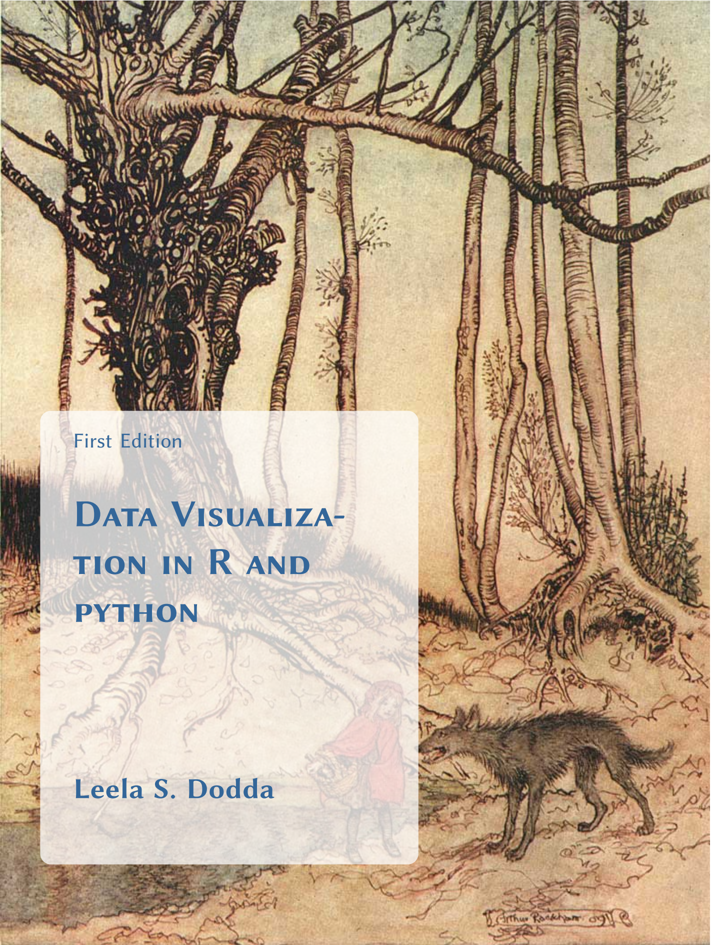

First Edition

Data Visualization in R and python

Leela S. Dodda

Contents

II R recipies 22

Introduction

This is one of my finer quotations.
–John Smith

This is a great place to write an introduction or prologue[1].

[1] *You can even use a footnote to seem smarter*

Part I

Python recipies

Chapter 1

Linear Regression

Packages required to run this code

pandas for reading csv files[1] format

scipy for doing linear regression analysis and obtaining the statistics

matplotlib for making the plots

[1] *data not shown as tables*

```
import matplotlib
matplotlib.use('Agg')
matplotlib.rc('font', family='serif')
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
cm5=pd.read_csv("CM5_ARRANGED_DATA_FROM_R.csv")
cm1=pd.read_csv("CM1A_ARRANGED_DATA_FROM_R.csv")
xcm5=cm5['Expt']
ycm5=cm5['G121']
xcm1=cm1['Expt']
ycm1=cm1['G105']
m1,c1,r1,p1,se1=stats.linregress(xcm1,ycm1)
m5,c5,r5,p5,se5=stats.linregress(xcm5,ycm5)
fig=plt.figure(figsize=(10, 5),dpi=300)
ax1 = plt.subplot(121)
cm1lab="$"+('y=%2.2fx+%2.2f, r^2=%1.2f'%(m1,c1,r1**2))+"$"
ax1.plot(xcm1,ycm1,'^',mfc='none',mec='b',mew=1.2)
ax1.plot(xcm1, m1*xcm1+c1,'k--',linewidth=2,label=cm1lab)
plt.grid()
plt.ylabel(r'$\Delta G^{GB/SA}_{hyd}~$ 1.05*CM1A (kcal/mol)',fontsize=16)
plt.xlabel(r'$\Delta G^{Expt}_{hyd}~$ (kcal/mol)',fontsize=16)
ax1.legend( loc='upper left')
ax2 = plt.subplot(122)
cm5lab="$"+('y=%2.2fx+%2.2f, r^2=%1.2f'%(m5,c5,r5**2))+"$"
ax2.plot(xcm5,ycm5,'o',mfc='none',mec='r',mew=1.2)
ax2.plot(xcm5, m5*xcm5+c5,'k--',linewidth=2,label=cm5lab)
ax2.legend( loc='upper left')
plt.ylabel(r'$\Delta G^{GB/SA}_{hyd}~$ 1.21*CM5 (kcal/mol)',fontsize=16)
```

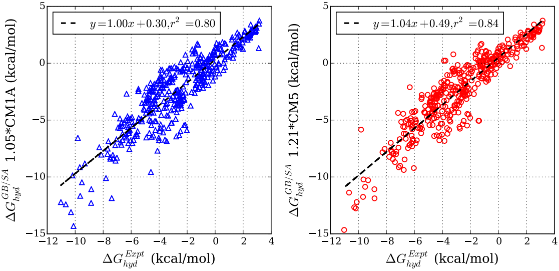

Figure 1.1: *Linear regression analysis has been performed for two sets of data and the resulting model is shown in the legends of each figure*

```
plt.xlabel(r'$\Delta G^{Expt}_{hyd}~$ (kcal/mol)',
     fontsize=16)
plt.grid()
fig.subplots_adjust(left = 0.15,hspace = .001)
fig.tight_layout()
plt.savefig('GBSA_comp.pdf')
```

Chapter 2

Heat Maps

Chapter 3

Barplots

Packages required to run this code

pandas for reading “Hvap.csv[1]" format

numpy for creating and manipulating vectors

matplotlib for making the plots

[1] *contains both the raw and devation data required for plot*

Table 3.1: *Data to be plotted using bar plots*

Molecules	OPLS	CM1A	CM5	Expt
Acetic acid	12.26	13.52	14.46	12.49
Acetone	7.23	7.74	8.92	7.48
Acetonitrile	7.57	7.63	9.76	8.01
Aniline	11.88	16.41	14.61	12.60
Benzonitrile	12.52	14.45	15.49	12.54
Cyclohexane	7.56	7.64	7.61	7.86
Diethylamine	7.68	7.54	7.46	7.48
Diethyl ether	6.90	7.01	7.22	6.56
N,N-dimethylacetamide	13.44	14.34	15.57	11.75
Ethanethiol	6.67	6.48	6.68	6.58
Ethanol	10.29	9.06	10.19	10.11
Furan	6.91	8.01	7.17	6.56
Hexane	7.54	7.48	7.34	7.54
Methanol	9.00	7.60	8.84	8.95
Methyl acetate	7.99	10.00	10.12	7.72
Nitroethane	9.78	14.16	11.72	9.94
N-methylacetamide	13.87	16.12	19.06	13.30
Phenol	14.58	14.63	14.30	13.82
Propylamine	7.90	8.93	7.23	7.47
Pyridine	9.76	11.16	11.16	9.61
Pyrrole	10.32	13.81	12.37	10.80
Tetrahydrofuran	7.52	7.66	8.08	7.61

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
hvap = pd.read_csv("Hvap.csv")
n_groups = len(hvap.D_OPLS)

opls = list(hvap.D_OPLS)
x_lab = list(hvap.Molecules)
cm5 = list(hvap.D_CM5)
cm1a = list(hvap.D_CM1A)
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.33
opacity = 0.5
rects1 = plt.bar(index, opls, bar_width,
                 alpha=opacity,color='r',label='OPLS')
rects2 = plt.bar(index + bar_width, cm5, bar_width,
                 alpha=opacity,color='g',label='1.27*CM5')
rects3 = plt.bar(index + 2 * bar_width, cm1a, bar_width,
                 alpha=opacity,color='b',label='1.14*CM1A')
plt.ylabel(r'$\Delta H_{vap}^{expt}-\Delta H_{vap}^{calc}~$ (kcal/mol)')
plt.xticks(index + bar_width, x_lab, rotation=90)
plt.grid()
plt.xlim(-0.5, n_groups + 0.5)
plt.legend(loc='lower left', ncol=3)
plt.tight_layout()
plt.savefig("Tesh_hvap.pdf")
```

Listing 3.1: *Bar plot of the data shown in Table above*

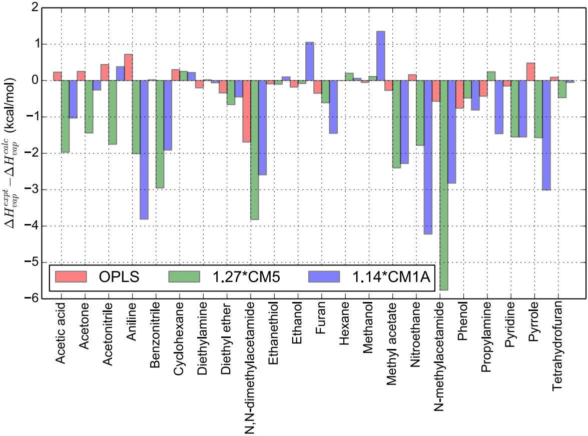

Figure 3.1: *Data in table above is plotted where instead of raw data, deviations from experiments for each method is plotted*

```
import matplotlib
matplotlib.use('Agg')
matplotlib.rc('font', family='serif')
from matplotlib import pylab
from pylab import rcParams
rcParams['figure.figsize'] = 11, 7
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.cm as cm
dat = pd.read_csv('all_cm5_dat.csv')
method=list(dat['Molecules'])
#################################
legend= (dat.columns.values)[1:]
#colors = cm.Greens(np.linspace(0, 1, len(legend)))
colors = cm.Spectral(np.linspace(0, 1, len(legend)))
index = np.arange(len(method))
bar_width = 1.0/len(legend)
opacity = 1.0
for i,c in zip(range(0,len(legend)),colors):
  plt.bar(index+bar_width*i,dat[legend[i]] , bar_width ,
                          alpha=opacity,
                          color=c,
                          label=legend[i])
plt.ylabel(r'$\Delta H_{vap}^{expt}-\Delta H_{vap}^{calc}~$ (kcal/mol)',fontsize=14)
plt.xticks(index + bar_width*len(legend)/2, method, rotation=90,fontsize=12)
plt.grid()
plt.xlim(-0.1, len(method) + 0.0)
#plt.legend(bbox_to_anchor=(1.0, 1.01),fontsize=9,loc=0,frameon=False)
plt.legend(loc='upper left', ncol=6,fontsize=10, frameon=False)
plt.tight_layout(rect=[0.0,0.01,0.99,1])
```

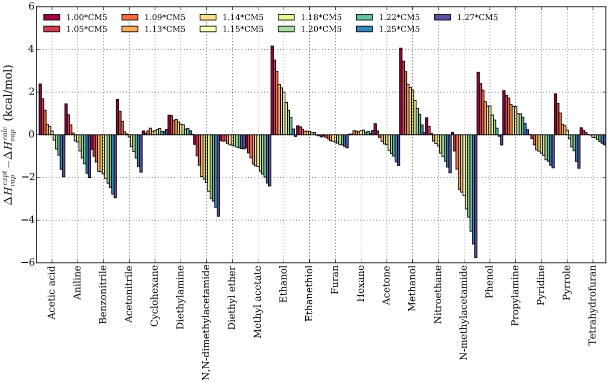

Figure 3.2: *Data in table above is plotted where instead of raw data, deviations from experiments for each method is plotted*

```
plt.savefig("Thh.pdf")
```

Listing 3.2: *Barplot liquid properties using CM5 charges with different scale factors*

```
import matplotlib
matplotlib.use('Agg')
matplotlib.rc('font', family='serif')
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.cm as cm
from matplotlib.ticker import FuncFormatter
hvap = pd.read_csv('Hvap.csv')
den = pd.read_csv('Den.csv')
hvap.drop(hvap.columns[[1,2,3,4,5,7]], axis=1, inplace=True)
den.drop(den.columns[[1,2,3,4,5,7]], axis=1, inplace=True)
m1=list(hvap['Molecules'])
m2=list(den['Molecules'])
################################
def millions(x, pos):
    'The two args are the value and tick position'
    return '%2.2f' % (x)
formatter = FuncFormatter(millions)
################################
l1= (hvap.columns.values)[1:]
l2= (den.columns.values)[1:]
colors = cm.Spectral(np.linspace(0, 1, len(l1)))
fig,(ax1, ax2) = plt.subplots(2, sharex=True)
index = np.arange(len(m1))
bar_width = 1.0/len(l1)
opacity = 1.0
patterns = [ "*", "o", "."]
for i,c in zip(range(0,len(l1)),colors):
  ax1.bar(index+bar_width*i,hvap[l1[i]] , bar_width,
                   alpha=opacity,
                   color=c,
       hatch=patterns[i],
                   label=l1[i][2:])
```

```
ax1.legend(fontsize=10,loc=9, bbox_to_anchor=(0.5, 1.2),ncol=3,frameon=False)
ax1.set_ylabel(r'$\Delta H_{vap}^{expt}-\Delta H_{vap}^{calc}~$ (kcal/mol)')
ax1.yaxis.set_major_formatter(formatter)
ax1.yaxis.set_ticks(np.arange(-6,3,2))
for i,c in zip(range(0,len(l2)),colors):
  ax2.bar(index+bar_width*i,den[l1[i]] , bar_width,
                   alpha=opacity,
                   color=c,
       hatch=patterns[i],
                   label=l2[i][2:])
ax2.set_ylabel(r'$\Delta \rho^{expt}-\Delta \rho^{calc}~$ (g/cc)')
ax2.yaxis.set_ticks(np.arange(-0.1,0.08,0.04))
plt.xticks(index + bar_width*len(l2)/2, m2, rotation=90,fontsize=10)
ax1.xaxis.grid()
ax2.xaxis.grid()
ax1.yaxis.grid()
ax2.yaxis.grid()
ax1.set_xlim(-0.1, len(m1) + 0.0)
ax2.set_xlim(-0.1, len(m2) + 0.0)
plt.tight_layout(rect=[0.0,0.01,0.89,0.99])
plt.savefig("Multi_bar.pdf")
```

Listing 3.3: *Multiple bar plots in matplotlib*

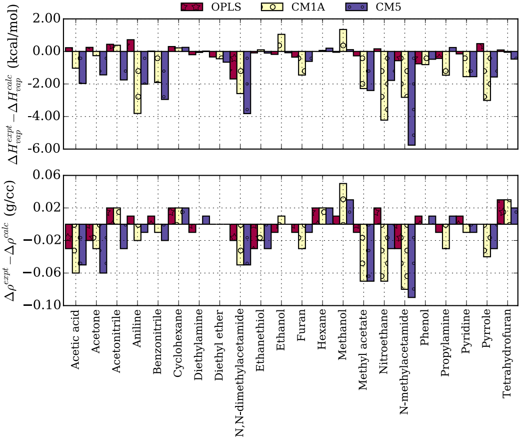

Figure 3.3: *Data in table above is plotted where instead of raw data, deviations from experiments for each method is plotted*

```python
import matplotlib
matplotlib.use('Agg')
matplotlib.rc('font', family='serif')
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.cm as cm
import seaborn as sns
sns.set()
from matplotlib.ticker import FuncFormatter
hvap = pd.read_csv('Hvap.csv')
den = pd.read_csv('Den.csv')
d_std=pd.read_csv('STD_Den.csv')
h_std=pd.read_csv('STD_Hvap.csv')
h_std['CM5']=h_std.CM5_127
hvap.drop(hvap.columns[[1,2,3,4,5,7]], axis=1, inplace=True)
den.drop(den.columns[[1,2,3,4,5,7]], axis=1, inplace=True)
m1=list(hvap['Molecules'])
m2=list(den['Molecules'])
################################
def millions(x, pos):
    'The two args are the value and tick position'
    return '%2.2f' % (x)
formatter = FuncFormatter(millions)
################################
l1= (hvap.columns.values)[1:]
l2= (den.columns.values)[1:]
colors = cm.rainbow(np.linspace(0, 1, len(l1)))
fig,(ax1, ax2) = plt.subplots(2, sharex=True)
index = np.arange(len(m1))
bar_width = 1.0/len(l1)
opacity = 1.0
patterns = [ "*", "o", "."]
for i,c in zip(range(0,len(l1)),colors):
  ax1.bar(index+bar_width*i,hvap[l1[i]] , bar_width,
                   alpha=opacity,
                   facecolor=c,
       edgecolor = c,
#      hatch=patterns[i],
                   label=l1[i][2:],yerr=h_std[l1[i][2:]],ecolor='k',capsize=1)
ax1.legend(fontsize=10,loc=9, bbox_to_anchor=(0.5,1.2),ncol=3,frameon=False)
ax1.set_ylabel(r'$\Delta H_{vap}^{expt}-\Delta H_{vap}^{calc}~$ (kcal/mol)')
ax1.yaxis.set_major_formatter(formatter)
ax1.yaxis.set_ticks(np.arange(-6,3,2))
for i,c in zip(range(0,len(l2)),colors):
  ax2.bar(index+bar_width*i,den[l1[i]] , bar_width,
                   alpha=opacity,
                   facecolor=c,
       edgecolor=c,
       #hatch=patterns[i],
                   label=l2[i][2:],yerr=d_std[l1[i][2:]],ecolor='k',capsize=1)
ax2.set_ylabel(r'$\Delta \rho^{expt}-\Delta \rho^{calc}~$ (g/cc)')
ax2.yaxis.set_ticks(np.arange(-0.1,0.08,0.04))
plt.xticks(index + bar_width*len(l2)/2, m2, rotation=90,fontsize=10)
#ax1.xaxis.grid()
#ax2.xaxis.grid()
#ax1.yaxis.grid()
#ax2.yaxis.grid()
ax1.set_xlim(-0.1, len(m1) + 0.0)
ax2.set_xlim(-0.1, len(m2) + 0.0)
plt.tight_layout(rect=[0.0,0.01,0.89,0.99])
plt.savefig("Ers_Multi_bar.pdf")
```

Listing 3.4: *Multiple bar plots in matplotlib*

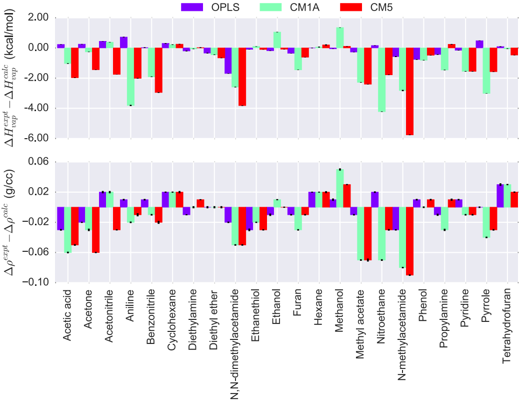

Figure 3.4: *Data in table above is plotted where instead of raw data, deviations from experiments for each method is plotted*

Part II

R recipies