Storytelling_ROSSMANN

September 12, 2021

[3]:
```
jupyter_settings()
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
<IPython.core.display.HTML object>
```

[ ]:

[ ]:

1 Predições de Vendas das Lojas ROSSMANN

2 Agenda

1. Contexto
2. Desafio
3. Desenvolvimento da Solução
4. Conclusão & Demonstração
5. Próximos Passos

3 1. Contexto

- Reunião Mensal de Resultados
- CFO pediu uma Previsão de Vendas das Próximas 6 semanas de cada Loja

4 2. Desafio

5 Problema

- Definição do Budget para a Reforma das Lojas.

6 Causas

- Predição de Vendas Atual apresentada muita Divergencia
- O processo de Predição de Vendas é baseado em Experiencias Passadas.
- Todo a Previsão de Vendas é feita Manualmente pelas 1.115 Lojas da Rossmann.
- A visualização das Vendas é Limitada ao Computador.

7 Solução

- Usar Machine Learning para realizar a Previsão de Vendas de Todas as Lojas
- Visualização das Predições de Vendas poderão ser feitas pelo Smartphone

8 3. Desenvolvimento da Solução

9 DESCRICAO DOS DADOS

[7]:
```
print( 'Number of Rows: {}'.format( df1.shape[0] ) )
print( 'Number of Cols: {}'.format( df1.shape[1] ) )
```

```
Number of Rows: 1017209
Number of Cols: 18
```

10 Descriptive Statistics

[15]:
```
# Central Tendency - mean, meadina
ct1 = pd.DataFrame( num_attributes.apply( np.mean ) ).T
ct2 = pd.DataFrame( num_attributes.apply( np.median ) ).T

# dispersion - std, min, max, range, skew, kurtosis
d1 = pd.DataFrame( num_attributes.apply( np.std ) ).T
d2 = pd.DataFrame( num_attributes.apply( min ) ).T
d3 = pd.DataFrame( num_attributes.apply( max ) ).T
d4 = pd.DataFrame( num_attributes.apply( lambda x: x.max() - x.min() ) ).T
d5 = pd.DataFrame( num_attributes.apply( lambda x: x.skew() ) ).T
d6 = pd.DataFrame( num_attributes.apply( lambda x: x.kurtosis() ) ).T

# concatenar
m = pd.concat( [d2, d3, d4, ct1, ct2, d1, d5, d6] ).T.reset_index()
m.columns = ['attributes', 'min', 'max', 'range', 'mean', 'median', 'std',␣
 ↪'skew', 'kurtosis']
m
```

[15]:

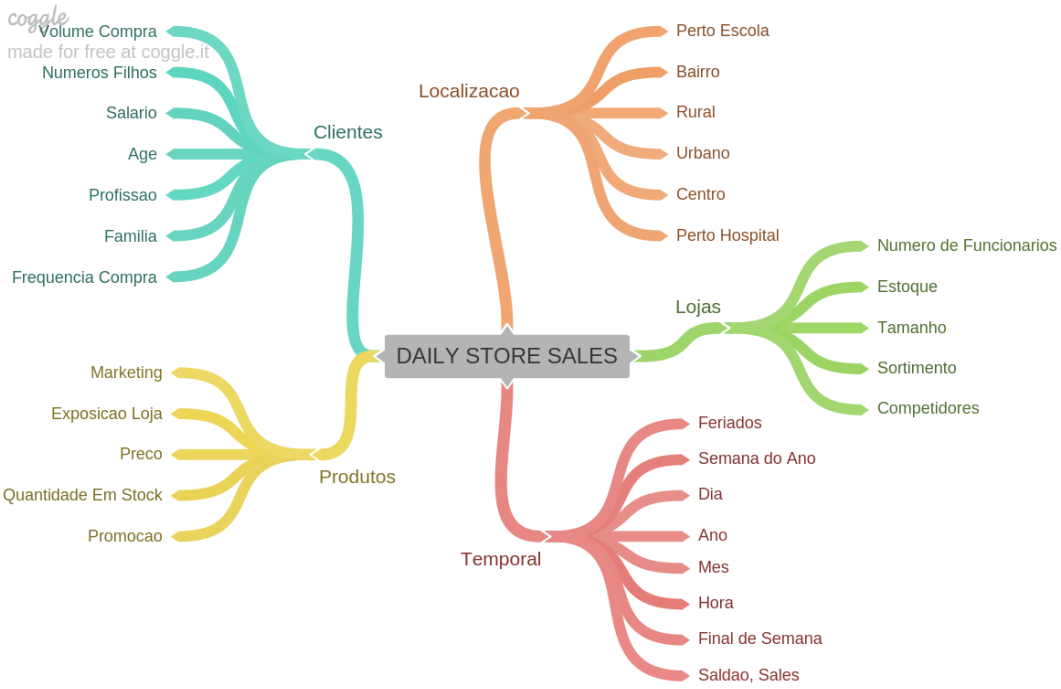
	attributes	min	max	range	mean	median	std	skew	kurtosis
0	store	1.0	1115.0	1114.0	558.429727	558.0	321.908493	-0.000955	-1.200524
1	day_of_week	1.0	7.0	6.0	3.998341	4.0	1.997390	0.001593	-1.246873
2	sales	0.0	41551.0	41551.0	5773.818972	5744.0	3849.924283	0.641460	1.778375
3	customers	0.0	7388.0	7388.0	633.145946	609.0	464.411506	1.598650	7.091773
4	open	0.0	1.0	1.0	0.830107	1.0	0.375539	-1.758045	1.090723

```
5                          promo     0.0       1.0       1.0     0.381515
0.0      0.485758   0.487838   -1.762018
6                 school_holiday     0.0       1.0       1.0     0.178647
0.0      0.383056   1.677842    0.815154
7           competition_distance    20.0  200000.0  199980.0  5935.442677
2330.0  12547.646829  10.242344  147.789712
8   competition_open_since_month     1.0      12.0      11.0     6.786849
7.0      3.311085  -0.042076   -1.232607
9    competition_open_since_year  1900.0    2015.0     115.0  2010.324840
2012.0      5.515591  -7.235657  124.071304
10                        promo2     0.0       1.0       1.0     0.500564
1.0      0.500000  -0.002255   -1.999999
11             promo2_since_week     1.0      52.0      51.0    23.619033
22.0     14.310057   0.178723   -1.184046
12             promo2_since_year  2009.0    2015.0       6.0  2012.793297
2013.0      1.662657  -0.784436   -0.210075
13                       is_promo     0.0       1.0       1.0     0.155231
0.0      0.362124   1.904152    1.625796
```

11 Mapa Mental de Hipoteses

[20]:
```
Image( 'img/MindMapHypothesis.png' )
```

[20]:



12 Hipoteses Da Análise Exploratória

1. Lojas com maior sortimentos deveriam vender mais.

2. Lojas com competidores mais próximos deveriam vender menos.

3. Lojas com competidores à mais tempo deveriam vendem mais.

4. Lojas com promoções ativas por mais tempo deveriam vender mais.

5. Lojas com mais dias de promoção deveriam vender mais.

7. Lojas com mais promoções consecutivas deveriam vender mais.

8. Lojas abertas durante o feriado de Natal deveriam vender mais.

9. Lojas deveriam vender mais ao longo dos anos.

10. Lojas deveriam vender mais no segundo semestre do ano.

11. Lojas deveriam vender mais depois do dia 10 de cada mês.

12. Lojas deveriam vender menos aos finais de semana.

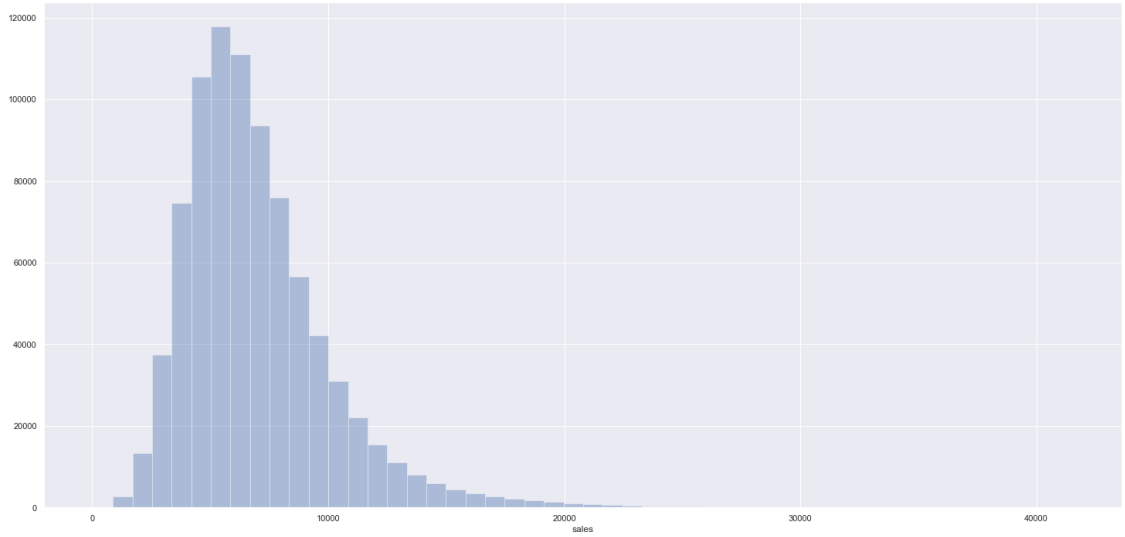
13. Lojas deveriam vender menos durante os feriados escolares.

13 ANALISE EXPLORATORIA DOS DADOS

14 Response Variable

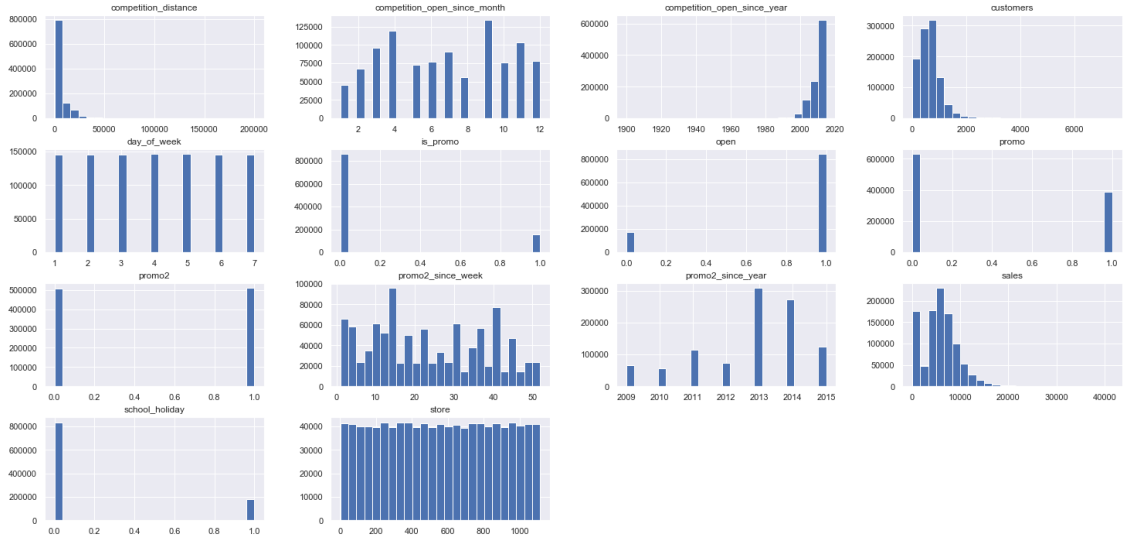
[26]:
```
sns.distplot( df4['sales'], kde=False  )
```

[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11f7a3910>
```



15 Numerical Variable

[27]:
```
num_attributes.hist( bins=25 );
```

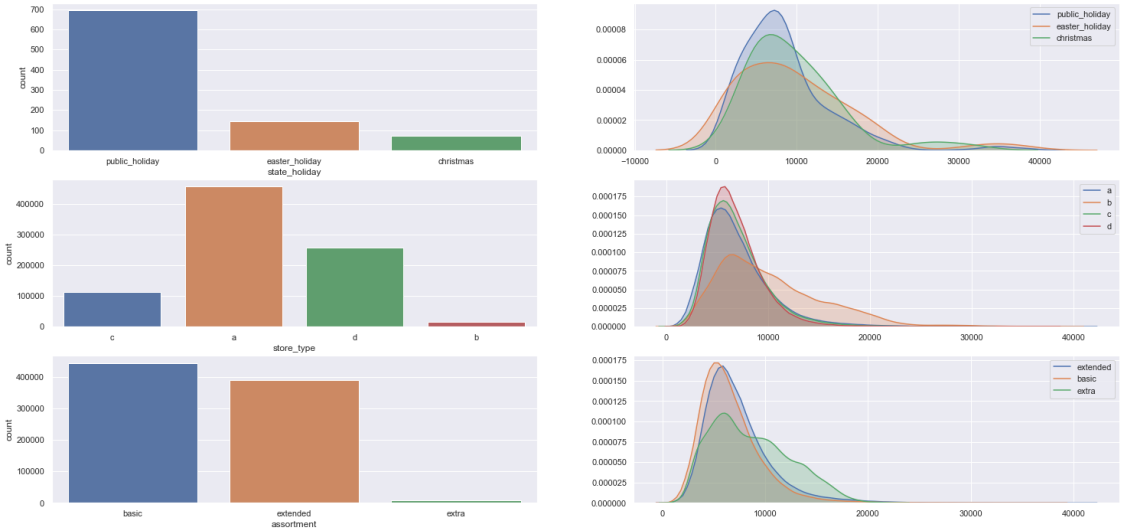


16 Categorical Variable

[28]:
```
# state_holiday
plt.subplot( 3, 2, 1 )
a = df4[df4['state_holiday'] != 'regular_day']
sns.countplot( a['state_holiday'] )

plt.subplot( 3, 2, 2 )
sns.kdeplot( df4[df4['state_holiday'] == 'public_holiday']['sales'],␣
 ↪label='public_holiday', shade=True )
sns.kdeplot( df4[df4['state_holiday'] == 'easter_holiday']['sales'],␣
 ↪label='easter_holiday', shade=True )
sns.kdeplot( df4[df4['state_holiday'] == 'christmas']['sales'],␣
 ↪label='christmas', shade=True )

# store_type
plt.subplot( 3, 2, 3 )
sns.countplot( df4['store_type'] )

plt.subplot( 3, 2, 4 )
sns.kdeplot( df4[df4['store_type'] == 'a']['sales'], label='a', shade=True )
sns.kdeplot( df4[df4['store_type'] == 'b']['sales'], label='b', shade=True )
sns.kdeplot( df4[df4['store_type'] == 'c']['sales'], label='c', shade=True )
sns.kdeplot( df4[df4['store_type'] == 'd']['sales'], label='d', shade=True )
```

```
# assortment
plt.subplot( 3, 2, 5 )
sns.countplot( df4['assortment'] )

