

```
In [263]: import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore
from google.oauth2 import service_account
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

## Read as Pandas Dataframe

```
In [16]: read as pd dataframe
credentials = service_account.Credentials.from_service_account_file('./open-i
oped_credentials = credentials.with_scopes(['https://www.googleapis.com/autl
=firestore.Client(credentials=scoped_credentials)

problemSubmissionsSum21 = list(db.collection(u'problemSubmissionsSum21').stre
problemSubmissionsSum21_dict = list(map(lambda x: x.to_dict(), problemSubmissi
```

```
-----
--
NameError                                Traceback (most recent call las
t)
<ipython-input-16-f067ec948915> in <module>
      2
      3 problemSubmissionsSum21_dict = list(map(lambda x: x.to_dict(), pr
problemSubmissionsSum21))
----> 4 sum21_df = pd.DataFrame(users_dict)

NameError: name 'users_dict' is not defined
```

```
In [23]: sum21_df_raw = pd.DataFrame(problemSubmissionsSum21_dict)
sum21_df_raw.head()
```

Out[23]:

	hintsCorrect	eventType	canvasStudentID	hintsFinished	correctAnswer	timeStamp	problemID
0	None	answerStep	None	[0, 0, 0, 0]	[\$22]	06-10-2021 17:55:24	real1
1	None	answerStep	None	[0, 0, 0, 0, 0, 0, 0, 0]	[4]	06-10-2021 17:57:03	real1
2	None	answerStep	None	[0, 0, 0, 0, 0, 0, 0, 0]	[4]	06-10-2021 17:57:26	real1
3	None	unlockHint	None	[1, 0, 0, 0, 0, 0, 0, 0]	[4]	06-10-2021 17:57:28	real1
4	None	answerStep	Anabelle%20Garcia	[0, 0, 0, 0, 0]	x = 6	[ 06-11-2021 18:17:25	LinEqua

```
In [24]: sum21_df_raw.to_csv("./data/submission_sum_21.csv")
```

## Number of users

```
In [45]: def get_num_users(df):  
         actual_users = df[(df["canvasStudentID"].notnull()) & (df["canvasStuden  
         return len(actual_users.groupby("canvasStudentID").size())
```

```
In [46]: # number of users during summer 21  
         get_num_users(sum21_df_raw)
```

```
Out[46]: 36
```

## Number of Actions Per Session

```
In [225]: # Config  
         BREAK_LEN = 5 # in min
```

```

In [268]: def count_sessions(df):
    """
    df: df of a specific user, sorted by time
    returns a list of number of actions per session
    """
    time_diff = df['time'].diff().dt.seconds.div(60, fill_value=float("inf"))
    new_session_start_index = df[time_diff > BREAK_LEN].index.tolist()

    actions_per_session = []
    for i in range(len(new_session_start_index) - 1):
        actions_per_session.append(len(df.loc[new_session_start_index[i]: new_session_start_index[i+1]]))
    actions_per_session.append(len(df.loc[new_session_start_index[-1] : ]))

    return actions_per_session

def get_actions_per_session(df):
    """
    df: raw df retrieved from firestore database
    """

    df = df.copy()
    df = df[(df["canvasStudentID"].notnull()) & (df["canvasStudentID"] != "")]
    df["time"] = pd.to_datetime(df['timeStamp'])
    df = df.sort_values(['time'], ascending=True)

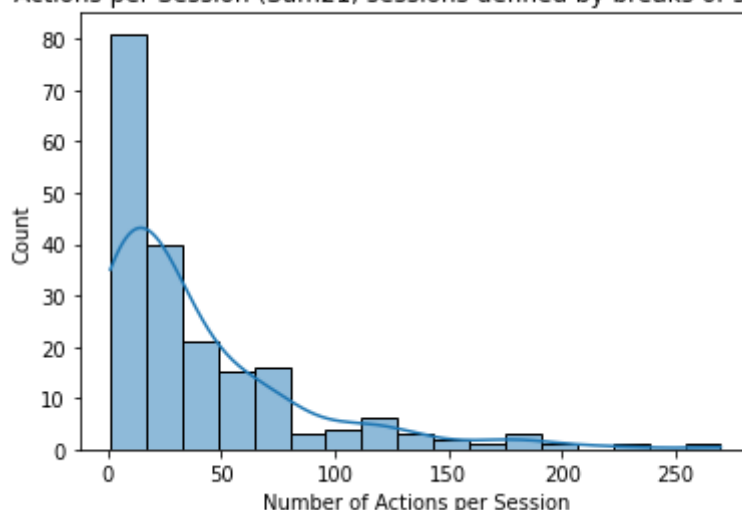
    actions_per_session_series = df.groupby("canvasStudentID").apply(count_sessions)
    actions_per_session = sum(actions_per_session_series.tolist(), [])

