

Analyse des Verhaltens von Benutzer 32

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

Vermutlicher Status Angehöriger des Hochschulpersonals

Definition des Benutzers

```
In [1]: specified_user = 32
```

Import von Bibliotheken und anderen Erweiterungen

```
In [2]: 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 [3]: sns.set_theme(style='white', font_scale=1.2, palette='Spectral')
```

Import der Originaldaten im CSV-Format (optional)

```
In [4]: # file = "vgh_moodle_ws20.csv"
# moodle_data = pd.read_csv(file, delimiter=',')
# moodle_data
```

Herstellung der Verbindung zur MySQL-Datenbank

```
In [5]: user = "root"
password = "root"
host = "localhost"
database = "vgh_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 des Untersuchungs sind zur Datensätze mit einer bestimmten userid
moodle_data = moodle_data[moodle_data.userid == specified_user]
```

```
Out[7]:
```

	courseid	Studiengang	userid	relateduserid	action	eventname	objecttable	objectid	timecreated	course_name
	9272	0	0	32	0	loggedin	/course/event/dashboard_loggedin	user	2163	2020-10-12 13:44:08
	9273	0	0	32	32	viewed	/course/event/course_viewed		0	2020-10-12 13:44:09
	9274	25939	0	32	0	viewed	/course/event/course_viewed		0	2020-10-12 13:44:27
	9275	0	0	32	0	loggedout	/course/event/user_loggedout	user	2163	2020-10-12 13:45:26
	9276	0	0	32	0	loggedin	/course/event/user_loggedin	user	2163	2020-10-12 13:45:38
...
	967390	28223	4	32	0	viewed	/course/event/course_viewed		0	2021-06-03 11:47:52
	967391	0	0	32	0	loggedout	/course/event/user_loggedout	user	2163	2021-06-03 11:48:05
	967435	0	0	32	0	loggedin	/course/event/user_loggedin	user	2163	2021-06-03 12:17:38
	967436	0	0	32	32	viewed	/course/event/dashboard_viewed		0	2021-06-03 12:17:37
	967437	0	0	32	0	loggedout	/course/event/user_loggedout	user	2163	2021-06-03 12:18:01

6608 rows x 11 columns

Menge der unterschiedlichen Werte pro Tabellenmerkmal

```
In [8]: attributes = moodle_data.nunique()
display(attributes)
```

```
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 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.tick_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 [10]: studies = moodle_data.Studiengang.groupby(moodle_data.Studiengang).count()
display(studies)
```

```
In [11]: # 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('total number records')
chart.tick_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 [12]: courses = moodle_data.courseid.groupby(moodle_data.courseid).count()
display(courses.sort_values())
```

```
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 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_ylabel('moodle_data.courseid')
chart.tick_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 [14]: actions = moodle_data.action.groupby(moodle_data.action).count()
display(actions)
```

```
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 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_ylabel('moodle_data.action')
chart.tick_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 [16]: eventtypes = moodle_data.eventname.groupby(moodle_data.eventname).count()
display(eventtypes)
```

```
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 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_ylabel('moodle_data.eventname')
chart.tick_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 [18]: objecttables = moodle_data.objecttable.groupby(moodle_data.objecttable).count()
display(objecttables)
```

```
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 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_ylabel('moodle_data.objecttable')
chart.tick_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 [20]: course_module_types = moodle_data.course_module_type.groupby(moodle_data.course_module_type).count()
display(course_module_types)
```

```
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 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_ylabel('moodle_data.course_module_type')
chart.tick_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 [22]: # Ausgabe der Tagesstunden der protokollierten Log-Einträge
hours_per_day = moodle_data.timecreated.dt.hour.sort_values()
display(hours_per_day)
```

```
Out[22]:
```

781878	5
781879	5
782030	5
781574	5
781573	5
739071	18
739070	18
739069	18
740509	18
739067	18

Name: timecreated, length: 6608, dtype: int64

Visualisierung der Menge der Log-Einträge pro Tagesstunde

```
In [23]: # 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='#6A3D2A', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.xaxis.set_ticks(range(0, 24, 1))
chart.set_ylabel('hours per day')
chart.set_ylabel('total number records')
chart.tick_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 [24]: # 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.cupplot(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.xaxis.set_ticks(range(0, 24, 1))
chart.set_ylabel('hours per day')
chart.set_ylabel('total number records')
chart.tick_params(left=False, bottom=False)
plt.legend(title='Studiengang', labels=studies.index)
plt.savefig('charts/hours_per_day_by_study.pdf', format='pdf')
plt.show()
```

