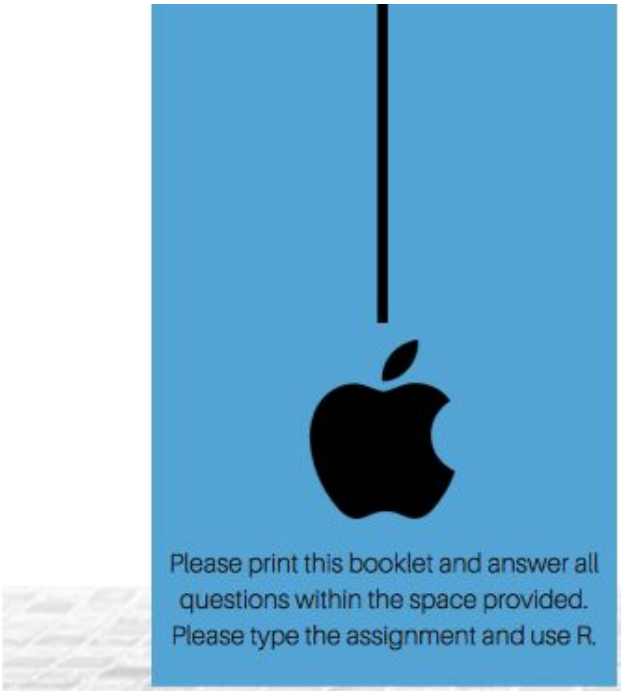
Statistical Foundations of Data Analytics

LAB 4: Non-linear models



YOUR TEAM

Last name: Toronga
First name: Nicholas

Last name: Tian
First name: Yuan

Last name: Greenes
First name: Sam

Lab 4: How to predict an app's rating in the Apple Store? A non-linear approach

The online app market is one of the fiercest. Out of thousands of apps, only a few survive. And the likelihood of making a profitable app is tiny. In order to boost the app's popularity and success, it is necessary to have good ratings.

To understand what drives an app's ratings, I have obtained data from 7200 apps from the apple store. This data includes the app's characteristics. We previously used a linear approach to build a model that helps us predict the app rating in the apple store. In today's lab, we will see how to improve our data modelling skills using non-linear models.

Response Variable

User_rating: The average rating of the app. This variable ranges from one to five.

0= terrible app; 5= fantastic app

Predictors

Continuous predictors

- **size_mb:** Size (in megabytes)
- **price:** Price amount (in USD)
- **rating_count_tot:** Number of users that have rated the app
- **rating_count_ver:** Number of users that have rated this version's app
- **ver:** Latest version
- **sup_devices.num:** Number of apple devices that support the app
- **screenshots_ipad:** Number of screenshots showed for display in the app store (ipad devices)
- **screenshots_iphone:** Number of screenshots showed for display in the app store (iphone devices)
- **lang.num:** Number of supported languages

1. Linear regression (5 points)

Run the following three linear regressions, where:

- $User\_rating = b_0 + b_1(rating\_count\_tot)$
- $User\_rating = b_0 + b_1(sup\_devices.num) + b_2(rating\_count\_tot)$

A) Paste the regression outputs below for the three regressions. I want you to export the regression output in one nice HTML table with three columns (one per each regression), using the stargazer package:[1]

Regression Output

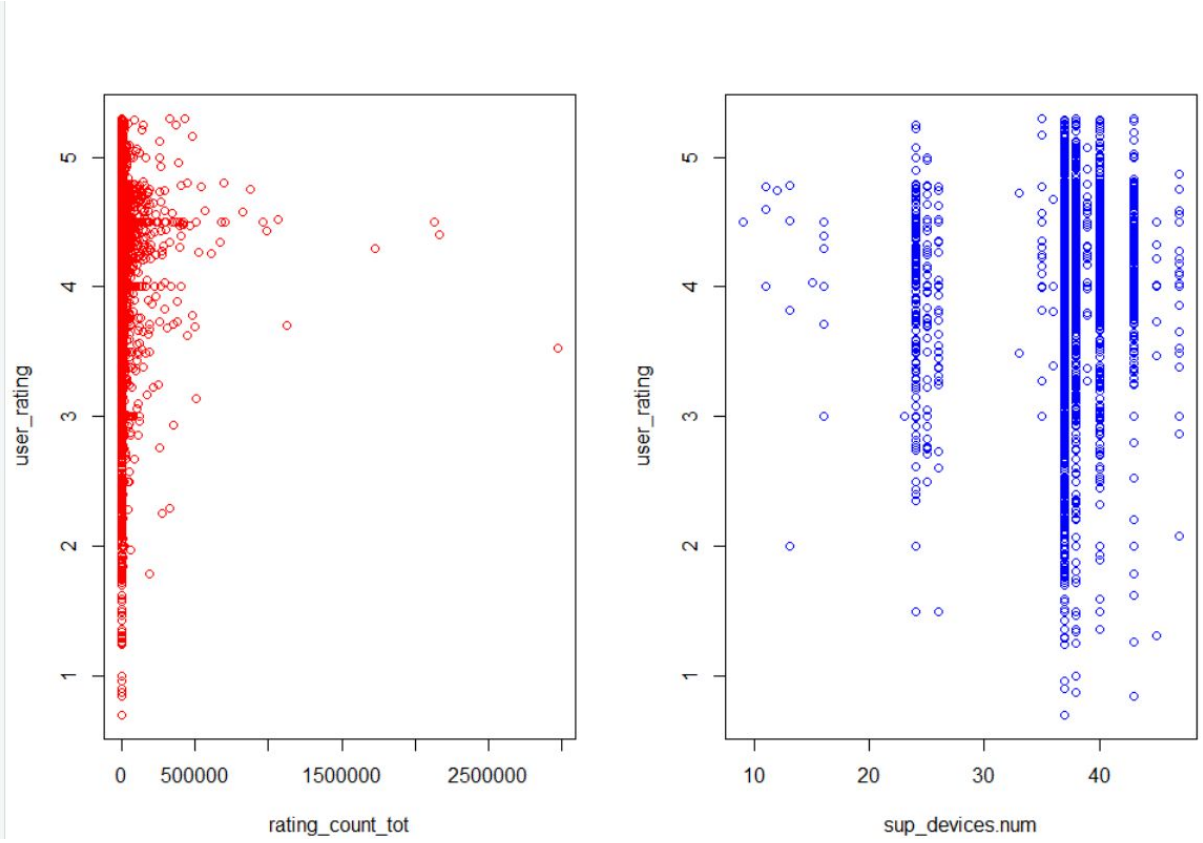
	Dependent variable:		
	user_rating		
	(1)	(2)	(3)
sup_devices.num	0.011***		0.011***
	(0.002)		(0.002)
rating_count_tot		0.00000***	0.00000***
		(0.00000)	(0.00000)
Constant	3.693***	4.092***	3.691***
	(0.087)	(0.009)	(0.087)
Observations	5,130	5,130	5,130
R^2	0.004	0.003	0.007
Adjusted R^2	0.004	0.003	0.007
Residual Std. Error	0.652 (df = 5128)	0.652 (df = 5128)	0.651 (df = 5127)
F Statistic	22.165*** (df = 1; 5128)	15.048*** (df = 1; 5128)	18.394*** (df = 2; 5127)
Note:			*p<0.1; **p<0.05; ***p<0.01

[1] Stargazer will make your reports more appealing --- as opposed simply pasting the output. Here's a link on how to export tables to HTML using stargazer. But you can find lots of documentation about this package in *r*. From now on, you should be getting used to exporting output with stargazer.

B) Based on these regression results, which type of applications receives higher ratings? Summarize your findings. Comment on the significance of the predictors and R-square of the regressions. Avoid jargon.

Based on these regression results applications that support a great number of devices receive higher ratings as that for each additional supported device the average user ratings increases by ~0.011 points.The number of ratings does not seem to be influential as that the coefficient is very close to zero, meaning that an increase in number of ratings is not associated with any significant increase in average star user ratings based on our model.

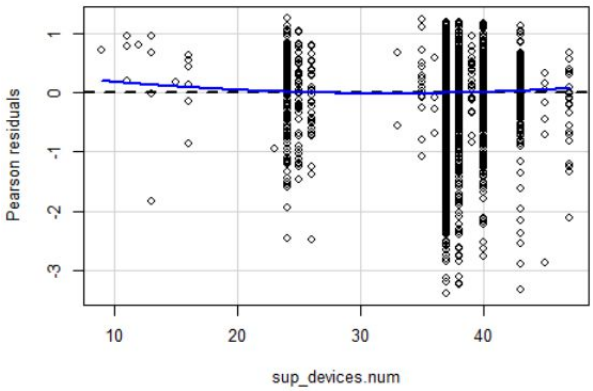
C) Scatter Plots: Create a matrix of two scatter plots and paste them below (one row, two columns). In the first scatter plot, you should include the predicted variable and rating_count_tot; in the second scatter plot, you should include the predicted variable and sup_devices.num.

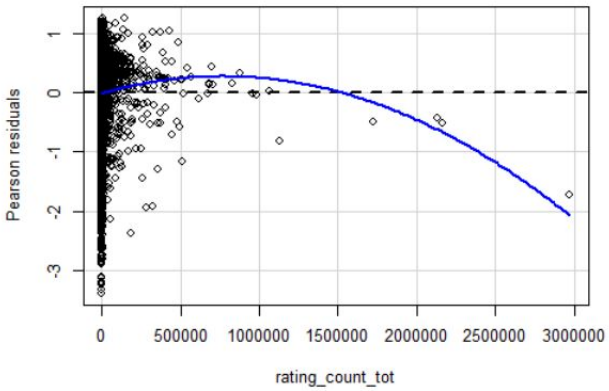


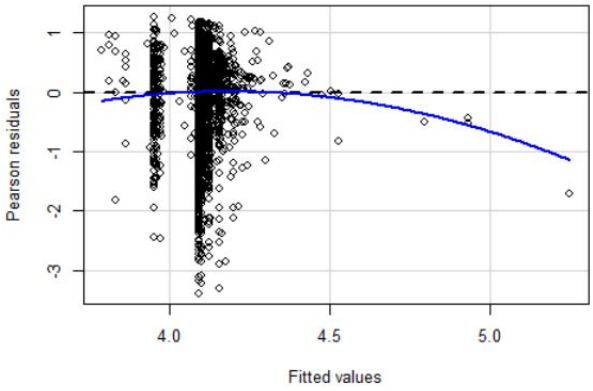
D) Based on the scatter plots above, are there apparent non-linearities in the data? Give me your honest opinion (visual clues only).

g.From the scatter plots above there seems to be apparent non-linearities.Indeed for both variables ,the data does not seem to fit a line : the rating_count_tot looks clustered around 0 ratings whereas the sup_devices have values that do not correspond to any user_rating(ex: 30)

E) Residual plot: Run a residual plot analysis of Model #3 (i.e., the model with the two predictors) using the residualPlots() function from the "car" package. Please paste the residual plots, and numerical results below:







```
> residualPlots(lm_rating1)
               Test stat Pr(>|Test stat|)
sup_devices.num    1.3618           0.1733
Tukey test         1.3618           0.1733
> residualPlots(lm_rating2)
                Test stat Pr(>|Test stat|)
rating_count_tot   -4.8937        1.020e-06 ***
Tukey test         -4.8937        9.898e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> residualPlots(lm_rating3)
                Test stat Pr(>|Test stat|)
sup_devices.num     1.4375           0.1506
rating_count_tot   -4.8477        1.286e-06 ***
Tukey test         -4.3751        1.214e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

F) What do the results above tell you? Discuss both visual evidence and numerical results.

From the visuals results we could already see that the linearity assumption is violated in model 2 and 3 as that we notice a slightly curved pattern but not in the 1st model.This observation is later confirmed as that the linearity test results show that model 1 with the variable ‘ sup_devices’ is not subject to non linearity as hat its p-value is > 0.05 in both model 1 and 3. On the other the p value for the ‘rating_count_tot” for the 2 other models is very close to 0 hence smaller than 0.05 making the it subject to non linearity.The ‘rating_cotunt_tot’ variable seems to create issues to the model as that adding it to ‘ sup_devices’ makes the regression fail the linear test.

2. Polynomial regression: number of supported devices (5 points)

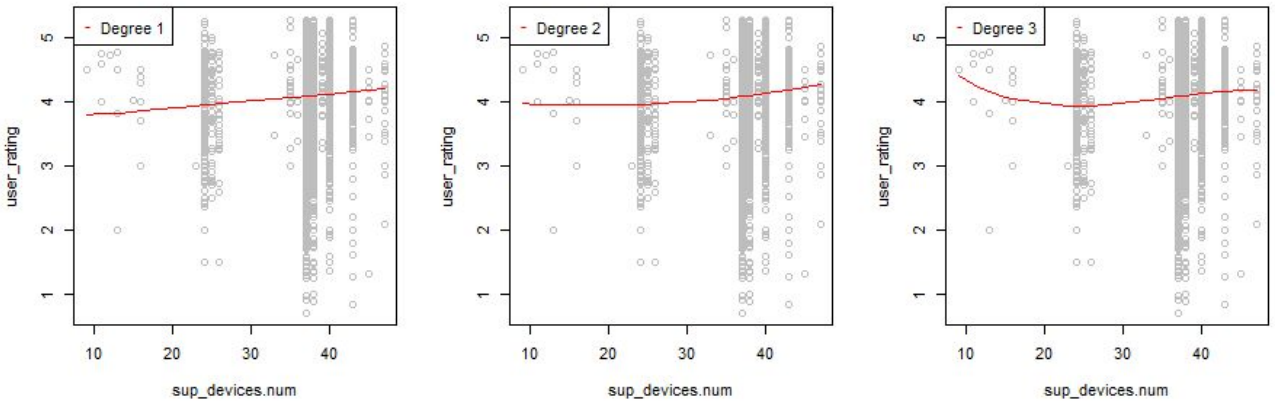
A) Let's try to figure out the true functional relationship between *user_rating* and *sup_devices.num*:

User_rating=f(sup_devices.num)

Try running three regressions: a linear (d=1), quadratic (d=2) and cubic (d=3) one. Paste the output of these three regressions below (using stargazer)

```
> stargazer(reg_poly_00, reg_poly_01, reg_poly_02, title="Results", align=TRUE, type = "text")

Results
================================================================================================
                                          Dependent variable:
                      ---------------------------------------------------------------------
                                                user_rating
                             (1)                    (2)                    (3)
------------------------------------------------------------------------------------------------
sup_devices.num            0.011***
                           (0.002)

poly(sup_devices.num, 2)1                        3.068***
                                                 (0.652)

poly(sup_devices.num, 2)2                         0.887
                                                 (0.652)

poly(sup_devices.num, 3)1                                               3.068***
                                                                        (0.651)

poly(sup_devices.num, 3)2                                                0.887
                                                                        (0.651)

poly(sup_devices.num, 3)3                                               -1.254*
                                                                        (0.651)

Constant                   3.693***               4.099***               4.099***
                           (0.087)                (0.009)                (0.009)

------------------------------------------------------------------------------------------------
Observations                 5,130                  5,130                  5,130
R2                           0.004                  0.005                  0.005
Adjusted R2                  0.004                  0.004                  0.005
Residual Std. Error    0.652 (df = 5128)      0.652 (df = 5127)      0.651 (df = 5126)
F Statistic        22.165*** (df = 1; 5128) 12.012*** (df = 2; 5127) 9.247*** (df = 3; 5126)
================================================================================================
Note:                                                        *p<0.1; **p<0.05; ***p<0.01
>
```

B) Create a matrix of three scatterplots (1 row, 3 columns) with the corresponding polynomial fit in each graph. Make sure that the dots of the scatter plot are in grey, the fitted polynomial is in red, and that there is a legend in the top-left corner of each graph (indicating the degree). Paste the matrix of graphs below:



C) Run an ANOVA test to determine which is the optimal polynomial model (between the linear, quadratic, and cubic regressions). Post the results below:

```
> anova(reg_poly_00, reg_poly_01, reg_poly_02)
Analysis of Variance Table

Model 1: user_rating ~ sup_devices.num
Model 2: user_rating ~ poly(sup_devices.num, 2)
Model 3: user_rating ~ poly(sup_devices.num, 3)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1   5128 2177.8
2   5127 2177.0  1   0.78741 1.8554 0.17322
3   5126 2175.4  1   1.57282 3.7060 0.05427 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

D) Based on the above results, which model would you use to describe this functional relationship (check one)?

a) Linear
~~b) Quadratic~~
~~c) Cubic~~

E) Based on the model chosen above, what did you learn about the relationship between the user_rating and the number of supported devices?

Your answer:

The p-value for increasing d from 1 to 2 does not justify that we use the quadratic model. It is way above 5% which is the threshold. However, it is interesting to note that even though we are not taking d =2, as d increases from 2 to 3, the p-value is almost proving that the model gets better.

3. Polynomial regression: Number of ratings (5 points)

A) Let's try to figure out the true functional relationship between *user_rating* and *rating_count_tot*:

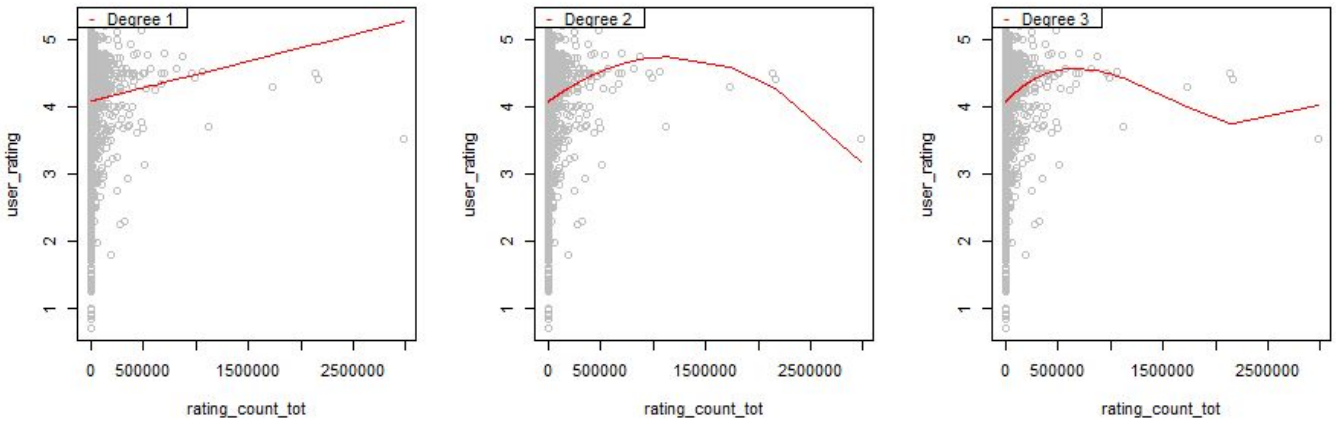
User_rating=f(rating_count_tot)

Try running three regressions: linear (d=1), quadratic (d=2) and cubic (d=3) regression. Paste the output of these three regressions below (row 1, column 3)

Regression Results

	Dependent variable:		
	user_rating		
	(1)	(2)	(3)
rating_count_tot	0.00000***		
	(0.00000)		
poly(rating_count_tot, 2)1		2.530***	
		(0.651)	
poly(rating_count_tot, 2)2		-3.184***	
		(0.651)	
poly(rating_count_tot, 3)1			2.530***
			(0.650)
poly(rating_count_tot, 3)2			-3.184***
			(0.650)
poly(rating_count_tot, 3)3			1.620**
			(0.650)
Constant	4.092***	4.099***	4.099***
	(0.009)	(0.009)	(0.009)
Observations	5,130	5,130	5,130
R2	0.003	0.008	0.009
Adjusted R2	0.003	0.007	0.008
Residual Std. Error	0.652 (df = 5128)	0.651 (df = 5127)	0.650 (df = 5126)
F Statistic	15.048*** (df = 1; 5128)	19.532*** (df = 2; 5127)	15.102*** (df = 3; 5126)
Note:			*p<0.1; **p<0.05; ***p<0.01

B) Create a matrix of three scatterplots with the corresponding polynomial fit in each graph. Make sure that the dots of the scatter plot are in grey, the fitted polynomial is in red, and that there is a legend in the top-left corner of each graph (indicating the degree). Paste the matrix of graphs below:



C) Run an ANOVA test to determine which is the optimal polynomial model (between the linear, quadratic, and cubic regressions). Post the results below:

Analysis of Variance Table

Model 1: user_rating ~ rating_count_tot
Model 2: user_rating ~ poly(rating_count_tot, 2)
Model 3: user_rating ~ poly(rating_count_tot, 3)

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)

D) Based on the above results, which model would you use to describe this functional relationship (check one)?

a) Linear_____
b) Quadratic _____
c) Cubic___x___

E) Based on the model chosen above, what did you learn about the relationship between the user_rating and the number of ratings?

It has been determined, according to the ANOVA test performed above, that the probability that going from a regression of degree 1 to a polynomial regression of degree 2 was not statistically significant was a mere 0.0001%. Similarly, it was determined that the probability that going from degree 2 to degree 3 was not statistically significant was only 1.278%. Thus, it is determined that the relationship between the user rating of an app and the number of ratings (total) is not linear and that it seemed to follow more a cubic pattern. In fact, a variable transformation to the third (cubic) degree contributed best in rendering the non-linear model more linear without incurring an overfitting model and/or a big loss in degrees of freedom.

4. Multiple polynomial regression (4 points)

A) Now, let's try to figure out the following relationship:

User_rating=f(sup_devices.num, rating_count_tot)

I want you to run the regression using the best polynomial relationship you found for supp_device.num (in Question #2) and rating_count_tot (in Question #3), and combine them in a multiple polynomial regression. Please write the multiple polynomial regression equation below:

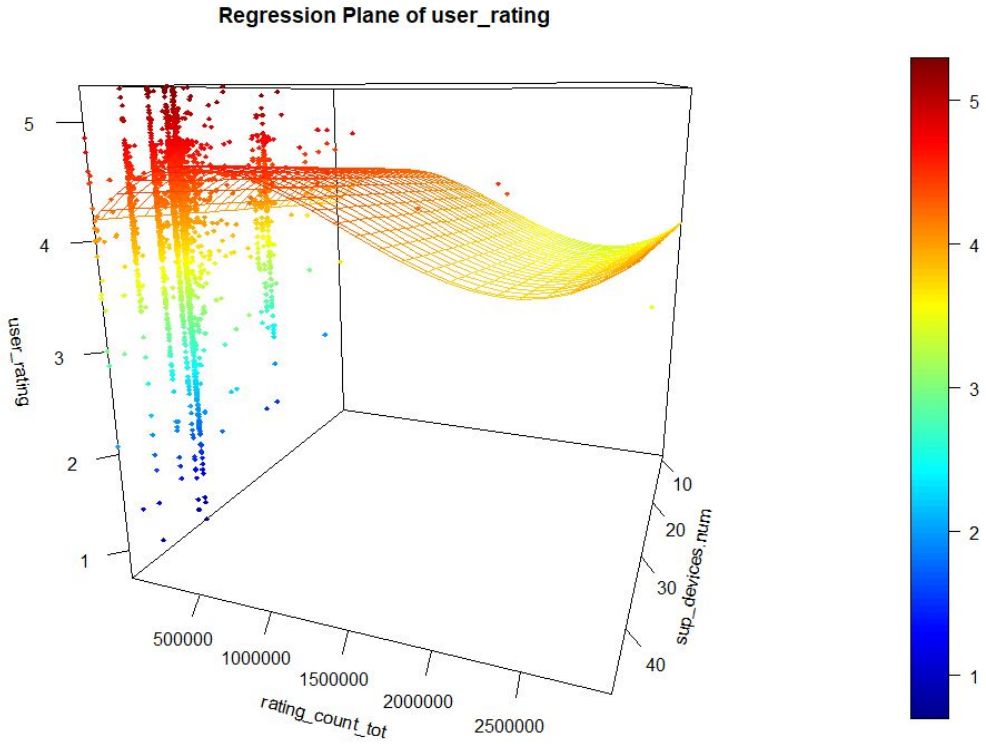
user_rating = 3.702544 + 0.010642 (sup_devices.num) + 2.484545 (rating_count_tot) - 3.148310 (rating_count_tot)^2 +1.622204 (rating_count_tot)^3

B) Paste the output of this regression below:

Regression Result

	Dependent variable:
	user_rating
sup_devices.num	0.011***
	(0.002)
poly(rating_count_tot, 3)1	2.485***
	(0.649)
poly(rating_count_tot, 3)2	-3.148***
	(0.649)
poly(rating_count_tot, 3)3	1.622**
	(0.649)
Constant	3.703***
	(0.086)
Observations	5,130
R2	0.013
Adjusted R2	0.012
Residual Std. Error	0.649 (df = 5125)
F Statistic	16.689*** (df = 4; 5125)
Note:	*p<0.1; **p<0.05; ***p<0.01

C) Create a 3D graph with the fitted regression 3D plane, and the scatter plot. You will need to figure out how—there are lots of packages online. Hint: do not forget to save your script before this question as your session might abort if you experiment with packages.[2]



D) In this new regression, what happened to the r-squared, compared to the multiple linear regression from question 1?

The r-squared of this polynomial regression increased to 0.013 compared to the r-squared of 0.007 observed in the multiple regression from question 1. Thus, we can

[2] If the range is too large and the plot keeps crashing, you may plot a restricted range.

effectively conclude that using the exact same variables and adding the non-linear terms to the regression had a positive impact on the predictive power of this model.

5. Spline regression (6 points)

A) Briefly, what is the difference between a polynomial regression and a spline regression? Why would we want to use splines as opposed to polynomials?

Your answer: Polynomial regression models force the regression function into a certain "standard" shape like parabola, cubic function, etc. It ignores the fact that data may have a very different shape that cannot be captured by the standard polynomial "rules". Spline regression models are useful in capturing unique patterns and exposing the shape of the data without being biased on the bases of polynomial theorems and standard shapes.

B)

Let's focus on the following relationship

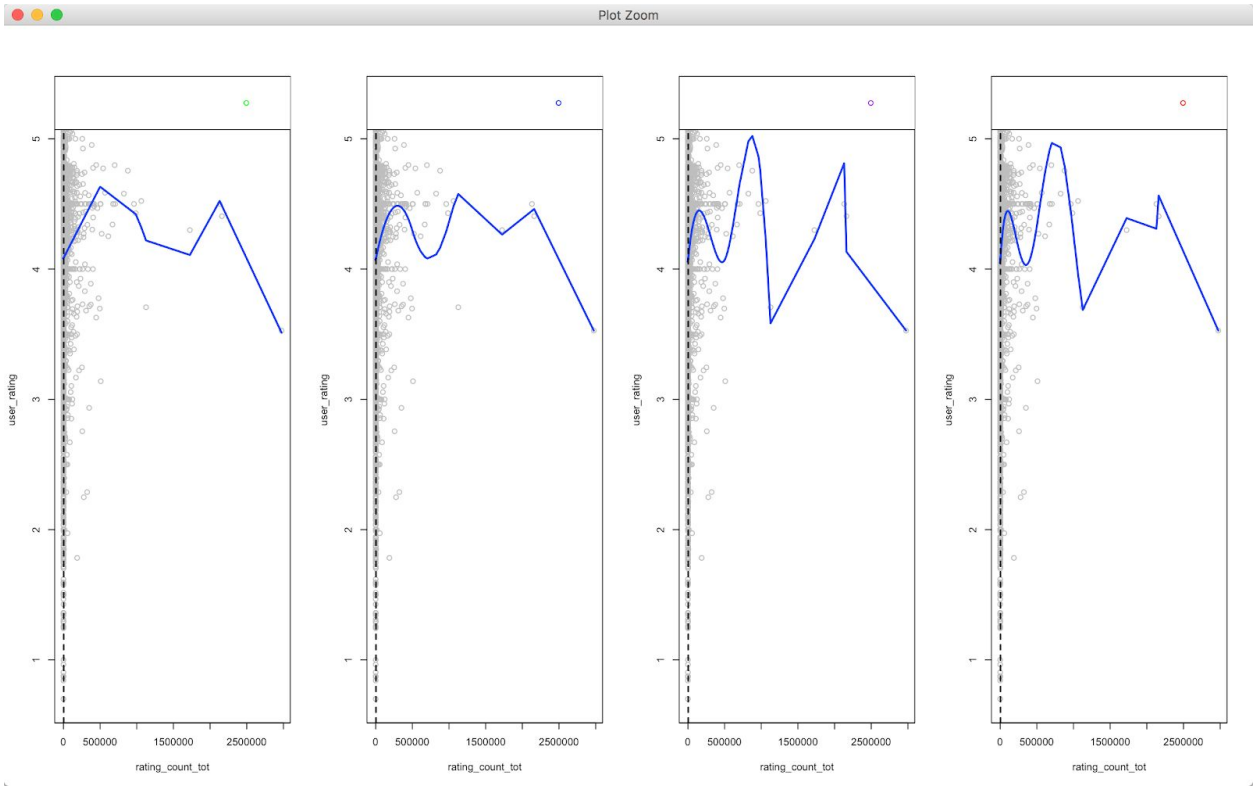
```
User_rating=f(rating_count_tot)
```

For the above relationship, I want you to run four spline regressions with four knots each: (i) a linear spline; (ii) a quadratic spline; (iii) a cubic spline; and (iv) a quartic spline. I want you to space the knots uniformly across the data, using the method we learned in class. Please paste the four regressions outputs below (again, using the stargazer package):

```
> stargazer(splin_reg_d1, splin_reg_d2, splin_reg_d3, splin_reg_d4, title = "Spline Regression Results", type = "text")

Spline Regression Results
==========================================================================================================================
=============================================
                                                                                                              Dependent var
iable:
                                                                                   ---------------------------------------------------
---------------------------------------------

user_rating
                                                                                        (1)                     (2)
(3)                      (4)
--------------------------------------------------------------------------------------------------------------------------
---------------------------------------------
bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 1)1               0.357
                                                                                     (0.611)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 1)2               0.908
                                                                                     (0.619)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 1)3               0.632
                                                                                     (0.666)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 1)4               0.846
                                                                                     (1.037)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 1)5


bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 2)1                                       0.504
                                                                                                             (0.646)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 2)2                                       1.193*
                                                                                                             (0.653)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 2)3                                       0.282
                                                                                                             (0.700)
```

```
-5.898***

(2.057)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 4)6
8.721**

(3.686)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 4)7
-6.506

(5.016)

bs(rating_count_tot, knots = c(0, 5e+05, 1e+06, 1500000), degree = 4)8


Constant                                                                    3.726***                3.568***
3.507***                3.529***
                                                                             (0.611)                (0.646)
(0.648)                 (0.646)

-------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------
Observations                                                                  5,130                   5,130
5,130                   5,130
R2                                                                            0.007                   0.011
0.016                   0.023
Adjusted R2                                                                   0.006                   0.010
0.015                   0.022
Residual Std. Error                                                  0.651 (df = 5125)       0.650 (df = 5124)
0.648 (df = 5123)       0.646 (df = 5122)
F Statistic                                                 9.387*** (df = 4; 5125) 11.740*** (df = 5; 5124) 14.
276*** (df = 6; 5123) 17.151*** (df = 7; 5122)
===============================================================================================================================
===============================================
Note:
*p<0.1; **p<0.05; ***p<0.01
> |
```

C) Create a matrix of four distinct scatterplots. In each graph, put each regression fitted splines, with: (i) dots in gray; (ii) splines in blue; (iii) vertical dashed lines where the knots are located; and (iv) a legend indicating the degree of the spline, and the number of knots.



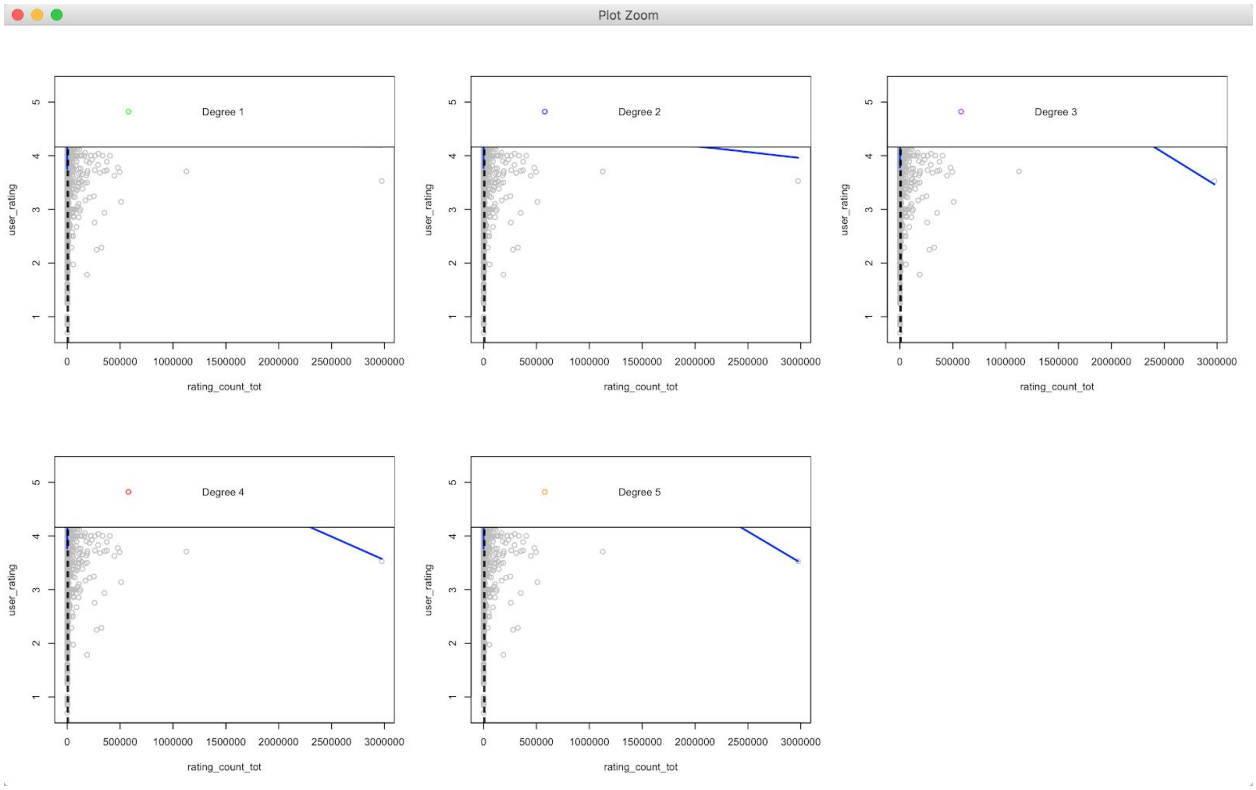
D) Now, run five spline regressions with five knots each: (i) a linear spline; (ii) a quadratic spline; (iii) a cubic spline; (iv) a quartic spline; and (v) a quintic spline. Space the knots uniformly across the quantiles of the data, using the method we learned in class. Please paste the five regressions output below (using stargazer)

Spline Regression Results

	Dependent variable:			
	user_rating			
	(1)	(2)	(3)	(4) (5)
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 1)1	0.120**			
	(0.052)			
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 1)2	0.241***			
	(0.043)			
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 1)3	0.400***			
	(0.045)			
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 1)4	0.520***			
	(0.044)			
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 1)5	0.542***			
	(0.041)			
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 1)6	0.407			
	(0.309)			
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 2)1		0.035		
		(0.070)		
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 2)2		0.214***		
		(0.053)		
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 2)3		0.310***		
		(0.058)		
				(2.032)
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 5)8				
-3.857				
				(4.645)
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 5)9				
3.865				
				(3.635)
bs(rating_count_tot, knots = c(109, 324.957, 966.5, 3048.516, 13101.113), degree = 5)10				
-0.304				
				(0.630)
Constant	3.759***	3.772***	3.793***	3.813*** 3.832***
	(0.036)	(0.046)	(0.055)	(0.062) (0.068)
Observations	5,130	5,130	5,130	5,130 5,130
R2	0.080	0.080	0.081	0.081 0.081
Adjusted R2	0.079	0.079	0.080	0.079 0.079
Residual Std. Error	0.627 (df = 5123)	0.627 (df = 5122)	0.627 (df = 5121)	0.627 (df = 5120) 0.627 (df = 5119)
F Statistic	74.365*** (df = 6; 5123)	63.958*** (df = 7; 5122)	56.392*** (df = 8; 5121)	50.193*** (df = 9; 5120) 45.273*** (df = 10; 5119)

Note:
*p<0.1; **p<0.05; ***p<0.01

E) Create a matrix of five distinct graphs. In each graph, please put each regression fitted splines, with: (i) dots in gray; (ii) splines in blue; (iii) vertical dashed lines where the knots are located; and (iv) a legend indicating the degree.



F) Based on the above regressions, did adding knots and higher degrees improve the fit of the regression substantially? Up to which point? Which model would you choose?

Your answer: Increasing the number of notes and degrees of polynomials did not substantially improve the model after a certain point degree 3. Afterwards the model seemed to be overfitting and modeling some noise.

6. Local Regression (4 points)

A) What is local regression? How does it work? What is it useful for? Please provide two sentences in your own words.

Local regression is a non-parametric type of regression that does not assume a function for the data.Instead of fitting the whole regression to a function local regression focuses on fitting the data to a ''neighborhood" of the data .It takes one point and creates a model by looking at the variabel close to that point on a certain range.It is quite useful for exploring the data looking for trends on how the data moves across the set , it can also assist you in choosing a polynomial by looking into the pattern of the data and provides good predictions.

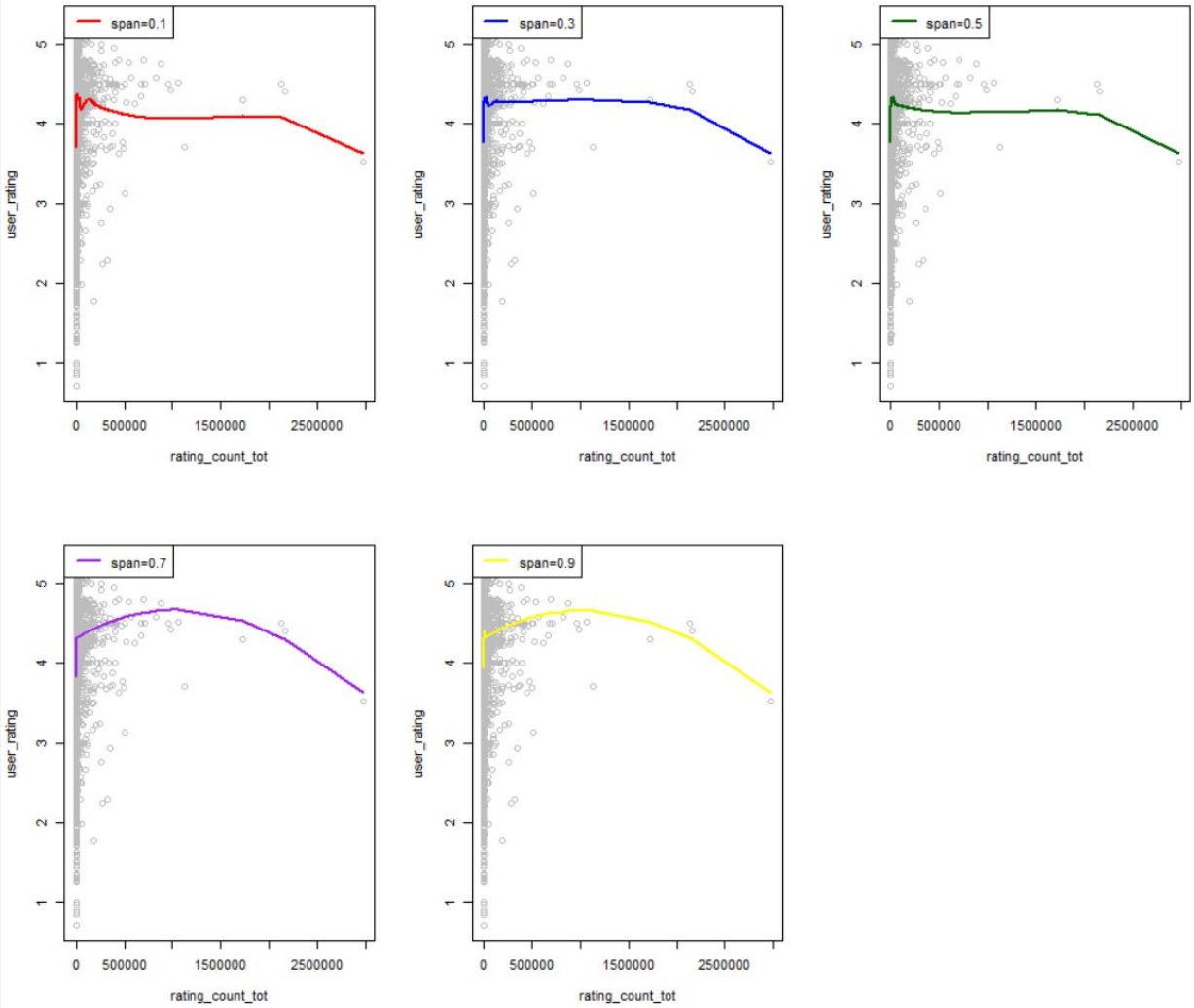
B) What is the span of a local regression? In general, would a regression curve tend to be more variable (visually) when the span is large or small? Why?

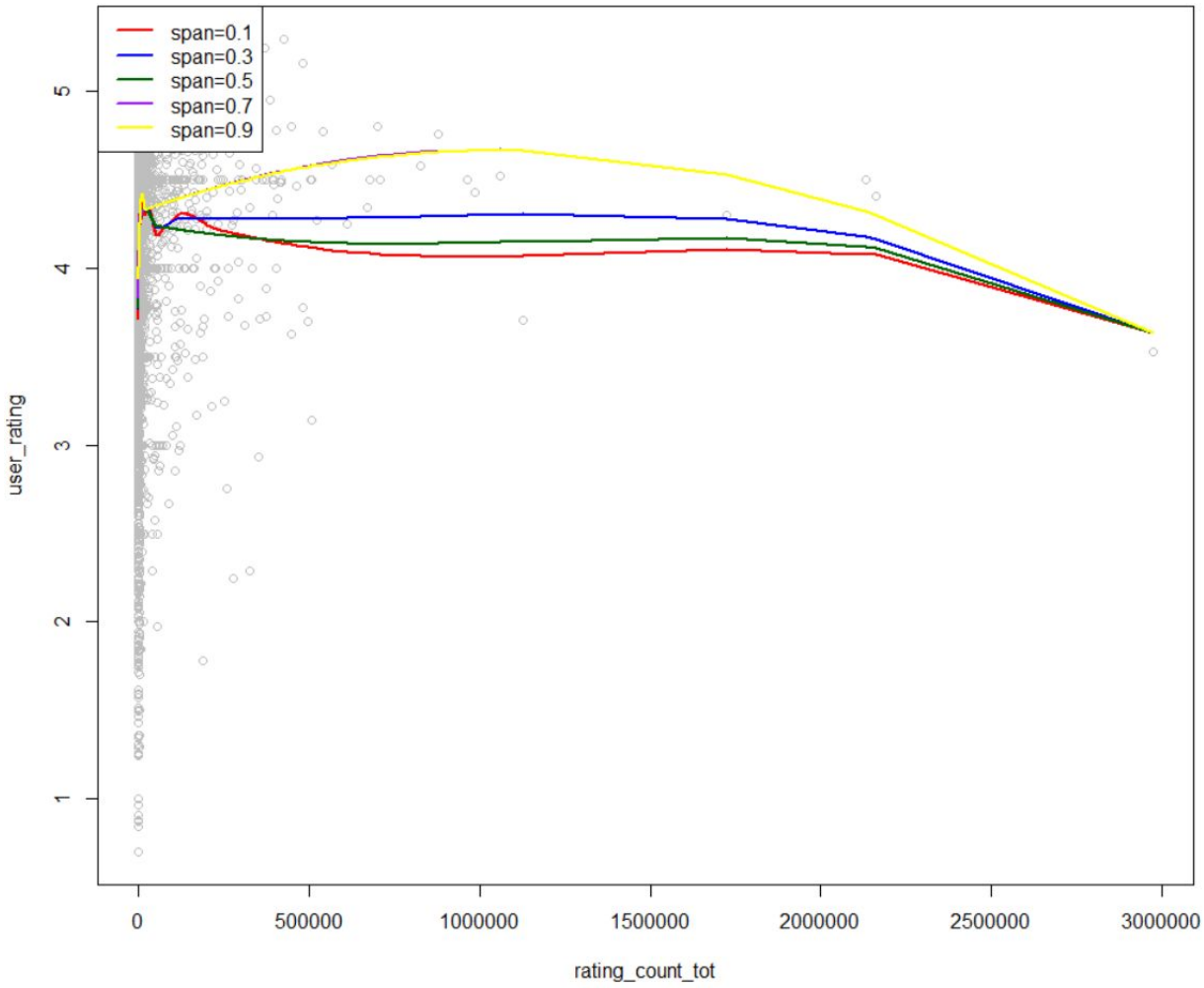
The span in a local regression defines the proportion of observation taken in the neighborhood.
The smaller the span the more flexible is the fit as the proportion of observation that need to be accounted for to make the fit is smaller .

C) Let's focus on the following relationship:

User_rating=f(rating_count_tot)

Run five local regressions using four different spans (0.1,0.3, 0.5, 0.7, 0.9). Plot the fitted lines below:


span=0.1
span=0.3
span=0.5
span=0.7
span=0.9
user_rating
rating_count_tot
0
500000
1500000
2500000
1
2
3
4
5



D) For each of these five regressions, "predict" the average rating of an app with (i) 1500 ratings, (ii) 3000 ratings, and (iii) 5000 ratings

Your predictions:

Span/Ratings	1500	3000	7000
Span=0.1	4.191185	4.284305	4.249666
Span=0.3	4.208398	4.267332	4.280554
Span=0.5	4.20564	4.27432	4.290623
Span=0.7	4.215771	4.281081	4.29347
Span=0.9	4.066335	4.172618	4.361502

```
############################
RCODE
############################


####Question1####
#A#


lab4_apple_store <- read.csv("C:/Users/Ritz/Downloads/lab4_apple_store.csv")
attach(lab4_apple_store)
lm_rating1=lm(user_rating~sup_devices.num,data=lab4_apple_store)
lm_rating2=lm(user_rating~rating_count_tot,data=lab4_apple_store)
lm_rating3=lm(user_rating~sup_devices.num+rating_count_tot,data =
lab4_apple_store)
install.packages("stargazer")
library(stargazer)

stargazer(lm_rating1,lm_rating2,lm_rating3,title = "Regression Output",align =
TRUE,type="html")
summary(lm_rating1)
summary(lm_rating2)
summary(lm_rating3)
#C#

#creating a 2 matrix plot
par(mfrow=c(1,2))
plot(rating_count_tot,user_rating,col="red")
plot(sup_devices.num,user_rating,col="blue")
#E#
library(car)
residualPlots(lm_rating1)
residualPlots(lm_rating2)
residualPlots(lm_rating3)


#Question 2

install.packages("stargazer")

library(stargazer)

#Running 3 regressions with different degrees polynomials
```

```
lab4_app_store = read.csv("lab4_apple_store.csv")

attach(lab4_app_store)

reg_poly_00 = lm(user_rating~sup_devices.num)

reg_poly_01 = lm(user_rating~poly(sup_devices.num,2))

reg_poly_02 = lm(user_rating~poly(sup_devices.num, 3))

stargazer(reg_poly_00, reg_poly_01, reg_poly_02, title="Results", align=TRUE, type =
"text")

library(car)


#running the anova for the three models with different degrees

anova(reg_poly_00, reg_poly_01, reg_poly_02)

plot(rating_count_tot, user_rating)


####Question 3####

#A#

reg_poly1=lm(user_rating~rating_count_tot)
reg_poly2=lm(user_rating~poly(rating_count_tot, 2))
reg_poly3=lm(user_rating~poly(rating_count_tot, 3))


install.packages("stargazer")
library(stargazer)

stargazer(reg_poly1, reg_poly2, reg_poly3, title="Regression Results", align=TRUE,
type="html")

#B#
par(mfrow=c(1,3))

plot(rating_count_tot, user_rating, col="gray")
lines(sort(rating_count_tot), predict(reg_poly1)[order(rating_count_tot)], col="red")
legend("topleft", pch="-", col=c("red"), c("Degree 1")) #legend

plot(rating_count_tot, user_rating, col="gray")
lines(sort(rating_count_tot), predict(reg_poly2)[order(rating_count_tot)], col="red")
legend("topleft", pch="-", col=c("red"), c("Degree 2")) #legend
```

```
plot(rating_count_tot, user_rating, col="gray")
lines(sort(rating_count_tot), predict(reg_poly3)[order(rating_count_tot)], col="red")
legend("topleft", pch="-", col=c("red"), c("Degree 3")) #legend


#C#
anova(reg_poly1, reg_poly2, reg_poly3)


####Question 4####

#A#
reg_poly_final=lm(user_rating~sup_devices.num+poly(rating_count_tot, 3))
summary(reg_poly_final)

#B#
stargazer(reg_poly_final, title="Regression Result", align=TRUE, type="html")


#C#
install.packages("plot3D")
library("plot3D")

# x, y, z variables
x <- lab4_apple_store$sup_devices.num
y <- lab4_apple_store$rating_count_tot
z <- lab4_apple_store$user_rating

# Compute the linear regression (z = ax + by + d)
fit <- lm(z ~ x + poly(y, 3))

# predict values on regular xy grid
grid.lines = 26
x.pred <- seq(min(x), max(x), length.out = grid.lines)
y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid( x = x.pred, y = y.pred)
z.pred <- matrix(predict(fit, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)

# fitted points for droplines to surface
fitpoints <- predict(fit)


# scatter plot with regression plane
par(mfrow=c(1,1))
scatter3D(x, y, z, pch = 18, cex = 0.7,
          theta = 110, phi = 15, ticktype = "detailed",
```

```
        xlab = "sup_devices.num", ylab = "rating_count_tot", zlab = "user_rating",
        surf = list(x = x.pred, y = y.pred, z = z.pred,
                    facets = NA), main = "Regression Plane of user_rating")
```

#Question 5: Spline Regression

```
#calculating for the knots

quantile(rating_count_tot, c(.2,.4,.6,.8))

# 135.0 490.6 1859.6 9517.4

library(splines)

splin_reg_d1 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6;
9517.4), degree = 1))

splin_reg_d2 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6;
9517.4), degree = 2))

splin_reg_d3 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6;
9517.4), degree = 3))

splin_reg_d4 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6;
9517.4), degree = 4))


stargazer(splin_reg_d1, splin_reg_d2, splin_reg_d3, splin_reg_d4, title = "Spline
Regression Results", align=TRUE, type = "html")

par(mfrow=c(1,5))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d1)[order(rating_count_tot)], lwd =2, col
= "blue")

abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch=1, col=c("green"), c("Degree 1"))

plot(rating_count_tot, user_rating, col = "gray")
```

```
lines(sort(rating_count_tot), predict(splin_reg_d2)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch= 1, col=c("blue"), c("Degree 2"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d3)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch=1, col=c("purple"), c("Degree 3"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d4)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch=1, col=c("red"), c("Degree 4"))


#Five Knots, calculating the quantiles

quantile(rating_count_tot, c(0.167, 0.333, 0.50, 0.667, 0.833))

splin_reg_d1_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957, 966.500,  3048.516, 13101.113 ), degree = 1))

splin_reg_d2_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957, 966.500,  3048.516, 13101.113 ), degree = 2))
```

```
splin_reg_d3_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 3))

splin_reg_d4_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 4))

splin_reg_d5_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 5))

stargazer(splin_reg_d1_00, splin_reg_d2_00, splin_reg_d3_00,
splin_reg_d4_00,splin_reg_d5_00, title = "Spline Regression Results", type = "text")


#plotting the scatterplot matrix

par(mfrow=c(2,3))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d1_00)[order(rating_count_tot)], lwd =2,
col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch=1, col=c("green"), c("Degree 1"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d2_00)[order(rating_count_tot)], lwd =2,
col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch= 1, col=c("blue"), c("Degree 2"))

plot(rating_count_tot, user_rating, col = "gray")
```

```
lines(sort(rating_count_tot), predict(splin_reg_d3_00)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch=1, col=c("purple"), c("Degree 3"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d4_00)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch=1, col=c("red"), c("Degree 4"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d5_00)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch=1, col=c("orange"), c("Degree 5"))
```

####Question6####

#C#

```
par(mfrow=c(2,3))
localreg_1=loess(user_rating~rating_count_tot,span=0.1)
localreg_2=loess(user_rating~rating_count_tot,span=0.3)
localreg_3=loess(user_rating~rating_count_tot,span=0.5)
localreg_4=loess(user_rating~rating_count_tot,span=0.7)
localreg_5=loess(user_rating~rating_count_tot,span=0.9)
plot(rating_count_tot,user_rating,col="grey")
lines(sort(rating_count_tot),predict(localreg_1)[order(rating_count_tot)],lwd=2,col="red
")
legend("topleft",lty=1,lwd = 2,col=c("red"),c("span=0.1"))
plot(rating_count_tot,user_rating,col="grey")
lines(sort(rating_count_tot),predict(localreg_2)[order(rating_count_tot)],lwd=2,col="blu
e")
legend("topleft",lty=1,lwd = 2,col=c("blue"),c("span=0.3"))
plot(rating_count_tot,user_rating,col="grey")
lines(sort(rating_count_tot),predict(localreg_3)[order(rating_count_tot)],lwd=2,col="dar
k green")
legend("topleft",lty=1,lwd = 2,col=c("dark green"),c("span=0.5"))
plot(rating_count_tot,user_rating,col="grey")
lines(sort(rating_count_tot),predict(localreg_4)[order(rating_count_tot)],lwd=2,col="pur
ple")
legend("topleft",lty=1,lwd = 2,col=c("purple"),c("span=0.7"))
plot(rating_count_tot,user_rating,col="grey")
lines(sort(rating_count_tot),predict(localreg_5)[order(rating_count_tot)],lwd=2,col="yell
ow")
legend("topleft",lty=1,lwd = 2,col=c("yellow"),c("span=0.9"))

#or grouped
par(mfrow=c(1,1))
localreg_1=loess(user_rating~rating_count_tot,span=0.1)
localreg_2=loess(user_rating~rating_count_tot,span=0.3)
localreg_3=loess(user_rating~rating_count_tot,span=0.5)
localreg_4=loess(user_rating~rating_count_tot,span=0.7)
localreg_5=loess(user_rating~rating_count_tot,span=0.9)
plot(rating_count_tot,user_rating,col="grey")
lines(sort(rating_count_tot),predict(localreg_1)[order(rating_count_tot)],lwd=2,col="red
")
lines(sort(rating_count_tot),predict(localreg_2)[order(rating_count_tot)],lwd=2,col="blu
e")
lines(sort(rating_count_tot),predict(localreg_3)[order(rating_count_tot)],lwd=2,col="dar
k green")
lines(sort(rating_count_tot),predict(localreg_4)[order(rating_count_tot)],lwd=2,col="pur
```

```
ple")
lines(sort(rating_count_tot),predict(localreg_5)[order(rating_count_tot)],lwd=2,col="yell
ow")
legend("topleft",lty=1,lwd = 2,col=c("red","blue","dark
green","purple","yellow"),c("span=0.1","span=0.3","span=0.5","span=0.7","span=0.
9"))
```

#D#

```
#local regression predictions

predict(localreg_1,1500)
predict(localreg_1,3000)
predict(localreg_1,7000)
predict(localreg_2,1500)
predict(localreg_2,3000)
predict(localreg_2,7000)
predict(localreg_3,1500)
predict(localreg_3,3000)
predict(localreg_3,7000)
predict(localreg_4,1500)
predict(localreg_4,3000)
predict(localreg_4,7000)
predict(localreg_5,1500)
predict(localreg_5,3000)
predict(localreg_5,7000)
```

#Code for 2 & 5

```
#Question 5: Spline Regression

#calculating for the knots

quantile(rating_count_tot, c(.2,.4,.6,.8))

# 135.0 490.6 1859.6 9517.4

library(splines)
```

```
splin_reg_d1 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6; 9517.4), degree = 1))

splin_reg_d2 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6; 9517.4), degree = 2))

splin_reg_d3 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6; 9517.4), degree = 3))

splin_reg_d4 = lm(user_rating~bs(rating_count_tot, knots = c(135.0, 490.6, 1859.6; 9517.4), degree = 4))


stargazer(splin_reg_d1, splin_reg_d2, splin_reg_d3, splin_reg_d4, title = "Spline Regression Results", align=TRUE, type = "html")

par(mfrow=c(1,5))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d1)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch=1, col=c("green"), c("Degree 1"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d2)[order(rating_count_tot)], lwd =2, col = "blue")

abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch= 1, col=c("blue"), c("Degree 2"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d3)[order(rating_count_tot)], lwd =2, col = "blue")
```

```
abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch=1, col=c("purple"), c("Degree 3"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d4)[order(rating_count_tot)], lwd =2, col
= "blue")

abline(v=135, lty =2)

abline(v=490.6, lty =2)

abline(v=1859.6, lty =2)

abline(v=9517.4, lty =2)

legend("topleft", pch=1, col=c("red"), c("Degree 4"))


#Five Knots, calculating the quantiles

quantile(rating_count_tot, c(0.167, 0.333, 0.50, 0.667, 0.833))

splin_reg_d1_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 1))

splin_reg_d2_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 2))

splin_reg_d3_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 3))

splin_reg_d4_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 4))

splin_reg_d5_00 = lm(user_rating~bs(rating_count_tot, knots = c(109.000,   324.957,
966.500,  3048.516, 13101.113 ), degree = 5))

stargazer(splin_reg_d1_00, splin_reg_d2_00, splin_reg_d3_00,
splin_reg_d4_00,splin_reg_d5_00, title = "Spline Regression Results", type = "text")


#plotting the scatterplot matrix

par(mfrow=c(2,3))
```

```
plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d1_00)[order(rating_count_tot)], lwd =2,
col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch=1, col=c("green"), c("Degree 1"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d2_00)[order(rating_count_tot)], lwd =2,
col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch= 1, col=c("blue"), c("Degree 2"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d3_00)[order(rating_count_tot)], lwd =2,
col = "blue")

abline(v=109, lty =2)

abline(v=324.957, lty =2)

abline(v=966.500, lty =2)

abline(v=3048.516, lty =2)

abline(v=13101.113, lty =2)

legend("topleft", pch=1, col=c("purple"), c("Degree 3"))

plot(rating_count_tot, user_rating, col = "gray")

lines(sort(rating_count_tot), predict(splin_reg_d4_00)[order(rating_count_tot)], lwd =2,
col = "blue")
```

```
abline(v=109, lty =2)
abline(v=324.957, lty =2)
abline(v=966.500, lty =2)
abline(v=3048.516, lty =2)
abline(v=13101.113, lty =2)
legend("topleft", pch=1, col=c("red"), c("Degree 4"))
plot(rating_count_tot, user_rating, col = "gray")
lines(sort(rating_count_tot), predict(splin_reg_d5_00)[order(rating_count_tot)], lwd =2,
col = "blue")
abline(v=109, lty =2)
abline(v=324.957, lty =2)
abline(v=966.500, lty =2)
abline(v=3048.516, lty =2)
abline(v=13101.113, lty =2)
legend("topleft", pch=1, col=c("orange"), c("Degree 5"))
```