In [1]:
```python
import pandas as pd
import numpy as np
import csv
import seaborn as sns
import datetime
from IPython.display import display
```

In [2]:
```python
project_states = pd.read_csv('project-states-15-sep-2021.csv', engine="python", delimiter ='\;')
class_7d = pd.read_csv('classifications-7-days-kobi-16-sep-2021.csv', delimiter = ';')
project_state_dict = {
    0: 'paused',
    1: 'finished'
}

project_states['state'] = project_states['state'].apply(lambda x: project_state_dict[x] if x in project_state_c
```

In [3]:
```python
project_states.head()
```

Out[3]:

	id	display_name	launch_date	state
0	5856	Southern Weather Discovery	2018-10-19 13:05:04.568858	paused
1	11402	FrogSong	2021-01-26 15:00:59.785477	active
2	10232	Snowflake ID	2021-08-10 15:43:06.356364	paused
3	7929	Planet Hunters TESS	2018-12-06 18:32:10.688484	active
4	6615	Rodent Little Brother: Secret Lives of Mice	2019-07-08 13:56:47.836319	paused

In [4]:
```python
class_7d
```

Out[4]:

	event_id	project_id	user_id	created_at	session_time
0	360139554	9532	2355710	2021-09-15 20:43:50.74287	799
1	360140363	9532	2355710	2021-09-15 20:46:41.184972	170
2	360141514	9532	2355710	2021-09-15 20:50:57.900403	256
3	360142269	9532	2355710	2021-09-15 20:53:49.11485	128
4	360142976	9532	2355710	2021-09-15 20:56:24.14991	157
...	...	...	...	...	...
1543402	360050059	15650	2355961	2021-09-15 15:29:08.943041	3
1543403	360050060	7929	503779	2021-09-15 15:29:09.923338	7
1543404	360050061	4996	2279693	2021-09-15 15:29:09.923338	12
1543405	360050062	7273	2355675	2021-09-15 15:29:09.923338	17
1543406	360050064	2218	2343479	2021-09-15 15:29:10.946471	2

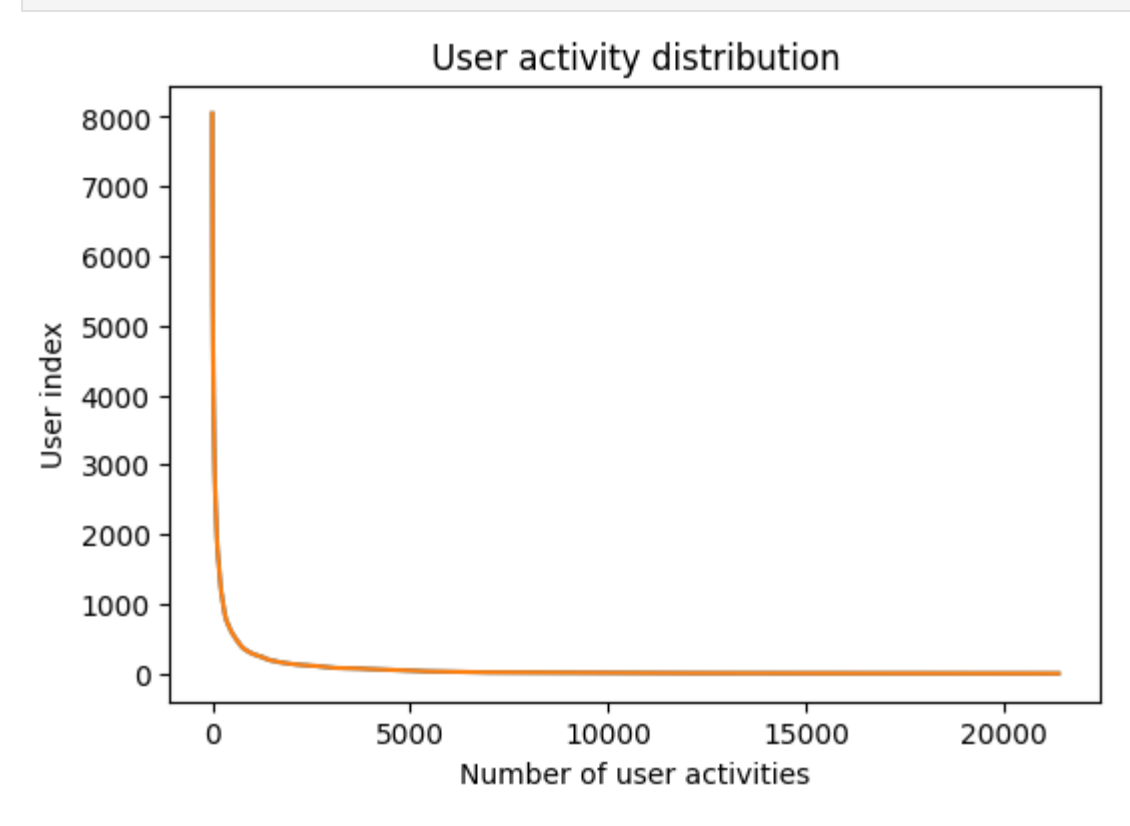
1543407 rows × 5 columns

In [5]:
```python
def plot_with_interpolation(x_data, y_data, title, xlabel, ylabel,figsize=(6, 4)):
    fig, ax = plt.subplots(figsize=figsize)

    y_axis = np.arange(0, x_data)
    x_axis = np.array(list(y_data))
    f = interpolate.interp1d(y_axis, x_axis)
    xnew = np.arange(0, x_data - 1, 0.1)
    ynew = f(xnew)

    ax.plot(x_axis, y_axis, ynew, xnew, '-')
    ax.set_title(title)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
    plt.show()
```

User activity distribution

We plot the number of activities each user had and plot a graph, from the result it is clear that we have a small subset of super users and the mean activity of users is considerably low compare to them.

In [6]:
```python
import matplotlib.pyplot as plt
from scipy import interpolate

user_id_count_dict = class_7d['user_id'].value_counts().to_dict()
users_count = len(class_7d['user_id'].unique())
plot_title = 'User activity distribution'
plot_xlabel = 'Number of user activities'
plot_ylabel = 'User index'
plot_with_interpolation(users_count, user_id_count_dict.values(), plot_title, plot_xlabel, plot_ylabel)
```



entries each day of the week

Along with user activity we would like to observe the histogram of activities, unique users and unique project related to the day of the week, this can help us understand when the masses of users spend most time on the site.
In the future we might be able to derive feature that will help us suggest project, for example we probably dont want to suggest a long and demending project to a user before saturday\sunday as they might just leave it and won't return to it.

In [7]:
```python
class_7d['created_at'] = pd.to_datetime(class_7d['created_at'])
class_7d['date'] = class_7d['created_at'].apply(lambda date: date.strftime("%m/%d/%Y"))
class_7d
```

Out[7]:

	event_id	project_id	user_id	created_at	session_time	date
0	360139554	9532	2355710	2021-09-15 20:43:50.742870	799	09/15/2021
1	360140363	9532	2355710	2021-09-15 20:46:41.184972	170	09/15/2021
2	360141514	9532	2355710	2021-09-15 20:50:57.900403	256	09/15/2021
3	360142269	9532	2355710	2021-09-15 20:53:49.114850	128	09/15/2021
4	360142976	9532	2355710	2021-09-15 20:56:24.149910	157	09/15/2021
...	...	...	...	...	...	...
1543402	360050059	15650	2355961	2021-09-15 15:29:08.943041	3	09/15/2021
1543403	360050060	7929	503779	2021-09-15 15:29:09.923338	7	09/15/2021
1543404	360050061	4996	2279693	2021-09-15 15:29:09.923338	12	09/15/2021
1543405	360050062	7273	2355675	2021-09-15 15:29:09.923338	17	09/15/2021
1543406	360050064	2218	2343479	2021-09-15 15:29:10.946471	2	09/15/2021

1543407 rows × 6 columns

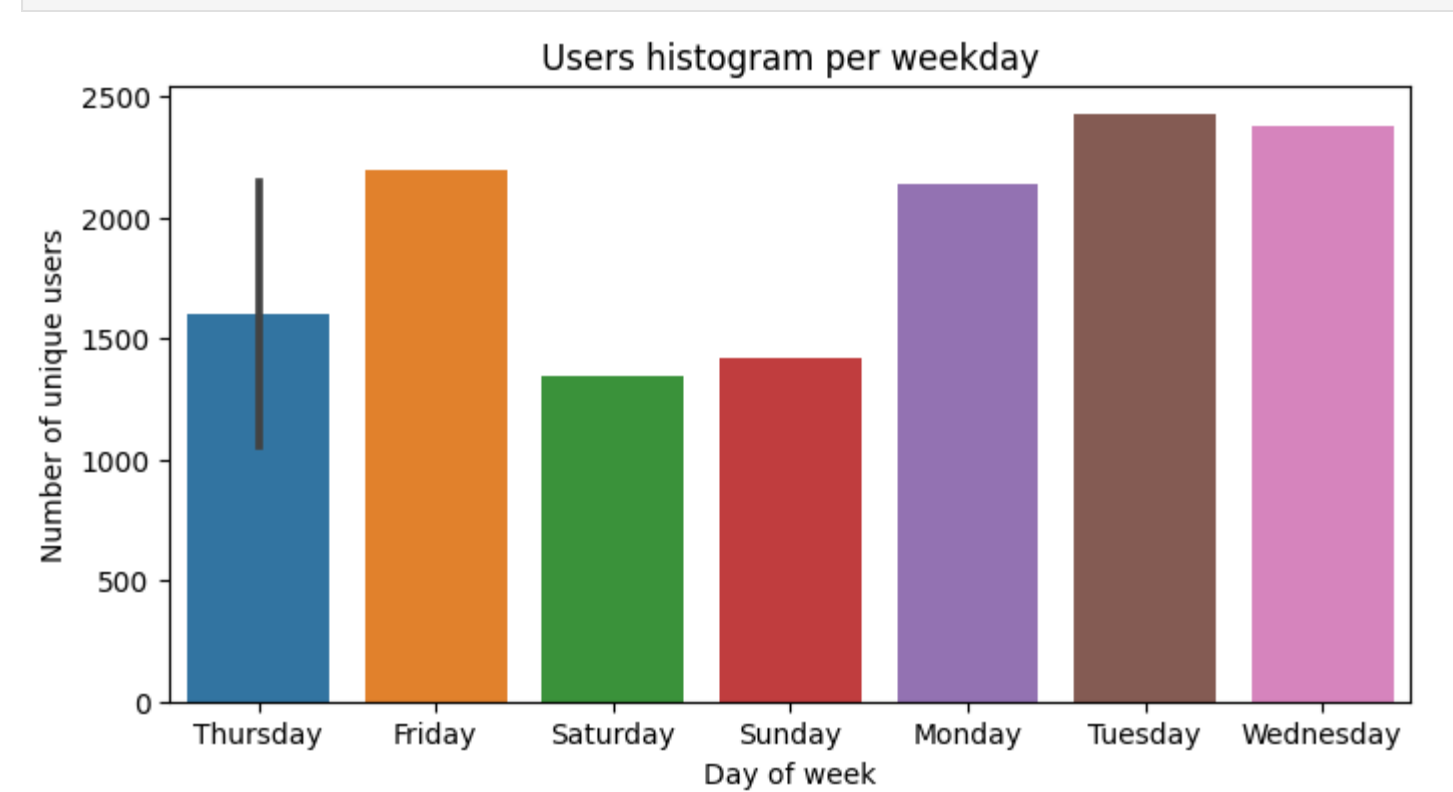
In [8]:
```python
date_groups = class_7d.groupby(['date'])
weekdays_summary = {}
datetime_weekday_dict = {
    0: 'Monday',
    1: 'Tuesday',
    2: 'Wednesday',
    3: 'Thursday',
    4: 'Friday',
    5: 'Saturday',
    6: 'Sunday'
}
weekdays = []
users = []
projects = []
entries = []

for name, group in date_groups:
    unique_users = group['user_id'].unique()
    unique_projects = group['project_id'].unique()
    total_activity = len(group)
    data = {
        'users': len(unique_users),
        'projects': len(unique_projects),
        'total_entries': total_activity
    }
    weekday = datetime.datetime.strptime(name, "%m/%d/%Y").weekday()
    weekdays.append(datetime_weekday_dict[weekday])
    users.append(len(unique_users))
    projects.append(len(unique_projects))
    entries.append(total_activity)
    weekdays_summary[datetime_weekday_dict[weekday]] = data
```

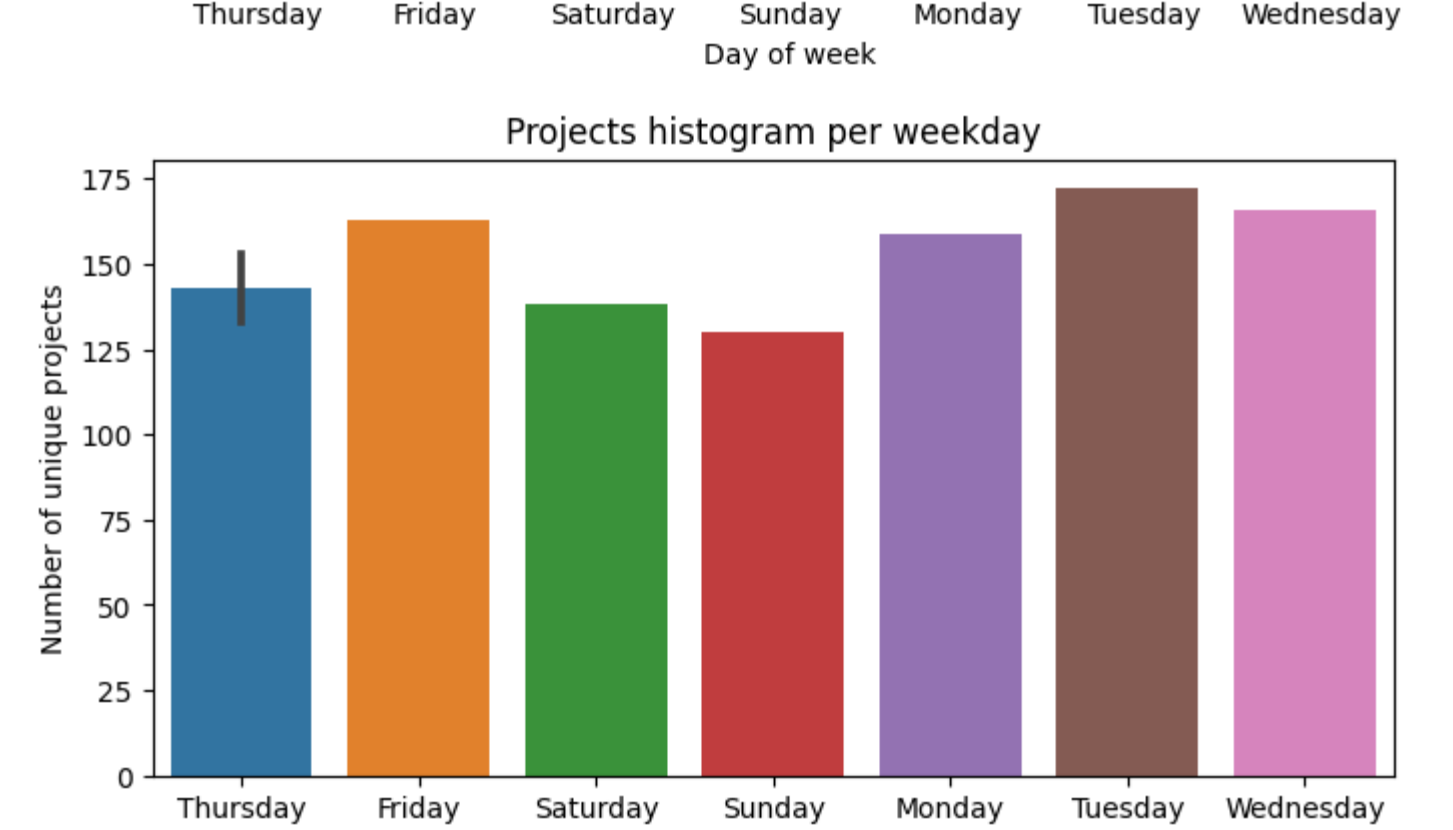
In [9]:
```python
weekdays_summary
```

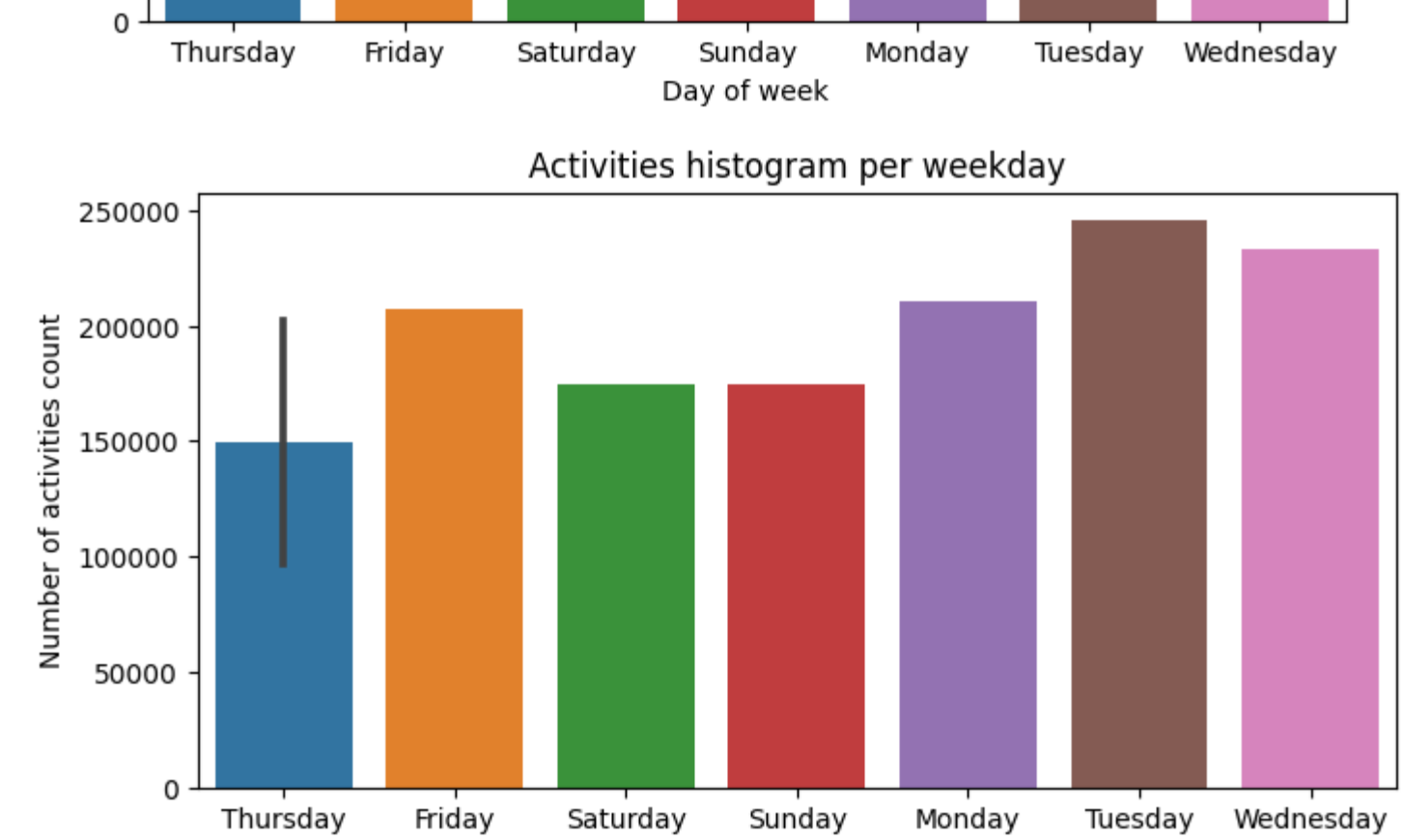
Out[9]:
```
{'Thursday': {'users': 1052, 'projects': 133, 'total_entries': 96726},
 'Friday': {'users': 2198, 'projects': 163, 'total_entries': 206727},
 'Saturday': {'users': 1341, 'projects': 138, 'total_entries': 174653},
 'Sunday': {'users': 1419, 'projects': 130, 'total_entries': 174466},
 'Monday': {'users': 2140, 'projects': 159, 'total_entries': 210245},
 'Tuesday': {'users': 2424, 'projects': 172, 'total_entries': 245281},
 'Wednesday': {'users': 2381, 'projects': 166, 'total_entries': 233132}}
```

