

## Material:

**Q: Onde posso encontrar os materiais das aulas do professor Seiji e professor Mangan?**

R: Ao fazer o download do material de apoio que consta no ícone de folha a4 ao lado do título da disciplina das aulas do professor Seiji é possível encontrar todos os links para os na apresentação PPT disponível.

A exceção consta próximo ao minuto 06 da parte 2 da aula 2, onde o professor Seiji abre o link do artigo sobre ensembles, segue o link do artigo:

<https://www.datasciencecentral.com/profiles/blogs/want-to-win-at-kaggle-pay-attention-to-your-ensembles>.

Sobre o link do Scipy Lecture é exatamente o que o professor utilizou para guiar a aula em questão de conceitos do Python.

O professor Mangan utilizou a documentação da Scipy para lecionar suas aulas. O material está disponível em:

<https://scipy-lectures.org>

O professor Mangan utilizou a tradução automática gerada pelo programa Tradutor da Google:

<https://translate.google.com/translate?sl=en&tl=pt&u=https://scipy-lectures.org>

## Ferramentas:

**Q: Nos instantes 37:40 e 49:50 da parte 2 da aula 2 o professor cita duas bibliotecas, uma para a área de teste e outra para área de reconhecimento de voz. Há algum link que forneça mais informações sobre estas duas ferramentas?**

R: Quanto a qualidade de código, é possível que o professor tenha se referido à biblioteca *Structure 101* disponível em: <https://structure101.com/products/studio/>.

O professor também se referia à biblioteca *Kaldi* disponível em:

<https://github.com/kaldi-asr/kaldi> para transcrição de voz.

Quanto à biblioteca *Understand* não encontramos nenhuma referência.

## Conceitos e Exercícios:

**Q: Tem algum material para indicar para estudar um pouco sobre gráficos, tipos e suas aplicações? Não necessariamente relacionada a alguma tecnologia como Python, para um estudo mais teórico.**

R: O curso oferece uma ótima disciplina sobre visualização de dados com a prof<sup>a</sup> Isabel e há uma demonstração de alguns com o professor Renan Xavier na disciplina de Introdução à Ciência de Dados e à Inteligência Artificial.

Quanto à tecnologia Python, há algumas bibliotecas que permitem algumas explorações visuais como:

Seaborn: <https://seaborn.pydata.org/>

Matplotlib: <https://matplotlib.org/>

Uma recomendação do professor Mangan para esse tópico também é um livro que trata sobre como usar gráficos:

*Knaflitz, Cole Nussbaumer (2015) Storytelling with data: a data visualization guide for business professionals.*

Na biblioteca Central da PUCRS: [http://primo-pmtna01.hosted.exlibrisgroup.com/PUC01:PUC01:oclcwiley\\_e000xww\\_923807870](http://primo-pmtna01.hosted.exlibrisgroup.com/PUC01:PUC01:oclcwiley_e000xww_923807870)

**Q: Onde consigo acesso aos links do drive com o material, por favor?**

Os links encontram-se no material de apoio, ao lado do nome da disciplina possui um ícone A4, ele levará aos materiais utilizados pelo professor, na aula 1 há um PDF, nele você encontra os links dos arquivos notebooks do professor com as respostas.

**Olá. Eu estava executando o exercício presente no minuto 12 da aula e recebi uma mensagem de erro. Ao pesquisar no *stackoverflow*, descobri o seguinte: o módulo *sklearn.datasets.samples.generator* foi substituído por *sklearn.datasets*. Sem essa substituição, o exercício dá erro.**

O comando em funciona normalmente dependendo da versão que se está utilizando, talvez seja o caso de utilizar ou esse comando que você indicado, ou instalar o pacote de *samples* do *scikit learning*.

**Q: Boa tarde! Não consegui descobrir como importar o *logistic regression* para o Python, quero dizer para o *Google Colaborate* .**

R: O *logistic regression* é uma *feature* do *scikit-learn*, a princípio, se você executar o comando `"from sklearn.linear_model import LogisticRegression"` que tem no *Google Colab* ele deve executar e ficar com um "checkzinho verde" ao lado da célula.

Caso queira instalar o *scikit learn* você pode executar o comando `"pip install scikit-learn"` que ele irá instalar os pacotes do *scikit* no seu ambiente.

**Q: Conforme imagem pcdd01.jpg, foi questionado sobre como executar a soma dos valores de um vetor a. Eu respondi *a.sum()* e foi considerado como errado. Porém, de acordo com a documentação na imagem pcdd02.jpg (ou link <http://scipy-lectures.org/intro/numpy/operations.html#computing-sums>), minha resposta está correta e a alternativa marcada como a certa é que está errada. Para fazer a soma utilizando a estrutura *sum(a)* seria necessário referenciar ao pacote *numpy* , com a seguinte estrutura: *numpy.sum(a)*.**

O gabarito está correto mesmo. Segue:

A premissa da questão é tratar das funções primitivas do Python, ou seja, as que não precisam de uma biblioteca como a *numpy*.

No seu próprio comentário, você indica soma de valores de um VETOR, mas a questão indica LISTA.

Para que a opção escolhida por você fosse correta, você deveria transformar a lista em um vetor de valores *numpy* "a" e depois sim, executar o comando *a.sum()*.

Por último, mas não menos importante, ao usar o *numpy* você deixa de atender ao texto-base. O *numpy* é uma biblioteca adicional.

*\*FAQ gerado com base em comentários até o dia 05/12/2021.*