Análise de Dados

“Science is knowledge which we understand so well that
we can teach it to a computer.
Everything else is art.”

Donald Knuth (1974) - Computer Programming as an Art

Estrutura da Análise de Dados

1. Definir a questão
2. Definir o conjunto de dados ideal

   (os dados irão depender dos objetivos: descritivo; exploratório; inferência; preditivo; causal; mecânico)
3. Determinar quais dados você pode acessar
4. Obter os dados (documentar)
5. Limpar os dados
6. Análise exploratória
7. Modelagem/previsão estatística
8. Interpretação dos resultados
9. Questionar os resultados

   (todos os passos anteriores: questão; fonte de dados; processamento; análise; conclusões)
10. Sintetizar/escrever os resultados
11. Criar resultados reprodutíveis

Estrutura da Análise de Dados

1. Definir a questão
2. Definir o conjunto de dados ideal
3. Determinar quais dados você pode acessar
4. Obter os dados
5. Limpar os dados
6. Análise exploratória
7. Modelagem/previsão estatística
8. Interpretação dos resultados
9. Questionar os resultados
10. Sintetizar/escrever os resultados
11. Criar resultados reprodutíveis

Objetivo: Desenvolver tópicos de interesse dentro desses assuntos:

I. Programação no R
II. Importar, Limpar e Manipular Dados
III. Análise Exploratória de Dados
IV. Pesquisa Reprodutível
V. Inferência Estatística
VI. Modelos de Regressão e Econometria
VII. Machine Learning

Como fazer?

- Documentação dos assuntos discutidos: códigos, roteiros, conteúdo, fórum
- Software padrão: R
- Organização de dados
- Sistema de controles de versão
- Foco em aplicações:
  - Desenvolvimento prático com aplicações reais com fins de agregar para as pesquisas
  - Ênfase em aplicações de modelagem, estimação e inferência
- Contribuição de todos

Organizando os dados

- Regras da análise de dados
- Dados brutos e processados
- Figuras: exploratória, final
- Código R:
  - Bruto: scripts não utilizados
  - Scripts finais
  - Arquivos de roteiro
- Texto:
  - arquivos readme
  - Texto da análise / relatório

Controle de Versão

- VCS (do inglês version control system); ou
- SCM (do inglês source code management)
- Gerenciar diferentes versões no desenvolvimento de um documento qualquer.
- Utilizados no desenvolvimento de software para controlar as diferentes versões — histórico e desenvolvimento — dos códigos-fontes e também da documentação.
- Principais vantagens:
  - Controle do histórico
  - Trabalho em equipe
  - Marcação e resgate de versões estáveis
  - Ramificação de projeto

Sistemas de Controle de versão

- Git
  - é um sistema de controle de versão distribuído e um sistema de gerenciamento de código fonte, com ênfase em velocidade.
  - inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do kernel Linux, mas foi adotado por muitos outros projetos.
  - Cada diretório de trabalho do Git é um repositório.
  - Linha de comando – *code school*: https://try.github.io/

- Github
  - Serviço de Web Hosting Compartilhado para projetos que usam o controle de versionamento Git
  - **Fazer conta**: https://github.com/
  - Software amigável: Github desktop: **instalar**: https://desktop.github.com/