Executive summary

The question was raised if mpg of 1974 cars was influenced by transmission type. To answer the question a linear model was build based on mtcars data using a linear model that showed the relation between mpg on the one hand and weight, hp and transmission type on the other. It was shown that after compensating for weight and hp there was no statistical significant relation between mpg and transmission type.

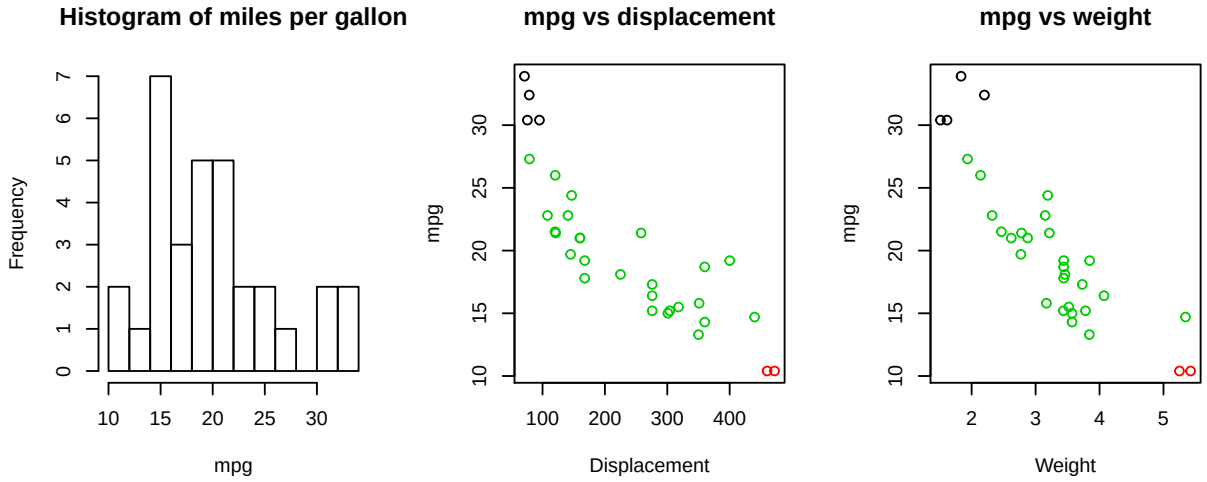
Data preparation and assumptions

Data is taken form mtcars. Mpg is the variable to be predicted. It was chosen to drop the 0.25 mile time (qsec) from the input variables becaus it is, like mpg, an outcome of the car's specs and not a design variable.

The variables cyl, vs, am, gear and carb are taken as factors. The other variables as continuous variable.

Exploratory analysis

When plotting a histogram of the mpg it can be seen that it looks a bit like a normal distribution with extremes at both sides of the spectrum. The lowest outliers have an mpg below 11 and the larger outliers have mpg above 30. When plotting the mpg against displacement and weight, it seems like the outliers follow a logical pattern so should not be discarded. Note, black are high mpg, green are middle mpg and red are low mpg)



Model selection

Model type selection

The goal is to quantify the relation between mgp and transmission type. To do so the decision was made to use a linear model with at max polynomials to the third degree. Because mpg is continuous, there is no need for a generalized linear model.

Model selection strategy

Model selection is an iterative process

1. Use lasso method (discussed in course 8 of the data science track) to find the next best value for a prediction

2. Use the new variable to fit the residual. For factors just us the factor itself -1, for continous variables test polinomials to the first second and third degree to fit and use anova (with a alpah value < 0.1) to select the best fit
3. Add the selected polinomail to the model fitting mpg and test if the new fit is better than the pervious one using anova
4. If the new fit is better then the previous one then update the residual to be fitted and remove the new fitting variable from the variables it can be fitted with

In the first iteration (see full calculation in the appendix) the lasso algorithm indicated the weight as the best fitting variable. The update_df_fit indicated the polynomial of second degree to outperform the polynomial of the first degree (Pr(>F) value of 0.0033 indicating a 0.3% chance of using the second degree polynomial while it is in fact overfitting the data).

In the second iteration the lasso algorithm indicated the hp as the second input variable. The first degree polynomial was indicated as the best as the second degree gave NA in anova and the third indicated a 64% chance of overfitting. Using anova again the fit based on weight, weight ^2 and hp was a better fit than weight and weight ^2 only with a Pr(>F) of 0.0021 (a 0.2 percent chance of using the algorithm while it is in fact overfitting the data).

In the third iteration the transmission type (a factor) was added (fitting mpg with weight, weight^2, hp and aim). Anova showed that there was a 84% chance (Pr(>F) = 0.84) that this algorithm was overfitting the data. When looking at the fit summary the transmission type was shown to have a Pr(|t|) value of 0.84 indicating agian a bad estimator to the data (see appendix for full results).

Conclusions

After compensating for weight and hp, manual or automatic transition has no statistica significant impact on fuel consumption.

Did the student do a residual plot and some diagnostics?

Appendix 1 Extended approach and results

In the first iteration it can be seen that the lasso algorithm defines weight as the variable to be used to fit.

```
next_var <- lasso_next_variable(df_fit)
```

```
## [[1]]
## wt
##  7
```

A second degree polynomial is shown to be the best fit as it has a Pr(>F) value in anova of only 0.00022 and the third degree polynomial has a Pr(>F) value of 0.47 (way larger than a boundary value of 0.1).

```
fitted_column <- 'wt'
best_fit <- find_fit_type(df_fit, df_fit$wt)
```

```
## second_degree  third_degree
##            NA   0.003388237
## [1] "A second degree polynomial is the best fit"
```

Next the y value in df_fit is updated to become the residual after the fit against weight.

```
df_fit <- update_df_fit(df_fit, fitted_column, best_fit)
```

Next hp is shown to be the next best value to fit.

```
next_var <- lasso_next_variable(df_fit)
```

```
## [[1]]
## hp
##  5
```

Using anova it is shown that a first degree polynomial is the best fit (the second and third degree have Pr(>F) values higher than 0.1).

```
best_fit <- find_fit_type(df_fit, df_fit$hp)
```

```
## second_degree  third_degree
##            NA     0.6432181
## [1] "A first degree polynomial is the best fit"
```

Using anova it is shown that there is only a 0.2% chance that we should reject the hypothesis that the fit with hp is better than the fit without hp so we continue with the fit with hp.

```
global_best_fit <- lm(mpg ~ wt + I(wt^2) + hp, data=df)
previous_best_fit <- lm(mpg ~ wt + I(wt^2), data=df)
anova(previous_best_fit, global_best_fit)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + I(wt^2)
## Model 2: mpg ~ wt + I(wt^2) + hp
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     29 203.75
## 2     28 144.29  1    59.452 11.537 0.002061 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The df_fit dataframe is updated accordingly.

```
fitted_column <- 'hp'
df_fit <- update_df_fit(df_fit, fitted_column, best_fit)
```

The next best value to fit is the transmission type.

```
next_var <- lasso_next_variable(df_fit)
```

```
## [[1]]
## am1
##   7
```

Because the transmission type is a factor the usual fit will be used.

```
best_fit <- find_fit_type(df_fit, df_fit$am)
```

```
## [1] "Use regular fit because the fitted value is a factor"
```

Using anova it can be seen that the chances of overfitting with automatic transmission added is 85%. way to high to assume that transmission type has significant influence on the mpg if corrected for weight of the car and hp of the engine.

```
previous_best_fit <- lm(mpg ~ wt + I(wt^2) + hp + I(hp^2), data=df)
global_best_fit <- lm(mpg ~ wt + I(wt^2) + hp + I(hp^2) + am, data=df)
anova(previous_best_fit, global_best_fit)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + I(wt^2) + hp + I(hp^2)
## Model 2: mpg ~ wt + I(wt^2) + hp + I(hp^2) + am
```

```
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     27 123.05
## 2     26 122.86  1   0.19045 0.0403 0.8424
```

If we would use the latest fit to interpret the coefficients (WHICH WE DO NOT) then we would get:

```
summary(lm(mpg ~ wt + I(wt^2) + hp + I(hp^2) + am, data=df))
```

```
##
## Call:
## lm(formula = mpg ~ wt + I(wt^2) + hp + I(hp^2) + am, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0533 -1.7553 -0.3802  1.2768  4.6422
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.028e+01  5.443e+00   9.236 1.08e-09 ***
## wt          -9.563e+00  2.872e+00  -3.329  0.00261 **
## I(wt^2)      8.815e-01  3.437e-01   2.564  0.01646 *
## hp          -9.460e-02  3.256e-02  -2.906  0.00739 **
## I(hp^2)      1.775e-04  8.374e-05   2.120  0.04372 *
## am1         -2.756e-01  1.373e+00  -0.201  0.84245
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.174 on 26 degrees of freedom
## Multiple R-squared:  0.8909, Adjusted R-squared:  0.8699
## F-statistic: 42.46 on 5 and 26 DF,  p-value: 1.07e-11
```

With the average value of weight (wt) and hp an automatic transmission car would burn 50 mpg. Going to manual transmission would save ~0.28 mpg. This number is however not statistically significant.

Supportive functions

Three functions are used in the analysis and described below:

- lasso_next_variable to find the next variabl using the lasso method
- find_fit_type to find the degree of best fitting polynomial
- update_df_fit update the fitting dataframe by setting y to the outcome residual and removing the column used in the previous fit

```
lasso_next_variable <- function(df_lasso){
    lasso_fit <- train(y ~ .-1, data=df_lasso, method='lasso')
    next_variable <- lasso_fit$finalModel$actions[1]
    print(next_variable)
}
```

```
find_fit_type <- function(df, x){
    if (class(x) == "factor"){
        print('Use regular fit because the fitted value is a factor')
        return(lm(y ~ x - 1, data=df_fit))
    }
    fit1 <- lm(y ~ x, data=df_fit)
    fit2 <- lm(y ~ x + I(x^2), data=df)
    fit3 <- lm(y ~ x + I(x^2) + I(x^3), data=df)
```

```
    anova_outcome <- anova(fit1, fit2, fit3)
    anova_outcome_short <- c(
        second_degree=anova_outcome$`Pr(>F)`[1],
        third_degree=anova_outcome$`Pr(>F)`[2]
        )
    print(anova_outcome_short)

    if (is.na(anova_outcome$`Pr(>F)`[2])){
        print('anova for second degree polynomial is NA so use first degree')
        return(fit1)
    }
    else if (anova_outcome$`Pr(>F)`[3] < 0.1){
        print('A third degree polynomial is the best fit')
        return(fit3)
    }
    else if (anova_outcome$`Pr(>F)`[2] < 0.1){
        print('A second degree polynomial is the best fit')
        return(fit2)
    }
    else {
        print('A first degree polynomial is the best fit')
        return(fit1)
    }
}
```

```
update_df_fit <- function(df_fit, fitted_column, fit) {
    df_fit$y <- df_fit$y - predict(fit)
    columns_to_keep <- colnames(df_fit)[which(fitted_column != colnames(df_fit))]
    df_fit <- df_fit[, columns_to_keep]
}
```

Appendix 2 Residual analysis

In the plots below it can be seen that the residual devaites a bit from the normal distribution as it tilts slightly to the left. It does however seem like there is littel relation between mpg and the residual indicating that most significant features are well approximated.