



2. Aprendizado Supervisionado e Regressão Linear

Após fazer os exercícios deste laboratório responda ao **questionário correspondente da aula no Moodle**.

Caso: Estimando a emissão de gases CO2 de veículos

Neste Lab você vai empregar modelos de regressão simples e múltipla para estimar as emissões de CO2 de veículos a partir de suas características como consumo de combustível, marca ou tamanho do motor.

Dados: <https://meusite.mackenzie.br/rogerio/TIC/FuelConsumptionCo2.csv>

Exercício. Acesse e Explore os dados.

Acesse e explore os dados antes de contruir os seus modelos. Verifique as quantidades e tipos de dados envolvidos, a qualidade dos dados etc. é fundamental conhecer os dados antes de se construir modelos sobre eles.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
df = pd.read_csv("https://meusite.mackenzie.br/rogerio/TIC/FuelConsumptionCo2.csv")
df.head()
```

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINE SIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUM
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	

```
# seu código
df.shape
df.dtypes
df.describe(include='all')
df.isnull().sum()
```

```
MODELYEAR      0
MAKE            0
MODEL           0
VEHICLECLASS    0
ENGINE SIZE     0
CYLINDERS       0
TRANSMISSION    0
FUELTYPE        0
FUELCONSUMPTION_CITY  0
FUELCONSUMPTION_Hwy  0
FUELCONSUMPTION_COMB  0
FUELCONSUMPTION_COMB_MPG  0
CO2EMISSIONS    0
dtype: int64
```

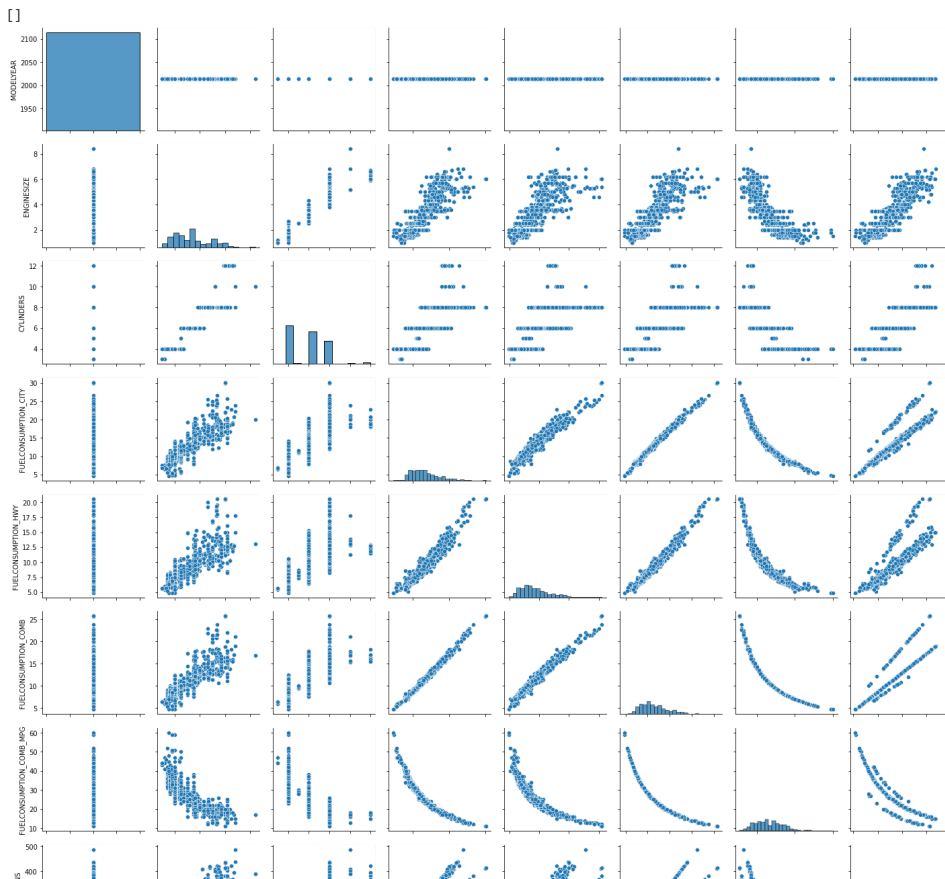
Exercício. Faça um gráfico de dispersão entre todos os pares de variáveis

Isso irá permitir você visualizar as relações de cada par de variáveis dos dados.

Dica: Empregue `sns.pairplot(df)`

```
# seu código

sns.pairplot(df)
plt.plot()
```



Exercício. Modelo Regressão Simples

Crie um modelo de regressão simples para estimar valores `CO2EMISSIONS` com base nos dados de consumo combinado dos veículos `FUELCONSUMPTION_COMB`. Encontre os coeficientes, seus p-values, e o R^2 do modelo.

```
# seu código
import statsmodels.formula.api as sm

model = sm.ols(formula='CO2EMISSIONS ~ FUELCONSUMPTION_COMB', data=df)
result = model.fit()
print(result.summary())
```

/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
import pandas.util.testing as tm

OLS Regression Results

Dep. Variable:	CO2EMISSIONS	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.796
Method:	Least Squares	F-statistic:	4153.
Date:	Wed, 09 Mar 2022	Prob (F-statistic):	0.00
Time:	18:31:22	Log-Likelihood:	-5092.7
No. Observations:	1067	AIC:	1.019e+04
Df Residuals:	1065	BIC:	1.020e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
Intercept	68.3871	3.044	22.467	0.000	62.414	74.360
FUELCONSUMPTION_COMB	16.2200	0.252	64.443	0.000	15.726	16.714

Omnibus:	152.161	Durbin-Watson:	2.195
Prob(Omnibus):	0.000	Jarque-Bera (JB):	240.073
Skew:	-0.954	Prob(JB):	7.39e-53
Kurtosis:	4.325	Cond. No.	42.2

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Exercício. Predição

A partir do seu modelo empregue a função `result.predict(x)` para estimar a emissão de gases por veículos que apresentam consumo de combustível com valores 4 e 28.

```
# seu código

X_novo = pd.DataFrame()
X_novo['FUELCONSUMPTION_COMB'] = [4, 28]
result.predict(X_novo)
```

```
0    133.267015
1    522.546301
dtype: float64
```

Exercício. Regressão Múltipla

Faça agora um modelo de regressão múltipla para estimar as emissões de CO2 a partir de `FUELCONSUMPTION_COMB` e `ENGINE_SIZE`. Em seguida faça a predição de emissões para um veículo com `FUELCONSUMPTION_COMB = 10` e `ENGINE_SIZE = 2`.

```
# seu código

# define o modelo
model = sm.ols(formula='CO2EMISSIONS ~ FUELCONSUMPTION_COMB + ENGINE_SIZE', data=df)

# calcula o modelo e mostra os resultados
result = model.fit()
print(result.summary())

# faz a previsão
X_novo = pd.DataFrame()
X_novo['FUELCONSUMPTION_COMB'] = [10]
X_novo['ENGINE_SIZE'] = [2]

print(result.predict(X_novo))
```

```

=====
                        OLS Regression Results
=====
Dep. Variable:          CO2EMISSIONS      R-squared:                0.858
Model:                  OLS              Adj. R-squared:            0.858
Method:                 Least Squares    F-statistic:                3220.
Date:                   Wed, 09 Mar 2022  Prob (F-statistic):          0.00
Time:                   18:31:22         Log-Likelihood:           -4898.4
No. Observations:       1067            AIC:                       9803.
Df Residuals:           1064            BIC:                       9818.
Df Model:                2
Covariance Type:        nonrobust
=====
                        coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept              78.3068      2.579      30.360      0.000      73.246      83.368
FUELCONSUMPTION_COMB    9.7300      0.366      26.569      0.000      9.011      10.449
ENGINE_SIZE             19.4963      0.902      21.626      0.000      17.727      21.265
=====
Omnibus:                 60.372    Durbin-Watson:              1.740
Prob(Omnibus):            0.000    Jarque-Bera (JB):            91.765
Skew:                    -0.462    Prob(JB):                   1.18e-20
Kurtosis:                 4.101    Cond. No.                     44.9
=====

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
0      214.598964
dtype: float64
```

▼ Exercício. Regressão com Atributos Categóricos (RESOLVIDO)

Faça agora um modelo de Regressão Múltipla adicionando o atributo categórico `VEHICLECLASS` ao modelo anterior. Sendo um atributo categórico o `statsmodel` fará automaticamente o *hot encode* desse atributo (o *hot encode* é uma importante técnica para tornar numérico atributos categóricos e é importante para uma série de modelos que requerem dados numéricos como a regressão. Se você não conhece, pesquise ou pergunte ao professor sobre esta transformação).

```
model = sm.ols(formula='CO2EMISSIONS ~ FUELCONSUMPTION_COMB + ENGINE_SIZE + VEHICLECLASS', data=df)

result = model.fit()
print(result.summary())
```

```

=====
                        OLS Regression Results
=====
Dep. Variable:          CO2EMISSIONS      R-squared:                0.870
Model:                  OLS              Adj. R-squared:            0.868
Method:                 Least Squares    F-statistic:                414.5
Date:                   Wed, 09 Mar 2022  Prob (F-statistic):          0.00
Time:                   18:31:22         Log-Likelihood:           -4850.3
No. Observations:       1067            AIC:                       9737.
Df Residuals:           1049            BIC:                       9826.
Df Model:                17
Covariance Type:        nonrobust
=====
                        coef      std err          t      P>|t|      [0.025      0.975]
-----
Intercept              85.1547      3.314      25.694      0.000      78.652      91.658
VEHICLECLASS[T.FULL-SIZE] -1.1773      3.158     -0.373      0.709     -7.375      5.020
VEHICLECLASS[T.MID-SIZE] -4.5891      2.482     -1.849      0.065     -9.460      0.282
VEHICLECLASS[T.MINICOMPACT] 0.7377      3.801      0.194      0.846     -6.720      8.196
VEHICLECLASS[T.MINIVAN]    0.8707      6.444      0.135      0.893    -11.774     13.516
VEHICLECLASS[T.PICKUP TRUCK - SMALL] 27.1642      6.916      3.928      0.000     13.593     40.735
VEHICLECLASS[T.PICKUP TRUCK - STANDARD] 1.4902      3.745      0.398      0.691     -5.858      8.839
VEHICLECLASS[T.SPECIAL PURPOSE VEHICLE] 18.1171      8.881      2.040      0.042      0.690     35.544
VEHICLECLASS[T.STATION WAGON - MID-SIZE] -5.8249      9.569     -0.609      0.543    -24.601     12.952
VEHICLECLASS[T.STATION WAGON - SMALL] 7.4700      4.217      1.771      0.077     -0.804     15.744
VEHICLECLASS[T.SUBCOMPACT] 7.6220      3.381      2.255      0.024      0.988     14.256
VEHICLECLASS[T.SUV - SMALL] 11.4515      2.580      4.439      0.000      6.390     16.513
VEHICLECLASS[T.SUV - STANDARD] 9.9109      3.148      3.148      0.002      3.734     16.088
VEHICLECLASS[T.TWO-SEATER] 10.3299      3.306      3.125      0.002      3.843     16.817
VEHICLECLASS[T.VAN - CARGO] 13.0886      5.854      2.236      0.026      1.601     24.576
VEHICLECLASS[T.VAN - PASSENGER] 33.0287      5.860      5.636      0.000     21.530     44.528
FUELCONSUMPTION_COMB    8.0833      0.435     18.600      0.000      7.231      8.936
ENGINE_SIZE             21.7192      0.924     23.495      0.000     19.905     23.533
=====
Omnibus:                 44.735    Durbin-Watson:              1.679
Prob(Omnibus):            0.000    Jarque-Bera (JB):            59.488
Skew:                    -0.408    Prob(JB):                   1.21e-13
Kurtosis:                 3.820    Cond. No.                     180.
=====

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
result.params

Intercept              85.154742
VEHICLECLASS[T.FULL-SIZE] -1.177255
VEHICLECLASS[T.MID-SIZE] -4.589137
VEHICLECLASS[T.MINICOMPACT] 0.737737
VEHICLECLASS[T.MINIVAN]    0.870664
```

```
VEHICLECLASS[T.PICKUP TRUCK - SMALL] 27.164208
VEHICLECLASS[T.PICKUP TRUCK - STANDARD] 1.490240
VEHICLECLASS[T.SPECIAL PURPOSE VEHICLE] 18.117114
VEHICLECLASS[T.STATION WAGON - MID-SIZE] -5.824881
VEHICLECLASS[T.STATION WAGON - SMALL] 7.469977
VEHICLECLASS[T.SUBCOMPACT] 7.622019
VEHICLECLASS[T.SUV - SMALL] 11.451534
VEHICLECLASS[T.SUV - STANDARD] 9.910910
VEHICLECLASS[T.TWO-SEATER] 10.329937
VEHICLECLASS[T.VAN - CARGO] 13.088594
VEHICLECLASS[T.VAN - PASSENGER] 33.028659
FUELCONSUMPTION_COMB 8.083254
ENGINE SIZE 21.719247
dtype: float64
```

result.params.index

```
Index(['Intercept', 'VEHICLECLASS[T.FULL-SIZE]', 'VEHICLECLASS[T.MID-SIZE]',
      'VEHICLECLASS[T.MINICOMPACT]', 'VEHICLECLASS[T.MINIVAN]',
      'VEHICLECLASS[T.PICKUP TRUCK - SMALL]',
      'VEHICLECLASS[T.PICKUP TRUCK - STANDARD]',
      'VEHICLECLASS[T.SPECIAL PURPOSE VEHICLE]',
      'VEHICLECLASS[T.STATION WAGON - MID-SIZE]',
      'VEHICLECLASS[T.STATION WAGON - SMALL]', 'VEHICLECLASS[T.SUBCOMPACT]',
      'VEHICLECLASS[T.SUV - SMALL]', 'VEHICLECLASS[T.SUV - STANDARD]',
      'VEHICLECLASS[T.TWO-SEATER]', 'VEHICLECLASS[T.VAN - CARGO]',
      'VEHICLECLASS[T.VAN - PASSENGER]', 'FUELCONSUMPTION_COMB',
      'ENGINE SIZE'],
      dtype='object')
```

O modelo acima é ainda melhor que os modelos anteriores. Ele apresenta, além do R2, um R2-Ajustado melhor (que inclui uma penalidade para o aumento do número de variáveis preditoras).

