Problem Set 1 Solution

Denis Mokanov

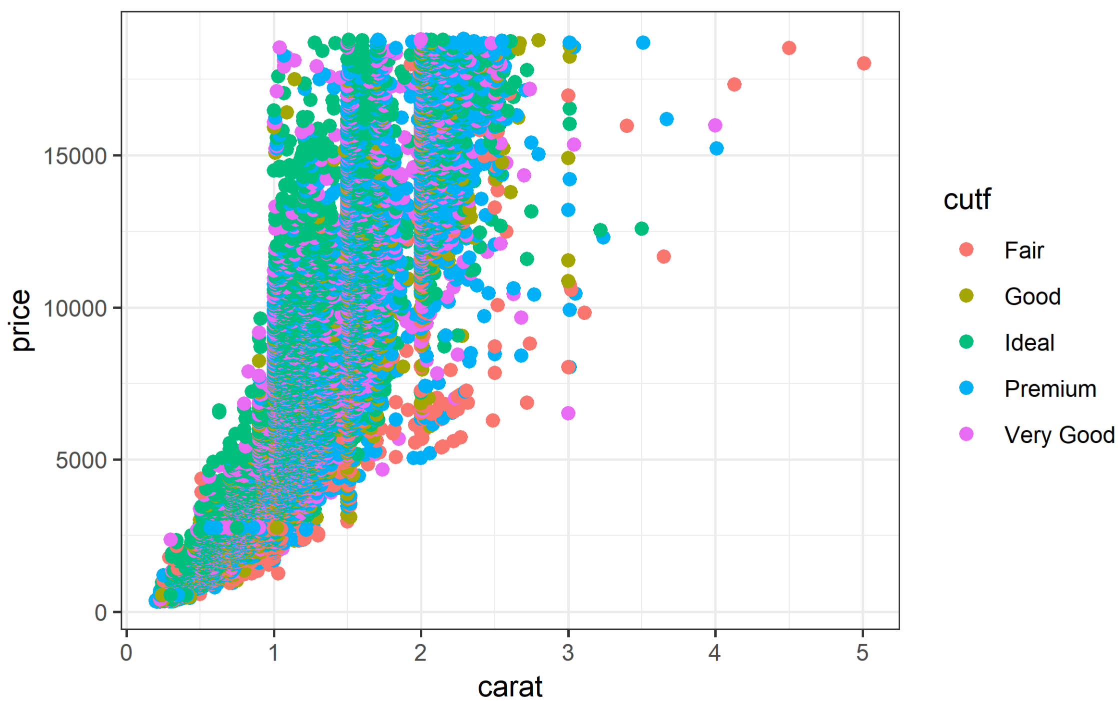
April 8, 2019

```
library(DataAnalytics)
library(data.table)
library(ggplot2)
library(lfe)
library(foreign)
```

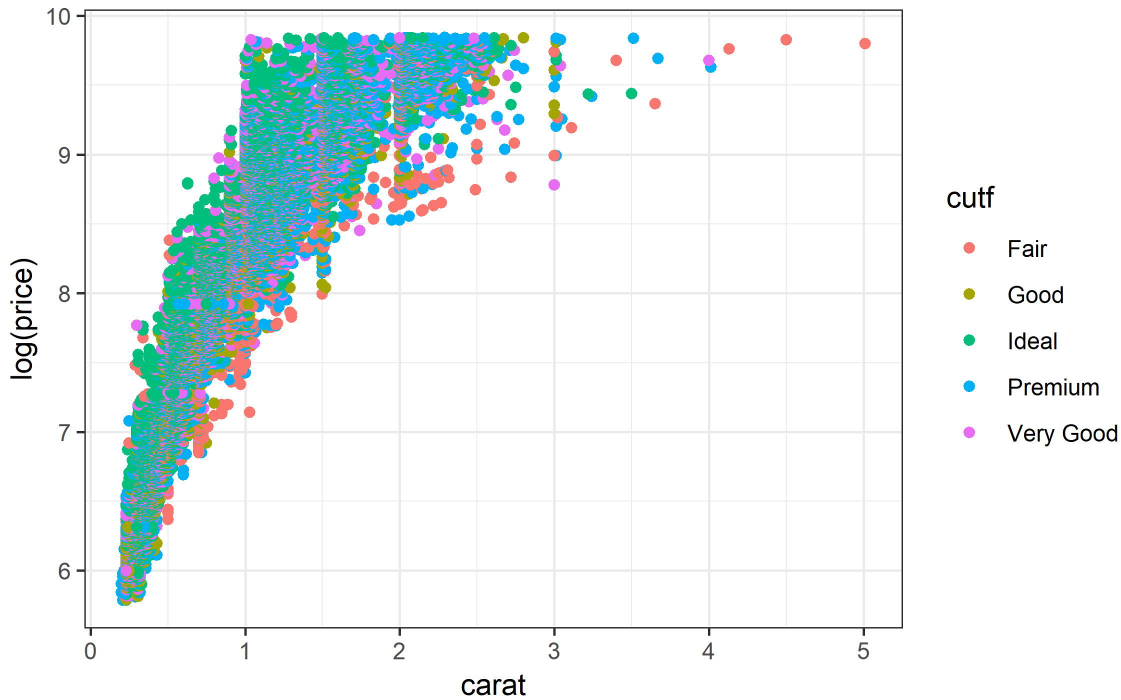
Question 1: More on ggplot2 and regression planes

1. Use ggplot2 to visualize the relationship between price and carat and cut. price in the dependent variable. Consider both the `log()` and `sqrt()` transformation of price.

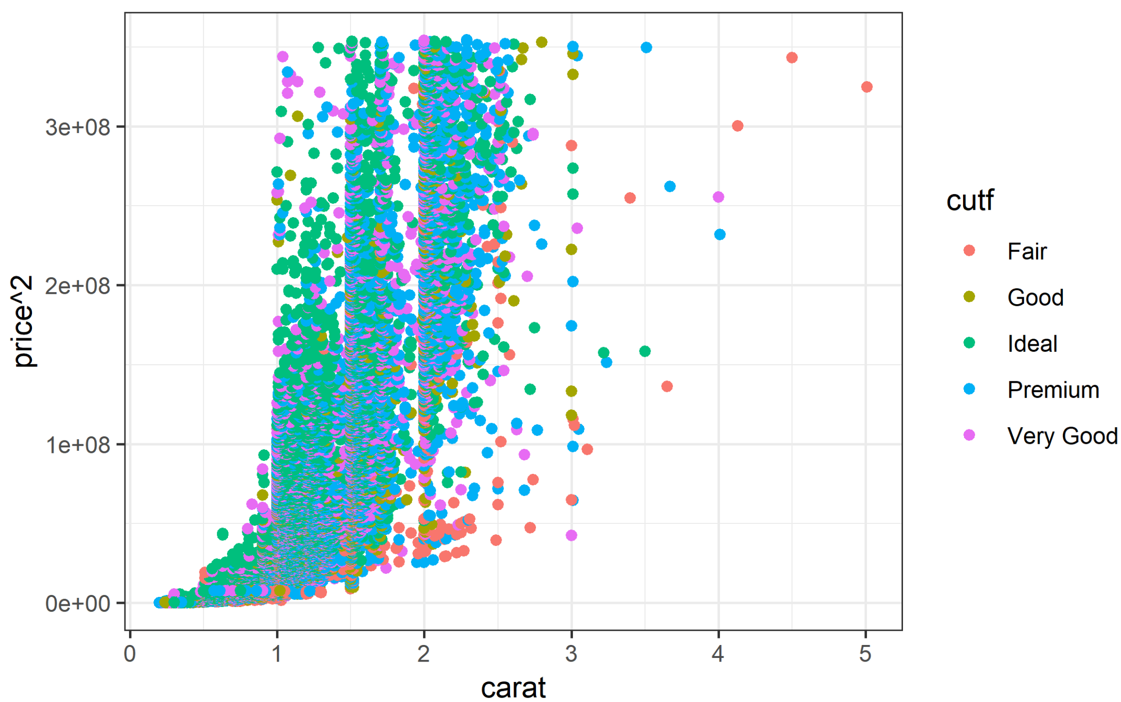
```
data(diamonds)
diamonds <- as.data.table(diamonds)
diamonds[, `:=`(cutf, as.character(cut))][, `:=`(cutf, as.factor(cutf))]
cutf <- as.character(diamonds$cut)
cutf <- as.factor(cutf)

ggplot(diamonds, aes(x = carat, y = price, color = cutf)) + geom_point(size = 2) +
    theme_bw()
```



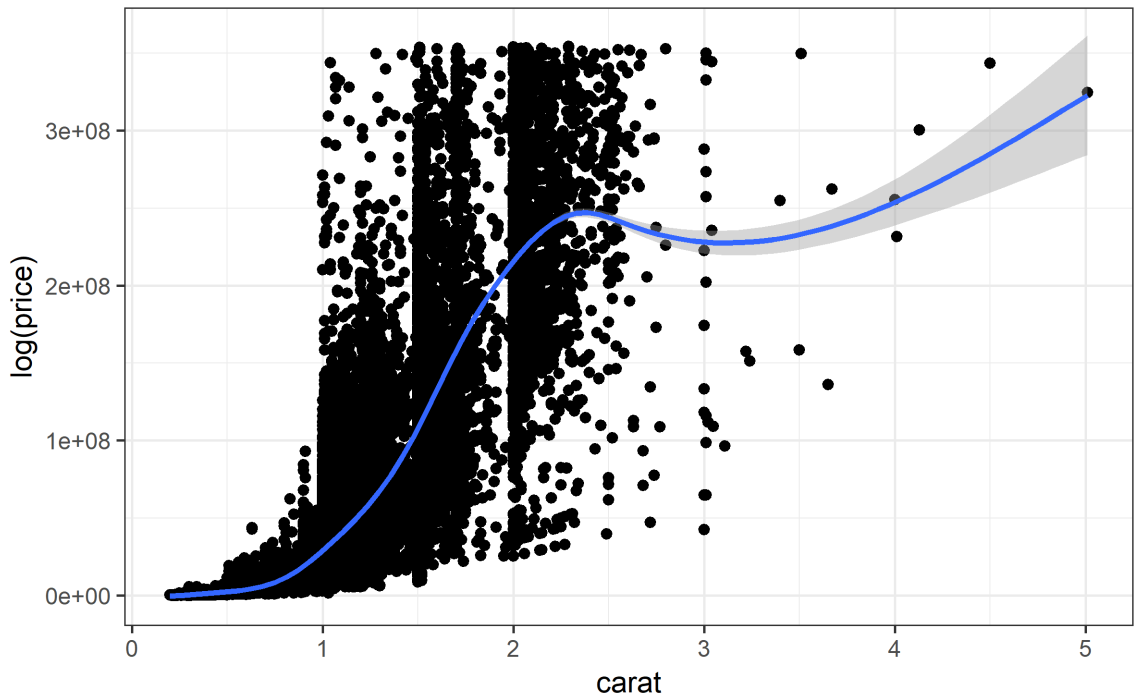

```
ggplot(diamonds, aes(x = carat, y = log(price), color = cutf)) + geom_point() +
    ylab("log(price)") + theme_bw()
```

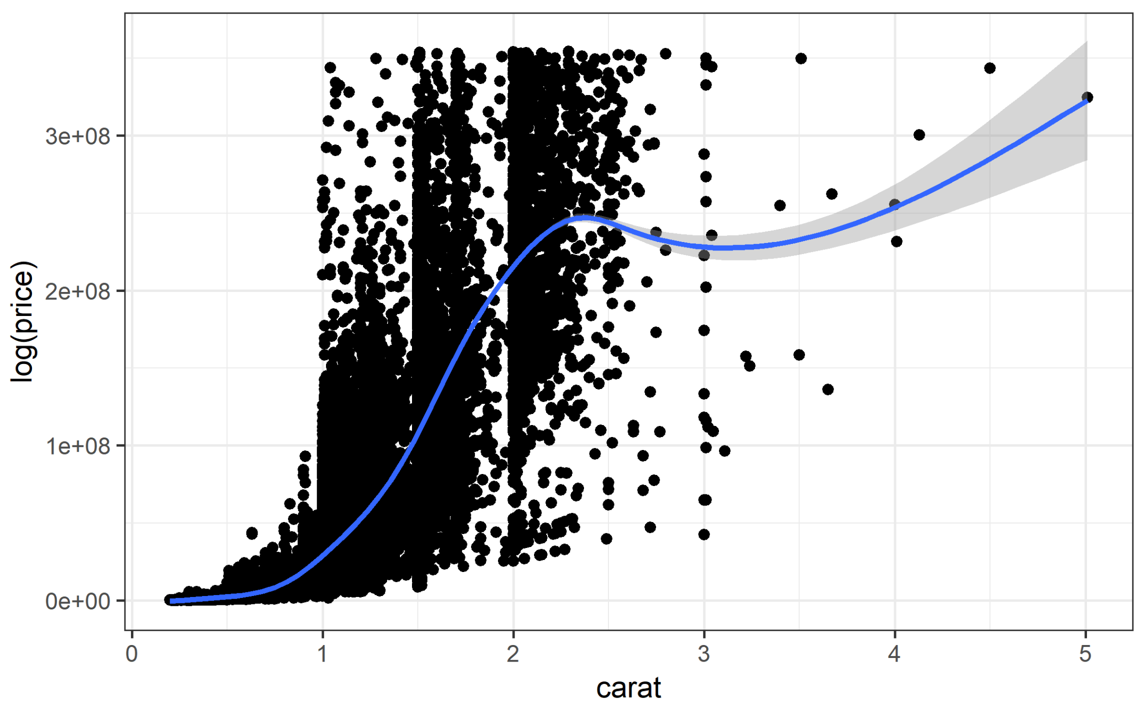



```r
ggplot(diamonds, aes(x = carat, y = (price)^2, color = cutf)) + geom_point() +
    ylab("price^2") + theme_bw()
```



You can also add a smoothed line to the data



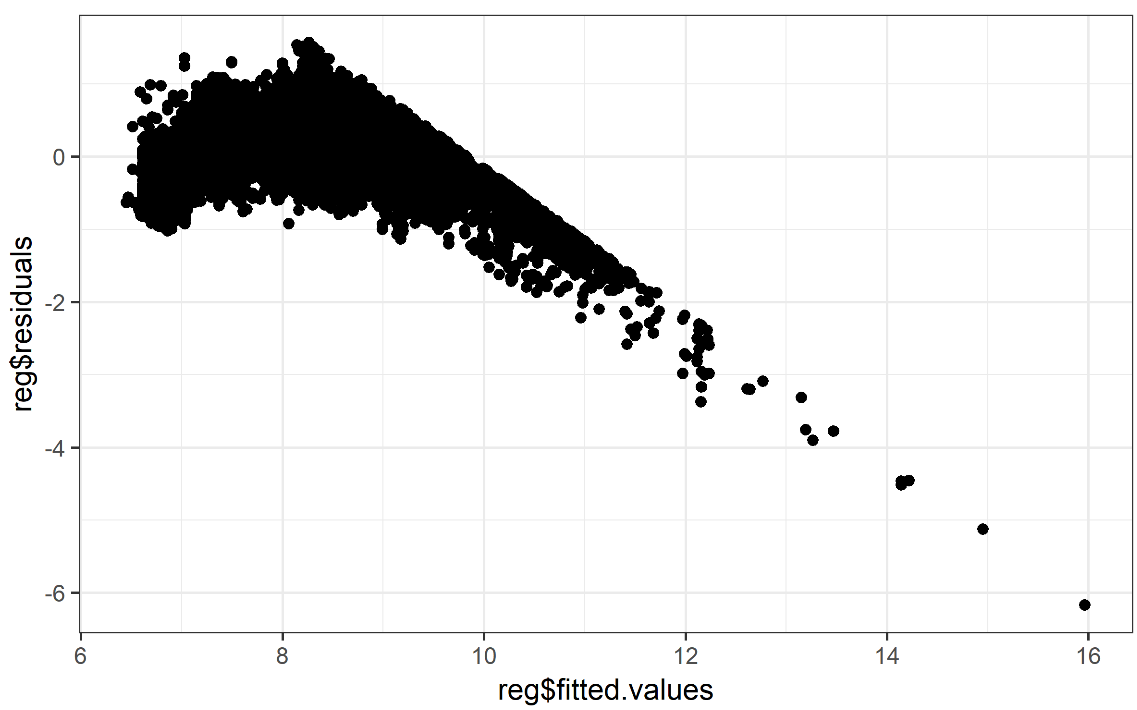


2. Run a regression of your preferred specification. Perform residual diagnostics as you learned in 237Q,1. What do you conclude from your regression diagnostic plots of residuals vs. fitted and residuals vs. carat?

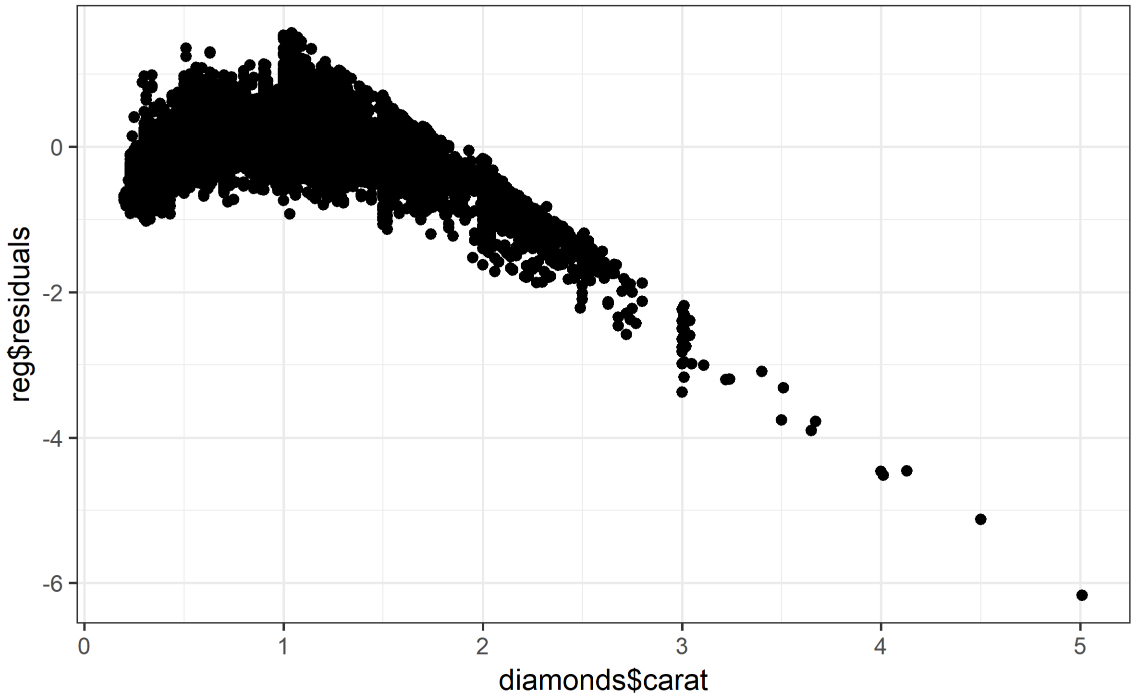
```r
reg <- felm(log(price) ~ carat + cutf, data = diamonds)
summary(reg)
```

```
##
## Call:
##    felm(formula = log(price) ~ carat + cutf, data = diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1680 -0.2439  0.0337  0.2576  1.5651
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.01544    0.01055  570.11   <2e-16 ***
## carat          1.98636    0.00365  544.16   <2e-16 ***
## cutfGood       0.14058    0.01136   12.38   <2e-16 ***
## cutfIdeal      0.22794    0.01027   22.19   <2e-16 ***
## cutfPremium    0.16361    0.01041   15.72   <2e-16 ***
## cutfVery Good  0.18146    0.01051   17.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3947 on 53934 degrees of freedom
## Multiple R-squared(full model): 0.8487   Adjusted R-squared: 0.8487
## Multiple R-squared(proj model): 0.8487   Adjusted R-squared: 0.8487
## F-statistic(full model):6.052e+04 on 5 and 53934 DF, p-value: < 2.2e-16
## F-statistic(proj model): 6.052e+04 on 5 and 53934 DF, p-value: < 2.2e-16
```

```r
qplot(reg$fitted.values,reg$residuals) + theme_bw()
```

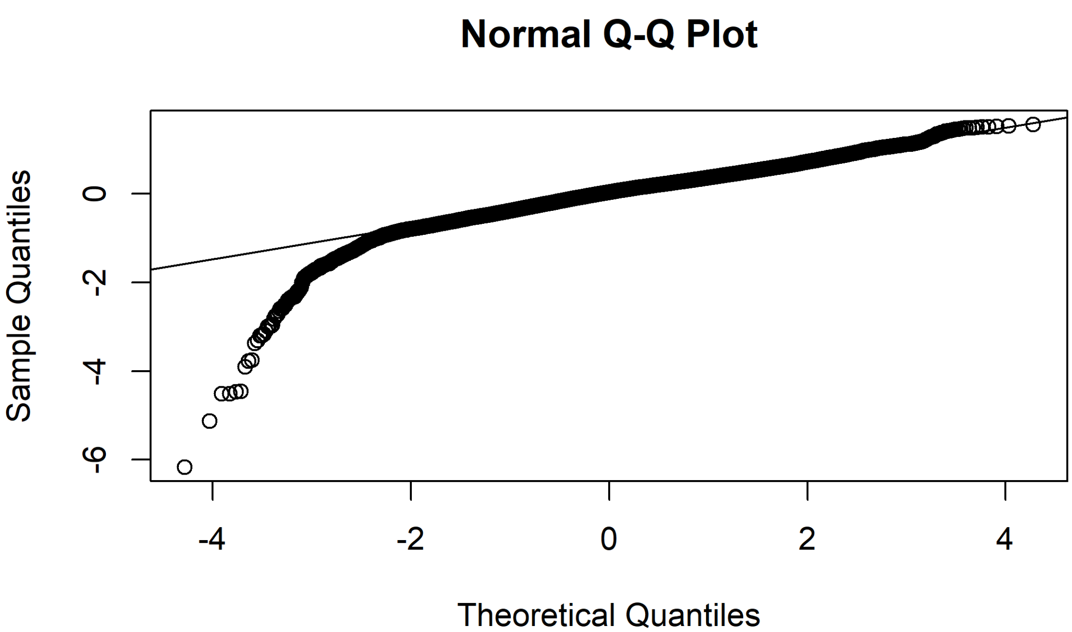


```r
qplot(diamonds$carat,reg$residuals) + theme_bw()
```

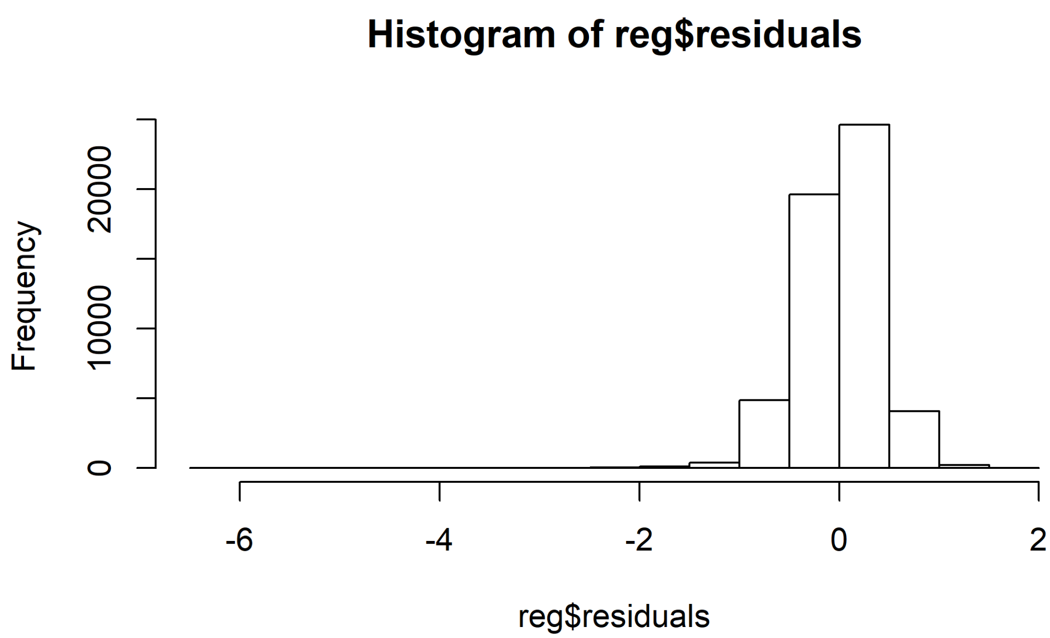


A Normal probability plot of the residuals can be used to check the normality assumption. Here each residual is plotted against its expected value under normality. We can also plot the histogram of residuals.

```r
qqnorm(reg$residuals)
qqline(reg$residuals)
```



```r
hist(reg$residuals, breaks = 20)
```



Question 2 : Nonlinear relations

a. Construct decile sorts (10 portfolios) as in the class notes, but now based on the issuance variable `lnIssue`. Give the average return to each decile portfolio, value-weighting stocks within each portfolio each year, equal-weighting across years.

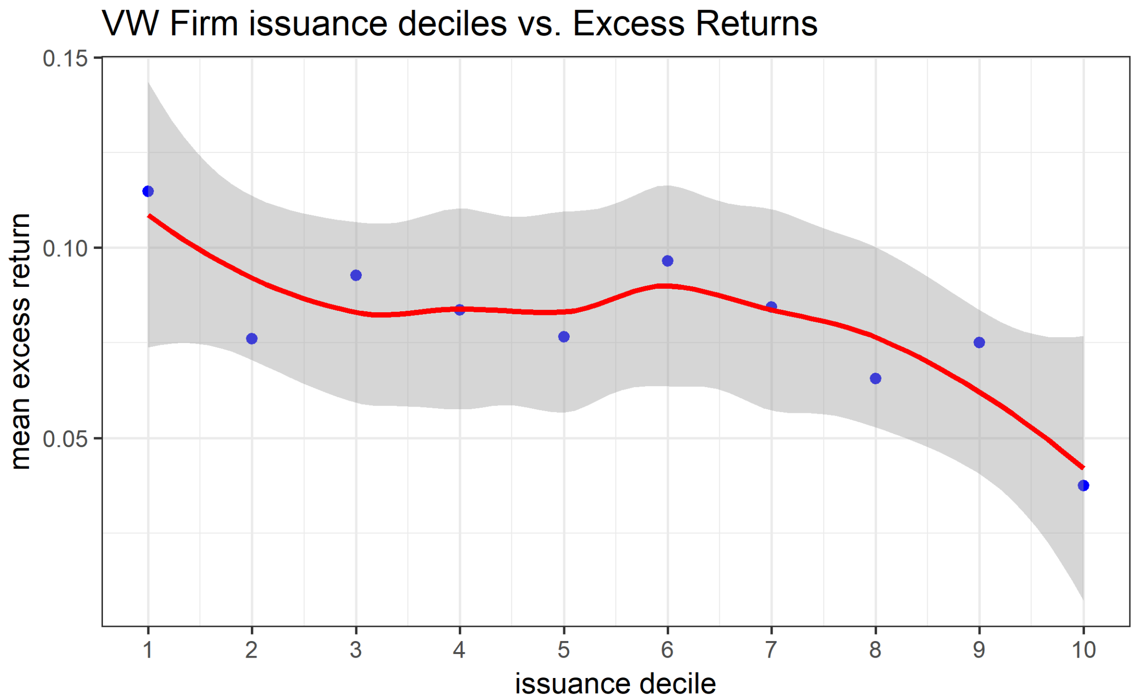
```
# we need the foreign package to import data in different format
rm(list = ls())
# Download data and set as data.table
StockRetAcct_DT <- as.data.table(read.dta("StockRetAcct_insample.dta"))
# set keys for StockRetAcct_DT
setkey(StockRetAcct_DT, FirmID, year)
# create excess returns in levels
StockRetAcct_DT[, `:=`(ExRet, exp(lnAnnRet) - exp(lnRf))]
# due to winsorizing of original data, we add a tiny amount of noise
# (jitter) to lnIssue, lnBM, and lnME before creating by year
# deciles/quintiles this is to avoid ties in the quantile sorts
StockRetAcct_DT[, `:=`(lnIssue, jitter(lnIssue, amount = 0))]
StockRetAcct_DT[, `:=`(lnBM, jitter(lnBM, amount = 0))]
StockRetAcct_DT[, `:=`(lnME, jitter(lnME, amount = 0))]
# create decile sorts based on lnIssue
StockRetAcct_DT[, `:=`(issue_decile_yr, cut(lnIssue, breaks = quantile(lnIssue,
    probs = c(0:10)/10, na.rm = TRUE), include.lowest = TRUE, labels = FALSE)),
    by = year]
# get the average return for each portfolio (VW across stocks, EW across
# years)
EW_ISSUE_MutualFunds_yr <- StockRetAcct_DT[, .(MeanExRetYr = weighted.mean(ExRet,
    MEwt)), by = .(issue_decile_yr, year)]
# then average across years
EW_ISSUE_MutualFunds_yr <- EW_ISSUE_MutualFunds_yr[, .(MeanExRet = mean(MeanExRetYr)),
    by = issue_decile_yr]
setkey(EW_ISSUE_MutualFunds_yr, issue_decile_yr)
EW_ISSUE_MutualFunds_yr[!is.na(issue_decile_yr)]
```

```
##     issue_decile_yr  MeanExRet
##  1:               1 0.11476799
##  2:               2 0.07603598
##  3:               3 0.09272832
##  4:               4 0.08368063
##  5:               5 0.07661145
##  6:               6 0.09647739
##  7:               7 0.08436958
##  8:               8 0.06560777
##  9:               9 0.07503727
## 10:              10 0.03743360
```

b. Plot the average return to these 10 portfolios, similar to what we did in the Topic 2e-f notes. Discuss whether the pattern seems linear or not.

```
ggplot(EW_ISSUE_MutualFunds_yr[!is.na(issue_decile_yr)], aes(x = issue_decile_yr,
    y = MeanExRet)) + geom_point(col = "blue") + geom_smooth(col = "red") +
    theme_bw() + scale_x_continuous(breaks = 1:10) + xlab("issuance decile") +
```

```
    ylab("mean excess return") + ggtitle("VW Firm issuance deciles vs. Excess Returns")
```