    sns.histplot(actions_per_session, kde=True)
    plt.title(f"Actions per Session (Sum21, sessions defined by breaks of {BREAK_LEN} mins)")
    plt.xlabel("Number of Actions per Session")
    plt.ylabel("Count");

get_actions_per_session(sum21_df_raw)

```

Actions per Session (Sum21, sessions defined by breaks of 5mins)

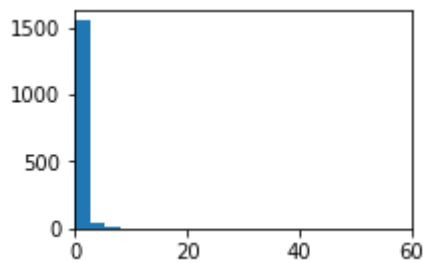


## Helper graphs for reference

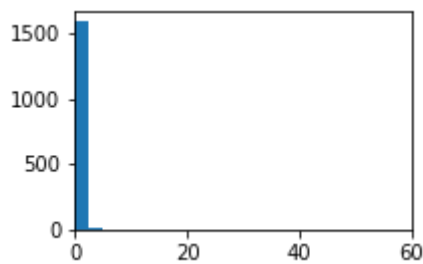
**To determine cutoff of break length between session**

```
In [115]: most_used_users = sum21_df.groupby(["canvasStudentID"]).size().sort_values(
for user in most_used_users:
    user_df = sum21_df[sum21_df["canvasStudentID"] == user].sort_values(["t
    time_diff = user_df['time'].diff().dt.seconds.div(60, fill_value=0)
    plt.figure()
    plt.figure(figsize=(3,2))
    plt.hist(time_diff, bins=500)
    plt.xlim(0, 60);
```

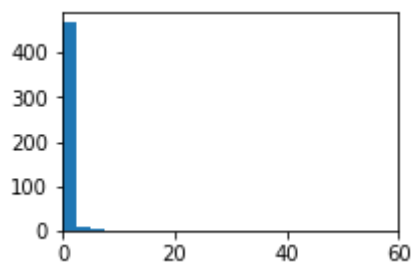
<Figure size 432x288 with 0 Axes>



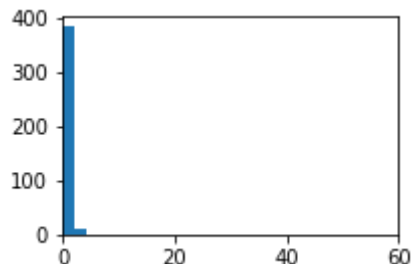
<Figure size 432x288 with 0 Axes>



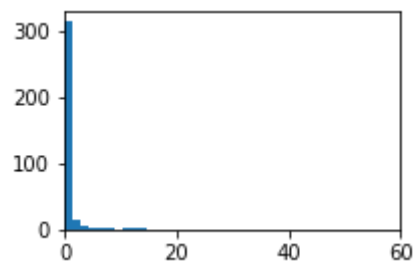
<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>



**To verify "large" action counts**

```
In [214]: s2 = sum21_df.sort_values(["time"], ascending=True)
s2 = s2[s2["canvasStudentID"] == "Rishi%20Patwardhan"]
time_diff = s2['time'].diff().dt.seconds.div(60, fill_value=float("inf"))
new_session_start_index = s2[time_diff > BREAK_LEN].index.tolist()
print(new_session_start_index)
print(s2.index.tolist().index(4789), s2.index.tolist().index(5008))
s2.iloc[200: 210]
# s2
```

