

Estatística no Python, SIAP 2020 (IC), ECEC- PUC Goiás.
Responsáveis: Maria José Pereira Dantas (ECEC),
Danilo Milhomem (MEPROS),
Matheus Sanclé (IC Eng Comp, ECEC) e
Joás Rodrigues (IC Matemática, ECEC)

Notas de apoio

Estatística - Análise de dados e visualização dos dados

Mercado Financeiro (aplicação)

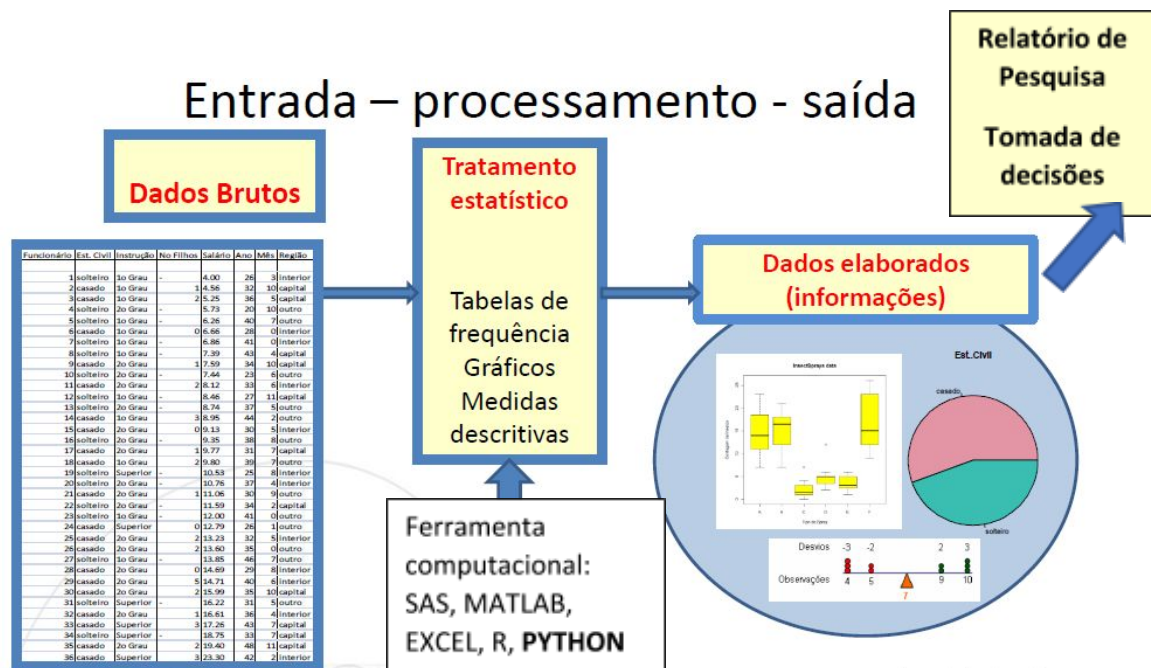
Python

Estatística - Análise de dados e visualização de dados

Maria José Pereira Dantas.

Para que uma análise estatística seja útil no processo de tomada de decisão, os dados de entrada devem ser apropriados. Se os dados forem distorcidos por tendências, ambiguidades ou outros tipos de erros, mesmo as metodologias estatísticas mais sofisticadas provavelmente não serão suficientes para compensar tais deficiências.

Atenção: “Garbage in, garbage out”



Fontes de dados:

Dados secundários:

dados já publicados por fontes governamentais, industriais e particulares;

• **Dados primários:** planejamentos experimentais (controle sobre as variáveis) ou pesquisa de levantamento (nenhum controle sobre as variáveis)

Era da tecnologia da informação (nunca antes estiveram disponíveis tantos dados – e de tantas fontes)

IMPORTANTE:

Levini, D.M, Berenson, M. L., Stephan, D. Estatística: Teoria e aplicações. LTC. 2000. Pag. 157.

“As **questões éticas** têm importância vital para todas as análises de dados. Como consumidores diários de informações temos a obrigação e o dever de questionar o que lemos em jornais e revistas e o que ouvimos no rádio e na televisão. **Ao longo do tempo, tem havido muito ceticismo em relação aos propósitos, ao foco e à objetividade dos estudos publicados.**”

“... **Considerações éticas** surgem quando estamos decidindo sobre quais resultados apresentar em um relatório e quais não apresentar.

É de importância fundamental documentar tanto os bons resultados quanto os ruins.

Além disso, ao se fazerem apresentações orais e apresentarem relatórios por escrito, é essencial que os resultados sejam dados de modo correto, objetivo e neutro.”

A questão chave é a **intenção**

“... Um comportamento antiético ocorre quando alguém intencionalmente escolhe uma medida resumida inapropriada (por exemplo, a média aritmética ou a média do intervalo para um conjunto de dados muito assimétrico) pode distorcer os fatos no sentido de apoiar determinada posição. Além disso, um comportamento antiético ocorre quando alguém seletivamente deixa de relatar resultados importantes quando estes seriam prejudiciais ao suporte de determinada posição.”

- **Problema de pesquisa (questões a serem respondidas e hipóteses), com base em uma população ou amostra. Nas Ciências e Engenharias é comum o uso de amostras.**
- **Coleta dos dados: dados primários (pesquisa de levantamento ou planejamento experimental)**
- **Variáveis de interesse são características (atributos). Podem ser qualitativas ou quantitativas discretas ou contínuas.**
- **Base de dados: o conjunto de dados coletados, pode ser representado por meio de uma matriz ou tabela, denotada por $X^{n \times m}$, em que n indica o número de objetos observados e m indica o número de características(ou atributos) que foram coletadas para cada objeto.**
- **Análise dos dados (Estatística descritiva, testes de hipóteses, inferência)**
- **Análise UNIVARIADA, $m=1$ (tabelas de frequências, medidas e gráficos,**
 - **BIVARIADA, $m=2$ (Diagrama de dispersão, correlação, covariância, ajuste de reta)**

- o MULTIVARIADA, $m \geq 2$ (análise de componentes principais, clusterização, ...)
- Apresentação
- Conclusões

A Estatística lida com incerteza e variabilidade. A incerteza é avaliada pela teoria das probabilidades. O estudo de modelos probabilísticos para descrever as populações de interesse é essencial.

Exemplos de alguns modelos probabilísticos:

Modelo	Parâmetros	Valores possíveis dos parâmetros	Função de Distribuição Acumulada	Média	Variância
Uniforme	a e b	$a < b$	$f(x) = \begin{cases} \frac{1}{b-a}, & \text{se } a \leq x \leq b \\ 0, & \text{caso contrário} \end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Exponencial	λ	$\lambda > 0$	$f(x) = \begin{cases} 0, & \text{se } x < 0 \\ \lambda \exp(-\lambda x), & \text{se } x \geq 0 \end{cases}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Normal	μ e σ^2	μ qualquer $\sigma > 0$	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right\},$ para todo x real	μ	σ^2

A variabilidade está presente nos dados coletados que são usados para responder questões de pesquisa e testar hipóteses no mundo real. O objetivo da Estatística Descritiva é tratar a variabilidade amostral e obter informações.

<http://www.portaaction.com.br/estatistica-basica>

<http://www.portaaction.com.br/estatistica-basica/12-coleta-de-dados>

ANÁLISE DESCRITIVA DE DADOS

Após a coleta e a digitação de dados das variáveis em um **banco de dados** apropriado, o próximo passo é a **análise descritiva**. Esta etapa é fundamental, pois uma análise descritiva detalhada permite ao pesquisador familiarizar-se com os dados, organizá-los e sintetizá-los de forma a obter as informações necessárias do conjunto de dados para responder as questões que estão sendo investigadas.

Tabelas, gráficos e medidas:

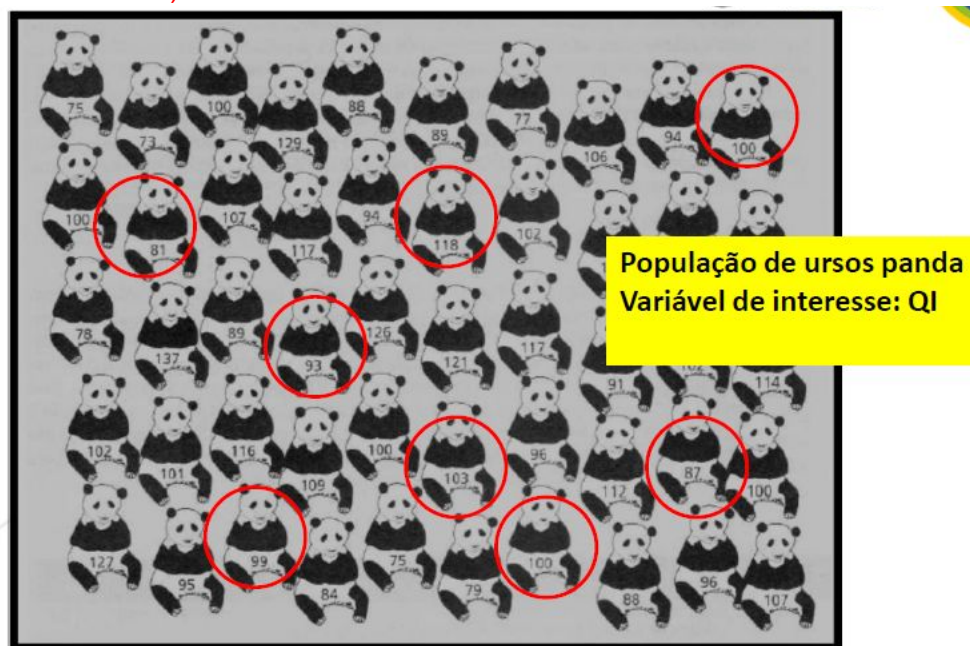
descrevem a tendência dos dados, quantificam a variabilidade, permitem a detecção de estruturas interessantes e valores atípicos no banco de dados. **Uma**

boa forma de iniciar uma análise descritiva adequada é verificar os tipos de variáveis disponíveis...

Variáveis podem ser classificadas da seguinte forma: **qualitativas**: nominais ou ordinais e **quantitativas** discretas ou Contínuas

Dados brutos obtidos em pesquisa de levantamento ou planejamento experimental -> **tratamento estatístico** -> **informações** para a tomada de decisões em presença de incerteza (geralmente com base em amostras) e variabilidade (nos dados da amostra existe variabilidade)

As Medidas de interesse podem ser obtidas para **todos os elementos** de uma população (**CENSO**) ou para **uma parte** denominada amostra (estudo por **AMOSTRAGEM**).



População $X = \{x_1, x_2, \dots, x_N\}$, dados de uma população de tamanho N .

Amostra, $X = \{x_1, x_2, \dots, x_n\}$, dados de uma amostra de tamanho n .

Estatística descritiva (organizar, resumir e descrever o comportamento dos dados).

Ferramentas descritivas da análise univariada:

tabelas de frequência (para categorias, valores individuais ou agrupamentos).
Consulte o material a seguir.

<http://www.portaaction.com.br/estatistica-basica/13-exposicao-dos-dados>

<http://www.portaaction.com.br/estatistica-basica/16-distribuicao-de-frequencias>

medidas:

síntese (média, moda, mediana),

População $X=\{x_1, x_2, \dots, x_N\}$, dados de uma população de tamanho N .

