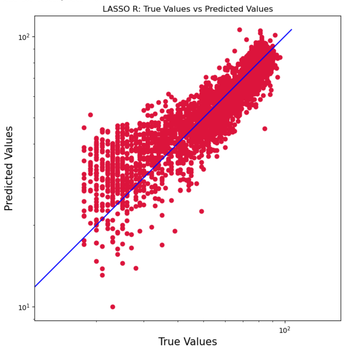
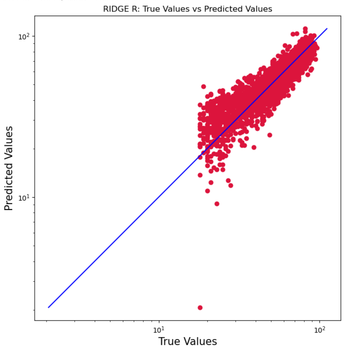
In [2]:
```
#Lasso Regression
import numpy as np  # Importing NumPy library
import pandas as pd  # Importing Pandas library
import matplotlib.pyplot as plt  # Importing Matplotlib library's "pyplot" module
import seaborn as sns  # Importing Seaborn library
import os
from sklearn.model_selection import train_test_split



from sklearn.linear_model import Lasso
data = pd.read_csv("/Users/catherinebetancourt-lee/BMEN 415/Volumetric_features.csv")
y = data['Age']
X = data.drop(['Age'], axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2, random_state=42)
lasso_model = Lasso(alpha=0.1)
lasso_model.fit(X_train, y_train)
y_pred = lasso_model.predict(X_test)
from sklearn.model_selection import cross_val_predict  # For K-Fold Cross Validation
from sklearn.metrics import r2_score  # For find accuracy with R2 Score
from sklearn.metrics import mean_squared_error  # For MSE
from math import sqrt  # For squareroot operation

y_pred_train = lasso_model.predict(X_train)
y_pred_test = lasso_model.predict(X_test)

accuracy_train = r2_score(y_train, y_pred_train)
print("Training R2 for Multiple Linear Regression Model: ", accuracy_train)

accuracy_test = r2_score(y_test, y_pred_test)
print("Testing R2 for Multiple Linear Regression Model: ", accuracy_test)

RMSE_train = sqrt(mean_squared_error(y_train, y_pred_train))
print("RMSE for Training Data: ", RMSE_train)

RMSE_test = sqrt(mean_squared_error(y_test, y_pred_test))
print("RMSE for Testing Data: ", RMSE_test)

true_val = y_train
pred_val = y_pred_train

plt.figure(figsize=(8,8))
plt.scatter(true_val, pred_val, c='crimson')
plt.yscale('log')
plt.xscale('log')

p1 = max(max(pred_val), max(true_val))
p2 = min(min(pred_val), min(true_val))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('True Values', fontsize=15)
plt.ylabel('Predicted Values', fontsize=15)
plt.title('LASSO R: True Values vs Predicted Values')
plt.axis('equal')
plt.show()
```

```
/Users/catherinebetancourt-lee/opt/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_coordinate_descent.py:647: ConvergenceWarning: Objective did not converge
. You might want to increase the number of iterations, check the scale of the features or consider increasing regularisation. Duality gap: 1.154e+05, tolerance: 1.353e+
02
 model = cd_fast.enet_coordinate_descent(
Training R2 for Multiple Linear Regression Model:  0.8340190793971338
Testing R2 for Multiple Linear Regression Model:  0.8380236251281098
RMSE for Training Data:  8.151095391734291
RMSE for Testing Data:  8.150853867092914
```



In [3]:
```
#Ridge Regression
import numpy as np  # Importing NumPy library
import pandas as pd  # Importing Pandas library
import matplotlib.pyplot as plt  # Importing Matplotlib library's "pyplot" module
import seaborn as sns  # Importing Seaborn library
import os
from sklearn.model_selection import train_test_split

from sklearn.linear_model import Ridge
data = pd.read_csv("/Users/catherinebetancourt-lee/BMEN 415/Volumetric_features.csv")
y = data['Age']
X = data.drop(['Age'], axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2, random_state=42)
ridge_model = Ridge(alpha=0.1)
ridge_model.fit(X_train, y_train)
y_pred = ridge_model.predict(X_test)
from sklearn.model_selection import cross_val_predict  # For K-Fold Cross Validation
from sklearn.metrics import r2_score  # For find accuracy with R2 Score
from sklearn.metrics import mean_squared_error  # For MSE
from math import sqrt  # For squareroot operation

y_pred_train = ridge_model.predict(X_train)
y_pred_test = ridge_model.predict(X_test)

accuracy_train = r2_score(y_train, y_pred_train)
print("Training R2 for Multiple Linear Regression Model: ", accuracy_train)

accuracy_test = r2_score(y_test, y_pred_test)
print("Testing R2 for Multiple Linear Regression Model: ", accuracy_test)

RMSE_train = sqrt(mean_squared_error(y_train, y_pred_train))
print("RMSE for Training Data: ", RMSE_train)

RMSE_test = sqrt(mean_squared_error(y_test, y_pred_test))
print("RMSE for Testing Data: ", RMSE_test)

true_val = y_train
pred_val = y_pred_train

plt.figure(figsize=(8,8))
plt.scatter(true_val, pred_val, c='crimson')
plt.yscale('log')
plt.xscale('log')

p1 = max(max(pred_val), max(true_val))
p2 = min(min(pred_val), min(true_val))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('True Values', fontsize=15)
plt.ylabel('Predicted Values', fontsize=15)
plt.title('RIDGE R: True Values vs Predicted Values')
plt.axis('equal')
plt.show()
```

```
/Users/catherinebetancourt-lee/opt/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_ridge.py:157: LinAlgWarning: Ill-conditioned matrix (rcond=7.09322e-17):
result may not be accurate.
  return linalg.solve(A, Xy, sym_pos=True, overwrite_a=True).T
Training R2 for Multiple Linear Regression Model:  0.8621413685080541
Testing R2 for Multiple Linear Regression Model:  0.8569246435683536
RMSE for Training Data:  7.428546952089393
RMSE for Testing Data:  7.660545346312268
```



In [4]:
```
#Elastic Net Regression
import numpy as np  # Importing NumPy library
import pandas as pd  # Importing Pandas library
import matplotlib.pyplot as plt  # Importing Matplotlib library's "pyplot" module
import seaborn as sns  # Importing Seaborn library
import os
from sklearn.model_selection import train_test_split

from sklearn.linear_model import ElasticNet
data = pd.read_csv("/Users/catherinebetancourt-lee/BMEN 415/Volumetric_features.csv")
y = data['Age']
X = data.drop(['Age'], axis = 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2, random_state=42)
elastic_model = ElasticNet(alpha=0.1, l1_ratio=0.5)
elastic_model.fit(X_train, y_train)
y_pred = elastic_model.predict(X_test)
from sklearn.model_selection import cross_val_predict  # For K-Fold Cross Validation
from sklearn.metrics import r2_score  # For find accuracy with R2 Score
from sklearn.metrics import mean_squared_error  # For MSE
from math import sqrt  # For squareroot operation

y_pred_train = elastic_model.predict(X_train)
y_pred_test = elastic_model.predict(X_test)

accuracy_train = r2_score(y_train, y_pred_train)
print("Training R2 for Multiple Linear Regression Model: ", accuracy_train)

accuracy_test = r2_score(y_test, y_pred_test)
print("Testing R2 for Multiple Linear Regression Model: ", accuracy_test)

RMSE_train = sqrt(mean_squared_error(y_train, y_pred_train))
print("RMSE for Training Data: ", RMSE_train)

RMSE_test = sqrt(mean_squared_error(y_test, y_pred_test))
print("RMSE for Testing Data: ", RMSE_test)

true_val = y_train
pred_val = y_pred_train

plt.figure(figsize=(8,8))
plt.scatter(true_val, pred_val, c='crimson')
plt.yscale('log')
plt.xscale('log')

p1 = max(max(pred_val), max(true_val))
p2 = min(min(pred_val), min(true_val))
plt.plot([p1, p2], [p1, p2], 'b-')
plt.xlabel('True Values', fontsize=15)
plt.ylabel('Predicted Values', fontsize=15)
plt.title('ELASTIC NET: True Values vs Predicted Values')
plt.axis('equal')
plt.show()
```

```
/Users/catherinebetancourt-lee/opt/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_coordinate_descent.py:647: ConvergenceWarning: Objective did not converge
. You might want to increase the number of iterations, check the scale of the features or consider increasing regularisation. Duality gap: 1.146e+05, tolerance: 1.353e+
02
 model = cd_fast.enet_coordinate_descent(
Training R2 for Multiple Linear Regression Model:  0.8350199509390727
Testing R2 for Multiple Linear Regression Model:  0.8394392528731143
RMSE for Training Data:  8.126482517040913
RMSE for Testing Data:  8.115157622397632
```

In [ ]: