

Quiz 3 - Take-home - Solutions

This is a 24-hour take-home quiz. It will be released on Crowdmark Friday, July 9 at 2pm, and will be due Saturday, July 10 at 2pm.

- You may not collaborate with each other on Quiz 3.
- You may not ask questions on external online forums (such as StackExchange, Kaggle, Chegg, etc.).
- You may ask clarification questions on the MATA35 class Piazza for me or your TAs to answer. This is to ensure that everyone has access to the same information.
- You may use the textbook, as well as general online math resources, so long as they are not specific to the problems on Quiz 3. e.g. You may look up a tutorial on how to do linear regression using Python, but if you come across one of the quiz problems on Chegg, you may not use the provided specific solution (indeed, I'd appreciate if you let me know if you encounter the exact quiz question on Chegg).
- You may make full use of calculators (online or physical). However, some problems may ask you to show your work, in which case you cannot simply answer that the calculator solved it for you. Some online calculators and computer algebra systems are extremely powerful and can fully solve some of the quiz problems. However, you should show still your work as if you solved it with only the help of an ordinary scientific calculator.
- Problems 1 through 4 should be submitted to Crowdmark as usual.
- For problems 5 and 6, you should write a report including all of your code, as well as your free-form answers to the questions. I recommend writing a report using the Markdown features of Google Colab (<https://colab.research.google.com> (<https://colab.research.google.com>)) / Jupyter Notebook via UToronto Syzygy (<https://utoronto.syzygy.ca> (<https://utoronto.syzygy.ca>)), such as found in the Practice Quiz 3 solutions guide: (https://colab.research.google.com/drive/18tze1JA0_u4v6mL1L5oOouFvgKiK9JVC?usp=sharing (https://colab.research.google.com/drive/18tze1JA0_u4v6mL1L5oOouFvgKiK9JVC?usp=sharing))

Problem 1: Multivariable functions [10pts]

Consider the function

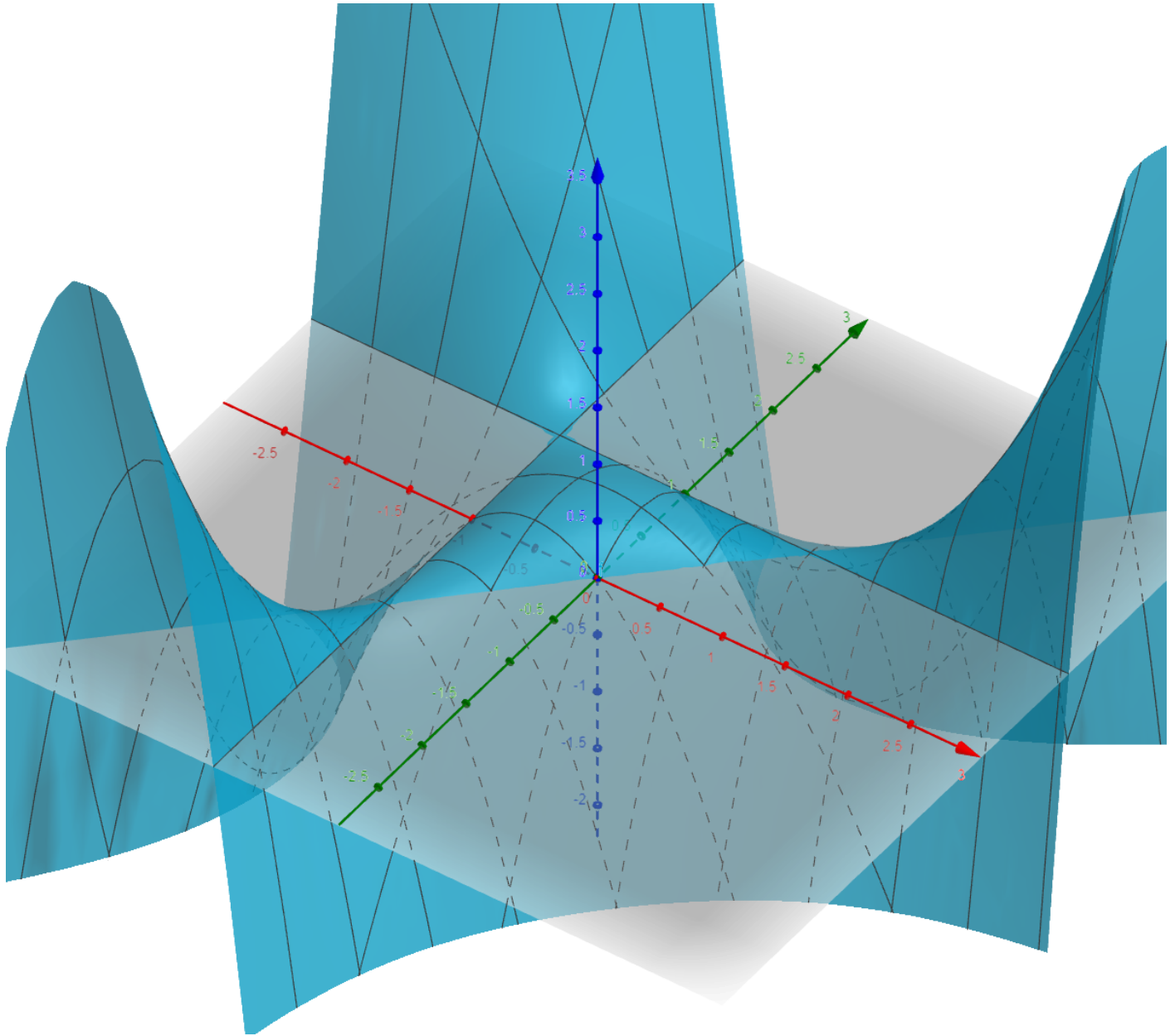
$$f(x, y) = (x - y)(y - 1)(x + 1)$$

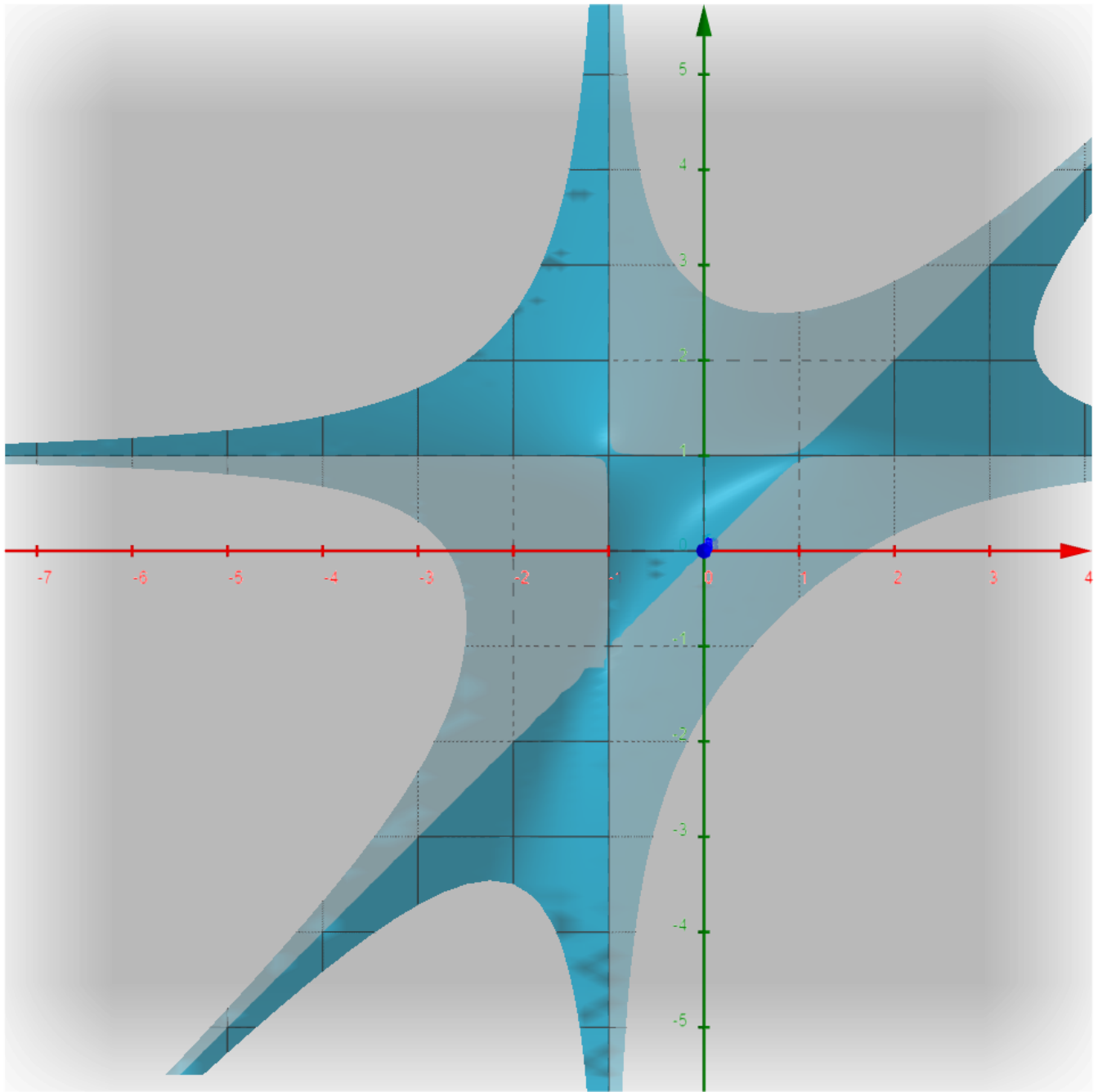
Graph this function at several different zoom levels and describe the behavior of the function in words.

1. **[5pts]** Do you see any local extrema or saddle points? Where are they?
2. **[5pts]** What is the behavior of the function as you go away from the origin in different directions? You should describe the behavior as you go in each direction along the axes, as well as along each of the diagonal directions.

Please include images of the function in your report, illustrating your description above.

Problem 1 Solutions





Problem 1.1 solution

There appear to be saddle points at $(-1, 1)$, $(1, 1)$, and $(-1, -1)$ from looking at the image above. Additionally, there looks to be a local maximum in the quadrant with negative x-value and positive y-value, closer to the origin than the saddle point $(-1, 1)$. This local maximum is actually at $(-\frac{1}{3}, \frac{1}{3})$, as we will compute in Problem 2, but that is difficult to see from the visualization.

Marking notes: Each of the saddle points is worth 1pt for noticing. The local maximum is worth 2pts. Note that the student does not need to specify the exact location to get credit, just approximately where it is.

Problem 1.2 solution

- **[1pt]** In both directions along the x-axis, the function appears to go to negative infinity.
- **[1pt]** In both directions along the y-axis, the function appears to go to negative infinity.
- **[1pt]** In the diagonal direction where both x and y are positive, as well as the opposite where both x and y are negative, the function appears to be 0, but balanced such that if you go slightly to the left or right of that diagonal line, you rapidly get very large or very small.
- **[1pt]** In the diagonal direction where x is positive, but y is negative, the function appears to go to negative infinity.
- **[1pt]** In the diagonal direction where x is negative, but y is positive, the function first goes through the local max and the saddle point, but then continues on towards positive infinity.

Problem 2: Partial derivatives [12pts]

Compute all of the 1st and 2nd-order partial derivatives of the function $f(x, y)$ from Problem 1 by hand.

Note: You must show all of your work for computing the derivatives by hand. You may however use a calculator / online resources to double check your work, which I encourage because Problem 3 depends on the solutions to Problem 2.

Note: Make sure you explicitly mark each of your steps. You will not get credit if you do not show your work.

Problem 2 solution

$$f(x, y) = (x - y)(y - 1)(x + 1)$$

1st-order partial derivatives:

- $\frac{\partial f}{\partial x} = (x - y)(y - 1) + (y - 1)(x + 1) = (y - 1)(2x - y + 1)$
- $\frac{\partial f}{\partial y} = (x - y)(x + 1) - (y - 1)(x + 1) = (x + 1)(x - 2y + 1)$

2nd-order partial derivatives:

- $\frac{\partial^2 f}{\partial x^2} = 2y - 2$
- $\frac{\partial^2 f}{\partial y \partial x} = 2x - 2y + 2$
- $\frac{\partial^2 f}{\partial x \partial y} = 2x - 2y + 2$
- $\frac{\partial^2 f}{\partial y^2} = -2x - 2$

Problem 3: Maximums and minimums [24pts]

1. **[8pts]** Find all critical points of $f(x, y)$ from Problem 1 using the **Sanity Check**: did you get all the critical points you saw in Problem 1?
2. **[16pts]** Use the various derivative tests (e.g. eigenvalues of the Hessian or the D-test from the book) we learned to classify each critical point as either a minimum, maximum, or saddle point. If it is not possible to classify, explain why.

Problem 3.1 Solutions

Setting both 1st-order partial derivatives equal to 0, you get the following set of conditions:

- $y = 1$ or $y = 2x + 1$ so that $f_x = 0$
- $x = -1$ or $x = 2y - 1$ so that $f_y = 0$

Thus, there are four cases:

- Case 1: $y = 1$ and $x = -1$, so we get $(-1, 1)$
- Case 2: $y = 1$ and $x = 2y - 1$, so we get $(1, 1)$
- Case 3: $y = 2x + 1$ and $x = -1$, so we get $(-1, -1)$
- Case 4: $y = 2x + 1$ and $x = 2y - 1$, so we get $(-\frac{1}{3}, \frac{1}{3})$

Marking notes: Finding each critical point is worth 2pts. The sanity check is not marked, just there as a hint to the students.

Problem 3.2 Solutions

Marking notes: The student has two options. One is the D-test:

- Let $D = f_{xx} \cdot f_{yy} - f_{xy}^2$.
- If $D > 0$, and $f_{xx} < 0$, then maximum.
- If $D > 0$, and $f_{xx} > 0$, then minimum.
- If $D < 0$, then saddle point.
- If $D = 0$, then test fails.

Another is to look at the eigenvalues of the Hessian. I am going to give the Hessian solution below, but either should get full credit. Each part is worth 4 points.

$$H(x, y) = \begin{bmatrix} 2y - 2 & 2x - 2y + 2 \\ 2x - 2y + 2 & -2x - 2 \end{bmatrix}$$

- $H(-1, 1) = \begin{bmatrix} 0 & -2 \\ -2 & 0 \end{bmatrix}$, which has eigenvalues $\lambda = \pm 2$, so saddle point.
- $H(1, 1) = \begin{bmatrix} 0 & 2 \\ 2 & -4 \end{bmatrix}$, which has eigenvalues $\lambda = -2 \pm 2\sqrt{2}$, so saddle point.
- $H(-1, -1) = \begin{bmatrix} -4 & 2 \\ 2 & 0 \end{bmatrix}$, which has eigenvalues $\lambda = \pm 2$, so saddle point.
- $H(-\frac{1}{3}, \frac{1}{3}) = \begin{bmatrix} -\frac{4}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{4}{3} \end{bmatrix}$, which has eigenvalues $\lambda = -\frac{2}{3}, -2$, so maximum.

Problem 4: Multivariable integration [14pts]

Integrate $g(x, y) = \frac{-x^3 + 2x^2 + x}{(y+1)^2}$ over the region bounded by the following inequalities:

- $y \leq -x^2 + 2x$
- $y \geq 0$
- $y \geq x - 1$

Give your answer to 3 decimal places. Show all of your work, but you may use a calculator for evaluating complicated algebraic expressions. Make sure to explain **in words** each of the steps you are doing.

Note: You must write out all the work for doing the double integration by hand. You will not get credit if you simply plug in the double integral into an online calculator.

Note: the function $g(x, y)$ may be negative in some places, so the interpretation as volume doesn't make as much sense.

Problem 4 solution

[4pts] Write down appropriate bounds for each region of integration by dividing into two regions to integrate.

[4pts] Solving the blue region.

[4pts] Solving the green region.

[2pts] Combining the two together to get the final answer.

You should give partial credit. Note that they **MUST** show their work, and explain roughly what they are doing, or else you should take off points.



Note that we can divide it up into the blue and green regions above for easy integration. Also note that the intersection of the red line with the parabola is at $x = \frac{1+\sqrt{5}}{2}$

The blue region is

$$\int_{x=0}^{x=1} \int_{y=0}^{y=-x^2+2x} g(x, y) dy dx$$

The green region is

$$\int_{x=1}^{x=\frac{1+\sqrt{5}}{2}} \int_{y=0}^{y=-x^2+2x} g(x, y) dy dx$$

Solving both equations gives:

$$\begin{aligned} & \int_0^1 \int_0^{-x^2+2x} \frac{x(-x^2+2x+1)}{(y+1)^2} dy dx \\ &= \int_0^1 \left[-\frac{x(-x^2+2x+1)}{y+1} \right]_0^{-x^2+2x} dx \\ &= \int_0^1 \left[-x + x(-x^2+2x+1) \right] dx \\ &= \int_0^1 \left[-x^3 + 2x^2 \right] dx \\ &= \left[-\frac{x^4}{4} + \frac{2x^3}{3} \right]_0^1 = -\frac{1}{4} + \frac{2}{3} = \frac{5}{12} \\ &\approx 0.416667 \end{aligned}$$

$$\begin{aligned} & \int_1^{\frac{1+\sqrt{5}}{2}} \int_{x-1}^{-x^2+2x} \frac{x(-x^2+2x+1)}{(y+1)^2} dy dx \\ &= \int_1^{\frac{1+\sqrt{5}}{2}} \left[-x + (-x^2+2x+1) \right] dx \\ &= \int_1^{\frac{1+\sqrt{5}}{2}} \left[-x^2 + x + 1 \right] dx \\ &= \left[-\frac{1}{3}x^3 + \frac{1}{2}x^2 + x \right]_1^{\frac{1+\sqrt{5}}{2}} \\ &= -\frac{1}{3} \left(\frac{1+\sqrt{5}}{2} \right)^3 + \frac{1}{2} \left(\frac{1+\sqrt{5}}{2} \right)^2 + \left(\frac{1+\sqrt{5}}{2} \right) + \frac{1}{3} - \frac{1}{2} - 1 \\ &\approx 0.348362 \end{aligned}$$

$$\text{Final answer} = 0.765029$$

Problem 5: Linear Regression Analysis [20pts]

For problems 5 and 6, please include your Python code and any generated graphs. Note that some of the answers are qualitative analyses, with more than one right answer, and they will be marked on the overall quality of your reasoning.

On Crowdmark, note that we have combined questions 5 and 6 into a single response to make it easier for you to submit. I also encourage you to additionally submit a link to your code within your PDF report if you used a Google colab, but this is not required.

Baby names vary in frequency over time. The table below shows the relative frequencies of the names "Holly" and "Henry" as a function of time from the US Social Security records. The numbers are in units of number per 1000 baby girls for "Holly" and number per 1000 baby boys for "Henry".

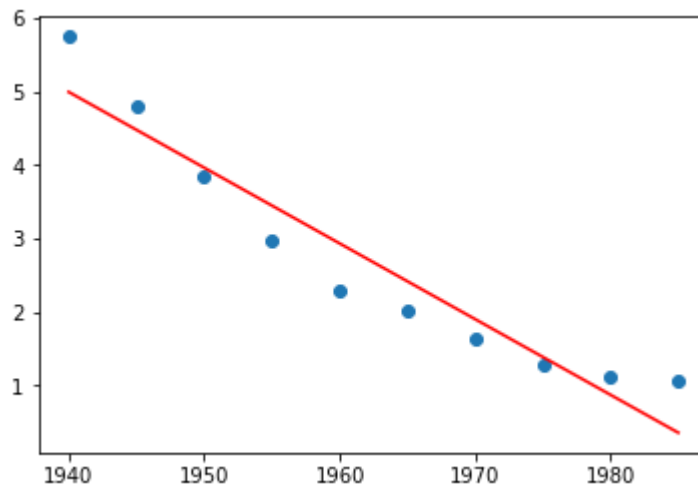
Year	Holly	Henry
1940	0.063	5.749
1945	0.284	4.811
1950	0.475	3.835
1955	1.248	2.966
1960	1.240	2.293
1965	1.836	2.006
1970	2.693	1.638
1975	3.235	1.282
1980	2.968	1.113
1985	3.037	1.071

1. [5pts] Use linear regression to find the best fit line for each name.
2. [5pts] What does your model predict to be the frequencies of each name in the year 1985?
3. [5pts] Predict the frequencies of each name in the year 2005.
4. [5pts] Without making reference to the actual values given in problem 6, explain why your predictions are likely to be good or bad.

```
In [1]: # We have to first import all the appropriate packages
import numpy as np # import numerical functions
from sklearn.linear_model import LinearRegression # import linear regression
from sklearn.metrics import mean_squared_error # import mean squared error
from matplotlib import pyplot as plt # import plotting tool
```

```
In [2]: # Solutions for parts 5.1-5.3 for Henry

# Let's now input the data from the table into Python
X = np.array([1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985]).res
hape((-1,1))
Y = np.array([5.749, 4.811, 3.835, 2.966, 2.293, 2.006, 1.638, 1.282, 1.113,
1.071]) # Henry
# We can also view it as a scatter plot
plt.scatter(X,Y)
# Then we build a Linear model:
linear_model = LinearRegression()
linear_model.fit(X,Y)
# And we also want to plot the predicted regression line:
plt.plot(X, linear_model.predict(X), color='red')
plt.show()
# And we can also print out the slope and intercept
print("Slope: ", linear_model.coef_[0])
print("Intercept:", linear_model.intercept_)
# Let's use the model we built above to predict the frequencies in 1985 and 20
05 for parts 5.2 and 5.3
print("The predicted frequencies in 1985 and 2005 for Henry are:", linear_mode
l.predict([[1985],[2005]]))
```



Slope: -0.10305939393939392

Intercept: 204.93046060606056

The predicted frequencies in 1985 and 2005 for Henry are: [0.35756364 -1.70362424]

```
In [8]: # Solutions for parts 5.1-5.3 for Holly
#(for ease, we're just copying the code, but in practice this is suboptimal)

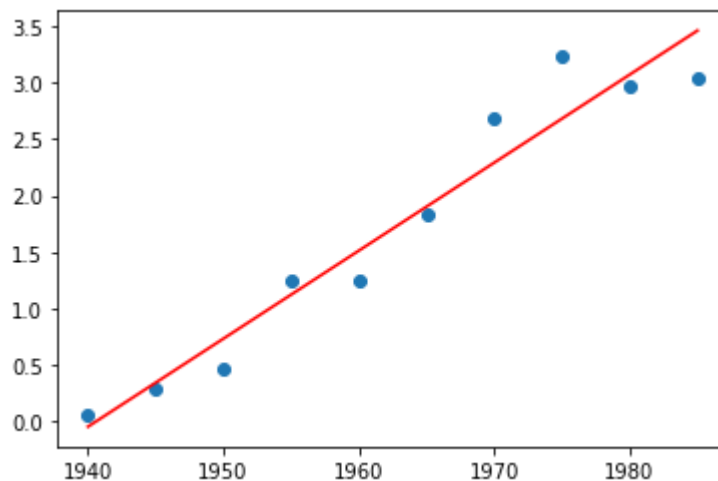
# Let's now input the data from the table into Python
X = np.array([1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985]).res
hape((-1,1))
Y = np.array([0.063, 0.284, 0.475, 1.248, 1.240, 1.836, 2.693, 3.235, 2.968,
3.037]) # Holly
# We can also view it as a scatter plot
plt.scatter(X,Y)
# Then we build a Linear model:
linear_model = LinearRegression()
linear_model.fit(X,Y)
# And we also want to plot the predicted regression line:
plt.plot(X, linear_model.predict(X), color='red')

# And we can also print out the slope and intercept
print("Slope: ", linear_model.coef_[0])
print("Intercept:", linear_model.intercept_)
# Let's use the model we built above to predict the frequencies in 1985 and 20
05 for parts 5.2 and 5.3
print("The predicted frequencies in 1985 and 2005 for Holly are:", linear_mode
l.predict([[1985],[2005]]))
plt.show()
```

Slope: 0.07792121212121214

Intercept: -151.21247878787884

The predicted frequencies in 1985 and 2005 for Holly are: [3.46112727 5.01955
152]



The best fit line for Henry is $y = -0.103059x + 204.93$. It predicts that in 1985, Henry will have frequency 0.3576, and in 2005, Henry will have frequency -1.7036.

The best fit line for Holly is $y = 0.077921x - 151.21$. It predicts that in 1985, Holly will have frequency 3.4611, and in 2005, Holly will have frequency 5.01955.

The predictions for 1985 both seem pretty good, as they track closely with the known values for 1985, so the model seems to be decent at at least explaining the training data. However, the predictions for 2005 are suspect, especially because Henry has a negative frequency, which is impossible---you cannot have a negative number of baby boys with the name "Henry". Thus, something must have gone wrong. One possibility is that we are just trying to extrapolate too far out, so our model is no longer very accurate that many years into the future. Note that although Holly's model predicts a positive value, we should also treat that with a grain of salt, given that it is also extrapolating far into the future.

Problem 6: Nonlinear regression [20pts]

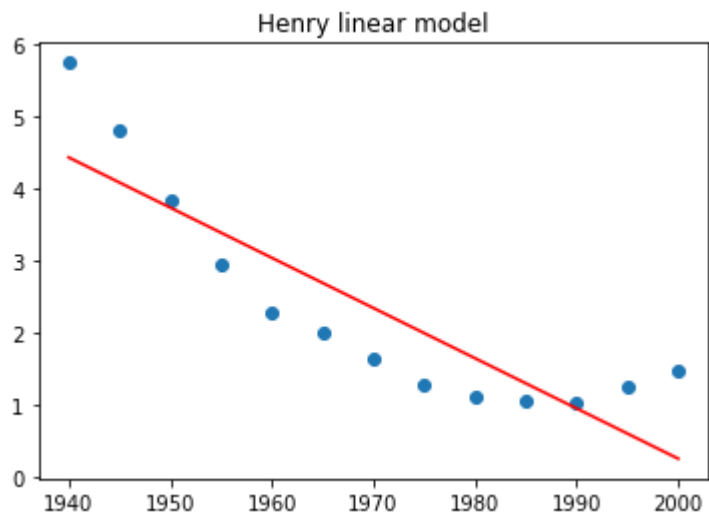
Consider the updated data below.

Year	Holly	Henry
1990	1.837	1.039
1995	1.325	1.253
2000	0.848	1.492

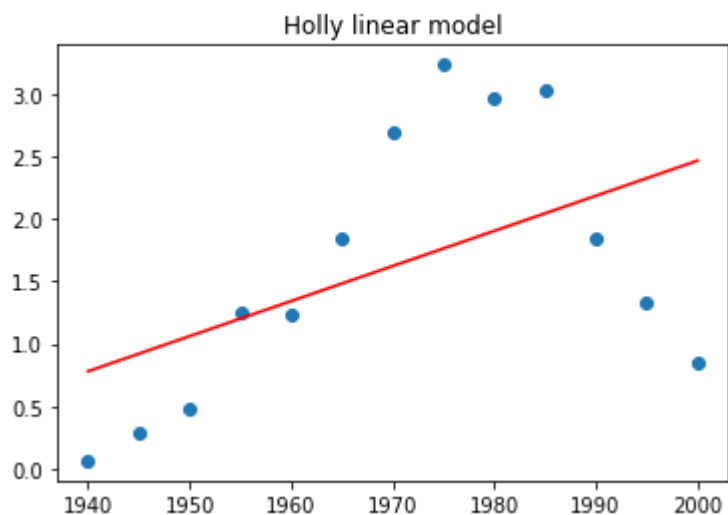
1. **[5pts]** Build a model that incorporates the updated data. Note that a linear model may no longer be appropriate, so you may wish to consider using nonlinear regression.
2. **[5pts]** Justify your choice of model(s) above.
3. **[5pts]** Predict the frequencies of each name in the year 2005 using your model. Note that the actual values are 0.510 and 1.964 for Holly and Henry respectively. How good was your model?
4. **[5pts]** Describe some of the remaining shortcomings of your model.

Let's start by trying out linear models with the updated data and see what happens.

```
In [4]: # Henry's updated Linear model
X = np.array([1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985, 1990,
1995, 2000]).reshape((-1,1))
Y_henry = np.array([5.749, 4.811, 3.835, 2.966, 2.293, 2.006, 1.638, 1.282, 1.113, 1.071, 1.039, 1.253, 1.492]) # Henry
plt.scatter(X,Y_henry)
linear_henry = LinearRegression()
linear_henry.fit(X,Y_henry)
plt.plot(X, linear_henry.predict(X), color='red')
plt.title('Henry linear model')
plt.show()
```



```
In [5]: # Holly's updated Linear model
X = np.array([1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985, 1990,
1995, 2000]).reshape((-1,1))
Y_holly = np.array([0.063, 0.284, 0.475, 1.248, 1.240, 1.836, 2.693, 3.235, 2.968, 3.037, 1.837, 1.325, 0.848]) # Holly
plt.scatter(X,Y_holly)
linear_holly = LinearRegression()
linear_holly.fit(X,Y_holly)
plt.plot(X, linear_holly.predict(X), color='red')
plt.title('Holly linear model')
plt.show()
```

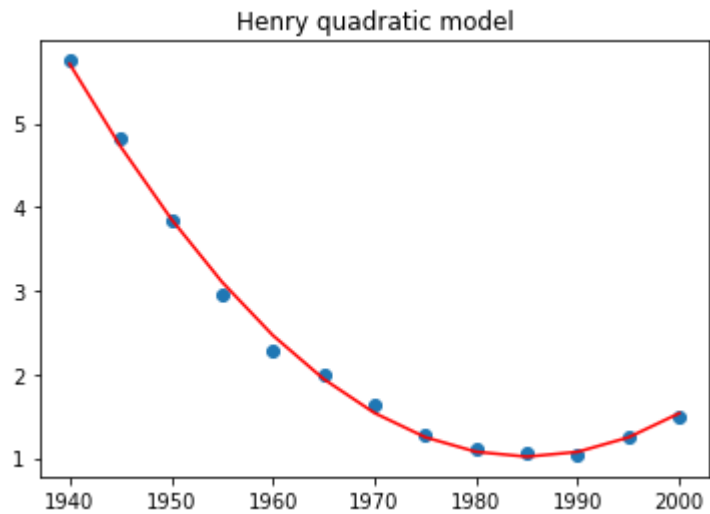


Unfortunately, upon getting the updated name frequency data, it becomes clear that the data is not actually linear. Indeed, upon examining the data, it looks almost quadratic, so perhaps a higher-order polynomial model would work better.

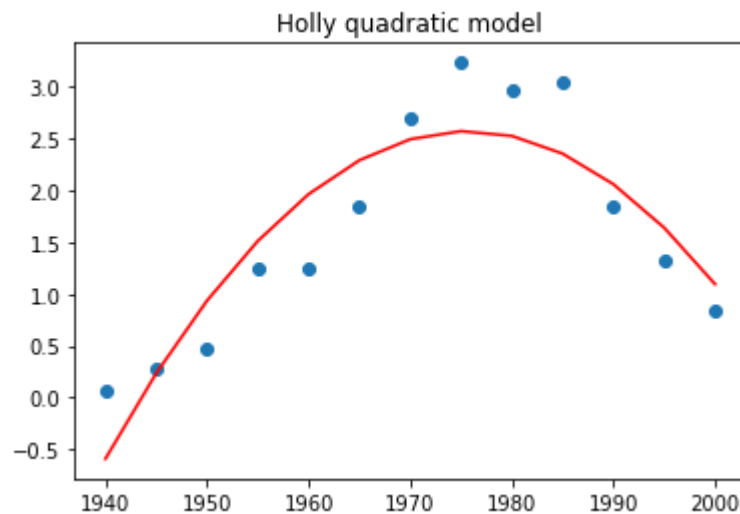
Let's try both, and look at how well the curves fit.

```
In [6]: # Quadratic regression - Henry
X_quad = np.hstack((X, X*X))
quadratic_henry = LinearRegression()
quadratic_henry.fit(X_quad, Y_henry)
plt.scatter(X, Y_henry)
plt.plot(X, quadratic_henry.predict(X_quad), color='red')
plt.title('Henry quadratic model')
plt.show()
print(quadratic_henry.predict([[2005, 2005*2005]]))

# Quadratic regression - Holly
quadratic_holly = LinearRegression()
quadratic_holly.fit(X_quad, Y_holly)
plt.scatter(X, Y_holly)
plt.plot(X, quadratic_holly.predict(X_quad), color='red')
plt.title('Holly quadratic model')
plt.show()
print(quadratic_holly.predict([[2005, 2005*2005]]))
```



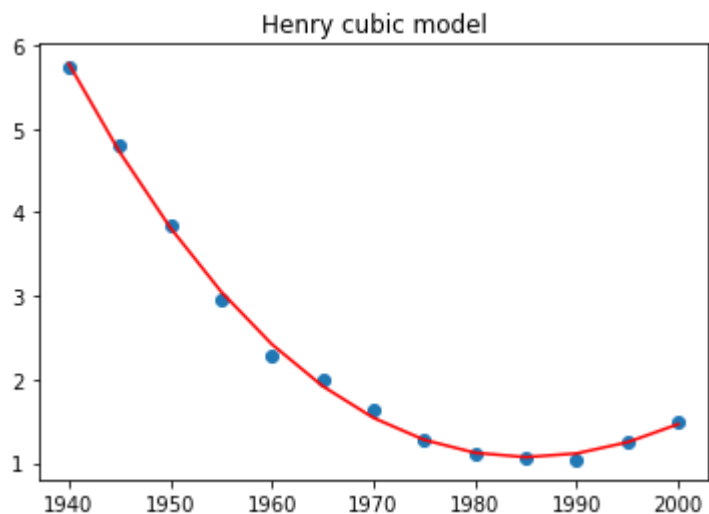
[1.94086014]



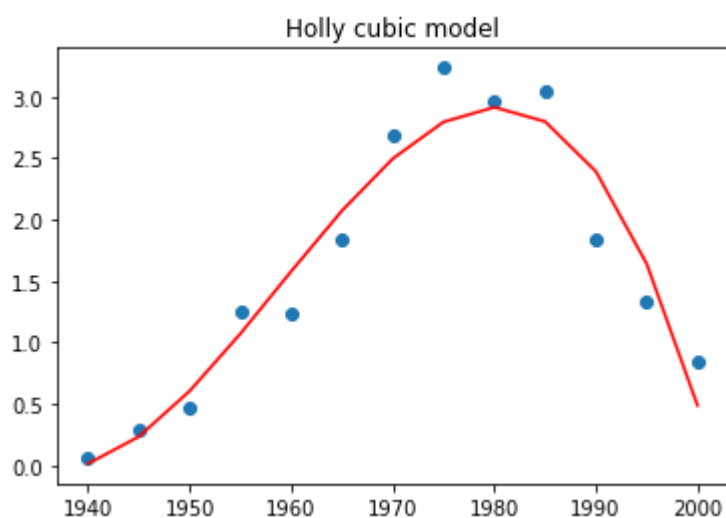
[0.42506294]


```
In [7]: # Cubic regression - Henry
X_cubic = np.hstack((X, X*X, X*X*X))
cubic_henry = LinearRegression()
cubic_henry.fit(X_cubic, Y_henry)
plt.scatter(X,Y_henry)
X_cubic = np.hstack((X, X*X, X*X*X))
plt.plot(X, cubic_henry.predict(X_cubic), color='red')
plt.title('Henry cubic model')
plt.show()
print(cubic_henry.predict([[2005, 2005*2005, 2005*2005*2005]]))

# Cubic regression - Henry
X_cubic = np.hstack((X, X*X, X*X*X))
cubic_holly = LinearRegression()
cubic_holly.fit(X_cubic, Y_holly)
plt.scatter(X,Y_holly)
X_cubic = np.hstack((X, X*X, X*X*X))
plt.plot(X, cubic_holly.predict(X_cubic), color='red')
plt.title('Holly cubic model')
plt.show()
print(cubic_holly.predict([[2005, 2005*2005, 2005*2005*2005]]))
```



[1.75802797]



[-1.11890214]

Given the above plots, we probably want to choose the quadratic model, as the quadratic model appears to fit nearly as well as the cubic model, and it is a good idea generally to minimize the number of parameters needed. Furthermore, an additional reason to choose the quadratic model is that the cubic model predicts Holly to have a negative frequency in 2005, which is what we're trying to predict here, and that's an aphysical prediction.

In terms of predictions for 2005, the quadratic model was quite close to the real answers (1.941 to 1.964 for Henry) and (0.425 to 0.510 for Holly). Thus, our model seems to have had accurate predictions for at least the near future.

Our models of course continue to have shortcomings. Notably, if we try to extrapolate too far out into either the past or the future, the numbers will get smaller than 0 or larger than 1000, either of which is impossible. This is of course always a problem with building models for extrapolation; once you exit the region of the training data, things might go wrong. Indeed, as a mathematical biologist who knows that the actual data is constrained between 0 and 1000, it might be a useful exercise to build a more sophisticated model that is guaranteed to lie between those constraints. However, that is beyond the scope of this course.