Armando Tejeda

HW 9 Solution

10/26/22

Question 12.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a design of experiments approach would be appropriate.

Solution:

Design for experiment would be very useful for, say, a fast food restaurant in a well traveled area. The manager could use DOE to come up with a series of "what-if" scenarios for the restaurant to maximize customer satisfaction and not miss any business due to long wait lines or short lunch breaks.

Question 12.2

To determine the value of 10 different yes/no features to the market value of a house (large yard, solar roof, etc.), a real estate agent plans to survey 50 potential buyers, showing a fictitious house with different combinations of features. To reduce the survey size, the agent wants to show just 16 fictitious houses. Use R's FrF2 function (in the FrF2 package) to find a fractional factorial design for this experiment: what set of features should each of the 16 fictitious houses have? Note: the output of FrF2 is "1" (include) or "-1" (don't include) for each feature.

Solution:

Below is the implementation of the fractional factorial design for 16 open houses based on 10 features. Output below shows which house should have which features to show to prospective house buyers.

In [ ]:

```r
library(tidyverse)
library(FrF2)

(houses <- FrF2(
  16, 10,
  factor.names = c(
    'Large Yard', 'Pool', 'Solar Roof', 'Long Driveway', 'Multi Car Garage',
    'Walk-In Closet', 'Man Cave', 'Full Bar', 'Golf Access', 'New Home'
    ),
  default.levels = c('Yes', 'No')
  ) %>%
  as_tibble() %>%
  rownames_to_column('House')
)
```

A tibble: 16 × 11

House	Large.Yard	Pool	Solar.Roof	Long.Driveway	Multi.Car.Garage	Walk.In.Closet	Man.Cave	Fu
<chr>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	<
1	Yes	No	No	No	Yes	Yes	No	
2	Yes	No	Yes	No	Yes	No	Yes	
3	Yes	Yes	Yes	Yes	No	No	No	
4	No	No	Yes	Yes	No	Yes	Yes	
5	Yes	No	Yes	Yes	Yes	No	Yes	
6	Yes	Yes	Yes	No	No	No	No	
7	No	Yes	No	No	Yes	No	Yes	
8	No	Yes	No	Yes	Yes	No	Yes	
9	No	No	Yes	No	No	Yes	Yes	
10	No	Yes	Yes	No	Yes	Yes	No	
11	Yes	No	No	Yes	Yes	Yes	No	
12	Yes	Yes	No	No	No	Yes	Yes	
13	No	Yes	Yes	Yes	Yes	Yes	No	
14	Yes	Yes	No	Yes	No	Yes	Yes	
15	No	No	No	No	No	No	No	
16	No	No	No	Yes	No	No	No	

Question 13.1

For each of the following distributions, give an example of data that you would expect to follow this distribution (besides the examples already discussed in class). a. Binomial
b. Geometric
c. Poisson
d. Exponential
e. Weibull

Solution:

- Binomial: probability of rolling a dice and getting the number 6 in a given number of trials.
- Geometric: related to above, probability of the number of times a dice is thrown until a 6 is gotten for the first time.
- Poisson: if normally a cafe sees 1 customer per minute on average, then the probability of 10 customers arriving all at once follows a Poisson distribution.
- Exponential: the amoun of time a car battery will last will follow a exponential distribution.

- Weibull: in contrast, how long will it take for my car battery to fail will then follow a Weibull distribution.

Question 13.2

In this problem you, can simulate a simplified airport security system at a busy airport. Passengers arrive according to a Poisson distribution with λ1 = 5 per minute (i.e., mean interarrival rate   1 = 0.2 minutes) to the ID/boarding-pass check queue, where there are several servers who each have exponential service time with mean rate   2 = 0.75 minutes. [Hint: model them as one block that has more than one resource.] After that, the passengers are assigned to the shortest of the several personal-check queues, where they go through the personal scanner (time is uniformly distributed between 0.5 minutes and 1 minute).

Use the Arena software (PC users) or Python with SimPy (PC or Mac users) to build a simulation of the system, and then vary the number of ID/boarding-pass checkers and personal-check queues to determine how many are needed to keep average wait times below 15 minutes. [If you're using SimPy, or if you have access to a non-student version of Arena, you can use λ1 = 50 to simulate a busier airport.]

Solution:

*Below is a few trials with different numbers of ID/boarding-pass-queues and personal-check-queues. Looks like **3** ID/boarding-pass-queues and **2** personal-check-queues will do the job at about 9 minutes on average.*

Solution below but code will be a different attachment.

```
(base) C:\Users\ateje\OneDrive\Desktop\VS Code
Projects>C:/Users/ateje/anaconda3/python.exe "c:/Users/ateje/OneDrive/Desktop/VS Code
Projects/Python-Projects/Airport_simulation.py"
Input # of ID_queues working: 1 Input # of personal_check_queues working: 1 Running
simulation... The average wait time is 32 minutes and 46 seconds.
```

```
(base) C:\Users\ateje\OneDrive\Desktop\VS Code
Projects>C:/Users/ateje/anaconda3/python.exe "c:/Users/ateje/OneDrive/Desktop/VS Code
Projects/Python-Projects/Airport_simulation.py"
Input # of ID_queues working: 2 Input # of personal_check_queues working: 2 Running
simulation... The average wait time is 21 minutes and 1 seconds.
```

```
(base) C:\Users\ateje\OneDrive\Desktop\VS Code
Projects>C:/Users/ateje/anaconda3/python.exe "c:/Users/ateje/OneDrive/Desktop/VS Code
Projects/Python-Projects/Airport_simulation.py"
Input # of ID_queues working: 3 Input # of personal_check_queues working: 3 Running
simulation... The average wait time is 9 minutes and 17 seconds.
```

```
(base) C:\Users\ateje\OneDrive\Desktop\VS Code
Projects>C:/Users/ateje/anaconda3/python.exe "c:/Users/ateje/OneDrive/Desktop/VS Code
Projects/Python-Projects/Airport_simulation.py"
Input # of ID_queues working: 2 Input # of personal_check_queues working: 3 Running
simulation... The average wait time is 21 minutes and 1 seconds.
```

```
(base) C:\Users\ateje\OneDrive\Desktop\VS Code
Projects>C:/Users/ateje/anaconda3/python.exe "c:/Users/ateje/OneDrive/Desktop/VS
Code Projects/Python-Projects/Airport_simulation.py"
Input # of ID_queues working: 3 Input # of personal_check_queues working: 2 Running
simulation... The average wait time is 9 minutes and 17 seconds.
```