MTH 4230 R Notes 7

1 Models with Quantitative and Qualitative Predictors

1.1 Qualitative Variables and the Design Matrix

- For models that include ***qualitative*** (or ***categorical***) variables, that is, ***factors***, the design matrix $\boldsymbol{X}$ will include columns of 0's and 1's that are used to ***code*** those variables.

  There are different ways to **code** such *factors* by ***indicator variables*** (0 or 1 variables). The particular coding system R uses can be set via the `contrasts` argument in the `options()` function.

1.2 Fitting ANOVA Models Using `lm()` with Indicator Variables

- A regression model that **only** includes qualitative predictors is called an ***ANOVA model***. There are three ways to fit an ***ANOVA model*** in R.

  1. We can **code** the levels of the qualitative predictor as **indicator variables** and then include them in the model *formula* passed to the `lm()` function.
  2. We can make sure our qualitative predictor is a `"character"` vector (or *factor*) and then include it in the model *formula* passed to `lm()`.
  3. We can make sure our qualitative predictor is a `"character"` vector (or *factor*) and then pass it via the model *formula* to the `aov()` function.

1.2.1 Qualitative Predictors with Two Levels

- Consider the following data.

```
y <- c(16, 19, 30, 21, 18, 17, 13, 15, 10, 13)
trt <- c("low", "low", "low", "low", "low", "high", "high", "high", "high", "high")
```

  We can fit an **ANOVA model** by coding `"low"` as 0 and `"high"` as 1, for example using the `ifelse()` function, and then include the **coded indicator variable** in the model *formula* in `lm()`.

  Here we create the **indicator variable**:

```
trt.coded <- ifelse(trt == "low", 0, 1)
trt.coded

##  [1] 0 0 0 0 0 1 1 1 1 1
```

Now we include it in the model *formula* passed to `lm()`:

```
my.reg <- lm(y ~ trt.coded)
summary(my.reg)

##
## Call:
## lm(formula = y ~ trt.coded)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4.80  -2.55  -0.60   1.10   9.20
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.800      1.910  10.887 4.48e-06
## trt.coded     -7.200      2.702  -2.665   0.0286
##
## Residual standard error: 4.272 on 8 degrees of freedom
## Multiple R-squared:  0.4702,Adjusted R-squared:  0.404
## F-statistic: 7.101 on 1 and 8 DF,  p-value: 0.02859
```

From the output, we conclude (among other things) that:

– Based on the ***t test statistic*** ($\boldsymbol{t = -2.665}$) and ***p-value*** (**0.0286**), the treatment has an effect on the response y.

– The ***estimate of*** $\boldsymbol{\beta_1}$ is $\boldsymbol{b_1 = -7.2}$, indicating that we estimate that the true mean response for ”high” is -7.2 units lower than the mean response for ”low”.

1.2.2 Qualitative Predictors with More than Two Levels

- Qualitative predictors with **more than two levels** can be coded explicitly using `ifelse()` too.

- Consider the following data:

```
col.vec <- c("red", "red", "green", "green", "blue", "blue")
y2 <- c(3.3, 4.2, 6.1, 4.9, 4.8, 4.7)
```

Here, we code the qualitative variable as **two indicator variables** (then use `cbind()` to look at them simultaneously as a *matrix*):

```
col.coded1 <- ifelse(col.vec == "red", 1, 0)
col.coded2 <- ifelse(col.vec == "green", 1, 0)
cbind(col.coded1, col.coded2)

##      col.coded1 col.coded2
## [1,]          1          0
## [2,]          1          0
## [3,]          0          1
## [4,]          0          1
## [5,]          0          0
## [6,]          0          0
```

The coded variables can then used with `lm()` to fit the ***ANOVA model***:

```
my.reg <- lm(y2 ~ col.coded1 + col.coded2)
```

The results are below:

```
summary(my.reg)

##
## Call:
## lm(formula = y2 ~ col.coded1 + col.coded2)
##
## Residuals:
##     1     2     3     4     5     6
## -0.45  0.45  0.60 -0.60  0.05 -0.05
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.7500     0.4340  10.945  0.00163
## col.coded1   -1.0000     0.6137  -1.629  0.20172
## col.coded2    0.7500     0.6137   1.222  0.30896
##
## Residual standard error: 0.6137 on 3 degrees of freedom
## Multiple R-squared:  0.7318,Adjusted R-squared:  0.553
## F-statistic: 4.093 on 2 and 3 DF,  p-value: 0.1389
```

From the output, among other things we conclude that:

- The ***estimate of*** $\boldsymbol{\beta_1}$ is $\boldsymbol{b_1 = -1}$. Thus we estimate that the mean response for ”red” is -1 unit lower than the mean response for ”blue”.
- The ***estimate of*** $\boldsymbol{\beta_2}$ is $\boldsymbol{b_2 = 0.75}$. Thus we estimate that the mean response for ”green” is 0.75 units higher than the mean response for ”blue”.

1.3 Fitting ANOVA Models Using `lm()` with `"character"` Vectors or *Factors*

- Instead of coding a qualitative predictor as ***indicator variables*** ourselves, we can let R do it internally as it fits the regression model. We just represent the predictor as a `"character"` vector or *factor*, and then include it in the *formula* in the call to `lm()`.

1.3.1 Qualitative Predictor with Two Levels

- Consider the `"character"` vector `trt` created above:

```
trt

##  [1] "low"  "low"  "low"  "low"  "low"  "high"
##  [7] "high" "high" "high" "high"
```

We can use it in the model *formula* in a call to `lm()`:

```
my.reg <- lm(y ~ trt)
```

Before looking at the results of the regression, it's helpful to look at the **design matrix** $\boldsymbol{X}$:

```
model.matrix(my.reg)

##    (Intercept) trtlow
## 1            1      1
## 2            1      1
## 3            1      1
## 4            1      1
## 5            1      1
## 6            1      0
## 7            1      0
## 8            1      0
## 9            1      0
## 10           1      0
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$trt
## [1] "contr.treatment"
```

We see that R created an **indicator variable** and named it `"trtlow"`, indicating that **1** represents the ”low” level of the variable `"trt"` and **0** represents ”high”. (Note that this is the opposite of how we coded them when we did it manually.)

Now we're ready to look at the results of the regression analysis:

```
summary(my.reg)

##
## Call:
## lm(formula = y ~ trt)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
##  -4.80  -2.55  -0.60   1.10   9.20
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   13.600      1.910   7.119   0.0001
## trtlow         7.200      2.702   2.665   0.0286
##
## Residual standard error: 4.272 on 8 degrees of freedom
## Multiple R-squared:  0.4702,Adjusted R-squared:  0.404
## F-statistic: 7.101 on 1 and 8 DF,  p-value: 0.02859
```

From the output, we can conclude (among other things) that:

- Based on the ***t* test statistic** ($\boldsymbol{t = 2.665}$) and ***p-value*** (**0.0286**), the treatment has an effect on the response y.
- The ***estimate of*** $\boldsymbol{\beta_1}$ is $\boldsymbol{b_1 = 7.2}$. Thus we estimate that the mean response for ”low” is 7.2 units higher than the mean response for ”high”.

1.3.2 Qualitative Predictors with More than Two Levels

- A qualitative predictor with *more than two* levels can be included as a `"character"` vector or *factor* in the *formula* passed to `lm()` too. For example:

```
cols <- c("red", "red", "green", "green", "blue", "blue")
y2 <- c(3.3, 4.2, 6.1, 4.9, 4.8, 4.7)
```

Next we fit the model using `lm()`:

```
my.reg <- lm(y2 ~ cols)
```

Before looking at the results, let’s see how the levels of the qualitative predictor were **coded** in R by looking at the **design matrix** $\boldsymbol{X}$:

```
model.matrix(my.reg)

##   (Intercept) colsgreen colsred
## 1           1         0       1
## 2           1         0       1
## 3           1         1       0
## 4           1         1       0
## 5           1         0       0
## 6           1         0       0
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$cols
## [1] "contr.treatment"
```

Notice that R calls the two **indicator variables** `colsgreen` and `colsred`. For the first, a **1** represents ”green” (and **0** everything else), and for the second a **1** represents ”red” (and **0** everything else). Therefore, when **both** `colsgreen` and `colsred` are **0**, it’s ”blue”.

Another way to view the coding is by using the `contrasts()` function:

```
contrasts(as.factor(cols))

##       green red
## blue      0   0
## green     1   0
## red       0   1
```

- The regression results are below:

```
summary(my.reg)

##
## Call:
## lm(formula = y2 ~ cols)
##
## Residuals:
##     1     2     3     4     5     6
## -0.45  0.45  0.60 -0.60  0.05 -0.05
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.7500     0.4340  10.945  0.00163
## colsgreen     0.7500     0.6137   1.222  0.30896
## colsred      -1.0000     0.6137  -1.629  0.20172
##
## Residual standard error: 0.6137 on 3 degrees of freedom
## Multiple R-squared:  0.7318,Adjusted R-squared:  0.553
## F-statistic: 4.093 on 2 and 3 DF,  p-value: 0.1389
```

From the output, among other things, we can conclude that:

  - The ***estimate of*** $\boldsymbol{\beta_1}$ is $\boldsymbol{b_1 = 0.75}$. Thus we estimate that the mean response for ”green” is 0.75 units higher than the mean response for ”blue”.
  - The ***estimate of*** $\boldsymbol{\beta_2}$ is $\boldsymbol{b_2 = -1}$, so we estimate that the mean response for ”red” is **-1** units lower than the mean response for ”blue”.

1.4 Fitting ANOVA Models Using `aov()`

- The usual way to carry out an **ANOVA** in R is with the `aov()` function.

```
aov()     # Fit an analysis of variance model and carry out an analysis
          # of variance
```

Like `lm()`, `aov()` takes a *formula* as it's main argument and an optional argument `data` (a *data frame* containing the variables used in the *formula*). The predictors in the *formula* should be `"character"` vectors or *factors*.

The `aov()` function actually just calls `lm()` for model fitting, but its output differs from that of `lm()`.

- Here we fit the ***one-factor ANOVA model*** with `"character"` vector `cols` (from above) as the factor and response variable `y2` (also from above):

```
my.anova <- aov(y2 ~ cols)
```

The object `my.anova` created above belongs to the class *aov*, which is a subset of the class *lm* (so it belongs to the *lm* class too).

```
class(my.anova)

## [1] "aov" "lm"
```

For *aov* objects, the function `summary()` produces the ***ANOVA table***:

```
summary(my.anova)

##             Df Sum Sq Mean Sq F value Pr(>F)
## cols         2  3.083  1.5417   4.093  0.139
## Residuals    3  1.130  0.3767
```

- ***Multi-factor ANOVA***s can be carried out by including the additional `"character"` vectors or *factors* in the *formula* passed to `aov()`. ***Interactions*** can be specified in the *formula* using the `:` or `*` symbols. Type `?formula` for more info.

1.5 Fitting ANCOVA Models Using `lm()` with Indicator Variables

- We'll refer to regression models that contain a mix of qualitative and quantitative predictors as ***analysis of covariance*** (or ***ANCOVA***) ***models***. *

* Some people use the term ***analysis of covariance*** specifically to refer to models in which the qualitative predictor is the main predictor of interest, and the quantitative ones are included in the model to *control* for them.

- There are two ways to fit ***ANCOVA models*** in R.

  1. We can code the levels of the qualitative predictor as **indicator variables** and then include them as predictors in the ***formula*** passed to `lm()`.

2. We can make sure our qualitative predictor is a `"character"` vector or *factor* and then include it in the model *formula* passed to `lm()`.

- Consider, for example, the following data.

```
y <- c(16, 19, 30, 21, 18, 17, 13, 15, 10, 13)
trt <- c("low", "low", "low", "low", "low", "high", "high", "high", "high", "high")
x <- c(1, 3, 7, 6, 6, 5, 2, 3, 0, 1)
```

Notice that one of the predictors (`trt`) is qualitative and the other (`x`) is quantitative.

As was done previously, we can code the levels of `trt` using 0 for `"low"` and 1 for `"high"`:

```
trt.coded <- ifelse(trt == "low", 0, 1)
```

1.5.1 The No-Interaction Model and Test for Equality of Two Lines

- Now the **ANCOVA model** is easily fit using `lm()`. Below we fit a model with **no interaction**:

```
my.reg <- lm(y ~ trt.coded + x)
summary(my.reg)

##
## Call:
## lm(formula = y ~ trt.coded + x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8300 -0.5975 -0.0350  0.4950  5.7200
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.1300     2.5390   5.565 0.000846
## trt.coded    -3.7200     2.1933  -1.696 0.133688
## x             1.4500     0.4702   3.084 0.017718
##
## Residual standard error: 2.974 on 7 degrees of freedom
## Multiple R-squared:  0.7754,Adjusted R-squared:  0.7112
## F-statistic: 12.08 on 2 and 7 DF,  p-value: 0.00537
```

From the output, among other things, we conclude that:

- The ***estimate of*** $\boldsymbol{\beta_1}$ is $\boldsymbol{b_1 = -3.72}$. Thus we estimate that the gap between the ”low” and ”high” lines is -3.72 units.
- The ***test statistic*** for testing for **equality** of the **two lines**, one representing ”low” and the other ”high”, (i.e. testing whether the gap between the lines is 0) is $\boldsymbol{t = -1.696}$ and the ***p-value*** is **0.1337**.

1.5.2 The Interaction Model and Test for Equal Slopes

- **Interactions** can be included in the model *formula* using the `:` or `*` symbol.
- Here we fit the model **with** an **interaction** term to the data in the vectors `y`, `trt.coded`, and `x` from above.

```
my.reg <- lm(y ~ trt.coded + x + trt.coded:x)
summary(my.reg)

##
## Call:
## lm(formula = y ~ trt.coded + x + trt.coded:x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9444 -0.6216  0.0405  0.6984  5.5238
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.7540     3.2630   4.215  0.00559
## trt.coded    -3.0378     4.0055  -0.758  0.47694
## x             1.5317     0.6375   2.403  0.05309
## trt.coded:x  -0.2209     1.0480  -0.211  0.84001
##
## Residual standard error: 3.2 on 6 degrees of freedom
## Multiple R-squared:  0.7771,Adjusted R-squared:  0.6656
## F-statistic: 6.971 on 3 and 6 DF,  p-value: 0.02212
```

Among other things, the output tells us that:

  - The ***test statistic*** for testing for **equality** of the **two slopes**, one for the ”low” line and the other the ”high” line, is $\boldsymbol{t = -0.211}$ and the ***p-value*** is **0.84**.

- For qualitative predictors with **more than two levels**, coding by **indicator variables** can be performed as described in the Subsection 1.2.2.

  Models with **more than one** qualitative predictor **and/or more than one** quantitative predictor can be fit as well.

1.6 Fitting ANCOVA Models Using `lm()` with `"character"` Vectors or *Factors*

- Instead of creating **indicator variables** ourselves, we can fit **ANCOVA models** by including a `"character"` vector or *factor* in the model *formula* passed to `lm()`, as was done previously (for **ANOVA**).

1.6.1 The No-Interaction Model and Test for Equality of Two Lines

- Now we fit the **ANCOVA model** (**without** an **interaction**) by including the `"character"` vector `trt` (from above) in the model *formula* passed to `lm()`:

  ```
  my.reg <- lm(y ~ trt + x)
  ```

  As with **ANOVA**, it's useful to look at the **design matrix** $\boldsymbol{X}$:

  ```
  model.matrix(my.reg)

  ##    (Intercept) trtlow x
  ## 1            1      1 1
  ## 2            1      1 3
  ## 3            1      1 7
  ## 4            1      1 6
  ## 5            1      1 6
  ## 6            1      0 5
  ## 7            1      0 2
  ## 8            1      0 3
  ## 9            1      0 0
  ## 10           1      0 1
  ## attr(,"assign")
  ## [1] 0 1 2
  ## attr(,"contrasts")
  ## attr(,"contrasts")$trt
  ## [1] "contr.treatment"
  ```

  We see that R created an **indicator variable** and named it `trtlow`, telling us that ”low” is **coded** as **1** and ”high” as **0**.

  We're now ready to look at the output:

  ```
  summary(my.reg)

  ##
  ## Call:
  ## lm(formula = y ~ trt + x)
  ##
  ## Residuals:
  ##     Min      1Q  Median      3Q     Max
  ## -4.8300 -0.5975 -0.0350  0.4950  5.7200
  ##
  ## Coefficients:
  ##             Estimate Std. Error t value Pr(>|t|)
  ## (Intercept)  10.4100     1.6848   6.179 0.000455
  ## trtlow        3.7200     2.1933   1.696 0.133688
  ## x             1.4500     0.4702   3.084 0.017718
  ```

```
##
## Residual standard error: 2.974 on 7 degrees of freedom
## Multiple R-squared:  0.7754,Adjusted R-squared:  0.7112
## F-statistic: 12.08 on 2 and 7 DF,  p-value: 0.00537
```

From the output we conclude, among other things,

- The ***estimate of*** $\boldsymbol{\beta_1}$ is $\boldsymbol{b_1 = 3.72}$. Thus we estimate that the gap between the ”low” and ”high” lines is 3.72 units.
- The ***test statistic*** for testing for **equality** of the **two lines**, one representing ”low” and the other ”high”, (i.e. testing whether the gap between the lines is 0) is $\boldsymbol{t = 1.696}$ and the ***p-value*** is **0.1337**.

1.6.2 The Interaction Model and Test for Equal Slopes

- **Interaction** terms can be included in the model *formula* using the `:` or `*` symbol.

Here we fit the model **with** an **interaction** to the `y`, `trt`, and `x` vectors (from above):

```
my.reg <- lm(y ~ trt + x + trt:x)
```

Here are the results:

```
summary(my.reg)

##
## Call:
## lm(formula = y ~ trt + x + trt:x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9444 -0.6216  0.0405  0.6984  5.5238
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.7162     2.3232   4.613  0.00364
## trtlow        3.0378     4.0055   0.758  0.47694
## x             1.3108     0.8318   1.576  0.16614
## trtlow:x      0.2209     1.0480   0.211  0.84001
##
## Residual standard error: 3.2 on 6 degrees of freedom
## Multiple R-squared:  0.7771,Adjusted R-squared:  0.6656
## F-statistic: 6.971 on 3 and 6 DF,  p-value: 0.02212
```

From the output we conclude, among other things,

- The ***test statistic*** for testing for **equality** of the **two slopes**, one for the ”low” line and the other the ”high” line, is $\boldsymbol{t = 0.211}$ and the ***p-value*** is **0.84**.

- **ANCOVA models** with **more than one** qualitative predictor **and/or more than one** quantitative predictor can be fit as well.