Assignment3 with Python and R

```r
# Library for transforming objects from Python to R and back
library(reticulate)
```

```python
import numpy as np;
import simpy
import random;
import settings
from Hospital import Hospital
# In case you want to save data to to json
import json

def printStatistics(hospital, sample_number, hospital_number):
    patients = hospital.patients

    patients_finished = list(filter(lambda p: p.finished, patients))
    n_finished = len(patients_finished)
    n_patients = len(patients)
    mean_blocking_time = hospital.time_operation_theatre_blocked / n_finished
    mean_queue_at_entrance = hospital.total_queue_at_entrance / n_patients
    utilization_rate_of_operation_theatre = hospital.total_time_operating / settings.SIM_TIME

    print("-" * 20)
    print("sample: ", sample_number)
    print("hospital: ", hospital_number)
    print("Patients came: ", n_patients)
    print("Patients got treated: ", n_finished)
    print("Average time operation theatre was blocked: %.3f" % mean_blocking_time)
    print("Average queue at entrance: %.3f" % mean_queue_at_entrance)
    print("Utilization rate of operation theatre: %.3f" % utilization_rate_of_operation_theatre)

    total_throughput_time = sum([p.end_time - p.start_time for p in patients_finished])
    print("Total throughput time %.3f" % (total_throughput_time))
    print("Average throughput time %.3f" % (total_throughput_time / n_patients))

    print("-" * 20)

def get_data(hospital):
    patients = hospital.patients

    patients_finished = list(filter(lambda p: p.finished, patients))
    n_finished = len(patients_finished)
    n_patients = len(patients)
    mean_blocking_time = hospital.time_operation_theatre_blocked / n_finished
    mean_queue_at_entrance = hospital.total_queue_at_entrance / n_patients
    utilization_rate_of_operation_theatre = hospital.total_time_operating / settings.SIM_TIME
```

```
    total_throughput_time = sum([p.end_time - p.start_time for p in patients_finished])

    patients_json = list(map(lambda x: x.to_dict(), patients))

    #"patients": patients_json,
    return {
        "mean_blocking_time": mean_blocking_time,
        "mean_queue_at_entrance": mean_queue_at_entrance,
        "utilization_rate_of_operation_theatre": utilization_rate_of_operation_theatre,
        "total_throughput_time": total_throughput_time
    }
```

Settings

```
# If you want to use values defined in settings.py file, call settings.N_SAMPLES/settings.SIM_TIME
SIM_TIME = 1000
N_SAMPLES = 20
#PREP_ROOMS = [3,3,4]
#REC_ROOMS = [4,5,5]
N_HOSP = len(settings.CONFIGURATIONS)
```

Independent simulations with simPy

```
#initialize results matrix
sim_que = np.zeros( (N_SAMPLES, N_HOSP) )
sim_uti = np.zeros( (N_SAMPLES, N_HOSP) )
#samples are independent, each sample has unique seed (N_SAMPLES*N_HOSP seeds in total)
RANDOM_SEEDS = [*range(N_SAMPLES*N_HOSP)]

for h, config in enumerate(settings.CONFIGURATIONS):
    for sample_i in range(N_SAMPLES):
        random.seed(RANDOM_SEEDS[h*N_SAMPLES + sample_i])
        env = simpy.Environment()
        hospital = Hospital(env, config["n_preparation_rooms"], config["n_recovery_rooms"])
        env.run(until=SIM_TIME)
        sample_data = get_data(hospital)
        sim_que[sample_i, h] = sample_data["mean_queue_at_entrance"]
        sim_uti[sample_i, h] = sample_data["utilization_rate_of_operation_theatre"]

# SAVING TO FILE
#    with open("data.json", mode="w", encoding="utf-8") as f:
#        json.dump(data, f, indent=4)
```

Calculating results with R

Differences between independent simulations

```r
# Transform Python objects to R. This is possible to replace with json data load
sim_que = py$sim_que
sim_uti = py$sim_uti
N_SAMPLES = py$N_SAMPLES
PREP_ROOMS = sapply(py$settings$CONFIGURATIONS, FUN = function(x) x$n_preparation_rooms)
REC_ROOMS = sapply(py$settings$CONFIGURATIONS, FUN = function(x) x$n_recovery_rooms)

# means and standard deviation on the variables
mean_que <- apply(sim_que, 2, mean)
mean_uti <- apply(sim_uti, 2, mean)
sd_que <- apply(sim_que, 2, sd)
sd_uti <- apply(sim_uti, 2, sd)

# Table of descriptive statistics
descriptives <- data.frame(
    PREP_ROOMS = PREP_ROOMS,
    REC_ROOMS = REC_ROOMS,
    MEAN_QUE = mean_que,
    SD_QUE = sd_que,
    MEAN_UTI = mean_uti,
    SD_UTI = sd_uti
    )
# Table of differences in means between hospitals and 95% confidence interval of
#       the difference based on t-distribution.
# Variable average queue
que_95ci <-data.frame(
    DIFFERENCE = c("1-2", "1-3", "2-3"),
    MEAN_D = c(mean_que[1] - mean_que[2],
               mean_que[1] - mean_que[3],
               mean_que[2] - mean_que[3]
               ),
    CI_low = c(t.test(sim_que[,1], sim_que[,2], paired = FALSE)$conf.int[1],
               t.test(sim_que[,1], sim_que[,3], paired = FALSE)$conf.int[1],
               t.test(sim_que[,2], sim_que[,3], paired = FALSE)$conf.int[1]
    ),
    CI_high = c(t.test(sim_que[,1], sim_que[,2], paired = FALSE)$conf.int[2],
               t.test(sim_que[,1], sim_que[,3], paired = FALSE)$conf.int[2],
               t.test(sim_que[,2], sim_que[,3], paired = FALSE)$conf.int[2]
    )
)
# Variable utilization rate
uti_95ci <-data.frame(
    DIFFERENCE = c("1-2", "1-3", "2-3"),
    MEAN_D = c(mean_uti[1] - mean_uti[2],
               mean_uti[1] - mean_uti[3],
               mean_uti[2] - mean_uti[3]
               ),
    CI_low = c(t.test(sim_uti[,1], sim_uti[,2], paired = FALSE)$conf.int[1],
               t.test(sim_uti[,1], sim_uti[,3], paired = FALSE)$conf.int[1],
```

```
              t.test(sim_uti[,2], sim_uti[,3], paired = FALSE)$conf.int[1]
    ),
    CI_high = c(t.test(sim_uti[,1], sim_uti[,2], paired = FALSE)$conf.int[2],
              t.test(sim_uti[,1], sim_uti[,3], paired = FALSE)$conf.int[2],
              t.test(sim_uti[,2], sim_uti[,3], paired = FALSE)$conf.int[2]
    )
)
```

```
## [1] "Descriptive statistics"

##   PREP_ROOMS REC_ROOMS  MEAN_QUE    SD_QUE  MEAN_UTI    SD_UTI
## 1          3         4 0.8322285 0.8730388 0.6196968 0.1640072
## 2          3         5 0.6168620 0.6784140 0.5639448 0.1301752
## 3          4         5 0.6578867 0.7708010 0.6524913 0.1588198

## [1] "95% confidence intervals for differences between configurations in mean of average entrance:"

##   DIFFERENCE      MEAN_D     CI_low   CI_high
## 1        1-2  0.21536644 -0.2861270 0.7168599
## 2        1-3  0.17434181 -0.3531090 0.7017926
## 3        2-3 -0.04102463 -0.5060843 0.4240350

## [1] "95% confidence intervals for differences between configurations in utilization rate:"

##   DIFFERENCE      MEAN_D      CI_low    CI_high
## 1        1-2  0.05575202 -0.03919262 0.15069666
## 2        1-3 -0.03279445 -0.13614332 0.07055442
## 3        2-3 -0.08854647 -0.18162047 0.00452752

## [1] "Seems that 0 is included in all intervals, no significant differences"
```

Dependent (i.e. contrafactual) hospitals with simPy

```
# We are using the same settings as earlier
#initialize results matrix
sim_que = np.zeros( (N_SAMPLES, N_HOSP) )
sim_uti = np.zeros( (N_SAMPLES, N_HOSP) )
#samples are dependent, seeds are replicated between hospitals (N_SAMPLE seeds)
RANDOM_SEEDS = [*range(N_SAMPLES)]

for h, config in enumerate(settings.CONFIGURATIONS):
    for sample_i in range(N_SAMPLES):
        random.seed(RANDOM_SEEDS[sample_i])
        env = simpy.Environment()
        hospital = Hospital(env, config["n_preparation_rooms"], config["n_recovery_rooms"])
        env.run(until=SIM_TIME)
        sample_data = get_data(hospital)
        sim_que[sample_i, h] = sample_data["mean_queue_at_entrance"]
        sim_uti[sample_i, h] = sample_data["utilization_rate_of_operation_theatre"]
```

```
# SAVING TO FILE
#    with open("data.json", mode="w", encoding="utf-8") as f:
#        json.dump(data, f, indent=4)
```