In [10]:
```python
def sns_barplot(x, y, xlabel, ylabel, title, figsize=(8, 4)):

    fig, ax = plt.subplots(figsize=figsize)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
    ax.set_title(title)
    sns.barplot(x=x, y=y, ax=ax)
    plt.show()
```

In [11]:
```python
sns_barplot(weekdays, users, "Day of week", "Number of unique users", "Users histogram per weekday")
sns_barplot(weekdays, projects, "Day of week", "Number of unique projects", "Projects histogram per weekday")
sns_barplot(weekdays, entries, "Day of week", "Number of activities count", "Activities histogram per weekday")
```





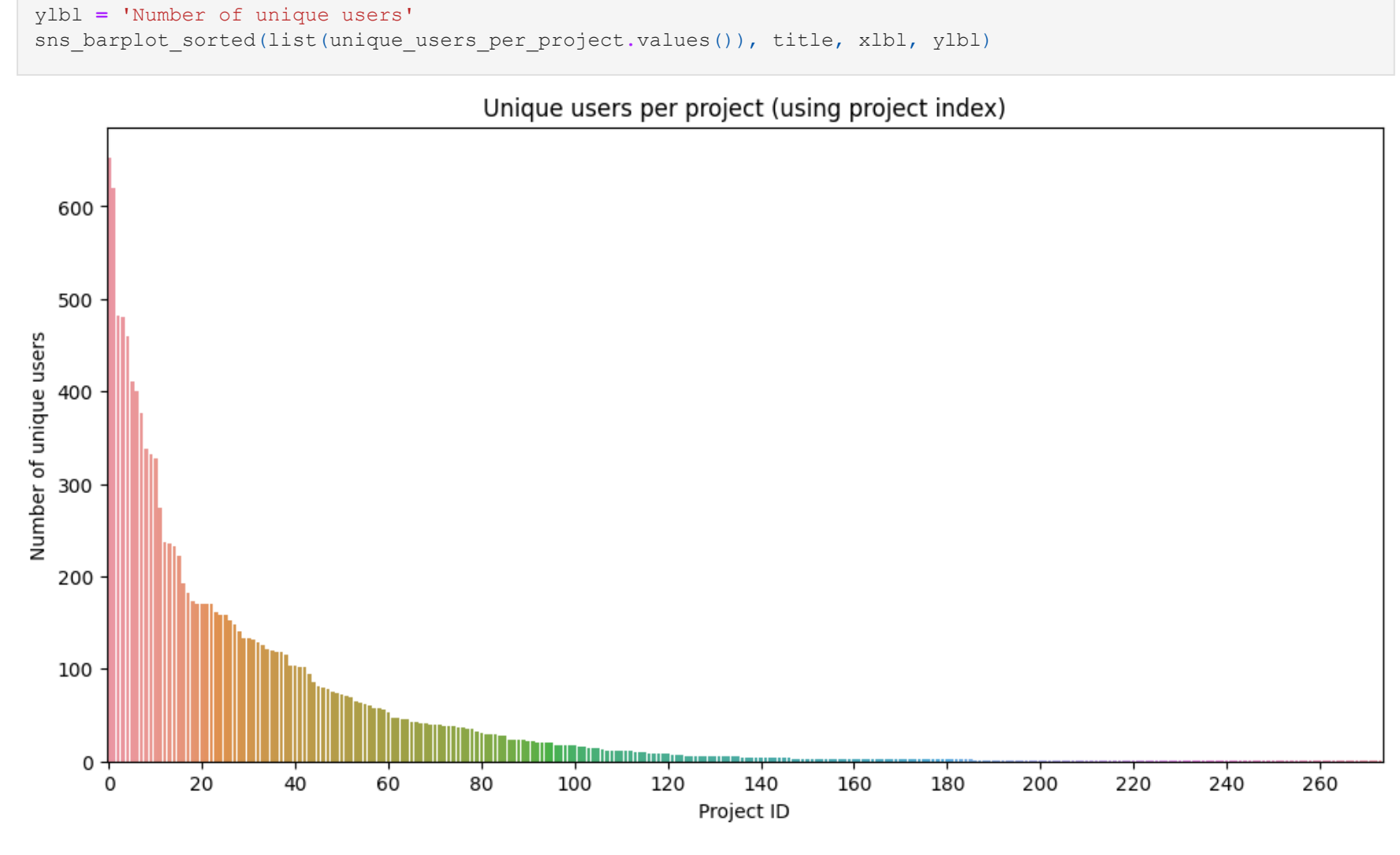


Unique users per project

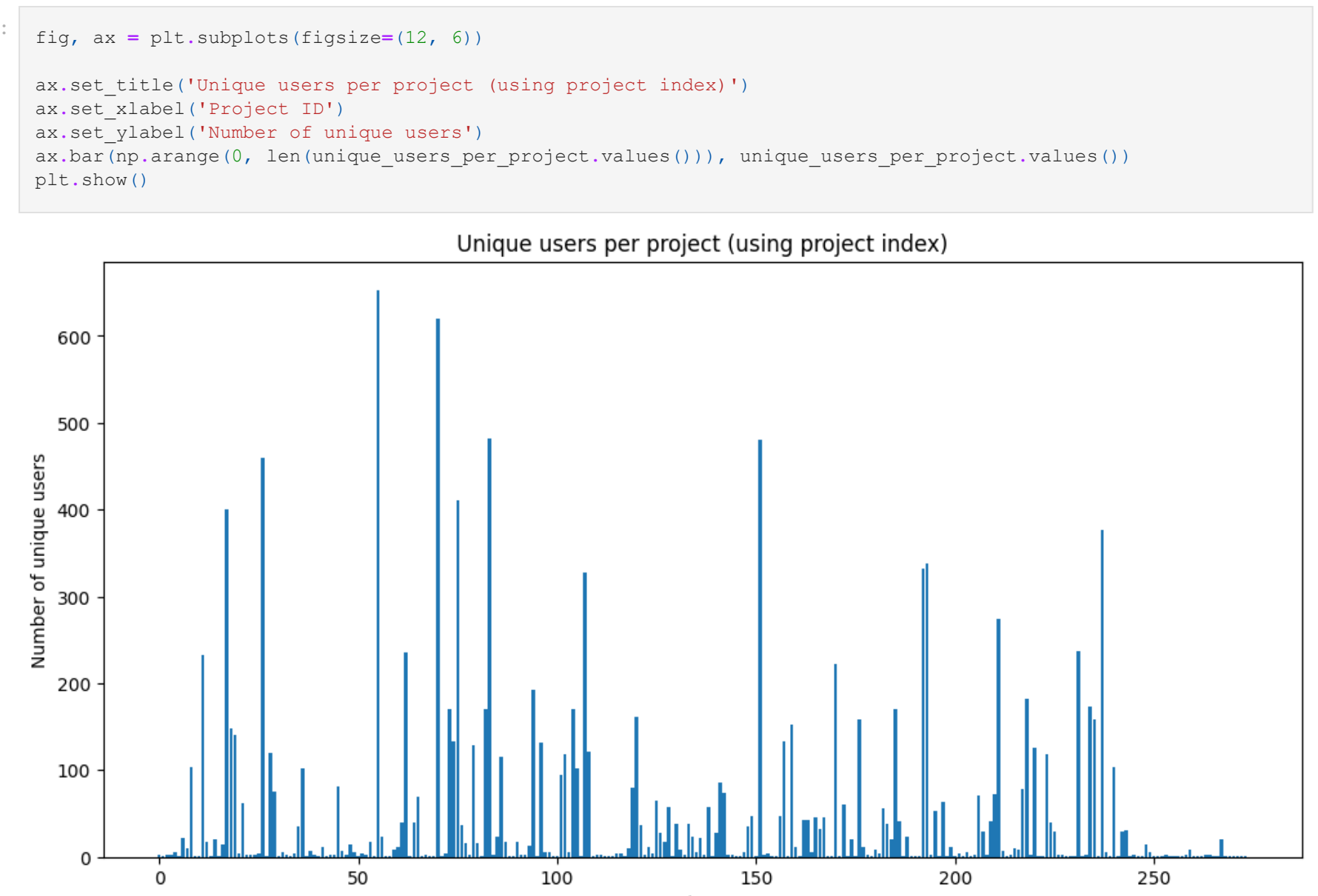
In [12]:
```python
unique_users_per_project = class_7d.groupby(['project_id'])['user_id'].nunique().to_dict()
```

In [13]:
```python
def sns_barplot_sorted(data, title, xlbl, ylbl, xticks_steps=20, desc=True, x=None):
    fig, ax = plt.subplots(figsize=(12, 6))

    sorted_data = list(data)
    sorted_data.sort(reverse=desc)
    ax.set_title(title)
    ax.set_xlabel(xlbl)
    ax.set_ylabel(ylbl)
    if x is None:
        x = list(np.arange(0, len(sorted_data)))
    g = sns.barplot(x=x, y=sorted_data)
    g.set_xticks(np.arange(0, len(sorted_data), xticks_steps))
    plt.show()
```

In [14]:
```python
title = 'Unique users per project (using project index)'
xlbl = 'Project ID'
ylbl = 'Number of unique users'
sns_barplot_sorted(list(unique_users_per_project.values()), title, xlbl, ylbl)
```



In [15]:
```python
fig, ax = plt.subplots(figsize=(12, 6))

ax.set_title('Unique users per project (using project index)')
ax.set_xlabel('Project ID')
ax.set_ylabel('Number of unique users')
ax.bar(np.arange(0, len(unique_users_per_project.values())), unique_users_per_project.values())
plt.show()
```



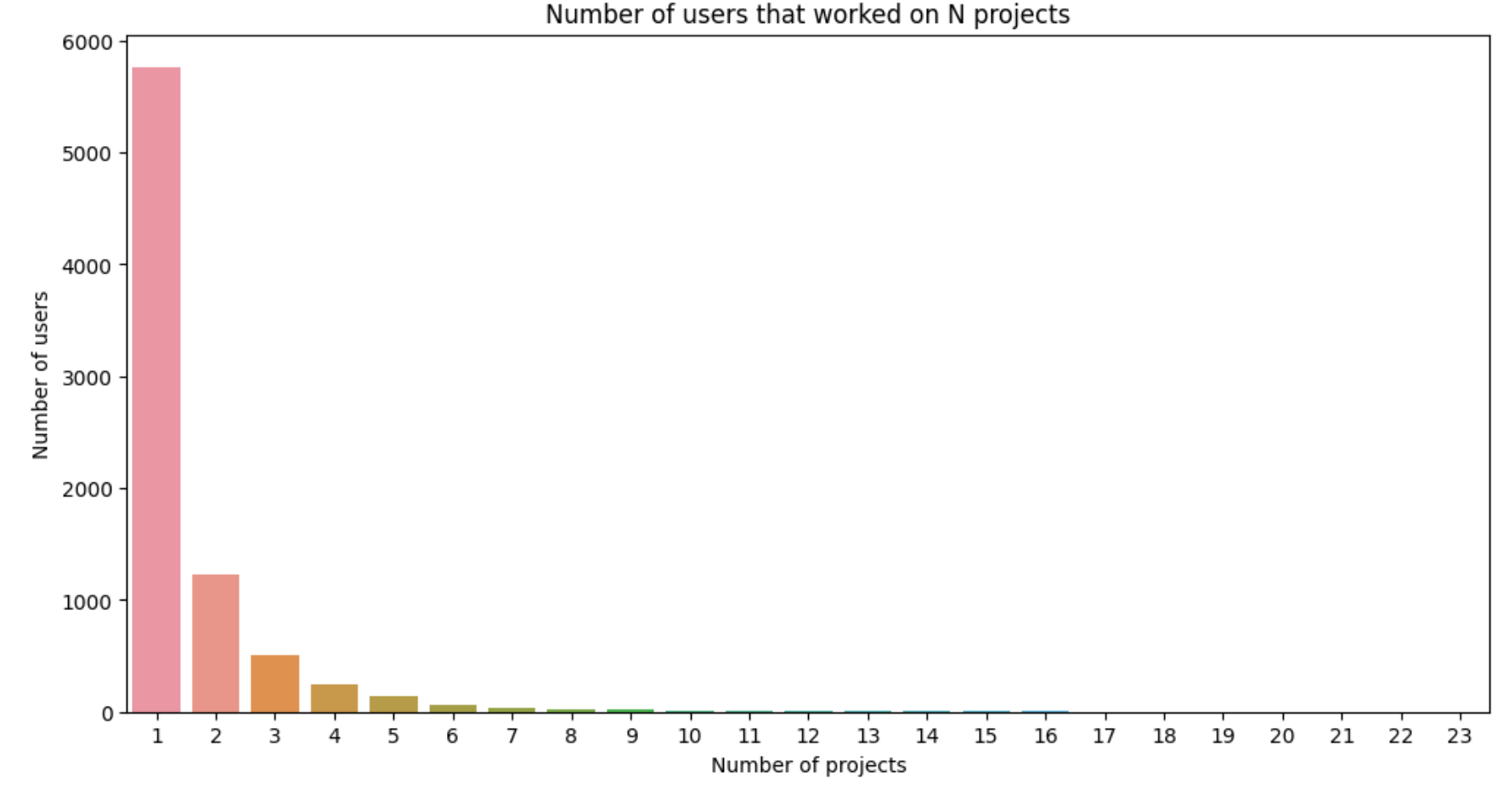
Unique projects by users

This will be an histogram that counts the unique projects each user had worked on during the given span of a week.

In [16]:
```python
projects_per_user = class_7d.groupby(['user_id'])['project_id'].nunique()
groupby_projects_per_user = projects_per_user.value_counts()
groupby_projects_per_user
```

Out[16]:
```
1     5766
2     1226
3      504
4      240
5      134
6       63
7       40
8       21
9       18
10       8
16       4
13       4
17       4
12       4
14       4
11       3
15       2
33       1
19       1
20       1
21       1
22       1
34       1
Name: project_id, dtype: int64
```

In [17]:
```python
title = 'Number of users that worked on N projects'
xlbl = 'Number of projects'
ylbl = 'Number of users'
sns_barplot_sorted(list(groupby_projects_per_user), title, xlbl, ylbl, xticks_steps=1,x=np.arange(1, len(groupb
```



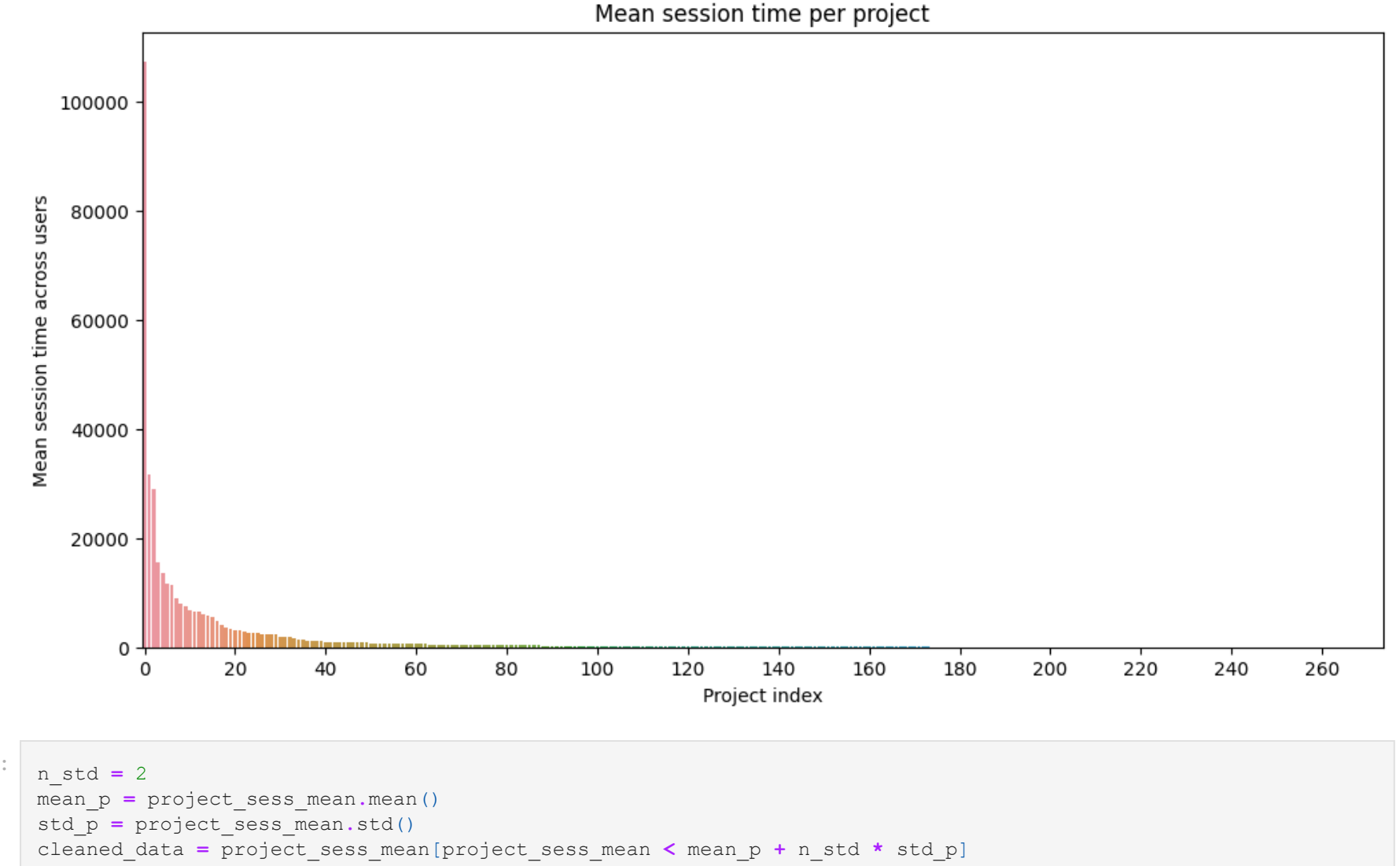
Session time exploration

Session time refers to the time it took a user to make the classification for that given event of project. session time varries alot and it might be because users leave there system running with the site open and then come back to classifiy.

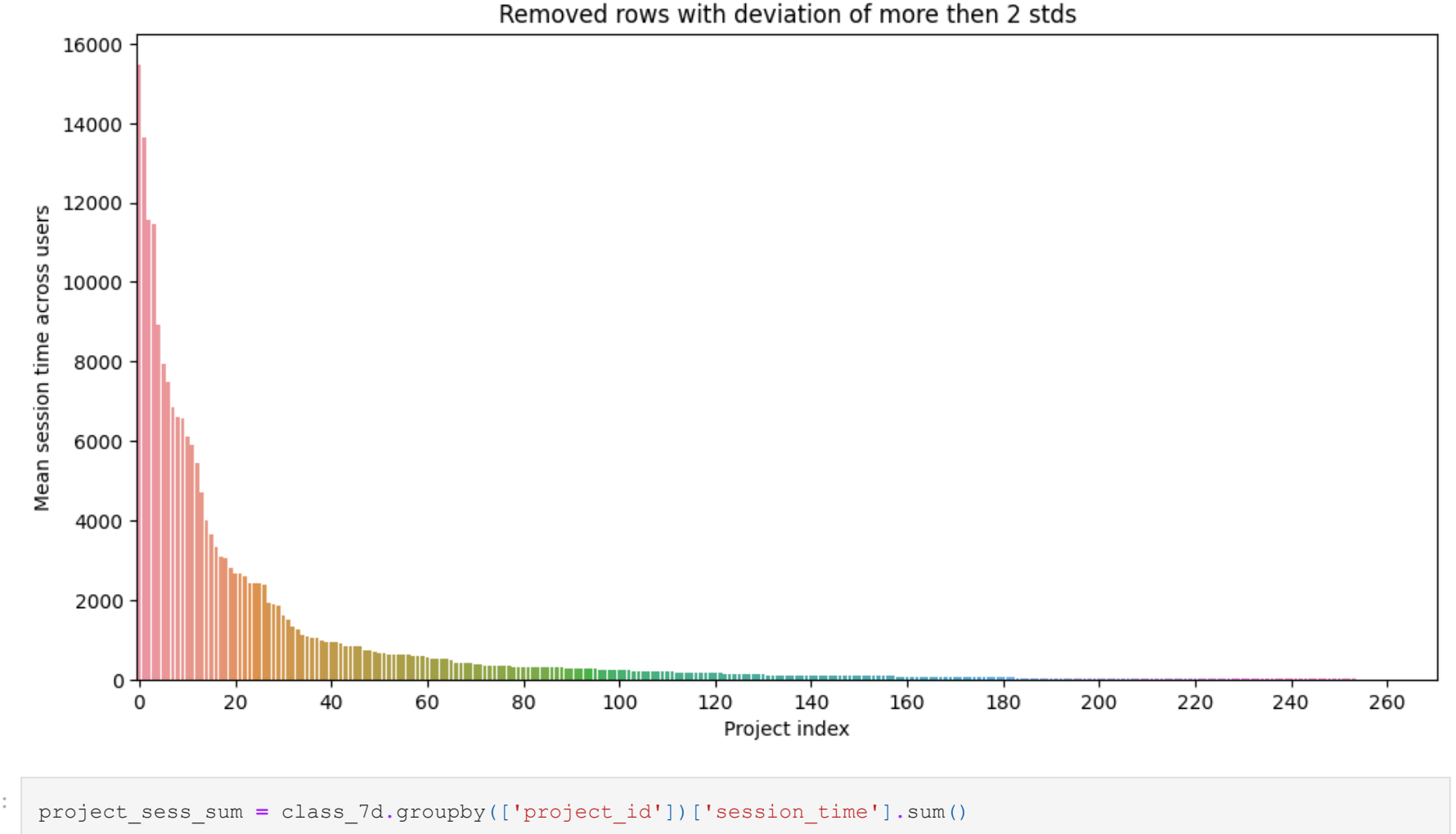
Session time per project

In [18]:
```python
project_sess_mean = class_7d.groupby(['project_id'])['session_time'].mean()
```

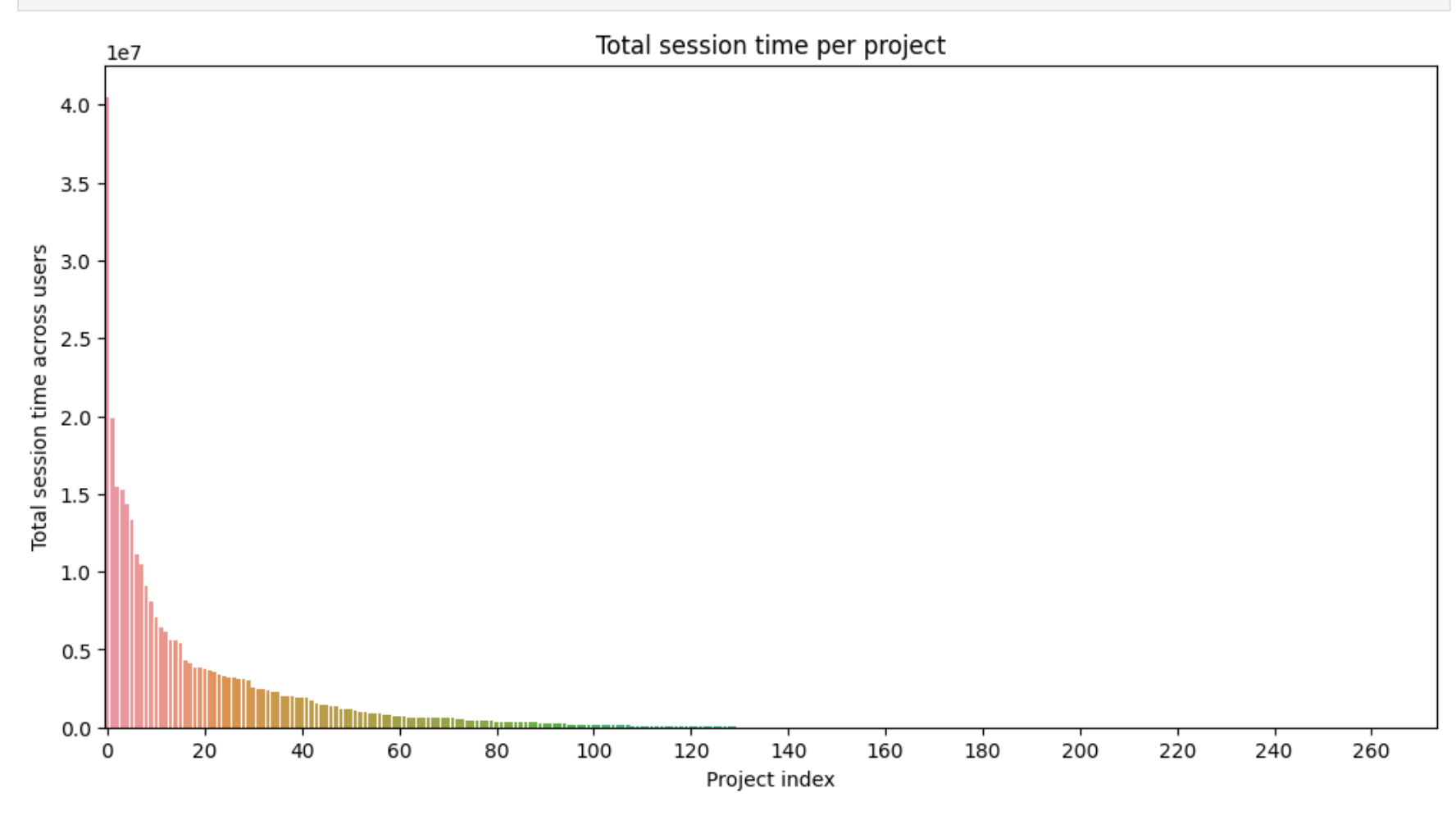
In [19]:
```python
title = 'Mean session time per project'
xlbl = 'Project index'
ylbl = 'Mean session time across users'
sns_barplot_sorted(project_sess_mean, title, xlbl, ylbl)
```



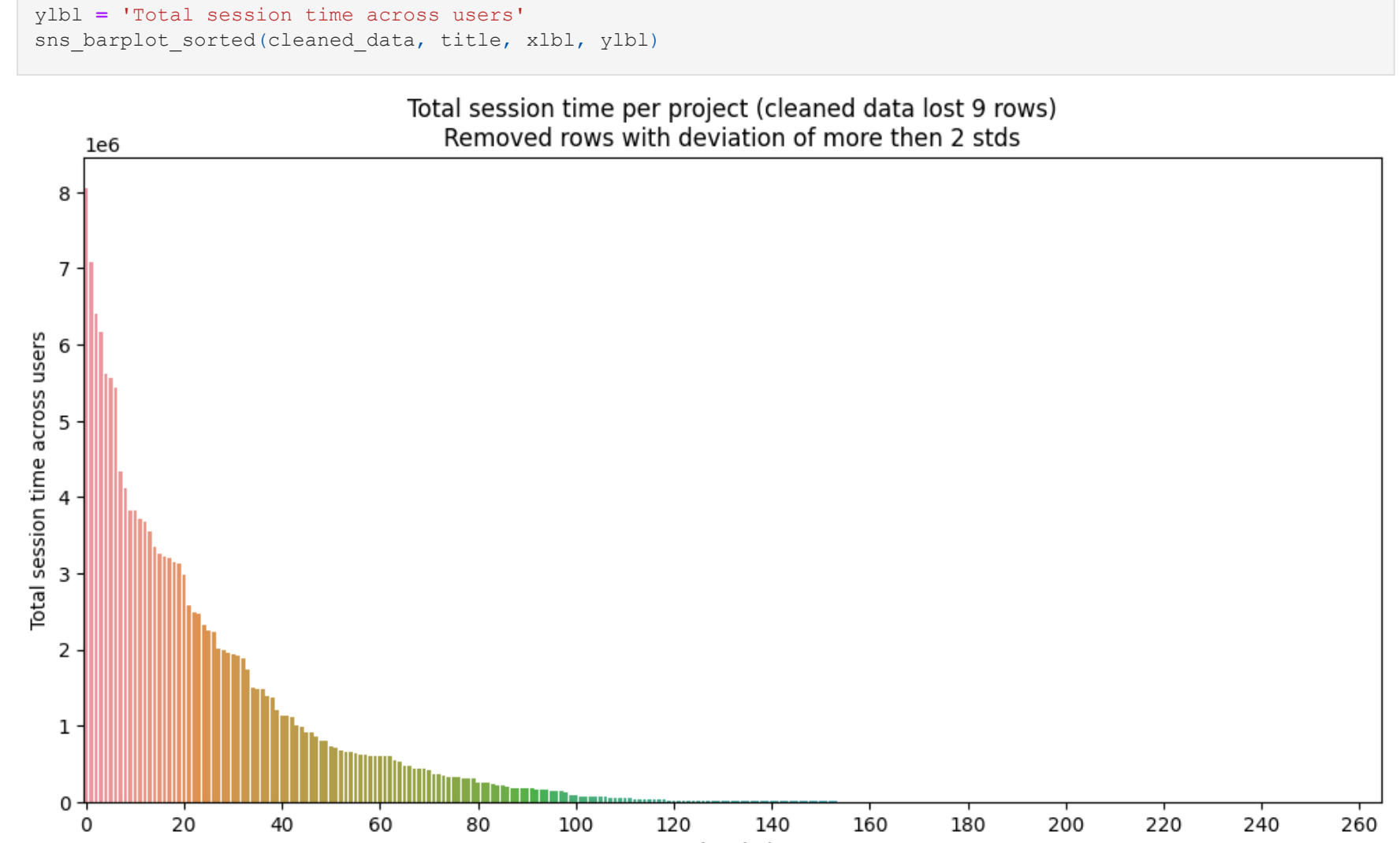
In [20]:
```python
n_std = 2
mean_p = project_sess_mean.mean()
std_p = project_sess_mean.std()
cleaned_data = project_sess_mean[project_sess_mean < mean_p + n_std * std_p]
cleaned_data = cleaned_data[cleaned_data > mean_p - n_std * std_p]
lost_rows = len(project_sess_mean) - len(cleaned_data)
title = f'Mean session time per project (cleaned data lost {lost_rows} rows)\nRemoved rows with deviation of mo
xlbl = 'Project index'
ylbl = 'Mean session time across users'
sns_barplot_sorted(cleaned_data, title, xlbl, ylbl)
```



In [21]:
```python
project_sess_sum = class_7d.groupby(['project_id'])['session_time'].sum()

title = 'Total session time per project'
xlbl = 'Project index'
ylbl = 'Total session time across users'
sns_barplot_sorted(project_sess_sum, title, xlbl, ylbl)
```



In [22]:
```python
n_std = 2
mean_p = project_sess_sum.mean()
std_p = project_sess_sum.std()
cleaned_data = project_sess_sum[project_sess_sum < mean_p + n_std * std_p]
cleaned_data = cleaned_data[cleaned_data > mean_p - n_std * std_p]
lost_rows = len(project_sess_mean) - len(cleaned_data)
title = f'Total session time per project (cleaned data lost {lost_rows} rows)\nRemoved rows with deviation of m
xlbl = 'Project index'
ylbl = 'Total session time across users'
sns_barplot_sorted(cleaned_data, title, xlbl, ylbl)
```



In [ ]: