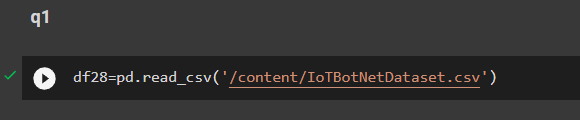
**Eval\_Lab 19CSE304**

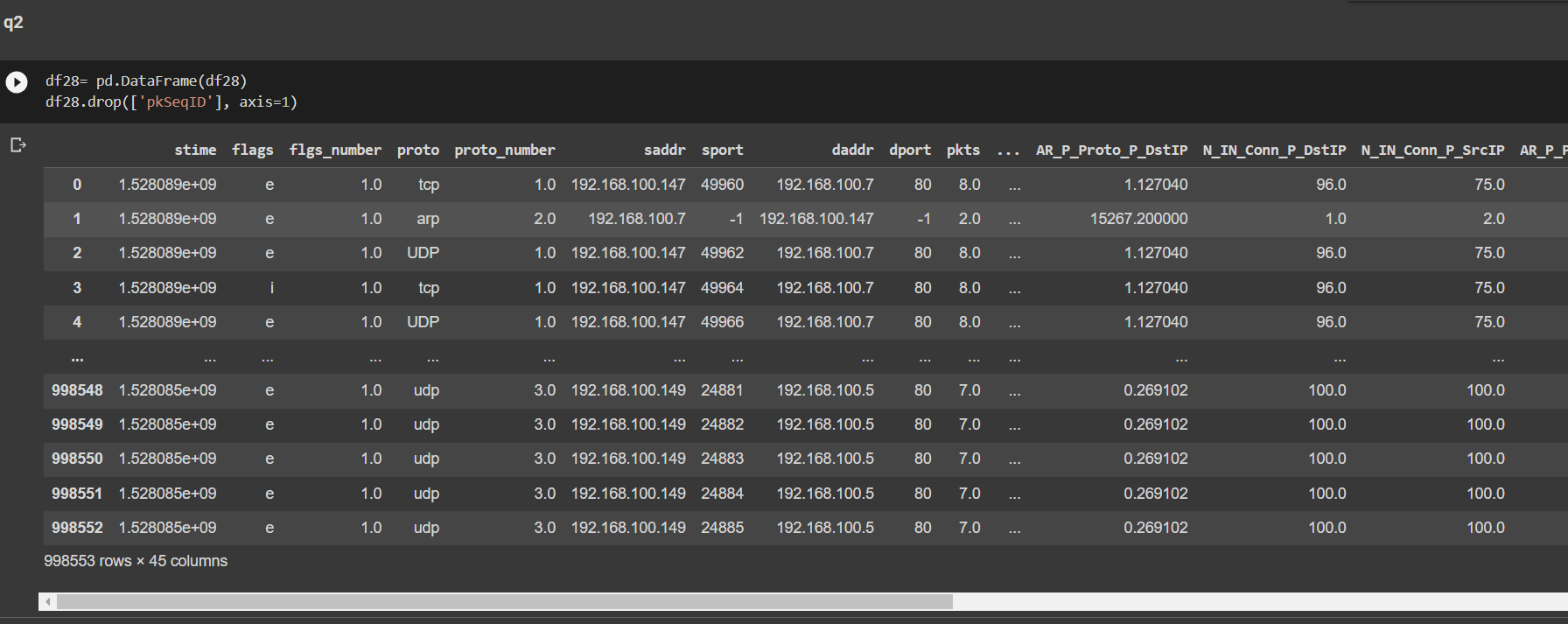
1. Import data and save it in your disk space



1. Drop the columns not required for your analysis.

Code: df28= pd.DataFrame(df28)

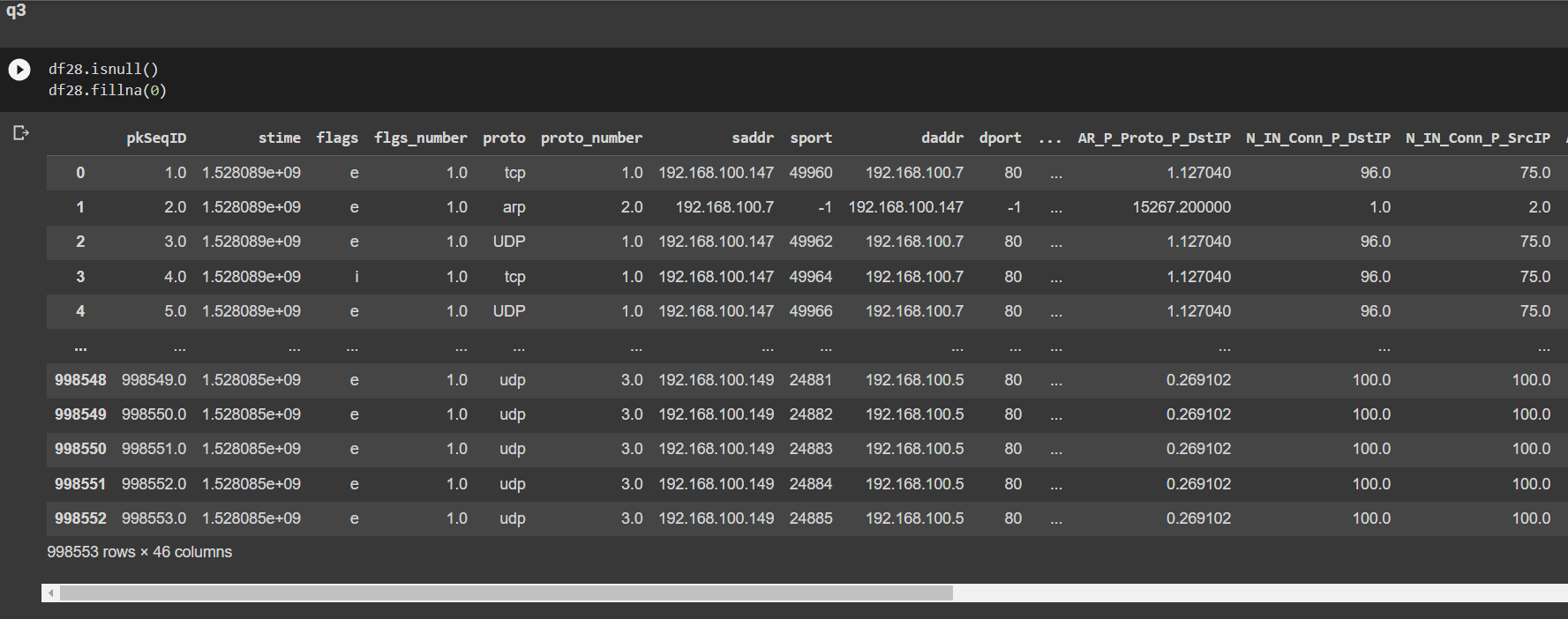
df28.drop(['pkSeqID'], axis=1)



1. Do a missing value analysis and perform appropriate steps to correct them

Code: df28.isnull()

df28.fillna(0)



1. Perform an outlier analysis for any two columns

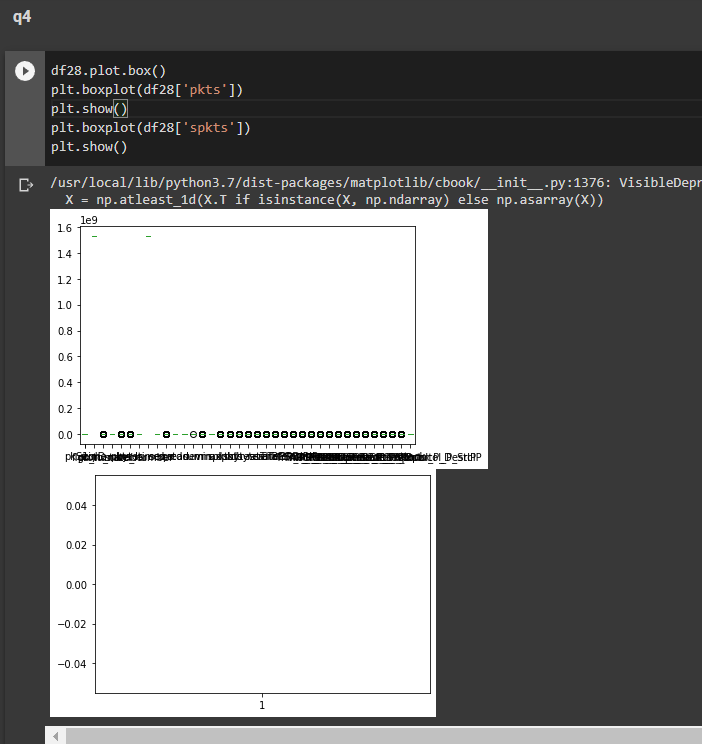
Code: df28.plot.box()

plt.boxplot(df28['pkts'])

plt.show()

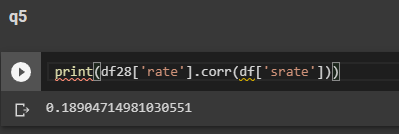
plt.boxplot(df28['spkts'])

plt.show()



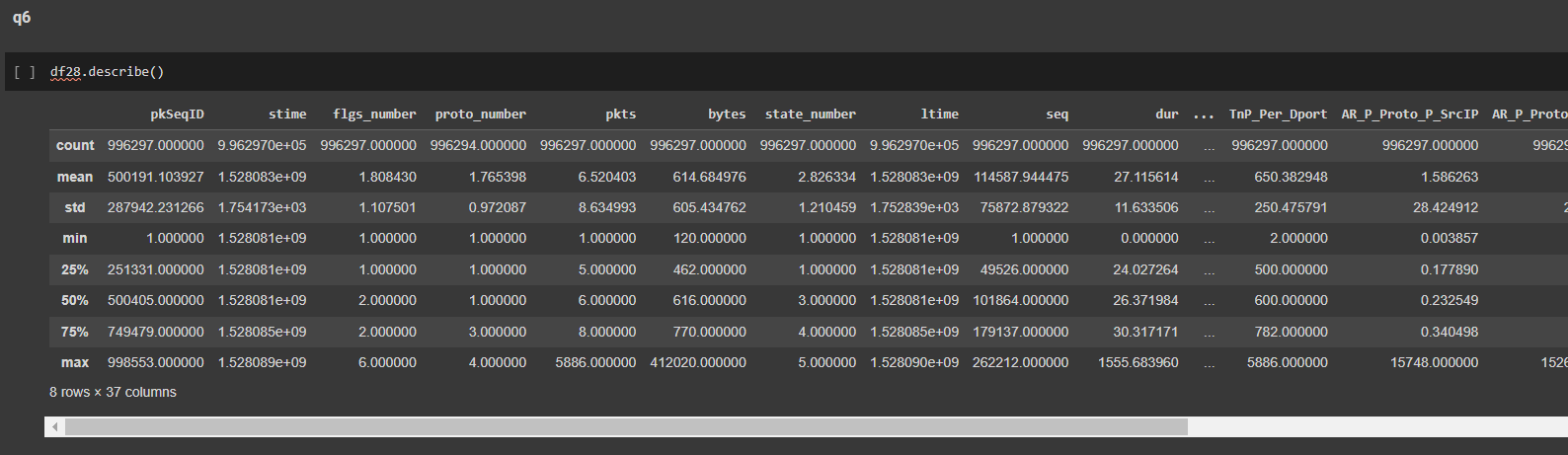
1. Do a correlation analysis of any two variables.

Code:  print(df28['rate'].corr(df['srate']))



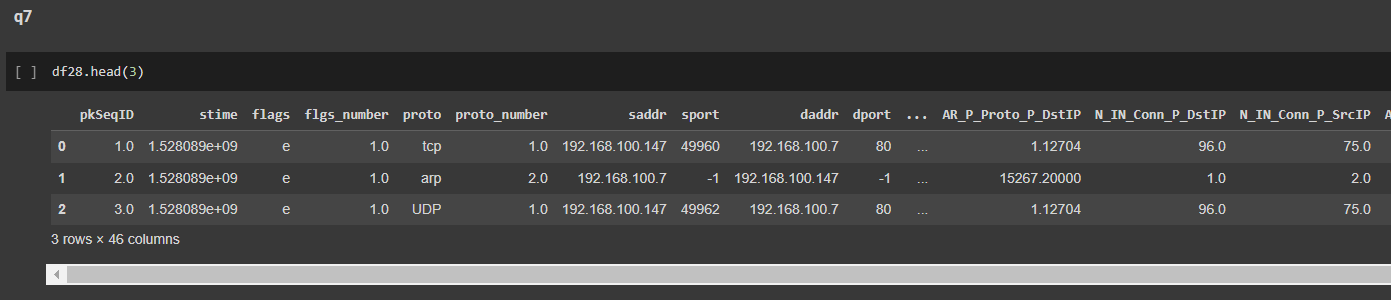
1. Describe the data

Code: df28.describe()



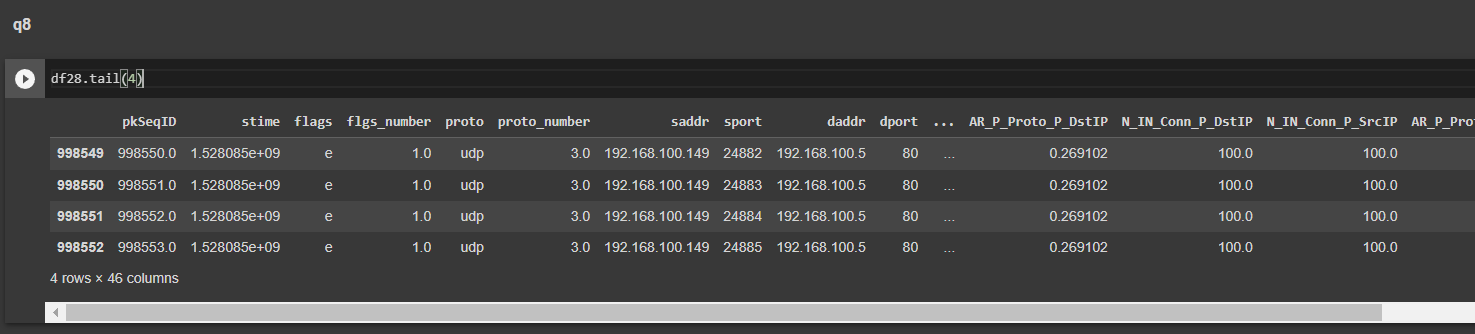
1. Display the first 3 rows

Code: df28.head(3)



1. Display the last 4 rows

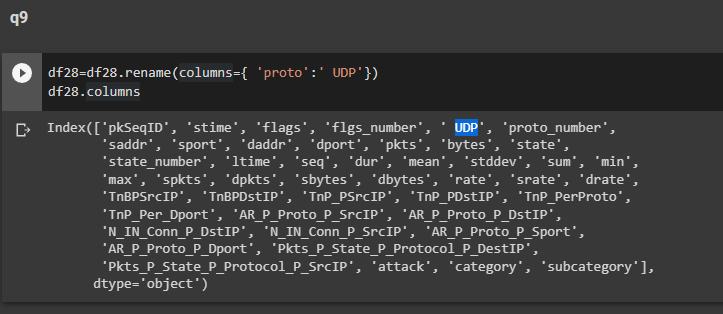
Code: df28.tail(4)



1. Locate the first row corresponding to the proto as “UDP”

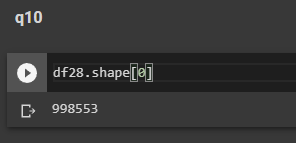
Code: df28=df28.rename(columns={ 'proto':' UDP'})

df28.columns



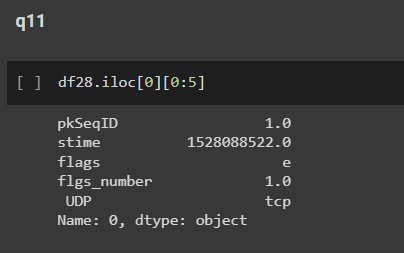
1. How many rows of data do you have?

Code: df28.shape[0]



1. Display the first 5 columns of the first row

Code: df28.iloc[0][0:5]



1. Count the number of packets affected based on flags as “i”

Code: df28['flags'].value\_counts()

Graphical user interface, application

Description automatically generated

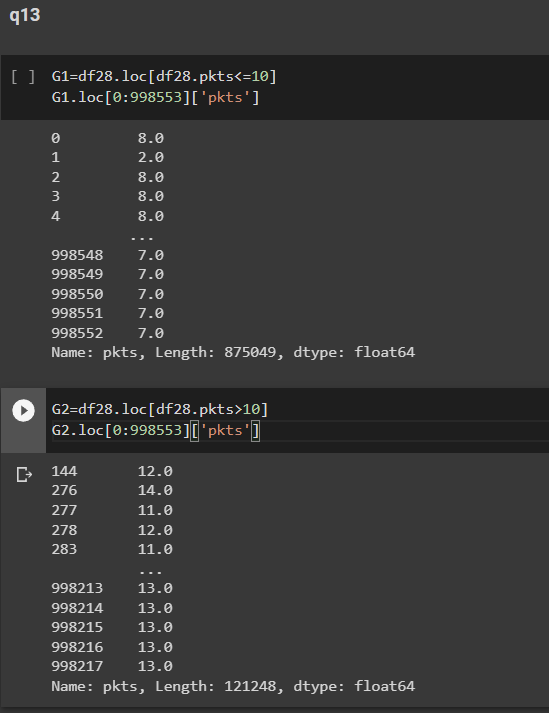
1. Split the traffic based on traffic into two groups Group 1: pkts<=10 and Group 2:pkts>10

Code: G1=df28.loc[df28.pkts<=10]

G1.loc[0:998553]['pkts']

G2=df28.loc[df28.pkts>10]

G2.loc[0:998553]['pkts']



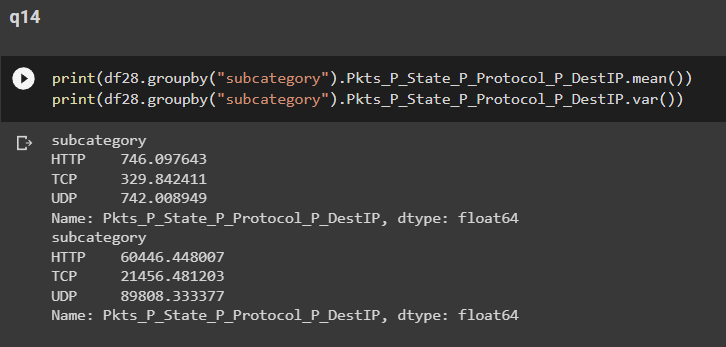
1. Compute the mean and the variance of " Pkts\_PState\_P\_Protocol\_P\_DestIP” for subcategory as HTTP and TCP

Code:print(df28.groupby("subcategory").Pkts\_P\_State\_P\_Protocol\_P\_DestIP.

mean())

print(df28.groupby("subcategory").Pkts\_P\_State\_P\_Protocol\_P\_DestI

P.var())



1. Draw a histogram for packets with stime as 1528088521, 1528088522., 1528088523. Histogram should be step-filled with bin of size 20. Find the skew of this distribution, and comment if it is positive or negative

Code: import seaborn as sns

sns.boxplot(df28['stime'])

Graphical user interface

Description automatically generated with low confidence