Galton Dataset

Rutvij Kortikar

September 14, 2019

Contents

Inquiry Questions

It was in investigating this historical dataset that Galton made the crucial observation of "regression towards the mean".

LAB GOAL: Create a narrative from your analysis of the Galton dataset, where you mention the following:

CONCEPTUAL GOAL

Explain the relation between correlation and linear regression.

Linear regression is a way to predict the dependent variable from the independent variable (which we set). The correlation is merely a metric corresponding to the residuals between the regression line and the actual data. This corresponds to a r^2 value which tells us exactly how much of the variation in the response is accounted for by the predictor.

When would regression towards the mean not happen?

Regression towards the mean would not happen if the predictor value was at the mean. For example, average height parents may have taller or shorter children, but it is less likely that incredibly tall parents would have even taller children. Case in point, None of JS Bach's 20 kids surpassed the ability of the great master, and none of Gauss's children would surpass him despite having talent for math.

When would regression towards the mean be extreme?

Consequently, regression towards the mean occurs strongly when the predictors are far away from the mean and minimally when the predictors are close to the mean.

Can you relate correlation to notions of dimensionality reduction geometrically?

Where $\mathbb{D}$ is the probability distribution of the data before and after applying dimensionality reduction. This relationship is given by:

$$R^2 = \frac{E\{\mathbb{D}_n\} - E\{\mathbb{D}_{n-k}\}}{E\{\mathbb{D}_n\}}.$$

Why does the "regression effect" take place?

The regression effect occurs because the underlying data is concentrated along a certain line or curve.

Visualization of the data to bring out the facts pertaining to the predictors of height.

Can you predict the father's and mother's height from the height of children?

Yes.

What do you observe about the parents' heights relative to children?

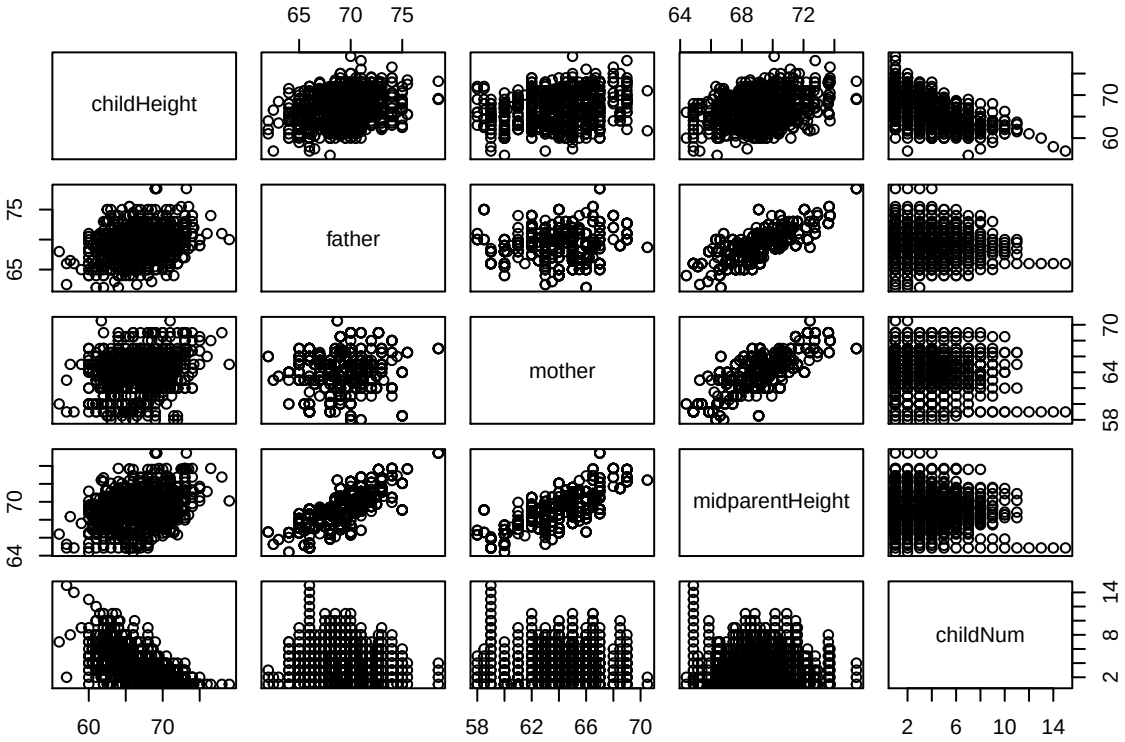
Parents' heights are generally similar to the children's heights. However, children tend to be taller. Furthermore, when parents are very tall we need to account for regression towards the mean.

Initial Exploration of Parameters

```
galt <- read.csv("/home/rwt/warmup/100datasets/warm-up/galton/galton-families.csv")
summary(galt)
```

```
##        X             family        father         mother
##  Min.   :  1.0   185    : 15   Min.   :62.0   Min.   :58.00
##  1st Qu.:234.2   066    : 11   1st Qu.:68.0   1st Qu.:63.00
##  Median :467.5   120    : 11   Median :69.0   Median :64.00
##  Mean   :467.5   130    : 11   Mean   :69.2   Mean   :64.09
##  3rd Qu.:700.8   166    : 11   3rd Qu.:71.0   3rd Qu.:65.88
##  Max.   :934.0   097    : 10   Max.   :78.5   Max.   :70.50
##                  (Other):865
##  midparentHeight    children         childNum         gender
##  Min.   :64.40   Min.   : 1.000   Min.   : 1.000   female:453
##  1st Qu.:68.14   1st Qu.: 4.000   1st Qu.: 2.000   male  :481
##  Median :69.25   Median : 6.000   Median : 3.000
##  Mean   :69.21   Mean   : 6.171   Mean   : 3.586
##  3rd Qu.:70.14   3rd Qu.: 8.000   3rd Qu.: 5.000
##  Max.   :75.43   Max.   :15.000   Max.   :15.000
##
##   childHeight
##  Min.   :56.00
##  1st Qu.:64.00
##  Median :66.50
##  Mean   :66.75
##  3rd Qu.:69.70
##  Max.   :79.00
##
```

```
attach(galt)
pairs(~ childHeight + father + mother + midparentHeight + childNum)
```




```
#scatterplotMatrix(~ childHeight + father + mother + midparentHeight + childNum)
```

A regression model for the offspring height based on parents' heights.

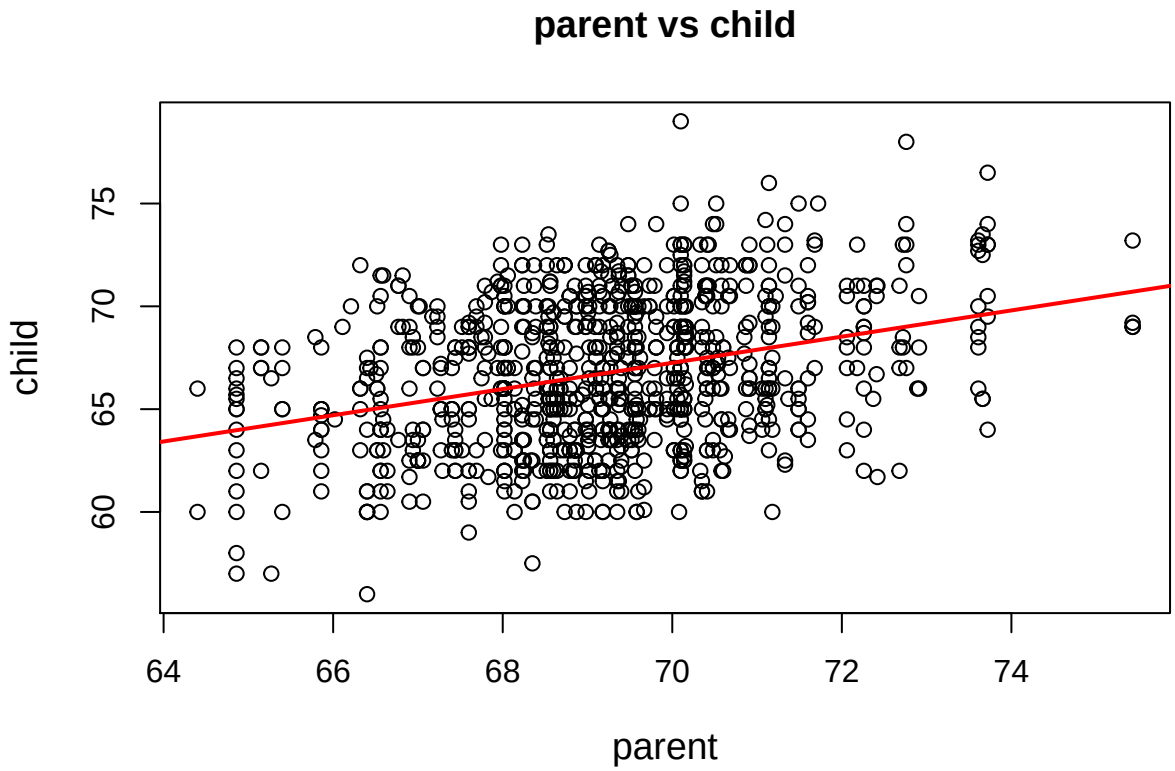
Naive Model

```
x = midparentHeight
Y = childHeight
plot(x,Y,xlab="parent",ylab="child",cex.lab=1.2)
title("parent vs child", cex.main=1.2)

mod.lm <- lm(Y ~ x)
yhat <- fitted(mod.lm)
summary(mod.lm)
```

```
##
## Call:
## lm(formula = Y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9570 -2.6989 -0.2155  2.7961 11.6848
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.63624    4.26511   5.307 1.39e-07 ***
## x            0.63736    0.06161  10.345  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.392 on 932 degrees of freedom
## Multiple R-squared:  0.103,  Adjusted R-squared:  0.102
## F-statistic:   107 on 1 and 932 DF,  p-value: < 2.2e-16
```

```
abline(a=22.63624,b=0.63736,col=2,lwd=2)
```



Clearly the R^2 of .102 means that much of the variation is explained by other parameters.

Automated Model Selection

Feature-engineering for a better model.

Which regression algorithm works best.

Use AdjR^2, Bayesian Information Criterion, Mallow's Cp to find best model.

```
library(leaps)

subset <- regsubsets(childHeight ~ father + mother + childNum + gender + midparentHeight + childHeight,
```

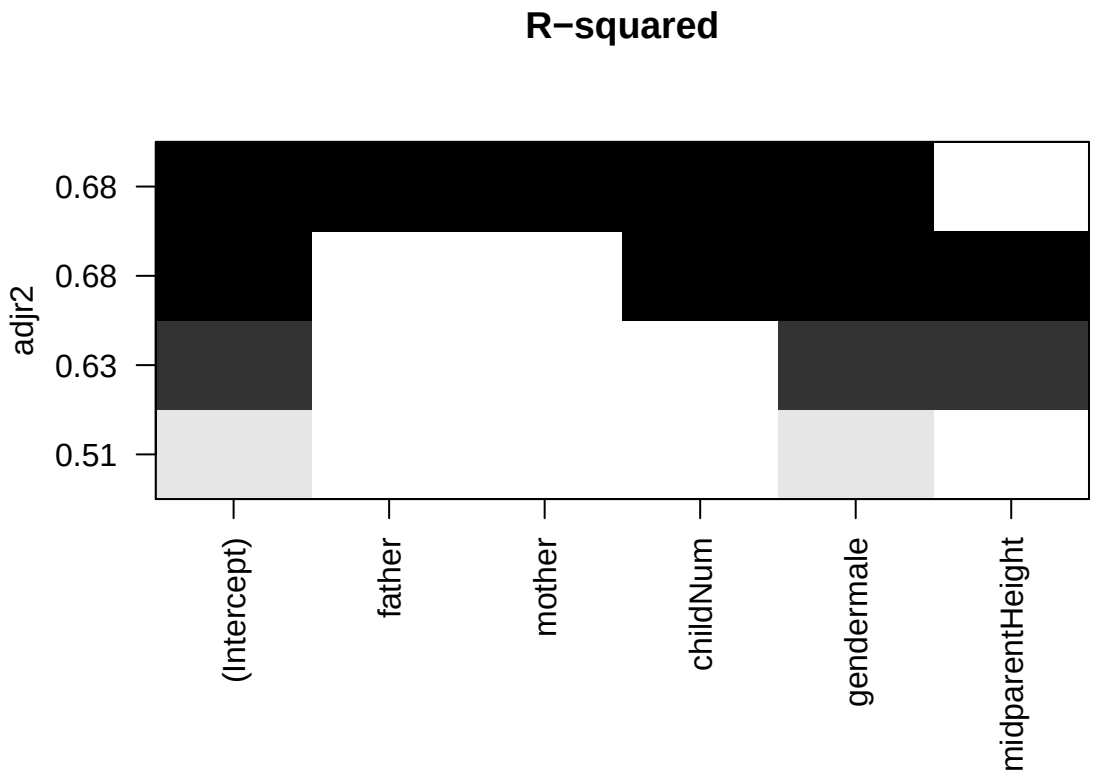
```
## Warning in model.matrix.default(terms(formula, data = data), mm): the
## response appeared on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(terms(formula, data = data), mm): problem
## with term 6 in model.matrix: no columns are assigned
```

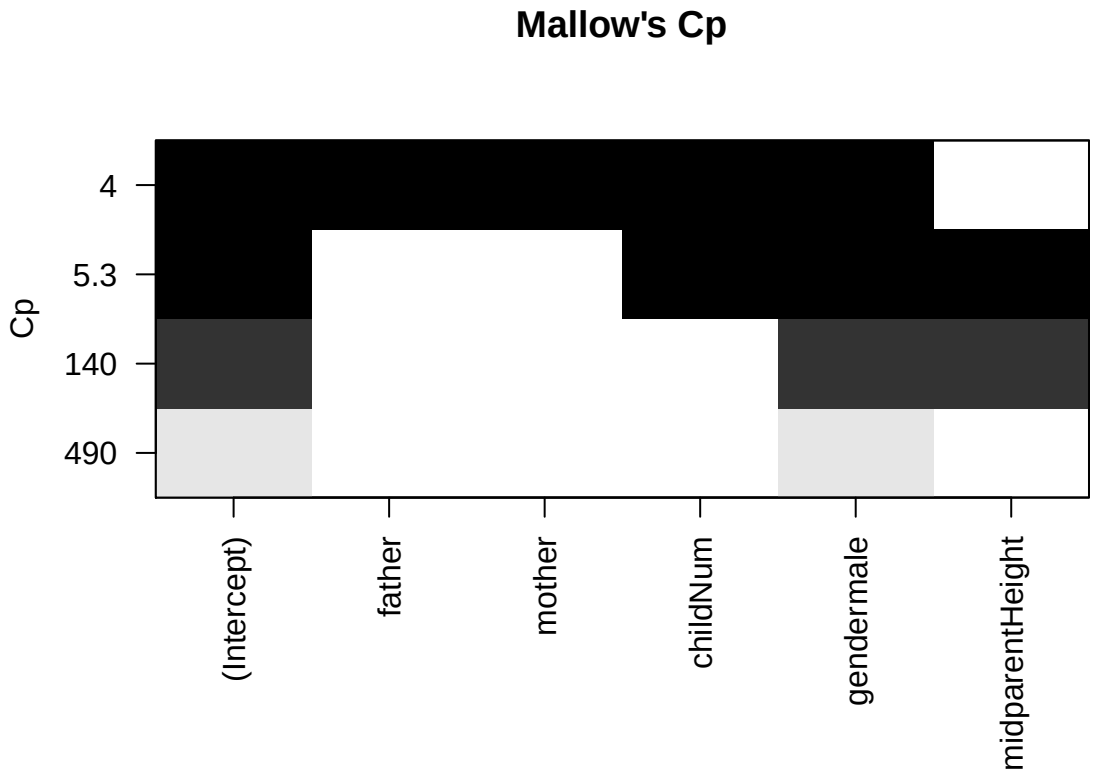
```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : nvmax reduced to 4
```

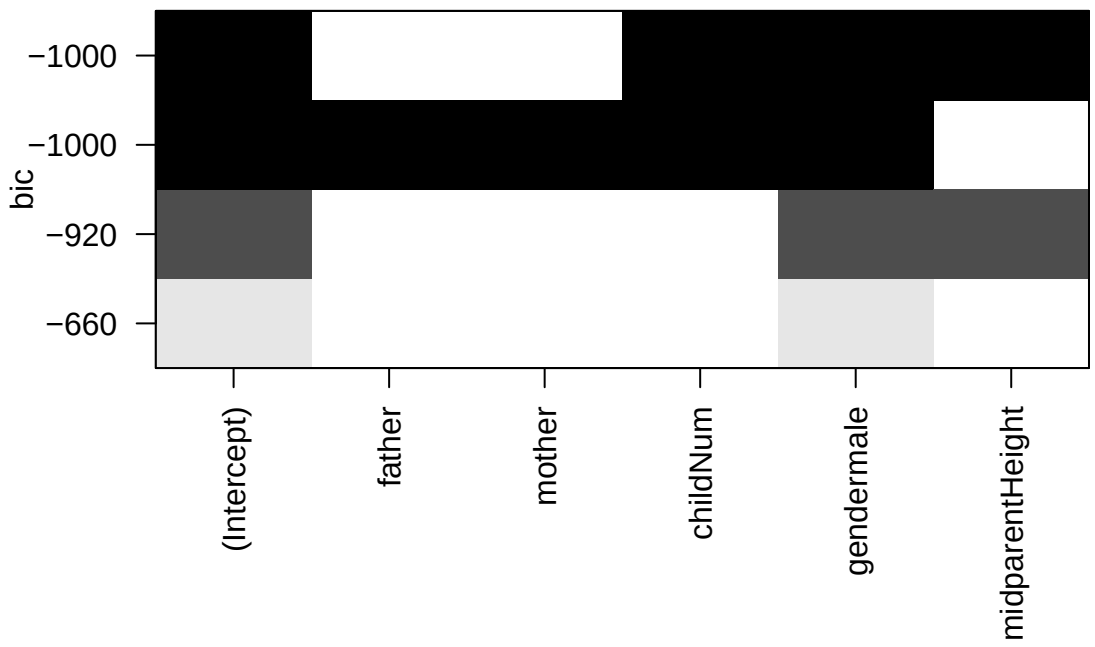
```
plot(subset,scale="adjr2",main="R-squared")
```




```
plot(subset,scale="Cp",main="Mallow's Cp")
```




```
plot(subset,scale="bic",main="BIC")
```



These techniques reveal some interesting results! Using R-squared is not as effective for such a large data set so we will compare Mallow's Cp and the Bayesian Information Criterion.

It is interesting to note that Mallow's CP prefers using mother height and father height separately, whereas BIC uses the midparent height, as it favours a more parsimonious model which aggregates the available information. Fortunately neither model uses all three, which shows that our automated model selection techniques are working in the right direction.

The final model ought to use midparentHeight, gender, and childNum as the predictors as they yield the most information regarding the predicted height of the child.

/// Here I ran out of time, but I would run a two factor segregation by gender and whether or not the child was a firstborn. This would allow me to achieve the highest R^2 value using the most straightforward model with the most information.

```
# read data
detach(galt)
galton=read.csv("galton-families.csv")
summary(galton)
```

```
##        X             family        father         mother
##  Min.   :  1.0   185    : 15   Min.   :62.0   Min.   :58.00
##  1st Qu.:234.2   066    : 11   1st Qu.:68.0   1st Qu.:63.00
##  Median :467.5   120    : 11   Median :69.0   Median :64.00
##  Mean   :467.5   130    : 11   Mean   :69.2   Mean   :64.09
##  3rd Qu.:700.8   166    : 11   3rd Qu.:71.0   3rd Qu.:65.88
##  Max.   :934.0   097    : 10   Max.   :78.5   Max.   :70.50
##                  (Other):865
##  midparentHeight    children         childNum         gender
##  Min.   :64.40   Min.   : 1.000   Min.   : 1.000   female:453
##  1st Qu.:68.14   1st Qu.: 4.000   1st Qu.: 2.000   male  :481
##  Median :69.25   Median : 6.000   Median : 3.000
```

```
##  Mean   :69.21   Mean   : 6.171   Mean   : 3.586
##  3rd Qu.:70.14   3rd Qu.: 8.000   3rd Qu.: 5.000
##  Max.   :75.43   Max.   :15.000   Max.   :15.000
##
##   childHeight
##  Min.   :56.00
##  1st Qu.:64.00
##  Median :66.50
##  Mean   :66.75
##  3rd Qu.:69.70
##  Max.   :79.00
##
```

```
attach(galton)
str(galton)
```

```
## 'data.frame':    934 obs. of  9 variables:
##  $ X              : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ family         : Factor w/ 205 levels "001","002","003",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ father         : num  78.5 78.5 78.5 78.5 75.5 75.5 75.5 75.5 75 75 ...
##  $ mother         : num  67 67 67 67 66.5 66.5 66.5 66.5 64 64 ...
##  $ midparentHeight: num  75.4 75.4 75.4 75.4 73.7 ...
##  $ children       : int  4 4 4 4 4 4 4 4 2 2 ...
##  $ childNum       : int  1 2 3 4 1 2 3 4 1 2 ...
##  $ gender         : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 1 1 2 1 ...
##  $ childHeight    : num  73.2 69.2 69 69 73.5 72.5 65.5 65.5 71 68 ...
```

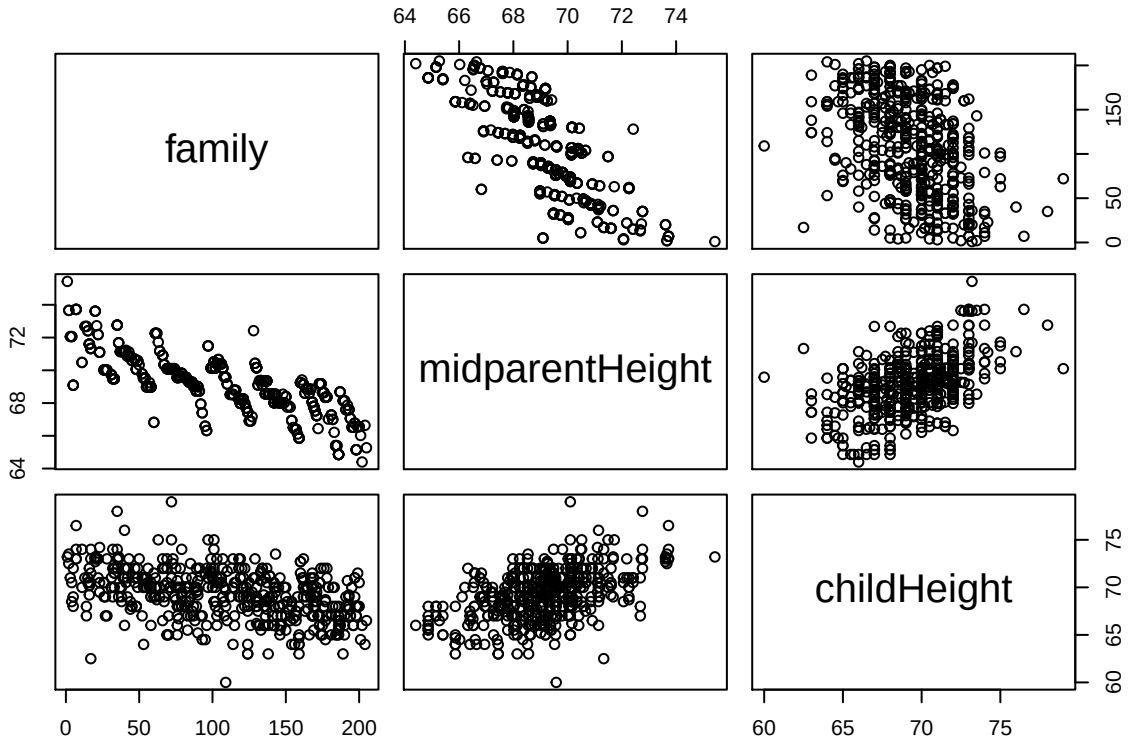
```
#select subset of data
data = subset(galton,select=-c(X,father,mother,children,childNum))
#segregate by gender and family
data$gender = as.numeric(data$gender)
data$family = as.numeric(data$family)

#create custom index for gender
datax = data[data$gender == '1',]
dataF = subset(datax, select = -gender)
datay = data[data$gender == '2',]
dataM = subset(datay, select = -gender)

str(dataM)
```

```
## 'data.frame':    481 obs. of  3 variables:
##  $ family         : num  1 2 2 3 4 4 5 5 5 7 ...
##  $ midparentHeight: num  75.4 73.7 73.7 72.1 72.1 ...
##  $ childHeight    : num  73.2 73.5 72.5 71 70.5 68.5 72 69 68 76.5 ...
```

```
summary(dataM)
```

```
##      family      midparentHeight  childHeight
##  Min.   :  1.0   Min.   :64.40   Min.   :60.00
##  1st Qu.: 64.0   1st Qu.:68.02   1st Qu.:67.50
##  Median :108.0   Median :69.18   Median :69.20
##  Mean   :107.7   Mean   :69.15   Mean   :69.23
##  3rd Qu.:154.0   3rd Qu.:70.14   3rd Qu.:71.00
##  Max.   :205.0   Max.   :75.43   Max.   :79.00
```

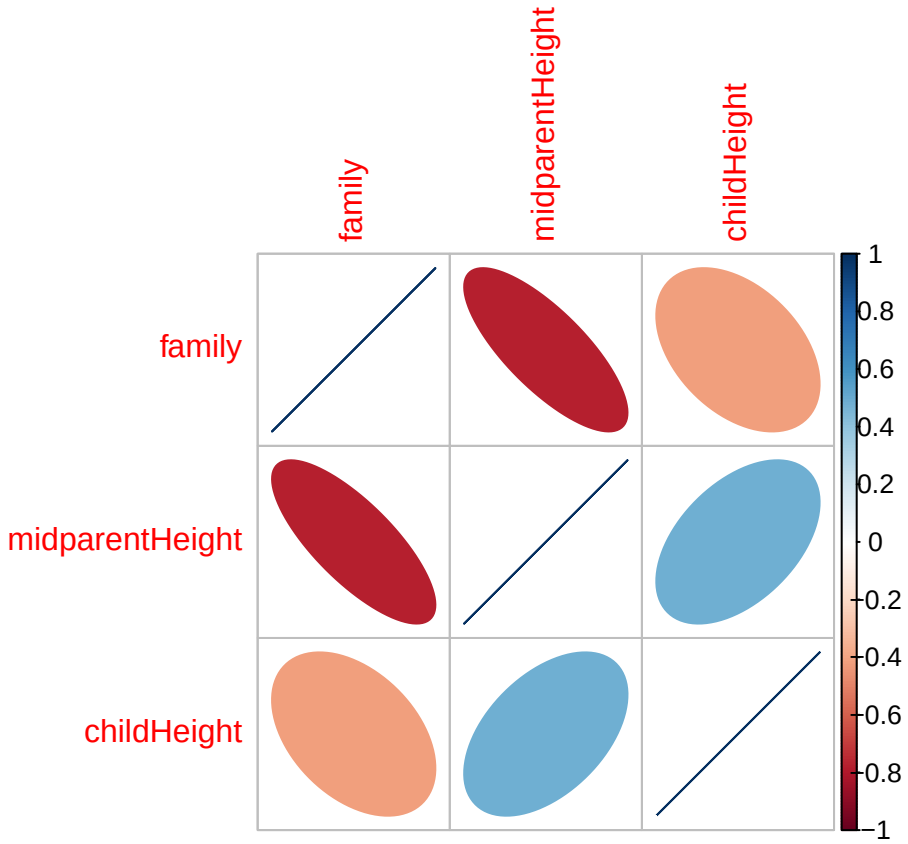
```
pairs(dataM)
```



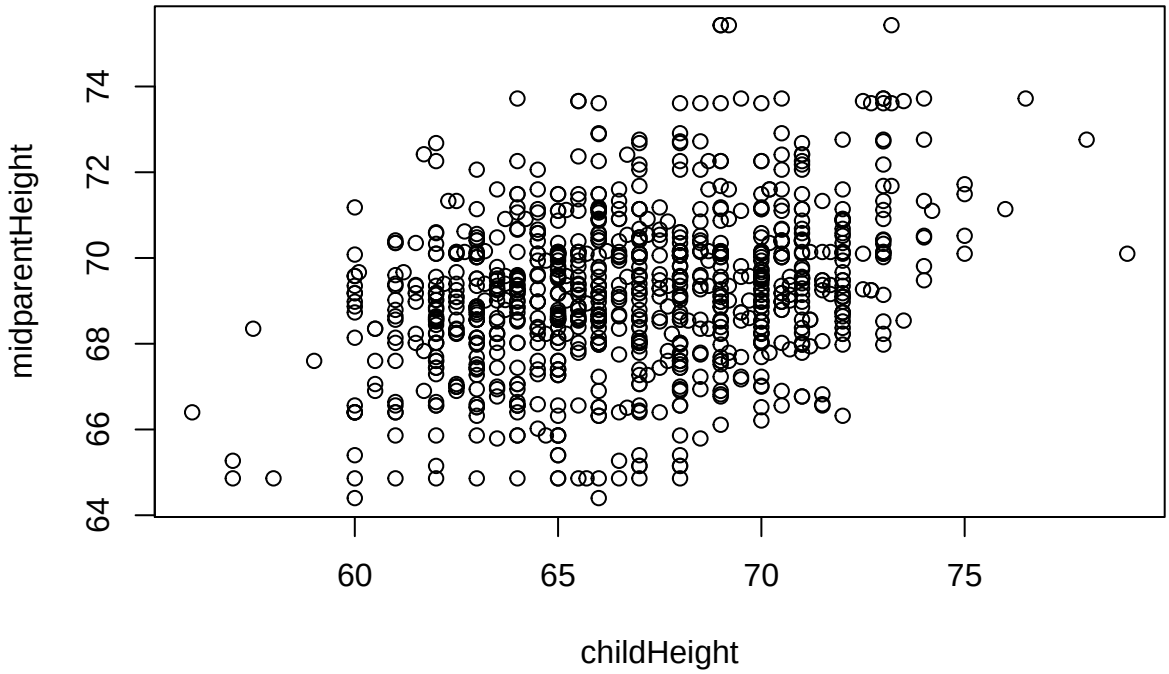

```
galton.corM= cor(dataM)

library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(galton.corM,method="ellipse")
```




```
plot(childHeight,midparentHeight)
```




```
male.model = lm(childHeight~midparentHeight + family, data = dataM)
summary(male.model)
```

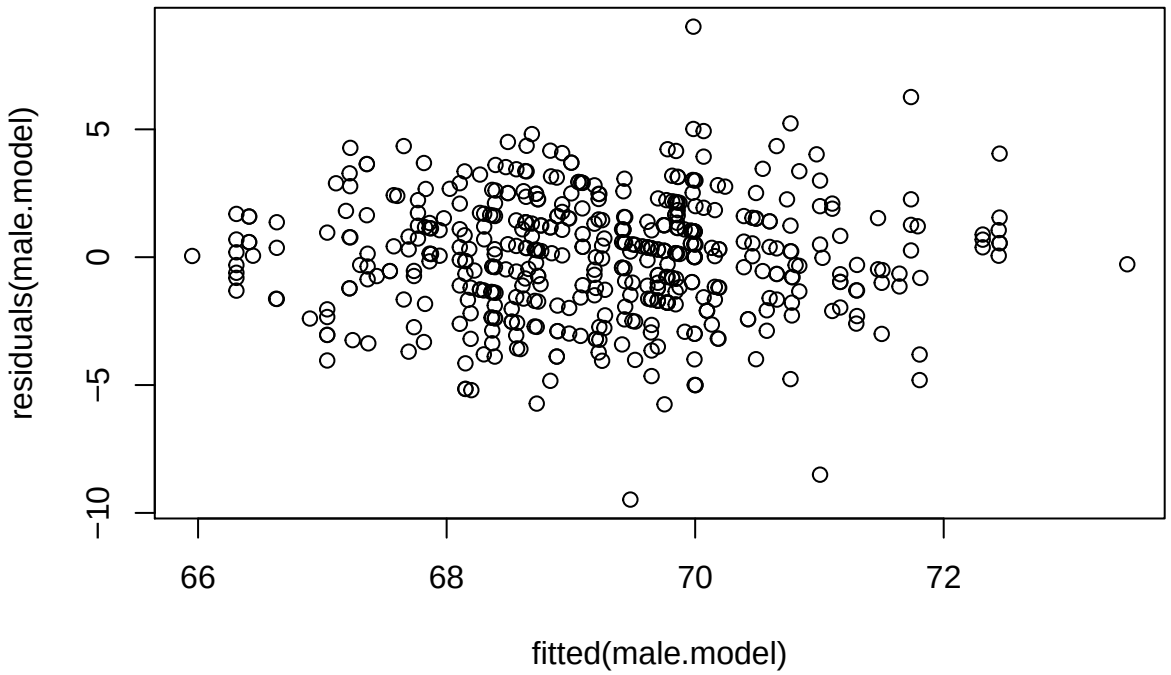
```
##
```

```
## Call:
## lm(formula = childHeight ~ midparentHeight + family, data = dataM)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4786 -1.5599  0.1916  1.5195  9.0144
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     29.625311   6.647491   4.457 1.04e-05 ***
## midparentHeight  0.581433   0.092493   6.286 7.34e-10 ***
## family          -0.005530   0.002989  -1.850   0.0649 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.295 on 478 degrees of freedom
## Multiple R-squared:  0.2385, Adjusted R-squared:  0.2353
## F-statistic: 74.86 on 2 and 478 DF,  p-value: < 2.2e-16
```

```
female.model = lm(childHeight~midparentHeight + family, data = dataF)
summary(female.model)
```

```
##
## Call:
## lm(formula = childHeight ~ midparentHeight + family, data = dataF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3298 -1.3783  0.0175  1.4602  6.7695
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     29.685367   5.940438   4.997 8.34e-07 ***
## midparentHeight  0.506157   0.082691   6.121 2.03e-09 ***
## family          -0.006182   0.002579  -2.397   0.0169 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.013 on 450 degrees of freedom
## Multiple R-squared:  0.2727, Adjusted R-squared:  0.2695
## F-statistic: 84.38 on 2 and 450 DF,  p-value: < 2.2e-16
```

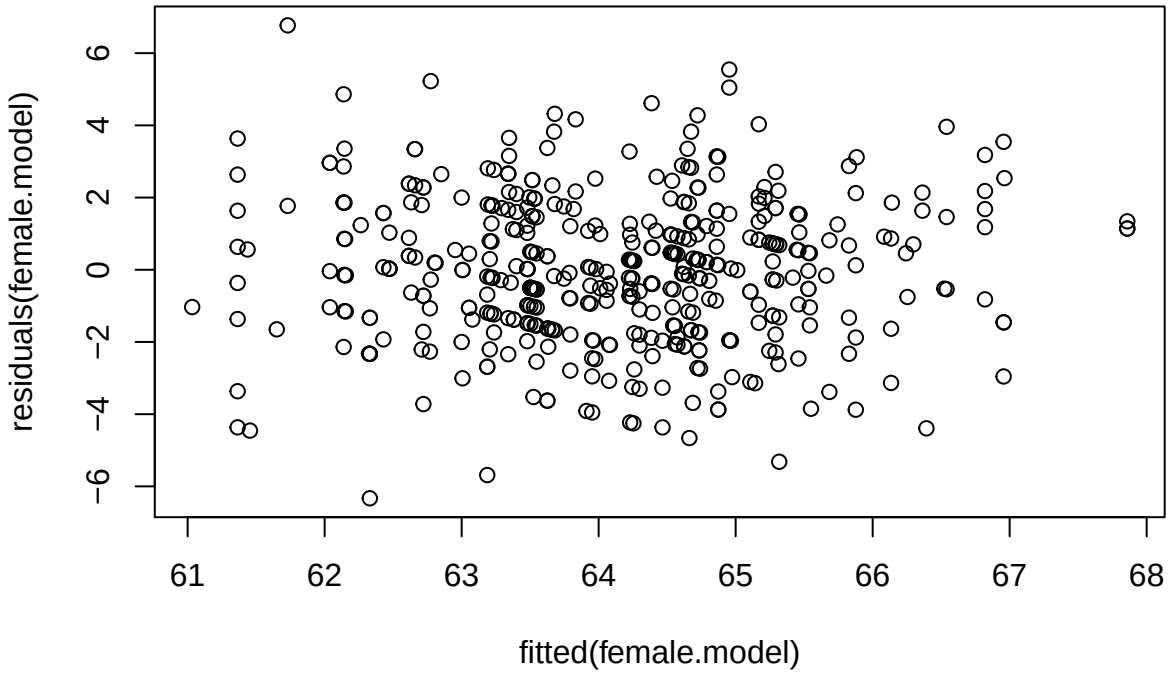
```
#plot
plot(fitted(male.model),residuals(male.model))
```




```
confint(male.model)
```

```
##                       2.5 %       97.5 %
## (Intercept)     16.56339582 4.268723e+01
## midparentHeight  0.39968912 7.631760e-01
## family          -0.01140309 3.426416e-04
```

```
#plot
plot(fitted(female.model),residuals(female.model))
```



```r
confint(female.model)
```

```
##                       2.5 %       97.5 %
## (Intercept)     18.01092352 41.359811429
## midparentHeight  0.34364808  0.668665465
## family          -0.01125063 -0.001113547
```