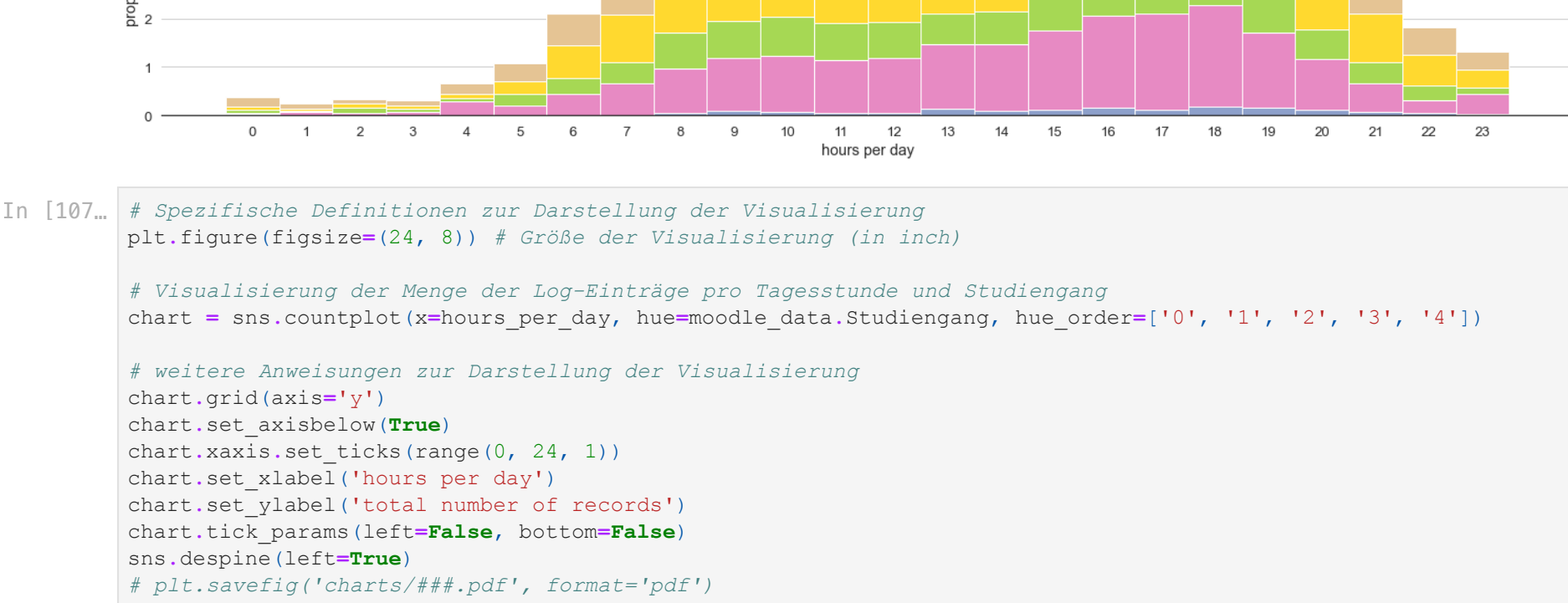



```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, 8)) # Größe der Visualisierung (in Inch)

# Visualisierung der Menge der Log-Einträge pro Tagesstunde und Studiengang
chart = sns.histplot(x=moodle_data.timecreated.dt.hour, bins=24, discrete=True, stat='percent', hue=moodle_data
```

