Problem 3:

(P7.38, textbook) Refer to the SENIC data set in Appendix C.1. For predicting the average length of stay of patients in a hospital (Y), it has been decided to include age ($X1$) and infection risk ($X2$) as predictor variables. The question now is whether an additional predictor variable would be helpful in the model and, if so, which variable would be most helpful. Assume that a first-order multiple regression model is appropriat

In [1]:
```
import pandas as pd, numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df1 = pd.read_csv('APPENC01.txt', sep = '\s+', header =None)
df1.head()
```

Out[2]:

	0	1	2	3	4	5	6	7	8	9	10	11
0	1	7.13	55.7	4.1	9.0	39.6	279	2	4	207	241	60.0
1	2	8.82	58.2	1.6	3.8	51.7	80	2	2	51	52	40.0
2	3	8.34	56.9	2.7	8.1	74.0	107	2	3	82	54	20.0
3	4	8.95	53.7	5.6	18.9	122.8	147	2	4	53	148	40.0
4	5	11.20	56.5	5.7	34.5	88.9	180	2	1	134	151	40.0

In [3]:
```
y = df1[1]
x1 = df1[2]
x2 = df1[3]
x3 = df1[4]
x4 = df1[9]
x5 = df1[10]
x6 = df1[11]
df = pd.DataFrame({'Y':y,'X1':x1,'X2':x2,'X3':x3,'X4':x4,'X5':x5,'X6':x6})
df.head()
```

Out[3]:

	Y	X1	X2	X3	X4	X5	X6
0	7.13	55.7	4.1	9.0	207	241	60.0
1	8.82	58.2	1.6	3.8	51	52	40.0
2	8.34	56.9	2.7	8.1	82	54	20.0
3	8.95	53.7	5.6	18.9	53	148	40.0
4	11.20	56.5	5.7	34.5	134	151	40.0

In [4]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model12 = smf.ols('y ~ x1+x2', data=df)
results12 = model12.fit()
```

```
sse12 = np.sum((results12.fittedvalues - df.Y)**2)
ssr12 = np.sum((results12.fittedvalues - df.Y.mean())**2)
```

In [5]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model123 = smf.ols('y ~ x1+x2+x3', data=df)
results123 = model123.fit()
sse123 = np.sum((results123.fittedvalues - df.Y)**2)
ssr123 = np.sum((results123.fittedvalues - df.Y.mean())**2)
```

In [6]:
```
ssr3_12 = sse12 - sse123
```

In [7]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model124 = smf.ols('y ~ x1+x2+x4', data=df)
results124 = model124.fit()
sse124 = np.sum((results124.fittedvalues - df.Y)**2)
ssr124 = np.sum((results124.fittedvalues - df.Y.mean())**2)
```

In [8]:
```
ssr4_12 = sse12 - sse124
```

In [9]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model125 = smf.ols('y ~ x1+x2+x5', data=df)
results125 = model125.fit()
sse125 = np.sum((results125.fittedvalues - df.Y)**2)
ssr125 = np.sum((results125.fittedvalues - df.Y.mean())**2)
```

In [10]:
```
ssr5_12 = sse12 - sse125
```

In [11]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
model126 = smf.ols('y ~ x1+x2+x6', data=df)
results126 = model126.fit()
sse126 = np.sum((results126.fittedvalues - df.Y)**2)
ssr126 = np.sum((results126.fittedvalues - df.Y.mean())**2)
```

In [12]:
```
ssr6_12 = sse12 - sse126
```

a. For each of the following variables, calculate the coefficient of partial determination given that $X1$ and $X2$ are included in the model: routine culturing ratio ($X3$), average daily census ($X4$), number of nurses ($X5$), and available facilities and services ($X6$).

In [13]:
```
R3_12 = ssr3_12 / sse12
print(R3_12)
```

```
0.011672927814615352
```

R^2 Y3|12 : the error sum of squares for the model containing both X1 and X2 (SSE(X1,X2)) is only reduced by 1.16 percent when X3 is added to the model.

In [14]:
```
R4_12 = ssr4_12 / sse12
print(R4_12)
```

```
0.13620333847831456
```

R^2 Y4|12 : when X4 is added to the regression model containing X1 and X2 here, the error sum of squares SSE(X1,X2) is reduced by 13.6 percent.

In [15]:
```
R5_12 = ssr5_12 / sse12
print(R5_12)
```

```
0.03736634595438007
```

R^2 Y5|12 : when X5 is added to the regression model containing X1 and X2 here, the error sum of squares SSE(X1,X2) is reduced by only 3.7 percent.

In [16]:
```
R6_12 = ssr6_12 / sse12
print(R6_12)
```

```
0.03638879218961961
```

R^2 Y6|12 : the error sum of squares for the model containing both X1 and X2 (SSE(X1,X2)) is only reduced by 3.6 percent when X6 is added to the model.

b. On the basis of the results in part (a), which of the four additional predictor variables is best? Is the extra sum of squares associated with this variable larger than those for the other three variables?

- **The additional predictor X4 (average daily census) is the best because the error sum of squares for the model containing both X1 and X2 could be reduced by 13.6 percent when X4 is added to the model. Meanwhile, adding X3 the error sum of squares for the model containing both X1 and X2 could be only reduced by 1.16 percent, adding X5 it could be only reduced by 3.7 percent, and adding X6 it could be only reduced by 3.6 percent**

In [21]:
```
print('SSR(X3|X1,X2)=',ssr3_12)
print('SSR(X4|X1,X2)=',ssr4_12)
print('SSR(X5|X1,X2)=',ssr5_12)
print('SSR(X6|X1,X2)=',ssr6_12)
```

```
SSR(X3|X1,X2)= 3.2479971689028844
SSR(X4|X1,X2)= 37.89863732548616
SSR(X5|X1,X2)= 10.397201781725528
SSR(X6|X1,X2)= 10.125197027578338
```

- **The extra sum of squares associated with X4 (SSR(X4|X1,X2) = 37.9) is larger than those for the other three variables: X3 (SSR(X3|X1,X2) = 3.2), X5 (SSR(X5|X1,X2)), X6 (SSR(X6|X1,X2))**

c. Using the F *test statistic, test whether or not the variable determined to be best in part (b) is helpful in the regression model when $X1$ and $X2$ are included in the model; use α = .05.*

State the alternatives, decision rule, and conclusion. Would the F test statistics for the other three potential predictor variables be as large as the one here? Discuss.

The alternatives

- H0: β2 = 0
- Ha: β2 # 0

The decision rule

- If F* ≤ F(1−α; 1 , n-p), conclude H0
- If F* > F(1−α; 1 , n-p), conclude Ha

In [24]:
```
n = len(y)
p = 4
Fstar = (ssr4_12/1 / (sse124/(n-p)))
print('F*=',Fstar)
import scipy.stats as stats
f = stats.f.ppf(q=1-0.05,dfn=1,dfd=n-4)
print('F=',f)
```

```
F*= 17.187104969800327
F= 3.9281951303723233
```

- **For α = 0.01, we have F = 3.9. Since F* = 17.2 > 3.9, we conclude Ha that X4 cannot be dropped from the regression model that already contains both X1 and X2. Means that the variable determined to be best in part (b) (X4) could be useful in the regression model when X1 and X2 are included in the model**

In [28]:
```
n = len(y)
p = 4
Fstar = (ssr3_12/1 / (sse123/(n-p)))
print('F*=',Fstar)
```

```
F*= 1.2873765857487445
```

In [29]:
```
n = len(y)
p = 4
Fstar = (ssr5_12/1 / (sse125/(n-p)))
print('F*=',Fstar)
```

```
F*= 4.231029833530429
```

In [30]:
```
n = len(y)
p = 4
Fstar = (ssr6_12/1 / (sse126/(n-p)))
print('F*=',Fstar)
```

```
F*= 4.116160456125622
```

- **The F *test statistics for the other three potential predictor variables (X3, X5, and X6) wouldn't be as large as the F* statistics for the potential predictor variables X4 because the extra sum of squares associated with X4 (SSR(X4|X1,X2) = 37.9) is larger than**

those for the other three variables: X3 (SSR(X3|X1,X2) = 3.2), X5 (SSR(X5|X1,X2)), X6 (SSR(X6|X1,X2))