Exploring Regression Models to Predict Certified General Aviation Aircraft Metrics

Liam Frank

Last compiled on 10/12/2023

Contents

1 Introduction

This study will use popular regression techniques to predict both performance and categorical metrics of popular certified general aviation aircraft. The aircraft included in this study are widely used across the world for various duties including business, agricultural application, personal travel, airshow performances, and pleasure flying. This study seeks to harness popular regression techniques to accurately predict aircraft metrics such as stall speed, wingspan, horsepower, and number of engines. With the end goal of exploring the relationships between aircraft features and performance.

1.1 Background

As far as regression analysis in the aviation sector is concerned, the many studies conducted regarding regression analysis tend to focus on predicting commercial aviation flight delay times, flight price, pilot accident rates, and fuel consumption for airliners. All of these topics are directly related to potential cost savings, safety, or gaining a market edge on competitors. The goal of this study has no monetary or safety benefit from the results, the soul purpose being to test relationships between variables that are commonly used to categorize and classify certified general aviation aircraft.

2 The Data

2.1 Source and Data Set

The data set used was sourced from Aircraft Bluebook (1). Aircraft Bluebook has specialized in data collection and the production of data visualizations specifically to the aviation sector for over 65 years. The data set used is comprised of information that was sourced and vetted in house by Aircraft Bluebook, and is currently published and consistently updated on aircraftbluebook.com for public usage. It is currently on rendition 23-04 and was last updated this year, Fall of 2023. The data set is comprised of 22 total variables and 859 total observations. All the aircraft in the data set are certified general aviation aircraft, this implies that no gliders, military, commercial or experimental aircraft were included.

2.2 Variables

The data set is comprised of 22 total variables.

1) Model, the model designation of the aircraft.
2) Company, the production company responsible for producing the aircraft.
3) Engine.Type, this describes the type of engine the aircraft uses, ie. piston, turboprop, jet.
4) Multi.Engine, this contains a factor variable, 1 corresponds to multi-engine, 0 corresponds to single engine.
5) TP.mods, if the aircraft contains any airframe or powerplant modifications or conversions done by a third party manufacturer with an STC from the FAA.
6) THR, foot pounds of thrust, this is only applicable to jet engine aircraft. If the aircraft is piston or turboprop powered then this column was left blank.
7) SHP, horsepower, this is only applicable to piston or turboprop aircraft. If the aircraft is jet engine powered then this column was left blank.
8) Height, measured in feet with inches converted to a decimal.
9) Length, measured in feet with inches converted to a decimal.
10) WS, wing span, measured in feet with inches converted to a decimal.
11) MEW, empty weight, measured in pounds.
12) GW, max gross takeoff weight, measured in pounds.

13) Vmax, max speed measured in knots.
14) Vcruise, cruise speed measured in knots.
15) Stall, stall speed with full flaps extended, measured in knots.
16) ROC, rate of climb, measured in feet per minute.
17) Hmax, operating ceiling, measured in feet.
18) Range, aircraft total range, measured in nautical miles.
19) Vlo, takeoff distance over a 50ft obstacle, measured in feet.
20) Slo, takeoff ground roll, measured in feet.
21) Vl, landing distance over a 50ft obstacle, measured in feet.
22) Sl, landing ground roll, measured in feet.

2.3 Observations

The data set is comprised of 859 total observations. Highlighting the first observation we can see… The aircraft is an Aeronca 15 AC Sedan. It is a single engine, non modified, piston powered aircraft producing a total of 145 horsepower. The Aeronca 15 AC Sedan measure 25.25 feet long, 10.25 feet tall, and has a wing span of 37.4 feet. It has an empty weight of 1180 pounds and a max gross takeoff weight of 2050 pounds. It has a max speed of 104 knots, cruises at 91 knots, and stalls with full flaps extended at 46 knots. The max operating ceiling for the Aeronca is 13,000 feet, and it gets there at a max rate of climb of 450 feet per minute. With the landing distance over a 50 foot obstacle being 1300 feet, and the takeoff distance over a 50 foot obstacle being 900 feet, it gets in the air rather quickly.

```
##      ï..Model Company Engine.Type Multi.Engine TP.mods THR SHP Length Height
## 1 15 AC Sedan Aeronca      Piston            0   FALSE  NA 145  25.25  10.25
##         WS    FW  MEW   GW Vmax Vcruise Stall  Hmax Hmax..One. ROC ROC..One.
## 1 37.41667 241.2 1180 2050  104      91    46 13000         NA 450        NA
##   Vlo Slo   Vl Sl Range
## 1 900  NA 1300 NA   370
```

2.4 Cleaning

Most of the cleaning regarding the data set was done in excel, with the post exploratory analysis cleaning, splitting, and transforming done in R. The first task for cleaning regarded the SHP (horsepower) and THR (thrust) variables. The data set recorded thrust and horsepower per engine, not the total. For the analysis to be conducted, it was essential that total horsepower for the aircraft was recorded. This was remedied by multiplying the previous recorded horsepower per engine, by the total number of engines on the aircraft. Giving us a total horsepower figure suitable for analysis. This same process was then repeated for the thrust measurements in regards to jet engine aircraft.

3 Methodology

For analysis purposes, a sub data set of the original data set was created containing only propeller driven aircraft. So this second data set was comprised of both piston and turboprop powered aircraft but not jet engine powered aircraft. This was done because after exploratory analysis was started in order to combat some discrepancies in the data. Because the design characteristics are so vastly different for general aviation propeller and jet aircraft, for an accurate predictive analysis it was essential to create a second data set containing only one propulsion type. A train and test set was then created for both the original data set and the propeller data set. The train and test sets used a standard 70:30 split with 70% of the data be allotted to the training set. This was done to test model accuracy and minimize the risk of over fitting.

3.1 Data Tranformations

Also for analysis purposes, the Multi.Engine variable was converted into a factor in order to run a logistic regression. Before being converted to a factor, if the aircraft featured multiple engines then a TRUE logical value was recorded. If the aircraft did not feature multiple engines then an NA logical value was recorded. When converting into a factor, 1 was assigned to multi engine aircraft and 0 was assigned to single engine aircraft. During analysis boxcox transformations were completed and the transformed variable values were stored under names denoted boxcox in there respective data set.

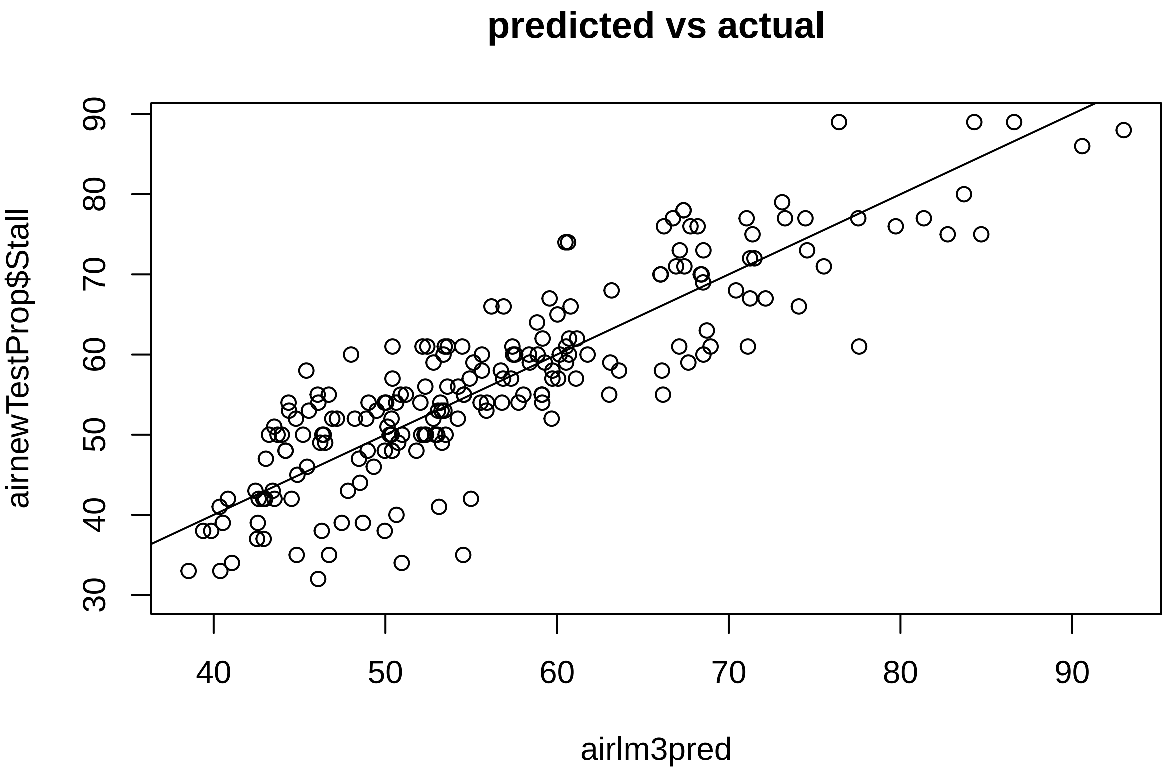
3.2 Types of Models

This study is consists of two different types of regression models. Two different iterations of a linear regression model, a second linear regression model with two iterations, a third linear regression model, and finally a logistic regression model. These models were selected both on the inherent makeup of the data, as well as linear relationships that were found in exploratory analysis through graphing.

3.2.1 Model 1 Using Linear Regression to Predict Stall Speed

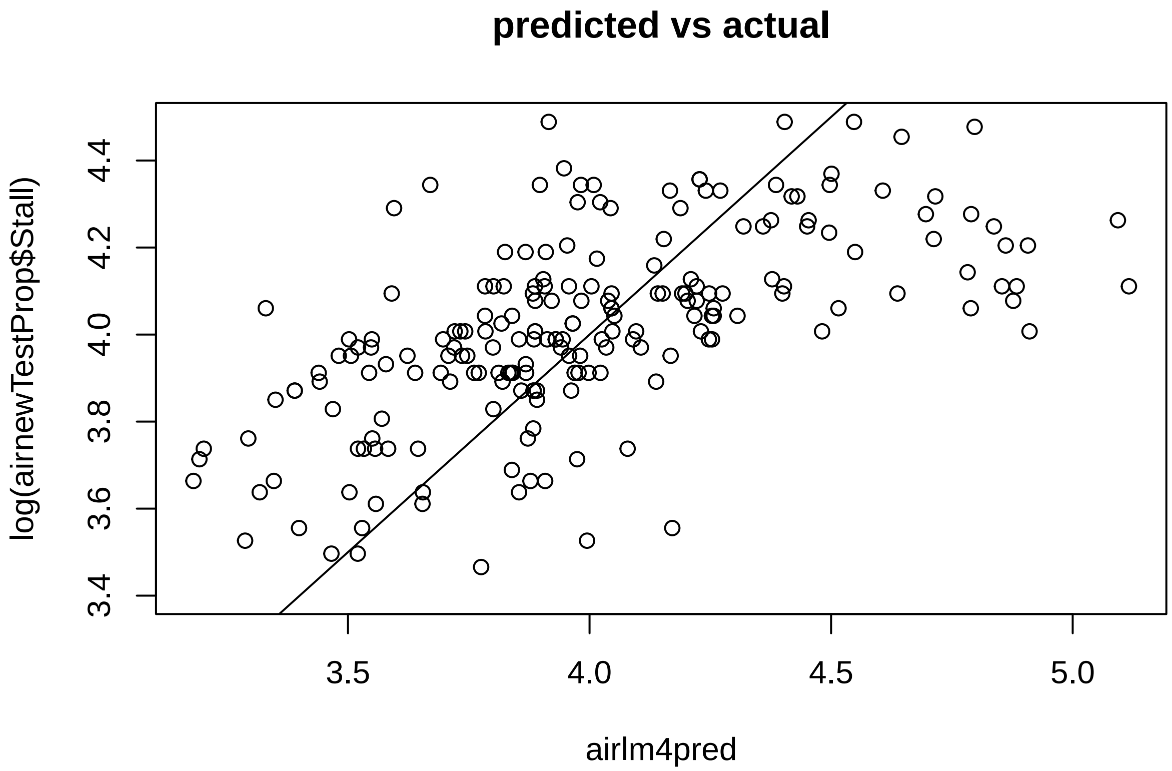
The first model created utilized the data set containing only propeller driven aircraft. It used stall speed as the predicted variable with horsepower, wing span, max speed, and length as explanatory variables, all of which are statistically significant with P values of less than 0.05. The first iteration of the model can be seen below. For both model iterations the intercept was forced through zero, and both models were trained using the training data set developed for propeller aircraft and tested on the testing data set developed for propeller driven aircraft.

```
##
## Call:
## lm(formula = Stall ~ SHP + WS + Vmax + Length - 1, data = airnewTrainProp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.5894  -3.4152   0.5753   4.2608  17.4712
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## SHP    -0.005136   0.001077  -4.770 2.47e-06 ***
## WS      0.278373   0.070044   3.974 8.18e-05 ***
## Vmax    0.133244   0.008001  16.654  < 2e-16 ***
## Length  0.848866   0.112985   7.513 2.97e-13 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6.284 on 467 degrees of freedom
##   (21 observations deleted due to missingness)
## Multiple R-squared:  0.9886, Adjusted R-squared:  0.9885
## F-statistic: 1.009e+04 on 4 and 467 DF,  p-value: < 2.2e-16
```



The second iteration of the model taking the log transformation of Stall Speed can be seen below.

```
##
## Call:
## lm(formula = log(Stall) ~ SHP + WS + Vmax + Length - 1, data = airnewTrainProp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43047 -0.12018  0.04643  0.21592  1.04300
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## SHP    -1.557e-03  6.104e-05 -25.506  < 2e-16 ***
## WS      5.583e-02  3.971e-03  14.059  < 2e-16 ***
## Vmax    6.152e-03  4.535e-04  13.565  < 2e-16 ***
## Length  5.087e-02  6.405e-03   7.942  1.5e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3562 on 467 degrees of freedom
##   (21 observations deleted due to missingness)
## Multiple R-squared:  0.9922, Adjusted R-squared:  0.9922
## F-statistic: 1.495e+04 on 4 and 467 DF,  p-value: < 2.2e-16
```

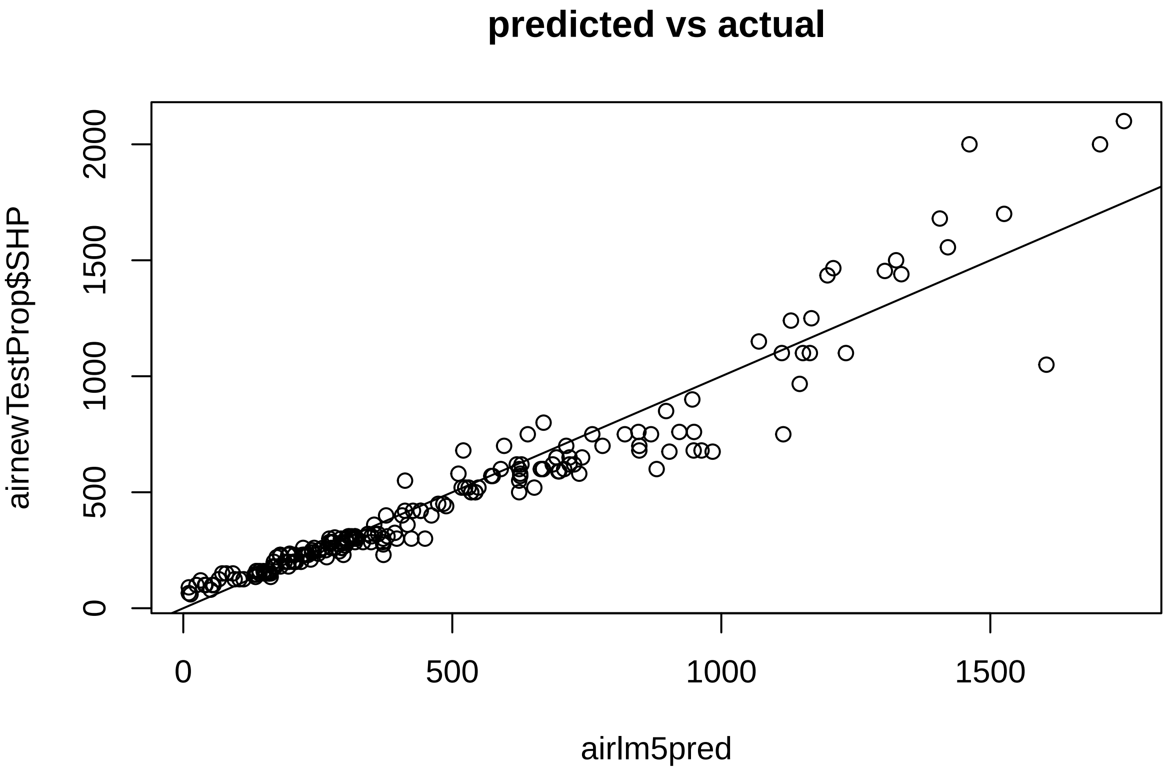


As seen from the summary results. the second iteration of model one containing the log transformation of Stall Speed resulted in a higher adjusted R squared, .99 as compared to .98, as well as a lower residual standard error, .35 as compared to 6.28. When examining the predicted vs actual plots for both model iterations, the first iteration containing no transformation to stall speed results in a more linear fit that is condensed closer to the plotted 45 degree abline. The residual vs fit, and normal qq plots can be seen in the appendix of the paper.

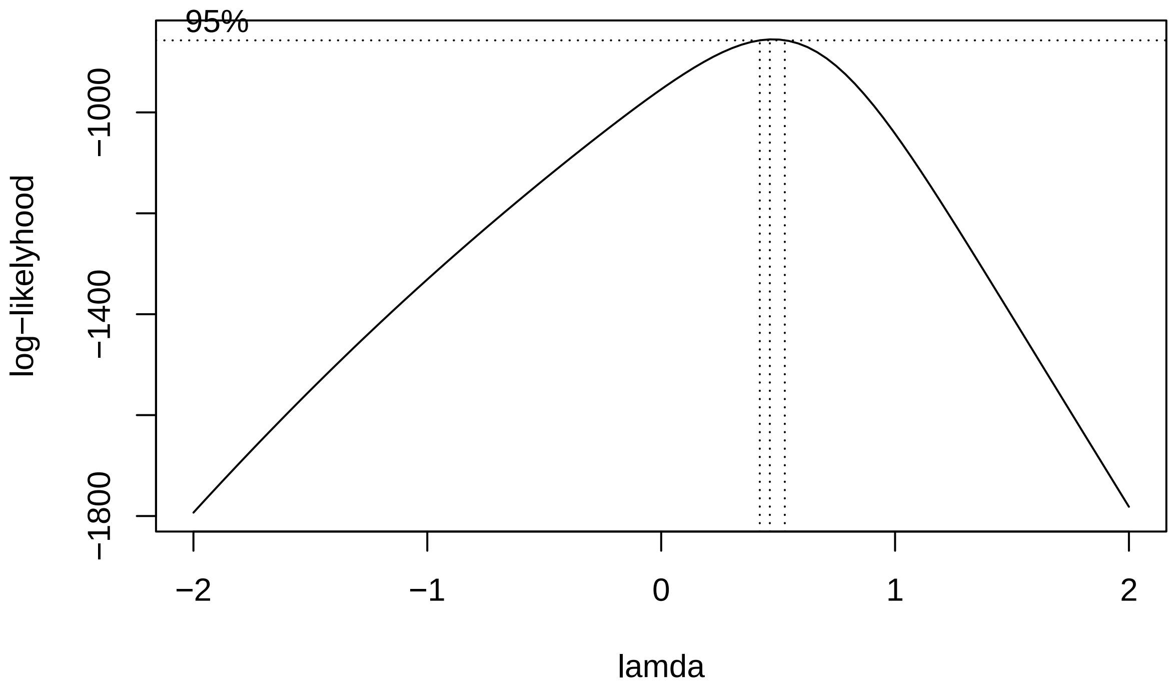
3.2.2 Model 2 Using Linear Regression to Predict Horsepower

The second model created utilized the data set containing only propeller driven aircraft. It used horsepower as the predicted variable with length, and max gross takeoff weight as explanatory variables, both of which are statistically significant with P values of less than 0.05. The first iteration of the model can be seen below. For both model iterations the intercept was forced through zero, and both models were trained using the training data set developed for propeller aircraft and tested on the testing data set developed for propeller driven aircraft.

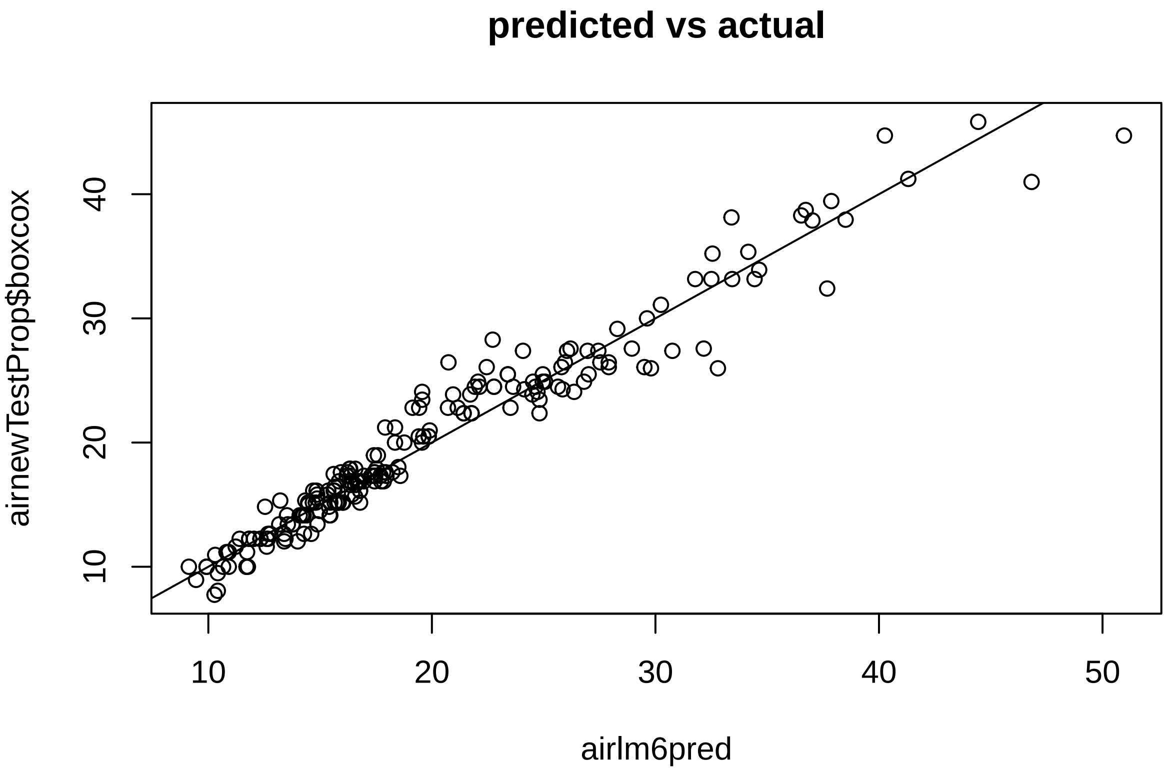
```
##
## Call:
## lm(formula = SHP ~ Length + GW - 1, data = airnewTrainProp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -554.30  -34.68   -4.15   27.50  870.02
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## Length -7.670519   0.476956  -16.08   <2e-16 ***
## GW      0.148901   0.002588   57.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 113.8 on 487 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.9662, Adjusted R-squared:  0.966
## F-statistic:  6952 on 2 and 487 DF,  p-value: < 2.2e-16
```



The second iteration of model two and boxcox transformation of the horsepower prediction variable can be seen below.



```
##
## Call:
## lm(formula = boxcox ~ Length + GW - 1, data = airnewTrainProp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9284 -0.8004 -0.0035  1.0393 12.9501
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## Length 3.539e-01  8.436e-03   41.95   <2e-16 ***
## GW     2.066e-03  4.578e-05   45.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.012 on 487 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.9913, Adjusted R-squared:  0.9912
## F-statistic: 2.759e+04 on 2 and 487 DF,  p-value: < 2.2e-16
```

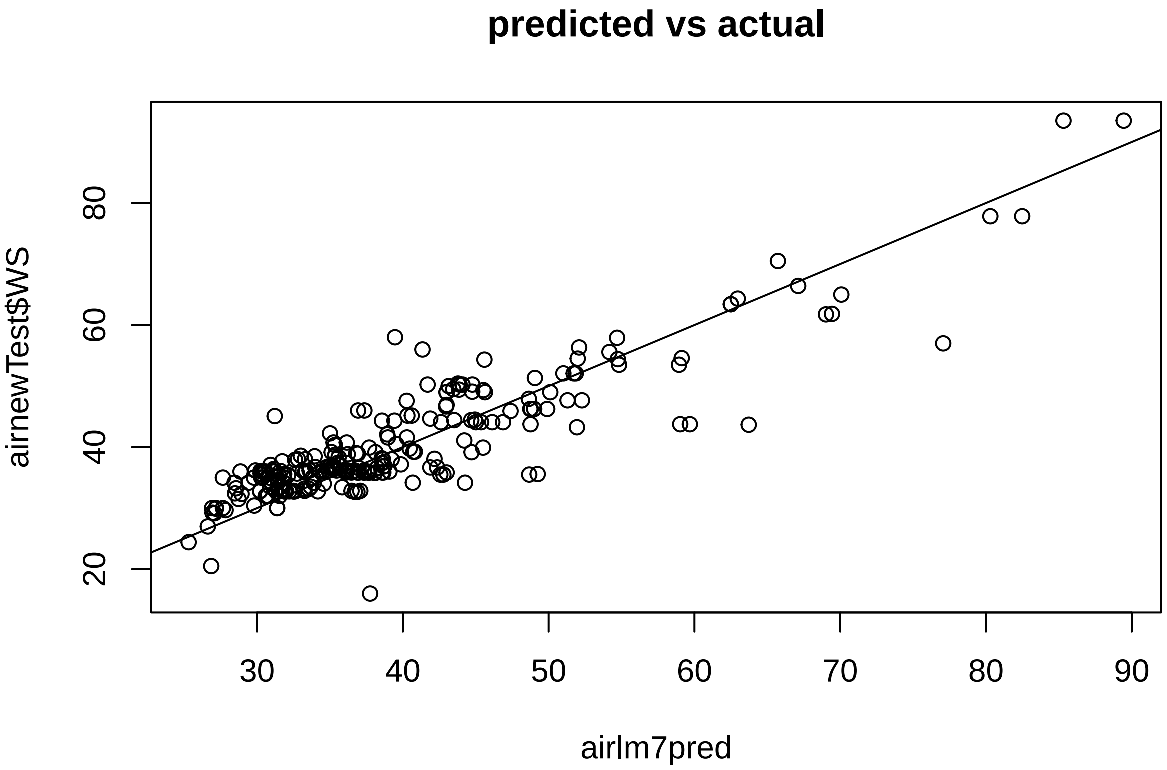


As seen from the summary results, The second iteration of model 2 containing the boxcox transformation of the square root of horsepower resulted in a higher adjusted R squared, .99 as compared to .96, as well as a lower residual standard error, 2.01 as compared to 113.8. The predicted vs actual plots also convey that the second iteration was a better fit, although the predicted values for both plots fit the 45 degree abline closely. That being said, the values towards the middle are of a more linear nature in the plot for the second iteration than that of first. The residual vs fit, and normal qq plots can be seen in the appendix of the paper.

3.2.3 Model 3 Using Linear Regression to Predict Wing Span

The third model created utilized the data set containing only propeller driven aircraft. It used wing span as the predicted variable with length, max gross takeoff weight,and max speed as explanatory variables, both of which are statistically significant with P values less of than 0.05. For the model the intercept was forced through zero, the model was trained using the training data set developed for propeller aircraft and tested on the testing data set developed for propeller driven aircraft.

```
##
## Call:
## lm(formula = WS ~ GW + Length + Vmax - 1, data = airnewTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.267  -1.507   1.007   4.106  22.974
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## GW     -5.020e-04  1.965e-05  -25.55   <2e-16 ***
## Length  1.633e+00  2.715e-02   60.14   <2e-16 ***
## Vmax   -4.479e-02  3.534e-03  -12.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.581 on 562 degrees of freedom
##   (36 observations deleted due to missingness)
## Multiple R-squared:  0.9835, Adjusted R-squared:  0.9834
## F-statistic: 1.114e+04 on 3 and 562 DF,  p-value: < 2.2e-16
```



As seen from the summary results, model 3 contained an adjusted R squared of .98 and a residual standard error of 5.581. Both explanatory variables contained significant P values. The predicted vs actual plot resembles a linear structure that closely follows the plotted abline. As wing span increases, more variance in the predicted values can be seen. The residual vs fit, and normal qq plots can be seen in the appendix of the paper.

3.2.4 Model 4 Using Logistic Regression to Predict and Classify Number of Engines

The fourth model created utilized the data set containing only propeller driven aircraft. It used Multi.Engine as the binary predicted variable with length, and max gross takeoff weight as explanatory variables, both of which are statistically significant with P values of less than 0.05. The model used the logit link function, and the model was trained using the training data set developed for propeller aircraft and tested on the testing data set developed for propeller driven aircraft.

```
##
```

```
## Call:
## glm(formula = Multi.Engine ~ GW + Length, family = binomial(link = "logit"),
##     data = airnewTrainProp)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8565  -0.4501  -0.2357   0.2360   2.4366
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.063e+01  1.628e+00  -6.525 6.79e-11 ***
## GW           6.623e-04  1.500e-04   4.416 1.00e-05 ***
## Length       2.432e-01  7.168e-02   3.393 0.000691 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 640.14  on 488  degrees of freedom
## Residual deviance: 290.53  on 486  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 296.53
##
## Number of Fisher Scoring iterations: 6
```

```
##       GW   Length
## 1.000663 1.275332
```

```
## [1] 6.044299e-77
```

```
## [1] 0.5461522
```

```
##
##     FALSE      TRUE
## 0.1753555 0.8246445
```

As seen in the results summary, the logistic model contained a AIC of 296, with a null deviance of 640 and residual deviance of 290. The logistic model used the logit link function as it produced a more accurate model in exploratory testing when compared to the probit link function. The coefficient estimates where 1.000663 for max gross takeoff weight and 1.275332 for length. This implies that for a one unit increase in max gross takeoff weight, the log odds of the aircraft being a twin engine increases 1.000663. For a one unit increase in length, the log odds of the aircraft being a twin engine increases 1.275332. The chi sq lack of fit for the model was 6.044299e-77 indicating that the model is a good fit. The model had a McFadden Pseudo R squared of .546. Finally, when testing the logistic model using the test set, we can see it accurately classified an aircraft as multi engine or single engine 82.4% of the time.

3.3 Methods for model selection

The iteration chosen for each model was based on a combination of factors in regards to each individual model. The main methods for the linear regression model selections were the adjusted R squared value, residual standard error, and an examination of the predicted vs actual plots. When choosing which link function to use for the logistic model, AIC, null deviance, residual deviance, McFaddens psuedo R squared, chi squared and accuracy percentage of predictions on the test set were all taken into account.

4 Results

To dive into the results for each linear regression model, this section will highlight adjusted R squared values, as well as coefficient estimates to draw conclusions from the model. For the logistic regression model, McFadden's Psuedo R squared, prediction accuracy and coefficient estimates will be examined.

4.1 Model 1 Using Linear Regression to Predict Stall Speed

The first iteration of model 1 contained an adjusted R squared value of .98. Leading to the conclusion that 98% of the variance in stall speed can be attributed to horsepower, wing span, max speed, and length. The coefficient estimates signify the change in the mean of stall speed for a one unit increase in the selected explanatory variable. For a one unit increase in horsepower, a decrease of .005 knots can be expected in stall speed. For a one unit increase in wing span, an increase of .278 knots can be expected in stall speed. For a one unit increase in max speed, an increase of .133 knots can be expected in stall speed. For a one unit increase in length, an increase of .848 knots can be expected in stall speed.

4.2 Model 2 Using Linear Regression to Predict Horsepower

The second iteration of model 2 featuring the square root transformation of the horsepower variable contained an adjusted R squared value of .99. Suggesting that 99% of the variance in horsepower can be attributed to only length and max gross takeoff weight. For a one unit increase in length, an increase of .35 can be seen in the square root of horsepower. For a one unit increase in max gross takeoff weight, an increase of .00266 can be seen in the square root of horsepower.

4.3 Model 3 Using Linear Regression to Predict Wing Span

Model 3 contained an adjusted R squared of .98. Suggesting that 98% of the variance in wing span can be attributed to length, max gross takeoff weight, and max speed. For a one unit increase in length, an increase of 1.633 feet can be seen in wing span. For a one unit increase in max gross takeoff weight, a decrease of .0005 feet can be seen in wing span. For a one unit increase in max speed, a decrease of .0479 feet can be seen in wing span.

4.4 Model 4 Using Logistic Regression to Predict and Classify Number of Engines

The logistic regression model featured a McFadden's Psuedo R squared value of .546. Suggesting that 54.6% of the variance in whether or not an aircraft featured multiple engines could be attributed to max gross takeoff weight and length. The coefficient estimates for a logistic regression signify log odds. For a one unit increase in max gross takeoff weight, the log odds of the aircraft being multi engine increase 1.000663. For a one unit increase in length, the log odds of the aircraft being multi engine increase 1.275332. The logistic regression model also featured an accuracy rate of 82.4% when tested on the test set of data, this leads to the conclusion that the model correctly classifies an aircraft a little of 4/5 of the time.

5 Discussion

5.1 Final model interpolation

As summarized above for each regression model, coefficient estimates and adjusted R squared values were quintessential to this study reaching its goal of harnessing popular regression techniques to accurately predict aircraft metrics such as stall speed, wingspan, horsepower, and number of engines with the end goal of exploring the relationships between aircraft features and performance. Using coefficient estimates to find change in the mean of the dependent variable for a one unit increase in the explanatory variable allowed relationships to be established and examined between the prediction and explanatory variables. R squared was also examined above in the results section to determine variance in the prediction variables as a result of the explanatory variables. The combination of these two metrics allowed the conclusions to be drawn from the analysis.

5.2 Use of Model

Also described in the introduction, use cases for these models feature no monetary gain or safety increase. The sole purpose was to explore the relationships between performance metrics and design characteristics. This leads to the use case of establishing a baseline relationships between the variables used in each regression model. This can be used for early aerospace engineering design exploration. Take model 3 for example, if a company is attempting to engineer a new aircraft and they know the target gross weight, max speed, and length. They can receive an accurate predicted wing span for the new aircraft based on the industry standard established in the linear regression model. Obviously, in aerospace engineering a lot goes into calculating the necessary wing span for ideal flight characteristics. More than just gross weight, max speed, and length, but it gives the engineers a good baseline for design parameters and possible size constraints.

6 Furture Work

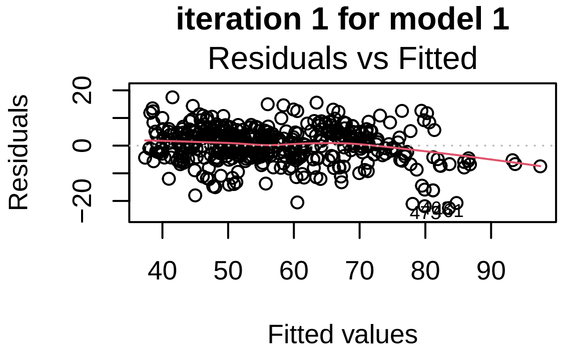
This study used a data set that was comprised of 859 total observations from all kinds of general aviation aircraft. The models were built on a training subset of this data set that contained only propeller driven aircraft and 492 total observations, with 22 variables. Future work to expand upon the findings of this study should consists of a data set that is more specific to a certain category of general aviation aircraft, such as only 4 seat fixed gear aircraft. The new data set should also consist of more variables, such as wing area, gear type, drag coefficient, and price just to name a few. With the addition of certain variables specific to an aircraft design type, more accurate predictions could be made especially in regards to performance metrics. With the addition of price to the data set, a model could be built to predict price of an aircraft based on customer mission needs. A regression model could be tailored to the customers needs for an individual aircraft. This could give the customer an accurate price prediction on an aircraft that meets all his or her specified requirements, allowing the customer to better manage their budget, and overall increase new involvement in general aviation when shrinking participation numbers have been a substantial industry problem in recent years.

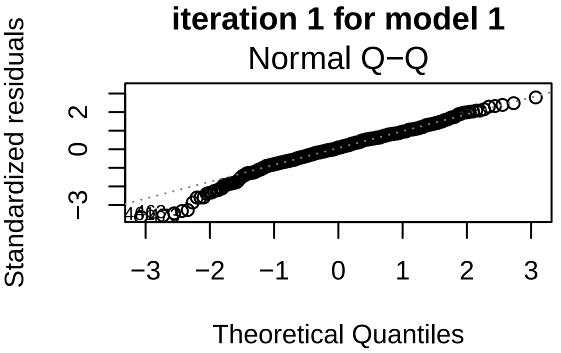
7 References

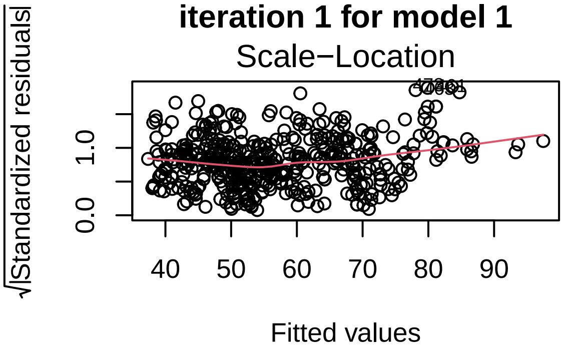
(1) "Performance & Specifications." Aircraftbluebook.com, aircraftbluebook.com/Tools/ABB/ShowSpecifications.do.

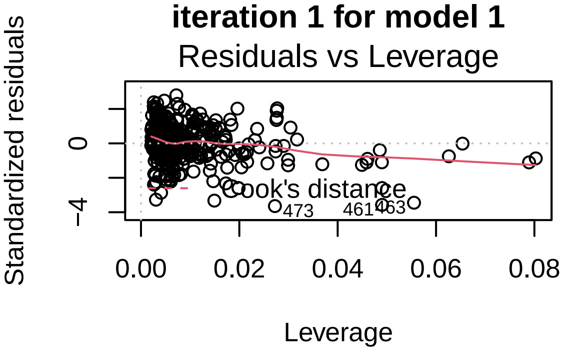
8 Appendix

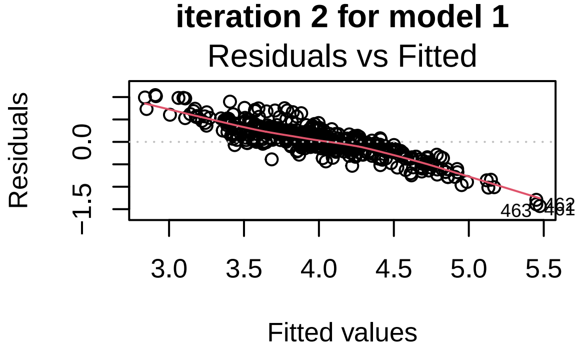
8.1 Linear Regression Model 1 Graphs

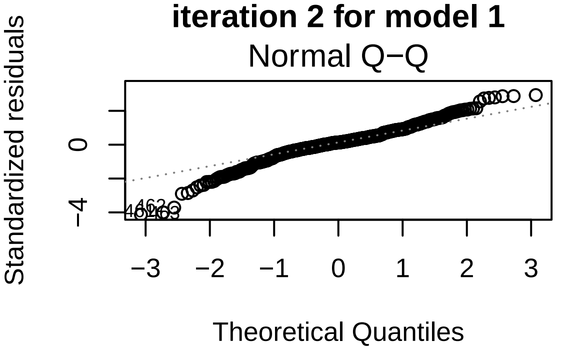


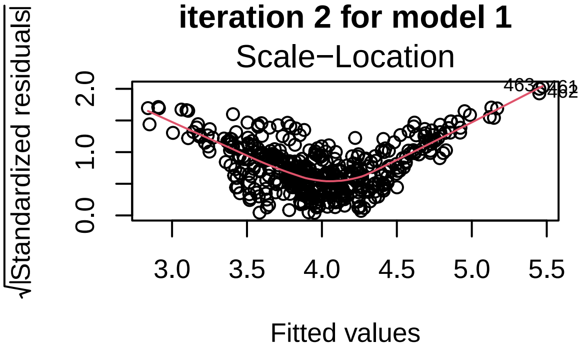


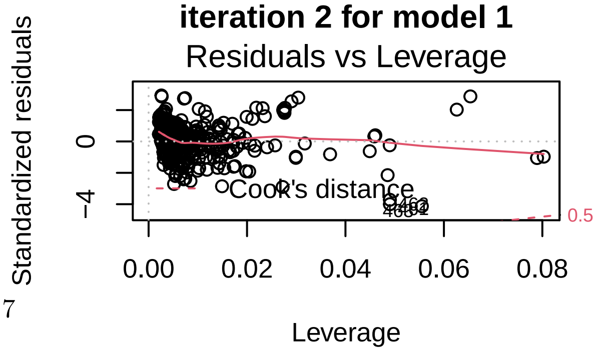




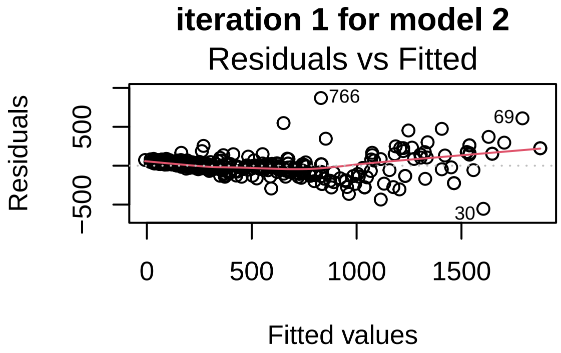


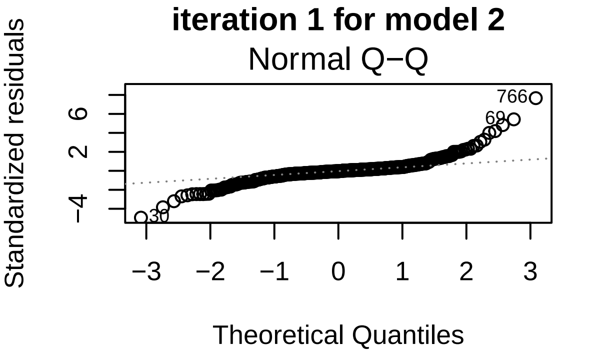


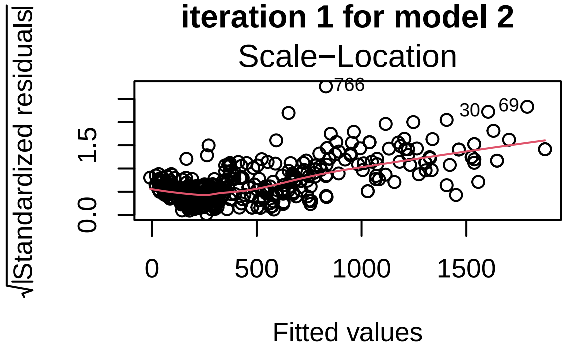


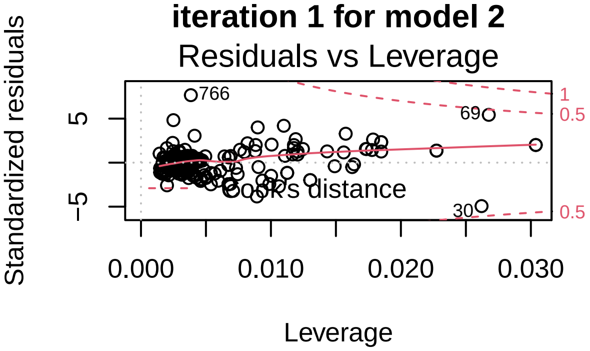


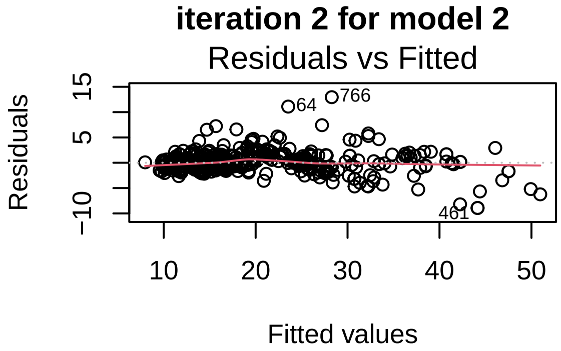
8.2 Linear Regression Model 2 Graphs

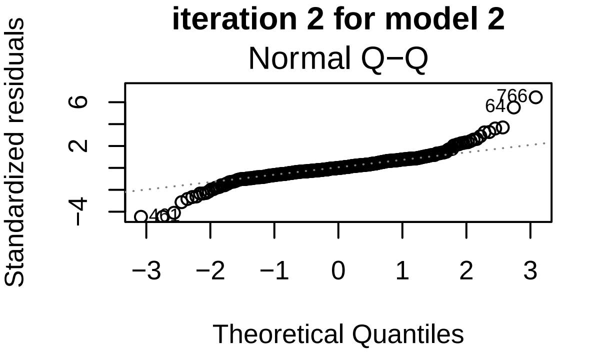

iteration 1 for model 2
Residuals vs Fitted

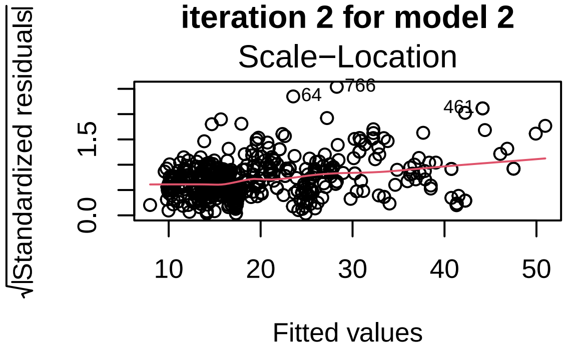

iteration 1 for model 2
Normal Q−Q

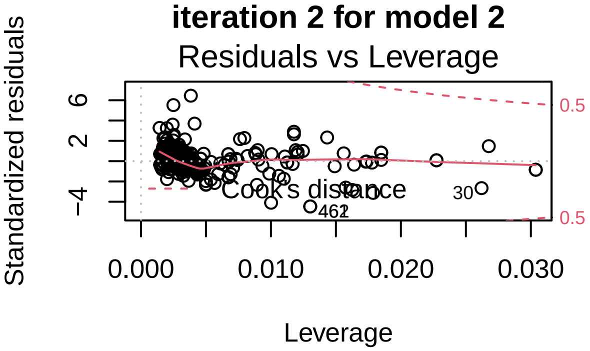

iteration 1 for model 2
Scale−Location


iteration 1 for model 2
Residuals vs Leverage

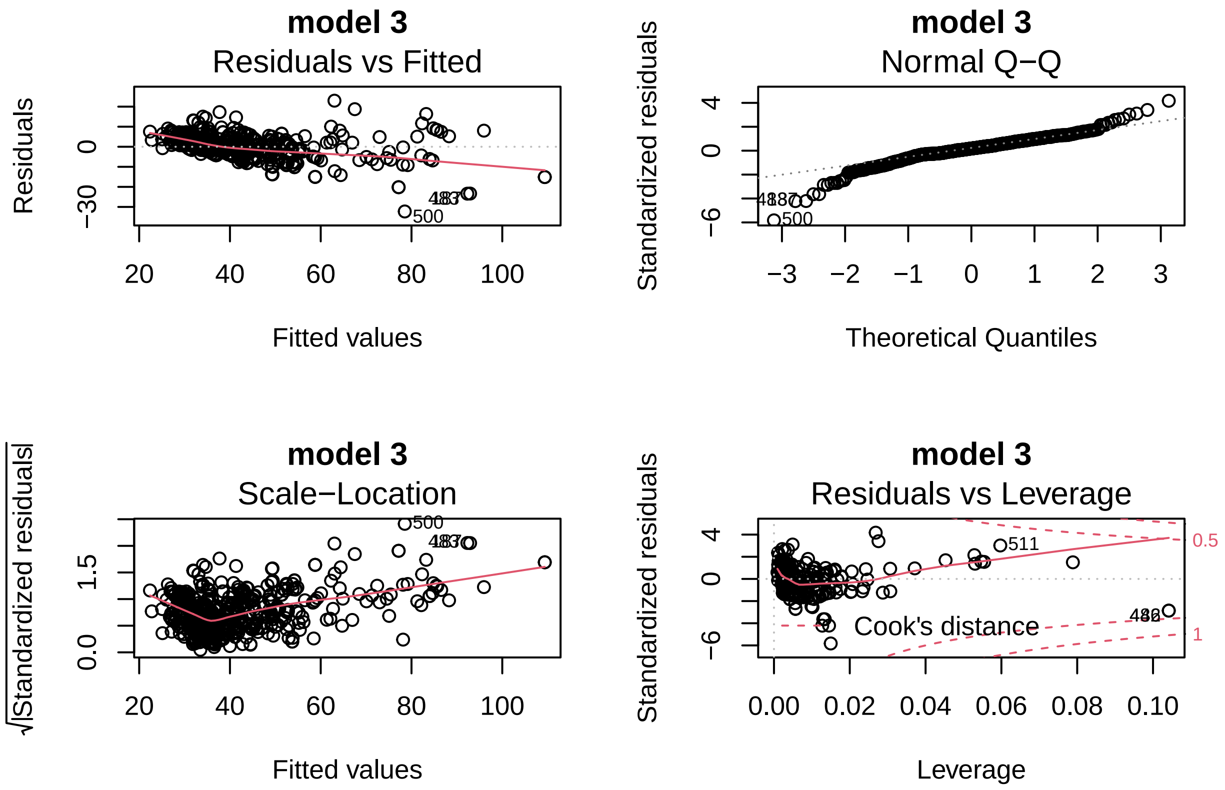

iteration 2 for model 2
Residuals vs Fitted


iteration 2 for model 2
Normal Q−Q


iteration 2 for model 2
Scale−Location


iteration 2 for model 2
Residuals vs Leverage

8.3 Linear Regression Model 3 Graphs



8.4 Code

```r
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
# tidyverse was used in exploratory analysis and visualization
library(dplyr)
# dplyr was used for cleaning data in a data frame
library(MASS)
# MASS was used for the logistic regression model
library(ggplot2)
# ggplot2 was used for visualizations of both exploratory analysis and graphing linear regression predi

set.seed(123)
airread<- read.csv("general aviation data cleaned.csv")
airnew1 <- data.frame(airread)
str(airnew1)

### Removing observation 529 as it seems to be an outlier in exploratory testing
airnew1[529,]
airnew <- airnew1 %>%  filter(!row_number() %in% c(529))
airnew[529,]

### Converting multi engine to factor for logistic regression
airnew$Multi.Engine <- ifelse(is.na(airnew$Multi.Engine), 0, ifelse(airnew$Multi.Engine, 1, NA))

### Creating separate data set for propelleraircraft
airprop <- airnew[!(airnew$Engine.Type == "Jet"),]
str(airnew)
str(airprop)
airnew$Multi.Engine

### Test and Train Data Split for both data sets
set.seed(123)
airnewtrain<- sample(nrow(airnew), 0.70*nrow(airnew), replace=FALSE)
airnewTrain <- airnew[airnewtrain,]
airnewTest <-  airnew[-airnewtrain,]

airproptrain<- sample(nrow(airprop), 0.70*nrow(airprop), replace=FALSE)
airnewTrainProp <- airprop[airproptrain,]
airnewTestProp <-  airprop[-airproptrain,]
airnew
airprop
airnew[1,]
air_lm3 <- lm(Stall~ SHP + WS + Vmax + Length-1,data=airnewTrainProp)
summary(air_lm3)

airlm3pred<- predict(air_lm3, newdata = airnewTestProp)
dataair3 <- data.frame(actual= airnewTestProp$Stall^2, predicted= airlm3pred)

par(mfrow=c(1,1))
plot(airlm3pred, airnewTestProp$Stall, main = "predicted vs actual")
abline(a=0,b=1)
air_lm4 <- lm(log(Stall) ~ SHP + WS + Vmax + Length-1, data = airnewTrainProp)
```

```r
summary(air_lm4)

airlm4pred<- predict(air_lm4, newdata = airnewTestProp)
dataair4 <- data.frame(actual= log(airnewTestProp$Stall), predicted= airlm4pred)

par(mfrow=c(1,1))
plot(airlm4pred, log(airnewTestProp$Stall), main = "predicted vs actual")
abline(a=0,b=1)
air_lm5<-lm(SHP ~ Length + GW-1, data = airnewTrainProp)
summary(air_lm5)

airlm5pred<- predict(air_lm5, newdata = airnewTestProp)
dataair5 <- data.frame(actual= airnewTestProp$SHP, predicted= airlm5pred)

par(mfrow=c(1,1))
plot(airlm5pred, airnewTestProp$SHP, main = "predicted vs actual")
abline(a=0,b=1)
boxcox(air_lm5, lamda = seq(-2,2,1/10), plotit = TRUE, eps =  1/50, xlab = expression(lamda), ylab ="log

airnewTrainProp$boxcox <- (airnewTrainProp$SHP)^0.5
airnewTestProp$boxcox <- (airnewTestProp$SHP)^0.5

air_lm6<- lm(boxcox ~ Length + GW-1, data = airnewTrainProp)
summary(air_lm6)

airlm6pred<- predict(air_lm6, newdata = airnewTestProp)
dataair6 <- data.frame(actual= airnewTestProp$boxcox, predicted= airlm6pred)

par(mfrow=c(1,1))
plot(airlm6pred, airnewTestProp$boxcox, main = "predicted vs actual")
abline(a=0,b=1)
air_lm7<- lm(WS ~ GW + Length + Vmax-1, data = airnewTrain)
summary(air_lm7)

airlm7pred<- predict(air_lm7, newdata = airnewTest)
dataair7 <- data.frame(actual= airnewTest$WS, predicted= airlm7pred)

par(mfrow=c(1,1))
plot(airlm7pred, airnewTest$WS, main = "predicted vs actual")
abline(a=0,b=1)
multi_probit <- glm(Multi.Engine~GW+Length,family=binomial(link="logit"), data=airnewTrainProp)
summary(multi_probit)

df1<-multi_probit$coefficients[2:3]
exp(df1)

chisq_probit<-multi_probit$null.deviance-multi_probit$deviance
p_probit<-length(multi_probit$coefficients)-1
dchisq(chisq_probit,df=p_probit)

ll.null <- multi_probit$null.deviance/-2
ll.proposed <- multi_probit$deviance/-2
```

```r
(ll.null - ll.proposed) / ll.null

set.seed(1234)
pVals <- predict(multi_probit, airnewTestProp, type = 'response')
labelpVals <- ifelse(pVals> 0.5, '1', '0')

table(labelpVals == airnewTestProp$Multi.Engine)/length(airnewTestProp$Multi.Engine)
par(mfrow=c(2,2))
plot(air_lm3, main = "iteration 1 for model 1")
plot(air_lm4, main = "iteration 2 for model 1")
par(mfrow=c(2,2))
plot(air_lm5, main = "iteration 1 for model 2")
plot(air_lm6, main = "iteration 2 for model 2")
par(mfrow=c(2,2))
plot(air_lm7, main = "model 3")

### Libraries Used for analysis
library(tidyverse)
library(MASS)
library(dplyr)
library(magrittr)

### Importing cleaned data from excel as a csv
airread<- read.csv("general aviation data cleaned.csv")
airnew1 <- data.frame(airread)
str(airnew1)

### Removing observation 529 as it seems to be an outlier in exploratory testing
airnew1[529,]
airnew <- airnew1 %>%  filter(!row_number() %in% c(529))
airnew[529,]

### Converting multi engine to factor for logistic regression
airnew$Multi.Engine <- ifelse(is.na(airnew$Multi.Engine), 0, ifelse(airnew$Multi.Engine, 1, NA))

### Creating separate data set for propelleraircraft
airprop <- airnew[!(airnew$Engine.Type == "Jet"),]
str(airnew)
str(airprop)
airnew$Multi.Engine

### Test and Train Data Split for both data sets
set.seed(123)
airnewtrain<- sample(nrow(airnew), 0.70*nrow(airnew), replace=FALSE)
airnewTrain <- airnew[airnewtrain,]
airnewTest <-  airnew[-airnewtrain,]

airproptrain<- sample(nrow(airprop), 0.70*nrow(airprop), replace=FALSE)
airnewTrainProp <- airprop[airproptrain,]
airnewTestProp <-  airprop[-airproptrain,]


### Exploratory analysis using ggplot to graph relationships
```

```r
ggplot(airnew, aes(x=Length, y=Stall))+geom_point()+geom_smooth(method = 'lm')
ggplot(airnew, aes(x=SHP,y=Stall))+geom_point()+geom_smooth(method = 'lm')
ggplot(airnew, aes(x=Length, y=Vmax))+geom_point()+geom_smooth(method = 'lm')
ggplot(airnew, aes(x=Vcruise, y=Vmax))+geom_point()+geom_smooth(method = 'lm')
ggplot(airnew, aes(x=Stall, y=Vmax))+geom_point()+geom_smooth(method = 'lm')
ggplot(airnew, aes(x=SHP, y=WS))+geom_point()+geom_smooth(method = 'lm')
ggplot(airnew, aes(x=WS, y=Length))+geom_point()+geom_smooth(method = 'lm')

ggplot(airprop, aes(x=Length, y=Stall))+geom_point()+geom_smooth(method = 'lm')
ggplot(airprop, aes(x=SHP,y=Stall))+geom_point()+geom_smooth(method = 'lm')
ggplot(airprop, aes(x=SHP,y=Stall^2))+geom_point()+geom_smooth(method = 'lm')
ggplot(airprop, aes(x=Length, y=Vmax))+geom_point()+geom_smooth(method = 'lm')
ggplot(airprop, aes(x=Vcruise, y=Vmax))+geom_point()+geom_smooth(method = 'lm')
ggplot(airprop, aes(x=Stall, y=Vmax))+geom_point()+geom_smooth(method = 'lm')
ggplot(airprop, aes(x=SHP, y=WS))+geom_point()+geom_smooth(method = 'lm')
ggplot(airprop, aes(x=WS, y=Length))+geom_point()+geom_smooth(method = 'lm')

#### Exploratory Stall speed models
str(airnewTrain)
exp1<- lm(Stall~Vmax+GW+WS+Sl+Vl+Slo+ROC+FW+SHP, data=airnewTrainProp)
summary(exp1)

exp1pred <- predict(exp1, newdata = airnewTestProp)
dataair<- data.frame(actual= airnewTestProp$Stall, predicted= exp1pred)
dataair

par(mfrow=c(1,1))
plot(exp1pred, airnewTestProp$Stall)
abline(a=0,b=1)

##### Predicting stall speed model
air_lm1 <- lm(Stall ~ SHP + GW + WS + Length + Vmax , data = airnew)
summary(air_lm1)
par(mfrow=c(2,2))
plot(air_lm1)


air_lm2<- lm(Stall ~ SHP + WS + Length + Vmax-1 , data = airnewTrainProp)
summary(air_lm2)
plot(air_lm2)
airlm2pred <- predict(air_lm2, newdata = airnewTestProp)
dataair2<- data.frame(actual= airnewTestProp$Stall, predicted= airlm2pred)
dataair2

par(mfrow=c(1,1))
plot(airlm2pred, airnewTestProp$Stall)
abline(a=0,b=1)

##### Stall Speed transformation models
plot(airprop$Stall^2,airprop$Vmax)
plot(log(airprop$Stall), airprop$Vmax)
plot(airprop$Stall,airprop$Vmax)
```

```
air_lm3 <- lm(Stall~ SHP + WS + Vmax + Length-1,data=airnewTrainProp)
summary(air_lm3)
par(mfrow=c(2,2))
plot(air_lm3)

par(mfrow=c(1,1))
airlm3pred<- predict(air_lm3, newdata = airnewTestProp)
dataair3 <- data.frame(actual= airnewTestProp$Stall^2, predicted= airlm3pred)
dataair3

par(mfrow=c(1,1))
plot(airlm3pred, airnewTestProp$Stall)
abline(a=0,b=1)

air_lm4 <- lm(log(Stall) ~ SHP + WS + Vmax + Length-1, data = airnewTrainProp)
summary(air_lm4)
par(mfrow=c(2,2))
plot(air_lm4)

par(mfrow=c(1,1))
airlm4pred<- predict(air_lm4, newdata = airnewTestProp)
dataair4 <- data.frame(actual= log(airnewTestProp$Stall), predicted= airlm4pred)
dataair4

par(mfrow=c(1,1))
plot(airlm4pred, log(airnewTestProp$Stall))
abline(a=0,b=1)

##### predicting prop horsepower model
air_lm5<-lm(SHP ~ Length + GW-1, data = airnewTrainProp)
summary(air_lm5)
par(mfrow=c(2,2))
plot(air_lm5)

airlm5pred<- predict(air_lm5, newdata = airnewTestProp)
dataair5 <- data.frame(actual= airnewTestProp$SHP, predicted= airlm5pred)
dataair5

par(mfrow=c(1,1))
plot(airlm5pred, airnewTestProp$SHP)
abline(a=0,b=1)


boxcox(air_lm5, lamda = seq(-2,2,1/10), plotit = TRUE, eps =  1/50, xlab = expression(lamda), ylab ="log
airnewTrainProp$boxcox <- (airnewTrainProp$SHP)^0.5
airnewTestProp$boxcox <- (airnewTestProp$SHP)^0.5

air_lm6<- lm(boxcox ~ Length + GW-1, data = airnewTrainProp)
summary(air_lm6)
par(mfrow=c(2,2))
plot(air_lm6)

airlm6pred<- predict(air_lm6, newdata = airnewTestProp)
```

```r
dataair6 <- data.frame(actual= airnewTestProp$boxcox, predicted= airlm6pred)
dataair6

par(mfrow=c(1,1))
plot(airlm6pred, airnewTestProp$boxcox)
abline(a=0,b=1)


##### Predicting wing span model
air_lm7<- lm(WS ~ GW + Length + Vmax-1, data = airnewTrain)
summary(air_lm7)
par(mfrow=c(2,2))
plot(air_lm7)


airlm7pred<- predict(air_lm7, newdata = airnewTest)
dataair7 <- data.frame(actual= airnewTest$WS, predicted= airlm7pred)
dataair7

par(mfrow=c(1,1))
plot(airlm7pred, airnewTest$WS)
abline(a=0,b=1)

##### Logistic regression to predict multi engine or not
str(airnewTestProp)
multi_probit <- glm(Multi.Engine~GW+Length,family=binomial(link="probit"), data=airnewTrainProp)
summary(multi_probit)

df1<-multi_probit$coefficients[2:3]
exp(df1)

chisq_probit<-multi_probit$null.deviance-multi_probit$deviance
p_probit<-length(multi_probit$coefficients)-1
dchisq(chisq_probit,df=p_probit)

ll.null <- multi_probit$null.deviance/-2
ll.proposed <- multi_probit$deviance/-2

# McFadden's Pseudo R^2
(ll.null - ll.proposed) / ll.null

# Testing Logistic Regression Model
set.seed(123)
predict(multi_probit, airnewTestProp)
predict(multi_probit, airnewTestProp, type = 'response')

pVals <- predict(multi_probit, airnewTestProp, type = 'response')
labelpVals <- ifelse(pVals> 0.5, '1', '0')
labelpVals
airnewTestProp$Multi.Engine

table(labelpVals == airnewTestProp$Multi.Engine)/length(airnewTestProp$Multi.Engine)
```