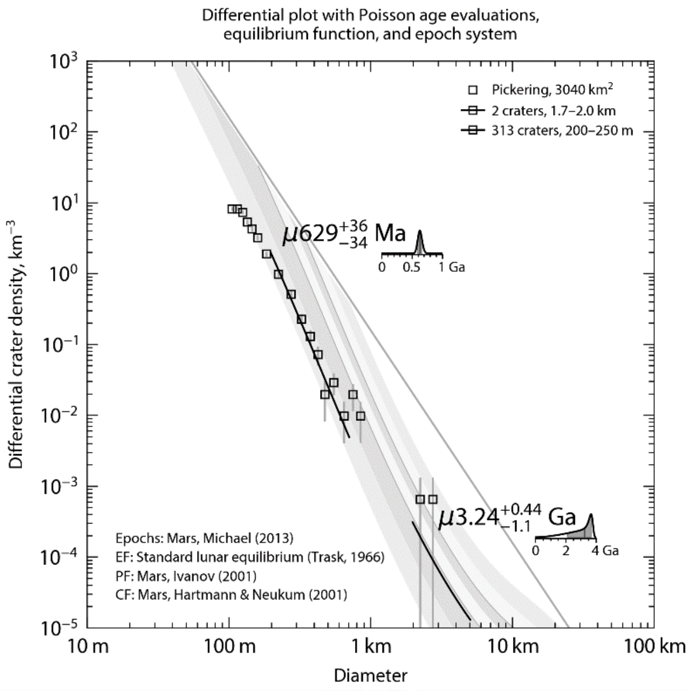


Planetary surface dating with Craterstats3 – a new open source implementation in Python

Greg Michael

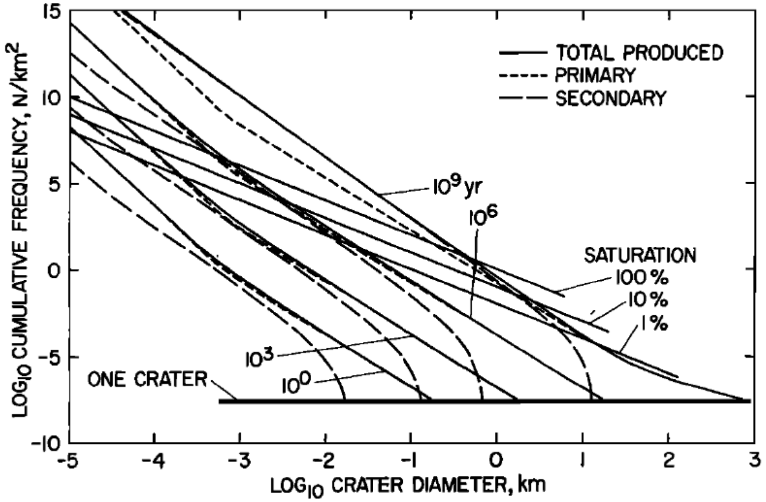
Geology & Planetary Mapping Winter School
7 – 11 February 2022

Overview

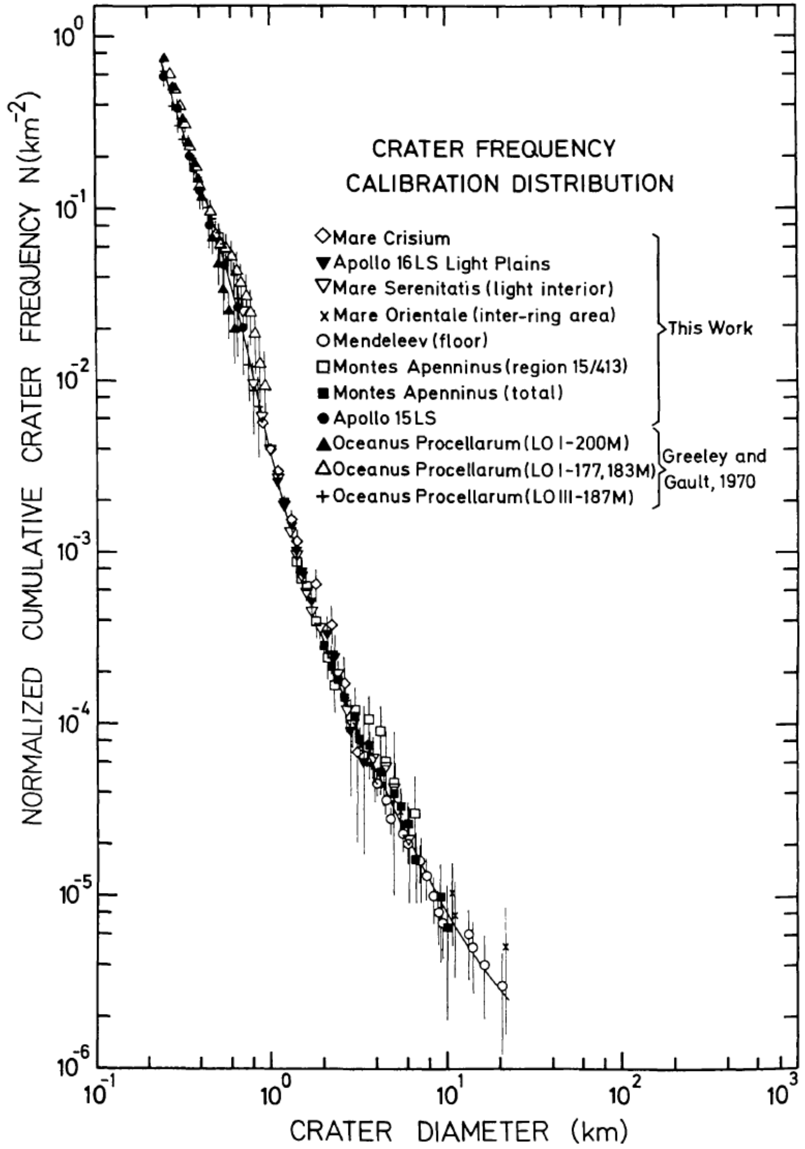
- What are the constituents of a crater chronology model?
- Common styles of crater count data presentation
- Measurement and analysis software
- *Craterstats3* – some examples of how to use it

What are the constituents of a crater chronology model?

1. Craters form with a size distribution that depends on the impactor population
2. The rate of crater formation is a function of time that depends on the impactor flux
3. The impactor size distribution could potentially be an evolving function



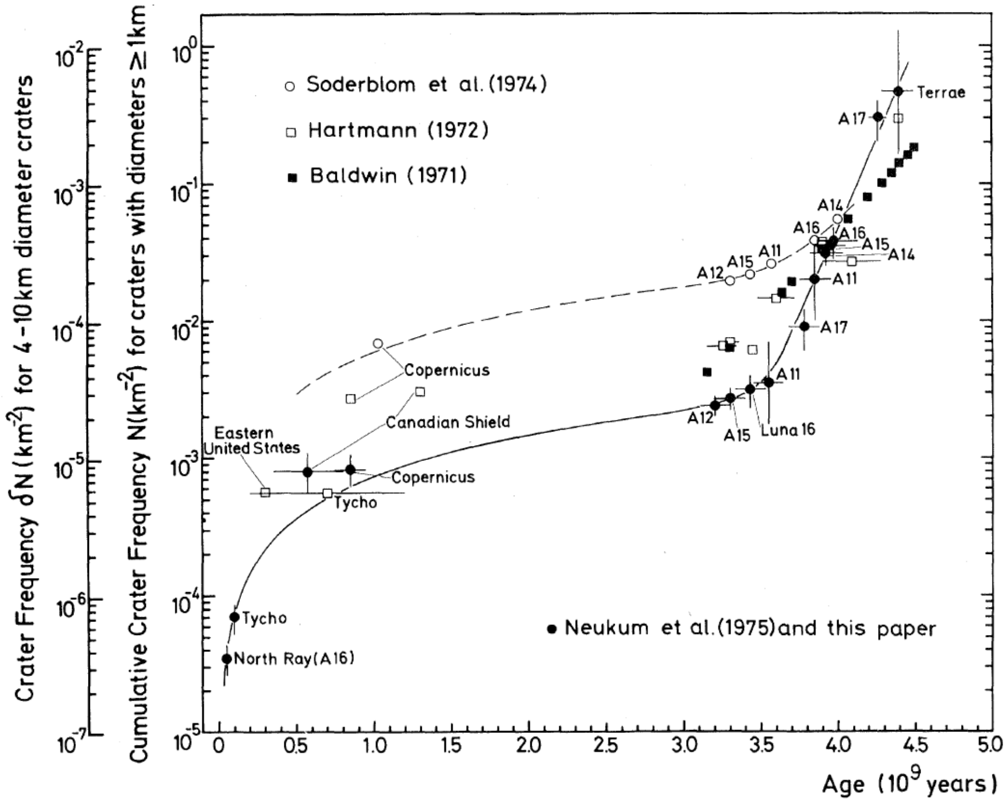
Flux model - Gault, 1970



Normalised crater counts – Neukum, König, Arkani-Hamed, 1975

What are the constituents of a crater chronology model?

1. Craters form with a size distribution that depends on the impactor population
2. The rate of crater formation is a function of time that depends on the impactor flux
3. The impactor size distribution could potentially be an evolving function

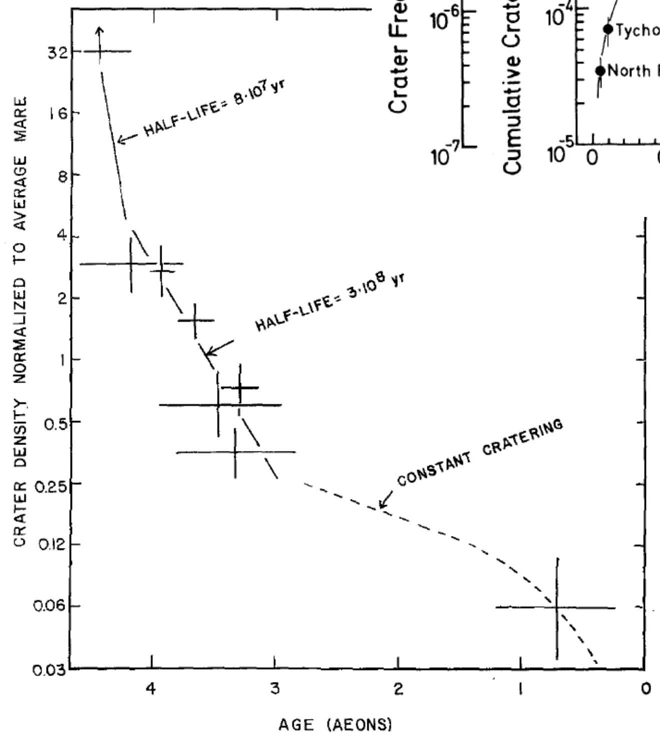


Neukum & Wise, 1976

$$T = -0.819 \log N_c + 1.651, \qquad (12)$$

where N_c is the cumulative number of 100 mile and wider lunar craters formed subsequent to given geologic times. The unit of age is 10^9 years. $T = 0$ at a time 4.5×10^9 years ago.

Baldwin, 1971



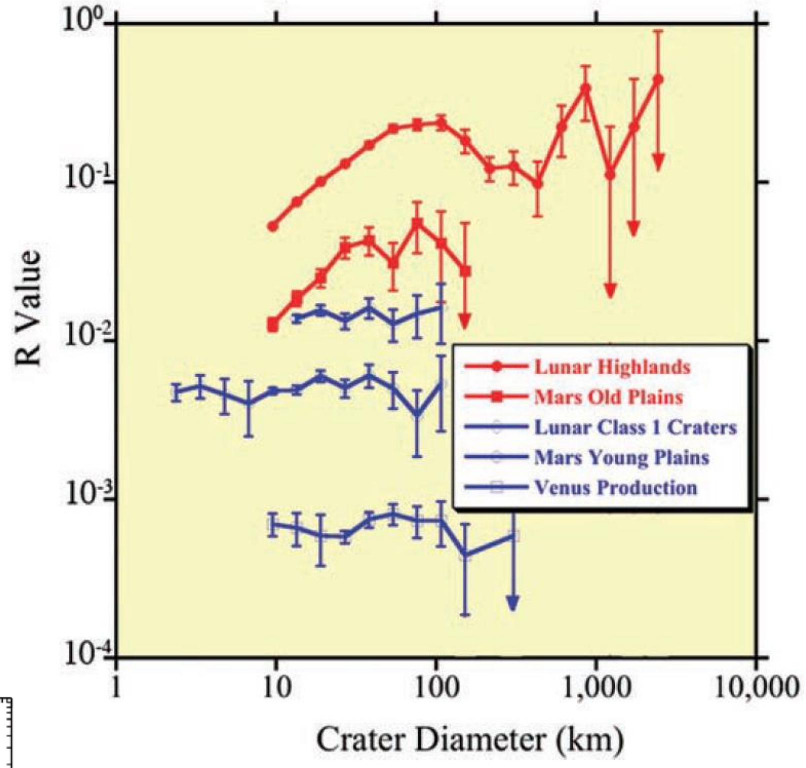
Hartmann, 1972

What are the constituents of a crater chronology model?

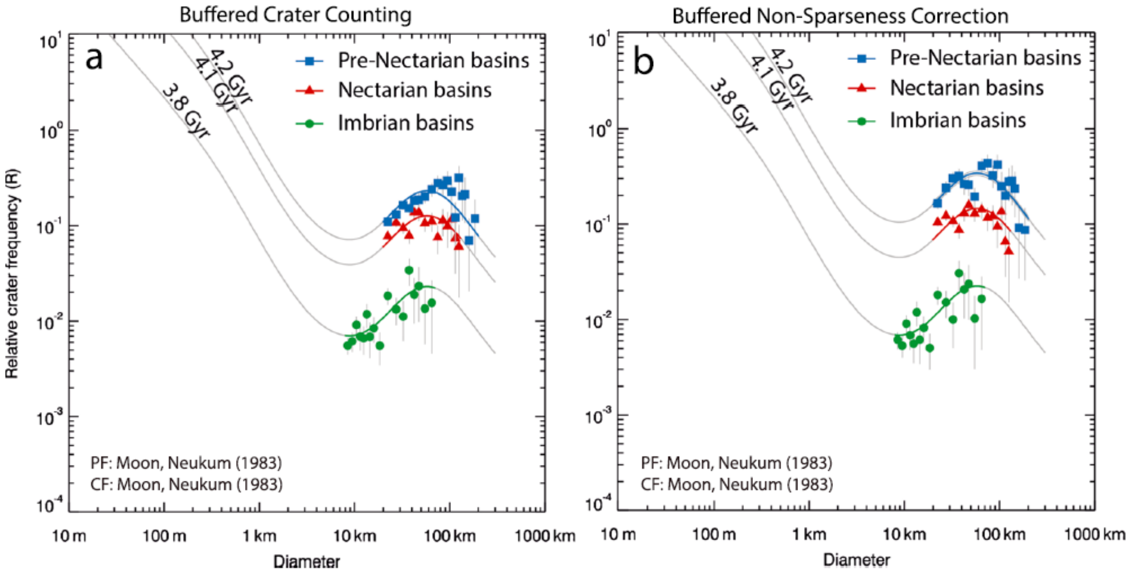
1. Craters form with a size distribution that depends on the impactor population

2. The rate of crater formation is a function of time that depends on the impactor flux

3. The impactor size distribution could potentially be an evolving function



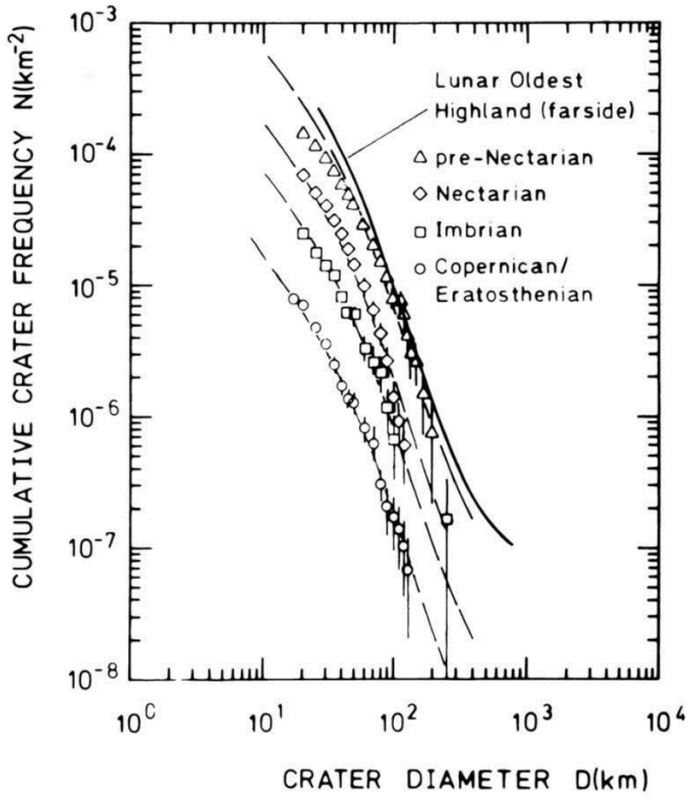
Strom et al, 2005



Orgel et al, 2017

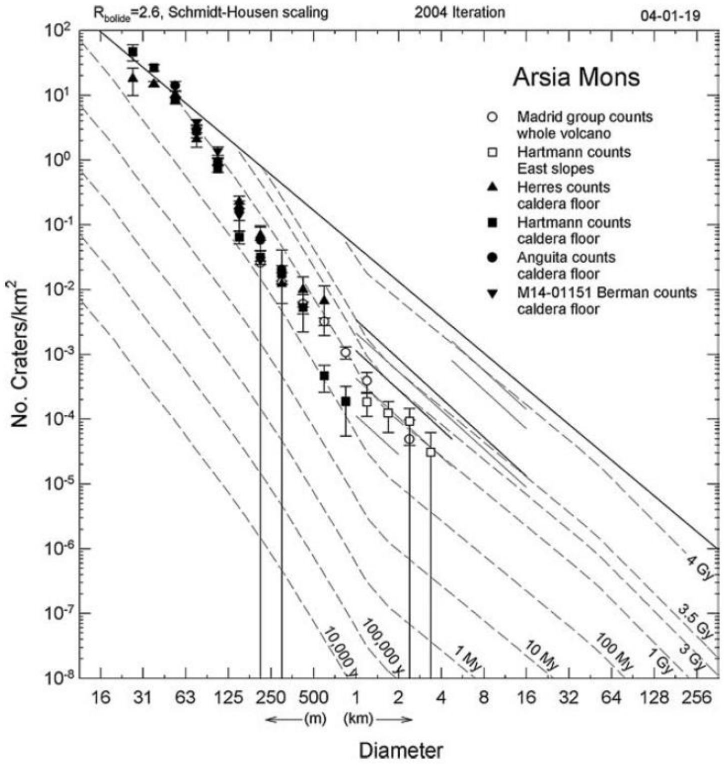
Common styles of crater count data presentation

log-log reverse cumulative histogram
18 bins/decade



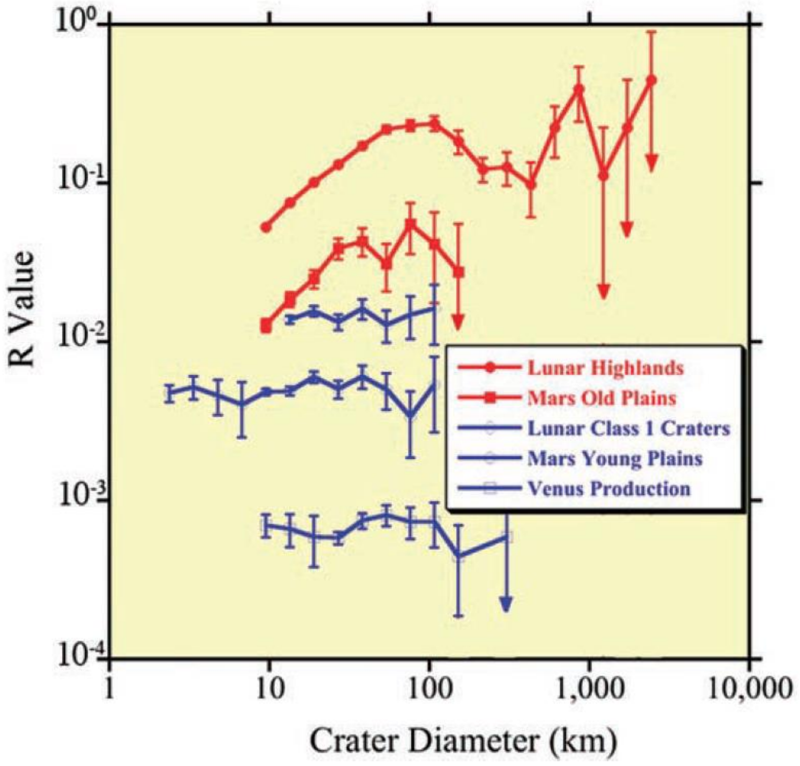
Neukum and Ivanov, 1994

log-log histogram
√2 bins (~6.5/decade)



Hartmann, 2005

'R-plot':
log-log histogram divided by
bin-widths * D^{-3}
√2 bins (~6.5/decade)

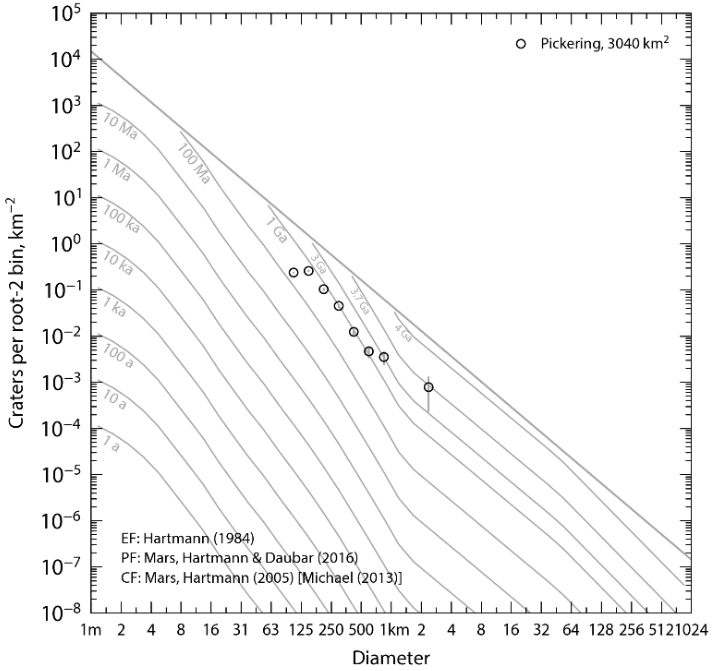


Strom et al, 2005

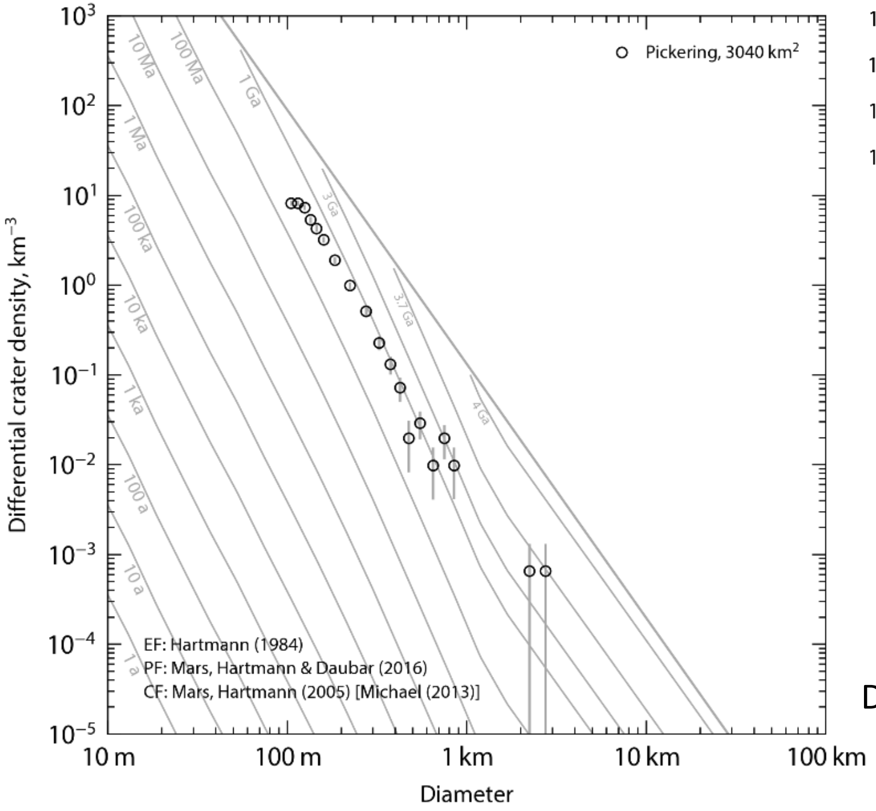
Competing systems of crater count data presentation

- Fourth variant – differential (Arvidson et al, 1979)
    - log-log histogram divided by bin-widths
    - any binning
    - data point independence (unlike cumulative)
    - easy to discern resurfacing effects
    - visually similar to Hartmann plot

    - good compromise



Hartmann-style

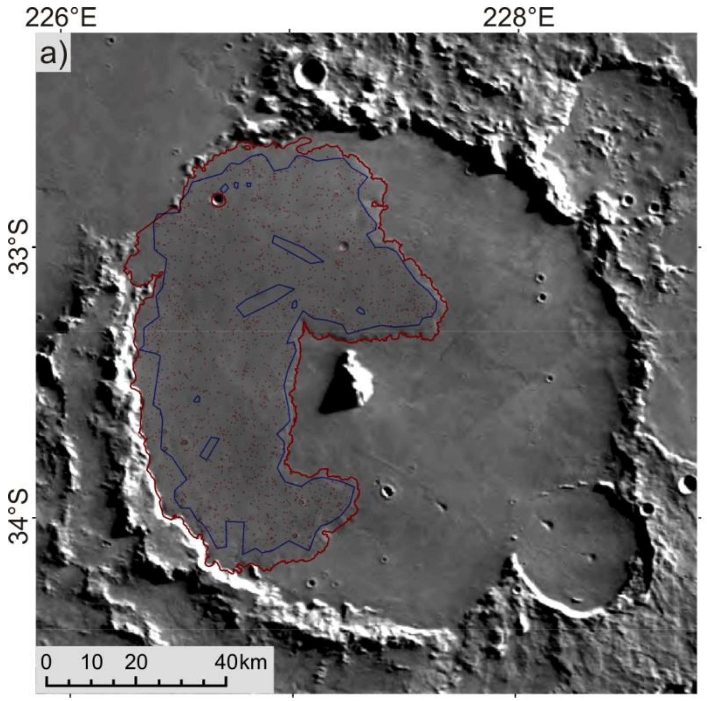


Differential

Crater measurement software

- Specify region of interest
- Mark out superposed impact craters
- Extract diameter measurements
    - correcting for projection and topographic effects

```
#Model .diam file for Craterstats
Area <km^2> = 3036.61
#diameter, km
0.511
0.166
0.095
0.100
0.261
0.208
0.208
0.103
0.112
0.170
0.100
0.130
...
```



CraterTools (Thomas Kneissl)
- most widely used and complete in features
- available from FUB planetary sciences website
- works only with older versions of ArcGIS

CSFD_Tools (Christian Riedel)
- for specialised analysis (non-sparseness correction)
- available on github

Circle-craters (Sarah Braden, Alessandro Frigeri)
- QGIS plugin

JMARS (ASU) and others

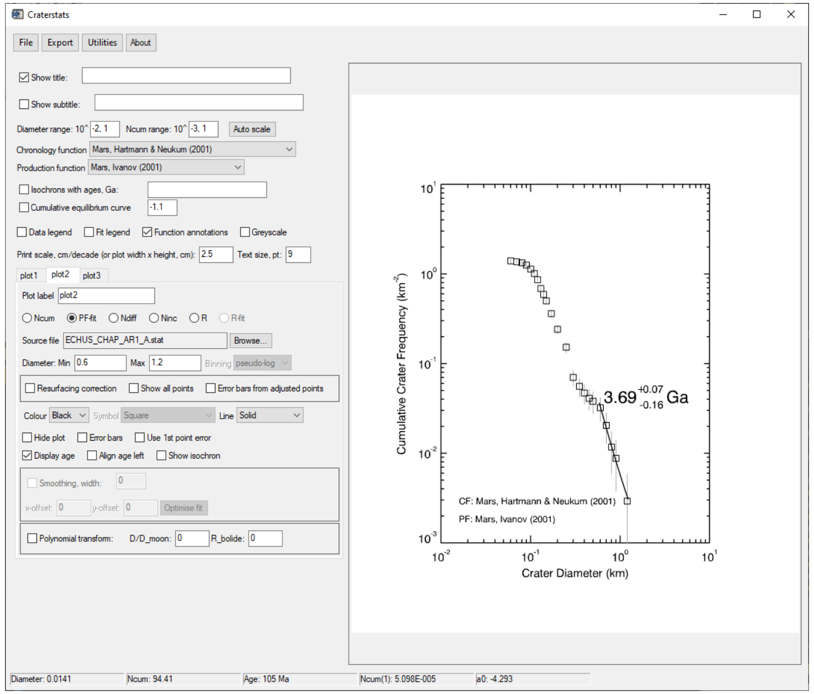
Comprehensive open source replacement for *CraterTools* needed
- in-GIS functions to aid crater-marking
- probably better to process shapefiles outside GIS to be independent of choice of GIS software

Analysis software - *Craterstats*

2008: first version
- replicated Gerhard Neukum's workflow which used measurements taken from photographic plates with a *stereo comparator*
- instead using crater measurements from GIS environment
    - the companion program, **CraterTools**, was written for this

2011: second version
- extended to allow direct comparison between common data presentations: cumulative, incremental, R-plot and differential

Scientific developments:
- resurfacing corrections (2010)
- spatial randomness and clustering analysis (2012)
- fitting of differential form of production function; compensation of exponential binning bias (2013)
- Poisson timing analysis (2016)

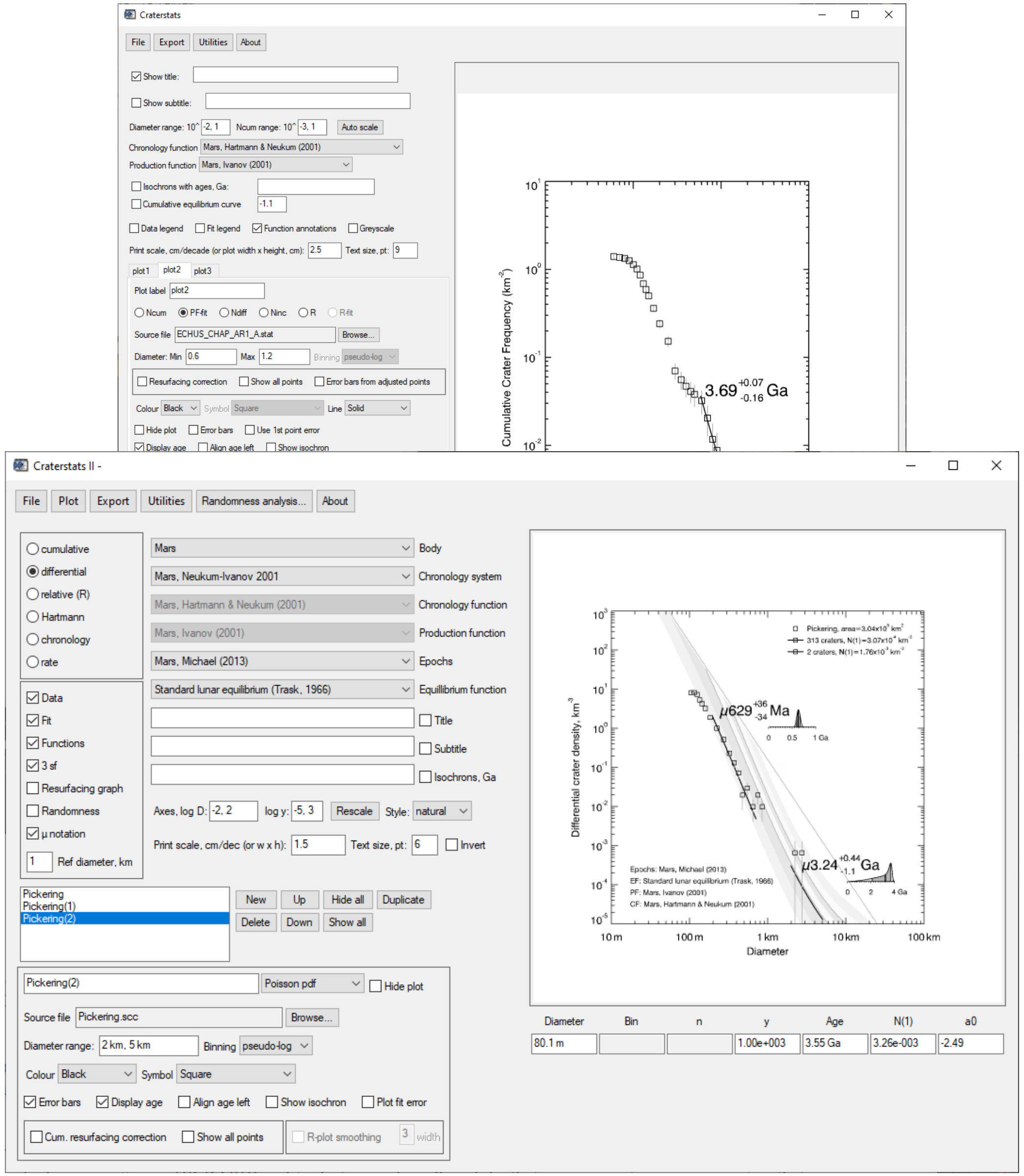


Analysis software - *Craterstats*

2008: first version
- replicated Gerhard Neukum’s workflow which used measurements taken from photographic plates with a *stereo comparator*
- instead using crater measurements from GIS environment
  - the companion program, **CraterTools**, was written for this

2011: second version
- extended to allow direct comparison between common data presentations: cumulative, incremental, R-plot and differential

Scientific developments:
- resurfacing corrections (2010)
- spatial randomness and clustering analysis (2012)
- fitting of differential form of production function; compensation of exponential binning bias (2013)
- Poisson timing analysis (2016)



Analysis software - *Craterstats*

2021: third version

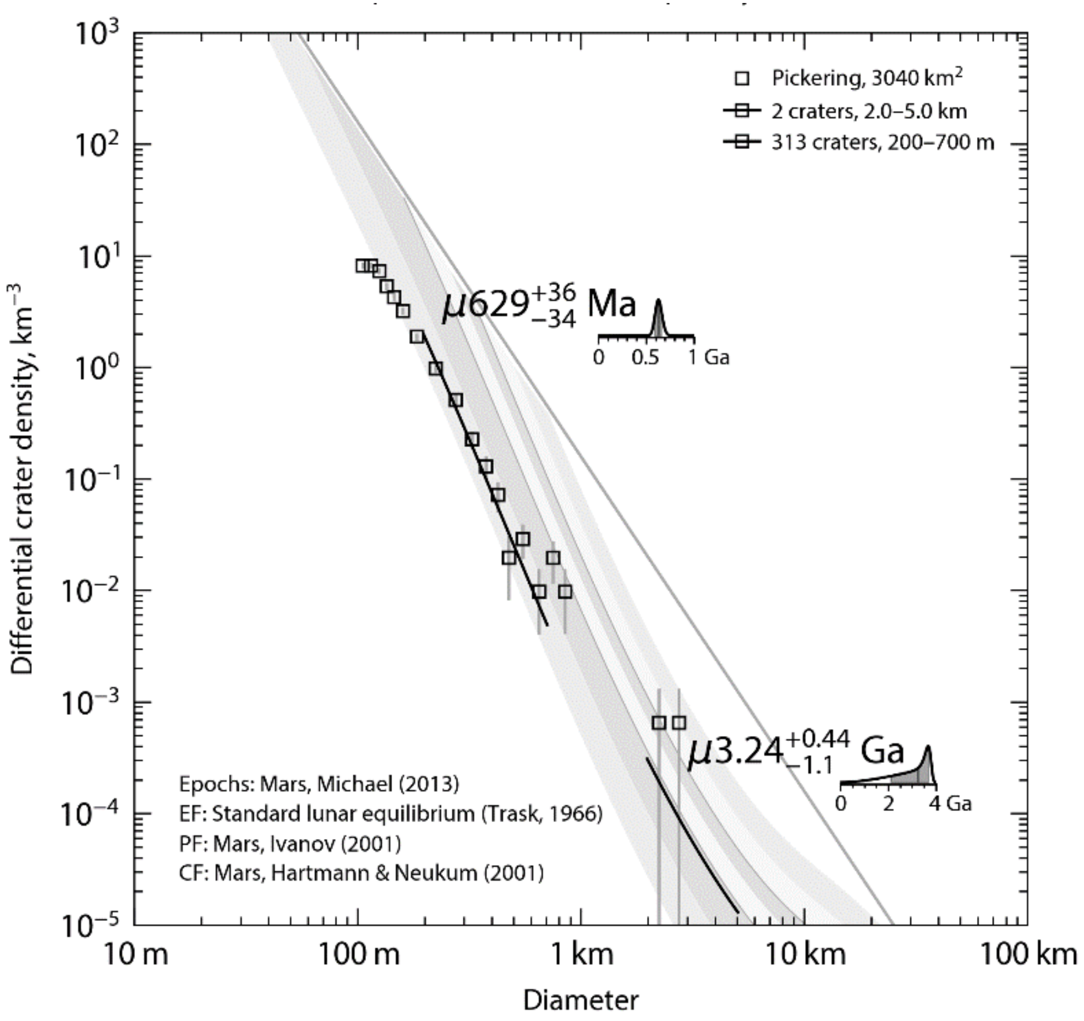
- Design goals
    - Rewrite in non-proprietory language (Python), make open source
    - Provide reference implementation for standard crater-count analysis calculations
    - Produce graphical output consistent with previous version (now using matplotlib)
    - Construct as analysis library which may be integrated into other software
    - Reproduce previous functionality using command line interface

- Updated features
    - More flexible control of legend information
    - Adjustable positioning of age annotations
    - Vector or raster graphic files; transparent backgrounds; inverted colour schemes for slides

- Ongoing development and maintenance transferred to this version

Usage example – differential plot

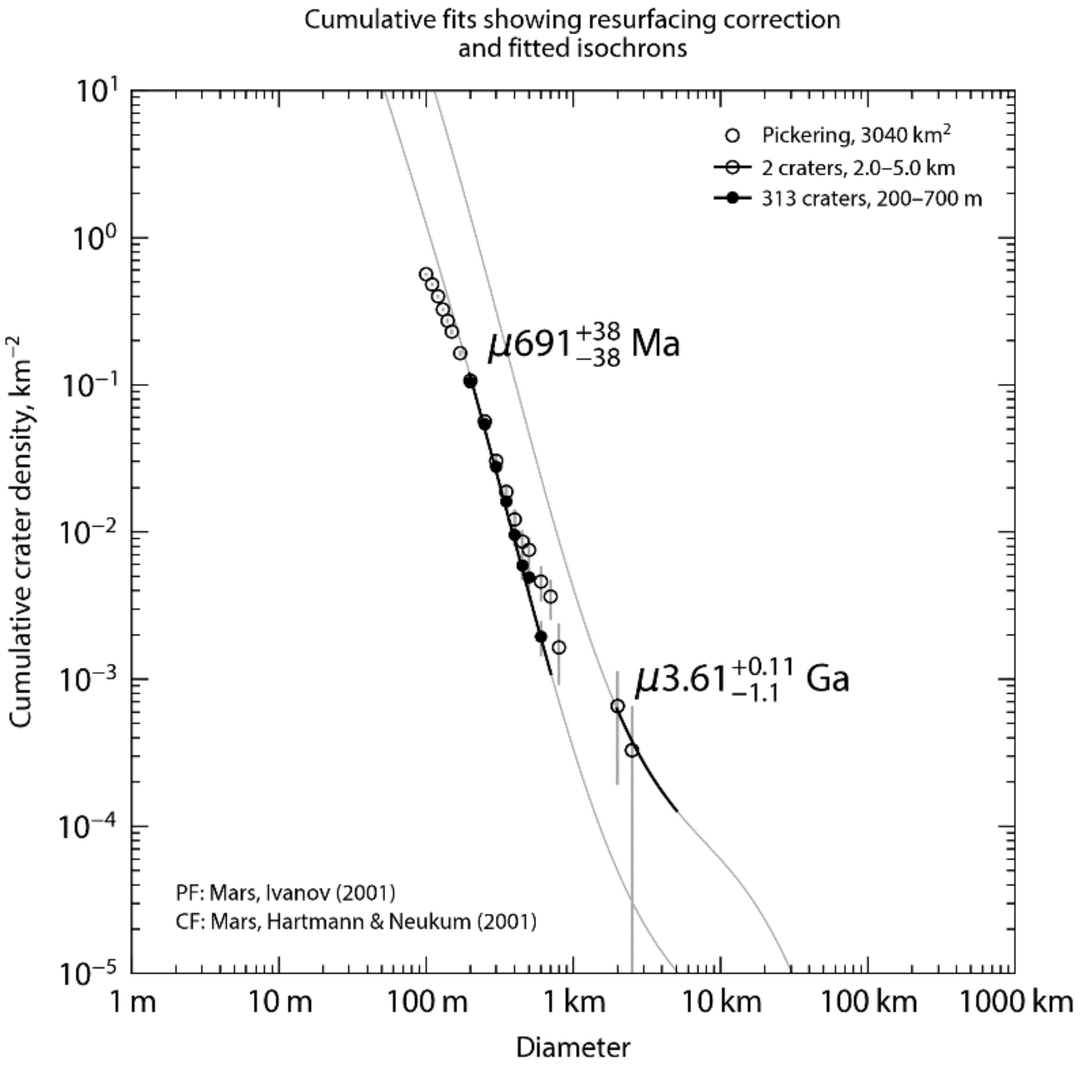
Command line option	Meaning
-cs HN01	chronology system – Mars, Hartmann and Neukum 2001
-ep mars	epoch system – Mars
-ef trask	equilibrium function – Trask 1966
-p	Start overplot definition…
source=xxx	Specify crater count file
-p	Next overplot definition…
type=poisson	Poisson age estimation
range=[2,5]	diameter range
offset_age=[2,-2]	*x,y* adjustment from default positioning of age label (in units of 1/20th decade)
-p	Next overplot definition…
range=[.2,.7]	diameter range




```
python craterstats.py -cs HN01 -ep mars -ef trask -p source=craterstats/sample/Pickering.scc
-p type=poisson,range=[2,5],offset_age=[2,-2] -p range=[.2,.7]
```

Usage example – cumulative plot (1)

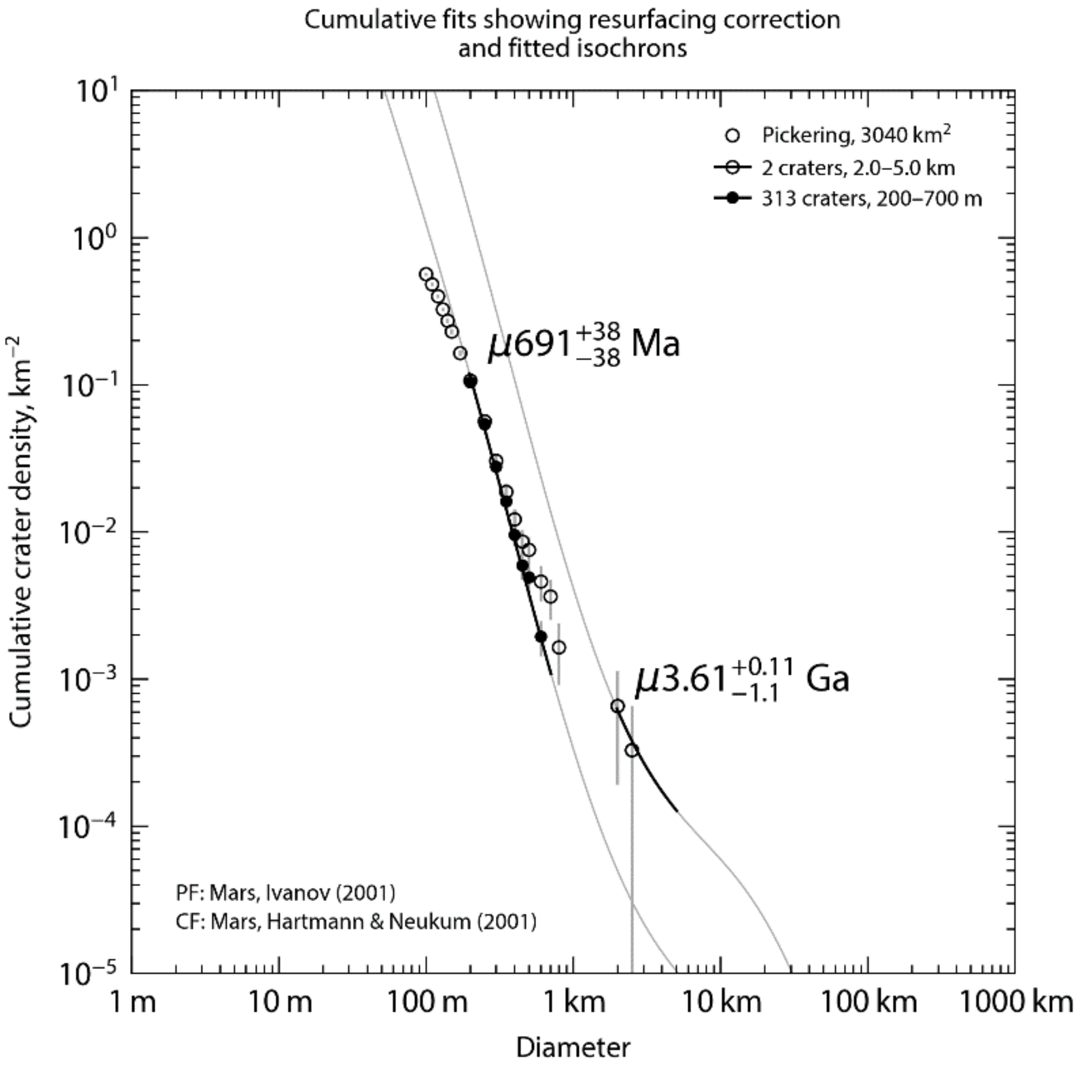
Command line option	Meaning
-pr cumul	plot style index - cumulative
-cs HN01	chronology system – Mars, Hartmann & Neukum 2001
-title xxx	*"Cumulative fits showing resurfacing correction"*
-subtitle xxx	*"and fitted isochrons"*
-p	Start overplot definition…
source=xxx	specify crater count file
psym=1	empty circles




```
python craterstats.py -pr cumul -cs HN01 -title Cumulative fits showing resurfacing correction -subtitle and fitted isochrons
 -p source=craterstats/sample/Pickering.scc,psym=1 -p type=c-fit,range=[2,5],isochron=1 -p range=[.2,.7],resurf=1,psym=10
```

Usage example – cumulative plot (2)

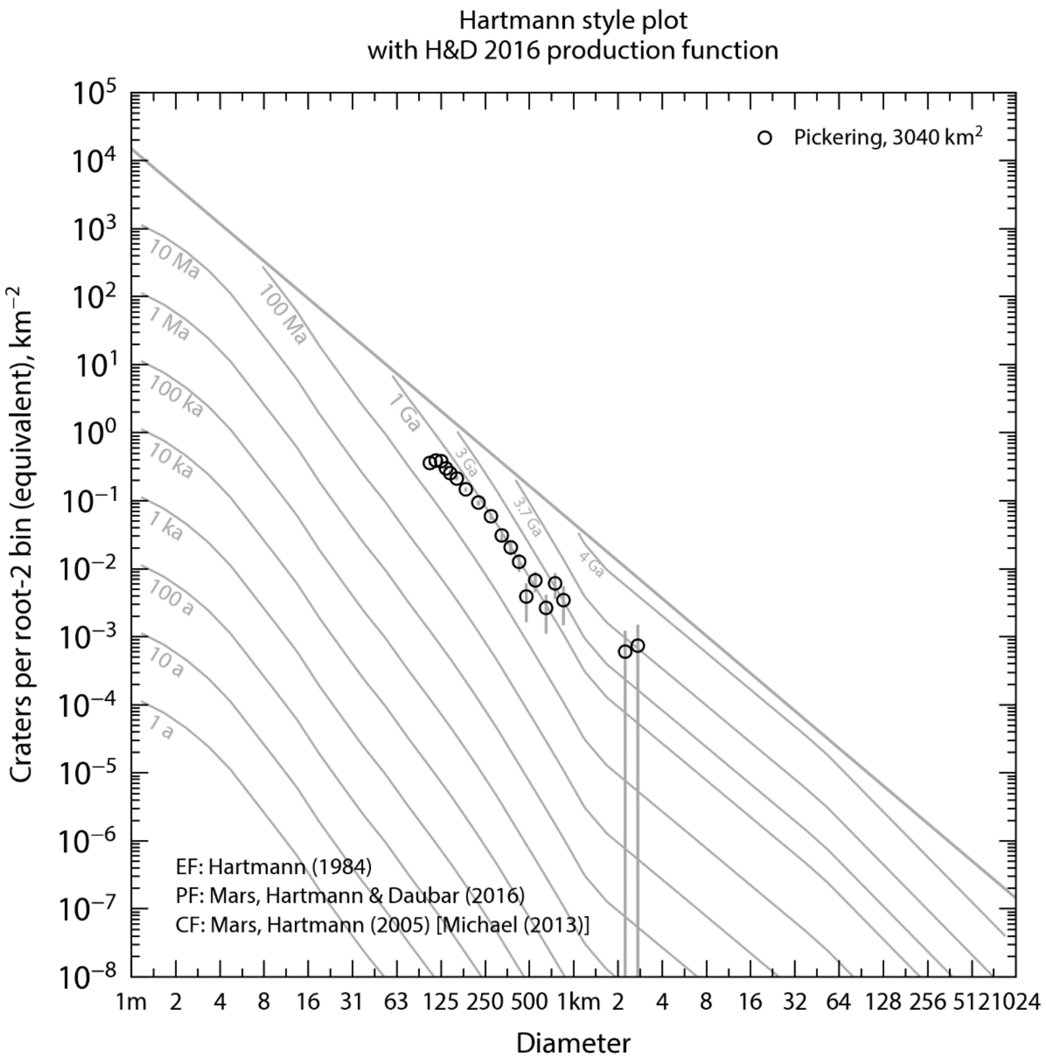
Command line option	Meaning
-p	Next overplot…
type=c-fit	cumulative fit
range=[2,5]	2-5 km diameter range
isochron=1	show complete isochron
-p	Next overplot…
range=[.2,.7]	diameter range
resurf=1	apply cumulative resurfacing correction
psym=10	filled circles




```
python craterstats.py -pr cumul -cs HN01 -title Cumulative fits showing resurfacing correction -subtitle and fitted isochrons
 -p source=craterstats/sample/Pickering.scc,psym=1 -p type=c-fit,range=[2,5],isochron=1 -p range=[.2,.7],resurf=1,psym=10
```

Usage example – Hartmann plot

Command line option	Meaning
-pr hartmann	plot style index - Hartmann
-cs hd16	chronology system – Mars, Hartmann & Daubar 2016
-ef h84	equilibrium function - Hartmann 1984
-title xxx	*"Hartmann style plot"*
-subtitle xxx	*"with H&D 2016 production function"*
-isochrons xx,yy,zz	isochron values in Ga (s suffix – small font)
-p	Start overplot definition…
source=xxx	Specify crater count file
psym=1	empty circles




```
python craterstats.py -pr hartmann -cs hd16 -ef h84 -title Hartmann style plot -subtitle with H&D 2016 production function
 -isochrons 4s,3.7s,3s,1,.1,.01,.001,1e-4,1e-5,1e-6,1e-7,1e-8,1e-9 -p source=craterstats/sample/Pickering.scc,psym=1
```

Usage example – results table

Command line option	Meaning
-f txt	Output results in text format

name	area	binning	d_min	d_max	method	resurf	n	n_event	age	age-	age+	a0	a0-	a0+	N(1)
Pickering	3036.6	pseudo-log	0.2	0.7	d-fit	0	313	313	0.668	0.613	0.722	-3.488	-3.525	-3.453	3.25E-04
Pickering	3036.6	pseudo-log	2	5	d-fit	0	2	2	3.76	0.0475	3.88	-2.063	-4.635	-1.762	8.66E-03

```
python craterstats.py -f txt -cs hn01 -p source=craterstats/sample/Pickering.scc -p type=d-fit,range=[.2,.7] -p range=[2,5]
```

Chronology systems and other functions

Chronology systems
1 Moon, Neukum (1983)
2 Moon, Neukum et al. (2001)
3 Mars, Hartmann & Neukum (2001)
4 Mars, Ivanov (2001)
5 Mars, Hartmann 2004 iteration
6 Mars, Hartmann & Daubar (2016)
7 Mercury, Strom & Neukum (1988)
8 Mercury, Neukum et al. (2001)
9 Mercury, Le Feuvre and Wieczorek 2011 non-porous
10 Mercury, Le Feuvre and Wieczorek 2011 porous
11 Vesta, Rev4, Schmedemann et al (2014)
12 Vesta, Rev3, Schmedemann et al (2014)
13 Vesta, Marchi & O'Brien (2014)
14 Ceres, Hiesinger et al. (2016)
15 Ida, Schmedemann et al (2014)
16 Gaspra, Schmedemann et al (2014)
17 Lutetia, Schmedemann et al (2014)
18 Phobos, Case A - SOM, Schmedemann et al (2014)
19 Phobos, Case B - MBA, Schmedemann et al (2014)

Equilibrium functions
1 Standard lunar equilibrium (Trask, 1966)
2 Hartmann (1984)

Epoch systems
1 Moon, Wilhelms (1987)
2 Mars, Michael (2013)

- This list is generated with:

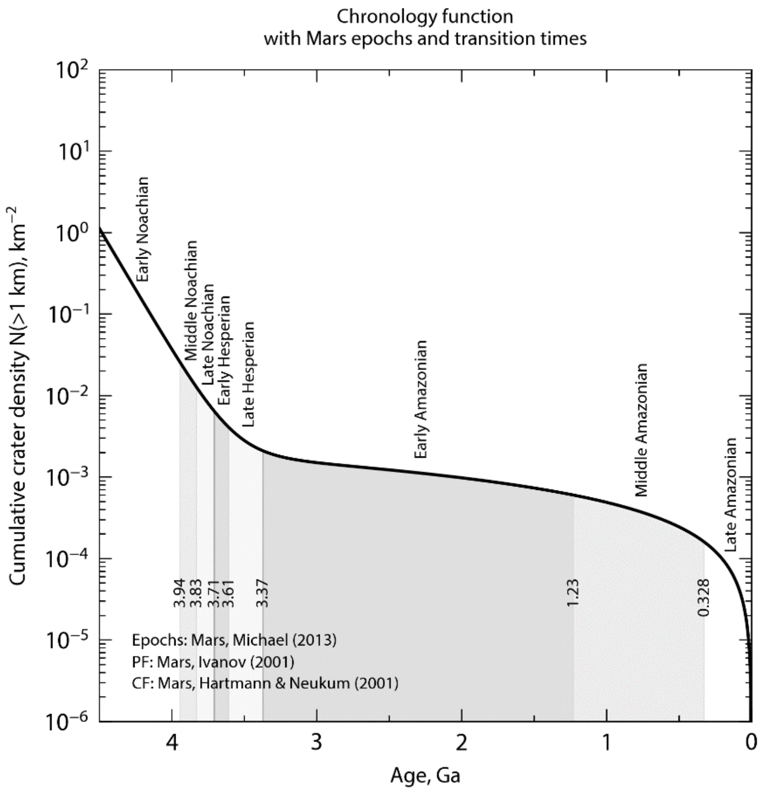
```
python craterstats.py -lcs
```

- new functions can be added by the user

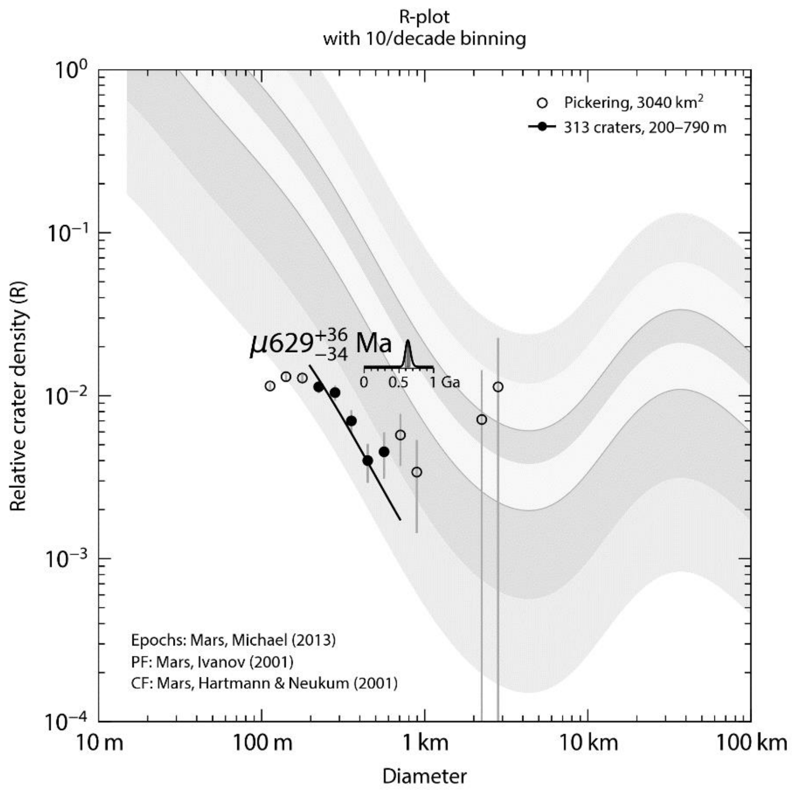
- similarly, plot symbols and colours can be listed with:

```
python craterstats.py -lpc
```

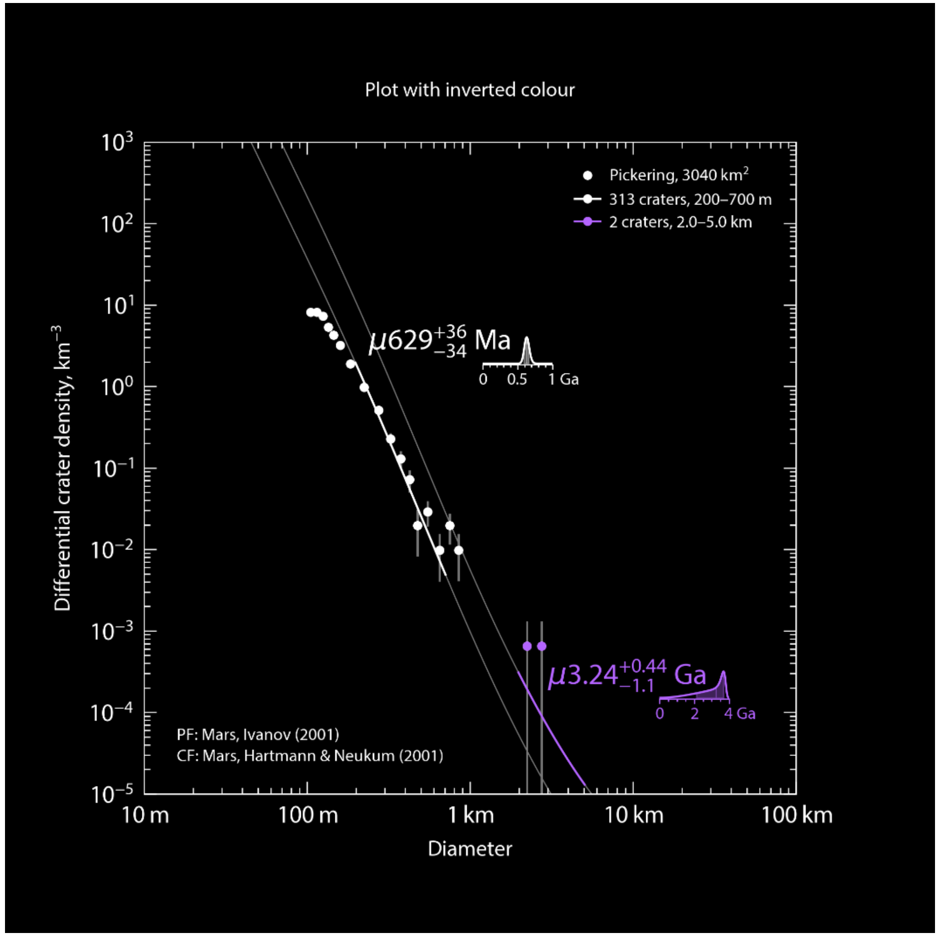
Other examples



Chronology function

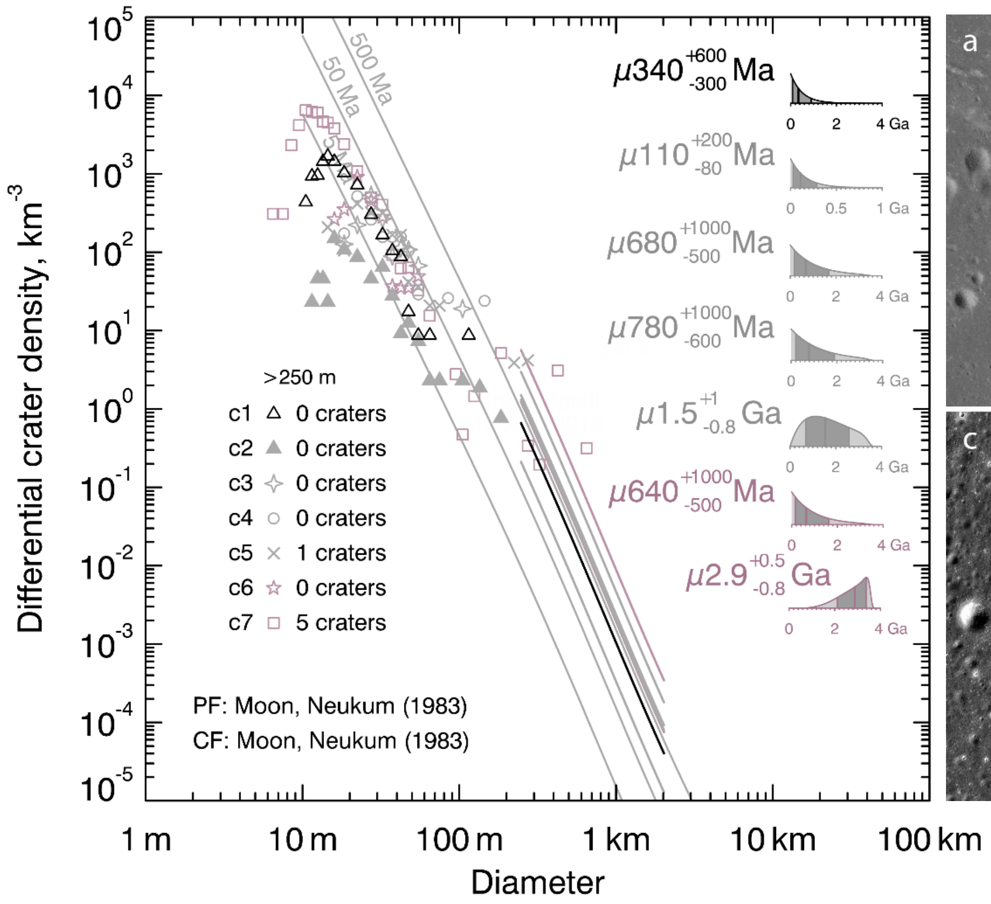


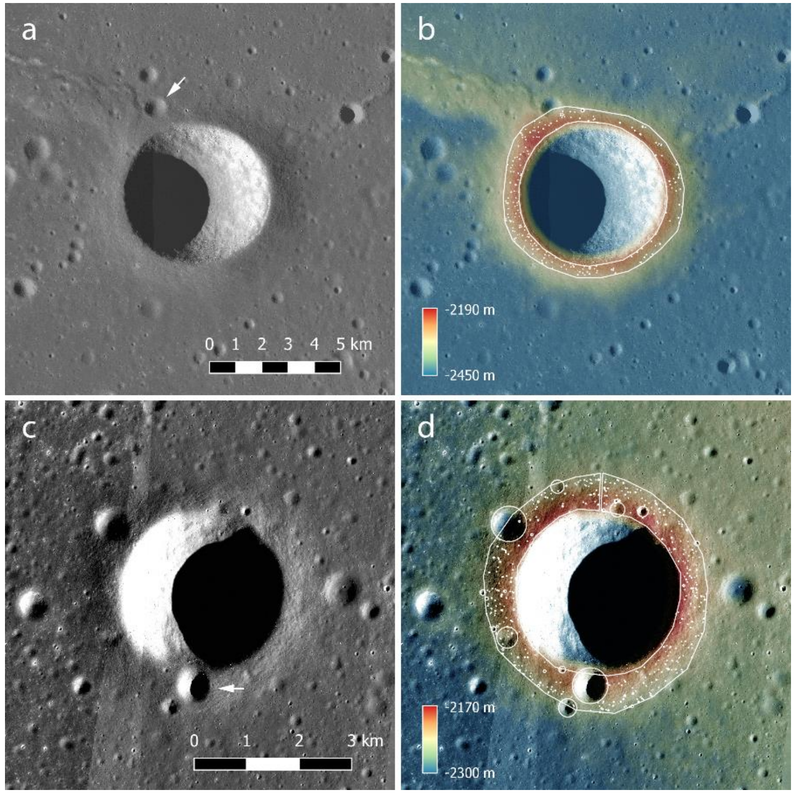
R-plot



Inverted colour scheme
for dark slides

Latest addition - Poisson age-likelihood calculation for a buffered crater counting area





- Probabilistic constraints even from very few craters
- These results for several km craters close to information limit
- New code added into *Craterstats3*
- Requires an extra measurement of the counting area perimeter (*CraterTools* doesn't record this)
- Add line into .scc file with `perimeter=xxx` (in km)

Michael, G., Yue, Z., Gou, S., Di, K., 2021. *Dating individual several-km lunar impact craters from the rim annulus in region of planned Chang'E-5 landing: Poisson age-likelihood calculation for a buffered crater counting area*. EPSL 568.

Where to find it?

https://github.com/ggmichael/craterstats

- The PlanetaryPy project have been helping me standardise the installation procedure for the *Craterstats* package. This is not working exactly as written here now, but will be updated soon!

I’m glad to help with any usage difficulties!
gregory.michael@fu-berlin.de

README.md

Craterstats

This is a reimplementation in Python 3.8 of the CraterstatsII software, a tool to analyse and plot crater count data for planetary surface dating.

Installation

There are various ways to install Python. If you are installing it specifically to run Craterstats, the following is suggested:

1. Install Python 3.8 or higher from python.org.
2. Download Craterstats and unzip the files where you choose.
3. Open a command prompt/terminal window and `cd` to the newly-created `craterstats-main/`
4. Enter these commands to establish a Python *virtual environment:*

Windows:

```
py -m venv venv
venv\Scripts\activate.bat
py -m pip install -r requirements.txt
```

Linux:

```
python3 -m venv venv
source venv/bin/activate
python3 -m pip install -r requirements.txt
```

After you see the required python packages installed, the set-up is complete. In the remaining sections, you should substitute `py` or `python3` for `python` according to your system.

Quick demonstration

After installation, the following commands will produce a series of example output plots and data, demonstrating the main features of the software. Plot image files are placed into the subfolder `demo/`, while text output – including the full command lines as they could be typed to generate the output – goes to the terminal window.

```
cd src
python craterstats.py -demo
```

Packages

No packages published
Publish your first package

Languages

Python 99.2% Other 0.8%