Zeitreihenanalyse

Python für Finanzen

- Es ist an der Zeit, dass wir uns auf Zeitreihendaten konzentrieren!
- Viele unserer Finanzinformationen werden in Form eines Wertes erscheinen, der gegen eine Zeitreihe aufgetragen wird.

Python für Finanzen

- Obwohl die in diesem Abschnitt des Kurses vorgestellten Konzepte sehr wichtig sind, können wir sie nicht oft verwenden, wenn wir direkt mit unseren algorithmischen Handelsmodellen arbeiten.

Python für Finanzen

- Einer unserer Hauptgründe, diese Themen zu behandeln, ist, dass wir in zukünftigen Abschnitten des Kurses zeigen können, warum einige dieser Analysetechniken für Bestandsinformationen eigentlich KEINE gute Idee sind.

Python für Finanzen

- Es kann sehr verlockend sein, einige dieser Techniken für Finanzdaten zu verwenden, aber manchmal ist es eigentlich keine gute Idee, und um zu verstehen, warum das so ist, müssen wir zuerst die Techniken selbst verstehen.

Python für Finanzen

- Als Gesamtkonzept für diesen Abschnitt solltest Du versuchen, ein besseres Verständnis für einige dieser Konzepte zu erlangen, aber kümmere Dich nicht zu sehr um die Details (was zukünftige Abschnitte des Kurses angeht).

Python für Finanzen

- Was wir in diesem Abschnitt diskutieren:
  - Zeitreihen-Grundlagen
  - Statsmodels Python-Bibliothek
  - ETS-Modelle und Dekomposition
  - EWMA-Modelle
  - ARIMA Modelle

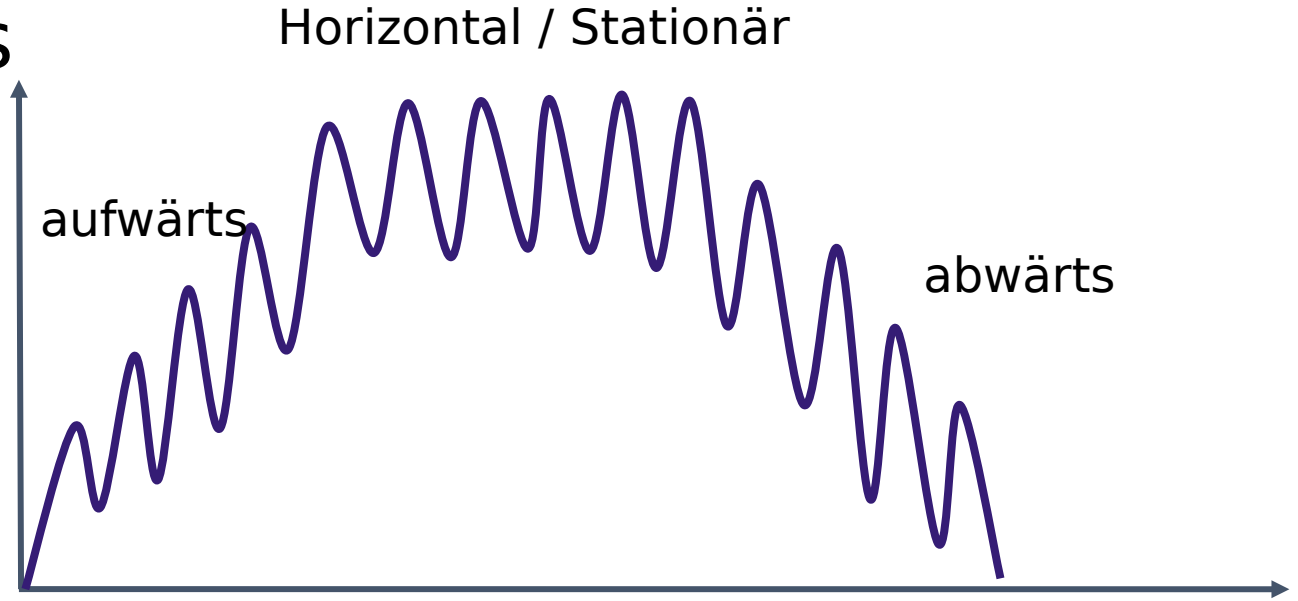
Zeitreihen-Grundlagen

Python für Finanzen

- Wir beginnen mit der Diskussion einiger wichtiger Zeitreihenkonzepte.
- Zeitreihendaten haben bestimmte Eigenschaften, werfen wir einen Blick auf einige Diagramme und diskutieren einige wichtige Begriffe!

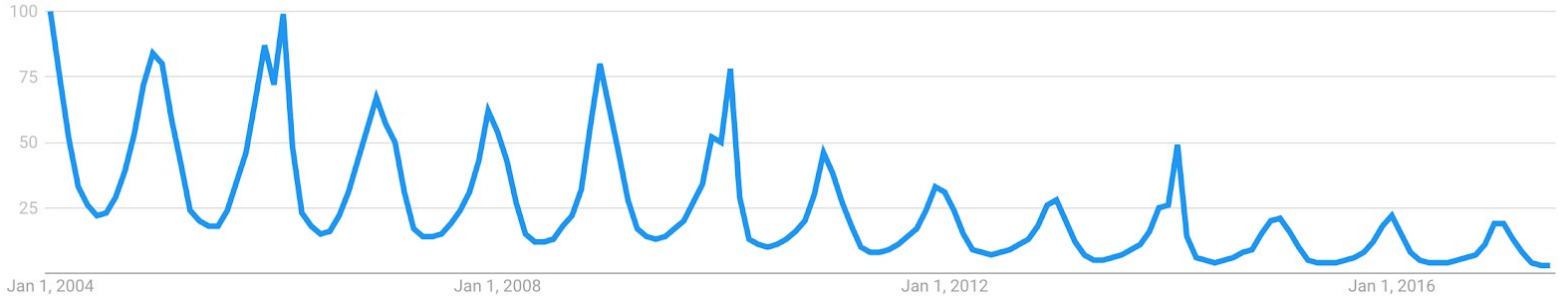
Python für Finanzen

- Trends



Python für Finanzen

- Saisonalität - sich wiederholende Trends

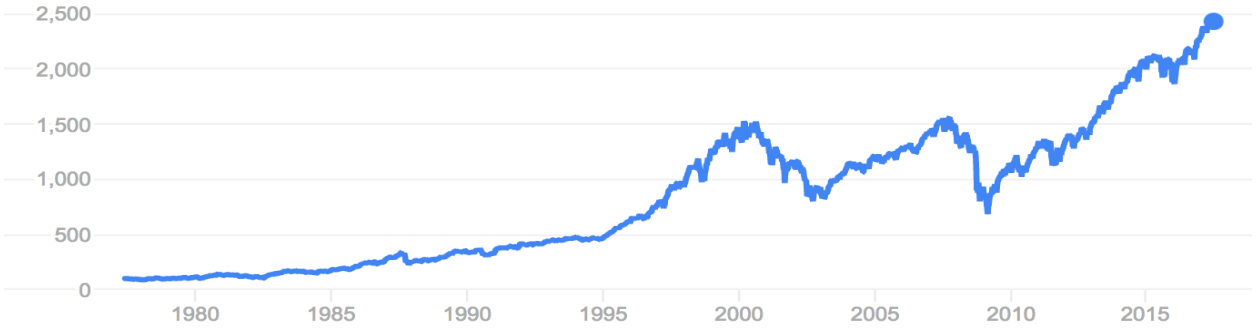


Google Trends - “Snowboarding”

Python für Finanzen

- Zyklisch - Trends ohne festgelegte Wiederholung

  (bei ökonomischen Zeitreihen auch Koniunktur genannt)



SP500

Python für Finanzen

- Nun, da wir einige der Grundlagen verstehen, lernen wir die populärste Bibliothek in Python kennen, um Zeitreihendaten zu behandeln: statsmodels!

Statsmodels

Python für Finanzen

- Die beliebteste Bibliothek in Python für den Umgang mit Zeitreihendaten ist die statsmodels-Bibliothek.
- Sie ist stark von der R-Programmiersprache inspiriert.

Python für Finanzen

- Statsmodels ist ein Python-Modul, mit dem Anwender Daten untersuchen, statistische Modelle schätzen und statistische Tests durchführen können.

Python für Finanzen

- Eine umfangreiche Liste von deskriptiven Statistiken, statistischen Tests, Plotfunktionen und Ergebnisstatistiken steht für verschiedene Datentypen und für jeden Schätzwert(estimator) zur Verfügung.

Python für Finanzen

- Zur manuellen Installation kannst du folgenden Befehl verwenden:
  - conda install statsmodels

    oder
  - Pip install statsmodels

Python für Finanzen

- Wir werden uns die Dokumentation anschauen und dann eine einfache Demonstration dessen durchführen, wofür wir Statistikmodelle in Bezug auf Zeitreihendaten verwenden können.

ETS-Modelle

Überblick

- Für die nächsten Lektionen werden wir die Themen konzeptionell in Folien diskutieren.
- Anschließend werden wir diese Themen mit Python und Statsmodels erneut durchgehen, um sie zu implementieren!

ETS (Error-Trend-Seasonality)

- ETS-Modelle (Fehler-Trend-Saisonalität)
  - Exponentielle Glättung(Smoothing)
  - Trend Methoden Modelle
  - ETS-Dekomposition
- Wir werden mit mehreren davon mit der Python Statsmodels-Bibliothek arbeiten!

Exponentielle Glättung

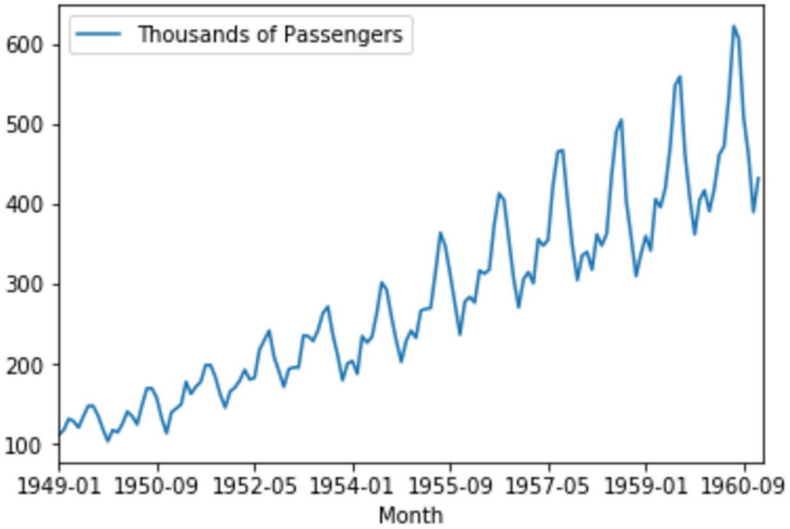
- ETS (Error-Trend-Seasonality)-Modelle verwenden jeden dieser Begriffe zum "Glätten (smoothing)" und können sie addieren, multiplizieren oder sogar einfach auslassen.
- Ausgehend von diesen Schlüsselfaktoren können wir versuchen, ein Modell zu erstellen, das unseren Daten entspricht.

Zeitreihenzerlegung

- Zeitreihenzerlegung mit ETS (Error-Trend-Seasonality).
- Die Visualisierung der Daten basierend auf ihrer ETS ist eine gute Möglichkeit, um ein Verständnis für ihr Verhalten zu entwickeln.

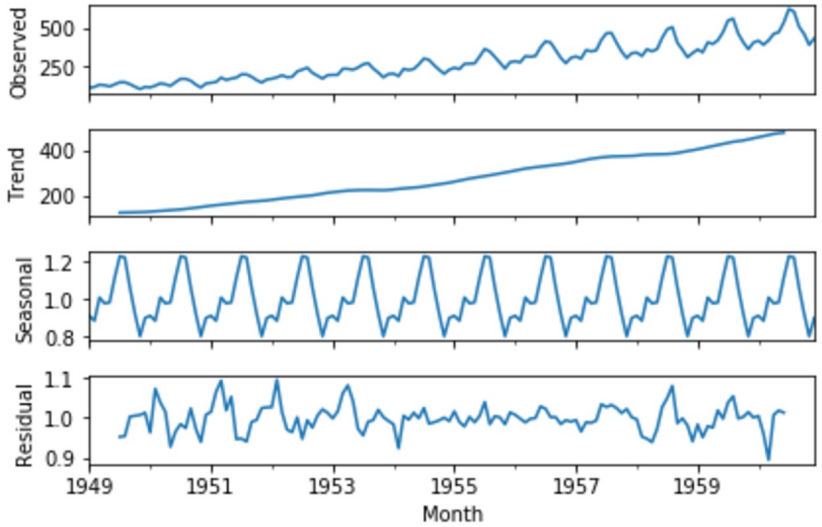
Zeitreihenzerlegung

- ETS Dekomposition - Flugpassagiere



Zeitreihenzerlegung

- ETS Dekomposition - Flugpassagiere



Ausblick

- Wir werden die Zeitreihenzerlegung erneut ansehen, wenn wir ARIMA-Modelle diskutieren.
- Lasst uns jetzt zu EWMA-Modellen übergehen!

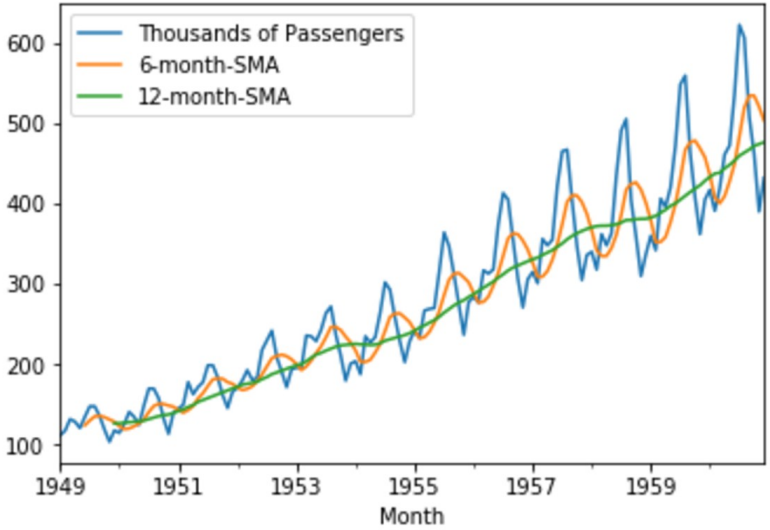
EWMA-Modelle

Überblick

- Wir haben bereits gesehen, wie die Berechnung einfacher gleitender Durchschnitte es uns ermöglichen kann, ein einfaches Modell zu erstellen, das ein Trendverhalten einer Zeitreihe beschreibt.

SMA - Einfache gleitende Durchschnitte

Simple Moving Averages



SMA “Schwächen”

- Kleinere Fenster führen eher zu mehr Rauschen als zu Signal
- Es wird immer um die Größe des Fensters hinterherlaufen (Lag)
- Aufgrund der Mittelwertbildung wird es niemals zu einem vollen Peak oder Tal der Daten gelangen

SMA “Schwächen”

- Informiert Dich nicht wirklich über mögliches zukünftiges Verhalten, sondern beschreibt Trends in Deinen Daten.
- Extreme historische Werte können Deine SMA erheblich verzerren.

EWMA

- Um einige dieser SMA Probleme zu beheben, können wir einen EWMA verwenden.

- EWMA - exponentiell gewichtete gleitende Mittelwerte (Exponentially Weighted Moving Averages)

EWMA

- EWMA wird es uns ermöglichen, den Lag-Effekt von SMA zu reduzieren, und es wird mehr Gewicht auf Werte legen, die in jüngerer Zeit aufgetreten sind (indem die neueren Werte stärker gewichtet werden, daher der Name).

Gewichtung

- Die Gewichtung der neuesten Werte hängt von den tatsächlichen Parametern ab, die in der EWMA verwendet werden, und von der Anzahl der Perioden, denen eine Fenstergröße zugewiesen wird.

Ausblick

- Wir werden in der nächsten Vorlesung ein Beispiel für die Verwendung von Pandas zur Erstellung von EWMA ausarbeiten!

ETS Dekomposition

Coding

EWMA
Anwendung

ARIMA
Modelle

Überblick

- Wir werden nun eines der gebräuchlichsten Zeitreihenmodelle, ARIMA, diskutieren.
- Bitte beachte, dass dies ein **optionaler** Teil des Kurses ist.

Überblick

- Aus verschiedenen Gründen, die wir später sehen werden, funktionieren ARIMA-Modelle oft nicht gut mit historischen Bestandsdaten (Aktiendaten).
- Sie sind jedoch so grundlegend für das Verständnis der Zeitreihenanalyse, dass es sich trotzdem lohnt, über sie zu reden.

Externe Links

- ARIMA-Modelle können komplex sein!
- Nutze die verschiedenen Links und zusätzlichen Quellen in diesem Abschnitt vollständig, wenn du später ARIMA-Modelle für andere Probleme verwenden möchtest.

ARIMA und ARMA

- Das ARIMA-Modell (AutoRegressive Integrated Moving Average) ist eine Verallgemeinerung eines ARMA-Modells (AutoRegressive Moving Average).

ARIMA und ARMA

- Beide Modelle (ARIMA und ARMA) sind an Zeitreihendaten angepasst, um entweder die Daten besser zu verstehen oder zukünftige Punkte in der Serie vorherzusagen (Prognose).

Saisonale ARIMA

- ARIMA (Autoregressive integrierte gleitende Durchschnitte)
  - Nicht saisonale ARIMA
  - Saisonale ARIMA

Saisonale ARIMA

- Wir beginnen mit der Diskussion nicht-saisonaler ARIMA-Modelle und gehen dann zu saisonalen ARIMA-Modellen über.
- Die Python-Beispiele am Ende verwenden saisonale ARIMA.

Stationarität

- ARIMA-Modelle werden in einigen Fällen angewendet, in denen Daten Anzeichen von Nicht-Stationarität zeigen, wobei ein anfänglicher Differenzierungsschritt (entsprechend dem "integrierten" Teil des Modells) einmal oder mehrmals angewendet werden kann, um die Nicht-Stationarität zu eliminieren.

Differenzierung

- Das Differenzieren ist eigentlich eine sehr einfache Idee, aber lass es uns erst mal beiseite legen und reden wir ein wenig mehr über ARIMA!
- Wir werden später auf die Differenzierung zurückkommen.
- Wir werden über die Hauptkomponenten von ARIMA sprechen.

Nicht Saisonale ARIMA

- Nicht saisonale ARIMA-Modelle werden allgemein als ARIMA (p, d, q) bezeichnet, wobei die Parameter p, d und q nicht-negative ganze Zahlen sind.
- Sehen wir uns an, was diese drei Komponenten sind!

Teile des ARIMA-Modells

AR (p): Autoregression

- Ein Regressionsmodell, das die abhängige Beziehung zwischen einer aktuellen Beobachtung und Beobachtungen in einer früheren Periode verwendet

Teile des ARIMA-Modells

I (d): integriert.

- Differenzieren von Beobachtungen (Subtrahieren einer Beobachtung von einer Beobachtung im vorherigen Zeitschritt), um die Zeitreihe stationär zu machen.

Teile des ARIMA-Modells

MA (q): gleitender Durchschnitt.

- Ein Modell, das die Abhängigkeit zwischen einer Beobachtungs und einem Restfehler aus einem gleitenden Durchschnittsmodell für verzögerte Beobachtungen verwendet.

Stationär vs nicht stationär

- Stationäre vs nicht stationäre Daten
  - Um ARIMA effektiv zu nutzen, müssen wir die Stationarität in unseren Daten verstehen.
  - Was macht einen Datensatz stationär?
- Eine stationäre Reihe hat einen konstanten Mittelwert und eine konstante Varianz über die Zeit.

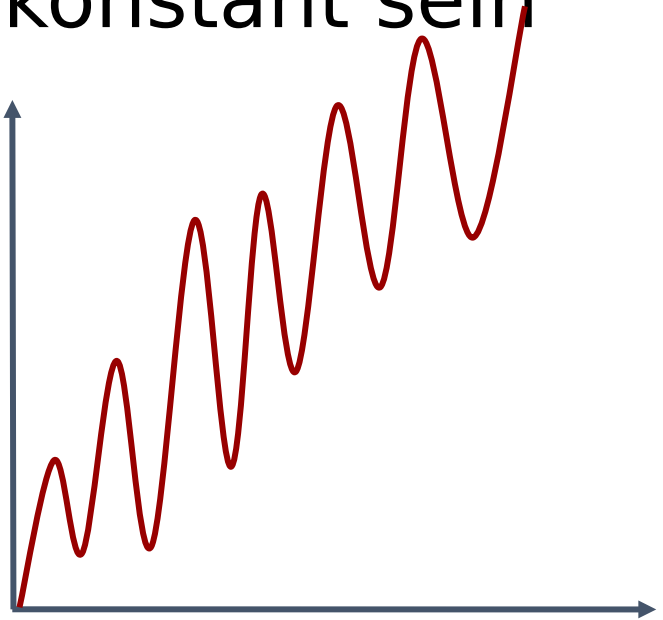
Stationär vs nicht stationär

- Ein stationärer Datensatz ermöglicht unserem Modell vorherzusagen, dass der Mittelwert und die Varianz in zukünftigen Perioden gleich sind.
- Schauen wir uns ein paar Beispiele an!

Stationär vs nicht stationär

- Der Mittelwert muss konstant sein

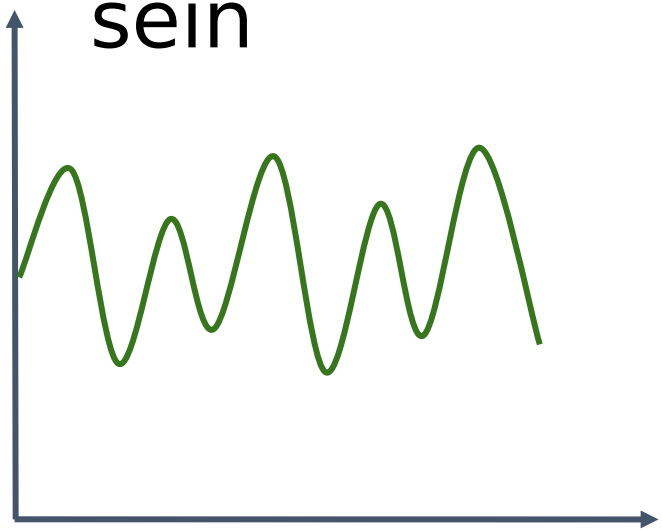
Stationär



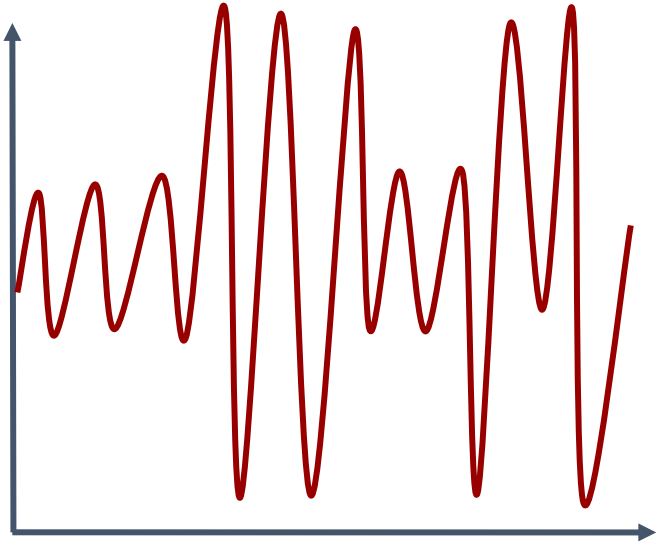
Nicht stationär

Stationär vs nicht stationär

- Varianz sollte keine Funktion der Zeit sein



Stationär

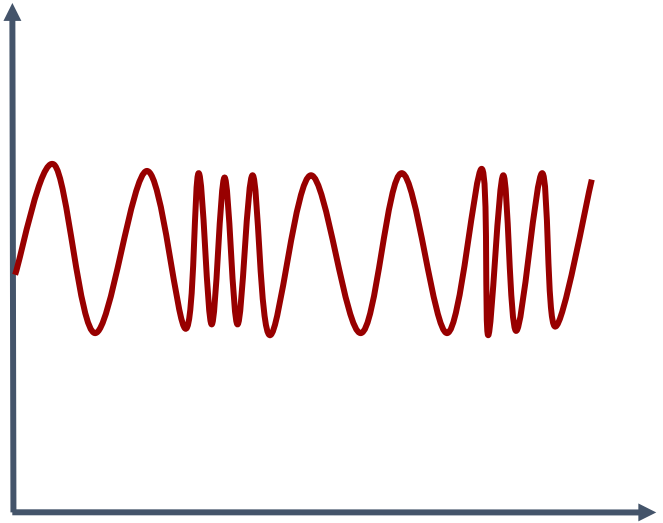


Nicht stationär

Stationär vs nicht stationär

Kovarianz sollte keine Funktion der Zeit sein

Stationär



Nicht stationär

Stationär vs nicht stationär

- Es gibt auch mathematische Tests, mit denen du die Stationarität in Deinen Daten testen kannst.
- Ein häufig verwendeter ist der Augmented Dickey-Fuller-Test (wir werden sehen, wie man das mit Pythons Statsmodels benutzt)

Stationär vs nicht stationär

- Wenn Du (entweder visuell oder mathematisch) festgestellt hast, dass Deine Daten nicht stationär sind, dann musst Du sie stationär machen. Erst danach kann man diese auszuwerten und entscheiden, welche Art von ARIMA-Termen du verwendest.

Differenzierung

- Ein einfacher Weg, dies zu tun, ist durch "Differenzierung".
- Die Idee hinter der Differenzierung ist ziemlich einfach, sehen wir uns ein Beispiel an ...

Differenzierung

Originale Daten

Time1	10
Time2	12
Time3	8
Time4	14
Time5	7

Erste Differenz

Time1	NA
Time2	2
Time3	-4
Time4	6
Time5	-7

Zweite Differenz

Time1	NA
Time2	NA
Time3	-6
Time4	10
Time5	-13

Differenzierung

- Wir können weiter differenzieren, bis wir Stationarität erreichen (was wir visuell und mathematisch überprüfen können)
- Jeder Differenzierungsschritt geht mit dem Verlust einer Datenzeile einher.

Differenzierung

- Für saisonale Daten kannst Du auch nach Saison unterscheiden.
- Wenn Du beispielsweise monatliche Daten mit jährlicher Saisonabhängigkeit hättest, kannst Du eine Differenz von 12 statt 1 angeben.

Differenzierung

- Eine andere übliche Technik mit saisonalen ARIMA-Modellen besteht darin, beide Methoden zu kombinieren, wobei der saisonale Unterschied der ersten Differenz betrachtet wird.

Autokorrelationsdiagramme

- Wenn Du nun stationäre Daten hast, ist es Zeit, zurückzugehen, um die p, d, q Ausdrücke zu besprechen und wie Du sie wählen solltest.
- Ein großer Teil davon sind Autokorrealtionsdiagramme und partielle Autokorrelationsdiagramme.
- Lass uns weiter zu ihrer Besprechung gehen!

Autokorrelationsdiagramme

Englisch: AutoCorrelation Plots

Übersicht

- Ein Autokorrelationsdiagramm (auch Correlogramm genannt) zeigt die Korrelation der Serie mit sich selbst, um x Zeiteinheiten verzögert.
- Daher ist die y-Achse die Korrelation und die x-Achse ist die Anzahl der Zeiteinheiten der Verzögerung.

Beispiel

- Wir werden diese Idee der Korrelation anhand eines einfachen Beispiels erklären.
- Beginnen wir damit, uns vorzustellen, wie man den Plot-Wert für $x = 1$ berechnet.

Beispiel

- Wir stellen uns vor, wir nehmen unsere Zeitreihe der Länge T, kopieren sie und löschen die erste Beobachtung der Kopie # 1 und die letzte Beobachtung der Kopie # 2.

Beispiel

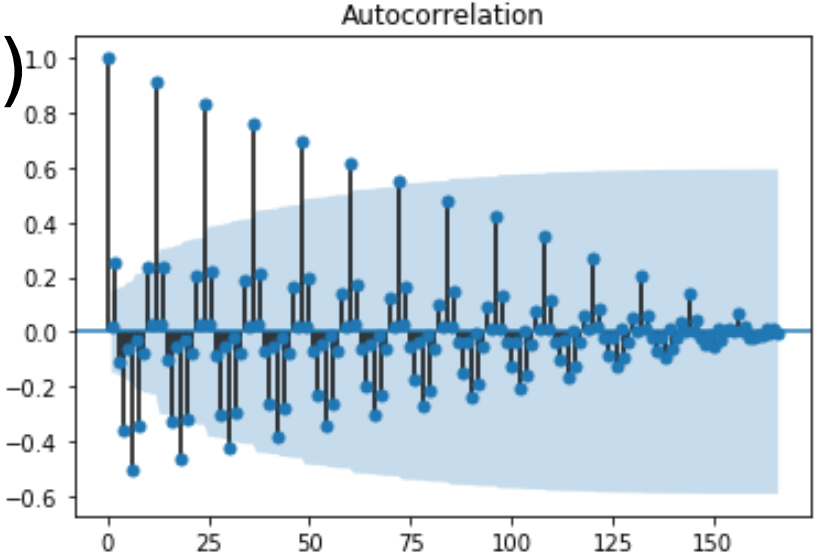
- Jetzt hast Du zwei Reihen der Länge T-1, für die Du einen Korrelationskoeffizienten berechnest.
- Dies ist der Wert der vertikalen Achse bei x = 1 in Deinen Diagrammen(Plots).

Beispiel

- Sie stellt die Korrelation der um eine Zeiteinheit verzögerten Reihe dar.
- So machst Du das nun weiter für alle möglichen Zeitverzögerungen x und dies definiert das Diagramm.
- Sehen wir uns ein paar typische Beispiele an!

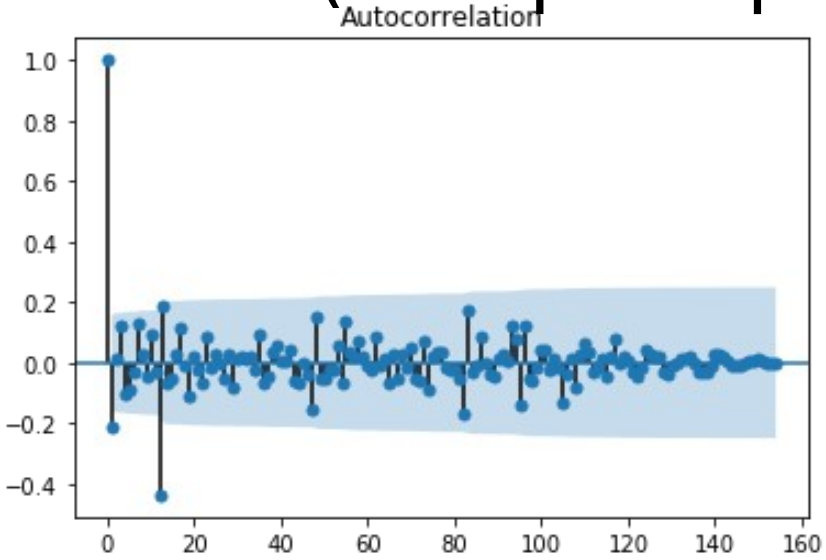
Beispiel

- Graduelle Abnahme (Gradual Decline)



Beispiel

- Scharfer Abfall (Sharp Drop-off)



Identifizierung des Modells

- Die tatsächliche Interpretation und der Bezug zu ARIMA-Modellen kann etwas kompliziert werden, aber es gibt einige grundlegende Methoden, die wir für das ARIMA-Modell verwenden können.

Identifizierung des Modells

- Unsere höchste Priorität besteht darin, herauszufinden, ob wir die AR- oder MA-Komponenten für das ARIMA-Modell (oder beide!) verwenden werden und wie viele Verzögerungen wir verwenden sollten.

Identifizierung des Modells

- Im Allgemeinen würden wir entweder AR oder MA verwenden; beides zusammen ist weniger üblich.
- Wenn wir tatsächlich die AR- und MA-Terme anwenden, werden wir die Werte von p oder q angeben.

Identifizierung des Modells

- Wenn das Autokorrelationsdiagramm eine positive Autokorrelation bei der ersten Verzögerung (Lag-1) zeigt, wird empfohlen, die AR-Terme in Bezug auf die Verzögerung zu verwenden.

Identifizierung des Modells

- Wenn das Autokorrelationsdiagramm bei der ersten Verzögerung eine negative Autokorrelation zeigt, wird empfohlen, MA-Terme zu verwenden.
- Dies erlaubt uns zu entscheiden, welche tatsächlichen Werte von p, d und q wir unserem ARIMA-Modell liefern.

Werte von p, d und q

- p: Die Anzahl der Lag-Beobachtungen, die im Modell enthalten sind.
- d: Die Häufigkeit, mit der die Roh-Beobachtungen differenziert werden
- q: Die Größe des Moving-Average-Fensters, auch Moving-Order genannt

Partielle Autokorrelationsdiagramme

- Es gibt auch partielle Autokorrelationsdiagramme!
- Diese sind etwas komplizierter als Autokorrelationsdiagramme, aber wir zeigen Dir die Grundlagen.

Partielle Autokorrelationsdiagramme

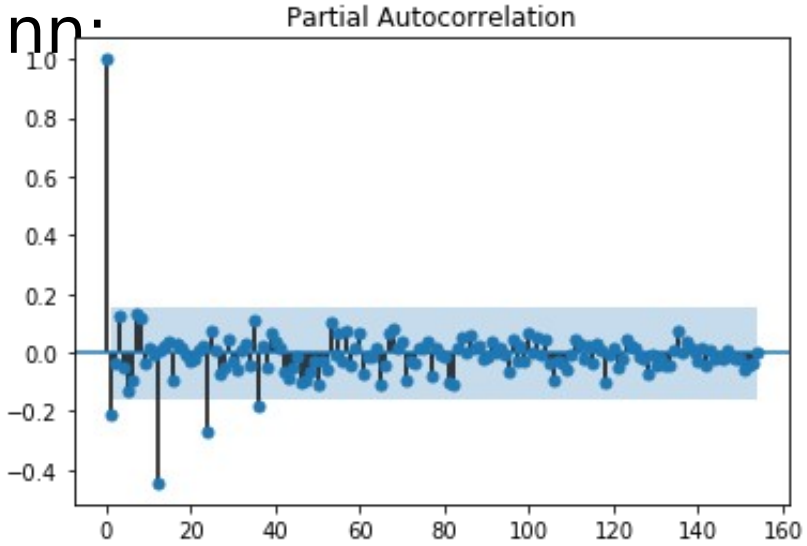
- Im Allgemeinen ist eine partielle Korrelation eine bedingte Korrelation.
- Es ist die Korrelation zwischen zwei Variablen unter der Annahme, dass wir die Werte einiger anderer Variablen kennen und berücksichtigen.

Partielle Autokorrelationsdiagramme

- Zum Beispiel betrachten wir einen Regressionskontext, in dem y Antwortvariable und x_1, x_2 und x_3 Prädiktorvariablen sind.
- Die Teilkorrelation zwischen y und x_3 ist die Korrelation zwischen den Variablen, berechnet unter Berücksichtigung, wie sowohl y als auch x_3 mit x_1 und x_2 zusammenhängen.

Partielle Autokorrelationsdiagramme

Hier ist ein Beispiel dafür, wie das Diagramm aussehen kann:



Identifizierung des Modells

- Typischerweise deutet ein scharfer Abfall nach der Verzögerung "k" darauf hin, dass ein AR-k-Modell verwendet werden sollte.
- Wenn es einen allmählichen Rückgang gibt, wird ein MA-Modell vorgeschlagen.

Identifizierung des Modells

- Die Identifizierung eines AR-Modells wird oft am besten mit dem PACF durchgeführt.
- Die Identifizierung eines MA-Modells wird oft am besten mit dem ACF und nicht mit dem PACF durchgeführt.
- Weitere Informationen findest du in den Links zu den Notebooks und Quellen.

Anwendung von ARIMA

- Sobald Du deine Daten mit ACF und PACF analysiert hast, kannst Du abhängig von deinen Originaldaten beginnen, ARIMA oder Saisonales ARIMA anzuwenden.
- Du gibst die p-, d- und q-Terme für das Modell an.

Anwendung von ARIMA

- Ein ARIMA wird dann drei Terme p, d und q annehmen. (Wir werden das im Codebeispiel sehen)
- Für saisonale ARIMA wird es einen zusätzlichen Satz von P-, D-, Q-Termen geben, die wir uns noch ansehen.

- Alles klar, jetzt ist es an der Zeit, all das in Aktion mit Python und Statistikmodellen zu sehen!
- Lass uns anfangen!

ARIMA-Code along

Teil 1: Statsmodel

Der grundlegende Prozess für ARIMA-Modelle ist wie folgt:

- Zeitreihendaten visualisieren
- Zeitreihendaten stationär machen
- Korrelations- und AutoKorrelations-Grafiken erstellen
- ARIMA-Modell konstruieren
- Das Modell verwenden, um Vorhersagen zu treffen

Wichtiger Hinweis!

- **Dies ist ein kompliziertes Thema! Denk dran, dieses Notebook ist optional und um es vollständig zu verstehen solltest Du die ergänzenden Quellen in den Links lesen und Dir das komplette Erklärungsvideo anschauen. Dieses Notebook und die Videovorlesungen dienen nicht dazu, ARIMA vollständig zu verstehen, sondern sind eine Übersicht über deren Anwendungen, damit Du später verstehen kannst, warum es (k)eine gute Idee sein könnte, ARIMA für Aktiendaten zu verwenden.**

ARIMA-Code along

Teil 2: Stationarität

ARIMA-Code along

Teil 3: Autokorrelationsdiagramme

ARIMA-Code along

Teil 4: Saisonale ARIMA-Modell