plt.subplot( 3, 2, 6 )
sns.kdeplot( df4[df4['assortment'] == 'extended']['sales'], label='extended',␣
 ↪shade=True )
sns.kdeplot( df4[df4['assortment'] == 'basic']['sales'], label='basic',␣
 ↪shade=True )
sns.kdeplot( df4[df4['assortment'] == 'extra']['sales'], label='extra',␣
 ↪shade=True )
```

[28]: <matplotlib.axes._subplots.AxesSubplot at 0x15bf1af40>



17 Validação das Hipóteses

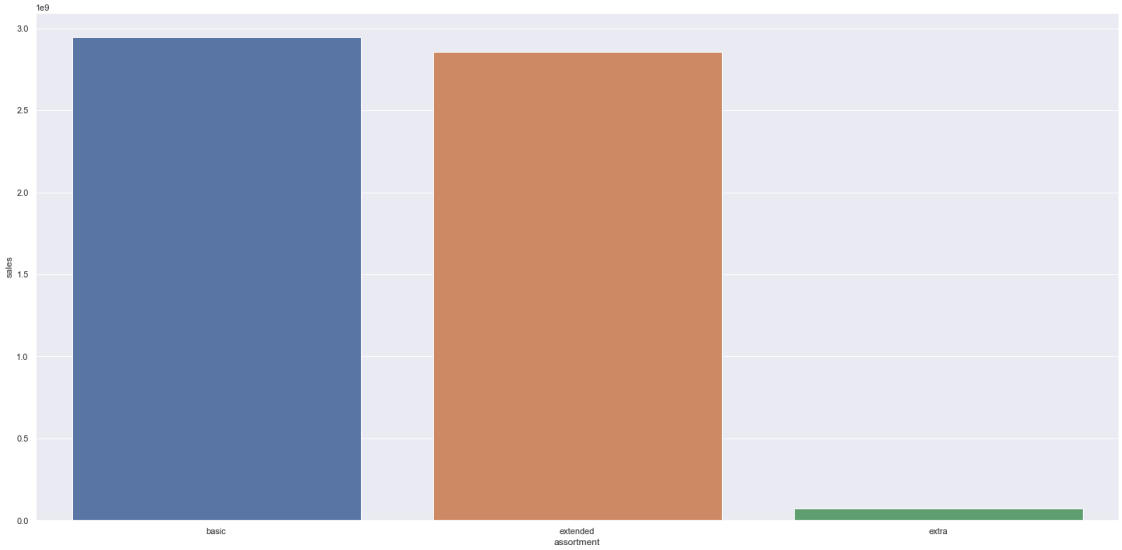
17.0.1 H1. Lojas com maior sortimentos deveriam vender mais.

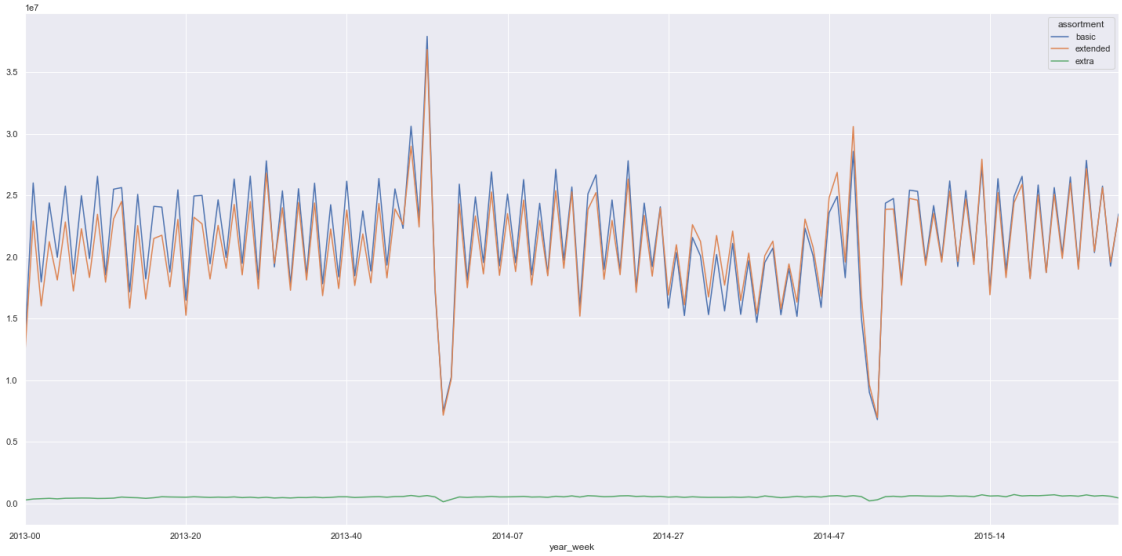
FALSA Lojas com MAIOR SORTIMENTO vendem MENOS.

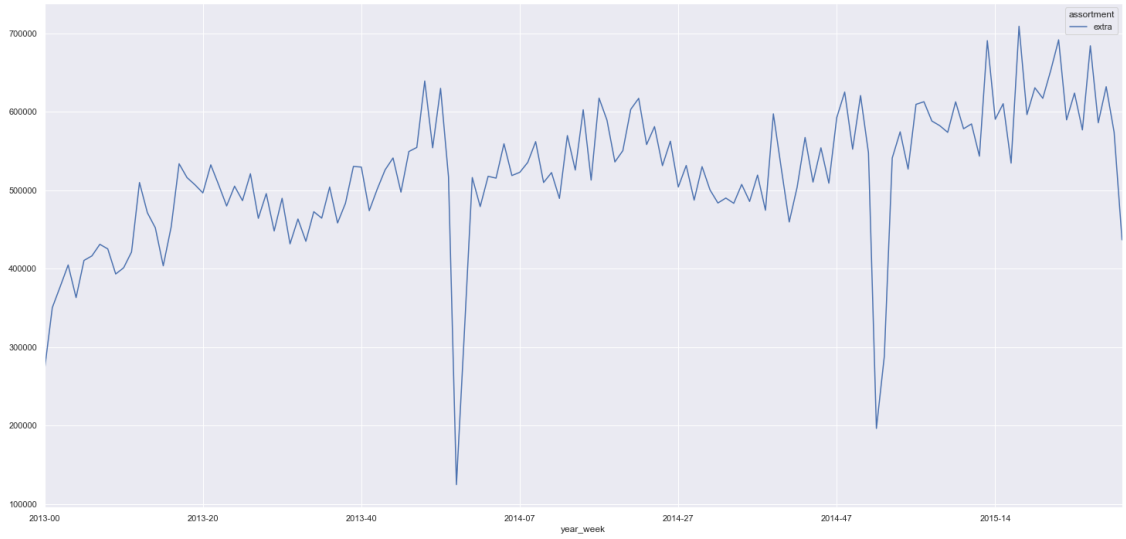
[29]:
```
aux1 = df4[['assortment', 'sales']].groupby( 'assortment' ).sum().reset_index()
sns.barplot( x='assortment', y='sales', data=aux1 );

aux2 = df4[['year_week', 'assortment', 'sales']].groupby(␣
 ↪['year_week','assortment'] ).sum().reset_index()
aux2.pivot( index='year_week', columns='assortment', values='sales' ).plot()

aux3 = aux2[aux2['assortment'] == 'extra']
```

```
aux3.pivot( index='year_week', columns='assortment', values='sales' ).plot()
```

[29]: <matplotlib.axes._subplots.AxesSubplot at 0x171f91a30>



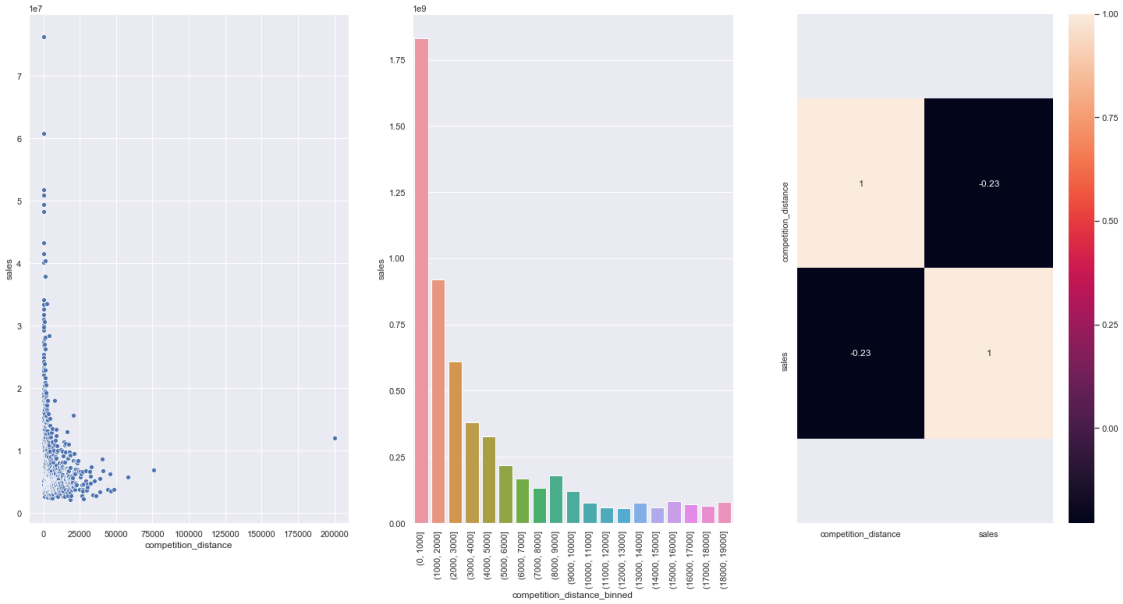




17.0.2 H2. Lojas com competidores mais próximos deveriam vender menos.

FALSA Lojas com COMPETIDORES MAIS PROXIMOS vendem MAIS.

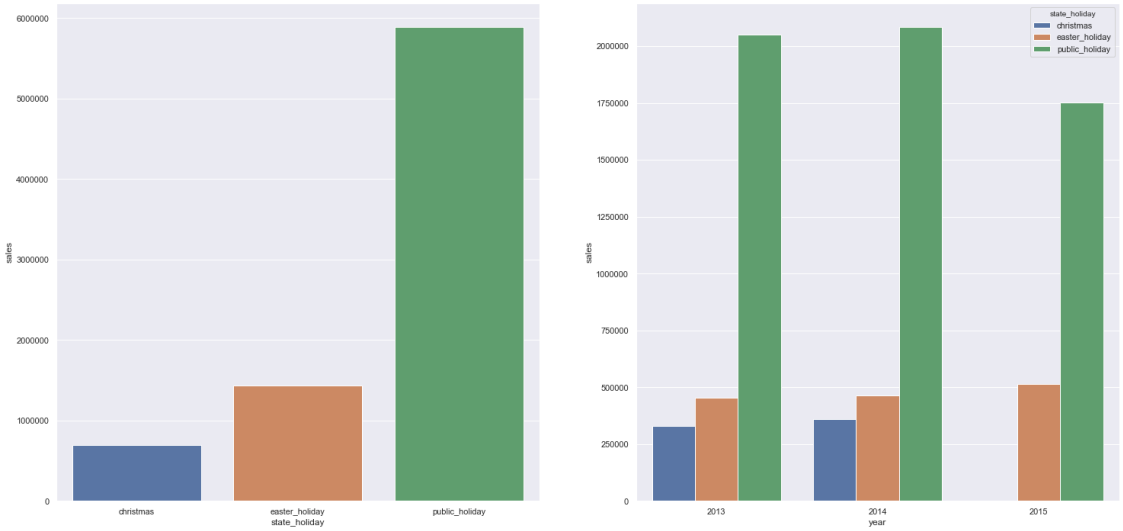
[30]:
```
aux1 = df4[['competition_distance', 'sales']].groupby( 'competition_distance' ).
 ↪sum().reset_index()

plt.subplot( 1, 3, 1 )
sns.scatterplot( x ='competition_distance', y='sales', data=aux1 );

plt.subplot( 1, 3, 2 )
bins = list( np.arange( 0, 20000, 1000) )
aux1['competition_distance_binned'] = pd.cut( aux1['competition_distance'],␣
 ↪bins=bins )
aux2 = aux1[['competition_distance_binned', 'sales']].groupby(␣
 ↪'competition_distance_binned' ).sum().reset_index()
sns.barplot( x='competition_distance_binned', y='sales', data=aux2 );
plt.xticks( rotation=90 );

