

Applied Data Analytics

Pandas basics

Reductions with DataFrames

Hans-Martin von Gaudecker and Aapo Stenhammar

Example

country	year	pop	gdpPercap
Egypt	1977	38.8	2785.5
Egypt	2002	73.3	4754.6
South Africa	1977	27.1	8028.7
South Africa	2002	44.4	7710.9

Reductions

- `df.mean()` , `df.median()` exist
- Series: Reduce 1-dimensional data to a scalar value
- DataFrames: Reduce 2-dimensional data to a Series
- By default, reductions are column-wise
- Reason:
 - Columns should always be homogenous
 - Rows not necessarily

First try

```
[1] df.mean()
```

```
[1] -----
TypeError                                Traceback (most recent call last)
Cell In[1], line 1
----> 1 df.mean()
```

```
[80 lines clipped]
```

```
File pandas/core/nanops.py:1686, in _ensure_numeric(x)
    1683 inferred = lib.infer_dtype(x)
    1684 if inferred in ["string", "mixed"]:
    1685     # GH#44008, GH#36703 avoid casting e.g. strings to numeric
-> 1686     raise TypeError(f"Could not convert {x} to numeric")
    1687 try:
    1688     x = x.astype(np.complex128)
```

```
TypeError: Could not convert ['EgyptEgyptSouth AfricaSouth Africa'] to numeric
```

Solution

Stick to numeric columns

```
[2] df[["year", "pop", "gdpPercap"]].mean()
```

```
[2] year      1989.500  
    pop       45.900  
    gdpPercap 5819.925  
    dtype: float64
```

Better solution

Stick to numeric columns that make sense

```
[3] df[["year", "pop"]].mean()
```

```
[3] year    1989.5  
    pop      45.9  
    dtype: float64
```