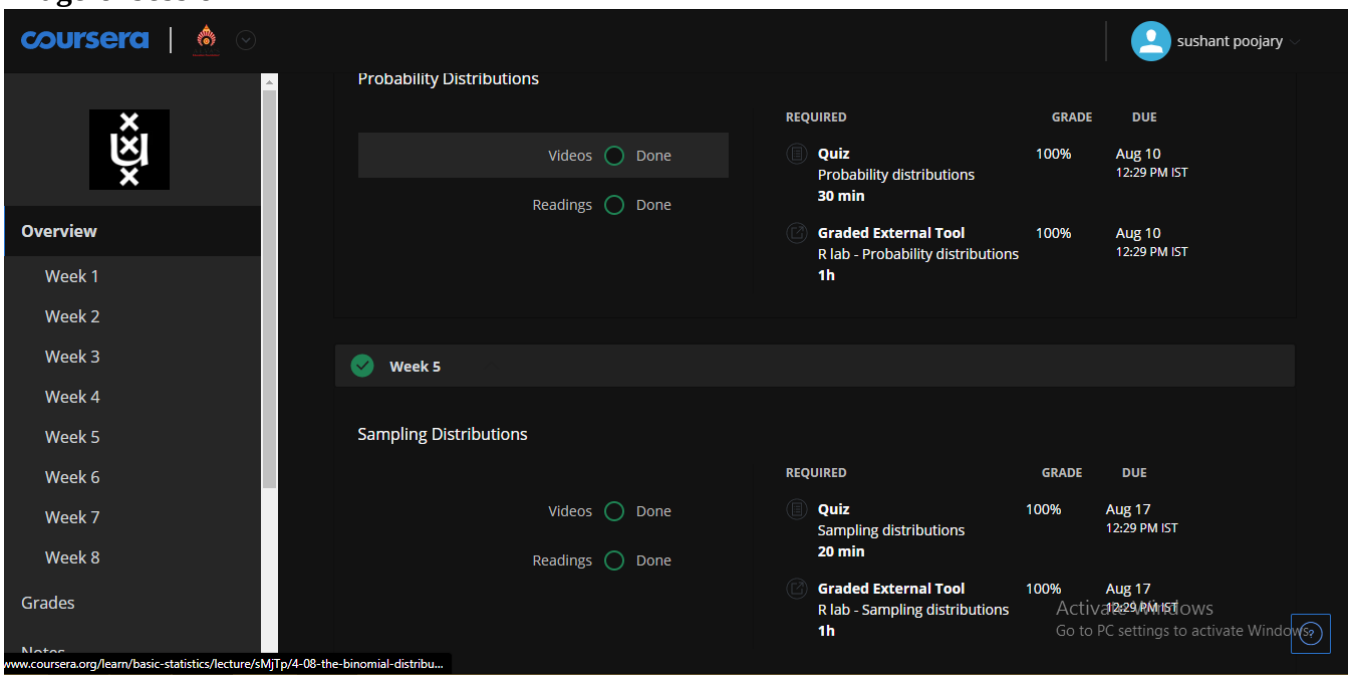
DAILY ASSESSMENT FORMAT

Date:	**July 2020**	**Name:**	**Poojary Sushant**
Course:	**Coursera**	**USN:**	**4AL18EC400**
Topic:	**Basic statistics**	**Semester & Section:**	**6th sem 'B"**
Github Repository:	**Sushant-poojary**		

FORENOON SESSION DETAILS

Image of session



Report –

Randomness is not an intrinsic property of a phenomenon. It also depends amongst others of prior knowledge, observation method, and a scale at which the phenomenon is considered.

While there are many words to express aspects of randomness, humans are not very good in assessing it quantitatively. We suffer from apophenia, the over interpretation of what are purely random patterns and are also bad in constructing randomness.Also the scale of your search matters. While you may not be very certain about finding another shell within a few minutes at a short stretch of beach, the chance to find one when you take a bit more time and cover a larger stretch of beach will increase.

But in spite of many words, our ability to memorize in our daily experience with randomness, we are not very good at assessing it quantitatively at all.

On one hand, we see all sorts of patterns in what is really random data. There's a word for it, apophenia.

And on the other hand, we are unable to make up random data ourselves.

An example of a failed attempt to create random data is this fabricated map of random shell locations which turn out to be spaced too regular.

This is how a realistic random point pattern looks like with much more clusters.

Another example of over interpreting randomness is the so called gambler's fallacy, the false idea that a random phenomenon can be predicted from a series of preceding random phenomena.Before we can understand probability we first have to understand another concept: randomness. The first video explains this concept. It also shows that even though randomness is everywhere around us, humans are nonetheless bad in assessing it.

