

# Εργαστήριο Στοχαστικών Ανελίξεων

## Οδηγίες Εγκατάστασης Λογισμικού

### Contents

Περίληψη . . . . .	1
Εγκατάσταση . . . . .	1
Προτεινόμενη μέθοδος . . . . .	1
Εναλλακτικές μέθοδοι . . . . .	2
Τρόπος Λειτουργίας Εργαστηρίου . . . . .	2
Python . . . . .	3
Markdown . . . . .	3
Jupyter Notebook . . . . .	3
Παράδοση Εργασιών . . . . .	6

### Περίληψη

Αυτές οι οδηγίες αφορούν το μάθημα “Στοχαστικές Ανελίξεις” των σχολών ΣΕΜΦΕ και ΣΗΜΜΥ του ΕΜΠ. Ο διδάσκων καθηγητής είναι ο [Δρ. Μιχαήλ Λουλάκης](#). Σε αυτές τις οδηγίες περιγράφεται το λογισμικό που πρέπει να εγκαταστήσετε ώστε να δουλέψετε τις ασκήσεις.

Οι ασκήσεις του μαθήματος βρίσκονται [εδώ](#). Για να αναφέρεται απορίες ή προβλήματα σχετικά με την εγκατάσταση και χρήση του λογισμικού χρησιμοποιείτε αυτό το [forum](#). Για να ανοίξετε καινούργια συζήτηση χρειάζεται να έχετε Github-account. Για απορίες σχετικά με τις ασκήσεις απευθυνθείτε στον διδάσκοντα.

Οι ασκήσεις είναι γραμμένες σε μορφή *Jupyter Notebook*. Το *Jupyter Notebook* είναι μια εφαρμογή που επιτρέπει τον συνδυασμό κειμένου και κώδικα στο ίδιο αρχείο έτσι ώστε η παρουσίαση των προβλημάτων και των λύσεων να είναι πλήρης και αυτόκλητη. Για να χρησιμοποιήσετε την εφαρμογή και για να δουλέψετε τις ασκήσεις απαιτούνται τα παρακάτω λογισμικά.

- Γλώσσα προγραμματισμού **Python 3**
- Οι βιβλιοθήκες **Numpy** και **Matplotlib**
- Η εφαρμογή **Jupyter Notebook**

Για την εγκατάσταση και ολοκλήρωση των ασκήσεων χρειάζονται μερικές επιπλέον βιβλιοθήκες αλλά δεν θα σας ζητηθεί να αλληλεπιδράσετε με αυτές.

### Εγκατάσταση

#### Προτεινόμενη μέθοδος

Ο προτεινόμενος και ευκολότερος τρόπος να εγκαταστήσετε τα πάντα είναι μέσω της [Anaconda](#). Η Anaconda είναι μια ειδική έκδοση της Python προσαρμοσμένη για χρήστες που κάνουν ανάλυση δεδομένων (data scientists). Για να εγκαταστήσετε την Anaconda, πρέπει να ανοίξετε αυτό το [link](#) και να επιλέξετε τον installer της **Python 3** (όχι 2!) που αντιστοιχεί στο λειτουργικό σύστημα σας (Windows, Mac, Linux).

## Εναλλακτικές μέθοδοι

Εναλλακτικά αν δεν θέλετε να εγκαταστήσετε την Anaconda, πρέπει να:

1. Εγκαταστήσετε την [Python 3](#)
2. Εγκαταστήσετε τις βιβλιοθήκες που αναφέρονται [εδώ](#)

Αν το λειτουργικό σας σύστημα είναι OSX ή GNU/Linux τότε ο ευκολότερος τρόπος να τα εγκαταστήσετε όλα είναι μέσω του package manager που σας παρέχει. Για Mac επειδή το OSX δεν έχει προεγκατεστημένο package manager, συνιστούμε τη χρήση του [HomeBrew](#).

Τέλος αν δεν μπορείτε να χρησιμοποιήσετε τον package manager του συστήματός σας (επειδή δεν υπάρχει ή δεν είστε διαχειριστής), μπορείτε να χρησιμοποιήσετε το [pip](#) που είναι ο package manager της Python. Για να εγκαταστήσετε το pip ακολουθείστε τις οδηγίες [εδώ](#). Προφανώς για να χρησιμοποιήσετε το pip χρειάζεται να έχετε ήδη εγκαταστήσει (με κάποιο τρόπο) την Python 3. Οι παρακάτω εντολές εγκαθιστούν όλα τα πακέτα που χρειάζονται:

```
# if you only have Python3 then you can write just python
python3 -m pip install -r requirements.txt
# or if you don't have admin rights
python3 -m pip install --user -r requirements.txt
```

## Τρόπος Λειτουργίας Εργαστηρίου

Το εργαστήριο είναι εικονικό. Όλες οι ασκήσεις βρίσκονται [εδώ](#), για να τις κατεβάσετε πατήστε το Clone or Download (Fig 1) (δεξιά πάνω από τη λίστα με αρχεία) και επιλέξτε Download ZIP (Fig 2).



Figure 1: Clone or Download

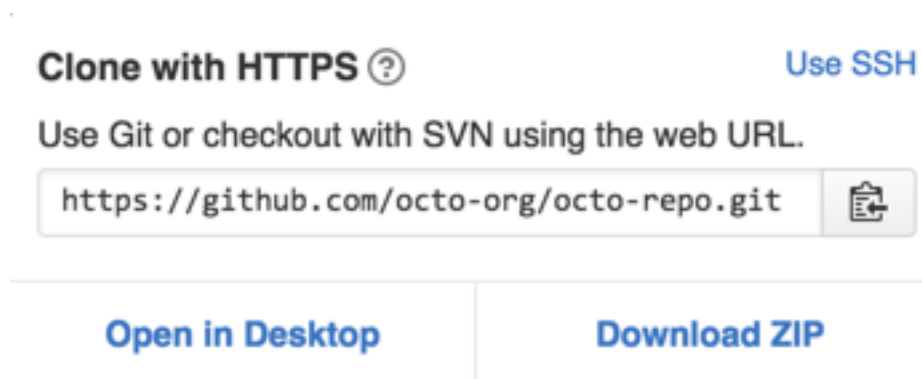


Figure 2: Download ZIP

Για κάθε άσκηση, θα σας δίνεται ένα *Jupyter Notebook* (.ipynb αρχείο) που θα πρέπει να συμπληρώσετε. Μέσα στο notebook, θα περιγράφεται κάποιο πρόβλημα και (συνήθως) θα υπάρχει κάποιο δείγμα κώδικα. Σκοπός σας είναι να χρησιμοποιήσετε τις δυνατότητες του notebook ώστε να τροποποιήσετε τον κώδικα και να απαντήσετε στις ερωτήσεις που σας δίνονται.

Στο υπόλοιπο αυτών των οδηγιών θα σας περιγράψουμε συνοπτικά την φιλοσοφία αυτού του εργαλείου, θα σας εξηγήσουμε τον τρόπο με τον οποίο πρέπει να απαντάτε τις ερωτήσεις. Τέλος θα σας υποδείξουμε κάποιες πηγές από τις οποίες μπορείτε να μάθετε τις τεχνικές δεξιότητες που απαιτούνται για να απαντήσετε.

---

## Python

Η γλώσσα που θα χρησιμοποιήσουμε για αυτό το εργαστήριο είναι η [Python 3](#). Η Python είναι μια πάρα πολύ δυνατή γλώσσα με μεγάλη αποδοχή από την επιστημονική κοινότητα. Για αυτό το εργαστήριο εκτός από τις βασικές δυνατότητες της Python θα χρειαστεί να χρησιμοποιήσετε και 2 από τις πιο δημοφιλείς επιστημονικές βιβλιοθήκες της:

1. [Numpy](#) για αριθμητικό προγραμματισμός
2. [Matplotlib](#) για την οπτικοποίηση των αποτελεσμάτων.

Αν δεν γνωρίζετε Python (ή κάποια από τις 2 βιβλιοθήκες) δεν χρειάζεται να απογοητεύεστε καθώς η Python έχει σχεδιαστεί με σκοπό να είναι ευανάγνωστη και εύκολη στην εκμάθηση. Για να μάθετε γρήγορα τα βασικά μπορείτε να παρακολουθήσετε τα 2 διαδικτυακά μαθήματα που παρέχονται δωρεάν από το [DataCamp](#)

1. [Intro to Python](#)
2. [Intermediate Python](#)

Η Python είναι μια *interpreted language* οπότε μπορείτε εύκολα να πειραματιστείτε. Αν θέλετε ένα διαδραστικό τερματικό για να δοκιμάσετε τις ιδέες σας, σας συνιστούμε το [IPython](#). Έχετε υπόψη ότι για την παράδοση των ασκήσεων πρέπει να γράψετε κώδικα Python σαν μέρος ενός Jupyter Notebook.

## Markdown

Οι ασκήσεις εκτός από κώδικα απαιτούν να απαντήσετε και μερικές ερωτήσεις κρίσεως. Γι αυτό το λόγο (και για να εξηγήσετε καλύτερα τις σκέψεις και την υλοποίηση τους) χρειάζεται να γράψετε ελεύθερο κείμενο. Για να διαμορφώσετε το κείμενο σας σωστά, (με τίτλους, στοιχειά έμφασης και λίστες), θα πρέπει να χρησιμοποιήσετε την σύνταξη **markdown**.

Το **markdown** είναι μια σύνταξη που σχεδιάστηκε με σκοπό να είναι εύκολη στην εκμάθηση, και εύκολα μεταφράσιμη σε άλλες (πιο πλούσιες) μορφές κειμένου. Από όλα όσα χρειάζεται να μάθετε για αυτό το εργαστήριο το markdown απαιτεί την λιγότερη πνευματική άσκηση!

Μία σύντομη περιγραφή όλων των δυνατοτήτων που θα χρειαστείτε για το εργαστήριο δίνεται σε αυτό τον σύνδεσμο: <https://goo.gl/PTQGn6> (παρατηρήστε ότι υπάρχει επιπλέον σύνδεσμος που οδηγεί στο βίντεο που περιγράφει το κείμενο). Για επιπλέον παραδείγματα μπορείτε να δείτε [εδώ](#) και [εδώ](#).

## Jupyter Notebook

Σε αυτό τον [σύνδεσμο](#) μπορείτε να δείτε ένα εισαγωγικό βίντεο για τη χρήση του notebook.

Το *Jupyter Notebook* είναι μια εφαρμογή που επιτρέπει την δημιουργία και διανομή αρχείων που περιέχουν κώδικα, εξισώσεις, εικόνες μαζί με το κείμενο που τα εξηγεί. Είναι οργανωμένο σε “κελιά” (cells). Κάθε κελί μπορεί να είναι είτε κελί κειμένου, στο οποίο μπορείτε να εξηγήσετε τι κάνετε, όπως και “γιατί” ή “πώς” το κάνετε. Στα κελιά αυτά γράφετε σύμφωνα με την σύνταξη **markdown** (βλέπε παρακάτω για λεπτομέρειες). Στα κελιά κώδικα, γράφετε κώδικα **Python** (βλέπε παρακάτω). Όταν “εκτελέσετε” το κελί ο κώδικας αυτός εκτελείται και τα αποτελέσματα του, αν είναι εμφανή, τυπώνονται ακριβώς από κάτω. Ότι υπολογίζεται σε ένα κελί είναι διαθέσιμο σε όλα τα μετέπειτα κελιά (το namespace είναι ενιαίο για όλα τα κελιά).

Για να ξεκινήσετε να δουλεύετε τις ασκήσεις πρέπει να ανοίξετε το notebook μέσα στο φάκελο που περιέχει τα `.ipynb` αρχεία. Υπάρχουν πολλοί τρόποι να το κάνετε αυτό, όπως για παράδειγμα:

- Με χρήση **terminal**:

1. Ανοίχτε το terminal (για Windows που έχουν εγκαταστήσει την Anaconda, ανοίχτε το *Anaconda prompt*)
2. Πλοηγηθείτε στο φάκελο που έχει τα notebooks (`cd path/to/folder`)
3. Δώστε την εντολή `jupyter notebook`

- Με χρήση **Anaconda Navigator** (Windows και OSX).

1. Τρέχτε το Anaconda Navigator
2. επιλέξτε την εφαρμογή *Jupyter Notebook*
3. Στο παράθυρο του browser που ανοίγει πλοηγηθείτε στο φάκελο με τα notebooks

- Με **Windows Shortcut** (μόνο για Windows με Anaconda):

1. Βρείτε την εφαρμογή *Jupyter Notebook* στη γραμμή εκκίνησης
2. Αντιγράψτε την στην επιφάνεια εργασίας
3. Κάντε δεξί-κλικ στο νέο εικονίδιο και συμπληρώστε στο πεδίο *Start in* το μονοπάτι για το φάκελο που περιέχει τα notebooks.

Όταν ξεκινήσετε το notebook ένα καινούργιο παράθυρο ή καρτέλα (tab) θα ανοίξει στον φυλλομετρητή (browser) της επιλογής σας (Fig 3). Αν δεν συμβεί αυτό ή κατά λάθος κλείσετε το παράθυρο τότε μπορεί να το ξαναβρείτε πηγαίνοντας στις διεύθυνση: <http://localhost:8888/>.

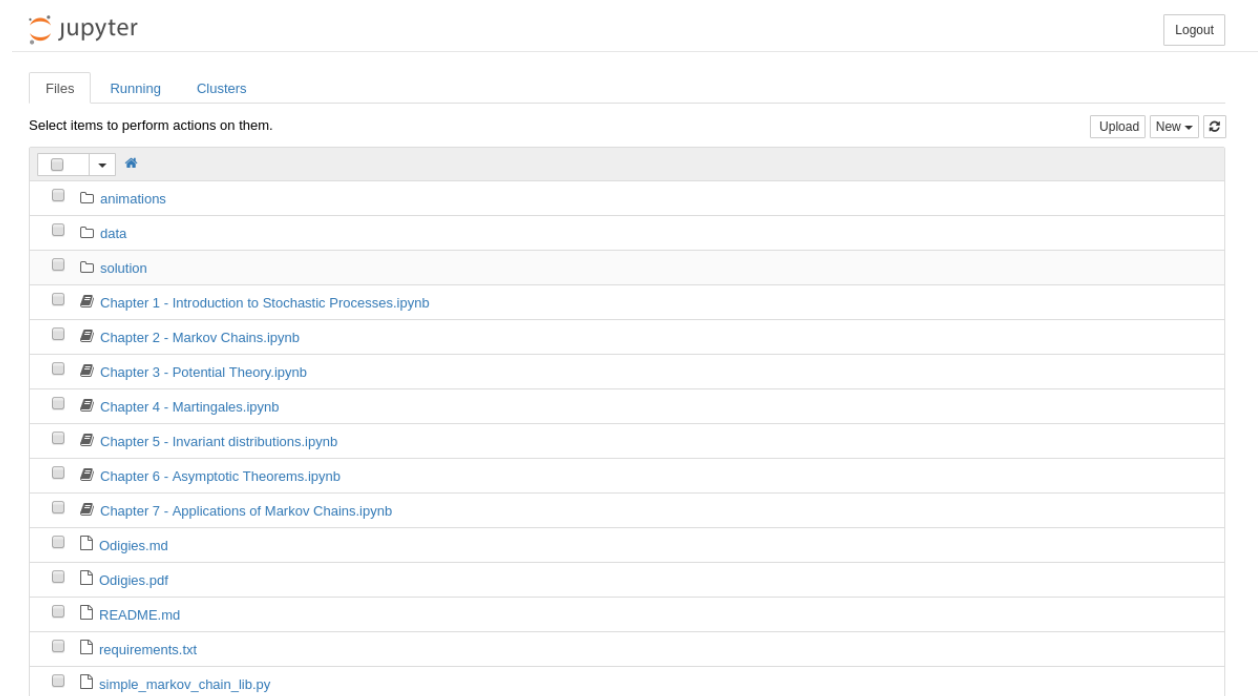





Figure 3: Κεντρική Σελίδα

Άμα ακολουθήσετε σωστά τις οδηγίες θα πρέπει να δείτε μία λίστα με τα διαθέσιμα αρχεία. Κάνοντας διπλό-κλικ στο αρχείο της άσκησης που θέλετε να δουλέψετε, μια καινούργια καρτέλα με την περιγραφή της άσκησης θα εμφανιστεί και εκεί μπορείτε να δουλέψετε (Fig 4).


**Jupyter** Chapter 1 - Introduction to Stochastic Processes (unsaved changes)
  Logout

File Edit View Insert Cell Kernel Help
 Python 3



```

In [1]: import numpy as np
        np.random.seed(2016) # for reproducibility
        np.set_printoptions(precision = 3)
    
```

### Exercise 16-17

For this exercise you should download the `simple_markov_chain_lib.py` which is available [here](#). It's not necessary to understand it in all detail, since this would require some experience in Python (or Object-Oriented Programming). We will just use it as a "library" that simulates a markov chain.

However, if you have some familiarity with Python, it's worth while taking a look at the "internal methods" `_partial_sums` and `_next_state` (the `_` designates an "internal" function not to be used by the end-user). These, implement the random sampler described at **exercise 10**.

#### simple\_markov\_chain\_lib

`simple_markov_chain_lib.py` contains only the `markov_chain` "object".

```

In [2]: from simple_markov_chain_lib import markov_chain
    
```

In this section we will see how you can interact with this object. In particular we will focus on 3 things:

1. How to initiate a markov chain
2. How to simulate a random walk across the state-space
3. How to investigate the attributes of this markov chain

#### Initiate a new markov chain

To initiate a new markov chain you have to provide 2 arguments:

1. `markov_table`: the matrix of the transition probabilities and
2. `init_dist` (optional): the initial distribution

Because transition matrices are sparse most of the time, `markov_table` and `init_dist` are not actual matrices and vectors but rather dictionaries.

Each dictionary describes a probability mass function by assigning a probability to the corresponding key. Thus the format for `init_dist` is

- `key` = state
- `value` = probability of starting from the corresponding state

The `markov_table` format is a little trickier. Each entry of the table corresponds to a row of the transition table. Thus each key is a state as before but now values are distributions themselves, ie dictionaries with states as keys and probabilities as values.

Figure 4: Σελίδα Άσκησης

## Παράδοση Εργασιών

Όταν είστε ικανοποιημένοι με το αποτέλεσμα της δουλειάς σας, τότε πρέπει:

1. Να τρέξετε ξανά όλο το notebook από την αρχή για να σιγουρευτείτε ότι δουλεύει (Kernel > Restart & Run all)
2. Να σώσετε το notebook σαν αρχείο html (File > Download as... > HTML)
3. Να στείλετε το html αρχείο στον [διδάσκοντα](#)

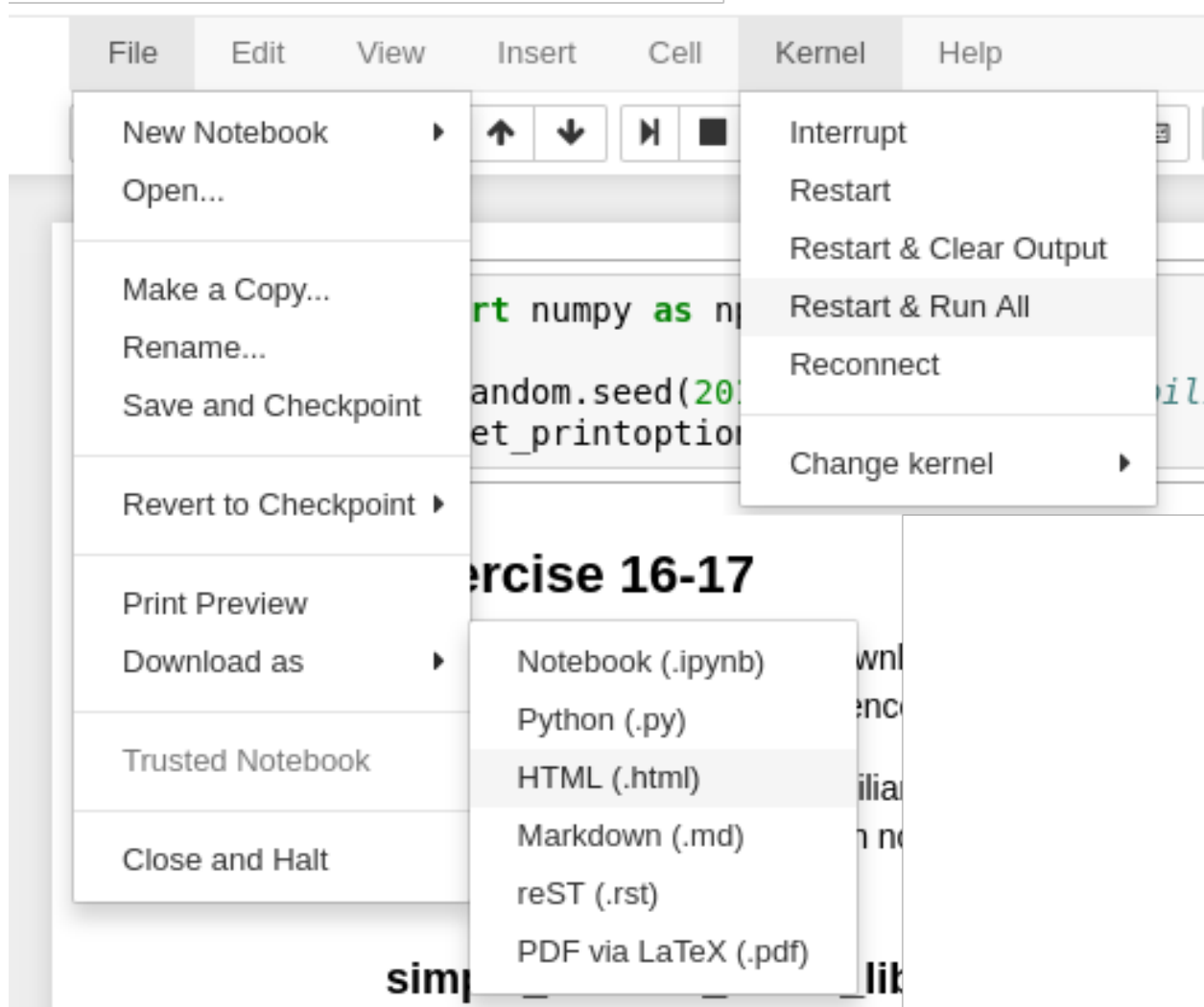


Figure 5: Κατάθεση άσκησης