Python for Data Science

Ch. 4
Data Cleaning

Ahmad Rio Adriansyah

Data Cleaning

- Kadang disebut sebagai Cleansing
- Proses memfilter, memperbaiki, menghapus, atau memodifikasi data agar data lebih mudah dilihat, dipahami, dan dibuat modelnya.
- Yang biasanya ditangani dalam proses ini adalah
  - Missing values / incomplete record
  - Pencilan (Outliers)
  - Data duplikat
  - Data yang tidak relevan / tidak akurat
  - Bad (dirty) data

Workflow

- Inspection
- Cleaning
- Verifying
- Reporting Changes


DATA CLEANING CYCLE
Importing data
Merging data sets
Rebuilding missing data
Standardization
Normalization
De-duplication
Verification & enrichment
Exporting data

Dataset

- Dataset yang akan kita gunakan untuk proses cleaning ini adalah :
  - Data dummy “dirtydata.csv”
  - The Metropolitan Museum of Art Open Access https://github.com/metmuseum/openaccess/

Data yang Bagus

- Beberapa kriteria untuk menentukan apakah datanya bagus atau tidak diantaranya adalah :
  - **Validitas** : seberapa sesuai data dengan aturan bisnis yang ada atau batasannya. Bahwa data tersebut benar dan berguna.
  - **Akurasi** : sejauh mana data yang ada dapat mengungkapkan kejadian sesungguhnya
  - **Kelengkapan** : seberapa komprehensif data dan ukurannya diketahui
  - **Konsistensi** : tingkat konsistensi data, baik dalam satu dataset atau lintas dataset
  - **Keseragaman** : kesamaan cara mengukur dan satuan yang digunakan
- Istilah **Integritas Data** meliputi validitas, akurasi, dan kelengkapan

Data yang Bagus

- Data akan lebih bagus dan kualitasnya lebih tinggi jika diperbaiki dari sumbernya
- Setelah diperbaiki dari sumbernya, kita dapat melakukan
  - **Standardisasi**, untuk memastikan tipe data di tiap kolom seragam
  - **Normalisasi**, untuk memastikan konsistensi semua data yang ada
  - **Merging datasets**, jika data berada di beberapa dataset yang terpisah
  - **Agregasi**, untuk mengurutkan dan mengelompokkan data dengan aturan tertentu
  - **Filtering**, untuk membatasi dataset ke informasi yang dibutuhkan saja

Inspeksi Data

- Preview dataframenya menggunakan
  - df.head(),
  - df.tail(), atau
  - df.sample()
- Periksa bentuk dimensi dataframenya
  - df.shape #note : (banyak baris, banyak kolom)
- Periksa daftar kolom dan tipe datanya
  - list(df.columns.values)
  - df.dtypes
- Melihat banyaknya isi tiap kolom
  - df.count() #note : missing value tidak dihitung

Inspeksi Data

- Melihat informasi dataframe lebih menyeluruh
  - df.info()
  - df.describe() #info : informasi statistik deskriptif dari data pada kolom bertipe numerik
- Bisa menggunakan modul untuk profiling data

  $ pip install pandas-profiling

```
import pandas_profiling
profile = df.profile_report(title="Profiling Reports")
profile.to_file(output_file="Report.html")
```

Profiling Report

Overview

Dataset info

Number of variables	9
Number of observations	312
Missing cells	0 (0.0%)
Duplicate rows	0 (0.0%)
Total size in memory	22.0 KiB
Average record size in memory	72.3 B

Variables types

Numeric	2
Categorical	5
Boolean	1
Date	0
URL	0
Text (Unique)	1
Rejected	0
Unsupported	0

Warnings

`Amount_Drink` has 32 (10.3%) zeros Zeros

Proses Cleaning

- Buka file dirtydata.csv ke dalam dataframe df
- Inspeksi data tersebut
- Daftarkan / tulis kesalahan apa saja yang perlu diperbaiki pada data tersebut

Pastikan Tipe Data Sesuai

Pandas dtype	Python type	NumPy type	Usage
object	str	string_, unicode_	Text
int64	int	int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64	Integer numbers
float64	float	float_, float16, float32, float64	Floating point numbers
bool	bool	bool_	True/False values
datetime64	NA	NA	Date and time values
timedelta[ns]	NA	NA	Differences between two datetimes
category	NA	NA	Finite list of text values

- Id seharusnya berupa string (object), dan unique
- Tanggal lahir seharusnya berupa tanggal (date)
- Tempat lahir, kota tempat tinggal, dan jenis kelamin seharusnya kategorial (category)
- No HP seharusnya integer / string, bukan float
- Berat seharusnya integer / float

