

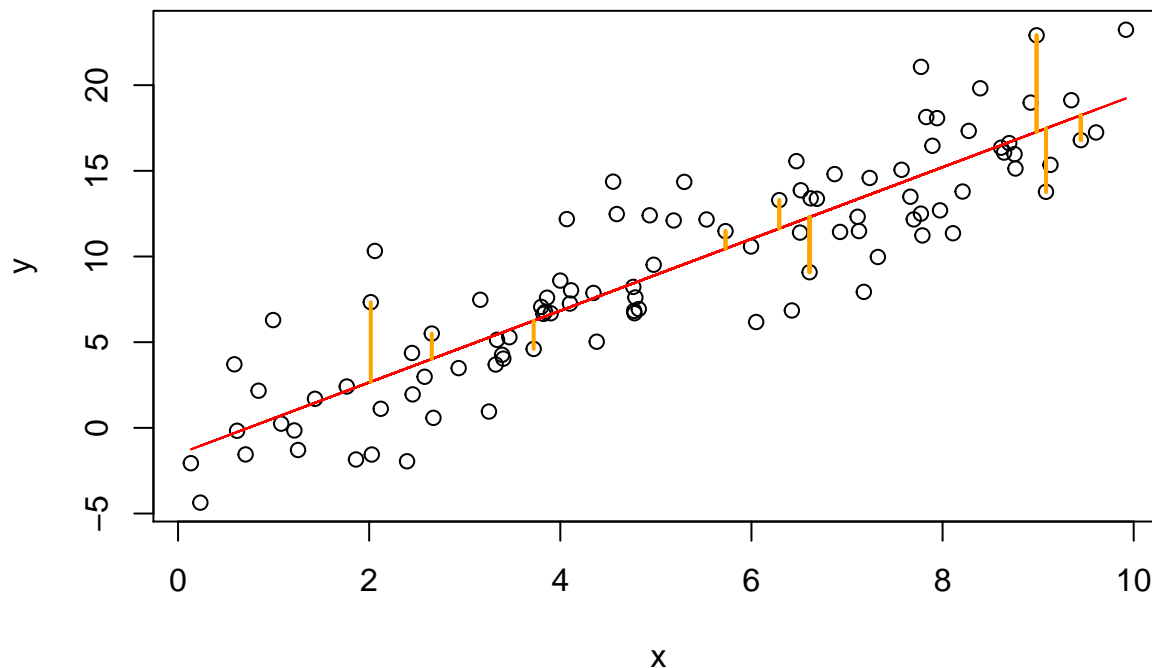
# Linear Models and ANOVA

Xiaonan Da

2021-03-19

## Simple Linear Regression: A Simulated Example

```
# set seed to obtain the same results everytime
set.seed(1)
# generate 100 random values from a Unif(0,10) distribution as independent variables
x <- runif(100,0,10)
# generate dependent variables according to the model y=-1+2x, where errors are normally
# distributed with mean 0 and standard deviation 3
y <- -1+2*x+rnorm(100,0,3)
# fit a simple linear regression model using simulated data
fit <- lm(y~x)
# scatter plot of simulated data
plot(x,y)
# add the line of best fit into the scatter plot
lines(x, predict(fit), col="red")
# sample some points
n <- x[sample(60,20)]
# drop some vertical lines from these points to the line of best fit
segments(x[n], y[n], x[n], fitted(fit)[n], col = "orange", lwd = 2)
```



## Simple Linear Regression: Sea Ice Data

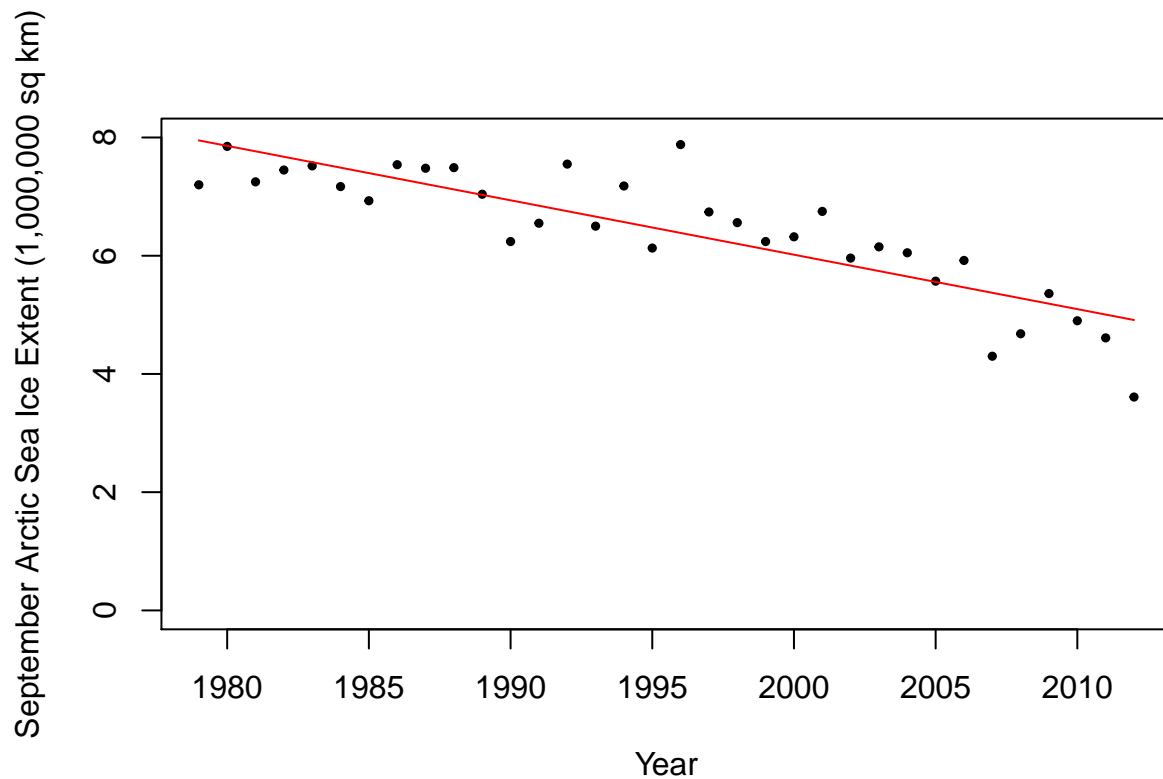
### Fit a Model with Linear Terms

```
# load data
# change file directory to your own path where datasets are saved locally on your device
SeaIce.data <- read.table(file="/Users/Xiaonan/Downloads/MiCM/sea-ice-data.txt")

# define variables
SeaIce <- SeaIce.data$SeaIce
Year <- SeaIce.data$Year

# fit the model
model.seaice <- lm(SeaIce~Year)

# scatter plot of data
plot(Year,SeaIce,
     ylab="September Arctic Sea Ice Extent (1,000,000 sq km)",
     pch=20, cex=0.75,ylim=range(0,8))
# add the fitted line
lines(Year, predict(model.seaice), col="red")
```



```
# model summary
summary(model.seaice)
```

```
##
## Call:
## lm(formula = SeaIce ~ Year)
##
## Residuals:
```

```
##      Min      1Q   Median      3Q      Max
## -1.30259 -0.34064  0.01161  0.36576  1.49456
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 190.12418   20.00964   9.502 7.80e-11 ***
## Year        -0.09205    0.01003  -9.180 1.76e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5736 on 32 degrees of freedom
## Multiple R-squared:  0.7248, Adjusted R-squared:  0.7162
## F-statistic: 84.28 on 1 and 32 DF,  p-value: 1.76e-10
```

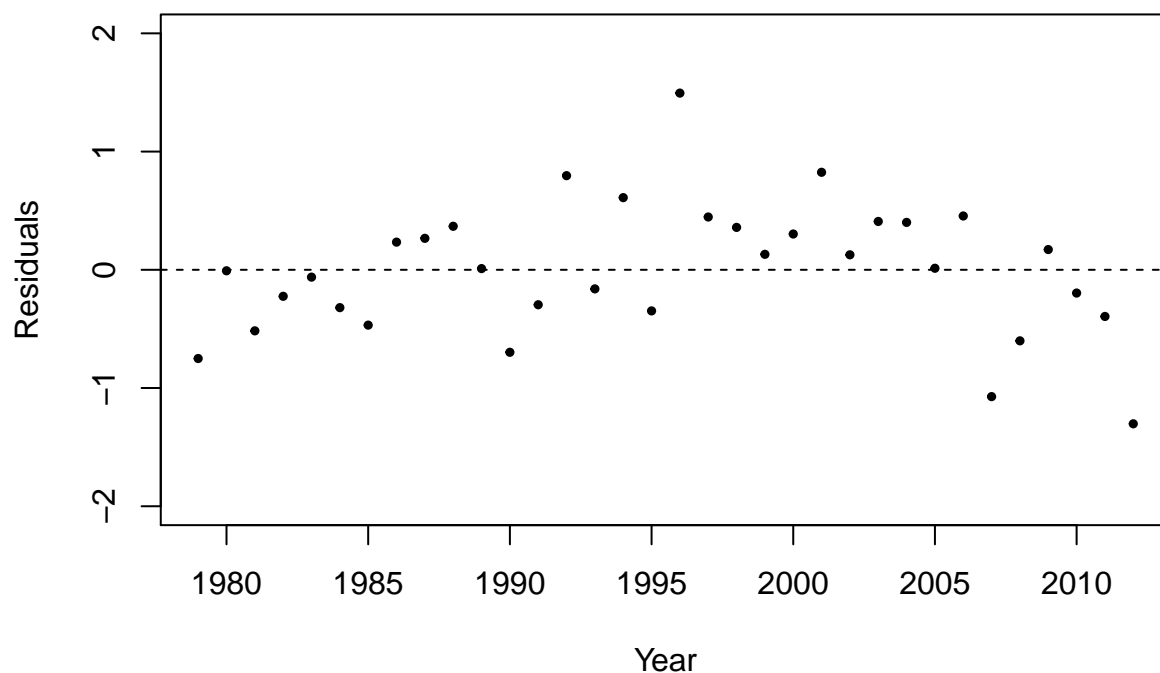
```
# anova of model
anova(model.seaice)
```

```
## Analysis of Variance Table
##
## Response: SeaIce
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Year       1 27.731  27.731   84.278 1.76e-10 ***
## Residuals 32 10.529   0.329
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Residual Analysis

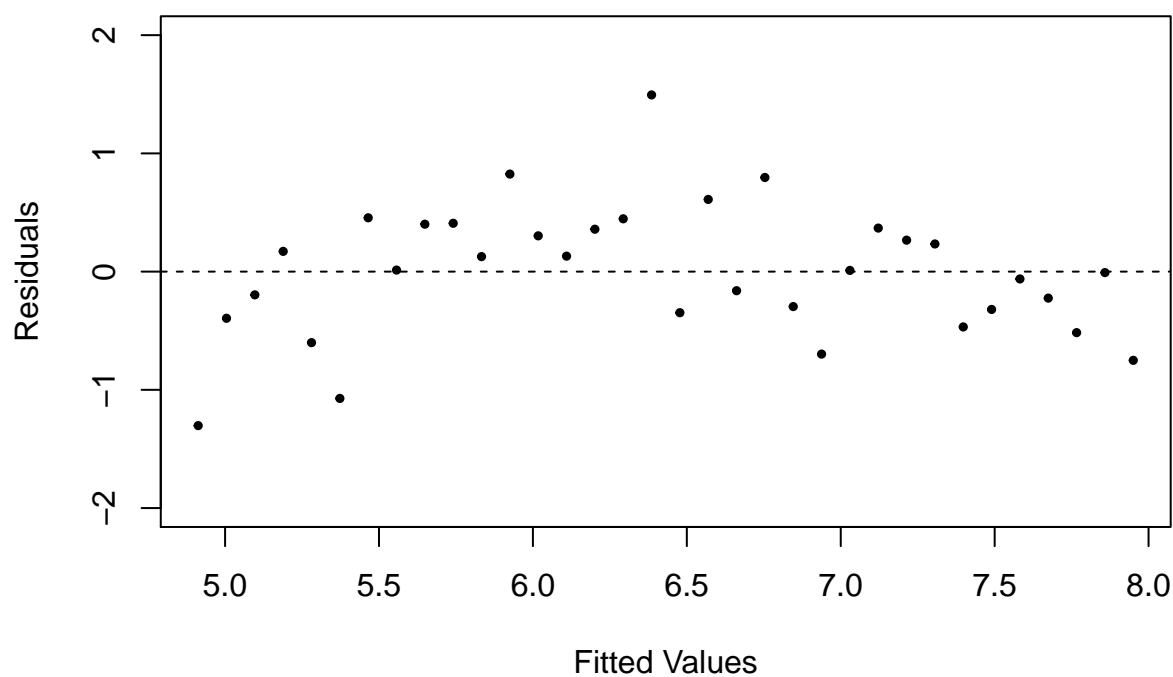
```
# obtain the residuals
residuals.seaice <- residuals(model.seaice)
plot(Year, residuals.seaice, xlab="Year", ylab="Residuals", pch=20,
     cex=0.75, ylim=range(-2, 2), main="Plot of Residuals against Predictor")
abline(h = 0, lty = 2)
```

### Plot of Residuals against Predictor



```
# obtain fitted values
fitted.seaice <- fitted(model.seaice)
plot(fitted.seaice, residuals.seaice, xlab="Fitted Values", ylab="Residuals",
     pch=20, cex=0.75, ylim=range(-2, 2), main="Plot of Residuals against Fitted Values")
abline(h = 0, lty = 2)
```

### Plot of Residuals against Fitted Values



## Fit a Model with Quadratic Terms

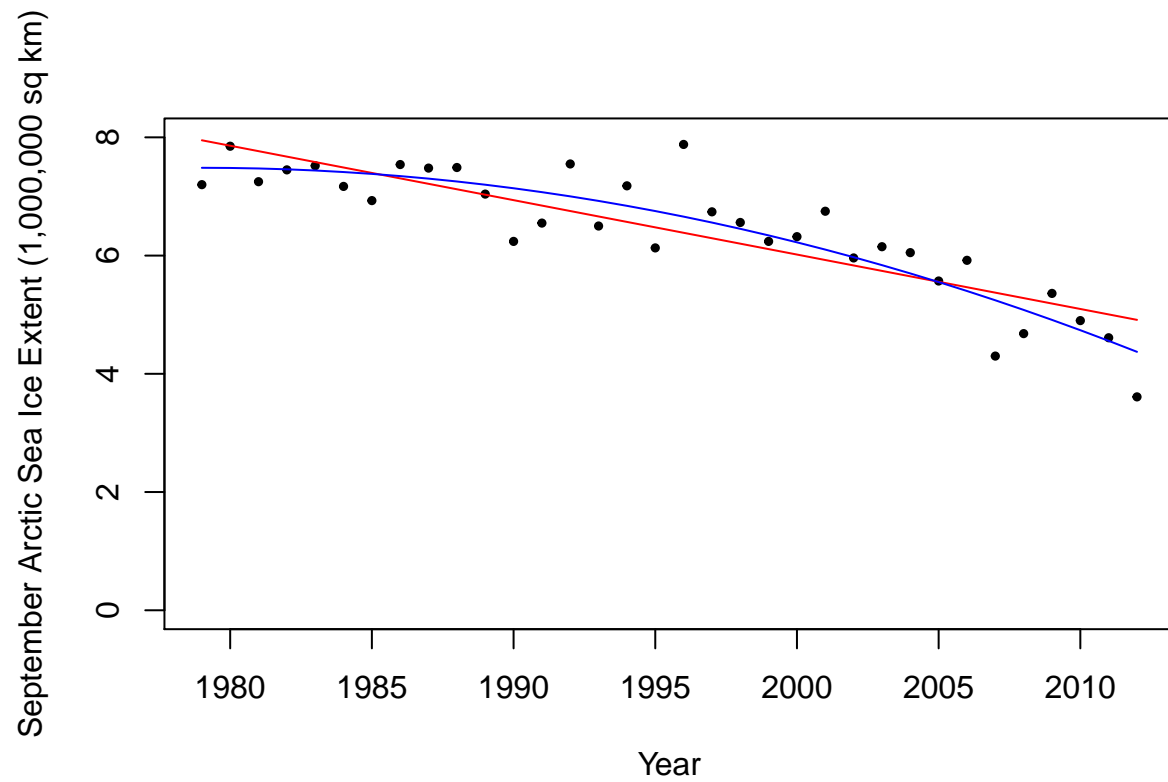
```
# quadratic transform of year variable
Year.transformed <- (Year-1979)^2
# fit the model
model.seaice.quadratic <- lm(SeaIce~Year.transformed)
# model summary
summary(model.seaice.quadratic)

##
## Call:
## lm(formula = SeaIce ~ Year.transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94366 -0.27455  0.06663  0.29295  1.22126
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.4849184   0.1199383   62.41 < 2e-16 ***
## Year.transformed -0.0028587   0.0002408  -11.87 2.92e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4704 on 32 degrees of freedom
## Multiple R-squared:  0.8149, Adjusted R-squared:  0.8091
## F-statistic: 140.9 on 1 and 32 DF,  p-value: 2.921e-13

# anova of model
anova(model.seaice.quadratic)

## Analysis of Variance Table
##
## Response: SeaIce
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Year.transformed  1 31.1785  31.1785  140.89 2.921e-13 ***
## Residuals       32  7.0814   0.2213
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

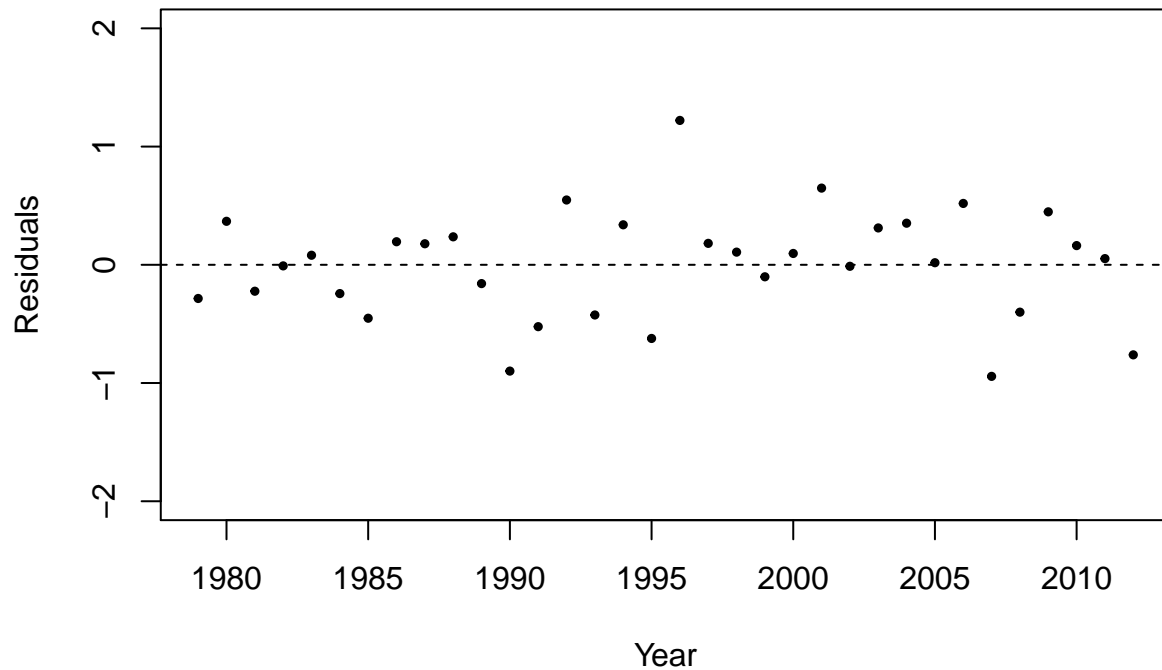
# scatter plot comparing both models
plot(Year,SeaIce,
     ylab="September Arctic Sea Ice Extent (1,000,000 sq km)",
     pch=20, cex=0.75,ylim=range(0,8))
# add the first fitted line
lines(Year, predict(model.seaice), col="red")
# add the second fitted curve
lines(Year, fitted(model.seaice.quadratic), col="blue")
```



### Residual Analysis

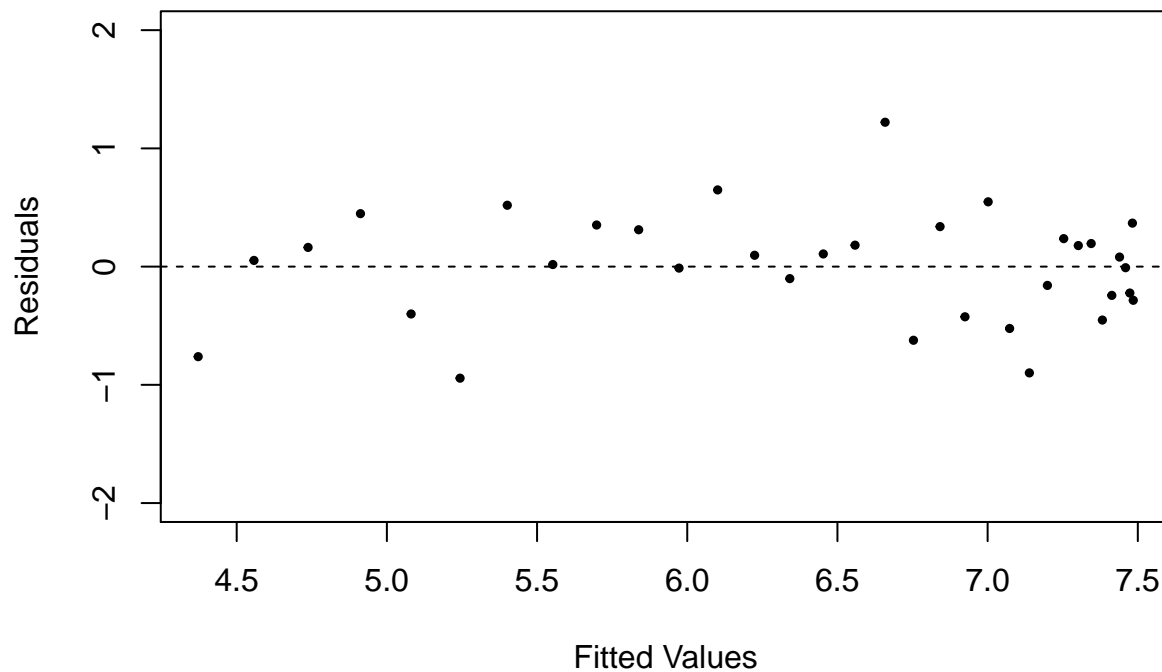
```
# obtain residuals from the second model
residuals.seaice.quadratic <- residuals(model.seaice.quadratic)
plot(Year, residuals.seaice.quadratic, xlab="Year", ylab="Residuals",
     pch=20, cex=0.75, ylim=range(-2, 2), main="Plot of Residuals against Predictor")
abline(h = 0, lty = 2)
```

## Plot of Residuals against Predictor

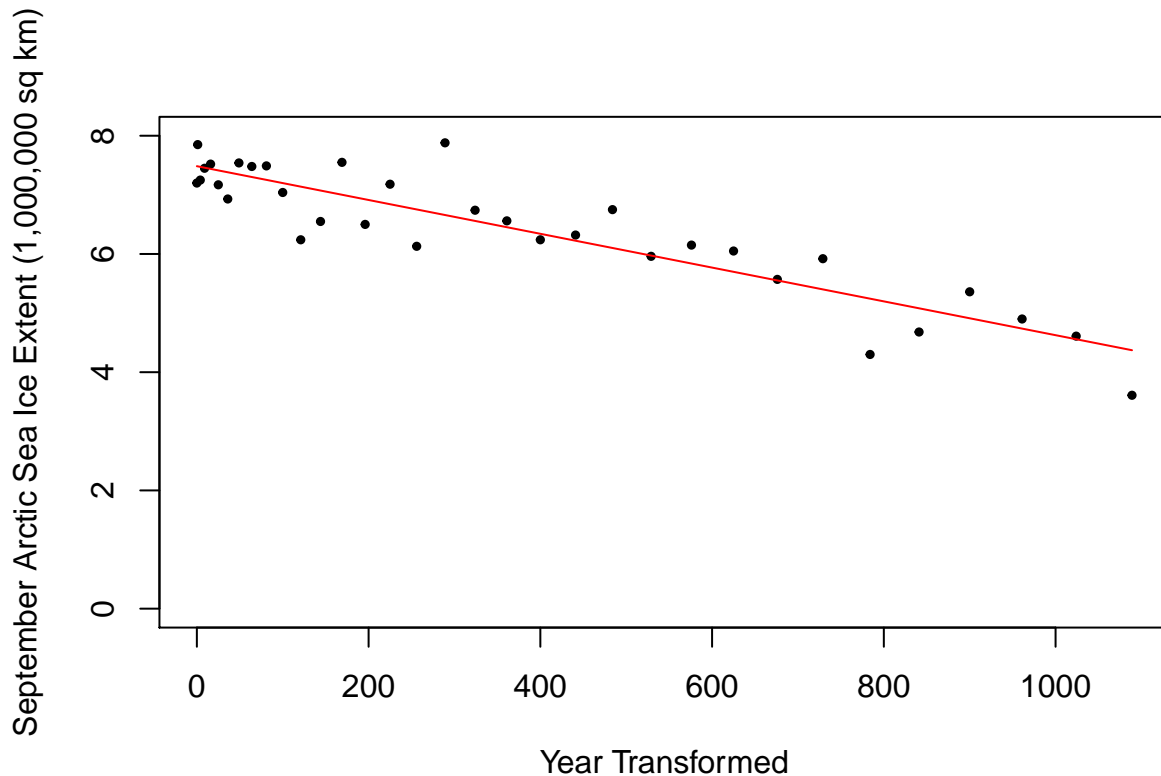


```
# obtain fitted value from the second model
fitted.seaice.quadratic <- fitted(model.seaice.quadratic)
plot(fitted.seaice.quadratic, residuals.seaice.quadratic, xlab="Fitted Values", ylab="Residuals",
     pch=20, cex=0.75, ylim=range(-2, 2), main="Plot of Residuals against Fitted Values")
abline(h = 0, lty = 2)
```

## Plot of Residuals against Fitted Values



```
# scatter plot in the transformed space
plot(Year.transformed,SeaIce,xlab="Year Transformed",
     ylab="September Arctic Sea Ice Extent (1,000,000 sq km)",
     pch=20, cex=0.75,ylim=range(0,8))
# add the line of best fit with transformed year variable
lines(Year.transformed, predict(model.seaice.quadratic), col="red")
```



## Multiple Regression: Blood Viscosity Data

### Fit Models with Different Main Effects

```
# load data
# change file directory to your own path where datasets are saved locally on your device
blood.data <- read.table(file="/Users/Xiaonan/Downloads/MiCM/blood.txt")

# define variables
visc <- blood.data$Viscosity
pcv <- blood.data$PCV
fib <- blood.data$Plasma.Fib.
pro <- blood.data$Plasma.Pro.

# model1 with three main effects
model1 <- lm(visc~pcv+fib+pro)
# model summary
summary(model1)
```

```
##
## Call:
## lm(formula = visc ~ pcv + fib + pro)
```



```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46615 -0.17339  0.02847  0.16932  0.55247
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.3782383  0.8966650  -1.537   0.136
## pcv          0.1168232  0.0136089   8.584 2.5e-09 ***
## fib          0.0004019  0.0003505   1.147   0.261
## pro          0.0400364  0.0971527   0.412   0.683
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3037 on 28 degrees of freedom
## Multiple R-squared:  0.7839, Adjusted R-squared:  0.7607
## F-statistic: 33.86 on 3 and 28 DF,  p-value: 1.876e-09
```

```
# model2 with two main effect
```

```
model2 <- lm(visc~pcv+fib)
```

```
# model summary
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = visc ~ pcv + fib)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48902 -0.19390  0.03387  0.14090  0.57556
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.1030187  0.5896907  -1.871   0.0715 .
## pcv          0.1159823  0.0132611   8.746 1.26e-09 ***
## fib          0.0004042  0.0003454   1.170   0.2514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2993 on 29 degrees of freedom
## Multiple R-squared:  0.7826, Adjusted R-squared:  0.7676
## F-statistic: 52.19 on 2 and 29 DF,  p-value: 2.458e-10
```

```
# model3 with one main effect
```

```
model3 <- lm(visc~fib)
```

```
# model summary
```

```
summary(model3)
```

```
##
## Call:
## lm(formula = visc ~ fib)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1752 -0.3729 -0.1222  0.3512  1.2194
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.8708133  0.2924499  13.236 4.64e-14 ***
## fib         0.0016595  0.0005892   2.816  0.0085 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5613 on 30 degrees of freedom
## Multiple R-squared:  0.2091, Adjusted R-squared:  0.1828
## F-statistic: 7.932 on 1 and 30 DF,  p-value: 0.008504
```

```
# model4 with one main effect
```

```
model4 <- lm(visc~pcv)
```

```
# model summary
```

```
summary(model4)
```

```
##
## Call:
## lm(formula = visc ~ pcv)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48814 -0.22152  0.04686  0.12955  0.64915
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.22336    0.58422  -2.094   0.0448 *
## pcv          0.12243    0.01214  10.088 3.73e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3012 on 30 degrees of freedom
## Multiple R-squared:  0.7723, Adjusted R-squared:  0.7647
## F-statistic: 101.8 on 1 and 30 DF,  p-value: 3.731e-11
```

## Model Selection

```
# Determine if model3 is an adequate simplification of model 1
```

```
anova(model3,model1)
```

```
## Analysis of Variance Table
##
## Model 1: visc ~ fib
## Model 2: visc ~ pcv + fib + pro
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      30 9.4513
## 2      28 2.5825  2     6.8688 37.237 1.293e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Determine if model4 is an adequate simplification of model 1
```

```
anova(model4,model1)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Model 1: visc ~ pcv
```

```
## Model 2: visc ~ pcv + fib + pro
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      30 2.7208
## 2      28 2.5825  2   0.13835 0.75 0.4816

# Compare several nested model together
anova(model4,model2,model1)

## Analysis of Variance Table
##
## Model 1: visc ~ pcv
## Model 2: visc ~ pcv + fib
## Model 3: visc ~ pcv + fib + pro
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      30 2.7208
## 2      29 2.5982  1   0.122691 1.3302 0.2585
## 3      28 2.5825  1   0.015663 0.1698 0.6834

# Compare several nested model together
anova(model3,model2,model1)

## Analysis of Variance Table
##
## Model 1: visc ~ fib
## Model 2: visc ~ pcv + fib
## Model 3: visc ~ pcv + fib + pro
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      30 9.4513
## 2      29 2.5982  1    6.8532 74.3037 2.293e-09 ***
## 3      28 2.5825  1    0.0157  0.1698    0.6834
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

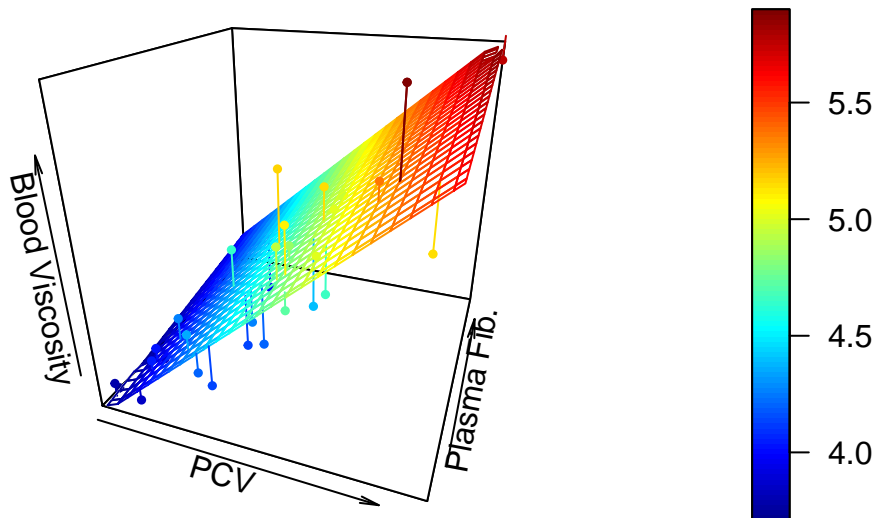
##### How to produce a 3D plot
x <- blood.data$PCV
y <- blood.data$Plasma.Fib.
z <- blood.data$Viscosity
fit <- lm(z ~ x + y)
# predict values on regular xy grid
grid.lines = 26
x.pred <- seq(min(x), max(x), length.out = grid.lines)
y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid( x = x.pred, y = y.pred)
z.pred <- matrix(predict(fit, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)

# instal.packages("plot3D")
library(plot3D)

## Warning in fun(libname, pkgname): couldn't connect to display ":0"

# fitted points for droplines to surface
fitpoints <- predict(fit)
scatter3D(x, y, z, pch = 20, cex = 0.75,
          theta = 20, phi = 25,
          xlab = "PCV", ylab = "Plasma Fib.", zlab = "Blood Viscosity",
          surf = list(x = x.pred, y = y.pred, z = z.pred,
```

```
facets = NA, fit = fitpoints))
```



## One-Way ANOVA: Honeycough Data

```
# load data
# change file directory to your own path where datasets are saved locally on your device
honeycough <- read.csv(file="/Users/Xiaonan/Downloads/MiCM/honeycough.csv")

# define variables
ImproveScore <- honeycough$ImproveScore
Treatment <- honeycough$Treatment

# One-way ANOVA
anova.cough <- aov(ImproveScore ~ Treatment)
summary(anova.cough)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## Treatment      2   318.5    159.2   17.51 2.9e-07 ***
## Residuals    102   927.7      9.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Simple Linear Regression with Treatment as a Factor Predictor
model.cough <- lm(ImproveScore ~ Treatment)
# anova of model
anova(model.cough)

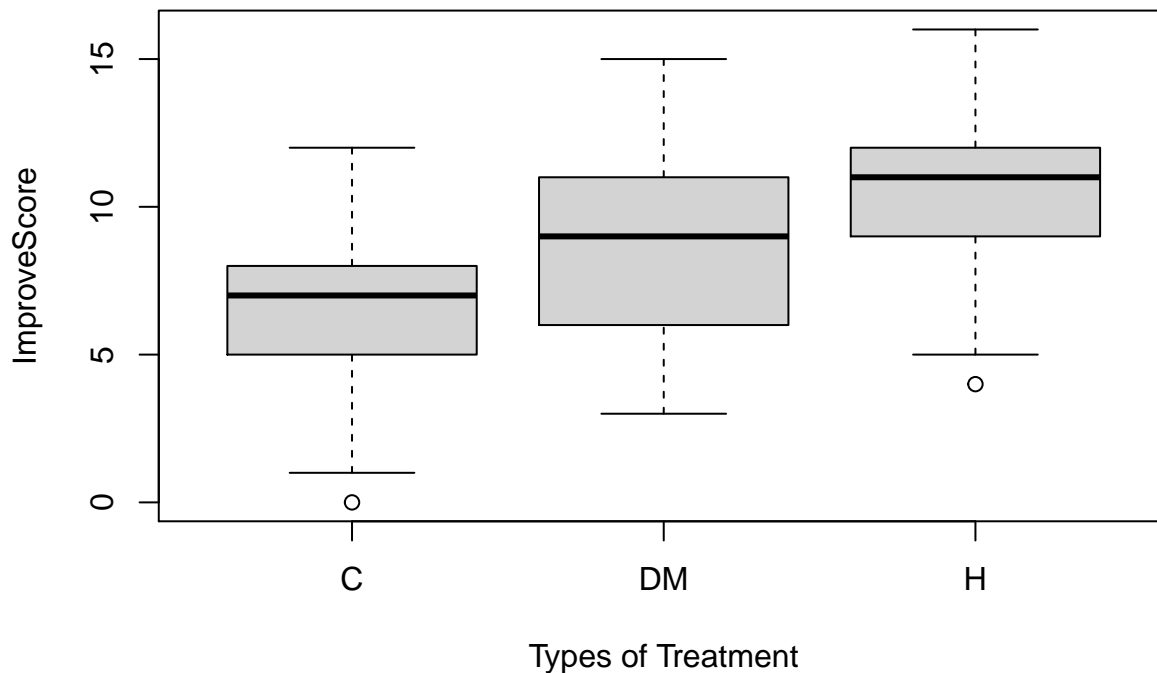
## Analysis of Variance Table
##
## Response: ImproveScore
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Treatment      2 318.51  159.255   17.51 2.902e-07 ***
## Residuals    102  927.72    9.095
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# model summary
summary(model.cough)
```

```
##
## Call:
## lm(formula = ImproveScore ~ Treatment)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.7143 -1.7143  0.4865  1.6667  6.6667
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5135     0.4958  13.137 < 2e-16 ***
## TreatmentDM    1.8198     0.7221   2.520  0.0133 *
## TreatmentH     4.2008     0.7111   5.907 4.62e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.016 on 102 degrees of freedom
## Multiple R-squared:  0.2556, Adjusted R-squared:  0.241
## F-statistic: 17.51 on 2 and 102 DF, p-value: 2.902e-07
```

### Model Assumptions

```
# boxplot by Treatment to check normality and equal variance
boxplot(ImproveScore ~ Treatment, xlab = "Types of Treatment")
```



```
# install.packages("car")
# load package car
library(car)
```

```
## Loading required package: carData
```

```
# Levene's test for checking assumption of equal variance, with mean as center
leveneTest(ImproveScore ~ Treatment, data=honeycough, center="mean")
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = "mean")
##      Df F value Pr(>F)
## group  2  0.9218 0.4011
##      102

# Levene's test for checking assumption of equal variance, with median as center, more robust
leveneTest(ImproveScore ~ Treatment, data=honeycough, center="median")

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = "median")
##      Df F value Pr(>F)
## group  2  0.9169 0.403
##      102
```

## Post Hoc Tests

```
# Tukey HSD multiple comparisons of means
TukeyHSD(anova.cough)

##      Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = ImproveScore ~ Treatment)
##
## $Treatment
##      diff      lwr      upr      p adj
## DM-C 1.819820 0.1023625 3.537277 0.0351562
## H-C  4.200772 2.5094509 5.892094 0.0000001
## H-DM 2.380952 0.6405157 4.121389 0.0043728

# Bonferroni correction
pairwise.t.test(ImproveScore, Treatment, p.adj = "bonf")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: ImproveScore and Treatment
##
##      C      DM
## DM 0.0398 -
## H  1.4e-07 0.0046
##
## P value adjustment method: bonferroni
```

## Randomized Block Design: Dosage Data

```
# load data
# change file directory to your own path where datasets are saved locally on your device
dosage <- read.csv(file="/Users/Xiaonan/Downloads/MiCM/dosage.csv")

# define and format variables
```

```

# Participant_ID as a factor predictor
ID <- as.factor(rep(dosage$Participant_ID,3))
# Dosage as a factor predictor
Dosage <- as.factor(c(rep("Control",5),rep("20mg",5),rep("60mg",5)))
# Response as a numeric vector
Response <- c(dosage$Control,dosage$X20mg,dosage$X60mg)

# anova without considering blocking effect
anova.noID <- aov(Response~Dosage)
summary(anova.noID)

```

```

##              Df Sum Sq Mean Sq F value Pr(>F)
## Dosage        2  276.2   138.1    1.012  0.393
## Residuals    12 1637.8   136.5

```

```

# anova accounting for blocking factor
anova.dosage <- aov(Response ~ Dosage + ID)
summary(anova.dosage)

```

```

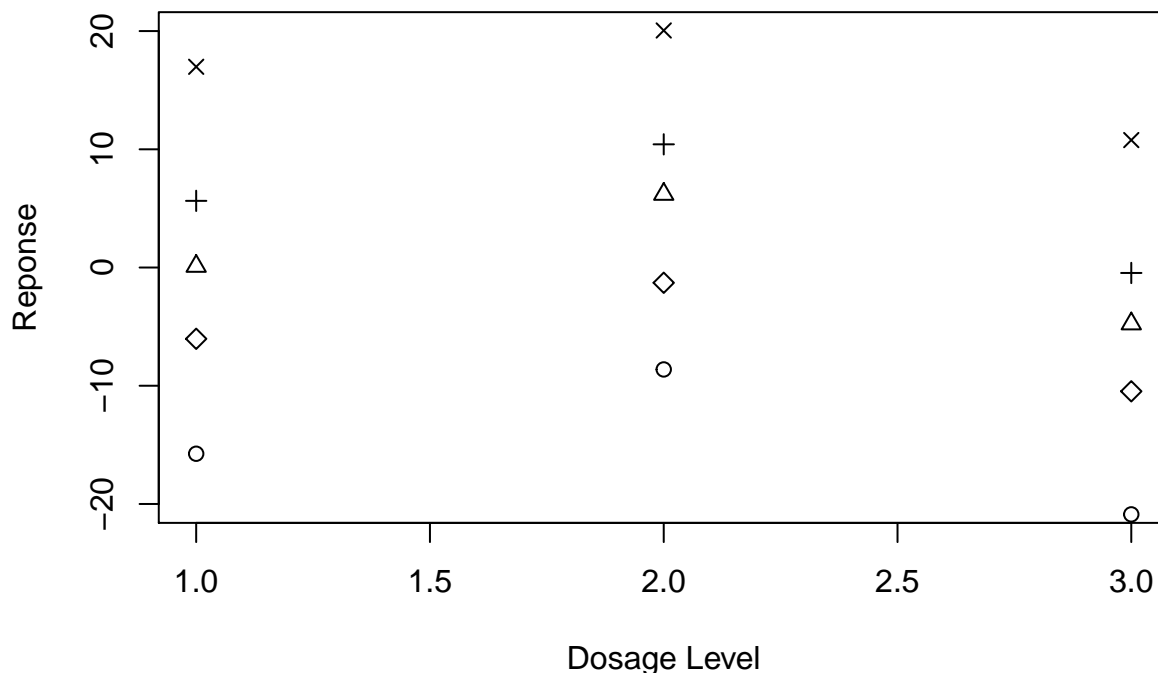
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Dosage        2  276.2   138.1   178.8 2.29e-07 ***
## ID            4 1631.6   407.9   528.2 1.01e-09 ***
## Residuals     8    6.2     0.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# scatter plot showing structure of data
plot(as.numeric(Dosage[ID==1]),Response[ID==1],ylim=c(-20,20),pch=1,
      xlab="Dosage Level", ylab="Reponse")
for (i in 2:5){
  points(as.numeric(Dosage[ID==i]),Response[ID==i],pch=i)}

```



```

# Post Hoc Tests
TukeyHSD(anova.dosage)

```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = Response ~ Dosage + ID)
##
## $Dosage
##          diff          lwr          upr      p adj
## 60mg-20mg    5.162    3.573826  6.750174 3.84e-05
## Control-20mg -5.348   -6.936174 -3.759826 2.96e-05
## Control-60mg -10.510  -12.098174 -8.921826 1.00e-07
##
## $ID
##          diff          lwr          upr      p adj
## 2-1  15.600000  13.121088  18.078912 0.0000001
## 3-1  20.283333  17.804421  22.762246 0.0000000
## 4-1  31.020000  28.541088  33.498912 0.0000000
## 5-1   9.153333   6.674421  11.632246 0.0000096
## 3-2   4.683333   2.204421   7.162246 0.0012339
## 4-2  15.420000  12.941088  17.898912 0.0000001
## 5-2  -6.446667  -8.925579  -3.967754 0.0001307
## 4-3  10.736667   8.257754  13.215579 0.0000029
## 5-3 -11.130000 -13.608912  -8.651088 0.0000022
## 5-4 -21.866667 -24.345579 -19.387754 0.0000000

# multiple linear regression with two factor predictors as main effects
model.dosage <- lm(Response ~ Dosage + ID)
anova(model.dosage)

## Analysis of Variance Table
##
## Response: Response
##          Df Sum Sq Mean Sq F value    Pr(>F)
## Dosage     2  276.18   138.09   178.81 2.292e-07 ***
## ID         4 1631.62   407.91   528.18 1.009e-09 ***
## Residuals   8    6.18    0.77
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(model.dosage)

##
## Call:
## lm(formula = Response ~ Dosage + ID)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1107 -0.4897 -0.0040  0.4187  1.2393
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -15.0213     0.6003  -25.022 6.96e-09 ***
## Dosage60mg     5.1620     0.5558   9.287 1.47e-05 ***
## DosageControl -5.3480     0.5558  -9.622 1.13e-05 ***
## ID2           15.6000     0.7175  21.741 2.11e-08 ***
## ID3           20.2833     0.7175  28.268 2.65e-09 ***
```



```
## ID4          31.0200      0.7175  43.231 9.04e-11 ***
## ID5          9.1533      0.7175  12.757 1.34e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8788 on 8 degrees of freedom
## Multiple R-squared:  0.9968, Adjusted R-squared:  0.9944
## F-statistic: 411.7 on 6 and 8 DF,  p-value: 1.62e-09
```

## Factorial Design: ToothGrowth Data

```
# anova for two factors and their interaction
anova.tooth <- aov(len ~ supp*as.factor(dose), data = ToothGrowth)
summary(anova.tooth)

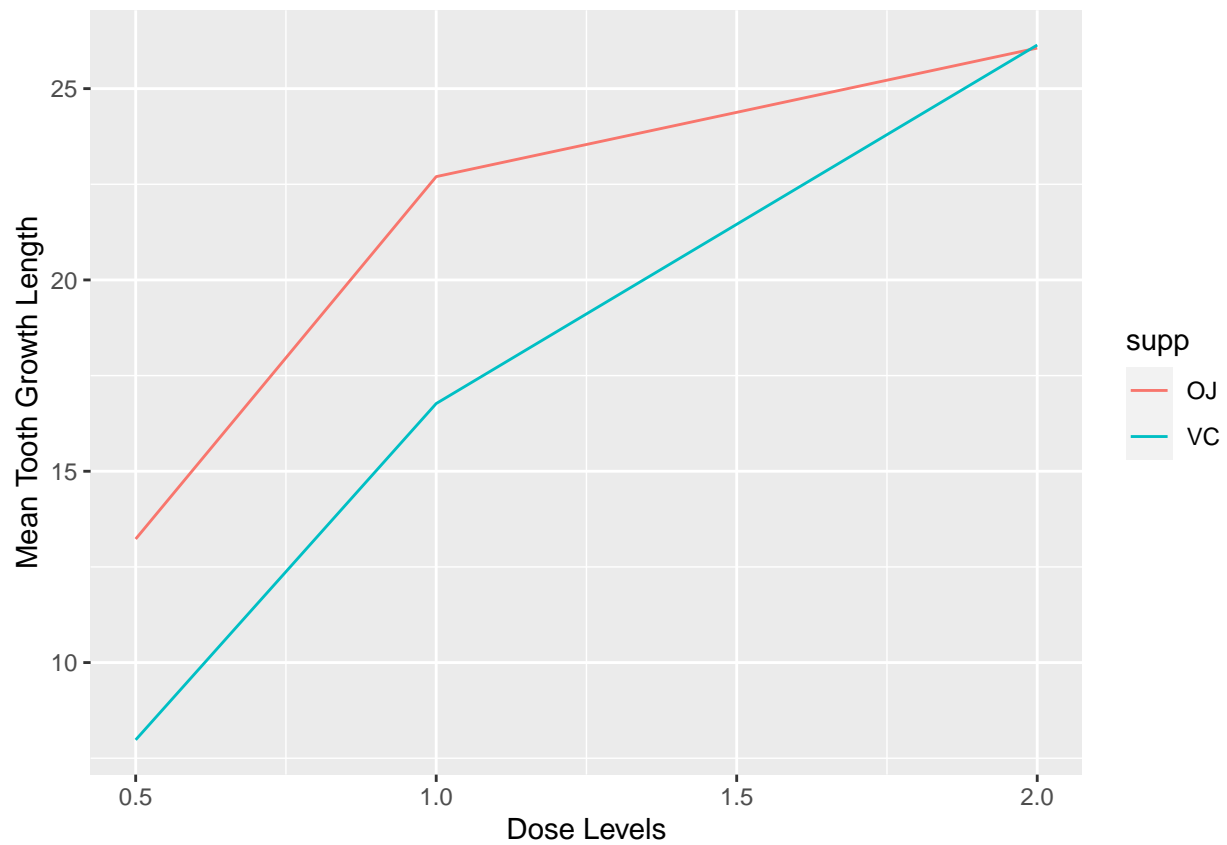
##              Df Sum Sq Mean Sq F value    Pr(>F)
## supp          1  205.4    205.4   15.572 0.000231 ***
## as.factor(dose) 2 2426.4   1213.2   92.000 < 2e-16 ***
## supp:as.factor(dose) 2  108.3     54.2    4.107 0.021860 *
## Residuals      54  712.1     13.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# fit a multiple regression model with two main effects and interaction term
model.tooth.full <- lm(len ~ supp*as.factor(dose), data = ToothGrowth)
# anova of model
anova(model.tooth.full)

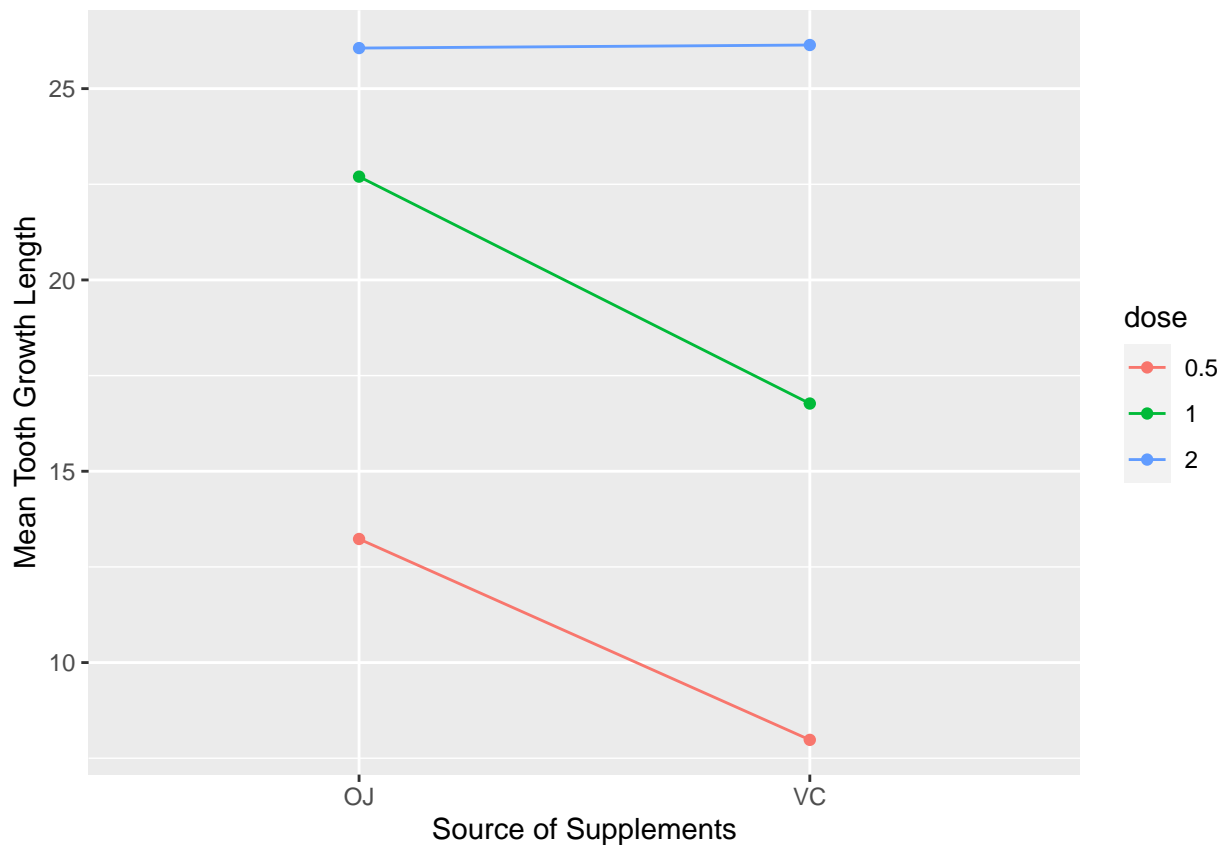
## Analysis of Variance Table
##
## Response: len
##              Df Sum Sq Mean Sq F value    Pr(>F)
## supp          1  205.35    205.35   15.572 0.0002312 ***
## as.factor(dose) 2 2426.43   1213.22   92.000 < 2.2e-16 ***
## supp:as.factor(dose) 2  108.32     54.16    4.107 0.0218603 *
## Residuals      54  712.11     13.19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# install.packages("emmeans")
# install.packages("ggplot2")
# load packages emmeans and ggplot2
library(emmeans)
library(ggplot2)

# plot of estimated marginal means by type of supplements
emmip(model.tooth.full, supp~as.factor(dose), type="response",
       xlab = "Dose Levels", ylab = "Mean Tooth Growth Length")
```



```
# plot of estimated marginal means by dose levels  
emmip(model.tooth.full, as.factor(dose)~supp, type="response",  
       xlab = "Source of Supplements", ylab = "Mean Tooth Growth Length")
```



```
# Levene's test for checking assumption of equal variance, with mean as center
leveneTest(len ~ supp*as.factor(dose), data = ToothGrowth, center = "mean")
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##      Df F value Pr(>F)
## group 5  1.9401 0.1027
##      54
```

```
# Levene's test for checking assumption of equal variance, with median as center
leveneTest(len ~ supp*as.factor(dose), data = ToothGrowth, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##      Df F value Pr(>F)
## group 5  1.7086 0.1484
##      54
```

```
# fit a multiple regression model with only main effects
model.tooth <- lm(len ~ supp + as.factor(dose), data = ToothGrowth)
summary(model.tooth)
```

```
##
## Call:
## lm(formula = len ~ supp + as.factor(dose), data = ToothGrowth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.085  -2.751  -0.800   2.446   9.650
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12.4550     0.9883  12.603 < 2e-16 ***
## suppVC         -3.7000     0.9883  -3.744 0.000429 ***
## as.factor(dose)1  9.1300     1.2104   7.543 4.38e-10 ***
## as.factor(dose)2 15.4950     1.2104  12.802 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.828 on 56 degrees of freedom
## Multiple R-squared:  0.7623, Adjusted R-squared:  0.7496
## F-statistic: 59.88 on 3 and 56 DF,  p-value: < 2.2e-16
```

```
# anova of model
anova(model.tooth)
```

```
## Analysis of Variance Table
##
## Response: len
##               Df Sum Sq Mean Sq F value    Pr(>F)
## supp           1  205.35   205.35   14.017 0.0004293 ***
## as.factor(dose) 2 2426.43 1213.22   82.811 < 2.2e-16 ***
## Residuals      56  820.43    14.65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# compare the two models and determine if interaction is a significnat term
anova(model.tooth,model.tooth.full)
```

```
## Analysis of Variance Table
##
## Model 1: len ~ supp + as.factor(dose)
## Model 2: len ~ supp * as.factor(dose)
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      56 820.43
## 2      54 712.11  2    108.32 4.107 0.02186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# compare if we did not treat dose as a factor predictor
model.dose <- lm(len ~ supp + dose, data = ToothGrowth)
summary(model.dose)
```

```
##
## Call:
## lm(formula = len ~ supp + dose, data = ToothGrowth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.600 -3.700  0.373  2.116  8.800
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.2725     1.2824   7.231 1.31e-09 ***
## suppVC         -3.7000     1.0936  -3.383  0.0013 **
## dose           9.7636     0.8768  11.135 6.31e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.236 on 57 degrees of freedom
## Multiple R-squared:  0.7038, Adjusted R-squared:  0.6934
## F-statistic: 67.72 on 2 and 57 DF,  p-value: 8.716e-16
```

## Example: Therapy Data

```
therapy.data <- read.csv(file="/Users/Xiaonan/Downloads/MiCM/therapy.csv")
therapy.data <- therapy.data[,2:5]
```

```
Response <- therapy.data$Y
Therapy <- therapy.data$X1
Dose <- therapy.data$X2
```

```
anova.therapy1 <- aov(Response ~ Therapy*Dose)
summary(anova.therapy1)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Therapy        3     231    76.97    3.148 0.02663 *
## Dose           2     351   175.29    7.169 0.00104 **
## Therapy:Dose    6      100    16.73    0.684 0.66259
## Residuals     164    4010    24.45
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova.therapy2 <- aov(Response ~ Therapy + Dose)
summary(anova.therapy2)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Therapy        3     231    76.97    3.183 0.025341 *
## Dose           2     351   175.29    7.250 0.000952 ***
## Residuals     170    4110    24.18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.therapy1 <- lm(Response ~ Therapy*Dose)
summary(model.therapy1)
```

```
##
## Call:
## lm(formula = Response ~ Therapy * Dose)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2673  -3.3374   0.2859   3.3652  11.5876
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.4300     1.0790  -0.399  0.69078
## TherapyTh2       4.2900     2.0544   2.088  0.03833 *
## TherapyTh3       4.4970     1.8998   2.367  0.01910 *
## TherapyTh4       2.2982     1.6133   1.425  0.15617
## Dose20mg         2.8524     1.3510   2.111  0.03626 *
## Dose60mg         5.1823     1.7450   2.970  0.00343 **
## TherapyTh2:Dose20mg -1.4516     2.6304  -0.552  0.58180
```

```
## TherapyTh3:Dose20mg -4.0021      2.4291 -1.648  0.10135
## TherapyTh4:Dose20mg -0.0780      2.0195 -0.039  0.96924
## TherapyTh2:Dose60mg -0.7190      3.7751 -0.190  0.84919
## TherapyTh3:Dose60mg -4.6613      3.2219 -1.447  0.14987
## TherapyTh4:Dose60mg  0.8135      3.0616  0.266  0.79081
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.945 on 164 degrees of freedom
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.08801
## F-statistic: 2.535 on 11 and 164 DF,  p-value: 0.005563
```

```
anova(model.therapy1)
```

```
## Analysis of Variance Table
##
## Response: Response
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Therapy        3  230.9   76.968   3.1479 0.026630 *
## Dose            2   350.6  175.294   7.1693 0.001035 **
## Therapy:Dose    6   100.4   16.730   0.6842 0.662594
## Residuals     164 4009.9   24.451
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.therapy2 <- lm(Response ~ Therapy + Dose)
summary(model.therapy2)
```

```
##
## Call:
## lm(formula = Response ~ Therapy + Dose)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1680  -3.6171   0.2184   3.4142  11.8784
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.1952     0.8102   0.241  0.809912
## TherapyTh2      3.3969     1.1813   2.875  0.004550 **
## TherapyTh3      1.6864     1.0712   1.574  0.117263
## TherapyTh4      2.3124     0.9021   2.563  0.011237 *
## Dose20mg        1.9364     0.8305   2.332  0.020894 *
## Dose60mg        4.3750     1.1721   3.733  0.000258 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.917 on 170 degrees of freedom
## Multiple R-squared:  0.1239, Adjusted R-squared:  0.09817
## F-statistic:  4.81 on 5 and 170 DF,  p-value: 0.0003869
```

```
anova(model.therapy2)
```

```
## Analysis of Variance Table
##
## Response: Response
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Therapy      3  230.9   76.968   3.1834 0.0253412 *
## Dose         2  350.6  175.294   7.2501 0.0009516 ***
## Residuals 170 4110.3   24.178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Example: Skin Cancer Data

```
skincancer.data <- read.table(file="/Users/Xiaonan/Downloads/MiCM/skincancer.txt")
lat <- skincancer.data$Lat
long <- skincancer.data$Long
ocean <- as.factor(skincancer.data$Ocean)
mort <- skincancer.data$Mort

class(ocean)
```

```
## [1] "factor"
```

```
model1 <- lm(mort~lat*long*ocean)
summary(model1)
```

```
##
## Call:
## lm(formula = mort ~ lat * long * ocean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.754 -10.851  -2.836   12.341   42.632
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    839.43876   400.96872    2.094   0.0425 *
## lat           -17.91942    10.00669   -1.791   0.0807 .
## long           -4.73779     3.98163   -1.190   0.2409
## ocean1        -634.86074   453.55586   -1.400   0.1691
## lat:long         0.12319     0.09923    1.241   0.2215
## lat:ocean1       16.45551    11.18922    1.471   0.1490
## long:ocean1       6.71813     4.62751    1.452   0.1542
## lat:long:ocean1  -0.16804     0.11363   -1.479   0.1468
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.8 on 41 degrees of freedom
## Multiple R-squared:  0.7842, Adjusted R-squared:  0.7474
## F-statistic: 21.29 on 7 and 41 DF,  p-value: 8.785e-12
```

```
model2 <- lm(mort~lat+long+ocean)
summary(model2)
```

```
##
## Call:
## lm(formula = mort ~ lat + long + ocean)
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -30.171 -10.070  -2.607   8.781  41.805
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 349.2369    27.0596  12.906 < 2e-16 ***
## lat         -5.4950     0.5289 -10.390 1.55e-13 ***
## long          0.1219     0.1732   0.704 0.485245
## ocean1       21.7976     5.2263   4.171 0.000137 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.48 on 45 degrees of freedom
## Multiple R-squared:  0.7721, Adjusted R-squared:  0.7569
## F-statistic: 50.83 on 3 and 45 DF,  p-value: 1.7e-14
```

```
model3 <- lm(mort~lat+ocean)
summary(model3)
```

```
##
## Call:
## lm(formula = mort ~ lat + ocean)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -30.435 -11.328  -1.884   9.327  44.372
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 360.6905    21.4984  16.778 < 2e-16 ***
## lat         -5.4888     0.5259 -10.437 1.03e-13 ***
## ocean1       20.4304     4.8251   4.234 0.000109 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.39 on 46 degrees of freedom
## Multiple R-squared:  0.7696, Adjusted R-squared:  0.7596
## F-statistic: 76.83 on 2 and 46 DF,  p-value: 2.169e-15
```

```
anova(model3,model2)
```

```
## Analysis of Variance Table
##
## Model 1: mort ~ lat + ocean
## Model 2: mort ~ lat + long + ocean
##   Res.Df  RSS Df Sum of Sq    F Pr(>F)
## 1      46 12357
## 2      45 12222  1    134.5 0.4952 0.4852
```

```
model4 <- lm(mort~lat*ocean)
summary(model4)
```

```
##
## Call:
## lm(formula = mort ~ lat * ocean)
##
```



```

## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.457 -11.307  -1.881   9.318  44.377
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 360.549460  35.498266  10.157 3.19e-13 ***
## lat         -5.485286   0.874315  -6.274 1.22e-07 ***
## ocean1      20.650094  43.987853   0.469  0.641
## lat:ocean1  -0.005534   1.101391  -0.005  0.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.57 on 45 degrees of freedom
## Multiple R-squared:  0.7696, Adjusted R-squared:  0.7543
## F-statistic: 50.11 on 3 and 45 DF,  p-value: 2.171e-14
anova(model3,model4)

## Analysis of Variance Table
##
## Model 1: mort ~ lat + ocean
## Model 2: mort ~ lat * ocean
##   Res.Df    RSS Df Sum of Sq  F Pr(>F)
## 1      46 12357
## 2      45 12357  1 0.0069323  0  0.996

```