Cheat Sheet

Pandas

Data manipulation and analysis essentials

Kostya Numan
AI & AGI Researcher

Pandas is a Python library for easy data manipulation and analysis using DataFrames and Series.

Kostya Numan

AI & AGI Researcher

Working With Files

Reading CSV

```
df = pd.read_csv("file.csv")
```

Reading Excel

```
df = pd.read_excel("file.xlsx")
```

Writing CSV

```
df.to_csv("file.csv")
```

Writing Excel

```
df.to_excel("file.csv")
```

Kostya Numan

AI & AGI Researcher

Cleaning Columns

Dropping columns

```
df = df.drop(
  ["age", "gender"], axis=1
)
```

Changing column type

```
df["age"] = df["age"].astype(int)
df["x"] = df["x"].astype(float)
```

Renaming columns

```
df = df.rename(
  columns={"old": "new"},
)
```

Kostya Numan

AI & AGI Researcher

Missing Values

Dropping NaNs in all columns

```
df = df.dropna()
```

Dropping based on a single column

```
mask = ~df["age"].isna()
df = df[mask]
```

Filling missing values with a constant

```
df["age"] = df["age"].fillna(0)
```

Filling missing values with the last present value

```
df["price"] = df["price"].ffill()
```

Kostya Numan

AI & AGI Researcher

Aggregating

Aggregating by a column

```
df.groupby("profit").sum()
df.groupby("age").mean()
```

Groupping each column separately

```
df.groupby("column_name").agg({
  "age": "min",
  "year": "max",
  "score": "mean"
})
```

Kostya Numan

AI & AGI Researcher

Time-series Aggregation

Resampling in 1 day intervals

```
df.resample("1D").last()
df.resample("1D").mean()
df.resample("1D").min()
```

Rolling 1-day aggregation

```
df.rolling("1D").mean()
df.rolling("1D").min()
```

Kostya Numan

AI & AGI Researcher

Sorting

Sorting by a single column

```
df.sort_values(
   by="column_name",
   ascending=True)
```

Sorting by multiple columns

```
df.sort_values(
   by=["column_a", "column_b"],
   ascending=[True, False])
```

Sorting by index

```
df.sort_index(ascending=True)
```

Kostya Numan

AI & AGI Researcher

Transforming Columns

Absolute value

```
df["change"] = df["change"].abs()
```

String manipulations

```
df["x"] = df["x"].str.lower()
```

Applying a function

```
df["x"] = df["x"].apply(my_func)
```

Kostya Numan

AI & AGI Researcher

Selecting & Indexing

Selecting a subset of columns

```
df["name", "age"]
```

Selecting every nth row

```
df.iloc[::3]
```

Boolean Indexing

```
df[df["name"] == "Mike"]
```

Kostya Numan

AI & AGI Researcher

Subscribe

to Numan Substack
for more insights

Kostya Numan
AI & AGI Researcher