In [1]:

```python
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
```

In [2]:

```python
#importing SVM Model
from sklearn.svm import SVR
```

In [3]:

```python
data = pd.read_csv("/Users/catherinebetancourt-lee/BMEN 415/Volumetric_features.csv")

data
```

Out[3]:

	S.No	Left-Lateral-Ventricle	Left-Inf-Lat-Vent	Left-Cerebellum-White-Matter	Left-Cerebellum-Cortex	Left-Thalamus	Left-Caudate	Left-Putamen	Left-Pallidum	3rd-Ventricle	...	rh_supramarginal_thickness	rh_frontalpole_thickness	rh_temporalpole_thickness	rh_transv
0	1	22916.9	982.7	15196.7	55796.4	6855.5	2956.4	4240.7	2223.9	2034.4	...	2.408	2.629	3.519	
1	2	22953.2	984.5	15289.7	55778.6	6835.1	3064.2	4498.6	2354.1	1927.1	...	2.417	2.640	3.488	
2	3	23320.4	1062.1	15382.1	55551.2	7566.0	3231.7	4456.2	1995.4	2064.7	...	2.374	2.601	3.342	
3	4	24360.0	1000.5	14805.4	54041.8	8004.6	3137.3	4262.2	1983.4	2017.7	...	2.366	2.639	3.361	
4	5	25769.4	1124.4	16331.1	54108.6	6677.4	2964.4	4204.6	2409.7	2251.8	...	2.381	2.555	3.450	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4221	4222	27065.6	532.4	12425.1	51042.9	6354.8	3822.6	4490.5	2019.4	1256.2	...	2.505	2.666	2.915	
4222	4223	28408.8	912.7	14024.8	43103.5	6060.7	3114.2	3731.0	1937.4	1669.9	...	2.385	3.008	3.572	
4223	4224	34467.9	1659.6	12744.5	54924.8	6256.7	3573.4	3526.6	2189.9	3063.1	...	2.028	2.995	3.706	
4224	4225	31627.5	1334.4	15883.2	57148.2	6982.4	4475.8	4464.4	2317.8	3809.0	...	2.491	2.865	3.456	
4225	4226	14879.4	704.2	11346.6	50468.5	6935.4	3258.5	3751.5	2226.5	1898.4	...	2.474	3.150	3.691	

4226 rows × 141 columns

In [4]:

```python
#Separating target variable and features
y = data['rh_supramarginal_thickness']
X = data.drop(['rh_supramarginal_thickness'], axis = 1)
X
```

Out[4]:

	S.No	Left-Lateral-Ventricle	Left-Inf-Lat-Vent	Left-Cerebellum-White-Matter	Left-Cerebellum-Cortex	Left-Thalamus	Left-Caudate	Left-Putamen	Left-Pallidum	3rd-Ventricle	...	rh_superiortemporal_thickness	rh_frontalpole_thickness	rh_temporalpole_thickness	rh_tra
0	1	22916.9	982.7	15196.7	55796.4	6855.5	2956.4	4240.7	2223.9	2034.4	...	2.648	2.629	3.519	
1	2	22953.2	984.5	15289.7	55778.6	6835.1	3064.2	4498.6	2354.1	1927.1	...	2.660	2.640	3.488	
2	3	23320.4	1062.1	15382.1	55551.2	7566.0	3231.7	4456.2	1995.4	2064.7	...	2.597	2.601	3.342	
3	4	24360.0	1000.5	14805.4	54041.8	8004.6	3137.3	4262.2	1983.4	2017.7	...	2.604	2.639	3.361	
4	5	25769.4	1124.4	16331.1	54108.6	6677.4	2964.4	4204.6	2409.7	2251.8	...	2.597	2.555	3.450	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4221	4222	27065.6	532.4	12425.1	51042.9	6354.8	3822.6	4490.5	2019.4	1256.2	...	2.457	2.666	2.915	
4222	4223	28408.8	912.7	14024.8	43103.5	6060.7	3114.2	3731.0	1937.4	1669.9	...	2.497	3.008	3.572	
4223	4224	34467.9	1659.6	12744.5	54924.8	6256.7	3573.4	3526.6	2189.9	3063.1	...	2.407	2.995	3.706	
4224	4225	31627.5	1334.4	15883.2	57148.2	6982.4	4475.8	4464.4	2317.8	3809.0	...	2.700	2.865	3.456	
4225	4226	14879.4	704.2	11346.6	50468.5	6935.4	3258.5	3751.5	2226.5	1898.4	...	2.746	3.150	3.691	

4226 rows × 140 columns

In [5]:

```python
#training and testing datasets
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2, random_state=42)
```

In [6]:

```python
#Create and fit in a SVM regression model
svr_model = SVR(kernel='rbf', C=5.0)
svr_model.fit(X_train, y_train)
```

