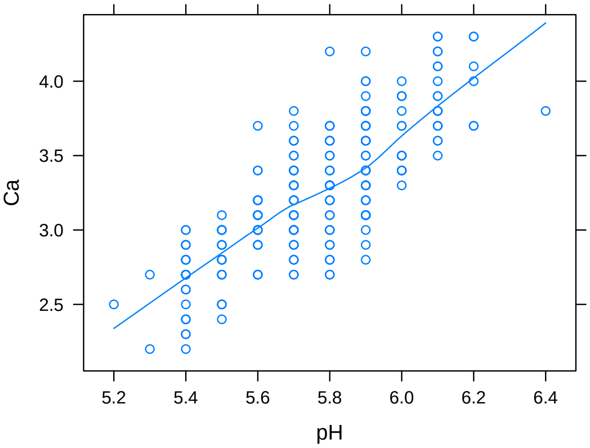
Stat 505 Assignment 6

22 points

Solutions

1. In a study of soil properties, samples were taken on a 10 point by 25 point grid. We'll work with two variables: response Ca (calcium concentration) and predictor pH (low numbers are acidic, high numbers basic, 7 is neither).

   (a) Make a scatterplot of the two variables and fit a model for Ca based on pH. (Choose the form of the model based on the scatterplot.) Print the estimated coefficients and discuss the relationship.



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-5.8657	0.4546	-12.90	0.0000
pH	1.5826	0.0789	20.05	0.0000

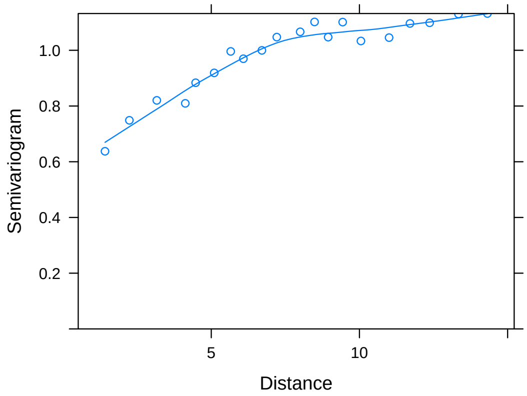
2

There seems to be a strong linear relationship between pH and Ca. As pH increases, so does Ca. The estimated line of best fit is $\widehat{y} = -5.87 + 1.58x$. ✓ *The model explains 61.8% of the variance of calcium.*

with SE(slope) = 0.08

   (b) Plot the semivariogram of the residuals using euclidean distance on column and row. Does it appear that there is some spatial correlation? Make a guess about values for range and nugget.

```r
calcium.gls0 <- gls(Ca ~ pH, data = soils)
plot(Variogram(calcium.gls0, form = ~column + row, max = 15))
```



2

I see a definite an upward trend in the variogram, I guess a range of 8 and nugget of .5

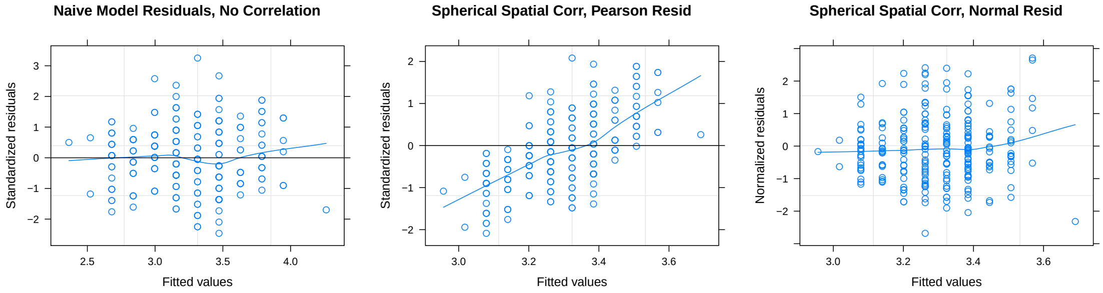
(c) Fit the five forms of spatial correlation available in the nlme library. Compare them with each other and with the original model. Do any of the spatial correlation fits improve AIC by more than 2 units? Are any of them "significant" improvements according to a LRT?

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
calcium.gls0	1	3	73.27	83.81	-33.64			
calcium.glsS	2	5	-97.09	-79.52	53.54	1 vs 2	174.36	0.00
calcium.glsE	3	5	-96.47	-78.91	53.24			
calcium.glsL	4	5	-64.98	-47.41	37.49			
calcium.glsG	5	5	-84.05	-66.48	47.02			
calcium.glsR	6	5	-94.18	-76.62	52.09			

2

Each spatial structure improves likelihood enormously, AIC by over 130, and is a huge improvement (p-value < 0.001). The spherical fit seems best, but only slightly better than exponential.

(d) Use the plot function on the first model with no spatial correlation and on the best of the spatial correlation models. Describe any problems you see.



1

I found these plots disturbing when I first saw them, because I expect residuals to be orthogonal to fitted values in plot 2, as well as plot 1. However, the Pearson residuals of the spatial model show a linear trend. This illustrates the fact that we are using a projection operator which is not the PPO, and fits are not orthogonal to residuals.

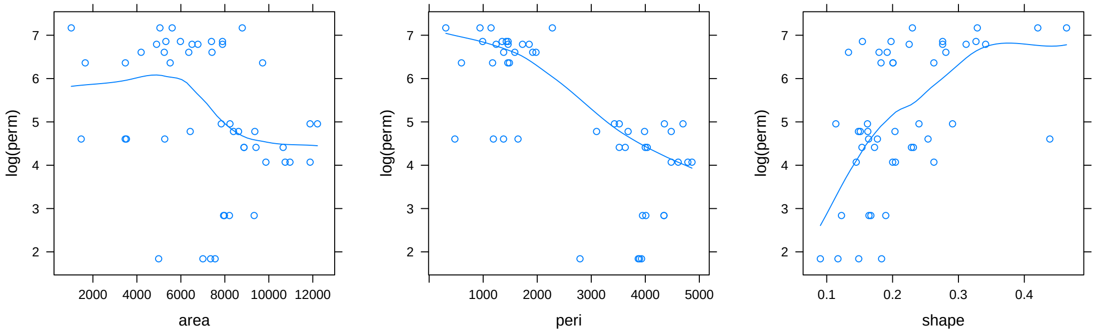
(e) Redo the second plot with normalized residuals instead of the default. Read the help on residuals.gls. Write out a matrix equation to show how the normalized residuals are different from the default residuals.

See plot above right. The Pearson residuals are adjusted so that all have the same variance: $\boldsymbol{e}_{pearson} = \boldsymbol{D}^{-1}\boldsymbol{e}$*, where* $\boldsymbol{D}$ *is diagonal with* $\sqrt{s_i^2(1-h_{ii})}$ *in the* (i,i) *position. However the Pearson correction does not account for the correlation structure. The "normalized" residuals are computed as* $\widehat{\boldsymbol{R}}^{-1/2}\widehat{\boldsymbol{D}}^{-1}\boldsymbol{e}$ *where* $\boldsymbol{R}$ *is the correlation matrix generated by, in this case, the spatial correlations and the* $-1/2$ *power means a Cholesky decomposition of the inverse of* $\widehat{\boldsymbol{R}}$*. The premultiplier actually rotates the residuals so that they again appear orthogonal to the fits. This relates back to work we did right after proing the Gauss–Markov Theorem. We showed that premultiplying our linear model by* $\boldsymbol{V}^{-1/2}$ *gives a new linear model where* $\boldsymbol{\epsilon} \sim (\boldsymbol{0}, \sigma^2\boldsymbol{I})$*. We can also write* $\boldsymbol{V}^{-1/2}$ *as* $\boldsymbol{R}^{-1/2}\boldsymbol{D}^{-1}$*, so we are multiplying by* $\widehat{\boldsymbol{V}}^{-1/2}$ *.*

2

2. Examine the data on rock cores available in R as `data(rock)` (see help, too). The object is to model log(permeability) based on observations taken from image analysis of core cross sections. Explanatory variables are the total area, total perimeter, and shape of microscopic pores.

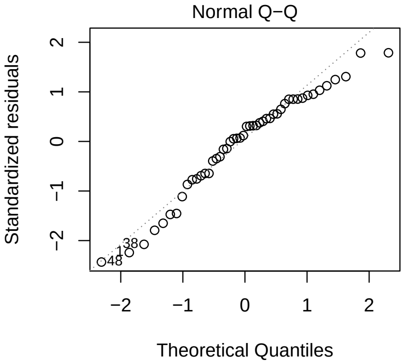
   (a) Plot each predictor against `log(perm)`.

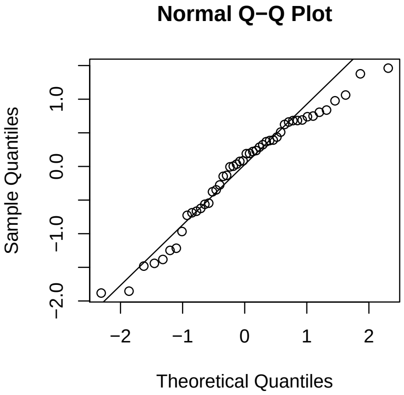


2

These are not strong and not linear relationships. Shape has the strongest correlation with log(perm).

   (b) Fit a linear model and a robust linear model using all three predictors. Use qqnorm and qqline on the plain residuals.





Coefficients for area and perimeter match well, and shape coefficient differs by only SE/4, so robust fit doesn't change things much. I see no problems in the residual plots. The right tail of the residuals is slightly shorter than for a normal distribution. The robust residuals are not very different from the regular OLS residuals.

(c) Due to some concern about normality, you are asked to give a non-normality-based estimate of the coefficient for shape. 1

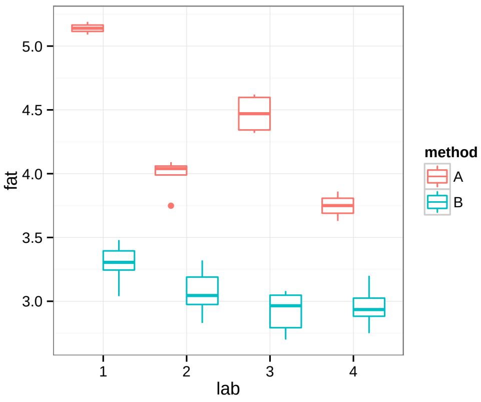
i. Discuss whether model-based or case-based bootstrapping would be most appropriate in this setting. (Here each core is considered a random draw from some population of cores of interest. Ignore that fact that these observations are really repeated measurements since each core was measured optically 4 times, but the permeability only once.)

Use case-based bootstrapping because the $\mathbf{X}$ *values are random and depend on the sample drawn. It is not the only* $\mathbf{X}$ *of interest, rather a representative data matrix from a bigger population of data (core samples).*

ii. Provide a 95% bootstrap CI for the coefficient for shape. 1

The 95% BCa interval for shape given area and perimeter are in the model is (-2.335, 6.231), which contains 0, so is seems that shape does not add much to the model.

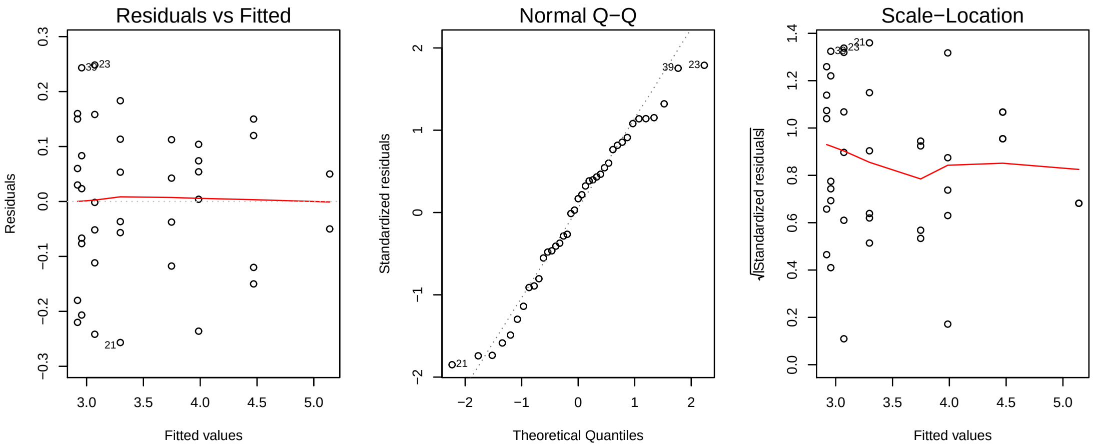
3. In a study of milkfat in yogurt, 1



samples known to have 3% milkfat were sent to four labs, and the labs were told to use either method A or B (assigned at random) to measure the milkfat. The primary concern is that method A might give higher readings than measure B. We need to look for an interaction between method and lab.

(a) Fit a 2-way ANOVA model with interaction and discuss the results (including diagnostics).

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
method	1	12.08	12.08	522.72	0.0000
lab	3	2.09	0.70	30.08	0.0000
method:lab	3	1.54	0.51	22.19	0.0000
Residuals	31	0.72	0.02		



Both main effects and the interaction have small p-values, so there are strong lab and method effects and the method effects change with lab. I see no problems in the residual diagnostic plots. There is no fan in the residuals and the residuals are just slightly shorter tailed than a normal distribution. 2

(b) To avoid the normality assumption, we will also use a bootstrap approach. Decide whether case-based or model-based bootstrapping is more appropriate and explain why.

These are the only labs and methods of interest, and hence there is only one $\boldsymbol{X}$ *matrix. Therefore model based bootstrap is appropriate.* 1

(c) Use your chosen bootstrap method to test to see if

i. Are the interaction terms needed? Use an approach similar to the extra sum of squares F test, but via bootstrapping.

I used 2 methods for computing a bootstrap F test, one using a contrast, as in the notes, gave a p-value of 0.002 because the test stat, 66.58 was larger than the largest of 500 bootstrapped F values under H_0. *The other collected F stats from the anova command where data were generated under the null model of no interaction, then tested under the full model with interaction gave a maximum F (under the null) of 5.85 compared to an observed* $F = 22.2$, *so again the p-value is less than 1/500.* 2

ii. If the interaction is ignorable, refit without it.

The interaction has a p-value $< .0001$ *under normality assumption, or of 1/500 for the bootstrap, so we can't drop it.* 1

iii. Provide a 90% confidence interval estimate of the method effect after adjusting out lab effects.

You should struggle a bit here, because we just said that there is a strong lab × method interaction, so we can't estimate a method effect without considering labs. I think it makes sense to think of the average difference in method means across the four labs. If our model is

$$y_{ijkm} = \mu + \tau_i + \alpha_j + \delta_{ij} + \epsilon_{ijkm}$$

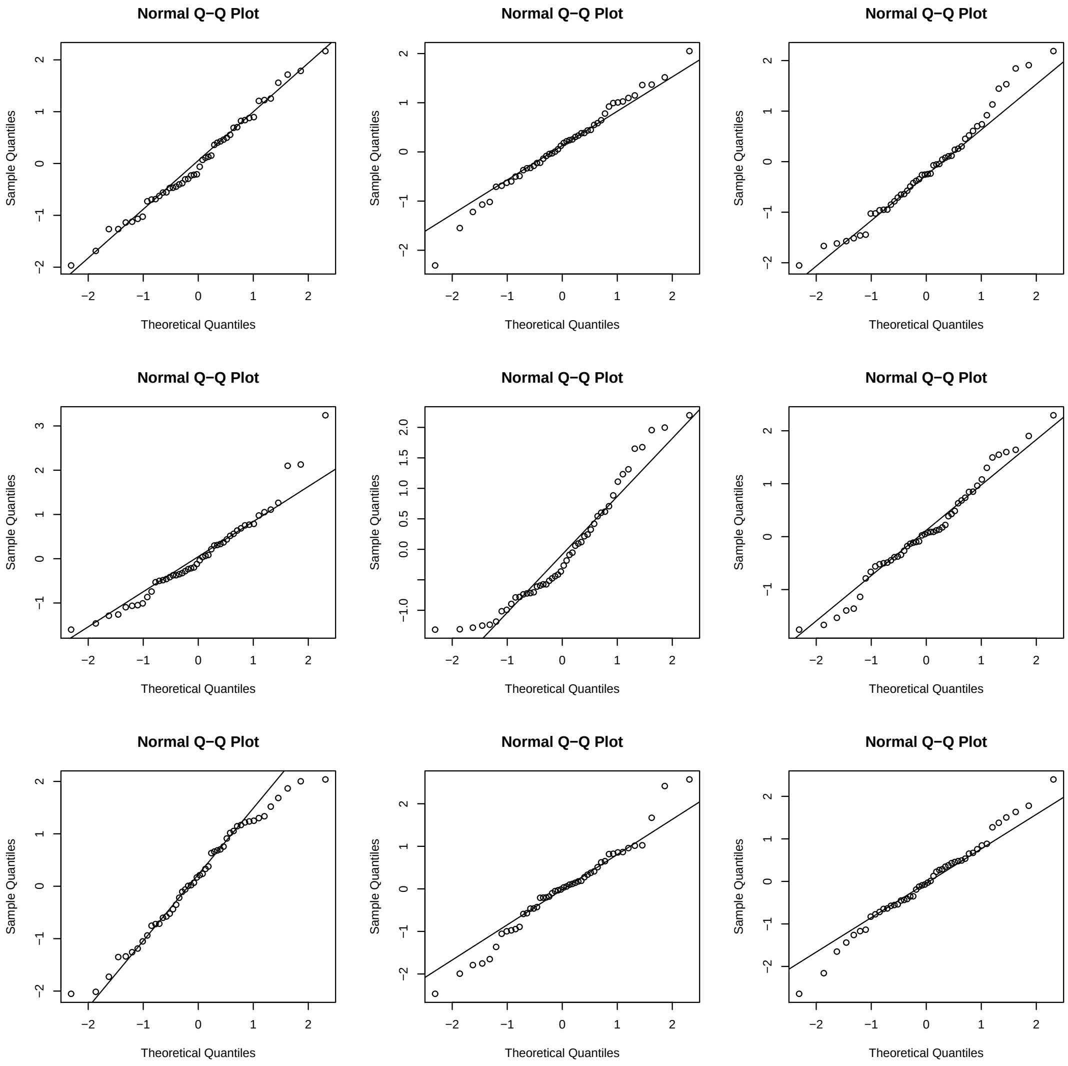
where $i = 1, 2$ *is the method effect and* $j = 1, 2, 3, 4$ *is the lab effect, then we want to estimate* $\frac{1}{4}\sum_{j=1}^{4} \tau_2 - \tau_1 + \delta_{2j} - \delta_{1j}$. *Using the usual R parametrization,* *that is* $\mathbf{c}^T\boldsymbol{\beta}$ *where* $\mathbf{c}^T = (0\ 4\ 0\ 0\ 0\ 1\ 1\ 1)/4$. *A cute trick is to fit the interaction with no method main effect as in* `lm(fat ~ lab + method:lab)`*. You would then get four estimates of the method effect, one for each lab, and could more easily average them. We then build a function to compute the contrast (or mean) for a new sample of 39 rows of data, and bootstrap it. The Bca interval is (-1.354, -1.2), which closely matches the normality based one (see appendix). Alternatively, you could give four CI's, one for each lab, which would be most appropriate if these were the only possible four labs to use.*

2

R Code

First, Here's code to use to see how far really 'normal' data can get from the line:

```
randqqplot <- function(n) {
    y <- rnorm(n)
    qqnorm(y)
    qqline(y)
}
par(mfrow = c(3, 3))
set.seed(123)
for (i in 1:9) randqqplot(48)
```



Only one plot fits the straight line well.

```
soils <- read.table("http://www.math.montana.edu/~jimrc/classes/stat505/data/soils.dat", head = TRUE)
require(xtable, quietly = TRUE)
require(MASS, quietly = TRUE)
require(boot, quietly = TRUE)
require(ggplot2, quietly = TRUE)
require(nlme, quietly = TRUE)
require(lattice, quietly = TRUE)
```

```
# qplot( x=pH, y=Ca, data=soils) + theme_bw() + geom_smooth()
xyplot(Ca ~ pH, data = soils, type = c("p", "smooth"))
xtable(CaFit0 <- lm(Ca ~ pH, data = soils))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-5.8657	0.4546	-12.90	0.0000
pH	1.5826	0.0789	20.05	0.0000

```
calcium.gls0 <- gls(Ca ~ pH, data = soils)
plot(Variogram(calcium.gls0, form = ~column + row, max = 15))
```

```
calcium.glsE <- update(calcium.gls0, corr = corExp(c(8, 0.5), form = ~column + row, nugg = T))
calcium.glsL <- update(calcium.gls0, corr = corLin(c(8, 0.5), form = ~column + row, nugg = T))
calcium.glsS <- update(calcium.gls0, corr = corSpher(c(14, 0.25), form = ~column + row, nugg = T))
calcium.glsG <- update(calcium.gls0, corr = corGaus(c(8, 0.5), form = ~column + row, nugg = T))
calcium.glsR <- update(calcium.gls0, corr = corRatio(c(5, 0.5), form = ~column + row, nugg = T))
myanova <- anova(calcium.gls0, calcium.glsS, calcium.glsE, calcium.glsL, calcium.glsG, calcium.glsR)
myanova$call <- NULL
xtable(myanova)
## summary(calcium.glsS )
```

```
require(gridExtra, quietly = TRUE)
p1 <- plot(calcium.gls0, main = "Naive Model Residuals, No Correlation", type = c("p", "smooth"))
p2 <- plot(calcium.glsS, main = "Spherical Spatial Corr, Pearson Resid", type = c("p", "smooth"))
p3 <- plot(calcium.glsS, resid(., type = "n") ~ fitted(.), main = "Spherical Spatial Corr, Normal Resid", type = c("p",
    "smooth"))
grid.arrange(p1, p2, p3, ncol = 3)
```

```
data(rock)
p1 <- xyplot(log(perm) ~ area, data = rock, type = c("p", "smooth"))
p2 <- xyplot(log(perm) ~ peri, data = rock, type = c("p", "smooth"))
p3 <- xyplot(log(perm) ~ shape, data = rock, type = c("p", "smooth"))
grid.arrange(p1, p2, p3, ncol = 3)
```

```
summary(rock.lm <- lm(log(perm) ~ ., data = rock))$coef

##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.333145   5.49e-01    9.72 1.59e-12
## area         0.000485   8.66e-05    5.60 1.29e-06
## peri        -0.001527   1.77e-04   -8.62 5.24e-11
## shape        1.756526   1.76e+00    1.00 3.23e-01

summary(rock.rlm <- rlm(log(perm) ~ area + shape + peri, data = rock, meth = "MM", maxit = 50))$coef

##                 Value Std. Error t value
## (Intercept)  5.449944   6.00e-01   9.081
## area         0.000498   9.47e-05   5.259
## shape        1.389453   1.92e+00   0.723
## peri        -0.001564   1.94e-04  -8.075

par(mfrow = c(1, 2))
plot(rock.lm, which = 2)
qqnorm(resid(rock.rlm))
qqline(resid(rock.rlm))
```

```
rockcoef <- function(data, i) coef(lm(log(perm) ~ ., data = data[i, ]))
rock.boot <- boot(rock, rockcoef, R = 499)
rockBootCI <- boot.ci(rock.boot, ind = 4, conf = 0.95)

## Warning:  bootstrap variances needed for studentized intervals
```

```
milkfat <- read.table("http://www.math.montana.edu/~jimrc/classes/stat505/data/milkfat.dat", head = T)
milkfat$lab <- factor(milkfat$lab)
qplot(x = lab, y = fat, data = milkfat, colour = method, geom = "boxplot") + theme_bw()
```

```
milk.fit <- lm(fat ~ .^2, milkfat)
xtable(anova(milk.fit))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
method	1	12.08	12.08	522.72	0.0000
lab	3	2.09	0.70	30.08	0.0000
method:lab	3	1.54	0.51	22.19	0.0000
Residuals	31	0.72	0.02		

```
par(mfrow = c(1, 3))
plot(milk.fit, which = 1:3)
```

```
XtXinv <- summary(milk.fit)$cov.unscaled
X <- model.matrix(milk.fit)
Ct <- cbind(matrix(0, 3, 5), diag(3))
varCbetahat <- t(Ct) %*% solve(Ct %*% XtXinv %*% t(Ct)) %*% Ct
middle <- X %*% XtXinv %*% varCbetahat %*% XtXinv %*% t(X)
fakeF <- function(e, i, df) {
    ## e is the vector of residuals from our model i is a sample -- with replacement-- of integers 1 to n=length(e) df
    ## = degrees of freedom for the model
    em <- e[i]  ## bootstrapped sample
    n <- length(e)
    s2 <- var(em)/df * (n - 1)  ## estimated variance
    as.numeric(em %*% middle %*% em/s2)
}
mresid <- resid(milk.fit)/sqrt(1 - hat(X))
bootFs <- rep(0, 499)  ## set up a vector for storage.
set.seed(45678)
for (i in 1:499) bootFs[i] <- fakeF(mresid, sample(39, repl = TRUE), 31)
summary(bootFs)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.01    1.09    2.07    2.58    3.57   12.50

Ctbetahat <- Ct %*% coef(milk.fit)
testF <- t(Ctbetahat) %*% solve(Ct %*% XtXinv %*% t(Ct)) %*% Ctbetahat/summary(milk.fit)$sigma^2
pval1 <- (1 + sum(which(bootFs >= as.numeric(testF))))/500
## Tans method
null.model <- lm(fat ~ method + lab, milkfat)
null.fits <- fitted(null.model)

mresd <- resid(null.model)/sqrt(1 - hat(model.matrix(null.model)))

interactionF <- function(e, i) {
    new.y <- null.fits + e[i]
    anova(lm(new.y ~ method * lab, milkfat))$F[3]
}
bootF2 <- rep(0, 499)
for (i in 1:499) bootF2[i] <- interactionF(mresd, sample(39, repl = TRUE))
summary(bootF2)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.40    0.80    1.06    1.33    5.85

pval2 <- (1 + sum(which(bootF2 >= interactionF(mresd, 1:39))))/500
```

```
summary(milk.fit <- lm(formula = fat ~ lab + method:lab, data = milkfat))
ct <- c(0, 0, 0, 0, 1, 1, 1, 1)/4
```

```r
dim(ct) <- c(1, 8)
center <- ct %*% coef(milk.fit)
stdErr <- sqrt(ct %*% vcov(milk.fit) %*% t(ct))
(normalCI <- center + c(-1, 1) * 1.645 * stdErr)
# -1.360049 -1.189201
mfits <- fitted(milk.fit)
milkfat$ystar <- 1:39 * 0
methodB <- function(data, i) {
    # print(mfits)
    milkfat$ystar <- mfits + data[i]
    ct %*% coef(update(milk.fit, ystar ~ .))
}
methodB(resid(milk.fit), sample(1:39, 39, replace = TRUE))
bootB.CI <- boot(data = resid(milk.fit), methodB, 999)
bca <- boot.ci(bootB.CI, conf = 0.9, type = "bca")$bca  ##  (-1.35, -1.20 )  )
```