Final_Esha

May 18, 2022

[ ]:
```python
# import necessary packages
#Import necessary packages
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from plotnine import *

from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors

from sklearn.cluster import DBSCAN


from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from sklearn.cluster import AgglomerativeClustering


from sklearn.metrics import silhouette_score
import scipy.cluster.hierarchy as sch
from matplotlib import pyplot as plt

from sklearn.linear_model import LinearRegression # Linear Regression Model
from sklearn.preprocessing import StandardScaler #Z-score variables
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score #model␣
↪evaluation

from sklearn.model_selection import train_test_split # simple TT split cv
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.linear_model import RidgeCV, LassoCV
from sklearn.decomposition import PCA
%matplotlib inline
```

[ ]:
```python
#getting csv data from google drive
```

```
url='https://drive.google.com/file/d/1sX4PEmbEnY4vPFkVhVJAX5X7F2PqKuF3/view?
↪usp=sharing'
url='https://drive.google.com/uc?id=' + url.split('/')[-2]
data = pd.read_csv(url)
data = data.replace(',','.', regex=True)
data.head()
```

```
[ ]:    Unnamed: 0  Happiness Rank      Country          Region Happiness Score  \
0           0               1  Switzerland  Western Europe           7.587
1           1               2      Iceland  Western Europe           7.561
2           2               3      Denmark  Western Europe           7.527
3           3               4       Norway  Western Europe           7.522
4           4               5       Canada   North America           7.427

  Economy (GDP per Capita) Family (Social Support) Health (Life Expectancy)  \
0                  1.39651                 1.34951                  0.94143
1                  1.30232                 1.40223                  0.94784
2                  1.32548                 1.36058                  0.87464
3                    1.459                 1.33095                  0.88521
4                  1.32629                 1.32261                  0.90563

   Freedom Trust (Government Corruption) Generosity  Year
0  0.66557                       0.41978    0.29678  2015
1  0.62877                       0.14145     0.4363  2015
2  0.64938                       0.48357    0.34139  2015
3  0.66973                       0.36503    0.34699  2015
4  0.63297                       0.32957    0.45811  2015
```

```
[ ]: data["New_Region"] = data["Region"]

data.loc[data["New_Region"] == "Australia and New Zealand", "New_Region"] =␣
↪"Australia"

data.loc[data["New_Region"] == "Central and Eastern Europe", "New_Region"] =␣
↪"Europe"
data.loc[data["New_Region"] == "Commonwealth of Independent States",␣
↪"New_Region"] = "Europe"
data.loc[data["New_Region"] == "Western Europe", "New_Region"] = "Europe"

data.loc[data["New_Region"] == "North America", "New_Region"] = "North America"
data.loc[data["New_Region"] == "North America and ANZ", "New_Region"] = "North␣
↪America"

data.loc[data["New_Region"] == "Middle East and North Africa", "New_Region"] =␣
↪"Middle East and Africa"
```

```
data.loc[data["New_Region"] == "Middle East and Northern Africa", "New_Region"]␣
↪= "Middle East and Africa"
data.loc[data["New_Region"] == "Sub-Saharan Africa", "New_Region"] = "Middle␣
↪East and Africa"

data.loc[data["New_Region"] == "East Asia", "New_Region"] = "Asia"
data.loc[data["New_Region"] == "Eastern Asia", "New_Region"] = "Asia"
data.loc[data["New_Region"] == "South Asia", "New_Region"] = "Asia"
data.loc[data["New_Region"] == "Southeast Asia", "New_Region"] = "Asia"
data.loc[data["New_Region"] == "Southeastern Asia", "New_Region"] = "Asia"
data.loc[data["New_Region"] == "Southern Asia", "New_Region"] = "Asia"

data.head()
```

[ ]:

```
   Unnamed: 0  Happiness Rank      Country          Region Happiness Score  \
0           0               1  Switzerland  Western Europe           7.587
1           1               2      Iceland  Western Europe           7.561
2           2               3      Denmark  Western Europe           7.527
3           3               4       Norway  Western Europe           7.522
4           4               5       Canada   North America           7.427

  Economy (GDP per Capita) Family (Social Support) Health (Life Expectancy)  \
0                  1.39651                 1.34951                  0.94143
1                  1.30232                 1.40223                  0.94784
2                  1.32548                 1.36058                  0.87464
3                    1.459                 1.33095                  0.88521
4                  1.32629                 1.32261                  0.90563

   Freedom Trust (Government Corruption) Generosity  Year     New_Region
0  0.66557                       0.41978    0.29678  2015         Europe
1  0.62877                       0.14145     0.4363  2015         Europe
2  0.64938                       0.48357    0.34139  2015         Europe
3  0.66973                       0.36503    0.34699  2015         Europe
4  0.63297                       0.32957    0.45811  2015  North America
```

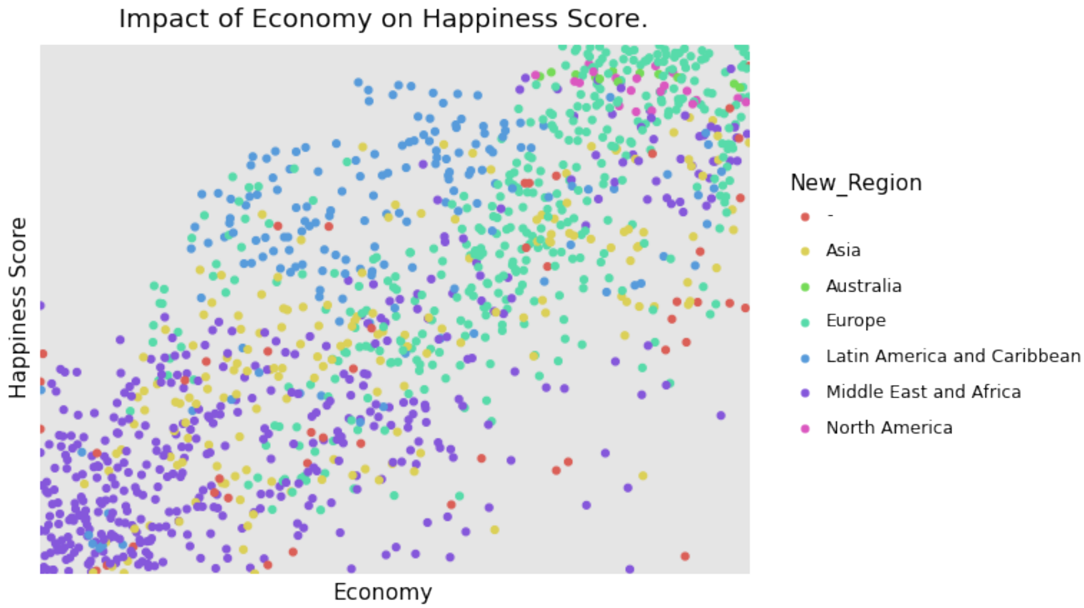
Question 1.

Since the dataset is focused on predicting happiness score, which of the predictors is the most influential and what type of relationship do they have with the happiness score? Which of the variables are least significant and can possibly be removed from the model?

