APPM2720 More than one predictor

The power of least squares regeression is to use several variables simulatanously for prediction. The AudiA4 data is an example where using mileage and year together work much better than either alone. This lecture outlines some principles for multiple regression and also the mechanics for prediction.

The basic formula

The model with more than one variable is

Given predictors X*1, X*2, ... X_p and the variable of interest Y. A useful model is to predict the Ys by a straight line in X:

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

Note that this is no longer just a simple slope and intercept although each variable comes in as a linear addition to the prediction equation.

In **lm** this model for say 3 variables is fit by

```
fit<- lm( Y~ X1+ X2 + X3)
```

Often the data is bundled together as a data frame and there is a shortcut to specifying the **lm** formula. For the **AudiA4** data the model for mileage, year and distance is

```
fitA4<- lm( price ~ year + mileage + distance, data=AudiA4)
```

Here is a formula for fitting quadratic terms for year and mileage.

```
fitA4quad<- lm( price ~ year + I(year^2) + mileage + I(mileage^2), data=AudiA4)
```

After fitting the model

Always take a look at the residuals from the fitted model to make sure that there are no obvious patterns that would make use of the model inappropriate. `plot( fitA4)` is a handy way to get 4 different plots that help in this step.

Uncertainty in the estimated coefficients

To get a summary table of the estimated coefficients

```
     look$coefficients
```

the second column are the standard errors (SE) of the estimate coefficients. To get approximate 95% confidence intervals:

estimate $\pm$ 2* SE

In R some tricky code to do this is

```
look$coefficients[,1] + 2.0* c( -1,1)* look$coefficients[,2]
```

The value 2.0 is just a handy approximation (and easy to remember). The exact value is closer to 1.96 and in general for a M% confidence interval use the value

```
qnorm( .5* (1- M/100), lower.tail=FALSE)
```

e.g.

```
> M<- 99
> qnorm( .5* (1- M/100), lower.tail=FALSE)
[1] 2.575829
```

Making predictions.

The **predict** function in R is very flexible and designed to give prediction at new (or old) data points and also give standard errors for the predictions.

usedCar<- data.frame(mileage = 50000, year =2014, distance =0)

```
# just the prediction
est<- predict( fitA4, newdata=usedCar)
# just standard error of fit
SE<- predict( fitA4, newdata=usedCar,
               se.fit=TRUE)
# with 95% prediction
intervals<- predict( fitA4, newdata=usedCar,
        interval="prediction")
# reverse engineering the SE used for this
SE.prediction<- sqrt( (SE$se.fit)^2 +
         (SE$residual.scale)^2 )
```

The advantage of using the I function in the formula is that predict will take this into account

```
SE2<- predict( fitA4quad, newdata=usedCar, se.fit=TRUE)
```