plt.subplot( 1, 3, 3 )
x = sns.heatmap( aux1.corr( method='pearson' ), annot=True );
bottom, top = x.get_ylim()
x.set_ylim( bottom+0.5, top-0.5 );
```



17.0.3 H8. Lojas abertas durante o feriado de Natal deveriam vender mais.

FALSA Lojas abertas durante o feriado do Natal vendem menos.

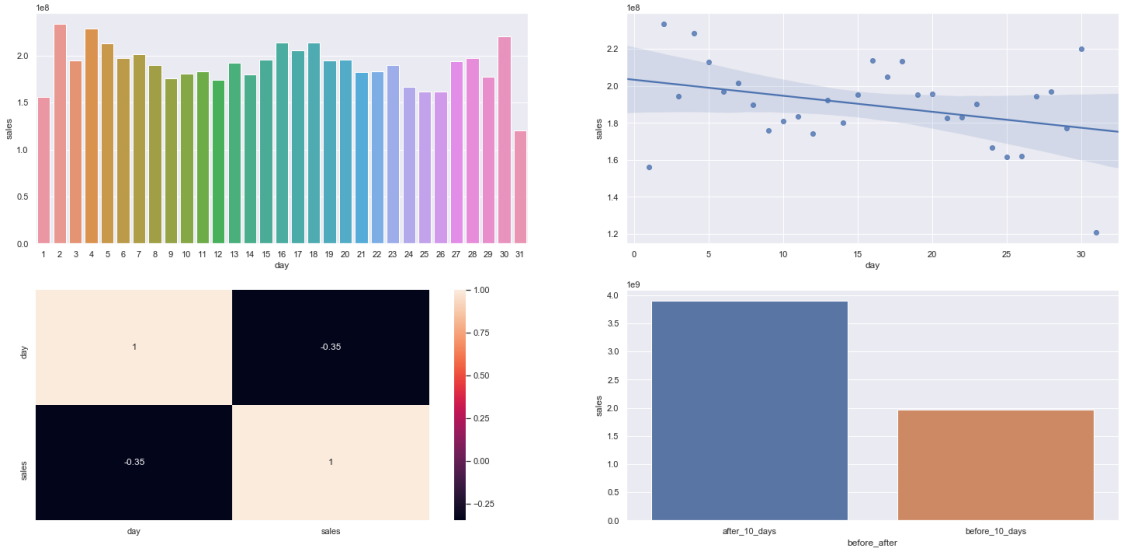
[35]:
```
aux = df4[df4['state_holiday'] != 'regular_day']

plt.subplot( 1, 2, 1 )
aux1 = aux[['state_holiday', 'sales']].groupby( 'state_holiday' ).sum().
 ↪reset_index()
sns.barplot( x='state_holiday', y='sales', data=aux1 );

plt.subplot( 1, 2, 2 )
aux2 = aux[['year', 'state_holiday', 'sales']].groupby( ['year',␣
 ↪'state_holiday'] ).sum().reset_index()
sns.barplot( x='year', y='sales', hue='state_holiday', data=aux2 );
```



17.0.4 H11. Lojas deveriam vender mais depois do dia 10 de cada mês.

VERDADEIRA Lojas vendem mais depois do dia 10 de cada mes.

[38]:
```
aux1 = df4[['day', 'sales']].groupby( 'day' ).sum().reset_index()

plt.subplot( 2, 2, 1 )
sns.barplot( x='day', y='sales', data=aux1 );

plt.subplot( 2, 2, 2 )
sns.regplot( x='day', y='sales', data=aux1 );

plt.subplot( 2, 2, 3 )
sns.heatmap( aux1.corr( method='pearson' ), annot=True );

aux1['before_after'] = aux1['day'].apply( lambda x: 'before_10_days' if x <= 10␣
 ↪else 'after_10_days' )
aux2 =aux1[['before_after', 'sales']].groupby( 'before_after' ).sum().
 ↪reset_index()

plt.subplot( 2, 2, 4 )
sns.barplot( x='before_after', y='sales', data=aux2 );
```



18 Resumo das Hipoteses

[42]:
```
tab =[['Hipoteses', 'Conclusao', 'Relevancia'],
      ['H1', 'Falsa', 'Baixa'],
      ['H2', 'Falsa', 'Media'],
      ['H3', 'Falsa', 'Media'],
      ['H4', 'Falsa', 'Baixa'],
      ['H5', '-', '-'],
      ['H7', 'Falsa', 'Baixa'],
      ['H8', 'Falsa', 'Media'],
      ['H9', 'Falsa', 'Alta'],
      ['H10', 'Falsa', 'Alta'],
      ['H11', 'Verdadeira', 'Alta'],
      ['H12', 'Verdadeira', 'Alta'],
      ['H13', 'Verdadeira', 'Baixa'],
     ]
print( tabulate( tab, headers='firstrow' ) )
```

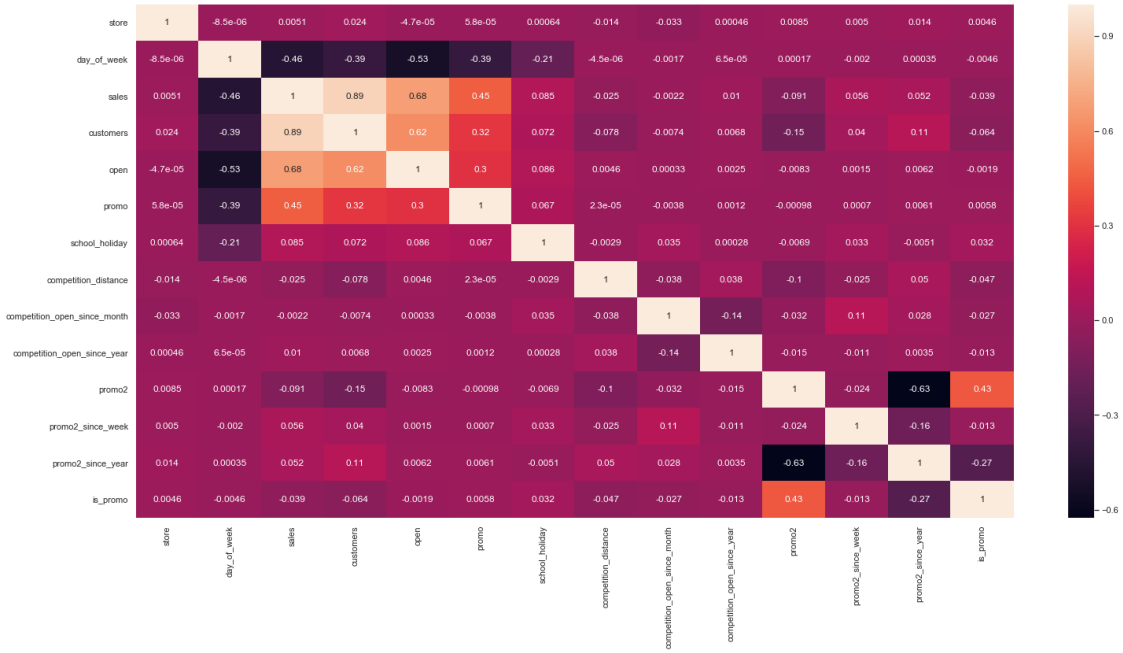
```
Hipoteses    Conclusao    Relevancia
-----------  -----------  ------------
H1           Falsa        Baixa
H2           Falsa        Media
H3           Falsa        Media
H4           Falsa        Baixa
H5           -            -
H7           Falsa        Baixa
H8           Falsa        Media
H9           Falsa        Alta
```

```
H10          Falsa        Alta
H11          Verdadeira   Alta
H12          Verdadeira   Alta
H13          Verdadeira   Baixa
```

19 Analise Multivariada

20 Numerical Attributes

[43]:
```
correlation = num_attributes.corr( method='pearson' )
sns.heatmap( correlation, annot=True );
```

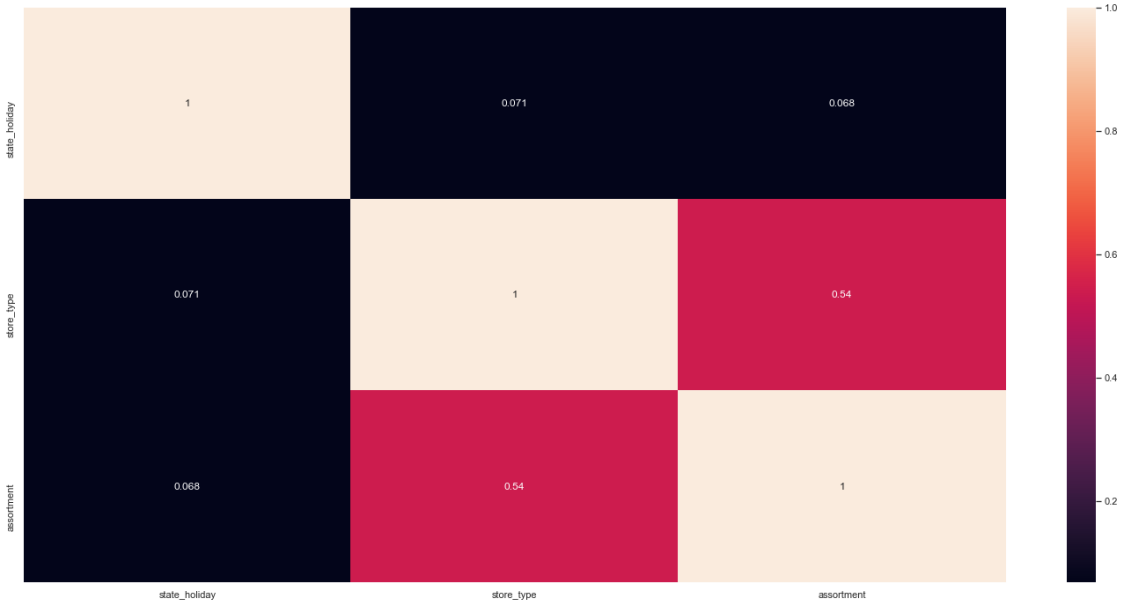


21 Categorical Attributes

[44]:
```
# only categorical data
a = df4.select_dtypes( include='object' )

# Calculate cramer V
a1 = cramer_v( a['state_holiday'], a['state_holiday'] )
a2 = cramer_v( a['state_holiday'], a['store_type'] )
a3 = cramer_v( a['state_holiday'], a['assortment'] )

a4 = cramer_v( a['store_type'], a['state_holiday'] )
a5 = cramer_v( a['store_type'], a['store_type'] )
```

```
a6 = cramer_v( a['store_type'], a['assortment'] )

a7 = cramer_v( a['assortment'], a['state_holiday'] )
a8 = cramer_v( a['assortment'], a['store_type'] )
a9 = cramer_v( a['assortment'], a['assortment'] )

# Final dataset
d = pd.DataFrame( {'state_holiday': [a1, a2, a3],
                   'store_type': [a4, a5, a6],
                   'assortment': [a7, a8, a9]  })
d = d.set_index( d.columns )

sns.heatmap( d, annot=True )
```

[44]: <matplotlib.axes._subplots.AxesSubplot at 0x122d1ad30>



22 MACHINE LEARNING MODELLING

23 Compare Model's Performance

[67]:
```
modelling_result_cv = pd.concat( [lr_result_cv, lrr_result_cv, rf_result_cv,␣
 ↪xgb_result_cv] )
modelling_result_cv
```

[67]:
```
                Model Name            MAE CV        MAPE CV              RMSE
CV
```

```
0        Linear Regression  2081.73 +/- 295.63   0.3 +/- 0.02  2952.52 +/-
468.37
0                    Lasso   2116.38 +/- 341.5  0.29 +/- 0.01  3057.75 +/-
504.26
0  Random Forest Regressor    837.68 +/- 219.1  0.12 +/- 0.02  1256.08 +/-
320.36
0        XGBoost Regressor  1030.28 +/- 167.19  0.14 +/- 0.02  1478.26 +/-
229.79
```

24 4. Conclusão & Demonstração

25 TRADUCAO E INTERPRETACAO DO ERRO

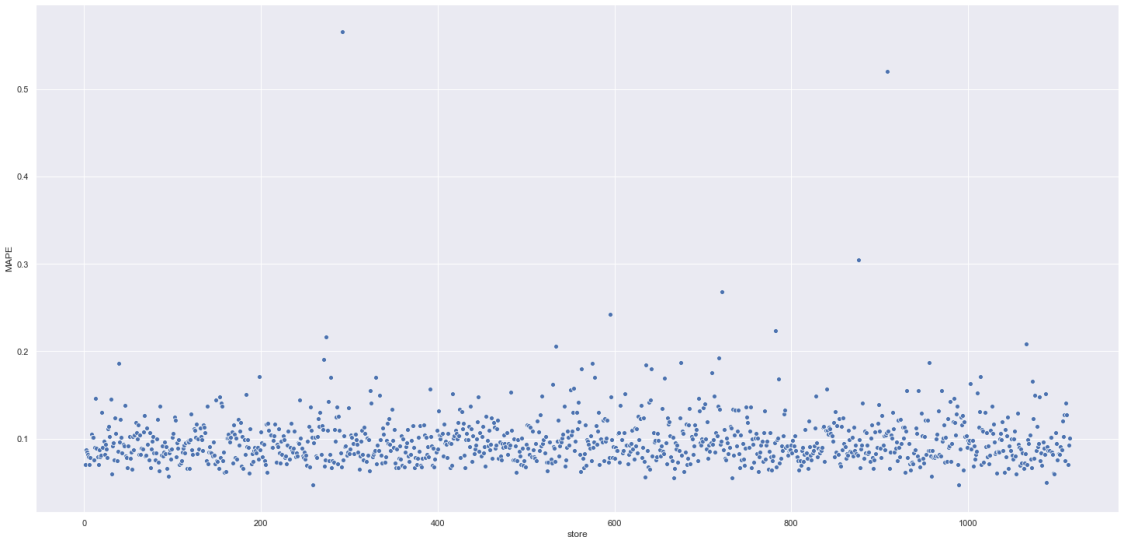
26 Business Performance

[446]:
```
df92.sort_values( 'MAPE', ascending=False ).head()
```

[446]:
```
     store    predictions  worst_scenario  best_scenario          MAE  \
291    292  104033.078125   100714.973723  107351.182527  3318.104402
908    909  238233.875000   230573.337190  245894.412810  7660.537810
875    876  203030.156250   199110.952435  206949.360065  3919.203815
721    722  353005.781250   351013.625224  354997.937276  1992.156026
594    595  400883.625000   397415.263170  404351.986830  3468.361830

         MAPE
291  0.565828
908  0.520433
875  0.305099
721  0.268338
594  0.242192
```

[448]:
```
sns.scatterplot( x='store', y='MAPE', data=df92 )
```

[448]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x16a890280>
```



27 Total Performance

[455]:
```
df93 = df92[['predictions', 'worst_scenario', 'best_scenario']].apply( lambda x:
 ↪ np.sum( x ), axis=0 ).reset_index().rename( columns={'index': 'Scenario', 0:
 ↪'Values'} )
df93['Values'] = df93['Values'].map( 'R${:,.2f}'.format )
df93
```

[455]:

	Scenario	Values
0	predictions	R$285,860,497.77
1	worst_scenario	R$285,115,015.71
2	best_scenario	R$286,605,979.84

28 Machine Learning Performance

[459]:
```
plt.subplot( 2, 2, 1 )
sns.lineplot( x='date', y='sales', data=df9, label='SALES' )
sns.lineplot( x='date', y='predictions', data=df9, label='PREDICTIONS' )

plt.subplot( 2, 2, 2 )
sns.lineplot( x='date', y='error_rate', data=df9 )
plt.axhline( 1, linestyle='--')

plt.subplot( 2, 2, 3 )
sns.distplot( df9['error'] )

plt.subplot( 2, 2, 4 )
```

```
sns.scatterplot( df9['predictions'], df9['error'] )
```

[459]: <matplotlib.axes._subplots.AxesSubplot at 0x1689cf700>

29 5. Próximos Passos

- Workshop do Modelo para os Business Users
- Coletar Feedbacks sobre a Usabilidade
- Aumentar em 10% a Acurácia do Modelo

30 Q & A

31 Muito Obrigado!