Effective Programming Practices for Economists

Data management with pandas

Functional data cleaning: The How

Janoś Gabler and Hans-Martin von Gaudecker

Three simple rules for cleaning data

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

Survey of course participants

	Q001	Q002	Q003
0	strongly disagree	agree	python
1	strongly agree	strongly agree	Python
2	-77	disagree	R
3	agree	-77	Python
4	-99	-99	Python
5	nan	strongly agree	Python
6	neutral	strongly agree	Python
7	disagree	agree	python
8	strongly agree	-99	PYTHON
9	agree	-99	Ypthon

- Q001: I am a coding genius
- Q002: I learned a lot
- Q003: What is your favourite language?

- -77 not readable
- -99 no reply

- All variables are strings after running

```
raw_survey = pd.read_csv("survey.csv")
```

Code

```
def clean_data(raw):
    df = pd.DataFrame(index=raw.index)
    df["coding_genius"] = clean_agreement_scale(raw["Q001"])
    df["learned_a_lot"] = clean_agreement_scale(raw["Q002"])
    df["favorite_language"] = clean_favorite_language(raw["Q003"])
    return df

def clean_agreement_scale(sr):
    sr = sr.replace({"-77": pd.NA, "-99": pd.NA})
    categories = ["strongly disagree", "disagree", "neutral", "agree", "strongly agree"]
    dtype = pd.CategoricalDtype(categories=categories, ordered=True)
    return sr.astype(dtype)

def clean_favorite_language(sr):
    sr = sr.str.lower().str.strip()
    sr = sr.replace("ypthon", "python")
    return sr.astype(pd.CategoricalDtype())


raw_survey = pd.read_csv("survey.csv")
cleaned_survey = clean_data(raw_survey)
cleaned_survey.to_feather("bld/survey_cleaned.feather")
```

Result

	coding_genius	learned_a_lot	favorite_language
0	strongly disagree	agree	python
1	strongly agree	strongly agree	python
2	NaN	disagree	r
3	agree	NaN	python
4	NaN	NaN	python
5	NaN	strongly agree	python
6	neutral	strongly agree	python
7	disagree	agree	python
8	strongly agree	NaN	python
9	agree	NaN	python

```
>>> df.dtypes
coding_genius     category
learned_a_lot     category
favorite_language category
dtype: object

>>> df["coding_genius"].cat.categories
[
    'strongly disagree',
    'disagree',
    'neutral',
    'agree',
    'strongly agree'
]

>>> df["favorite_language"].cat.categories
['python', 'r']
```

1. Start with an empty DataFrame

```
def clean_data(raw):
    df = pd.DataFrame(index=raw.index)
    df["coding_genius"] = clean_agreement_scale(raw["Q001"])
    df["learned_a_lot"] = clean_agreement_scale(raw["Q002"])
    df["favorite_language"] = clean_favorite_language(raw["Q003"])
    return df
```

2. Touch every variable just once

```
def clean_data(raw):
    df = pd.DataFrame(index=raw.index)
    df["coding_genius"] = clean_agreement_scale(raw["Q001"])
    df["learned_a_lot"] = clean_agreement_scale(raw["Q002"])
    df["favorite_language"] = clean_favorite_language(raw["Q003"])
    return df
```

3. Touch with a pure function

```
def clean_agreement_scale(sr):
    sr = sr.replace({"-77": pd.NA, "-99": pd.NA})
    categories = ["strongly disagree", "disagree", "neutral", "agree", "strongly agree"]
    dtype = pd.CategoricalDtype(categories=categories, ordered=True)
    return sr.astype(dtype)


def clean_favorite_language(sr):
    sr = sr.str.lower().str.strip()
    sr = sr.replace("ypthon", "python")
    return sr.astype(pd.CategoricalDtype())
```

3. Touch with a pure function

- Sometimes you just need to rename a variable
- Remember that the identity function (doing nothing) is a pure function
- So that would be:

```
df["sensible_name"] = raw["sxn3"]
```

Python scripts vs. Jupyter notebooks

```
def clean_data(raw):
    df = pd.DataFrame(index=raw.index)
    df["coding_genius"] = clean_agreement_scale(raw["Q001"])
    df["learned_a_lot"] = clean_agreement_scale(raw["Q002"])
    df["favorite_language"] = clean_favorite_language(raw["Q003"])
    return df

def clean_agreement_scale(sr):
    ...

def clean_favorite_language(sr):
    ...

raw_survey = pd.read_csv("survey.csv")
cleaned_survey = clean_data(raw_survey)
cleaned_survey.to_feather("bld/survey_cleaned.feather")
```

Python scripts vs. Jupyter notebooks

```
df = pd.DataFrame(index=raw.index)


def clean_agreement_scale(sr):
    ...


df["coding_genius"] = clean_agreement_scale(raw["Q001"])


df["learned_a_lot"] = clean_agreement_scale(raw["Q002"])


def clean_favorite_language(sr):
    ...


df["favorite_language"] = clean_favorite_language(raw["Q003"])
```