Practical 1 Solutions

Jumping Rivers

Course R package

Installing the course R package is straightforward. First install `drat`, a package that makes it easy to host and distribute packages.

```r
install.packages("drat")
```

Then

```r
drat::addRepo("rcourses")
install.packages("jrPredictive")
```

This R package contains copies of the practicals, solutions and data sets that we require. It will also automatically install any packages, that we use during the course. For example, we will need the `caret`, `mlbench`, `pROC` and `splines` to name a few. To load the course package, use

```r
library("jrPredictive")
```

During this practical we will mainly use thehe `caret` package, we should load that package as well

```r
library("caret")
```

The `cars2010` data set

The `cars2010` data set contains information about car models in 2010. The aim is to model the `FE` variable which is a fuel economy measure based on 13 predictors. Further information can be found in the help page, `help("cars2010", package = "AppliedPredictiveModeling")`.

The data is part of the `AppliedPredictiveModeling` package and can be loaded by

```r
data(FuelEconomy, package = "AppliedPredictiveModeling")
```

There are a lot of questions below marked out by bullet points. Don't worry if you can't finish them all, the intention is that there is material for different backgrounds and levels

Exploring the data

- Prior to any analysis we should get an idea of the relationships between variables in the data. Use the `pairs` function to explore the data.

- Create a simple linear model fit of `FE` against `EngDispl` using the `train` function. Hint: use the `train` function with the `lm` method.

```r
m1 = train(FE ~ EngDispl, method = "lm", data = cars2010)
```

- Examine the residuals of this fitted model, plotting residuals against fitted values

```r
rstd = rstandard(m1$finalModel)
plot(fitted.values(m1$finalModel), rstd)
```

fitted.values(m1$finalModel)
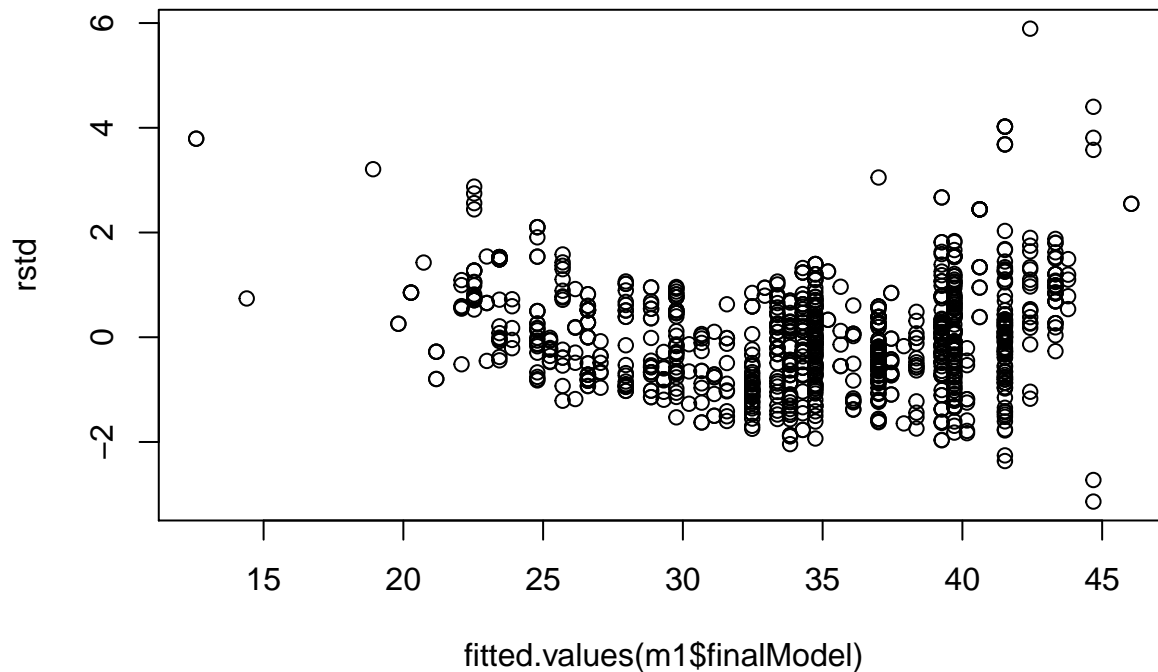
We can add the lines showing where we expect the residuals to fall to aid graphical inspection

```r
abline(h = c(-2, 0, 2), col = 2:3, lty = 2:1)
```

- What do the residuals tell us about the model fit using this plot?

```r
# There definitely appears to be some trend in the
# residuals.  The curved shape indicates that we
# potentially require some transformation of variables.
# A squared term might help.
```

- Plot the fitted values vs the observed values

```r
plot(cars2010$FE, fitted.values(m1$finalModel), xlab = "FE",
    ylab = "Fitted values", xlim = c(10, 75), ylim = c(10,
        75))
abline(0, 1, col = 3, lty = 2)
```
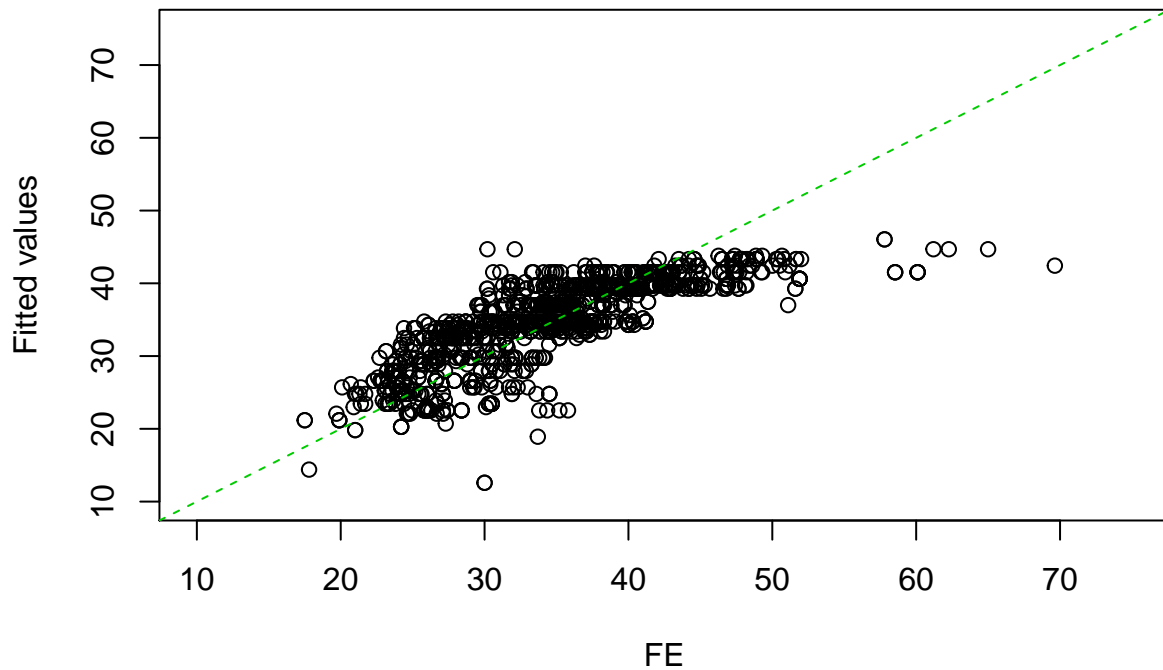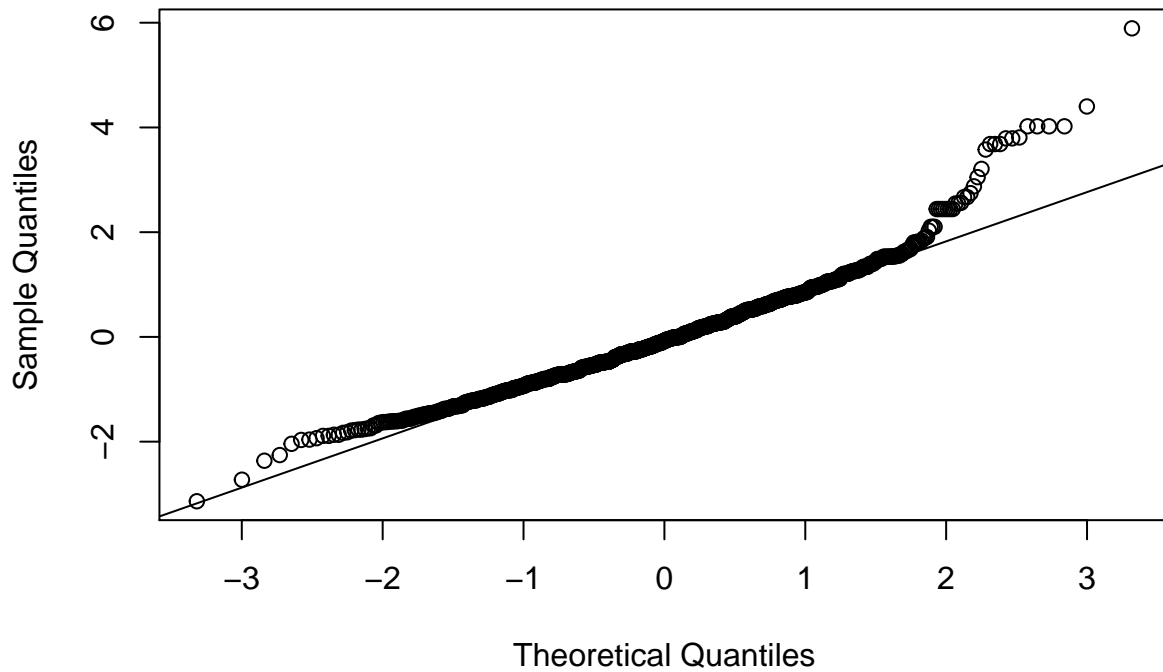
- What does this plot tell us about the predictive performance of this model across the range of the response?

```
# We seem to slightly over estimate more often than not
# in the 25-35 range. For the upper end of the range we
# seem to always under estimate the true values.
```
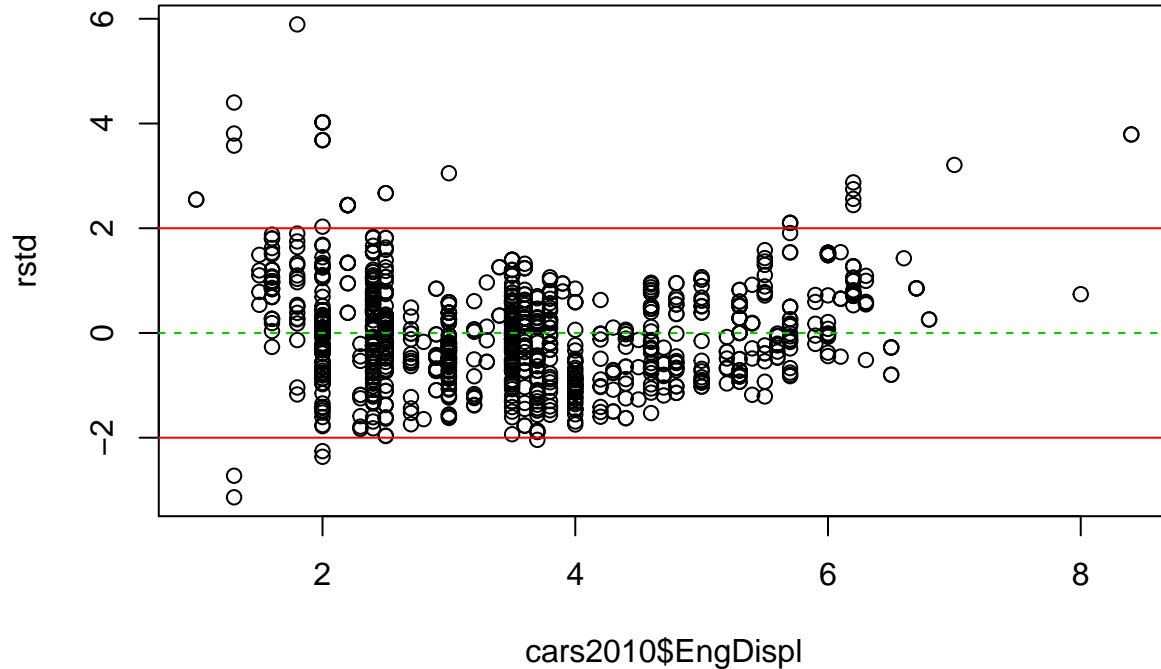
- Produce other diagnostic plots of this fitted model, e.g. a q-q plot

```
qqnorm(rstd)
qqline(rstd)
```

Normal Q–Q Plot


```
plot(cars2010$EngDispl, rstd)
abline(h = c(-2, 0, 2), col = 2:3, lty = 1:2)
```


- Are the modelling assumptions justified?

```
# We are struggling to justify the assumption of
# normality in the residuals here, all of the diagnostics
# indicate patterns remain in the residuals that are
# currently unexplained by the model.
```

Extending the model

- Do you think adding a quadratic term will improve the model fit?

```
# We are struggling to justify the assumption of
# normality in the residuals here, all of the diagnostics
# indicate patterns remain in the residuals that are
# currently unexplained by the model
```

- Fit a model with the linear and quadratic terms for `EngDispl` and call it `m2`

```
m2 = train(FE ~ poly(EngDispl, 2, raw = TRUE), data = cars2010,
    method = "lm")
```

- Assess the modelling assumptions for this new model.
- How do the two models compare?

```
# The residual diagnostics indicate a better fit now that
# the quadratic term has been included.
```

- How does transforming the response variable affect the fit? Common transformations may be a log or square root function.

```
# Perhaps the residuals more closely match the assumption
# of normality under this transformation. However we need
# to be careful about interpretation now as the response
# is on the log scale. Likewise for prediction we need to
# remember to undo the transformation.
```

- Add `NumCyl` as a predictor to the simple linear regression model `m1` and call it `m3`

```
m3 = train(FE ~ EngDispl + NumCyl, data = cars2010, method = "lm")
```

- Examine model fit and compare to the original.
- Does the model improve with the addition of an extra variable?

Visualising the model

The `jrPredictive` package contains a `plot3d` function to help with viewing these surfaces in 3D.

```
## points = TRUE to also show the points
plot3d(m3, cars2010$EngDispl, cars2010$NumCyl, cars2010$FE,
    points = FALSE)
```
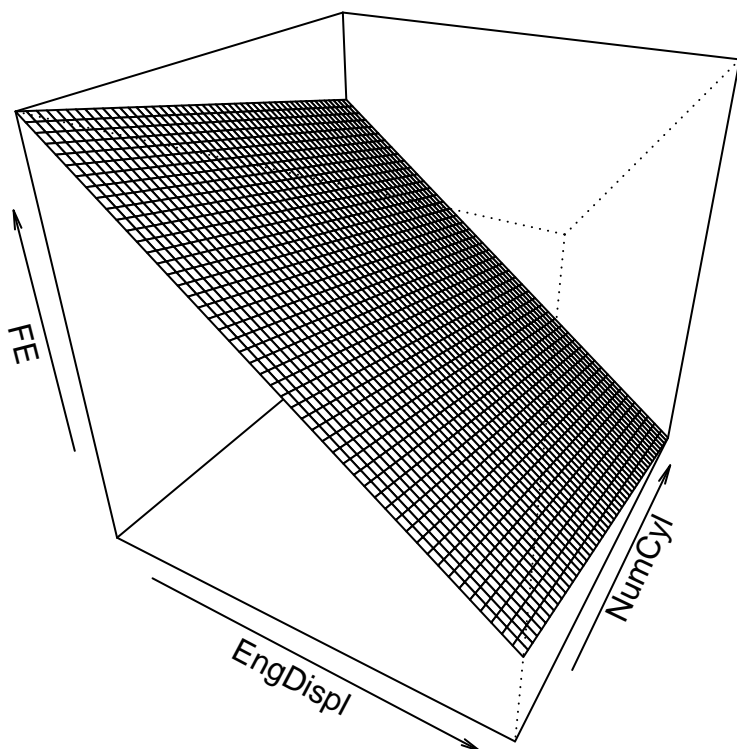
We can also examine just the data interactively, via

```
threejs::scatterplot3js(cars2010$EngDispl, cars2010$NumCyl,
    cars2010$FE, size = 0.5)
```

- Try fitting other variations of this model using these two predictors. For example, try adding polynomial and interaction terms

```
m4 = train(FE ~ EngDispl * NumCyl + I(NumCyl^5), data = cars2010,
    method = "lm")
```

How is prediction affected in each case? Don't forget to examine residuals, R squared values and the predictive surface.

- If you want to add an interaction term you can do so with the : operator, how does the interaction affect the surface?

Other data sets

A couple of other data sets that can be used to try fitting linear regression models.

Data set	Package	Response
diamonds	'ggplot2'	price
Wage	'ISLR'	wage
BostonHousing	'mlbench'	medv