

Simulation
Projet d'examen:
Étude du caractère pseudo-aléatoire de π

Libert Robin BA3 Info
Umons

Juin 2018

Table des matières

1	Introduction	2
1.1	Énoncé	2
1.2	Logiciels utilisés	2
2	Intuitions	3
3	Tester le caractère pseudo-aléatoire de 1 million de décimales de π	5
3.1	Test de χ^2 avec 9 ddl	5
3.1.1	Rappel théorique	5
3.1.2	Résultats et implémentation	5
3.2	Test du Gap avec 49 ddl	7
3.3	Test du Poker avec 4 ddl	7

Chapitre 1

Introduction

1.1 Énoncé

Dans le cadre du projet du cours de Simulation, dispensé par Monsieur Alain BUYS, il nous a été demandé d'étudier le caractère pseudo-aléatoire des décimales de π . Pour ce faire, nous allons utiliser des tests vus au cours dans le but de savoir si les décimales de π suivent une loi uniforme ou non.

Dans un second temps, nous devons utiliser les décimales de π pour créer un générateur de loi uniforme $[0,1[$ et le comparer au générateur par défaut de Python.

1.2 Logiciels utilisés

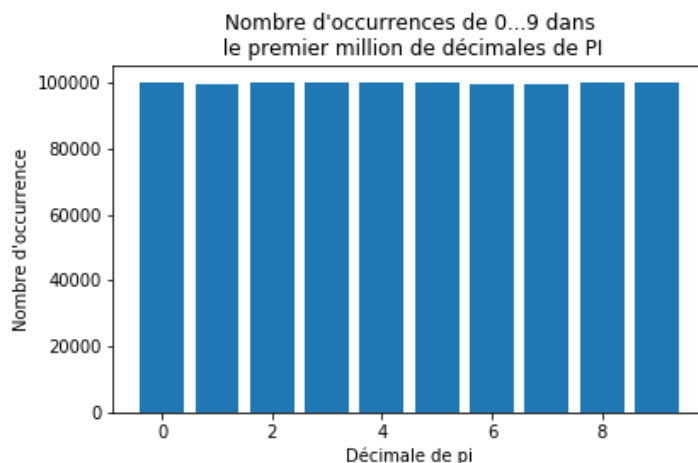
- Spyder : ide python
- Python 3 : pour la réalisation du code
- Mathplotlib : pour la réalisation des graphiques. (Bibliothèque python)
- Windows 10
- TeXnicCenter : pour la rédaction du rapport en Latex

Chapitre 2

Intuitions

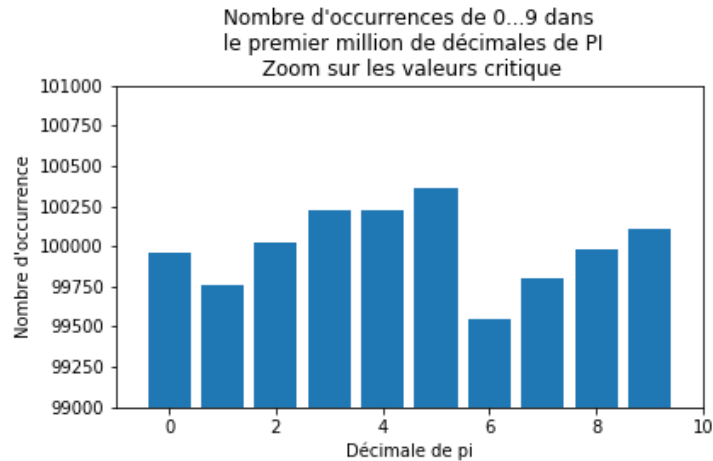
Pour nous donner une première intuition, nous avons créé un diagramme à batons comptant le nombre de fois que chaque digit de π apparaît dans le premier million de décimales de π . En théorie, vu que nous considérons 10 digits allant de 0 à 9, une loi uniforme devrait compter 100000 occurrences de chaque digit, vu que nous considérons 1 million de décimales.

Voici un aperçu de notre diagramme à baton. Chaque digit semble bien apparaître 100000 fois et donc, les décimales de π semblent suivre une loi normale.



Faisons un zoom sur les valeurs critiques du diagramme. Nous constatons ci-dessous que les digits n'apparaissent pas 100000 fois. Ce n'est pas pour autant que les décimales de π ne suivent pas une loi uniforme. En théorie, ce serait le cas si nous avions et pouvions observer une infinité de décimale, mais comme ce n'est pas le cas, il est normal d'avoir une erreur. Nous allons donc devoir vérifier avec différents tests, si cette erreur est significative ou pas en fonction

du nombre de décimales que nous avons à notre disposition.



En mettant les valeurs dans un tableau et calculant le pourcentage d'erreur relative, nous constatons que les résultats expérimentaux de différent jamais de plus de 0.5% des résultats théoriques.

Digit	Théorie	Expérimentaux	Erreur relative %
0	100000	99959	0.041
1	100000	99758	0.242
2	100000	100026	0.026
3	100000	100229	0.229
4	100000	100230	0.230
5	100000	100359	0.359
6	100000	99548	0.452
7	100000	99800	0.200
8	100000	99985	0.015
9	100000	100106	0.106

En résumé, les résultats préliminaires tendent à nous faire penser que les décimales de π suivent une loi uniforme, du moins pour le premier million. Dans le chapitre suivant, nous allons tenter de prouver que le premier million de décimales de π suivent bien une loi uniforme. Nous allons utiliser 3 tests différents, le test de χ^2 , le test du Gap et le test de la main de Poker.

Chapitre 3

Tester le caractère pseudo-aléatoire de 1 million de décimales de π

3.1 Test de χ^2 avec 9 ddl

3.1.1 Rappel théorique

Dans le test de χ^2 , nous devons créer un histogramme de r intervalles et compter le nombre de valeurs générées dans chaque intervalle n_i . Dans chaque intervalle, on s'attend à avoir à peu près le même nombre de points car les probabilités p_i sont égales.

Dans un premier temps, nous devons calculer $K_n = \sum_{i=1}^r \left(\frac{n_i - Np_i}{\sqrt{Np_i}} \right)^2$. Ensuite nous devons comparer K_n et χ^2 pour un certain α , qui est la probabilité de rejeter l'hypothèse nulle H_0 et un certain degré de liberté qui vaut $r - 1$. Nous retenons notre hypothèse nulle si $K_n \leq \chi^2$ et nous la rejetons sinon.

3.1.2 Résultats et implémentation

Rapportons le test de χ^2 à notre problème.

r est le nombre d'événements distincts et vaut 10

n_i est le nombre d'occurrences observées de l'événement i

N est la taille de l'échantillon et vaut 1000000

p_i est la probabilité d'obtenir l'événement i et vaut $\frac{1}{10}$

Cette première fonction prend en paramètre la liste des décimales de π . Elle retourne la liste du nombre d'occurrence de chaque digit de 0 à 9 dans les décimales de π . Le nombre de fois qu'apparaît 0 dans les décimales de π est le nombre se trouvant à l'indice 0 de la liste.

```

def countOccurrences(l):
    n_occurrences = [0,0,0,0,0,0,0,0,0,0]
    for n in range(len(l)):
        n_occurrences[l[n]] += 1
    return n_occurrences

```

Cette fonction va prendre en paramètres la liste retournée par la fonction countOccurrences appliquée sur les décimales de π et retournera la valeur de K_n avec comme probabilité $\frac{1}{10}$.

```

def khi2(listReal):
    sizeEch = 0
    for e in listReal:
        sizeEch += e
    Kn = 0
    for i in range(10):
        p = sizeEch*1/10
        Kn += ((listReal[i] - p) / math.sqrt(p))**2
    return Kn

```

α	K_n	χ^2	$K_n \leq \chi^2$
0.001	5.51	27.88	True
0.05	5.51	16.92	True
0.1	5.51	14.68	True

3.2 Test du Gap avec 49 ddl

Numéro	α	K_n	χ^2	$K_n \leq \chi^2$
1	0.001	49.99	85.35	True
1	0.05	49.99	66.34	True
1	0.1	49.99	62.04	True
2	0.001	36.34	85.35	True
2	0.05	36.34	66.34	True
2	0.1	36.34	62.04	True
3	0.001	37.81	85.35	True
3	0.05	37.81	66.34	True
3	0.1	37.81	62.04	True
4	0.001	72.07	85.35	True
4	0.05	72.07	66.34	False
4	0.1	72.07	62.04	False
5	0.001	54.14	85.35	True
5	0.05	54.14	66.34	True
5	0.1	54.14	62.04	True
6	0.001	47.72	85.35	True
6	0.05	47.72	66.34	True
6	0.1	47.72	62.04	True
7	0.001	44.87	85.35	True
7	0.05	44.87	66.34	True
7	0.1	44.87	62.04	True
8	0.001	38.03	85.35	True
8	0.05	38.03	66.34	True
8	0.1	38.03	62.04	True
9	0.001	36.10	85.35	True
9	0.05	36.10	66.34	True
9	0.1	36.10	62.04	True

3.3 Test du Poker avec 4 ddl

α	K_n	χ^2	$K_n \leq \chi^2$
0.001	4.61	18.47	True
0.05	4.61	9.49	True
0.1	4.61	7.78	True