Response Surface Methodology, Stat 579	Homework 05	Name: Answer Key
Fall 2014		Prof. Erik B. Erhardt

Part I. (50 points) I recommend reading through all the parts of the HW (with my adjustments) before starting; this may save you some work.
MMA-RSM Chapter 5: 5.5, 5.7.

- For part 5.5(a), fit the first order model and use the results in parts (b) and (c). However, I think if you were doing this for real, fitting the first order model, or even the first order model with two-way interactions, is a bad idea. Do you agree with me? Back up your answer with some analysis.
- For part 5.5(b), use all main effects and make one unit steps.
- For part 5.5(c), the constraint makes no sense to me. Tell why. Instead of the given constraint, use this one: $x_1 + x_2 = -2.7$ (remember, these are coded units).

General: Try to do all calculations in R. All R code for the assignment should be included with the part of the problem it addresses (for code and output use a fixed-width font, such as Courier). Code is used to calculate result; text is used to report and interpret results – do not report or interpret results in the code.

(35^{pts}) **1. 5.5** In a metallurgy experiment it is desired to test the effect of four factors and their interactions on the concentration (percent by weight) of a particular phosphorus compound in costing material. The variables are: A, percent phosphorus in the refinement; B, percent remelted material; C, fluxing time, and D, holding time. The four factors are varied in a 24 factorial experiment with two castings taken at each factor combination. The 32 castings were made in random order, and the data are shown in Table E5.1.

Fit the first order model and use the results in parts (b) and (c). However, I think if you were doing this for real, fitting the first order model, or even the first order model with two-way interactions, is a bad idea. Do you agree with me? Back up your answer with some analysis.

(a) (15 pts) Build a first-order response function.

Solution: Read data.

```
#### 5.5
fn.data <- "http://statacumen.com/teach/RSM/data/RSM_HW_05-05.txt"
df.5.5 <- read.table(fn.data, header=TRUE)
str(df.5.5)

## 'data.frame': 16 obs. of  6 variables:
##  $ a : int  -1 1 -1 1 -1 1 -1 1 -1 1 ...
##  $ b : int  -1 -1 1 1 -1 -1 1 1 -1 -1 ...
##  $ c : int  -1 -1 -1 -1 1 1 1 1 -1 -1 ...
##  $ d : int  -1 -1 -1 -1 -1 -1 -1 -1 1 1 ...
##  $ y1: num  30.3 28.5 24.5 25.9 24.8 26.9 24.8 22.2 31.7 24.6 ...
##  $ y2: num  28.6 31.4 25.6 27.2 23.4 23.8 27.8 24.9 33.5 26.2 ...

# reshape data into long format
library(reshape2)
df.5.5 <- melt(df.5.5, id.vars = c("a", "b", "c", "d"), variable.name = "rep", value.name = "y")
str(df.5.5)

## 'data.frame': 32 obs. of  6 variables:
##  $ a  : int  -1 1 -1 1 -1 1 -1 1 -1 1 ...
##  $ b  : int  -1 -1 1 1 -1 -1 1 1 -1 -1 ...
##  $ c  : int  -1 -1 -1 -1 1 1 1 1 -1 -1 ...
##  $ d  : int  -1 -1 -1 -1 -1 -1 -1 -1 1 1 ...
##  $ rep: Factor w/ 2 levels "y1","y2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ y  : num  30.3 28.5 24.5 25.9 24.8 26.9 24.8 22.2 31.7 24.6 ...
```

Interaction model:
Fit first-order with four-way interaction linear model.

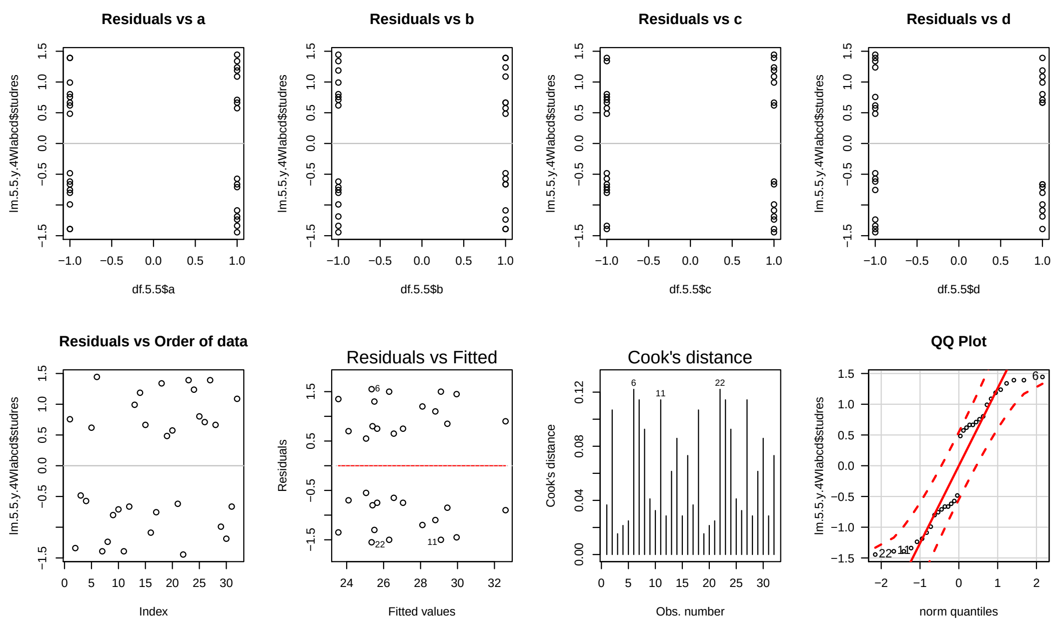
```
lm.5.5.y.4WIabcd <- lm(y ~ (a + b + c + d)^4, data = df.5.5)
# externally Studentized residuals
lm.5.5.y.4WIabcd$studres <- rstudent(lm.5.5.y.4WIabcd)
summary(lm.5.5.y.4WIabcd)

##
```

```
## Call:
## lm(formula = y ~ (a + b + c + d)^4, data = df.5.5)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1.55  -0.95   0.00   0.95   1.55
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   27.031      0.277   97.49   <2e-16 ***
## a             -0.600      0.277   -2.16   0.0459 *
## b             -0.612      0.277   -2.21   0.0421 *
## c             -1.113      0.277   -4.01   0.0010 **
## d              0.744      0.277    2.68   0.0163 *
## a:b            0.494      0.277    1.78   0.0939 .
## a:c            0.306      0.277    1.10   0.2857
## a:d           -0.662      0.277   -2.39   0.0295 *
## b:c            0.594      0.277    2.14   0.0480 *
## b:d            0.312      0.277    1.13   0.2763
## c:d            0.350      0.277    1.26   0.2249
## a:b:c         -0.275      0.277   -0.99   0.3360
## a:b:d          0.869      0.277    3.13   0.0064 **
## a:c:d          0.744      0.277    2.68   0.0163 *
## b:c:d         -0.431      0.277   -1.56   0.1394
## a:b:c:d        0.350      0.277    1.26   0.2249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.57 on 16 degrees of freedom
## Multiple R-squared:  0.819,Adjusted R-squared:  0.649
## F-statistic: 4.83 on 15 and 16 DF,  p-value: 0.00165
```

The full 2^4 design with two replicates has enough df to estimate all main effects and interactions, so we fit a full model. In a full interaction model, we find all main effects A, B, C, D and interactions AD, BC, ABD, ACD to be significant. We would have overlooked significant effects if we had performed only the first-order model.
The plot below indicates the residuals are not normal, they are bimodal with no values near zero. This is because we are estimating all effects and the fitted value is the mean of two replicates, where half the distance between replicates is the $+$ and $-$ residual. This is not a problem in the model with only main effects.

```
# plot diagnostics
par(mfrow=c(2,4))

plot(df.5.5$a, lm.5.5.y.4WIabcd$studres, main="Residuals vs a")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(df.5.5$b, lm.5.5.y.4WIabcd$studres, main="Residuals vs b")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(df.5.5$c, lm.5.5.y.4WIabcd$studres, main="Residuals vs c")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(df.5.5$d, lm.5.5.y.4WIabcd$studres, main="Residuals vs d")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# residuals vs order of data
plot(lm.5.5.y.4WIabcd$studres, main="Residuals vs Order of data")
  # horizontal line at zero
  abline(h = 0, col = "gray75")
```

```
plot(lm.5.5.y.4WIabcd, which = c(1,4))

# Normality of Residuals
library(car)
qqPlot(lm.5.5.y.4WIabcd$studres, las = 1, id.n = 3, main="QQ Plot")

##  6 22 11
## 32  1  2

cooks.distance(lm.5.5.y.4WIabcd)

##       1       2       3       4       5       6       7       8       9      10      11
## 0.03671 0.10683 0.01537 0.02147 0.02490 0.12208 0.11433 0.09261 0.04116 0.03252 0.11433
##      12      13      14      15      16      17      18      19      20      21      22
## 0.02858 0.06148 0.08587 0.02858 0.07317 0.03671 0.10683 0.01537 0.02147 0.02490 0.12208
##      23      24      25      26      27      28      29      30      31      32
## 0.11433 0.09261 0.04116 0.03252 0.11433 0.02858 0.06148 0.08587 0.02858 0.07317
```



Main-effects model:

Note, that the model fit with only main effects below has only C as significant.
Fit first-order linear model.

```
library(rsm)
rsm.5.5.y.FOabcd <- rsm(y ~ FO(a, b, c, d), data = df.5.5)
# externally Studentized residuals
rsm.5.5.y.FOabcd$studres <- rstudent(rsm.5.5.y.FOabcd)
summary(rsm.5.5.y.FOabcd)

##
## Call:
## rsm(formula = y ~ FO(a, b, c, d), data = df.5.5)
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   27.031      0.398   67.96   <2e-16 ***
## a             -0.600      0.398   -1.51   0.1430
## b             -0.612      0.398   -1.54   0.1352
## c             -1.113      0.398   -2.80   0.0094 **
## d              0.744      0.398    1.87   0.0724 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.372,Adjusted R-squared:  0.279
## F-statistic: 3.99 on 4 and 27 DF,  p-value: 0.0114
##
```

```
## Analysis of Variance Table
##
## Response: y
##                Df Sum Sq Mean Sq F value Pr(>F)
## FO(a, b, c, d)  4   80.8   20.21    3.99  0.011
## Residuals      27  136.7    5.06
## Lack of fit    11   97.3    8.85    3.60  0.010
## Pure error     16   39.4    2.46
##
## Direction of steepest ascent (at radius 1):
##       a       b       c       d
## -0.3775 -0.3854 -0.7000  0.4680
##
## Corresponding increment in original units:
##       a       b       c       d
## -0.3775 -0.3854 -0.7000  0.4680
```

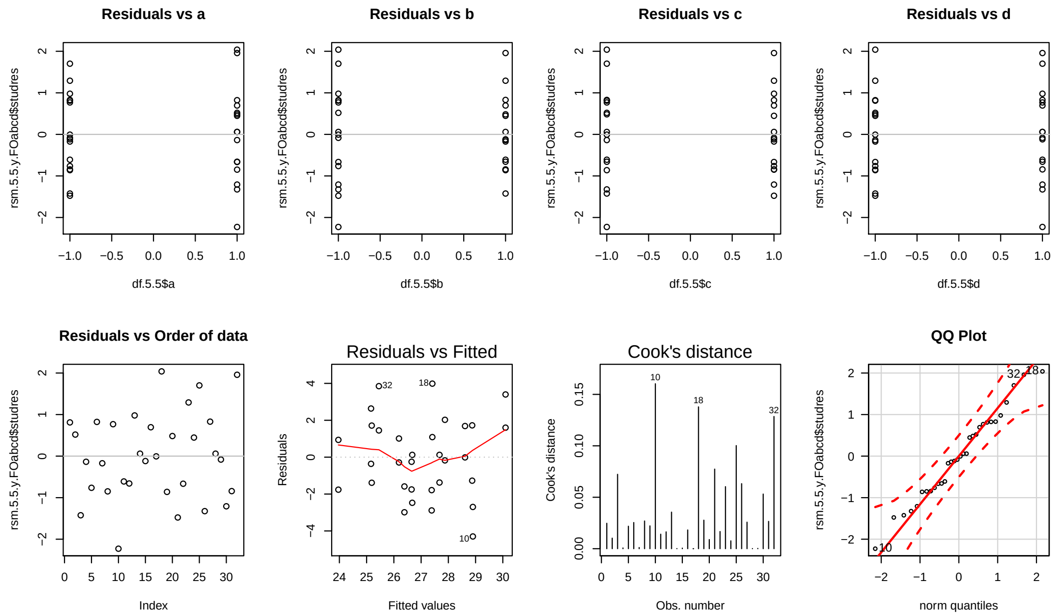
In the figure below the residuals do not give us any cause for concern. None of the Cook's D values appear large enough to investigate.

```
# plot diagnistics
par(mfrow=c(2,4))

plot(df.5.5$a, rsm.5.5.y.FOabcd$studres, main="Residuals vs a")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(df.5.5$b, rsm.5.5.y.FOabcd$studres, main="Residuals vs b")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(df.5.5$c, rsm.5.5.y.FOabcd$studres, main="Residuals vs c")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(df.5.5$d, rsm.5.5.y.FOabcd$studres, main="Residuals vs d")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# residuals vs order of data
plot(rsm.5.5.y.FOabcd$studres, main="Residuals vs Order of data")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(rsm.5.5.y.FOabcd, which = c(1,4))

# Normality of Residuals
library(car)
qqPlot(rsm.5.5.y.FOabcd$studres, las = 1, id.n = 3, main="QQ Plot")
```

```
## 10 18 32
##  1 32 31
```

```
cooks.distance(rsm.5.5.y.FOabcd)
```

```
##         1         2         3         4         5         6         7         8         9
## 2.469e-02 1.026e-02 7.230e-02 7.167e-04 2.185e-02 2.543e-02 1.139e-03 2.694e-02 2.220e-02
##        10        11        12        13        14        15        16        17        18
## 1.603e-01 1.410e-02 1.639e-02 3.556e-02 1.355e-04 5.420e-04 1.823e-02 1.355e-06 1.379e-01
##        19        20        21        22        23        24        25        26        27
## 2.771e-02 8.890e-03 7.739e-02 1.669e-02 6.032e-02 7.621e-03 1.002e-01 6.321e-02 2.580e-02
##        28        29        30        31        32
## 1.355e-04 2.656e-04 5.312e-02 2.656e-02 1.285e-01
```



(b) (5 pts) Construct a table of the path of steepest ascent in the coded design variables. Use all main effects and make one unit steps.

Solution: The first-order model is

$$y = 27.03125 - 0.60000a - 0.61250b - 1.11250c + 0.74375d.$$

The default step size is

$$r = \sqrt{0.60000^2 + 0.61250^2 + 1.11250^2 + 0.74375^2} = 1.589332.$$

Thus, to take one-unit steps, the steps satisfy (note: $1/r = 0.6291951$)

$$1 = \sqrt{(-0.60000/r)^2 + (-0.61250/r)^2 + (-1.11250/r)^2 + (0.74375/r)^2}.$$

Therefore, the unit steps along the path of steepest ascent are given in the table below.

	Coded units			
	a	b	c	d
Base	0	0	0	0
Increment = Δ	−0.377517	−0.385382	−0.6999796	0.467964
Base + 1Δ	−0.377517	−0.385382	−0.699980	0.467964
Base + 2Δ	−0.755034	−0.770764	−1.399959	0.935928
Base + 3Δ	−1.132551	−1.156146	−2.099939	1.403892
Base + 4Δ	−1.510068	−1.541528	−2.799918	1.871855
Base + 5Δ	−1.887585	−1.926910	−3.499898	2.339819
Base + 6Δ	−2.265102	−2.312292	−4.199877	2.807783
Base + 7Δ	−2.642619	−2.697674	−4.899857	3.275747

Because we fit the first-order model with `rsm()`, the direction of steepest ascent is given to us and can be computed in unit increments using `steepest()`.

```
summary(rsm.5.5.y.FOabcd)$sa
##       a       b       c       d
## -0.3775 -0.3854 -0.7000  0.4680
steepest.5.5 <- steepest(rsm.5.5.y.FOabcd, dist = seq(0, 7, by = 1))
## Path of steepest ascent from ridge analysis:
```

	dist	a	b	c	d	yhat
1	0.000	0.000	0.000	0.000	0.000	27.031
2	1.000	−0.378	−0.385	−0.700	0.468	28.621
3	2.000	−0.755	−0.771	−1.400	0.936	30.210
4	3.000	−1.132	−1.156	−2.100	1.404	31.799
5	4.000	−1.510	−1.542	−2.800	1.872	33.389
6	5.000	−1.888	−1.927	−3.500	2.340	34.978
7	6.000	−2.265	−2.312	−4.200	2.808	36.567
8	7.000	−2.643	−2.698	−4.900	3.276	38.157

(c) (15 pts) It is important to constrain the percentage of phosphorus and the percentage of remelted material. In fact, in the metric of the coded variables we obtain $x_1 + x_2 = 2.7$, where x_1 is percent phosphorus and x_2 is percent remelted material. Recalculate the path of steepest ascent subject to the above constraint.
The constraint makes no sense to me. Tell why. Instead of the given constraint, use this one: $x_1 + x_2 = -2.7$ (remember, these are coded units).

Solution: Reference section 5.5. The original constraint makes no sense because it is "down hill" from the origin, so our path of steepest ascent does not intercept the constraint. Rewrite our constraint as $c_0 + c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 = 0$ to give $2.7 + 1x_1 + 1x_2 + 0 + 0 = 0$.
Note that Δ is a scaled version of $\underset{\sim}{\hat{\beta}}$, thus we may use Δ instead for all the calculations. We intercept the contraint in this many units from the origin,

$$\begin{aligned}
\rho_0 &= -c_0/(\underset{\sim}{c}^\top \underset{\sim}{\hat{\beta}}) = -2.7/(-0.60000(1) - 0.61250(1) - 1.11250(0) + 0.74375(0)) \\
&= 2.226804 \qquad \underset{\sim}{\hat{\beta}} \text{ units from the origin, or} \\
\rho_{0\Delta} &= -c_0/(\underset{\sim}{c}^\top \underset{\sim}{\Delta}) = -2.7/(-0.377517(1) - 0.385382(1) - 0.699980(0) + 0.467964(0)) \\
&= 3.539132 \quad \Delta \text{ unit steps from the origin.}
\end{aligned}$$

Thus, the point of interception is

$$\rho_0\underset{\sim}{\hat{\beta}} = \rho_{0\Delta}\Delta = (-1.336083, -1.363918, -2.477320, 1.656186).$$

When we intercept the constraint, we use Lagrange multipliers to calculate our modified path. Calculate the path modifier constant, d_0,

$$\begin{aligned}
d_0 &= (\underset{\sim}{c}^\top \underset{\sim}{\hat{\beta}})/(\underset{\sim}{c}^\top \underset{\sim}{c}) \\
&= (-0.60000(1) - 0.61250(1) - 1.11250(0) + 0.74375(0))/(1^2 + 1^2 + 0^2 + 0^2) \\
&= -0.60625 \qquad \underset{\sim}{\hat{\beta}} \text{ units, or} \\
d_{0\Delta} &= (\underset{\sim}{c}^\top \underset{\sim}{\Delta})/(\underset{\sim}{c}^\top \underset{\sim}{c}) \\
&= (-0.377517(1) - 0.385382(1) - 0.699980(0) + 0.467964(0))/(1^2 + 1^2 + 0^2 + 0^2) \\
&= -0.3814495 \quad \Delta \text{ units.}
\end{aligned}$$

The modified path is given by (using Δ units to continue the steps in our table),

$$\begin{aligned}
a &= -1.336083 + +\lambda(-0.377517 - -0.3814495(1)) = -1.336083 + 0.0039325\lambda \\
b &= -1.363918 + \lambda(-0.385382 - -0.3814495(1)) = -1.363918 - 0.0039325\lambda \\
c &= -2.477320 + \lambda(-0.699980 - -0.3814495(0)) = -2.477320 - 0.699980\lambda \\
d &= 1.656186 + \lambda(+0.467964 - -0.3814495(0)) = 1.656186 + 0.467964\lambda.
\end{aligned}$$

That is, $\Delta_2 = (0.0039325, -0.0039325, -0.699980, 0.467964)$. To continue with step size of one unit, let

$$\lambda = 1.187624 = 1/\sqrt{0.0039325^2 + (-0.0039325)^2 + (-0.699980)^2 + 0.467964^2}.$$

Therefore, path is modified as given in the table below.

	Coded units			
	a	b	c	d
Base	0	0	0	0
Increment = Δ	−0.377517	−0.385382	−0.6999796	0.467964
Base + 1Δ	−0.377517	−0.385382	−0.699980	0.467964
Base + 2Δ	−0.755034	−0.770764	−1.399959	0.935928
Base + 3Δ	−1.132551	−1.156146	−2.099939	1.403892
Modified path				
Increment = $\lambda\Delta_2$	0.004670	−0.004670	−0.831313	0.555765
Base + $\rho_{0\Delta}\Delta$	−1.336082	−1.363918	−2.477320	1.656186
Base + $\rho_{0\Delta}\Delta + 1\lambda\Delta_2$	−1.331412	−1.368588	−3.308633	2.211951
Base + $\rho_{0\Delta}\Delta + 2\lambda\Delta_2$	−1.326742	−1.373258	−4.139946	2.767716
Base + $\rho_{0\Delta}\Delta + 3\lambda\Delta_2$	−1.322071	−1.377929	−4.971259	3.323481
Base + $\rho_{0\Delta}\Delta + 4\lambda\Delta_2$	−1.317401	−1.382599	−5.802572	3.879247

```
## contrained path of steepest ascent
c.constraint <- c(1,1,0,0);

# Delta scale
Delta <- summary(rsm.5.5.y.FOabcd)$sa
rho0.Delta = -2.7 / sum(Delta * c.constraint)
rho0.Delta

## [1] 3.539

constraint.intercept <- rho0.Delta * Delta
constraint.intercept

##      a      b      c      d
## -1.336 -1.364 -2.477  1.656

# path modified constant
d0.Delta <- sum(Delta * c.constraint) / sum(c.constraint^2)
d0.Delta

## [1] -0.3814

# modified path
Delta2 <- Delta - d0.Delta * c.constraint
step.Delta2 <- 1/sqrt(sum(Delta2^2))  # lambda
step.Delta2

## [1] 1.188

step.Delta2 * Delta2

##        a        b        c        d
##  0.00467 -0.00467 -0.83131  0.55577

# 5 steps along path of steepest ascent
n.step <- 5
modified.path <- matrix(rep(constraint.intercept, n.step), nrow = n.step, byrow = TRUE) +
                   matrix(seq(0, (n.step - 1) * step.Delta2, length = n.step), ncol=1) %*% matrix(Delta2, nrow = 1)
colnames(modified.path) <- c("a", "b", "c", "d")
```

	a	b	c	d
1	−1.336	−1.364	−2.477	1.656
2	−1.331	−1.369	−3.309	2.212
3	−1.327	−1.373	−4.140	2.768
4	−1.322	−1.378	−4.971	3.323
5	−1.317	−1.383	−5.803	3.879

(15[pts]) **2. 5.7** It is stated in the text that the development of the path of steepest ascent makes use of the assumption that the model is truly first-order in nature. However, even if there is a modest amount of curvature or interaction in the system, the use of steepest ascent can be extremely useful in determining a future experimental region. Suppose that in a system involving x_1 and x_2 the actual model is given by $\mathrm{E}(y) = 14 + 5x_1 - 10x_2 + 3x_1x_2$. Assume that x_1 and x_2 are in coded form.

(a) (5 pts) Show a plot of the path of steepest ascent (based on actual parameters) if the interaction is ignored.

Solution: $\mathrm{E}(y) = 14 + 5x_1 - 10x_2 + 3x_1x_2$
The default step size is

$$r = \sqrt{5^2 + 10^2} = 125.$$

Thus, to take one-unit steps, the steps satisfy (note: $1/r = 0.008$)

$$1 = \sqrt{(5/r)^2 + (-10/r)^2}.$$

Therefore, the unit steps along the path of steepest ascent are given by Table 1.

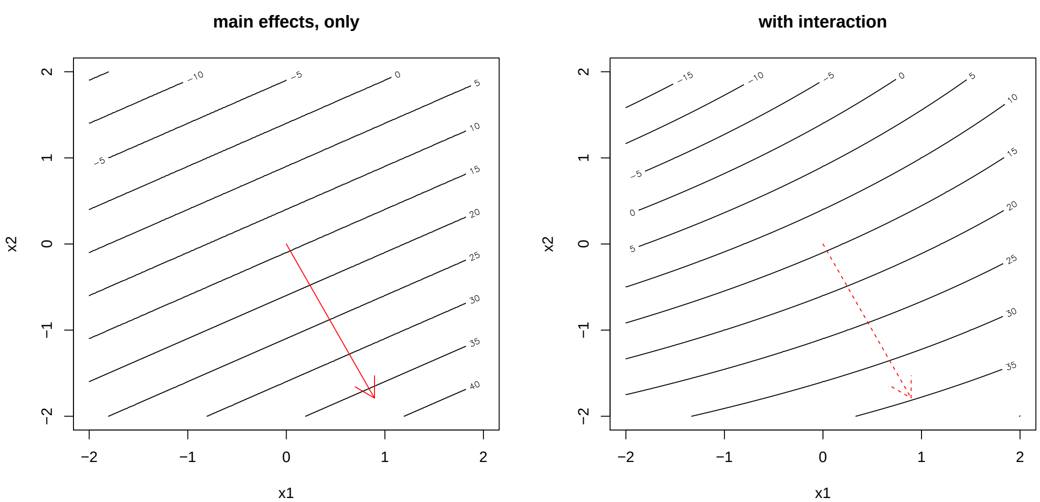
	Coded units	
	x_1	x_2
Base	0	0
Increment = Δ	0.4472136	-0.8944272
Base + 1Δ	0.4472136	-0.8944272
Base + 2Δ	0.8944272	-1.788854
Base + 3Δ	1.341641	-2.683282
Base + 4Δ	1.788854	-3.577709
Base + 5Δ	2.236068	-4.472136
Base + 6Δ	2.683282	-5.366563
Base + 7Δ	3.130495	-6.26099

Table 1: 57a

The plot below shows the path of steepest ascent over the first-order model and with interaction. Note, the arrow is the same on both plots, based on the main effects model, since it's tricky to draw the curved line (which does not have much curvature, anyway).

```
#### 5.7
# radius of default step length
r = sqrt(5^2+10^2)
# Delta step for unit length
D = c(5,-10)/r
# 7 steps along path of steepest ascent
for (i in 0:7) cat(i*D,"\n")

## 0 0
## 0.4472 -0.8944
## 0.8944 -1.789
## 1.342 -2.683
## 1.789 -3.578
## 2.236 -4.472
## 2.683 -5.367
## 3.13 -6.261

# functions for calculating expected response for main-effects and interaction model
f.yhat1 <- function (x) { # main effects, only
  yhat <- 14 + 5 * x[1] - 10 * x[2]
  return(yhat);
}
f.yhat2 <- function (x) { # with cross-product
```

```
  yhat <- 14 + 5 * x[1] - 10 * x[2] + x[1] * x[2]
  return(yhat);
}

# evaluate points over a square to create contour plot
x1 <- seq(-2,2,.01);
x2 <- seq(-2,2,.01);
par(mfrow = c(1, 2))
y1 <- matrix(NA, ncol=length(x1), nrow=length(x2));
y2 <- matrix(NA, ncol=length(x1), nrow=length(x2));
for (i.x2 in x2) {
  for (i.x1 in x1) {
    ind.x1 <- which(x1 == i.x1);
    ind.x2 <- which(x2 == i.x2);
    x <- matrix(c(i.x1, i.x2),ncol=1);
    y1[ind.x1, ind.x2] <- f.yhat1(x)
    y2[ind.x1, ind.x2] <- f.yhat2(x)
  }
}

# plot contour plots with arrow in direction of steepest ascent (for main-effects model)
par(mfrow=c(1,2))
contour(x1, x2, y1, method = "edge", vfont = c("sans serif", "plain"),
  main = "main effects, only", xlab = "x1", ylab = "x2")
arrows(0, 0, 2 * D[1], 2 * D[2], col = 2)
contour(x1, x2, y2, method = "edge", vfont = c("sans serif", "plain"),
  main = "with interaction", xlab = "x1", ylab = "x2")
arrows(0, 0, 2 * D[1], 2 * D[2], col = 2, lty = 2)
```



(b) (5 pts) Show a plot of the path of steepest ascent for the model with interaction. Note that this path is not linear.

Solution: See part (a).

(c) (5 pts) Comment on the difference in the two paths.

Solution: Because the coeffient of the interaction model is small relative to the main effects, the correct path of steepest ascent for the interaction model is not much different from the the main effects model.