Notebook

March 26, 2021

[1]:
```
dataset_name = 'mcswain'
```

[2]:
```
%reload_ext autoreload
%autoreload 2
default_figsize=(14,12)
```

[3]:
```
import datasets
import numpy as np
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['figure.figsize'] = (14, 12)


dataset_module = datasets.datasets_by_name_all[dataset_name]
x,y,metadata = dataset_module.load(dropna=True,verbose=True)
y = datasets.map_y_em(y,dataset_name)

# generate dataframe with both x and y
xy = pd.concat([x,y],axis=1)
xy.describe()
```

```
Warning loading data from McSwain2005-2009_VPHAS-2MASS.csv:
Dropped 2313 rows with missing values.
Rows (original):   5455
Rows (after drop): 3142
```

[3]:

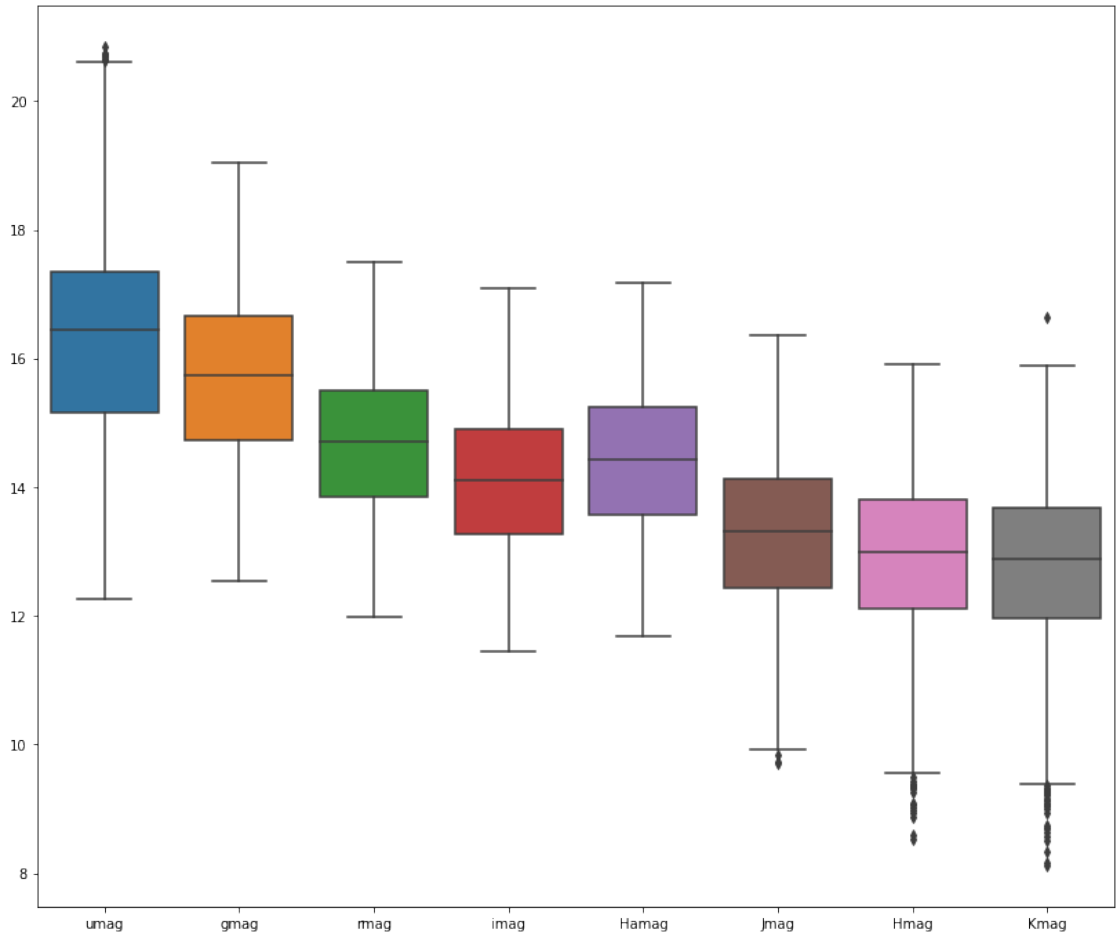
	umag	gmag	rmag	imag	Hamag	\
count	3142.000000	3142.000000	3142.000000	3142.000000	3142.000000	
mean	16.304806	15.668695	14.651512	14.076811	14.398883	
std	1.572447	1.225343	1.047182	1.035455	1.041201	
min	12.260000	12.530000	11.990000	11.450000	11.690000	
25%	15.150000	14.722500	13.842500	13.260000	13.580000	
50%	16.445000	15.740000	14.700000	14.100000	14.435000	
75%	17.337500	16.650000	15.510000	14.900000	15.250000	
max	20.840000	19.050000	17.500000	17.090000	17.170000	

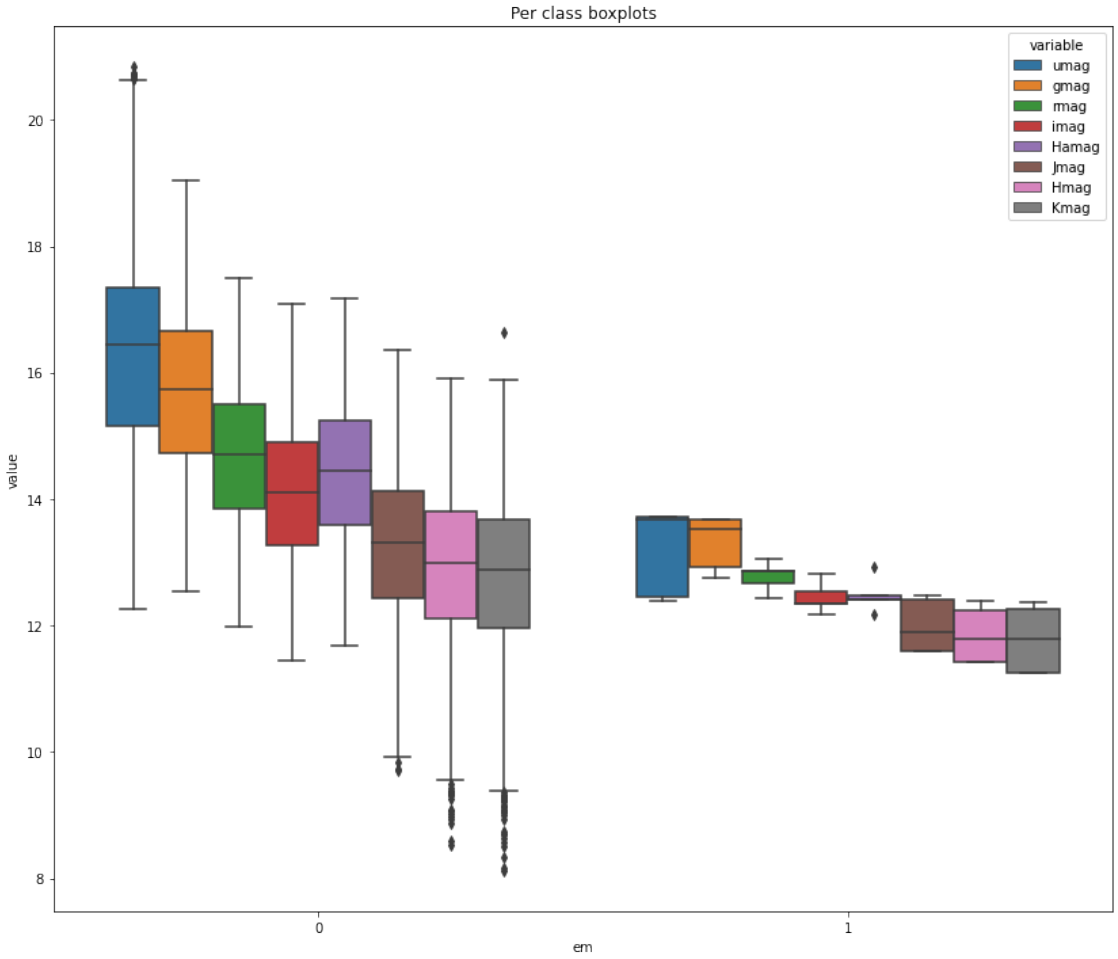
	Jmag	Hmag	Kmag	em
count	3142.000000	3142.000000	3142.000000	3142.000000
mean	13.248104	12.876838	12.739313	0.001591
std	1.128721	1.221364	1.259059	0.039866
min	9.700000	8.521000	8.118000	0.000000
25%	12.434000	12.101000	11.957000	0.000000
50%	13.306000	13.000000	12.876000	0.000000
75%	14.126750	13.801000	13.682000	0.000000
max	16.368000	15.912000	16.631000	1.000000

1 Variable visualization

[4]:
```
sn.boxplot(data=x)

plt.figure()
xy_long = pd.melt(xy, id_vars='em')
sn.boxplot(x='em', y='value', hue='variable', data=xy_long)
plt.title("Per class boxplots")
```

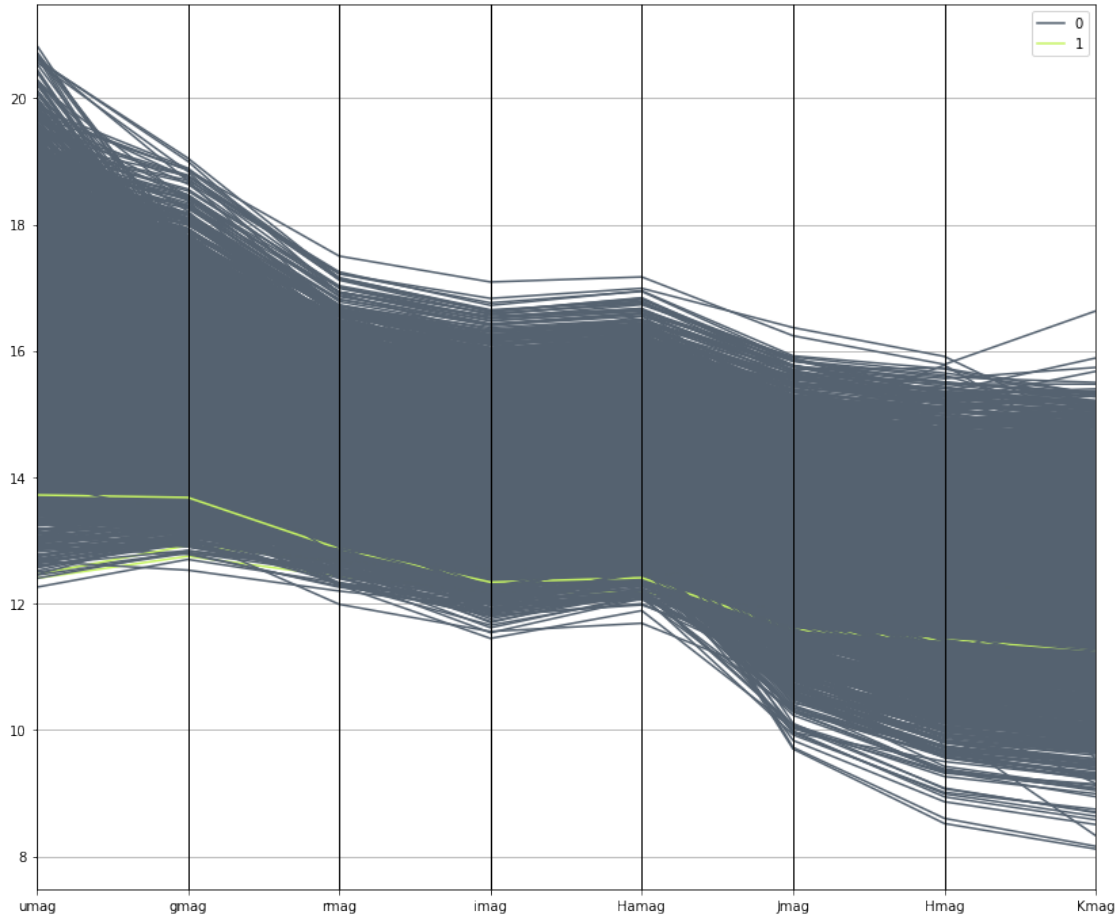
[4]:
```
Text(0.5, 1.0, 'Per class boxplots')
```


