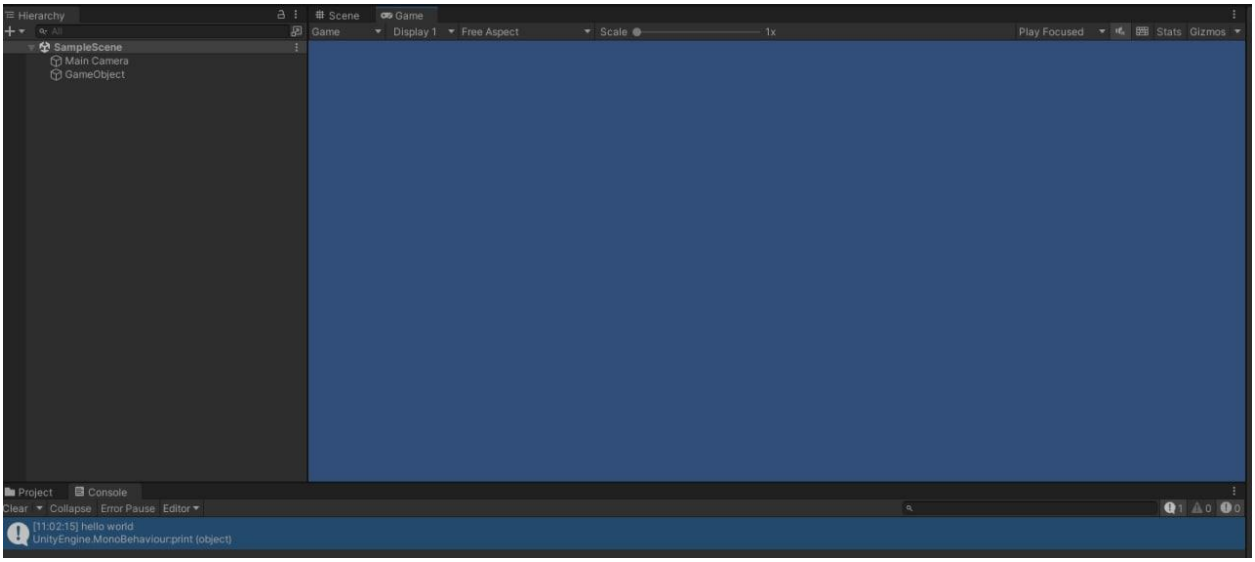
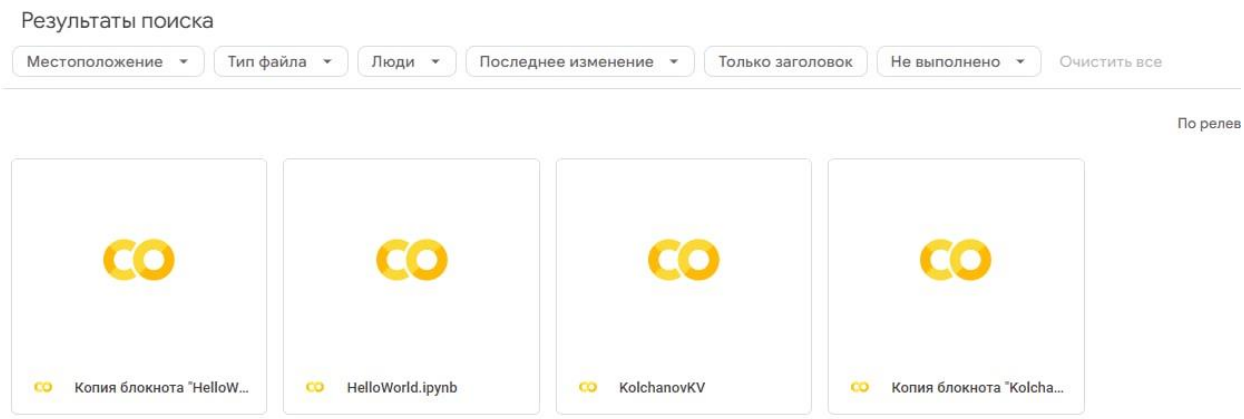
1. АНАЛИЗ ДАННЫХ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ [in GameDev]
2. Написание программ на языке Python и Unity “Hello world”, а также реализация линейной регрессии.
3. Колчанов Константин РИ-110949
4. Задание 1 - Написать программы Hello world на на языке Python и Unity.
   Задание 2 - Пошагово выполнить каждый пункт раздела "ход работы" с описанием и примерами реализации задач
   Задание 3 - Должна ли величина loss стремиться к нулю при изменении исходных данных? Ответьте на вопрос, приведите пример выполнения кода, который подтверждает ваш ответ./ Какова роль параметра Lr? Ответьте на вопрос, приведите пример выполнения кода, который подтверждает ваш ответ. В качестве эксперимента можете изменить значение параметра.

5. Задание 1



https://colab.research.google.com/drive/1Xmw1zRCs0fmEXhhWYQDsCnEjUFl3lzyF?usp=sharing

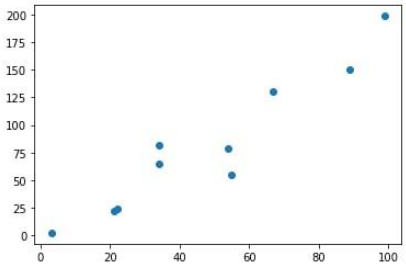


Задание 2

Подготовка данных

Ввод [3]:
```
#Import the required modules, numpy for calculation, and Matplotlib for drawing
import numpy as np
import matplotlib.pyplot as plt
#This code is for jupyter Notebook only
%matplotlib inline

# define data, and change list to array
x = [3,21,22,34,54,34,55,67,89,99]
x = np.array(x)
y = [2,22,24,65,79,82,55,130,150,199]
y = np.array(y)

#Show the effect of a scatter plot
plt.scatter(x,y)
```

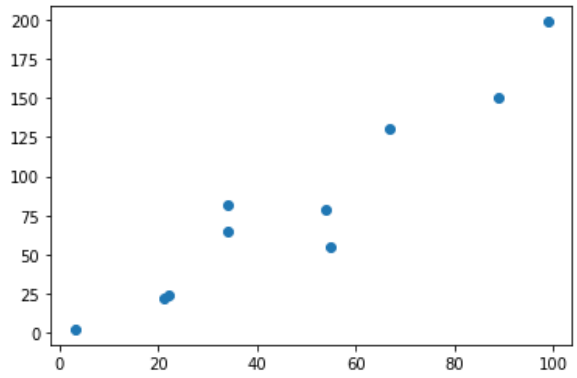
Out[3]:
```
<matplotlib.collections.PathCollection at 0x1d5135a0ee0>
```



Ввод [ ]:

Определение функций

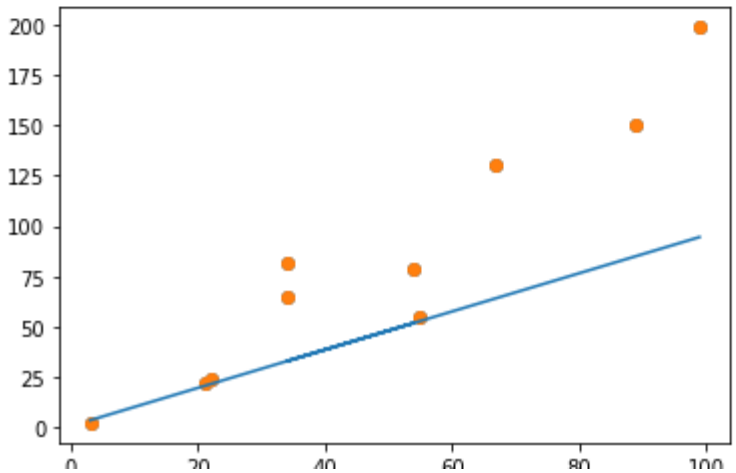
```
    num = len(x)
    prediction = model(a, b, x)
    return (0.5/num)*(np.square(prediction-y)).sum()

def optimize(a, b, x, y):
    num=len(x)
    prediction=model(a, b, x)
    da = (1.0/num)*((prediction-y)*x).sum()
    db = (1.0/num)*((prediction-y).sum())
    a = a - Lr*da
    b = b - Lr*db
    return a, b

def iterate(a,b,x,y,times):
    for i in range(times):
        a,b = optimize(a,b,x,y)
    return a,b

a = np.random.rand(1)
print(a)
b = np.random.rand(1)
print(b)
Lr=0.000001

a,b = iterate(a,b,x,y,1)
prediction = model(a,b,x)
loss = loss_function(a, b, x, y)
print (a, b, loss)
plt.scatter(x, y)
plt.plot(x,prediction)
```



Итерации

```
    num = len(x)
    prediction = model(a, b, x)
    return (0.5/num)*(np.square(prediction-y)).sum()

def optimize(a, b, x, y):
    num=len(x)
    prediction=model(a, b, x)
    da = (1.0/num)*((prediction-y)*x).sum()
    db = (1.0/num)*((prediction-y).sum())
    a = a - Lr*da
    b = b - Lr*db
    return a, b

def iterate(a,b,x,y,times):
    for i in range(times):
        a,b = optimize(a,b,x,y)
    return a,b

a = np.random.rand(1)
print(a)
b = np.random.rand(1)
print(b)
Lr=0.000001

a,b = iterate(a,b,x,y,1)
prediction = model(a,b,x)
loss = loss_function(a, b, x, y)
print (a, b, loss)
plt.scatter(x, y)
plt.plot(x,prediction)
```

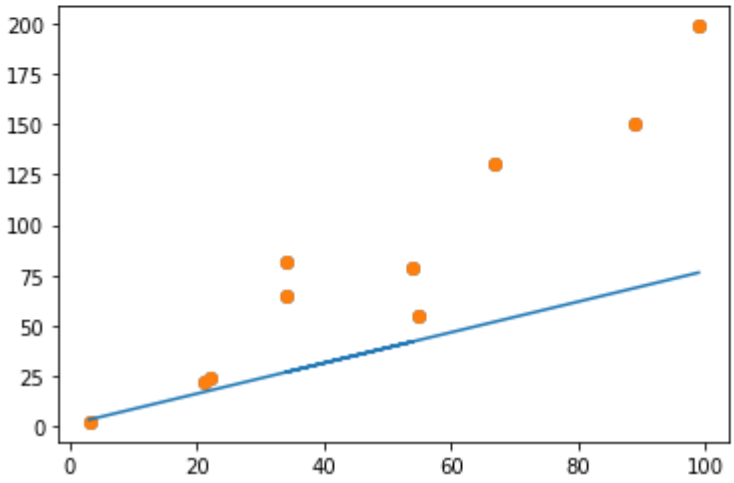
```
[0.94756358]
[0.41963207]
[0.95007296] [0.41966715] 1186.5182716965664

[<matplotlib.lines.Line2D at 0x1d515c187c0>]
```



[1]

```
a = np.random.rand(1)
print(a)
b = np.random.rand(1)
print(b)
Lr=0.000001

a,b = iterate(a,b,x,y,2)
prediction = model(a,b,x)
loss = loss_function(a, b, x, y)
print (a, b, loss)
plt.scatter(x, y)
plt.plot(x,prediction)
```

```
[0.75630192]
[0.76205055]
[0.76247928] [0.76213818] 1696.0435557097837

[<matplotlib.lines.Line2D at 0x1d515b8cb50>]
```

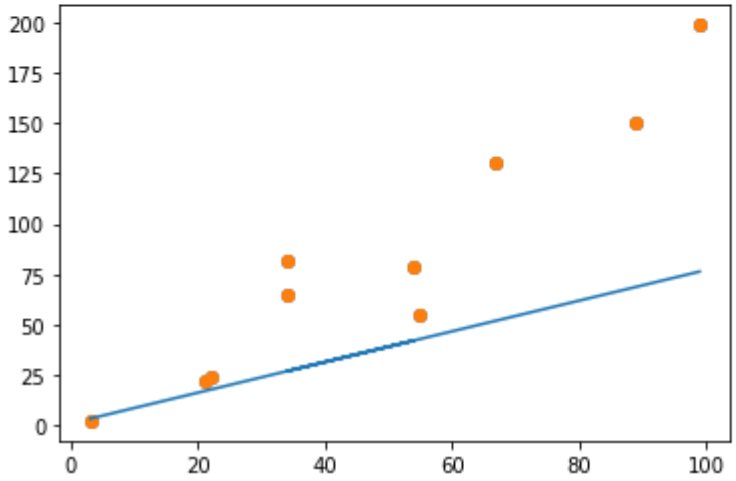


[2]

```
a,b = iterate(a,b,x,y,3)
prediction = model(a,b,x)
loss = loss_function(a, b, x, y)
print (a, b, loss)
plt.scatter(x, y)
plt.plot(x,prediction)
```

```
[0.75630192]
[0.76205055]
[0.76247928] [0.76213818] 1696.0435557097837
```

```
: [<matplotlib.lines.Line2D at 0x1d515b8cb50>]
```

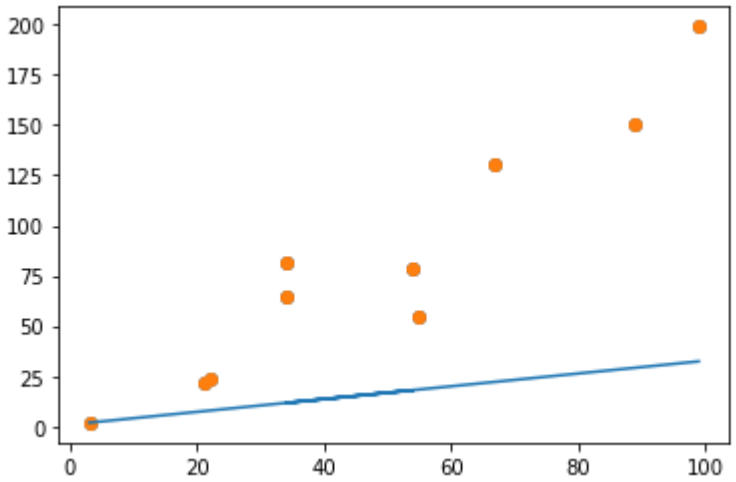


[3]

```
a,b = iterate(a,b,x,y,4)
prediction = model(a,b,x)
loss = loss_function(a, b, x, y)
print (a, b, loss)
plt.scatter(x, y)
plt.plot(x,prediction)
```

```
[0.29960715]
[0.91191609]
[0.31760337] [0.91217706] 3364.6395853894132
```

```
Out[17]: [<matplotlib.lines.Line2D at 0x1d513d9a340>]
```

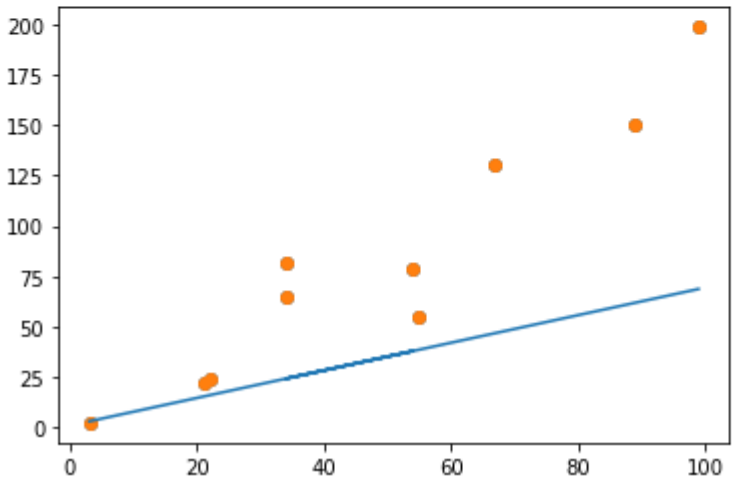


[4]

```
Lr=0.000001

a,b = iterate(a,b,x,y,5)
prediction = model(a,b,x)
loss = loss_function(a, b, x, y)
print (a, b, loss)
plt.scatter(x, y)
plt.plot(x,prediction)
```

```
[0.66941784]
[0.64622884]
[0.68617164] [0.64646801] 1945.23583120364
```

18]: `[<matplotlib.lines.Line2D at 0x1d515b84370>]`

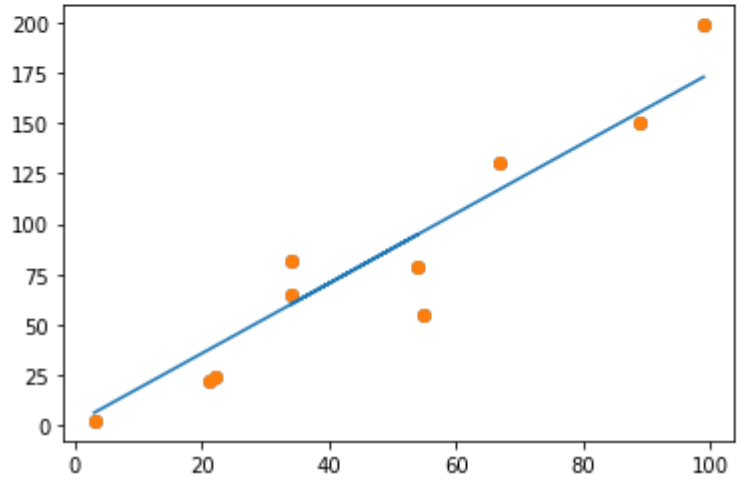


[5]

```
a = np.random.rand(1)
print(a)
b = np.random.rand(1)
print(b)
Lr=0.000001

a,b = iterate(a,b,x,y,10000)
prediction = model(a,b,x)
loss = loss_function(a, b, x, y)
print (a, b, loss)
plt.scatter(x, y)
plt.plot(x,prediction)
```

```
[0.51537661]
[0.80555431]
[1.74109605] [0.79200094] 191.21555933872645
```

```
: [<matplotlib.lines.Line2D at 0x1d515cc7520>]
```



[6]

Задание 3

#

Вывод

Были изучены базовые навыки работы с Unity и Python, а также использованы основные операторы зыка Python на примере реализации линейной регрессии.