Final_P3_code

1. Import Dataset

```
data1 <- read.csv("13100420.csv")
```

2.Clean the Dataset

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)


colnames(data1)[1] <- "REF_DATE"
#Remove the unwanted columns
data_clean <- data1[ , ! names(data1) %in% c("UOM_ID", "units", "SCALAR_ID","STATUS","SYMBOL","TERMINAT

#Keep only the rows that are not Percentage
data_clean<-data_clean%>%filter(UOM != "Percentage")%>%select(!UOM)

#Keep only the rows with the detailed marital status of mother
data_clean<-data_clean%>%filter(Marital.status.of.mother != "Total, marital status of mother")%>%filter

#Keep only the rows with the detailed age of mother
data_clean<-data_clean%>%filter(Age.of.mother != "Age of mother, all ages")%>%filter(Age.of.mother != "
a <- data_clean %>% group_by(Marital.status.of.mother)%>%summarize(n())

#Transform Categorical Variable
#Age category
data_clean<-data_clean %>% mutate(Age.Category = ifelse(Age.of.mother == "Age of mother, under 15 years
#Marital status category
data_clean <- data_clean%>% mutate(marital.status.single = ifelse(Marital.status.of.mother == "Marital

#creating a new column which assigns an unique number to each observation
data_clean <- data_clean%>% mutate(observationID = 1:n())
```

3. Summarize data

```
mod1 <- lm(VALUE ~ Age.Category + marital.status.single + marital.status.married + marital.status.divor
summary(mod1)
```

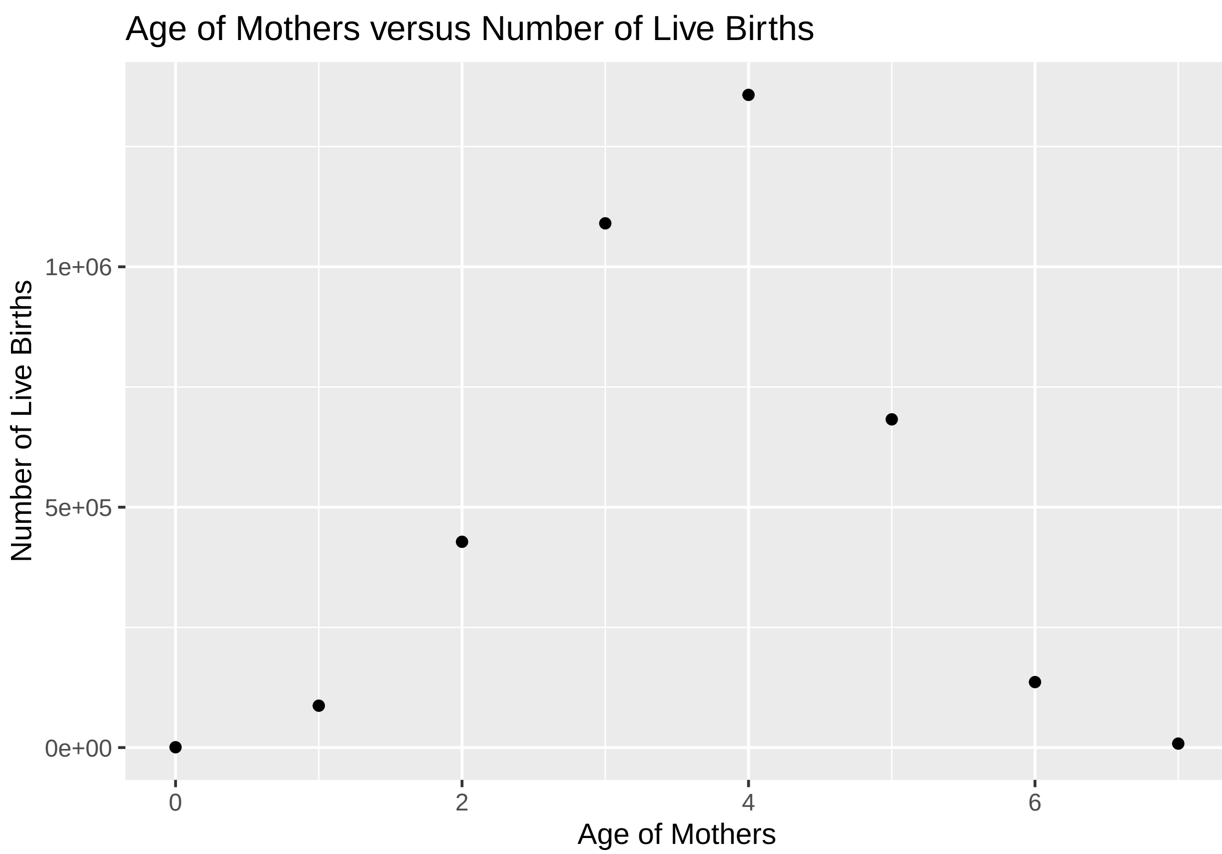
```
##
```

```
## Call:
## lm(formula = VALUE ~ Age.Category + marital.status.single + marital.status.married +
##     marital.status.divorced + marital.status.widowed, data = data_clean)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -29272  -1366    104    620  68462
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -841.6     2077.8  -0.405    0.686
## Age.Category               287.8      334.9   0.860    0.391
## marital.status.single    13645.3     2426.3   5.624 3.35e-08 ***
## marital.status.married   28520.8     2426.3  11.755  < 2e-16 ***
## marital.status.divorced    221.5     2426.3   0.091    0.927
## marital.status.widowed    -139.1     2426.3  -0.057    0.954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16090 on 434 degrees of freedom
## Multiple R-squared:  0.3355, Adjusted R-squared:  0.3278
## F-statistic: 43.82 on 5 and 434 DF,  p-value: < 2.2e-16
```

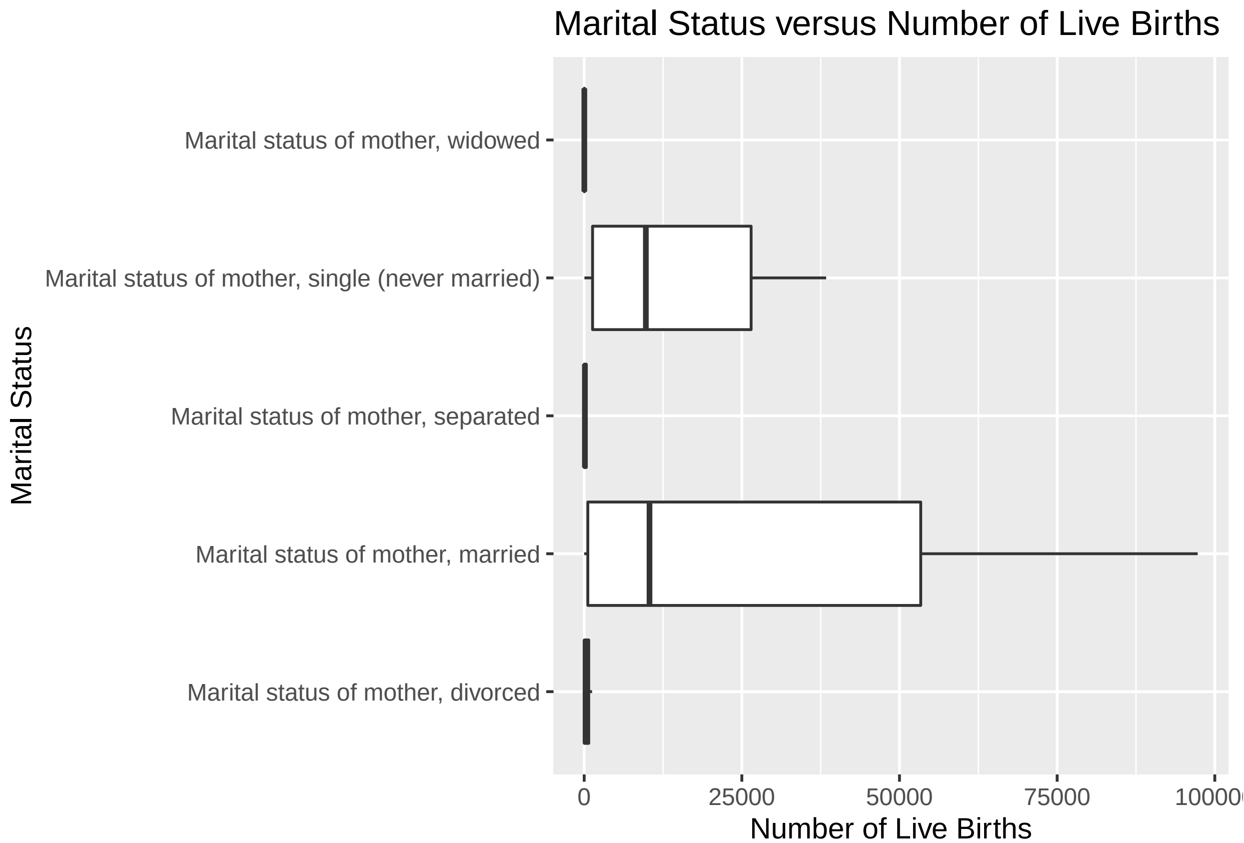
```
#Value by age of mother

