Probability Theory and Statistics. Mini-Research Project.

Independent testing in Ukraine - what does it depend on?

The aim of this project is to examine, analyze, and visualize open data about External independent testing in Ukraine.

For this, we'll be using a dataset provided by Ukrainian center for educational quality assessment. You can find and download the archive here: https://zno.testportal.com.ua/opendata?fbclid=IwAR0IIXAhPpj728UHE6JlCCheWRCXAVaUo10QxKDZ359yELKlOSjb5B3RIgU (https://zno.testportal.com.ua/opendata?fbclid=IwAR0IIXAhPpj728UHE6JlCCheWRCXAVaUo10QxKDZ359yELKlOSjb5B3RIgU)

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
%matplotlib inline
import warnings; warnings.simplefilter('ignore')
```

Data Overview

Let's first take a brief overview of the data we have. The dataset includes information about the participants and results of the External independent testing conducted in 2018.

In [2]:
```
DATA = pd.read_csv("OpenData2018.csv", sep=";")
```

In [3]:
```
DATA.head()
```

Out[3]:

	OUTID	Birth	SEXTYPENAME	REGNAME	AREANAME	TERNAME	REGTY
0	a4e039d0-07ef-4a3d-820d-106dcaf01b96	2001	чоловіча	Закарпатська область	Берегівський район	с.Косонь	Випуск заклад загальн середн 2...
1	c1fa1ed0-d2af-4964-9a88-3c138008304b	2001	жіноча	Київська область	Київська область	м.Бровари	Випуск заклад загальн середн 2...
2	6504a7f3-7dfd-4a64-a2d9-4d49f5021303	2000	жіноча	Харківська область	м.Харків	Червонозаводський район міста	Випуск заклад загальн середн 2...
3	5cbcd343-2e63-48b8-b36d-48797a76be31	2000	чоловіча	Рівненська область	Млинівський район	смт Млинів	Випуск заклад загальн середн 2...
4	063c6498-7345-492c-b567-174938bac671	1999	чоловіча	Рівненська область	Рівненська область	м.Острог	Учень (заклад профе (профе

5 rows × 125 columns

The dataset includes information about every participant of independent testing: their id, birth date, sex, region, school, class (specialization), and test results for subjects they passed: test score, and the results in 12- and 200-point grade scale. You can see the detalized description in the attached xls file, *"OpenDataInfo2018.xls"*. For now, we will need just the information about the results of testing. The *grades* dataframe includes just the test results in 200-point scale.

In [4]:
```
grades = DATA[["UkrBall100", "mathBall100", "engBall100", "physBall100", "geoBall100"
, "histBall100",
               'chemBall100', 'bioBall100', 'fraBall100',  'deuBall100',  'spaBall10
0']].replace({0:np.nan})
```

Just calclating the basic statistics impresses. Average grades are, for most of the subjects, below 150 :(

In [5]:
```
grades.describe()
```

Out[5]:

	UkrBall100	mathBall100	engBall100	physBall100	geoBall100	histBall100
count	276433.000000	86692.000000	73321.000000	17331.000000	65493.000000	136810.000000
mean	142.394597	140.756863	145.286657	136.923201	137.526018	136.032874
std	27.450331	27.565387	26.758247	26.499079	24.548281	24.294403
min	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000
25%	119.000000	117.000000	122.000000	113.000000	117.000000	117.000000
50%	139.000000	138.000000	146.000000	131.000000	136.000000	133.000000
75%	165.000000	163.000000	167.000000	157.000000	156.000000	154.000000
max	200.000000	200.000000	200.000000	200.000000	200.000000	200.000000

Fitting Distribution

The first natural assumption that comes to mind is that grades are normally distributed, i.e. most of the people resulted somewhere around 150, there are some who did very well and some who almost failed. Let's take a look at the disributions of grades for some subjects.

Let's take, for example, history. From the 75-percentille, which is only 154, I can guess that most of the people had not wery good results, and the distribution is skewed to the right. My other assumption now is that distribution is exponential. Let's deal with the numbers: we will test H_0 : the grades for the history test are normally distributed vs. H_1: the grades for the history test have exponential distribution.

The Normal distribution is a location-scale family. This means that we need to use a test that is invariant to scale and location. The only meaningful statistics I was able to find is the one mentioned by H.C. Thode. in his **Testing for Normality (https://books.google.com.ua/books?id=gbegXB4SdosC&pg=PA471&lpg=PA471&dq=most+powerful+location+and+scale+invariant+test+grubbs&sourc** (Section 4.2.4):

$$T = \frac{\overline{x} - x_0}{s}$$

- $\overline{x}$ is the sample mean
- x_0 is the minimum observation in the sample
- s is the standard deviation of the sample

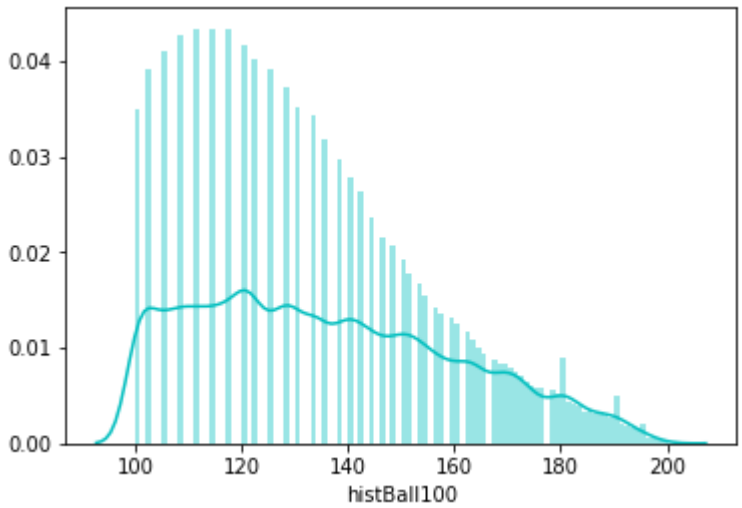
We reject H_0 when test statiscic is large enough. As I found later, this is just a one-sided Grubb's test, which is used to determine outlier. This makes sense for us: performing this test, under the assumption of normality, we are 'checking' if x_0 is the outlier. If it is, then, for us, this means that we reject null hypothesis of normality in favour of exponentiality.

In [6]:
```
dat = grades.histBall100
st = (dat.mean() - dat.min()) / dat.std() # calculate the statistics

print("Test Statistics: ", st)
```

```
Test Statistics:  1.4831759495706975
```

We can reject H_0. Bu I'm still not sure the distribution is exponential. Let's plot it, finally. *(Maybe I should have done it before (and I did), but the dataset is not very informative and I was not able to come up with other meaningfull hypothesis to test)*

In [7]:
```
def plot_distr(subject):
    col = subject+"Ball100"
    dat = DATA[col].replace({0:np.nan}).dropna()
    sns.distplot(dat, bins=100, color="c")

plot_distr("hist") # History
```



It is not Normal. Now, we are not sure we'll be able to fit it to some known distribution, but will try fitting several distributions that are likely to appear here to see what we get.

For this, we first use *fit()*, which, according to docs, gives us the maximum likelihood estimates (MLE) for parameters for each of our possible distributions. After that, when we have the estimated parameters, we can get the theoretical distribution and use Kolmogorov-Smirnov test for goodness of fit. Here, we will test the following hypothesis:

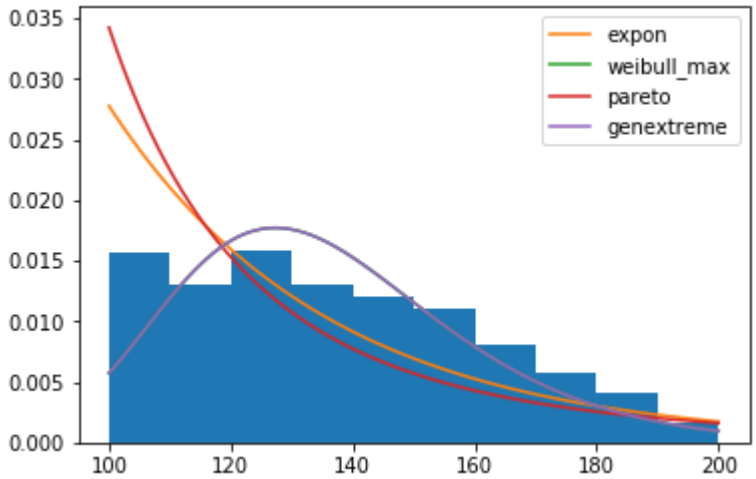
H_0: the two distributions are identical
H_1: the two distributions are different

kstest() performs a test of the distribution of an observed random variable against a given distribution. Under the null hypothesis the two distributions are identical. The alternative hypothesis is 'two-sided'. As a result, we get the p-value and corresponding test statistic, D. The p-value returned by the test is just the same as other p-values :) The D statistic is the absolute maximum distance (supremum) between the CDF's of the two samples. The closer this number is to 0, the more likely it is that the two samples were drawn from the same distribution.

We reject the null hypothesis that the two samples were drawn from the same distribution if the p-value is less than our significance level. *But there is no need to choose significance level here, because we completely fail and reject each of our hopes to fit at least some distribution.* 🙈 Look:

In [8]:
```
def fit_distribution(data):
    distributions = ["expon", "weibull_max", "pareto", "genextreme"]
    dist_results = []
    params = {}

    plt.hist(data, normed=True)
    rX = np.linspace(100,200, 100)

    for dist_name in distributions:
        # fit the distribution
        dist = getattr(scipy.stats, dist_name)
        param = dist.fit(data)
        params[dist_name] = param

        # Use the Kolmogorov-Smirnov test
        D, p = scipy.stats.kstest(data, dist_name, args=param)

        dist_results.append((dist_name, p, D))

        rP = dist.pdf(rX, *param)
        plt.plot(rX, rP, label=dist_name)

        print(dist_name.ljust(16) + ("p-value: "+str(p)).ljust(16) + "D: "+str(D))
    plt.legend()
    plt.show()

    return params, dist_results


params, results = fit_distribution(list(grades.histBall100.dropna()))
```

```
expon           p-value: 0.0    D: 0.1386813749241967
weibull_max     p-value: 0.0    D: 0.06220309695264062
pareto          p-value: 0.0    D: 0.17699573679331132
genextreme      p-value: 0.0    D: 0.06220400661351275
```



As we can see, all the p-values are 0's. This means that either Python is not able to display such small values, or it is *impossible* that our distribution fits any of the tested above. At leat, we can find the closest one, considering CDF's.

In [9]:
```
# find minimum distance between the CDF's of the two samples
best_dist, best_p, best_d = (min(results, key=lambda item: item[2]))

print("Most close fitting distribution: "+str(best_dist))
print("It's p-value: "+ str(best_p))
print("Parameters for the most close fit: "+ str(params[best_dist]))
```

```
Most close fitting distribution: weibull_max
It's p-value: 0.0
Parameters for the most close fit: (11.471008156603279, 364.66400620282, 239.2674956
4014898)
```

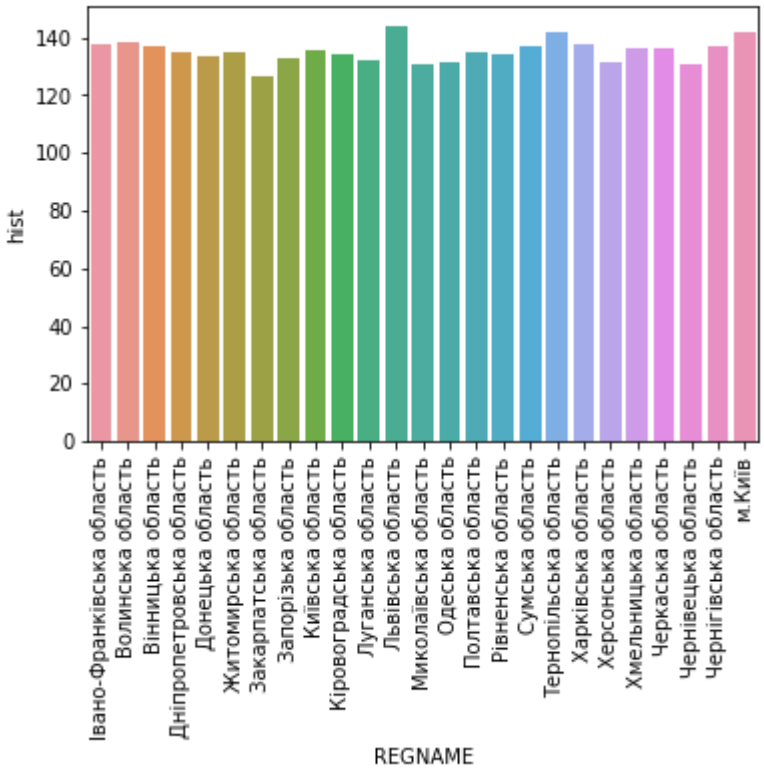
The closest distribution is **Frechet left (https://en.wikipedia.org/wiki/Fr%C3%A9chet_distribution)** (or Weibull maximum), which is from exponential family. But we can not say that this is the distribution of our grades.

The only hypothesis we accept is that Ukrainian students are unique and do not fit any known distribution.

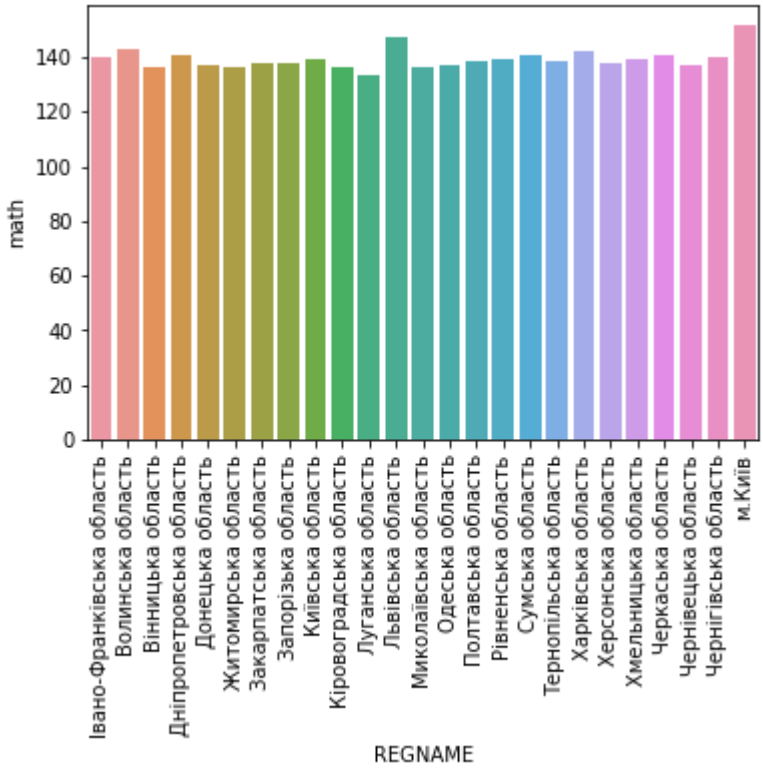
Leading Regions

Now, let's take a look at some other interesting facts we have noticed. For example, this is interesting that Kyiv city and Lviv region perform better, on average, for almost every subject, that other reegions of the country.

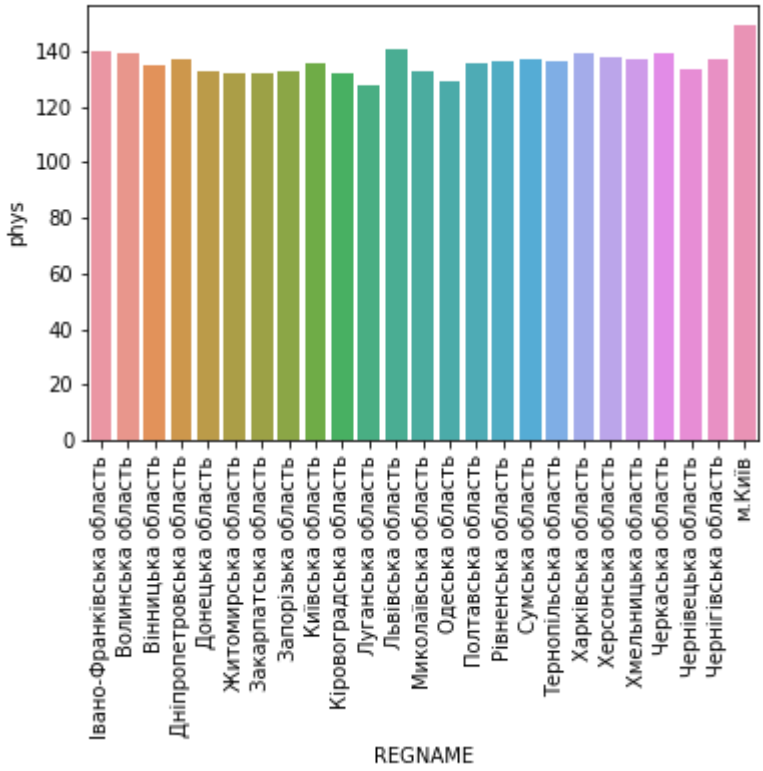
In [10]:
```
regdata = DATA[["REGNAME", "UkrBall100", "mathBall100", "engBall100", "physBall100",
"geoBall100", "histBall100",
               'chemBall100', 'bioBall100', 'fraBall100',  'deuBall100',  'spaBall10
0']].replace({0:np.nan})

def plot_region_avg(sub):
    col1 = "REGNAME"
    col2 = sub +"Ball100"
    ss = regdata[[col1, col2]].dropna() # select subject and clean data
    b1, b2 = [], []

    for region in sorted(set(ss[col1])):
        people = ss[ss[col1] == region]
        mean = people[col2].mean()
        b1.append(region)
        b2.append(mean)
        #print(region, mean)
    regData = pd.DataFrame( {col1: b1, sub: b2})
    regplot = sns.barplot(x="REGNAME", y=sub, data=regData)
    regplot.set_xticklabels(regplot.get_xticklabels(), rotation=90)
    plt.show()

plot_region_avg("hist")
```



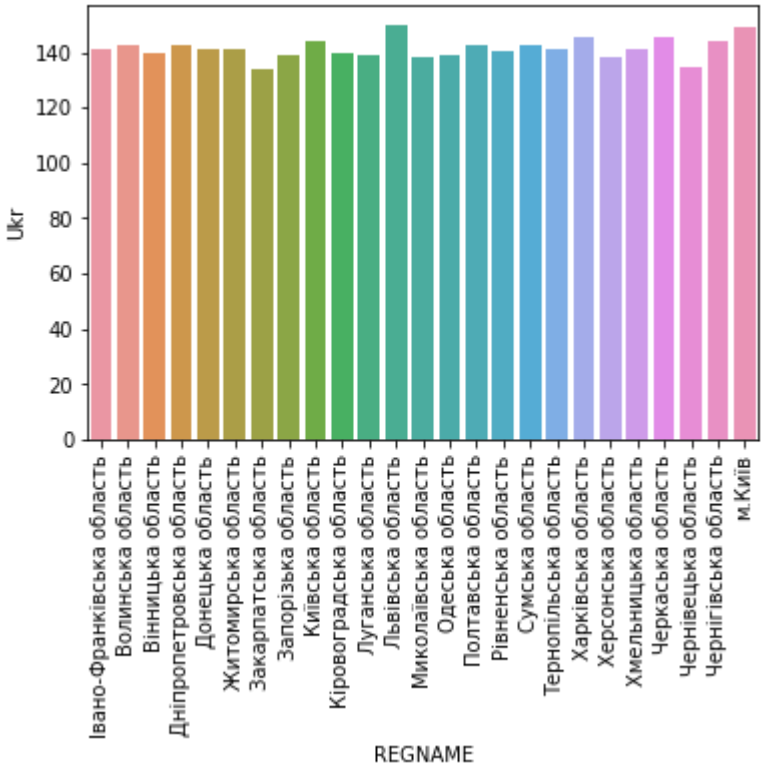
In [11]:
```
plot_region_avg("math")
```



In [12]:
```
plot_region_avg("phys")
```



In [13]:
```
plot_region_avg("Ukr")
```

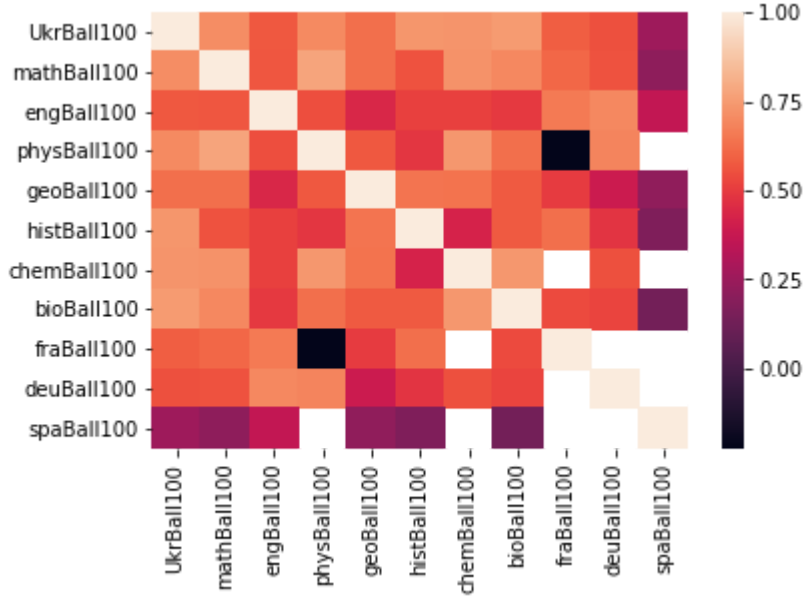


(Sorry for plotting it all separately, one by one)

Correlations

Let's take a look at the correlation between grades for several subjects.

In [14]:
```
sns.heatmap(grades.corr())
plt.show()
```



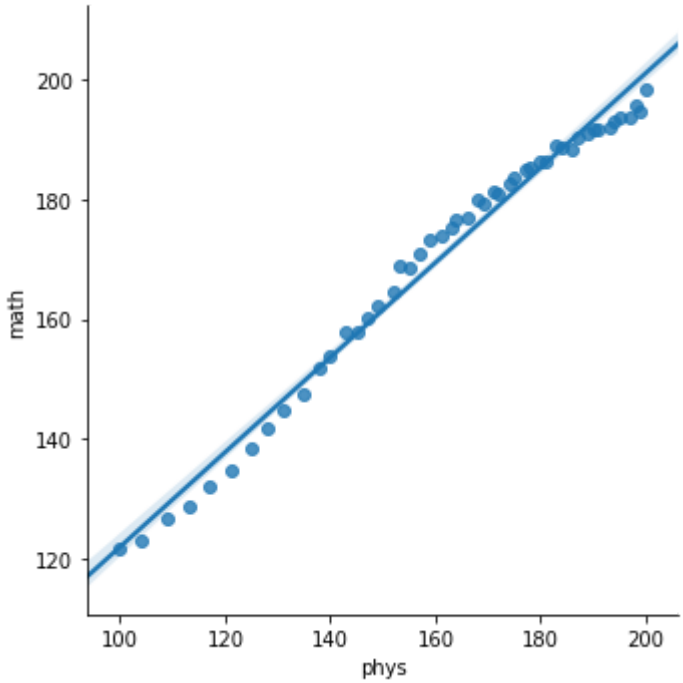
We can see that people who wrote mathematics well enough have good results for physics, too. So, our hypothesis is linear dependency between these two grades. To test it, we need to select people who participated in testing in both of these subjects and run a linear regression.

In [15]:
```
# physics and mathematics

df = DATA[np.isfinite(DATA['physBall100'])]
df = df[np.isfinite(df['mathBall100'])]
scipy.stats.linregress(df.mathBall100, df.physBall100)
```

Out[15]:
```
LinregressResult(slope=0.6042171183742883, intercept=39.25736364855727, rvalue=0.639
9468705820024, pvalue=0.0, stderr=0.0052183278735903355)
```

We can see that there is a slope, and the *rvalue*, which is a correlation coefficient, is even greater than 0.5 for mathematics and physics. The grades for these two subjects have linear dependency.

In [16]:
```
def lin(sub1, sub2):
    col1 = sub1+"Ball100"
    col2 = sub2+"Ball100"
    ss = DATA[[col1, col2]].replace({0:np.nan}).dropna() #select two subjects and cle
an data
    b1, b2 = [], []
    for ball in sorted(set(ss[col1])):
        people = ss[ss[col1] == ball]
        mean = people[col2].mean()
        b1.append(ball)
        b2.append(mean)
    return pd.DataFrame( {sub1: b1, sub2: b2})

s1, s2 = "phys", "math"
df = lin(s1, s2)
sns.lmplot(s1, s2, df)
plt.show()
```

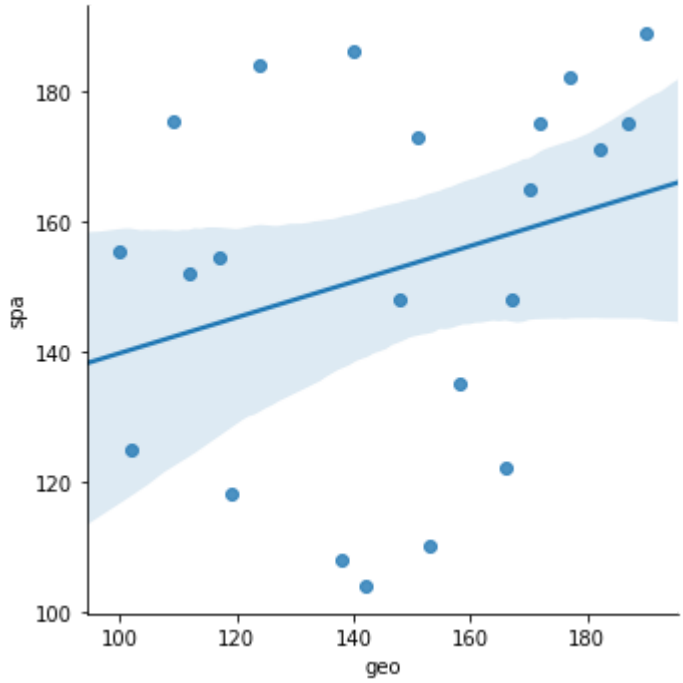


In [17]:
```
# geography and spanish

df = DATA[np.isfinite(DATA['spaBall100'])]
df = df[np.isfinite(df['geoBall100'])]
scipy.stats.linregress(df.spaBall100, df.geoBall100)
```

Out[17]:
```
LinregressResult(slope=0.079246816502899, intercept=114.9903040259158, rvalue=0.0928
6813714393391, pvalue=0.6072312734899009, stderr=0.1525995794607709)
```

Here, regression result shows us that there is no dependency linear dependency between these two subjects, i.e. we would not be able to predict the results for spanish based on results for geography test.

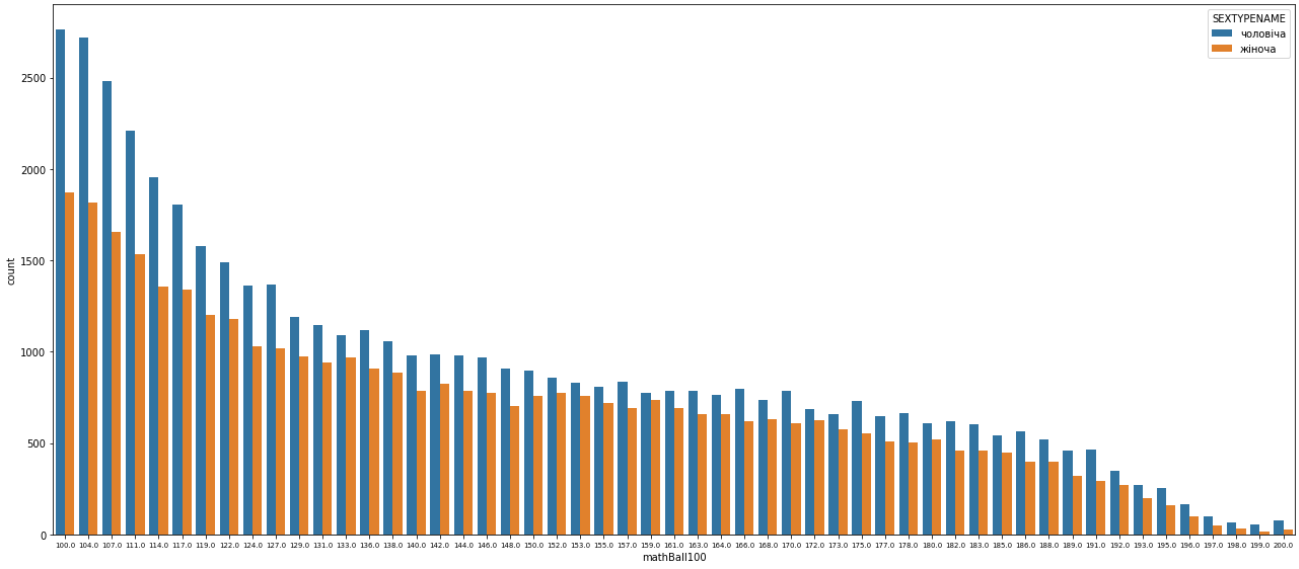
In [18]:
```
s1, s2 = "geo", "spa"
df = lin(s1, s2)
sns.lmplot(s1, s2, df)
plt.show()
```

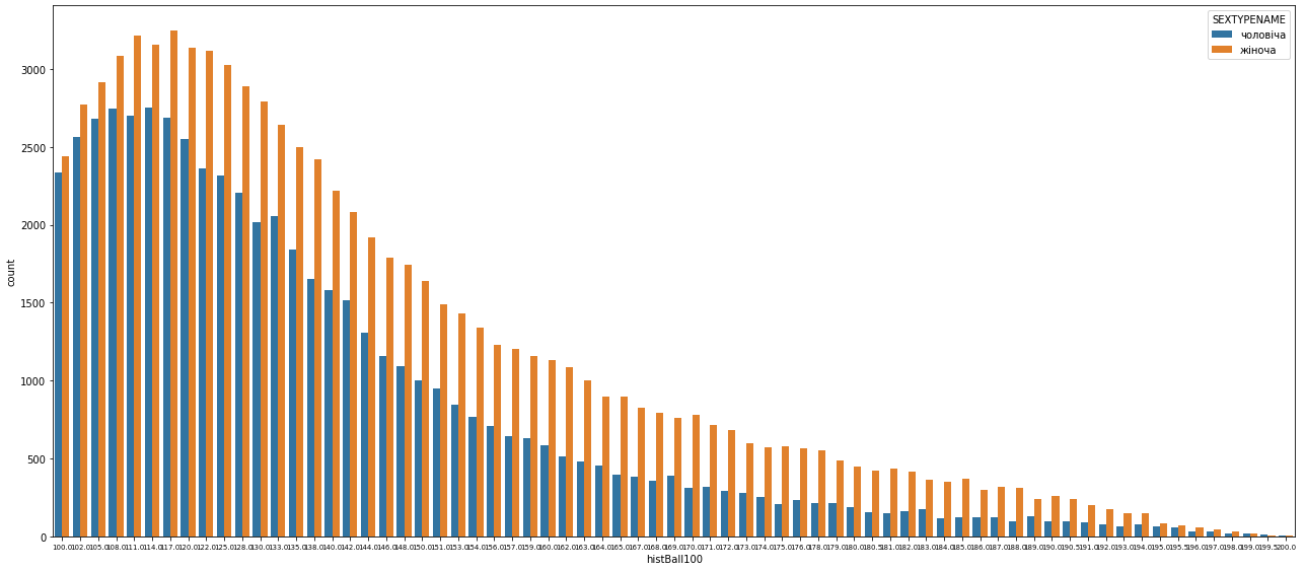


Boys vs. Girls

Some other interesting facts about boys and girls. For subjects, such as history or mathematics, boys and girls have basically the same results.

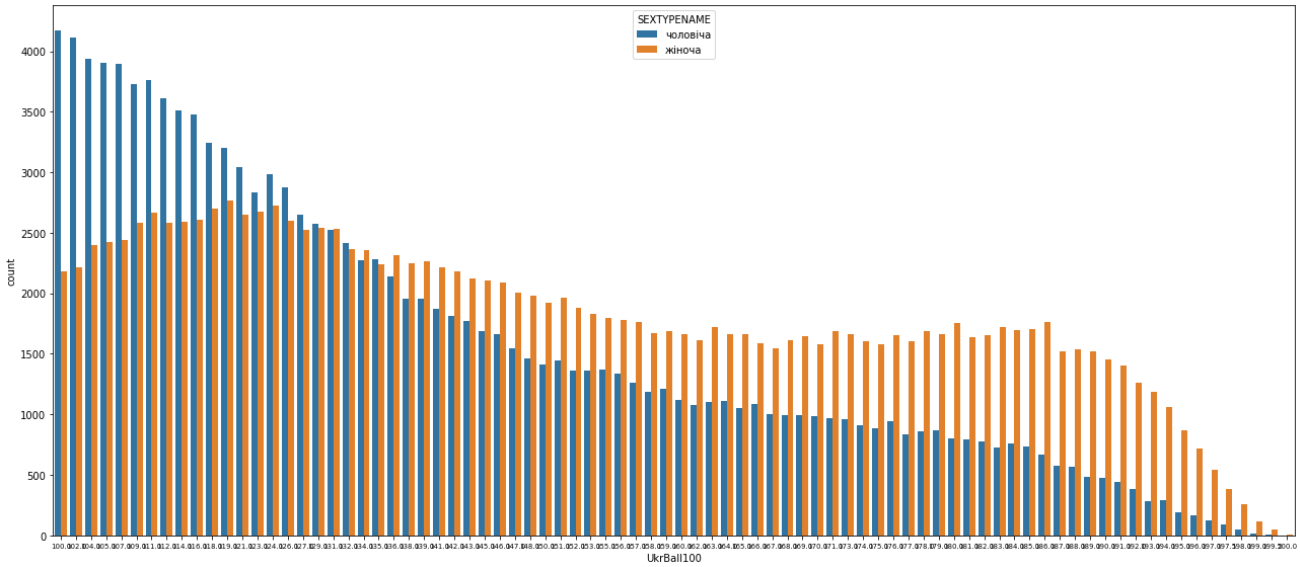
In [19]:
```
df = DATA.replace({0:np.nan})

def plot_subject(subject):
    count = ["SEXTYPENAME", "ClassProfileNAME", "ClassLangName"]
    plt.figure(figsize=(18,8))
    ax = sns.countplot(x=subject+'Ball100', hue=count[0], data=df, orient="h")
    #ax = sns.countplot(x="SEXTYPENAME", hue=subject+'TestStatus', data=DATA, orient
="h")
    ax.set_xticklabels(ax.get_xticklabels(), fontsize=7)
    plt.tight_layout()
    plt.show()

plot_subject("math")
plot_subject("hist")
```





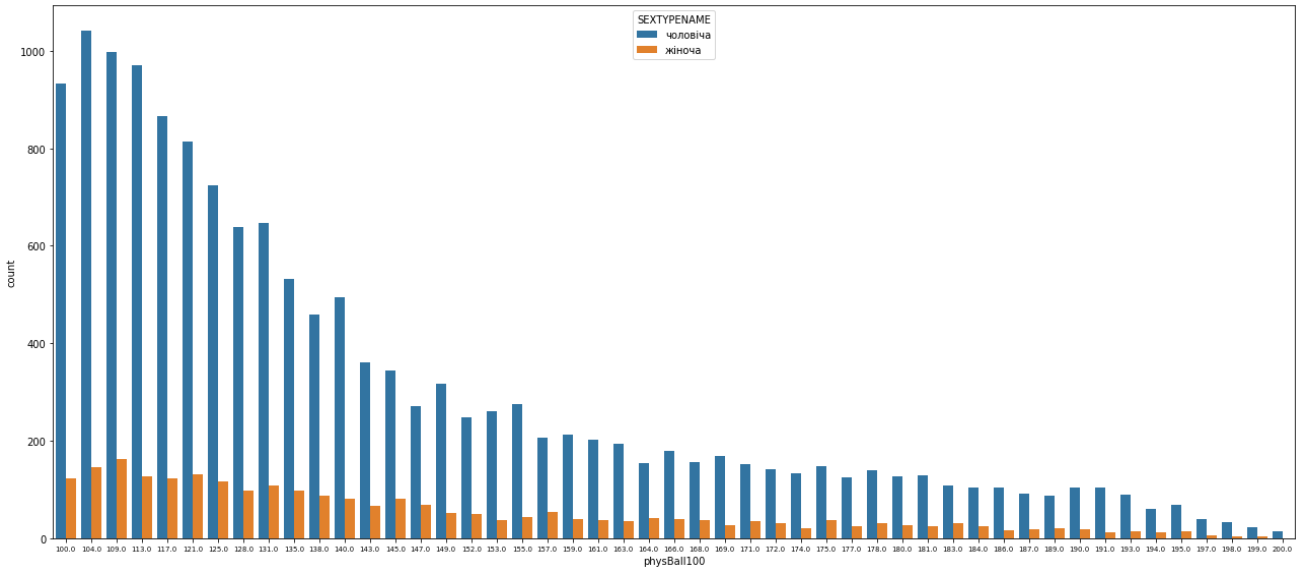
But for Ukrainian language, girls perform better.

In [20]:
```
plot_subject("Ukr")
```



What about physics? Unfortunately, girls do not like physics.

In [21]:
```
plot_subject("phys")
```



Conclusions

Working with real-world data is hard. For example, I was not able to just "drop all NaN's" at the very beginning to have clear data, because this way I would have dropped everything from the table. The data is not stored very vell; there are some mistakes in the recordings, problems with outliers. I had to clean parts of the data every time I needed it.

From the very beginning, when the research project was announced, and we were told to find some data ourselves, I knew that want to analyze some Ukrainian open data. There are many messy datasets over **here (https://data.gov.ua/)**, but the topic that grabbed my attention was education. It was also hard to come from simple data observation to making a hypothesis and finding ways to test them, to come up with some reasonable ideas. Few times I just wanted to leave it all and switch to some typical dataset from Kaggle, run a linear regression, make predictions, and that's it.

I guess that some of the hypothesis proposed above may probably be irrelevant. For example, running linear regression for grades for related subjects. It is ok to talk about correlation in this case, but is it ok to talk about dependency? Anyway, the results gave some extra info and, what's more important, *practice*.

It is hard to fit real-world data to a known distribution. Maybe, it's unique for this particular case. But, as one can see above, the distribution isn't looking the way I expected it to be.

Simply calculating basic statistics for every subject gives us a good overview of the data we have. Even though finding out the results of the testing was, literally, hurtful (i.e. my expectations for average grades were higher), some hypothesis were accepted. For example, we approved relations for grades for the natural sciences (e.g. mathematics and physics) and showed that there is no correlation between grades for Spanish language and geography.

Another conclusion: sometimes just visualizations can help — most of the times. As for me, the last bar plot is a great example of it. I was shocked when I saw how little girls are passing tests in physics.

The other strange observation is that the two regions which perform the best are Lviv region and Kyiv city. It would be interesting to find the reasons for such results.

To sum up, this project gave me the opportunity to practice hypothesis testing and practice approaches to working with data, finally understand what is p-value, find out about 80+ distributions that are implemented in SciPy and that I haven't heard about before.

It also lead me to some thoughts about our educational system, in general. In my opinion, the testing results presented here are bad. We need to find reasons for such results. We also need to think about the fact that all the pupils go to universities after schools. But the question is **for what reason** most of them are doing this? We should concentrate on quality, not quantity.