Einführung in die mathematische Datenanalyse

Jan Heiland

FAU Erlangen-Nürnberg – Sommersemester 2022

Contents

Vorwort

Das ist ein Aufschrieb.

Korrekturen und Wünsche immer gerne als *issues* oder *pull requests* ans github-repo.

Chapter 1

Was ist Data Science?

Data Science umfasst unter anderem folgende Aufgaben:

1. Strukturieren/Aufbereiten (Umgehen mit falschen, korrumpierten, fehlenden, unformatierten Daten)
2. Data Exploration (Daten "verstehen")
3. Data Analysis (quantitative Analysen, Hypothesen aufstellen)
4. Data Visualization (Hypothesen graphisch kommunizieren)
5. Modelle erzeugen/validieren (Regeln/Muster erkennen, Vorhersagen treffen) – *das* ist Machine Learning aber es gibt auch viele andere Ansätze.
6. Daten Reduktion

1.1 Wie passiert die Datenanalyse?

Mit mathematischen Methoden aus den Bereichen der

- linearen Algebra (z.B. Matrizen, Basen, lineare Gleichungssysteme)
- Statistik (z.B. Mittelwerte, Korrelationen, Verteilungen)
- Analysis (Grenzwerte, Abschätzungen)
- …

Dabei hilft Software, z.B.,

- Excel
- **Python**
- Matlab
- R

bei der Berechnung, Automatisierung, Visualisierung.

Python ist ein de-facto Standard in Data Science und Machine Learning.

1.2 Was sind Daten?

Wie sehen Daten aus?

- Numerisch reell, z.B. Temperatur
- Numerisch diskret, z.B. Anzahl
- Ordinal: Element einer festen Menge mit expliziter Ordnung, z.B. {neuwertig, mit Gebrauchsspuren, defekt}
- Binär: Eine von zwei Möglichkeiten, z.B. Wahr/Falsch oder aktiv/inaktiv
- Kategoriell: Element einer festen Menge ohne klare Ordnung, z.B. {Säugetier, Vogel, Fisch}
- sonstige strukturierte Daten, z.B. geographische Daten, Graphen
- reiner Text, z.B. Freitext in Restaurantbewertung

Außerdem können wir noch allgemeine Eigenschaften (Qualitätsmerkmale) von Daten unterscheiden

- strukturiert
- lückenhaft
- fehlerbehaftet (*verrauscht*)
- interpretierbar
- geordnet (oder nicht zu ordnen)

1.3 Beispiele

1.3.1 Tabellendaten – Mietpreise

Hier wären die Aufgaben von Data Science:

- Daten "verstehen", Zusammenhänge zwischen Variablen aufdecken,
- visualisieren.
- Gegebenenfalls fehlende Einträge bei (z.B.) `kaltmiete` vorhersagen

Datenexploration und -analyse für einzelne Variablen 1/3 Wir betrachten eine *numerische* Variable in einem rechteckigen Datensatz, also eine *Spalte* (z.B. `kaltmiete`). Wir bezeichnen den i-ten Eintrag in dieser Spalte mit x_i, wobei $i = 1, \ldots, N$ (N Anzahl der Zeilen).

Folgende *Schätzer/Metriken* können dabei helfen, diese Spalte besser zu verstehen:

- Mittelwert $\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$

	bundesland	stadt	baujahr	etage	hat_kueche	kaltmiete	wohnflaeche	zimmer
0	Nordrhein_Westfalen	Duisburg	1973	1	0	640	105	3
1	Baden_Württemberg	Ulm	1936	1	0	302	43	2
2	Nordrhein_Westfalen	Wuppertal	1986	3	0	150	67	2
3	Rheinland_Pfalz	Rhein_Lahn_Kreis	1970	3	0	315	47	3
4	Sachsen	Chemnitz	1900	2	0	315	63	2
5	Rheinland_Pfalz	Mainz	1989	1	0	1200	96	4
6	Bayern	Aschaffenburg_Kreis	1965	2	0	722	85	3
7	Berlin	Berlin	1952	0	1	706	63	2
8	Sachsen	Mittelsachsen_Kreis	1996	3	1	220	29	1
9	Brandenburg	Potsdam	1985	2	1	400	34	1