Calculating results with R

Differences between dependent simulations

```r
sim_que = py$sim_que
sim_uti = py$sim_uti
PREP_ROOMS = sapply(py$settings$CONFIGURATIONS, FUN = function(x) x$n_preparation_rooms)
REC_ROOMS = sapply(py$settings$CONFIGURATIONS, FUN = function(x) x$n_recovery_rooms)

# Calculate differences of original simulated values
d_que = cbind(sim_que[,1] - sim_que[,2], sim_que[,1] - sim_que[,3], sim_que[,2] - sim_que[,3])
d_uti = cbind(sim_uti[,1] - sim_uti[,2], sim_uti[,1] - sim_uti[,3], sim_uti[,2] - sim_uti[,3])

mean_d_que <- apply(d_que, 2, mean)
mean_d_uti <- apply(d_uti, 2, mean)
sd_d_que <- apply(d_que, 2, sd)
sd_d_uti <- apply(d_uti, 2, sd)

# Table of descriptive statistics of the differences
descriptives <- data.frame(
    PREP_ROOMS = PREP_ROOMS,
    REC_ROOMS = REC_ROOMS,
    MEAN_D_QUE = mean_d_que,
    SD_D_QUE = sd_d_que,
    MEAN_D_UTI = mean_d_uti,
    SD_D_UTI = sd_d_uti
    )

# Similar confidence intervals now with paired samples (paired = TRUE)
que_95ci <-data.frame(
    DIFFERENCE = c("1-2", "1-3", "2-3"),
    MEAN_D = descriptives$MEAN_D_QUE,
    CI_low = c(t.test(sim_que[,1], sim_que[,2], paired = TRUE)$conf.int[1],
               t.test(sim_que[,1], sim_que[,3], paired = TRUE)$conf.int[1],
               t.test(sim_que[,2], sim_que[,3], paired = TRUE)$conf.int[1]
    ),
    CI_high = c(t.test(sim_que[,1], sim_que[,2], paired = TRUE)$conf.int[2],
               t.test(sim_que[,1], sim_que[,3], paired = TRUE)$conf.int[2],
               t.test(sim_que[,2], sim_que[,3], paired = TRUE)$conf.int[2]
    )
)

uti_95ci <-data.frame(
    DIFFERENCE = c("1-2", "1-3", "2-3"),
    MEAN_D = descriptives$MEAN_D_UTI,
    CI_low = c(t.test(sim_uti[,1], sim_uti[,2], paired = TRUE)$conf.int[1],
               t.test(sim_uti[,1], sim_uti[,3], paired = TRUE)$conf.int[1],
```

```r
              t.test(sim_uti[,2], sim_uti[,3], paired = TRUE)$conf.int[1]
    ),
    CI_high = c(t.test(sim_uti[,1], sim_uti[,2], paired = TRUE)$conf.int[2],
              t.test(sim_uti[,1], sim_uti[,3], paired = TRUE)$conf.int[2],
              t.test(sim_uti[,2], sim_uti[,3], paired = TRUE)$conf.int[2]
    )
)
```

```
## [1] "Descriptive statistics"

##   PREP_ROOMS REC_ROOMS MEAN_D_QUE   SD_D_QUE   MEAN_D_UTI   SD_D_UTI
## 1          3         4 0.01557908 0.03397063 -0.003135373 0.01402182
## 2          3         5 0.49976521 0.50060561 -0.025976386 0.04030450
## 3          4         5 0.48418613 0.49957933 -0.022841013 0.03832065

## [1] "95% confidence intervals for mean differences between configurations in average entrance:"

##   DIFFERENCE     MEAN_D        CI_low    CI_high
## 1        1-2 0.01557908 -0.0003196628 0.03147782
## 2        1-3 0.49976521  0.2654745702 0.73405584
## 3        2-3 0.48418613  0.2503758043 0.71799645

## [1] "95% confidence intervals for mean differences between configurations in utilization rate:"

##   DIFFERENCE       MEAN_D       CI_low      CI_high
## 1        1-2 -0.003135373 -0.009697785  0.003427038
## 2        1-3 -0.025976386 -0.044839472 -0.007113299
## 3        2-3 -0.022841013 -0.040775629 -0.004906396

## [1] "Seems that this method is more efficient. Differences between hospitals (1,3) and (2,3)"

## [1] "are differing from zero statistically significantly with both variables, average queue and util
```