Prediction of Residual Resistance from Hull Geometry Coefficients and Froude Number

R Tumber

19/10/2020

Introduction

This investigation was performed on the dataset produced from a series of experiments that took place at the Delft Ship Hydromechanics Laboratory. The goal was to predict the value of residual resistance per unit weight of displacement from a variety of hull geometry coefficients and Froude number as accurately as possible. The dataset was downloaded from the UCI Machine Learning Laboratory and then imported, the information for the column names came from scraping the html on the webpage and extracting the relevant information. Since the column names were longer than ideal they were shortened, with the definitions stored separetley for reference.

att_inf	short_names
Longitudinal position of the center of buoyancy, adimensional.	lcb
Prismatic coefficient, adimensional.	cp
Length-displacement ratio, adimensional.	dlr
Beam-draught ratio, adimensional.	bt
Length-beam ratio, adimensional.	lb
Froude number, adimensional.	fn
Residuary resistance per unit weight of displacement, adimensional.	resid_reswei

The dataset consisted of 308 instances of 7 attributes:

```
## Classes 'data.table' and 'data.frame':   5 obs. of  7 variables:
##  $ lcb         : num  -2.3 -2.3 -2.3 -2.3 -2.3
##  $ cp          : num  0.568 0.568 0.568 0.568 0.568
##  $ dlr         : num  4.78 4.78 4.78 4.78 4.78
##  $ bt          : num  3.99 3.99 3.99 3.99 3.99
##  $ lb          : num  3.17 3.17 3.17 3.17 3.17
##  $ fn          : num  0.125 0.15 0.175 0.2 0.225
##  $ resid_reswei: num  0.11 0.27 0.47 0.78 1.18
##  - attr(*, ".internal.selfref")=<externalptr>
```

All attributes were numerical and there were no missing entries in the dataset. The variable to predict was the Residuary resistance per unit weight of displacement.

In examining the data, each variable was plotted against the residual resistance to determine if there were any obvious correlations and the correlation coefficients calculated. Following this, a correlation and covariance matrix was produced, looking for relationships not seen in the visualisations. The only variable showing a clear relationship was the Froude number. Where no clear relationship was observed, further visualisations

were made using the average residual resistance at different values of variables in an attempt to identify a weak relationship. Here, the Prismatic Coefficient showed a possible relationship. The Froude number was then examined more closely, the observed exponential style curve was straightened to allow the option of modelling using a linear regression approach.

It was decided that all variables should, in theory, have an effect on the residual resistance.

Before modelling, the data were split into a training and test set. Initially, two models were built using Caret to predict residual resistance from Froude Number only and then with the Prismatic Coefficient, using two separate ensembles, one using linear regression methods and the log of the Froude Number and the other using non-linear regression methods. Model accuracy was assessed using RMSE as this could be applied to both model types.

The more sucessful, non-linear regression model was kept and within this model the Random Forest type method was most accurate. As a result, a new ensemble was put together with a selection of Random Forest Type methods, and the generated models assessed again, with the most accurate two being carried forward. At this stage each of the remaining variables was added one at a time and only where RMSE was improved was the variable set aside to add at the end. Only one other variable had a beneficial effect on model accuracy, Length-Displacement Ratio. This was added to the model before tuning. The tuned values were used to produce a model from which predicted residual resistance values were generated from the test set and a final RMSE calculated.

Method and Analysis

Downloading and wrangling the dataset

The Yacht Hydrodynamics dataset URL was downloaded from the UCI Machine Learning Laboratory and imported as a data table with no column headers. The column headers were scraped from the UCI page for that dataset (https://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics) the html manually inspected, split with '>' for easier viewing and the table containing the required data isolated as a list. This list was then then split and saved as a data table before a series of edits to this data table left it in a position in which to extract the column names. This data was tidied further with one more split, saved as a new object and a regex used to remove superfluous information. Pertinent information concerning the nature of the dataset was contained in this object so it was extracted and saved for use elsewhere. The remaining data were set as the column names to the imported dataset.

Investigating the dataset structure

According to data scraped from the UCI Machine Learening Repository, variables 1-6 are described by the information previously retained from the attribute names object

```
## [1] "Variations concern hull geometry coefficients and the Froude number:"
```

The seventh variable is that we looked to predict, described by the second retained data object

```
## [1] "The measured variable is the residuary resistance per unit weight of displacement:"
```

Variables 1-6 would now be explored in relation to variable 7 however, before proceeding the column names were shortened to aid clarity and the attribute information object used for the column names was repurposed as a key.

att_inf	short_names
Longitudinal position of the center of buoyancy, adimensional.	lcb
Prismatic coefficient, adimensional.	cp
Length-displacement ratio, adimensional.	dlr
Beam-draught ratio, adimensional.	bt
Length-beam ratio, adimensional.	lb
Froude number, adimensional.	fn
Residuary resistance per unit weight of displacement, adimensional.	resid_reswei

Exploring the data

```
## Classes 'data.table' and 'data.frame':   5 obs. of  7 variables:
##  $ lcb         : num  -2.3 -2.3 -2.3 -2.3 -2.3
##  $ cp          : num  0.568 0.568 0.568 0.568 0.568
##  $ dlr         : num  4.78 4.78 4.78 4.78 4.78
##  $ bt          : num  3.99 3.99 3.99 3.99 3.99
##  $ lb          : num  3.17 3.17 3.17 3.17 3.17
##  $ fn          : num  0.125 0.15 0.175 0.2 0.225
##  $ resid_reswei: num  0.11 0.27 0.47 0.78 1.18
##  - attr(*, ".internal.selfref")=<externalptr>
```

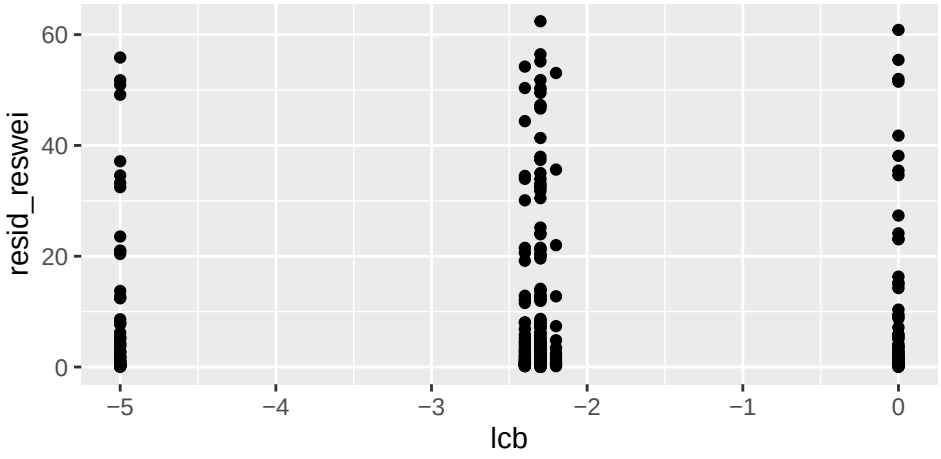
The dataset structure showed it contained 308 observations of seven variables, all numeric. The variable to predict was the Residual Resistance per Unit Weight of Displacement and there were no missing values.

To begin the investigation the mean and standard deviation of the target variable were ascertained.

mean	std_dev
10.49536	15.16049

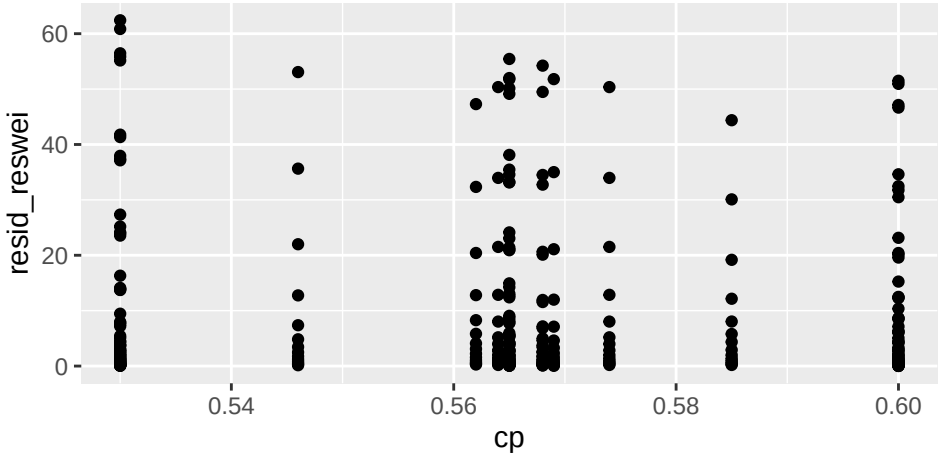
The variation in the values indicates either there are one or more factors in the dataset that influence the residual resistance by weight, the factors that influence the residual resistance are not in the dataset or that the values are random. By looking at the factors in the dataset we will attempt to determine which, if any, of them influence the result. Starting from the first and working through we look at how the residual resistance changes with each factor value.

1. Longitudinal position of the center of buoyancy, lcb.

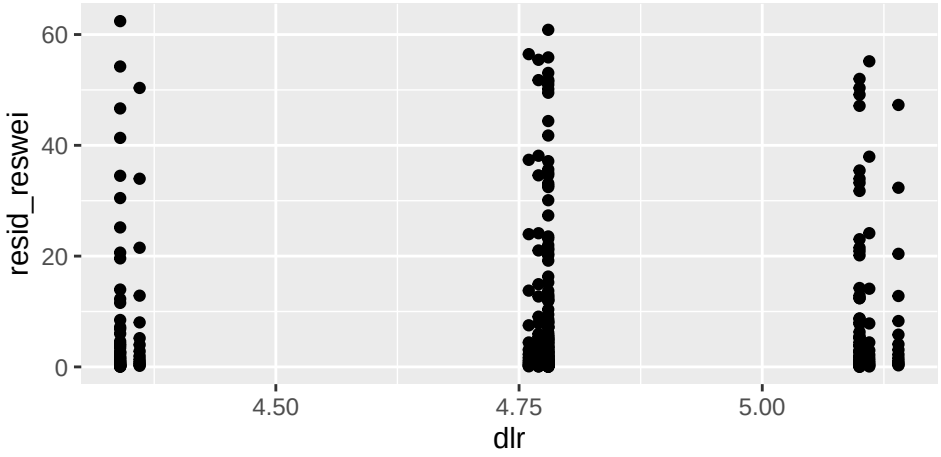


The plot shows while there are few different values for the Longitudinal position of the center of buoyancy, for each one there are many different residual resistance values. Additionally there appear to be some clusters of values, a possible indication of a relationship with another factor or factors. One final note on this chart, it looks like no matter the value of lcb, the residual resistance values tend to be closer to zero and not down to random variation.

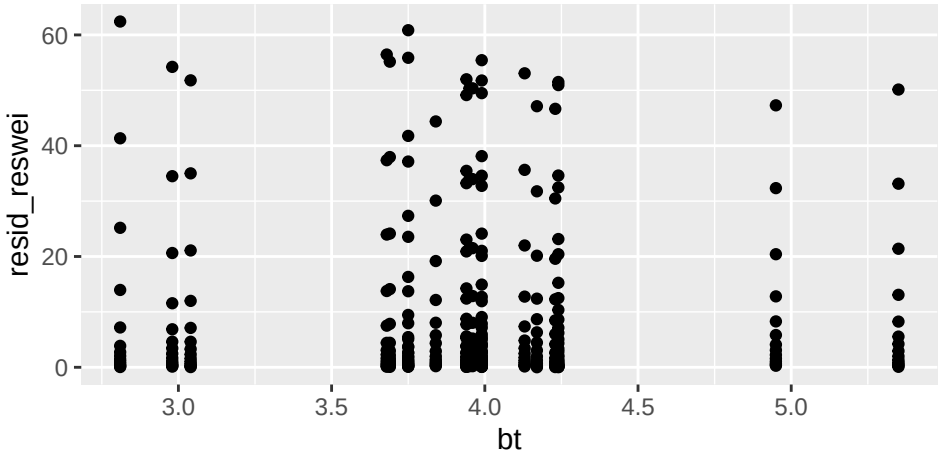
2. Prismatic coefficient, cp



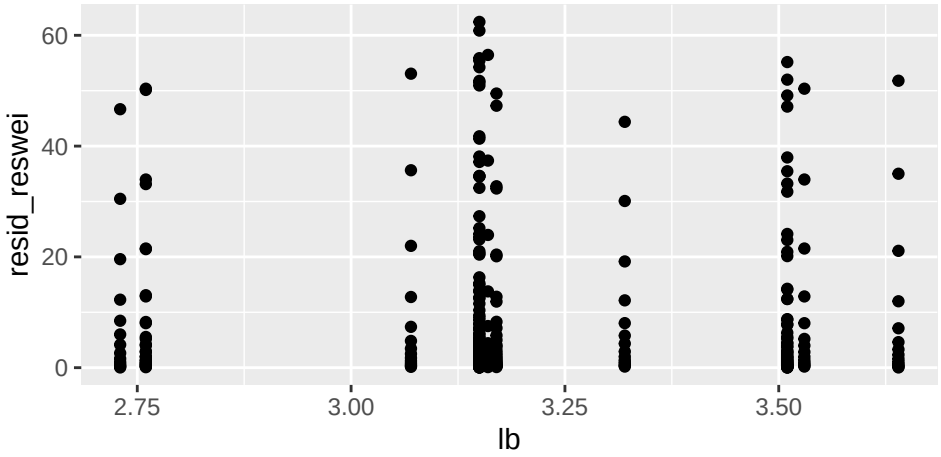
3. Length-displacement ratio, dlr



4. Beam-draught ratio, bt

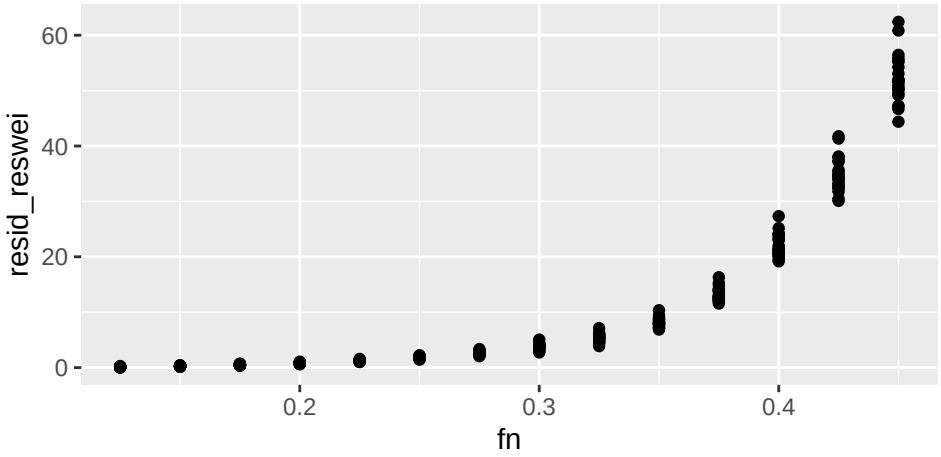


5. Length-beam ratio, lb



The four scatterplots above for the second to fifth variables follow a similar pattern as that for the first, a range of residual resistances for individual variable values with no obvious correlation observed and residual resistances being weighted towards zero. The clustering observed in the first plot persists and could be argued to be down to the controlled variation in experimental values used in the course of data gathering.

6. Froude number, fn



This plot appears to show the Froude number as having a bearing on the residual resistance. The shape could be related to the equation by which the Froude number is calculated.

$$Fr_l = u/\sqrt{gL_{wl}}$$

It could be the square root element that is influencing the shape of the curve when the equation is rearranged away from Froude number as the subject. It should be noted, there is still variation in residual resistance for identical Froude numbers, indicating this is likely not the only factor to influence the residual resistance. So, while the plots for the previous factors did not give any strong indications of a relationship with the residual resistance, they should not be discounted entirely.

Correlation values calculated for the above variables confirm as is observed in the scatterplots, there is no direct relationship between the variables on their own and the residual resistance with the exception of the final variable, the Froude Number.

Correlation and covariance matrices were constructed to assess relationships between factors and residual resistance to look for any relationships not seen by comparing individual factors to the residual resistance. For reference, the correlation values for the individual factors against residual resistance are also seen here.

Correlation

	lcb	cp	dlr	bt	lb	fn	resid_reswei
lcb	1.00	-0.01	0.00	0.00	0.00	0.00	0.02
cp	-0.01	1.00	-0.05	0.34	-0.09	0.00	-0.03
dlr	0.00	-0.05	1.00	0.38	0.68	0.00	0.00
bt	0.00	0.34	0.38	1.00	-0.38	0.00	-0.01
lb	0.00	-0.09	0.68	-0.38	1.00	0.00	0.00
fn	0.00	0.00	0.00	0.00	0.00	1.00	0.81
resid_reswei	0.02	-0.03	0.00	-0.01	0.00	0.81	1.00

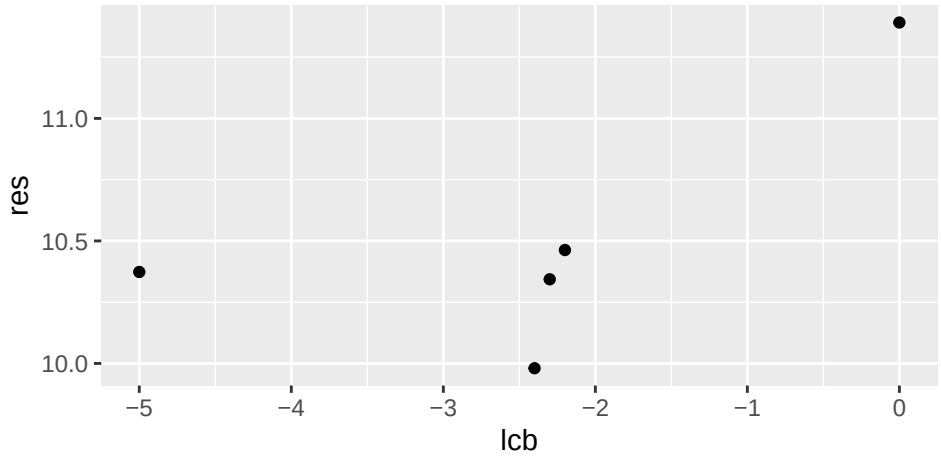
Covariance

	lcb	cp	dlr	bt	lb	fn	resid_reswei
lcb	2.29	0.00	0.00	0.00	0.00	0.00	0.44
cp	0.00	0.00	0.00	0.00	0.00	0.00	-0.01
dlr	0.00	0.00	0.06	0.05	0.04	0.00	-0.01
bt	0.00	0.00	0.05	0.30	-0.05	0.00	-0.10
lb	0.00	0.00	0.04	-0.05	0.06	0.00	0.00
fn	0.00	0.00	0.00	0.00	0.00	0.01	1.24
resid_reswei	0.44	-0.01	-0.01	-0.10	0.00	1.24	229.84

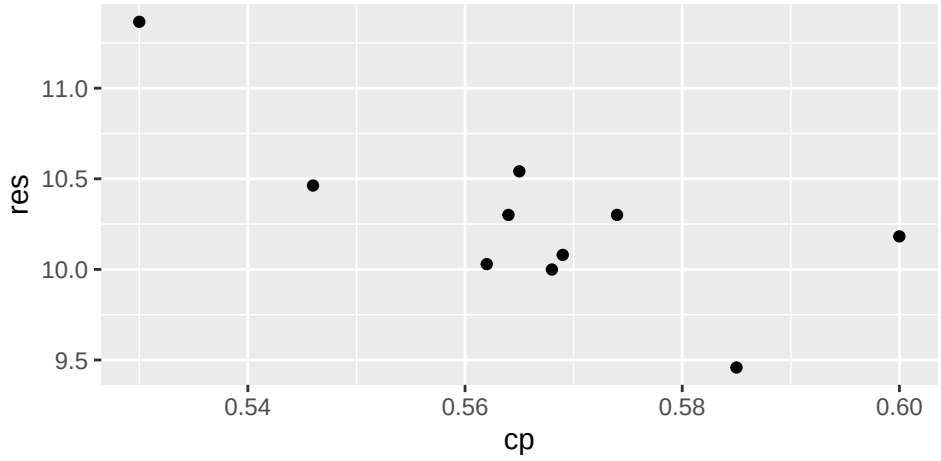
For the most part there was little observed in the way of correlation and covariance between the factors and the residual resistance, with the exception of the Froude number, which we already identified as potentially being significant. This result does not assist in explaining the variation in residual resistance for a constant Froude Number. However, in order for the experimental data to be at all useful we must assume that all parameters, with the exception of those in the dataset, have remained the same so the variation observed must either be down to random variation or down to one, some or all of the remaining factors.

With the above in mind the average residual resistance for each factor value was taken and the results plotted to determine if a weak relationship could be observed.

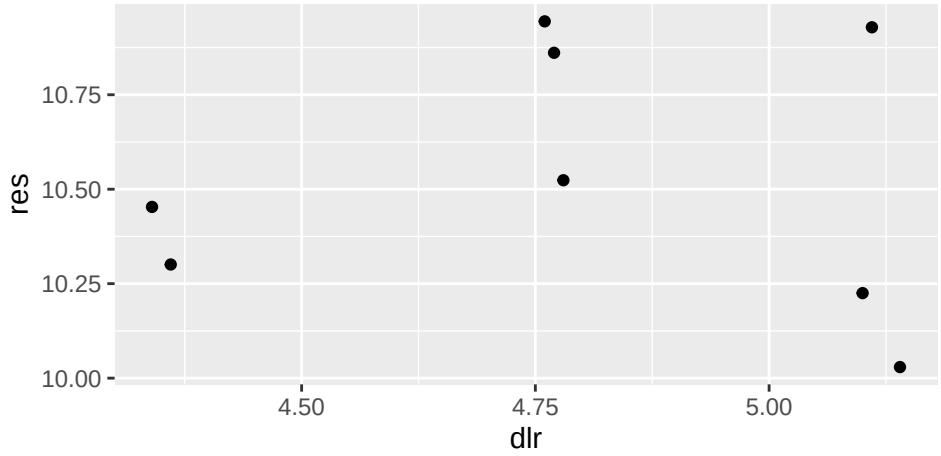
1. Longitudinal position of the center of buoyancy



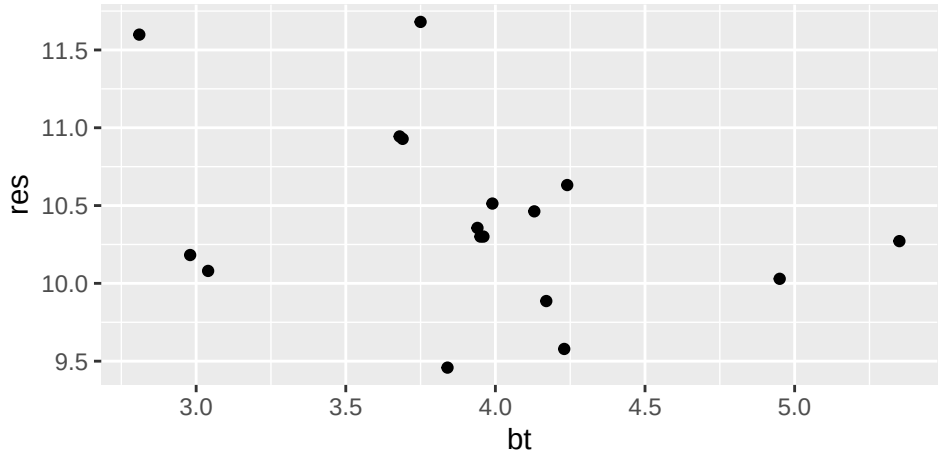
2. Prismatic coefficient



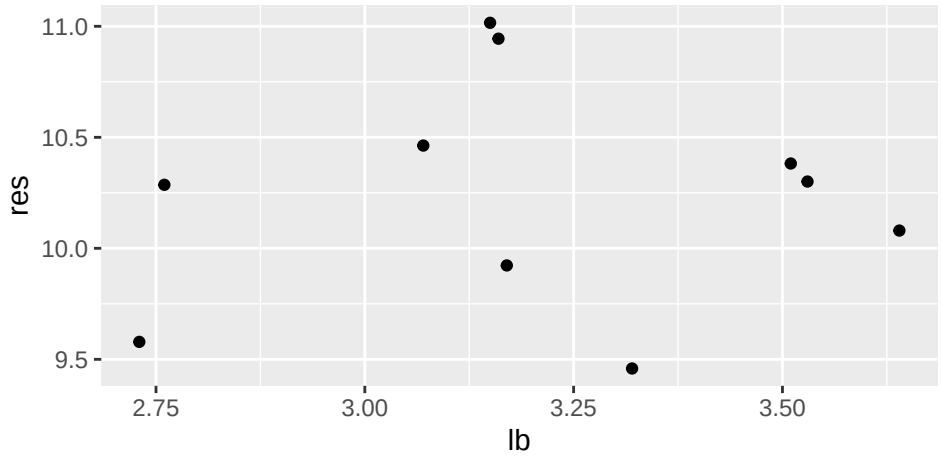
3. Length-displacement ratio



4. Beam-draught ratio



5. Length-beam ratio



An examination of the graphs only produced an indication of a possible relationship between the Prismatic coefficient and the residual resistance. While not entirely clear, further support for this relationship can be found by looking at the equations for both the Prismatic Coefficient, which may show a relationship, and the Froude number, which displays a strong relationship. The equation for calculating the Prismatic Coefficient is

$$C_p = V/L_{wl}A_m$$

where C_p is the Prismatic Coefficient, V is Volume, L_{wl} is length at the waterline and A_m is Cross sectional area.

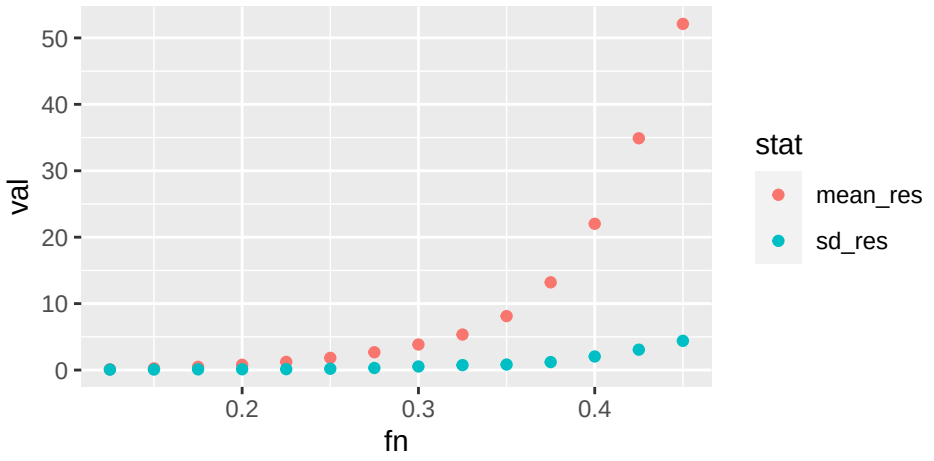
The equation for calculating the Froude number is

$$Fr_l = u/\sqrt{gL_{wl}}$$

where Fn_l is the Froude Number, u is relative flow velocity, g is the acceleration due to gravity and L_{wl} is length at the waterline. Since u is not defined we must consider it constant as we do with g. This means the only remaining part of the equation for calculating the Froude Number is L_{wl} and must have a bearing on residual resistance.

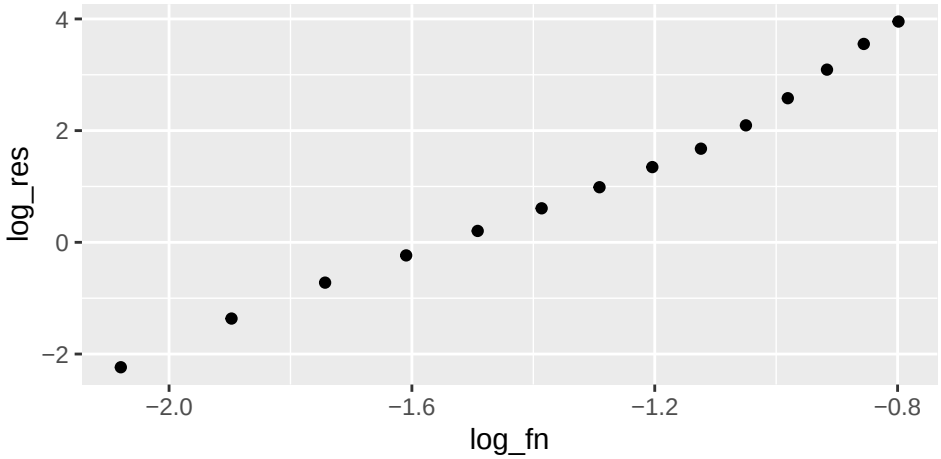
Relating this to Prismatic coefficient we see L_{wl} is a common factor in the above equations, so where there is a relationship with residual resistance for one factor, it is likely to continue for another. For this reason this factor will be included in creating a prediction model.

Returning to the Froude Number/residual resistance plot data, the mean and standard deviation of the residual resistance for each Froude number was calculated and plotted. The idea being to simplify the plot and illustrate any irregularities in the data points. The below plot demonstrates no such iregularities were found, the resistance values remained exponential and the standard deviation appeared almost linear, in line with expectations.



The residual resistance plot is non linear so it presents us two options when it comes to modelling. Either use a non-linear regression method or attempt to straighten the curve and factor this into a linear regression model.

The curve is partially straightened, producing a linear relationship, by using log values of Froude number to predict log values of Residual Resistance. While the result was not completely straight it was decided this could prove a useful approximation given it would be unlikely for this process to produce a straight line where there are other influences on the residual resistance.



Before proceeding to modelling it is important to return to the four factors that did not appear to have any relationship with the residual resistance. It should first be noted for all factors the number of data points was low, making it challenging to observe any relationships. While it was not possible to determine a definite relationship with the remaining factors, we should not discount them from further analysis, the reasons for which are outlined below.

Factor 1. Longitudinal position of the center of buoyancy, lcb.

The longitudinal centre of buoyancy is at the centre of the underwater volume, that is, the volume of the hull that lies beneath the waterline. Since this relates to volume below the waterline it could be argued that the length of the hull at the waterline is important which may indicate a relationship with resistance as it has previously.

Factor 3. Length-displacement ratio, dlr.

The length-displacement ratio describes how heavy a craft is in relation to its length at the waterline. Defined

as

$$dlr = (displacement(lb)/2240)/((0.01 * L_{wl})^3)$$

L_{wl}, length at the waterline once again has a bearing on the calculated value, in the same way it does for Prismatic Coefficient and Froude Number.

Factor 4. Beam-draught ratio, bt.

It would be unwise to pivot all arguements solely on the length of a hull at the waterline. The beam-draught ratio, the ratio of the beam, the width of the ship at the widest point at the waterline, and the draught, the distance between the waterline and deepest part of the hull, provides us a rough idea of the cross sectional area of submerged hull. If we look back to the equation for calculating the Prismatic Coefficient, $C_p = V/L_{wl}A_m$, A_m represented the cross sectional area. While this by no means proves a link to the residual resistance, it could be argued that there are reasonable grounds to consider the Beam-draught ratio as having some influence.

Factor 5. Length-beam ratio, lb.

The Length-beam ratio describes the rough shape of the hull at the waterline. If we consider the length at the waterline and the beam measurement in the context of the other dataset factors, and we are happy that they, in part or as a whole contribute to the residual resistance, it must follow that this ratio also has an effect.

With the above in mind we were left with a choice, produce a model that discounts these factors, based on a lack of evidence, a model that includes all these factors or produce a model that adds them stepwise thus retaining more control over the training in light of the small sample size.

It was decided the structured approach would be to first model the factors Froude Number followed by Prismatic Coefficient using three different linear regression models, followed by three models that do not use linear regression. At each stage the performance of the models would be assessed and after modelling Prismatic coefficient, the best performing retained. Model types similar to the most sucessful would then be tested to attempt to improve accuracy further.
Following this, the remaining factors, lcb, dlr, bt and lb, would be added individually one at a time, model performance assessed and the factors removed again as appropriate. Once all factors that improve model accuracy are isolated, the model hyperparameters would then be tuned.

The assessment of the models would be based on the RMSE, on the bases that it can be applied to both linear and non-linear regression models allowing for direct comparison of model performance, and that it can be applied to continuous data.

Model Creation and Testing

Before model building could begin, a training and test set was created. Since the dataset was small a 80/20 train/test split was used along with k-fold cross validation on the training set to improve accuracy before assessing the model on the test set. 10 fold cross validation with 20% sections of the training data was applied to strike a balance between having a sample size that is not too closely related to the full training data set, overtraining and processing time.

The following model types were used, grouped as linear and non-linear and saved as character objects, to first get an idea of accuracy as these represent a reasonable overview of the different models available to us, before moving forward with the more sucessful of these models.

- Linear regression models
  - lm
  - glm
  - svmLinear

- Non-linear regression models
  - svmPoly
  - knn
  - RRF

```
lr_model_ens <- c("lm", "glm", "svmLinear")
nlr_model_ens <- c("svmPoly", "knn", "RRF")
```

10-fold cross validation control to be used in training was defined

```
control <- trainControl(method = "cv", number = 10, p = 0.8)
```

The model name was added to the training function to gauge progress and identify any potential bottlenecks and the defined cross validation control was added.

Linear Regression For the linear regression models, log values for fn and resid_reswei are used. Before training, the seed was set to ensure models use the same cross validation samples. The training function was first applied to the Froude Number.

```
set.seed(1, sample.kind="Rounding")
lr_fits <- lapply(lr_model_ens, function(model){
  print(model)
  train(log(resid_reswei) ~ log(fn), method = model, data = res_train,
                                                        trControl = control)
})
```

The RMSE for each method was then calculated. It should be noted the RMSE generated by the fit was that for log residual resistance and needed converting back to resistance.

```
n_models <- seq(1:3)
lr_rmses <- sapply(n_models, function(m_number){
  lr_fits[[m_number]][["results"]][["RMSE"]]
})

lr_rmses <- as.data.frame(lr_rmses) %>%
  mutate(model_name = lr_model_ens)
colnames(lr_rmses) <- c("RMSE", "model_name")
lr_rmses <- lr_rmses %>%
  mutate(adjusted_RMSE = exp(RMSE))
```

This was repeated but with the Prismatic Coefficient added.

```
set.seed(1, sample.kind="Rounding")
lr_fits_cp <- lapply(lr_model_ens, function(model){
  print(model)
  train(log(resid_reswei) ~ log(fn) + cp, method = model, data = res_train,
                                                        trControl = control)
})

lr_rmses_cp <- sapply(n_models, function(m_number){
  lr_fits_cp[[m_number]][["results"]][["RMSE"]]
```

```
})

lr_rmses_cp <- as.data.frame(lr_rmses_cp) %>%
  mutate(model_name = lr_model_ens)
colnames(lr_rmses_cp) <- c("RMSE", "model_name")
lr_rmses_cp <- lr_rmses_cp %>%
  mutate(adjusted_RMSE = exp(RMSE))
```

When both sets of RMSEs were examined there was no great difference made by adding the prismatic coefficient to the models, with one of the three models tested producing a worse RMSE. This may reflect the observations made in exploring the data, where the relationship between Prismatic Coefficient and Residual Resistance was not necessarily clear.

Froude Number Only

RMSE	Model Name	Adjusted RMSE
0.3131871	lm	1.367778
0.3168552	glm	1.372804
0.3132965	svmLinear	1.367927

Froude Number with Prismatic Coefficient

RMSE	Model Name	Adjusted RMSE
0.3146906	lm	1.369835
0.3154315	glm	1.370851
0.3125321	svmLinear	1.366882

Non-Linear Regression A similar process was undertaken for the non-linear regression models, however for these models the normal values of Froude Number and Residual resistance were used and no adjusted RMSE was required. The models were trained using caret and this automatically chooses the most favourable tuning parameters, however in the process it produces a number of different evaluation metrics. As a result of this, the minimum value for RMSE was selected on the basis that this result is achievable by tuning the model.

```
set.seed(1, sample.kind="Rounding")
nlr_fits <- lapply(nlr_model_ens, function(model){
  print(model)
  train(resid_reswei ~ fn, method = model, data = res_train, trControl = control)
})

n_models <- seq(1:3)
nlr_rmses <- sapply(n_models, function(m_number){
  min(nlr_fits[[m_number]][["results"]][["RMSE"]])

nlr_rmses <- as.data.frame(nlr_rmses) %>%
  mutate(model_name = nlr_model_ens)
colnames(nlr_rmses) <- c("RMSE", "model_name")
```

As was the case for the Linear Regression models, the above was repeated but with the addition of the Prismatic Coefficient. This gave the below RMSEs.

Froude Number Only

RMSE	Model Name
3.635869	svmPoly
1.620178	knn
1.639099	RRF

Froude Number with Prismatic Coefficient

RMSE	Model Name
3.385447	svmPoly
1.976563	knn
1.167411	RRF

In both the linear and non-linear regression models, addition of the prismatic coefficient produced either a small rise or a small reduction in the RMSE for the residual resistance, however it is clear a Random Forest approach gives better results.

Model Name	RMSE
lm	1.369835
glm	1.370851
svmLinear	1.366882
svmPoly	3.385447
knn	1.976563
RRF	1.167411

With this in mind, other Random Forest models were investigated to see if further improvements could be made before attempting to add any of the remaining factors.

Random Forest The Random Forest models used were Ranger, Rborist, Random Forest (rf) and Regularized Random Forest (RRF).

A similar approach to the above in terms of testing and evaluating multiple model types at once was taken.

```
rf_model_ens <- c("ranger", "Rborist", "rf", "RRF")

set.seed(1, sample.kind="Rounding")
rf_fits_cp <- lapply(rf_model_ens, function(model){
  print(model)
  train(resid_reswei ~ fn + cp, method = model, data = res_train, trControl = control)
})

n_rf_models <- seq(1:4)
rf_rmses_cp <- sapply(n_rf_models, function(m_number){
  min(rf_fits_cp[[m_number]][["results"]][["RMSE"]])
})

rf_rmses_cp <- as.data.frame(rf_rmses_cp) %>%
  mutate(model_name = rf_model_ens)
colnames(rf_rmses_cp) <- c("RMSE", "model_name")
```

RMSE	Model Name
1.073457	ranger
1.211106	Rborist
1.222072	rf
1.141862	RRF

There were two models performing noticably better than the others, ranger and RRF. The fitted models were examined and based on the information on current tuning at RMSE minimum, the hyperparameters were tuned where necessary to provide a solid baseline RMSE before the investigation of the effect of addition of further factors.

Tuning

ranger

mtry	min.node.size	splitrule	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
2	5	variance	1.150540	0.9945605	0.6264000	0.1921253	0.0030948	0.0961569
2	5	extratrees	1.073457	0.9951323	0.5891589	0.2335123	0.0023280	0.1092169

	mtry	splitrule	min.node.size
2	2	extratrees	5

The above describes the combination of parameters used in modelling along with those which were most sucessful.

For *mtry* the default is the *number of predictor variables/3*. In the dataset this number would not change, even if all the variables were added to the model, but to be thorough the performance of mtry values 1-2 was checked. *split rule* set as *extratrees* appeared more successful than *variance* so was kept as is. Since *min.node.size* has a default value of 5 for regression, and a regression was performed, this parameter was left as 5. Putting this all together and building the model again with these adjusted parameters gave the below

```
ranger_grid <- expand.grid(mtry = seq(1:2), splitrule = "extratrees", min.node.size = 5)
set.seed(1, sample.kind="Rounding")
ranger_fn_cp_tune <- train(resid_reswei ~ fn + cp, method ="ranger", data = res_train,
                                          trControl = control, tuneGrid = ranger_grid)
```

mtry	splitrule	min.node.size	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	extratrees	5	3.840226	0.9734552	2.5296861	1.2708271	0.0070935	0.6672442
2	extratrees	5	1.073457	0.9951323	0.5891589	0.2335123	0.0023280	0.1092169

The model generated using the tuning parameters showed the original tuning produced the lowest RMSE.

RRF

mtry	coefReg	coefImp	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
2	0.010	0.0	1.142863	0.9948754	0.6405615	0.2845344	0.0022987	0.1607977
2	0.010	0.5	1.146034	0.9948691	0.6409753	0.2846782	0.0022848	0.1600841
2	0.010	1.0	1.145452	0.9948743	0.6406824	0.2841741	0.0023271	0.1620032
2	0.505	0.0	1.145433	0.9948647	0.6415541	0.2838069	0.0022629	0.1584682
2	0.505	0.5	1.150009	0.9948643	0.6409012	0.2806234	0.0022783	0.1587817
2	0.505	1.0	1.145654	0.9948822	0.6399472	0.2827085	0.0022692	0.1588434
2	1.000	0.0	1.148958	0.9948450	0.6422624	0.2800365	0.0022198	0.1576600
2	1.000	0.5	1.141862	0.9948915	0.6392063	0.2849963	0.0022655	0.1615622
2	1.000	1.0	1.147231	0.9948426	0.6422006	0.2876507	0.0023107	0.1614819

	mtry	coefReg	coefImp
8	2	1	0.5

Since the *mtry* value had already been tested on the ranger model and the results for a value of 1 were significantly worse than that of 2, this parameter was left as it is. The *coefReg* values and *coefImp* values in the best tune of 1 and 0.5 respectively, can only take values from zero to one so figures around the best tune figures were examined.

```
rrf_grid <- expand.grid(mtry = 2, coefReg = seq(0.75, 1, 0.05), coefImp =
                                                              seq(0.4, 0.5, 0.01))
set.seed(1, sample.kind="Rounding")
rrf_fn_cp_tune <- train(resid_reswei ~ fn + cp, method ="RRF", data = res_train,
                                        trControl = control, tuneGrid = rrf_grid)
```

	mtry	coefReg	coefImp
29	2	0.85	0.46

Optimal hyperparameters for the RRF model are mtry = 2, coefReg = 0.85, coefImp = 0.46

After tuning, the ranger model returned lowest RMSE but both models were retained to see if the greater accuracy displayed by the ranger model descreased with the addition of other factors, indicating a different approach may be stronger. The hyperparameter values determined through tuning were discarded as the addition of extra factors could alter the optimal values leading to inaccurate results, however the RMSE values determined were recorded.

Investigation of extra factors At this stage the factors for which there was no visual proof of a relationship to the residual resistance but where a mathematical relationship was possible were added individually, one at a time. Since the precise nature of any relationship, present or otherwise was unknown, this was purely experimental and improvements in RMSE were not assumed. The approach consisted of defining the two model types in the same fashion as that for every other modelling operation so far before adding the factor in question to the model containing the Froude number and Prismatic coefficient. The lowest RMSE for those models with that factor was then recorded. To demonstrate, for the Longitudinal position of the center of buoyancy, the following operation was run.

```
rf_model_ens_exp <- c("ranger", "RRF")
set.seed(1, sample.kind="Rounding")
rf_fits_lcb <- lapply(rf_model_ens_exp, function(model){
  print(model)
  train(resid_reswei ~ fn + cp + lcb, method = model, data = res_train,
                                                          trControl = control)
})

n_rf_exp_models <- seq(1:2)
rf_rmses_lcb <- sapply(n_rf_exp_models, function(m_number){
  min(rf_fits_lcb[[m_number]][["results"]][["RMSE"]])
})

rf_rmses_lcb <- as.data.frame(rf_rmses_lcb) %>%
  mutate(model_name = rf_model_ens_exp)
colnames(rf_rmses_lcb) <- c("RMSE", "model_name")
```

For all other factors not already included in the model this was repeated, with *lcb* replaced by the factor being tested. The results of these experiments are below

RMSE	Model Name	Factors
1.07345732232521	ranger	fn & cp
1.14235647113087	RRF	fn & cp
0.936189376356244	ranger	fn, cp & lcb
0.983656995709385	RRF	fn, cp & lcb
1.13766799848297	ranger	fn, cp & dlr
1.19853863671552	RRF	fn, cp & dlr
1.08095737569748	ranger	fn, cp & bt
1.14480749117976	RRF	fn, cp & bt
1.1194101557532	ranger	fn, cp & lb
1.14203535445489	RRF	fn, cp & lb

Final model build and evaluation on test set With all the above factors except the longitudinal position of the center of buoyancy producing negative effects on prediction accuracy, the model containing only the factors fn, cp, and lcb will be carried forward and tuned for prediction on the test set. The model hyperparameters were tuned using the same method as that for the model containing just Froude number and Prismatic coefficient and as before, the ranger model required no further tuning. The tuned RMSE values below show the ranger model as being the most accurate

method	RMSE
RRF	0.9860142
ranger	0.9361894

Displayed below, the final model was used to predict residual resistance values on the test set and calculate a final RMSE.

```
set.seed(1, sample.kind="Rounding")
ranger_final_fit <- train(resid_reswei ~ fn + cp + lcb, method ="ranger",
                                                  data = res_train, trControl = control)
res_test_pred <- predict(ranger_final_fit, res_test)
```

```
fin_model_RMSE <- RMSE(res_test_pred, res_test$resid_reswei)
```

Results and Discussion

The below table details the calculated RMSEs of the tuned ranger and RRF models for the Froude Number and Prismatic Coefficient and untuned model RMSEs for Froude Number, Prismatic Coefficient and each of the addtional factors.

RMSE	Model Name	Factors
1.07345732232521	ranger	fn & cp
1.14235647113087	RRF	fn & cp
0.936189376356244	ranger	fn, cp & lcb
0.983656995709385	RRF	fn, cp & lcb
1.13766799848297	ranger	fn, cp & dlr
1.19853863671552	RRF	fn, cp & dlr
1.08095737569748	ranger	fn, cp & bt
1.14480749117976	RRF	fn, cp & bt
1.1194101557532	ranger	fn, cp & lb
1.14203535445489	RRF	fn, cp & lb

Looking at these results we can see the improvement in RMSE for both model types with the addition of Longitudinal position of the center of buoyancy, lcb to the models containing Froude Number and Prismatic Coefficient.

There is an increase in RMSE for the Length-Displacement ratio for both ranger and RRF models.

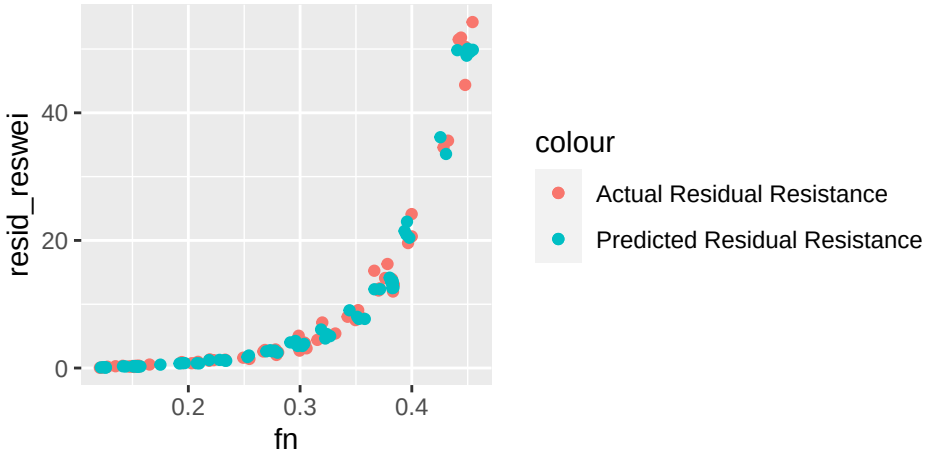
A small decrease in RMSE for the RRF model with Length-Beam ratio, lb is observed, however this improvement was not reflected in the ranger model.

The RRF model containing Beam-Draught ratio, bt, showed a small decrease in RMSE. It could be argued that this small increase could be reduced to near zero or a small decrease by tuning the model, but again, the RMSE value for the ranger model was larger.

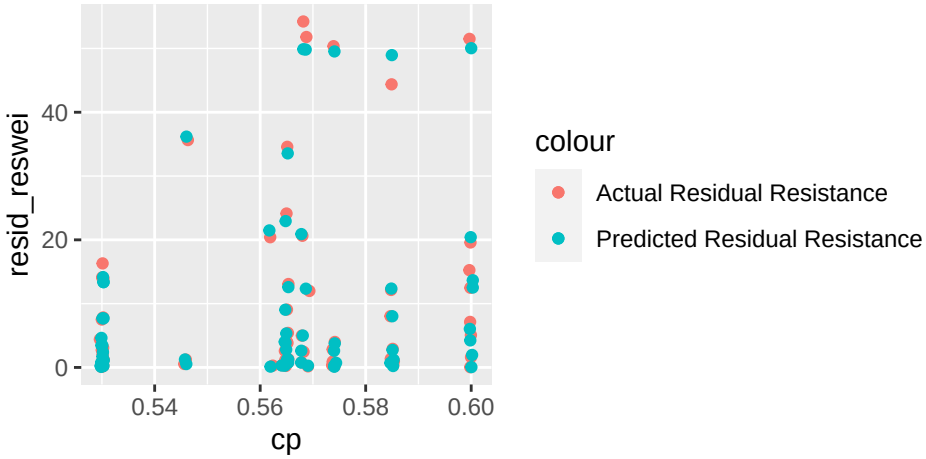
For every factor added, the ranger model was more accurate so in the above cases the models were not tuned. However, the proximity to the previous RMSE could lend weight to the idea that these variables are predictors for Residual Resistance, following the theory presented earlier. This is something that could be investigated with a larger dataset.

The final RMSE calculated from predictions made using the final model was *0.9936*. While an increase is not completely unexpected given the limited sample size it is important to get an idea of the distribution of the differences between the predicted residual resistance and the actual residual resistance to identify any strengths or weaknesses within the model. To that end we will plot the two aganist the three factors used to make the predictions.

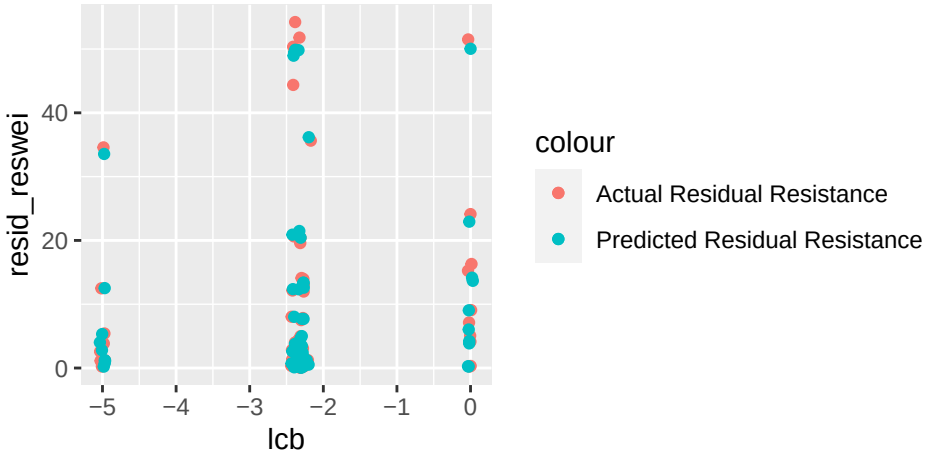
Froude Number



Prismatic Coefficient

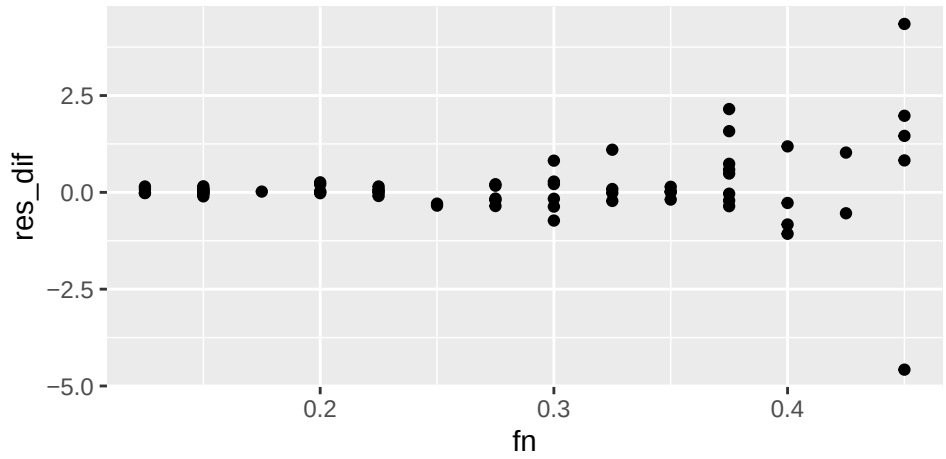


Longitudinal position of the center of buoyancy

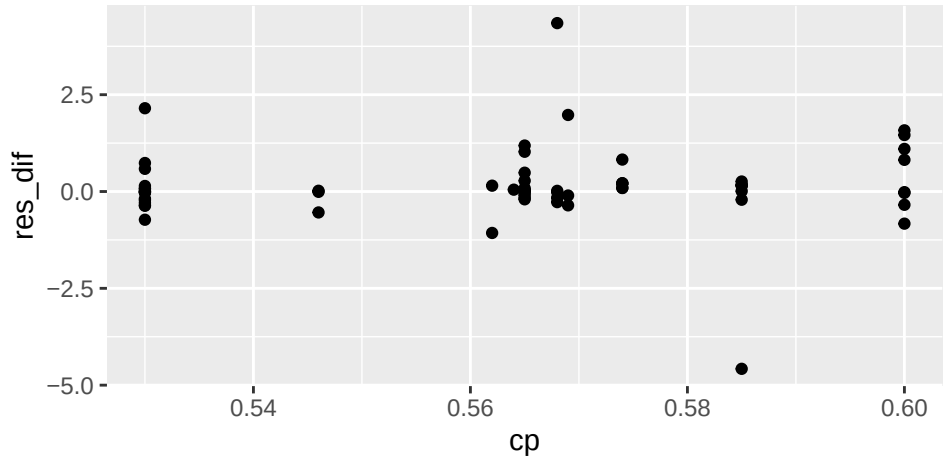


Examining the plots it seems that model accuracy decreases where Froude Number is large. This can be seen clearer by plotting the differences against each variable.

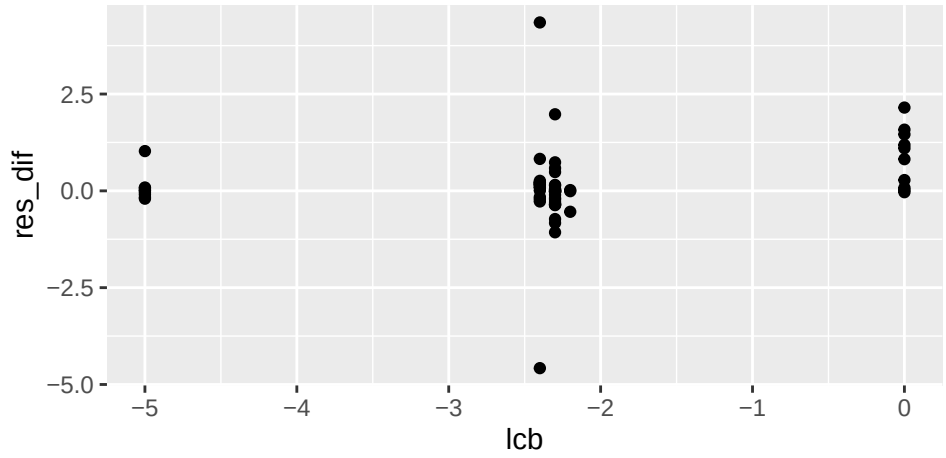
Froude Number



Prismatic Coefficient



Longitudinal position of the center of buoyancy



Looking at the distribution of the differences in the plots it seems the model loses accuracy as Froude Number increases, this seems logical as the magnitude of the residual resistance increases in an exponential manner with an increase in Froude Number. The other factors, Prismatic coefficient, cp, and Longitudinal position

of the center of buoyancy, lcb, show a more even distribution of inaccuracy, indicating the Froude Number is the main driver. Again, this is a pattern shown earlier when exploring the data.

Conclusion

Froude Number, Prismatic Coefficient and Longitudinal position of the center of buoyancy were used to build a model that predicted Residual Resistance. The RMSE of these predictions was 0.9936 and by examining the distribution of the errors it was demonstrated the model performed better at smaller values of Froude Number. With this in mind it could be argued that this could reduce the need to physically produce hull models where the Froude Number, Prismatic Coefficient and Longitudinal position of the center of buoyancy are known.

Moving forward, it could investigation of a larger dataset could allow for the development of a more accurate model and perhaps allow the incorporation of some or all the remaining variables into the prediction model. Additionally, there are other models types that were not produced here and these may produce more accurate results.

References

Dataset URL https://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science