

Analyse des Verhaltens von Benutzer 26

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

Vermutlicher Status Angehöriger des Hochschulpersonals

Definition des Benutzers

```
In [34]: specified_user = 26
```

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 = 'vrfn_module_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 = "vrfn_module_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]: # "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	timecreate
3669	0	0	26	-2	sent	/core/event/notification_sent	notifications	3474219	2020-11-12 04:44
3670	0	0	26	-2	sent	/core/event/notification_sent	notifications	3474235	2020-11-12 44
3671	0	0	26	-2	sent	/core/event/notification_sent	notifications	3474245	2020-11-12 44
3677	0	0	26	-2	sent	/core/event/notification_sent	notifications	3493921	2020-11-12 19
3788	0	0	26	6	sent	/core/event/notification_sent	notifications	3493929	2020-11-12 18
...
969025	0	0	26	26	viewed	/core/event/dashboard_viewed		0	2021-06-10 00:04
969026	1750	0	26	0	viewed	/core/event/course_viewed		0	2021-06-10 00:04
969027	1750	0	26	0	viewed	/mod/forum/event/course_module_viewed	forum	7082	2021-06-10 00:04
969028	1750	0	26	0	viewed	/mod/forum/event/discussion_viewed	forum_discussions	334595	2021-06-10 00:04
969029	1750	0	26	0	viewed	/core/event/course_viewed		0	2021-06-10 00:04

92242 rows x 11 columns

Menge der unterschiedlichen Werte pro Tabellenmerkmal

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

	count
courseid	168
Studiengang	5
userid	1
relateduserid	12
action	20
eventname	60
objecttable	40
objectid	34568
timecreated	7456
course_module_type	13
instancid	268
type	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
# 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_title('total number distinct values')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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	51966
1	57
2	95
3	6
4	116

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_title('moodle_data.Studiengang')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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
26656	2
27548	2
3913	2
2465	2
27957	2
...	...
2466	612
11807	762
24044	814
0	8308

Name: courseid, Length: 168, 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_title('moodle_data.courseid')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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
assigned	6
created	336
deleted	340
graded	6
failed	30
enrolled	2
joined	22
left	10
logged	152
loggedin	51
loggedout	14
reset	4
restored	4
searched	1906
submitted	79020
unassigned	10
updated	2
uploaded	194
viewed	9449

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('moodle_data.action')
chart.set_ylabel('total number records')
chart.set_title('total number records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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
/core/event/badge_listing_viewed	2
/core/event/course_category_viewed	706
/core/event/course_module_created	206
/core/event/course_module_deleted	239
/core/event/course_module_updated	162
/core/event/course_restored	6
/core/event/course_section_created	28
/core/event/course_section_deleted	30
/core/event/course_section_updated	52
/core/event/course_viewed	4034
/core/event/courses_searched	1906
/core/event/dashboard_reset	34
/core/event/dashboard_viewed	882
/core/event/grade_item_updated	4
/core/event/message_sent	2
/core/event/message_viewed	5
/core/event/notification_sent	79018
/core/event/role_assigned	6
/core/event/role_unassigned	10
/core/event/user_enrollment_deleted	4
/core/event/user_graded	10
/display/event/user_list_viewed	554
/core/event/user_loggedin	452
/core/event/user_loggedinas	51
/core/event/user_logout	14
/core/event/user_login_failed	30
/core/event/user_profile_viewed	142
/core/event/user_updated	6
/gradesreport/grade/event/grade_report_viewed	2
/mod_assign/event/course_module_viewed	4
/mod_assign/event/submit/status_viewed	4
/mod_bigbluebuttonbtn/event/activity_viewed	32
/mod_bigbluebuttonbtn/event/meeting_created	10
/mod_bigbluebuttonbtn/event/meeting_ended	6
/mod_bigbluebuttonbtn/event/meeting_joined	22
/mod_bigbluebuttonbtn/event/meeting_left	10
/mod_choice/event/answer_created	28
/mod_choice/event/course_module_viewed	6
/mod_choice/event/report_viewed	78
/mod_etherpad/event/course_module_viewed	78
/mod_folder/event/course_module_viewed	320
/mod_folder/event/folder_updated	34
/mod_forum/event/accessible_uploaded	194
/mod_forum/event/course_module_viewed	962
/mod_forum/event/discussion_created	54
/mod_forum/event/discussion_deleted	4
/mod_forum/event/discussion_viewed	1016
/mod_forum/event/post_created	16
/mod_forum/event/post_deleted	14
/mod_forum/event/post_updated	124
/mod_glossary/event/course_module_viewed	2
/mod_hvp/event/attempt_submitted	2
/mod_hvp/event/course_module_viewed	128
/mod_page/event/course_module_viewed	2
/mod_quiz/event/attempt_viewed	180
/mod_resource/event/course_module_viewed	330
/report_log/event/report_viewed	2