age_value_data <- data_clean%>%group_by(Age.of.mother, Age.Category)%>%summarize(TotalLiveBirths = sum(V
```

```
## `summarise()` has grouped output by 'Age.of.mother'. You can override using the `.groups` argument.
```

```
ggplot(age_value_data, aes(x = Age.Category, y = TotalLiveBirths))+geom_point()+ggtitle("Age of Mothers
```




```
#marital status
ggplot(data_clean, aes(x = VALUE, y = Marital.status.of.mother))+geom_boxplot()+ggtitle("Marital Status
```




```r
r <- resid(mod1)
```

4. Model Validation

```r
# create a 50/50 split in the data
set.seed(1)
train <- data_clean[sample(1:nrow(data_clean), 220, replace=F), ]
test <- data_clean[which(!(data_clean$observationID %in% train$observationID)),]

dim(train)
```

```
## [1] 220  10
```

```r
dim(test)
```

```
## [1] 220  10
```

```r
#Check similarity between training dataset and test dataset
mtr <- apply(train[,-c(1,2,3,12)], 2, mean)
sdtr <- apply(train[,-c(1,2,3,12)], 2, sd)

mtest <- apply(test[,-c(1,2,3,12)], 2, mean)
sdtest <- apply(test[,-c(1,2,3,12)], 2, sd)
```

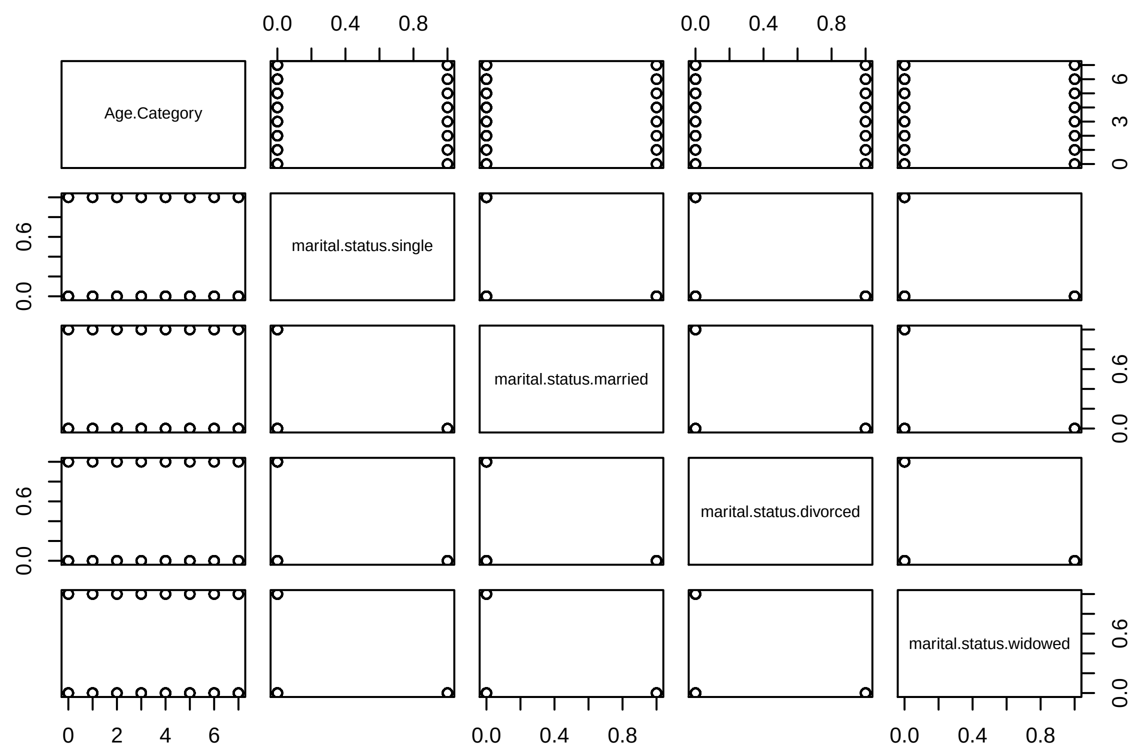
We can take these and add them nicely to a table:

Table 1: Summary statistics in training and test dataset, each of size 220.

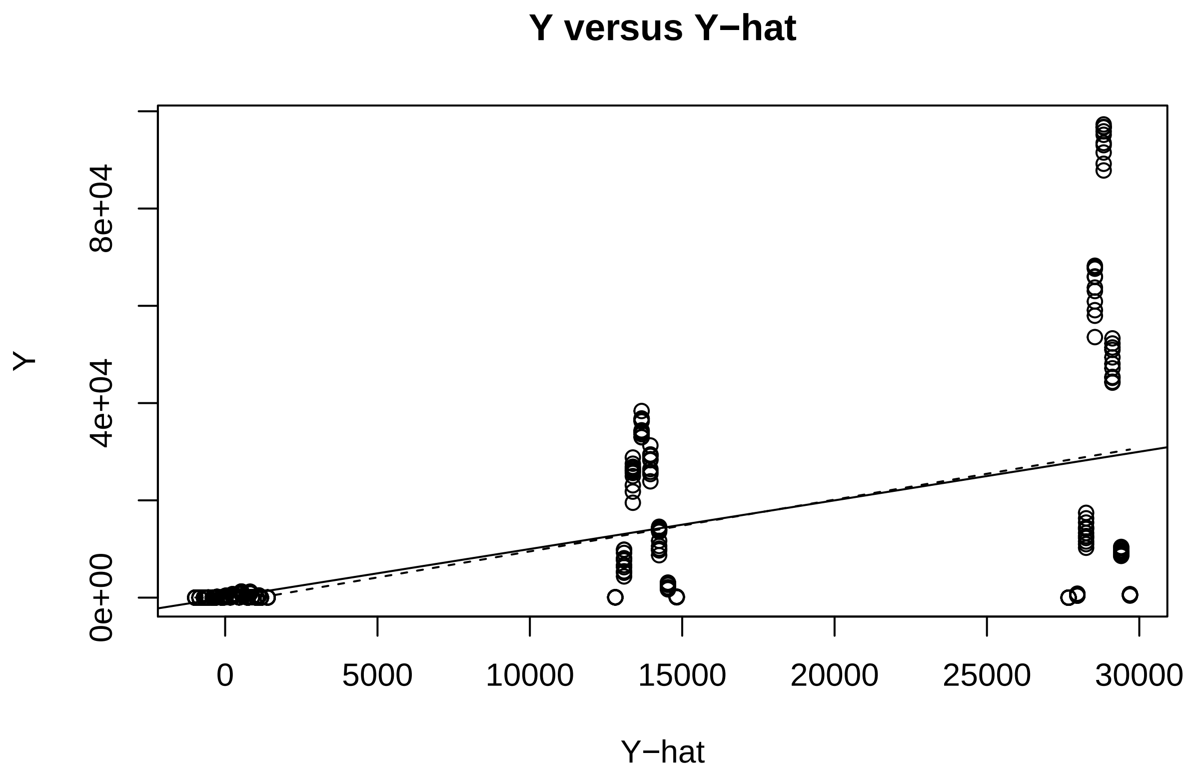
Variable	mean (s.d.) in training	mean (s.d.) in test
VALUE	7252.991 (1.8121751×10^4)	9978.136 (2.0984194×10^4)
Age.Category	3.377 (2.309)	3.623 (2.277)
marital.status.single	0.164 (0.371)	0.236 (0.426)
marital.status.married	0.191 (0.394)	0.209 (0.408)
marital.status.divorced	0.223 (0.417)	0.177 (0.383)
marital.status.widowed	0.205 (0.404)	0.195 (0.397)
observationID	218.755 (129.267)	222.245 (125.29)
NA	NA (NA)	NA (NA)

5. Model Diagnostic

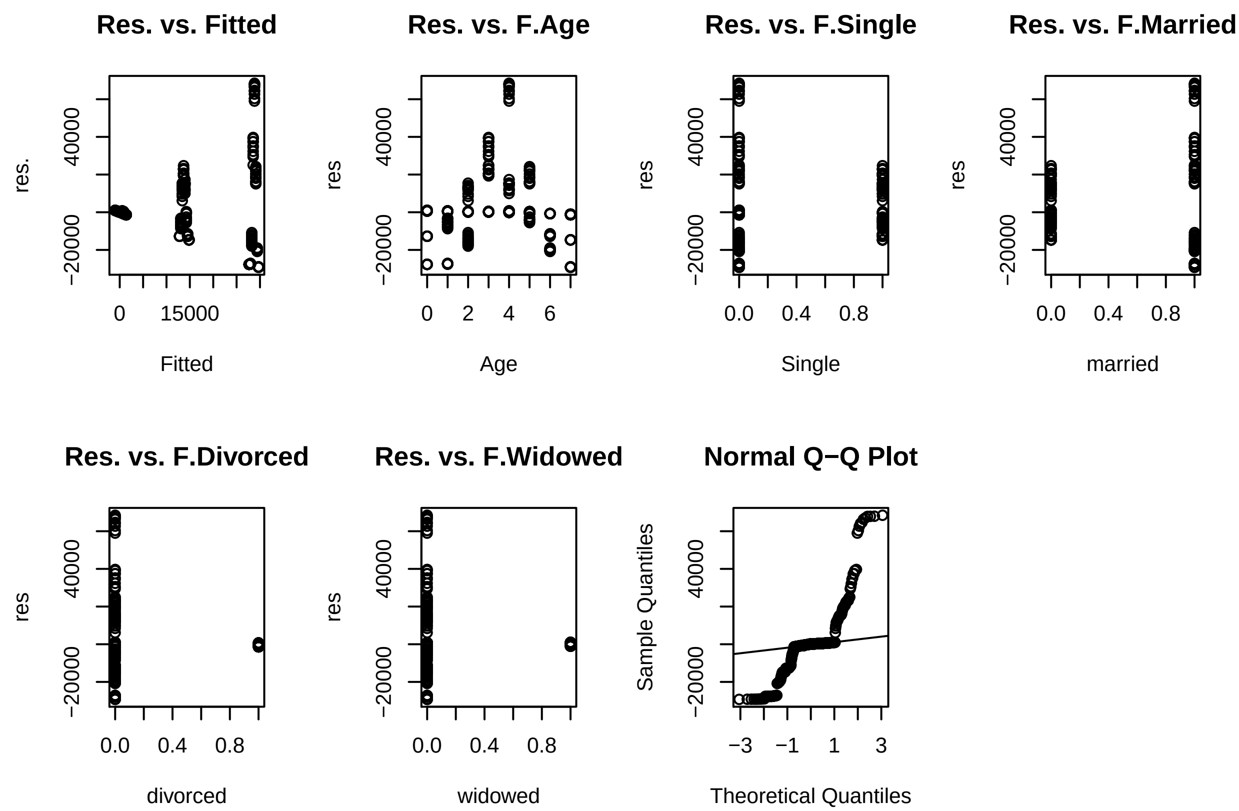
```r
pairs(data_clean[,5:9])
```




```r
plot(data_clean$VALUE ~ fitted(mod1), main="Y versus Y-hat", xlab="Y-hat", ylab="Y")
abline(a = 0, b = 1)
lines(lowess(data_clean$VALUE ~ fitted(mod1)), lty=2)
```




```
par(mfrow=c(2,4))
plot(r ~ fitted(mod1), main="Res. vs. Fitted", xlab="Fitted", ylab="res.")
plot(r ~ data_clean$Age.Category, main="Res. vs. F.Age", xlab="Age", ylab="res")
plot(r ~ data_clean$marital.status.single, main="Res. vs. F.Single", xlab="Single", ylab="res")
plot(r ~ data_clean$marital.status.married, main="Res. vs. F.Married", xlab="married", ylab="res")
plot(r ~ data_clean$marital.status.divorced, main="Res. vs. F.Divorced", xlab="divorced", ylab="res")
plot(r ~ data_clean$marital.status.widowed, main="Res. vs. F.Widowed", xlab="widowed", ylab="res")

qqnorm(r)
qqline(r)
```



6. Transformation

```r
#Transformation

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

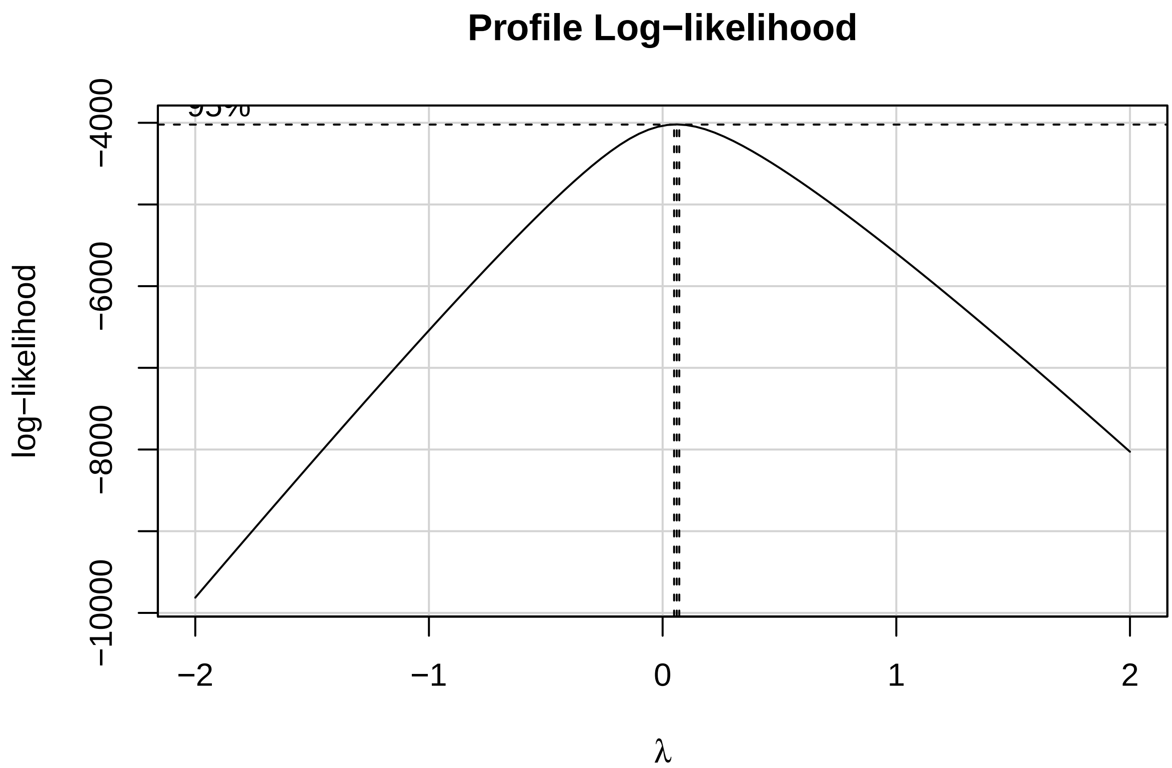
```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```r
data_clean$VALUE <- data_clean$VALUE+0.05
data_clean$Age.Category <- data_clean$Age.Category+0.05

# first Box-Cox on Y
boxCox(mod1)
```




```r
#Transform
mod2 <- lm(log(VALUE+0.01) ~ Age.Category + marital.status.single + marital.status.married + marital.sta
summary(mod2)
```

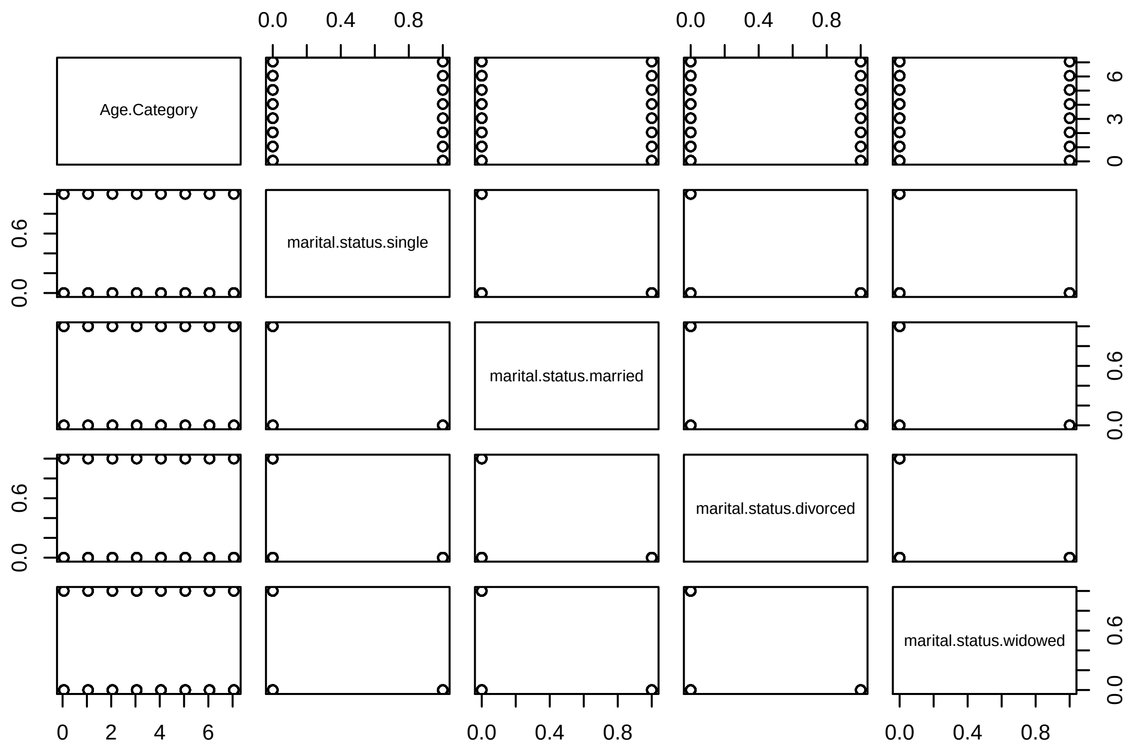
```
##
## Call:
## lm(formula = log(VALUE + 0.01) ~ Age.Category + marital.status.single +
##     marital.status.married + marital.status.divorced + marital.status.widowed,
##     data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7309 -2.0252  0.7109  2.1333  3.5231
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.59832    0.35288   4.529 7.66e-06 ***
## Age.Category             0.56336    0.05661   9.951  < 2e-16 ***
## marital.status.single    4.63113    0.41020  11.290  < 2e-16 ***
## marital.status.married   4.29105    0.41020  10.461  < 2e-16 ***
## marital.status.divorced  0.38692    0.41020   0.943 0.346077
## marital.status.widowed  -1.59725    0.41020  -3.894 0.000114 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.721 on 434 degrees of freedom
```

```
## Multiple R-squared:  0.5165, Adjusted R-squared:  0.5109
## F-statistic: 92.71 on 5 and 434 DF,  p-value: < 2.2e-16
```

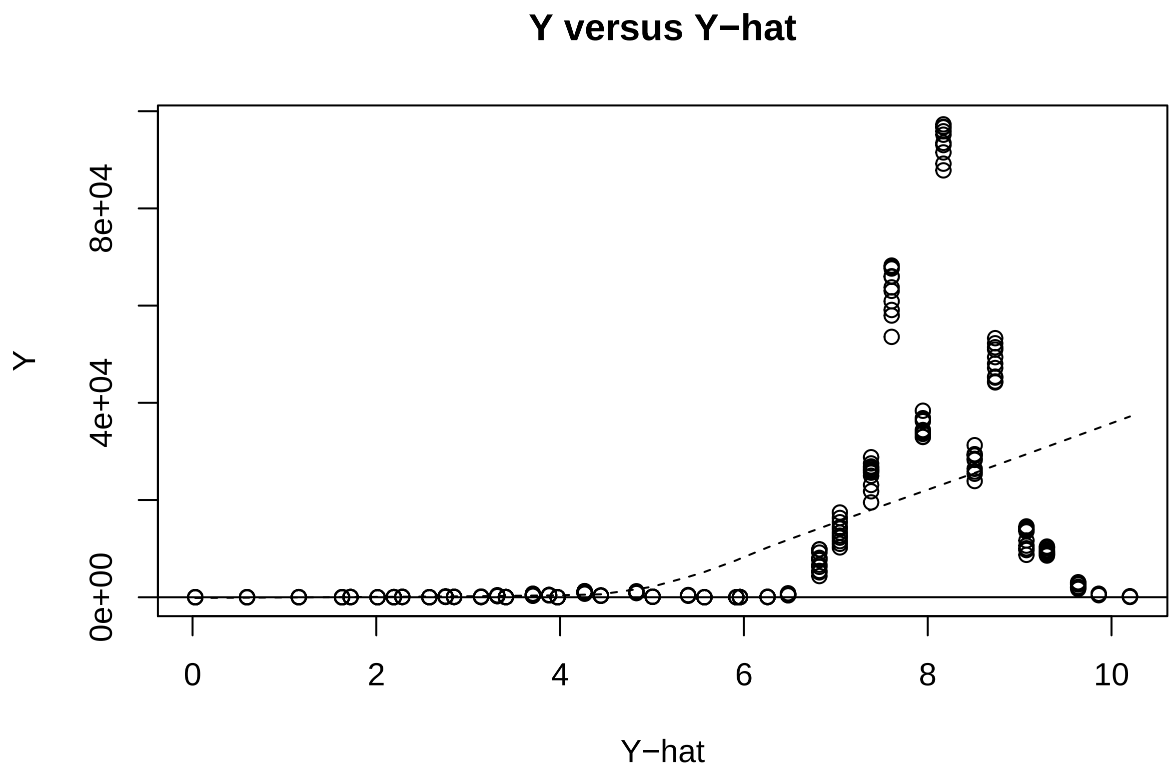
```r
data_transform <- data_clean%>%mutate(log.value = log(VALUE+0.01))
```

7. Model Diagnostic (After Transformation)

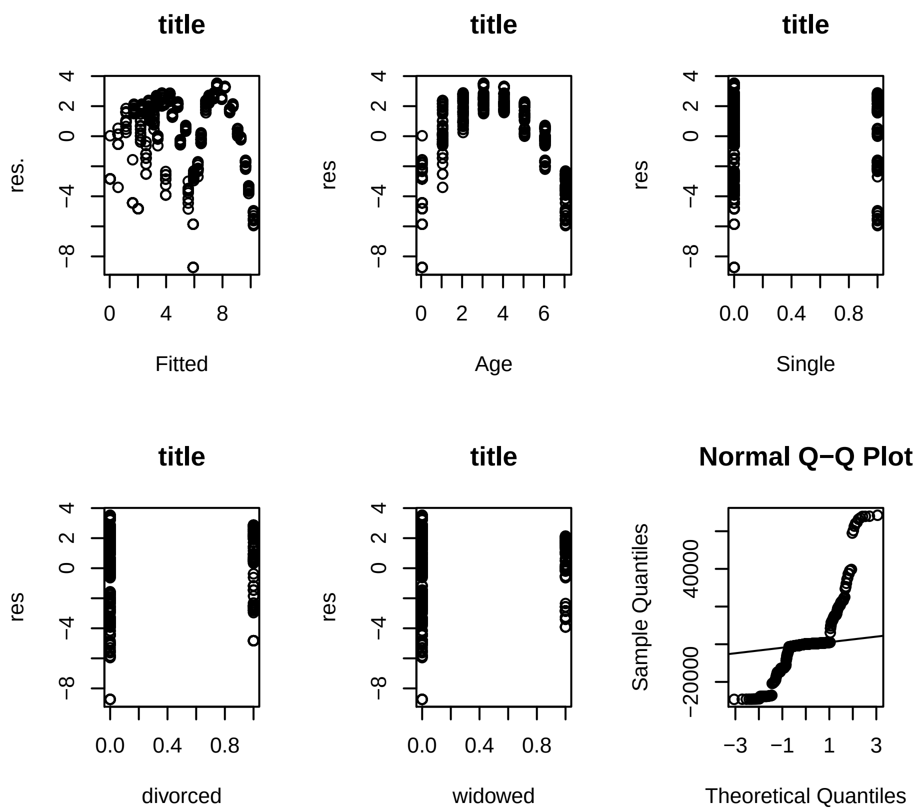
```r
r2 <- resid(mod2)
pairs(data_transform[,5:9])
```




```r
plot(data_transform$VALUE ~ fitted(mod2), main="Y versus Y-hat", xlab="Y-hat", ylab="Y")
abline(a = 0, b = 1)
lines(lowess(data_transform$VALUE ~ fitted(mod2)), lty=2)
```




```
par(mfrow=c(2,4))
plot(r2 ~ fitted(mod2), main="title", xlab="Fitted", ylab="res.")
plot(r2 ~ data_transform$Age.Category, main="title", xlab="Age", ylab="res")
plot(r2 ~ data_transform$marital.status.single, main="title", xlab="Single", ylab="res")
plot(r2 ~ data_transform$marital.status.married, main="title", xlab="married", ylab="res")
plot(r2 ~ data_transform$marital.status.divorced, main="title", xlab="divorced", ylab="res")
plot(r2 ~ data_transform$marital.status.widowed, main="title", xlab="widowed", ylab="res")

qqnorm(r)
qqline(r)
```



8. Model Selection

```r
summary(mod2)
```

```
##
## Call:
## lm(formula = log(VALUE + 0.01) ~ Age.Category + marital.status.single +
##     marital.status.married + marital.status.divorced + marital.status.widowed,
##     data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7309 -2.0252  0.7109  2.1333  3.5231
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.59832    0.35288   4.529 7.66e-06 ***
## Age.Category             0.56336    0.05661   9.951  < 2e-16 ***
## marital.status.single    4.63113    0.41020  11.290  < 2e-16 ***
## marital.status.married   4.29105    0.41020  10.461  < 2e-16 ***
## marital.status.divorced  0.38692    0.41020   0.943 0.346077
## marital.status.widowed  -1.59725    0.41020  -3.894 0.000114 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.721 on 434 degrees of freedom
## Multiple R-squared:  0.5165, Adjusted R-squared:  0.5109
```

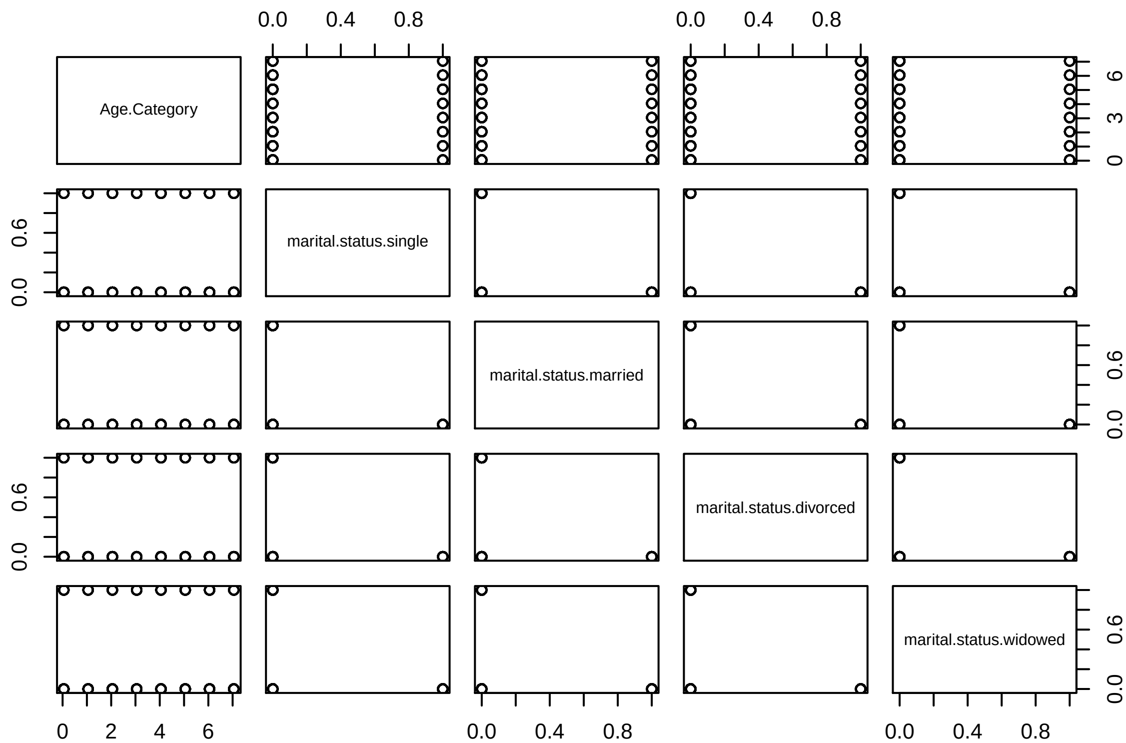
```
## F-statistic: 92.71 on 5 and 434 DF,  p-value: < 2.2e-16
```

```r
#Applying Backward Selection
mod3 <- lm(log(VALUE+0.01) ~ Age.Category + marital.status.single + marital.status.married + marital.sta
summary(mod3)
```

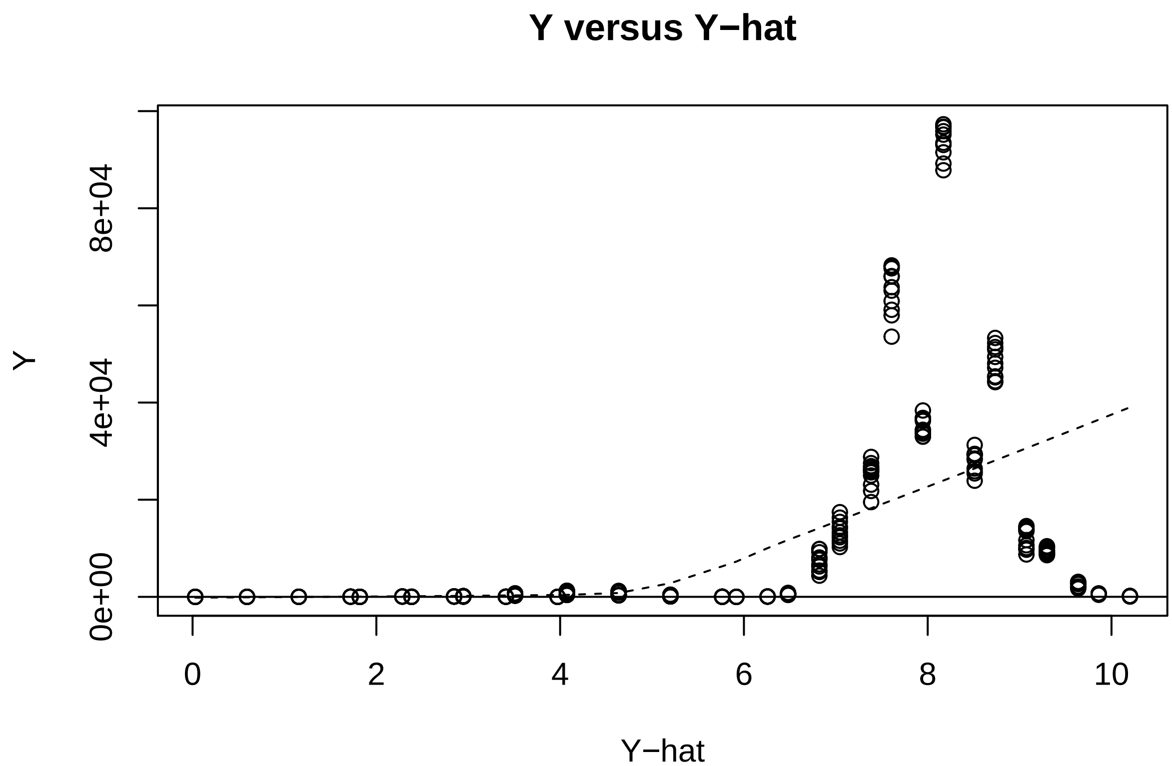
```
##
## Call:
## lm(formula = log(VALUE + 0.01) ~ Age.Category + marital.status.single +
##     marital.status.married + marital.status.widowed, data = data_transform)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7309 -2.0252  0.8112  2.1037  3.5231
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.79178    0.28712   6.241 1.04e-09 ***
## Age.Category            0.56336    0.05661   9.952  < 2e-16 ***
## marital.status.single   4.43767    0.35520  12.494  < 2e-16 ***
## marital.status.married  4.09759    0.35520  11.536  < 2e-16 ***
## marital.status.widowed -1.79071    0.35520  -5.041 6.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.721 on 435 degrees of freedom
## Multiple R-squared:  0.5155, Adjusted R-squared:  0.511
## F-statistic: 115.7 on 4 and 435 DF,  p-value: < 2.2e-16
```

9. Model Diagnostic (After Model Selection)

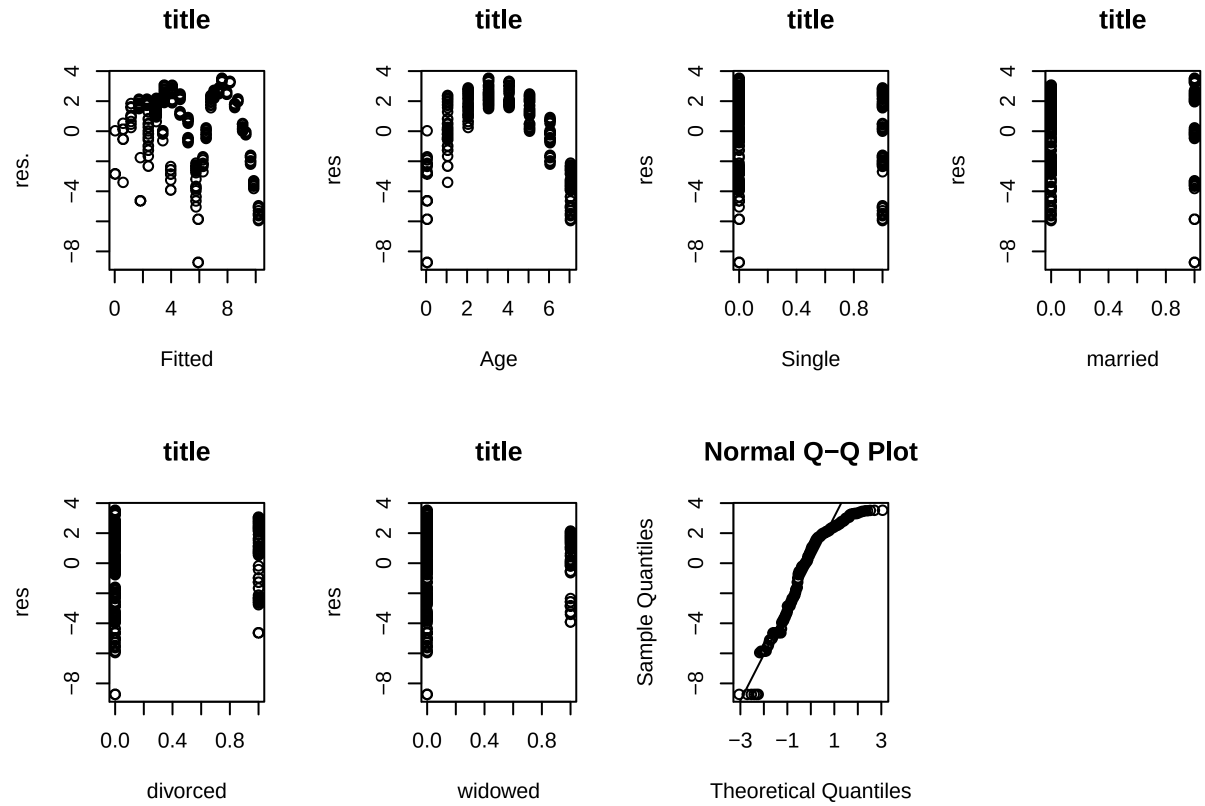
```r
r2 <- resid(mod3)
pairs(data_transform[,5:9])
```




```r
plot(data_transform$VALUE ~ fitted(mod3), main="Y versus Y-hat", xlab="Y-hat", ylab="Y")
abline(a = 0, b = 1)
lines(lowess(data_transform$VALUE ~ fitted(mod3)), lty=2)
```




```r
par(mfrow=c(2,4))
plot(r2 ~ fitted(mod3), main="title", xlab="Fitted", ylab="res.")
plot(r2 ~ data_transform$Age.Category, main="title", xlab="Age", ylab="res")
plot(r2 ~ data_transform$marital.status.single, main="title", xlab="Single", ylab="res")
plot(r2 ~ data_transform$marital.status.married, main="title", xlab="married", ylab="res")
plot(r2 ~ data_transform$marital.status.divorced, main="title", xlab="divorced", ylab="res")
plot(r2 ~ data_transform$marital.status.widowed, main="title", xlab="widowed", ylab="res")

qqnorm(r2)
qqline(r2)
```



10. Check for Leverage Points

```
#leverage point

#Information from the model
n<-length(data_transform$VALUE)
p<-length(coef(mod3))-1

#Calculate the leverage values& compare to cutoff
h<-hatvalues(mod3)
hcut<-2*(p+1)/n

# Which are leverage points
w1<- which(h > hcut)
w1
```

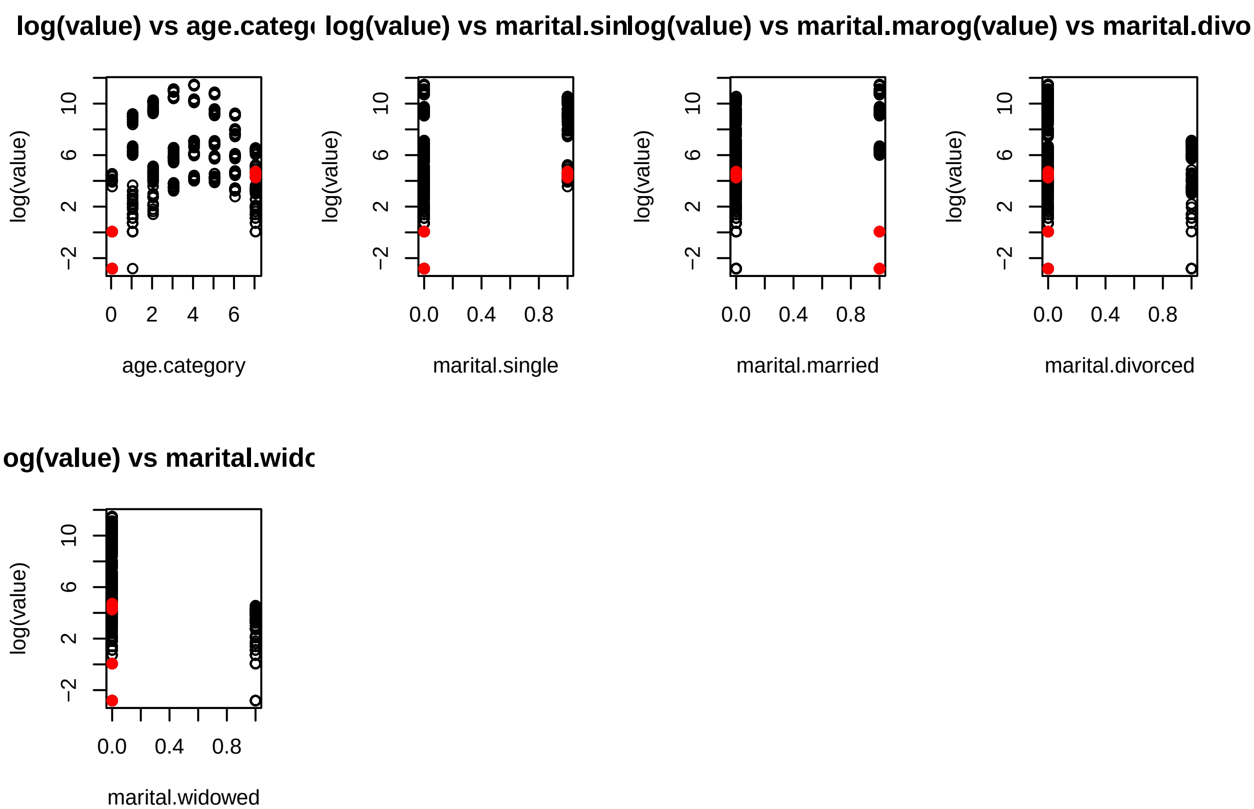
```
## named integer(0)
```

11. Check for Outliers

```
#Outliers
#Calculate standardized residuals and compare to cutoff
r<-rstandard((mod3))
w2<-which(r < -2 |r > 2)
w2
```

```
##    2  36  42  76  82 116 122 156 162 196 202 242 282 322 362 402
##    2  36  42  76  82 116 122 156 162 196 202 242 282 322 362 402
```

```r
par(mfrow=c(2,4))
plot(data_transform[,11]~data_transform[,5], main="log(value) vs age.category", xlab="age.category", yla

points(data_transform[w2,11]~data_transform[w2,5], col="red", pch=19)
plot(data_transform[,11]~data_transform[,6], main="log(value) vs marital.single", xlab="marital.single"

points(data_transform[w2,11]~data_transform[w2,6], col="red", pch=19)
plot(data_transform[,11]~data_transform[,7], main="log(value) vs marital.married", xlab="marital.marrie

points(data_transform[w2,11]~data_transform[w2,7], col="red", pch=19)
plot(data_transform[,11]~data_transform[,8], main="log(value) vs marital.divorced", xlab="marital.divor

points(data_transform[w2,11]~data_transform[w2,8], col="red", pch=19)
plot(data_transform[,11]~data_transform[,9], main="log(value) vs marital.widowed", xlab="marital.widowe

points(data_transform[w2,11]~data_transform[w2,9], col="red", pch=19)
```



12. Check for Influential Points

```r
#Influential Points

#Find the Cooks distance
Dcutoff <- qf(0.5, p+1, n-p-1)
D <- cooks.distance(mod2)
which(D > Dcutoff)
```

```
## named integer(0)
```

```r
# find the DFFITS and compare to cutoff
DFFITScut <- 2*sqrt((p+1)/n)
dfs <- dffits(mod2)
w3 <- which(abs(dfs) > DFFITScut)
w3
```

```
##   2   4   5  36  42  44  45  76  82  84  85 116 120 122 124 125 156 160 162 164
##   2   4   5  36  42  44  45  76  82  84  85 116 120 122 124 125 156 160 162 164
## 165 196 202 204 205 236 242 244 245 276 282 284 285 316 322 324 325 356 362 364
## 165 196 202 204 205 236 242 244 245 276 282 284 285 316 322 324 325 356 362 364
## 396 402 404 405 436
## 396 402 404 405 436
```

```r
# find the DFBETAS and compare to cutoff (notice the dimension of DFBETAS)
DFBETAcut <- 2/sqrt(n)
dfb <- dfbetas(mod2)
w4 <- which(abs(dfb[,1]) > DFBETAcut)
w4
```

```
##   5  42  45  85 125 162 165 202 205 242 245 285 325 362 402 405
##   5  42  45  85 125 162 165 202 205 242 245 285 325 362 402 405
```

```r
w5 <- which(abs(dfb[,2]) > DFBETAcut)
w5
```

```
##   2   4   5  36  37  38  40  42  44  45  76  77  78  80  82  84  85 116 117 120
##   2   4   5  36  37  38  40  42  44  45  76  77  78  80  82  84  85 116 117 120
## 122 124 125 156 157 158 160 162 164 165 196 197 202 204 205 236 240 242 244 245
## 122 124 125 156 157 158 160 162 164 165 196 197 202 204 205 236 240 242 244 245
## 276 280 282 284 285 316 322 324 325 356 360 362 364 396 400 402 404 405 436
## 276 280 282 284 285 316 322 324 325 356 360 362 364 396 400 402 404 405 436
```

```r
w6 <- which(abs(dfb[,3]) > DFBETAcut)
w6
```

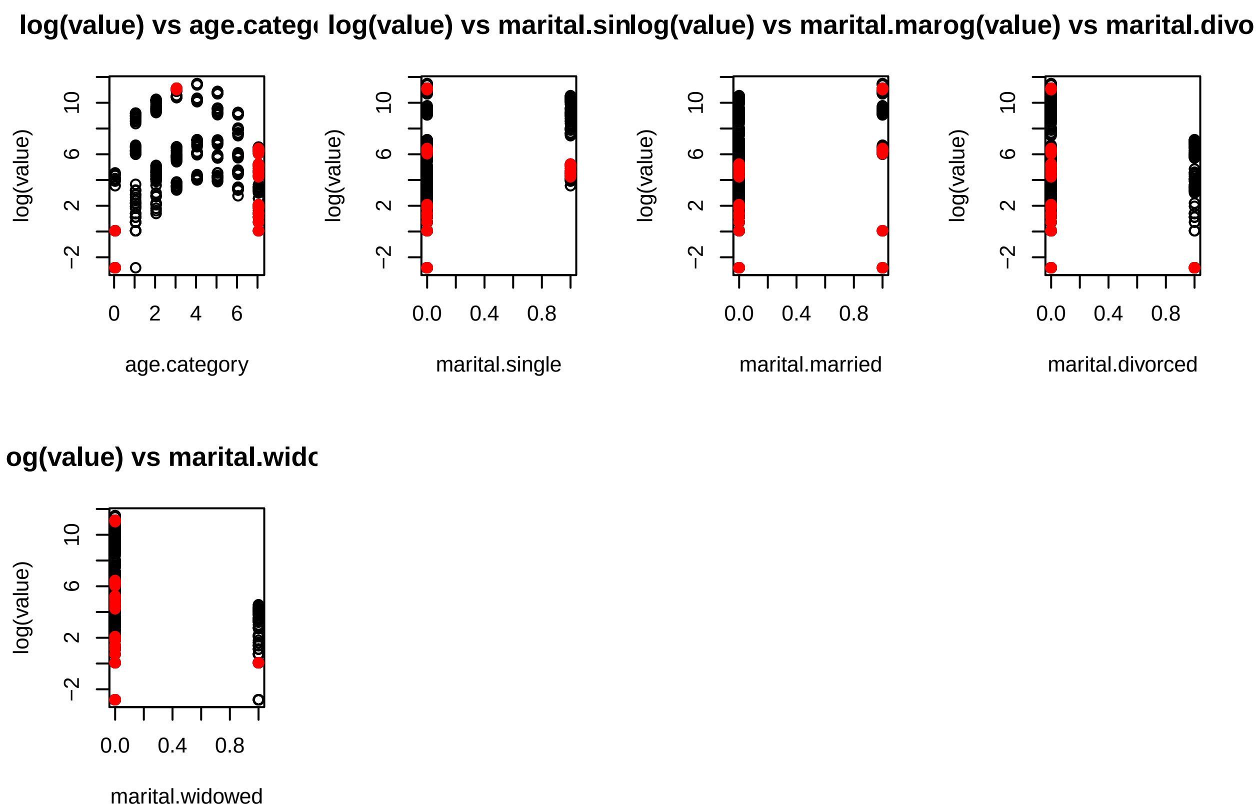
```
##   5  36  40  45  76  80  85 116 120 125 156 160 165 196 200 205 236 240 245 276
##   5  36  40  45  76  80  85 116 120 125 156 160 165 196 200 205 236 240 245 276
## 280 285 316 320 325 356 360 396 400 405 436
## 280 285 316 320 325 356 360 396 400 405 436
```

```r
w7 <- which(abs(dfb[,4]) > DFBETAcut)
w7
```

```
##   2   5  17  37  40  42  45  57  77  80  82  85  97 117 120 122 125 137 157 160
##   2   5  17  37  40  42  45  57  77  80  82  85  97 117 120 122 125 137 157 160
## 162 165 177 197 200 202 205 217 237 240 242 245 257 277 280 282 285 297 317 320
## 162 165 177 197 200 202 205 217 237 240 242 245 257 277 280 282 285 297 317 320
## 322 325 360 362 400 402 405
## 322 325 360 362 400 402 405
```

```r
#Plot them
w <- unique(c(w3, w4, w5, w6, w7))
par(mfrow=c(2,4))

plot(data_transform[,11]~data_transform[,5], main="log(value) vs age.category", xlab="age.category", yla

points(data_transform[w,11]~data_transform[w,5], col="red", pch=19)
plot(data_transform[,11]~data_transform[,6], main="log(value) vs marital.single", xlab="marital.single"
```

```r
points(data_transform[w,11]~data_transform[w,6], col="red", pch=19)
plot(data_transform[,11]~data_transform[,7], main="log(value) vs marital.married", xlab="marital.marrie

points(data_transform[w,11]~data_transform[w,7], col="red", pch=19)
plot(data_transform[,11]~data_transform[,8], main="log(value) vs marital.divorced", xlab="marital.divor

points(data_transform[w,11]~data_transform[w,8], col="red", pch=19)
plot(data_transform[,11]~data_transform[,9], main="log(value) vs marital.widowed", xlab="marital.widowe

points(data_transform[w,11]~data_transform[w,9], col="red", pch=19)
```



13. Check for multi-collinearity

```r
#Check multi collinearity in each model
#install.packages("car")
library(car)
vif(mod1)
```

```
##            Age.Category   marital.status.single  marital.status.married
##                     1.0                     1.6                     1.6
## marital.status.divorced  marital.status.widowed
##                     1.6                     1.6
```

```r
vif(mod2)
```

```
##            Age.Category   marital.status.single  marital.status.married
##                     1.0                     1.6                     1.6
## marital.status.divorced  marital.status.widowed
```

```
##                      1.6                     1.6
```

```r
vif(mod3)
```

```
##           Age.Category  marital.status.single marital.status.married
##                    1.0                    1.2                    1.2
## marital.status.widowed
##                    1.2
```