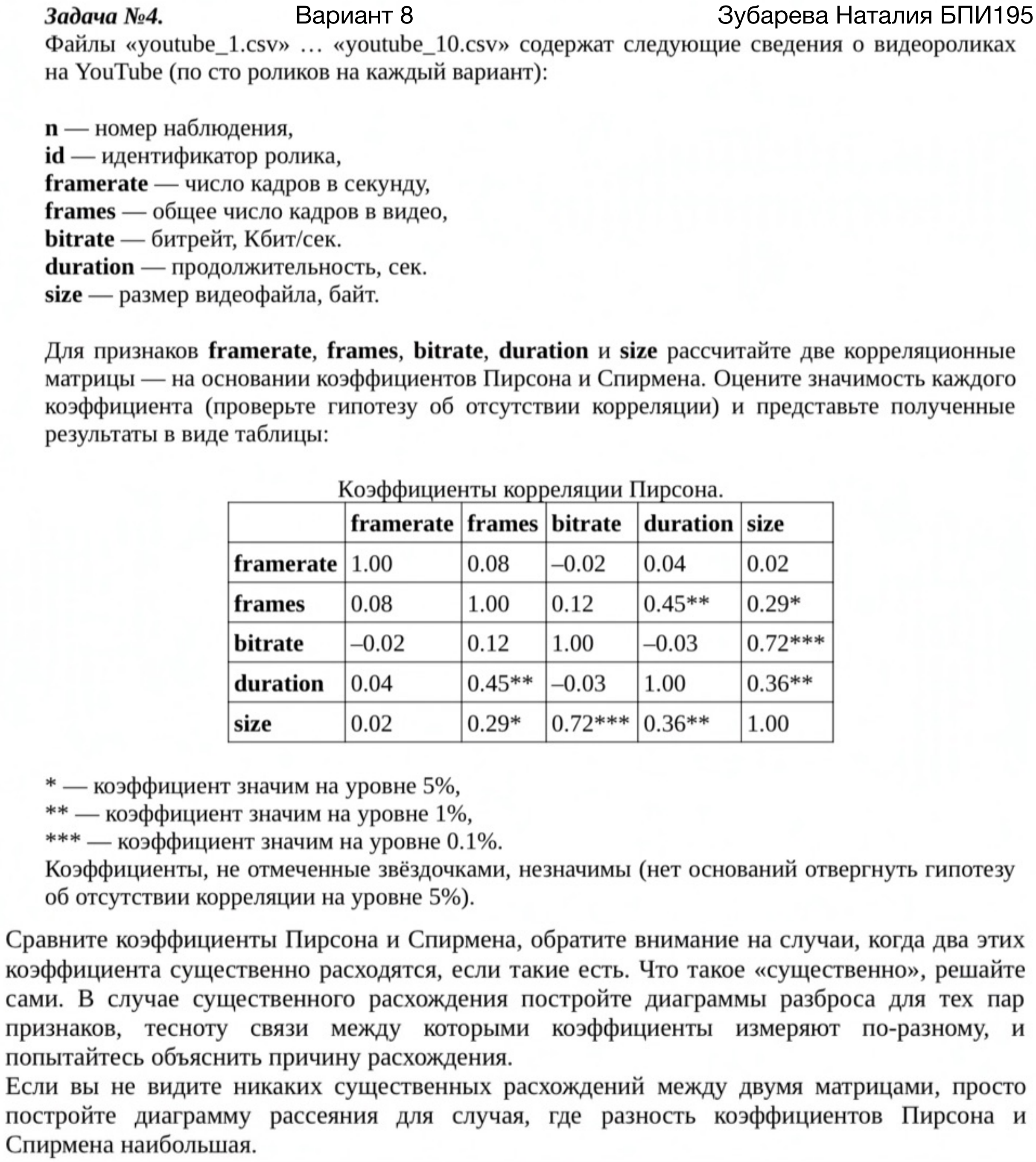
Вариант 8 Зубарева Наталия БПИ195



Задача №4.

