Coding Sample 2

June 29, 2019

1 Health Dataset Exploration

Background Information: 1. A large company, Company A, provides health insurance to its employees.

2. Every four years, Company A's insurer, InsurAHealth, reviews the health status of the employees. To do this, InsurAHealth calculates a health score between 0 and 6 for each employee on a quarterly basis. 0 denotes a very healthy person, and 6 denotes a very sick person. The 'health score' is a proprietary tool used by InsurAHealth. The items that go into its formula are not public.

3. This past review cycle InsurAHealth claimed that the employees have gotten sicker. Mean Health Score in Quarter 1 was 3.4, in Quarter 6 it was 3.5, and Quarter 12 it was 3.9.

Company A has hired you to evaluate InsurAHealth's claim that employees are sicker.
To facilitate your analysis, InsurAHealth has provided you with data for 12 quarters that includes 2,000 employees from Company A. Each quarter is a representative sample of the employees at Company A in that quarter. The demographic information included in this data is not part of InsurAHealth's health score calculation.

1.1 1. Understanding the data

1.1.1 import the data

```
In [853]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt
          import seaborn as sns
          %matplotlib inline
          np.random.seed(1337)
```

```
In [854]: data = pd.read_csv('acu_data.csv')
          data.head()
```

```
Out[854]:    Observation Number  Quarter  Employee Id  Sex (Male=1)  Race  Age  \
          0                   1        1            1           0.0   3.0   27
          1                   2        2            1           0.0   3.0   28
          2                   3        3            1           0.0   3.0   28
```

```
3                    4         4             1            0.0    3.0    28
4                    5         5             1            0.0    3.0    29

   Hospital Visit This Quarter (1=Yes)   Salary  Health Score
0                                    0  $36,907           3.7
1                                    0  $37,907           5.0
2                                    0  $38,907           4.0
3                                    0  $39,907           2.3
4                                    0  $40,907           2.1
```

```
In [855]: data.shape
```

```
Out[855]: (19103, 9)
```

There are 9 columns and 19103 rows in the dataset.

```
In [856]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19103 entries, 0 to 19102
Data columns (total 9 columns):
Observation Number                     19103 non-null int64
Quarter                                19103 non-null int64
Employee Id                            19103 non-null int64
Sex (Male=1)                           19032 non-null float64
Race                                   16980 non-null float64
Age                                    19103 non-null int64
Hospital Visit This Quarter (1=Yes)    19103 non-null int64
Salary                                 19103 non-null object
Health Score                           19103 non-null float64
dtypes: float64(3), int64(5), object(1)
memory usage: 1.3+ MB
```

We can see from here that there are many missing values as not all of them have values in all 19103 rows.

```
In [857]: data.groupby('Qsauarter')['Employee Id'].count()
```

```
Out[857]: Quarter
          1       684
          2       891
          3      1139
          4      1448
          5      1671
          6      1775
          7      1850
          8      1885
          9      1914
```

```
10     1934
11     1950
12     1962
Name: Employee Id, dtype: int64
```

For future analysis, we will now change the type of salary into integer.

1.1.2 Cleaning salary values

```
In [859]: data['Salary'] = data['Salary'].str[1:]   # remove the $
          data['Salary'] = data['Salary'].str.replace(',', '')  # remove the comma
          data['Salary'] = [int(i) for i in data['Salary']]  # convert them to integers
```

1.2 Convert the sex column to two dummy columns

```
In [860]: data['Sex (Male=1)'] = data['Sex (Male=1)'].map({1:'Male', 0:'Female'})

In [861]: data_sex = pd.get_dummies(data['Sex (Male=1)'])
          data = pd.concat([data, data_sex],axis=1)

In [862]: data.head()

Out[862]:    Observation Number  Quarter  Employee Id Sex (Male=1)  Race  Age  \
          0                   1        1            1       Female   3.0   27
          1                   2        2            1       Female   3.0   28
          2                   3        3            1       Female   3.0   28
          3                   4        4            1       Female   3.0   28
          4                   5        5            1       Female   3.0   29

             Hospital Visit This Quarter (1=Yes)  Salary  Health Score  Female  Male
          0                                    0   36907           3.7       1     0
          1                                    0   37907           5.0       1     0
          2                                    0   38907           4.0       1     0
          3                                    0   39907           2.3       1     0
          4                                    0   40907           2.1       1     0
```

1.2.1 Convert the Hospital visit to 2 dummy columns

```
In [863]: data['Hospital Visit This Quarter (1=Yes)'] = data['Hospital Visit This Quarter (1=Ye
          data_hosp = pd.get_dummies(data['Hospital Visit This Quarter (1=Yes)'])
          data = pd.concat([data, data_hosp],axis=1)

In [864]: data.head()

Out[864]:    Observation Number  Quarter  Employee Id Sex (Male=1)  Race  Age  \
          0                   1        1            1       Female   3.0   27
          1                   2        2            1       Female   3.0   28
          2                   3        3            1       Female   3.0   28
          3                   4        4            1       Female   3.0   28
```

```
4                    5        5            1       Female   3.0   29

  Hospital Visit This Quarter (1=Yes)  Salary  Health Score  Female  Male  \
0                             hosp_no   36907           3.7       1     0
1                             hosp_no   37907           5.0       1     0
2                             hosp_no   38907           4.0       1     0
3                             hosp_no   39907           2.3       1     0
4                             hosp_no   40907           2.1       1     0

   hosp_no  hosp_yes
0        1         0
1        1         0
2        1         0
3        1         0
4        1         0
```

Check the outliers in Age Column

```
In [866]: data.describe() #check overall outliers
```

```
Out[866]:        Observation Number       Quarter   Employee Id          Race  \
          count        19103.000000  19103.000000  19103.000000  16980.000000
          mean          9552.000000      7.342826    998.012249      1.597055
          std           5514.705432      3.166792    577.313902      0.739656
          min              1.000000      1.000000      1.000000      1.000000
          25%           4776.500000      5.000000    498.000000      1.000000
          50%           9552.000000      8.000000    996.000000      1.000000
          75%          14327.500000     10.000000   1498.000000      2.000000
          max          19103.000000     12.000000   2000.000000      3.000000

                         Age        Salary  Health Score        Female          Male  \
          count  19103.000000  19103.000000  19103.000000  19103.000000  19103.000000
          mean      30.592263  48297.612940      3.588379      0.491860      0.504423
          std        7.018862   5351.301686      1.985285      0.499947      0.499994
          min        7.000000  28351.000000      0.600000      0.000000      0.000000
          25%       26.000000  44550.500000      2.400000      0.000000      0.000000
          50%       29.000000  48196.000000      3.100000      0.000000      1.000000
          75%       32.000000  51958.500000      4.100000      1.000000      1.000000
          max      172.000000  68826.000000     10.000000      1.000000      1.000000

                     hosp_no      hosp_yes
          count  19103.000000  19103.000000
          mean       0.888552      0.111448
          std        0.314695      0.314695
          min        0.000000      0.000000
          25%        1.000000      0.000000
          50%        1.000000      0.000000
          75%        1.000000      0.000000
          max        1.000000      1.000000
```

In [867]: data["Age"].value_counts(ascending = True).sort_index(ascending = False)[:10]

```
Out[867]: 172      1
          171      4
          170      3
          72       1
          71       4
          70       4
          62       2
          61       4
          60       4
          59      14
          Name: Age, dtype: int64
```

In [868]: data["Age"].value_counts(ascending = True).sort_index(ascending = True)[:10]

```
Out[868]: 7        3
          8        4
          16       4
          17       7
          18       4
          19       4
          20       1
          22      20
          23     319
          24     915
          Name: Age, dtype: int64
```

We can see from the age group that there are some outliers. Say eighteen years old is the bar for working at this company, there are: 4 people 8 years old, 3 people 7 years old, 7 people 17 years old, 4 people 16 years old, 4 people 171 years old, 1 person 172 years old, 3 people 170 years old.

1.2.2 Check the outliers in health score data:

We do the same with health scores, as we saw in the describe that there were some scores > 6

In [869]: bad_health_scores = data[data['Health Score'] == 10]

In [870]: bad_health_scores.head()

```
Out[870]:      Observation Number  Quarter  Employee Id Sex (Male=1)  Race  Age  \
          77                   78        3            9         Male   1.0   29
          105                 106       12           11         Male   1.0   35
          107                 108        6           12         Male   2.0   32
          121                 122       11           13       Female   NaN   32
          137                 138        4           15         Male   1.0   24

              Hospital Visit This Quarter (1=Yes)  Salary  Health Score  Female  Male  \
          77                              hosp_no   50493          10.0       0     1
```

```
105                                 hosp_no   62588          10.0       0      1
107                                 hosp_no   43595          10.0       0      1
121                                 hosp_no   47246          10.0       1      0
137                                 hosp_no   52559          10.0       0      1

     hosp_no  hosp_yes
77         1         0
105        1         0
107        1         0
121        1         0
137        1         0
```

```
In [871]: data["Health Score"].value_counts(ascending = True).sort_index(ascending = False)[:6
```

```
Out[871]: 10.0    1238
          6.0       28
          Name: Health Score, dtype: int64
```

1.2.3 We start by dropping all Null values and removing obvious outliers

```
In [872]: #find the null values and count them
          data.isna().any()
```

```
Out[872]: Observation Number                       False
          Quarter                                  False
          Employee Id                              False
          Sex (Male=1)                              True
          Race                                      True
          Age                                      False
          Hospital Visit This Quarter (1=Yes)      False
          Salary                                   False
          Health Score                             False
          Female                                   False
          Male                                     False
          hosp_no                                  False
          hosp_yes                                 False
          dtype: bool
```

```
In [873]: data['Race'].isnull().sum()
```

```
Out[873]: 2123
```

```
In [874]: data['Age'].isnull().sum()
```

```
Out[874]: 0
```

```
In [875]: data['Sex (Male=1)'].isnull().sum()
```

```
Out[875]: 71
```

In [876]: data.dropna().shape

Out[876]: (16927, 13)

In [877]: data.dropna(inplace=True)

In [878]:
```
#remove the age that are older than 100 and younger than 18
age_mask = (data['Age'] >= 18) & (data['Age'] <=100)
#remove the health score higher than 6
health_score_mask = data['Health Score'] <= 6.0
```

In [879]:
```
data = data[age_mask & health_score_mask]
```

In [880]:
```
data.describe() #check some outliers
```

Out[880]:

	Observation Number	Quarter	Employee Id	Race
count	15867.000000	15867.000000	15867.000000	15867.000000
mean	9533.825046	7.331317	996.031008	1.599546
std	5453.117067	3.159958	570.788420	0.740055
min	1.000000	1.000000	1.000000	1.000000
25%	4893.500000	5.000000	510.000000	1.000000
50%	9572.000000	8.000000	997.000000	1.000000
75%	14220.500000	10.000000	1487.000000	2.000000
max	19031.000000	12.000000	1993.000000	3.000000

	Age	Salary	Health Score	Female	Male
count	15867.000000	15867.000000	15867.000000	15867.000000	15867.000000
mean	30.436756	48396.840108	3.148598	0.494044	0.505956
std	6.263306	5375.858441	1.080626	0.499980	0.499980
min	18.000000	28351.000000	0.600000	0.000000	0.000000
25%	26.000000	44628.000000	2.300000	0.000000	0.000000
50%	29.000000	48319.000000	3.000000	0.000000	1.000000
75%	32.000000	52089.000000	3.900000	1.000000	1.000000
max	72.000000	68826.000000	6.000000	1.000000	1.000000

	hosp_no	hosp_yes
count	15867.000000	15867.000000
mean	0.891851	0.108149
std	0.310578	0.310578
min	0.000000	0.000000
25%	1.000000	0.000000
50%	1.000000	0.000000
75%	1.000000	0.000000
max	1.000000	1.000000

Check the race data outliers

In [881]:
```
data["Race"].value_counts(ascending = True)
```

```
Out[881]: 3.0    2440
          2.0    4633
          1.0    8794
          Name: Race, dtype: int64
```

We can see 1.0 race is the majority in the company.

```
In [882]: pd.crosstab(data['Race'], data['Sex (Male=1)'])
```

```
Out[882]: Sex (Male=1)  Female  Male
          Race
          1.0             4420  4374
          2.0             2263  2370
          3.0             1156  1284
```

We can see in each race group, there are more male employees than female.

1.2.4 Data Preprocessing:

Outliers and missing values:

Health Score data: there are 1238 records(6.48% of total obs) that are denoted as 10, which are out of the reasonable range 0-6.

Age data: There are 26 people having unreasonable ages out of the reasonabe age 18-100: 3 people age seven, 4 people age eight, 4 people age sixteen, 7 people age seventeen, 1 person age 172, 4 people age 171, 3 people age 170. So I removed these erroneous ages.

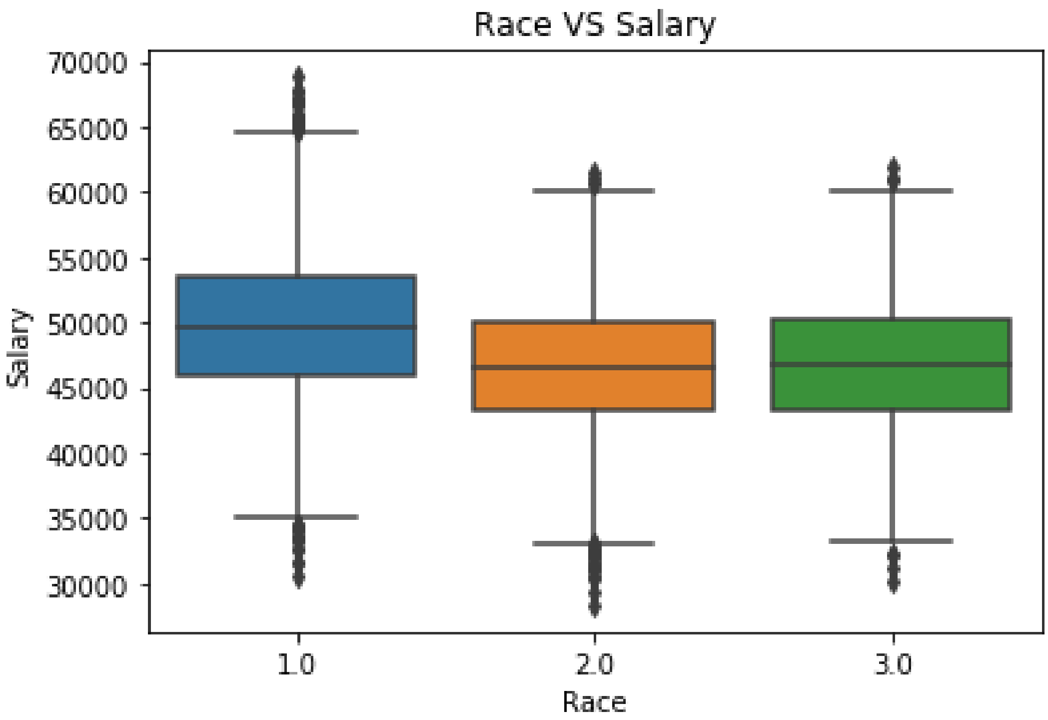
As for Sex data, I found 71 missing values(0.37% of total obs) and 2123 missing values in Race data(11.11% of total obs).

I removed all the outliers and missing data values. All the following charts and analysis stemmed from the modified dataset.

