Cheat Sheet – Deskriptive Statistik mit Python

Für Statistik I – Technische Hochschule Ingolstadt

0) Setup & Imports

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

Tipps:

- `pd.set_option("display.max_columns", None)`
- `np.random.seed(42)` für Reproduzierbarkeit

1) Daten einlesen & speichern

```python
df = pd.read_csv("file.csv")
# df = pd.read_excel("file.xlsx")
# df.to_csv("output.csv", index=False)
```

2) Überblick verschaffen

```python
df.head()
df.info()
df.describe(include="all")
df.shape
df.columns
```

3) Auswahl, Filtern, Sortieren

```python
df.loc[df["Age"] > 18, ["Sex","Fare"]]
df.query("Age > 18 and Sex == 'female'")
df.sort_values(["Sex","Age"], ascending=[True,False])
df.drop(columns=["Ticket"])
df.rename(columns={"old":"new"})
df.assign(BMI = df["Weight"] / (df["Height"]**2))
```

4) Datentypen, Strings, Kategorien

```python
df["Age"] = pd.to_numeric(df["Age"], errors="coerce")
df["Date"] = pd.to_datetime(df["Date"])
df["Name"] = df["Name"].str.strip().str.lower()
df["CityFlag"] = df["City"].str.contains("Ingolstadt", case=False, na=False)
df["Class"] = pd.Categorical(df["Class"], ordered=True,
                             categories=["low","mid","high"])
```

5) Fehlende Werte & Duplikate

```
df.isna().sum()
df.dropna(subset=["Age"])
df.fillna({"Age": df["Age"].median()})
df.duplicated().sum()
df.drop_duplicates()
```

6) Grundlegende Kennzahlen

```
df["x"].mean()
df["x"].median()
df["x"].std()
df["x"].quantile([0.25,0.5,0.75])
# Gruppiert:
df.groupby("Group")["x"].agg(["count","mean","median","std","min","max"])
```

7) Häufigkeiten & Kreuztabellen

```
df["cat"].value_counts(normalize=True).mul(100).round(1)
pd.crosstab(df["Sex"], df["Survived"], normalize="index", margins=True)
pd.crosstab(df["Class"], df["Sex"], normalize="columns", margins=True)
```

8) Binning (Klassen bilden)

```
pd.cut(df["x"], bins=[0,10,20,30], labels=["0-10","10-20","20-30"])
pd.qcut(df["x"], q=4)  # Quartile
```

9) Ausreißer-Checks

```
Q1, Q3 = df["x"].quantile([0.25,0.75])
IQR = Q3 - Q1
lo, hi = Q1 - 1.5*IQR, Q3 + 1.5*IQR
mask_out = ~df["x"].between(lo, hi)

z = stats.zscore(df["x"], nan_policy="omit")
out_z = np.abs(z) > 3
```

10) Gewichtete Kennzahlen

```
wmean = np.average(df["x"], weights=df["w"])
# from statsmodels.stats.weightstats import DescrStatsW
# d = DescrStatsW(df["x"], weights=df["w"])
# d.mean, d.std
```

11) Visualisierung – Basics

```
plt.hist(df["x"].dropna()); plt.show()
plt.boxplot(df["x"].dropna()); plt.show()
plt.scatter(df["a"], df["b"]); plt.show()
```

12) Visualisierung – Seaborn

```
sns.histplot(data=df, x="x", bins=30)
sns.boxplot(data=df, x="cat", y="x")
sns.countplot(data=df, x="cat")
sns.scatterplot(data=df, x="a", y="b", hue="Group")
sns.heatmap(df.corr(numeric_only=True), annot=False)
```

13) Gruppierung + Visualisierung

```
g = df.groupby("cat")["x"].mean().reset_index()
plt.bar(g["cat"], g["x"]); plt.show()

shares = df["cat"].value_counts(normalize=True)
shares.plot(kind="bar"); plt.show()
```

14) Titanic-Beispiele

```
pd.crosstab(df["Pclass"], df["Survived"], normalize="index")
pd.crosstab(df["Sex"], df["Survived"], normalize="index")
df.groupby("Pclass")["Age"].agg(["count","mean","median","std"])
sns.histplot(data=df, x="Fare", bins=40)
```

15) Export von Ergebnissen

```
summary = df.groupby("Group")["x"].agg(["count","mean","median"]).
    reset_index()
summary.to_csv("summary.csv", index=False)
plt.savefig("boxplot.png", dpi=150)
```

16) Häufige Stolperfallen

- Zahlen als Strings → `to_numeric()` verwenden
- `describe()` ignoriert NAs
- Immer `reset_index()` nach `groupby()` bei Bedarf
- Bei Visualisierungen Achsen prüfen und beschriften

17) Mini-Workflow

1. Laden (`read_csv`, `info`)
2. Typen & NAs bereinigen
3. Kennzahlen, Häufigkeiten, Kreuztabellen
4. Visualisierung (Histogramm, Box, Count, Scatter)
5. Ergebnisse exportieren und dokumentieren