Name: eventname, 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_title('moodle_data.eventname')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

Menge der Log-Einträge pro Objecttable

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

objecttable	count
assign	4
bigbluebuttonbn	80
choice	34
choice_answers	10
course	558
course_categories	706
course_modules	666
course_sections	110
etherpad	78
folder	354
forum	962
forum_discussions	1078
forum_posts	338
forum_subscriptions	12
glossary	26
grade_grades	2
grade_items	4
hvp	2
message_user_actions	5
messages	2
notifications	79018
page	1208
quiz	2
quiz_attempts	2
resource	180
role	16
url	630
user	667
user_enrollments	4

Name: objecttable, dtype: int64

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

# Ermittlung der Menge der Log-Einträge pro Objecttable
result = objectables
# 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('moodle_data.objecttable')
chart.set_ylabel('total number records')
chart.set_title('total number records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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
assign	91576
bigbluebuttonbn	68
etherpad	12
folder	10
forum	18
glossary	2
hvp	6
label	4
loop	214
page	64
resource	176
url	190

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_title('moodle_data.course_module_type')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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
913371	1
913373	1
913370	1
913369	1
913368	1
...	...
271334	18
913359	1
272025	1
272022	18
580477	18
60147	18

Name: timecreated, Length: 92242, dtype: int64

Visualisierung der Menge der Log-Einträge pro Tagesstunde

```
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(x=hours_per_day, bins=24, discrete=True, color='#6DAE2B', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('total number records')
chart.set_title('total number records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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.countplot(x=hours_per_day, hue=moodle_data.Studiengang, order=studies.index)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('x')
chart.set_ylabel('total number records')
chart.set_title('total number records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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(x=hours_per_day, bins=24, discrete=True, cumulative=True, stat='proportion', color='#6DAE2B')

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.set_ylabel('total number records')
chart.set_title('total number records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.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())) # Größe der V

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

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='x')
chart.set_xlabel('x')
chart.set_ylabel('total number records')
chart.set_title('total number records')
chart.tick_params(left=False, bottom=False)
plt.legend(loc='upper right')
plt.savefig('charts/###.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
866149	0
869551	0
869554	0
869557	0
...	...
930279	4
930278	4
930277	4
930259	4
3669	4

Name: timecreated, Length: 92242, dtype: int64

Visualisierung der Menge der Log-Einträge pro Wochentag

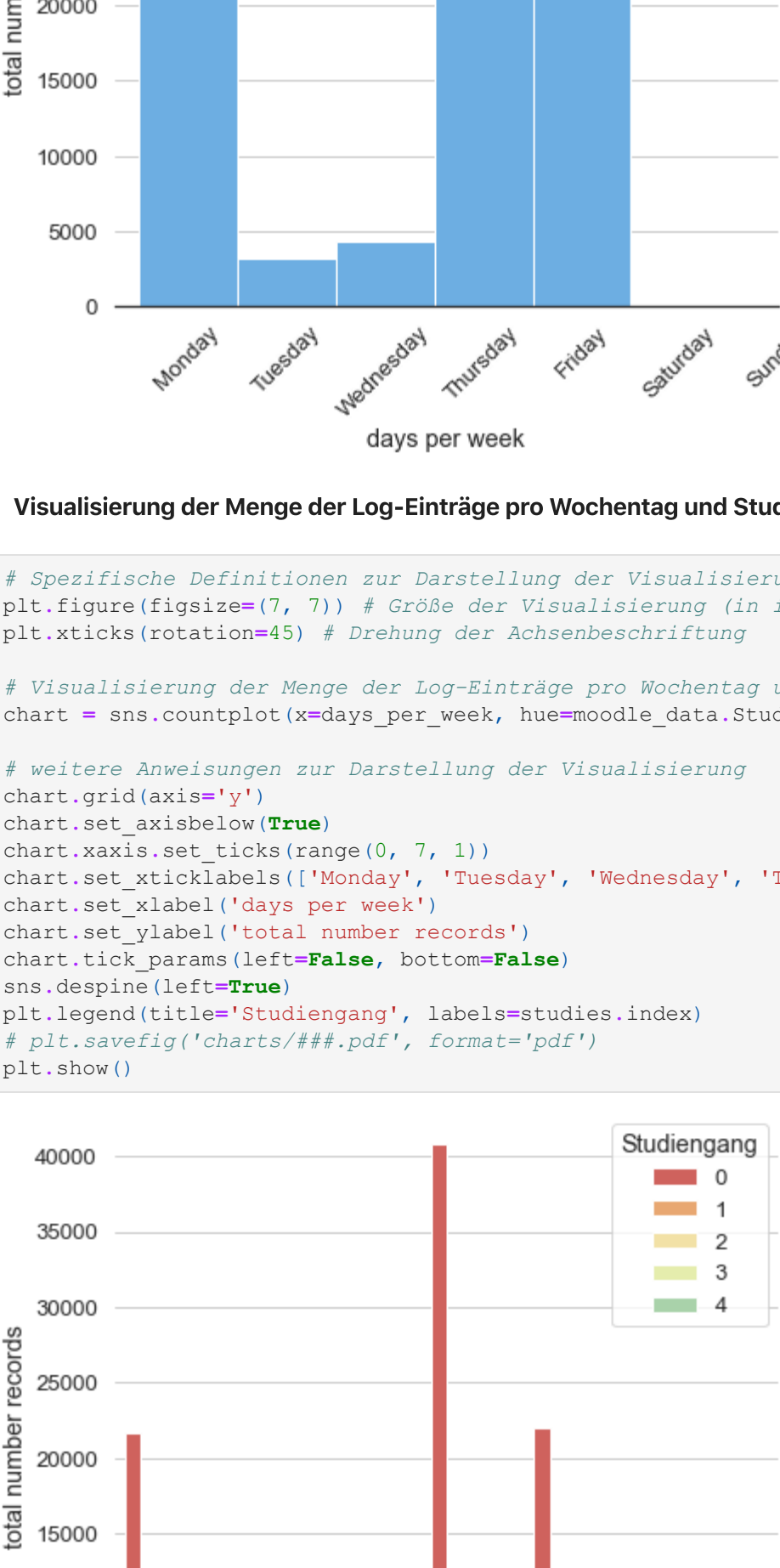

```
In [62]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der Menge der Log-Einträge pro Wochentag
chart = sns.histplot(days_per_week, bins=7, discrete=True, color='#6A5ACD', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('days per week')
chart.set_ylabel('total number records')
sns.despine(left=True)

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

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('days per week')
chart.set_ylabel('total number records')
sns.despine(left=True, bottom=False)
plt.legend(title='Studiengang', labels=studies.index)
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

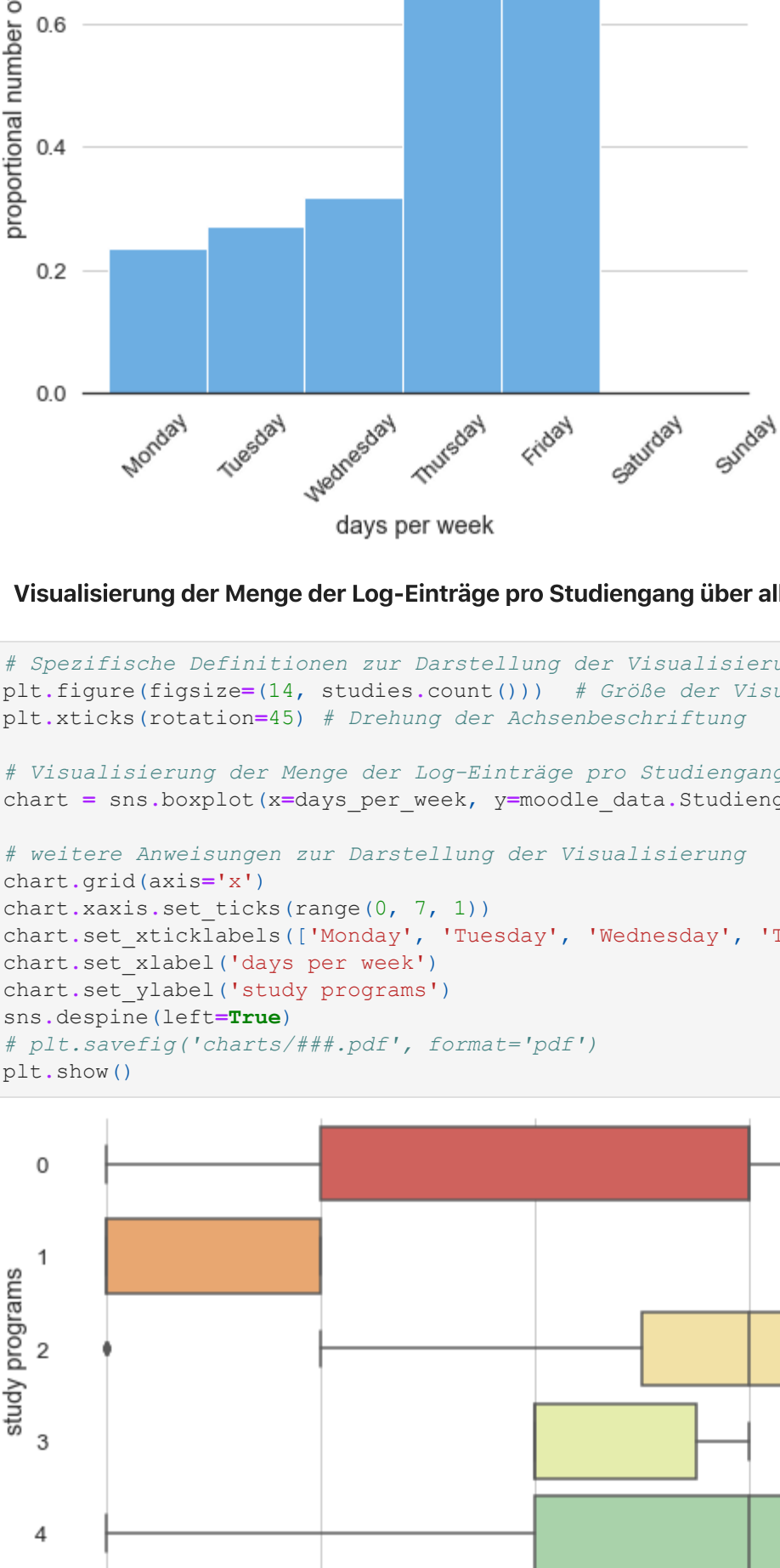


Visualisierung der Menge der Log-Einträge pro Wochentag und Studiengang

```
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.countplot(x=days_per_week, hue=moodle_data.Studiengang, order=studies.index)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('days per week')
chart.set_ylabel('total number records')
sns.despine(left=True, bottom=False)
plt.legend(title='Studiengang', labels=studies.index)
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

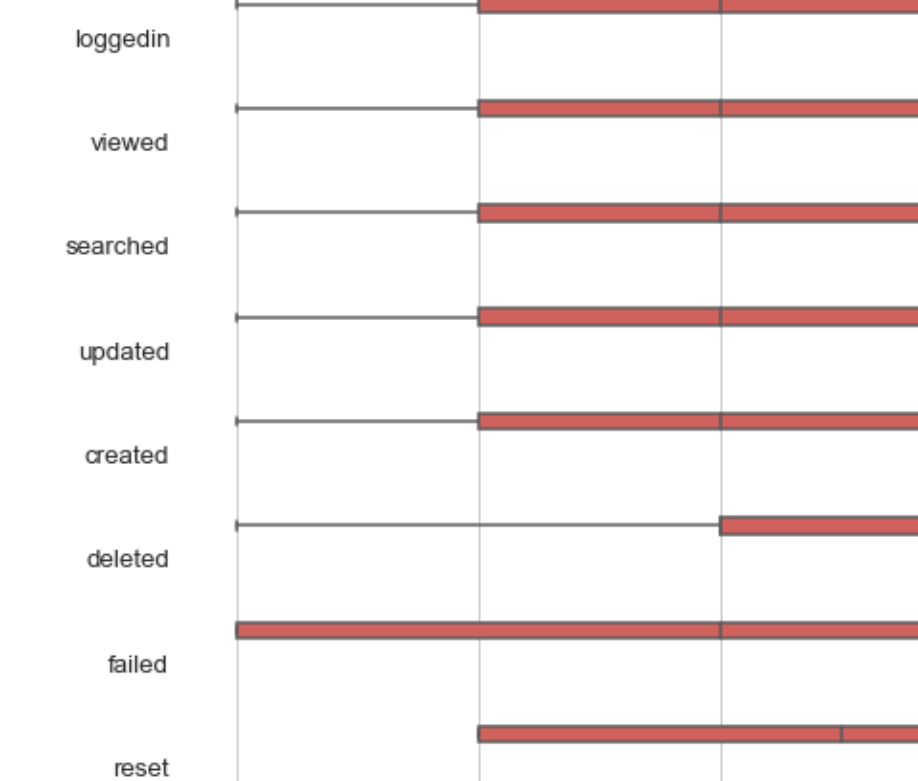


Visualisierung der kumulierten Mengen der Log-Einträge über alle Wochentage

```
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='#6A5ACD', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('days per week')
chart.set_ylabel('proportional number of records')
sns.despine(left=True, bottom=False)
plt.legend(title='Studiengang', labels=studies.index)
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

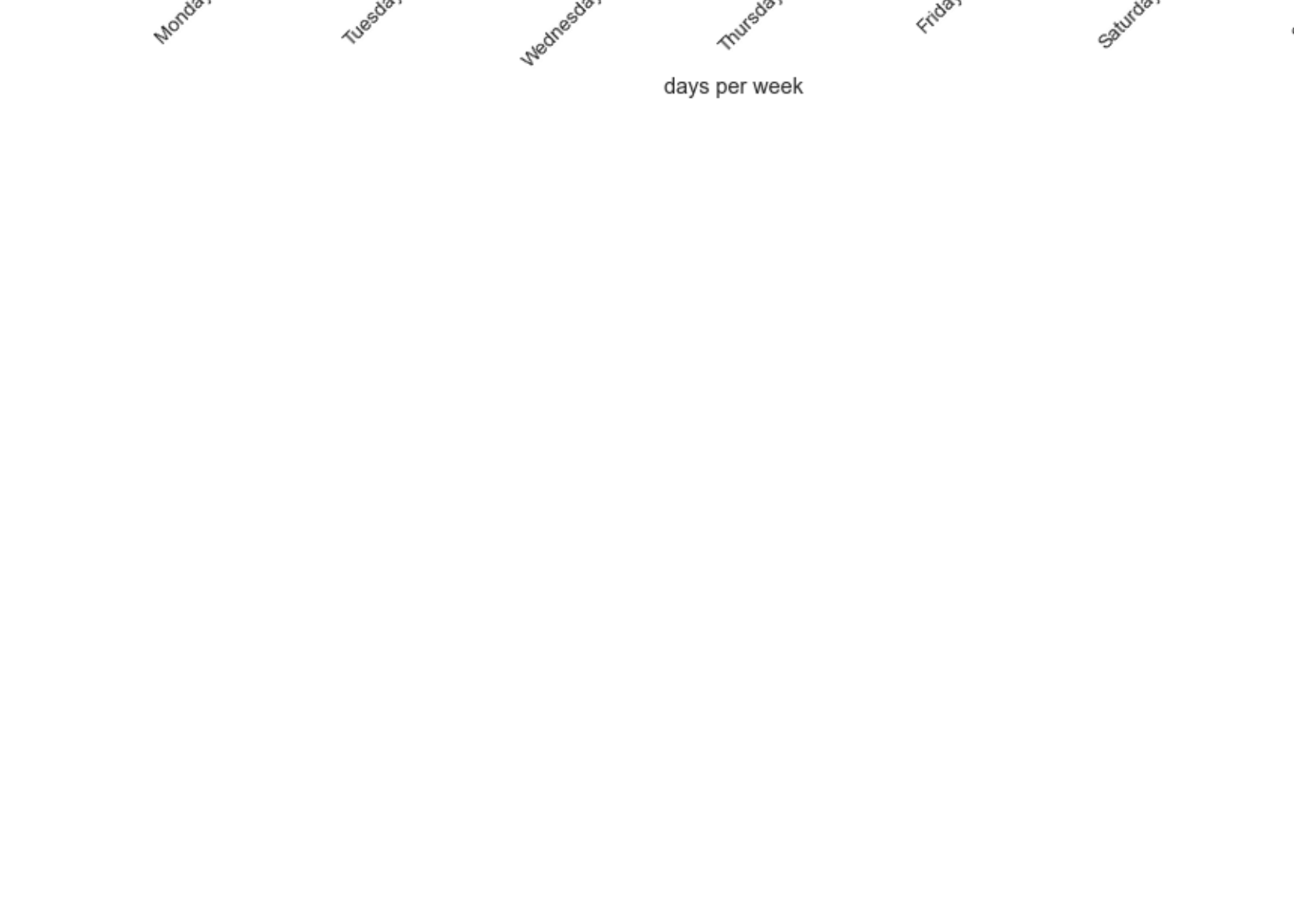


Visualisierung der Menge der Log-Einträge pro Studiengang über alle Wochentage

```
In [65]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # 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('days per week')
chart.set_ylabel('days per week')
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 Merkmal und Studiengang über alle Wochentage

```
In [66]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(14, 14)) # 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) == 0], orient='h')

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='x')
chart.set_xlabel('days per week')
chart.set_ylabel('actions')
sns.despine(left=True)
plt.legend(loc='upper right')
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