1.2.5 Understanding the employees' characteristics

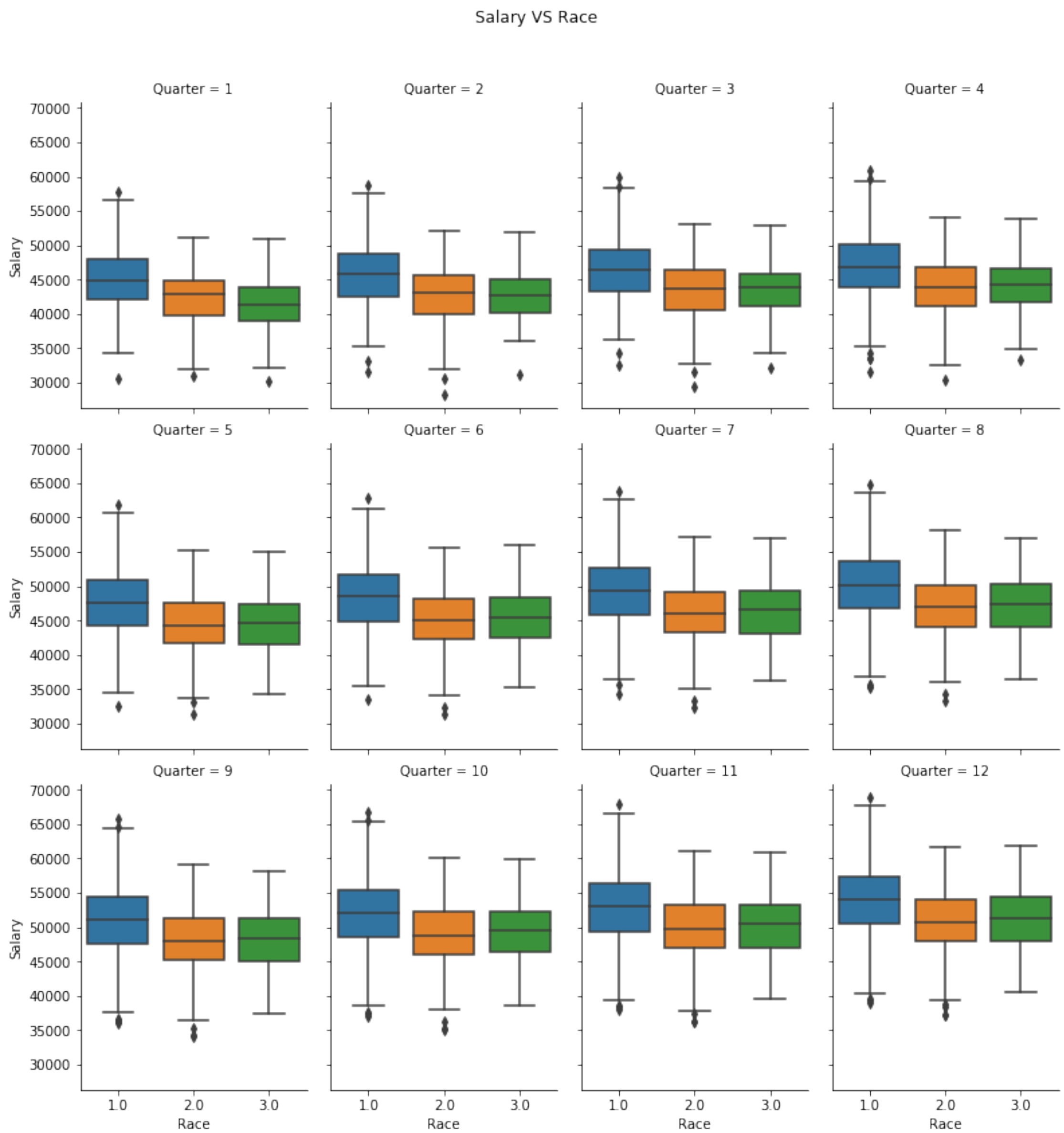
Salary VS Race

```
In [885]: sns.boxplot(y='Salary', x= 'Race', data=data).set_title('Race VS Salary')
          plt.figure(figsize=(10,5))
          plt.show();
```



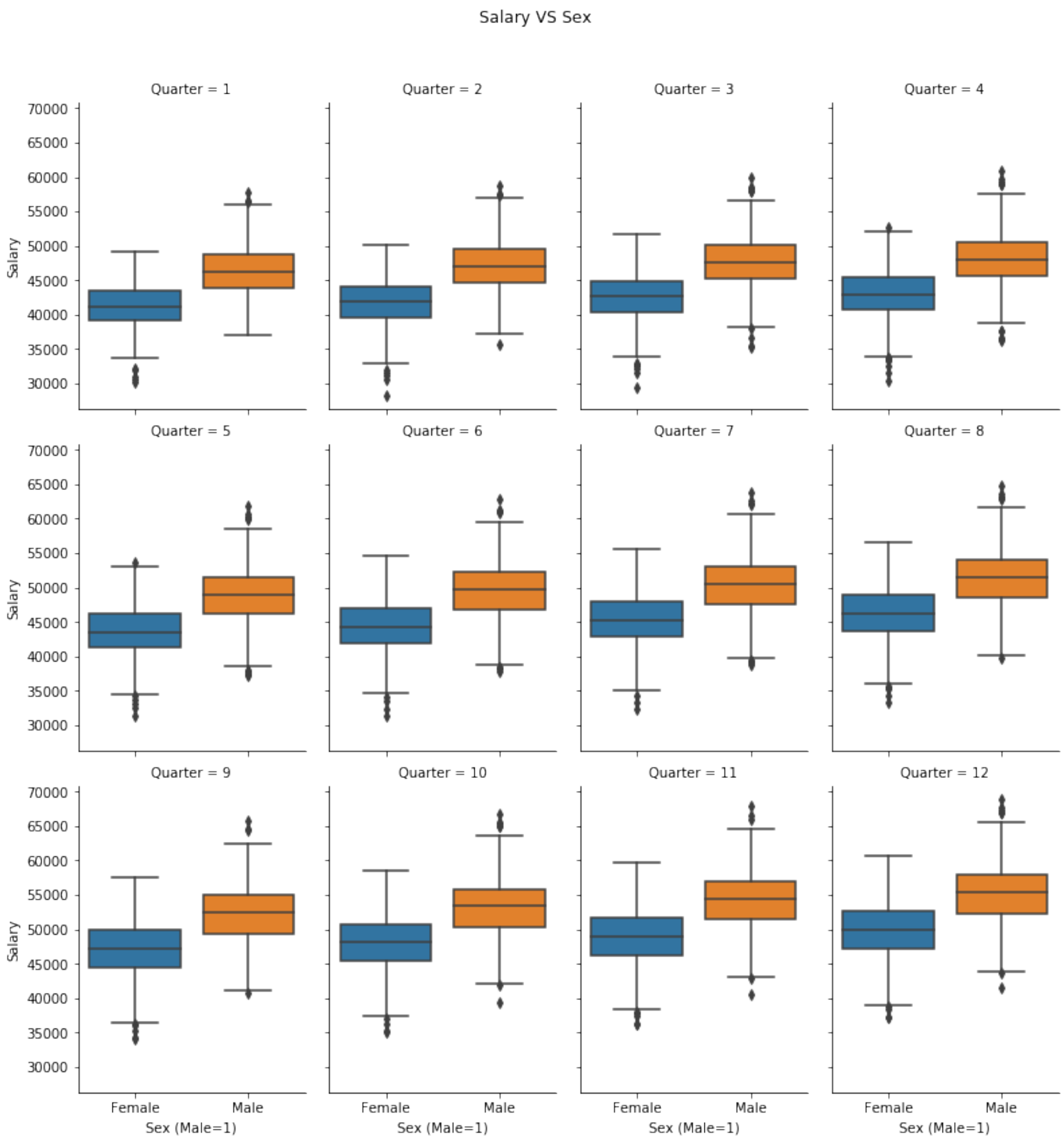

```
<Figure size 720x360 with 0 Axes>
```

```
In [921]: g = sns.catplot(y='Salary', x= 'Race', data=data,col = 'Quarter', kind='box',col_wrap
          g.fig.suptitle('Salary VS Race')
          g.fig.subplots_adjust(top=.9)
```



1.2.6 Salary VS Sex

In [920]:
```
g = sns.catplot(y='Salary', x= 'Sex (Male=1)', data=data,col = 'Quarter', kind='box'
g.fig.suptitle('Salary VS Sex')
g.fig.subplots_adjust(top=.9)
```




```
In [886]: sns.boxplot(y='Salary', x= 'Sex (Male=1)', data=data).set_title('Salary VS Sex')
          plt.figure(figsize=(10,5))
          #plt.title('Race VS Salary')
          plt.show();
```


```
<Figure size 720x360 with 0 Axes>
```

Salary: we can see that male employees are on the higher end of salary distribution, while female are in the lower spectrum. Male has a much higher median value insalary than women. And we can see that 1.0 race is on the higher end of salary distribution than the other two races.

