Statistics Assignment 2

BM Corser

January 26, 2017

1. Let $R = (280, 155, 329, 140, 307, 116, 202, 262, 130, 131, 187, 187, 292, 83, 207, 197, 134, 294, 163, 217)$

   (i)

$$\begin{aligned}
\bar{x} &= \frac{1}{n} \cdot \sum_{i=0}^{n} R_i \\
&= \frac{1}{20} \cdot (280 + 155 + 329 + 140 + 307 + 116 + 202 + 262 + 130 + \ \cdots + R_n) \\
&= 200.65
\end{aligned}$$

   (ii)

$$\begin{aligned}
\sigma^2 &= \frac{1}{n-1} \cdot \sum_{i=0}^{n} (\bar{x} - r)^2 \\
&= \frac{1}{19} \cdot ((200.65 - 280)^2 + (200.65 - 155)^2 + \ \dots) \\
&= \frac{1}{19} \cdot ((-79.35)^2 + 45.65^2 + \ \dots) \\
&\approx 5161.60789
\end{aligned}$$

   (iii) Python code:

```
import math
import pandas as pd

DATA = [
    280, 155, 329, 140, 307,
    116, 202, 262, 130, 131,
    187, 187, 292, 83, 207,
    197, 134, 294, 163, 217
]

series = pd.Series(DATA)
print("Sample mean: {0}".format(series.mean()))
print("Sample variance: {0}".format(series.var()))
```

Output:

```
Sample mean: 200.65
Sample variance: 5161.607894736842
```

2. (i) $\Pr(B^c) = 1 - \Pr(B) = 0.85$
   (ii) $\Pr(A \cap B) = \Pr(B) \cdot \Pr(A|B) = 0.15 \times 0.67 = 0.1139$
   (iii) $\Pr(A \cap B^c) = \Pr(A) - \Pr(A \cap B) = 0.3 - 0.1139 = 0.1861$
   (iv) $\Pr(A|B^c) = \frac{\Pr(A \cap B^c)}{\Pr(B^c)} = \frac{0.1861}{0.85} = 0.21894117647058822$
   (v) The events are not independent, because $\Pr(A|B) \neq \Pr(A)$
   (vi) The events are not mutually exclusive, because $\Pr(A \cap B) \neq 0$

3. (i) $\binom{3000}{1} \cdot 0.002^1 \cdot 0.998^{2999} = 0.0148130528123\ldots$

   Python code:

```
from scipy.stats import binom
n, p = 3000, 0.002
dist = binom(n, p)
print(dist.pmf(1))
```

   Output:

```
0.0148130528124
```

   (ii) If an event happens r times in n samples the probability it happens is $\frac{r}{n}$. In the case where this event either happened an amount of 0 (ie. it didn't happen) or an amount of 1 (ie. it completely happened) for a given sample, the mean is therefore equivalent to the probability because $\frac{1}{n}(r \cdot 1 + (n - r) \cdot 0) = \frac{r}{n}$ so in this case we may use a Poisson distribution with $\mu = 0.002$. Hence, according to the Poisson estimation, the probabilty that one dog gets sick is

$$\frac{0.002^1}{e^{0.002} \cdot 1!} = 0.00199600\ldots$$

   Python code:

```
from scipy.stats import poisson
mu = 0.002
dist = poisson(mu)
print(dist.pmf(1))
```

   Output:

```
0.00199600399733
```

(iii) I'm not entirely sure how to scale a Normal distribution based on probability of a discrete random variable instead of μ and σ^2 on continuous data, but going by the summary notes the μ value from a Poisson distribution can be used as μ and σ^2 in $N(\mu, \sigma^2)$, hence let $\mu = \sigma^2 = 0.002$ in

$$f(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

then

$$\begin{aligned} f(1) &= \frac{1}{0.004\pi} \exp\left(-\frac{(1-0.002)^2}{0.004}\right) \\ &= 5.76806103956629468683242514 3... \times 10^{-107} \end{aligned}$$

Python code:

```
from scipy.stats import norm
mu = 0.002
dist = norm(mu, mu)
print(dist.pdf(1))
```

Output:

```
0.0
```

Where presumabley the float representation in the Python library I used wasn't accurate enough to show this number.

4. $\Pr(X \leq r)$ for a Poisson distribution with $\mu = 2$ is given by

$$F_r = \sum_{i=0}^{r} e^{-2} \frac{2^i}{i!},$$

so evaulating for $r = 2$

$$\begin{aligned} F_2 = \sum_{i=0}^{2} e^{-2} \frac{2^i}{i!} &= e^{-2} \frac{2^0}{0!} + e^{-2} \frac{2^1}{1!} + e^{-2} \frac{2^2}{2!} \\ &= e^{-2} + 2e^{-2} + 2e^{-2} \\ &= 5e^{-2}. \end{aligned}$$

As such, the probability two or fewer items will be sold in a day is $5e^{-2}$. The probability that F_r will occur on 7 successive is $(F_r)^7$, which evaluating for $r = 2$, as before, $\left(5e^{-2}\right)^7 = 5^7 e^{-14} < 0.95$. We can write a Python program to try increasing values for r until we get a result $(F_r)^7 > 0.95$.

Python code:

```
from scipy.stats import poisson
from numpy import power
mu = 2
dist = poisson(mu)
Fr7 = 0
r = 2
while Fr7 < 0.95:                   # <-
    r = r + 1                       #    this is a loop
    Fr7 = power(dist.cdf(r), 7)     # <-
print("F_{0} gives {1}".format(r, Fr7))
```

Output:

```
F_6 gives 0.9686917775126511
```

Therefore, in order to have 96.8% probability of not running out of stock, the salesperson will have to buy 7 days $\times$ 6 items = 42.

Because the above assumes that the same number of items will need to be bought each day, it gives us a kind of "upper bound". Let's see if we can shave an item off and still achieve our desired probability.

Python code:

```
from scipy.stats import poisson
from numpy import power, product
mu = 2
dist = poisson(mu)
for r in range(6):
    result = product([power(dist.cdf(6), 6), dist.cdf(r)])
    print("(F_6)^6 * F_{0} gives {1}".format(r, result))
```

Output:

```
(F_6)^6 * F_0 gives 0.13169525676153868
(F_6)^6 * F_1 gives 0.39508577028461594
(F_6)^6 * F_2 gives 0.6584762838076935
(F_6)^6 * F_3 gives 0.8340699594897448
(F_6)^6 * F_4 gives 0.9218667973307707
(F_6)^6 * F_5 gives 0.956985532467181
```

We can see here that $(F_6)^6 \cdot F_5$ also results in a probability > 0.95, so the total number of items the salesperson needs to buy is $6 \times 6 + 5 = 41$.

5. $X \sim N(68, 10)$

   (i) $1 - \Phi(75) = 0.241964$

   (ii) $\Phi(75) - \Phi(70) = 0.758036 - 0.57926 = 0.178776$