Homework_9

Claire Kraft

2024-10-22

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.

I work remote/hybrid. Sometimes i'll drive to campus (downtown). I tend to prefer one route out of 3 possible routes. I can use design of experiments approach to figure out which of the two highway systems is the most efficient. One highway is more direct but gets more traffic as it has other highways funneled into it. The other highway is less direct, shorter, but has less traffic.

Assuming there are no special events or holidays that may change the ordinary traffic patterns, I can test the two highways in over a month. I will leave the house at the exact same time every Tuesday and Thursday and drive the same car. I will alternate each week which highway to drive. I can record my drive times after each trip to see whether one highway is faster than the other.

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.

For house features I googled house features and Zillow [1] had a blog post listing some features that help with premium or selling speed (reducing the time a house is on the market). Those features are bucketed into broader categories: high end cooking, open floor plan, updated kitchen, hardwood floors, large yard, good schools, modern bathrooms, energy efficient, walk-in closets, move in ready.

```
# Install and load FrF2 package if not already done
#install.packages("FrF2")
library(FrF2)
```

```
## Loading required package: DoE.base
```

```
## Loading required package: grid
```

```
## Loading required package: conf.design
```

```
## Registered S3 method overwritten by 'DoE.base':
##   method           from
##   factorize.factor conf.design
```

```
##
## Attaching package: 'DoE.base'
```

```
## The following objects are masked from 'package:stats':
##
##     aov, lm
```

```
## The following object is masked from 'package:graphics':
##
##     plot.design
```

```
## The following object is masked from 'package:base':
##
##     lengths
```

```
# Clear the workspace
rm(list = ls())

# Set seed for reproducibility
set.seed(1)

# Define features based on Zillow's research
features <- c("high_end_cooking", "open_floor_plan", "updated_kitchen", "hardwood_floors", "larg
e_yard", "good_schools", "modern_bathrooms", "energy_efficient", "walkin_closets", "move_in_read
y")

# Generate the fractional factorial design
house_features_design <- FrF2(nruns = 16, factor.names = features)
house_features_design
```

```
##    high_end_cooking open_floor_plan updated_kitchen hardwood_floors large_yard
## 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
##    good_schools modern_bathrooms energy_efficient walkin_closets move_in_ready
## 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 b. Geometric c. Poisson d. Exponential e. Weibull

a. Binomial : How many flashes for 10 various climbs that are at my max grade (v5)? Flashes in climbing mean you successfully ascend a climbing problem, first try, without needing extra attempts.

b. Geometric : How many hours of finger strength training before leveling up on another climbing grade (v6).

c. Poisson : How many rubiks cubes can be solved in an hour?

d. Exponential : Time between boarding two consecutive zones on a big international aircraft. Most planes for international flights (Airbus A380 and Boeing 747) have double deckers [5].

e. Weibull : Keyboard clicks before the key breaks

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.]

Please look at my `airport_security_sim.py` and `airport_security_sim_output.png` for the code and outputs. I will summarize the methodology and interpretation here.

The evidence seems to suggest a single queue filtering to ID checking stations and scanners is more efficient than two queues dispersing to the ID checking stations and scanners. Each step of the airport security routine is time stamped. The steps are 500 passengers arrive, stand in line for ID checking (2), stand in line again for scanning (2), then exit. The number of ID checkers and scanners do not change for either simulations (simple or complex). The simple simulation has every passenger in one queue and they are sent to the most available ID checking and scanning lines. The complex simulations follows the same flow except there are two queues.

The average wait times and full routine times are shorter by a few seconds in the single queue compared to dual queues. Perhaps funneling passengers from a single line is easier to manage than two lines. A single queuing system minimizes the wait times as it balances the line across all resources without concerning the calculations of another line.

Like in SQL one-to-many relationship vs many-to-many relationships. It's simpler to manage an one-to-many relationship than many-to-many and yields better performance with less complexity and more compute efficiency.

References:

[1] Tucker, J. (2019, April 19). Listing Features That Sell: If You've Got These, Flaunt Them. Zillow. https://www.zillow.com/research/listing-features-that-sell-23814/ (https://www.zillow.com/research/listing-features-that-sell-23814/)

[2] Microsoft. (2024). Copilot: AI companion. Accessed 2024-10-15. Prompt: 'Add comments to my lines of code.' Generated using https://www.microsoft.com (https://www.microsoft.com)."

[3] Microsoft. (2024). Copilot: AI companion. Accessed 2024-10-15. Prompt: 'using my single queue simulation code as inpiration to generate similar code but for two queue system.' Generated using https://www.microsoft.com (https://www.microsoft.com)."

[4] Microsoft. (2024). Copilot: AI companion. Accessed 2024-10-15. Prompt: 'Look at my code, do not modify the code. Just understand the code and create a markeddown documentation for the code.' Generated using https://www.microsoft.com (https://www.microsoft.com)."

[5] Kylie, N. (2023, September 25). Double-Decker Die-Hards: The Airlines That Have Flown Both The Airbus A380 & The Boeing 747. Simple Flying. https://simpleflying.com/airlines-with-both-airbus-a380-and-boeing-747/ (https://simpleflying.com/airlines-with-both-airbus-a380-and-boeing-747/)