Quantopian für Fortgeschrittene

Python for Finance

- Nun, da wir die Grundlagen Algorithmischen Handels mit der Quantopian Plattform verstehen, machen wir weiter mit ein paar weiterführenden Themen!

Python for Finance

- Wir werden die folgenden Themen besprechen
  - Leverage
  - Hedging
  - PyFolio zur Portfolio Analyse
  - Aktienstimmung Handelsstrategie
  - Termingeschäfte

Lass uns anfangen!

Leverage

Python for Finance

- Leverage ist die Fähigkeit, sich Geld zu leihen, um es zu investieren.
- Ein einfaches Beispiel von Leverage wäre das Aufnehmen einer Hypothek, wenn man Grundbesitz kauft.

Python for Finance

- Für algorithmisches Handeln bedeutet Leverage, Schulden erneut zu investieren, um eine größere Rendite aus unseren Investitionen zu erhalten.
- Dies würde man nur machen, wenn man sich seiner Strategie sehr sicher ist und glaubt, dass sie ein niedriges Risiko besitzt.

Python for Finance

- In der echten Welt würde Leverage von deinem Makler angeboten werden und würde stark von deiner Strategie, deinem Kontostand, früheren Zahlungen, Gebühren etc. abhängen.

Python for Finance

- Typischerweise können wir Leverage anhand eines Quotienten messen:
  - Leverage Quotient = (Schulden + Grundwert) / Grundwert
  - Die Summe Deiner Schulden und Grundkapital geteilt durch Dein Grundkapital.

Python for Finance

- Schauen wir uns Leverage in Quantopian an.
- Wir werden sowohl mit dem Research-Notebook als auch mit der IDE arbeiten, also öffne zwei Tabs in Deinem Browser!
- Stelle sicher, dass Du Dich für den hier verwendeten Code auf das Notebook beziehst.

Hedging

Python for Finance

- In dieser Lektion werden wir Hedging etwas weiter erkunden.
- Vieles von dem, was wir hier diskutieren, steht in Verbindung zum CAPM, also schau Dir am besten diese Lektionen noch einmal an!

Python for Finance

- Erinnerung: Beta repräsentiert, wie stark ein Vermögen dem Markt ausgesetzt ist (typischerweise dargestellt als S&P500).
- Handelsstrategien mit niedrigem Beta sind attraktiv, da sie unabhängig vom Markt operieren sollten.

Python for Finance

- In diesem Notebook werden wir Beta und Alpha eines Vermögens berechnen und dann eine einfache Methode zum “Hedging” gegen den Beta demonstrieren, als ein Versuch alle Risikopositionen des Marktes zu eliminieren.

Python for Finance

- Wir wählen eine Aktie wie AAPL und besorgen uns deren Alpha und Beta.
- Dann können wir eine short-Position auf SPY berechnen, um zu verhindern, dass sie weiter dem Markt ausgesetzt ist und handeln nur auf dem Alpha des Vermögens.

Lass uns anfangen!

Hedging
(Teil 2)

Hedging
(Teil 3)

Portfolioanalyse Mit PyFolio

Python for Finance

- Quantopian hat eine eingebaute Bibliothek, PyFolio (ebenfalls open source), die schnell und einfach “tear-sheets” und Diagramme mit nützlichen Informationen erstellt.
- Schauen wir uns das an!

Aktienstimmungs-Analyse

(Sentimentanalyse)

Python for Finance

- Wir haben bisher nur numerische Daten als Handelssignal betrachtet, aber es gibt noch sehr viel mehr Informationen auf der Welt!
- Eine übliche Signalquelle ist die Stimmungsanalyse.

Python for Finance

- Stimmungsanalyse verwendet die Verarbeitung natürlicher Sprache, um die Stimmung in einem Text herauszufinden, z.B.:
  - Das ist schlecht! Es wird scheitern!
    - Negative Stimmung
  - Großartig! Wir haben einen Gewinner.
    - Positive Stimmung

Python for Finance

- Die eigentliche Funktionsweise hinter den Stimmunganalyse-Algorithmen übersteigt die Ausmaße dieses Kurses, aber zum Glück kannst Du einfach neg/pos Stimmungswerte (-1.0 bis 1.0) aus einer Vielzahl an Quellen auf Quantopian erhalten!

Python for Finance

- Diese Art von Daten gibt es fast nie umsonst.
- Quantopian verfügt über eine kostenlose Version der Stimmungsanalyse (verfügbar nur für die Jahre 2013-2019 weiterführend).
- Sentdex bietet einen Stimmungswert von -3 bis 6

Python for Finance

- Die Basis für die Sentiment-Werte sind 20 Nachrichtenquellen wie CNBC, WSJ, Yahoo, etc.
- In den folgenden Lektionen werden wir Dir zeigen, wie man auf diese kostenlosen Daten zugreift und eine Handelsstrategie mit dir erstellen!

Python for Finance

- Unsere Hauptstrategie wird darin bestehen, Aktien zu kaufen, wenn der Einflusswert niedrig ist und zu verkaufen, wenn er hoch ist.
- Wir werden auch in anderem Zusammenhang mit Pipelines arbeiten!
- Lass uns anfangen!

Einführung in Termingeschäfte

(Futures)

Python for Finance

- Bevor wir uns näher mit dem Handeln von Termingeschäften auf Quantopian beschäftigen schauen wir uns zunächst einmal an, was Termingeschäfte eigentlich sind!

Python for Finance

- Vielleicht hast Du schon einmal den Ausdruck “Derivate” gehört. Termingeschäfte sind ein Beispiel für einfache Derivate.

Python for Finance

- Ein Derivat ist ein Vertrag zwischen zwei oder mehr Parteien, dessen Wert auf einem zugrundeliegenden Finanzposten, Index oder Wertpapier basiert, auf welches sich zuvor geeinigt wurde. Häufige zugrundeliegende Instrumente: Anleihen, Rohstoffe, Währungen, Zinssätze, Marktindizes und Aktien.

Python for Finance

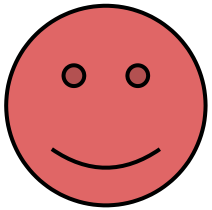
- Derivate können extrem komplex sein, aber zum Glück sind Terminkontrakte relativ einfache Derivate.
- Bevor wir Terminkontrakte diskutieren, lass uns über ein sogenanntes “individuelles Termingeschäft (Forwards) ” (“Forward Contract”) sprechen.

Python for Finance

- Forwards sind eine Einigung zwischen zwei Parteien, einen Lieferpreis **K** für einen Vermögenswert zu einem späteren Zeitpunkt zu zahlen.
- Diejenige Person, die den Vermögenswert verkauft, befindet sich in der **short**-Position, die kaufende Person in der **long-**Position.

Python for Finance

- Sehen wir uns ein einfaches Beispiel an!

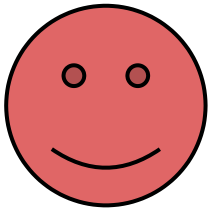


SANDY

BILLY

Python for Finance

- Wir haben zwei Personen, Sandy und Billy.

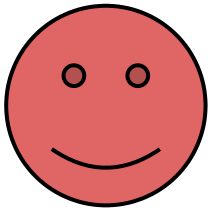


SANDY

BILLY

Python for Finance

- Billy betreibt eine Tankstelle und braucht Benzin.

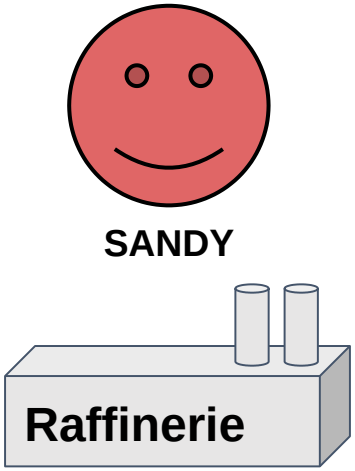


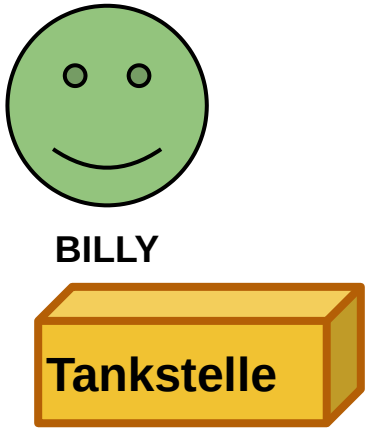
SANDY

BILLY

Python for Finance

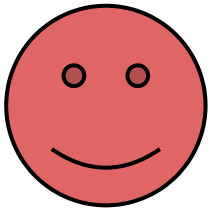
- Sandy produziert Benzin in ihrer Raffinerie.



