Statistics and Machine Learning 1

Lecture 2A: Univariate Exploratory Data Analysis

Mark Muldoon
Department of Mathematics, Alan Turing Building
University of Manchester

Week 2

Motivation

- Imagine that your boss has just dropped a dataset in your inbox, and asked you to provide a report to her about what it means for the business. This kind of 'open ended' task is quite common!
- Where should you start? The answer is almost certainly with *exploratory data analysis* (EDA).
- EDA is about getting an intuitive understanding of the data, and as such different people will find different techniques useful.
- I am going to show you some of the things I do, but ultimately you need to work with data and find what works for you!
- Often statistics is a formal way of testing whether what the data seems to tell you is truly a strong signal or or could have arisen by chance. But you need first to decide what to test …

Data quality

The first thing understand is where the data come from and how accurate they are. Professor David Spiegelhalter (Cambridge Professor, OBE FRS) suggests giving data a 'star rating'. This is based on experience rather than any formal theory, but I like it and have adapted his ratings as below:

- 4⋆ – Numbers we can believe. Examples: official statistics; well controlled laboratory experiments.
- 3⋆ – Numbers that are reasonably accurate. Examples: well conducted surveys / samples; field measurements; less well controlled experiments.
- 2⋆ – Numbers that could be out by quite a long way. Examples: poorly conducted surveys / samples; measurements of very noisy systems.
- 1⋆ – Numbers that are unreliable. Examples: highly biased / unrepresentative surveys / samples; measurements using biased / low-quality equipment.
- 0⋆ – Numbers that have just been made up. Examples: urban legends / memes; fabricated experimental data.

Earthquakes dataset and its quality

- We are going to work with data from the United States Geological Survey.
- They're the result of a query on `earthquake.usgs.gov/earthquakes/search/` that sought all US events in a 30 day period, downloaded as a CSV file.
- The data are downloaded from the official website of a US Governmental agency.
- The CSV file does not need significant wrangling—this may correlate with high accuracy, but not reliably so.
- But even with advanced modern measurement techniques, seismology is complicated and as such should not be expected to be completely accurate or comprehensive (e.g. some smaller events may be completely missing).
- So I'd say they are 3⋆; I would take them at face value, but keep open the possibility of measurement error etc. in any analysis.

Packages

The original version of this lecture used only Python, but *in response to student feedback* it now includes R code as well. The key packages to use are:

- *Data wrangling:* Pandas

```
>>> import pandas as pd
```

- *Visualisation:* Seaborn (based on MatPlotLib)

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
```

- *Data wrangling:* dplyr (based on plyr)

```
> library(dplyr)
```

- *Visualisation:* GGPlot

```
> library(ggplot2)
```

Earthquakes dataset into Python

Pulling this file in to Python as a Pandas dataframe produces:

```
In [15]: # Read in the data as a Pandas frame
         f = pd.read_csv('./earthquakes_US_14Jul-13Aug_2018.csv')
         f
```

Out[15]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	type	horizontalErr
0	2018-08-13T20:32:00.830Z	33.860333	-117.501167	1.50	1.56	ml	29.0	44.00	0.063370	0.2600	...	2018-08-13T20:35:51.429Z	3km SE of Home Gardens, CA	quarry blast	0.
1	2018-08-13T20:15:08.350Z	33.523667	-116.747333	3.69	0.40	ml	14.0	90.00	0.042460	0.0800	...	2018-08-13T20:18:36.325Z	8km WSW of Anza, CA	earthquake	0.
2	2018-08-13T20:11:13.973Z	40.601000	-115.955500	0.00	2.20	ml	12.0	108.30	0.438000	0.1944	...	2018-08-13T20:28:20.650Z	17km SE of Carlin, Nevada	explosion	Na
3	2018-08-13T20:09:39.080Z	33.482333	-116.744333	7.91	0.34	ml	18.0	65.00	0.065270	0.1800	...	2018-08-13T20:13:23.288Z	10km SW of Anza, CA	earthquake	0.
4	2018-08-13T19:59:26.490Z	33.491167	-116.795500	2.54	0.39	ml	11.0	109.00	0.088570	0.0800	...	2018-08-13T20:02:57.234Z	8km NE of Aguanga, CA	earthquake	0.
5	2018-08-13T19:44:27.930Z	33.158667	-116.520167	8.01	0.78	ml	22.0	82.00	0.101100	0.2100	...	2018-08-13T19:48:11.784Z	12km NE of Julian, CA	earthquake	0.

Earthquakes dataset into R

Pulling this file in to R as a table produces:

Console Terminal

~/work/teaching/ds/class6/

```
>
>
> f <- read.table("earthquakes_US_14Jul-13Aug_2018.csv", header=TRUE, sep=",")
> f
                       time latitude  longitude  depth  mag magType nst    gap      dmin    rms
1    2018-08-13T20:32:00.830Z 33.86033 -117.50117  1.500 1.56      ml  29  44.00 0.0633700 0.2600
2    2018-08-13T20:15:08.350Z 33.52367 -116.74733  3.690 0.40      ml  14  90.00 0.0424600 0.0800
3    2018-08-13T20:11:13.973Z 40.60100 -115.95550  0.000 2.20      ml  12 108.30 0.4380000 0.1944
4    2018-08-13T20:09:39.080Z 33.48233 -116.74433  7.910 0.34      ml  18  65.00 0.0652700 0.1800
5    2018-08-13T19:59:26.490Z 33.49117 -116.79550  2.540 0.39      ml  11 109.00 0.0885700 0.0800
6    2018-08-13T19:44:27.930Z 33.15867 -116.52017  8.010 0.78      ml  22  82.00 0.1011000 0.2100
7    2018-08-13T19:26:27.550Z 33.94967 -116.84283 13.930 0.78      ml  16  81.00 0.0901000 0.2200
8    2018-08-13T19:25:39.810Z 37.98950 -122.45216  1.680 1.34      md   9 182.00 0.0217200 0.1200
9    2018-08-13T19:08:08.240Z 46.02233 -112.47317 -2.000 1.50      ml  10 106.00 0.0590000 0.1700
10   2018-08-13T18:55:23.170Z 40.29583 -124.50383 20.180 2.81      md  21 252.00 0.1699000 0.0800
11   2018-08-13T18:36:37.800Z 37.32617 -121.69450  5.310 1.16      md   9  85.00 0.0441800 0.0300
12   2018-08-13T18:16:07.790Z 33.09583 -116.03017  9.640 1.31      ml  23  44.00 0.1301000 0.2500
```

Univariate Data Vectors

- We are going to work today with the *univariate* case—one measurement per 'thing'—so for the earthquakes we might concentrate on the magnitudes alone.

- We pull these into a NumPy array using code like:

```
>>> x = f.mag.values
```

- We pull these into an R array using code like:

```
> x <- f$mag
```

Notation

- Mathematically, we represent a univariate dataset as a length-n vector,

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)$$

- The *sample mean* of a function $f(x)$ is

$$\langle f(x) \rangle \;=\; \frac{1}{n} \sum_{i=1}^{n} f(x_i) \;=\; \frac{1}{n} \left[f(x_1) + f(x_2) + \cdots + f(x_n) \right] \qquad (1)$$

  which is, strictly speaking, a function of the vector $\mathbf{x}$, but we will use the simpler notation above. I've also introduced a notation that we'll use again: angle-braces $\langle \cdot \rangle$ to indicate the average over the data of the thing enclosed.

Visualisation and Information

There is an important distinction in visualisations between:

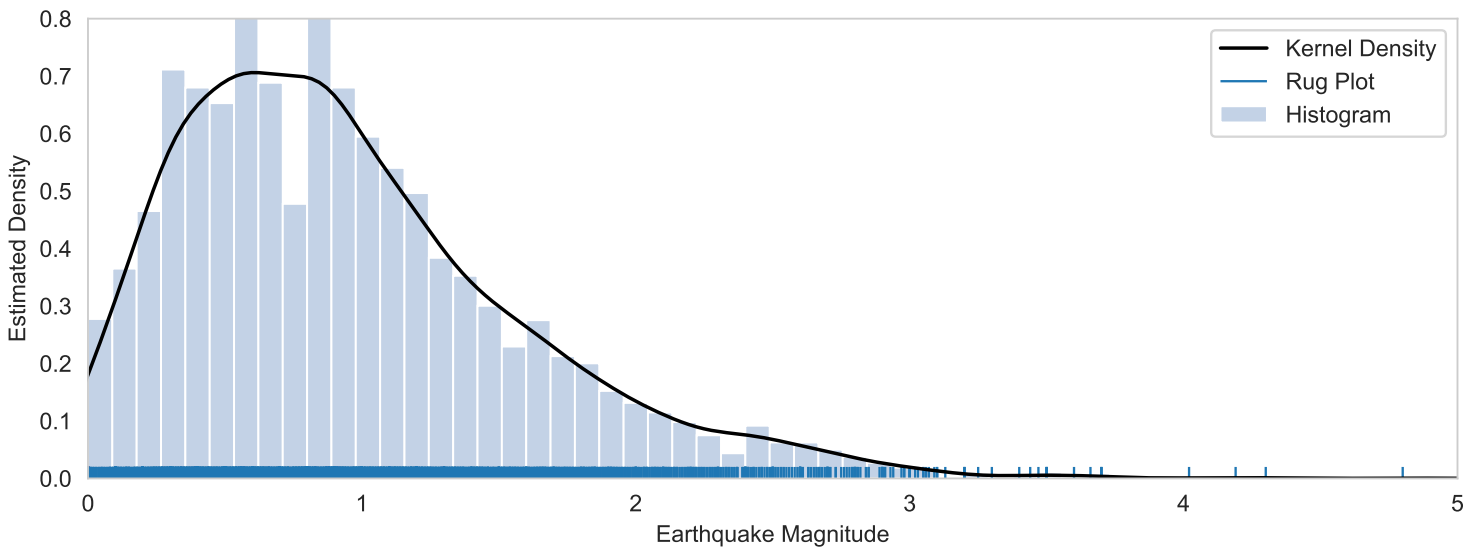
- *Lossless* ones from which, if viewed at sufficiently high resolution, one could recover the original dataset;
- *Lossy* ones, where a given plot would be consistent with many different raw datasets.

Typically for complex data, choosing the lossy visualistaion that loses the 'right' information is key to successful visualisation.

Main visualisations in Python

Three univariate visualisations: the *rug plot*, which has a small vertical tick on the x-axis for each data point and is hence lossless, but here is not very useful for anything other than the values $\gtrsim 3$; and the lossy *histogram* and *kernel density* plots. The full code is online, but the basic sequence of commands is:

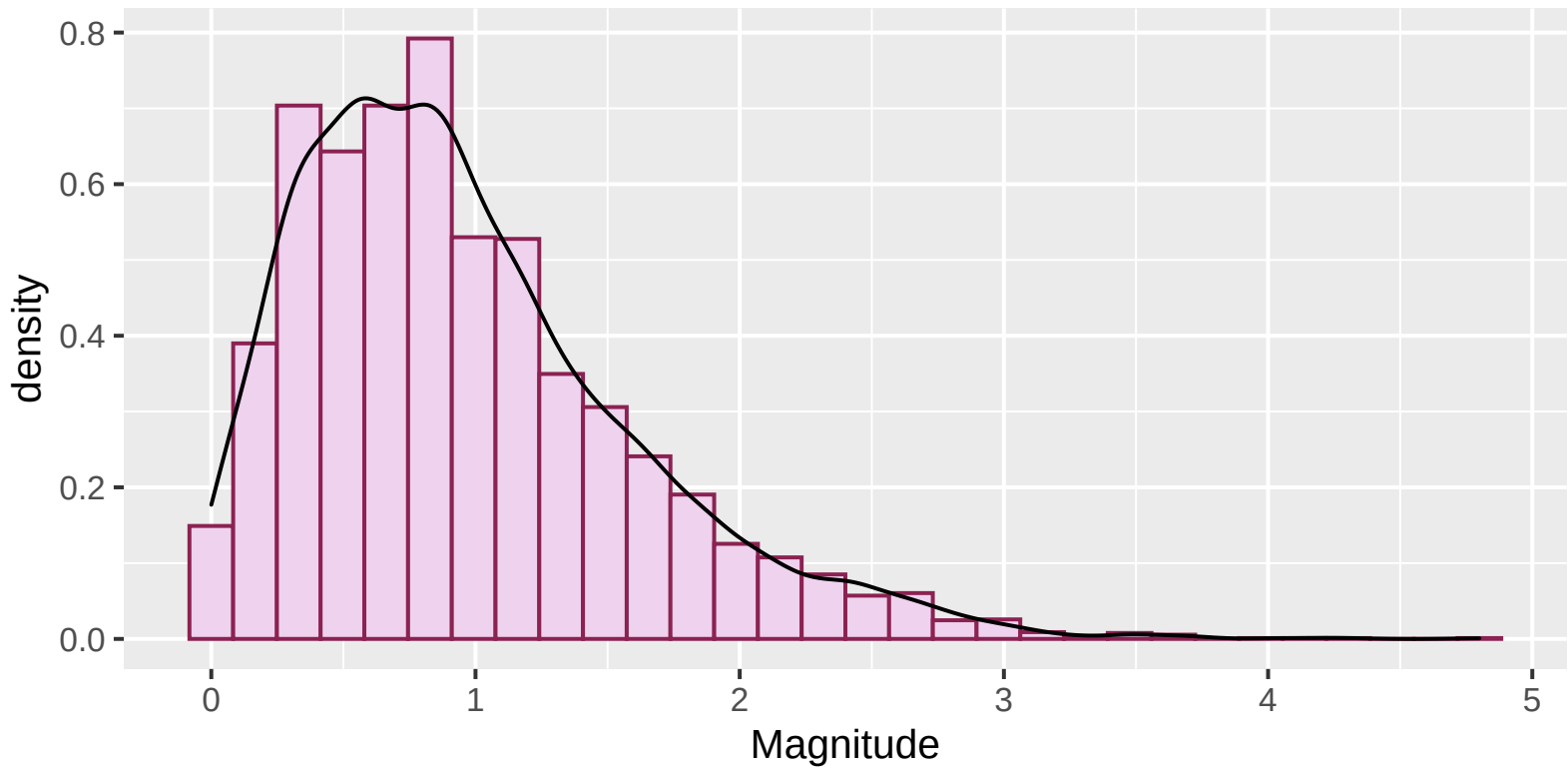
```
sns.histplot( magnitudes, stat="density" )
sns.kdeplot( magnitudes )
sns.rugplot( magnitudes )
```



Main visualisations in R

Here we also have *histogram* and *kernel density* plots, produced using code similar to

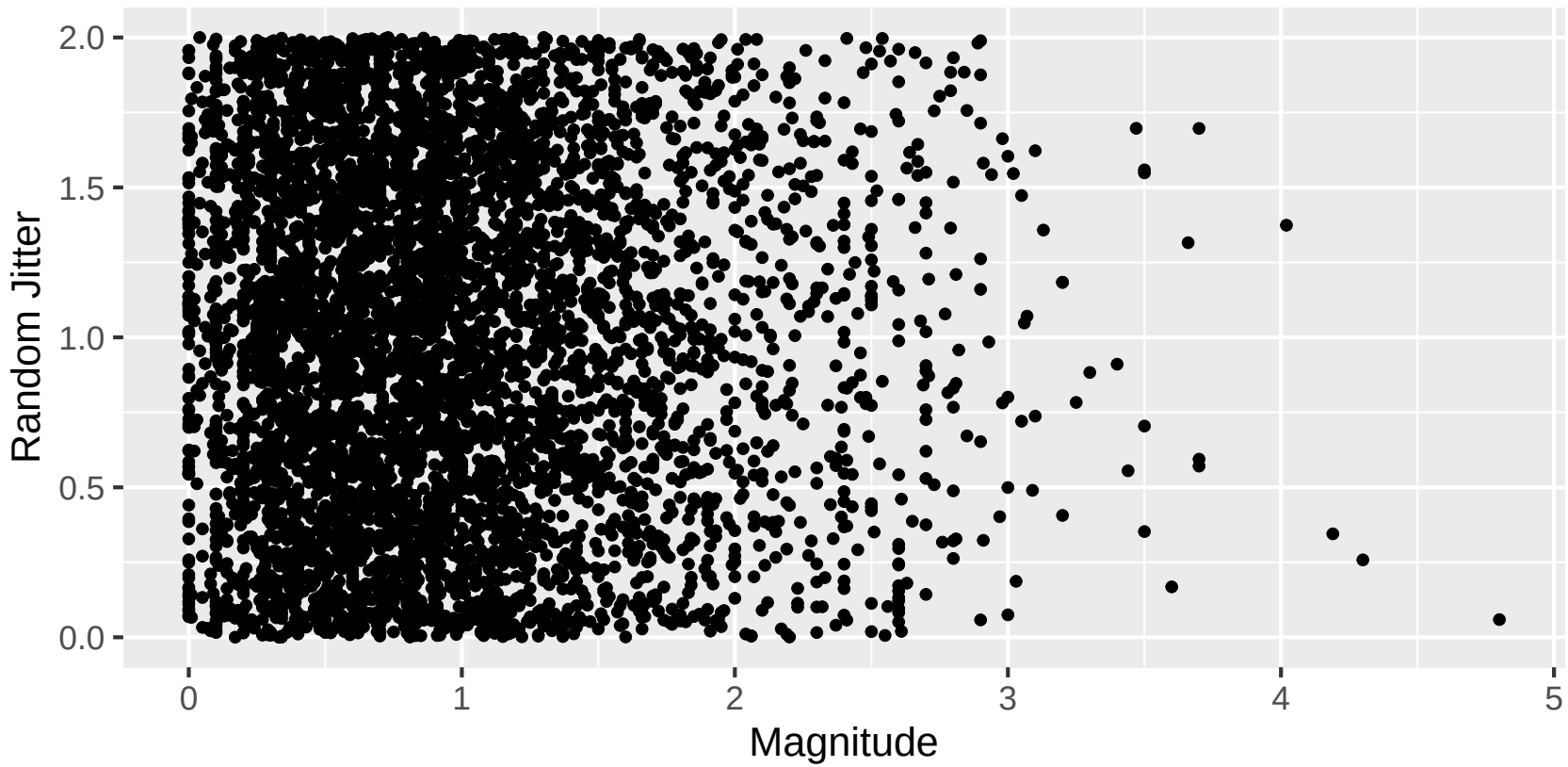
```
ggplot(earthquake.df, aes(x=mag)) +
geom_histogram(aes(y=..density..)) +
geom_density() + xlab("Magnitude")
```



Jitter in R

In ggplot2, we also have the lossless *jitter plot* as an alternative to the rug:

```
ggplot(earthquake.df, aes(x=mag,y=rep(1,n),stroke=0)) +
geom_jitter(width=0,height=1)
```



Further reading

Much of EDA rests on visualisation, a huge topic that depends not only on statistics, but also psychology and graphic design.

- One of the earliest books about EDA is: J.W. Tukey (1977), *Exploratory Data Analysis*, Addison-Wesley. ISBN: 9780201076165.
- Perhaps the most celebrated guide to statistical graphics is E. R. Tufte (2001), *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT, 2nd ed., 2001.
- There are online galleries of visualisations, with code, available for python's `matplotlib` and in the `R Graph Gallery`.
- There's a Canadian publisher, `Visual Capitalist`, who makes excellent visualisations about all sorts of data.
- David McCandless has written a book about presenting data entitled *Information is Beautiful* and runs a business by the same name. Their `web site` has lots of gorgeous visualisations, as well as an interesting collection of `data sets`