

$$1) X \sim N(9, 3), Y \sim N(5, 3)$$

$$Z \sim N(13, 2), W \sim N(6, 1.5)$$

Da X, Y, Z, W i.i.d. gilt demnach:

$$A = Z + W + X + Y$$

$$A \sim N(9 + 5 + 13 + 6, 3^2 + 3^2 + 2 + 1.5^2)$$

$$\Leftrightarrow A \sim N(33, 24.25)$$

$$2) E(\overset{A}{\cancel{Z}}) = E(X + Y + Z + W) = E(X) + E(Y) + E(Z) + E(W)$$

$$= 9 + 5 + 13 + 6 = 33$$

$$3) \text{Var}(X + Y + W + Z) \stackrel{\text{i.i.d.}}{=} \text{Var}(X) + \text{Var}(Y) + \text{Var}(Z) + \text{Var}(W)$$

$$= 24.25$$

$$\sigma = \sqrt{24.25}$$

4) Sei $X \sim N(\mu, \sigma^2)$

Es gilt $P(a \leq X \leq b) = \Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)$

$$\Leftrightarrow P(30 \leq A \leq 40) = \Phi\left(\frac{40-33}{\sqrt{24,25}}\right) - \Phi\left(\frac{30-33}{\sqrt{24,25}}\right)$$

$$= \Phi\left(\frac{7}{\sqrt{24,25}}\right) - \Phi\left(\frac{-3}{\sqrt{24,25}}\right)$$

Nach $\Phi\left(\frac{-3}{\sqrt{24,25}}\right) = 1 - \Phi\left(\frac{3}{\sqrt{24,25}}\right)$

$$\Rightarrow \Phi\left(\frac{7}{\sqrt{24,25}}\right) + \Phi\left(\frac{3}{\sqrt{24,25}}\right) - 1$$

$$= \Phi(1,42) + \Phi(0,60) - 1$$

$$= 0,92220 + 0,72907 - 1$$

$$= 0,65127 \approx 65,13\% \quad \square$$

2.1)

Zeilen: 12

Spalten: 4

2.2)

<f8 bedeutet soviel wie float64, das heißt man kann Dezimalwerte lagern und verwendet 32 bit für jede Nummer.

<U12 bedeutet soviel wie Unicode, 12 steht für die Anzahl an Characters.

2.3)

<class 'numpy.ndarray'>

<class 'numpy.recarray'>

a ist ein ndarray, das heißt in diesem Array liegen ausschließlich Elemente eines einzigen Typs und in dem Falle wäre dies der float Datentyp.

b ist ein Array, welches Felder enthält, analog zu Spalten in einem Arbeitsblatt. Jede Eintrag kann sowohl integer als auch floats oder strings beinhalten. Das bedeutet die Spalten haben des Arrays haben demnach unterschiedliche Typen. Man kann auf die Werte mit arr.y zugreifen anstelle von arr['x'].

2.4)

möchte man Aggregieren über mathematische Operationen und sollte man eventuell ndarray verwenden. Es gibt aber kein wirkliches schwarz weiß -verfahren. Es hängt speziell von den Daten ab

2.5)

[7. 3.8 4.9 1.5]

bei a wird für jede Spalte ein Maximum bestimmt, d.h. man geht komplett die eine Spalte von oben nach unten durch und bestimmt für Spalte 1 den Max. Analog macht man dies für alle weiteren Spalten. Man erhält letztendlich einen Vektor mit allen Maxima spaltenweise

2.6)

Vermutlich liegt es daran weil max eine mathematische Operation ist aber die letzte Spalte von b ist eine Spalte mit String Elementen, wie soll man dafür den max bestimmen? Die vorherigen Spalten sind nicht kompatibel mit der letzten Spalte

2.7)

[6 0 8 7]

3)

```
n = 10
xs = np.random.rand(n)
type(xs)

def transform_nobbranch(xs):
    return (xs < 0.5).astype(int)*(2*xs-1) + (xs >= 0.5).astype(int)*np.round((3*xs+0.5))

ys = transform_nobbranch(xs)
```

Argument1: Zeitintensiv, man sollte nicht versuchen jetzt jede Methode zu vektorisieren, da dies Zeit in Anspruch nimmt und das Projekt wichtiger ist. -> Es ist ein Trade-Off

Argument2: Da man mehrere Prozessoren verwendet, steigen die Kosten und der Verbrauch der CPU, nicht jeder Computer kann sich das leisten

4)

```
from scipy.stats import nbinom
prob = 1 - nbinom.cdf(3,2,0.5) #3 Misserfolge, 2 Erfolge und 1/2 als p für ein Bernoulli-Versuch
print(prob)
```

prob=0.1875