

Das Wetter als Risikofaktor in der Wirtschaft: Wie lassen sich Wetterrisiken von Unternehmen absichern und wie kann Python dabei behilflich sein?

Thomas Robert Holy

Vortrag

Im Rahmen der
AG Finance & Risk

29. August 2020

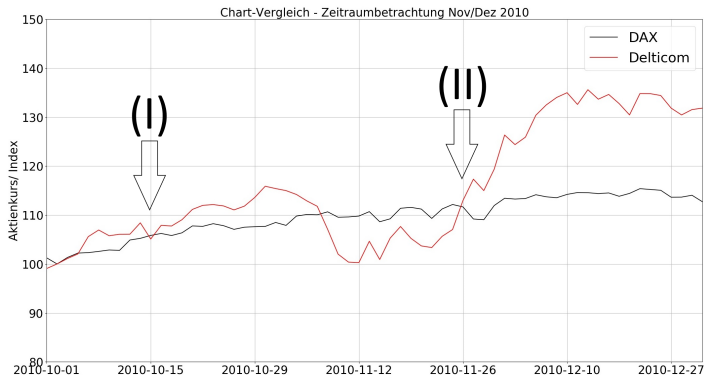
1) Einleitung

- Wetterphänomene beeinflussen die Wirtschaft (u.a. Aktienkurse, Umsätze, Gewinne)

Beispiel:

- (I) Oktober 2010:** Einführung gesetzlicher Winterreifenpflicht
→ Hohe Nachfrage nach Winterreifen
→ Aktienkurs von Delticom stieg im Vergleich zum Gesamtmarkt stark an
- (II) November 2010:** Kräftiger Wintereinbruch
→ Aktienkurs von Delticom stieg im Vergleich zum Gesamtmarkt stark an
- Aber: Entwicklungen nicht immer vorteilhaft
→ Risikoabsicherung notwendig

Abbildung: Aktienkurs der Delticom AG v.s. DAX (normiert auf 100 am 04.10.2010), eigene Darstellung.



1) Einleitung

Tabelle: Wetterabhängige Branchen und deren Risiken, eigene Darstellung in Anlehnung an Cui/Swishchuk, S. 2, (2015).

Branche	Meteorologische Elemente	Risiko
Energieversorger	Temperatur, Niederschlag, Wind	Übermäßige oder verringerte Versorgung
Landwirtschaft	Niederschlag, Temperatur	Ernteertrag, Lagerung, Schädlinge
Bauwirtschaft	Temperatur, Niederschlag, Schneefall	Verzögerungen, höhere Baukosten
Reisebranche	Temperatur, Niederschlag, Schneefall	Stornierungen, geringere Einnahmen

1) Einleitung

Abbildung: Das Wetter als Fluch und Segen zugleich? Quelle:
https://media0.faz.net/ppmedia/aktuell/gesellschaft/735499708/1.5824294/format_top1_breit/ausnahmejahrgang-traubenernte.jpg;
<https://www.spektrum.de/fm/912/thumbnails/iStock-177506643.jpg.4654334.jpg>.

- Fehlender Niederschlag, hohe Temperaturen im Sommer 2018
→ Dürreschäden in der Landwirtschaft
→ Weinbauern hoffen auf einen guten Jahrgang



1) Einleitung

Fragestellung dieses Vortrags:

- 1) Wie können sich betroffene Unternehmen vor Wetterrisiken absichern?
- 2) Wie kann Python dabei als Hilfsmittel eingesetzt werden?

Inhaltsverzeichnis

- 1) Einleitung
- 2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen
- 3) Python als Hilfsmittel zur Datenanalyse und zur Bewertung von Wetterderivaten
 - 3.1) Einblick in das Programm
 - 3.2) Eine Programmdemonstration
- 4) Betrachtung der Ergebnisse anhand des Beispiels Jena-Sternwarte
 - 4.1) Datenherkunft und Datensatz
 - 4.2) Datenauswertung und Berechnungen
 - 4.3) Problem Basisrisiko
- 5) Fazit

Inhaltsverzeichnis

- 1) Einleitung
- 2) Absicherung von Wetters Risiken mit Wetterderivaten: Grundlagen**
- 3) Python als Hilfsmittel zur Datenanalyse und zur Bewertung von Wetterderivaten
 - 3.1) Einblick in das Programm
 - 3.2) Eine Programmdemonstration
- 4) Betrachtung der Ergebnisse anhand des Beispiels Jena-Sternwarte
 - 4.1) Datenherkunft und Datensatz
 - 4.2) Datenauswertung und Berechnungen
 - 4.3) Problem Basisrisiko
- 5) Fazit

2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen

Definition Wetterderivat

- Geschäft auf dem Terminmarkt zur Absicherung von Wetterrisiken
- Stützt sich auf meteorologische Variablen wie z.B. die Temperatur
- Konstruktion eines Index, welcher die Ausprägungen der Variable innerhalb einer Periode notiert bspw. Degree-Day-Indizes
 - ↪ Unterscheidung in Heating-Degree-Days (HDD) und Cooling-Degree-Days (CDD)
 - HDDs: Ermittlung i.d.R. vom 1. November bis zum 28/ 29. Februar
 - CDDs: Ermittlung i.d.R. vom 1. Mai bis 30. September
- Messung der relativen „Kälte“/ „Wärme“, welche sich aus der Unterschreitung/ Überschreitung des Referenzwertes i.H.v. 18°C ergibt

2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen

Ermittlung der Indizes

- Berechnung des HDD/ CDD-Index durch

$$\text{HDD}(T_1, T_2) = \sum_{t=T_1}^{T_2} \max\{18 - Y_t, 0\} \text{ für kumulierte HDDs in den Wintermonaten;} \quad (1)$$

$$\text{CDD}(T_1, T_2) = \sum_{t=T_1}^{T_2} \max\{Y_t - 18, 0\} \text{ für kumulierte CDDs in den Sommermonaten,} \quad (2)$$

wobei $T_1 < T_2$ gilt und Y_t die Tagesdurchschnittstemperatur repräsentiert

2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen

Long-Call Wetter-Option

- Absicherung eines kalten Winters
- Mit dem Kauf des Calls wird eine Options-Prämie fällig.

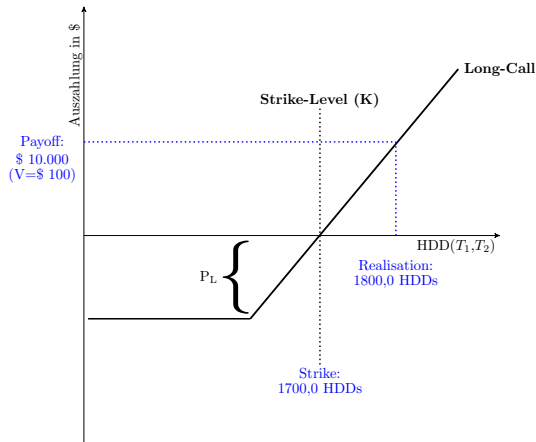
Payoff einer HDD-Option

- Options-Payoff:

$$P_{HDD}^{LC}(T_1, T_2) = V \cdot \max\{HDD(T_1, T_2) - K, 0\} - P_L$$

- Strike-Level K : z.B. \emptyset HDD-Index
- Tick-Size V : Zahlungsbetrag pro Indexpunkt
- Optionsprämie: P_L

Abbildung: Payoff-Struktur einer HDD-Option, eigene Darstellung in Anlehnung an Berg et al., S. 160, (2005).



2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen

Bewertungsmöglichkeit Burn-Analysis

- Grundannahme: Aktuelle Verteilung der Wetter-Indizes gleicht der zukünftigen Verteilung
- Empfehlung: Betrachtung einer Periode im Umfang von 20-30 Jahren (Normalperiode)
- Ermittlung des „Fair-Values“ (Optionsprämie, bei dem der erwartete Gewinn des Kontraktes Null beträgt)
- Bestimmung des Fair-Values F einer Call-Option durch

$$F = e^{-r\tau} \cdot \left[\frac{1}{n} \sum_{t=1}^n P_t \right], \quad (3)$$

mit Diskontfaktor $e^{-r\tau}$, risikolosem Zinssatz r , Fälligkeitszeitpunkt τ , n Payoffs P_t im Zeitpunkt t

3.1) Grundlagen von Derivaten und Wetterderivaten und deren Bewertung

Vorgehen bei Anwendung der Burn-Analysis

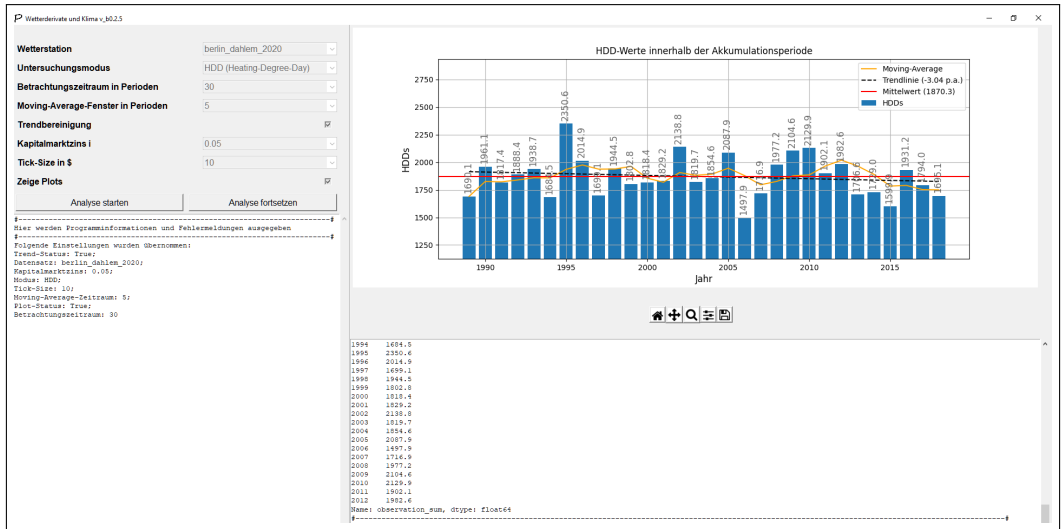
- 1) Bereinigung des Datensatzes
- 2) Errechnung historischer Wetter-Indizes
- 3) Bildung des arithmetischen Mittels aus der Gesamtheit dieser Indizes (Strike-Level der Option)
- 4) Bestimmung des historischen Payoffs für jedes Jahr im Betrachtungszeitraum
- 5) Ermittlung des „Schadensdurchschnittes“
- 6) Diskontierung auf den Zeitpunkt der Fälligkeit

Inhaltsverzeichnis

- 1) Einleitung
- 2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen
- 3) Python als Hilfsmittel zur Datenanalyse und zur Bewertung von Wetterderivaten**
 - 3.1) Einblick in das Programm
 - 3.2) Eine Programmdemonstration
- 4) Betrachtung der Ergebnisse anhand des Beispiels Jena-Sternwarte
 - 4.1) Datenherkunft und Datensatz
 - 4.2) Datenauswertung und Berechnungen
 - 4.3) Problem Basisrisiko
- 5) Fazit

3.1) Einblick in das Programm

Abbildung: Programm-Überblick, eigene Darstellung.



3.1) Einblick in das Programm

Datensatz einlesen

```
1 def read_in_raw_dataset(self, raw_data_):  
2     raw_data = read_csv(str(raw_data_) + '.csv', sep=";", decimal=".")  
3     raw_data.columns = [column.replace(' ', '') for column in raw_data.columns]  
4     return raw_data
```

- Verwendung des Moduls "Pandas" (Vgl. Excel)
 - Lies den gewünschten Datensatz ein!
 - Trenne bei ","!
 - Interpretierte "." als Zeichen für die Dezimaltrennung!
 - Benenne die Spalten um und gib den "DataFrame" zurück!

3.1) Einblick in das Programm

Datensatz Manipulieren

```

1 def create_data_df(self, raw_data_):
2     measurement_date = to_datetime(raw_data_.MESS_DATUM, format='%Y%m%d')
3     temperature = self.replace_missing_value(raw_values_=raw_data_.TMK, values_to_replace=[-999.0, -999],
                                                new_value_=nan)
4     rainfall = self.replace_missing_value(raw_values_=raw_data_.RSK, values_to_replace=[-999.0, -999],
                                                new_value_=nan)
5     HDD = self.calculate_HDD(temperature_=temperature)
6     CDD = self.calculate_CDD(temperature_=temperature)
7
8     create_data_df_dict = {'TMK': temperature,
9                            'RSK': rainfall,
10                           'HDD': HDD,
11                           'CDD': CDD}
12
13     data_df = DataFrame(create_data_df_dict, index=measurement_date)
14     data_df.index.name = 'MESS_DATUM'
15     return data_df

```

- Erzeugung eines DataFrames mit den benötigten Informationen
 - Erstelle eine interpretierbare Datumsspalte!
 - Ersetze fehlende Temperatur-/ Niederschlagswerte usw.!
 - Gib den DataFrame zurück!

3.1) Einblick in das Programm

Datensatz analysieren - Prüfe auf fehlende Messwerte

```

1 def get_should_date_range(self, first_last_year_, mode_):
2     observed_years = list(range(first_last_year_[0], (first_last_year_[1] + 1)))
3     if str(mode_) == 'CDD': start_end_date = [(str(year) + '-05-01', str(year) + '-09-30')
4                                               for year in observed_years]
5     elif str(mode_) == 'HDD':
6         leap_years = self.get_leap_years(observations_=observed_years)
7         start_end_date = [(str(year - 1) + '-11-01', str(year) + '-02-29')
8                           if year in leap_years else (str(year - 1) + '-11-01', str(year) + '-02-28')
9                           for year in observed_years]
10    else: start_end_date = [(str(year) + '-01-01', str(year) + '-12-31') for year in observed_years]
11    mask = [date_range(date[0], date[1], freq='1D') for date in start_end_date]
12    whole_mask = [mask_ for sub_list in mask for mask_ in sub_list]
13    return to_datetime(whole_mask)
14
15 # -----
16 def compare_date_range(self, first_last_year_, mode_, data_df_):
17     should_date_range = self.get_should_date_range(first_last_year_=first_last_year_, mode_=str(mode_))
18     return setdiff1d(should_date_range, data_df_.index)

```

- Ermittle alle im Datensatz vorhandenen Aufzeichnungen und vergleiche diese mit einer generierten Zeitspanne!
- Gib ggf. anfallende Differenzen zurück!

3.1) Einblick in das Programm

Datensatz analysieren - Messwerte kumulieren

```

1 def calculate_observation_sum(self, first_last_year_, data_df_, mode_):
2     observed_years = list(range(first_last_year_[0], (first_last_year_[1] + 1)))
3     if str(mode_) == 'CDD': start_end_date = [(str(year) + '-05-01', str(year) + '-09-30')
4                                               for year in observed_years]
5
6     elif str(mode_) == 'HDD':
7         leap_years = self.get_leap_years(observations_=observed_years)
8         start_end_date = [(str(year) + '-11-01', str(year + 1) + '-02-29')
9                           if year in leap_years else (str(year) + '-11-01', str(year + 1) + '-02-28')
10                          for year in observed_years[:-1]]
11     else: start_end_date = [(str(year) + '-01-01', str(year) + '-12-31') for year in observed_years]
12     mask = [date_range(date[0], date[1], freq='1D') for date in start_end_date]
13     if str(mode_) in ['HDD', 'CDD']: observation_sum = [round(data_df_.loc[mask_, str(mode_)].sum(), 2)
14                                                         for mask_ in mask]
15     else: observation_sum = [round((data_df_.loc[mask_, str(mode_)].sum() /
16                                   len(data_df_.loc[mask_, str(mode_)])), 2) for mask_ in mask]
17
18     observation_sum_df = DataFrame({'observation_sum': observation_sum}, index=observed_years)
19     observation_sum_df.index.name = 'observed_years'
20     return observation_sum_df

```

- Erzeuge iterativ Akkumulationsperioden in Abhängigkeit vom Modus und summiere Messwerte auf!

3.2) Eine Programmdemonstration

Beispiel Programmdurchlauf

Stadt: Nürnberg München Nürnberg-Netzstall Brocken Berlin-Tempelhof Berlin-Schönfeld

Betrachtungsperiode: 20 - 30 Jahre

Modus: HDD CDD

Kapitalmarktzins: 1-15%

Tick-Size: 10-100 \$

Inhaltsverzeichnis

- 1) Einleitung
- 2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen
- 3) Python als Hilfsmittel zur Datenanalyse und zur Bewertung von Wetterderivaten
 - 3.1) Einblick in das Programm
 - 3.2) Eine Programmdemonstration
- 4) Betrachtung der Ergebnisse anhand des Beispiels Jena-Sternwarte**
 - 4.1) Datenherkunft und Datensatz
 - 4.2) Datenauswertung und Berechnungen
 - 4.3) Problem Basisrisiko
- 5) Fazit

4.1) Datenherkunft und Datensatz

Datenherkunft

- Der DWD stellt Wetteraufzeichnungen für über 600 deutsche Wetterstationen zur Verfügung¹
- Beispiel: Messstation 2444 (Jena-Sternwarte)
- Der Datensatz umfasst den 01.01.1824 - 31.12.2019
→ ca. 70.000 Einträge, wobei einige Einträge fehlen (z.B. Mai 1945)

Datensatzauszug

```

1 STATIONS_ID;MESS_DATUM;QN_3;FX;FM;QN_4;RSK;RSKF;SDK;SHK_TAG;NM;VPM;PM;TMK;UPM;TXK;...;eor
2
3 2444;18240101;-999;-999;-999;2;-999;-999;-999;-999;-999;-999;990.70;4.5;-999;5.5;...;eor
4 2444;18240102;-999;-999;-999;2;-999;-999;-999;-999;-999;-999;984.00;5.0;-999;6.5;...;eor
5
6 ...
7
8 2444;20191230;-999;-999;-999;3;0.0;0;-999;0;-999;3.9;-999;3.9;50.79;9.6;-1.3;-2.2;...;eor
9 2444;20191231;-999;-999;-999;3;0.0;0;-999;0;-999;5.5;-999;3.3;70.92;6.3;-0.5;-2.7;...;eor

```

¹Siehe https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/kl/historical/

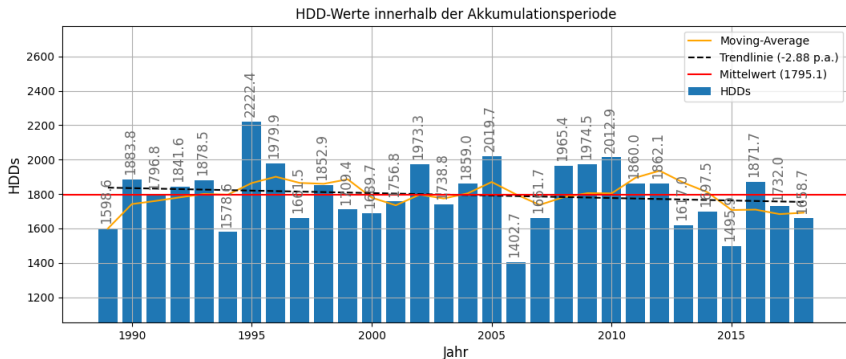
4.2) Datenauswertung und Berechnungen

Abbildung: HDD-Werte innerhalb der jeweiligen Akkumulationsperiode im Betrachtungszeitraum (30 Jahre; Jena-Sternwarte), eigene Darstellung.

- **Blau:** Kumulierte HDDs für 1989 - 2019
- **Rot:** Mittelwert ($\hat{=}$ Strike/ 1795.1 HDDs)
- **Schwarz:** Trendlinie (-2,88 HDDs p.a.)

Bspw.:

- 2016/17: 1871.7 HDDs
- 2017/18: 1732.0 HDDs
- 2018/19: 1658.7 HDDs



4.2) Datenauswertung und Berechnungen

→ Ticksiz: \$ 10

→ KM-Zins: 5%

Trendbereinigt:

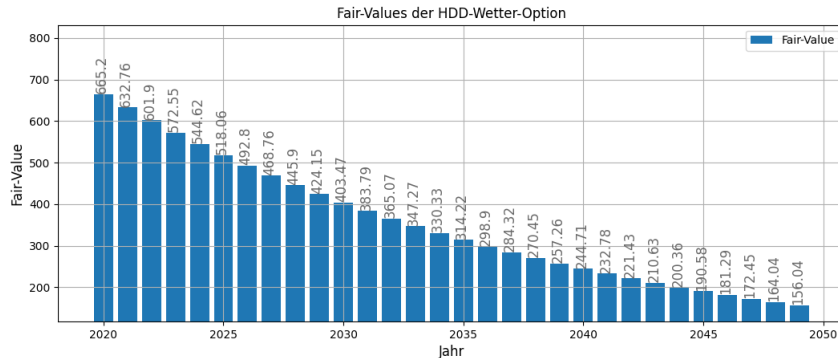
- 2049: \$ 156.04
- ...
- 2021: \$ 632.76
- 2020: \$ **665.20**

Nicht trendbreinigt:

- 2049: \$ 158.62
- ...
- 2021: \$ 643.24
- 2020: \$ **676.22**

⇒ $\Delta \$ 2020$: **11.02** ($\hat{=}$ 1.66%)

Abbildung: Fair-Values der HDD-Wetter-Option für die jeweiligen Laufzeiten (bis zu 30 Jahre; Jena-Sternwarte), eigene Darstellung.



4.2) Datenauswertung und Berechnungen

- Kumulierte CDDs für 1989 - 2019

Trendbereinigt:

- 2017: 162.75 CDDs
- 2018: 288.09 CDDs
- 2019: **231.03** CDDs

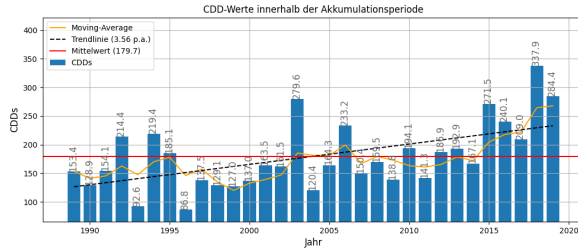
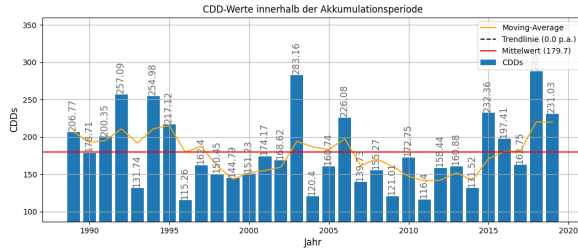
Nicht Trendbereinigt:

- 2017: 209.0 CDDs
- 2018: 337.9 CDDs
- 2019: **284.4** CDDs

⇒ $\Delta\text{CDD } 2019$: **23.10%**

↪ $\Delta\text{\$ } 2020$: **15.32%**

Abbildung: Vergleich (nicht)-trendbereinigte CDD-Werte, eigene Darstellung.



4.2) Datenauswertung und Berechnungen

Zeitreihenanalyse - Trendkomponente berechnen:

Jahr	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
CDD	120	164	233	150	169	138	194	141	186	192	167	271	240	209	337	284

Formeln zur Berechnung der Trendgeraden:

$$a = \bar{y} - b \cdot \bar{t},$$

$$b = \frac{\sum_{t=1}^T t \cdot y_t - T \cdot \bar{y} \cdot \bar{t}}{\sum_{t=1}^T t^2 - T \cdot (\bar{t})^2}$$

Berechnung der notwendigen Größen:

$$b = \frac{30.234 - 16 \cdot 199.69 \cdot 8.5}{1496 - 16 \cdot 72.25} = 9.05,$$

$$\text{mit } t \cdot y_t = 30.234, \quad T = 16, \quad t^2 = 1.496, \quad (\bar{t})^2 = 72.25$$

$$a = 199.69 - 9.05 \cdot 8.5 = 122.78,$$

$$\text{mit } \bar{y} = 199.69, \quad \bar{t} = \frac{16 + 1}{2} = 8.5$$

Anwendung und Schätzung:

$$\hat{m}_t = 122.78 + 9.05 \cdot t$$

$$\hat{m}_{16} = 122.78 + 9.05 \cdot 16 = 267.55$$

$$\hat{m}_{17} = 122.78 + 9.05 \cdot 17 = 276.60$$

Trend laut Programm: 9.05 CDDs/Jahr

4.2) Datenauswertung und Berechnungen

Messwerte transformieren - Trendbereinigung

```

1 def trend_adjustment_by_linear_regression(self, x_, y_):
2     linear_model = LinearRegression()
3     linear_model.fit(x_, y_)
4     predictions = linear_model.predict(x_)
5     return [true_value - predicted_value for true_value, predicted_value in zip(y_, predictions)] + mean(y_)
6
7 # -----
8 def trend_adjustment(self, observation_sum_df_):
9     x_values, y_values = array(observation_sum_df_.index)\
10        .reshape(-1, 1), array(observation_sum_df_.observation_sum)
11     adjusted_values = self.trend_adjustment_by_linear_regression(x_=x_values, y_=y_values)
12     observation_sum_df_['observation_sum'] = adjusted_values
13     return observation_sum_df_

```

- Verwende ein lineares Regressionsmodell zur In-Sample Prognose!
- Bilde Residuen (Wahrer Wert - Prognosewert) und addiere Mittelwert zur Berechnung trendbereinigter Messwerte!

4.3) Problem Basisrisiko

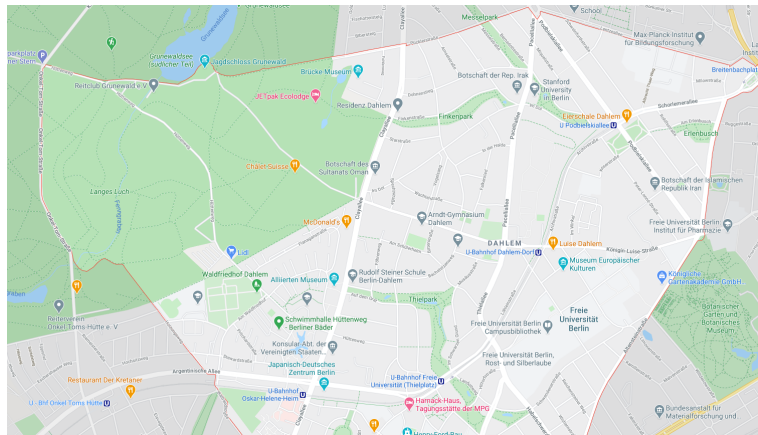
Definition

- „Restrisiko“, welches beim Halter des Derivates verbleibt
- Ursachen:
 - Geringe Korrelation zwischen Wetter- und Produktionsvariable
 - Geographisches Basisrisiko

Berlin-Dahlem

- Relativ viele Grünflächen
- Anstieg der Trendgeraden: 1.68 CDDs p.a.

Abbildung: Lage der Wetterstation Berlin-Dahlem, Quelle: Google Maps.

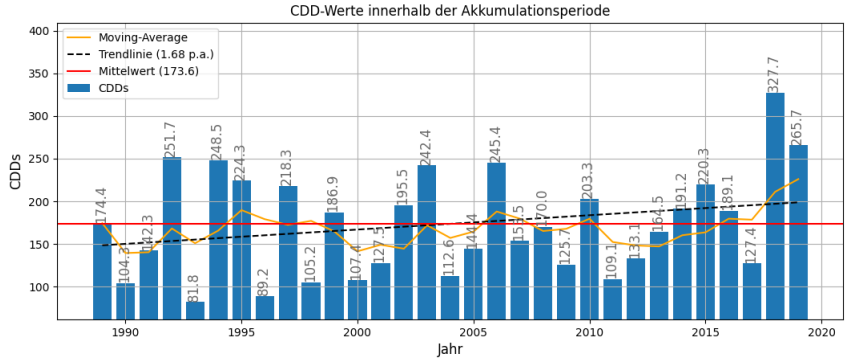


4.3) Problem Basisrisiko

Abbildung: CDD-Werte innerhalb der jeweiligen Akkumulationsperiode im Betrachtungszeitraum (30 Jahre; Berlin-Dahlem), eigene Darstellung.

Berlin-Dahlem

- Relativ viele Grünflächen
- Anstieg der Trendgeraden: 1.68 CDDs p.a.



4.3) Problem Basisrisiko

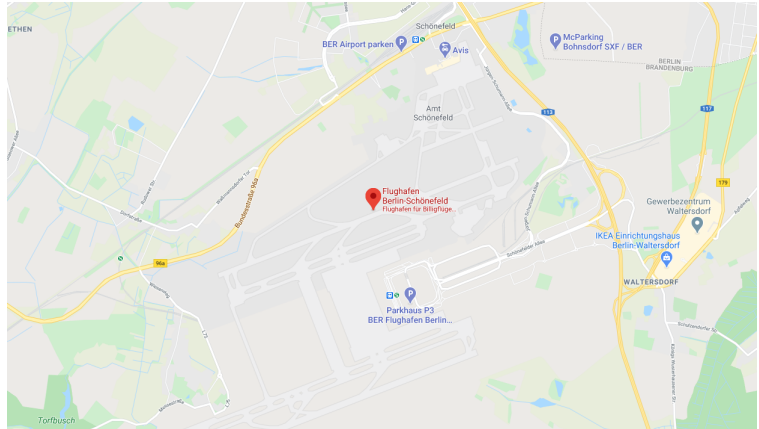
Berlin-Schönefeld

- Verhältnismäßig stark bebautes Gebiet, Nähe Flughafen
- Entfernung zu Berlin-Dahlem: ca. 20km Luftlinie
- Anstieg der Trendgeraden: 3.2 CDDs p.a.

Beobachtung:

- Relativ starke Abweichung des Trends
- U.a. verursacht durch Urbanisierungseffekt

Abbildung: Lage der Wetterstation Berlin-Schönefeld, Quelle: Google Maps.



4.3) Problem Basisrisiko

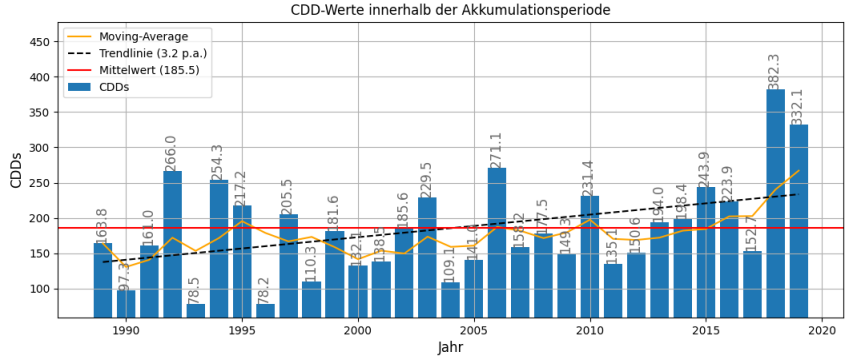
Berlin-Schönefeld

- Recht stark bebautes Gebiet, nahe Flughafen
- Entfernung zu Berlin-Dahlem: ca. 20km Luftlinie
- Anstieg der Trendgeraden: 3.2 CDDs p.a.

Beobachtung:

- Relativ starke Abweichung des Trends
- U.a. verursacht durch Urbanisierungseffekt

Abbildung: CDD-Werte innerhalb der jeweiligen Akkumulationsperiode im Betrachtungszeitraum (30 Jahre; Berlin-Schönefeld), eigene Darstellung.



4.3) Problem Basisrisiko

Wetterstation Hohenpeißenberg

- Südwestlich von München
- Nahe Waldgebiet und Feldern
- Wetteraufzeichnungen seit 1781
- Lückenlos seit 1812
→ 205 Jahre

Abbildung: Lage Wetterstation Hohenpeißenberg, Quelle: Google Maps.



Abbildung: Lage Hohenpeißenberg, Quelle: Google Maps.

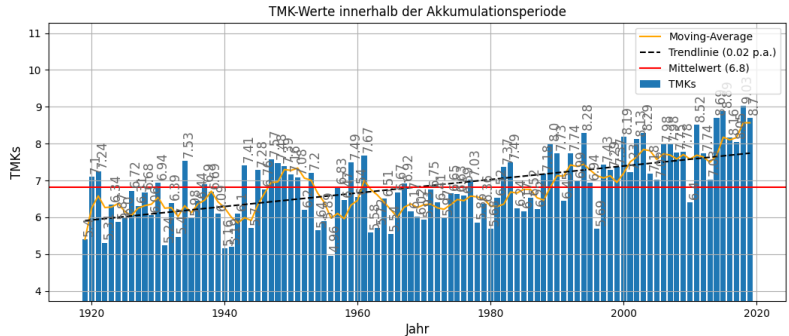


4.3) Problem Basisrisiko

Jahresdurchschnittstemperatur

- Anstieg Trendgerade:
 $0.02\text{ }^{\circ}\text{C}/\text{Jahr}$
 $\rightarrow \Delta$ seit 1920: ca. $2,0^{\circ}\text{C}$
- Laut DWD in 136 Jahren:
 $1,4^{\circ}\text{C}$ (Mittelwert
 mehrerer Stationen)
- Moving-Average seit ca. 30
 Jahren über langjährigem
 Durchschnitt
 (deutschlandweit)
- Wetterrisiken in Zukunft
 evtl. von größerer
 Bedeutung für
 Unternehmen

Abbildung: Entwicklung der Jahresdurchschnittstemperatur (TMK) Hohenpeißenberg (100 Jahre), eigene Darstellung.



Inhaltsverzeichnis

- 1) Einleitung
- 2) Absicherung von Wetterrisiken mit Wetterderivaten: Grundlagen
- 3) Python als Hilfsmittel zur Datenanalyse und zur Bewertung von Wetterderivaten
 - 3.1) Einblick in das Programm
 - 3.2) Eine Programmdemonstration
- 4) Betrachtung der Ergebnisse anhand des Beispiels Jena-Sternwarte
 - 4.1) Datenherkunft und Datensatz
 - 4.2) Datenauswertung und Berechnungen
 - 4.3) Problem Basisrisiko
- 5) Fazit**

5) Fazit

- Wetterphänomene haben einen Einfluss auf die Wirtschaft
- Für einige Branchen ist es vorteilhaft sich gegen Wetterrisiken abzusichern
- Python ist zur Analyse von großen Datensätzen geeignet
- Fehlerhafte/ unvollständige Datensätze stellen den Analysten vor Probleme
- Potenzielle Einflussfaktoren müssen in Erfahrung gebracht und ggf. berücksichtigt werden
- Änderung des Klimas erkennbar

Vielen Dank für Ihre Aufmerksamkeit!

Quellenverzeichnis

- Berg, E., Schmitz, B., Starp, M., & Trenkel, H. (2005): Wetterderivate: Ein Instrument im Risikomanagement für die Landwirtschaft?. In: German Journal of Agricultural Economics, 54(3), S. 158-170.
- Bösch, M. (2014): Derivate: verstehen, anwenden und bewerten, 3. Auflage. Vahlen, München.
- Bloss, M., & Ernst, D. (2008): Derivate: Handbuch für Finanzintermediäre und Investoren: Handbuch für Finanzintermediäre und Investoren. Oldenbourg, München [u.a.].
- Brockett, P. L., Wang, M., & Yang, C. (2005): Weather derivatives and weather risk management. In: Risk Management and Insurance Review, 8(1), S. 127-140.
- Cao, M., Li, A., & Wei, J. Z. (2003): Weather derivatives: A new class of financial instruments. URL: https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID1016123_code16402.pdf?abstractid=1016123&mirid=1, 27.08.2020.
- Cao, M., & Wei, J. (2004): Weather derivatives valuation and market price of weather risk. In: Journal of Futures Markets, 24(11), S. 1065-1089.
- Cui, K., & Swishchuk, A. V. (2015): Applications of weather derivatives in the energy market, University of Calgary, Calgary, S. 1-16.
- Delticom AG (2011): Geschäftsbericht 2010, URL: https://www.delti.com/Investor_Relations/Delticom_Geschaeftsbericht_2010.pdf, 27.08.2020.

Quellenverzeichnis

DWD (o.J.)a: Messdaten für der Messstation Jena-Sternwarte, URL:

https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/kl/historical/tageswerte_KL_02444_18240101_20191231_hist.zip, 27.08.2020.

DWD (o.J.)b: „Klima“ im Wetterlexikon, URL:

<https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?nn=103346&lv2=101334&lv3=101462>, 27.08.2020.

DWD (2017): Nationaler Klimareport, URL: https://www.dwd.de/DE/leistungen/klimastatusbericht/publikationen/ksb_2017.pdf?__blob=publicationFile&v=2, 07.08.2020.

Ellithorpe, D., & Putnam, S. (2000). Weather derivatives and their implications for power markets. The Journal of risk finance, 1(2), 19-28.

Google (2020)a: Dahlem, Berlin, URL:

<https://www.google.com/maps/place/Dahlem,+Berlin/@52.4561088,13.2611936,5560m/data=!3m2!1e3!4b1!4m5!3m4!1s0x47a85a130a2917e5:0x52120465b5facc0!8m2!3d52.4582353!4d13.2870827>, 27.08.2020.

Google (2020)b: Schönefeld, Berlin, URL: <https://www.google.com/maps/search/sch%C3%B6nefeld,+Berlin/@52.3657605,13.4420915,22285m/data=!3m2!1e3!4b1>, 27.08.2020.

Google (2020)c: Wetterstation, Albin-Schwaiger-Weg, Hohenpeißenberg, URL:

<https://www.google.com/maps/search/Wetterstation,+Albin-Schwaiger-Weg,+Hohenpei%C3%9Fenberg/@47.7915893,11.0034547,9482m/data=!3m1!1e3>, 27.08.2020.

Quellenverzeichnis

- Geman, H., & Leonardi, M. P. (2005): Alternative approaches to weather derivatives pricing. In: Managerial Finance, 31(6), S. 46-72.
- Hull, J., Mader, W., & Wagner, M. (2015): Optionen, Futures und andere Derivate, 9. Auflage. Pearson, Hallbergmoos.
- Jewson, S., & Brix, A. (2005): Weather derivative valuation: the meteorological, statistical, financial and mathematical foundations. Cambridge University Press, Cambridge.
- Kriener, M. (2018): Der 2018er wird der perfekte Wein für den knausrigen Deutschen, tagesspiegel.de. URL: <https://www.tagesspiegel.de/wirtschaft/rekord-weinlese-der-2018er-wird-der-perfekte-wein-fuer-den-knausrigen-deutschen/23149446.html>, 27.08.2020.
- Koers, M. (o.D.): Abkühlung mit der Höhe, URL: <http://www.belecke-wetter.de/index.php/83-wetter/aktuell/51-abkuehlung-mit-der-hoehe>, 27.08.2020.
- Lee, Y., & Oren, S. S. (2009): An equilibrium pricing model for weather derivatives in a multi-commodity setting. In: Energy Economics, 31(5), S. 702-713.
- Mußhoff, O., Odening, M. & Xu, W. (2007): Management klimabedingter Risiken in der Landwirtschaft-Zum Anwendungspotenzial von Wetterderivaten. In: Agrarwirtschaft und Agrarsoziologie, 1, 07, S. 27-48.

Quellenverzeichnis

- Nelken, I. (2000): Weather derivatives - pricing and hedging. Super Computer Consulting Inc. Mundelein, Illinois, URL: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj23Y2x5rrrAhWmNOwKHSTWBRcQFjAAegQIAxAB&url=http%3A%2F%2Fquantlabs.net%2Facademy%2Fdownload%2Ffree_quant_institutional_books_%2F%5BSuper%2520Computer%2520Consulting%2C%2520Nelken%5D%2520Weather%2520Derivatives%2520-%2520Pricing%2520and%2520Hedging.pdf&usg=AOvVaw1Nlq_EFGghNMzrYK_yd0m6, 27.08.2020.
- Odening, M., Mußhoff, O., & Xu, W. (2007): Analysis of rainfall derivatives using daily precipitation models: Opportunities and pitfalls. In: Agricultural Finance Review, 67(1), S. 135-156.
- Odening, M., & Xu, W. (2007). Management klimabedingter Risiken in der Landwirtschaft-Zum Anwendungspotenzial von Wetterderivaten. Agrarwirtschaft und Agrarsoziologie, 1, 07.
- Richards, T. J., Manfredo, M. R., & Sanders, D. R. (2004). Pricing weather derivatives. American Journal of Agricultural Economics, 86(4), 1005-1017.
- Riedler, D. W. I. J., Voll, S. S., & Philipps, F. (2013): Wetter-und Katastrophenderivate, Bundesministerium für Bildung und Forschung, Mannheim. URL: http://www.cfi21.org/fileadmin/user_upload/pdfs/Berichte/2013_Wetter-_und_Katastrophenderivate.pdf.
- Rudolph, B., & Schäfer, K. (2010): Derivative Finanzmarktinstrumente: eine anwendungsbezogene Einführung in Märkte, Strategien und Bewertung, 2. Auflage. Springer, Berlin [u.a.].

Quellenverzeichnis

- Roustant, O., Laurent, J. P., Bay, X., & Carraro, L. (2003): Model risk in the pricing of weather derivatives. *Banque and Marche*, S. 1-25.
- Schirm, A. (2000): Wetterderivate–Finanzmarktprodukte für das Management wetterbedingter Geschäftsrisiken. In: *Finanz Betrieb*, 2(11), S. 722-730.
- Schirm, A. (2001): Wetterderivate–Einsatzmöglichkeiten und Bewertung. *Research in Capital Markets and Finance*. 2001, 2, Universität Mannheim, Mannheim.
- Torró, H., Meneu, V., & Valor, E. (2003): Single factor stochastic models with seasonality applied to underlying weather derivatives variables. In: *The Journal of Risk Finance*, 4(4), S. 6-17.