Análise Exploratória de Dados.

Unidade II – Análise Univariada

INSTITUTO DE COMPUTAÇÃO

UFMT

Introdução a Análise Univariada

- O que é Análise Univariada.
  - É a análise individual dos atributos do conjunto de dados.
  - Um atributo é uma coluna do conjunto de dados.
  - Para essa análise pode ser utilizada estatística descritiva (média, mediana, moda, etc).
  - Ou podem ser utilizados gráficos (Histogramas, Gráfico de Barras, Gráficos de Densidade, Gráficos de distribuição, etc).

Medidas estatísticas básicas

- Média – É o valor médio dos dados. É afetada pela presença de valores extremos.
- Mediana – É o ponto central quando os dados estão ordenados. Se a quantidade de amostras é par, a mediana é a média dos dois valores centrais. Não é afetada por valores extremos.
- Moda – Descreve o valor com maior frequência. É possível ter mais de um moda.

Medidas estatísticas básicas

- *import pandas as pd*
- *red_wine_df = pd.read_csv('winequality-red.csv', delimiter=';')*
- *red_wine_df.info()*
- *red_wine_df.mean()*
- *red_wine_df['alcohol'].mean()*
- *red_wine_df.median()*
- *red_wine_df['alcohol'].median()*
- *red_wine_df.mode()*
- *red_wine_df['alcohol'].mode()*

Medidas estatísticas básicas

- red_wine_df['quality'].mode()
- sns.countplot(red_wine_df, x="quality")

Variância

- Mede como os dados estão espalhados.
- Baixos valores indicam que as amostras estão mais próximas umas das outras e próximas da média.
- Variância $= \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$

- *red_wine_df.var()*
- *red_wine_df['alcohol'].var()*

Desvio Padrão

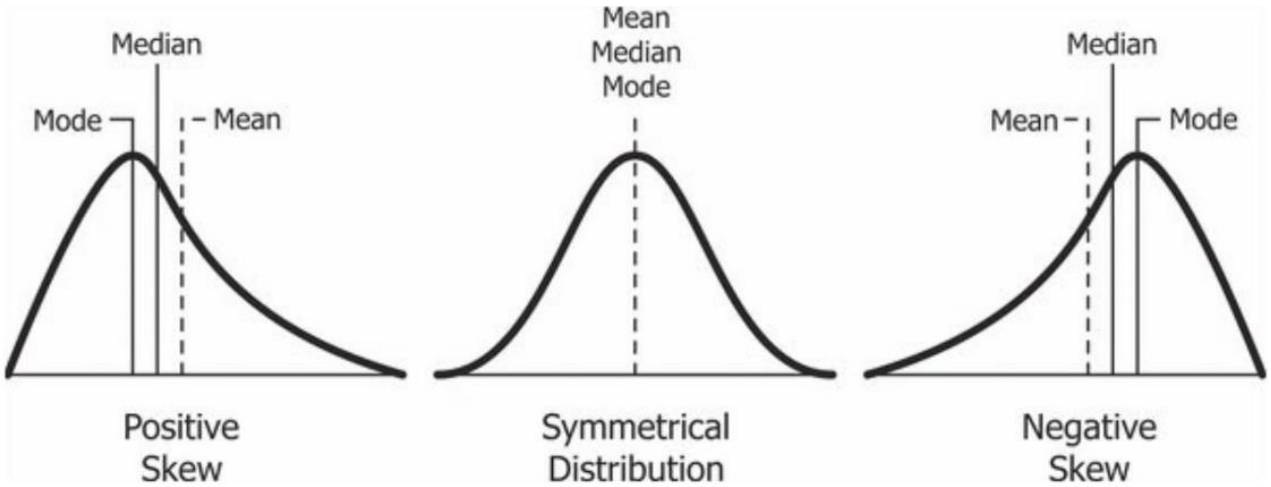
- É a medida da variação em relação a média.
- É similar a variância, porém é expressa na mesma unidade de medida.
- Desvio padrão $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$

- *red_wine_df.std()*
- *red_wine_df['alcohol'].std()*

Skewness e Kurtosis

- O que é Skewness?
  - Skewness é uma medida de assimetria de uma distribuição.
  - Uma medida do quanto difere da distribuição normal.
  - A distribuição normal tem um valor skewness de 0.

Skewness e Kurtosis

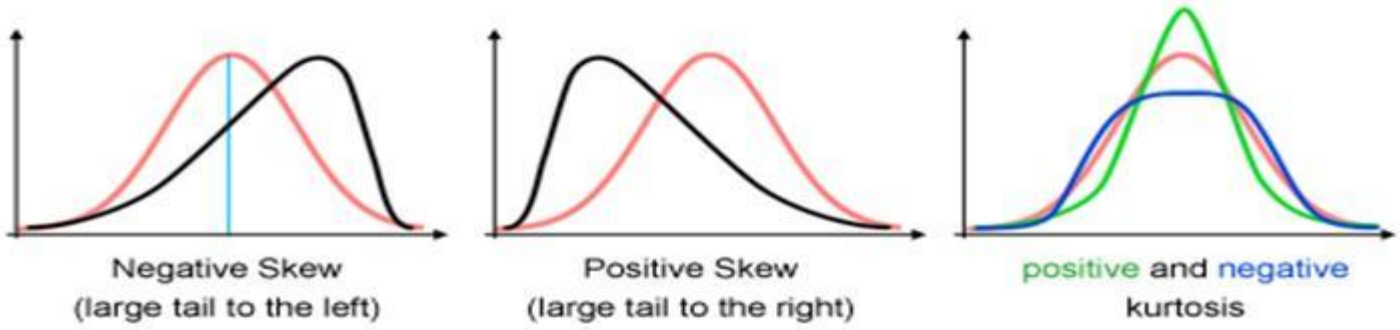


https://devopedia.org/exploratory-data-analysis

Skewness e Kurtosis

- O que é Kurtosis?
  - É a medida do grau de achatamento de uma distribuição.
  - Uma medida do quanto difere da distribuição normal.
  - A distribuição normal tem um valor kurtosis de 0.
  - Indica como os valores são agrupados em torno da tendência central

Skewness e Kurtosis



https://devopedia.org/exploratory-data-analysis

Skewness e Kurtosis

- *red_wine_df.skew()*
- *red_wine_df['alcohol'].skew()*

- *red_wine_df.kurt()*
- *red_wine_df['alcohol'].kurt()*

- *print("Kurtosis = {}|nSkewness = {}".format(red_wine_df['alcohol'].kurt(),red_wine_df['alcohol'].skew()))*

Percentis

- Percentis denotam o percentual de valores nos dados (ordenados) que estão abaixo de um determinado valor.
- Fórmula:
  - Percentil de x = (Número de valores menores do que x/total de observações)*100
  - Posição do valor no percentil = (Percentil/100)*(total de observações +1)
  - A parte fracionária indica que o valor não está nos dados, mas entre dois valores.

Percentis

- *red_wine_df.quantile(q=0.5)*
- *red_wine_df['alcohol'].quantile(q=0.5)*

Quartis

- Quartis dividem as amostras de dados em 4 partes.
- Primeiro quartil – Q1 = 25th percentil
- Segundo quartil – Q2 = 50th percentil
- Terceiro quartil – Q3 = 75th percentil

Quartis

- *red_wine_df.quantile(q=0.25)*
- *red_wine_df.quantile(q=0.5)*
- *red_wine_df.quantile(q=0.75)*

- *red_wine_df['alcohol'].quantile(q=0.25)*
- *red_wine_df['alcohol'].quantile(q=0.5)*
- *red_wine_df['alcohol'].quantile(q=0.75)*

Intervalo interquartil

- O intervalo interquartil é outra medida para a variabilidade dentro do conjunto de dados.
- Intervalor interquartil (IIQ) = Q3 – Q1
- Aproximadamente 50% dos dados devem estar nesse intervalo.
- Não são afetados por outliers.

Estatística Descritiva

- Pode ser utilizada a função describe() para calcular:
  - Quantidade
  - Média
  - Desvio Padrão
  - Mínimo
  - 25%
  - 50%
  - 75%
  - Max
- *red_wine_df.describe()*

Gráficos de Contagem

- *sns.countplot(red_wine_df, x='quality')*
- *plt.xlabel('Qualidade')*
- *plt.ylabel('Quantidade')*
- *plt.show()*

Gráfico de Pizza

- *red_wine_df['quality'].unique(),red_wine_df.groupby('quality').size()*
- *np.sort(red_wine_df['quality'].unique())*
- *contagem_amostras_por_qualidade = pd.DataFrame({'Qualidade': np.sort(red_wine_df['quality'].unique()), 'Quantidade': red_wine_df.groupby('quality').size()})*
- *contagem_amostras_por_qualidade['Quantidade']*

Gráfico de Pizza

- *colors = sns.color_palette('pastel')[0:6]*
- *plt.pie(contagem_amostras_por_qualidade['Quantidade'], labels = contagem_amostras_por_qualidade['Qualidade'],colors=colors, startangle=90, pctdistance = 1.05, autopct='%1.1f%%', radius=3, labeldistance = None)*

- *plt.show()*

Gráfico de Pizza

- *colors = sns.color_palette('pastel')[0:5]*
- *plt.pie(contagem_amostras_por_qualidade['Quantidade'], labels = contagem_amostras_por_qualidade['Qualidade'], colors=colors, startangle=90, shadow=True,explode=(0.1, 0.1, 0.1, 0.1, 0.1, 0.1), radius=1, labeldistance=1.1)*
- *plt.title('Quantidade de amostras por classificação de vinho')*
- *plt.show()*

Histograma de Frequência

- *import pandas as pd*
- *import seaborn as sns*
- *import matplotlib.pyplot as plt*
- *sns.histplot(red_wine_df['alcohol'], kde=False, bins = 10)*
- *plt.xlabel('Intervalo')*
- *plt.ylabel('Frequência')*
- *plt.show()*

Histograma de Frequência Relativa

- *sns.histplot(red_wine_df['alcohol'], kde=False, bins=30, stat='density')*
- *plt.xlabel('Intervalo')*
- *plt.ylabel('Frequencia Relativa')*
- *plt.show()*

Gráfico de Densidade

- *sns.histplot(red_wine_df['alcohol'], kde=True, bins=15, stat ='percent', edgecolor = 'r', linewidth=2)*
- *plt.xticks(range(7, 16, 1))*
- *plt.ylabel('Densidade')*
- *plt.show()*

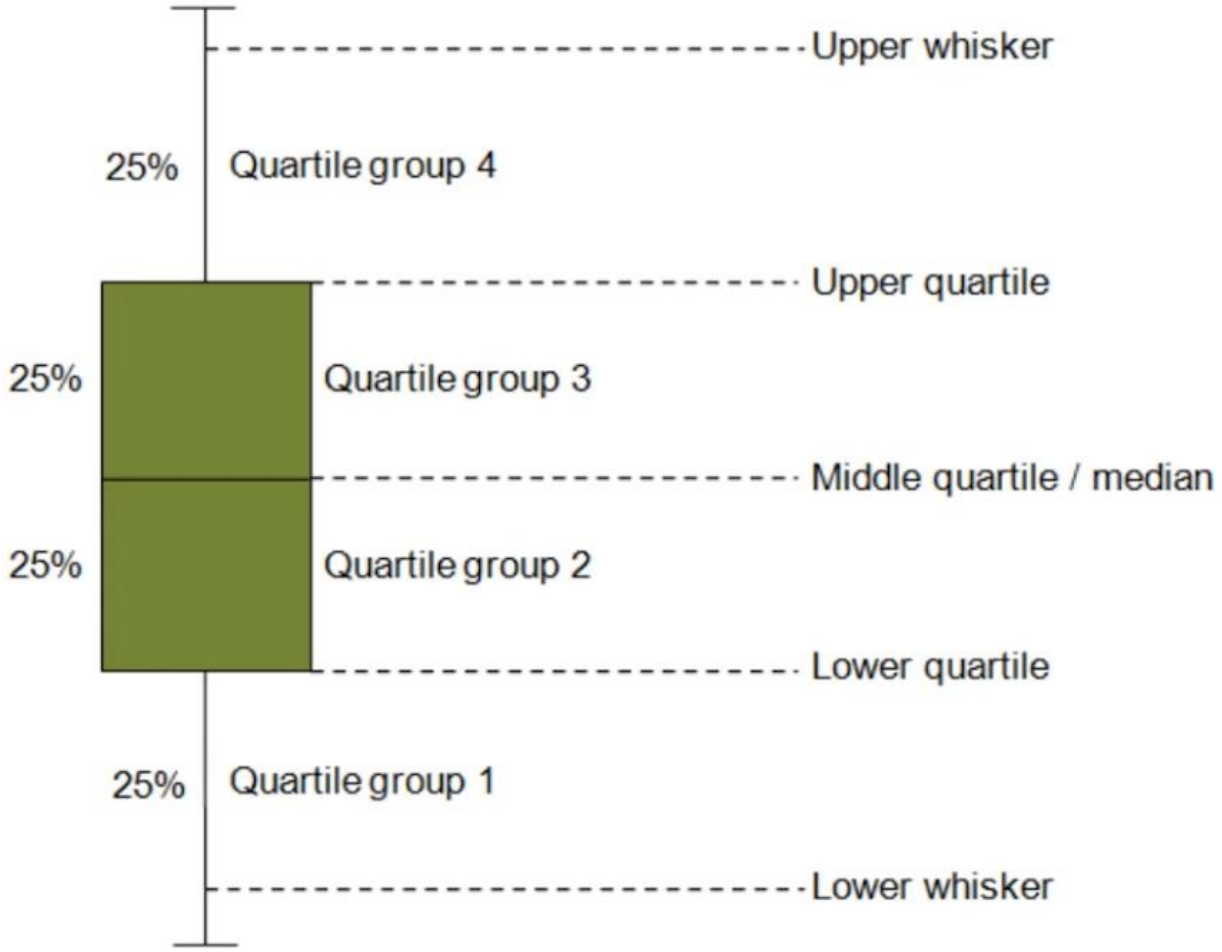
Gráfico de Densidade

- *plt.axvline(red_wine_df['alcohol'].mean(), color='b', linestyle='dashed', linewidth=1)*
- *plt.axvline(red_wine_df['alcohol'].median(), color='r', linestyle='dashed', linewidth=4)*
- *sns.kdeplot(red_wine_df['alcohol'])*
- *plt.ylabel('Densidade')*
- *plt.show()*

Box Plots

- Box plots permitem visualizar as seguintes informações no gráfico:
  - Valor mínimo.
  - Primeiro Quartil.
  - Mediana.
  - Terceiro Quartil.
  - Valor Máximo.
- Adicionalmente pode visualizar o IIQ/IQR.

Box Plots


Upper whisker
25%
Quartile group 4
Upper quartile
25%
Quartile group 3
Middle quartile / median
25%
Quartile group 2
Lower quartile
25%
Quartile group 1
Lower whisker

Box Plots

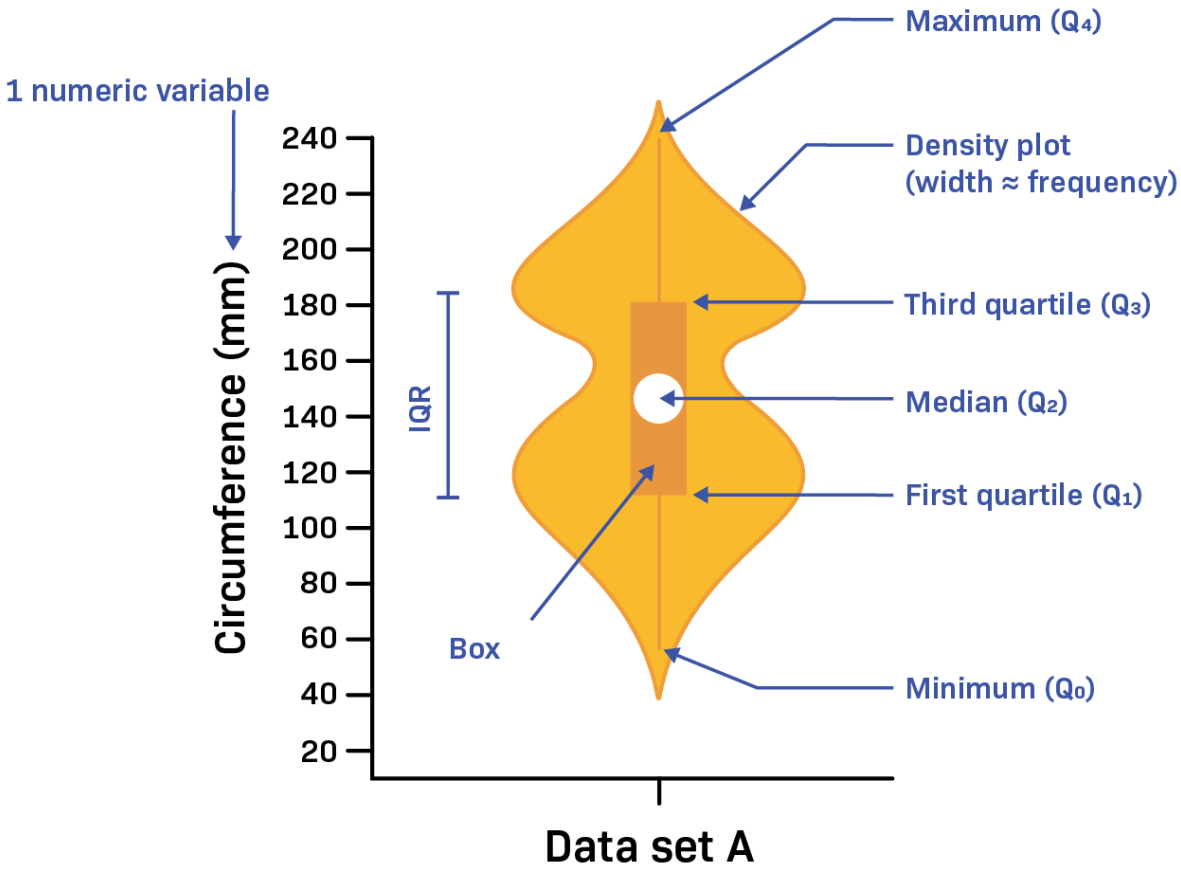
- *sns.boxplot(data=red_wine_df['alcohol'])*
- *plt.xlabel('Álcool')*
- *plt.show()*

- *sns.boxplot(data=red_wine_df['alcohol'], orient='h', showfliers=False)*
- *plt.xlabel('Álcool')*
- *plt.show()*

Gráfico de Violino

- Um gráfico de violino é um híbrido de um Box plot com um gráfico de densidade que mostra picos nos dados.
- É usado para visualizar a distribuição de dados numéricos.
- Além das informções dos blox plots, os gráficos de violino apresentam a distribuição de cada variável.

Gráfico de Violino



https://www.labxchange.org/library/items/lb:LabXchange:46f64d7a:html:1

Gráfico de Violino

- *sns.violinplot(data=red_wine_df['alcohol'], orient='v')*
- *plt.xlabel('Álcool')*
- *plt.show()*

- *sns.violinplot(data=red_wine_df['alcohol'], orient='h')*
- *plt.xlabel('Álcool')*
- *plt.show()*

Outliers

- O que são Outliers?
  - Um outlier é um valor que se encontra distante dos outros valores do conjunto de dados.
  - Pode ser muito menor ou muito maior do que outros valores no conjunto de dados.
  - Está fora do padrão dos dados.

Outliers

- Razões dos Outliers
  - Erros no registro dos dados.
  - Dados corrompidos durante o processo de transferência.
  - Está fora do padrão dos dados.
  - Os dados podem ter realmente valores altos ou baixos.

Outliers

- Exemplos de outliers

Gênero	Idade	Altura (cm)
M	20	150
F	21	151
M	24	155
F	24	153
M	28	280
F	26	160
M	19	158
F	22	157
M	26	158

Outliers

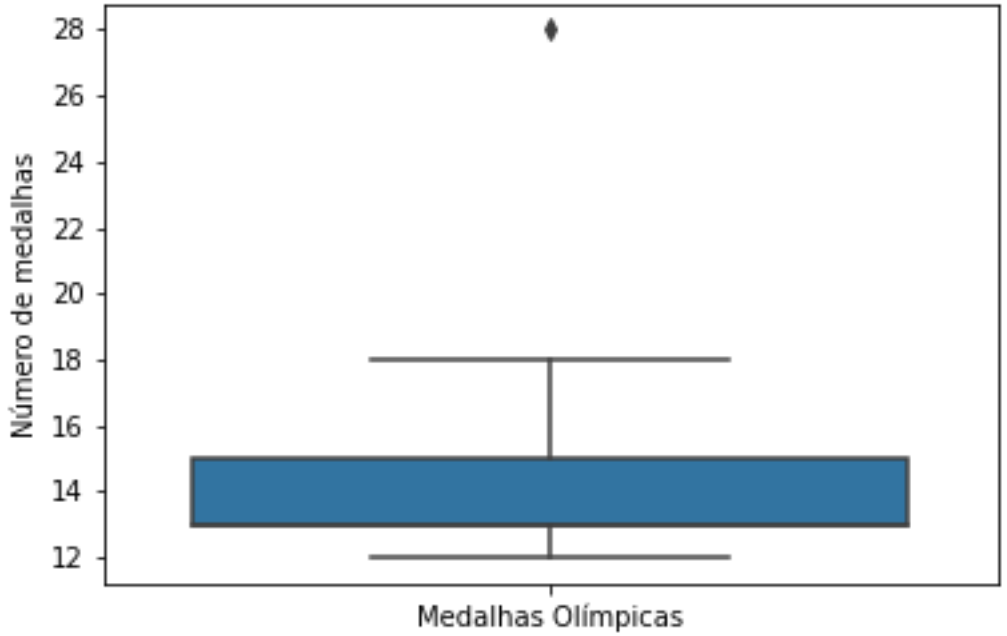
- Exemplos de outliers

Jogador	Medalhas Olímpicas
Michael Phelps	28
Larisa Latynina	18
Marit Bjørgen	15
Nikolai Andrianov	15
Ole Einar Bjørndalen	13
Boris Shakhlin	13
Edoardo Mangiarotti	13
Ireen Wüst	13
Takashi Ono	13
Paavo Nurmi	12

https://en.wikipedia.org/wiki/List_of_multiple_Olympic_medalists

Outliers

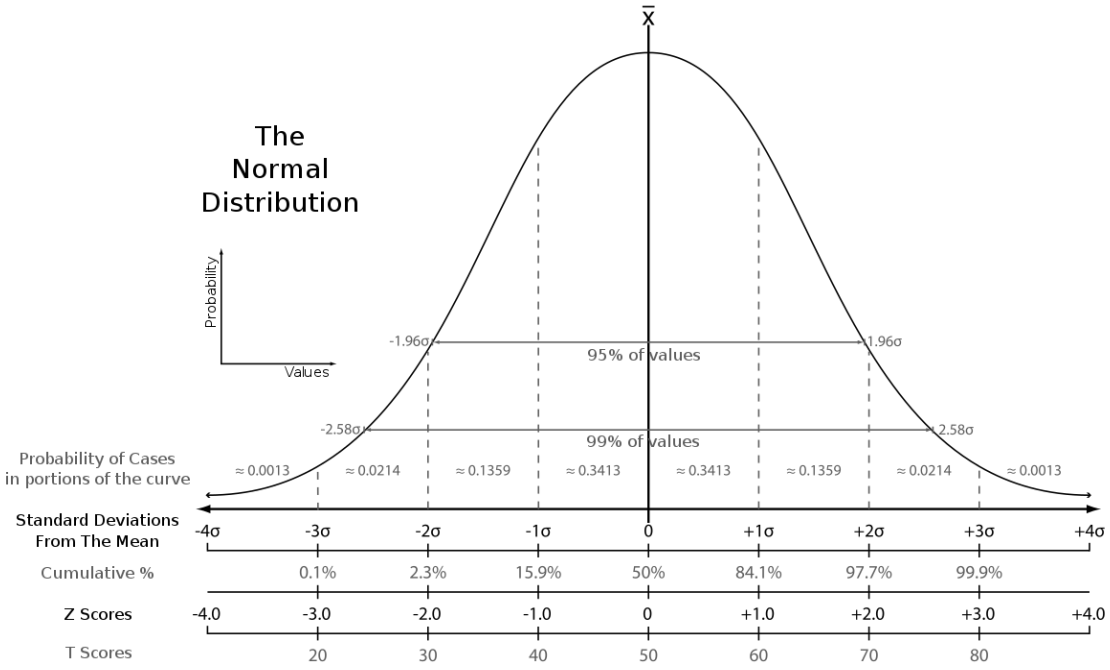
- Exemplos de outliers



Usando Z-scores para identificar outliers

- O que são Z-scores?
  - É uma medida de distância entre um valor e a média de um grupo de valores. Ele é medido em termos de desvios padrão da média.
  - O Z-score de um ponto é definido da seguinte forma:
    - Z-score de X_i = (X_i – média)/(Desvio padrão)
  - Geralmente, pontos com Z-scores >=3 são considerados outliers.

Usando Z-scores para identificar outliers



https://en.wikipedia.org/wiki/Standard_score

Usando Z-scores para identificar outliers

- *import scipy.stats*
- *import pandas as pd*
- *scipy.stats.zscore(red_wine_df['alcohol']) # Calculando Z-score*
- *z_score_alcohol = np.abs(scipy.stats.zscore(red_wine_df['alcohol'])) # Lista valores com z-score elevado, considerando o módulo.*
- *red_wine_df.iloc[np.where(z_score_alcohol>3)]*
- *red_wine_df_alcohol_no_outliers = red_wine_df.iloc[np.where(z_score_alcohol<=3)]*

Z-score modificado

- Limitações do Z-Score
  - Ele assume que os dados apresentam uma distribuição normal.
  - Utiliza média e desvio padrão que podem ser facilmente distorcidos por outliers.

Z-score modificado

- Utiliza a mediana.
- Utiliza Median Absolute Deviation (MAD) – Desvio Absoluto Mediano
- MAD = Mediana de ($|X_i$ – Mediana|) para todo X_i
- MAD é menos afetado por outliers porque utiliza a mediana.
- Z-score modificado para todo X_i = 0.6745 * (X_i – Mediana)/MAD

Z-score modificado

- *def z_score_modificado(dados):*
  - *mediana_dados = np.median(dados)*
  - *# Median Absolute Deviation*
  - *mad = np.median(dados.map(lambda x: np.abs(x - mediana_dados)))*
  - *# Z-score modificado*
  - *z_score_modificado = list(dados.map(lambda x: 0.6745* (x - mediana_dados)/mad))*
  - *return z_score_modificado*

Z-score modificado

- *z_score_modificado(red_wine_df['alcohol'])*
- *z_score_modificado_alcohol = z_score_modificado(red_wine_df['alcohol'])*
- *red_wine_df.iloc[np.where(np.abs(z_score_modificado_alcohol)>=3)]*
- *red_wine_df_alcohol_no_outliers = red_wine_df.iloc[np.where(np.abs(z_score_modificado_alcohol)<3)]*
- *red_wine_df_alcohol_no_outliers*

