Name: ______________________________

Date: ______________________________

Collaborators: ____

(If applicable, the collaborators submit their individually written assignments together)

Question:	1	2	3	4	Total
Points:	10	35	30	10	85
Score:					

Instructor/grader comments:

1. (10 points) Before starting coding for this assignment:
   - ssh to your virtual machine
   - Create a directory, e.g. **hw08**.
   - Change to that directory.
   - Start Julia, switch to the package mode, and activate your project. Install the packages that you use for the assignment: `IJulia`, `PyPlot`, `DataFrames`, `CSV`. Exit Julia.
   - Create an empty `README.md` file.
   - Copy `.gitignore` file to your working directory.
   - Use jupyter notebook interface to write the code for this homework assignment.

2. **Hubble's law, Hubble constant, and the age of the Universe**

   Hubble's law is the observation that galaxies are moving away from Earth at speeds proportional to their distance: $v = H_0 D$, where H_0 is the *Hubble constant*, D is the distance to a galaxy, and v is the speed of separation.

   Hubble constant is most frequently quoted in (km/s)/Mpc, thus giving the speed in km/s of a galaxy 1 megaparsec (3.09×10^{19} km) away. However, the SI unit of H_0 is simply s^{-1}. The reciprocal of H_0 is known as the *Hubble time*. The Hubble time is the age the Universe would have had if the expansion had been linear; it is different from the real age of the Universe because the real expansion is not linear. However, the Hubble time and the age of the Universe are related by a dimensionless factor which depends on the mass-energy content of the Universe; it assumed to be close to 0.96.

   The goal of this assignment is to determine the Hubble constant from the experimental data on the magnitude and redshift of supernovae by following the steps described in the article *A simple determination of Hubble's constant* by E. Benedetto at. al., Eur. J. Phys., **37** 025601 (2016).

   (a) (10 points) Download the catalog of red shift and distance observational data from The VizieR database of astronomical catalogues and clean the data. Use the following fragment of the code:

   ```
   url = "http://vizier.u-strasbg.fr/viz-bin/asu-txt?-source=
      J/ApJ/716/712/tableb2&-out=SN&-out=zCMB&-out=mu"
   catalog = download(url)
   ```

```
df = CSV.read(catalog, DataFrame, skipto=38,
              delim=' ', ignorerepeated=true,
              types=[String, Float64, Float64],
              silencewarnings=true,
              header=["name", "redshift", "modulus"])
dropmissing!(df)
```

The dataframe that you create contains three columns: the names of the stars, the observed dimensionless red shifts $z = (\lambda_o - \lambda)/\lambda$, and dimensionless log-like measure of distances called *distance modulus*.

As described in the article, select the stars with a relative red shift between 0.0166 and 0.0333. Sort the observational data by redshifts.

```
filter!(row -> row.redshift > 0.0166  &&
        row.redshift < 0.0333, df)
sort!(df, "redshift")
```

(b) (10 points) Use the function `linear_regression` that you wrote for Midterm 2 to find the best linear fit of the data. On the same figure plot the observational data (distance modulus vs redshift,) as disconnected symbols and the best fit line as a solid line (without data points). Provide the axis labels, grid, title, legend.

(c) (5 points) Convert the value of the slope of your linear fit of distance modulus vs redshift, β, to the value of Hubble constant in (km/s)/Mpc and print it. Use the relation that is derived in the article:

$$H_0 = 3000 * (5/\log(10))/\beta.$$

(d) (10 points) Convert the value of the Hubble constant to SI units and print it. Estimate and print the age of the Universe in billion of years. (Assume that 1 year $= 60 \times 60 \times 24 \times 365$ seconds.) Describe your results for the Hubble constant and the age of the Universe in your README.md file.

Place your code for Problem 2 into a notebook **hw08p2.ipynb**

3. **Monte Carlo methods for calculating of mathematical constants**

Consider the following Monte Carlo algorithm for calculating the value of e, the base of natural logarithm:

> Let X_i be independent random numbers from the uniform distribution on $[0,1)$. Let n be the minimum number such that $\sum_{i=1}^{n} X_i > 1$. (note that n is a random variable, $n \geq 2$.) Then, the average value of n is equal to e.

The proof of the statement - elegant and not required for the coding - is available on the class website at https://www.phys.uconn.edu/~rozman/Courses/P2200_23F/.

(a) (15 points) Write a Julia function, `mce(m)`, that accepts the number of trials m, and implement the Monte Carlo method described above.

(b) (15 points) Use the following code fragment to determine the convergence rate of the algorithm and the errors.

```
n = 14
es = zeros(n)
estds = zeros(n)
np = zeros(Int64, n)
k = 256
ns = ones(Int64, k)
res = zeros(k)
for i = 1:n
    @show i
    np[i] = 2^(i+12)
    res .= map(mce, np[i]*ns)
    es[i] = mean(res)
    estds[i] = std(res, mean=es[i])/sqrt(k)
end
```

Seed your random number generator with a seed of your choice **before running your calculations**.

Use the function `linear_regression` that you wrote for Midterm 2 to find the best linear fit of the data.

```
alpha, beta, sigma = linear_regression(log.(np), log.(
   estds))
```

Plot the results of your numerical experiment as follows.

```
loglog(np, abs.(es .- exp(1.0)), label="Actual err",
       marker="o", markersize=2, linestyle="none")
loglog(np, estds, label="Std err", marker="o",
       markersize=3, linestyle="none")
loglog(np, exp.(alpha .+ beta .* log.(np)),
       label="LSq fit std err", linestyle="dashed")
grid(true)
xlabel("MC steps")
ylabel("Error")
legend()
```

Round your convergence rate to the correct numbers of digits. Describe your result for the convergence rate of the algorithm in your README.md file.

Place your code for Problem 3 into a notebook **hw08p3.ipynb**

4. (10 points) **Submission of the assignment**

   1. Create an empty GitLab project called **hw08** (name it exactly as shown).
   2. Clean the cells of your jupyter notebook and save the notebook. Delete unneeded notebooks if you created ones (e.g. `Untitled.ipynb`). Initialize a git repository for your project. Check in your notebook(s), Project.toml and Manifest.toml, an empty README.md file, and your .gitignore file into the repository. Provide a meaningful commit message. Push the content of your git repository to GitLab hw08 project.
   3. Edit README.md file to add content as requested in the assignment.
   4. Pull the updated README.md file to your local git repository (`git pull`).
   5. Share your GitLab project with the instructor (GitLab user name p2200_23f_in) and grant him **Reporter** privileges.