Lesson 13

Christopher A. Swenson (chris@cswenson.com)

11/27/2021

Galton peas (nonconstant variance and weighted least squares)

Load the galton data. Fit an ordinary least squares (OLS) simple linear regression model of Progeny vs Parent. Fit a weighted least squares (WLS) model using weights = . Create a scatterplot of the data with a regression line for each model.

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

model.1 <- lm(Progeny ~ Parent)
summary(model.1)
```

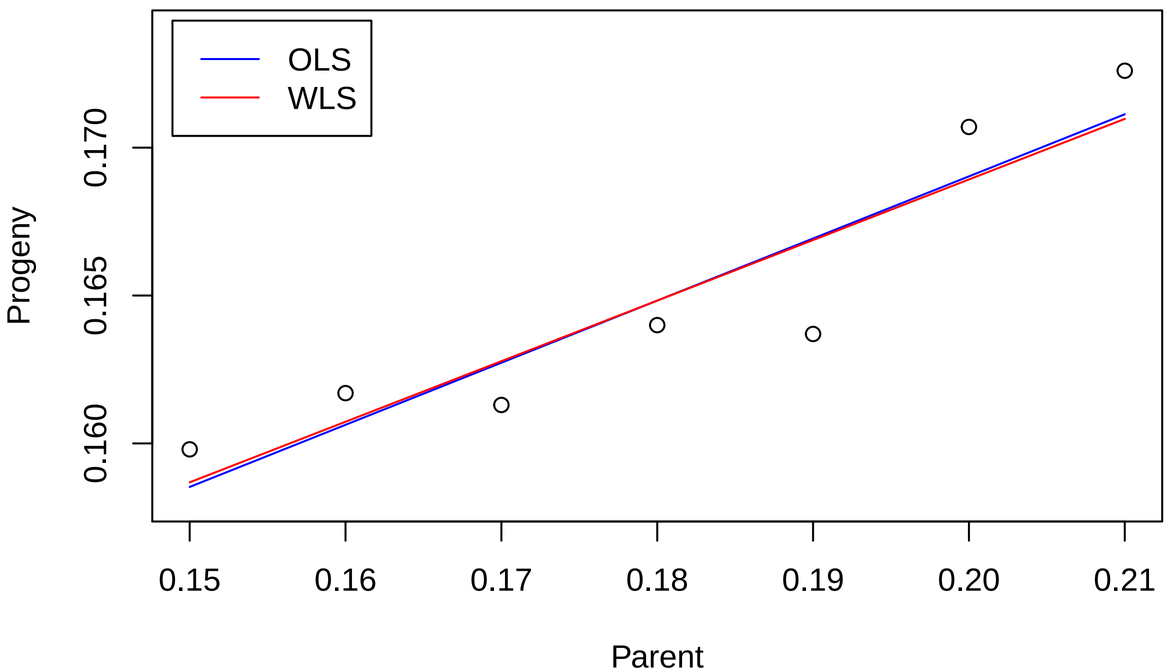
```
##
## Call:
## lm(formula = Progeny ~ Parent)
##
## Residuals:
##          1          2          3          4          5          6          7
##  0.0014714  0.0016714 -0.0032286 -0.0008286 -0.0014286  0.0010714  0.0012714
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.127029   0.006993  18.164 9.29e-06 ***
## Parent      0.210000   0.038614   5.438  0.00285 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002043 on 5 degrees of freedom
## Multiple R-squared:  0.8554, Adjusted R-squared:  0.8265
## F-statistic: 29.58 on 1 and 5 DF,  p-value: 0.002852
```

```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) 0.127029   0.006993  18.164 9.29e-06 ***
# Parent      0.210000   0.038614   5.438  0.00285 **

model.2 <- lm(Progeny ~ Parent, weights=1/SD^2)
summary(model.2)
```

```
##
## Call:
## lm(formula = Progeny ~ Parent, weights = 1/SD^2)
##
## Weighted Residuals:
##       1        2        3        4        5        6        7
```

```
##  0.08187  0.09162 -0.16753 -0.04067 -0.08950  0.06071  0.06328
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.127964   0.006811  18.787 7.87e-06 ***
## Parent      0.204801   0.038155   5.368  0.00302 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.11 on 5 degrees of freedom
## Multiple R-squared:  0.8521, Adjusted R-squared:  0.8225
## F-statistic: 28.81 on 1 and 5 DF,  p-value: 0.003021
```

```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept) 0.127964   0.006811  18.787 7.87e-06 ***
# Parent      0.204801   0.038155   5.368  0.00302 **

plot(x=Parent, y=Progeny, ylim=c(0.158,0.174),
     panel.last = c(lines(sort(Parent), fitted(model.1)[order(Parent)], col="blue"),
                    lines(sort(Parent), fitted(model.2)[order(Parent)], col="red")))
legend("topleft", col=c("blue","red"), lty=1,
       inset=0.02, legend=c("OLS", "WLS"))
```



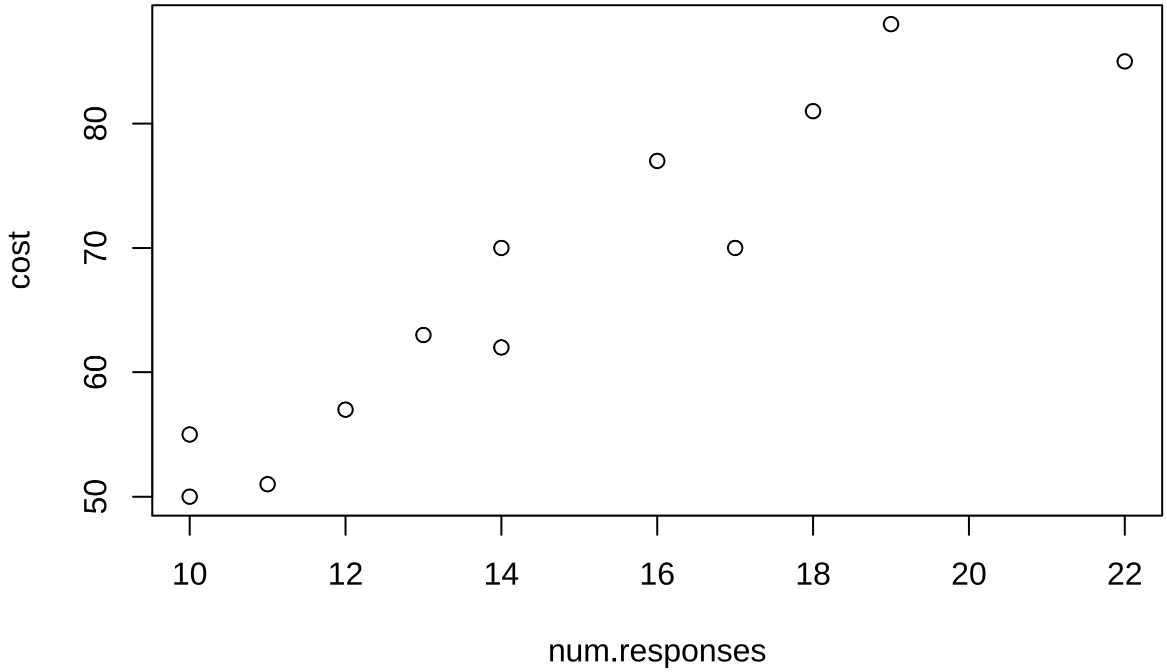

```
detach(galton)
```

Computer-assisted learning (nonconstant variance and weighted least squares)

Load the ca_learning data. Create a scatterplot of the data. Fit an OLS model. Plot the OLS residuals vs num.responses. Plot the absolute OLS residuals vs num.responses. Calculate fitted values from a regression of absolute residuals vs num.responses. Fit a WLS model using weights = . Create a scatterplot of the data with a regression line for each model. Plot the WLS standardized residuals vs num.responses.

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

