Effective Programming Practices for Economists

Data management with pandas

Functional data cleaning: The Why

Janoś Gabler and Hans-Martin von Gaudecker

Three simple rules for data cleaning

1. Start with an empty DataFrame
2. Touch every variable just once
3. Touch with a pure function

Reproducibility & researchers' sanity

1. Code must be readable at different levels of abstraction
2. Pipeline always starts from the original data, which is never modified
3. Must be able to find out quickly what a variable contains
4. There must not be state in any column's contents
5. Data management and analysis should be separated

1. Readability at different levels

- Same as always — also works for data management!
- Highest level: read - manage - write
- Middle level: `clean_agreement_scale`
- Lowest level: `sr.replace({"-77": pd.NA, "-99": pd.NA})`

1. Readability at different levels

1. Start with an empty DataFrame
2. **Touch every variable just once**
3. **Touch with a pure function**

2. Pipeline always starts from the original data, which is never modified

- **Source data**: Original dataset as downloaded or collected
- *(Commit the source data to VC and)* never change the source data.
- All modified datasets should be stored under different names
- Modified datasets should not be under version control!

2. Pipeline always starts from the original data, which is never modified

1. **Start with an empty DataFrame**
2. Touch every variable just once
3. **Touch with a pure function**

3. Quickly find out variable contents

- Debugging (making sense of code and/or results) is hard
- If you need to look in 5 places to understand a variable's contents, you will go insane
- Typical case:

  1. Regression results are unexpected
  2. Code seems correct
  3. Need to understand exactly what each variable contains

  Implies that you need to track provenance of each variable from source to what is used in the regression.

3. Quickly find out variable contents

1. Start with an empty DataFrame
2. **Touch every variable just once**
3. **Touch with a pure function**

Implies that you will find the correct spot by searching for regex:

```
coding_genius.+ =
```

(spaces around `=`)

4. No state in any column’s contents

- Say you made sense of your results after you checked the first place where you touched a variable
- Realize much later it didn’t quite make sense
- Only then find out you actually changed the variable midstream
- Put differently, state means you will go insane even more quickly.

4. No state in any column’s contents

1. **Start with an empty DataFrame**
2. **Touch every variable just once**
3. **Touch with a pure function**

5. Separate data mgm’t and analysis

- **Data management**: Converting source data to formats your analysis programs need
- Separate data management code from analysis code
- Never modify the content of a variable outside the data management code!

5. Separate data mgm’t and analysis

Corrollary of 4.

```
# Load
raw_survey = pd.read_csv("../management_definitions_example/survey.csv")
# Manage / clean up
cleaned_survey = clean_data(raw_survey)
# Save
cleaned_survey.to_feather("bld/survey_cleaned.feather")
```