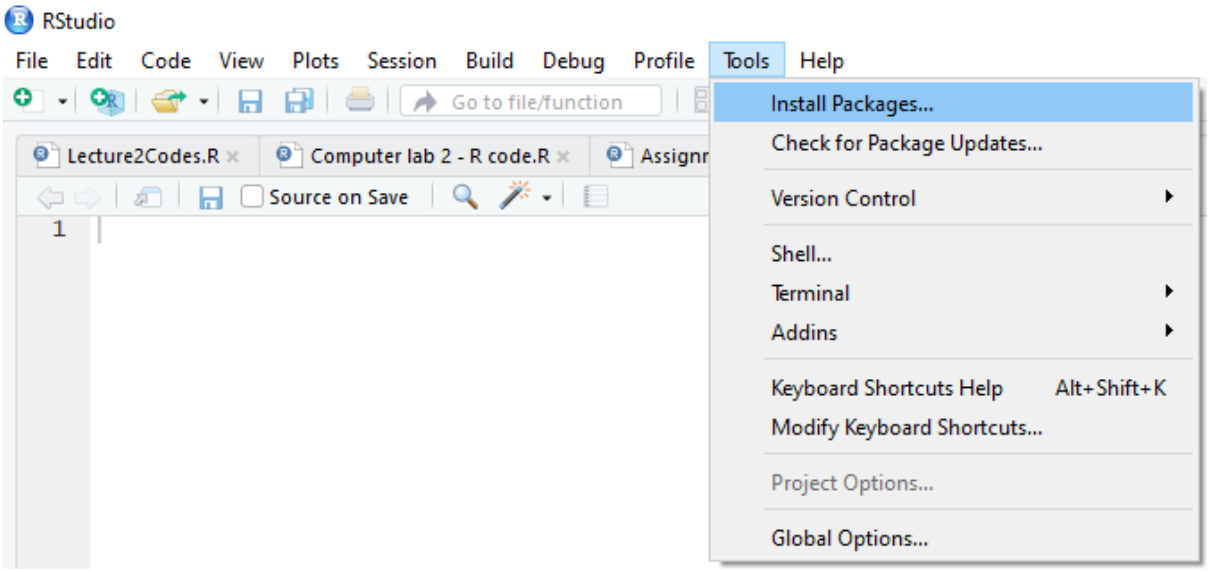
Computer lab 3 in MMS075, Feb 6, 2020

1. In this computer lab, we will work with the Carseats dataset that is used in Section 3.6.6 of the ISL book. Several datasets from ISL are available in R; to access these, type:
library(ISLR)

   Did you get an error message? That's probably because this library is not installed on your computer. To install it, either use the command install.package("ISLR") or use the Install Packages function of RStudio (see picture below)



   Now that you have installed ISLR, you need to tell RStudio that you want to use it, and hopefully, the corresponding command will not cause an error message this time:
library(ISLR)

   To see the content and description of the ISLR library, type:
help(package=ISLR)

   Now you can check the Carseats dataset:
fix(Carseats)
Don't forget to close the editor before you continue with other commands.

   If you just wanted to see the names of variables included in the data, you could type:
names(Carseats)

   To see the size of the dataset, type:
dim(Carseats)
The output shows the dimensions of the dataset, i.e. number of rows and number of columns.

   To learn what the different variables mean, type:
?Carseats

2. How good are the different variables in predicting the sales? Fit a linear model that includes all variables to learn about this. Perform backward variable selection: in each step, remove the variable with the highest p-value until you get a model where all variables are significant. Recall: the relevant function for fitting linear regression models is "lm(…)" and you get a summary of the model by typing "summary(…)"

3. Was it boring to type all variable names to get the all variable model? Replace CompPrice+Income+Advertising+…+US in your code with a single dot (yes, just the "." symbol without the quotation signs!). What to do if we need almost all variables except Population and Education? Use ".-Population-Education".

4. According to this dataset, what is the contribution of a good shelf location compared to a bad one? What is the contribution of a medium shelf location? How is the qualitative variable ShelveLoc represented in this model?

5. To see the dummy variables that R defined for this variable, type:
   contrasts(ShelveLoc)

6. We will now learn how to do best subset variable selection in R. The relevant function is called "regsubsets" and is included in a library called "leaps". Therefore, type:
   library(leaps)
   If you get an error message, install the relevant package and type the above command again

7. To tell R to do the model selection and display a summary, type:
   BestSSModel=regsubsets(Sales~.,data=Carseats)
   summary(BestSSModel)

   This will select the best models up to 8 variables. We would like to consider all possible models, so we use the "nvmax" argument to increase the number of considered variables to the number of variables in the dataset and look at the summary:
   BestSSModel=regsubsets(Sales~.,data=Carseats,nvmax=11)
   summary(BestSSModel)

   We will soon see that it makes sense to define an object that contains the relevant data in the summary, so let's do that and check what kind of information it contains:
   BestSSsummary=summary(BestSSModel)
   names(BestSSsummary)

   For example, we can see the R^2 values of the best models of different sizes:
   BestSSsummary$rsq

   We will now choose the best model of *any* size from these models (which is the point with the best subset selection algorithm). Recall, however, that there are different possibilities for measuring the quality of the model, e.g. Mallow's C_p, AIC, BIC and adjusted R^2. We will now see that they do not always give the same best model. For BIC, the best model is the one with the lowest value. For adjusted R^2, the best model is the one with the highest value. Use the "which.max" and "which.min" functions to check which models these are:

which.max(BestSSsummary$adjr2)
which.min(BestSSsummary$bic)

To get an even better understanding, we make some plots to show how the quality of models of different sizes changes. We want to show 4 plots together, so it is good to divide the plotting window as 2x2:
par(mfrow=c(2,2))

We plot the different values for adjusted R^2 in the first quarter:
plot(BestSSsummary$adjr2,xlab="Number of variables",ylab="Adjusted R2")
After this, plots will fill up the quarters of this window. If you make a mistake and want to start again with the plot, type the following command:
dev.off()

We mark the point with the highest value:
BestModelSizeAdjR2=which.max(BestSSsummary$adjr2)
points(BestModelSizeAdjR2,BestSSsummary$adjr2[BestModelSizeAdjR2],col="red",cex=2, pch=20)

We do the same using BIC and mark the point with the lowest value:
plot(BestSSsummary$bic,xlab="Number of variables",ylab="BIC")
BestModelSizeBIC=which.min(BestSSsummary$bic)
points(BestModelSizeBIC,BestSSsummary$bic[BestModelSizeBIC],col="red",cex=2,pch=20)

To avoid the manual work with plotting, we can use a built-in function in R:
plot(BestSSModel,scale="adjr2")
plot(BestSSModel,scale="bic")

To get the coefficients of the best models, type:
coef(BestSSModel,7)
coef(BestSSModel,10)

8. Polynomial regression is quite easy to do in R; we will demonstrate this with the Auto dataset in the (already loaded) ISLR library. We can attach Auto to avoid having to write Auto$ all the time:
attach(Auto)

We can then define polynomial regression models of different degrees:
hpmodel=lm(mpg~horsepower)
summary(hpmodel)
hp2model=lm(mpg~I(horsepower^2)+horsepower)
summary(hp2model)
hp3model=lm(mpg~I(horsepower^3)+I(horsepower^2)+horsepower)
summary(hp3model)
hp4model=lm(mpg~I(horsepower^4)+I(horsepower^3)+I(horsepower^2)+horsepower)
summary(hp4model)
hp5model=lm(mpg~I(horsepower^5)+I(horsepower^4)+I(horsepower^3)+I(horsepower^2) +horsepower)

summary(hp5model)

For higher degrees, it is even more convenient to use the poly function:
hp10model=lm(mpg~poly(horsepower, 10))
summary(hp10model)

Alternatively, we can start with a linear model:
lin=lm(mpg~horsepower)
summary(lin)
and update it step by step:
quad=update(lin,~.+I(horsepower^2))
summary(quad)
cub=update(quad,~.+I(horsepower^3))
summary(cub)

We can also create the plots shown during the lecture:
plot(horsepower,mpg)
abline(hpmodel$coefficients[1],hpmodel$coefficients[2],col="orange",lwd=2)
lines(seq(40,250,0.01), predict(hp2model,data.frame(horsepower=seq(40,250,0.01)))
,lwd=2, col="blue")
lines(seq(40,250,0.01), predict(hp5model,data.frame(horsepower=seq(40,250,0.01)))
,lwd=2, col="green")
lines(seq(40,250,0.01), predict(hp10model,data.frame(horsepower=seq(40,250,0.01)))
,lwd=2, col="red")

9. If you are done with the previous exercises, do exercises 8 and 10 in Section 2.4 of ISL (pages 54-56). Recall that the book can be downloaded for free at http://faculty.marshall.usc.edu/gareth-james/ISL/index.html. Note that the College dataset is included in the ISLR library, so it does not need to be downloaded from the internet.

10. If you would like to ask something or give feedback about this computer lab, feel free to talk to me or enter your question/feedback at www.menti.com, using the code 35 97 09.