Report on learning practice # 2

Analysis of multivariate random variables

Performed by:

Grandilevskii Aleksei

J4133c

Dubinin Ivan

Sorokin Mikhail

J4132c

Saint-Petersburg

2021

Table of contents:

Dataset preparation:

```
# Colums renaming and data preparation
source_df = source_df[['gameDuraton', # this is the value we will predict (target)
                       'blueWins', # this is our category sorter
                       'blueWardPlaced', # all other values are the predictors
                       'blueWardkills',
                       'blueKills',
                       'blueDeath',
                       'blueChampionDamageDealt',
                       'blueTotalGold',
                       'blueTotalMinionKills',
                       'blueJungleMinionKills',
                       'blueTotalHeal',
                       'blueObjectDamageDealt']]

# show new dataset
source_df.head(7)
```

✓ 0.2s Python

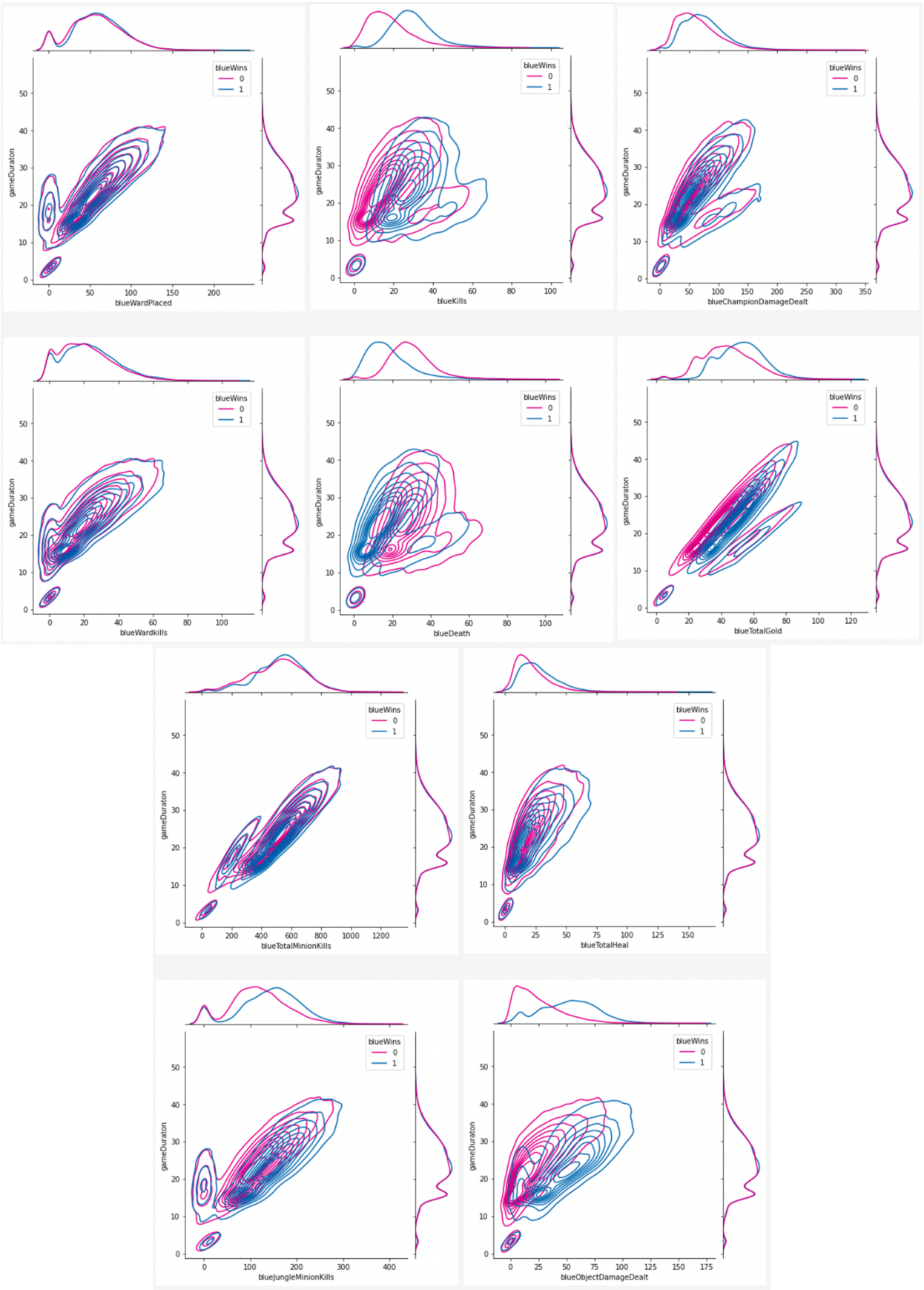
	gameDuraton	blueWins	blueWardPlaced	blueWardkills	blueKills	blueDeath	blueChampionDamageDealt	blueTotalGold	blueTotalMinionKills	blueJungleMi
0	22.050000	0	38	13	15	31	56.039	37.001	440	
1	21.950000	1	57	18	19	8	60.243	41.072	531	
2	15.533333	0	28	7	5	20	24.014	22.929	306	
3	34.966667	0	129	39	26	36	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. Dataset preparation.

In this lab, 12 dimensions will be considered, 1 target for predictive analysis, one dimension for categorization, and 10 predictor values.

From Lab 1: *"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."*

1. Plotting a non-parametric estimation of PDF in form of a histogram and kernel density function for MRV (or probability law in case of discrete MRV).



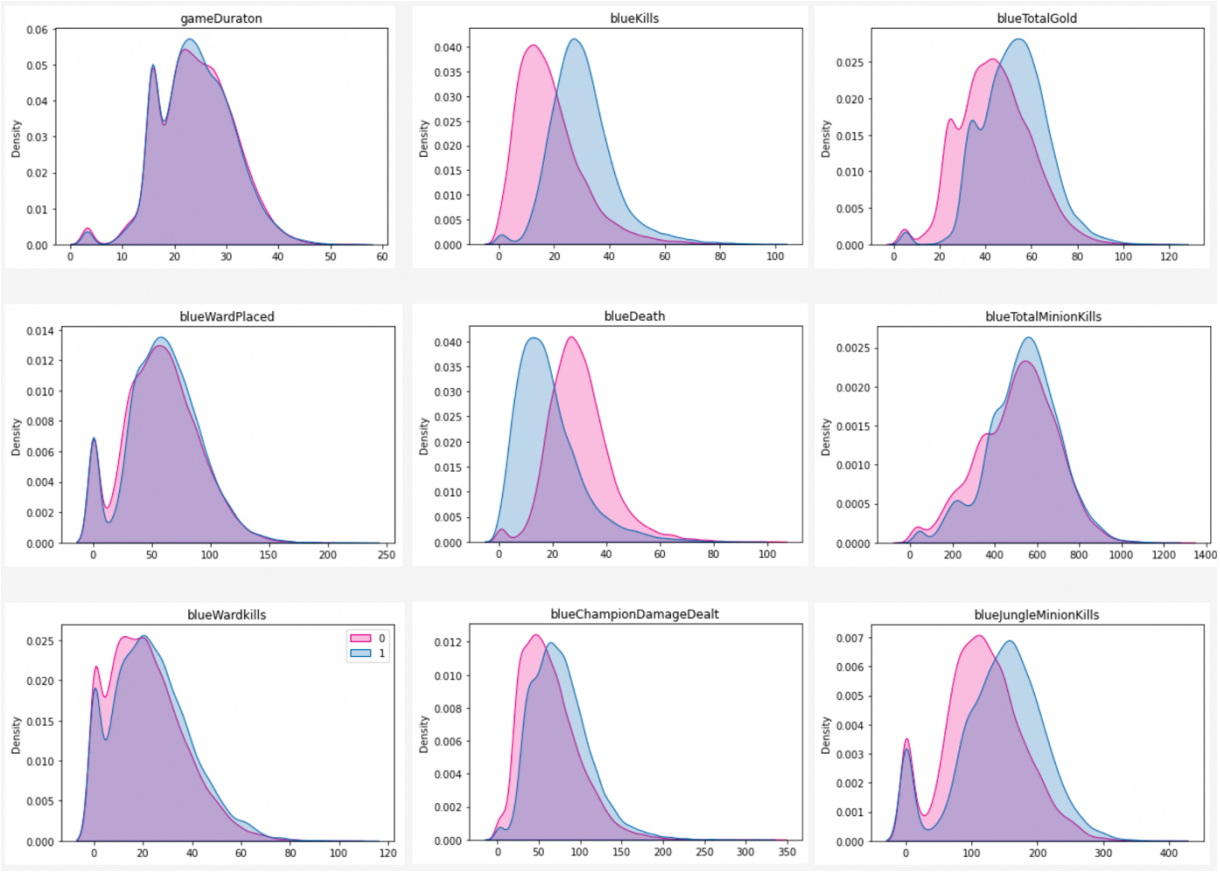
Pic.2. KDF plots.

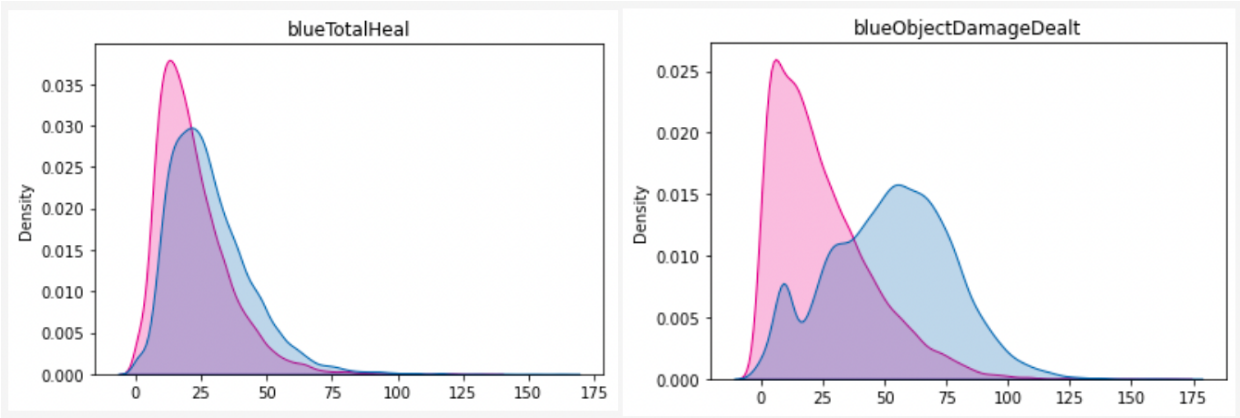
2. Estimation of multivariate mathematical expectation and variance.

Column name	mathematical expectation	variance
gameDuraton	24.14422762414511	49.60321233051361
blueWins	0.5000743383883437	0.25000928711754167
blueWardPlaced	58.64035087719298	982.9231697241096
blueWardkills	22.330136782634554	240.32047448599184
blueKills	24.18941421349985	167.81358009025033
blueDeath	24.180567945286946	170.11615064943672
blueChampionDamageDealt	69.746341882248	1279.0189829913902
blueTotalGold	48.16912990633363	239.51386762751284
blueTotalMinionKills	520.446587867975	30965.83865170302
blueJungleMinionKills	129.586938745168	4181.303298898823
blueTotalHeal	25.050636225096643	228.39203289844016

Pic.3. Multivariate m.e. and var

3. Non-parametric estimation of conditional distributions, mathematical expectations and variances.





Pic.4. NPE visualization.

Win or not	Column name	m.expectation	variance
0	gameDuraton	24.1541908	50.5142995
	blueWardPlaced	57.4631970	985.511211
	blueWardkills	21.2373234	230.256410
	blueKills	18.3136059	136.538866
	blueDeath	29.9173234	131.714927
	blueChampionDamageDealt	63.3016408	1216.63186
	blueTotalGold	43.8896875	239.328282
	blueTotalMinionKills	506.568847	33365.2066
	blueJungleMinionKills	116.153011	3668.88103
	blueTotalHeal	21.7904883	189.946343
	blueObjectDamageDealt	24.5633782	396.906863
1	gameDuraton	24.1342673	48.6958847
	blueWardPlaced	59.8171547	977.638608
	blueWardkills	23.4226252	248.011815
	blueKills	30.0634755	130.066884
	blueDeath	18.4455180	142.722334
	blueChampionDamageDealt	76.1891268	1258.44515
	blueTotalGold	52.4472999	203.103558
	blueTotalMinionKills	534.320202	28184.4455
	blueJungleMinionKills	143.016872	4333.07731
	blueTotalHeal	28.3098148	245.594039
	blueObjectDamageDealt	52.2202759	607.767401

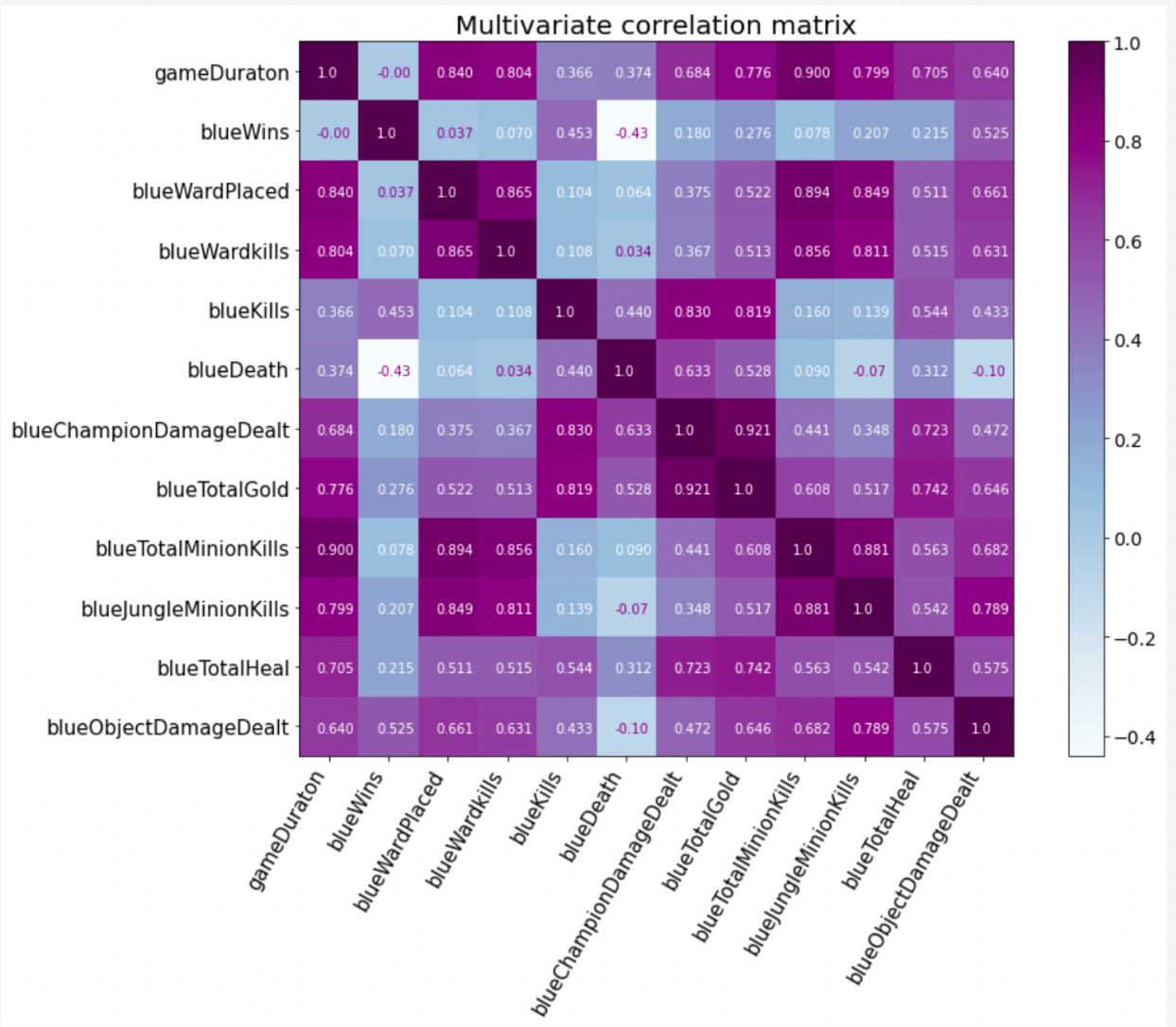
Pic.5. NPE results.

4. Estimation of pair correlation coefficients, confidence intervals for them and significance levels.

Variable	Corr coefficient	Significance level	Confidence interval
blueWardPlaced	0.84068	0.0	[0.840796944012507 ... 0.840572712435002]
blueWardkills	0.80495	0.0	[0.8050915612888215 ... 0.8048223721695263]
blueKills	0.36672	0.0	[0.36705468573493055 ... 0.36639287444607904]
blueDeath	0.37405	0.0	[0.37438825689224364 ... 0.37373060078444564]
blueChampionDamageDealt	0.68465	0.0	[0.6848618641984795 ... 0.684455652023169]
blueTotalGold	0.77683	0.0	[0.7769865142939419 ... 0.7766833112605076]
blueTotalMinionKills	0.90069	0.0	[0.9007719784361153 ... 0.9006276593497923]
blueJungleMinionKills	0.79951	0.0	[0.7996567529951315 ... 0.7993808920589569]
blueTotalHeal	0.70549	0.0	[0.7056893741476609 ... 0.7053053129069643]
blueObjectDamageDealt	0.64038	0.0	[0.6406147030873127 ... 0.640163637310247]

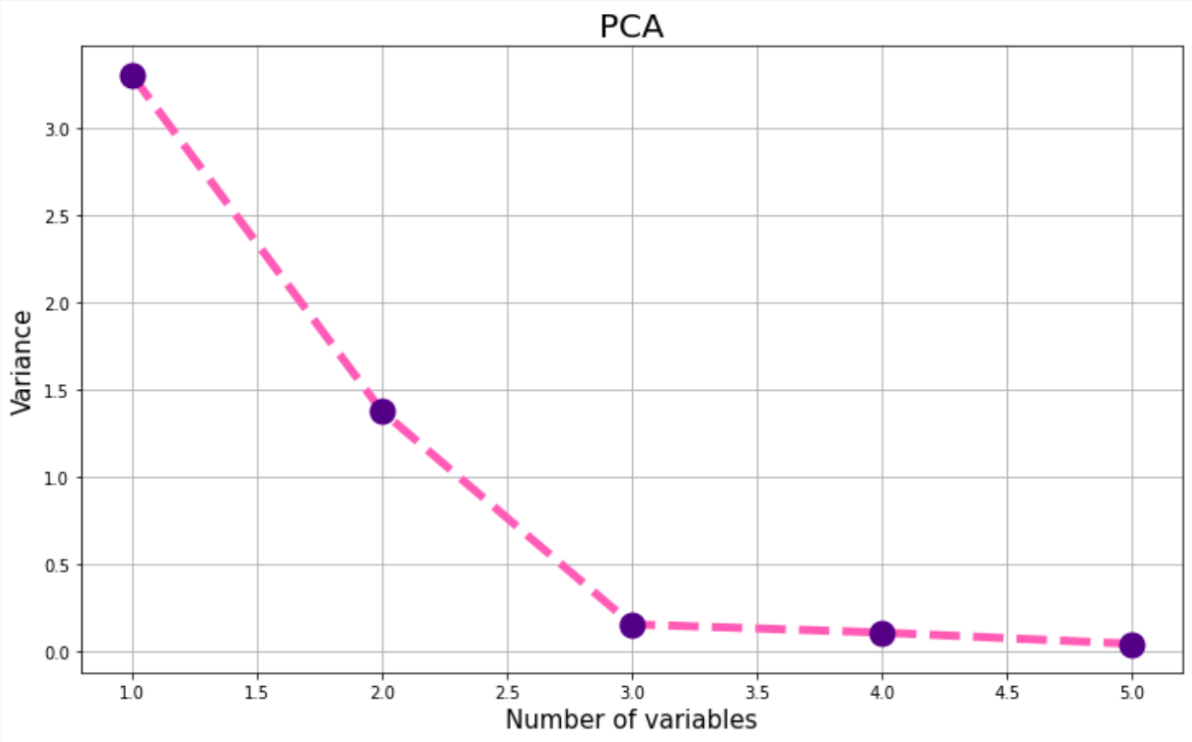
Pic.6. Pair coefficients results.

5. Task formulation for regression, multivariate correlation.



Pic.7. MVCM.

```
[3.30702653 1.37955568 0.15723538 0.11032265 0.04585976]
```



Pic.8. PCA analysis.

PCA algorithm was used in order to reduce feature dimensionality. When the number of components goes from 1 to 3, the decrease in the variance is significant and more variables are not descriptive.

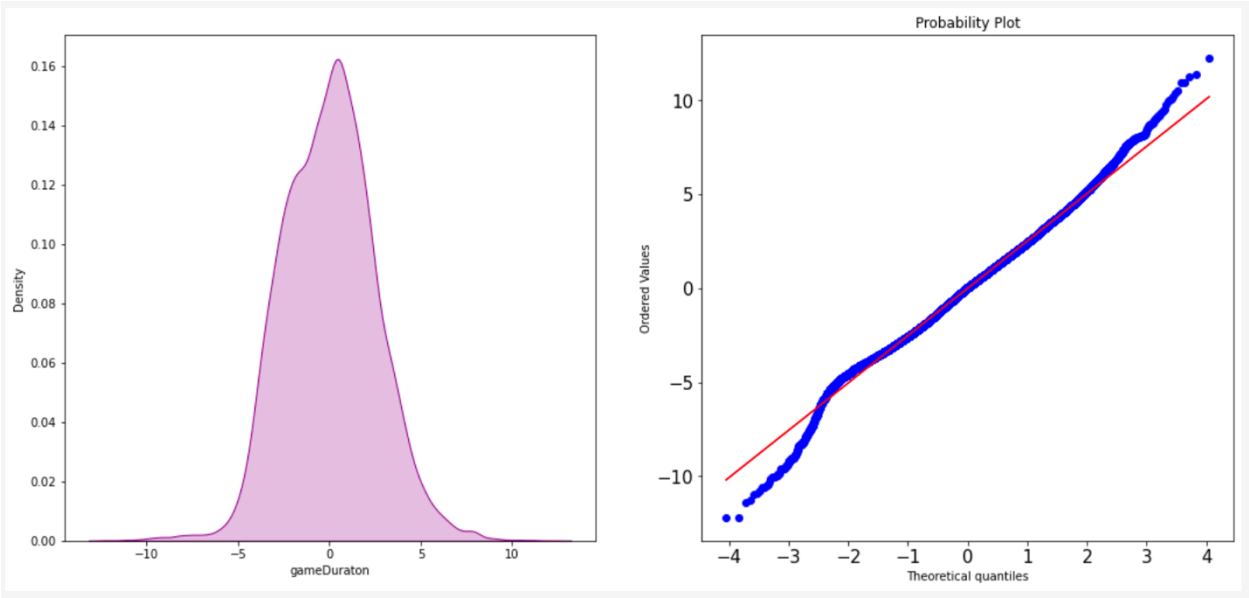
So, the number of chosen variables for the regression problem should be 3. (Further analysis corrected in accordance with the corrected PCA analysis)

6. Regression model, multicollinearity and regularization (if needed).

Type	Alpha	MSE	VAR	Coeff
Least Squares model	-	6.4681834800895075	0.8632681597947957	[0.02669256 0.04958153 0.12906242]
Best Lasso model	0.17200000000000001	6.467917214355652	0.8632737884235885	[0.02684182 0.04870542 0.12793369]
Best Ridge model	1.0	6.468183464621369	0.8632681601217788	[0.02669257 0.04958152 0.12906238]

Pic.9. LSM, Lasso and Ridge models.

7. Quality analysis.



Pic.10. Results visual analysis.

```
Statistic: 14.080
+------+-------+-------------------------------------------------+
|  SL  |   CV  |                       H0                        |
+------+-------+-------------------------------------------------+
| 15.0 | 0.576 | data doesn't look normal (fail to reject H0) |
| 10.0 | 0.656 | data doesn't look normal (fail to reject H0) |
| 5.0  | 0.787 | data doesn't look normal (fail to reject H0) |
| 2.5  | 0.918 | data doesn't look normal (fail to reject H0) |
| 1.0  | 1.092 | data doesn't look normal (fail to reject H0) |
+------+-------+-------------------------------------------------+
KstestResult(statistic=0.22342211976628512, pvalue=0.0)
Residuals are not distributed normally
```

Pic.11. Mathematical results.

Sourcecode:

- 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_2/lab_2.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.

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 publicated in Datasets folder

Participants of the project

- Ivan Dubinin: vandosik
- Alexey Grandilevskii: zer0deck
- Mikhail Sorokin: Mikha1lSorokin