Файлы «youtube_1.csv» … «youtube_10.csv» содержат следующие сведения о видеороликах на YouTube (по сто роликов на каждый вариант):

n — номер наблюдения,
id — идентификатор ролика,
framerate — число кадров в секунду,
frames — общее число кадров в видео,
bitrate — битрейт, Кбит/сек.
duration — продолжительность, сек.
size — размер видеофайла, байт.

Для признаков **framerate**, **frames**, **bitrate**, **duration** и **size** рассчитайте две корреляционные матрицы — на основании коэффициентов Пирсона и Спирмена. Оцените значимость каждого коэффициента (проверьте гипотезу об отсутствии корреляции) и представьте полученные результаты в виде таблицы:

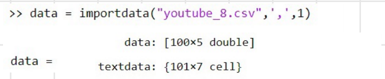
Коэффициенты корреляции Пирсона.

	framerate	**frames**	**bitrate**	**duration**	**size**
framerate	1.00	0.08	–0.02	0.04	0.02
frames	0.08	1.00	0.12	0.45**	0.29*
bitrate	–0.02	0.12	1.00	–0.03	0.72***
duration	0.04	0.45**	–0.03	1.00	0.36**
size	0.02	0.29*	0.72***	0.36**	1.00

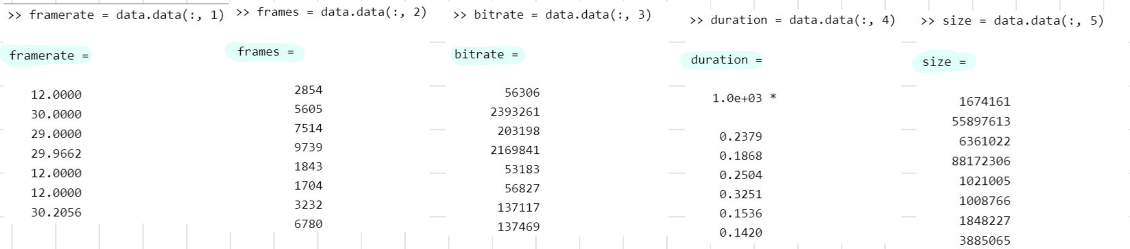
* — коэффициент значим на уровне 5%,
** — коэффициент значим на уровне 1%,
*** — коэффициент значим на уровне 0.1%.
Коэффициенты, не отмеченные звёздочками, незначимы (нет оснований отвергнуть гипотезу об отсутствии корреляции на уровне 5%).

Сравните коэффициенты Пирсона и Спирмена, обратите внимание на случаи, когда два этих коэффициента существенно расходятся, если такие есть. Что такое «существенно», решайте сами. В случае существенного расхождения постройте диаграммы разброса для тех пар признаков, тесноту связи между которыми коэффициенты измеряют по-разному, и попытайтесь объяснить причину расхождения.
Если вы не видите никаких существенных расхождений между двумя матрицами, просто постройте диаграмму рассеяния для случая, где разность коэффициентов Пирсона и Спирмена наибольшая.

Сначала считаем данные из файла csv. Matlab делает это с разделением данных на 2 массива: textdata и data, в первом находятся заголовки столбцов, id и названия видео, которые нам в принципе не нужны, во втором - столбцы значений framerate, frames, bitrate, duration и size, выделим их в отдельные массивы



```
>> data = importdata("youtube_8.csv",',',1)

              data: [100×5 double]
data =
          textdata: {101×7 cell}
```



```
>> framerate = data.data(:, 1)

framerate =

   12.0000
   30.0000
   29.0000
   29.9662
   12.0000
   12.0000
   30.2056
```

```
>> frames = data.data(:, 2)

frames =

        2854
        5605
        7514
        9739
        1843
        1704
        3232
        6780
```

```
>> bitrate = data.data(:, 3)

bitrate =

       56306
     2393261
      203198
     2169841
       53183
       56827
      137117
      137469
```

```
>> duration = data.data(:, 4)

duration =

   1.0e+03 *


    0.2379
    0.1868
    0.2504
    0.3251
    0.1536
    0.1420
```

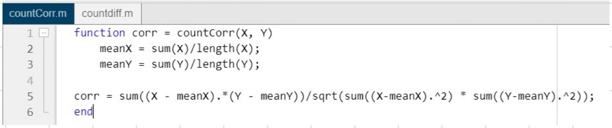
```
>> size = data.data(:, 5)

size =

     1674161
    55897613
     6361022
    88172306
     1021005
     1008766
     1848227
     3885065
```

Рассчитаем для требуемых параметров корреляционные матрицы, для этого будем сначала использовать коэффициент Пирсона, затем Спирмена

Выборочный коэффициент Пирсона рассчитывается по следующей формуле:

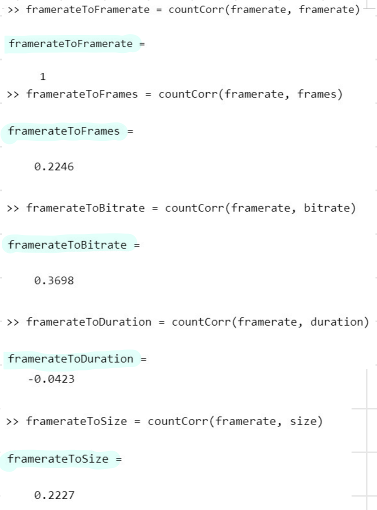
$$r_{X,Y} = \hat{\mathrm{Corr}}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \times \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

Напишем функцию и применим ее к парам массивов, результаты занесём в таблицу

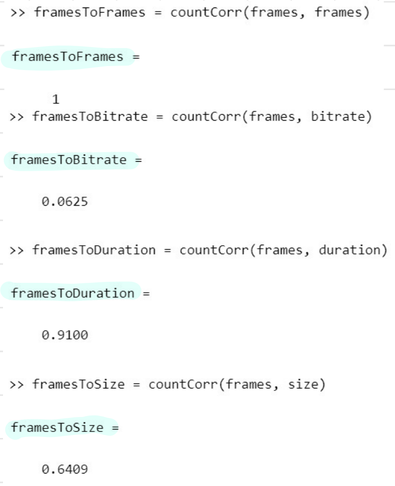


countCorr.m | countdiff.m

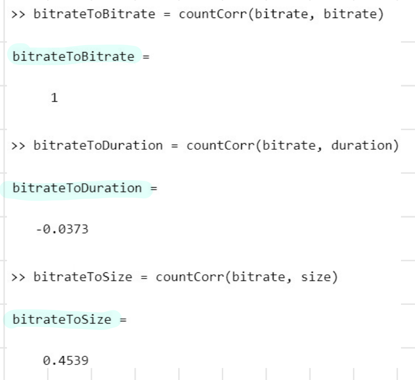
```
function corr = countCorr(X, Y)
    meanX = sum(X)/length(X);
    meanY = sum(Y)/length(Y);

corr = sum((X - meanX).*(Y - meanY))/sqrt(sum((X-meanX).^2) * sum((Y-meanY).^2));
end
```



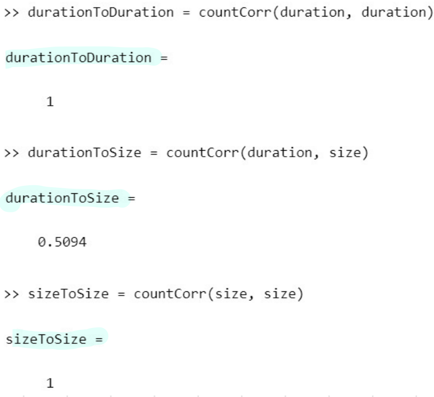
```
>> framerateToFramerate = countCorr(framerate, framerate)

framerateToFramerate =

     1
>> framerateToFrames = countCorr(framerate, frames)

framerateToFrames =

    0.2246

>> framerateToBitrate = countCorr(framerate, bitrate)

framerateToBitrate =

    0.3698

>> framerateToDuration = countCorr(framerate, duration)

framerateToDuration =
   -0.0423

>> framerateToSize = countCorr(framerate, size)

framerateToSize =

    0.2227
```



```
>> framesToFrames = countCorr(frames, frames)

framesToFrames =

     1
>> framesToBitrate = countCorr(frames, bitrate)

framesToBitrate =

    0.0625

>> framesToDuration = countCorr(frames, duration)

framesToDuration =

    0.9100

>> framesToSize = countCorr(frames, size)

framesToSize =

    0.6409
```



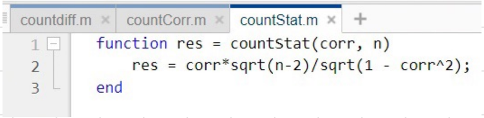
```
>> bitrateToBitrate = countCorr(bitrate, bitrate)

bitrateToBitrate =

     1

>> bitrateToDuration = countCorr(bitrate, duration)

bitrateToDuration =

   -0.0373

>> bitrateToSize = countCorr(bitrate, size)

bitrateToSize =

    0.4539
```



```
>> durationToDuration = countCorr(duration, duration)

durationToDuration =

     1

>> durationToSize = countCorr(duration, size)

durationToSize =

    0.5094

>> sizeToSize = countCorr(size, size)

sizeToSize =

     1
```

Проверим гипотезы о наличии корреляции:
H0 - выборки независимы - основная гипотеза
HA - выборки зависимы - альтернативная гипотеза

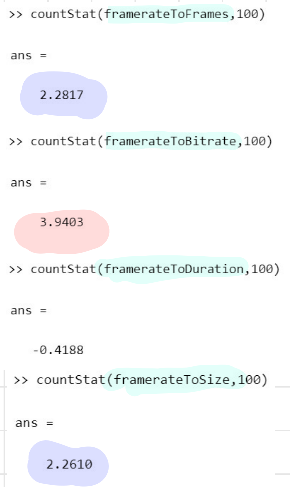
Сравним квантили и рассчитанные по коэффициентам статистики $t = \frac{r_{X,Y}\sqrt{n-2}}{\sqrt{1-r^2_{X,Y}}} \overset{H_0}{\sim} t_{n-2}$ - распределение Стьюдента
Если коэффициент по модулю больше значения квантили t (n-1, α/2), отвергаем H0 и считаем выборки зависимыми, в нашем случае n = 100, α1 = 0.05, α2 = 0.01, α3 = 0.001



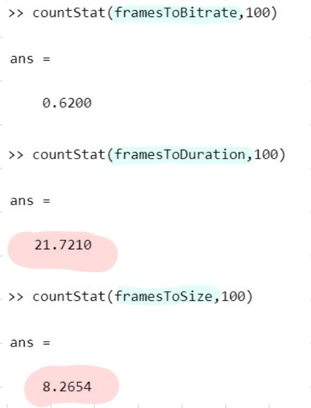
```
function res = countStat(corr, n)
    res = corr*sqrt(n-2)/sqrt(1 - corr^2);
end
```

t(99, 0.025) = 1.96
t(99, 0.005) = 2.576
t(99, 0.0005) = 3.291

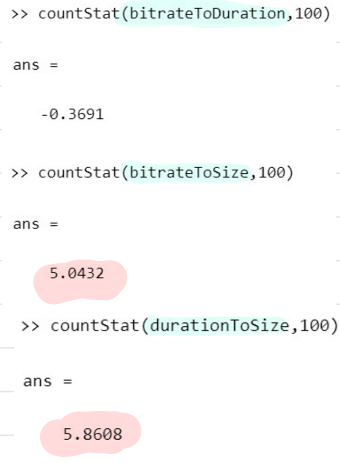
Теперь вычислим статистики для каждого коэффициента и сравним с квантилями



```
>> countStat(framerateToFrames,100)

ans =

    2.2817

>> countStat(framerateToBitrate,100)

ans =

    3.9403

>> countStat(framerateToDuration,100)

ans =

   -0.4188

>> countStat(framerateToSize,100)

ans =

    2.2610
```



```
>> countStat(framesToBitrate,100)

ans =

    0.6200

>> countStat(framesToDuration,100)

ans =

   21.7210

>> countStat(framesToSize,100)

ans =

    8.2654
```



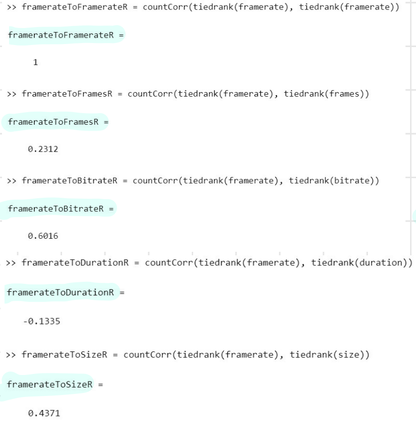
```
>> countStat(bitrateToDuration,100)

ans =

   -0.3691

>> countStat(bitrateToSize,100)

ans =

    5.0432

>> countStat(durationToSize,100)

ans =

    5.8608
```

Коэффициенты корреляции Пирсона

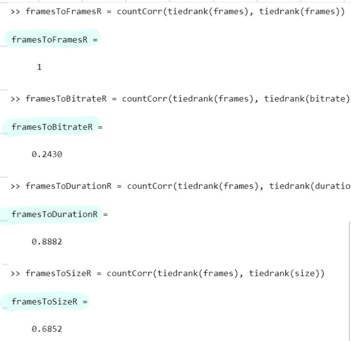
	framerate	frames	bitrate	duration	size
framerate	1	0.2246 *	0.3698 ***	-0.0423	0.2227 *
frames	0.2246 *	1	0.0625	0.9100 ***	0.6409 ***
bitrate	0.3698 ***	0.0625	1	-0.0373	0.4539 ***
duration	-0.424	0.9100 ***	-0.0373	1	0.5094 ***
size	0.2227 *	0.6409 ***	0.4539 ***	0.5094 ***	1

* - коэффициент значим на уровне 5%
** - коэффициент значим на уровне 1%
*** - коэффициент значим на уровне 0.1%

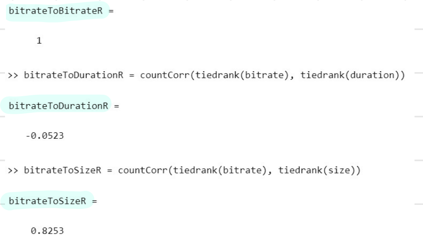
Для проверки гипотез через коэффициент Спирмена оставим все так же, кроме самих коэффициентов: там будем считать корреляцию не для самих значений, а для их рангов:



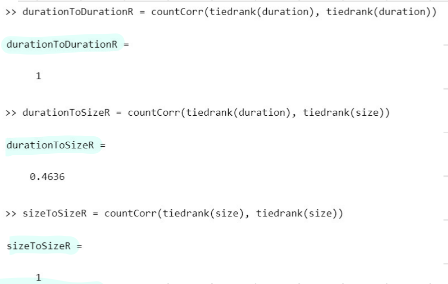
```
>> framerateToFramerateR = countCorr(tiedrank(framerate), tiedrank(framerate))

framerateToFramerateR =

     1

>> framerateToFramesR = countCorr(tiedrank(framerate), tiedrank(frames))

framerateToFramesR =

    0.2312

>> framerateToBitrateR = countCorr(tiedrank(framerate), tiedrank(bitrate))

framerateToBitrateR =

    0.6016

>> framerateToDurationR = countCorr(tiedrank(framerate), tiedrank(duration))

framerateToDurationR =

   -0.1335

>> framerateToSizeR = countCorr(tiedrank(framerate), tiedrank(size))

framerateToSizeR =

    0.4371
```



```
>> framesToFramesR = countCorr(tiedrank(frames), tiedrank(frames))

framesToFramesR =

     1

>> framesToBitrateR = countCorr(tiedrank(frames), tiedrank(bitrate))

framesToBitrateR =

    0.2430

>> framesToDurationR = countCorr(tiedrank(frames), tiedrank(duration))

framesToDurationR =

    0.8882

>> framesToSizeR = countCorr(tiedrank(frames), tiedrank(size))

framesToSizeR =

    0.6852
```

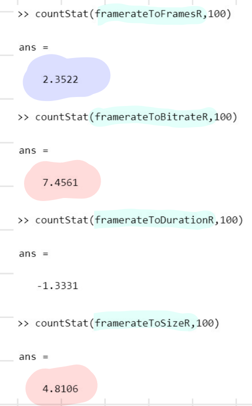


```
bitrateToBitrateR =

     1

>> bitrateToDurationR = countCorr(tiedrank(bitrate), tiedrank(duration))

bitrateToDurationR =

   -0.0523

>> bitrateToSizeR = countCorr(tiedrank(bitrate), tiedrank(size))

bitrateToSizeR =

    0.8253
```

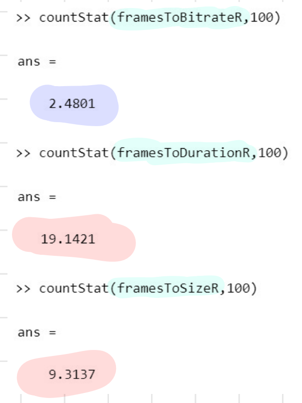


```
>> durationToDurationR = countCorr(tiedrank(duration), tiedrank(duration))

durationToDurationR =

     1

>> durationToSizeR = countCorr(tiedrank(duration), tiedrank(size))

durationToSizeR =

    0.4636

>> sizeToSizeR = countCorr(tiedrank(size), tiedrank(size))

sizeToSizeR =

     1
```

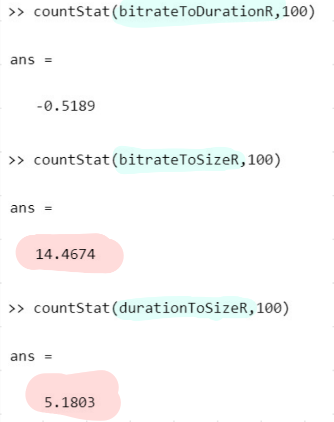
Квантили остаются прежними, считаем статистики:



```
>> countStat(framerateToFramesR,100)

ans =

    2.3522

>> countStat(framerateToBitrateR,100)

ans =

    7.4561

>> countStat(framerateToDurationR,100)

ans =

   -1.3331

>> countStat(framerateToSizeR,100)

ans =

    4.8106
```



```
>> countStat(framesToBitrateR,100)

ans =

    2.4801

>> countStat(framesToDurationR,100)

ans =

   19.1421

>> countStat(framesToSizeR,100)

ans =

    9.3137
```



```
>> countStat(bitrateToDurationR,100)

ans =

   -0.5189

>> countStat(bitrateToSizeR,100)

ans =

   14.4674

>> countStat(durationToSizeR,100)

ans =

    5.1803
```

Коэффициенты корреляции Спирмена

	framerate	frames	bitrate	duration	size
framerate	1	0.2312 *	0.6016 ***	-0.1335	0.4371 ***
frames	0.2312 *	1	0.2430 *	0.8882 ***	0.6852 ***
bitrate	0.6016 ***	0.2430 *	1	-0.0523	0.8253 ***
duration	-0.1335	0.8882 ***	-0.0523	1	0.4636 ***
size	0.4371 ***	0.6852 ***	0.8253 ***	0.4636 ***	1

* - коэффициент значим на уровне 5%

** - коэффициент значим на уровне 1%

*** - коэффициент значим на уровне 0.1%

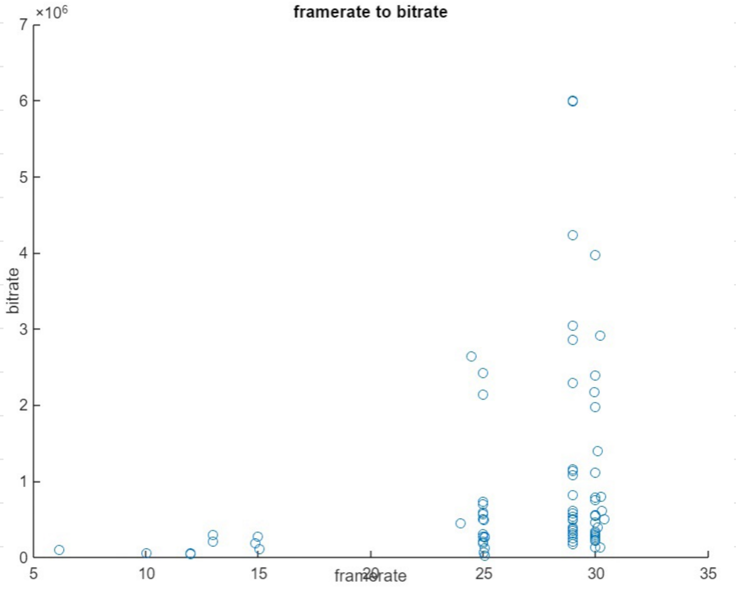
Теперь сравним то, что мы получили

Для каждой пары соответствующих коэффициентов Пирсона и Спирмена найдём, какой процент от соответствующего коэффициента Пирсона составляет модуль их разности (это кажется довольно осознанным способом сравнивать). Будем считать, что разница существенная, если она превосходит 50%

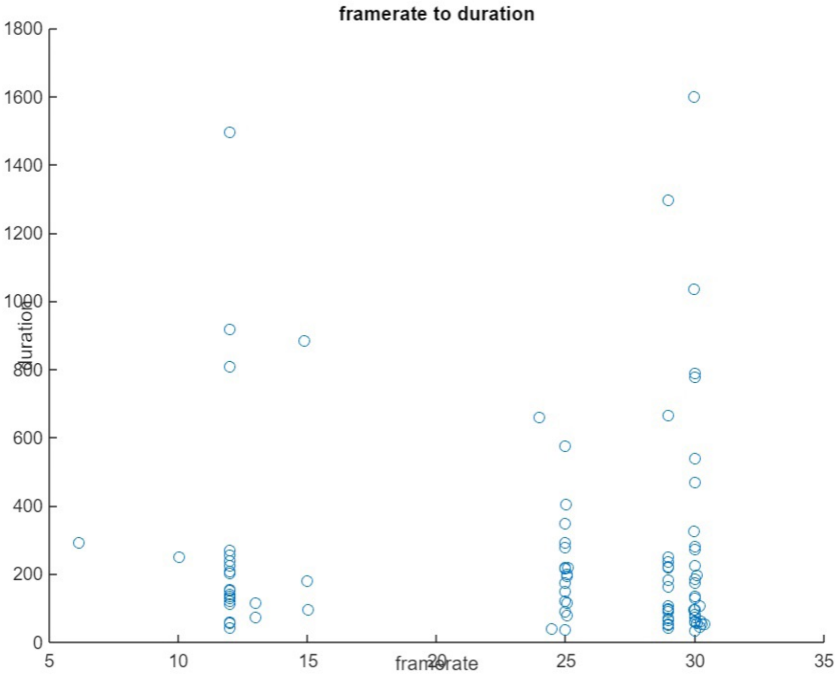
Процент, который составляет модуль разности коэффициентов Пирсона и Спирмена от коэффициента Пирсона:

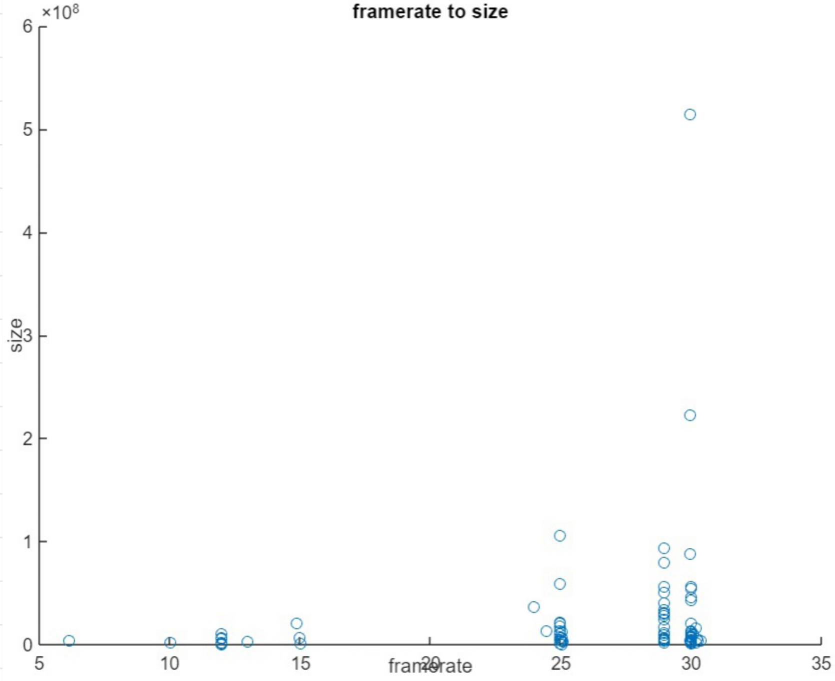
	framerate	frames	bitrate	duration	size
framerate	0	2.9386	62.6825	215.6028	96.2730
frames	2.9386	0	288.8	2.3956	6.9122
bitrate	62.6825	288.8	0	40.2145	81.8242
duration	215.6028	2.3956	40.2145	0	8.991
size	96.2730	6.9122	81.8242	8.991	0

Теперь построим диаграммы разброса для этих пар:

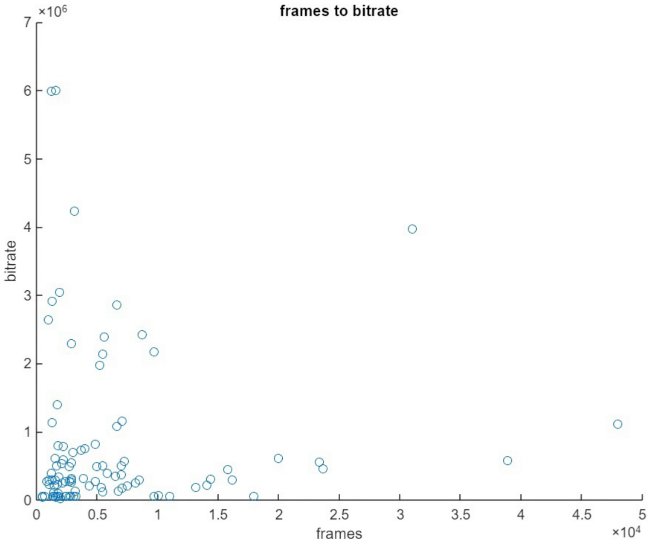


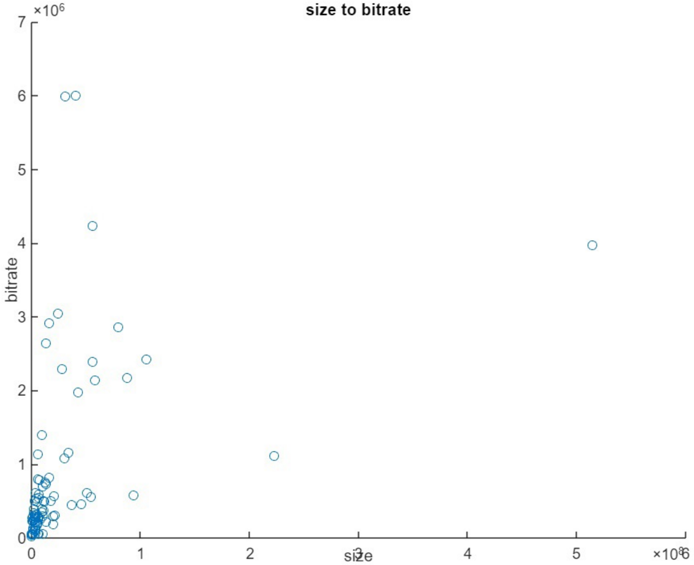
Так как коэффициент Пирсона измеряет монотонную линейную связь, он меньше коэффициента Спирмена (измеряющего монотонную связь) для выборок на графике - как можно видеть, framerate достаточно дискретный, поэтому линейность не получается. В этом случае коэффициент Спирмена должен быть больше (что мы и наблюдаем)





В этих двух случаях, как и в случае выборок framerate и bitrate, дискретность framerate мешает коэффициенту Пирсона увидеть связь между выборками.





На этих графиках мы видим, что точки распределяются по радиальным линиям, и глобально это не линейная связь, поэтому коэффициент Пирсона ее не учитывает, однако коэффициент Спирмена учитывает такую связь тоже, поэтому он в несколько раз больше.

В целом можно сказать, что коэффициент Спирмена благодаря работе с рангами учитывает большее разнообразие нелинейных связей между выборками, поэтому он может существенно отличаться от коэффициента Пирсона для выборок с нелинейной связью.