Testes de Hipóteses (Lab)

Caso: **Mais do Mercado de Imóveis de Melbourne**

O Mercado de Imóveis de Melbourne ficou "*esfriou*" em 2018, com preços significativamente menores que no ano anterior?

Empregue os dados da URL:
http://meusite.mackenzie.br/rogerio/TIC/Melbourne_housing_FULL.csv

Exercício 0. Imports e Acesso aos Dados

Aqui você deve **importar as bibliotecas das ferramentas** que você pretende empregar na sua EDA. A seguir você deve fazer a aquisição dos dados. Lembre-se de verificar a origem dos dados para empregar as funções e parâmetros corretos de leitura dos dados (**extensão do arquivo, headers, separador**).

```
# imports
import pandas                  as pd
import numpy                   as np
import matplotlib.pyplot       as plt
import seaborn                 as sns
import statsmodels.formula.api as sm
import warnings
warnings.filterwarnings("ignore")  # Suppress all warnings
```

```
# read data
houses = pd.read_csv('http://meusite.mackenzie.br/rogerio/TIC/Melbourne_housing_FULL.csv')
houses.head()
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016	2.5
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5

Exercício 1. Revistando os Valores Nulos (**RESOLVIDO**)

Antes de seguirmos vamos revisitar o tratamento de nulos. lembra-se de nossa análise no Lab T5. Nossa estratégia de imputar os valores, embora adequada para aquele momento, não foi uma solução tão boa. Recorda a alternativa discutida?

```
print(houses.isnull().sum())
pr = pd.DataFrame(houses.Price.dropna())
houses.Price = houses.Price.fillna(houses.Price.mean())

sns.distplot(pr)
sns.distplot(houses.Price)
```

```
Suburb            0
Address           0
Rooms             0
Type              0
Price          7610
Method            0
SellerG           0
Date              0
Distance          1
```

Imputando dados com a mesma distribuição

O código aqui é talvez um pouco complexo. Fique a vontade de pular o detalhe do código. Ele não tem maior importância no que segue mas, **concentre-se no conceito e resultado que ele produz**. O mesmo preenchimento é feito para todos os dados nulos.

```
Lattitude      7976
```

```
houses = pd.read_csv('http://meusite.mackenzie.br/rogerio/TIC/Melbourne_housing_FULL.csv')

def filldist(df,col):
    v = pd.DataFrame(df[col].dropna()).reset_index()
    df[col] = df[col].fillna(-1)
    df[col] = df[col].apply(lambda x: (v.iloc[np.random.randint(len(v))][col]) if x==-1 el
    return df[col].isnull().sum()

filldist(houses,'Price')

sns.distplot(pr)
sns.distplot(houses.Price)

s = houses.isnull().sum()
for i in s.index:
    if s[i]!=0:
        filldist(houses,i)

houses.isnull().sum()
```

```
Suburb          0
Address         0
Rooms           0
Type            0
Price           0
Method          0
SellerG         0
Date            0
Distance        0
Postcode        0
Bedroom2        0
Bathroom        0
Car             0
Landsize        0
BuildingArea    0
YearBuilt       0
CouncilArea     0
Lattitude       0
Longtitude      0
Regionname      0
```

De qualquer modo essa estratégia, embora não apresente qualquer distorção na distribuição dos dados, ainda pode introduzir viés na análise dos dados, sendo sempre um risco a introdução artificial dos dados e, tanto maior quanto maior o número de dados introduzidos.

Assim para análise à seguir adotaremos a estratégia de simples exclusão dos dados nulos.

Vamos então retomar os dados originais para análise e resposta às perguntas do caso.

```
houses = pd.read_csv('http://meusite.mackenzie.br/rogerio/TIC/Melbourne_housing_FULL.csv')

houses = houses.dropna()
houses.head()
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5
4	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5
6	Abbotsford	55a Park St	4	h	1600000.0	VB	Nelson	4/06/2016	2.5
11	Abbotsford	124 Yarra St	3	h	1876000.0	S	Nelson	7/05/2016	2.5
14	Abbotsford	98 Charles St	2	h	1636000.0	S	Nelson	8/10/2016	2.5

Exercício 2. Qual a média de preços dos imóveis por ano?

DICA:

1. Empregue `pd.DatetimeIndex(houses.Date).year` para criar um atributo com o ano.
2. Empregue então `df.groupby(attr).attr.mean()`

```
houses.dtypes
from datetime import datetime

houses['Year'] = pd.DatetimeIndex(houses.Date).year

print(houses.groupby('Year').Price.mean())
```

```
Year
2016    1.103024e+06
2017    1.075872e+06
2018    1.130886e+06
Name: Price, dtype: float64
```

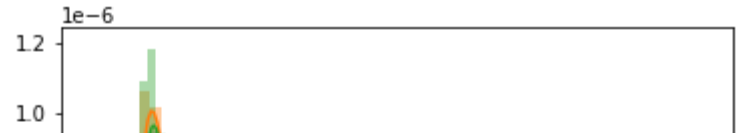
Exercício 3. Alterou-se a distribuição de preços de um ano para o outro?

DICA:

1. Empregue `sns.distplot(x)`, sendo x a seleção de dados de Preço para um único ano
2. Você pode fazer `sns.displot()` seguidos para sobrepor os gráficos em uma mesma exibição

```
sns.distplot(houses[houses.Year==2016].Price)
sns.distplot(houses[houses.Year==2017].Price)
sns.distplot(houses[houses.Year==2018].Price)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdca2839590>
```



Exercício 4. Qual o % de diferença da média de Preços 2017/2018? A diferença é significativa?

DICA:

1. Faça um teste de hipótese, sintaxe

`s, p = stats.ttest_ind( serie1 , serie2 , equal_var=False)`

onde `serie1` e `serie2` são as séries que você deseja comparar.

2. Para valor `p < 0.05` rejeite a Hipótese Nula (médias iguais).

```
# imports
from scipy import stats

print('Diferença de preços 2018/2017: {:0.2f} %'.format(( houses.groupby('Year').Price.mea
```

```
Diferença de preços 2018/2017: 5.11 %
```

```
s, p = stats.ttest_ind(houses[houses.Year==2017].Price,
                       houses[houses.Year==2018].Price, equal_var=False)
s, p

if p < 0.05:
  print('Rejeita a Hipótese Nula')
else:
  print('Aceita a Hipótese Nula')

print(p)
```

```
Rejeita a Hipótese Nula
0.018918751400867774
```

Exercício 5. Teste de Hipóteses

Você quer vender um imóvel em Melbourne e acredita que vendas no método 'PI' ('passed in') - digamos, com o uso de um intermediador - não tem qualquer vantagem sobre o método 'S' ('sold') - digamos, venda direta - em termos de preço e que, portanto você estaria perdendo tempo e dinheiro buscando a venda por um intermediário (digamos, corretor). Você estaria correto?

DICA:

1. Faça um teste de hipótese, sintaxe

```
s, p = stats.ttest_ind( serie1 , serie2 , equal_var=False)
```

onde `serie1` e `serie2` são as séries que você deseja comparar.

2. Para valor `p < 0.05` rejeite a Hipótese Nula (médias iguais).

```
print('Diferença de preços PI/S: {:0.2f} %'.format(( houses[houses.Method=='PI'].Price.mea

s, p = stats.ttest_ind(houses[houses.Method=='PI'].Price,
                       houses[houses.Method=='S'].Price, equal_var=False)
s, p

if p < 0.05:
  print('Rejeita a Hipótese Nula')
else:
  print('Aceita a Hipótese Nula')

print(p)
```

```
Diferença de preços PI/S: 8.83 %
Rejeita a Hipótese Nula
3.317384526538828e-05
```

Caso: `Teste A/B` Udacity, Redução de Cancelamentos pela Alteração do Site

O Teste A/B visa comparar a efetividade de um experimento ou mudança e tem grande aplicação desde testes clínicos a avaliações de e-commerce.

Vamos verificar de modo bastante resumido o caso real do site **Udacity**.

Condição Inicial. As páginas iniciais do curso Udacity têm duas opções: "iniciar teste gratuito" e "acessar materiais do curso". Clicar em "Iniciar avaliação gratuita" solicita que o usuário insira as informações do cartão de crédito, subseqüentemente, inscrevendo-as em uma avaliação gratuita de 14 dias do curso, após o que são cobradas automaticamente. Os usuários que clicarem em "acessar materiais do curso" poderão visualizar o conteúdo do curso, mas não receberão suporte de treinamento, certificado verificado ou feedback do projeto.

Mudança. A Udacity testou uma mudança na qual os usuários que clicaram em "iniciar a avaliação gratuita" eram questionados sobre quanto tempo estavam dispostos a dedicar ao curso. Os usuários que escolherem 5 ou mais horas por semana serão submetidos ao processo de check-out, como de costume. Para usuários que indicam menos de 5 horas por semana, uma mensagem seria exibida indicando a necessidade de um compromisso maior de tempo para

permitir o sucesso no curso e sugerindo para eles acessarem o conteúdo gratuito. Nesse ponto, o aluno teria a opção de continuar se matriculando na avaliação gratuita ou acessar os materiais do curso gratuitamente.

Ojetivo. Aumentar a efetividade dos alunos inscritos no teste gratuíto permitindo a empresa focar seus esforços nesses alunos.

Desenho do Experimento.

> **Métricas**
>
> > Invariant Metrics (controle): number of cookies, number of clicks, click-through-probability
> >
> > Evaluation Metrics (objetivo): gross conversion, net conversion

Exercício 6. Aquisição e Preparação dos Dados (**RESOLVIDO**)

Você pode pular os detalhes se quiser e focar na avaliação dos resultados.

```
path = 'http://meusite.mackenzie.br/rogerio/TIC/udacity_ABTesting-master/data/'
df_control = pd.read_csv(path + "Control.csv")
df_experiment = pd.read_csv(path+ "Experiment.csv")

df_control.describe()
df_experiment.describe()

df_control_notnull = df_control[pd.isnull(df_control.Enrollments) != True]
df_experiment_notnull = df_experiment[pd.isnull(df_control.Enrollments) != True]

df_SignTest = pd.merge(df_control_notnull,df_experiment_notnull,on="Date")
df_SignTest['GrossConversion_cont'] = df_SignTest.Enrollments_x/df_SignTest.Clicks_x
df_SignTest['GrossConversion_exp'] = df_SignTest.Enrollments_y/df_SignTest.Clicks_y
df_SignTest['NetConversion_cont'] = df_SignTest.Payments_x/df_SignTest.Clicks_x
df_SignTest['NetConversion_exp'] = df_SignTest.Payments_y/df_SignTest.Clicks_y

cols = ['Date','GrossConversion_cont','GrossConversion_exp','NetConversion_cont','NetConve

df_SignTest = df_SignTest[cols]
# df_SignTest.describe()

print(df_SignTest)
print('NetConversion controle......: {:.2f} %'.format(df_SignTest.NetConversion_cont.mean(
print('NetConversion experimento...: {:.2f} %'.format(df_SignTest.NetConversion_exp.mean()

print('GrossCoversion controle.....: {:.2f} %'.format(df_SignTest.GrossConversion_cont.mea
print('GrossConversion experimento.: {:.2f} %'.format(df_SignTest.GrossConversion_exp.mean
```

```
             Date  GrossConversion_cont  ...  NetConversion_cont  NetConversion_exp
0   Sat, Oct 11              0.195051  ...            0.101892           0.049563
1   Sun, Oct 12              0.188703  ...            0.089859           0.115924
2   Mon, Oct 13              0.183718  ...            0.104510           0.089367
3   Tue, Oct 14              0.186603  ...            0.125598           0.111245
4   Wed, Oct 15              0.194743  ...            0.076464           0.112981
5   Thu, Oct 16              0.167679  ...            0.099635           0.077411
6   Fri, Oct 17              0.195187  ...            0.101604           0.056410
7   Sat, Oct 18              0.174051  ...            0.110759           0.095092
8   Sun, Oct 19              0.189580  ...            0.086831           0.110473
9   Mon, Oct 20              0.191638  ...            0.112660           0.113953
10  Tue, Oct 21              0.226067  ...            0.121107           0.082176
11  Wed, Oct 22              0.193317  ...            0.109785           0.087391
12  Thu, Oct 23              0.190977  ...            0.084211           0.105919
13  Fri, Oct 24              0.326895  ...            0.181278           0.134864
14  Sat, Oct 25              0.254703  ...            0.185239           0.121076
15  Sun, Oct 26              0.227401  ...            0.146893           0.145743
16  Mon, Oct 27              0.306983  ...            0.163373           0.154345
17  Tue, Oct 28              0.209239  ...            0.123641           0.163043
18  Wed, Oct 29              0.265223  ...            0.116373           0.132050
19  Thu, Oct 30              0.227520  ...            0.102180           0.092033
20  Fri, Oct 31              0.246459  ...            0.143059           0.170360
21   Sat, Nov 1              0.229075  ...            0.136564           0.143885
22   Sun, Nov 2              0.297258  ...            0.096681           0.142265

[23 rows x 5 columns]
NetConversion controle......: 11.83 %
NetConversion experimento...: 11.34 %
GrossCoversion controle.....: 22.04 %
GrossConversion experimento.: 19.96 %
```

Exercício 7. Teste de Hipóteses

a. Aplique o t-test agora para as duas métricas de avaliação. Quais os resultados para essas duas métricas?

DICA: É um teste de duas amostras em que você emprega para cada métrica a variável _cont (controle) e a variável _exp (experimento)

b. Esses dados não estão aqui disponíveis, mas qual seria a utilidade de fazer teste nas métricas consideradas invariantes do experimento e qual o resultado desejado?

```
s, p = stats.ttest_ind(df_SignTest.GrossConversion_cont,
                                  df_SignTest.GrossConversion_exp, equal_var=False)
print('p-value: {:.4f}'.format(p))

if p < 0.05:
  print('Rejeita a Hipótese Nula')
else:
  print('Aceita a Hipótese Nula')

s, p = stats.ttest_ind(df_SignTest.NetConversion_cont,
                                  df_SignTest.NetConversion_exp, equal_var=False)
```

```
print('p-value: {:.4f}'.format(p))

if p < 0.05:
  print('Rejeita a Hipótese Nula')
else:
  print('Aceita a Hipótese Nula')
```

```
p-value: 0.1308
Aceita a Hipótese Nula
p-value: 0.5928
Aceita a Hipótese Nula
```

Conclusão: A mudança no site não foi efetiva para aumentar a efetividade dos alunos que aderiam ao teste gratuíto dos cursos.

Caso: **Covid X Temperatura**

Exercício 8. Empregue o data set abaixo para testar as hipóteses,

> H_0: a temperatura >= 24 C não afeta o surto de COVID-19

> H_1: a temperatura >= 24 C afeta o surto de COVID-19

DICA: Empregue o valor de temperatura para separar os casos confirmados da doença.

```
covid = pd.read_csv('http://meusite.mackenzie.br/rogerio/Corona_Updated.csv')
covid.head()
```

	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered	Latit
0	Hubei	Mainland China	2020-03-10T15:13:05	67760	3024	47743	30.9
1	NaN	Italy	2020-03-10T17:53:02	10149	631	724	43.(
2	NaN	Iran (Islamic Republic of)	2020-03-10T19:13:20	8042	291	2731	32.(

```
covid['Temp_Maior'] = covid['Temprature'].apply(lambda x : 0 if x < 24 else 1)
d1 = covid[(covid['Temp_Maior']==1)][['Confirmed']]
d2 = covid[(covid['Temp_Maior']==0)][['Confirmed']]


s, p = stats.ttest_ind(d1.Confirmed,
                       d2.Confirmed, equal_var=False)
s, p

if p < 0.05:
  print('Rejeita a Hipótese Nula')
else:
  print('Aceita a Hipótese Nula')

print(p)
```

```
Aceita a Hipótese Nula
0.10386045490678472
```