كليـــــة الحاسبات والذكاء الاصطناعي

كونترول الفرقة الثانية

العـــام الجامعـــى 2019 / 2020 – دور مايـــو

أولا: البيانات الخاصة بالطالب

الفرقة الدراسية	الثانية	التخصص	عام
اسم القسم	عام		
اسم المقرر	نمذكه و محكاه		
استاذ المقرر	Dr.Mai el defrawi		

ثانيا: البيانات الخاصة بالبحث

عنوان البحث	Monte carlo		
طبيعة المشاركة	بحث فردى [نعم]		بحث جماعى [ ]
ارسال البحث	بواسطة البريد الالكتروني		

	م	الاسم رباعى	رقم الجلوس	الرقم القومى
اسماء الطلاب المشاركين فى البحث (يكتب الاسم رباعيا)	1	هاجر عبدالتواب زكريا عمر	2717	29912160102525
	2			
	3			
	4			
	5			
تاريخ الإرسال	7 / 6 / 2020			

ثالثا: البيانات الخاصة بالكونترول

النتيجة	ناجح [ ]		راسب [ ]

		الاسماء	التوقيع
أعضاء لجنة تقييم البحث	1		
	2		
	3		

فى حالة عدم قبول البحث يرجى ذكر الأسباب	- ..... - ..... - ..... - .....

Monte Carlo

Introduction:

Simulations are useful for several purposes. You explore situations that it would be difficult or impossible to undertake and do research for practical or ethical reasons. Simulations put models in context by letting us observe how the model behaves during a very complex environment.
Monte carlo simulations are used to model the probability of assorted outcomes during a very process that can't easily be predicted because of the intervention of random variables. It is some way used to understand the impact of risk and uncertainty in prediction and forecasting models.
monte carlo simulation are going to be used to tackle an expansion of problems in virtually every field like finance, engineering, supply chain, and science.

Monte Carlo Explanation:

When faced with significant uncertainty within the method of making a forecast or estimation, rather than just replacing the uncertain variable with one average number, the town simulation might sway be a more robust solution.
Since business and finance are full of random variables, town simulations have an unlimited array of potential applications in these fields.
They are used to estimate the probability of cost overruns in large projects and also the likelihood that an asset price will move during a very certain way.
Telecoms use them to assess network performance in numerous scenarios, helping them to optimize the network.
Analysts use them to assess the danger that an entity will default and to research derivatives like options.
Insurers and oiler drillers also use them. town simulations have countless applications outside of business and finance, like in meteorology, astronomy and physics.

The problems used :

Monte carlo methods are especially useful for simulating phenomena application include: physical sciences , engineering , global global climate change and radiative forcing , computational biology , special effects , applied statistics , computing for games , design and visuals , search and rescue , finance and business , law and use in mathematics
In general, the town methods are employed in mathematics to resolve various problems by generating suitable random numbers.

Advantage:

Very flexible. Empirical distributions are going to be handled.
Can generally be easily extended and developed PRN.
Easily understood by non-mathematicians.

Disadvantage:

Usually requires a computer.
Calculations can hold for much longer than analytical models.
Solutions don't seem to be exact, but rely on the quantity of repeated runs used to produce the output statistics. That is, all outputs are estimates.

Application :

Table1 consists of rows , onInputChangeD , demands , onInputChangeF , frequency, prob , cum and randomNumber. For i =1 to I <= rows , push in eleArray .
Coulum onInputChangeD value =demands[‘d${i }’]
Coulum onInputChangeF value = frequency [‘f${i }’]
Coulum prob value = prob[i - 1]
Coulum cum value = cum[i -1]

Table1.jsx Table2.jsx

```jsx
const Table1 = () => {
  const {
    rows,
    onInputChangeD,
    demands,
    onInputChangeF,
    Frequency,
    prob,
    cum,
    randomNumbers,
  } = useContext(AppContext);
  return (
    <Table borderless hover>
      <thead>
        <tr>
          <th>Demand</th>
          <th>Frequency</th>
          <th>P{"ROBABILITY".toLowerCase()}</th>
          <th>C{"UMULATIVE".toLowerCase()}</th>
          <th>Number-Range</th>
        </tr>
      </thead>
      <tbody>
        {(() => {
          let eleArray = [];
          for (let i = 1; i <= rows; i++) {
            eleArray.push(
              <tr key={`${i}tr`}>
                <td className="inputter">
                  <Input
                    type="number"
                    name={`d${i}`}
                    onChange={onInputChangeD}
                    value={demands[`d${i}`]}
                  />
                </td>
                <td className="inputter">
                  <Input
                    type="number"
                    name={`f${i}`}
                    onChange={onInputChangeF}
                    value={Frequency[`f${i}`]}
                  />
                </td>
                <td>{prob ? prob[i - 1] : null}</td>
                <td>{cum ? cum[i - 1] : null}</td>
                <td>
                  {randomNumbers[i - 1]
                    ? `${randomNumbers[i - 1][0]} to ${
                        randomNumbers[i - 1][randomNumbers[i -
                        1].length - 1]
                      }`
                    : null}
                </td>
              </tr>
```

Table2 consists of simSum , expected , random and sim.
Simulation average daily demand fr 10 days = simSum/10
Create table head : Day , Number , demand
That body , for i =1 to i<=10 push value in arr ,
i , random[i -1] , sim [i -1] and return arr.

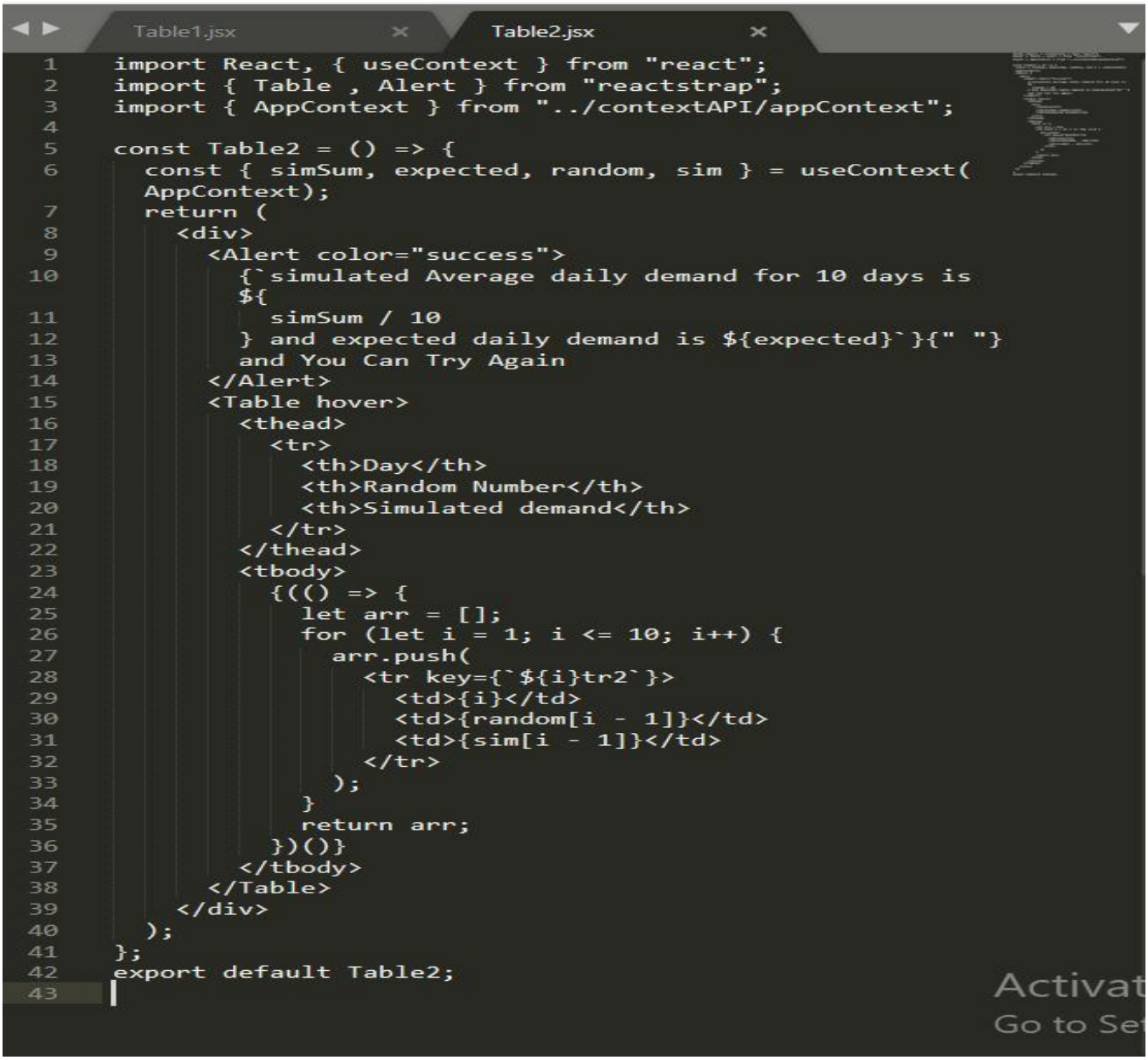
Table1.jsx
Table2.jsx
import React, { useContext } from "react";
import { Table , Alert } from "reactstrap";
import { AppContext } from "../contextAPI/appContext";
const Table2 = () => {
const { simSum, expected, random, sim } = useContext(
AppContext);
return (

<Alert color="success">
{`simulated Average daily demand for 10 days is
${
simSum / 10
} and expected daily demand is ${expected}`}{" "}
and You Can Try Again
</Alert>
<Table hover>
<thead>
<tr>
<th>Day</th>
<th>Random Number</th>
<th>Simulated demand</th>
</tr>
</thead>
<tbody>
{(() => {
let arr = [];
for (let i = 1; i <= 10; i++) {
arr.push(
<tr key={`${i}tr2`}>
<td>{i}</td>
<td>{random[i - 1]}</td>
<td>{sim[i - 1]}</td>
</tr>
);
}
return arr;
})()}
</tbody>
</Table>

);
};
export default Table2;

6 Basic

Demand	Frequency	Probability	Cumulative	Number-Range
0	15	0.1	0.1	1 to 10
1	30	0.05	0.15	11 to 15
2	60	0.1	0.25	16 to 25
3	120	0.2	0.45	26 to 45
4	45	0.4	0.85	46 to 85
5	30	0.15	1	86 to 100

Get Result

Day	Random Number	Simulated demand
1	93	0
2	13	4
3	30	2
4	14	4
5	44	2
6	89	0
7	54	1
8	80	1
9	85	1
10	73	1

References

Book quantitative analysis for management