

Report on learning practice # 1
Analysis of univariate random variables

Performed by:
Grandilevskii Aleksei
J4133c
Dubinin Ivan
Sorokin Mikhail
J4132c

Saint-Petersburg
2021

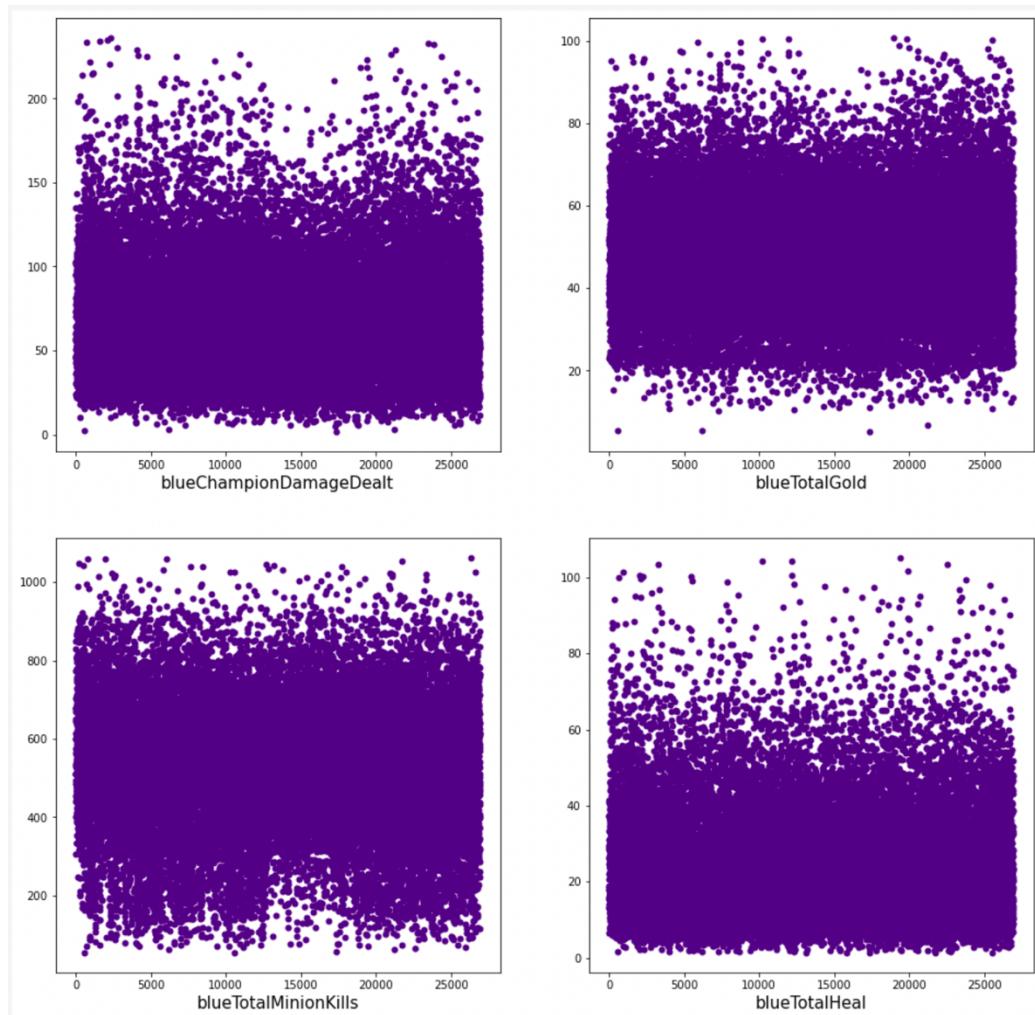
Table of contents:

1. Substantiation of chosen subsample:

	gameDuration	blueWins	blueWardPlaced	blueWardkills	blueKills	blueDeath	blueChampionDamageDealt	blueTotalGold	blueTotalMinionKills
0	22.050000	0	38	13	15	31	56.039	37.001	440
1	21.950000	1	57	18	19	8	60.243	41.072	531
2	15.533333	0	28	7	5	20	24.014	22.929	306
3	34.966667	0	129	39	26	36	101.607	63.447	774
4	39.066667	1	114	35	27	40	134.826	74.955	831
5	26.116667	1	65	23	26	18	59.839	52.221	576
6	28.100000	0	72	26	16	31	70.270	47.107	601

Pic.1. Short DS pandas presentation.

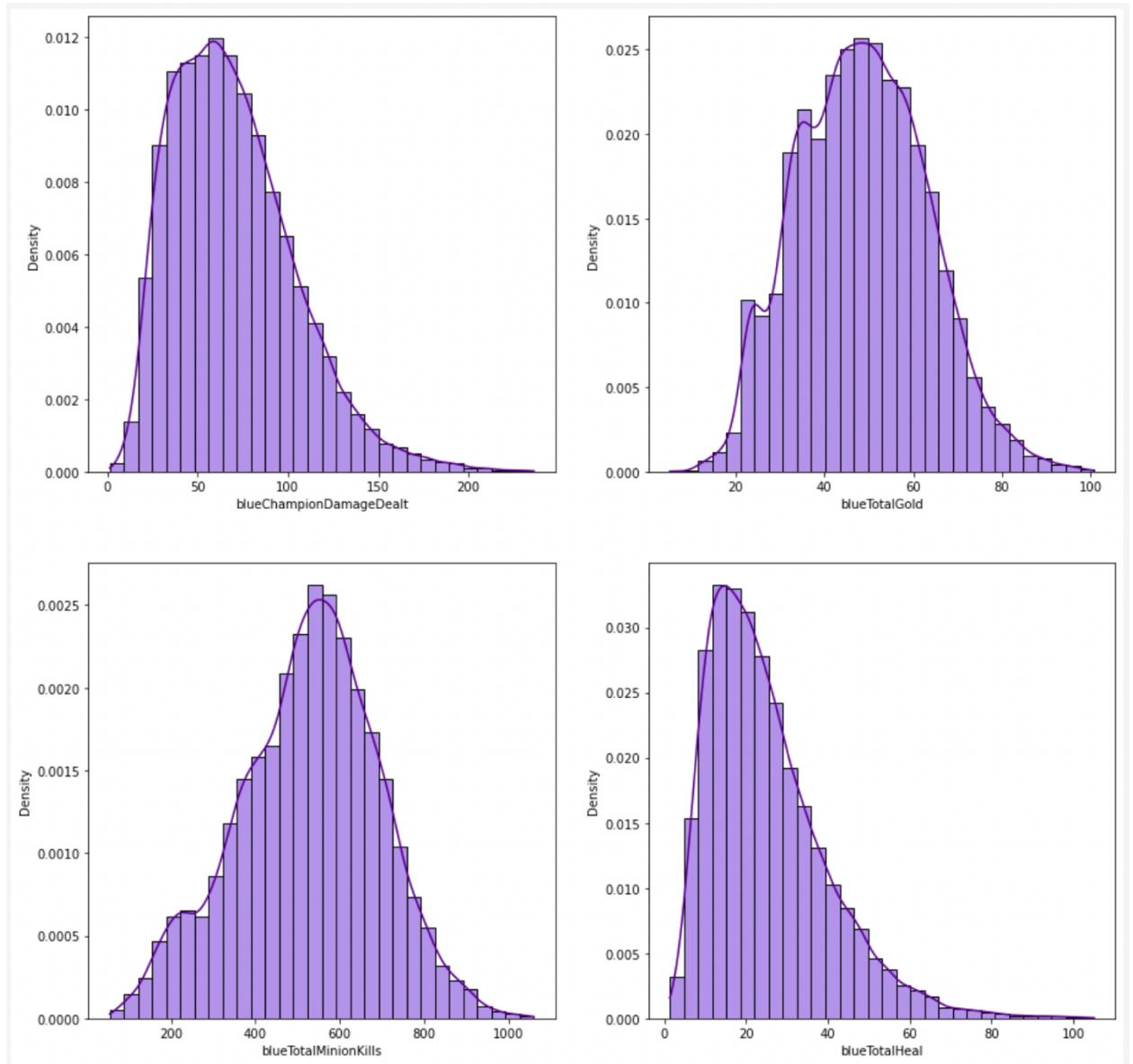
Our dataset is game statistics data from the League of Legends game for 2020 from rated games in the "challenger" rank. The dataset is built using Riot.API (open public API for various in-game parameters from online games from Riot Games). The dataset contains many statistical parameters of past matches, including damage done, in-game currency earned, data on victories and defeats, etc. More details can be found in the README.MD file in the datasets folder.



Pic.2. Display Lab 1 used data.

This Lab four of variables have been used. In our laboratory work, we analyze only the statistics of the "blue team".

2. Plotting a non-parametric estimation of PDF in form of a histogram and using kernel density function (all chosen variables are continuous in our case)

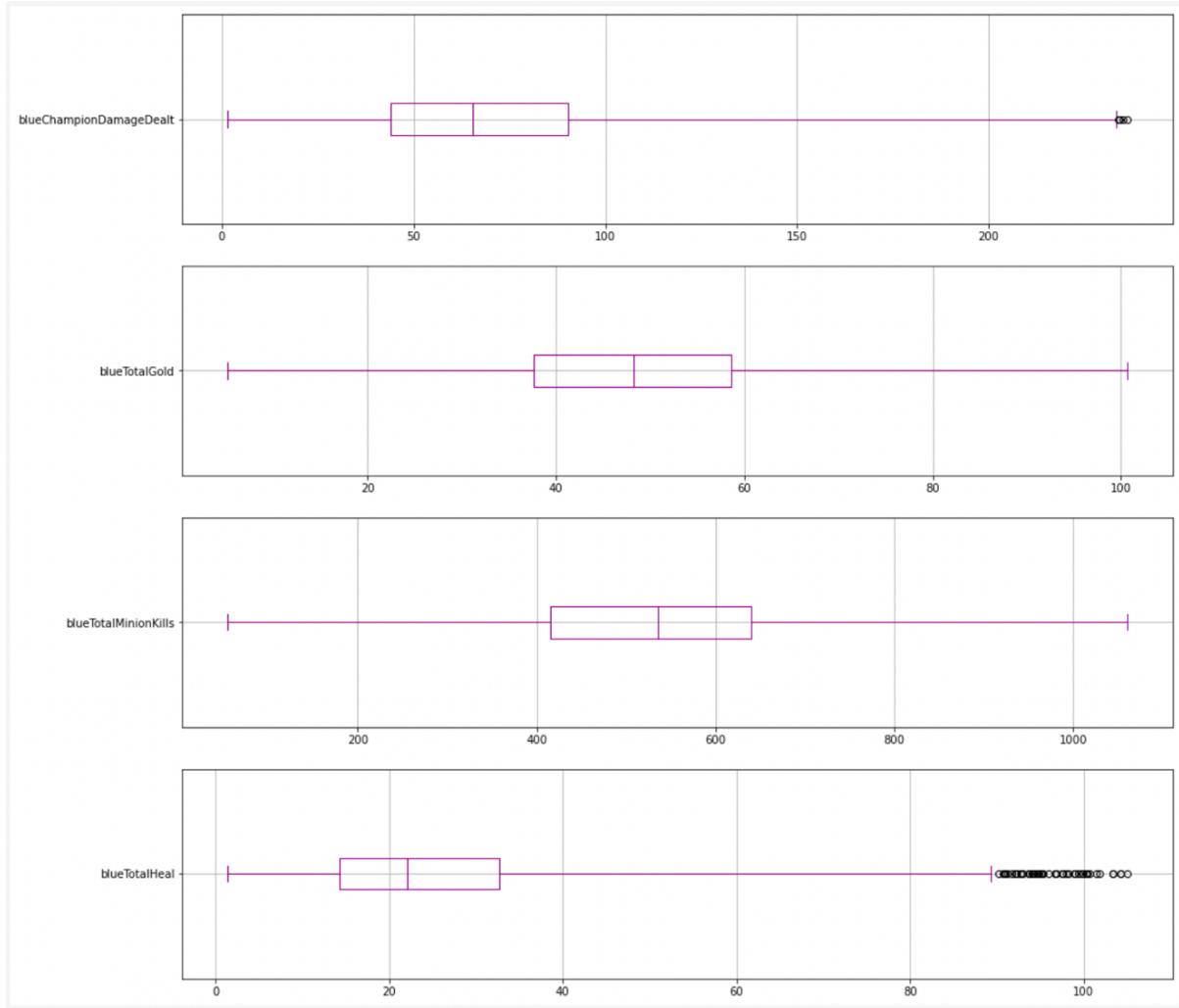


Pic.3. Histogram and KDE.