Transformasi Tipe Data

pd.to_numeric

- Mengubah nilai suatu kolom ke dalam format numerik
- Ditambahkan argumen errors untuk menangani apabila terjadi error, bisa berupa :
  - Ignore : tidak jadi ditranformasi
  - Coerce : dipaksa untuk jadi numerik. Jika tidak bisa, akan dianggap NaN (not a number)

```
df['Berat'].apply(pd.to_numeric, errors = 'coerce')
```

Transformasi Tipe Data

- Untuk menghilangkan elemen yang bukan numerik dan mengambil nilai numeriknya saja, bisa digunakan regex

```
df['Berat'].str.extract('(\d+)')
```

```
df['Berat'].str.replace('(\D+)','')
```

https://docs.python.org/3/howto/regex.html

Transformasi Tipe Data

pd.to_datetime

- Mengubah nilai suatu kolom ke dalam format tanggal

```
pd.to_datetime(df['Tanggal Lahir'])
```

Transformasi Tipe Data

df.astype()

- Mengubah kolom pandas ke dalam tipe data yang ditentukan : int, str, float, ataupun category

```
df.astype({'ID':int,'Tempat Lahir':'category','No HP':str})
```

Lihat Nilai Unik Tiap Kolom

```
df['Jenis Kelamin'].unique()
```

```
array(['Pria', 'L', nan, 'Wanita', 'Laki-Laki', 'P', 'Perempuan'],
      dtype=object)
```

```
df['ID'].duplicated()
```

```
4     False
5     False
6      True
7     False
8     False
```

```
df[df.duplicated(['ID','Nama Pegawai'],keep=False)]
```

Kasus : Penulisan Tidak Seragam

- Jenis kelamin diinputkan sebagai Pria, Laki-laki, Lelaki, L, Wanita, Perempuan, dll.
- Padahal secara kategori hanya ada 2 : Laki-laki (L) dan Perempuan (P)
- Solusinya dilakukan mapping

```
#mapping
df['Jenis Kelamin'].map({'L':'L', 'Laki-Laki': 'L', 'Pria': 'L',
                         'P':'P','Wanita':'P', 'Perempuan':'P'})
```

Kasus : Duplikat Data

```
df[df['ID']==df['ID'][6]]
```

	ID	Nama Pegawai	Tempat Lahir	Tanggal Lahir	Jenis Kelamin	Kota Tempat Tinggal	No HP	Jam Kerja	Berat
2	10010107	Misna Oktina	Jakarta	14-04-1971	NaN	NaN	NaN	NaN	74
6	10010107	Misna Oktina	Jakarta	14-04-1971	Wanita	Depok	NaN	8:00	97

- Bisa didrop

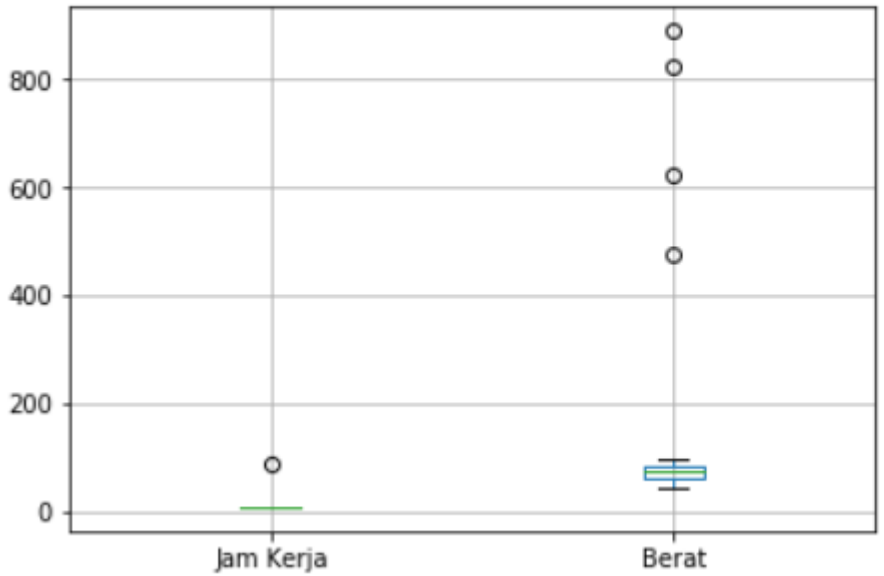
```
df.drop(2)
```

Kasus : Pencilan

- Gunakan boxplot untuk melihat sebaran data pada kolom bertipe numerik

```
df.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x6877095390>
```



Kasus : Missing value

- Drop
- Replace dengan nilai tertentu
  - Mean
  - Mode
  - categorial