



TÜBINGER BEITRÄGE ZUR HOCHSCHULDIDAKTIK

Computergestützte Methodenkompetenzvermittlung

für Studienanfänger in den
wirtschaftswissenschaftlichen Studiengängen



Johannes Bleher und Thomas Dimpfl

Band 17/1 · Tübingen 2021

DEZERNAT III – STUDIUM UND LEHRE
Arbeitsstelle Hochschuldidaktik



Bibliographische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie. Detaillierte bibliographische Daten sind im Internet über <http://ddb.de> abrufbar.

Bitte zitieren Sie dieses Dokument als:

<http://hdl.handle.net/10900/116067>

<http://nbn-resolving.de/urn:nbn:de:bsz:21-dspace-1160674>

<http://dx.doi.org/10.15496/publikation-57442>

ISSN: 1861-213X

Redaktion und Satz dieses Bandes:

Andrea Fausel, Heiko Heil

Layout: Heiko Heil

© Universität Tübingen

Eberhard Karls Universität Tübingen

Dezernat III – Studium und Lehre

Arbeitsstelle Hochschuldidaktik

Sigwartstr. 20

72076 Tübingen

Tel.: +49 (0) 7071 29-78385

Fax: +49 (0) 7071 29-5615

hochschuldidaktik@uni-tuebingen.de

www.uni-tuebingen.de/hochschuldidaktik

Titelbild:

© Maximusnd – istockphoto.com

Dieses Dokument wird bereitgestellt von TOBIAS-lib

Eberhard Karls Universität Tübingen

Universitätsbibliothek

Hochschulpublikationen/Dissertationen

Wilhelmstr. 32

72074 Tübingen

Tel.: +49 (0) 7071 29-76999

Fax: +49 (0) 7071 29-3123

hochschuldidaktik@uni-tuebingen.de

edl-publ@ub.uni-tuebingen.de

<http://tobias-lib.uni-tuebingen.de>

Editorial

Mit den Tübinger Beiträgen zur Hochschuldidaktik haben wir uns zum Ziel gesetzt, den hochschuldidaktischen Diskurs an der Universität Tübingen zu fördern und zugleich öffentlich zu machen. Die Zeitschrift ist ein Ort für fachwissenschaftliche Überlegungen, begleitet aber auch bewusst den interdisziplinären Dialog. Dabei wurden in den letzten Jahren eine Reihe von Themen angesprochen, Perspektiven beleuchtet und Impulse zur Gestaltung der Lehre gegeben.

Die Tübinger Beiträge zur Hochschuldidaktik bieten zum einen ein Forum für Abschlussarbeiten, die in der Qualifizierung für Lehrende im Rahmen des Hochschuldidaktikzentrums Baden-Württemberg entstehen. Die hohe Qualität dieser Arbeiten und das Ziel, diese einer hochschuldidaktisch interessierten Öffentlichkeit bekannt zu machen, führte 2005 zur Gründung der Zeitschrift. Zum Zweiten werden in der Zeitschrift in einem weiteren Rahmen aktuelle hochschuldidaktische Fragestellungen, Projekte aus der Praxis und innovative Lehrformate aufgegriffen. Die Beiträge sollen Lehrenden, die didaktischen Herausforderungen begegnen, Reflexionsansätze und praktische Hilfestellungen bieten. In einer Zeit sich verändernder Rahmenbedingungen für das Lehren und Lernen können auch erfahrene Lehrende neue Anregungen finden.

Der vorliegende Band widmet sich der computergestützten Methodenvermittlung in den wirtschaftswissenschaftlichen Studiengängen. Er zeigt, wie Studierende gerade in der Studieneingangsphase mittels optimaler technischer Unterstützung an komplexe Themenfelder wie statistische Programmierung herangeführt werden können. Studierende erhalten ganz selbstverständlich und ortsunabhängig die Möglichkeit, Programmiersprachen zu nutzen und einzuüben. Das hier vorgestellte Projekt erhielt 2019 den Lehrpreis der Universität Tübingen.

Wir wünschen Ihnen eine anregende Lektüre!

Andrea Fausel, Regine Richter

Tübingen, im April 2021

Inhalt

Abstract	5
1. Einleitung	5
2. Mathematisches Propädeutikum	8
3. JupyterHub	11
3.1. Softwarekonfiguration	11
3.2. Anwendung in der Lehre	13
4. Online-Spiel zur Einübung linearer Algebra	14
4.1. Erzeugung ganzzahliger Matrizen mit ganzzahligen Eigenwerten	16
4.1.1. Theoretische Grundlagen	17
4.1.2. Implementierung in Python	18
4.2. Backend der WebApp	20
5. Fakultätsgitlab	21
6. vPCL: Virtuelles PC-Labor	22
7. Erste Evaluationsergebnisse	24
8. Fazit	25
Abbildungsverzeichnis	26
Literatur	26
Anhang	27
Autoren	32

Computergestützte Methodenkompetenzvermittlung für Studienanfänger in den wirtschaftswissenschaftlichen Studiengängen

Abstract

Mit dem Projekt „Computergestützte Methodenkompetenzvermittlung für Studienanfänger in den wirtschaftswissenschaftlichen Studiengängen“ unterstützen wir die grundlegende Methodenausbildung der Studierenden in den ersten Semestern. Den Studierenden stehen computerbasierte Hilfsmittel wie der Zugang zu unserem Lehrserver, darauf abgestimmte Lehrvideos und ein Online-Spiel zur Verfügung, um Inhalte aus den Mathematischen Propädeutika und den Vorlesungen zu Mathematik und Statistik zu illustrieren und zu vertiefen. Im Hintergrund läuft technisch ein Server, der insbesondere den JupyterHub, das Online-Spiel und ein Gitlab bereitstellt. Die niederschwellige Bereitstellung von Software und Infrastruktur mittels Browser auf technischer Seite und die Integration in die Vorlesungen senkt die Hemmschwelle auf Seiten der Studierenden, die bereitgestellten Angebote auch zu nutzen. Das Feedback der Studierenden war dementsprechend positiv und es zeigen sich auch erste Lernerfolge in den Klausurergebnissen. Das Projekt erhielt den Lehrpreis der Universität Tübingen 2019 sowie den von der Fachschaft des Fachbereichs Wirtschaftswissenschaften verliehenen WiWi-Impuls 2019 für Innovation in der Lehre. Das Projekt wurde vom Ministerium für Wissenschaft, Forschung und Kunst des Landes Baden-Württemberg im Rahmen des Strukturmodelle-Projekts „Initiationswege in die Wissenschaft“ und dem Nachfolgeprojekt „Studienstart“ gefördert.

1. Einleitung

Im Rahmen des Projekts „Computergestützte Methodenkompetenzvermittlung für Studienanfänger in den wirtschaftswissenschaftlichen Studiengängen“ haben wir in den vergangenen Jahren damit begonnen, die grundlegenden Methoden der Datenaufbereitung, der Datenanalyse und deren Visualisierung an Studierende ab dem ersten Semester praktisch am Computer zu vermitteln. Die genannten Kompetenzen werden auch gerne unter dem Schlagwort *Data Literacy* subsumiert. Gekoppelt ist das Projekt an die mathematische und statistische Grundausbildung in den wirtschaftswissenschaftlichen Studiengängen der Universität Tübingen.

Dabei steht das Ziel im Vordergrund, die Studierenden nach und nach zu befähigen, passende Analysemethoden für Problemstellungen selbstständig auszuwählen, zu implementieren oder gar zu entwickeln. Wir sind davon überzeugt, dass eine wirtschaftswissenschaftliche Ausbildung nicht umhinkommt, die Studierenden neben der mathematischen und statistischen Grundausbildung auch sehr früh an grundlegende Programmier- und Datenarbeit heranzuführen. Die Problematik, informationstechnologische Inhalte neben den Kernthemen eines Fachbereichs zu vermitteln, ist auch in Studiengängen anderer Fachgebiete der Sozialwissenschaften, aber auch in den Naturwissenschaften, anzutreffen. Die Digitalisierung macht genauso wenig vor Textanalysen in den Sprachwissenschaften halt, wie vor der Automatisierung von Verwaltungsabläufen oder der statistischen Auswertung ökonomischer Daten.

Dabei gibt es zwei Problemstellungen, mit denen Lehrende konfrontiert sind. Zum einen stehen oftmals zu wenig Ressourcen von Seiten der Universität zur Verfügung, um jedem Studierenden einen Arbeitsplatz an einem mit der notwendigen Software ausgestatteten Rechner zur Verfügung zu stellen. Werden jedoch – um die-

ser Problematik zu begegnen – Veranstaltungen durchgeführt, bei denen Studierende ihre eigenen Laptops mitbringen, werden die ersten Unterrichtseinheiten oft damit zugebracht, Geräte (oder betriebssystemspezifische) Fehler zu beheben. Das schmälert den für die Implementation theoretischer Konzepte verfügbaren Zeitrahmen und frustriert beide Seiten, Studierende als auch Lehrende. Bei Lehrenden, die Experten in ihrem jeweiligen Fachbereich sind, übersteigt das geräte- und betriebssystemspezifische Fehlersuchen auf den Geräten der Studierenden oftmals die IT-Kompetenzen. Für Studierende ist es frustrierend, wenn der Lehrende sie mit ihren Problemen alleine lässt oder sich gefühlt nur um die anderen kümmert. Zum anderen erweitern die IT-Kurse oftmals das Curriculum erheblich und es ist schwer, Studierende zu zusätzlichen Veranstaltungen zu bewegen, ohne Kernthematiken der fachspezifischen Ausbildung zu opfern. Am Ende soll das Ziel noch immer sein, dass in den Wirtschaftswissenschaften Wirtschaftswissenschaftler ausgebildet werden und keine (schlechteren) Informatiker. Allein ein kleiner Kreis an Studierenden kämpft sich durch die IT-Veranstaltungen und profitiert von den erlangten IT-Kenntnissen oftmals auch in anderen Studienbereichen.

Um diese beiden Punkte anzugehen, haben wir eine Open-Source-Datenanalyse-Werkstatt, einen JupyterHub¹, auf einem Universitätsserver eingerichtet. Auf diesen JupyterHub können die Studierenden mit allen internetfähigen Endgeräten zugreifen. Dies hebt die Beschränkung auf ein PC-Labor mit gegebener Platzzahl auf und erleichtert die Laptop-gestützte Lehre, da nur noch ein Browser zur Teilnahme an der Lehrveranstaltung nötig ist. Ferner haben wir ein virtuelles PC-Labor in Zusammenarbeit mit der bwCloud eingerichtet, um auch lizenzierte Software nutzen zu können. So können die gestellten Aufgaben ortsunabhängig und im individuellen Lerntempo bearbeitet werden. Auf diese Art erreichen wir einen niederschweligen Zugang zur Datenanalyse, der insbesondere die Installation von Software auf heimischen Endgeräten im ersten Schritt unnötig macht. Auch die von den Studierenden genutzte standardisierte Umgebung auf dem JupyterHub, das JupyterLab, erlaubt es, die gerätespezifische Fehlersuche in den ersten Veranstaltungseinheiten zu vermeiden. Fehler, die im JupyterLab gemacht werden, werden in einer kontrollierten Umgebung gemacht und sind daher leichter zu beheben. Sowohl auf Seiten der Studierenden als auch der Lehrenden sinkt daher das Frustrationspotenzial enorm. Für die Studierenden ist es zudem nützlich, dass sie zuhause auf die exakt gleiche Umgebung zugreifen, welche auch in der Vorlesung verwendet wird. Dies senkt die Hemmschwelle, zur Verfügung gestellte Programme tatsächlich anzuschauen und zu verwenden. Da wir bei der Einrichtung des JupyterHubs ausschließlich auf Open-Source-Software zurückgegriffen haben, ist das Projekt sofort ohne weitere Lizenzen reproduzierbar, sofern ein Server mit ausreichender Rechenkapazität zur Verfügung steht oder noch besser ein Cluster an Servern, auf dem die für den Betrieb notwendigen Ressourcen dynamisch angefordert und freigegeben werden können. Den Studierenden steht natürlich ebenfalls der Weg offen, sich die Software bei Bedarf lokal zu installieren und damit weiterzuarbeiten.

Konkret ist das Projekt derzeit für fünf Vorlesungen des Lehrstuhls für Statistik, Ökonometrie und empirische Wirtschaftsforschung konzipiert. Andere Lehrstühle nutzen die JupyterHub-Ressourcen auch, waren aber an der ursprünglichen Konzeption nicht beteiligt. Dennoch zeigt die Teilnahme anderer Lehrstühle, dass das System beliebig erweiterbar ist und zum einfachen Gebrauch einlädt. In allen Kursen erhalten die Studierenden in der Regel vorgefertigte Skripte, welche auch in der Präsenz-Vorlesung verwendet werden, um diese zu adaptieren und dabei die Programmierung in R sowie die Inhalte der Vorlesung zu verstehen und zu verinnerlichen. Dabei sind manche Skripte lediglich zur Veranschaulichung bestimmter Thematiken gedacht, in der Regel sind sie aber mit konkreten Aufgaben verbunden.

Den Open-Source-Skripten, die auf dem JupyterHub zur Verfügung stehen, wurden auch Skripte für proprietäre Software in SAS und Stata zur Seite gestellt. Sowohl SAS als auch Stata können aus Lizenzgründen nur eingeschränkt im PC-Labor der Wirtschaftswissenschaften verwendet werden. Um auch diese Ortsrest-

1 Der Name JupyterHub wurde von der Open-Source-Community gewählt, weil es sich um eine Plattform bzw. ein Verteilzentrum (daher auch der englische Begriff Hub) handelt, welches einzelnen Nutzern eigene Umgebungen mit Jupyter-Notebooks zuweist. Jupyter-Notebooks wurden ursprünglich dafür konzipiert, Programmcode in den Sprachen Julia, Python und R (daher der Name Jupyter) mit Markup-Code (in dem auch HTML und Latex-Code enthalten sein kann) zu vermengen, zu interpretieren und zu interaktiven Dokumenten zusammenzustellen. Durch die Interaktion kann Datenarbeit spielerisch erlebt und einfach dokumentiert werden.

riktion für Studierende und Lehrende zu lockern haben wir einen Virtualisierungsdienst implementiert, mit dem auf die Infrastruktur der bwCloud zurückgegriffen werden kann. Die bwCloud ist ein von baden-württembergischen Universitäten gegründeter Anbieter von „Infrastructure as a Service“ (IaaS). Durch die auf dem Server-Konglomerat der bwCloud installierte Open-Source Software *Openstack* können eine Vielzahl an virtuellen Maschinen orchestriert werden. Wir haben daher auf einem unserer Universitätsserver (auf dem auch der JupyterHub-Service angeboten wird) einen Vermittlungsdienst, *das virtuelle PC-Labor (vPCL)* entworfen. Fakultätsangehörige, die eine eigene virtuelle Maschine (mit von uns standardisierter Software) auf der bwCloud starten wollen, bekommen durch unseren Dienst einfachen Zugriff auf diese Maschine. Da die auf den virtuellen Maschinen installierten Softwareprodukte für lediglich 30 Nutzer lizenziert sind, wird der Zugriff zunächst für 10 Tage gewährt und kann zweimal bis auf 30 Tage verlängert werden. So wird das Nutzen von proprietärer Software wie beim JupyterHub ortsunabhängig ermöglicht. Für die Nutzung des vPCLs ist jedoch die Installation bestimmter Software notwendig und nicht ganz so niederschwellig wie für den JupyterHub.

Durch die Möglichkeit, weitere Webapps auf den Servern bereitzustellen, wird unser Angebot um zwei weitere Applikationen erweitert. Zum einen haben wir ein fakultätsinternes Gitlab installiert. Damit müssen Forschungsinhalte oder auch Vorlesungsmaterialien von Fakultätsangehörigen nicht auf universitätsfremden Servern von gitlab.com gespeichert werden. Die Zusammenarbeit an Projekten wird durch das Versionierungswerkzeug GIT deutlich vereinfacht. Die im Gitlab angelegten Inhalte können durch ein JupyterHub-Plugin einfach mit Studierenden geteilt werden. Damit vereinfacht sich der Administrationsaufwand für das Bereitstellen und Einpflegen von Inhalten auf dem JupyterHub erheblich.

Zum anderen haben wir unabhängig von der Vermittlung von computergestützten Methoden für die Vorlesung „Mathematische Methoden“ ein Online-Spiel zum täglichen Üben von einfachen Aufgaben aus der linearen Algebra auf den vom Projekt genutzten Server bereitgestellt. Ziel hier ist es, ganz im Geiste von Gamification, die Studierenden auf einer spielerischen Ebene an die Konzepte der linearen Algebra heranzuführen. Durch die Möglichkeit an einem freiwilligen Ranking teilzunehmen, wollten wir den Kampfgeist wecken und Anreize für die regelmäßige Teilnahme bieten. Für weitere mathematische Methoden sind weitere Spiele angedacht.

Das Projekt adressiert also im Wesentlichen zwei Elemente: Erstens den niederschweligen Erwerb von *data literacy* und damit einhergehend Softwarekenntnissen, insbesondere in R und Python, anhand von konkreten Beispiel-Datensätzen, wobei der Gebrauch der von uns bereitgestellten Dienste für Studierende und Lehrende stets so einfach wie möglich gehalten werden soll. Zweitens adressiert das Projekt durch das Matrix-Spiel auch den spielerischen Erwerb von mathematischen Methoden. Durch das an der Fakultät geweckte Know-How zur Umsetzung solcher Spiele könnten in Zukunft auch weitere fachspezifische Methoden und Inhalte entsprechend verpackt werden. Kurzfristig scheint die Strategie erfolversprechend: Sowohl die Evaluationen der betroffenen Veranstaltungen als auch die Klausuren fielen im vergangenen Wintersemester 2018/19 besser aus als in den Vorjahren. Wir hoffen, die Grundlagen für einen langfristigen Erfolg geschaffen zu haben. Dies wird sich letztlich jedoch erst zeigen, wenn die Studierenden erfolgreich das Studium beendet haben.

Im Folgenden erläutern wir den Aufbau der einzelnen Elemente unseres Projekts im Detail. Abschnitt 3 beschäftigt sich mit den technischen Spezifikationen der JupyterHub Einrichtung und den für die lehrstuhleigenen Vorlesungen erstellten Inhalten. In Abschnitt 4 gehen wir auf die konkrete Implementation unseres Matrixspiels ein, gefolgt von der Implementation des fakultätseigenen Gitlafs, dessen Verknüpfung zum JupyterHub in Abschnitt 5 erläutert wird, gefolgt von der Einrichtung des virtuellen PC-Labors in Abschnitt 6. Abschnitt 7 präsentiert erste Evaluationsergebnisse für die eingerichteten Dienste. Abschnitt 8 beschließt die Ausführungen.

2. Mathematisches Propädeutikum

Der Ursprung des Projekts liegt im mathematischen Vorkurs der Wirtschafts- und Sozialwissenschaftlichen Fakultät. Ziel war es, den Studierenden eine Teilnahme zu ermöglichen, auch wenn Sie unter Umständen noch nicht persönlich in Tübingen anwesend sein können. Daher wurde beginnend mit dem Jahr 2017 ein Online-Portal über ILIAS² eingerichtet, das den weiterhin in der Woche vor Vorlesungsbeginn des Wintersemesters durchgeführten Präsenzkurs ergänzt. Auf diesem Portal haben die Studierenden die Möglichkeit, sich mittels Videos und (zusätzlichen) Aufgaben Schulstoff vor, während oder nach dem Präsenzkurs erneut zu vergegenwärtigen. Inhaltlich orientiert sich der Vorkurs an den ersten vier Kapiteln aus dem auch in der Vorlesung verwendeten Buch von Sydsæter, Strøm et al. (2013). Abbildung 1 zeigt die erste Seite des Online-Portals.

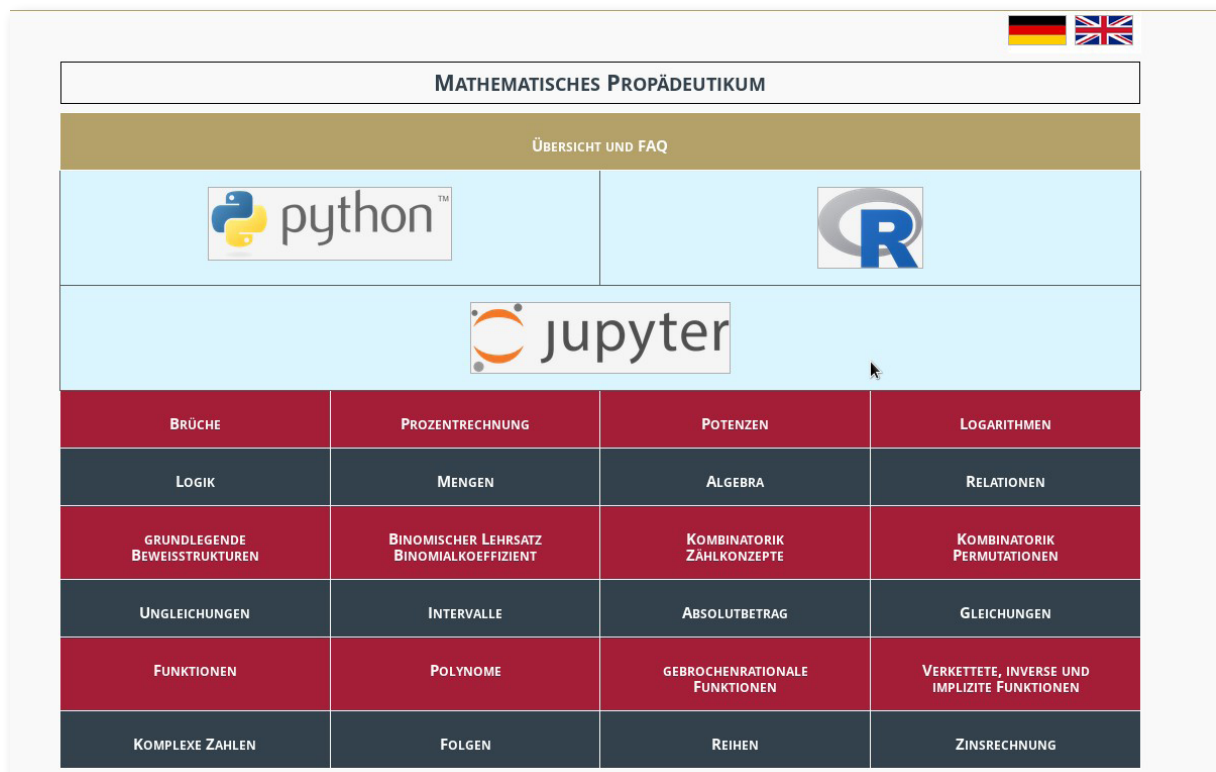


Abb. 1: Erste Seite des Mathematischen Propädeutikums

Auf der FAQ-Seite werden den Studierenden Tipps für die Bearbeitung der Videos, aber auch allgemeine Tipps für die Bearbeitung und Organisation von Studieninhalten gegeben. Die Studierenden sind angehalten, nicht alle Videos anzusehen, sondern lediglich die Themen zu bearbeiten, bei denen sie sich unsicher fühlen. Auch die Checkliste für die adäquate Vorbereitung von Klausuren von Felder (1999) ist verlinkt. Alle Angebote sind freiwillig und sowohl auf Deutsch als auch auf Englisch verfügbar.

Mit Hilfe der Videos und Aufgaben sollen die Studierenden selbstständig eigene Lücken identifizieren und diese mit deren Hilfe schließen. Hinter jeder Kachel erwartet die Studierenden ein Lernmodul mit einem oder

² https://vitruv.uni-tuebingen.de/ilias3/goto_pr01_crs_4803.html (zuletzt abgerufen: 31.03.2021)

mehreren Videos, Aufgaben und Programmierübungen. Abbildung 2 zeigt den Aufbau des Moduls für das Themengebiet *Funktionen*. Alle verwendeten Softwarekomponenten zum Erstellen der Videos sind frei verfügbare Open-Source-Produkte.

Die Lehrvideos wurden mit der Open-Source-Software *SimpleScreenRecorder*³ auf einem konvertiblen Laptop mit OpenSUSE-Betriebssystem aufgenommen. Für die Nachbearbeitung und Fehlerkorrektur wurde auf das Open-Source-Videobearbeitungsprogramm KDElive zurückgegriffen. Anfänglich wurde der libLAME-Codec verwendet, der unter MacOS nicht standardmäßig abspielbar war. Die Konvertierung des Audio-Tracks wurde mit Hilfe der Open-Source-Tools *ffmpeg*⁴ und *sox*⁵ durchgeführt. Derzeit verwenden wir als Audio-Codec H.264 mit der Einstellung „superfast“ für die Kompressionsvorlage. Um die Dateigröße der Videos weiterhin zu beschränken, wurden diese mit einer Bildrate von 15 FPS aufgezeichnet. Auch wenn das menschliche Auge in der Lage ist, 24 Bilder pro Sekunde zu unterscheiden, ist die Qualität für Bild und Sprache ausreichend. Eine höhere Auflösung und damit einhergehend eine deutlich intensivere Nutzung von Speicherplatz scheint nicht notwendig zu sein.

Abb. 2: ILIAS Seite mit Video

Um die Studierenden bereits so früh wie möglich an das Arbeiten mit statistischer Software heranzuführen, gibt es seit Beginn des Wintersemesters 2018/2019 für ausgewählte Themen sogenannte Jupyter-Notebooks sowie R- und Python-Skript-Dateien. Diese veranschaulichen die theoretischen Inhalte oder wenden diese an einem Beispiel an. Die Skripte können direkt ausgeführt werden. Die Studierenden können die Skripte

3 Projektseite: <https://www.maartenbaert.be/simplescreenrecorder> (zuletzt abgerufen: 31.03.2021)

4 Projektseite: <https://www.ffmpeg.org> (zuletzt abgerufen: 31.03.2021)

5 Projektseite: <http://sox.sourceforge.net> (zuletzt abgerufen: 31.03.2021)

aber auch heruntergeladen, um sie auf ihren eigenen Rechnern lokal zu verwenden. Um R-Skripte auszuführen, ist es dann jedoch notwendig R⁶ und RStudio⁷ zu installieren. Für die verfügbaren Python-Skripte ist eine Installation von Python 3.1 oder höher Voraussetzung⁸. Außerdem ist ein Editor, mit dem der Python-Code bearbeitet und ausgeführt werden kann, zu wählen. Sowohl der Python-Code als auch der R-Code stehen als Jupyter-Notebook bereit. Um Jupyter-Notebooks bearbeiten zu können, muss neben Python auch das *jupyter*-Paket installiert werden.⁹ R-Code kann in Jupyter-Notebooks bearbeitet werden, wenn der IRKernel eingerichtet ist.¹⁰ Eine kurze Einführung in die Installation der notwendigen Software findet sich auf dem ILIAS-Portal. Um den Studierenden die Installationsarbeit abzunehmen, können sie den bereitgestellten JupyterHub nutzen, dessen detaillierter Aufbau und Funktionsweise in Abschnitt 3 beschrieben wird.

Abbildung 3 zeigt eine Bildschirmaufnahme eines geöffneten Notebooks auf dem JupyterHub. Neben dem R- oder Python-Programmcode können Markdown-Zellen¹¹ eingefügt werden, in denen die Funktionsweise des Codes oder theoretische Grundlagen vor oder nach dem zugehörigen Code erläutert werden. Darüber hinaus werden durch den Code generierte Schaubilder und anderer Output direkt unter Code-Zellen angezeigt, sobald dieser ausgeführt wurde. In den Markdown-Zellen können auch mathematische Formeln in üblicher Latex¹²-Notation eingetragen und dargestellt werden. Die Jupyter-Notebook-Umgebung bietet die Möglichkeit, nicht nur R oder Python Code ausführbar zu halten, sondern kann für über 40 verschiedene Programmiersprachen erweitert werden.

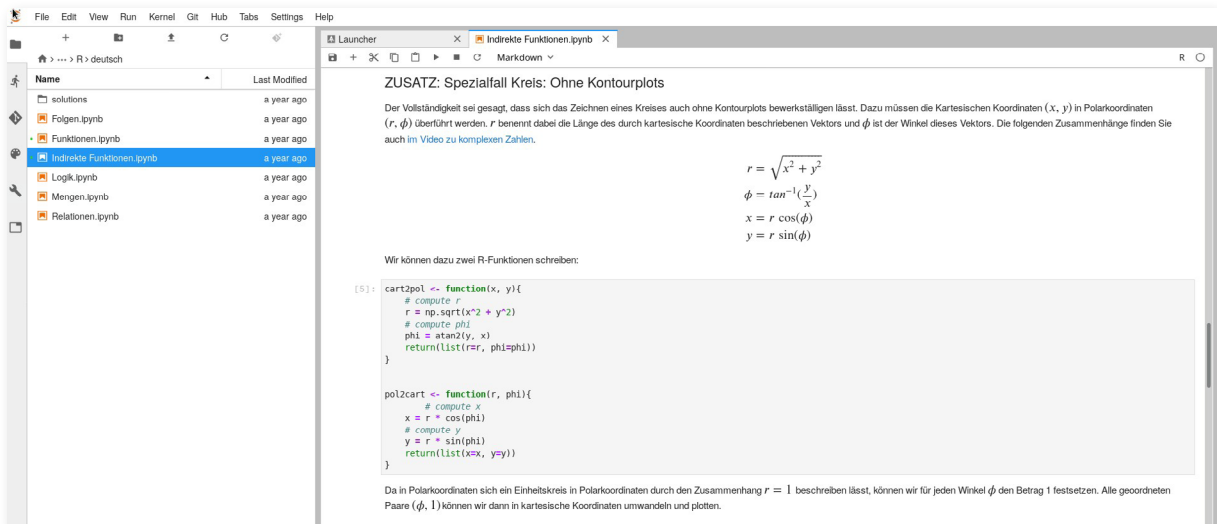


Abb. 3: Bildschirmaufnahme eines Notebooks auf dem JupyterHub

Die einfache Anbindung an die Rechenpower von High-Performance Computing, Datenbanken etc. sowie das einfache Teilen von Ideen zusammen mit erstem Programmcode in allen möglichen Programmiersprachen ma-

- 6 Projektseite: <https://www.r-project.org> (zuletzt abgerufen: 31.03.2021)
- 7 Projektseite: <https://rstudio.com> (zuletzt abgerufen: 31.03.2021)
- 8 Projektseite: <https://www.python.org> (zuletzt abgerufen: 31.03.2021)
- 9 Projektseite: <https://jupyter.org> (zuletzt abgerufen: 31.03.2021)
- 10 Projektseite: <https://irkernel.github.io> (zuletzt abgerufen: 31.03.2021)
- 11 Projektseite: <https://daringfireball.net/projects/markdown> (zuletzt abgerufen: 31.03.2021)
- 12 Projektseite: <https://www.latex-project.org> (zuletzt abgerufen: 31.03.2021)

chen Jupyter-Notebooks attraktiv für die Datenanalyse von bestehenden Datensätzen. Immer mehr verwenden auch namhafte Unternehmen die Software in ihren Entwicklungsabteilungen und fordern entsprechende Qualifikationen in Stellenausschreibungen. Die Studierenden haben im besten Fall bereits in ihrem Mathevorkurs ersten Kontakt zu diesem wichtigen Tool.

3. JupyterHub

Die erste Version des JupyterHubs für das Wintersemester 2017/18 wurde mittels der im Github Repository verfügbaren Grundstruktur eingerichtet und hinter einem Docker-basierten Nginx-Reverse Proxy veröffentlicht.¹³ Grund dafür war vor allem das Hardware-Setup auf dem Projektserver (Idefix¹⁴). Ende des Wintersemesters 2019/20 wurde ein zweiter Server (Obelix) in Betrieb genommen. Die Dopplung an Ressourcen ermöglicht nun den Betrieb des JupyterHubs über ein Kubernetes-Cluster. Das Projekt „From Zero to JupyterHub with Kubernetes“¹⁵ stellt ein voll funktionsfähiges System inklusive einer Schritt-für-Schritt-Anleitung zur Verfügung. Um das Setup an die Nutzungsanforderungen anzupassen bedarf es jedoch einiger Modifikationen. Diese besonderen Anforderungen und die notwendigen Softwarekonfigurationen werden in Abschnitt 3.1 beschrieben. Abschnitt 3.2 beschreibt die Anwendung in der Lehre.

3.1. Softwarekonfiguration

Für die im JupyterHub verfügbare Software (hauptsächlich R inklusive RStudio, Python und Julia) wurde ein Dockercontainer mittels eines eigenen Dockerfiles erstellt. Darin wurden Elemente verschiedener bereits vorhandener Notebookimages aus dem Notebookstack¹⁶ zusammengestellt und erweitert. Abb. 4 stellt die Konfigurationsstruktur dar. Für jeden Service (Gitlab, Matrixspiel, vPCL) wird ein eigenes Containersystem gestartet. Für den JupyterHub besteht dieses Containersystem aus dem JupyterHub selbst, einer für die Funktionsfähigkeit des Hubs wichtigen Nutzerdatenbank und beliebig vielen Notebookservern. Ruft ein Nutzer die Domain `courses.wiso.uni-tuebingen.de` auf, wird seine HTTP-Anfrage über den Nginx Server zum JupyterHub weitergeleitet.

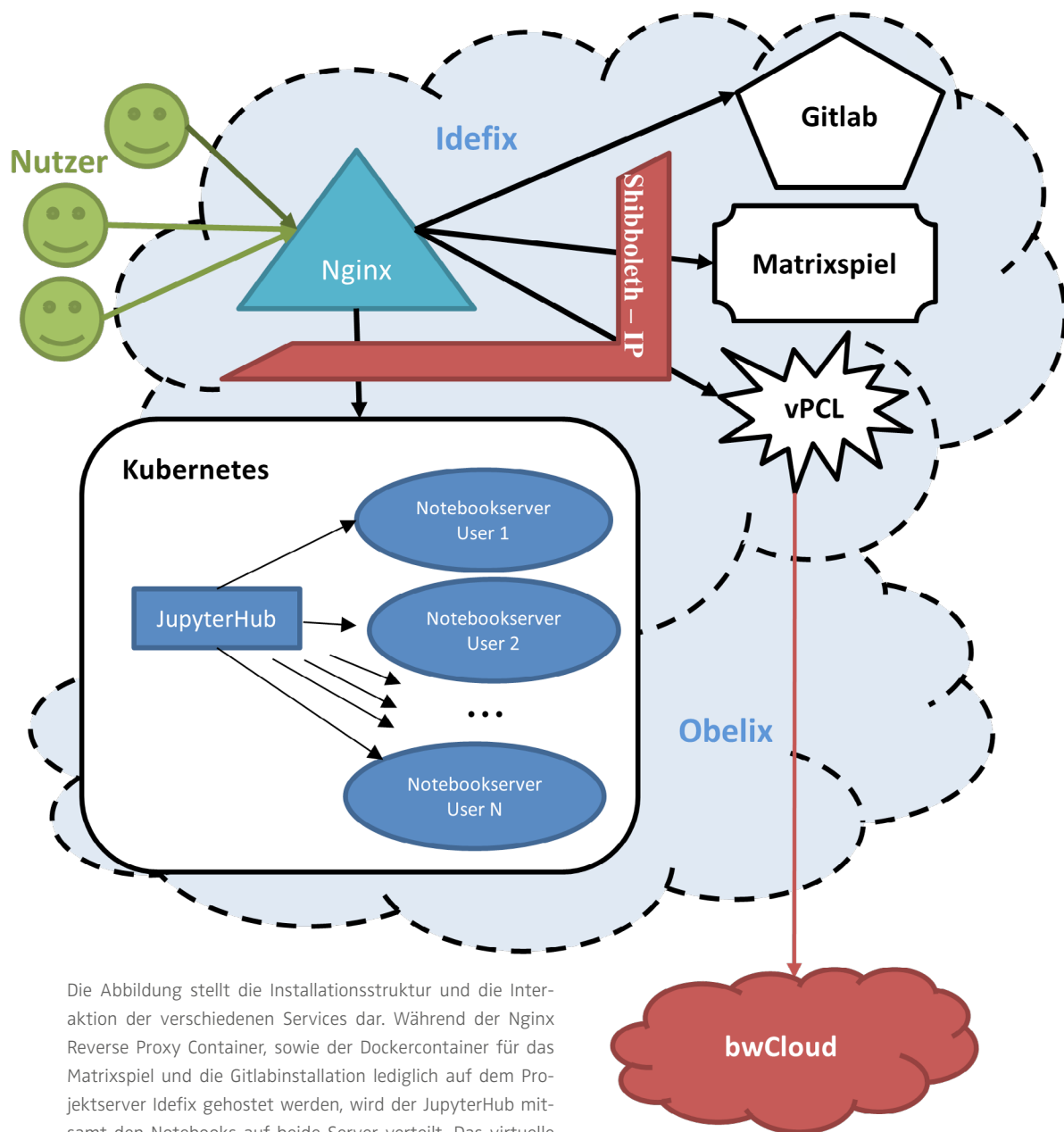
Zunächst muss der Nutzer sich jedoch über den Shibboleth Identity Provider (IP) der Universität Tübingen ausweisen. Die (universitäre) Emailadresse sowie die Fakultätszugehörigkeit des Nutzers werden abgefragt. Für den JupyterHub spielt die Fakultätszugehörigkeit bislang keine Rolle, jedoch für den Zugang zum vPCL. Hier haben nur Mitglieder der Wirtschafts- und Sozialwissenschaftlichen Fakultät Zugang. Die universitäre Emailadresse wird dazu genutzt, die Nutzer im Zweifel kontaktieren zu können und bei einem erneuten Login die von ihnen gespeicherten Dateien und Projekte erneut anzubieten. Hat der Nutzer sich über den Shibboleth-IP der Universität Tübingen authentifiziert, wird er automatisch auf den in einem Kubernetes-Cluster laufenden JupyterHub mit seiner Emailadresse angemeldet. Für den Nutzer wird sodann ein eigener Notebookserver bzw. JupyterLab-Server gestartet. Während des Startvorgangs wird auch der Ordner „shared“ aus einem im Fakultätsgitlab hinterlegten Ordner aktualisiert. Lehrende können ganz einfach Inhalte in diesem Ordner hinterlegen, welche den Studierenden direkt nach erneuter Anmeldung zur Verfügung stehen.

13 Projektseite: <https://github.com/jupyterhub/jupyterhub-deploy-docker> (zuletzt abgerufen: 31.03.2021)

14 Der Projektserver wird liebevoll Idefix genannt. Aus der Abkürzung für den Server aus dem Projekt zu „Initiationswegen in die Wissenschaft“ (iwidw-Server) wurde die besser aussprechbare Abkürzung Idefix. Der zweite Server wurde dann konsistenterweise Obelix getauft.

15 Projektseite: <https://zero-to-jupyterhub.readthedocs.io/en/latest> (zuletzt abgerufen: 31.03.2021)

16 Liste der Jupyter-Notebooks auf DockerHub: <https://hub.docker.com/u/jupyter> (zuletzt abgerufen: 31.03.2021)



Die Abbildung stellt die Installationsstruktur und die Interaktion der verschiedenen Services dar. Während der Nginx Reverse Proxy Container, sowie der Dockercontainer für das Matrixspiel und die Gitlabinstallation lediglich auf dem Projektserver Idefix gehostet werden, wird der JupyterHub mit samt den Notebooks auf beide Server verteilt. Das virtuelle PC-Labor (vPCL) greift zusätzlich noch auf die bwCloud zurück. Die bwCloud kann auch genutzt werden, um dem Kubernetes-Cluster mehr Ressourcen zu verschaffen.

Abb. 4: Konfigurationsstruktur

Innerhalb des JupyterLab-Servers kann der Nutzer nun verschiedene Notebooks mit verschiedenen Kernen starten. Zur Verfügung stehen derzeit Julia, Python, R und Octave. Für jede der Programmiersprachen ist ein Grundstock an üblichen Paketen bereits installiert. Der JupyterLab-Server ist im Grunde jedoch vergleichbar mit einer virtuellen Linux-Maschine. Fehlen dem Nutzer bestimmte features, kann er diese nachinstallieren. Um den einzelnen Nutzern keine Administrator-Rechte einräumen zu müssen, wird auf conda zurückgegriffen. Über `conda install <Paket-Name>` kann der Nutzer individuelle Pakete nachladen. Auch Python's Paketmanager `pip` kann dazu genutzt werden. In R besteht über den Befehl `install.packages()` die Möglichkeit, Pakete zu laden. Generell stehen viele Möglichkeiten zur Verfügung, die es gilt zu entdecken. Mittels des R-shiny Pakets können eigene kleine interaktive Webapps entwickelt werden, um Daten interaktiv

erlebbar zu machen. Außerdem steht ein Latex-Compiler zur Verfügung, länger dauernde Rechenaufgaben können auf dem Server abgeschickt werden und vieles mehr. Die Studierenden sind dazu angehalten, die Möglichkeiten für sich zu entdecken. Git-Repositories können über die Konsole einfach „gepullt“ werden.

3.2. Anwendung in der Lehre

Für die lehrstuhleigenen Vorlesungen „Explorative Datenanalyse“, „Wahrscheinlichkeit & Risiko“ und „Quantitative Methoden“ von Professor Dr. Joachim Grammig wurden neben Skripten und Jupyter-Notebooks in R auch Aufgaben und Erklärvideos produziert, die den Studierenden den Einstieg in statistische Analysesoftware auf dem JupyterHub erleichtern sollen. Auch für das „Mathematische Propädeutikum“, die Erstsemestervorlesung im Bachelor „Mathematische Methoden der Wirtschaftswissenschaft“ und den mathematischen Vorkurs für Master-Studierende von Prof. Dr. Thomas Dimpfl wurden eine Vielzahl an vorlesungsbegleitenden Übungsaufgaben und Skripten erstellt. Für die Vorlesung „Applied Econometrics“ für Bachelor-Studierende wurden ebenfalls weitere Inhalte erarbeitet. Ziel in allen genannten Vorlesungen ist es, einen Zugang zur Programmiersprache auf einfachstem Wege zu ermöglichen.

Für das mathematische Propädeutikum wurde ein kleines Tutorial in R und Python geschrieben. Zusätzlich werden kleine kursbegleitende Inhalte zu verschiedenen Themengebieten entworfen. Die Studierenden können die Aufgaben selbstständig bearbeiten und ihre Lösungen überprüfen. Entsprechende Lösungsnotebooks liegen auf dem JupyterHub zur Selbstkontrolle bereit.

In der Vorlesung „Explorative Datenanalyse“ im ersten Bachelorsemester werden theoretische Inhalte der Vorlesung anhand eines Beispieldatensatzes in R, Stata und SAS umgesetzt. Zusätzlich gibt es kurze Erklärvideos, in denen die einzelnen Skripte vorgestellt werden. Die R-Skripte werden auf dem JupyterHub verfügbar gemacht. Für die proprietären Softwarepakete (Stata und SAS) steht das PC-Labor der Fakultät zur Verfügung. Darüber hinaus können die Studierenden in den vorlesungsbegleitenden Tutorien freiwillig Präsentationen für bestimmte Themen übernehmen. Für die Präsentationen ist ein erster Kontakt mit einem Datenverarbeitungsprogramm notwendig. Die Studierenden können für die Vorbereitung den JupyterHub nutzen. Aus dem Feedback der Tutoren zeigt sich, dass in der überwältigenden Mehrheit der Präsentationen der JupyterHub zum Einsatz kommt. Weiterhin werden in den Tutorien auch Aufgaben besprochen, die mit R umgesetzt wurden. Mit der Möglichkeit, innerhalb eines Jupyter-Notebooks theoretische Inhalte neben praktischem Code darzustellen, können Tutoren in Übungen Aufgaben über den JupyterHub bearbeiten und die vorgestellten Notebooks an die Studierenden weitergeben. Auch innerhalb der Vorlesung wurden bereits Jupyter-Notebooks eingesetzt um zum Beispiel anhand von online verfügbaren Finanzdaten Histogramme zu erstellen. Somit werden den Studierenden nicht nur theoretische Methoden, sondern auch praktische Umsetzungsmuster vermittelt.

Dieser Vermittlungsmodus wird auch in der Erstsemestervorlesung „Mathematische Methoden für Wirtschaftswissenschaftler“ gepflegt, zum Beispiel im Zusammenhang mit Optimierungsverfahren oder Verfahren zur Nullstellenbestimmung, die in live laufendem Code veranschaulicht werden, und vieles mehr.

Im zweiten Semester wird dann der Faden in „Wahrscheinlichkeit und Risiko“ erneut aufgenommen. Über interaktive Shiny-Apps können die Studierenden gleichverteilte Zufallszahlen generieren, zufällige Ziehungen aus verschiedenen Verteilungen vornehmen oder das Monty-Hall-Spiel simulieren. Sie können auch das Grenzverhalten von Verteilungsfunktionen, Erwartungswerten und anderen stochastischen Konzepten erfahren, manipulieren und damit verinnerlichen.

Auch im dritten Semester in der Vorlesung „Quantitative Methoden“ und der Vorlesung „Applied Econometrics“ im vierten Semester im Spezialisierungsbereich des Bachelorstudiums wird der JupyterHub genutzt, um Schätzverfahren wie die Maximum-Likelihood-Methode, die Momentenmethode und andere Konzepte über Notebooks und Shiny-Apps zu vermitteln. Wichtige Konvergenzergebnisse der asymptotischen Theorie werden damit veranschaulicht und sind somit nicht mehr nur trockene Theorie.

An diesem Punkt angelangt, haben die Studierenden mehrere Jahre Erfahrungen gesammelt mit über Notebooks vermitteltem Code. Viele Studierende haben dann bereits einen an der Fakultät angebotenen R- oder Python-Kurs belegt und müssen nicht mehr auf die im JupyterHub verfügbaren Ressourcen zurückgreifen. Dennoch bietet sich der JupyterHub auch in der Lehre in höheren Semestern an, um lauffähigen Code an die Studierenden zu verteilen. Debugging-Problemen, die an unterschiedlichen Systeminstallationen und Betriebssystemen liegen, wird damit aus dem Weg gegangen. Die Studierenden lernen damit zwangsläufig auch, sich mit ihrem eigenen System, ihrem eigenen Computer auseinanderzusetzen und die eigenen Installationsprobleme zu lösen. Missmut über nicht lauffähigen Code wird vermieden. Immerhin können die in Vorlesung und Übungen erlernten Konzepte im JupyterHub ohne weiteres zum Laufen gebracht werden.

4. Online-Spiel zur Einübung linearer Algebra

Um Studierenden einen spielerischen Einstieg in das oft als trocken empfundene Thema der Matrixalgebra oder ganz generell der linearen Algebra zu bieten, haben wir ein kleines Online-Spiel (siehe Abb. 5) entwickelt. Das Spiel besteht darin, dass Studierende die Eigenschaften einer ganzzahligen Matrix ermitteln und die Lösung dann in einer Eingabemaske (siehe Abb. 6) einreichen können. Dazu müssen sich die Studierenden einklicken und über das SingleSignOn-System der Universität Tübingen (Shibboleth) wird die studentische Email hinterlegt. Das Hinterlegen der Email hat zwei Gründe. Zum einen können sich die Studierenden die Matrix des Tages zur Erinnerung an ihre studentische Emailadresse schicken lassen, zum anderen wird jedes Semester ein Preis für den Gewinner eines Rankings ausgelobt. Um die erreichten Punkte zuzuordnen und den Gewinner informieren zu können, ist eine Authentifizierung notwendig.

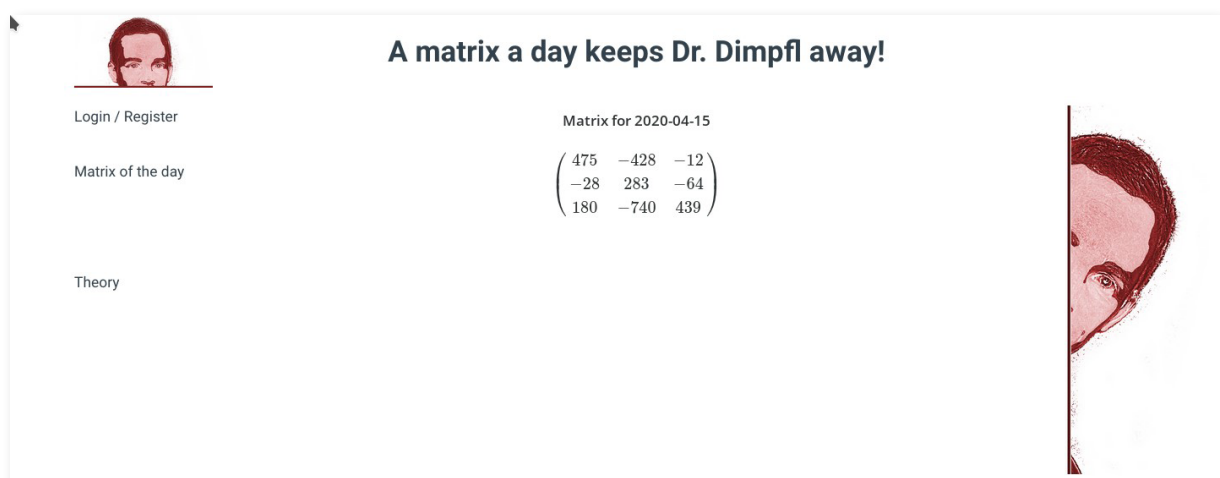



Abb. 5: Startseite des Matrixspiels

Haben die Studierenden sich also eingeloggt und eine Lösung für die Matrix des Tages eingereicht, wird diese mit der richtigen Lösung verglichen und automatisch bewertet. Als kleine, sofortige Belohnung oder – im Fall eines schlechten Ergebnisses – als sofortiger Tadel wird ein kleines GIF angezeigt. Bei schlechten Ergebnissen erscheint auch der Hinweis auf die Theorieseite (siehe Abb. 7). Die Studierenden erhalten Punkte für jede richtige, eingereichte Lösung und können diese Punkte über das Semester hinweg ansammeln. Mit den erzielten Punkten können sie sich in eine Rangliste eintragen lassen. Standardmäßig werden die Studierenden nicht gerankt. Die Studierenden können auch einen eigenen Nutzernamen wählen, der im Ranking erscheinen soll (siehe Abb. 8).



A matrix a day keeps Dr. Dimpfl away!

Logout

Matrix of the day

Archive

Theory

Ranking

My Settings

Matrix for 2020-04-15

$$\begin{pmatrix} 475 & -428 & -12 \\ -28 & 283 & -64 \\ 180 & -740 & 439 \end{pmatrix}$$

When submitting your request below, ...

... stick to Matlab's matrix convention. Separate rows with a semi-colon ; and columns with a comma ,.

... for the eigenvalues submit a row vector, e.g. a 1×3 vector can be written [1,2,3].

... a 1×3 vector can be written [1,2,3], while a 3×1 vector is written [1;2;3].

... the 2×2 matrix $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ in Matlab's matrix convention this would be [1,2;3,4].

... round all decimal numbers with more than four significant digits to four decimal places i.e. $\frac{2}{3} = 0.6667$.

Determinant:

Eigenvalues:

Is the matrix singular? Yes No

Submit solution





Abb. 6: Eingabemaske für die Lösung zur Matrix des Tages



A matrix a day keeps Dr. Dimpfl away!

Logout

Matrix of the day

Archive

Theory

Ranking

My Settings

The determinant

The determinant of a quadratic matrix is a scalar i.e. a number. It is calculated from its entries. By means of the determinant, the invertability of a matrix can be assessed. If $\det \mathbf{A} = |\mathbf{A}| = 0$, the matrix is singular and the inverse of the matrix does not exist. If $\det \mathbf{A} \neq 0$, the matrix is regular and the inverse exists.

Calculation of the determinant

There exist several procedures to calculate the determinant. For 2×2 and 3×3 matrices, the determinant can be computed via simplified calculation schemes.

2×2 matrices

In the case of 2×2 , the determinant is calculated as the difference of the product of the diagonal elements and the anti-diagonal entries of the matrix.

$$|\mathbf{A}| = \det \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

3×3 matrices

For 3×3 matrices, the rule of Sarrus can be used

$$|\mathbf{A}| = \det \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}$$

To easily memorize the rule of Sarrus, it is helpful to extend the matrix on the right with the first and second column. Then, starting with the product from the entries of the main diagonal, one adds up the products of the adjacent...

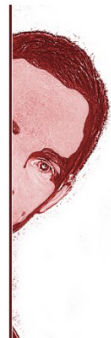


Abb. 7: Homepage mit theoretischen Grundlagen zu Matrizen

A matrix a day keeps Dr. Dimpfl away!

You have 0 points right now.

	Rank	Name	Points
Theory	1.	GibmirdieSchokolade	748
Ranking	2.	warumlogststudichnochein	596
	3.	MatrizenManni	559
My Settings	4.	SarrusSarah	473
	5.	Weihnachtspause	319
	6.	Madeleine	298
	7.	Dennis1893	279
	8.	Knock Knock Knock Penny	255
	9.	FrechesFrüchtchen	242
	10.	Dousnbiersaufn	140

Abb. 8: Rankingseite

Jeden Tag können die Studierenden die gleichen Aufgaben anhand einer neuen Matrix lösen. Damit werden die Studierenden auch dazu animiert, eine gewisse Routine für das tägliche Lernen einzuüben. Die Matrix des Tages kann der Anlass sein, sich an den Schreibtisch zu begeben und sich kurz mit dem mathematischen Lehrstoff zu beschäftigen. Nach dem Motto „steter Tropfen höhlt den Stein“ steht den Studierenden auch keine Übungsumgebung zur Verfügung, in der sie sich beliebig viele Matrizen generieren können. Lediglich die letzten 7 Matrizen werden im Archiv zur Übung angezeigt.

Im Folgenden soll nun das technische Rückgrat der Seite erläutert werden.

4.1. Erzeugung ganzzahliger Matrizen mit ganzzahligen Eigenwerten

Lernziel für die Studierenden ist es, ein Verständnis für Matriceigenschaften, deren Beziehungen und deren Berechnung zu entwickeln. Für beliebige, zufällig generierte 2×2 oder 3×3 Matrizen sind jedoch reelle oder gar komplexe Eigenwerte die Regel. Matrizen mit ganzzahligen Eigenwerten sind selten. Zwar ist der Umgang mit komplexen Zahlen Teil des Curriculums und auch die Bestimmung von Nullstellen mittels Newtonverfahren steht auf dem Programm der Vorlesung „Mathematik für Wirtschaftswissenschaftler“ im ersten Semester, dennoch sollen die Studierenden mit dem Spiel in erster Linie mit den Eigenschaften von Matrizen vertraut gemacht werden.

Die grundlegenden Konzepte erscheinen den Studierenden oft eingängig. In der Klausur kann dann jedoch eine gewisse Lernillusion konstatiert werden. Regelmäßig bereiten die Aufgaben zur linearen Algebra den Studierenden unerwartete Probleme. Das Matrizenspiel hat daher auch den Zweck, die Studierenden in einem Wettbewerb mit verhältnismäßig einfachen Übungen anzulocken und zu motivieren, die Lernillusion mit Stift und Papier zu überwinden.

Für Fragen wie „Was ist die Determinante einer Matrix?“, „Was die Inverse? Wann existiert eine Inverse? Wie wird sie berechnet?“, „Was sind Eigenwerte?“ oder auch „Wie werden Matrizen multipliziert?“ und viele mehr hilft es, anhand einfacher Beispiele die Konzepte selbständig durchzugehen. Sind die Probleme zu schwer gewählt, gibt es computergestützte Verfahren zuhauf, die die Berechnung in wenigen Millisekunden übernehmen. Die händische Berechnung mit Stift und Papier ist jedoch oftmals in der Lage, Lücken im Verständnis zu offenbaren.

Um die Studierenden beim Üben also nicht direkt bei der ersten Matrix, zu deren „Lösung“ sie sich durchgewunden haben, zu entmutigen und sie damit in die Hände schneller computergestützter Lösungen (wie unseres JupyterHub) zu treiben, werden in unserem Matrizenspiel nur ganzzahlige Matrizen mit ganzzahligen Eigenwerten generiert. Hier ist der Einstieg in die Identifikation von Eigenschaften am einfachsten möglich. Eine Bearbeitung ohne numerische Nullstellensuchverfahren und damit endlose Rechenorgien oder komplexen Zahlen ist nicht notwendig.

Um ganzzahlige Matrizen mit ganzzahligen Eigenwerten zu generieren, greifen wir auf die in Campbell und Towse (2016) vorgestellten Grundlagen zurück. Diese und deren Umsetzung in Python sollen in den folgenden Abschnitten kurz dargelegt werden.

4.1.1. Theoretische Grundlagen

Campbell und Towse (2016) präsentieren in ihrem Artikel mehrere Wege, um ganzzahlige Matrizen mit ganzzahligen Eigenwerten zu generieren. Grundlegend ist die Diagonalisierung von quadratischen Matrizen. Campbell und Towse (2016) stellen vier Theoreme bereit, die hierfür von Bedeutung sind.

Angenommen, A sei eine ganzzahlige diagonalisierbare, quadratische Matrix und kann aus PDP^{-1} generiert werden. Dabei sei P eine invertierbare ganzzahlige Matrix, die aus den Eigenvektoren von A besteht, während D eine ganzzahlige Diagonalmatrix mit den Eigenwerten von A auf der Hauptdiagonalen ist:

$$D = \begin{pmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{pmatrix}$$

Für die Determinante soll weiterhin $\det(P) = \delta$ gelten.

Theorem 1

Gegeben zwei n -dimensionale und orthogonale ganzzahlige Vektoren \vec{u} und \vec{v} hat die Matrix $P = I_n - \vec{u} \vec{v}'$ die Determinante $\delta = 1$ und die Inverse $P^{-1} = I_n - \vec{u} \vec{v}'$.

Um ganzzahlige Matrizen zu generieren ist hier wichtig, dass sobald eine Matrix P mit Determinante $\delta = 1$ existiert, jede Auswahl an Eigenwerten in D eingesetzt werden kann um eine Matrix A mit ganzzahligen Eigenwerten zu generieren. Die resultierende Matrix $A = PDP^{-1}$ ist dann eine diagonalisierbare, ganzzahlige Matrix mit ganzzahligen Eigenwerten.

Um P zu ermitteln werden also zu Beginn zwei orthogonale ganzzahlige Vektoren und ihr äußeres Produkt $\vec{u} \vec{v}'$ benötigt. Diese orthogonalen Vektoren zufällig zu ziehen ist jedoch nicht ohne weiteres möglich. Dennoch bietet dieses Theorem, gegeben zweier orthogonaler, ganzzahliger Vektoren oder gegeben einer Matrix P mit $\delta = 1$, eine einfache Möglichkeit, Ganzzahlmatrizen mit ganzzahligen Eigenwerten zu generieren. Dabei müssten lediglich die Eigenwerte auf der Hauptdiagonalen von D verändert werden, um verschiedene Matrizen zu erhalten. Dies würde aber bedeuten, dass die Eigenbasis immer aus denselben Vektoren bestehen würde.

Es stellt sich also die Frage, wie auch Matrizen P mit $\delta \neq 1$ für die Konstruktion der gewünschten ganzzahligen Matrizen mit ganzzahligen Eigenwerten verwendet werden können. In Theorem 2 liefern Campbell und Towse (2016) eine Möglichkeit.

Theorem 2

Sei P eine ganzzahlige Matrix der Größe $n \times n$ mit $\delta \neq 0$. Sei D eine ganzzahlige Diagonalmatrix, deren Einträge zueinander kongruent modulo δ sind. Dann ist $A = PDP^{-1}$ eine ganzzahlige Matrix mit ganzzahligen Eigenwerten.

Matrizen P mit $\delta \neq 1$ sind also für die Konstruktion von Ganzzahlmatrix mit ganzzahligen Eigenwerten geeignet, wenn die Eigenwerte von A kein Vielfaches der Determinante von P sind, sondern kongruent modulo δ . Das heißt, die Differenz von zwei Eigenwerten muss ein Vielfaches der Determinanten sein. Dies erlaubt es, mehr Ganzzahlmatrix zu bestimmen mit ganzzahligen Eigenwerten. Damit können beispielsweise der Eigenwert $\lambda_1 = 1$ und die restlichen Eigenwerte aus der Menge $\{k \in \mathbb{Z} \mid k \delta + \lambda_1\}$ gewählt werden. Damit ist eine Faktorisierung der charakteristischen Gleichung vom Grad 3 (später durch die Studierenden) deutlich vereinfacht.

Theorem 3

Sei P eine ganzzahlige Matrix der Größe $n \times n$ mit $\delta \neq 0$. Sei D eine ganzzahlige Diagonalmatrix, deren Einträge zueinander kongruent modulo δ sind und sei N eine beliebige Matrix, deren Einträge Vielfache von δ sind. Dann ist $B = P(D + N)P^{-1}$ eine ganzzahlige Matrix mit ganzzahligen Eigenwerten.

Die Theoreme 1 und 2 generieren diagonalisierbare Matrizen mit ganzzahligen Eigenwerten. Aber auch nicht diagonalisierbare Matrizen mit ganzzahligen Eigenwerten sollen erzeugt werden. Um dies zu erreichen wird N zu D addiert, deren von 0 verschiedene Einträge jeweils ein Vielfaches der Determinanten von P sind. Die Summe der beiden Matrizen wird dann in der Operation $P(D + N)P^{-1}$ verwendet, um eine nicht diagonalisierbare Matrix B zu generieren. Mit Hilfe dieses Theorems ist es möglich, unterschiedliche Matrizen zu generieren, deren Eigenschaften dennoch verhältnismäßig einfach zu berechnen sind. Nicht diagonalisierbare Matrizen dienen dazu, die Aufgabenstellung weiter zu variieren.

Der Vollständigkeit halber soll auch Theorem 4 erwähnt werden, das zu allen möglichen ganzzahligen Matrizen mit ganzzahligen Eigenwerten führt, allerdings beschränkt auf Matrizen der Dimension 2×2 .

Theorem 4

Sei P eine vereinfachte 2×2 Matrix, deren Eigenwerte kongruent modulo $k = \det(P)$ sind. Dann und nur dann ist $A = PDP^{-1}$ eine ganzzahlige Matrix mit ganzzahligen Eigenwerten.

4.1.2. Implementierung in Python

Zum Generieren der Matrizen wurde eine Funktion in Python geschrieben. Als Input wird sowohl die Dimension n der quadratischen Matrix als auch die Wahrscheinlichkeit x , mit der die resultierende Matrix singular sein soll, vorgegeben.

Um zwischen singulären und regulären Matrizen zu variieren wird zunächst eine Zufallszahl u aus einer im Intervall $[0,1]$ gleichverteilten Zufallsvariablen \tilde{U} gezogen. Ist u kleiner als die zuvor vorgegebenen Wahrscheinlichkeit x soll die zu generierende Matrix singular sein.

Der dazugehörige Code ist:

```
# Draw a random number that determines the type
u = np.random.rand()
# Is the random number smaller than our probability X
# --> singular
thetype = u<=X
```

In jedem Fall wird für den Algorithmus von Campbell und Towse (2016) eine reguläre Ganzzahlmatrix P benötigt, die mit folgenden Zeilen generiert wird.

```
# Generate a random matrix P
P = np.random.randint(-5,5, size=(n,n))
# Get the determinant of P
delta = np.linalg.det(P)

# Check whether we have a singular matrix
is_singular = delta==0

# We cannot accept P to be a singular matrix
while is_singular:
    # Generate a random matrix P
    P = np.random.randint(-5,5, size=(n,n))

    # Get the determinant of P
    delta = np.linalg.det(P)

    # Check whether we have a singular matrix
    is_singular = delta==0
```

Hierfür wird eine Diagonalmatrix D erstellt, die aus zufällig generierten Faktoren multipliziert mit der Determinante von P besteht. Schließlich wird die Matrixoperation PDP^{-1} durchgeführt, wodurch die Matrix A mit den gewünschten Dimensionen $n \times n$ generiert wird.

```
# So now we have a convenient matrix
# Should our matrix be singular?
if thetype: # yes
    factors = np.random.randint(-10,10,size=n)
    pos = np.random.randint(0,n,size=1)
    rest = -factors[pos]*delta
else: #no
    # Get the factors
    factors = np.random.randint(-10,10,size=n)
    rest = np.random.randint(-10,10,size=1)
    while (-rest in factors*delta):
        rest = np.random.randint(-10,10,size=1)

# Set the eigenvalues
D = np.diag(factors)*delta + rest*np.identity(n)
# Invert P
A = (P @ D) @ np.linalg.solve(P, np.identity(n))
return(np.round(A).astype("int"))
```

Die ausgegebene Matrix A kann dann von den Studierenden bearbeitet werden. Wir beschränken uns absichtlich auf Matrizen der Dimensionen 2×2 oder 3×3 , um den Rechenaufwand mit Stift und Papier in einem sinnvollen Rahmen zu halten. Auch in der Klausur würde keine höherdimensionierte Matrix zur händischen Berechnung herangezogen werden.

4.2. Backend der WebApp

Das Backend der WebApp wurde mittels des Python Tornado-Pakets¹⁷ umgesetzt. Die WebApp läuft in einem Docker-Container, während die für das Funktionieren notwendigen Daten in einer PostgreSQL-Datenbank gespeichert werden, die in einem weiteren Docker-Container läuft. Für jede aufgerufene Seite wurde ein Handler geschrieben. Jeder Handler setzt die ankommenden HTTP-Requests (GET, POST) um und setzt die variablen Inhalte auf verschiedene Templates.

```
handlers = [  
    (r"/$", RestHandler),  
    (r"/"+self.base_url+"$", MatrixHandler),  
    (r"/"+self.base_url+"/matrix-of-the-day",  
     MatrixHandler),  
    (r"/"+self.base_url+"/theory", TheoryHandler),  
    (r"/"+self.base_url+"/settings", SettingsHandler),  
    (r"/"+self.base_url+"/ranking", RankingHandler),  
    (r"/"+self.base_url+"/archive", ArchiveHandler),  
    (r"/"+self.base_url+"/delete_me", DeleteHandler),  
    (r"/"+self.base_url+"/auth/login", AuthLoginHandler),  
    (r"/"+self.base_url+"/auth/logout", AuthLogoutHandler),  
    (r"/"+self.base_url+"/static/(.*)",  
     tornado.web.StaticFileHandler,  
     {'path': './static/' } ),  
    (r"/"+self.base_url+"/.*", CatchAllHandler)  
]
```

Der Handler für den Login greift die per Shibboleth übermittelte, studentische Emailadresse ab und speichert diese zusammen mit einem standardisierten Nutzernamen in der PostgreSQL-Datenbank ab. Außer der studentischen Emailadresse, um die Nutzer kontaktieren zu können, werden keine personenbezogenen Daten gespeichert. In der PostgreSQL-Datenbank werden auch die abgegebenen Lösungen sowie die erreichten Punkte gespeichert. Die Nutzer können alle abgegebenen Lösungen und ihre Daten aus der Datenbank jederzeit über das Menü der persönlichen Settings mittels DeleteMe-Handler löschen. In den Settings können die Nutzer auch ihren Nutzernamen, die Emailerinnerung sowie die Einstellungen zum Ranking ändern. Standardmäßig sind das Ranking und die Emailerinnerung ausgeschaltet.

Über den Theory-Handler wird eine Seite mit den theoretischen Grundlagen aufgerufen. Im Hintergrund wird in der Eventloop des Tornado-Webserver eine Funktion ausgeführt, die überprüft, ob für den aktuellen Tag bereits eine Matrix generiert wurde. Für den Fall, dass die Matrix noch nicht existiert, wird eine Matrix generiert

17 <https://www.tornadoweb.org/en/stable> (zuletzt abgerufen: 31.03.2021)

und wird ebenfalls in der Datenbank abgespeichert. Die generierte Matrix wird sodann über den Emailverteiler versandt. Dies erfordert das Hinterlegen von Authentifizierungsinformationen einer universitären Emailadresse und wir verwenden daher eine Funktions- und keine persönliche Emailadresse.

5. Fakultätsgitlab

Für die Integration von Lehrmaterialien in den JupyterHub wurde ein Fakultätsgitlab über die opensource Community Edition eingerichtet.¹⁸

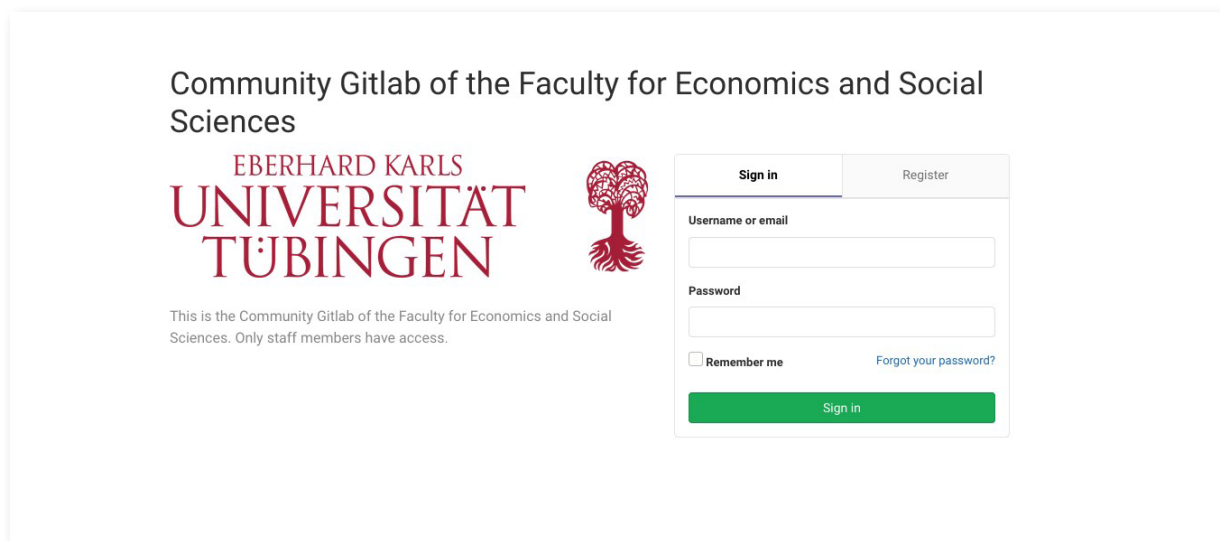


Abb. 9: Landingpage des WiSo-Gitlabs

Lehrende können sich selbstständig mit ihrer Mitarbeiter-Emailadresse einloggen. Nach Rücksprache mit dem JupyterHub-Administrator können Lehrende dann dem GIT-Projekt zugeordnet werden, in dem die Dateien auf dem JupyterHub verwaltet werden. Änderungen in diesem Projekt werden bei jedem Neustart der Notebookserver der einzelnen Nutzer aktualisiert. Möglich macht dies das im JupyterHub installierte Modul nb-gitpuller¹⁹. So können Lehrende, die den JupyterHub in Ihre Vorlesung einbinden wollen, ihre Lehrmaterialien allen Studierenden einfach über das Fakultätsgitlab freigeben.

Das Bereitstellen von Inhalten über das Fakultätsgitlab kann auch über das Anlegen eines GIT-Projekts auf den Servern von Gitlab, Github oder Bitbucket erfolgen. Mit den entsprechenden Zugriffsrechten können Studierende sich die Inhalte in den Nutzercontainer auf dem JupyterHub ziehen. Die Verwendung des Versionierungstools git findet in vielen Bereichen immer weitere Verwendung. Die Möglichkeiten für eine verbesserte, effizientere Zusammenarbeit, die das Versionierungstool bietet, sind für alle Beteiligten, Lehrende als auch Studierende, vorteilhaft.

Um beispielsweise die Covid-19-Datensätze des Zentrums für System- und Ingenieurwissenschaften an der Johns Hopkins Universität in den Notebookserver in den Ordner **corona** zu laden, können im Terminal die folgenden Befehle abgesetzt werden.

18 Projektseite: <https://gitlab.com/gitlab-org/gitlab> (zuletzt abgerufen: 31.03.2021)

19 Projektseite: <https://github.com/jupyterhub/nb-gitpuller> (zuletzt abgerufen: 31.03.2021)

```
git clone https://github.com/CSSEGISandData/COVID-19.git corona
```

Schon können die Daten genutzt und in R oder Python eingelesen werden. Um den Datensatz am nächsten Tag um die aktuellen Zahlen zu erneuern, muss der Datensatz nicht erneut geklont werden, sondern kann mit folgenden Befehlen auf den neusten Stand gebracht werden.

```
cd corona && git pull
```

Sofern der Projektbesitzer Schreibrechte für ein GIT-Repository gewährt hat, können mittels **git push** und einer eingerichteten SSH-Verbindung auch Änderungen in das Projekt-Repository geschrieben werden. Die weitreichenden Möglichkeiten, in Gitlab Projekte zu organisieren und Aufgaben zu delegieren, bieten weitere Vorteile.

Alternativ können Lehrende daher das Fakultätsgitlab auch nutzen, um Forschungs- oder Lehrinhalte anderen Projektbeteiligten bereitzustellen, Projekte zu koordinieren oder Studierenden größere Dateien zur Verfügung zu stellen. Der Vorteil von Daten auf einem universitären Rechner (insbesondere in Bezug auf Datenschutz bei sensiblen Daten) mit allen Vorteilen der sonst auf Fremdservern gehosteten Gitlab-Umgebung liegt auf der Hand. Gerade in der Coronakrise im Sommersemester 2020 wurde das Fakultätsgitlab auch genutzt, um große Videodateien Studierenden einfach zur Verfügung zu stellen. Da zu Beginn der Pandemie nicht abzusehen war, ob die bestehende Universitätsinfrastruktur (ILIAS und Moodle) eine zeitweilig erhöhte Nachfrage bewältigen könne, wurde für Lehrende der Wirtschafts- und Sozialwissenschaftlichen Fakultät die Verteilung der Videodateien an die Studierenden über das Gitlab zusätzlich abgesichert.

6. vPCL: Virtuelles PC-Labor

Während auf dem JupyterHub lediglich Open-Source-Software zur Verfügung steht, wird von Studierenden in höheren Semestern auch der ortsungebundene Zugang zu proprietärer lizenzierter Software wie beispielsweise Stata nachgefragt. Diese Software kann den Studierenden auf den Maschinen im PC-Labor zur Verfügung gestellt werden. Dort kollidiert die Nutzung der Maschinen vor Ort jedoch oft mit dem vielfältigen Kursangebot der Fakultät.

Seit Anfang 2020 kann den Studierenden auch über virtuelle Maschinen auf der bwCloud die entsprechende Software zur Verfügung gestellt werden. Damit kann die Nachfrage von Studierenden, aber auch Mitarbeitern mit größeren Projekten, nach interaktiven Rechenkapazitäten, die mehrere Wochen verfügbar sein sollen, gedeckt werden. Solche interaktiven Rechenkapazitäten sind über die High-Performance-Cluster nicht immer uneingeschränkt verfügbar. Auch finden viele Nutzer die terminalbasierte Interaktion mit dem Cluster wenig zugänglich. Daher haben wir einen Managementservice geschrieben, der virtuelle Maschinen auf Anfrage in der bwCloud²⁰ generiert und einrichtet. Die Webapplikation kontrolliert auch die Laufzeiten der Maschinen, sodass spätestens nach 30 Tagen (außer der Nutzer drückt den Verlängerungsbutton) die Ressourcen wieder freigegeben werden. Zugriffen wird auf die virtuellen Maschinen per SSH oder per Remotedesktop (Windows RDP oder x2go²¹). Als Verbindungsclient wird auf Windowsmaschinen Bitvise²² empfohlen.

20 Projektseite: <https://www.bw-cloud.org> (zuletzt abgerufen: 31.03.2021)

21 Projektseite: <https://wiki.x2go.org/doku.php> (zuletzt abgerufen: 31.03.2021)

22 Projektseite: <https://www.bitvise.com> (zuletzt abgerufen: 31.03.2021)

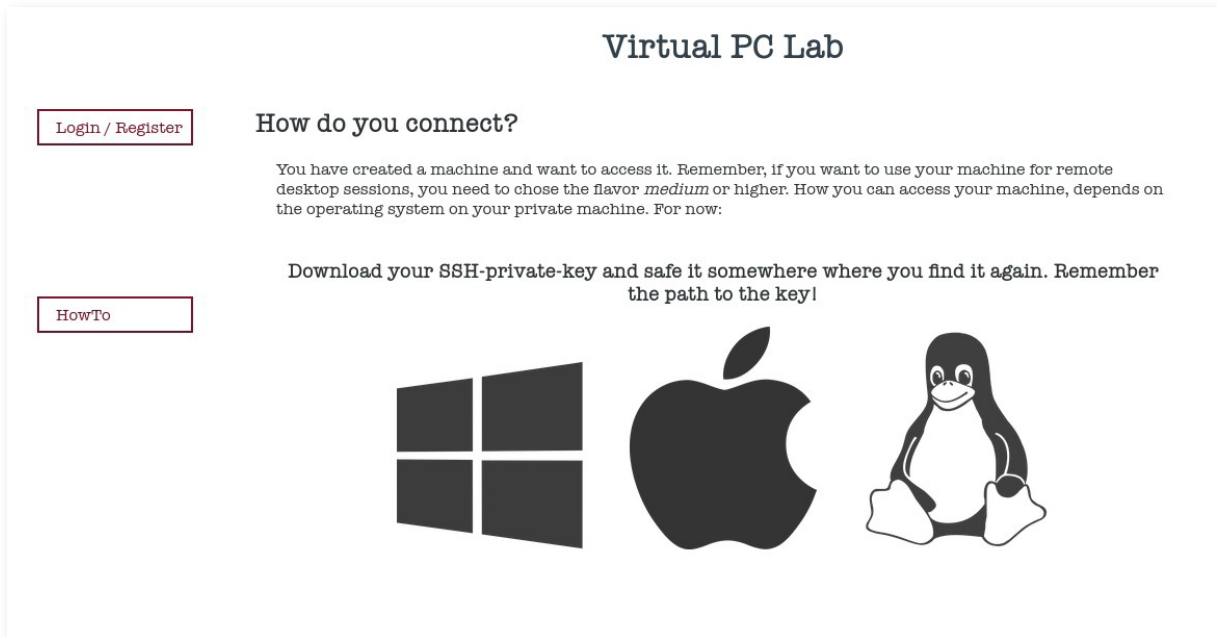


Abb. 10: Landing Page - Virtuelles PC Labor

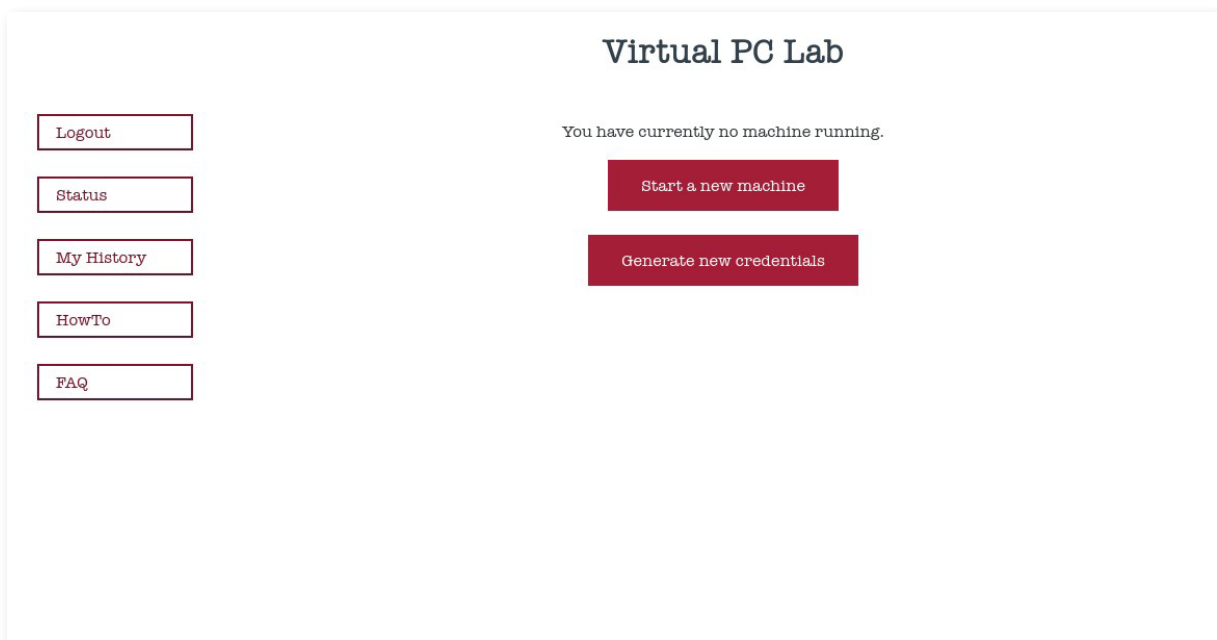


Abb. 11: vPCL nach Login

Die bwCloud bietet Serverinfrastruktur als Dienstleistung an, basierend auf einem Openstack²³. Mittels sogenannter Heat-Templates können Konfigurationsstrukturen für die Serverinfrastruktur festgelegt werden. So können mehrere Systeme über eine REST-API mit wenigen Befehlen aus Python heraus oder über ein entsprechendes Kommandozeilentool orchestriert werden. Ein Mustertemplate findet sich in Appendix B.

23 Projektseite: <https://www.openstack.org> (zuletzt abgerufen: 31.03.2021)

Darin werden das Image mit dem entsprechenden Betriebssystem sowie weitere zusätzliche Volumen zum Speichern von Daten oder beispielsweise zum Einbinden von Software angespielt. Die entsprechende Software muss einmalig auf einem Musterimage installiert werden und kann dann dynamisch kopiert und in die Maschinen verschiedener User eingebunden werden. Als Betriebssystem wurde für die virtuellen Maschinen OpenSuse 15.0 gewählt. In den Maschinen haben die Nutzer vollen root-Zugriff und können die Maschinen nach ihren Wünschen gestalten.

7. Erste Evaluationsergebnisse

Mit dem Projekt haben wir seit dem Wintersemester 2017/18 1200 Studierende in den Präsenzkursen erreicht. Mindestens 800 Studierende kannten und nutzten die Online Videos, mehr als 400 Studierende nutzen das Angebot, über den Server programmieren zu lernen oder vorlesungsbegleitende Übungen des ersten Semesters zu bearbeiten. 250 Studierende nahmen an unserem Matrix-Spiel teil. Die Wahrnehmung auf Seiten der Studierenden ist durchweg positiv. Dies zeigt sich auch darin, dass das Projekt mit dem Preis für Innovative Lehre der Fachschaft des Fachbereichs Wirtschaftswissenschaften, dem Wiwi-Impuls 2019, und auf Antrag der Studierenden mit dem Lehrpreis der Universität Tübingen 2019 ausgezeichnet wurde.

Beim Haupttermin 2018/19 der Vorlesung „Mathematische Methoden der Wirtschaftswissenschaft“ sank die Durchfallquote auf 18,75%. Insgesamt schienen in der ersten Kohorte die computerbasierten Angebote einen positiven Effekt auf die Leistung der Studierenden in den Fächern „Mathematische Methoden der Wirtschaftswissenschaft“, „Explorative Datenanalyse“, „Wahrscheinlichkeit und Risiko“ sowie „Quantitative Methoden“ zu haben. Eine exakte Quantifizierung dieser Wahrnehmung gestaltet sich jedoch schwierig. Zwar machte sich in Mathematik insbesondere im Vergleich der Klausurergebnisse zum Vorjahr eine Verbesserung bemerkbar, jedoch ist der einmalige Vergleich nicht aussagekräftig. Im Wintersemester 2018/2019 wurden in der Klausur vier Fragen mit Bezug zur linearen Algebra gestellt, die direkt mit dem Online-Spiel täglich geübt werden konnten. Insgesamt haben im Haupttermin im Wintersemester 2018/19 rund 80% der Studierenden diese Aufgaben korrekt gelöst. Eine vergleichbare (technisch identische) Aufgabe aus der Nachklausur des Wintersemesters 2017/18 wurde hingegen nur von ca. 50% der Studierenden richtig gelöst. Ein Teil dieses Erfolgs könnte auf die Möglichkeit des intensiven Übens dieser Thematik mittels unseres Online-Spiels zurückzuführen sein. Selbstverständlich ist dabei nicht ausgeschlossen, dass alleine die Existenz des Spiels die Wichtigkeit des Themenblocks „Lineare Algebra“ an sich verdeutlicht und damit zu einer Zunahme der Lernanstrengungen geführt hat. Beides sind jedoch willkommene Effekte, die am Ende den Studierenden auch durch bessere Noten (besserer Durchschnitt und geringere Durchfallquote Wintersemester 2018/19 im Vergleich zu Wintersemester 2017/18) zu Gute kommen. Dieses Ergebnis sollte jedoch nicht überbewertet werden. Eine kausale Analyse über den direkten Effekt der Angebote ist nicht ohne weiteres möglich und wurde bisher nicht unternommen. Zudem war das Üben mit ganzzahligen Matrizen erst ab dem Wintersemester 2019/20 möglich, im Wintersemester 2018/19 waren im Matrizenspiel noch komplexe Einträge möglich. Das Spiel war damit in einer anderen Form zugänglich als derzeit.

Auch wenn ein Vergleich über die Jahre hinweg schwer fällt, da die Aufgabenstellungen variieren (auch wenn die nötigen Techniken immer die gleichen bleiben), scheint in Bezug auf die Lineare Algebra dennoch ein positiver Effekt vorzuherrschen. Ein direkter Vergleich über die Klausuren mehrerer Kohorten ist über die Aufgabe zur Berechnung der Inversen einer 2×2 oder 3×3 Matrix möglich. Die nachfolgende Tabelle zeigt die Raten der Studierenden, die eine solche (Teil-)Aufgabe überwiegend richtig beantwortet, d.h. mehr als 60% der in der (Teil-)Aufgabe vergebenen Punkte erreicht hatten.

	NT	HT	NT	HT	NT	HT	NT	HT	NT	HT
	14/15	15/16	15/16	16/17	16/17	17/18	17/18	18/19	18/19	19/20
2 x 2	–	–	–	–	–	–	61%	63%	–	–
3 x 3	59%	56%	56%	56%	75%	–	–	–	75%	85%

Die 85% im Haupttermin des WS 2019/20 können durchaus als bemerkenswert angesehen werden, immerhin wurde bei keinem Haupttermin zuvor ein solches Ergebnis erzielt. Eine detaillierte Analyse des Übungsverhaltens der Studierenden im WS 2019/20 wurde von Schwerter et al. (2021) durchgeführt. Es zeigt sich, dass sich eine erfolgreiche Teilnahme am Matrixspiel für die Studierenden mit Blick auf das Klausurergebnis lohnt. Dies gilt vor allem dann, wenn die Studierenden solange übten bis eine weitgehend korrekte Lösung erreicht wurde. Die durchgeführte Quantilsanalyse zeigt auch, dass gerade für Studierende am unteren Ende der Notenverteilung eine Teilnahme vorteilhaft war.

Generell ist jedoch auch die subjektive Wahrnehmung der Studierenden wichtig. Inwieweit werden die Angebote als hilfreich und verständlich angesehen? In der freiwilligen Onlinebefragung zum Mathevorkurs gab die Mehrheit der teilnehmenden Studierenden an, dass die im Rahmen des Mathevorkurses eingerichteten Dienste (Videos und JupyterHub) hilfreich seien und die Inhalte verständlich vermittelt wurden. Insgesamt bewerteten sie den Kurs überwiegend positiv.

8. Fazit

Alles in allem zeigt das Projekt eine Reihe von Möglichkeiten auf, wie klassische Lehrformate digital, computerbasiert und remote unterstützt werden können. Wir haben im Rahmen des Strukturmodelleprojekts zwei Server beschafft und einen JupyterHub, ein Gitlab, ein Matrixspiel und ein virtuelles PC-Labor darauf eingerichtet. Der JupyterHub läuft auf einem Kubernetes-Cluster auf den beiden Servern und bietet damit Übungskapazitäten in Hörsaalstärke. Wir haben all diese Dienste mit Inhalten befüllt und damit vorlesungsbegleitende Unterstützung und Material für mehrere Lehrveranstaltungen erstellt. Darüber hinaus haben wir den Mathevorkurs durch Videos zu 24 Themenbereichen jeweils auf Deutsch und Englisch erweitert und weitere 140 Übungsaufgaben inklusive Lösungen erarbeitet. Weiterhin haben wir Kurtutorials zu R und Python geschrieben. Angefangen vom mathematischen Propädeutikum bis zur Vertiefungsvorlesung im vierten Semester und darüber hinaus bietet der Server die Möglichkeit, niederschwellig an jedem Punkt während des Studienverlaufs in die computergestützte Datenverarbeitung mittels R und Python einzusteigen. Ebenfalls haben wir einen Anmeldedienst für ein virtuelles PC-Labor für 30 virtuelle Arbeitsplätze auf der bwCloud geschrieben. Damit steht den Lehrenden der Fakultät eine Auswahl an Tools zur Verfügung, um Daten in den Hörsaal zu bringen. Die Studierenden profitieren in mehrerlei Hinsicht: Zum einen werden sie ab dem ersten Semester an Datenarbeit und die entsprechende Software herangeführt. Zum anderen werden die ab und an als trocken empfundenen methodischen Einführungsveranstaltungen mit Leben gefüllt. Wir wollen bei den Studierenden den Spaß an eigenständigem, empirischem Arbeiten wecken und fördern. Mit dem beschriebenen Projekt haben wir sowohl die technischen Voraussetzung geschaffen als auch einen ersten, großen Schritt in diese Richtung getan, um schließlich die Studierenden „data literate“ in die Zukunft zu entlassen.

Abbildungsverzeichnis

- Abb. 1 Erste Seite des Mathematischen Propädeutikums
- Abb. 2 ILIAS Seite mit Video
- Abb. 3 Bildschirmaufnahme eines Notebooks auf dem JupyterHub
- Abb. 4 Konfigurationsstruktur
- Abb. 5 Startseite des Matrixspiels
- Abb. 6 Eingabemaske für die Lösung zur Matrix des Tages
- Abb. 7 Homepage mit theoretischen Grundlagen zu Matrizen
- Abb. 8 Rankingseite
- Abb. 9 Landingpage des WiSo-Gitlabs
- Abb. 10 Landingpage des virtuellen PC Labors
- Abb. 11 vPCL nach Login

Literatur

Campbell, E. / Towse, C. (2016): Constructing Integer Matrices with Integer Eigenvalues. *Mathematical Scientist* 41(1).

Felder, R. M. (1999): Memo to students who are disappointed with their last test grade. *Chemical Engineering Education* 33(2), S. 136-137.

Schwerter, J. / Dimpfl, T. / Bleher, J. / Murayama, K. (2021): Practice makes perfect? Self-testing with external rewards. Available at SSRN: <https://ssrn.com/abstract=3830777>

Sydsæter, K., et al. (2013): *Mathematik für Wirtschaftswissenschaftler. Basiswissen mit Praxisbezug.* München, Pearson.

Anhang A – Python Code zum Generieren von ganzzahligen Zufallsmatrizen

```
def Matrix(n,X):
    # This function takes
    # n -- the dimension of the resulting matrix
    # X -- the probability with which the resulting
    #       matrix should be singular
    #
    # based on Christopher Towse and Eric Campbell's Algorim (2016)
    # "Constructing Integer Matrices with Integer Eigenvalues"
    # published in Applied Probability Trust

    if n < 2 or n > 5:
        print ("The dimension of the matrix
                must be between 2 and 5")
        return

    # Draw a random number that gives the action (a scalar)
    u = np.random.rand()
    # Is the random number smaller than our probability
    # --> singular
    thetype = u<=X
    # Generate a random matrix P
    P = np.random.randint(-5,5, size=(n,n))

    # Get the determinant of P
    delta = np.linalg.det(P)

    # Check whether we have a singular matrix
    is_singular = delta==0

    # If we have a singular matrix, we need one that is invertible
    while is_singular:
        # Generate a random matrix P
        P = np.random.randint(-5,5, size=(n,n))

        # Get the determinant of P
        delta = np.linalg.det(P)

        # Check whether we have a singular matrix
        is_singular = delta==0
```

```

# So now we have a convenient matrix

# Should our matrix be singular?
if thetype: # yes
    factors = np.random.randint(-10,10,size=n)
    pos = np.random.randint(0,n,size=1)
    rest = -factors[pos]*delta
else: #no
    # Get the factors
    factors = np.random.randint(-10,10,size=n)
    rest = np.random.randint(-10,10,size=1)
    while (-rest in factors*delta):
        rest = np.random.randint(-10,10,size=1)

# Set the eigenvalues
D = np.diag(factors)*delta + rest*np.identity(n)
# Invert P
A = (P @ D) @ np.linalg.solve(P, np.identity(n))
return(np.round(A).astype("int"))

```

Anhang B – HEAT-template

```
heat_template_version: "2018-08-31"
description: "version 2017-09-01"

parameters:
  username:
    type: string
    label: User Name
    description: This is the user name, and will be also the name of the
key and the server as referenced in My::Server::Key
    default: test
  imagename:
    type: string
    label: Image Name
    description: This is the Name of the Image e.g. Ubuntu 18.04
    default: "OpenSUSE Leap 15"
  ssh_pub_key:
    type: string
    label: ssh public key
    description: This is the public key with which the machine can be acce
ssed

  flavorname:
    type: string
    label: Flavor Name
    description: This is the Name of the Flavor e.g. m1.small
    default: "m1.small"
  vol_size:
    type: number
    label: Volume Size
    description: This is the size of the volume that should be attached in
GB
    default: 10
  password:
    type: string
    label: password
    description: This is the su password and user password

resources:

  init:
    type: OS::Heat::SoftwareConfig
```

```

properties:
  group: ungrouped
  config:
    str_replace:
      template:
        {get_file: init_image.sh }
    params:
      $USERNAME: {get_param: username}
      $SSHPUBKEY: {get_param: ssh_pub_key}
      $PASSWD: {get_param: password}
      $VOLDATA: {get_resource: volume}
      $STATAVOL: {get_resource: stata_vol}
      $CONDAVOL: {get_resource: conda_vol}

my_key:
  type: "OS::Nova::KeyPair"
  properties:
    name:
      list_join:
        ["_", [ {get_param: username}, 'key']]
#   save_private_key: True
    public_key: {get_param: ssh_pub_key}

my_server:
  type: "OS::Nova::Server"
  properties:
    #block_device_mapping_v2: [{ device_name: "vda", image : { get_param :
imagename }, delete_on_termination : "false", volume_size: 20 }]
    image: {get_param: imagename }
    name: {get_param: username}
    flavor: {get_param: flavorname}
    key_name: {get_resource: my_key}
    admin_pass: {get_param: password}
    user_data_format: RAW
    user_data: {get_resource: init}
    networks:
      - network: "public-belwue"

depends_on:
  - my_key
  - init
  - volume
  - stata_vol
  - conda_vol

volume:
  type: "OS::Cinder::Volume"

```

```

properties:
  # Size is given in GB
  size: {get_param: vol_size}
  name:
    list_join: ["-", ["vol_",{get_param: username }]]
volume_attachment:
  type: "OS::Cinder::VolumeAttachment"
  properties:
    volume_id: { get_resource: volume }
    instance_uuid: { get_resource: my_server }
  depends_on:
    - volume

stata_vol:
  type: "OS::Cinder::Volume"
  properties:
    image: "<IMAGE-ID>"
    size: 3
stata_attachment:
  type: "OS::Cinder::VolumeAttachment"
  properties:
    volume_id: { get_resource: stata_vol }
    instance_uuid: { get_resource: my_server }
  depends_on:
    - stata_vol

conda_vol:
  type: "OS::Cinder::Volume"
  properties:
    image: "<IMAGE-ID>"
    size: 5
conda_attachment:
  type: "OS::Cinder::VolumeAttachment"
  properties:
    volume_id: { get_resource: conda_vol }
    instance_uuid: { get_resource: my_server }
  depends_on:
    - conda_vol

outputs:
  instance_ip:
    description: The IP address of the deployed instances
    value: { get_attr: [my_server, first_address] }
  private_key:
    description: The private key to access instance
    value: { get_attr: [my_key, private_key] }

```

Autoren

Johannes Bleher, MSc.

Eberhard Karls Universität
Wirtschafts- und Sozialwissenschaftliche Fakultät
Abteilung Statistik, Ökonometrie und empirische Wirtschaftsforschung

Sigwartstraße 18
72076 Tübingen
johannes.bleher@uni-tuebingen.de
www.uni-tuebingen.de/de/18867

Prof. Dr. Thomas Dimpfl

Eberhard Karls Universität
Wirtschafts- und Sozialwissenschaftliche Fakultät
Abteilung Statistik, Ökonometrie und empirische Wirtschaftsforschung

Sigwartstraße 18
72076 Tübingen
Tel. +49 (0) 70 71-29 76 417
Fax. +49 (0) 70 71-29 50 13
thomas.dimpfl@uni-tuebingen.de
www.uni-tuebingen.de/de/18867

Herausgeberinnen der Reihe TBHD

Dr. Andrea Fausel

Wissenschaftliche Mitarbeiterin und stellvertretende Leiterin der Arbeitsstelle Hochschuldidaktik

Dipl.-Päd. Regine Richter

Leiterin der Arbeitsstelle Hochschuldidaktik

Anschrift der Redaktion

Eberhard Karls Universität Tübingen
Dezernat III – Studium und Lehre
Arbeitsstelle Hochschuldidaktik
Sigwartstr. 20
72076 Tübingen
Tel. +49 (0) 7071 29-78385
Fax +49 (0) 7071 29-5615
hochschuldidaktik@uni-tuebingen.de
www.uni-tuebingen.de/hochschuldidaktik

Tübinger Beiträge zur Hochschuldidaktik

Herausgeberinnen: Dr. Andrea Fausel, Dipl.-Päd. Regine Richter

Die Beiträge sind abrufbar unter <http://tobias-lib.uni-tuebingen.de/portal/tbhd/>

(2021) 17/1

Computergestützte Methodenkompetenzvermittlung für Studienanfänger in den wirtschaftswissenschaftlichen Studiengängen

Johannes Bleher & Thomas Dimpfl

(2020) 16/1

Zukunftslabor Universität. ESIT – Erfolgreich studieren in Tübingen: Ideen und Entwicklungen für Studium und Lehre

Sabine Merckens

(2019) 15/1

Turn Knowledge Into Action – Service Learning an der Universität Tübingen

Franziska Müller

(2018) 14/2

Digital unterstützte Lehre in den Geisteswissenschaften

Kristin Weingart: DAAT – Digitale Arbeitshilfe zum Alten Testament

Hans-Peter Nill: Gaudium Latinum. Gamification in der Vorbereitung auf das Latinum anhand des digitalen Antwortsystems Socrative

(2018) 14/1

Hochschulbildung und Hochschuldidaktik – (K)ein Thema in der Erziehungswissenschaft? Eine historische Spurensuche

(überarbeitete Neuauflage)

Karin Reiber & Ludwig Huber

(2017) 13/2

„Keiner liest...“ Lesekompetenz fördern – ein Modell aus dem Fach Kirchengeschichte

Daniela Blum

(2017) 13/1

Hochschulbildung und Hochschuldidaktik – (K)ein Thema in der Erziehungswissenschaft? Eine historische Spurensuche

Karin Reiber & Ludwig Huber

(2016) 12/1

Curriculumentwicklung im interdisziplinären Kontext

Modell für eine Vernetzung der wissenschaftlichen Fächer im Studiengang Evangelische Kirchenmusik B

Christoph Scheerer

(2015) 11/2

Interprofessionell Lehren und Lernen im Medizinstudium

Michael Kraus & Gerd Krischak

(2015) 11/1

Wissenschaftliches Schreiben als Herausforderung im Studienalltag

Jasmina Gherairi & Nicole Hirschfelder

(2014) 10/2

Lehrveranstaltungen lernförderlich gestalten

Ulrike Hanke & Samantha Winandy

(2014) 10/1

Bildung für nachhaltige Entwicklung. Das Konzept und seine Potenziale für traditionelle Volluniversitäten

Simon Meisch

(2013) 9/3

Kritische Übergänge. Theoretische Überlegungen und empirische Befunde zur Frage der Studierbarkeit

Rolf Frankenberger

(2013) 9/2

Vielfalt gestalten. Konstruktiver Umgang mit Heterogenität in Lehrveranstaltungen (2. Auflage)

Regine Richter

(2013) 9/1

Portfolio »International Studieren«. Internationalisierung des Curriculums durch interkulturelle Kompetenz und Integration

Katharina Kilian-Yasin

(2012) 8/1

Hochschuldidaktik für gesundheitsbezogene Studiengänge. Eine theoretische Grundlegung

Karin Reiber

(2011) 7/1

Konflikte in der Hochschullehre. Die Moderationsmethode als Möglichkeit der Bearbeitung

Martina Wanner

(2010) 6/1

Projektarbeit mit Symposium als neue Lehrform im Fach »Medizinische Psychologie und Soziologie«

Andrea Kübler; Ute Strehl

(2009) 5/2

Integration der Kategorie Gender in die Hochschuldidaktik

Patricia Graf

(2009) 5/1

Ein Lehrportfolio als Teil von Bewerbungsunterlagen

Miriam Noël Haidle

(2008) 4/3

Lernerzentrierte und aktivierende Lehre: Konzeption und Umsetzung am Beispiel des teilvirtuellen Seminars »Multimediale Landeskunde«

Birke Dockhorn

(2008) 4/2

Strengthening Cooperation and Enhancing Activation in Problem-Based Learning through Concrete External Representations

Manfred Künzle; Daniel Inderbitzin

(2008) 4/1

»Das zersägte Klassenzimmer«: Ein Gruppenpuzzle-Experiment in Zeiten der Ökonomisierung von Bildungsprozessen

Rolf Frankenberger

(2007) 3/2

Aktive Studierende – kompetenzorientierte Ausbildung: Fallbeispiele lernender Lehrender

Manfred Künzle; Silvana Rizzi; Heike Zinsmeister; Andreas Oelze; Thomas D'Souza; Patricia Graf

(2007) 3/1

Forschendes Lernen als hochschuldidaktisches Prinzip – Grundlegung und Beispiele

Karin Reiber

(2006) 2/1

Wissen – Können – Handeln: ein Kompetenzmodell für lernorientiertes Lehren

Karin Reiber

(2005) 1/2

Vielfalt als Chance: konstruktiver Umgang mit Heterogenität in Lehrveranstaltungen

Regine Richter

(2005) 1/1

Schriftliches Gutachten und Checkliste als Form der Rückmeldung zu studentischen Haus- und Abschlussarbeiten: ein Modell aus dem Fach Evangelische Theologie

Ulrike Treusch