3. Order statistics estimation and its representation as “box with whiskers” plot.

column name	m.expectation	median	variance	s.deviation
blueChampionDamageDealt	70.07366307756241	65.353	1170.6887454414887	34.21532910029492
blueTotalGold	48.509690392534665	48.272	212.2499566304128	14.56880079589301
blueTotalMinionKills	525.625750878386	536.0	28098.582767424865	167.62631883873388
blueTotalHeal	25.137725905776566	22.14	209.51045629713036	14.47447602841396

Pic.4. Statistics estimation.



Pic.5. Variables' boxplot visualization.

As one can see boxplot for *blueTotalHeal* shows noticeable number of outliers. Nevertheless, this number is actually about 1/1000 of all values and does not have bad influence on distribution parameters estimation.

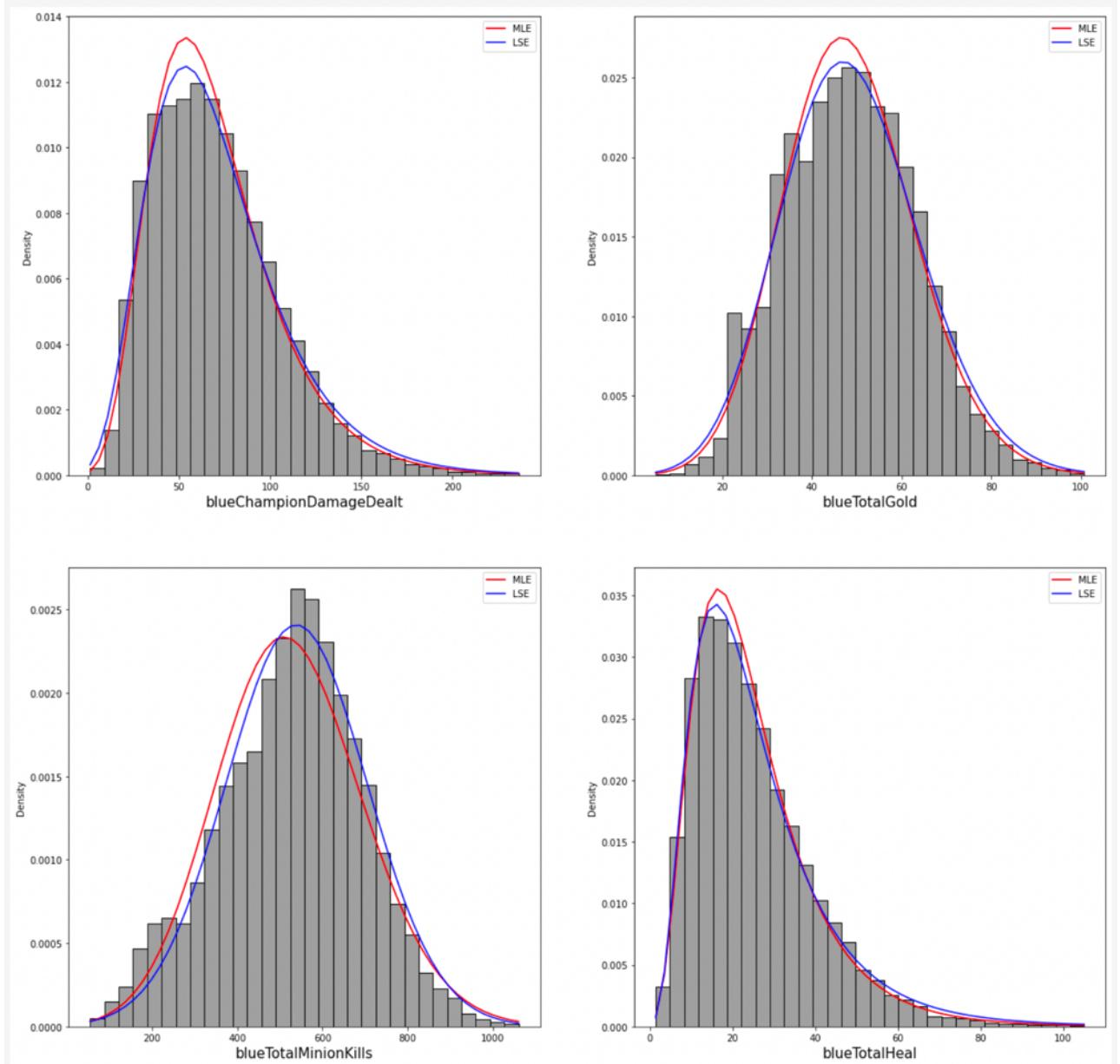
4. Selection of theoretical distributions that best reflect empirical data and 5. Estimation of random variable distribution parameters using maximum likelihood technique and least squares methods:

