Lab 3 report
StudentID 83459

Loading the data

```
damages = []
with open('damages.txt','r') as csvfile:
   reader = csv.reader (csvfile, delimiter=";")
   for row in reader:
      damages.append(int(row[1]))

damages_ln = sc.log(damages)
```

K-Test

```
test2 = kstest(damages_ln, sc.stats.norm.cdf,
           args = (sc.mean(damages_ln), sc.std(damages_ln)))
if test2[1] > 0.05:
   print ("p-value: ", round(test2[1], 4),
        "- fail to reject the null hypothesis")
else:
   print("null hypothesis should be rejected")
```

```
p-value:  0.9979 - fail to reject the null hypothesis
```

simulation parameters

```
damages_ln_avg = sc.mean(damages_ln)
damages_ln_std = sc.std(damages_ln)
number_of_policies = 1500
number_of_damages_avg
time_horizon = 2
balance = 10000
insurance_premium = 500
```

Simulation model

```
def model (number_of_policies, number_of_damages_avg,
        damages_ln_avg, damages_ln_std, time_horizon,
        balance, insurance_premium, seed):

   # Common Random Numbers:
   # sc.random.seed(seed)

   # contracts signing calendar:
   dates = [sc.random.randint(0, 364) for i in range(number_of_policies)]
   calendar_contracts = [0]*365
```

```
    for day in dates:
        calendar_contracts[day] += 1

    # number of damages per one policy:
    number_of_damages = sc.random.poisson(number_of_damages_avg,
                                  number_of_policies)

    # compensations calendar:
    calendar_compensations = [0]*(365*time_horizon)
    for k in range(number_of_policies):
        for s in range(number_of_damages[k]):
            compensation_date = dates[k] + sc.random.randint(0, 364)
            calendar_compensations[compensation_date] += 1


    # time_horizon simulation:
    for day in range(365*time_horizon):
        if day <= 364:
            balance += calendar_contracts[day] * insurance_premium

        number_of_compensations = calendar_compensations[day]
        compensations = 0
        if number_of_compensations > 0:
            compensations = sum(sc.exp(sc.random.normal(damages_ln_avg,
                                        damages_ln_std,
                                        number_of_compensations)))
        if balance < compensations:
            return balance - compensations
        else:
            pass
        balance -= compensations
    return balance
```

run function

```
def run(balance, insurance_premium, repeats,
        number_of_policies, number_of_damages_avg ,
        damages_ln_avg, damages_ln_std, time_horizon):

    results = []
    bankruptcy = 0
    results_positve = []

    for seed in range(repeats):
        results.append(model(number_of_policies, number_of_damages_avg ,
                    damages_ln_avg, damages_ln_std, time_horizon,
                    balance, insurance_premium, seed))
```

```
        if results[seed] < 0:
            bankruptcy += 1
        if results[seed] > 0:
            results_positve.append(results[seed])

    results_average = sc.mean(results_positve)
    bankruptcy_risk = bankruptcy / repeats

    return [bankruptcy, bankruptcy_risk, results_average]
```

Simulation results

```
# model_parameters
results = [] # średni wynik finansowy firmy
insurance_cost = []
bankruptcy_risk = []
bankruptcy_cnt = []

repeats = 100
number_of_policies = 1500
time_horizon = 2 # 2 years


for balance in range(10000, 50000,10000):
    for insurance_premium in range(500,1000,100):
        run_result = run(balance, insurance_premium, repeats,
                    number_of_policies, number_of_damages_avg,
                    damages_ln_avg, damages_ln_std, time_horizon)

        insurance_cost.append(insurance_premium)
        bankruptcy_cnt.append(run_result[0])
        bankruptcy_risk.append(run_result[1])
        results.append(run_result[2])

        print("BALANCE: ", balance, "INSURANCE PREMIUM: ", insurance_premium,
            "BANKRUPTCY CNT: ", run_result[0], "AVERAGE RESULT: ", round(run_result[2]),
            "BANKRUPTCY RISK: ", run_result[1])

plt.plot(insurance_cost, bankruptcy_risk)
plt.ylabel('BANKRUPTCY RISK')
plt.show()
```

```
BALANCE:  10000 INSURANCE PREMIUM:  500 BANKRUPTCY CNT:  99 AVERAGE RESULT:  32196.0 BANKRUPTCY RISK:  0.99
BALANCE:  10000 INSURANCE PREMIUM:  600 BANKRUPTCY CNT:  96 AVERAGE RESULT:  16962.0 BANKRUPTCY RISK:  0.96
BALANCE:  10000 INSURANCE PREMIUM:  700 BANKRUPTCY CNT:  57 AVERAGE RESULT:  78164.0 BANKRUPTCY RISK:  0.57
BALANCE:  10000 INSURANCE PREMIUM:  800 BANKRUPTCY CNT:  16 AVERAGE RESULT:  162960.0 BANKRUPTCY RISK:  0.1
6
BALANCE:  10000 INSURANCE PREMIUM:  900 BANKRUPTCY CNT:  1 AVERAGE RESULT:  282981.0 BANKRUPTCY RISK:  0.01
BALANCE:  20000 INSURANCE PREMIUM:  500 BANKRUPTCY CNT:  100 AVERAGE RESULT:  nan BANKRUPTCY RISK:  1.0
BALANCE:  20000 INSURANCE PREMIUM:  600 BANKRUPTCY CNT:  95 AVERAGE RESULT:  57776.0 BANKRUPTCY RISK:  0.95
BALANCE:  20000 INSURANCE PREMIUM:  700 BANKRUPTCY CNT:  44 AVERAGE RESULT:  73724.0 BANKRUPTCY RISK:  0.44
BALANCE:  20000 INSURANCE PREMIUM:  800 BANKRUPTCY CNT:  13 AVERAGE RESULT:  166453.0 BANKRUPTCY RISK:  0.1
3
BALANCE:  20000 INSURANCE PREMIUM:  900 BANKRUPTCY CNT:  2 AVERAGE RESULT:  296120.0 BANKRUPTCY RISK:  0.02
BALANCE:  30000 INSURANCE PREMIUM:  500 BANKRUPTCY CNT:  99 AVERAGE RESULT:  17201.0 BANKRUPTCY RISK:  0.99
BALANCE:  30000 INSURANCE PREMIUM:  600 BANKRUPTCY CNT:  93 AVERAGE RESULT:  35208.0 BANKRUPTCY RISK:  0.93
BALANCE:  30000 INSURANCE PREMIUM:  700 BANKRUPTCY CNT:  51 AVERAGE RESULT:  78487.0 BANKRUPTCY RISK:  0.51
BALANCE:  30000 INSURANCE PREMIUM:  800 BANKRUPTCY CNT:  7 AVERAGE RESULT:  184889.0 BANKRUPTCY RISK:  0.07
BALANCE:  30000 INSURANCE PREMIUM:  900 BANKRUPTCY CNT:  1 AVERAGE RESULT:  305703.0 BANKRUPTCY RISK:  0.01
BALANCE:  40000 INSURANCE PREMIUM:  500 BANKRUPTCY CNT:  98 AVERAGE RESULT:  40194.0 BANKRUPTCY RISK:  0.98
BALANCE:  40000 INSURANCE PREMIUM:  600 BANKRUPTCY CNT:  90 AVERAGE RESULT:  59247.0 BANKRUPTCY RISK:  0.9
BALANCE:  40000 INSURANCE PREMIUM:  700 BANKRUPTCY CNT:  43 AVERAGE RESULT:  105880.0 BANKRUPTCY RISK:  0.4
3
BALANCE:  40000 INSURANCE PREMIUM:  800 BANKRUPTCY CNT:  7 AVERAGE RESULT:  181489.0 BANKRUPTCY RISK:  0.07
BALANCE:  40000 INSURANCE PREMIUM:  900 BANKRUPTCY CNT:  1 AVERAGE RESULT:  305768.0 BANKRUPTCY RISK:  0.01
```

Conclusion:

After running simulation with 100 repeats and 1500 policies we can clearly see that the there are few scenarios of the lowest bankruptcy risk:

* bankruptcy risk = 0.01: balance - 10000, insurance premium - 800/900
* bankruptcy risk = 0.01/0.02: balance - 20000, insurance premium - 800/900
* bankruptcy risk = 0.01: balance - 30000, insurance premium - 900
* bankruptcy risk = 0.01: balance - 40000, insurance premium - 900

Interestingly, that all balances with insurance premium 500/600 give bankruptcy risk > 0.9