```r
plot(x=num.responses, y=cost)
```

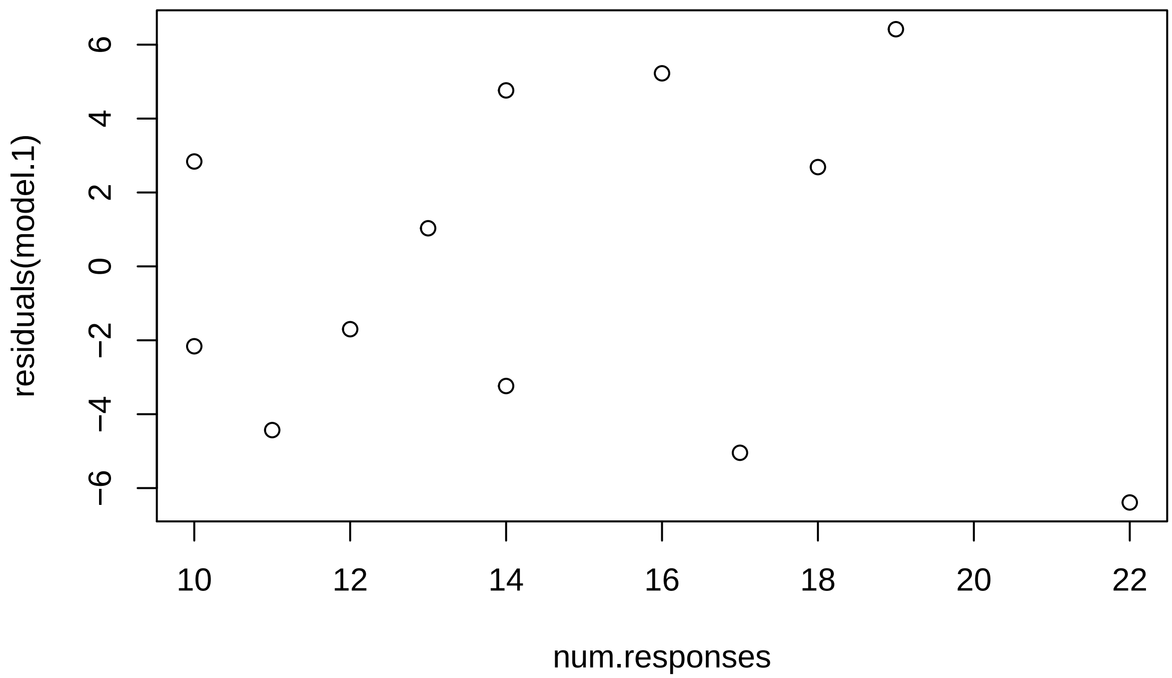



```r
model.1 <- lm(cost ~ num.responses)
summary(model.1)
```

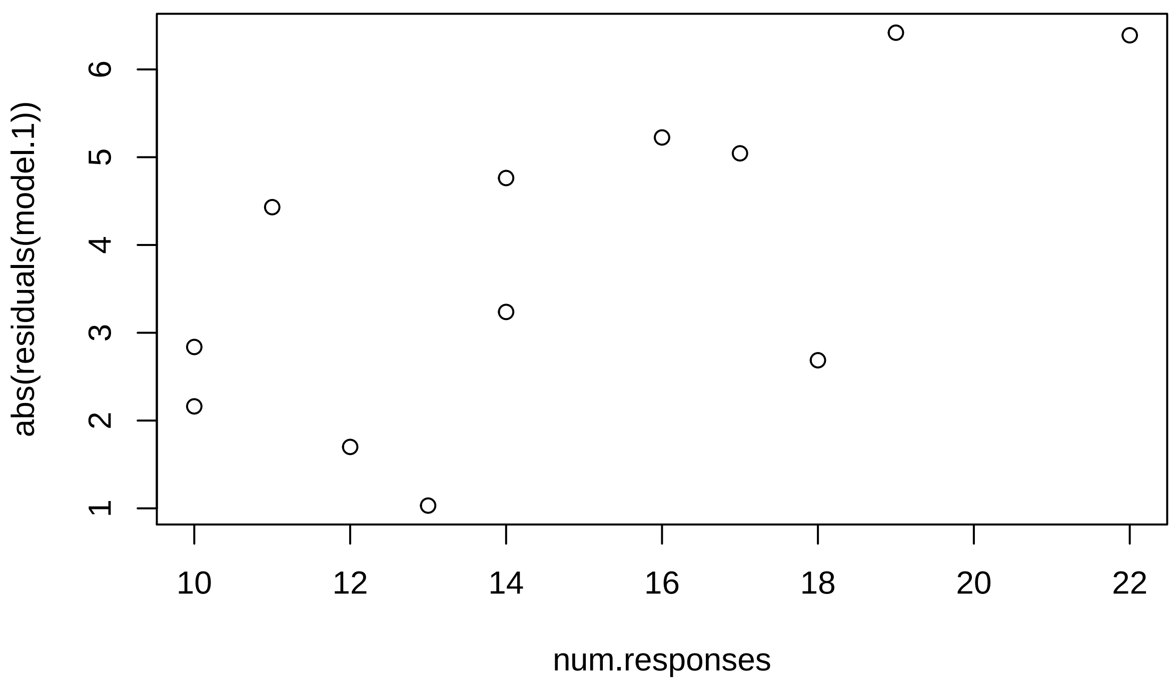
```
##
## Call:
## lm(formula = cost ~ num.responses)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.389 -3.536 -0.334  3.319  6.418
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    19.4727     5.5162   3.530  0.00545 **
## num.responses   3.2689     0.3651   8.955 4.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.598 on 10 degrees of freedom
## Multiple R-squared:  0.8891, Adjusted R-squared:  0.878
## F-statistic: 80.19 on 1 and 10 DF,  p-value: 4.33e-06
```

```r
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)    19.4727     5.5162   3.530  0.00545 **
# num.responses   3.2689     0.3651   8.955 4.33e-06 ***
# ---
# Residual standard error: 4.598 on 10 degrees of freedom
# Multiple R-squared:  0.8891,  Adjusted R-squared:  0.878
# F-statistic: 80.19 on 1 and 10 DF,  p-value: 4.33e-06
```

```r
plot(num.responses, residuals(model.1))
```




```r
plot(num.responses, abs(residuals(model.1)))
```

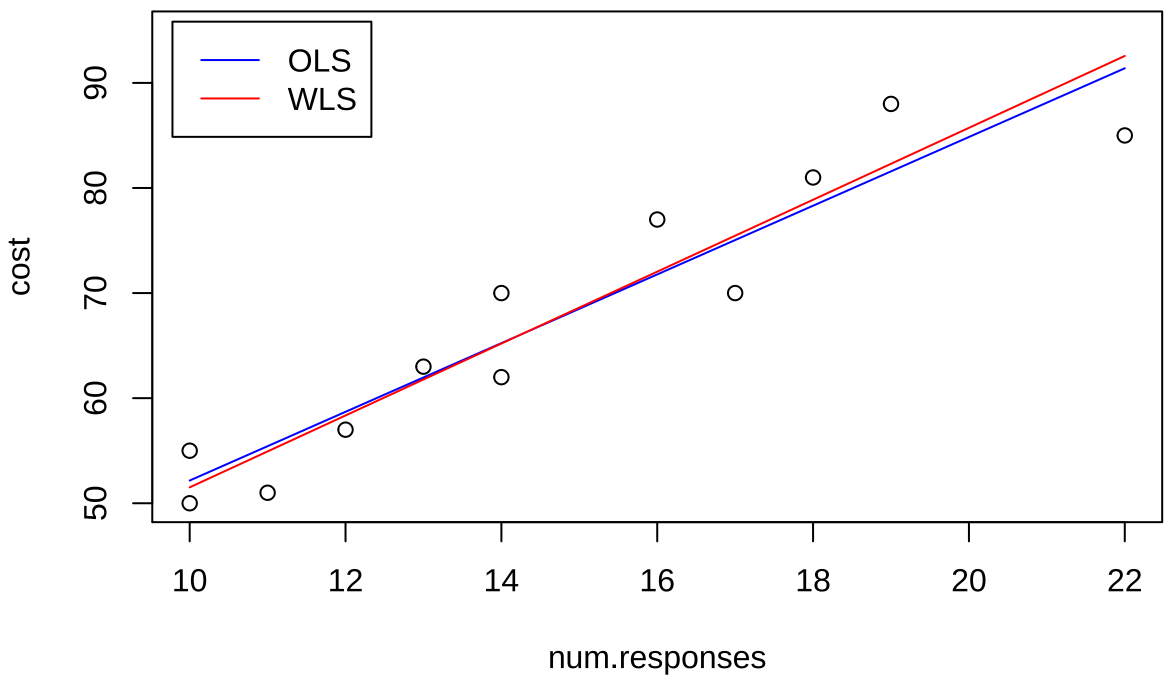



```r
wts <- 1/fitted(lm(abs(residuals(model.1)) ~ num.responses))^2

model.2 <- lm(cost ~ num.responses, weights=wts)
summary(model.2)
```

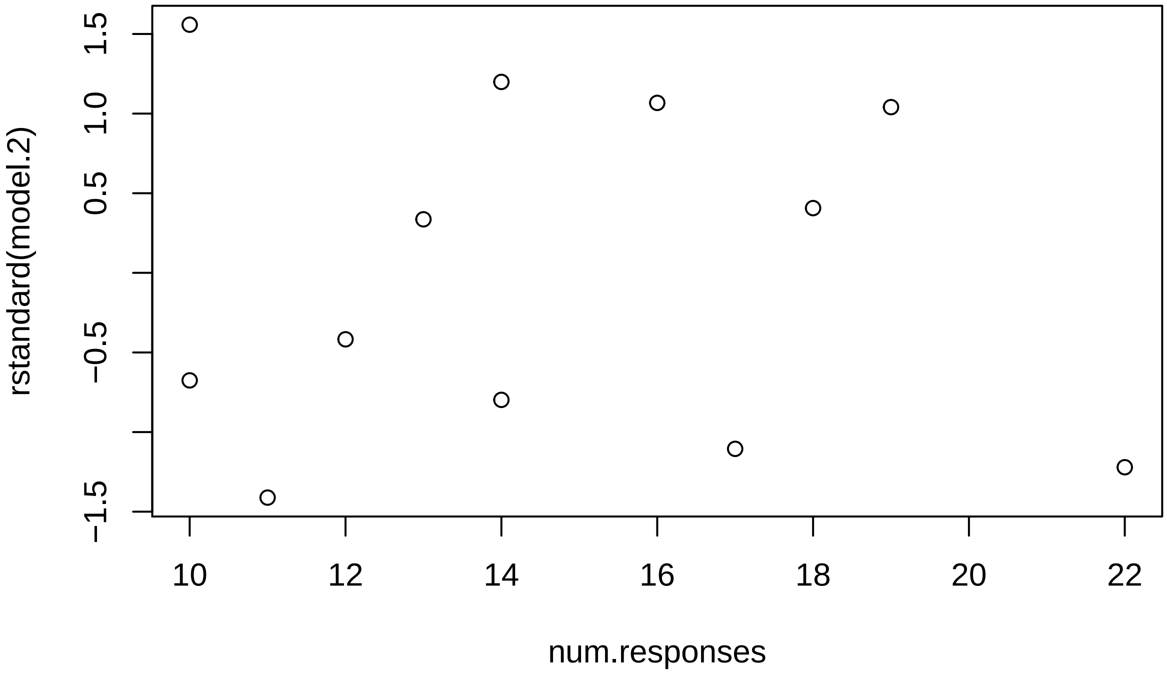
```
##
## Call:
```

```
## lm(formula = cost ~ num.responses, weights = wts)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48741 -0.96167 -0.04198  1.10930  1.50265
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.3006     4.8277   3.584  0.00498 **
## num.responses   3.4211     0.3703   9.238 3.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.159 on 10 degrees of freedom
## Multiple R-squared:  0.8951, Adjusted R-squared:  0.8846
## F-statistic: 85.35 on 1 and 10 DF,  p-value: 3.269e-06
```

```r
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)    17.3006     4.8277   3.584  0.00498 **
# num.responses   3.4211     0.3703   9.238 3.27e-06 ***
# ---
# Residual standard error: 1.159 on 10 degrees of freedom
# Multiple R-squared:  0.8951,  Adjusted R-squared:  0.8846
# F-statistic: 85.35 on 1 and 10 DF,  p-value: 3.269e-06

plot(x=num.responses, y=cost, ylim=c(50,95),
     panel.last = c(lines(sort(num.responses), fitted(model.1)[order(num.responses)], col="blue"),
                    lines(sort(num.responses), fitted(model.2)[order(num.responses)], col="red")))
legend("topleft", col=c("blue","red"), lty=1,
       inset=0.02, legend=c("OLS", "WLS"))
```



```
plot(num.responses, rstandard(model.2))
```




```
detach(ca_learning)
```

Market share (nonconstant variance and weighted least squares)

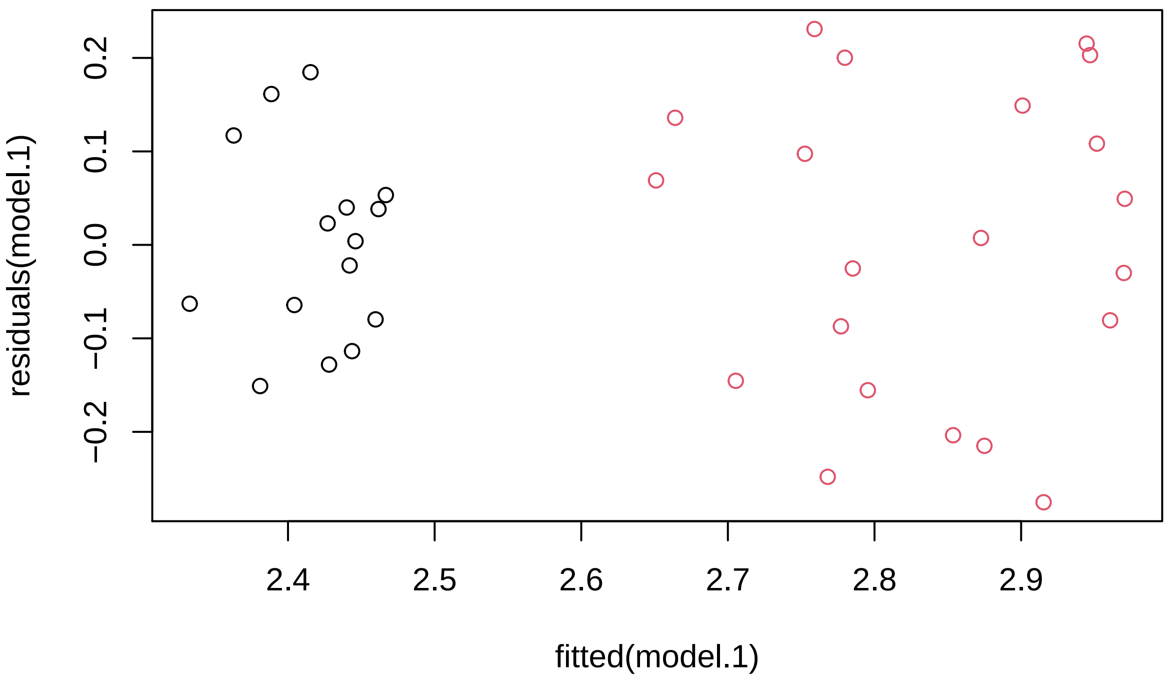
Load the marketshare data. Fit an OLS model. Plot the OLS residuals vs fitted values with points marked by Discount. Use the tapply function to calculate the residual variance for Discount=0 and Discount=1. Fit a WLS model using weights = 1/variance for Discount=0 and Discount=1. Plot the WLS standardized residuals vs fitted values.

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

model.1 <- lm(MarketShare ~ Price + P1 + P2)
summary(model.1)
```

```
##
## Call:
## lm(formula = MarketShare ~ Price + P1 + P2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27532 -0.09376  0.00568  0.11050  0.23090
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.19592    0.35616   8.973 3.00e-10 ***