Once we understand randomness we can define probability as a way to quantify randomness. The second video explains how this quantification can be accomplished by experiments which record the relative frequency that certain events of interest occur. It follows that probabilities are always larger or equal to zero and smaller or equal to one; and also that the sum of the probabilities for all possible events equals one. Due to the very nature of random events, the experiments may have to continue for a while before the relative frequencies represent the probabilities accurately, but the law of large numbers dictates that it will do so eventually.

Recognizing and understanding randomness as well as the ability to reason about it are important skills, not only to apply statistical analysis but also to make sense of things happening around us every day. Here, I will explain that humans are pathologically bad at dealing with randomness. I will use an example along the way.

Imagine you're on a beach watching the waves roll in.

And then your attention is caught by a beautiful shell which is distinctive in shape and larger than it's neighboring shells. So you start to search for another one. This will be an unpredictable enterprise. The shells may be distributed at random at this huge beach. Hence, the time it will take you to find another will be uncertain. You may not be able to find a similar at all.

The more you think about it, the more you'll realize that randomness is pervasive in everyday life. It is therefore not surprising that we have a rich vocabulary to communicate it,

with terms like uncertainty, chance, risk and likelihood.

Also, degrees of variability and uncertainty can expressed quite subtly. Consider for example this sequence. Rarely. Seldom. Sometimes. Common. Frequent. Often.

Importantly, whether something is random is not just a property of that phenomenon, but very much also a consequence of our knowledge about it.

If you'd have been at this beach before, you might have spotted this special shell previously, so that this may change your search strategy and increase your chance of finding more of them.

And who doesn't recognize this. If you have thrown a six four times in a row with a die, it feels as if it's going to be very unlikely to throw a six again the fifth time.

Yet, the probability of this outcome was and continues to be one sixth.
The reason that we have a bad head for randomness is that our brain tends to measure randomness in the effort it takes to memorize a pattern.
And it turns out that memorizing frequently changing short sequences is harder than longer sequences.
Given this state of affairs it's really important to learn about formal ways for quantifying randomness, reasoning about it and generating realistic random patterns. It will help to avoid mistakes, predict more accurate and be more efficient when you try to make sense of the world around you. Let me summarize what I hope you understood from this video.

Date:	**24th June 2020**	**Name:**	**Poojary Sushant**
Course:	**Pythonic coding**	**USN:**	**4AL18EC400**
Topic:	**Python**	**Semester & Section:**	**6th & 'B,**
GitHub Repository :	**Sushant-poojary**		

Report –

Writing Pythonic Code

Example #1

```
x=[1, 2, 3, 4, 5, 6]
result = []
for idx in range(len(x)):
        result.append(x[idx] * 2)
print(result)
```

Output: [2, 4, 6, 8, 10, 12]

Consider the above code, where you're trying to multiply some elements, "x" by 2.
So, what we did here was, we created an empty list to store the results. We would then append the solution of the computation into the result. The result now contains a function which is 2 multiplied by each of the elements.

Now, if you were to write the same code in a Pythonic way, you might want to simply use list comprehensions.
Here's how:

```
x=[1, 2, 3, 4, 5, 6]
print([(element * 2) for element in x])
```

only 2 line! it is pythonic

Example #2

```
x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
result = []
for idx in range(len(x)):
        if x[idx] % 2 == 0:
                result.append(x[idx] * 2)
        else:
                result.append(x[idx])
print(result|)
```

We've actually created an if else statement to solve this problem, but there is a simpler way of

doing things the Pythonic way. The Pythonic way is to combine for and if using list comprehension

```
x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[(element * 2 if element % 2 == 0 else element) for element in x]
```

Output: [1, 4, 3, 8, 5, 12, 7, 16, 9, 20]

Example #3 filtering only even numbers

```
x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
result = []
for idx in range(len(x)):
        if x[idx] % 2 == 0
print(result|)
```

We've actually created an if else statement to solve this problem, but there is a simpler way of doing things the Pythonic way.

```
x=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[element * 2 for element in x if element % 2 == 0]
```

- How to Write Beautiful Python Code With PEP 8

PEP stands for Python Enhancement Proposal, and there are several of them. A PEP is a document that describes new features proposed for Python and documents aspects of Python, like design and style, for the community.

- Naming Conventions

When you write Python code, you have to name a lot of things: variables, functions, classes, packages, and so on. Choosing sensible names will save you time and energy later. You'll be able to figure out, from the name, what a certain variable, function, or class represents. You'll also avoid using inappropriate names that might result in errors that are difficult to debug.

Never use l, O, or I single letter names as these can be mistaken for 1 and 0, depending on typeface:

```
O = 2  # This may look like you're trying to reassign 2 to zero
```

- Naming Styles