Figure 1.1: Abbildung: Tabelle von Wohnungsangeboten

	bundesland	stadt	baujahr	etage	hat_kueche	kaltmiete	wohnflaeche	zimmer
0	Nordrhein_Westfalen	Duisburg	1973	1	0	640	105	3
1	Baden_Württemberg	Ulm	1936	1	0	302	43	2
2	Nordrhein_Westfalen	Wuppertal	1986	3	0	150	67	2
3	Rheinland_Pfalz	Rhein_Lahn_Kreis	1970	3	0	315	47	3
4	Sachsen	Chemnitz	1900	2	0	315	63	2
5	Rheinland_Pfalz	Mainz	1989	1	0	1200	96	4
6	Bayern	Aschaffenburg_Kreis	1965	2	0	722	85	3
7	Berlin	Berlin	1952	0	1	706	63	2
8	Sachsen	Mittelsachsen_Kreis	1996	3	1	220	29	1
9	Brandenburg	Potsdam	1985	2	1	400	34	1

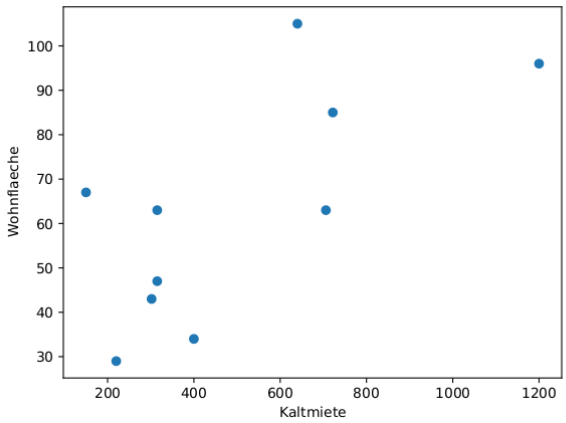
Figure 1.2: Abbildung: Eine Spalte der Tabelle

- gewichteter Mittelwert $\overline{x}_w = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{j=1}^{N} w_j}$, wobei w_i das Gewicht des i-ten Eintrages ist (z.B. eine andere Variable).
- Varianz: $s_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2$
- Standardabweichung $s = \sqrt{s_x^2}$.
- Median $= \frac{315+400}{2} = 357.5$.

Datenexploration und -analyse für mehrere Variablen Wir betrachten zwei Spalten $x = (x_1, \dots, x_N)$ und $y = (y_1, \dots, y_N)$.

Das Verteilung von zwei Variablen läßt sich im sogenannte **Scatter Plot** visualisieren.

che	kaltmiete	wohnflaeche	zin
0	640	105	
0	302	43	
0	150	67	
0	315	47	
0	315	63	
0	1200	96	
0	722	85	
1	706	63	
1	220	29	
1	400	34	



Datenexploration und -analyse für mehrere Variablen Wir betrachten zwei Spalten $x = (x_1, \dots, x_N)$ und $y = (y_1, \dots, y_N)$.

- Kovarianz $s_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})$
- Korrelation $\rho_{xy} = \frac{s_{xy}}{s_x \cdot s_y} \in [-1, 1]$.
- $\rho \approx 1$: Starke positive Korrelation, wenn x groß ist, ist y auch groß.
- $\rho \approx -1$: Starke negative Korrelation, wenn x groß ist, ist y klein
- $\rho \approx 0$: Wenig/keine Korrelation.

1.3.2 COVID-19 Daten

Vergleiche die Einführung in *Mathematik für Data Science 1* vom letzten Semester.

