Lea Setruk 345226179
Aviva Shneor Simchon 317766731

Optimization

Optimal parameters of data distribution

Nelder-Mead Algorithm

Data

Our dataset is « Baltimore City Employee Salaries FY2019 ».

This dataset captures gross salary of Baltimore City from July 1, 2018 through June 30, 2019 and includes only those employees who were employed on June 30, 2019.

This is how the dataset looks like :

	NAME	JOBTITLE	DEPTID	DESCR	HIRE_DT	ANNUAL_RT	Gross
0	Aaron,Kareem D	Utilities Inst Repair I	A50550	DPW-Water & Waste Water (550)	08/27/2018 12:00:00 AM	32470.0	25743.94
1	Aaron,Patricia G	Facilities/Office Services II	A03031	OED-Employment Dev (031)	10/24/1979 12:00:00 AM	60200.0	57806.13
2	Abadir,Adam O	Council Technician	A02002	City Council (002)	12/12/2016 12:00:00 AM	64823.0	64774.11
3	Abaku,Aigbolosimuan O	Police Officer	A99094	Police Department (094)	04/17/2018 12:00:00 AM	53640.0	59361.55
4	Abbeduto,Mack	Assistant State's Attorney	A29011	States Attorneys Office (011)	05/22/2017 12:00:00 AM	68562.0	61693.59

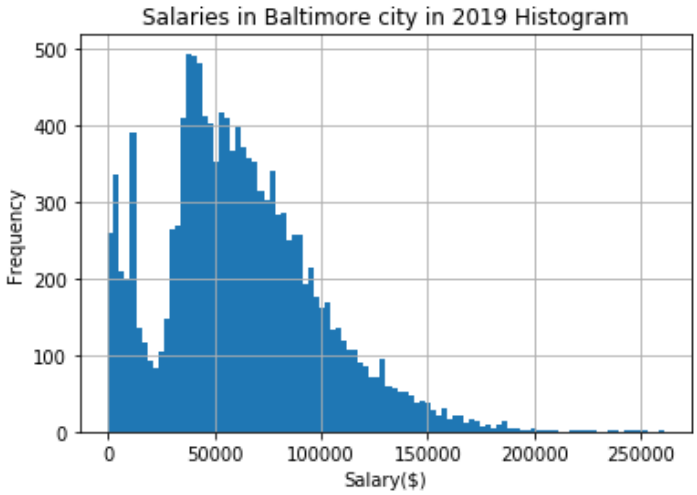
The first column represents the name of the employees. The second one represens their job. The third, the department ID, the fourth the description of the department. The fifth, the date when they have been hired. The sixth, the anual salary and the last one represents the gross salary.

We are interested in the last column : the gross salary.

Goal

First of all, we need to graphically represent our dataset (the gross salary column). From it, we would like to find the right probability distribution and its optimal parameters.

Here is the histogram of the data :

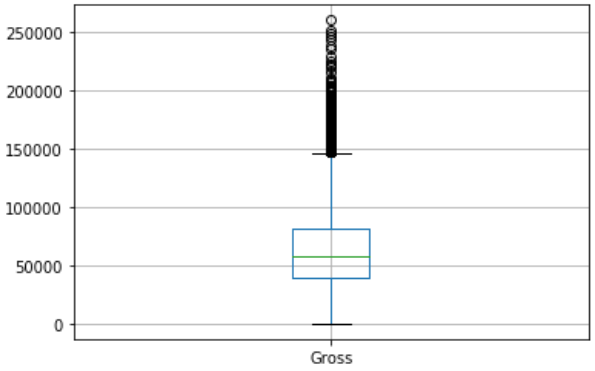


Lea Setruk 345226179
Aviva Shneor Simchon 317766731

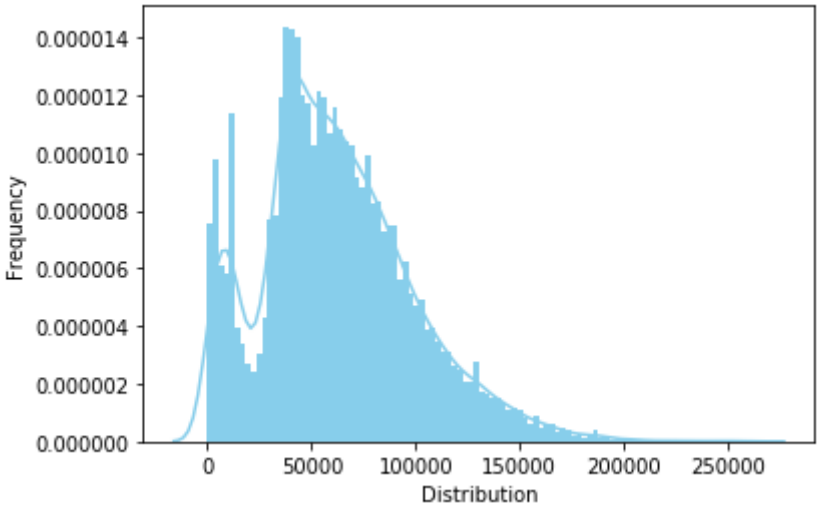
We also made a statistics analysis of our data. Let's describe our data (mean, standard variation…) :

	ANNUAL_RT	Gross
count	13811.00000	13811.000000
mean	57845.28268	62099.928316
std	26934.44789	35939.422182
min	1750.00000	0.000000
25%	36312.00000	38854.575000
50%	53640.00000	58049.395000
75%	75806.00000	82098.535000
max	275000.00000	260775.260000

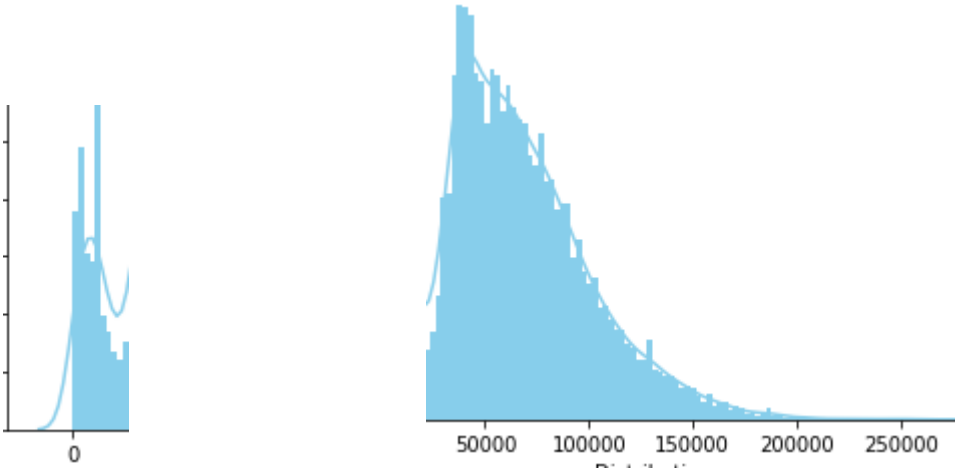
Box plot :



Application



Lea Setruk 345226179
Aviva Shneor Simchon 317766731



We can see that we have at least 2 different distributions (from 0 to 25000 and from 25000 to the end), maybe even 3 if we see closely. We will try both and check which number of distributions is better later, using AIC.

We tried different distributions for the linear combination (such as beta, gamma, lognormal…) that seemed, according to the eye, good candidates. We also tried mix of different distributions.

After observations and test, we concluded that those different distributions are lognormal.

Here's the density function of lognormal distribution :

$$f_Y(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) = \frac{1}{x} f_X(\ln(x);\mu,\sigma)$$

μ represents the mean and σ the standard deviation.

- We created a function that gets the parameters and returns this density function.

- Another function is needed : the linear combination. Indeed, the data isn't distributed with only one distribution. We want to create a linear combination : y = af(x) + bg(x) with a+b =1 (our constraint), y = af(x) + (1-a)g(x). f represents the density function of the first distribution and g of the second one. Y represents the theoritical distribution. We will also check the model with a linear combination of three distributions with y = af(x) + bg(x) + ch(x) =af(x) + bg(x) + (1-a-b)h(x) with a+b+c=1 (our constraint).

- We evaluate the parameters, in the beggining, randomly and we try to fit them to our data. For our first case (two distributions), our parameters are a, μ_1, σ_1, μ_2, σ_2. For the second case (three distributions), our parameters are a, b, μ_1, σ_1, μ_2, σ_2, μ_3, σ_3.

- We then, created a function that get all of the data, and counts how many salaries there are in every parts (in every bins). It gives us the points that represent the empirical distribution.

- We implemented another function that calculates the Kolmogorov distance, which is the

Lea Setruk 345226179
Aviva Shneor Simchon 317766731

maximal distance between the empiric distribution and the theoritical one (Python has this function but we implemented it, checked and found the same results). This cost function is not derivable, so we chose to work with an optimization algorithm that doesn't need derivatives : Nelder Mead (downhill simplex method).

Nelder-Mead algorithm

Our goal is to minimize the cost function (Kolmogorov distance). We want to find the parameters that helps us to do so.

The Nelder-Mead algorithm is a heuristic numeric method which minimizes an objective function in a multidimension space.

It uses the symplex concept which is a special polytope of N + 1 vertices in N dimensions.

During the iterations, the simplex moves, get transformed and reduced until its vertices approach a point where the function is minimal.

By successive iterations, the process consists in determining the point of the simplex where the function is maximum in order to replace it by the reflection of this point with respect to the center of gravity of N remaining points.

Implementation :

We implemented this algorithm following those steps :

1) Choice of N+1 points in N dimension. The simplex is $\{x_1, x_2, \ldots, x_{N+1}\}$

2) Calculation of the values of the cost function at these points, sorting of the points : $f(x_1) \leq f(x_2) \leq \ldots \leq f(x_{N+1})$

3) Calculation of the initial point, center of gravity of all points except the last one.

4) Calculation of the reflection : $x_r = x_0 + \alpha(x_0 - x_{N+1})$

5) $f(x_1) \leq f(x_r) \leq f(x_N)$ replacement of x_{N+1} by x_r

6) $f(x_r) < f(x_1)$, calculation of $x_l = x_0 + \gamma(x_r - x_0)$. This is the expension of the simplex If $f(x_l) < f(x_r)$, remplacement de x_{N+1} by x_l otherwise, remplacement of x_{N+1} by x_r et back to step 2.

7) $f(x_r) > f(x_N)$, calculation of $x_c = x_0 + \rho(x_{N+1} - x_0)$. This is the contraction of the simplex.

If $f(x_r) \geq f(x_N)$, replacement of x_{N+1} by x_c and back to step 2, otherwise, back to step 8.

8) Ratio homothety σ and the center x_1 : replacement of x_i by $x_1 + \sigma(x_i - x_1)$, back to step 2.

Lea Setruk 345226179
Aviva Shneor Simchon 317766731

With α, γ, ρ, σ that are coefficients such as $\alpha > 0$, $\gamma > 1$, $0<\rho<0.5$.

We used the standard values : $\alpha=1$, $\gamma=2$, $\rho=0.5$, $\sigma=0.5$.

Also, in Python, this algorithm exists. We used our implementation and then checked with the Python function. We found good results with our implementation and used the parameters we found with it.

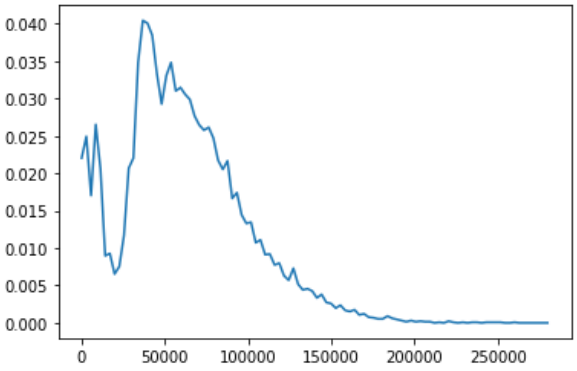
Results :

After using our algorithm, we got the optimal paramaters that minimizes our cost function.

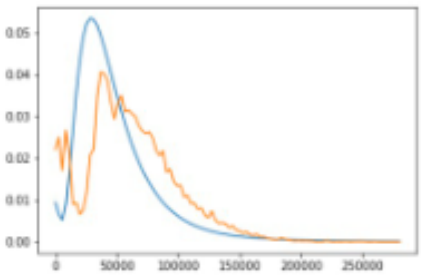
For the first model, we got those parameters : a = 0.3084, μ_1= 10.9212, σ_1 = 2.627825, μ_2 = 10.584, σ_2 = 0.514

For the second model : a = 0.06320954, b = 0.11235451, μ_1 = 14.87070052, σ_1 = 3.93917328, μ_2= 8.90019041, σ_2 = 0.87787672, μ_3 = 11.03283469, σ_3 = 0.48644798

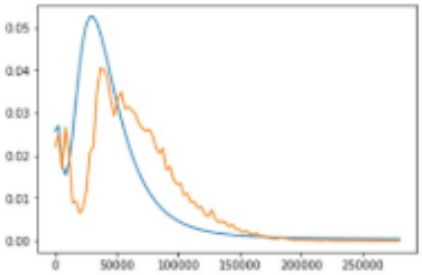
We get those theoretical functions (densities), which are very close to our emperical function :



Emperical function

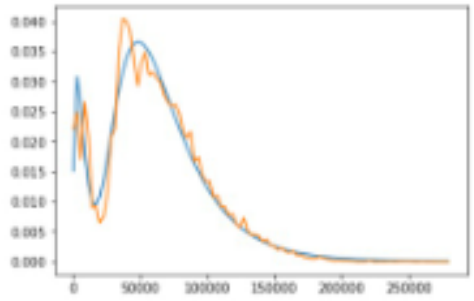


Theoretical function (in blue) for a linear combination of two lognormal distributions with our implementation

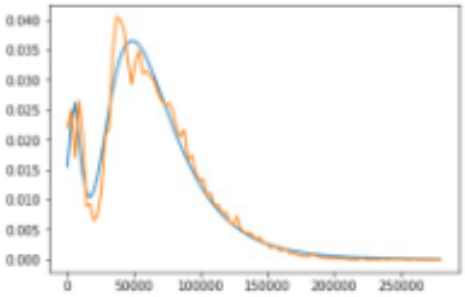


Theoretical function (in blue) for a linear combination of two lognormal distributions with Python function

Lea Setruk 345226179
Aviva Shneor Simchon 317766731



Theoretical function (in blue) for a linear combination of three lognormal distributions with our implementation



Theoretical function (in blue) for a linear combination of three lognormal distributions with Python function

AIC Criterion

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

This is the AIC formula with *k* the number of estimated parameters in the model. And ^L the maximum value of the likelihood function for the model.

We then, calculated ^L.

The likelihood function is defined by :

$$\mathcal{L}(\theta \mid x) = f_\theta(x) = p(X = x \mid \theta),$$

We found its maximum, calculated the AIC for our two models (for two distributions and for three distributions in the linear combination).

AIC_for_2 = 61.69

AIC_for_3 = 48.2

We got AIC smaller for the linear combination with three distributions. Therefore, this is the best model for us (as we could have seen in the graphs bellow).