Lesson 08

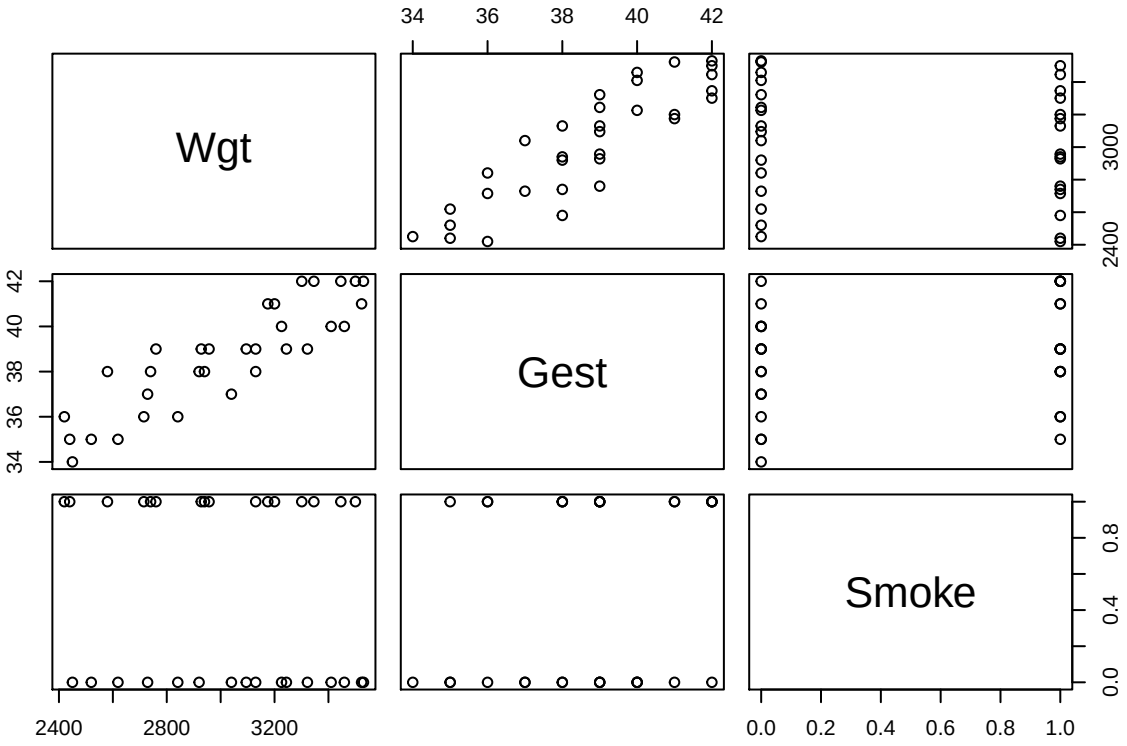
Christopher A. Swenson (chris@cswenson.com)

11/27/2021

Birthweight and smoking (2-level categorical predictor, additive model)