c. Since most of the 'action' is in the extreme portfolios, consider a model where expected returns to stocks is linear in a transformed issuance-characteristic that takes three values: `-1` if the stock's issuance is in Decile 1, 1 if the stock's issuance is in decile 10, and 0 otherwise. Create this transformed issuance variable and run a Fama-MacBeth regression with it. Report the results. What is the nature of the portfolio implied by the Fama-MacBeth regression? That is, what stocks do you go long, short, no position?

```
StockRetAcct_DT[, `:=`(trans_issue_decile_yr, ifelse(issue_decile_yr == 1, -1,
    ifelse(issue_decile_yr == 10, 1, 0)))]

# Fama-MacBeth Regressions
port_ret = StockRetAcct_DT[, .(lambda = felm(ExRet ~ trans_issue_decile_yr,
    na.action = na.omit)$coef[2]), by = year]
fm_output = list(MeanReturn = mean(port_ret$lambda), StdReturn = sqrt(var(port_ret$lambda)),
    SR_Return = mean(port_ret$lambda)/sqrt(var(port_ret$lambda)), tstat_MeanRet = sqrt(1 +
        2014 - 1980) * mean(port_ret$lambda)/sqrt(var(port_ret$lambda)))
fm_output
```

```
## $MeanReturn
## [1] -0.03355347
##
## $StdReturn
## [1] 0.05968752
##
## $SR_Return
## [1] -0.5621523
##
## $tstat_MeanRet
## [1] -3.325738
```

This portfolio goes long on stocks in decile 10 and short on stocks in decile 1 and takes no positions on the stocks in the other 8 deciles. Based on the results of the Fama-Macbeth regression, we would want to hold the opposite; go long on stocks in decile 1, and short on stocks in decile 1.

d. Repeat the procedure in c. using `MEwt` as the weighting variable.

```
# Fama-MacBeth Regressions loop through the years in the database
port_ret = NULL
for (i in 1980:2014) {
    temp = StockRetAcct_DT[year == i]
    fit_yr <- felm(ExRet ~ trans_issue_decile_yr, data = temp, weights = temp$MEwt,
        na.action = na.omit)  #weights = temp$MEwt, data = temp)
    temp <- coefficients(fit_yr)
    port_ret = rbind(port_ret, temp[2])
}
fm_output = list(MeanReturn = mean(port_ret), StdReturn = sqrt(var(port_ret)),
    SR_Return = mean(port_ret)/sqrt(var(port_ret)), tstat_MeanRet = sqrt(1 +
        2014 - 1980) * mean(port_ret)/sqrt(var(port_ret)))
fm_output
```

```
## $MeanReturn
## [1] -0.03769551
##
## $StdReturn
##                       trans_issue_decile_yr
## trans_issue_decile_yr            0.05353883
##
## $SR_Return
##                       trans_issue_decile_yr
## trans_issue_decile_yr             -0.704078
##
## $tstat_MeanRet
##                       trans_issue_decile_yr
## trans_issue_decile_yr             -4.165382
```

The resulting portfolio does not correspond to the $1 long/short portfolio. Relative to the standard Fama-Macbeth, this method over-weights stocks with larger market equities.

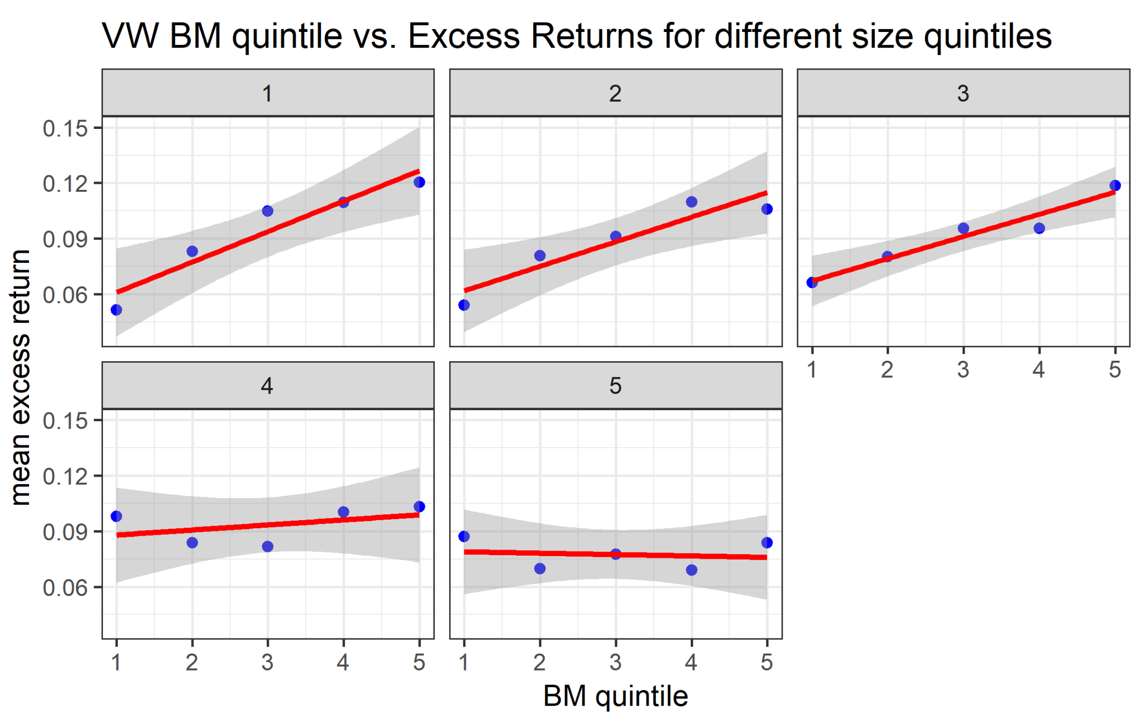
Question 3: Double-sorts and functional forms

a. Create independent quintile sorts based on book-to-market (`lnBM`) and size (`lnME`). That is create a quintile variable by year for book-to-market and then create a quintile variable by year for size.

```
StockRetAcct_DT[, `:=`(bm_quintile_yr = cut(lnBM, breaks = quantile(lnBM, probs = c(0:5)/5,
    na.rm = TRUE, include.lowest = TRUE), labels = FALSE), size_quintile_yr = cut(lnME,
    breaks = quantile(lnME, probs = c(0:5)/5, na.rm = TRUE, include.lowest = TRUE),
    labels = FALSE)), by = year]
```

b. For each size quintile, plot the average returns to the five book-to-market quintile portfolios. So, for size quintile 1, and book-to-market quintile 3, the stocks in this portfolio all have size quintile equal to 1 and book-to-market quintile equal to 3. Thus, I'm looking for five plots here, one for each size quintile.

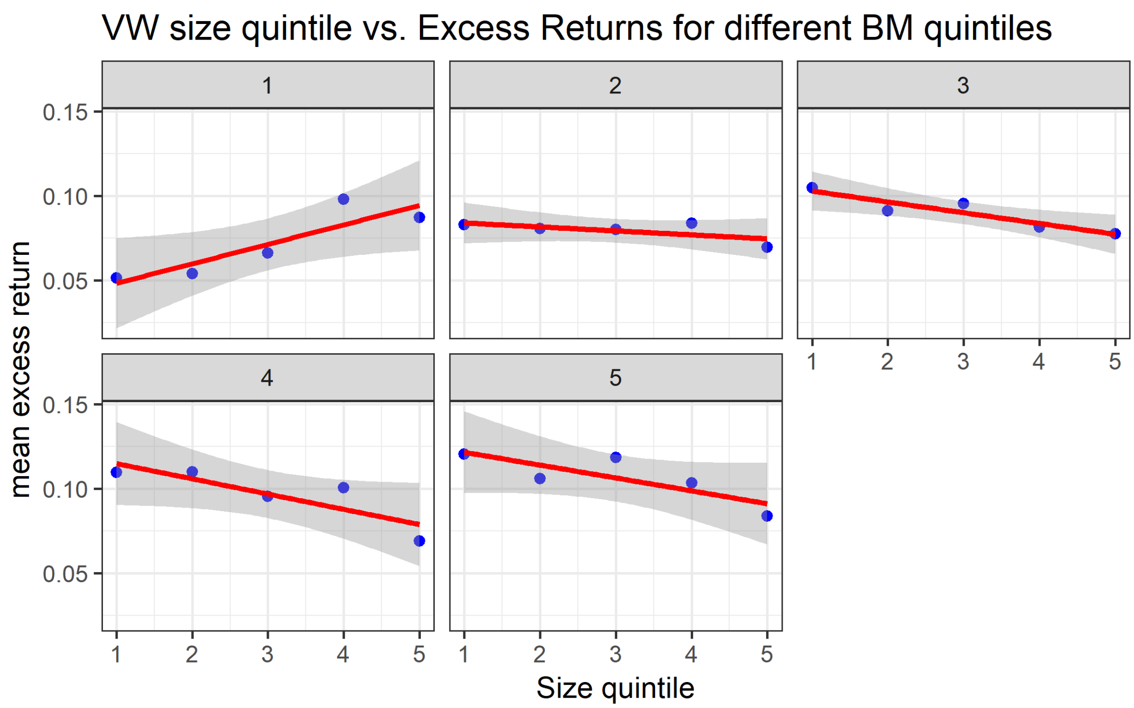
```
# get the average return for each portfolio (VW across stocks, EW across
# years)
EW_SIZE_BM_MutualFunds_yr <- StockRetAcct_DT[!is.na(bm_quintile_yr) & !is.na(size_quintile_yr),
    .(MeanExRetYr = weighted.mean(ExRet, MEwt)), by = .(bm_quintile_yr, size_quintile_yr,
        year)]

# then average across years
EW_SIZE_BM_MutualFunds_yr <- EW_SIZE_BM_MutualFunds_yr[, .(MeanExRet = mean(MeanExRetYr)),
    by = .(bm_quintile_yr, size_quintile_yr)]

ggplot(EW_SIZE_BM_MutualFunds_yr, aes(x = bm_quintile_yr, y = MeanExRet)) +
    geom_point(col = "blue") + geom_smooth(col = "red", method = "lm") + theme_bw() +
    xlab("BM quintile") + ylab("mean excess return") + facet_wrap(~size_quintile_yr) +
    ggtitle("VW BM quintile vs. Excess Returns for different size quintiles")
```



From the plots above, we see that the conditional linearity assumption (that expected returns are linear in the BM ratio as well as the interaction between BM and size) seems to be a pretty good assumption.

c. For each book-to-market quintile, plot the average returns to the five size quintile portfolios.

```r
ggplot(EW_SIZE_BM_MutualFunds_yr, aes(x = size_quintile_yr, y = MeanExRet)) +
    geom_point(col = "blue") + geom_smooth(col = "red", method = "lm") + theme_bw() +
    xlab("Size quintile") + ylab("mean excess return") + facet_wrap(~bm_quintile_yr) +
    ggtitle("VW size quintile vs. Excess Returns for different BM quintiles")
```



From the plots above, we see that the conditional linearity assumption (that expected returns are linear in the size ratio as well as the interaction between BM and size) seems to be a pretty good assumption.