1. Goals of the project:

For this final project I gathered information about top five most viewed youtube videos within 24 hours and the related youtube channels to conduct video performance analysis and revenue analysis. As mentioned in homework four, the data extraction and storage codes are saved in the finalproject_data.py file, and are imported to the homework five codes. I have gathered two sets data and saved them in dfvideo and dfuser variables separately. The dfvideo data frame stored information on most viewed videos, including estimated video value evaluated on the www.noxinfluencer.com website, and video view counts, like counts, dislike counts, comment counts from the google youtube videos api. The dfuser saved the data on associated youtube channels that posted these videos, including their ranks, ratings, recent published video counts, and estimated monthly partner earnings, estimated potential earnings per video posted on the website; then also extracted description of the youtube channel and subscriber counts, total video viewing counts, total video counts from youtube channel api. These two data frames both store string data scraped from live source and they share the channel names and subscriber counts accordingly. Based on these data, I want to calculate ratios to show the audience feedback and engagement to these hit videos and further using graphs to present them. Most importantly I want to establish models that can predict the value and potential earnings of each video and channel. I wish to determine the significance of relevant factors using ordinary least squared regression, then also graph out the predicted values against real values.

2. Description of code purpose (flow diagrams)

My homework five codes have six specific purposes. The first part is to import the data extracted from sources, then assign variables to represent two data sets. I also imported different packages that will be used further in this section. The second step is to further edit and conduct analysis on the raw data. This includes transferring data frame format to reset the indexes from video and channel names to default format, so that the data can be used easily. After adding subscriber counts to the dfvideo table, I used different elements in the table to calculate new indicators to show video performance, such as view ratios from video views and subscriber counts, like ratios based on video view counts and like counts, dislike ratios from dislikes and views, engagement ratio based on comment counts and view counts. Then I added new variables with these values for each video in the dfvideo table. I also demonstrated the comparison of the like counts, dislike counts and video view counts of each video in bar graph. In order to improve the usability of the financial data, I also converted the original estimated video value range to average value based on the minimum and maximum video value. The third part of my codes aimed at constructing linear regression on the video data to build a multiple linear model to predict the value of each video. I first set the dependent variable Y to be the estimated average video value and independent variables to be view views, likes, and comment counts. I want to determine the significance of each variable. According to the p value in the results table, these three variables all have rather big p value. I then changed the independent variable to be only the video views, in this linear regression table, the p value is less than 0.05 with certain coefficient of which means the video is worth some amount with each view. I then constructed the fitting line from this model with the scatter plot of the real values in a graph to visualize the correctness of the prediction. The fourth part of my codes is similar to the first part, I edited on the raw youtube channel data. I refined the financial data, including calculating average partner earning from the value range from min and max values, converting earning per video's units million and thousand to one dollar, getting average video views from total video views divided by the total video counts. I also assigned these values to new variables in the table. In the fifth part, I constructed linear regression separately on the estimated monthly partner earnings and estimated average earning per video against different factors including subscriber counts, total video view counts, total video number and average video views. I judge the significance of these elements from p value, smaller the value more important the element is. I then construct the model using the most significant factor for the monthly partner earning, fitting this linear model in a graph with scattered real value plots. On the other hand, the average video view counts factor is significant in the model with certain coefficient for the estimated average earning per video, meaning with every view each video can have fixed amount average earning. I also graphed this linear model with the fitted line and scattered real value plots. In the sixth part of my code. I wan to verify my prediction, I searched for revenue calculation for youtube channels. I then learned that main source of income for youtube channels are advertisement. Though most advertisement income is enclosed, the common knowledge is s that the average YouTube creator could expect to receive from ads $4.18 per 1,000 views, and this factor is called CPM. Thus, this market rule actually proves my model. I extracted this CPM value from the influencermarketinghub.com website to calculate the average video earning per video from the average video views. At last, I also fitted this CPM linear model in a line and graph it with scattered real average earning per video plots from the dfuser table.

First

- Import data extracted from two live sources
- Import useful packages
- Assign variables to imported data sets

Second

- Refine estimated video financial data
- Calculate new indicators and add new variables
- Edit video information and print the table
- Demostrate the audience feedback and engement in graph

Third

- Conduct ordinary least square on average video value and other performance data to determine factor's significance
- Build liner model on most significant factor
- Graph the linear model's fitted line and scattered real value plots

Fourth

- refine estimated channel financial data
- Calculate new indicators and add new variables
- Edit video information and print the table

Fifth

- Conduct ordinary least square on Est. Avg Earning per video, Est. Avg Partner Earning(Monthly) with other performance data to determine factor's significance seperately
- Build liner model on most significant factor
- Graph the linear model's fitted line and scattered real value plots

Sixth

- Extract CPM value from live source
- Build the CPM model to calculate average CPM based earning per video
- Graph the linear model's fitted line and scattered average earning plots

3. Description of your code itself

To further describe these six sections, I will give more details on the codes. In the first part, I imported data from homework four and packages including re, matplotlib, numpy, pandas, sklearn.linear_model, and statsmodels. Then I assigned variable names dfvideo, dfuser to the two data frames imported from homework four. I also included two lists videos and users of the video names and channel names.

In the second part of editing dfvideo raw data, I first reset the index of dfuser from channel name to default indexes and saved a new data frame dfu, so that I can use the column data in the table directly. I extracted the subscriber counts group in a list. I then wrote a for loop to assign subscriber numbers to each video by index because they are listed with the same order in two tables sharing same channel names. I also saved different values in variables values, views, likes, dislikes, comm, subs for each video using loc method to locate estimated video value, video views, likes, dislikes, comments, subscribers counts. Due to the format of the estimated video value, I used re.findall to get the number without the unit k. I also calculated the average value of the video using minimum value and maximum value, then saved these values under Est. Avg Video Value variable. At last I used the views variable divided by subs variable to calculate the view ration, likes variable divided by views variable to calculate the like ratio, dislikes variable divided by views variable to calculate the dislike ratio, comm variable divided by views variable to calculate the engagement ratio. Apart from displaying the video analyzing data, I utilized matplotlib to create graph based on the like counts, dislike counts and views counts to reflect audiences' feedback. I reset the dfvideo data frame as df using default indexes, so that I can directly use data arrays. When creating the graph, I named V1-5 to label top five videos and converted Likes, Dislikes and Video Views data series to integer, also saved them in lists. I then used the arrange method to assign location on X axis to different video labels and created three rectangle bars using the three data lists with fixed width and according label. I then added some text for labels, title and custom x-axis tick labels, also attached a text label above each bar in rectangles, displaying its height. After labeling the three bars, I used tight_layout to set the graph and show the plots. This graph show three bar with different number and color to demonstrate the relationships of the audiences' engagement data.

In the third part, I used the statsmodel package which sets video views, likes, comments variables converted to float values in df as independent variables X. I then set Est. Avg Video Value variable converted to float values in df as dependent variable Y. I used the OLS(ordinary least square) method from the statsmodels.api package to construct a model fitting the X,Y variables. I displayed the linear regression model summary to get p value for each independent variable. According to the summary table, when p value is less than 0.05, that element is significant. In this model, video views variable is relevantly important. I then fit the video views and Est. Avg Video Value to a linear model, the p value is fairly small with a fixed coefficient. I also saved the prediction of this model based on video views. Then I further graphed the fitted line using plots of predicted values and video views, also included the scatter plots of the real Est. Avg Video Value and video views data set. In the fourth part, I did similar editing on the dfuser raw data. First, I converted money unit k(grand) to dollar and getting the average value of Est. Partner Earning(Monthly). I also converted money unit of Est. Potential Earnings per video M and K to dollar. Second, I saved the number of total videos and total video views of each channel to calculate the average view for each video. These improvements can make it easier for data analyzing.

In the fourth part, after resetting the dfuser data frame to default format to dfus, I again used OLS regression to establish models to determine significance based on 'Subscribers','Total Video Views','Total Video','Average video views', 'Est. Avg Partner Earning(Monthly)' and 'Est. Avg Earning per video' variables. I then used the most important independent variable get the linear model separately against 'Est. Avg Partner Earning(Monthly)' and 'Est. Avg Earning per video' variables. Then I graphed the linear model with scattered real values plots separately.

In the last part, I scrapped information from the live website, getting information under certain class to get the market acknowledged revenue estimating rule. I split the paragraph extracted online and get the fixed CPM. I then use CPM as the coefficient in the linear model to predict the Est. video Earning CPM based value. I assigned x variable to 'Average video views' value and y variable to 'Est. video Earning CPM based' value, and Z variable to 'Est. Avg Earning per video'. I graphed the (x,y) variabels in a fitted line and scattered (x,z) plots too.

4. Description of any additional packages

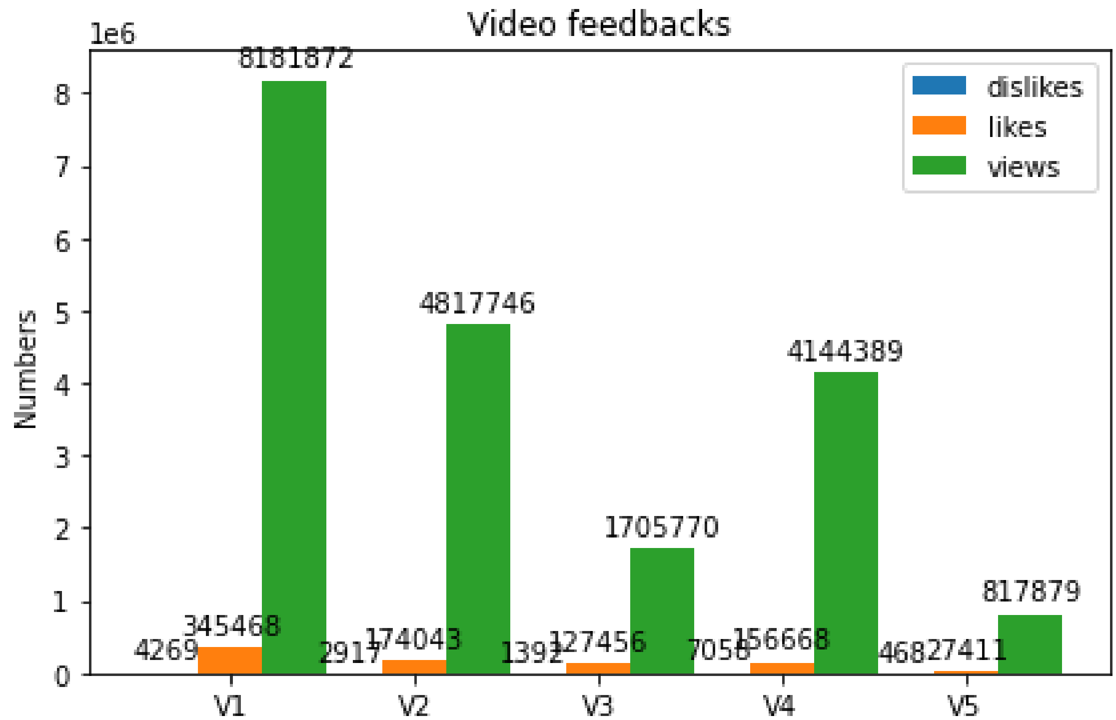
Statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. I imported this package to conduct the OLS regression and get the results summary afterwards.

5. Summary/Presentation of results as of May 12th, 2020

1) Dfvideo data frame after editing after converting units and adding new variables

	Est. Video Value	Video Views	Likes	Dislikes	Comments	Subscribers	Est. Avg Video Value	View Ratio	Like Ratio	Dislike Ratio	Engagement Ratio
The Office Cast Reunites for Zoom Wedding: Some Good News with John Krasinski Ep. 7	124.71K – 144.66K	8181872	345468	4269	16247	2440000	134685	3.353226	0.042224	0.000522	0.001986
Meet Loba – Apex Legends Character Trailer	75.11K – 87.13K	4817746	174043	2917	20754	1430000	81120	3.369053	0.036125	0.000605	0.004308
Our Fertility Journey: Episode 4	29.47K – 34.18K	1705770	127456	1392	16856	542000	31825	3.147177	0.074721	0.000816	0.009882
7 Insane Life Hacks + Funny TikTok Pranks!! How To Make The Best New Candy Art & Ball Pit Challenge	68.94K – 79.98K	4144389	156668	7058	10109	21300000	74460	0.194572	0.037802	0.001703	0.002439
It's Time to go BACK TO THE FUTURE! | Reunited Apart with Josh Gad	7.24K – 8.4K	817879	27411	468	3347	64400	7820	12.699984	0.033515	0.000572	0.004092

2) Number of dislikes, likes and views for each video



3) OLS regression summary table on dependent variabel of Est. Avg Video Value, and independent variabels Likes, Video Views, Comments; Video Views for each video has the smallest p value, meaning it's the most significant variable

OLS Regression Results

Dep. Variable:	Est. Avg Video Value	**R-squared (uncentered):**	0.998
Model:	OLS	**Adj. R-squared (uncentered):**	0.995
Method:	Least Squares	**F-statistic:**	308.0
Date:	Tue, 12 May 2020	**Prob (F-statistic):**	0.00324
Time:	20:15:30	**Log-Likelihood:**	-48.147
No. Observations:	5	**AIC:**	102.3
Df Residuals:	2	**BIC:**	101.1
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Video Views	0.0159	0.004	4.157	0.053	-0.001	0.032
Likes	0.0114	0.102	0.112	0.921	-0.426	0.449
Comments	0.1589	0.419	0.380	0.741	-1.642	1.960

Omnibus:	nan	**Durbin-Watson:**	2.234
Prob(Omnibus):	nan	**Jarque-Bera (JB):**	0.112
Skew:	0.109	**Prob(JB):**	0.945
Kurtosis:	2.299	**Cond. No.**	764.

4) Establish a linear model using 'video views' as independent variable and 'Est. Avg Video Value' as dependent variable, y=0.0168x. The coefficient is 0.0168, meaning for every 1000 video views for each video, the video increases $16.8 value.

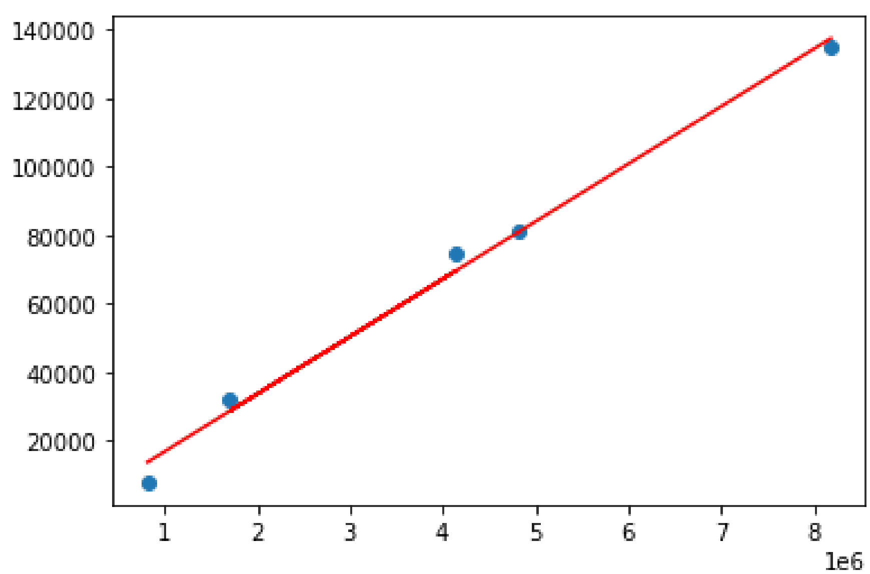
OLS Regression Results

Dep. Variable:	Est. Avg Video Value	**R-squared (uncentered):**	0.998
Model:	OLS	**Adj. R-squared (uncentered):**	0.997
Method:	Least Squares	**F-statistic:**	1642.
Date:	Tue, 12 May 2020	**Prob (F-statistic):**	2.22e-06
Time:	20:15:31	**Log-Likelihood:**	-48.442
No. Observations:	5	**AIC:**	98.88
Df Residuals:	4	**BIC:**	98.49
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Video Views	0.0168	0.000	40.523	0.000	0.016	0.018

Omnibus:	nan	**Durbin-Watson:**	1.793
Prob(Omnibus):	nan	**Jarque-Bera (JB):**	0.412
Skew:	-0.215	**Prob(JB):**	0.814
Kurtosis:	1.661	**Cond. No.**	1.00

5) The fitted prediction line in 4) model with scattered plots of real values



6) Dfuser data frame after editing raw data including converting units and adding nre variables

	Rank	Rating	Published Videos	Est. Partner Earning(Monthly)	Est. Potential Earnings	Description	Subscribers	Total Video Views	Total Videos	Est. Avg Partner Earning(Monthly)	Average video views	Est. Avg Earning per video
iodNews	1,808th (Top 1%)	4.72	6 (Recent Month)	$52.82K-$ 163.49K	$ 77.46K (Each Video)	We would love to hear Some Good News and share...	2440000	62989685	13	108155	4.845360e+06	77460
(Legends	3,389th (Top 1%)	4.43	5 (Recent Month)	$45.64K-$ 141.27K	$ 59.48K (Each Video)	Welcome to the official Apex Legends™ YouTube ...	1430000	154094957	68	93455	2.266102e+06	59480
ie Perkins	10,456th (Top 1.29%)	3.74	4 (Recent Month)	$4K-$ 12.37K	$ 5.49K (Each Video)		542000	20032544	51	8185	3.927950e+05	5490
ollins Key	55th (Top 1%)	4.59	2 (Recent Month)	$174.05K-$ 538.72K	$ 494.08K (Each Video)	New videos every Saturday at 11am PST! \n\nCol...	21300000	4763387658	242	356385	1.968342e+07	494080
Josh Gad	57,740th (Top 7.13%)	3.6	6 (Recent Month)	814– 2.52K	$ 1.46K (Each Video)	Josh Gad is an American actor, comedian, and s...	64400	3380656	11	1667	3.073324e+05	1460

7) OLS regression summary table on dependent variabel of 'Est. Partner Earning(Monthly)', independent variables 'Subscribers','Total Video Views' ,'Average video views'; Average Video Views for each video has the smallest p value, meaning it’s the most significant variable

OLS Regression Results

Dep. Variable:	Est. Avg Partner Earning(Monthly)	**R-squared (uncentered):**	0.988
Model:	OLS	**Adj. R-squared (uncentered):**	0.969
Method:	Least Squares	**F-statistic:**	52.74
Date:	Tue, 12 May 2020	**Prob (F-statistic):**	0.0187
Time:	20:49:45	**Log-Likelihood:**	-56.405
No. Observations:	5	**AIC:**	118.8
Df Residuals:	2	**BIC:**	117.6
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Subscribers	-0.0087	0.095	-0.091	0.935	-0.416	0.399
Total Video Views	-8.752e-06	0.000	-0.036	0.974	-0.001	0.001
Average video views	0.0297	0.045	0.657	0.579	-0.165	0.224

Omnibus:	nan	**Durbin-Watson:**	2.404
Prob(Omnibus):	nan	**Jarque-Bera (JB):**	1.211
Skew:	1.204	**Prob(JB):**	0.546
Kurtosis:	2.901	**Cond. No.**	1.65e+04

8) Establish a linear model using ‘Average video views’ as independent variable and ‘Est. Avg Partner Earning(Monthly)’ as dependent variable, y=0.0186x. The coefficient is 0.0186, meaning for every 1000 video views for each video, the channel earns $18.6 value.

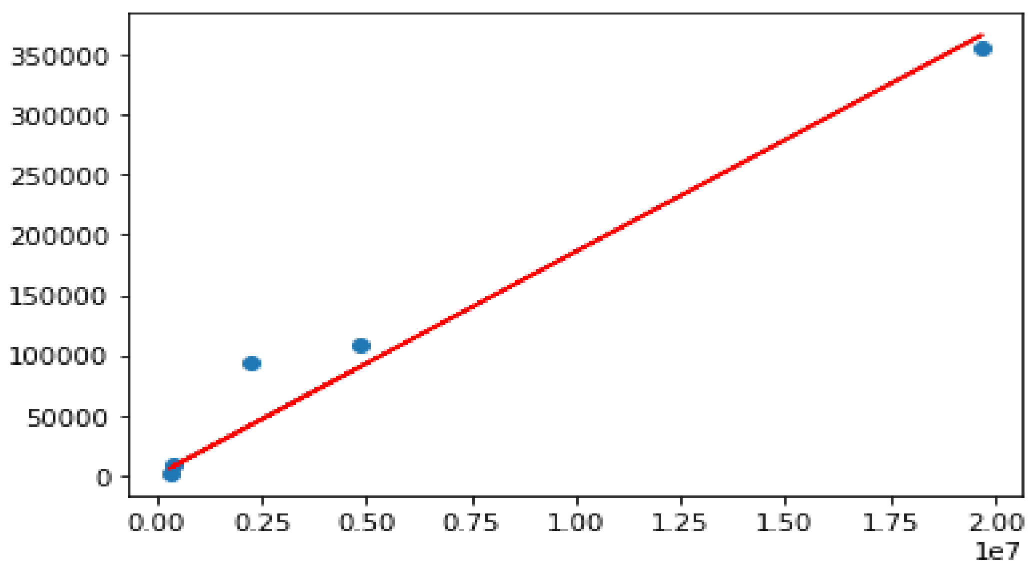
OLS Regression Results

Dep. Variable:	Est. Avg Partner Earning(Monthly)	**R-squared (uncentered):**	0.979
Model:	OLS	**Adj. R-squared (uncentered):**	0.974
Method:	Least Squares	**F-statistic:**	188.3
Date:	Tue, 12 May 2020	**Prob (F-statistic):**	0.000163
Time:	20:15:36	**Log-Likelihood:**	-57.683
No. Observations:	5	**AIC:**	117.4
Df Residuals:	4	**BIC:**	117.0
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Average video views	0.0186	0.001	13.721	0.000	0.015	0.022

Omnibus:	nan	**Durbin-Watson:**	1.242
Prob(Omnibus):	nan	**Jarque-Bera (JB):**	0.800
Skew:	0.931	**Prob(JB):**	0.670
Kurtosis:	2.388	**Cond. No.**	1.00

9) The fitted prediction line in 8) model with scattered plots of real values



10) OLS regression summary table on dependent variabel of 'Est. Avg Earning per video', independent variables 'Subscribers','Average video views','Total Videos','Total Video Views'; 'Average Video Views' for each video has the smallest p value, meaning it's the most significant variable

OLS Regression Results

Dep. Variable:	Est. Avg Earning per video	R-squared (uncentered):	1.000
Model:	OLS	Adj. R-squared (uncentered):	0.998
Method:	Least Squares	F-statistic:	628.7
Date:	Tue, 12 May 2020	Prob (F-statistic):	0.0299
Time:	20:15:37	Log-Likelihood:	-49.144
No. Observations:	5	AIC:	106.3
Df Residuals:	1	BIC:	104.7
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Subscribers	-0.0355	0.037	-0.949	0.517	-0.511	0.440
Average video views	0.0318	0.017	1.849	0.316	-0.187	0.250
Total Videos	238.6045	165.468	1.442	0.386	-1863.859	2341.068
Total Video Views	0.0001	9.27e-05	1.283	0.421	-0.001	0.001

Omnibus:	nan	Durbin-Watson:	1.371
Prob(Omnibus):	nan	Jarque-Bera (JB):	0.437
Skew:	-0.680	Prob(JB):	0.804
Kurtosis:	2.498	Cond. No.	7.85e+07

11) Establish a linear model using 'Average video views' as independent variable and 'Est. Avg Earning per video' as dependent variable, y=0.0246x. The coefficient is 0.0246, meaning for every 1000 video views for each video, the channel earns $24.6 value.

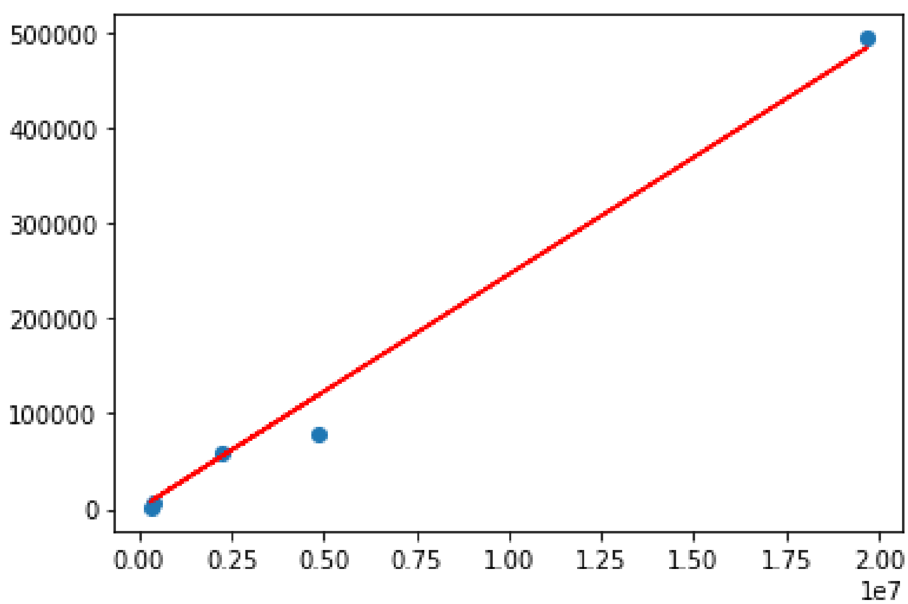
OLS Regression Results

Dep. Variable:	Est. Avg Earning per video	**R-squared (uncentered):**	0.992
Model:	OLS	**Adj. R-squared (uncentered):**	0.991
Method:	Least Squares	**F-statistic:**	527.9
Date:	Tue, 12 May 2020	**Prob (F-statistic):**	2.13e-05
Time:	20:15:38	**Log-Likelihood:**	-56.494
No. Observations:	5	**AIC:**	115.0
Df Residuals:	4	**BIC:**	114.6
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Average video views	0.0246	0.001	22.976	0.000	0.022	0.028

Omnibus:	nan	**Durbin-Watson:**	1.357
Prob(Omnibus):	nan	**Jarque-Bera (JB):**	1.051
Skew:	-1.118	**Prob(JB):**	0.591
Kurtosis:	2.795	**Cond. No.**	1.00

12) The fitted prediction line in 11) model with scattered plots of real values



13) CPM and Est. video earning based on CPM and average video views

Subscribers	Total Video Views	Total Videos	Est. Avg Partner Earning(Monthly)	Average video views	Est. Avg Earning per video	CPM	Est. video Earning CPM based
2440000	62989685	13	108155	4.845360e+06	77460	4.18	20253.606408
1430000	154094957	68	93455	2.266102e+06	59480	4.18	9472.307651
542000	20032544	51	8185	3.927950e+05	5490	4.18	1641.883018
21300000	4763387658	242	356385	1.968342e+07	494080	4.18	82276.695911
64400	3380656	11	1667	3.073324e+05	1460	4.18	1284.649280

14) The fitted prediction line for 'Est. video earning CPM based' in using CPM model with scattered plots of 'Average video views' and 'Est. Avg Earning per video' values