```
[5369, 5391, 5398, 4208, 4752, 4789, 5008, 5022, 5069, 196, 399, 983, 98
9, 1002, 1006, 1033, 1050, 1075, 1161, 1169, 1188, 1239, 1262, 1270, 136
2, 1474, 1639, 1731, 1735, 1743, 1761, 1818, 1993, 2087]
93 286
```

Out[214]:

	hintsCorrect	eventType	canvasStudentID	hintsFinished	correctAnswer	timeStamp	
5089	True	hintScaffoldLog	Rishi%20Patwardhan	[1, 1, 0.5, 0, 0, 0]	None	6-10-2021 20:4:25	F
5090	None	unlockHint	Rishi%20Patwardhan	[1, 1, 1, 0.5, 0, 0]	[(3,-3/2)]	6-10-2021 20:4:26	F
5091	None	unlockHint	Rishi%20Patwardhan	[1, 1, 1, 1, 0.5, 0]	[(3,-3/2)]	6-10-2021 20:4:30	F
5094	True	hintScaffoldLog	Rishi%20Patwardhan	[1, 1, 1, 1, 0.5, 0]	None	6-10-2021 20:4:35	F
5095	None	unlockHint	Rishi%20Patwardhan	[1, 1, 1, 1, 1, 1]	[(3,-3/2)]	6-10-2021 20:4:36	F
5096	None	answerStep	Rishi%20Patwardhan	[1, 1, 1, 1, 1, 1]	[(3,-3/2)]	6-10-2021 20:4:49	F
5104	None	unlockHint	Rishi%20Patwardhan	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[6]	6-10-2021 20:5:05	F
5114	None	answerStep	Rishi%20Patwardhan	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[6]	6-10-2021 20:5:29	F
5116	None	unlockHint	Rishi%20Patwardhan	[1, 0, 0, 0, 0, 0, 0, 0]	[(2,-4)]	6-10-2021 20:5:33	F
5117	None	unlockHint	Rishi%20Patwardhan	[1, 1, 0, 0, 0, 0, 0, 0]	[(2,-4)]	6-10-2021 20:5:34	F

## Length of Each Session

```
In [248]: # Config  
BREAK_LEN = 5 # in min
```



```

In [266]: def session_lengths_for_one_user(df):
    '''
    df: df of a specific user, sorted by time
    returns a list of number of actions per session
    '''
    time_diff = df['time'].diff().dt.seconds.div(60, fill_value=float("inf"))
    new_session_start_index = df[time_diff > BREAK_LEN].index.tolist()

    session_lengths = []
    for i in range(1, len(new_session_start_index)):
        time_elapsed = (df.iloc[df.index.tolist().index(new_session_start_index[i]) - 1]['time'] -
                        df.loc[new_session_start_index[i - 1]]['time']).seconds
        session_lengths.append(time_elapsed)
    session_lengths.append((df.iloc[-1]['time'] - \
                            df.loc[new_session_start_index[-1]]['time']).seconds)

    return session_lengths

def get_session_lengths(df):
    '''
    df: raw df retrieved from firestore database
    '''

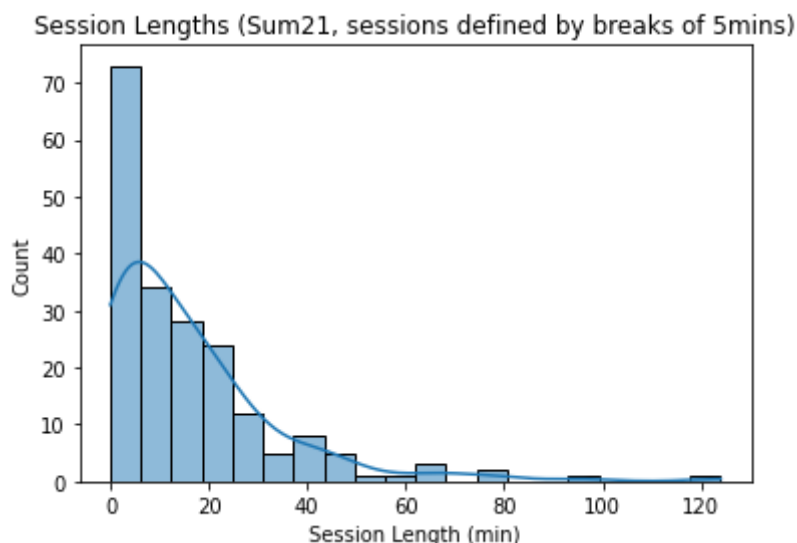
    df = df.copy()
    df = df[(df["canvasStudentID"].notnull()) & (df["canvasStudentID"] != "")]
    df["time"] = pd.to_datetime(df['timeStamp'])
    df = df.sort_values(['time'], ascending=True)

    session_lengths_series = df.groupby("canvasStudentID").apply(session_lengths_for_one_user)
    session_lengths = sum(session_lengths_series.tolist(), [])

    sns.histplot(session_lengths, kde=True)
    plt.title(f"Session Lengths (Sum21, sessions defined by breaks of {BREAK_LEN} mins)")
    plt.xlabel("Session Length (min)")
    plt.ylabel("Count");

get_session_lengths(sum21_df_raw)

```



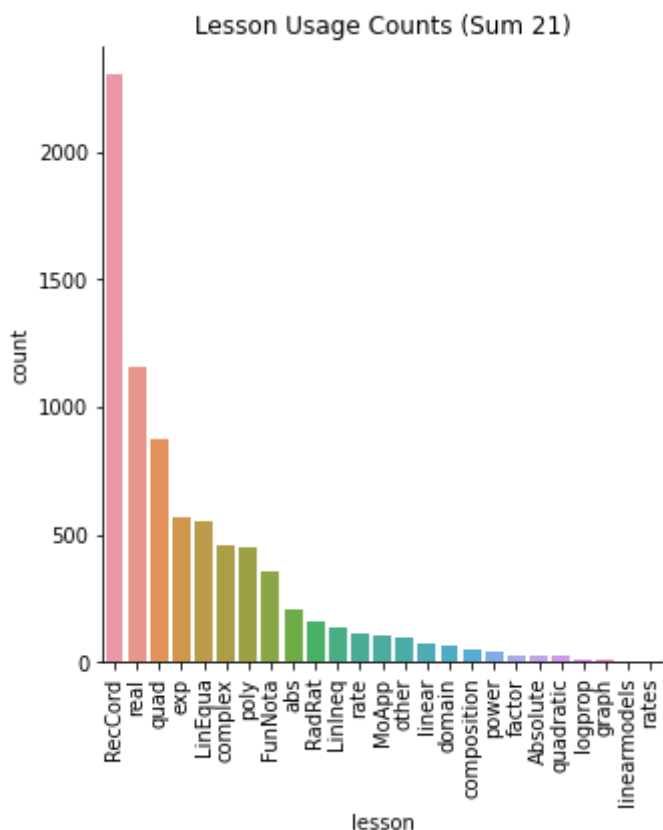
# Material Usage

```
In [307]: def get_lessons(df):
df = df.copy()
df = df[(df["canvasStudentID"].notnull()) & (df["canvasStudentID"] != "")]
df["time"] = pd.to_datetime(df['timeStamp'])
df = df.sort_values(['time'], ascending=True)
df["lesson"] = df["problemID"].str.findall("(\\D+)\\d{1,2}").str[0]
lesson_size = df.groupby("lesson").size().sort_values(ascending=False)
lesson_size = lesson_size.reset_index()
return lesson_size

lesson_size = get_lessons(sum21_df_raw)
print("Lessons used: (sorted by usage)")
print(lesson_size["lesson"].tolist())
# lesson_size
sns.catplot(x="lesson", y="count", kind="bar", data=lesson_size)
plt.title("Lesson Usage Counts (Sum 21)")
plt.xticks(rotation=90);
```

Lessons used: (sorted by usage)

['RecCord', 'real', 'quad', 'exp', 'LinEqua', 'complex', 'poly', 'FunNot a', 'abs', 'RadRat', 'LinIneq', 'rate', 'MoApp', 'other', 'linear', 'domain', 'composition', 'power', 'factor', 'Absolute', 'quadratic', 'logprop', 'graph', 'linearmodels', 'rates']



In [ ]:

In [ ]:

