

# **Elektronisches Prüfen mit Jupyter Notebook und JupyterHub**

**Vor Ort, Online oder Hybrid**

Prof. Dr. Paul G. Plöger / M.Sc. Tim Metzler / M.Sc. Mohammad Wasil

Fachbereich Informatik

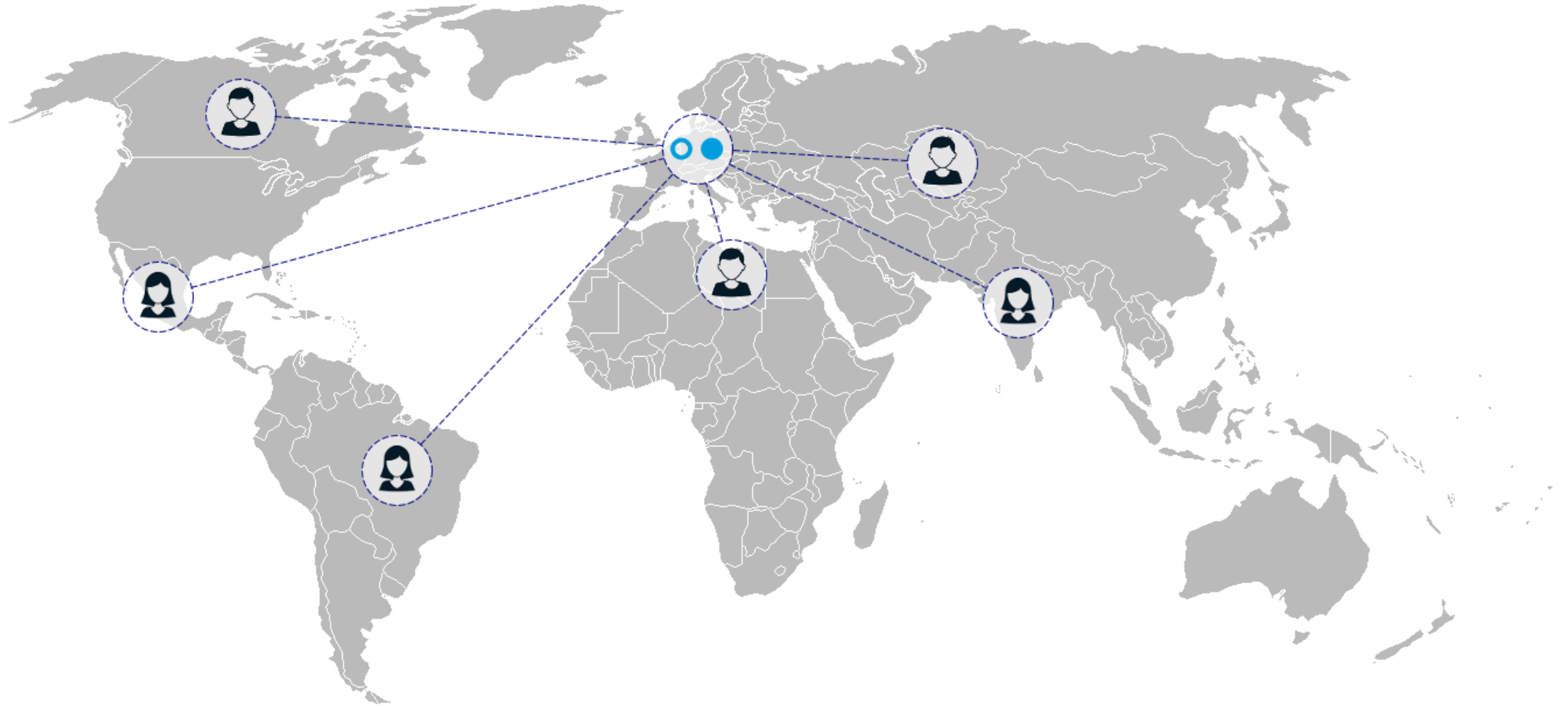
Hochschule Bonn-Rhein-Sieg



# Wer sind wir?

- Fachbereich Informatik
- Prof. Dr. Paul G. Plöger:  
Seit 2003 Professor im internationalen Masterstudiengang „Autonome Systeme“ (MAS)  
Lehre im Bereich Mathematik / Machine Learning
- M.Sc. Tim Metzler:  
Seit 2019 wissenschaftlicher Mitarbeiter im E-Assessment Projekt  
Unterstützung bei der Lehre (u.a Natural Language Processing, Neural Networks)  
Entwicklung von Software zur Bewertung von Klausuren und Übungen
- M.Sc. Mohammad Wasil:  
Seit 2019 wissenschaftlicher Mitarbeiter im E-Assessment Projekt  
Entwicklung und Wartung der Infrastruktur

# Klausuren von überall



Hochschule  
Bonn-Rhein-Sieg

Fachbereich  
Informatik

Prof. Dr.  
Paul G. Plöger

M.Sc. Tim Metzler

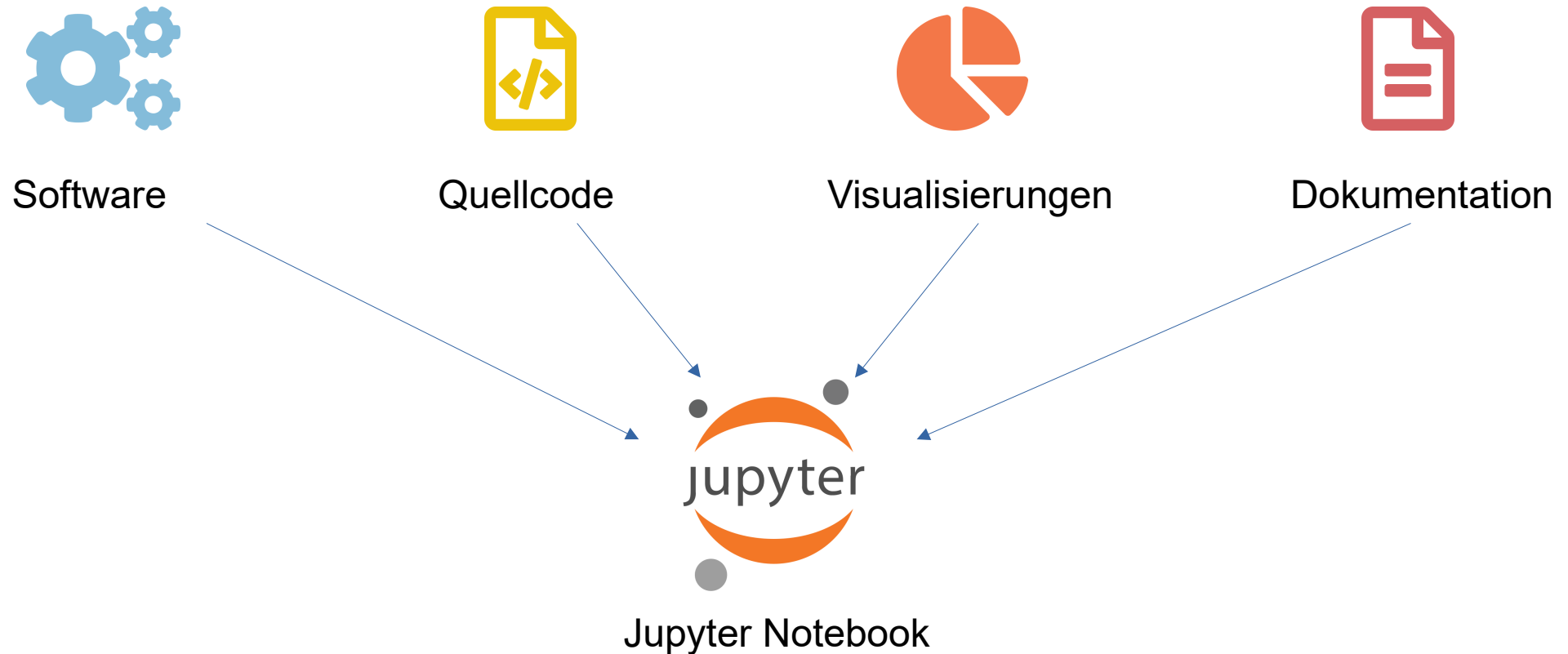
M.Sc. Mohammad Wasil



# Warum Jupyter Notebooks?



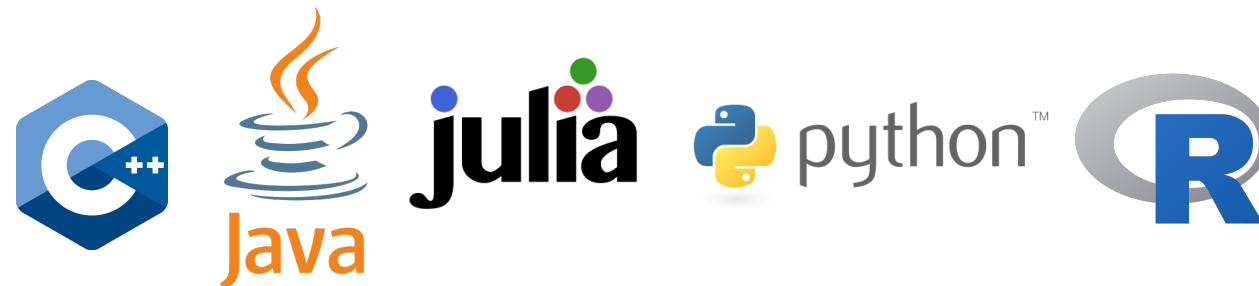
# Artefakte in der Informatik



# Jupyter Notebook (I/II)



- Werden mit der web-basierten Software Jupyter Notebook Server erstellt und bearbeitet
- Wird im Webbrowser ausgeführt
- Bestehen aus einer Liste von Zellen
- Zellen enthalten entweder Text, Bilder, PDFs, Code oder HTML
- Lauffähige Programmierumgebung (Studierende können Code ausführen)
- Kompatibel mit einer Vielzahl an Programmiersprachen



