Saeugetiere_Stoffwechsel_Logarithmisch

April 5, 2023

1 Beispiel: Lineare Regression nach Transformation mit Logarithmus

Wir verwenden einen Datensatz von Säugetieren und betrachten den Stoffwechsel (BasalMetRate_mLO2hr) in Abhängigkeit von dem Gewicht eines Säugetieres (AdultBodyMass_g). Führen wir "blind" eine Lineaere Regression durch, so erhalten wir (immerhin) ein R^2 von ca. 0,6. Der Koeffizient für AdultBodyMass_g hat einen p-Wert < 0,05.

[1]:
```python
import pandas as pd
df = pd.read_csv("https://raw.githubusercontent.com/troescherw/datasets/master/
↪Saeugetiere.csv")
df
```

[1]:
```
      MSW05_Order      MSW05_Family    MSW05_Genus  MSW05_Species  \
0      Chiroptera       Mormoopidae     Pteronotus     quadridens
1      Chiroptera  Vespertilionidae         Myotis        velifer
2     Afrosoricida       Tenrecidae        Geogale         aurita
3      Chiroptera       Mormoopidae       Mormoops    blainvillei
4      Chiroptera        Natalidae        Natalus  tumidirostris
..            …                 …              …              …
567  Artiodactyla        Camelidae        Camelus    dromedarius
568     Carnivora          Ursidae          Ursus      maritimus
569     Carnivora         Phocidae     Cystophora       cristata
570  Artiodactyla         Cervidae      Capreolus      capreolus
571     Carnivora         Phocidae  Leptonychotes      weddellii

             MSW05_Binomial  AdultBodyMass_g  BasalMetRate_mLO2hr  Met-Gram  \
0      Pteronotus quadridens             5.64                 6.12  1.085106
1             Myotis velifer             9.82                 7.70  0.784114
2             Geogale aurita             6.69                 7.72  1.153961
3        Mormoops blainvillei             8.69                 7.99  0.919448
4      Natalus tumidirostris             6.30                 8.31  1.319048
..                         …                …                    …         …
567      Camelus dromedarius        492714.47             40293.00  0.081778
568         Ursus maritimus        371703.81             44346.00  0.119305
569      Cystophora cristata        278896.81             62991.00  0.225858
570      Capreolus capreolus         22502.01             78470.00  3.487244
```

```
571  Leptonychotes weddellii        400000.00            113712.00  0.284280

    MaxLongevity_m
0                *
1           135.96
2                *
3                *
4                *
..               …
567            480
568          458.4
569            420
570            204
571            300

[572 rows x 9 columns]
```

[2]:
```python
from statsmodels.formula.api import ols

model = ols("BasalMetRate_mLO2hr~AdultBodyMass_g", data=df).fit()
model.summary()
```

[2]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                             OLS Regression Results
==============================================================================
Dep. Variable:     BasalMetRate_mLO2hr   R-squared:                       0.573
Model:                             OLS   Adj. R-squared:                  0.572
Method:                  Least Squares   F-statistic:                     764.9
Date:                 Wed, 05 Apr 2023   Prob (F-statistic):          2.05e-107
Time:                         04:46:29   Log-Likelihood:                -5714.3
No. Observations:                  572   AIC:                         1.143e+04
Df Residuals:                      570   BIC:                         1.144e+04
Df Model:                            1
Covariance Type:             nonrobust
================================================================================
===
                      coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
Intercept         626.0711    225.473      2.777      0.006     183.212
1068.930
AdultBodyMass_g     0.1247      0.005     27.658      0.000       0.116
0.134
==============================================================================
Omnibus:                      811.543   Durbin-Watson:                   1.020
```

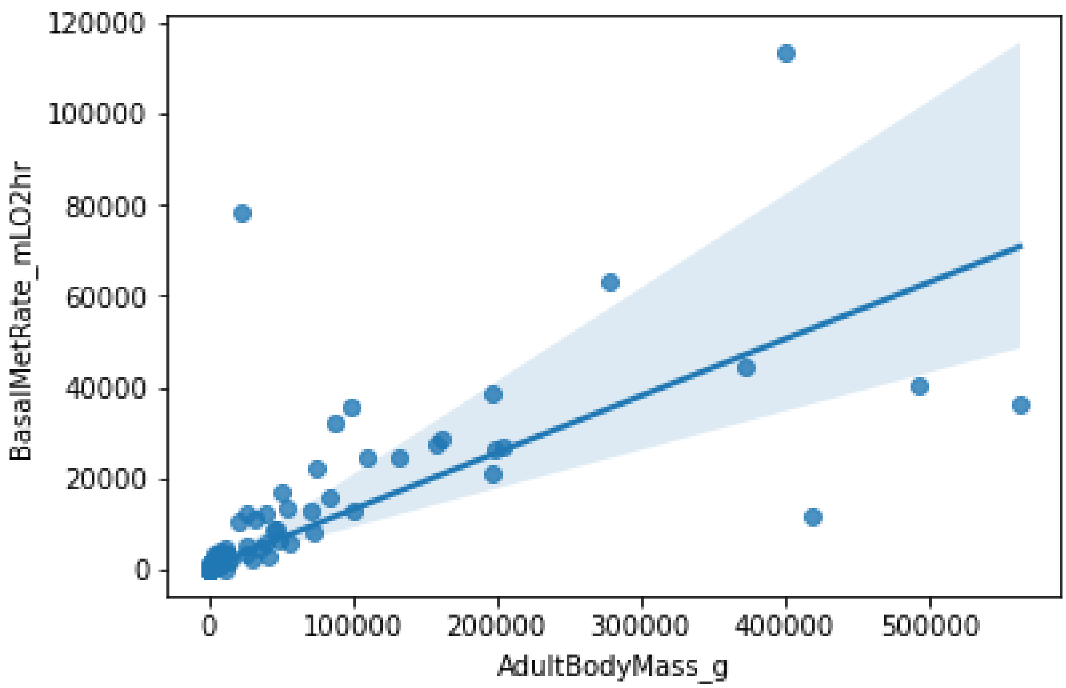
```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           332623.128
Skew:                           7.177   Prob(JB):                         0.00
Kurtosis:                     120.261   Cond. No.                     5.10e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.1e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

Ein Scatterplot zeigt allerdings, dass die Daten hier sehr streuen, und zwar zunehmend mit dem Körpergewicht (Heteroskedastizität).

[3]:
```python
import seaborn as sns
sns.regplot(x=df.AdultBodyMass_g, y=df.BasalMetRate_mLO2hr)
```

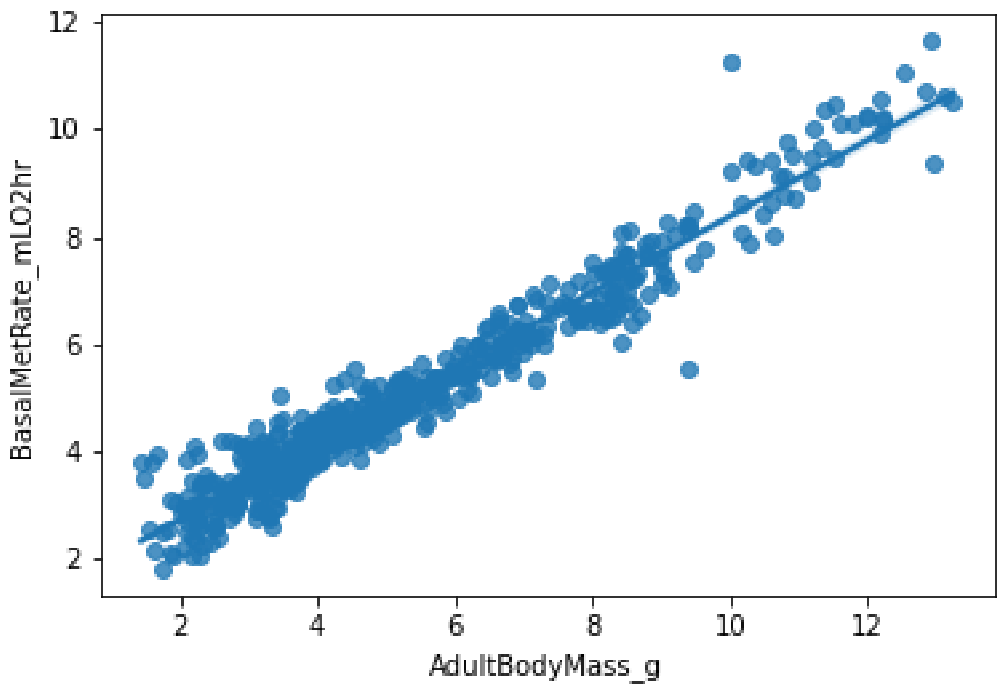
[3]: `<AxesSubplot:xlabel='AdultBodyMass_g', ylabel='BasalMetRate_mLO2hr'>`



Wir führen eine Log-Transformation durch: Von beiden Features berechnen wir den Logarithmus (zur Basis e). Wir sehen, dass dann ein sehr guter linearer Zusammenhang zwischen den transformierten Daten besteht!

[4]:
```
import numpy as np
sns.regplot(x=np.log(df.AdultBodyMass_g), y=np.log(df.BasalMetRate_mLO2hr))
```

[4]:
```
<AxesSubplot:xlabel='AdultBodyMass_g', ylabel='BasalMetRate_mLO2hr'>
```



Eine lineare Regression mit den transformiertn Daten ergibt dann sogar ein R^2 von ca. 0.94! Ein enorm hoher Wert wenn man bedenkt, dass es sich um die unterschiedlichsten Tiere handelt, deren Daten in den Datensatz eingeflossen sind.

[5]:
```
pd.options.mode.chained_assignment = None  # default='warn'
df["BasalMetRate_mLO2hr_log"] = np.log(df.BasalMetRate_mLO2hr)
df["AdultBodyMass_g_log"] = np.log(df.AdultBodyMass_g)
model = ols("BasalMetRate_mLO2hr_log~AdultBodyMass_g_log", data=df).fit()
model.summary()
```

[5]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                               OLS Regression Results
================================================================================
===
Dep. Variable:     BasalMetRate_mLO2hr_log   R-squared:
0.943
Model:                                 OLS   Adj. R-squared:
0.943
Method:                      Least Squares   F-statistic:
```

```
9440.
Date:                    Wed, 05 Apr 2023   Prob (F-statistic):
0.00
Time:                            04:46:31   Log-Likelihood:
-344.11
No. Observations:                     572   AIC:
692.2
Df Residuals:                         570   BIC:
700.9
Df Model:                               1
Covariance Type:                nonrobust
================================================================================
=======
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-------
Intercept               1.3259      0.043     30.876      0.000       1.242
1.410
AdultBodyMass_g_log     0.7063      0.007     97.158      0.000       0.692
0.721
==============================================================================
Omnibus:                       86.538   Durbin-Watson:                   1.558
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              501.744
Skew:                           0.498   Prob(JB):                    1.12e-109
Kurtosis:                       7.479   Cond. No.                         14.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

Wollen wir eine Prognose durchführen, müssen wir für die unabhängige variable natürlich auch vorher eine Transformation durchführen, ebenso mit dem Ergebnis der Vorhersage:

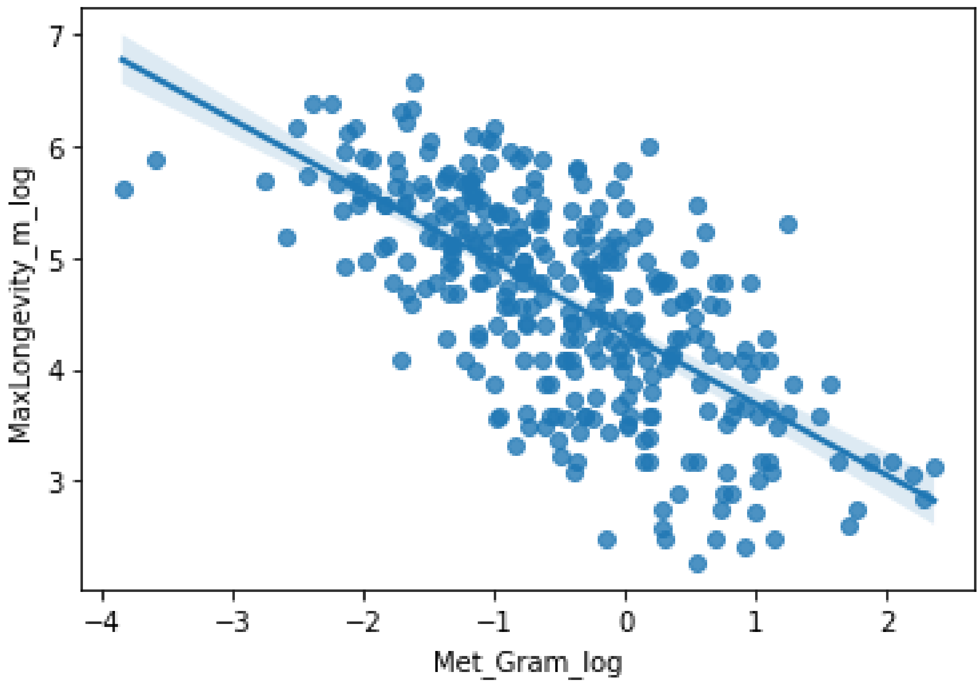
[6]:
```python
gewicht = 80000 # Gewicht in Gramm
gewicht_log = np.log(gewicht)
pred = model.predict(pd.DataFrame([{"AdultBodyMass_g_log":gewicht_log}]))
np.exp(pred)
```

[6]:
```
0    10936.404929
dtype: float64
```

Interessant ist auch die Lebenserwartung in Abhängigkeit vom Stoffwechsel (Met-Gram)! Tiere (und auch Menschen) mit einem geringeren Stochwechsel haben eine höhere Lebenserwartung. Auch hier führen wir eine Log-Transformation durch:

[7]:
```python
import numpy as np
df = df[df.MaxLongevity_m!="*"]
df["Met_Gram_log"] = np.log(df["Met-Gram"])
df["MaxLongevity_m_log"] = np.log(df.MaxLongevity_m.astype(float))
sns.regplot(x=df.Met_Gram_log, y=df.MaxLongevity_m_log)
```

[7]: <AxesSubplot:xlabel='Met_Gram_log', ylabel='MaxLongevity_m_log'>



[8]:
```python
model2 = ols("MaxLongevity_m_log~df.Met_Gram_log", data=df).fit()
model2.summary()
```

[8]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:     MaxLongevity_m_log   R-squared:                       0.458
Model:                            OLS   Adj. R-squared:                  0.456
Method:                 Least Squares   F-statistic:                     272.9
Date:                Wed, 05 Apr 2023   Prob (F-statistic):           7.36e-45
Time:                        04:46:32   Log-Likelihood:                -339.48
No. Observations:                 325   AIC:                             683.0
Df Residuals:                     323   BIC:                             690.5
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
```

```
===
                    coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
Intercept          4.3263      0.042    101.917      0.000       4.243
4.410
df.Met_Gram_log   -0.6383      0.039    -16.518      0.000      -0.714
-0.562
==============================================================================
Omnibus:                        4.655   Durbin-Watson:                   1.764
Prob(Omnibus):                  0.098   Jarque-Bera (JB):                4.665
Skew:                          -0.264   Prob(JB):                       0.0970
Kurtosis:                       2.743   Cond. No.                         1.61
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

Immerhin wird die Lebenserwartung zu ca. 46 Prozent vom Stochwechsel erklärt. Auf die Lebenswerwartung haben natürlich noch viele andere Faktoren einen Einfluss. Wir haben eine fallende Regressionsgerade. Dies spiegelt das vorherige Ergebnis wider: Mit zunehmendem Stochwechsel sinkt die Lebenserwartung.