Phillies R&D Questionnaire

Problem a

Improve this code:

```
def is_palindrone(s):
    r=""
    for c in s:
        r = c +r
    for x in range(0, len(s)):
        if s[x] == r[x]:
            x = True
        else:
            return False
    return x
```

While this code is successful in determining whether the passed string is a palindrome or not, it's inefficient as it goes character by character rather than using Python's built in string methods that are optimized and much faster.

The first improvement I would make is in lines 2-4. This loop reverses the passed string *s* which can be done by using string slicing, rather than building it backwards by each character. Therefore, lines 2-4 can be replaced with:

```
r = s[::-1]
```

The second bit that can be improved is the second for loop. The current code is good in that it exits the for loop once a spot is found where the characters don't match, instead of finishing the whole string, however it can be optimized more. Again, we don't have to compare each character but can instead can do so with the whole string at once. Thus lines 5-10 can be replaced with:

```
if (s == r):
    return True
else:
    return False
```

These changes are made to eliminate unnecessary operations which would increase runtime when the size of the data/string get very large. The final optimized code looks like:

```
def is_palindrone(s):
    r = s[::-1]

    if (s == r):
        return True
    else:
        return False
```

Problem b

Here is an example of a visualization from the contract data. This is a histogram plotting all the valid salary data I pulled from the webpage and highlights the top 125 contracts by dollar value and the average of those which gives the qualifying offer value.