

# Analyse des Verhaltens von Benutzer 53

Das Verhalten des definierten Benutzers wird nachfolgend im Kontext verschiedener Merkmale und Zeiträume betrachtet.

Vermutlicher Status Student im Bachelor-Studiengang Wirtschaftsingenieurwesen Online (BWIO)

## Definition des Benutzers

```
In [34]: specified_user = 53
display(specified_user)
```

## Import von Bibliotheken und anderen Erweiterungen

```
In [35]: from sqlalchemy import create_engine
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from IPython.core.display_functions import display
```

## Definitionen zur Darstellung der Visualisierungen

```
In [36]: sns.set_theme(style='white', font_scale=1.2, palette='Spectral')
```

## Import der Originaldaten im CSV-Format (optional)

```
In [37]: # file = 'vrfh_moodle_ws20.csv'
# moodle_data = pd.read_csv(file, delimiter=';')
# moodle_data
```

## Herstellung der Verbindung zur MySQL-Datenbank

```
In [38]: user = "root"
password = "root"
host = "localhost"
database = "moodle_ws20"
port = 3306

engine = create_engine(f'mysql+pymysql://{user}:{password}@{host}/{database}', pool_recycle=port)
connection = engine.connect()
```

## Import der Arbeitsdaten aus der MySQL-Datenbank

```
In [39]: query = """SELECT * FROM moodle_data"""
moodle_data = pd.read_sql(query, connection) # Definition der Arbeitsdaten
```

## Auswahl der Arbeitsdaten

```
In [40]: # Konvertierung des Datentyps des Tabellenmerkmals timecreated
moodle_data['timecreated'] = pd.to_datetime(moodle_data['timecreated'], unit='s')

# Gegenstand des Untersuchungen sind zur Datensätze mit einer bestimmten userid
moodle_data = moodle_data[moodle_data.userid == specified_user]
moodle_data
```

	courseid	Studiengang	userid	relateduserid	action	eventname	objecttable	objectid	timecreated
14054	27502	2	53	29	viewed	/core/user/profile/viewed	user	1593	2020-10-23 14:57:02
39603	0	0	53	0	loggedin	/core/event/user_loggedin	user	37806	2020-10-12 04:25:36
39604	0	0	53	53	viewed	/core/event/dashboard_viewed		0	2020-10-12 04:25:37
39605	27502	2	53	0	viewed	/core/event/course_viewed		0	2020-10-12 04:25:42
39606	27502	2	53	0	viewed	/mod/resource/event/course_module_viewed	resource	202381	2020-10-12 04:25:46
...	...	...	...	...	...	...	...	...	...
966489	28256	2	53	0	viewed	/core/event/course_viewed		0	2021-06-03 06:38:49
966490	28256	2	53	0	viewed	/core/event/course_viewed		0	2021-06-03 06:38:57
966491	28256	2	53	0	viewed	/core/event/course_viewed		0	2021-06-03 06:39:00
966492	28256	2	53	0	viewed	/core/event/course_viewed		0	2021-06-03 06:39:18
966493	28256	2	53	0	viewed	/core/event/course_viewed		0	2021-06-03 06:39:22

11699 rows x 11 columns

## Menge der unterschiedlichen Werte pro Tabellenmerkmal

```
In [41]: attributes = moodle_data.nunique()
display(attributes)
```

	courseid	Studiengang	userid	relateduserid	action	eventname	objecttable	objectid	timecreated
	33	1	1	15	16	67	34	689	10508
	2	2	2	2	2	2	2	2	2
	dtype: int64								

```
In [42]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(attributes.count(), 8)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Ermittlung der Menge der unterschiedlichen Werte pro Tabellenmerkmal
result = attributes
# Visualisierung der Menge der unterschiedlichen Werte pro Tabellenmerkmal
chart = sns.barplot(x=result.index, y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('total number distinct values')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/attributes.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Studiengang

