DEMO 9 - DESIGN OF EXPERIMENTS AND SIMULATORS

QUESTION 1

One of the examples where design of experiments practice would carry significant value, is to test the safety of a newly produced vehicle by a car company. Every new model designed by a car company has a certain set of cars - say 5% - that undergo pre-release testing through certain tests to determine the car's weaknesses. This 5% may further be divided into 10 parts to assess interactive risks by varying the combinations and parameters of these standard tests. These standard tests are usually -

1. Frontal Crash Test
2. Side Barrier Crash Test
3. Side Pole Crash Test
4. Internal Passenger Protection Tests - seat belt impact absorption capacity, inflatable rescue packs, etc.
5. Interactive Tests - For cases where one or more critical weaknesses compound the issue

This is a highly needed design of experiments, since the failure or negligence in design of even a small cross-examination can result in the death and injury of human life, and in lawsuits worth expensive sums of money, car recalls, and business failure.

QUESTION 2

Results of simulation of 16 combinations of 10 features for 16 fictitious houses -

Where the first column is house and the rest of features are as follows -

'Patio ', 'Pool ', 'Garden ', 'Furnished Study ', 'Car Garage ','Inbuilt Cupboards ', 'Bay Window ', 'Wooden Floors ', 'Ventilated Kitchen ', 'Ventilated Bathrooms '

CODE FOR 2

In [ ]:
```
#code - 12.2

library(dplyr)
library(FrF2)
library(tidyverse)
library(magrittr)
library(tidyr)
library(fpp2)
library(caTools)
library(reshape2)
library(psych)
library(fBasics)
library(kableExtra)

set.seed(512)

houses <- FrF2(
  nruns=16, nfactors=10,
  factor.names = c(
    'Patio ', 'Pool ', 'Garden ', 'Furnished Study ', 'Car Garage ',
    'Inbuilt Cupboards ', 'Bay Window ', 'Wooden Floors ', 'Ventilated Kitchen ', 'Ventilated Bathrooms '
    ), #ten features
  default.levels = c('Yes', 'No')
  ) %>%
  as_tibble() %>%
  rownames_to_column('House ')


kable(houses)
```

QUESTION 3

1. Binomial - You roll two dice for a 12 to win the jackpot. You either roll a twelve and win, or you don't.

1. Geometric - You teach a player how to play a game over and over again until he naturally gets the hand of it and finally wins his first game. The number of games he failed before his first win would represent the geometric distribution.

1. Poisson - The number of movie watchers arriving at the popcorn counter at the theater a few hours before the movie starts. I've mentioned few hours because, in the hour right before the movie, the traffic before the show would suddenly go up due to people wanting fresh popcorn for the movie - this might nor be a purely poisson distribution.

1. Exponential - The amount of time a harmful radioactive substance needs to decay to half-life before it becomes ineffective upon exposure.

1. Weibull - Product cycles are often classified into an infant mortality period with a decreasing failure rate followed by a normal life period with a low, relatively constant failure rate and concluding with a wear-out period that exhibits an increasing failure rate. A good example of this is how a gaming company I originally worked at would time their product releases in such a manner that the tail of one distribution overlapped with the head of the next distribution , making sure players were always hooked to their products and the rigth products were being shipped out at the right times.

QUESTION 4

This is the poisson experiment with a busy airport of 50 people in a single queue and 3 service counters with given waiting and service times, having the average waiting time of 3.548 mins, and here is the flow of customers at the airport as follows -

```
0.0000 Customer00: Here I am
0.0000 Customer00: Waited  0.000
0.0038 Customer01: Here I am
0.0038 Customer01: Waited  0.000
0.0841 Customer02: Here I am
0.4021 Customer01: Finished
0.4021 Customer02: Waited  0.318
0.4339 Customer00: Finished
0.6088 Customer03: Here I am
0.6088 Customer03: Waited  0.000
0.6460 Customer04: Here I am
0.6469 Customer05: Here I am
0.7922 Customer02: Finished
0.7922 Customer04: Waited  0.146
0.8160 Customer03: Finished
0.8160 Customer05: Waited  0.169
0.9234 Customer06: Here I am
1.0706 Customer07: Here I am
1.2124 Customer05: Finished
1.2124 Customer06: Waited  0.289
1.3106 Customer04: Finished
1.3106 Customer07: Waited  0.240
1.3196 Customer08: Here I am
1.5399 Customer07: Finished
1.5399 Customer08: Waited  0.220
1.6149 Customer08: Finished
1.6833 Customer09: Here I am
1.6833 Customer09: Waited  0.000
2.0670 Customer10: Here I am
2.0713 Customer11: Here I am
2.0854 Customer12: Here I am
2.1321 Customer13: Here I am
2.3833 Customer14: Here I am
2.5028 Customer15: Here I am
2.5057 Customer09: Finished
2.5057 Customer10: Waited  0.439
2.6388 Customer16: Here I am
2.6433 Customer17: Here I am
2.8919 Customer06: Finished
2.8919 Customer11: Waited  0.821
3.0075 Customer18: Here I am
3.1943 Customer10: Finished
3.1943 Customer12: Waited  1.109
3.3026 Customer19: Here I am
3.5840 Customer20: Here I am
3.6397 Customer12: Finished
3.6397 Customer13: Waited  1.508
3.7313 Customer21: Here I am
4.0515 Customer13: Finished
4.0515 Customer14: Waited  1.668
4.2370 Customer14: Finished
4.2370 Customer15: Waited  1.734
4.2888 Customer11: Finished
4.2888 Customer16: Waited  1.650
4.3727 Customer15: Finished
4.3727 Customer17: Waited  1.729
4.3992 Customer22: Here I am
4.4149 Customer23: Here I am
4.4335 Customer17: Finished
4.4335 Customer18: Waited  1.426
4.5014 Customer24: Here I am
4.7121 Customer25: Here I am
```

```
 4.7468 Customer26: Here I am
 4.7686 Customer16: Finished
 4.7686 Customer19: Waited  1.466
 5.1195 Customer27: Here I am
 5.4999 Customer28: Here I am
 5.6557 Customer29: Here I am
 5.7069 Customer30: Here I am
 5.9186 Customer31: Here I am
 5.9389 Customer32: Here I am
 6.0424 Customer18: Finished
 6.0424 Customer20: Waited  2.458
 6.4636 Customer33: Here I am
 6.9280 Customer19: Finished
 6.9280 Customer21: Waited  3.197
 7.0124 Customer20: Finished
 7.0124 Customer22: Waited  2.613
 7.1794 Customer34: Here I am
 7.3646 Customer35: Here I am
 7.4574 Customer36: Here I am
 7.4963 Customer37: Here I am
 7.8577 Customer38: Here I am
 7.9263 Customer39: Here I am
 8.1191 Customer21: Finished
 8.1191 Customer23: Waited  3.704
 8.4178 Customer40: Here I am
 8.5075 Customer22: Finished
 8.5075 Customer24: Waited  4.006
 8.8600 Customer41: Here I am
 8.9135 Customer42: Here I am
 9.1436 Customer43: Here I am
 9.4431 Customer44: Here I am
 9.4948 Customer45: Here I am
 9.6663 Customer23: Finished
 9.6663 Customer25: Waited  4.954
 9.8096 Customer46: Here I am
 9.8233 Customer47: Here I am
 9.8852 Customer48: Here I am
 9.9020 Customer49: Here I am
10.2030 Customer25: Finished
10.2030 Customer26: Waited  5.456
10.2058 Customer24: Finished
10.2058 Customer27: Waited  5.086
10.8179 Customer26: Finished
10.8179 Customer28: Waited  5.318
11.0253 Customer28: Finished
11.0253 Customer29: Waited  5.370
11.7490 Customer29: Finished
11.7490 Customer30: Waited  6.042
11.8384 Customer27: Finished
11.8384 Customer31: Waited  5.920
12.3938 Customer30: Finished
12.3938 Customer32: Waited  6.455
12.4307 Customer32: Finished
12.4307 Customer33: Waited  5.967
12.5137 Customer33: Finished
12.5137 Customer34: Waited  5.334
12.9314 Customer34: Finished
12.9314 Customer35: Waited  5.567
13.1364 Customer35: Finished
13.1364 Customer36: Waited  5.679
13.2700 Customer31: Finished
13.2700 Customer37: Waited  5.774
13.3360 Customer37: Finished
13.3360 Customer38: Waited  5.478
13.6867 Customer36: Finished
13.6867 Customer39: Waited  5.760
13.7411 Customer39: Finished
13.7411 Customer40: Waited  5.323
13.9620 Customer40: Finished
13.9620 Customer41: Waited  5.102
14.4582 Customer41: Finished
14.4582 Customer42: Waited  5.545
14.5664 Customer38: Finished
14.5664 Customer43: Waited  5.423
14.8173 Customer43: Finished
14.8173 Customer44: Waited  5.374
16.1066 Customer42: Finished
16.1066 Customer45: Waited  6.612
16.5209 Customer45: Finished
```

```
  16.5209 Customer46: Waited  6.711
  16.7082 Customer44: Finished
  16.7082 Customer47: Waited  6.885
  17.3803 Customer46: Finished
  17.3803 Customer48: Waited  7.495
  17.7370 Customer47: Finished
  17.7370 Customer49: Waited  7.835
  17.8190 Customer48: Finished
  17.9764 Customer49: Finished

 Average wait for  50 checks was 3.548 minutes.
```

CODE FOR 4

In [ ]:

```
#code - 13.2

""" Airport: Parameterized Number of Counters but a Single Queue """
from SimPy.Simulation import *
from random import expovariate, seed

## Model components ------------------------

class Source(Process):
    """ Source generates customers randomly """

    def generate(self,number,interval,resource):
        for i in range(number):
            c = Customer(name = "Customer%02d"%(i,))
            activate(c,c.visit(b=resource))
            t = expovariate(1.0/interval)
            yield hold,self,t

class Customer(Process):
    """ Customer arrives, is served and leaves """

    def visit(self,b):
        arrive = now()
        print("%8.4f %s: Here I am     "%(now(),self.name))
        yield request,self,b
        wait = now()-arrive
        wM.observe(wait) #stored for final averge wait time calculation
        print("%8.4f %s: Waited %6.3f"%(now(),self.name,wait))
        tib = expovariate(1.0/timeInBank)
        yield hold,self,tib
        yield release,self,b
        print("%8.4f %s: Finished      "%(now(),self.name))

## Experiment data -------------------------

maxNumber = 50
maxTime = 100.0 # minutes
timeInBank = 0.75  # service mean, minutes
ARRint = 0.2    # waiting mean, minutes
Nc = 2          # number of ticket counters
theseed = 512

## Model/Experiment   ----------------------

seed(theseed)
k = Resource(capacity=Nc,name="Check-In Counters")
wM = Monitor()
initialize()
s = Source('Source')
activate(s,s.generate(number=maxNumber,interval=ARRint,
                      resource=k),at=0.0)
simulate(until=maxTime)

result = wM.count(),wM.mean()
print("Average wait for %3d checks was %5.3f minutes."% result)
```

THE END----------

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js