Problem Set 8

Shane T. Mueller shanem@mtu.edu

October 23, 2018

1. Orthogonal Designs

In an new experiment, you have two 3-level independent variables: * training (no training, standard training, interactive training) * feedback (no feedback, feedback on errors, feedback on correct) You have create the following orthogonal design:

```
training <- rep(c(1,1,1,0,0,0,-1,-1,-1),each=3)
feedback <- rep(c(1,0,-1,1,0,-1,1,0,-1),each=3)
```

You can verify that they are orthogonal by finding their dot product, and that they are uncorrelated by finding the correlation:

```
cor(training, feedback)
```

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

```
training %*% feedback
```

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

(a) Suppose you want to add a third 3-level IV (with values -1, 0, and 1) to this design called group. Create one that is orthogonal to the two existing IVs, and verify that it is orthogonal. To do this, look carefully at the design; you will notice that each level of training has a level of feedback, but there are three identical replications of this design. Which of the following grouping variables are both orthogonal and uncorrelated with both feedback and train

```
set.seed(100)
group1 <- 1:27
group2 <- rep(-1:1, 9)
group3 <- 1:27-mean(1:27)
group4 <- rep(1:3, 9)
group5 <- rep(-1:1,each=9)
group6 <- runif(27)-.5
group7 <- rep(1:9,each=3)
```

2. Regression with orthogonal and uncorrelated predictors.

Suppose we have a data set in which we have three records from a bunch of buyers. The first is their initial purchase on our website, the second is their purchase during a buy-one get-one promotion, and the third is their most recent purchase. We also have recorded their age, as we suspect older customers buy more.

```
data <- data.frame(buyerid = group7,
                   timeframe=group4)

set.seed(103)

buyerbase <- runif(9)
```

```
timebase <- c(1,2,2.5)
agebase  <- 20+runif(9)* 40

data$age <- round(agebase[data$buyerid])
data$purchase <- round( 25 + buyerbase[data$buyerid] * 50 +
                          data$age * .5+
                          timebase[data$timeframe] * 7 +
                          rnorm(27)*4,2)
```

```
data <- read.table(text=
"row  buyerid timeframe age purchase
1        1         1  33    57.80
2        1         2  33    62.57
3        1         3  33    69.93
4        2         1  38    54.65
5        2         2  38    71.48
6        2         3  38    63.58
7        3         1  31    70.04
8        3         2  31    82.43
9        3         3  31    84.55
10       4         1  52    82.26
11       4         2  52    93.35
12       4         3  52    97.40
13       5         1  53    65.57
14       5         2  53    70.54
15       5         3  53    69.12
16       6         1  55    59.10
17       6         2  55    73.83
18       6         3  55    72.33
19       7         1  25    70.92
20       7         2  25    76.99
21       7         3  25    77.22
22       8         1  26    50.54
23       8         2  26    64.44
24       8         3  26    67.53
25       9         1  31    50.79
26       9         2  31    56.01
27       9         3  31    57.03",header=T)
```

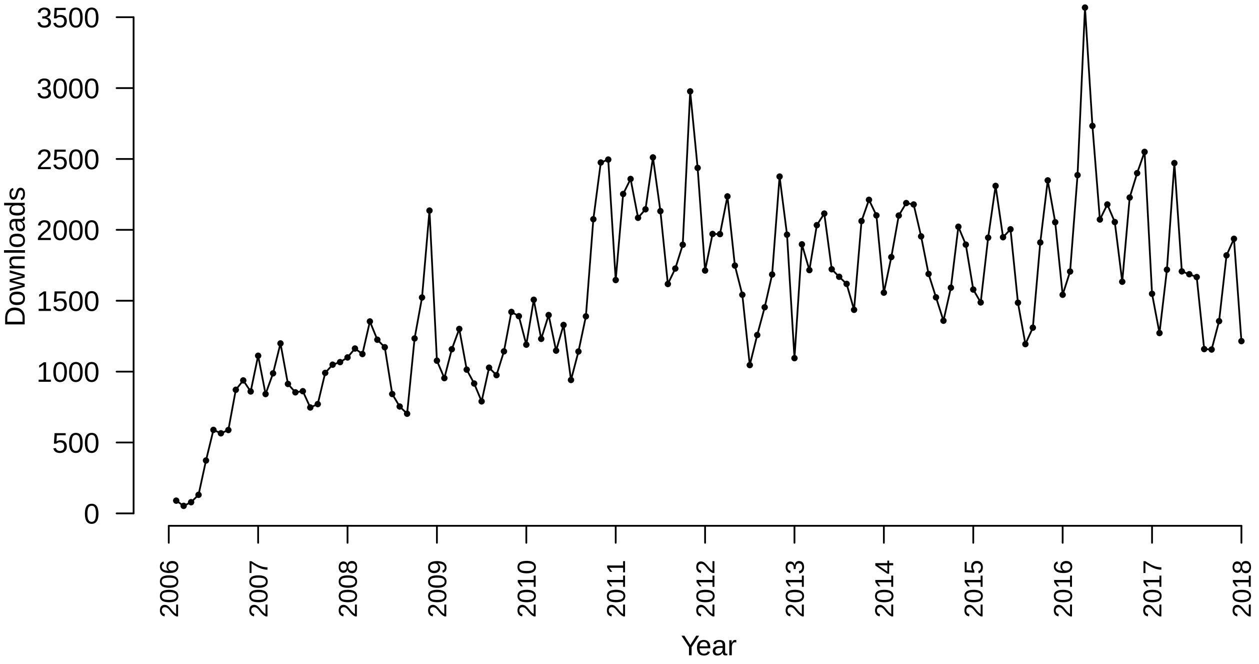
Create a regression model to predict purchase amount by timeframe and age, and compare this model to the models using only timeframe and only age as predictors. Do the coefficients differ across models (including intercept)? Then, transform the predictors so that they are also orthogonal. Examine the same models with the orthogal predictors. Now, do the coefficient differ across models? For the second set of models, give an explanation of how to interpret each coefficient of the orthogonal predictors, and especially explain how it differs from your interpretation for the original predictors. Finally, suppose that the first observation was missing, (you can specify data2 <- data[-1,] to remove that point, and then use data2 in the model). Look at the same models again. Do they produce the same estimates? Why?

3. Modeling a time series.

The following data show download records for a piece of software over roughly a 10-year period.

```
down <- read.csv("DownloadData.csv")
down$Month <- as.factor(substr(down$Date,6,8))
down$MonthNumber <- 1:nrow(down)

plot(as.numeric(down$Date),down$Downloads,xaxt="n",bty="n",pch=16,cex=.5,type="o",las=1,
     ylab="Downloads",xlab="Year")
axis(1,0:12*12,2006:2018,las=3,cex.axis=.95)
```



The pattern involves both growth over time and cyclic variations across the year.

A. Loess regression.

Fit a loess model to the down, using the same predictor. Select a span parameter that is reasonable, and compute Rˆ2 (observed versus fitted) and RSE.

B. Polynomial regression.

Create a polynomial regression of the down, using just MonthNumber as a predictor within poly(). Use either an information metric (AIC or BIC) or series of F-tests to determine your preferred model. Plot the down as well as the prediction on the same graph, and describe how good the model fits in overall terms (using Rˆ2, RMSE, or similar measures). Compare to the model in Part 1 with respect to Rˆ2 and RMSE, as well as the equivalent number of parameters. Does one seem better than the other? Why?

C. Adding month predictor

We know that there are cyclic patterns across the year related to summer/winter. To address this, add down$Month as a predictor to a polynomial model (Also add the +0 intercept so each month has a unique value). Be sure Month is a categorical variable/factor, so it does not get fitted with a linear trend. Try reducing the polynomial predictor coefficient until you have a model that fits equivalently well to the ones in part A and B (using RMSE and Rˆ2 criteria). Compare the number of parameters total in this model to the previous two models, and discuss whether this model is better than the models in Parts 1 and 2. Finally, show the average effect across the year by plotting the monthly parameters alone.