ANALYSIS OF ENGINEERING & SCIENTIFIC DATA

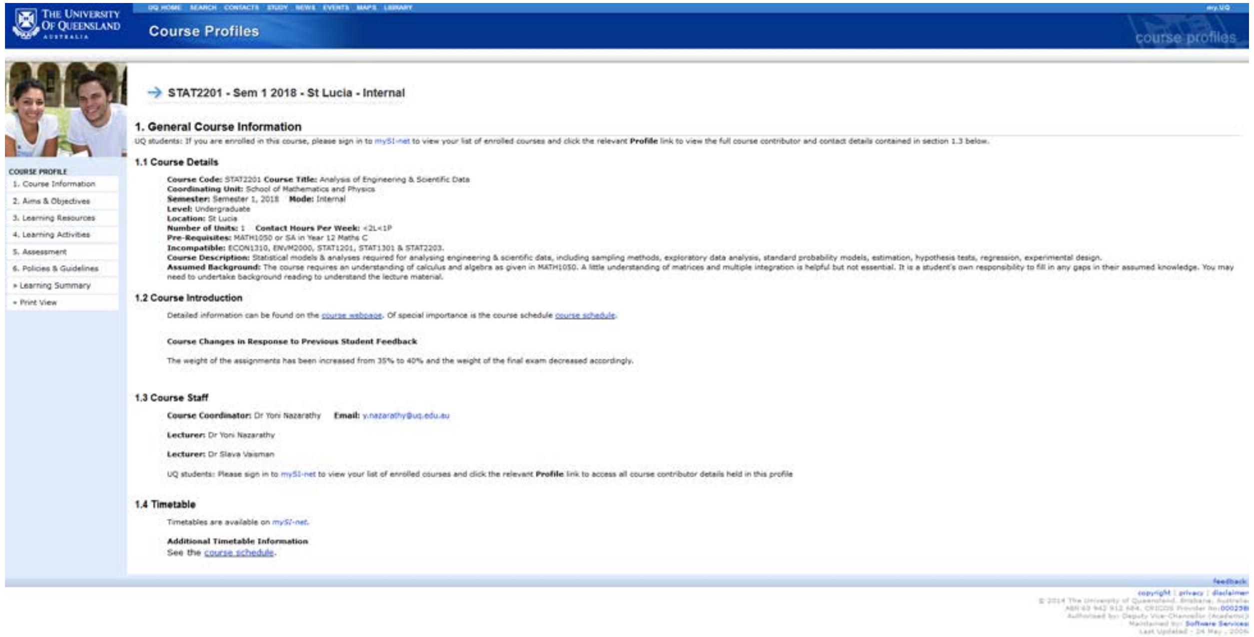
STAT2201

Slava Vaisman

THE UNIVERSITY OF QUEENSLAND

ADMINISTRATION

- This course is for engineering (civil, mechanical, software…)
- Electronic Course Profile http://www.courses.uq.edu.au/student_section_loader.php?section=1&profileId=92234



THE UNIVERSITY OF QUEENSLAND AUSTRALIA

UQ HOME SEARCH CONTACTS STUDY NEWS EVENTS MAPS LIBRARY my.UQ

Course Profiles

course profiles

COURSE PROFILE
1. Course Information
2. Aims & Objectives
3. Learning Resources
4. Learning Activities
5. Assessment
6. Policies & Guidelines
» Learning Summary
» Print View

STAT2201 - Sem 1 2018 - St Lucia - Internal

1. General Course Information

UQ students: If you are enrolled in this course, please sign in to mySI-net to view your list of enrolled courses and click the relevant **Profile** link to view the full course contributor and contact details contained in section 1.3 below.

1.1 Course Details

Course Code: STAT2201 **Course Title:** Analysis of Engineering & Scientific Data
Coordinating Unit: School of Mathematics and Physics
Semester: Semester 1, 2018 **Mode:** Internal
Level: Undergraduate
Location: St Lucia
Number of Units: 1 **Contact Hours Per Week:** <2L<1P
Pre-Requisites: MATH1050 or SA in Year 12 Maths C
Incompatible: ECON1310, ENVM2000, STAT1201, STAT1301 & STAT2203.
Course Description: Statistical models & analyses required for analysing engineering & scientific data, including sampling methods, exploratory data analysis, standard probability models, estimation, hypothesis tests, regression, experimental design.
Assumed Background: The course requires an understanding of calculus and algebra as given in MATH1050. A little understanding of matrices and multiple integration is helpful but not essential. It is a student's own responsibility to fill in any gaps in their assumed knowledge. You may need to undertake background reading to understand the lecture material.

1.2 Course Introduction

Detailed information can be found on the course webpage. Of special importance is the course schedule course schedule.

Course Changes in Response to Previous Student Feedback

The weight of the assignments has been increased from 35% to 40% and the weight of the final exam decreased accordingly.

1.3 Course Staff

Course Coordinator: Dr Yoni Nazarathy **Email:** y.nazarathy@uq.edu.au

Lecturer: Dr Yoni Nazarathy

Lecturer: Dr Slava Vaisman

UQ students: Please sign in to mySI-net to view your list of enrolled courses and click the relevant **Profile** link to access all course contributor details held in this profile

1.4 Timetable

Timetables are available on mySI-net.

