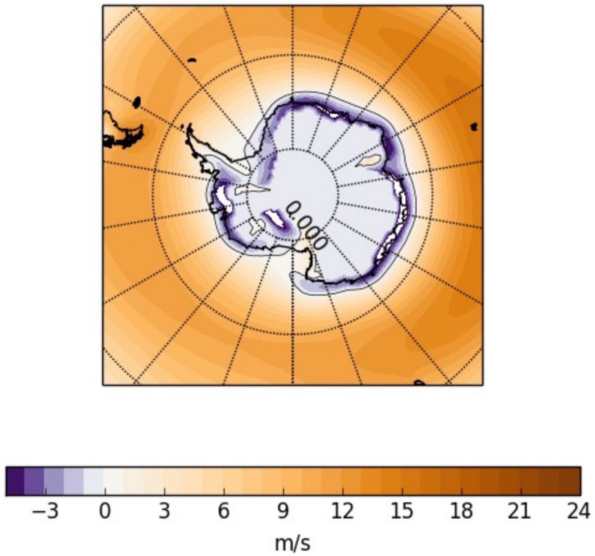
Lecture 2
Statistical Significance Tests and Linear Regression

Statistical Significance Tests

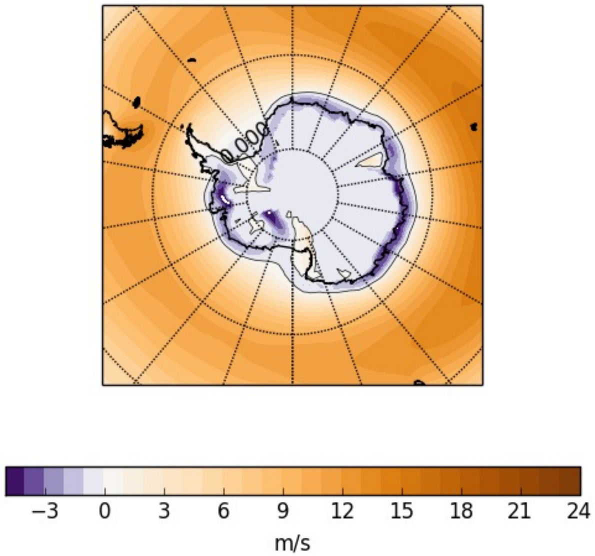
Long-term **means** of the U component of the wind at 850hPa for CTRL (left) and EXP1 (right):

CTRL



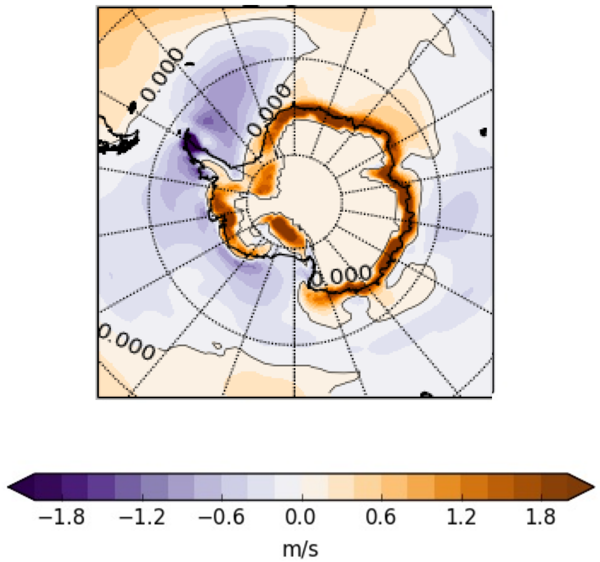
$\overline{x_a}$

EXP1



$\overline{x_b}$

EXP1-CTRL



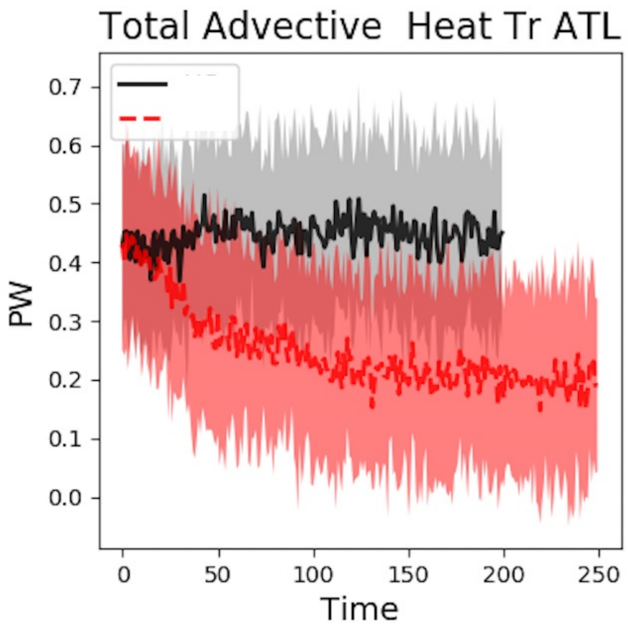
$\overline{x_b} - \overline{x_a}$

Are they different?

Are they *statistically* different?

Statistical Significance Tests

An anomaly is statistically significant if the observed value lies ***outside of the expected range of variability*** for that given variable.   To visualize this concept let's consider the following:



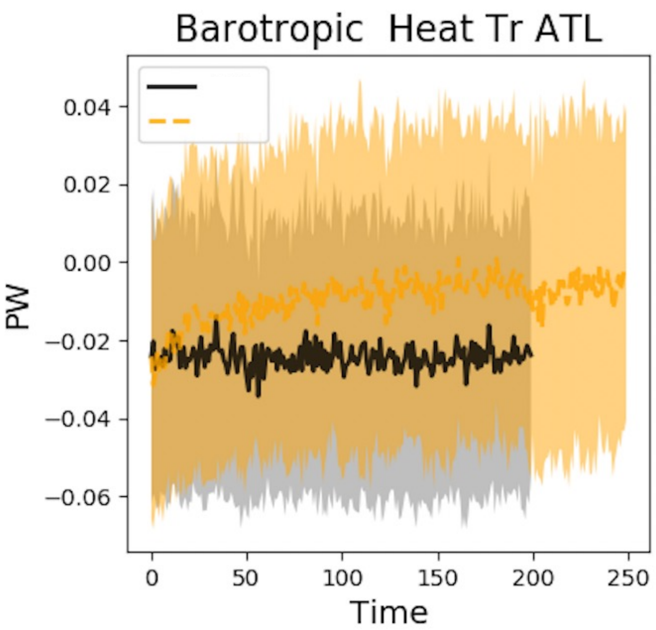
Annual mean: thick solid lines
Standard Deviation: shading

These are time-series of the Total oceanic heat transport in the North Atlantic for two numerical simulations.

The mean of EXP1 (red line) lies almost at all times outside of the CTRL (black line) range of variability (i.e. standard deviation) (gray shading).
EXP1-CTRL anomalies will certainly be statistically significant. This means that ***we can be reasonably sure that EXP1 and CTRL are different***.

Statistical Significance Tests

An anomaly is statistically significant if the observed value lies ***outside of the expected range of variability*** for that given variable. To visualize this concept let's consider the following:



These are time-series of the Barotropic oceanic heat transport in the North Atlantic for two numerical simulations.

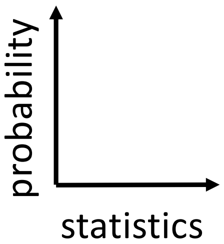
Is EXP1 (yellow line) statistically different from CTRL (black line)?
How can we be reasonably sure that EXP1 and CTRL are different?

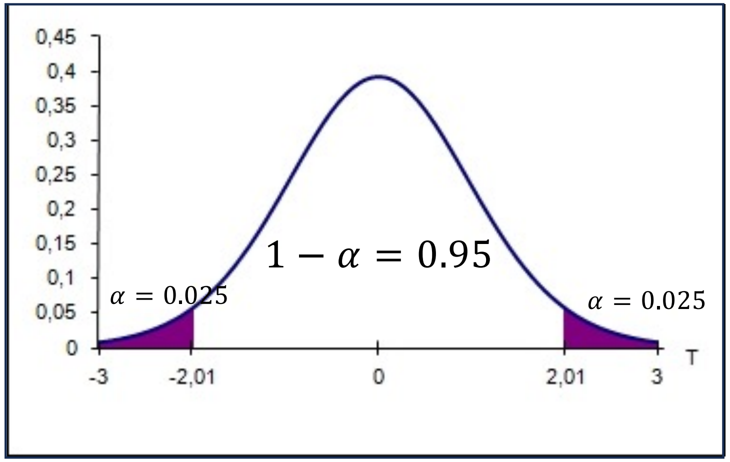
Annual mean: thick solid lines
Standard Deviation: shading

Statistical Significance Tests

We cannot prove something to be TRUE, but we can test *whether it is very unlikely to be False*.

t DISTRIBUTION:





Example t distribution for a bi-lateral test.

1. Define your Hypotheses

H_0: $\mu_a = \mu_b$ **Null Hypothesis**

H_1: $\mu_a \neq \mu_b$ **Alternative Hypothesis**

2. Set your significance level

$\alpha = 0.05$ (95% confident in the test results)

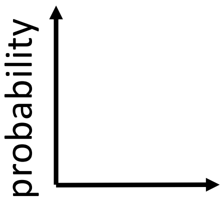
3. Compute your statistics

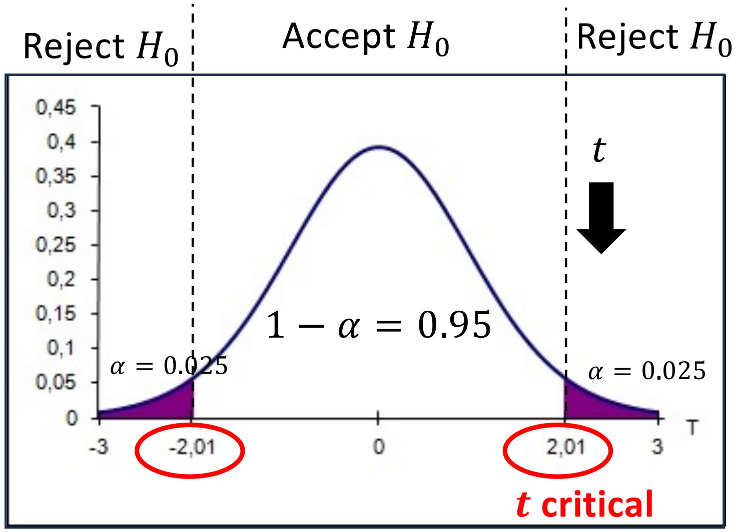
$$t_{(n_A+n_B-2)} = \frac{(\bar{x}_A - \bar{x}_B) - \overbrace{(\mu_A - \mu_B)}^{\mu_a - \mu_b = 0}}{\sqrt{s_p^2 \cdot \left(\frac{1}{n_A} + \frac{1}{n_B}\right)}}$$

Statistical Significance Tests

We cannot prove that something is TRUE, but we can test whether it is very unlikely to be False.

t DISTRIBUTION:





Example t distribution for a bi-lateral test.

4. Where in the distribution is your statistics?

$$t = 2.56$$

$$t > t_c \quad \Rightarrow \quad \begin{array}{l} \text{Reject } H_0 \\ \text{Accept } H_1 \end{array}$$

Statistically speaking the probability that H_0 is true (μ_a=μ_b) is very small! We can be reasonably confident that $\mu_a \neq \mu_b$.

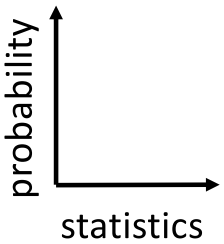
α is our margin of error! By setting alpha to 0.05 we decided that the there is a 5% chance that we will be wrong (this is the risk we are willing to take).

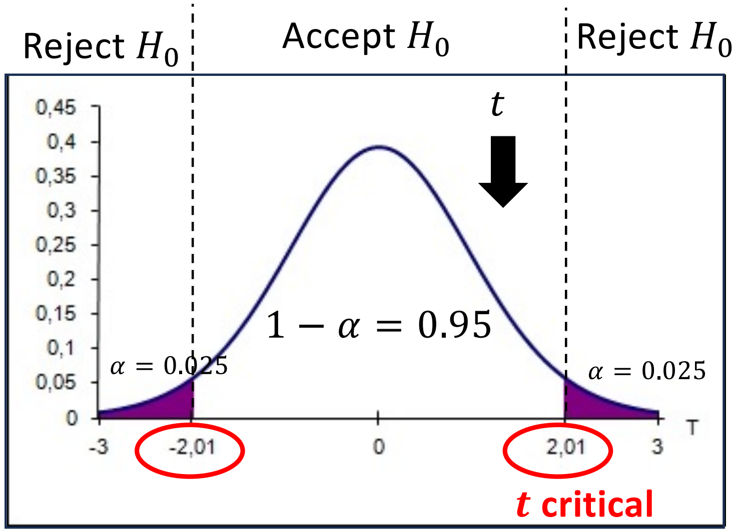
$\alpha = 0.10$
$\alpha = 0.05$ Climate science
$\alpha = 0.01$ Biological and Medical science

Statistical Significance Tests

We cannot prove that something is TRUE, but we can test whether it is very unlikely to be False.

t DISTRIBUTION:





Example t distribution for a bi-lateral test.

4. Where in the distribution is your statistics?

$$t = 1.5$$

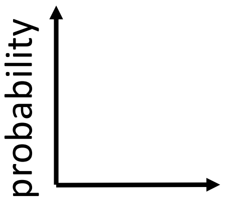
$$t < t_c \quad \Rightarrow \quad \begin{array}{l} \text{Accept } H_0 \\ \text{Reject } H_1 \end{array}$$

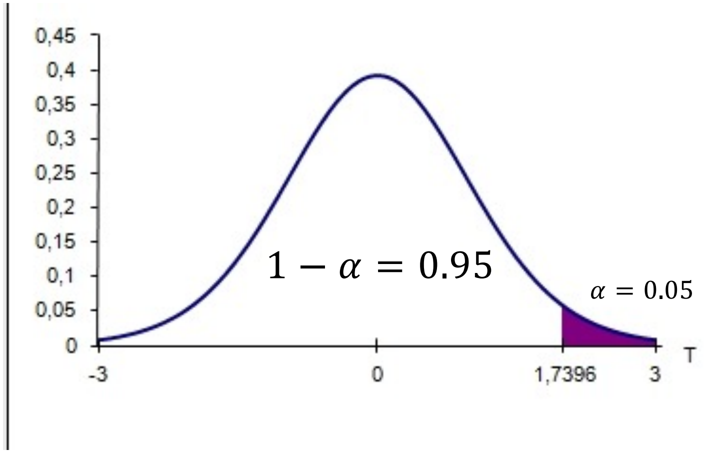
Statistically speaking the probability that H_0 is true is too high to be ignored, so we must assume that μ_a=μ_b.

Statistical Significance Tests

We cannot prove something to be TRUE, but we can test *whether it is very unlikely to be False*.

t DISTRIBUTION:





Example t distribution for a unilateral test.

1. Define your Hypotheses

H_0: $\mu_a = \mu_b$ **Null Hypothesis**

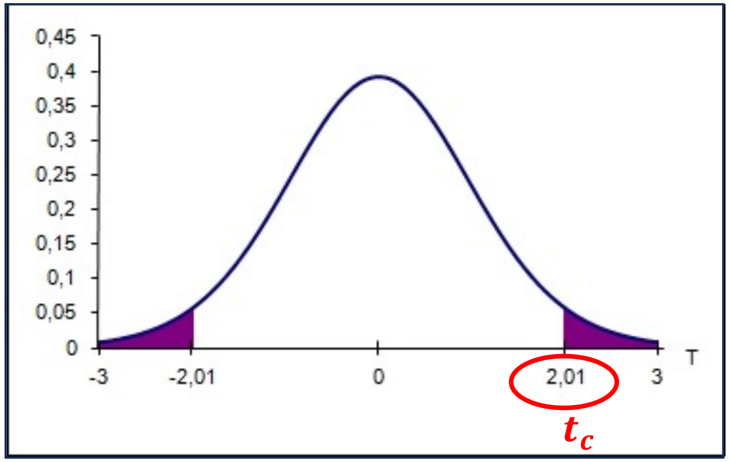
H_1: $\mu_a > \mu_b$ **Alternative Hypothesis**

$or \ \ \mu_a < \mu_b$

Statistical Significance Tests

How to know the "t critical"?

t_c is set by the degrees of freedom (ν) and α, i.e. for a given ν and α there exist a given t-distribution.



- Look-up table for t_c
- Statistical software (XLStats)
- Python package (**scipy.stats.**)

α

ν

df	0.1	0.05	0.025	0.02	0.01	0.005
1	3.078	6.314	12.706	15.895	31.821	63.657
2	1.886	2.920	4.303	4.849	6.965	9.925
3	1.638	2.353	3.182	3.482	4.541	5.841
4	1.533	2.132	2.776	2.999	3.747	4.604
5	1.476	2.015	2.571	2.757	3.365	4.032
6	1.440	1.943	2.447	2.612	3.143	3.707
7	1.415	1.895	2.365	2.517	2.998	3.499
8	1.397	1.860	2.306	2.449	2.896	3.355
9	1.383	1.833	2.262	2.398	2.821	3.250

Link to look-up table:
https://www.usu.edu/math/cfairbourn/Stat2300/t-table.pdf

Statistical Significance Tests

t −Test

The t-test can be used to test differences between:

- the sample mean and the population mean
- the sample means of independent samples [are my two simulations different?]

CONDITIONS

- Observations must be normally distributed [PARAMETRIC TEST]
- Observations must be independent
- The variances of the two datasets must be similar

The t distribution is ROBUST: it is approximately valid even if the observations are not strictly normally distributed.

Statistical Significance Tests

t –Test for two independent samples (**scipy.stats.ttest_ind**)

H_0: $\mu_A = \mu_B$

$$t_{(n_A+n_B-2)} = \frac{(\bar{x}_A - \bar{x}_B) - (\mu_A - \mu_B)}{\sqrt{s_p^2 \cdot \left(\frac{1}{n_A} + \frac{1}{n_B}\right)}}$$

$\overline{x_A}$, $\overline{x_B}$: sample means
μ_A, μ_B: population means
n_A, n_B: number of observations for sample A and B
s_p^2: pooled variance of the two samples
Degrees of freedom: $\nu = (n_A + n_B - 2)$

$$s_p^2 = \frac{\sum_{i=1}^{n_A} (x_{Ai} - \bar{x}_A)^2 + \sum_{i=1}^{n_B} (x_{Bi} - \bar{x}_B)^2}{n_A - 1 + n_B - 1}$$

Statistical Significance Tests

$$\text{scipy.stats.ttest\_ind}(\overline{x_A}, \overline{x_B}, \underbrace{\text{alternative='two-sided', equal\_var=True}}^{\text{DEFAULT}})$$

alternative= two-sided, greater, less — BILATERAL or UNILATERAL test

equal_var= True, False — Standard T-test or Welch's t-test (for unequal variances)

OUTPUT :

t_statistics, p_value, df= scipy.stats.ttest_ind($\overline{x_A}$, $\overline{x_B}$)

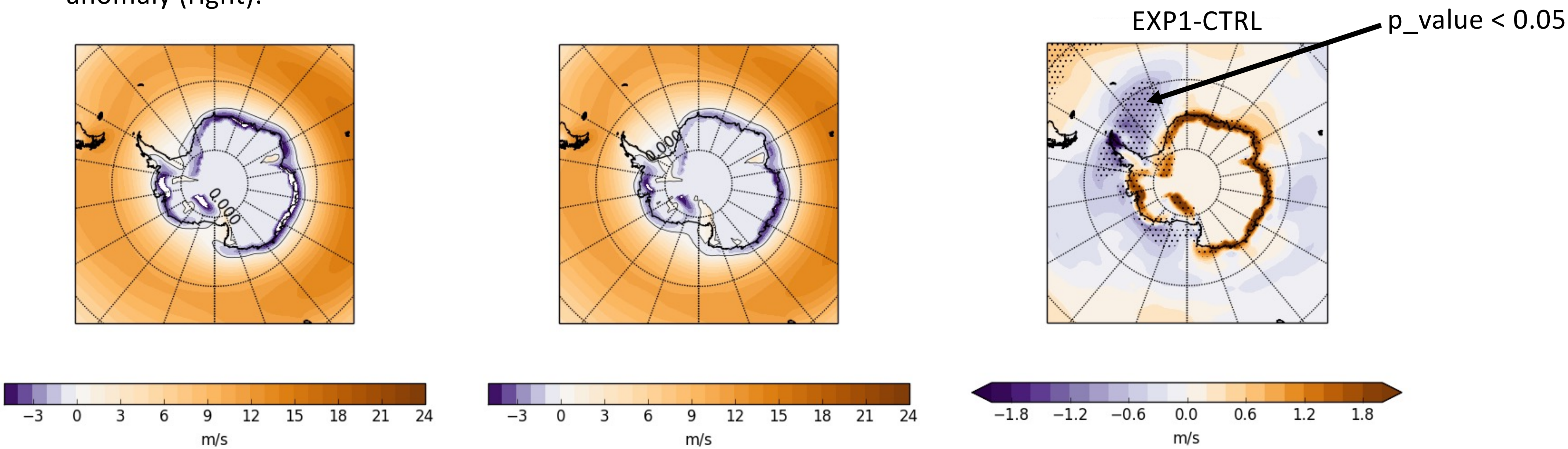
```
from scipy import stats

#use the 2-tailed Welch's t-test to test anomalies for significance
t_statistic_DJF, p_value_DJF = stats.ttest_ind(DJF_D1.data,DJF_D2.data, equal_var=False)
```

Link to python documentation: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

Statistical Significance Tests

Long-term mean of the U component of the wind at 850hPa for CTRL (left), EXP1 (center) and their anomaly (right):

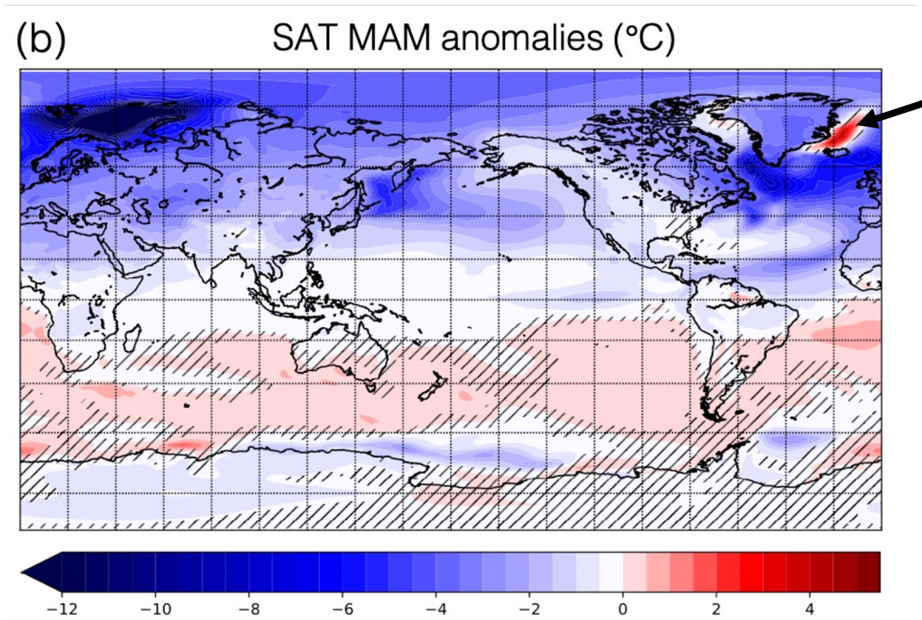


$\overline{x_a}$ $\overline{x_b}$ $\overline{x_b} - \overline{x_a}$

Are they different?

Are they *statistically* different? YES.

Statistical Significance Tests

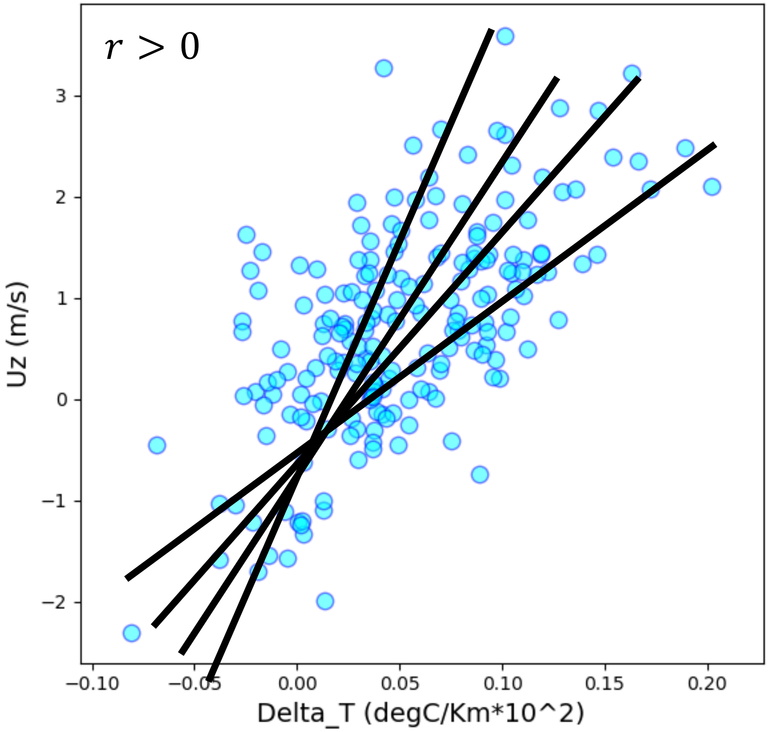


Long-term mean of surface air temperature anomalies: EXP-CTRL.
Hatched areas are non-significant at 95% confidence interval.

p_value > 0.05
Variability in the region is too high!
Although anomalies are large, they still fall within the expected range of variability for theat area.

Linear regression

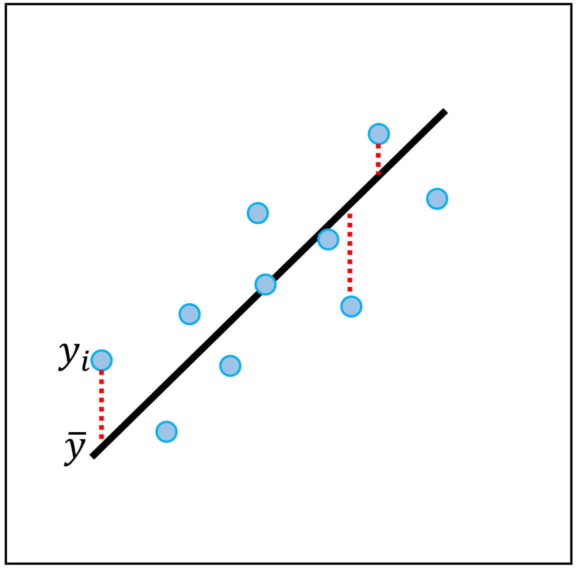
Simple linear regression is the simplest model we can use to study the linear relationship between two variables.
Question needs answering: is there a cause-and-effect relationship between x and y?



- Fit a straight-line to the data cloud: yes… but what line?

Linear regression

Simple linear regression is the simplest model we can use to study the linear relationship between two variables.
Question needs answering: is there a cause-and-effect relationship between x and y?



- Fit a straight-line to the data cloud:
  yes… but what line?

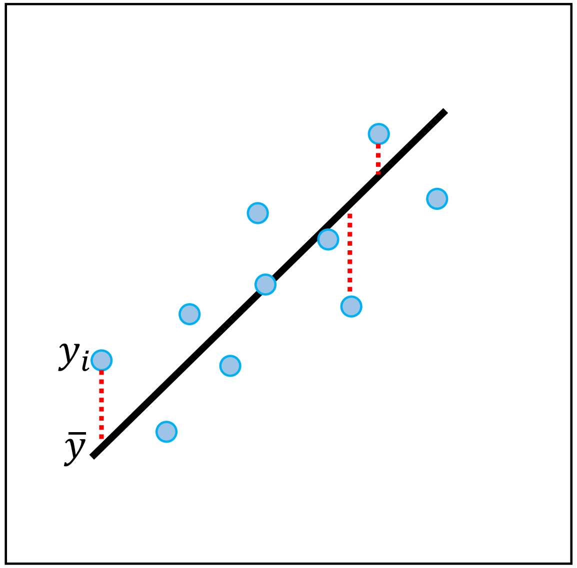
The regression line is the "best fitting" line, i.e. the one that minimizes the distance/error between the data-points and the line itself using the **least squares** method.

$$\varepsilon = \sum_{i=1}^{n} (y_i - \bar{y})^2 \qquad \varepsilon: total\ error$$

scipy.stats.linregress(*x*, *y*, *alternative='two-sided'*)

Linear regression

Simple linear regression is the simplest model we can use to study the linear relationship between two variables.
Question needs answering: is there a cause-and-effect relationship between x and y?



$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

β_0: intercept

β_1: angular coefficient (slope)

ε_i: error

t −test on the slope:
Is β_1 significantly different from zero?

scipy.stats.linregress(*x*, *y*, *alternative='two-sided'*)

Linear regression

For sample data, β_0 and β_1 (population quantities) are not known. We must use the sample intercept and slope: b_0, b_1.

t –test takes the form:

$$\boxed{t_{n-2} = \frac{b_1 - \beta_1}{s_{b1}}}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

$$b_1 = \frac{Cod_{XY}}{Dev_X}$$

$$s_{b1}^{\ 2} = \frac{s_{err.}^{\ 2}}{Dev_X} \longrightarrow s_{err.}^{\ 2} = \frac{Dev(y)*(1-r^2)}{n-2}$$

b_0: sample intercept
b_1: sample slope
$\bar{x}$: mean of x
$\bar{y}$: mean of y
Cod_{xy}: Co-deviance of x and y
Dev_x: Deviance of x
Dev_y: Deviance of y
s_{err}^2: variance of the errors
s_{b1}: standard deviation of the errors
r: Pearson correlation coefficient
n: number of observations

$\boldsymbol{H_0}$: $\boldsymbol{\beta_1} = \mathbf{0}$

$\boldsymbol{H_1}$: $\boldsymbol{\beta_1 \neq 0}$

Linear regression

$$Cod_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y})$$

$Note: \; b_1 = \Delta y / \Delta x$ Classic formula for the slope

$$Dev_x = \sum (x_i - \bar{x})^2$$

$$s_{err}^2 = \frac{1}{n-2}\left[\sum (y_i - \bar{y})^2 - \frac{\sum (x_i - \bar{x})^2 \, (y_i - \bar{y})^2}{\sum (x_i - \bar{x})^2}\right]$$

Linear regression

t –test on the slope of the regression line:

1. Define your Hypotheses

H_0: $\beta_1 = 0$

H_1: $\beta_1 \neq 0$

2. Set your significance level

$\alpha = 0.05$

3. Compute your statistics

$$\boxed{t_{n-2} = \frac{b_1 - \beta_1}{s_{b1}}}$$

4. Use a look-up table to find t_c

$t_c = X$

5. Conclude the test

$t > t_c \quad \Rightarrow$ Reject H_0, Accept H_1

$t < t_c \quad \Rightarrow$ Accept H_0, Reject H_1

Linear regression

scipy.stats.linregress(*x*, *y*)

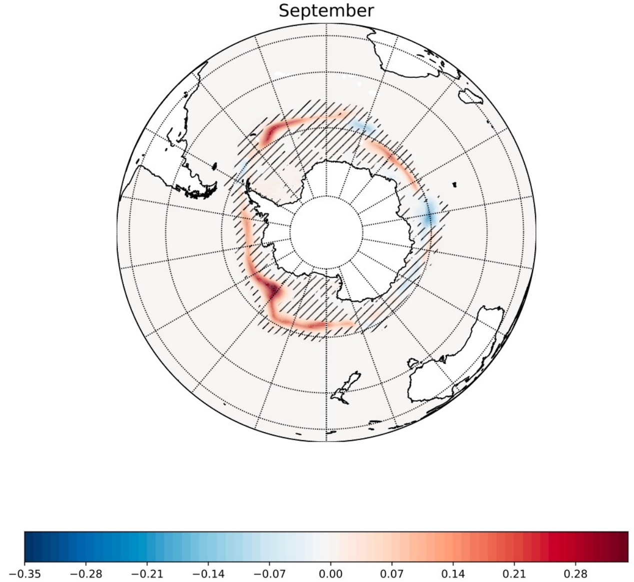
OUTPUT:

```
slope, intercept, pearson, p_value, standard_error =
scipy.stats.linregress(time, data_in)
```

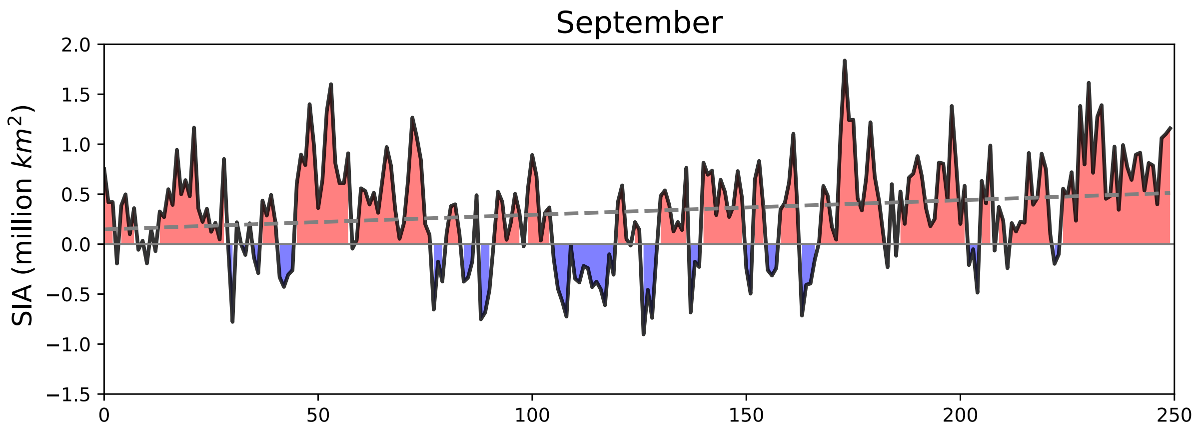
Linear regression

Linear regression in climate science is very commonly used to study temporal trends

`scipy.stats.ttest_ind(`$\overline{x_A}, \overline{x_B}$`)`



`scipy.stats.linregress(time, anomalies)`

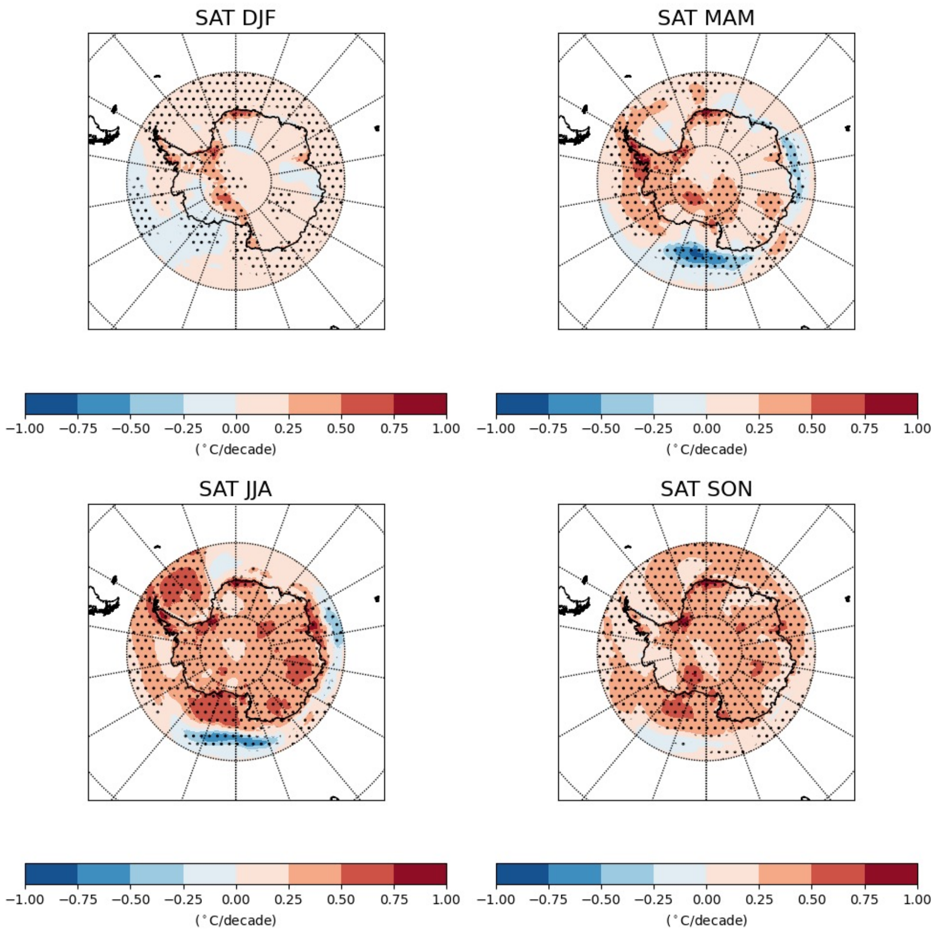


Right: September EXP-CTRL sea ice concentration anomalies. Non-hatched areas correspond to statistically significant differences (at 95 % confidence).

Left: Time series of September EXP-CTRL sea ice area from year 0 to 250 of simulation. Dashed grey lines represent best fit for data. The trend is positive and statistically significant (p value < 0.05).

Linear regression

Linear regression in climate science is very commonly used to study temporal trends




```
scipy.stats.linregress(time, SAT)
```

Seasonal maps for Surface Air Temperature (SAT) trends (°C/decade) below 60°S from ERA5 reanalysis from 1950 to 2022. Dots indicate statistical significance (95% confidence).

Exercises

1. Using the same input time-series of Exercise 1 (`ERA5_2m_SAT_TS_1990_2023.txt`), use the built-in python function `stats.linregress` to fit a straight-line to the time-series and plot the regression line to your line plot of Exercise 1 (see example provided).

2. Do you see a trend? Test the trend for statistical significance:
   carry out a $\boldsymbol{t}$ −test on the slope of the linear regression line using the equations provided at page 18-19 of Lecture 2. This means: i) compute b_1 , sb_1 and t_{n-2}; ii) find t_c using the look-up table* provided, and iii) setting $\alpha = 0.05$ determine whether the slope is significantly different from zero.
   To do the above you will have to use the functions you created in Exercise 1. Tip: the function for the standard deviation can be edited to compute DEV(x,y) and COD(x,y).

*for df > 100 take t_c value at df=100

3. Compare your results with the outputs from the python function `stats.linregress`: `slope` and `std_err`. Note: `slope=b1, std_err=sb1.`
   Based on your t_score, is the result of your test in agreement with python's p_value?

3. Using the provided function (`compute_annual`) compute a time-series of the annual means from the monthly means provided (i.e. `ERA5_2m_SAT_TS_1990_2023.txt`). Repeat now steps 1 to 3 on the averaged dataset: plot the annual mean time-series, fit a line and plot the regression line, test the slope for significance. Has the result changed?

4. So, is the increase in surface air temperature in Trieste from 1990 statistically significant?

Exercises

Example script: **DA_exercise_2.py**

```
############### Load datasets #############
#Load txt file and call functions
data_in=np.loadtxt('ERA5_2m_SAT_TS_1990_2023.txt', usecols=2)


mean=compute_mean(data_in)
stdev=compute_stdev(data_in, mean)


#Plot data with regression line and print out 'slope, std_err, p_value'
plot_tseries(data_in, mean, stdev, label='ERA5', color='black',
linestyle='-', title='2m air Temperature @ Trieste 1990-2023')


#call function to perform t-test
slope, std_err, t_score=t_student(data_in)
plt.show()
```

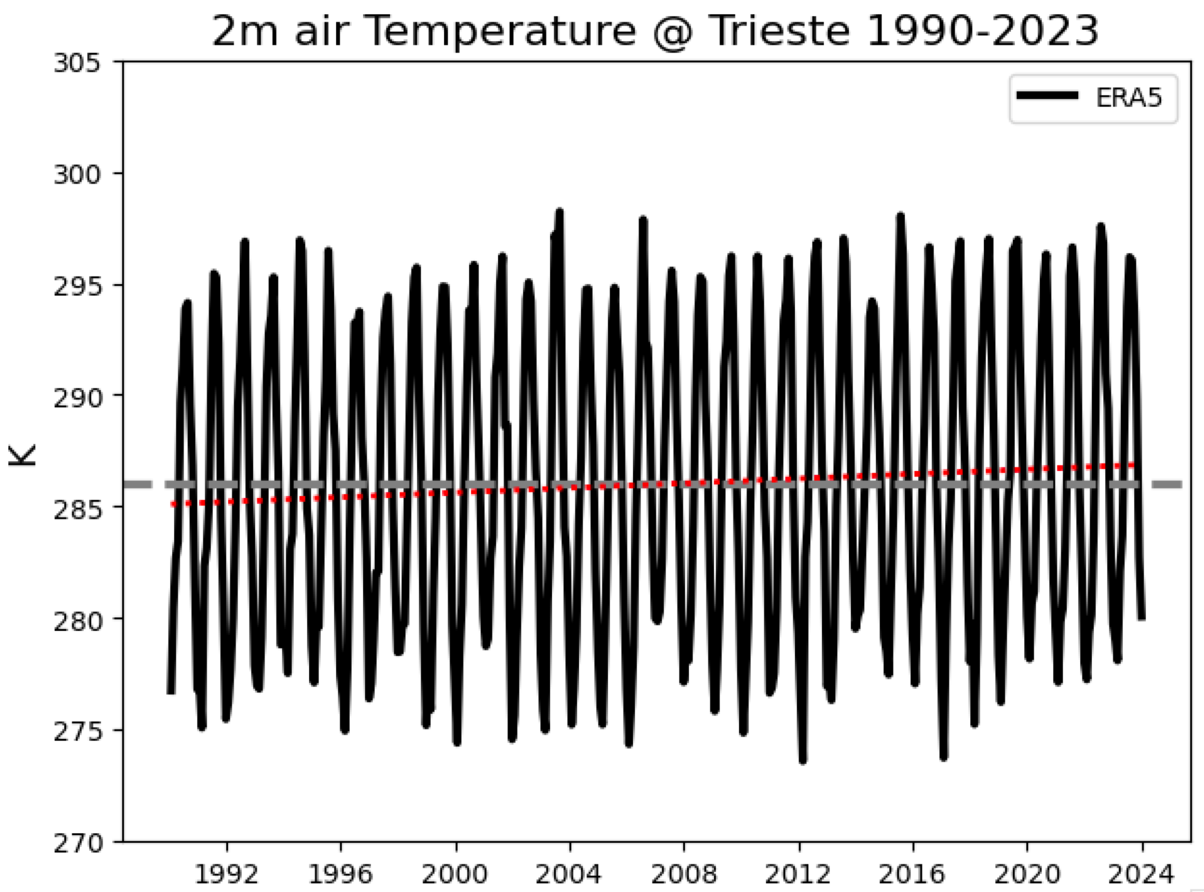
Exercises

```
def plot_tseries(data_in, mean, stdev, label, linestyle, color, title):
```

```
 slope, intercept, r_value, p_value, std_err=stats.linregress(times,  data_in)
 line_fit=slope*times +intercept
 plt.plot(date, line_fit, c='red', linestyle=':',linewidth=2)
 #Print p values and etc.
 print ('P_VALUE', p_value)
 print ('SLOPE', slope)
 print ('STD_ERR', std_err)
 print ('     ')
```

Exercises

`plot_tseries()`



Exercises

```
######## Compute slope and t-statistics  for input dataset #########
def t_student(data_in):


 #our 'x' is time, let's define the x axis accordingly
 time=np.arange(0,len(data_in),1)
 #compute time mean
 mean_t=compute_mean(time)
 #compute data_in mean
 mean_data=compute_mean(data_in)


 #compute Cod_xy
 Cod_xy=..


 #Now compute Dev_x
 Dev_x=..


 #call a function to compute the error variance (s_err^2)
 df=len(data_in*2)-2 #define degrees of freedom as (n_x+n_y-2)
 err_var=my_function(...)
```

Exercises

Continued…

```
 #Compute now the slope (b1)
 b1=...
 #and the standard deviation of the residual of the slope (s_b1) -also known as
standard error
 s_b1=...

 #Finally, compue the t-statistics: t=b1-beta_1/s_b1. Note: beta_1=0 because of our
null hypothesis
 t_score=b1/s_b1




 print ('SLOPE my_function' , b1)
 print ('STD_ERR SLOPE my_function' , s_b1)
 print ('t-score my_function' , t_score)
 print ('        ')


 return(b1, s_b1, t_score)
```

Exercises

```
#Compute annual means
data_in=compute_annual(data_in)
mean=compute_mean(data_in)
stdev=compute_stdev(data_in, mean)

plot_tseries(data_in, mean, stdev, label='ERA5', color='black',
linestyle='-', title='Annual 2m air Temperature @ Trieste 1990-2023')

slope, std_err, t_score=t_student(data_in)
plt.show()
```

```
def compute_annual(data):


 #compute annual means:
 months=len(data)
 years=[]
 for i in range (12, months+1, 12):
  st=i-12
  years.append(np.average(data[st:i]))


 #print (len(years))
 return(years)
```