20
18
16
14
12
10
8
umag
gmag
rmag
imag
Hamag
Jmag
Hmag
Kmag

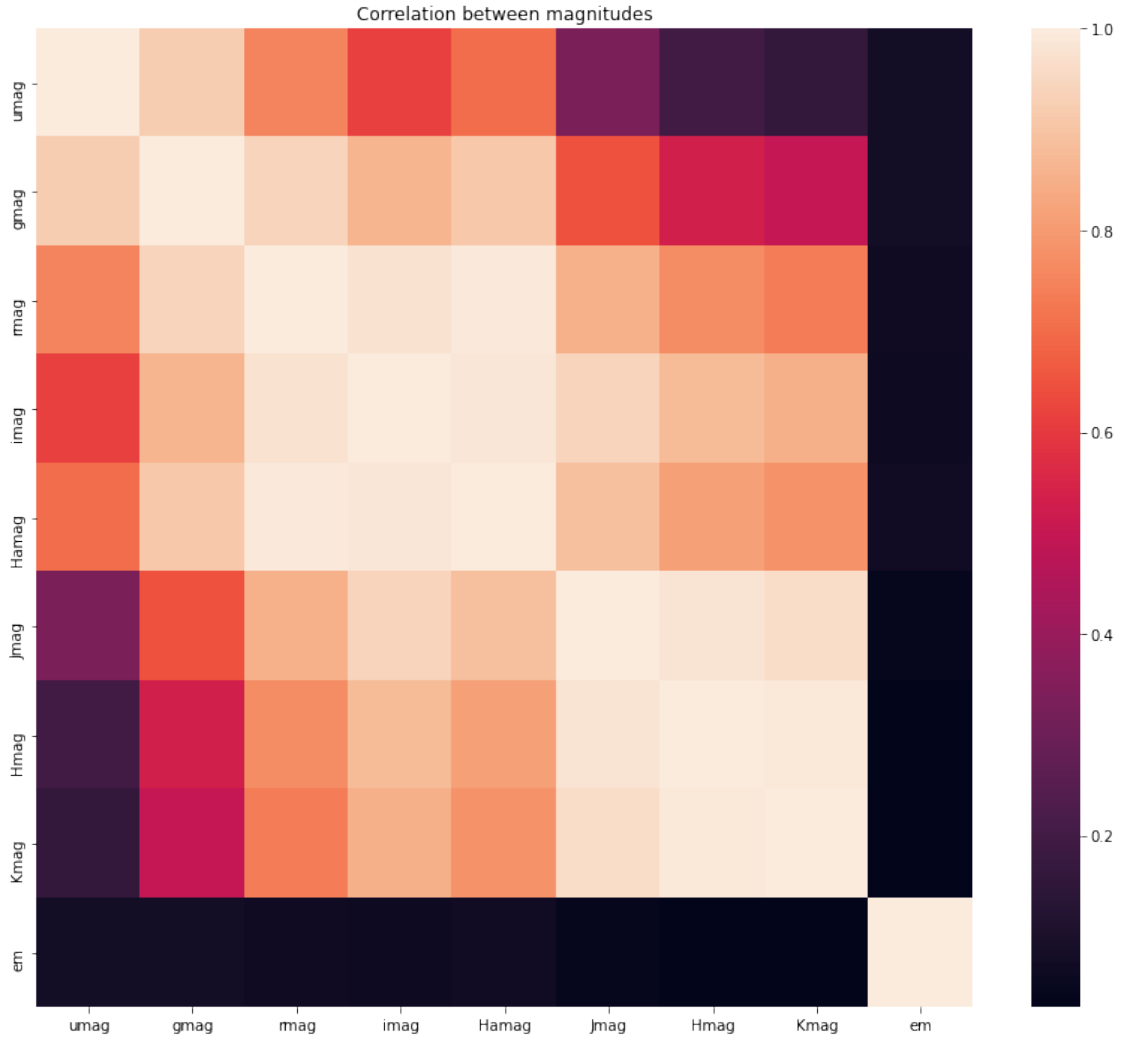


[5]:
```
pd.plotting.parallel_coordinates(xy,"em",color=('#556270','#C7F464'))
```

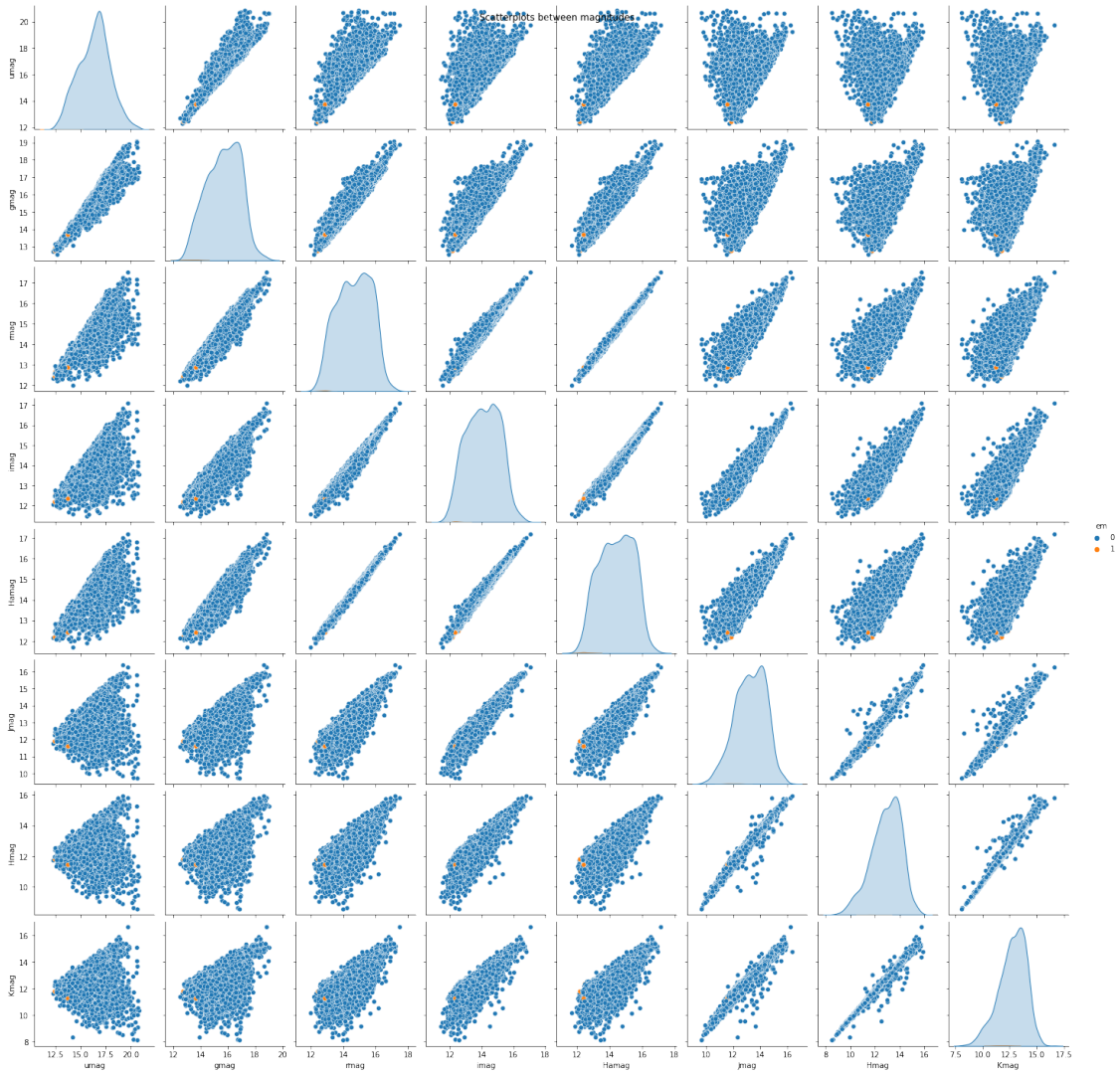
[5]: <AxesSubplot:>



[6]:
```
sn.heatmap(xy.corr().abs())
plt.title("Correlation between magnitudes")
plt.show()

sn.pairplot(xy,hue="em")
plt.suptitle("Scatterplots between magnitudes")
# axes=pd.plotting.scatter_matrix(x,c=y["em"],alpha=0.
 ↪9,grid=False,figsize=(14,12))
```



[6]: Text(0.5, 0.98, 'Scatterplots between magnitudes')



2 Outlier detection via confidence interval

[7]:
```
from scipy import stats
m = len(x.columns) # number of columns = number of hypothesis
confidence= 0.99
adjusted_confidence = 1- (1-confidence)/m  # bonferroni-adjusted confidence
max_zscore = stats.norm.ppf(adjusted_confidence)
print(f"Confidence  (desired): {confidence}")
print(f"Confidence (adjusted): {adjusted_confidence}")
print(f"Z-score    (adjusted): {max_zscore}")

indices = (np.abs(stats.zscore(x-x.mean())) > max_zscore).any(axis=1)
outliers_x = x[indices]
```

```
if dataset_name != "all_em":
    outliers_metadata = metadata[indices]
    outliers_x = pd.concat([outliers_x,outliers_metadata],axis=1)
outliers_x
```

```
Confidence  (desired): 0.99
Confidence (adjusted): 0.99875
Z-score    (adjusted): 3.023341439739154
```

[7]:

	umag	gmag	rmag	imag	Hamag	Jmag	Hmag	Kmag	WHa06	e_b-y	\
240	19.72	18.86	17.50	17.09	17.17	16.239	15.786	16.631	NaN	0.118	
1272	16.87	14.44	12.51	11.62	12.13	9.924	9.015	8.747	NaN	0.080	
1462	19.19	15.99	13.72	12.08	13.14	9.836	8.866	8.505	NaN	0.081	
1994	18.69	15.60	13.52	12.09	13.01	10.053	9.082	8.692	NaN	0.042	
2163	20.27	16.75	14.24	12.35	13.66	9.725	8.603	8.161	NaN	0.064	
2483	14.21	13.95	13.34	12.93	13.15	12.348	9.971	8.329	NaN	0.084	
2683	18.88	15.78	13.35	12.23	12.90	9.957	8.990	8.636	NaN	0.105	
2718	20.66	16.88	14.00	12.59	13.46	9.700	8.521	8.118	NaN	0.111	
3015	20.29	16.89	14.10	12.64	13.60	10.106	9.070	8.703	NaN	0.040	
3028	20.14	16.66	14.01	12.48	13.43	9.984	8.943	8.580	NaN	0.090	

	…	vsini	e_umag	y-Ha	WHa07	e_Teff	e_ymag	e_Hmag	Rad	e_Hamag	\
240	…	NaN	0.02	0.512	NaN	NaN	0.087	0.126	NaN	0.02	
1272	…	NaN	0.01	1.286	NaN	NaN	0.051	0.022	NaN	0.00	
1462	…	NaN	0.05	1.607	NaN	NaN	0.051	0.042	NaN	0.00	
1994	…	NaN	0.03	1.655	NaN	NaN	0.024	0.025	NaN	0.00	
2163	…	NaN	0.08	2.015	NaN	NaN	0.034	0.018	NaN	0.00	
2483	…	NaN	0.00	0.423	NaN	NaN	0.064	NaN	NaN	0.00	
2683	…	NaN	0.04	1.575	NaN	NaN	0.054	0.029	NaN	0.00	
2718	…	NaN	0.13	1.660	NaN	NaN	0.071	0.034	NaN	0.00	
3015	…	NaN	0.11	0.588	NaN	NaN	0.027	0.023	NaN	0.00	
3028	…	NaN	0.07	1.176	NaN	NaN	0.057	0.023	NaN	0.00	

	ymag
240	17.701
1272	13.404
1462	14.629
1994	14.603
2163	15.619
2483	13.641
2683	14.621
2718	15.233
3015	12.547
3028	15.855

[10 rows x 43 columns]

3 Outlier detection via IQR

[8]:
```
iqr_factor=1.5
q25,q75=x.quantile(0.25),x.quantile(0.75)
iqr=q75-q25
min_values = q25-iqr_factor*iqr
max_values = q75+iqr_factor*iqr
# ou
indices = (np.logical_or(x<min_values,x>max_values)).any(axis=1)
outliers_x = x[indices]
if dataset_name != "all_em":
    outliers_metadata = metadata[indices]
    outliers_x = pd.concat([outliers_x,outliers_metadata],axis=1)
outliers_x
```

[8]:

	umag	gmag	rmag	imag	Hamag	Jmag	Hmag	Kmag	WHa06	e_b-y	\
165	19.95	16.73	14.29	12.98	13.77	10.709	9.637	9.278	NaN	0.070	
235	18.90	16.00	13.75	12.54	13.20	10.597	9.569	9.236	NaN	0.054	
240	19.72	18.86	17.50	17.09	17.17	16.239	15.786	16.631	NaN	0.118	
256	20.66	18.69	16.47	15.72	16.01	14.595	13.883	13.670	NaN	0.103	
470	20.62	19.00	16.91	16.25	16.49	15.197	14.309	14.071	NaN	0.112	
906	20.11	16.77	14.16	12.97	13.82	10.665	9.660	9.270	NaN	0.083	
1272	16.87	14.44	12.51	11.62	12.13	9.924	9.015	8.747	NaN	0.080	
1345	18.19	15.56	13.42	12.18	12.96	10.290	9.387	9.042	NaN	0.079	
1395	19.78	16.67	14.23	12.84	13.77	10.572	9.561	9.217	NaN	0.095	
1405	19.27	16.12	13.86	12.60	13.40	10.574	9.658	9.369	NaN	0.090	
1462	19.19	15.99	13.72	12.08	13.14	9.836	8.866	8.505	NaN	0.081	
1533	20.84	17.27	14.97	13.54	14.47	11.570	10.644	10.342	NaN	0.108	
1539	20.73	17.59	15.46	14.16	15.01	12.216	11.319	11.070	NaN	0.119	
1570	16.64	14.45	12.64	11.76	12.22	10.061	9.265	8.993	NaN	0.076	
1577	15.15	14.38	12.88	12.04	12.51	10.323	9.570	9.301	NaN	0.064	
1836	20.73	17.62	15.27	13.97	14.77	11.809	10.792	10.536	NaN	0.106	
1967	20.71	16.97	14.51	12.96	13.92	10.553	9.426	9.080	NaN	0.064	
1994	18.69	15.60	13.52	12.09	13.01	10.053	9.082	8.692	NaN	0.042	
2163	20.27	16.75	14.24	12.35	13.66	9.725	8.603	8.161	NaN	0.064	
2415	15.63	13.97	12.29	11.45	11.89	10.025	9.318	9.145	NaN	0.046	
2437	17.55	14.91	12.91	11.91	12.47	10.235	9.362	9.122	NaN	0.067	
2463	16.69	14.61	12.67	11.70	12.25	10.084	9.333	9.075	NaN	0.060	
2483	14.21	13.95	13.34	12.93	13.15	12.348	9.971	8.329	NaN	0.084	
2661	18.26	15.52	13.27	12.22	12.79	10.289	9.340	9.070	NaN	0.070	
2674	16.20	14.57	12.89	11.99	12.51	10.388	9.647	9.247	NaN	0.106	
2678	15.84	14.97	13.10	11.84	12.66	9.967	9.502	9.238	NaN	0.064	
2683	18.88	15.78	13.35	12.23	12.90	9.957	8.990	8.636	NaN	0.105	
2699	16.36	16.05	15.11	14.53	14.80	13.671	10.260	9.155	NaN	0.079	
2713	16.57	14.70	12.85	12.04	12.47	10.359	9.581	9.304	NaN	0.051	
2718	20.66	16.88	14.00	12.59	13.46	9.700	8.521	8.118	NaN	0.111	
2750	18.60	15.82	13.67	12.50	13.21	10.497	9.607	9.329	NaN	0.074	

```
3015  20.29  16.89  14.10  12.64  13.60  10.106   9.070   8.703    NaN  0.040
3027  20.69  17.51  14.94  13.62  14.48  11.284  10.356  10.017    NaN  0.062
3028  20.14  16.66  14.01  12.48  13.43   9.984   8.943   8.580    NaN  0.090
3102  20.42  16.94  14.25  12.54  13.67  10.398   9.341   8.945    NaN  0.120

      …  vsini  e_umag   y-Ha  WHa07  e_Teff  e_ymag  e_Hmag  Rad  e_Hamag  \
165   …    NaN    0.08  1.544    NaN     NaN   0.038   0.022  NaN     0.00
235   …    NaN    0.03  1.464    NaN     NaN   0.031   0.024  NaN     0.00
240   …    NaN    0.02  0.512    NaN     NaN   0.087   0.126  NaN     0.02
256   …    NaN    0.04  1.169    NaN     NaN   0.070   0.022  NaN     0.01
470   …    NaN    0.05  1.040    NaN     NaN   0.079     NaN  NaN     0.01
906   …    NaN    0.08  1.634    NaN     NaN   0.050   0.022  NaN     0.00
1272  …    NaN    0.01  1.286    NaN     NaN   0.051   0.022  NaN     0.00
1345  …    NaN    0.03  1.537    NaN     NaN   0.050   0.022  NaN     0.00
1395  …    NaN    0.08  1.616    NaN     NaN   0.058   0.027  NaN     0.01
1405  …    NaN    0.05  1.628    NaN     NaN   0.056   0.021  NaN     0.00
1462  …    NaN    0.05  1.607    NaN     NaN   0.051   0.042  NaN     0.00
1533  …    NaN    0.15  1.662    NaN     NaN   0.066   0.026  NaN     0.01
1539  …    NaN    0.13  1.567    NaN     NaN   0.070   0.023  NaN     0.01
1570  …    NaN    0.01  1.408    NaN     NaN   0.055   0.023  NaN     0.00
1577  …    NaN    0.00  1.158    NaN     NaN   0.047   0.028  NaN     0.00
1836  …    NaN    0.13  1.689    NaN     NaN   0.061   0.021  NaN     0.01
1967  …    NaN    0.12  1.873    NaN     NaN   0.035   0.022  NaN     0.00
1994  …    NaN    0.03  1.655    NaN     NaN   0.024   0.025  NaN     0.00
2163  …    NaN    0.08  2.015    NaN     NaN   0.034   0.018  NaN     0.00
2415  …    NaN    0.01  1.150    NaN     NaN   0.027   0.024  NaN     0.00
2437  …    NaN    0.01  1.392    NaN     NaN   0.039   0.024  NaN     0.00
2463  …    NaN    0.01  1.344    NaN     NaN   0.035   0.026  NaN     0.00
2483  …    NaN    0.00  0.423    NaN     NaN   0.064     NaN  NaN     0.00
2661  …    NaN    0.02  1.523    NaN     NaN   0.040   0.023  NaN     0.00
2674  …    NaN    0.01  1.243    NaN     NaN   0.065   0.030  NaN     0.00
2678  …    NaN    0.01  1.375    NaN     NaN   0.040   0.024  NaN     0.00
2683  …    NaN    0.04  1.575    NaN     NaN   0.054   0.029  NaN     0.00
2699  …    NaN    0.01  0.699    NaN     NaN   0.054     NaN  NaN     0.01
2713  …    NaN    0.01  1.210    NaN     NaN   0.029   0.026  NaN     0.00
2718  …    NaN    0.13  1.660    NaN     NaN   0.071   0.034  NaN     0.00
2750  …    NaN    0.03  1.501    NaN     NaN   0.040   0.025  NaN     0.00
3015  …    NaN    0.11  0.588    NaN     NaN   0.027   0.023  NaN     0.00
3027  …    NaN    0.13  0.785    NaN     NaN   0.042   0.024  NaN     0.00
3028  …    NaN    0.07  1.176    NaN     NaN   0.057   0.023  NaN     0.00
3102  …    NaN    0.10  1.935    NaN     NaN   0.069   0.022  NaN     0.01

        ymag
165   15.265
235   14.757
240   17.701
256   17.186
```

```
470   17.462
906   15.421
1272  13.404
1345  14.437
1395  15.182
1405  14.998
1462  14.629
1533  16.097
1539  16.327
1570  13.644
1577  13.745
1836  16.493
1967  15.812
1994  14.603
2163  15.619
2415  13.117
2437  13.932
2463  13.656
2483  13.641
2661  14.378
2674  13.702
2678  14.052
2683  14.621
2699  15.519
2713  13.829
2718  15.233
2750  14.762
3015  12.547
3027  15.072
3028  15.855
3102  15.703

[35 rows x 43 columns]
```

4 Analysis of q-features (q_3) (all magnitudes)

[9]:
```
x_np=x.to_numpy()
import qfeatures
coefficients = dataset_module.coefficients
systems = dataset_module.systems
coefficients_np = np.array([coefficients[k] for k in x.columns])
systems = [systems[k] for k in x.columns]
q=qfeatures.calculate(x_np,coefficients_np,x.columns,systems,combination_size=3)
m = q.magnitudes

q_df = pd.DataFrame(m, columns = q.column_names)
```

```
q_df.describe()
```

[9]:

	umag_gmag_rmag	umag_gmag_imag	umag_gmag_Hamag	umag_gmag_Jmag
count	3142.000000	3142.000000	3142.000000	3142.000000
mean	0.156141	-0.378599	-0.010663	-3.028395
std	0.477268	0.333868	0.437317	0.940956
min	-0.759654	-1.321813	-0.903692	-7.115069
25%	-0.126526	-0.615351	-0.282897	-3.564993
50%	0.016104	-0.469503	-0.131121	-2.951368
75%	0.231807	-0.232851	0.094217	-2.415524
max	2.579221	1.192398	2.186495	-0.466389

	umag_gmag_Hmag	umag_gmag_Kmag	umag_rmag_imag	umag_rmag_Hamag
count	3142.000000	3142.000000	3142.000000	3142.000000
mean	-5.979377	-9.798617	0.954243	1.407748
std	2.229224	3.864487	0.827491	0.977587
min	-16.027196	-27.431046	-0.711111	-0.596636
25%	-7.162250	-11.795907	0.425526	0.813995
50%	-5.624293	-9.041667	0.664269	1.106449
75%	-4.540071	-7.278962	1.343070	1.737780
max	-0.952413	-1.686405	4.944912	6.135140

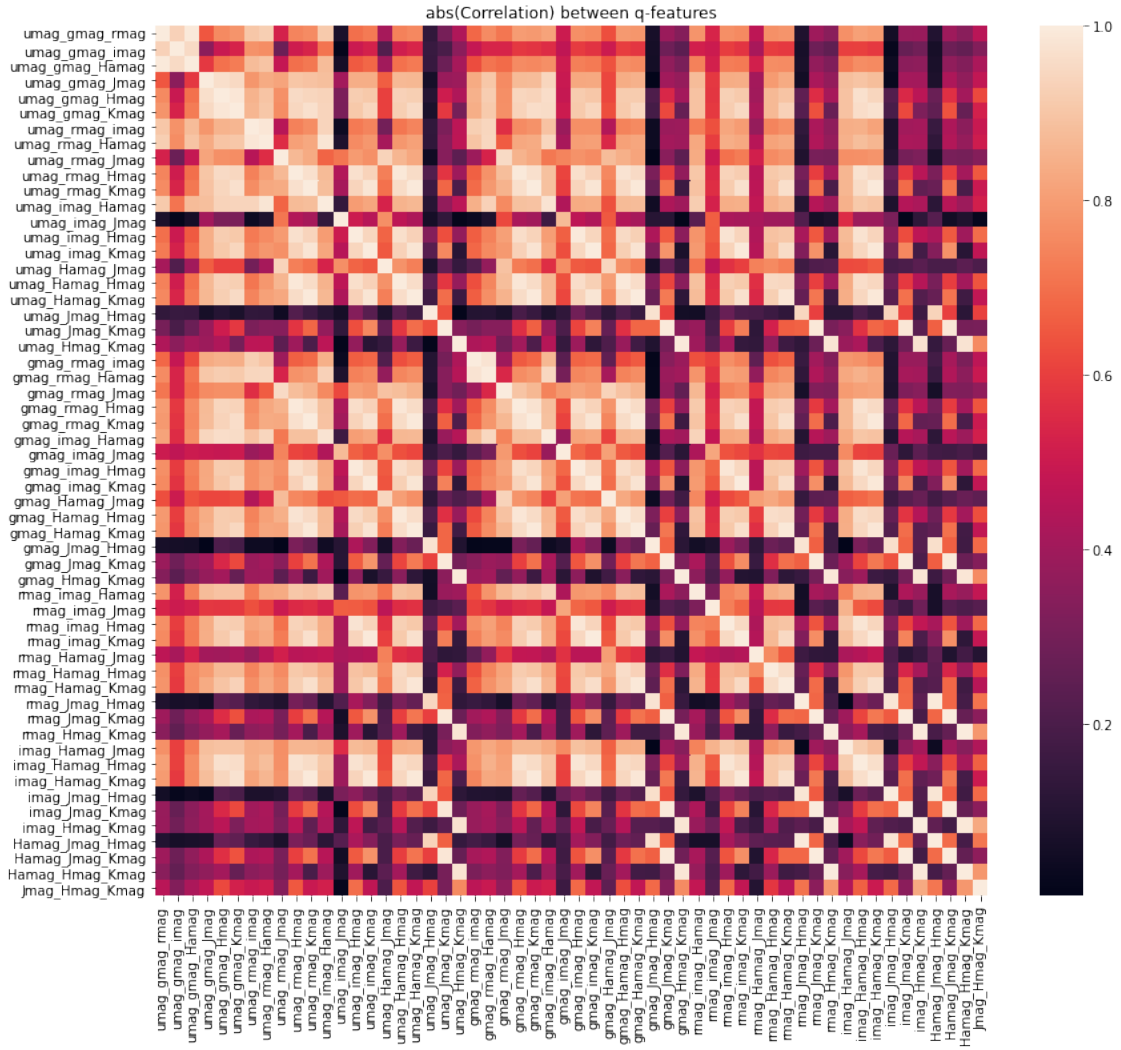
	umag_rmag_Jmag	umag_rmag_Hmag	…	imag_Hamag_Jmag	imag_Hamag_Hmag
count	3142.000000	3142.000000	…	3142.000000	3142.000000
mean	-2.400995	-6.371318	…	0.365199	1.100709
std	0.870262	2.649514	…	0.184691	0.539457
min	-7.689333	-21.841478	…	-0.028903	0.046913
25%	-2.914667	-7.681826	…	0.241431	0.755772
50%	-2.415333	-5.949609	…	0.319889	0.973728
75%	-1.835278	-4.581130	…	0.463094	1.322918
max	0.383778	-0.032348	…	1.375556	4.200283

	imag_Hamag_Kmag	imag_Jmag_Hmag	imag_Jmag_Kmag	imag_Hmag_Kmag
count	3142.000000	3142.000000	3142.000000	3142.000000
mean	2.010004	0.029678	-0.817382	0.638187
std	0.976299	0.406704	0.884542	0.772481
min	0.031895	-6.482065	-13.751588	-11.218902
25%	1.366315	-0.095897	-1.125529	0.350629
50%	1.780310	0.037130	-0.677735	0.625464
75%	2.436038	0.174163	-0.364721	0.986067
max	7.662516	4.936783	4.142765	7.507065

	Hamag_Jmag_Hmag	Hamag_Jmag_Kmag	Hamag_Hmag_Kmag	Jmag_Hmag_Kmag
count	3142.000000	3142.000000	3142.000000	3142.000000
mean	0.004697	-1.210278	0.767005	0.254415
std	0.581073	1.279983	1.001007	0.264408
min	-9.400609	-19.827601	-15.024588	-2.622131

```
25%          -0.192011      -1.668518       0.427647       0.119691
50%           0.027500      -0.991654       0.778657       0.231033
75%           0.236761      -0.543185       1.206074       0.380542
max           6.374304       5.383026       9.521608       2.715843

[8 rows x 56 columns]
```

[10]:
```
sn.heatmap(q_df.corr().abs())
plt.title("abs(Correlation) between q-features")
plt.show()
```



5 Analysis of q-features (q_4) (calculated by system to avoid combinatory explosion)

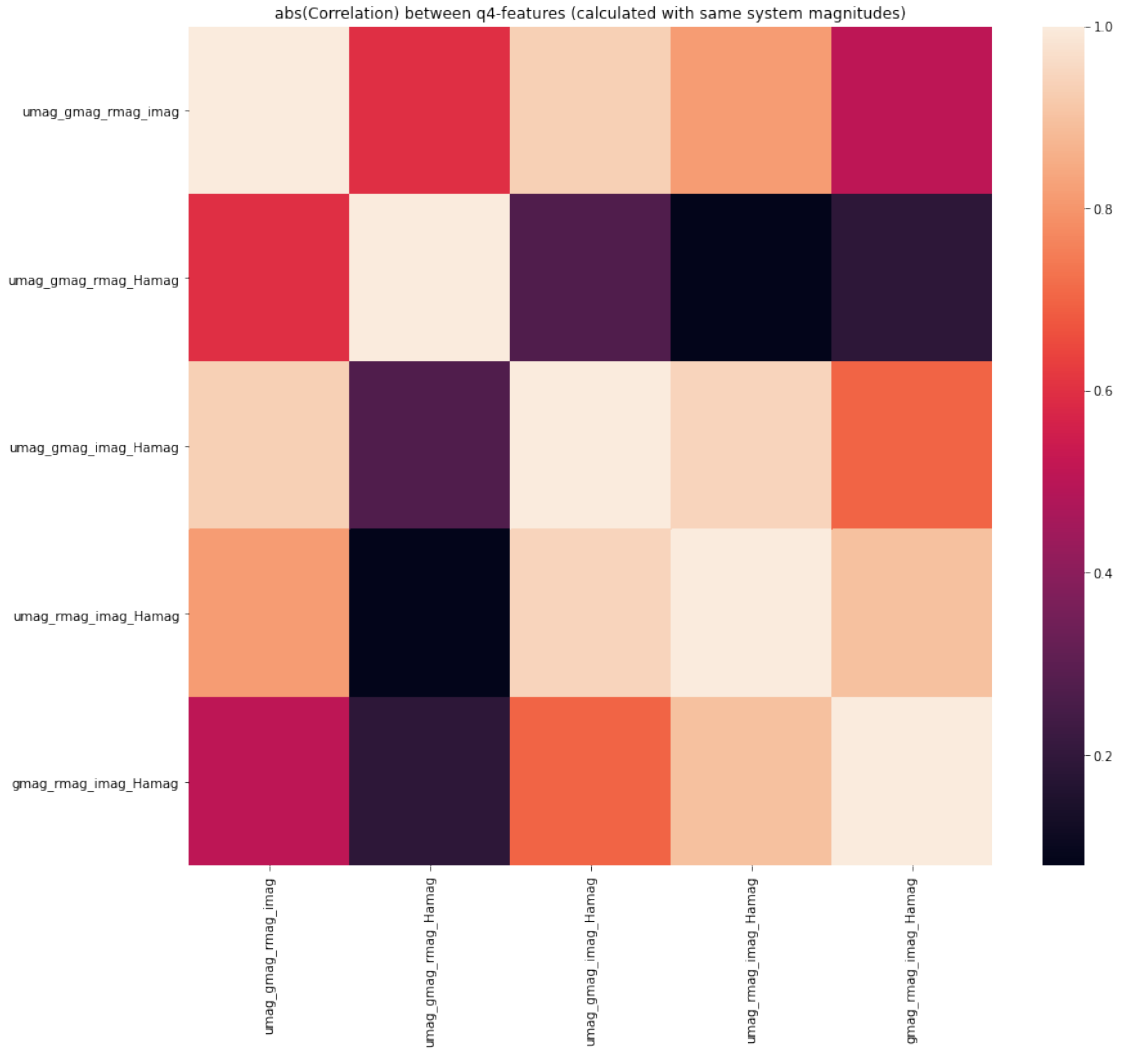
[11]:
```
x_np=x.to_numpy()
import qfeatures
coefficients = dataset_module.coefficients
systems = dataset_module.systems
coefficients_np = np.array([coefficients[k] for k in x.columns])
systems = [systems[k] for k in x.columns]
q= qfeatures.calculate(x_np,coefficients_np,x.
 ↪columns,systems,combination_size=4,by_system=True)

m = q.magnitudes

q_df = pd.DataFrame(m, columns = q.column_names)
q_df.describe()
```

[11]:

	umag_gmag_rmag_imag	umag_gmag_rmag_Hamag	umag_gmag_imag_Hamag	\
count	3142.000000	3142.000000	3142.000000	
mean	-0.407929	-0.983686	-0.180304	
std	0.381832	0.505124	0.442765	
min	-1.514333	-3.421176	-1.336047	
25%	-0.685500	-1.351176	-0.497384	
50%	-0.483167	-1.013824	-0.279884	
75%	-0.222167	-0.636618	0.065000	
max	1.218500	1.160000	1.574651	

	umag_rmag_imag_Hamag	gmag_rmag_imag_Hamag
count	3142.000000	3142.000000
mean	0.095365	0.275669
std	0.718001	0.334845
min	-1.405116	-0.506047
25%	-0.429535	0.050465
50%	-0.058256	0.216279
75%	0.452616	0.421221
max	2.936047	1.621628

[12]:
```
sn.heatmap(q_df.corr().abs())
_=plt.title("abs(Correlation) between q4-features (calculated with same system␣
 ↪magnitudes)")
```



[13]:
```
q_dfy=pd.concat([q_df,y],axis=1)
sn.pairplot(q_dfy,hue="em")
_=plt.suptitle("Scatter plots between q4-features (calculated with same system␣
 ↪magnitudes)")
```


Scatter plots between q4 features (calculated with same system magnitudes)
umag_gmag_rmag_imag
umag_gmag_rmag_Hamag
umag_gmag_imag_Hamag
umag_rmag_imag_Hamag
gmag_rmag_imag_Hamag
em
0
1