标题：

Test hypothesis: use time zone to judge programmer level to judge code level

Based on theory: good programmers will compile good code

1、Project thinking process:

Since there is no parameter in linux-stable that can indicate the quality of the code, or a tag, we try to define ourselves as a programmer's ability. We build a nonlinear regression model ourselves to represent a person's ability, referred to as Weights. Through this weights, we try to use the personnel in V4 to generate the weights table, and use these data to verify and search our hypothesis, *use time zone to judge programmer level to judge code level*, and calculate the statistical value to verify .

We mainly use the git package in python for operation, and perform CSV data storage and data visualization, packaging and upgrade. Since some calculations in the middle part are just calculation formulas, we directly use the calculator to check the statistical results.

6、Check results：

From the perspective of the world map, to be honest, we don’t have high self-confidence in this model, but in the end, we found a general rule that the world is divided into two parts with Europe as the boundary, and each part contributes to linux The amount is gradually decreasing from west to east

, The correlation coefficient is around 0.6. This may indicate that human capabilities are partially related to the region, but we have no idea how to derive downwards, and we don’t know whether our model process is wrong.

ps. We also found an interesting fact. From the map, it is true that the contribution of developers in economically developed areas is relatively high, such as the East Coast of the United States, China, Japan, and South Korea.

7、Feelings

We tried a variety of methods and processes to consider this "impossible" task, so we also made this result that we don't know if it is correct. The project process is still very interesting. Everyone came up with their strange but brilliant ideas. Of course, we gave up because we couldn't realize it. The difficulty this time is that it is impossible to accurately judge the programmer's ability or code quality. We tried to quantify him and achieve comparison with them, which gave us valuable experience.