# Jupyter Notebook (II/II)



- Jupyter Notebooks können einfach erweitert werden
- Neue Zelltypen können erstellt werden (bspw. Webcam Zelle, Dateiupload)
- Es existiert eine Vielzahl an Plugins, bspw. Nbgrader<sup>1</sup> zur Bewertung von Jupyter Notebooks

<sup>1</sup><https://github.com/nbgrader>



## Verteilung [50 Punkte]

Zusätzlich zu diesem Notebook wurde Ihnen die Datei `health_cleaned.csv` zur Verfügung gestellt.

Dieser Datensatz enthält Umfrageergebnisse einer [Gesundheitsumfrage](#) unter Jugendlichen und jungen Erwachsenen aus den USA .

Im Rahmen der National Longitudinal Study of Adolescent to Adult Health wurden 20000 Teilnehmer zu verschiedensten Themen rund um die Gesundheit befragt. Der Datensatz `health_cleaned.csv` enthält einen Auszug der Fragen und Antworten von ca. 5000 Teilnehmern.

Der Datensatz enthält folgende Werte:

- AID - Pseudonym für jeden Teilnehmer
- Gewicht - Gewicht der Teilnehmer
- Groesse - Größe der Teilnehmer
- Geschlecht - Geschlecht der Teilnehmer
- Geburtsjahr - Das Geburtsjahr der Teilnehmer

## Verteilung a) [2 Punkte]

Laden Sie die Daten in einen pandas *DataFrame* mit dem Namen `health` .

Teilen Sie die Daten nach Geschlecht auf und nennen Sie die neuen *DataFrames* `men` und `women` .

```
In [3]: import pandas as pd
        ### BEGIN SOLUTION
        health = pd.read_csv('data/health_cleaned.csv')
        men = health.loc[health.Geschlecht == 'Male']
        women = health.loc[health.Geschlecht == 'Female']
        ### END SOLUTION

        health.head(n=1000)
```



### Verteilung a) [2 Punkte]

Laden Sie die Daten in einen pandas *DataFrame* mit dem Namen `health`.

Teilen Sie die Daten nach Geschlecht auf und nennen Sie die neuen *DataFrames* `men` und `women`.

```
In [3]: import pandas as pd
### BEGIN SOLUTION
health = pd.read_csv('data/health_cleaned.csv')
men = health.loc[health.Geschlecht == 'Male']
women = health.loc[health.Geschlecht == 'Female']
### END SOLUTION

health.head(n=1000)
```

Out [3]: Show 10 entries Search:

AID	Gewicht	Groesse	Geschlecht	Geburtsjahr
57101310	113.9	180	Female	1976
57103869	107.8	202	Male	1976
57109625	68	161	Male	1981
57111071	89.4	177	Male	1981
57113943	150.6	185.5	Male	1979
57117542	96	185	Male	1979
57118381	52.6	169	Female	1982
57118943	71.3	163.5	Female	1979
57120005	95.1	184	Male	1982
57120046	108	178	Male	1976

Showing 1 to 10 of 1,000 entries Previous 1 2 3 4 5 ... 100 Next



### Verteilung b) [12 Punkte]

Bestimmen Sie die charakteristischen Parameter für das Merkmal **Gewicht** der Männer.

Plotten Sie ein Histogramm (mit aktiviertem Parameter `density=True`).

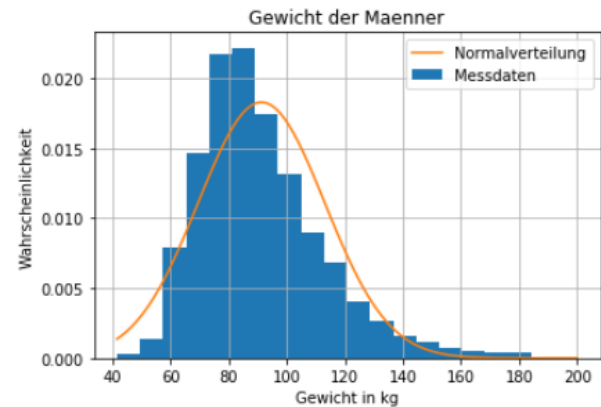
Fügen Sie eine Normalverteilung auf Basis der charakteristischen Parameter hinzu.

*Hinweis:* Benutzen Sie das Modul `scipy.stats.norm`.

```
In [6]: import matplotlib.pyplot as plt
from scipy.stats import norm
import numpy as np

def plot_merkmal(merkmal, resolution=100, bins=20):
    xs = np.linspace(min(merkmal), max(merkmal), resolution)
    ys = [norm.pdf(x, loc=merkmal.mean(), scale=merkmal.std()) for x in xs]
    plt.hist(merkmal, density=True, bins=bins, label='Messdaten')
    plt.plot(xs, ys, label='Normalverteilung')
    plt.grid()
    plt.legend()

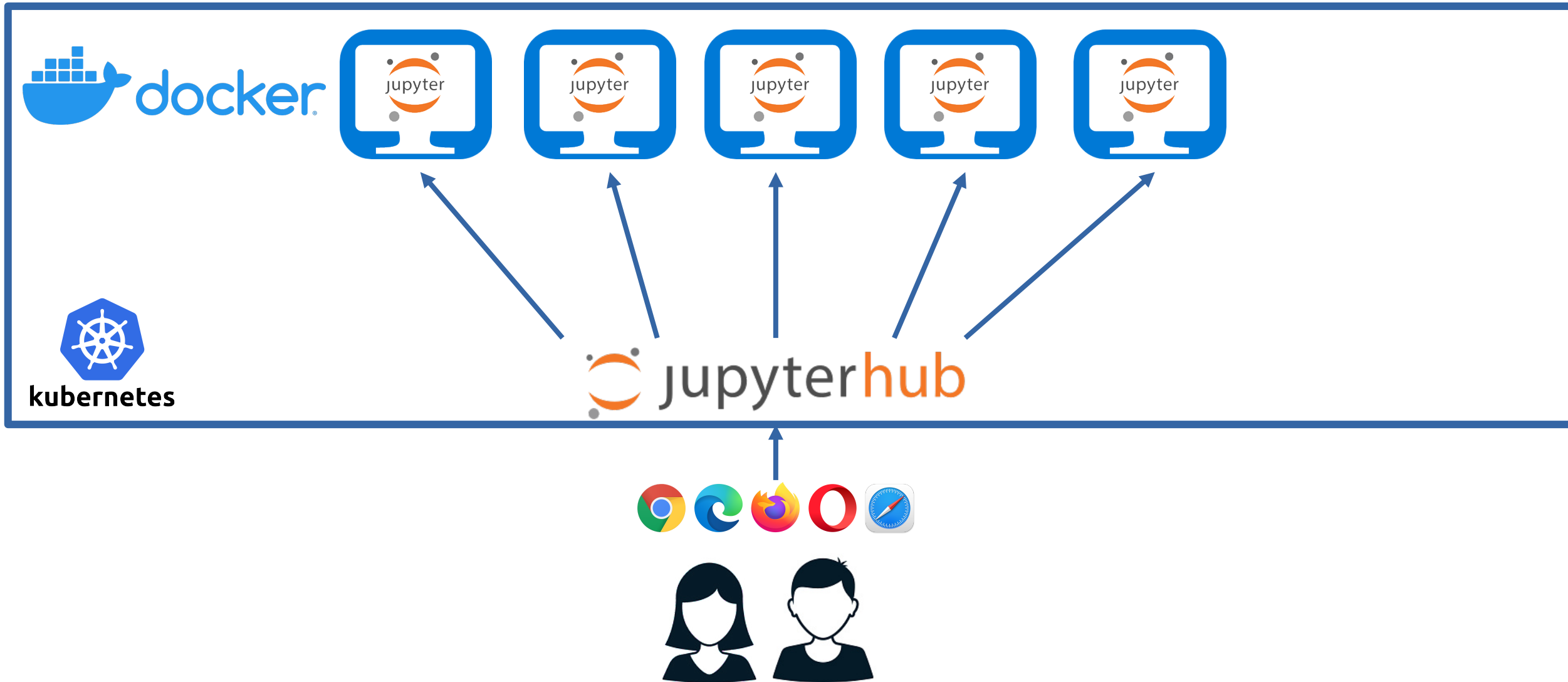
plot_merkmal(men.Gewicht)
plt.xlabel('Gewicht in kg')
plt.ylabel('Wahrscheinlichkeit')
plt.title('Gewicht der Maenner')
plt.show()
```





- Ein web-basierter Multi-User-Server für Jupyter Notebooks
- Studierende haben einen Account auf dem JupyterHub und benötigen nur einen Webbrowser
- Die Infrastruktur und Programmierumgebung wird zentral gewartet
- Vorteile:
  - Alle Studierenden haben die gleichen Ressourcen
  - Alle Werkzeuge liegen in festgelegten Versionen vor
  - Niedrige Einstiegshürde

# Infrastruktur



## Welcome to e2x JupyterHub server

Sign up to get access to the server!

Documentations and tutorials: [graders](#) and [students](#).

### Sign In

**Username:**

**Password:**

Don't have an account? [Signup!](#)

# Constructive Alignment

- Abstimmung von Lehrveranstaltungen, Prüfungen und Lernergebnissen
- Studierende sollen Umgang mit relevanten Werkzeugen und Methoden lernen und diese anwenden
- Lehrmethoden: Übungsbetrieb während des Semester, häufiges Feedback
- Prüfung soll Lernziele widerspiegeln: Übung = Prüfung
- Hoher Aufwand nur mit modernen Bewertungsmethoden beherrschbar

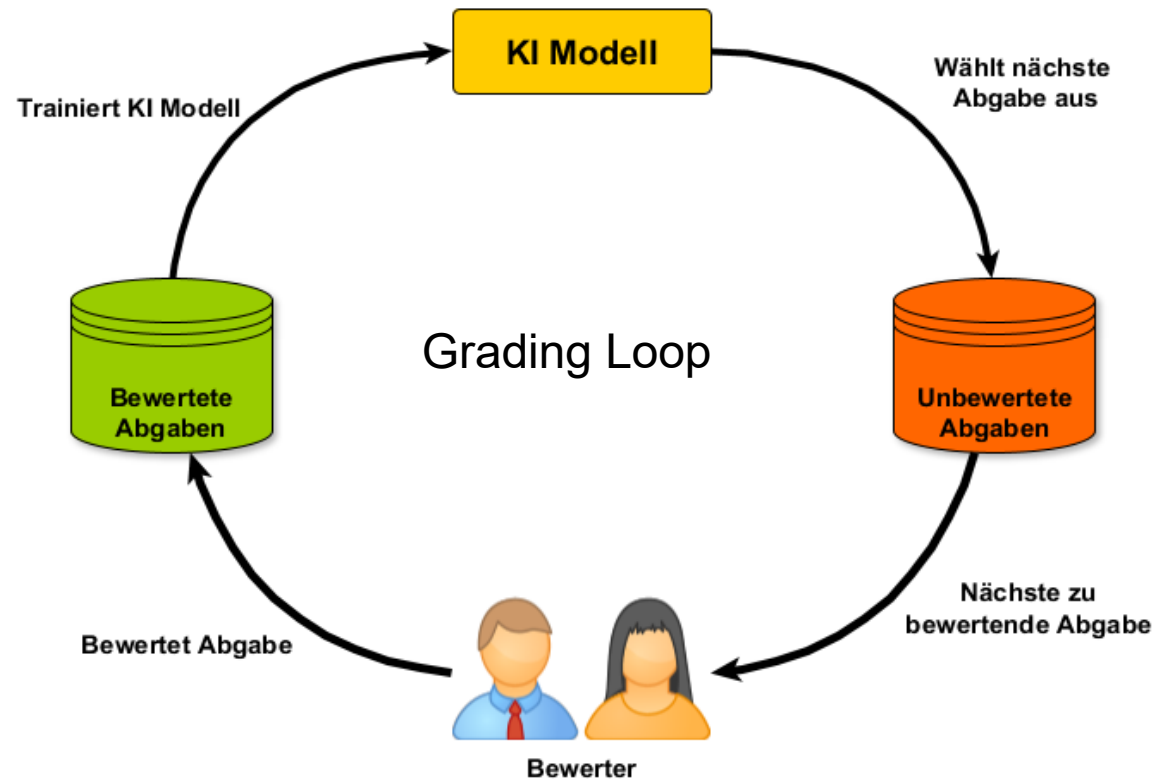
# Rechenbeispiel: Bewertung von Übungen und Klausuren

- ~300 Studierende
- 10 Aufgaben pro Übung
- 6 Übungen im Semester
- 1 Minute pro Aufgabe pro Teilnehmer
- → Bewertungsaufwand: 18.000 Minuten = 300 Stunden



# Bewertungsmethoden

- Ziel: Hoher Anteil an computergestützten Bewertungsmethoden
- Bewertungsmodule als Webservices
- Automatische Tests für Code
- Aktuelles Forschungsthema: Active Learning Grader für Textantworten





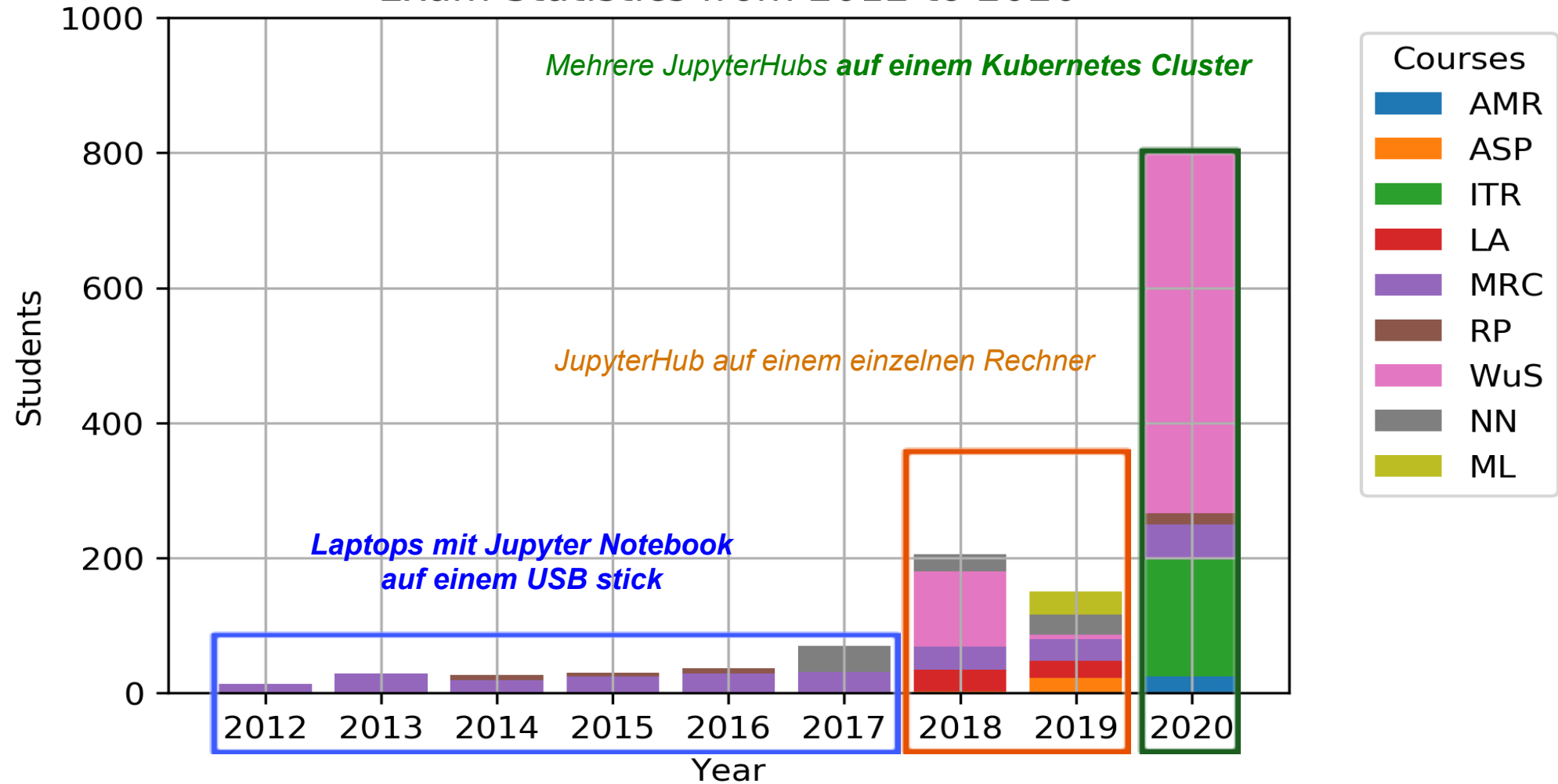
# E2XGrader

- Plugin für Jupyter Notebooks zur Bewertung von Abgaben
- Open Source
- Modular
- Erweiterbar durch Bewertungsservices
- Randomisierung
- Plagiatsprüfung

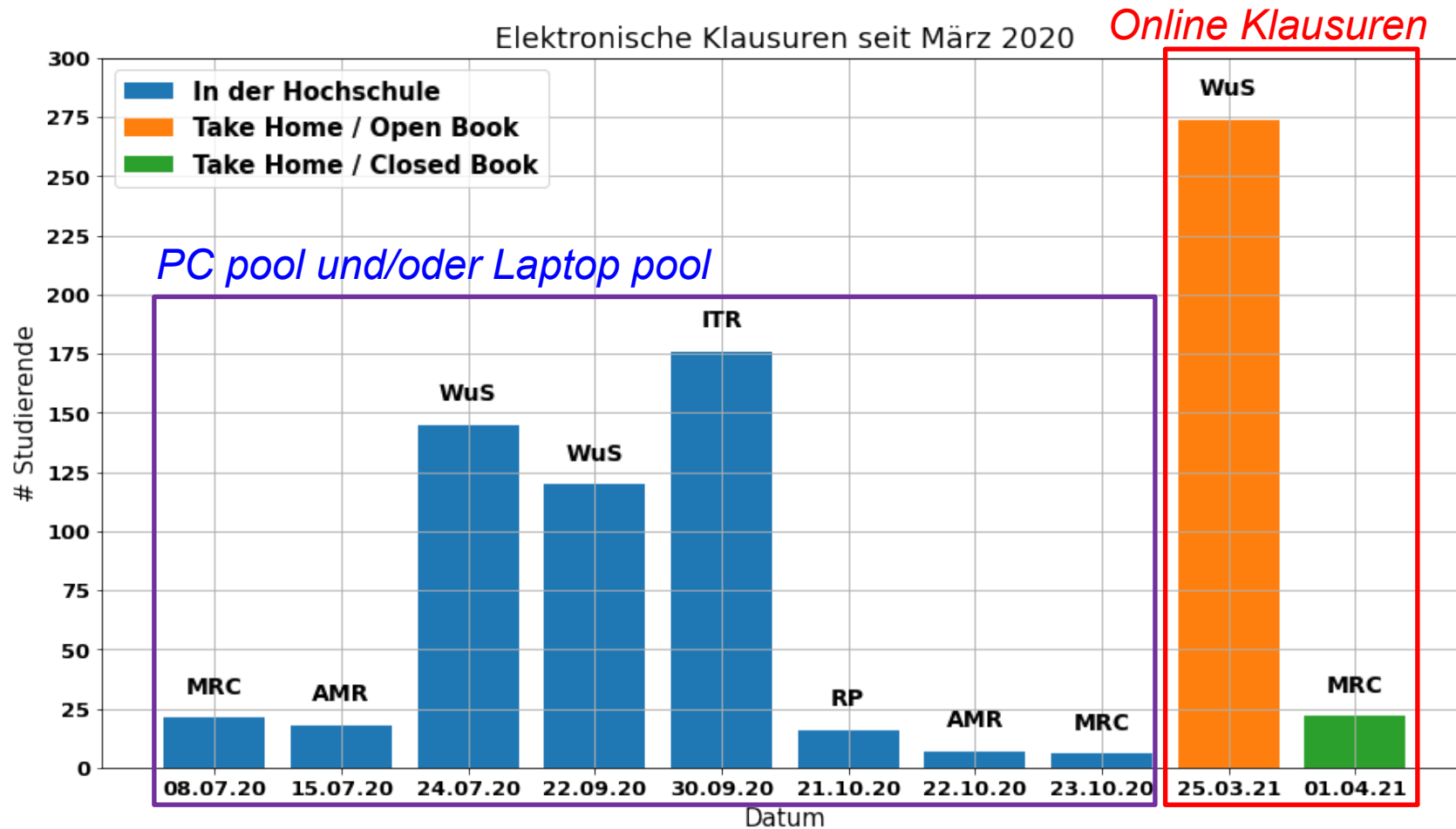


# Historie von Klausuren seit 2012

Exam Statistics from 2012 to 2020



# Prüfungen während der Pandemie



# Klausur aus Sicht der Studierenden



Hochschule  
Bonn-Rhein-Sieg

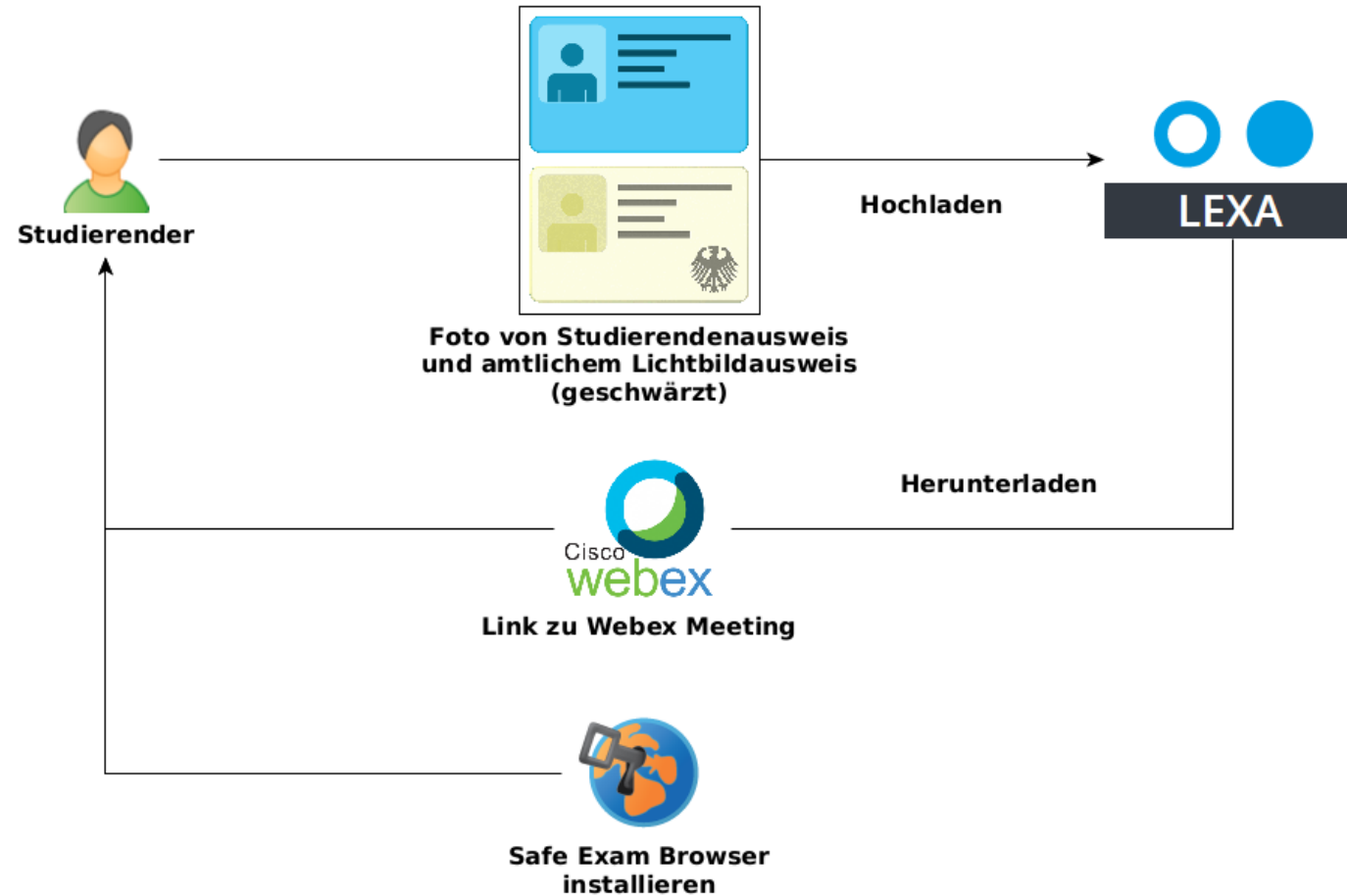
Fachbereich  
Informatik

Prof. Dr.  
Paul G. Plöger

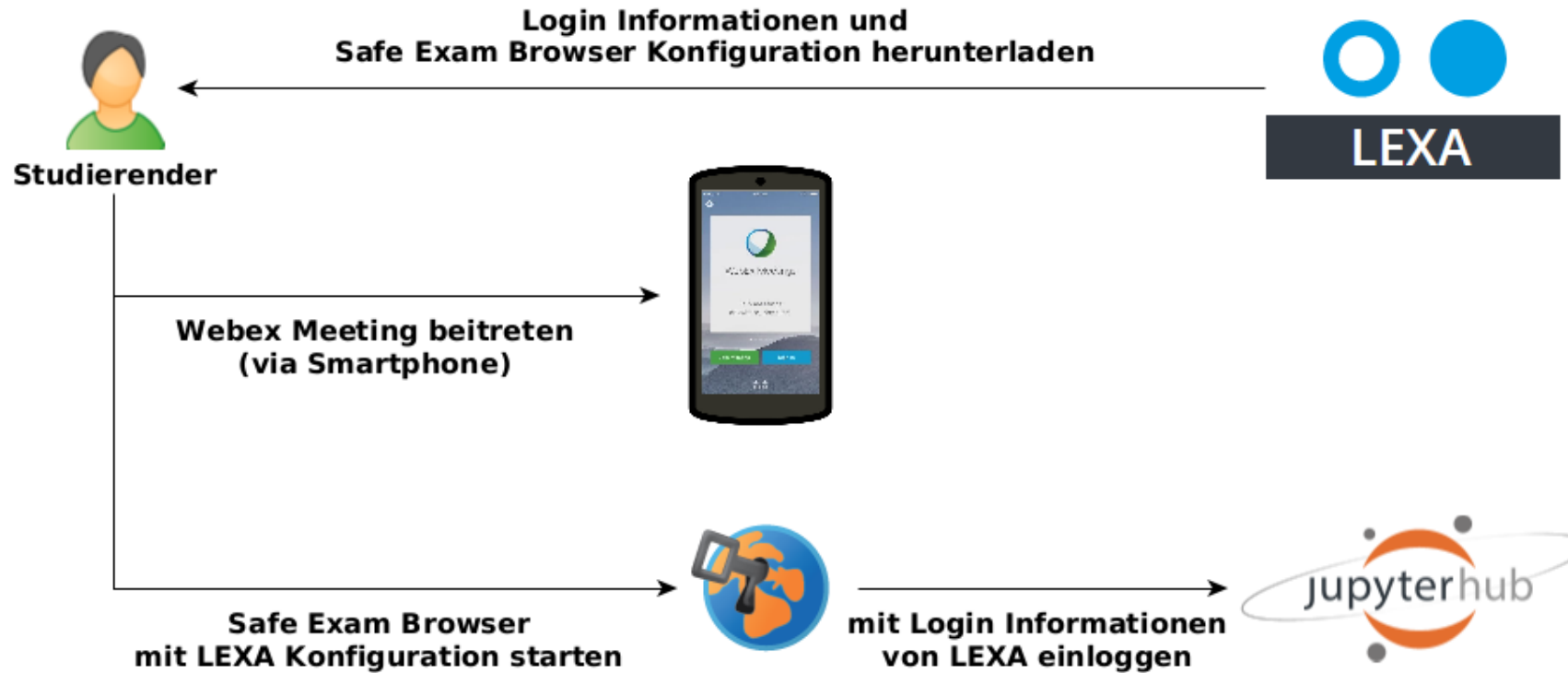
M.Sc. Tim Metzler

M.Sc. Mohammad Wasil

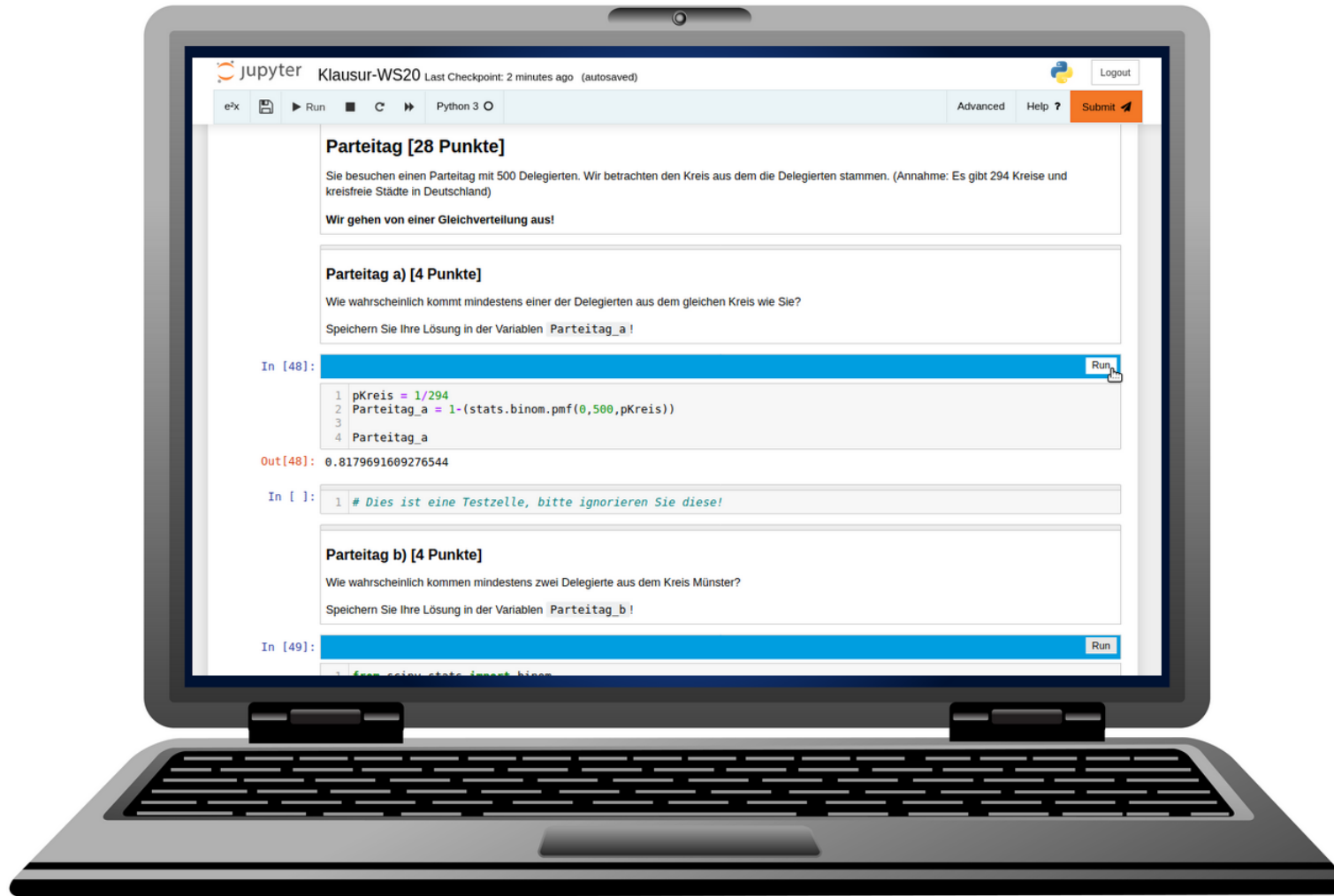
# Studierende vor der Klausur



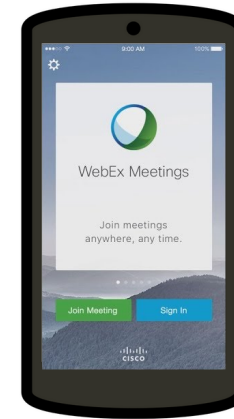
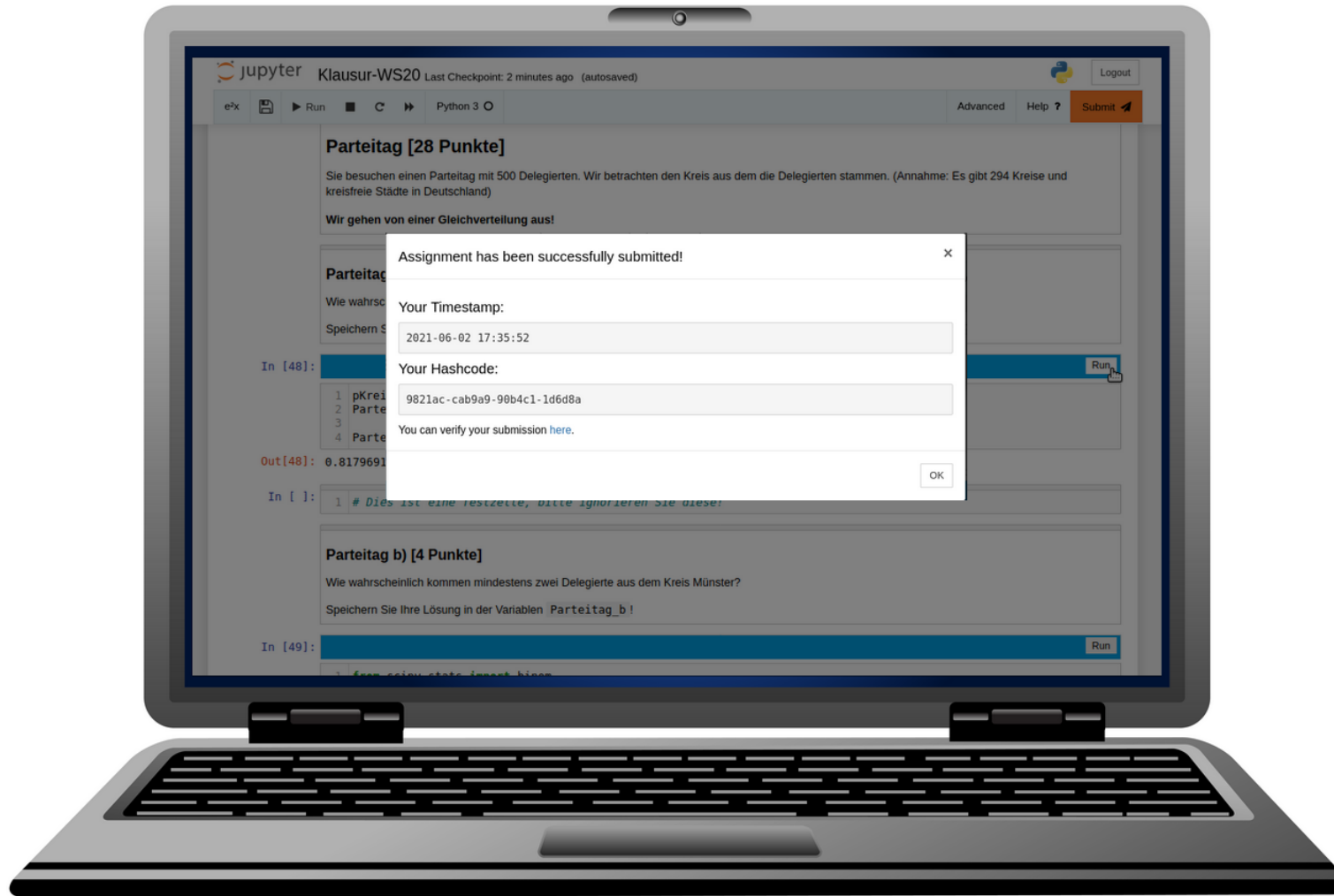
# Studierende zu Beginn der Klausur



# Studierende während der Klausur



# Studierende: Abgabe der Klausur (1/2)





# Studierende: Abgabe der Klausur (2/2)

Assignment has been successfully submitted! ✕

Your Timestamp:

2021-06-02 17:35:52

Your Hashcode:

9821ac - cab9a9 - 90b4c1 - 1d6d8a

You can verify your submission [here](#).

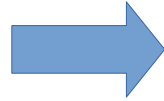
**Attention: You marked your exam as **valid**** OK



2. Aufsicht über  
Abgabe  
informieren

1. Notieren des Hashcodes (Prüfsumme)  
und des Zeitstempels auf einem Blatt  
Papier.

# Abgabe der Klausur: Prüfsumme berechnen



**Prüfsumme  
berechnen**



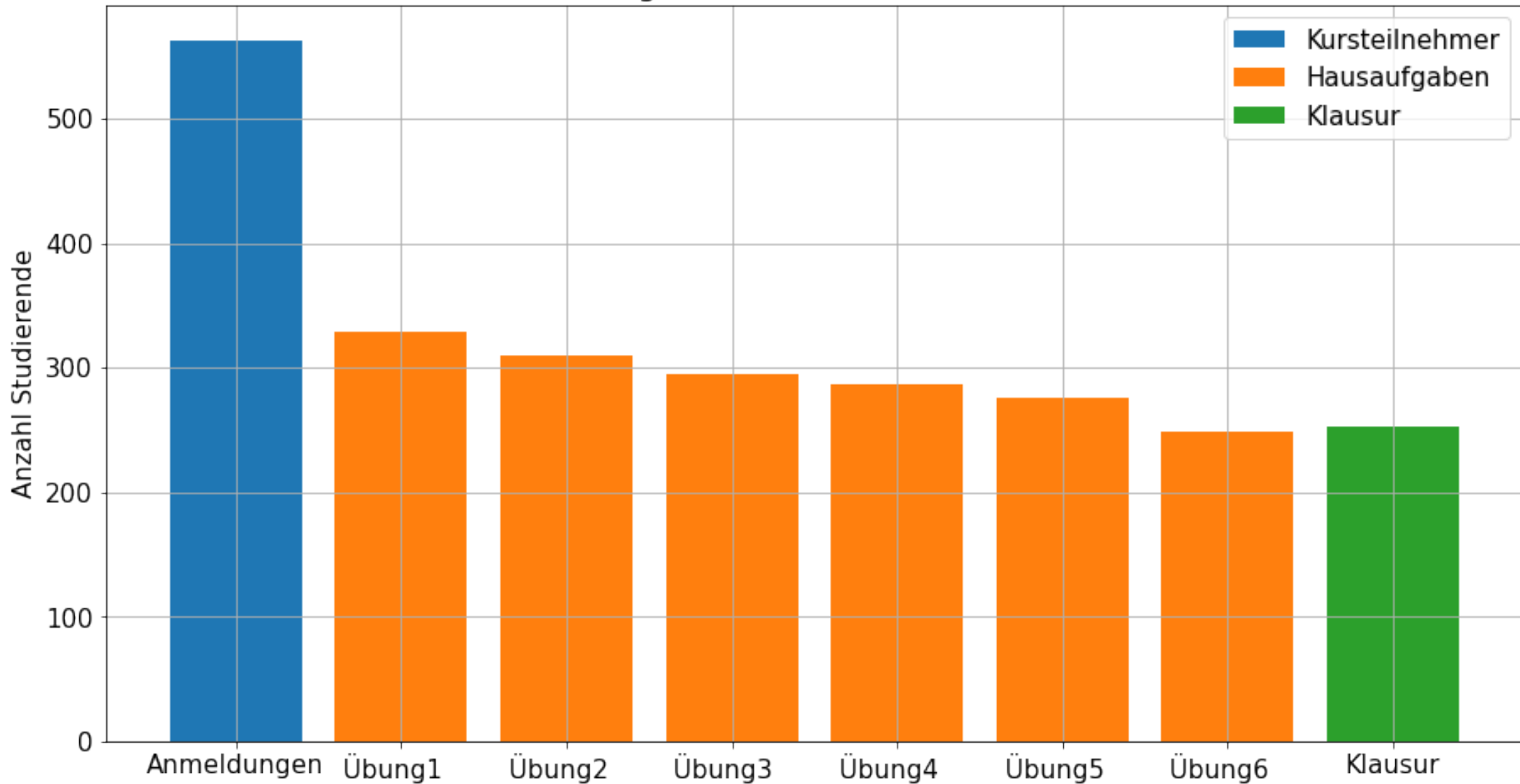
**7dc1d-f5a18-  
3c23b-28199**

# **Erfahrungen aus dem Kurs Wahrscheinlichkeitstheorie und Statistik**



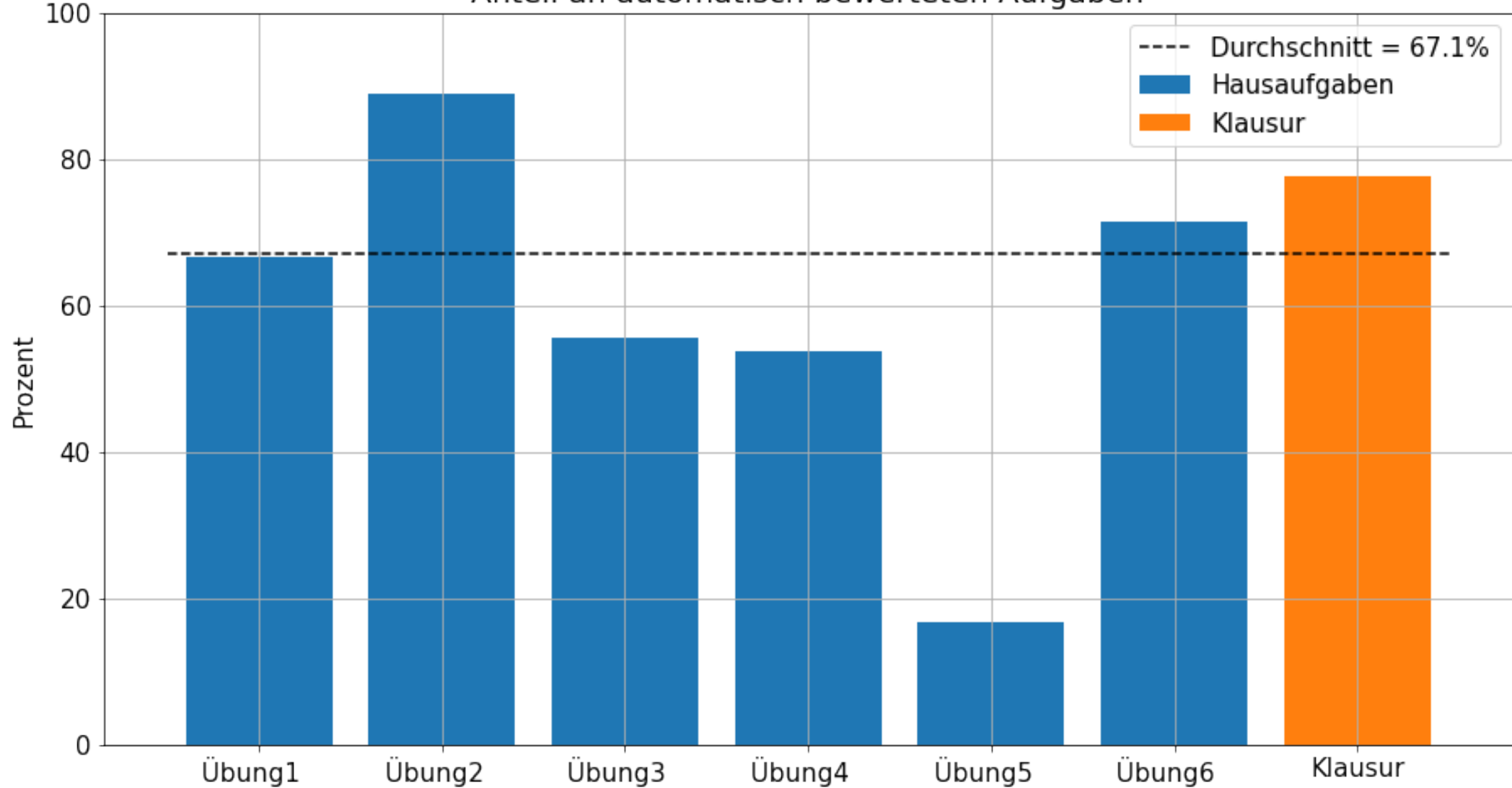
# WuS - Übungen

Abgaben über das Semester



# WuS - Übungen

Anteil an automatisch bewerteten Aufgaben

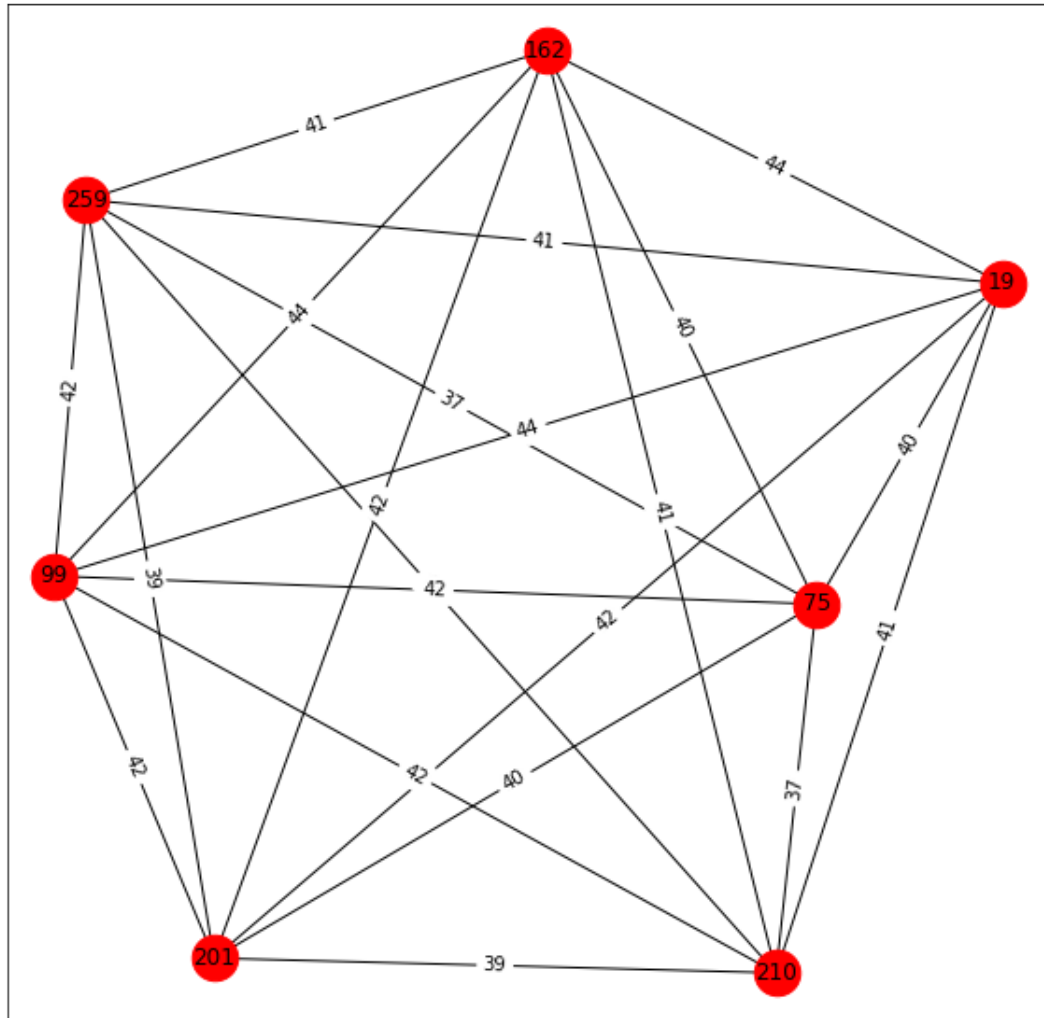


# WuS Übungen: Statistik

- **1.998** Abgaben von Übungsblättern / Klausuren
- **24.637** zu bewertende Einzelaufgaben
- **16.538** automatisch bewertet
- **8.099** händisch bewertete Einzelaufgaben
- **4** Bewerter
- **2.025** zu bewertende Einzelaufgaben pro Bewerter über das Semester
- Feedback zu jeder Übung **14 Tage** nach der Abgabe
- Klausurergebnisse nach **25 Tagen** (wegen Feiertagen)



# Plagiatsprüfung



- Studierende erhalten während der Übungen Bonuspunkte für die Klausur
- Automatische Code Plagiatsprüfung (Stanford Moss)
- Automatische Erstellung von Ähnlichkeitsgraphen

# Plagiatsprüfung

```
13 def muenzSpiel(folge1: str, folge2: str) -> str:
14     """
15     Werfe eine Muenze solange, bis eine der beiden Eingabefolgen auftritt
16
17     Arguments:
18         folge1 -- Folge fuer Spieler 1
19         folge2 -- Folge fuer Spieler 2
20     Returns:
21         gewinner -- Wessen Folge zuerst auftritt
22     """
23     temp = "" + muenze()
24     gewinner = ""
25     while gewinner == "":
26         temp += muenze()
27         if temp[-2:] == folge1:
28             gewinner = "Spieler 1"
29         if temp[-2:] == folge2:
30             gewinner = "Spieler 2"
31     return gewinner
32
33 muenzSpiel('KK', 'ZZ')
```

Abgabe 162

```
13 def muenzSpiel(folge1: str, folge2: str) -> str:
14     """
15     Werfe eine Muenze solange, bis eine der beiden Eingabefolgen auftritt
16
17     Arguments:
18         folge1 -- Folge fuer Spieler 1
19         folge2 -- Folge fuer Spieler 2
20     Returns:
21         gewinner -- Wessen Folge zuerst auftritt
22     """
23     # YOUR CODE HERE
24
25     folge = "" + muenze()
26     sieger = ""
27     while sieger == "":
28         folge += muenze()
29         if folge[-2:] == folge1:
30             sieger = "Spieler1"
31         if folge[-2:] == folge2:
32             sieger = "Spieler2"
33     return sieger
```

Abgabe 201



# WuS Klausur: Randomisierung

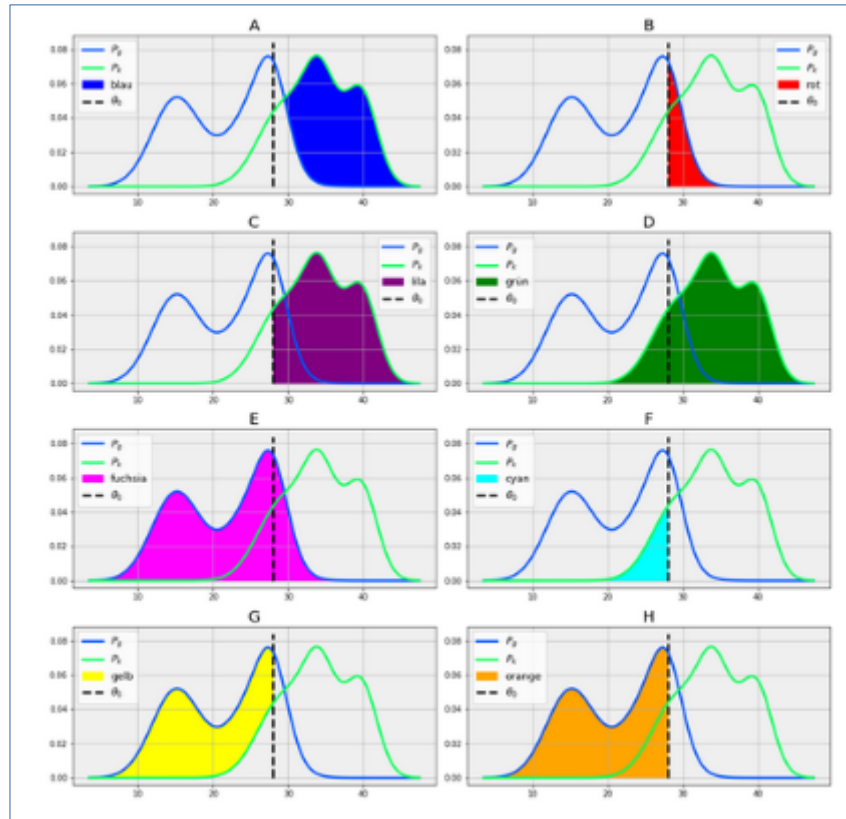
Aufgabe	Textvarianten	Variationen
Geburtstag	5	81
Urne	2	6.845.094
Wahlen	6	542.148
Kartenspiel	1	4.530.960
Lohngruppe	6	24.000
Nutzer	6	1.069.200
Population	1	5.284.823.040.000
Sportler	224	173.408.256
Würfelsumme	1	205

Theoretische Anzahl verschiedener Klausuren:

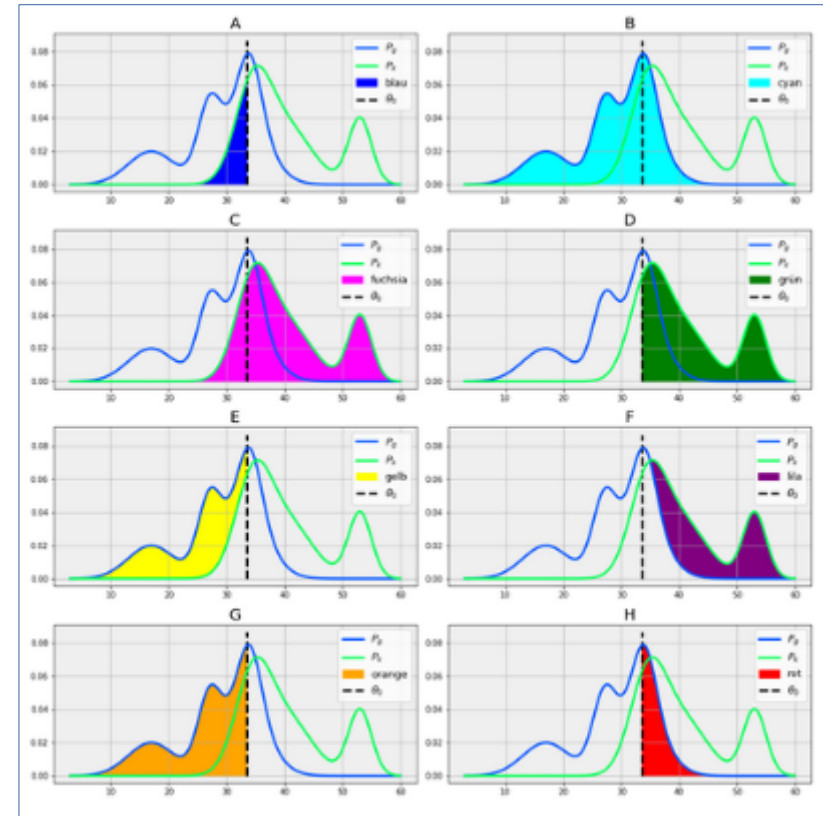
$$6 * 10^{54}$$

# Randomisierung - Beispiel

Aufgabe: Wählen Sie den richtigen Plot aus



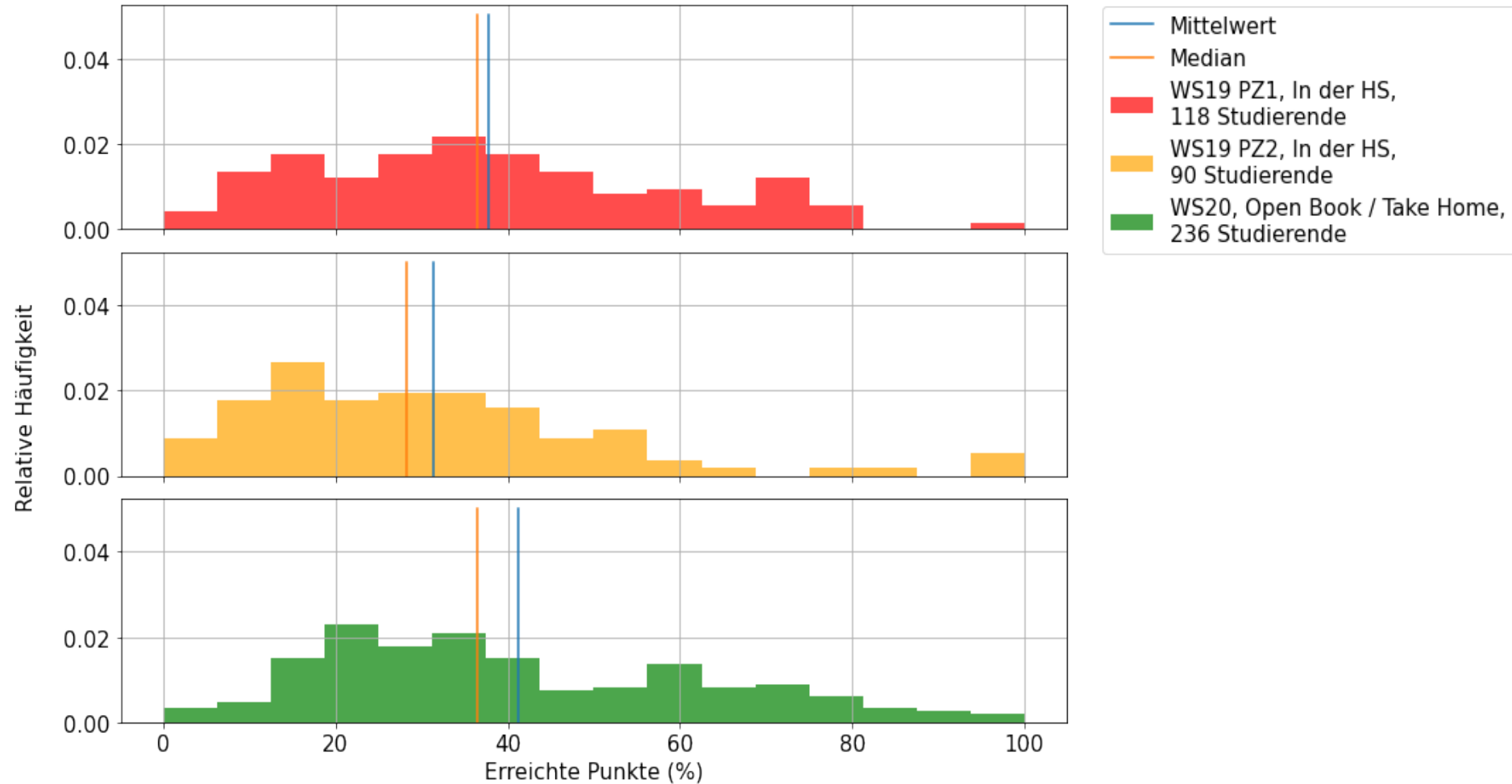
Variante 1



Variante 2

# WuS Vergleich Take Home / In House

Prozent der erreichten Punkte in der Klausur  
100% = höchste erreichte Punktzahl



# Vielen Dank für Ihre Aufmerksamkeit! Haben Sie Fragen?

[Tim.Metzler@h-brs.de](mailto:Tim.Metzler@h-brs.de)  
[Mohammad.Wasil@h-brs.de](mailto:Mohammad.Wasil@h-brs.de)  
[Paul.Ploeger@h-brs.de](mailto:Paul.Ploeger@h-brs.de)

<https://e2x.inf.h-brs.de>  
<https://github.com/Digiklausur>



# Randomisierung - Beispiel

## Aufgabe 6

### Verdorbene Ware [12 Punkte]

Ein Lieferant von Salat verkauft Lieferungen, bei der eine gewisse Anzahl an Salatköpfen verdorben ist.

Der Lieferant möchte nicht jeden Salat prüfen, sondern beschränkt sich auf eine Stichprobe. Wird ein verdorbener Salat gefunden, überprüft der Lieferant die gesamte Lieferung.

### Verdorbene Ware a) [4 Punkte]

Es seien 9 Salate von 195 Salate verdorben.

Sie überprüfen 21 verschiedene Salate. Wie hoch ist die Wahrscheinlichkeit die verdorbene Ware zu erkennen? Hierbei reicht es mindestens einen verdorbenen Salat zu erkennen.

Speichern Sie Ihre Lösung als Fließkommazahl in der Variablen `wahrscheinlichkeit_erkannt`.

# Randomisierung - Beispiel

## Aufgabe 6

### Losbude [12 Punkte]

Sie gehen auf einen Jahrmarkt mit einer Losbude. Als Gewinn gibt es ein Stofftier, welches Sie haben möchten.

#### Losbude a) [4 Punkte]

Es seien 76 Lose von 815 Losen Gewinnlose.

Sie kaufen 21 verschiedene Lose. Wie hoch ist die Wahrscheinlichkeit ein Gewinnlos zu kaufen? Hierbei reicht wenn mindestens ein Gewinnlos unter den Losen ist.

Speichern Sie Ihre Lösung als Fließkommazahl in der Variablen `wahrscheinlichkeit_gewonnen`.