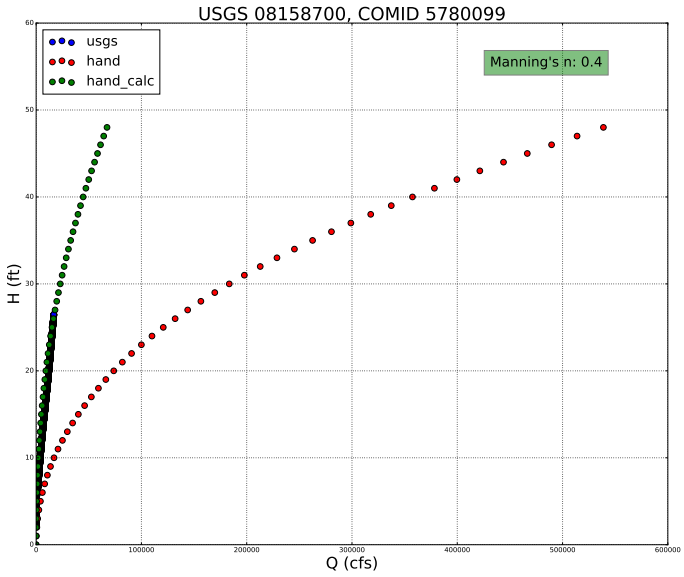
Homework 3 – Problem 4

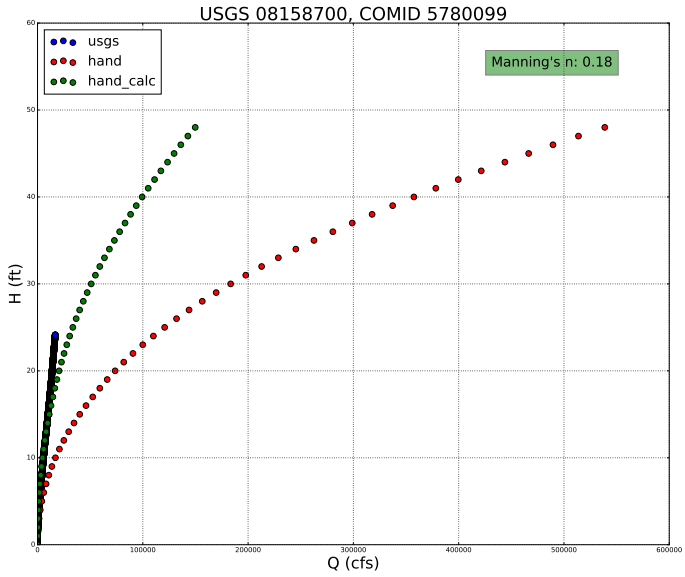
This project has been improved since the previous deliverable in the following ways:

1. Using the hydraulic properties computed with the Height Above Nearest Drainage (HAND) method, I have back-calculated the average manning's n for each United States Geological Survey (USGS) rating curve in the Onion Creek, TX watershed.
2. For each USGS gage station, I have found the first instance of flow (ie. flow of 0.01 cfs) and mapped the associated stage-height as the channel bottom-depth.
3. Using the back-calculated manning's n, I have tilted the HAND rating curve for an approximate fit to the USGS rating curve with a new bottom-depth.

Examples of a few rating curves, as compared to their previously "manually" optimized manning's n values, are included in figures 1, 2 and 3. Note that these graphs are not strictly comparable, due to the fact that changes have been made to both the USGS curve (bottom-depth shift) and the HAND curve (different manning's n used). Furthermore, it is important to be aware that an average manning's n (from the various manning's n values calculated from the USGS data at each 1-foot interval) was applied to the HAND rating curve, and this averaging is likely not the best approach for this application; possible improvements could be made by using a root-mean-square-error (RMSE) approach for assessing the optimal manning's n with this back-calculating approach. Finally, it is important to be aware that, though these automatic adjustments seem to fit the USGS rating curves fairly well, this will not be a sustainable model for large-scale rating curve optimization as it relies entirely on USGS rating curves for determining which manning's n to use for the HAND rating curve snapping.

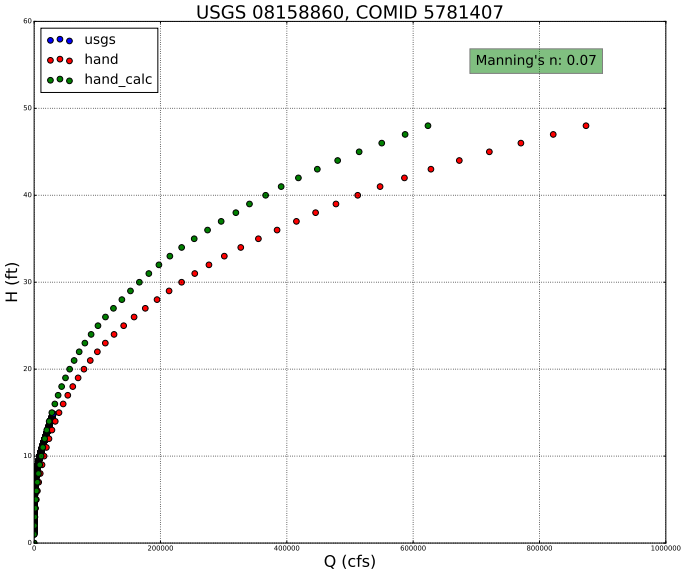


(a) Manually-Adjusted Rating Curve

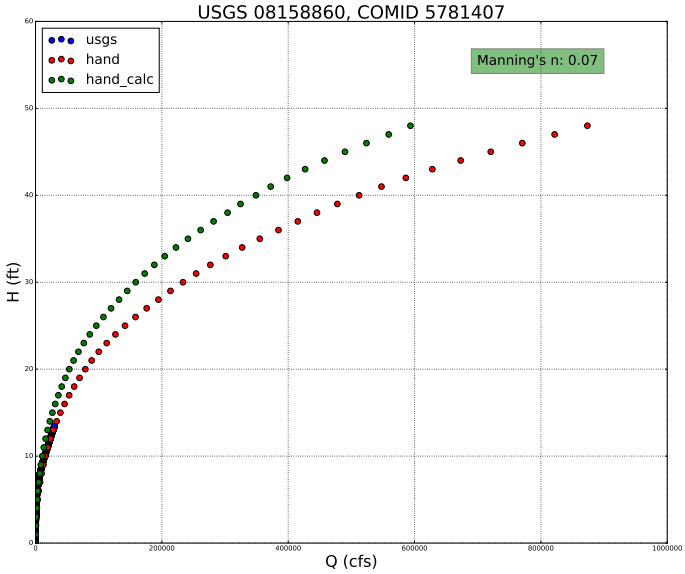


(b) Automatically-Adjusted Rating Curve

Figure 1: Manual vs. Automatic Adjustments for COMID 5780099 Rating Curve

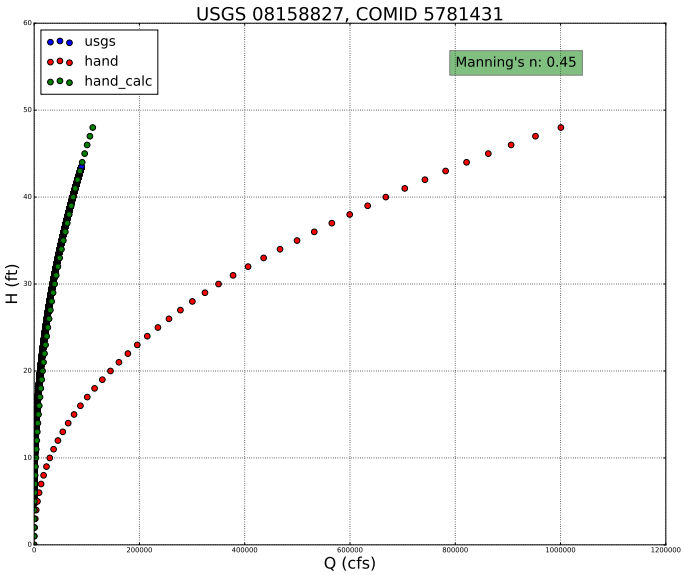


(a) Manually-Adjusted Rating Curve

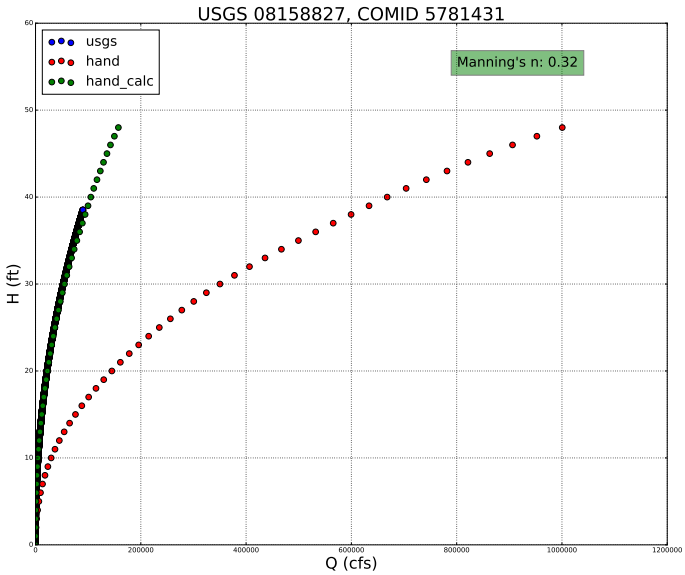


(b) Automatically-Adjusted Rating Curve

Figure 2: Manual vs. Automatic Adjustments for COMID 5781407 Rating Curve



(a) Manually-Adjusted Rating Curve



(b) Audomatically-Adjusted Rating Curve

Figure 3: Manual vs. Automatic Adjustments for COMID 5781431 Rating Curve

As stated previously, the intent of this research is to programmatically fit a HAND rating curve to its corresponding USGS rating curve. That said, manipulation of manning's n is not strictly acceptable, seeing as it is a physical parameter ideally representative of the stream reach in question (see figure 4 below). Changes to manning's n are effectively a claim that physical streambed material is changing, which is unreasonable. Instead, this project has changed to focus on understanding the statistical differences between "theoretical" hydrologic properties (specifically wetted area (A_w), hydraulic radius(H_R), and slope (S_0)) and their "actual" counterparts, which will likely be retrieved from HEC-RAS models with detailed cross-section descriptions. Once these differences are understood, corrections can be approximated for each parameter that can, ideally, be used to improve the fit of each HAND rating curve.

Roughness Coefficients (Manning's "n") for Channel and Pipe Flow

Reach Description	**n**
Natural stream, clean, straight, no rifts or pools	0.03
Natural stream, clean, winding, some pools or shoals	0.04
Natural stream, winding, pools, shoals, stoney with some weeds	0.05
Natural stream, sluggish deep pools and weeds	0.07
Natural stream or swale, very weedy or with timber underbrush	0.10

Figure 4: Acceptable Manning's n values, retrieved from: http://www.shippensburgtownship.com/2010-01/media/development/Table%20B-4.pdf

It is my current understanding that the differences between the HAND and USGS rating curves are primarily related to Digital Elevation Model (DEM) imperfections. DEMs are generally constructed with the use of remote sensing technologies, which frequently fail to penetrate through water surfaces (and, Paola, feel free to correct me if I'm wrong; you know a lot more about this than me). This failure to penetrate means that, at the time a land surface is remotely sensed, the lowest elevations for stream reaches will actually represent the stage height at that time, rather than the stream's bottom depth. Because of this discrepancy, and because HAND rasters are created from DEMs, the A_w and and wetted perimeter (which together influence the H_R) computed from the HAND rasters may likely be incorrect (see figure 5); these discrepencies may further extend to the S_0, introducing another potentially faulty error to the HAND raster creation.

No work has yet been done to access the impact of these discrepencies, but the intent of this project going forward will be to statistically analyze how different these "theoretical" (from HAND) hydraulic properties may be from their "actual" (HEC-RAS cross-sections) counterparts; note that HEC-RAS cross-section information for the Onion Creek watershed has already been retrieved (from Xing Zheng's Master's thesis work). I'm not sure yet how to approach this, but the first step will be to understand the influence of these parameters. So, for example, I will take one stream reach and gradually increase the bottom-depth shift (effectively making the HAND cross-section's bottom-depth deeper) and re-compute the resulting cross-section's A_w, H_R, and S_0 (this one may be a bit tricky). Once I have computed these characteristics, I will re-compute the manning's n and plot the shifted HAND rating curve, comparing it to the USGS rating curve (using some form of RMSE analysis). Ultimately, I will automate this process and iterate over all possible bottom-depth shifts to arrive at the optimal bottom-depth shift. Once this optimal shift is computed for a single stream reach, I will attempt to apply my methodology to the remaining stream reaches in the Onion Creek watershed. The end goal for this approach would probably be to relate each bottom-depth shift to the average flowrate of each stream reach in some generalizable way, such that the methodology would be extendable to un-gaged stream reaches.

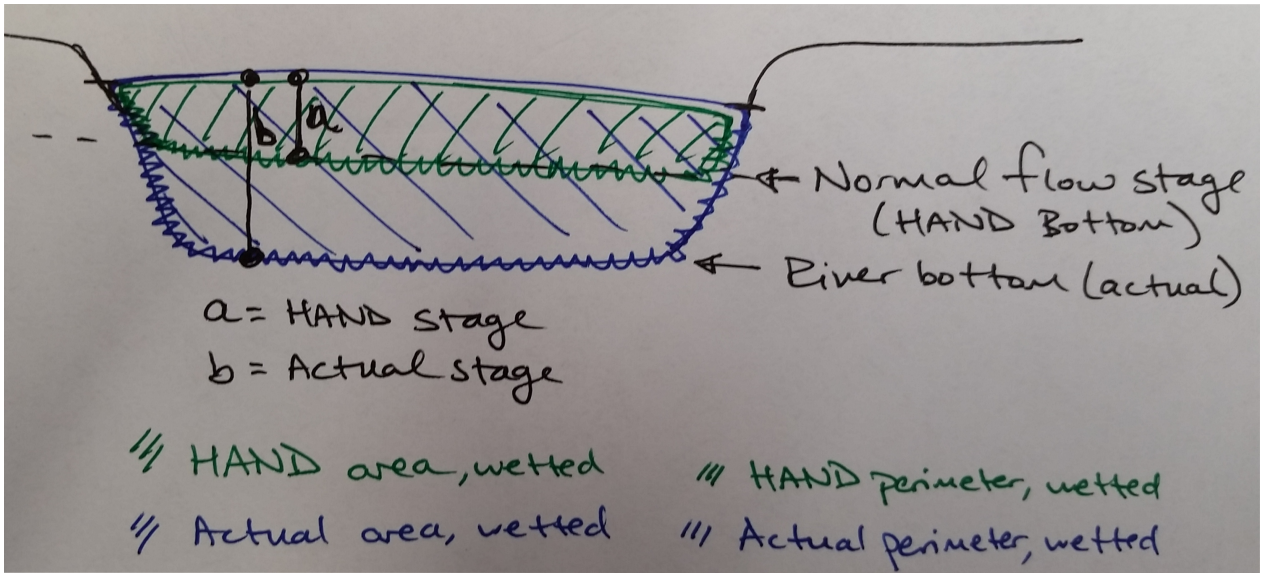


Figure 5: Sketch of variable wetted area (A_w) and wetted perimeter (P_w)

If the bottom-depth shift can not be related to the average flowrate in a statistically significant manner, there may be opportunities for me to collaborate with Shahab Afshari at New York University to relate these bottom-depth shifts to the National Land Cover Dataset (NLCD). This would be a fairly complex procedure, and will likely not be completed in time for this course, but I would like very much to hear any feedback you have on that idea. Shahab is trying to relate his at-a-station hydraulic geometry computations (for approximately 5,000 USGS gage stations) to the NLCD, so we may be able to collaborate on extracting NLCD data within a buffer around each stream reach.

See next page for Python script used to create curves. The major changes since the last iteration are the shift to USGS rating curves (by setting the first instance of 0.1 cfs flow as the bottom-depth) and the back-calculating of manning's n values using HAND hydraulic properties (A_w, H_R, and S_0) and the shifted USGS rating curve.

Listing 1: Class CompareRC defined in ratingcurves-onionck.py module to quickly collect, extract, calculate, and display the USGS/HAND rating curves shown in Figures 2 through 4.

```
class CompareRC:

  def __init__(self,comid,idlookup,handrc,handrcidx,handnetcdf,handnetcdfidx):
    self.comid = comid
    self.idlookup = idlookup
    self.handnetcdf = handnetcdf
    self.handrc = handrc
    self.usgsid = self.idlookup.loc[self.idlookup['FLComID'] == self.comid]['SOURCE_FEA'].values[0]
    self.usgsrc = 'http://waterdata.usgs.gov/nwisweb/get_ratings?file_type=exsa&site_no={0}'
    self.handnetcdfidx = handnetcdfidx.loc[handnetcdfidx['comid'] == self.comid]['index'].values[0]
    self.handrcidx = handrcidx.loc[handrcidx['comid'] == self.comid]['index'].values[0]

  def get_values(self):
    if self.get_usgsrc() == 0:
      return
    self.get_usgsrc() # Fetch self.usgsq and self.usgsh
    self.get_handrc() # Fetch self.handq and self.handh
    self.get_handnetcdf() # Fetch self.handarea, self.handrad, self.handslope, and self.handstage

  def get_usgsrc(self):
    """ Initializes self.usgsq and self.usgsh """
    urlfile = urllib.urlopen(self.usgsrc.format(str(self.usgsid)))
    urllines = urlfile.readlines()
    findData = False
    usgsq = scipy.array([])
    usgsh = scipy.array([])
    for j in range(len(urllines)):
      line = urllines[j]
      if not findData and not re.search('[a-zA-Z]',line): # No letters
        findData = True
      if findData and float(line.split('\t')[2]) >= 1: # Remove data where Q < 1
        current = line.split('\t')
        usgsq = scipy.append( usgsq, float(current[2]) )
        # apply shift in [1] to stage height
        usgsh = scipy.append( usgsh, (float(current[0]) - float(current[1])) )
    try:
      self.bottomdepth = usgsh[0] # Set first Q = 1.0 equal to bottom depth
    except IndexError:
      print 'USGS Rating Curve does not exist for USGSID {0}'.format(str(self.usgsid))
      return 0
    self.usgsh = (usgsh - self.bottomdepth) # Normalize usgsh over bottom depth
```

```
42      self.usgsq = usgsq
43
44    def get_handrc(self):
45      """ Initializes self.handq (cfs) and self.handh (ft) """
46      handq = self.handrc.variables['Q_cfs']
47      handh = self.handrc.variables['H_ft']
48      handc = self.handrc.variables['COMID']
49      if handc[self.handrcidx] == self.comid:
50        self.handq = handq[self.handrcidx]
51      self.handh = handh
52
53    def get_handnetcdf(self):
54      """Initializes self.handarea (sqmeters), self.handrad (m),
55        self.handslope (-), and self.handstage (ft) """
56      handc = self.handnetcdf.variables['COMID']
57      handslope = self.handnetcdf.variables['Slope']
58      handarea = self.handnetcdf.variables['WetArea']
59      handrad = self.handnetcdf.variables['HydraulicRadius']
60      handstage = self.handnetcdf.variables['StageHeight']
61      if handc[self.handnetcdfidx] == self.comid:
62        self.handarea = handarea[self.handnetcdfidx]*10.7639 # Convert sqm to
     sqft
63        self.handrad = handrad[self.handnetcdfidx]*3.28084 # Convert m to ft
64        self.handslope = handslope[self.handnetcdfidx]
65      handstagenew = scipy.array([])
66      for i in handstage:
67        handstagenew = scipy.append(handstagenew, handstage)
68      self.handstage = handstagenew
69      self.handstage = self.handstage[:49]*3.28084 # Convert m to ft
70      self.handstage = scipy.rint(self.handstage) # Round to nearest int, to
     clean up conversion
71
72    def mannings_q(self,area,hydrad,slope,en):
73      """ Calculates discharge from manning's roughness using Wet Area = self.
     handarea,
74        Hydraulic Radius = self.handrad, and Slope = self.handslope """
75      return 1.49*area*scipy.power(hydrad,(2/3.0))*sqrt(slope)/en
76
77    def mannings_n(self,area,hydrad,slope,disch):
78      """ Calculates manning's roughness from discharge using Wet Area = self.
     handarea,
79        Hydraulic Radius = self.handrad, and Slope = self.handslope """
80      return 1.49*area*scipy.power(hydrad,(2/3.0))*sqrt(slope)/disch
81
82    def calc_handrc(self,en):
83      """ Initializes self.handdisch (cfs)"""
84      handdisch = self.mannings_q(area=self.handarea,hydrad=self.handrad,slope=
     self.handslope,en=en)
85      self.handdisch = handdisch[:49]
86
87    def convert_to_english(self):
```

```
88      pass
89
90    def data_to_csv(self):
91      import itertools
92      import csv
93      if self.get_usgsrc() == 0:
94        return
95      self.get_values() # Fetch usgsq,usgsh,handq,handh,handarea,handrad,
      handslope, handstage
96      self.get_usgs_n() # Fetch self.usgsroughness
97      self.calc_handrc(self.usgsroughness) # Fetch self.handdisch and self.
      handstage
98      info = [self.comid,self.usgsid,self.handslope]
99      rows = itertools.izip_longest(info,self.handstage,self.handarea, \
100       self.handrad,self.usgsh,self.usgsq,fillvalue='')
101     with open('onionck/results/newcsvdata/{0}_data.csv'.format(self.comid), 'w
      ') as f:
102       csv.writer(f).writerows(rows)
103     f.close()
104
105   def get_usgs_n(self):
106     if self.get_usgsrc() == 0:
107       return
108     self.get_values() # Fetch usgsq,usgsh,handq,handh,handarea,handrad,
      handslope, handstage
109
110     # Find indices for integer stageheight values in usgsh, and apply to usgsq
111     usgsidx = scipy.where(scipy.equal(scipy.mod(self.usgsh,1),0)) # Find
      indices of integer values in usgsh
112     usgsh = self.usgsh[usgsidx]
113     usgsq = self.usgsq[usgsidx]
114
115     # Find indices where usgsh[usgsidx] occur in handstage, and apply to
      handarea and handrad
116     handidx = scipy.where(scipy.in1d(self.handstage,usgsh))
117     area = self.handarea[handidx]
118     hydrad = self.handrad[handidx]
119
120     # Remove usgsq values for duplicate usgsh heights (keep first instance
      only)
121     if usgsh.shape != area.shape:
122       for i in range(usgsh.shape[0]):
123         if i == 0: pass
124         elif usgsh[i] == usgsh[i-1]:
125           usgsq = scipy.delete(usgsq,i)
126
127     # Calculate average manning's n after converting discharge units
128     disch = usgsq #*0.0283168 # Convert cfs to cms
129     self.usgsroughness_array = self.mannings_n(area=area,hydrad=hydrad,slope=
      self.handslope,disch=disch)
130     self.usgsroughness = scipy.average(self.usgsroughness_array)
```

```
131     print 'Average roughness: {0:.2f}'.format(self.usgsroughness)
132
133   def plot_rcs(self):
134     if self.get_usgsrc() == 0:
135       return
136     self.get_values() # Fetch usgsq,usgsh,handq,handh,handarea,handrad,
      handslope, handstage
137     self.get_usgs_n() # Fetch self.usgsroughness
138     self.calc_handrc(self.usgsroughness) # Fetch self.handdisch
139     fig, ax = plt.subplots()
140     fig.set_size_inches(20,16, forward=True)
141     ax.scatter(self.usgsq,self.usgsh,c='b',s=100,label='usgs')
142     ax.scatter(self.handq,self.handh,c='r',s=100,label='hand')
143     ax.scatter(self.handdisch,self.handstage,c='g',s=100,label='hand_calc')
144     plt.gca().set_xlim(left=0)
145     plt.gca().set_ylim(bottom=0)
146     ax.set_title('USGS {0}, COMID {1}'.format(str(self.usgsid),str(self.comid)
      ),fontsize=32)
147     plt.xlabel('Q (cfs)',fontsize=28)
148     plt.ylabel('H (ft)',fontsize=28)
149     ax.text((self.handq[-1]*0.8),55,"Manning's n: {0:.2f}".format(self.
      usgsroughness),horizontalalignment='left',
150       fontsize=24,bbox={'facecolor':'green', 'alpha':0.5, 'pad':10})
151     plt.legend(loc='upper left',fontsize=24)
152     plt.grid()
153     plt.show()
154     fig.savefig('onionck/results/autoresults/rc_comid_{0}.pdf'.format(self.
      comid))
155
156 if __name__ == '__main__':
157
158   idlookup = pandas.read_csv('streamgages.csv',usecols=['SOURCE_FEA','FLComID'
      ])
159   idlookup['SOURCE_FEA'] = idlookup['SOURCE_FEA'].apply(lambda x: '{0:0>8}'.
      format(x))
160   handrc = Dataset('handratingcurves.nc', 'r')
161   handrcidx = pandas.read_csv('handrc_idx.csv')
162   handnetcdfidx = pandas.read_csv('handnc_idx.csv')
163   handnetcdf = Dataset('onionck/OnionCreek.nc', 'r')
164
165   for i in range(len(idlookup)):
166     comid = idlookup['FLComID'][i]
167     rcs = CompareRC(comid,idlookup,handrc,handrcidx,handnetcdf,handnetcdfidx)
168     rcs.plot_rcs()
169     # rcs.data_to_csv()
```