MTH 4230 Lab 7

Due Wed., Wed. Apr. 22

1 Part A: Model Selection Using the t Test Results

1.1 Patient Satisfaction Data Set

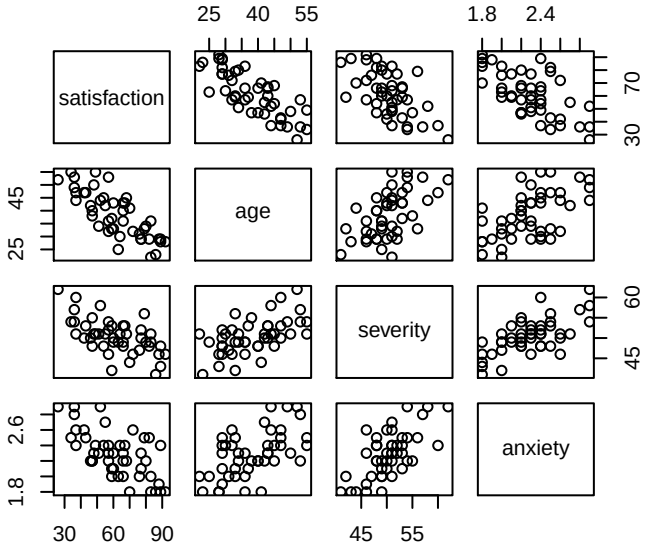
A hospital administrator wished to study the relation between patient satisfaction (Y) and patient's age (X_1, in years), severity of illness (X_2, an index), and anxiety (X_3, an index). The administrator collected the data on $\boldsymbol{n = 46}$ randomly selected patients. This is the **Patient Satisfaction** data from **Problem 6.15** of the textbook. They're in the file **satisfaction.txt**.

One simple method of **model selection** is to successively drop the term whose p-value is largest (as long as that term isn't significant and isn't included in a significant higher-order interaction), **refitting** the model each time a term is dropped.

1. Read the data from **satisfaction.txt** into R using `read.table()`.
2. Create a **scatterplot matrix**, for example by typing something like:

```
pairs(my.data)
```

   It should look like this:



3. Fit the multiple regression model, with **satisfaction** as the response and with **three** predictors and their **two-** and **three-way interactions**:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_1 X_2 + \beta_5 X_1 X_3 + \beta_6 X_2 X_3 + \beta_7 X_1 X_2 X_3 + \epsilon$$

where

$\boldsymbol{Y}$ = **Satisfaction**
$\boldsymbol{X_1}$ = **Age**
$\boldsymbol{X_2}$ = **Severity**
$\boldsymbol{X_3}$ = **Anxiety**

either by listing the interactions explicitly (using the `:` operator):

```
my.reg <- lm(satisfaction ~ age + severity + anxiety +
             age:severity + age:anxiety + severity:anxiety +
             age:severity:anxiety,
             data = my.data)
```

or by using the `*` operator:

```
my.reg <- lm(Rent.Rate ~ age * severity * anxiety, data = my.data)
```

(You'll be using modified versions of the lengthier first command above in later steps.)

Then view the results using `summary()`.

4. You should've found (in the last step) that the three-way interaction isn't significant, so it can be **dropped** from the model. **Refit** the model *without* the three-way interaction term:

```
my.reg <- lm(satisfaction ~ age + severity + anxiety +
             age:severity + age:anxiety + severity:anxiety,
             data = my.data)
```

5. You should've found (after refitting the model in the last step) that not all of the two-way interactions are significant, so the one whose **p-value** is **largest** can be **dropped** from the model. **Refit** the model of Step 4 *without* that two-way interaction term.

6. You should've found (after refitting the model in the last step) that at least one of the remaining two-way interactions isn't significant, so the one whose **p-value** is **largest** can be **dropped** from the model. **Refit** the model of Step 5 *without* that two-way interaction term.

7. You should have found (after refitting the model in the last step) that the one remaining two-way interaction isn't significant, so it too can be **dropped**. **Refit** the model without any interactions:

```
my.reg <- lm(satisfaction ~ age + severity + anxiety, data = my.data)
```

Now repeat the process above to decide which, if any, of the three single-variable predictors should be dropped from the model. You should terminate the process, ending with a **final model**, when all of the remaining predictors are significant.

2 Part B: Model Selection Using the AIC Criterion

2.1 Patient Satisfaction Data Set (Cont'd)

Another way to decide whether to drop a predictor from a model is the ***AIC model selection criterion***. If dropping the predictor **lowers** the **AIC** value, it should be **dropped**.

We select our **final model** by successively dropping the term that **lowers** the **AIC** the most (as long as that term isn't included in a higher-order interaction in the model), **refitting** the model each time a term is dropped.

(We could also base the decision on whether dropping the predictor **raises** the $\boldsymbol{R^2_{\text{adj}}}$ value.)

1. To illustrate, we'll apply the method to the model with **no interactions**:

$$Y \;=\; \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

   Fit the model by typing:

```
my.reg <- lm(satisfaction ~ age + severity + anxiety, data = my.data)
```

2. The function `drop1()`, when passed an *lm* object like `my.reg`, will drop one predictor at a time from a model, and report the **AIC** value for each model. Type:

```
drop1(my.reg)
```

   Here's what's shown in the output:

   - The first column in the indicates which predictor (if any) is **dropped** from the model. Each line corresponds to the model with just that predictor omitted from the model (i.e. they're *not* dropped sequentially).
   - The second and third columns give the ***degrees of freedom*** for and value of the ***extra sum of squares*** associated with the predictor in that line.
   - The fourth column gives the ***error sum of squares*** (or ***residual sum of squares***) for the model in which the predictor in that line is **left out**.
   - The fifth column gives the **AIC** value for the model in which the predictor in that line is **left out**.

   (In practice, we'd **drop** the term that **lowers** the **AIC** the most, then **refit** the model *without* that term, apply `drop1()` again, and repeat the process until arriving at the **final model** for which no more term deletions can lower the **AIC**. We won't do it here, though.)

3. In Step 2, we applied `drop1()` to a model with **no interactions**.

   Which terms does `drop1()` drop when you apply it to a model with **two-** and **three-way interactions**:

```
my.reg <- lm(satisfaction ~ age + severity + anxiety +
             age:severity + age:anxiety + severity:anxiety +
             age:severity:anxiety,
             data = my.data)

drop1(my.reg)
```

Which terms does `drop1()` drop when you apply it to a model with **two-way interactions** only:

```
my.reg <- lm(satisfaction ~ age + severity + anxiety +
             age:severity + age:anxiety + severity:anxiety,
             data = my.data)

drop1(my.reg)
```

3 Part C: Automated Model Selection

3.1 Pesticide Biodegradation Data Set

Biodegradation of pesticides in the environment results from the activities of soil microorganisms. Degradation rates can vary due the abundance of such organisms and soil chemical properties such as organic matter content, pH, and nutrient supply.

A study was carried out to investigate the influence of these soil characteristics on the **degradation rate** of the herbicide *isoproturon*. Soil specimens collected from $\boldsymbol{n = 20}$ sites were analyzed for nitrate, potassium, phosphorus, pH, organic matter, and microbial biomass. Each soil specimen was then treated with *isoproturon* and monitored for 65 days for residues of the herbicide.

The file **degradation_rates.txt** contains data on the following variables:

$\mathbf{DT_{50}}$ = The time (days) required for the isoproturon concentration to decrease by 50%
Kd = Kinetic degradation, or rate constant exponential decrease in isoproturon (day^{-1})
Site = Collection site
Nitrate = Nitrate concentration (mg/kg)
Potassium = Potassium concentration (mg/kg)
Phosphorus = Phosphorus concentration (mg/kg)
PH = Acidity (pH)
OrganicMatter = Organic matter (%)
Biomass = Microbial biomass (mg C/kg)

We'll consider models with $\mathbf{DT_{50}}$ as the response and **Nitrate**, **Potassium**, **Phosphorus**, **PH**, **OrganicMatter**, and **Biomass** as possible predictors.

There are $2^6 = \mathbf{64}$ possible models containing all six or fewer of the predictors. We want one that **fits** the data **well** but is **parsimonious**.

We *could* fit all 64 models and use the one with the **lowest AIC** value (or **largest** R^2_{adj}).

Instead, we'll perform an ***automated model selection*** procedure. We have four choices:

- ***Backward elimination***
- ***Forward selection***
- ***Backward stepwise***
- ***Forward stepwise***

The four procedures *don't* always lead to the same **final model**.

1. Read the data into R using `read.table()`, then remove the (unneeded) `Kd` and `Site` columns:

```
my.data$Kd <- NULL
my.data$Site <- NULL
```

2. Create a ***scatterplot matrix*** of the data and compute the ***correlation matrix***:

```
pairs(my.data)

cor(my.data)
```

3. We'll first do a ***backward elimination*** procedure.

   The **starting model** contains **all six** predictors. One at a time, the predictor whose removal leads to the **largest reduction** in the **AIC** is removed from the model, until none of the remaining predictors' removals would decrease the **AIC**. Each time a predictor is removed, the model is refitted before deciding which one to remove next.

   (Alternatively, the largest **p-value** could be used as the removal criterion instead of the largest **AIC** reduction. This is what we did in Part A.)

   Fit the **starting model**:

```
my.reg <- lm(DT.50 ~ Nitrate + Potassium + Phosphorus + PH + OrganicMatter + Biomass,
             data = my.data)
```

   Use `step()` to carry out the **backward elimination** procedure by typing:

```
step(my.reg, direction = "backward")
```

4. Now we'll do a ***forward selection*** procedure.

   The **starting model** contains **just an intercept**. One at a time, the predictor whose addition to the model leads to the **largest reduction** in the **AIC** is added to the model, until none of the remaining predictors' additions would decrease the **AIC**. Each time a predictor is added, the model is refitted before deciding which one to add next.

   Fit the **starting model**:

```
my.reg <- lm(DT.50 ~ 1, data = my.data)
```

   To carry out the **forward selection** procedure, we specify the "scope" of the models we want to investigate, i.e. the largest set of predictors we're willing to include in the model, via the `scope` argument in `step()`:

```
step(my.reg, direction = "forward",
     scope = DT.50 ~ Nitrate + Potassium + Phosphorus + PH + OrganicMatter + Biomass)
```

5. Now we'll do a ***backward stepwise*** procedure.

   The **starting model** contains **all six** predictors. One at a time, at each step, the predictor whose **addition *or* removal** leads to the **largest reduction** in the **AIC** is **added** to ***or* removed** from the model, until none of the remaining predictors' additions or removals would decrease the **AIC**.

   (Alternatively, the largest **p-value** could be used as the or removal criterion, and the smallest **p-value** as the addition criterion, instead of the largest **AIC** reduction.)

   Fit the **starting model**:

```
my.reg <- lm(DT.50 ~ Nitrate + Potassium + Phosphorus + PH + OrganicMatter + Biomass,
             data = my.data)
```

   Use `step()` to carry out the **backward stepwise** procedure by typing:

```
step(my.reg, direction = "both")
```

6. Now we'll do a ***forward stepwise*** procedure.

   The **starting model** contains **just an intercept**. One at a time, at each step, the predictor whose **addition *or* removal** leads to the **largest reduction** in the **AIC** is **added** to ***or* removed** from the model, until none of the remaining predictors' additions or removals would decrease the **AIC**.

   Fit the **starting model**:

```
my.reg <- lm(DT.50 ~ 1, data = my.data)
```

Use `step()` to carry out the **forward stepwise** procedure by typing:

```
step(my.reg, direction = "both",
     scope = DT.50 ~ Nitrate + Potassium + Phosphorus + PH + OrganicMatter + Biomass)
```