Out[6]:

```
SVR(C=5.0)
```

In [7]:

```python
#predict on the dataset
y_pred = svr_model.predict(X_test)
```

In [8]:

```python
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 = svr_model.predict(X_train)
y_pred_test = svr_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)
```

```
Training R2 for Multiple Linear Regression Model:  0.6290578198420312
Testing R2 for Multiple Linear Regression Model:  0.6415445394915742
RMSE for Training Data:  0.11167245415649324
RMSE for Testing Data:  0.1160629552873467
```

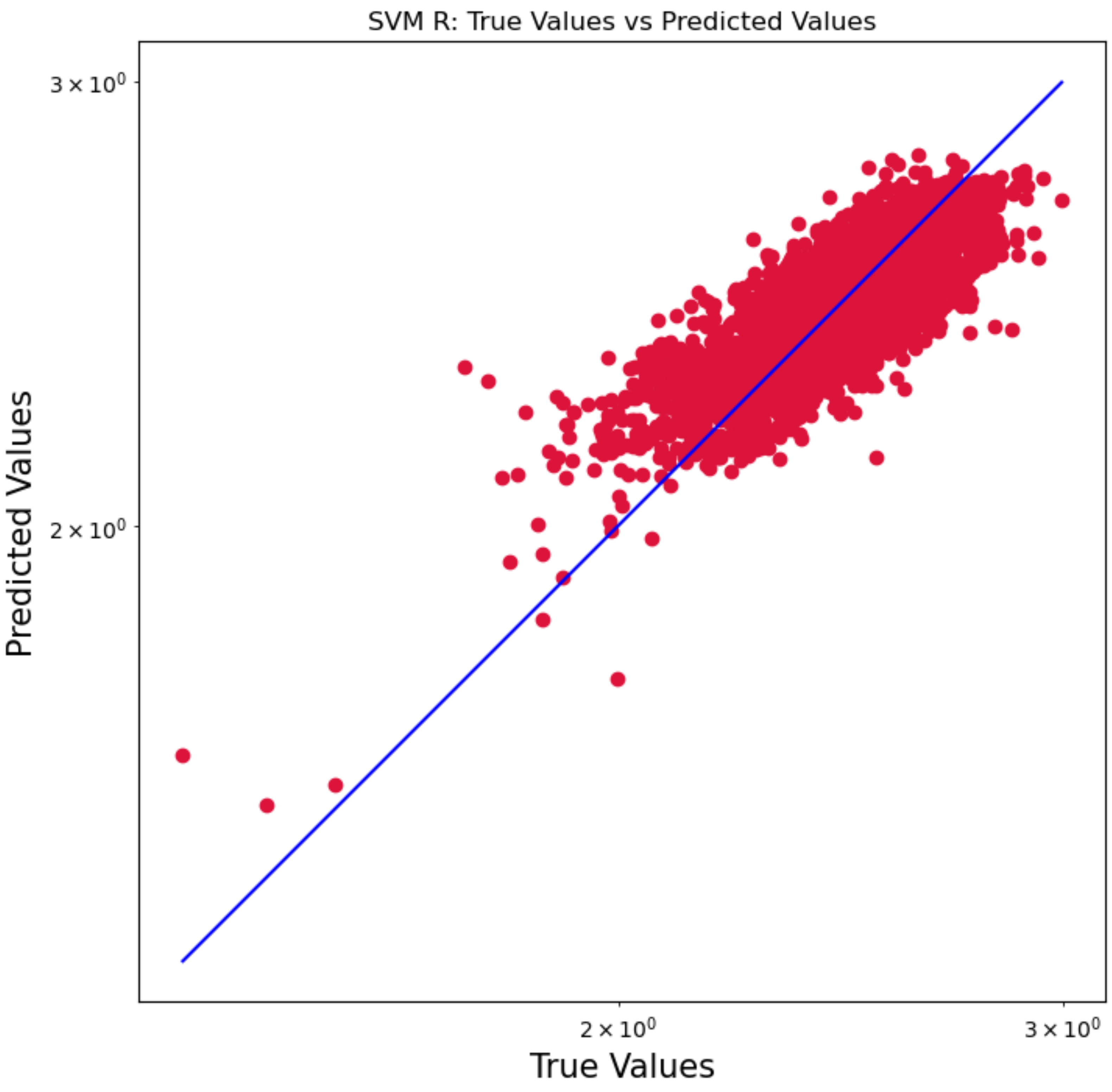
In [9]:

```python
true_val = y_train
pred_val = y_pred_train
```

In [10]:

```python
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("SVM R: True Values vs Predicted Values")
plt.axis('equal')
plt.show()
```


SVM R: True Values vs Predicted Values
3×10^0
2×10^0
Predicted Values
2×10^0
3×10^0
True Values