
Facens

Exercício

Escreva um código para rodar amostragem aleatória com 1.000 repetições, para prever X19 em função de X9

Facens

Exercício

```
# Variável independente
X = pd.DataFrame(dados["x9"])

# Variável dependente
y = pd.DataFrame(dados["x19"])

model = []
mae = []
mse = []
rmse = []
for i in range(1,10):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
    linearRegressor = LinearRegression()
    linearRegressor.fit(X_train, y_train)
    linearRegressor.intercept_
    linearRegressor.coef_
    y_pred = linearRegressor.predict(X_test)
    mae.append(metrics.mean_absolute_error(y_test, y_pred))
    mse.append(metrics.mean_squared_error(y_test, y_pred) )
    rmse.append(np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

Facens

Exercício

```
print("MAE")
print("MAE médio: ", round(statistics.mean(mae),2))
print("MAE desvio padrão: ", round(np.std(mae),2))
print("MSE")
print("MSE médio: ", round(statistics.mean(mse),2))
print("MSE desvio padrão: ", round(np.std(mse),2))
print("RMSE")
print("RMSE médio: ", round(statistics.mean(rmse),2))
print("RMSE desvio padrão: ", round(np.std(rmse),2))
```

```
MAE
MAE médio:  0.76
MAE desvio padrão:  0.05
MSE
MSE médio:  0.92
MSE desvio padrão:  0.22
RMSE
RMSE médio:  0.95
RMSE desvio padrão:  0.11
```

Facens

Regressão Múltipla

```
import statsmodels.api as sm

os.chdir("D:\Dropbox\Fund Prog e Estatistica\db")
dados = pd.read_csv("HBAT.csv")

# Variável independente
X = dados.iloc[:,10:20]

# Variável dependente
y = dados["x19"]

X = sm.add_constant(X)

model = sm.OLS(y, X).fit()
predictions = model.predict(X)
model.summary()
```

Facens

Regressão Múltipla

Dep. Variable:	x19	**R-squared:**	0.656
Model:	OLS	**Adj. R-squared:**	0.617
Method:	Least Squares	**F-statistic:**	16.98
Date:	Fri, 06 Dec 2019	**Prob (F-statistic):**	9.32e-17
Time:	21:40:47	**Log-Likelihood:**	-105.58
No. Observations:	100	**AIC:**	233.2
Df Residuals:	89	**BIC:**	261.8
Df Model:	10		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[95.0% Conf. Int.]**
const	2.0075	1.296	1.550	0.125	-0.567 4.582
x9	0.2708	0.132	2.053	0.043	0.009 0.533
x10	0.0202	0.081	0.249	0.804	-0.141 0.181
x11	0.2459	0.342	0.719	0.474	-0.434 0.926
x12	0.5622	0.090	6.228	0.000	0.383 0.742
x13	-0.0739	0.062	-1.189	0.238	-0.197 0.050
x14	-0.1120	0.099	-1.136	0.259	-0.308 0.084
x15	0.0280	0.051	0.555	0.580	-0.072 0.128
x16	0.2371	0.131	1.804	0.075	-0.024 0.498
x17	-0.1672	0.348	-0.481	0.632	-0.858 0.524
x18	-0.0381	0.663	-0.057	0.954	-1.355 1.279

Omnibus:	3.312	**Durbin-Watson:**	2.341
Prob(Omnibus):	0.191	**Jarque-Bera (JB):**	2.548
Skew:	-0.251	**Prob(JB):**	0.280
Kurtosis:	2.400	**Cond. No.**	324.

Facens

Estratégias: Leave-one-out

- Bom para entender o impacto de uma observação em todo o modelo.
- Se o resultado desse modelo for diferente dos demais, então a observação retirada é considerada um outlier (observação extrema)
- Para isso, podemos fazer um teste de hipótese, comparando a métrica de qualidade do modelo (MAE, MSE, RMSE) com os demais

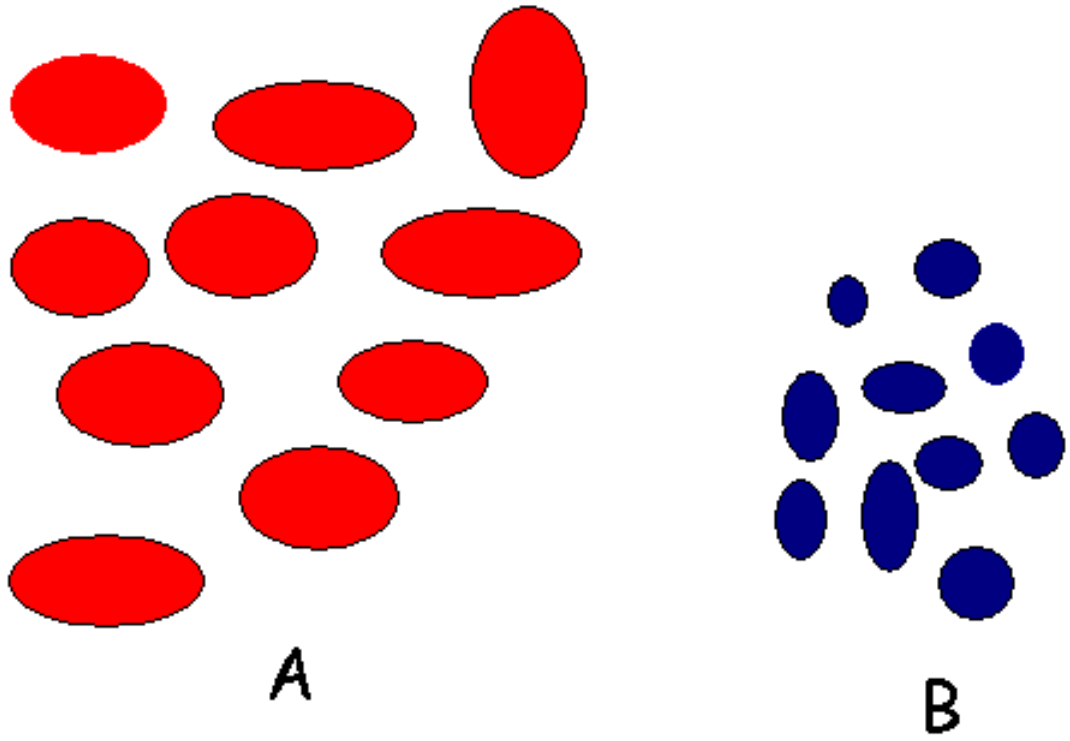
→ Comparação da média dos resultados de MAE (MSE ou RMSE), com o respectivo do modelo sem a observação outlier.

```
from scipy import stats
stats.ttest_1samp(mae_demais,const_mae)
```

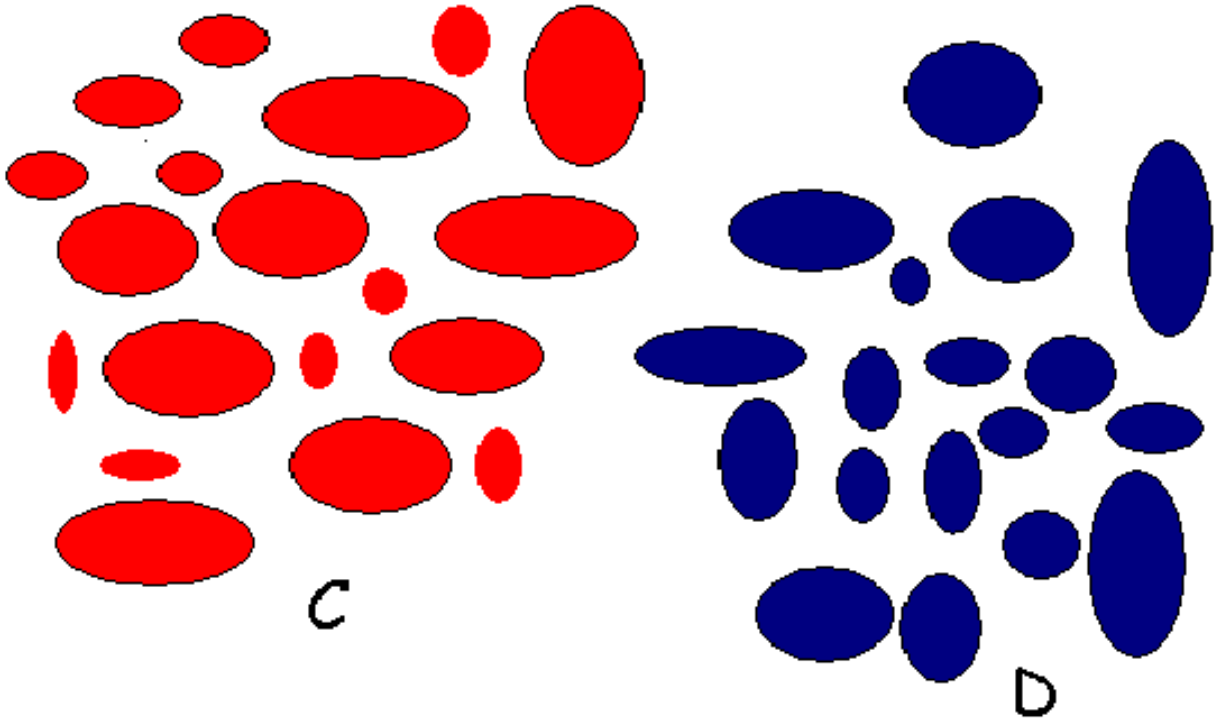
Facens

ANÁLISE DE VARIÂNCIA (ANOVA)

Facens



Facens


C
D

Análise de Variância (ANOVA)

- Detecção e estimação de relações entre médias
- Detecção e estimação entre componentes de variabilidade
- Variabilidade associada a "m" fontes de variação
- Propriedade aditiva da variância:
  - Variância total = dentre amostras + entre amostras
    - total: variação de todas as medidas em relação à média geral
    - dentre: variação de cada amostra em relação à sua média
    - entre: variação das "n" médias em relação à média geral

Facens

Análise de Variância (ANOVA)

OBSERVAÇÕES

- A ANOVA não considera que os tratamentos tenham algum ordenamento específico

  Para agregar esta informação na análise, usa-se a Análise de Regressão
- ANOVA com 2 tratamentos (r = 2) não deve ser realizada, uma vez que corresponde a um teste t homocedástico bilateral
- ANOVA pode ter mais do que 2 fatores avaliados (ANOVA multivariada)

	TA1	TA2	TA3	TA4
TB1	10 11 15	9 9 10	9 15 13	14 10
TB2	20 21 18	23 23 21	21 22	29 28 25

Análise de Variância (ANOVA)

PRESSUPOSIÇÕES:

- Cada observação deve ser independente das demais;

  condição garantida pelo processo de amostragem

- Cada tratamento deve ter distribuição normal;

Teste alternativo: Kruskal-Wallis (teste não paramétrico)

ANOVA - Python

```
from scipy import stats

dados['x1'] = dados['x1'].astype('category’)
dados["x1_cat"] = dados["x1"].cat.codes
f, p = stats.f_oneway(dados['x19'],
                      dados['x1_cat’])

print ('One-way ANOVA')
print ('=============')
print ('F value:', f)
print ('p value:', p, '\n')
```

Facens

ANOVA - Tabela

```
from statsmodels.formula.api import ols
model = ols('x19 ~ C(x1_cat)', dados).fit()
res = sm.stats.anova_lm(model, typ= 2)
res
```

	sum_sq	df	F	PR(>F)
C(x1_cat)	68.942925	2.0	46.645002	6.408468e-15
Residual	71.684675	97.0	NaN	NaN

Facens

Database

Variáveis Independentes	
X_1 Tipo de Cliente	X_4 Região
X_2 Tipo de Indústria	X_5 Sistema de Distribuição
X_3 Tamanho da Empresa	

Variável Dependente
X_{19} Satisfação
X_{20} Probabilidade de Recomendar
X_{21} Probabilidade de Comprar

Facens

Análise de Variância

Quando a ANOVA indica a aceitação de H_0, conclui-se que todas as médias dos tratamentos são iguais entre si, ou melhor, que não há diferenças significativas entre as médias dos tratamentos.

Neste caso, encerra-se a análise.

No entanto, quando H_0 é rejeitada, a ANOVA não é capaz de identificar quais as médias são diferentes entre si.

Basta que apenas uma média seja diferente para que a ANOVA indique a rejeição da H_0.

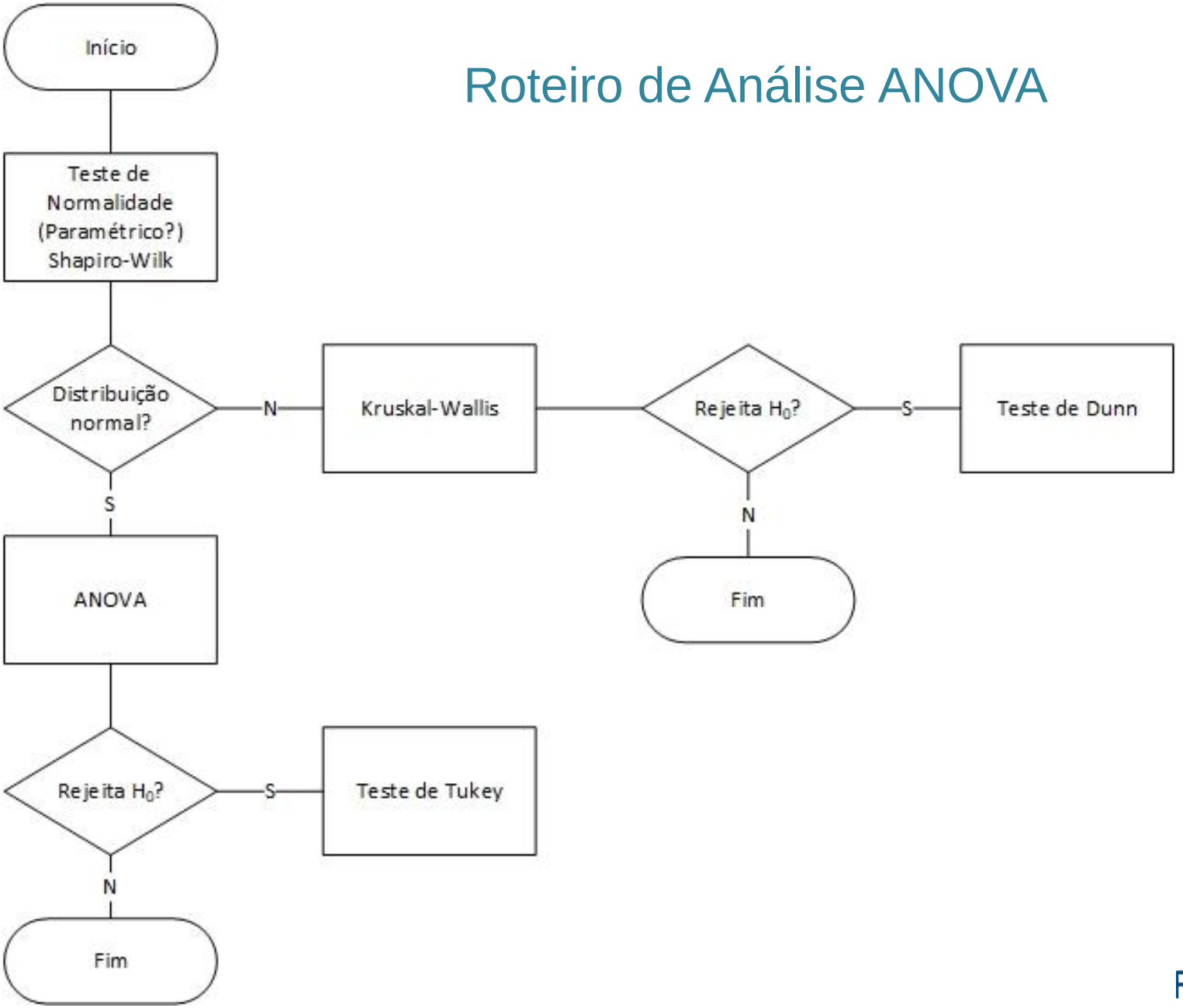
Como descobrir quais médias são diferentes?

Através de um Teste de Comparação Múltipla

Exemplos:
- Teste de Tukey
- Teste de Duncan
- Teste de Dunnet
- Teste de Scheffe
- Teste de Bonferroni
- Teste de Fisher

Facens

Roteiro de Análise ANOVA



Teste de Tukey

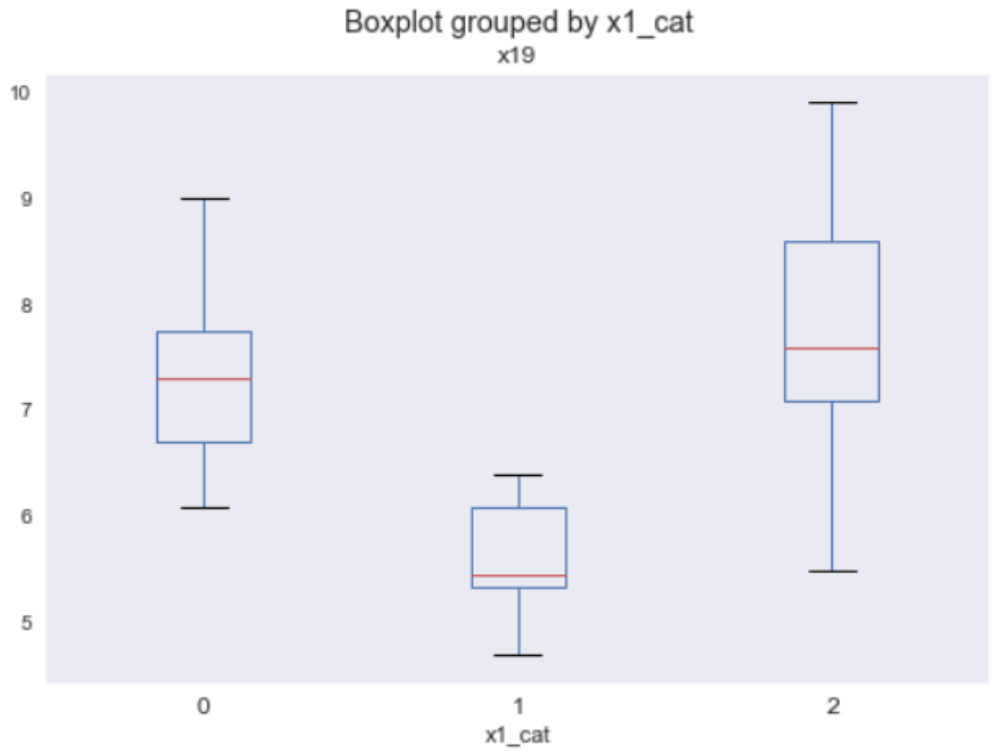
```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.multicomp import MultiComparison

mc = MultiComparison(dados['x19'], dados['x1_cat'])
result = mc.tukeyhsd()

print(result)
print(mc.groupsunique)
```

```
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================
group1 group2 meandiff  lower   upper  reject
---------------------------------------------
  0      1    -1.5893  -2.0898 -1.0888  True
  0      2     0.3403  -0.1563  0.8368 False
  1      2     1.9295   1.4219  2.4372  True
---------------------------------------------
```

Facens

Teste de Tukey

```
import matplotlib.pyplot as plt
dados.boxplot(column="x19", by="x1_cat", grid=False)
plt.show()
```



Facens

Teste de Kruskal-Wallis

```
from scipy import stats

dados['x1'] = dados['x1'].astype('category’)
dados["x1_cat"] = dados["x1"].cat.codes
kw, p = stats.kruskal(dados["x19"],
                      dados['x1_cat’])


print (Kruskal-Wallis')
print ('=============')
print (‘KW value:’, kw)
print ('p value:', p, '\n')
```

Facens

Teste de Dunn

```
$ pip install scikit-posthocs

import scikit-posthocs as sp

x = [dados[dados['x1_cat']==0]['x19'],
     dados[dados['x1_cat']==0]['x19'],
     dados[dados['x1_cat']==0]['x19']]

sp.posthoc_dunn(x, p_adjust='holm')
```

Facens

ANOVA two-way – 2 Variáveis Independentes

```
def anova_table(aov):
    aov['mean_sq'] = aov[:]['sum_sq']/aov[:]['df']

    aov['eta_sq'] = aov[:-1]['sum_sq']/sum(aov['sum_sq'])

    aov['omega_sq'] = (aov[:-1]['sum_sq']-(aov[:-1]['df']*aov['mean_sq'][-
1]))/(sum(aov['sum_sq'])+aov['mean_sq'][-1])

    cols = ['sum_sq', 'mean_sq', 'df', 'F', 'PR(>F)', 'eta_sq', 'omega_sq']
    aov = aov[cols]
    return aov


from statsmodels.formula.api import ols
model2 = ols('x19 ~ x1_cat + x5_cat', dados).fit()

res2 = sm.stats.anova_lm(model2, typ= 2)
res2

anova_table(res2)
```

Facens

ANOVA two-way – Tukey

```
mc = MultiComparison(dados['x19'], dados['x1_cat'])
result = mc.tukeyhsd()
print(result)
#print(mc.groupsunique)
```

```
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================
group1 group2 meandiff  lower   upper  reject
---------------------------------------------
  0      1    -1.5893  -2.0898 -1.0888  True
  0      2     0.3403  -0.1563  0.8368 False
  1      2     1.9295   1.4219  2.4372  True
---------------------------------------------
```

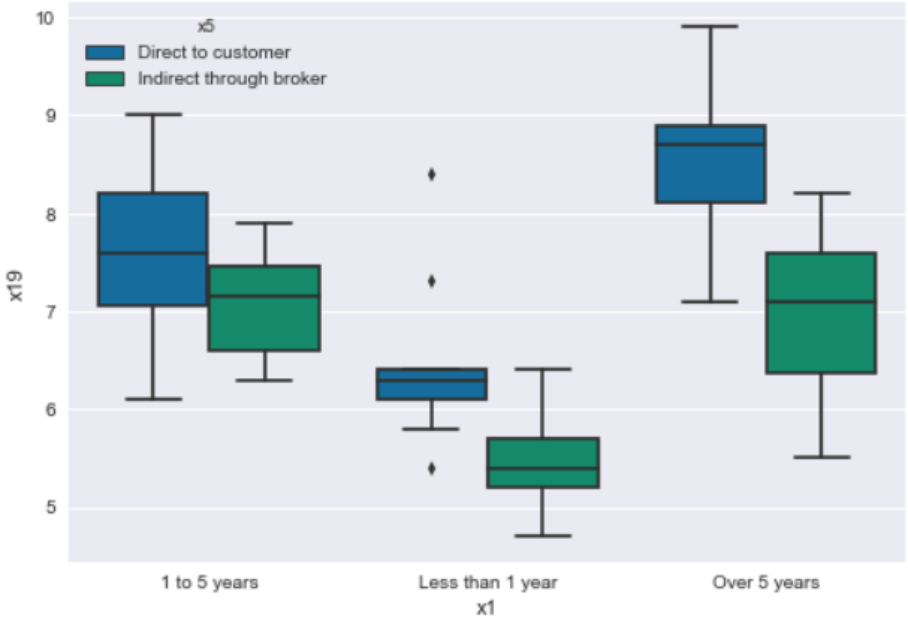
```
mc = MultiComparison(dados['x19'], dados['x5_cat'])
result = mc.tukeyhsd()
print(result)
```

```
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================
group1 group2 meandiff  lower   upper  reject
---------------------------------------------
  0      1    -1.2455  -1.6556 -0.8353  True
---------------------------------------------
```

Facens

ANOVA two-way – 2 Variáveis Independentes

```
import seaborn as sns
sns.boxplot(y='x19', x='x1',
                    data=dados,
                    palette="colorblind",
                    hue='x5')
plt.show()
```



Facens

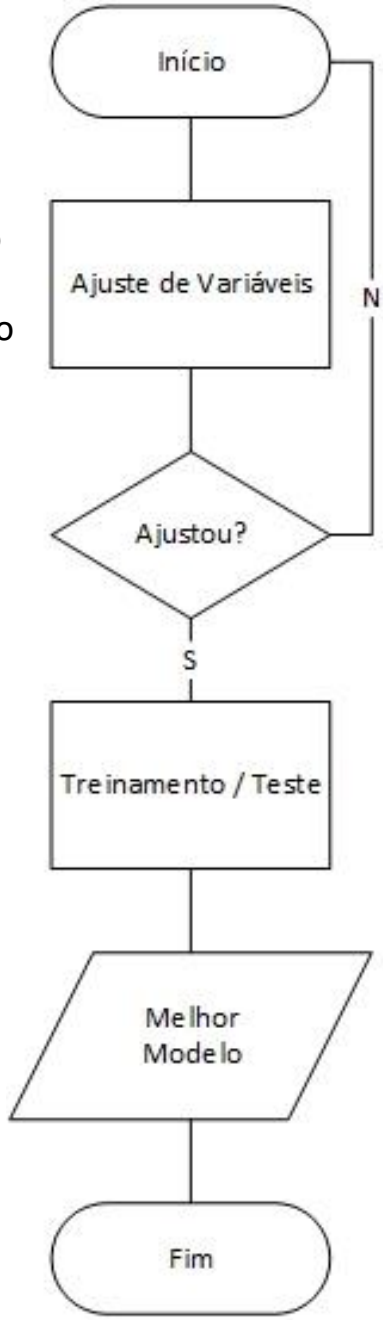
Kruskal-Wallis – 2 ou mais Variáveis Independentes

```
kw, p = stats.kruskal(dados["x19"],
                      dados['x1_cat'],
                      dados['x5_cat'])

print ('Kruskal-Wallis')
print ('=============')
print ('KW value:', kw)
print ('p value:', p, '\n')
```

Facens

Roteiro Regressão Linear

- correlação
- R^2
- R^2 ajustado



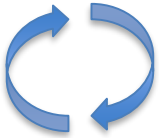
- MAE
- MSE
- RMSE

Facens

Roteiro de Pesquisa

- Obter base de dados
- Pergunta da Pesquisa / Objetivo
- Estatística Descritiva

média	3º quartil
desvio padrão	máximo
mínimo	boxplot
1º quartil	histograma
2º quartil	distribuição



- Limpeza de Dados
- Análise / Inferência Estatística
  - Regressão Linear
    - Treinamento / Teste
  - ANOVA / Kruskal-Wallis

Facens


Facens