STAT2201, Semester 1, 2017

Solution for Assignment 1

Question 1 - A digital scale

Errata Notes:

- In an earlier version of the assignment, (e) was missing brackets.
- In the current version (i) is exactly like (e)

A digital scale that provides weights to the nearest gram is used.

(a) What is the sample space for this experiment?

Let A denote the event that a weight exceeds 11 grams, let B denote the event that a weight is less than or equal to 15 grams, and let C denote the event that a weight is greater than or equal to 8 grams and less than 12 grams. Describe the events:

(b) $A \cup B$

(c) $A \cap B$

(d) $\overline{A}$

(e) $A \cup (B \cap C)$

(f) $\overline{(A \cup C)}$

(g) $A \cap B \cap C$

(h) $\overline{B} \cap C$

(i) $A \cup (B \cap C)$

Solution:

(a) One option for the sample space: $\Omega = \{x \;:\; x \in \{0,1,2,3,\dots\}\}$

(b) $A \cup B = \Omega$

(c) $A \cap B = \{12,13,14,15\}$

(d) $\overline{A} = \{0,1,2,\dots,11\}$

(e) Note that on the first release this HW, the question was $A \cup B \cap C$ in which case you need to take care for the order of operations. We don't want to worry about that for this course and wish to avoid such expressions. Now with $A \cup (B \cap C)$ the answer is

$$A \cup (B \cap C) = \{8,9,10,\dots\}$$

(f) $\overline{A \cup C} = \{0,1,2,3,4,5,6,7\}$

(g) $A \cap B \cap C = \emptyset$

(h) $\overline{B} \cap C = \emptyset$

(i) This is the same as (e) by mistake.

Question 2 - Transmitting bits

Four bits are transmitted over a digital communications channel. Each bit is either distorted or received without distortion. Let A_i denote the event that the i^{th} bit is distorted, $i = 1,\dots,4$.

(a) Describe the sample space for this experiment.

(b) Are the $A_i's$ mutually exclusive?

Describe the outcomes in each of the following events:

(c) A_1

(d) $\overline{A_1}$

(e) $A_1 \cap A_2 \cap A_3 \cap A_4$

(f) $(A_1 \cap A_2) \cup (A_3 \cap A_4)$

Solution:

(a) $\Omega = \{0,1\}^4 = \{0000, 0001, 0010, 0011, \ldots, 1110, 1111\}$

(b) No. One indirect way to argue this is to see that $|A_i| = 8$. If they were mutually exclusive then it must hold that $|\Omega| > 32$, but actually $|\Omega| = 16$.

(c) $A_1 = \{1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}$

(d) $\overline{A}_1 = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111\}$

(e) $A_1 \cap A_2 \cap A_3 \cap A_4 = \{1111\}$

(f) $(A_1 \cap A_2) \cup (A_3 \cap A_4) = \{1100, 1101, 1110, 1111, 0011, 0111, 1011\}$

Question 3 - Basic simulation with Julia

Errata Notes:

- The code given after (e) is fundamemtally flawed! There is much to learn from this (common) mistake, made by the lecturer! See description.

The sample space of a random experiment is $\{a, b, c, d, e\}$ with probabilities $0.1, 0.1, 0.2, 0.4, 0.2$ respectively. Let A denote the event $\{a, b, c\}$, and let B denote the event $\{c, d, e\}$. Determine the following:

(a) $P(A)$

(b) $P(B)$

(c) $P(\overline{A})$

(d) $P(A \cup B)$

(e) $P(A \cap B)$

The Julia code below generates 1000 samples with values "1", "2", "3" or "4" based on the weights, $0.1, 0.6, 0.2$ and 0.1 respectively.

```
using StatsBase
w = weights([0.1, 0.6, 0.2, 0.1])

#A proportion of the number of samples equalling either 1 or 3
prop2 = sum([sample(w) == 1 || sample(w) == 3 for _ in 1:1000])/1000
```

(f) Modify this case to simulate the experiment of the question (with $\Omega = \{a, b, c, d, e\}$) using 10^6 replications. Based on the simulation runs, present your estimates for the probabilities in (a)–(e) and compare then to your exact answers for (a)–(e).

Solution:

(a) $P(A) = 0.4$

(b) $P(B) = 0.8$

(c) $P(A^c) = 0.6$

(d) $P(A \cup B) = P(\Omega) = 1$

(e) $P(A \cap B) = P(\{c\}) = 0.2$

(f) To describe the problem with the code, consider random variables (they are taught in the next unit, but slightly easier to describe using this way).

Let W_1 and W_2 be two independent random variables distributed with a pmf described by [0.1, 0.6, 0.2, 0.1].

When we carry out "sample(w)" we are generating such a random variable. Then carrying out "sample(w) == 1 || sample(w) == 3" is equivlent to looking at the event, $W_1 = 1 \quad or \quad W_2 = 3$. This is VERY DIFFERENT from $W_1 == 1 \quad or \quad W_1 == 3$ since it uses two indpendent random variables and not the same one.

To see this:
$P(W_1 = 1 \quad or \quad W_2 = 3) = P(\{W_1 = 1\} \cup \{W_2 = 3\}) = P(\{W_1 = 1\}) + P(\{W_2 = 3\}) - P(\{$

Now since they are independent, this means that
$P(\{W_1 = 1\} \cap \{W_2 = 3\}) = P(\{W_1 = 1\})P(\{W_2 = 3\})$

Hence, $P(W_1 = 1 \quad or \quad W_2 = 3) = 0.1 + 0.2 + 0.1 * 0.2 = 0.28$

And this is indeed what the simulation gives:

In [5]:

```
using StatsBase
w = weights([0.1, 0.6, 0.2, 0.1])

prop2 = sum([sample(w) == 1 || sample(w) == 3 for _ in 1:10^6])/10^6
```

Out[5]:

```
0.279963
```

The way to fix this is to use only one copy of the random variable. For example:

In [3]:

```
using StatsBase
w = weights([0.1, 0.6, 0.2, 0.1])

function eventHolds(wInstance)
    wInstance == 1 || wInstance == 3
end

prop2 = sum([eventHolds(sample(w)) for _ in 1:10^6])/10^6
```

Out[3]:

```
0.299981
```

Now carrying this for (a)--(e):

In [11]:

```
# (a)
function eventHolds(wInstance)
    wInstance <= 3
end
w = weights([0.1, 0.1, 0.2, 0.4, 0.2])
sum([eventHolds(sample(w)) for _ in 1:10^6])/10^6
```

```
WARNING: Method definition eventHolds(Any) in module Main at In[10]:3 over
written at In[11]:3.
```

Out[11]:

```
0.399126
```

In [12]:

```
# (b)
function eventHolds(wInstance)
    wInstance >= 3
end
w = weights([0.1, 0.1, 0.2, 0.4, 0.2])
sum([eventHolds(sample(w)) for _ in 1:10^6])/10^6
```

```
WARNING: Method definition eventHolds(Any) in module Main at In[11]:3 over
written at In[12]:3.
```

Out[12]:

```
0.799873
```

In [13]:

```
# (c)
function eventHolds(wInstance)
    wInstance > 3
end
w = weights([0.1, 0.1, 0.2, 0.4, 0.2])
sum([eventHolds(sample(w)) for _ in 1:10^6])/10^6
```

```
WARNING: Method definition eventHolds(Any) in module Main at In[12]:3 over
written at In[13]:3.
```

Out[13]:

```
0.600057
```

In [18]:

```
# (d)
function eventHolds(wInstance)
    true
end
w = weights([0.1, 0.1, 0.2, 0.4, 0.2])
sum([eventHolds(sample(w)) for _ in 1:10^6])/10^6
```

```
WARNING: Method definition eventHolds(Any) in module Main at In[17]:3 over
written at In[18]:3.
```

Out[18]:

```
1.0
```

In [19]:

```
# (e)
function eventHolds(wInstance)
    wInstance == 3
end
w = weights([0.1, 0.1, 0.2, 0.4, 0.2])
sum([eventHolds(sample(w)) for _ in 1:10^6])/10^6
```

```
WARNING: Method definition eventHolds(Any) in module Main at In[18]:3 over
written at In[19]:3.
```

Out[19]:

```
0.200087
```

Question 4 - NiCD battery

In an NiCd battery, a fully charged cell is composed of Nickelic Hydroxide. Nickel is an element that has multiple oxidation states and that is usually found in the following states:

Nickel Charge	Proportions Found
0	0.17
+2	0.35
+3	0.33
+4	0.15

For both of these items, formulate the events as sets as you present your answer.

(a) What is the probability that a cell has at least one of the positive nickel-charged options?

(b) What is the probability that a cell is not composed of a positive nickel charge greater than +3?

Solution:

(a) $P(\{+2,+3,+4\}) = 1 - P(\{0\}) = 1 - 0.17 = 0.83$

(b) $P(\overline{\{+4\}}) = 1 - P(\{+4\}) = 1 - 0.15 = 0.85$

Question 5 - Hacking the NSA

Errata Notes:

- The code given after (d) is (again) flawed! Items (e) and (f) are cancelled.

A computer system uses passwords that contain exactly eight characters, and each character is one of 26 lower-case letters (a-z) or 26 upper-case letters (A-Z) or 10 integers (0-9). Let Ω denote the set of all possible passwords, and let A and B denote the events that consist of passwords with only letters or only integers, respectively. Suppose that all passwords Ω are equally likely. Determine the probability of each of the following:

(a) A

(b) B

(c) A password contains at least 1 integer.

(d) A password contains exactly 2 integers.

The following Julia code generates 100 random passwords and counts how many of them contain 1 or less lower case letters.

```
allPossibleChars = ['a':'z';'A':'Z';'0':'9']
pasSamples = [string(rand(allPossibleChars,8)) for _ in 1:100]

#Returns the number of lower case characters in the string str
function numLowerCaseChars(str)
    sum([contains(str,string(ch)) for ch in 'a':'z'])
end

# The proportion of passwords with 1 or less lower case characters
proportion = sum([numLowerCaseChars(pw)<=1  for pw in pasSamples])/100
```

(e) In your view, are 100 passwords sufficient for obtaining a sensible estimate for the event of having 1 or less lower case characters? Modify the code to obtain a more accurate estimate.

(f) Modify the code to obtain estimates for the probabilities of the events in (a)–(d). Compare with your theoretical results.

Solution:

$\Omega = \big(\{a,\dots,z\} \cup \{A,\dots,Z\} \cup \{0,\dots,9\}\big)^8$ and hence $|\Omega| = 62^8$

In [2]:

```
62^8
```

Out[2]:

```
218340105584896
```

(a) $|A| = 52^8, P(A) = \frac{|A|}{|\Omega|} = \left(\frac{52}{62}\right)^8$

In [3]:

```
(52/62)^8
```

Out[3]:

```
0.24484612384081483
```

(b) $|B| = 10^8$ so $P(B) = \left(\frac{10}{62}\right)^8$

In [4]:

```
(10/62)^8
```

Out[4]:

```
4.5800106092335626e-7
```

(c) $C = \overline{A}$. $P(C) = 1 - P(A) = 1 - \left(\frac{52}{62}\right)^8$

In [5]:

```
1- (52/62)^8
```

Out[5]:

```
0.7551538761591852
```

(d) $D \equiv$ A password has exactly 2 integers.

$D_L \equiv$ The two integers are in locations L where L is one of the $\binom{8}{2} = 8!/(6!2!) = 28$ possible locations

Note that the events D_L are mutually exclusive and

$D = D_1 \cup D_2 \cup \ldots \cup D_{28}$

So $|D| = |D_1| + \ldots + |D_{28}| = 28\,|D_L|$ (same for all L)

Now $|D_L| = 10^2 \cdot 52^6$ (two digits and 6 letters)

$P(D) = \frac{|D|}{|\Omega|} = \frac{28 \cdot 10^2 \cdot 52^6}{62^8}$

In [6]:

```
(28*10^2*52^6)/(62^8)
```

Out[6]:

```
0.2535388856339797
```

Question 6 - Cast aluminium

Samples of a cast aluminium part are classified on the basis of surface finish (in microinches) and length measurements. The results of 100 parts are summarized as follows:

		Length	
		Excellent	Good
Surface	Excellent	80	2
Finish	Good	10	8

Let A denote the event that a sample has excellent surface finish, and let B denote the event that a sample has excellent length. Determine:

(a) $P(A)$

(b) $P(B)$

(c) $P(A \mid B)$

(d) $P(B \mid A)$

(e) If the selected part has excellent surface finish, what is the probability that the length is excellent?

(f) If the selected part has good length, what is the probability that the surface finish is excellent?

Solution:

(a) $P(A) = 0.82$

(b) $P(B) = 0.9$

(c) $P(A \mid B) = P(AB)/P(B) = 0.8/0.9 = 0.8888$

(d) $P(B \mid A) = P(AB)/P(A) = 0.8/0.82 = 0.9756$

(e) This is just (d)

(f) $P(A \mid \overline{B}) = 0.2$

Question 7 - Cotton fabric and nylon fabric

Suppose 2% of cotton fabric rolls and 3% of nylon fabric rolls contain flaws. Of the rolls used by a manufacture, 70% are cotton and 30% nylon. What is the probability that a randomly selected roll used by the manufacturers contain flaws?

Solution:

By the law of total probability:

P(flaws) = P(flaws | cotton) P(cotton) + P(flaws | nylon) P(nylon) = $0.02 \times 0.7 + 0.03 \times 0.3$

In [7]:

```
0.02*0.7+0.03*.3
```

Out[7]:

```
0.023
```

Question 8 - Computer keyboard failure

Computer keyboard failures are due to faulty electrical connects (12%) or mechanical defects (88%). Mechanical defects are related to loose keys (27%) or improper assembly (73%). Electrical connect defects are caused by defective wires (35%), improper connections (13%), or poorly welded wires (52%).

(a) Find the probability that a failure is due to loose keys.

(b) Find the probability that a failure is due to improperly connected or poorly welded wires.

Solution:

(a) P(fail due to loose keys) = P(mechanical defect | loose key) * P(loose key) = 0.27 $\times$ 0.88

In [8]:

```
0.27*0.88
```

Out[8]:

```
0.2376
```

(b) P(fail due to improperly connected OR fail due to poorley welded) = P(improper connection | elect) *P(elect) + P(poorley welded | elect)* P(elect) = (0.13+0.52) $\times$ 0.12

In [10]:

```
(0.13+0.52)*0.12
```

Out[10]:

```
0.078
```

In [ ]:

In [ ]:

In [ ]: