Data Science - Regressão Linear

Conhecendo o Dataset

Importando bibliotecas

https://matplotlib.org/

https://pandas.pydata.org/

http://www.numpy.org/

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

Bibliotecas opcionais

https://docs.python.org/3/library/warnings.html

In [ ]:
```

```

O Dataset e o Projeto

Fonte: https://www.kaggle.com/dongeorge/beer-consumption-sao-paulo

Descrição:

> A cerveja é uma das bebidas mais democráticas e consumidas no mundo. Não sem razão, é perfeito para quase todas as situações, desde o happy hour até grandes festas de casamento.
>
> O objetivo deste treinamento será estimar um modelo de **Machine Learning** utilizando a técnica de **Regressão Linear** para demonstrar os impactos das variáveis disponibilizadas neste dataset sobre o consumo de cerveja (Y). No final do projeto teremos um modelo de previsão para o consumo médio de cerveja segundo os inputs de um conjunto de variáveis (X's).

Os dados (amostra) foram coletados em São Paulo - Brasil, em uma área universitária, onde existem algumas festas com grupos de alunos de 18 a 28 anos de idade (média).

Dados:

- **data** - Data
- **temp_media** - Temperatura Média (°C)
- **temp_min** - Temperatura Mínima (°C)
- **temp_max** - Temperatura Máxima (°C)
- **chuva** - Precipitação (mm)
- **fds** - Final de Semana (1 = Sim; 0 = Não)
- **consumo** - Consumo de Cerveja (litros)

Leitura dos dados

In [2]:
```
dados = pd.read_csv("../Dados/Consumo_cerveja.csv", sep = ";")
```

Visualizar os dados

In [3]:
```
dados.head(5)
```

Out[3]:

	data	temp_media	temp_min	temp_max	chuva	fds	consumo
0	01/01/2015	27.30	23.9	32.5	0.0	0	25461
1	02/01/2015	27.02	24.5	33.5	0.0	0	28972
2	03/01/2015	24.82	22.4	29.9	0.0	1	30814
3	04/01/2015	23.98	21.5	28.6	1.2	1	29799
4	05/01/2015	23.82	21.0	28.3	0.0	0	28900

Verificando o tamanho do dataset

In [4]:
```
dados.shape[0]
```

Out[4]:
```
365
```

Análises Preliminares

Estatísticas descritivas

In [5]:

```
dados.describe().round(2)
```

Out[5]:

	temp_media	temp_min	temp_max	chuva	fds	consumo
count	365.00	365.00	365.00	365.00	365.00	365.00
mean	21.23	17.46	26.61	5.20	0.28	25401.37
std	3.18	2.83	4.32	12.42	0.45	4399.14
min	12.90	10.60	14.50	0.00	0.00	14343.00
25%	19.02	15.30	23.80	0.00	0.00	22008.00
50%	21.38	17.90	26.90	0.00	0.00	24867.00
75%	23.28	19.60	29.40	3.20	1.00	28631.00
max	28.86	24.50	36.50	94.80	1.00	37937.00

Matriz de correlação

O **coeficiente de correlação** é uma medida de associação linear entre duas variáveis e situa-se entre **-1** e **+1** sendo que **-1** indica associação negativa perfeita e **+1** indica associação positiva perfeita.

In [6]:
```
dados.corr().round(4)
```

Out[6]:

	temp_media	temp_min	temp_max	chuva	fds	consumo
temp_media	1.0000	0.8628	0.9225	0.0244	-0.0508	0.5746
temp_min	0.8628	1.0000	0.6729	0.0986	-0.0595	0.3925
temp_max	0.9225	0.6729	1.0000	-0.0493	-0.0403	0.6427
chuva	0.0244	0.0986	-0.0493	1.0000	0.0016	-0.1938
fds	-0.0508	-0.0595	-0.0403	0.0016	1.0000	0.5060
consumo	0.5746	0.3925	0.6427	-0.1938	0.5060	1.0000

Comportamento da Variável Dependente (Y)

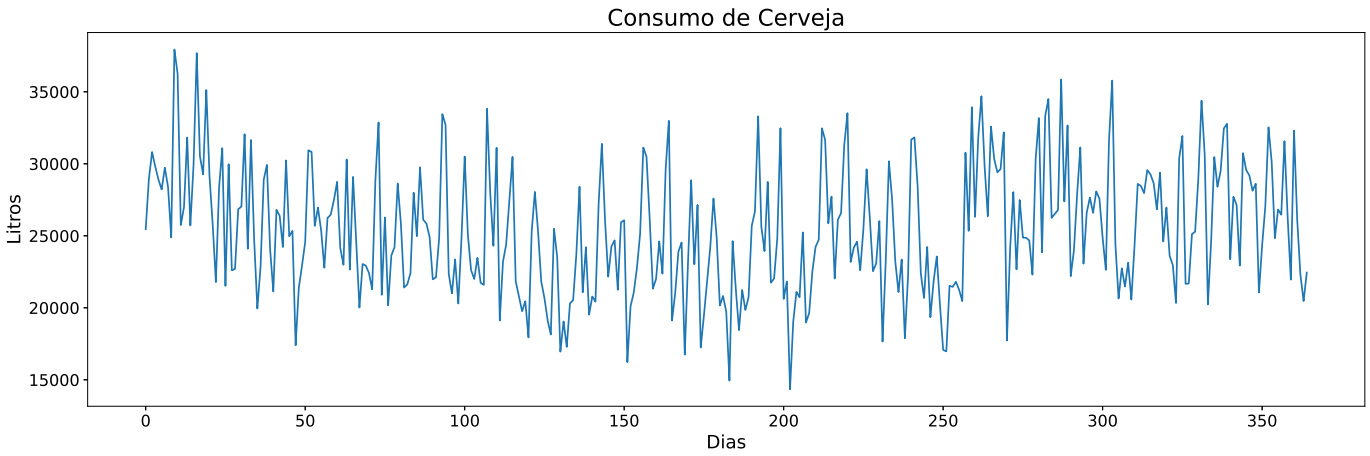
Análises gráficas

Plotando a variável *dependente* (y)

