**SCHOOL OF COMPUTING (SOC)**

**CA1 Sample Resource**

**SPECIALIST DIPLOMA IN DATA SCIENCE**

**IT8701  
Introduction to Programming for Data Science**

**What’s in this document?**

1. This document shows some useful code snippets which you can reference and apply in your IT8701 CA1 assignment
2. Do note that these sample codes are not the “best” solutions you can implement.
3. If you can, come up with better algorithms or codes and impress your lecturer with them! 😊
4. Good luck!

**Table of Contents**

[Section 1 2](#_Toc531265233)

[Example 1 Simple Text-based Analysis using Numpy 2](#_Toc531265234)

[Output of code 2](#_Toc531265235)

[Code snippet 3](#_Toc531265236)

[Example 2 Create a Line Chart 4](#_Toc531265237)

[Output of code 4](#_Toc531265238)

[Code snippet 4](#_Toc531265239)

[Example 3 Create a Scatterplot 6](#_Toc531265240)

[Output of code 6](#_Toc531265241)

[Code snippet 6](#_Toc531265242)

[Example 3 Create a Barchart 8](#_Toc531265243)

[Output of code 8](#_Toc531265244)

[Code snippet 8](#_Toc531265245)

[Example 4 Create a Boxplot 10](#_Toc531265246)

[Output of code 10](#_Toc531265247)

[Code snippet 11](#_Toc531265248)

[Example 5 Create a Histogram 13](#_Toc531265249)

[Output of code 13](#_Toc531265250)

[Code snippet 13](#_Toc531265251)

# Section 1

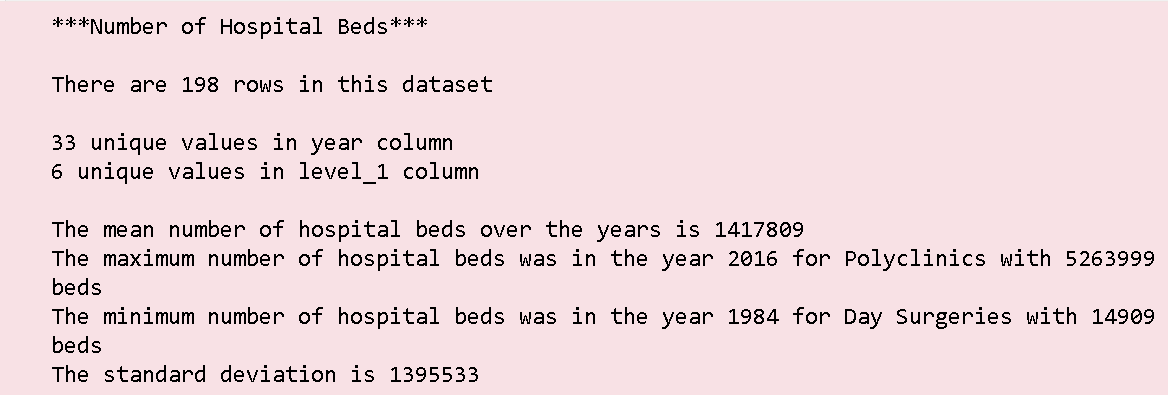
## Example 1 Simple Text-based Analysis using Numpy

This output uses the Numpy library to load a MOH (Ministry of Health) CSV that contains the number of hospital beds by year, and quickly breaks down the data with some simple useful-to-know information.

With this quick breakdown, we quickly realise the price column may have n/a values since the isnumeric is False for this column.

It also helps us to think about how we may want to extract subsets of this dataset and the choice of chart type for data visualization later.

### Output of code



### Code snippet

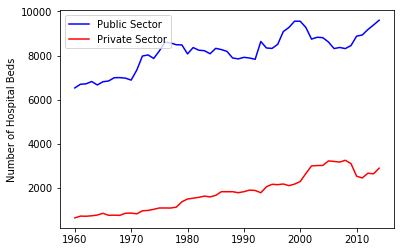
The dataset is taken from <https://data.gov.sg/dataset/number-of-hospital-beds>

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  title = "Number of Hospital Beds"  titlelen = len(title)  print("{:\*^{titlelen}}".format(title, titlelen=titlelen+6))  print()  data = np.genfromtxt('data/hospital-admissions-and-public-sector-outpatient-attendances-annual.csv',  skip\_header=1,  dtype=[('year','U10'), ('level\_1','U50'),('value','f8')], delimiter=",",  missing\_values=['na','-'],filling\_values=[0])  print("There are {} rows in this dataset".format(len(data)))  print()  set\_year = set(data['year'])  set\_level\_1 = set(data['level\_1'])  print(str(len(set\_year)) + " unique values in year column")  print(str(len(set\_level\_1 )) + " unique values in level\_1 column")  max\_value = np.max(data['value'])  min\_value = np.min(data['value'] )  max\_row\_index = np.argmax(data['value'])  min\_row\_index = np.argmin(data['value'])  var = np.var(data['value'])  std = np.std(data['value'])  print()  print("The mean number of hospital beds over the years is {:.0f}".format(np.mean(data['value'])))  print("The maximum number of hospital beds was in the year {} for {} with {:.0f} beds".format(data[max\_row\_index][0],data[max\_row\_index][1],data[max\_row\_index][2]))  print("The minimum number of hospital beds was in the year {} for {} with {:.0f} beds".format(data[min\_row\_index][0],data[min\_row\_index][1],data[min\_row\_index][2]))  print("The standard deviation is {:.0f}".format(std)) |

## Example 2 Create a Line Chart

### Output of code

This sample shows you how to code the following output using Matplotlib.



### Code snippet

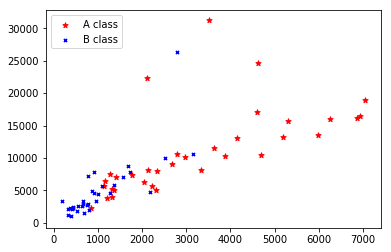
The dataset is taken from <https://data.gov.sg/dataset/number-of-hospital-beds>

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  title = "Number of Hospital Beds"  titlelen = len(title)  print("{:\*^{titlelen}}".format(title, titlelen=titlelen+6))  print()  data = np.genfromtxt('data/number-of-hospital-beds-annual.csv',  skip\_header=1,  dtype=[('year','U10'), ('sector','U50'),('no\_of\_hospital\_beds','i8')], delimiter=",",  missing\_values=['na','-'],filling\_values=[0])  x\_public = data[data['sector']=='Public Sector']['year']  y\_public = data[data['sector']=='Public Sector']['no\_of\_hospital\_beds']  x\_private = data[data['sector']=='Private Sector']['year']  y\_private = data[data['sector']=='Private Sector']['no\_of\_hospital\_beds']  fig = plt.figure()  ax1 = fig.add\_subplot(111)  ax1.plot(x\_public, y\_public, c='b', label='Public Sector')  ax1.plot(x\_private,y\_private, c='r', label='Private Sector')  plt.ylabel('Number of Hospital Beds')  plt.legend(loc='upper left');  plt.show() |

## Example 3 Create a Scatterplot

### Output of code

This sample shows you how to code the following output using Matplotlib.



### Code snippet

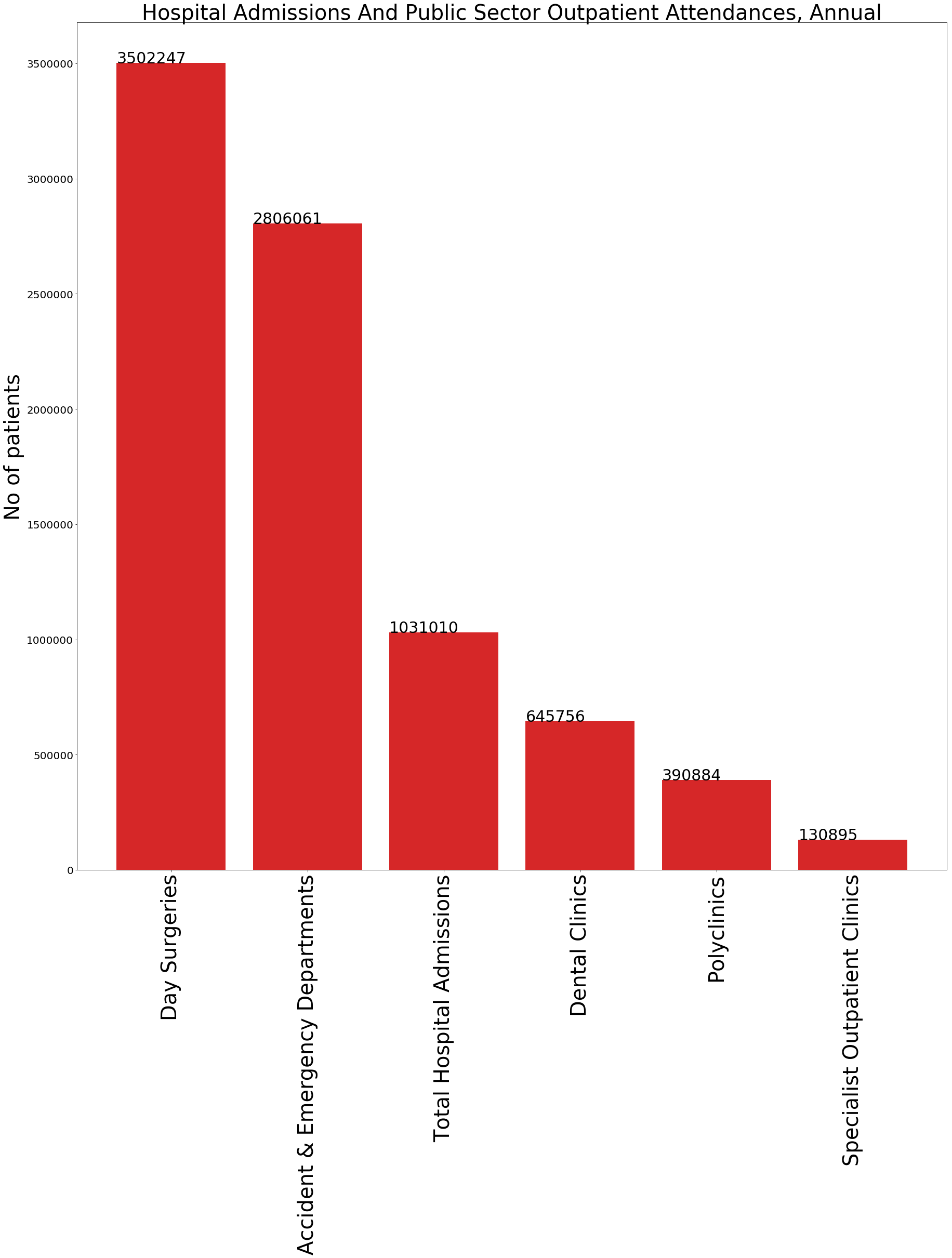
The dataset is taken from https://data.gov.sg/dataset/average-hospital-inpatient-bill-size

|  |
| --- |
| import numpy as np  import numpy as np  import matplotlib.pyplot as plt  filename = 'data/average-hospital-inpatient-bill-size.csv'  title = "Average hospital inpatient bill size"  titlelen = len(title)  print("{:\*^{titlelen}}".format(title, titlelen=titlelen+6))  print()  data = np.genfromtxt(filename, skip\_header=1,  dtype=[('year','U10'),  ('specialty','U50'),  ('sector','U50'),  ('wardclass','U50'),  ('hospital','U50'),('avg\_per\_day','i8'),  ('avg\_total\_bill','i8'), ('total\_bill\_95p','i8')],  delimiter=",",  missing\_values=['na','-'],filling\_values=[0])  x\_avgperday\_A = data[data['wardclass']=='A']['avg\_per\_day']  y\_avgperday\_A = data[data['wardclass']=='A']['avg\_total\_bill']  criteria = (data['wardclass']=='B1') | (data['wardclass']=='B2' )  x\_avgperday\_B = data[criteria]['avg\_per\_day']  y\_avgperday\_B = data[criteria]['avg\_total\_bill']  fig = plt.figure()  ax1 = fig.add\_subplot(111)  ax1.scatter(x\_avgperday\_A , y\_avgperday\_A , s=30, c='red', marker="\*", label='A class')  ax1.scatter(x\_avgperday\_B , y\_avgperday\_B , s=10, c='blue', marker="x", label='B class')  plt.legend(loc='upper left');  plt.show() |

## Example 3 Create a Barchart

### Output of code

This sample shows you how to code the following output using Matplotlib.



### Code snippet

The dataset is taken from <https://data.gov.sg/dataset/hospital-admissions-and-public-sector-outpatient-attendances-annual> (Note: Data is no longer available at original URL, you can download a copy of it at <https://drive.google.com/open?id=1pmbL-p8xDAiLj_Ap7ipXSj6mSgBe0dAn>)

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  title = "Hospital Admissions"  titlelen = len(title)  print("{:\*^{titlelen}}".format(title, titlelen=titlelen+6))  print()  filename ='data/hospital-admissions-and-public-sector-outpatient-attendances-annual.csv'  data = np.genfromtxt(filename,  skip\_header=1,  dtype=[('year','U10'), ('level\_1','U50'),('value','f8')], delimiter=",",  missing\_values=['na','-'],filling\_values=[0])  #print("Original data: " + str(data.shape))  null\_rows = np.isnan(data['value'])  nonnull\_values = data[null\_rows==False]  #print("Filtered data: " + str(nonnull\_values.shape))  labels = list(set(data['level\_1']))  levels = np.arange(0,len(labels))  levels\_values = data[['level\_1','value']]  values = levels\_values['value']  #print(values)  avg\_values = {}  for i in labels:  valuesforlevel = values[levels\_values['level\_1']==i]  avg = np.average(valuesforlevel)  print("Average for level " + i + " is {:.0f}".format(avg))  avg\_values[i] = avg  **# Note this part is to allow the barcharts to be displayed in descending order**  from collections import OrderedDict  from operator import itemgetter  avg\_values = OrderedDict(sorted(avg\_values.items(), key = itemgetter(1), reverse = True))  plt.figure(1, figsize=(30,30))  barchart = plt.bar(list(avg\_values.keys()), list(avg\_values.values()), color='#d62728')  for i in range(len(barchart)):  bar = barchart[i]  x,y = bar.get\_xy()  h = bar.get\_height()  plt.text(x,h,"{:.0f}".format(list(avg\_values.values())[i]),fontsize=30)  plt.title('Hospital Admissions And Public Sector Outpatient Attendances, Annual',fontsize=40)  plt.ylabel('No of patients',fontsize=40)  plt.yticks(fontsize=20)  plt.xticks(levels, labels, fontsize=40,rotation='vertical')  plt.show() |

## Example 4 Create a Boxplot

### Output of code

This sample shows you how to code the following output using Matplotlib.

A screenshot of a cell phone

Description generated with very high confidence

### Code snippet

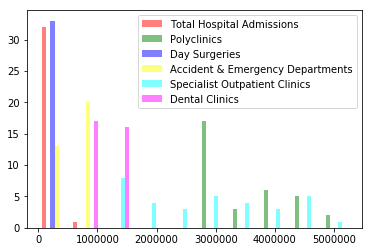
The dataset is taken from <https://data.gov.sg/dataset/hospital-admissions-and-public-sector-outpatient-attendances-annual> (Note: Data is no longer available at original URL, you can download a copy of it at <https://drive.google.com/open?id=1pmbL-p8xDAiLj_Ap7ipXSj6mSgBe0dAn>)

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  title = "Hospital Admissions"  titlelen = len(title)  print("{:\*^{titlelen}}".format(title, titlelen=titlelen+6))  print()  filename = 'data/hospital-admissions-and-public-sector-outpatient-attendances-annual.csv'  data = np.genfromtxt(filename,  skip\_header=1,  dtype=[('year','U10'), ('level\_1','U50'),('value','f8')], delimiter=",",  missing\_values=['na','-'],filling\_values=[0])  null\_rows = np.isnan(data['value'])  nonnull\_values = data[null\_rows==False]  labels = list(set(data['level\_1']))  labels.sort()  print(labels)  levels = np.arange(0,len(labels))  levels\_values = data[['level\_1','value']]  values = levels\_values['value']  values\_TotalHospitalAdmissions = values[levels\_values ['level\_1'] == 'Total Hospital Admissions']  values\_Polyclinics = values[levels\_values['level\_1'] == 'Polyclinics']  values\_DaySurgeries = values[levels\_values['level\_1'] == 'Day Surgeries']  values\_AE = values[levels\_values['level\_1'] == 'Accident & Emergency Departments']  values\_SpecialistOutpatientClinics = values[levels\_values ['level\_1'] == 'Specialist Outpatient Clinics']  values\_DentalClinics = values[levels\_values ['level\_1'] == 'Dental Clinics']  values\_combined =[values\_AE, values\_DaySurgeries,values\_DentalClinics, values\_Polyclinics,  values\_SpecialistOutpatientClinics,  values\_TotalHospitalAdmissions]  print(len(values\_combined))  plt.figure(2, figsize=(30,30))  plt.title(title,fontsize=60)  plt.ylabel('Number of patients',fontsize=50)  plt.yticks(fontsize=40)  plt.xticks(fontsize=40,rotation='vertical')  bp\_dict = plt.boxplot(values\_combined,labels=labels,patch\_artist=True)  ## change outline color, fill color and linewidth of the boxes  for box in bp\_dict['boxes']:  # change outline color  box.set( color='#7570b3', linewidth=2)  # change fill color  box.set( facecolor = '#1b9e77' )    ## change color and linewidth of the whiskers  for whisker in bp\_dict['whiskers']:  whisker.set(color='#7570b3', linewidth=2)  ## change color and linewidth of the caps  for cap in bp\_dict['caps']:  cap.set(color='#7570b3', linewidth=2)  ## change color and linewidth of the medians  for median in bp\_dict['medians']:  median.set(color='#b2df8a', linewidth=2)  ## change the style of fliers and their fill  for flier in bp\_dict['fliers']:  flier.set(marker='D', color='#e7298a', alpha=0.5)  print(bp\_dict.keys())  for line in bp\_dict['medians']:  # get position data for median line  x, y = line.get\_xydata()[1] # top of median line  # overlay median value  plt.text(x, y, '%.1f' % y,  horizontalalignment='center',fontsize=30) # draw above, centered  fliers = []  for line in bp\_dict['fliers']:  ndarray = line.get\_xydata()  if (len(ndarray)>0):  max\_flier = ndarray[:,1].max()  max\_flier\_index = ndarray[:,1].argmax()  x = ndarray[max\_flier\_index,0]  print("Flier: " + str(x) + "," + str(max\_flier))    plt.text(x,max\_flier,'%.1f' % max\_flier,horizontalalignment='center',fontsize=30,color='green')  plt.show() |

## Example 5 Create a Histogram

### Output of code

This sample shows you how to code the following output using Matplotlib.



### Code snippet

The dataset is taken from <https://data.gov.sg/dataset/hospital-admissions-and-public-sector-outpatient-attendances-annual> (Note: Data is no longer available at original URL, you can download a copy of it at <https://drive.google.com/open?id=1pmbL-p8xDAiLj_Ap7ipXSj6mSgBe0dAn>)

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  title = "Hospital Admissions"  titlelen = len(title)  print("{:\*^{titlelen}}".format(title, titlelen=titlelen+6))  print()  data = np.genfromtxt('data/hospital-admissions-and-public-sector-outpatient-attendances-annual.csv',  skip\_header=1,  dtype=[('year','U10'), ('level\_1','U50'),('value','f8')], delimiter=",",  missing\_values=['na','-'],filling\_values=[0])  #print("Original data: " + str(data.shape))  null\_rows = np.isnan(data['value'])  nonnull\_values = data[null\_rows==False]  #print("Filtered data: " + str(nonnull\_values.shape))  labels = list(set(data['level\_1']))  labels = ['Total Hospital Admissions','Polyclinics','Day Surgeries','Accident & Emergency Departments','Specialist Outpatient Clinics','Dental Clinics']  levels = np.arange(0,len(labels))  levels\_values = data[['level\_1','value']]  values = levels\_values['value']  values\_TotalHospitalAdmissions = values[levels\_values ['level\_1'] == 'Total Hospital Admissions']  values\_Polyclinics = values[levels\_values['level\_1'] == 'Polyclinics']  values\_DaySurgeries = values[levels\_values['level\_1'] == 'Day Surgeries']  values\_AE = values[levels\_values['level\_1'] == 'Accident & Emergency Departments']  values\_SpecialistOutpatientClinics = values[levels\_values ['level\_1'] == 'Specialist Outpatient Clinics']  values\_DentalClinics = values[levels\_values ['level\_1'] == 'Dental Clinics']  values\_combined =[values\_TotalHospitalAdmissions,  values\_Polyclinics,  values\_DaySurgeries,  values\_AE,  values\_SpecialistOutpatientClinics,  values\_DentalClinics]  # Create bins of 2000 each  #bins = np.arange(data1.min(), data2.max(), 2000) # fixed bin size  plt.figure(1) # first figure  hist\_all = plt.hist(values\_combined,  alpha=0.5,  color=['red','green','blue','yellow','cyan','magenta'],  label=labels)  #,'Day Surgeries','A&E','Specialised OC','Dental'  plt.legend()  plt.figure(2) # second figure  plt.hist(values\_TotalHospitalAdmissions,  alpha=0.5,  color=['red'])  plt.title("Total Hospital Admissions",fontsize=20)  plt.ylabel('Number of patients',fontsize=10)  plt.figure(3) # second figure  plt.hist(values\_Polyclinics,  alpha=0.5,  color=['green'])  plt.title("Polyclinics",fontsize=20)  plt.ylabel('Number of patients',fontsize=10)  plt.figure(4) # second figure  plt.hist(values\_DaySurgeries,  alpha=0.5,  color=['blue'])  plt.title("Day Surgeries",fontsize=20)  plt.ylabel('Number of patients',fontsize=10)  plt.figure(5) # second figure  plt.hist(values\_DaySurgeries,  alpha=0.5,  color=['yellow'])  plt.title("A&E",fontsize=20)  plt.ylabel('Number of patients',fontsize=10)  plt.figure(6) # second figure  plt.hist(values\_SpecialistOutpatientClinics,  alpha=0.5,  color=['cyan'])  plt.title("Specialist Outpatient Clinics",fontsize=20)  plt.ylabel('Number of patients',fontsize=10)  plt.xticks(rotation='vertical')  plt.figure(7) # second figure  plt.hist(values\_DentalClinics,  alpha=0.5,  color=['magenta'])  plt.title("Dental Clinics",fontsize=20)  plt.ylabel('Number of patients',fontsize=10)  plt.xticks(rotation='vertical')  plt.show() |

**-- End of Assignment Sample --**