Visualisierung der kumulierten Mengen der Log-Einträge 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 der Log-Einträge über alle Tagesstunden
chart = sns.kdeplot(hours_per_day, bins=24, discrete=True, cumulative=True, stat='proportion', color='#4682B4', alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.xaxis.set_ticks(range(0, 24, 1))
chart.set_ylabel('hours per day')
chart.set_ylabel('proportional number of records')
chart.tick_params(left=False, bottom=False)
sns.despine(left=True)
plt.savefig('charts/cumulative_hours_per_day.pdf', format='pdf')
plt.show()
```

Visualisierung der Menge der Log-Einträge 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 Menge der Log-Einträge pro Studiengang über alle Tagesstunden
chart = sns.lolplot(hours_per_day, ymoodle_data.Studiengang, order=studies.index)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.xaxis.set_ticks(range(0, 24, 1))
chart.set_ylabel('hours per day')
chart.set_ylabel('study programs')
chart.set_ylabel('total number records')
sns.despine(left=True)
plt.savefig('charts/hours_per_day_by_study_lol.pdf', format='pdf')
plt.show()
```

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

```
In [27]: # Spezifische Definitionen zur Darstellung der Visualisierung
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.lolplot(hours_per_day, ymoodle_data.action[moodle_data.Studiengang.astype(int) == 0], orient='h')

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('y')
chart.xaxis.set_ticks(range(0, 24, 1))
chart.set_ylabel('hours per day')
chart.set_ylabel('total number records')
sns.despine(left=True)
plt.savefig('charts/hours_per_day_by_action.pdf', format='pdf')
plt.show()
```

Menge der Log-Einträge pro Wochentag

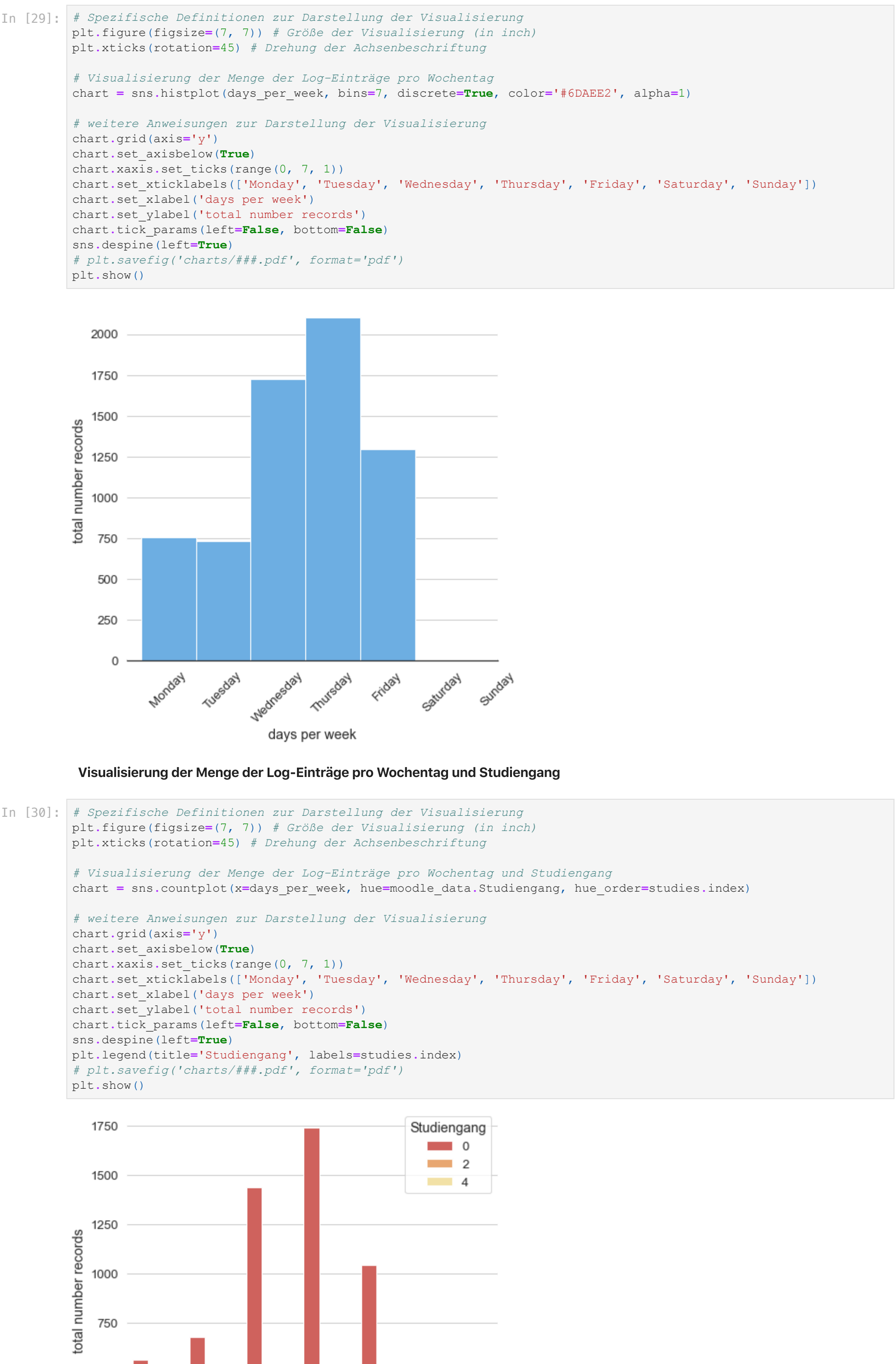
```
In [28]: # Ausgabe der Wochentage der protokollierten Log-Einträge
days_per_week = moodle_data.timecreated.dt.weekday.sort_values()
display(days_per_week)
```

```
Out[28]:
```

7272	0
765659	0
765654	0
765653	0
765652	0
554327	4
554326	4
554253	4
554303	4
406094	4

Name: timecreated, length: 6608, dtype: int64

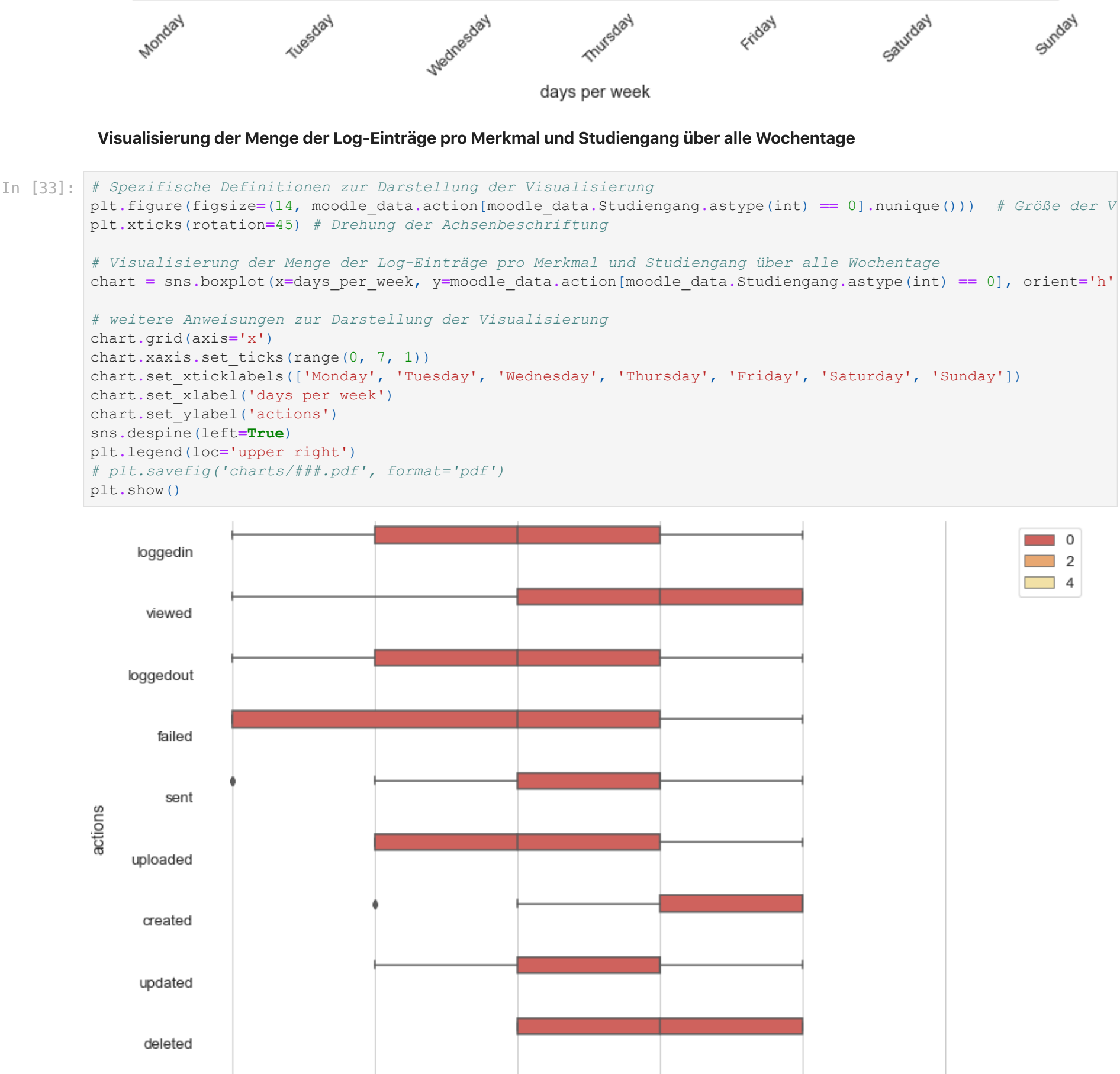
Visualisierung der Menge der Log-Einträge pro Wochentag



