A Study on Physical Fitness

Sreejit Roy
Vaishnavi Jaiswal
Mousam Sinha

December 25, 2021

Contents

1 Introduction

The dataset contains various measures of heart and pulse rates taken on men in a physical fitness course. The following are defined:

$\boldsymbol{Y}$:= oxygen used to complete the given task.

X_1 := age of the participant.

X_2 := weight of the participant.

X_3 := time to run a given distance on a treadmill.

X_4 := measure of pulse-rate at rest.

X_5 := average pulse-rate during the run.

X_6 := maximum pulse-rate during the run.

For the purpose of this study, $\boldsymbol{Y}$ is taken as the response variable; the remaining as explanatory.

```
## Error in setwd("/Users/roysreejit/OneDrive/Academics/ISI/Sem 1/Regression Techniques/Project
cannot change working directory

## Warning in file(file, "rt"):  cannot open file 'fitness1.csv':  No such file or
directory

## Error in file(file, "rt"):  cannot open the connection

## Error in eval(expr, envir, enclos):  object 'fitdata' not found

## Error in colnames(fitdata) <- c("Y", "X1", "X2", "X3", "X4", "X5", "X6"):  object
'fitdata' not found

## Error in eval(expr, envir, enclos):  object 'fitdata' not found

## Error in attach(fitdata):  object 'fitdata' not found
```

2 Preliminary Analysis

For the given dataset, the correlation matrix is as follows:

```
## Error in is.data.frame(x):  object 'fitdata1' not found
```

(Table 2.1)

Remark:

1. There exists a high correlation between X_5 and X_6, viz., 0.93.

Consider the plots of all pairs of columns as follows:

```
## Error in pairs(fitdata1):  object 'fitdata1' not found
```

(Figure 2.1)

Remarks:

1. There exists a high positive relationship between X_5 and X_6
2. There is a linear relationship between Y and X_3 with a negative slope.

3 Model Fitting

Primarily, consider the simple linear regression model

$$\underset{\sim}{Y} = \mathbf{X}\underset{\sim}{\beta} + \underset{\sim}{\epsilon} \tag{1}$$

where $\mathbf{X}_{31\times 7} = (\underset{\sim}{1},\ X_1,\ X_2,\ ...,\ X_6)$, $\underset{\sim}{\beta} = (\beta_0,\ \beta_1,\ ...,\ \beta_6)'$, $\underset{\sim}{\epsilon} = (\epsilon_1,\ \epsilon_2,\ ...,\ \epsilon_{31})'$;

with the assumptions:

- $\underset{\sim}{\epsilon} \sim N(\underset{\sim}{0}, \sigma^2\mathbf{I_{31}})$
- $\mathbf{X}$ is non-stochastic and has full rank.

The following is a summary for the model considered.

```
fm <- lm(Y~ ., fitdata1)
```

```
## Error in is.data.frame(data):  object 'fitdata1' not found
```

```
summary(fm)
```

```
## Error in summary(fm):  object 'fm' not found
```

The following figure plots the response values (blue) and the predicted values (red). It serves as a visual indication of the goodness of fit of the model.

```
## Error in predict(fm):  object 'fm' not found
```

```
## Error in eval(i, data, env):  object 'fitdata1' not found
```

(Figure 3.1)

4 Verification of Model Assumptions

Along with model (1) (i.e., the full model), the following assumptions were made:

- $\underset{\sim}{\epsilon} \sim N(\underset{\sim}{0}, \sigma^2 \mathbf{I_{31}})$
- $\mathbf{X}$ is non-stochastic and has full rank.

In particular, the first assumption is that of homoscedasticity, normality, and absense of autocorrelation. The second assumption is about independence of explanatory variables and the non-stochastic nature of them.

In the following subsubsections we verify all the aforementioned assumptions except for that of non-stochastic nature. We just assume that the explanatory variables are indeed non-stochastic in nature.

4.1 Homoscedasticity

The assumption of homoscedasticity states that all the error terms have equal variance.

First, the residual plot is made for visually inspecting whether the assumption of homoscedasticity is violated or not:

```
## Error in check_model(model):  object 'fm' not found
```

(Figure 4.1)

Remark:

1. There is no evident pattern in the residual plot, implying that the assumption of homoscedasticity is preserved.

We now perform formal checks for violation of the assumption.

4.1.1 Breusch–Pagan Test

Breusch–Pagan test is performed for obtaining a formal (and more mathematical) conclusion about whether heteroscedasticity is present or not.

```
## Error in bptest(fm):  object 'fm' not found
```

Remark:

1. p-value of 0.8302 means that there is no heteroscedasticity in the data-set.

4.2 Normality

The assumption of normality states that all the errors are normally distributed.

The check the validy of the assumption, we first do the QQ-plot is as follows:

```
## Error in check_model(model):  object 'fm' not found
```

(Figure 4.2)

Remark:

1. Although most of the points are around the $y = x$ line, it is to be noted that there is an oscillating pattern centred around the line. Proper mathematical tests are performed for a more concrete conclusion.

4.2.1 Shapiro-Wilk Test

First, the Shapiro-Wilk test is performed for testing whether the assumption of normality is violated or not.

```
shapiro.test(residuals(fm))
```

```
## Error in residuals(fm):  object 'fm' not found
```

Remark:

1. p-value of 0.4603 indicates that the assumption of normality is not violated.

4.2.2 Kolmogorov-Smirnov Test

Kolmogorov-Smirnov test is also performed just to ensure whether the conclusion agrees with that obtained using Shapiro-Wilk test or not.

```
ols_test_normality(fm)[1]
```

```
## Error in ols_test_normality(fm):  object 'fm' not found
```

Remark:

1. p-value of 0.563 implies that the assumption of normality is valid.

4.3 Autocorrelation

Autocorrelation means that the errors are correlated with each other. It is assumed that autocorrelation is not present in the model. For the verification, first the ACF of the residuals are plotted:

```
## Error in residuals(fm):  object 'fm' not found
```

(Figure 4.3)

Remark:

1. It is easy to observe that there exists no pattern denoting the presence of autocorrelation. I.e., there is no autocorrelation.

Formal tests are performed to validate the claim.

4.3.1 Durbin-Watson Test

Durbin-Watson test is performed to check (mathematically) whether autocorrelation is present or not.

```
dwtest(fm)
```

```
## Error in dwtest(fm):  object 'fm' not found
```

Remark:

1. From the test, we can conclude that there is no autocorrelation in model. Thus, the errors are independently distributed.

4.3.2 Runs Test

Runs test is also performed as follows:

```
runs.test(residuals(fm), plot = TRUE)
```

```
## Error in residuals(fm):  object 'fm' not found
```

(Figure 4.4)

Remark:

1. The test concludes that the residuals do not follow any pattern. This, in turn, implies that the residuals are not auto-correlated.

4.4 Collinearity

A model is said to have multicollinearity when its regression matrix is not of full rank. In paricular, when some explanatory variables are linearly dependent, multicollinearity occurs. It is assumed that there is no multicollinearity in model (1). To verify the claim, the following are calculated.

4.4.1 VIF

First, the VIFs are checked for the model. As a rule of thumb, if VIF value for an explanatory variable is higher that 5, we suspect that multicollinearity is present.

```
vif(fm)
```

```
## Error in vif(fm):  object 'fm' not found
```

Remark:

1. We see that X_5 and X_6 have high VIF values. Also table 2.1 and figure 2.1 indicate presence of high positive correlation among these two variables.

4.4.2 Condition Number

The condition number is as follows:

```
kappa(fitdata1[, -c(1)])
```

```
## Error in kappa(fitdata1[, -c(1)]):  object 'fitdata1' not found
```

Remark:

1. On the basis of high condition number and VIF values for X_5 and X_6, we suspect that there is a presence of multicollinearity. Hence we perform remedies for multicollinearity in Section 6.

5 Detection of Influential Points

5.1 Cook's D Plot

We find points with high cook's distance.

```
## Error in check_model(model):  object 'fm' not found
```

(Figure 5.1)

Remark:

1. On the basis of above plot, we can see that observations 10, 15 and 20 are potential influential points.

5.2 DFFITS Plot

This plot will help us to detect influential data points.

```
## Error in check_model(model):  object 'fm' not found
```

(Figure 5.2)

Remark:

1. On the basis of DFFITS plot, we can see that observations 10, 15 and 20 are potential influential points.

5.3 DFBETAS Plot

This plot will help us to detect influential data points with respect to various regressor variables.

```
## Error in check_model(model):  object 'fm' not found
```

(Figure 5.3 and 5.4)

Remark:

1. On the basis of DFBETAS plot, we can see that observations 4, 10, 15, 17, and 20 are potential influential points.

5.4 Added Variable Plot

This plot gives the relationship between response variable and a regressor variable when other variables are held constant. Each added variable plot notes four different values, two of which are points with highest residuals, the others are those with highest partial leverage.

```
avPlots(fm)
```

```
## Error in avPlots(fm):  object 'fm' not found
```

(Figure 5.5)

Remark:

1. On the basis of above plots, we observe the 15^{th} and 17^{th} points are definitely outliers.

5.5 Studentised Residual Plot

We will try to find the outliers with the help of this plot. An observation is considered to be an outlier if the absolute value of studentised residual is greater than 2.

```
## Error in check_model(model):  object 'fm' not found
```

(Figure 5.6)

```
## Error in outlierTest(fm):  object 'fm' not found
```

Remark:

1. On the basis of the above plot, we can see that observations 15 and 17 are outliers.

5.6 Hat Values

We will calculate the hat diagonal values for all observations. Observations for which the hat diagonal is greater than $2p/n$ where p is the number of regression parameters and n is the number of observations.

```
## Error in hatvalues(fm):  object 'fm' not found

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):  plot.new has
not been called yet

## Error in hatvalues(fm):  object 'fm' not found

## Error in text(se2[idc2], hatvalues(fm)[idc2], labels = rownames(fitdata1)[idc2],
:  object 'idc2' not found
```

(Figure 5.7)

Remark:

1. On the basis of the above plot, we can see that observations 10 is a high leverage point.

5.7 Residual vs Leverage Plot

```
## Error in plot(fm, which = 5):  object 'fm' not found
```

(Figure 5.8)

Remarks:

1. On the basis of the above plot, note that 15^{th} is an outlier; 10^{th} point is high leverage.
2. We also note that 20^{th} is an influential point, mostly because it has moderately high residual as well as leverage.

Conclusion

- From all the above subsubsections, we note that the following observations are outliers: 15^{th}, 17^{th}, and 20^{th}. And the 10^{th} observation has high leverage.

Remedy

We remove 15^{th}, 17^{th}, and 20^{th} observations from the dataset and fit the model (1). We estimate the Y values for the 15^{th}, 17^{th}, and 20^{th} observations from this model. Then we use the obtained Y values as observed values in the dataset. Then we use the newly formed dataset for fitting model (1) again.

The following is the summary of the model (1) fitted with rectified Y vector.

```
## Error in eval(expr, envir, enclos):  object 'fitdata' not found
## Error in is.data.frame(data):  object 'fitdata2' not found
## Error in is.data.frame(data):  object 'fitdata' not found
## Error in summary(fm2):  object 'fm2' not found
```

The following tests are performed to check whether the assumptions are being satisfied or not.

```
bptest(fm2)
## Error in bptest(fm2):  object 'fm2' not found
shapiro.test(residuals(fm2))
## Error in residuals(fm2):  object 'fm2' not found
dwtest(fm2)
## Error in dwtest(fm2):  object 'fm2' not found
vif(fm2)
## Error in vif(fm2):  object 'fm2' not found
```

Remarks:

1. Heteroscedasticity, non-normality, and autocorrelation are not present.
2. Multicollinearity still exists, which is obvious as multicollinearity mainly depends on the regression matrix.

6 Remedies For Multicollinearity

6.1 Ridge Regression

Ridge regression is performed as a remedy for multicollinearity.

```
## Error in model.frame(formula = fitdata):  object 'fitdata' not found

## Error in object[[i]]:  object of type 'closure' is not subsettable

## Error in UseMethod("predict"):  no applicable method for 'predict' applied to an
object of class "function"

## Error in eval(i, data, env):  object 'fitdata1' not found
```

Remarks:

1. This model does not have the problem of multicollinearity. This directly follows from the theory of ridge regression.
2. In the following subsections, other methods of removing multicollinearity are explored in detail.

6.2 Removing Covariates

Another possible way of removing multicollinearity is that of selecting subsets of explanatory variables. It is already noted that the VIF values for X_5 and X_6 are high. It is also noted previously that these two are correlated. Therefore, if one of those two is removed, the problem of multicollinearity may be resolved. In the following two subsections, we remove each of those two (one at a time) and perform further analysis to obtain a good model.

6.2.1 Removing Runpulse

We remove Run pulse (X_5) from the list of explanatory variables. We then fit a linear model using the remaining five explanatory variables. Tests are performed to ensure that the model satisfies all the model assumptions.

```
## Error in is.data.frame(data):  object 'fitdata' not found

## Error in summary(m1):  object 'm1' not found

## Error in check_model(model):  object 'm1' not found

## Error in bptest(m1):  object 'm1' not found

## Error in check_model(model):  object 'm1' not found

## Error in residuals(m1):  object 'm1' not found

## Error in ols_test_normality(m1):  object 'm1' not found

## Error in residuals(m1):  object 'm1' not found

## Error in dwtest(m1):  object 'm1' not found

## Error in residuals(m1):  object 'm1' not found

## Error in vif(m1):  object 'm1' not found
```

```
## Error in kappa(fitdata[, -c(1, 6)]):  object 'fitdata' not found
```

Remarks:

1. Note that the problem of multicollinearity is resolved in this model.
2. Although it is to be noted that all explanatory variables are not significant.

6.2.2 Removing Maxpulse

We remove Max pulse (X_6) from the list of explanatory variables. We then fit a linear model using the remaining five explanatory variables. Tests are performed to ensure that the model satisfies all the model assumptions.

```
## Error in is.data.frame(data):  object 'fitdata' not found
## Error in summary(m2):  object 'm2' not found
## Error in check_model(model):  object 'm2' not found
## Error in bptest(m2):  object 'm2' not found
## Error in check_model(model):  object 'm2' not found
## Error in residuals(m2):  object 'm2' not found
## Error in ols_test_normality(m2):  object 'm2' not found
## Error in residuals(m2):  object 'm2' not found
## Error in dwtest(m2):  object 'm2' not found
```

```
## Error in residuals(m2):  object 'm2' not found
```

```
vif(m2)
## Error in vif(m2):  object 'm2' not found
kappa(fitdata[, -c(1, 7)])
## Error in kappa(fitdata[, -c(1, 7)]):  object 'fitdata' not found
```

Remarks:

1. Note that the problem of multicollinearity is resolved in this model.
2. Although it is to be noted that all explanatory variables are not significant.

7 Model Selection

7.1 Model 1

Here, consider the model obtained in section (6.2.1). We consider the added variable plot for the considered model.

```
## Error in avPlots(m1):  object 'm1' not found
```

Note that the slope in each plot reflects the partial regression coefficients from the actual regression model. Therefore, a slope closer to zero would mean that the partial regression coefficient is closer to zero corresponding to that explanatory variable. Since partial regression coefficients measure the expected change in Y for one unit change in the corresponding explanatory variable, provided all the other explanatory variables are held constant, a slope closer to zero signifies that the corresponding explanatory does not have much of an effect in the model.

Remark:

1. X_2 and X_4 are expected not to have highly significant effect on the response.

Subset Selection

The following code snippet chooses the best model for each possible number of explanatory variables.

```
regfit1 <- regsubsets(Y ~ X1 + X2 + X3 + X4 + X6, data = fitdata)

## Error in is.data.frame(data):  object 'fitdata' not found

regsumm1 <- summary(regfit1)

## Error in summary(regfit1):  object 'regfit1' not found

regsumm1

## Error in eval(expr, envir, enclos):  object 'regsumm1' not found
```

The following table table compares the obtained models (5 models) interms of AIC, BIC, Adjusted R-squared, and Mallow's Cp.

```
## Error in cbind(regsumm1$cp, regsumm1$adjr2, regsumm1$bic, ols_step_all_possible(m1)$aic[c(1,
:  object 'regsumm1' not found

## Error in rownames(mat1) <- c("X3", "X1, X3", "X1, X2, X3", "X1, X2, X3, X6", :  object
'mat1' not found

## Error in colnames(mat1) <- c("cp", "adjr2", "bic", "aic"):  object 'mat1' not found

## Error in kable(mat1):  object 'mat1' not found
```

Remarks:

1. In terms of Mallow's cp and AIC, the fourth model seems to be the best.
2. In terms of adjusted R-squared, model 5 is the better one.
3. The third model is best in terms of BIC.

We now use stepwise selection method to evaluate the best possible model as follows.

```
ols_step_both_aic(m1, details = TRUE)
```

```
## Error in ols_step_both_aic(m1, details = TRUE): object 'm1' not found
```

Remarks:

1. The stepwise selection method chooses the model Y on X1, X2, X3, and X6.
2. This model is preferable in terms of Mallow's cp and AIC also.
3. The adjusted R-square for this model is 0.807, which is close to the model preferable in terms of adjusted R-squared criterion.
4. Therefore, we select model 4 (i.e., Y on X1, X2, X3, and X6) among all possible subsets of model 1 (i.e., Y on X1, X2, X3, X4, and X6).

7.2 Model 2

Here, consider the model obtained in section (6.2.2). We consider the added variable plot for the considered model.

```
## Error in avPlots(m2):  object 'm2' not found
```

Remark:

1. X_2 and X_4 are expected not to have highly significant effect on the response.

Subset Selection

The following code snippet chooses the best model for each possible number of explanatory variables.

```
regfit2 <- regsubsets(Y ~ X1 + X2 + X3 + X4 + X5, data = fitdata)
```

```
## Error in is.data.frame(data):  object 'fitdata' not found
```

```
regsumm2 <- summary(regfit2)
```

```
## Error in summary(regfit2):  object 'regfit2' not found
```

```
regsumm2
```

```
## Error in eval(expr, envir, enclos):  object 'regsumm2' not found
```

The following table table compares the obtained models (5 models) interms of AIC, BIC, Adjusted R-squared, and Mallow's Cp.

```
## Error in cbind(regsumm2$cp, regsumm2$adjr2, regsumm2$bic, ols_step_all_possible(m2)$aic[c(1,
:  object 'regsumm2' not found
```

```
## Error in rownames(mat2) <- c("X3", "X1, X3", "X1, X3, X5", "X1, X2, X3, X5", :  object
'mat2' not found
```

```
## Error in colnames(mat2) <- c("cp", "adjr2", "bic", "aic"):  object 'mat2' not found
```

```
## Error in kable(mat2):  object 'mat2' not found
```

Remarks:

1. In terms of Mallow's cp, BIC, and AIC, the fourth model seems to be the best.
2. In terms of adjusted R-squared, model 5 is the better one, although the adjusted R-squared is very close for model 4 and 5.

We now use stepwise selection method to evaluate the best possible model as follows.

```
ols_step_both_aic(m2, details = TRUE)
```

```
## Error in ols_step_both_aic(m2, details = TRUE): object 'm2' not found
```

Remark:

1. The stepwise selection method chooses the model Y on X1, X2, X3, and X5, which is also preferable in terms of the other criteria. Therefore, we select model 4 (i.e., Y on X1, X2, X3, and X5) among all possible subsets of model 1 (i.e., Y on X1, X2, X3, X4, and X5).

7.3 Selecting the Final Model

We compare the models selected from sections 7.1 and 7.2. I.e., we compare the following models:

Model A: $E(Y) = \gamma_0 + \gamma_1 X_1 + \gamma_2 X_2 + \gamma_3 X_3 + \gamma_6 X_6$

Model B: $E(Y) = \lambda_0 + \lambda_1 X_1 + \lambda_2 X_2 + \lambda_3 X_3 + \lambda_5 X_5$

We compare these in terms of adjusted R-squared, AIC, BIC, and Mallow's Cp to choose the better one among these.

```
## Error in is.data.frame(data):  object 'fitdata' not found
```

```
## Error in is.data.frame(data):  object 'fitdata' not found
```

```
## Error in matrix(c(regsumm1$adjr2[4], AIC(ma), BIC(ma), regsumm1$cp[4], :  object
'regsumm1' not found
```

```
## Error in rownames(matf) <- c("Model A", "Model B"):  object 'matf' not found
```

```
## Error in colnames(matf) <- c("adjr2", "aic", "bic", "cp"):  object 'matf' not found
```

```
## Error in kable(matf):  object 'matf' not found
```

Conclusion

- From the above table, it is easy to observe that model B is better than model A. Therefore, we choose model B (i.e., Y on X1, X2, X3, and X5) as our final model.

8 Final Verification

As our final model, we take

$$E(Y) = \lambda_0 + \lambda_1 X_1 + \lambda_2 X_2 + \lambda_3 X_3 + \lambda_5 X_5 \tag{2}$$

The following is the summary of the model.

```
## Error in summary(mb):  object 'mb' not found
```

We now check for the influential points.

```
## Error in check_model(model):  object 'mb' not found
```

```
## Error in check_model(model):  object 'mb' not found
```

```
## Error in plot(mb, which = 5):  object 'mb' not found
```

Remark:

1. We have 2^{nd}, 4^{th}, and 10^{th} as potential influential points. Among these, we only remove the 10^{th} point, as this differs from the most.
2. The other two points can be removed, but are not, as this will lead to huge data loss for a small dataset like this one.

We thus remove the 10^{th} point from the dataset altogether and fitted the model (2) again.

```
## Error in eval(expr, envir, enclos):  object 'fitdata' not found
```

```
## Error in is.data.frame(data):  object 'fitdata3' not found
```

```
## Error in summary(m):  object 'm' not found
```

We perform all the tests for checking whether the assumptions are still holding for this model.

```
## Error in bptest(m):  object 'm' not found
```

```
## Error in residuals(m):  object 'm' not found
```

```
## Error in ols_test_normality(m):  object 'm' not found
```

```
## Error in dwtest(m):  object 'm' not found
```

```
vif(m)
```

```
## Error in vif(m):  object 'm' not found
```

Remark:

1. For model (2), none of the assumptions are violated. Therefore, we can indeed consider this as our final model.

The following figure plots the response values (blue) and the predicted values (red). It serves as a visual indication of the goodness of fit for the final model.

```
## Error in predict(m):  object 'm' not found
```

```
## Error in eval(expr, envir, enclos):  object 'fitdata1' not found
```

```
## Error in eval(i, data, env):  object 'fitdata4' not found
```

9 References and Bibliography

1. Seber, G.A.F., Lee, A.J., *Linear Regression Analysis, Second Edition*, John Wiley and Sons.
2. Hastie, T., Tibshirani, R., Friedman, J., *The Elements of Statistical Learning, Second Edition*, Springer.
3. James, G., Witten, D., Hastie, T., Tibshirani, R., *An Introduction to Statistical Learning with Applications in R*, Springer.
4. https://www.rdocumentation.org
5. https://www.isid.ac.in/~deepayan/R-tutorials/index.html

10 Acknowledgements

We would like to express our deeply felt gratitude and regards towards **Prof. Dr. Swagata Nandi ma'am**. Her continuous support, suggestions, and guidance helped us in solving our doubts and figuring out methodologies and procedures. Her help and vision made it possible for us to finish the project.