

# Analyse des Verhaltens von Benutzer 1

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

**Vermittelter Status Student im Master-Studiengang Medieninformatik Online (MMIO)**

## Definition des Benutzers

```
In [133]: specified_user = 1
```

## Import der Bibliotheken und anderen Erweiterungen

```
In [134]: 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 [135]: sns.set_theme(style='white', font_scale=1.2, palette='Spectral')
```

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

```
In [136]: # file = 'mfm_moodle_ws20.csv'
# moodie_data = pd.read_csv(file, delimiter=',')
# moodie_data
```

## Herstellung der Verbindung zur MySQL-Datenbank

```
In [137]: user = "root"
password = "root"
host = "localhost"
database = "mfm_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 [138]: query = """SELECT * FROM moodie_data"""
moodie_data = pd.read_sql(query, connection) # Definition der Arbeitsdaten
```

## Auswahl der Arbeitsdaten

```
In [139]: # Konvertierung des Datentyps des Tabellenmerkmals timecreated
moodie_data['timecreated'] = pd.to_datetime(moodie_data['timecreated'], unit='s')

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

courseid	Studiengang	userid	relateduserid	action	eventname	objecttable	objectid	timecreated	course_module
0	0	0	1	-2	sent	/core/event/notification_sent	notifications	3232247	2020-10-12 04:54:03
534	0	0	1	-2	sent	/core/event/notification_sent	notifications	3290921	2020-10-19 15:54:03
1004	0	0	1	-2	sent	/core/event/notification_sent	notifications	3392636	2020-10-26 15:44:03
2217	0	0	1	-2	sent	/core/event/notification_sent	notifications	3387595	2020-11-02 15:25:18
2226	0	0	1	-2	sent	/core/event/notification_sent	notifications	3388891	2021-01-12 17:00:38
...	...	...	...	...	...	...	...	...	...
959253	0	0	1	-2	sent	/core/event/notification_sent	notifications	4567293	2021-06-01 09:47:12
959254	0	0	1	-2	sent	/core/event/notification_sent	notifications	4567294	2021-06-01 09:47:12
959255	0	0	1	-2	sent	/core/event/notification_sent	notifications	4567296	2021-06-01 09:47:12
959256	0	0	1	-2	sent	/core/event/notification_sent	notifications	4567298	2021-06-01 09:47:12
959257	0	0	1	-2	sent	/core/event/notification_sent	notifications	4567301	2021-06-01 09:47:12

3865 rows x 11 columns

## Menge der unterschiedlichen Werte pro Tabellenmerkmal

```
In [140]: attributes = moodie_data.nunique()
display(attributes)
```

courseid	Studiengang	userid	relateduserid	action	eventname	objecttable	objectid	timecreated	course_module
0	2123	1	1142	15	49	15	1128	2036	2
dtype:	int64								

```
In [141]: # 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_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Studiengang

```
In [142]: studies = moodie_data.Studiengang.groupby(moodie_data.Studiengang).count()
display(studies)
```

Studiengang	count
0	2123
1	1142

Name: Studiengang, dtype: int64

```
In [143]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(studies.count(), 8)) # Größe der Visualisierung (in Inch)
plt.xticks(rotation=90) # Drehung der Achsenbeschriftung

# 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('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Kurs

```
In [144]: courses = moodie_data.courseid.groupby(moodie_data.courseid).count()
display(courses.sort_values())
```

courseid	count
26661	1
1750	2
4209	4
28223	19
4237	89
28230	137
28491	198
27583	643
27637	651
0	2123

Name: courseid, dtype: int64

```
In [145]: # 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('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Action

```
In [146]: actions = moodie_data.action.groupby(moodie_data.action).count()
display(actions)
```

action	count
accepted	3
created	43
downloaded	1
ended	4
failed	4
joined	57
left	4
loggedin	178
loggedout	14
searched	1
sent	1571
submitted	12
updated	20
chart_xaxis.set	44
viewed	1909

Name: action, dtype: int64

```
In [147]: # 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('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Eventname

```
In [148]: eventtypes = moodie_data.eventname.groupby(moodie_data.eventname).count()
display(eventtypes)
```

eventname	count
assignsubmission_comments/event/comment_created	2
assignsubmission_file/event/assessable_uploaded	6
assignsubmission_file/event/submition_created	0
assignsubmission_file/event/submition_updated	2
assignsubmission_online/text/event/assessable_uploaded	1
assignsubmission_online/text/event/submition_created	1
/core/event/badge_viewed	1
/core/event/course_category_viewed	1
/core/event/course_viewed	661
/core/event/dashboard_viewed	341
/core/event/message_sent	9
/core/event/message_viewed	9
/core/event/notification_sent	1566
/core/event/recent_activity_viewed	3
/core/event/user_login	178
/core/event/user_loggedout	14
/core/event/user_login_failed	4
/core/event/user_profile_viewed	11
/gradereport_overview/event/grade_report_viewed	1
/mod_assign/event/assessable_submitted	12
/mod_assign/event/course_module_viewed	131
/mod_assign/event/feedback_viewed	23
/mod_assign/event/statement_accepted	3
/mod_assign/event/submition_created	1
/mod_assign/event/submition_status_viewed	10
/mod_bigbluebuttonchat/event/activity_viewed	161
/mod_bigbluebuttonchat/event/meeting_created	12
/mod_bigbluebuttonchat/event/meeting_ended	4
/mod_bigbluebuttonchat/event/meeting_joined	4
/mod_bigbluebuttonchat/event/meeting_left	4
/mod_bigbluebuttonchat/event/recorded_viewed	1
/mod_choice/event/course_module_viewed	18
/mod_choice/event/course_viewed	4
/mod_etherpad/event/course_module_viewed	4
/mod_feedback/event/course_module_viewed	4
/mod_feedback/event/response_submitted	4
/mod_forum/event/all_files_downloaded	1
/mod_forum/event/course_module_viewed	3
/mod_forum/event/assessable_uploaded	35
/mod_forum/event/course_module_viewed	176
/mod_forum/event/discussion_created	1
/mod_forum/event/discussion_viewed	215
/mod_forum/event/post_created	18
/mod_forum/event/post_updated	30
/mod_forum/event/user_report_viewed	3
/mod_page/event/course_module_viewed	30
/mod_resource/event/course_module_viewed	57
/mod_url/event/course_module_viewed	19

Name: eventname, dtype: int64

```
In [149]: # 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('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Objecttable

```
In [150]: objecttables = moodie_data.objecttable.groupby(moodie_data.objecttable).count()
display(objecttables)
```

objecttable	count
assign	1137
assign_grades	131
assign_submission	23
assign_submission_file	8
assign_submission_online/text	1
bigbluebutton	156
choice	18
choice_answers	1
comments	1
course_categories	1
etherpad	4
feedback	4
feedback_completed	3
folder	9
forum	176
forum_discussions	216
forum_posts	69
message_user_actions	6
messages	5
notifications	1566
page	30
resource	57
url	19
user	203

Name: objecttable, dtype: int64

```
In [151]: # 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('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Coursemoduletype

```
In [152]: course_module_types = moodie_data.course_module_type.groupby(moodie_data.course_module_type).count()
display(course_module_types)
```

course_module_type	count
0	3861
1	4

Name: course\_module\_type, dtype: int64

```
In [153]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(course_module_types.count(), 8)) # Größe der Visualisierung (in Inch)
plt.xticks(rotation=45) # 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('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Tagesstunde

```
In [154]: # Ausgabe der Tagesstunden der protokollierten Log-Einträge
hours_per_day = moodie_data.timecreated.dt.hour.sort_values()
display(hours_per_day)
```

hours_per_day	count
0	3757
1	37600
2	0
3	37378
4	0
5	554205
6	432586
7	0
8	554190
9	23
10	554199
11	23
12	554202
13	23
14	488778
15	23

Name: timecreated, Length: 3865, dtype: int64

## Visualisierung der Menge der Log-Einträge pro Tagesstunde

```
In [155]: # 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='#6A8EE2', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('hours per day')
chart.set_ylabel('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

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

```
In [156]: # 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, hue_order=[studies.index])

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('hours per day')
chart.set_ylabel('total number records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

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

```
In [157]: # 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, stat='proportion', color='#6A8EE2')

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('hours per day')
chart.set_ylabel('proportional number of records')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

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

```
In [158]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, studies.count())) # 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('hours per day')
chart.set_ylabel('study programs')
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

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

```
In [159]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, moodie_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, 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('hours per day')
chart.set_ylabel('hours per day')
chart.set_params(left=True, bottom=False)
sns.despine(left=True, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```

## Menge der Log-Einträge pro Wochentag

```
In [160]: # Ausgabe der Wochentage der protokollierten Log-Einträge
days_per_week = moodie_data.timecreated.dt.weekday.sort_values()
display(days_per_week)
```

days_per_week	count
0	0
1	330104
2	330105
3	330106
4	330107
5	0
6	414874
7	414875
8	414973
9	414975
10	667126

Name: timecreated, Length: 3865, dtype: int64

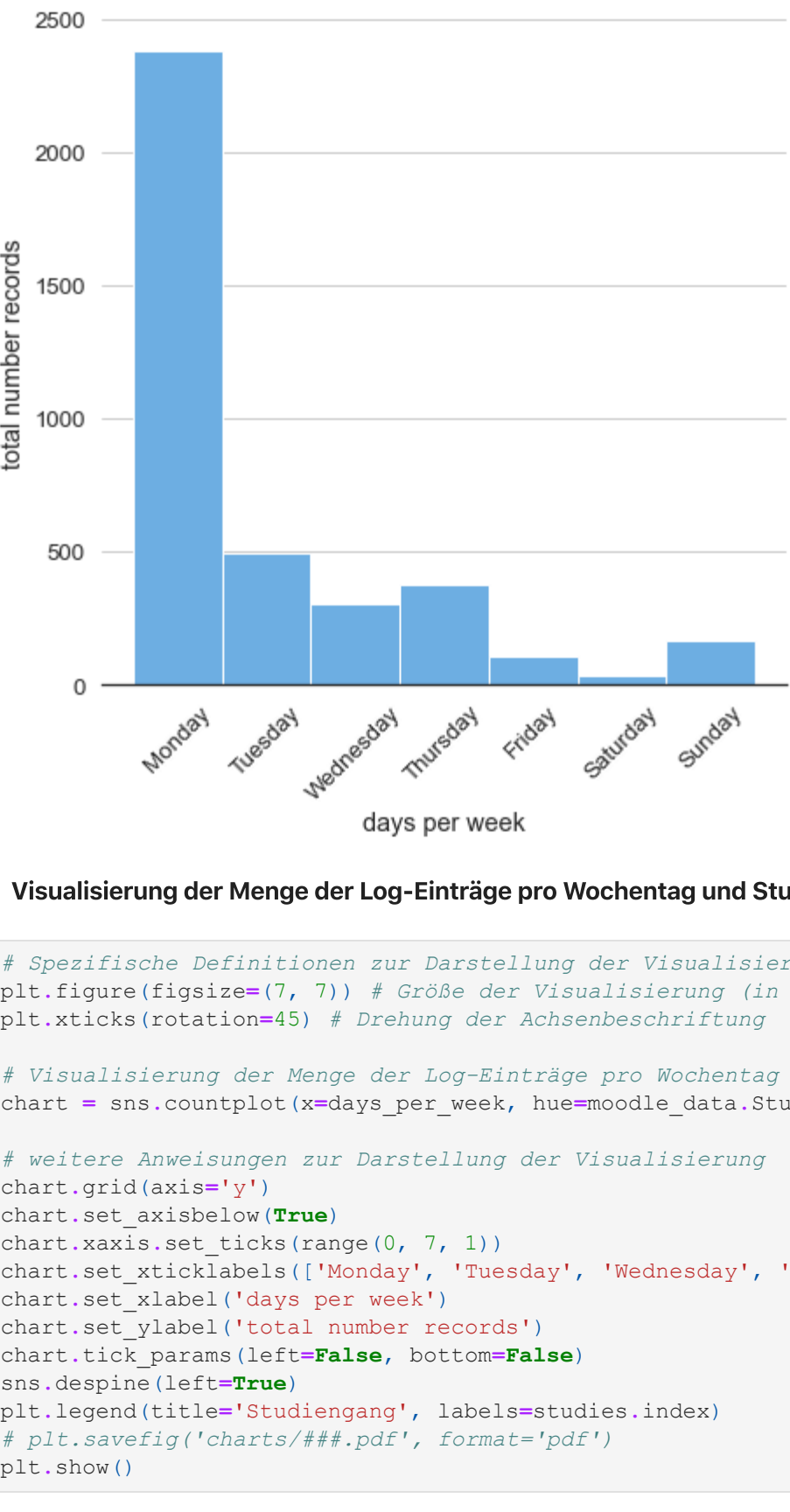
## Visualisierung der Menge der Log-Einträge pro Wochentag

```
In [161]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # Größe der Visualisierung (in Inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

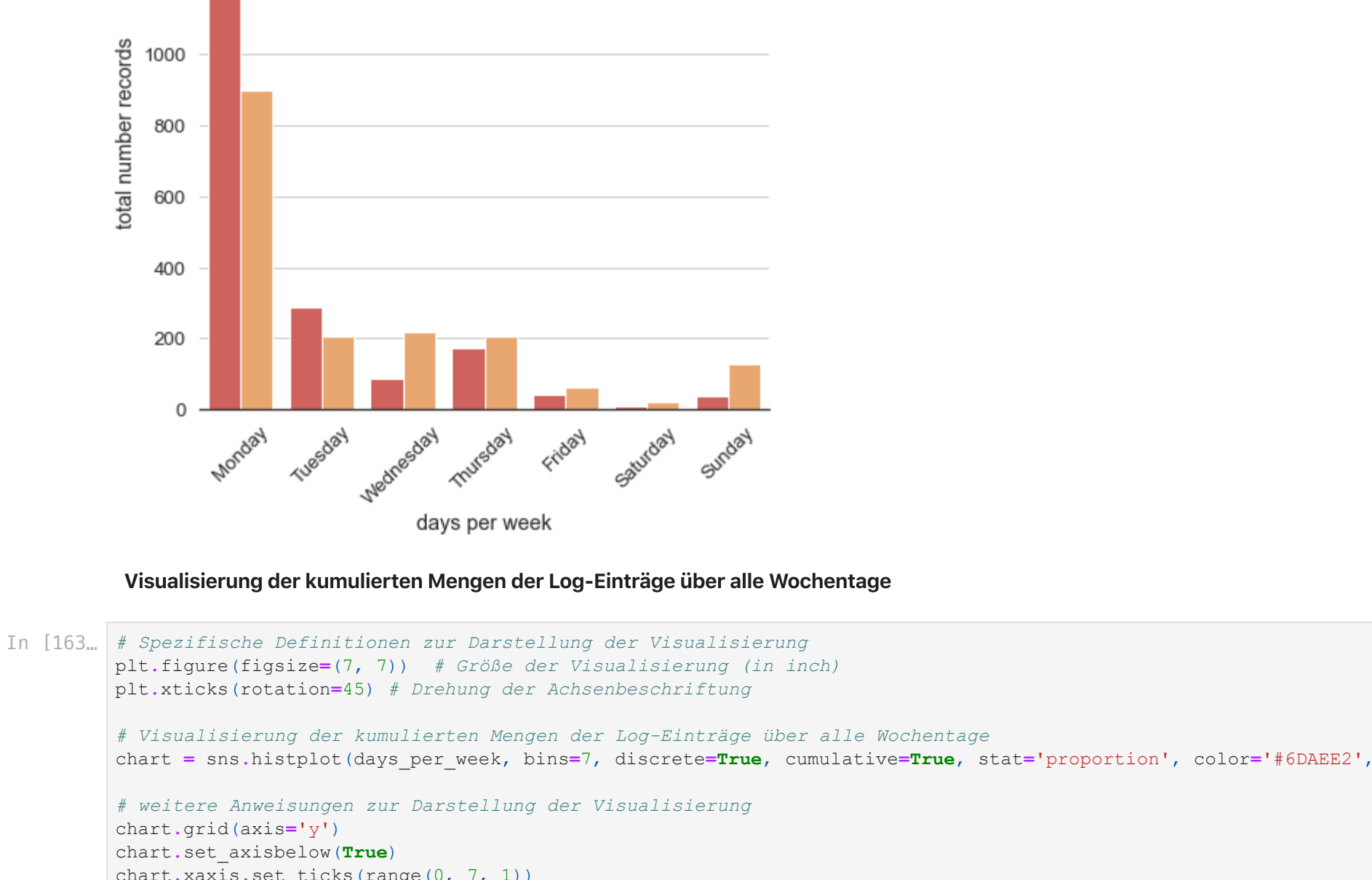
# Visualisierung der Menge der Log-Einträge pro Wochentag
chart = sns.histplot(days_per_week, bins=7, discrete=True, color='#6A8EE2', alpha=1)

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

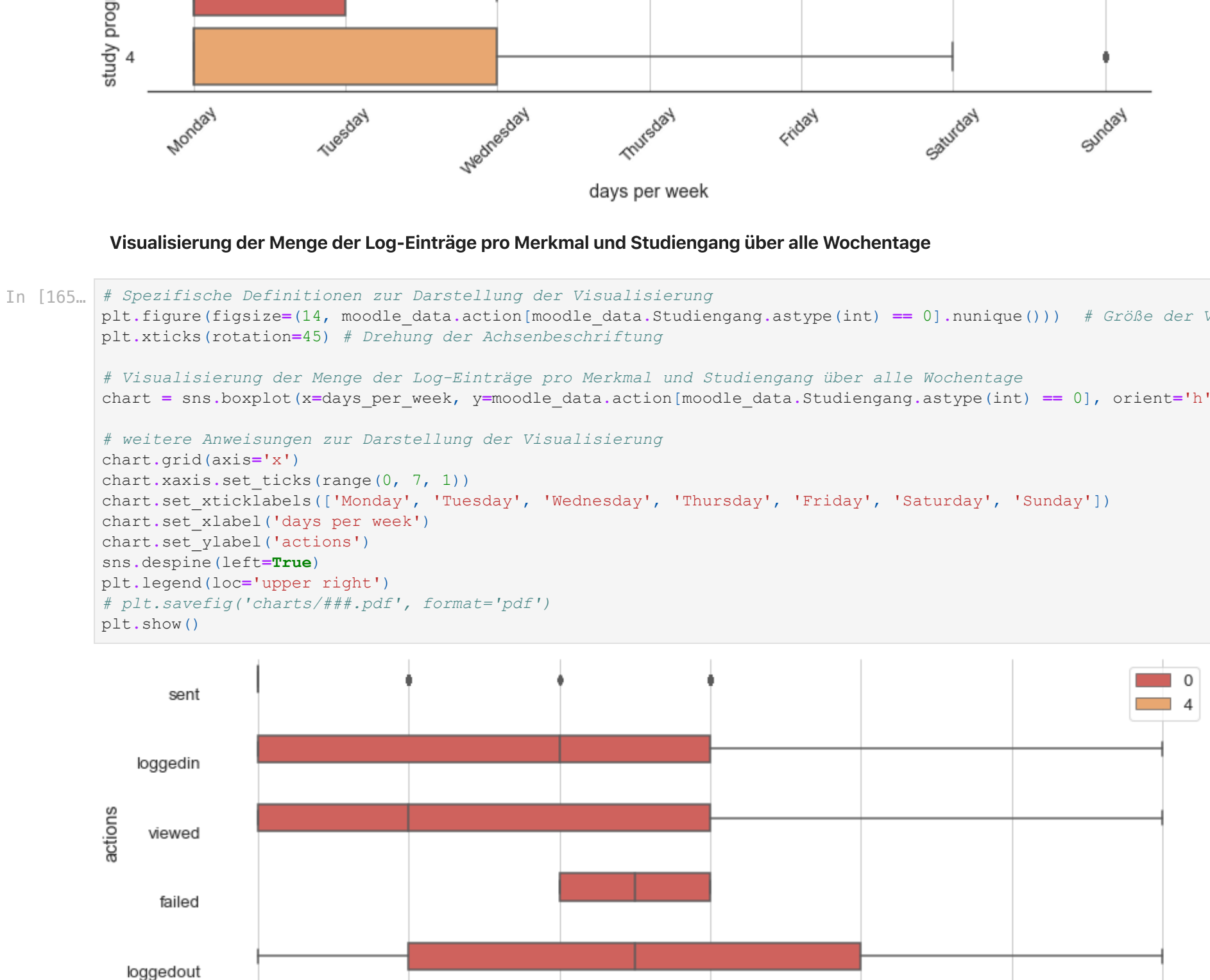




Visualisierung der Menge der Log-Einträge pro Wochentag und Studiengang



Visualisierung der kumulierten Mengen der Log-Einträge über alle Wochentage



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



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

