**Supplementary Material “Morphometric Analysis of Pluto’s Impact Craters ”:**

1. **Python Code:**

For the code processing, two initial global rasters (DEM and slope map) are required. Firstly, the user should identify the crater for analyzing purposes. With that, the user should only create a circular polygon that approximately involves the crater. From the central point of the polygon, the code creates the profile lines and obtains the values ​​of slope and height at each pixel. The program then creates two files: maximum elevation (EM) and control point (PC). In this work, we use a method to outline the impact craters similar to the one used by Geiger et al. (2013). The main difference between Geiger et al. (2013) method and ours is that they use as the main way to find the crater rim the point of slope break, but as Pluto's surface is much chaotic this method created more errors. EM points are then selected from the pixel with the highest Z value (altitude) in each profile. The EM points, composed by the pixel with the highest altitude Z value of each profile, are selected within a radius equivalent to 1/(4.5) of the radius of the polygon generated by the user, calculated from its edge. This is useful for eliminating points near the center of the craters and also integrating an area beyond the user polygon. The PC, on the other hand, is the file that contains all the pixels of each profile. The code, from a starting point of EM, looks in the next profile (clockwise) for another EM point that is within a radius stipulated by the user (we use 1000 meters in this work). The points to be searched for in the next profile follow the order of priority of: EM> PC. If the code does not find the highest priority point (EM), it looks for the lowest priority point (PC) that is closest to the EM point in its same profile. The code finishes delineating the crater as soon as all profiles are analyzed.

After delineating the edges of the impact structure, the code extracts the morphometric values. The average of the diameter is calculated, since the delineation line is not a perfect circle, from the value of each profile that extends from the centroid to the edges of the impact crater. The depth of the structure is calculated by: (point of lowest elevation of the base of the crater) – (value of the average height of the crater wall), thus obtaining an absolute depth value in meters. We used the average wall height in the depth measurement because it seems to reduce the influence of the pre-impact topography in the analyzes. The standard deviation of the height of the crater wall can be used to help the interpretation of the rate of modification of structures (well-preserved craters should have little variation in the edge) and the influence of the terrain in the analyzes. To obtain the average slope value of the internal walls of impact craters, it is necessary to separate the boundary between the base and the internal wall of the structure. Otherwise, the analysis would be influenced by the low slope of the structure base. For this purpose, the code eliminates from the analysis around ⅛ of the total radius of the impact crater from the center of the crater. This generates a new polygon in order to prevent the features of the central peak from interfering in the analysis. Then, the code separates the base of the crater from the inner wall, calculating ⅕ of the total depth value of each profile calculated in the new polygon. The height values ​​(Z) of the pixels present in each new profile are analyzed, and if they have a value greater than ⅕ of the total depth, the points are selected representing the beginning of the wall. With the selected points, a new file is created that mostly covers only the inner wall of the crater. From the newly generated file, the total slope is averaged across all profiles.

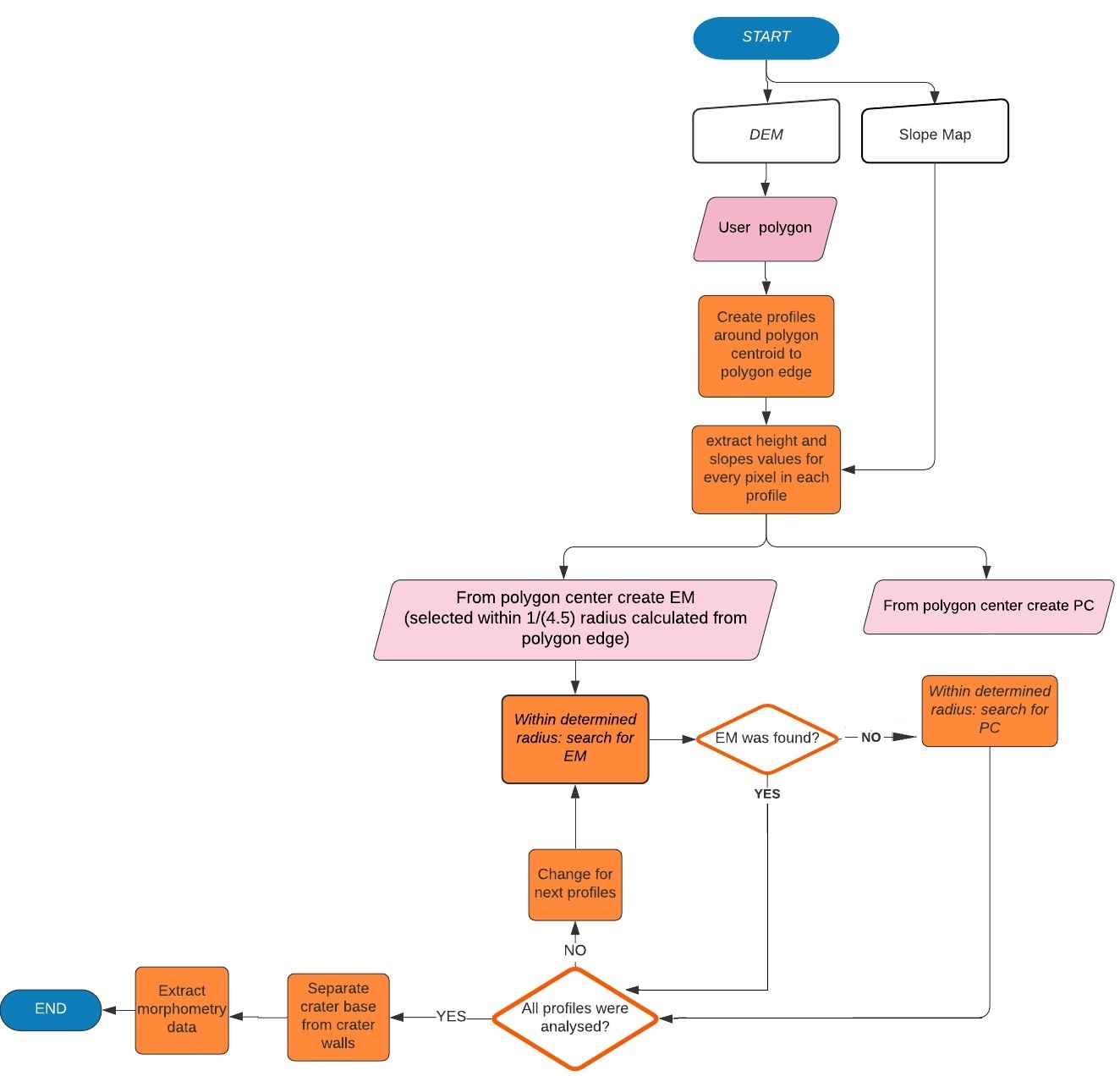


Figure S1. Fluxogram summarizing the Python code logic.

**Appendix S1: Python Code**

|  |
| --- |
| Crater = '' output = '' slopeproj=  demproj =  dem =  slope =   csv\_file =  output\_pc =  output\_me =  output\_me\_select =  output\_pc\_select =  output\_line =  output\_merge = output\_base =  output\_merge\_baserim =  output\_line\_slope =  output\_alllines =  output\_poly =  #------- Reprojection ---------------- point = iface.activeLayer() feat = point  fieldcalc = processing.run("qgis:fieldcalculator", {'INPUT':feat,'FIELD\_NAME':'x','FIELD\_TYPE':0,'FIELD\_LENGTH':10,'FIELD\_PRECISION':3,'NEW\_FIELD':**True**,'FORMULA':'to\_dm(x(transform($geometry,\'USER:100000\',\'EPSG:104969\')), \'x\', 2)','OUTPUT':'memory:'}) fieldcalc2 = processing.run("qgis:fieldcalculator", {'INPUT':fieldcalc['OUTPUT'],'FIELD\_NAME':'y','FIELD\_TYPE':0,'FIELD\_LENGTH':10,'FIELD\_PRECISION':3,'NEW\_FIELD':**True**,'FORMULA':'to\_dm(y(transform($geometry,\'USER:100000\',\'EPSG:104969\')), \'x\', 2)','OUTPUT':'memory:'})  layer = fieldcalc2['OUTPUT'] featfeat=layer.getFeatures()#get first feature list\_coord =[] **for** featurecent **in** featfeat:  list\_coord.append(featurecent['x'])  list\_coord.append(featurecent['y'])  my\_crs = QgsCoordinateReferenceSystem() my\_crs.createFromProj4("+proj=stere +lat\_0="+list\_coord[1]+" +lon\_0="+list\_coord[0]+" +x\_0=0 +y\_0=0 +a=1188300 +b=1188300 +units=m +no\_defs") my\_crs.saveAsUserCrs("crater") QgsProject.instance().setCrs(my\_crs)  buff =processing.run("native:buffer", {'INPUT':feat,'DISTANCE':100000,'SEGMENTS':5,'END\_CAP\_STYLE':0,'JOIN\_STYLE':0,'MITER\_LIMIT':2,'DISSOLVE':**False**,'OUTPUT':'TEMPORARY\_OUTPUT'}) clip=processing.run("gdal:cliprasterbymasklayer", {'INPUT':dem,'MASK':buff['OUTPUT'],'SOURCE\_CRS':**None**,'TARGET\_CRS':**None**,'NODATA':**None**,'ALPHA\_BAND':**False**,'CROP\_TO\_CUTLINE':**True**,'KEEP\_RESOLUTION':**False**,'SET\_RESOLUTION':**False**,'X\_RESOLUTION':**None**,'Y\_RESOLUTION':**None**,'MULTITHREADING':**False**,'OPTIONS':'','DATA\_TYPE':0,'OUTPUT':'TEMPORARY\_OUTPUT'}) clip2=processing.run("gdal:cliprasterbymasklayer", {'INPUT':slope,'MASK':buff['OUTPUT'],'SOURCE\_CRS':**None**,'TARGET\_CRS':**None**,'NODATA':**None**,'ALPHA\_BAND':**False**,'CROP\_TO\_CUTLINE':**True**,'KEEP\_RESOLUTION':**False**,'SET\_RESOLUTION':**False**,'X\_RESOLUTION':**None**,'Y\_RESOLUTION':**None**,'MULTITHREADING':**False**,'OPTIONS':'','DATA\_TYPE':0,'OUTPUT':'TEMPORARY\_OUTPUT'})  layer1 = QgsRasterLayer(clip['OUTPUT']) layer2 = QgsRasterLayer(clip2['OUTPUT'])   crsup = iface.mapCanvas().mapSettings().destinationCrs().authid() warp1 = processing.run("gdal:warpreproject", {'INPUT':layer1,'SOURCE\_CRS':QgsCoordinateReferenceSystem('USER:100000'),'TARGET\_CRS':QgsCoordinateReferenceSystem(crsup),'RESAMPLING':0,'NODATA':**None**,'TARGET\_RESOLUTION':**None**,'OPTIONS':'','DATA\_TYPE':0,'TARGET\_EXTENT':**None**,'TARGET\_EXTENT\_CRS':**None**,'MULTITHREADING':**False**,'EXTRA':'','OUTPUT':output\_dem}) warp2 = processing.run("gdal:warpreproject", {'INPUT':layer2,'SOURCE\_CRS':QgsCoordinateReferenceSystem('USER:100000'),'TARGET\_CRS':QgsCoordinateReferenceSystem(crsup),'RESAMPLING':0,'NODATA':**None**,'TARGET\_RESOLUTION':**None**,'OPTIONS':'','DATA\_TYPE':0,'TARGET\_EXTENT':**None**,'TARGET\_EXTENT\_CRS':**None**,'MULTITHREADING':**False**,'EXTRA':'','OUTPUT':output\_slope})  #---------------------Delineation ------------------------------------------------------ poly = iface.activeLayer()  save\_poly = QgsVectorFileWriter.writeAsVectorFormat(poly, output\_poly , "UTF-8", poly.crs(), "ESRI Shapefile")  list\_data = [] list\_data2 = [list\_data] list\_data.append(''+crater+'') crs = iface.mapCanvas().mapSettings().destinationCrs().authid() fieldcalculator1 = processing.run("qgis:fieldcalculator",  {'INPUT': poly, 'FIELD\_NAME': '1.3-field', 'FIELD\_TYPE': 0, 'FIELD\_LENGTH': 10, 'FIELD\_PRECISION': 3,  'NEW\_FIELD': **True**,  'FORMULA': '((length(shortest\_line(centroid($geometry),boundary($geometry)))))\*1.3',  'OUTPUT': 'memory:'}) fieldcalculator2 = processing.run("qgis:fieldcalculator",  {'INPUT': poly, 'FIELD\_NAME': '1-4\_raio', 'FIELD\_TYPE': 0, 'FIELD\_LENGTH': 10, 'FIELD\_PRECISION': 3,  'NEW\_FIELD': **True**,  'FORMULA': '((length(shortest\_line(centroid($geometry),boundary($geometry)))))/4.5',  'OUTPUT': 'memory:'}) fieldcalculator3 = processing.run("qgis:fieldcalculator",  {'INPUT': poly, 'FIELD\_NAME': '8\_raio', 'FIELD\_TYPE': 0, 'FIELD\_LENGTH': 10, 'FIELD\_PRECISION': 3,  'NEW\_FIELD': **True**,  'FORMULA': '((length(shortest\_line(centroid($geometry),boundary($geometry)))))/8',  'OUTPUT': 'memory:'}) polytoline = processing.run("native:polygonstolines", {'INPUT': fieldcaculator2['OUTPUT'], 'OUTPUT': 'TEMPORARY\_OUTPUT'}) centroid = processing.run("saga:polygoncentroids",  {'POLYGONS': fieldcalulator1['OUTPUT'], 'METHOD ': **True**, 'CENTROIDS': 'TEMPORARY\_OUTPUT'}) buffervar = processing.run("saga:variabledistancebuffer",  {'SHAPES': centroid['CENTROIDS'], 'DIST\_FIELD': '1.3-field', 'DIST\_SCALE': 1, 'NZONES': 1, 'DARC': 5,  'DISSOLVE ': **True**, 'POLY\_INNER ': **False**, 'BUFFER': 'TEMPORARY\_OUTPUT'}) polytoline2 = processing.run("native:polygonstolines", {'INPUT': c['BUFFER'], 'OUTPUT': 'TEMPORARY\_OUTPUT'}) desify = processing.run("native:densifygeometriesgivenaninterval",  {'INPUT': polytoline2['OUTPUT'], 'INTERVAL': 600, 'OUTPUT': 'TEMPORARY\_OUTPUT'}) vertice = processing.run("native:extractvertices", {'INPUT': densify['OUTPUT'], 'OUTPUT': 'TEMPORARY\_OUTPUT'}) geometry = processing.run("qgis:exportaddgeometrycolumns",  {'INPUT': vertice['OUTPUT'], 'CALC\_METHOD': 0, 'OUTPUT': 'TEMPORARY\_OUTPUT'}) geometry2 = processing.run("qgis:exportaddgeometrycolumns",  {'INPUT': centroid['CENTROIDS'], 'CALC\_METHOD': 0, 'OUTPUT': 'TEMPORARY\_OUTPUT'}) fieldcalculator4 = processing.run("qgis:fieldcalculator",  {'INPUT': geometry2['OUTPUT'], 'FIELD\_NAME': 'ID', 'FIELD\_TYPE': 0, 'FIELD\_LENGTH': 10,  'FIELD\_PRECISION': 3, 'NEW\_FIELD': **True**, 'FORMULA': '0', 'OUTPUT': 'memory:'}) join = processing.run("native:joinattributestable",  {'INPUT': geometry['OUTPUT'], 'FIELD': 'ID', 'INPUT\_2': h2['OUTPUT'], 'FIELD\_2': 'ID', 'FIELDS\_TO\_COPY': [],  'METHOD': 1, 'DISCARD\_NONMATCHING': **False**, 'PREFIX': '', 'OUTPUT': 'TEMPORARY\_OUTPUT'}) xyline = processing.run("shapetools:xy2line",  {'InputLayer': join['OUTPUT'], 'InputCRS': QgsCoordinateReferenceSystem(crs),  'OutputCRS': QgsCoordinateReferenceSystem(crs), 'LineType': 0, 'StartUseLayerGeom': **False**,  'StartXField': 'xcoord\_2', 'StartYField': 'ycoord\_2', 'EndUseLayerGeom': **False**,  'EndXField': 'xcoord', 'EndYField': 'ycoord', 'ShowStartPoint': **True**, 'ShowEndPoint': **True**,  'DateLineBreak': **False**, 'OutputLineLayer': 'TEMPORARY\_OUTPUT',  'OutputPointLayer': 'TEMPORARY\_OUTPUT'}) profile = processing.run("saga:profilesfromlines",  {'DEM': dem, 'VALUES': '', 'LINES': j['OutputLineLayer'], 'NAME': 'fid', 'SPLIT ': **False**,  'PROFILE': 'TEMPORARY\_OUTPUT', 'PROFILES': 'TEMPORARY\_OUTPUT'}) valuetopoint = processing.run("saga:addrastervaluestopoints",  {'SHAPES': profile['PROFILE'], 'GRIDS': slope, 'RESAMPLING': 0, 'RESULT': 'TEMPORARY\_OUTPUT'}) buffervar2 = processing.run("saga:variabledistancebuffer",  {'SHAPES': polytoline['OUTPUT'], 'DIST\_FIELD': '1-4\_raio', 'DIST\_SCALE': 1, 'NZONES': 1, 'DARC': 5,  'DISSOLVE ': **True**, 'POLY\_INNER ': **False**, 'BUFFER': 'TEMPORARY\_OUTPUT'}) clip = processing.run("saga:clippointswithpolygons",  {'POINTS': valuetopoint['RESULT'], 'POLYGONS': m['BUFFER'], 'FIELD': 'Drawings', 'METHOD': 0,  'CLIPS': 'TEMPORARY\_OUTPUT'}) centroid2 = processing.run("saga:polygoncentroids",  {'POLYGONS': fieldcalculator3['OUTPUT'], 'METHOD ': **True**, 'CENTROIDS': 'TEMPORARY\_OUTPUT'}) buffervar3 = processing.run("saga:variabledistancebuffer",  {'SHAPES': centroid2['CENTROIDS'], 'DIST\_FIELD': '8\_raio', 'DIST\_SCALE': 1, 'NZONES': 1, 'DARC': 5,  'DISSOLVE ': **True**, 'POLY\_INNER ': **False**, 'BUFFER': 'TEMPORARY\_OUTPUT'}) difference = processing.run("native:difference", {'INPUT': valuetopoint['RESULT'], 'OVERLAY': a5['BUFFER'], 'OUTPUT': 'TEMPORARY\_OUTPUT'})  layer11 = difference['OUTPUT'] QgsProject.instance().addMapLayer(layer11) layer1 = QgsVectorLayer(clip['CLIPS']) QgsProject.instance().addMapLayer(layer1)  # Maximum Elevation byexoress = processing.run("qgis:selectbyexpression",  {'INPUT': layer1, 'EXPRESSION': '\"Z\" = maximum(\"z\",\"LINE\_ID\")', 'METHOD': 0}) saveselected = processing.run("native:saveselectedfeatures", {'INPUT': layer1, 'OUTPUT': output\_me}) layer1.removeSelection() layer2 = QgsVectorLayer(o2['OUTPUT'])   # Delineation all\_features = layer2.getFeatures() all\_features2 = layer2.getFeatures() all\_features3 = layer2.getFeatures()  lista\_repetition = [] **for** alllineids **in** all\_features:  lista\_repetition.append(alllineids['LINE\_ID']) repetition = max(lista\_repetition)  lista\_repetition2 = [] **for** alllineids2 **in** all\_features2:  lista\_repetition2.append(alllineids2['LINE\_ID']) repetition2 = max(lista\_repetition2)  lista\_repetition3 = [] **for** alllineids3 **in** all\_features3:  lista\_repetition3.append(alllineids3['LINE\_ID']) repetition3 = max(lista\_repetition2)  layer1.startEditing() layer2.startEditing() layer11.startEditing() idx1 = layer2.fields().indexFromName('LINE\_ID') u1 = layer2.minimumValue(idx1) processing.run("qgis:selectbyexpression",  {'INPUT': layer2, 'EXPRESSION': '\"fid\"= maximum (\"fid\",\"LINE\_ID\") AND \"LINE\_ID\" = {}'.format(u1),  'METHOD': 0}) selected\_features = layer2.selectedFeatures() **for** point **in** selected\_features:  u2 = point['fid'] #print(u1) #print(u2) exp\_me0 = QgsExpression("\"LINE\_ID\"='{}'AND\"fid\"='{}'".format(u1, u2)) feature\_me = layer2.getFeatures(QgsFeatureRequest(exp\_me0)) x = u1 lista = list(map(str, range(repetition + 1))) listlayer1 = [] listlayer2 = [] **for** f **in** feature\_me:  geom\_me0 = f.geometry() **for** times **in** range(repetition + 1):  x = x + 1  #print('for1')  #print(x)  y = 0  **try**:  lineid = lista[x]  **except** IndexError:  **continue**  exp\_me = QgsExpression('\"LINE\_ID\" = (' + lineid + ')')  exp\_control = QgsExpression('\"LINE\_ID\" = (' + lineid + ')')  feature\_me = layer2.getFeatures(QgsFeatureRequest(exp\_me))  feature\_control = layer1.getFeatures(QgsFeatureRequest(exp\_control))  **for** f\_me **in** feature\_me:  geom\_me = f\_me.geometry()  dist2 = geom\_me0.distance(geom\_me)  #print('forme')  # print(f\_me['ID'])  # print(lineid)  **if** dist2 < 1000:  geom\_me0 = geom\_me  #print('me')  #print(f\_me['LINE\_ID'])  #print(f\_me['ID'])  layer2.select(f\_me.id())  **break**  **else**:  list\_id\_key = []  list\_dist\_value = []  list2\_id\_key = []  list2\_dist\_value = []  **for** f\_control **in** feature\_control:  geom\_control = f\_control.geometry()  dist3 = geom\_me0.distance(geom\_control)  list\_id\_key.append(f\_control['ID'])  list\_dist\_value.append(dist3)  keys = list\_id\_key  values = list\_dist\_value  dictionary = dict(zip(keys, values))  dictionary2 = dict(zip(keys, values))  list\_pcid = []  **for** times **in** range(3):  min\_val = min(dictionary.values())  **for** k, v **in** dictionary.items():  **if** v == min\_val:  list\_pcid.append(k)  **del** dictionary[k]  **break**  lista2 = list(map(str, list\_pcid))  y = 0  **for** times **in** range(3):  lineid2 = lista2[y]  exp\_control2 = QgsExpression('\"ID\" = (' + lineid2 + ') ')  feature\_pcid1 = layer1.getFeatures(QgsFeatureRequest(exp\_control2))  y = y + 1  **for** f\_pcid1 **in** feature\_pcid1:  feature\_pcid1\_geom = f\_pcid1.geometry()  dist4 = geom\_me.distance(feature\_pcid1\_geom)  list2\_id\_key.append(f\_pcid1['ID'])  list2\_dist\_value.append(dist4)  ke = list2\_id\_key  va = list2\_dist\_value  dictionary\_dist = dict(zip(ke, va))  min\_val2 = min(dictionary\_dist.values())  **for** ky, vs **in** dictionary\_dist.items():  **if** vs == min\_val2:  new\_select\_pcid = ky  pc\_slect\_str = str(new\_select\_pcid)  exp\_control3 = QgsExpression('\"ID\" = (' + pc\_slect\_str + ') ')  feature\_pcid2 = layer1.getFeatures(QgsFeatureRequest(exp\_control3))  **for** pc\_feature **in** feature\_pcid2:  geom\_me0 = pc\_feature.geometry()  #print('pc1')  #print(pc\_feature['LINE\_ID'])  #print(pc\_feature['ID'])  # print(list\_pcid)  # print(dictionary\_dist)  # print(new\_select\_pcid)  layer1.selectByExpression('\"ID\" IN (' + pc\_slect\_str + ')', QgsVectorLayer.AddToSelection)  **break**  listtostr = list(map(str, listlayer1)) p = 0 **for** feature **in** listlayer1:  pointtoselect = listtostr[p]  layer1.selectByExpression('\"ID\" IN (' + pointtoselect + ')', QgsVectorLayer.AddToSelection)  p = p + 1  saveselected\_pc = processing.run("native:saveselectedfeatures", {'INPUT': layer1, 'OUTPUT': output\_pc\_select}) saveselected\_me = processing.run("native:saveselectedfeatures", {'INPUT': layer2, 'OUTPUT': output\_me\_select})  layer\_pc\_select = QgsVectorLayer(saveselected\_pc['OUTPUT']) layer\_me\_select = QgsVectorLayer(saveselected\_me['OUTPUT'])  merge = processing.run("native:mergevectorlayers",  {'LAYERS': [layer\_pc\_select, layer\_me\_select], 'CRS': **None**, 'OUTPUT': output\_merge})  pointtopath = processing.run("qgis:pointstopath",  {'INPUT': merge['OUTPUT'], 'ORDER\_FIELD': 'LINE\_ID', 'GROUP\_FIELD': **None**, 'DATE\_FORMAT': '',  'OUTPUT':output\_line}) layer\_line = QgsVectorLayer(merge['OUTPUT']) QgsProject.instance().addMapLayer(layer\_line)   # rim hight **import** statistics  layer\_points = QgsVectorLayer(merge['OUTPUT']) feature\_points = layer\_points.getFeatures()  list\_altura = [] **for** alturas **in** feature\_points:  list\_altura.append(alturas['Z'])  features\_selected = layer\_points.selectAll() numb\_feature = layer\_points.selectedFeatureCount() layer\_points.removeSelection() sum\_ = sum(list\_altura) medium\_hight = sum\_ / (numb\_feature)  hight\_variance = statistics.stdev(list\_altura)  # depth layer\_all = QgsVectorLayer(valuetopoint['RESULT']) feature\_all = layer\_all.getFeatures() list\_zmin = []  **for** z **in** feature\_all:  list\_zmin.append(z['Z'])  depth = medium\_hight - min(list\_zmin)  # radius rim = QgsVectorLayer(merge['OUTPUT']) QgsProject.instance().addMapLayer(rim)  pointtopath2 = processing.run("qgis:pointstopath",   {'INPUT':rim,'ORDER\_FIELD':'LINE\_ID','GROUP\_FIELD':**None**,'DATE\_FORMAT':'','OUTPUT':'TEMPORARY\_OUTPUT'}) linetopoly2 = processing.run("qgis:linestopolygons", {'INPUT':pointtopath2['OUTPUT'],'OUTPUT':'TEMPORARY\_OUTPUT'}) centroid3 = processing.run("saga:polygoncentroids", {'POLYGONS':linetopoly2['OUTPUT'],'METHOD ':**True**,'CENTROIDS':'TEMPORARY\_OUTPUT'}) layer\_centroid = QgsVectorLayer(centroid3['CENTROIDS']) feature\_points = layer\_points.getFeatures() feature\_centroid = layer\_centroid.getFeatures() list\_radious = [] list\_radious2 = [] **for** centroid **in** feature\_centroid:  geom\_centrois = centroid.geometry() **for** points **in** feature\_points:  geom\_points = points.geometry()  dist = geom\_centrois.distance(geom\_points)  list\_radious.append(dist)  list\_radious2.append(dist) features\_selected = layer\_points.selectAll() numb\_feature = layer\_points.selectedFeatureCount() layer\_points.removeSelection() sum\_ = sum(list\_radious) diameter = sum\_ \* 2 medium\_diameter = diameter/(numb\_feature)  #base and wall i = 0 lista2 = list(map(str, range(repetition2 + 1))) **for** time **in** range(repetition2 + 1):  i = i + 1  **try**:  lineid2 = lista2[i]  **except** IndexError:  **continue**  exp\_slope = QgsExpression('\"LINE\_ID\" = (' + lineid2 + ') ')  feature\_slope = layer11.getFeatures(QgsFeatureRequest(exp\_slope))  list\_zmin = []  select\_wall\_list = []  parede\_id = []  **for** f\_slope **in** feature\_slope:  list\_zmin.append(f\_slope['Z'])  z\_max = max(list\_zmin)  z\_min = min(list\_zmin)  profundidade2 = z\_max - z\_min  z\_min\_prox = z\_min + ((profundidade2 / 5))  **for** zs **in** list\_zmin:  **if** zs >= z\_min\_prox:  select\_wall\_list.append(zs)   base\_wall = (str(min(select\_wall\_list)))   layer11.selectByExpression('\"Z\" IN (' + base\_wall + ') AND \"LINE\_ID\" = (' + lineid2 + ')',  QgsVectorLayer.AddToSelection) list\_data.append ((str(medium\_diameter))) list\_data.append ((str(depth))) list\_data.append ((str(hight\_variance)))  saveselectedbase = processing.run("native:saveselectedfeatures", {'INPUT': layer11, 'OUTPUT': output\_base}) join2 = processing.run("native:joinattributestable",  {'INPUT': saveselectedbase ['OUTPUT'], 'FIELD': 'LINE\_ID', 'INPUT\_2': merge['OUTPUT'], 'FIELD\_2': 'LINE\_ID',  'FIELDS\_TO\_COPY': [], 'METHOD': 1, 'DISCARD\_NONMATCHING': **False**, 'PREFIX': '',  'OUTPUT': 'TEMPORARY\_OUTPUT'}) xytoline2 = processing.run("shapetools:xy2line",  {'InputLayer': join2['OUTPUT'], 'InputCRS': QgsCoordinateReferenceSystem(crs),  'OutputCRS': QgsCoordinateReferenceSystem(crs), 'LineType': 0, 'StartUseLayerGeom': **False**,  'StartXField': 'X', 'StartYField': 'Y', 'EndUseLayerGeom': **False**, 'EndXField': 'X\_2',  'EndYField': 'Y\_2', 'ShowStartPoint': **True**, 'ShowEndPoint': **True**, 'DateLineBreak': **False**,  'OutputLineLayer': output\_line\_slope, 'OutputPointLayer': 'TEMPORARY\_OUTPUT'})  densify2 = processing.run("native:densifygeometriesgivenaninterval",  {'INPUT': xytoline2['OutputLineLayer'], 'INTERVAL': 300, 'OUTPUT': 'TEMPORARY\_OUTPUT'}) vertices2 = processing.run("native:extractvertices", {'INPUT': densify2['OUTPUT'], 'OUTPUT': 'TEMPORARY\_OUTPUT'}) valuetopoint2 = processing.run("saga:addrastervaluestopoints",  {'SHAPES': vertices2['OUTPUT'], 'GRIDS': slope, 'RESAMPLING': 0, 'RESULT': 'TEMPORARY\_OUTPUT'})  layer12 = QgsVectorLayer(valuetopoint2['RESULT'])  layer13 = QgsVectorLayer(xytoline2['OutputLineLayer']) QgsProject.instance().addMapLayer(layer13) i = 0 listslope = [] lista3 = list(map(str, range(repetition3 + 1))) **for** time **in** range(repetition3 + 1):  i = i + 1  # print('for1')  # print(c)  **try**:  lineid3 = lista3[i]  **except** IndexError:  **continue**  exp\_lineslope = QgsExpression('\"LINE\_ID\" = (' + lineid3 + ') ')  feature\_lineslope = layer12.getFeatures(QgsFeatureRequest(exp\_lineslope))  **for** slop **in** feature\_lineslope:  listslope.append(slop[''+slopeproj+'']) **print** ('sloepe='+ (str(statistics.mean(listslope)))) print('stdv slope ='+(str(statistics.stdev(listslope)))) list\_data.append ((str(statistics.mean(listslope)))) list\_data.append ((str(statistics.stdev(listslope))))  #wall widht and base\_diameter x = processing.run("qgis:exportaddgeometrycolumns",  {'INPUT':t['OutputLineLayer'],'CALC\_METHOD':0,'OUTPUT':'TEMPORARY\_OUTPUT'}) layer14 = x['OUTPUT'] QgsProject.instance().addMapLayer(layer14) featparede = layer14.getFeatures() parede = [] **for** f **in** featparede:  parede.append(f['length']) wall\_widht = statistics.mean(parede)  base\_diameter = media\_diametro - (espessura\_parede \* 2)  # lat long feat = QgsVectorLayer(b['CENTROIDS']) featfeat = feat.getFeatures() # get first feature **for** featurecent **in** featfeat:  geo = QgsGeometry.asPoint(featurecent.geometry()) # get the geometry of the feature  pxy = QgsPointXY(geo)  list\_data.append ((str((pxy.x())))) list\_data.append ((pxy.y())) list\_data.append ((wall\_widht)) list\_data.append ((base\_diameter))  # --------------------------------------------------------------------- # depth error  feature\_points2 = rim.getFeatures() list\_hight = [] **for** hights **in** feature\_points2:  list\_hight.append(hights['Z']) hight\_variation = statistics.stdev(list\_hight) medium\_hight = statistics.mean(list\_hight) number\_hight = len(list\_hight) filtro = midium\_hight - hight\_variation list\_points\_filto = [] **for** points **in** list\_hight:  **if** points >= filtro:  list\_points\_filto.append(points) mediun\_high2 = statistics.mean(list\_points\_filto) hight\_variation2 = statistics.stdev(list\_points\_filto) number\_hight2 = len(list\_points\_filto)  rim2 = QgsVectorLayer(merge['OUTPUT'])  assignporj= processing.run("native:assignprojection",   {'INPUT':rim2,  'CRS':QgsCoordinateReferenceSystem(crs),'OUTPUT':'TEMPORARY\_OUTPUT'})  pointtopath4 = processing.run("qgis:pointstopath", {'INPUT':assignporj['OUTPUT'],'ORDER\_FIELD':'LINE\_ID','GROUP\_FIELD':**None**,'DATE\_FORMAT':'','OUTPUT':'TEMPORARY\_OUTPUT'}) linetopoly4 = processing.run("qgis:linestopolygons", {'INPUT':pointtopath4['OUTPUT'],'OUTPUT':'TEMPORARY\_OUTPUT'}) fieldcalculator5 = processing.run("qgis:fieldcalculator",  {'INPUT':linetopoly4['OUTPUT'],'FIELD\_NAME':'raio','FIELD\_TYPE':0,'FIELD\_LENGTH':10,'FIELD\_PRECISION':3,  'NEW\_FIELD':**True**,'FORMULA':'((length(shortest\_line(centroid($geometry),boundary($geometry)))))/2','OUTPUT':'memory:'}) centroid4 = processing.run("saga:polygoncentroids",  {'POLYGONS':fieldcalculator5['OUTPUT'],'METHOD ':**True**,'CENTROIDS':'TEMPORARY\_OUTPUT'})  buffervar5 = processing.run("saga:variabledistancebuffer",  {'SHAPES': centroid4 ['CENTROIDS'], 'DIST\_FIELD': 'raio', 'DIST\_SCALE': 1, 'NZONES': 1, 'DARC': 5,  'DISSOLVE ': **True**, 'POLY\_INNER ': **False**, 'BUFFER': 'TEMPORARY\_OUTPUT'}) pixels = processing.run("qgis:generatepointspixelcentroidsinsidepolygons",  {'INPUT\_RASTER': dem, 'INPUT\_VECTOR': buffervar5['BUFFER'], 'OUTPUT': 'TEMPORARY\_OUTPUT'}) valuetopoint5 = processing.run("saga:addrastervaluestopoints",  {'SHAPES': pixels['OUTPUT'], 'GRIDS': dem, 'RESAMPLING': 0, 'RESULT': 'TEMPORARY\_OUTPUT'})  base\_layer = QgsVectorLayer(valuetopoint['RESULT']) feature\_points\_base = base\_layer.getFeatures() QgsProject.instance().addMapLayer(base\_layer) list\_base = [] **for** hights **in** feature\_points\_base:  list\_base.append((hights[''+demproj+''])) variation\_base = statistics.stdev(list\_base) medium\_base = statistics.mean(list\_base) number\_base = len(list\_base) filtro2 = medium\_base - variation\_base list\_points\_filtro\_base = [] **for** points2 **in** list\_base:  **if** points2 <= filtro2:  list\_points\_filtro\_base.append(points2) medium\_base2 = statistics.mean(list\_points\_filtro\_base) variation\_base2 = statistics.stdev(list\_points\_filtro\_base) number\_base2 = len(list\_points\_filtro\_base)  erro\_hight = (math.sqrt(number\_altura2 \* (550 \*\* 2))) / number\_hight2 erro\_hight2 = math.sqrt((erro\_alt \*\* 2) + hight\_variation2)  erro\_base = (math.sqrt(number\_base2 \* (550 \*\* 2))) / number\_base2 erro\_base2 = math.sqrt((erro\_base \*\* 2) + variation\_base2)  error\_depth = math.sqrt((erro\_alt2 \*\* 2) + (erro\_base2 \*\* 2))  # diameter error medium\_radius = statistics.mean(list\_radious2) list\_var= [] **for** dist **in** list\_radious2:  var = dist - medium\_radius  list\_var.append(var)  square\_var = [] **for** var **in** list\_var:  var\_square = var\*\*2  square\_var.append(var\_square)  mean\_squarevar = (sum(square\_var))/ (numb\_feature - 1)  error\_diam = math.sqrt(mean\_squarevar)  list\_data.append ((str(error\_depth))) list\_data.append ((str(error\_diam)))  #-------------------------------------  **import** csv file2 = open(''+csv\_file+'','a',newline="") writer = csv.writer(file2) data= list\_data2 writer.writerows(data) file2.close() |

**Table S1. Complete data.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Crater** | **Diameter (m)** | **Diameter Error** | **Depth (m)** | **Depth Error** | **d/D** | **Depth Variation** | **Slope (degrees)** | **Slope Variation** | **Wall Width (m)** | **Base Diameter (m)** | **Long.** | **Lat.** | **S-C** | **Fresh**  **Crater** | **Area** |
| 1 | 16,405 | 541.00 | 1983 | 91.17 | 0.12 | 291.35 | 20.5 | 9.89 | 3787 | 8830 | 147.8 | 9.2 | c | x | 4 |
| 2 | 34,825 | 640.47 | 2585 | 55.22 | 0.07 | 300.40 | 15.4 | 9.04 | 8131 | 18,562 | 149.8 | 8.1 | c | x | 4 |
| 3 | 19,753 | 276.17 | 2184 | 70.38 | 0.11 | 227.00 | 14.6 | 7.19 | 5540 | 8674 | 151.4 | 10.6 | c | x | 4 |
| 4 | 14,353 | 419.97 | 1246 | 84.89 | 0.09 | 353.27 | 13.1 | 5.67 | 3666 | 7021 | 148.6 | 12.3 | c |  | 4 |
| 5 | 33,187 | 1523.36 | 3335 | 57.09 | 0.10 | 389.20 | 19.6 | 11.30 | 7934 | 17,319 | 150.8 | 3.7 | c | x | 4 |
| 6 | 12,786 | 336.64 | 1843 | 107.12 | 0.14 | 261.47 | 22.5 | 10.62 | 3366 | 6054 | 147.9 | 5.1 | c | x | 4 |
| 7 | 11,690 | 258.15 | 2441 | 114.63 | 0.21 | 359.94 | 26.8 | 12.26 | 3158 | 5374 | 146.3 | 2.6 | s |  | 4 |
| 8 | 9295 | 234.23 | 1751 | 138.26 | 0.19 | 243.84 | 24.4 | 11.36 | 2854 | 3586 | 146.4 | 1.8 | s |  | 4 |
| 9 | 13,870 | 261.92 | 963 | 91.39 | 0.07 | 169.87 | 11.9 | 5.54 | 3262 | 7346 | 151.2 | 8.6 | c | x | 4 |
| 10 | 29,107 | 716.29 | 2591 | 58.57 | 0.09 | 607.70 | 12.2 | 9.16 | 7423 | 14,262 | 153.7 | 8.4 | c | x | 4 |
| 11 | 22,140 | 564.05 | 2162 | 76.23 | 0.10 | 513.99 | 14.1 | 8.21 | 5166 | 11,808 | 155.1 | 8.4 | c | x | 4 |
| 12 | 11,121 | 329.19 | 1668 | 126.83 | 0.15 | 185.56 | 18.5 | 10.00 | 3376 | 4369 | 144.5 | 8.8 | c | x | 4 |
| 13 | 14,126 | 261.74 | 1813 | 91.76 | 0.13 | 102.50 | 20.1 | 9.93 | 3214 | 7699 | 145.5 | 14.6 | c | x | 4 |
| 14 | 9703 | 201.49 | 1071 | 130.39 | 0.11 | 237.33 | 17.7 | 8.73 | 2439 | 4826 | 144.0 | 10.1 | c | x | 4 |
| 15 | 20,862 | 455.29 | 1274 | 70.74 | 0.06 | 145.81 | 12.2 | 7.28 | 4686 | 11,491 | 118.7 | 14.3 | c | x | 3 |
| 16 | 37,252 | 735.65 | 2162 | 45.94 | 0.06 | 177.21 | 14.2 | 9.74 | 7208 | 22,836 | 115.6 | 15.9 | c | x | 3 |
| 17 | 20,800 | 358.10 | 1502 | 73.24 | 0.07 | 176.61 | 15.4 | 8.64 | 4187 | 12,426 | 112.1 | 16.8 | c | x | 3 |
| 18 | 28,961 | 316.33 | 2189 | 53.14 | 0.08 | 473.05 | 11.2 | 6.60 | 7410 | 14,142 | 97.8 | 6.3 | c | x | 3 |
| 19 | 30,960 | 581.05 | 2359 | 56.57 | 0.08 | 424.64 | 12.6 | 6.83 | 7588 | 15,785 | 96.7 | 4.8 | c | x | 3 |
| 20 | 30,198 | 764.30 | 2981 | 57.17 | 0.10 | 414.81 | 19.2 | 10.34 | 7901 | 14,397 | 155.6 | −11.9 | c | x | 5 |
| 21 | 28,047 | 310.47 | 3018 | 54.76 | 0.11 | 259.12 | 18.1 | 9.12 | 8358 | 11,330 | 159.0 | −9.3 | c | x | 5 |
| 22 | 23,943 | 639.60 | 2048 | 55.88 | 0.09 | 252.59 | 14.1 | 7.29 | 6161 | 11,621 | 109.5 | 11.8 | c | x | 3 |
| 23 | 23,898 | 411.03 | 1724 | 66.94 | 0.07 | 206.92 | 12.1 | 6.13 | 5887 | 12,124 | 138.7 | 20.2 | c | x | 1 |
| 24 | 10,952 | 125.27 | 1016 | 108.98 | 0.09 | 124.17 | 13.7 | 5.58 | 3045 | 4863 | 136.7 | 20.5 | s | x | 1 |
| 25 | 6588 | 111.72 | 843 | 163.77 | 0.13 | 108.11 | 15.6 | 6.21 | 1943 | 2703 | 140.2 | 20.5 | s | x | 1 |
| 26 | 10,983 | 355.63 | 1525 | 121.29 | 0.14 | 115.25 | 17.3 | 7.45 | 3275 | 4434 | 138.3 | 18.9 | s | x | 1 |
| 27 | 11,961 | 304.35 | 1379 | 106.16 | 0.12 | 85.48 | 16.7 | 7.85 | 3203 | 5555 | 140.1 | 17.4 | c | x | 1 |
| 28 | 9623 | 210.14 | 1258 | 126.97 | 0.13 | 213.31 | 17.4 | 6.49 | 2889 | 3844 | 144.3 | 20.1 | s | x | 1 |
| 29 | 12,374 | 270.24 | 937 | 96.88 | 0.08 | 147.20 | 11.3 | 5.22 | 2926 | 6521 | 143.5 | 23.3 | c | x | 1 |
| 30 | 8307 | 239.77 | 921 | 147.55 | 0.11 | 117.82 | 15.2 | 6.75 | 2252 | 3803 | 145.8 | 18.9 | s | x | 1 |
| 31 | 35,847 | 909.52 | 2303 | 50.12 | 0.06 | 210.03 | 13.2 | 6.33 | 7057 | 21,732 | 132.2 | 20.3 | c | x | 1 |
| 32 | 17,806 | 323.97 | 1322 | 72.81 | 0.07 | 149.33 | 12.5 | 5.50 | 4056 | 9694 | 130.9 | 28.4 | c | x | 1 |
| 33 | 6083 | 92.97 | 584 | 189.01 | 0.10 | 79.43 | 12.5 | 3.91 | 1698 | 2688 | 136.6 | 22.1 | s | x | 1 |
| 34 | 17,570 | 259.79 | 1608 | 76.63 | 0.09 | 236.21 | 13.9 | 6.59 | 4573 | 8423 | 147.1 | 19.4 | c | x | 1 |
| 35 | 7643 | 194.36 | 777 | 152.72 | 0.10 | 271.43 | 14.5 | 4.65 | 1798 | 4048 | 145.6 | 17.5 | s | x | 1 |
| 36 | 16,177 | 467.93 | 1200 | 88.45 | 0.07 | 219.07 | 13.1 | 7.29 | 3826 | 8524 | 140.8 | 30.9 | c | x | 1 |
| 37 | 14,739 | 304.20 | 1198 | 81.51 | 0.08 | 105.82 | 10.4 | 6.53 | 4622 | 5494 | 126.5 | 28.1 | c | x | 1 |
| 38 | 21,192 | 650.36 | 1928 | 71.62 | 0.09 | 156.57 | 13.6 | 6.34 | 5833 | 9525 | 133.3 | 24.2 | c | x | 1 |
| 39 | 12,109 | 280.52 | 1254 | 90.89 | 0.10 | 160.97 | 14.2 | 6.18 | 3768 | 4572 | 139.2 | 24.1 | s | x | 1 |
| 40 | 29,454 | 603.89 | 1316 | 52.92 | 0.04 | 123.30 | 9.7 | 5.61 | 6483 | 16,488 | 136.0 | 29.0 | c | x | 1 |
| 41 | 11,529 | 243.63 | 1333 | 107.66 | 0.12 | 256.11 | 15.4 | 6.30 | 3121 | 5286 | 140.3 | 26.2 | s | x | 1 |
| 42 | 17,264 | 460.87 | 1233 | 80.63 | 0.07 | 236.62 | 11.4 | 6.18 | 4362 | 8540 | 121.0 | 32.1 | c | x | 1 |
| 43 | 26,626 | 851.90 | 1477 | 54.89 | 0.06 | 254.88 | 11.8 | 7.31 | 6198 | 14,229 | 100.5 | 26.5 | c | x | 2 |
| 44 | 20,850 | 595.84 | 1749 | 79.99 | 0.08 | 179.89 | 12.1 | 7.11 | 6115 | 8621 | 100.2 | 23.0 | c | x | 3 |
| 45 | 15,580 | 298.88 | 825 | 85.29 | 0.05 | 154.38 | 8.1 | 4.40 | 3728 | 8124 | 98.4 | 22.4 | c | x | 3 |
| 46 | 8661 | 243.14 | 570 | 132.50 | 0.07 | 181.64 | 9.0 | 3.14 | 2642 | 3376 | 101.7 | 18.5 | s | x | 3 |
| 47 | 21,020 | 643.90 | 1011 | 73.03 | 0.05 | 151.57 | 7.6 | 3.52 | 5867 | 9286 | 89.7 | 20.5 | c |  | 3 |
| 48 | 18,512 | 470.80 | 702 | 75.41 | 0.04 | 178.67 | 6.6 | 3.72 | 4897 | 8718 | 90.0 | 22.4 | c |  | 3 |
| 49 | 10,993 | 173.96 | 780 | 117.40 | 0.07 | 192.47 | 10.9 | 4.78 | 2917 | 5159 | 101.9 | 16.1 | s | x | 3 |
| 50 | 24,052 | 654.13 | 1669 | 60.69 | 0.07 | 172.83 | 13.3 | 8.22 | 7130 | 9792 | 137.6 | 5.1 | c | x | 4 |
| 51 | 14,830 | 387.82 | 1800 | 91.93 | 0.12 | 221.61 | 19.2 | 9.74 | 3756 | 7318 | 142.7 | 4.1 | c | x | 4 |
| 52 | 18,915 | 472.70 | 1318 | 74.31 | 0.07 | 109.69 | 12.7 | 8.18 | 4099 | 10,716 | 131.2 | 7.9 | c | x | 4 |
| 53 | 13,386 | 346.90 | 2055 | 101.97 | 0.15 | 204.51 | 21.9 | 11.81 | 3401 | 6585 | 135.4 | 9.0 | c | x | 4 |
| 54 | 8632 | 229.17 | 1178 | 150.82 | 0.14 | 162.28 | 20.0 | 9.14 | 2255 | 4122 | 133.9 | 9.7 | s |  | 4 |
| 55 | 7244 | 220.54 | 1006 | 149.99 | 0.14 | 138.00 | 18.3 | 7.94 | 1901 | 3443 | 142.9 | 2.6 | s |  | 4 |
| 56 | 14,393 | 355.88 | 2048 | 81.70 | 0.14 | 308.47 | 23.0 | 10.74 | 4495 | 5403 | 145.2 | −5.9 | c | x | 5 |
| 57 | 13,177 | 329.49 | 1717 | 101.36 | 0.13 | 326.74 | 20.3 | 10.33 | 3372 | 6434 | 151.6 | −7.8 | c | x | 5 |
| 58 | 16,342 | 517.74 | 2626 | 74.92 | 0.16 | 511.58 | 24.4 | 10.36 | 4188 | 7965 | 144.4 | −10.0 | c | x | 5 |
| 59 | 10,496 | 409.58 | 1319 | 165.57 | 0.13 | 322.72 | 20.2 | 9.23 | 2674 | 5148 | 152.4 | −11.3 | c | x | 5 |
| 60 | 10,737 | 233.00 | 2025 | 119.99 | 0.19 | 282.02 | 23.5 | 11.47 | 3138 | 4460 | 145.4 | −2.5 | s |  | 5 |
| 61 | 6652 | 139.90 | 1157 | 168.54 | 0.17 | 244.29 | 27.7 | 11.14 | 1431 | 3789 | 142.9 | −0.6 | s | x | 5 |
| 62 | 13,045 | 317.46 | 1973 | 105.82 | 0.15 | 233.22 | 23.0 | 11.37 | 3384 | 6278 | 145.6 | 0.9 | c | x | 5 |
| 63 | 24,481 | 499.23 | 1690 | 78.95 | 0.07 | 395.81 | 13.1 | 5.92 | 6587 | 11,308 | 140.0 | −14.2 | ~~c~~ | x | 5 |
| 64 | 44,857 | 988.35 | 3061 | 43.40 | 0.07 | 361.01 | 13.2 | 7.68 | 12,082 | 20,693 | 139.0 | −6.4 | c | x | 5 |
| 65 | 11,386 | 326.04 | 1520 | 137.88 | 0.13 | 178.81 | 19.4 | 9.79 | 2282 | 6822 | 148.1 | −10.9 | c | x | 5 |
| 66 | 14,178 | 328.54 | 1733 | 98.56 | 0.12 | 349.67 | 18.8 | 11.86 | 3183 | 7811 | 139.3 | −2.2 | c | x | 5 |
| 67 | 12,862 | 347.46 | 1967 | 106.67 | 0.15 | 488.59 | 23.4 | 11.95 | 3507 | 5848 | 155.0 | −1.3 | c | x | 5 |
| 68 | 6193 | 221.73 | 952 | 208.10 | 0.15 | 186.76 | 18.8 | 10.66 | 1739 | 2714 | 154.7 | −0.7 | s | x | 5 |
| 69 | 34,412 | 646.64 | 2360 | 57.13 | 0.07 | 293.18 | 13.4 | 7.41 | 7739 | 18,933 | 157.8 | 4.8 | c | x | 4 |
| 70 | 12,415 | 230.34 | 1032 | 104.87 | 0.08 | 323.91 | 12.2 | 5.55 | 2985 | 6445 | 160.8 | 2.6 | c |  | 4 |
| 71 | 8984 | 233.14 | 1391 | 124.89 | 0.15 | 98.85 | 17.0 | 8.64 | 3039 | 2906 | 137.7 | 16.5 | s |  | 4 |
| 72 | 24,134 | 342.50 | 1716 | 65.66 | 0.07 | 267.71 | 15.5 | 7.84 | 4847 | 14,439 | 141.6 | 16.0 | c | x | 4 |
| 73 | 28,690 | 355.82 | 1655 | 57.68 | 0.06 | 189.99 | 11.6 | 7.44 | 6535 | 15,620 | 135.1 | 16.5 | c | x | 4 |
| 74 | 8653 | 313.95 | 614 | 155.31 | 0.07 | 203.80 | 10.9 | 4.52 | 2377 | 3899 | 148.1 | 15.8 | s | x | 4 |
| 75 | 14,599 | 319.64 | 1675 | 87.73 | 0.11 | 218.45 | 17.6 | 8.28 | 3468 | 7663 | 144.4 | 26.9 | c | x | 1 |
| 76 | 12,925 | 311.66 | 933 | 91.85 | 0.07 | 147.09 | 12.8 | 7.44 | 2662 | 7602 | 122.8 | 35.7 | c | x | 1 |
| 77 | 12,593 | 335.96 | 908 | 87.24 | 0.07 | 120.20 | 11.7 | 5.31 | 2978 | 6638 | 129.3 | 35.0 | c | x | 1 |
| 78 | 29,608 | 638.31 | 1910 | 54.29 | 0.06 | 433.10 | 11.0 | 6.99 | 7342 | 14,924 | 146.1 | 37.7 | c | x | 1 |
| 79 | 20,779 | 757.39 | 1116 | 73.01 | 0.05 | 138.25 | 10.7 | 6.37 | 5207 | 10,366 | 128.5 | 36.5 | c | x | 1 |
| 80 | 10,359 | 307.42 | 660 | 116.31 | 0.06 | 135.79 | 10.9 | 5.75 | 2853 | 4654 | 127.1 | 36.1 | s |  | 1 |
| 81 | 30,959 | 715.05 | 902 | 65.42 | 0.03 | 156.58 | 6.9 | 5.15 | 7744 | 15,471 | 109.2 | 35.7 | c |  | 2 |
| 82 | 22,132 | 396.60 | 806 | 69.67 | 0.04 | 123.57 | 6.3 | 3.76 | 5831 | 10,469 | 106.6 | 36.7 | c |  | 2 |
| 83 | 20,886 | 531.24 | 892 | 68.69 | 0.04 | 188.56 | 7.1 | 4.51 | 6765 | 7356 | 111.9 | 38.7 | c |  | 2 |
| 84 | 15,755 | 328.28 | 453 | 73.05 | 0.03 | 91.48 | 4.2 | 2.50 | 4388 | 6979 | 89.0 | 40.3 | c |  | 2 |
| 85 | 18,017 | 774.89 | 1248 | 74.71 | 0.07 | 195.82 | 11.7 | 6.89 | 4859 | 8298 | 138.4 | 39.8 | c | x | 1 |
| 86 | 13,133 | 319.39 | 1095 | 94.80 | 0.08 | 171.47 | 12.4 | 5.99 | 3394 | 6345 | 140.2 | 40.9 | c |  | 1 |
| 87 | 8074 | 192.86 | 703 | 115.76 | 0.09 | 106.73 | 11.1 | 5.19 | 2104 | 3866 | 136.2 | 42.4 | s | x | 1 |
| 88 | 12,756 | 132.68 | 662 | 90.79 | 0.05 | 201.92 | 11.3 | 5.56 | 4342 | 4073 | 124.4 | 44.6 | s |  | 1 |
| 89 | 36,276 | 840.67 | 2453 | 46.10 | 0.07 | 371.64 | 11.4 | 6.53 | 9908 | 16,460 | 135.0 | 47.3 | c | x | 1 |
| 90 | 12,816 | 354.98 | 1293 | 88.10 | 0.10 | 186.78 | 13.8 | 7.68 | 3550 | 5716 | 138.5 | 47.3 | c | x | 1 |
| 91 | 22,403 | 777.80 | 1364 | 65.78 | 0.06 | 250.29 | 10.5 | 6.14 | 6177 | 10,049 | 138.3 | 49.1 | c | x | 1 |
| 92 | 17,184 | 372.76 | 1422 | 68.84 | 0.08 | 358.70 | 11.4 | 5.94 | 4429 | 8326 | 140.0 | 55.8 | c | x | 1 |
| 93 | 11,675 | 320.97 | 1175 | 101.40 | 0.10 | 170.05 | 15.2 | 7.49 | 3111 | 5452 | 144.3 | 38.6 | c | x | 1 |
| 94 | 12,417 | 365.86 | 1057 | 128.21 | 0.09 | 190.91 | 13.4 | 6.21 | 3061 | 6295 | 132.3 | 38.7 | c | x | 1 |
| 95 | 7225 | 110.45 | 758 | 124.91 | 0.10 | 104.62 | 13.1 | 5.91 | 1811 | 3603 | 136.1 | 43.0 | s | x | 1 |
| 96 | 11,135 | 382.11 | 883 | 109.15 | 0.08 | 137.79 | 9.5 | 5.12 | 3288 | 4560 | 139.9 | 51.2 | s |  | 1 |
| 97 | 15,612 | 650.60 | 1167 | 89.38 | 0.07 | 271.17 | 12.2 | 6.29 | 4628 | 6356 | 134.3 | 52.7 | c | x | 1 |
| 98 | 18,881 | 471.80 | 1567 | 71.29 | 0.08 | 188.45 | 12.5 | 7.87 | 5915 | 7051 | 173.0 | 52.4 | c |  | 1 |
| 99 | 13,588 | 474.67 | 1067 | 83.16 | 0.08 | 282.12 | 12.4 | 7.54 | 3246 | 7096 | 176.5 | 51.3 | c |  | 1 |
| 100 | 18,432 | 745.83 | 1857 | 72.48 | 0.10 | 523.73 | 12.7 | 7.15 | 4973 | 8486 | 179.3 | 50.8 | c | x | 1 |
| 101 | 17,518 | 449.21 | 1447 | 66.59 | 0.08 | 292.51 | 11.5 | 5.93 | 5338 | 6843 | 180.0 | 52.2 | c |  | 1 |
| 102 | 19,611 | 458.17 | 1634 | 66.15 | 0.08 | 382.44 | 13.0 | 8.25 | 4966 | 9679 | 161.3 | 50.7 | c | x | 1 |
| 103 | 33,963 | 640.06 | 1807 | 54.32 | 0.05 | 341.80 | 11.2 | 7.04 | 9273 | 15,416 | 152.5 | 48.0 | c | x | 1 |
| 104 | 40,192 | 1251.83 | 2001 | 48.12 | 0.05 | 720.31 | 12.1 | 8.04 | 8735 | 22,722 | 151.8 | 44.2 | c | x | 1 |
| 105 | 32,435 | 813.35 | 1880 | 55.68 | 0.06 | 780.82 | 11.7 | 7.68 | 7675 | 17,085 | 161.7 | 44.8 | c | x | 1 |
| 106 | 11,043 | 269.11 | 1164 | 130.90 | 0.11 | 300.82 | 14.5 | 7.16 | 2938 | 5166 | 162.8 | 49.9 | c | x | 1 |
| 107 | 16,029 | 681.00 | 1488 | 75.08 | 0.09 | 272.29 | 11.9 | 6.68 | 5043 | 5943 | 167.8 | 54.8 | c | x | 1 |
| 108 | 10,723 | 195.47 | 1133 | 107.85 | 0.11 | 344.57 | 15.4 | 9.16 | 2715 | 5292 | 150.6 | 43.1 | c | x | 1 |
| 109 | 12,729 | 284.35 | 844 | 89.68 | 0.07 | 176.60 | 11.8 | 7.57 | 2680 | 7369 | 146.2 | 29.3 | c | x | 1 |
| 110 | 11,015 | 368.15 | 800 | 130.28 | 0.07 | 182.49 | 12.9 | 6.89 | 2421 | 6172 | 141.5 | 30.2 | s |  | 1 |
| 111 | 13,591 | 474.73 | 992 | 89.27 | 0.07 | 174.42 | 11.4 | 6.28 | 3832 | 5926 | 145.4 | 31.7 | c | x | 1 |
| 112 | 9208 | 352.72 | 748 | 127.66 | 0.08 | 237.90 | 12.4 | 6.32 | 2398 | 4412 | 146.2 | 30.8 | s |  | 1 |
| 113 | 10,013 | 683.10 | 1100 | 128.70 | 0.11 | 138.42 | 12.9 | 5.60 | ~~3104~~ | 3806 | 142.9 | 27.0 | s | x | 1 |
| 114 | 10,290 | 275.91 | 513 | 138.87 | 0.05 | 151.54 | 9.5 | 5.75 | 2299 | 5691 | 151.6 | 19.0 | s |  | 1 |
| 115 | 31,177 | 447.45 | 2174 | 55.58 | 0.07 | 449.86 | 12.0 | 6.97 | 8255 | 14,668 | 106.7 | 9.0 | c | x | 3 |
| 116 | 8888 | 107.98 | 813 | 117.93 | 0.09 | 154.50 | 15.4 | 5.95 | 1889 | 5111 | 155.1 | 6.5 | s | x | 4 |
| 117 | 16,293 | 290.01 | 1122 | 72.21 | 0.07 | 222.51 | 9.6 | 5.28 | 3751 | 8792 | 101.1 | 50.0 | c | x | 2 |
| 118 | 24,398 | 1016.26 | 839 | 61.39 | 0.03 | 116.69 | 5.1 | 2.61 | 7098 | 10,202 | 85.7 | 47.2 | c |  | 2 |
| 119 | 11,986 | 124.76 | 434 | 101.45 | 0.04 | 133.38 | 5.7 | 2.61 | 3399 | 5188 | 91.2 | 37.1 | s | x | 2 |
| 120 | 20,112 | 463.79 | 1091 | 77.35 | 0.05 | 194.40 | 8.4 | 4.46 | 5437 | 9238 | 97.1 | 38.1 | c | x | 2 |
| 121 | 34,766 | 819.00 | 1792 | 52.99 | 0.05 | 167.64 | 9.4 | 5.27 | 7091 | 20,584 | 95.9 | 35.8 | c | x | 2 |
| 122 | 24,949 | 418.53 | 1229 | 57.74 | 0.05 | 164.09 | 8.7 | 4.46 | 6340 | 12,269 | 95.0 | 32.7 | c | x | 2 |
| 123 | 19,162 | 541.66 | 978 | 67.87 | 0.05 | 110.52 | 7.6 | 3.30 | 4759 | 9644 | 98.3 | 32.9 | c |  | 2 |
| 124 | 15,158 | 538.76 | 386 | 83.81 | 0.03 | 77.70 | 3.9 | 1.67 | 4204 | 6750 | 98.1 | 30.7 | c |  | 2 |
| 125 | 52,486 | 1175.18 | 1953 | 40.28 | 0.04 | 280.28 | 10.2 | 6.53 | 8750 | 34,985 | 94.6 | 27.9 | c | x | 2 |
| 126 | 10,820 | 501.92 | 390 | 120.80 | 0.04 | 124.09 | 5.3 | 2.81 | 2833 | 5154 | 97.7 | 24.3 | s |  | 3 |
| 127 | 13,325 | 402.92 | 937 | 98.42 | 0.07 | 236.89 | 13.3 | 7.68 | 3071 | 7183 | 107.1 | 19.6 | c | x | 3 |
| 128 | 9347 | 404.32 | 968 | 136.81 | 0.10 | 151.08 | 14.0 | 7.16 | 2699 | 3950 | 121.9 | 18.5 | s | x | 1 |
| 129 | 13,462 | 542.23 | 903 | 143.43 | 0.07 | 150.08 | 13.0 | 7.25 | 2837 | 7789 | 122.8 | 15.3 | c | x | 4 |
| 130 | 13,745 | 107.05 | 967 | 81.93 | 0.07 | 302.01 | 11.8 | 5.62 | 3212 | 7320 | 130.1 | 20.9 | c | x | 1 |
| 131 | 43,102 | 995.09 | 2368 | 49.74 | 0.05 | 475.79 | 13.5 | 6.48 | 7685 | 27,732 | 150.6 | 15.3 | c | x | 4 |
| 132 | 8586 | 110.83 | 1329 | 126.85 | 0.15 | 163.54 | 22.5 | 7.00 | 2021 | 4543 | 148.7 | 15.4 | s |  | 4 |
| 133 | 14,972 | 449.24 | 928 | 105.28 | 0.06 | 319.97 | 10.9 | 5.22 | 3616 | 7740 | 147.2 | 16.8 | c |  | 4 |
| 134 | 12,469 | 104.71 | 944 | 97.79 | 0.08 | 168.87 | 12.1 | 5.22 | 2673 | 7124 | 140.1 | 16.4 | c |  | 4 |
| 135 | 13,821 | 220.43 | 961 | 92.68 | 0.07 | 138.72 | 13.0 | 7.06 | 2905 | 8010 | 137.7 | 15.7 | c | x | 4 |
| 136 | 10,757 | 435.71 | 586 | 122.02 | 0.05 | 77.98 | 8.1 | 3.98 | 3097 | 4563 | 134.3 | 15.3 | s |  | 4 |
| 137 | 12,667 | 430.22 | 1221 | 119.45 | 0.10 | 199.03 | 15.9 | 8.47 | 3308 | 6051 | 133.0 | 14.4 | c | x | 4 |
| 138 | 9523 | 123.05 | 984 | 129.65 | 0.10 | 174.96 | 17.9 | 7.80 | 2117 | 5288 | 141.0 | 13.4 | s | x | 4 |
| 139 | 21,074 | 466.94 | 1160 | 78.08 | 0.06 | 170.88 | 11.7 | 7.42 | 5931 | 9212 | 128.6 | 14.1 | c | x | 4 |
| 140 | 18,773 | 330.81 | 1323 | 84.05 | 0.07 | 283.21 | 12.6 | 8.22 | 4939 | 8895 | 127.7 | 14.7 | c |  | 4 |
| 141 | 9688 | 140.94 | 1023 | 122.37 | 0.11 | 191.82 | 18.0 | 7.55 | 2203 | 5282 | 127.0 | 13.2 | s | x | 4 |
| 142 | 11,354 | 320.56 | 1057 | 127.90 | 0.09 | 210.06 | 11.1 | 5.78 | 3594 | 4166 | 106.8 | 11.6 | s | x | 3 |
| 143 | 19,582 | 570.34 | 1503 | 72.87 | 0.08 | 404.41 | 12.3 | 6.28 | 5524 | 8534 | 111.3 | 11.6 | c | x | 3 |
| 144 | 19,342 | 362.73 | 790 | 74.85 | 0.04 | 164.74 | 7.3 | 4.82 | 4896 | 9549 | 124.3 | 9.3 | c |  | 4 |
| 145 | 11,198 | 340.75 | 1113 | 125.45 | 0.10 | 346.90 | 15.6 | 5.85 | 2844 | 5510 | 120.4 | 7.3 | c | x | 4 |
| 146 | 18,616 | 419.36 | 1660 | 77.33 | 0.09 | 185.29 | 14.5 | 6.18 | 4827 | 8961 | 116.6 | 7.9 | c | x | 3 |
| 147 | 10,091 | 150.78 | 729 | 122.45 | 0.07 | 214.04 | 14.1 | 6.11 | 1923 | 6246 | 153.3 | 12.7 | s |  | 4 |
| 148 | 45,789 | 309.91 | 2950 | 69.45 | 0.06 | 368.32 | 11.1 | 6.87 | 12,347 | 21,095 | 104.0 | −13.8 | c | x | 3 |
| 149 | 42,355 | 310.15 | 2432 | 69.52 | 0.06 | 520.52 | 10.9 | 5.99 | 10,918 | 20,518 | 107.3 | −10.4 | c | x | 3 |
| 150 | 31,604 | 1066.86 | 1894 | 54.76 | 0.06 | 232.81 | 12.3 | 5.78 | 9444 | 12,717 | 116.7 | −7.7 | c | x | 3 |
| 151 | 42,545 | 1226.90 | 2481 | 49.68 | 0.06 | 525.04 | 11.4 | 5.58 | 12,037 | 18,471 | 121.9 | −9.8 | c | x | 5 |
| 152 | 49,572 | 735.27 | 3269 | 43.21 | 0.07 | 291.72 | 12.3 | 7.01 | 14,586 | 20,399 | 132.9 | −10.8 | c | x | 5 |
| 153 | 59,414 | 1758.99 | 2882 | 43.63 | 0.05 | 366.01 | 11.4 | 6.49 | 15,917 | 27,581 | 135.3 | −11.0 | ~~c~~ | x | 5 |
| 154 | 13,892 | 600.56 | 1629 | 89.79 | 0.12 | 379.41 | 16.5 | 7.03 | 4145 | 5601 | 136.5 | −29.5 | s | x | 5 |
| 155 | 15,269 | 396.08 | 1519 | 84.30 | 0.10 | 285.96 | 16.6 | 7.69 | 3852 | 7565 | 136.6 | −24.2 | c | x | 5 |
| 156 | 53,326 | 1773.98 | 4318 | 45.81 | 0.08 | 467.20 | 14.6 | 7.89 | 14,225 | 24,876 | 133.4 | −35.0 | c | x | 5 |
| 157 | 23,046 | 493.49 | 2897 | 74.19 | 0.13 | 289.35 | 18.1 | 8.36 | 6130 | 10,787 | 140.6 | −25.6 | c | x | 5 |
| 158 | 25,895 | 544.82 | 2917 | 64.30 | 0.11 | 412.40 | 19.2 | 8.92 | 6486 | 12,923 | 138.9 | −19.0 | c | x | 5 |
| 159 | 14,451 | 385.01 | 1873 | 117.51 | 0.13 | 355.54 | 20.7 | 8.12 | 3315 | 7822 | 146.4 | −17.8 | c | x | 5 |
| 160 | 47,201 | 915.52 | 3298 | 46.16 | 0.07 | 453.21 | 17.0 | 9.19 | 15,410 | 16,381 | 157.9 | −13.1 | c | x | 5 |
| 161 | 14,365 | 532.67 | 1637 | 83.81 | 0.11 | 440.48 | 14.7 | 8.30 | 4377 | 5611 | 166.7 | 48.6 | c | x | 1 |
| 162 | 12,447 | 453.63 | 1271 | 87.86 | 0.10 | 224.44 | 13.4 | 6.78 | 3351 | 5745 | 165.6 | 51.3 | c | x | 1 |
| 163 | 13,759 | 604.79 | 1247 | 73.71 | 0.09 | 316.35 | 11.6 | 5.86 | 3665 | 6429 | 163.1 | 62.3 | c | x | 1 |
| 164 | 17,562 | 507.64 | 1448 | 72.24 | 0.08 | 177.23 | 12.2 | 7.00 | 4900 | 7762 | 155.6 | 55.1 | c |  | 1 |
| 165 | 10,610 | 276.82 | 1257 | 104.06 | 0.12 | 160.31 | 16.3 | 9.20 | 2500 | 5611 | 149.3 | 37.3 | s | x | 1 |
| 166 | 31,583 | 584.01 | 1828 | 52.49 | 0.06 | 507.79 | 8.8 | 4.94 | 9432 | 12,720 | 147.8 | 21.2 | c | x | 1 |
| 167 | 22,588 | 647.27 | 1605 | 61.90 | 0.07 | 184.12 | 11.7 | 7.09 | 5940 | 10,707 | 127.5 | 50.0 | c | x | 1 |
| 168 | 9870 | 336.27 | 940 | 107.54 | 0.10 | 266.81 | 10.7 | 4.81 | 2577 | 4716 | 132.0 | 50.9 | s | x | 1 |
| 169 | 19,106 | 481.77 | 1861 | 81.23 | 0.10 | 258.60 | 15.5 | 9.13 | 5396 | 8313 | 164.4 | −7.6 | c | x | 5 |
| 170 | 12,564 | 523.89 | 1101 | 111.65 | 0.09 | 181.11 | 13.3 | 6.16 | 3312 | 5941 | 162.2 | −5.4 | c |  | 5 |
| 171 | 15,859 | 699.55 | 2084 | 92.59 | 0.13 | 336.82 | 18.1 | 8.83 | 4541 | 6778 | 149.3 | −30.0 | s | x | 5 |
| 172 | 9344 | 468.86 | 1036 | 142.84 | 0.11 | 297.70 | 20.4 | 10.47 | 2386 | 4572 | 155.0 | −9.9 | c | x | 5 |
| 173 | 5333 | 99.11 | 865 | 196.40 | 0.16 | 94.32 | 20.6 | 10.30 | 1551 | 2230 | 162.0 | −10.0 | s |  | 5 |
| 174 | 13,409 | 450.16 | 1338 | 89.01 | 0.10 | 227.84 | 14.2 | 8.02 | 3544 | 6320 | 135.1 | 1.5 | c |  | 4 |
| 175 | 12,838 | 363.74 | 1515 | 96.97 | 0.12 | 337.85 | 18.1 | 5.67 | 3343 | 6152 | 141.3 | −6.0 | s | x | 5 |
| 176 | 17,140 | 481.29 | 1408 | 84.32 | 0.08 | 334.03 | 14.2 | 7.22 | 5392 | 6356 | 137.6 | −7.3 | c |  | 5 |