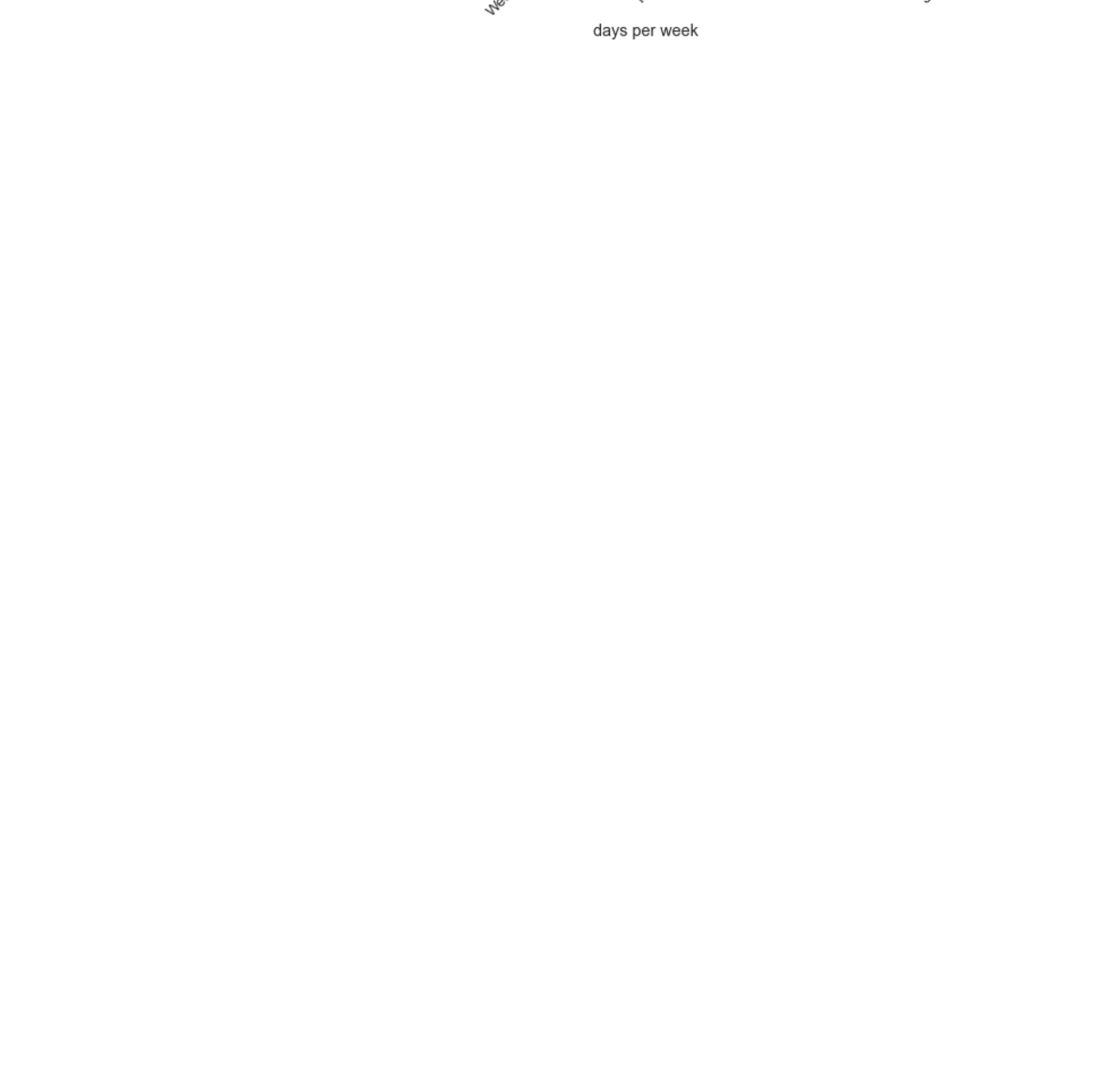
Visualisierung der Menge der Log-Einträge pro Wochentag und Studiengang



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