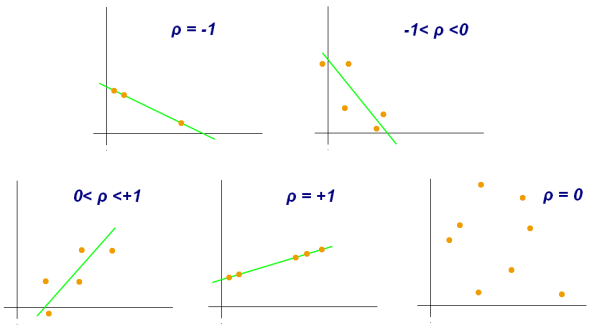


Figure 1.3: Von Kiatdd - Eigenes Werk, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=37108966

1.3.3 Netflix Prize

Hierbei geht es darum, ob aus bekannten Bewertungen von vielen verschiedenen Benutzern für viele verschiedene Filme abgeleitet werden kann, ob ein bestimmter Nutzer einen bestimmten Film mag (also positiv bewerten würde).

Vergleiche auch Wikipedia:Netflix_Prize

Das (Trainings-)Daten bestehen über `480189` Benutzer, die für `17770` Filme insgesamt `100480507` Bewertungen als ganze Zahlen zwischen `1` und `5` verteilten.

Ziel der Datenanalyse war es, für `2817131` "Paare" von Benutzern und Filmen, die Bewertung vorauszusagen. Neben der schieren Masse an Daten kamen noch Einschränkungen hinzu, die ein Mindestmaß an Qualität der Vorhersage sicherstellen sollten.

Das Problem ließe sich wie folgt darstellen.

Benutzer \ Film	`F1`	`F2`	...	`Fn`	...
`B1`	–	3	...	5	...
`B2`	3	4	...	2	...
`B3`	1	2	...	?	...
...	3	4	...	–	...

Gegeben viele (aber bei weitem nicht alle) Einträge in einer riesigen Tabelle. Können wir aus den Zusammenhängen bestimmte fehlende Einträge (z.B. wie findet Nutzer `B3` den Film `Fn`) herleiten?

Die besten Lösungen für dieses Problem basieren durchweg auf *Machine Learning* Ansätzen.

1.4 Python

Die Programmiersprache `python` wird uns durchs Semester begleiten. Einfach weil sie so wichtig ist für *Data Science* aber auch weil sie (meiner Meinung nach) einfach zu erlernen und zu benutzen ist.

1.5 Aufgaben

1.5.1 Python

Bringen sie ihr `python` zum Laufen, installieren sie `numpy`, `scipy` und

```
N = 20
xmax = 2
xmin = 0

xdata = np.linspace(xmin, xmax, N)
ydata = np.exp(xdata)

plt.figure(1)
plt.plot(xdata, ydata, '.')

plt.figure(2)
plt.semilogy(xdata, ydata, '.')
plt.show()
```

1.5.2 Einheitsmatrix

Schreiben sie ein script, dass die `5x5` Einheitsmatrix auf 3 verschiedene Arten erzeugt. (Eine Art könnte die eingebaute `numpy` Funktion `eye` sein).

```
import numpy as np

idfive = np.eye(5)
print(idfive)
```

Hinweis: schauen sie sich mal an wie `numpy`'s `arrays` funktionieren.

1.5.3 Matrizen Multiplikation und Potenz

Schreiben sie ein script, das die Übungsaufgabe aus der Vorlesung (potenzieren der Matrizen M_i, $i = 1, 2, 3, 4$) löst. Zum Beispiel mit

```
import numpy as np
mone = np.array([[0.9, 0.9], [0.9, 0.9]])

mone_ptwo = mone @ mone
print(mone_ptwo)

mone_pfour = mone_ptwo @ mone_ptwo
print(mone_pfour)
```

Oder so:

```
import numpy as np
mone = np.array([[0.9, 0.9], [0.9, 0.9]])
mone_p = np.eye(2)

for k in range(16):
```

```
    mone_p = mone_p @ mone
    if k == 1 or k == 3 or k == 15:
        print('k=', k+1)
        print(mone_p)
```

Achtung:

- bei Matrizen kann auch `*` benutzt werden – das ist aber nicht die richtige Matrizenmultiplikation (sondern die Multiplikation eintragsweise)
- Moegliche Realisierung der Matrizenmultiplikation
  - `np.dot(A, B)` – die klassische Methode
  - `A.dot(B)` – das selbe (manchmal besser, wenn `A` etwas allgemeiner ist (zum Beispiel eine `scipy.sparse` matrix)
  - `A @ B` – convenience Notation

Chapter 2

Lineare Regression

Ein wesentlicher Aspekt von *Data Science* ist die Analyse oder das Verstehen von Daten. Allgemein gesagt, es wird versucht, aus den Daten heraus Aussagen über Trends oder Eigenschaften des Phänomens zu treffen, mit welchem die Daten im Zusammenhang stehen.

Wir kommen nochmal auf das Beispiel aus der Einführungswoche zurück, werfen eine bereits geschärften Blick darauf und gehen das mit verbesserten mathematischen Methoden an.

Gegeben seien die Fallzahlen aus der CoVID Pandemie 2020 für Bayern für den Oktober 2020.

Table 2.1: Anzahl der SARS-CoV-2 Neuinfektionen in Bayern im Oktober 2020.

Tag	1	2	3	4	5	6	7	8	9	10	11
Fälle	352	347	308	151	360	498	664	686	740	418	320

Tag	12	13	14	15	16	17	18	19	20	21
Fälle	681	691	1154	1284	127	984	573	1078	1462	2239

Tag	22	23	24	25	26	27	28	29	30	31
Fälle	2236	2119	1663	1413	2283	2717	3113	2972	3136	2615

Wieder stellen wir uns die Frage ob wir **in den Daten einen funktionalen**

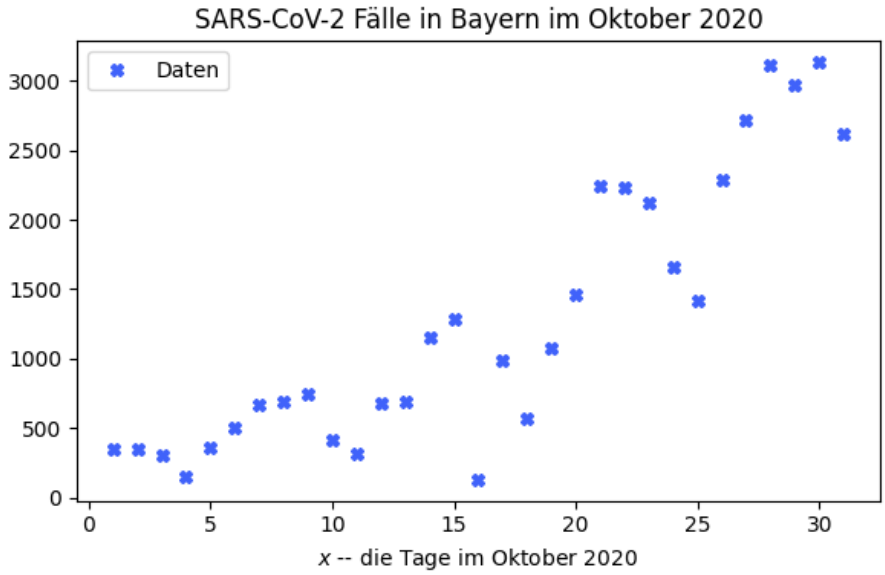


Figure 2.1: Fallzahlen von Sars-CoV-2 in Bayern im Oktober 2020

Zusammenhang feststellen können. Also ob wir die Datenpaare

(Tag x, Infektionen am Tag x)

die wir als

(x_i, y_i)

über eine Funktion f und die Paare

$(x, f(x))$

beschreiben (im Sinne von gut darstellen oder approximieren) können.

2.1 Rauschen und Fitting

Beim obigen Beispiel (und ganz generell bei Daten) ist davon auszugehen, dass die Daten **verrauscht** sind, also einem Trend folgen oder in einem funktionalen Zusammenhang stehen aber zufällige Abweichungen oder Fehler enthalten.

Unter diesem Gesichtspunkt ist eine Funktion, die

$f(x_i) = y_i$

nicht zielführend. (Wir wollen Trends und größere Zusammenhänge erkennen und nicht kleine Fehler nachzeichnen.) Das zu strenge Anpassen an möglicherweise verrauschte Daten wird **overfitting** genannt.

Vielmehr werden wir nach einer Funktion f suchen, die die Daten näherungsweise nachstellt:

$$f(x_i) \approx y_i$$

Hierbei passen jetzt auch Funktionen, die vielleicht einfach zu handhaben sind aber die Daten kaum noch repräsentieren. Jan spricht von **underfitting**.

Eine gute Approximation besteht im Kompromiss von *nah an den Daten* aber mit wenig *overfitting*.

2.2 Ansätze für lineare Regression

Um eine solche Funktion f zu finden, trifft Jan als erstes ein paar Modellannahmen. Modellannahmen legen fest, wie das f im Allgemeinen aussehen soll und versuchen dabei

1. die Bestimmung von f zu ermöglichen
2. zu garantieren, dass f auch die gewollten Aussagen liefert
3. und sicherzustellen, dass f zum Problem passt.

Jan bemerke, dass die ersten beiden Annahmen im Spannungsverhältnis zur dritten stehen.

Lineare Regression besteht darin, dass die Funktion als Linearkombination

$$f(x) = \sum_{i=1}^{N} w_i b_i(x)$$

von Basisfunktionen geschrieben wird und dann die *Koeffizienten* w_i so bestimmt werden, dass f die Daten bestmöglich annähert.

> Jan bemerke, dass *bestmöglich* wieder *overfitting* bedeuten kann aber auch, bei schlechter Wahl der Basis, wenig aussagekräftig sein kann. Der gute Kompromiss liegt also jetzt in der Wahl der passenden Basisfunktionen und deren Anzahl. (Mehr Basisfunktionen bedeutet möglicherweise bessere Approximation aber auch die Gefahr von *overfitting*.)

Typische Wahlen für die Basis $\{b_1, b_2, \dots, b_N\}$ sind

- Polynome: $\{1, x, x^2, \dots, x^{N-1}\}$ – für $N = 2$ ist der Ansatz *eine Gerade*
- Trigonometrische Funktionen: $\{1, \cos(x), \sin(x), \cos(2x), \sin(2x), \dots\}$
- Splines – Polynome, die abschnittsweise definiert werden
- Wavelets – Verallgemeinerungen von trigonometrischen Funktionen

2.3 Fehlerfunktional und Berechnung der Approximierenden

Wir setzen nun also an

$$f_w(x) = \sum_{i=1}^{N} w_i b_i(x)$$

und wollen damit $y_i \approx f_w(x_i)$ *bestmöglich* erreichen (indem wir die Koeffizienten $(w_1, \ldots, w_N)$ *optimal* wählen. Bestmöglich und optimal spezifizieren wir über den Mittelwert der Abweichungen in der Approximation über alle Datenpunkte

$$\frac{1}{N}$$

In unserem Falle, wollen wir annehmen, dass f eine lineare Funktion (vielleicht auch noch als *Gerade* bekannt)

$$f(x) = \theta_1 + \theta_2 x$$

ist. Damit können wir sagen, dass

1. f einigermassen einfach zu bestimmen ist (wir brauchen nur zwei Parameter θ_1 und θ_2)
2. wenn wir an Trends interessiert sind, sind lineare Funktionen eine gute Wahl
3. allerdings sind Prozesse meistens nichtlinear.

Vor allem haben wir sicher oft gehört, dass die Virusausbreitung gerne exponentiell verläuft, also zum Beispiel durch eine Funktion

$$g(x) = k_1 2^{k_2 x}$$

beschrieben werden sollte. Hier ist k_1 ein Skalierungsfaktor und k_2 ein Parameter, der die Wachstumsrate bestimmt.

In der Tat scheint die Annäherung der Datenpunkte aus Abbildung durch eine Gerade nicht zielführend.

> Allerdings sind exponentielle Zusammenhänge auf einer logarithmischen Skala linear.

Deshalb schauen wir uns die Daten über den Logarithmus (zur Basis 2) der Infektionszahlen an

> (Tag x, $\log_2$(Infektionen am Tag x))

Im Bild der Daten () kann Jan erkennen, dass eine Gerade vielleicht (abgesehen von einigen Werten) ganz gut reinpassen könnte.

2.4 Lineare Regression

Die Methode der *linearen Regression* besteht aus dem Anpassen einer linearen Funktion (einer *Geraden* im 2-dimensionalen Raum) an eine Datenwolke. Aus den unendlich vielen Möglichkeiten die Funktion zu definieren, wird die gewählt, die im Mittel den kleinsten Fehler zwischen den Datenpunkten und der Funktionsbeschreibung erzeugt.

Für unser Beispiel der CoVID-19 Zahlen, können wir *ad hoc* die Parameter θ_1 und θ_2 bestimmen. Wenn es soweit ist, werden wir die zugrundeliegenden mathematischen Konzepte und allgemein gültige Formeln zur Lösung kennenlernen.

Konkret gehen wir davon aus, dass wir N Datenpunkte

$$(x_j, h_j), \quad j = 1, 2, 3, \dots, N$$

haben wobei x_j die Variable ist und h_j das Merkmal dazu. In unserem Beispiel ist $N = 31$ (so viele Tage hat der Oktober 2020), $x_{10} = 10$ wäre der 10. Oktober und h_{10} der Logarithmus der Anzahl der am 10. Oktober registrierten Fälle.

Und durch diese Datenpunkte wollen wir nun bestmöglich eine Gerade der Form

$$h(x) = \theta_1 + \theta_2 x$$

legen. Bestmöglich bedeutet, dass der Fehler über alle Datenpunkte gemittelt möglichst klein sein soll. Dafür definieren wir zunächst den Fehler im Datenpunkt x_j als

$$e_j := \frac{1}{2}(h(x_j) - h_j)^2 = \frac{1}{2}(\theta_1 + \theta_2 x_j - f_j)^2$$

wobei das Quadrat (u.a.) sicherstellt, dass alle Fehler positiv sind und das "$\frac{1}{2}$" einfach ranmultipliziert wird um nachher beim Ableiten einen Faktor zu sparen.

Und der gesammelte Fehler ist dann gegeben durch die Fehlerfunktion e als die Summer aller Fehler:

$$e = e_1 + e_2 + \dots + e_N = \sum_{j=1}^{N} e_j = \frac{1}{2}\sum_{j=1}^{N}(\theta_1 + \theta_2 x_j - f_j)^2$$

Jan kann erkennen, dass die Funktion e für verschiedene (θ_1, θ_2) verschiedene Werte annimmt und dass ein minimaler Wert von e einen minimalen Fehler in der Approximation der Daten durch $h(x) = \theta_1 + \theta_2 x$ bedeutet.

Und wie finden wir die Minimalstelle von e beziehungsweise die optimalen (θ_1, θ_2)? Fast wie in der Schule:

1. die Funktion e ableiten,
2. eine Nullstelle der Ableitung finden.

Dafür müssen wir erstmal verstehen, dass die Funktion eine Funktion von 2 Variablen ist (und dass das ein grosses Thema der Vorlesung werden wird).

Fürs erste sei gesagt, dass Jan die Variablen (θ_1, θ_2) separat betrachten kann und danach ableiten und null setzen. Also nach der Summen– und der Kettenregel:

$$\begin{aligned}\frac{d}{d\theta_1}e(\theta_1,\theta_2) &= \frac{1}{2}\sum_{j=1}^{N}\frac{d}{d\theta_1}(\theta_1+\theta_2 x_j - f_j)^2 = \frac{1}{2}\sum_{j=1}^{N}2(\theta_1+\theta_2 x_j - f_j)\\ &= \sum_{j=1}^{N}(\theta_1+\theta_2 x_j - f_j)\end{aligned}$$

und

$$\begin{aligned}\frac{d}{d\theta_2}e(\theta_1,\theta_2) &= \frac{1}{2}\sum_{j=1}^{N}\frac{d}{d\theta_2}(\theta_1+\theta_2 x_j - f_j)^2 = \frac{1}{2}\sum_{j=1}^{N}2(\theta_1+\theta_2 x_j - f_j)x_j\\ &= \sum_{j=1}^{N}(\theta_1+\theta_2 x_j - f_j)x_j\end{aligned}$$

abgeleitet sodass das Nullsetzen

$$\begin{aligned}0 &= \sum_{j=1}^{N}(\theta_1+\theta_2 x_j - f_j)\\ 0 &= \sum_{j=1}^{N}(\theta_1+\theta_2 x_j - f_j)x_j\end{aligned}$$

uns zwei Gleichungen ergibt, die (in aller Regel) die zwei unbekannten Parameter θ_1 und θ_2 (eindeutig) bestimmen.

Mit den konkreten Zahlen von oben gerechnet (und auf 2 Nachkommastellen gekürzt) identifizieren wir

$$\theta_1^* = 7.88, \quad \theta_2^* = 0.12$$

als die optimale Wahl und damit

$$h^*(x) = 7.88 + 0.12x$$

als die beste Gerade; Jan vergleiche Abbildung 3.

Natürlich können wir mit

$$y = 2^{\log_2(y)}$$

auch das *logarithmieren* wieder rückgängig und auch die Ausgleichsgerade auf die lineare (nicht *logarithmische*) Datenskalierung transformieren

$$g(x) = 2^{\theta_1+\theta_2 x} = 2^{\theta_1}\cdot 2^{\theta_2 x}$$

was uns mit den Parametern von oben die optimale Ausgleichskurve als

$$g^*(x) = 2^{7.88+0.12x} = 2^{7.88}\cdot 2^{0.12x} = 236.16\cdot 2^{0.12x}$$

ergibt (dargestellt in Abbildung 4).

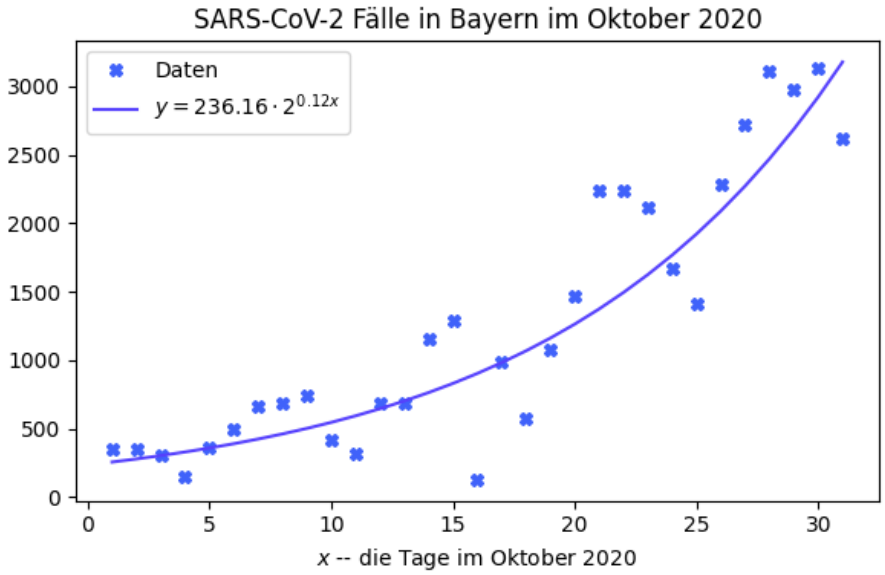


Figure 2.2: SARS-CoV-2 Fälle in Bayern im Oktober 2020 und die mittels linearer Regression auf der logarithmischen Skala bestimmte Ausgleichskurve

2.5 Mathematische Konzepte in der Linearen Regression

In diesem Beispiel haben wir schon einige Konzepte und Methoden benutzt, die wir im Laufe der Vorlesung in allen Details kennenlernen werden.

Neben der grundlegenden Technik der *linearen Regression* zur Datenanalyse werden wir uns eingehend mit

- Elementarfunktionen (wie hier der Logarithmus und die Exponentialfunktion, aber auch trigonometrische Funktionen und weitere),
- Differentiation und Integration von Funktionen und
- dem Lösen linearer Gleichungssysteme

beschäftigen.