First steps in R: the 10 commands to survive based on the Cars93 dataset

Patrick Jermann

October 18, 2016

1 Loading data

First you want to load the data into R. The `read.table` command will load text files into a data frame that is similar to a excel workbook. The parameters are `header=T` to indicate that the first line of the file contains the variable names, and `sep="\t"` to indicate that the columns are separated by tabulators. If your file is separated by commas, simply use `sep=","`. The `<-` operator assigns the result of reading the text to a variable named `d`.

```
> d <- read.table("Cars93.txt", header=T, sep="\t")
```

Now we can look at the column names of the dataframe that we created. The data frame contains quantitative variables that were measured on different cars (See Lock, R. H. (1993) 1993 New Car Data. Journal of Statistics Education 1(1)).

```
> names(d)
```

```
 [1] "Manufacturer"       "Model"              "Type"
 [4] "Min.Price"          "Price"              "Max.Price"
 [7] "MPG.city"           "MPG.highway"        "AirBags"
[10] "DriveTrain"         "Cylinders"          "EngineSize"
[13] "Horsepower"         "RPM"                "Rev.per.mile"
[16] "Man.trans.avail"    "Fuel.tank.capacity" "Passengers"
[19] "Length"             "Wheelbase"          "Width"
[22] "Turn.circle"        "Rear.seat.room"     "Luggage.room"
[25] "Weight"             "Origin"             "Make"
```

Note that in R, variable names can contain a dot or underscores.

2 Descriptive statistics

We can get simple descriptive statistics about the dataframe by using the summary method.

```
> summary(d)
```

If we were interested only in one variable, we would write the same command but separate the name of the data frame and the name of the variable with a $ sign. The summary for quantitative variables (e.g. Price) includes, the minimum, the maximum, mean, median as well as the first and third quartiles (same as 25th and 75th percentiles). The median is the value that splits the sample into two equal parts. This means that half the observations are smaller than the median and half the observations are larger than the median. Following the same lines, 1/4 of the observations are smaller than the first quartile and 3/4 of the observations are smaller than the third quartile.

```
> summary(d$Price)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   7.40   12.20   17.70   19.51   23.30   61.90
```

For nominal variables (e.g. Type), the summary command displays a table with the number of cases in the dataset for each distinct value of the factor.

```
> summary(d$Type)

Compact   Large Midsize   Small  Sporty     Van
     16      11      22      21      14       9
```

2.1 Standard deviation

The standard deviation indicates how much the obsevations vary around the mean. If in our sample, all observations had the same value, the standard deviation would be equal to zero. The variance is the square of the standard deviation. To compute the standard deviation in R, we use the `sd` command.

```
> sd(d$Price) # Computes the standard deviation

[1] 9.65943
```

If the variable is normally distributed, 68% of the observations lie within one standard deviation of the mean and 95% of the observations lie within two standard deviations of the mean. In our cars example, this means that 95% of the cars will cost between 0.191 ($19.5 - 2 * 9.66$) and 38.8 ($19.5 + 2 * 9.66$).

2.2 Confidence interval

We have 93 cars in our sample. These 93 cars do by no means correspond to all possible cars in the universe (the population). Statistics allow us to make inferences about this hypothetical population based on the sample that we have. The most basic of these statistics is the confidence interval of the mean, which corresponds to the range in which the mean of the population lies with a probability of 95% (or 99% if the confidence level is 99%). The confidence interval is computed from the standard deviation.

$ci = \frac{sd}{\sqrt{N}} * 1.96$

The confidence interval is based on the standard error, which is the standard deviation divided by the square root of the number of observations. This means that the larger the sample, the smaller the confidence interval, the more

confident we are about the true value of the mean. The standard error is then multiplied by 1.96 (which is the value of the normal distribution that corresponds to a 95% confidence interval). Note that typing the name of the variable (the second line below) prints its value.

```
> conf.int <- (sd(d$Price) / sqrt(length(d$Price))) * 1.96
> conf.int

[1] 1.963207
```

2.3 Graphs

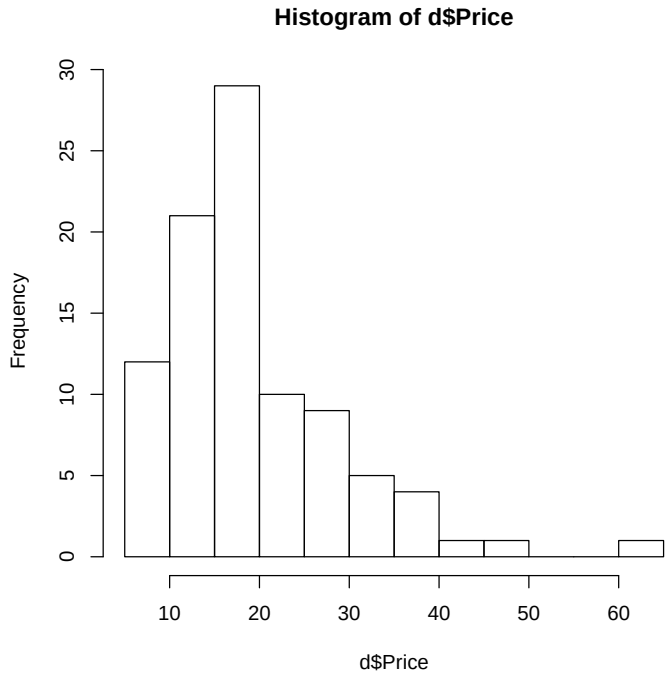
Finally, let's plot the histogram and save the plot into a file.

```
> png("price.png", width=600, height=600) # Opens the file
> hist(d$Price, breaks=50) # Produces the plot
> dev.off() # Closes the file

null device
          1
```

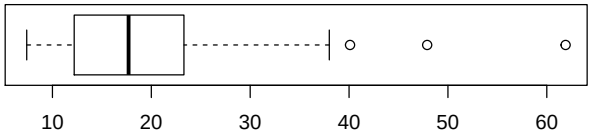
If you only type the `hist` command at the command line, the plot will appear on the screen rather than being saved to a file.

```
> hist(d$Price)
```



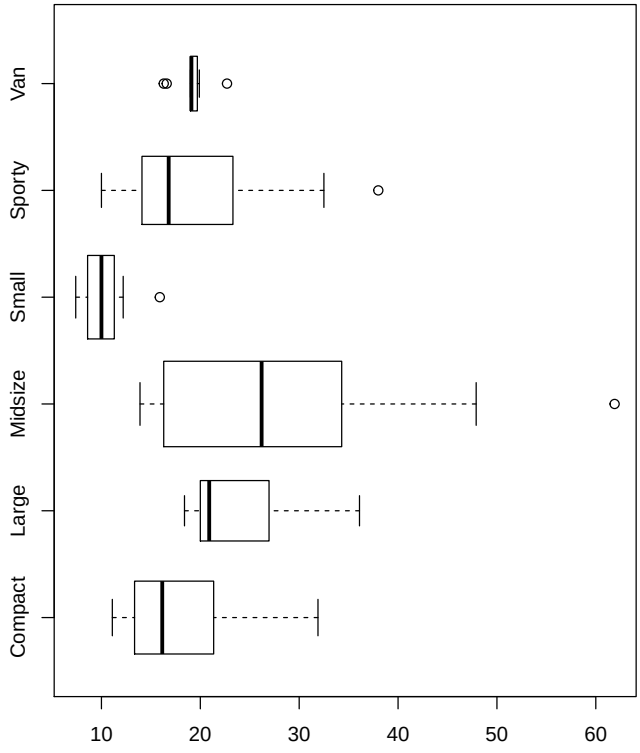
The boxplot is another useful descriptive plot that summarizes the distribution of a variable. The boxplot shows the median (the dark line in the box), as well as the first and thrird quartiles (the boundaries of the box).

```
> boxplot(d$Price, horizontal=TRUE, varwidth=TRUE)
```



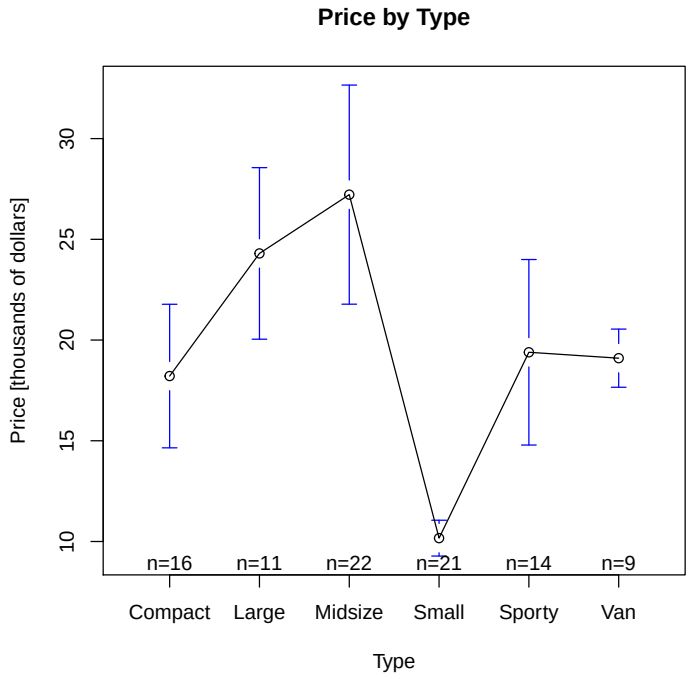
Boxplots also can be used to compare different categories. For example, we wonder whether the different car types in the sample have the same price range. The tilde (~) sign is used in R formulas as a sign for "depends" or "given". For example, the fomula below means "Price given Type".

```
> boxplot(d$Price ~ d$Type, horizontal=TRUE, varwidth=TRUE)
```



Let's also plot the means and confidence intervals with `plotmeans`. First we load the plotting library and then plot. Note that we added some nice titles and labels to the plot by using parameters.

```
> library(gplots)
> plotmeans(d$Price ~ d$Type,
+           main="Price by Type",
+           xlab="Type", ylab="Price [thousands of dollars]")
```



3 One factor analysis of variance

Analysis of variance (ANOVA) tests the null hypothesis that the mean of a variable is the same in several subsets of the sample that are defined by a categorical variable. In our example, we want to know whether the Price of cars is the same across different types of cars. We have a quantitative variable (Price) and a categorical variable (Type).

$H_0 : \mu_1 = \mu_2 = ... = \mu_n$

$H_1 : \mu_1 ! = \mu_2 ! = ... ! = \mu_n$

The command `oneway.test` does the ANOVA calculation for us. If we can reject the null hypothesis, then we would conclude that the mean Price is different across the Types of cars. Note that we tell it that the variances in the groups are equal with the `var.equal=T` parameter, but this has to be tested beforehand (see section below).

The difference is statistically significant !

```
> oneway.test(d$Price ~ d$Type, var.equal=T)

        One-way analysis of means
```

```
data:  d$Price and d$Type
F = 11.532, num df = 5, denom df = 87, p-value = 1.477e-08
```

The test output informs about:

- F (the ratio of variance between groups over the variance within the groups). F-values below 1 are a strong indicator that there is no difference between the groups.
- numerator degrees of freedom: the number of categories that are compared (df = 5 => 6 categories -1)
- denominator degrees of freedom: the number of observations (df = 87 => 93 cars - 6 categories)
- the p-value, the probability to be wrong by affirming that we reject H_0. The limit for p-values in social science is .05. This means that if the p-value is smaller than .05, we reject H_0 and accept H_1 (and hence conclude that the means are different in the groups). If the p-value is larger than .05, we cannot reject H_0 (and hence we conclude that the means are the same in the groups).

3.1 Computing a new factor from categories

The price of a car obviously depends on the Manufacturer. Asian cars for instance are known to be cheaper than other cars.

Let's compute a new variable called `Asian` that groups the Asian car manufacturers into a new category of "Asian" cars and the others as "non-Asian". This operation combines three commands.

First, we do a if/else test. The first argument is the test (the vertical bar means "or" and == tests equality). The second argument is the value returned by the test if it is true ("Asian") and the second argument is the value returned by the test if it is false ("non-Asian").

Second, we transform the output of the test into a categorical variable with the `factor` command.

Finally we assign the new values to a new column of the data frame called "Asian".

```
> d$Asian <- factor(ifelse(d$Manufacturer == "Honda" |
+                          d$Manufacturer == "Hyundai" |
+                          d$Manufacturer == "Mazda" |
+                          d$Manufacturer == "Mitsubishi" |
+                          d$Manufacturer == "Nissan" |
+                          d$Manufacturer == "Suzuki" |                    d$Manufact
+                          , "Asian", "non-Asian")
+                   )
> tapply(d$Price, d$Asian, mean)

    Asian non-Asian
 15.71739  20.75571

> tapply(d$Price, d$Asian, sd)
```

```
    Asian non-Asian
 6.295789 10.266417

>
```

Let's see if the price of the cars depends on their asian origin ! And yes it does, as the F-value is quite large and the p-value is smaller than .05. We would report the difference in a report as follows: A one-way analysis of variance with the asian origin of the car as independent variable and the price as dependent variable was conducted. Asian cars (M=15.7, sd=6.3) are less expensive than non-asian cars (M=20.8,sd=10.3) (F[1,91]=4.9, p=.03).

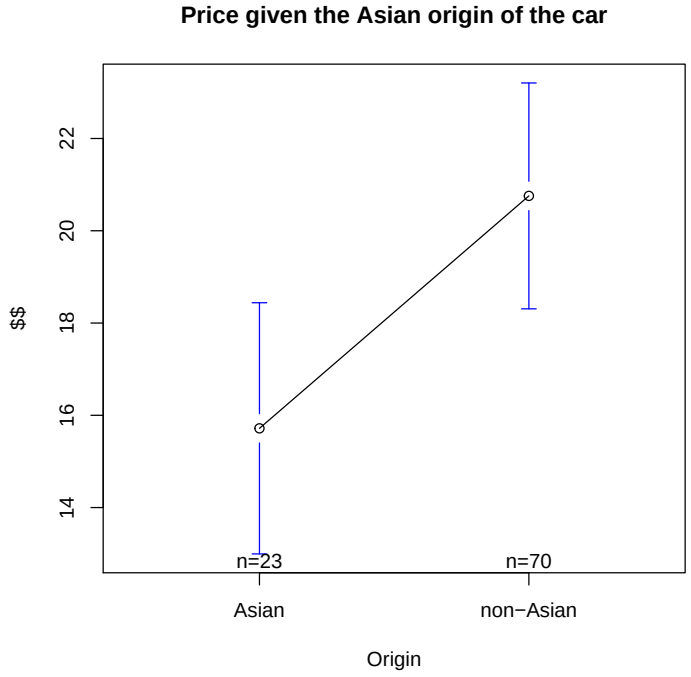
```
> oneway.test(d$Price ~ d$Asian, var.equal=T)

        One-way analysis of means

data:  d$Price and d$Asian
F = 4.9101, num df = 1, denom df = 91, p-value = 0.02919
```

And we accompany the analysis with a figure that shows the mean and the confidence intervals

```
> plotmeans(d$Price ~ d$Asian,
+           main="Price given the Asian origin of the car",
+           xlab="Origin", ylab="$$")
```



3.2 Computing a new factor based on a median cut

Imagine we want to split the cars depending on their gas consumption (a combination of `MPG.city` and `MPG.highway`). The operations are very similar to the previous case, except that this time the test involves the median.

```
> d$Guzzler <- factor(ifelse(d$MPG.city > median(d$MPG.city),"GUZZLER","ECONOMY"))
```

We could now look at the number of asian and non-asian cars by consumption category and do a chi-square test to see if these are equally distributed. For a valid and significant test, expected values should be larger than 5 and some residuals larger than 1.96. We see from this analysis that there is a tendency for asian cars to be guzzling more than non-asian cars. The residuals indicate that the guzzlers are over-represented in the Asian category (which is by the way rather surprising).

```
> table(d$Asian, d$Guzzler)

           ECONOMY GUZZLER
  Asian          7      16
  non-Asian     42      28

> c <- chisq.test(table(d$Asian, d$Guzzler))
> c

        Pearson's Chi-squared test with Yates' continuity correction

data:  table(d$Asian, d$Guzzler)
X-squared = 4.9424, df = 1, p-value = 0.02621

> c$expected # These should be larger than 5

             ECONOMY  GUZZLER
  Asian     12.11828 10.88172
  non-Asian 36.88172 33.11828

> c$residuals # These should be larger than 1.96

               ECONOMY    GUZZLER
  Asian     -1.4702918  1.5515837
  non-Asian  0.8427881 -0.8893857
```

4 More than one factor analysis of variance

Let's now do a two factor analysis with the `Price` as a dependent variable and the `Asian` and `Guzzler` variables as categories. We use the command `aov` to compute the analysis and separate the factors with the `*` sign to indicate that we are interested in simple effects for each of the factors as well as in the interaction effect. The `f1 * f2` operator is a short hand for `f1 + f2 + f1:f2` where the `:` sign means "interaction". The `summary` command shows the details of the analyis.

```
> summary(aov(d$Price ~ d$Asian * d$Guzzler))

                  Df Sum Sq Mean Sq F value   Pr(>F)
d$Asian            1    439   439.5   7.894   0.0061 **
d$Guzzler          1   3140  3139.7  56.396 4.35e-11 ***
d$Asian:d$Guzzler  1     50    50.0   0.898   0.3459
Residuals         89   4955    55.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We would report the results as follows: A two way analysis of variance was conducted with the Price as dependent variable and Asian origin and Gas consumption as independent variables. There were simple effects of the Asian origin ($F[1,90]=7.9$, $p<.01$) as well as of the Gaz consumption factor ($F[1,90]=56.4$, $p<.001$). There was not interaction effect. We see from table 1 and figure 3 that Non-Asian cars are more expensive that Asian cars and that Economy cars are more expensive than Guzzler cars.

To present the results nicely in LaTeX, we can also present the details of the analysis as a table as well as the means and standard deviations in a separate table by using the `xtable` command.

```
> library(xtable)
> xtable(summary(aov(d$Price ~ d$Asian * d$Guzzler)))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
d$Asian	1	439.46	439.46	7.89	0.0061
d$Guzzler	1	3139.73	3139.73	56.40	0.0000
d$Asian:d$Guzzler	1	49.98	49.98	0.90	0.3459
Residuals	89	4954.85	55.67		

```
> xtable(tapply( d$Price, list(d$Asian, d$Guzzler),
+                function(x) {
+                  paste(round(mean(x),2)," (",round(sd(x),2),")", sep="")
+                }),
+        label="tab:asian-guzzler",
+        caption="Car prices by asian origin and gas consumption.
+                 Numbers represent means and standard deviation in parentheses.")
```

	ECONOMY	GUZZLER
Asian	22.13 (5.92)	12.91 (4.07)
non-Asian	25.9 (9.84)	13.04 (4.42)

Table 1: Car prices by asian origin and gas consumption. Numbers represent means and standard deviation in parentheses.

There are several ways to plot the results. By using `plotmeans` and the `interaction.plot` method. If we plot the price by asian origin (see figure 1) and by whether the car is a gaz guzzler (see figure 2) we see that the difference

```
> plotmeans(d$Price ~ d$Asian, ylim=c(10,30))
```

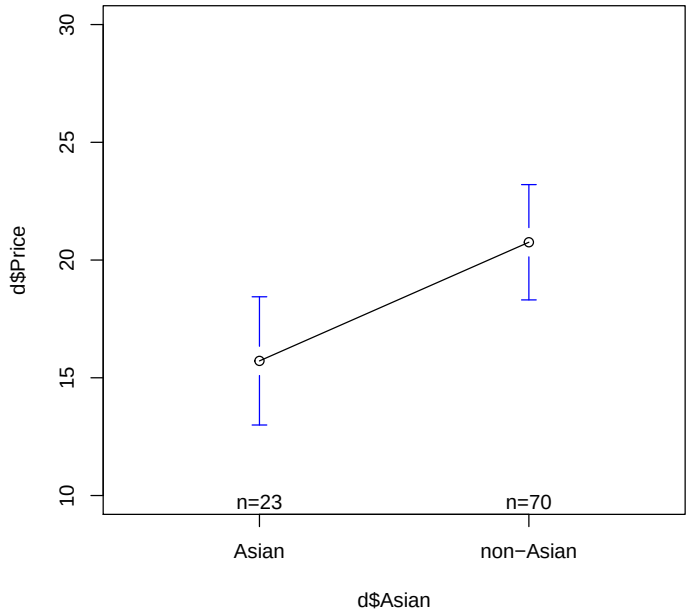


Figure 1: Price by Asian origin.

```
> plotmeans(d$Price ~ d$Guzzler, ylim=c(10,30))
```

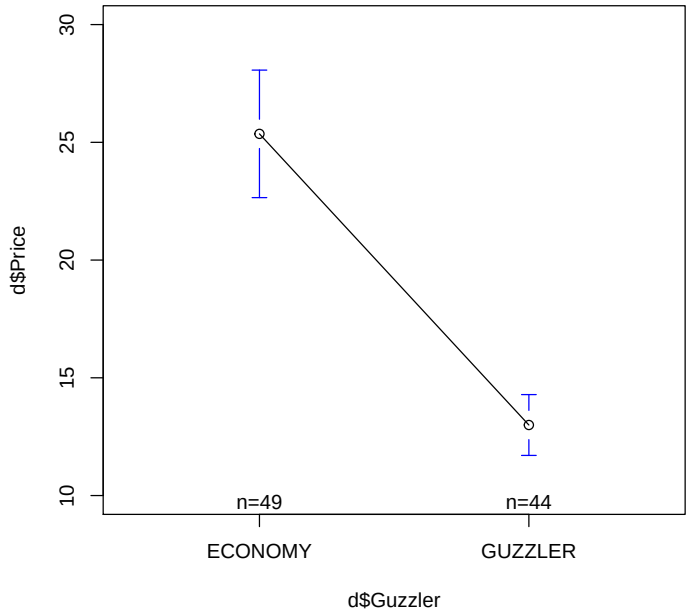


Figure 2: Price by Guzzler category.

```
> plotmeans(d$Price ~ interaction(d$Asian, d$Guzzler), connect=list(c(1,2),c(3,4)))
```

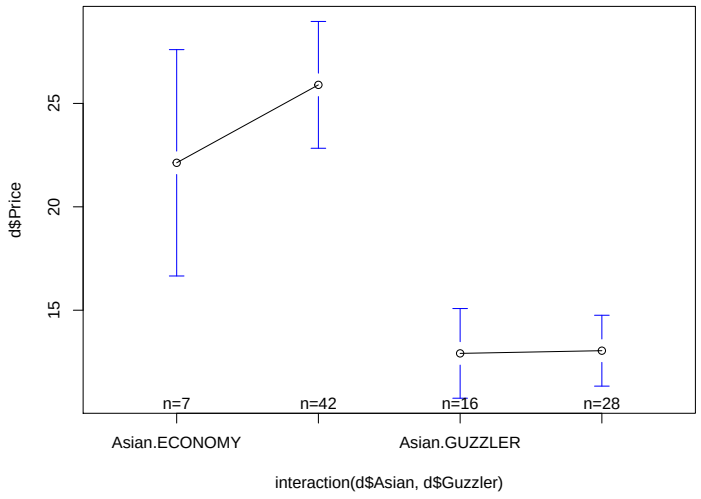


Figure 3: Price by Asian origin and Gas consumption.

```
> interaction.plot(d$Asian, d$Guzzler, d$Price, ylim=c(10,30))
```

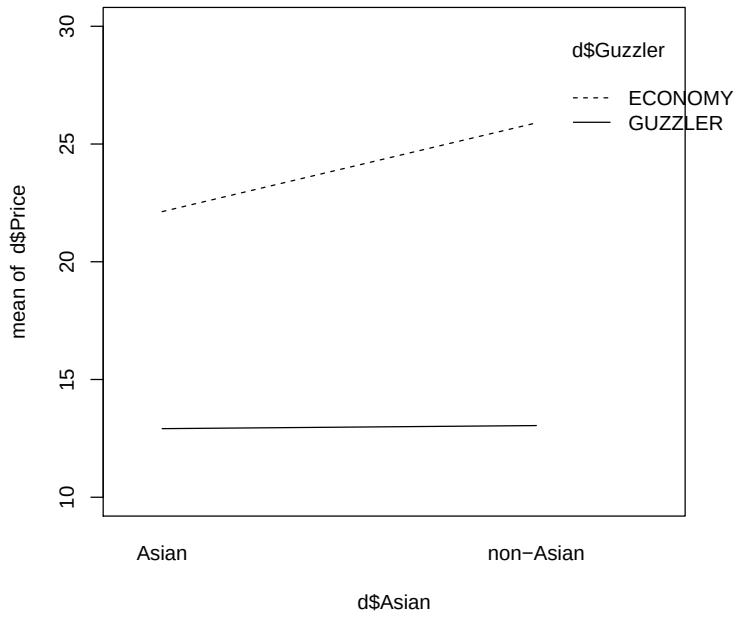


Figure 4: Price by Asian origin and Gas consumption.

between asian and non-asian cars is smaller than between economy and gas guzzlers.

It is also possible to plot both categories in the same plot.

The second way uses `interaction.plot` and does not include confidence intervals but crosses the categories instead.

5 Linear Models

Linear models, often called linear regressions, allow to predict the variations of a quantitative dependent variable with a linear combination of other quantitative variables. For example, we could try to explain the price of a car (the dependent variable) by a combination of the power of the car (Ferraris with 300 HP are ususally more expensive than Fiat 500 with 50 HP) and the size of the car (e.g. via the wheelbase which is the distance between the rotational centers of the wheels).

5.0.1 Fitting linear models

In R, you can build linear models with the `lm` command (lm stands for linear model). A linear model estimates the β parameters in the equation $y = \beta_0 + \beta_1 * x + \epsilon$. This equation defines a line where β_0 is the intercept and β_1 is the

slope of the line. Because a linear combination of predictors only aproximates the observed data, there is an residual error ϵ which remains unexplained. These residuals ϵ are required to follow a normal distribution with mean 0.

The simplest regression model is be the null model that predicts the outcome with a constant ($y = \beta_0$). As a convention in R, the term $\sim$1 refers to the intercept. Hence the simplest model predicts that the Price best predicted by it's average value: 19.51. This corresponds to the horizontal line in figure 5.

```
> mean(d$Price)

[1] 19.50968

> m0 <- lm(Price ~ 1, data=d)
> m0

Call:
lm(formula = Price ~ 1, data = d)

Coefficients:
(Intercept)
      19.51
```

The next step consists of adding predictors to the simple m0 model, for example, a model (m1) where the `Horsepower` variable predicts the `Price` variable. The m1 model is of the form $y = ax + b$.

Inspection of the model (simply typing m1) shows that the Price is equal to a constant (the intercept) -1.4 plus 0.15 points for each additional Horsepower. Oddly enough, a car with zero horsepower would have a negative cost !

```
> m1 <- lm(Price ~ Horsepower, data=d)
> m1

Call:
lm(formula = Price ~ Horsepower, data = d)

Coefficients:
(Intercept)   Horsepower
    -1.3988       0.1454
```

We can plot the data and the result in a simple scatterplot as follows (see figure 5). The xlim and ylim parameters set the range of the axes that have to be drawn. The main parameter refers to the title of the graph and the xlab and ylab parameters set the labels for the axes.

```
> plot(d$Price ~ d$Horsepower,
+      main="Price given Horsepower",
+      xlab="Horsepower", ylab="Price")
> abline(lm(d$Price ~ 1), lty=2, lwd=2)
> abline(lm(d$Price ~ d$Horsepower))
> legend("topleft", legend=c("m0","m1"), lty=c(2,1))
```

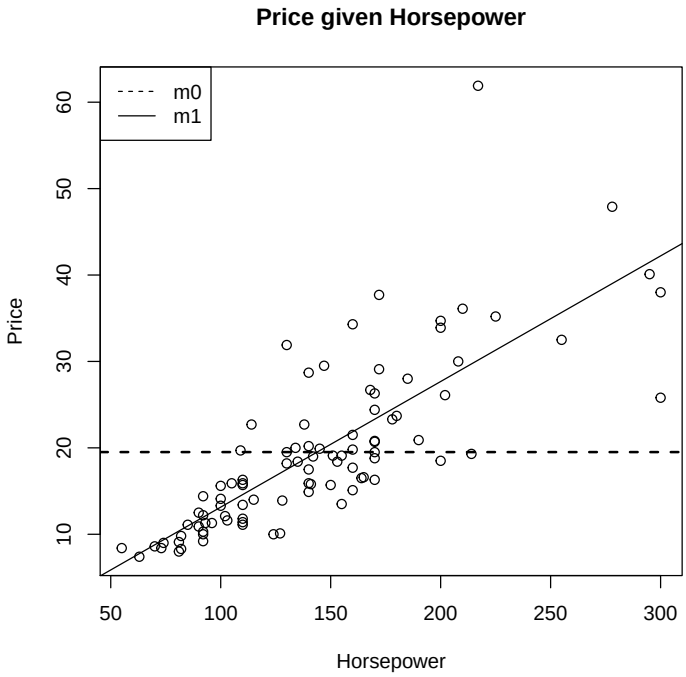
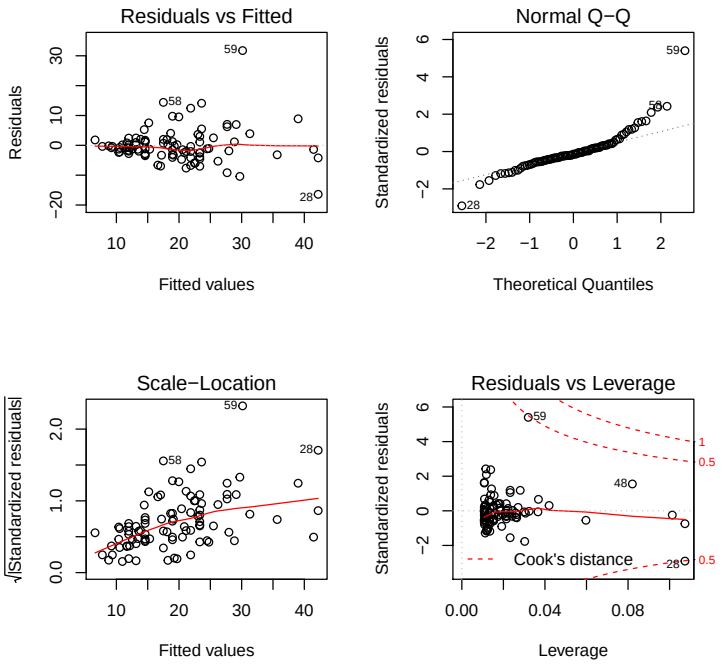


Figure 5: Scatterplot of price give horsepower.

