kidney_diseade_EDA (2)

September 20, 2019

[ ]:

1 Kidney disease dataset EDA

2 columns and attributes info

```
age     -   age
bp      -   blood pressure
sg      -   specific gravity
al      -       albumin
su      -   sugar
rbc     -   red blood cells
pc      -   pus cell
pcc     -   pus cell clumps
ba      -   bacteria
bgr     -   blood glucose random
bu      -   blood urea
sc      -   serum creatinine
sod     -   sodium
pot     -   potassium
hemo        -   hemoglobin
pcv     -   packed cell volume
wc      -   white blood cell count
rc      -   red blood cell count
htn     -   hypertension
dm      -   diabetes mellitus
cad     -   coronary artery disease
appet       -   appetite
pe      -   pedal edema
ane     -   anemia
class       -   class
```

- 4.Number of Instances: 400 (250 CKD, 150 notckd) %
- 5.Number of Attributes: 24 + class = 25 (11 numeric ,14 nominal) %

- 6.Attribute Information : >1.Age(numerical) age in years 2.Blood Pressure(numerical) bp in mm/Hg 3.Specific Gravity(nominal) sg - (1.005,1.010,1.015,1.020,1.025) 4.Albumin(nominal) al - (0,1,2,3,4,5) 5.Sugar(nominal) su - (0,1,2,3,4,5) 6.Red Blood Cells(nominal) rbc - (normal,abnormal) 7.Pus Cell (nominal) pc - (normal,abnormal) 8.Pus Cell clumps(nominal) pcc - (present,notpresent) 9.Bacteria(nominal) ba - (present,notpresent) 10.Blood Glucose Random(numerical)
  bgr in mgs/dl 11.Blood Urea(numerical)
  bu in mgs/dl 12.Serum Creatinine(numerical)
  sc in mgs/dl 13.Sodium(numerical) sod in mEq/L 14.Potassium(numerical) pot in mEq/L 15.Hemoglobin(numerical) hemo in gms 16.Packed Cell Volume(numerical) 17.White Blood Cell Count(numerical) wc in cells/cumm 18.Red Blood Cell Count(numerical)
  rc in millions/cmm 19.Hypertension(nominal)
  htn - (yes,no) 20.Diabetes Mellitus(nominal)
  dm - (yes,no) 21.Coronary Artery Disease(nominal) cad - (yes,no) 22.Appetite(nominal)
  appet - (good,poor) 23.Pedal Edema(nominal) pe - (yes,no)
  24.Anemia(nominal) ane - (yes,no) 25.Class (nominal)
  class - (ckd,notckd)

- 7. Missing Attribute Values: Yes(Denoted by "?") %

- 8. Class Distribution: (2 classes) Class Number of instances ckd 250 notckd 150

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import math
```

[ ]:
```
sns.set_style(style='whitegrid')
```

[9]:
```
kidney.head(8)
```

[9]:
```
   id   age    bp     sg   al   su  ...   dm cad appet   pe  ane classification
0   0  48.0  80.0  1.020  1.0  0.0  ...  yes  no  good   no   no            ckd
1   1   7.0  50.0  1.020  4.0  0.0  ...   no  no  good   no   no            ckd
2   2  62.0  80.0  1.010  2.0  3.0  ...  yes  no  poor   no  yes            ckd
3   3  48.0  70.0  1.005  4.0  0.0  ...   no  no  poor  yes  yes            ckd
4   4  51.0  80.0  1.010  2.0  0.0  ...   no  no  good   no   no            ckd
5   5  60.0  90.0  1.015  3.0  0.0  ...  yes  no  good  yes   no            ckd
6   6  68.0  70.0  1.010  0.0  0.0  ...   no  no  good   no   no            ckd
7   7  24.0   NaN  1.015  2.0  4.0  ...  yes  no  good  yes   no            ckd

[8 rows x 26 columns]
```

[3]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id
=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redire
```

```
ct_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&scope=email%20https%3A%2F%2Fwww.googl
eapis.com%2Fauth%2Fdocs.test%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive%2
0https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%20https%3A%2F%2
Fwww.googleapis.com%2Fauth%2Fpeopleapi.readonly&response_type=code

Enter your authorization code:
ûûûûûûûûûû
Mounted at /content/drive
```

[ ]:

```
kidney=pd.read_csv('/content/drive/My Drive/CSV files/kidney_disease.csv')
```

[258]:

```
kidney.dtypes
```

[258]:

```
id                  int64
age               float64
bp                float64
sg                float64
al                float64
su                float64
rbc                object
pc                 object
pcc                object
ba                 object
bgr               float64
bu                float64
sc                float64
sod               float64
pot               float64
hemo              float64
pcv               float64
wc                float64
rc                float64
htn                object
dm                 object
cad                object
appet              object
pe                 object
ane                object
classification     object
dtype: object
```

[ ]:

```
def clas(x):
  if x=='ckd\t':
    return 'ckd'
  else:
    return x
```

[ ]:

```
kidney.classification=kidney.classification.apply(clas,convert_dtype=True)
```

[259]:

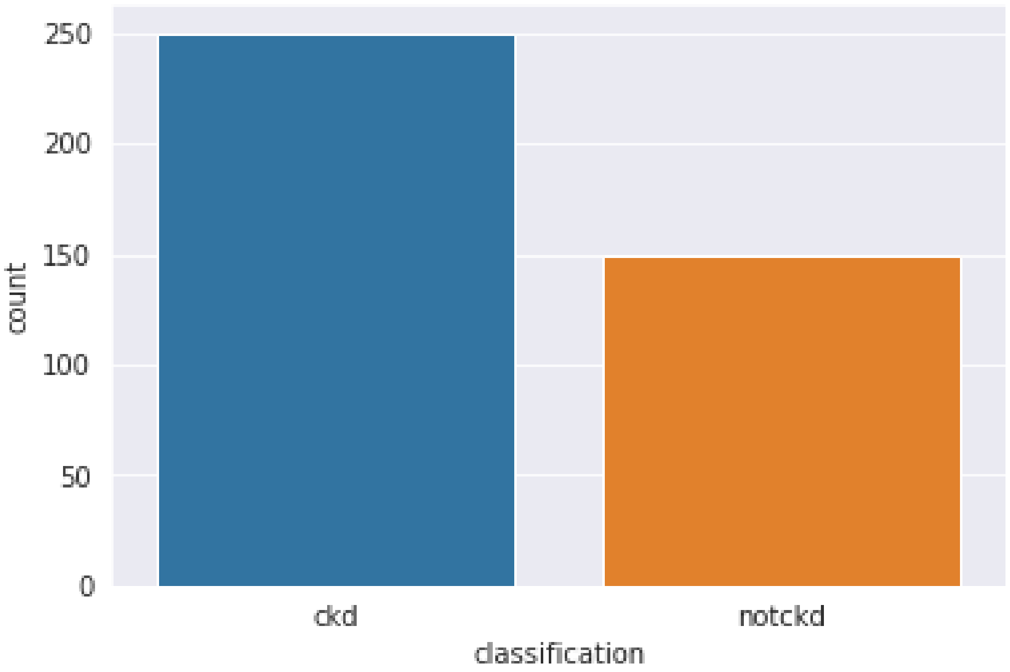
```
kidney.classification.unique()
```

[259]:
```
array(['ckd', 'notckd'], dtype=object)
```

[ ]:
```
kidney.wc=pd.to_numeric(kidney.wc,errors='coerce')
kidney.rc=pd.to_numeric(kidney.rc,errors='coerce')
```

[158]:
```
sns.countplot(x='classification',data=kidney)
```

[158]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbd7c18c160>
```

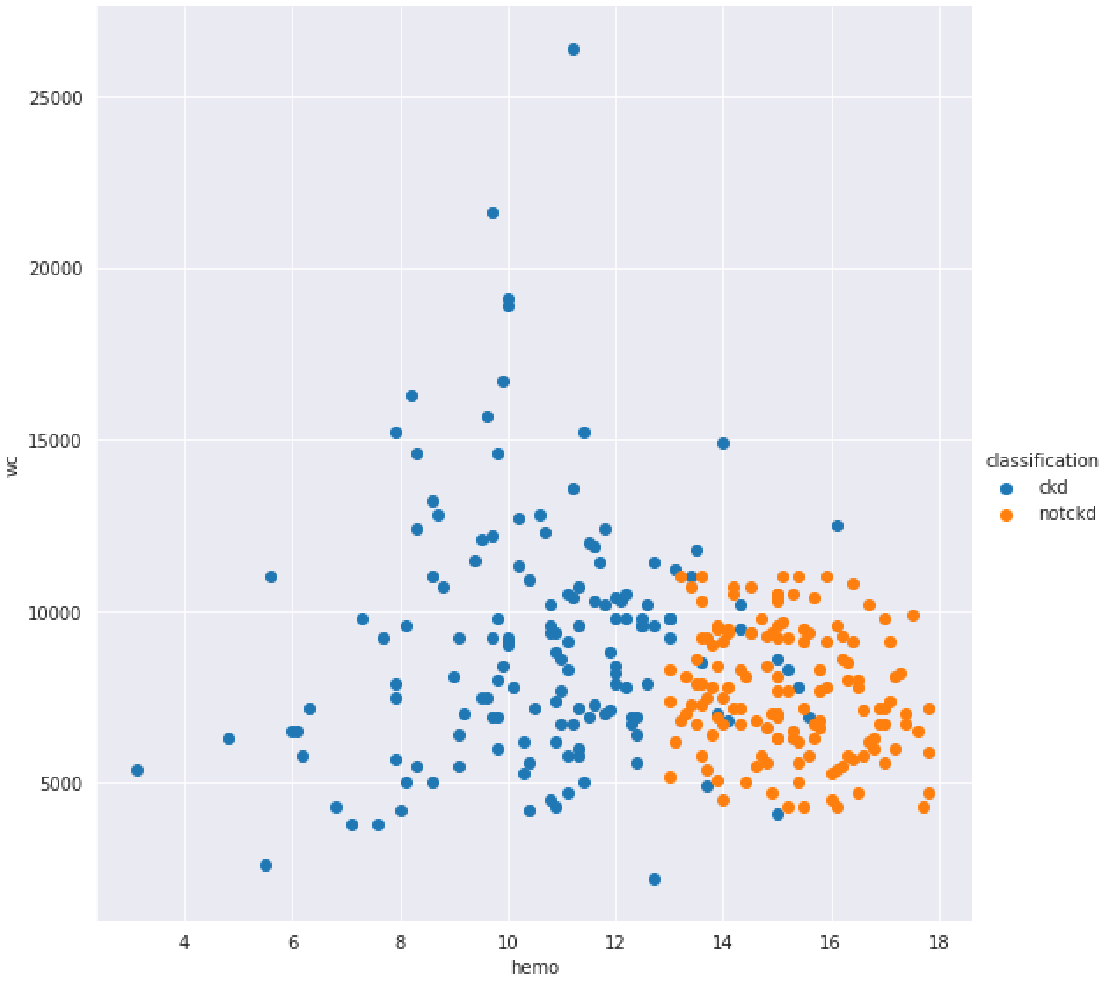


[ ]:
```

```

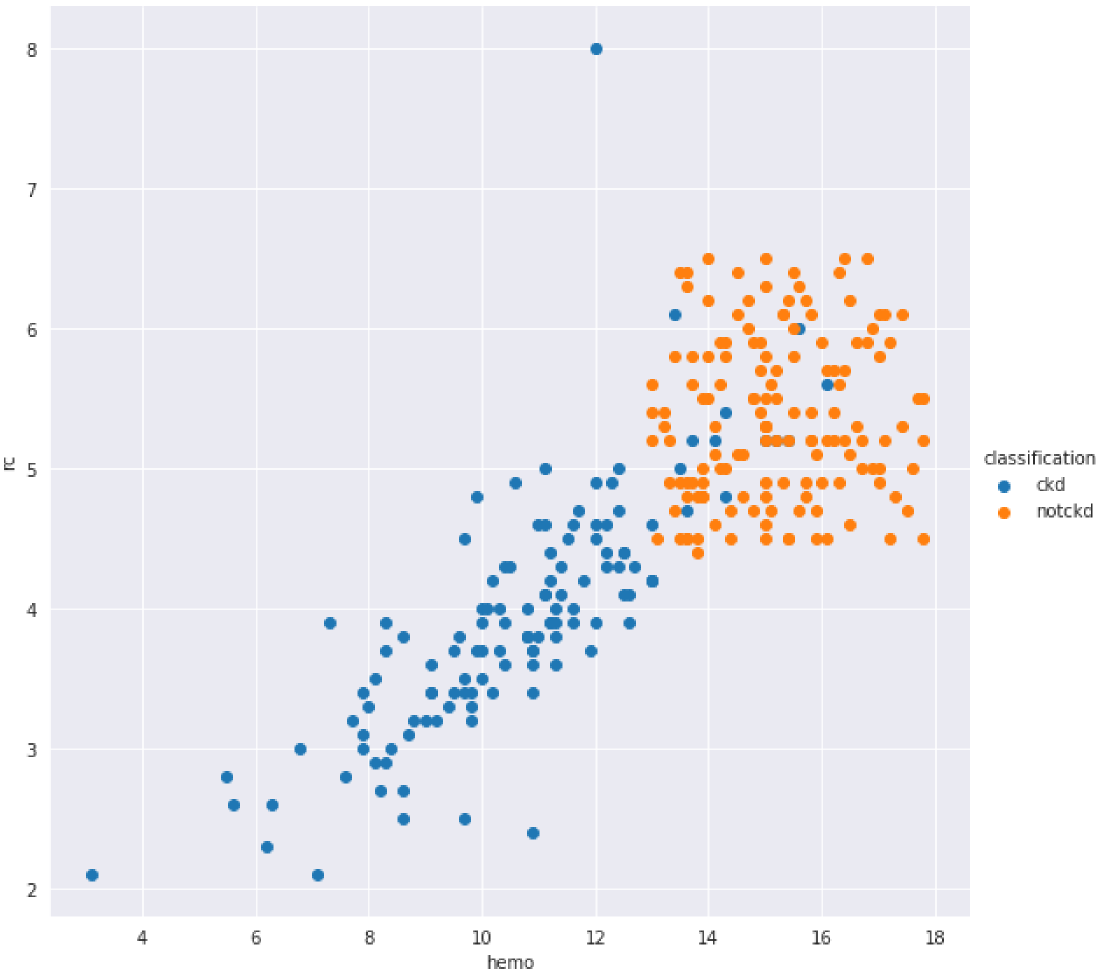
[ ]:
```

```

[ ]:
```
sns.kdeplot(kidney[kidney.classification=='ckd'].i,shade=True)
```

[77]:
```
sns.FacetGrid(kidney,hue='classification',height=8) \
    .map(plt.scatter,'hemo','wc') \
    .add_legend()
plt.show()
```

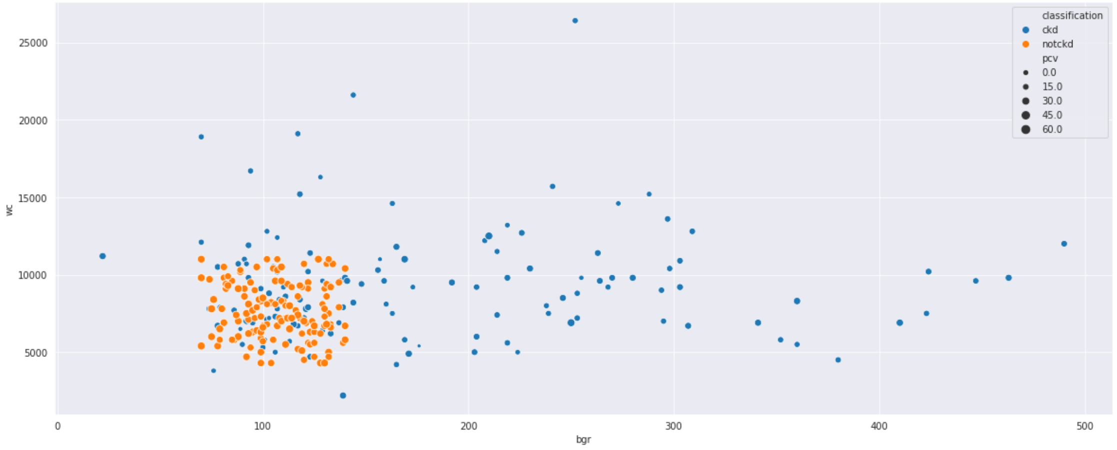


[76]:
```
sns.FacetGrid(kidney,hue='classification',height=8) \
    .map(plt.scatter,'hemo','rc',size=) \
    .add_legend()
plt.show()
```



```
[165]: plt.figure(1,figsize=(20,8))
       sns.scatterplot(x='bgr',y='wc',hue='classification',data=kidney,size='pcv')
```

```
[165]: <matplotlib.axes._subplots.AxesSubplot at 0x7fbd7dc101d0>
```



[ ]:

```
def clas(x):
  if x=='\tno':
    return 'no'
  elif x=='yes':
    return 'yes'
  elif x=='no':
    return 'no'
  else:
    return 'unknown'
```

[64]:

```
kidney.cad.unique()
```

[64]:

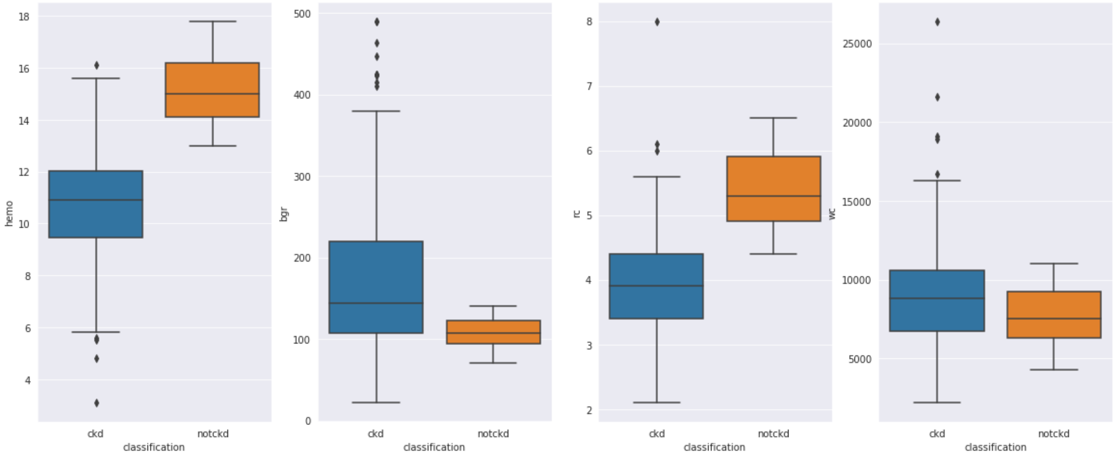
```
array(['no', 'yes', 'unknown'], dtype=object)
```

[ ]:

```
kidney.cad=kidney.cad.apply(clas,convert_dtype=True)
```

[310]:

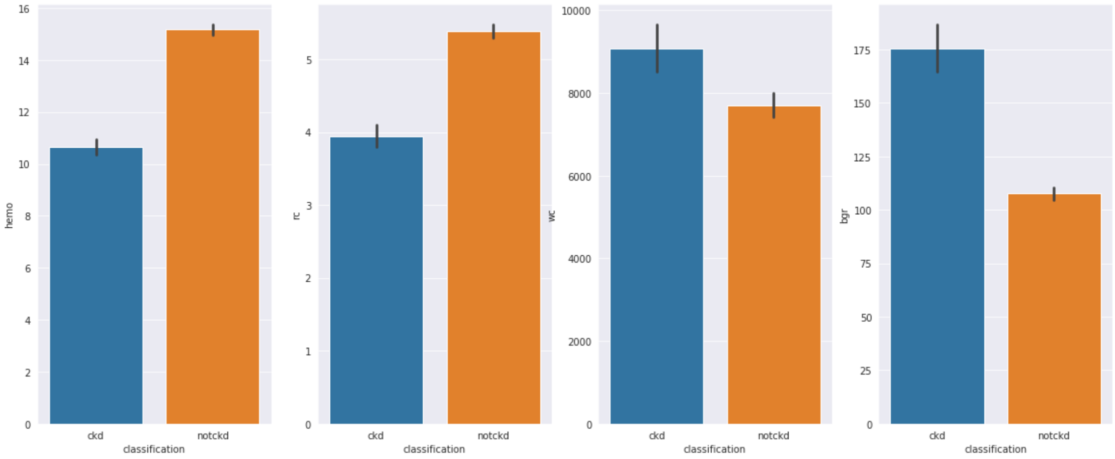
```
plt.figure(1,figsize=(20,8))
plt.subplot(141)
sns.boxplot(x='classification',y='hemo',data=kidney)
plt.subplot(142)
sns.boxplot(x='classification',y='bgr',data=kidney)
plt.subplot(143)
sns.boxplot(x='classification',y='rc',data=kidney)
plt.subplot(144)
sns.boxplot(x='classification',y='wc',data=kidney)

plt.show()
```



[162]:

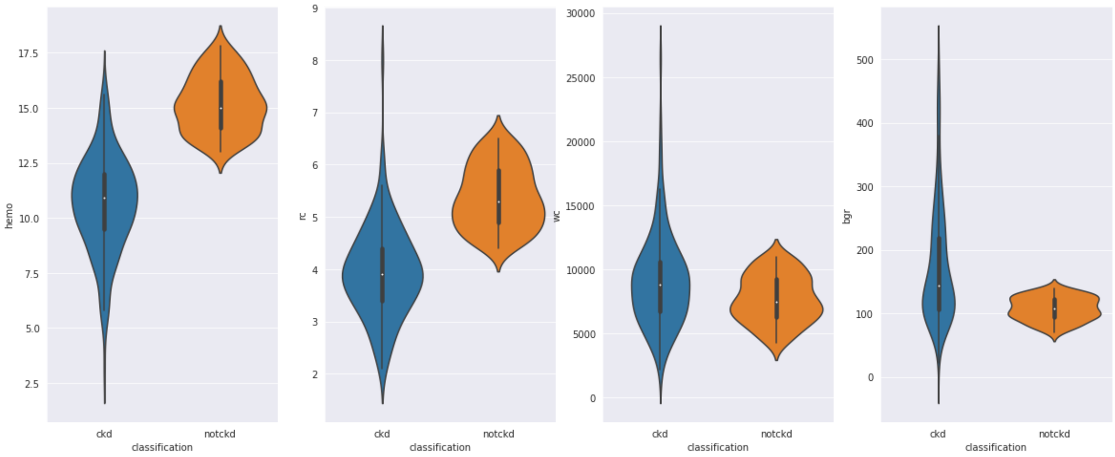
```
plt.figure(1,figsize=(20,8))
plt.subplot(141)
sns.barplot(x='classification',y='hemo',data=kidney)
plt.subplot(142)
```

```
sns.barplot(x='classification',y='rc',data=kidney)
plt.subplot(143)
sns.barplot(x='classification',y='wc',data=kidney)
plt.subplot(144)
sns.barplot(x='classification',y='bgr',data=kidney)
plt.show()
```



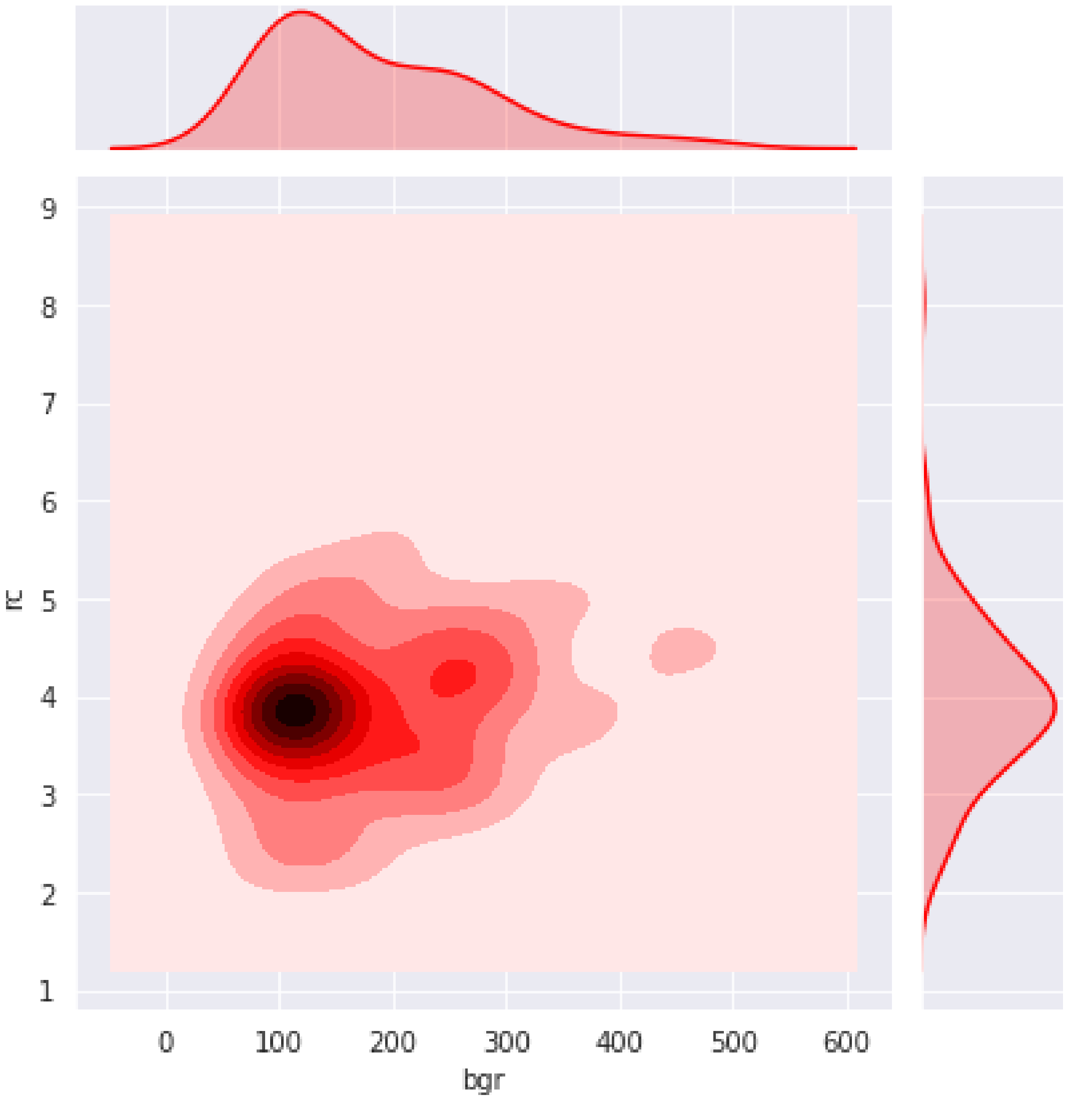
[ ]:
```
sns.countplot(x='classification',data=kidney)
```

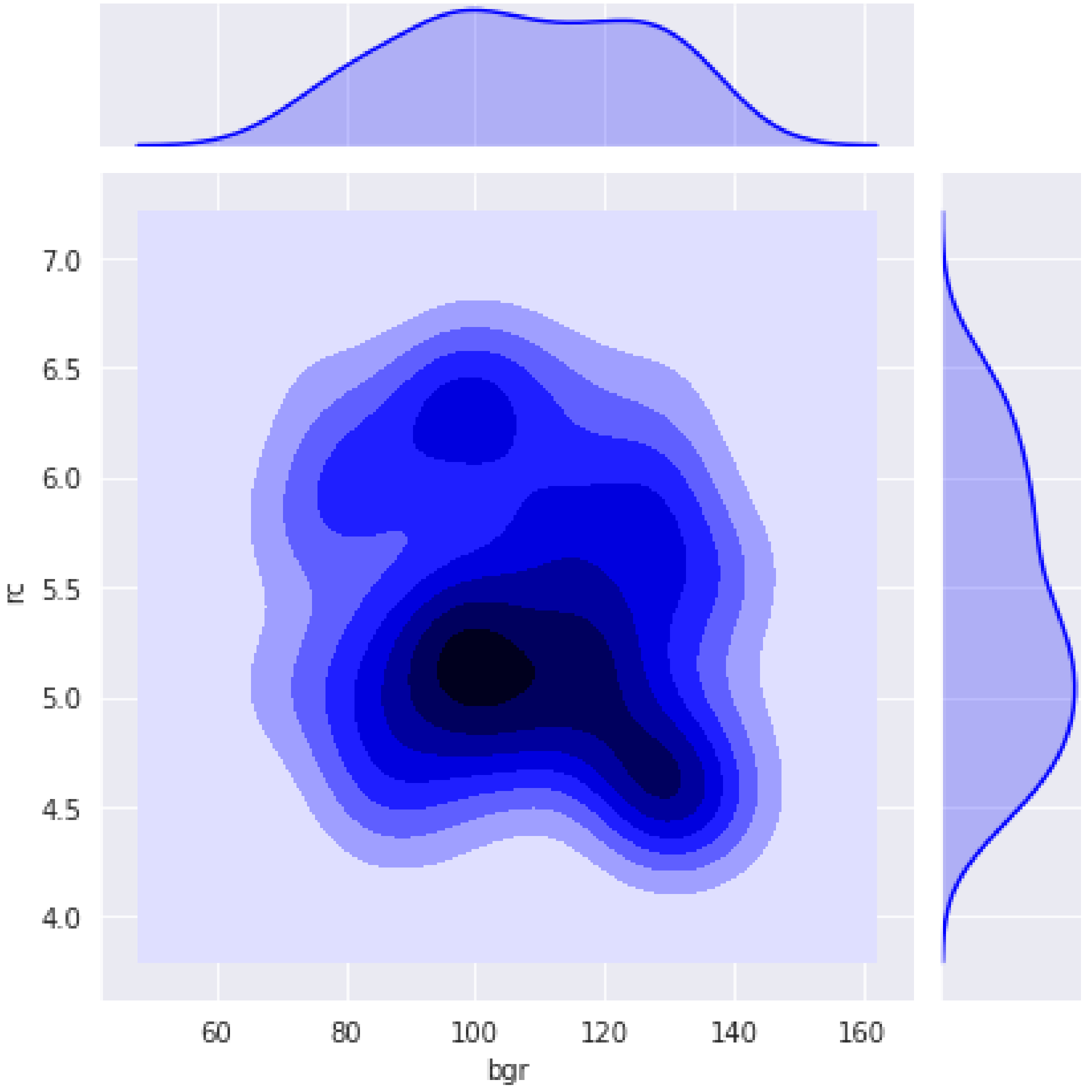
[163]:
```
plt.figure(1,figsize=(20,8))
plt.subplot(141)
sns.violinplot(x='classification',y='hemo',data=kidney)
plt.subplot(142)
sns.violinplot(x='classification',y='rc',data=kidney)
plt.subplot(143)
sns.violinplot(x='classification',y='wc',data=kidney)
plt.subplot(144)
sns.violinplot(x='classification',y='bgr',data=kidney)
plt.show()
```

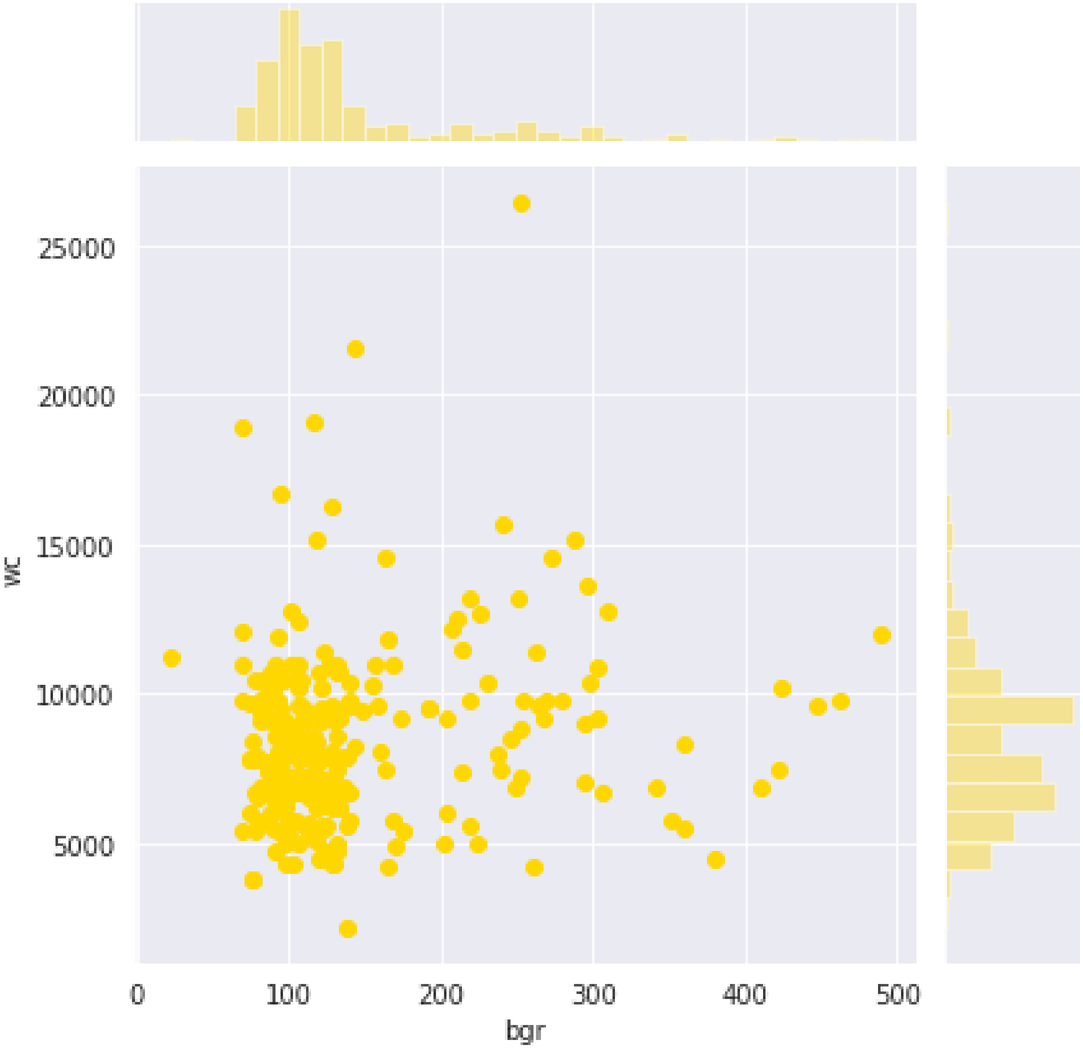


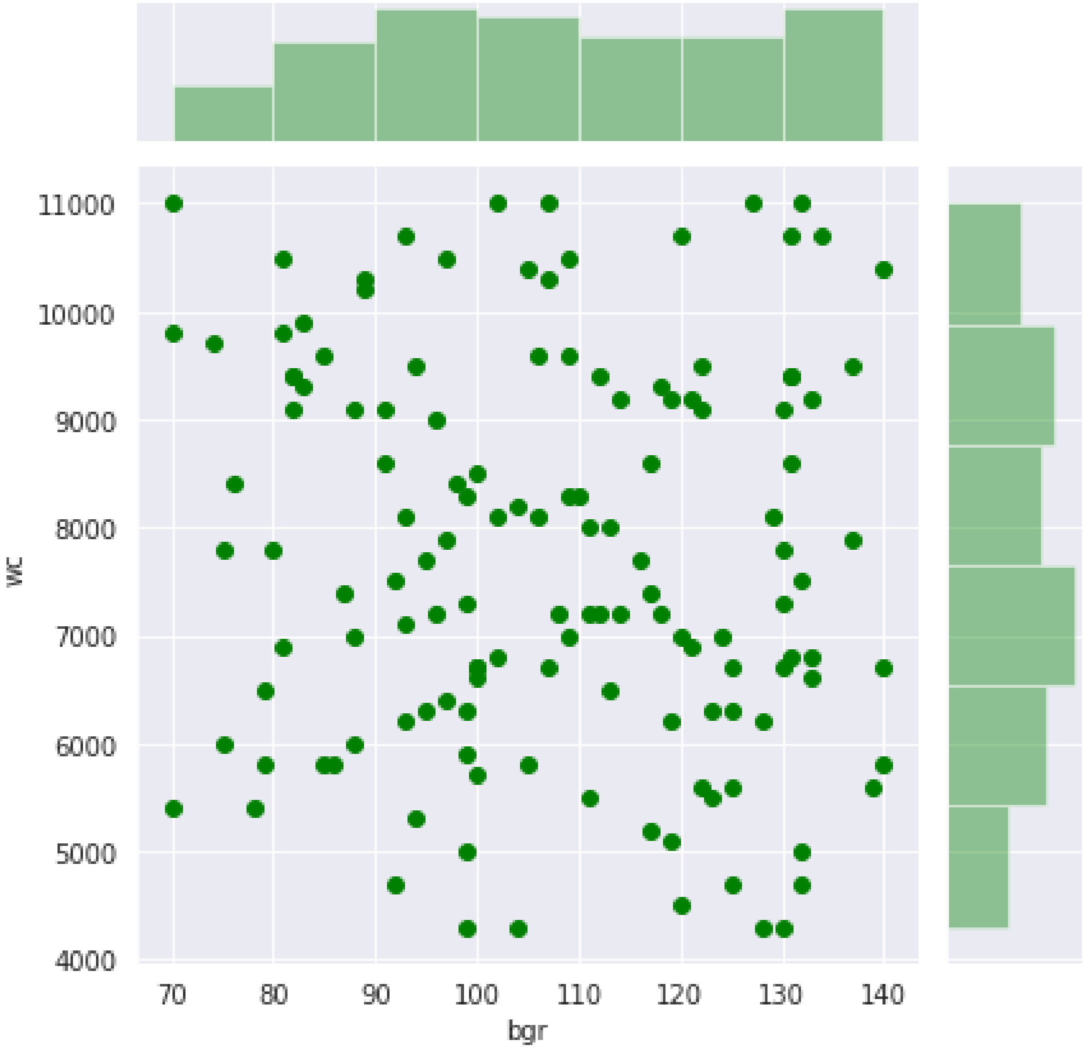
[167]:

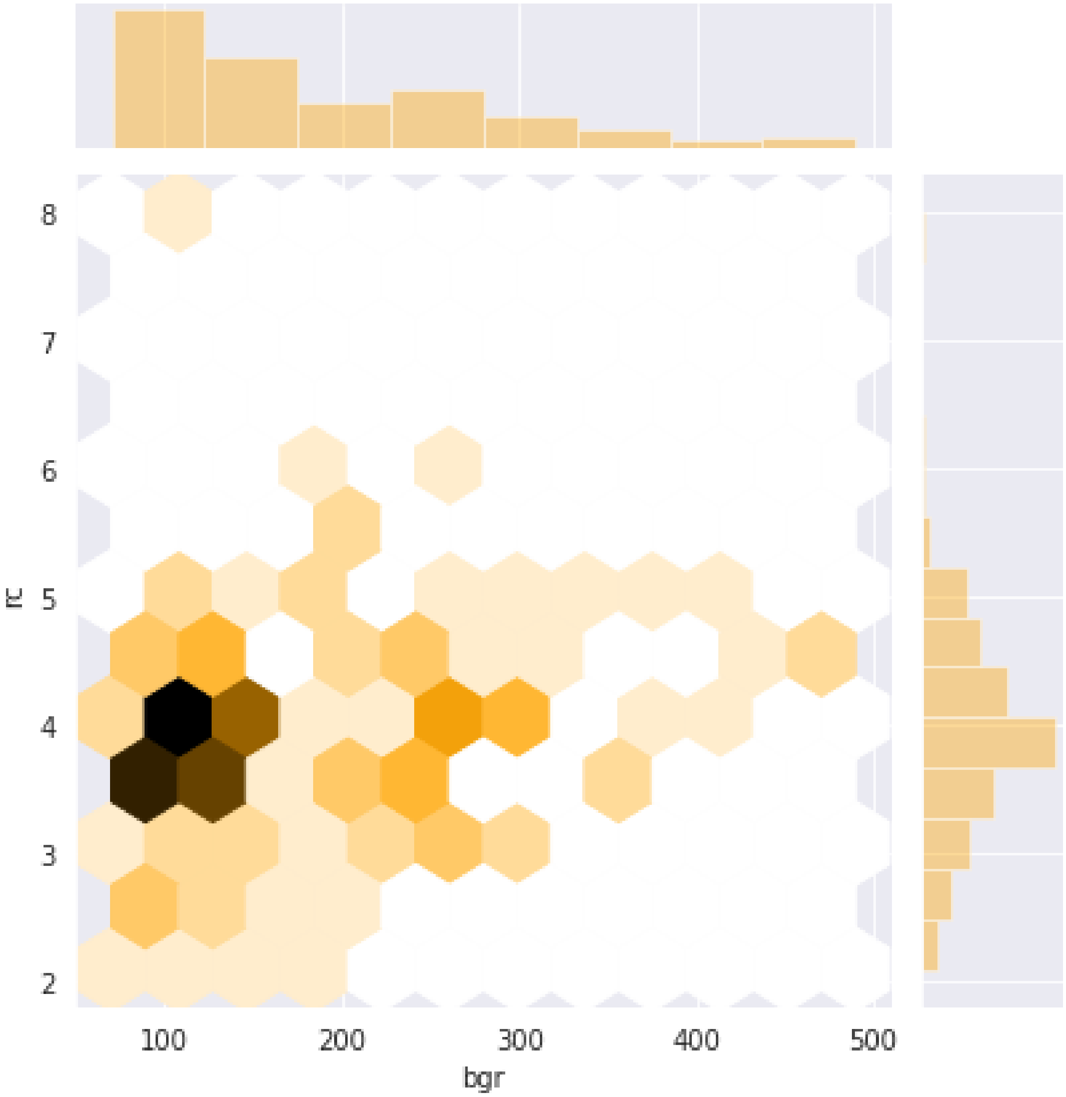
```
ax=sns.jointplot('bgr','rc',data=kidney[kidney.
 ↪classification=='ckd'],kind='kde',height=6,color='r')
ax=sns.jointplot('bgr','rc',data=kidney[kidney.
 ↪classification=='notckd'],kind='kde',height=6,color='b')
ax=sns.jointplot('bgr','wc',data=kidney,kind='scatter',height=6,color='gold')
ax=sns.jointplot('bgr','wc',data=kidney[kidney.
 ↪classification=='notckd'],kind='scatter',height=6,color='green')
ax=sns.jointplot('bgr','rc',data=kidney[kidney.
 ↪classification=='ckd'],kind='hex',height=6,color='orange')
ax=sns.jointplot('bgr','rc',data=kidney[kidney.
 ↪classification=='notckd'],kind='hex',height=6,color='yellow')
```

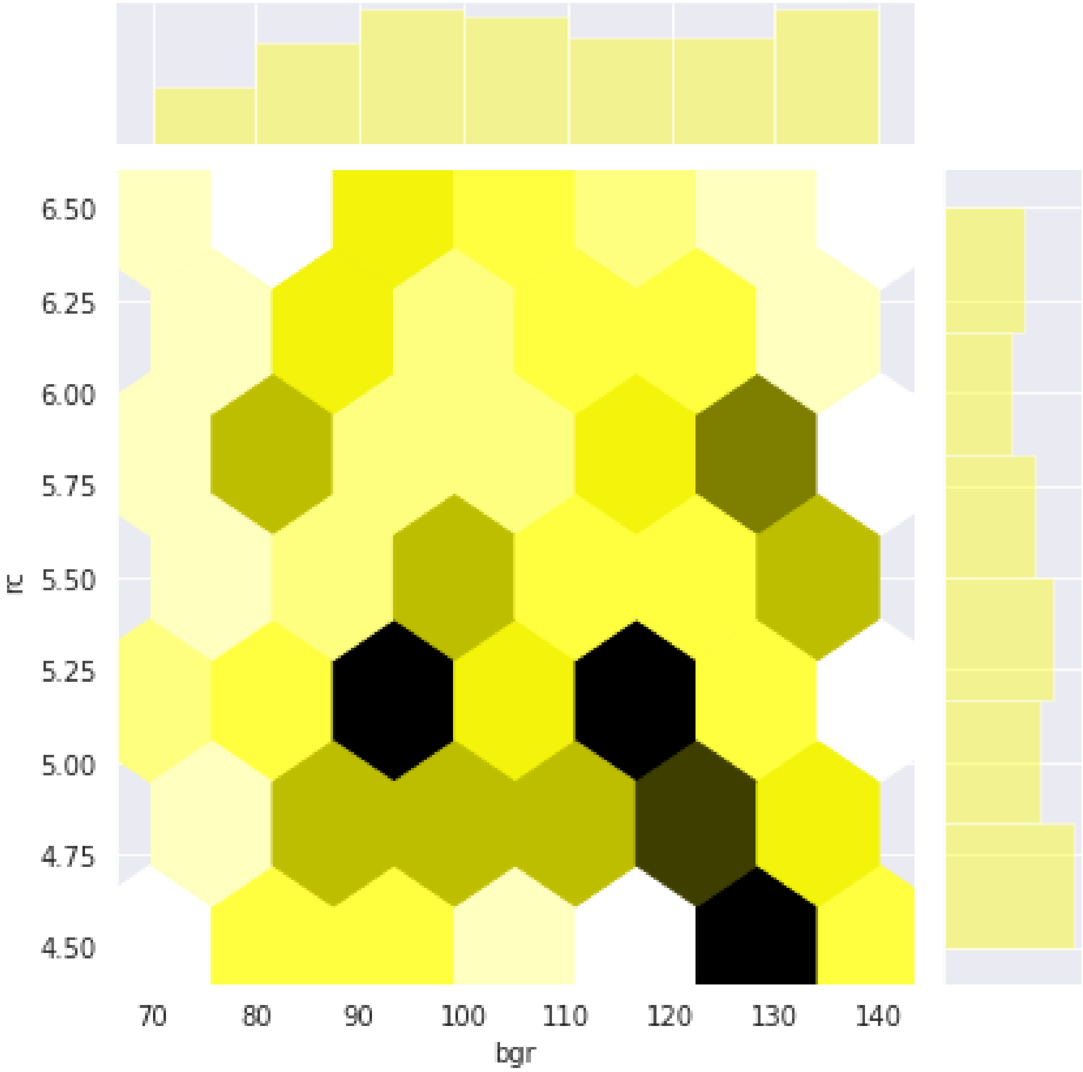



7.0
6.5
6.0
5.5
5.0
4.5
4.0
rc
60
80
100
120
140
160
bgr


25000
20000
15000
10000
5000
wc
0
100
200
300
400
500
bgr


11000
10000
9000
8000
7000
6000
5000
4000
wc
70
80
90
100
110
120
130
140
bgr


8
7
6
5
4
3
2
rc
100
200
300
400
500
bgr



[166]:
```
fig = px.scatter_3d(kidney, x='rc', z='wc',␣
 ↪y='bgr',color='classification',opacity=0.8,symbol='cad')
fig.show()
```

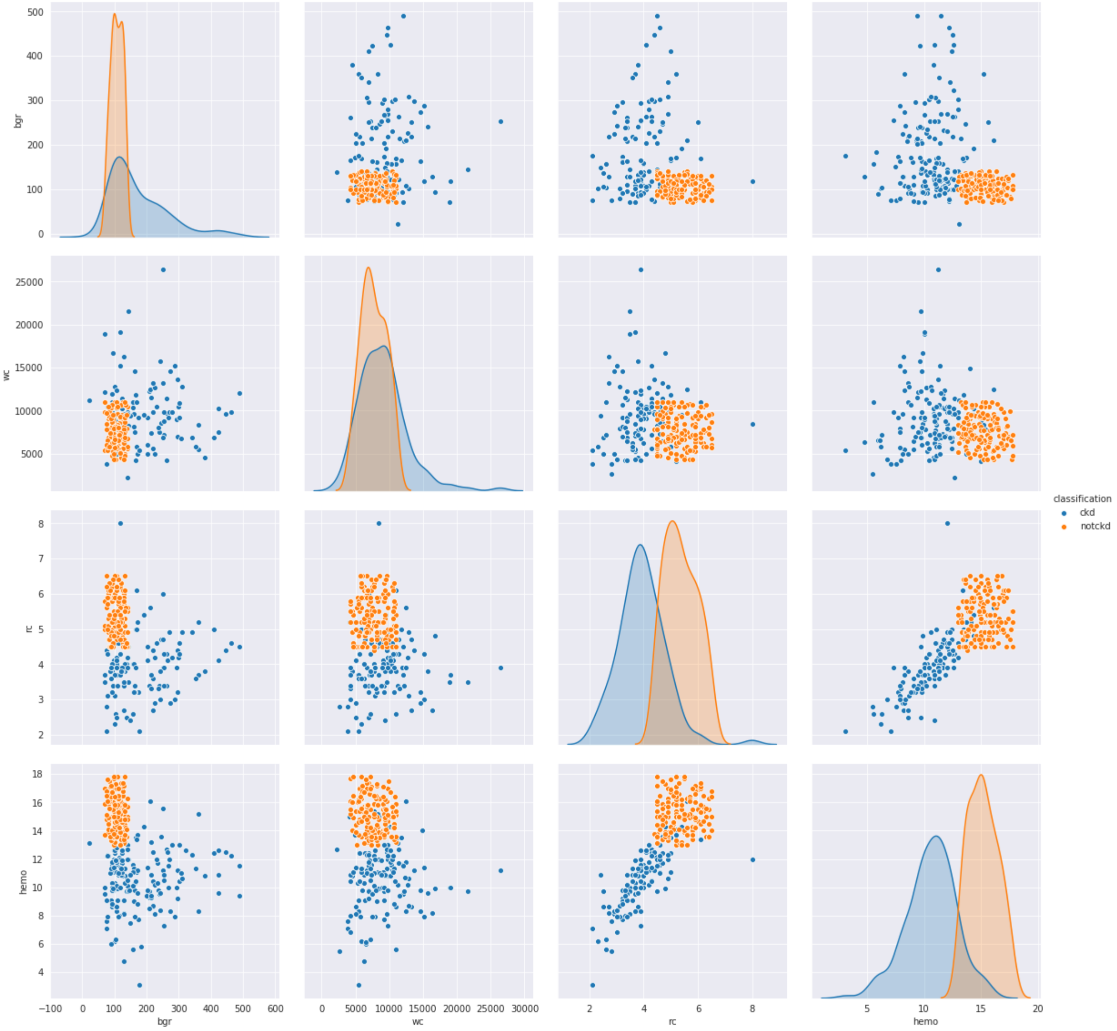
[ ]:
```
kidney=kidney.drop(['id'],axis=1)
```

[319]:
```
sns.
 ↪pairplot(kidney[['bgr','wc','rc','classification','hemo']],hue='classification',height=4)
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kde.py:447:
RuntimeWarning:

invalid value encountered in greater

/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kde.py:447:
```

```
RuntimeWarning:

invalid value encountered in less
```

[319]:
```
<seaborn.axisgrid.PairGrid at 0x7fbd783fb4e0>
```



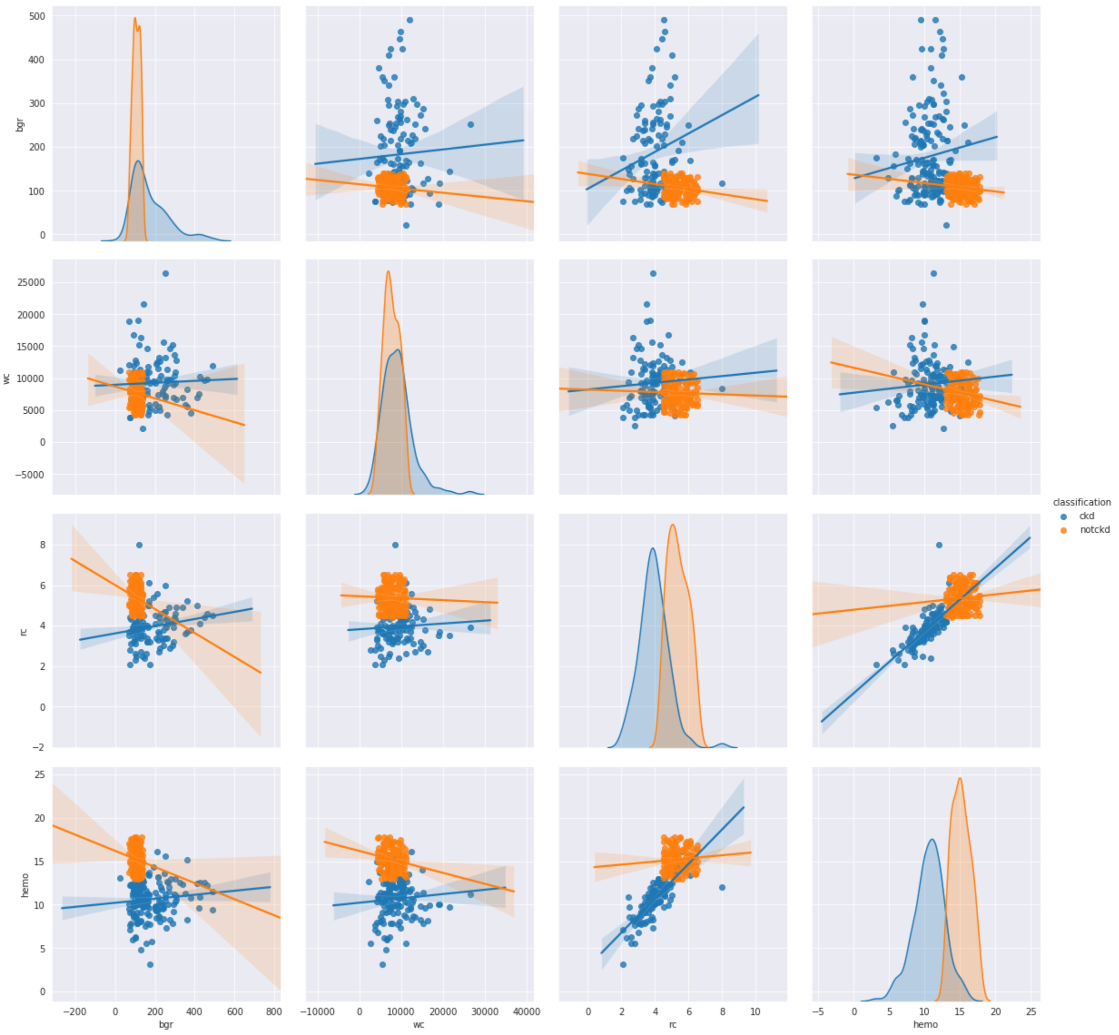
[193]:
```
sns.
  ↪pairplot(kidney[['bgr','wc','rc','classification','hemo']],kind='reg',hue='classification',
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kde.py:447:
RuntimeWarning:

invalid value encountered in greater
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/nonparametric/kde.py:447:
RuntimeWarning:

invalid value encountered in less
```

[193]:

```
<seaborn.axisgrid.PairGrid at 0x7fbd7d0a3278>
```



[174]:

```
kidney[kidney.classification=='ckd'][['bgr','wc','rc']].var()
```

[174]:

```
bgr    8.479136e+03
wc     1.282013e+07
rc     7.487371e-01
dtype: float64
```

[173]:

```
kidney[kidney.classification=='notckd'][['bgr','wc','rc']].var()
```

[173]:

```
bgr      3.446496e+02
wc       3.384757e+06
rc       3.553314e-01
dtype: float64
```

[187]:

```
kidney[kidney.classification=='ckd'][['bgr','wc','rc']].describe()
```

[187]:

```
              bgr            wc          rc
count  212.000000    151.000000  126.000000
mean   175.419811   9069.536424    3.945238
std     92.082223   3580.521254    0.865296
min     22.000000   2200.000000    2.100000
25%    106.750000   6750.000000    3.400000
50%    143.500000   8800.000000    3.900000
75%    219.250000  10600.000000    4.400000
max    490.000000  26400.000000    8.000000
```

[194]:

```
kidney[kidney.classification=='notckd'][['bgr','wc','rc']].describe()
```

[194]:

```
              bgr            wc          rc
count  144.000000    143.000000  143.000000
mean   107.722222   7705.594406    5.379021
std     18.564740   1839.770968    0.596097
min     70.000000   4300.000000    4.400000
25%     93.750000   6300.000000    4.900000
50%    107.500000   7500.000000    5.300000
75%    123.250000   9250.000000    5.900000
max    140.000000  11000.000000    6.500000
```

[ ]:

```

```

[ ]:

```
def man(x):
  if x=='ckd':
    return 1
  elif x=='notckd':
    return 2
```

[ ]:

```

```

[ ]:

```

```

[ ]:

```

```

[ ]:

```

```

[ ]:

```

```

[ ]:

```

```

[ ]:

```
messi1 = pd.DataFrame(dict(
    r=[95, 96, 97, 86, 91],
    theta=['finishing','Ball Control','dribbling',
           'sprint speed', 'agility']))
```

[ ]:

```
neymar=pd.DataFrame(dict(
    r=[87, 95, 96, 90, 96],
    theta=['finishing','Ball Control','dribbling',
           'sprint speed', 'agility']))
```

[ ]:

[ ]:

[ ]: