

# Analyse des Verhaltens von Benutzer 1

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

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

## Definition des Benutzers

In [1]:

```
specified_user = 1
```

## Import von Bibliotheken und anderen Erweiterungen

In [2]:

```
from sqlalchemy import create_engine
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from matplotlib.ticker import MaxNLocator
from IPython.core.display import display
```

## Definitionen zur Darstellung der Visualisierungen

In [3]:

```
sns.set_theme(style='white', font_scale=1.2, palette='Spectral1')
```

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

In [4]:

```
!file "vrh_moodle_ws20.csv"
# moodle_data = pd.read_csv(file, delimiter=',')
# moodle_data
```

## Herstellung der Verbindung zur MySQL-Datenbank

In [5]:

```
username = "root"
password = "root"
host = "localhost"
database = "vrh_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 [6]:

```
query = """SELECT * FROM moodle_data"""
moodle_data = pd.read_sql(query, connection) # Definition der Arbeitsdaten
```

## Auswahl der Arbeitsdaten

In [7]:

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

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

Out [7]:

	courseid	Studiengang	userid	relateduserid	action	eventname	objecttable	objectid	timecreated	course_module
	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	2323247	2020-10-12 04:54:03
534	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	3290921	2020-10-19 15:54:03
1004	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	3332636	2020-10-26 15:44:03
2217	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	3387595	2020-11-02 15:25:18
2226	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	3386891	2020-11-02 17:00:38
...	...	...	...	...	...	...	...	...	...	...
959253	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	4567293	2021-06-01 09:47:12
959254	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	4567294	2021-06-01 09:47:12
959255	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	4567296	2021-06-01 09:47:12
959256	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	4567298	2021-06-01 09:47:12
959257	0	0	0	1	-2	sent	/core/eventnotification_sent	notifications	4567301	2021-06-01 09:47:12

3865 rows x 11 columns

## Ausgabe der Mengen an unterschiedlichen Werten pro Tabellenmerkmal

In [8]:

```
attributes = moodle_data.nunique()
display(attributes)

courseid      11
Studiengang    2
userid         1
relateduserid   8
action        15
eventname      49
objecttable    25
objectid       1123
timecreated    2036
course_module_type  2
userid         2
dtype: int64
```

In [9]:

```
# 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 Mengen an unterschiedlichen Werten pro Tabellenmerkmal
result = attributes
# Visualisierung der Mengen an unterschiedlichen Werten pro Tabellenmerkmal
chart = sns.barplot(x=result.index, y=result)
```

```
# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('all attributes')
chart.set_ylabel('total number distinct values')
chart.tick_params(left=False, bottom=False)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



## Ausgabe der Mengen an Log-Einträgen pro Studiengang

In [10]:

```
studies = moodle_data.Studiengang.groupby(moodle_data.Studiengang).count()
display(studies)
```

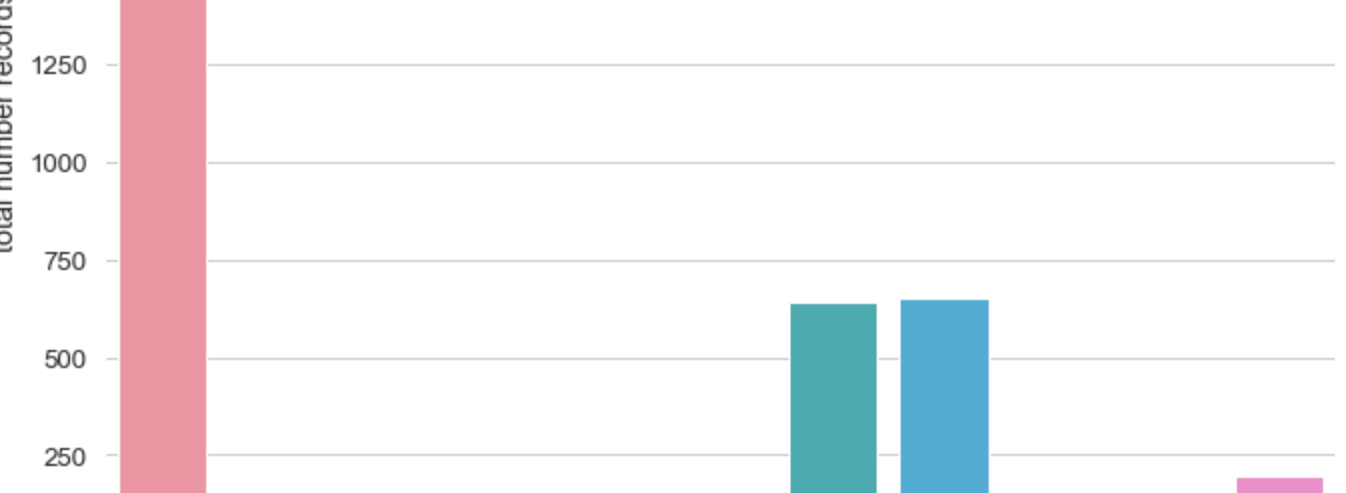
```
Studiengang
0      2123
1       1742
Name: Studiengang, dtype: int64
```

In [11]:

```
# 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 Mengen an Log-Einträgen pro Studiengang
result = studies
# Visualisierung der Mengen an Log-Einträgen pro Studiengang
chart = sns.barplot(x=result.index, y=result)
```

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



## Ausgabe der Mengen an Log-Einträgen pro courseid

In [12]:

```
courses = moodle_data.courseid.groupby(moodle_data.courseid).count()
display(courses.sort_values())
```

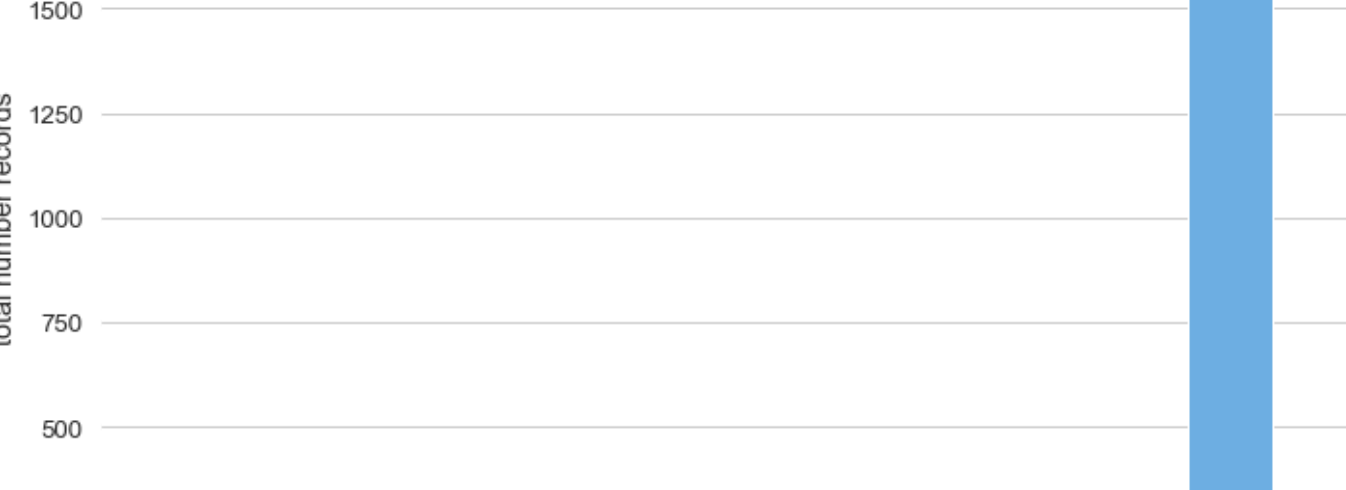
```
courseid
26681      2
1          2
1750       2
4209       4
28238     19
4237      89
28230     137
28461     198
27583     443
27637     651
0         2119
Name: courseid, dtype: int64
```

In [13]:

```
# 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 Mengen an Log-Einträgen pro courseid
result = courses
# Visualisierung der Mengen an Log-Einträgen pro courseid
chart = sns.barplot(x=result.index, y=result)
```

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



## Ausgabe der Mengen an Log-Einträgen pro action

In [14]:

```
actions = moodle_data.action.groupby(moodle_data.action).count()
display(actions)
```

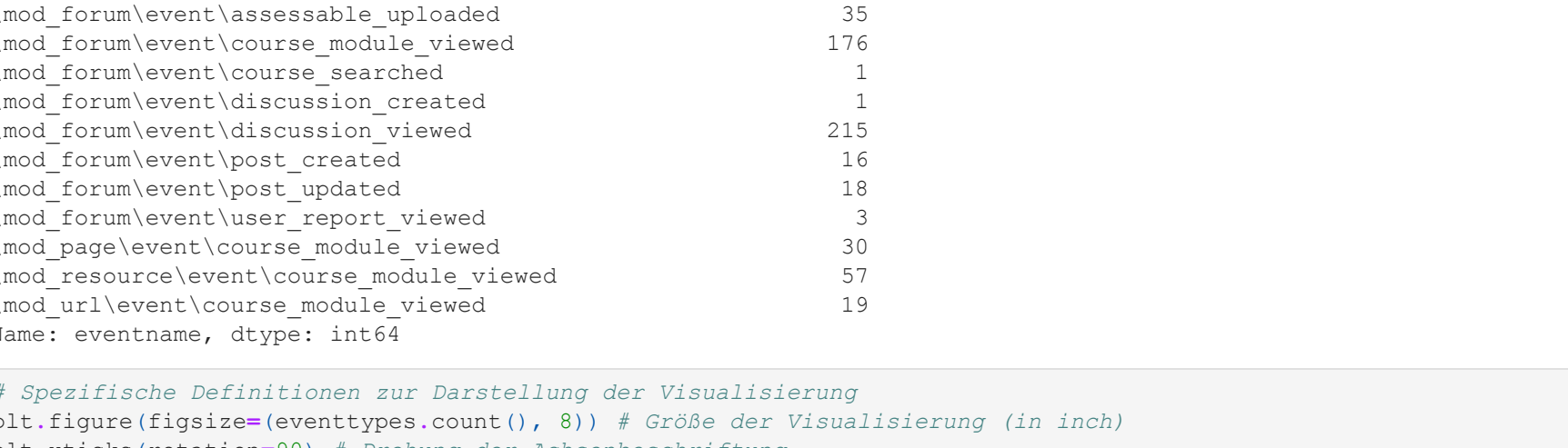
```
action
accepted      3
created       43
downloaded     1
ended         4
failed         4
joined        57
left          4
loggedin      178
loggedout     14
searched      1
sent         1571
submitted     12
updated       20
uploaded      44
viewed       1909
Name: action, dtype: int64
```

In [15]:

```
# 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 Mengen an Log-Einträgen pro action
result = actions
# Visualisierung der Mengen an Log-Einträgen 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.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



## Ausgabe der Mengen an Log-Einträgen pro eventname

In [16]:

```
eventtypes = moodle_data.eventname.groupby(moodle_data.eventname).count()
display(eventtypes)
```

```
eventname
assignsubmission_comments/event/comment_created      2
assignsubmission_file/event/assessable_uploaded      8
assignsubmission_file/event/submition_uploaded       6
assignsubmission_online/text/event/assessable_uploaded 2
assignsubmission_online/text/event/submition_created 1
/core/event/badge_viewed                             1
/core/event/course_category_viewed                   1
/core/event/course_viewed                             4
/core/event/dashboard_viewed                         341
/core/event/message_sent                             1566
/core/event/message_viewed                           5
/core/event/notification_sent                        1566
/core/event/recent_activity_viewed                   4
/core/event/user_loggedin                             178
/core/event/user_logout                              14
/core/event/user_login_failed                        14
/core/event/user_profile_viewed                       4
/grade/report/overview/event/grade_report_viewed      1
vmoodle_assign/event/assessable_viewed                9
vmoodle_assign/event/course_module_viewed             23
vmoodle_assign/event/feedback_viewed                  3
vmoodle_assign/event/statements_accepted              101
vmoodle_assign/event/submition_form_viewed            20
vmoodle_assign/event/submition_status_viewed           3
vmoodle_bigbluebutton/event/activity_viewed           71
vmoodle_bigbluebutton/event/meeting_created           16
vmoodle_bigbluebutton/event/meeting_ended            57
vmoodle_bigbluebutton/event/meeting_joined            57
vmoodle_bigbluebutton/event/meeting_left              4
vmoodle_bigbluebutton/event/recording_viewed          4
vmoodle_choice/event/answer_created                   1
vmoodle_choice/event/course_module_viewed             1
vmoodle_ethperpad/event/course_module_viewed          4
vmoodle_feedback/event/course_module_viewed           1
vmoodle_feedback/event/response_submitted             35
vmoodle_folder/event/all_files_downloaded             4
vmoodle_forum/event/course_module_viewed              1
vmoodle_forum/event/assessable_uploaded               35
vmoodle_forum/event/course_module_viewed              16
vmoodle_forum/event/course_searched                   1
vmoodle_forum/event/discussion_created                215
vmoodle_forum/event/discussion_viewed                 101
vmoodle_forum/event/post_created                     16
vmoodle_forum/event/post_updated                      3
vmoodle_forum/event/user_report_viewed                 3
vmoodle_page/event/course_module_viewed               39
vmoodle_resource/event/course_module_viewed           57
vmoodle_url/event/course_module_viewed                10
Name: eventname, dtype: int64
```

In [17]:

```
# 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 Mengen an Log-Einträgen pro eventname
result = eventtypes
# Visualisierung der Mengen an Log-Einträgen pro eventname
chart = sns.barplot(x=result.index, y=result)
```

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



## Ausgabe der Mengen an Log-Einträgen pro objecttable

In [18]:

```
objecttables = moodle_data.objecttable.groupby(moodle_data.objecttable).count()
display(objecttables)
```

```
objecttable
assign_grades      1137
assign_grades      131
assign_grades      23
assign_submission  21
assign_submission  8
assignsubmission_online/text 156
choice            18
choice_answers    1
comments          2
course_categories 4
ethperpad         4
feedback          4
feedback_completed 3
folder           5
forum             176
forum_discussions 216
forum_posts      69
message_user_actions 9
messages         5
notifications     1566
page             19
resource         57
uri              30
user            203
Name: objecttable, dtype: int64
```

In [19]:

```
# 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 Mengen an Log-Einträgen pro objecttable
result = objecttables
# Visualisierung der Mengen an Log-Einträgen 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.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



## Ausgabe der Mengen an Log-Einträgen pro course\_module\_type

In [20]:

```
course_module_types = moodle_data.course_module_type.groupby(moodle_data.course_module_type).count()
display(course_module_types)
```

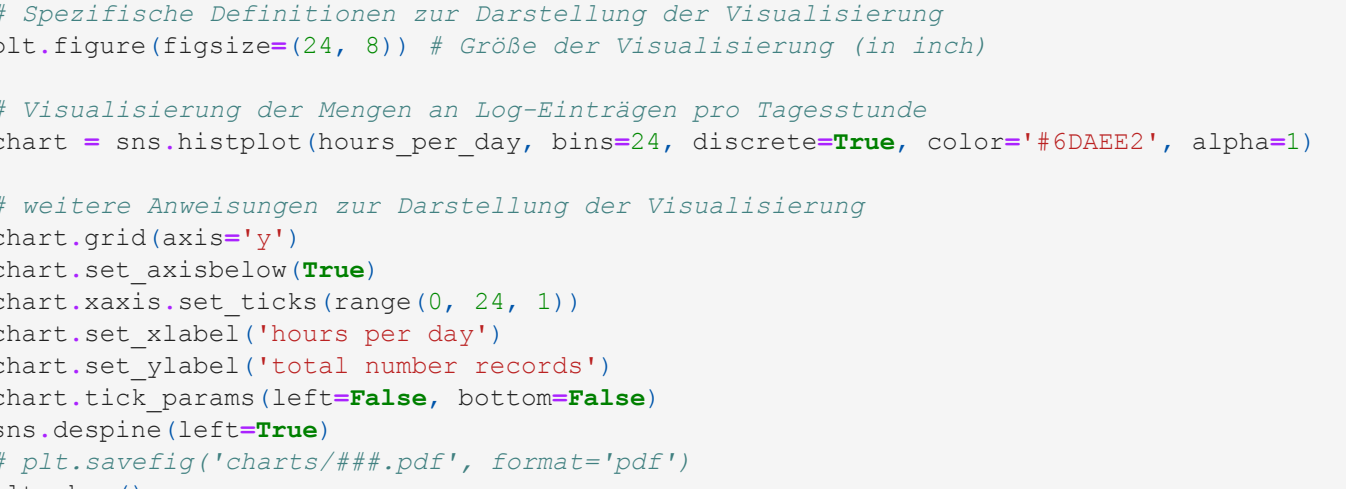
```
course_module_type
0      3865
Name: course_module_type, dtype: int64
```

In [21]:

```
# 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 Mengen an Log-Einträgen pro course_module_type
result = course_module_types
# Visualisierung der Mengen an Log-Einträgen pro course_module_type
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_ylabel('total number records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



## Ausgabe der Mengen an Log-Einträgen nach Tagesstunden

In [22]:

```
hours_per_day = moodle_data.timecreated.dt.hour.sort_values()
display(hours_per_day)
```

```
hours_per_day
37577      0
37600      0
37578      0
554205     0
432586     0
Name: timecreated, Length: 3865, dtype: int64
```

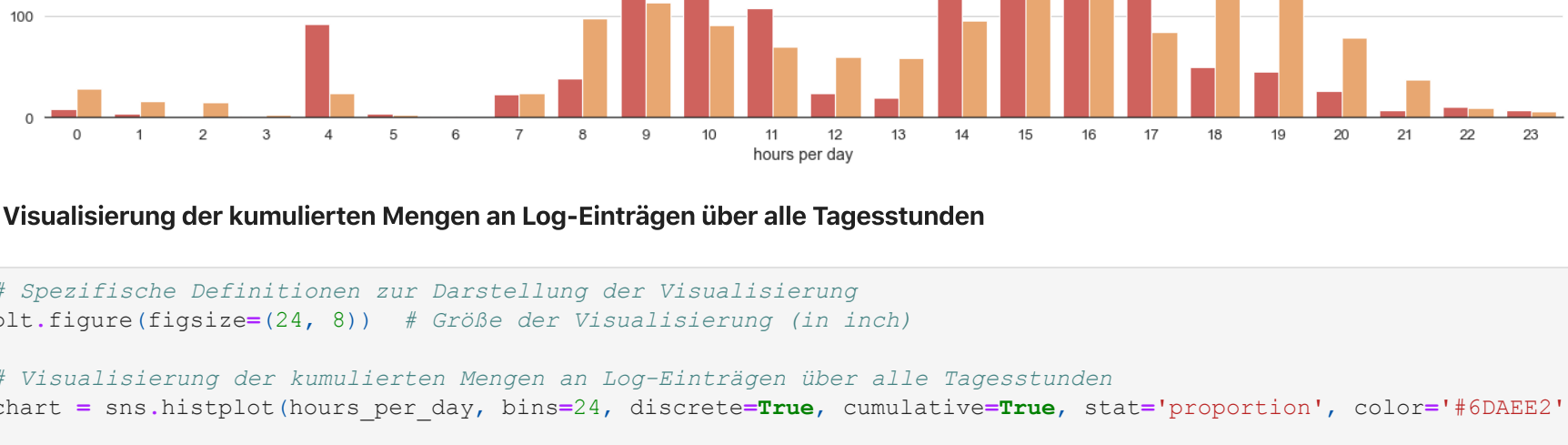
## Visualisierung der Mengen an Log-Einträgen pro Tagesstunde

In [23]:

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

# Visualisierung der Mengen an Log-Einträgen pro Tagesstunde
chart = sns.histplot(hours_per_day, bins=24, discrete=True, color='#60A8B2', alpha=1)
```

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



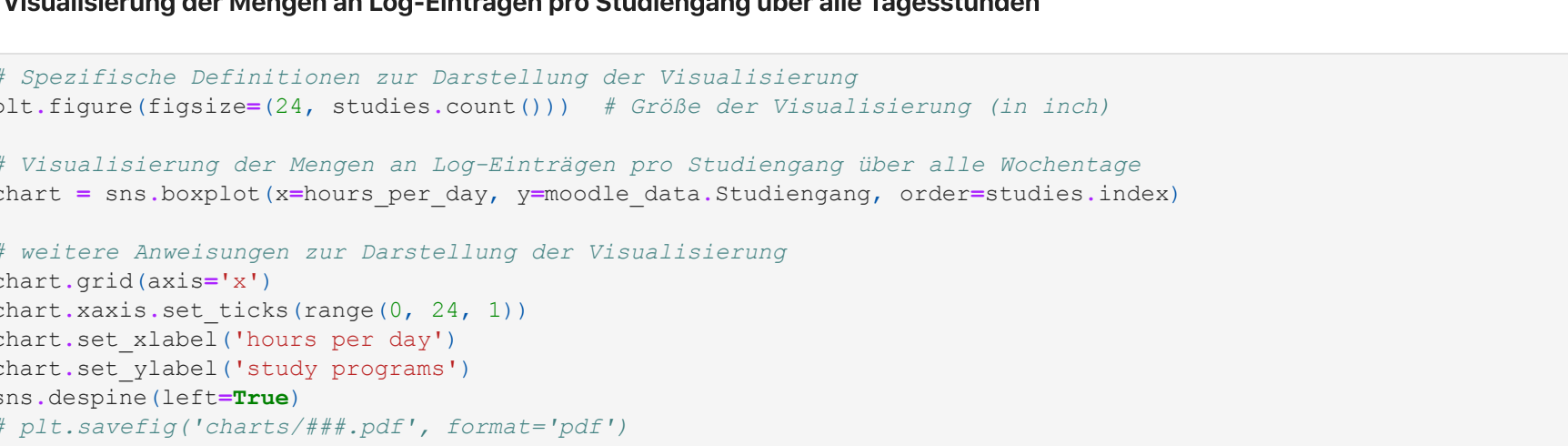
## Visualisierung der Mengen an Log-Einträgen pro Tagesstunde und Studiengang

In [24]:

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

# Visualisierung der Mengen an Log-Einträgen pro Tagesstunde
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('total number records')
chart.set_ylabel('hours per day')
chart.set_ylabel('study programs')
plt.legend(title='Studiengang', labels=studies.index)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



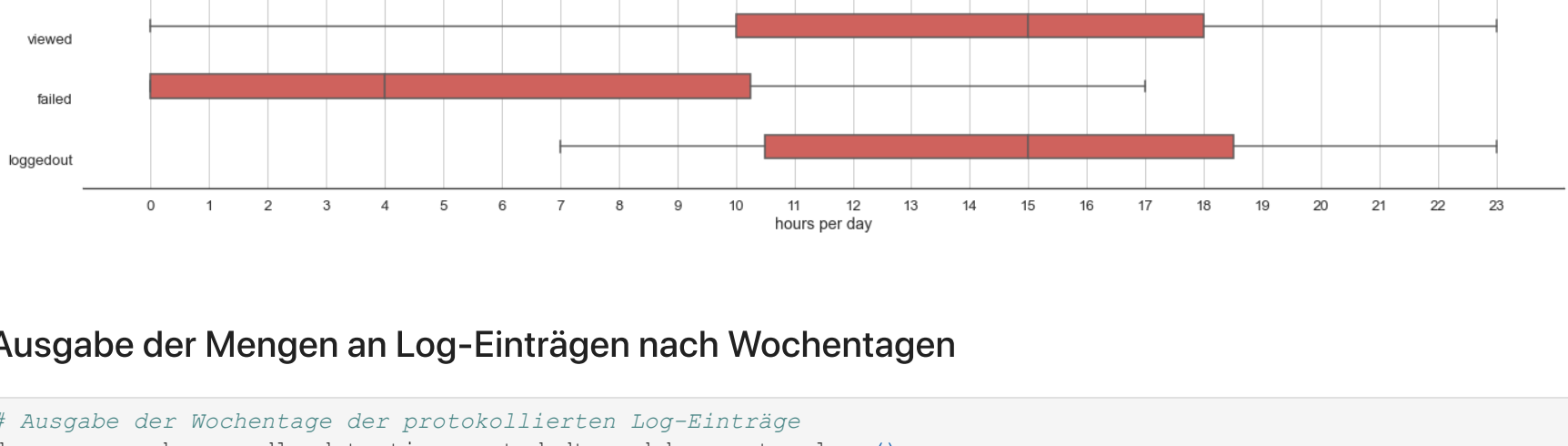
## Visualisierung der kumulierten Mengen an Log-Einträgen über alle Tagesstunden

In [25]:

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

# Visualisierung der kumulierten Mengen an Log-Einträgen über alle Tagesstunden
chart = sns.histplot(hours_per_day, bins=24, discrete=True, cumulative=True, stat='proportion', color='#60A8B2', alpha=1)
```

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



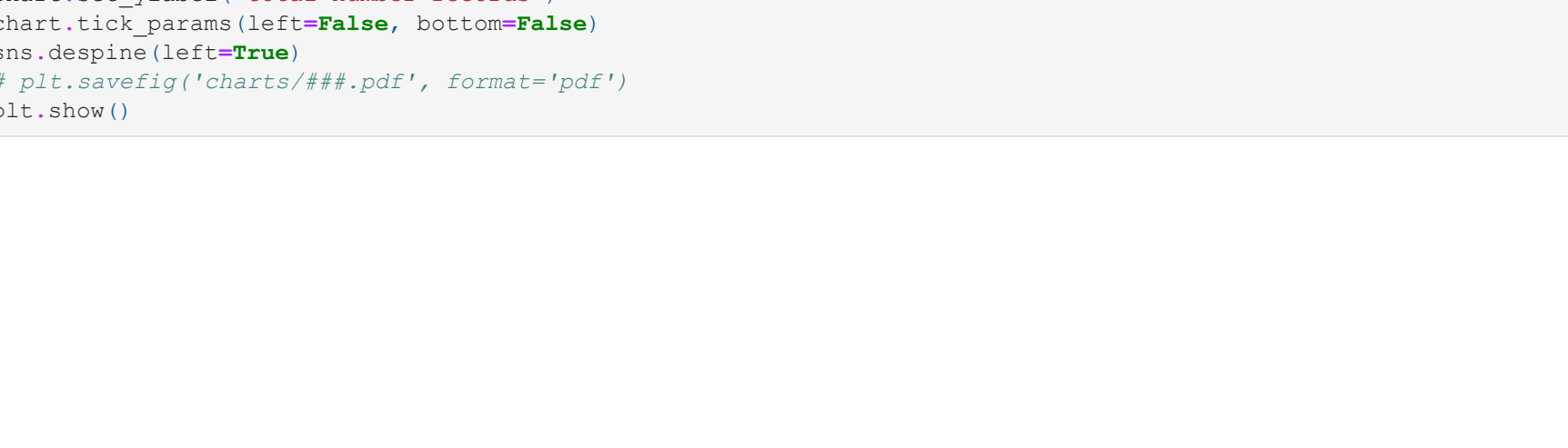
## Visualisierung der Mengen an Log-Einträgen pro Studiengang über alle Tagesstunden

In [26]:

```
# Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, studies.count())) # Größe der Visualisierung (in inch)

# Visualisierung der Mengen an Log-Einträgen 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='y')
chart.set_xlabel('x')
chart.set_ylabel('hours per day')
chart.set_ylabel('actions')
sns.despine(left=True)
plt.legend(title='hours per day')
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



## Ausgabe der Mengen an Log-Einträgen nach Wochentagen

In [28]:

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

```
days_per_week
0      0
33014    6
33015    6
33016    6
33017    6
414874   '
414875   6
414876   6
414877   6
617224   6
Name: timecreated, Length: 3865, dtype: int64
```

## Visualisierung der Mengen an Log-Einträgen pro Wochentag

In [29]:

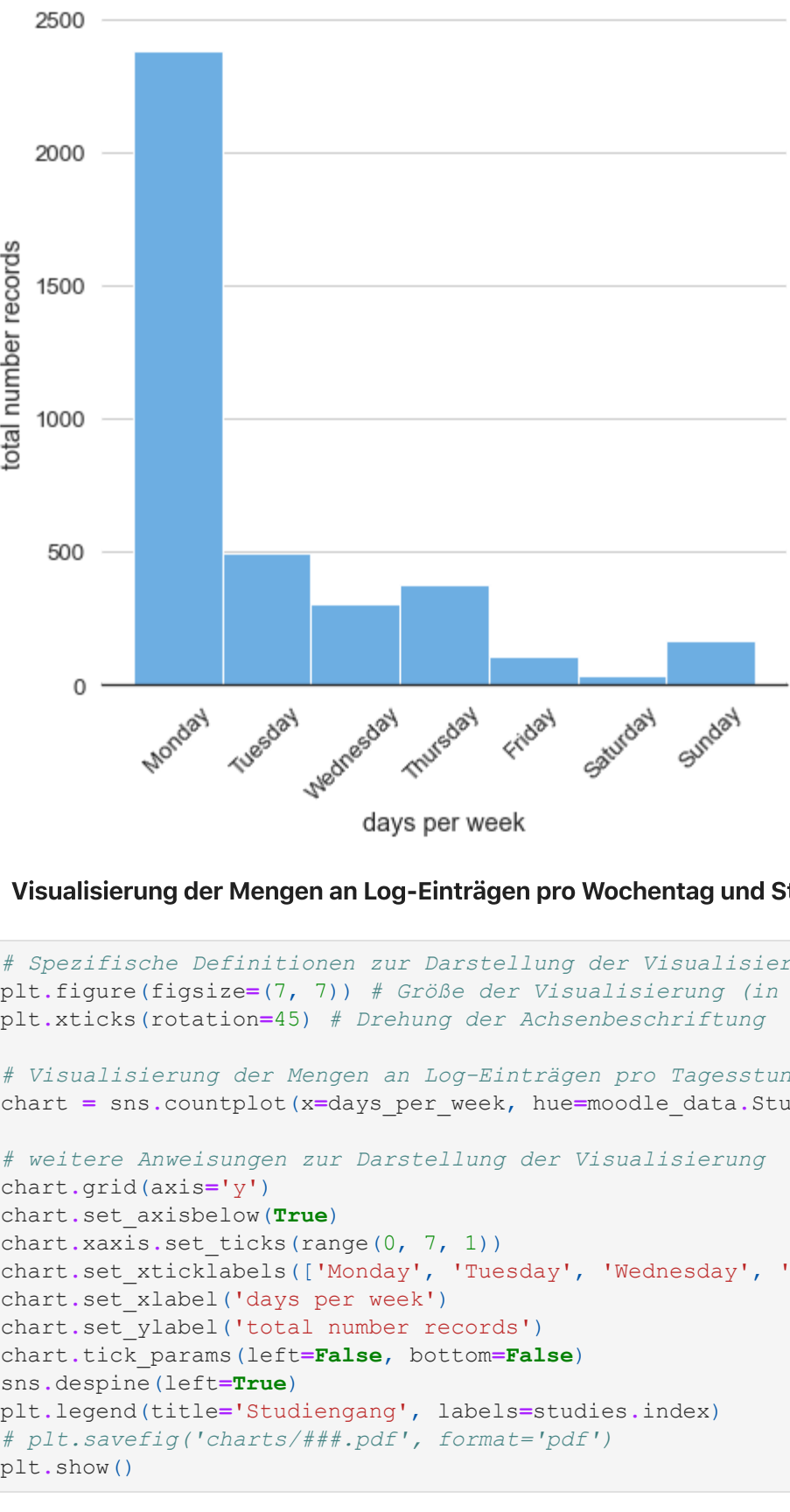
```
# Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(7, 7)) # Größe der Visualisierung (in inch)
plt.xticks(rotation=45) # Drehung der Achsenbeschriftung

# Visualisierung der Mengen an Log-Einträgen pro Wochentag
chart = sns.histplot(days_per_week, bins=7, discrete=True, color='#60A8B2', alpha=1)
```

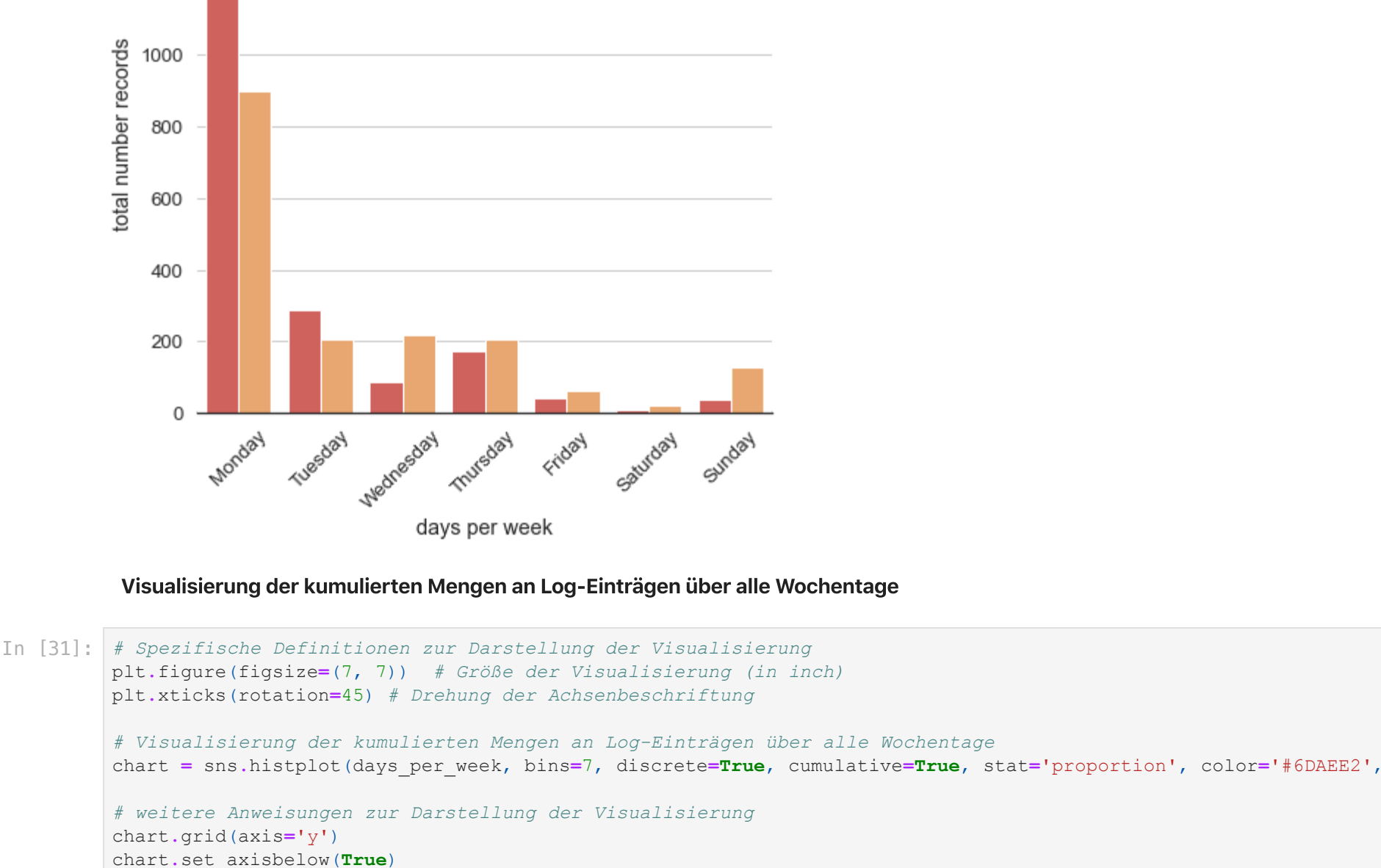
```
# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('x')
chart.set_ylabel('days per week')
chart.set_ylabel('actions')
sns.despine(left=True)
plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



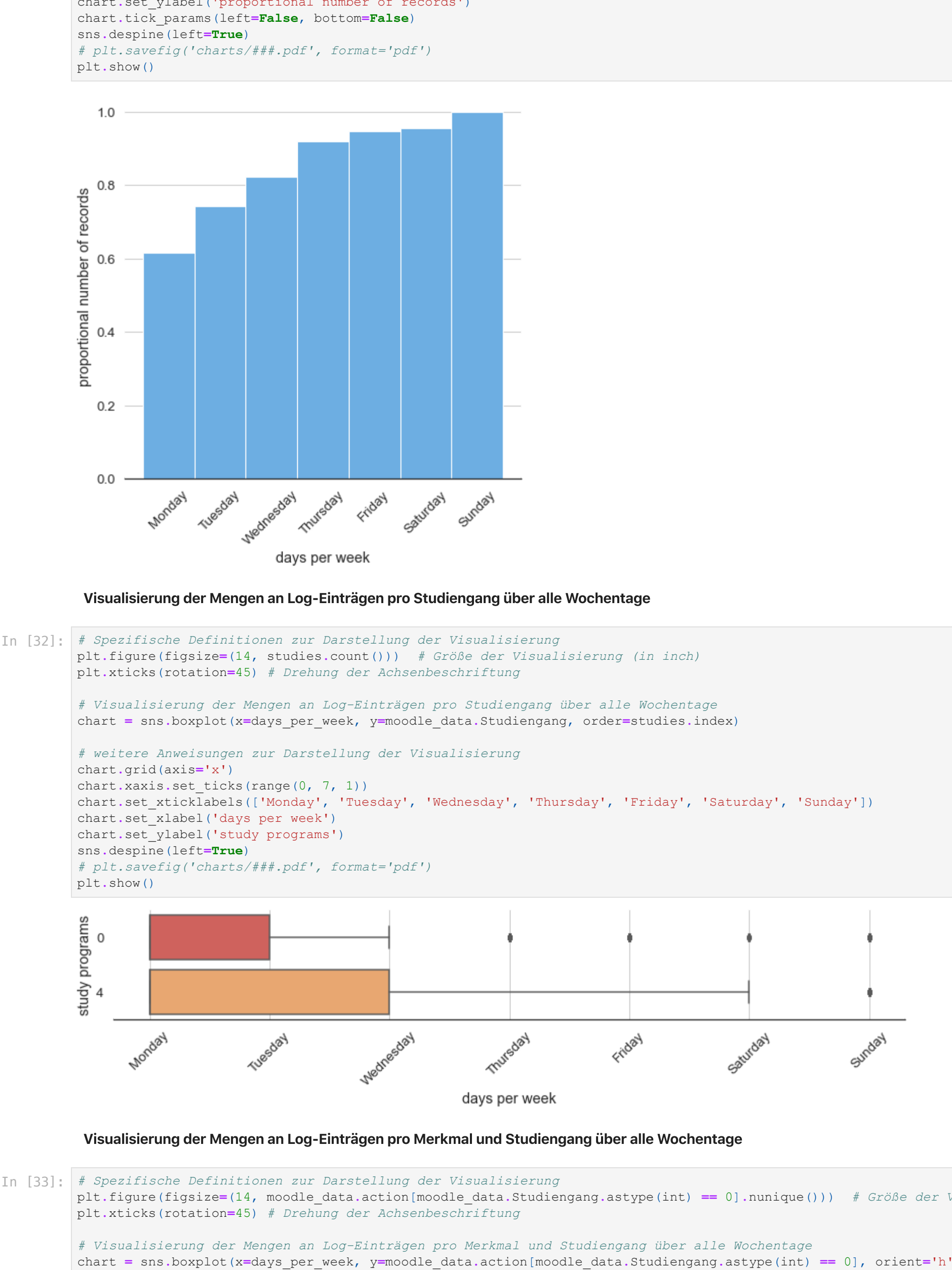




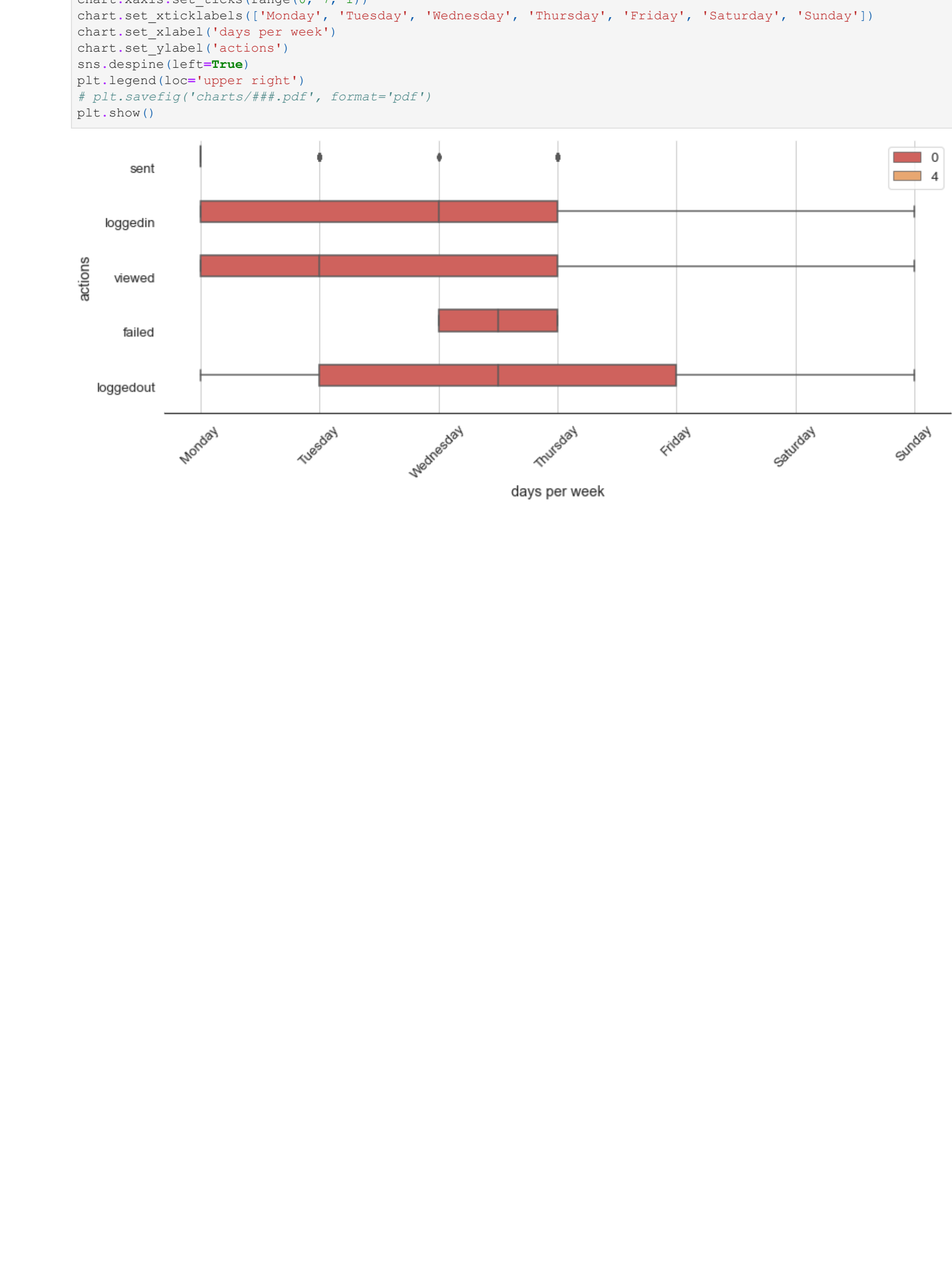
Visualisierung der Mengen an Log-Einträgen pro Wochentag und Studiengang



Visualisierung der kumulierten Mengen an Log-Einträgen über alle Wochentage



Visualisierung der Mengen an Log-Einträgen pro Studiengang über alle Wochentage



Visualisierung der Mengen an Log-Einträgen pro Merkmal und Studiengang über alle Wochentage

