

Submission - Exercise [2]

Visual Data Analysis

[Maryam Assaedi, maryam.assaedi@rwth-aachen.de]
[Mst. Mahfuja Akter, s6msakte@uni-bonn.de]
[Mahpara Hyder Chowdhury, s6machow@uni-bonn.de]

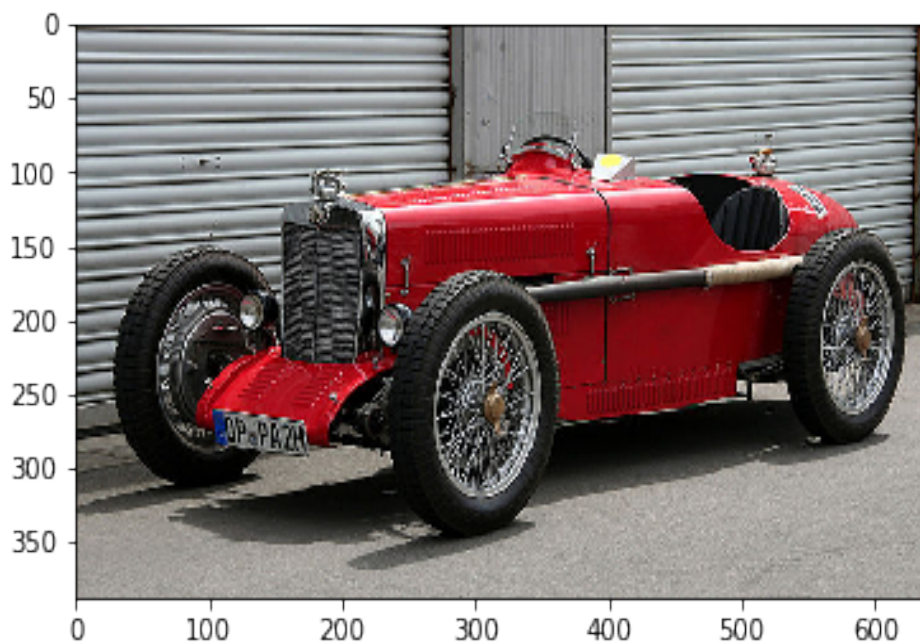
April 14, 2019

Exercise 1 (Working with HSV Color Space)

a) Read the given oldtimer.png image.
Using this line of code:

```
#a  
img = mpimg.imread('oldtimer.png')  
plt.imshow(img)
```

We were able to read in the image:

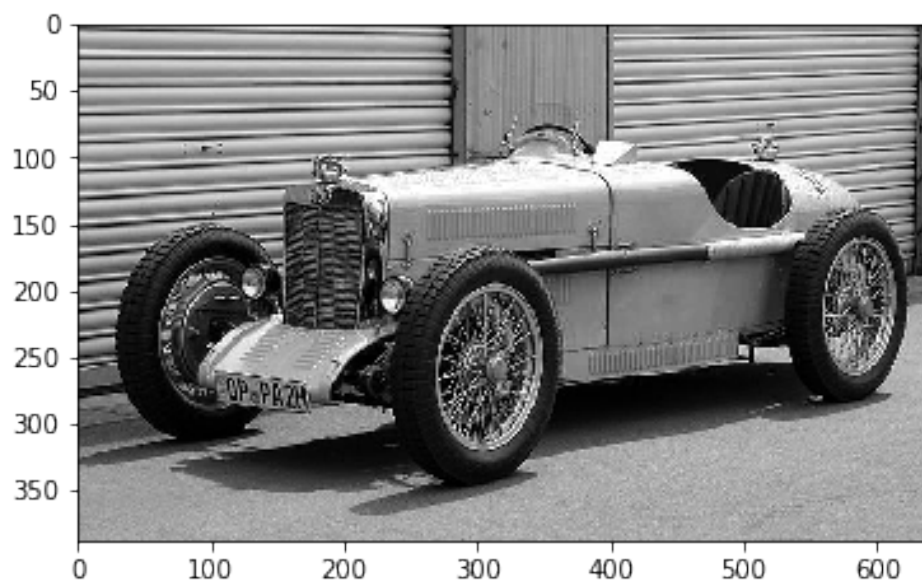


b) Convert the color image to HSV color space, and create a grayscale version by retaining only the value.

To create this effect we used this code:

```
#b
hsv_img = clr.rgb_to_hsv(img)
value_img= hsv_img[:, :, 2]
plt.imshow(value_img, cmap=plt.get_cmap('gray'), vmin=0, vmax=1)
```

To get this image:

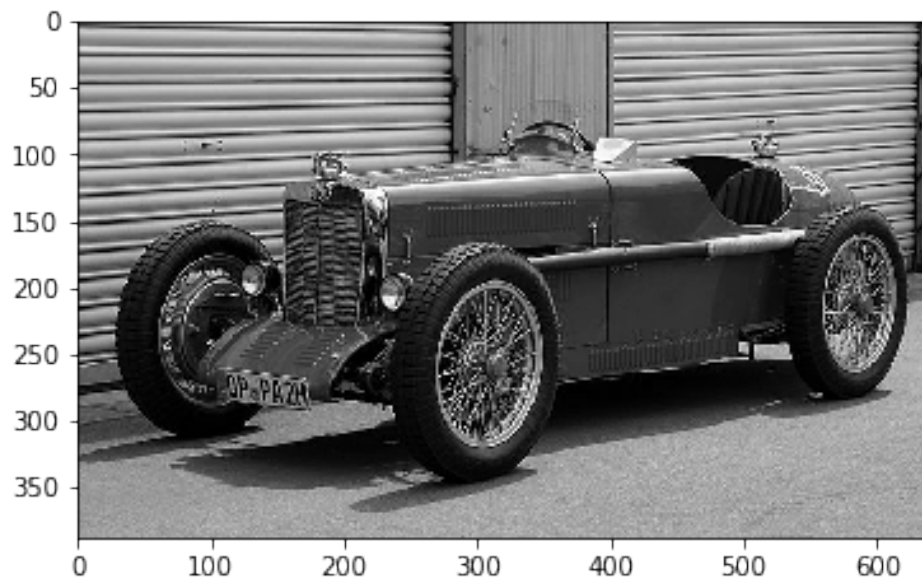


c) Think about another way to convert color image to grayscale. Its result should look reasonable (with bright regions appearing bright), but it should differ from the one you obtained from the HSV color space.

To create this effect we used this code:

```
#c
gray_img = np.mean(img, -1)
plt.imshow(gray_img, cmap=plt.get_cmap('gray'), vmin=0, vmax=1)
```

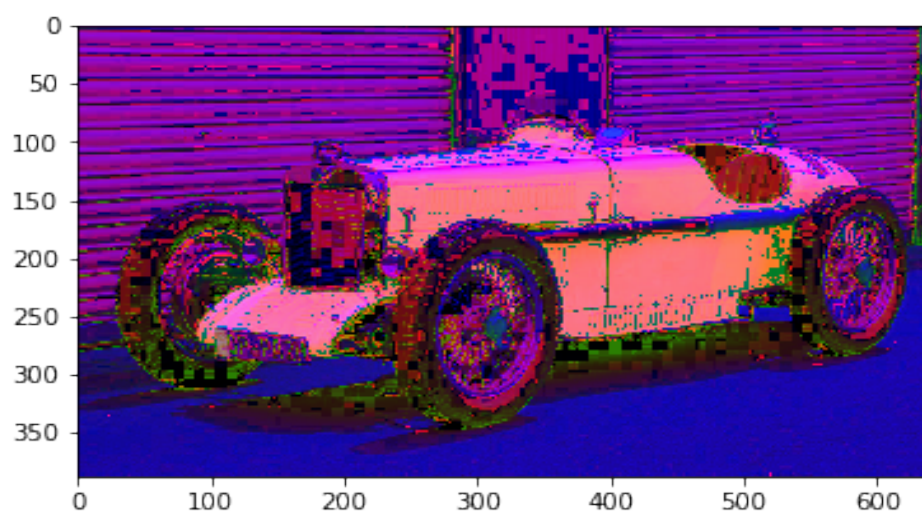
To get this image:



d) Output a version of the image in which saturation has been reduced by 50%. To create this effect we used this code:

```
#d
hsv_img[:, :, 1] *= 0.5
plt.imshow(hsv_img)
```

To get this image:

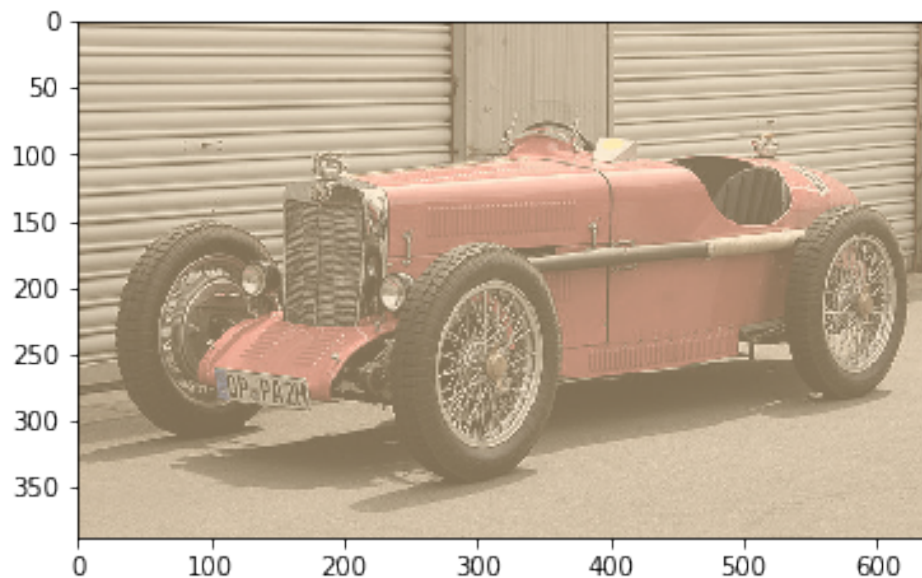


e) Create a simple "aged photograph" effect by globally blending (via simple weighted averaging or some other blend mode of your choice) the half-desaturated image from the previous step with some amount of brown.

To create this effect we used this code:

```
#e
image = np.zeros((388,640,3), dtype="uint8")
image[np.where((image==[0,0,0]).all(axis=2))] = [255,222,173]
brown_img=np.divide(image,255)
sat_img_rgb= clr.hsv_to_rgb(hsv_img)
old_img=np.add(brown_img,sat_img_rgb)
rgb_old_img = np.divide(old_img,2)
plt.imshow(rgb_old_img)
```

To get this image:

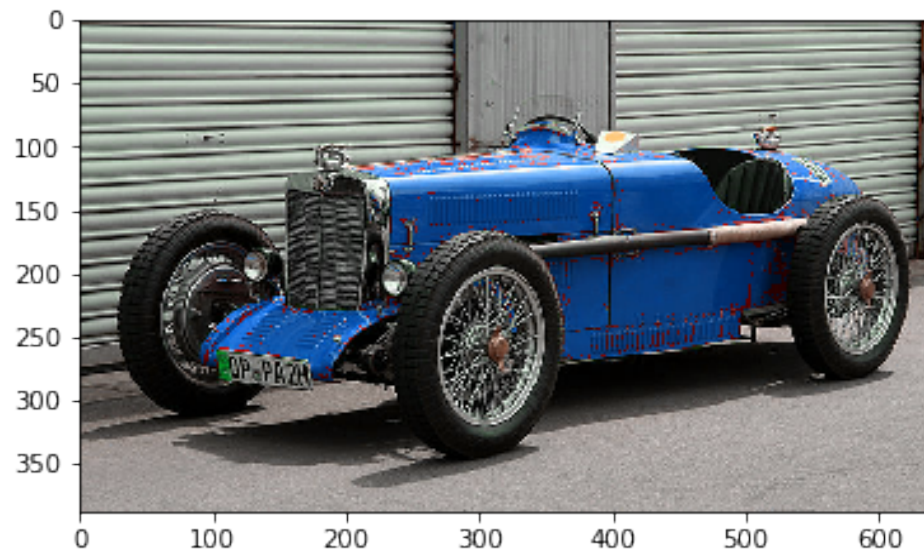


f) Rotate the hues of the original image so that the car obtains a different color. Provide two images in which the car is blue, and has another color of your choice.

To create the blue car effect we used this code:

```
#f
hsv_diff_hues=clr.rgb_to_hsv(img)
hsv_diff_hues[:, :, 0]*=0.6
rgb=clr.hsv_to_rgb(hsv_diff_hues)
plt.imshow(rgb)
```

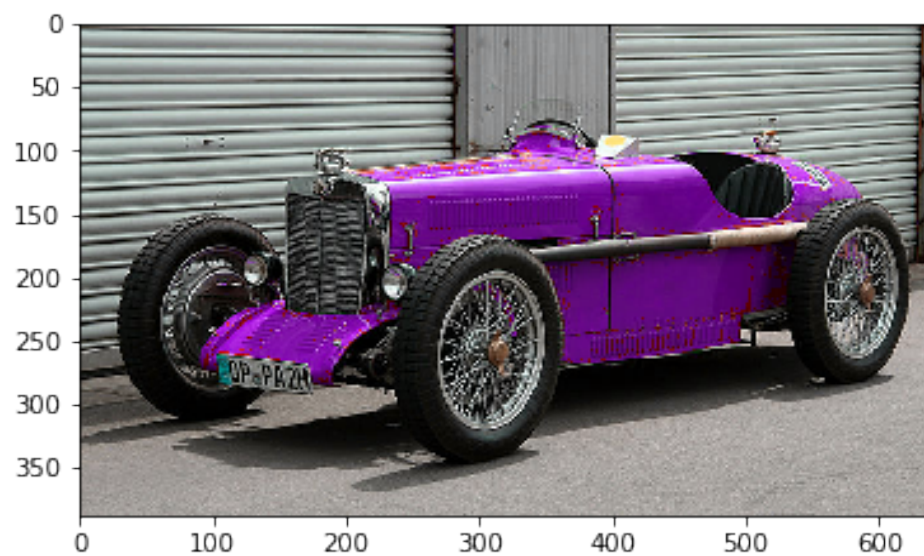

To get this image:



To create another color we used this code”

```
hsv_diff_hues2 =clr.rgb_to_hsv(img)
hsv_diff_hues2[:, :, 0] *=0.8
rgb2=clr.hsv_to_rgb(hsv_diff_hues2)
plt.imshow(rgb2)
```

To get this image:



Exercise 2 (Face-based Luminance Matching)

a) What is the Helmholtz-Kohlrausch effect?

Answer: Helmholtz-Kohlrausch or HK effect is that color light appears brighter to us than white light of the same luminance.

b) Why are the authors proposing to use images of faces?

Answer: The authors are proposing to use images of faces because human brain has dedicated brain circuitry to recognize faces and thus they are naturally good for it.

c) To what alternative method do the authors compare their newly proposed one in the user study?

Answer: The alternative method the authors compare their newly proposed one in the user study is Minimally Distinct Border(MDB). The new approach is similar to MDB and robust to the HK effect.

d) Why do the authors have to know the monitor gamma while creating a colormap based on the result of the user study?

Answer: The authors have found out that there double face task was better in terms of the precision of responses, than MDB as assessed through the standard deviation calculated over the six trials for each color and participant.

e) Why do the authors have to know the monitor gamma while creating a colormap based on the result of the user study?

Answer: They can produce continuous color range between the 6 main hues in the RGB space.

Exercise 3 (Human Vision and Color Spaces)

a) How do the related concepts of luminance, brightness, and lightness differ?

Answer: Luminance is objective physical measure of the amount of visible light.

Unit: candela/meter square

Brightness is subjective attribute of light. It is perceived, cannot be measured objectively. Brightness is perceived Luminance.

Lightness is subjective reflectance of a surface. Lightness is perceived Brightness.

b) What motivated development of the CIELAB and CIELUV color spaces?

Answer: The main goal was to provide perceptual equal space. this means, the Euclidean distance between two colors in the CIELUV/CIELAB color space is strongly correlated with the human visual perception.

c) How could we use the Corn sweet effect to our advantage when creating visualizations?

Answer: While creating visualization, we can use corn sweet effect in cases where we need to show fine details(e.g. text).

d) Sketch a CIE chromaticity diagram and mark the following features:

Spectral locus

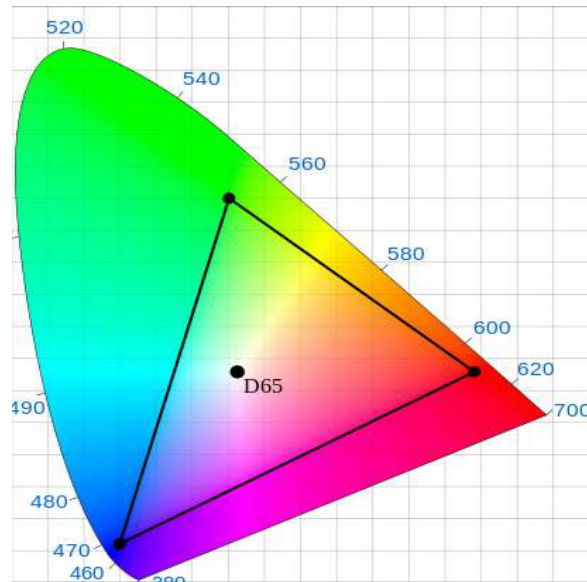
Purple boundary

D65 whitepoint (approximate location)

Answer: The outer curve boundary is the Spectral locus.

The straight line of the diagram is called Purple boundary of Line of Purples

D65 point is located in the diagram is the white point.



Exercise 4 ((Plotting Categorical Data))

a) What is an advantage of stripplots over swarmplots? What is an advantage of swarmplots over stripplots?

Answer: In swarmplots, points are adjusted on categorical axis so that they don't overlap. It gives a better representation of distribution.

Use of swarmplots in code:

```
sns.catplot(x="day", y="total_bill", hue="sex", kind="swarm", data=tips):
```

In stripplots, One variable is categorical, it can be used in conjunction with other plots to show each observation. It can scale large number of distribution.

Use of stripplots in code:

```
sns.stripplot(x="day", y="total_bill", hue="sex", data=tips):
```

b) What is meant by IQR?

Answer:

IQR means interquartile range, where bulk of the values lie. It is calculated by subtracting the 1st quartile from the 3rd quartile or equal to the difference between 75th and 25th percentiles

$IQR = Q3 - Q1$

Unlike range, IQR tells where the majority data lies. It can be used to identify outliers of data sets.

c) What would be a reason to use violinplots instead, or in addition to boxplots?

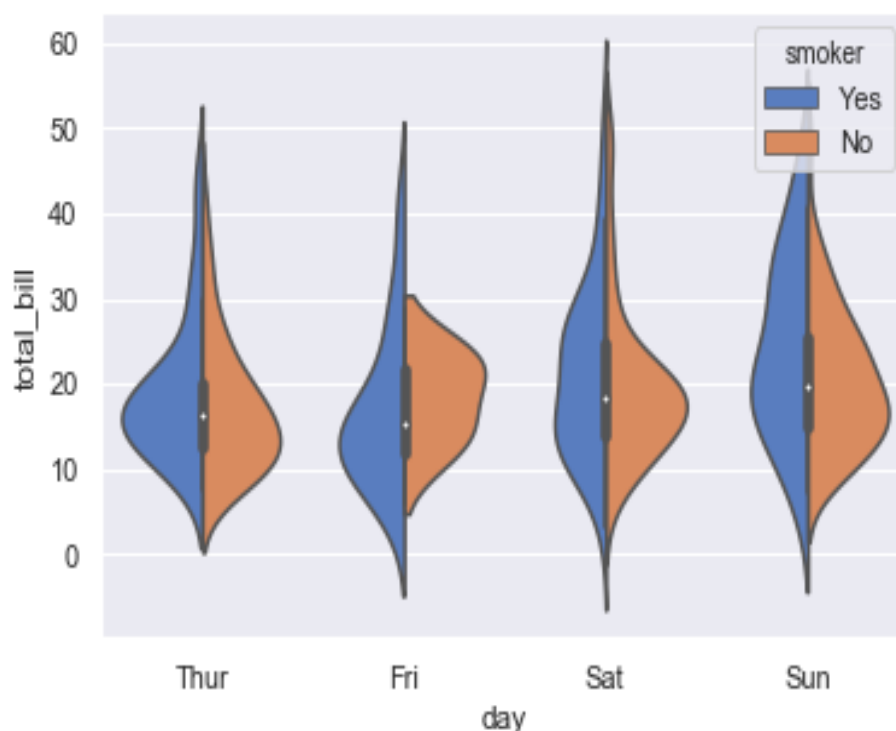
Answer:

In addition to boxplots, a new approach is violinplots, It combines a boxplot with the kernel density estimation to provide a richer description of distribution of values. Additionally the boxplots can be shown inside the violin. Violinplot uses a KDE.

Use of code for violinplot:

```
sns.violinplot(x="day", y="total_bill", hue="smoker", data=tips, palette="muted", split=True):
```

ViolinPlot diagram



Programming Question Part:

a) Read the dataset using pandas read_excel function.

Answer:

Reading excel data using this line of code:

```
#a
data_from_excel = pd.read_excel("chronic_kidney_disease_numerical.xls",
                                sheet_name="chronic_kidney_disease_full"):
```

b) Use pandas.melt to transform the data from "wide" to "long" format, using class (indicating ckd for chronic kidney disease or notckd for its absence) as the identifier variable.

Using this line of code transformed data is written in wide_to_long_output.xlsx file:


```
#b
melt_wide_to_long = pd.melt(data_from_excel, id_vars=["class"],
                             var_name="Name", value_name="value");
melt_wide_to_long.to_excel(r'wide_to_long_output.xlsx',
                           sheet_name='transformedData', index=False);
```

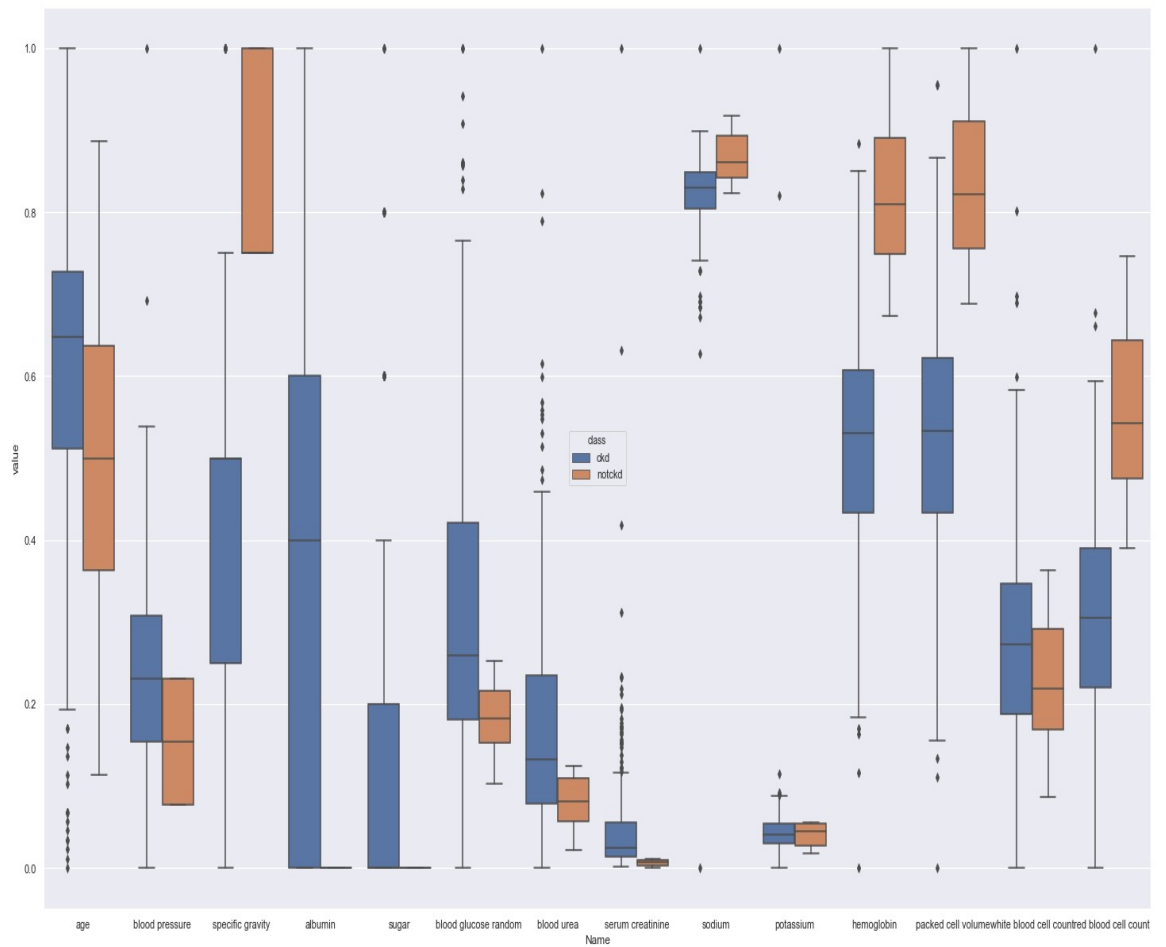
c) For each numerical attribute, such as age or blood pressure, create two boxplots side-by-side. One should show the attribute's distribution among patients suffering from chronic kidney disease, the other one from patients who do not suffer from the disease. Hint: Due to the different numerical ranges, you will have to disable sharing of y axes between plots of different attributes.

Answer:

Using this line of code we have create 2 boxplot side by side:

```
#c
normalizedf=pd.DataFrame()
normalizedf['class'] = data_from_excel['class'].values
def norm_to_zero_one(data_from_excel):
    return (data_from_excel - data_from_excel.min())*1.0 / (data_from_excel.max()-data_from_excel.min())
for column in data_from_excel:
    if(column != 'class'):
        columndfe = data_from_excel[column]
        normalized_x =norm_to_zero_one(columndfe)
        normalizedf[column]=normalized_x.values
dnormalizedf = pd.melt(normalizedf, id_vars=["class"], var_name= "Name", value_name="value");
sns.boxplot(x="Name", y="value", hue=dnormalizedf._series['class'],data=dnormalizedf)
plt.gcf().set_size_inches(25, 15)
```

We found below distribution.



d) Based on viewing the plots, name an attribute that appears to be highly indicative of chronic kidney disease, and one that seems to be mostly unrelated to it.

Answer:

From above plot from question c, we can find the highly indicative of chronic kidney disease attribute is "albumin" and mostly unrelated or less indicative attribute is "potassium".