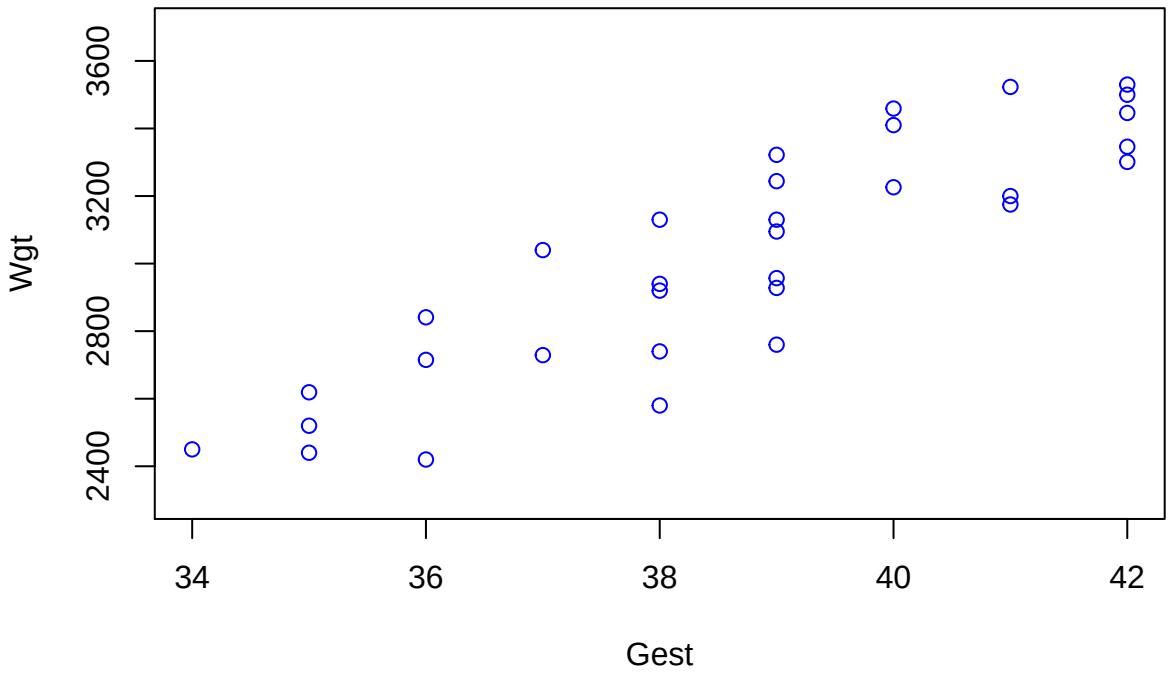
Load the birthsmokers data. Create a scatterplot matrix of the data Fit a multiple linear regression model of Wgt on Gest + Smoke. Display scatterplot of Wgt vs Gest with points marked by Smoke and add parallel regression lines representing Smoke=0 and Smoke=1. Display regression results and calculate confidence intervals for the regression parameters. Display confidence intervals for expected Wgt at Gest=38 (for Smoke=1 and Smoke=0). Repeat analysis separately for Smoke=0 and Smoke=1. Repeat analysis using (1, -1) coding.

```
birthsmokers <- read.table("./Data/birthsmokers.txt", header=T)
attach(birthsmokers)

pairs(cbind(Wgt, Gest, Smoke))
```




```
model <- lm(Wgt ~ Gest + Smoke)

plot(x=Gest, y=Wgt, ylim=c(2300, 3700),
     col=ifelse(Smoke=="yes", "red", "blue"),
     panel.last = c(lines(sort(Gest[Smoke=="no"]),
                          fitted(model)[Smoke=="no"][order(Gest[Smoke=="no"])],
```

```
      col="blue"),
lines(sort(Gest[Smoke=="yes"]),
      fitted(model)[Smoke=="yes"][order(Gest[Smoke=="yes"])],
      col="red")))
```




```
summary(model)
```

```
##
## Call:
## lm(formula = Wgt ~ Gest + Smoke)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -223.693  -92.063   -9.365   79.663  197.507
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2389.573    349.206  -6.843 1.63e-07 ***
## Gest          143.100      9.128  15.677 1.07e-15 ***
## Smoke        -244.544     41.982  -5.825 2.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115.5 on 29 degrees of freedom
## Multiple R-squared:  0.8964, Adjusted R-squared:  0.8892
## F-statistic: 125.4 on 2 and 29 DF,  p-value: 5.289e-15
```

```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) -2389.573    349.206  -6.843 1.63e-07 ***
# Gest          143.100      9.128  15.677 1.07e-15 ***
# Smokeyes     -244.544     41.982  -5.825 2.58e-06 ***
# ---
# Residual standard error: 115.5 on 29 degrees of freedom
```

```
# Multiple R-squared:  0.8964,  Adjusted R-squared:  0.8892
# F-statistic: 125.4 on 2 and 29 DF,  p-value: 5.289e-15

confint(model)
```

```
##                  2.5 %     97.5 %
## (Intercept) -3103.7795 -1675.3663
## Gest          124.4312   161.7694
## Smoke        -330.4064  -158.6817
```

```
#                  2.5 %     97.5 %
# (Intercept) -3103.7795 -1675.3663
# Gest          124.4312   161.7694
# Smokeyes     -330.4064  -158.6817

predict(model, interval="confidence",
        newdata=data.frame(Gest=c(38, 38), Smoke=c(1, 0)))
```

```
##        fit      lwr      upr
## 1 2803.693 2740.599 2866.788
## 2 3048.237 2989.120 3107.355
```

```
#        fit      lwr      upr
# 1 2803.693 2740.599 2866.788
# 2 3048.237 2989.120 3107.355

model.0 <- lm(Wgt ~ Gest, subset=Smoke==0)
summary(model.0)
```

```
##
## Call:
## lm(formula = Wgt ~ Gest, subset = Smoke == 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -171.52 -101.59   23.28   83.63  139.48
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2546.14     457.29  -5.568 6.93e-05 ***
## Gest          147.21      11.97  12.294 6.85e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 106.9 on 14 degrees of freedom
## Multiple R-squared:  0.9152, Adjusted R-squared:  0.9092
## F-statistic: 151.1 on 1 and 14 DF,  p-value: 6.852e-09
```

```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -2546.14     457.29  -5.568 6.93e-05 ***
# Gest          147.21      11.97  12.294 6.85e-09 ***

predict(model.0, interval="confidence",
        newdata=data.frame(Gest=38))
```

```
##        fit      lwr     upr
```

```
## 1 3047.724 2990.298 3105.15
```

```
#        fit      lwr     upr
# 1 3047.724 2990.298 3105.15

model.1 <- lm(Wgt ~ Gest, subset=Smoke==1)
summary(model.1)
```

```
##
## Call:
## lm(formula = Wgt ~ Gest, subset = Smoke == 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -228.53  -64.86  -19.10   93.89  184.53
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2474.56     553.97  -4.467 0.000532 ***
## Gest          139.03      14.11   9.851 1.12e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 126.6 on 14 degrees of freedom
## Multiple R-squared:  0.8739, Adjusted R-squared:  0.8649
## F-statistic: 97.04 on 1 and 14 DF,  p-value: 1.125e-07
```

```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) -2474.56     553.97  -4.467 0.000532 ***
# Gest          139.03      14.11   9.851 1.12e-07 ***

predict(model.1, interval="confidence",
        newdata=data.frame(Gest=38))
```

```
##        fit      lwr      upr
## 1 2808.528 2731.726 2885.331
```

```
#        fit      lwr      upr
# 1 2808.528 2731.726 2885.331

Smoke2 <- ifelse(Smoke==1, 1, -1)
model.3 <- lm(Wgt ~ Gest + Smoke2)
summary(model.3)
```

```
##
## Call:
## lm(formula = Wgt ~ Gest + Smoke2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -223.693  -92.063   -9.365   79.663  197.507
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2511.845    353.449  -7.107 8.07e-08 ***
## Gest          143.100      9.128  15.677 1.07e-15 ***
```

```
## Smoke2       -122.272     20.991  -5.825 2.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115.5 on 29 degrees of freedom
## Multiple R-squared:  0.8964, Adjusted R-squared:  0.8892
## F-statistic: 125.4 on 2 and 29 DF,  p-value: 5.289e-15
```

```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) -2511.845    353.449  -7.107 8.07e-08 ***
# Gest          143.100      9.128  15.677 1.07e-15 ***
# Smoke2       -122.272     20.991  -5.825 2.58e-06 ***

# Alternatively
#model.3 <- lm(Wgt ~ Gest + Smoke, contrasts=list(Smoke="contr.sum"))
model.3 <- lm(Wgt ~ Gest + Smoke)
summary(model.3)
```

```
##
## Call:
## lm(formula = Wgt ~ Gest + Smoke)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -223.693  -92.063   -9.365   79.663  197.507
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2389.573    349.206  -6.843 1.63e-07 ***
## Gest          143.100      9.128  15.677 1.07e-15 ***
## Smoke        -244.544     41.982  -5.825 2.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 115.5 on 29 degrees of freedom
## Multiple R-squared:  0.8964, Adjusted R-squared:  0.8892
## F-statistic: 125.4 on 2 and 29 DF,  p-value: 5.289e-15
```

```
detach(birthsmokers)
```

