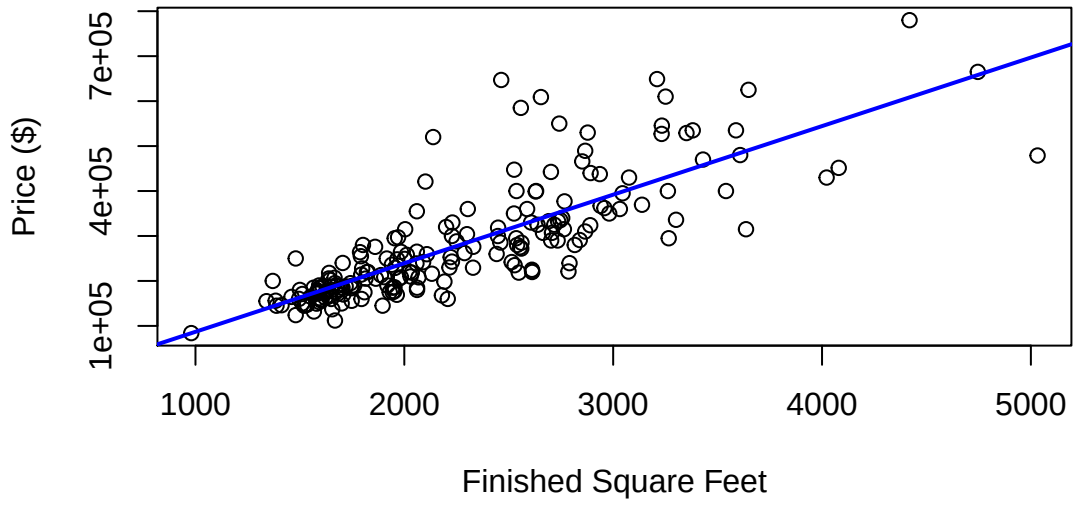
4. Do problem 3.31 on page 153.

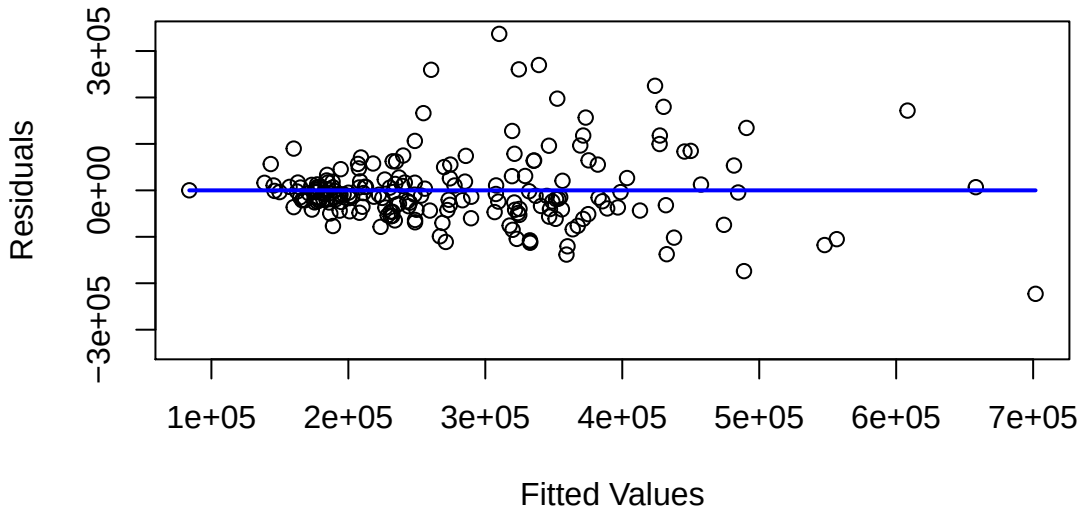
Problem 3.31: Refer to the real estate data set in Appendix C.7. Obtain a random sample of 200 cases from the 522 cases in this dataset. Using the random sample, build a regression model to predict sales price (Y) as a function of finished square feet (X). The analysis should include an assessment of the degree to which the key regression assumptions are satisfied. If the regression assumptions are not met, include and justify appropriate remedial measures. Use the final model to predict sales price for two houses that are about to come on the market: the first has X = 1100 finished square feet and the second has X = 4900 finished square feet. Assess the strengths and weaknesses of the final model.

```
df <- read.csv("data/Case 07.csv")
df <- df[sample(1:dim(df)[1], 200),]

names(df)[2] = "PriceUSD"
names(df)[3] = "FinishedSqFt"

write.csv(df, file="data/problem4sample.csv")

result1 <- lm(PriceUSD ~ FinishedSqFt, data=df)

plot(df$FinishedSqFt, df$PriceUSD, main="",
     xlab="Finished Square Feet", ylab="Price ($)")
abline(result1, col="blue", lwd="2")
```

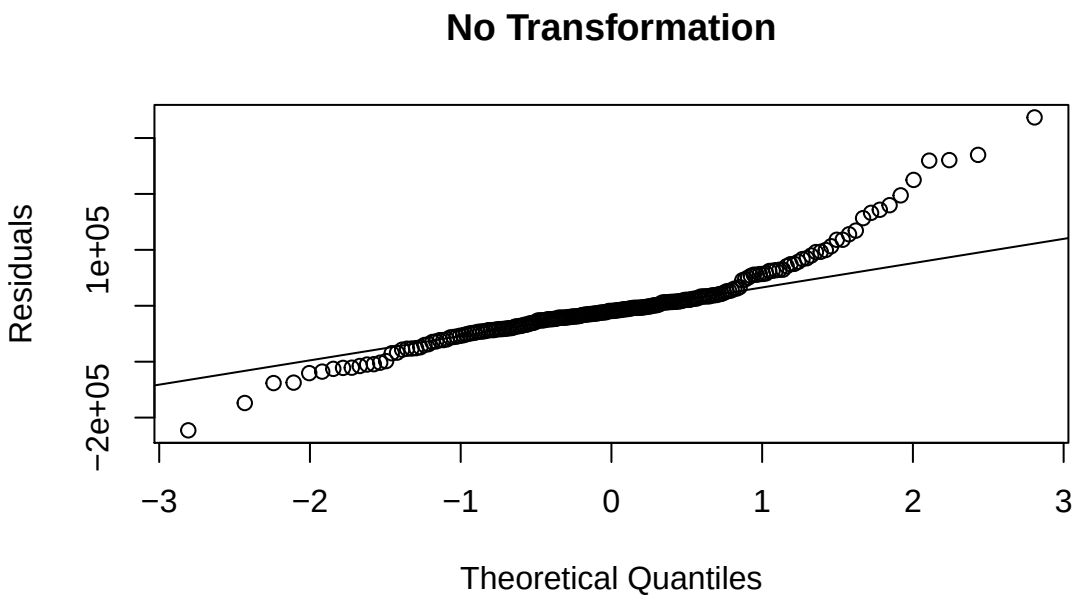



```
with(result1, {
     plot(x=fitted.values, y=residuals,
```

```
          ylim=c(-max(residuals), max(residuals)),
          xlab="Fitted Values", ylab="Residuals", main="")

     points(c(min(fitted.values), max(fitted.values)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```




```
qqnorm(result1$residuals, ylab="Residuals", main="No Transformation")
qqline(result1$residuals)
```



The raw explanatory variable, finished square feet, appears to depart significantly from normality and, in fact, the plot of residuals vs fitted values shows a characteristic known as heteroskedasticity.

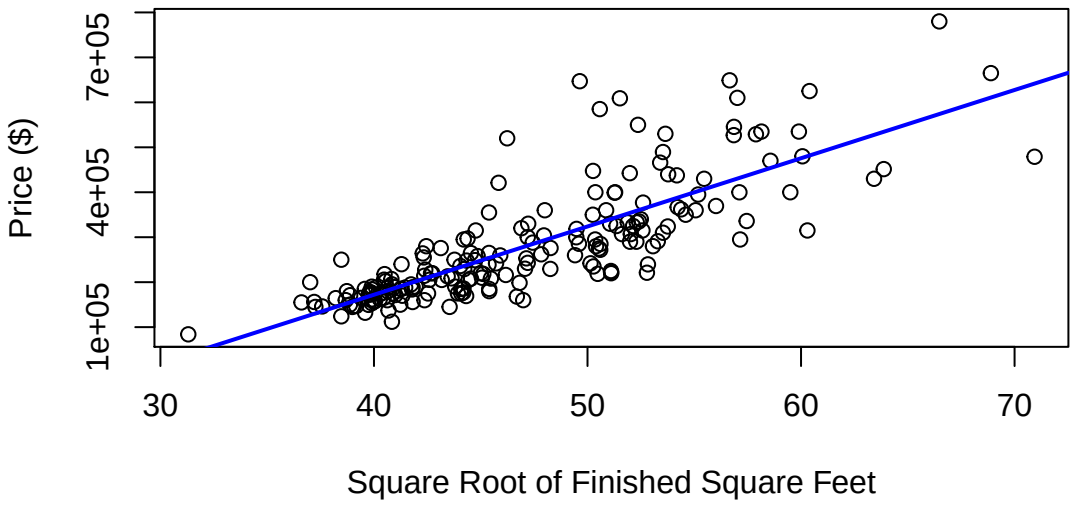
Methods of transformation should be applied to develop a model that conforms to the assumption of normality.

Square Root and Logarithm Transformations to FinishedSqFt Variable

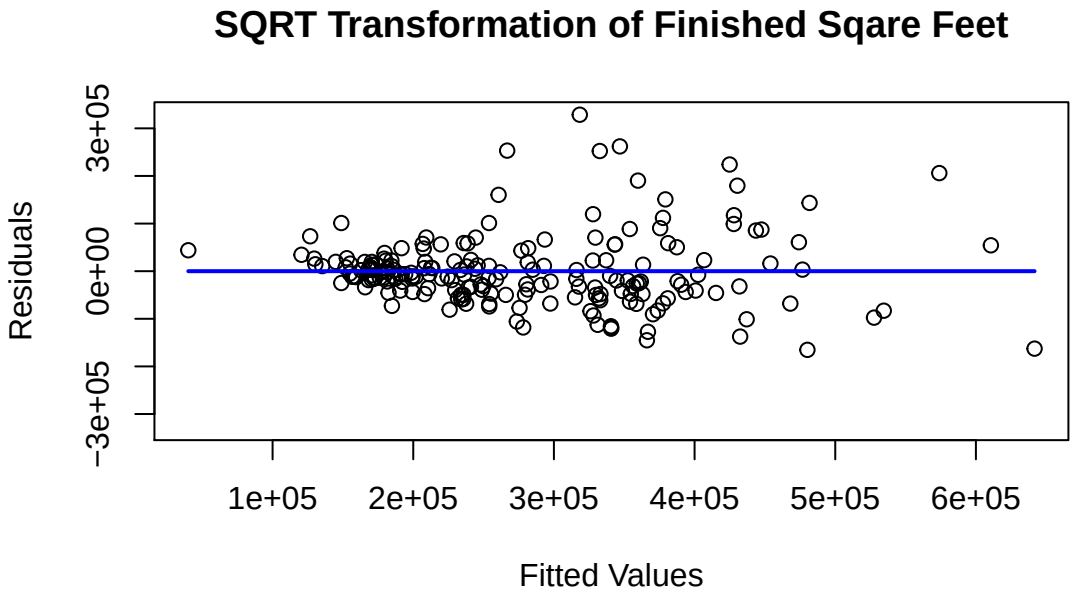
```
df$SqRtFinishedSqFt <- sqrt(df$FinishedSqFt)
df$LogFinishedSqFt <- log(df$FinishedSqFt)

result2 <- lm(PriceUSD ~ SqRtFinishedSqFt, data=df)
result3 <- lm(PriceUSD ~ LogFinishedSqFt, data=df)
```

Assessing Square Root Transformation of FinishedSqRt

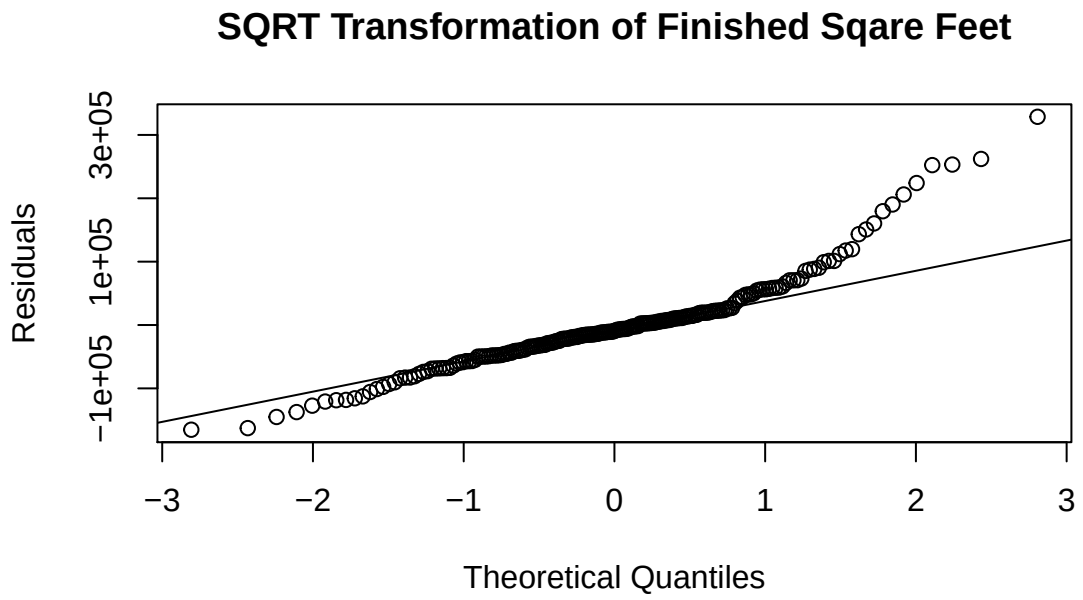
```
plot(df$SqRtFinishedSqFt, df$PriceUSD, main="",
     xlab="Square Root of Finished Square Feet", ylab="Price ($)")
abline(result2, col="blue", lwd="2")
```




```
with(result2, {
     plot(x=fitted.values, y=residuals,
          ylim=c(-max(residuals), max(residuals)),
          xlab="Fitted Values", ylab="Residuals",
          main="SQRT Transformation of Finished Sqare Feet")

     points(c(min(fitted.values), max(fitted.values)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



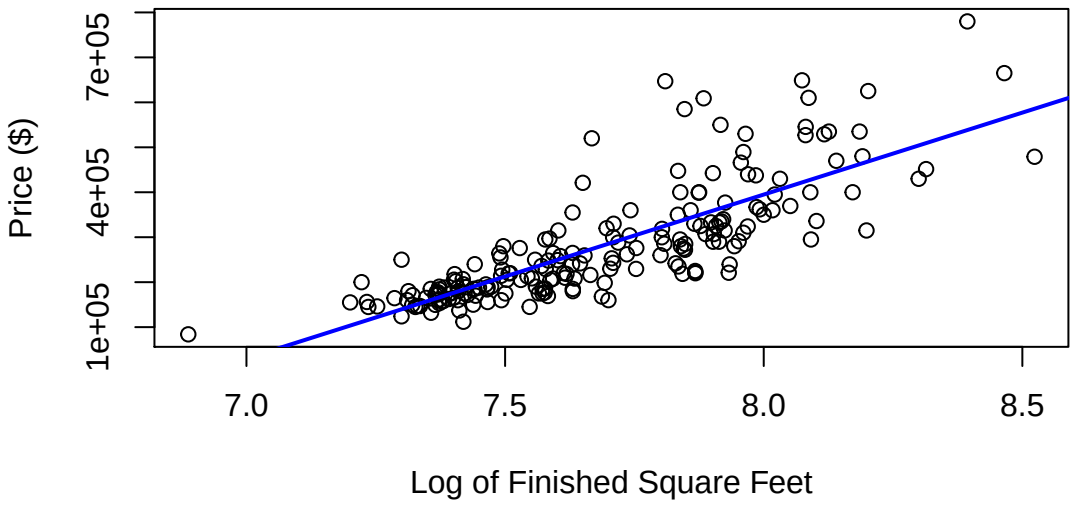
```r
qqnorm(result2$residuals, ylab="Residuals",
       main="SQRT Transformation of Finished Sqare Feet")
qqline(result2$residuals)
```



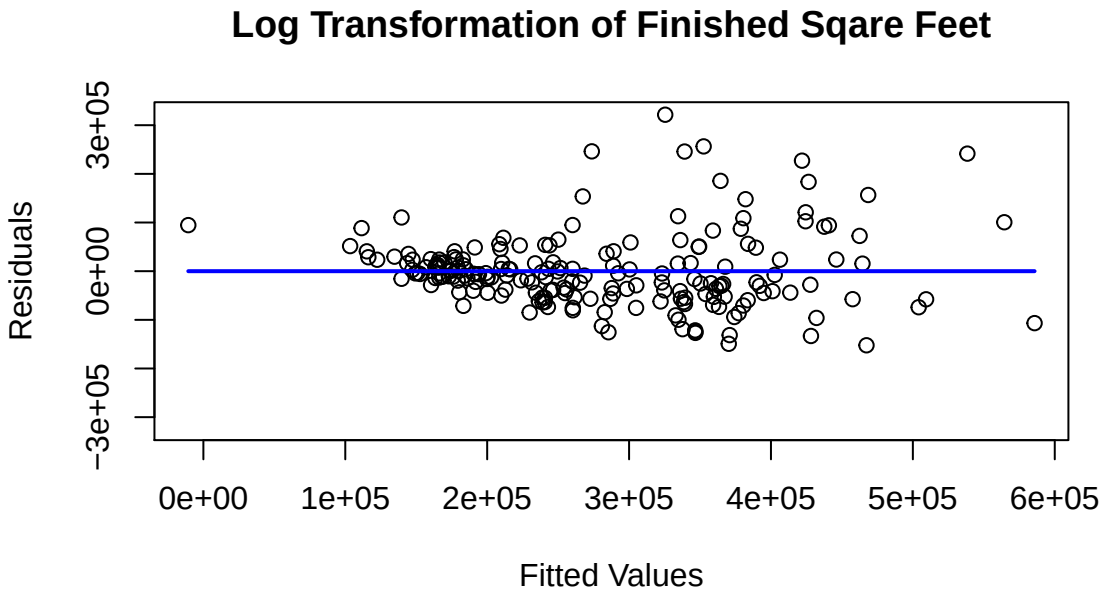
The square root tranformation of the variable representing finished square feet still exhibits significant departures from normality and heteroskedasticity.

Assessing Log Transformation of FinishedSqRt

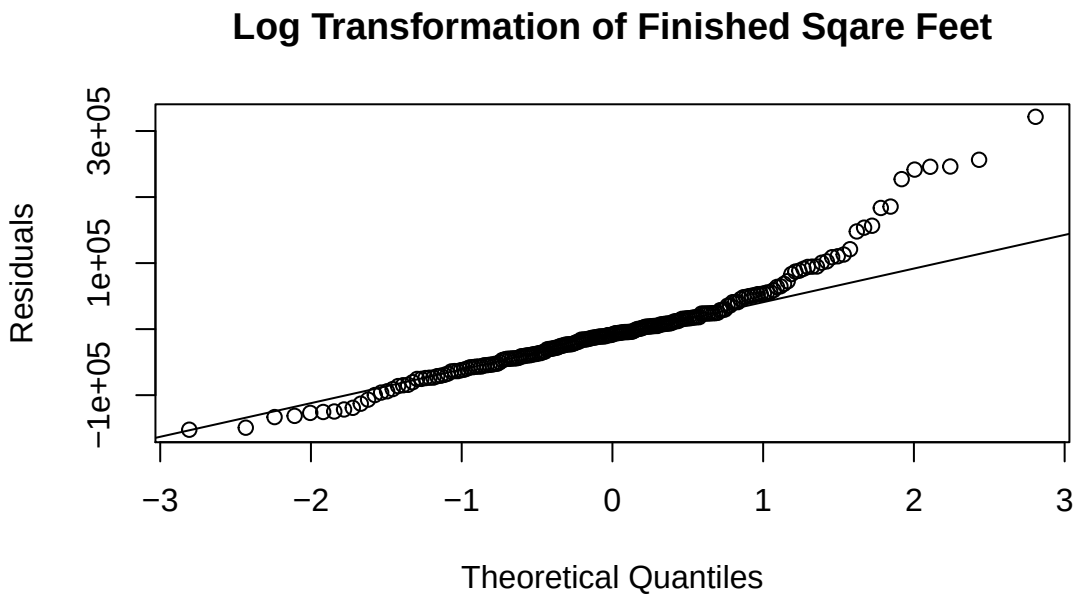
```r
plot(df$LogFinishedSqFt, df$PriceUSD, main="",
     xlab="Log of Finished Square Feet", ylab="Price ($)")
abline(result3, col="blue", lwd="2")
```




```
with(result3, {
     plot(x=fitted.values, y=residuals,
          ylim=c(-max(residuals), max(residuals)),
          xlab="Fitted Values", ylab="Residuals",
          main="Log Transformation of Finished Sqare Feet")

     points(c(min(fitted.values), max(fitted.values)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



```
qqnorm(result3$residuals, ylab="Residuals",
       main="Log Transformation of Finished Sqare Feet")
qqline(result3$residuals)
```



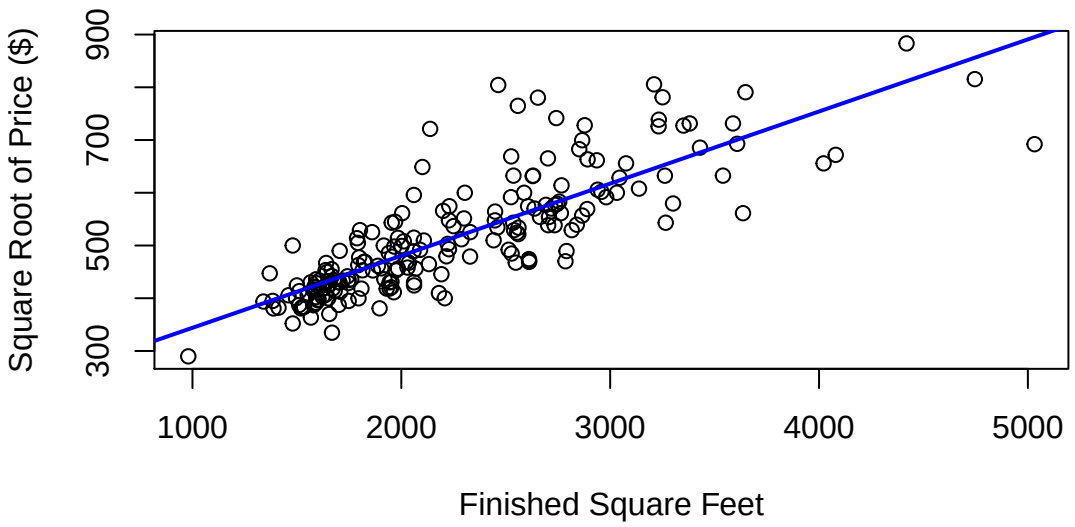
The log tranformation of the variable representing finished square feet still exhibits significant departures from normality and heteroskedasticity.

Square Root and Log Transformation of PriceUSD Outcome Variable

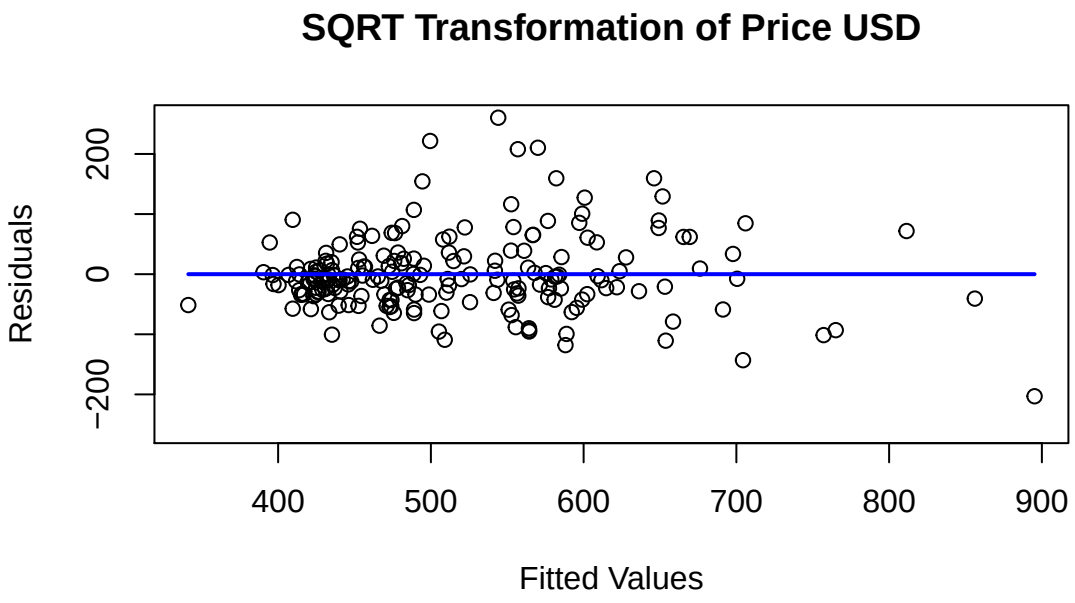
```
df$SqRtPriceUSD <- sqrt(df$PriceUSD)
df$LogPriceUSD <- log10(df$PriceUSD)

result4 <- lm(SqRtPriceUSD ~ FinishedSqFt, data=df)
result5 <- lm(LogPriceUSD ~ FinishedSqFt, data=df)
```

Assessing Square Root Transformation of PriceUSD

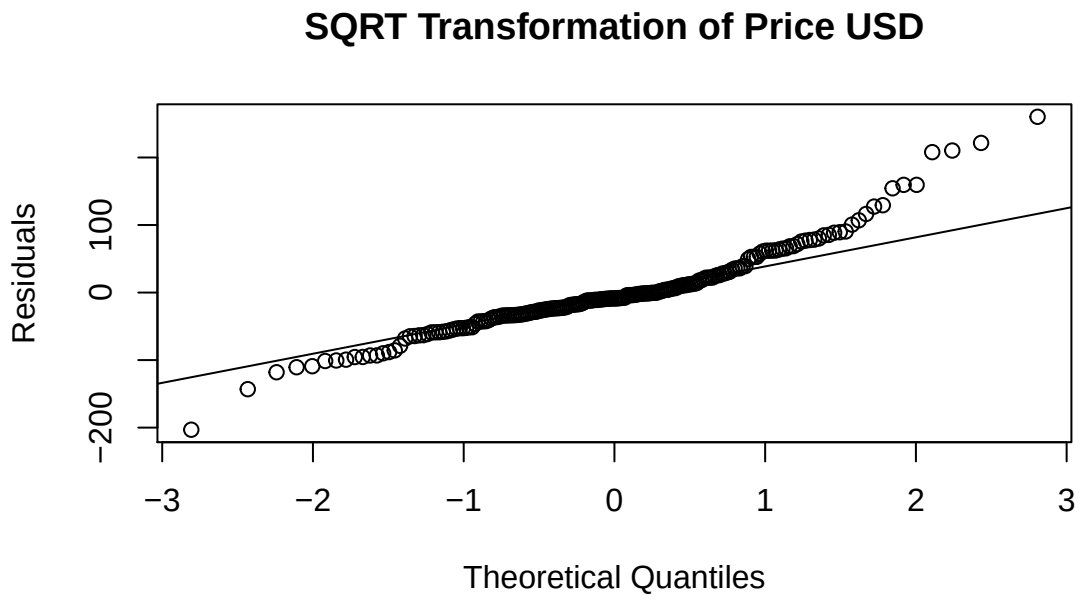
```
plot(df$FinishedSqFt, df$SqRtPriceUSD, main="",
     xlab="Finished Square Feet", ylab="Square Root of Price ($)")
abline(result4, col="blue", lwd="2")
```