Additional Timetable Information
See the course schedule.

feedback

copyright | privacy | disclaimer

ADMINISTRATION

- If you are doing STAT2201 – 1 unit course
- If you are doing CIVL2530 (2 units), then STAT2201 is about half of the CIVL2530 course.
- The Black Board site is joined for both courses

CIVL2530	
STAT2201 – 50%	50%

ADMINISTRATION - SCHEDULE

- 10 lectures
- 2 streams (**same material**):
  - Yoni Nazarathy (coordinator)
  - Slava Vaisman
- 6 tutorial meetings during the semester, each mapping to a homework assignment
- 2 hour final exam during the examination period, June 2017
- CIVL2530 students should attend both civil and stat (details below) activities. See the CIVL2530 course profile for the time table of the CIVL2530 activities.
- https://courses.smp.uq.edu.au/STAT2201/2018a/2018_weekByWeek.pdf

ASSESSMENT

- *Assignments:* 40% Best 5 out of 6, 8% each
- *Final Exam:* 60% Must get at least 40/100 on the exam to pass the course (2 hours)
  - Exam examples : https://courses.smp.uq.edu.au/STAT2201/2018a/

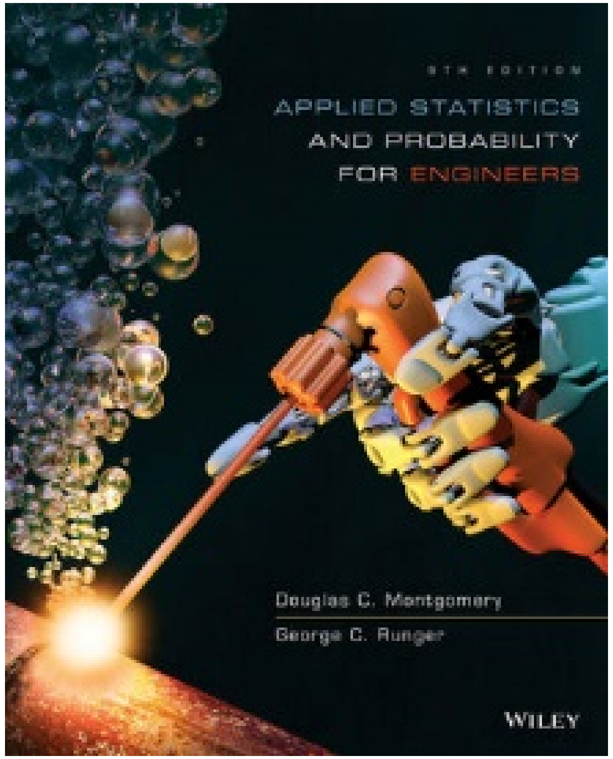
CONSULTATION HOURS

- Tutors will not provide consultations (use the lecturers)
- Yoni - Thursdays12pm –1pm, 67-753.
- Slava – Tuesday 2pm-3pm 67-450
- For technical questions, please come to consultation hours.

LEARNING MATERIAL

Study units are (mostly) mapped to

Applied Statistics and Probability for Engineers, by D. C. Montgomery and G. C. Run



LEARNING MATERIAL

- Condensed Notes **(can take to the exam!!!)**
- Get it from **https://courses.smp.uq.edu.au/STAT2201/2018a/**

STAT2201

Analysis of
Engineering & Scientific Data

Condensed Course Notes.

Semester 1, 2017.

Last Edited: April 23, 2017
Contains All Units 1 – 10

These condensed notes summarise definitions, procedures, theorems and results relevant for STAT2201. Further material is available in the course book, *Applied Statistics and Probability for Engineers" by D. C. Montgomery and G. C. Runger*, [MonRun2014] and on the course web-site: **https://courses.smp.uq.edu.au/STAT2201/2017a** .
It is recommended to bring printouts of these notes to course lectures and tutorials.

PROGRAMMING LANGUAGES

- In this course, we use Julia

- Languages that worth noting
  - R – major statistical language
  - Matlab/Octave
  - Python

UNIT 1

- **Probability vs Statistics and Data Science**
- **Deterministic vs Stochastic systems**
- **Inference**
- **Mechanistic and Empirical models**

PROBABILITY VS STATISTICS

- ***Probability*** deals with predicting the likelihood of future events. *Probability* is about creating *models (learning complex relationships)*.
  - What is the likelihood of a rainy day (tomorrow) ?
- ***Statistics*** involves the analysis of the frequency of past events. *Statistics* is about collecting data.
  - What is the average number of rainy days in Brisbane?
- Note that:
  - *Probability* can help us to collect data in a better way, and,
  - *Statistics* can be used for creating probabilistic models.
- **Data Science** is an emerging field, combining statistics, **big-data**, **machine learning** and computational techniques.

DETERMINISTIC VS STOCHASTIC SYSTEMS (1)

- Deterministic systems

  - Consider the (deterministic) function $y = x^2$.
  - For any $x \in R$, the outcome y is determined exactly.

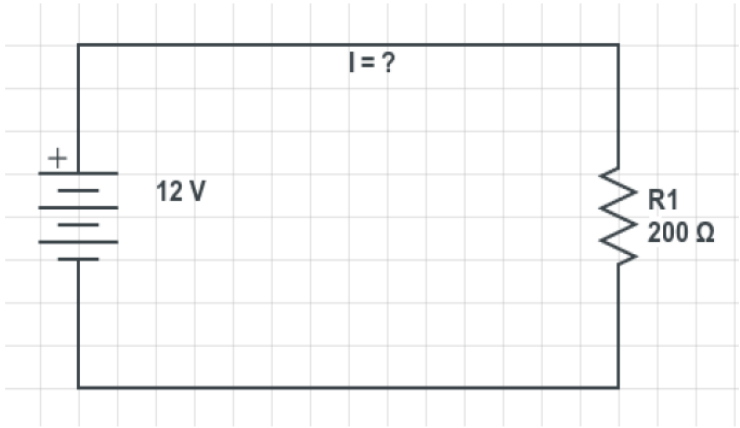
- Example Ohm's law:

$$I = \frac{V}{R}$$

  - I is the current
  - V is the voltage
  - R is the resistance

DETERMINISTIC VS STOCHASTIC SYSTEMS (2)

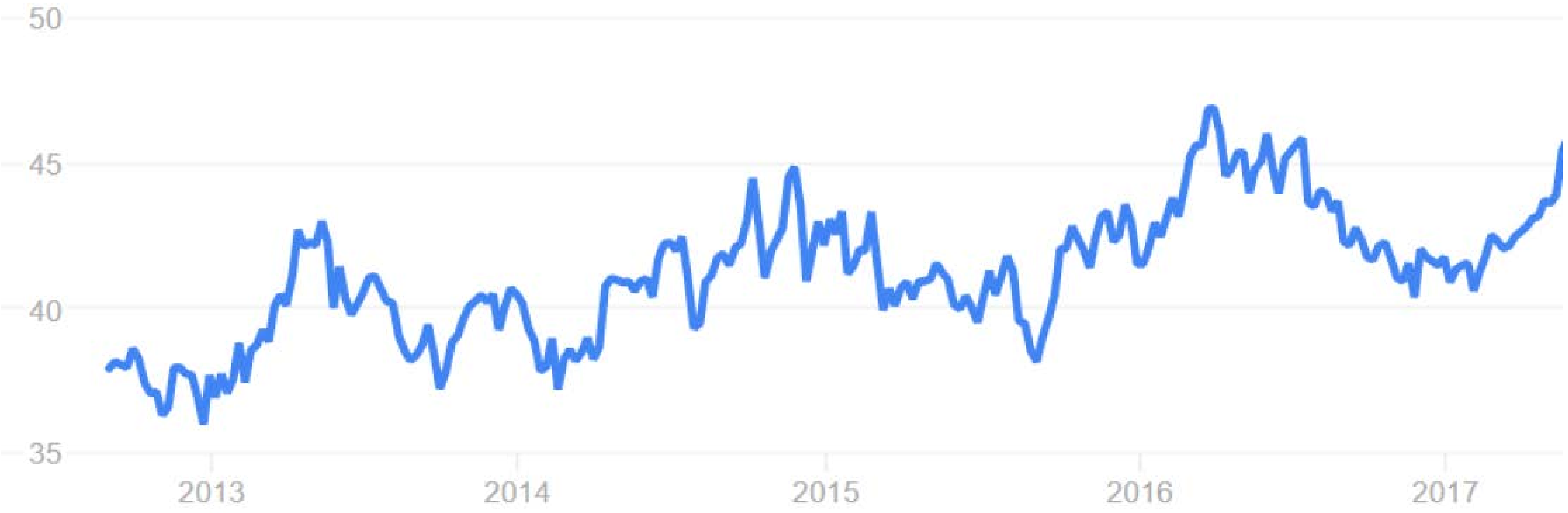
- Have you ever used *ampere-meter*?
- Do you always get the same measurement?
- Is there a noise involved?



$$I = \frac{V}{R} \Rightarrow I = \frac{12}{200} = 0.06A = 6mA \text{ (millie Ampere)}.$$

DETERMINISTIC VS STOCHASTIC SYSTEMS (3)

DETERMINISTIC VS STOCHASTIC SYSTEMS (4) THE COCA-COLA CO



THIS IS VERY INTERESTING, BUT WHY DO I NEED TO KNOW THIS?

- GPS navigation;
- Voice and video transmission systems;
- Communication over unreliable channels;
- Compression of signals;
- System reliability (system components fail randomly);
- Resource-sharing systems (random demand);
- Machine learning (visit https://www.kaggle.com/competitions);

INFERENCE

- ***Inference*** is the process of collecting data and say something about the world.
- ***Data analysis*** is the process of curating, organizing and analyzing data sets to make inferences.
- ***Statistical Inference*** is the process of making inferences about population parameters (often never fully observed) based on observations collected as part of samples.

Example:

A certain smartphone manufacturer claims that

The battery lasts 2.3 days (on average).

BATTERY LIFE (1)

- Buy **all** smartphones (say N phones were produced)
- For each smartphone, measure and record its battery life $b_1, b_2, \dots, b_N$
- **Calculate** the average battery life via

$$\frac{b_1 + b_2 + \cdots + b_N}{N}$$

BATTERY LIFE (2)

- Buy **some small number of** smartphones (say n $\ll N$)
- For each smartphone, measure and record its battery life $b_1, b_2, \dots, b_n$
- **Estimate** the average battery life via

$$\frac{b_1 + b_2 + \cdots + b_n}{n}$$

- This number is called a *statistic*.
- *Statistic* is just a quantity (a number) that we calculate from our sample.
- Sounds reasonable.
  - However, we want our estimations to be reliable.
  - How large n should be? That is, what is the *sample size*?

MECHANISTIC AND EMPIRICAL MODELS

- **Mechanistic model** is a model for which we understand the basic physical mechanism (like Ohm's law):

$$I = \frac{V}{R} + \varepsilon$$

Here, ε is a random term added to the model to account for the fact that the observed values of current flow do not perfectly conform to the mechanistic model.

- **Empirical models** are used by engineers where were is no simple or well understood mechanistic model that explains the phenomenon.

EMPIRICAL MODEL EXAMPLE (1)

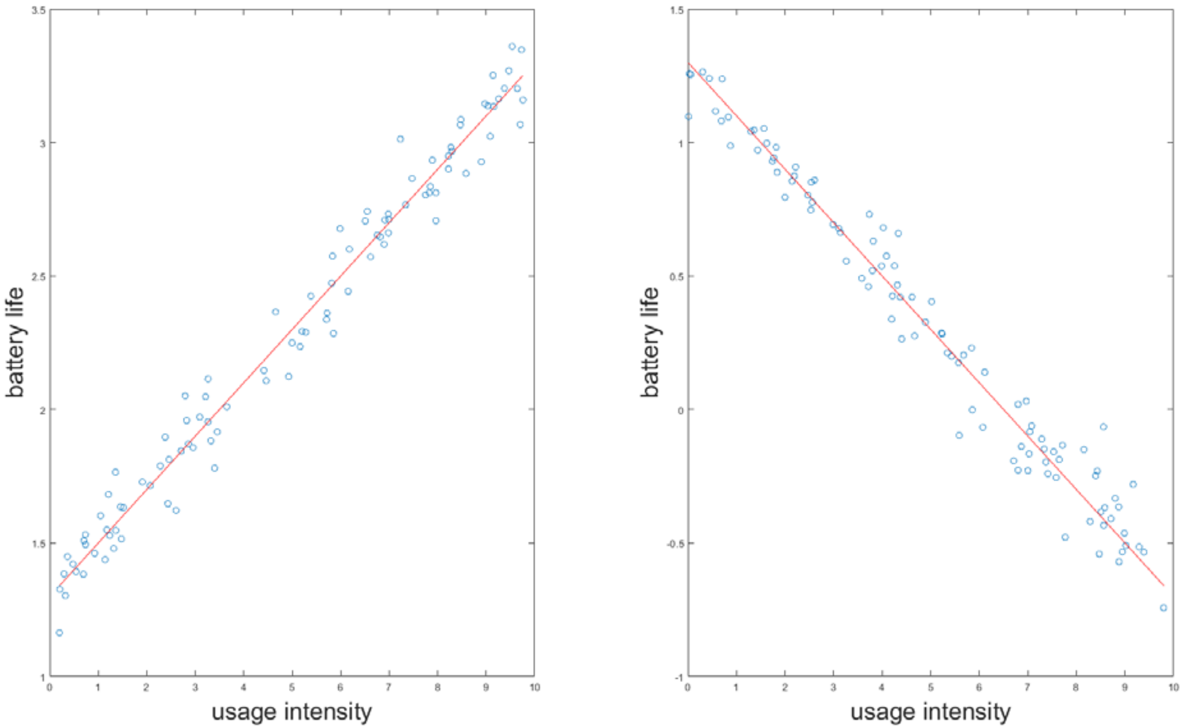
- Consider the smartphone battery life example.
- We know that the battery life (L) depends on the phone usage (U) . That is, there exists a function $L = f(U)$.
- However, f is **unknown**.
- We can try the first-order Taylor series expansion to achieve a (maybe) reasonable approximation. Namely,

$$L = \beta_0 + \beta_1 \cdot U.$$

- Here, β_0 and β_1 are unknown parameters.
- In addition, we should account for other sources of variability (like a measurement error) by adding a random parameter ε, and hence:

$$L = \beta_0 + \beta_1 \cdot U + \varepsilon.$$

EMPIRICAL MODEL EXAMPLE (2)



- We can now make an inference about the intercept (β_0) and slope (β_1), respectively.

STOCHASTIC SIMULATION

- ***Stochastic Simulation*** is about generating random numbers.
- It might be not very clear now why one would like to do so, but you will have to trust me (for now).
- Suppose for example, that I would like to play the best paying slot machine.
  - I can try to observe several machines and perform some statistics.
- Alternatively, suppose that I am slot machine designer.
  - I am building probabilistic model that specifies the likelihood of getting all kinds of winning combinations.
  - Then, I can ask a computer to generate many *"spins"* based on my model.
  - As soon as these spins are available, I can calculate many quantities of interest, such as
    - What is the machine "payout"
    - How often a player "wins"

SOURCES OF RANDOMNESS

- Natural phenomena like atmospheric and white noise, or temperature, can be used for a generation of random numbers. These are expensive.
- We would like to get random numbers using a computer. However, we have a few requirements.
  - It should be robust and reliable.
  - It should be fast.
  - It should be reproducible. That is, one should be able to recover the stream without storing it in the memory. This property is important for testing.
  - The period of the generator is the smallest number of steps taken before entering the previously visited state. A good generator should have a large period.
  - It should be application dependent. For example, in cryptography, it is crucial that the generated sequence will be hard to predict.

PSEUDORANDOM NUMBERS

- Pseudorandom number generators is an important field of study.
- We will only care about it during this lecture.
- All modern pseudorandom number generators are capable of producing a sequence

  $U_1, U_2, \ldots$ of "random" numbers such that

  1. $0 \leq U_i \leq 1,$ and
  2. $U_1, U_2, \ldots$ have a "sort of" uniform **spread** on the unit interval.
- Such a uniform spread is called ***uniform distribution*** and is denoted by $\mathbf{U}(0,1)$

A GENERAL PSEUDORANDOM NUMBER GENERATOR

- A general pseudorandom number generator will be of the following form:

Algorithm Pseudo-random number generator

---

input : An initial number $X_0 \in \mathcal{S}$ called the seed, $f : \mathcal{S} \to \mathcal{S}$, $g : \mathcal{S} \to (0,1)$.
output: A stream $U_1, U_2, \ldots,$ of pseudo-random numbers $\sim \mathsf{U}(0,1)$.

```
for t = 1 to ··· do
    X_t ← f(X_{t−1}).
    U_t ← g(X_t).
```

- In order to create such a generator, we need the following.
  - Specify an initial number (seed) for reproducibility; (this is X_0).
  - Define some ***appropriate*** functions f and g.

LINEAR CONGRUENTIAL GENERATOR (1)

Algorithm Pseudo-random number generator

input : An initial number $X_0 \in \mathcal{S}$ called the seed, $f : \mathcal{S} \to \mathcal{S}$, $g : \mathcal{S} \to (0,1)$.

output: A stream $U_1, U_2, \ldots,$ of pseudo-random numbers $\sim \mathsf{U}(0,1)$.

