Introduction to Geostatistics

6. Sampling: strategies, point estimation, confidence intervals

Edzer J. Pebesma

edzer.pebesma@uni-muenster.de
Institute for Geoinformatics (**ifgi**)
University of Münster

summer semester 2007/8,
June 9, 2008

ifgi

Sampling

Sampling can be

- Random (any population elements can enter the sample with a given, non-zero probability)
- Non-random (population elements enter the sample with an unknown, or with zero probability)

ifgi

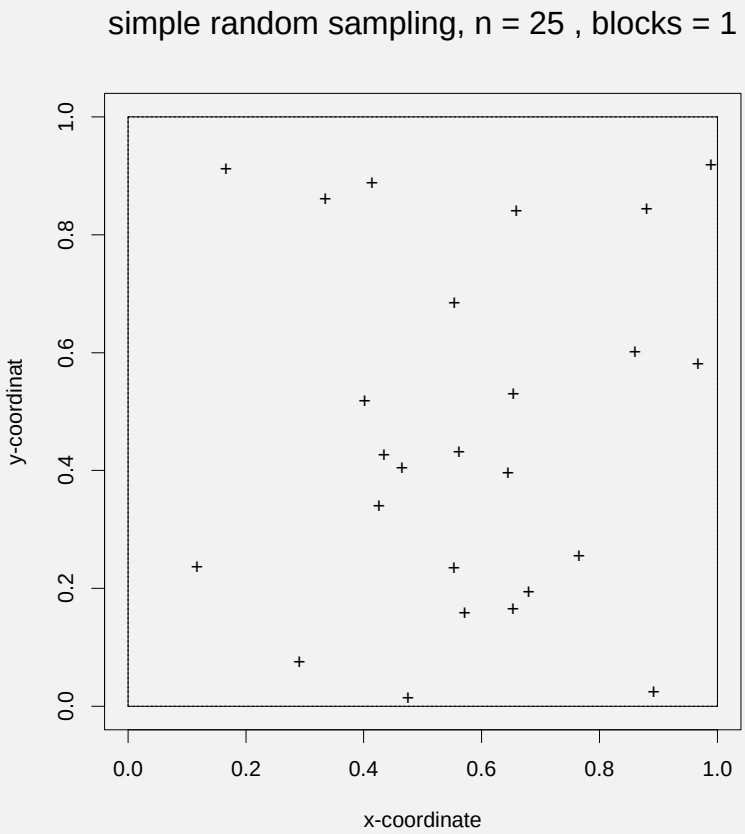
Examples of non-random sampling

Three examples:

- no random process was used to generate the sampling locations
- part of the area had zero inclusion probability.
  Suppose we want to sample vegetation in NRW. For this, we choose 100 sites randomly, but in forests only. We can now estimate e.g. which fraction of the NRW forests are deciduous (population: forests in NRW), but we cannot estimate which area of NRW consists of forests (population: all of NRW; the non-forest locations had zero inclusion probability in the sample.)

The question remains what to do when a particular sample *could* have come from a random process, but didn't.

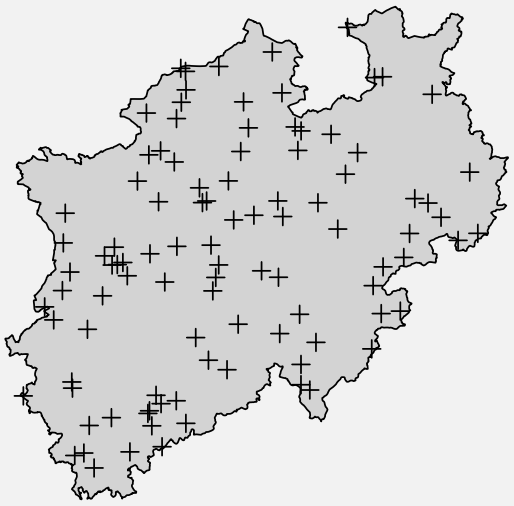
ifgi



ifgi

Simple random sampling (1)

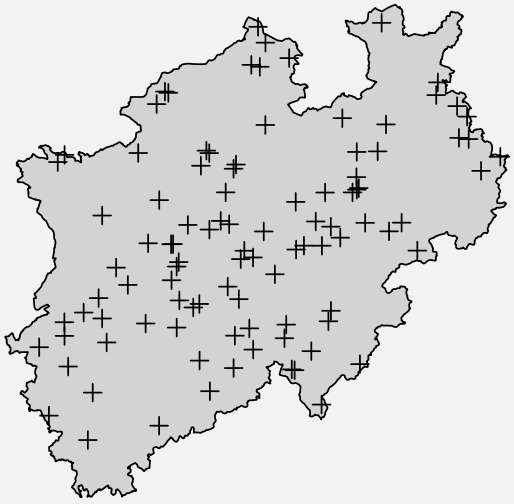
```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "random")
> points(pts, pch = 3)
```



ifgi

Simple random sampling (2)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "random")
> points(pts, pch = 3)
```



ifgi

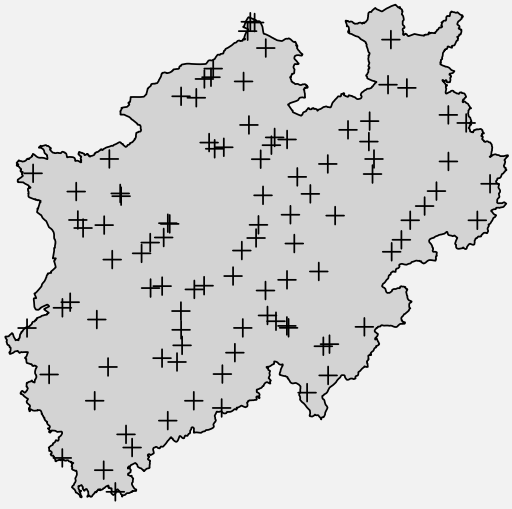
Simple random sampling (3)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "random")
> points(pts, pch = 3)
```

ifgi

Simple random sampling (4)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "random")
> points(pts, pch = 3)
```



ifgi

Simple random sampling (5)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "random")
> points(pts, pch = 3)
```

ifgi

Properties and strategy

with simple random sampling,

- every element has identical probability to enter the population
- every point is drawn independently from the others
- as the number of points in an area is infinite, replacement is not an issue (in theory; in practice, the choice may be quite constrained)
- this makes the sample elements completely independent (in a *design-based perspective*).
- How to do this? For an area A, consider its bounding box B. For n points, repeat
  1. draw a uniform random coordinate x' from the x-range of B
  2. draw a uniform random coordinate y' from the y-range of B
  3. accept (x', y') if it is inside (or on) A (point-in-polygon)

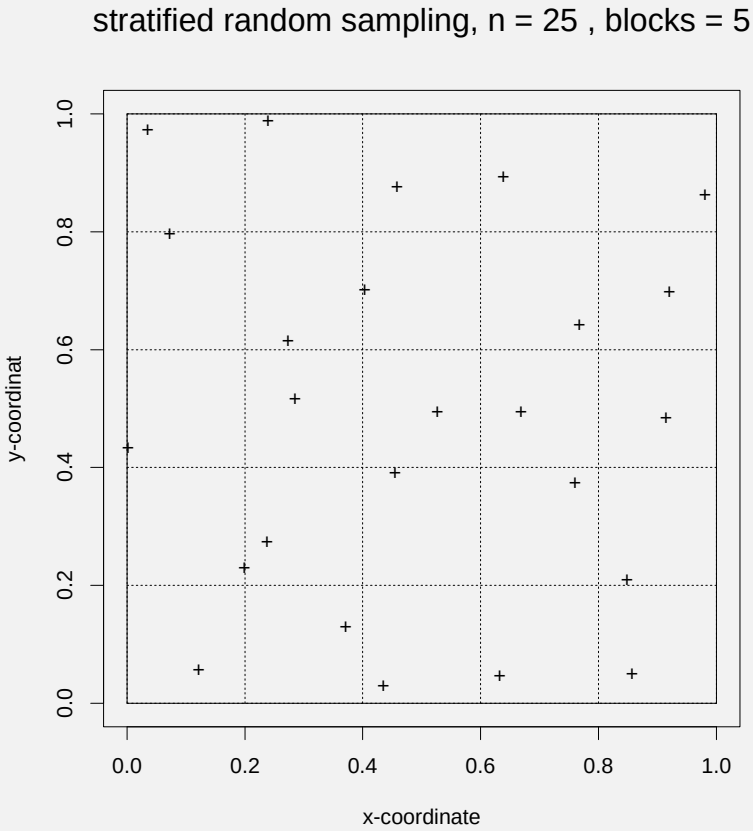
  until we have accepted n points

ifgi

Other spatial sampling approaches

- Random sampling over another agent then space: (e.g. if you randomly sample people, the spatial pattern of selected persons will follow the population density pattern.)
- Spatial random sampling that uses map information to vary densities (e.g., for a bird inventory sample forest locations with 0.05 pts/km, agricultural locations with 0.01 pts/km, urban areas with 0.005 pts/km; density may depends on expected variety and viewing conditions)
- Spatially homogeneous, but non-simple random sampling (Ripley, 1981, Spatial Statistics):
  - Stratified random sampling
  - Regular (systematically aligned) sampling
  - Non-aligned random sampling
  - Clustered sampling

ifgi



ifgi

Stratified random sampling (1)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "stratified")
> points(pts, pch = 3)
```

ifgi

Stratified random sampling (2)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "stratified")
> points(pts, pch = 3)
```

ifgi

Stratified random sampling (3)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "stratified")
> points(pts, pch = 3)
```

ifgi

Stratified random sampling (4)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "stratified")
> points(pts, pch = 3)
```

ifgi

Stratified random sampling (5)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "stratified")
> points(pts, pch = 3)
```

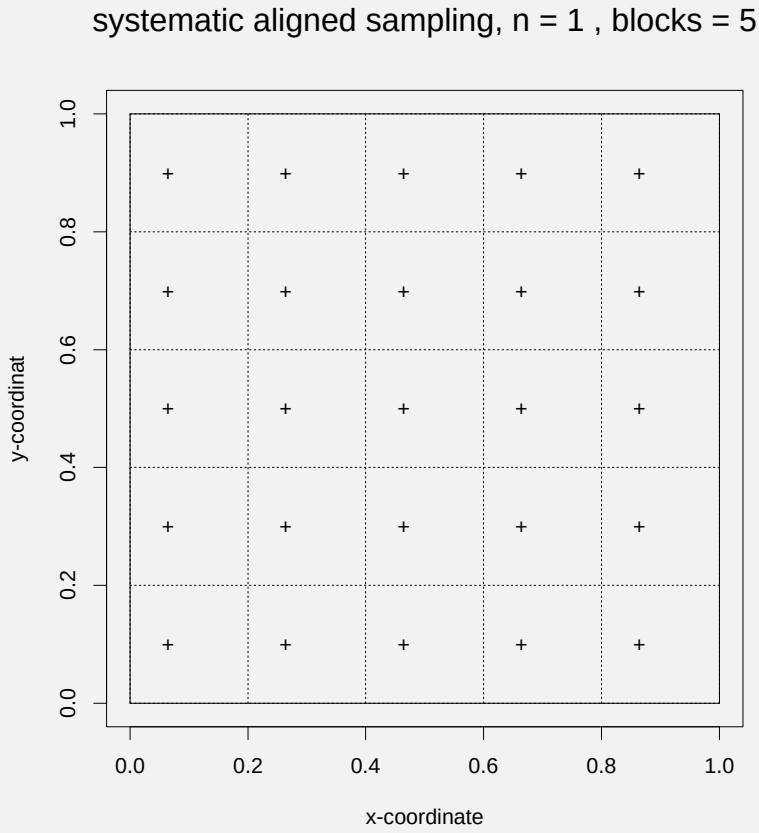
ifgi

Stratified random sampling

Approach:

- Put a lattice over the area, with (approximately) n cells
- Randomly sample one point in each cell
- Number of random coordinates: n, constrained to one per lattice cell

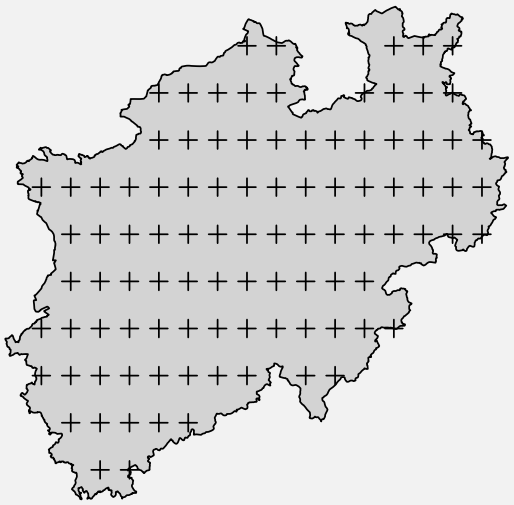
ifgi



ifgi

Systematically aligned random sampling (1)

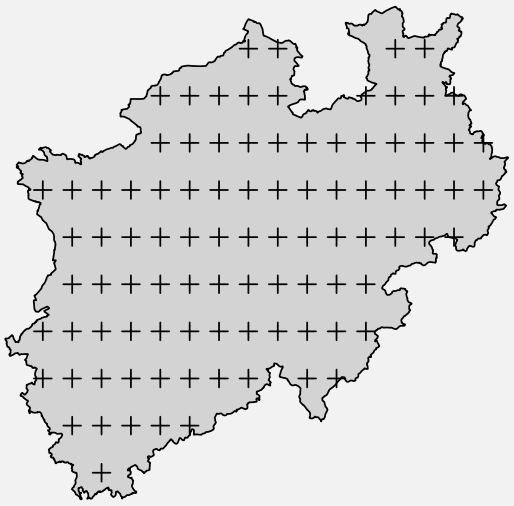
```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "regular")
> points(pts, pch = 3)
```



ifgi

Systematically aligned random sampling (2)

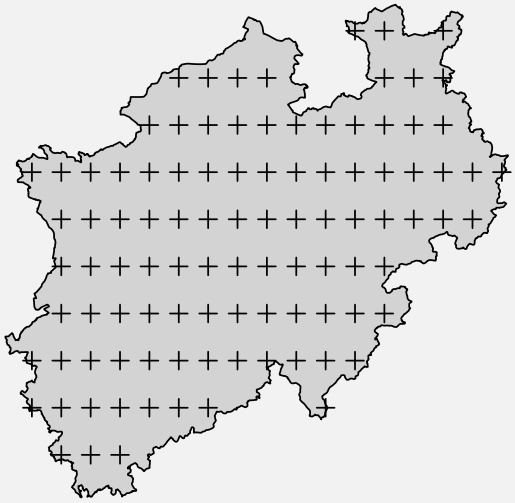
```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "regular")
> points(pts, pch = 3)
```



ifgi

Systematically aligned random sampling (3)

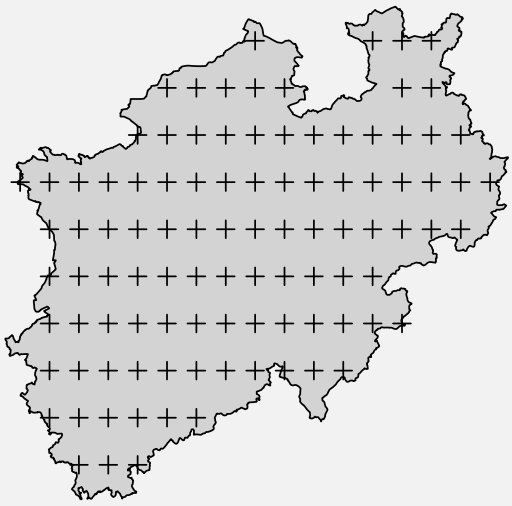
```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "regular")
> points(pts, pch = 3)
```



ifgi

Systematically aligned random sampling (4)

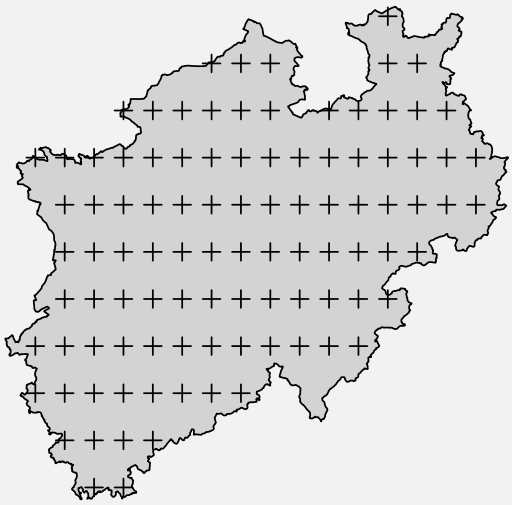
```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "regular")
> points(pts, pch = 3)
```



ifgi

Systematically aligned random sampling (5)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "regular")
> points(pts, pch = 3)
```



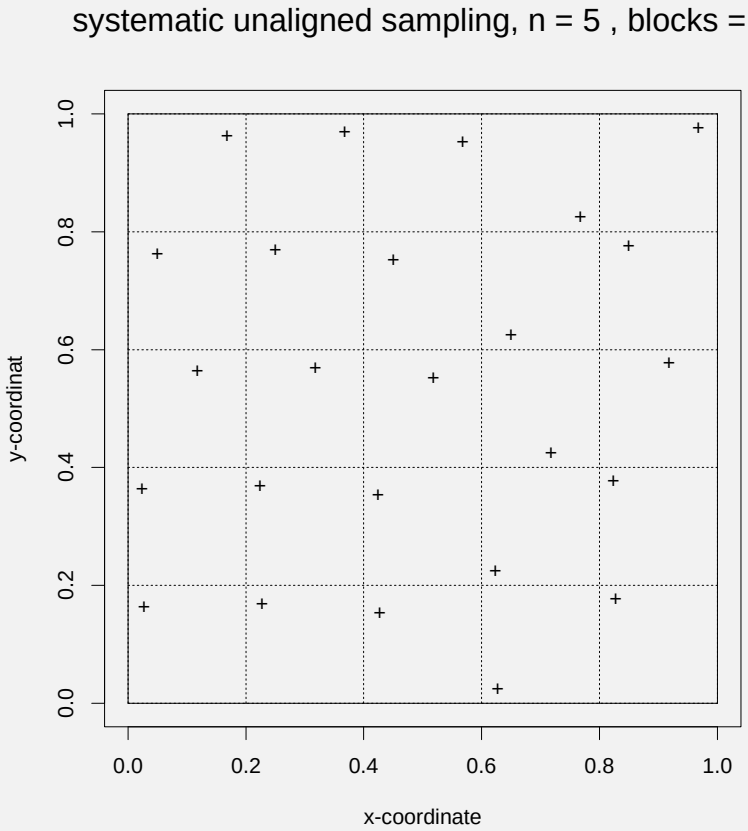
ifgi

Systematically aligned (regular) random sampling

Approach:

- Put a lattice over the area, with (approximately) n cells
- Randomly sample one point in the first cell
- Take the same point in all the other cells
- Number of random coordinates: 1

ifgi



ifgi

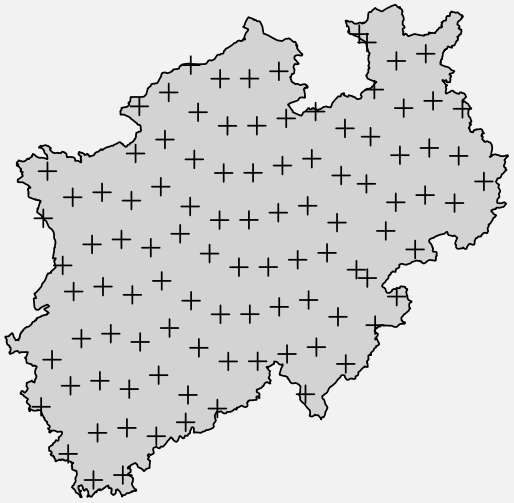
Systematically unaligned random sampling (1)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "nonaligned")
> points(pts, pch = 3)
```

ifgi

Systematically unaligned random sampling (2)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "nonaligned")
> points(pts, pch = 3)
```



ifgi

Systematically unaligned random sampling (3)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "nonaligned")
> points(pts, pch = 3)
```

ifgi

Systematically unaligned random sampling (4)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "nonaligned")
> points(pts, pch = 3)
```

ifgi

Systematically unaligned random sampling (5)

```
> plot(nrw, col = "lightgrey")
> pts = spsample(nrw, 100, "nonaligned")
> points(pts, pch = 3)
```

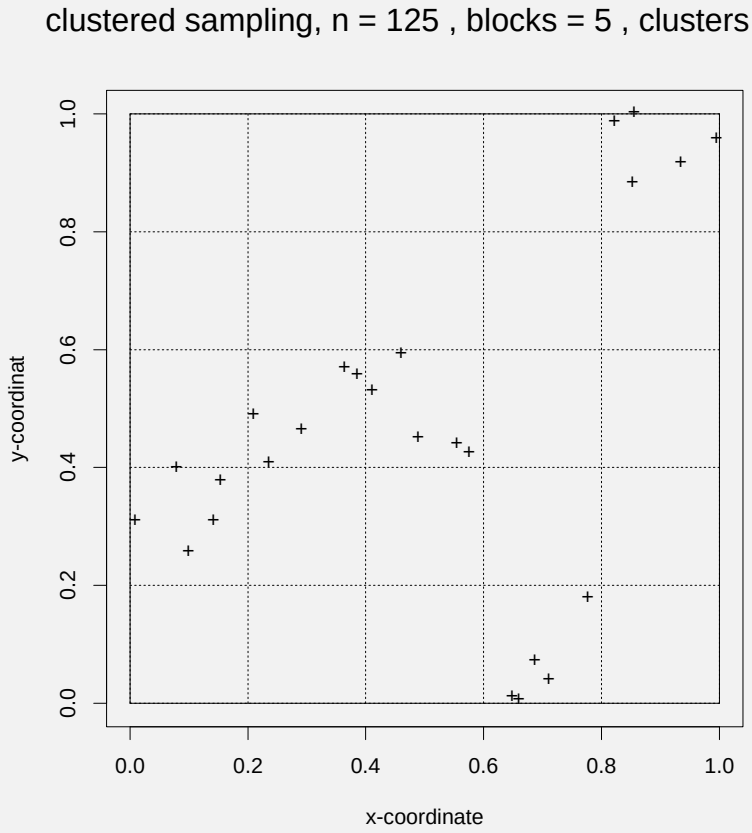
ifgi

Systematically unaligned random sampling

Approach:

- Put a lattice over the area, with (approximately) n cells
- Use a single random y coordinate for a single column
- Use a single random x coordinate for a single row
- Number of random coordinates $\approx \sqrt{\text{nrow} \times \text{ncol}}$

ifgi



ifgi

Clustered sampling

Possible approach:

- Sample n lattice cells at random,
- Within each lattice cell select m points at random
- results in a sample of size nm
- alternatives: we could apply regular sampling or stratfied sampling within the selected lattice cells
- advantage: travel time
- disadvantage: less efficient: suppose the lattice cells are very small, we may end up with effectively the same information as n single random points.

Alternative sampling methods:

- line sampling, where lines are placed at random (not e.g. existing, such as roads)
- hybrid methods

ifgi

Properties of the arithmetic mean

What about the mean value?

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{n}(X_1 + X_2 + ... + X_n)$$

$$\mathrm{E}\bar{X} = \sum_{i=1}^{n} \mathrm{E}(X_i) = \frac{1}{n}(\mathrm{E}(X_1) + \mathrm{E}(X_2) + ... + \mathrm{E}(X_n))$$

$$= \frac{1}{n}[\mu + \mu + ... + \mu] = \frac{1}{n}[n\mu] = \mu$$

What about it's variability? If all observations are independent, then

$$\mathrm{Var}(\sum_{i=1}^{n} X_i) = \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}(X_i)$$

$$\mathrm{Var}(\bar{X}) = \mathrm{Var}(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}(X_i) = \sigma^2/n$$

with $\sigma^2 = \mathrm{Var}(X)$

ifgi

Properties of the arithmetic mean, 2

Remember: standard error of $\bar{X}$, for independent observations is

$$\mathrm{SE} = \frac{\sigma}{\sqrt{n}}$$

- if $n = \infty$, $\mathrm{SE} = 0$ meaning that $\bar{X} = \mu$.
- in other cases, the variability in the difference, $\bar{X} - \mu$ has standard deviation SE
- **ONLY** if $n = 1$, $\mathrm{SE} = \sigma$; this is the case where we take a sample of size 1, and estimate the mean μ by this single value.

ifgi

Confidence interval for the mean

If the difference $\bar{X} - \mu$ follows a normal distribution, we can e.g. state that

$$Pr(\bar{X} - 1.96\text{SE} < \mu < \bar{X} + 1.96\text{SE}) = 0.95$$

and form a 95% confidence interval for μ by

$$[\bar{X} - 1.96\text{SE}, \bar{X} + 1.96\text{SE}]$$

Note that in the above $Pr()$ statement, randomness is associated with $\bar{X}$, as this fluctuates from one sample to another, and not with μ.
However, we still don't know σ!