ptsd-project (/github/Eliabicocca/ptsd-project/tree/master) / tesi_filippo.ipynb (/github/Eliabicocca/ptsd-project/tree/master/tesi_filippo.ipynb)

In [104]:
```
# apro il dataset 'definitivo.csv'

import pandas as pd
data=pd.read_csv('./definitivo.csv', sep=';')
data.head(10)
```

Out[104]:

	cod	nome	cognome	age	studM	profM	studP	profP	steril	sterilTer	...	F3Beck	F3intrus	F3arous	F3avoid	F3PTSD	F3PTSDpart	F3intrus.1
0	10	laura	bellingeri	38	4.0	2.0	3.0	3.0	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	12	francesca	era	41	3.0	2.0	NaN	3.0	1.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	39	elena	sala	26	2.0	2.0	2.0	1.0	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	76	silvia	dalla valle	34	4.0	3.0	4.0	3.0	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	84	camilla	ricciardi	34	4.0	3.0	4.0	NaN	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	86	Stefania	Honegger	39	3.0	3.0	4.0	NaN	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	106	francesca	busio	30	4.0	3.0	NaN	3.0	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	117	elena	ventura	34	3.0	3.0	3.0	2.0	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	126	assunta	giammarino	28	3.0	2.0	3.0	2.0	0.0	0.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	127	anna maria	atzori	33	4.0	3.0	NaN	3.0	0.0	1.0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10 rows × 244 columns

Proporzione depresse in F2 (con valore F2Beck>10)

In [51]:
```
d=data.groupby('F2Beck').size();
d2=data['F2Beck']; # 351 osservazioni
len(d2.dropna())  # 346 osservazioni se levo gli NaN
sum(d[11:]) # 26 depresse in F2Beck
sum(d[11:])/len(d2.dropna()) # 0.075 è la proporzione di depresse in F2 (F2Beck > 10)
```

Out[51]: 0.07514450867052024

Proporzione depresse in F1 (con valore F1Beck>10) e proporzione depresse gravi (con valore F1Beck>30)

In [61]:
```
d=data.groupby('F1Beck').size();
d2=data['F1Beck']; # 351 osservazioni
len(d2.dropna())  # 81 osservazioni se levo gli NaN
sum(d[11:]) # 13 depresse in F2Beck
sum(d[11:])/len(d2.dropna()) # 0.160 è la proporzione di depresse in F2 (F2Beck > 10)
```

Out[61]: 0.16049382716049382

In [63]:
```
sum(d[31:]) # 13 depresse in F2Beck
sum(d[31:])/len(d2.dropna()) # nessuna depressione grave (con F1Beck > 30)
```

Out[63]: 0.0

Statistiche descrittive per F1Beck e F2Beck:

In [176]:
```
data['F2STAIS'].mean()    # 45.204 è la media per stais
data['F2STAIS'].var()     # 25.812 è la varianza per stais
data['F2STAIT'].mean()    # 45.034 è la media per stait
data['F2STAIT'].var()     # 20.338 è la varianza per stait
```

Out[176]: 20.33795372785705

Quante presentano ansia di stato (F2STAIS>49) e quante ansia di tratto (F2STAIT>46)

In [78]:
```
d=data.groupby('F2STAIS').size();
sum(d[50:]) # 53 presentano valori di ansia di stato superiori a 49
d2=data['F2STAIS'];
len(d2.dropna())   # se levo i missing, rimangono 343 osservazioni
sum(d[50:])/len(d2.dropna())   # 0.155 è la proporzione di quante presentano ansia di stato
```

Out[78]: 0.15451895043731778

In [84]:
```
d=data.groupby('F2STAIT').size();
sum(d[47:]) # 122 presentano valori di ansia di tratto superiori a 46
d2=data['F2STAIT'];
len(d2.dropna())   # se levo i missing, rimangono 343 osservazioni
sum(d[47:])/len(d2.dropna())   # 0.356 è la proporzione di quante presentano ansia di stato
```

Out[84]: 0.3556851311953353

Statistiche descrittive per F2STAIS e F2STAIT:

In [93]:
```
data['F2STAIS'].mean()    # 45.204 è la media per stais
data['F2STAIS'].var()     # 25.812 è la varianza per stais
data['F2STAIT'].mean()    # 45.034 è la media per stait
data['F2STAIT'].var()     # 20.338 è la varianza per stait
```

Out[93]: 20.33795372785705

Quante presentano ptsd completo (f2ptsdcompl) e quante presentano ptsd parziale (f2ptsdpart)

In [116]:
```
data.groupby('F2PTSDpart').size() # 254 pazienti su un totale di 346
len(data[data.F2PTSDpart==1])/346 # la proporzione è di 0.734 per ptsd parziale
```

Out[116]: 0.7341040462427746

In [114]:
```
data.groupby('F2PTSDcomp').size() # 23 pazienti su un totale di 344
len(data[data.F2PTSDcomp==1])/344 # la proporzione è di 0.067 per ptsd completo
```

Out[114]: 0.06686046511627906

Calcolo frequenze assolute e percentuali di quante hanno sintomi di tipo avoid, intrusion e arousal (1 o più sintomi per avoid, 3 o più sintomi per intrusion e 2 o più sintomi per arousal). Con la vecchia scala:

In [164]:
```
data2=data[['F2avoid','F2intrusion','F2arousal']]
data2_a=data2['F2avoid'].dropna(); len(data2_a) # 348 osservazioni se escludo i missing
len(data2[data2.F2avoid!=0])  # 172 osservazioni presentano diverso pari a zero
prop_a=172/348; prop_a   # 0.494 è la frequenza relativa (49,425%) di f2 arousal
```

Out[164]: 0.4942528735632184

In [165]:
```
data2_b=data2['F2intrusion'].dropna(); len(data2_b) # 348 osservazioni se escludo i missing
len(data2[data2.F2intrusion>2])  # 108 osservazioni presentano diverso pari a zero
prop_b=108/348; prop_b   # 0.310 è la frequenza relativa (31,034%) di f2 intrusion
```