5.1 Diagnosing linear models

The quality of a fit can be assessed by plotting diagnosis plots. These plots allow to check whether the residuals are more or less equally variable for the range of predicted values (residuals versus fitted) and that the residuals are distributed normally (Normal Q-Q plot of the residuals). The Scale-Location and Residuals vs Leverage plots indicate whether some observations affect the estimates of the model too strongly.

```
> par(mfrow=c(2,2))
> plot(m1)
```



In the first plot, the red line represents a "local summary" of the residuals given the predictions of the model. We see that the red line stays more or less horizontal at zero. This is a good sign since it indicates that the errors of the models' prediction are as much positive and negative and that they do not increase with the predicted values.

The second plot shows whether the errors are normally distributed. If the errors were normally distributed they should follow the diagonal dotted line. We see that observation 28 and 59 are outliers, which means that the error is particularly large for these two cars. Car 28 is a Dodge Stealth. For this car, the residual is especially large and negative, which means that the model predicts a much too high price (since $residual = observed - predicted$). On the contrary, for car 59, a Mercedes-Benz 300E, the model predicts a price which is too low.

```
> # Residual for car 28
> d$Price[28] - fitted(m1)[28]
```

```
      28
-16.4126

> # Residual for car 59
> d$Price[59] - fitted(m1)[59]

      59
31.75321
```

It appears clearly from the third plot (Scale-Location plot) that observations 28 and 59 appear as outliers in the graphs. Their standardized residuals are much larger than those of most of the cars. Also, the red line shows a tendency for the square root of standardized residuals is increasing with the fitted values.

Lastly, the leverage plot shows that observations 28 and 59 exert a large influence on the parameters estimated by the model. Cook's distance is an indication of how much the prediction of the model changes for a point once this point is removed from the model. Points which are too influencial merit closer examination. Points with large values of Cook's distance (larger than 1) may be deleted from the sample to compute a better estimate.

5.2 Evaluating linear models

There are two ways to evaluate linear models. First we may be interested to know which of the β_i parameters in the equation $y = \beta_0 + \beta_1 * x$ are different from zero. The summary function uses t-tests to test whether each parameter of the function is significantly different from zero. We see in the tests below that the intercept does not statistically differ from zero (it's p-value is larger than .05) and that the `Horsepower` variable significantly contributes to influence the contributions.accepted variable (p-value is smaller than .05).

```
> summary(m1)

Call:
lm(formula = Price ~ Horsepower, data = d)

Residuals:
    Min      1Q  Median      3Q     Max
-16.413  -2.792  -0.821   1.803  31.753

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.3988     1.8200  -0.769    0.444
Horsepower    0.1454     0.0119  12.218   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.977 on 91 degrees of freedom
Multiple R-squared:  0.6213,        Adjusted R-squared:  0.6171
F-statistic: 149.3 on 1 and 91 DF,  p-value: < 2.2e-16
```

The second way to evaluate our models is to compare them with each other. The question that we ask in this case is to know whether adding predictors

significantly reduces the residual error. The `anova` method allows to compare several models. The result of the comparison shows that the model m1 better describes the data than the m0 model (the RSS term indicates residuals sum of squares and should be as small as possible).

```
> anova(m0,m1)

Analysis of Variance Table

Model 1: Price ~ 1
Model 2: Price ~ Horsepower
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     92 8584.0
2     91 3250.9  1    5333.1 149.29 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

5.3 Model selection

The more variables we add to a model, the smaller the residual error. But this does not satisfy the requirement of parcimony in building explanatory models. Also, the inclusion of variables that are highly correlated among each other (e.g. the number of cylinders and the horsepower) introduces biases in the parameter estimates.

Model selection is an iterative process where we either start with a complex model and remove variables (backwards elimination) or we start with a simple mode and add variables (forward selection). The criteria for inclusion or removal of variables are first of all *theoretical* (you should add variables only if you have a clear hypothesis about it's effect), based on an significant improvement of the model (the residual sum of square RSS is smaller) or based on statistical criteria (e.g. AIC Akaike Information Criterion).

For example, adding the weight variable to the model for the price does not improve the prediction.

```
> m2 <- lm(Price ~ Horsepower + Weight, data=d)
> anova(m0,m1,m2)

Analysis of Variance Table

Model 1: Price ~ 1
Model 2: Price ~ Horsepower
Model 3: Price ~ Horsepower + Weight
  Res.Df    RSS Df Sum of Sq        F Pr(>F)
1     92 8584.0
2     91 3250.9  1    5333.1 151.3469 <2e-16 ***
3     90 3171.4  1      79.5   2.2554 0.1367
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Let's see whether the Wheelbase (empattement) is a better predictor. We suppose that there are as well small cars that have big motors and big bars that

have smaller motors. Indeed the correlation between Horsepower and Wheelbase is not too large. But even though the model m2 is significantly better than m1, we see that the gain in Resuduals is rather small and that the increase in R^2 is not spectacular either.

```
> cor.test(d$Horsepower, d$Wheelbase)

        Pearson's product-moment correlation

data:  d$Horsepower and d$Wheelbase
t = 5.317, df = 91, p-value = 7.483e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3143189 0.6282546
sample estimates:
      cor
0.4868542

> m2 <- lm(Price ~ Horsepower + Wheelbase, data=d)
> summary(m2)

Call:
lm(formula = Price ~ Horsepower + Wheelbase, data = d)

Residuals:
     Min       1Q   Median       3Q      Max
-12.7498  -3.1960  -0.2457   1.9887  31.4455

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -22.01636    9.89771  -2.224   0.0286 *
Horsepower    0.13159    0.01337   9.844 6.03e-16 ***
Wheelbase     0.21742    0.10266   2.118   0.0369 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.866 on 90 degrees of freedom
Multiple R-squared:  0.6393,        Adjusted R-squared:  0.6312
F-statistic: 79.75 on 2 and 90 DF,  p-value: < 2.2e-16

> anova(m0,m1,m2)

Analysis of Variance Table

Model 1: Price ~ 1
Model 2: Price ~ Horsepower
Model 3: Price ~ Horsepower + Wheelbase
  Res.Df    RSS Df Sum of Sq        F  Pr(>F)
1     92 8584.0
2     91 3250.9  1    5333.1 155.0051 < 2e-16 ***
3     90 3096.6  1     154.3   4.4852 0.03695 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

An automatic selection procedure is offered by the `step` function which takes the most complex model as a start and removes variables based on the AIC criteria. However, caution is required in using such automatic procedures, as they do not guarantee accurate predictions, nor models that make sense from the theoretical point of view.

We can choose whether the step method searches the space of potential models by following a backward (removing the variable which explains the least variance), forward (adding the variable which adds most explanatory power to the model) or both directions (starting from the full model and checking both whether the model gets better by removing or adding a variable). By default, R steps in both directions.

```
> step(lm(Price ~ Horsepower + Cylinders + EngineSize + RPM + Fuel.tank.capacity + Length
```

The model that is retained by the step procedure is:

```
> summary(lm(formula = Price ~ Horsepower + Wheelbase + Turn.circle, data = na.omit(d)))

Call:
lm(formula = Price ~ Horsepower + Wheelbase + Turn.circle, data = na.omit(d))

Residuals:
     Min       1Q   Median       3Q      Max
-14.3355  -2.6228  -0.1341   2.3187  26.9886

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -24.44673   12.50910  -1.954  0.05425 .
Horsepower    0.13857    0.01733   7.994 9.67e-12 ***
Wheelbase     0.54653    0.16463   3.320  0.00137 **
Turn.circle  -0.83325    0.31428  -2.651  0.00971 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.833 on 78 degrees of freedom
Multiple R-squared:  0.6697,        Adjusted R-squared:  0.657
F-statistic: 52.72 on 3 and 78 DF,  p-value: < 2.2e-16
```

It is possible to force the direction of the stepping by using the *direction* parameter. When you formulate a “forward” direction it is necessary to provide the starting model with *lower* and the most complex model with *upper*.

```
> step(lm(Price ~ 1, data=na.omit(d)),
+      scope=list(
+        lower=Price ~ 1,
+        upper=Price ~ MPG.city + Cylinders + EngineSize + Horsepower + RPM + Length + Wid
+        ),
+      direction="forward",
+      )
```

From running the above command we notice that the resulting model is slightly different from stepping in both directions, since it replaced Wheelbase and Turn.Circle with Weight and Width.

```
> summary(lm(Price ~ Horsepower + Weight + Width, data=na.omit(d)))

Call:
lm(formula = Price ~ Horsepower + Weight + Width, data = na.omit(d))

Residuals:
     Min       1Q   Median       3Q      Max
-15.0106  -2.6641  -0.6295   2.2425  28.6247

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 56.379692  18.340663   3.074 0.002910 **
Horsepower   0.093544   0.024829   3.767 0.000318 ***
Weight       0.013084   0.003283   3.985 0.000150 ***
Width       -1.297441   0.354446  -3.660 0.000456 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.673 on 78 degrees of freedom
Multiple R-squared:  0.6876,        Adjusted R-squared:  0.6756
F-statistic: 57.22 on 3 and 78 DF,  p-value: < 2.2e-16
```

6 Categorical variables

6.1 Chi-square test of independance

The relationship between two categorical variables can be assessed by a cross-tabulation and a chi-square test.

For example, we can answer the question whether the asian origin of a car is related to the fact that the car belongs to the economic cars. In chi-square terms, the question is whether the *Guzzler* variable (either ECONOMY or GUZZLER) is independent of the *Asian* (Asian versus non-Asian) variable.

The cross-tabulation is either obtained through the table command or through the xtabs command as follows:

```
> # xtabs( ~ Asian + Guzzler, data=d)
> tt <- table(d$Asian,d$Guzzler)
> tt

           ECONOMY GUZZLER
  Asian          7      16
  non-Asian     42      28
```

Under the hypothesis that these variables are independent, we would expect the following counts. We see that in our sample there are less (7) Asian Economy cars than expected (12.11) but theat there are more Asian Guzzlers (16) than expected (10.88). The expected value for a cell is obtained by multiplying the column total and the row total of the cell and dividing the result by the grand total of the table.

```
> chi <- chisq.test(tt)
> chi$expected
```

```
            ECONOMY  GUZZLER
Asian     12.11828 10.88172
non-Asian 36.88172 33.11828
```

The decision variable that is used to measure the deviance from independence is given by the following formula:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where O are the observed tallies and E are the expected values.

The chi-square tests whether the deviation between the observed and the expected distribution is larger than a theoretical χ^2 value for a given α value. In our case we see that the χ^2 deviation (4.94) is large enough to yield to a p-value of 0.02. We can therefore reject the null hypothesis that the variables are independent (and therefore conclude that they are dependent). The residuals show which cell in the table contribute most to the difference. On our case, the largest deviation from independence comes from the Asian Guzzlers (residual is 1.55).

```
> chi

        Pearson's Chi-squared test with Yates' continuity correction

data:  tt
X-squared = 4.9424, df = 1, p-value = 0.02621

> chi$residuals

              ECONOMY    GUZZLER
  Asian     -1.4702918  1.5515837
  non-Asian  0.8427881 -0.8893857
```

The conditions for validity of a chi-square test are that the expected counts are larger than 5. Also, the observations (the tallies) should be independent. It is questionable for instance to do chi-square anaylses if the same person was measured on repeated occasions (e.g. to compare the types of dialogue interventions across levels of expertise).

6.2 Linear Discriminant Analysis

This technique allows to predict which linear combination of quantitative variables best predicts a nominal variable. A useful documentation is available here: http://www.statmethods.net/advstats/discriminant.html. The grouping factor with n categories is known in advance (for example the type of the car) and we want to know which linear combination of quantitative variables best predicts the belonging of observations to a given category. The approach is quite similar to stepping a linear model, except that the outcomes is categorical.

Here we see a model that identifies which variables best predict the car type. The lda function from the MASS package allows to fit the linear discriminant model. The analysis results in $n - 1$ dimensions.

```
> library(MASS)
> fit <- lda(Type ~ Price  + Horsepower  + EngineSize +
+            RPM + Fuel.tank.capacity + Length +
+            Wheelbase + Turn.circle , data=d, na.action="na.omit")
> fit

Call:
lda(Type ~ Price + Horsepower + EngineSize + RPM + Fuel.tank.capacity +
    Length + Wheelbase + Turn.circle, data = d, na.action = "na.omit")

Prior probabilities of groups:
   Compact      Large    Midsize      Small     Sporty        Van
0.17204301 0.11827957 0.23655914 0.22580645 0.15053763 0.09677419

Group means:
           Price Horsepower EngineSize      RPM Fuel.tank.capacity   Length
Compact 18.21250   131.0000   2.331250 5362.500           16.06875 182.1250
Large   24.30000   179.4545   4.209091 4672.727           19.09091 204.8182
Midsize 27.21818   173.0909   3.059091 5336.364           18.45000 192.5455
Small   10.16667    91.0000   1.595238 5633.333           12.61905 167.1905
Sporty  19.39286   160.1429   2.492857 5392.857           15.95000 175.2143
Van     19.10000   149.4444   3.200000 4744.444           20.94444 185.6667
        Wheelbase Turn.circle
Compact 102.75000    38.25000
Large   113.27273    42.63636
Midsize 107.40909    40.18182
Small    96.57143    35.14286
Sporty   98.14286    38.85714
Van     112.44444    41.77778

Coefficients of linear discriminants:
                            LD1           LD2          LD3           LD4
Price                0.0095530829  0.0616598313  0.066809254  0.0896251099
Horsepower          -0.0045789017 -0.0034053059  0.021699231 -0.0210707828
EngineSize          -0.2216308017  0.7078967664 -1.944313856 -0.7816175935
RPM                 -0.0003475812  0.0006157647 -0.001357877  0.0009418157
Fuel.tank.capacity  0.2728224967 -0.3650896711  0.209439744  0.1946771720
Length              -0.0248211417  0.1285703509  0.019580153  0.0354913507
Wheelbase            0.2735770122 -0.1178441341 -0.203806062  0.0321248432
Turn.circle          0.1469508142 -0.0891722678  0.287268511 -0.0520805933
                            LD5
Price                0.054268965
Horsepower          -0.012536241
EngineSize           1.629440398
RPM                  0.001575098
Fuel.tank.capacity -0.042310383
Length              -0.083823078
Wheelbase            0.063049205
Turn.circle         -0.017640794
```

```
Proportion of trace:
   LD1    LD2    LD3    LD4    LD5
0.7401 0.1339 0.0911 0.0324 0.0025
```

From the "Proportion of trace" we see that the two first dimensions already account for 87% of the variation between classes (this is similar to the R^2 measure of variance explained in regression). The dimensions of the discriminant model can be interpreted based on the "Coefficients of linear discriminants". The larger a coefficient, the more the variable contributes to the discriminant function. Attention, these coefficients are not standardized and depend on the unit used in the variables.

It is possible to check the model's ability to discriminate between groups by running a cross-validation (add CV=TRUE) and comparing the predicted class (fit.predict$class) with the original class (d$Type). The table command below displays the confusion matrix for the discriminant model. When items are placed on the diagonal, it means that the model has correctly classified them and when the items are off the diagonal, they are mis-classified. We can compute the percentage of correctly classified elements by adding up the diagonal elements of the confusion matrix. As a rule of thumb, correct classification proportions below 80% are not really satisfactory.

```
> fit.predict <- lda(Type ~ Price  + Horsepower  + EngineSize +
+                    RPM + Fuel.tank.capacity + Length +
+                    Wheelbase + Turn.circle,
+                    data=d, na.action="na.omit", CV=TRUE)
> confusion <- table(d$Type, fit.predict$class)
> prop.table(confusion)
```

```
            Compact      Large    Midsize      Small     Sporty        Van
  Compact 0.13978495 0.00000000 0.02150538 0.00000000 0.01075269 0.00000000
  Large   0.00000000 0.09677419 0.01075269 0.00000000 0.00000000 0.01075269
  Midsize 0.04301075 0.03225806 0.16129032 0.00000000 0.00000000 0.00000000
  Small   0.02150538 0.00000000 0.00000000 0.19354839 0.01075269 0.00000000
  Sporty  0.02150538 0.00000000 0.00000000 0.04301075 0.08602151 0.00000000
  Van     0.00000000 0.01075269 0.01075269 0.00000000 0.00000000 0.07526882
```

```
> correct.pct <- 100 * sum(diag(prop.table(confusion)))
> correct.pct
```

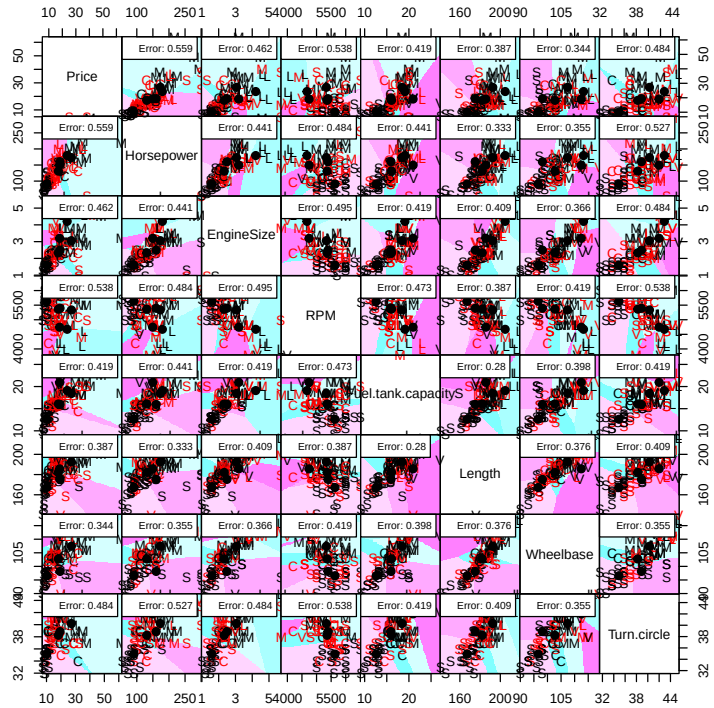
```
[1] 75.26882
```

In our model, we have 75.27% cases correctly classified. From the confusion table, we see for example that Midsize cars are confused with Compact and Large cars, that Vans are confused with Large and Midsize cars, and that Compact cars are confused with Midsize and Sporty cars.

To explore a bit more the quality of an lda model as a method to categorize observations, the partimat command from the klaR package command plots a matrix of variables taken two by two.

```
> # Run if the package is not installed
> # install.packages("klaR")
```

```
> library(klaR)
> partimat(Type ~ Price  + Horsepower  + EngineSize +
+           RPM + Fuel.tank.capacity + Length +
+           Wheelbase + Turn.circle,
+           data=d,
+           method="lda",
+           plot.matrix=TRUE)
```



7 Resources

- Quick R ... http://www.statmethods.net/
- Nice graphics ... http://had.co.nz/ggplot2/