Solutions for Assignment Unit 6 -- MATH1280

(Element 1, 5-Scale Excellent/Very Poor) (Rate the answer on a 5-point scale. The correct answer gets the maximum number of stars**)**

a) A quartile is one of the numbers that divides a range of numbers into four equal parts.

b) A quantile is a more generic version of a quartile: a quantile can divide a range of numbers into an arbitrary number of pieces. Additional info, not required for grading: In R, a quantile is a value from 0 to 1 that represents a portion of a distribution. If you enter a quantile of .05 in qnorm(.05, mean=0, sd=1), then the .05 says that you want the 5th percentile (the value from the distribution below which 5% of the values would occur.

c) The arguments to the `pbinom` function are:

1. quantile,
2. size = the number of experiments that are repeated,
3. prob = the probability of success,
4. lower.tail = a flag indicating whether to return the probability below or above that specified q value,
5. log.p = a value that, if set, returns the log of the answer as a means to reduce loss of precision for very small numbers

(Element 2, 4-point scale) (Rate the answer on a 4-point scale. The correct answer gets the maximum number of stars**)**

probability of getting 1 head: ___**0.25**__
probability of getting 2 heads:___**0.375**__
probability of getting 3 heads: ___**0.25**_
probability of getting 4 heads: ___**0.0625**_

```
Explanation:
n = 4
p = .5
> dbinom(0, size=n, prob=p)
[1] 0.0625
> dbinom(1, size=n, prob=p)
[1] 0.25
> dbinom(2, size=n, prob=p)
[1] 0.375
> dbinom(3, size=n, prob=p)
[1] 0.25
> dbinom(4, size=n, prob=p)
[1] 0.0625
```

(Element 3, 5-point scale) (Rate the answer on a 5-point scale. The correct answer gets the maximum number of stars**)**

probability of getting no more than 0 heads: ___.0625__
probability of getting no more than 1 head: ____.3125__
probability of getting no more than 2 heads:____.6875__
probability of getting no more than 3 heads: ____.9375_
probability of getting no more than 4 heads: ____1.000_

Explanation:

```
n = 4
p = .5
> pbinom(0,size=n, prob=p)
[1] 0.0625
> pbinom(1,size=n, prob=p)
[1] 0.3125
> pbinom(2,size=n, prob=p)
[1] 0.6875
> pbinom(3,size=n, prob=p)
[1] 0.9375
> pbinom(4,size=n, prob=p)
[1] 1
```

(Element 4, 2-point scale)(Correct answer = "Yes", incorrect answer = "No"**)**

The answer is: **`TRUE`**.

Explanation: `dbinom(6, size=10, prob=0.5)` would shows the probability of getting exactly 6 successes in 10 trials of an experiment that has a probability of success of .5).

(Element 5, YES/NO scale)

The answer is: **`.50`** (anything that rounds to this number is good).
R commands:

```
> pbinom(1, size=3, prob=.5)

[1] 0.5

>

> # OR you could add the probabilities of getting exactly 0 or 1 successes:

> dbinom(0, size=3, prob=.5) + dbinom(1, size=3, prob=.5)

[1] 0.5

>
```

(Element 6, YES/NO scale) (Correct answer = "Yes", incorrect answer = "No")

The answer is `.81` (or anything that rounds to that -- I show two decimal places on the answer sheet to avoid somebody being marked wrong for being .001 units off).

```
> round(pbinom(3, size=5, prob=.5), 2)

[1] 0.81

>
```

(Element 7, 2-point scale) (Correct answer = "Yes", incorrect answer = "No")

The answer is **`.25 minutes`** (or **`15 seconds`**) or anything that rounds to one of those numbers.

If the rate is 4 customer per minute then lambda is 4 minutes (Yakir, 2011, p. 79). The expectation is 1 divided by the rate (Yakir, 2011, p. 80), which is 1/4 minute or 15 seconds. Note that "lambda" refers to a special function in some languages, so out of habit, I avoid using it as a variable name.

```
> # Minutes  (let “lmb” be my lambda value)

> lmb <- 4

> expectation <- 1 / lmb

> expectation

[1] 0.25

>

> # For seconds, a rate of 4 customer per minute is the same as 4 customer per

> # 60 seconds which is a rate of 1/15 of a customer per second.

> lmb.seconds <- 4/60

> expectation.seconds <- 1 / lmb.seconds

> expectation.seconds

[1] 15
```

(Element 8, 2-point scale)(Correct answer = "Yes", incorrect answer = "No")

The answer is **`.06`** for minutes or **`225`** for seconds. The official units of measure are either *customers-per-minute squared* or *customers-per-second squared*, but “minutes” or “seconds” will suffice.

I use variables to hold the rate, but you can type “5” whereever you see “Q7.lambda.min” in the R commands.

```
> # Minutes

> lmb <- 4
> v <- 1 / (lmb ^ 2)

> # The answer here is in "units per minutes squared", but “minutes” is good
enough

> round(v, 2)

[1] 0.06

>

> # Seconds

> lmb.seconds <- 4 / 60

> v <- 1 / (lmb.seconds ^ 2)

> round(v, 2)

[1] 225

> # The answer here is in "units per second, squared" but “seconds” is good
enough.
```

(Element 9, 2-point scale) (Correct answer = "Yes", incorrect answer = "No")

The answer is: **.64 (or something that rounds to that value; note that I show two decimal places on the answer sheet to avoid somebody being marked wrong for being .001 units off)**.

Explanation:

```
What is the probability that the time interval between customers entering the
store will be less than 15.5 seconds. Be sure to enter values so that everything
is in the same unit of measurement.  Be sure to specify the units of measurement
in your answer.  Round your answer to 2 decimal places.

>

> round(pexp(15.5/60, rate=4), 2) # using minutes

[1] 0.64

> round(pexp(15.5, rate=4/60), 2) # using seconds

[1] 0.64
```

(Element 10, 2-point scale) (Correct answer = "Yes", incorrect answer = "No")

a) The answer is: **.42 (or anything that rounds to that value)**.

b) Any answer that is similar to one of the following three examples is correct. It is usually easiest to keep the same units of measure as in the original problem, which was minutes. The question was about seconds, so I converted them to minutes by dividing by 60. There is also an example of using decimals instead of fractions or expressing everything in terms of seconds, but the all produce the same answer:

```
i> pexp(40.2/60, rate=4) - pexp(10.7/60, rate=4)
[1] 0.421445
>
> pexp(.67, rate=4) - pexp(0.1783333, rate=4)
[1] 0.421445
>
> pexp(40.2, rate=4/60) - pexp(10.7, rate=4/60) # using seconds
[1] 0.421445
>
>
```

(Element 11, 2-point scale) (Correct answer = "Yes", incorrect answer = "No"**)**

a) Any answer is **.75** minutes (or anything that rounds to that value) OR to **44.94** seconds (or anything that rounds to that value).

```
> round(qexp(.95, rate=4), 2) # minutes

[1] 0.75

> round(qexp(.95, rate=4/60), 2) # seconds

[1] 44.94
```

(Element 12, 2-point scale) (Correct answer = "Yes", incorrect answer = "No"**)**

Read the following answer and see if the student provided the main idea of being less than 12 (it is OK to say "less than" or "less than or equal to" for a continuous distribution like this) in an exponential distribution with a rate of 3. Here is an example answer:

> **The R command pexp(1.2, rate=3) tells you the estimated probability of randomly selecting a value less than or equal to 1.2 from an exponential distribution that has a rate of 3. Optional detail: because the exponential distribution is continuous there is no difference between being "less than or equal to 1.2" and "less than 1.2."**

(Element 13, 2-point scale) (Correct answer = "Yes", incorrect answer = "No"**)**

Answer **.25 (or anything that rounds to that value; note that I show two decimal places on the answer sheet to avoid somebody being marked wrong for being .001 units off).**

```
>  round(1 - pnorm(9, mean=7, sd=3), 2)

[1] 0.25
```

(Element 14, 2-point scale) (Correct answer = "Yes", incorrect answer = "No"**)**

Answer: **1.75 (or anything that rounds to that value; note that I show two decimal places on the answer sheet to avoid somebody being marked wrong for being .001 units off).**

```
> round(qnorm(.04, mean=7, sd=3), 2)

[1] 1.75

>
```

(Element 15, 2-point scale) (Correct answer = "Yes", incorrect answer = "No"**)**

Answer: **.16 (or anything that rounds to that value; note that I show two decimal places on the answer sheet to avoid somebody being marked wrong for being .001 units off).**

```
> round(1 - pnorm(1, mean=0, sd=1), 2)
[1] 0.16
>
>  round(1 - pnorm(7+3, mean=7, sd=3), 2)
[1] 0.16
```