MAE 430 HW 2 Solution

11/11/2020

Question 1.

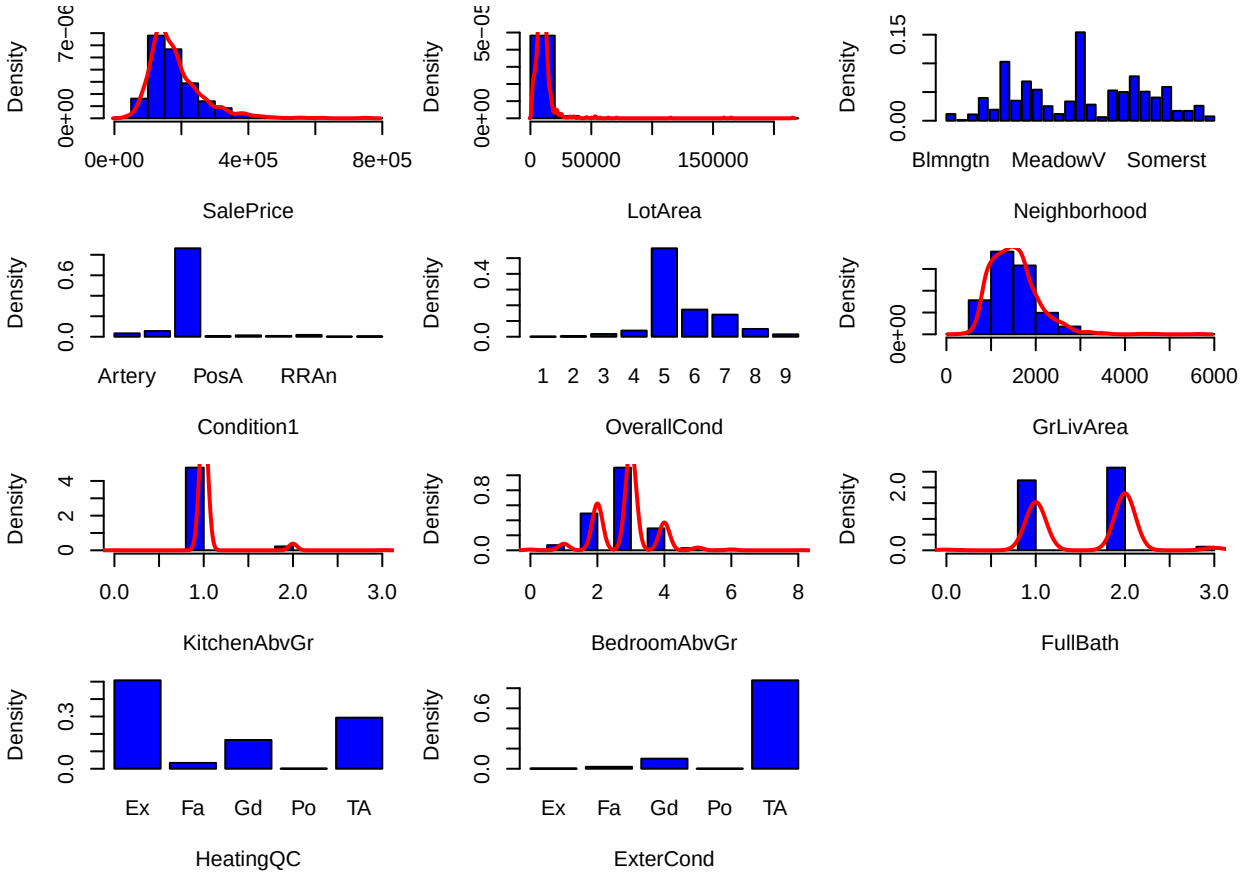
The dataset *train.csv* contains 79 explanatory variables. The data description and csv file can be downloaded directly from kaggle. Your task, as suggested on the kaggle website, is to "build a model to predict final home prices". Before you start the parts below, identify any 10 variables of your choice and write a brief paragraph of why you selected them. Theses are the predictors you will use for solving the problem.

You can choose any 10 variables, but you should give an explanation of your choices. For instance, I chose to include "Condition1" for my regressor because the closer the house is to main roads, the more convenient it is for residents in terms of transporation. Thus, I would expect the sale price of houses to be higher if they are more close to main roads.

(a) Provide a descriptive analysis of your variables. This should include histograms and fitted distributions, quantile plots, correlation plot, boxplots, scatterplots, and statistical summaries (e.g. the five-number summary). All figures must include comments.

Histograms

```
df <- read.csv("train.csv", header = TRUE)

df <- df[, c("SalePrice", "LotArea", "Neighborhood", "Condition1",
    "OverallCond", "GrLivArea", "KitchenAbvGr", "BedroomAbvGr",
    "FullBath", "HeatingQC", "ExterCond")]

# Identify factor variables in train.csv
df <- transform(df, Neighborhood = as.factor(Neighborhood), OverallCond = as.factor(OverallCond),
    HeatingQC = as.factor(HeatingQC), Condition1 = as.factor(Condition1),
    ExterCond = as.factor(ExterCond))

## Histogram Draw histograms of all variables 14 figures
## arranged in 4 by 4
par(mar = c(4, 4, 1, 1))
par(mfrow = c(4, 3))

for (i in 1:11) {
    if (is.numeric(df[, i])) {
        hist(df[, i], freq = FALSE, xlab = colnames(df)[i], ylab = "Density",
            col = "blue", main = NULL)
        lines(density(df[, i]), lwd = 2, col = "red")
    } else {
        barplot(prop.table(table(df[, i])), xlab = colnames(df)[i],
            ylab = "Density", col = "blue")
    }
}
```

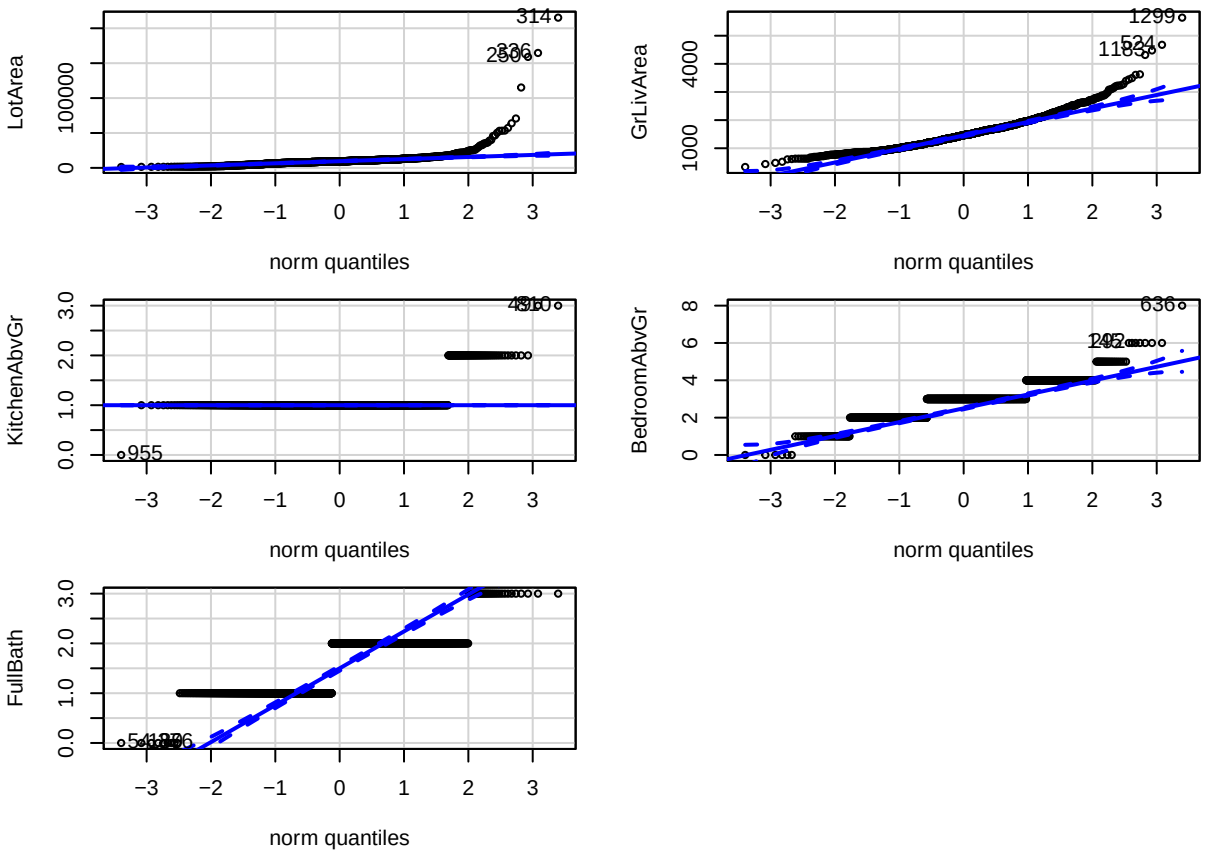


Quantile Plots for Numerical Variables

```
## Q-Q Plots
library(car)
```

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

```
par(mar = c(4, 4, 1, 2))
par(mfrow = c(3, 2))
for (i in 1:11) {
    if (is.numeric(df[, i]) && i != 1) {
        qqPlot(~df[, i], main = NULL, ylab = colnames(df)[i],
            id = list(n = 3))
    }
}
```

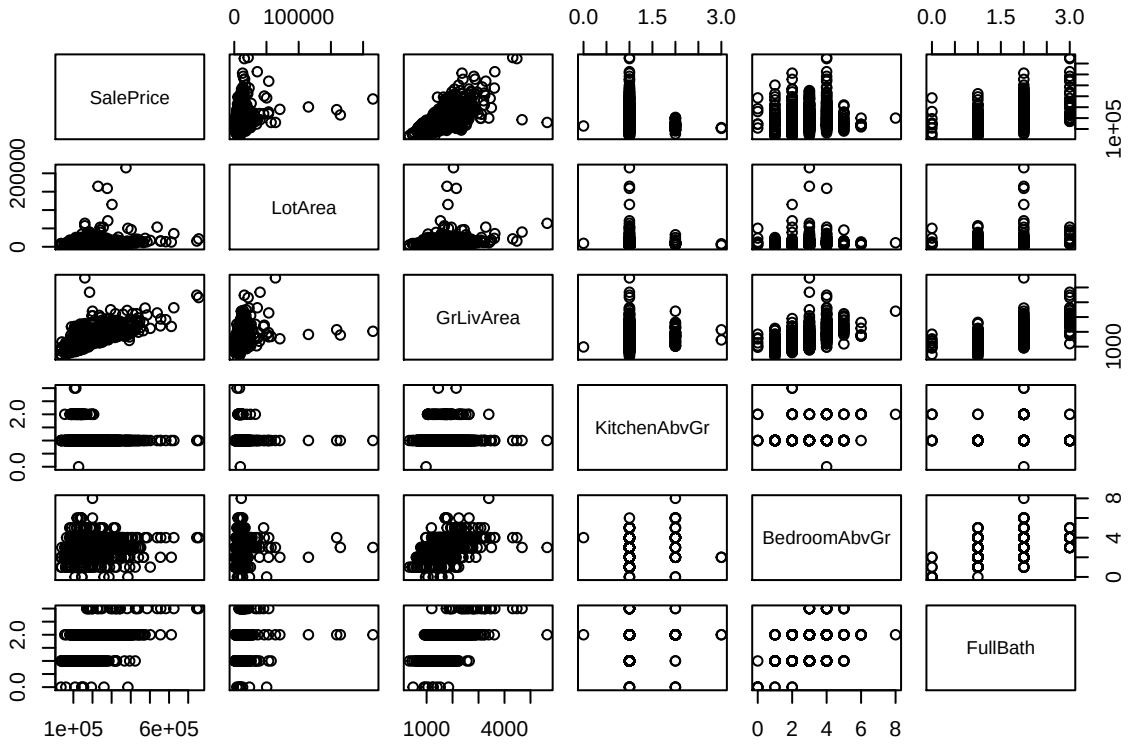


Correlation Plots for Numerical Variables

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

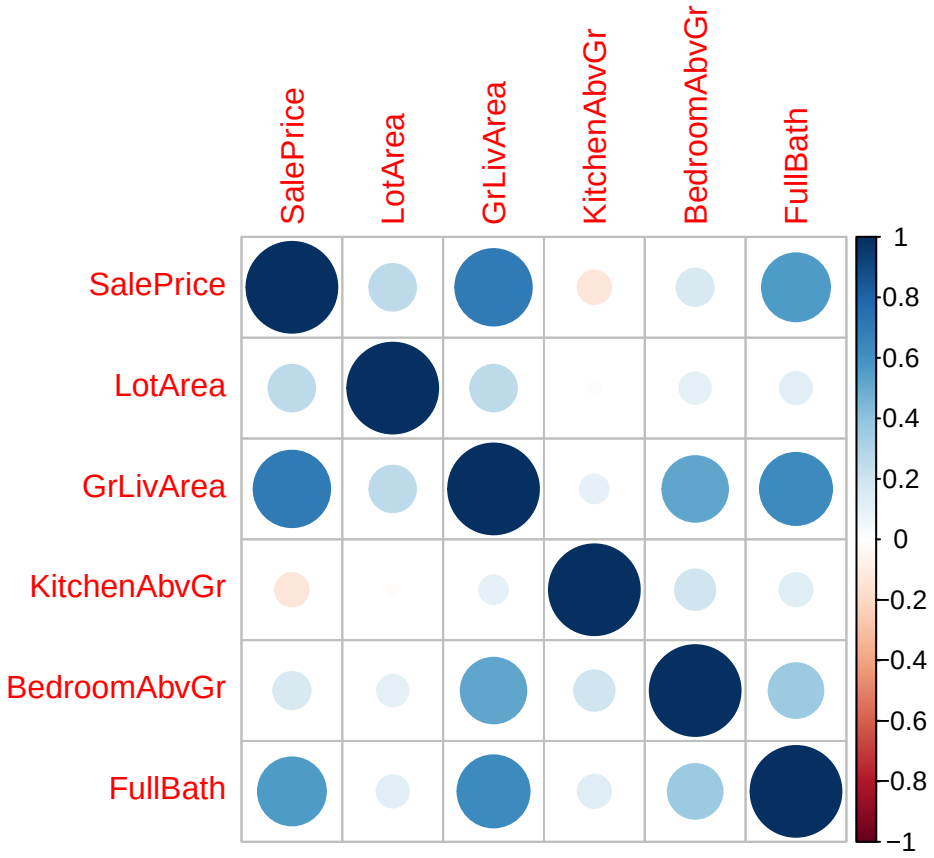
```
## Correlation Plots
plot(df[sapply(df, is.numeric)])
```




```
cor(df[sapply(df, is.numeric)])
```

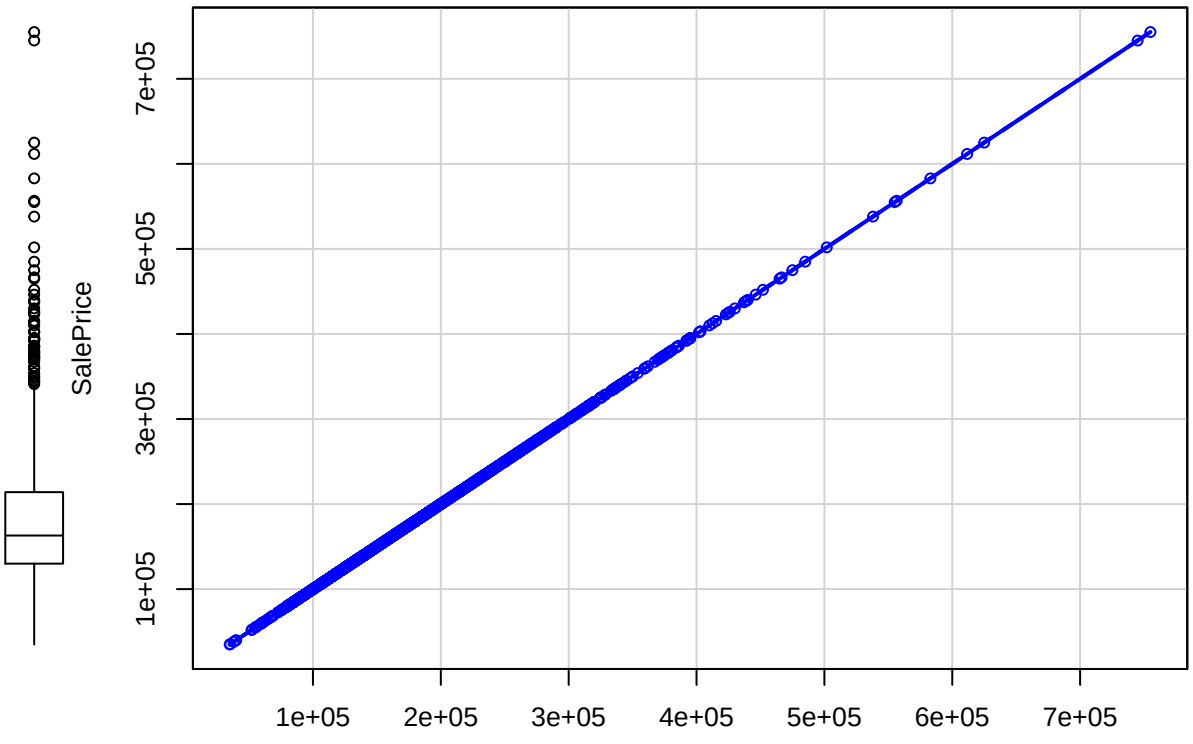
```
##                SalePrice     LotArea GrLivArea KitchenAbvGr BedroomAbvGr
## SalePrice     1.0000000  0.26384335 0.7086245  -0.13590737    0.1682132
## LotArea       0.2638434  1.00000000 0.2631162  -0.01778387    0.1196899
## GrLivArea     0.7086245  0.26311617 1.0000000   0.10006316    0.5212695
## KitchenAbvGr -0.1359074 -0.01778387 0.1000632   1.00000000    0.1985968
## BedroomAbvGr  0.1682132  0.11968991 0.5212695   0.19859676    1.0000000
## FullBath      0.5606638  0.12603063 0.6300116   0.13311521    0.3632520
##                FullBath
## SalePrice     0.5606638
## LotArea       0.1260306
## GrLivArea     0.6300116
## KitchenAbvGr 0.1331152
## BedroomAbvGr 0.3632520
## FullBath      1.0000000
```

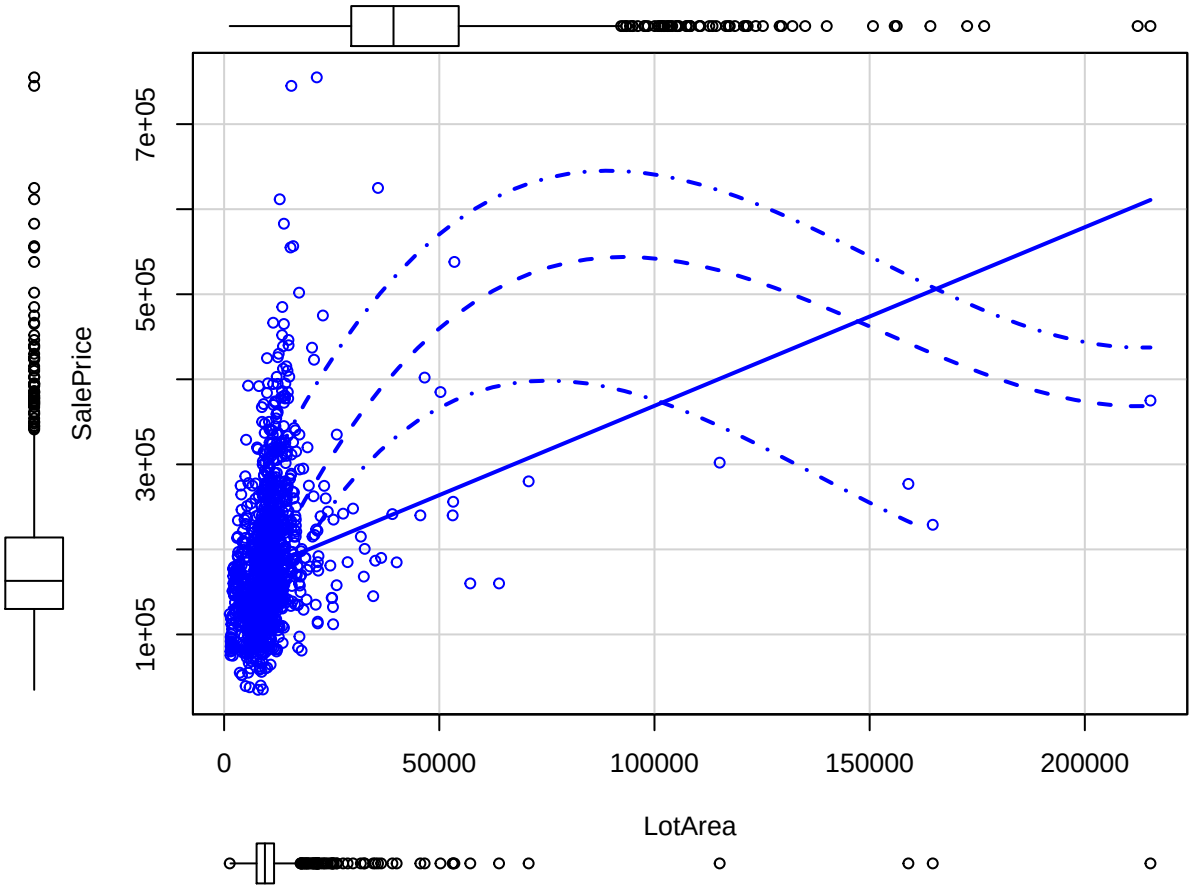
```
corrplot(cor(df[sapply(df, is.numeric)]))
```

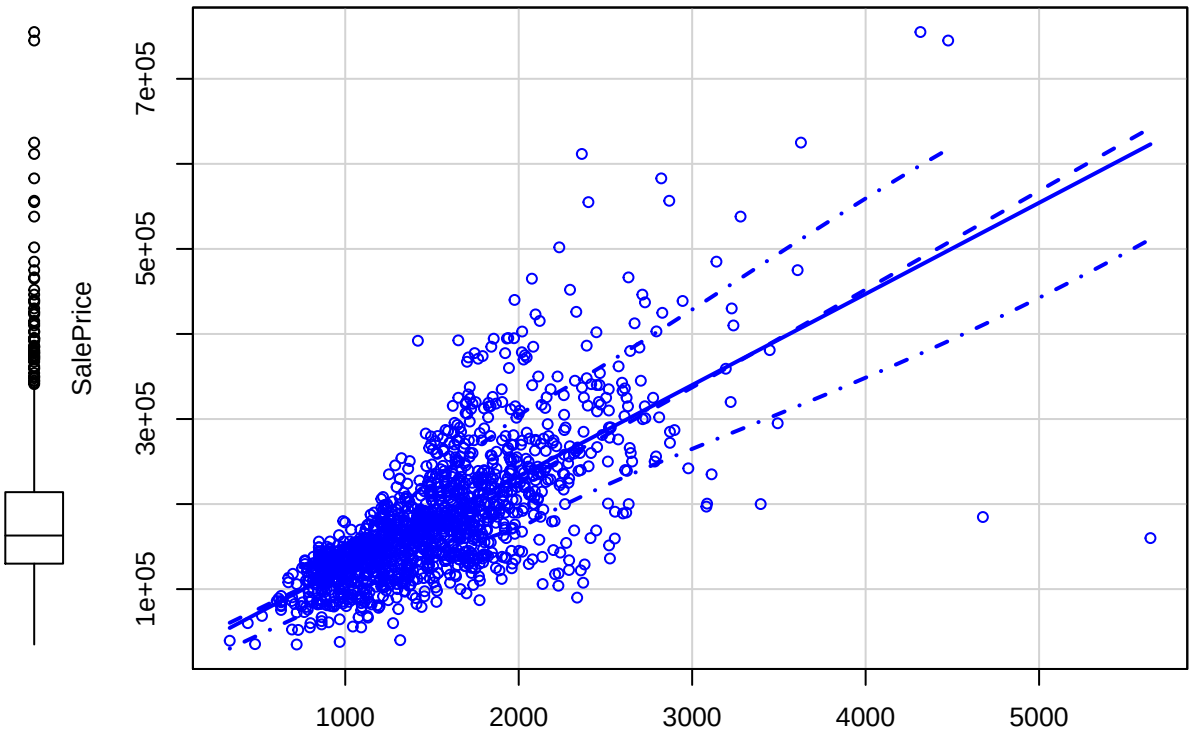


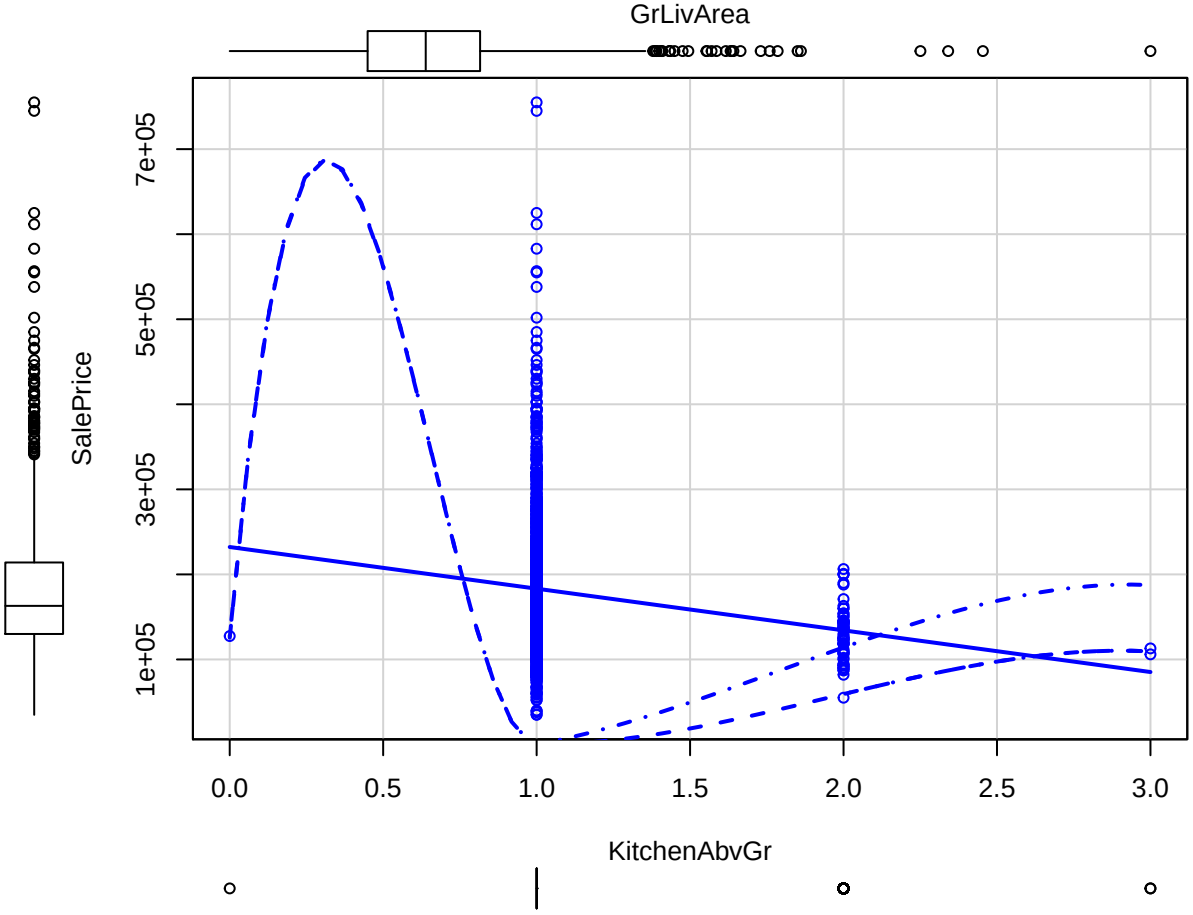
Scatterplots

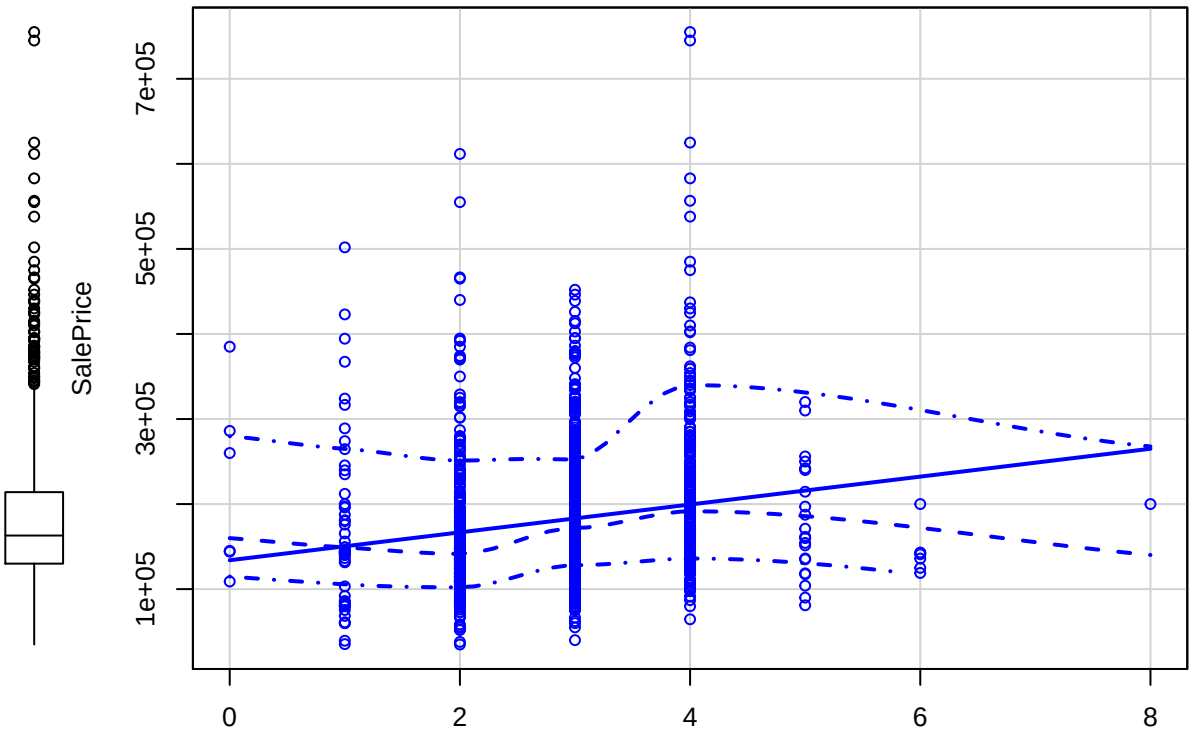
```
## Scatterplots against 'SalePrice'

par(mar = c(4, 4, 1, 2))
for (i in 1:11) {
    if (is.numeric(df[, i])) {
        scatterplot(df[, 1] ~ df[, i], main = NULL, ylab = "SalePrice",
            xlab = colnames(df)[i])
    }
}
```

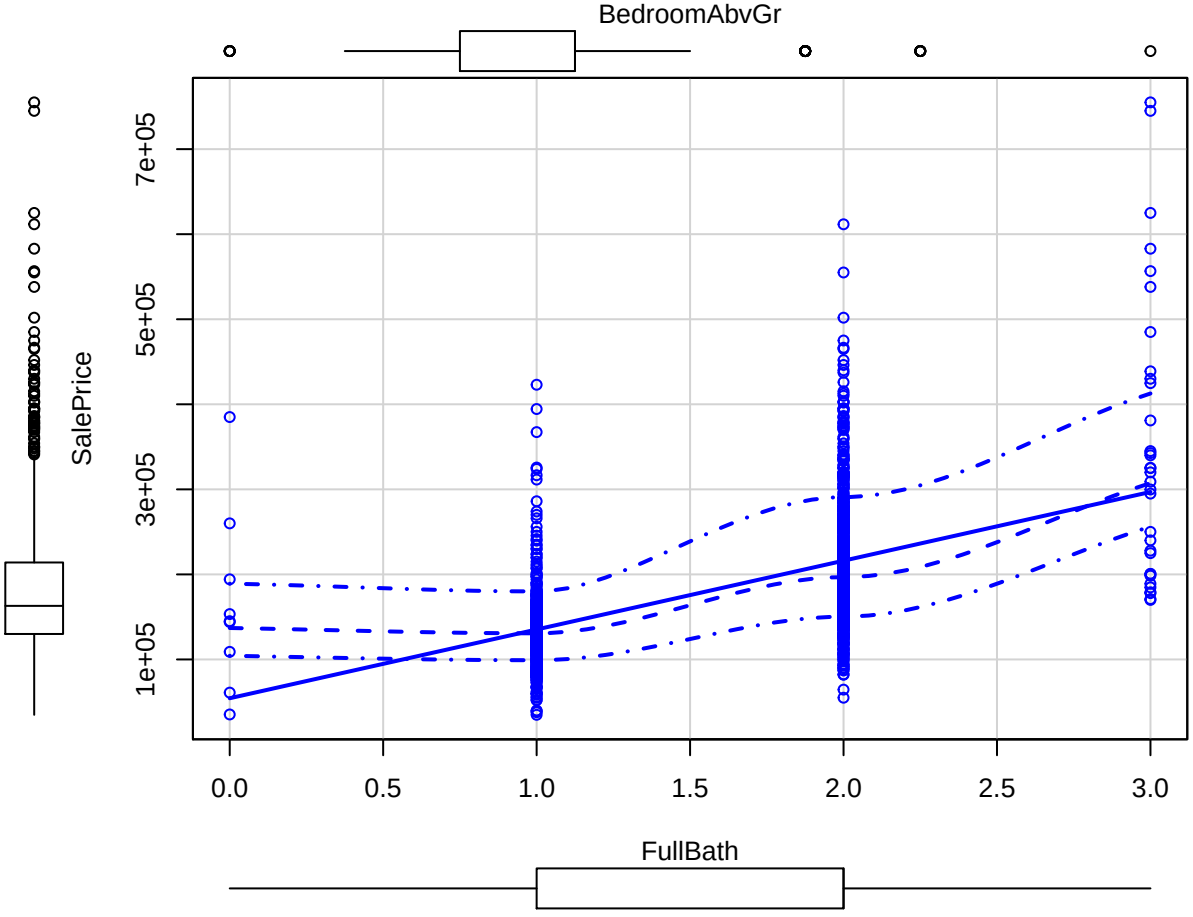

SalePrice
1e+05
3e+05
5e+05
7e+05
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
7e+05
SalePrice


SalePrice
1e+05
3e+05
5e+05
7e+05
0
50000
100000
150000
200000
LotArea


SalePrice
1e+05
3e+05
5e+05
7e+05
1000
2000
3000
4000
5000
GrLivArea


SalePrice
1e+05
3e+05
5e+05
7e+05
0.0
0.5
1.0
1.5
2.0
2.5
3.0
KitchenAbvGr





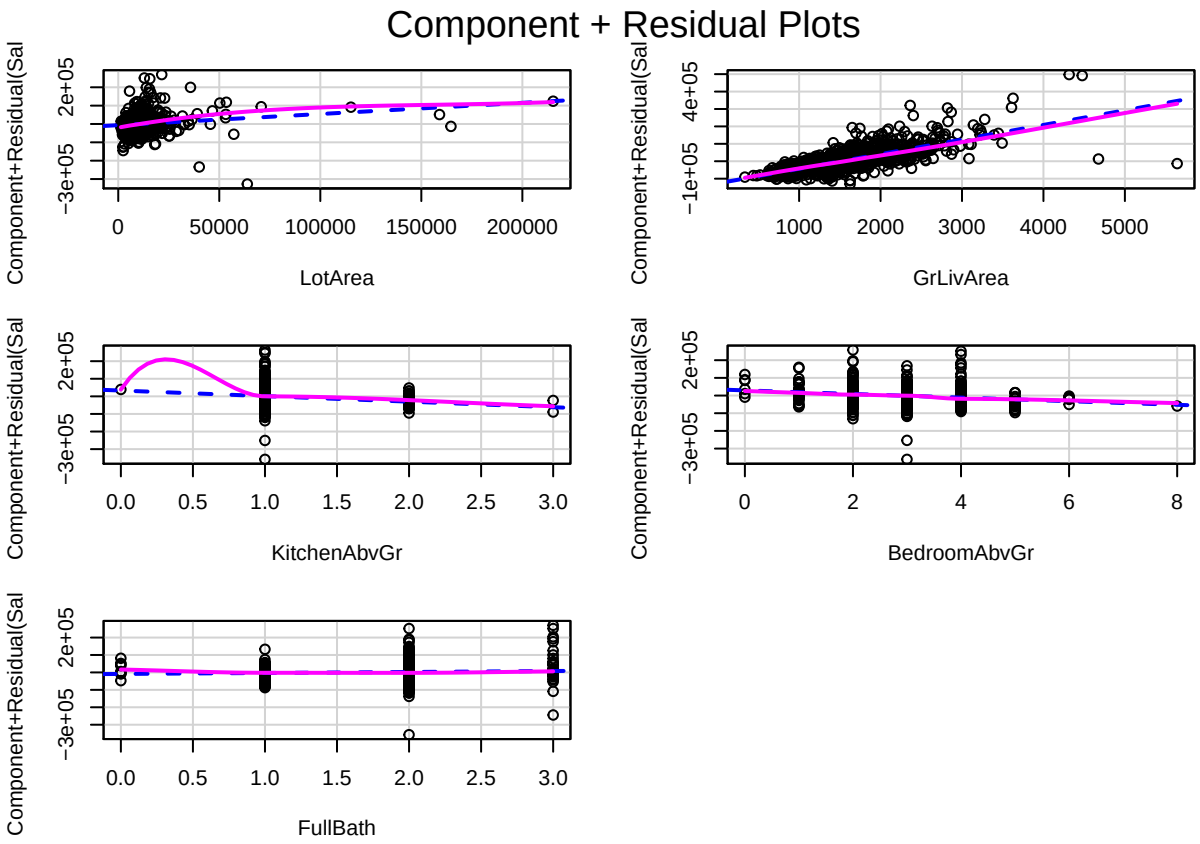
Statistical Summaries

```r
## Five-number Summaries of each variable
summary(df)
```

```
##    SalePrice         LotArea        Neighborhood   Condition1
##  Min.   : 34900   Min.   :  1300   NAmes  :225   Norm   :1260
##  1st Qu.:129975   1st Qu.:  7554   CollgCr:150   Feedr  :  81
##  Median :163000   Median :  9478   OldTown:113   Artery :  48
##  Mean   :180921   Mean   : 10517   Edwards:100   RRAn   :  26
##  3rd Qu.:214000   3rd Qu.: 11602   Somerst: 86   PosN   :  19
##  Max.   :755000   Max.   :215245   Gilbert: 79   RRAe   :  11
##                                    (Other):707   (Other):  15
##   OverallCond    GrLivArea     KitchenAbvGr    BedroomAbvGr
##  5      :821   Min.   : 334   Min.   :0.000   Min.   :0.000
##  6      :252   1st Qu.:1130   1st Qu.:1.000   1st Qu.:2.000
##  7      :205   Median :1464   Median :1.000   Median :3.000
##  8      : 72   Mean   :1515   Mean   :1.047   Mean   :2.866
##  4      : 57   3rd Qu.:1777   3rd Qu.:1.000   3rd Qu.:3.000
##  3      : 25   Max.   :5642   Max.   :3.000   Max.   :8.000
##  (Other): 28
##     FullBath     HeatingQC ExterCond
##  Min.   :0.000   Ex:741    Ex:   3
##  1st Qu.:1.000   Fa: 49    Fa:  28
##  Median :2.000   Gd:241    Gd: 146
##  Mean   :1.565   Po:  1    Po:   1
##  3rd Qu.:2.000   TA:428    TA:1282
##  Max.   :3.000
##
```

(b) For each variable (except indicator ones), test if a transformation to linearity is appropriate, and if so, apply the respective transformation.

```r
df <- na.omit(df)
# Note that numerical predictor variables in our dataset are
# 'LotArea', 'GrLivArea', 'KitchenAbvGr', 'BedroomAbvGr',
# 'FullBath'.  First, take a look at component-plus-residual
# plots to test if transformation to linearity of any
# variable is appropriate.
df.mod <- lm(SalePrice ~ ., data = df)
crPlots(df.mod, terms = ~LotArea + GrLivArea + KitchenAbvGr +
    BedroomAbvGr + FullBath)
```




```
# Comment : From the plots above, it seems like we might need
# to transform some variables such as KitchenAbvGr and
# LotArea. However, we need to be careful that in multiple
# regression, transforming only one variable might lead to
# non-linearity in other variables, so we may need to
# transform more than one variable at the same time.

# Multivariate Box Cox transformation (Yeo-Johnson
# transformations)
summary(linearity <- powerTransform(cbind(LotArea, GrLivArea,
    KitchenAbvGr, BedroomAbvGr, FullBath) ~ 1, data = df, family = "yjPower"))
```

```
## yjPower Transformations to Multinormality
##              Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## LotArea         0.0307        0.00      -0.0142       0.0755
## GrLivArea       0.0601        0.00      -0.0373       0.1576
## KitchenAbvGr   -1.8104       -1.81      -1.9375      -1.6833
## BedroomAbvGr    1.0589        1.00       0.9261       1.1917
## FullBath        0.9684        1.00       0.7603       1.1764
##
##  Likelihood ratio test that all transformation parameters are equal to 0
##                                    LRT df       pval
## LR test, lambda = (0 0 0 0 0) 909.8171  5 < 2.22e-16
```

```
testTransform(linearity, c(0.03, 0.06, -1.8, 1, 1))
```

```
##                                         LRT df  pval
```

```
## LR test, lambda = (0.03 0.06 -1.8 1 1) 0.8891736  5 0.971
```

```r
# Comment : From the p-value of the likelihood ratio test, we
# do not reject the null hypothesis that the stated
# transformation is needed.

# Apply the respective transformation
LotArea_t <- yjPower(df$LotArea, 0.03)
GrLivArea_t <- yjPower(df$GrLivArea, 0.06)
KitchenAbvGr_t <- yjPower(df$KitchenAbvGr, -1.8)
BedroomAbvGr_t <- yjPower(df$BedroomAbvGr, 1)
FullBath_t <- yjPower(df$FullBath, 1)
```

(c) Estimate a multiple linear regression model for *target* that includes all the main effects only (i.e., no interactions nor hhigher order terms). We will use this model as a baseline. Comment on the statistical and economic significance of your estimates. Also, make sure to provide an interpretation of your estimates.

```r
# For interpretation-wise, I am going to only transform
# LotArea and KitchenAbvGr in my base model.  Baseline Model
# Regression
df.base <- lm(SalePrice ~ LotArea + GrLivArea + KitchenAbvGr_t +
    BedroomAbvGr + FullBath + Neighborhood + OverallCond + Condition1 +
    HeatingQC + ExterCond, data = df)
summary(df.base)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + GrLivArea + KitchenAbvGr_t +
##     BedroomAbvGr + FullBath + Neighborhood + OverallCond + Condition1 +
##     HeatingQC + ExterCond, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -359341  -17934    -470   14942  262387
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.707e+05  5.125e+04   3.330 0.000890 ***
## LotArea              6.132e-01  1.131e-01   5.422 6.94e-08 ***
## GrLivArea            8.437e+01  3.111e+00  27.123  < 2e-16 ***
## KitchenAbvGr_t      -3.003e+05  5.254e+04  -5.716 1.33e-08 ***
## BedroomAbvGr        -1.051e+04  1.618e+03  -6.496 1.14e-10 ***
## FullBath             4.349e+03  2.875e+03   1.512 0.130670
## NeighborhoodBlueste -3.288e+04  2.874e+04  -1.144 0.252855
## NeighborhoodBrDale  -4.451e+04  1.362e+04  -3.269 0.001105 **
## NeighborhoodBrkSide -3.295e+04  1.115e+04  -2.955 0.003182 **
## NeighborhoodClearCr -9.627e+03  1.230e+04  -0.783 0.433989
## NeighborhoodCollgCr  6.878e+03  9.875e+03   0.697 0.486212
## NeighborhoodCrawfor -4.263e+03  1.120e+04  -0.381 0.703456
## NeighborhoodEdwards -3.967e+04  1.044e+04  -3.801 0.000150 ***
## NeighborhoodGilbert -8.759e+03  1.041e+04  -0.842 0.400147
```

```
## NeighborhoodIDOTRR  -5.021e+04  1.184e+04  -4.242 2.36e-05 ***
## NeighborhoodMeadowV -4.319e+04  1.347e+04  -3.207 0.001371 **
## NeighborhoodMitchel -1.046e+04  1.103e+04  -0.948 0.343041
## NeighborhoodNAmes   -1.763e+04  1.018e+04  -1.732 0.083413 .
## NeighborhoodNoRidge  5.960e+04  1.137e+04   5.243 1.82e-07 ***
## NeighborhoodNPkVill -1.858e+04  1.603e+04  -1.159 0.246761
## NeighborhoodNridgHt  8.457e+04  1.027e+04   8.234 4.08e-16 ***
## NeighborhoodNWAmes  -1.107e+04  1.086e+04  -1.020 0.308028
## NeighborhoodOldTown -5.017e+04  1.056e+04  -4.751 2.23e-06 ***
## NeighborhoodSawyer  -1.629e+04  1.087e+04  -1.498 0.134229
## NeighborhoodSawyerW -2.817e+03  1.074e+04  -0.262 0.793076
## NeighborhoodSomerst  2.280e+04  1.016e+04   2.244 0.024964 *
## NeighborhoodStoneBr  8.111e+04  1.205e+04   6.732 2.43e-11 ***
## NeighborhoodSWISU   -5.574e+04  1.249e+04  -4.462 8.78e-06 ***
## NeighborhoodTimber   2.358e+04  1.141e+04   2.067 0.038951 *
## NeighborhoodVeenker  3.866e+04  1.501e+04   2.575 0.010135 *
## OverallCond2         3.271e+04  4.323e+04   0.757 0.449306
## OverallCond3         1.313e+04  3.960e+04   0.332 0.740148
## OverallCond4         1.927e+04  3.940e+04   0.489 0.624850
## OverallCond5         3.620e+04  3.924e+04   0.922 0.356470
## OverallCond6         3.758e+04  3.925e+04   0.957 0.338542
## OverallCond7         4.470e+04  3.929e+04   1.138 0.255382
## OverallCond8         4.824e+04  3.943e+04   1.224 0.221341
## OverallCond9         6.986e+04  4.028e+04   1.734 0.083050 .
## Condition1Feedr     -7.231e+03  7.249e+03  -0.998 0.318678
## Condition1Norm       9.398e+03  5.937e+03   1.583 0.113631
## Condition1PosA       8.718e+03  1.498e+04   0.582 0.560715
## Condition1PosN       9.186e+02  1.069e+04   0.086 0.931549
## Condition1RRAe      -7.137e+03  1.392e+04  -0.513 0.608250
## Condition1RRAn       9.162e+03  9.794e+03   0.935 0.349698
## Condition1RRNe      -1.320e+04  2.802e+04  -0.471 0.637743
## Condition1RRNn       3.219e+04  1.817e+04   1.771 0.076718 .
## HeatingQCFa         -1.341e+04  6.046e+03  -2.218 0.026742 *
## HeatingQCGd         -1.082e+04  3.114e+03  -3.474 0.000528 ***
## HeatingQCPo          1.493e+04  4.062e+04   0.368 0.713278
## HeatingQCTA         -1.221e+04  2.864e+03  -4.264 2.15e-05 ***
## ExterCondFa         -2.156e+04  2.524e+04  -0.854 0.393036
## ExterCondGd         -1.568e+04  2.379e+04  -0.659 0.510072
## ExterCondPo         -7.070e+04  4.945e+04  -1.430 0.153000
## ExterCondTA         -1.405e+04  2.395e+04  -0.587 0.557475
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37920 on 1406 degrees of freedom
## Multiple R-squared:  0.7805, Adjusted R-squared:  0.7722
## F-statistic: 94.31 on 53 and 1406 DF,  p-value: < 2.2e-16
```

```
# Comment : It seems like LotArea, Neighborhood, GrLivArea,
# KitchenAbvGR_t, BedroomAbvGr, and HeatingQC are
# statistically significant. + Need to make some comments on
# economic significance of the estimates. For instance,
# although the estimated coefficient of LotArea is
# statistically significant, the effect of an increase in lot
# size by 1 square feet increases the house price by $0.6132.
```

```r
# This seems to have almost no economic significance.
```

(d) Identify if there are any outliers worth removing, If so, remove them but justify your reason for doing so and reestimate your model.

```r
# Perform an outlier test known as the Bonferroni-corrected
# t-test
outlierTest(df.base)  # The small p-values sugges that we should remove all those outliers.
```

```
##        rstudent unadjusted p-value Bonferonni p
## 1299 -10.584593         3.0447e-25   4.4362e-22
## 692    7.212854         8.9264e-13   1.3006e-09
## 524   -7.136411         1.5302e-12   2.2295e-09
## 899    6.716744         2.6910e-11   3.9208e-08
## 1183   6.676120         3.5225e-11   5.1322e-08
## 804    5.304387         1.3119e-07   1.9114e-04
## 441    5.024766         5.6880e-07   8.2874e-04
## 1170   5.018020         5.8877e-07   8.5784e-04
## 1182   4.934955         8.9741e-07   1.3075e-03
## 1047   4.606869         4.4581e-06   6.4954e-03
```

```r
df_no_outlier <- df[-c(1299, 692, 524, 899, 1183, 804, 441, 1170,
    1182, 1047), ]
KitchenAbvGr_t <- KitchenAbvGr_t[-c(1299, 692, 524, 899, 1183,
    804, 441, 1170, 1182, 1047)]

df_new <- cbind(df_no_outlier[, -7], KitchenAbvGr_t)

# Reestimate the model in part (c)
df.base.no_outlier <- lm(SalePrice ~ LotArea + GrLivArea + KitchenAbvGr_t +
    BedroomAbvGr + FullBath + Neighborhood + OverallCond + Condition1 +
    HeatingQC + ExterCond, data = df_new)
summary(df.base.no_outlier)
```

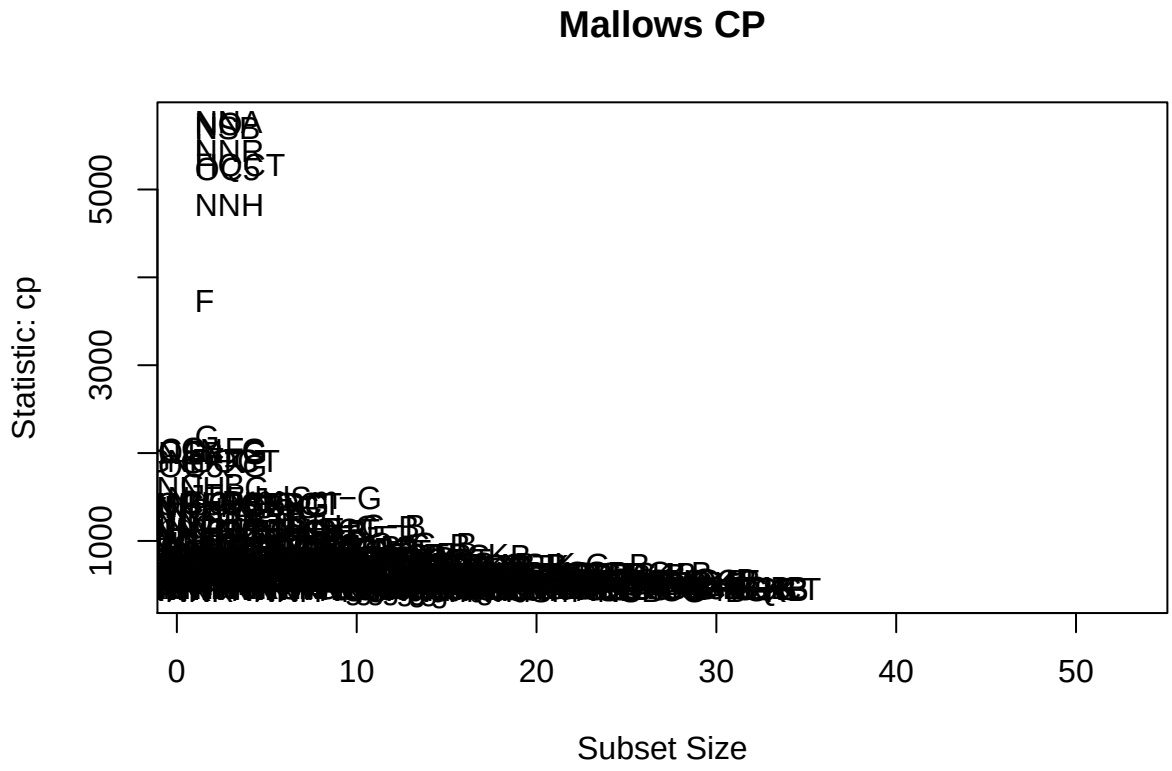
```
##
## Call:
## lm(formula = SalePrice ~ LotArea + GrLivArea + KitchenAbvGr_t +
##     BedroomAbvGr + FullBath + Neighborhood + OverallCond + Condition1 +
##     HeatingQC + ExterCond, data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -129906  -17533    -265   15123  168056
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.543e+05  4.327e+04   3.565 0.000376 ***
## LotArea              7.657e-01  9.623e-02   7.956 3.64e-15 ***
## GrLivArea            8.986e+01  2.866e+00  31.351  < 2e-16 ***
## KitchenAbvGr_t      -2.934e+05  4.438e+04  -6.611 5.41e-11 ***
## BedroomAbvGr        -1.199e+04  1.406e+03  -8.528  < 2e-16 ***
```

```
## FullBath             6.723e+01  2.444e+03   0.028 0.978059
## NeighborhoodBlueste -2.948e+04  2.426e+04  -1.215 0.224476
## NeighborhoodBrDale  -4.337e+04  1.150e+04  -3.771 0.000170 ***
## NeighborhoodBrkSide -3.313e+04  9.421e+03  -3.517 0.000451 ***
## NeighborhoodClearCr -1.469e+04  1.039e+04  -1.414 0.157691
## NeighborhoodCollgCr  6.782e+03  8.337e+03   0.814 0.416042
## NeighborhoodCrawfor -9.593e+03  9.494e+03  -1.010 0.312471
## NeighborhoodEdwards -3.254e+04  8.849e+03  -3.678 0.000244 ***
## NeighborhoodGilbert -8.704e+03  8.784e+03  -0.991 0.321945
## NeighborhoodIDOTRR  -5.007e+04  1.000e+04  -5.006 6.25e-07 ***
## NeighborhoodMeadowV -4.117e+04  1.138e+04  -3.617 0.000308 ***
## NeighborhoodMitchel -1.042e+04  9.317e+03  -1.118 0.263553
## NeighborhoodNAmes   -1.820e+04  8.601e+03  -2.117 0.034473 *
## NeighborhoodNoRidge  3.804e+04  9.675e+03   3.932 8.83e-05 ***
## NeighborhoodNPkVill -1.408e+04  1.354e+04  -1.040 0.298547
## NeighborhoodNridgHt  7.407e+04  8.701e+03   8.513  < 2e-16 ***
## NeighborhoodNWAmes  -1.152e+04  9.169e+03  -1.256 0.209308
## NeighborhoodOldTown -5.034e+04  8.919e+03  -5.644 2.01e-08 ***
## NeighborhoodSawyer  -1.702e+04  9.184e+03  -1.853 0.064044 .
## NeighborhoodSawyerW -3.645e+03  9.064e+03  -0.402 0.687649
## NeighborhoodSomerst  2.285e+04  8.574e+03   2.665 0.007783 **
## NeighborhoodStoneBr  7.171e+04  1.026e+04   6.991 4.21e-12 ***
## NeighborhoodSWISU   -5.601e+04  1.055e+04  -5.310 1.27e-07 ***
## NeighborhoodTimber   2.103e+04  9.635e+03   2.183 0.029219 *
## NeighborhoodVeenker  3.477e+04  1.268e+04   2.743 0.006173 **
## OverallCond2         4.473e+04  3.649e+04   1.226 0.220464
## OverallCond3         2.344e+04  3.342e+04   0.701 0.483147
## OverallCond4         3.117e+04  3.326e+04   0.937 0.348808
## OverallCond5         5.047e+04  3.313e+04   1.523 0.127912
## OverallCond6         4.958e+04  3.314e+04   1.496 0.134874
## OverallCond7         5.733e+04  3.317e+04   1.729 0.084100 .
## OverallCond8         6.193e+04  3.329e+04   1.860 0.063028 .
## OverallCond9         8.291e+04  3.400e+04   2.439 0.014863 *
## Condition1Feedr     -1.963e+03  6.138e+03  -0.320 0.749153
## Condition1Norm       9.959e+03  5.012e+03   1.987 0.047115 *
## Condition1PosA       9.410e+03  1.265e+04   0.744 0.457139
## Condition1PosN       1.607e+04  9.180e+03   1.750 0.080293 .
## Condition1RRAe      -6.191e+03  1.175e+04  -0.527 0.598426
## Condition1RRAn       9.465e+03  8.266e+03   1.145 0.252372
## Condition1RRNe      -1.129e+04  2.365e+04  -0.478 0.633066
## Condition1RRNn       3.248e+04  1.534e+04   2.118 0.034361 *
## HeatingQCFa         -1.420e+04  5.106e+03  -2.781 0.005491 **
## HeatingQCGd         -1.089e+04  2.629e+03  -4.143 3.64e-05 ***
## HeatingQCPo          1.635e+04  3.428e+04   0.477 0.633555
## HeatingQCTA         -1.286e+04  2.420e+03  -5.315 1.24e-07 ***
## ExterCondFa         -2.051e+04  2.130e+04  -0.963 0.335860
## ExterCondGd         -1.571e+04  2.008e+04  -0.782 0.434161
## ExterCondPo         -6.866e+04  4.173e+04  -1.645 0.100137
## ExterCondTA         -1.280e+04  2.021e+04  -0.633 0.526694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32000 on 1396 degrees of freedom
## Multiple R-squared:  0.8139, Adjusted R-squared:  0.8068
```

```
## F-statistic: 115.2 on 53 and 1396 DF,  p-value: < 2.2e-16
```

(e) Use Mallows Cp for identifying which terms you will keep in the model and also test for multicollinearity.

```
library(leaps)

ss = regsubsets(SalePrice ~ ., method = c("exhaustive"), really.big = T,
    nbest = 10, nvmax = 10, data = df_new)
subsets(ss, statistic = "cp", legend = F, main = "Mallows CP",
    col = "steelblue4")
```




```
##                      Abbreviation
## LotArea                         L
## NeighborhoodBlueste           NgB
## NeighborhoodBrDale            NBD
## NeighborhoodBrkSide           NBS
## NeighborhoodClearCr    NghbrhdClrC
## NeighborhoodCollgCr    NghbrhdCllC
## NeighborhoodCrawfor           NgC
## NeighborhoodEdwards            NE
## NeighborhoodGilbert            NG
## NeighborhoodIDOTRR             NI
## NeighborhoodMeadowV           NMV
## NeighborhoodMitchel           NgM
## NeighborhoodNAmes             NNA
## NeighborhoodNoRidge           NNR
## NeighborhoodNPkVill           NNP
```

```
## NeighborhoodNridgHt          NNH
## NeighborhoodNWAmes           NNW
## NeighborhoodOldTown           NO
## NeighborhoodSawyer     NghbrhdSw
## NeighborhoodSawyerW         NgSW
## NeighborhoodSomerst    NghbrhdSm
## NeighborhoodStoneBr          NSB
## NeighborhoodSWISU           NSWI
## NeighborhoodTimber            NT
## NeighborhoodVeenker           NV
## Condition1Feedr              C1F
## Condition1Norm               C1N
## Condition1PosA              C1PA
## Condition1PosN              C1PN
## Condition1RRAe        Condtn1RRA
## Condition1RRAn        Cndtn1RRAn
## Condition1RRNe        Condtn1RRN
## Condition1RRNn        Cndtn1RRNn
## OverallCond2                OC2
## OverallCond3                OC3
## OverallCond4                OC4
## OverallCond5                OC5
## OverallCond6                OC6
## OverallCond7                OC7
## OverallCond8                OC8
## OverallCond9                OC9
## GrLivArea                     G
## BedroomAbvGr                  B
## FullBath                      F
## HeatingQCFa                HQCF
## HeatingQCGd                HQCG
## HeatingQCPo                HQCP
## HeatingQCTA                HQCT
## ExterCondFa                 ECF
## ExterCondGd                 ECG
## ExterCondPo                 ECP
## ExterCondTA                 ECT
## KitchenAbvGr_t                K
```

```
subsets(ss, statistic = "cp", legend = F, main = "Mallows CP",
    col = "steelblue4", ylim = c(350, 410))
```



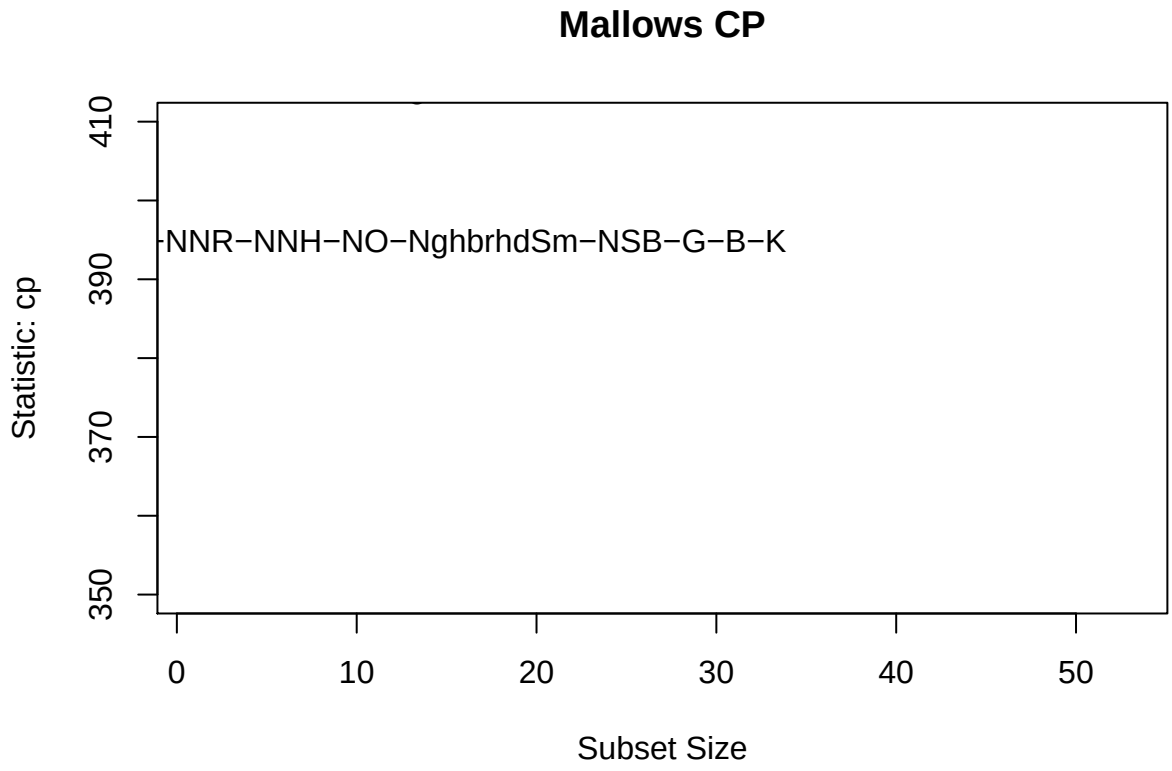
```
##                     Abbreviation
## LotArea                        L
## NeighborhoodBlueste          NgB
## NeighborhoodBrDale           NBD
## NeighborhoodBrkSide          NBS
## NeighborhoodClearCr  NghbrhdClrC
## NeighborhoodCollgCr  NghbrhdCllC
## NeighborhoodCrawfor          NgC
## NeighborhoodEdwards           NE
## NeighborhoodGilbert           NG
## NeighborhoodIDOTRR            NI
## NeighborhoodMeadowV          NMV
```

```
## NeighborhoodMitchel          NgM
## NeighborhoodNAmes            NNA
## NeighborhoodNoRidge          NNR
## NeighborhoodNPkVill          NNP
## NeighborhoodNridgHt          NNH
## NeighborhoodNWAmes           NNW
## NeighborhoodOldTown           NO
## NeighborhoodSawyer     NghbrhdSw
## NeighborhoodSawyerW         NgSW
## NeighborhoodSomerst    NghbrhdSm
## NeighborhoodStoneBr          NSB
## NeighborhoodSWISU           NSWI
## NeighborhoodTimber            NT
## NeighborhoodVeenker           NV
## Condition1Feedr              C1F
## Condition1Norm               C1N
## Condition1PosA              C1PA
## Condition1PosN              C1PN
## Condition1RRAe        Condtn1RRA
## Condition1RRAn        Cndtn1RRAn
## Condition1RRNe        Condtn1RRN
## Condition1RRNn        Cndtn1RRNn
## OverallCond2                OC2
## OverallCond3                OC3
## OverallCond4                OC4
## OverallCond5                OC5
## OverallCond6                OC6
## OverallCond7                OC7
## OverallCond8                OC8
## OverallCond9                OC9
## GrLivArea                     G
## BedroomAbvGr                  B
## FullBath                      F
## HeatingQCFa                HQCF
## HeatingQCGd                HQCG
## HeatingQCPo                HQCP
## HeatingQCTA                HQCT
## ExterCondFa                 ECF
## ExterCondGd                 ECG
## ExterCondPo                 ECP
## ExterCondTA                 ECT
## KitchenAbvGr_t                K
```

```
s = summary(ss)
coef(ss, which.min(s$cp))
```

```
##         (Intercept)             LotArea NeighborhoodCollgCr
##        2.055707e+05        9.674641e-01        3.133071e+04
## NeighborhoodNoRidge NeighborhoodNridgHt NeighborhoodOldTown
##        5.366640e+04        9.335389e+04       -3.145845e+04
## NeighborhoodSomerst NeighborhoodStoneBr           GrLivArea
##        4.602485e+04        8.632436e+04        1.032291e+02
##        BedroomAbvGr      KitchenAbvGr_t
##       -1.683014e+04       -3.878938e+05
```

```
# Comment : Using Mallows Cp, we will keep 5 variables, which
# are LotArea, Neighborhood, GrLivArea, BedroomAbvGr,
# KitchenAbvGr_t.


# Test for multicollinearity
library(broom)
```




```
df.mod_Cp <- lm(SalePrice ~ LotArea + Neighborhood + GrLivArea +
    BedroomAbvGr + KitchenAbvGr_t, data = df_new)
vif(df.mod_Cp)  # No evidence of multicollinearity because of low VIFs
```

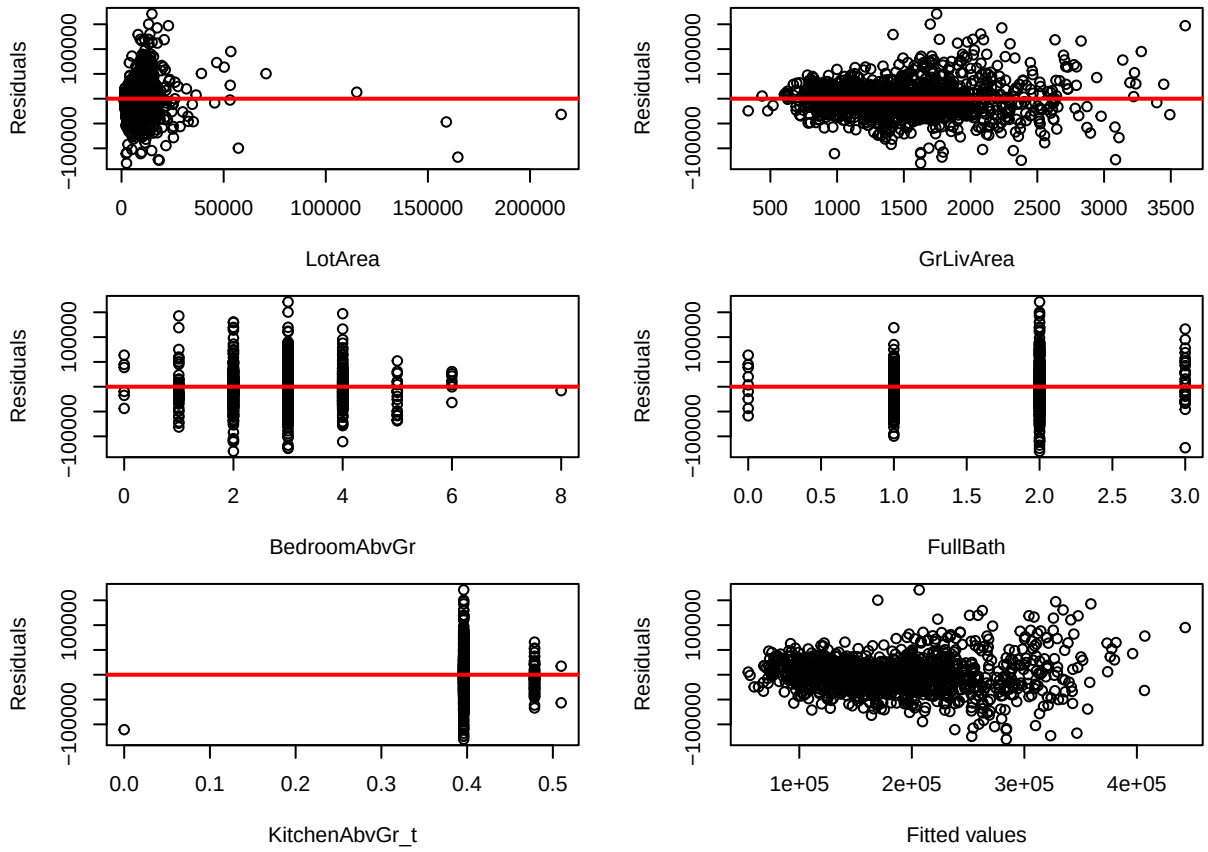
```
##                    GVIF Df GVIF^(1/(2*Df))
## LotArea        1.257885  1        1.121555
## Neighborhood   2.147493 24        1.016050
## GrLivArea      2.206797  1        1.485529
## BedroomAbvGr   1.745625  1        1.321221
## KitchenAbvGr_t 1.088570  1        1.043345
```

(f) For your model in part (e), plot the respective residuals vs. x, and y vs. $\hat{y}$, and comment on your results.

```
# Plot the respective residuals vs. x
par(mar = c(4, 4, 1, 2))
par(mfrow = c(3, 2), new = TRUE)
```

```
## Warning in par(mfrow = c(3, 2), new = TRUE): calling par(new=TRUE) with no
## plot
```

```r
for (i in 1:11) {
    if (is.numeric(df_new[, i]) && i != 1) {
        plot(df_new[, i], df.mod_Cp$residuals, ylab = "Residuals",
            xlab = colnames(df_new)[i])
        abline(h = 0, col = "red", lwd = 2)
    }
}

# Plot residuals vs y_hat
plot(df.mod_Cp$fitted.values, df.mod_Cp$residuals, xlab = "Fitted values",
    ylab = "Residuals")
```




```r
# Comment : From the plots of residuals against fitted y,the
# second assumption of E[e|X] = 0 seems to be satisfied, but
# the error is heteroskedastic.
```

(g) Using AIC and BIC for model comparison, identify which model is better, (c) or (e). Why?

```r
# AIC
AIC(df.base, df.mod_Cp)
```

```
## Warning in AIC.default(df.base, df.mod_Cp): models are not all fitted to
## the same number of observations
```

```
##           df      AIC
## df.base   55 34984.39
## df.mod_Cp 30 34355.90
```

```
# BIC
BIC(df.base, df.mod_Cp)
```

```
## Warning in BIC.default(df.base, df.mod_Cp): models are not all fitted to
## the same number of observations
```

```
##           df      BIC
## df.base   55 35275.13
## df.mod_Cp 30 34514.27
```

```
# Comment : Both AIC and BIC are lower for the model selected
# in (e), df.mod_Cp. Thus, in terms of predictive power,
# df.mod_Cp is a better model based on AIC, BIC, and Mallows
# Cp.
```

(h) Estimate a model based on (g) that includes interaction terms and if needed, any higher power terms. Comment on the performance of this model compared to your other two models.

```
# Note that depending on each individual's intuition, the
# interaction terms added can be different, so there is no
# one definitive answer.  Here, I include interaction term
# between NeighborHood and BedroomAbvGr to the new model.

df.mod_i <- lm(SalePrice ~ LotArea + Neighborhood + GrLivArea +
    BedroomAbvGr + KitchenAbvGr_t + Neighborhood * BedroomAbvGr,
    data = df_new)
summary(df.mod_i)  # Seems like the interaction term is not statistically significant.
```

```
##
## Call:
## lm(formula = SalePrice ~ LotArea + Neighborhood + GrLivArea +
##     BedroomAbvGr + KitchenAbvGr_t + Neighborhood * BedroomAbvGr,
##     data = df_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -122770  -18066    -128   16697  171654
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.078e+05  4.318e+04   4.811 1.66e-06
## LotArea                         6.751e-01  9.955e-02   6.782 1.75e-11
## NeighborhoodBlueste            -4.608e+04  1.256e+05  -0.367   0.7138
## NeighborhoodBrDale             -3.803e+04  5.767e+04  -0.659   0.5097
## NeighborhoodBrkSide            -4.719e+04  4.220e+04  -1.118   0.2636
## NeighborhoodClearCr             2.791e+04  4.222e+04   0.661   0.5086
## NeighborhoodCollgCr             2.646e+03  4.139e+04   0.064   0.9490
```

```
## NeighborhoodCrawfor                3.877e+04  4.416e+04   0.878   0.3802
## NeighborhoodEdwards               -3.116e+04  4.135e+04  -0.753   0.4513
## NeighborhoodGilbert               -1.534e+03  4.862e+04  -0.032   0.9748
## NeighborhoodIDOTRR                -6.820e+04  4.511e+04  -1.512   0.1308
## NeighborhoodMeadowV               -1.643e+04  4.338e+04  -0.379   0.7050
## NeighborhoodMitchel                5.869e+03  4.249e+04   0.138   0.8902
## NeighborhoodNAmes                 -7.188e+03  4.029e+04  -0.178   0.8584
## NeighborhoodNoRidge                1.186e+05  4.739e+04   2.502   0.0125
## NeighborhoodNPkVill                2.362e+04  6.997e+04   0.338   0.7357
## NeighborhoodNridgHt                5.481e+04  4.185e+04   1.310   0.1904
## NeighborhoodNWAmes                 2.847e+04  4.621e+04   0.616   0.5380
## NeighborhoodOldTown               -1.607e+04  4.061e+04  -0.396   0.6925
## NeighborhoodSawyer                -1.232e+04  4.234e+04  -0.291   0.7711
## NeighborhoodSawyerW                2.191e+04  4.248e+04   0.516   0.6061
## NeighborhoodSomerst                1.682e+04  4.235e+04   0.397   0.6913
## NeighborhoodStoneBr                6.498e+04  4.257e+04   1.526   0.1271
## NeighborhoodSWISU                 -2.687e+04  4.389e+04  -0.612   0.5405
## NeighborhoodTimber                 6.309e+04  4.614e+04   1.367   0.1718
## NeighborhoodVeenker                1.153e+05  4.575e+04   2.520   0.0119
## GrLivArea                          9.266e+01  2.688e+00  34.474  < 2e-16
## BedroomAbvGr                      -1.392e+03  2.105e+04  -0.066   0.9473
## KitchenAbvGr_t                    -3.660e+05  4.518e+04  -8.101 1.18e-15
## NeighborhoodBlueste:BedroomAbvGr -2.346e+03  5.133e+04  -0.046   0.9635
## NeighborhoodBrDale:BedroomAbvGr  -9.579e+03  2.679e+04  -0.358   0.7208
## NeighborhoodBrkSide:BedroomAbvGr -1.181e+03  2.181e+04  -0.054   0.9568
## NeighborhoodClearCr:BedroomAbvGr -2.097e+04  2.162e+04  -0.970   0.3323
## NeighborhoodCollgCr:BedroomAbvGr -2.565e+03  2.156e+04  -0.119   0.9053
## NeighborhoodCrawfor:BedroomAbvGr -2.097e+04  2.208e+04  -0.950   0.3425
## NeighborhoodEdwards:BedroomAbvGr -8.068e+03  2.151e+04  -0.375   0.7076
## NeighborhoodGilbert:BedroomAbvGr -7.695e+03  2.298e+04  -0.335   0.7378
## NeighborhoodIDOTRR:BedroomAbvGr  -5.895e+02  2.273e+04  -0.026   0.9793
## NeighborhoodMeadowV:BedroomAbvGr -1.830e+04  2.214e+04  -0.826   0.4087
## NeighborhoodMitchel:BedroomAbvGr -1.162e+04  2.181e+04  -0.533   0.5943
## NeighborhoodNAmes:BedroomAbvGr   -1.088e+04  2.127e+04  -0.512   0.6090
## NeighborhoodNoRidge:BedroomAbvGr -2.887e+04  2.237e+04  -1.291   0.1970
## NeighborhoodNPkVill:BedroomAbvGr -2.289e+04  3.063e+04  -0.747   0.4551
## NeighborhoodNridgHt:BedroomAbvGr  3.369e+03  2.172e+04   0.155   0.8767
## NeighborhoodNWAmes:BedroomAbvGr  -1.974e+04  2.232e+04  -0.885   0.3766
## NeighborhoodOldTown:BedroomAbvGr -1.846e+04  2.136e+04  -0.864   0.3876
## NeighborhoodSawyer:BedroomAbvGr  -9.144e+03  2.170e+04  -0.421   0.6736
## NeighborhoodSawyerW:BedroomAbvGr -1.569e+04  2.174e+04  -0.722   0.4706
## NeighborhoodSomerst:BedroomAbvGr -1.406e+03  2.186e+04  -0.064   0.9487
## NeighborhoodStoneBr:BedroomAbvGr -7.813e+02  2.212e+04  -0.035   0.9718
## NeighborhoodSWISU:BedroomAbvGr   -1.545e+04  2.165e+04  -0.714   0.4756
## NeighborhoodTimber:BedroomAbvGr  -1.974e+04  2.257e+04  -0.874   0.3821
## NeighborhoodVeenker:BedroomAbvGr -4.101e+04  2.319e+04  -1.768   0.0772
##
## (Intercept)                       ***
## LotArea                           ***
## NeighborhoodBlueste
## NeighborhoodBrDale
## NeighborhoodBrkSide
## NeighborhoodClearCr
## NeighborhoodCollgCr
```

```
## NeighborhoodCrawfor
## NeighborhoodEdwards
## NeighborhoodGilbert
## NeighborhoodIDOTRR
## NeighborhoodMeadowV
## NeighborhoodMitchel
## NeighborhoodNAmes
## NeighborhoodNoRidge              *
## NeighborhoodNPkVill
## NeighborhoodNridgHt
## NeighborhoodNWAmes
## NeighborhoodOldTown
## NeighborhoodSawyer
## NeighborhoodSawyerW
## NeighborhoodSomerst
## NeighborhoodStoneBr
## NeighborhoodSWISU
## NeighborhoodTimber
## NeighborhoodVeenker              *
## GrLivArea                        ***
## BedroomAbvGr
## KitchenAbvGr_t                   ***
## NeighborhoodBlueste:BedroomAbvGr
## NeighborhoodBrDale:BedroomAbvGr
## NeighborhoodBrkSide:BedroomAbvGr
## NeighborhoodClearCr:BedroomAbvGr
## NeighborhoodCollgCr:BedroomAbvGr
## NeighborhoodCrawfor:BedroomAbvGr
## NeighborhoodEdwards:BedroomAbvGr
## NeighborhoodGilbert:BedroomAbvGr
## NeighborhoodIDOTRR:BedroomAbvGr
## NeighborhoodMeadowV:BedroomAbvGr
## NeighborhoodMitchel:BedroomAbvGr
## NeighborhoodNAmes:BedroomAbvGr
## NeighborhoodNoRidge:BedroomAbvGr
## NeighborhoodNPkVill:BedroomAbvGr
## NeighborhoodNridgHt:BedroomAbvGr
## NeighborhoodNWAmes:BedroomAbvGr
## NeighborhoodOldTown:BedroomAbvGr
## NeighborhoodSawyer:BedroomAbvGr
## NeighborhoodSawyerW:BedroomAbvGr
## NeighborhoodSomerst:BedroomAbvGr
## NeighborhoodStoneBr:BedroomAbvGr
## NeighborhoodSWISU:BedroomAbvGr
## NeighborhoodTimber:BedroomAbvGr
## NeighborhoodVeenker:BedroomAbvGr .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33090 on 1397 degrees of freedom
## Multiple R-squared:  0.8009, Adjusted R-squared:  0.7935
## F-statistic: 108.1 on 52 and 1397 DF,  p-value: < 2.2e-16
```

```r
# Comparison of performance with other two models through AIC
# and BIC
AIC(df.base, df.mod_Cp, df.mod_i)
```

```
## Warning in AIC.default(df.base, df.mod_Cp, df.mod_i): models are not all
## fitted to the same number of observations
```

```
##           df      AIC
## df.base   55 34984.39
## df.mod_Cp 30 34355.90
## df.mod_i  54 34349.05
```

```r
BIC(df.base, df.mod_Cp, df.mod_i)
```

```
## Warning in BIC.default(df.base, df.mod_Cp, df.mod_i): models are not all
## fitted to the same number of observations
```

```
##           df      BIC
## df.base   55 35275.13
## df.mod_Cp 30 34514.27
## df.mod_i  54 34634.13
```

```r
# Comment : Based on AIC, it seems like df.mod_i is better
# compared to other models, and based on BIC, it seems that
# df.mode_Cp is better. Considering the fact that the
# coeffficients of the interaction term are not statistically
# significant, I would suggest df.mod_Cp performs the best.
```

(i) Lastly, choose your favorite model from all the ones estimated and perform a five-fold cross validation test on it. Then, divide train.csv dataset to evaluate how well your model predicts hoe prices for out of sample data.

```r
# I choose the model from (e) using Mallows Cp and perform a
# five-fold cross validation test.
library(DAAG)
```

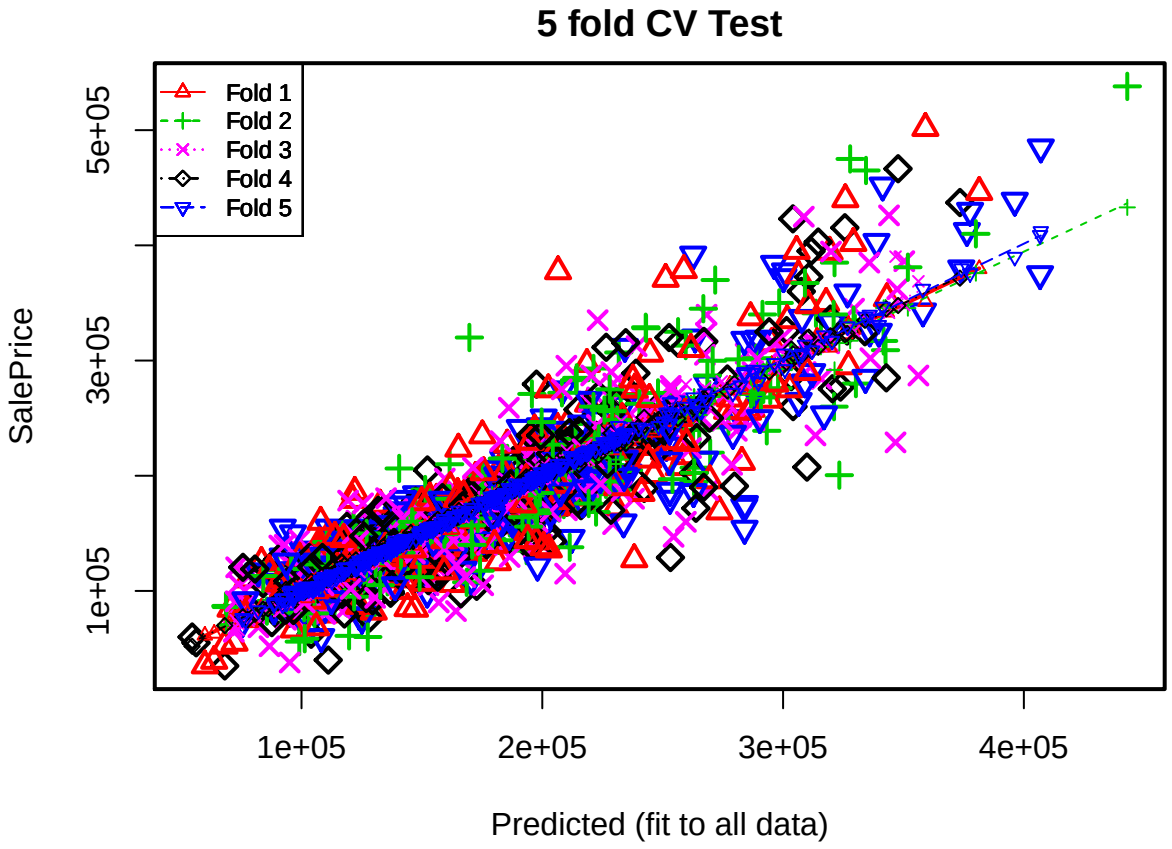
```
## Loading required package: lattice
```

```
##
## Attaching package: 'DAAG'
```

```
## The following object is masked from 'package:car':
##
##     vif
```

```r
set.seed(15)
cvResults <- CVlm(data = df_new, form.lm = formula(SalePrice ~
    LotArea + Neighborhood + GrLivArea + BedroomAbvGr + KitchenAbvGr_t),
    m = 5, legend.pos = "topleft", printit = FALSE, main = "5 fold CV Test")
```

```
## Warning in CVlm(data = df_new, form.lm = formula(SalePrice ~ LotArea + Neighborhood + :
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
# Comment : We check that the model's prediction accuracy
# (MSE) is not varying too much for any one particular sample
# (to see all 5 MSEs, please let printit=TRUE), and from the
# graph, we wee that the lines of best fit do not vary too
# much with respect to the slope and level.

# Divide my dataset into a train and a test dataset
ind <- sample(seq_len(nrow(df_new)), size = floor(0.8 * nrow(df_new)))
train <- df_new[ind, ]
test <- df_new[-ind, ]

# Fit a model in the train dataset
df.train <- lm(SalePrice ~ LotArea + Neighborhood + GrLivArea +
    BedroomAbvGr + KitchenAbvGr_t, data = train)
f <- test[, "SalePrice"] - predict(df.train, newdata = test)
mean(f^2)  # MSE
```

```
## [1] 1214705241
```

```r
mean(abs(f/test[, "SalePrice"]))  #Mean Absolute Percentage Error (MAPE)
```

```
## [1] 0.1332512
```

```r
# Seeing from the values such as MAPE, it seems like my model
# does predict with a mean error of 13.9%.
```

QUESTION 2

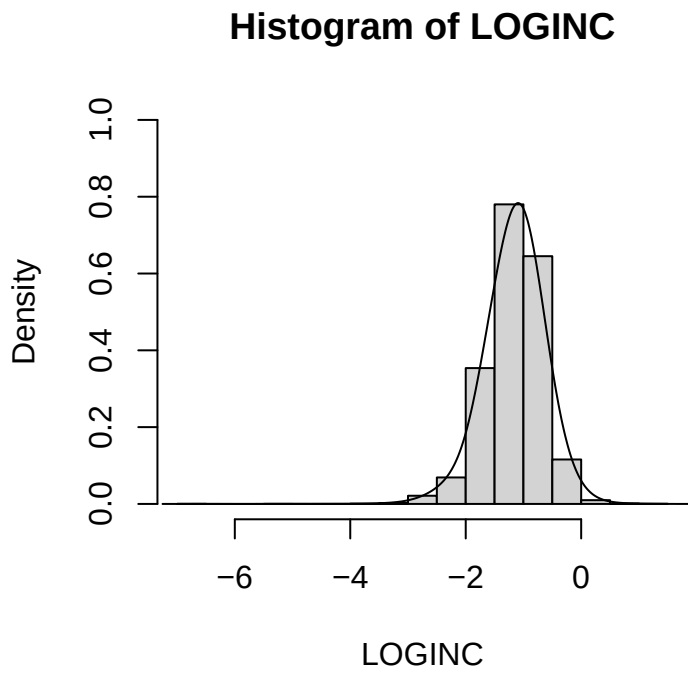
```
german_data <- na.omit(read.csv("german_healthcare_usage.csv"))
```

Variables Chosen: FEMALE, NEWHSAT, PUBLIC, AGE, WORKING, EDUC, LOGINC HHKIDS

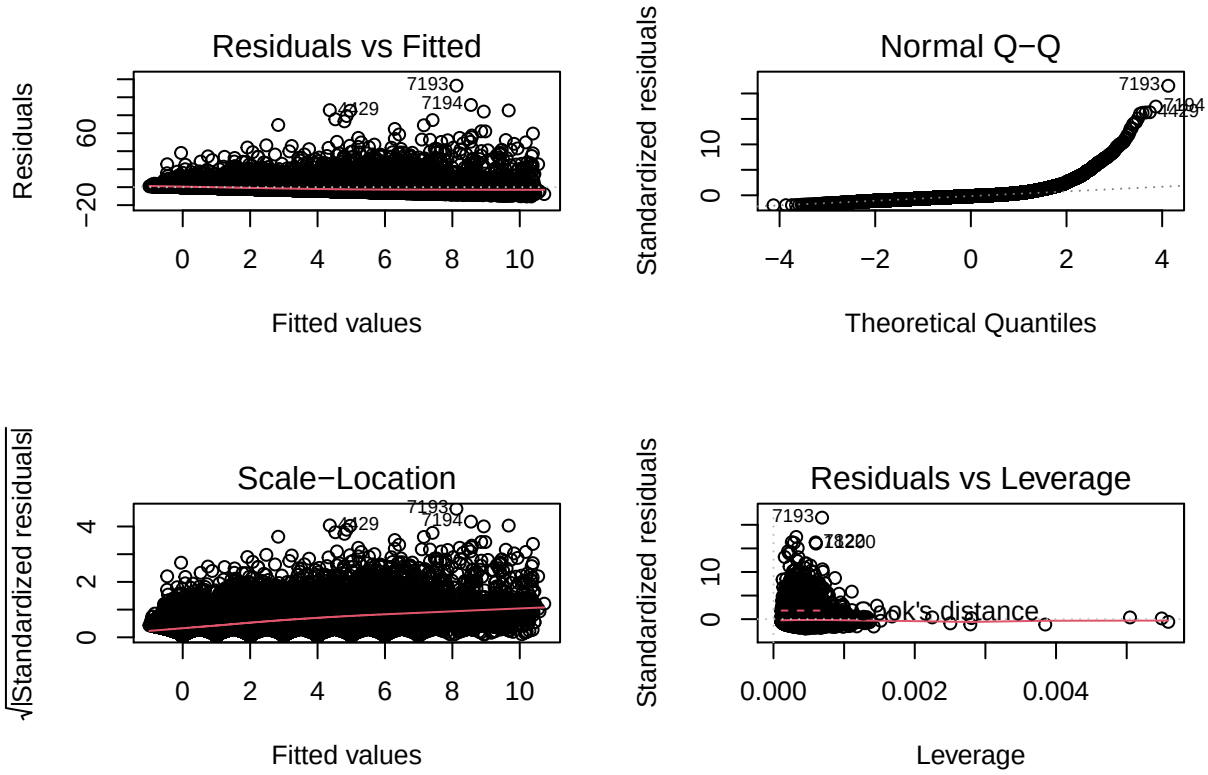
You could have chosen any variables that you deemed important in the prediction in the number of doctor visits.

Most of the chosen variables are categorical and thus do not need any transformation. Only Age and logincome are continuous.

```
with(german_data,
     {hist(LOGINC,freq = FALSE, ylim = c(0,1))
     lines(density(LOGINC, bw=0.25))}
)
```




```
mod <- lm(DOCVIS~ FEMALE + EDUC + WORKING + NEWHSAT + PUBLIC + AGE  + LOGINC + HHKIDS, german_data)
par(mfrow=c(2,2))
plot(mod)
```




```
summary(mod)
```

```
Call:
lm(formula = DOCVIS ~ FEMALE + EDUC + WORKING + NEWHSAT + PUBLIC +
    AGE + LOGINC + HHKIDS, data = german_data)

Residuals:
    Min      1Q  Median      3Q     Max
-10.340  -2.428  -0.909   0.812 112.880

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.764701   0.344643  22.530  < 2e-16 ***
FEMALE       0.685677   0.069782   9.826  < 2e-16 ***
EDUC        -0.002812   0.015287  -0.184 0.854031
WORKING     -0.427760   0.078792  -5.429 5.72e-08 ***
NEWHSAT     -0.856667   0.014413 -59.438  < 2e-16 ***
PUBLIC       0.334019   0.106896   3.125 0.001782 **
AGE          0.018238   0.003191   5.715 1.11e-08 ***
LOGINC      -0.257265   0.069800  -3.686 0.000228 ***
HHKIDS      -0.403406   0.069605  -5.796 6.88e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.248 on 27288 degrees of freedom
```

```
Multiple R-squared:  0.1497,    Adjusted R-squared:  0.1495
F-statistic: 600.6 on 8 and 27288 DF,  p-value: < 2.2e-16
```

The results indicate that the assumptions for linear regression have been violated. Even if we were to transform each variable, we would never be able to achieve normality. This is because we are estimating count data, which is highly non-normal with a gaussian link.

What happens when we transform?

```
library(car)
```

```
Warning: package 'car' was built under R version 4.0.3
```

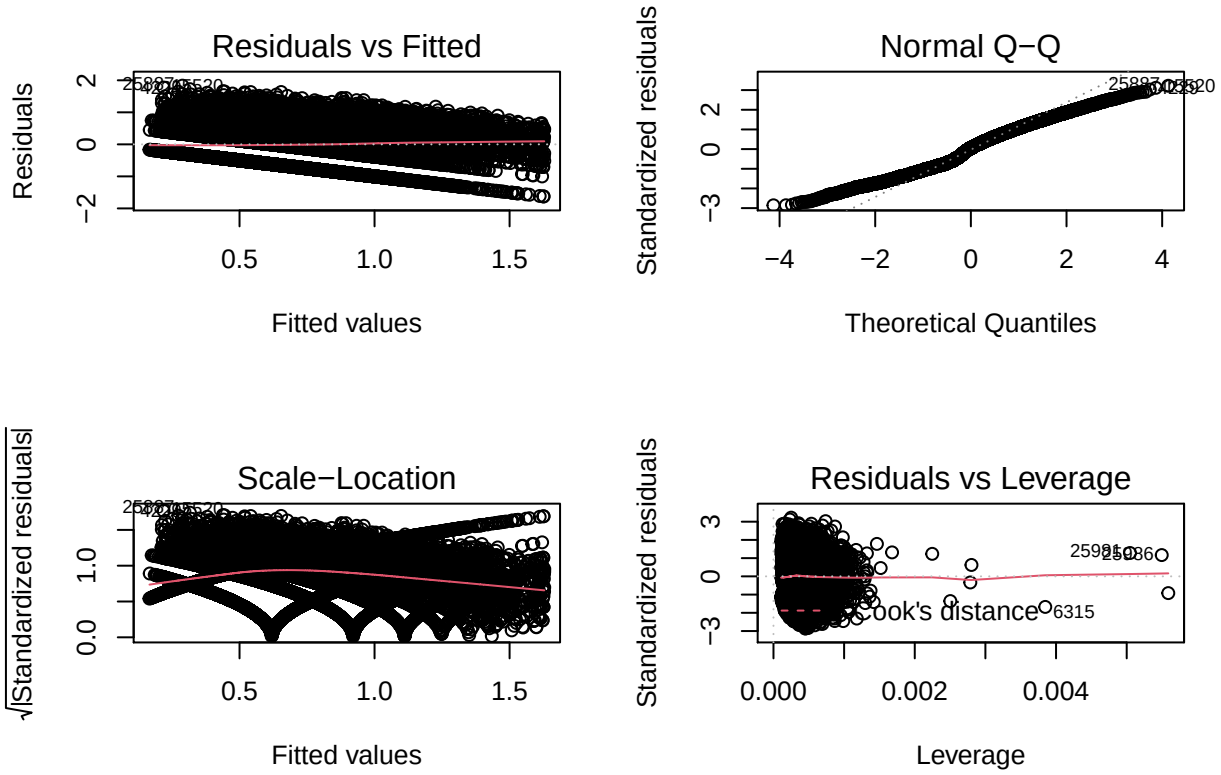
```
Loading required package: carData
```

```
lam = powerTransform(mod, family = 'yjPower')$lam
mod1 = lm(yjPower(DOCVIS, lam)~FEMALE + EDUC + WORKING + NEWHSAT + PUBLIC + AGE  +
            LOGINC + HHKIDS, german_data)
summary(mod1)
```

```

Call:
lm(formula = yjPower(DOCVIS, lam) ~ FEMALE + EDUC + WORKING +
    NEWHSAT + PUBLIC + AGE + LOGINC + HHKIDS, data = german_data)

Residuals:
     Min       1Q   Median       3Q      Max
-1.62623 -0.47814  0.03299  0.44580  1.83262

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.2088872  0.0373812  32.339  < 2e-16 ***
FEMALE       0.1520554  0.0075688  20.090  < 2e-16 ***
EDUC         0.0003137  0.0016580   0.189    0.850
WORKING     -0.0431985  0.0085461  -5.055 4.34e-07 ***
NEWHSAT     -0.1032476  0.0015632 -66.047  < 2e-16 ***
PUBLIC       0.0460617  0.0115943   3.973 7.12e-05 ***
AGE          0.0033176  0.0003461   9.586  < 2e-16 ***
LOGINC      -0.0025305  0.0075708  -0.334    0.738
HHKIDS      -0.0601891  0.0075496  -7.973 1.62e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5693 on 27288 degrees of freedom
Multiple R-squared:  0.1942,    Adjusted R-squared:  0.1939
F-statistic: 821.8 on 8 and 27288 DF,  p-value: < 2.2e-16
```

This does improve the R^2 abit. How do the plots look?

```
par(mfrow=c(2,2))
plot(mod1)
```



We see that the transformed data has an huge effect on the model. The qq-norm plot more or less shows that the data is quite normal, although the patterns in residuals vs fitted indicate that there is no independence. The data is not homogeneous and thus liner model assumptions are still violated.

How should we tackle this?

The best/correct way to approach this is by fitting a model which is used for count data for example a poison model or even negative binomial model. For this specific example, since the number of zeros in the response valriable is 37% of the total data, we would use a zero-inflated model. Please check here for models with zero inflated data.

PART B

PART I)

```
year_dummy <- +(german_data$YEAR>=1987)
summary(lm(DOCVIS~year_dummy *  FEMALE, german_data))
```

```
Call:
lm(formula = DOCVIS ~ year_dummy * FEMALE, data = german_data)

Residuals:
    Min      1Q  Median      3Q     Max
 -3.917  -2.643  -1.643   0.387 118.357

Coefficients:
```

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        2.64255    0.07335  36.026   <2e-16 ***
year_dummy        -0.02944    0.09620  -0.306    0.760
FEMALE             1.27462    0.10590  12.036   <2e-16 ***
year_dummy:FEMALE -0.18560    0.13899  -1.335    0.182
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.661 on 27293 degrees of freedom
Multiple R-squared:  0.01067,   Adjusted R-squared:  0.01056
F-statistic: 98.08 on 3 and 27293 DF,  p-value: < 2.2e-16
```

The number of doctor visits for females reduced by 0.186 after the policy implementation. We cannot reject that this effect is equal to zero since the p-value is greater than 0.05

PART II)

```
summary(lm(DOCVIS~year_dummy *  UNEMPLOY, german_data))
```

```
Call:
lm(formula = DOCVIS ~ year_dummy * UNEMPLOY, data = german_data)

Residuals:
    Min      1Q  Median      3Q     Max
 -4.308  -2.722  -1.722   0.324 116.692

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.67609    0.06569  40.741   <2e-16 ***
year_dummy           0.04550    0.08480   0.537   0.5916
UNEMPLOY             1.63193    0.11037  14.786   <2e-16 ***
year_dummy:UNEMPLOY -0.25961    0.14752  -1.760   0.0784 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.648 on 27293 degrees of freedom
Multiple R-squared:  0.01509,   Adjusted R-squared:  0.01498
F-statistic: 139.4 on 3 and 27293 DF,  p-value: < 2.2e-16
```

The number of doctor visits for unemployed reduced by 0.26 after the policy implementation. We cannot reject that this effect is equal to zero since the p-value is greater than 0.05

PART C

$$H_0 : \beta_{Female} \leq 0$$
$$H_a : \beta_{Female} > 0$$

```
summary(mod)
```

```
Call:
lm(formula = DOCVIS ~ FEMALE + EDUC + WORKING + NEWHSAT + PUBLIC +
    AGE + LOGINC + HHKIDS, data = german_data)

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-10.340  -2.428  -0.909   0.812 112.880

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.764701   0.344643  22.530  < 2e-16 ***
FEMALE       0.685677   0.069782   9.826  < 2e-16 ***
EDUC        -0.002812   0.015287  -0.184 0.854031
WORKING     -0.427760   0.078792  -5.429 5.72e-08 ***
NEWHSAT     -0.856667   0.014413 -59.438  < 2e-16 ***
PUBLIC       0.334019   0.106896   3.125 0.001782 **
AGE          0.018238   0.003191   5.715 1.11e-08 ***
LOGINC      -0.257265   0.069800  -3.686 0.000228 ***
HHKIDS      -0.403406   0.069605  -5.796 6.88e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.248 on 27288 degrees of freedom
Multiple R-squared:  0.1497,    Adjusted R-squared:  0.1495
F-statistic: 600.6 on 8 and 27288 DF,  p-value: < 2.2e-16
```

We can see that the Female coefficient is significant hence it is different from zero. nd since the t-value is positive, it definately shows that the coefficient is greater than zero. The P value for the one sided t-test will be the p-value given above times 2. Which is practically zero.

Also we can do a linear hypothesis testing:

```
car::lht(mod, 'FEMALE = 0')
```

```
Linear hypothesis test

Hypothesis:
FEMALE = 0

Model 1: restricted model
Model 2: DOCVIS ~ FEMALE + EDUC + WORKING + NEWHSAT + PUBLIC + AGE + LOGINC +
    HHKIDS

  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1  27289 754340
2  27288 751681  1    2659.6 96.549 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We note that the p-value will be 0. Ans since $\beta_{Female} > 0.68$ which is greater than 0, we reject the null hypothesis and conclude that the number of doctor visits a patient has over a 3 month period is greater for women than for men.

PART D)

Would like to test whether the effect of working is half the effect of health satisfaction.

$$H_0 : \beta_{working} = \frac{1}{2}\beta_{newhsat} \implies 2\beta_{working} - \beta_{newhsat} = 0$$
$$H_1 : 2\beta_{working} + \beta_{newhsat} \neq 0$$

```r
car::lht(mod, "WORKING - 0.5NEWHSAT = 0")
```

```
Linear hypothesis test

Hypothesis:
WORKING - 0.5 NEWHSAT = 0

Model 1: restricted model
Model 2: DOCVIS ~ FEMALE + EDUC + WORKING + NEWHSAT + PUBLIC + AGE + LOGINC +
    HHKIDS

  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1  27289 751681
2  27288 751681  1 0.0014313 1e-04 0.9942
```

From the above analysis, we do not reject the null hypothesis and thus conclude that there is not enough evidence to reject the notion that the effect of working is half the effect of health satisfaction. The probability of the effect of working being half the effect of health satisfaction is 0.9942.