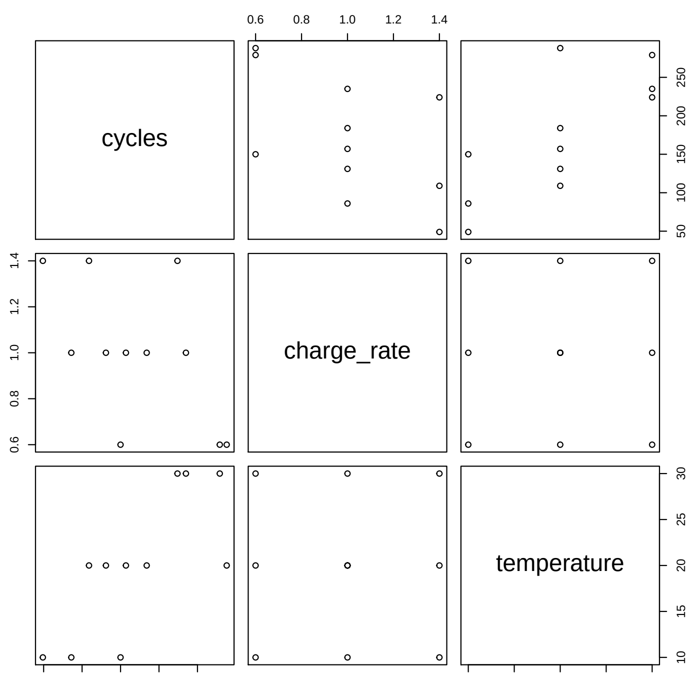
Stat 5100 Handout 3.1.1 - R: Alternative Predictor Variable Types

Stat 5100: Dr. Bean

Example 1: (Table 8.1) Study looks at the effects of the charge rate and temperature on the life of a new type of power cell. A small-scale preliminary study was conducted using 11 power cells. Variables reported are the charge rate (X_1, in amperes), the ambient temperature (X_2, in degrees Celsius), and the life of the power cell (Y, in the number of discharge-charge cycles before failure).

```
# Input data -- see Table 8.1 in text
library(stat5100)
data(powercells)
head(powercells)

##   cycles charge_rate temperature
## 1    150         0.6          10
## 2     86         1.0          10
## 3     49         1.4          10
## 4    288         0.6          20
## 5    157         1.0          20
## 6    131         1.0          20

# Create scatterplot matrix to see relationships with Y
pairs(~ cycles + charge_rate + temperature, data = powercells)
```



```r
# Define higher-order predictors
powercells <- cbind(powercells,
                    cr_temp = powercells$charge_rate * powercells$temperature,
                    cr2 = powercells$charge_rate^2,
                    temp2 = powercells$temperature^2)

# Create a regression model with an interaction term.
# NOTE: The following two lines are equivalent. The second line below is
# probably "better" in the sense that it is a more efficient R way to include
# an interaction term in a model.
powercells_int_lm <- lm(cycles ~ charge_rate + temperature + cr_temp,
                        data = powercells)
powercells_int_lm <- lm(cycles ~ charge_rate*temperature, data = powercells)

anova(powercells_int_lm)

## Analysis of Variance Table
##
## Response: cycles
##                         Df Sum Sq Mean Sq F value    Pr(>F)
## charge_rate              1  18704   18704 18.2573 0.0036877 **
## temperature              1  34201   34201 33.3844 0.0006787 ***
## charge_rate:temperature  1    529     529  0.5164 0.4956777
## Residuals                7   7171    1024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Create a regression model with all higher-order predictors
powercells_higher_lm <- lm(cycles ~ charge_rate + temperature + cr_temp +
                             cr2 + temp2, data = powercells)

# Test the null hypothesis that cr_temp = cr2 = temp2 = 0
# (This tests to see if there is any sort of higher-order interaction going
# on here)

# To test the above null hypothesis, we create a reduced model that is missing
# the above higher order predictors, then we call the ANOVA function with
# the two models to compare them.
powercells_reduced_lm <- lm(cycles ~ . -cr_temp -cr2 -temp2, data = powercells)
anova(powercells_higher_lm, powercells_reduced_lm)

## Analysis of Variance Table
##
## Model 1: cycles ~ charge_rate + temperature + cr_temp + cr2 + temp2
## Model 2: cycles ~ (charge_rate + temperature + cr_temp + cr2 + temp2) -
##     cr_temp - cr2 - temp2
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      5 5240.4
## 2      8 7700.3 -3   -2459.9 0.7823 0.5527

# Above we get a p-value of 0.4957 which tells us that we would fail to reject
# the null hypothesis that all higher-order interaction terms are 0
```

Look at higher-order variables, but standardize first

```r
# Standardize first
powercells_stdz <- data.frame(scale(powercells))
powercells_stdz$cr_temp <- powercells_stdz$charge_rate * powercells_stdz$temperature
powercells_stdz$cr2 <- powercells_stdz$charge_rate^2
powercells_stdz$temp2 <- powercells_stdz$temperature^2

# look for an interaction by looking at the ANOVA table
powercells_stdz_lm <- lm(cycles ~ charge_rate + temperature + cr_temp,
                         data = powercells_stdz)
anova(powercells_stdz_lm)

## Analysis of Variance Table
##
## Response: cycles
##             Df Sum Sq Mean Sq F value    Pr(>F)
## charge_rate  1 3.0862  3.0862 18.2573 0.0036877 **
## temperature  1 5.6433  5.6433 33.3844 0.0006787 ***
## cr_temp      1 0.0873  0.0873  0.5164 0.4956777
## Residuals    7 1.1833  0.1690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# (Our p-value checking for the interaction term above would be 0.4957)

# Check for the presence of higher-order predictor significance. Again,
# we accomplish this by creating a model that includes all terms and all higher
# order terms, and creating another model that does not have any higher order
# terms. We can then pass in the two models into the ANOVA function to test
# the null hypothesis that all the higher order coefficients are 0.
powercells_stdz_all_terms <- lm(cycles ~ ., data = powercells_stdz)
powercells_stdz_no_higher <- lm(cycles ~ . -cr2 -temp2 -cr_temp, data = powercells_stdz)
anova(powercells_stdz_all_terms, powercells_stdz_no_higher)

## Analysis of Variance Table
##
## Model 1: cycles ~ charge_rate + temperature + cr_temp + cr2 + temp2
## Model 2: cycles ~ (charge_rate + temperature + cr_temp + cr2 + temp2) -
##     cr2 - temp2 - cr_temp
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1      5 0.86467
## 2      8 1.27056 -3  -0.40588 0.7823 0.5527

# Above our p-value for the test would be 0.5527
```

Ending note: You don't need to standardize predictors to look at higher-order predictors like this. Instead, you can include a higher-order predictor and test it; if not significant, drop it; if significant, don't worry about significance of lower-order term. If higher-order term is significant and you really need to look at significance of lower-order term, or if the context of the data would allow the lower-order and higher-order terms to be 'stand-alone' interpretable, then standardize.

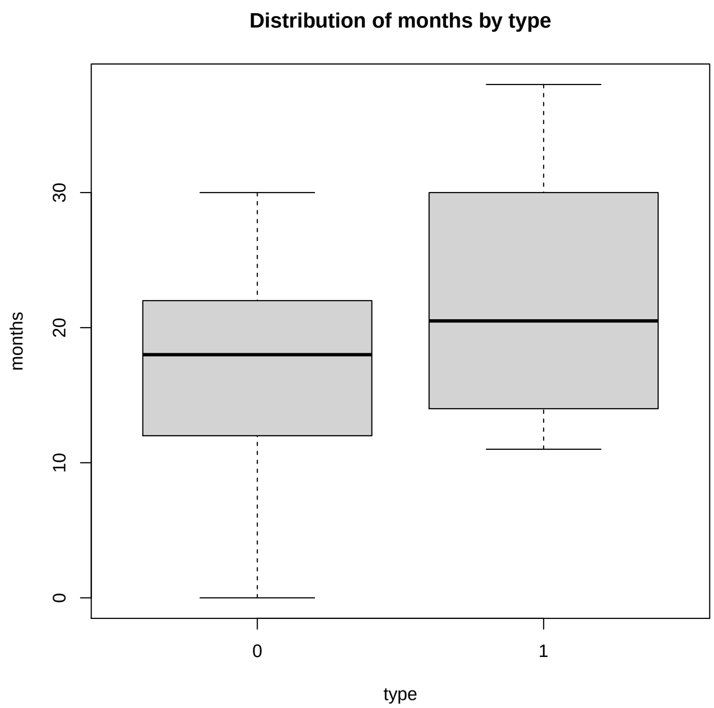
Tests for higher-order terms are the same whether data are standardized or not.

Example 2: An economist wishes to relate the speed with which a particular insurance innovation is adopted (Y, in months) to the size of the insurance firm (X_1, in millions of dollars) and the type of firm (X_2, either mutual (0) or stock firms (1)).

```
# Load the data
data(insurance)
head(insurance)

##   months size type
## 1     17  151    0
## 2     26   92    0
## 3     21  175    0
## 4     30   31    0
## 5     22  104    0
## 6      0  277    0

# Model with only the quantitative predictor
insurance_lm_quant <- lm(months ~ size, data = insurance)
summary(insurance_lm_quant)

##
## Call:
## lm(formula = months ~ size, data = insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4621  -4.7236   0.7912   4.3427   7.9055
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.48211    2.84425  12.827 1.71e-10 ***
## size        -0.09394    0.01426  -6.589 3.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.231 on 18 degrees of freedom
## Multiple R-squared:  0.7069,Adjusted R-squared:  0.6906
## F-statistic: 43.41 on 1 and 18 DF,  p-value: 3.452e-06

# Model with only the qualitative predictor
insurance_lm_qual <- lm(months ~ type, data = insurance)
summary(insurance_lm_qual)

##
## Call:
## lm(formula = months ~ type, data = insurance)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -16.70  -7.35  -0.20   6.40  15.90
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    16.70       2.92   5.719 2.02e-05 ***
## type            5.40       4.13   1.308    0.207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 9.235 on 18 degrees of freedom
## Multiple R-squared:  0.08674,Adjusted R-squared:  0.03601
## F-statistic:  1.71 on 1 and 18 DF,  p-value: 0.2075
```

```r
# Create a boxplot of the variable "months" by the two different types
boxplot(months ~ type, data = insurance,
        main = "Distribution of months by type")
```

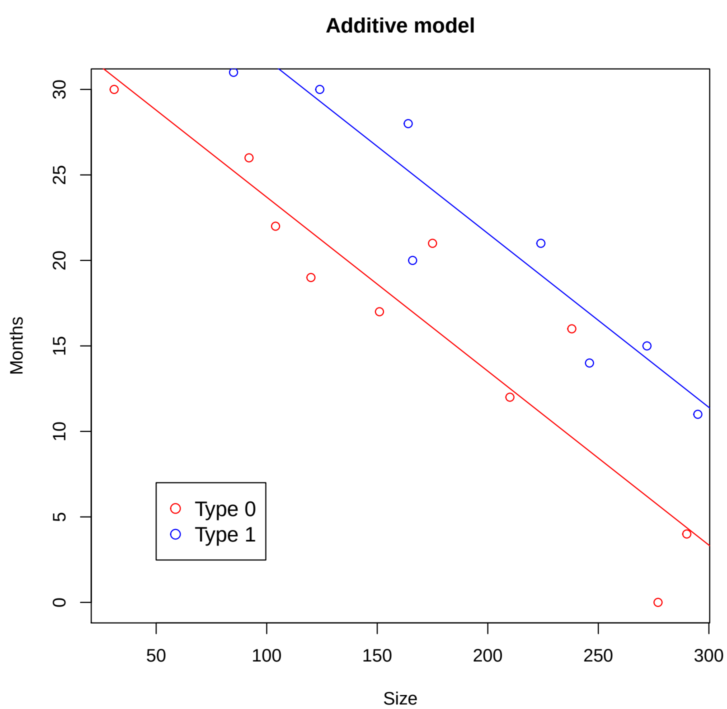


Create a linear model (no interaction present):

```r
# Create the additive model with both predictor types
insurance_lm <- lm(months ~ size + type, data = insurance)
summary(insurance_lm)
```

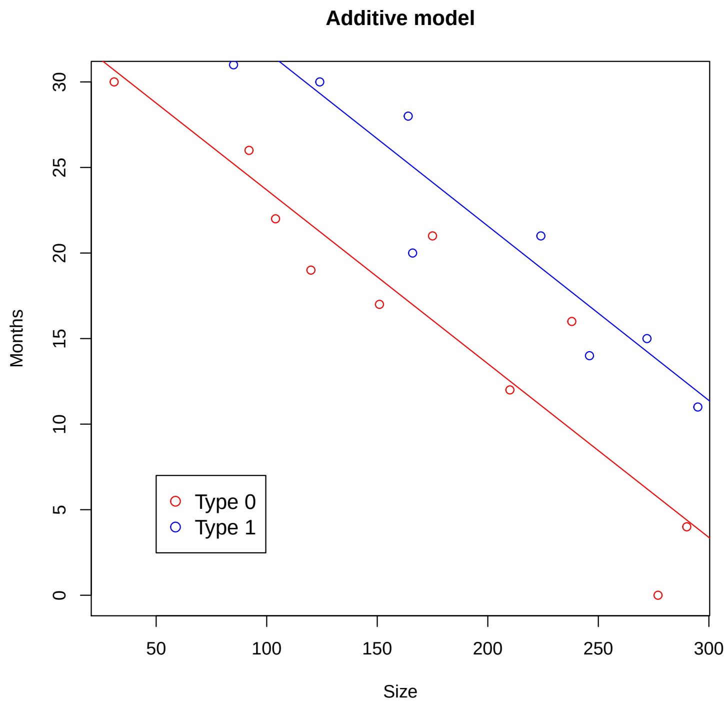
```
##
## Call:
## lm(formula = months ~ size + type, data = insurance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6915 -1.7036 -0.4385  1.9210  6.3406
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.874069   1.813858  18.675 9.15e-13 ***
## size        -0.101742   0.008891 -11.443 2.07e-09 ***
## type         8.055469   1.459106   5.521 3.74e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.221 on 17 degrees of freedom
## Multiple R-squared:  0.8951,Adjusted R-squared:  0.8827
## F-statistic:  72.5 on 2 and 17 DF,  p-value: 4.765e-09

# Create a fit plot individually for each of the two type levels. This isn't
# something that's natively supported in R or the stat5100 package, so we
# will have to grab the coefficients manually from the linear model.

# Each of these vectors below contain (intercept, slope). Note that in the
# type1, we find the intercept by adding the estimate of "type" onto the
# existing intercept
type0_coeff <- c(33.874069, -0.101742)
type1_coeff <- c(33.874069 + 8.055469, -0.101742)

# Type 0
type0 <- insurance[insurance$type == 0, ]
plot(type0$size, type0$months, col = "red", main = "Additive model",
     xlab = "Size", ylab = "Months")
abline(a = type0_coeff[1], b = type0_coeff[2], col = "red")

# Type 1
type1 <- insurance[insurance$type == 1, ]
points(type1$size, type1$months, col = "blue")
abline(a = type1_coeff[1], b = type1_coeff[2], col = "blue")

# Add a legend
legend(x = 50, y = 7, c("Type 0","Type 1"), cex = 1.2,
       col = c("red", "blue"), pch = c(1,1))
```



Create an interaction model:

```r
# Create an interaction model
insurance_int_lm <- lm(months ~ size*type, data = insurance)
summary(insurance_int_lm)

##
## Call:
## lm(formula = months ~ size * type, data = insurance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7144 -1.7064 -0.4557  1.9311  6.3259
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.8383695  2.4406498  13.864 2.47e-10 ***
## size        -0.1015306  0.0130525  -7.779 7.97e-07 ***
## type         8.1312501  3.6540517   2.225   0.0408 *
## size:type   -0.0004171  0.0183312  -0.023   0.9821
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.32 on 16 degrees of freedom
## Multiple R-squared:  0.8951,Adjusted R-squared:  0.8754
## F-statistic: 45.49 on 3 and 16 DF,  p-value: 4.675e-08

# To include the interaction in a similar plot above, we change the
# slope of the type 1 line to be the sum of the estimate for type and the
# estimate for the interaction coefficient.
type0_coeff <- c(33.8383695, -0.1015306)
type1_coeff <- c(33.8383695 + 8.1312501, -0.1015306 - 0.0004171)

# Type 0
type0 <- insurance[insurance$type == 0, ]
plot(type0$size, type0$months, col = "red", main = "Additive model",
     xlab = "Size", ylab = "Months")
abline(a = type0_coeff[1], b = type0_coeff[2], col = "red")

# Type 1
type1 <- insurance[insurance$type == 1, ]
points(type1$size, type1$months, col = "blue")
abline(a = type1_coeff[1], b = type1_coeff[2], col = "blue")

# Add a legend
legend(x = 50, y = 7, c("Type 0","Type 1"), cex = 1.2,
       col = c("red", "blue"), pch = c(1,1))
```



Additive model
Months
Size
0
5
10
15
20
25
30
50
100
150
200
250
300
Type 0
Type 1