ISYE 6501-HW 9

Ryan Cherry

2024-03-10

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.

A design of experiments approach would be appropriate to use when a political campaign is determining which version of an ad to run on TV. A/B testing could be used, where one segment of people is shown one version of the ad and then each participant is then asked how likely it is they will vote for the candidate. The same thing would be done with a second version of the ad with a different group of people. Afterwards, the ad that resulted in the higher percentage of respondents who say they would vote for the political candidate would be the one the campaign would send to air on TV.

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.

FrF2 was used to determine which factors would be evaluated on 16 select houses below. The output table is shown below.

```
house_market_features <- FrF2(nruns = 16, nfactors = 10, factor.names =
c('Large Yard', 'Privacy', 'Solar Roof', 'Hottub', 'More than 2 Floors',
'More than 2 Bathrooms', 'Patio', 'Fireplace', 'Screen Porch', 'More than 2
Bedrooms'), default.levels = c(1, -1))

print(house_market_features)

##    Large.Yard Privacy Solar.Roof Hottub More.than.2.Floors
## 1          -1       1          1      1                  1
## 2          -1       1         -1      1                  1
## 3          -1      -1         -1      1                 -1
## 4          -1       1          1     -1                  1
## 5           1      -1          1      1                  1
## 6          -1      -1          1      1                 -1
## 7           1       1          1     -1                 -1
## 8           1      -1          1     -1                  1
```

```
## 9          -1      -1         1     -1                 -1
## 10          1      -1        -1      1                  1
## 11          1       1        -1     -1                 -1
## 12          1       1         1      1                 -1
## 13          1       1        -1      1                 -1
## 14          1      -1        -1     -1                  1
## 15         -1      -1        -1     -1                 -1
## 16         -1       1        -1     -1                  1
##    More.than.2.Bathrooms Patio Fireplace Screen.Porch More.than.2.Bedrooms
## 1                      1    -1         1            1                    1
## 2                     -1     1         1           -1                   -1
## 3                     -1    -1         1            1                    1
## 4                      1    -1        -1           -1                   -1
## 5                     -1     1        -1           -1                    1
## 6                      1     1         1           -1                   -1
## 7                     -1    -1         1           -1                    1
## 8                     -1     1         1            1                   -1
## 9                      1     1        -1            1                    1
## 10                     1    -1        -1            1                   -1
## 11                     1     1         1            1                   -1
## 12                    -1    -1        -1            1                   -1
## 13                     1     1        -1           -1                    1
## 14                     1    -1         1           -1                    1
## 15                    -1    -1        -1           -1                   -1
## 16                    -1     1        -1            1                    1
## class=design, type= FrF2
```

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 I would expect the number of times I make a three-pointer out of 10 tries to follow a binomial distribution, which models the number of successes out of n trials.

b. Geometric I would expect the number of drives before my car breaks down to follow a geometric distribution, which is useful for modeling the number of trials until failure.

c. Poisson I would expect the number of cars that enter a car wash per hour to follow a Poisson distribution, which is useful for modeling queuing systems.

d. Exponential I would expect the time between defective chicken sandwiches processed at Popeyes to follow an exponential distribution, which is useful for measuring time between failures.

e. Weibull I would expect the time until a set of car brakes fails to follow a Weibull distribution, which is used to model time to failure.

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 $\lambda1 = 5$ 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 $\lambda1 = 50$ to simulate a busier airport.]

An airport security system was simulated in Arena starting with two personal check queues with one resource at ID/boarding pass check queue. This resulted in a bottleneck, which merited an increase in the number of personal ID/boarding pass checkers. Several combinations of personal check queues and personal ID/boarding pass checkers were tested before arriving at the optimal combination of six personal check queues and 11 personal ID/boarding pass checkers. The optimal combination was determined based on which one had the shortest throughput time. The combination of six personal check queues and 11 personal ID/boarding pass checkers resulted in a throughput time of under 10 minutes. Unfortunately, this is highly unrealistic because other environmental factors like cost, time of day, people who may need to be searched (which can slow down traffic), or events such as a bomb threat could easily increase throughput time by a substantial amount. Overall, though, Arena proved to be an easy-to-use tool for simulating complex situations, which is helpful in the fast-paced, always changing world today.