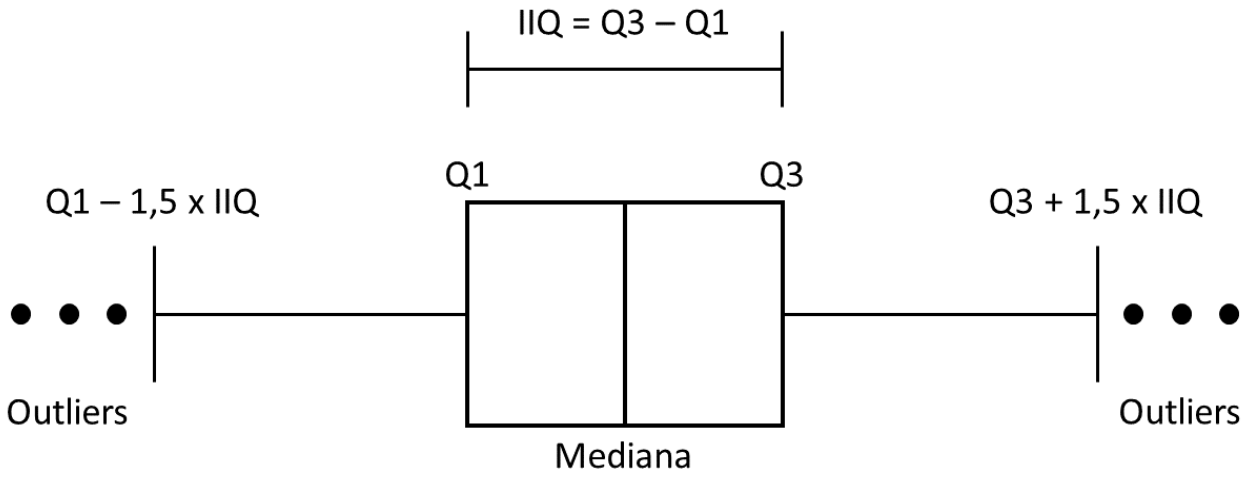
https://pandas.pydata.org/pandas-docs/stable/visualization.html

In [7]:
```
fig, ax = plt.subplots(figsize = (20, 6))

ax.set_title("Consumo de Cerveja", fontsize = 20)
ax.set_ylabel("Litros", fontsize = 16)
ax.set_xlabel("Dias", fontsize = 16)
ax = dados.consumo.plot(fontsize = 14)
```



Box Plot



Box-plot

Importando biblioteca seaborn

https://seaborn.pydata.org/

> O Seaborn é uma biblioteca Python de visualização de dados baseada no matplotlib. Ela fornece uma interface de alto nível para desenhar gráficos estatísticos.

In [8]:
```
import seaborn as sns
```

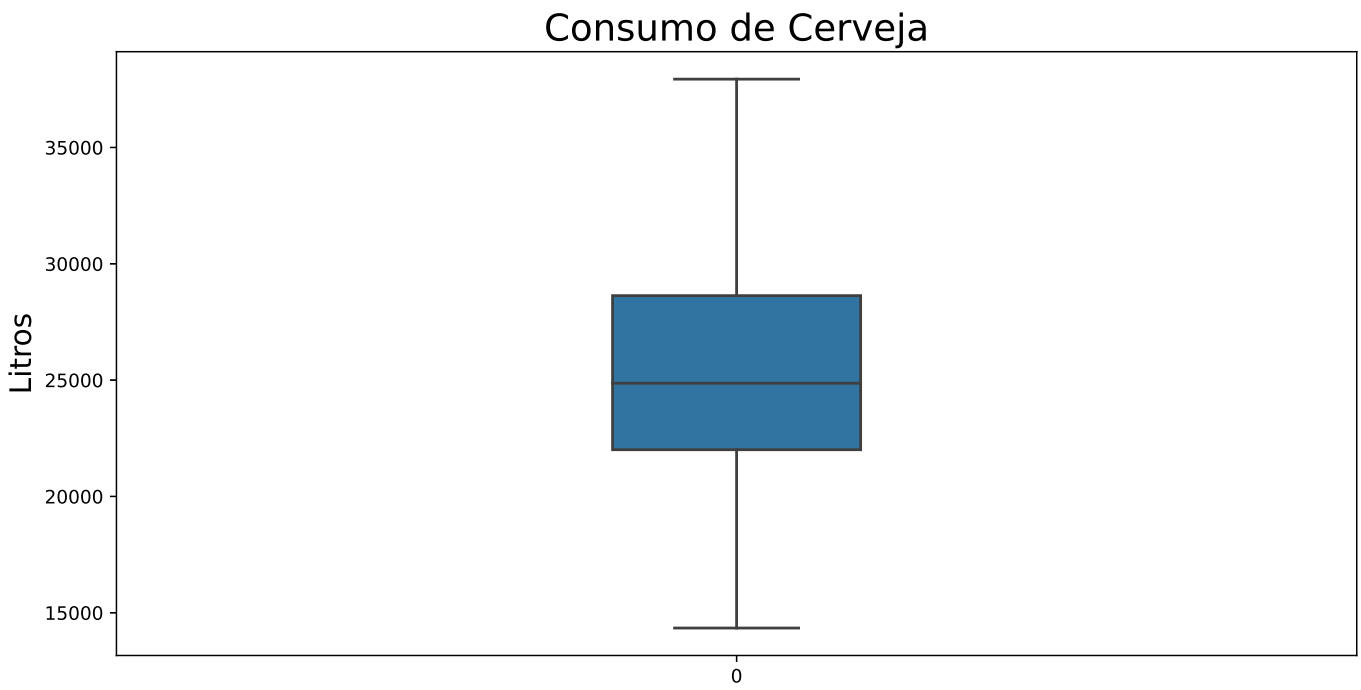
Box plot da variável *dependente* (y)

https://seaborn.pydata.org/generated/seaborn.boxplot.html?highlight=boxplot#seaborn.boxplot

In [9]:
```
ax = sns.boxplot(data = dados.consumo, orient = 'v', width = 0.2)
ax.figure.set_size_inches(12, 6)
ax.set_title("Consumo de Cerveja", fontsize = 20)
```

```
ax.set_ylabel("Litros", fontsize = 16)
ax
```

Out[9]: <AxesSubplot:title={'center':'Consumo de Cerveja'}, ylabel='Litros'>



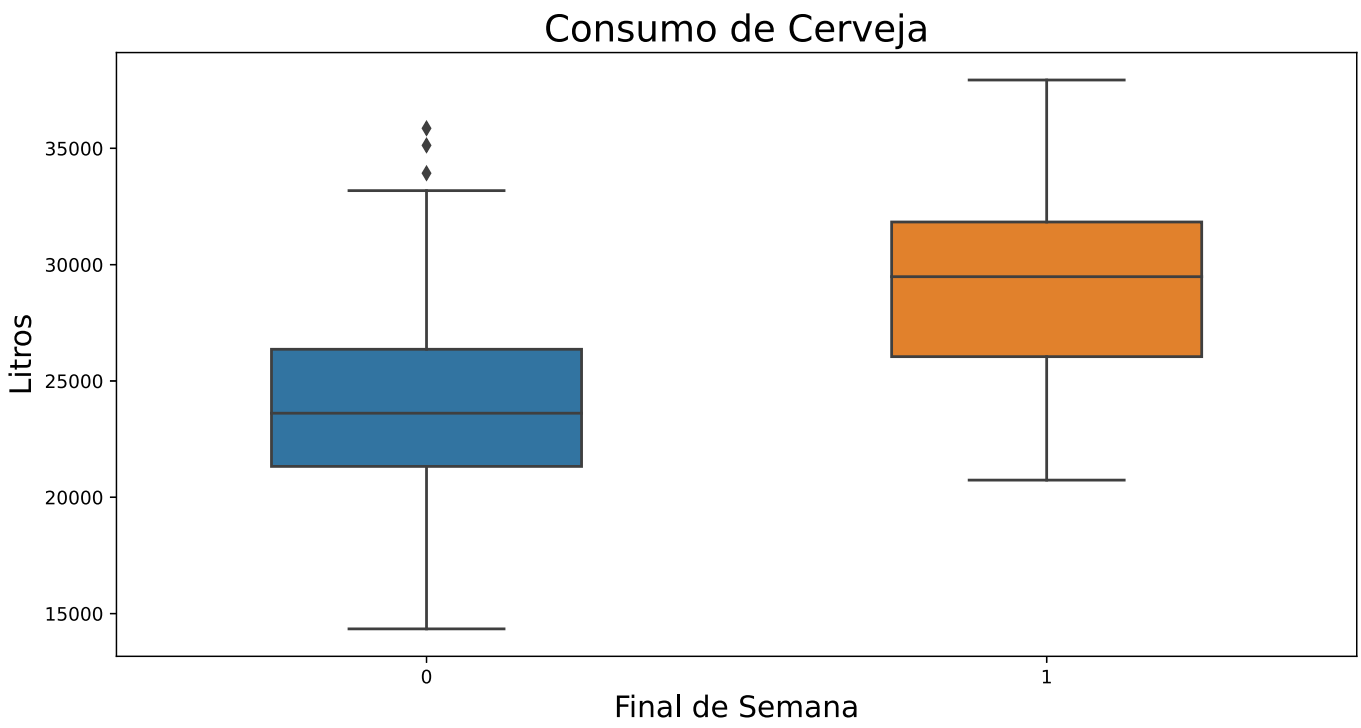
Box Plot com Duas Variáveis

Investigando a variável *dependente* (y) segundo determinada característica

In [10]:

```
ax = sns.boxplot(y = 'consumo', x = 'fds', data = dados, orient = 'v', width = 0.5)
ax.figure.set_size_inches(12, 6)
ax.set_title("Consumo de Cerveja", fontsize = 20)
ax.set_ylabel("Litros", fontsize = 16)
ax.set_xlabel("Final de Semana", fontsize = 16)
ax
```

Out[10]: <AxesSubplot:title={'center':'Consumo de Cerveja'}, xlabel='Final de Semana', ylabel='Litros'>



Configurações de estilo e cor da biblioteca *seaborn*

Controle de estilo

> ### API
>
> https://seaborn.pydata.org/api.html#style-api
>
> ### Tutorial
>
> https://seaborn.pydata.org/tutorial/aesthetics.html#aesthetics-tutorial

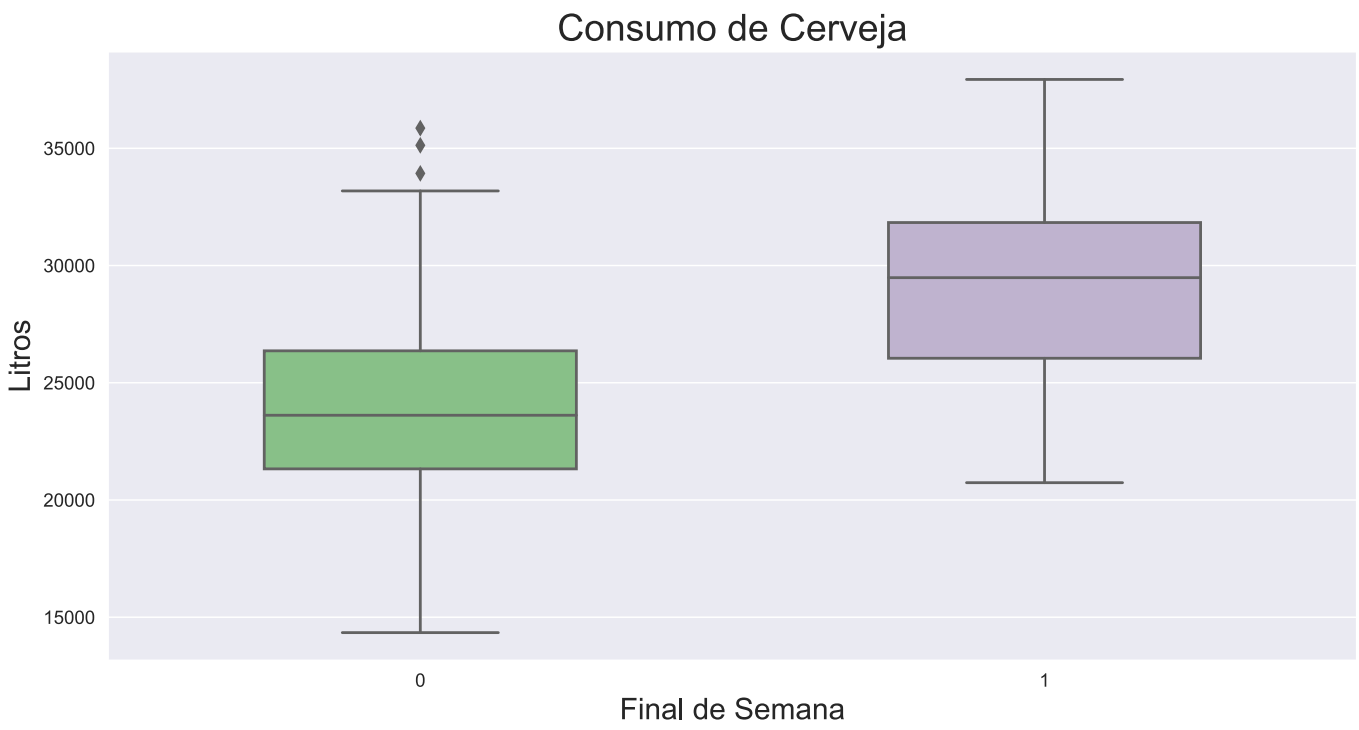
Paleta de cores

> ### API
>
> https://seaborn.pydata.org/api.html#palette-api
>
> ### Tutorial
>
> https://seaborn.pydata.org/tutorial/color_palettes.html#palette-tutorial

In [11]:
```
sns.set_palette("Accent")
sns.set_style("darkgrid")
```

In [12]:
```
ax = sns.boxplot(y = 'consumo', x = 'fds', data = dados, orient = 'v', width = 0.5)
ax.figure.set_size_inches(12, 6)
ax.set_title("Consumo de Cerveja", fontsize = 20)
ax.set_ylabel("Litros", fontsize = 16)
ax.set_xlabel("Final de Semana", fontsize = 16)
ax
```

Out[12]:
```
<AxesSubplot:title={'center':'Consumo de Cerveja'}, xlabel='Final de Semana', ylabel='Litros'>
```



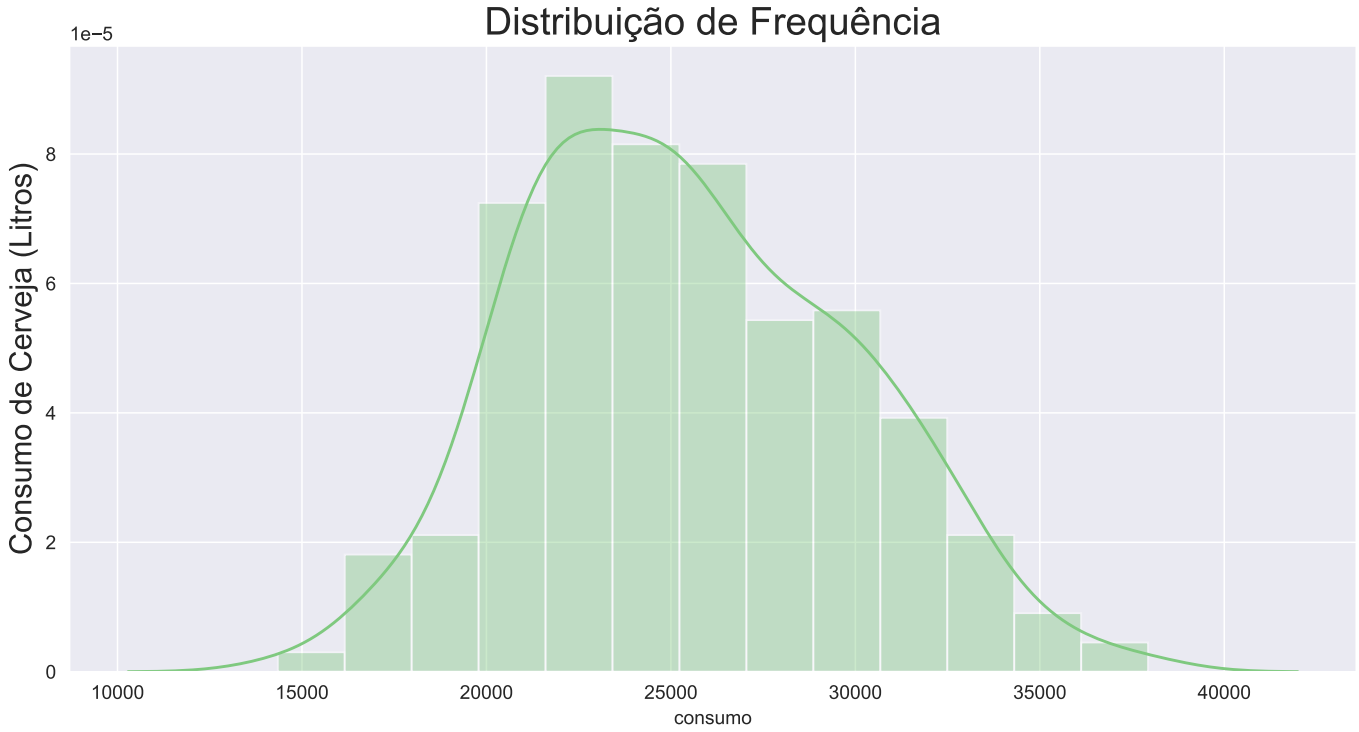
Distribuição de Frequências

Distribuição de frequências da variável *dependente* (y)

https://seaborn.pydata.org/generated/seaborn.distplot.html?highlight=distplot#seaborn.distplot

In [13]:
```
ax = sns.distplot(dados.consumo)
ax.figure.set_size_inches(12, 6)
ax.set_title("Distribuição de Frequência", fontsize = 20)
ax.set_ylabel("Consumo de Cerveja (Litros)", fontsize = 16)
ax
```

Out[13]:
```
<AxesSubplot:title={'center':'Distribuição de Frequência'}, xlabel='consumo', ylabel='Consumo de Cerveja (Litros)'>
```



Variável Dependente X Variáveis Explicativas

(pairplot)

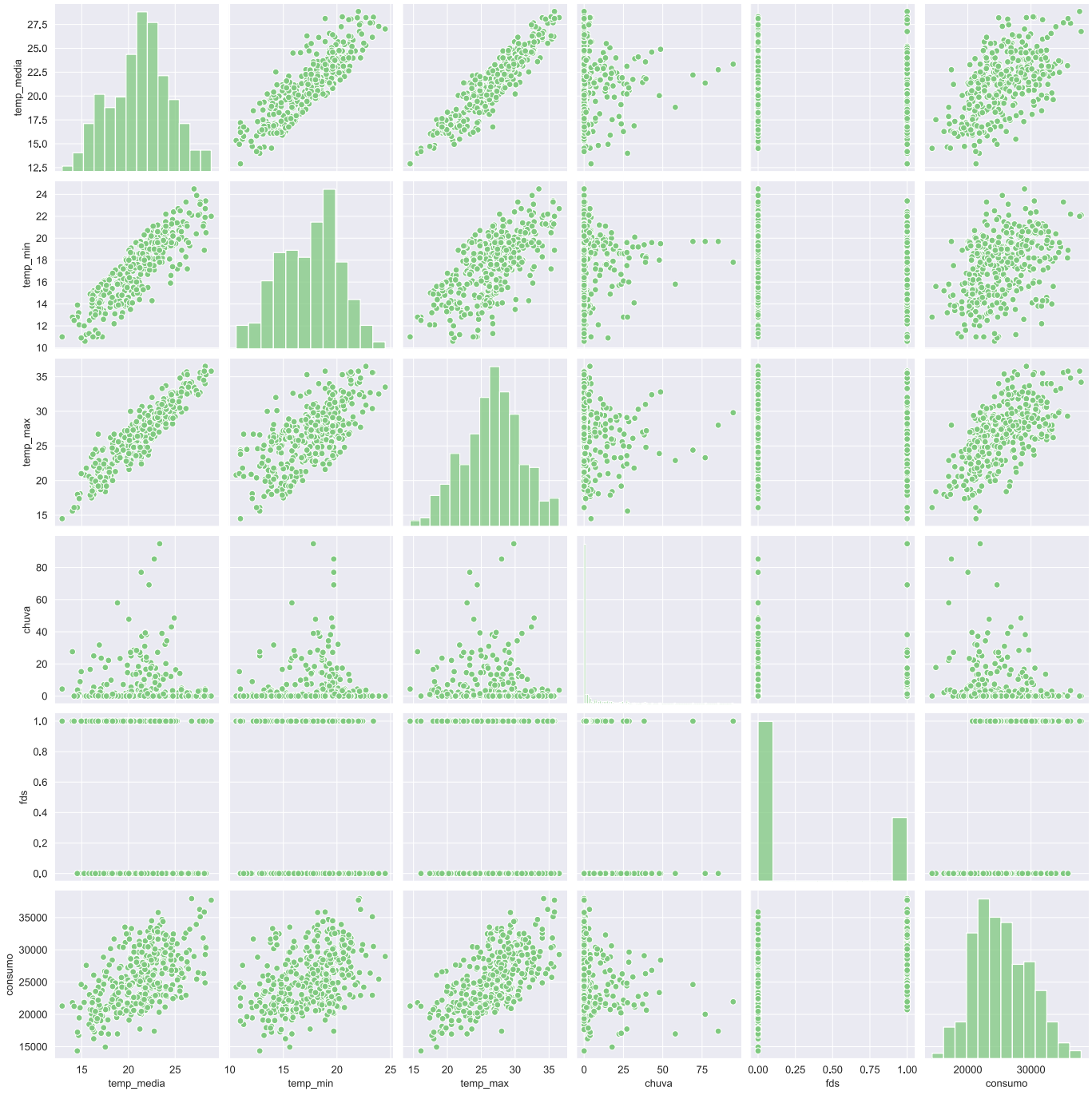
Gráficos de dispersão entre as variáveis do dataset

seaborn.pairplot

https://seaborn.pydata.org/generated/seaborn.pairplot.html?highlight=pairplot#seaborn.pairplot

Plota o relacionamento entre pares de variáveis em um dataset.

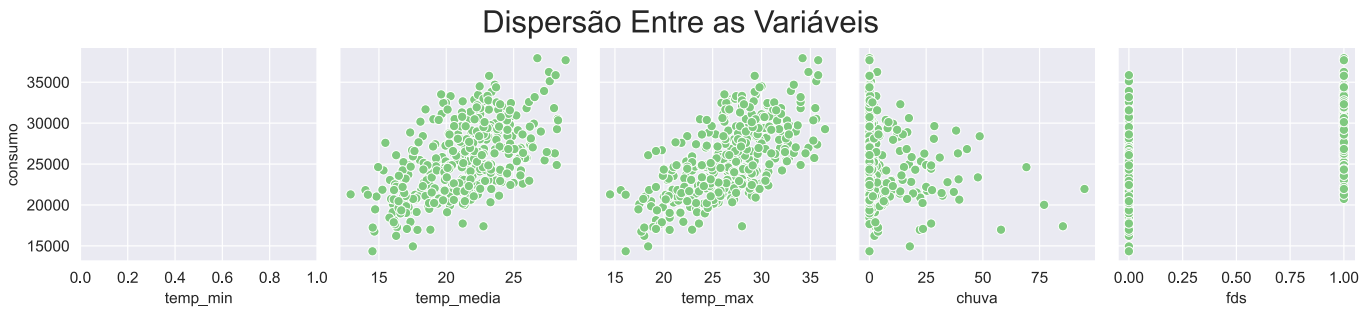
In [14]:
```
ax = sns.pairplot(dados)
```



Plotando o pairplot fixando somente uma variável no eixo y

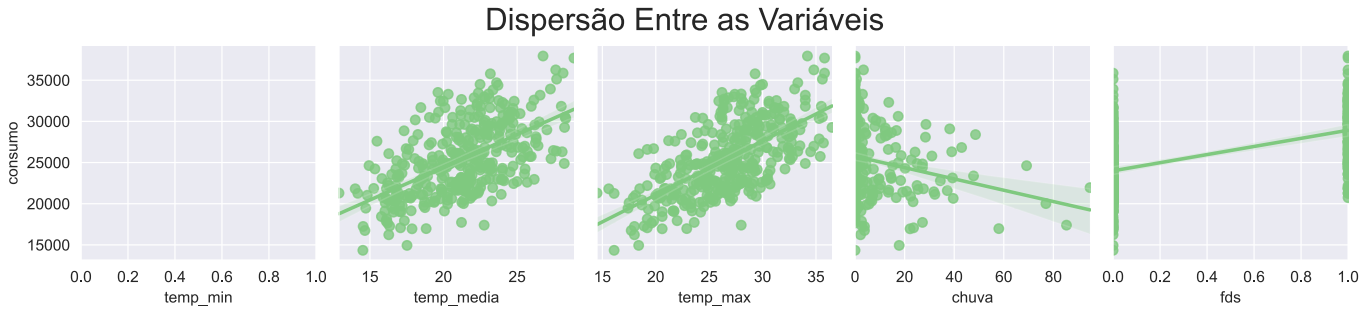
In [15]:
```
ax = sns.pairplot(dados, y_vars = 'consumo', x_vars = ['temp_min', 'temp_media', 'temp_max', 'c
ax.fig.suptitle("Dispersão Entre as Variáveis", fontsize = 20, y = 1.1)
ax
```

Out[15]: <seaborn.axisgrid.PairGrid at 0xcf5e310>



In [16]:
```
ax = sns.pairplot(dados, y_vars = 'consumo', x_vars = ['temp_min', 'temp_media', 'temp_max', 'c
ax.fig.suptitle("Dispersão Entre as Variáveis", fontsize = 20, y = 1.1)
ax
```

Out[16]: <seaborn.axisgrid.PairGrid at 0xec846d0>



Variável Dependente X Variáveis Explicativas (jointplot)

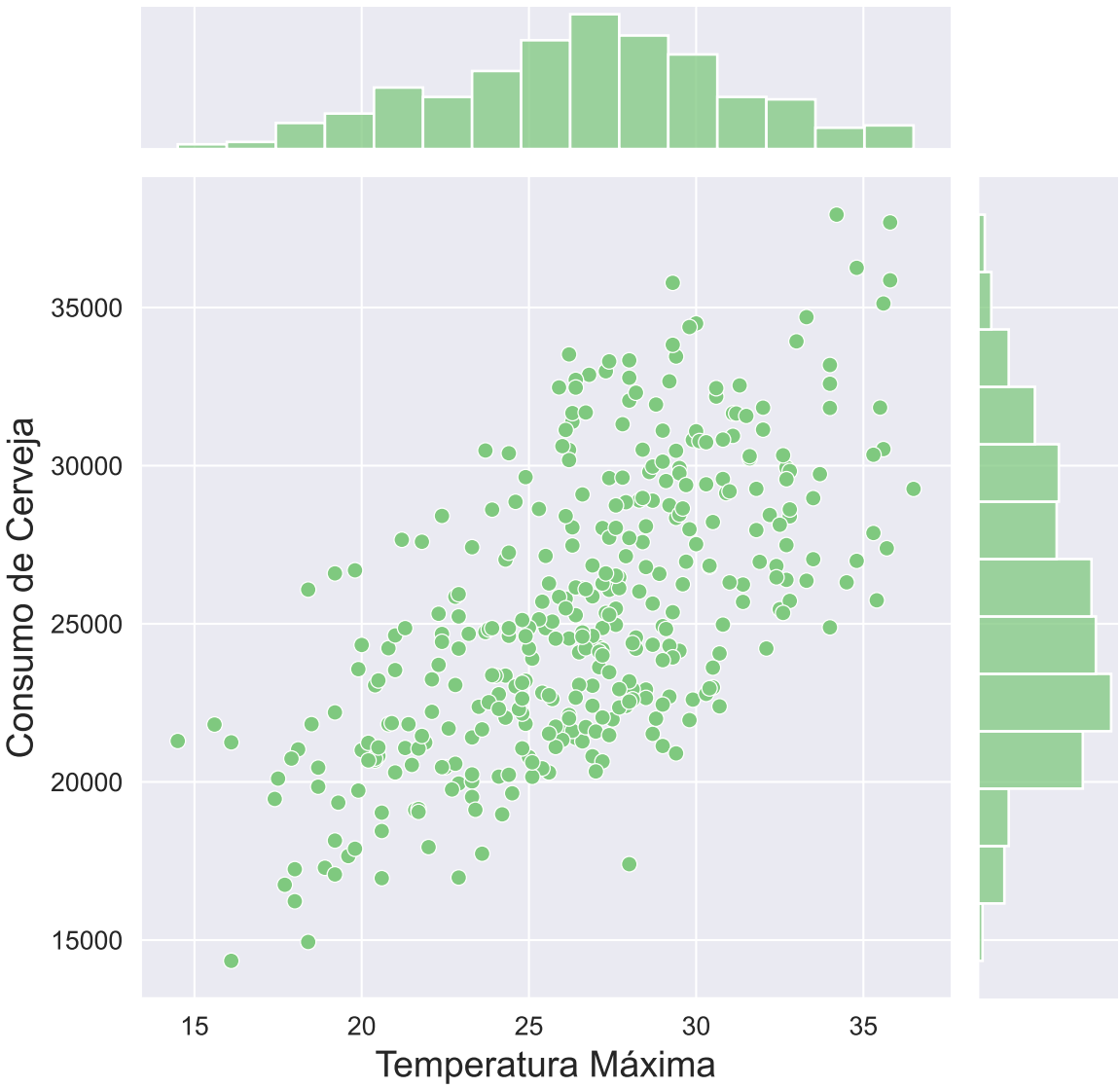
seaborn.jointplot

https://seaborn.pydata.org/generated/seaborn.jointplot.html?highlight=jointplot#seaborn.jointplot

Plota o relacionamento entre duas variáveis e suas respectivas distribuições de frequência.

In [17]:
```
ax = sns.jointplot(x = 'temp_max', y = 'consumo', data = dados)
ax.fig.suptitle("Dispersão - Consumo x Temperatura", fontsize = 18, y = 1.05)
ax.set_axis_labels("Temperatura Máxima", "Consumo de Cerveja", fontsize = 14)
ax
```

Out[17]: <seaborn.axisgrid.JointGrid at 0xefe1670>



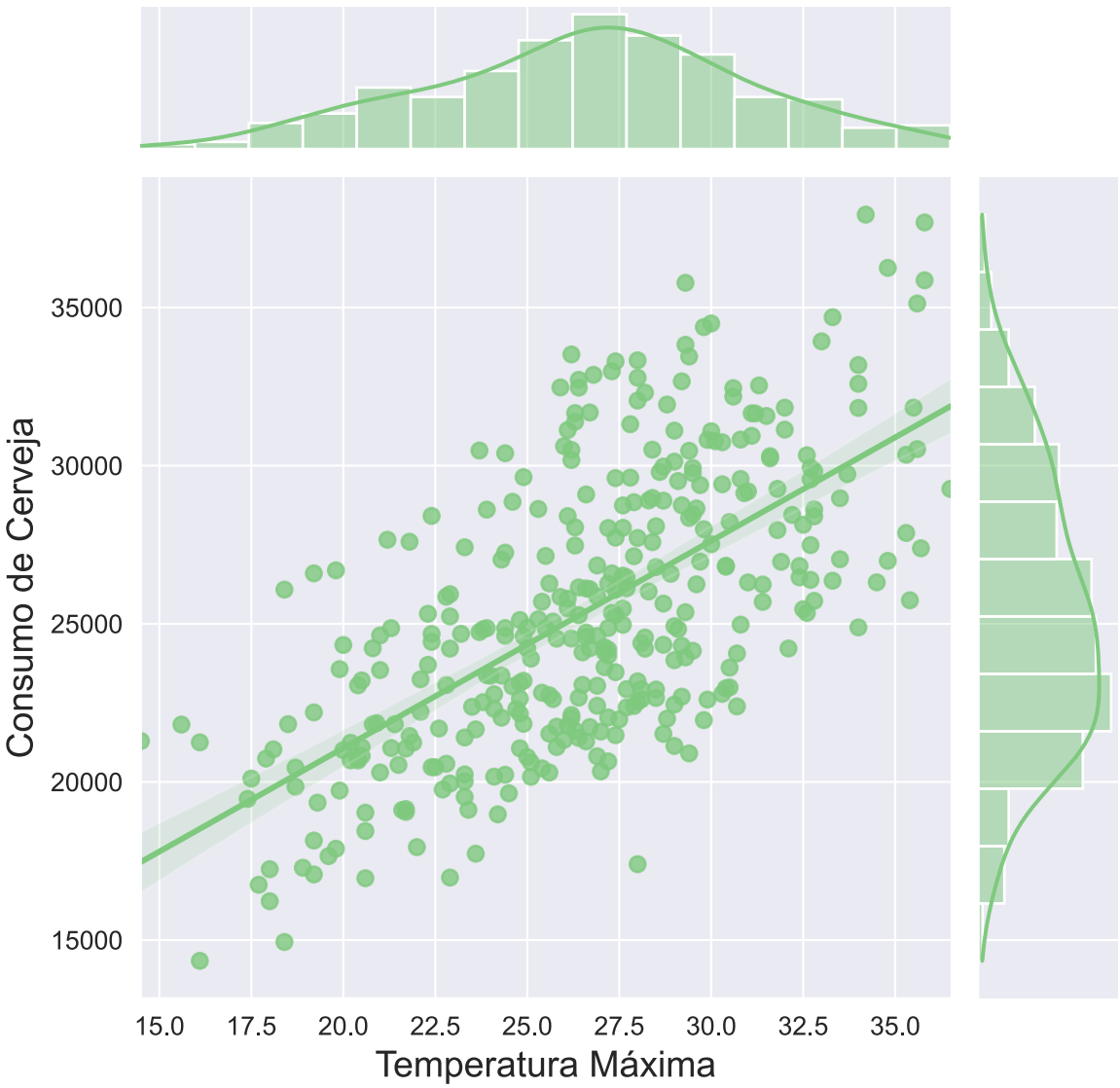
Plotando um jointplot com a reta de regressão estimada

In [18]:

```
ax = sns.jointplot(x = 'temp_max', y = 'consumo', data = dados, kind = 'reg')
ax.fig.suptitle("Dispersão - Consumo x Temperatura", fontsize = 18, y = 1.05)
ax.set_axis_labels("Temperatura Máxima", "Consumo de Cerveja", fontsize = 14)
ax
```

Out[18]:

```
<seaborn.axisgrid.JointGrid at 0xf522ca0>
```



Variável Dependente X Variáveis Explicativas (lmplot)

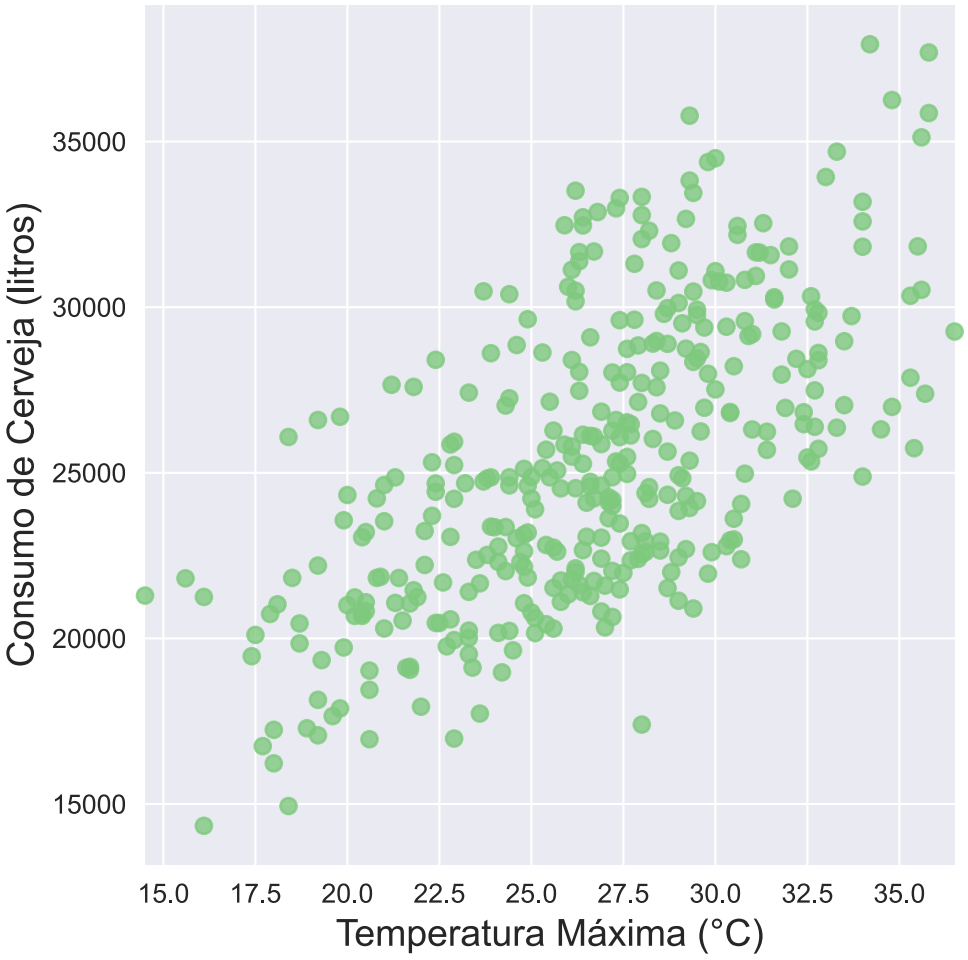
seaborn.lmplot

https://seaborn.pydata.org/generated/seaborn.lmplot.html?highlight=lmplot#seaborn.lmplot

Plota a reta de regressão entre duas variáveis juntamente com a dispersão entre elas.

In [19]:
```
ax = sns.lmplot(x = 'temp_max', y = 'consumo', data = dados)
ax.fig.suptitle("Reta de Regressão - Consumo X Temperatura", fontsize = 16, y = 1.02)
ax.set_xlabels("Temperatura Máxima (°C)", fontsize = 14)
ax.set_ylabels("Consumo de Cerveja (litros)", fontsize = 14)
ax
```

Out[19]:
```
<seaborn.axisgrid.FacetGrid at 0x10a618e0>
```


Consumo de Cerveja (litros)
35000
30000
25000
20000
15000
15.0
17.5
20.0
22.5
25.0
27.5
30.0
32.5
35.0
Temperatura Máxima (°C)