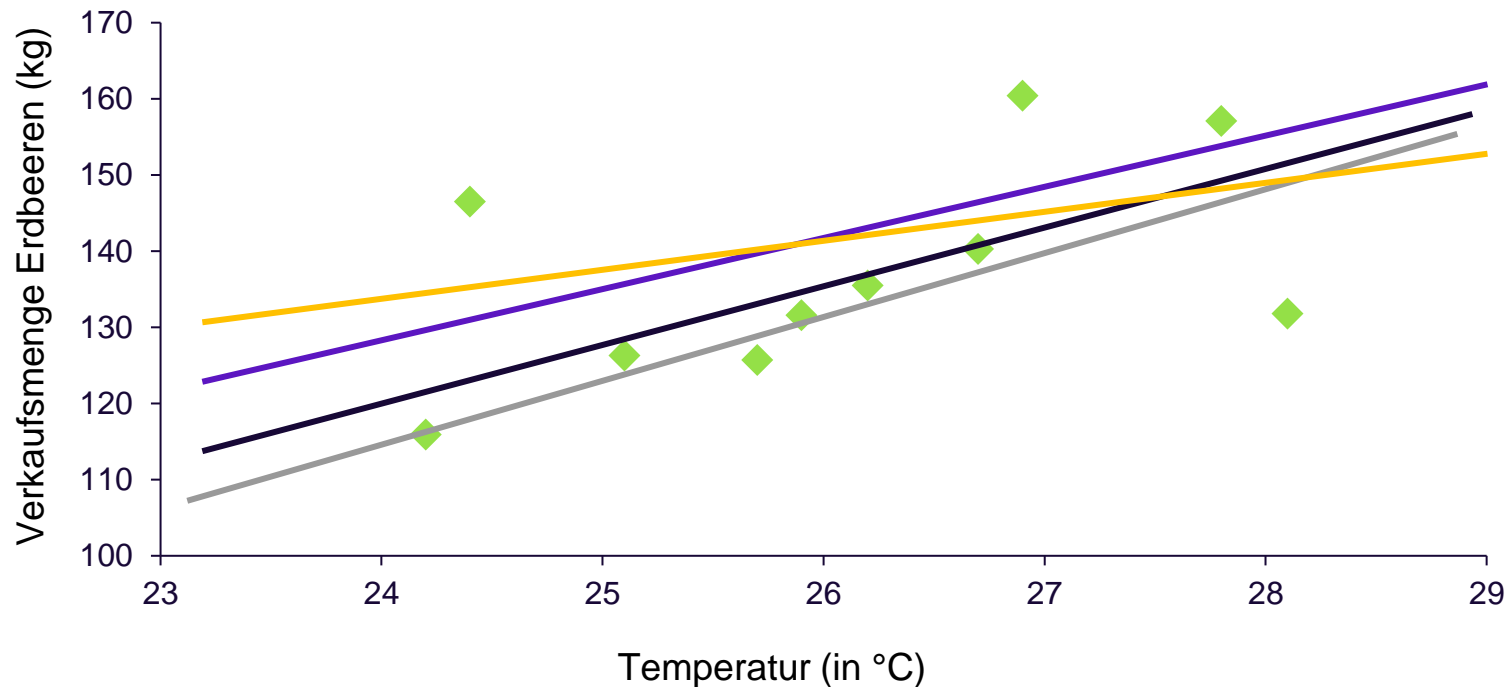


KI & Machine Learning

2

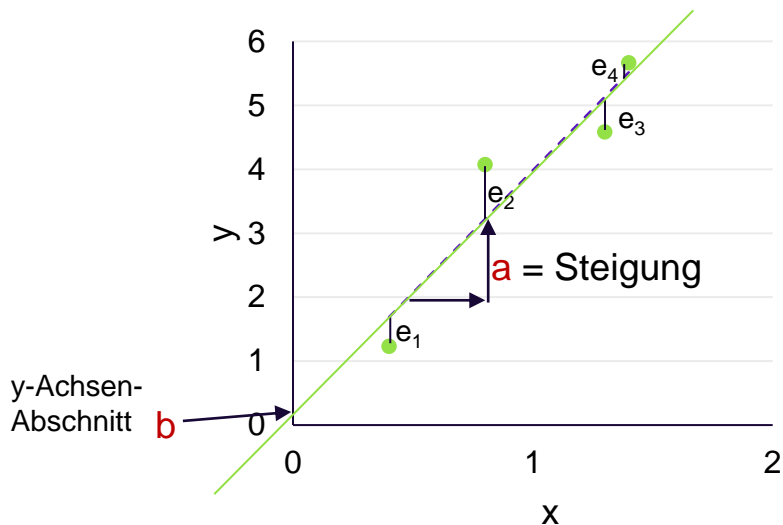
Linear Regression

Beschreibung eines Zusammenhangs durch eine Gerade (linearer Zusammenhang)



Lineare Regression = Methode der kleinsten Quadrate (OLS = ordinary least square)

Lege eine Gerade durch die Punkte, so dass die Summe der quadrierten Fehler möglichst klein ist



Geradengleichung: $y = a \cdot x + b$

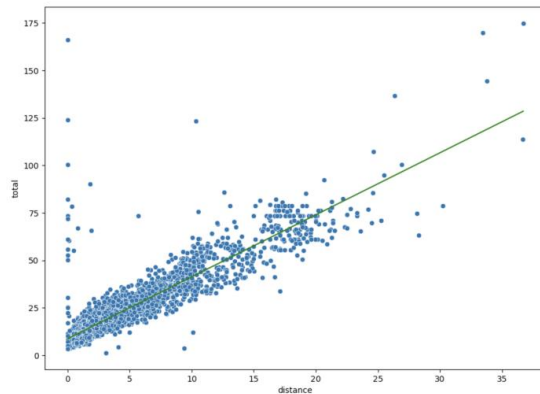
Finde **a** und **b**, so dass
 $e_1^2 + e_2^2 + e_3^2 + e_4^2$
minimal wird

In Python gibt es mehrere Möglichkeiten, eine lineare Regression durchzuführen. Die zwei gebräuchlichsten sind mittels **statsmodels** oder **scikit-learn**. Achtung, für statsmodel muss Konstante manuell ergänzt werden.

```
import statsmodels.api as sm
import seaborn as sns
```

```
model = sm.OLS(df["y"], df[["1", "x"]])
results = model.fit()
print(results.summary())
```

```
# Scatterplot mit Linie
ax = sns.scatterplot(data=df, x="x", y="y")
line_x = (df["x"].min(), taxis["x"].max())
a,b = results.params
line_y = (a + b*line_x[0], a + b*line_x[1])
sns.lineplot(x=line_x, y=line_y, ax=ax, color="green")
```

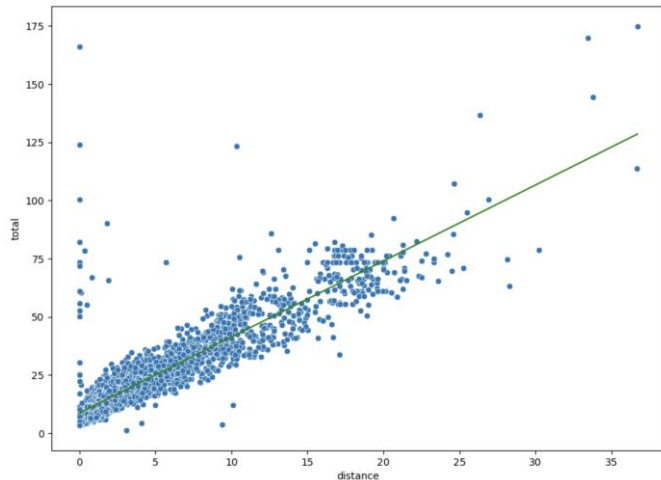


Mit scikit-learn sieht das so aus:

```
import sklearn.linear_model as lm

model = lm.LinearRegression()
model.fit(X=df[["x"]], y=taxis["y"])
print(model.coef_, model.intercept_)
# R2
print(model.score(X=df[["x"]], y=df["y"]))

# Scatterplot mit Linie
ax = sns.scatterplot(data=df, x="x", y="y")
line_x = (df["x"].min(), df["x"].max())
a, b = model.intercept_, model.coef_[0]
line_y = (a + b*line_x[0], a + b*line_x[1])
sns.lineplot(x=line_x, y=line_y, ax=ax, color="green")
```



scikit-learn stellt nicht alle Statistiken der Regression zur Verfügung. Dafür gibt es drei weitere Varianten, die im Machine Learning üblich sind und versuchen, Overfitting zu vermeiden. Dazu wird die Verlustfunktion (OLS), welche minimiert wird, durch Strafterme ergänzt

- **Ridge** (oder L2-Regulierung): $OLS + \alpha(b_1^2 + \dots + b_n^2)$
- **Lasso** (oder L1-Regulierung): $OLS + \alpha(|b_1| + \dots + |b_n|)$
- **ElasticNet**: $OLS + \alpha\lambda(b_1^2 + \dots + b_n^2) + \alpha(1 - \lambda)(|b_1| + \dots + |b_n|)$

α gibt die Stärke der Regulierung an.

Ridge Regression ist besser, wenn viele Prädiktoren signifikant und deren Koeffizienten ähnlich gross sind. Lasso Regression ist besser, wenn nur wenige Prädiktoren signifikant sind. ElasticNet kombiniert beide Ansätze