### Ime, priimek:

LV04: Psihofiziološki signali

# Uvod

Namen vaje:

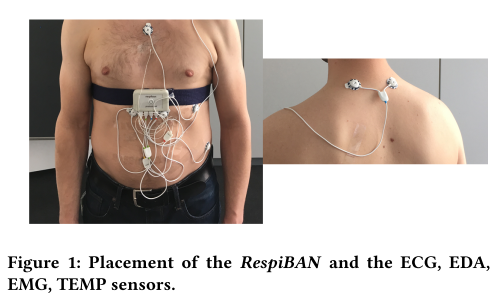
* spoznati psihofiziološke signale, merjenje in njihove značilnosti
* določiti značilke signalov, potrebne za strojno učenje
* izdelati model strojnega učenja, ki napoveduje nivo stresa na podlagi psihofizioloških signalov.

Problem se povezuje s področjem čustvenega računalništva (angl. *Affective computing*), ki vključuje področja računalništva, kognitivne znanosti in psihologije. Omogoča, da naprave na podlagi fizioloških, vizualnih, glasovnih in drugih podatkov določajo čustveno stanje človeka. V ta namen se uporablja strojno učenje (angl. *Machine learning*)ali globoko učenje (angl. *Deep learning*) in velike količine podatkov, na katerih se modeli strojnega učenja trenirajo.

## Opis problema in baza WESAD

Znano je, da ima stres slab vpliv na naše zdravje in počutje. Zato se je pojavilo zanimanje za detekcijo stresa preko psiho-fizioloških meritev. Podatkovna zbirka WESAD vključuje psiho-fiziološke meritve dveh naprav na 15 uporabnikih, ki so v teku poskusa šli skozi različna emocionalna stanja. Zapestna in prsna naprava sta merili lastnosti kot so temperatura telesa, utrip, elektrodermalno aktivnost, dihanje, triosni pospešek. Prav tako je v podatkovno zbirko vključeno lastno poročilo vsakega uporabnika glede njegovega počutja.

Na podlagi zbranih podatkov se je z uporabo metod strojnega učenja določalo počutje uporabnika. Počutje je bilo klasificirano v stanja: osnovno, pod stresom, zabava, meditacija.



# Psihofiziološki signali in njihove značilnosti

## WESAD podatki

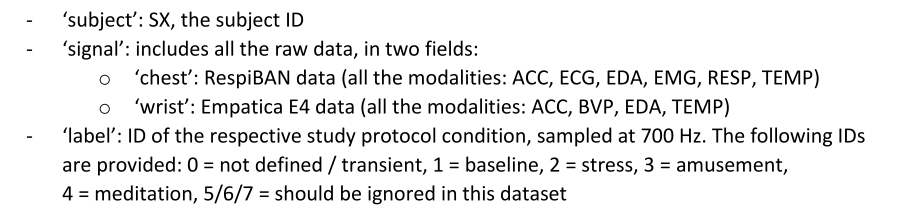
To so sinhronizirani podatki iz dveh naprav: RespiBAN (naprava meri na prsih), ter Empatica E4 (zapestje).

Podatki so v obliki slovarja, imamo 3 komponente

\* subject : ID osebe

\* signal : vsebuje nov slovar s ključi : chest, wrist za obe napravi. Vrednost obeh je zopet slovar, kjer so ključi oznake senzorjev (ACC, ECG, EDA, EMG, RESP, TEMP za chest).

\* label : oznaka stanja osebe (po protokolu testiranja): 0 nedefinirano, 1 osnovno (baseline), 2 stres, 3 zabava (amusement), 4 meditacija, 5,6,7 ni uporabljeno



### Nalaganje

P1 : Nalaganje podatkov v s2\_data

import pickle

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

pickle.HIGHEST\_PROTOCOL

# Nastavi pot do podatkov

data\_set = r'W:\WESAD\\'

s2\_path = data\_set + 'S2/S2.pkl'

# P.1 : Nalaganje podatkov ene osebe

with open(s2\_path, 'rb') as file:

    s2\_data = pickle.load(file, encoding='latin1')

### P2: Struktura podatkov

Preglej podatke, kaj vsebujejo.

# P.2 : Struktura podatkov ene osebe

# Izberi podatke osebe kot data

data = s2\_data

# Izpiši podatke

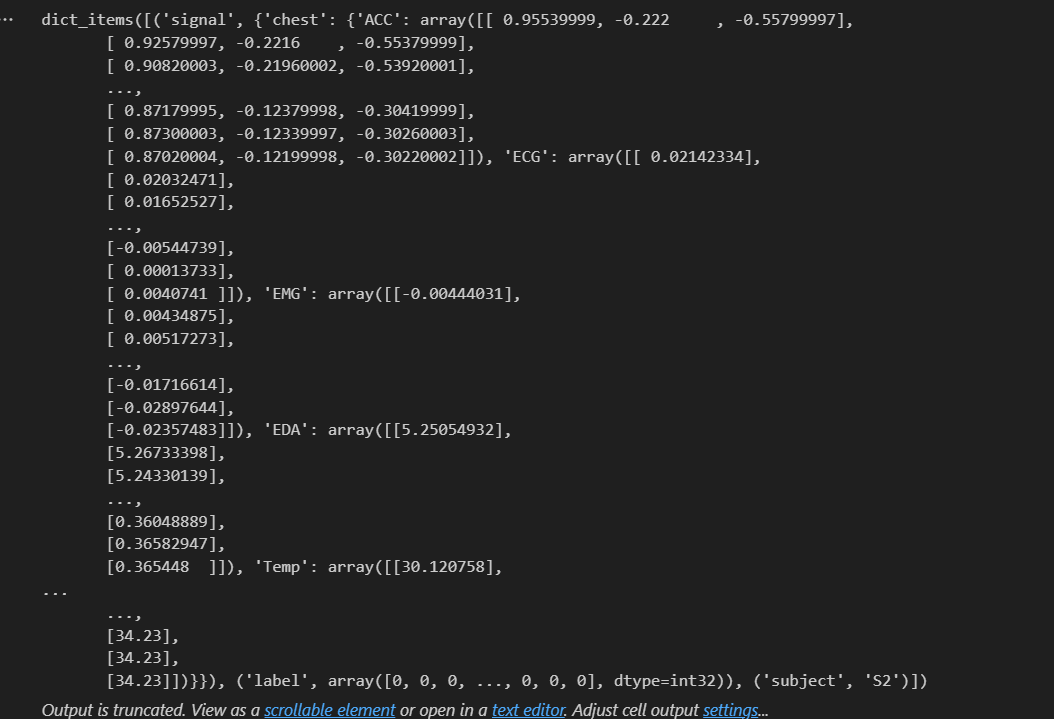
#print(data.keys())

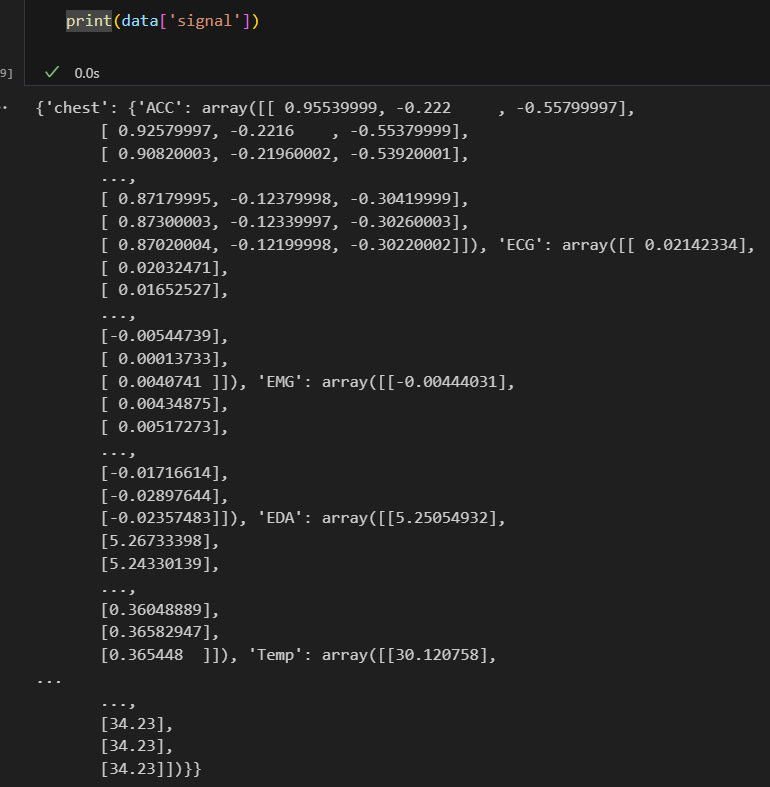
#print(data.items())

#print(data['signal'])

Vstavi izpis signalov:







### P.3 Labele: stanja osebe tekom poskusa

# P.3: Labele: stanja osebe

count=len(data['label'])

print('Stevilo vzorcev  = ', count)

sample\_n = np.arange(1,count+1)

print(sample\_n)

# Komentar

d\_label = pd.DataFrame(data['label'], columns=['label'])

d\_label['sec']=d\_label['label']

d\_label['sec']=sample\_n/700

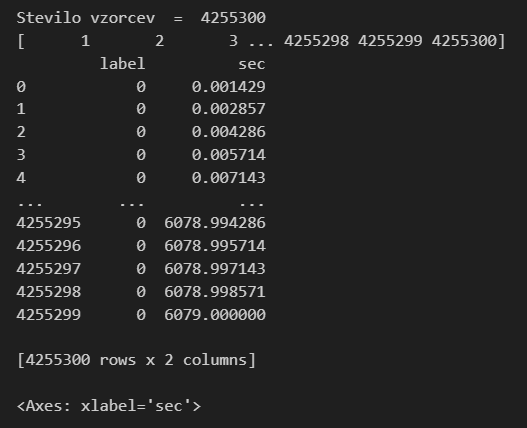
print(d\_label)

# Komentar

d\_label['label'].hist()

d\_label.plot(x='sec', y='label')

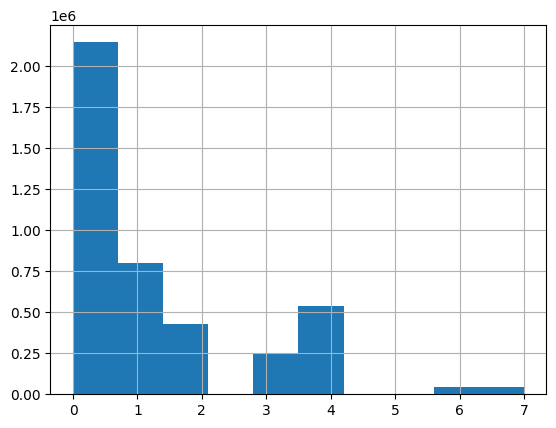
Katere vrednosti label so pomembne glede stanja osebe?

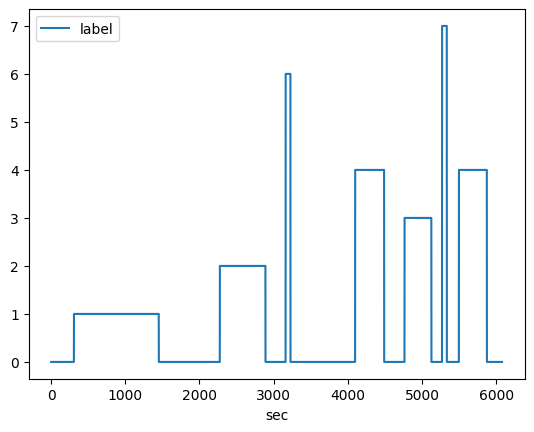


Vstavi in komentiraj grafe, kaj prikazujejo.

Histograme podatkov.

Koliko časa je trajalo določeno stanje





### P.4 : Izloči podatke po senzorjih za 1 osebo

c\_ax=data['signal']['chest']['ACC'][0:,0]

c\_ay=data['signal']['chest']['ACC'][0:,1]

c\_az=data['signal']['chest']['ACC'][0:,2]

c\_ecg=data['signal']['chest']['ECG'][:,0]

c\_emg=data['signal']['chest']['EMG'][:,0]

c\_eda=data['signal']['chest']['EDA'][:,0]

c\_temp=data['signal']['chest']['Temp'][:,0]

c\_resp=data['signal']['chest']['Resp'][:,0]

w\_ax=data['signal']['wrist']['ACC'][0:,0]

w\_ay=data['signal']['wrist']['ACC'][0:,1]

w\_az=data['signal']['wrist']['ACC'][0:,2]

w\_bvp=data['signal']['wrist']['BVP'][:,0]

w\_eda=data['signal']['wrist']['EDA'][:,0]

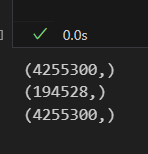
w\_temp=data['signal']['wrist']['TEMP'][:,0]

c\_label=data['label']

print(c\_ax.shape)

print(w\_ax.shape)

print(c\_label.shape)



### P.5 : Določi indekse, ki povejo podatke po stanjih

baseline\_indices = np.nonzero(data['label']==1)[0]

stress\_indices = np.nonzero(data['label']==2)[0]

amusement\_indices = np.nonzero(data['label']==3)[0]

meditation\_indices = np.nonzero(data['label']==4)[0]

print("baseline\_indices", len(baseline\_indices))

print("stress\_indices", len(stress\_indices))

print("amusement\_indices", len(amusement\_indices))

print("meditation\_indices", len(meditation\_indices))

fig = plt.figure()

ax = fig.add\_axes([0.1,0.1,1,1])

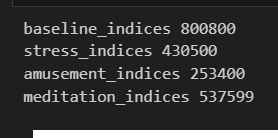
langs = ["baseline", "stress", "amusement","meditation"]

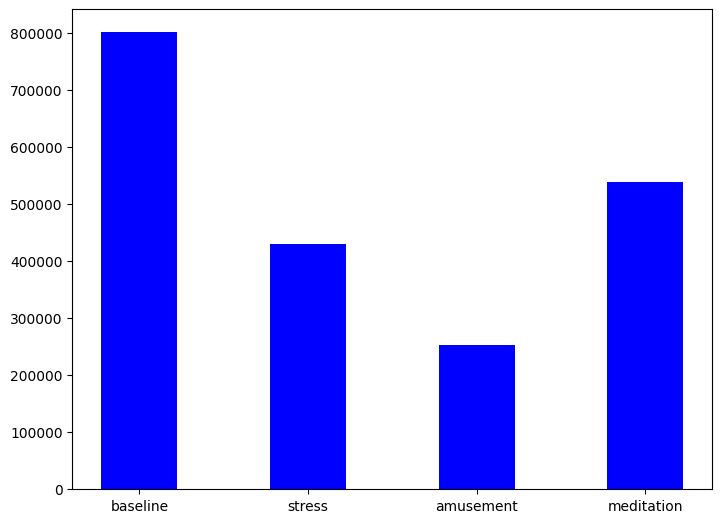
y\_ax = [len(baseline\_indices),len(stress\_indices),len(amusement\_indices),len(meditation\_indices)]

ax.bar(langs,y\_ax,color = 'b', width = 0.45)

fig.savefig('f1.png')

plt.show()





### P.6 Naredimo dataframe vseh signalov

# P.6 : Naredimo DataFrame z vsemi signali (Chest)

numpy\_data1=np.array([sample\_n/700, c\_ax, c\_ay, c\_az,c\_ecg,c\_emg,c\_eda,c\_temp,c\_resp,c\_label])

numpy\_data1=numpy\_data1.T

#df = pd.DataFrame(data=numpy\_data1, columns=["c\_ax", "c\_ay", "c\_az","c\_ecg","c\_emg","c\_eda","c\_temp","c\_resp","w\_ax","w\_ay","w\_az","w\_bvp","w\_eda","w\_temp"],orient='index')

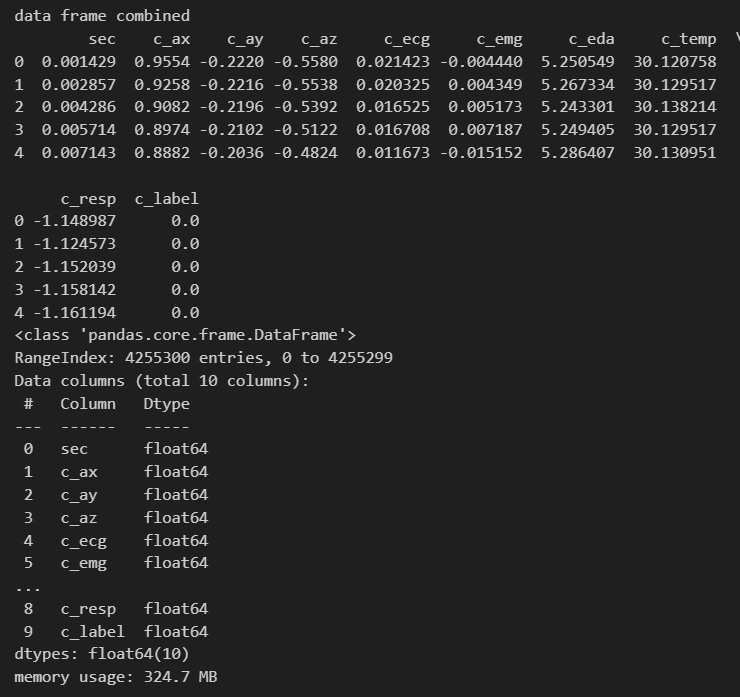
# Chest signali

dfc = pd.DataFrame(data=numpy\_data1, columns=["sec","c\_ax", "c\_ay", "c\_az","c\_ecg","c\_emg","c\_eda","c\_temp","c\_resp","c\_label"])

print("data frame combined")

print(dfc.head())

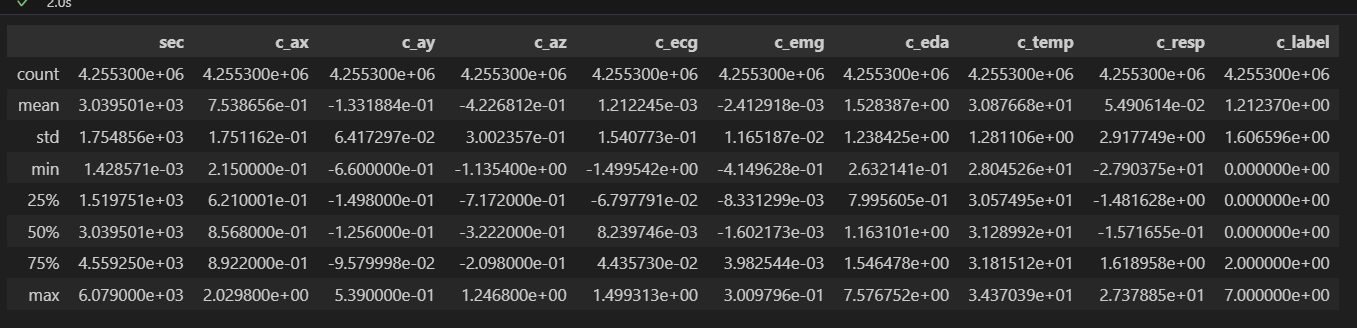
dfc.info()



### Statistični opis signalov

# P.7 Statistika vrednosti

dfc.describe()

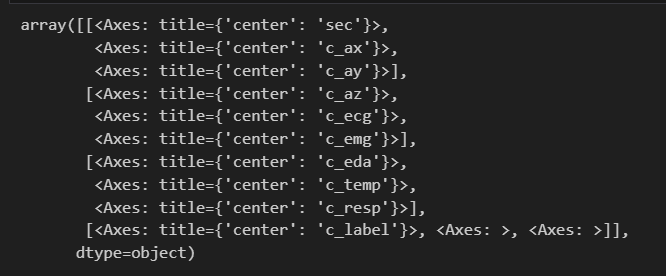


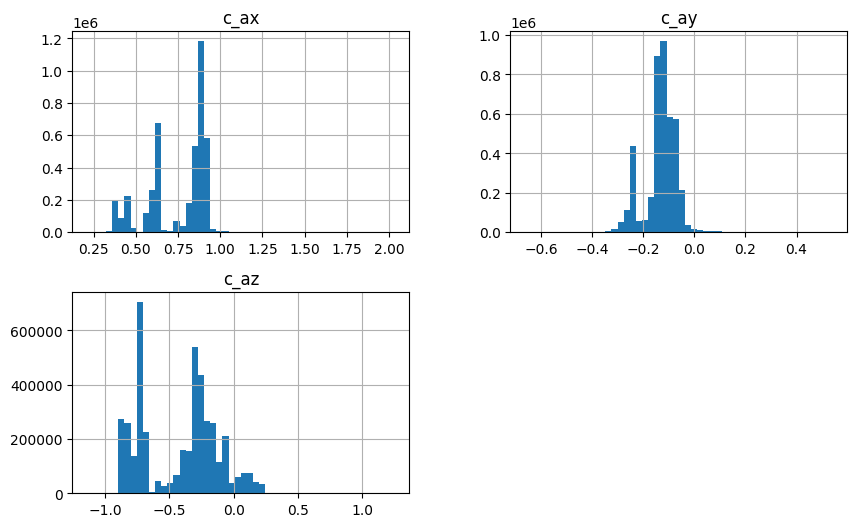
### Histogrami vrednosti signalov

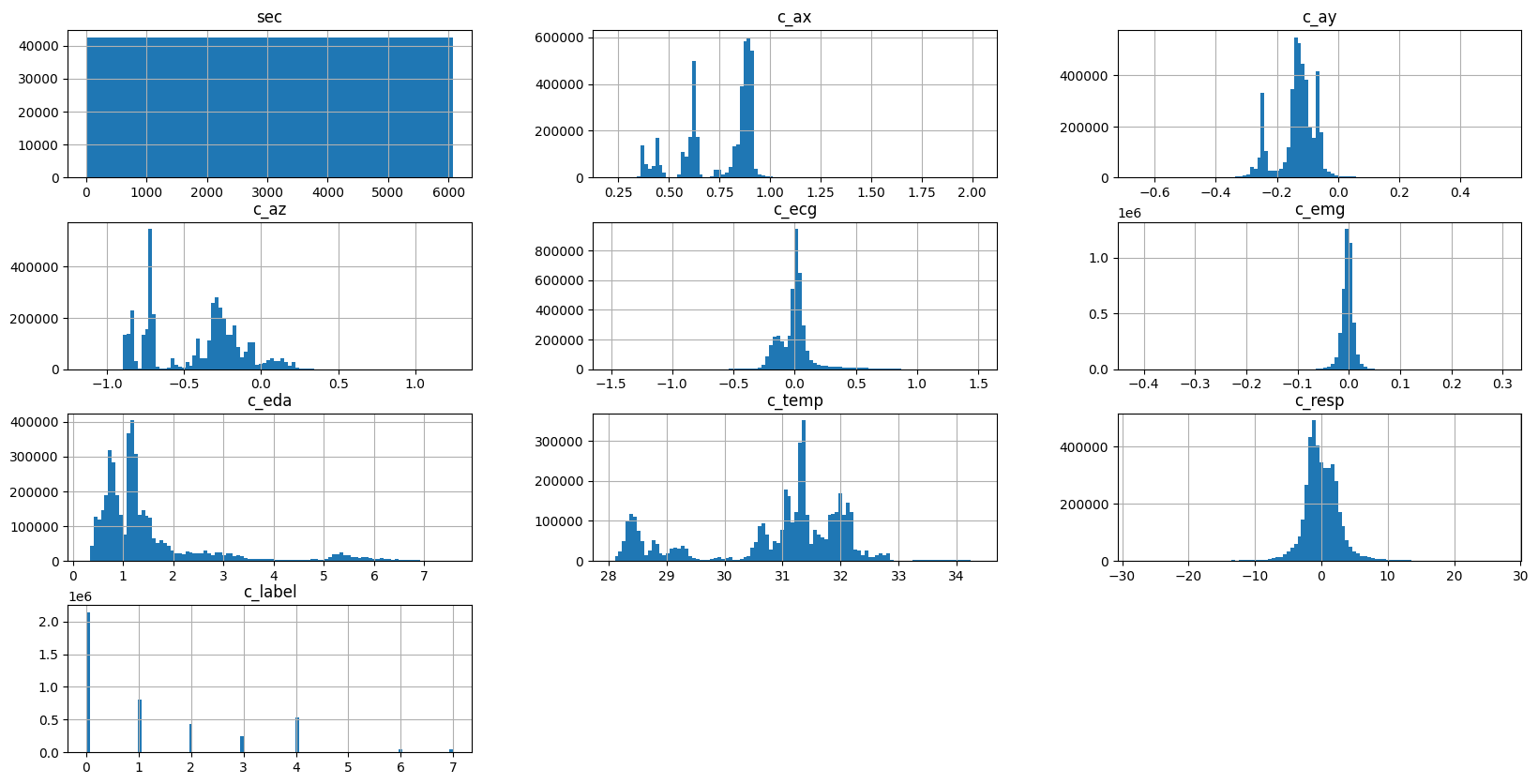
# P.8 Histogrami vrednosti

dfc.hist(column=['c\_ax','c\_ay','c\_az'], bins=50, figsize=(10,6))

dfc.hist(bins=100, figsize=(20,10))







## Podatki ločeno po stanjih osebe

### Dataframe za vsako stanje

dfc\_base = dfc.loc[(dfc['c\_label'] == 1)]

dfc\_stress = dfc.loc[(dfc['c\_label'] == 2)]

dfc\_amus = dfc.loc[(dfc['c\_label'] == 3)]

dfc\_medi = dfc.loc[(dfc['c\_label'] == 4)]

print("Baseline samples: ", dfc\_base.shape)

print("Stress samples: ", dfc\_stress.shape)

ind\_b = dfc['c\_label']==1

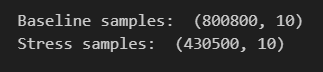
ind\_s = dfc['c\_label']==2

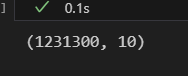
ind\_base\_stress =  (dfc['c\_label']==1) | (dfc['c\_label']==2)

# Dataframe baseline, stres

dfc\_base\_str = dfc.loc[ind\_base\_stress]

print(dfc\_base\_str.shape)





## P.10 Izris poteka signalov po stanjih

Izriši izbrano število vzorcev signala.

Primer za pospešek:

# P.10 Izris signalov

sample\_beg = 120000

sample\_length = 10000

#window\_sec = 20

#sample\_length = 700\*window\_sec

sample\_end = sample\_beg + sample\_length

y\_ax =  dfc\_base["c\_ax"].iloc[sample\_beg:sample\_end].values

y\_ax\_s= dfc\_stress["c\_ax"].iloc[sample\_beg:sample\_end].values

y\_ax\_a= dfc\_amus["c\_ax"].iloc[sample\_beg:sample\_end].values

y\_ax\_m= dfc\_medi["c\_ax"].iloc[sample\_beg:sample\_end].values

x\_ax=np.arange(len(y\_ax))

fig= plt.figure(figsize=(6,3))

ax1= fig.add\_axes([0.1,0.1,2,2])

#fig1, ax1 = plt.subplots()

ax1.plot(x\_ax,y\_ax,color='green',label="acc X base")

ax1.plot(x\_ax,y\_ax\_s,color='red',label="acc X stress")

ax1.plot(x\_ax,y\_ax\_a,color='blue',label="acc X amusement")

ax1.plot(x\_ax,y\_ax\_m,color='black',label="acc X meditation")

ax1.set(xlabel='number of samples', ylabel = 'sensor amplitude', title = 'AX base vs stress vs amusement vs meditation')

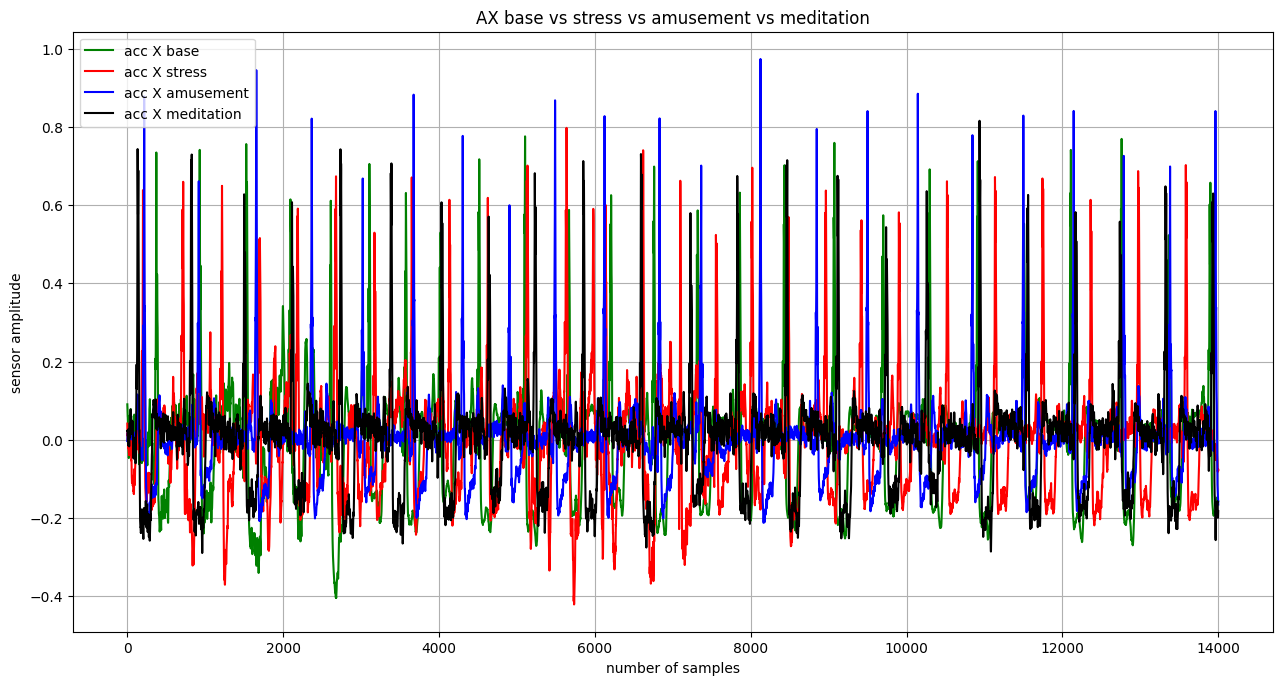
ax1.grid()

plt.legend(loc='upper left')

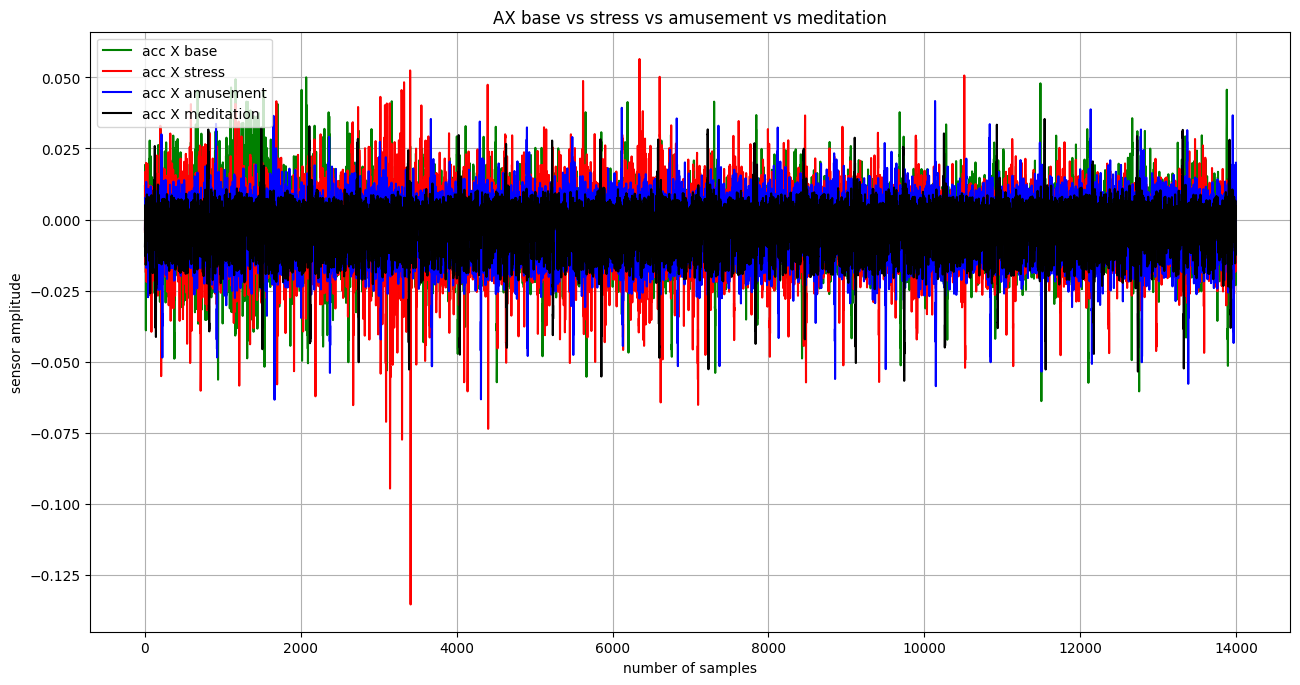
plt.show()

**Izriši časovni potek vzorcev : ECG, EMG, EDA, RESP, TEMP po stanjih. Preskusi različen začetek in število vzorcev. Ali se iz oblike signalov lahko razlikuje stanja (v vašem primeru)?**

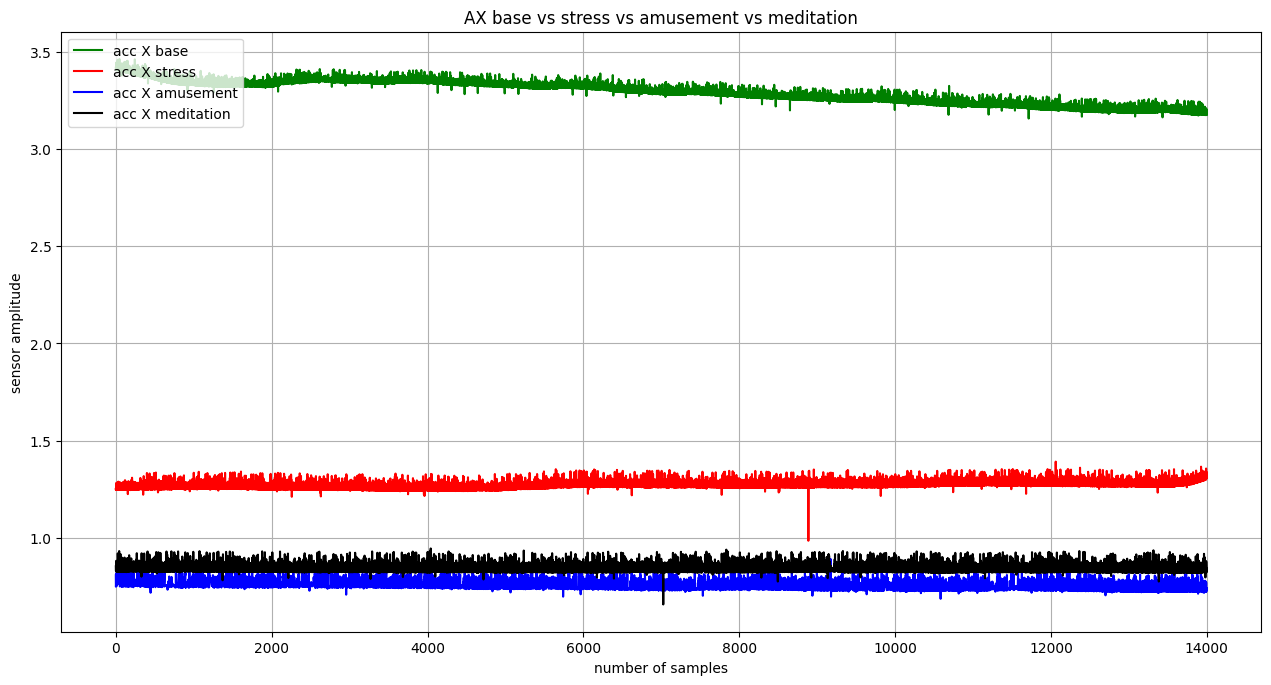
**ECG (Elektrokardiogram): Elektrokardiogram (EKG ali ECG) je zapis električne aktivnosti srca. S senzorji, pritrjenimi na kožo, se merijo električni signali, ki jih ustvarja srčna mišica med utripanjem. Na podlagi teh podatkov lahko zdravniki ocenijo delovanje srca, odkrijejo morebitne nepravilnosti v ritmu ali srčni prevodnosti.**



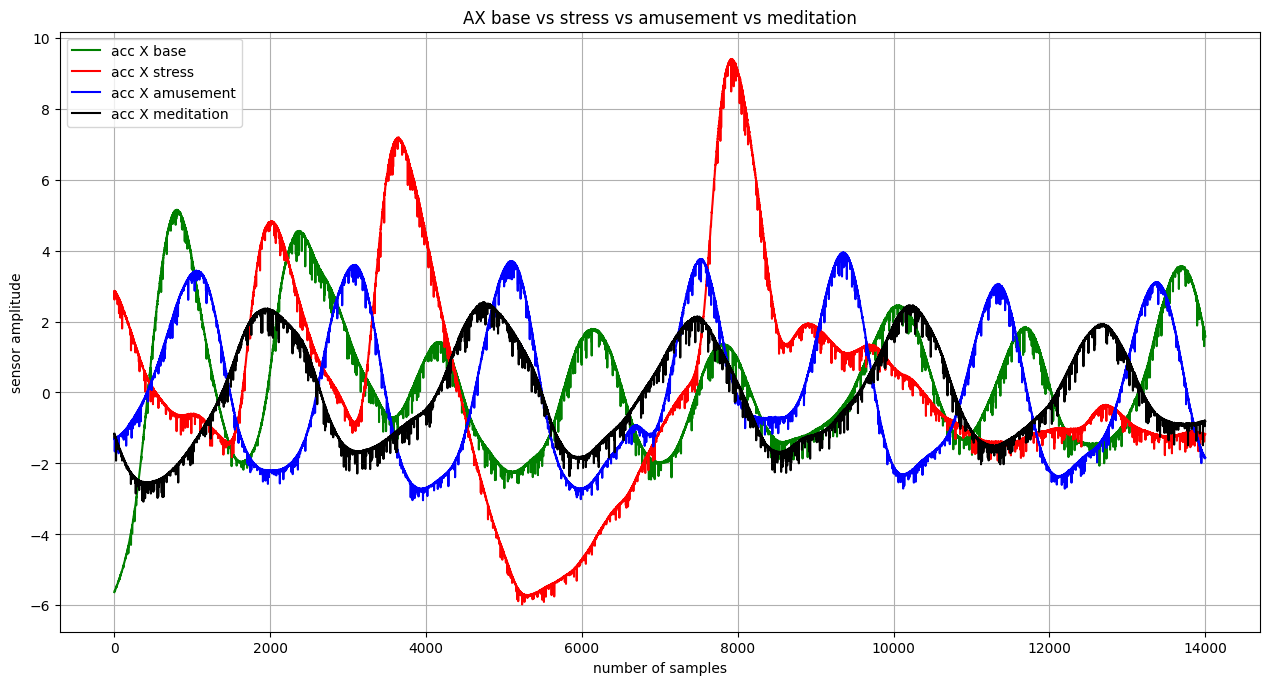
**EMG (Elektromiogram): Elektromiogram (EMG) meri električno aktivnost mišic. Ta test zaznava in zapisuje električne signale, ki jih mišice proizvajajo med krčenjem in sproščanjem. EMG se uporablja pri diagnosticiranju mišičnih bolezni in oceni delovanja živčno-mišičnega sistema.**



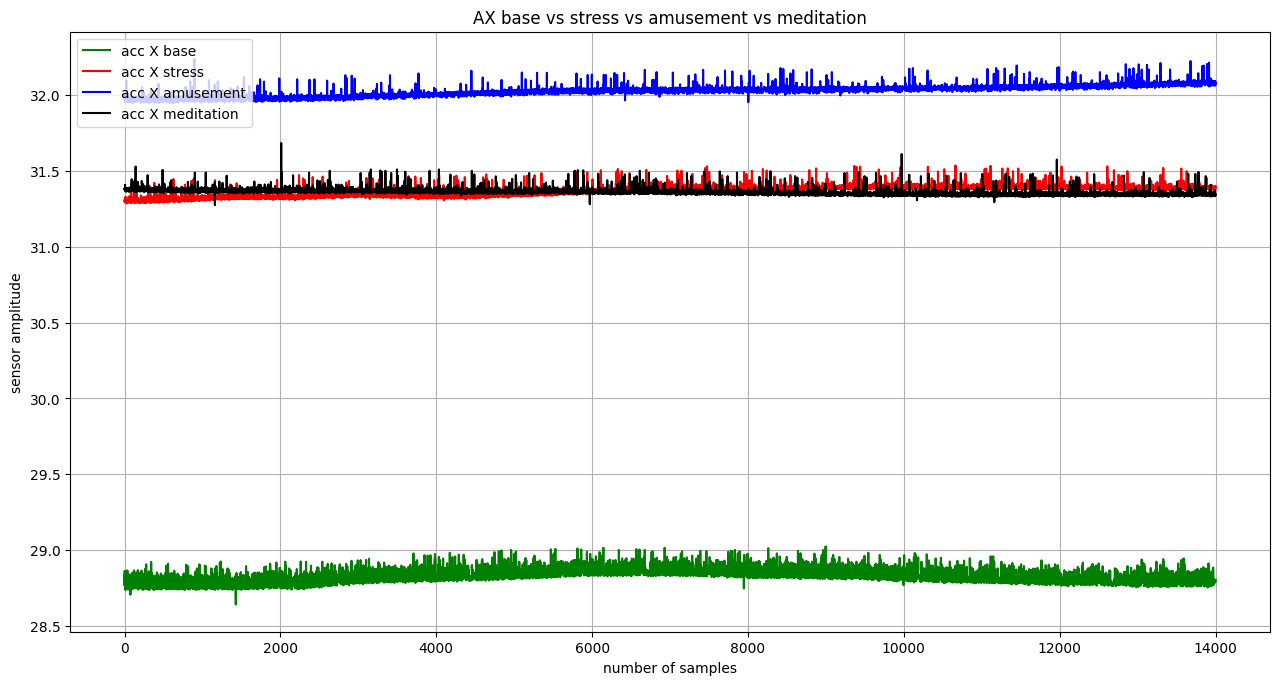
**EDA (Elektrodermalna Aktivnost): Elektrodermalna aktivnost (EDA) je meritev sprememb v električni prevodnosti kože, ki se pojavijo zaradi znojenja. Te spremembe so povezane z aktivnostjo simpatičnega živčnega sistema, zato je EDA pogosta metoda za spremljanje čustvenih odzivov, stresa in vzburjenosti.**

****

**RESP (Respiracija): Respiracija (RESP) označuje meritev dihalne aktivnosti, tj. kako pogosto in kako globoko človek diha. S tem lahko spremljamo stanje dihalnega sistema, odkrivamo nepravilnosti pri dihanju, ter spremljamo stresne odzive ali sproščenost.**

****

**TEMP (Temperatura): Temperatura (TEMP) meri telesno temperaturo, kar je pomemben vitalni znak za spremljanje zdravja. Spremembe telesne temperature lahko kažejo na prisotnost okužbe, vnetja ali druge fiziološke spremembe.**



## P.11 Primerjava histogramov bazičnega in stresa

import seaborn as sns

for col in dfc\_base\_str.columns[1:3]:

    fg = sns.FacetGrid(dfc\_base\_str, hue="c\_label", aspect=5)

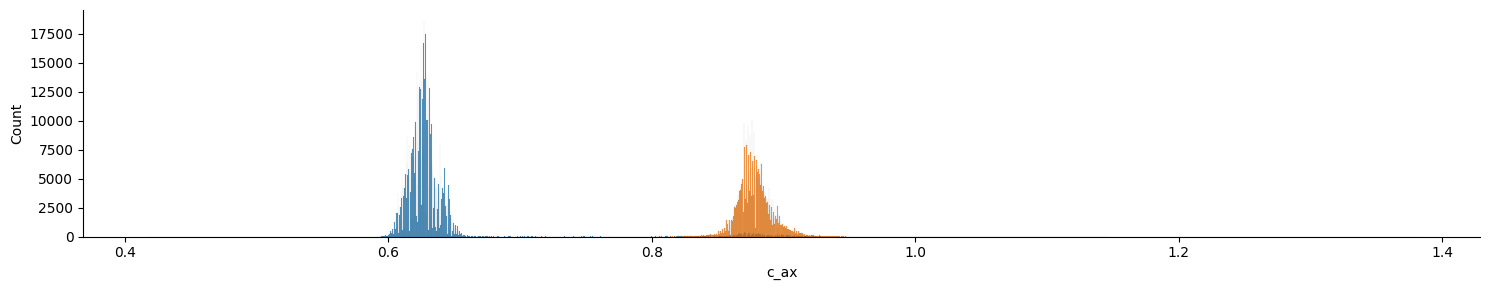
    fg.map(sns.histplot, col, fill=True)

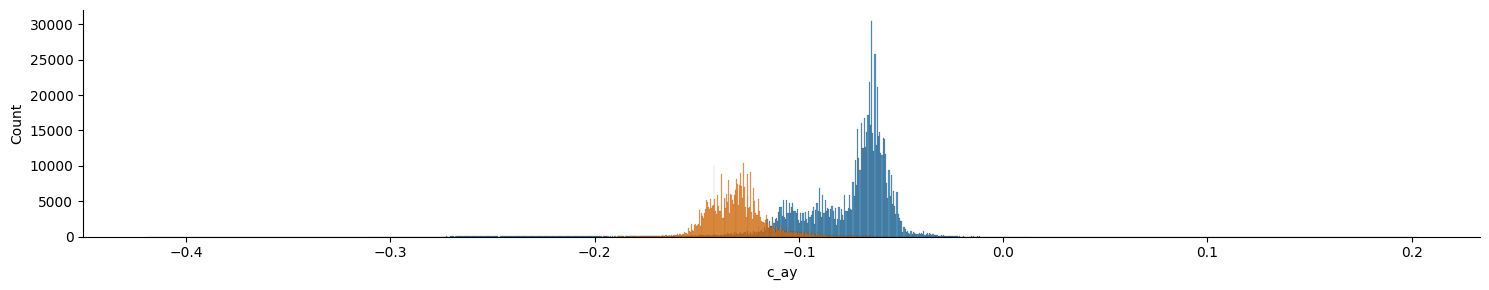
plt.show()

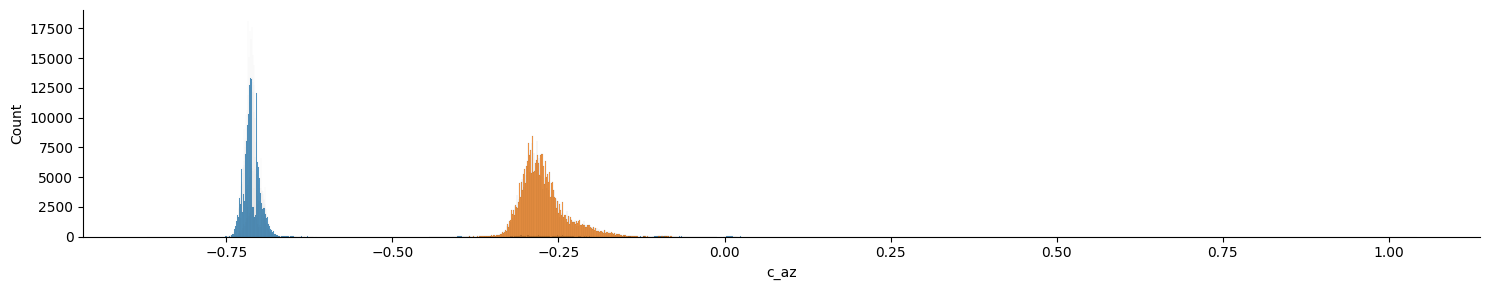
**Izriši in komentiraj histograme vseh signalov, ali se razlikujejo med stanji, ali je torej iz vrednosti signala možno ločiti dve stanji.**

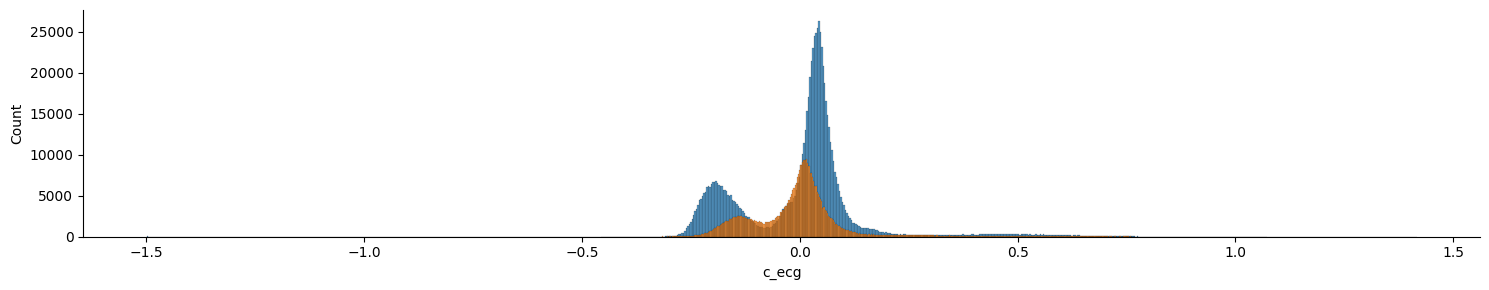
**Primerjal sem bazo (modra) in stres (oranžna) stanja. Bolj kot se podatkovne točke prekrivajo, manj primerna je značilka za razlikovanje med tema dvema stanji. Če so podatkovne točke za bazo in stresna stanja tesno skupaj, to pomeni, da značilka ni primerna za zaznavanje razlik med tema stanjema. To nakazuje, da je bila oseba pod stresom, vendar ta značilka ni bila učinkovita pri zaznavanju značilnosti stresa.**

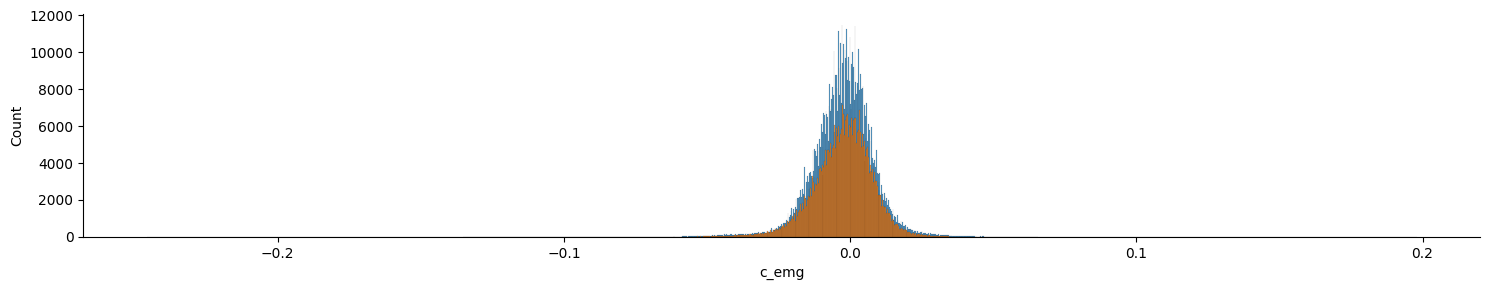
**Primerjamo bazo in stresna stanja, da ocenimo učinkovitost značilk pri razlikovanju med tema dvema pogojema.**

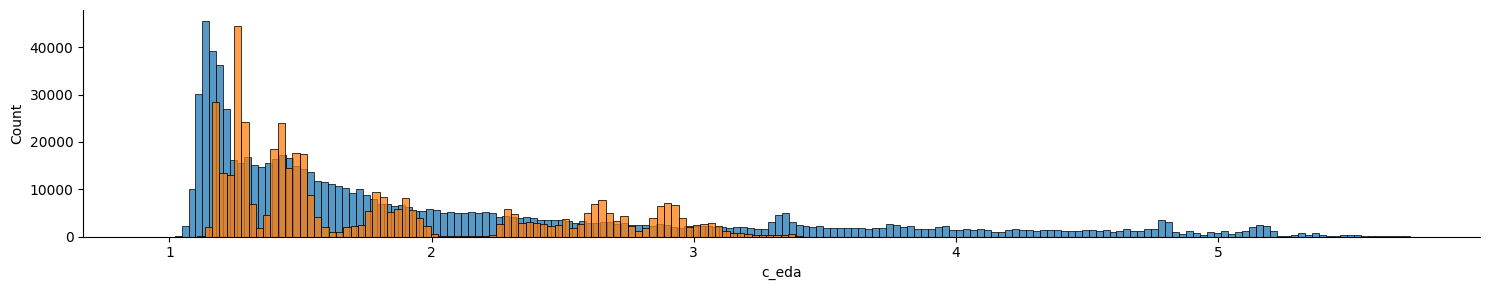


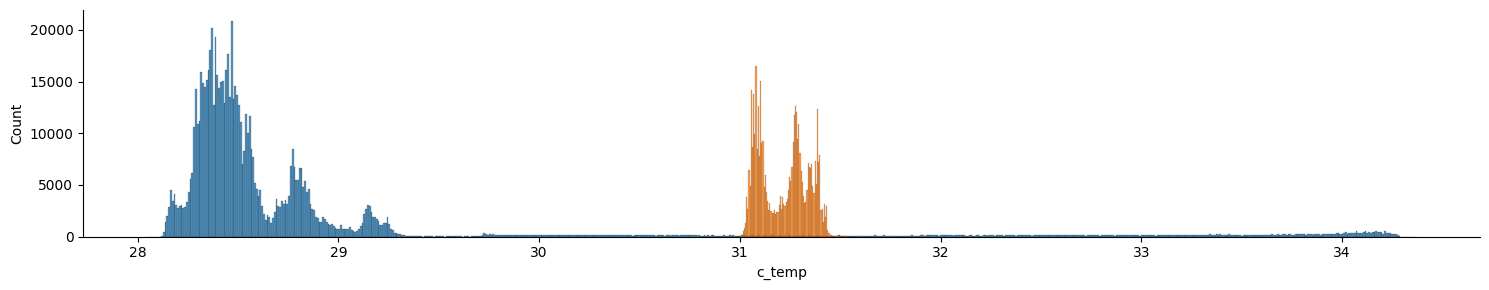


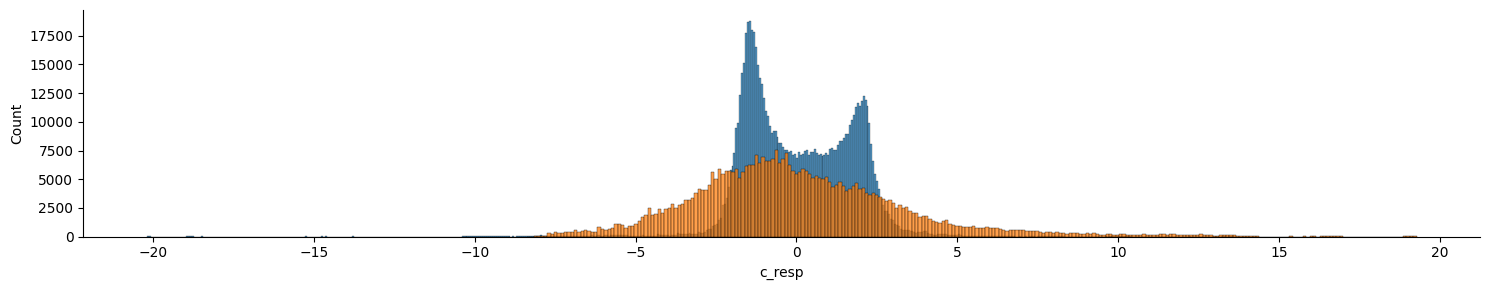












Preskusi še KDE plot (kernel density estimation) za izris verjetnostne porazdelitve vrednosti.

for col in dfc\_base\_str.columns[1:6]:

    fg = sns.FacetGrid(dfc\_base\_str, hue="c\_label", aspect=5)

    fg.map(sns.kdeplot, col, fill=True)

plt.show()

