# ASTRO STATS

## Error Estimation:
## Covariance matrix estimation techniques

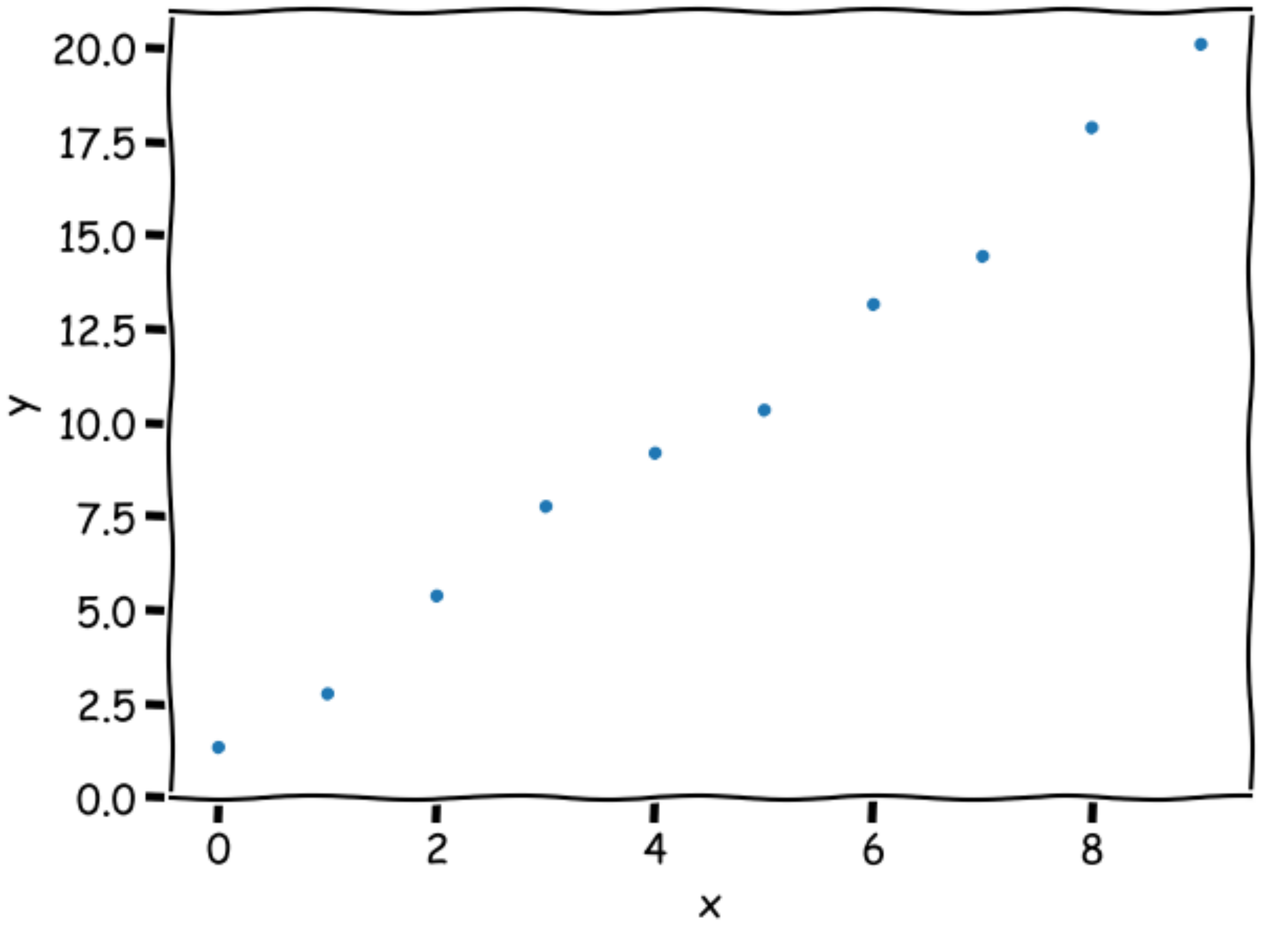
Week of Feb 5th 2024

# Covariance estimation

**LAST WEEK** you will have heard about what a covariance matrix is and some examples of how to generate a covariance matrix from first principles:

- Analytic covariance
- Sample Variance
- Cosmic Variance?
- Shot noise
- Covariance from simulations

# Covariance estimation



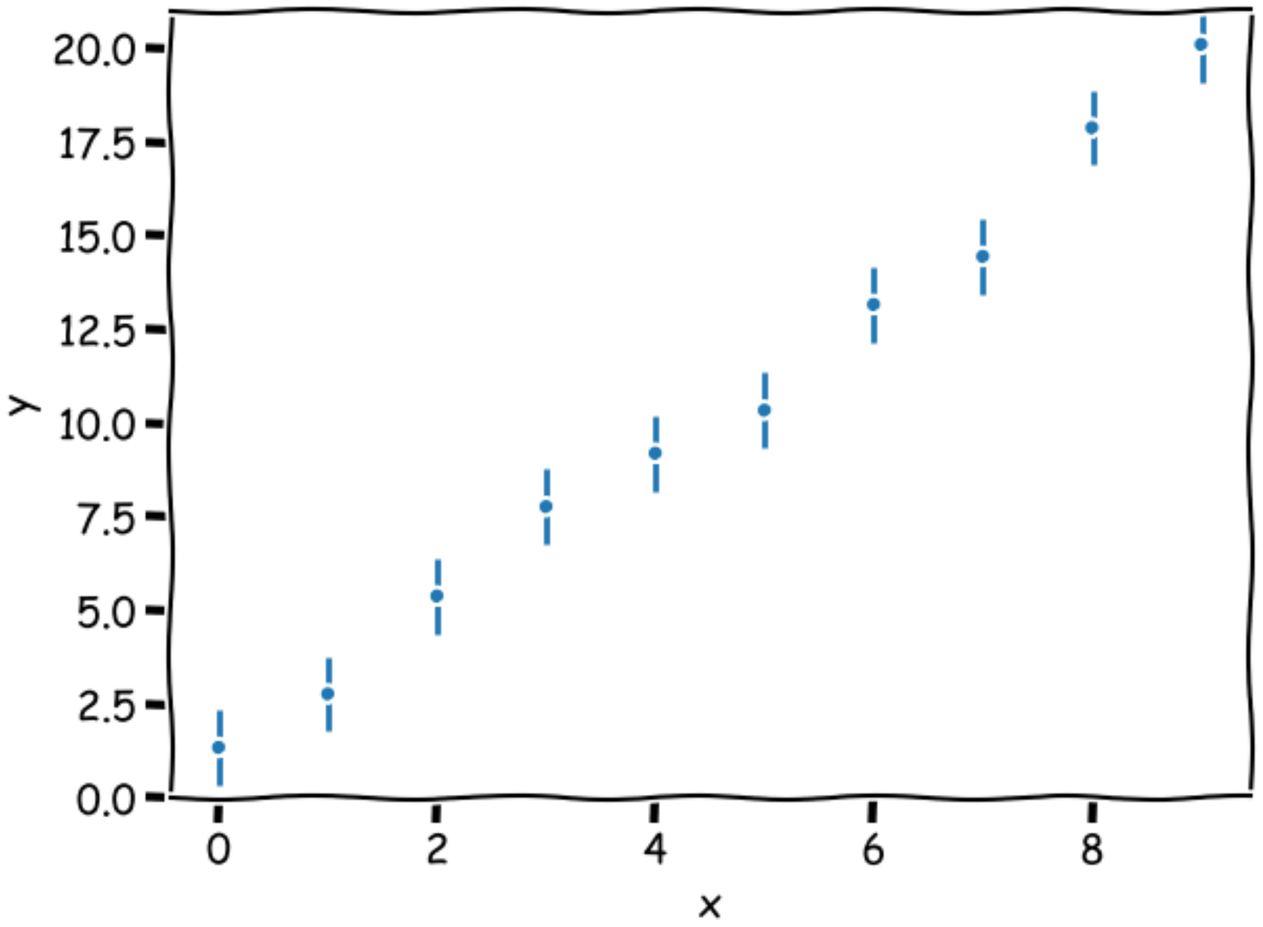

**Where are the error bars?**

In this module we assume we have some data vector $y_i$, which is constructed from a large dataset D

The points may be correlated

How do we add error bars (and covariance)?

# Covariance estimation



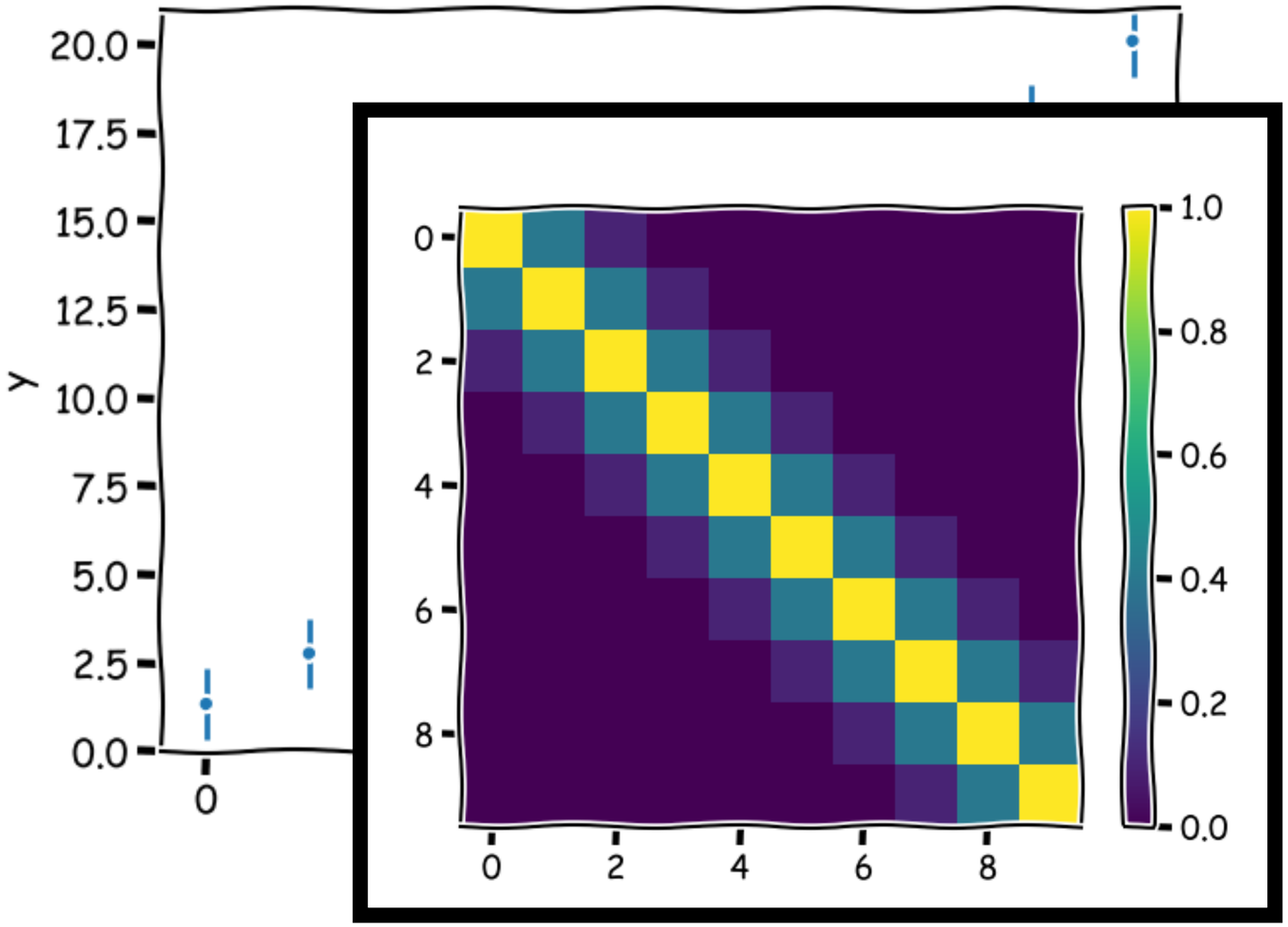

**Where are the error bars?**

In this module we assume we have some data vector $y_i$, which is constructed from a large dataset D

The points may be correlated

How do we add error bars (and covariance)?

# Covariance estimation




**Where are the error bars?**

In this module we assume we have some data vector $y_i$, which is constructed from a large dataset D

The points may be correlated

How do we add error bars (and covariance)?

# Extracting errors from the data

Ideally we have access to many (simulated) realizations of our data

We can use the definition of covariance to get the (sample) covariance matrix

$$C(X_i, X_j) = \frac{1}{N} \sum_{k=1}^{N} (X_i^k - \bar{X}_i)(X_j^k - \bar{X}_j)$$

Or maybe we have some analytic model for the covariance

However.....

# Extracting errors from the data

Often we don't have a model for the errors *a priori*

In these cases we can derive the errors from the data itself

**First, set up the notation:**

Say we have measured some data vector, $\vec{y} \equiv f(D)$ with dimensions $(N_y,\ 1)$

D is some large data set $\overrightarrow{D} = (D_1, D_2,\ \ldots,\ D_{N_D})$ with dimensions $(N_D,\ k)$

And $N_D >> N_y$

$\hat{f}(D)$ can be some arbitrarily complicated estimator for $\vec{y}$

(I'm going to drop the vector notation from now on, remember everything has many elements)

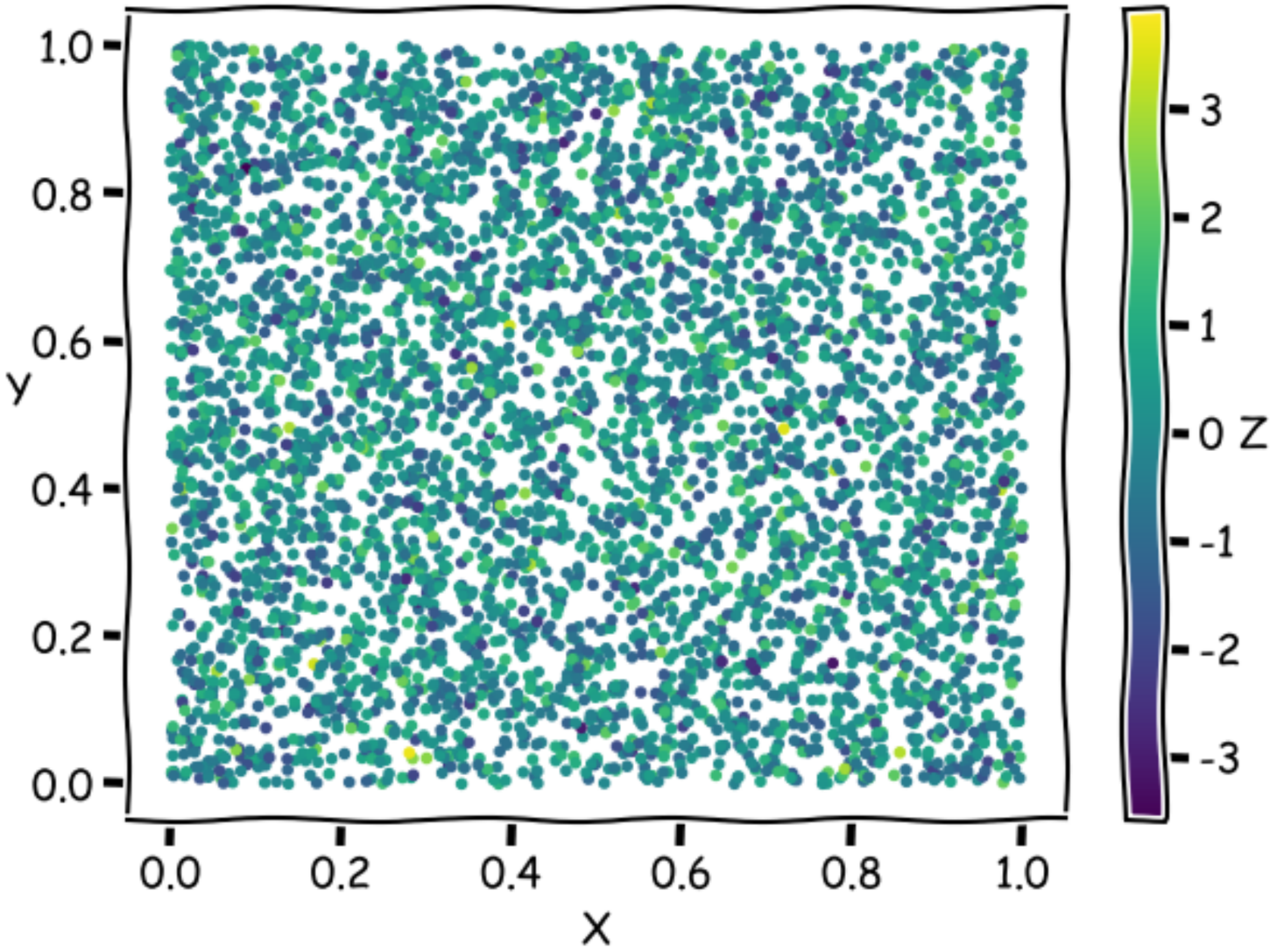
# Extracting errors from the data

IDEA: Chop the data into sub-samples, and derive the variance of the estimator from the variance of these sub-samples

These are called *re-sampling* methods

We will look at two types of re-sampling, ***Jackknife*** and ***Bootstrap***

# Re-sampling methods



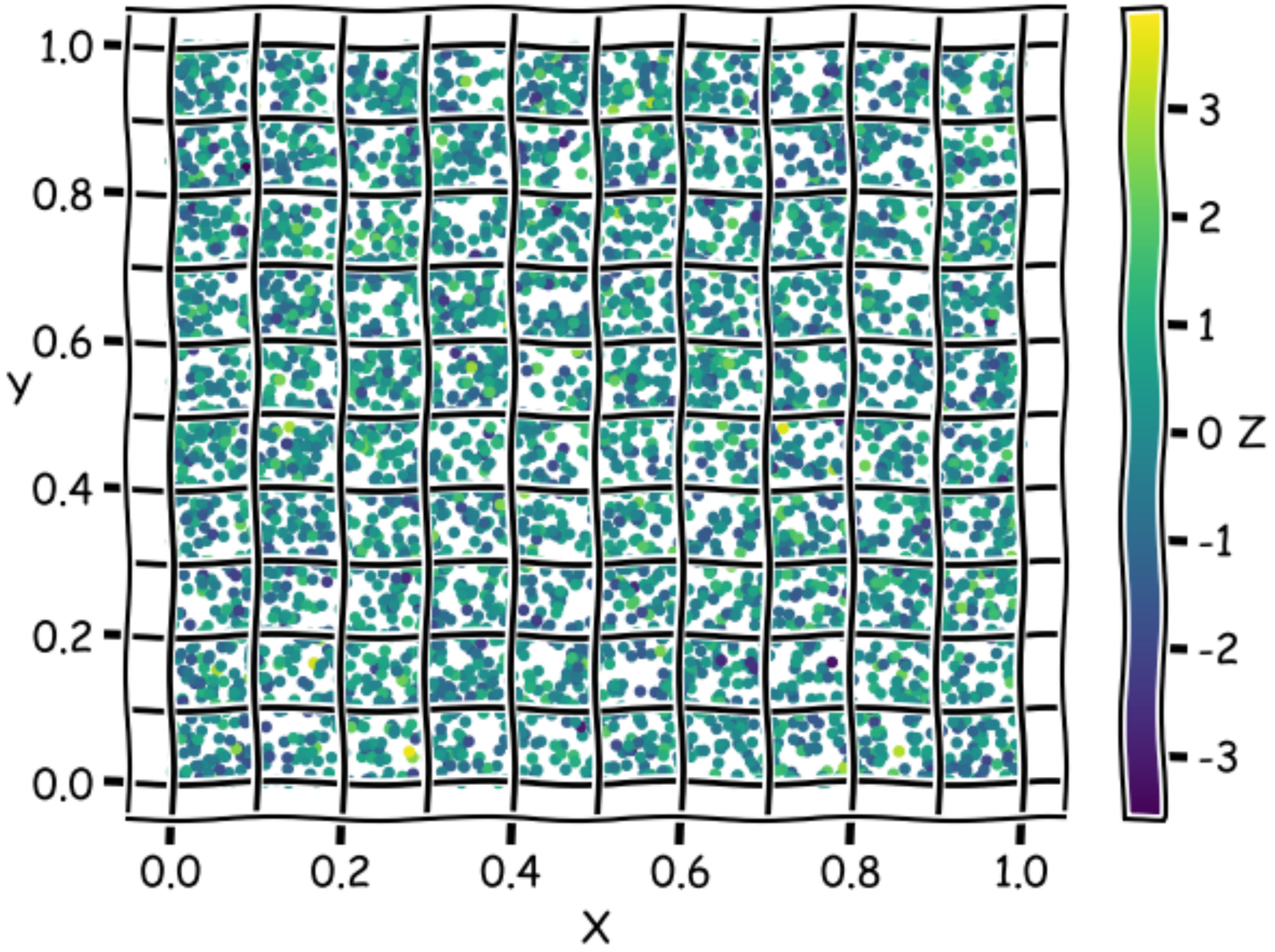

$D_i = (x_i, y_i, Z_i)$

Let's say we are scientifically interested in some estimator that uses Z as an input (e.g. variance of Z)

$Z$ is independent of $(x, y)$

What error do we put on our estimates?

# Re-sampling methods



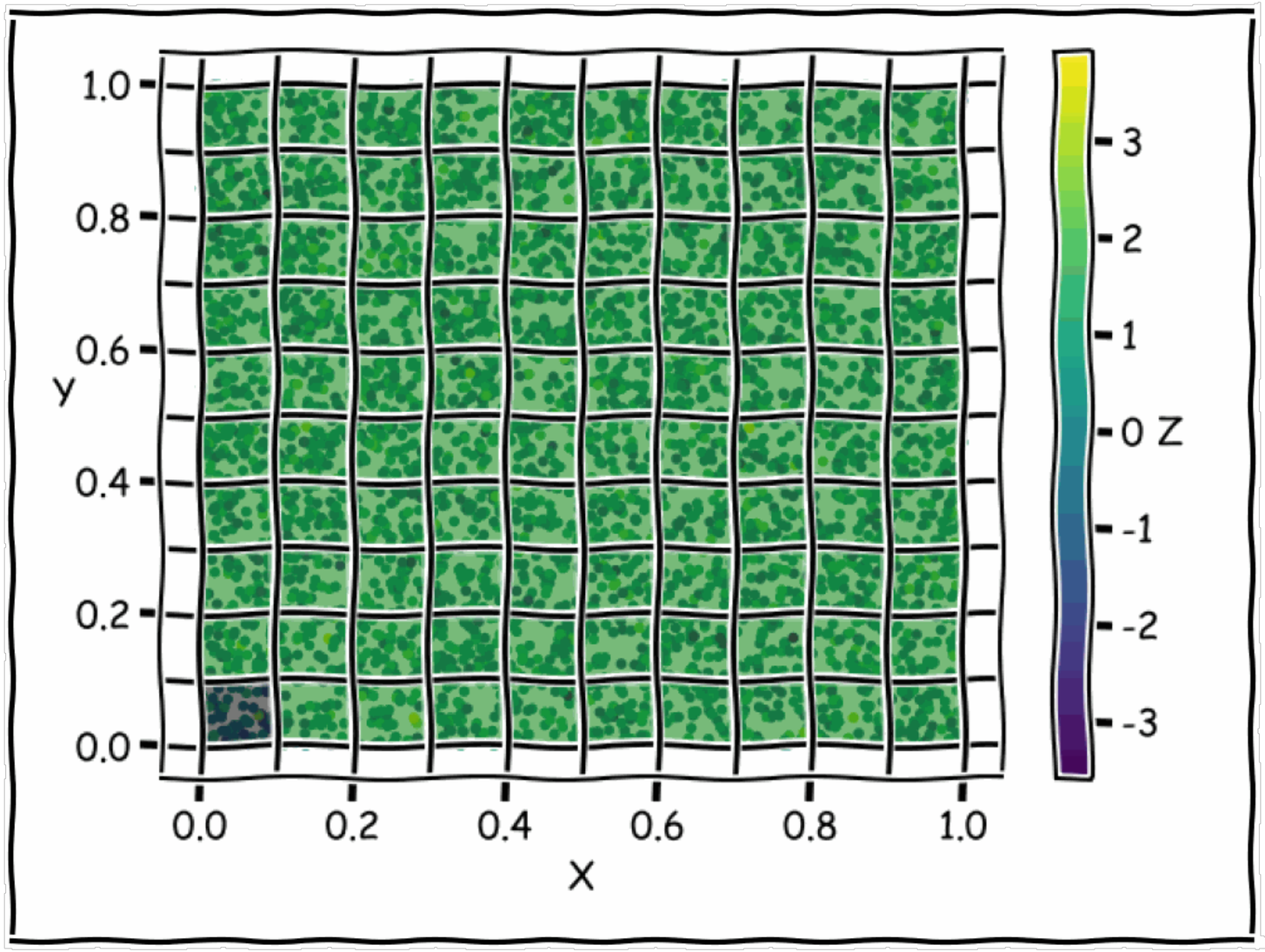

$D_i = (x_i, y_i, Z_i)$

Let's say we are scientifically interested in some estimator that uses Z as an input (e.g. variance of Z)

$Z$ is independent of $(x, y)$

What error do we put on our estimates?

# Re-sampling methods




***Jackknife***

Split data into N sub-samples

Remove each sub-sample and compute the estimator on the rest

# Jackknife Equation

$\hat{f}(D)$ ***is our estimator***

$D = [D_0, D_1, \ldots, D_N]$ ***is our raw data made up of*** $N_{\mathrm{JK}}$ ***sub-volumes***

$D_{[i]} = [D_0, D_1, \ldots, D_N]$ ***with sub-volume*** $i$ ***omitted***

$\overline{\hat{f}(D_{[i]})} = \sum_{i}^{N_{\mathrm{JK}}} \hat{f}(D_{[i]})$ *is the mean of the jackknife sub-volumes*

# Jackknife Equation

$\hat{f}(D)$ ***is our estimator***

$D = [D_0, D_1, \ldots, D_N]$ ***is our raw data made up of*** $N_{\mathrm{JK}}$ ***sub-volumes***

$D_{[i]} = [D_0, D_1, \ldots, D_N]$ ***with*** $D_i$ ***omitted***

$\overline{\hat{f}(D_{[i]})} = \sum_{i}^{N_{\mathrm{JK}}} \hat{f}(D_{[i]})$ *the mean of the jackknife sub-volumes*

## Jackknife Covariance:

$$C_{\mathrm{JK}}(\hat{f}_i(D), \hat{f}_j(D)) = \frac{N_{\mathrm{JK}} - 1}{N_{\mathrm{JK}}} \sum_{k=1}^{N_{\mathrm{JK}}} [\hat{f}_i(D_{[k]}) - \overline{\hat{f}_i(D_{[i']})}]\ [\hat{f}_j(D_{[k]}) - \overline{\hat{f}_j(D_{[i']})}]$$

# Jackknife Equation

$\hat{f}(D)$ ***is our estimator***

$D = [D_0, D_1, \ldots, D_N]$ ***is our raw data made up of*** $N_{\mathrm{JK}}$ ***sub-volumes***

$D_{[i]} = [D_0, D_1, \ldots, D_N]$ ***with*** $D_i$ ***omitted***

$\overline{\hat{f}(D_{[i]})} = \sum_i^{N_{\mathrm{JK}}} \hat{f}(D_{[i]})$ *the mean of the jackknife sub-volumes*

$$C_{\mathrm{JK}}(\hat{f}_i(D), \hat{f}_j(D)) = \frac{N_{\mathrm{JK}} - 1}{N_{\mathrm{JK}}} \sum_{k=1}^{N_{\mathrm{JK}}} [\hat{f}_i(D_{[k]}) - \overline{\hat{f}_i(D_{[i']})}]\ [\hat{f}_j(D_{[k]}) - \overline{\hat{f}_j(D_{[i']})}]$$

This is (N-1)/N, because jackknife re-samples are not independent

Reminder the definition of sample covariance:

$$C(X_i, X_j) = \frac{1}{N_{\mathrm{samples}}} \sum_{k=1}^{N_{\mathrm{samples}}} (X_i^k - \bar{X}_i)(X_j^k - \bar{X}_j)$$

# Jackknife Equation

$\hat{f}(D)$ ***is our estimator***

$D = [D_0, D_1, \ldots, D_N]$ ***is our raw data made up of*** $N_{\rm JK}$ ***sub-volumes***

$D_{[i]} = [D_0, D_1, \ldots, D_N]$ ***with*** $D_i$ ***omitted***

$\overline{\hat{f}(D_{[i]})} = \sum_i^{N_{\rm JK}} \hat{f}(D_{[i]})$ *the mean of the jackknife sub-volumes*

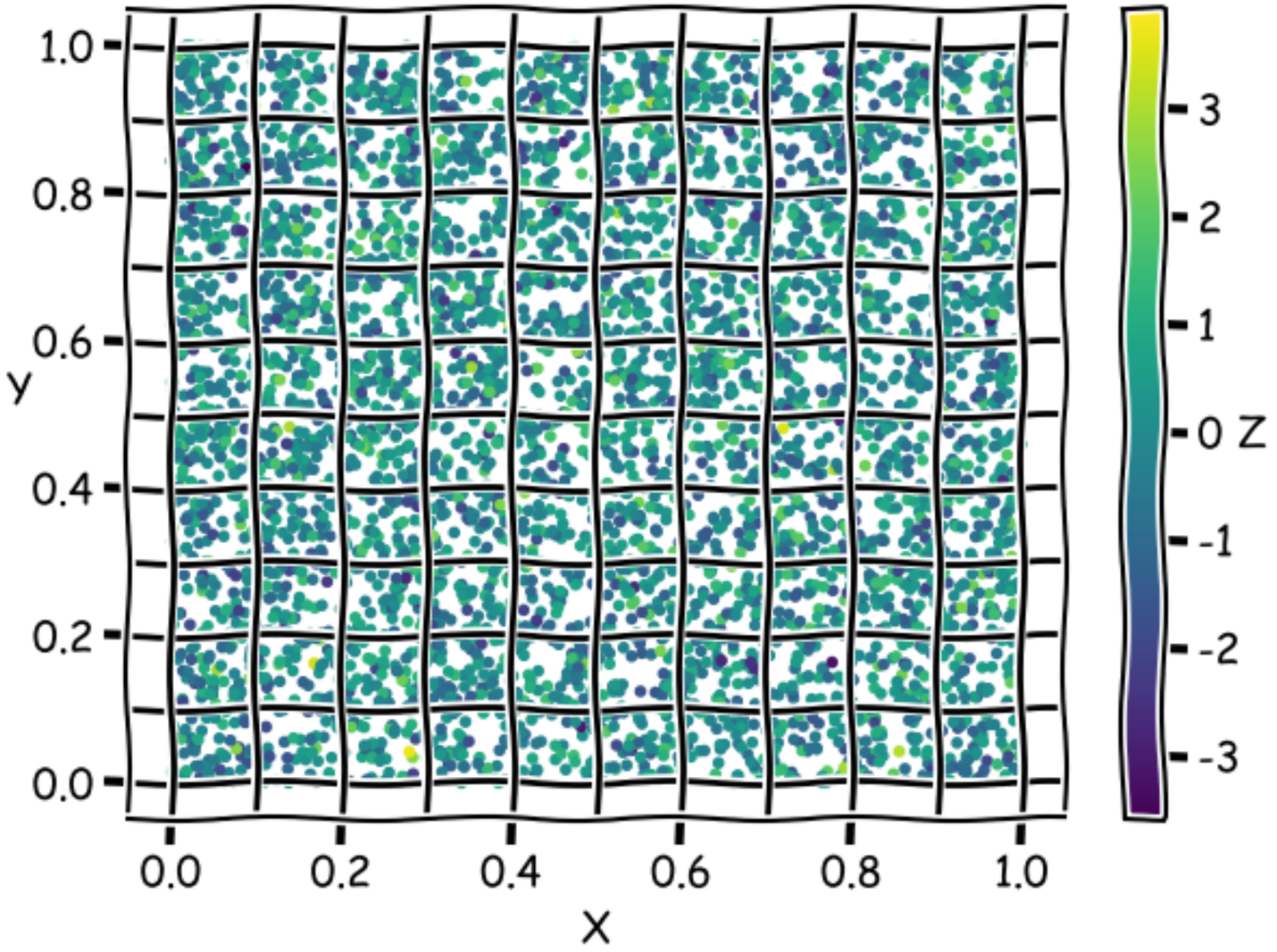
$$C_{\rm JK}(\hat{f}_i(D), \hat{f}_j(D)) = \frac{N_{\rm JK} - 1}{N_{\rm JK}} \sum_{k=1}^{N_{\rm JK}} [\hat{f}_i(D_{[k]}) - \overline{\hat{f}_i(D_{[i']})}]\ [\hat{f}_j(D_{[k]}) - \overline{\hat{f}_j(D_{[i']})}]$$

Note:

As $N_{\rm JK}$ -> large, the average $(X_i^k - \bar{X}_i)$ -> small

$\overline{\hat{f}_i(D_{[k]})}$ is the mean of the JK realizations, not the total signal

# Jackknife example




Our estimator is:

$$\hat{V} = \frac{\sum_i (Z_i - \langle Z \rangle)^2}{N - 1}$$

True variance is 1 (input)

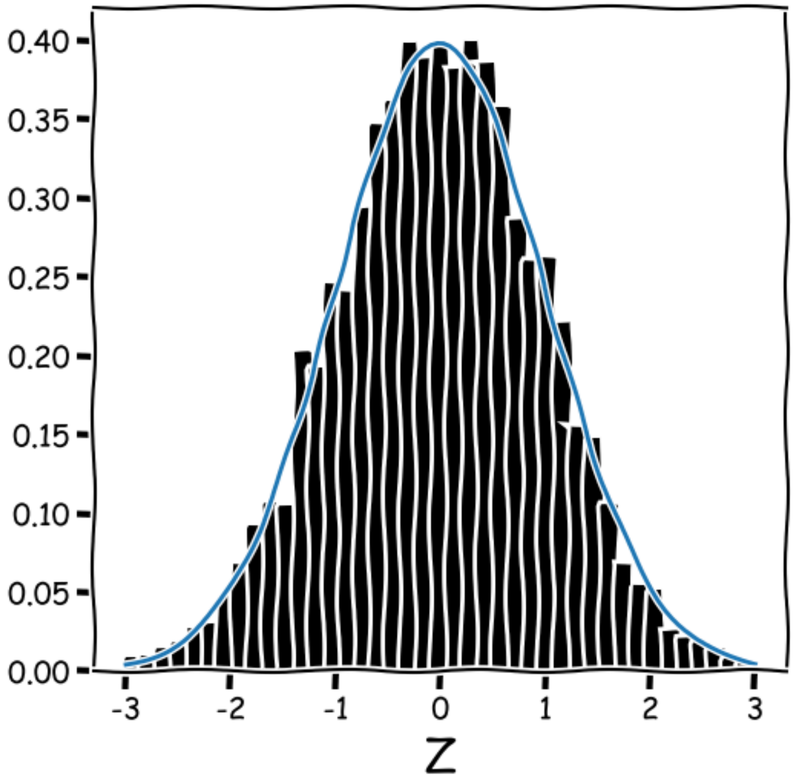
What is error in our estimate?

(The variance of the variance)

# Jackknife example

The Data

$N_D = 5000$

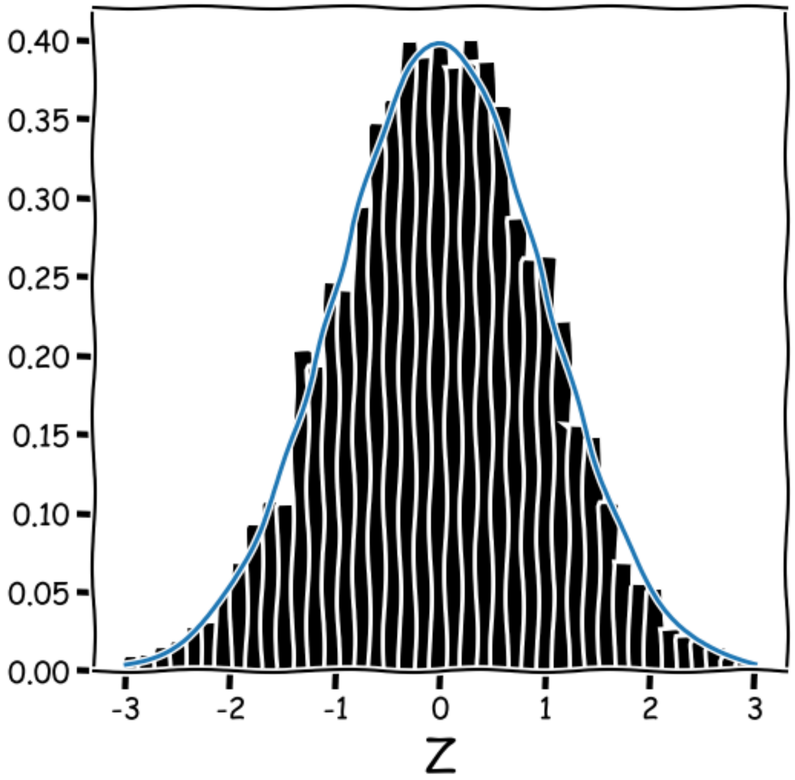



$\mathrm{Var}(Z) = 1.0$

# Jackknife example

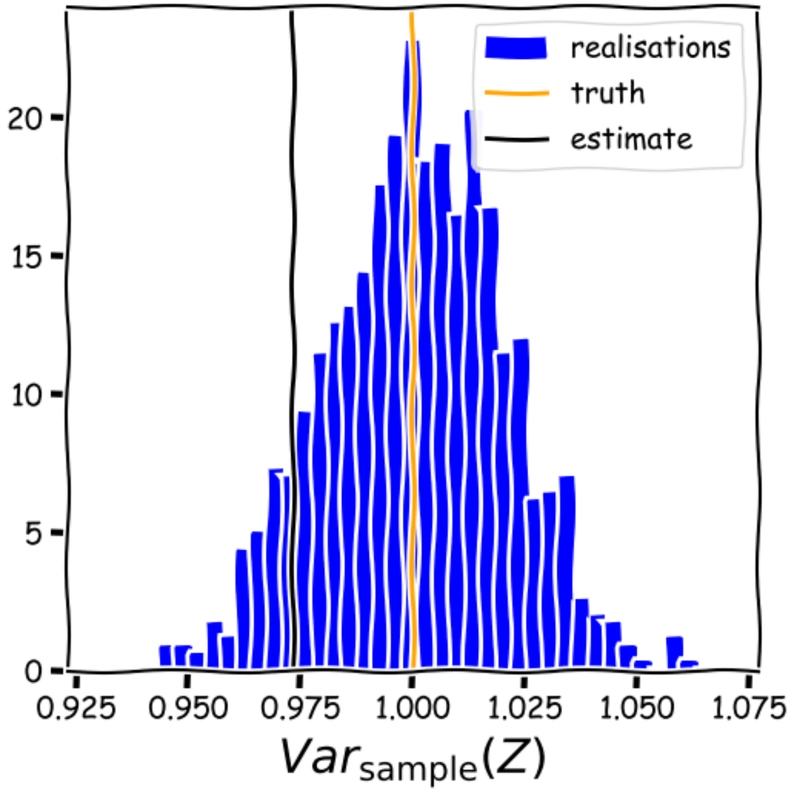
The Data

$N_D = 5000$




$\mathrm{Var}(Z) = 1.0$

Error from simulations/samples

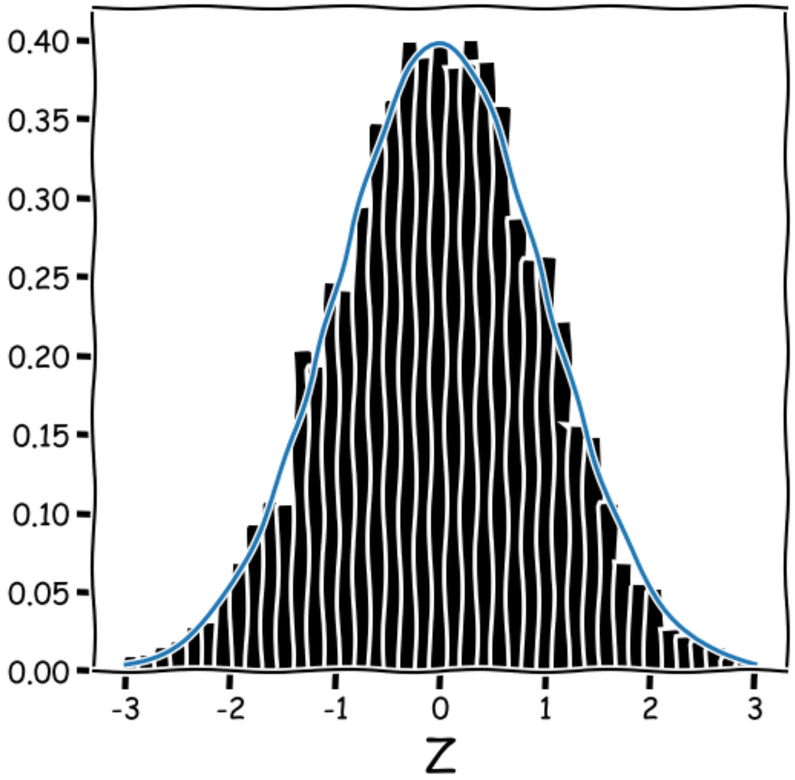



$\sigma^{\mathrm{sample}}_{\mathrm{Var}(Z)} = 0.019920$

# Jackknife example

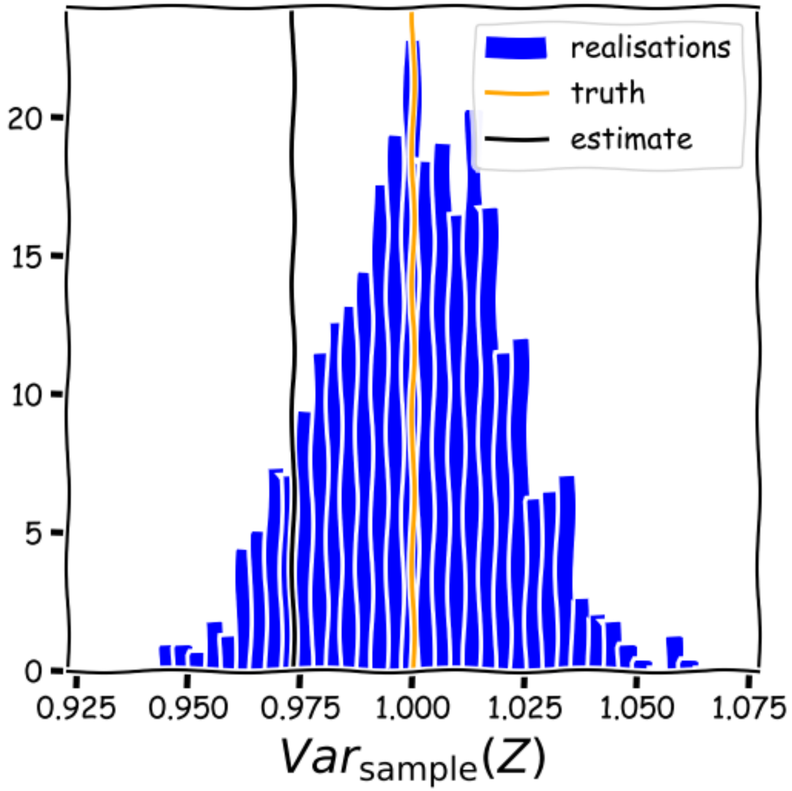
The Data

$N_D = 5000$



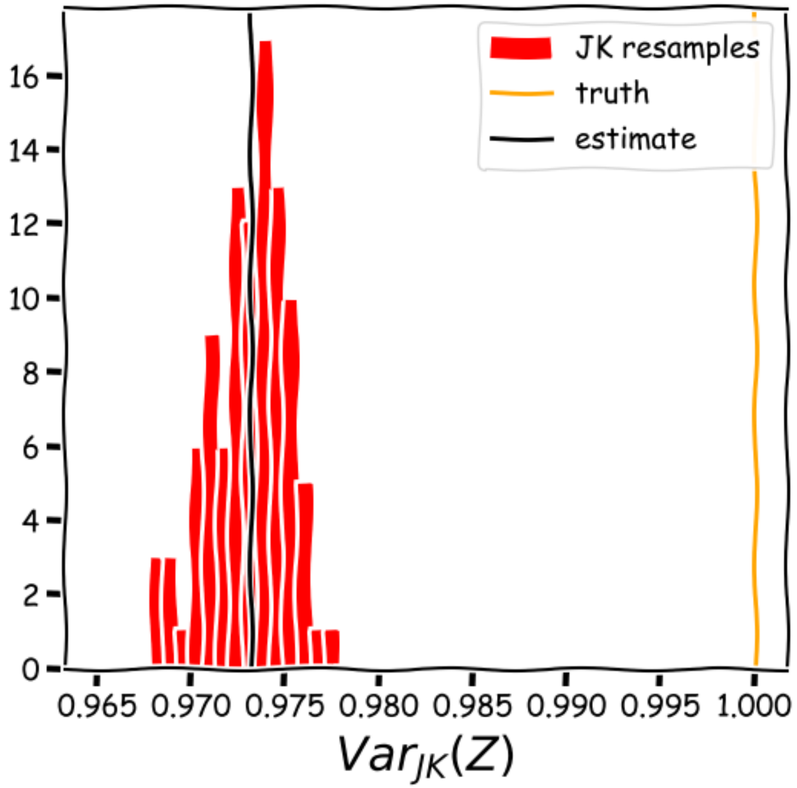

$\mathrm{Var}(Z) = 1.0$

Error from simulations/samples




$\sigma^{\mathrm{sample}}_{\mathrm{Var}(Z)} = 0.019920$

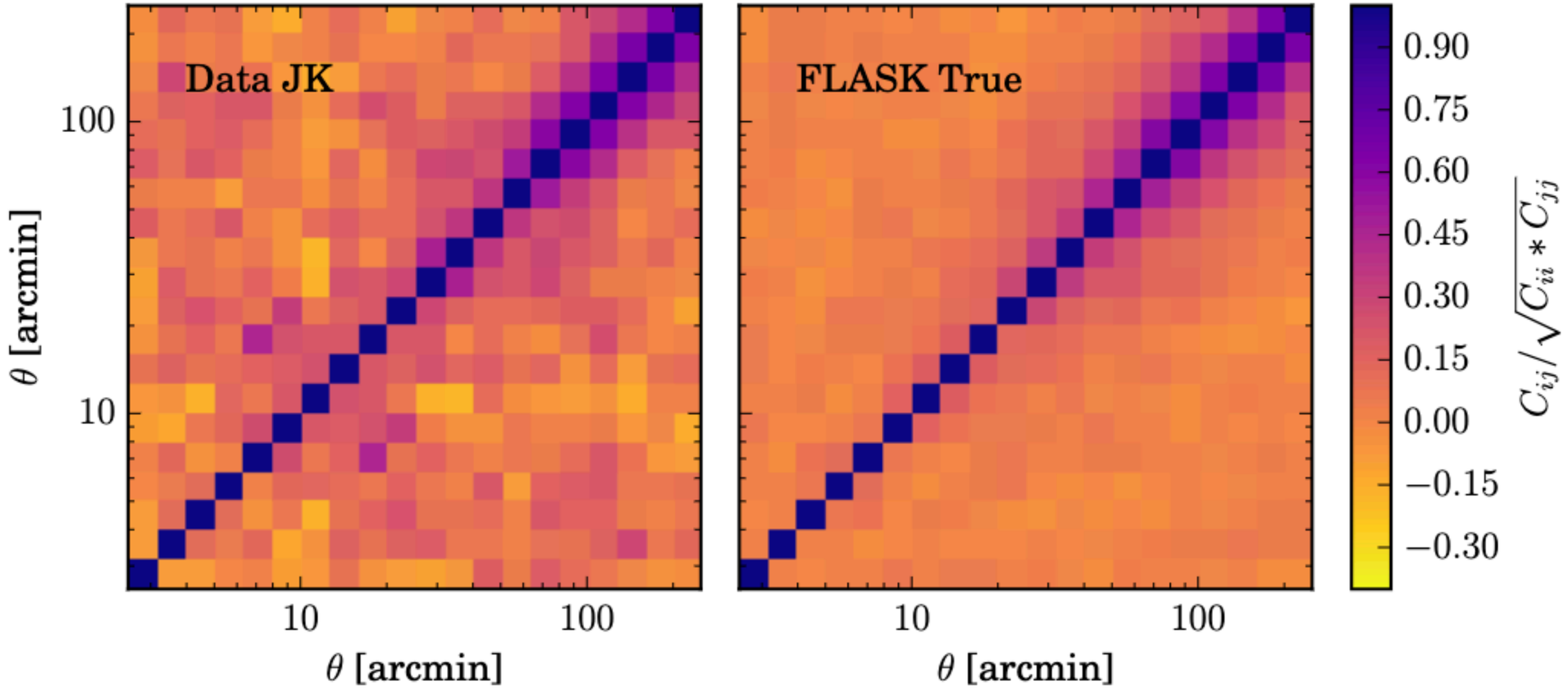
Error from Jackknife ($N_{\mathrm{JK}} = 100$)




$\sigma^{\mathrm{JK}}_{\mathrm{Var}(Z)} = 0.019927$

# Jackknife example

A more complex example:
Two-point correlation function between galaxy density and shear
Prat et al 2018



# Why does Jackknife work?

$\hat{f}(D)$ ***is our estimator***

$D = [D_0, D_1, \ldots, D_N]$ ***is our raw data made up of*** $N_{\mathrm{JK}}$ ***sub-volumes***

$D_{[i]} = [D_0, D_1, \ldots, D_N]$ ***with*** $D_i$ ***omitted***

$\overline{\hat{f}(D_{[i]})} = \sum_i^{N_{\mathrm{JK}}} \hat{f}(D_{[i]})$ *the mean of the jackknife sub-volumes*

*Formally, the jackknife covariance is the covariance of the "pseudo-values" of the data*

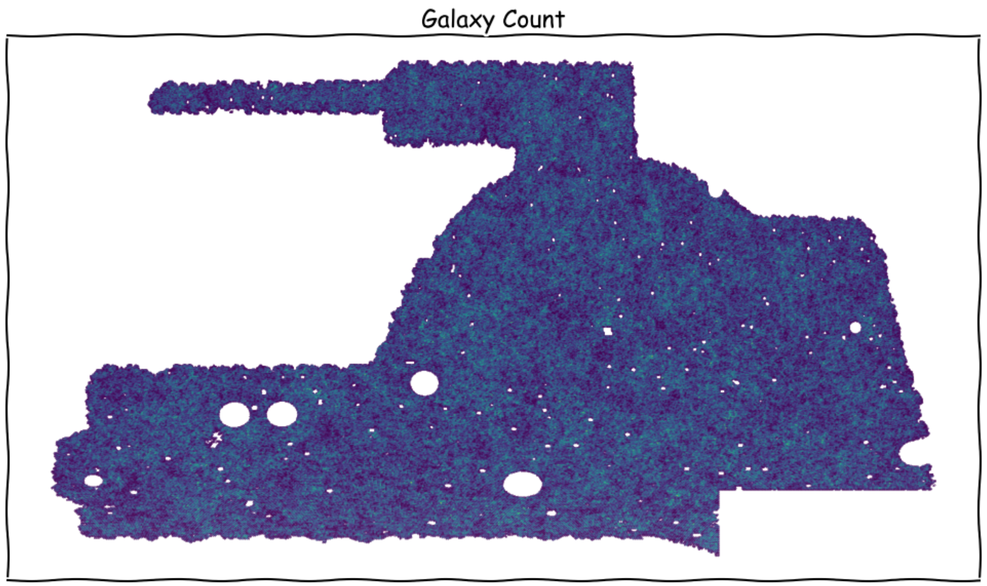
$$p_i = N_{\mathrm{JK}}\ \hat{f}(D)\ -\ (N_{\mathrm{JK}} - 1)\ \hat{f}(D_{[i]})$$

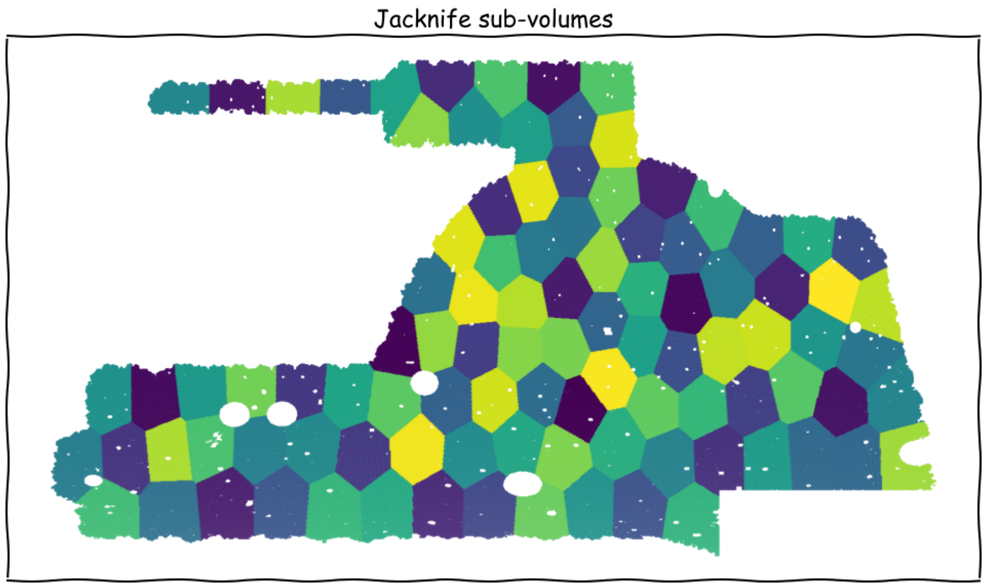
$$= \hat{f}(D)\ -\ (N_{\mathrm{JK}} - 1)\ (\hat{f}(D) - \hat{f}(D_{[i]}))$$

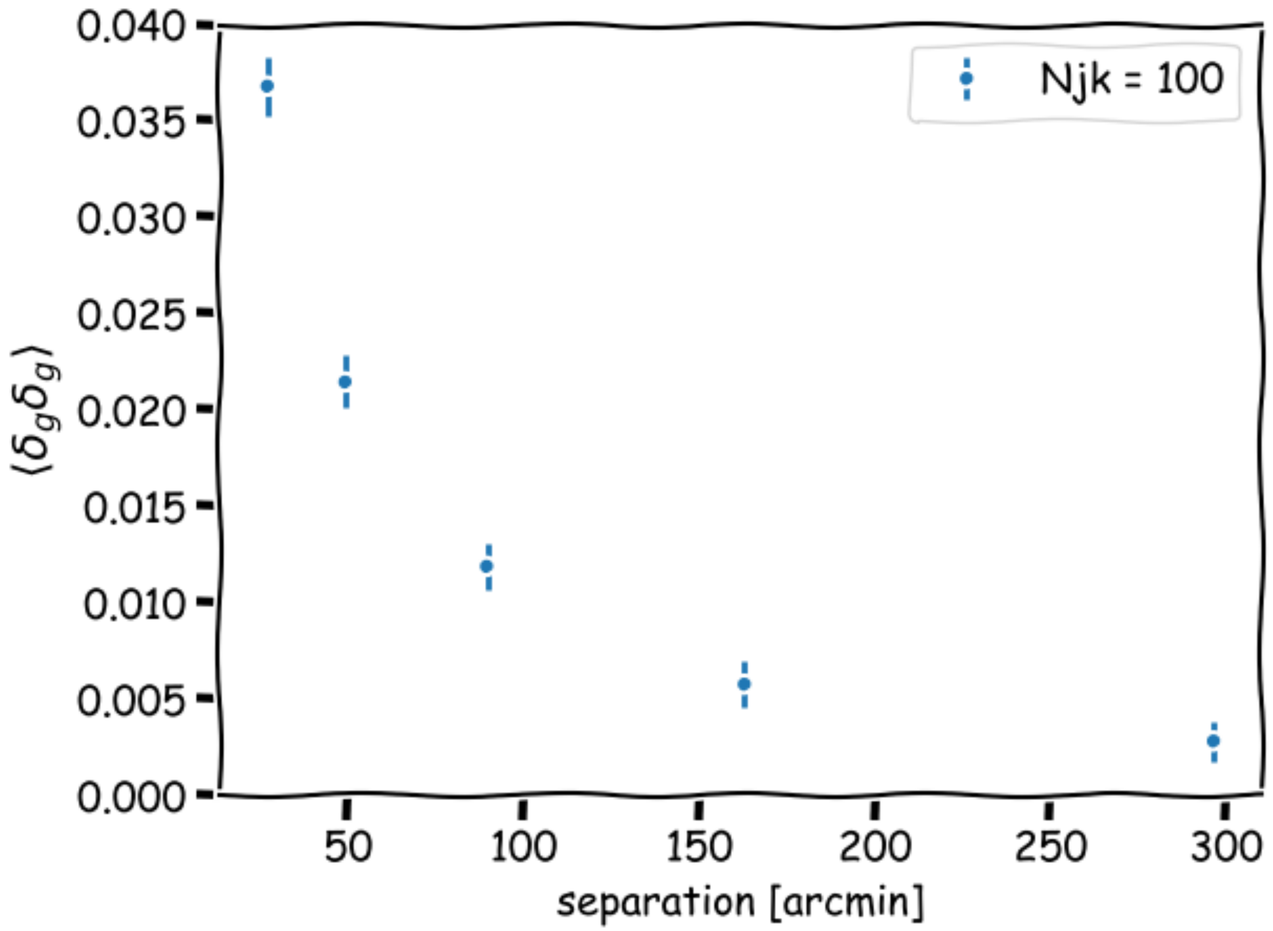
*But is a very good estimator of* $\mathrm{Cov}(\ \overline{\hat{f}(D_{[i]})}\ )$ *and therefore also* $\mathrm{Cov}(\ \hat{f}(D)\ )$

*(See refs for proof)*

# How can it go wrong?

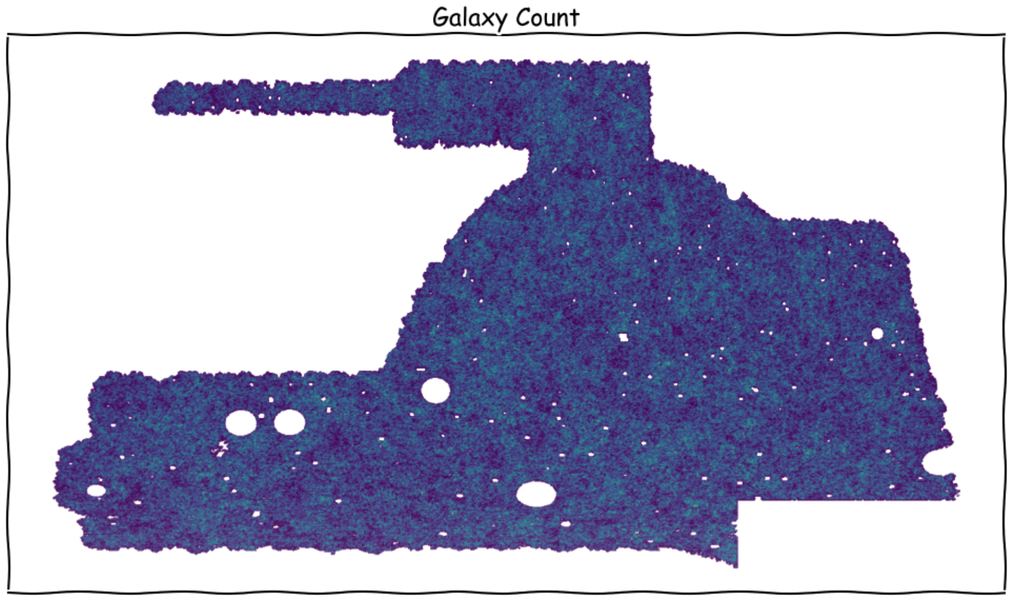


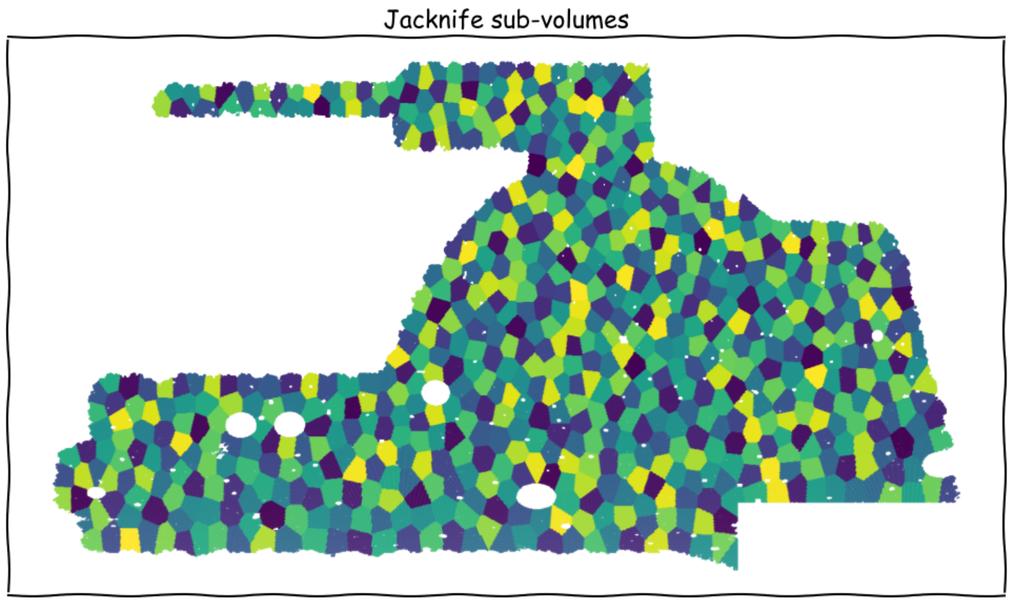



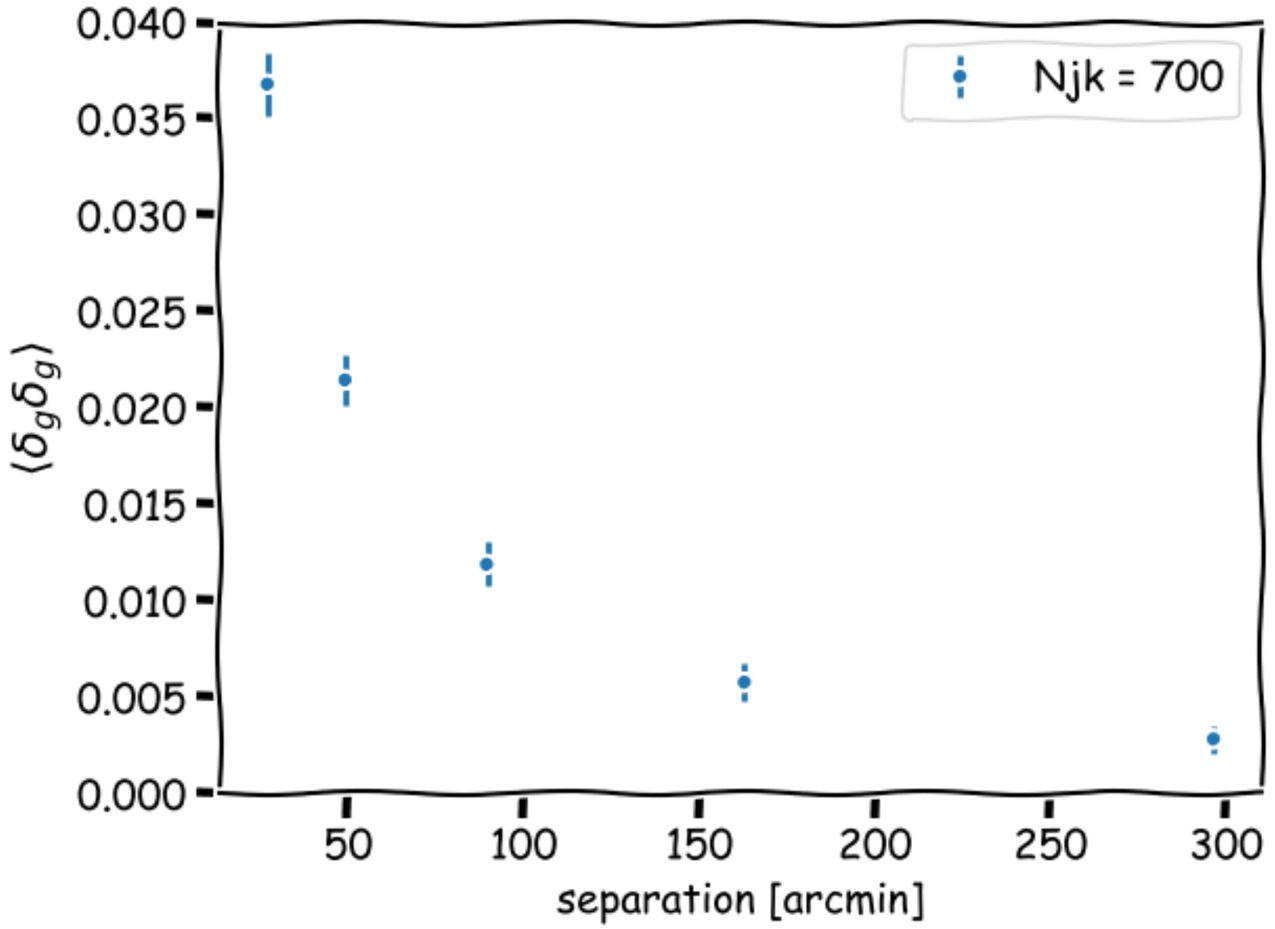




Patches from K-means : see ML module

# How can it go wrong?

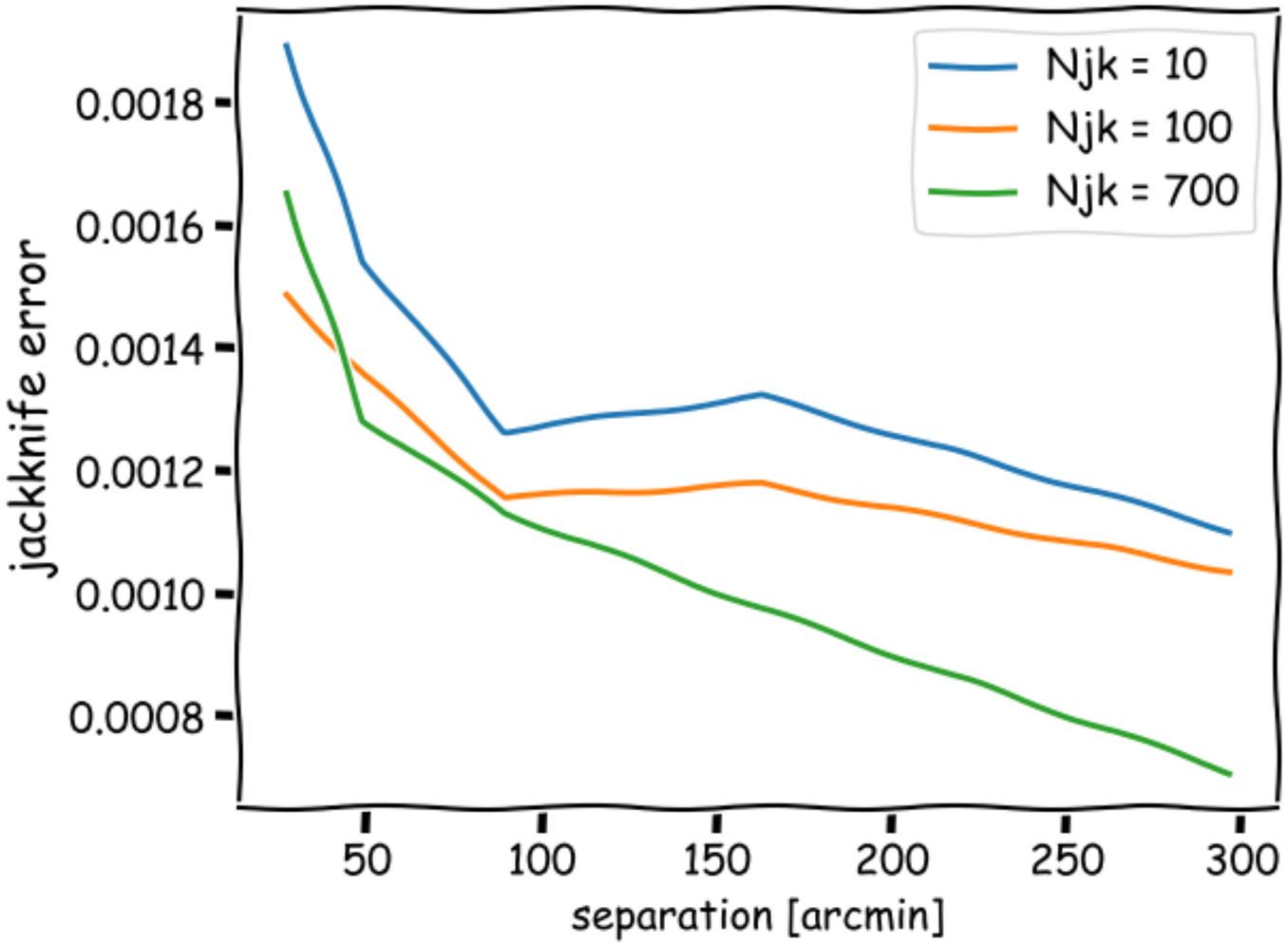



Patches from K-means : see ML module

# How can it go wrong?



# How can it go *right*?

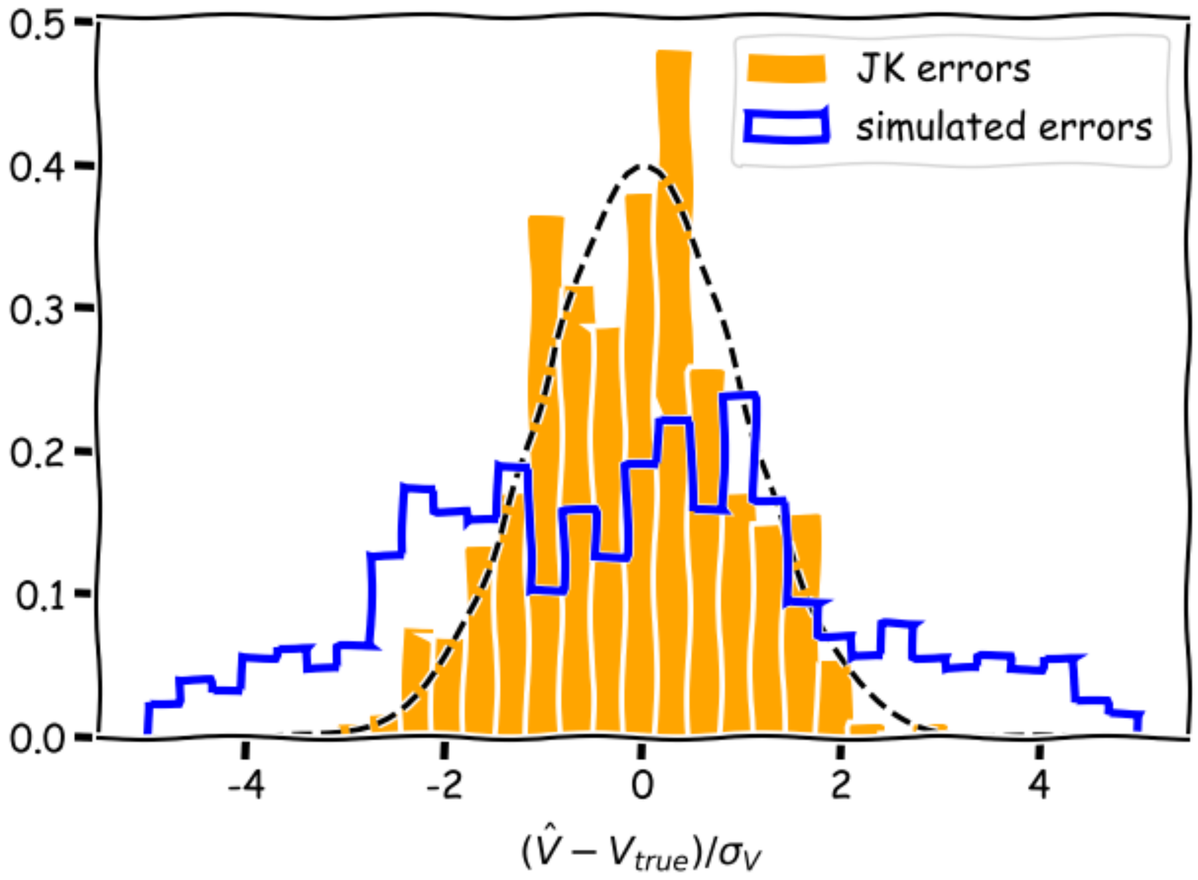
Say your data contains some unknown (multiplicative?) systematic

Any analytic or simulated covariance estimate will not account for this and you will underestimate the error

e.g. multiply $Z$ by some bias factor (in this example I picked 1.5)

Jackknife captures this additional uncertainty

(Also applies if your simulated errors guessed the wrong underlying statistic )




(Note “true” here includes the systematic)

# Using Jackknife to get Bias

*The mean of the jackknife pseudo-values*

$$\overline{p_i} = \overline{[\hat{f}(D) \; - \; (N_{\mathrm{JK}} - 1)\; (\hat{f}(D) - \hat{f}(D_{[i]})]}$$

*is generally a better estimator of $f$ than*

$$\hat{f}(D)$$

# Using Jackknife to get Bias

*The mean of the jackknife pseudo-values* $\overline{p_i} = \overline{[\hat{f}(D) \; - \; (N_{\mathrm{JK}} - 1)\; (\hat{f}(D) - \hat{f}(D_{[i]})]}$

*is generally a better estimator of* $f$ *than* $\hat{f}(D)$

***Expectation value of estimator, Taylor expansion***

$$E(\hat{f}(D)) = f + \frac{a_1}{N_{\mathrm{D}}} + \frac{a_2}{N_{\mathrm{D}}^2} + \ldots$$

A more thorough proof here

# Using Jackknife to get Bias

*The mean of the jackknife pseudo-values* $\overline{p_i} = \overline{[\hat{f}(D) \; - \; (N_{\mathrm{JK}} - 1)\; (\hat{f}(D) - \hat{f}(D_{[i]})]}$

*is generally a better estimator of* $f$ *than* $\hat{f}(D)$

***Expectation value of estimator, Taylor expansion***

$$E(\hat{f}(D)) = f + \frac{a_1}{N_{\mathrm{D}}} + \frac{a_2}{N_{\mathrm{D}}^2} + \dots$$

$$E(\hat{f}(D_{[i]})) = \overline{E(\hat{f}(D_{[i]}))} = f + \frac{a_1}{N_{\mathrm{D}} - 1} + \frac{a_2}{(N_{\mathrm{D}} - 1)^2} + \dots$$

A more thorough proof here

# Using Jackknife to get Bias

*The mean of the jackknife pseudo-values* $\overline{p_i} = \overline{[\hat{f}(D) \; - \; (N_{\mathrm{JK}} - 1)\; (\hat{f}(D) - \hat{f}(D_{[i]})]}$

*is generally a better estimator of* $f$ *than* $\hat{f}(D)$

***Expectation value of estimator, Taylor expansion***

$$E(\hat{f}(D)) = f + \frac{a_1}{N_{\mathrm{D}}} + \frac{a_2}{N_{\mathrm{D}}^2} + \dots$$

$$E(\hat{f}(D_{[i]})) = E(\overline{\hat{f}(D_{[i]})}) = f + \frac{a_1}{N_{\mathrm{D}} - 1} + \frac{a_2}{(N_{\mathrm{D}} - 1)^2} + \dots$$

$$\hat{\mathrm{Bias}}(\hat{f}(D_{[i]})) = \hat{f}(D_{[i]}) - \overline{p}_i$$

$$= (N_D - 1)(\overline{\hat{f}(D_{[i]})} - \hat{f}(D))$$

A more thorough proof here

# Using Jackknife to get Bias

*The mean of the jackknife pseudo-values* $\overline{p_i} = \overline{[\hat{f}(D) \; - \; (N_{\mathrm{JK}} - 1)\ (\hat{f}(D) - \hat{f}(D_{[i]})]}$

*is generally a better estimator of* $f$ *than* $\hat{f}(D)$

***e.g.***

***You can use JK to prove that Bessel's correction is an unbiased estimator of variance***

***i.e. this:***

$$\mathrm{Var}(X) = \frac{1}{n-1}\sum_i (X_i - \bar{X})^2$$

***Is less biased than this:***

$$\mathrm{Var}(X) = \frac{1}{n}\sum_i (X_i - \bar{X})^2$$

A more thorough proof here

# Jackknife - Things to Note

Number of samples should be selected carefully

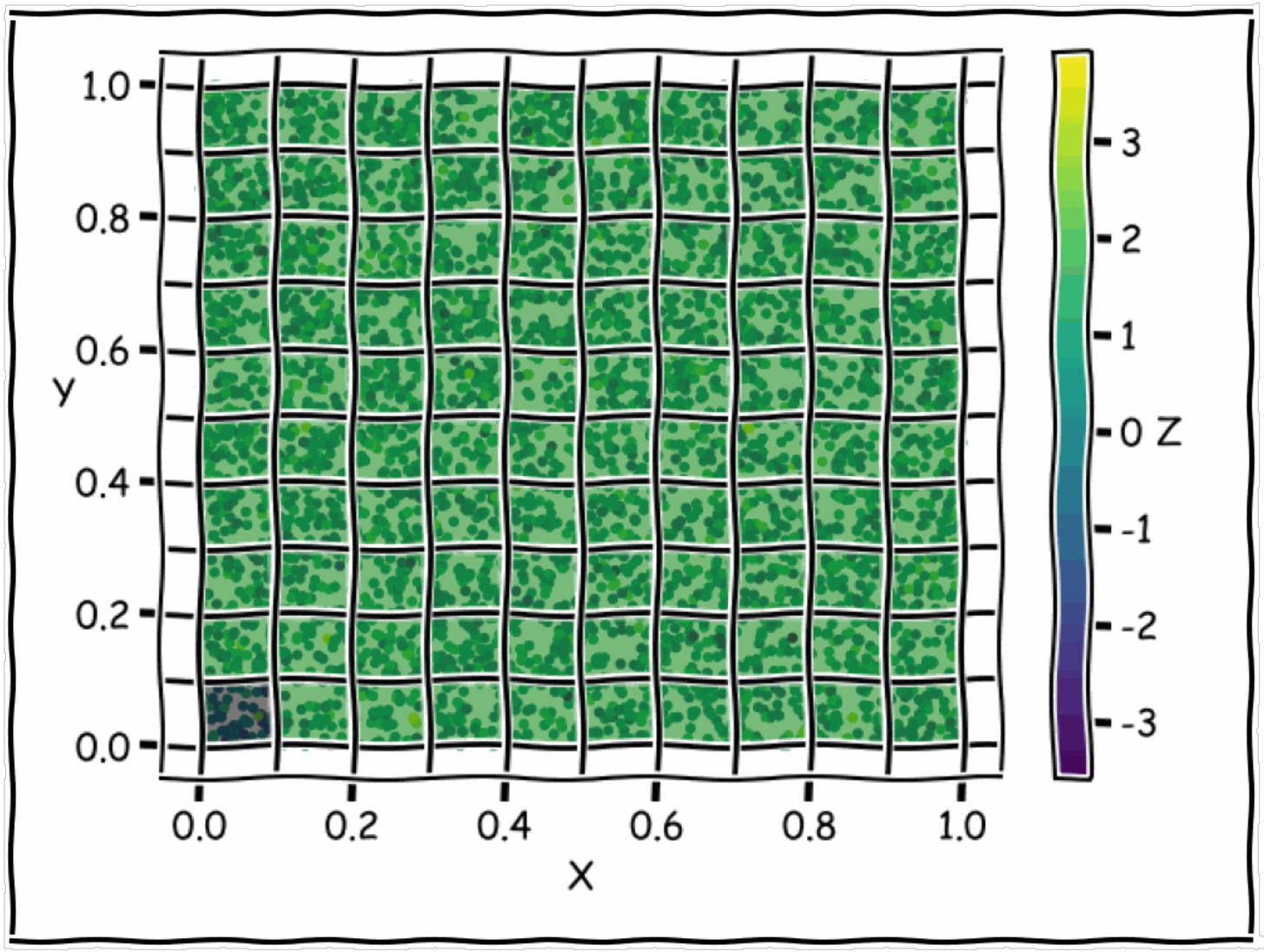
Size of the sub-volumes needs to be the same

Jackknife is generally slightly biased upwards (proven numerically)

Generally faster than alternative re-sampling (e.g. bootstrap, see later slides)

Doesn’t work very well for non smooth quantities, e.g. medians

# Thursday
# Jackknife - one slide recap



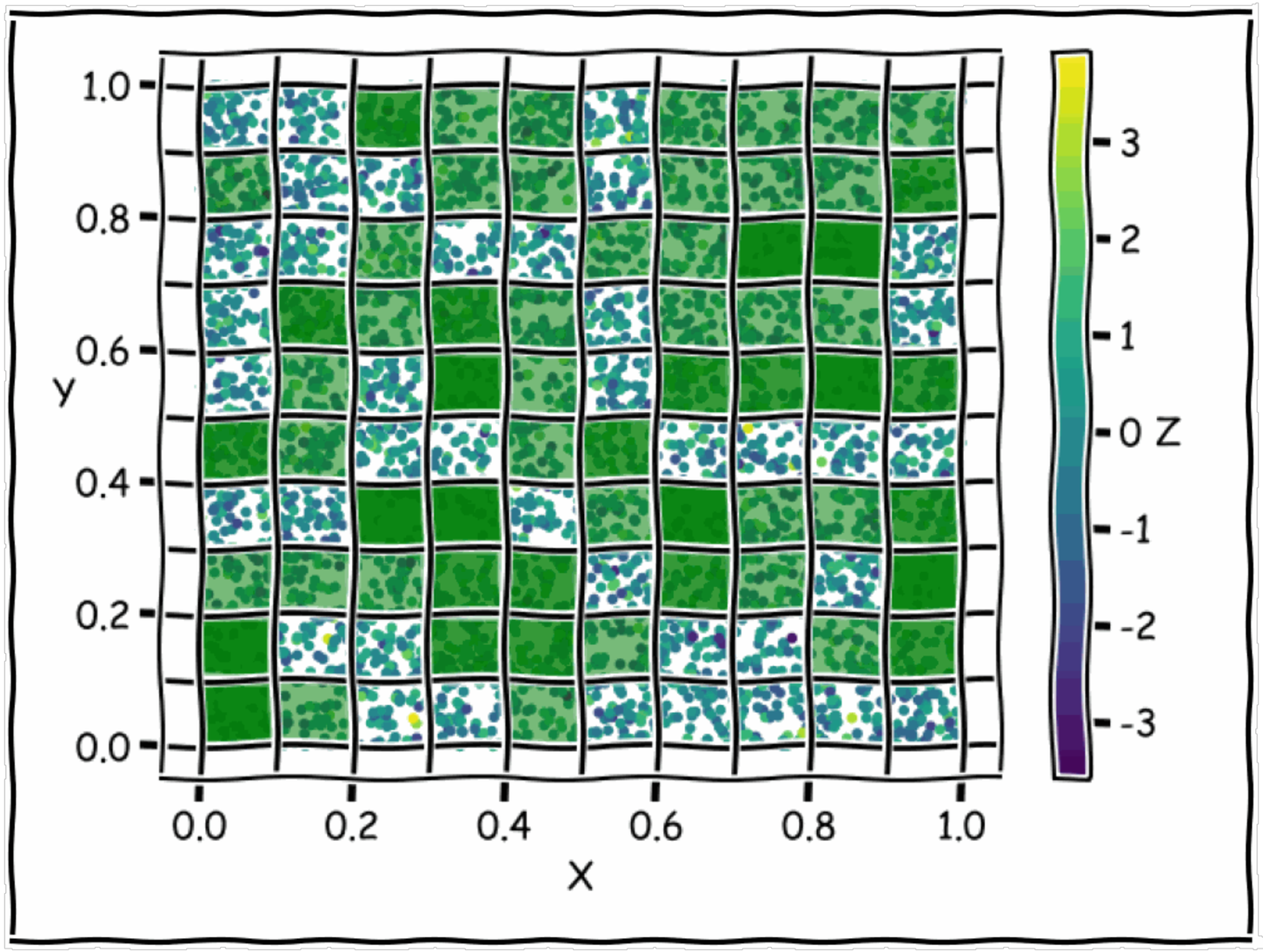

$$C_{\mathrm{JK}}(\hat{f}_i(D), \hat{f}_j(D)) =$$

$$\frac{N_{\mathrm{JK}} - 1}{N_{\mathrm{JK}}} \sum_{k=1}^{N_{\mathrm{JK}}} [\hat{f}_i(D_{[k]}) - \overline{\hat{f}_i(D_{[i']})}]\ [\hat{f}_j(D_{[k]}) - \overline{\hat{f}_j(D_{[i']})}]$$

One JK Realization

Mean of the JK realizations

# Re-sampling methods




***Bootstrap***

Split data into N sub-samples

Randomly select N subsamples *with replacement*

Compute estimator on each realisation

# Re-sampling methods

**Bootstrap equation (looks similar to the covariance definition)**

$$C_{\text{boot}}(\hat{f}_i(D), \hat{f}_j(D)) = \frac{1}{N_{\text{boot}} - 1} \sum_{k=1}^{N} [\hat{f}_i(D_{[k]}) - \overline{\hat{f}_i(D_{[i']})}]\ [\hat{f}_j(D_{[k]}) - \overline{\hat{f}_j(D_{[i']})}]$$

I have seen it claimed that this is N-1, not N

Because the samples are not independent, but I want proof

As $N_{\text{boot}}$ -> large, the average $(X_i^k - \bar{X}_i)$ -> converges to some value

# Re-sampling methods - bootstrap

Easy to implement as there are python packages that can do it for you!
One on astropy and one on scipy (update your script to get all the new features)

## bootstrap

**astropy.stats.bootstrap**(*data*, *bootnum=100*, *samples=None*, *bootfunc=None*) [source]

Performs bootstrap resampling on numpy arrays.

Bootstrap resampling is used to understand confidence intervals of sample estimates. This function returns versions of the dataset resampled with replacement ("case bootstrapping"). These can all be run through a function or statistic to produce a distribution of values which can then be used to find the confidence intervals.

## scipy.stats.bootstrap #

**scipy.stats.bootstrap**(*data*, *statistic*, ***, *n_resamples=9999*, *batch=None*, *vectorized=None*, *paired=False*, *axis=0*, *confidence_level=0.95*, *alternative='two-sided'*, *method='BCa'*, *bootstrap_result=None*, *random_state=None*) [source]

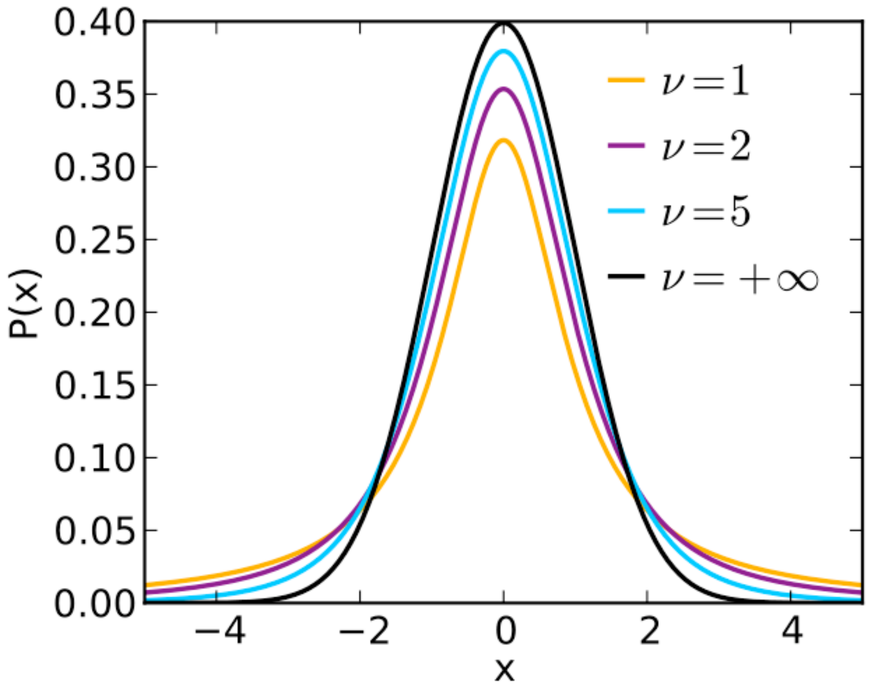
Compute a two-sided bootstrap confidence interval of a statistic.

When *method* is `'percentile'` and *alternative* is `'two-sided'`, a bootstrap confidence interval is computed according to the following procedure.

1. Resample the data: for each sample in *data* and for each of *n_resamples*, take a random sample of the original sample (with replacement) of the same size as the original sample.
2. Compute the bootstrap distribution of the statistic: for each set of resamples, compute the test statistic.
3. Determine the confidence interval: find the interval of the bootstrap distribution that is
    - symmetric about the median and
    - contains *confidence_level* of the resampled statistic values.

# Re-sampling methods - Bootstrap

***Bootstrap is great if you want know the probability distribution of your estimator  e.g. Is it Gaussian?***




Example student's-t distribution

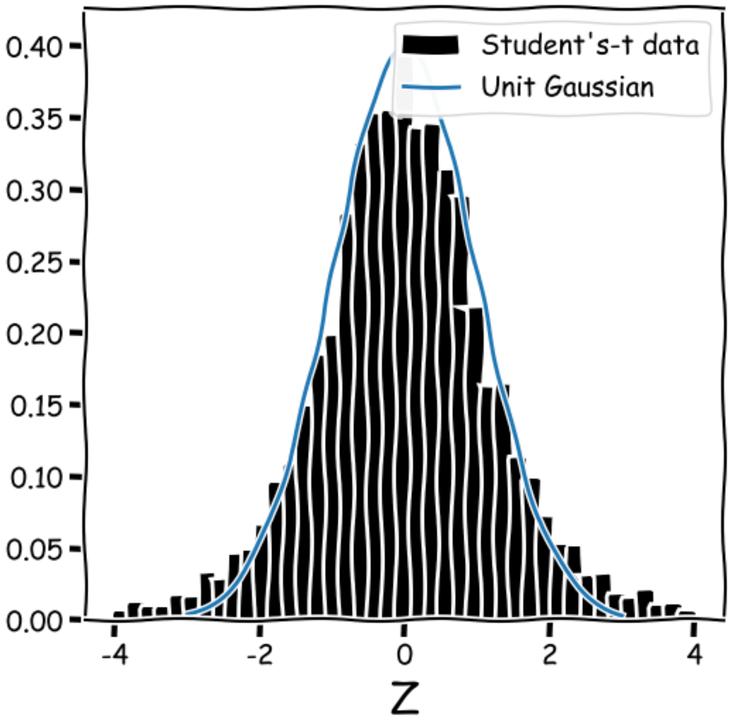
PDF $$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\,\nu}\;\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

When $\nu$=1 student's-t -> Cauchy

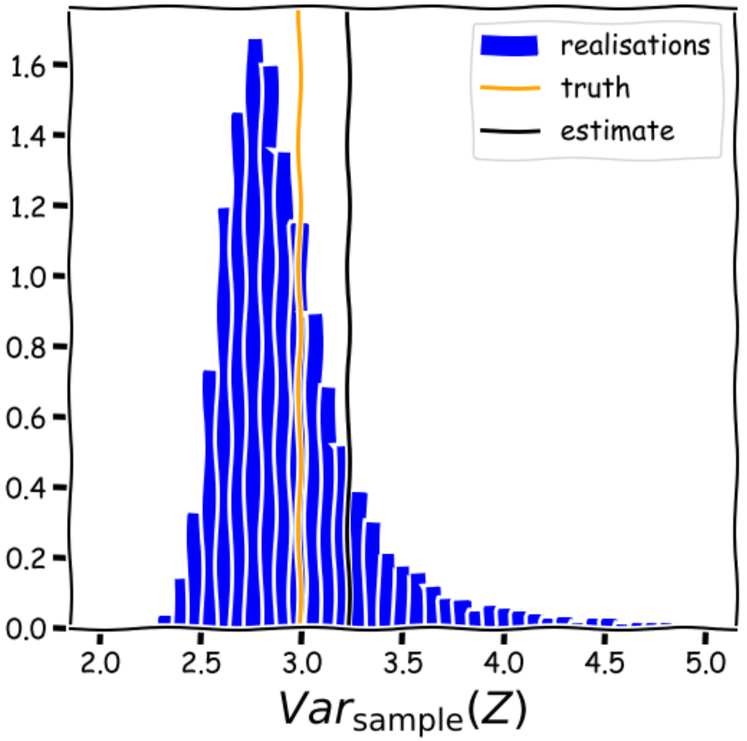
When $\nu \rightarrow \infty$, student's-t -> Gaussian

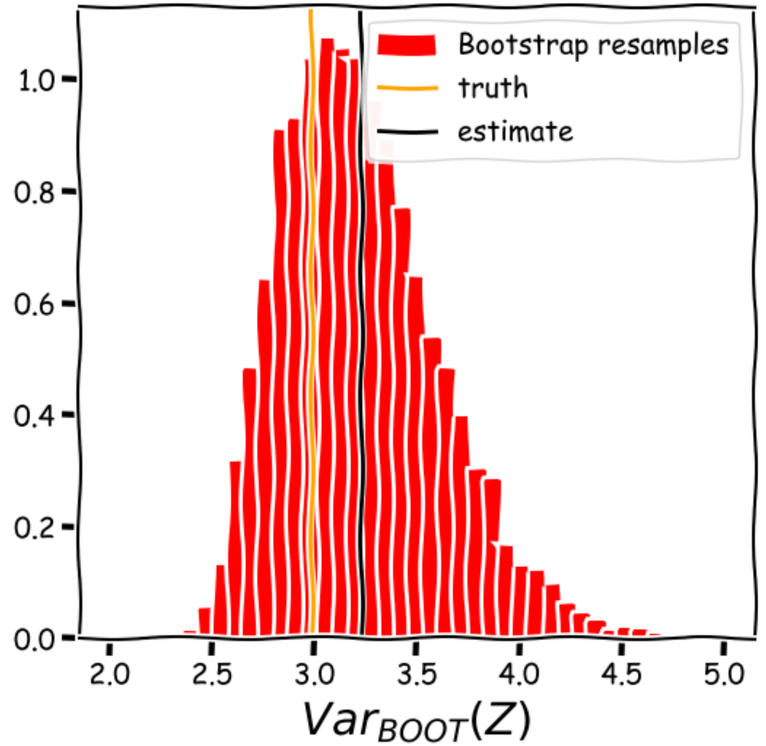
# Re-sampling methods - Bootstrap

***Bootstrap is great if you want know the probability distribution of your estimator  e.g. Is it Gaussian?***




Draws from Student's-T with $\nu = 3$ (thick tails)




NOTE: The bootstrap realizations are centred on the estimate which is itself a random variable. The simulated samples are centred on the input to the simulation. We don't know which is the truth!

# Re-sampling methods - Bootstrap

***REMINDER:***

***When re-sampling you should still use the so-called "Hartlap-factor". Same as you would when using simulations  to get your covariance***

$$\mathrm{C}_{\mathrm{corrected}}^{-1} = \frac{N_{\mathrm{JK}} - N_y - 2}{N_{\mathrm{JK}} - 1}\mathrm{C}^{-1}$$

# Re-sampling methods - Bootstrap

***Pros and cons***

***Can generate very large number of realizations*** $\frac{(2N-1)!}{(N-1)!\ N!}$ *(fun proof here)*

***Can probe non-Gaussian distributions***

***Covariance converges slowly***

***Can be computational expensive*** *(depends on estimator)*

***Doesn't work if you samples are from a "pathological distribution"*** *e.g. A Cauchy (Lorentz) distribution with an undefined mean (ratio of two gaussians)*

# References


- First proposal of jackknife by Quenouille. **Approximate tests of correlation in time series, J. of the Royal Statistical Society B 11: 68–84 (1949)**
- Named “Jackknife” by John Turkey in: **“Bias and Confidence in Not-Quite Large Sample. Annals of Mathematical Statistics, 29, 614. (1958)”** *I think…. *I can’t actually find this paper online!*


***More useful ones:***


- The Jackknife--A Review - R.G. Miller 1974
- A good primer
- Another good primer