

# Übung 04

## Regression in **Pandas**

### INFI-IS

### 5xHWII

Albert Greinöcker

November 13, 2024



CCA - COMPETENCE CENTRE

**HTL Anichstraße**

## 1 Bevölkerungsentwicklung Tirol Gesamt

Die Aufgaben dieser Übung beziehen sich auf den Datensatz zur Bevölkerungsentwicklung aus Moodle bzw aus der vorigen Übung.

Ein Modell, dass die Bevölkerungsentwicklung Tirols voraussagt, soll erstellt werden. Wir haben die Regression allgemein besprochen und für die Umsetzung gibt es ein Beispiel dazu: `ex_06_pandas_regression.py` in den Pandas Examples.

Zu diesem Zweck müssen noch eine Bibliothek installiert werden: `statsmodels`

## 2 Erste Auswertung

### 2.1 Grafische Veranschaulichung

Eine grafische Veranschaulichung, welches die einzelnen Werte für die Gesamtbevölkerung beinhaltet (also eine Aufsummierung aller Bewohner pro Jahresspalte), soll erstellt werden. Die Regressionsgerade soll später über diese Darstellung gelegt werden.

### 2.2 Prognose

Berechne die prognostizierte Tiroler Gesamtbevölkerung, einmal direkt mit den Koeffizienten (nur für das Jahr 2030) und mit dem Befehl `predict` in den Jahren 2030-2100.

Hinweis:

- Sinnvoll wäre hier die Erstellung einer eigenen Methode, in der die Zahlenwerte als auch die Regressionsmethode daürbergelegt wird, dann kann man in der nächsten Aufgabe davon profitieren.
- Wenn man die Modellbildung wie im erwähnten Beispiel durchführt bekommt man mit `model.params` `a` und `b` für die Regressionsgerade  $y = ax + b$

### 3 Bevölkerungsentwicklung in der Wohngemeinde

Erstelle dasselbe für die Gemeinde, in der man lebt.

### 4 Gegenüberstellung von Bezirken

2 Bezirke (z.B. IL und RE) sollen bezüglich Bevölkerungswachstum gegenübergestellt werden. Bitte beides grafisch darstellen und die Steigungen beider Modelle in gesonderten Grafiken gegenüberstellen.

Hinweis: Möchte man Grafiken in Subplots darstellen sind nur wenige Änderungen notwendig (ein komplettes Demo ist im Regressionsbeispiel):

```
1 fig, axes = plt.subplots(nrows=2, ncols=2) # 2x2, also 4 subplots
2 t1.plot(ax=axes[0,0]); #mit ax gibt man an, wo der Plot hin soll
3 t4.plot(ax=axes[0,1]);
4 t9.plot(ax=axes[1,0]);
5 t12.plot(ax=axes[1,1]);
6
7 fig.tight_layout() # Dann überlagert sich die Ausgabe nicht
8 plt.show();
```

Dass die Grafiken auch wirklich übereinander gelegt werden und alles sichtbar ist, muss man die Zeichenbereiche (mit `plt.xlim([1980, 2030])` und `plt.ylim([10, 20])`)<sup>1</sup> festlegen.

### 5 Erkenntnisse

Wie gut eignen sich die erzeugten Regressionsgeraden für die Prognose?

Welche Erkenntnisse können nun aus den Ergebnissen über das Datenmaterial gewonnen werden? Bitte einfach zu den einzelnen Auswertungen schreiben.

---

<sup>1</sup>richtige Werte einsetzen