

The Mean Square Error(MSE) or J(theta) is: 0.18729622322981862

R square obtain for normal equation method is : 0.7795687545055323

| *# sklearn regression module*  y\_pred\_sk **=** lin\_reg**.**predict(X\_test)  *#Evaluvation: MSE*  **from** sklearn.metrics **import** mean\_squared\_error  J\_mse\_sk **=** mean\_squared\_error(y\_pred\_sk, y\_test)  *# R\_square*  R\_square\_sk **=** lin\_reg**.**score(X\_test,y\_test)  print('The Mean Square Error(MSE) or J(theta) is: ',J\_mse\_sk)  print('R square obtain for scikit learn library is :',R\_square\_sk) |
| --- |

The Mean Square Error(MSE) or J(theta) is: 0.18729622322981884

R square obtain for scikit learn library is : 0.779568754505532

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 15 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM

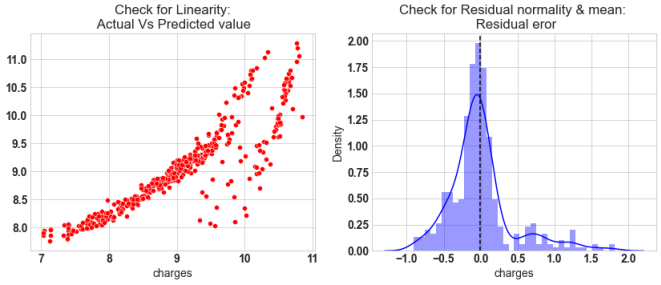
| *# Check for Linearity*  f **=** plt**.**figure(figsize**=**(14,5))  ax **=** f**.**add\_subplot(121)  sns**.**scatterplot(y\_test,y\_pred\_sk,ax**=**ax,color**=**'r')  ax**.**set\_title('Check for Linearity:\n Actual Vs Predicted value')  *# Check for Residual normality & mean*  ax **=** f**.**add\_subplot(122)  sns**.**distplot((y\_test **-** y\_pred\_sk),ax**=**ax,color**=**'b')  ax**.**axvline((y\_test **-** y\_pred\_sk)**.**mean(),color**=**'k',linestyle**=**'--')  ax**.**set\_title('Check for Residual normality & mean: \n Residual eror'); |
| --- |

In [29]:

/Users/dr.renukar.patil/opt/anaconda3/lib/python3.8/site-packages/seaborn/\_decorators.py:36: FutureWarning: Pass t he following variables as keyword args: x, 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(

/Users/dr.renukar.patil/opt/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2557: FutureWarning: `d istplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `d isplot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms). warnings.warn(msg, FutureWarning)

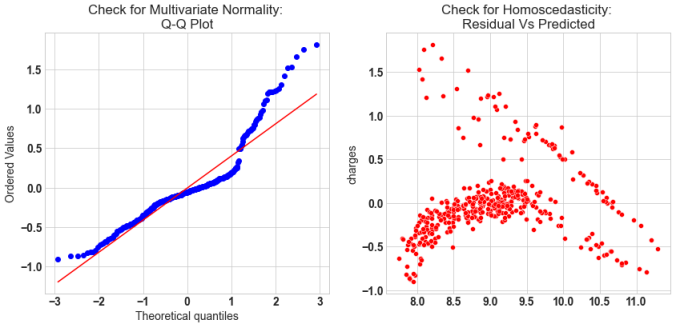
http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 16 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| *# Check for Multivariate Normality*  *# Quantile-Quantile plot*  f,ax **=** plt**.**subplots(1,2,figsize**=**(14,6))  **import** scipy **as** sp  \_,(\_,\_,r)**=** sp**.**stats**.**probplot((y\_test **-** y\_pred\_sk),fit**=True**,plot**=**ax[0])  ax[0]**.**set\_title('Check for Multivariate Normality: \nQ-Q Plot')  *#Check for Homoscedasticity*  sns**.**scatterplot(y **=** (y\_test **-** y\_pred\_sk), x**=** y\_pred\_sk, ax **=** ax[1],color**=**'r')  ax[1]**.**set\_title('Check for Homoscedasticity: \nResidual Vs Predicted'); |
| --- |

In [30]:

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 17 of 18

Lab\_3\_Linear\_Regression 24/01/22, 10:35 AM 

| *# Check for Multicollinearity*  *#Variance Inflation Factor*  VIF **=** 1**/**(1**-** R\_square\_sk)  VIF |
| --- |

In [31]: Out[31]:

4.5365619459111395

http://localhost:8888/nbconvert/html/6th\_ML\_LAB/Lab\_3\_Linear\_Regression.ipynb?download=false Page 18 of 18