

Exploratory Analysis of Mars Crater Codebook

In [1]:

```
"""
Created on Sun Oct 18 16:02:19 2015

@author: Ivan Georgiev
"""

import pandas;
import numpy;

data = pandas.read_csv('/Data/Sandbox/Learn/Data Analysis and Interpretation/codebook
```

Dataset Characteristics

We will get first the number of observations and the number of variables retrieved from the file. Also we will output the variable (column) names.

In [2]:

```
print ("Number of observations(rows): {}".format(len(data)))
print ("Number of variables(columns): {}".format(len(data.columns)))
print ("Variable (column) names      : {}".format(",".join(data.columns)))
```

```
Number of observations(rows): 384343
Number of variables(columns): 10
Variable (column) names      : CRATER_ID,CRATER_NAME,LATITUDE_CIRCLE_IMA
GE, LONGITUDE_CIRCLE_IMAGE,DIAM_CIRCLE_IMAGE,DEPTH_RIMFLOOR_TOPOG,MORPHO
LOGY_EJECTA_1,MORPHOLOGY_EJECTA_2,MORPHOLOGY_EJECTA_3,NUMBER_LAYERS
```

We will continue our exploratory analysis by focusing on the longitude (LONGITUDE_CIRCLE_IMAGE), latitude (LATITUDE_CIRCLE_IMAGE) and diameter (DIAM_CIRCLE_IMAGE) variables.

Study Individual Variables

Let's first see what are the extreme values of the studied variables.

In [3]:

```
long_low = data['LONGITUDE_CIRCLE_IMAGE'].min()
long_high = data['LONGITUDE_CIRCLE_IMAGE'].max()

lat_low = data['LATITUDE_CIRCLE_IMAGE'].min()
lat_high = data['LATITUDE_CIRCLE_IMAGE'].max()

diam_low = data['DIAM_CIRCLE_IMAGE'].min()
diam_high = data['DIAM_CIRCLE_IMAGE'].max()

print ("Latitude (Min, Max): ({}, {})".format(lat_low, lat_high))
print ("Longitude (Min, Max): ({}, {})".format(long_low, long_high))
print ("Diameter (Min, Max): ({}, {})".format(diam_high, diam_high))
```

```
Latitude (Min, Max): (-86.7, 85.702)
Longitude (Min, Max): (-179.997, 179.997)
Diameter (Min, Max): (1164.22, 1164.22)
```

For further analysis we will use the GraphlabCreate package which provides rich graphics capabilities. First we will create a SFrame using the already loaded data.

In [4]:

```
import graphlab
graphlab.canvas.set_target('ipynb')    # Setting GraphLab's output to IPython

gldata = graphlab.SFrame(data)         # Create graphlab SFrame from the loaded data
```

```
[INFO] This non-commercial license of GraphLab Create is assigned to ivan.georgiev@gmail.com and will expire on October 08, 2016. For commercial licensing options, visit https://dato.com/buy/. (https://dato.com/buy/.)
```

```
[INFO] Start server at: ipc:///tmp/graphlab_server-9440 - Server binary: C:\Users\baobab\AppData\Local\Dato\Dato Launcher\lib\site-packages\graphlab\unity_server.exe - Server log: C:\Users\baobab\AppData\Local\Temp\graphlab_server_1445178256.log.0
[INFO] GraphLab Server Version: 1.6.1
```

Let's see a sample of the data.

In [5]:

```
gldata
```

Out[5]:

CRATER_ID	CRATER_NAME	LATITUDE_CIRCLE_IMAGE	LONGITUD
01-000000		84.367	
01-000001	Korolev	72.76	
01-000002		69.244	
01-000003		70.107	
01-000004		77.996	
01-000005		68.547	
01-000006		69.492	
01-000007		78.716	
01-000008		75.539	
01-000009		69.371	



MORPHOLOGY_EJECTA_1	MORPHOLOGY_EJECTA_2	MORPHOLOGY_
Rd/MLERS	HuBL	
Rd		

[384343 rows x 10 columns]
Note: Only the head of the SFrame is printed.
You can use print_rows(num_rows=m, num_columns=n) to print more rows and columns.

In [6]:

```
# See the general characteristics of the three variables.

my_vars = ['LONGITUDE_CIRCLE_IMAGE', 'LATITUDE_CIRCLE_IMAGE', 'DIAM_CIRCLE_IMAGE']
gldata[my_vars].show()
```

LONGITUDE_CIRCLE_IMAGE		LATITUDE_CIRCLE_IMAGE	
dtype:	float	dtype:	float
num_unique (est.):	232,620	num_unique (est.):	128,000
num_undefined:	0	num_undefined:	0
min:	-179.997	min:	-89.997
max:	179.997	max:	89.997
median:	12.699	median:	-10.000
mean:	10.128	mean:	-7.000
std:	96.641	std:	33.000
distribution of values:		distribution of values:	
			

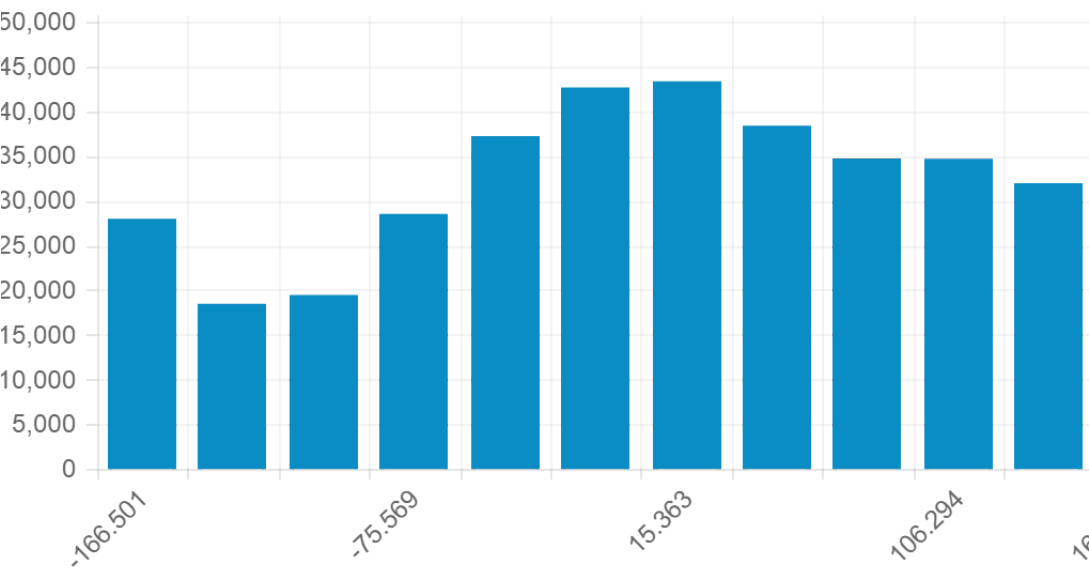
We can see that the longitude is relatively symmetric and is with wider distribution than the latitude. The latitude is narrower with a mean being offsetted.

And looking into the specific variables.

In [7]:

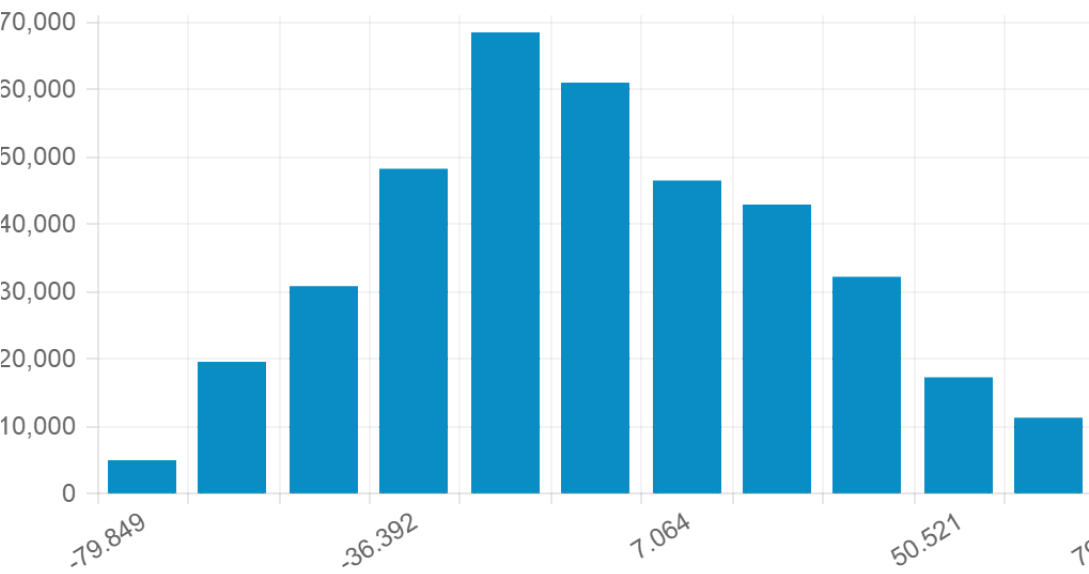
```
gldata[my_vars[0]].show() # Longitude
gldata[my_vars[1]].show() # Latitude
gldata[my_vars[2]].show() # Crater
```

Distribution of values from <SArray>



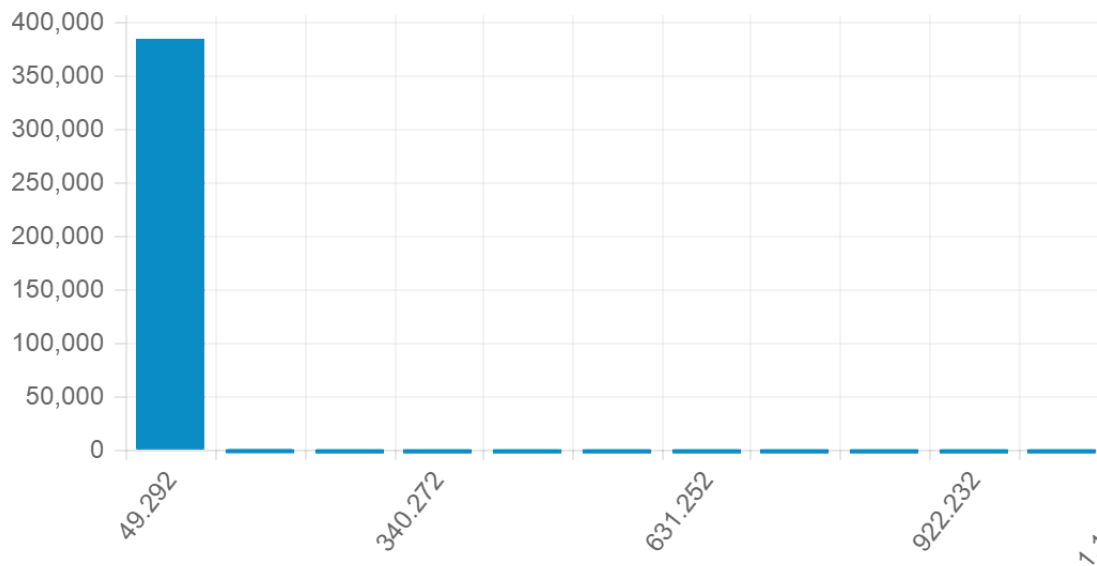
min	-179.997
max	179.997
mean	10.128
std	96.641
quantile(0.01) (est.)	-176.628
quantile(0.25) (est.)	-58.566
quantile(0.5) (est.)	12.699
quantile(0.75) (est.)	89.459
quantile(0.99) (est.)	175.701

Distribution of values from <SArray>



min	-86.7
max	85.702
mean	-7.199
std	33.609
quantile(0.01) (est.)	-73.897
quantile(0.25) (est.)	-30.922
quantile(0.5) (est.)	-10.105
quantile(0.75) (est.)	17.251
quantile(0.99) (est.)	68.436

Distribution of values from <SArray>



min	1
max	1,164.22
mean	3.557
std	8.592
quantile(0.01) (est.)	1.01
quantile(0.25) (est.)	1.19
quantile(0.5) (est.)	1.53
quantile(0.75) (est.)	2.56
quantile(0.99) (est.)	37.21

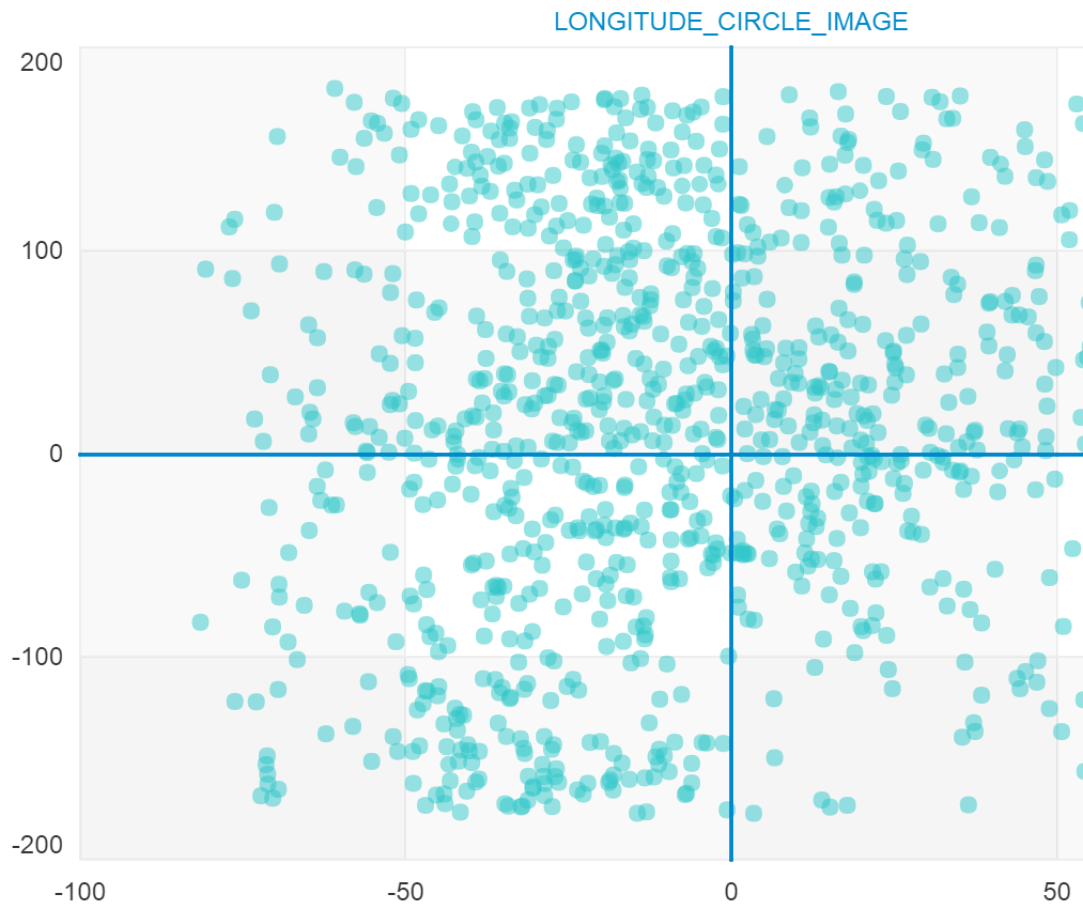
We can see that 98.8% of the craters are under 100 kilometers in diameter. Also the Longitude frequency has two extremes - a minimum (-150 to -90) and maximum (around zero - -30 to +30). The Latitude frequency has a strong maximum between -14 and -29.

Basic Variable Relations Exploration

We can perform a frequency distribution analysis on the data. I know that the data is pretty randomly distributed, so I prefer the visual look of the scattered plot.

In [8]:

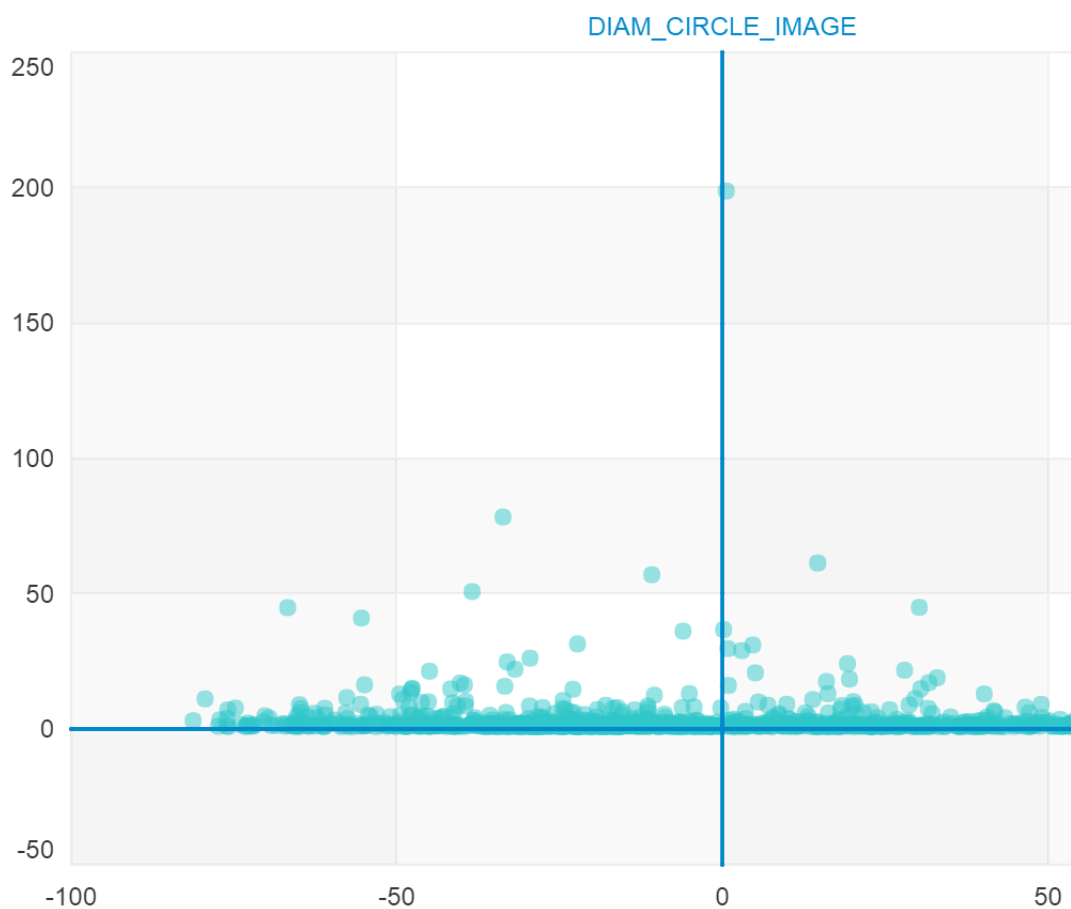
```
gldata.show(view="Scatter Plot", x="LATITUDE_CIRCLE_IMAGE", y="LONGITUDE_CIRCLE_IMAGE")
```



I want also to check the distribution of latitude vs diameter and longitude vs diameter. We can see that the majority of the craters have diameter below 10 kilometers.

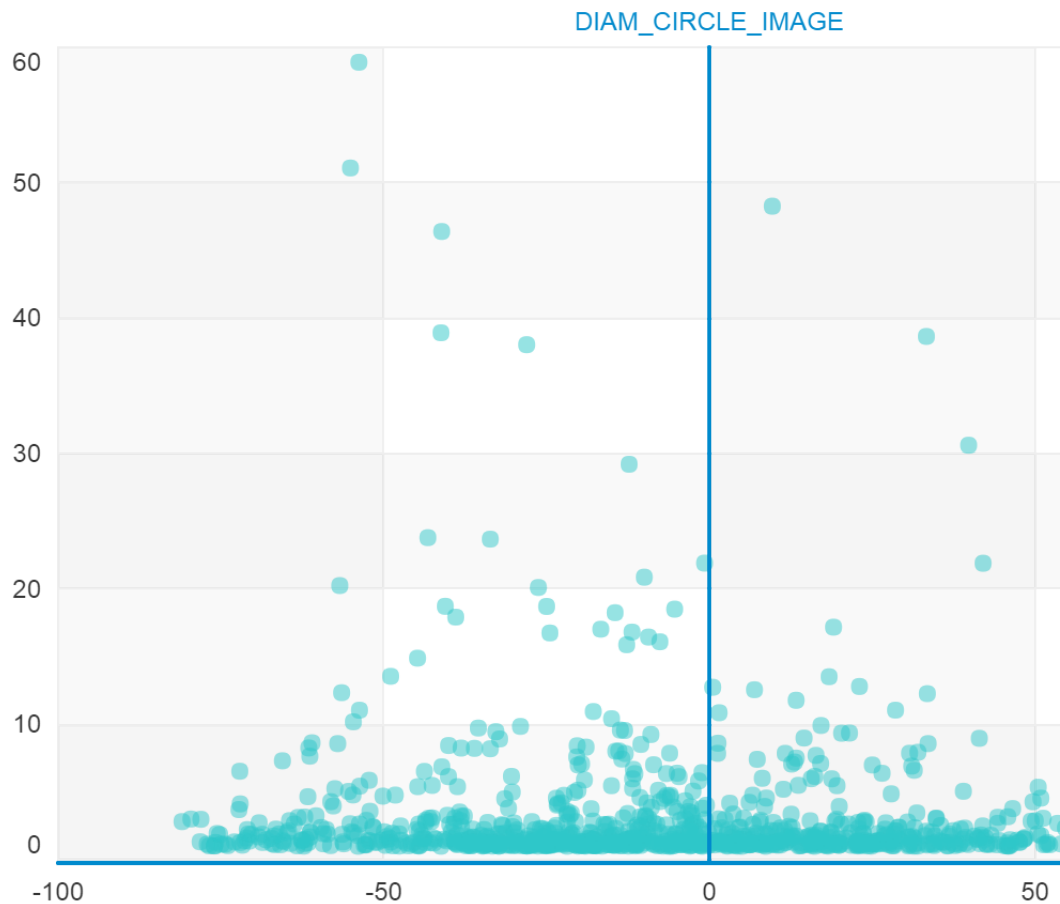
In [9]:

```
gldata.show(view="Scatter Plot", x="LATITUDE_CIRCLE_IMAGE", y="DIAM_CIRCLE_IMAGE")
```



In [10]:

```
gldata.show(view="Scatter Plot", x="LAONGITUDE_CIRCLE_IMAGE", y="DIAM_CIRCLE_IMAGE")
```



Exploring a Subset of Data

We want to explore the craters that are close to the latitude distribution maximum.

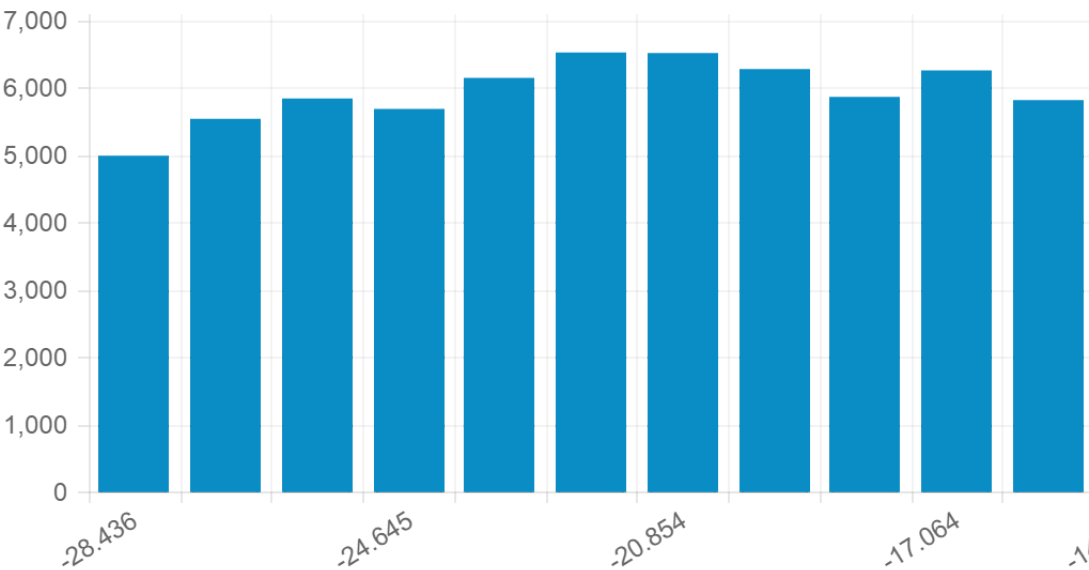
In [11]:

```
lat_max = gldata[(gldata['LATITUDE_CIRCLE_IMAGE'] <= -14) & (gldata['LATITUDE_CIRCLE_
```

In [12]:

```
lat_max['LATITUDE_CIRCLE_IMAGE'].show()
```

Distribution of values from <SArray>



min	-29
max	-14
mean	-21.393
std	4.231
quantile(0.01) (est.)	-28.829
quantile(0.25) (est.)	-24.983
quantile(0.5) (est.)	-21.36
quantile(0.75) (est.)	-17.749
quantile(0.99) (est.)	-14.175

In []: