Berkeley
MSSE
MOLECULAR SCIENCE & SOFTWARE ENGINEERING

Lecture 8:

Statistics for Data Science

Markus Hohle

University California, Berkeley

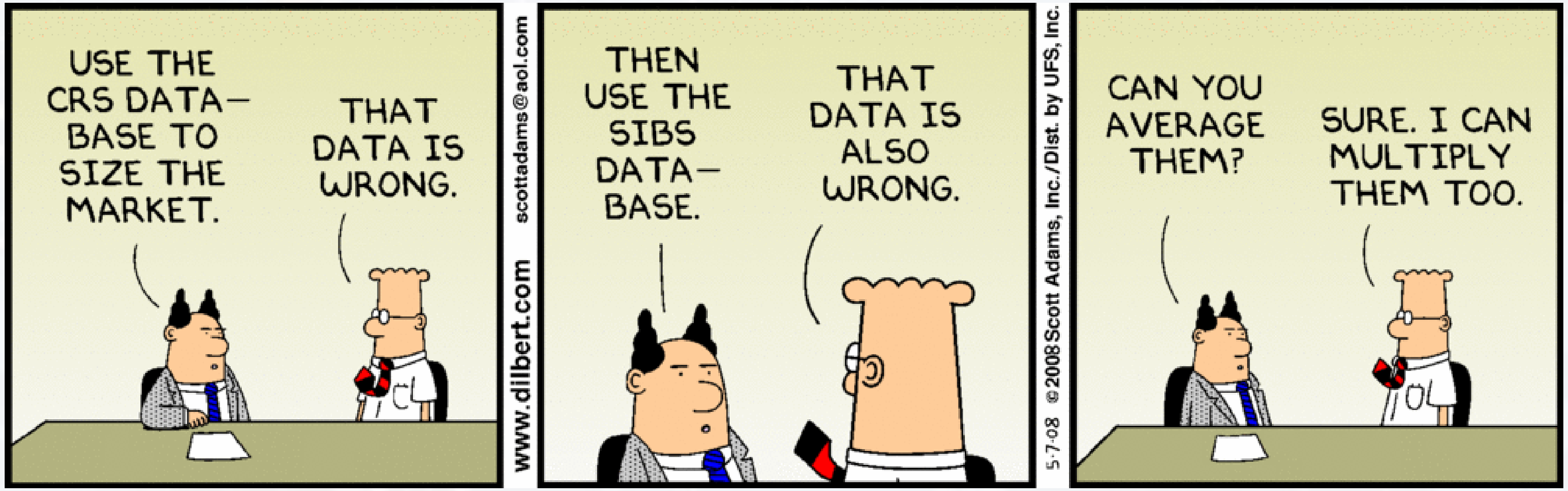
Numerical Methods for Computational Science

Course Map

Week 1:	Introduction to Scientific Computing and Python Libraries
Week 2:	Linear Algebra Fundamentals
Week 3:	Vector Calculus
Week 4:	Numerical Differentiation and Integration
Week 5:	Solving Nonlinear Equations
Week 6:	Probability Theory Basics
Week 7:	Random Variables and Distributions
Week 8:	**Statistics for Data Science**
Week 9:	Eigenvalues and Eigenvectors
Week 10:	Simulation and Monte Carlo Method
Week 11:	Data Fitting and Regression
Week 12:	Optimization Techniques
Week 13:	Machine Learning Fundamentals

Outline

- Expectation, Variance, and Covariance
- Hypothesis Testing
- Confidence Intervals

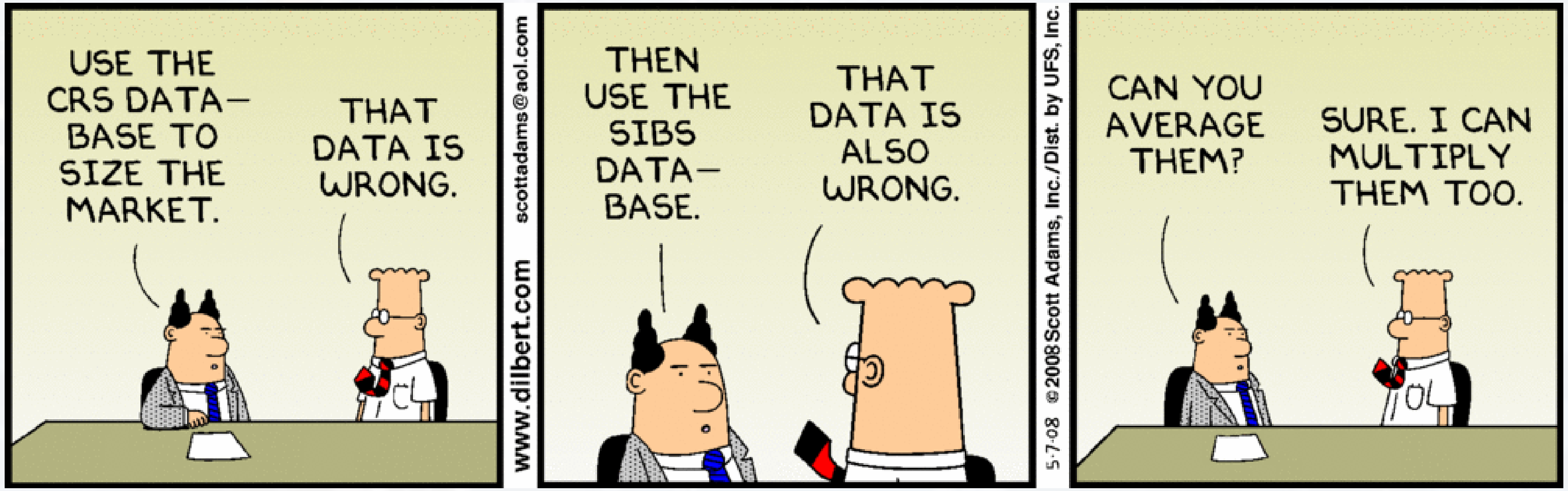


Outline

- Expectation, Variance, and Covariance

- Hypothesis Testing

- Confidence Intervals



	the mean μ	**the variance σ^2**
	(barycenter)	(natural scatter)
discrete (= countable)	$\mu = E(x) = \sum_i x_i\, p(x_i)$	$\sigma^2 = var(x) = \sum_i (x_i - \mu)^2\, p(x_i)$
continuous	$\mu = E(x) = \int x\, p(x)\, dx$	$\sigma^2 = var(x) = \int (x - \mu)^2\, p(x)\, dx$

$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

$$var(x) = \int (x-\mu)^2\, p(x)\, dx = E([x-\mu]^2)$$

variance can be interpreted as **mean of** $[x-\mu]^2$

$$= E(x^2 - 2x\mu + \mu^2)$$

$$= \int [x^2 - 2x\mu + \mu^2]\, p(x)\, dx$$

$$= \int x^2\, p(x)\, dx - 2\mu \int x\, p(x)\, dx + \mu^2 \int p(x)\, dx$$

$$= E(x^2) - 2\mu\, E(x) + \mu^2\, E(1)$$

2nd axiom $\int p(x)\, dx = 1$

$$= E(x^2) - 2\mu\, E(x) + \mu^2$$

$\mu = E(x)$

$$\boxed{\boldsymbol{\sigma^2 = E(x^2) - E(x)^2}}$$

$$\boxed{\sigma^2 = E(x^2) - E(x)^2}$$

$$\mu = E(x) = \int x\, p(x)\, dx$$

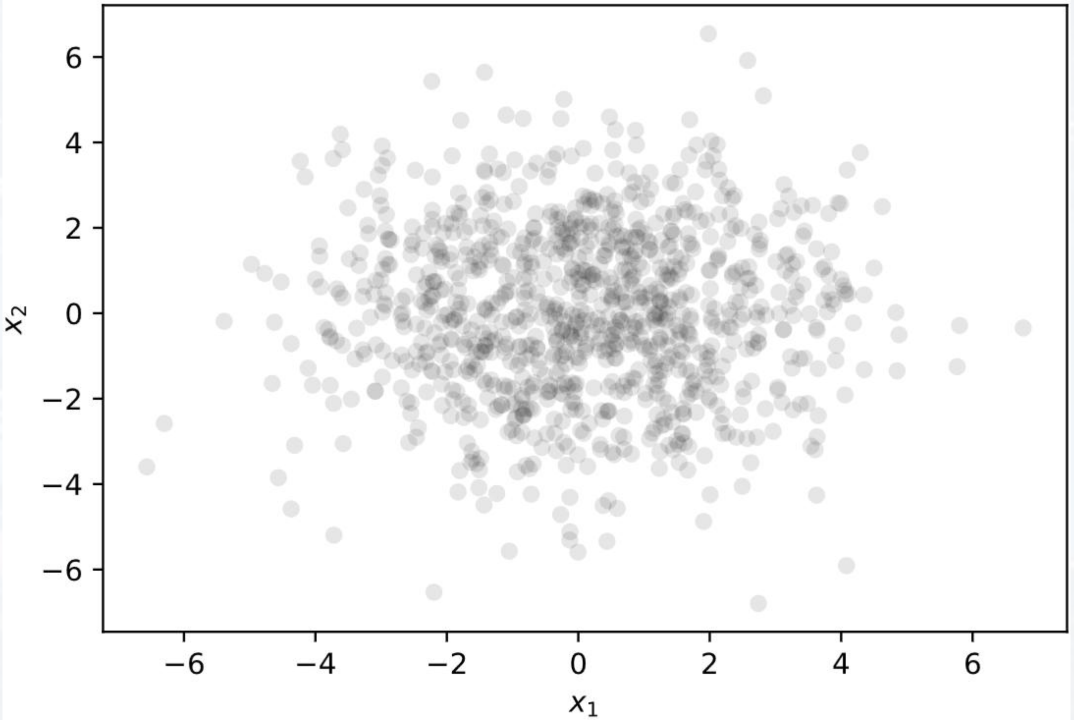
$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

plotting two sets of random number: x_1 and x_2

```
x1 = np.random.normal(0,2,(1000,))
x2 = np.random.normal(0,2,(1000,))

plt.scatter(x1, x2, color = 'k', alpha = 0.1, edgecolor = 'none')
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
```

x_1 and x_2 are unrelated and mutually **independent**
→ featureless data cloud



$$\boxed{\sigma^2 = E(x^2) - E(x)^2}$$

$$\mu = E(x) = \int x\, p(x)\, dx$$

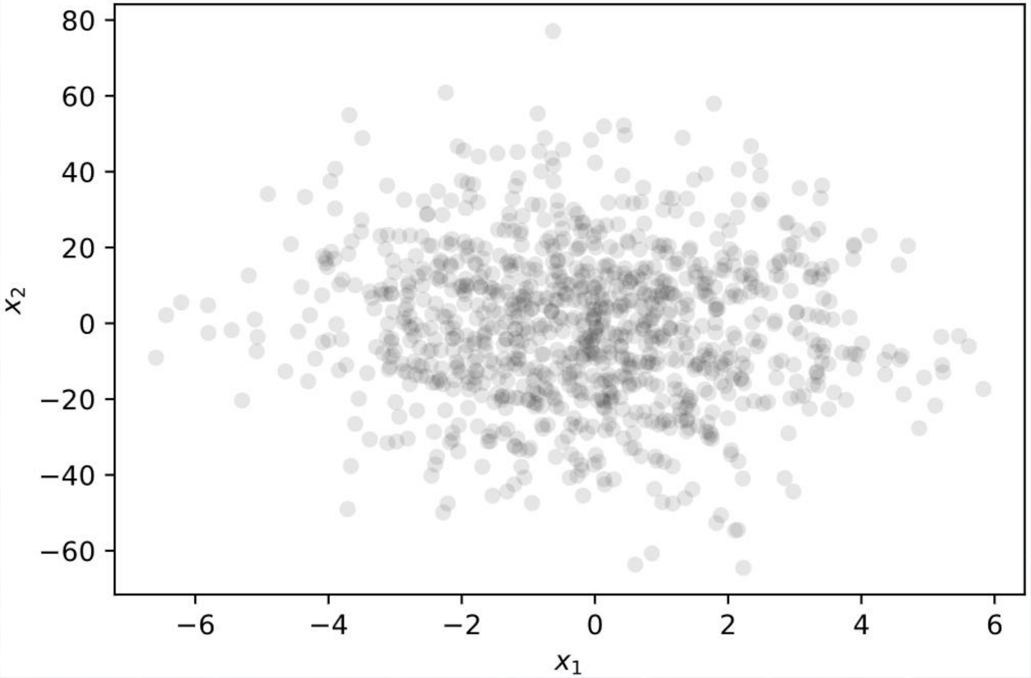
$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

plotting two sets of random number: x_1 and x_2

```
x1 = np.random.normal(0,2,(1000,))
x2 = np.random.normal(0,20,(1000,))

plt.scatter(x1, x2, color = 'k', alpha = 0.1, edgecolor = 'none')
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
```

x_1 and x_2 are unrelated and mutually **independent**
→ featureless data cloud



$$\boxed{\sigma^2 = E(x^2) - E(x)^2}$$

$$\mu = E(x) = \int x\, p(x)\, dx$$

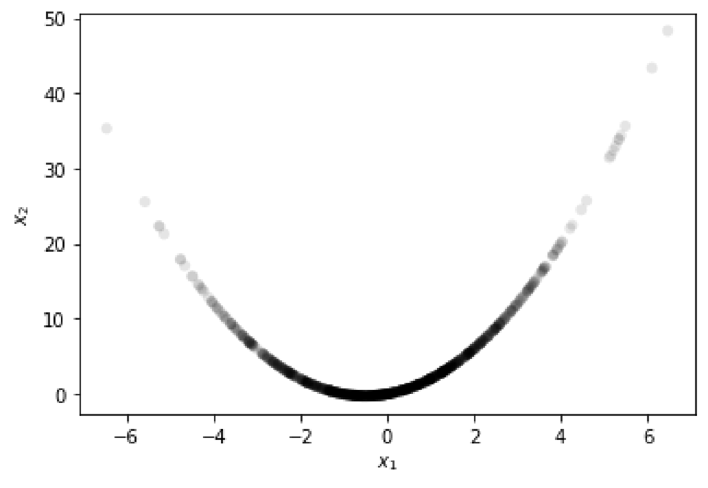
$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

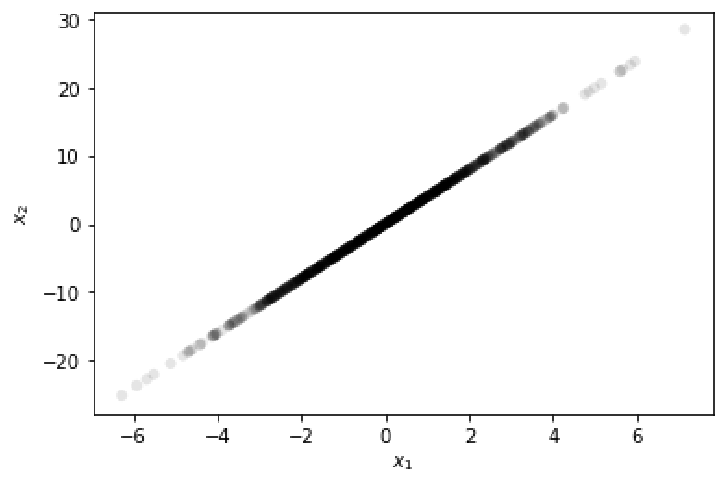
plotting two sets of random number: x_1 and x_2

```
x1 = np.random.normal(0,2,(1000,))
x2 = x1**2 + x1
#x2 = 4*x1

plt.scatter(x1, x2, color = 'k', alpha = 0.1, edgecolor = 'none')
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
```

based on the shape of the data cloud
→ prediction how x_1 and x_2 are related, i. e. how they **correlate**





$$\boxed{\sigma^2 = E(x^2) - E(x)^2}$$

$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

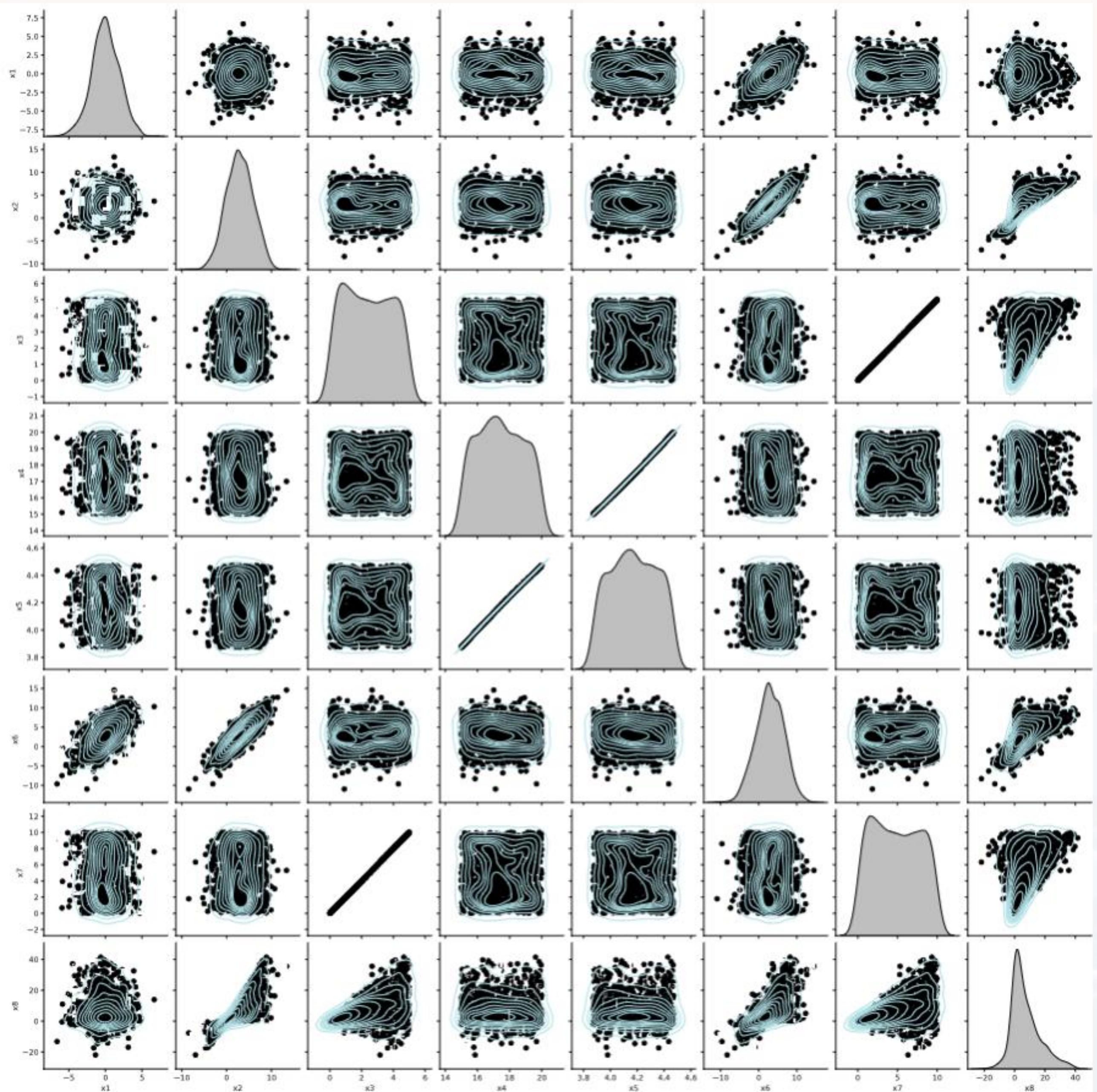
```
x1 = np.random.normal(0,2,(1000,))
x2 = np.random.normal(3,3,(1000,))

x3 = np.random.uniform(0,5,(1000,))
x4 = 5*np.random.uniform(3,4,(1000,))

x5 = np.sqrt(x4)
x6 = x1 + x2
x7 = 2*x3
x8 = x3*x2

All  = np.vstack((x1, x2, x3, x4, x5, x6, x7, x8))
data = pd.DataFrame(All.transpose(),
                    columns = ['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8'])

out = sns.pairplot(data, kind = "kde", \
                   plot_kws = {'color':[176/255, 224/255, 230/255]}, \
                   diag_kws = {'color':'black'})
out.map_offdiag(plt.scatter, color = 'black')
```



$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

based on the shape of the data cloud

→ prediction how x_1 and x_2 are related, i. e. how they **correlate**

→ how to quantify?

a, b = const

$$\boxed{\sigma^2 = E(x^2) - E(x)^2}$$

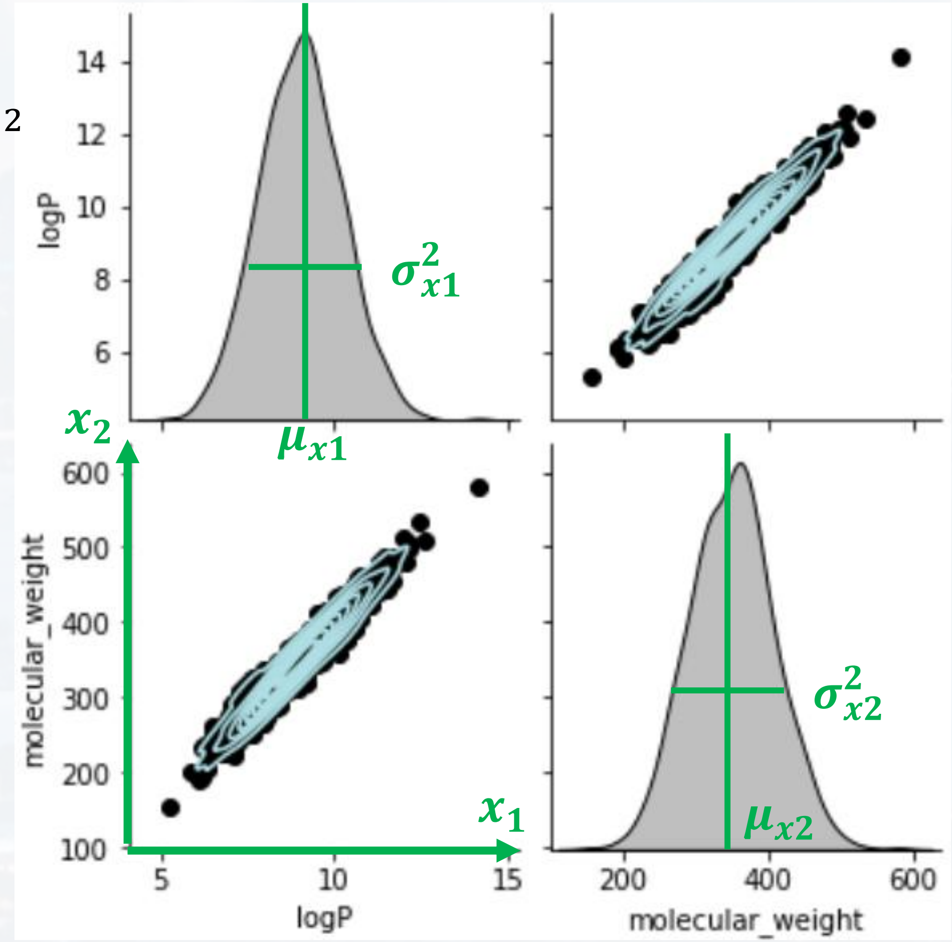
$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

$$var([a\,x_1 + b\,x_2]) = E([a\,x_1 + b\,x_2]^2) - E(a\,x_1 + b\,x_2)^2$$

$$= E(a^2\,x_1^2 + 2ab\,x_1 x_2 + b^2\,x_2^2) - E(a\,x_1 + b\,x_2)^2$$

$$= a^2 E(x_1^2) + 2ab\,E(x_1 x_2) + b^2\,E(x_2^2) - E(a\,x_1 + b\,x_2)^2$$



a, b = const

$$\boldsymbol{\sigma^2 = E(x^2) - E(x)^2}$$

$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

$$var([a\,x_1 + b\,x_2]) = E([a\,x_1 + b\,x_2]^2) - E(a\,x_1 + b\,x_2)^2$$

$$= E(a^2\,x_1^2 + 2ab\,x_1x_2 + b^2\,x_2^2) - E(a\,x_1 + b\,x_2)^2$$

$$= a^2E(x_1^2) + 2ab\,E(x_1x_2) + b^2\,E(x_2^2) - E(a\,x_1 + b\,x_2)^2$$

$$= a^2E(x_1^2) + 2ab\,E(x_1x_2) + b^2\,E(x_2^2) - [aE(x_1) + b\,E(x_2)]^2$$

$$= \underbrace{a^2E(x_1^2) - a^2E(x_1)^2}_{\boldsymbol{a^2\,var(x_1)}} + \underbrace{b^2\,E(x_2^2) - b^2E(x_2)^2}_{\boldsymbol{b^2\,var(x_2)}} + \underbrace{2ab\,E(x_1x_2) - 2abE(x_1)E(x_2)}_{\boldsymbol{2ab\,cov(x_1,x_2)}}$$

$$= \boldsymbol{a^2\,var(x_1)} + \boldsymbol{b^2\,var(x_2)} + \boldsymbol{2ab\,cov(x_1,x_2)}$$

$$\boldsymbol{cov(x_1,x_2) = cov(x_2,x_1)} = E(x_1x_2) - E(x_1)E(x_2)$$

covariance

$$\boldsymbol{cov(x_1, x_2) = cov(x_2, x_1)} = E(x_1 x_2) - E(x_1)E(x_2)$$

$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x - \mu)^2\, p(x)\, dx$$

What does the covariance tell us?

1) geometrical interpretation → next lecture
2) arithmetical interpretation

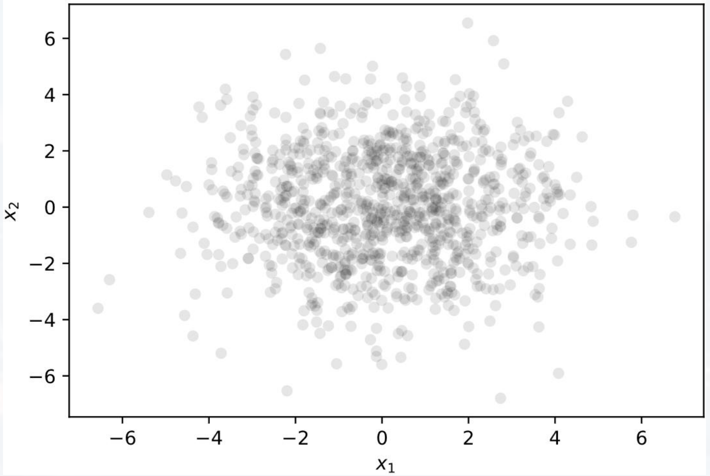
$$\boldsymbol{cov(x_1, x_2) = cov(x_2, x_1)} = E(x_1 x_2) - E(x_1)E(x_2)$$

$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

What does the covariance tell us?

1) geometrical interpretation → next lecture
2) **arithmetical interpretation**



a) x_1 and x_2 are independent

$$E(x_1 x_2) - E(x_1)E(x_2)$$

$$= \iint x_1 x_2\, p(x_1)\, p(x_2)\, dx_1 dx_2 - \int x_1\, p(x_1)\, dx_1 \int x_2\, p(x_2)\, dx_2$$

x_1 and x_2 are independent: x_1 is not a function of x_2 and vise verse
x_1 cannot be predicted by x_2 and vise verse

$$= \int x_1\, p(x_1)\, dx_1 \int x_2\, p(x_2)\, dx_2 - \int x_1\, p(x_1)\, dx_1 \int x_2\, p(x_2)\, dx_2 = 0$$

Covariance equals **zero** if samples are **independent**!

$$\boldsymbol{cov(x_1,x_2)} = \boldsymbol{cov(x_2,x_1)} = E(x_1 x_2) - E(x_1)E(x_2)$$

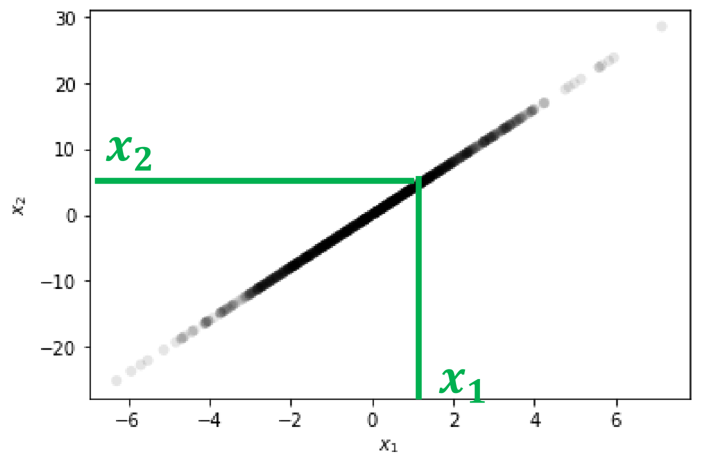
$$\mu = E(x) = \int x\, p(x)\, dx$$

$$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

What does the covariance tell us?

1) geometrical interpretation → next lecture

2) arithmetical interpretation



b) x_1 and x_2 are **not** independent

$$E(x_1 x_2) - E(x_1)E(x_2)$$

$$= \iint x_1 x_2\, p(x_1)\, p(x_2)\, dx_1 dx_2 - \int x_1\, p(x_1)\, dx_1 \int x_2\, p(x_2)\, dx_2$$

x_1 and x_2 are **not** independent:

x_1 ***is*** a function of x_2 and vise verse

x_1 ***can*** be predicted by x_2 to certain degree and vise verse

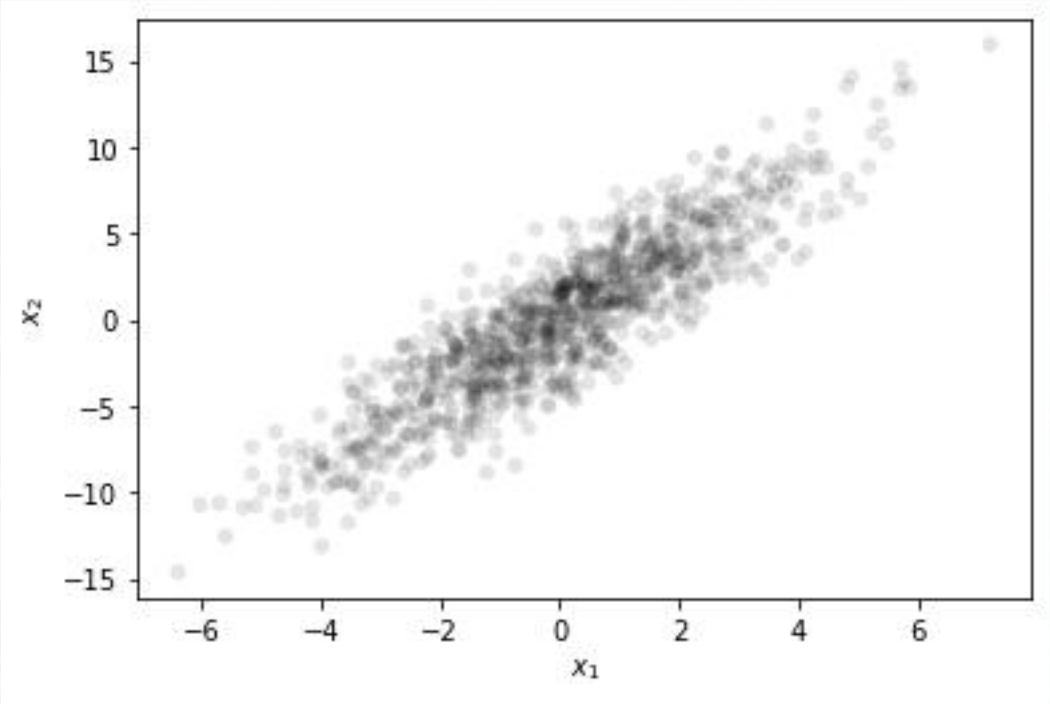
$$= \iint x_1\, p(x_1)\;\; x_2(x_1)\, p(x_2(x_1))\, dx_1 dx_2(x_1) - \int x_1\, p(x_1)\, dx_1 \int x_2\, p(x_2)\, dx_2$$

Covariance **does not** equal **zero**!

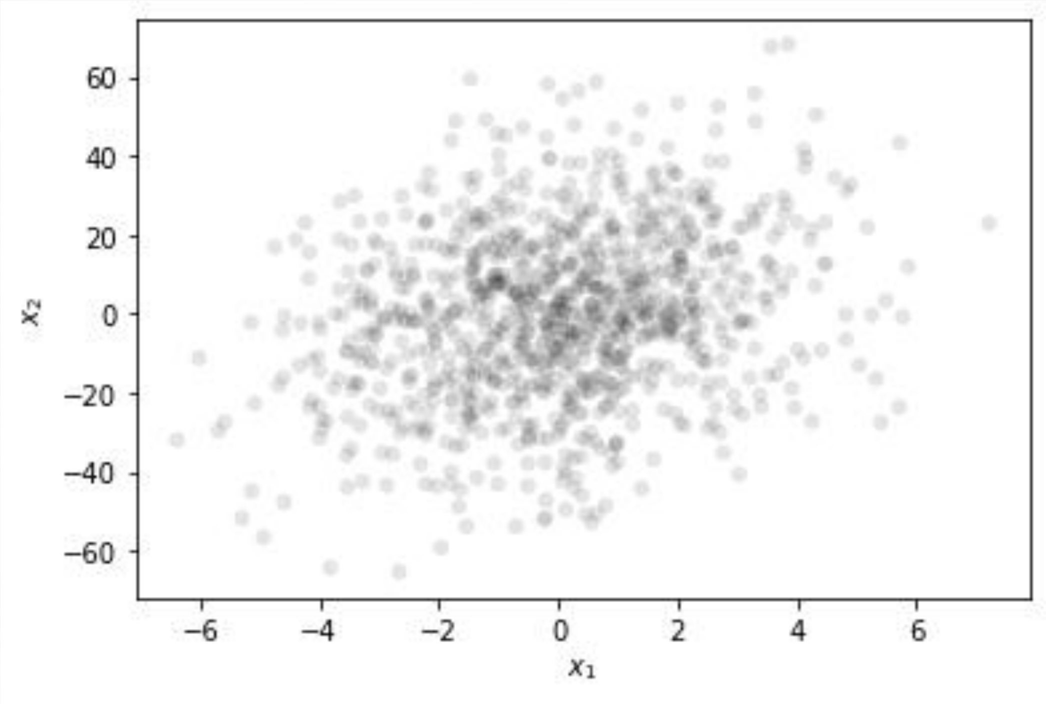
$$\boxed{cov(x_1, x_2) = cov(x_2, x_1) = E(x_1 x_2) - E(x_1)E(x_2)}$$

covariance

```
x1 = np.random.normal(0, 2, (1000,))
x2 = 2*x1 + np.random.normal(0, 2, (1000,))
```



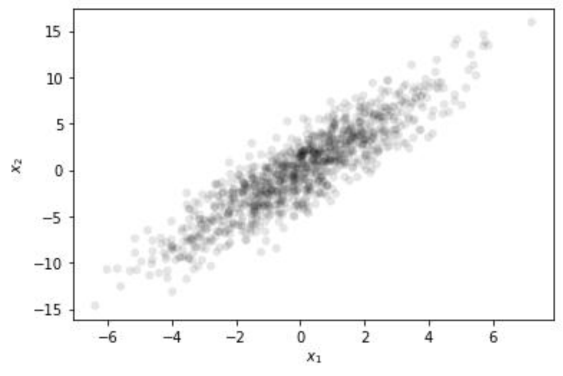

```
x1 = np.random.normal(0, 2, (1000,))
x2 = 2*x1 + np.random.normal(0, 20, (1000,))
```

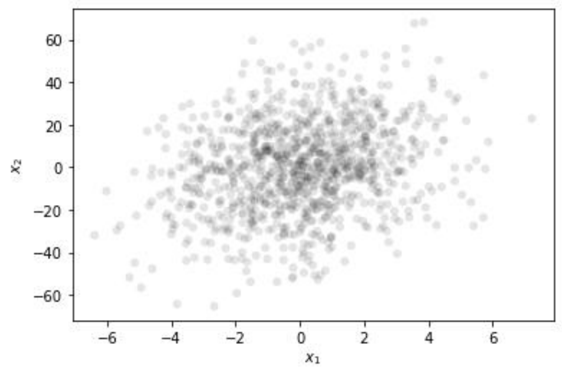


Same dependency, but different variance!

$$\boxed{cov(x_1, x_2) = cov(x_2, x_1) = E(x_1 x_2) - E(x_1)E(x_2)}$$

covariance





Same dependency, but different variance!

Need to scale for the variance!

Pearson's correlation coefficient

$$\boxed{\rho(x_1, x_2) = \frac{cov(x_1, x_2)}{\sqrt{\sigma_1^2 \sigma_2^2}}}$$

$\rho(x_1, x_2)$:

- ranges from -1 to +1
- zero: no correlation
  (completely independent)
- -1: max anti correlation
- +1: max correlation

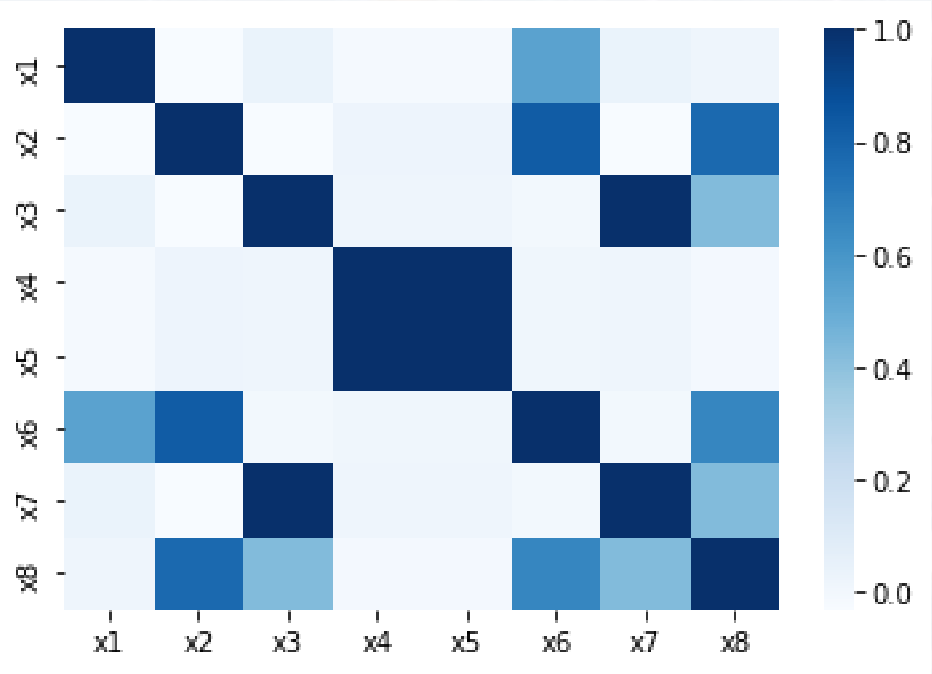
$$\boldsymbol{cov(x_1, x_2) = cov(x_2, x_1)} = E(x_1 x_2) - E(x_1)E(x_2)$$

covariance

$$\boldsymbol{\rho(x_1, x_2) = \frac{cov(x_1, x_2)}{\sqrt{\sigma_1^2 \sigma_2^2}}}$$

Pearson's correlation coefficient

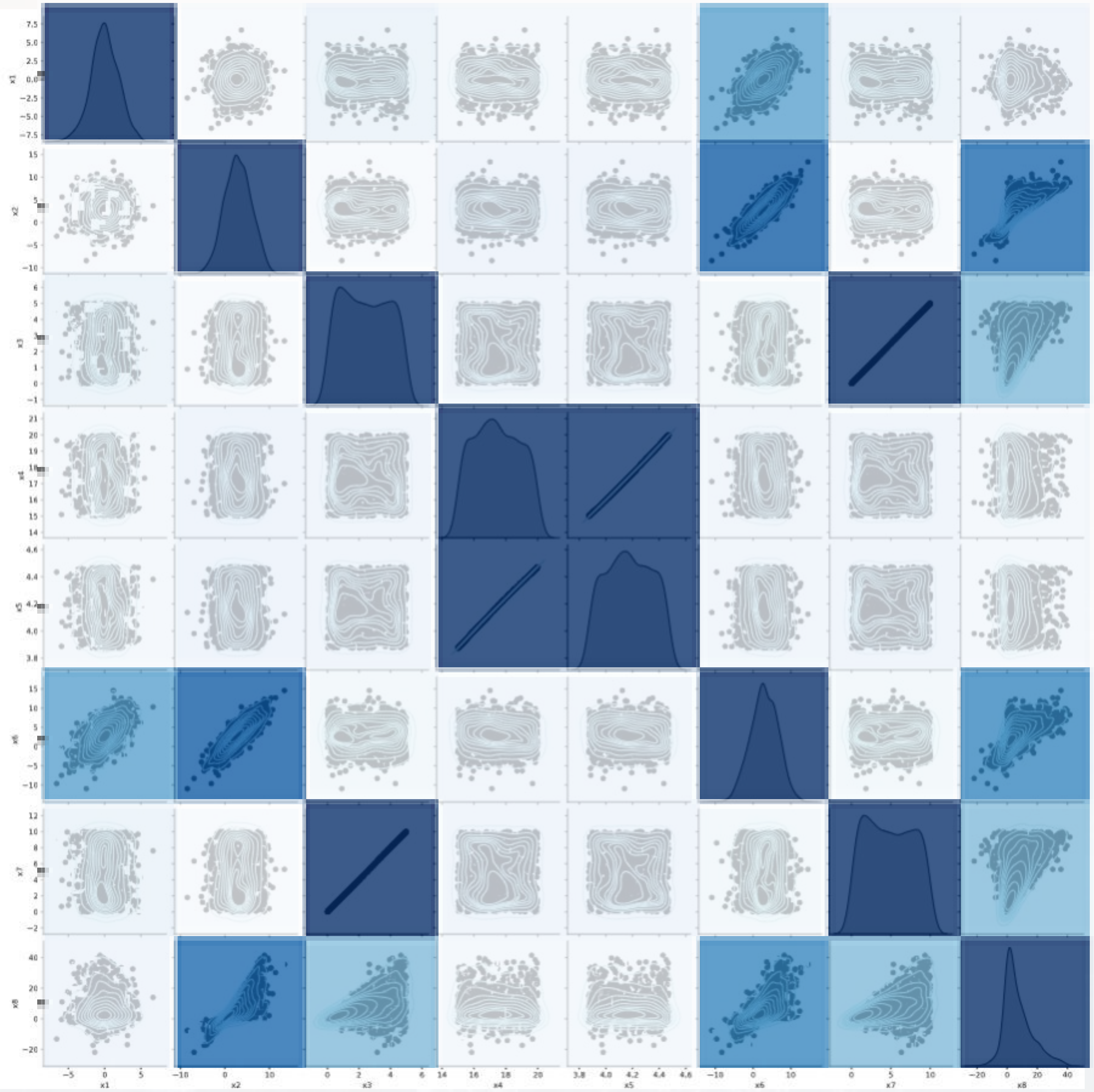
```
sns.heatmap(data.corr(), cmap = "Blues")
```



$\boldsymbol{\rho(x_1, x_2)}$:

- ranges from -1 to +1
- zero: no correlation

  (completely independent)
- -1: max anti correlation
- +1: max correlation



$\rho(x_1, x_2)$:

- ranges from -1 to +1
- zero: no correlation
  (completely independent)
- -1: max anti correlation
- +1: max correlation

Important quantities you should know:

mean $$\mu = E(x) = \int x\, p(x)\, dx$$

median *m* $$\int_a^{m} p(x)\, dx = \frac{1}{2}$$

variance $$\sigma^2 = var(x) = \int (x-\mu)^2\, p(x)\, dx$$

$$\sigma^2 = E(x^2) - E(x)^2$$

$$\sigma_{tot}^2 = \sigma_1^2 + \sigma_2^2 + 2\, cov(x_1, x_2)$$

covariance $$cov(x_1, x_2) = E(x_1 x_2) - E(x_1)E(x_2)$$

correlation coefficient $$\rho(x_1, x_2) = \frac{cov(x_1, x_2)}{\sqrt{\sigma_1^2 \sigma_2^2}}$$

note:

$$\int (x-\mu)^n\, p(x)\, dx$$

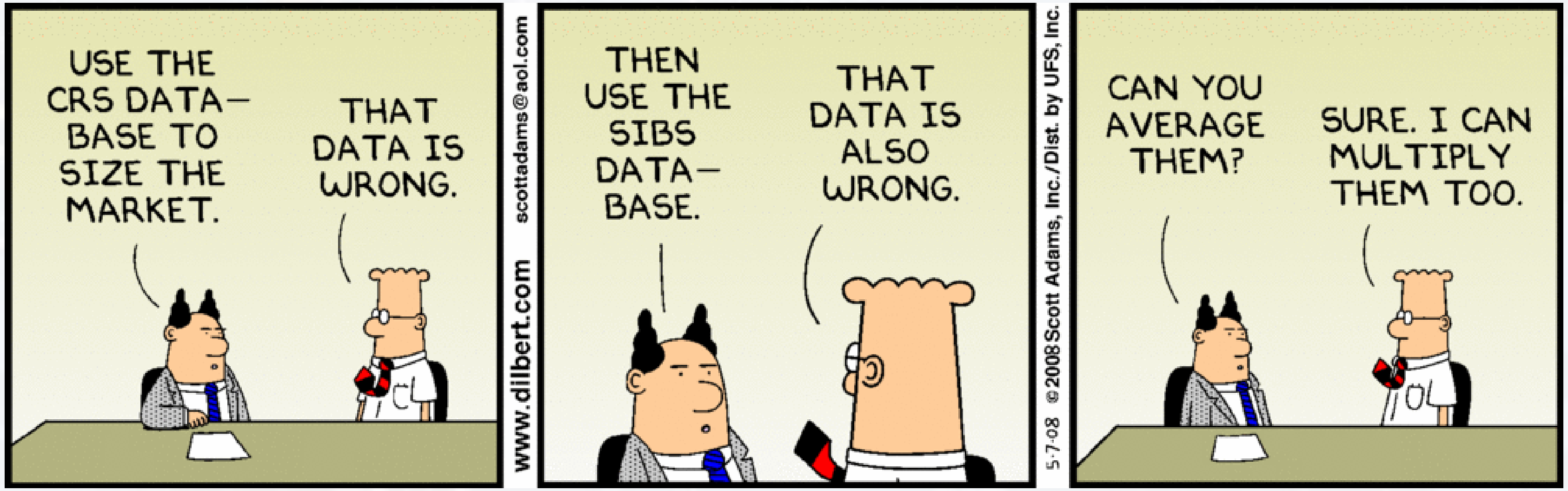
called n-th *moment* of a pdf

Outline

- Expectation, Variance, and Covariance

- Hypothesis Testing

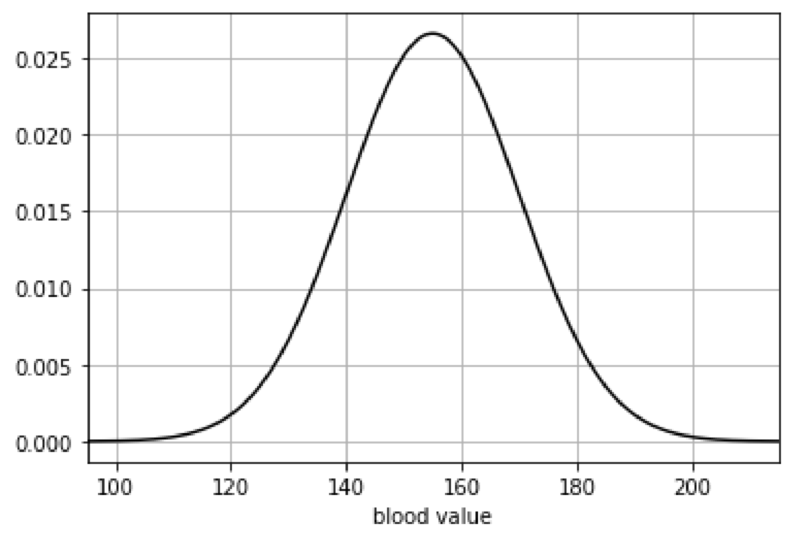
- Confidence Intervals



standard frequentist way (not Bayesian):

1) assume a **likelihood function *L* as your model**, aka **null hypothesis H_0**
2) take a datapoint x_0
3) calculate the ***probability P*** given ***L*** i. e. **given H_0** is true, **that x_0 has this value** or a **more extreme value**
4) accepting or rejecting H_0 based on ***P*** and the threshold α

example: 1) a healthy person has a blood value that follows a **normal distribution** with $\mu = 155, \sigma = 15$, i. e. $H_0: N(\mu = 155, \sigma = 15)$



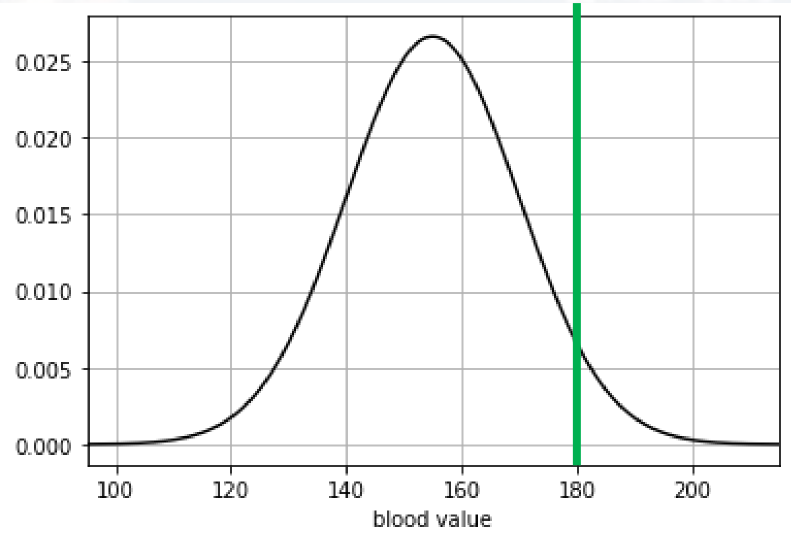
standard frequentist way (not Bayesian):

1) assume a **likelihood function *L* as your model**, aka **null hypothesis H_0**
2) take a datapoint x_0
3) calculate the ***probability P*** given ***L*** i. e. **given H_0** is true, **that x_0 has this value** or a **more extreme value**
4) accepting or rejecting H_0 based on ***P*** and the threshold α

example:

1) a healthy person has a blood value that follows a **normal distribution** with $\mu = 155, \sigma = 15$, i. e. $H_0: N(\mu = 155, \sigma = 15)$

2) a patient has the value x_0 **= 180**



standard frequentist way (not Bayesian):

1) assume a **likelihood function *L* as your model**, aka **null hypothesis H_0**
2) take a datapoint x_0
3) calculate the ***probability P*** given ***L*** i. e. **given H_0** is true, **that x_0 has this value** or a **more extreme value**
4) accepting or rejecting H_0 based on ***P*** and the threshold α

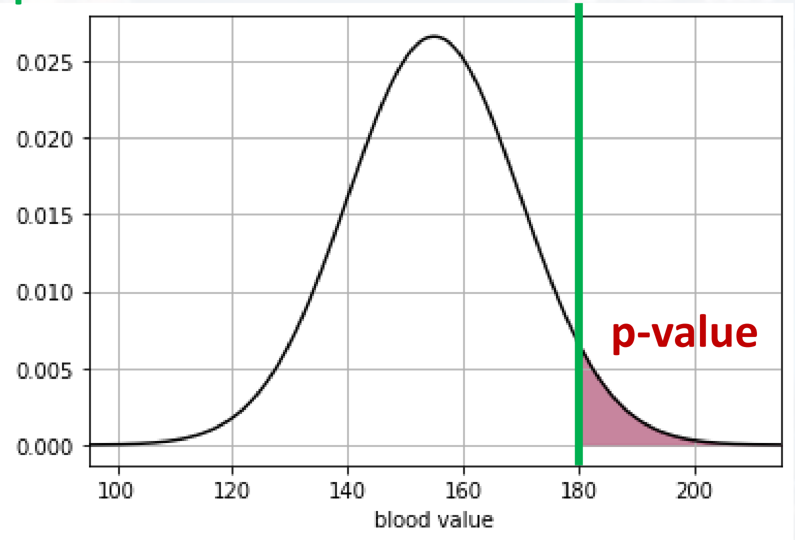
example:

1) a healthy person has a blood value that follows a **normal distribution** with

$\mu = 155, \sigma = 15$, i. e. $H_0: N(\mu = 155, \sigma = 15)$

2) a patient has the value x_0 **= 180**

3) probability $P(x \geq 180 | H_0) = 0.048$ called **p-value**



standard frequentist way (not Bayesian):

1) assume a **likelihood function *L* as your model**, aka **null hypothesis H_0**
2) take a datapoint x_0
3) calculate the ***probability P*** given ***L*** i. e. **given H_0** is true, **that x_0 has this value** or a **more extreme value**
4) accepting or rejecting H_0 based on ***P*** and the threshold α

example:

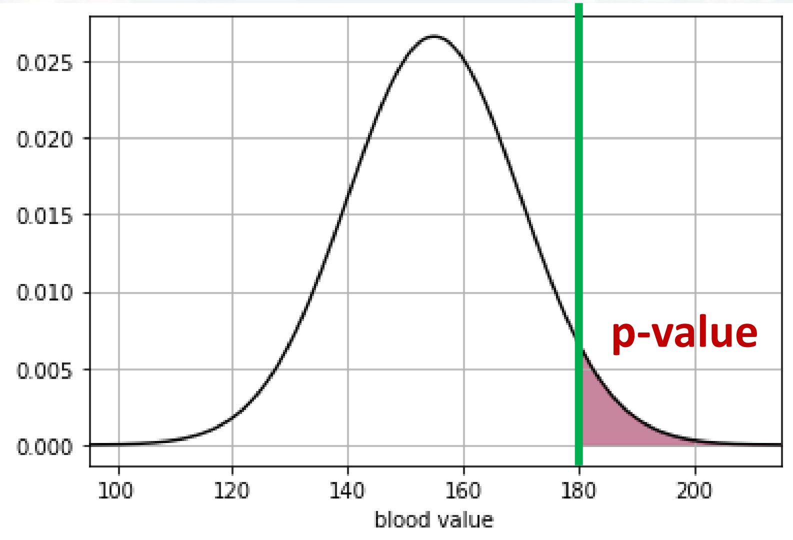
1) a healthy person has a blood value that follows a **normal distribution** with $\mu = 155, \sigma = 15$, i. e. $H_0: N(\mu = 155, \sigma = 15)$

2) a patient has the value x_0 **= 180**

3) probability $P(x \geq 180 | H_0) = 0.048$ called **p-value**

4) for **α = 0.05**, H_0 is **rejected**, i. e. patient is **not healthy**

→ alternative hypothesis H_1: **not healthy**



example:

1) a healthy person has a blood value that follows a **normal distribution** with $\mu = 155, \sigma = 15$, i. e. $\boldsymbol{H_0}: N(\boldsymbol{\mu = 155, \sigma = 15})$

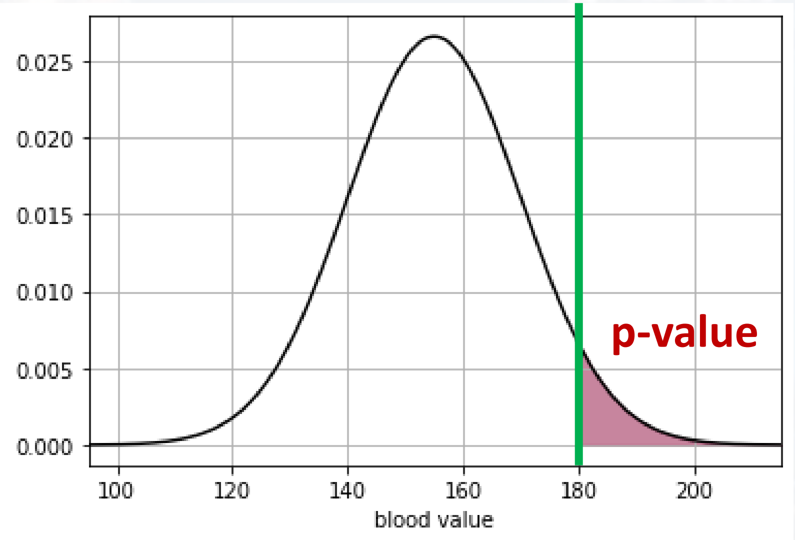
2) a patient has the value $\boldsymbol{x_0}$ **= 180**

3) probability $\boldsymbol{P(x \geq 180|H_0) = 0.048}$ called **p-value**

4) for **α = 0.05**, H_0 is **rejected**, i. e. patient is **not healthy**

→ alternative hypothesis $\boldsymbol{H_1}$: **not healthy**

We just performed a so-called Z – Test
(comparing one value to a normal distribution)



standard frequentist way (not Bayesian):

1) assume a **likelihood function *L* as your model**, aka **null hypothesis H_0**
2) take a datapoint x_0
3) calculate the ***probability P*** given ***L*** i. e. **given H_0** is true, **that x_0 has this value** or a **more extreme value**
4) accepting or rejecting H_0 based on ***P*** and the threshold α

caveats:

- the model ***L*** for H_0 has to **be known**

- the p-value $P(x \geq x_0|H_0)$ **does not** tell if H_0 is true or not

- the p-value $P(x \geq x_0|H_0)$ **does not** tell which hypothesis is more likely

- the p-value just gives $P(x \geq x_0|H_0)$

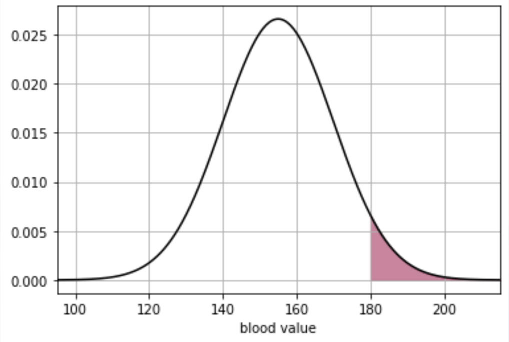
- the threshold α for accepting/rejecting H_0 is **arbitrary**

- we are aiming on disproving a hypothesis, by assuming it is true,

without leading to a contradiction

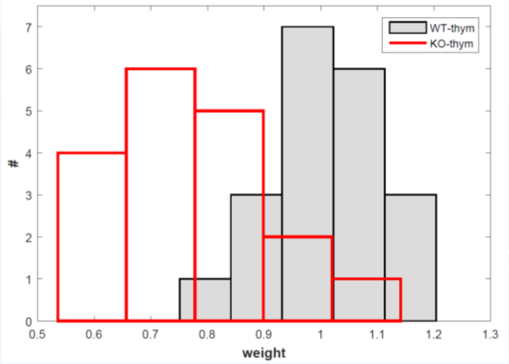
- data point *x* versus **normal**
- $H_0: x \epsilon N(\mu, \sigma)$

Z-test



- **normal** versus another **normal**
- two samples of sizes n_1 and n_2

t-test

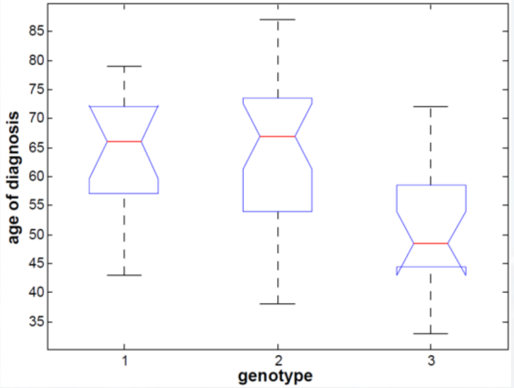


- two tail $H_0: \mu_1 = \mu_2, H_1: \mu_1 \neq \mu_2$
- right tail $H_0: \mu_1 < \mu_2, H_1: \mu_1 > \mu_2$
- left tail $H_0: \mu_1 > \mu_2, H_1: \mu_1 < \mu_2$

- **N normal dist.**
- N samples of sizes n_i

- $H_0: \mu_i = \mu_j \ \forall \ i, j$, $H_1: \mu_i \neq \mu_j$ for at least one pair i, j

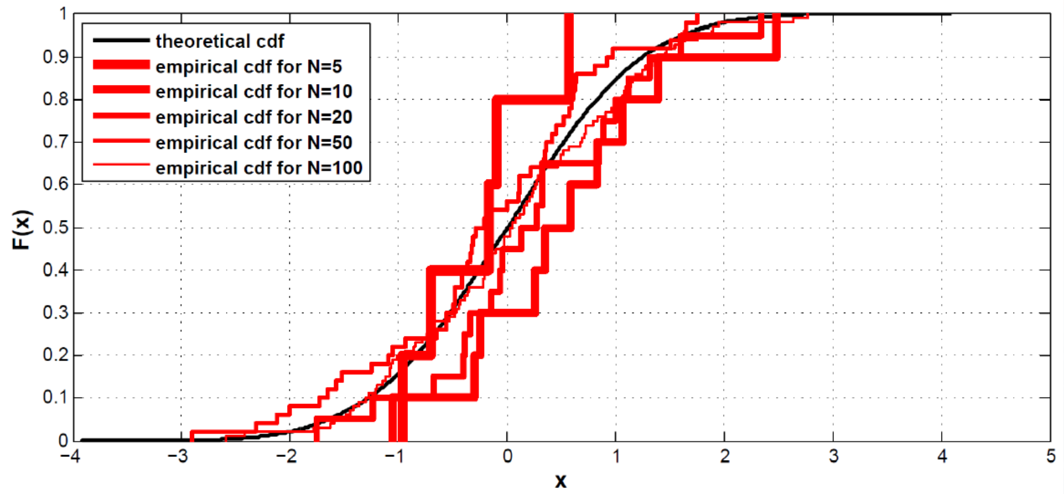
ANalysis **O**f **VA**riance



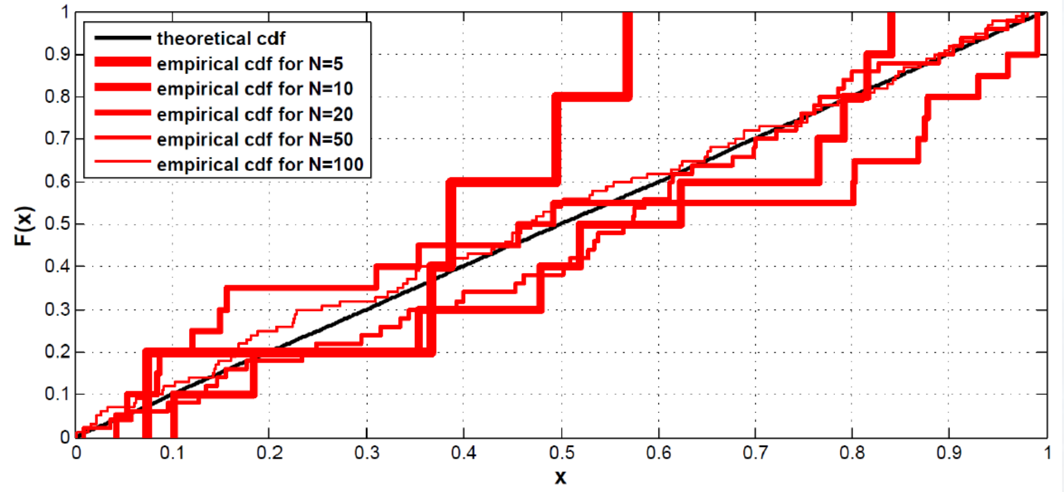
testing if a data set follows a particular distribution

Kolmogorov – **S**mirnov – test (KS test)

example: normal distribution:



example: uniform distribution:



testing if a data set follows a particular distribution

ranking tests (Wilcoxon)

…and many more…

They all generate a test statistic: Z-value, t-value, F-value etc
→ from that: calculating a p-value

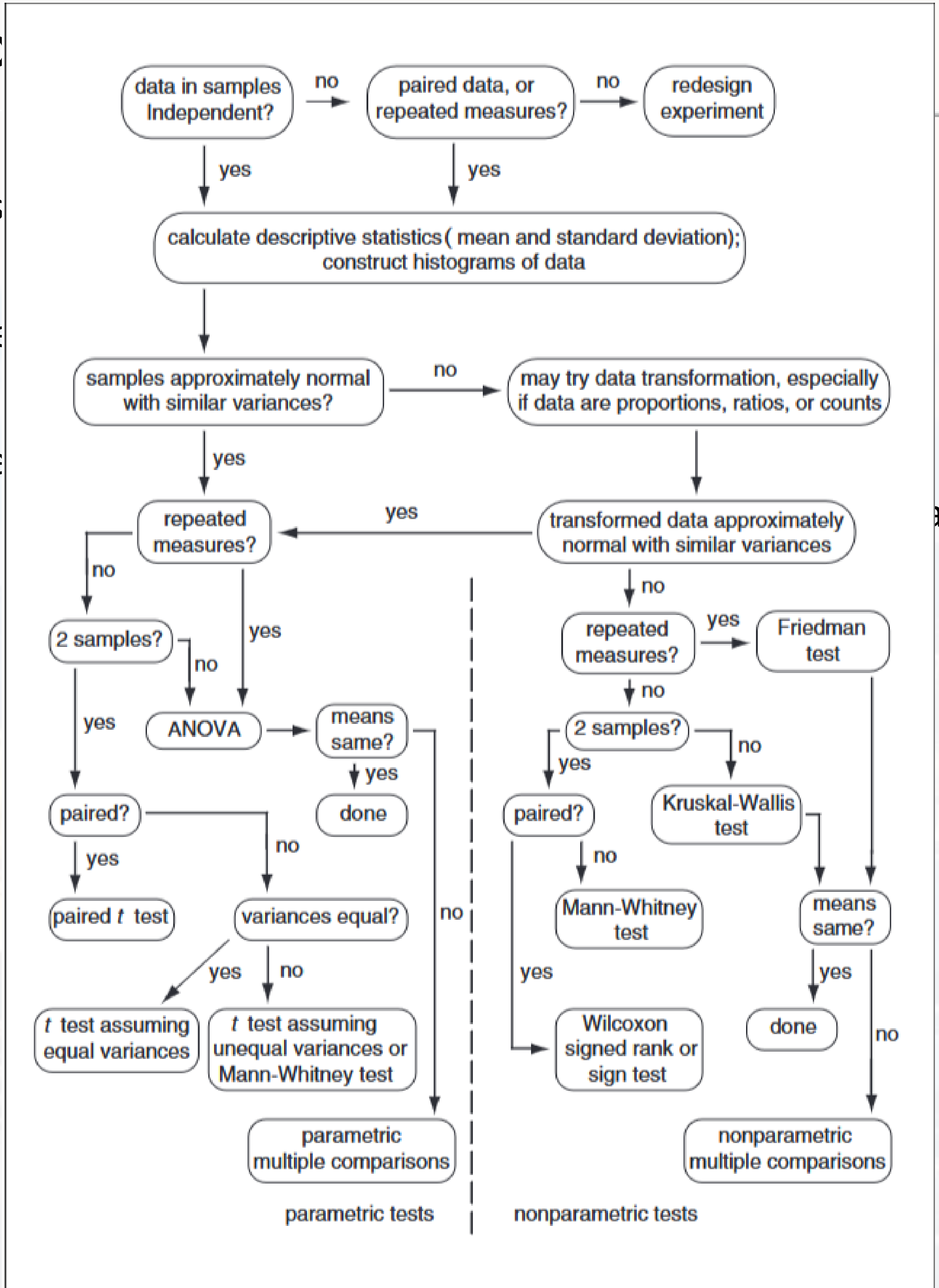
testing if a data s

ranking tests (Wilcoxon)

…and many more

They all generate

alue



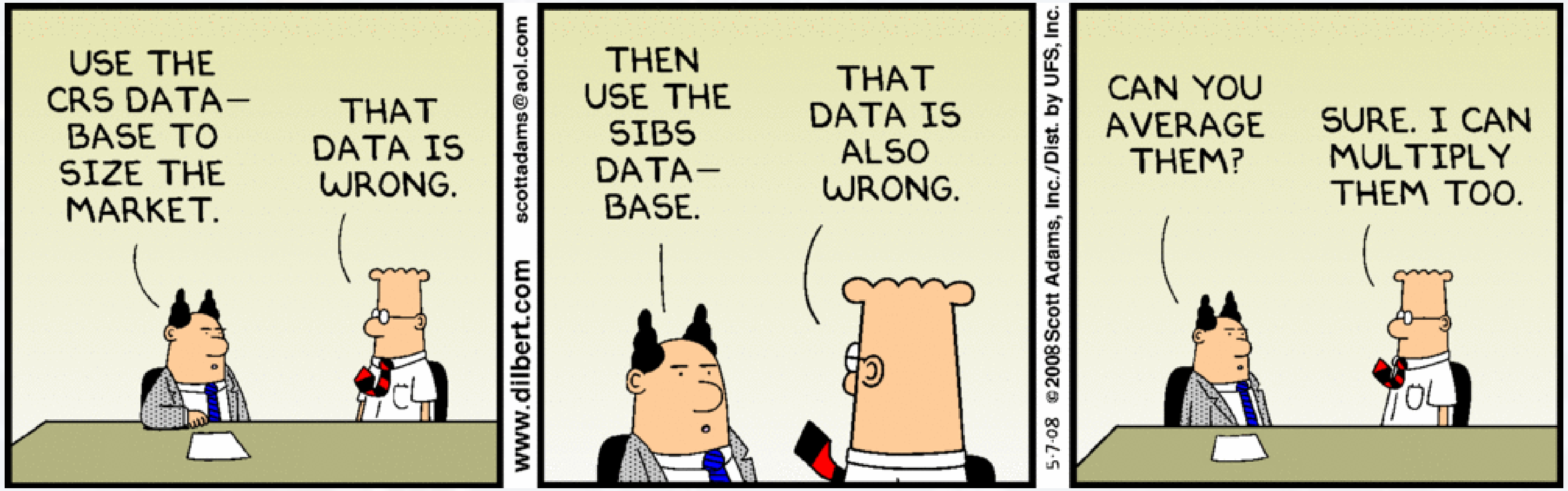
Current Protocols in Molecular Biology (1998) A.3I.1-A.3I.22

Outline

- Expectation, Variance, and Covariance

- Hypothesis Testing

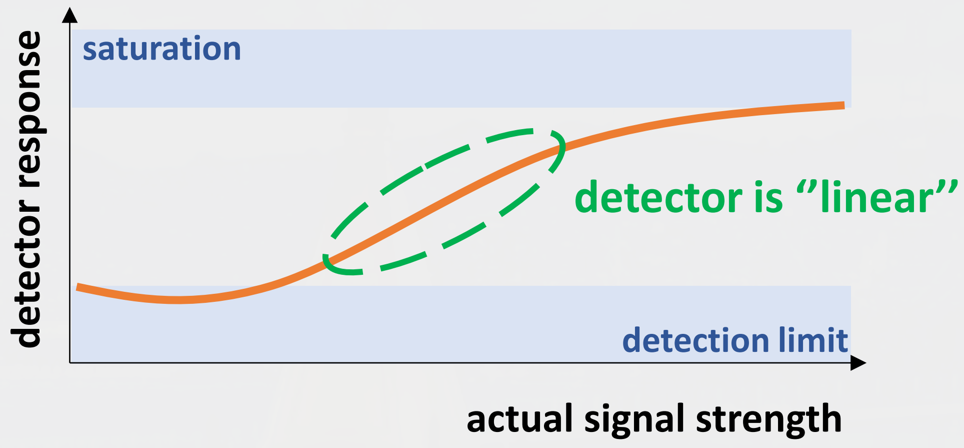
- **Confidence Intervals**



How sure can we be about a measured value?

two kinds of errors

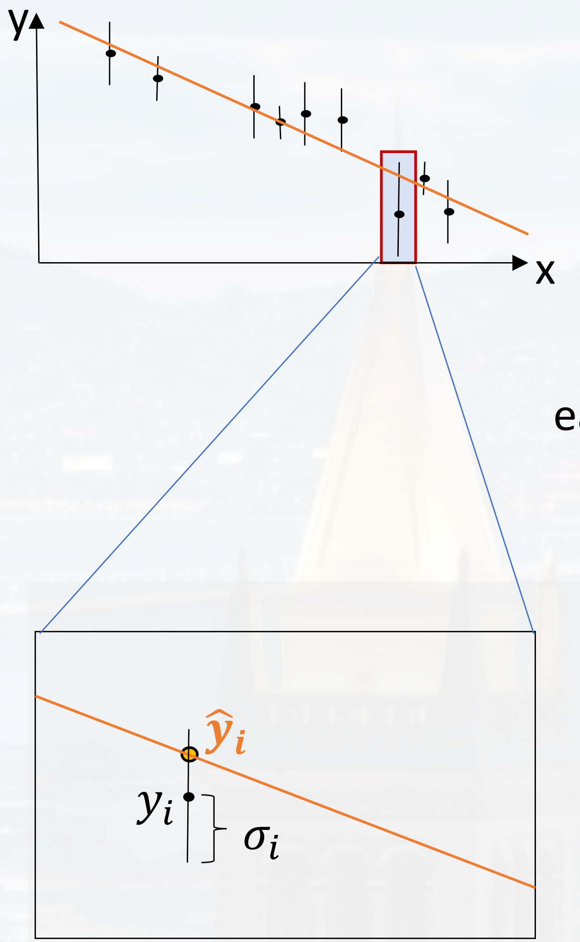
systematic errors: calibration, non-linearity of the detector



statistical errors: limited precision, natural variance of the data
→ spread of the data around **an average value**

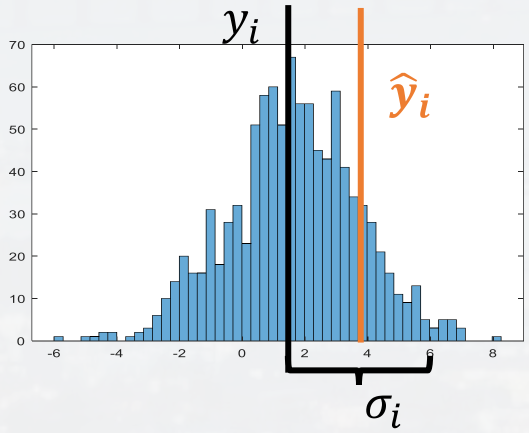
assumption: far from the detection limit and saturation:

the spread follows approximately a **normal distribution.**

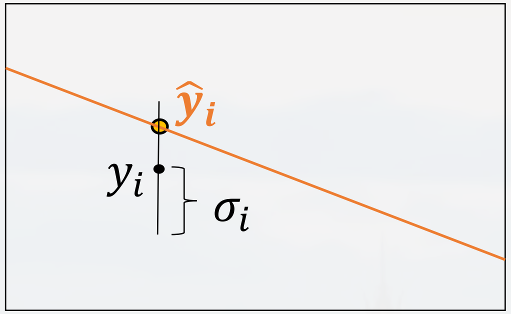


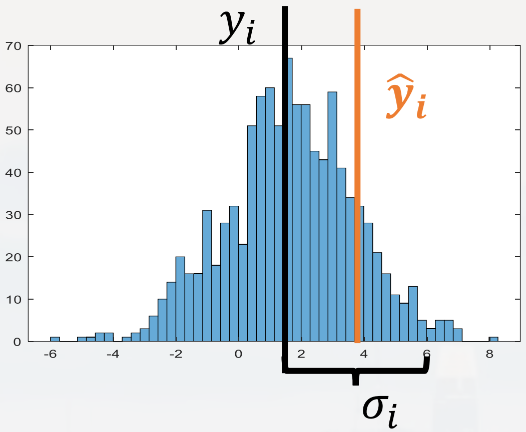
fitting a model ($\hat{y}_i$, orange line) to data points (x_i, y_i) each with an error bar $\boldsymbol{\sigma_i}$

each data point x_i has been drawn from $\boldsymbol{N(\mu_i = y_i, \sigma_i)}$

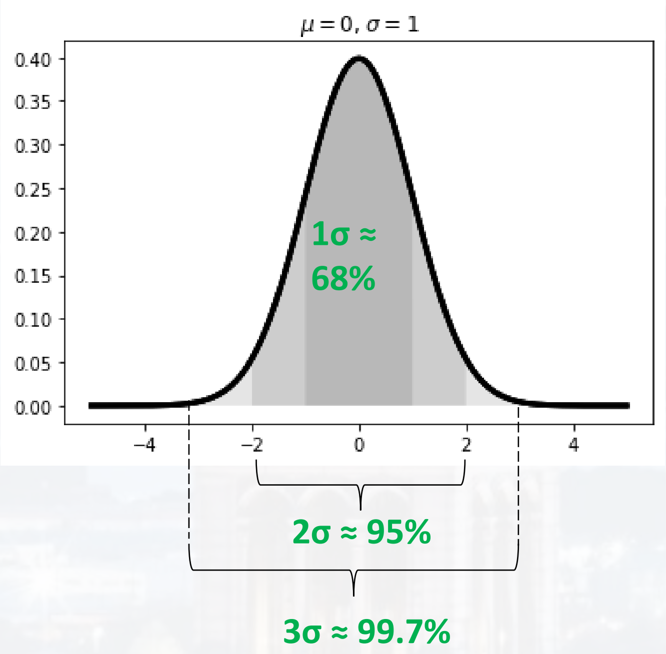


$$p_i(y_i|\hat{y}_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} exp\left[-\frac{1}{2}\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}\right]$$





$$p_i(y_i|\hat{y}_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} exp\left[-\frac{1}{2}\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}\right]$$



for large **(> 50…100)** N **(number of data points)**:

≈ **2/3** of the data points should be consistent with the model within their **1σ** error bars

≈ **95%** of the data points should be consistent with the model within their **2σ** error bars

≈ **99.7%** of the data points should be consistent with the model within their **3σ** error bars

$$p_i(y_i|\hat{y}_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} exp\left[-\frac{1}{2}\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}\right]$$

based on this model: reduced chi square

$$\boxed{\chi^2_{red} = \frac{1}{df}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right)^2}$$

$$df = N - p - 1$$

N : number of data points
p: number of fit parameter (model)

given a fitted model: χ^2_{red} is a **measure of the fit quality**!

$$p_i(y_i|\hat{y}_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} exp\left[-\frac{1}{2}\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}\right]$$

based on this model: reduced chi square

$$\boxed{\chi^2_{red} = \frac{1}{df}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right)^2}$$

$$df = N - p - 1$$

N : number of data points
p: number of fit parameter (model)

given a fitted model: χ^2_{red} is a **measure of the fit quality**!

example

good fit: $y_i - \hat{y}_i$ should be within σ_i for 2/3 of all data points, see $\boldsymbol{N(\mu_i = y_i, \sigma_i)}$

therefore $\frac{y_i - \hat{y}_i}{\sigma_i} \approx 1$

therefore $\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right)^2 \approx N$

N should be **much larger** than *p*, therefore df $\approx N$

hence, $\chi^2_{red} \approx 1$

$$p_i(y_i|\hat{y}_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} exp\left[-\frac{1}{2}\frac{(y_i - \hat{y}_i)^2}{\sigma_i^2}\right]$$

based on this model: reduced chi square

$$\boxed{\chi^2_{red} = \frac{1}{df}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right)^2}$$

$$df = N - p - 1$$

N : number of data points
p: number of fit parameter (model)

given a fitted model:
χ^2_{red} is a measure of the fit quality

for large **(> 50…100)** N **(number of data points)**:

≈ **2/3** of the data points should be consistent with the model within their **1σ** error bars

≈ **95%** of the data points should be consistent with the model within their **2σ** error bars

≈ **99.7%** of the data points should be consistent with the model within their **3σ** error bars

$$p_i(y_i|\hat{y}_i,\sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} exp\left[-\frac{1}{2}\frac{(y_i-\hat{y}_i)^2}{\sigma_i^2}\right]$$

based on this model: reduced chi square

$$\boxed{\chi^2_{red} = \frac{1}{df}\sum_{i=1}^{N}\left(\frac{y_i - \hat{y}_i}{\sigma_i}\right)^2}$$

$$df = N - p - 1$$

N : number of data points
p: number of fit parameter (model)

given a fitted model:
χ^2_{red} is a measure of the fit quality

for large **(> 50…100)** N **(number of data points):**

≈ **2/3** of the data points should be consistent with the model within their **1σ** error bars

≈ **95%** of the data points should be consistent with the model within their **2σ** error bars

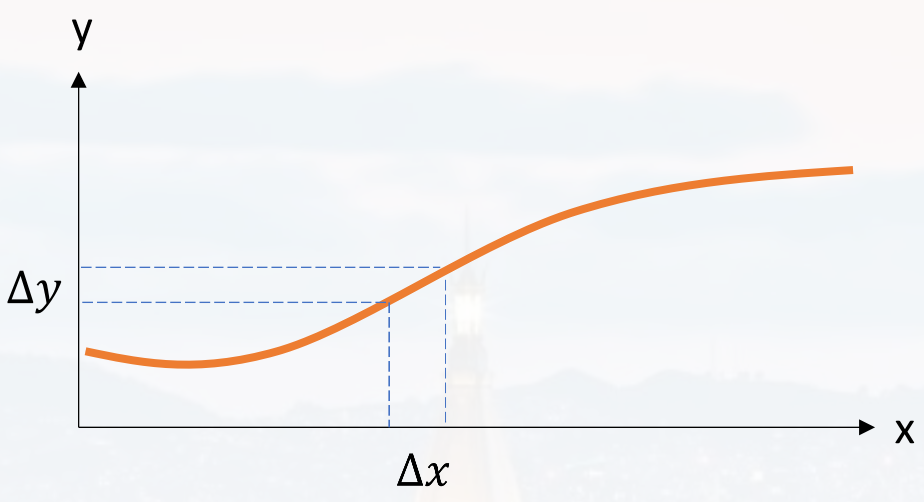
≈ **99.7%** of the data points should be consistent with the model within their **3σ** error bars

$\chi^2_{red} \approx$

1.0 excellent fit
1.0…1.5 acceptable fit
1.5…1.7 bad fit
>2.0 not acceptable

<<1.0 suspicious, errors are overestimated!

error propagation



$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x}$$

for $\Delta x \ll$ x

$$\Delta x \left|\frac{dy}{dx}\right| \approx \Delta y$$

example:

$$V = \frac{4}{3}\pi r^3$$

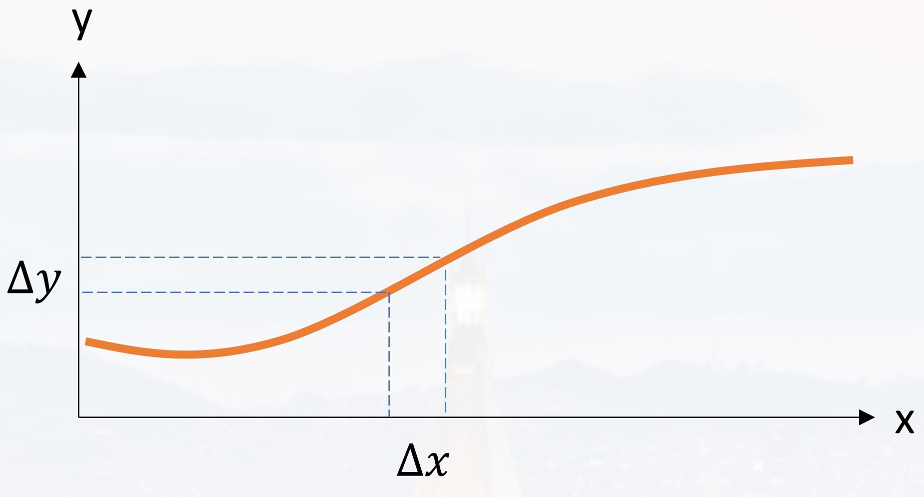
$\Delta V = ?$ given Δr

$$\Delta V = \frac{dV}{dr}\Delta r = 4\pi r^2 \Delta r$$

$$\boxed{\frac{\Delta V}{V} = 3\frac{\Delta r}{r}}$$

$\Delta r, \Delta$V ≈ 1σ

error propagation



$$\boxed{\frac{\Delta V}{V} = 3\,\frac{\Delta r}{r}}$$

$\Delta r, \Delta V \approx 1\sigma$

$$\Delta V = \frac{dV}{dr}\Delta r = 4\,\pi\,r^2\Delta r$$

radius: $r =$ 1.0µm

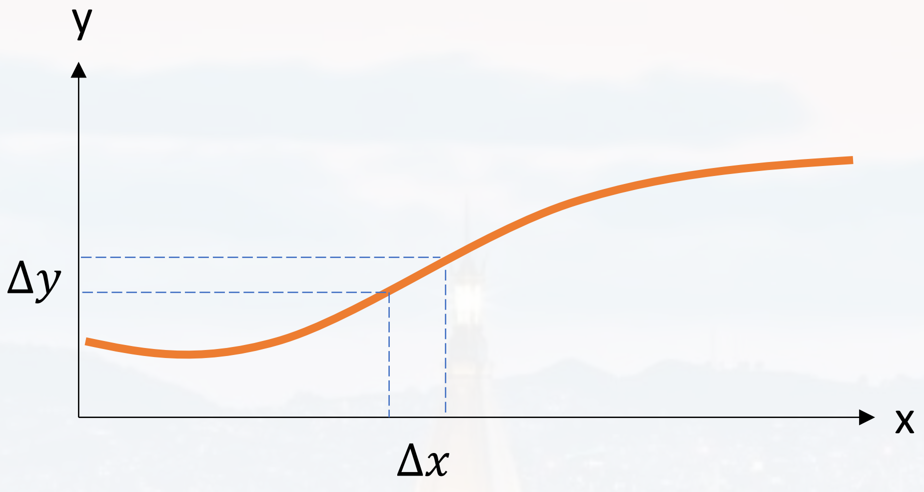
$\Delta r =$ 0.1µm

$$V = \frac{4}{3}\,\pi\,(1.0\mu\text{m})^3 = 4.18879020..\mu\text{m}^3$$

$$\Delta V = 4\,\pi\,r^2\Delta r = 4\,\pi\,(1.0\mu\text{m})^2 0.1\mu\text{m}$$

$$= 1.2566370614359172\mu\text{m}^3$$

error propagation



$$\boxed{\frac{\Delta V}{V} = 3\,\frac{\Delta r}{r}}$$

$\Delta r, \Delta V \approx 1\sigma$

$$\Delta V = \frac{dV}{dr}\Delta r = 4\,\pi\, r^2 \Delta r$$

radius: $r =$ 1.0μm
$\Delta r =$ 0.1μm

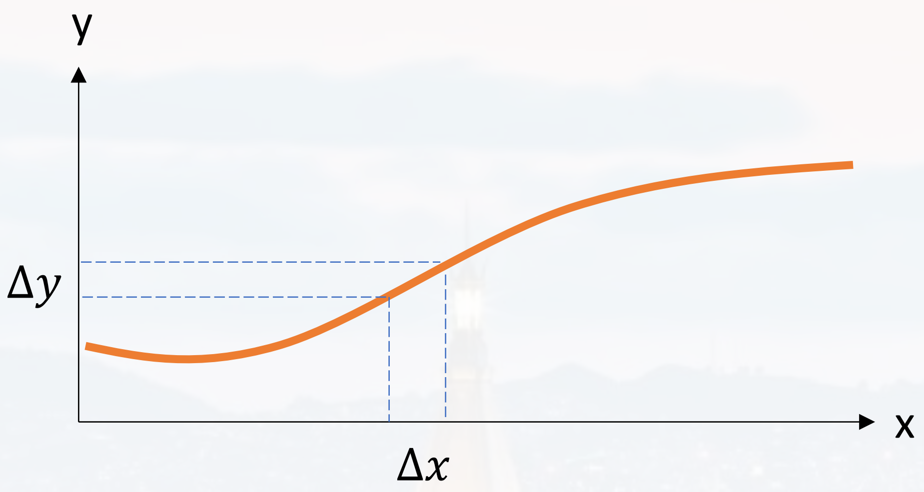
$$V = \frac{4}{3}\,\pi\,(1.0\mu\text{m})^3 = \mathbf{4.18}879020..\mu\text{m}^3$$

$$\Delta V = 4\,\pi\, r^2 \Delta r = 4\,\pi\,(1.0\mu\text{m})^2 0.1\mu\text{m}$$

$$= \mathbf{1.25}66370614359172\mu\text{m}^3$$

$$\boldsymbol{V} = (\mathbf{4.2 \pm 1.3})\mu\text{m}^3$$

error propagation



general:

$$\Delta f(max) = \sum_{i=1}^{I} \left|\frac{\partial f}{\partial x_i}\right| \Delta x_i$$ **maximum** error estimation

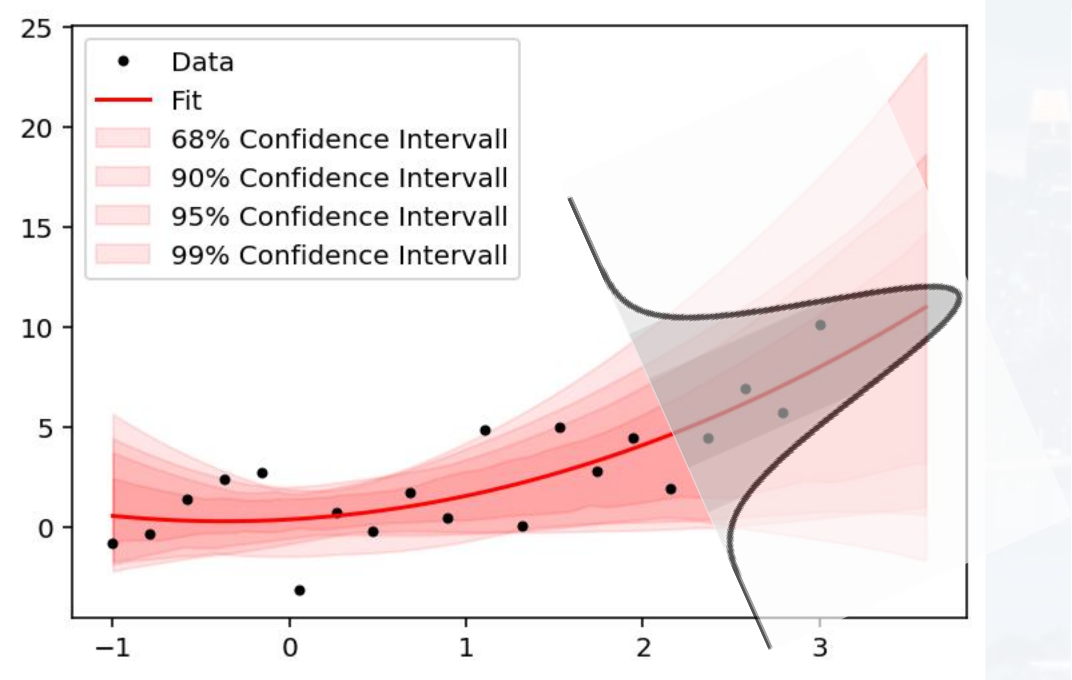
if x_i do **not correlate**, i. e. are **mutually independent**:

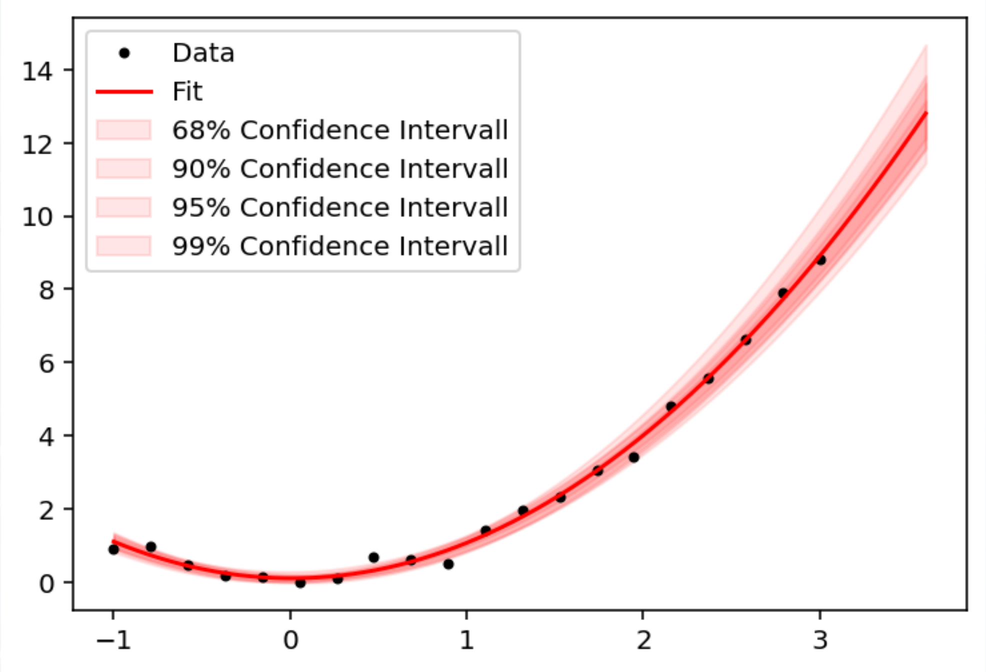
$$\Delta f^2 = \sum_{i=1}^{I} \left|\frac{\partial f}{\partial x_i}\right|^2 (\Delta x_i)^2$$

Note: $\Delta f(max)^2 > \Delta f^2$ because of the mixed terms $\left|\frac{\partial f}{\partial x_i}\right| \left|\frac{\partial f}{\partial x_j}\right| \Delta x_i \Delta x_j$ in $\Delta f(max)^2$

curve fitting (Module 11)

noisy data





Thank you very much for you attention!