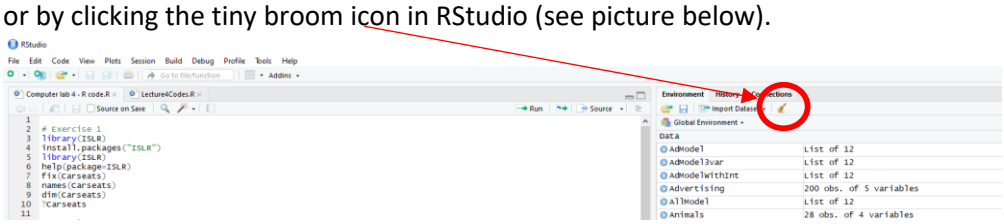
Computer lab 4 in MMS075, Feb 12, 2020

1. We can start the lab by removing all variables that we could possibly find from earlier sessions to ensure that we can start from scratch and previously defined variables do not interfere with new code that we may want to enter. We can do that in two ways; either using a command:
   rm(list=ls())
   or by clicking the tiny broom icon in RStudio (see picture below).



   We can do this later at any time when we feel that we want to start from scratch.

2. As usual, download the advertising example in ISL from http://faculty.marshall.usc.edu/gareth-james/ISL/data.html and save it on the Desktop (i.e. in a computer-specific folder). Import the by using the menu in RStudio choosing File > Import Dataset > From Text (base)…

   This time, we will not change the name of the dataset while importing it, which means that it will be called Advertising.

3. We can ask for a summary of the data which gives us some basic measures for each variable:
   summary(Advertising)
   Which variable has the largest mean and median values? Which value has the widest range of values?

4. When interpreting the summary, it may have been distracting that it included the index variable X. Therefore, we first remove that variable, ask for a summary again and also ask for a visualization of the data using "pairs":
   Advertising=Advertising[,-1]
   summary(Advertising)
   pairs(Advertising)

   The command "pairs" has created scatter plots for each pair of variables in the dataset. Do all predictors seem to be linearly related to the response "sales"? Do you see any signs of nonlinear relationship?

5. During the lecture, we have seen that interaction terms are relevant for predicting sales, and we now learn how to include interaction terms in linear regression in R. We first attach the dataset so the we can avoid writing "Advertising$" when referring to variables in this data frame:
   attach(Advertising)

We then look at the model with TV and radio as predictors and specify that we want to include their interaction term as well:
AdModelInt=lm(sales~TV+radio+TV:radio)

Note: I named this model as AdModelInt because it models the effect of ads and includes an interaction term. You should feel free to choose another name that you like more – all you need to do is to change AdModelInt in the command line above with the name that you choose. You just need to remember to use that name in all subsequent command lines when you refer to this model.

We can look at the summary of the model and see that it indeed includes the interaction term:
summary(AdModelInt)

Let us now try another command to define a model and look at the summary:
AdModelInt2=lm(sales~TV*radio)
summary(AdModelInt2)

Check that this output is exactly the same as the one above. It should be, because TV*radio is a shorthand for "include TV, radio and their interaction term", i.e. it does exactly the same thing as writing TV+radio+TV:radio.

Now, however, we do not need two identical models, so we remove the one that we defined later:
rm(AdModelInt2)

6. Revisit exercise 2 from yesterday, with the addition of quantifying uncertainty: using AdModelInt, predict the number of sold units and specify confidence and prediction intervals for the following advertisement budget distributions:
    a) TV budget: \$0, radio budget: \$100 000;
    b) TV budget: \$100 000, radio budget: \$0;
    c) TV budget: \$50 000, radio budget: \$50 000;
    d) TV budget: \$50 000, radio budget: \$50 000, newspaper budget: \$30 000.

    Furthermore, estimate the effect of:
    e) \$1000 increase of TV advertisement on sold units if the radio budget is \$10 000;
    f) \$1000 increase of TV advertisement on sold units if the radio budget is \$100 000;
    g) \$1000 increase of radio advertisement on sold units if the TV budget is \$50 000.

    Finally, predict the number of sold units for:
    TV budget: \$50 000, radio budget: \$51 000.

7. We now want to look at potential problems with this model. As the starting point, we can use the default graphs provided by R for linear models:
    plot(AdModelInt)

    This command will show you 4 different plots and you need to press Enter to switch between them. If you want to see them all together, you can divide the plotting screen as we did last time, using the "**par(mfrow=c(2,2))**" command before plotting the model.

The first plot shown is the residual plot, having the predicted values on the x-axis and the residuals on the y-axis. The last one showing standardized residuals vs leverage is used for identifying high leverage points.

For finding outliers, we prefer to look at studentized residuals that are very similar to standardized residuals but are not the same thing. Therefore, we produce a studentized residual plot as follows:
plot(predict(AdModelInt),rstudent(AdModelInt), xlab="Predicted sales (1000 units)",ylab="Studentized residuals",cex=1.5,pch=20)
If we include lines at y-values of -3 and 3, that will make it even easier to spot outliers:
abline(-3,0,lty="dashed",col="red")
abline(3,0,lty="dashed",col="red")

We may also want to produce our own residual plot and leverage plot as well:
plot(predict(AdModelInt),residuals(AdModelInt), xlab="Predicted sales (1000 units)",ylab="Residuals",cex=1.5,pch=20)
plot(hatvalues(AdModelInt),rstudent(AdModelInt), xlab="Leverage",ylab="Studentized residuals",cex=1.5,pch=20)
Give titles to these figures by using the title("…") command!

8. Revisit Exercise 14 on page 125 of ISL that was considered at the end of the exercise class yesterday. If you produce the required plots and summaries yourself instead of using the ones provided by me, then you will get a much better understanding of the relevant concepts. Additionally, you can experiment with other options and check whatever you find interesting about that model.

9. We check polynomial regression again, this time on a different dataset called Wage that is included in the ISLR library.
library(ISLR)
attach(Wage)
For a better understanding of the underlying data, we ask for the description of this dataset:
?Wage

We want to investigate the dependence of wage on age, so we first plot these variables:
plot(age,wage)

It looks like wage does not linearly depend on age. We can try to get a better fit with polynomials. We could try to include different powers of age as predictors; the following examples, taken from Section 7.8.1 of ISL, consider polynomials of degree 4:
fit=lm(wage~poly(age,4),data=Wage)
coef(summary(fit))
fit2=lm(wage~poly(age,4,raw=T),data=Wage)
coef(summary(fit2))
fit2a=lm(wage~age+I(age^2)+I(age^3)+I(age^4),data=Wage)
coef(fit2a)

Compare the results from the different outputs! The difference comes from the different ways of defining the predictors. The models fit2 and fit2a really use age, age^2, age^3 and age^4 as predictors while fit defines a degree 4 polynomial of age in a slightly different way.

Make an even more detailed comparison by looking at the full summaries:
summary(fit)
summary(fit2)

You will see that the general properties of the models are exactly the same. We will see that they also give the same predictions; copy the code below to make a nice plot using the first model:

```
agelims=range(age)
age.grid=seq(from=agelims[1],to=agelims[2])
preds=predict(fit,newdata=list(age=age.grid),se=TRUE)
se.bands=cbind(preds$fit+2*preds$se.fit,preds$fit-2*preds$se.fit)
par(mfrow=c(1,2),mar=c(4.5,4.5,1,1),oma=c(0,0,4,0))
plot(age,wage,xlim=agelims,cex=.5,col="darkgrey")
title("Degree-4 Polynomial",outer=T)
lines(age.grid,preds$fit,lwd=2,col="blue")
matlines(age.grid,se.bands,lwd=1,col="blue",lty=3)
```

Now try adding a similar plot using the second model (i.e. fit2)!

We can also check that the difference in predictions is essentially zero:

```
preds2=predict(fit2,newdata=list(age=age.grid),se=TRUE)
max(abs(preds$fit-preds2$fit))
```

10. If you are done with the previous exercises, do exercises 9 and 13 in Section 3.7 of ISL (pages 122-125). The Auto dataset is included in the ISLR library, so it does not need to be downloaded separately.

11. If you would like to ask something or give feedback, feel free to talk to me or enter your question/feedback at www.menti.com, using the code 69 36 52.