```
In [43]: studies = moodle_data.Studiengang.groupby(moodle_data.Studiengang).count()
display(studies)
```

	Studiengang	count
0	3853	
2	7846	
Name:	Studiengang, dtype: int64	

```
In [44]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(studies.count(), 8)) # Größe der Visualisierung (in inch)

# Ermittlung der Menge der Log-Einträge pro Studiengang
result = studies
# Visualisierung der Menge der Log-Einträge pro Studiengang
chart = sns.barplot(x=result.index, y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('moodle_data.Studiengang')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/studies.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Kurs

```
In [45]: courses = moodle_data.courseid.groupby(moodle_data.courseid).count()
display(courses.sort_values())
```

	courseid	count
1	2	
13910	3	
28240	7	
25038	7	
19591	10	
25911	8	
25043	9	
25918	12	
1750	14	
26762	14	
25039	19	
26764	22	
27562	32	
26761	72	
28265	83	
28267	86	
26769	100	
28255	107	
28229	120	
28238	132	
27502	161	
4217	188	
27503	239	
27515	262	
27501	527	
27488	541	
28256	570	
27499	606	
27512	778	
28259	968	
4245	1016	
27500	1249	
0	3718	
Name:	courseid, dtype: int64	

```
In [46]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(courses.count(), 8)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=90) # Drehung der Achsenbeschriftung

# Ermittlung der Menge der Log-Einträge pro Kurs
result = courses
# Visualisierung der Menge der Log-Einträge pro Kurs
chart = sns.barplot(x=result.index, y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('moodle_data.courseid')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/courses.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Action

```
In [47]: actions = moodle_data.action.groupby(moodle_data.action).count()
display(actions)
```

	action	count
added	2	
created	35	
deleted	2	
failed	173	
graded	10	
joined	18	
loggedin	2	
loggedout	757	
reviewed	3	
sent	30	
started	9	
submitted	34	
updated	46	
uploaded	3	
viewed	10320	
Name:	action, dtype: int64	

```
In [48]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(actions.count(), 8)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=90) # Drehung der Achsenbeschriftung

# Ermittlung der Menge der Log-Einträge pro Action
result = actions
# Visualisierung der Menge der Log-Einträge pro Action
chart = sns.barplot(x=result.index, y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('moodle_data.action')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/actions.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Eventname

```
In [49]: eventtypes = moodle_data.eventname.groupby(moodle_data.eventname).count()
display(eventtypes)
```

	eventname	count
assignsubmission_file(event)assignable_uploaded	21	
assignsubmission_file(event)submission_created	17	
assignsubmission_file(event)submission_updated	4	
core/event/loading_viewed	4	
core/event/course_category_viewed	8	
...	...	
vmoud_wiki(event)course_module_viewed	169	
vmoud_wiki(event)page_diff_viewed	1	
vmoud_wiki(event)page_history_viewed	9	
vmoud_wiki(event)page_map_viewed	5	
vmoud_wiki(event)page_viewed	2	
Name:	eventname, Length: 67, dtype: int64	

```
In [50]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(eventtypes.count(), 8)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=90) # Drehung der Achsenbeschriftung

# Ermittlung der Menge der Log-Einträge pro Eventname
result = eventtypes
# Visualisierung der Menge der Log-Einträge pro Eventname
chart = sns.barplot(x=result.index, y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('moodle_data.eventname')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/eventtypes.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Objecttable

```
In [51]: objecttables = moodle_data.objecttable.groupby(moodle_data.objecttable).count()
display(objecttables)
```

	objecttable	count
assign	764	
assign_grades	57	
assign_submission	43	
assignsubmission_file	5	
highbluebuttonb	88	
course	276	
course_categories	8	
course_modules_completion	40	
ethelpad	1	
feedback	2	
feedback_completed	1	
forum	46	
forum_discussion	210	
forum_discussions_subs	5	
forum_posts	11	
glossary	113	
glossary_entries	10	
grade_grades	113	
groups	2	
groupselect	54	
login	11	
messages	7	
notifications	10	
page	59	
quiz	93	
quit_attempts	48	
resource	142	
url	945	
user	102	
wiki	842	
wiki_pages	169	
Name:	objecttable, dtype: int64	

```
In [52]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(objecttables.count(), 8)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=90) # Drehung der Achsenbeschriftung

# Ermittlung der Menge der Log-Einträge pro Objecttable
result = objecttables
# Visualisierung der Menge der Log-Einträge pro Objecttable
chart = sns.barplot(x=result.index, y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('moodle_data.objecttable')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/objecttables.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Coursemoduletype

```
In [53]: course_module_types = moodle_data.course_module_type.groupby(moodle_data.course_module_type).count()
display(course_module_types)
```

	course_module_type	count
0	11679	
154	20	
Name:	course_module_type, dtype: int64	

```
In [54]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(course_module_types.count(), 8)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=90) # Drehung der Achsenbeschriftung

# Ermittlung der Menge der Log-Einträge pro Coursemoduletype
result = course_module_types
# Visualisierung der Menge der Log-Einträge pro Coursemoduletype
chart = sns.barplot(x=result.index, y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('moodle_data.course_module_type')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/course_module_types.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Tagesstunde

```
In [55]: # Ausgabe der Tagesstunden der protokollierten Log-Einträge
hours_per_day = moodle_data.timecreated.dt.hour.sort_values()
display(hours_per_day)
```

	hours_per_day	count
782660	0	
782659	0	
782658	0	
53034	1	
53036	1	
...	...	
445884	23	
346666	23	
346665	23	
346664	23	
346663	23	
Name:	timecreated, Length: 11699, dtype: int64	

```
In [56]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, 8)) # Größe der Visualisierung (in inch)

# Visualisierung der Menge der Log-Einträge pro Tagesstunde
chart = sns.histplot(hours_per_day, bins=24, discrete=True, color='#6A522E', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('hours per day')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/hours_per_day.pdf', format='pdf')
plt.show()
```

## Visualisierung der Menge der Log-Einträge pro Tagesstunde und Studiengang

```
In [57]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, 8)) # Größe der Visualisierung (in inch)

# Visualisierung der Menge der Log-Einträge pro Tagesstunde und Studiengang
chart = sns.complot(x=hours_per_day, hue=moodle_data.Studiengang, hue_order=studies.index)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('hours per day')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.legend(loc='upper right')
plt.savefig('charts/hours_per_day_and_studies.pdf', format='pdf')
plt.show()
```

## Visualisierung der kumulierten Mengen der Log-Einträge über alle Tagesstunden

```
In [58]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, 8)) # Größe der Visualisierung (in inch)

# Visualisierung der kumulierten Mengen der Log-Einträge über alle Tagesstunden
chart = sns.histplot(hours_per_day, bins=24, discrete=True, cumulative=True, state='proportion', color='#6A522E', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('hours per day')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/hours_per_day_cumulative.pdf', format='pdf')
plt.show()
```

## Visualisierung der Menge der Log-Einträge pro Studiengang über alle Tagesstunden

```
In [59]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, studies.count(), 8)) # Größe der Visualisierung (in inch)

# Visualisierung der Menge der Log-Einträge pro Studiengang über alle Tagesstunden
chart = sns.boxplot(x=hours_per_day, y=moodle_data.Studiengang, order=studies.index)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='x')
chart.set_xlabel('x')
chart.set_ylabel('hours per day')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/hours_per_day_boxplot.pdf', format='pdf')
plt.show()
```

## Visualisierung der Menge der Log-Einträge pro Merkmal und Studiengang über alle Tagesstunden

```
In [60]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, moodle_data.action[moodle_data.Studiengang.astype(int) == 0].nunique(), 8)) # Größe der V

# Visualisierung der Menge der Log-Einträge pro Merkmal und Studiengang über alle Tagesstunden
chart = sns.boxplot(x=hours_per_day, y=moodle_data.action[moodle_data.Studiengang.astype(int) == 0], orient='h')

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='x')
chart.set_xlabel('x')
chart.set_ylabel('hours per day')
chart.set_xticks(range(0, 24, 1))
chart.set_xtick_params(left=False, bottom=False)
plt.legend(loc='upper right')
plt.savefig('charts/hours_per_day_action_boxplot.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Wochentag

```
In [61]: # Ausgabe der Wochentage der protokollierten Log-Einträge
days_per_week = moodle_data.timecreated.dt.weekday.sort_values()
display(days_per_week)
```

	days_per_week	count
734865	0	
735196	0	
735195	0	
735193	0	
735192	0	
...	...	
571463	6	
571462	6	
571461	6	
59831	6	
642075	6	
Name:	timecreated, Length: 11699, dtype: int64	

```
In [62]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der Menge der Log-Einträge pro Wochentag
chart = sns.histplot(days_per_week, bins=7, discrete=True, color='#6A522E', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('days per week')
chart.set_xticks(range(0, 7, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/days_per_week.pdf', format='pdf')
plt.show()
```

## Visualisierung der Menge der Log-Einträge pro Wochentag und Studiengang

```
In [63]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der Menge der Log-Einträge pro Wochentag und Studiengang
chart = sns.complot(x=days_per_week, hue=moodle_data.Studiengang, hue_order=studies.index)

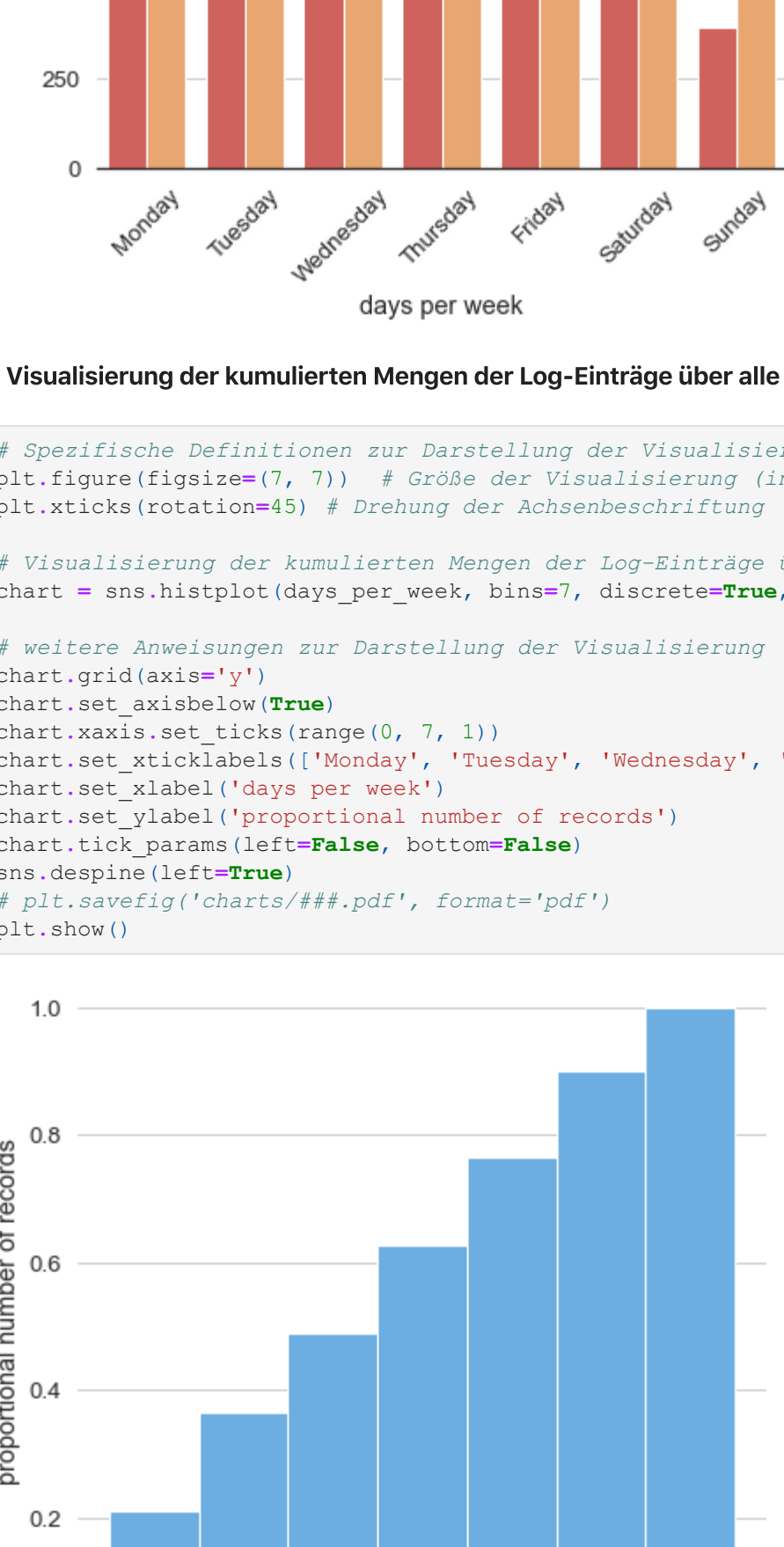
# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('days per week')
chart.set_xticks(range(0, 7, 1))
chart.set_xtick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/days_per_week_and_studies.pdf', format='pdf')
plt.show()
```



```
[63]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der Menge der Log-Einträge pro Wochentag und Studiengang
chart = sns.countplot(x=days_per_week, hue=moodle_data.Studiengang, hue_order=studies.index)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('Studienang')
chart.set_ylabel('total number records')
chart.set_xticks(range(0, 7, 1))
chart.set_xticklabels(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
chart.set_ylabel('total number records')
chart.set_yticks(range(0, 1750, 250))
sns.despine(left=True)
plt.legend(title='Studiengang', labels=studies.index)
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

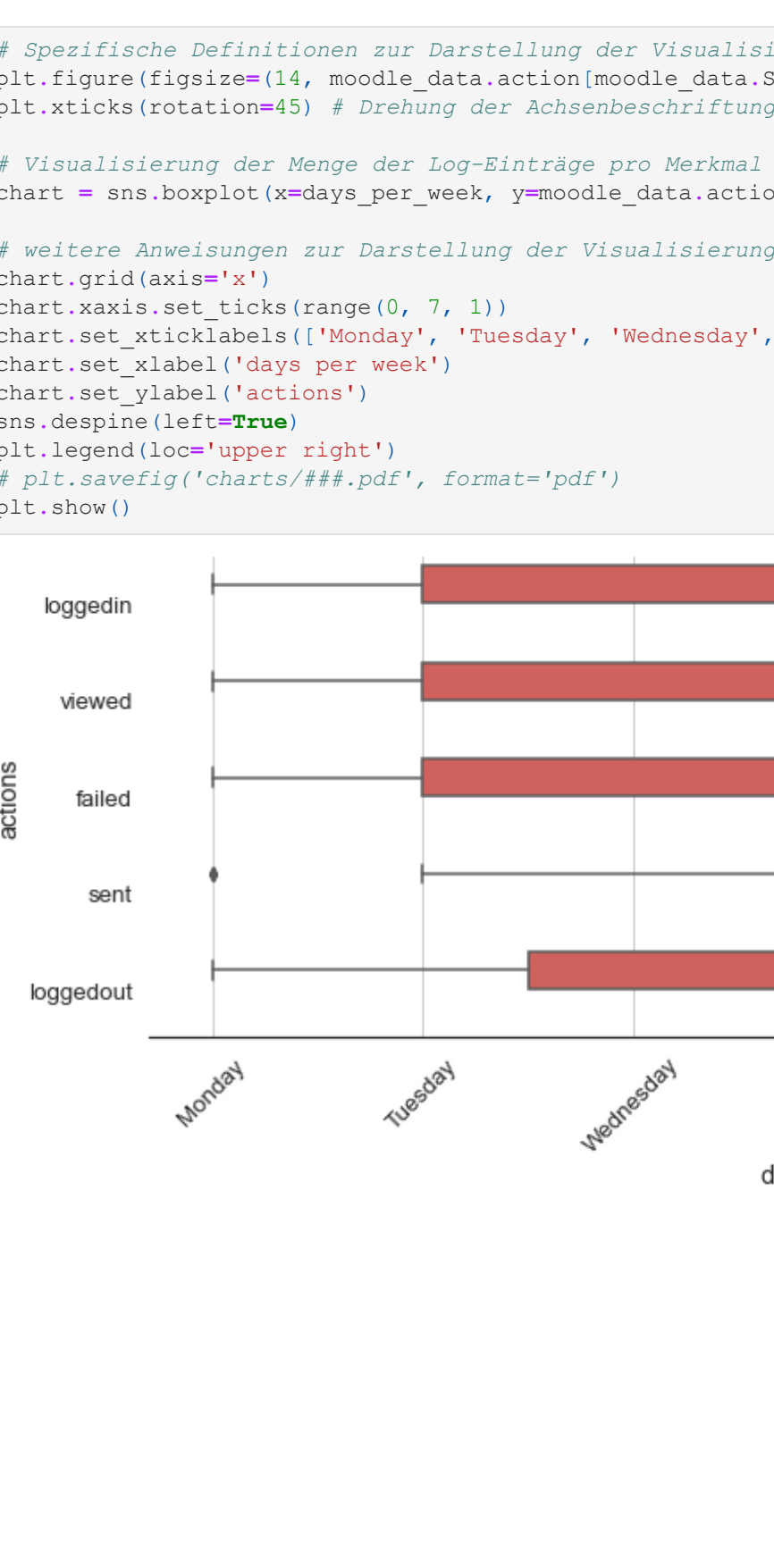


Visualisierung der kumulierten Mengen der Log-Einträge über alle Wochentage

```
In [64]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der kumulierten Mengen der Log-Einträge über alle Wochentage
chart = sns.histplot(days_per_week, bins=7, discrete=True, cumulative=True, stat='proportion', color='#D9AEE2')

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('Studienang')
chart.set_ylabel('proportional number of records')
chart.set_xticks(range(0, 7, 1))
chart.set_xticklabels(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
chart.set_ylabel('proportional number of records')
chart.set_yticks(range(0, 1, 0.2))
sns.despine(left=True)
plt.legend(title='Studiengang', labels=studies.index)
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



Visualisierung der Menge der Log-Einträge pro Studiengang über alle Wochentage

```
In [65]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(14, studies.count())) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der Menge der Log-Einträge pro Studiengang über alle Wochentage
chart = sns.boxplot(x=days_per_week, y=moodle_data.Studiengang, order=studies.index)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='x')
chart.set_xlabel('x')
chart.set_ylabel('y')
chart.set_xticks(range(0, 7, 1))
chart.set_xticklabels(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
chart.set_ylabel('y')
chart.set_yticks(range(0, 1, 0.2))
sns.despine(left=True)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



Visualisierung der Menge der Log-Einträge pro Merkmal und Studiengang über alle Wochentage

```
In [66]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(14, moodle_data.action[moodle_data.Studiengang.astype(int)].nunique())) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der Menge der Log-Einträge pro Merkmal und Studiengang über alle Wochentage
chart = sns.boxplot(x=days_per_week, y=moodle_data.action[moodle_data.Studiengang.astype(int)].nunique(), orient='h')

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='x')
chart.set_xlabel('x')
chart.set_ylabel('y')
chart.set_xticks(range(0, 7, 1))
chart.set_xticklabels(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])
chart.set_ylabel('y')
chart.set_yticks(range(0, 1, 0.2))
sns.despine(left=True)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

