Übung 03
Import und erste Auswertung eines Datensatzes mit **Pandas**
INFI-IS
5xHWII

Albert Greinöcker

October 9, 2024

Ziel der Übung:

- Aufbereiten und importieren eines Datensatzes
- Erste Analysen mit `Pandas` und `Matplotlib`

1 Importieren Aufbereiten des Datensatzes `Wintertourismus`

1.1 Beschreibung des Datensatzes

Es sind die Anzahl der Nächtigungen in den Wintersaisonen von 2000 bis 2023 - aufgegliedert nach Gemeinden - tabellarisch dargestellt.

Hinweis: Eine gute Unterstützung bekomment man generell vom Beispiel `ex\_05\_pandas\_descriptive\_statistics`

1.2 Aufbereiten der Daten

Die entsprechende Datei, die Basis für die Analyse sein soll, wurde von `https://www.tirol.gv.at/statistik-budget/statistik/tourismus/#c76985` bezogen (liegt auch bei der Aufgabenstellung in Moodle). Leider ist sie in dieser Form nicht für die Datenanalyse geeignet. Bitte diesen entsprechend aufbereiten (im Allgemeinen müsste es reichen, die oberen Beschreibungszeilen zu löschen).

1.3 Import und Kontrolle

- Es soll der Datensatz *Zeitreihe Wintertourismus 2000 2022* aus Moodle in Python importiert werden. In diesem Fall ist die Beschriftung der Spalten schon vorhanden. Bitte die Namen so ändern dass die Jahre Zeichenketten und keine Zahlen sind. Die Jahreszahlen sollen mit einem 'x' beginnen, damit man besser damit arbeiten kann.
- Wenn man als ersten Schritt die importierten Daten grundsätzlich kontrollieren möchte, bietet sich der Befehl (*describe*) an. Was ist aus diesem ersichtlich bzw. warum ist er für die Kontrolle wichtig?

Hinweis:

Für eine bessere Ausgabe der Daten gibt es das Python-Paket `tabulate`, dass man einfach mit `pip` installieren kann. Verwendung: `print(tabulate(df, headers=df.columns))`

2 Erste Auswertung

2.1 Wachstum darstellen

Hole die Zahlenwerte zu den einzelnen Jahren in Innsbruck und stelle den zeitlichen Verlauf als Punktdiagramm dar. Als Zeichen soll ein `i` Verwendet werden.

Hinweis:

- Aus einem Dataframe bekommt man mit dem Attibut `values` die Werte als numpy-Array, z.B. `df.values[0, 3:]` Zeile 0, Alle Spalten ab 3 (wir wollen nur die Zahlen für die Jahre)
- Vorsicht: Auch wenn die Werte von Innsbruck nur aus einer Zeile bestehen, sind die Ergebnisse trotzdem 2-Dimensional.

2.2 Wachstum des eigenen Bezirks (Falls Innsbruck, bitte einen anderen auswählen)

- Dazu ist zuerst die Auswahl der Gemeinden im Bezirk notwendig
- Diese Werte müssen aufsummiert werden

  Hinweis: Verwende den Befehl `sum` - Spaltenweises Summieren: `sum(axis=0)`, Zeilenweises Summieren: `sum(axis=1)`
- Bitte wieder einerseits die Zahlenwerte in der Konsole als auch ein Liniendiagramm ausgeben

3 Berechnen von Werten

3.1 Min, Max, Range, Avg

Zu den einzelnen Gemeinden sollen das Minimum, Maximum, Range (also Maximum - Minimum) und der Durchschnitt berechnet werden. Diese sollen in einer eigenen Spalten-Variable abgelegt werden. Die Befehle sind einfach `max` und `min`.

Hinweis:

- Die einzelnen Funktionen wie `min` bietet auch wieder den Parameter `axis` an.
- Die Zuweisung in eine neue Spalte geht mit eckigen Klammern:

```
w['min'] = ...Hier kommt dann der Befehl fuer die Berechnung hin
```

- der Durchschnitt berechnet sich mit `mean`.

3.1.1 Standardisierung

Der Range ist jetzt absolut. Wie könnte man diesen standardisieren, damit er vergleichbar ist?

3.2 Gesamtzahl an Touristen

- Die Gesamtzahl an Touristen pro Jahr soll berechnet werden (Befehl: `sum` mit `axis` = 0)
- Diese Werte sollen weiterverarbeitet werden, so dass man die Gesamtzahl über alle Jahre bekommt (also 1 Wert).

- Wie kann die Zusammenfassung nach Bezirken gemacht werden? (Befehl: `sum`)

  Pandas hat übrigens einen eigenen, vereinfachten Plot-Mechanismus eingebaut, der auf Matplotlib aufbaut:

```
sum_bez.plot.bar() # sum_bez ist das Ergebnis der Zusammenfassung mit \texttt{groupby}
plt.show()
```

4 Gegenüberstellung von Bezirken

4.1 Boxplots

Stelle die (standardisierten) Ranges der einzelnen Bezirke als Boxplot gegenüber. Jeder Bezirk soll eine eigene Farbe haben.
Hinweis: Es gibt mehrere unterschiedliche Möglichkeiten, diesen Plot zu machen:

a. Am Einfachsten ist es mit der eingebauten boxplot-Methode von Pandas:

```
w.boxplot(column="spaltenname", by="Bezirk")
plt.show()
```

b. Gruppierung ist in Matplotlib bei Boxplots nicht vorgesehen, deshalb muss man die Boxplots in einer Schleife erstellen, was nicht so toll ist:

```
pos = 0
labels =w['Bezirk'].unique()
for b in labels:
   bez = w[w.Bezirk == b]
   plt.boxplot(bez['range_std'], positions=[pos])
   pos += 1
plt.xticks(range(len(labels)), labels)
plt.show()
```

c. Was immer eine gute Alternative für die grafische Darstellung ist, ist `seaborn` (mit `pip` zu installieren)

```
import seaborn as sns
sns.boxplot(x=w['Bezirk'], y=w['spaltenname'], data=w)
plt.show()
```

4.2 Barplot

Stelle die die Jahreswerte für Innsbruck als `barplot` dar.

Hinweis:

Hier lohnt es sich auch wieder einen Blick auch die Seaborn-Variante zu werfen:

```
sns.barplot(x = labels, y= values, palette='terrain')
plt.xticks(rotation=70)
plt.show()
```

5 Gegenüberstellung mit den Einwohnerzahlen

Hier soll noch der Datensatz über die Einwohnerzahlen (ist im `git`-Projekt unter `bev_meld.xls` zu finden), den wir zur gemeinsamen Besprechung verwendet haben, miteinbezogen werden.

Man kann relativ einfach die beiden Datensätze "joinen":

```
both = pd.merge(df1, df2, how='inner', on = 'Gemnr')
```

Um ein wenig aufzuräumen bitte die Spalten wieder umbenennen und die Spalten, die doppelt sind, löschen, das geht so:

```
df = df.drop(columns='Gemnr')
```

Bitte mit diesem neu erzeugten Datensatz folgende Fragen beantworten:

a. Standardisiere die Anzahl Nächtigungen im Jahr 2018 mit der Bevölkerung pro Gemeinde . Es soll also berechnet werden, auf wie viele Einwohner kommen wie viele Touristen

b. Stelle diese Zahl als Boxplot, gruppiert nach Bezirk, dar und versuche, dieses Ergebnis zu interpretieren.

c. Interessant wären jetzt noch die 10 Gemeinden, wo diese Verhältniszahl am größten ist und die, wo es am kleinsten ist. Dazu gibt es folgende Möglichkeit:

```
df_high = df.sort_values('j2000', ascending=False)
```

d. Wie sieht das Verhältnis in der Heimatgemeinde aus?

6 Hinweise

- Bei den Gemeindenamen sind Leerzeichen am Ende enthalten. Diese bekommt man mit dem Befehl strip() weg, also z.B. `df['Gemeindename'] = df['Gemeindename'].str.strip()`