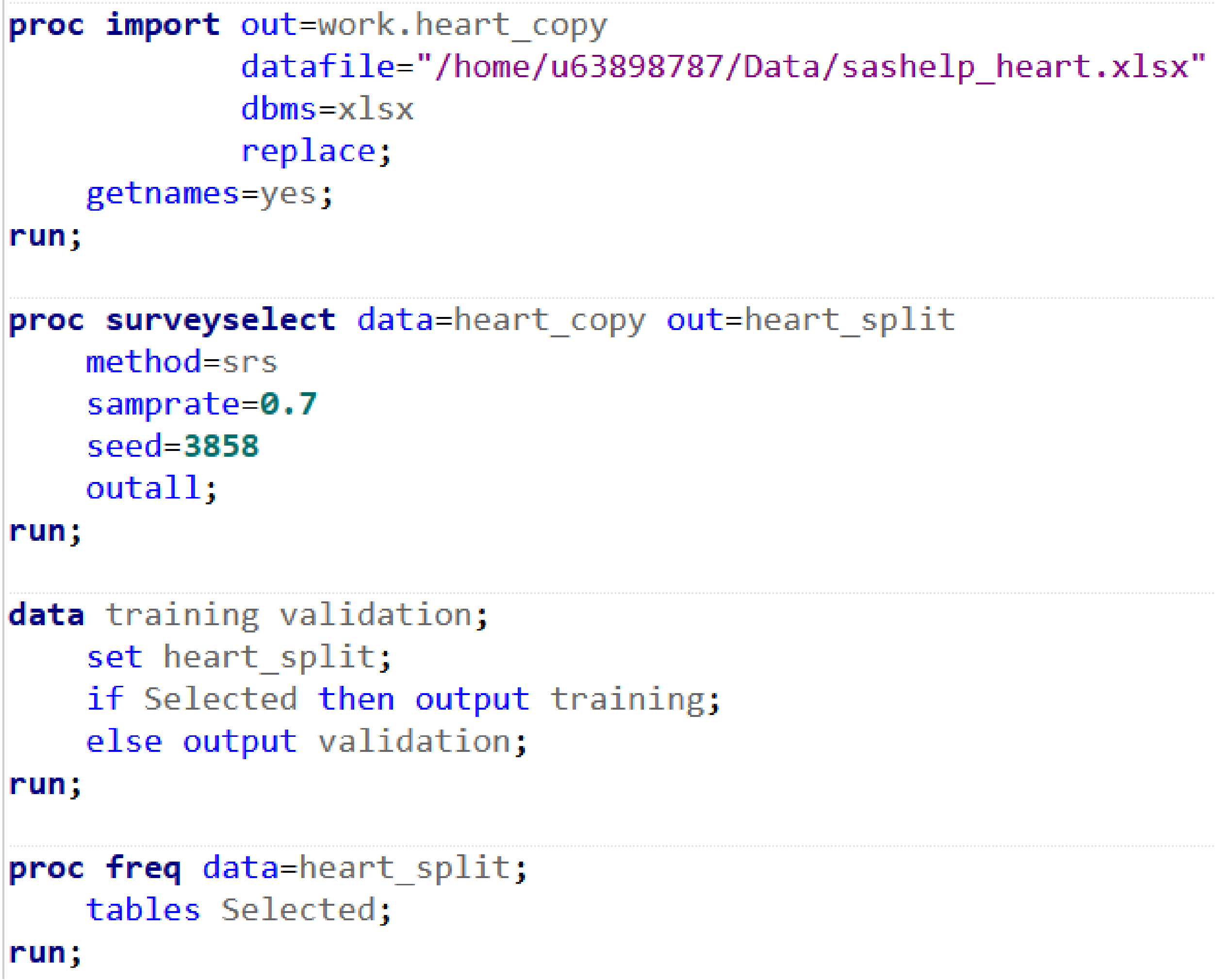
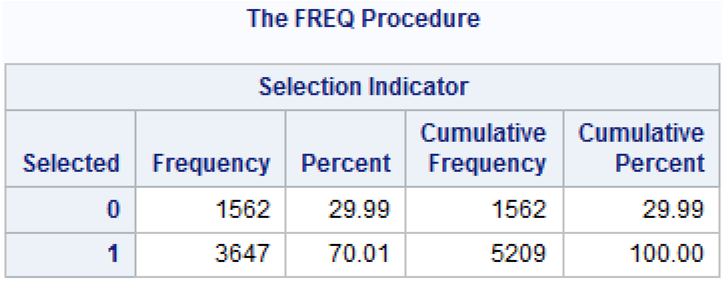
Part 1: Creating Training and Test Sets