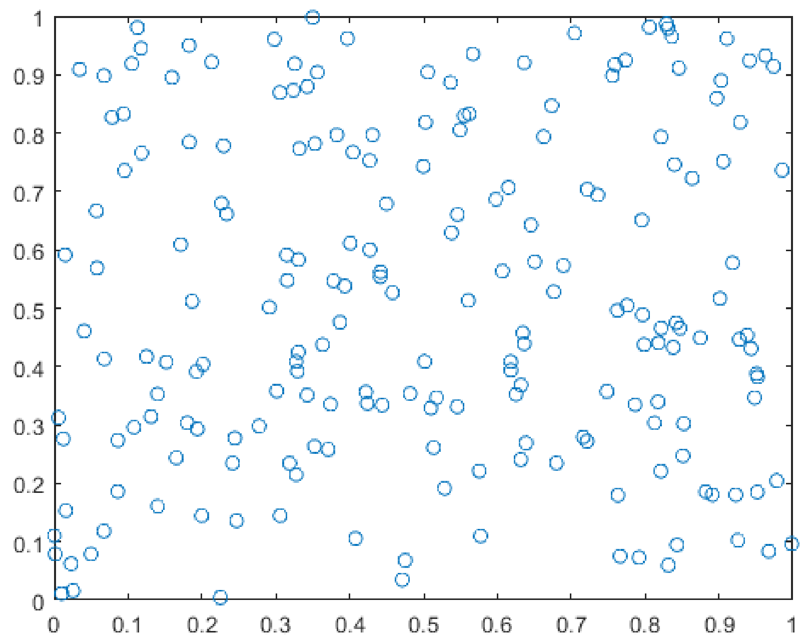
for $t = 1$ **to** $\cdots$ **do**

- $X_t \leftarrow f(X_{t-1})$.
- $U_t \leftarrow g(X_t)$.

- Define $f(X) = (aX + c) \bmod m$, and $g(X) = X/m$ for some constants a, c and m.
- Let us set for example $a = 3, c = 1$, and $m = 10{,}000$.
- Finally, set the seed $X_0 = 1$.
- We can show that:
  - $X_1 = 4 \;\; \Rightarrow \;\; U_1 = \frac{4}{10{,}000}$
  - $X_2 = 13 \;\; \Rightarrow U_2 = \frac{13}{10{,}000}$

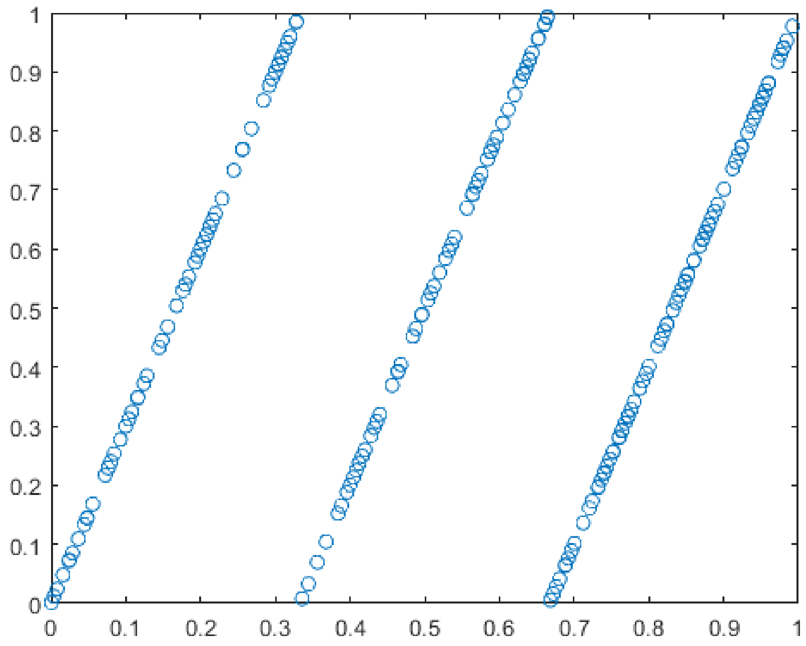
LINEAR CONGRUENTIAL GENERATOR (2)

- Suppose that we would like to use such a generator to plot two-dimensional random uniform points.
- The algorithm is simple, plot pairs $(U_1, U_2), (U_3, U_4), \ldots$
- We expect to get:



LINEAR CONGRUENTIAL GENERATOR (3)

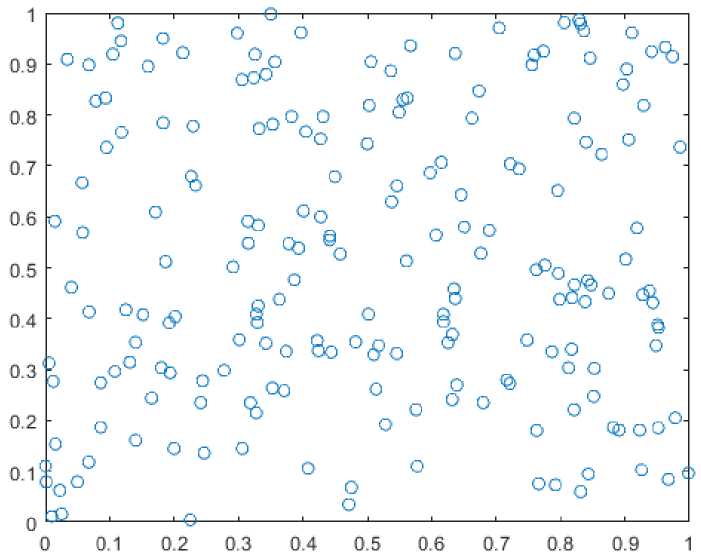
- However, using $a = 3, c = 1$, and $m = 10{,}000$, we get:



- Conclusion: $a = 3, c = 1$, and $m = 10{,}000$ is a **very bad** choice!

LINEAR CONGRUENTIAL GENERATOR (4)

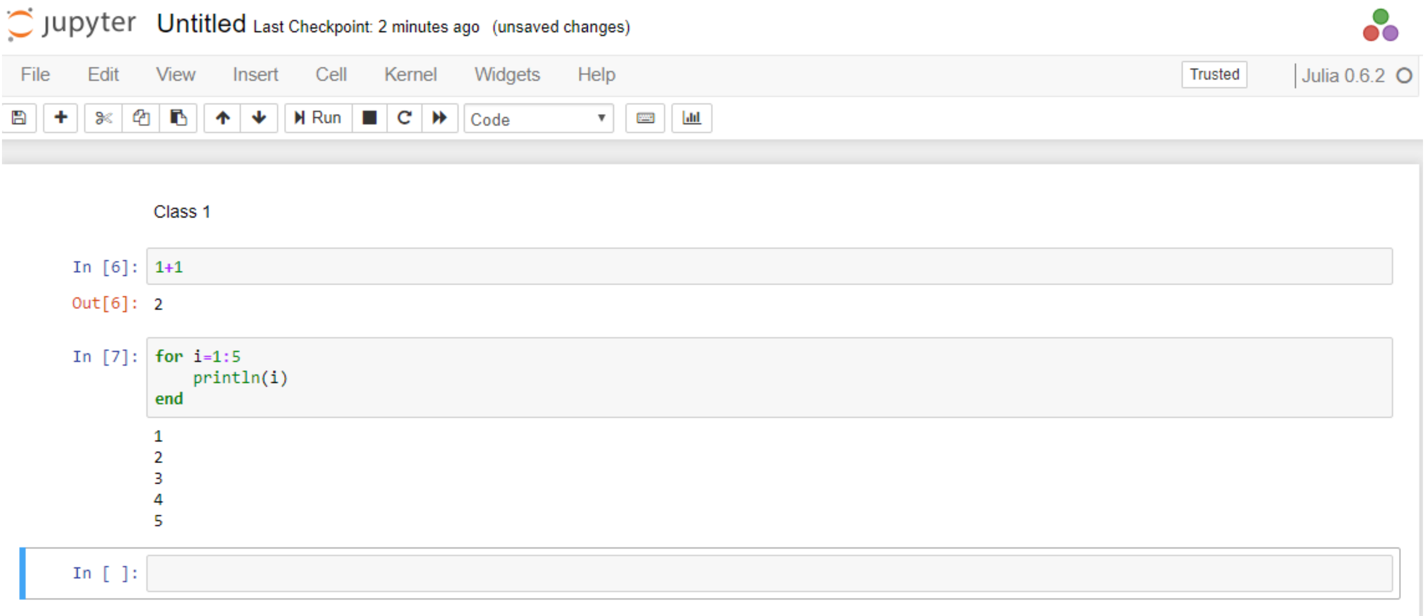
- Nevertheless, by using $a = 69069, c = 1$, and $m = 2^{32}$, we get a nice spread.



- Conclusion: except of this lecture, we do not implement random generators! We use the ones that passed appropriate statistical tests!

INTRODUCTION TO JULIA

- https://juliabox.com/up/uq/AJUF5NQ
- Comfortable web interface



INTRODUCTION TO JULIA – CELL TYPES (1)

Cell types: ***Markdown*** and ***Code***

Class 1

```
In [8]: 1+1
Out[8]: 2
```

```
# This is a text
# this is a formula $ \frac{\sum_{i=1}^n b_i}{n}$
```

```
In [ ]:
```

Class 1

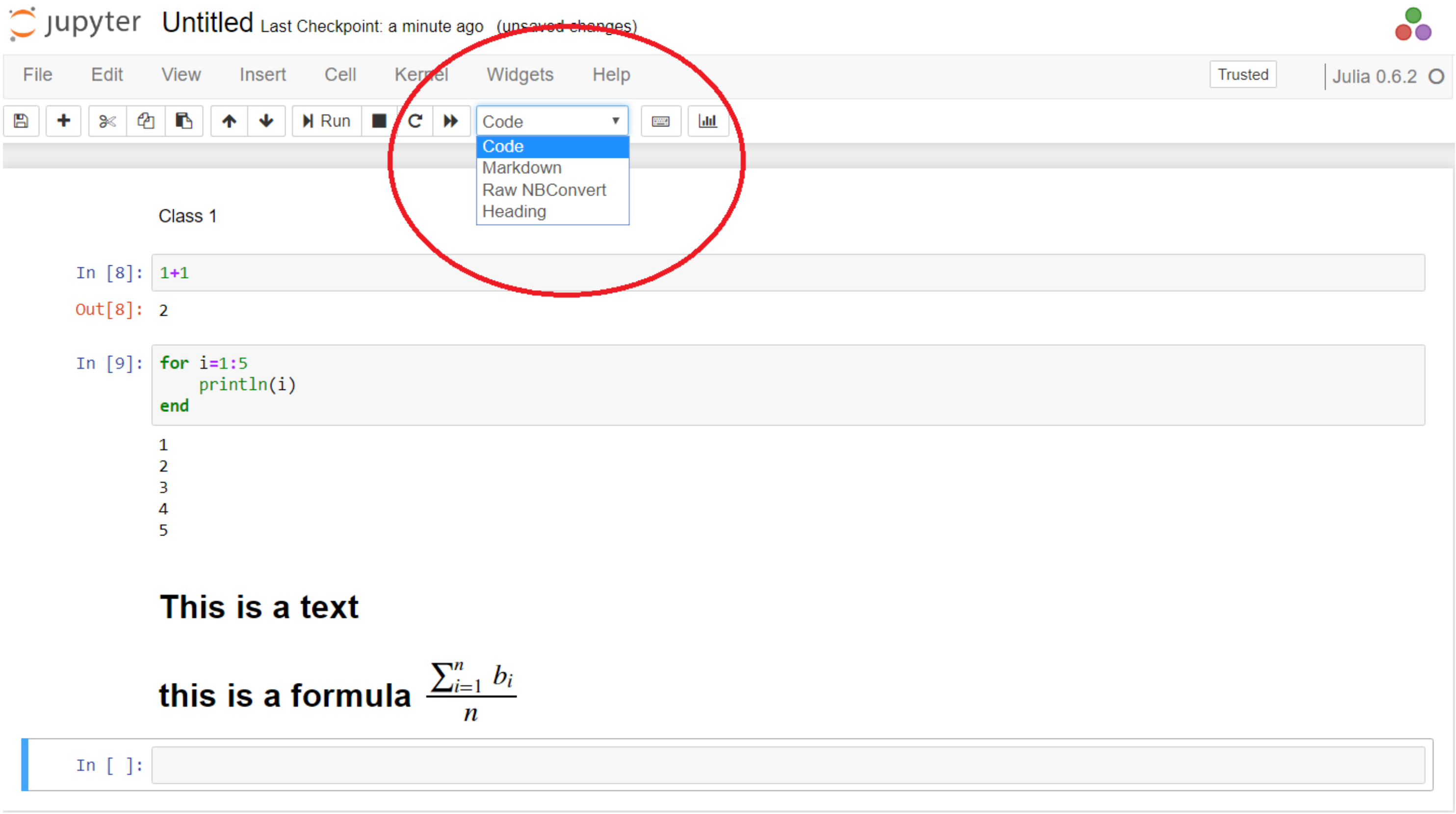
```
In [8]: 1+1
Out[8]: 2
```

This is a text

this is a formula $\frac{\sum_{i=1}^n b_i}{n}$

```
In [ ]:
```

INTRODUCTION TO JULIA – CELL TYPES (2)



Useful command: mark a cell and press "x" to delete it.

LINEAR CONGRUENTIAL GENERATOR IN JULIA (1)

- Check JuliaReferenceSheet.pdf

Linear congruential generator

In [51]:

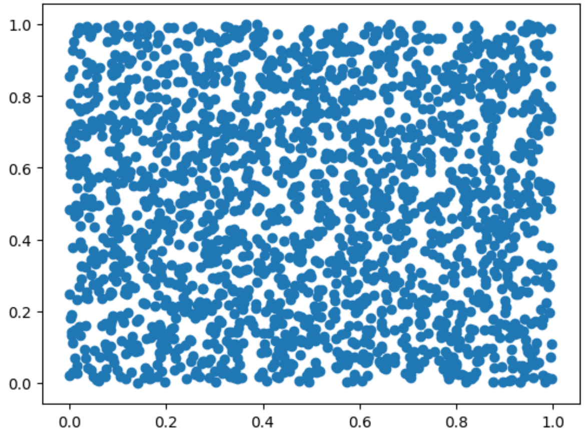
```
#a = 3
#c = 1
#m = 10000

a = 69069
c = 1
m = 2^32


function f(X)
    return mod((a*X + c), m);
end

# set seed
X = 1979

X_1 = f(X)
X_2 = f(X_1)

println(X_1)
println(X_2)
```

```
136687552
534412481
```

LINEAR CONGRUENTIAL GENERATOR IN JULIA (2)

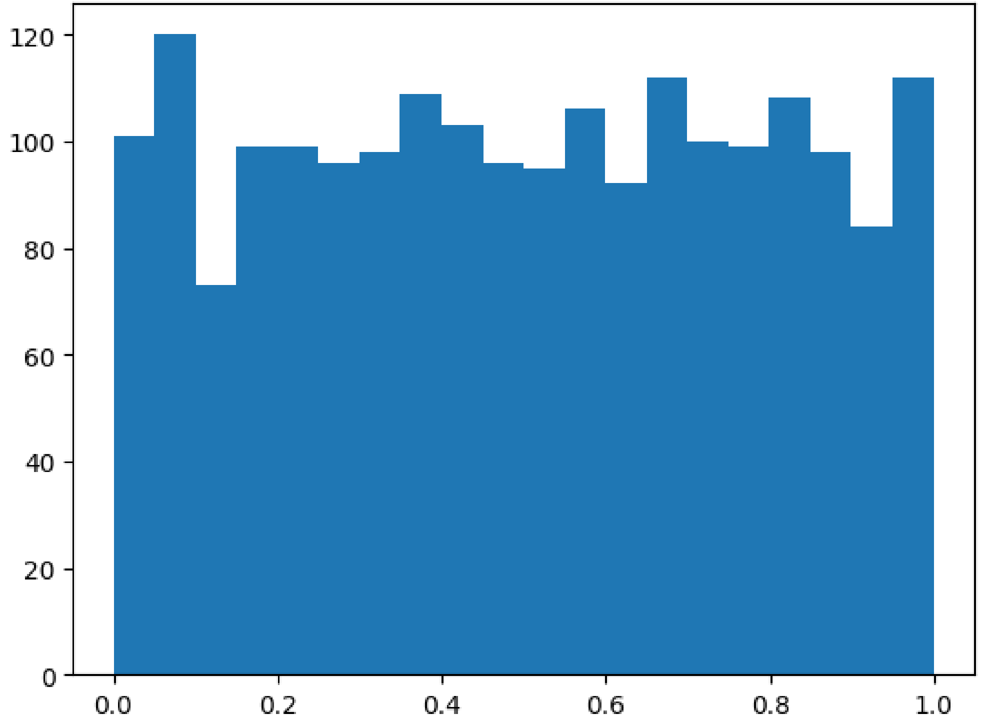
In [54]:
```
# number of observations
n = 2000

U1 = []
U2 = []

for i=1:n
   X = f(X)
   push!(U1,X/m)
   X = f(X)
   push!(U2,X/m)
end

using PyPlot
```

In [55]:
```
PyPlot.scatter(U1,U2)
```



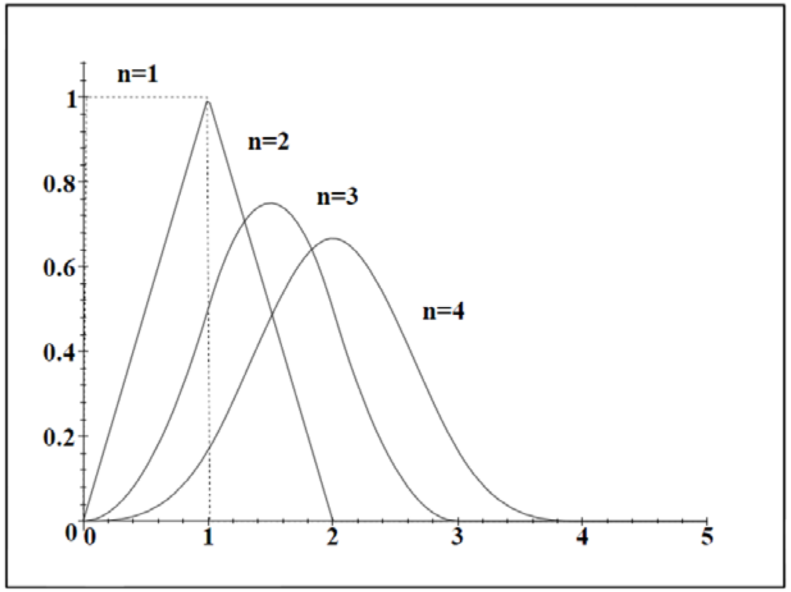
LINEAR CONGRUENTIAL GENERATOR IN JULIA (3)

In [59]:
```
PyPlot.plt[:hist](U1,20)
```



ADDING RANDOM NUMBERS

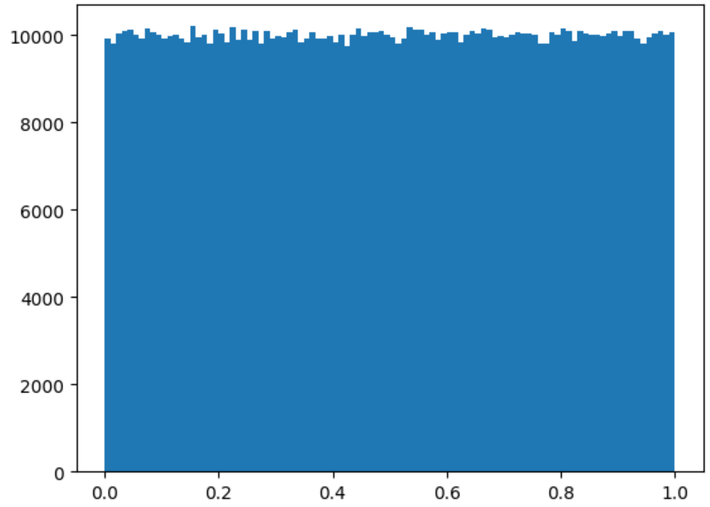
- Suppose that we have some random numbers $\{X_1, \dots, X_n\}$.
- Let us sum them to get a new random number $S_n = X_1 + X_2 + \cdots + X_n$.
- Then, S_n has a special spread (distribution), called a Gaussian distribution.



ADDING RANDOM NUMBERS IN JULIA (1)

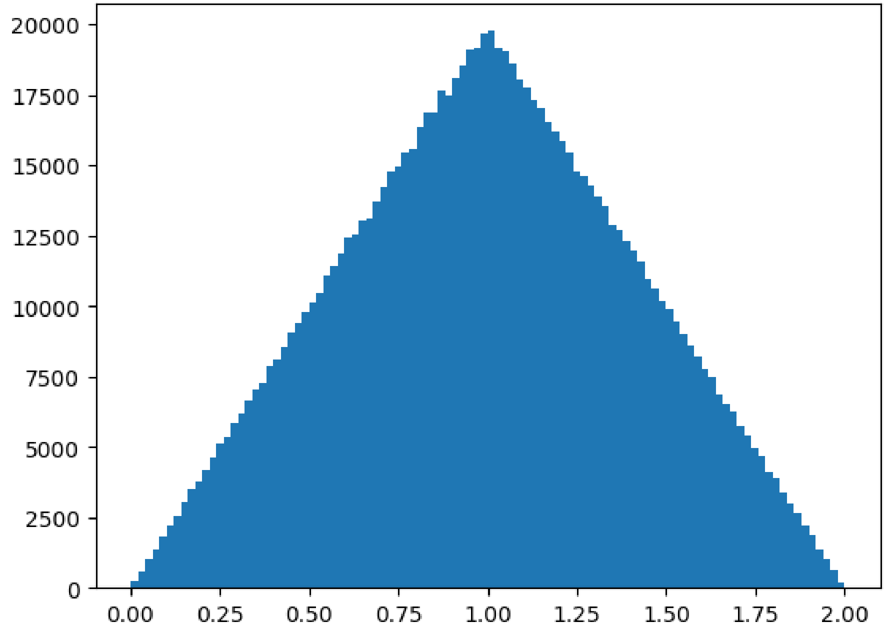
Adding random numbers

In [15]:
```
using PyPlot
n = 1000000;
X1 = rand(n,1);
S = X1;
PyPlot.plt[:hist](S,100)
```



ADDING RANDOM NUMBERS IN JULIA (2)

In [16]:
```
X2 = rand(n,1);

S = X1+X2;

PyPlot.plt[:hist](S,100)
```



ADDING RANDOM NUMBERS IN JULIA (3)

In [17]:
```
X3 = rand(n,1);

S = X1+X2+X3;

PyPlot.plt[:hist](S,100)
```