## Price       -0.33358    0.15226  -2.191   0.0359 *
## P1           0.30808    0.06412   4.804 3.51e-05 ***
## P2           0.48431    0.05541   8.740 5.49e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1461 on 32 degrees of freedom
## Multiple R-squared:  0.7206, Adjusted R-squared:  0.6944
## F-statistic: 27.51 on 3 and 32 DF,  p-value: 5.462e-09
```

```
  #             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  3.19592    0.35616   8.973 3.00e-10 ***
# Price       -0.33358    0.15226  -2.191   0.0359 *
# P1           0.30808    0.06412   4.804 3.51e-05 ***
# P2           0.48431    0.05541   8.740 5.49e-10 ***

plot(fitted(model.1), residuals(model.1), col=Discount+1)
```

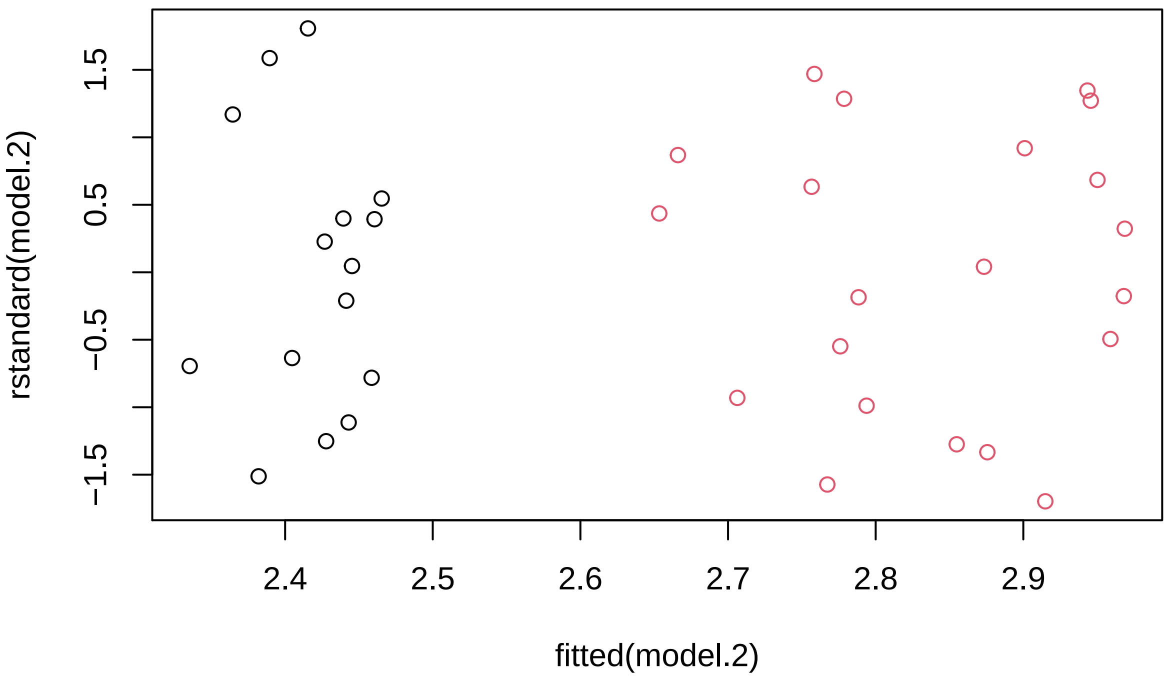



```
vars <- tapply(residuals(model.1), Discount, var)
#          0          1
# 0.01052324 0.02680546

wts <- Discount/vars[2] + (1-Discount)/vars[1]

model.2 <- lm(MarketShare ~ Price + P1 + P2, weights=wts)
summary(model.2)
```

```
##
## Call:
## lm(formula = MarketShare ~ Price + P1 + P2, weights = wts)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67901 -0.79798  0.04314  0.70729  1.79894
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.17437    0.35671   8.899 3.63e-10 ***
```

```
## Price       -0.32432    0.15291  -2.121   0.0418 *
## P1           0.30834    0.06575   4.689 4.89e-05 ***
## P2           0.48419    0.05422   8.930 3.35e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.031 on 32 degrees of freedom
## Multiple R-squared:  0.7425, Adjusted R-squared:  0.7184
## F-statistic: 30.76 on 3 and 32 DF,  p-value: 1.501e-09
```

```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  3.17437    0.35671   8.899 3.63e-10 ***
# Price       -0.32432    0.15291  -2.121   0.0418 *
# P1           0.30834    0.06575   4.689 4.89e-05 ***
# P2           0.48419    0.05422   8.930 3.35e-10 ***

plot(fitted(model.2), rstandard(model.2), col=Discount+1)
```




```
detach(marketshare)
```

Home price (nonconstant variance and weighted least squares)

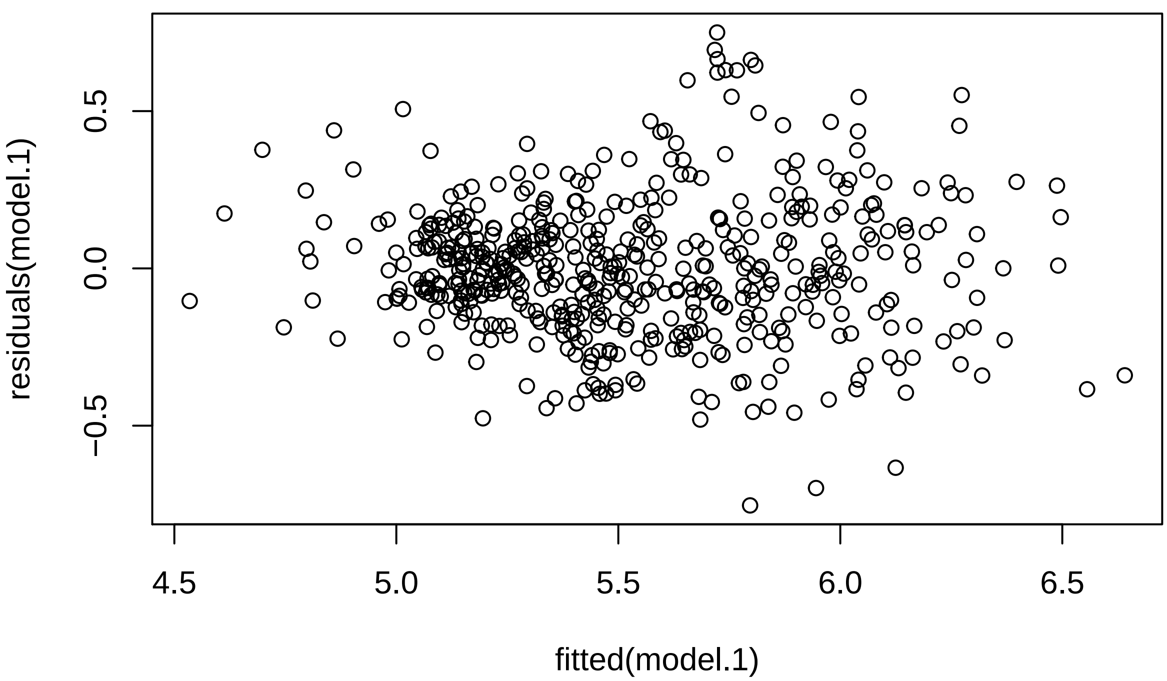
Load the realestate data. Calculate log transformations of the variables. Fit an OLS model. Plot the OLS residuals vs fitted values. Calculate fitted values from a regression of absolute residuals vs fitted values. Fit a WLS model using weights = . Plot the WLS standardized residuals vs fitted values.

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

logY <- log(SalePrice)
logX1 <- log(SqFeet)
logX2 <- log(Lot)

model.1 <- lm(logY ~ logX1 + logX2)
```

```r
summary(model.1)
```

```
##
## Call:
## lm(formula = logY ~ logX1 + logX2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7534 -0.1354 -0.0063  0.1246  0.7499
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.25485    0.07353  57.864  < 2e-16 ***
## logX1        1.22141    0.03371  36.234  < 2e-16 ***
## logX2        0.10595    0.02394   4.425 1.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2196 on 518 degrees of freedom
## Multiple R-squared:  0.7411, Adjusted R-squared:  0.7401
## F-statistic: 741.4 on 2 and 518 DF,  p-value: < 2.2e-16
```

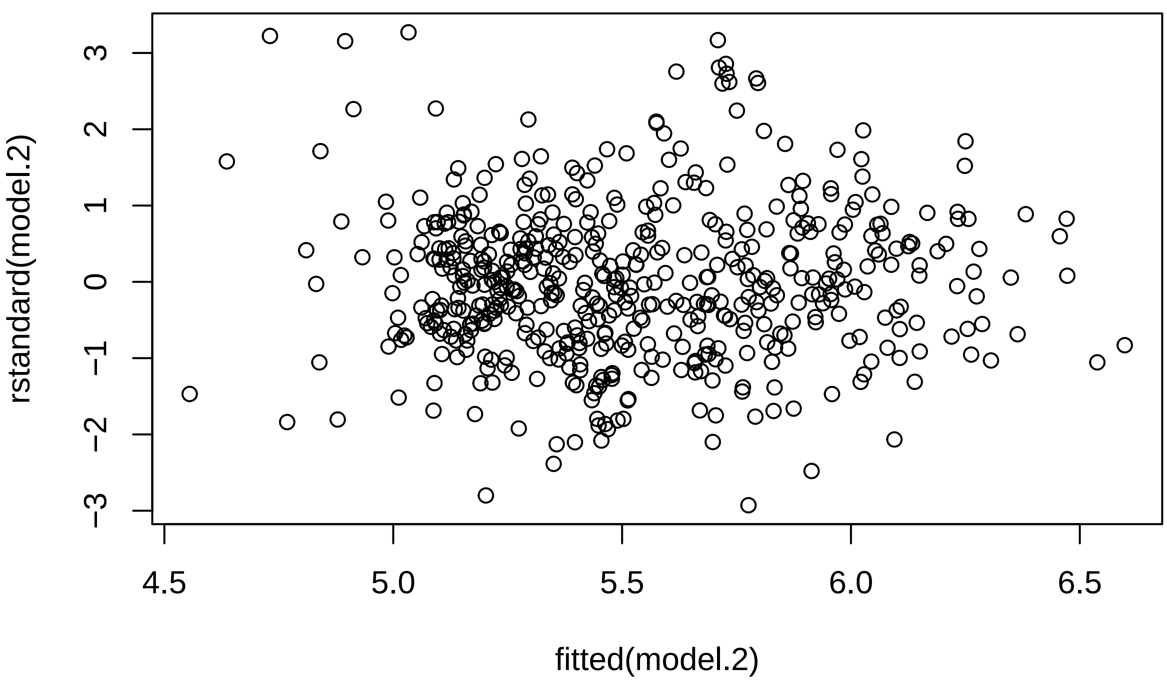
```r
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  4.25485    0.07353  57.864  < 2e-16 ***
# logX1        1.22141    0.03371  36.234  < 2e-16 ***
# logX2        0.10595    0.02394   4.425 1.18e-05 ***

plot(fitted(model.1), residuals(model.1))
```




```r
wts <- 1/fitted(lm(abs(residuals(model.1)) ~ fitted(model.1)))^2

model.2 <- lm(logY ~ logX1 + logX2, weights=wts)
summary(model.2)
```

```
##
## Call:
## lm(formula = logY ~ logX1 + logX2, weights = wts)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7794 -0.8686 -0.0410  0.7753  4.2153
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.35189    0.06330  68.755  < 2e-16 ***
## logX1        1.20150    0.03332  36.065  < 2e-16 ***
## logX2        0.07924    0.02152   3.682 0.000255 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.294 on 518 degrees of freedom
## Multiple R-squared:  0.744,  Adjusted R-squared:  0.743
## F-statistic: 752.6 on 2 and 518 DF,  p-value: < 2.2e-16
```

```
#             Estimate Std. Error t value Pr(>|t|)
# (Intercept)  4.35189    0.06330  68.755  < 2e-16 ***
# logX1        1.20150    0.03332  36.065  < 2e-16 ***
# logX2        0.07924    0.02152   3.682 0.000255 ***

plot(fitted(model.2), rstandard(model.2))
```




```
detach(realestate)
```