1.2.7 Do demographic factors change over time?

In [914]:
```
result = data.groupby(['Quarter','Sex (Male=1)']).agg({'Quarter':np.size,'Age':np.ave
result.rename(columns={'Age': 'avg_age', 'Salary': 'avg_salary', 'Quarter': 'Obs'}, i
result
```

Out[914]:

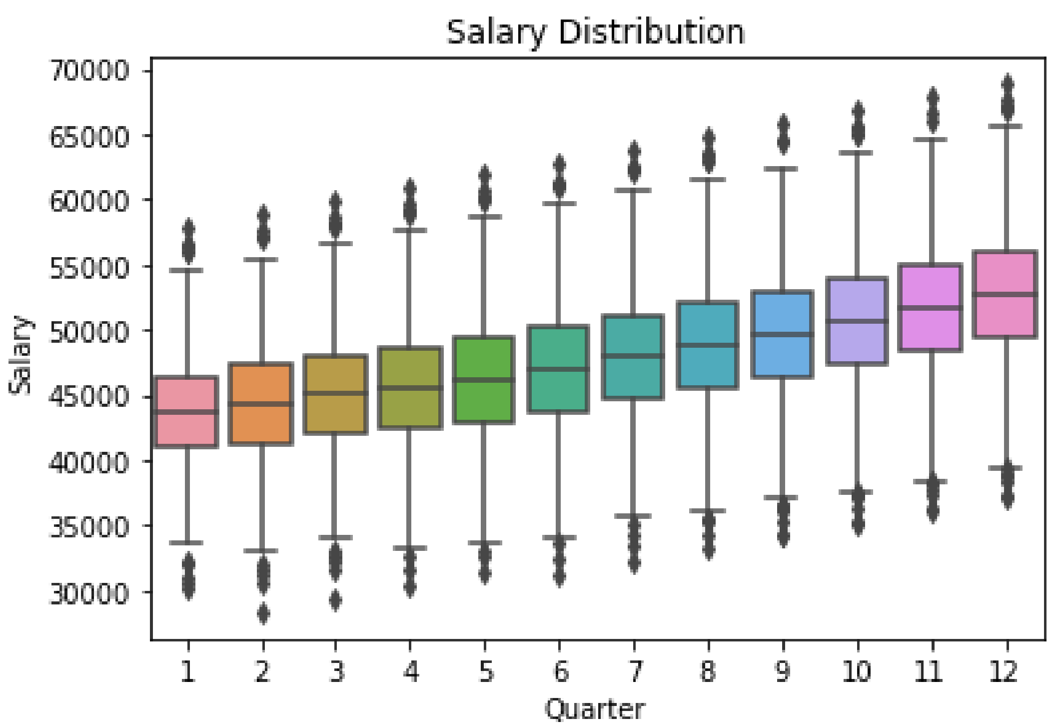
Quarter	Sex (Male=1)	Obs	avg_age	avg_salary
1	Female	287	28.665505	41156.839721
	Male	274	28.715328	46412.310219
2	Female	379	28.306069	41766.229551
	Male	357	28.708683	47081.521008
3	Female	475	28.667368	42567.726316
	Male	476	28.590336	47661.571429
4	Female	604	28.701987	43050.943709
	Male	612	28.772876	48071.101307
5	Female	693	29.057720	43604.213564

```
         Male          708   29.269774   48804.081921
6        Female        730   29.838356   44414.619178
         Male          751   29.684421   49565.834887
7        Female        747   30.283802   45334.453815
         Male          801   30.491885   50353.918851
8        Female        765   30.768627   46192.368627
         Male          799   30.739675   51283.957447
9        Female        779   31.369705   47077.189987
         Male          799   31.118899   52175.344180
10       Female        783   31.578544   47993.574713
         Male          822   31.515815   53120.951338
11       Female        810   32.230864   48869.516049
         Male          817   31.872705   54198.216646
12       Female        787   32.481576   49853.050826
         Male          812   31.987685   55152.822660
```

The number of employees in the company is increasing over time. And there is a balanced ratio between male and female employees.

1.2.8 Salary Distribution

In [888]:
```
sns.boxplot(y='Salary', x= 'Quarter', data=data).set_title('Salary Distribution')
plt.figure(figsize=(10,5))
plt.show();
```

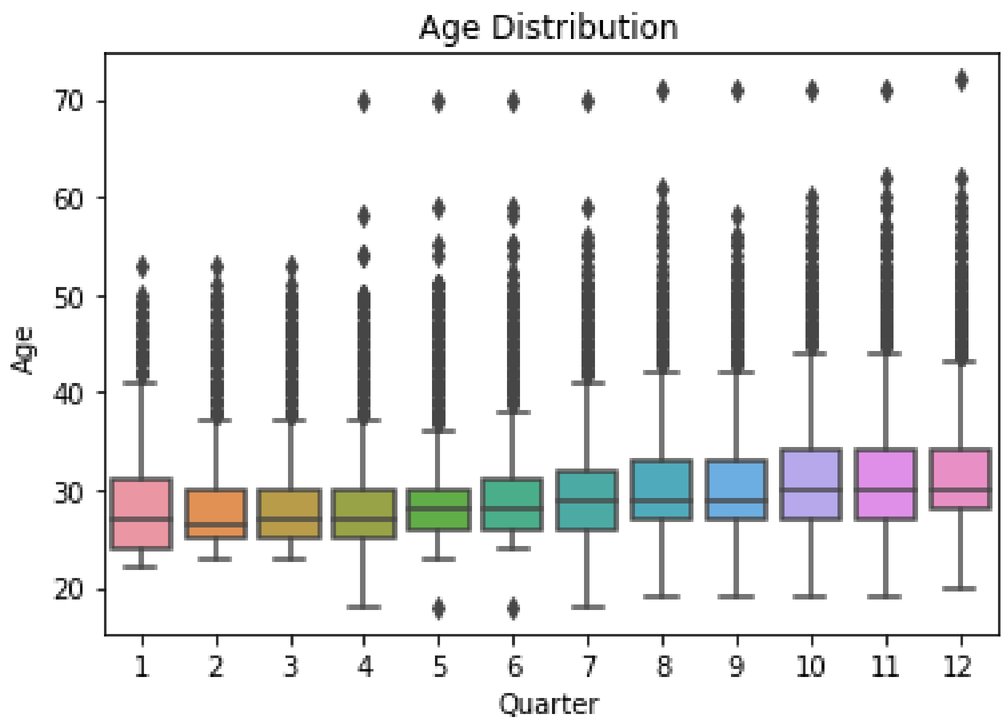


```
<Figure size 720x360 with 0 Axes>
```

The employees' salary is centered around \$42000 ~ \$52000 . Additionally, their salaries are increasing over time.

1.2.9 Age Distribution

```
In [889]: sns.boxplot(y='Age', x= 'Quarter', data=data).set_title('Age Distribution')
          plt.figure(figsize=(10,5))
          plt.show();
```




```
<Figure size 720x360 with 0 Axes>
```

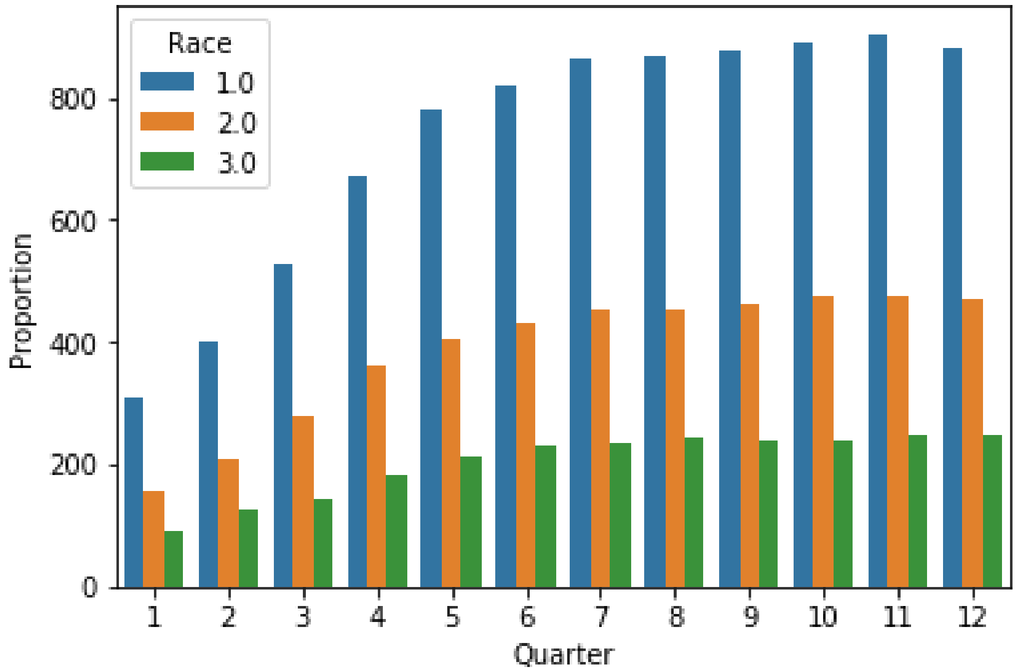
Employees in the company are young, their age concentrate in the range 28-32. And we can see that all the employees are aging over time,

1.2.10 Race change over time

```
In [919]: df = data
          x, y, hue = "Quarter", "Proportion", "Race"
          hue_order = ["1.0", "2.0","3.0"]
```

```
(df[x]
 .groupby(df[hue])
 .value_counts()
 .rename(y)
 .reset_index()
 .pipe((sns.barplot, "data"), x=x, y=y, hue=hue))
```

Out[919]: <matplotlib.axes._subplots.AxesSubplot at 0x1c3086c470>



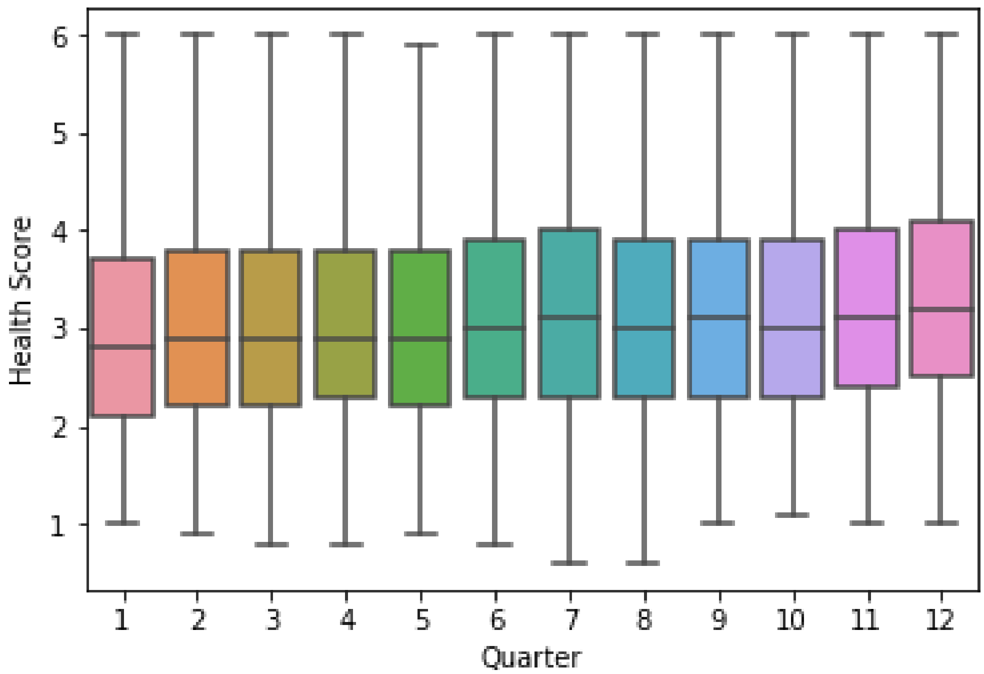
The 1.0 race has the largest proportion in the company. But all races are increasing over time.

1.3 Explore Relationships

Which characteristics are associated with the health score?

1.3.1 Health Score VS Quarter

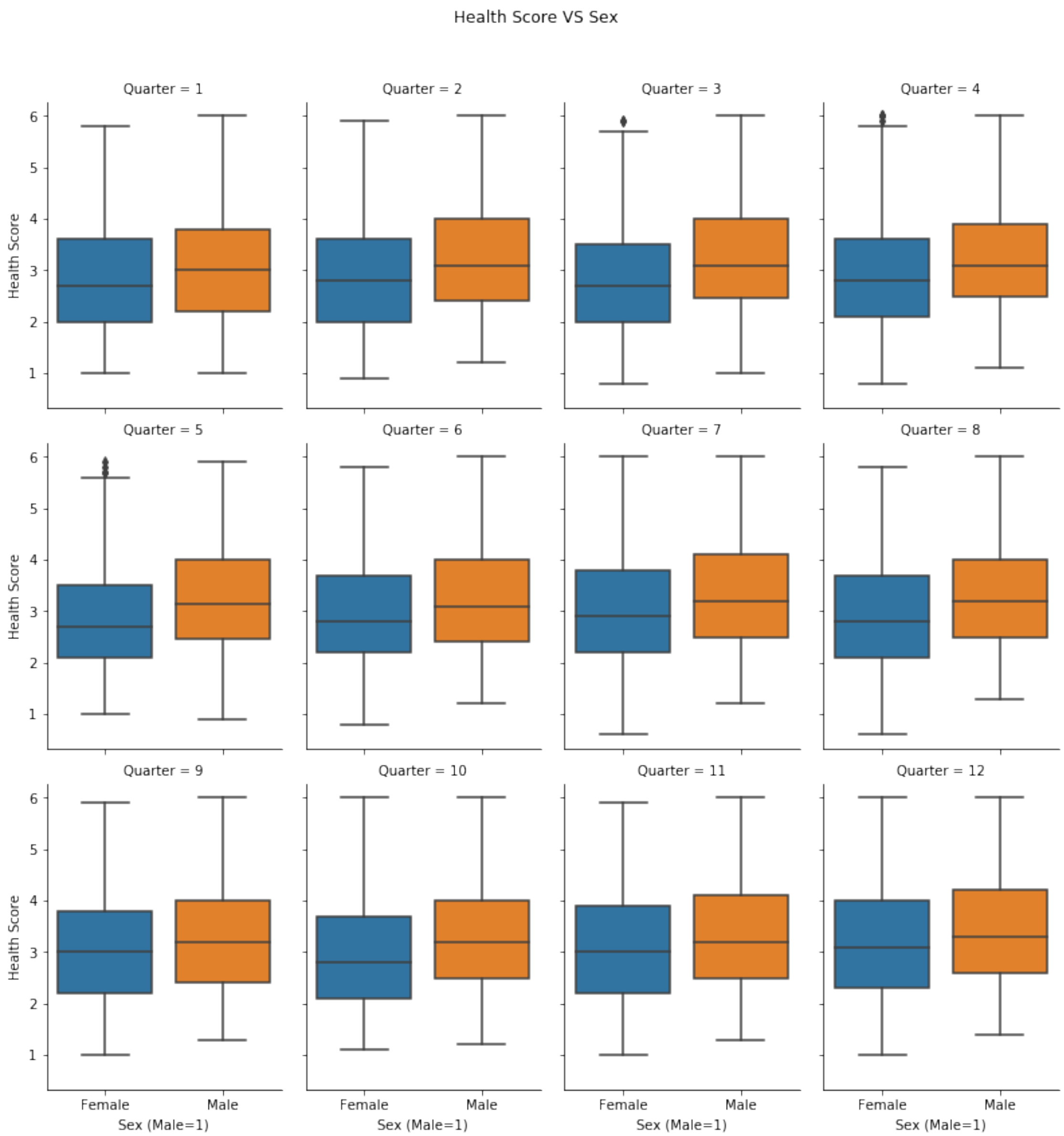
In [893]:
```
sns.boxplot(y='Health Score', x= 'Quarter', data=data)
plt.figure(figsize=(10,5))
plt.show();
```




```
<Figure size 720x360 with 0 Axes>
```

1.3.2 Health Score VS Sex

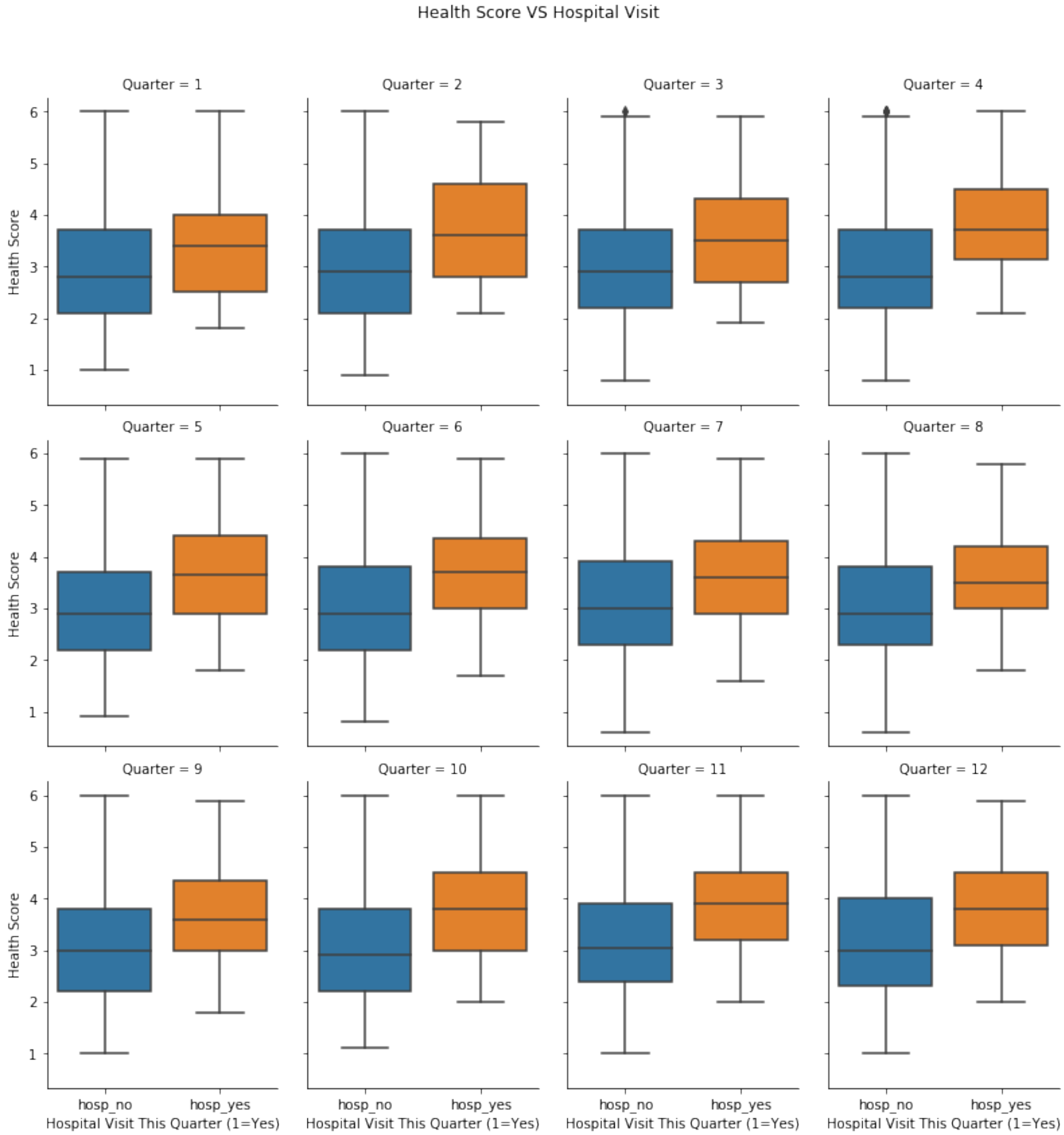
```
In [894]: g = sns.catplot(y='Health Score', x= 'Sex (Male=1)', data=data,col = 'Quarter', kind=
          g.fig.suptitle('Health Score VS Sex')
          g.fig.subplots_adjust(top=.9)
```



Male employees have higher health scores than female employees

1.3.3 Health Score VS Hospital Visit

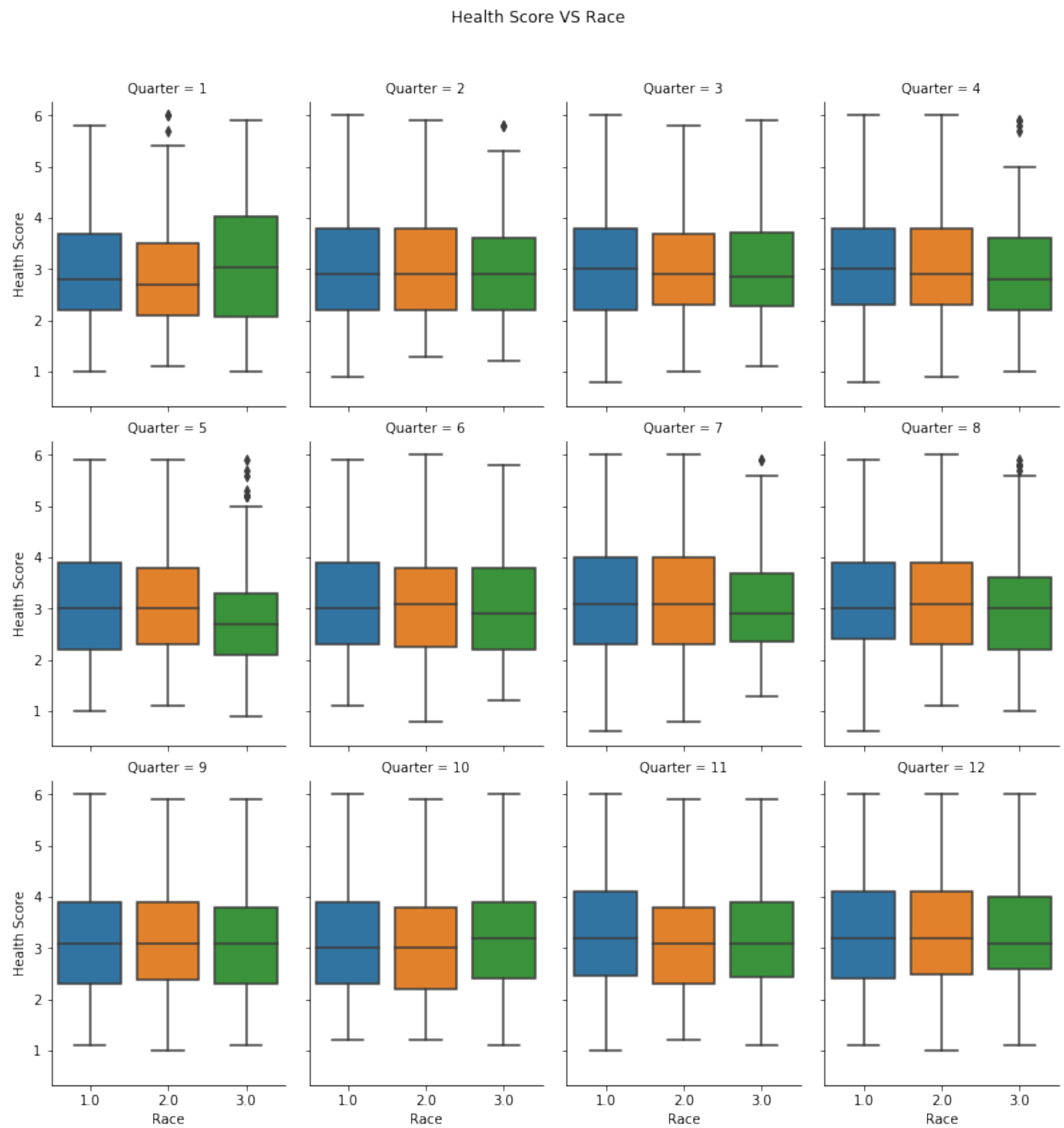
```
In [895]: g = sns.catplot(y='Health Score', x= 'Hospital Visit This Quarter (1=Yes)', data=data
          g.fig.suptitle('Health Score VS Hospital Visit')
          g.fig.subplots_adjust(top=.9)
```



We can see employees who visit hospital have higher health scores.

1.3.4 Health Score VS Race

```
In [897]: g = sns.catplot(y='Health Score', x= 'Race', data=data,col = 'Quarter', kind='box',co
          g.fig.suptitle('Health Score VS Race')
          g.fig.subplots_adjust(top=.9)
```

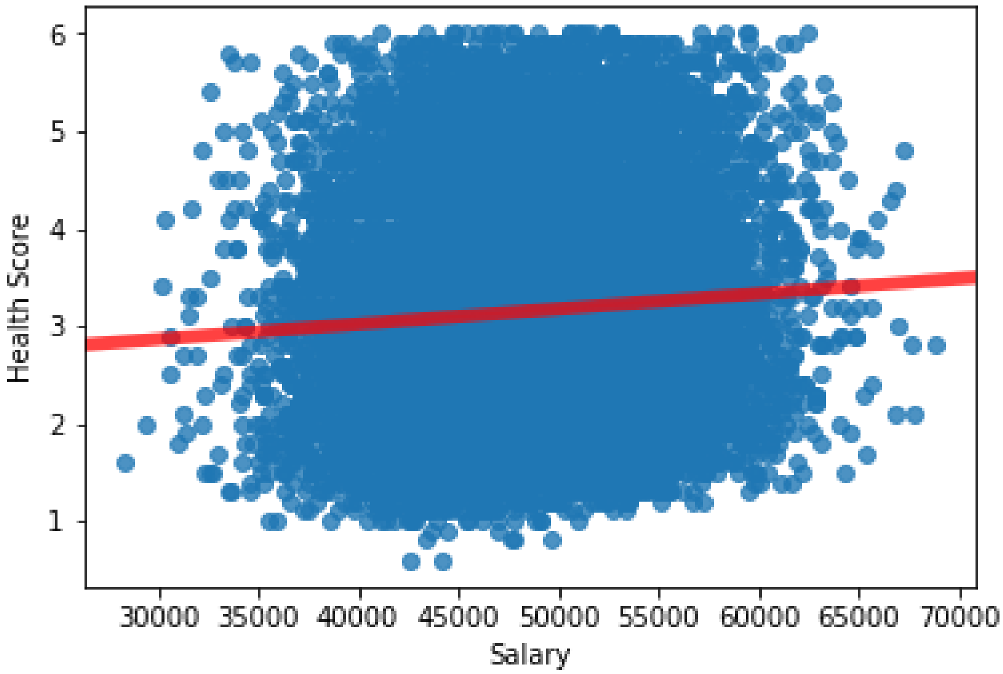


There is no relationship between health score and race.

1.3.5 Health Score VS Salary

```
In [899]: sns.regplot(x='Salary', y='Health Score',data=data,line_kws={"color":"r","alpha":0.7

Out[899]: <matplotlib.axes._subplots.AxesSubplot at 0x1c30a22278>
```



Employees with higher salaries have higher health scores.

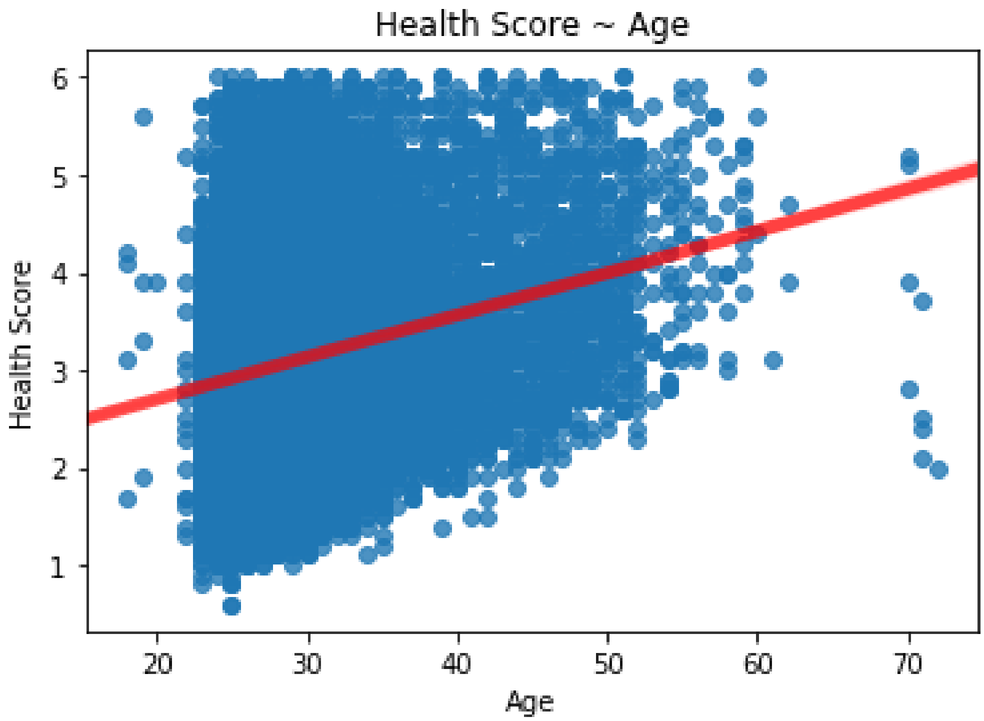
1.3.6 Health Score VS Age

It is well known that people's health deteriorates as they get older, so here is simple linear model health score ~ age

```
In [900]: import statsmodels.api as sm
```

```
In [901]: sns.regplot(x='Age', y='Health Score',data=data,line_kws={"color":"r","alpha":0.7,"lw
```

```
Out[901]: Text(0.5, 1.0, 'Health Score ~ Age')
```




```
In [902]: y = data['Health Score']
          X = data['Age']

          # Note the difference in argument order
          model = sm.OLS(y, X).fit()
          predictions = model.predict(X) # make the predictions by the model

          # Print out the statistics
          model.summary()

Out[902]: <class 'statsmodels.iolib.summary.Summary'>
          """
                                      OLS Regression Results
          ==============================================================================
          Dep. Variable:           Health Score   R-squared:                       0.889
          Model:                            OLS   Adj. R-squared:                  0.889
          Method:                 Least Squares   F-statistic:                 1.269e+05
          Date:                Thu, 11 Apr 2019   Prob (F-statistic):               0.00
          Time:                        13:36:48   Log-Likelihood:                -24167.
          No. Observations:               15867   AIC:                         4.834e+04
          Df Residuals:                   15866   BIC:                         4.834e+04
          Df Model:                           1
          Covariance Type:            nonrobust
          ==============================================================================
```

```
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Age            0.1010      0.000    356.203      0.000       0.100       0.102
==============================================================================
Omnibus:                      419.848   Durbin-Watson:                   1.765
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              445.256
Skew:                           0.400   Prob(JB):                     2.06e-97
Kurtosis:                       2.819   Cond. No.                         1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
"""
```

Employees who are older have higher health scores than younger employees. As I delve deeper, I found that older age is also correlated with higher salaries. Because male is also positively correlated with high salaries, I think older age and being male are factors behind the salary and heath correlation.

Based on the insights from the findings, I would assume that male, hospital visit and age will lead to higher health scores.

1.4 Evaluating the Claim

Using the information from Questions 1 and 2, describe how you would evaluate InsurAHealth's claim that employees are getting sicker.

I would like to know if health score is a reliable measurement of employees' actual health conditions.There are two approaches I would take to examine this 1. As new employees come to the companuy each quarter, the employee id records vary in quarter time.These new employees might be the reason why health scores are high.I select only the employees who have been working here for 12 quarters to examine if any external factors have any effect on their health conditions, as the new employees may drive up the health scores.

1.Select employees who have been working here for 12 quarters

In [903]:
```python
#Find out the employee Id which has 12 quarter data records
data_list = data.loc[data['Quarter'] == 1]
data_emlist = data_list['Employee Id'].to_list()

data_12q = data.loc[data['Employee Id'].isin(data_emlist)]
data_12q.head()
```

Out[903]:

	Observation Number	Quarter	Employee Id	Sex (Male=1)	Race	Age	\
0	1	1	1	Female	3.0	27	
1	2	2	1	Female	3.0	28	
2	3	3	1	Female	3.0	28	
3	4	4	1	Female	3.0	28	
4	5	5	1	Female	3.0	29	

```
  Hospital Visit This Quarter (1=Yes)  Salary  Health Score  Female  Male  \
0                             hosp_no   36907           3.7       1     0
1                             hosp_no   37907           5.0       1     0
2                             hosp_no   38907           4.0       1     0
3                             hosp_no   39907           2.3       1     0
4                             hosp_no   40907           2.1       1     0

   hosp_no  hosp_yes
0        1         0
1        1         0
2        1         0
3        1         0
4        1         0
```

Calculate the percentage of people visiting hospital

In [904]:
```
data_hosp = data_12q.groupby(['Quarter', 'Hospital Visit This Quarter (1=Yes)']).cou
data_hosp.head()
data_hosp.shape
#data_hosp
```

Out[904]: (24, 11)

In [905]:
```
data_hosp = data_hosp.pivot_table(index='Quarter',columns= 'Hospital Visit This Quar
```

In [906]:
```
data_hosp = data_hosp['Employee Id']
data_hosp.head()
```

Out[906]:
```
Hospital Visit This Quarter (1=Yes)  hosp_no  hosp_yes
Quarter
1                                        520        41
2                                        478        47
3                                        469        59
4                                        471        55
5                                        482        48
```

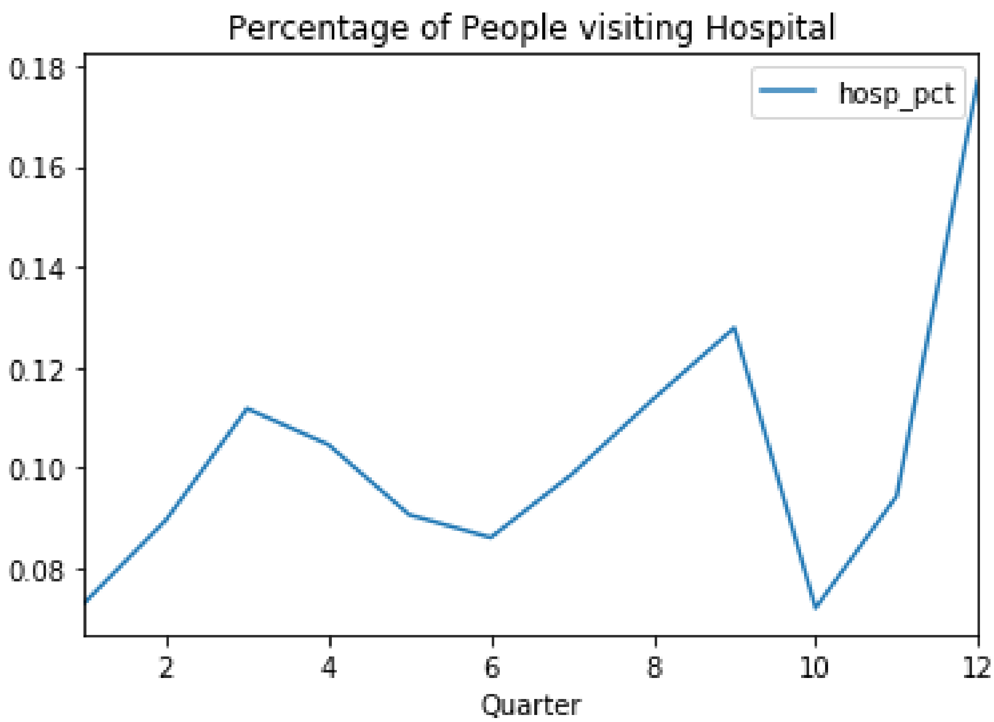
In [907]:
```
data_hosp['hosp_pct'] =data_hosp['hosp_yes']/(data_hosp['hosp_no'] + data_hosp['hosp_
data_hosp = data_hosp.reset_index()
data_hosp
```

Out[907]:
```
Hospital Visit This Quarter (1=Yes)  Quarter  hosp_no  hosp_yes  hosp_pct
0                                          1      520        41  0.073084
1                                          2      478        47  0.089524
2                                          3      469        59  0.111742
3                                          4      471        55  0.104563
4                                          5      482        48  0.090566
5                                          6      478        45  0.086042
6                                          7      475        52  0.098672
7                                          8      468        60  0.113636
```

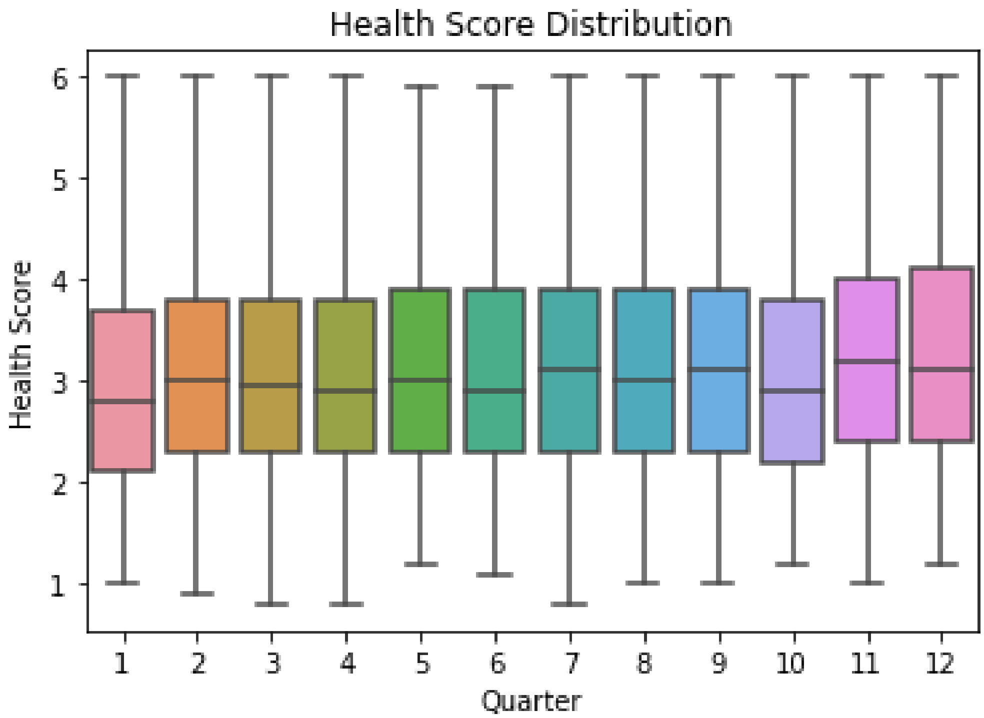
8	9	457	67	0.127863
9	10	490	38	0.071970
10	11	480	50	0.094340
11	12	427	92	0.177264

In [909]:
```
data_hosp.plot(x='Quarter',y='hosp_pct',title='Percentage of People visiting Hospital
```

Out[909]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1c3224ab70>
```



In [910]:
```
sns.boxplot(y='Health Score', x= 'Quarter', data=data_12q).set_title('Health Score D
plt.figure(figsize=(10,5))
plt.show();
```



<Figure size 720x360 with 0 Axes>

So I first plotted the percentage of people visiting hospital in each quarter and I found that the number of people going to hospitals vary in quarters. To validate this assumption, I used the health score distribution in each quarter to look into this. However, this doesn't support my previous assumption. But I think we could dig deeper and see if some interesting results can be found.

2. As we learn from the findings, going to hospital also relates to higher health scores. We might be led to assume that going to hospital indicates employee's poor health. But this is not necessarily the case, the employees might care a lot about their health. They might go to the hospital for more frequent checkups, thus more medical documents are generated, which I assume would be related to health scores. In this case, going to hospital does get us to higher health scores(from the findings), but it doesn't necessarily relate to poorer health. Although the formula for developing the health score is not public, I would raise a question to the correlation between going to hospital and higher health scores.