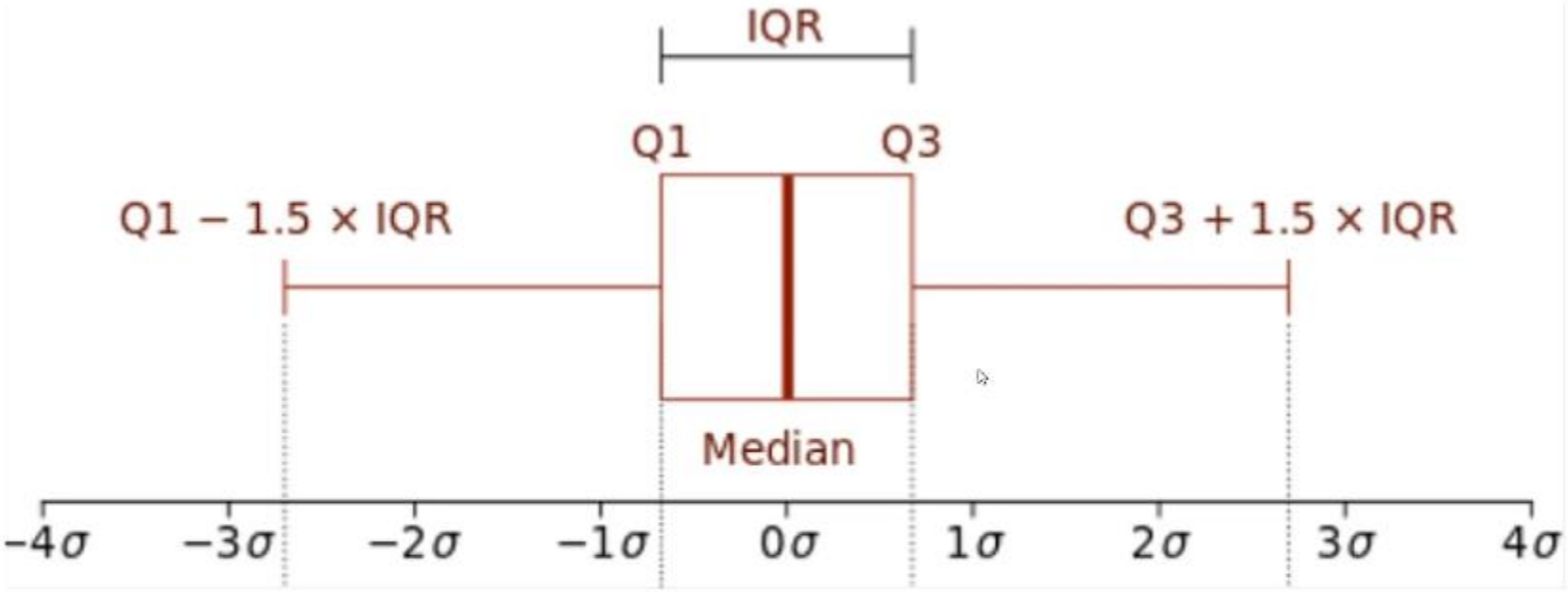
Z-score modificado

- *sns.boxplot(data=red_wine_df_alcohol_no_outliers['alcohol'], orient='h')*
- *plt.xlabel('Álcool')*
- *plt.show()*

Usando IIQ para Detectar Outliers

- Como usar IIQ/IQR para detectar outliers?
  - Qualquer valor que seja maior do que 1,5*IIQ+Q3 ou menor do Q1-1,5*IIQ pode ser considerado outlier.

Usando IIQ para Detectar Outliers



Source: https://en.wikipedia.org/wiki/Interquartile_range

Usando IIQ para Detectar Outliers

- *def limite_inferior_superior(dados):*
  - *#Calcular primeiro e tercerio quartis*
  - *q1 = np.percentile(dados, 25)*
  - *q3 = np.percentile(dados, 75)*

  - *# Calcular intervalor interquartil*
  - *iqr = q3 - q1*

  - *# Calcula o limite inferior e superior*
  - *limite_inferior = q1 - (iqr * 1.5)*
  - *limite_superior = q3 + (iqr * 1.5)*

  - *return limite_inferior, limite_superior*

Usando IIQ para Detectar Outliers

- *def encontrar_outliers_iqr(dados):*
  - *limite_inferior, limite_superior = limite_inferior_superior(dados)*
  - *# Retorna os dados fora dos limites*
  - *return dados[np.where((dados > limite_superior) | (dados < limite_inferior))]*

- *encontrar_outliers_iqr(red_wine_df['alcohol'].values)*

Usando IIQ para Detectar Outliers

- *red_wine_df_no_outliers_alcool = red_wine_df[~red_wine_df['alcohol'].isin(encontrar_outliers_iqr(red_wine_df['alcohol'].values))]*
- *red_wine_df_no_outliers_alcool*

- *sns.boxplot(data=red_wine_df_no_outliers_alcool['alcohol'], orient='h')*
- *plt.xlabel('Álcool')*
- *plt.show()*

Teste de Hipótese

- Utilizado para testar hipótese sobre uma população.
- Teste de significância.

Teste de Hipótese

- *from scipy.stats import normaltest*
- *sns.histplot(red_wine_df['alcohol'], kde=True, bins=50, stat ='percent')plt.ylabel('Densidade')plt.show()*
- *normaltest(red_wine_df['alcohol'])*

Teste de Hipótese

- *media = 0desvio_padrao = 0.1s = np.random.normal(media, desvio_padrao, 1000)*
- *normaltest(s)*
- *sns.histplot(s, kde=True, bins=50, stat ='percent')plt.ylabel('Densidade')plt.show()*