Adjusted R-squared - Exercise Solution

Using the code from the lecture, create a function which will calculate the adjusted R-squared for you, given the independent variable(s) (x) and the dependent variable (y).

Check if you function is working properly.

Solution at the bottom.

Import the relevant libraries

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

from sklearn.linear_model import LinearRegression
```

Load the data

In [2]:
```
data = pd.read_csv('1.02. Multiple linear regression.csv')
data.head()
```

Out[2]:

	SAT	Rand 1,2,3	GPA
0	1714	1	2.40
1	1664	3	2.52
2	1760	3	2.54
3	1685	3	2.74
4	1693	2	2.83

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

Out[3]:

	SAT	Rand 1,2,3	GPA
count	84.000000	84.000000	84.000000
mean	1845.273810	2.059524	3.330238
std	104.530661	0.855192	0.271617
min	1634.000000	1.000000	2.400000
25%	1772.000000	1.000000	3.190000
50%	1846.000000	2.000000	3.380000
75%	1934.000000	3.000000	3.502500
max	2050.000000	3.000000	3.810000

Create the multiple linear regression

Declare the dependent and independent variables

In [4]:
```
x = data[['SAT','Rand 1,2,3']]
y = data['GPA']
```

Regression itself

In [5]:
```
reg = LinearRegression()
reg.fit(x,y)
```

Out[5]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [6]:
```
reg.coef_
```

Out[6]:
```
array([ 0.00165354, -0.00826982])
```

In [7]:
```
reg.intercept_
```

Out[7]:
```
0.29603261264909486
```

Calculating the R-squared

In [8]:
```
reg.score(x,y)
```

Out[8]:
```
0.4066811952814285
```

Formula for Adjusted R^2

$R^2_{adj.} = 1 - (1-R^2)*\frac{n-1}{n-p-1}$

In [9]:
```
x.shape
```

Out[9]:
```
(84, 2)
```

In [10]:
```
r2 = reg.score(x,y)
n = x.shape[0]
p = x.shape[1]

adjusted_r2 = 1-(1-r2)*(n-1)/(n-p-1)
adjusted_r2
```

Out[10]:
```
0.39203134825134023
```

Adjusted R^2 function

In [11]:
```
# There are different ways to solve this problem
# To make it as easy and interpretable as possible, we have preserved the or
def adj_r2(x,y):
    r2 = reg.score(x,y)
    n = x.shape[0]
    p = x.shape[1]
    adjusted_r2 = 1-(1-r2)*(n-1)/(n-p-1)
    return adjusted_r2
```

In [12]:
```
# Here's the result
adj_r2(x,y)
```

Out[12]: 0.39203134825134023