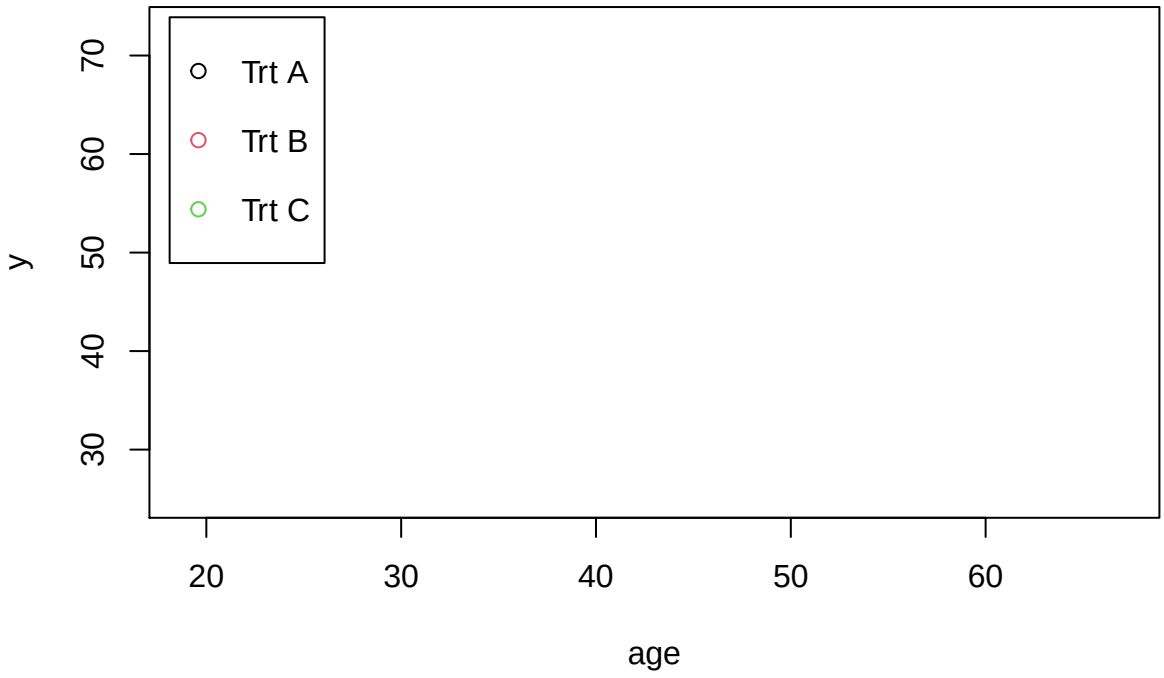
Depression treatments (3-level categorical predictor, interaction model)

Load the depression data. Display scatterplot of y (treatment effectiveness) vs age with points marked by treatment. Create interaction variables and fit a multiple linear regression model of y on age + x2 + x3 + age.x2 + age.x3. Add non-parallel regression lines representing each of the three treatments to the scatterplot. Display a residuals vs fits plot and a normal probability plot of the residuals, and conduct an Anderson-Darling normality test using the nortest package. Conduct an F-test to see if at least one of x2, x3, age.x2, and age.x3 are useful (i.e., the regression functions differ). Conduct an F-test to see if at least one of age.x2 and age.x3 are useful (i.e., the regression functions have different slopes).

```
depression <- read.table("./Data/depression.txt", header=T)
attach(depression)

plot(x=age, y=y, col=as.numeric(TRT))
```

```
## Warning in plot.xy(xy, type, ...): NAs introduced by coercion
```

```
legend("topleft", col=1:3, pch=1,
       inset=0.02, x.intersp = 1.5, y.intersp = 1.8,
       legend=c("Trt A", "Trt B", "Trt C"))
```



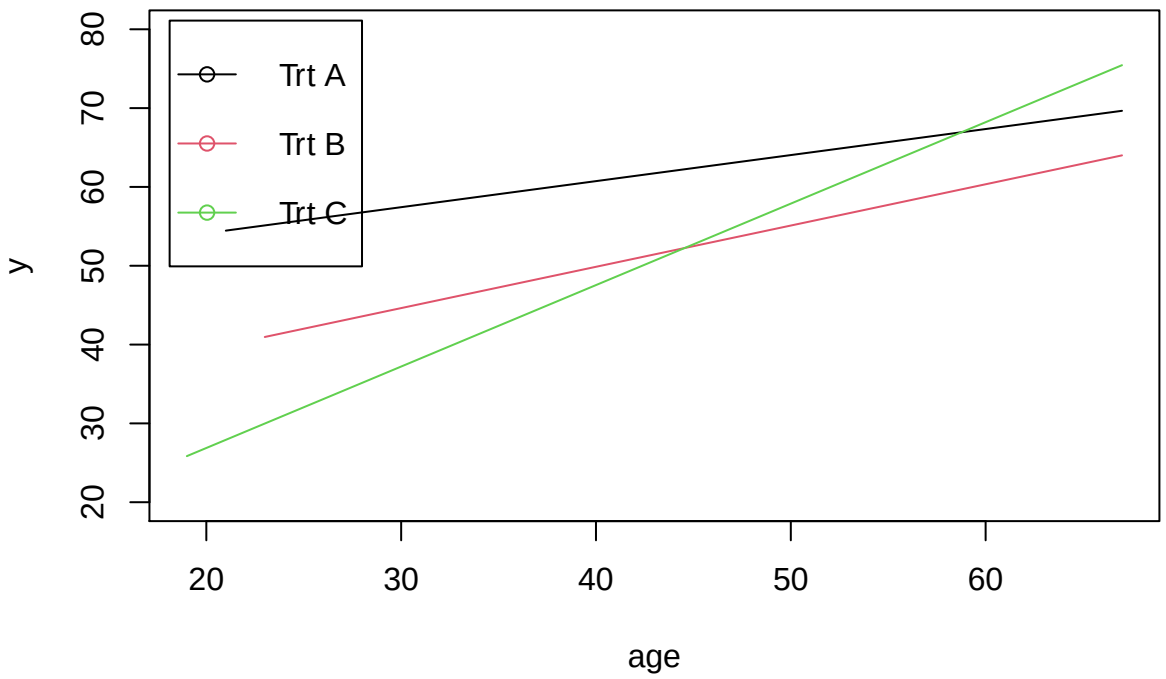

```
age.x2 <- age*x2
age.x3 <- age*x3

model.1 <- lm(y ~ age + x2 + x3 + age.x2 + age.x3)
summary(model.1)
```

```
##
## Call:
## lm(formula = y ~ age + x2 + x3 + age.x2 + age.x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4366 -2.7637  0.1887  2.9075  6.5634
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.21138    3.34964   1.854 0.073545 .
## age          1.03339    0.07233  14.288 6.34e-15 ***
## x2          41.30421    5.08453   8.124 4.56e-09 ***
## x3          22.70682    5.09097   4.460 0.000106 ***
## age.x2      -0.70288    0.10896  -6.451 3.98e-07 ***
## age.x3      -0.50971    0.11039  -4.617 6.85e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.925 on 30 degrees of freedom
## Multiple R-squared:  0.9143, Adjusted R-squared:  0.9001
## F-statistic: 64.04 on 5 and 30 DF,  p-value: 4.264e-15
```

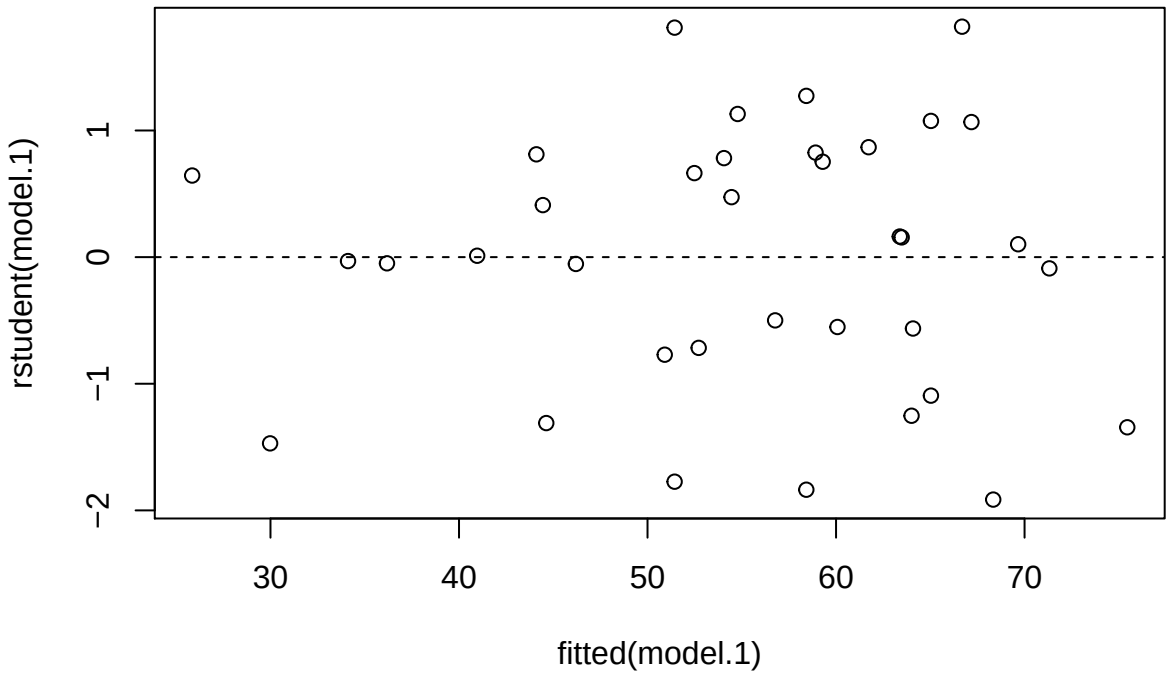
```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  6.21138    3.34964   1.854 0.073545 .
# age          1.03339    0.07233  14.288 6.34e-15 ***
# x2          41.30421    5.08453   8.124 4.56e-09 ***
# x3          22.70682    5.09097   4.460 0.000106 ***
# age.x2      -0.70288    0.10896  -6.451 3.98e-07 ***
# age.x3      -0.50971    0.11039  -4.617 6.85e-05 ***

plot(x=age, y=y, ylim=c(20, 80), col=as.numeric(TRT),
     panel.last = c(lines(sort(age[TRT=="A"]),
                          fitted(model.1)[TRT=="A"][order(age[TRT=="A"])],
                          col=1),
                    lines(sort(age[TRT=="B"]),
                          fitted(model.1)[TRT=="B"][order(age[TRT=="B"])],
                          col=2),
                    lines(sort(age[TRT=="C"]),
                          fitted(model.1)[TRT=="C"][order(age[TRT=="C"])],
                          col=3)))
```

```
## Warning in plot.xy(xy, type, ...): NAs introduced by coercion
```

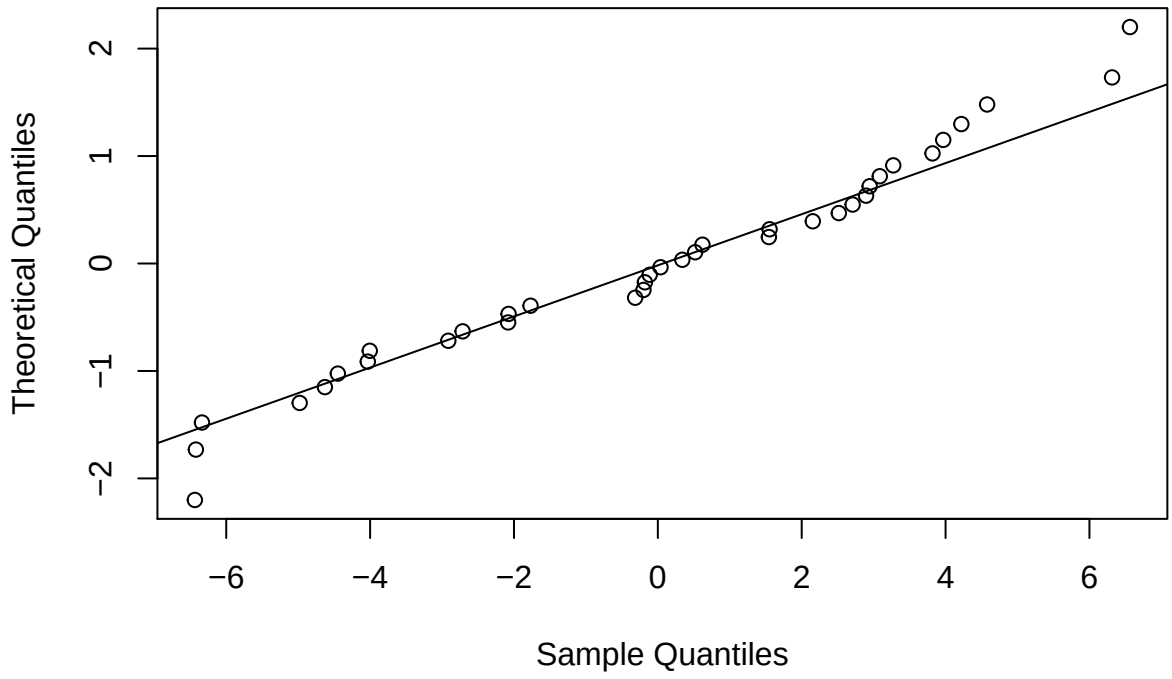
```
legend("topleft", col=1:3, pch=1, lty=1,
       inset=0.02, x.intersp = 1.5, y.intersp = 1.8,
       legend=c("Trt A", "Trt B", "Trt C"))
```




```
plot(x=fitted(model.1), y=rstudent(model.1),
     panel.last = abline(h=0, lty=2))
```




```
qqnorm(residuals(model.1), main="", datax=TRUE)
qqline(residuals(model.1), datax=TRUE)
```




```
library(nortest)
ad.test(residuals(model.1)) # A = 0.4057, p-value = 0.3345
```

```
##
##  Anderson-Darling normality test
##
## data:  residuals(model.1)
## A = 0.40575, p-value = 0.3345
```

```r
anova(model.1)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## age        1 3424.4  3424.4 222.2946 2.059e-15 ***
## x2         1  803.8   803.8  52.1784 4.857e-08 ***
## x3         1    1.2     1.2   0.0772    0.7831
## age.x2     1  375.0   375.0  24.3430 2.808e-05 ***
## age.x3     1  328.4   328.4  21.3194 6.850e-05 ***
## Residuals 30  462.1    15.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#           Df Sum Sq Mean Sq  F value    Pr(>F)
# age        1 3424.4  3424.4 222.2946 2.059e-15 ***
# x2         1  803.8   803.8  52.1784 4.857e-08 ***
# x3         1    1.2     1.2   0.0772    0.7831
# age.x2     1  375.0   375.0  24.3430 2.808e-05 ***
# age.x3     1  328.4   328.4  21.3194 6.850e-05 ***
# Residuals 30  462.1    15.4

model.2 <- lm(y ~ age)
anova(model.2, model.1)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ age
## Model 2: y ~ age + x2 + x3 + age.x2 + age.x3
##   Res.Df     RSS Df Sum of Sq     F    Pr(>F)
## 1     34 1970.57
## 2     30  462.15  4    1508.4 24.48 4.458e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Model 1: y ~ age
# Model 2: y ~ age + x2 + x3 + age.x2 + age.x3
#   Res.Df     RSS Df Sum of Sq     F    Pr(>F)
# 1     34 1970.57
# 2     30  462.15  4    1508.4 24.48 4.458e-09 ***

model.3 <- lm(y ~ age + x2 + x3)
anova(model.3, model.1)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ age + x2 + x3
## Model 2: y ~ age + x2 + x3 + age.x2 + age.x3
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     32 1165.57
## 2     30  462.15  2    703.43 22.831 9.41e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model 1: y ~ age + x2 + x3
# Model 2: y ~ age + x2 + x3 + age.x2 + age.x3
#   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
# 1     32 1165.57
# 2     30  462.15  2    703.43 22.831 9.41e-07 ***

detach(depression)
```

Real estate air conditioning (2-level categorical predictor, interaction model, transformations)

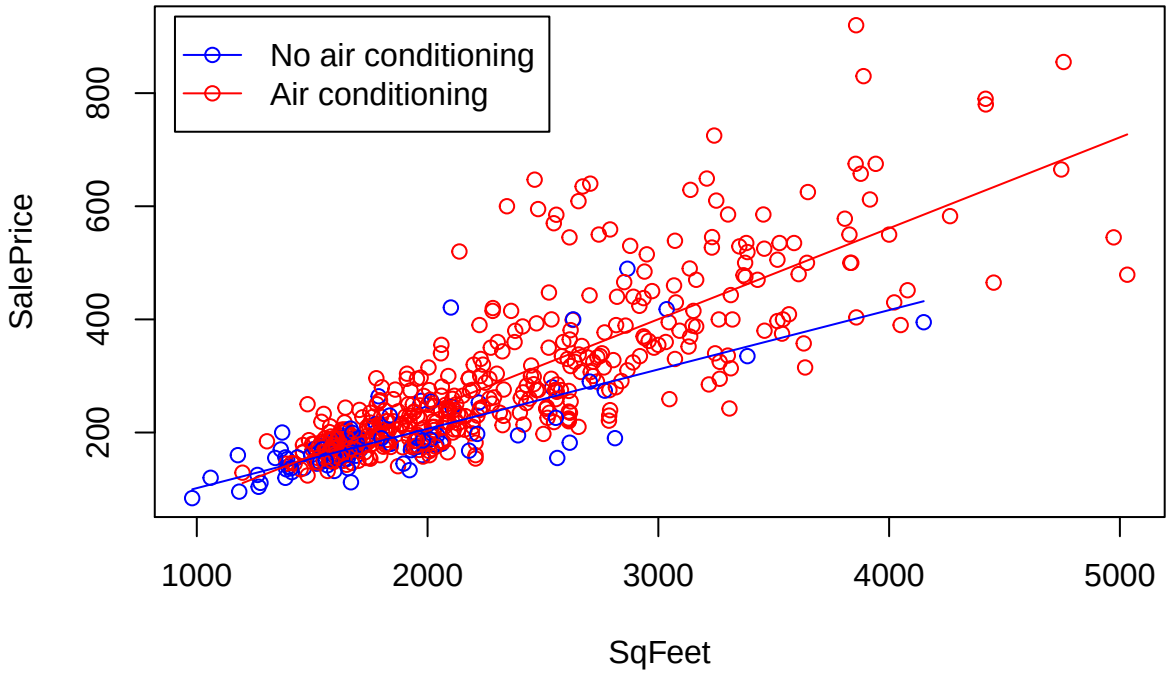
Load the realestate data. Create an interaction variable and fit a multiple linear regression model of SalePrice on SqFeet + Air + SqFeet.Air. Display scatterplot of SalePrice vs SqFeet with points marked by Air and add non-parallel regression lines representing Air=0 and Air=1. Display residual plot with fitted (predicted) values on the horizontal axis. Create log(SalePrice), log(SqFeet), and log(SqFeet).Air variables and fit a multiple linear regression model of log(SalePrice) on log(SqFeet) + Air + log(SqFeet).Air. Display scatterplot of log(SalePrice) vs log(SqFeet) with points marked by Air and add non-parallel regression lines representing Air=0 and Air=1. Display residual plot with fitted (predicted) values on the horizontal axis.

```
realestate <- read.table("./Data/realestate_sales.txt", header=T)
attach(realestate)

SqFeet.Air <- SqFeet*Air
model.1 <- lm(SalePrice ~ SqFeet + Air + SqFeet.Air)
summary(model.1)
```

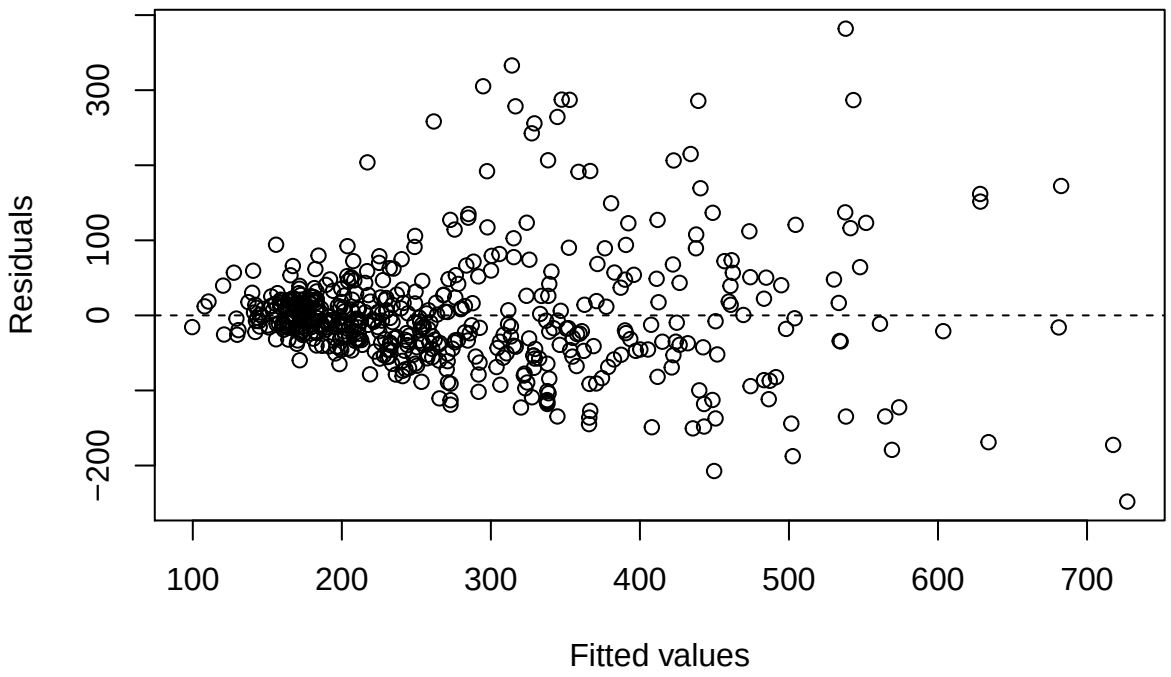
```
##
## Call:
## lm(formula = SalePrice ~ SqFeet + Air + SqFeet.Air)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -248.01  -37.13   -7.80   22.25  381.92
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.21755   30.08504  -0.107 0.914871
## SqFeet        0.10490    0.01575   6.661 6.96e-11 ***
## Air         -78.86783   32.66333  -2.415 0.016100 *
## SqFeet.Air    0.05589    0.01658   3.371 0.000805 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 77.01 on 517 degrees of freedom
## Multiple R-squared:  0.6887, Adjusted R-squared:  0.6869
## F-statistic: 381.2 on 3 and 517 DF,  p-value: < 2.2e-16
```

```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)   -3.218     30.085  -0.107 0.914871
# SqFeet       104.902     15.748   6.661 6.96e-11 ***
# Air          -78.868     32.663  -2.415 0.016100 *
# SqFeet.Air    55.888     16.580   3.371 0.000805 ***

plot(x=SqFeet, y=SalePrice,
```

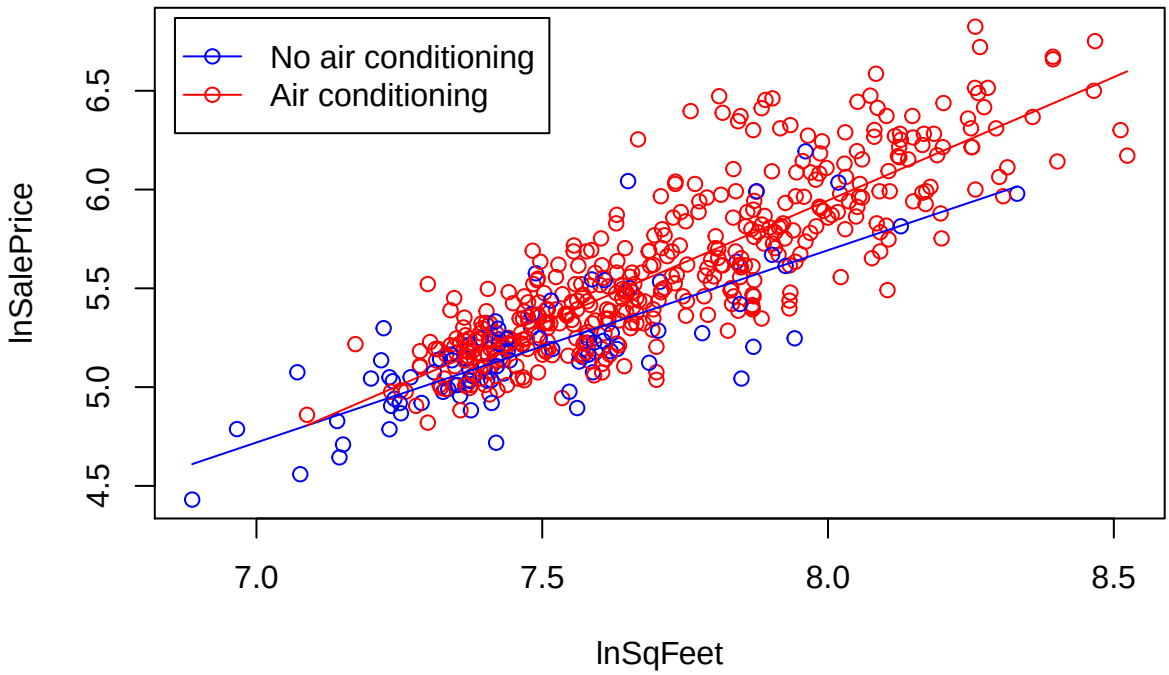
```
     col=ifelse(Air, "red", "blue"),
     panel.last = c(lines(sort(SqFeet[Air==0]),
                          fitted(model.1)[Air==0][order(SqFeet[Air==0])],
                          col="blue"),
                    lines(sort(SqFeet[Air==1]),
                          fitted(model.1)[Air==1][order(SqFeet[Air==1])],
                          col="red")))
legend("topleft", col=c("blue", "red"), pch=1, lty=1, inset=0.02,
       legend=c("No air conditioning", "Air conditioning"))
```



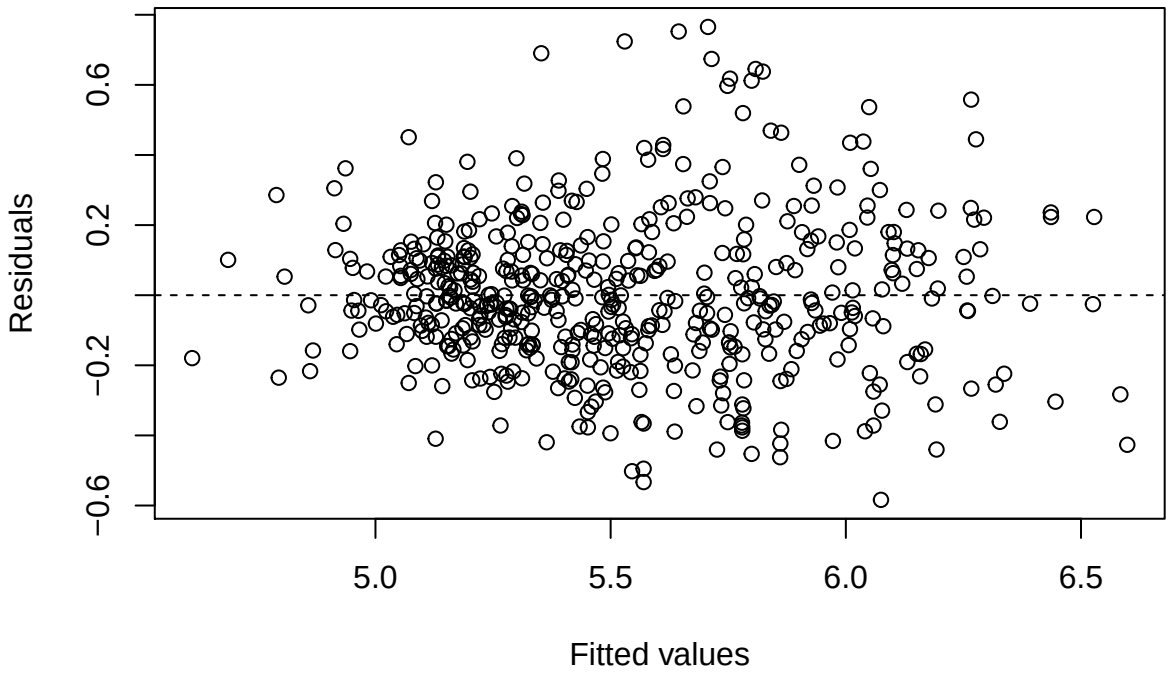

```
plot(x=fitted(model.1), y=residuals(model.1),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```




```
lnSalePrice <- log(SalePrice)
lnSqFeet <- log(SqFeet)
lnSqFeet.Air <- lnSqFeet*Air

model.2 <- lm(lnSalePrice ~ lnSqFeet + Air + lnSqFeet.Air)

plot(x=lnSqFeet, y=lnSalePrice,
     col=ifelse(Air, "red", "blue"),
     panel.last = c(lines(sort(lnSqFeet[Air==0]),
                          fitted(model.2)[Air==0][order(lnSqFeet[Air==0])],
                          col="blue"),
                    lines(sort(lnSqFeet[Air==1]),
                          fitted(model.2)[Air==1][order(lnSqFeet[Air==1])],
                          col="red")))
legend("topleft", col=c("blue", "red"), pch=1, lty=1, inset=0.02,
       legend=c("No air conditioning", "Air conditioning"))
```




```
plot(x=fitted(model.2), y=residuals(model.2),
     xlab="Fitted values", ylab="Residuals",
     panel.last = abline(h=0, lty=2))
```




```
detach(realestate)
```

Hospital infection risk (4-level categorical predictor, additive model)

Load the infectionrisk data and select observations with Stay <= 14. Create indicator variables for regions. Fit a multiple linear regression model of InfctRsk on Stay + Xray + i2 + i3 + i4. Conduct an F-test to see if at least one of i2, i3, and i4 are useful (conclusion: the regression functions differ by region). Conduct an

F-test to see if at least one of i2 and i3 are useful (conclusion: only the west region differs).

```
infectionrisk <- read.table("./Data/infectionrisk.txt", header=T)
infectionrisk <- infectionrisk[infectionrisk$Stay<=14,]
attach(infectionrisk)

i1 <- ifelse(Region==1,1,0) # NE
i2 <- ifelse(Region==2,1,0) # NC
i3 <- ifelse(Region==3,1,0) # S
i4 <- ifelse(Region==4,1,0) # W

model.1 <- lm(InfctRsk ~ Stay + Xray + i2 + i3 + i4)
summary(model.1)
```

```
##
## Call:
## lm(formula = InfctRsk ~ Stay + Xray + i2 + i3 + i4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.65509 -0.54889  0.02168  0.56091  2.48797
##
## Coefficients: (2 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.996407   1.044276  -1.912   0.0612 .
## Stay         0.486941   0.101319   4.806 1.27e-05 ***
## Xray         0.018207   0.007202   2.528   0.0144 *
## i2           0.160550   0.272069   0.590   0.5576
## i3                 NA         NA      NA       NA
## i4                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9852 on 54 degrees of freedom
## Multiple R-squared:  0.4601, Adjusted R-squared:  0.4301
## F-statistic: 15.34 on 3 and 54 DF,  p-value: 2.438e-07
```

```
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept) -2.134259   0.877347  -2.433  0.01668 *
# Stay         0.505394   0.081455   6.205 1.11e-08 ***
# Xray         0.017587   0.005649   3.113  0.00238 **
# i2           0.171284   0.281475   0.609  0.54416
# i3           0.095461   0.288852   0.330  0.74169
# i4           1.057835   0.378077   2.798  0.00612 **
# ---
# Residual standard error: 1.036 on 105 degrees of freedom
# Multiple R-squared:  0.4198,  Adjusted R-squared:  0.3922
# F-statistic: 15.19 on 5 and 105 DF,  p-value: 3.243e-11

model.2 <- lm(InfctRsk ~ Stay + Xray)
anova(model.2, model.1)
```

```
## Analysis of Variance Table
##
## Model 1: InfctRsk ~ Stay + Xray
```

```
## Model 2: InfctRsk ~ Stay + Xray + i2 + i3 + i4
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     55 52.754
## 2     54 52.416  1   0.33801 0.3482 0.5576
```

```
#   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
# 1    108 123.56
# 2    105 112.71  3    10.849 3.3687 0.02135 *
```

```
model.3 <- lm(InfctRsk ~ Stay + Xray + i4)
anova(model.3, model.1)
```

```
## Analysis of Variance Table
##
## Model 1: InfctRsk ~ Stay + Xray + i4
## Model 2: InfctRsk ~ Stay + Xray + i2 + i3 + i4
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     55 52.754
## 2     54 52.416  1   0.33801 0.3482 0.5576
```

```
#   Res.Df    RSS Df Sum of Sq      F Pr(>F)
# 1    107 113.11
# 2    105 112.71  2   0.39949 0.1861 0.8305
```

```
detach(infectionrisk)
```