$\mu = \frac{x_1 + \dots + x_N}{N}$	Média populacional
$\sigma^2 = \sum_{i=1}^N \frac{(x_i - \mu)^2}{N}$	Variância populacional
$\sigma = \sqrt{\sigma^2} = \sqrt{\sum_{i=1}^N \frac{(x_i - \mu)^2}{N}}$	Desvio padrão populacional

Amostra, $X=\{x_1, x_2, \dots, x_n\}$, dados de uma amostra de tamanho n .

$\bar{x} = \frac{x_1 + \dots + x_n}{n}$	Média amostral
$s^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1}$	Variância amostral
$s = \sqrt{s^2} = \sqrt{\sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1}}$	Desvio padrão amostral

Mediana: Considerando a existência de n observações para uma variável qualquer x , a mediana pode ser definida como o valor central das observações ordenadas quando n for ímpar. Para os casos em que n for par, a mediana é definida como valor médio entre as duas observações centrais ordenadas.

Moda: É o valor que ocorre com maior frequência em um conjunto de valores. A moda é facilmente reconhecida, basta, de acordo com definição, procurar o valor que mais se repete.

- Observação: há conjuntos em que não existe valor modal, isto é, conjuntos nos quais nenhum valor apareça mais vezes que os outros. Nestes casos dizemos que a série é **amodal**.
- Porém, em outros casos, pode haver dois ou mais valores de concentração. Dizemos, então, que o conjunto tem dois valores (**bimodal**) ou mais (**multimodal**).

Variância: é a medida de dispersão mais empregada geralmente, pois leva em consideração a totalidade dos valores da variável em estudo. Baseia-se nos desvios em torno da média aritmética, sendo um indicador de variabilidade.

Desvio padrão: Seguindo a mesma linha de raciocínio usado para o cálculo da variância, precisamos, agora, aproximar a medida de dispersão da variável original. Para isso, calculamos o desvio padrão, que é a raiz quadrada da variância:

Medida de posição (quantil - percentil),

Um **percentil ou quantil**, denotado por $p_y\%$, representa, dentre m observações, o valor tal que $y\%$ das observações são menores do que ele. Esta medida trata-se de uma generalização da mediana (que corresponde a $p_{50\%}$) e do primeiro e terceiro quartis (que correspondem a $p_{25\%}$ e $p_{75\%}$, respectivamente).

<http://www.portalection.com.br/estatistica-basica/21-medidas-de-posicao>

<http://www.portalection.com.br/estatistica-basica/31-boxplot>

medida dispersão absoluta (amplitude, desvio médio, variância, desvio padrão, desvio interquartil),

Amplitude: consiste na diferença entre o valor máximo e mínimo de um conjunto de observações.

<http://www.portalection.com.br/estatistica-basica/22-medidas-de-dispersao>

medida dispersão relativa (coeficiente de variação), assimetria, curtose.

<http://www.portalection.com.br/estatistica-basica/25-coeficiente-de-assimetria>

<http://www.portalection.com.br/estatistica-basica/26-curtose>

Coefficiente de Variação CV:

- Trata-se de uma medida relativa de dispersão, útil para a comparação em termos relativos do grau de concentração em torno da média de séries

distintas. É dado por: $CV = \frac{S}{\bar{X}} 100\%$

- A importância de se estudar o coeficiente de variação se dá, pois o desvio-padrão é relativo à média. E como duas distribuições podem ter médias diferentes, o desvio destas distribuições não é comparável. Logo, o coeficiente de variação é muito utilizado para comparação entre amostras.

Assimetria: a assimetria (ou *skewness*, em inglês) trata-se de uma medida que avalia uma distribuição dos dados em torno de sua média. Ela pode assumir valores negativos, positivos ou próximos de 0. No primeiro caso, a cauda da distribuição é mais alongada à esquerda e, por consequência, a distribuição dos dados concentra-se mais à direita no seu respectivo gráfico. No segundo caso, a cauda da distribuição é mais alongada para a esquerda, o que aponta uma maior concentração dos dados à direita do seu respectivo gráfico. Por fim, no

terceiro caso, a distribuição possui caudas aproximadamente balanceadas e, como resultado, ela terá uma maior simetria.

Curtose: a curtose é uma medida que caracteriza o achatamento da distribuição dos dados. Assim como a assimetria, os seus valores podem ser negativos, positivos ou próximos de 0. No primeiro caso, a distribuição é mais achatada e apresenta picos mais baixos e caudas

mais leves quando comparada à distribuição normal. No segundo caso, a distribuição dos dados apresenta picos mais elevados e caudas mais pesadas ao se comparar à distribuição normal. Por fim, no último caso, a distribuição dos dados apresenta achatamento e caudas próximas ao que ocorre com a distribuição normal.

Gráficos (barras, setorial, histograma, boxplot, quantil-quantil)

Os gráficos constituem uma ferramenta visual importante, pois facilitam a análise e a interpretação de um conjunto de dados. Tendo em vista que através deles podemos interpretar as informações e apresentá-las de uma forma mais clara e objetiva.

- Para variáveis qualitativas (que são aquelas em que o resultado da variável não é um número), tais como: grau de escolaridade, raça, sexo, etc, usa-se os gráficos de **colunas, barras e pizza**.

- Para variáveis quantitativas (que são aquelas em que o resultado da variável é um número), tais como: altura, peso, idade, usa-se os gráficos de: **diagrama de pontos, diagrama de barras, ramo-e-folhas, histograma, box plot**.

Box plot (Um box plot é um gráfico apresentado em formato de caixa).

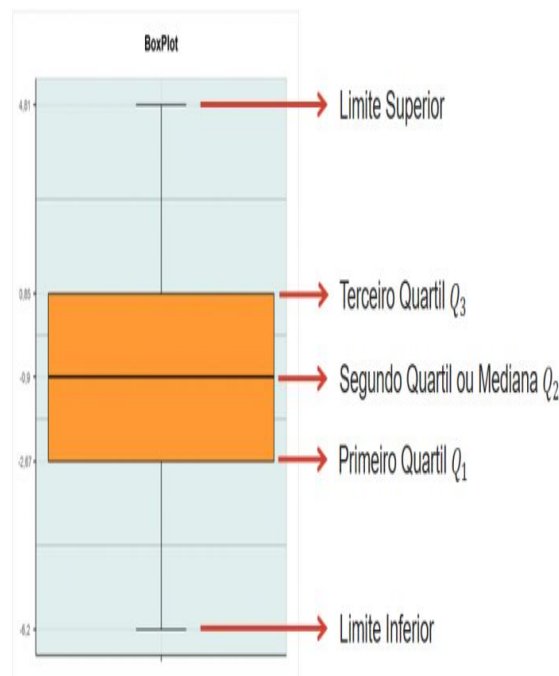


Figura extraída de <http://www.portalaction.com.br/estatistica-basica/31-boxplot>

Em que:

Limite inferior: $\max\{\min(\text{dados}); Q_1 - 1,5(Q_3 - Q_1)\}$.

Limite superior: $\min\{\max(\text{dados}); Q_3 + 1,5(Q_3 - Q_1)\}$.

linha central marca a mediana do conjunto de dados;

A parte inferior da caixa é delimitada pelo primeiro quartil (Q_1) e a *parte superior* pelo terceiro quartil (Q_3);

Podemos, com isso, verificar também o intervalo interquartil dado pela diferença

entre o primeiro e o terceiro quartil ($IQR = Q_3 - Q_1$);

As hastes inferiores e superiores se estendem, respectivamente, do quartil inferior até o menor valor não inferior a $Q_1 - 1,5 IQR$ e do quartil superior até o maior valor não superior a $Q_3 + 1,5 IQR$;

Os valores inferiores a $Q_1 - 1,5 IQR$ e superiores a $Q_3 + 1,5 IQR$ são *representados* individualmente no gráfico sendo estes valores caracterizados como outliers, ou seja, que estão fora do intervalo $Q_1 - 1,5 IQR < \text{valor} < Q_3 + 1,5 IQR$;

As quantidades $Q_1 - 1,5 IQR$ e $Q_3 + 1,5 IQR$ *delimitam as cercas inferior e superior*, respectivamente, e constituem limites para além dos quais, como visto, os dados passam a ser considerados *outliers*.

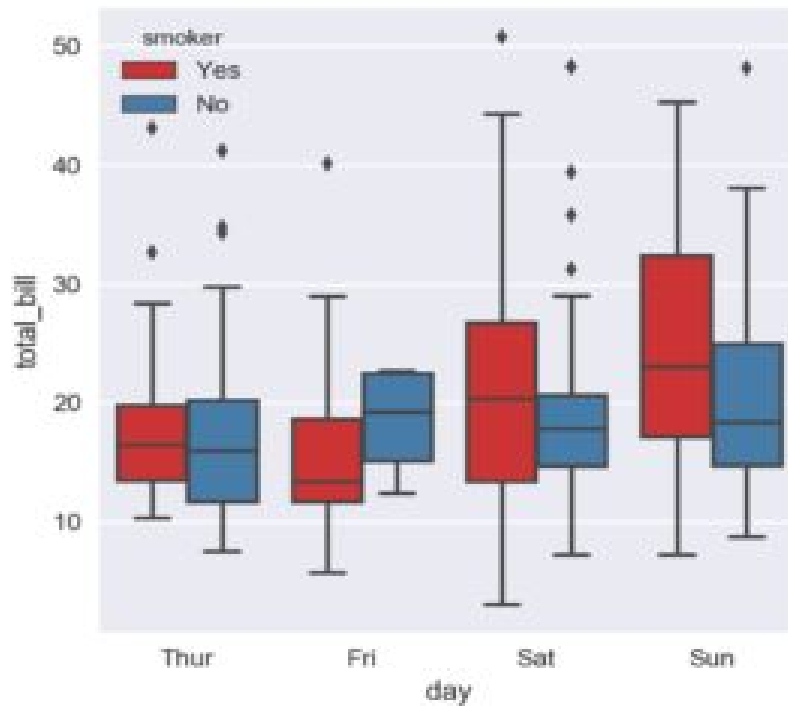


Figura extraída de <https://python-graph-gallery.com/seaborn/>

Histograma

Uma das formas mais simples de ilustrar a distribuição de um conjunto de valores de uma variável é o uso de histogramas. Neste tipo de gráfico tem-se, no eixo horizontal, o conjunto (ou intervalos) de valores observados, enquanto que no eixo vertical, apresenta-se a frequência de ocorrência de cada valor (ou valores dentro de um intervalo) presente na amostra analisada.

<http://www.portaaction.com.br/estatistica-basica/17-histograma>

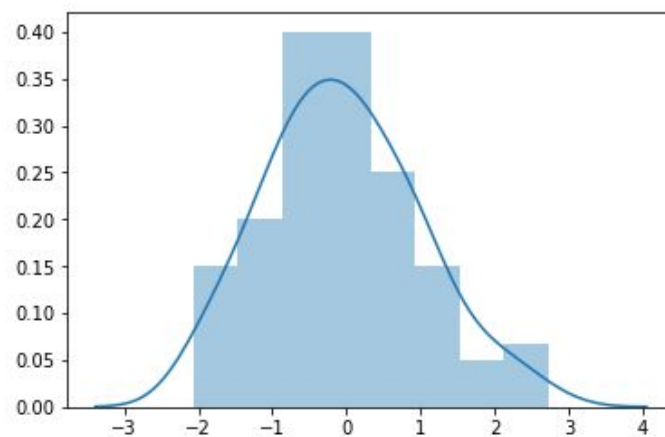
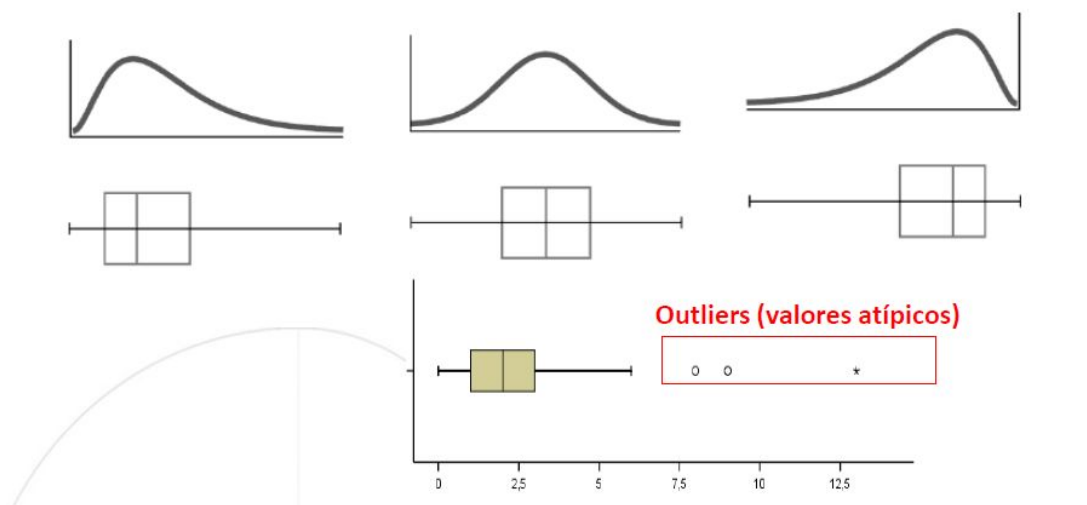


Figura extraída de <https://python-graph-gallery.com/seaborn/>

Exemplo de comparação entre um histograma que representa a distribuição normal e um boxplot para os mesmos dados.

Interpretação do box plot



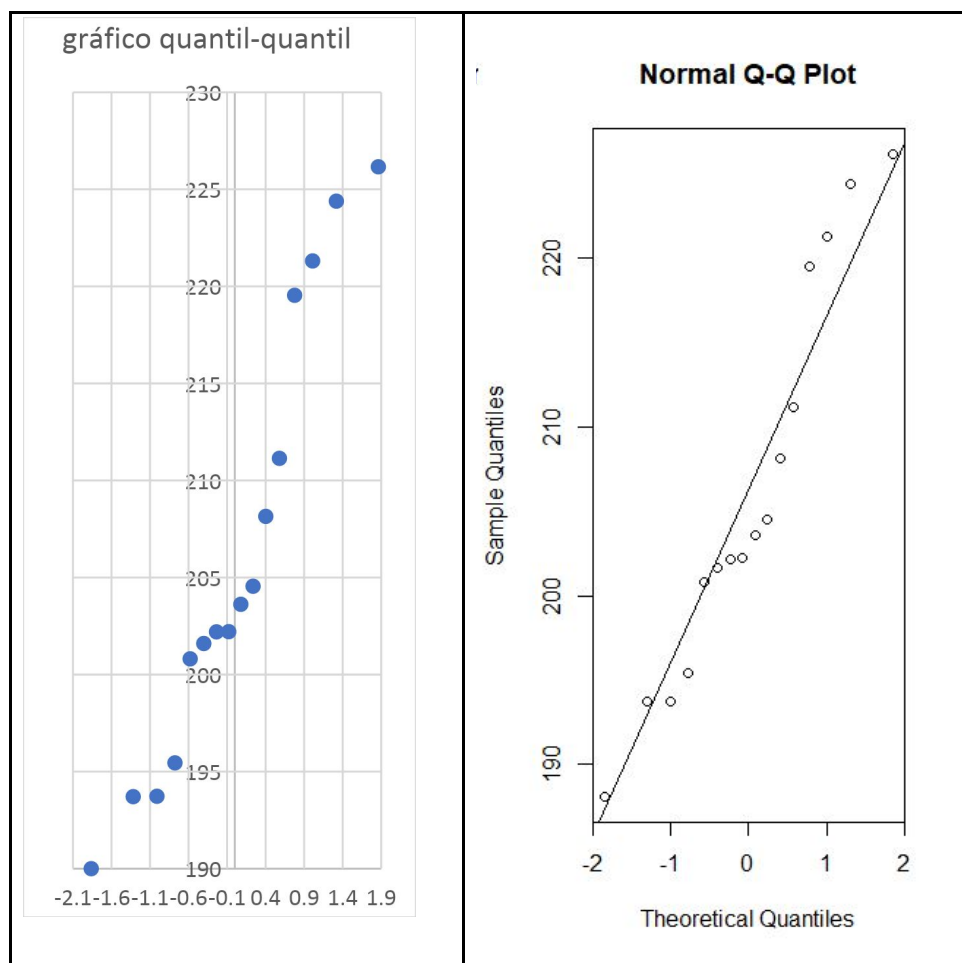
Avaliando a normalidade dos dados com o gráfico quantil-quantil, ou de probabilidade normal.

Gráfico quantil-quantil.

Para se construir um gráfico de probabilidade, quantil-quantil, as observações da amostra são ordenadas da menor para a maior, ou seja, a amostra x_1, x_2, \dots, x_n é ordenada como $x_{(1)}, x_{(2)}, \dots, x_{(n)}$, em que $x_{(1)}$ é a menor observação, $x_{(2)}$ é a segunda menor observação e assim por diante, com $x_{(n)}$ sendo a maior observação. Um gráfico de probabilidade normal pode ser construído plotando os escores normais padrões z_i versus $x_{(i)}$, em que os escores satisfazem $(i-0,5)/n = P(Z \leq z) = \Phi(z)$. Se a distribuição normal descrever adequadamente os dados, os pontos no gráfico se distribuirão, aproximadamente ao longo de uma reta; se os pontos desviarem significativamente de uma linha reta, então o modelo hipotético não será apropriado.

	Quantil amostra, eixo y		quantil teórico,eixo x
	valor ordenado, $x_{(i)}$	$pi=(i-0,5)/n$	$\Phi^{-1}(z_i)=z_i$, obtenha z_i na tabela da normal
1	188,12	0,03125	-1,862731867
2	193,71	0,09375	-1,318010897

3	193,73	0,15625	-1,009990169
4	195,45	0,21875	-0,776421761
5	200,81	0,28125	-0,579132162
6	201,6	0,34375	-0,402250065
7	202,2	0,40625	-0,237202109
8	202,21	0,46875	-0,078412413
9	203,62	0,53125	0,078412413
10	204,55	0,59375	0,237202109
11	208,15	0,65625	0,402250065
12	211,14	0,71875	0,579132162
13	219,54	0,78125	0,776421761
14	221,31	0,84375	1,009990169
15	224,39	0,90625	1,318010897
16	226,16	0,96875	1,862731867



Interpretação:

Se a distribuição representada for uma distribuição Normal os dados estarão alinhados sobre a reta;

Se a distribuição é simétrica e tem “caudas mais curtas” do que a normal, ou seja, a curva de densidade cresce mais rapidamente nas caudas do que a curva normal, no gráfico aparecerá um padrão em forma de S, em que os pontos na extrema direita tendem a estar acima da reta que está no centro do gráfico, enquanto que os pontos da extrema esquerda tendem a estar abaixo.

Se a distribuição é simétrica e tem “caudas longas”, em comparação com a normal. **Identificação no gráfico:** padrão em forma de S, em que os pontos na extrema esquerda se curvam para baixo.

Se a distribuição é simétrica e tem “caudas longas”, em comparação com a normal. **Identificação no gráfico:** padrão em forma de S, em que os pontos na extrema esquerda se curvam para baixo.

Se a distribuição tiver inclinação positiva (cauda curta a esquerda e longa à direita) os pontos estarão nas extremidades acima da reta, gerando um padrão curvo.

Ferramentas descritivas da análise bivariada

Dados: $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

A **correlação** assim como a **covariância**, são medidas estatísticas que indicam como duas variáveis se relacionam entre si

O coeficiente de correlação de Pearson, r , é uma medida do grau de relação linear entre duas variáveis quantitativas. Este coeficiente varia entre os valores -1 e 1. O valor 0 (zero) significa que não há relação linear, o valor 1 indica uma relação linear perfeita e o valor -1 também indica uma relação linear perfeita, mas inversa, ou seja, quando uma das variáveis aumenta a outra diminui. Quanto mais próximo estiver de 1 ou -1, mais forte é a associação linear entre as duas variáveis.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum (x_i - \bar{x})^2)(\sum (y_i - \bar{y})^2)}}$$

A covariância mede a relação linear entre duas variáveis. A covariância é semelhante à correlação entre duas variáveis, no entanto, elas diferem nas seguintes maneiras:

- os coeficientes de correlação são padronizados. Assim, um relacionamento linear perfeito resulta em um coeficiente de correlação 1. A correlação mede tanto a força como a direção da relação linear entre duas variáveis.
- os valores de covariância não são padronizados. Portanto, a covariância pode variar de menos infinito a mais infinito. Assim, o valor para uma relação linear ideal depende dos dados. Como os dados não são padronizados, é difícil determinar a força da relação entre as variáveis.

É possível utilizar a covariância para compreender a direção da relação entre as variáveis. Valores de covariância positivos indicam que valores acima da média de uma variável estão associados a valores médios acima da outra variável e abaixo dos valores médios são igualmente associados. Valores de covariância negativos indicam que valores acima da média de uma variável estão associados com valores médios abaixo da outra variável.

O coeficiente de correlação é uma função da covariância. O coeficiente de correlação é igual à covariância dividida pelo produto dos desvios padrão das variáveis. Portanto, uma covariância positiva sempre resulta em uma correlação positiva e uma covariância negativa sempre resulta em uma correlação negativa.

$$r = \frac{S_{xy}}{S_x S_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Onde

S_{xy} é a covariância;

S_x e S_y representam o desvio padrão, respectivamente, das variáveis x e y .

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

O cálculo da **Covariância** é o seguinte:

$$Cov(x, y) = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{n}$$

Esta é a fórmula tradicional da Covariância. Percebamos que o numerador da fórmula contempla justamente os desvios dos elementos X_i e Y_i , em relação às suas respectivas médias \bar{X} e \bar{Y} .

Então, o entendimento da Covariância é: diz-nos se os pares de informação (X_i, Y_i) recolhidos na pesquisa estão próximos ou afastados do par que representa a média das variáveis (\bar{X}, \bar{Y}) .

Ocorre que existe uma forma ainda mais fácil de calcularmos essa Covariância. Precisariamos apenas desenvolver algebricamente a fórmula tradicional que aprendemos acima, e chegaríamos ao seguinte formato:

$$Cov(x, y) = \overline{XY} - \bar{X}\bar{Y}.$$

$$\text{Onde: } \overline{XY} = \frac{\sum X_i Y_i}{n} ; \bar{X} = \frac{\sum X_i}{n} ; \bar{Y} = \frac{\sum Y_i}{n}$$

E n é o número de pares de informações.

Gráfico: diagrama de dispersão (scatter plot)

x	y
x_1	y_1
x_2	y_2
x_3	y_3
.	.
.	.
x_n	y_n

Padrão de associação entre as variáveis.

Interesse: investigar a associatividade linear.

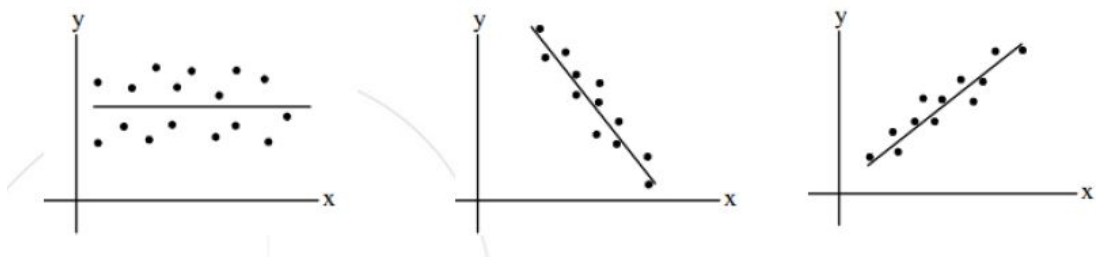
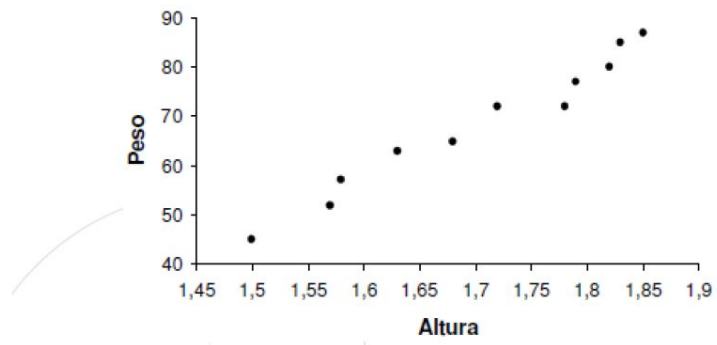


Diagrama de dispersão



Aplicação: análise de dados de Ativos financeiros.

Responsável: Danilo Milhomem

Base de dados do site Yahoo Finanças.

Ativos

INSERIR OS TICKERS DOS ATIVOS QUE SE DESEJAS ANALISAR

ativos = ['CSAN3.SA', 'EQTL3.SA', 'RADL3.SA', 'CGAS5.SA'] #, 'LREN3.SA',]

1. Mercado Financeiro

A economia se divide em duas correntes, a microeconomia e a macroeconomia. A microeconomia estuda de forma sucinta o comportamento das pessoas e empresas. A macroeconomia estuda os agregados econômicos, que são: os mercados de bens e serviços, o mercado de trabalho e o mercado financeiro. O mercado financeiro, por sua vez, é subdividido em mercado de crédito, mercado de capitais, mercado de câmbio e mercado monetário [1], [2]. Como este trabalho aborda apenas o mercado de capitais, este seria a única ramificação estudada dentro do mercado financeiro. Um dos principais componentes do mercado financeiro é a bolsa de valores, representada no Sistema Financeiro Nacional (SFN) e pela Comissão de Valores Mobiliários (CVM), que apresenta como principal finalidade manter condições para a compra e venda de títulos. No Brasil, a principal bolsa de valores é a Bolsa de Valores de São Paulo, Bovespa, instituição que atualmente realiza o comércio de títulos de forma totalmente eletrônica e é conhecida como Bolsa de Valores, Mercadorias e Futuros (BM&F). A maneira de demonstrar o desempenho médio ocorrido durante as negociações a partir das cotações das ações negociadas no mercado brasileiro é o índice IBOVESPA [3].

Entre os investimentos de risco não definidos, estão as ações, que são ativos de renda variável, ou seja, não existem garantias ao investidor sobre a sua rentabilidade, que dependerá do desempenho da empresa. Como formas de rendimento os investidores contam apenas com a participação nos resultados, com benefícios concedidos pela empresa e com os dividendos, além do eventual ganho de capital sobre a venda da ação no mercado.

1.1 Mercado Brasileiro

A composição das bolsas de valores que formam o mercado brasileiro e integram o Sistema Financeiro é fiscalizada pela Comissão de Valores Mobiliários (CVM). O objetivo da CVM é prover um ambiente com condições adequadas para a realização de operações de compra e venda de títulos e de valores mobiliários, bem como conservar valores éticos nas negociações realizadas em seu âmbito e disponibilizar os resultados das transações realizadas de maneira rápida. Dentre seus propósitos, destacam-se:

1. Proporcionar segurança e eficiência durante a liquidação das negociações realizadas;
2. Oferecer aos membros um sistema de negociação capaz de registrar e liquidar de forma segura as operações realizadas em títulos e valores mobiliários;
3. Fazer cumprir as normas e disposições legais que regem as operações em bolsa.

Denomina-se pregão o local onde são realizadas as operações de compra e venda de ações registradas em bolsa de valores. Os participantes do pregão, devidamente representados e credenciados pelas sociedades corretoras, devem possuir informações de acontecimentos relevantes que influenciam nos preços de negociação das ações, os quais devem ocorrer conforme as ordens de compra e venda ofertadas pelos membros. No âmbito brasileiro, considera-se que as negociações são realizadas de forma contínua, ou seja, as cotações são efetuadas durante o período de funcionamento das bolsas de valores, que compreende o período das 9 (nove) horas e 30 (trinta) minutos as 18 (dezoito) horas [3], [1].

1.2 Índice IBOVESPA

Os índices de preços de mercado tem como principal objetivo demonstrar o comportamento de determinado portfólio baseado em seu histórico de preços em determinado período de tempo, servindo como base para a análise dos preços das ações que o compõem [2]. Altamente confiável e de fácil metodologia, o IBOVESPA baseia-se em uma carteira teórica de ativos desde 1968. O índice IBOVESPA é o principal indicador de desempenho médio do mercado de ações brasileiro, refletindo o desempenho médio dos negócios realizados nos pregões da Bolsa

de Valores de São Paulo.

O IBOVESPA analisa os negócios realizados em ações no mercado à vista em tempo real, publicando estas informações instantaneamente online [3]. Esse índice demonstra em pontos a situação atual do mercado considerando as variações que aconteceram desde 1968. Nessa premissa, o IBOVESPA tem demonstrado o perfil dos negócios realizados na BM&FBovespa, que, por sua vez, realiza periodicamente atualizações hipotéticas no portfólio de ações, de maneira a incluir ou manter os papéis mais representativos, com maior grau de negociabilidade, e retirar os mais irrelevantes de determinado período. No momento, essa atualização é realizada a cada quadrimestre com base nas principais ações nos últimos 12 (doze) meses, formando um novo portfólio teórico. O cálculo desse índice não leva em consideração o ganho em qualquer outro investimento, apenas o retorno dos dividendos recebidos, o montante obtido com a venda dos direitos de subscrição e a manutenção das ações distribuídas a título de bonificação. Toda e qualquer ação tomada para definir o IBOVESPA é baseada no Manual de Definições e Procedimentos dos Índices da BM&FBovespa. Para a formação desse portfólio teórico, são selecionadas as ações mais representativas, ou seja, as que representam 80% do volume das transações no mercado à vista nos últimos 12 (doze) meses.

1.3 Mercados Financeiros

O mercado financeiro brasileiro é formado por instituições que recebem recursos financeiros, distribuem e circulam valores, sendo regidas pela regulação desse processo. O processo de captação de recursos mesmo como base para estudo é frequentemente difundido na prática, o que permite a interação de um amplo sistema de comunicação. Em comum, as instituições apresentam uma referência para as várias negociações financeiras e como principal moeda de troca está a taxa de juros. O processo de captação subdivide-se em cinco partes; são elas: mercado monetário, mercado de crédito, mercado de capitais, mercado cambial e mercado de derivados [3].

1.4 Mercado de Ações

Entre as diversas formas de investimento, como, por exemplo, os títulos, existe a possibilidade de investir em empresas. O mercado acionário capta recursos por meio de financiamento de dívidas, na forma de títulos ou empréstimos, e por meio de financiamento por participação acionária, ou seja, a emissão de ações. No mercado de ações, diferente do que

acontece com os títulos, não existe pagamentos pré-determinados, mas é permitido ao investidor capturar os dividendos, que, por sua vez, são pagos com os lucros da empresa, quando assim decidem os seus representantes [4]. Ações são ativos de renda variável, isto é, ao investidor não é disponibilizada garantia de rentabilidade conhecida antecipadamente.

É considerado um investimento de risco, uma vez que não oferece uma garantia de retorno. Os investidores têm como rentabilidade os dividendos ou a participação nos resultados e nos benefícios concedidos pela empresa emissora e também o eventual ganho de capital advindo da venda de ação no mercado (Bolsa de Valores). O retorno no investimento provém do desempenho da empresa e do comportamento das economias brasileira e internacional. Existem argumentos entre economistas de que o preço de determinada ação nem sempre corresponde ao seu Valor Fundamental e, em alguns momentos, essa ação pode encontrar-se subvalorizada ou sobrevalorizada; logo esta ação, em um período de tempo, poderá chegar a um colapso [4].

Uma ação é definida como a menor parcela do capital social de determinada empresa pertencente a uma sociedade por ações. Existem dois tipos de ações: as ordinárias, que permitem o direito a voto, e as preferenciais, que garantem a preferência sobre os lucros, dividendos e juros sobre capital próprio distribuídos aos acionistas. Com o Plano Collor, a emissão das ações passou a ser obrigatoriamente na forma nominativa ou escritural. Dividido em dois segmentos, o mercado de ações pode ser caracterizado como mercado primário, no qual as ações de uma empresa são emitidas diretamente ou através de oferta pública, ou como mercado secundário, no qual as ações já emitidas são comercializadas por meio das bolsas de valores [2]. As bolsas de valores são entidades que oferecem condições para que seus membros possam realizar compras e vendas de títulos, valores mobiliários e outros.

É denominado de pregão o ambiente em que as transações de compra e venda de ações são realizadas. Durante os pregões, apenas os operadores (representantes credenciados pelas Sociedades Corretoras) podem ter acesso e negociar ações segundo ordens de compra e venda enviadas por investidores. Empresas registradas em bolsa de valores estão submetidas a uma série de exigências, como dar amplo acesso às notícias e ao estado da empresa para o público. Aos participantes do pregão, essas informações são essenciais para que possam influir nos preços das ações negociadas. Integrantes do Sistema Financeiro Nacional, as Bolsas de Valores são fiscalizadas pela Comissão de Valores Mobiliários. O valor de cada ação na bolsa de valores é definido conforme o fluxo de oferta

e procura de cada papel. O aumento da demanda pode valorizar o mercado, assim como uma maior oferta de venda pode causar a sua desvalorização. Esse comportamento é definido por indicadores econômicos, por mudanças tecnológicas e pela situação atual e esperada da empresa [3]. No Brasil, a única bolsa de valores em operação é a Bolsa de Valores de São Paulo, que basicamente negocia ações em pregão. Geralmente as ações refletem as expectativas dos agentes econômicos em relação às perspectivas do País. Logo, as escolas Gráfica e Fundamentalista procuram entender as tendências de preço para tomarem uma decisão. A Escola Gráfica toma como base a análise gráfica, tanto pelo método de barras quanto pelo de ponto figura. Já a Escola Fundamentalista baseia-se em resultados setoriais e específicos de cada empresa. Existem quatro tipos de investidores: pessoas físicas, pessoas jurídicas, investidores externos e investidores institucionais. E desde que não provoque uma distorção de preços ou a manipulação do mercado, destaca-se a figura do especulador, que investe apenas com objetivo de ganho imediato, mas que por outro lado, garante o nível de liquidez do mercado. Os investidores institucionais desempenham papel fundamental, pois garantem o nível de estabilidade do mercado. De acordo com as normas de composição e diversificação do Conselho Monetário Nacional (CMN), os investidores institucionais estão sempre presentes no mercado, pois suas aplicações são compulsórias [2].

1.5 Risco do Mercado de Ações

O risco representa a probabilidade de perda associada ao capital do investidor. Algumas maneiras de reduzir a exposição ao risco estão na contratação de seguros, na diversificação e em outros instrumentos [3]. O risco está associado a eventos como desastres aéreos, acidentes de carro, roubos, etc. No caso, em uma avaliação em empresas, o risco total pode ser dividido em econômico e financeiro. No risco econômico, estão os fatores de natureza conjuntural, de mercado, de planejamento e de gestão da empresa. Quanto ao risco financeiro, este envolve o endividamento da empresa. Logo, pode-se assumir que o risco total de qualquer ativo está entre o sistemático ou conjuntural e o não sistemático. O risco sistemático é um conjunto de risco comum a todos os ativos, sendo este definido por eventos de natureza política, econômica e social; portanto, não existem maneiras de evitar completamente o risco sistemático e, como forma de solucionar o problema, é indicada a diversificação do portfólio, com fins de diminuir o risco. O risco não sistemático, ou diversificável, é um conjunto de risco e não afeta outros investimentos do portfólio e existe a

possibilidade de eliminar tal risco com a inclusão de novos ativos, que não contenham relações de risco entre si no portfólio [3]. O risco total é um conjunto de risco que pode ser calculado da seguinte maneira:

$$Risco\ Total = Risco\ sistemático + Risco\ Não\ sistemático$$

Entre os riscos financeiros, destacam-se o risco de crédito, o risco operacional e o risco de mercado [3]. O risco de mercado está associado principalmente à possibilidade de perdas devido às variações dos preços e à volatilidade no mercado financeiro. O risco de crédito existe quando a contraparte não está disposta a cumprir com suas obrigações contratuais. E o risco operacional é a possibilidade de perda devido a processos internos, sistemas e eventos externos [5].

Segundo Jorion [5], mesmo oferecendo uma intuição do risco, as primeiras ferramentas desenvolvidas para medir o risco não eram satisfatórias, pois não permitiam agregar os riscos de perda entre mercados e não mediam parâmetros como volatilidade, correlação e probabilidade de fatores adversos, ou seja, probabilidade de perda, parâmetros importantes para mensurar o risco de queda total do portfólio. Supondo que o investidor pretenda aplicar seu capital em dois investimentos diferentes, os riscos dos diferentes mercados não podiam ser agregados. Como uma forma de resolver o problema, foi desenvolvido o método estatístico conhecido como Valor em Risco (VaR) [3], [6] que permite o controle e a análise dos riscos entre mercados, considerando o valor máximo de perda esperado no portfólio e toda a distribuição dos lucros com um grau de confiabilidade baseado no horizonte de tempo [3]. O VaR permite agregar os riscos em todo o portfólio, levando em consideração a diversificação, de forma a mensurar o risco com uma probabilidade associada, permitindo ao investidor uma ideia do nível de confiança em percentual. Entretanto, a ideia por trás do VaR não é recente, sendo atribuída ao pioneirismo do modelo proposto por Markowitz e tendo sido resultado da fusão da teoria de instrumentos financeiros e da estatística [5].

1.6 Perfil do Investidor

A tolerância ao risco é fator determinante para definir o perfil do investidor. Na estrutura do mercado financeiro, é importante que se conheça o perfil do investidor para, com base nesses dados, formar um plano de investimento adequado a cada tipo de perfil. O perfil demonstra as noções das expectativas de risco, de quais são as necessidades e objetivos a curto, médio e longo prazos, além do volume de recursos disponíveis. Existem pessoas

que estão dispostas a correr maiores riscos do que outras e, portanto, tendem a suportar melhor as flutuações do mercado. Geralmente, a Risk Profile Analysis (RPA) trata-se de um questionário no qual os clientes respondem sobre suas particularidades; as respostas agrupadas mostram o perfil do investidor. No exterior, a RPA é conhecida como suitability e é bastante aplicada em países onde o mercado de investimentos é maduro. Conforme o questionário da Merrill Lynch, a instituição identifica três perfis básicos de risco: conservador, moderado e agressivo (ou dinâmico). Em síntese, o estilo conservador é o de menor tolerância ao risco, enquanto o estilo agressivo é o de maior tolerância ao risco.

O estilo conservador busca segurança em sua carteira de investimentos e um rendimento real e positivo, aceitando oscilações nos retornos dos seus investimentos, desde que eventuais instabilidades não ponham em risco seu patrimônio em médio prazo. Nesta classe, a carteira é formada por produtos de baixo risco, mas podendo-se investir uma pequena parcela em produtos que ofereçam níveis de riscos diferenciados, visando atingir ganhos melhores a longo prazo. Um investidor conservador quer ter ganhos, mas sua meta maior é não perder dinheiro [7], [8].

O estilo moderado busca maiores taxas de retorno com segurança nos seus investimentos; o investidor aceita possíveis perdas de capital como resultado natural. O grau de risco geralmente é reduzido pela diversificação de portfólio em produtos que gerem ganhos melhores no longo prazo, como, por exemplo, a renda variável, porém já aceitando uma volatilidade maior. Diversificar seus recursos é o ideal para sua estratégia de investimentos [7], [8].

O estilo agressivo é a classe de risco em que o investidor está disposto a aceitar perdas, experimentando uma maior variância dos seus retornos de um ano para o outro na expectativa de maiores ganhos em seus investimentos. Por isso, nesta classe, é possível aceitar maiores riscos, na esperança de que rendam em torno de 10%. Entretanto, mesmo para estratégias mais arrojadas, é necessário manter uma fatia de seus recursos em produtos de baixo risco, como forma de proteção do seu patrimônio [7], [8].

Quando se define o perfil do investidor, também estará definida a estratégia de investimentos para ele. É importante ressaltar que, assim como vantagens, todos os investimentos possuem riscos e é preciso conhecer a fundo

todos eles [8], [9].

Referências

- [1] Neto A. A. *Finanças Corporativas e Valor*. Atlas, Rua Conselheiro Nébias, 1384, Campos Elísios, São Paulo - SP, 2010.
- [2] Fortuna E. *Mercado Financeiro: Produtos e Serviços*. Qualitymark, Rua Teixeira Ju´nior, 441, São Cristovão, Rio de Janeiro - RJ, 2008.
- [3] Neto A. A. *Mercado Financeiro*. Editora Atlas, Rua Conselheiro Nébias, 1384, Campos Elísios, São Paulo - SP, 2009.
- [4] Blanchard O. *Macroeconomia*. Pearson Prentice Hall, 2007.
- [5] Jorion P. *Financial Risk Manager Handbook*. GARP, 2007.
- [6] Galdi F. C. and Pereira L. M. Valor em risco (VaR) utilizando modelos de previsão de volatilidade: Ewma, garch e volatilidade estocástica. *Brazilian Business Review*, 4:74–95, 2007.
- [7] Pasotto D., Pazo M. G., and Lobão S. V. *Uso de sistemas especialistas para decisão do perfil de um investidor*. PhD thesis, Universidade Presbiteriana Mackenzie, R. da Consolação (Pista Esquerda), 930 - Consolação, São Paulo - SP, junho 2003.
- [8] Souza C. R. V. *Avaliando questionários de risco e o comportamento do investidor sobre a ótica de Behavioral Finance*. PhD thesis, Fundação Getulio Vargas, Escola de Pós- Graduação em Economia, Praia de Botafogo 190, 22250-900 - Rio de Janeiro - RJ, agosto 2005.
- [9] Grable J. and Lytton R. H. Financial risk tolerance revisited: the development of a risk assessment instrument. *Financial Services Review*, 8:163–181, 1999.

Uma Introdução ao Python.

Responsáveis: Joás Rodrigues (IC ECEC)

1. Introdução

Python é uma linguagem de programação de alto nível e, que implementa vários paradigmas de programação, tais como a orientação a objetos, programação funcional. Algumas de suas principais características são:

- Tipagem dinâmica e forte.
- Multiparadigma.
- Fácil leitura de código.

A linguagem Python foi desenvolvida pelo matemático e programador holandês **Guido Von Rossum**, lançada em sua primeira versão no ano de 1991. É uma linguagem de código aberto e de desenvolvimento comunitário. É gerenciada pela Python Software Foundation.

As principais linguagens de programação a influenciar Python foram:

- ABC
- ALGOL 68
- Haskell
- Icon
- Java
- Lisp
- Modula-3
- Perl
- Smalltalk

As principais linguagens de programação que python influenciou foram:

- Boo
- D
- Falcon
- Fantom
- Groovy
- JavaScript
- Nimrod
- Py
- Ruby
- Squirrel
- Swift

2. Uma introdução simples

Como primeiro exemplo de uso da linguagem digite no terminal ou na IDE de sua escolha o seguinte comando abaixo:

```
print("Hello, World!")  
Hello, World!
```

3. Comentários

Ao decorrer da programação de alguma rotina, se faz necessário descrever o que certo trecho de código faz. Nesse sentido utilizamos comentários, que vão auxiliar o entendimento do código.

```
# Isso é um comentário de uma linha  
print("Hello, World!")
```

Caso queiramos comentar mais de uma linha, devemos seguir a regra do exemplo abaixo, digitar o comentário entre três aspas simples no início e três no final.

```
"""  
Isso é um comentário  
que contém mais de  
uma linha  
"""  
print("Hello, World!")
```

4. Variáveis

Um conceito objetivo para a ideia de variável é a seguinte: um container que tem como finalidade armazenar valores, sejam eles números, texto ou algum tipo heterogêneo de dados. O seguinte código abaixo nos dá um exemplo do uso de variáveis no Python.

```
x = 5  
y = 10  
print(x + y)
```

5. Tipos de dados

Os principais tipos de dados em python são:

Text Type:	<code>str</code>
Numeric Types:	<code>int</code> , <code>float</code> , <code>complex</code>
Sequence Types:	<code>list</code> , <code>tuple</code> , <code>range</code>
Mapping Type:	<code>dict</code>
Set Types:	<code>set</code> , <code>frozenset</code>
Boolean Type:	<code>bool</code>

Binary Types: `bytes`, `bytearray`, `memoryview`

Nesse breve manual nos concentraremos nos tipos: ***str***, ***int***, ***float*** e ***bool***, pois são elementares para qualquer tipo de operação que queiramos fazer. Os códigos abaixo utilizam cada um dos tipos e verificam seus tipos com uma função específica.

```
x = 5
print(type(x))
```

6. Entrada de dados

Para fazer as operações mais comuns, precisamos que sejam informados valores. Nesse caso precisamos de alguns meios para possibilitar a entrada de dados. Nesse texto iremos utilizar a entrada de dados pelo teclado. O código abaixo implementa a entrada de dados pelo teclado e armazena em uma variável.

```
valor = input ("Digite um valor: ")
```

Nesse tipo de leitura, os dados sempre serão armazenados como strings, caso você informe um número é necessário fazer um processo chamado casting, que é, converter os dados de entrada para o tipo que deseja. O código abaixo implementa essa operação:

```
valor = input("Digite um valor: ")

#para converter para int
valor = int(valor)

#para converter para float
valor = float(valor)
```

7. Tipos de dados

Os principais tipos de dados que iremos abordar são: *int*, *float*, *str* e *bool*

```
# tipo int
Valor = 5

#tipo float
Valor2 = 4.6

#tipo str
```

```
Nome = "william"
```

```
""" tipo bool. Uma variável do tipo bool pode receber apenas os
valores true ou false, que são palavras reservadas """
```

```
Resposta = True
```

```
Resposta2 = False
```

8. Operadores

Existem diversas operações que podemos fazer em uma linguagem de programação, operações matemáticas, operações lógicas, operações booleanas entre outras. As estruturas que implementam essas operações são chamadas operadores. Os principais operadores da linguagem Python são:

- Operadores aritméticos
- Operadores de atribuição
- Operadores de comparação
- Operadores lógicos
- Operadores de identidade
- Operadores de adesão
- Operadores bitwise

Iremos demonstrar o uso dos principais operadores

Operadores aritméticos

Nessa tabela o operador “=” é utilizado com o significado de igualdade.

Operador	Nome	exemplo	Código
+	Adição	x + y	<pre>x = 5 y = 3 print(x + y)</pre>
-	Subtração	x - y	<pre>x = 5 y = 3 print(x - y)</pre>

*	Multiplicação	$x * y$	<pre> x = 5 y = 3 print(x * y) </pre>
/	Divisão	x / y	<pre> x = 12 y = 3 print(x / y) </pre>
%	Resto de divisão inteira	$x \% y$	<pre> x = 5 y = 2 print(x \% y) </pre>
**	Exponenciação	$x ** y$	<pre> x = 2 y = 5 print(x ** y) #same as 2*2*2*2*2 </pre>
//	Floor division	$x // y$	<pre> x = 15 y = 2 print(x // y) #the floor division // rounds the result down to the nearest whole number </pre>

Operadores de atribuição

Operador	Exemplo	O mesmo que	Código
=	x = 5	x = 5	<pre>x = 5 print(x)</pre>
+=	x += 3	x = x + 3	<pre>x = 5 x += 3 print(x)</pre>
-=	x -= 3	x = x - 3	<pre>x = 5 x -= 3 print(x)</pre>
*=	x *= 3	x = x * 3	<pre>x = 5 x *= 3 print(x)</pre>
/=	x /= 3	x = x / 3	<pre>x = 5 x /= 3 print(x)</pre>
%=	x %= 3	x = x % 3	<pre>x = 5 x%=3 print(x)</pre>

//=	x //= 3	x = x // 3	<pre>x = 5 x//=3 print(x)</pre>
**=	x **= 3	x = x ** 3	<pre>x = 5 x **= 3 print(x)</pre>

Operadores de comparação

Operador	Nome	Exemplo	Código
==	Igualdade	x == y	<pre>x = 5 y = 3 print(x == y) # returns False because 5 is not equal to 3</pre>
!=	Diferença	x != y	<pre>x = 5 y = 3 print(x != y) # returns True because 5 is not equal to 3</pre>
>	Maior que	x > y	<pre>x = 5 y = 3</pre>

			<pre>print(x > y) # returns True because 5 is greater than 3</pre>
<	Menor que	$x < y$	<pre>x = 5 y = 3 print(x > y) # returns True because 5 is greater than 3</pre>
>=	Maior que ou igual a	$x \geq y$	<pre>x = 5 y = 3 print(x >= y) # returns True because five is greater, or equal, to 3</pre>
<=	Menor que ou igual a	$x \leq y$	<pre>x = 5 y = 3 print(x <= y) # returns False because 5 is neither less than or equal to 3</pre>

Operadores lógicos

Operator	Description	Example	Código
----------	-------------	---------	--------

and	Retorna Verdadeiro se ambas as condições forem verdadeiras	$x < 5$ and $x < 10$	<pre>x = 5 print(x > 3 and x < 10) # returns True because 5 is greater than 3 AND 5 is less than 10</pre>
or	Retorna verdadeiro se uma das condições for verdadeira	$x < 5$ or $x < 4$	<pre>x = 5 print(x > 3 or x < 4) # returns True because one of the conditions are true (5 is greater than 3, but 5 is not less than 4)</pre>
not	Reverte o resultado, retorna Falso se o resultado for verdadeiro	$\text{not}(x < 5 \text{ and } x < 10)$	<pre>x = 5 print(not(x > 3 and x < 10)) # returns False because not is used to reverse the result</pre>

9. Algumas funções úteis

Para a maioria das funções matemáticas em python, usualmente importamos a biblioteca "math". Veja abaixo o código que utiliza algumas funções básicas.

```
import math

# calcula o fatorial
math.factorial(10) # returns 3628800
# seno
math.sin(math.pi/4) # returns 0.7071067811865476
#cosseno
```

```
math.cos(math.pi)      # returns -1.0
#tangente
math.tan(math.pi/6)    # returns 0.5773502691896257

#logaritmo
# math.log(número, base)
math.log(8, 2)

# exponencial
math.exp(5) # o mesmo que e**5

#potenciação
math.pow(144, 0.5)
#raiz quadrada
math.sqrt(144)
```

Seguem resumos das Bibliotecas: Numpy, Pandas, Matplotlib, e Seaborn provenientes de <https://datacamp-community-prod.s3.amazonaws.com>

Python For Data Science Cheat Sheet

Matplotlib

Learn Python interactively at www.datacamp.com

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms.

1 Prepare The Data

Also see Lists & NumPy

1D Data

```
>>> import numpy as np
>>> x = np.linspace(0, 10, 100)
>>> y = np.cos(x)
>>> z = np.sin(x)
```

2D Data or Images

```
>>> data = 2 * np.random.random((10, 10))
>>> data2 = 3 * np.random.random((10, 10))
>>> V, X = np.mgrid[-3:3:100j, -3:3:100j]
>>> U = -1 - X**2 + Y
>>> V = 1 + X - Y**2
>>> from matplotlib.cbook import get_sample_data
>>> img = np.load(get_sample_data('axes_grid/bivariate_normal.npy'))
```

2 Create Plot

```
>>> import matplotlib.pyplot as plt
```

Figure

```
>>> fig = plt.figure()
>>> fig2 = plt.figure(figsize=plt.figaspect(2.0))
```

Axes

All plotting is done with respect to an Axes. In most cases, a subplot will fit your needs. A subplot is an axes on a grid system.

```
>>> fig.add_axes()
>>> ax1 = fig.add_subplot(221) # row-col-num
>>> ax3 = fig.add_subplot(212)
>>> fig3, axes = plt.subplots(nrows=2, ncols=2)
>>> fig4, axes2 = plt.subplots(ncols=3)
```

3 Plotting Routines

1D Data

```
>>> fig, ax = plt.subplots()
>>> lines = ax.plot(x, y)
>>> ax.scatter(x, y)
>>> axes[0,0].bar([1,2,3],[3,4,5])
>>> axes[1,0].barh([0.5,1.2,5],[0,1,2])
>>> axes[1,1].axhline(0.45)
>>> axes[0,1].axvline(0.65)
>>> ax.fill(x,y,color='blue')
>>> ax.fill_between(x,y,color='yellow')
```

Draw points with lines or markers connecting them
Draw unconnected points, scaled or colored
Plot vertical rectangles (constant width)
Plot horizontal rectangles (constant height)
Draw a horizontal line across axes
Draw a vertical line across axes
Draw filled polygons
Fill between y-values and 0

2D Data or Images

```
>>> fig, ax = plt.subplots()
>>> im = ax.imshow(img,
>>>                  cmap='gist_earth',
>>>                  interpolation='nearest',
>>>                  vmin=-2,
>>>                  vmax=2)
```

Colormapped or RGB arrays

Vector Fields

```
>>> axes[0,1].arrow(0,0,0.5,0.5)
>>> axes[1,1].quiver(y,z)
>>> axes[0,1].streamplot(X,Y,U,V)
```

Add an arrow to the axes
Plot a 2D field of arrows
Plot a 2D field of arrows

Data Distributions

```
>>> ax1.hist(y)
>>> ax3.boxplot(y)
>>> ax3.violinplot(z)
```

Plot a histogram
Make a box and whisker plot
Make a violin plot

Pseudocolor Plot of 2D Array

```
>>> axes2[0].pcolor(data2)
>>> axes2[0].pcolormesh(data)
>>> CS = plt.contour(V,X,U)
>>> axes2[2].contourf(data1)
>>> axes2[2] = ax.clabel(CS)
```

Pseudocolor plot of 2D array
Pseudocolor plot of 2D array
Plot contours
Plot filled contours
Label a contour plot

4 Customize Plot

Colors, Color Bars & Color Maps

```
>>> plt.plot(x, x, x, x**2, x, x**3)
>>> ax.plot(x, y, alpha = 0.4)
>>> ax.plot(x, y, c='k')
>>> fig.colorbar(img, orientation='horizontal')
>>> im = ax.imshow(img,
>>>                  cmap='seismic')
```

Markers

```
>>> fig, ax = plt.subplots()
>>> ax.scatter(x, y, marker='*')
>>> ax.plot(x, y, marker='o')
```

Linestyles

```
>>> plt.plot(x, y, linewidth=4.0)
>>> plt.plot(x, y, ls='solid')
>>> plt.plot(x, y, ls='--')
>>> plt.plot(x, y, '--', x**2, y**2, '-.')
>>> plt.setp(lines, color='r', linewidth=4.0)
```

Text & Annotations

```
>>> ax.text(1,
>>>        -2.1,
>>>        'Example Graph',
>>>        style='italic')
>>> ax.annotate("Bina",
>>>              xy=(8, 0),
>>>              xycoords='data',
>>>              xytext=(10.5, 0),
>>>              textcoords='data',
>>>              arrowprops=dict(arrowstyle="->",
>>>                              connectionstyle="arc3"),)
```

Math Text

```
>>> plt.title(r'Sigma_i=156', fontsize=20)
```

Limits, Legends & Layouts

Limits & Autoscaling

```
>>> ax.margins(x=0.0, y=0.1)
>>> ax.axis('equal')
>>> ax.set(xlim=(0,10.5), ylim=[-1.5,1.5])
>>> ax.set_xlim(0,10.5)
```

Legends

```
>>> ax.set(title='An Example Axes',
>>>        ylabel='Y-Axis',
>>>        xlabel='X-Axis')
>>> ax.legend(loc='best')
```

Ticks

```
>>> ax.xaxis.set(ticks=range(1,5),
>>>               ticklabels=[3,100,-12,"foo"])
>>> ax.tick_params(axis='y',
>>>                 direction='inout',
>>>                 length=10)
```

Subplot Spacing

```
>>> fig3.subplots_adjust(wspace=0.5,
>>>                       hspace=0.3,
>>>                       left=0.125,
>>>                       right=0.9,
>>>                       top=0.9,
>>>                       bottom=0.1)
```

Axis Spines

```
>>> ax1.spines['top'].set_visible(False)
>>> ax1.spines['bottom'].set_position(('outward',10))
```

Add padding to a plot
Set the aspect ratio of the plot to 1
Set limits for x and y-axis
Set limits for x-axis
Set a title and x and y-axis labels
No overlapping plot elements
Manually set x-ticks
Make y-ticks longer and go in and out
Adjust the spacing between subplots
Fit subplot(s) in to the figure area
Make the top axis line for a plot invisible
Move the bottom axis line outward

5 Save Plot

Save figures

```
>>> plt.savefig('foo.png')
```

Save transparent figures

```
>>> plt.savefig('foo.png', transparent=True)
```

6 Show Plot

```
>>> plt.show()
```

Close & Clear

```
>>> plt.cla()
>>> plt.clf()
>>> plt.close()
```


Clear an axis
Clear the entire figure
Close a window

<https://datacamp-community-prod.s3.amazonaws.com/28b8210c-60cc-4f13-b0b4-5b4f2ad4790b>

Python For Data Science Cheat Sheet

Pandas Basics

Learn Python for Data Science Interactively at www.DataCamp.com



Pandas

The Pandas library is built on NumPy and provides easy-to-use data structures and data analysis tools for the Python programming language.


Use the following import convention:

```
>>> import pandas as pd
```

Pandas Data Structures

Series

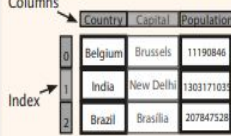
A one-dimensional labeled array capable of holding any data type



```
>>> s = pd.Series([3, -5, 7, 4], index=['a', 'b', 'c', 'd'])
```

DataFrame

A two-dimensional labeled data structure with columns of potentially different types



```
>>> data = {'Country': ['Belgium', 'India', 'Brazil'],
            'Capital': ['Brussels', 'New Delhi', 'Brasilia'],
            'Population': [11190846, 1303171035, 207847528]}
>>> df = pd.DataFrame(data,
                      columns=['Country', 'Capital', 'Population'])
```

Asking For Help

```
>>> help(pd.Series.loc)
```

Selection

Getting

```
>>> s['b']
-5
```

Get one element

```
>>> df[1:]
```

Get subset of a DataFrame

```
Country    Capital    Population
1    India    New Delhi    1303171035
2    Brazil    Brasilia    207847528
```

Selecting, Boolean Indexing & Setting

By Position

```
>>> df.iloc[[0], [0]]
'Belgium'
>>> df.iat[0, [0]]
'Belgium'
```

Select single value by row & column

By Label

```
>>> df.loc[0, ['Country']]
'Belgium'
>>> df.at[0, ['Country']]
'Belgium'
```

Select single value by row & column labels

By Label/Position

```
>>> df.ix[2]
Country    Brazil
Capital    Brasilia
Population    207847528
```

Select single row of subset of rows

```
>>> df.ix[:, 'Capital']
0    Brussels
1    New Delhi
2    Brasilia
```

Select a single column of subset of columns

```
>>> df.ix[1, 'Capital']
'New Delhi'
```

Select rows and columns

Boolean Indexing

```
>>> s[~(s > 1)]
>>> s[(s < -1) | (s > 2)]
>>> df[df['Population'] > 1200000000]
```

Series *s* where value is not >1
s where value is <-1 or >2
Use filter to adjust DataFrame

Setting

```
>>> s['a'] = 6
```

Set index *a* of Series *s* to 6

Dropping

```
>>> s.drop(['a', 'c'])
>>> df.drop('Country', axis=1)
```

Drop values from rows (axis=0)
Drop values from columns (axis=1)

Sort & Rank

```
>>> df.sort_index()
>>> df.sort_values(by='Country')
>>> df.rank()
```

Sort by labels along an axis
Sort by the values along an axis
Assign ranks to entries

Retrieving Series/DataFrame Information

Basic Information

```
>>> df.shape
>>> df.index
>>> df.columns
>>> df.info()
>>> df.count()
```

(rows, columns)
Describe index
Describe DataFrame columns
Info on DataFrame
Number of non-NA values

Summary

```
>>> df.sum()
>>> df.cumsum()
>>> df.min()/df.max()
>>> df.idxmin()/df.idxmax()
>>> df.describe()
>>> df.mean()
>>> df.median()
```

Sum of values
Cumulative sum of values
Minimum/maximum values
Minimum/Maximum index value
Summary statistics
Mean of values
Median of values

Applying Functions

```
>>> f = lambda x: x*2
>>> df.apply(f)
>>> df.applymap(f)
```

Apply function
Apply function element-wise

Data Alignment

Internal Data Alignment

NA values are introduced in the indices that don't overlap:

```
>>> s3 = pd.Series([7, -2, 3], index=['a', 'c', 'd'])
>>> s + s3
a    10.0
b     NaN
c     5.0
d     7.0
```

Arithmetic Operations with Fill Methods

You can also do the internal data alignment yourself with the help of the fill methods:

```
>>> s.add(s3, fill_value=0)
a    10.0
b     -5.0
c     5.0
d     7.0
>>> s.sub(s3, fill_value=2)
>>> s.div(s3, fill_value=4)
>>> s.mul(s3, fill_value=3)
```

I/O

Read and Write to CSV

```
>>> pd.read_csv('file.csv', header=None, nrows=5)
>>> df.to_csv('myDataFrame.csv')
```


Read and Write to Excel

```
>>> pd.read_excel('file.xlsx')
>>> df.to_excel('dir/myDataFrame.xlsx', sheet_name='Sheet1')
Read multiple sheets from the same file
>>> xlsx = pd.ExcelFile('file.xls')
>>> df = pd.read_excel(xlsx, 'Sheet1')
```

Read and Write to SQL Query or Database Table

```
>>> from sqlalchemy import create_engine
>>> engine = create_engine('sqlite:///memory:')
>>> pd.read_sql("SELECT * FROM my_table;", engine)
>>> pd.read_sql_table('my_table', engine)
>>> pd.read_sql_query("SELECT * FROM my_table;", engine)
read_sql() is a convenience wrapper around read_sql_table() and read_sql_query()
>>> df.to_sql('myDf', engine)
```

DataCamp
Learn Python for Data Science Interactively




<http://datacamp-community-prod.s3.amazonaws.com/dbed353d-2757-4617-8206-8767ab379ab3>

Python for Data Science Cheat Sheet

NumPy Basics

Learn Python for Data Science [Interactively](#) at [www.DataCamp.com](#)




NumPy

The **NumPy** library is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Use the following import convention:

```
>>> import numpy as np
```



NumPy Arrays

1D array

```
[1 2 3]
```

2D array

```
axis 1
axis 0
[[1.5 2. 3.]
 [4. 5. 6.]]
```

3D array

```
axis 2
axis 1
axis 0
```

Creating Arrays

```
>>> a = np.array([1,2,3])
>>> b = np.array([(1.5,2,3), (4,5,6)], dtype=float)
>>> c = np.array([(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)], dtype=float)
```

Initial Placeholders

```
>>> np.zeros(3,4)
>>> np.ones((2,3,4),dtype=np.int16)
>>> d = np.arange(10,25,5)
>>> np.linspace(0,2,9)
>>> e = np.full((2,2),7)
>>> f = np.eye(2)
>>> np.random.random((2,2))
>>> np.empty((3,2))
```

Create an array of zeros
Create an array of ones
Create an array of evenly spaced values (step value)
Create an array of evenly spaced values (number of samples)
Create a constant array
Create a 2x2 identity matrix
Create an array with random values
Create an empty array

I/O

Saving & Loading On Disk

```
>>> np.save('my_array', a)
>>> np savez('array.npz', a, b)
>>> np.load('my_array.npz')
```

Saving & Loading Text Files

```
>>> np.loadtxt('myfile.txt')
>>> np.genfromtxt('my_file.csv', delimiter=',')
>>> np.savetxt('myarray.txt', a, delimiter=" ")
```

Data Types

```
>>> np.int64
>>> np.float32
>>> np.complex
>>> np.bool
>>> np.object
>>> np.string_
>>> np.unicode_
```

Signed 64-bit integer types
Standard double-precision floating point
Complex numbers represented by 128 floats
Boolean type storing TRUE and FALSE values
Python object type
Fixed-length string type
Fixed-length unicode type

Inspecting Your Array

```
>>> a.shape
>>> len(a)
>>> b.ndim
>>> e.size
>>> b.dtype
>>> b.dtype.name
>>> b.astype(int)
```

Array dimensions
Length of array
Number of array dimensions
Number of array elements
Data type of array elements
Name of data type
Convert an array to a different type

Asking For Help

```
>>> np.info(np.ndarray.dtype)
```

Array Mathematics

Arithmetic Operations

```
>>> g = a - b
>>> array([-0.5, 0., 0.],
        [-3., -3., -3.])
>>> np.subtract(a,b)
>>> b + a
>>> array([[2.5, 4., 6.],
        [5., 7., 9.]])
>>> np.add(b,a)
>>> a / b
>>> array([[0.66666667, 1., 1.],
        [0.25, 0.4, 0.5]])
>>> np.divide(a,b)
>>> a * b
>>> array([[1.5, 4., 9.],
        [4., 10., 18.]])
>>> np.multiply(a,b)
>>> np.exp(b)
>>> np.sqrt(b)
>>> np.sin(a)
>>> np.cos(b)
>>> np.log(a)
>>> e.dot(f)
>>> array([[7., 7.],
        [7., 7.]])
```

Subtraction
Subtraction
Addition
Addition
Division
Division
Division
Multiplication
Multiplication
Exponentiation
Square root
Print sines of an array
Element-wise cosine
Element-wise natural logarithm
Dot product

Comparison

```
>>> a == b
>>> array([[False,  True,  True],
        [False, False, False]], dtype=bool)
>>> a < 2
>>> array([[True,  False, False],
        [True,  True,  True]])
>>> np.array_equal(a,b)
```

Element-wise comparison
Element-wise comparison
Array-wise comparison

Aggregate Functions

```
>>> a.sum()
>>> a.min()
>>> b.max(axis=0)
>>> b.cumsum(axis=1)
>>> a.mean()
>>> b.median()
>>> a.corrcoeff()
>>> np.std(b)
```

Array-wise sum
Array-wise minimum value
Maximum value of an array row
Cumulative sum of the elements
Mean
Median
Correlation coefficient
Standard deviation

Copying Arrays

```
>>> h = a.view()
>>> np.copy(a)
>>> h = a.copy()
```

Create a view of the array with the same data
Create a copy of the array
Create a deep copy of the array

Sorting Arrays

```
>>> a.sort()
>>> c.sort(axis=0)
```

Sort an array
Sort the elements of an array's axis

Subsetting

```
>>> a[2]
>>> b[1,2]
>>> b[0:2,1]
>>> array([1., 2., 3.])
>>> b[1:]
>>> array([[1.5, 2., 3.],
        [4., 5., 6.]])
>>> c[1,...]
>>> array([[3., 2., 1.],
        [4., 5., 6.]])
>>> a[: -1:]
>>> array([3, 2, 1])
>>> a[a<2]
>>> array([1])
>>> b[[1, 0, 1, 0], [0, 1, 2, 0]]
>>> array([[4., 2., 6., 1.5],
        [1.5, 2., 3., 1.],
        [4., 5., 6., 1.5],
        [1.5, 2., 3., 1.]])
```

Select the element at the 2nd index
Select the element at row 1 column 2 (equivalent to b[1][2])
Select items at index 0 and 1
Select items at rows 0 and 1 in column 1
Select all items at row 0 (equivalent to b[0:1, :])
Same as [1, :, :]
Reversed array a
Select elements from a less than 2
Select elements (1,0),(0,1),(1,2) and (0,0)
Select a subset of the matrix's rows and columns

Boolean Indexing

```
>>> a[a<2]
>>> array([1])
```

Fancy Indexing

```
>>> b[[1, 0, 1, 0], [0, 1, 2, 0]]
>>> array([[4., 2., 6., 1.5],
        [1.5, 2., 3., 1.],
        [4., 5., 6., 1.5],
        [1.5, 2., 3., 1.]])
```

Array Manipulation

Transposing Array

```
>>> i = np.transpose(b)
>>> i.T
```

Permute array dimensions
Permute array dimensions

Changing Array Shape

```
>>> b.ravel()
>>> g.reshape(3,-2)
```

Flatten the array
Reshape, but don't change data

Adding/Removing Elements

```
>>> h.resize((2,6))
>>> np.append(h,g)
>>> np.insert(a, 1, 5)
>>> np.delete(a,[1])
```

Return a new array with shape (2,6)
Append items to an array
Insert items in an array
Delete items from an array

Combining Arrays

```
>>> np.concatenate((a,d),axis=0)
>>> array([1, 2, 3, 10, 15, 20])
>>> np.vstack((a,b))
>>> array([[1., 2., 3.],
        [1.5, 2., 3.],
        [4., 5., 6.]])
>>> np.r_[a,e]
>>> np.hstack((e,f))
>>> array([[7., 7., 1., 0.],
        [7., 7., 0., 1.]])
>>> np.column_stack((a,d))
>>> array([[1, 10],
        [2, 15],
        [3, 20]])
>>> np.c_[a,d]
```

Concatenate arrays
Stack arrays vertically (row-wise)
Stack arrays vertically (row-wise)
Stack arrays horizontally (column-wise)
Create stacked column-wise arrays
Create stacked column-wise arrays


Splitting Arrays

```
>>> np.hsplit(a,(3))
>>> [array([1]),array(2),array(3)]
>>> np.vsplit(c,(2))
>>> [array([[1.5, 2., 1.],
        [4., 5., 6.]])
        array([[3., 2., 3.],
        [4., 5., 6.]])]
```

Split the array horizontally at the 3rd index
Split the array vertically at the 2nd index

DataCamp

Learn Python for Data Science [Interactively](#)



Python For Data Science Cheat Sheet 3 Plotting With Seaborn

Seaborn
Learn Data Science Interactively at www.DataCamp.com

Statistical Data Visualization With Seaborn

The Python visualization library Seaborn is based on matplotlib and provides a high-level interface for drawing attractive statistical graphics.

Make use of the following aliases to import the libraries:

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
```

The basic steps to creating plots with Seaborn are:

1. Prepare some data
2. Control figure aesthetics
3. Plot with Seaborn
4. Further customize your plot

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
>>> tips = sns.load_dataset("tips")
>>> sns.set_style("whitegrid")
>>> g = sns.lmplot(x="time", y="total_bill", data=tips, aspect=2)
>>> g = (g.set_axis_labels("Time", "Total bill (USD)")).set(xlim=(0,10), ylim=(0,100))
>>> plt.title("Title")
>>> plt.show()
```

1 Data

Also see [Lists, NumPy & Pandas](#)

```
>>> import pandas as pd
>>> import numpy as np
>>> uniform_data = np.random.rand(10, 12)
>>> data = pd.DataFrame({'x': np.arange(1,101), 'y': np.random.normal(0,4,100)})
```

Seaborn also offers built-in data sets:

```
>>> titanic = sns.load_dataset("titanic")
>>> iris = sns.load_dataset("iris")
```

2 Figure Aesthetics

Also see [Matplotlib](#)

Context Functions

```
>>> f, ax = plt.subplots(figsize=(5,6))
>>> sns.set_context("talk")
>>> sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth":2.5})
```

Seaborn styles

```
>>> sns.set()
>>> sns.set_style("whitegrid")
>>> sns.set_style("ticks", {"tick.major.size":8, "tick.minor.size":4})
>>> sns.axes_style("whitegrid")
```

(Re)set the seaborn default
Set the matplotlib parameters
Set the matplotlib parameters
Return a dict of params or use with with to temporarily set the style

Color Palette

```
>>> sns.set_palette("husl", 3)
>>> sns.color_palette("husl")
>>> flutui = ["#f05b4f", "#4398bb", "#95a5a6", "#46788c", "#34495e", "#2ecc71"]
>>> sns.set_palette(flutui)
```

Define the color palette
Use with with to temporarily set palette
Set your own color palette

3 Plotting With Seaborn

Axis Grids

```
>>> g = sns.FacetGrid(titanic, col="survived", row="sex")
>>> g = g.map(plt.hist, "age")
>>> sns.factorplot(x="pclass", y="survived", hue="sex", data=titanic)
>>> sns.lmplot(x="sepal_width", y="sepal_length", hue="species", data=iris)
```

Subplot grid for plotting conditional relationships
Draw a categorical plot onto a FacetGrid
Plot data and regression model fits across a FacetGrid

```
>>> h = sns.PairGrid(iris)
>>> h = h.map(plt.scatter)
>>> sns.pairplot(iris)
>>> i = sns.JointGrid(x="x", y="y", data=data)
>>> i = i.plot(sns.regplot, sns.distplot)
>>> sns.jointplot("sepal_length", "sepal_width", data=iris, kind="kde")
```

Subplot grid for plotting pairwise relationships
Plot pairwise bivariate distributions
Grid for bivariate plot with marginal univariate plots
Plot bivariate distribution

Categorical Plots

Scatterplot

```
>>> sns.stripplot(x="species", y="petal_length", data=iris)
>>> sns.swarmplot(x="species", y="petal_length", data=iris)
```

Scatterplot with one categorical variable
Categorical scatterplot with non-overlapping points

Bar Chart

```
>>> sns.barplot(x="sex", y="survived", hue="class", data=titanic)
```

Show point estimates and confidence intervals with scatterplot glyphs

Count Plot

```
>>> sns.countplot(x="deck", data=titanic, palette="Greens_d")
```

Show count of observations

Point Plot

```
>>> sns.pointplot(x="class", y="survived", hue="sex", data=titanic, palette={"male": "g", "female": "m"}, markers=["o", "x"], linestyle=["-", "--"])
```

Show point estimates and confidence intervals as rectangular bars

Boxplot

```
>>> sns.boxplot(x="alive", y="age", hue="adult_male", data=titanic)
>>> sns.boxplot(data=iris, orient="h")
```

Boxplot
Boxplot with wide-form data

Violinplot

```
>>> sns.violinplot(x="age", y="sex", hue="survived", data=titanic)
```

Violin plot

Regression Plots

```
>>> sns.regplot(x="sepal_width", y="sepal_length", data=iris, ax=ax)
```

Plot data and a linear regression model fit

Distribution Plots

```
>>> plot = sns.distplot(data.y, kde=True, color="b")
```

Plot univariate distribution

Matrix Plots

```
>>> sns.heatmap(uniform_data, vmin=0, vmax=1)
```

Heatmap

4 Further Customizations

Also see [Matplotlib](#)

Axisgrid Objects

```
>>> g.despine(left=True)
>>> g.set_ylabels("Survived")
>>> g.set_xticklabels(rotation=45)
>>> g.set_axis_labels("Survived", "Sex")
>>> h.set(xlim=(0,5), ylim=(0,5), xticks=[0,2.5,5], yticks=[0,2.5,5])
```

Remove left spine
Set the labels of the y-axis
Set the tick labels for x
Set the axis labels
Set the limit and ticks of the x-and y-axis

Plot

```
>>> plt.title("A Title")
>>> plt.ylabel("Survived")
>>> plt.xlabel("Sex")
>>> plt.ylim(0,100)
>>> plt.xlim(0,10)
>>> plt.setp(ax, yticks=[0,5])
>>> plt.tight_layout()
```

Add plot title
Adjust the label of the y-axis
Adjust the label of the x-axis
Adjust the limits of the y-axis
Adjust the limits of the x-axis
Adjust a plot property
Adjust subplot params

5 Show or Save Plot

Also see [Matplotlib](#)

```
>>> plt.show()
>>> plt.savefig("foo.png")
>>> plt.savefig("foo.png", transparent=True)
```

Show the plot
Save the plot as a figure
Save transparent figure

Close & Clear

Also see [Matplotlib](#)

```
>>> plt.cla()
>>> plt.clf()
>>> plt.close()
```

Clear an axis
Clear an entire figure
Close a window

DataCamp
Learn Python for Data Science Interactively

<https://datacamp-community-prod.s3.amazonaws.com/f9f06e72-519a-4722-9912-b5de742dbac4>

Alguns links para consultas complementares.

Fontes:

<https://research.google.com/colaboratory/faq.html>
https://pt.wikipedia.org/wiki/Projeto_Jupyter#cite_note-2
<https://www.geeksforgeeks.org/how-to-use-google-colab/>
https://www.tutorialspoint.com/google_colab/google_colab_quick_guide.htm
<https://realpython.com/python-modules-packages/>
<http://lucasparreiras.cc/2017/10/10/top-15-bibliotecas-python-para-ciencias-de-dados/>

Código e Comandos:

<https://docs.scipy.org/doc/numpy-1.14.0/reference/routines.random.html>
<http://felipegalvao.com.br/blog/2016/03/31/estatistica-descritiva-com-python/>
<https://www.statsmodels.org/stable/index.html>
<http://www.statsmodels.org/stable/stats.html>
<https://docs.scipy.org/doc/numpy/user/basics.html>
<https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html>

<https://docs.python.org/3/library/statistics.html>
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.table.html
<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.describe.html>
https://www.inf.ufsc.br/~andre.zibetti/probabilidade/aed.html#vari%C3%A1vel_quantitativa_discreta
<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.relfreq.html>
<https://paulovasconcellos.com.br/28-comandos-%C3%BAteis-de-pandas-que-talvez-voc%C3%AA-n%C3%A3o-conhe%C3%A7a-6ab64beefa93>
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.qcut.html>
<https://www.sangakoo.com/en/unit/absolute-relative-cumulative-frequency-and-statistical-tables>
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.value_counts.html
<https://seaborn.pydata.org/generated/seaborn.barplot.html>
https://seaborn.pydata.org/generated/seaborn.color_palette.html#seaborn.color_palette
<https://seaborn.pydata.org/generated/seaborn.countplot.html#seaborn.countplot>
<https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.cut.html>
<https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.cov.html>
<https://docs.scipy.org/doc/numpy/reference/generated/numpy.corrcoef.html>
<https://support.minitab.com/pt-br/minitab/18/help-and-how-to/modeling-statistics/anova/supporting-topics/anova-statistics/what-is-the-variance-covariance-matrix/>
<https://minerandodados.com.br/dominando-o-pandas-datascience-dozero/>
<https://seaborn.pydata.org/generated/seaborn.scatterplot.html>
<https://seaborn.pydata.org/generated/seaborn.heatmap.html>
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.crosstab.html>
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html>
<https://heartbeat.fritz.ai/seaborn-heatmaps-13-ways-to-customize-correlation-matrix-visualizations-f1c49c816f07>
<https://metodosdigitaisufg.wordpress.com/2018/02/24/analise-de-dados-utilizando-python/>
https://seaborn.pydata.org/examples/many_pairwise_correlations.html
<https://seaborn.pydata.org/tutorial/regression.html>
<https://seaborn.pydata.org/generated/seaborn.lmplot.html>
<https://seaborn.pydata.org/generated/seaborn.regplot.html>
<https://docs.scipy.org/doc/numpy/reference/generated/numpy.cumsum.html>
<https://docs.scipy.org/doc/numpy/reference/generated/numpy.cumsum.html>