[ ]:

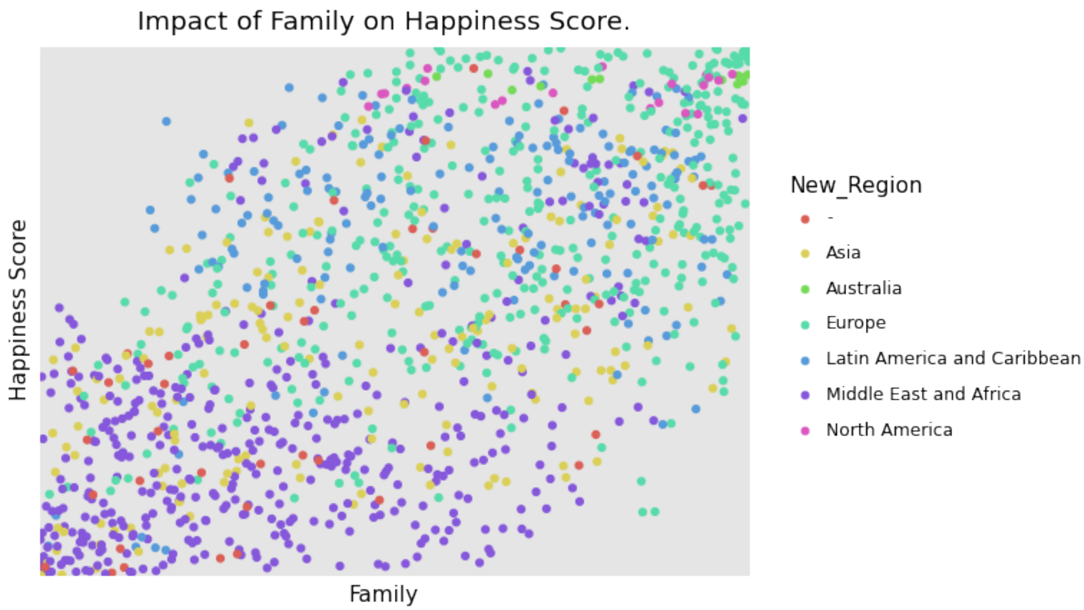
```
#set the variables
predictors = ["Economy (GDP per Capita)","Family (Social Support)","Health␣
↪(Life Expectancy)",
                   "Freedom","Trust (Government Corruption)","Generosity"]
X = data[predictors]
y = data["Happiness Score"]
```

[ ]:
```
(ggplot(data, aes(x = "Economy (GDP per Capita)", y = "Happiness Score", color␣
↪= "New_Region")) + \
geom_point() + theme_minimal() + ggtitle("Impact of Economy on Happiness Score.
↪") + labs(x = "Economy", y = "Happiness Score") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



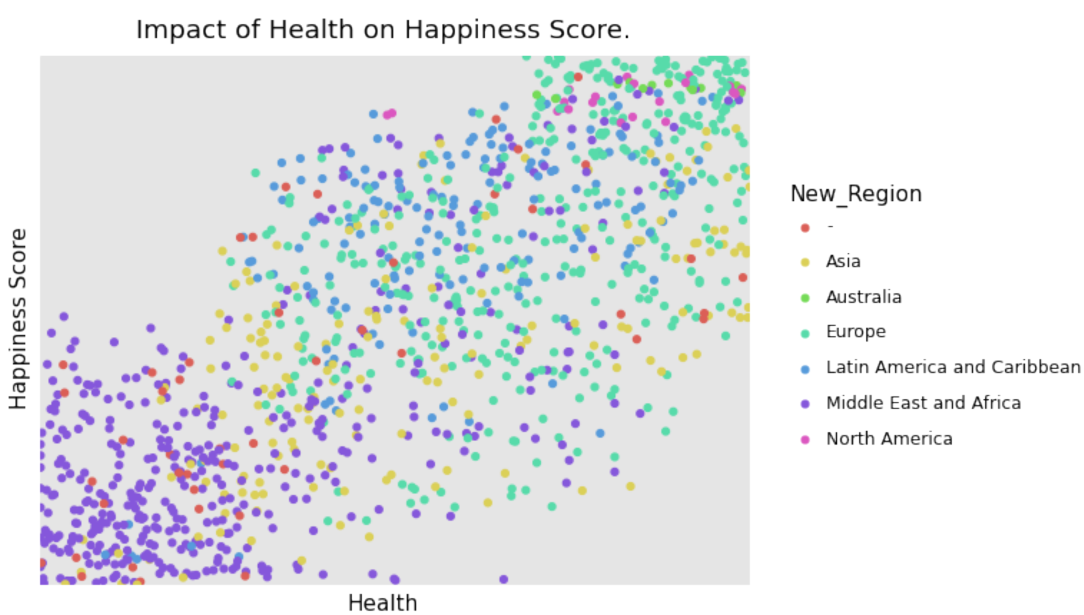
[ ]: <ggplot: (8763692805025)>

[ ]:
```
(ggplot(data, aes(x = "Family (Social Support)", y = "Happiness Score", color =␣
↪"New_Region")) + geom_point() + \
theme_minimal() + ggtitle("Impact of Family on Happiness Score.") + labs(x =␣
↪"Family", y = "Happiness Score") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



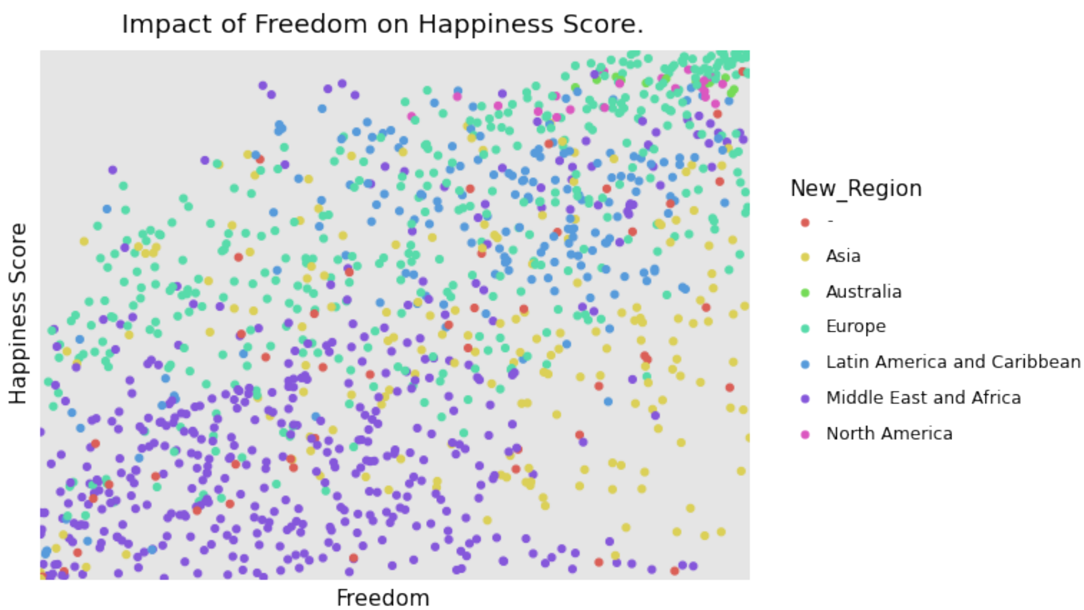
[ ]: <ggplot: (8763693208409)>

[ ]:
```
(ggplot(data, aes(x = "Health (Life Expectancy)", y = "Happiness Score", color␣
↪= "New_Region")) + geom_point() + \
theme_minimal() + ggtitle("Impact of Health on Happiness Score.") + labs(x =␣
↪"Health", y = "Happiness Score") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



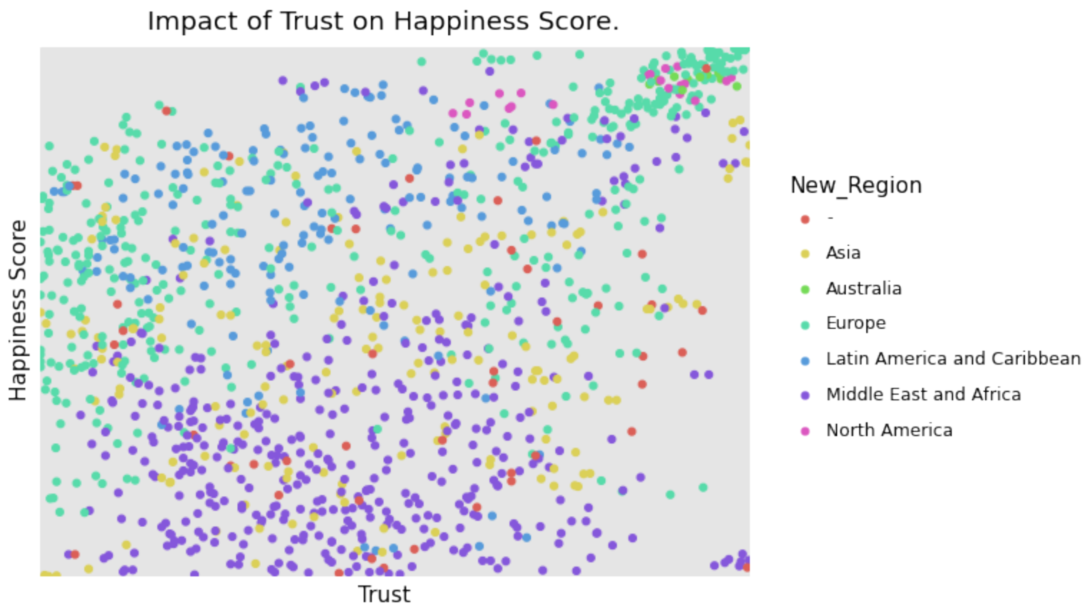
[ ]: <ggplot: (8763692766157)>

[ ]:
```
(ggplot(data, aes(x = "Freedom", y = "Happiness Score", color = "New_Region"))␣
↪+ geom_point() + \
theme_minimal() + ggtitle("Impact of Freedom on Happiness Score.") + labs(x =␣
↪"Freedom", y = "Happiness Score") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



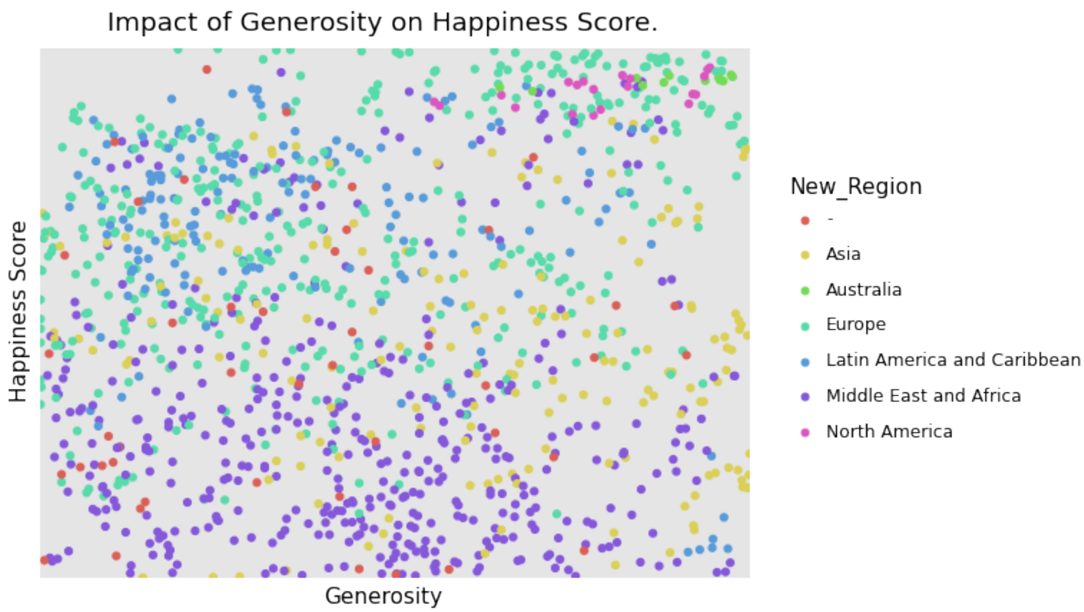
[ ]: <ggplot: (8763692777509)>

[ ]:
```
(ggplot(data, aes(x = "Trust (Government Corruption)", y = "Happiness Score",␣
↪color = "New_Region")) + geom_point() + \
theme_minimal() + ggtitle("Impact of Trust on Happiness Score.") + labs(x =␣
↪"Trust", y = "Happiness Score") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



[ ]: <ggplot: (8763697345849)>

```
[ ]: (ggplot(data, aes(x = "Generosity", y = "Happiness Score", color =
     "New_Region")) + geom_point() + \
     theme_minimal() + ggtitle("Impact of Generosity on Happiness Score.") + labs(x
     = "Generosity", y = "Happiness Score") + \
     theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



[ ]: <ggplot: (8763696365817)>

The first two variables seem to have a pretty linear relationship with the happiness score. The data points are still spread apart a lot but we can see a clear trend upwards.

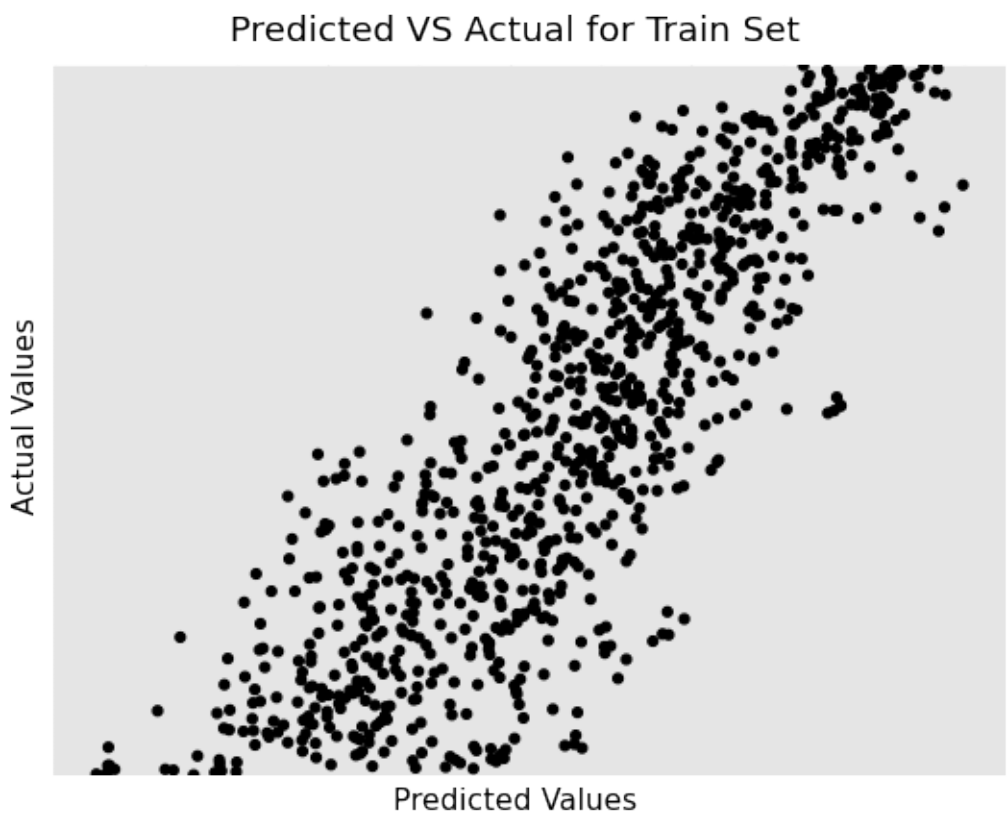
The next two variables also seem to have linear relationships. Health definitely has a stronger one because the data points are less spread apart. Freedom still has some sort of linearity but the data points are really spread apart.

The last two variables don't seem to have much of a linear relationship with happiness score. The data points are spread all across the graph and there is no pattern. These two variables could potentially be removed later on when trying to improve our model.

[ ]:
```
#creating a linear regression model
X_train, X_test, y_train, y_test = train_test_split(data[predictors],y,␣
↪test_size = 0.2)

z = StandardScaler()
z.fit(X_train)

X_train = z.transform(X_train)
X_test = z.transform(X_test)
```

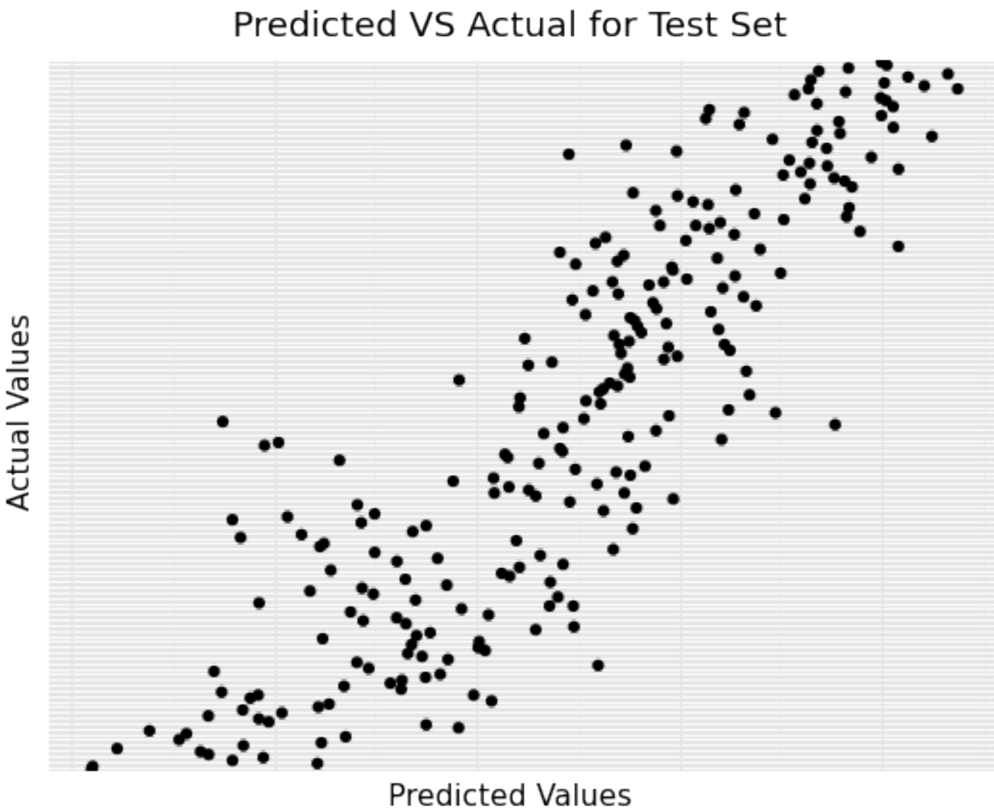
[ ]:
```
lr = LinearRegression()
lr.fit(X_train, y_train)

#getting the different predictions
y_train_preds = lr.predict(X_train)
#error_train = y_train - y_train_preds
assump_train = pd.DataFrame({"predicted":y_train_preds,"actual":y_train})

y_test_preds = lr.predict(X_test)
#error_test = y_test - y_test_preds
assump_test = pd.DataFrame({"predicted":y_test_preds,"actual":y_test})
```

[ ]:
```
(ggplot(assump_train, aes(x = "predicted", y = "actual")) + geom_point() +␣
↪theme_minimal() + \
 ggtitle("Predicted VS Actual for Train Set") + labs(x = "Predicted Values", y␣
↪= "Actual Values") + \
 theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



[ ]: <ggplot: (8763695414857)>

[ ]:
```
(ggplot(assump_test, aes(x = "predicted", y = "actual")) + geom_point() +␣
↪theme_minimal() + \
ggtitle("Predicted VS Actual for Test Set") + labs(x = "Predicted Values", y =␣
↪"Actual Values") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



[ ]: <ggplot: (8763693440877)>

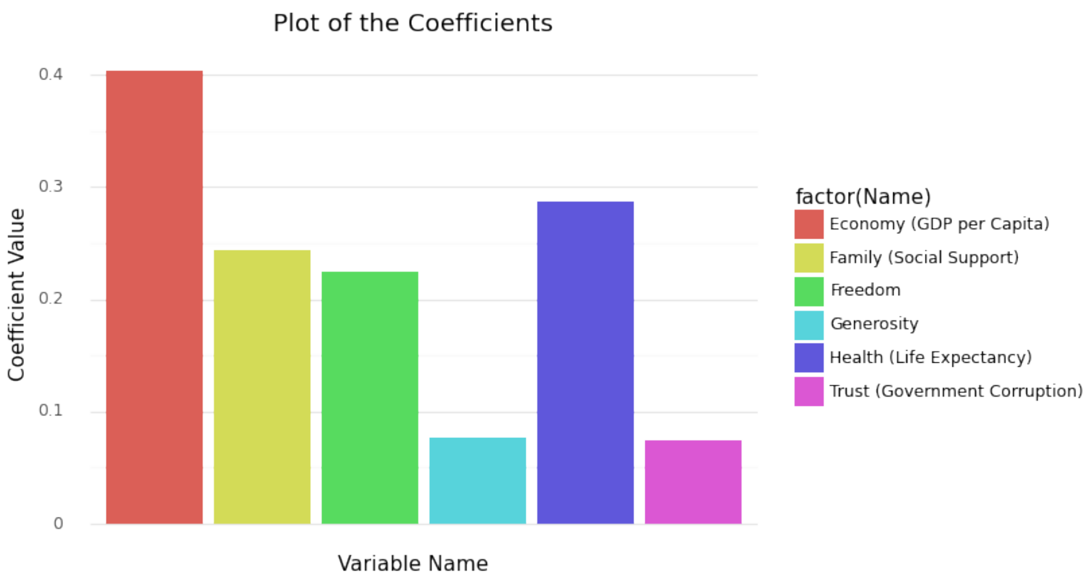
[ ]:
```
#model validation
print("For the Train Set")
print("MSE:",mean_squared_error(y_train,y_train_preds))
print("R^2:",r2_score(y_train,y_train_preds))

print("For the Test Set")
print("MSE:",mean_squared_error(y_test,y_test_preds))
print("R^2:",r2_score(y_test,y_test_preds))
```

```
For the Train Set
MSE: 0.3317945986070136
R^2: 0.7346216654462445
For the Test Set
MSE: 0.27678687161669524
R^2: 0.7679566307140836
```

The MSE for both sets seem to be relatively low. The R^2 value is around 75% for the test set and 74% for our training set. Since the R^2 value for the training set is lower so we can say our model is not overfit.

[ ]:
```
coefficients = pd.DataFrame({"Coefficients":lr.coef_,"Name":predictors})
```

[ ]:
```
(ggplot(coefficients, aes(x = "Name", y = "Coefficients", fill =␣
↪"factor(Name)")) + geom_bar(stat = "identity") + theme_minimal() + \
ggtitle("Plot of the Coefficients") + labs(x = "Variable Name", y =␣
↪"Coefficient Value") + \
theme(panel_grid_major_x = element_blank(), axis_text_x = element_blank()))
```



[ ]: <ggplot: (8763696428925)>

[ ]:
```
coefficients
```

[ ]:

	Coefficients	Name
0	0.403514	Economy (GDP per Capita)
1	0.244217	Family (Social Support)
2	0.287454	Health (Life Expectancy)
3	0.224509	Freedom
4	0.074362	Trust (Government Corruption)
5	0.077274	Generosity

Answer for Question 1

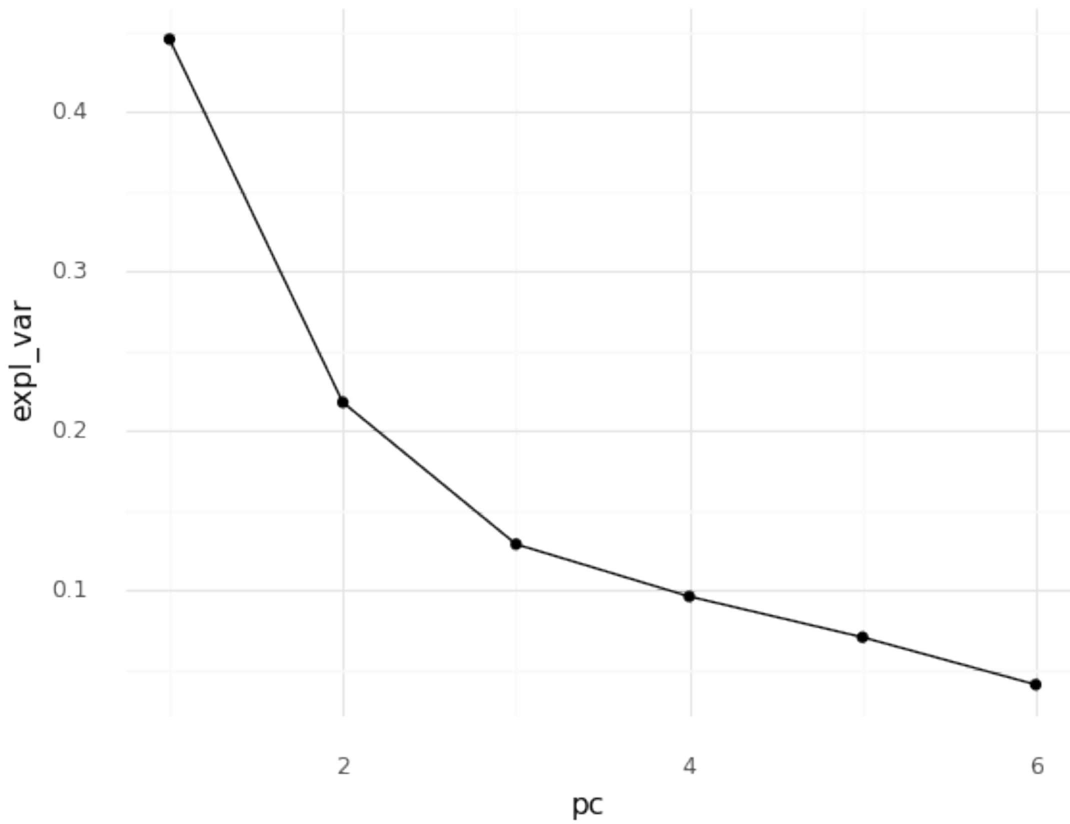
Looking at the coefficients we have as a result of our linear regression model we see that we have low coefficients for generosity and trust which are the same two variables that didn't have a linear relationship with the happiness score. We can also identify our most influential variables which are economy, health and freedom because they have the biggest coefficients.

We supported our theory of possibly removing trust and generosity to better our model because of their weak linear relationship with happiness score and their low coefficients.

Question 2

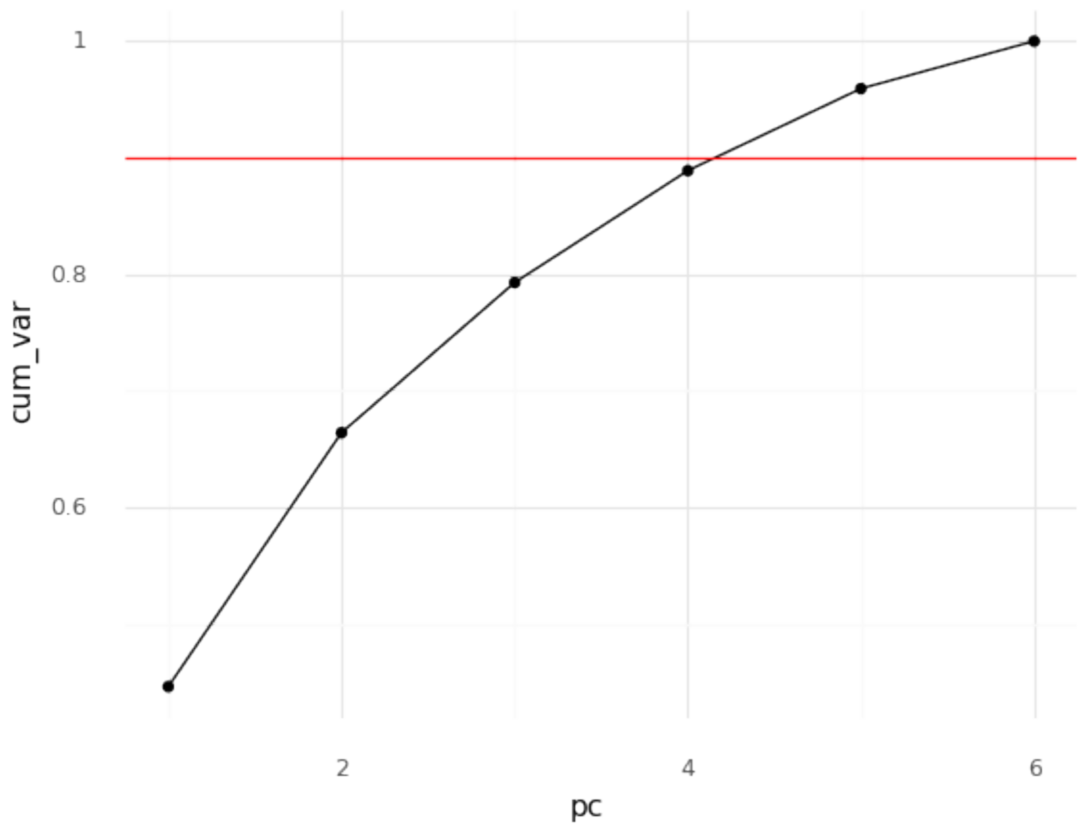
How much of a difference do we see in the mean absolute error when comparing the model with all the predicting variables to a model using PCA that retains enough PC's to keep 85% of the variance in the data? Can we compare the results with those of a Lasso Model to check which variables would be considered noise?

[ ]:
```
#pca moment
pca = PCA()
pca.fit(X_train)

pca_df = pd.DataFrame({"expl_var" : pca.explained_variance_ratio_, "pc":␣
↪range(1,7), "cum_var": pca.explained_variance_ratio_.cumsum()})

(ggplot(pca_df, aes(x = "pc", y = "expl_var")) + geom_line() + geom_point())␣
↪+theme_minimal()
```



[ ]: <ggplot: (8763695481141)>

[ ]:
```
(ggplot(pca_df, aes(x = "pc", y = "cum_var")) + geom_line() +
 geom_point() + geom_hline(yintercept = 0.90, color = "red"))+ theme_minimal()
```



[ ]: <ggplot: (8763693384501)>

From the PCA graph we see that we can use 4 variables and still maintain 90% of the variance. To figure out which variables to keep and which to remove lets look at a Lasso Model.

[ ]:
```
lsr = Lasso(alpha = 0.2)

lsr.fit(X_train,y_train)

coefficients_lsr = pd.DataFrame({"Coefficients":lsr.coef_,"Name":predictors})
print("TEST : ", r2_score(y_test, lsr.predict(X_test)))
coefficients_lsr
```

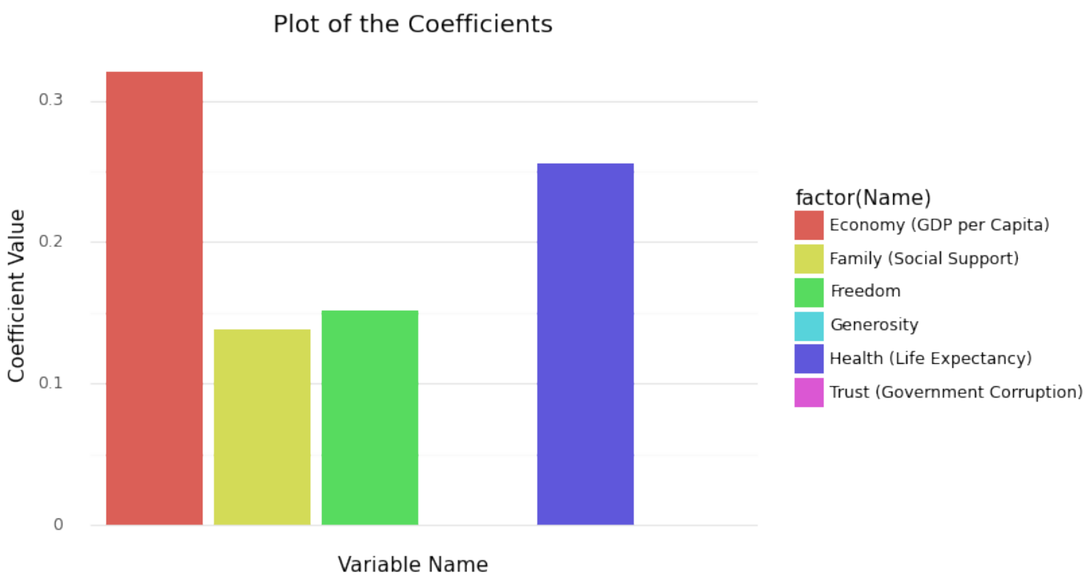
TEST : 0.6877757450812187

[ ]:

	Coefficients	Name
0	0.320583	Economy (GDP per Capita)
1	0.138617	Family (Social Support)
2	0.255362	Health (Life Expectancy)
3	0.152229	Freedom
4	0.000000	Trust (Government Corruption)

```
5      0.000000                         Generosity
```

```
[ ]: (ggplot(coefficients_lsr, aes(x = "Name", y = "Coefficients", fill =␣
      ↪"factor(Name)")) + geom_bar(stat = "identity") + theme_minimal() + \
      ggtitle("Plot of the Coefficients") + labs(x = "Variable Name", y =␣
      ↪"Coefficient Value") + \
      theme(panel_grid_major_x = element_blank(), axis_text_x = element_blank()))
```




```
[ ]: <ggplot: (8763693510713)>
```

```
[ ]: print("TRAIN: ", r2_score(y_train, lsr.predict(X_train)))
     print("TEST : ", r2_score(y_test, lsr.predict(X_test)))
```

```
TRAIN:  0.6713629522667822
TEST :  0.6877757450812187
```

Looking at the coefficients we can see that two variables shurnk down to 0 and to further understand how removing these variables from our model I will remake the model.

```
[ ]: #set the variables
     predictors2 = ["Economy (GDP per Capita)","Family (Social Support)",
                    "Health (Life Expectancy)","Freedom"]
     X2 = data[predictors]
     y2 = data["Happiness Score"]
```

```
[ ]: #creating a linear regression model
     X2_train, X2_test, y2_train, y2_test = train_test_split(X2,y2, test_size = 0.2)
```

```python
z2 = StandardScaler()
z2.fit(X2_train)

X2_train = z2.transform(X2_train)
X2_test = z2.transform(X2_test)

lr2 = LinearRegression()
lr2.fit(X2_train, y2_train)
```

[ ]: LinearRegression()

[ ]:
```python
#getting the different predictions
y2_train_preds = lr2.predict(X2_train)
#error_train = y_train - y_train_preds
assump2_train = pd.DataFrame({"predicted":y2_train_preds,"actual":y2_train})

y2_test_preds = lr2.predict(X2_test)
#error_test = y_test - y_test_preds
assump2_test = pd.DataFrame({"predicted":y2_test_preds,"actual":y2_test})
```

[ ]:
```python
#model validation
print("For the Train Set")
print("MSE:",mean_squared_error(y2_train,y2_train_preds))
print("R^2:",r2_score(y2_train,y2_train_preds))

print("For the Test Set")
print("MSE:",mean_squared_error(y2_test,y2_test_preds))
print("R^2:",r2_score(y2_test,y2_test_preds))
```

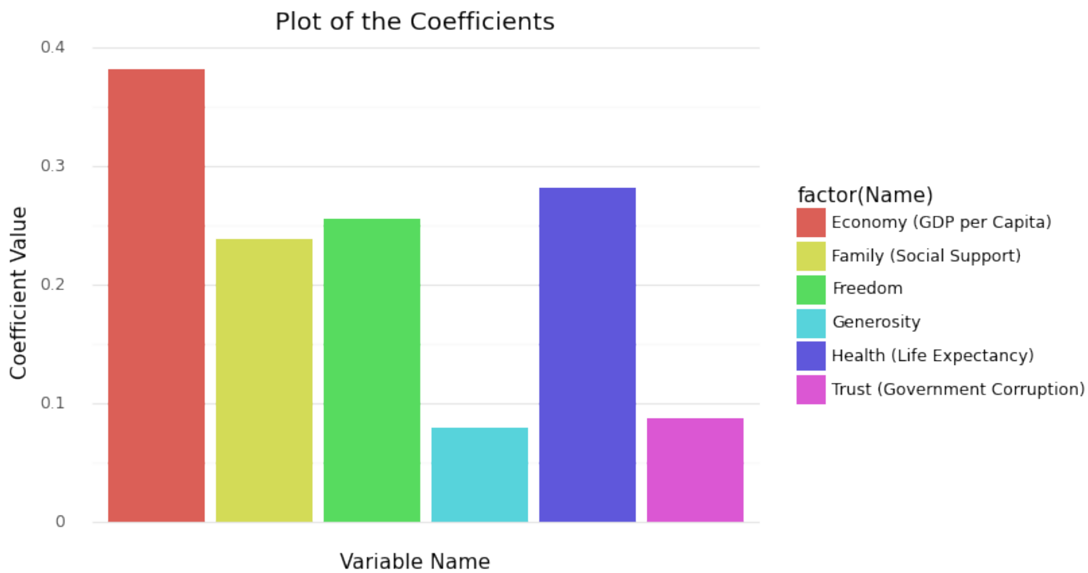
```
For the Train Set
MSE: 0.3252028642442948
R^2: 0.7389343974637071
For the Test Set
MSE: 0.2978360949625567
R^2: 0.7577638136508775
```

[ ]:
```python
coefficients2 = pd.DataFrame({"Coefficients":lr2.coef_,"Name":predictors})
coefficients2
```

[ ]:

	Coefficients	Name
0	0.382235	Economy (GDP per Capita)
1	0.239223	Family (Social Support)
2	0.281807	Health (Life Expectancy)
3	0.256486	Freedom
4	0.088117	Trust (Government Corruption)
5	0.079808	Generosity

[ ]:
```
(ggplot(coefficients2, aes(x = "Name", y = "Coefficients", fill =
↪"factor(Name)")) + geom_bar(stat = "identity") + theme_minimal() + \
ggtitle("Plot of the Coefficients") + labs(x = "Variable Name", y =
↪"Coefficient Value") + \
theme(panel_grid_major_x = element_blank(), axis_text_x = element_blank()))
```



[ ]: <ggplot: (8763692495297)>

Answer for Question 2

The results from our PCA tells us that we can use 4 variables instead of 6 but still manage to keep 90% of the variance in our data. To check which variables we could keep we looked at a Lasso Model.

The results of the Lasso Model showed us two variables that completely shrunk down to 0 which are trust and generosity, which are the same variables we identified as removable in Question 1.

I also remade the regression model using just the four variables to see how the model is affected. When comparing the results for both we see that the MSE values are around the same and the R^2 value only decreased a little from 75% to 73% which is expected because we have fewer variables. Overall the model is performing the same so we can say trust and generosity can be removed.

Question 3

When considering the three most influential variables, in our case Economy, Health and Freedom, what kind of clusters do we get, and what conclusions can we draw about the characteristics of those clusters? Can we factor in regions to further expand the model? What kind of pattern do we see between the two graphs (one where we use the clusters as the factor and one where we use regions as the factor) for each of the variables?

[ ]:
```
print(ggplot(data, aes(x = "Economy (GDP per Capita)", y = "Health (Life␣
↪Expectancy)")) + geom_point() + theme_minimal() + \
ggtitle("Economy VS Health.") + labs(x = "Economy", y = "Health") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```


```
<ggplot: (8763695326681)>
```

[ ]:
```
print(ggplot(data, aes(x = "Economy (GDP per Capita)", y = "Freedom")) +␣
↪geom_point() + theme_minimal() + \
ggtitle("Economy VS Freedom.") + labs(x = "Economy", y = "Freedom") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```

<ggplot: (8763692680181)>

```
[ ]: print(ggplot(data, aes(x = "Health (Life Expectancy)", y = "Freedom")) +␣
     ↪geom_point() + theme_minimal() + \
     ggtitle("Health VS Freedom.") + labs(x = "Health", y = "Freedom") + \
     theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```

<ggplot: (8763692934713)>

I plotted each of the variables against each other to identify which clustering method would be the best to use.

KMeans would not work as well because we don't really see any spherical clusters and the outcome would most liekly be 1 cluster. DBSCAN won't work that well either because we don't really have areas of different densities and there is a lot of overlap with the data points. Therefore, Gaussian Mixture Models would be the best to use because we are not restricted to spherical clusters.

[ ]:
```
#Gaussian
features = ["Economy (GDP per Capita)", "Health (Life Expectancy)", "Freedom"]
X3 = data[features]

z3 = StandardScaler()

X3[features] = z3.fit_transform(X3)
```

[ ]:
```
#Choosing a value for n_components
n_components = [2,3,4,5,6,7]

sils = []
```

```
for n in n_components:
    EM = GaussianMixture(n_components = n)
    EM.fit(X3)

    cluster = EM.predict(X3)
    data["cluster"] = cluster

    sils.append(silhouette_score(X3, cluster))

print(sils)
```

```
[0.3643163047054758, 0.3237486259817478, 0.3332865584492677,
0.32300917577145305, 0.27187948772901, 0.2601797793874351]
```

To pick n_components we looked at the different silhouetter scores and picked the best one.

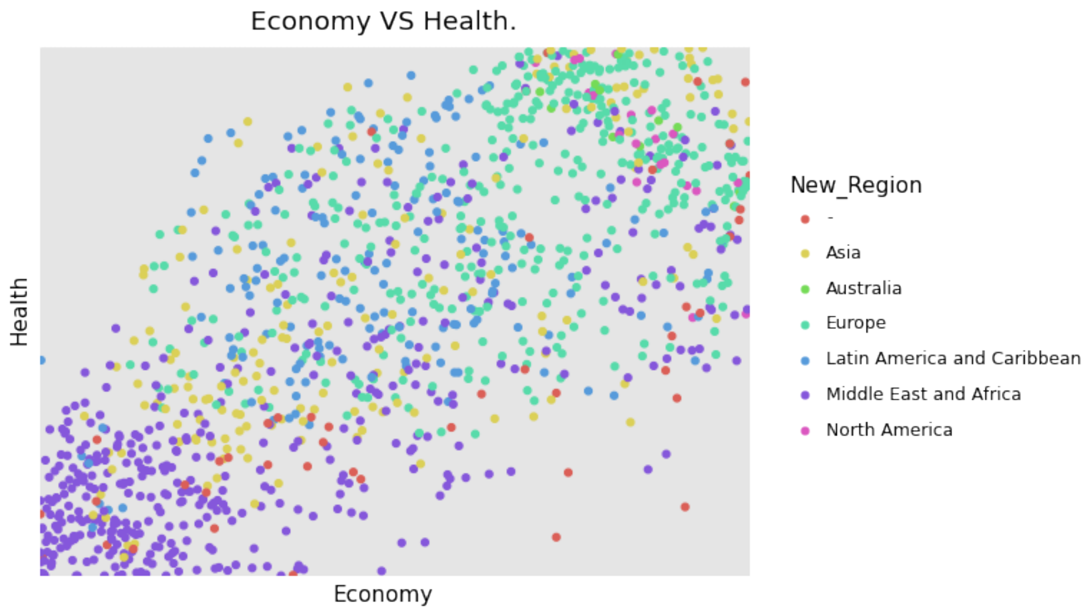
```
[ ]: #Using n_components based on highest silhouette score
EM = GaussianMixture(n_components = 2)

EM.fit(X3)

cluster = EM.predict(X3)
data["cluster"] = cluster

print("SILHOUETTE: ", silhouette_score(X3, cluster))
```

```
SILHOUETTE:  0.37368352494628476
```

```
[ ]: (ggplot(data, aes(x = "Economy (GDP per Capita)", y = "Health (Life␣
 ↪Expectancy)", color = "factor(cluster)")) + \
 geom_point() + theme_minimal() + ggtitle("Economy VS Health.") + labs(x =␣
 ↪"Economy", y = "Health") + \
 theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```

[ ]: <ggplot: (8763695549633)>

```
[ ]: (ggplot(data, aes(x = "Economy (GDP per Capita)", y = "Health (Life␣
     ↪Expectancy)", color = "New_Region")) + \
     geom_point() + theme_minimal() + ggtitle("Economy VS Health.") + labs(x =␣
     ↪"Economy", y = "Health") + \
     theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```

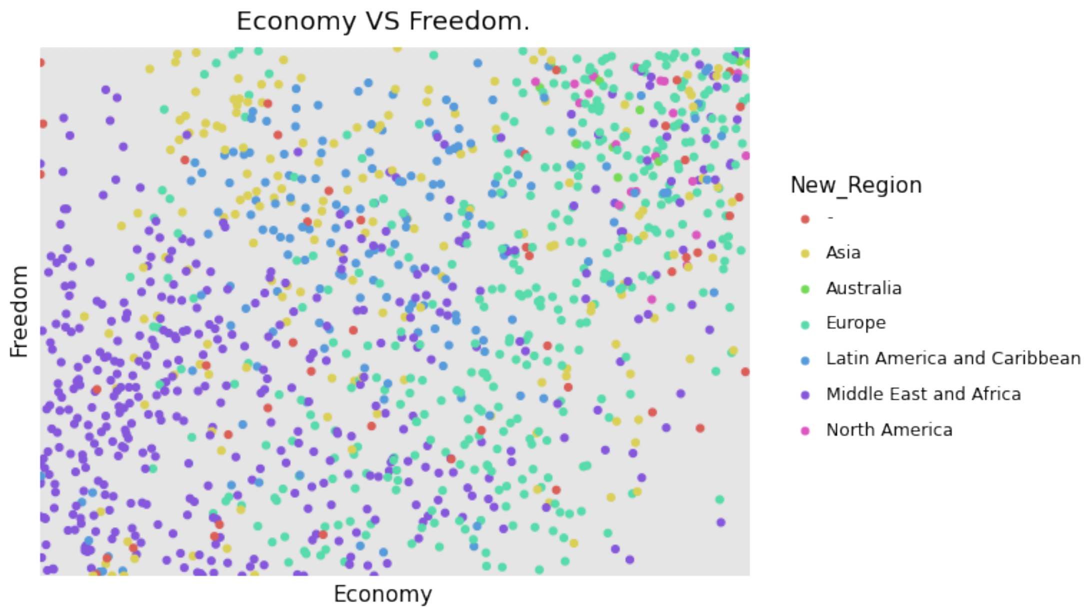



```
[ ]: <ggplot: (8763695325461)>
```

```
[ ]: (ggplot(data, aes(x = "Economy (GDP per Capita)", y = "Freedom", color =␣
     ↪"factor(cluster)")) + \
     geom_point() + theme_minimal() + ggtitle("Economy VS Freedom.") + labs(x =␣
     ↪"Economy", y = "Freedom") + \
     theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```

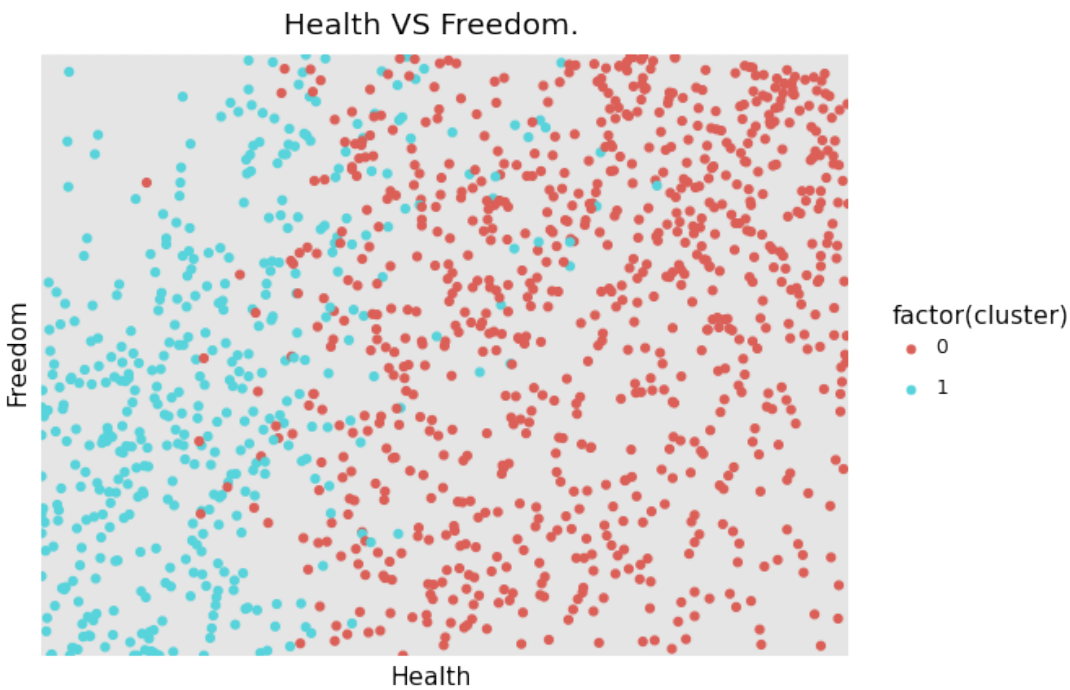
[ ]: <ggplot: (8763692355389)>

[ ]:
```
(ggplot(data, aes(x = "Economy (GDP per Capita)", y = "Freedom", color =␣
↪"New_Region")) + \
geom_point() + theme_minimal() + ggtitle("Economy VS Freedom.") + labs(x =␣
↪"Economy", y = "Freedom") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



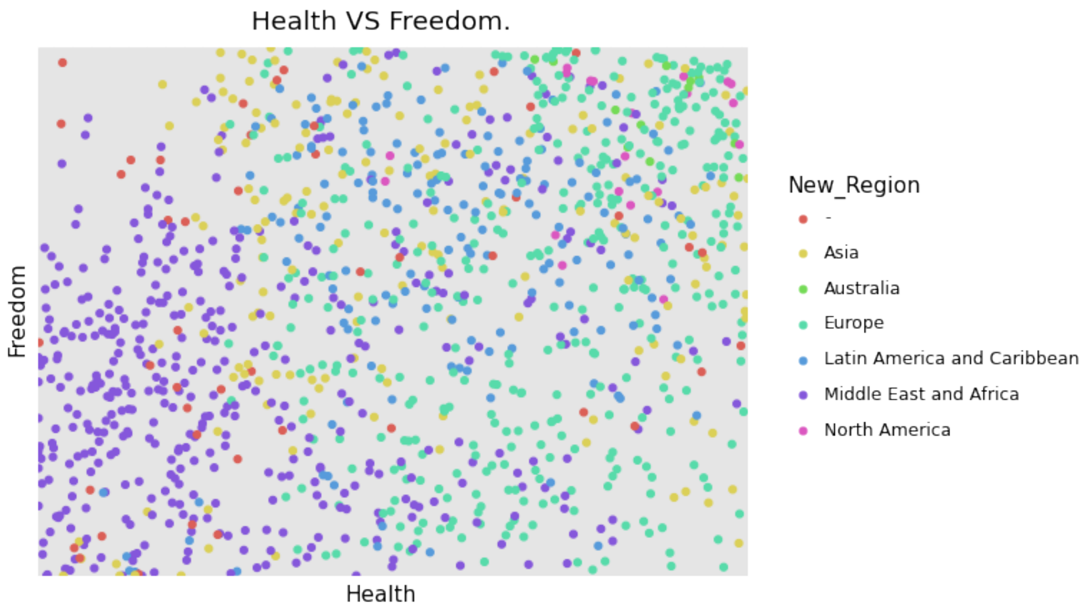
[ ]: <ggplot: (8763693671697)>

[ ]:
```
(ggplot(data, aes(x = "Health (Life Expectancy)", y = "Freedom", color =␣
↪"factor(cluster)")) + \
geom_point() + theme_minimal() + ggtitle("Health VS Freedom.") + labs(x =␣
↪"Health", y = "Freedom") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



[ ]: <ggplot: (8763697501653)>

[ ]:
```
(ggplot(data, aes(x = "Health (Life Expectancy)", y = "Freedom", color =␣
↪"New_Region")) + \
geom_point() + theme_minimal() + ggtitle("Health VS Freedom.") + labs(x =␣
↪"Health", y = "Freedom") + \
theme(axis_text_x = element_blank(), axis_text_y = element_blank()))
```



[ ]: <ggplot: (8763695448469)>

Answer for Question 3

The first graph is colored based on clusters and the following graph is colored based on regions and I did this for all three variables. All the clusters seem to overlap a bit and aren't very cohesive so our clustering is not the best.

First we have Economy vs Health. We see a clear divide in the data points with the top being a cluster and the bottom being one. We see that most middle east and african countries are in the blue cluster that tells us both health and economy are low so we could consider them to be LEDC. On the other hand we see a lot of European countries in the red cluster that tells us they are high in both variables hence could be MEDC. Asian countries are kinda all over the place which we see in the real world too as some are more economically developed than others.

Mext we have Economy vs Freedom. This time with the left being a cluster and the right being the other. When comparing it to the different regions, we see that most middle east and african countries are in the blue cluster that tells us they have low economy but vary in freedom. This could be due to wars and differences in governments. The red cluster has most of the remaining regions and so they vary a lot in both variables and this is probably due to differences in governments and their laws.

Lastly we have Health vs Freesdom. We see a clear divide in the data points with the left being a cluster and the right being the other just like the previous one. When comparing it to the different regions, we see that most middle east and african countries are in the blue cluster that tells us they are lower in health but still vary in freedom. This could be due to differences in governments and conflicts between them. The red cluster has most of the remaining regions and so they vary a lot in both variables and this is probably due to differences in governments and access to healthcare.