| 177 | 12,229 | 475.03 | 1821 | 126.99 | 0.15 | 525.16 | 22.3 | 10.45 | 3236 | 5756 | 149.9 | −10.5 | c | x | 5 |
| 178 | 11,171 | 452.00 | 965 | 117.54 | 0.09 | 186.82 | 14.8 | 6.72 | 2690 | 5791 | 162.9 | −2.0 | c | x | 5 |
| 179 | 6357 | 128.65 | 748 | 180.83 | 0.12 | 148.56 | 17.6 | 9.80 | 1778 | 2800 | 143.5 | −2.9 | s |  | 5 |
| 180 | 34,715 | 535.85 | 1777 | 53.94 | 0.05 | 334.53 | 10.0 | 7.32 | 7776 | 19,163 | 97.1 | 12.0 | c |  | 3 |
| 181 | 28,153 | 501.09 | 1145 | 55.81 | 0.04 | 213.26 | 7.1 | 5.02 | 6556 | 15,041 | 95.2 | 11.1 | c |  | 3 |
| 182 | 17,187 | 338.71 | 1255 | 79.94 | 0.07 | 221.34 | 11.6 | 6.32 | 4418 | 8352 | 95.5 | 9.3 | c |  | 3 |
| 183 | 31,766 | 1035.83 | 1815 | 59.14 | 0.06 | 419.43 | 11.1 | 6.65 | 8651 | 14,464 | 118.4 | 5.9 | c |  | 3 |
| 184 | 33,561 | 650.79 | 2468 | 51.18 | 0.07 | 355.28 | 13.8 | 6.52 | 8891 | 15,779 | 131.6 | 2.3 | c | x | 4 |
| 185 | 19,120 | 719.31 | 1540 | 78.26 | 0.08 | 316.44 | 12.8 | 6.77 | 5305 | 8509 | 106.1 | 3.9 | c | x | 3 |
| 186 | 14,338 | 742.28 | 558 | 94.95 | 0.04 | 59.17 | 5.3 | 2.48 | 4689 | 4961 | 104.2 | 35.2 | c |  | 2 |
| 187 | 14,784 | 479.18 | 1007 | 86.72 | 0.07 | 336.31 | 12.1 | 7.18 | 4117 | 6550 | 113.6 | 30.0 | c | x | 2 |
| 188 | 14,938 | 615.38 | 649 | 81.50 | 0.04 | 229.84 | 8.2 | 6.61 | 4186 | 6565 | 110.3 | 32.5 | c |  | 2 |
| 189 | 17,848 | 795.94 | 1461 | 87.55 | 0.08 | 245.24 | 12.6 | 6.01 | 4948 | 7951 | 124.5 | −4.4 | c | x | 5 |
| 190 | 28,441 | 847.34 | 2740 | 58.39 | 0.10 | 333.99 | 17.7 | 7.60 | 6664 | 15,113 | 140.8 | −18.0 | c | x | 5 |
| 191 | 28,528 | 713.31 | 3044 | 59.79 | 0.11 | 622.72 | 18.0 | 7.87 | 7857 | 12,813 | 145.8 | −14.2 | c | x | 5 |
| 192 | 12,816 | 357.71 | 1176 | 89.95 | 0.09 | 242.60 | 13.6 | 5.17 | 3661 | 5493 | 102.7 | −1.9 | s |  | 3 |
| 193 | 12,182 | 426.38 | 812 | 92.64 | 0.07 | 107.45 | 11.7 | 6.41 | 3138 | 5906 | 122.8 | 32.9 | c |  | 1 |
| 194 | 30,300 | 746.19 | 1679 | 54.37 | 0.06 | 319.61 | 8.8 | 4.72 | 8180 | 13,939 | 134.3 | 28.5 | c | x | 1 |
| 195 | 22,145 | 279.94 | 2010 | 80.24 | 0.09 | 589.39 | 15.0 | 6.66 | 5802 | 10,540 | 143.3 | 25.7 | c | x | 1 |
| 196 | 49,474 | 950.64 | 3855 | 45.64 | 0.08 | 465.20 | 10.4 | 5.59 | 16,369 | 16,736 | 154.9 | 10.4 | c | x | 4 |
| 197 | 20,908 | 412.82 | 1118 | 73.91 | 0.05 | 211.31 | 10.6 | 5.66 | 4543 | 11,821 | 127.6 | 19.2 | c | x | 1 |
| 198 | 17,967 | 530.82 | 1363 | 82.53 | 0.08 | 258.53 | 15.8 | 9.01 | 4680 | 8608 | 143.2 | 8.4 | c |  | 4 |
| 199 | 8439 | 304.90 | 606 | 131.29 | 0.07 | 219.68 | 10.6 | 5.73 | 2256 | 3927 | 142.9 | 47.4 | s |  | 1 |
| 200 | 7493 | 159.35 | 702 | 151.28 | 0.09 | 134.34 | 14.5 | 5.69 | 1790 | 3914 | 159.1 | 4.9 | s |  | 4 |
| 201 | 49,726 | 643.18 | 1766 | 44.02 | 0.04 | 280.99 | 8.0 | 4.67 | 11,803 | 26,119 | 130.6 | 21.9 | c |  | 1 |
| 202 | 16,944 | 313.20 | 1284 | 74.36 | 0.08 | 133.25 | 12.5 | 6.48 | 4822 | 7301 | 131.9 | 25.9 | c | x | 1 |
| 203 | 6847 | 269.00 | 758 | 175.03 | 0.11 | 121.38 | 14.5 | 4.60 | 1897 | 3053 | 128.0 | 25.3 | s | x | 1 |
| 204 | 5888 | 198.57 | 986 | 206.38 | 0.17 | 129.30 | 20.9 | 7.88 | 1520 | 2849 | 139.2 | 6.6 | s |  | 4 |
| 205 | 20,930 | 276.99 | 1728 | 73.69 | 0.08 | 444.18 | 14.3 | 8.71 | 5963 | 9005 | 117.4 | −28.7 | c | x | 3 |
| 206 | 9724 | 196.01 | 940 | 175.02 | 0.10 | 347.17 | 16.6 | 5.59 | 2187 | 5349 | 115.2 | 10.1 | s | x | 3 |
| 207 | 13,815 | 328.37 | 2124 | 87.77 | 0.15 | 340.39 | 22.0 | 9.45 | 3336 | 7144 | 146.1 | −20.7 | c | x | 5 |
| 208 | 29,497 | 1365.85 | 3128 | 66.90 | 0.11 | 219.68 | 16.0 | 8.02 | 8305 | 12,887 | 129.9 | −31.7 | c | x | 5 |
| 209 | 13,934 | 490.69 | 1810 | 127.33 | 0.13 | 435.65 | 19.2 | 7.76 | 4461 | 5013 | 131.4 | −19.7 | s | x | 5 |
| 210 | 14,194 | 135.32 | 1467 | 85.74 | 0.10 | 255.98 | 16.2 | 5.64 | 4057 | 6079 | 138.0 | −4.1 | c |  | 5 |
| 211 | 22,293 | 676.81 | 953 | 110.38 | 0.04 | 339.99 | 12.1 | 9.71 | 4839 | 12,615 | 109.3 | 13.6 | c |  | 3 |
| 212 | 14,104 | 140.97 | 1513 | 86.22 | 0.11 | 232.82 | 16.7 | 7.52 | 3555 | 6994 | 123.3 | 4.0 | c | x | 4 |
| 213 | 5992 | 137.68 | 941 | 236.21 | 0.16 | 346.71 | 19.8 | 7.58 | 1773 | 2447 | 121.9 | 4.5 | s |  | 4 |
| 214 | 60739 | 915.53 | 2695 | 41.73 | 0.04 | 352.61 | 9.9 | 5.43 | 17,894 | 24,952 | 114.1 | −4.3 | c | x | 3 |
| 215 | 6082 | 107.05 | 878 | 187.65 | 0.14 | 309.25 | 19.1 | 5.97 | 1628 | 2825 | 144.9 | −11.8 | s | x | 5 |
| 216 | 22,755 | 309.91 | 2061 | 69.45 | 0.09 | 398.24 | 14.1 | 5.48 | 5531 | 11,694 | 128.4 | −9.4 | c | x | 5 |
| 217 | 12,226 | 309.95 | 599 | 118.20 | 0.05 | 157.91 | 9.4 | 4.53 | 3113 | 6000 | 108.3 | 47.5 | s | x | 2 |
| 218 | 6085 | 227.82 | 664 | 147.70 | 0.11 | 222.33 | 12.5 | 4.67 | 1544 | 2996 | 151.4 | 49.4 | s | x | 1 |
| 219 | 10,073 | 357.35 | 445 | 120.22 | 0.04 | 169.69 | 6.4 | 2.71 | 3402 | 3268 | 101.8 | 44.8 | s | x | 2 |
| 220 | 8165 | 404.45 | 923 | 119.78 | 0.11 | 236.77 | 13.0 | 7.65 | 2507 | 3150 | 136.3 | 48.7 | s | x | 1 |
| 221 | 7644 | 237.20 | 788 | 137.98 | 0.10 | 97.85 | 13.5 | 6.36 | 1760 | 4124 | 146.9 | 43.5 | s | x | 1 |
| 222 | 8319 | 233.44 | 546 | 167.29 | 0.07 | 194.48 | 10.0 | 6.47 | 2194 | 3930 | 113.5 | 6.1 | s | x | 3 |
| 223 | 10,739 | 258.40 | 407 | 105.51 | 0.04 | 109.51 | 5.1 | 2.47 | 2966 | 4806 | 98.4 | 36.1 | s | x | 2 |
| 224 | 10,568 | 139.57 | 662 | 107.39 | 0.06 | 104.15 | 8.8 | 2.31 | 3054 | 4459 | 107.1 | −16.8 | s | x | 3 |
| 225 | 9481 | 98.13 | 730 | 117.46 | 0.08 | 144.68 | 10.7 | 3.63 | 2907 | 3667 | 102.0 | 2.4 | s | x | 3 |
| 226 | 7913 | 128.20 | 822 | 149.08 | 0.10 | 151.65 | 13.3 | 4.77 | 2417 | 3079 | 126.5 | −4.1 | s | x | 5 |
| 227 | 10,877 | 435.15 | 767 | 122.12 | 0.07 | 132.40 | 10.7 | 4.43 | 3157 | 4562 | 117.5 | −24.7 | s | x | 3 |
| 228 | 9384 | 399.05 | 1362 | 129.26 | 0.15 | 210.89 | 19.6 | 6.77 | 2837 | 3711 | 139.9 | −20.8 | s | x | 5 |
| 229 | 12,089 | 465.28 | 974 | 107.98 | 0.08 | 307.27 | 13.0 | 4.59 | 3462 | 5164 | 107.6 | −2.8 | s | x | 3 |
| 230 | 8662 | 161.67 | 757 | 131.72 | 0.09 | 305.10 | 12.2 | 5.21 | 2611 | 3441 | 114.2 | −6.4 | s | x | 3 |
| 231 | 9198 | 211.77 | 1087 | 126.21 | 0.12 | 251.40 | 15.7 | 6.22 | 2781 | 3635 | 123.2 | −1.4 | s | x | 5 |
| 232 | 9305 | 295.47 | 518 | 133.55 | 0.06 | 146.49 | 7.8 | 2.56 | 2716 | 3873 | 100.9 | 19.2 | s | x | 3 |
| 233 | 6913 | 266.05 | 420 | 164.48 | 0.06 | 176.27 | 7.7 | 3.23 | 1876 | 3162 | 102.7 | 18.7 | s | x | 3 |
| 234 | 6617 | 243.99 | 244 | 196.45 | 0.04 | 109.96 | 5.9 | 2.23 | 1678 | 3262 | 103.0 | 20.7 | s | x | 3 |
| 235 | 8954 | 258.85 | 603 | 135.10 | 0.07 | 162.31 | 7.6 | 3.26 | 2702 | 3551 | 95.2 | 23.5 | s | x | 3 |
| 236 | 10,403 | 387.92 | 639 | 134.03 | 0.06 | 89.06 | 9.1 | 3.55 | 2901 | 4601 | 101.8 | 21.0 | s | x | 3 |
| 237 | 10,424 | 391.23 | 814 | 118.75 | 0.08 | 120.21 | 9.4 | 4.42 | 2927 | 4570 | 101.1 | 5.5 | s | x | 3 |