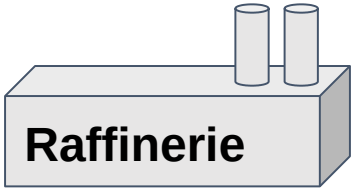


Python for Finance

- Billy weiß, dass er in einem Jahr Benzin brauchen wird, aber aber die Ölproduktion fluktuiert.
- Billy möchte einen Weg finden, um einen garantierten Preis für ein Jahr später zu erhalten.



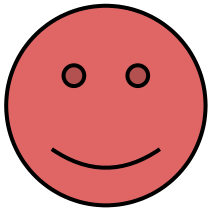
SANDY



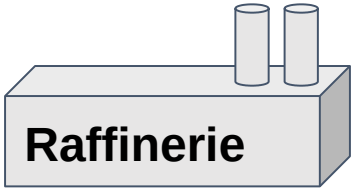
BILLY

Python for Finance

- Sandy ist ebenfalls denselben Schwankungen ausgesetzt, daher sucht sie einen Weg, um die Volatilität in ihrer Preisgebung für das nächste Jahr zu reduzieren.



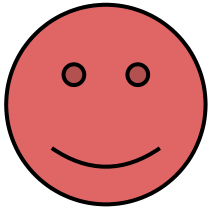
SANDY



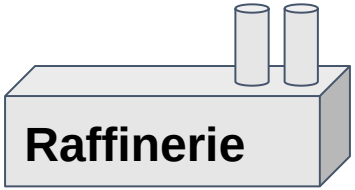
BILLY

Python for Finance

- Hier kommt das individuelle Termingeschäft ins Spiel.
- Billy stimmt zu, Sandy in einem Jahr einen festen Betrag für ein Fass Benzin zu bezahlen.



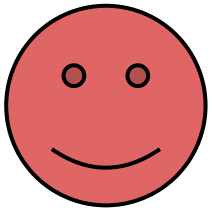
SANDY



BILLY

Python for Finance

- Billy hat nun die long-Position inne, Sandy die short-Position.
- Wer profitiert also von diesem Geschäftsvorgang?

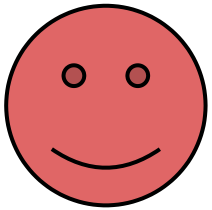


SANDY

BILLY

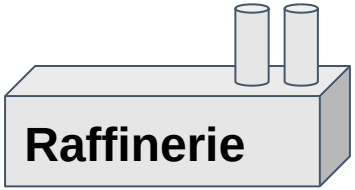
Python for Finance

- Das hängt vom zukünftigen Benzinpreis ab.
- Gehen wir die zwei möglichen Szenarien durch!



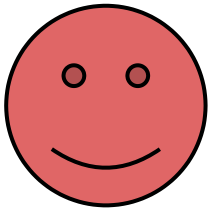
SANDY

BILLY



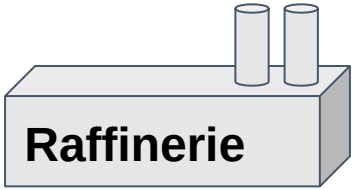
Szenario 1: Marktpreis höher in Zukunft

- Billy wird in einem Jahr 100€ für ein Fass Benzin zahlen. Sandy ist einverstanden.



SANDY

BILLY

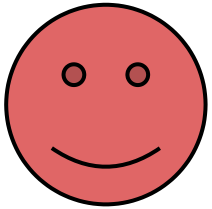


Billy muss Sandy in einem Jahr €100 für ein Fass Benzin zahlen!

Szenario 1: Marktpreis höher in Zukunft

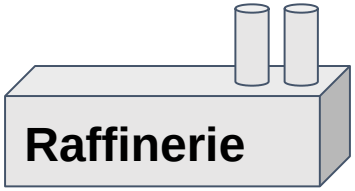
Markt

- Ein Jahr vergeht und ein Fass Benzin kostet in Wirklichkeit 150€!



SANDY

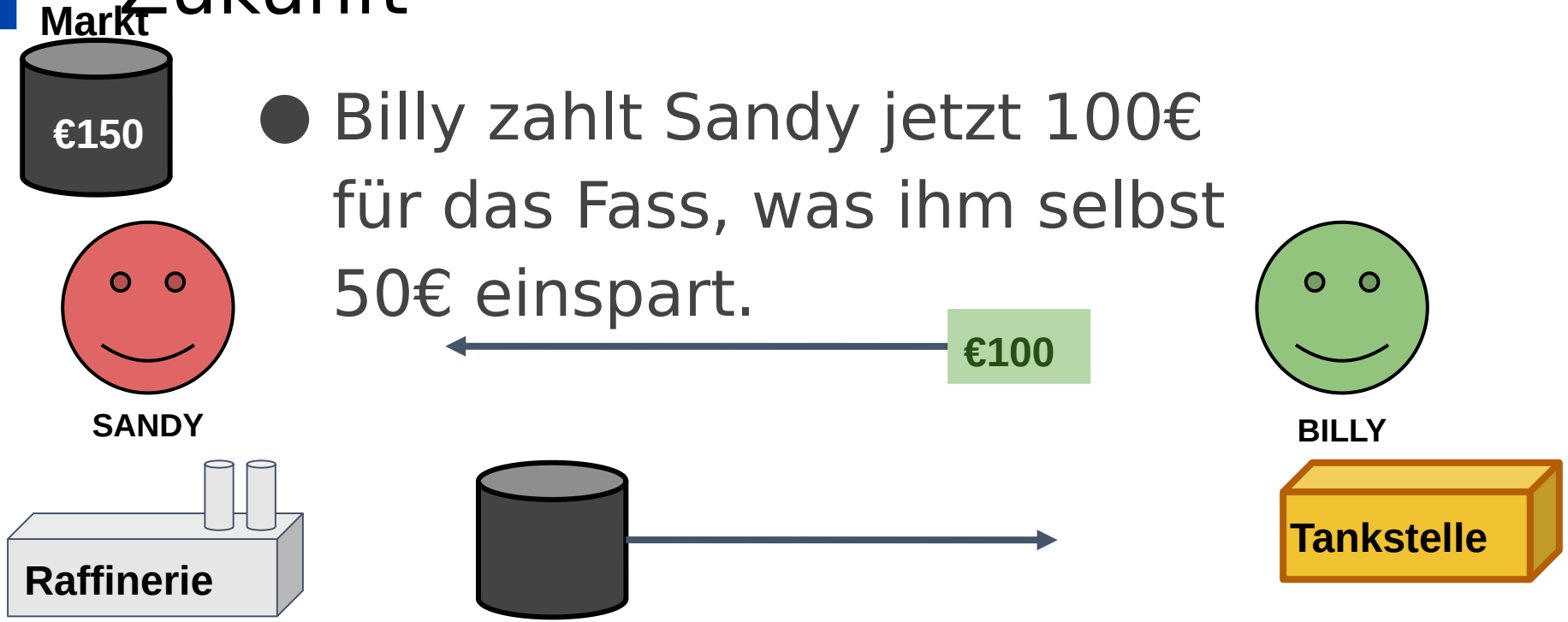
BILLY



Billy muss Sandy in einem Jahr €100 für ein Fass Benzin zahlen!

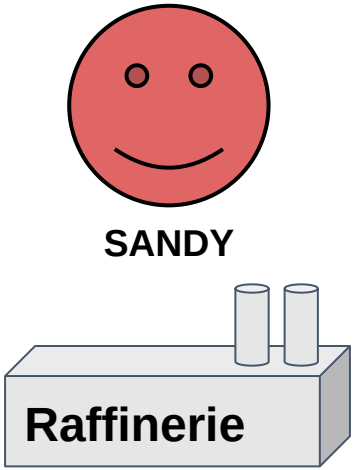
Szenario 1: Marktpreis höher in Zukunft

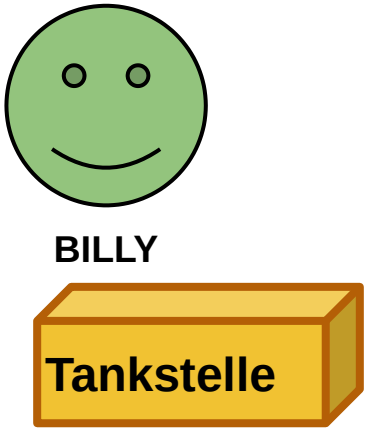
- Billy zahlt Sandy jetzt 100€ für das Fass, was ihm selbst 50€ einspart.



Python for Finance

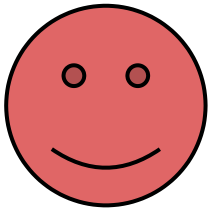
- Im letzten Fall ging der Marktpreis hoch, was bedeutet, dass Sandy potenziellen Profit verloren hat.





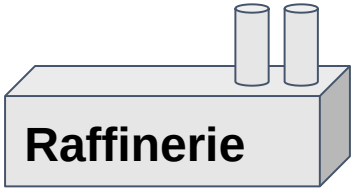
Python for Finance

- Aber was wäre, wenn der zukünftige Preis niedriger wäre als der Betrag des Forwards?



SANDY

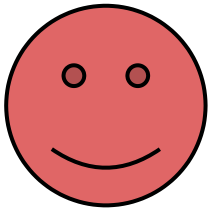
BILLY



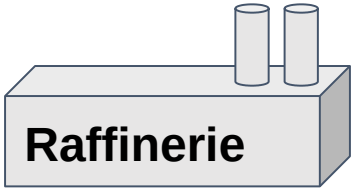
Szenario 2: Marktpreis niedriger in Zukunft

- Billy wird Sandy in einem Jahr 100€ für ein Fass Benzin zahlen.

  Sandy ist einverstanden.



SANDY



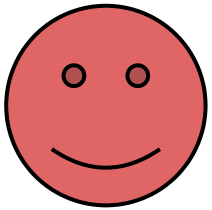
Billy muss Sandy in einem Jahr €100 für ein Fass Benzin zahlen!

BILLY

Szenario 2: Marktpreis niedriger in Zukunft

Markt

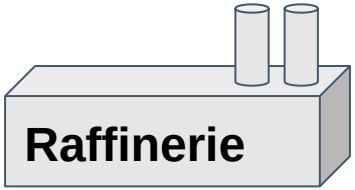
- Ein Jahr geht vorbei und der Marktpreis für Benzin ist auf 25€ pro Fass gefallen!



SANDY

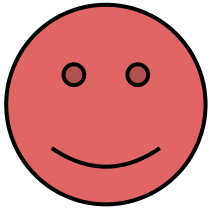
BILLY

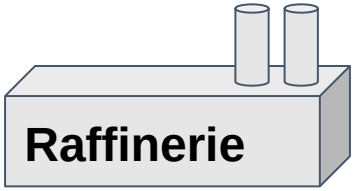
Billy muss Sandy in einem Jahr €100 für ein Fass Benzin zahlen!



Szenario 2: Marktpreis niedriger in Zukunft

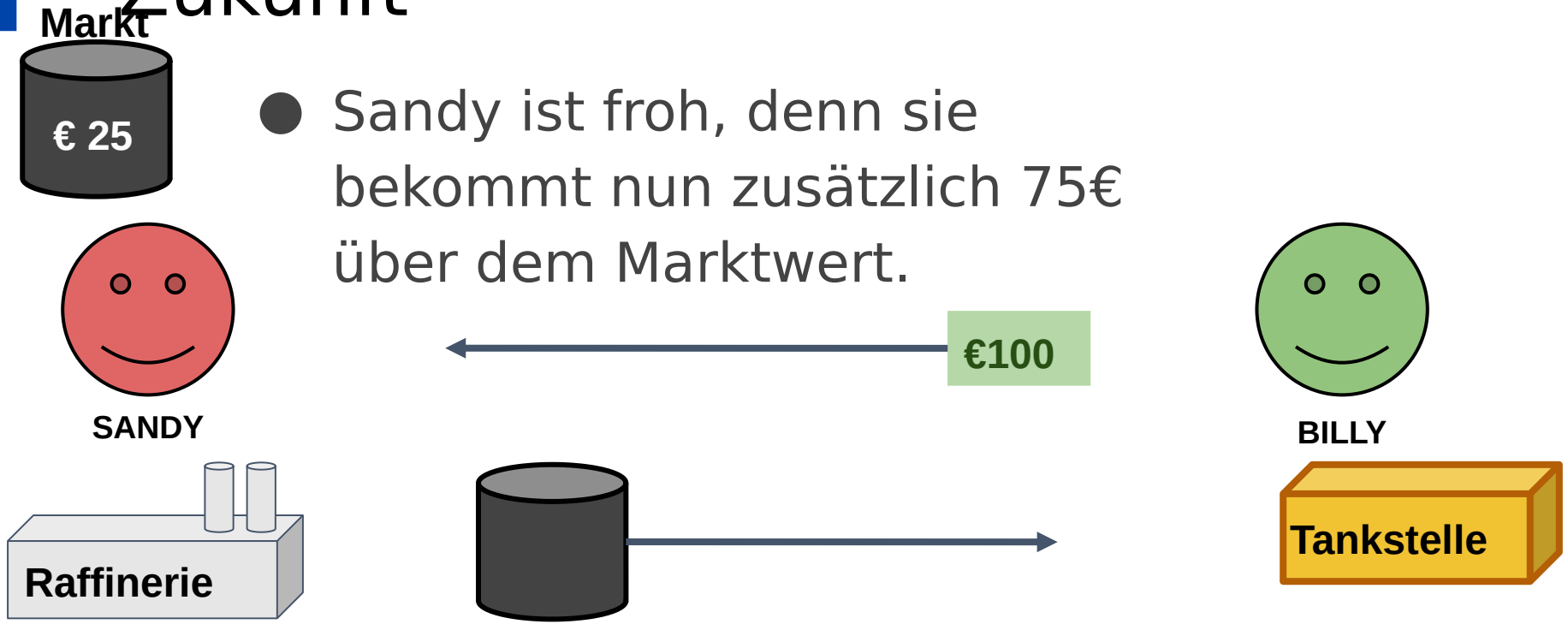
- Billy muss Sandy jetzt immer noch 100€ pro Fass zahlen! Ungeachtet des Marktes!





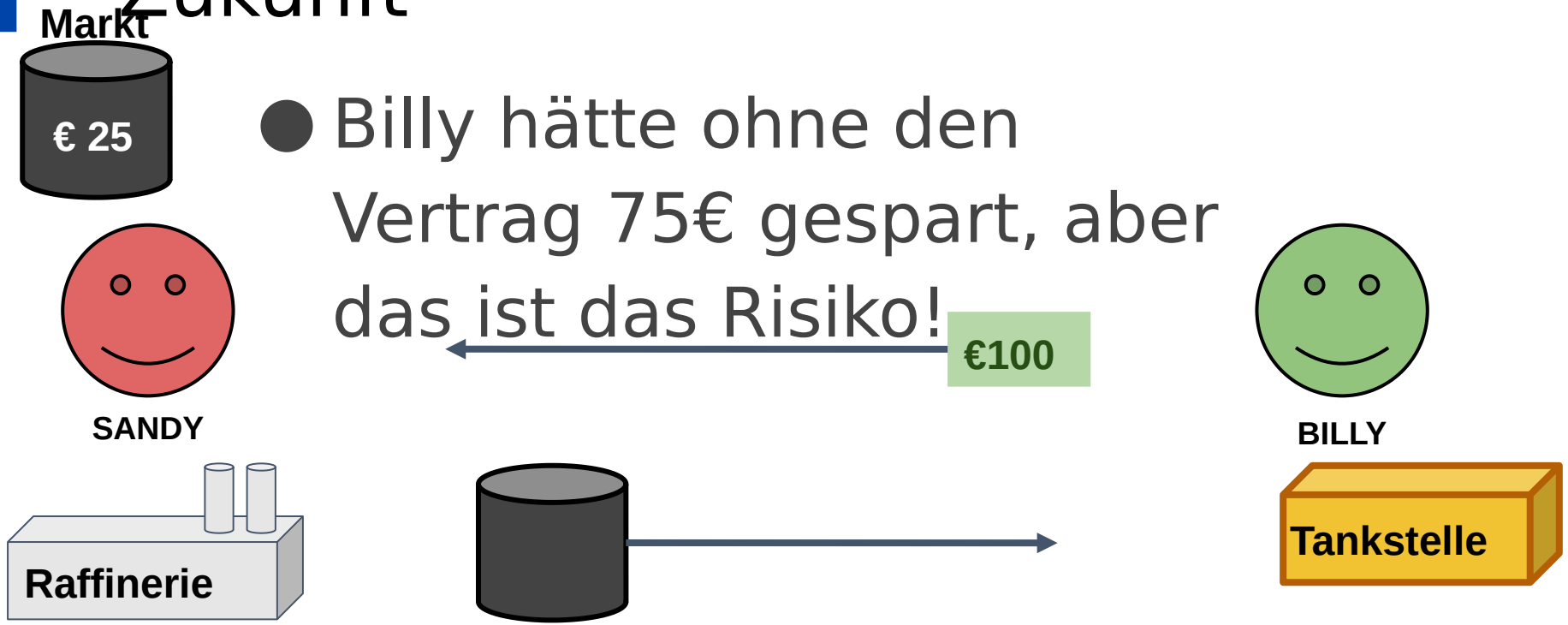
Szenario 2: Marktpreis niedriger in Zukunft

- Sandy ist froh, denn sie bekommt nun zusätzlich 75€ über dem Marktwert.



Szenario 2: Marktpreis niedriger in Zukunft

- Billy hätte ohne den Vertrag 75€ gespart, aber das ist das Risiko!



Python for Finance

- Warum sollten also Billy und Sandy einem Forward überhaupt zustimmen?
- Es reduziert das Risiko auf beiden Seiten, da beide Parteien wissen, wie hoch der Preis in Zukunft sein wird.

Python for Finance

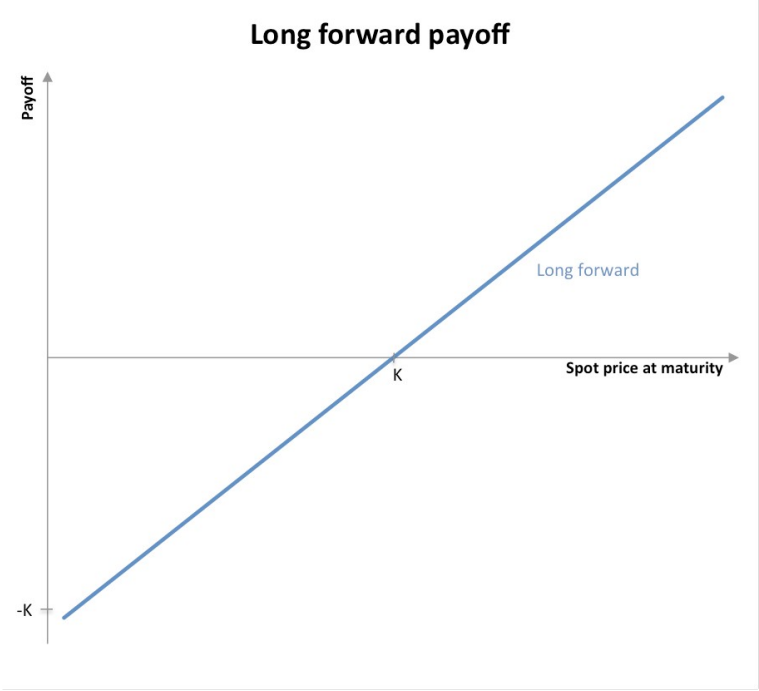
- Offensichtlich ist die Berechnung des zukünftigen Preises entscheidend; beide Parteien haben den Anreiz, für sich selbst den besten Deal zu bekommen!
- Wir können den “Ausgleichsbetrag” für jeden Teil formal mathematisch beschreiben.

Python for Finance

- Wenn wir sagen, **K** ist der vorherbestimmte zukünftige Preis und $\mathbf{S_T}$ ist der wahre Marktpreis zum Zeitpunkt T, können wir den Ausgleichsbetrag sowohl für die long-Partei als auch für die short-Partei definieren.

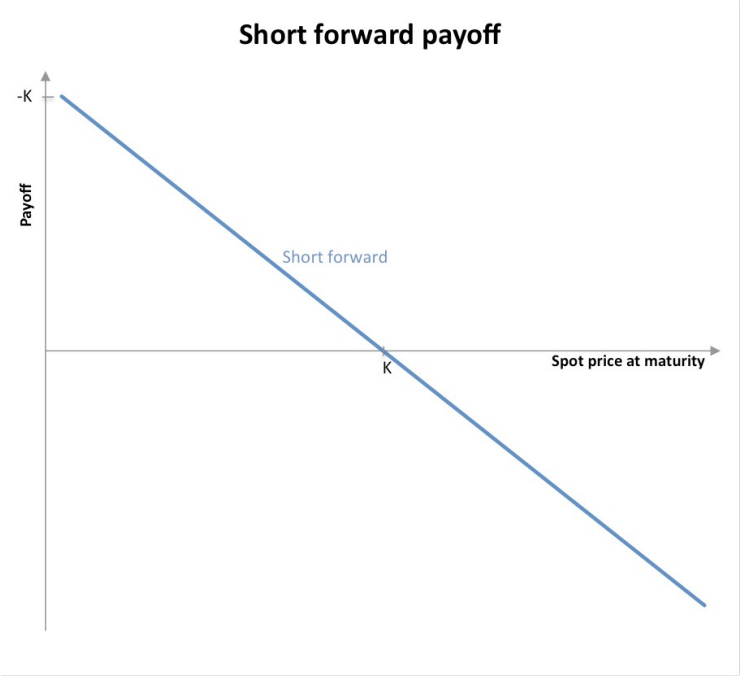
Python for Finance

- Der Ausgleichbetrag für die long-Position ist:
  - $\mathbf{S_T - K}$



Python for Finance

- Der Ausgleichsbetrag für die short-Position ist:
  - **K - S_T**



Python for Finance

- Auch wenn individuelle Termingeschäfte (Forwards) einfach sind, gibt es ein paar Probleme mit ihnen:
  - Kontrahentenausfallrisiko (Counterparty Risk)
  - Mangel an Liquidität
  - Es müssen viele individuelle Vertragsbedingungen verhandelt werden.

Python for Finance

- Hier kommen Terminkontrakte (Futures) ins Spiel!
- Terminkontrakte sind Forwards, die für den Handel an der Börse standardisiert wurden.

Python for Finance

- Ein einzelner Terminkontrakt gilt für eine bestimmte Menge des Basiswerts mit gemeinsam beschlossener Zahlung, Lieferdatum und Bedingungen.
- Zusätzlich wirkt die Börse als Vermittler, der das Kontrahentenausfallrisiko virtuell eliminiert.

Python for Finance

- Die grundlegende Idee eines Futures im Vergleich zu einem Forward ist die Einführung einer dritten Partei.
- Futures sind für einen bestimmten Vermögenswert standardisiert, das heißt die Bedingungen für Mais-Futures können sich von den Bedingungen für Schweinebauch- Futures unterscheiden.

Python for Finance

- Die Existenz von Terminkontrakten führt zur Spekulation und Hedging mit Futures.
- Spekulation tritt auf, wenn Marktteilnehmer Preisänderungen des zugrunde liegenden Vermögenswerts vorhersagen.

Python for Finance

- Eine letzte Eigenschaft von Futures!
- Sie werden täglich mit einem Einschusskonto bei einem Makler festgelegt, welches von dem Inhaber des Terminkontrakts geführt wird.

Python for Finance

- Jeden Tag spiegelt sich die Preisänderung des Basiswerts in einem An- oder Abstieg der Menge des Geldes auf dem Einschusskonto wider.
- Dieser Prozess wird Marktpreisbewertung ("marking to market") genannt.

Python for Finance

- Nun, da wir etwas darüber gelernt haben, was Futures sind, schauen wir uns an, wie sie auf Quantopian funktionieren!
- Beachte, dass dies ein fortgeschrittenes Thema ist und es noch viel mehr zu Futures gibt, das wir nicht in diesen Lektionen behandeln!

Termingeschäfte mit Quantopian

(Teil 1)

Termingeschäfte mit Quantopian

(Teil 2)

Termingeschäfte mit Quantopian

(Teil 3)