```
with(result4, {
     plot(x=fitted.values, y=residuals,
          ylim=c(-max(residuals), max(residuals)),
          xlab="Fitted Values", ylab="Residuals",
          main="SQRT Transformation of Price USD")

     points(c(min(fitted.values), max(fitted.values)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



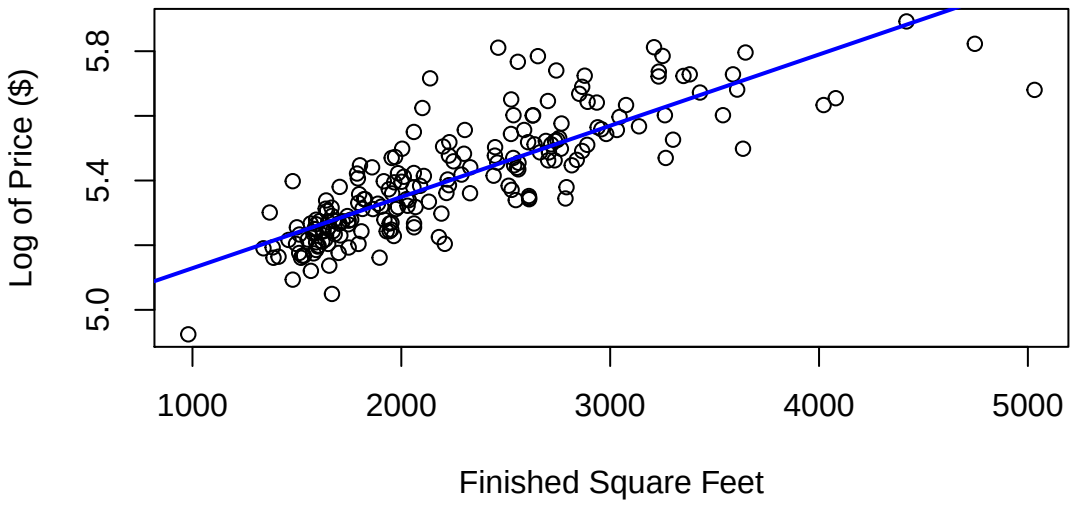
```
qqnorm(result4$residuals, ylab="Residuals",
       main="SQRT Transformation of Price USD")
qqline(result4$residuals)
```



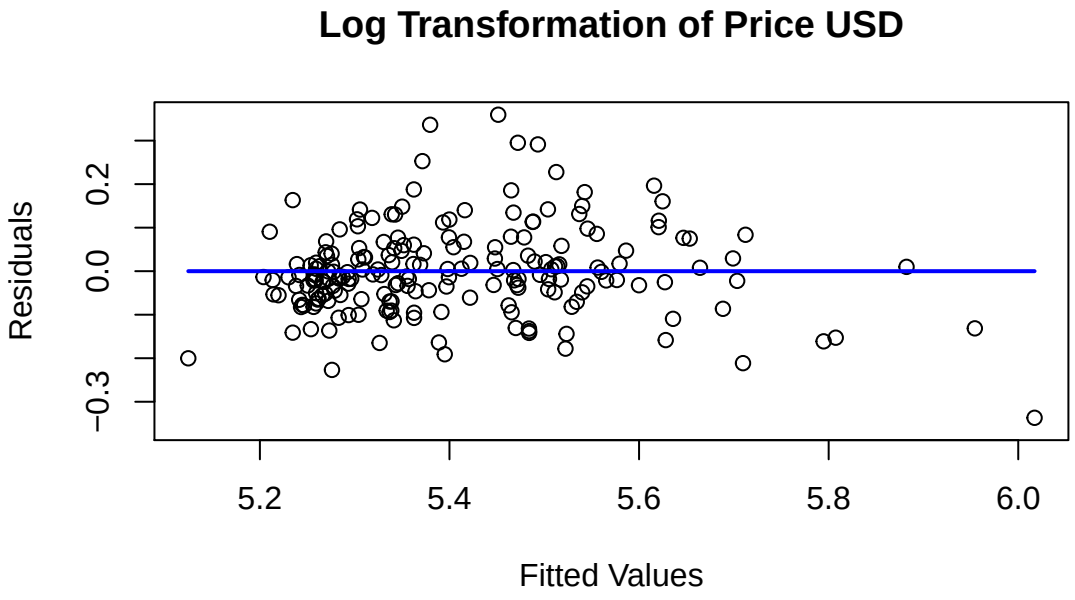
The square root tranformation of the variable representing sale price in USD to be predicted still exhibits significant departures from normality and heteroskedasticity.

Assessing Log Transformation of PriceUSD

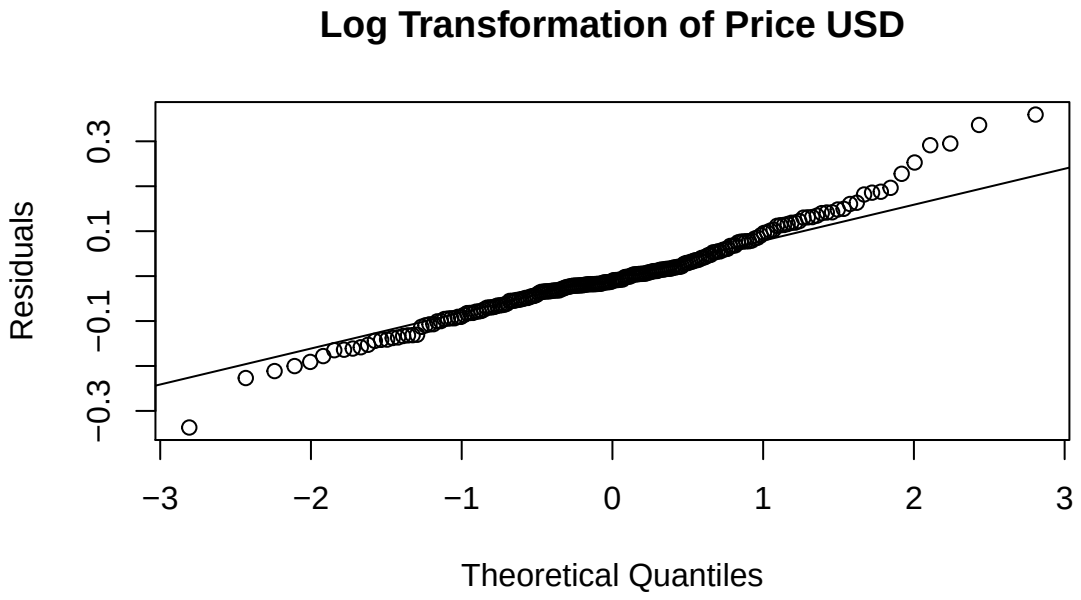
```
plot(df$FinishedSqFt, df$LogPriceUSD, main="",
     xlab="Finished Square Feet", ylab="Log of Price ($)")
abline(result5, col="blue", lwd="2")
```




```
with(result5, {
     plot(x=fitted.values, y=residuals,
          ylim=c(-max(residuals), max(residuals)),
          xlab="Fitted Values", ylab="Residuals",
          main="Log Transformation of Price USD")

     points(c(min(fitted.values), max(fitted.values)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



```
qqnorm(result5$residuals, ylab="Residuals",
       main="Log Transformation of Price USD")
qqline(result5$residuals)
```



The log tranformation of the sale price in USD variable in the training dataset appears to be an appropriate method for achieving the approximation of linearity for the use of a linear model.

Formal Testing of Normality

Coefficient of correlation test for ordered residuals and expected values. Test the reasonableness of the normality assumption using Table B.6 and $\alpha = 0.05$ Since the tablehas a max n value of 100 this value of 0.987 will be used

```
df$Residuals <- result5$residuals
df$Rank <- rank(df$Residuals)
df$Prob <- (df$Rank - 0.375) / (length(df$Rank) + 0.25)
df$Z <- qnorm(df$Prob, mean=0, sd=1)
result_smry <- summary(result5)
df$EV <- result_smry$sigma * df$Z
n <- dim(df)[1]
cor_coef <- round(with(df,cor(Residuals, EV)), 4)
txt = paste("Training Dataset Size: ", n,
            "\nCorr coef (ordered residuals vs expected values): ", cor_coef,
            sep="")
cat(txt)
```

```
## Training Dataset Size: 200
## Corr coef (ordered residuals vs expected values): 0.984
```

If r > r critical (0.987, value taken from table B.6) conclude the residuals are normally distributed.

The value of the correlation coefficient is larger than the critical value which indicates that the residuals of the linear model are normally distributed.

Shapiro-Wilk test for normality of Residuals

```
shapiro.test(df$Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Residuals
## W = 0.9713, p-value = 0.0004105
```

Condition of Kolmogorov-Smirnov test: the null hypothesis is that the population is normally distributed.

H_o: If p-value $>$ 0.05 the data is normally distributed

H_o: If p-value $<=$ 0.05 the data is not normally distributed

The Shapiro-Wilk test for normality's null hypothesis is that the sample being tested is from a normally distributed population. Thus at $\alpha = 0.05$ a p-value < 0.05 suggests that the null hypothesis should be accepted and the data is normally distributed.

Review Summary of Model for Log Transformed Price in USD

```
smry <- summary(result5)
b0 <- coef(smry)[1,1]
b1 <- coef(smry)[2,1]
pctRsq <- round(smry$r.squared*100)
smry
```

```
##
## Call:
## lm(formula = LogPriceUSD ~ FinishedSqFt, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33688 -0.05512 -0.01076  0.05282  0.35948
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.909e+00  2.497e-02  196.61   <2e-16 ***
## FinishedSqFt 2.203e-04  1.062e-05   20.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1013 on 198 degrees of freedom
## Multiple R-squared:  0.6849, Adjusted R-squared:  0.6833
## F-statistic: 430.3 on 1 and 198 DF,  p-value: < 2.2e-16
```

Lack of Fit Test

```r
#df <- read.csv("data/CH. 1, PR 20.csv")
#names(df) = c("ServiceTime", "NumCopiers")
#result <- lm(ServiceTime ~ factor(NumCopiers), data=df)
#F_crit <- qf(0.95, df1=1, df2=43)

result1 <- lm(LogPriceUSD ~ FinishedSqFt, data=df)
result2 <- lm(LogPriceUSD ~ factor(FinishedSqFt), data=df)
F_crit <- qf(0.95, df1=1, df2=198)

n <- dim(df)[1]
distinct_vals <- length(unique(df$FinishedSqFt))

# lack of fit degrees of freedom: number of distinct values - 2
lof_degf <- distinct_vals - 2
tot_degf <- n-1

# pure error degrees of freedom: n - lack of fit degrees of freedom
pe_degf <- n - distinct_vals

# error degrees of freedom
err_degf <- n - 2

SSR <- anova(result1)$"Sum Sq"[1]
SSE <- anova(result1)$"Sum Sq"[2]

SSPE <- anova(result2)$"Sum Sq"[2]
SSLF <- SSE - SSPE
SST <- SSR + SSE


MSR <- anova(result1)$"Mean Sq"[1]
MSE <- anova(result1)$"Mean Sq"[2]
MSPE <- anova(result2)$"Mean Sq"[2]
MSLF <- SSLF / (distinct_vals - 2)
F_mod <- anova(result1)$"F value"[1]

F_lof <- MSLF / MSPE
Source = c("Regression",
           "Residual Error",
           "Lack of Fit Error",
           "Pure Error",
           "Total")
DF <- c(1,err_degf,
        lof_degf,
        pe_degf,
        tot_degf)
SS <- c(SSR,
        SSE,
        SSLF,
        SSPE,
        SST)
```

```
MS <- c(as.character(MSR),
        as.character(MSE),
        as.character(MSLF),
        as.character(MSPE),"")
F_value <- c(as.character(F_mod),
             "",
             as.character(F_lof),
             "",
             "" )
result_df <- data.frame(Source, DF, SS, MS, F_value)

library(knitr)
kable(result_df)
```

Source	DF	SS	MS	F_value
Regression	1	4.4117463	4.41174634748127	430.281395675373
Residual Error	198	2.0301268	0.0102531654675809	
Lack of Fit Error	183	1.9033299	0.0104007098607636	1.23039837340485
Pure Error	15	0.1267969	0.00845312387075248	
Total	199	6.4418731		

```
txt = paste("F*: ", F_lof, "\nF: ", F_crit, "\n", sep="")
cat(txt)
```

```
## F*: 1.23039837340485
## F: 3.88885293289187
```

The F statistic can be used to assess the lack of fit for a linear model which provides a formal way to test the following:

$H_o : E\{Y\} = \beta_0 + \beta_1 X$ * Concludes that the regression function is linear

$H_a : E\{Y\} \neq \beta_0 + \beta_1 X$ * Concludes that there is a lack of linear fit

Let $\alpha = 0.05$. Since n = 200, F(0.95; 1,198) = 3.8888529. The decision rule is as follows:

- If $F* \leq 3.8888529$, conclude H_o
- If $F* > 3.8888529$, conclude H_a

Conclusion: there is a linear association between number of log10 price in USD and Finished square feet.

The R^2 value of 0.6848546 means that the model explains 68 % of the log tranformation of the selling price in USD.

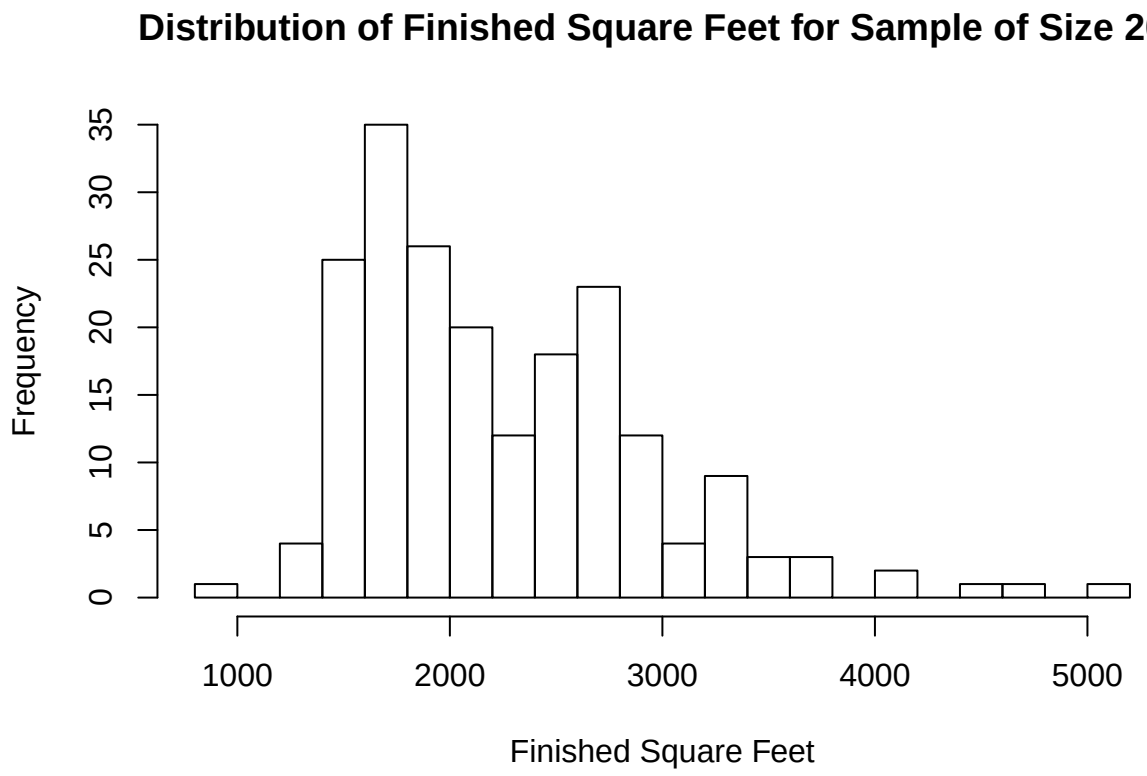
The final model is:

$log_{10}(PriceUSD) = 4.9085565 + 2.2032169 \times 10^{-4} FinishedSqFt$

Use the final model to predict sales price for houses with 1100 and 4900 finished square feet. Assess the strengths and weaknesses of the final model.

Assess the range of values in the model and Make Predictions with the Model

```
title = paste("Distribution of Finished Square Feet for Sample of Size ",
              length(df$FinishedSqFt), sep="")
hist(df$FinishedSqFt, main=title, xlab="Finished Square Feet", breaks=20)
```




```
summary(df$FinishedSqFt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     980    1701    2060    2252    2671    5032
```

```
new_data <- data.frame(FinishedSqFt=c(1100, 4900))

pred <- round(10^predict(result5, new_data))

pred_df <- data.frame(FinishedSqFt=c(1100,4900), PredictedPriceUSD=pred)

library(knitr)
kable(pred_df)
```

FinishedSqFt	PredictedPriceUSD
1100	141550
4900	973045

The values of the predictor variables are near the lower and upper bounds of the training data used to build the model. Also the bounds of the model appeared to be where the largest deviation from normality of the residuals which warrants a word of caution. These two factors reduce the robustness of the strength of the models predictions.