STAT303-2 Assignment 2 Complete v1

January 18, 2022

1 STAT303-2: Assignment 2

1.1 Instructions:

a. You may talk to a friend, discuss the questions and potential directions for solving them. However, you need to write your own solutions and code separately, and not as a group activity.

b. Do not write your name on the assignment. (1 point)

c. Export your Jupyter notebook as a PDF file. If you get an error, make sure you have downloaded the MikTex software (for windows) or MacTex (for mac). Note that after installing MikTex/MacTex, you will need to check for updates, install the updates if needed, and re-start your system. Submit the PDF file. (1 point)

d. Please include each question (or question number) followed by code and your answer (if applicable). Write your code in the 'Code' cells and your answer in the 'Markdown' cells of the Jupyter notebook. Ensure that the solution is well-organized, clear, and concise (3 points)

1. It's easy enough to identify different sections of the homework assignment (e.g., if there are different sections of an assignment, they're clearly distinguishable by section headers or the like)

2. It's clear which code/markdown blocks correspond to which questions.
3. There aren't excessively long outputs of extraneous information (e.g., no printouts of entire data frames without good reason)

This assignment is **due at 11:59pm on Wednesday, January 26th**. Good luck!

Submissions will be graded with a maximum of **43 points** – 38 points for code & answers, 5 points for anonymity and proper formatting.

1.2 Part 1

The dataset *infmort.csv* gives the infant mortality rates of different countries in the world.

This part is worth 17 points overall.

(1a) Make one plot including the following subplots: (i) a boxplot of mortality against region, (ii) a boxplot of income against region, and (iii) a scatter plot of mortality against income. Be sure to include appropriate axis labels.

What trends do you see in these plots? Mention the trend separately for each subplot.

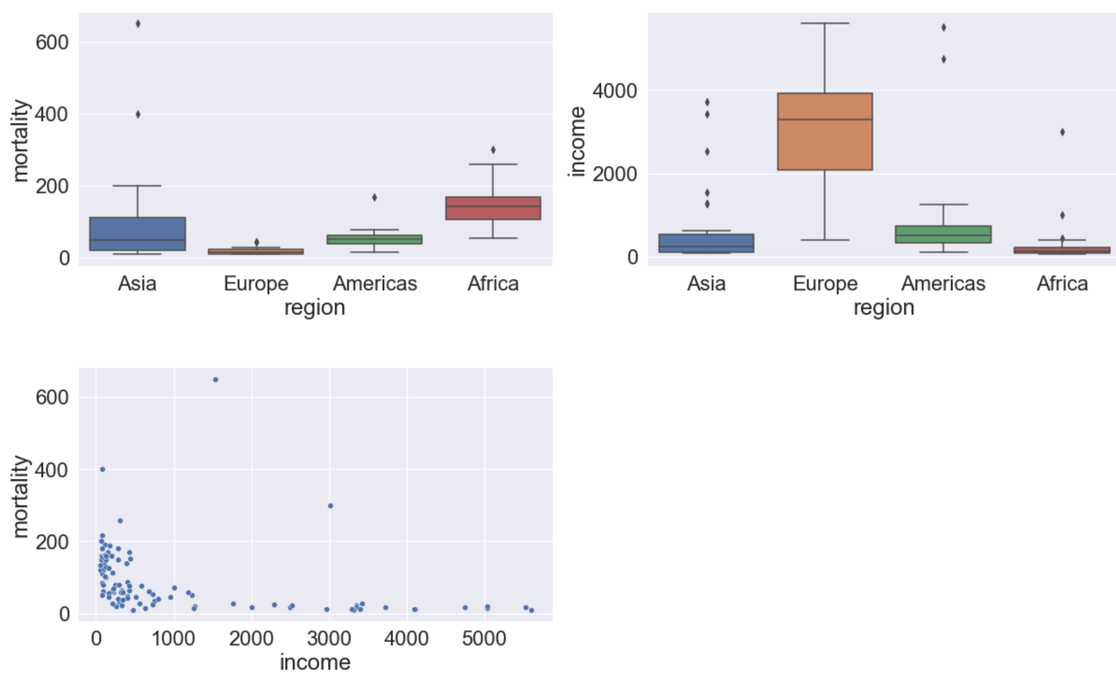
(3 points for code and plots, 2 points for written answers)

[428]:
```
import pandas as pd
import seaborn as sns
import statsmodels.formula.api as smf
import numpy as np
import matplotlib.pyplot as plt
```

[429]:
```
data = pd.read_csv('infmort.csv')
```

[430]:
```
fig = plt.figure()
fig.subplots_adjust(hspace=0.4, wspace=0.2)
sns.set(rc = {'figure.figsize':(20,12)})
sns.set(font_scale = 2)
ax = fig.add_subplot(2, 2, 1)
sns.boxplot(x = 'region', y = 'mortality', data = data)
ax = fig.add_subplot(2, 2, 2)
sns.boxplot(x = 'region', y = 'income', data = data)
ax = fig.add_subplot(2, 2, 3)
sns.scatterplot(x = 'income', y = 'mortality', data = data)
```

[430]:
```
<AxesSubplot:xlabel='income', ylabel='mortality'>
```



Sample Answer (accept any reasonable answer that makes observations about each subplot):

The first boxplot (top, left) shows that many countries in Europe have low mortality, whereas African countries have high mortality. The second boxplot (top, right) shows that many Europian countries have high per capita income, whereas African countries have low per capita income. The scatterplot shows an inverse relationship between mortality and income, that is, mortality seems to decrease with increase in income.

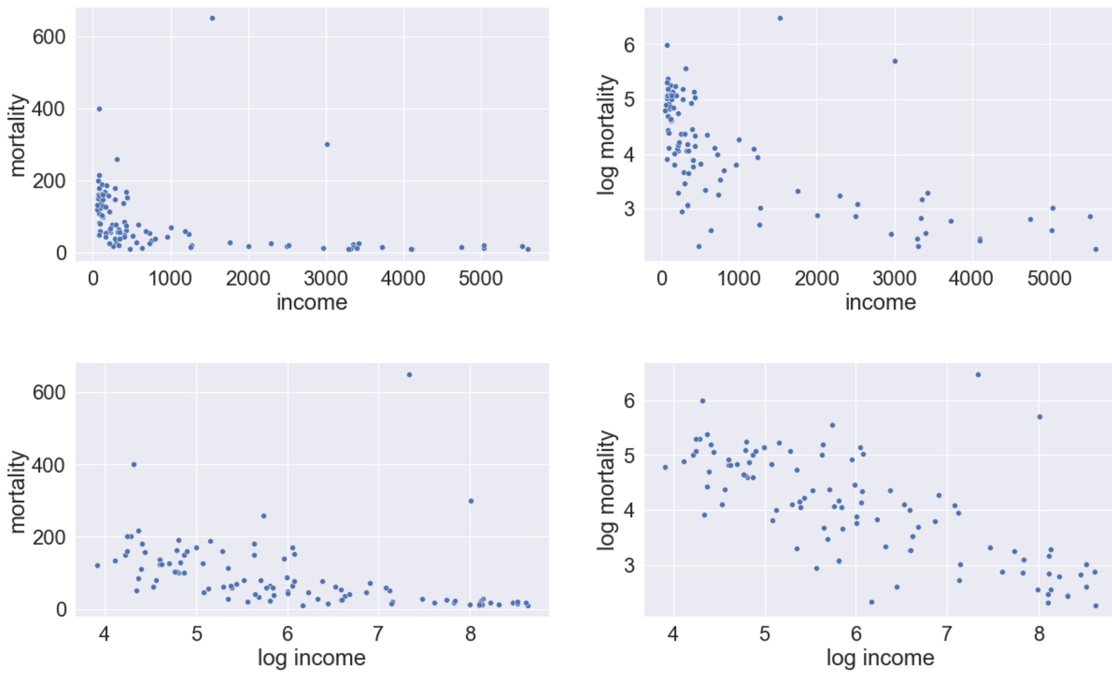
(1b) Europe seems to have the lowest infant mortality, but it also has the highest per capita annual income. We want to see if Europe still has the lowest mortality if we remove the effect of income from the mortality. We will answer this question with the following steps.

(1b-i) Within one visualization, use subplots to plot: (1) mortality against income, (2) log(mortality) against income, (3) mortality against log(income), and (4) log(mortality) against log(income). Be sure to include appropriate axis labels.

(2 points for code)

[103]:
```
fig = plt.figure()
fig.subplots_adjust(hspace=0.4, wspace=0.2)
sns.set(rc = {'figure.figsize':(20,12)})
sns.set(font_scale = 2)
ax = fig.add_subplot(2, 2, 1)
sns.scatterplot(x = 'income', y = 'mortality', data = data)
ax = fig.add_subplot(2, 2, 2)
p2 = sns.scatterplot(x = data.income, y = np.log(data.mortality))
p2.set_ylabel('log mortality')
ax = fig.add_subplot(2, 2, 3)
p3 = sns.scatterplot(x = np.log(data.income), y = 'mortality', data = data)
p3.set_xlabel('log income')
ax = fig.add_subplot(2, 2, 4)
p4 = sns.scatterplot(x = np.log(data.income), y = np.log(data.mortality))
p4.set_xlabel('log income')
p4.set_ylabel('log mortality')
```

[103]:
```
Text(0, 0.5, 'log mortality')
```



(1b-ii) Based on the plots from (1b-i), postulate (describe) an appropriate model to predict mortality as a function of income. Fit the corresponding model and print the model summary.

(1 point for answer, 2 points for code)

From these plots a linear model for log(mortality) against log(income) seems appropriate. Below is the regression.

[431]:
```
model = smf.ols(formula='np.log(mortality)~np.log(income)',data = data).fit()
model.summary()
```

[431]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:      np.log(mortality)   R-squared:                       0.502
Model:                            OLS   Adj. R-squared:                  0.497
Method:                 Least Squares   F-statistic:                     99.84
Date:                Tue, 18 Jan 2022   Prob (F-statistic):           1.14e-16
Time:                        01:24:56   Log-Likelihood:                -104.34
No. Observations:                 101   AIC:                             212.7
Df Residuals:                      99   BIC:                             217.9
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
==
```

```
                          coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
--
Intercept           7.1458      0.317     22.575      0.000       6.518
7.774
np.log(income)     -0.5118      0.051     -9.992      0.000      -0.613
-0.410
==============================================================================
Omnibus:                       38.668   Durbin-Watson:                   1.898
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              129.408
Skew:                           1.255   Prob(JB):                     7.93e-29
Kurtosis:                       7.945   Cond. No.                         29.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

(1b-iii) Update the model developed in the previous question (1b-ii) by adding *region* as a predictor. Print the model summary.

(2 points for code)

[115]:
```
model = smf.ols(formula='np.log(mortality)~region+np.log(income)',data = data).
 ↪fit()
model.summary()
```

[115]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:      np.log(mortality)   R-squared:                       0.616
Model:                            OLS   Adj. R-squared:                  0.600
Method:                 Least Squares   F-statistic:                     38.55
Date:                Mon, 17 Jan 2022   Prob (F-statistic):           3.29e-19
Time:                        17:11:32   Log-Likelihood:                -91.189
No. Observations:                 101   AIC:                             192.4
Df Residuals:                      96   BIC:                             205.5
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
======
                              coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
```

```
------
Intercept              6.4030      0.358     17.871      0.000       5.692
7.114
region[T.Americas]    -0.6022      0.190     -3.166      0.002      -0.980
-0.225
region[T.Asia]        -0.7233      0.163     -4.431      0.000      -1.047
-0.399
region[T.Europe]      -1.2028      0.259     -4.647      0.000      -1.717
-0.689
np.log(income)        -0.2994      0.067     -4.441      0.000      -0.433
-0.166
==============================================================================
Omnibus:                       44.959   Durbin-Watson:                   1.847
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              174.758
Skew:                           1.428   Prob(JB):                     1.13e-38
Kurtosis:                       8.777   Cond. No.                         42.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

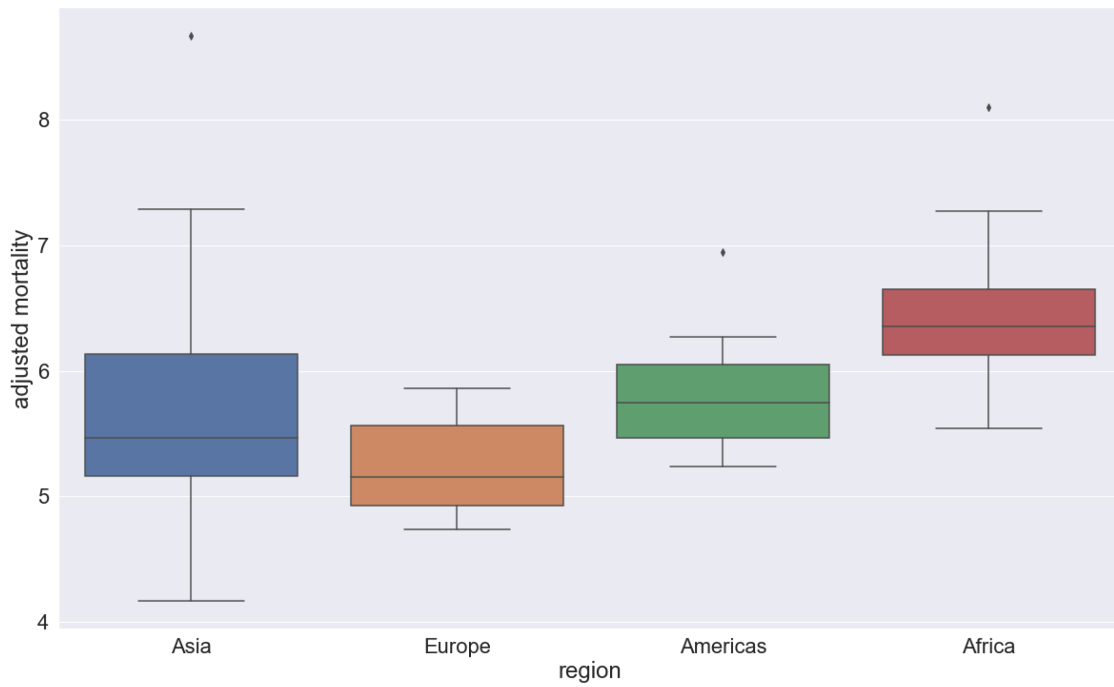
(1b-iv) Use the model developed in the previous question (1b-iii) to compute *adjusted_mortality* for each observation in the data, where adjusted mortality is the mortality after removing the estimated effect of income. Make a boxplot of adjusted mortality against region. Be sure to include appropriate axis labels.

(3 points for code)

[117]:
```
adj_mortality = np.log(data.mortality) - model.params['np.log(income)']*np.
 ↪log(data.income)
```

[121]:
```
p = sns.boxplot(x = data.region, y = adj_mortality)
p.set_ylabel('adjusted mortality')
```

[121]:
```
Text(0, 0.5, 'adjusted mortality')
```



(1b-v) Does Europe still has the lowest mortality after removing the effect of income from mortality? After adjusting for income, do more African / Asian / American countries seem to do better than European countries with regard to mortality?

(2 points for answers)

Yes, Europe still has the lowest mortality, on an average. However, once we adjust for the income, some of the Asian countries seem to be doing better than the European countries.

1.3 Part 2

The dataset *soc_ind.csv* contains the GDP per capita of some countries along with several social indicators.

This part is worth 21 points overall.

(2a) For a simple linear regression model predicting *gdpPerCapita*, which predictor will provide the best model fit *(ignore categorical predictors)*? Let that predictor be P.

(1 point for code, 1 point for answer)

[432]:
```
data = pd.read_csv('soc_ind.csv')
```

[433]:
```
data.corr()
```

[433]:
```
                              Index  economicActivityFemale  \
Index                      1.000000                0.963079
```

```
economicActivityFemale  0.963079                1.000000
economicActivityMale    0.083285                0.096822
gdpPerCapita            0.073671                0.052964
illiteracyFemale       -0.192510               -0.177559
illiteracyMale         -0.169524               -0.141644
infantMortality        -0.008166                0.011667
lifeFemale             -0.000561               -0.029456
lifeMale               -0.082617               -0.103137

                        economicActivityMale  gdpPerCapita  illiteracyFemale  \
Index                               0.083285      0.073671         -0.192510
economicActivityFemale              0.096822      0.052964         -0.177559
economicActivityMale                1.000000     -0.167231          0.428277
gdpPerCapita                       -0.167231      1.000000         -0.457012
illiteracyFemale                    0.428277     -0.457012          1.000000
illiteracyMale                      0.450352     -0.471689          0.959948
infantMortality                     0.382949     -0.584060          0.792230
lifeFemale                         -0.367124      0.604029         -0.783343
lifeMale                           -0.295536      0.592267         -0.709332

                        illiteracyMale  infantMortality  lifeFemale  lifeMale
Index                        -0.169524        -0.008166   -0.000561 -0.082617
economicActivityFemale       -0.141644         0.011667   -0.029456 -0.103137
economicActivityMale          0.450352         0.382949   -0.367124 -0.295536
gdpPerCapita                 -0.471689        -0.584060    0.604029  0.592267
illiteracyFemale              0.959948         0.792230   -0.783343 -0.709332
illiteracyMale                1.000000         0.755333   -0.750800 -0.684587
infantMortality               0.755333         1.000000   -0.947045 -0.915713
lifeFemale                   -0.750800        -0.947045    1.000000  0.974262
lifeMale                     -0.684587        -0.915713    0.974262  1.000000
```

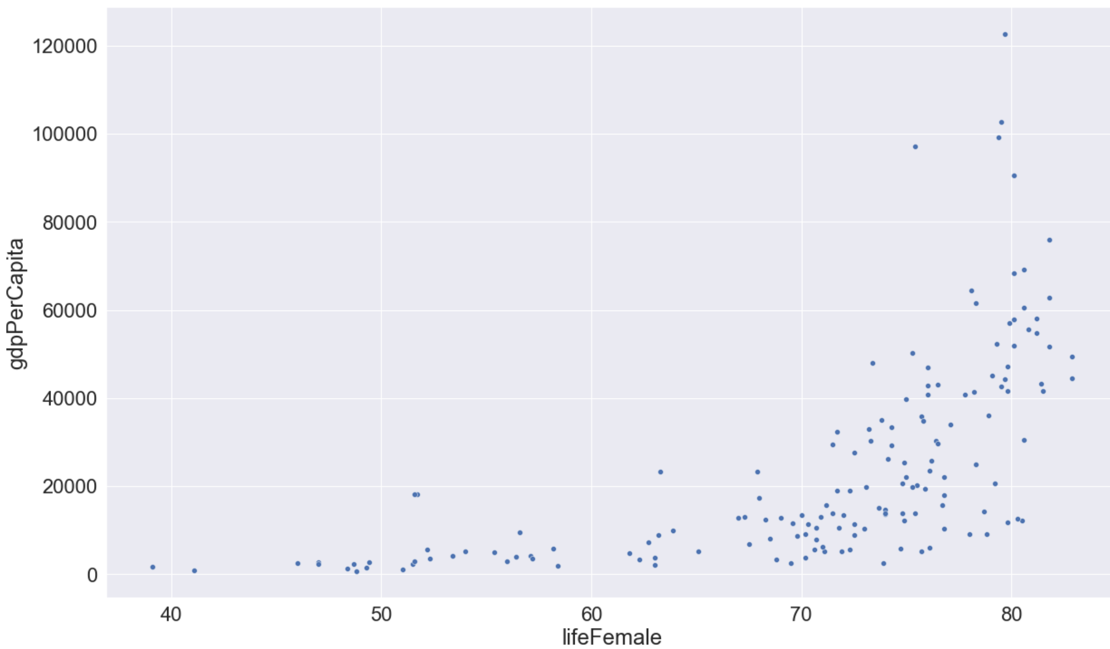
As *lifeFemale* has the highest linear correlation with *gdpPerCapita*, it will provide the best model fit.

(2b) Make a scatterplot of *gdpPerCapita* vs P. Does the relationship between *gdpPerCapita* and P seem linear or non-linear?

(1 point for code, 1 point for answer)

[434]:
```
sns.scatterplot(x = data.lifeFemale, y = data.gdpPerCapita)
```

[434]:
```
<AxesSubplot:xlabel='lifeFemale', ylabel='gdpPerCapita'>
```



Based on the scatterplot above, the relationship between *gdpPerCapita* and *lifeFemale* seems non-linear.

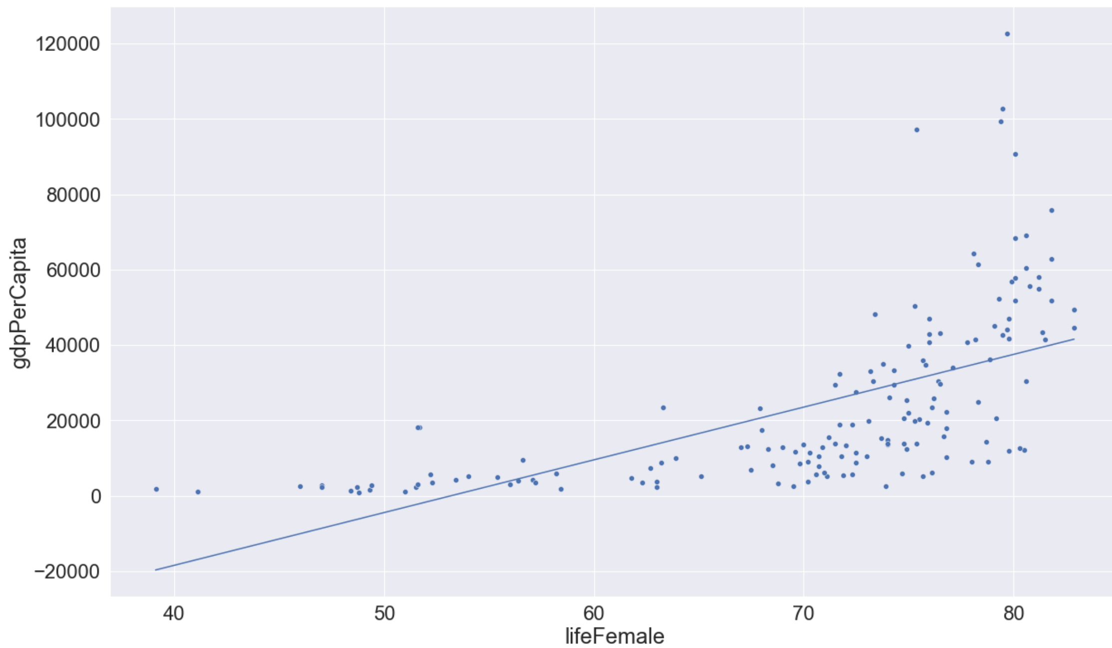
(2c) If the relationship identified in (2b) is non-linear, investigate transformation(s) of the predictor P in the model that might improve the model fit. To do so, use scatterplots displaying values of *gdpPerCapita* against corresponding values of P under different transformation(s).

When you've settled on an optimal model, report the predictors included in that model. Fit your new model and report the change in the R-squared value of this transformed model as compared to the simple linear regression model with only P.

(4 points for code, 2 points for answers)

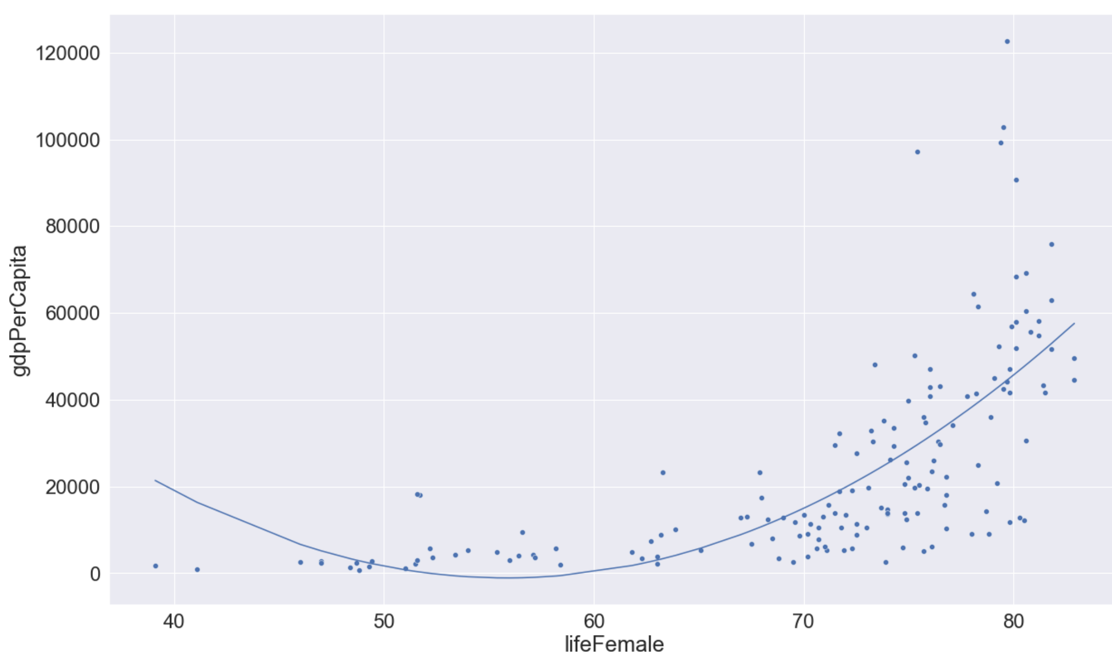
[435]:
```
#Visualzing model fit with only P as predictor
model = smf.ols(formula='gdpPerCapita~lifeFemale',data = data).fit()
sns.scatterplot(x = data.lifeFemale, y = data.gdpPerCapita)
sns.lineplot(x = data.lifeFemale, y = model.fittedvalues)
```

[435]:
```
<AxesSubplot:xlabel='lifeFemale', ylabel='gdpPerCapita'>
```



[436]:
```
#Adding square of lifeFemale as a predictor and visualizing model fit
model = smf.ols(formula='gdpPerCapita~lifeFemale+I(lifeFemale**2)',data = data).
 ↪fit()
sns.scatterplot(x = data.lifeFemale, y = data.gdpPerCapita)
sns.lineplot(x = data.lifeFemale, y = model.fittedvalues)
```

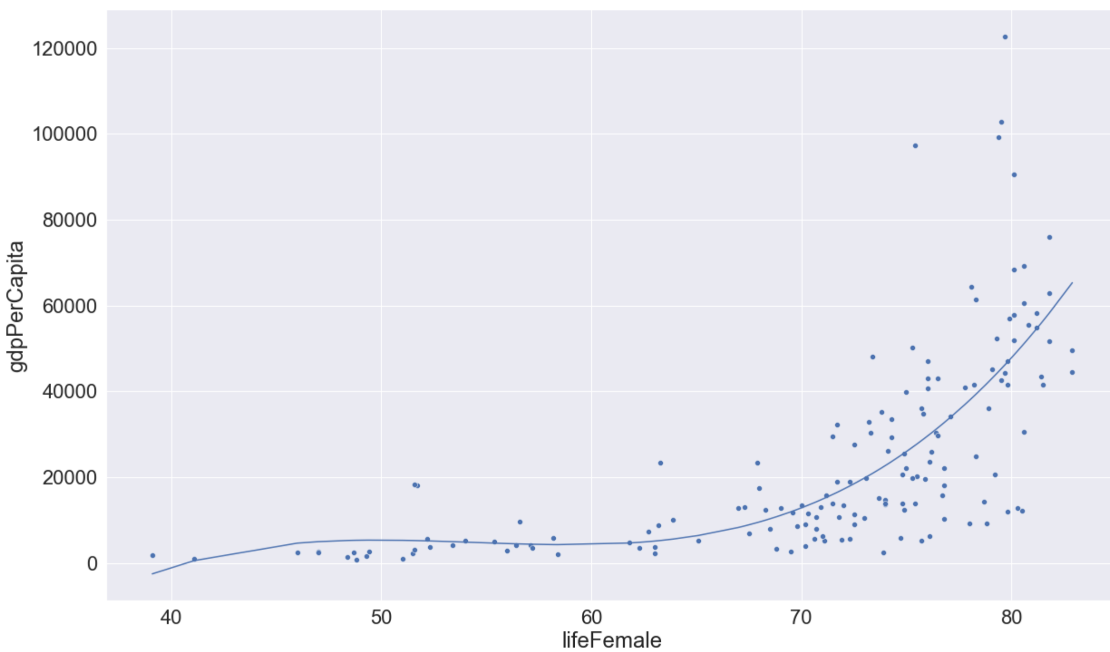
[436]: <AxesSubplot:xlabel='lifeFemale', ylabel='gdpPerCapita'>



The transformation seems to improve the model fit for higher values of *lifeFemale*, but not for its smaller values.

[420]:
```
#Adding square and cube of lifeFemale as predictors and visualizing model fit
model = smf.
 ↪ols(formula='gdpPerCapita~lifeFemale+I(lifeFemale**2)+I(lifeFemale**3)',data␣
 ↪= data).fit()
sns.scatterplot(x = data.lifeFemale, y = data.gdpPerCapita)
sns.lineplot(x = data.lifeFemale, y = model.fittedvalues)
```

[420]: <AxesSubplot:xlabel='lifeFemale', ylabel='gdpPerCapita'>



The squared and cube transformations of *lifeFemale* seem to provide a better model fit than previous models.

[437]:
```
#R-squared of the simple linear regression model with only P as predictor
model = smf.ols(formula='gdpPerCapita~lifeFemale',data = data).fit()
model.rsquared
```

[437]: 0.3648504777962377

[438]:
```
#R-squared of the transformed model
```

```
model = smf.
↪ols(formula='gdpPerCapita~lifeFemale+I(lifeFemale**2)+I(lifeFemale**3)',data␣
↪= data).fit()
model.rsquared
```

[438]: 0.5216581673999969

The predictors of the transformed model are *lifeFemale*, *lifeFemale*^{2}, and *lifeFemale*^{3}

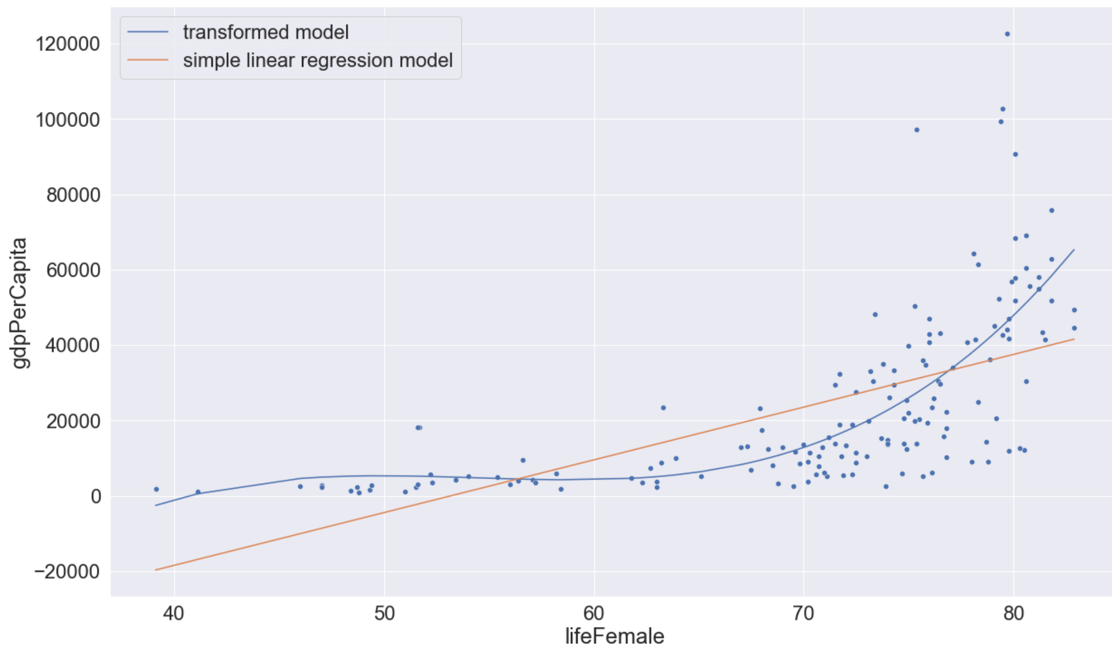
The R-squared of the model with transformed model is around 16 percentage points higher than the R-squared of the model with only P.

(2d) Plot the regression curve of the transformed model (developed in the previous question (2c)) over the scatterplot in (2b) to visualize model fit. Also include the regression line of the simple linear regression model with only P on the same plot. Be sure to include a legend to distinguish the two models.

(3 points for code)

[423]:
```
#Transformed model
model = smf.
↪ols(formula='gdpPerCapita~lifeFemale+I(lifeFemale**2)+I(lifeFemale**3)',data␣
↪= data).fit()
sns.scatterplot(x = data.lifeFemale, y = data.gdpPerCapita)
sns.lineplot(x = data.lifeFemale, y = model.fittedvalues,label = 'transformed␣
↪model')
#Simple linear regression model
model = smf.ols(formula='gdpPerCapita~lifeFemale',data = data).fit()
sns.lineplot(x = data.lifeFemale, y = model.fittedvalues, label = 'simple␣
↪linear regression model')
plt.legend(loc='upper left')
```

[423]: <matplotlib.legend.Legend at 0x1fc84908910>



(2e) Develop a model to predict *gdpPerCapita* with *P* and *continent* as predictors. For a given value of *P*, which continents **do not** have a signficant difference between their mean *gdpPerCapita* and that of Africa? Consider a significance level of 5%.

(1 point for code, 1 point for answer)

[439]:
```
model = smf.ols(formula='gdpPerCapita~lifeFemale+continent',data = data).fit()
model.summary()
```

[439]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:           gdpPerCapita   R-squared:                       0.508
Model:                            OLS   Adj. R-squared:                  0.488
Method:                 Least Squares   F-statistic:                     25.43
Date:                Tue, 18 Jan 2022   Prob (F-statistic):           1.28e-20
Time:                        01:34:19   Log-Likelihood:                -1723.6
No. Observations:                 155   AIC:                             3461.
Df Residuals:                     148   BIC:                             3483.
Df Model:                           6
Covariance Type:            nonrobust
================================================================================
==============
                                  coef    std err          t      P>|t|
[0.025      0.975]
```

```
--------------------------------------------------------------------------------
--------------
Intercept                    -7.208e+04   1.14e+04     -6.305      0.000
-9.47e+04   -4.95e+04
continent[T.Asia]             1324.7980   4805.099      0.276      0.783
-8170.667    1.08e+04
continent[T.Europe]           9167.0203   5785.650      1.584      0.115
-2266.134    2.06e+04
continent[T.North America] -1.446e+04   5947.502     -2.431      0.016
-2.62e+04   -2704.270
continent[T.Oceania]        -1.429e+04   6063.764     -2.357      0.020
-2.63e+04   -2307.304
continent[T.South America] -1.329e+04   6462.516     -2.056      0.042
-2.61e+04    -516.198
lifeFemale                    1393.4213    194.062      7.180      0.000
1009.931    1776.911
==============================================================================
Omnibus:                       67.873   Durbin-Watson:                   1.942
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              231.081
Skew:                           1.701   Prob(JB):                     6.63e-51
Kurtosis:                       7.920   Cond. No.                         721.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

Based on the p-values, Asia and Europe do not have a significant difference between their mean *gdpPerCapita* and that of Africa, for a given value of P. Note that the level with no dummy variable - Africa is the baseline in this model. So we can directly make comparisons with Africa using the p-values of other dummy variables.

(2f) The model developed in (e) has a limitation. It assumes that the increase in mean *gdpPerCapita* with a unit increase in P does not depend on the *continent*. Eliminate this limitation by including interaction of *continent* with P in the model developed in (e). Print the model summary of the model with interactions.

(2 points for code)

[425]:
```
model = smf.ols(formula='gdpPerCapita~lifeFemale*continent',data = data).fit()
model.summary()
```

[425]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:           gdpPerCapita   R-squared:                       0.605
```

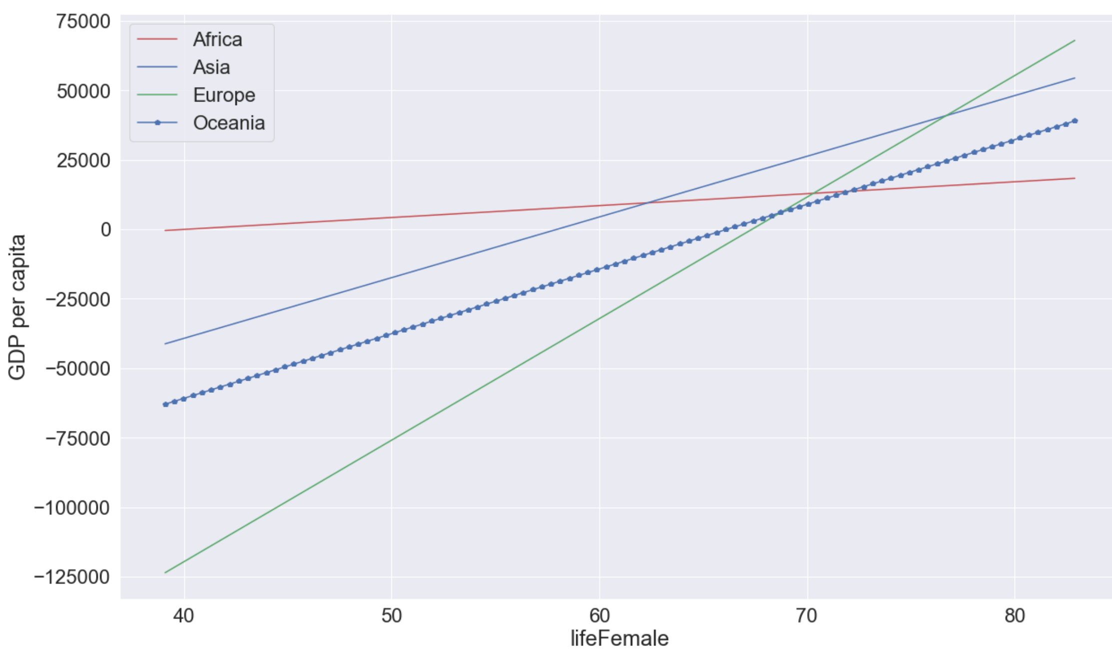
```
Model:                            OLS   Adj. R-squared:                  0.574
Method:                 Least Squares   F-statistic:                     19.90
Date:                Tue, 18 Jan 2022   Prob (F-statistic):           7.99e-24
Time:                        01:21:19   Log-Likelihood:                -1706.6
No. Observations:                 155   AIC:                             3437.
Df Residuals:                     143   BIC:                             3474.
Df Model:                          11
Covariance Type:            nonrobust
================================================================================
=========================
                                             coef    std err          t
P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
-------------------------
Intercept                              -1.723e+04   1.53e+04     -1.129
0.261   -4.74e+04    1.29e+04
continent[T.Asia]                      -1.094e+05   2.63e+04     -4.156
0.000   -1.61e+05   -5.74e+04
continent[T.Europe]                    -2.774e+05   6.63e+04     -4.185
0.000   -4.08e+05   -1.46e+05
continent[T.North America]               -6.6e+04   4.88e+04     -1.352
0.178   -1.62e+05    3.05e+04
continent[T.Oceania]                   -1.367e+05   5.78e+04     -2.364
0.019   -2.51e+05   -2.24e+04
continent[T.South America]             -7830.3082   8.18e+04     -0.096
0.924   -1.69e+05    1.54e+05
lifeFemale                               428.5595    264.214      1.622
0.107     -93.711     950.830
lifeFemale:continent[T.Asia]            1755.1049    400.782      4.379
0.000     962.882    2547.328
lifeFemale:continent[T.Europe]          3944.6364    869.916      4.535
0.000    2225.080    5664.193
lifeFemale:continent[T.North America]    921.1328    667.359      1.380
0.170    -398.031    2240.297
lifeFemale:continent[T.Oceania]         1898.4382    812.766      2.336
0.021     291.851    3505.026
lifeFemale:continent[T.South America]    135.3138   1134.388      0.119
0.905   -2107.022    2377.650
==============================================================================
Omnibus:                       73.348   Durbin-Watson:                   1.900
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              290.376
Skew:                           1.785   Prob(JB):                     8.82e-64
Kurtosis:                       8.676   Cond. No.                     5.23e+03
==============================================================================

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

```
specified.
[2] The condition number is large, 5.23e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

(2g) Use the model developed in (2f) to plot regression lines for Africa, Asia, Europe, and Oceania. Put *gdpPerCapita* on the vertical axis and P on the horizontal axis. Use a legend to distinguish among the regression lines of the continents.

(3 points for code)

[426]:
```
#Visualizing the developed model with interaction terms
x = np.linspace(data.lifeFemale.min(),data.lifeFemale.max(),100)
plt.plot(x, model.params['lifeFemale']*x+model.params['Intercept'], '-r',␣
↪label='Africa')
plt.plot(x, (model.params['lifeFemale']+model.params['lifeFemale:continent[T.
↪Asia]'])*x+model.params['Intercept']+model.params['continent[T.Asia]'],␣
↪'-b', label='Asia')
plt.plot(x, (model.params['lifeFemale']+model.params['lifeFemale:continent[T.
↪Europe]'])*x+model.params['Intercept']+model.params['continent[T.Europe]'],␣
↪'-g', label='Europe')
plt.plot(x, (model.params['lifeFemale']+model.params['lifeFemale:continent[T.
↪Oceania]'])*x+model.params['Intercept']+model.params['continent[T.
↪Oceania]'], '-p', label='Oceania')
plt.legend(loc='upper left')
plt.xlabel('lifeFemale')
plt.ylabel('GDP per capita')
plt.show()
```



(2h) Based on the plot develop in the previous question, which continent has the highest increase in mean *gdpPerCapita* for a unit increase in P, and which one has the least?

(1 point for answers)

Europe has the highest increase in mean *gdpPerCapita* for a unit increase in P as the slope of its regression line is the highest. Africa has the least increase in mean *gdpPerCapita* for a unit increase in P, as the slope of its regression line is the least.