SAS Code:



```
proc import out=work.heart_copy
            datafile="/home/u63898787/Data/sashelp_heart.xlsx"
            dbms=xlsx
            replace;
    getnames=yes;
run;

proc surveyselect data=heart_copy out=heart_split
    method=srs
    samprate=0.7
    seed=3858
    outall;
run;

data training validation;
    set heart_split;
    if Selected then output training;
    else output validation;
run;

proc freq data=heart_split;
    tables Selected;
run;
```

SAS Output:



The FREQ Procedure

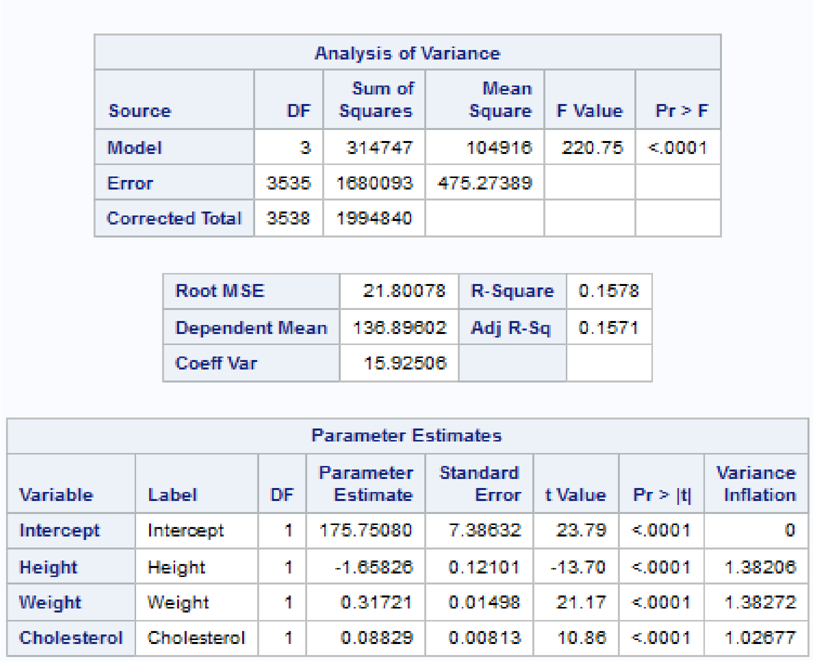
Selection Indicator				
Selected	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1562	29.99	1562	29.99
1	3647	70.01	5209	100.00

Part 2: Training Set Regression Model

SAS Code:

```
proc reg data=training;
    model systolic = height weight cholesterol / vif;
    output out=reg_out p=Predicted r=Residual student=StudentRes cookd=CookD;
run;
```

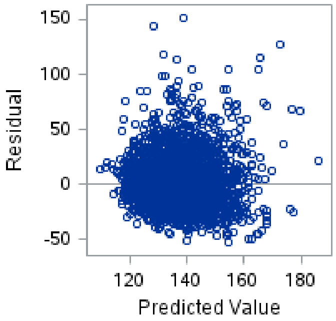
SAS Output:

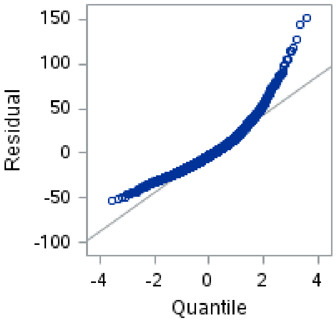


Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	314747	104916	220.75	<.0001
Error	3535	1680093	475.27389		
Corrected Total	3538	1994840			

Root MSE	21.80078	R-Square	0.1578
Dependent Mean	136.89602	Adj R-Sq	0.1571
Coeff Var	15.92506		

Parameter Estimates							
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	Intercept	1	175.75080	7.38632	23.79	<.0001	0
Height	Height	1	-1.65826	0.12101	-13.70	<.0001	1.38206
Weight	Weight	1	0.31721	0.01498	21.17	<.0001	1.38272
Cholesterol	Cholesterol	1	0.08829	0.00813	10.86	<.0001	1.02677

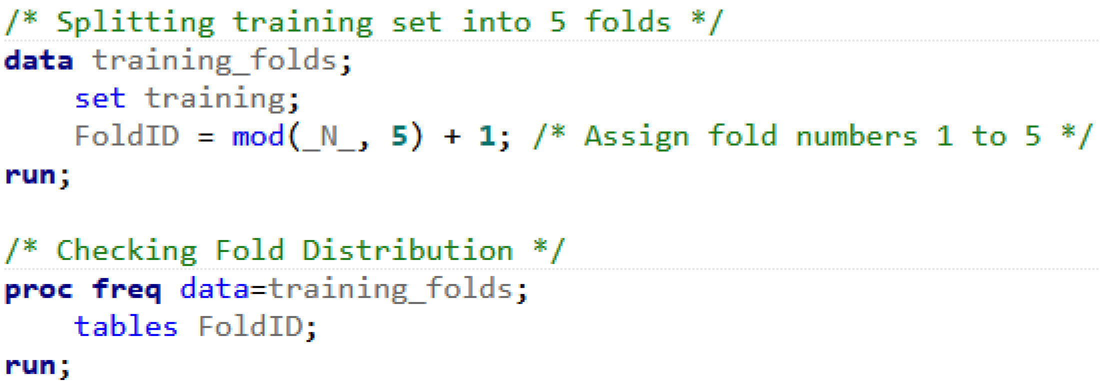




- The model as a whole and parameter estimates are statistically significant, as shown by the p-values, high F-statistic (for the model) and t-statistic values (for the parameters).
- All VIF values suggest low levels of collinearity.
- Residual plots suggest the normality assumption of the data may need transformations.
- Regression model equation: S = 175.74 - 1.66H + 0.32W + 0.09C, where:
  - S is systolic (response variable)
  - H is Height (predictor)
  - W is weight (predictor)
  - C is Cholesterol (predictor)

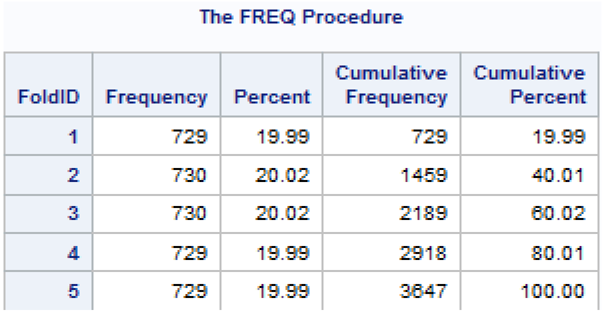
Part 3: 5-fold Cross Validation

SAS Code (creating folds):



```
/* Splitting training set into 5 folds */
data training_folds;
    set training;
    FoldID = mod(_N_, 5) + 1; /* Assign fold numbers 1 to 5 */
run;

/* Checking Fold Distribution */
proc freq data=training_folds;
    tables FoldID;
run;
```

SAS Output (fold distribution):



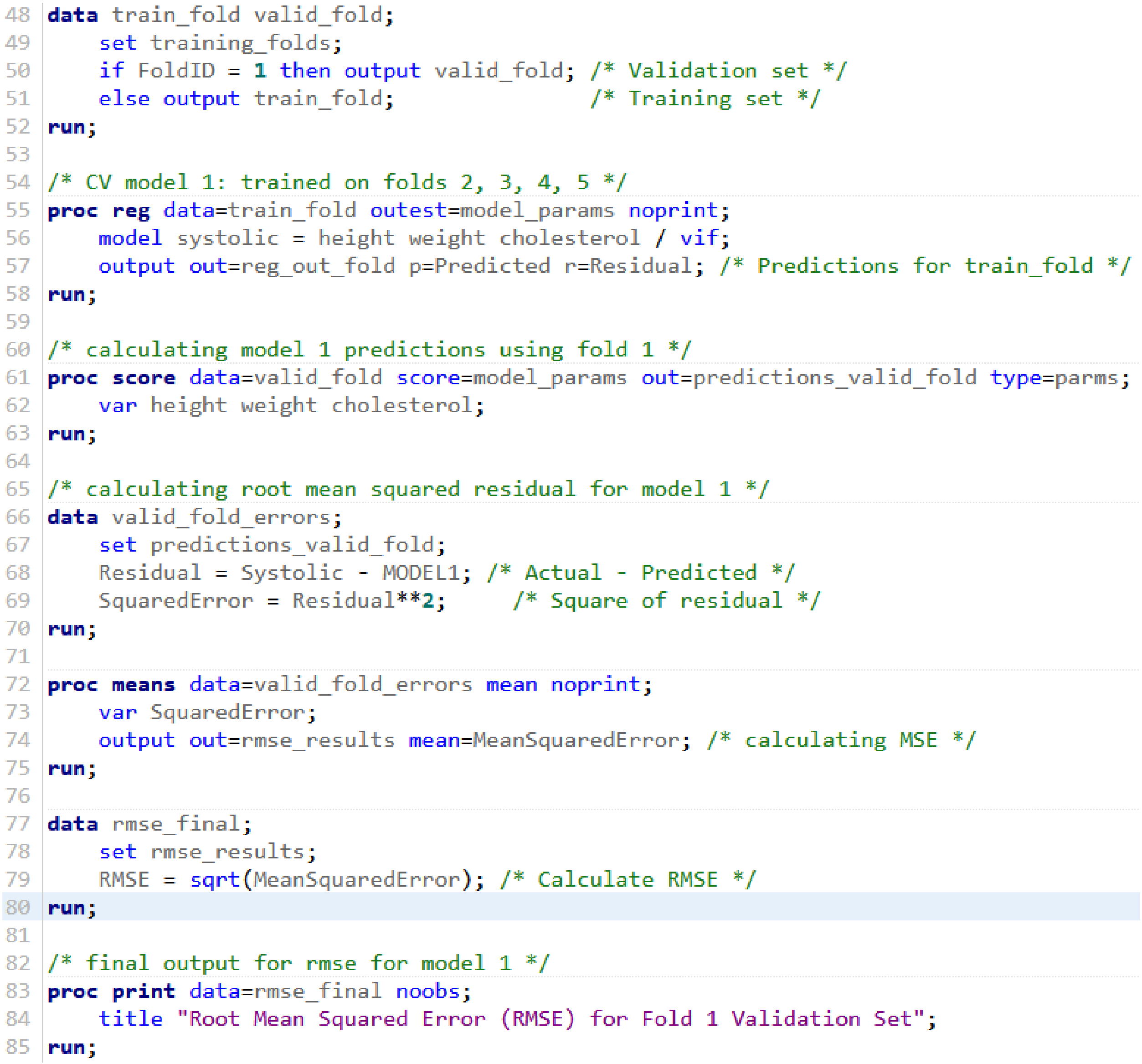
The FREQ Procedure

FoldID	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	729	19.99	729	19.99
2	730	20.02	1459	40.01
3	730	20.02	2189	60.02
4	729	19.99	2918	80.01
5	729	19.99	3647	100.00

Summary of Cross Validation steps used (code and output on following pages):

1. For each model, 1 fold is used as a validation set, and a model is created using the remaining 4 folds.
2. The created model is then tested using the validation set (fold that was left out).
3. Root mean squared error is calculated using the predictions in step 2.
4. RMSE will be compared across the 5 created models to determine the best model.

SAS Code (calculating root mean squared error for cross validation model):



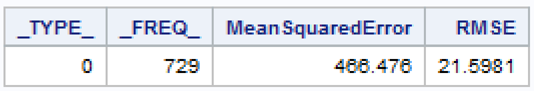
```
48 data train_fold valid_fold;
49     set training_folds;
50     if FoldID = 1 then output valid_fold; /* Validation set */
51     else output train_fold;               /* Training set */
52 run;
53 
54 /* CV model 1: trained on folds 2, 3, 4, 5 */
55 proc reg data=train_fold outest=model_params noprint;
56     model systolic = height weight cholesterol / vif;
57     output out=reg_out_fold p=Predicted r=Residual; /* Predictions for train_fold */
58 run;
59 
60 /* calculating model 1 predictions using fold 1 */
61 proc score data=valid_fold score=model_params out=predictions_valid_fold type=parms;
62     var height weight cholesterol;
63 run;
64 
65 /* calculating root mean squared residual for model 1 */
66 data valid_fold_errors;
67     set predictions_valid_fold;
68     Residual = Systolic - MODEL1; /* Actual - Predicted */
69     SquaredError = Residual**2;     /* Square of residual */
70 run;
71 
72 proc means data=valid_fold_errors mean noprint;
73     var SquaredError;
74     output out=rmse_results mean=MeanSquaredError; /* calculating MSE */
75 run;
76 
77 data rmse_final;
78     set rmse_results;
79     RMSE = sqrt(MeanSquaredError); /* Calculate RMSE */
80 run;
81 
82 /* final output for rmse for model 1 */
83 proc print data=rmse_final noobs;
84     title "Root Mean Squared Error (RMSE) for Fold 1 Validation Set";
85 run;
```

NOTES:

- For each model, the FoldID parameter is changed on line 50 (FoldID = 1, 2, 3, 4 ,5).
- The above code shows the code used to generate the first model (using folds 2, 3, 4, 5).
- The above code tests the created model using the first fold as a validation set.
- The typical prog reg outputs for CV models are hidden using the noprint keyword.
- On line 68, the model predictions are stored in a column labeled MODEL1.

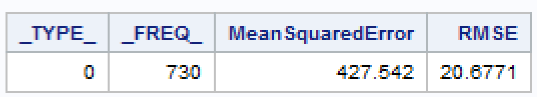
SAS Outputs:

Model 1:



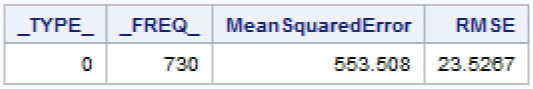
TYPE	_FREQ_	MeanSquaredError	RMSE
0	729	466.476	21.5981

Model 2:



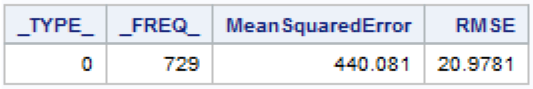
TYPE	_FREQ_	MeanSquaredError	RMSE
0	730	427.542	20.6771

Model 3:



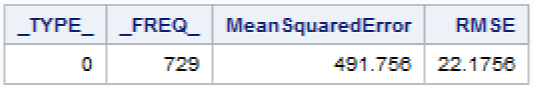
TYPE	_FREQ_	MeanSquaredError	RMSE
0	730	553.508	23.5267

Model 4:



TYPE	_FREQ_	MeanSquaredError	RMSE
0	729	440.081	20.9781

Model 5:



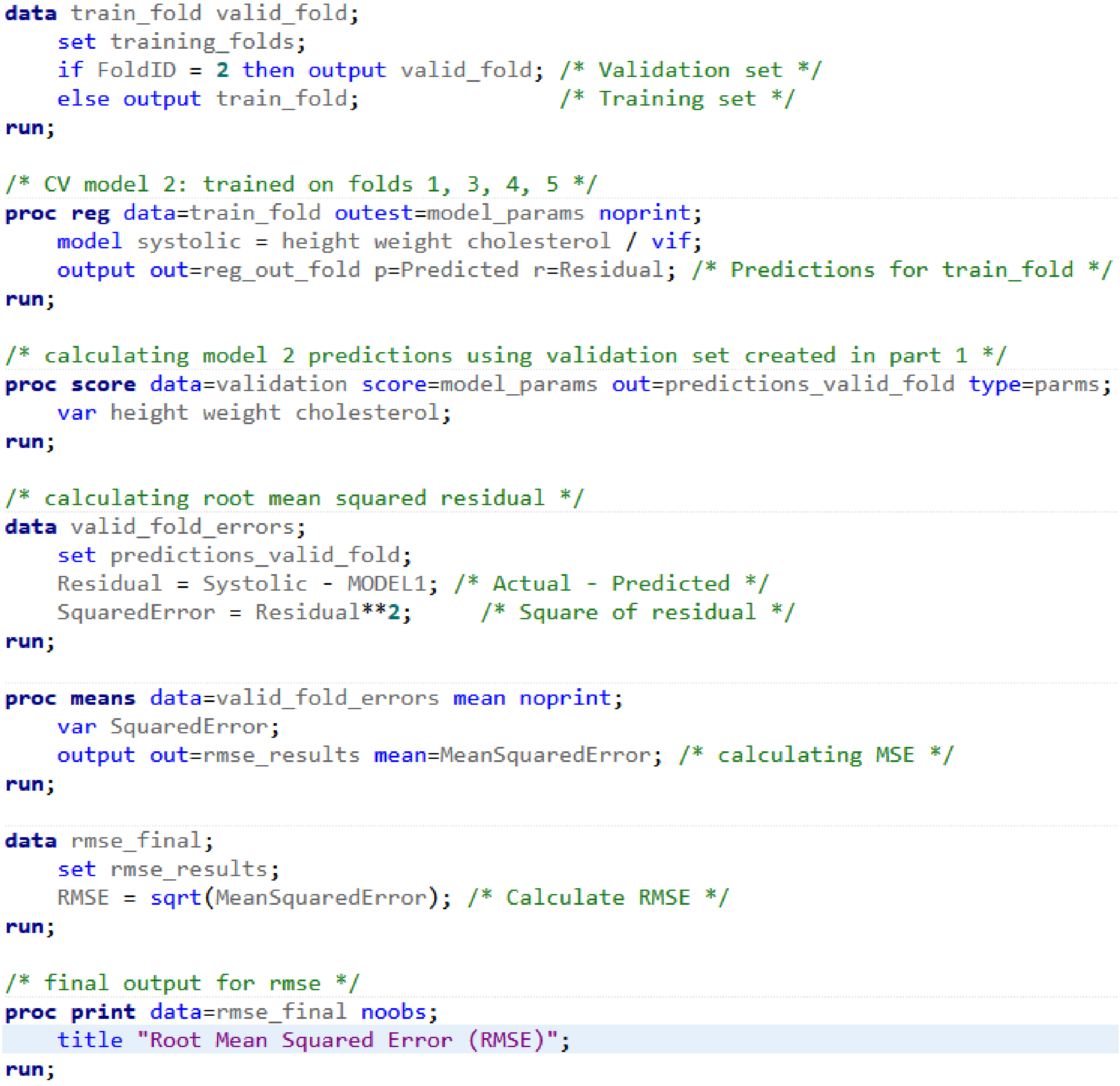
TYPE	_FREQ_	MeanSquaredError	RMSE
0	729	491.756	22.1756

Notes:

- Model 2 predictions resulted in the least RMSE out of all the models.
- Model 2 will be used on the validation step created in part 1.

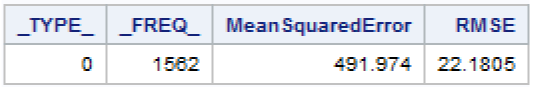
Part 4: Final Model performance

SAS Code (testing model 2 on validation set):



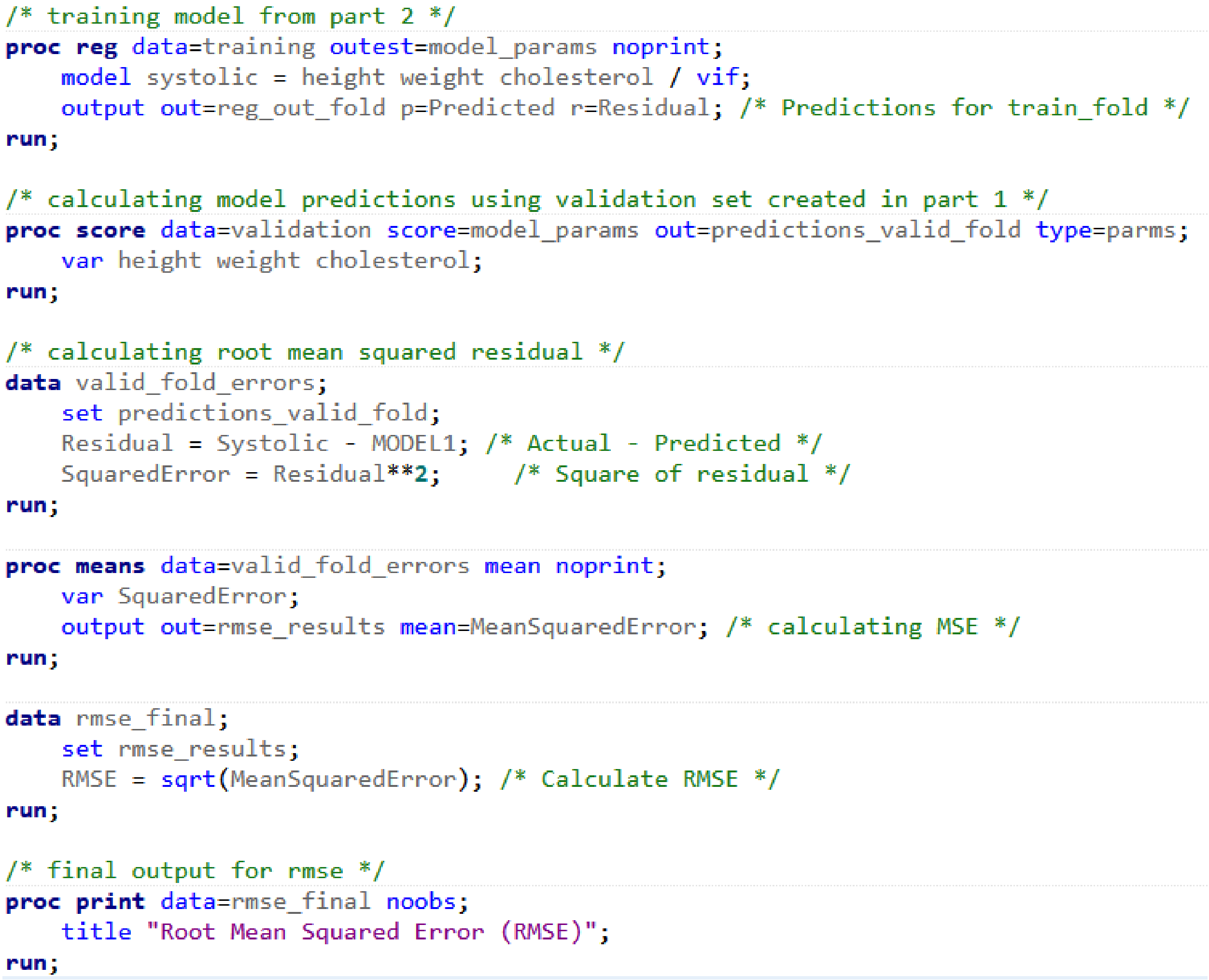
```
data train_fold valid_fold;
    set training_folds;
    if FoldID = 2 then output valid_fold; /* Validation set */
    else output train_fold;               /* Training set */
run;

/* CV model 2: trained on folds 1, 3, 4, 5 */
proc reg data=train_fold outest=model_params noprint;
    model systolic = height weight cholesterol / vif;
    output out=reg_out_fold p=Predicted r=Residual; /* Predictions for train_fold */
run;

/* calculating model 2 predictions using validation set created in part 1 */
proc score data=validation score=model_params out=predictions_valid_fold type=parms;
    var height weight cholesterol;
run;

/* calculating root mean squared residual */
data valid_fold_errors;
    set predictions_valid_fold;
    Residual = Systolic - MODEL1; /* Actual - Predicted */
    SquaredError = Residual**2;     /* Square of residual */
run;

proc means data=valid_fold_errors mean noprint;
    var SquaredError;
    output out=rmse_results mean=MeanSquaredError; /* calculating MSE */
run;

data rmse_final;
    set rmse_results;
    RMSE = sqrt(MeanSquaredError); /* Calculate RMSE */
run;

/* final output for rmse */
proc print data=rmse_final noobs;
    title "Root Mean Squared Error (RMSE)";
run;
```

SAS Output:



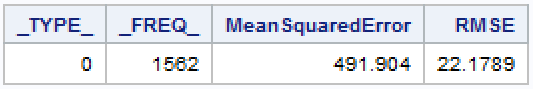
TYPE	_FREQ_	MeanSquaredError	RMSE
0	1562	491.974	22.1805

SAS Code (testing whole training set model on validation set):



```
/* training model from part 2 */
proc reg data=training outest=model_params noprint;
    model systolic = height weight cholesterol / vif;
    output out=reg_out_fold p=Predicted r=Residual; /* Predictions for train_fold */
run;

/* calculating model predictions using validation set created in part 1 */
proc score data=validation score=model_params out=predictions_valid_fold type=parms;
    var height weight cholesterol;
run;

/* calculating root mean squared residual */
data valid_fold_errors;
    set predictions_valid_fold;
    Residual = Systolic - MODEL1; /* Actual - Predicted */
    SquaredError = Residual**2;     /* Square of residual */
run;

proc means data=valid_fold_errors mean noprint;
    var SquaredError;
    output out=rmse_results mean=MeanSquaredError; /* calculating MSE */
run;

data rmse_final;
    set rmse_results;
    RMSE = sqrt(MeanSquaredError); /* Calculate RMSE */
run;

/* final output for rmse */
proc print data=rmse_final noobs;
    title "Root Mean Squared Error (RMSE)";
run;
```

SAS Output:



TYPE	_FREQ_	MeanSquaredError	RMSE
0	1562	491.904	22.1789

Conclusion:

- The accuracy of the model created using the entire training set is identical to the accuracy of model 2 created using 4/5 of the training set.
- While cross validation has not improved the predictive accuracy of a candidate model, it has managed to produce a model using a lower amount of training instances.