

# 490 Group Project Experiment 2

```
library(AlgDesign)

#1.
#factor A has 3 levels, factor B and C each has 2 levels

data<-gen.factorial(levels=c(3,2,2),nVars=3,varNames=c("A","B","C"))
data

desD<-optFederov(~quad(A) + B + C, data, nTrials=10)
desD

design<-data.frame(desD$design)
design_rep<-design[rep(seq_len(nrow(design)), each = 3), ]
design_rep
```

```
> desD<-optFederov(~quad(A) + B + C, data, nTrials=10)
> desD
$D
[1] 0.6602178

$A
[1] 2.142857

$Ge
[1] 0.7

$Dea
[1] 0.651

$design
  A B C
2  0 -1 -1
3  1 -1 -1
4 -1  1 -1
5  0  1 -1
6  1  1 -1
7 -1 -1  1
8  0 -1  1
10 -1  1  1
11  0  1  1
12  1  1  1

$rows
[1] 2 3 4 5 6 7 8 10 11 12
```

```
> design_rep
  A B C
2  0 -1 -1
2.1  0 -1 -1
2.2  0 -1 -1
3  1 -1 -1
3.1  1 -1 -1
3.2  1 -1 -1
4 -1  1 -1
4.1 -1  1 -1
4.2 -1  1 -1
5  0  1 -1
5.1  0  1 -1
5.2  0  1 -1
6  1  1 -1
6.1  1  1 -1
6.2  1  1 -1
7 -1 -1  1
7.1 -1 -1  1
7.2 -1 -1  1
8  0 -1  1
8.1  0 -1  1
8.2  0 -1  1
10 -1  1  1
10.1 -1  1  1
10.2 -1  1  1
11  0  1  1
11.1  0  1  1
11.2  0  1  1
12  1  1  1
12.1  1  1  1
12.2  1  1  1
```

```
#2.

library(readr)
group_response <- read_csv("group project/group_response.csv")
y1<- group_response["y1"][complete.cases(group_response["y1"]), ]
y1

df<- cbind(design_rep, y1)
df

library(dplyr)
df <- df %>% mutate(A_squared = A^2)
df

fit <- aov(data = df, y1 ~ A_squared * A * B * C)
summary(fit)
```

```
> fit <- aov(data = df, y1 ~ A_squared * A * B * C)
> summary(fit)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A_squared	1	2266	2266	3.661	0.070111 .
A	1	924	924	1.494	0.235799
B	1	40	40	0.065	0.800740
C	1	1030	1030	1.664	0.211770
A_squared:B	1	686	686	1.108	0.305036
A:B	1	616	616	0.995	0.330463
A_squared:C	1	2083	2083	3.367	0.081451 .
A:C	1	10740	10740	17.357	0.000477 ***
B:C	1	1408	1408	2.276	0.147030
Residuals	20	12376	619		

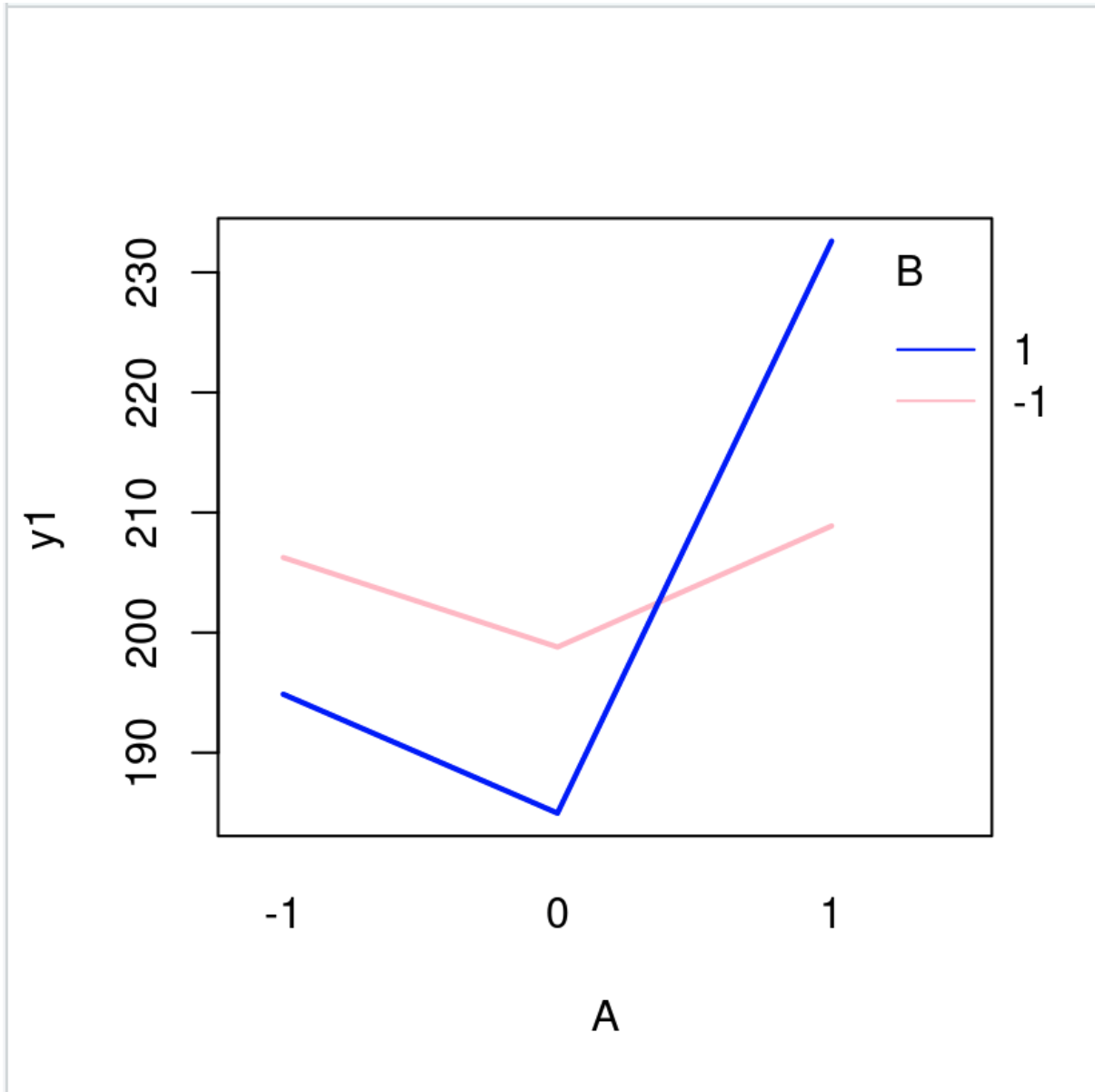
```
---
```

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

Based on anova table, interaction factors "A:C" is significant, and "A\_squared:C", "A\_squared" have p\_value very close to significance, but main factors do not have significant effects.

```
#3.
interaction.plot(x.factor = df$A, #x-axis variable
                 trace.factor = df$B, #variable for lines
                 response = df$y1, #y-axis variable
                 fun = mean, #metric to plot)
```

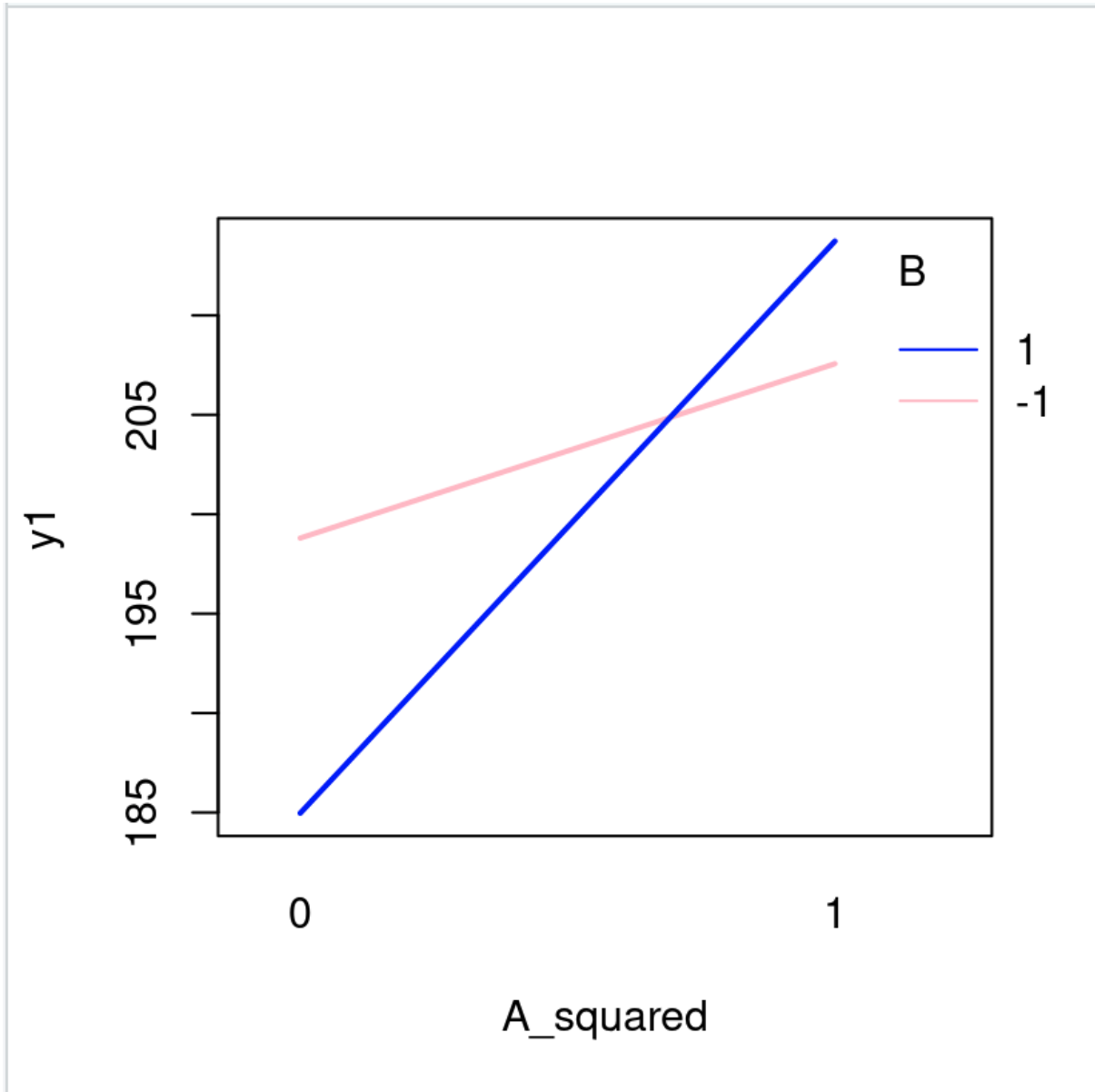
```
ylab = "y1",
xlab = "A",
col = c("pink", "blue", "Red"),
lty = 1, #line type
lwd = 2, #line width
trace.label = "B")
```



A:B has nonparallel lines, so likely there is a “cross-over” interaction effect.

```
interaction.plot(x.factor = df$A_squared, #x-axis variable
               trace.factor = df$B, #variable for lines
               response = df$y1, #y-axis variable
               fun = mean, #metric to plot)
```

```
ylab = "y1",
xlab = "A_squared",
col = c("pink", "blue", "Red"),
lty = 1, #line type
lwd = 2, #line width
trace.label = "B")
```



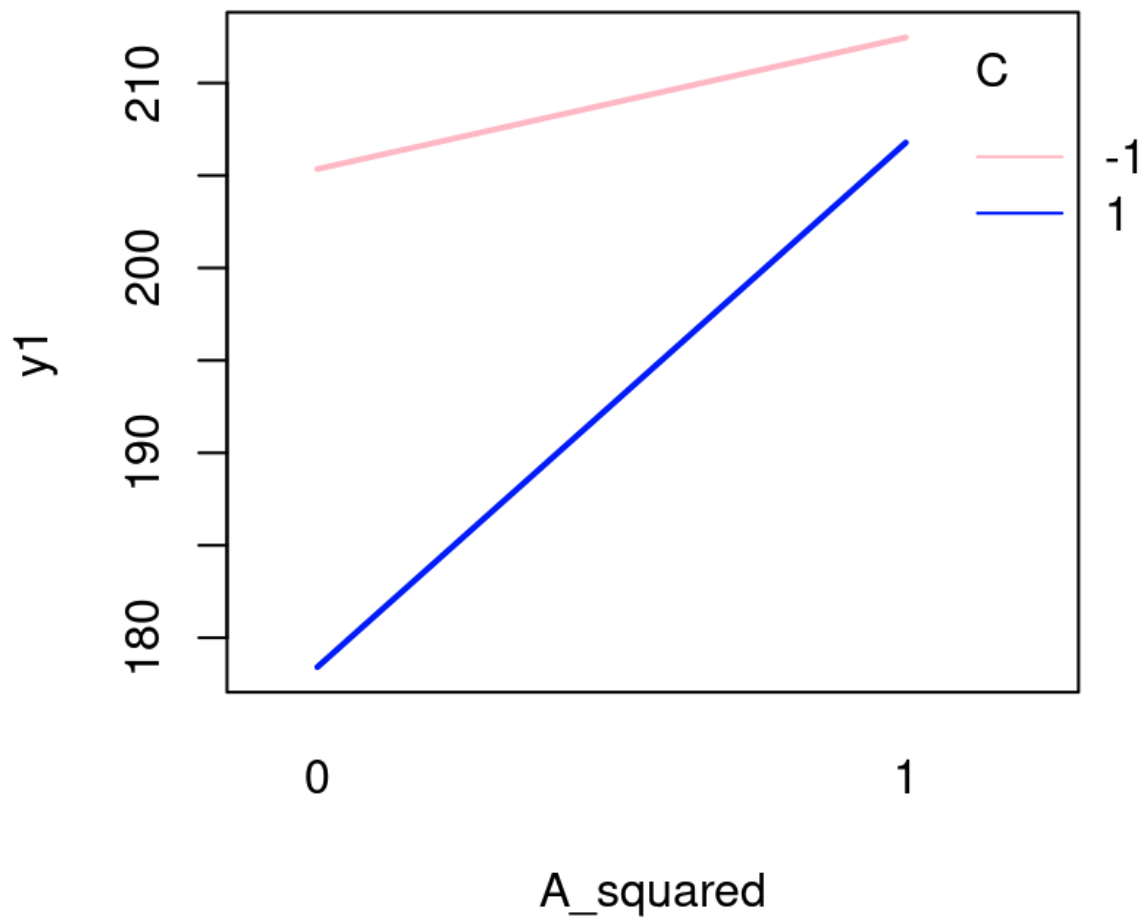
A\_squared: B has nonparallel lines, so likely there is a “cross-over” interaction effect.

```
interaction.plot(x.factor = df$A_squared, #x-axis variable
                trace.factor = df$C, #variable for lines
```

```

response = df$y1, #y-axis variable
fun = mean, #metric to plot
ylab = "y1",
xlab = "A_squared",
col = c("pink", "blue", "Red"),
lty = 1, #line type
lwd = 2, #line width
trace.label = "C")

```



A\_squared: C has nonparallel lines, there is an interaction effect.

```

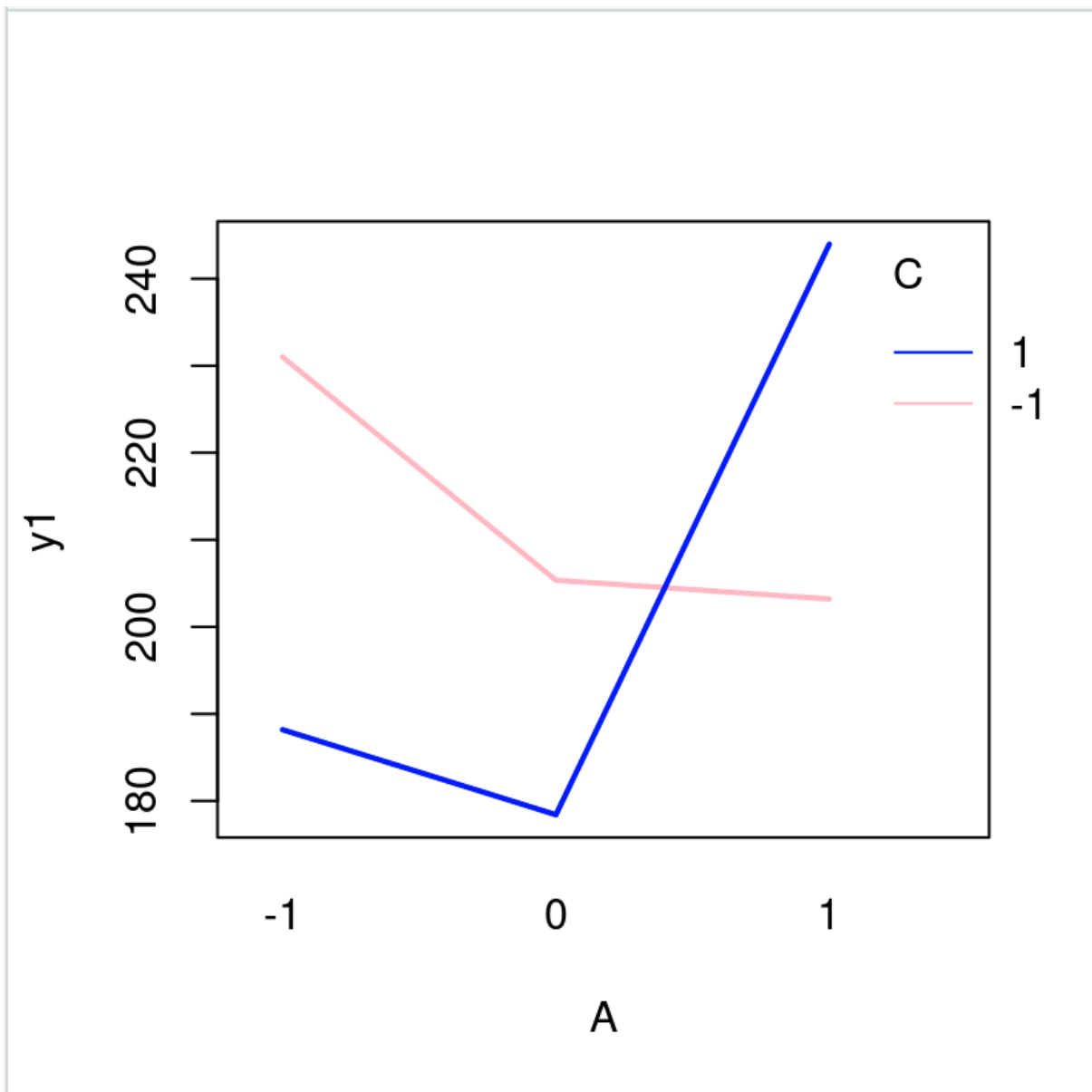
interaction.plot(x.factor = df$A, #x-axis variable
               trace.factor = df$C, #variable for lines

```

```

response = df$y1, #y-axis variable
fun = mean, #metric to plot
ylab = "y1",
xlab = "A",
col = c("pink", "blue", "Red"),
lty = 1, #line type
lwd = 2, #line width
trace.label = "C")

```



A: C has nonparallel lines, there is an interaction effect as "crossover"

```

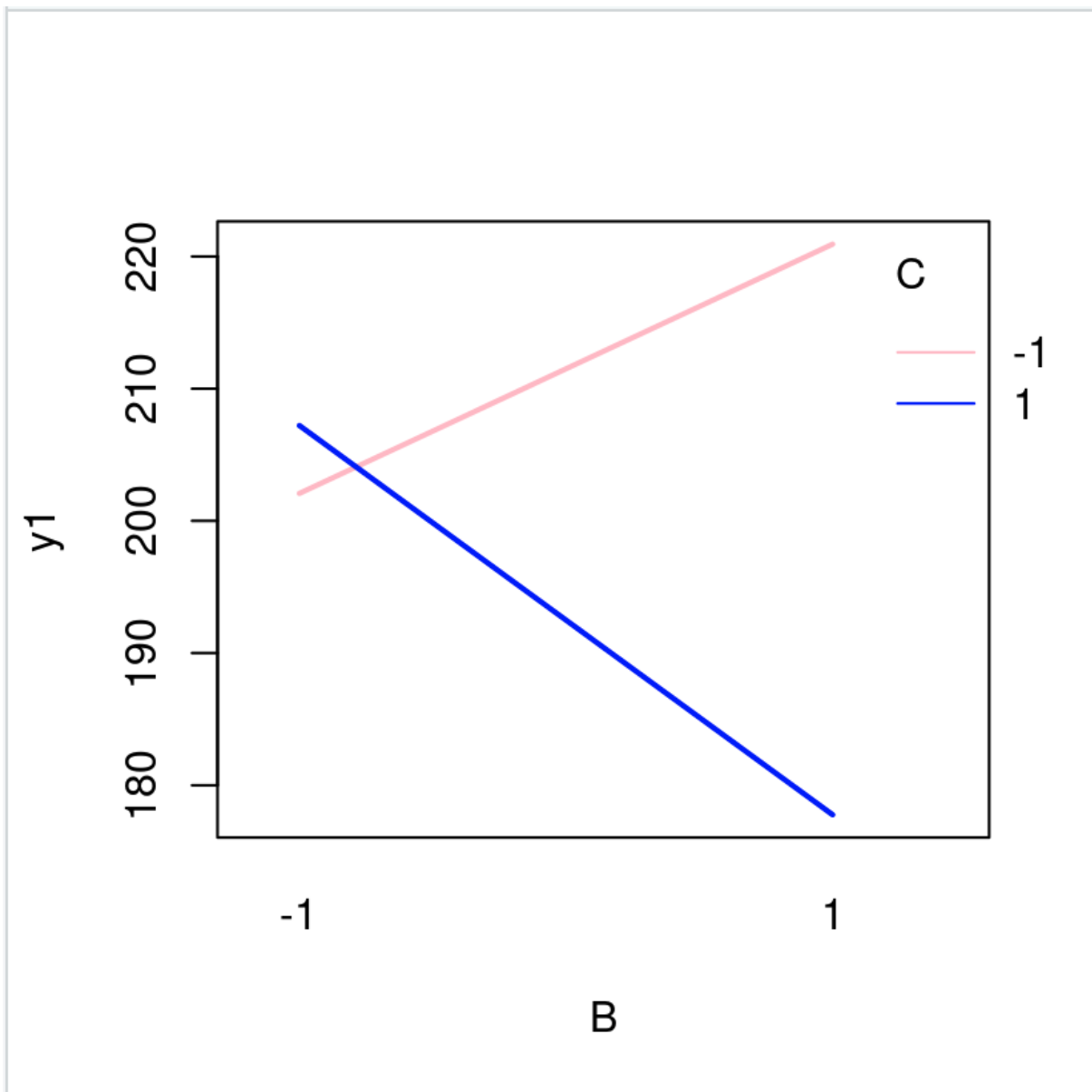
interaction.plot(x.factor = df$B, #x-axis variable
               trace.factor = df$C, #variable for lines

```

```

response = df$y1, #y-axis variable
fun = mean, #metric to plot
ylab = "y1",
xlab = "B",
col = c("pink", "blue", "Red"),
lty = 1, #line type
lwd = 2, #line width
trace.label = "C")

```



B: C has nonparallel lines, there is an interaction effect as "crossover"

anova table only shows the significance between "A\_squared:C", "A:C", while interaction plot shows somewhat interaction effect between all of these terms,

which could be resulted by variability in sample.

```
#regression
# regression model include all the significant terms from anova table.
m1<-lm(data=df, y1 ~ A_squared*C+A*C+A_squared)
summary(m1)
```

```
> summary(m1)
```

Call:

```
lm.default(formula = y1 ~ A_squared * C + A * C + A_squared,
            data = df)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-47.550	-17.742	-3.725	15.125	62.083

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	191.883	8.294	23.135	< 2e-16	***
A_squared	24.704	10.972	2.252	0.03378	*
C	-13.467	8.294	-1.624	0.11752	
A	6.996	7.183	0.974	0.33980	
A_squared:C	12.954	10.972	1.181	0.24931	
C:A	20.896	7.183	2.909	0.00769	**

---

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

Residual standard error: 28.73 on 24 degrees of freedom

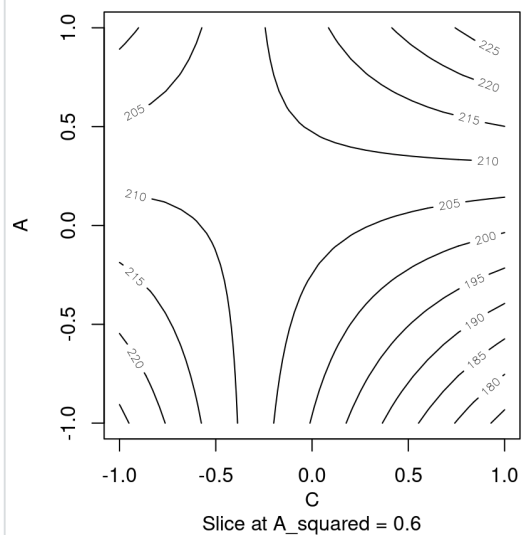
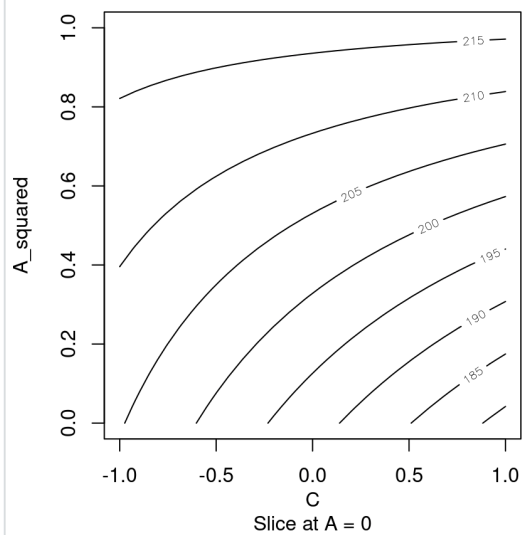
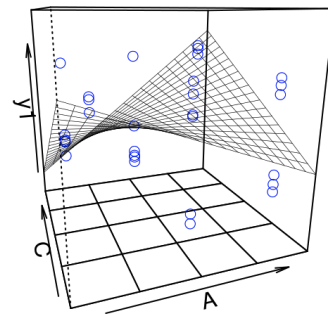
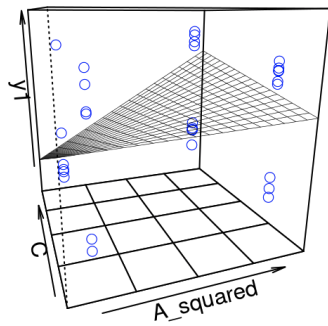
Multiple R-squared: 0.3841, Adjusted R-squared: 0.2558

F-statistic: 2.994 on 5 and 24 DF, p-value: 0.03073

```
# perspective and contour plot
plotPlane(m1, plotx1="A_squared", plotx2="C")
plotPlane(m1, plotx1="A", plotx2="C")

contour(m1, A_squared~C)
contour(m1, A~C)
```

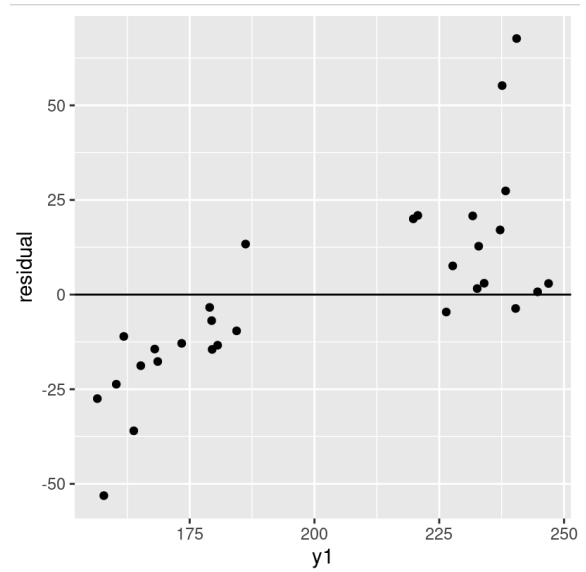
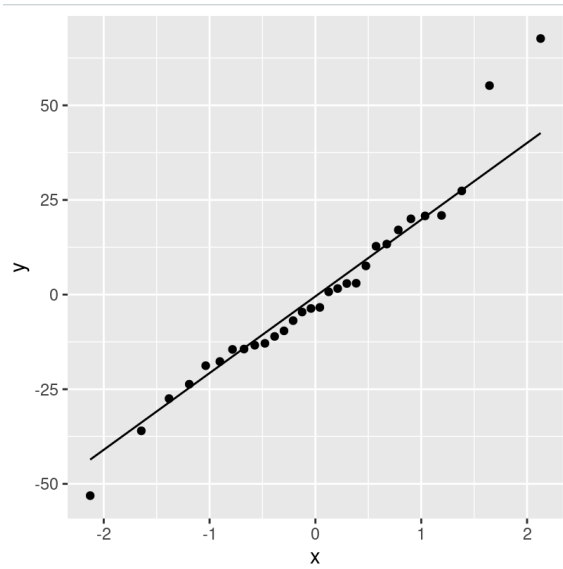




```
#4.
points<-get_regression_points(m1)
points

shapiro.test(points$residual)

ggplot(data=points, aes(sample=residual)) + stat_qq() + stat_qq_line()
ggplot(data=points, aes(x=y1, y=residual)) + geom_point() + geom_hline(yintercept=0)
```



Q-Q plot shows an acceptable straight line, so the residuals should be normally distributed

while the residuals against the predicted values shows a somewhat increasing trend rather than a constant spread out, which is the violation of equal variance assumption.