column name	MLE	LSE
blueChampionDamageDealt	(0.3567127289928922, -24.95902586794658, 89.2304579717554)	[0.35788322 -30.34002972 95.20364279]
blueTotalGold	(119.68833938433767, -64.3620826424339, 0.9430486790073629)	[119.26712721 -70.36638666]
blueTotalMinionKills	(239.90587126038213, -1351.387304764572, 7.823179868540159)	[13719.33891574 -13178.90377034]
blueTotalHeal	(0.4841462143044093, -4.116846783371216, 26.064832754593976)	[0.54214586 -2.7222856 24.85686619]

Pic.6. Parametric representation of MLE and LSE.

This table shows the distribution parameters calculated using the MLE and LSE algorithms. For MLE of parameters `scipy.stats` distribution fitting functions were used. For least squares method `minimize` function from `scipy.optimize` was used. X and Y data for optimization problem were got from histograms. As one can see on the picture below for *blueChampionDamageDealt* and *blueTotalHeal* variables **lognormal** distribution was chosen. For *blueTotalGold* and *blueTotalMinionKills* **chi-squared** one showed the best results.

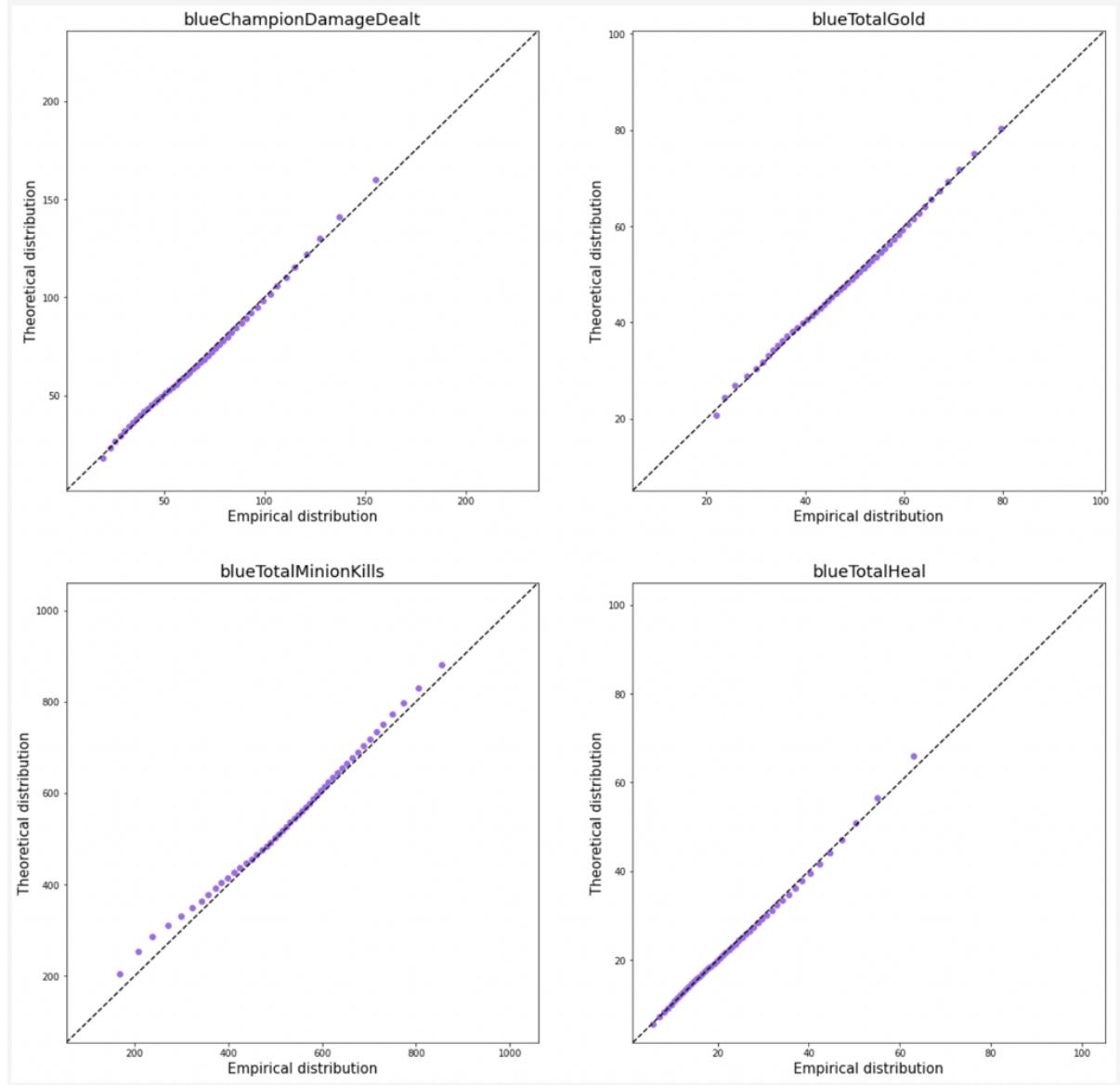
On the picture blue line represents results obtained by Least Squares method. PDF with parameters, estimated by MLE is represented by red line.



Pic. 7. Hist MLE and LSE visualization.

6. Validation of empirical and theoretical distributions using quantile biplots.

On this plot one can see, that tails of all QQ plots are quite good. At the same time for *blueTotalMinionKills* noticeable deviations for the lower percentiles can be found. This is due to rather sharp form of the appropriate histogram.



Pic.8. QQ biplots.

7. Statistical tests (2 at least).

For statistical test we used Kstest and CramerVonMises functions. For these both MLE and LSE estimated parameters are used. The gained results are shown in the table below.

Actually the obtained results do not favor the acceptance of the hypothesis that theoretical distribution correctly fit our data. Only for *blueTotalHeal* the hypothesis can be accepted with good significance level.

column name	type	Kstest	CramerVonMises
blueChampionDamageDealt	statistic	0.019030160618218384	1.7042937496546595
	pvalue	9.309208484143118e-09	5.930765115680359e-05
blueTotalGold	statistic	0.020360264121477956	2.0405842004896475
	pvalue	5.804583091103113e-10	1.0355533281991569e-05
blueTotalMinionKills	statistic	0.03156755099874624	10.700231253847136
	pvalue	2.3798715411823202e-23	1.3100864837412018e-10
blueTotalHeal	statistic	0.013289612452947575	1.8739174179727605
	pvalue	0.0001724190332847931	2.454761702808117e-05

Pic.9. Statistical tests.

Source code:

- The full repository with all the labs: <https://github.com/vandosik/M-M-MSA>
- The repo with Datasets and additional used Data info: <https://github.com/vandosik/M-M-MSA/tree/master/Datasets>
- The Lab 1 ipynb file: https://github.com/vandosik/M-M-MSA/blob/master/Lab_1/lab_1.ipynb

We recommend to use the first link because our GitHub project has README file with similar links and instructions which is really easy to use.

16 lines (16 sloc) | 1011 Bytes

[Raw](#) [Blame](#) [Copy](#) [Edit](#) [Delete](#)

Instruction

This is M&M MSA group 19 repo.

Choose Lab 1 - 4 folder to get access to relevant materials

1. [Lab_1](#). Analysis of univariate random variables
2. [Lab_2](#).
3. [Lab_3](#).
4. [Lab_4](#).

Inside each folder you can find the list of files include lab_XXX.ipynb file, lab_XXX_task.txt and lab_XXX_task.pptx, report.docx and README file which is a copy of the Markdown github report

Dataset files

All used datasets are published in [Datasets folder](#)

Participants of the project

- Ivan Dubinin: [vandosik](#)
- Alexey Grandilevskii: [zer0deck](#)
- Mikhail Sorokin: [MikhailSorokin](#)