Out[165]: 0.3103448275862069

In [289]:
```
data2_c=data2['F2arousal'].dropna(); len(data2_c) # 348 osservazioni se escludo i missing
len(data2[data2.F2arousal>1])  # 64 osservazioni presentano diverso pari a zero
prop_c=64/348; prop_c   # 0.184 è la frequenza relativa (46,839%) di f2 intrusion
```

Out[289]: 0.1839080459770115

Con la nuova scala (avoid la proporzione rimane la stessa):

In [290]:
```
len(data2[data2.F2intrusion>0])  # 251 osservazioni presentano diverso pari a zero
prop_b2=251/348; prop_b2   # 0.721 è la frequenza relativa (31,034%) di f2 intrusion
```

Out[290]: 0.7212643678160919

In [293]:
```
len(data2[data2.F2arousal>0])  # 148 osservazioni presentano diverso pari a zero
prop_c2=148/348; prop_c2   # 0.425 è la frequenza relativa (46,839%) di f2 arousal
```

Out[293]: 0.42528735632183906

Correlazione tra F2Beck e avoid, intrusion e arousal

In [175]:
```
data2['F2Beck']=data['F2Beck']; data2
d=data2.corr(); d['F2Beck']   # correlazione lineare positiva lieve
```

```
/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-vers
us-copy
  """Entry point for launching an IPython kernel.
```

Out[175]:
```
F2avoid        0.265713
F2intrusion    0.242470
F2arousal      0.327280
F2Beck         1.000000
Name: F2Beck, dtype: float64
```

In [186]:
```
print(data2.corr(method='spearman')['F2Beck'])
print(data2.corr(method='kendall')['F2Beck'])  # correlazione più scarsa con gli altri metodi
```

```
F2avoid        0.240704
F2intrusion    0.235113
F2arousal      0.260044
F2Beck         1.000000
Name: F2Beck, dtype: float64
F2avoid        0.190864
F2intrusion    0.178834
F2arousal      0.212895
F2Beck         1.000000
Name: F2Beck, dtype: float64
```

Correlazione tra f2beck con f2ptsdpart e con f2ptsdcomp

In [174]:
```
data3=data[['F2Beck','F2PTSDpart','F2PTSDcomp']]
data3.corr()    # correlazione lineare positiva tenue
```

Out[174]:

	F2Beck	F2PTSDpart	F2PTSDcomp
F2Beck	1.000000	0.168323	0.293375
F2PTSDpart	0.168323	1.000000	0.133956
F2PTSDcomp	0.293375	0.133956	1.000000

I valori scarsi di correlazione possono essere dovuti alla costruzione delle variabili (che sono di tipo categoriali ordinate); proviamo con il metodo di spearman (d
default viene utilizzato pearson):

In [181]:
```
data3.corr(method='spearman')    # con spearman è più basso
```

Out[181]:

	F2Beck	F2PTSDpart	F2PTSDcomp
F2Beck	1.000000	0.161822	0.234041
F2PTSDpart	0.161822	1.000000	0.133956
F2PTSDcomp	0.234041	0.133956	1.000000

In [182]:
```
data3.corr(method='kendall')     # kendall peggio ancora
```

Out[182]:

	F2Beck	F2PTSDpart	F2PTSDcomp
F2Beck	1.000000	0.138396	0.200170
F2PTSDpart	0.138396	1.000000	0.133956
F2PTSDcomp	0.200170	0.133956	1.000000

Correlazione tra f2beck con f2stais e con f2stait

In [189]:
```
data4=data[['F2Beck','F2STAIS','F2STAIT']]
data4.corr()['F2Beck']
```

Out[189]:
```
F2Beck     1.000000
F2STAIS    0.002306
F2STAIT    0.094572
Name: F2Beck, dtype: float64
```

In [190]:
```
data4.corr(method='spearman')['F2Beck']
```

Out[190]:
```
F2Beck     1.000000
F2STAIS   -0.060677
F2STAIT    0.094288
Name: F2Beck, dtype: float64
```

In [191]:
```
data4.corr(method='kendall')['F2Beck']
# lieve correlazione negativa per stais, lieve correlazione ma positiva per stait
```

Out[191]:
```
F2Beck     1.000000
F2STAIS   -0.044426
F2STAIT    0.068068
Name: F2Beck, dtype: float64
```

Correlazione tar ptsd (part) con: studiM, studiF, profM, profP, steril, sterilT

In [194]:
```
data5=data[['F2PTSDpart','studM','studP','profM','profP','steril','sterilTer']]
data5.corr()['F2PTSDpart']
```

Out[194]:
```
F2PTSDpart    1.000000
studM         0.061986
studP        -0.080017
profM         0.050552
profP        -0.099515
steril        0.112852
sterilTer    -0.003597
Name: F2PTSDpart, dtype: float64
```

In [195]:
```
data5.corr(method='spearman')['F2PTSDpart']
```

Out[195]:
```
F2PTSDpart    1.000000
studM         0.066965
studP        -0.068089
profM         0.047280
profP        -0.088787
steril        0.112852
sterilTer    -0.003597
Name: F2PTSDpart, dtype: float64
```

In [196]:
```
data5.corr(method='kendall')['F2PTSDpart']  # segni uguali, valori di correlazione lievi
```

Out[196]:
```
F2PTSDpart    1.000000
studM         0.065057
studP        -0.065115
profM         0.044672
profP        -0.083954
steril        0.112852
sterilTer    -0.003597
Name: F2PTSDpart, dtype: float64
```

Correlazione tar ptsd (comp) con: studiM, studiF, profM, profP, steril, sterilT

In [197]:
```
data6=data[['F2PTSDcomp','studM','studP','profM','profP','steril','sterilTer']]
data6.corr()['F2PTSDcomp']
```

Out[197]:
```
F2PTSDcomp    1.000000
studM        -0.008954
studP        -0.008549
profM         0.025156
profP        -0.079560
steril       -0.041345
sterilTer     0.016215
Name: F2PTSDcomp, dtype: float64
```

In [199]:
```
data6.corr(method='spearman')['F2PTSDcomp']
```

Out[199]:
```
F2PTSDcomp    1.000000
studM        -0.022721
studP        -0.009654
profM         0.018234
profP        -0.079093
steril       -0.041345
sterilTer     0.016215
Name: F2PTSDcomp, dtype: float64
```

In [200]:
```
data6.corr(method='kendall')['F2PTSDcomp']
```

Out[200]:
```
F2PTSDcomp    1.000000
studM        -0.022071
studP        -0.009231
profM         0.017227
profP        -0.074843
steril       -0.041345
sterilTer     0.016215
Name: F2PTSDcomp, dtype: float64
```

Distribuzioni per le variabili 24h e per tipo di parto, analgesia

In [203]:
```
data7=data[['>24h','parto','analgesia']]
data7.describe()
```

Out[203]:

	>24h	parto	analgesia
count	350.000000	350.000000	350.000000
mean	1.254286	1.394286	1.411429
std	0.536273	0.737044	0.492797
min	1.000000	1.000000	1.000000
25%	1.000000	1.000000	1.000000
50%	1.000000	1.000000	1.000000
75%	1.000000	1.000000	2.000000
max	3.000000	3.000000	2.000000

In [208]:
```
data7.groupby('>24h').size()   # valori assoluti
data7.groupby('>24h').size()/len(data7['>24h'])  # frequenze relative
(data7.groupby('>24h').size()/len(data7['>24h']))*100 # categorie percentuali sul totale
```

Out[208]:
```
>24h
1.0    79.202279
2.0    15.669516
3.0     4.843305
dtype: float64
```

In [209]:
```
data7.groupby('parto').size()   # valori assoluti
data7.groupby('parto').size()/len(data7['parto'])  # frequenze relative
(data7.groupby('parto').size()/len(data7['parto']))*100 # categorie percentuali sul totale
```

Out[209]:
```
parto
1.0    75.498575
2.0     9.116809
3.0    15.099715
dtype: float64
```

In [211]:
```
data7.groupby('analgesia').size()   # valori assoluti
data7.groupby('analgesia').size()/len(data7['analgesia'])  # frequenze relative
(data7.groupby('analgesia').size()/len(data7['analgesia']))*100 # percentuali sul totale
```

Out[211]:
```
analgesia
1.0    58.689459
2.0    41.025641
dtype: float64
```

Correlazione tra ptsd e nullipare

In [261]:
```
d3=data[['F2PTSDcomp','para']]
data8=d3[d3.para==0]; len(data8)  # 219 nullipare
d=data8.dropna(axis=0); d # 1 osservazione ha NaN per Ptsd
data8 = data8.drop(data8.index[5]); data8
data8.corr()  # proviamo in un altro modo
```

Out[261]:

	F2PTSDcomp	para
F2PTSDcomp	1.0	NaN
para	NaN	NaN

In [263]:
```
data8=data8[data8.F2PTSDcomp==0]; len(data8)
# 200 donne su 218 nullipare presentano f2ptsdcomp=0
```

Out[263]: 200

Calcolo frequenze assolute, relative e percentuali di: para, pre-tc, aborti, grav

In [269]:
```
data9=data[['para','grav','pre-tc','aborti']];
data9.groupby('para').size()     # frequenze assolute
data9.groupby('para').size()/len(data9['para'])  # frequenze relative
(data9.groupby('para').size()/len(data9['para']))*100 # categorie percentuali sul totale
```

Out[269]:
```
para
0.0    62.393162
1.0    37.321937
dtype: float64
```

In [270]:
```
data9.groupby('grav').size()     # frequenze assolute
data9.groupby('grav').size()/len(data9['grav'])  # frequenze relative
(data9.groupby('grav').size()/len(data9['grav']))*100 # categorie percentuali sul totale
```

Out[270]:
```
grav
1.0    44.444444
2.0    55.270655
dtype: float64
```

In [271]:
```
data9.groupby('pre-tc').size()     # frequenze assolute
data9.groupby('pre-tc').size()/len(data9['pre-tc'])  # frequenze relative
(data9.groupby('pre-tc').size()/len(data9['pre-tc']))*100 # categorie percentuali sul totale
```

Out[271]:
```
pre-tc
0.0    91.168091
1.0     8.547009
dtype: float64
```

In [272]:
```
data9.groupby('aborti').size()     # frequenze assolute
data9.groupby('aborti').size()/len(data9['aborti'])  # frequenze relative
(data9.groupby('aborti').size()/len(data9['aborti']))*100 # categorie percentuali sul totale
```

Out[272]:
```
aborti
0.0    80.626781
1.0    19.088319
dtype: float64
```

Calcolo quante hanno ptsd molto gravi (variabile 'f2ptsd>33')

In [281]:
```
d4=data.groupby('F2Tptsd').size()
sum(d4[34:])    # 47 pazienti presentano ptsd grave
sum(d4[34:])/sum(d4[0:])   # 0.135 è la frequenza relativa di chi presenta ptsd grave
sum(d4[34:])/sum(d4[0:])*100 # 13.584 è la percentuale di colore che presentano ptsd grave
```

Out[281]:
```
13.583815028901732
```

Correlazione PPO con f2stais e f2stait

In [295]:
```
data9=data[['PPO','F2STAIT','F2STAIS']]
data9.corr()
```

Out[295]:

	PPO	F2STAIT	F2STAIS
PPO	1.000000	0.084594	0.011985
F2STAIT	0.084594	1.000000	0.430778
F2STAIS	0.011985	0.430778	1.000000

In [299]:
```
data9.corr(method='spearman')
```

Out[299]:

	PPO	F2STAIT	F2STAIS
PPO	1.000000	0.087810	0.043371
F2STAIT	0.087810	1.000000	0.410559
F2STAIS	0.043371	0.410559	1.000000

Correlazione ptsd con: episio, perditaem, LA, second

In [352]:
```
data10=data[['F2Tptsd','perditaem','LA','second','episio']]
data10.corr()
```

Out[352]:

	F2Tptsd	perditaem	LA	second	episio
F2Tptsd	1.000000	0.060161	-0.024849	0.043557	0.147706
perditaem	0.060161	1.000000	0.035261	0.311368	0.190789
LA	-0.024849	0.035261	1.000000	0.068198	0.091271
second	0.043557	0.311368	0.068198	1.000000	-0.012635
episio	0.147706	0.190789	0.091271	-0.012635	1.000000

Correlazione f2beck e ptsd (part) e alto/basso rischio (calcolato come: - alto rischio: più di due valori diversi da zero tra vomito, altritmi, problgrav, ricgrav, abort PPO, minaccab e basso rischio altrimenti)

In [526]:
```
data11=data[['F2Beck','F2PTSDpart','vomito','altritmi','problgrav','ricgrav',
             'aborti','PPO','minaccab','F2STAIS','F2STAIT','F2Tptsd','peso','age','posizparto']];
data11=data11.dropna()
data11['somma sintomi']=data11['vomito']+data11['altritmi']+data11['problgrav']+data11['ricgrav']+data11['aborti']+data
['PPO']+data11['minaccab']
data11.head(10)

l=[]
for i in data11['somma sintomi']:
    if i>2:
        l.append(1)
    else:
        l.append(0)

data11['rischio']=l
data11.groupby('rischio').size()   # quanti alto rischio e quanti basso rischio
```

Out[526]:
```
rischio
0     48
1    222
dtype: int64
```

In [504]:
```
data11.head(20)
data11bis=data11[['F2PTSDpart','F2Beck','F2STAIS','F2STAIT','rischio']]; data11bis.head()
```

Out[504]:

	F2PTSDpart	F2Beck	F2STAIS	F2STAIT	rischio
0	1.0	2.0	49.0	47.0	1
1	1.0	17.0	49.0	43.0	1
2	0.0	5.0	45.0	46.0	0
3	1.0	9.0	44.0	48.0	1
4	1.0	4.0	38.0	44.0	1

In [342]:
```
data11bis.corr()
data11bis.corr(method='spearman')
data11bis.corr(method='kendall')
```

Out[342]:

	F2PTSDpart	F2Beck	rischio
F2PTSDpart	1.000000	0.139735	0.029162
F2Beck	0.139735	1.000000	0.042821
rischio	0.029162	0.042821	1.000000

F3ptsd come correla con i vari sintomi

In [338]:
```
data12=data[['F3Tptsd','vomito','altritmi','problgrav','ricgrav','aborti','PPO','minaccab']]
data12=data12.dropna(); len(data12)
```

Out[338]: 81

In [339]:
```
data12['somma sintomi']=data12['vomito']+data12['altritmi']+data12['problgrav']+data12['ricgrav']+data12['aborti']+data
['PPO']+data12['minaccab']
data12.head(10)

l2=[]
for i in data12['somma sintomi']:
    if i>2:
        l2.append(1)
    else:
        l2.append(0)

data12['rischio']=l2

data12.corr()['F3Tptsd']
```

Out[339]:
```
F3Tptsd          1.000000
vomito          -0.029313
altritmi         0.262897
problgrav       -0.128114
ricgrav          0.216813
aborti          -0.052633
PPO              0.218504
minaccab         0.146592
somma sintomi    0.240040
rischio          0.240996
Name: F3Tptsd, dtype: float64
```

Verifica se alzando la soglia a 3 la correlazione tra somma di sintomi e f2ptsd e f3ptsd si alza:

Quanti hanno ptsd molto gravi (variabile 'f3ptsd>33').

In [351]:
```
d5=data.groupby('F3Tptsd').size(); # totale pazienti con f3ptsd rilevato è 85
sum(d5[34:])    # 21 pazienti presentano ptsd grave
sum(d5[34:])/sum(d5[0:])   # 0.247 è la frequenza relativa di chi presenta ptsd grave
sum(d5[34:])/sum(d5[0:])*100 # 24,706 è la percentuale di colore che presentano ptsd grave
```

Out[351]:
```
24.705882352941178
```

Età media delle donne

In [354]:
```
data['age'].mean()
```

Out[354]:
```
33.267806267806264
```

Correlazione f2stait e f2 stait

In [357]:
```
data.corr()['F2STAIS']['F2STAIT']
```

Out[357]:
```
0.4307783434533827
```

Correlazione f1beck e f2beck

In [358]:
```
data.corr()['F1Beck']['F2Beck']
```

Out[358]:
```
0.4929296733362172
```

Conteggio depresse e ansiose

In [385]:
```
l=[]
for i in data['F2STAIS']:
    if i>49:
        l.append(1)
    else:
        l.append(0)

data['stais']=l

l2=[]
for i in data['F2STAIT']:
    if i>46:
        l2.append(1)
    else:
        l2.append(0)

data['stait']=l2

data12=data[['stais','stait','F2Beck']]; data12.head()

data12a=data12[data12.stais==1];
data12a
d12a=data12a.groupby('F2Beck').size(); d12a
sum(d12a[11:])      # sono 4 depresse e ansiose stais
```

Out[385]:
```
F2Beck
0.0     11
1.0      5
2.0      7
3.0      9
4.0      8
5.0      1
6.0      3
8.0      1
9.0      3
12.0     2
14.0     1
15.0     1
dtype: int64
```

In [388]:
```
data12b=data12[data12.stait==1];
data12b
d12b=data12b.groupby('F2Beck').size(); d12b
sum(d12b[11:])      # sono 11 depresse e ansiose stait
```

Out[388]: 11

Correlazione f2beck, f2stais e f2stait con PPO=2

In [397]:
```
data13=data[['F2Beck','F2STAIS','F2STAIT','PPO']]
data13=data13.dropna()
data13=data13[data13.PPO==2]; data13
data13.corr() # NaN le corr con PPO, dato che PPO=2 è degenere (assume sempre lo stesso valore)
```

Out[397]:

	F2Beck	F2STAIS	F2STAIT	PPO
F2Beck	1.000000	0.003153	0.182596	NaN
F2STAIS	0.003153	1.000000	0.306651	NaN
F2STAIT	0.182596	0.306651	1.000000	NaN
PPO	NaN	NaN	NaN	NaN

Correlazione f2beck, f2stais e f2stait con PPO=2

In [398]:
```
data14=data[['F3Beck','F3STAIS','F3STAIT','PPO']]
data14=data14.dropna()
data14=data14[data14.PPO==2]; data13
data14.corr() # NaN le corr con PPO, dato che PPO=2 è degenere (assume sempre lo stesso valore)
```

Out[398]:

	F3Beck	F3STAIS	F3STAIT	PPO
F3Beck	1.000000	-0.217810	0.286316	NaN
F3STAIS	-0.217810	1.000000	0.286488	NaN
F3STAIT	0.286316	0.286488	1.000000	NaN
PPO	NaN	NaN	NaN	NaN

Correlazione Temozsèpos, Temozsèneg, Temozsèmist, Tnoemozsè con: f2ptsdpart f2ptsdcomp e f3ptsdpart e f3ptsdcomp

In [402]:
```
data14=data[['Temozsèpos','Temozsèneg','Temozsèmist','Tnoemozsè','F2PTSDpart',
             'F2PTSDcomp','F3PTSDpart','F3Tptsd']]
data14.corr()
```

Out[402]:

	Temozsèpos	Temozsèneg	Temozsèmist	Tnoemozsè	F2PTSDpart	F2PTSDcomp	F3PTSDpart	F3Tptsd
Temozsèpos	1.000000	0.127818	0.004269	-0.582931	0.009774	-0.087482	0.021855	0.126152
Temozsèneg	0.127818	1.000000	0.057125	-0.554591	0.051875	0.130767	0.226525	0.310968
Temozsèmist	0.004269	0.057125	1.000000	-0.109421	0.041283	0.048830	-0.123432	-0.178340
Tnoemozsè	-0.582931	-0.554591	-0.109421	1.000000	-0.011350	-0.029836	-0.063036	-0.109892
F2PTSDpart	0.009774	0.051875	0.041283	-0.011350	1.000000	0.133956	-0.013675	0.046622
F2PTSDcomp	-0.087482	0.130767	0.048830	-0.029836	0.133956	1.000000	0.079017	0.135390
F3PTSDpart	0.021855	0.226525	-0.123432	-0.063036	-0.013675	0.079017	1.000000	0.610378
F3Tptsd	0.126152	0.310968	-0.178340	-0.109892	0.046622	0.135390	0.610378	1.000000

Correlazione Temozsèpos, Temozsèneg, Temozsèmist, Tnoemozsè e emozpos, emozneg, emozmiste, noemoz (+ bimbo e bimbomio) con: f2ptsdpart, f2ptsdco
f3ptsdpart, f3ptsdcomp, f2avoid, f2intrusion, f3arousal, f3avoid, f3intrusion, f2beck, f2stais, f2stait, f3beck, f3stais, f3stait.

In [409]:
```
data15=data[['Temozsèpos','Temozsèneg','Temozsèmist','Tnoemozsè','emozpos','emozneg',
             'emozmiste','bimbo','bimbomio','F2PTSDpart','F2PTSDcomp','F2Beck',
             'F2STAIS','F2STAIT','F3PTSDpart','F3Tptsd',
            'F2avoid','F2arousal','F2intrusion','F3avoid','F3arous','F3intrus','F3Beck',
             'F3STAIS','F3STAIT',]]
data15.dropna()
d=data15.corr(); round(d, 3)
```

Out[409]:

	Temozsèpos	Temozsèneg	Temozsèmist	Tnoemozsè	emozpos	emozneg	emozmiste	bimbo	bimbomio	F2PTSDpart	...	F3Tptsd	F2avoid
Temozsèpos	1.000	0.128	0.004	-0.583	0.528	0.074	-0.078	-0.014	0.039	0.010	...	0.126	-0.099
Temozsèneg	0.128	1.000	0.057	-0.555	-0.087	0.659	-0.032	0.071	-0.008	0.052	...	0.311	0.082
Temozsèmist	0.004	0.057	1.000	-0.109	0.086	0.035	0.207	0.085	-0.021	0.041	...	-0.178	0.066
Tnoemozsè	-0.583	-0.555	-0.109	1.000	-0.362	-0.355	0.066	0.005	0.004	-0.011	...	-0.110	0.048
emozpos	0.528	-0.087	0.086	-0.362	1.000	-0.244	-0.049	0.030	0.072	-0.049	...	0.107	-0.042
emozneg	0.074	0.659	0.035	-0.355	-0.244	1.000	-0.048	0.087	0.034	0.107	...	0.016	0.072
emozmiste	-0.078	-0.032	0.207	0.066	-0.049	-0.048	1.000	0.345	-0.014	-0.090	...	NaN	-0.067
bimbo	-0.014	0.071	0.085	0.005	0.030	0.087	0.345	1.000	-0.039	-0.100	...	0.222	-0.065
bimbomio	0.039	-0.008	-0.021	0.004	0.072	0.034	-0.014	-0.039	1.000	0.097	...	-0.177	0.052
F2PTSDpart	0.010	0.052	0.041	-0.011	-0.049	0.107	-0.090	-0.100	0.097	1.000	...	0.047	0.189
F2PTSDcomp	-0.087	0.131	0.049	-0.030	-0.059	0.181	-0.025	-0.059	-0.042	0.134	...	0.135	0.540
F2Beck	-0.042	0.177	0.009	-0.059	0.016	0.079	-0.033	0.009	-0.134	0.168	...	0.281	0.266
F2STAIS	0.054	0.063	0.053	-0.053	0.059	0.065	-0.015	-0.025	0.009	0.090	...	0.019	0.159
F2STAIT	0.007	0.006	0.004	0.034	-0.012	-0.008	-0.003	-0.077	-0.047	0.177	...	0.122	0.188
F3PTSDpart	0.022	0.227	-0.123	-0.063	0.062	-0.109	NaN	0.177	-0.102	-0.014	...	0.610	0.177
F3Tptsd	0.126	0.311	-0.178	-0.110	0.107	0.016	NaN	0.222	-0.177	0.047	...	1.000	0.298
F2avoid	-0.099	0.082	0.066	0.048	-0.042	0.072	-0.067	-0.065	0.052	0.189	...	0.298	1.000
F2arousal	-0.003	0.172	0.039	-0.057	0.000	0.153	-0.064	-0.090	-0.014	0.315	...	0.128	0.393
F2intrusion	0.038	0.106	0.104	-0.060	-0.012	0.110	-0.084	-0.122	0.156	0.593	...	-0.015	0.354
F3avoid	-0.084	-0.080	-0.022	0.149	NaN	NaN	NaN	-0.023	-0.019	0.070	...	0.372	0.266
F3arous	-0.056	0.117	-0.112	0.002	0.061	-0.149	NaN	0.205	-0.092	0.054	...	0.635	0.133
F3intrus	0.079	0.212	-0.057	-0.020	0.105	0.033	NaN	0.191	-0.048	-0.024	...	0.442	0.423
F3Beck	0.061	0.099	-0.149	-0.077	0.163	0.011	NaN	0.192	-0.118	0.139	...	0.528	0.234
F3STAIS	-0.143	0.025	-0.067	-0.005	-0.097	0.019	NaN	-0.106	-0.067	-0.074	...	-0.064	0.060
F3STAIT	0.073	0.251	-0.146	-0.101	0.028	0.224	NaN	-0.128	-0.147	0.033	...	0.174	0.217

25 rows × 25 columns

Per ogni modalità di partoid correlazione con: f2ptsdpart, f2ptsdcomp, f3ptsdpart, f3ptsdcomp, f2avoid, f2intrusion, f3arousal, f3avoid, f3intrusion, f3arousal, f2beck, f2stais, f2stait, f3beck, f3stais, f3stait.

In [420]:
```
# Partoid codificato con valori da 1 a 6

data16=data[['F1PartoId','F2PTSDpart','F2PTSDcomp','F3PTSDpart','F3Tptsd','F2avoid',
             'F2intrusion','F2arousal','F3avoid','F3intrus','F3arous','F2Beck','F2STAIS',
             'F3STAIT','F3Beck','F3STAIS','F3STAIT']]
data16.dropna()
```

Out[420]:

	F1PartoId	F2PTSDpart	F2PTSDcomp	F3PTSDpart	F3Tptsd	F2avoid	F2intrusion	F2arousal	F3avoid	F3intrus	F3arous	F2Beck	F2STAIS	F3STAIT	F
12	3.0	1.0	1.0	0.0	60.0	5.0	3.0	2.0	1.0	1.0	1.0	5.0	49.0	52.0	
21	3.0	1.0	0.0	0.0	26.0	0.0	5.0	0.0	0.0	0.0	0.0	8.0	48.0	46.0	
22	3.0	1.0	0.0	0.0	22.0	8.0	5.0	1.0	0.0	0.0	0.0	1.0	61.0	43.0	
31	2.0	1.0	0.0	0.0	40.0	2.0	2.0	0.0	0.0	0.0	0.0	4.0	42.0	47.0	
32	3.0	1.0	0.0	0.0	31.0	1.0	2.0	1.0	0.0	0.0	0.0	2.0	51.0	48.0	
33	3.0	0.0	0.0	0.0	33.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	43.0	41.0	
36	1.0	1.0	1.0	1.0	28.0	3.0	3.0	4.0	0.0	0.0	1.0	7.0	40.0	48.0	
58	2.0	1.0	0.0	1.0	39.0	4.0	5.0	1.0	0.0	0.0	1.0	6.0	46.0	46.0	
62	1.0	0.0	0.0	1.0	29.0	1.0	0.0	0.0	0.0	1.0	0.0	3.0	49.0	45.0	
70	2.0	0.0	0.0	0.0	35.0	1.0	0.0	0.0	0.0	0.0	0.0	4.0	43.0	49.0	

In [421]:
```
data16 = pd.get_dummies(data16, columns=['F1PartoId'])
round(data16.corr(), 3)
```

Out[421]:

	F2PTSDpart	F2PTSDcomp	F3PTSDpart	F3Tptsd	F2avoid	F2intrusion	F2arousal	F3avoid	F3intrus	F3arous	...	F3STAIT	F3Beck	F3STAIS
F2PTSDpart	1.000	0.134	-0.014	0.047	0.189	0.593	0.315	0.070	-0.024	0.054	...	0.033	0.139	-0.074
F2PTSDcomp	0.134	1.000	0.079	0.135	0.540	0.313	0.559	0.436	0.321	0.105	...	0.246	0.256	-0.007
F3PTSDpart	-0.014	0.079	1.000	0.610	0.177	-0.034	0.229	-0.066	0.349	0.848	...	-0.063	0.260	-0.062
F3Tptsd	0.047	0.135	0.610	1.000	0.298	-0.015	0.128	0.372	0.442	0.635	...	0.174	0.528	-0.064
F2avoid	0.189	0.540	0.177	0.298	1.000	0.354	0.393	0.266	0.423	0.133	...	0.217	0.234	0.060
F2intrusion	0.593	0.313	-0.034	-0.015	0.354	1.000	0.518	0.079	0.171	-0.095	...	0.038	0.026	-0.048
F2arousal	0.315	0.559	0.229	0.128	0.393	0.518	1.000	0.134	0.380	0.061	...	0.188	0.133	-0.021
F3avoid	0.070	0.436	-0.066	0.372	0.266	0.079	0.134	1.000	0.396	0.197	...	0.225	0.421	0.062
F3intrus	-0.024	0.321	0.349	0.442	0.423	0.171	0.380	0.396	1.000	0.064	...	0.247	0.261	0.001
F3arous	0.054	0.105	0.848	0.635	0.133	-0.095	0.061	0.197	0.064	1.000	...	-0.065	0.358	-0.045
F2Beck	0.168	0.293	0.254	0.281	0.266	0.242	0.327	0.021	0.264	0.151	...	0.081	0.490	0.027
F2STAIS	0.090	-0.021	0.027	0.019	0.159	0.155	0.022	0.076	0.087	0.028	...	0.080	0.103	0.016
F3STAIT	0.033	0.246	-0.063	0.174	0.217	0.038	0.188	0.225	0.247	-0.065	...	1.000	0.036	0.236
F3Beck	0.139	0.256	0.260	0.528	0.234	0.026	0.133	0.421	0.261	0.358	...	0.036	1.000	-0.034
F3STAIS	-0.074	-0.007	-0.062	-0.064	0.060	-0.048	-0.021	0.062	0.001	-0.045	...	0.236	-0.034	1.000
F3STAIT	0.033	0.246	-0.063	0.174	0.217	0.038	0.188	0.225	0.247	-0.065	...	1.000	0.036	0.236
F1PartoId_1.0	-0.052	0.029	0.171	0.031	0.083	-0.038	0.010	-0.021	0.196	0.044	...	0.093	-0.031	-0.043
F1PartoId_2.0	-0.045	-0.004	0.027	0.192	0.042	-0.075	-0.042	-0.021	-0.053	0.044	...	0.156	0.074	0.008
F1PartoId_3.0	0.053	0.042	-0.152	0.152	0.127	0.003	-0.014	0.436	0.126	-0.021	...	0.119	0.183	0.033
F1PartoId_4.0	0.006	-0.036	0.179	0.145	-0.010	0.118	0.037	-0.012	-0.030	0.197	...	-0.170	0.162	-0.083
F1PartoId_5.0	0.046	-0.020	NaN	NaN	0.111	-0.017	-0.050	NaN	NaN	NaN	...	NaN	NaN	NaN
F1PartoId_6.0	0.032	-0.014	NaN	NaN	-0.036	0.002	0.158	NaN	NaN	NaN	...	NaN	NaN	NaN

22 rows × 22 columns

Conteggi e statistiche descrittive per: peso medio, apgar1, apgar5 e correlazione con f2ptsd, f3ptsd

In [432]:
```
data17=data[['peso','apgar 1','apgar 5','F2Tptsd','F3Tptsd']]
data17.dropna()
data17.describe()
```

Out[432]:

	peso	apgar 1	apgar 5	F2Tptsd	F3Tptsd
count	350.000000	350.000000	350.000000	346.000000	85.000000
mean	3302.648571	9.048571	9.877143	19.661850	28.858824
std	484.799317	0.710491	0.459591	12.921327	9.175735
min	910.000000	3.000000	5.000000	0.000000	17.000000
25%	3050.000000	9.000000	10.000000	10.000000	22.000000
50%	3330.000000	9.000000	10.000000	17.000000	28.000000
75%	3640.000000	9.000000	10.000000	26.750000	33.000000
max	4400.000000	10.000000	10.000000	71.000000	60.000000

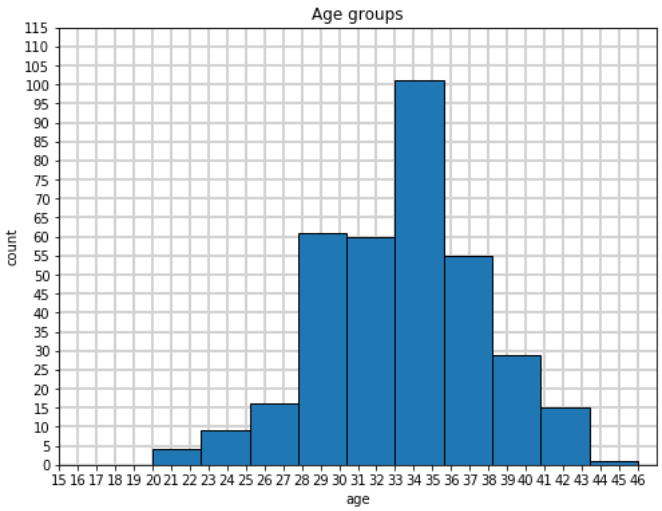
In [435]:
```
round(data17.corr(), 3)
```

Out[435]:

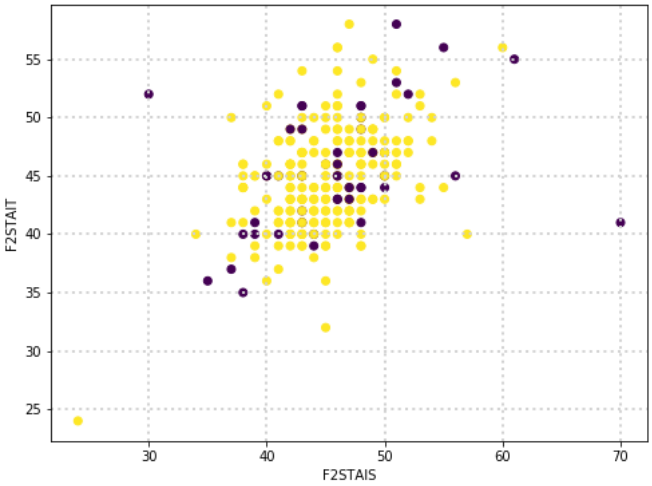
	peso	apgar 1	apgar 5	F2Tptsd	F3Tptsd
peso	1.000	0.109	0.240	0.021	0.084
apgar 1	0.109	1.000	0.597	-0.088	-0.026
apgar 5	0.240	0.597	1.000	-0.146	0.032
F2Tptsd	0.021	-0.088	-0.146	1.000	0.369
F3Tptsd	0.084	-0.026	0.032	0.369	1.000

Istogramma delle età

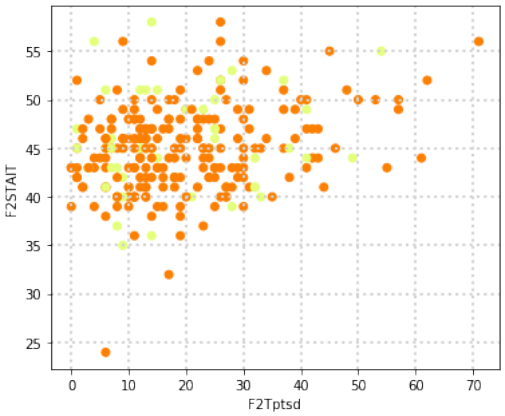
In [564]:
```
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure(figsize=(8,6))
ax = fig.gca()
plt.hist(data['age'], ec='black')
plt.xlabel('age')
plt.ylabel('count')
plt.ylim((0,110))
plt.xlim((18,47))
plt.title('Age groups')
plt.grid(color='lightgrey', linestyle='-', linewidth=2)
ax.set_yticks(np.arange(0, 120, 5))
ax.set_axisbelow(True)
ax.set_xticks(np.arange(15, 47, 1))
plt.show()
```



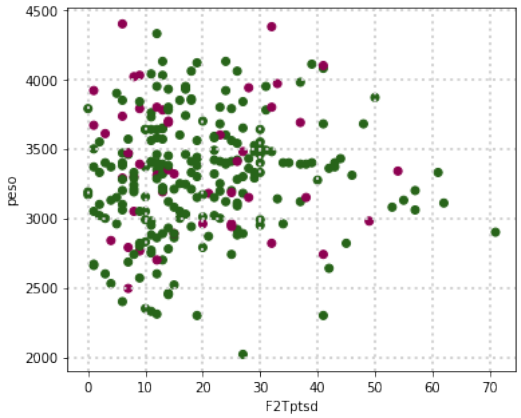
In [548]:
```
fig = plt.figure(figsize=(8,6))
plt.scatter(data11['F2STAIS'], data11['F2STAIT'], c=data11['rischio'])
plt.xlabel('F2STAIS')
plt.ylabel('F2STAIT')     # in viola gli alto rischio, in giallo i basso rischio
plt.grid(color='lightgrey', linestyle=':', linewidth=2)
```



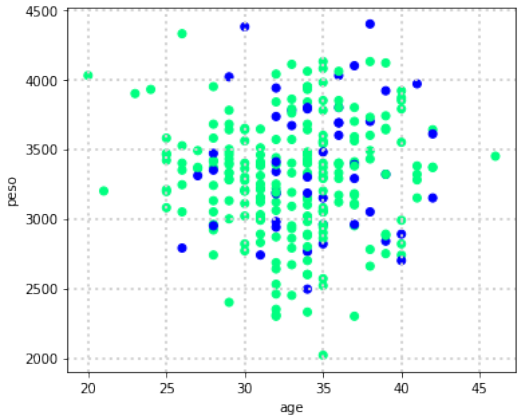
In [550]:
```
fig = plt.figure(figsize=(6,5))
plt.scatter(data11['F2Tptsd'], data11['F2STAIT'], c=data11['rischio'], cmap='Wistia')
plt.xlabel('F2Tptsd')
plt.ylabel('F2STAIT')
plt.grid(color='lightgrey', linestyle=':', linewidth=2)
```



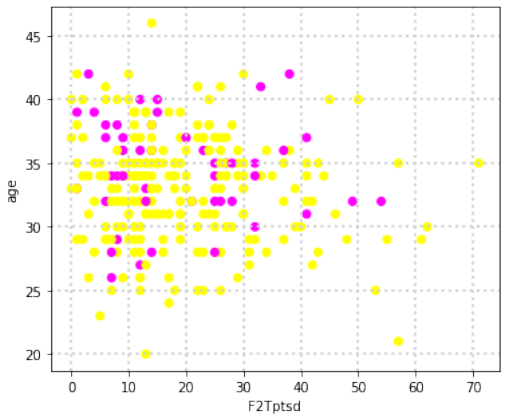
In [551]:
```
fig = plt.figure(figsize=(6,5))
plt.scatter(data11['F2Tptsd'], data11['peso'], c=data11['rischio'], cmap='PiYG')
plt.xlabel('F2Tptsd')
plt.ylabel('peso')
plt.grid(color='lightgrey', linestyle=':', linewidth=2)
```



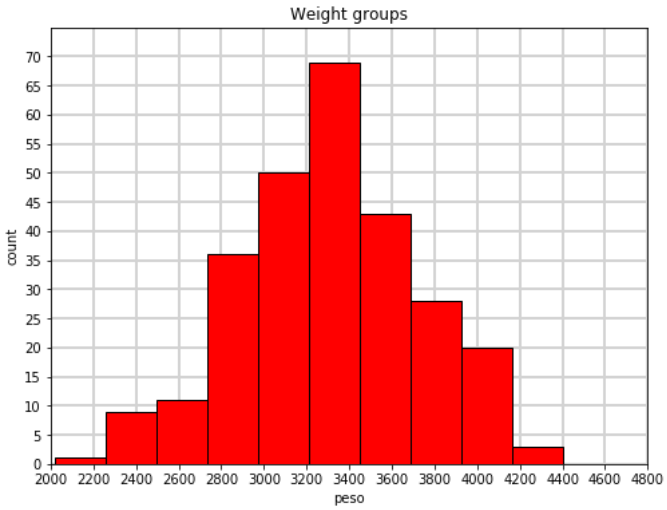
In [552]:
```
fig = plt.figure(figsize=(6,5))
plt.scatter(data11['age'], data11['peso'], c=data11['rischio'], cmap='winter')
plt.xlabel('age')
plt.ylabel('peso')
plt.grid(color='lightgrey', linestyle=':', linewidth=2)
```



In [554]:
```
fig = plt.figure(figsize=(6,5))
plt.scatter(data11['F2Tptsd'], data11['age'], c=data11['rischio'], cmap='spring')
plt.ylabel('age')
plt.xlabel('F2Tptsd')
plt.grid(color='lightgrey', linestyle=':', linewidth=2)
```



In [565]:
```
fig = plt.figure(figsize=(8,6))
ax = fig.gca()
plt.hist(data11['peso'], ec='black', color='red')
plt.xlabel('peso')
plt.ylabel('count')
plt.ylim((0,75))
plt.xlim((2000,4750))
plt.title('Weight groups')
plt.grid(color='lightgrey', linestyle='-', linewidth=2)
ax.set_yticks(np.arange(0, 75, 5))
ax.set_axisbelow(True)
ax.set_xticks(np.arange(2000, 5000, 200))
plt.show()
```



In [616]:
```
from pandas import DataFrame
d=data.groupby('episio').size()
l=[11,58,51,94,81,1]
l2=[1,2,3,4,5,6]
df={'categoria':l, 'conteggio':l2}
df=DataFrame(df, columns= ['categoria','conteggio'])
df
```

Out[616]:

	categoria	conteggio
0	11	1
1	58	2
2	51	3
3	94	4
4	81	5
5	1	6

In [ ]: