

ANOVA and Design of Experiments

Problem #1 - DoE

Use the following data to conduct a Design of Experiments analysis by hand (i.e. not using R). This is a 2^3 design, which is about the limit of complexity you'd want to handle by hand. You should produce a table similar to those in page 52 the DOE slides. I've provided a generic 2^3 table below; you would augment it to include the data below.

```
In [268]: signTable = read.csv("3-factor-sign-table.csv")
```

```
In [269]: signTable$Y = c(100, 15, 40, 30, 120, 110, 20, 50)
signTable
```

```
Out[269]:
```

	I	A	B	C	AB	AC	BC	ABC	Y
1	1	-1	-1	-1	1	1	1	-1	100
2	1	-1	-1	1	1	-1	-1	1	15
3	1	-1	1	-1	-1	1	-1	1	40
4	1	-1	1	1	-1	-1	1	-1	30
5	1	1	-1	-1	-1	-1	1	1	120
6	1	1	-1	1	-1	1	-1	-1	110
7	1	1	1	-1	1	-1	-1	-1	20
8	1	1	1	1	1	1	1	1	50

You should compute terms q_A , q_B , ..., q_{ABC} , or the effects size of each term.

```
In [270]: l=length(signTable[[1]])
qA = sum(signTable[[2]]*signTable[[9]])/l
qB = sum(signTable[[3]]*signTable[[9]])/l
qC = sum(signTable[[4]]*signTable[[9]])/l
qAB = sum(signTable[[5]]*signTable[[9]])/l
qAC = sum(signTable[[6]]*signTable[[9]])/l
qBC = sum(signTable[[7]]*signTable[[9]])/l
qABC = sum(signTable[[8]]*signTable[[9]])/l

effects = c(qA, qB, qC, qAB, qAC, qBC, qABC)
effects
```

```
Out[270]:      14.375 -25.625 -9.375 -14.375  14.375  14.375 -4.375
```

```
In [300]: a1=anova(lm(Y~A*B*C,data=signTable))
a1[[2]]/sum(a1[[2]])
#lm(Y~A*B*C,data=signTable)
```

Warning message:

In anova.lm(lm(Y ~ A * B * C, data = signTable)): ANOVA F-tests on an essentially perfect fit are unreliable

```
Out[300]:      0.129943502824859  0.41292065831491  0.0552689756816506
      0.129943502824859  0.129943502824859  0.129943502824859
      0.0120363547040039  0
```

Now, compute the percentage variation due to each

```
In [271]: y=(signTable[[9]])
total_variation = sum( (y - mean(y) )^2 )
variation = c((((2^3)*effects^2) / total_variation)*100)
variation
```

```
Out[271]:      12.9943502824859  41.292065831491  5.52689756816507
      12.9943502824859  12.9943502824859  12.9943502824859
      1.20363547040039
```

You can now compare it to the results using the **lm** and **anova** functions

```
In [274]: lm1=lm(Y~A*B*C,data=signTable)
          a1=anova(lm1)
          100*a1[[2]]/sum(a1[[2]])
```

Warning message:

In anova.lm(lm1): ANOVA F-tests on an essentially perfect fit are unreliable

```
Out[274]:      12.9943502824859  41.292065831491  5.52689756816506
          12.9943502824859  12.9943502824859  12.9943502824859
          1.20363547040039  0
```

Problem #2 - 2-Level Dhrystone

This file (dhry-2level.csv) contains the following table

```
In [2]: dhry2lvl = read.csv("dhry-2level.csv")
        #dhry2lvl # - print this out and it will have 24 rows
```

This is the result of running a 3×2^3 experimental design for evaluating the importance of certain compiler optimizations -- you can ignore their meaning for this problem (this is a subset of data from an experiment described below). Use R to conduct an analysis of variance for this data.

a) Set up a linear model. Treat the a, f and g level as categorical factors. You can do this using e.g.

```
data$bits = factor(data$bits).
```

```
In [301]: l2=lm(mips~a*f*g, data=dhry2lvl)
#summary(l2)
#dhry2lvl$a=factor(dhry2lvl$a)
#dhry2lvl$f=factor(dhry2lvl$f)
#dhry2lvl$g=factor(dhry2lvl$g)
#l2=lm(mips~dhry2lvl$a*dhry2lvl$f*dhry2lvl$g,data=dhry2lvl)
summary(l2)
a2=anova(l2)
#a2
```

```
Out[301]: Call:
lm(formula = mips ~ a * f * g, data = dhry2lvl)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-15.319	-6.148	1.662	5.069	17.420

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1440.5070	6.3922	225.355	< 2e-16 ***
a1	4.1047	9.0399	0.454	0.6559
f1	-114.7363	9.0399	-12.692	9.09e-10 ***
g1	4.8220	9.0399	0.533	0.6011
a1:f1	0.6523	12.7843	0.051	0.9599
a1:g1	17.5640	12.7843	1.374	0.1884
f1:g1	8.5200	12.7843	0.666	0.5146
a1:f1:g1	-34.2030	18.0798	-1.892	0.0768 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.07 on 16 degrees of freedom

Multiple R-squared: 0.9777, Adjusted R-squared: 0.968

F-statistic: 100.4 on 7 and 16 DF, p-value: 5.093e-12

b) Report the linear model.

```
In [6]: l2
```

```
Out[6]: Call:
lm(formula = mips ~ a * f * g, data = dhry2lvl)
```

Coefficients:

	a	f	g	a:f
(Intercept)				
a:g				
1390.0109	2.3310	-59.3505	4.6566	-4.1123
0.1156				
f:g				
-2.1454		-4.2754		

c) Note that R reports this differently for categorical data. The report should be missing terms for "a", "f" and so on. Why is that? What does the intercept represent? Note that categorical values must take one some value and the intercept needs to include the results of some value(s).

```
In [302]: lm(mips~a*f*g, data=dhry2lv1)

print("Equation will of form y=intercept+ax1+fx2+gx3. For x1=x2=x3=0, y
takes intercept value
Intercept is value of dependent variable when all independent variables
are zero")
```

```
Out[302]: Call:
lm(formula = mips ~ a * f * g, data = dhry2lv1)

Coefficients:
(Intercept)          a1          f1          g1          a1:f1
a1:g1
1440.5070          4.1047      -114.7363          4.8220          0.6523          1
7.5640
          f1:g1      a1:f1:g1
          8.5200      -34.2030

[1] "Equation will of form y=intercept+ax1+fx2+gx3. For x1=x2=x3=0, y t
akes intercept value \nIntercept is value of dependent variable when al
l independent variables are zero"
```

d) Determine the percentage of variation attributable to each factor. You can do this using the anova table as described in the lecture notes.

```
In [54]: (a2[[2]]/sum(a2[[2]]))*100

Out[54]:      0.148013552746935  95.9510130529515  0.590668677002191
          0.460648041919966  0.000364170616711544  0.125374103650913
          0.497908839991378  2.2260095611204
```

e) Compute the 95% confidence interval for the "opt" factor. The "anova" function doesn't do this, but you can see something similar using

```
confint( aov (1 ) )
```

where 'l' is your linear model. For models involving factors, the confidence interval is expressed for the different levels in each categorical factor.

```
In [135]: confint( aov (l2 ) )
```

```
Out[135]:
```

	2.5 %	97.5 %
(Intercept)	1426.956	1454.058
a1	-15.05903	23.26836
f1	-133.90003	-95.57264
g1	-14.3417	23.9857
a1:f1	-26.44923	27.75389
a1:g1	-9.537558	44.665558
f1:g1	-18.58156	35.62156
a1:f1:g1	-72.530392	4.124392

f) Plot out the diagnostic plots for the model. You can do this using "plot(aov(l))" where "l" is your model. Can you justify that your model is reasonable?

```
In [305]: #L2=Lm(mips~a+f+g, data=dhry2lvL)
par(mfrow=c(2,2))
aov(l2)
plot(aov(l2))
print("model looks reasonable as there is some residual error and points
are fitted")
```

```
Out[305]: Call:
aov(formula = l2)
```

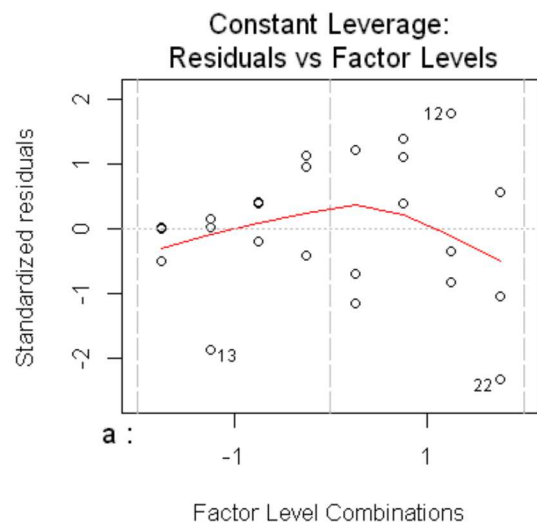
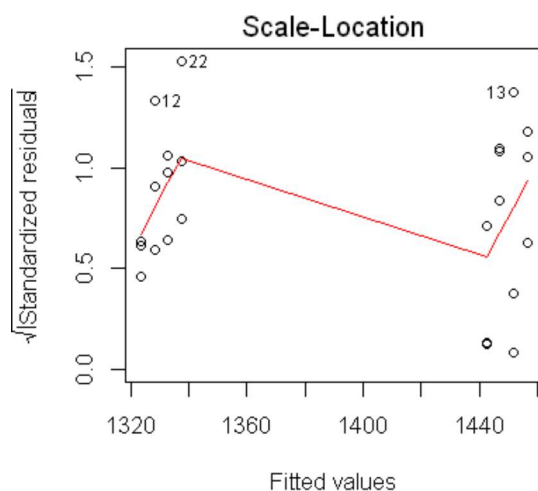
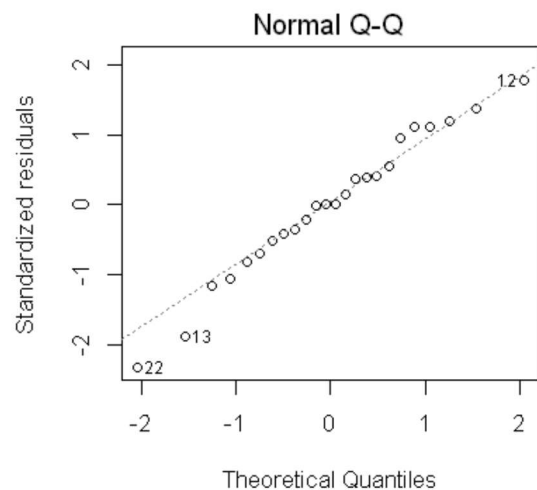
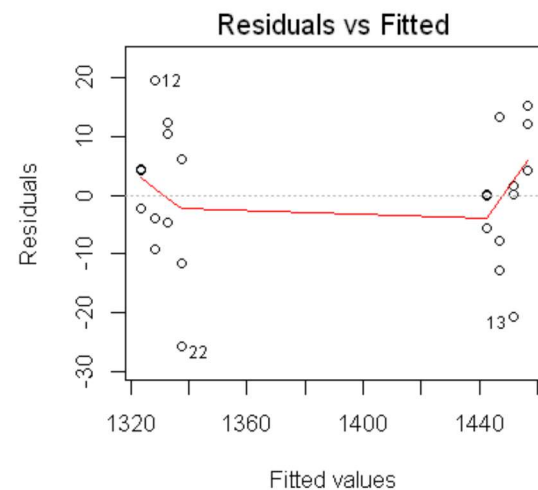
Terms:

	a	f	g	Residuals
Sum of Squares	130.41	84539.45	520.42	2916.61
Deg. of Freedom	1	1	1	20

Residual standard error: 12.07602

Estimated effects may be unbalanced

```
[1] "model looks reasonable as there is some residual error and points
are fitted"
```



Problem #3 -

The file OUTPUT-optlevel (see below) contains data from a 422 study with 2 replicates. The first column is the trial number (you shouldn't include this in your model, or you won't get any "residuals"). The first factor ('opt') has four levels (-O0, -O1, -O2, -O3). The second factor ('bits') has two levels (specifying -m32 or -m64 for compilation as a 32-bit or 64-bit application) and the third ('benchmark') has two levels (dhry11 or dhry21). In each case, the dependent variable is the performance in VAX MIPS (this is the output of the Dhrystone program). Use R to conduct an analysis of variance for this data.

```
In [142]: optlevels = read.csv("OUTPUT-optlevels.csv")
#optlevels # - this will have 32 rows
```

a) Set up a linear model. Treat the bits, progs and optimization level as categorical factors.

```
In [46]: l3=lm(mips~opt*bits*prog,data=optlevels)
```

b) Report the linear model. Note that R reports this differently for categorical data. The report should be missing terms for "opt0", "bits0" and so on. Why is that? What does the intercept represent?

```
In [264]: l3
lm(mips~opt+bits+prog,data=optlevels)
```

```
Out[264]: Call:
lm(formula = mips ~ opt * bits * prog, data = optlevels)
```

```
Coefficients:
(Intercept)          opt          bits          prog      opt:bi
ts          2873.35      1786.52      2032.69      266.12      -73.
68
      opt:prog      bits:prog  opt:bits:prog
      -105.66       -248.70        133.99
```

```
Out[264]: Call:
lm(formula = mips ~ opt + bits + prog, data = optlevels)
```

```
Coefficients:
(Intercept)          opt          bits          prog
      2969.53      1730.35      1898.32      83.77
```

c) Determine the percentage of variation attributable to each factor. You can do this using the anova table.

```
In [100]: a3=anova(l3)
var3=(a3[[2]]/sum(a3[[2]]))*100
var3[1:length(var3)-1]
```

```
Out[100]: 67.0970984584577 16.1512309713891 0.0314539445861829
0.000249938810763303 0.00837517072277202 0.00255039793738284
0.0251464314591841
```

d) Compute the 95% confidence interval for the "opt" factor. The "anova" function doesn't do this, but you can see something similar using

```
confint( aov (l ) )
```

where 'l' is your linear model. For models involving factors, the confidence interval is expressed for the different levels in each categorical factor.

```
In [58]: confint( aov (l3 ) )
```

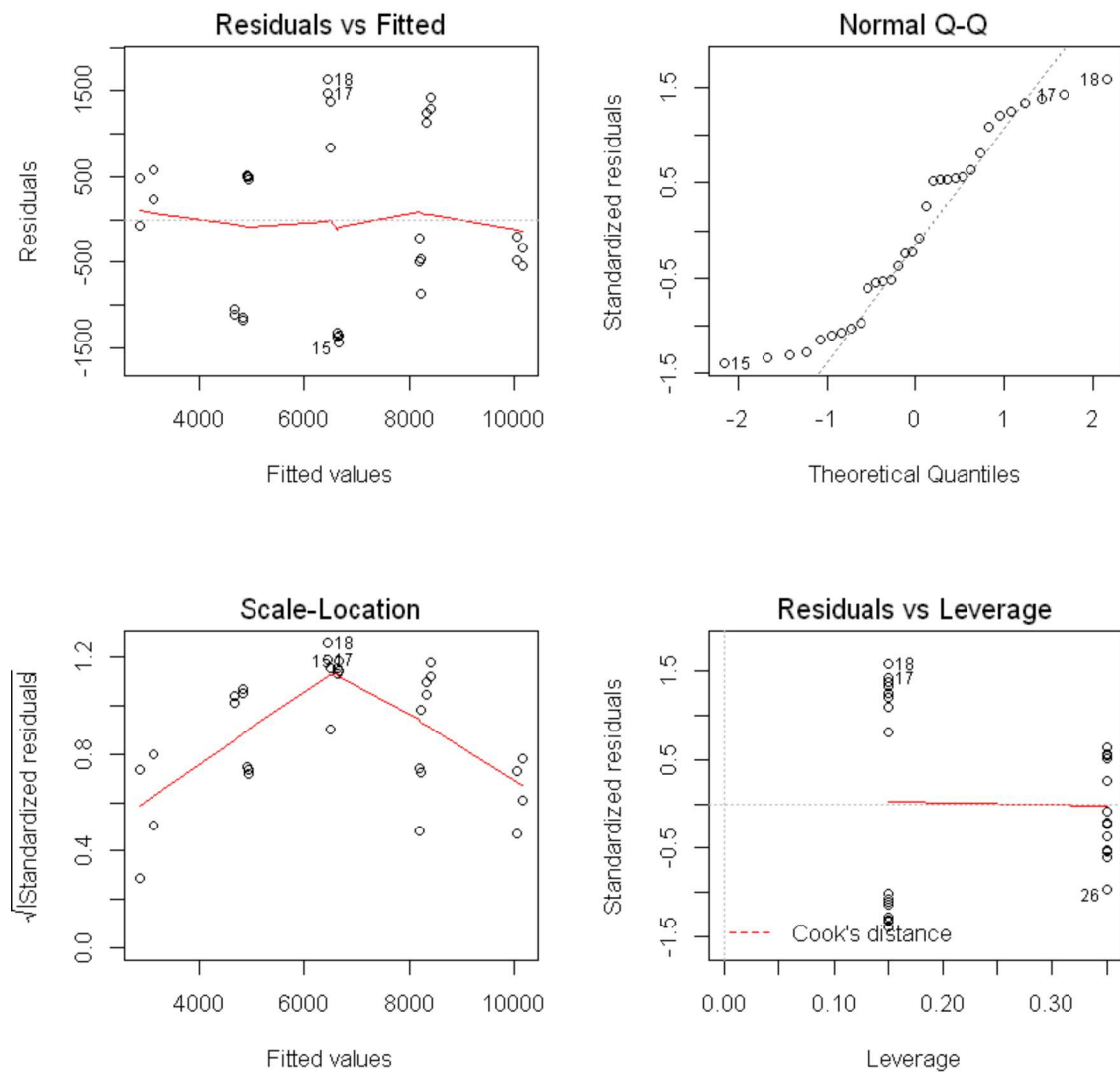
```
Out[58]:
```

	2.5 %	97.5 %
(Intercept)	1513.224	4233.469
opt	1059.506	2513.537
bits	109.1884	3956.1950
prog	-1657.385	2189.622
opt:bits	-1101.8313	954.4802
opt:prog	-1133.8163	922.4953
bits:prog	-2968.942	2471.547
opt:bits:prog	-1320.039	1588.024

e) Plot out the diagnostic plots for the model. You can do this using "plot(aov(l))" where 'l' is your model. Can you justify that your model is reasonable

```
In [306]: par(mfrow=c(2,2))
plot(aov(13))
print("model looks reasonable as there is some residual error and points
are fitted")
```

```
[1] "model looks reasonable as there is some residual error and points
are fitted"
```



Problem #4 -

This file (OUTPUT-21.csv) contains a $3 * 2^7$ experiment. The data was generated from the dhrystone 2.1 benchmark on a specific machine. The factors in the experiment correspond to different optimizations as indicated in the RUN script:

```
DATA = (
  ('-fstrength-reduce', '-fno-strength-reduce'), # a
  ('-fgcse', '-fno-gcse'), # b
  ('-floop-optimize', '-fno-loop-optimize'), # c
  ('-fpeephole', '-fno-peephole'), # d
  ('-finline-functions', '-fno-inline-functions'), # e
  ('-fomit-frame-pointer', '-fno-omit-frame-pointer'), # f
  ('-fwhole-program', '-fno-whole-program') # g
)
```

each factor has two levels - either the optimization is enabled or not. In my model, I called the factors o.a, o.b, o.c, etc for the different possible optimizations.

```
In [263]: fullOpt = read.csv("OUTPUT-21.csv")
          #fullOpt # - if you print this out, it will have 384 rows
```

a) There are several terms that contribute more variation to the results than the error/residual term, but most of the terms contribute little variation to the overall performance. Determine which single factor contributes in some meaningful way to variation in performance and report the percentage of variation explained by that single term. Explain how you did this.

```
In [262]: l4=lm(mips~o.a+o.b+o.c+o.d+o.e+o.f+o.g,data=fullOpt)
          a4=anova(l4)
          variation4=(a4[[2]]/sum(a4[[2]]))*100
          variation4[1:length(variation4)-1]
          print("o.f contributes to larger variation with 64.9%")
```

```
Out[262]:      0.0614008052380949  0.00109307819420532  0.375303890660707
          0.0344632153641588  0.118023848214343  64.9967839195917
          0.25399839722671
```

```
[1] "o.f contributes to larger variation with 64.9%"
```

b) What 2-factor interaction of factors contributes in some meaningful way to variation in performance. Explain how you determine which factors you're including.

```
In [235]: l4m=aov(mips~o.a*o.b,data=full0pt)
l4m
summary(aov(mips~o.a*o.b*o.c*o.d*o.e*o.f*o.g,data=full0pt))

#a4m=anova(l4m)
#a4m
#max(a4m[[2]])
#var4m=(a4m[[2]]/sum(a4m[[2]]))*100

#(var4m[1:length(var4m)-1])

print("Interaction between o.c & o.f lead to more variation. I found by
printing the linear model")
print("above and finding the maximum of 2-factor interaction")
```

Out[235]: Call:

```
aov(formula = mips ~ o.a * o.b, data = fullOpt)
```

Terms:

	o.a	o.b	o.a:o.b	Residuals
Sum of Squares	1962	35	8494	3185070
Deg. of Freedom	1	1	1	380

Residual standard error: 91.55198

Estimated effects are balanced

Out[235]:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
o.a	1	1962	1962	0.686	0.40820
o.b	1	35	35	0.012	0.91207
o.c	1	11993	11993	4.195	0.04157 *
o.d	1	1101	1101	0.385	0.53538
o.e	1	3772	3772	1.319	0.25181
o.f	1	2077012	2077012	726.494	< 2e-16 ***
o.g	1	8117	8117	2.839	0.09322 .
o.a:o.b	1	8494	8494	2.971	0.08598 .
o.a:o.c	1	7	7	0.002	0.96174
o.b:o.c	1	14	14	0.005	0.94391
o.a:o.d	1	348	348	0.122	0.72760
o.b:o.d	1	71	71	0.025	0.87490
o.c:o.d	1	311	311	0.109	0.74171
o.a:o.e	1	6828	6828	2.388	0.12349
o.b:o.e	1	2668	2668	0.933	0.33494
o.c:o.e	1	1971	1971	0.689	0.40714
o.d:o.e	1	190	190	0.066	0.79687
o.a:o.f	1	30	30	0.010	0.91901
o.b:o.f	1	647	647	0.226	0.63475
o.c:o.f	1	18188	18188	6.362	0.01227 *
o.d:o.f	1	3564	3564	1.247	0.26522
o.e:o.f	1	63	63	0.022	0.88225
o.a:o.g	1	825	825	0.289	0.59157
o.b:o.g	1	2329	2329	0.815	0.36758
o.c:o.g	1	63	63	0.022	0.88179
o.d:o.g	1	1821	1821	0.637	0.42557
o.e:o.g	1	721	721	0.252	0.61607
o.f:o.g	1	836	836	0.292	0.58922
o.a:o.b:o.c	1	11938	11938	4.176	0.04204 *
o.a:o.b:o.d	1	258	258	0.090	0.76432
o.a:o.c:o.d	1	217	217	0.076	0.78325
o.b:o.c:o.d	1	1470	1470	0.514	0.47404
o.a:o.b:o.e	1	14477	14477	5.064	0.02528 *
o.a:o.c:o.e	1	903	903	0.316	0.57459
o.b:o.c:o.e	1	6	6	0.002	0.96201
o.a:o.d:o.e	1	16	16	0.006	0.94045
o.b:o.d:o.e	1	3006	3006	1.051	0.30613
o.c:o.d:o.e	1	3107	3107	1.087	0.29816
o.a:o.b:o.f	1	325	325	0.114	0.73641
o.a:o.c:o.f	1	7074	7074	2.474	0.11696
o.b:o.c:o.f	1	2367	2367	0.828	0.36377
o.a:o.d:o.f	1	1066	1066	0.373	0.54206
o.b:o.d:o.f	1	4419	4419	1.546	0.21491
o.c:o.d:o.f	1	781	781	0.273	0.60159
o.a:o.e:o.f	1	2689	2689	0.940	0.33307
o.b:o.e:o.f	1	147	147	0.051	0.82080
o.c:o.e:o.f	1	84	84	0.029	0.86440
o.d:o.e:o.f	1	453	453	0.158	0.69096
o.a:o.b:o.g	1	354	354	0.124	0.72525
o.a:o.c:o.g	1	1495	1495	0.523	0.47033
o.b:o.c:o.g	1	925	925	0.324	0.56993
o.a:o.d:o.g	1	2960	2960	1.035	0.30983

o.b:o.d:o.g	1	27	27	0.010	0.92222	
o.c:o.d:o.g	1	5031	5031	1.760	0.18585	
o.a:o.e:o.g	1	31	31	0.011	0.91656	
o.b:o.e:o.g	1	286	286	0.100	0.75192	
o.c:o.e:o.g	1	721	721	0.252	0.61592	
o.d:o.e:o.g	1	927	927	0.324	0.56949	
o.a:o.f:o.g	1	2374	2374	0.831	0.36297	
o.b:o.f:o.g	1	3177	3177	1.111	0.29277	
o.c:o.f:o.g	1	6346	6346	2.220	0.13749	
o.d:o.f:o.g	1	32	32	0.011	0.91636	
o.e:o.f:o.g	1	64	64	0.022	0.88155	
o.a:o.b:o.c:o.d	1	8800	8800	3.078	0.08055	.
o.a:o.b:o.c:o.e	1	1352	1352	0.473	0.49226	
o.a:o.b:o.d:o.e	1	1087	1087	0.380	0.53808	
o.a:o.c:o.d:o.e	1	3426	3426	1.198	0.27471	
o.b:o.c:o.d:o.e	1	3601	3601	1.260	0.26279	
o.a:o.b:o.c:o.f	1	1472	1472	0.515	0.47373	
o.a:o.b:o.d:o.f	1	2410	2410	0.843	0.35941	
o.a:o.c:o.d:o.f	1	463	463	0.162	0.68766	
o.b:o.c:o.d:o.f	1	220	220	0.077	0.78190	
o.a:o.b:o.e:o.f	1	832	832	0.291	0.59009	
o.a:o.c:o.e:o.f	1	2795	2795	0.978	0.32370	
o.b:o.c:o.e:o.f	1	605	605	0.212	0.64593	
o.a:o.d:o.e:o.f	1	11040	11040	3.862	0.05048	.
o.b:o.d:o.e:o.f	1	4648	4648	1.626	0.20347	
o.c:o.d:o.e:o.f	1	166	166	0.058	0.80996	
o.a:o.b:o.c:o.g	1	875	875	0.306	0.58069	
o.a:o.b:o.d:o.g	1	899	899	0.315	0.57537	
o.a:o.c:o.d:o.g	1	76	76	0.026	0.87102	
o.b:o.c:o.d:o.g	1	11059	11059	3.868	0.05029	.
o.a:o.b:o.e:o.g	1	7244	7244	2.534	0.11266	
o.a:o.c:o.e:o.g	1	169	169	0.059	0.80831	
o.b:o.c:o.e:o.g	1	76	76	0.027	0.87029	
o.a:o.d:o.e:o.g	1	994	994	0.348	0.55597	
o.b:o.d:o.e:o.g	1	2456	2456	0.859	0.35486	
o.c:o.d:o.e:o.g	1	12459	12459	4.358	0.03782	*
o.a:o.b:o.f:o.g	1	946	946	0.331	0.56563	
o.a:o.c:o.f:o.g	1	12197	12197	4.266	0.03988	*
o.b:o.c:o.f:o.g	1	685	685	0.240	0.62483	
o.a:o.d:o.f:o.g	1	6866	6866	2.402	0.12245	
o.b:o.d:o.f:o.g	1	1082	1082	0.378	0.53897	
o.c:o.d:o.f:o.g	1	12	12	0.004	0.94927	
o.a:o.e:o.f:o.g	1	595	595	0.208	0.64869	
o.b:o.e:o.f:o.g	1	5977	5977	2.090	0.14944	
o.c:o.e:o.f:o.g	1	1512	1512	0.529	0.46778	
o.d:o.e:o.f:o.g	1	6706	6706	2.345	0.12688	
o.a:o.b:o.c:o.d:o.e	1	21	21	0.007	0.93238	
o.a:o.b:o.c:o.d:o.f	1	2205	2205	0.771	0.38068	
o.a:o.b:o.c:o.e:o.f	1	4058	4058	1.419	0.23459	
o.a:o.b:o.d:o.e:o.f	1	2933	2933	1.026	0.31204	
o.a:o.c:o.d:o.e:o.f	1	109	109	0.038	0.84521	
o.b:o.c:o.d:o.e:o.f	1	965	965	0.338	0.56173	
o.a:o.b:o.c:o.d:o.g	1	4313	4313	1.509	0.22047	

o.a:o.b:o.c:o.e:o.g	1	250	250	0.088	0.76760	
o.a:o.b:o.d:o.e:o.g	1	3192	3192	1.117	0.29165	
o.a:o.c:o.d:o.e:o.g	1	5173	5173	1.809	0.17979	
o.b:o.c:o.d:o.e:o.g	1	9050	9050	3.165	0.07640	.
o.a:o.b:o.c:o.f:o.g	1	6507	6507	2.276	0.13261	
o.a:o.b:o.d:o.f:o.g	1	11	11	0.004	0.95086	
o.a:o.c:o.d:o.f:o.g	1	3118	3118	1.091	0.29731	
o.b:o.c:o.d:o.f:o.g	1	601	601	0.210	0.64700	
o.a:o.b:o.e:o.f:o.g	1	68	68	0.024	0.87753	
o.a:o.c:o.e:o.f:o.g	1	3107	3107	1.087	0.29816	
o.b:o.c:o.e:o.f:o.g	1	19859	19859	6.946	0.00891	**
o.a:o.d:o.e:o.f:o.g	1	0	0	0.000	0.99754	
o.b:o.d:o.e:o.f:o.g	1	15	15	0.005	0.94226	
o.c:o.d:o.e:o.f:o.g	1	1673	1673	0.585	0.44498	
o.a:o.b:o.c:o.d:o.e:o.f	1	4533	4533	1.586	0.20911	
o.a:o.b:o.c:o.d:o.e:o.g	1	24353	24353	8.518	0.00383	**
o.a:o.b:o.c:o.d:o.f:o.g	1	162	162	0.057	0.81217	
o.a:o.b:o.c:o.e:o.f:o.g	1	816	816	0.285	0.59371	
o.a:o.b:o.d:o.e:o.f:o.g	1	4706	4706	1.646	0.20067	
o.a:o.c:o.d:o.e:o.f:o.g	1	2306	2306	0.807	0.36995	
o.b:o.c:o.d:o.e:o.f:o.g	1	1718	1718	0.601	0.43897	
o.a:o.b:o.c:o.d:o.e:o.f:o.g	1	8516	8516	2.979	0.08556	.
Residuals	256	731892	2859			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

[1] "Interaction between o.c & o.f lead to more variation. I found by printing the linear model"

[1] "above and finding the maximum of 2-factor interaction"

c) There any two significant intercatons that involve more than 2 factors. List those & explain how you did this.

```
In [260]: #print("Interaction between o.b:o.c:o.d:o.g 5.366 Lead to more variatio
n. ")
print("Interaction between o.b:o.c:o.e:o.f:o.g & o.a:o.b:o.c:o.d:o.e:o.g
lead to more variation. ")
print("I found by printing the aov above and finding the maximum of all
interaction")
```

[1] "Interaction between o.b:o.c:o.e:o.f:o.g & o.a:o.b:o.c:o.d:o.e:o.g lead to more variation. "

[1] "I found by printing the aov above and finding the maximum of all interaction"

Problem #4 - reduced

I've taken the data from the previous question and prepared a $3 \times 2^{7-4}$ design table. You're left with a 24 row table (8 factors, repeated 3 times). This is in the following table.

```
In [237]: reducedOpt = read.csv("OUTPUT-21-reduced.csv")
#reducedOpt # if you print this out, it will have 24 rows
```

a) Again, there should be a single factor that explains the most variation, but the percentage of variation explained by that factor should now be different. Compute the percentage for that single factor.

```
In [258]: #l4r=lm(mips~o.a+o.b+o.c+o.d+o.e+o.f+o.g,data=reducedOpt)
l4r=lm(mips~o.a*o.b*o.c*o.d*o.e*o.f*o.g,data=reducedOpt)
a4r=anova(l4r)
#summary(aov(mips~o.a*o.b*o.c*o.d*o.e*o.f*o.g,data=fullOpt))
summary(aov(mips~o.a*o.b*o.c*o.d*o.e*o.f*o.g,data=reducedOpt))

print("% variation is different for reduced output because
  apart from factors a,b,c all other factors are confounded with these th
ree values")
#l4r
var4r=(a4r[[2]]/sum(a4r[[2]]))*100
var4r[1:length(var4r)-1]
print("o.f factor generates more variation")
```

```
Out[258]:
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
o.a	1	668	668	0.313	0.584
o.b	1	609	609	0.285	0.601
o.c	1	1565	1565	0.733	0.405
o.d	1	155	155	0.073	0.791
o.e	1	1705	1705	0.799	0.385
o.f	1	126683	126683	59.340	9.02e-07 ***
o.g	1	545	545	0.255	0.620
Residuals	16	34158	2135		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

[1] "% variation is different for reduced output because \n apart from
factors a,b,c all other factors are confounded with these three values"
```

```
Out[258]: 0.402482072477033 0.366647991524676 0.942312347705059
0.093467756933563 1.02663791199098 76.2742739498329
0.32829259422425

[1] "o.f factor generates more variation"
```

b) Now, determine why that variation is different. Show that the "F" factor is confounded with the ABCDEG factor and explain why that changes the variation attributed to F.

```
In [259]: #a4
reducedOpt$o.a*reducedOpt$o.b*reducedOpt$o.c*reducedOpt$o.d*reducedOpt
$o.e*reducedOpt$o.g==reducedOpt$o.f
print("Yes, Factor o.f is confounded with abcdeg")
```

```
Out[259]:      TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
      TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
      TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
[1] "Yes, Factor o.f is confounded with abcdeg"
```