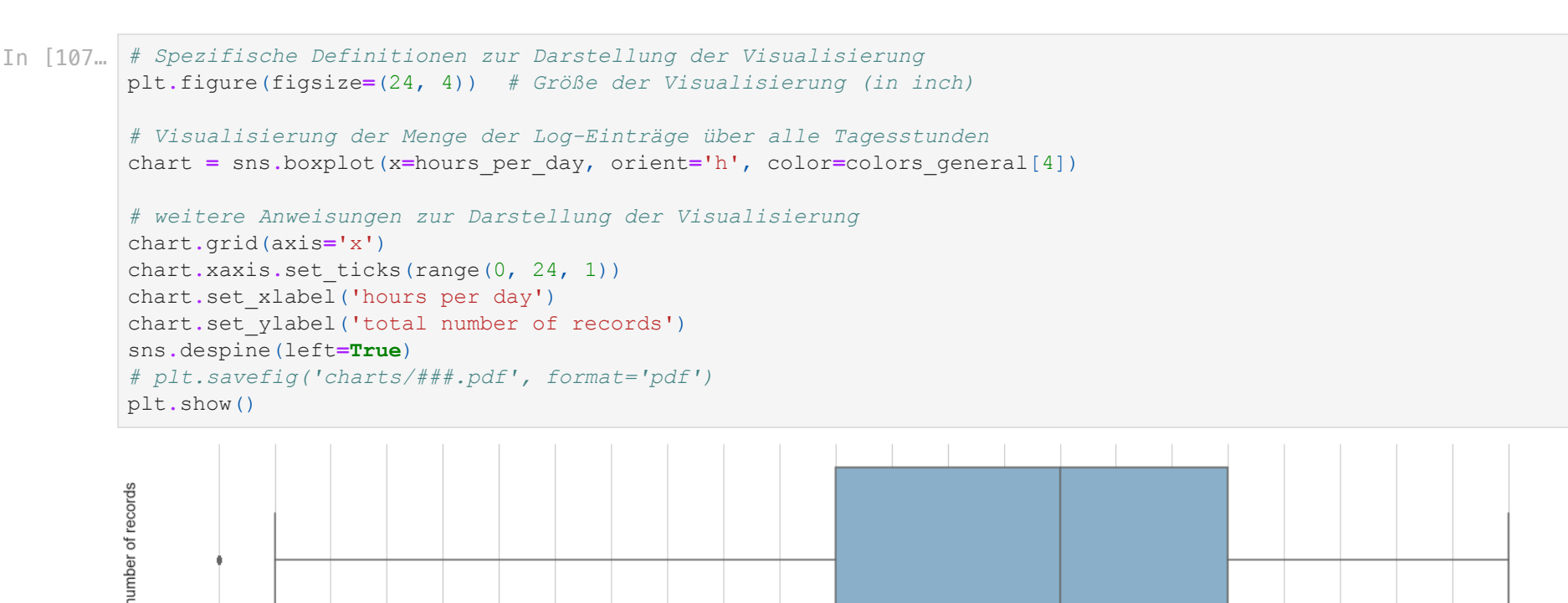
```
# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_axisbelow(True)
chart.set_xlabel('hours per day')
chart.set_ylabel('proportional number of records in percent')
chart.xaxis.set_ticks(range(0, 24, 1))
chart.yaxis.set_major_locator(MultipleLocator(1))
chart.tick_params(left=False,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



```
In [107]: # 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.cupmplot(x=hours_per_day, hue=moodle_data.Studiengang, hue_order=['0', '1', '2', '3', '4'])

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

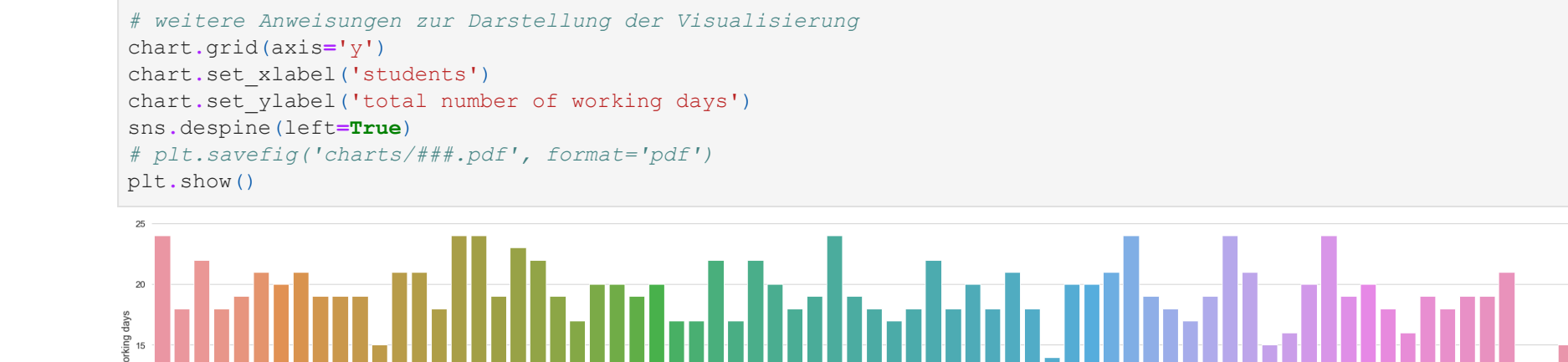


Menge der Log-Einträge über alle Tagesstunden

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, 4)) # Größe der Visualisierung (in Inch)

# Visualisierung der Menge der Log-Einträge über alle Tagesstunden
chart = sns.boxplot(x=hours_per_day, orient='h', color=colors_general[4])

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



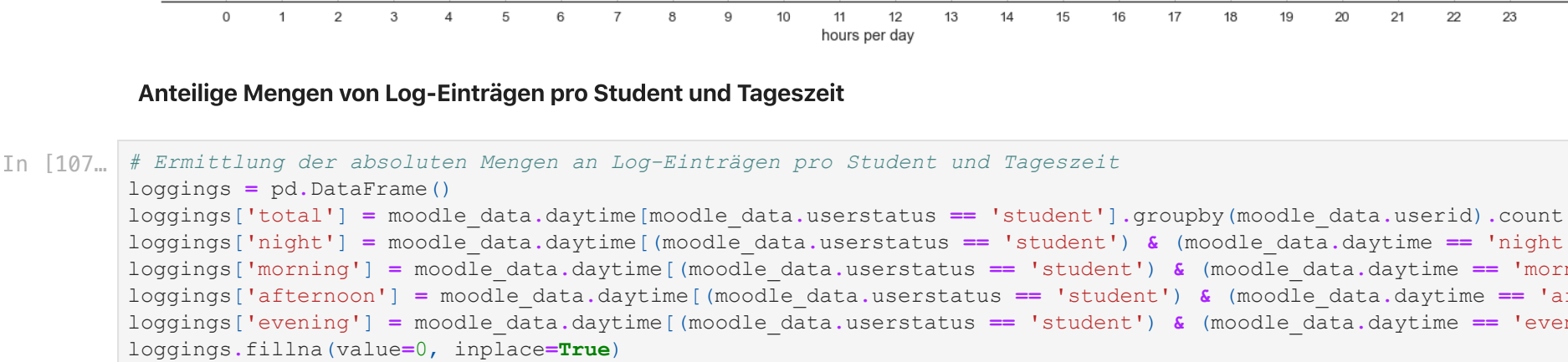
Anmerkung: Die Benutzeraktivitäten orientierten sich weitgehend an einem normalen Arbeitstag, die Hälfte aller Interaktionen wurden im Zeitraum von 11 bis 19 Uhr verzeichnet. Vor 9 Uhr wurden vergleichsweise nur wenig Aktivitäten festgestellt. Ab 19 Uhr nahm die Menge an protokollierten Interaktionen überraschenderweise deutlich ab.

Menge der Log-Einträge pro Student und Tagesstunde Betrachtet wird nur an wie vielen einzelnen Tagesstunden ein Student aktiv war und nicht wie lange ein Student insgesamt in Stunden bemessen aktiv war.

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 8)) # Größe der Visualisierung (in Inch)

# Ermittlung der Menge der Arbeitstage pro Student
result = moodle_data.timecreated.dt.hour.groupby(moodle_data.userid[moodle_data.userid == 'student']).nunique()
# Visualisierung der Menge der Arbeitstage pro Student
chart = sns.barplot(x=result.index.astype(int), y=result)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='y')
chart.set_xlabel('students')
chart.set_ylabel('total number of working days')
sns.despine(left=True)
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



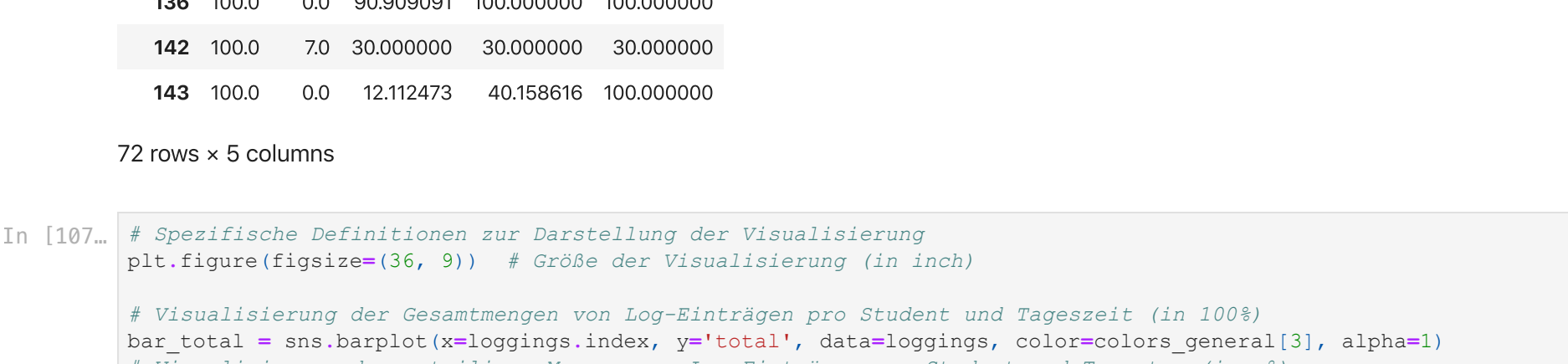
Anmerkung: Auch wenn die Aktivitäten über den Tag sich im wesentlichen auf die normalen Arbeitszeiten konzentrieren, so waren die allermeisten Studenten im Durchschnitt über den ganzen Tag verteilt aktiv.

Menge der Log-Einträge über alle Tagesstunden pro Studiengang

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(24, 8)) # Größe der Visualisierung (in Inch)

# Visualisierung der Menge der Log-Einträge über alle Tagesstunden pro Studiengang
chart = sns.boxplot(x=hours_per_day, y=moodle_data.Studiengang, order=['0', '1', '2', '3', '4'])

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



Anteilung der Log-Einträge pro Student und Tageszeit

```
In [107]: # Ermittlung der anteiligen Mengen an Log-Einträgen pro Student und Tageszeit
loggings = pd.DataFrame()
loggings['total'] = moodle_data.daytime[moodle_data.userid == 'student'].groupby(moodle_data.userid).count()
loggings['night'] = moodle_data.daytime[moodle_data.userid == 'student'] & (moodle_data.daytime == 'night')
loggings['morning'] = moodle_data.daytime[moodle_data.userid == 'student'] & (moodle_data.daytime == 'morn')
loggings['afternoon'] = moodle_data.daytime[moodle_data.userid == 'student'] & (moodle_data.daytime == 'aft')
loggings['evening'] = moodle_data.daytime[moodle_data.userid == 'student'] & (moodle_data.daytime == 'even')
loggings.fillna(value=0, inplace=True)

# Ermittlung der anteiligen Mengen an Log-Einträgen pro Student und Tageszeit
# Aus Darstellungsgründen werden die anteiligen Mengen aufsummiert
loggings['morning'] = [i / j * 100 for i, j in zip(loggings['morning'], loggings['total'])]
loggings['afternoon'] = [i / j * 100 for i, j in zip(loggings['afternoon'], loggings['total'])] # nicht aufsummiert
loggings['evening'] = [i / j * 100 for i, j in zip(loggings['evening'], loggings['total'])] # nicht aufsummiert
loggings['night'] = [i / j * 100 for i, j in zip(loggings['night'], loggings['total'])] # nicht aufsummiert
loggings['total'] = [i / j * 100 for i, j in zip(loggings['total'], loggings['total'])]

display(loggings)
```

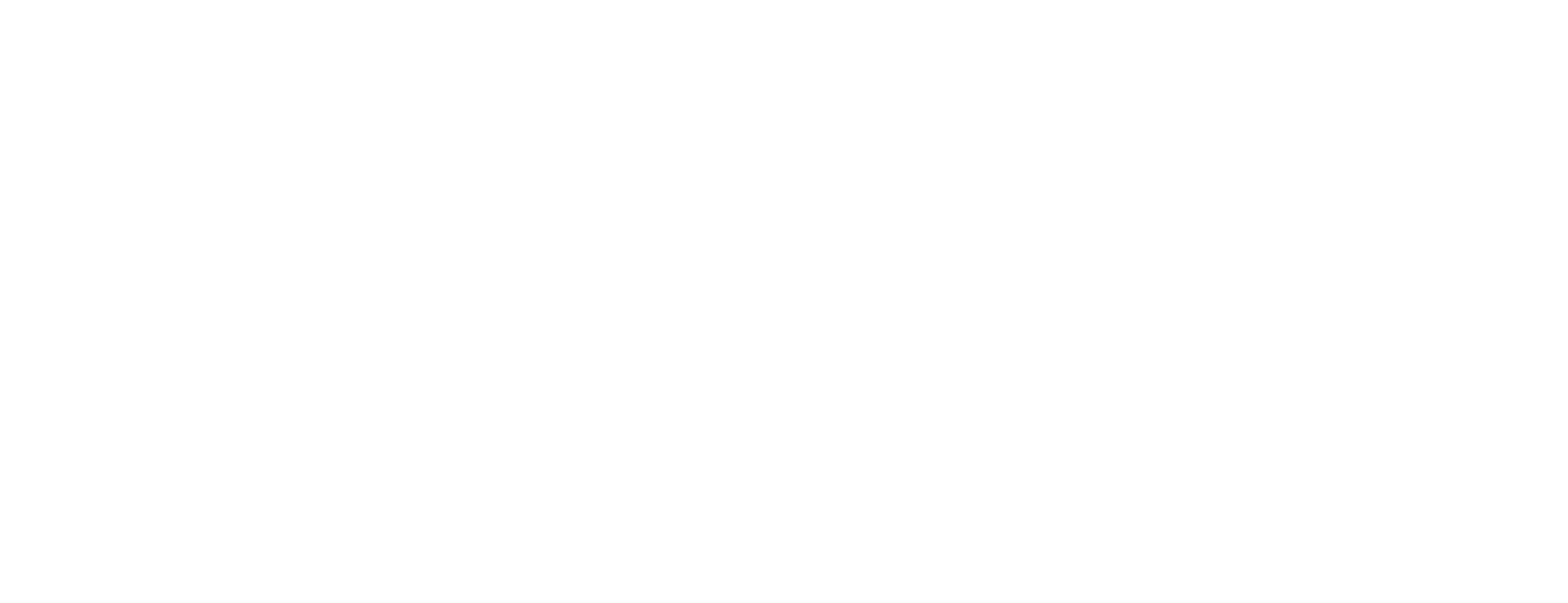
	total	night	morning	afternoon	evening
userid					
1	100.0	23.10	30.349828	77.981889	94.023286
13	100.0	6.0	35.450346	75.173210	99.861432
18	100.0	76.0	46.866521	81.445905	96.157735
19	100.0	25.0	43.259488	63.523957	99.570668
20	100.0	55.0	43.956830	86.833644	99.069216
...
132	100.0	42.0	45.072318	80.827447	98.597286
134	100.0	52.0	28.778113	73.644416	98.876647
136	100.0	0.0	90.909091	100.000000	100.000000
142	100.0	7.0	30.000000	30.000000	30.000000
143	100.0	0.0	12.112473	40.158616	100.000000

72 rows x 5 columns

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der Gesamtmenge von Log-Einträgen pro Student und Tageszeit (in 100%)
bar_total = sns.barplot(x=loggings.index, y='total', data=loggings, color=colors_general[3], alpha=1)
bar_night = sns.barplot(x=loggings.index, y='night', data=loggings, color=colors_general[2], alpha=1)
bar_afternoon = sns.barplot(x=loggings.index, y='afternoon', data=loggings, color=colors_general[1], alpha=1)
bar_morning = sns.barplot(x=loggings.index, y='morning', data=loggings, color=colors_general[0], alpha=1)

# weitere Anweisungen zur Darstellung der Visualisierung
chart.grid(axis='x')
chart.set_xlabel('students')
chart.set_ylabel('proportional number of records in percent')
plt.legend(title='daytime', handles=[bar1, bar2, bar3, bar4], bbox_to_anchor=(1, 1))
sns.despine(left=True)
# plt.savefig('charts/###.pdf', format='pdf')
plt.show()
```



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



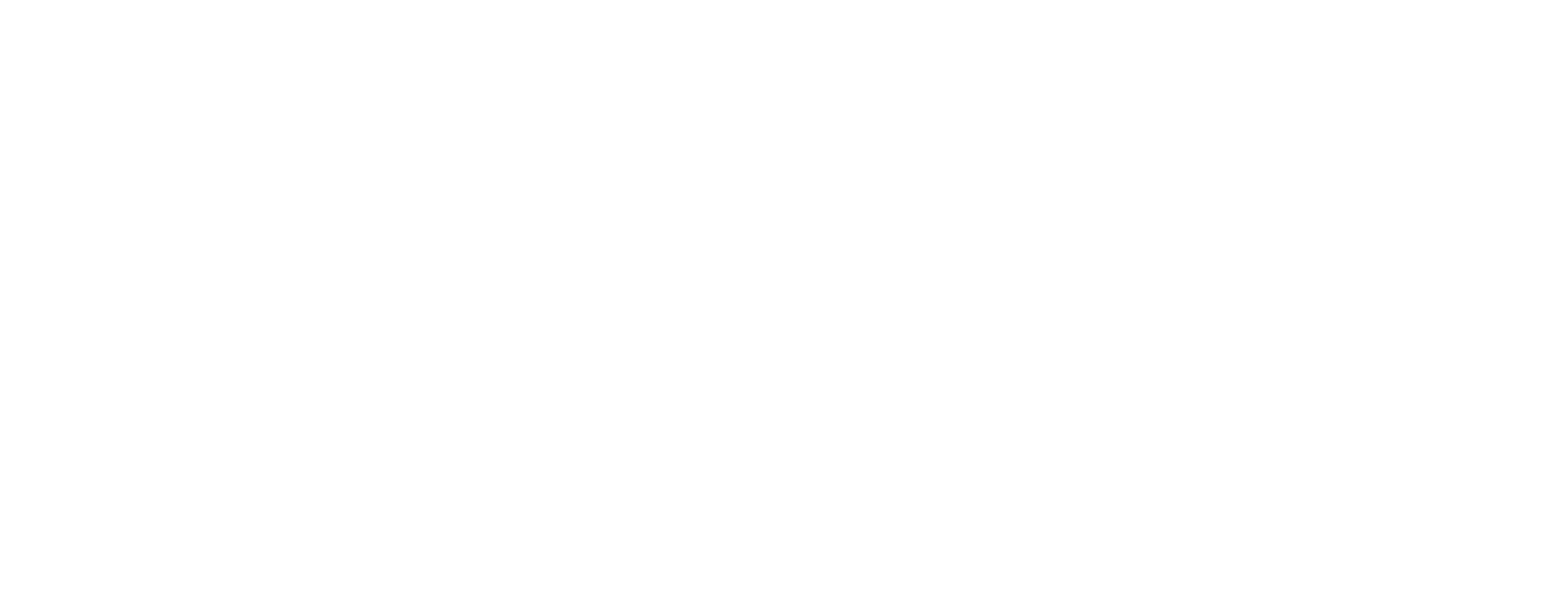
Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



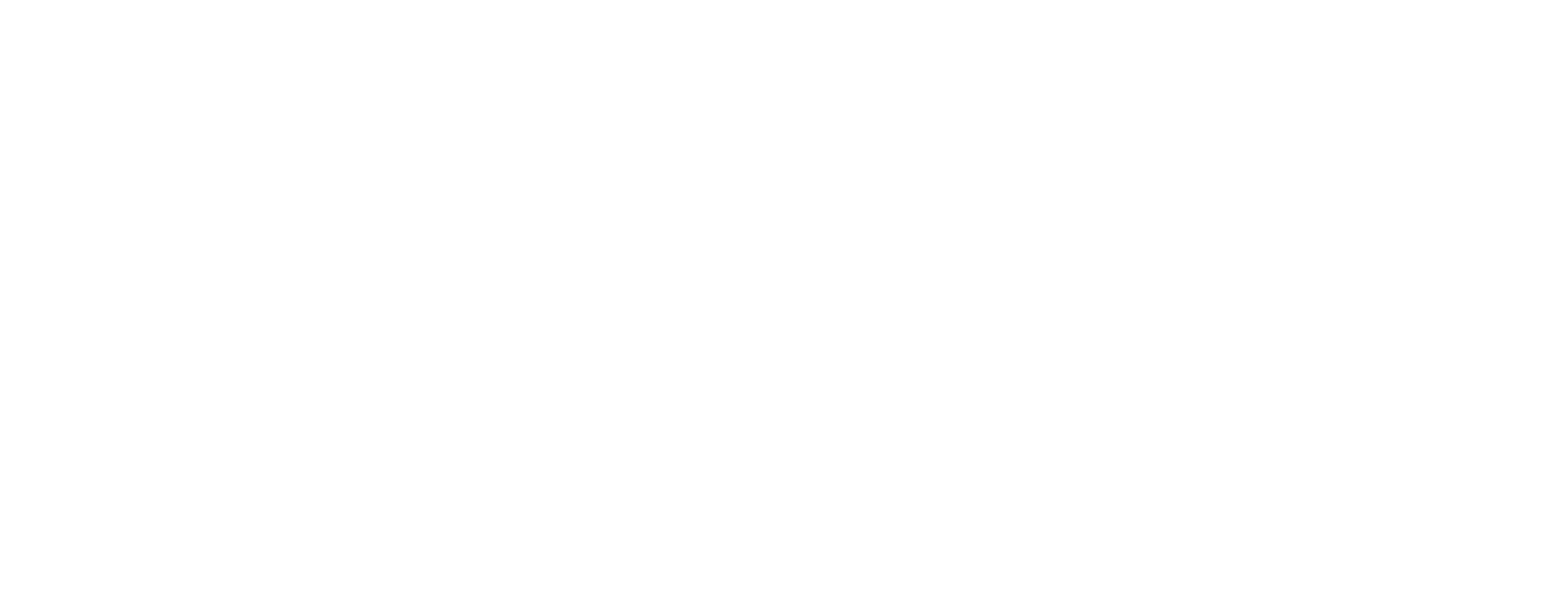
Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



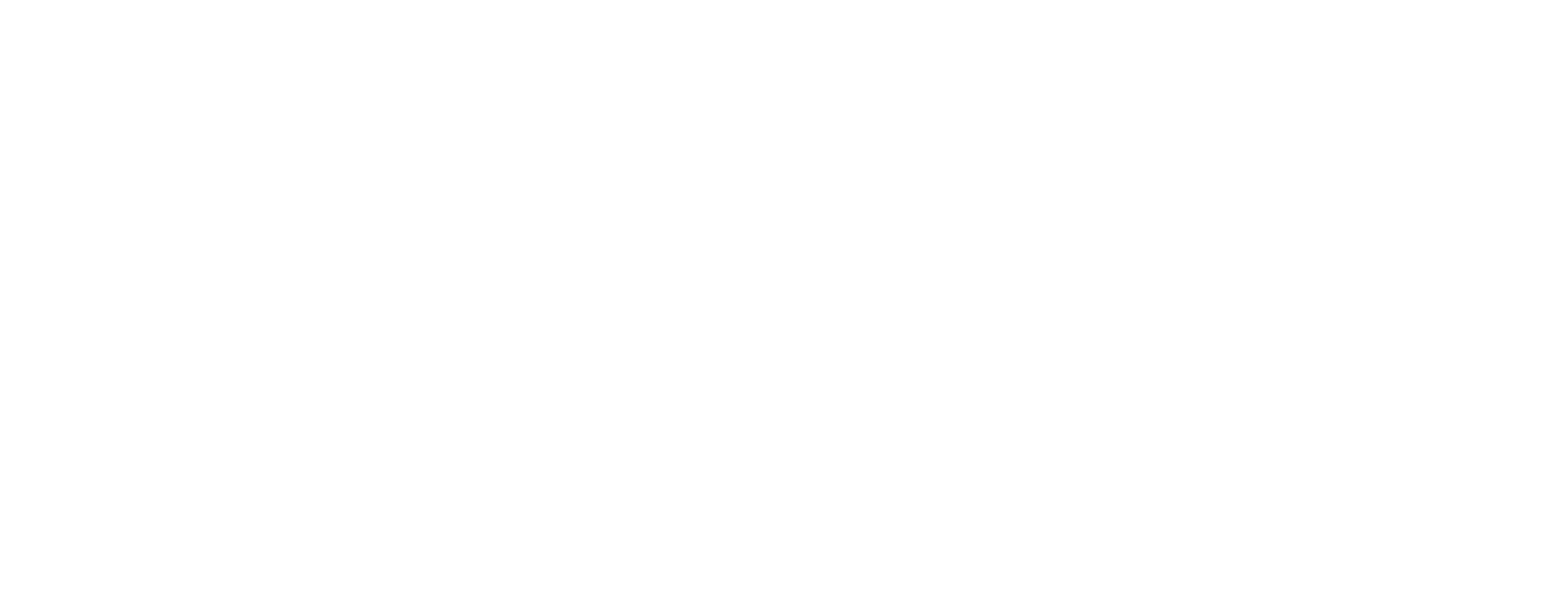
Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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



Anmerkung: Diese und auch die nachfolgende Visualisierung zeigen noch einmal deutlich, dass die Studenten überwiegend zu normalen Arbeitszeiten aktiv waren.

Absolute Mengen von Log-Einträgen pro Student und Tageszeit

```
In [107]: # Spezifische Definitionen zur Darstellung der Visualisierung
plt.figure(figsize=(36, 9)) # Größe der Visualisierung (in Inch)

# Visualisierung der absoluten Mengen von Log-Einträgen pro Student und Tageszeit
chart = sns.histplot(x=moodle_data.userid[moodle_data.userid == 'student'].astype(str), hue=moodle_data.daytime)

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