Question 2a Recall the optimal value of θ should minimize our loss function. One way we've approached solving for θ is by taking the derivative of our loss function with respect to θ, like we did in HW5.

Write/derive the expressions for following values and write them with LaTeX in the space below.

- $R(\mathbf{x}, \mathbf{y}, \theta_1, \theta_2)$: our loss function, the empirical risk/mean squared error
- $\frac{\partial R}{\partial \theta_1}$: the partial derivative of R with respect to θ_1
- $\frac{\partial R}{\partial \theta_2}$: the partial derivative of R with respect to θ_2

Recall that $R(\mathbf{x}, \mathbf{y}, \theta_1, \theta_2) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2$

The expressions are as follows:

$R(\mathbf{x}, \mathbf{y}, \theta_1, \theta_2) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \theta_1 x_i - sin(\theta_2 x_i))^2$

$\frac{\partial R}{\partial \theta_1} = \frac{1}{n}\sum_{i=1}^{n}(-2x_i(\mathbf{y}_i - \theta_1 x_i - sin(\theta_2 x_i)))$

$\frac{\partial R}{\partial \theta_2} = \frac{1}{n}\sum_{i=1}^{n}((-2x_i cos(\theta_2 x_i))(\mathbf{y}_i - \theta_1 x_i - sin(\theta_2 x_i)))$

In 1-2 sentences, describe what you notice about the path that theta takes with a static learning rate vs. a decaying learning rate. In your answer, refer to either pair of plots above (the 3d plot or the contour plot).

The path that theta takes with a static learning rate is a has a lot more ”jagged” lines while the path that theta takes with a decaying learning rate appears to be a lot more ”direct” (comparing contour plots). Theta appears to have fewer straight lines until it reaches the 0 (on the bottom left) with the decaying learning rate. This shows that we get faster convergence with our decaying learning rate.

0.0.1 Question 4b

Is this model reasonable? Why or why not?

It is not reasonable. Our model predicts a clear upward trend between number of points and win probability. When compared with the actual data, this is not necessarily true. E.g. at 125 points, our model predicts that the win probability is roughly 0.75 i.e. we are closer to a win than a defeat. However, there is a substantial number of actual results that lost with 125 points. The overlap between the points for a win and a defeat is too large for this model to be effective. At 125 points, we'd expect our win probability to be roughly 0.5, simply because there seems to be an equal cluster of points that lost with 125 points, and won with 125 points.

The model also doesn't deal well with outliers. For 70 points, the model predicts a win probability of roughly 0.65, which is alarming because this data point is our left-most point and contributed to the team losing. However, the model predicts that the team is closer to a win than a defeat ($P > 0.5$). Also, another strange feature of the model is that the win probability never falls below 0.5.

0.0.2 Question 4c

Try playing around with other theta values. You should observe that the models are all pretty bad, no matter what θ you pick. Explain why below.

This is because that doesn't solve the underlying issue that the win probability does not fall below 0.5. Because of this, we cannot take care of the outliers well enough.

0.0.3 Question 5b

Using the plot above, try adjusting θ_2 (only). Describe how changing θ_2 affects the prediction curve. Provide your description in the cell below.

Changing θ_2 from -5 to -6 causes the curve to become less steep, i.e. it takes a longer time to come close to the y=1 line. Making θ_2 a large negative value (such as -100) causes the prediction curve to lie flat on the x axis (becomes roughly a horizontal line at y=0). On the other hand, making θ_2 a large positive value (such as 100) causes the prediction curve to roughly look like the y=1 line. This happens because the prediction curve loses so much of its steepness, that we do not see it look like a ”curve” in our x values of [0, 160] because our x range is small.

0.0.4 Question 7c

Look at the coefficients in `theta_19_hat` and identify which of the parameters have the biggest effect on the prediction. For this, you might find `useful_numeric_fields.columns` useful. Which attributes have the biggest positive effect on a team's success? The biggest negative effects? Do the results surprise you?

In [143]:
```
theta_19_hat
```

Out[143]:
```
array([ 2.1239e+00, -4.5115e-01, -2.1898e+01,  9.2304e-01, -3.5553e-03,
        2.7252e+00,  8.8270e-01, -7.1648e-02,  2.1081e+00,  3.0481e-01,
        3.2550e-01,  4.8715e-02,  1.9556e-02,  3.8726e-01,  6.9643e-02,
       -3.1119e-01, -5.5789e-02, -7.5458e-01,  5.1914e+00])
```

In [134]:
```
useful_numeric_fields.columns
```

Out[134]:
```
Index(['FGM', 'FGA', 'FG_PCT', 'FG3M', 'FG3A', 'FG3_PCT', 'FTM', 'FTA',
       'FT_PCT', 'OREB', 'DREB', 'REB', 'AST', 'STL', 'BLK', 'TOV', 'PF',
       'PTS', 'BIAS'],
      dtype='object')
```

Parameters that have the biggest effect on the prediction include FGM, FG_PCT, FG_3_PCT, FT_PCT and BIAS.

The parameter that has the biggest positive effect on a team's success is BIAS. Note that BIAS is a term we added on our own, so we will ignore this and consider FGM (the attribute that has the second largest positive effect on a team's sucess). FGM refers to Field Goals Made. It makes sense that the larger the value of FGM, the larger the chance of the team's success (more baskets--> more points --> greater chance of a win). Other parameters that have a positive effect on a team's sucess include FG3_PCT which is the 3-Point Field Goal Percentage. This also makes sense because a 3-Point Field Goal is just a field goal made from beyond the three-point line, a designated arc surrounding the basket. The higher this value, the more likely the team is to win.

The parameter that has the biggest negative effect on a team's success is FG_PCT. The field goal percentage is the ratio of field goals made to field goals attempted. If this ratio is large, we would expect the number of field goals made to be quite high when compared to the number of field goals attempted, which would cause our team to be MORE likely to win. So yes, these results are surprising. This may be because so many of our parameters are related to one another. Another parameter that has a negative effect on a team's sucess is FGA (Field Goal Attempts). This doesn't tell us much, because we don't know if the attempts were successful or not (but chances are that the number of FGM increase with the number of FGA). Hence, this too, is surprising.

To double-check your work, the cell below will rerun all of the autograder tests.

```
In [145]: grader.check_all()
```

```
Out[145]: q1:

              All tests passed!

          q2b:

              All tests passed!

          q3a:

              All tests passed!

          q3b:

              All tests passed!

          q4a:

              All tests passed!

          q5a:

              All tests passed!

          q5c:

              All tests passed!

          q6a:

              All tests passed!

          q6b:

              All tests passed!

          q6c:
```

```
    All tests passed!

q6d:

    All tests passed!

q7a:

    All tests passed!

q7b:

    All tests passed!
```

0.1 Submission

Make sure you have run all cells in your notebook in order before running the cell below, so that all images/graphs appear in the output. The cell below will generate a zip file for you to submit. **Please save before exporting!**

```
In [ ]: # Save your notebook first, then run this cell to export your submission.
        grader.export("hw7.ipynb")
```