

Financial Econometrics

Non-linear functional form

Darya Yuferova

Does managerial ownership affect firm value?

In order to incentivize management to act in shareholder's interests, they are often given ownership in the firm.

The rationale is simple, given that the manager holds ownership, his wealth increases if the stock price increases, which is the ultimate goal of the shareholders. Hence, given that management's ownership increases, their own interests align with shareholders.

However, if the management's stake is too big, their voting power will make their work place more secure and might reduce their incentive to provide effort. In addition, the probability of a hostile takeover (a market discipline channel) decreases.

In short, while initially an increase in managerial ownership seems beneficial, if the ownership share is too large it might have adverse effects. The described relationship is non-linear and hence a simple linear model might not be the best approach. However, a non-linear model allows us to estimate this relationship.

Data provided in `MB_Own_1993.csv`:

- Firm characteristics for a sample of firms in 1993,
 - *gvkey*: firm identifier
 - *year*: year
 - *mtbe*: market-to-book ratio (in this assignment, we will use the market-to-book ratio as a proxy for firm value)
 - *own*: share of ownership owned by management

The main goal of this assignment is to introduce you to different functional forms of a linear regression. You will model non-linear relationships between the independent and dependent variable. This assignment also touches on the issue of model selection.

Your task:

- Load the data, investigate, and adjust if necessary
- You need to select a regression model that can describe a non-linear relationship between market-to-book and ownership. In particular, you want to allow that initially there is a positive relationship between both variables that might disappear or potentially reverse if the data supports the hypothesis. You can select between these four models. Which model(s) are appropriate?
 1. $mtbe = \alpha + \beta \times own + u$
 2. $\log(mtbe) = \alpha + \beta \times \log(own) + u$
 3. $mtbe = \alpha + \beta_1 \times own + \beta_2 \times own^2 + u$
 4. $\log(mtbe) = \alpha + \beta_1 \times own + \beta_2 \times D(own > median) + \beta_3 \times own \times D(own > median) + u$
- Estimate all four models and report the results in a table. Is there evidence of the relationship that we hypothesize? Make sure that you use heteroskedasticity robust standard errors.

- Display the predicted relationship in two plots. In each plot, you display ownership on the x-axis and market-to-book on the y-axis. Each line indicates the prediction based on each of the four models. In the first plot, you use raw values for both axes while you use logarithmic values for the second plot.
 - Small coding advice. Construct a new data.frame with values for x on a reasonable interval. Then, use the estimated regression equations to predict y values for each model which you add to the new data.frame. If you use *plot()* to construct the plot, you can use *lines()* to add additional lines.
- Display two histograms for ownership. Similar to before, either use the raw data or the log transform.
- How do you reconcile these findings with our initial assumption of a linear relationship between dependent and independent variable(s) when introducing OLS?

```
#####
#      Solution      #
#####

# Packages
require(DescTools)

## Warning: package 'DescTools' was built under R version 4.5.2
require(car)

## Warning: package 'carData' was built under R version 4.5.2
require(lmtest)

## Warning: package 'lmtest' was built under R version 4.5.2
## Warning: package 'zoo' was built under R version 4.5.2
require(stargazer)

## Warning: package 'stargazer' was built under R version 4.5.2
require(ggplot2)
require(dplyr)

## Warning: package 'dplyr' was built under R version 4.5.2
# set working directory
setwd("C:/Users/s13163/Dropbox/FIE401/data/data_labs/")

# load the data
DATA <- read.csv("MB_Own_1993.csv")

# Winsorizing
# both variables have some suspicious observations
DATA$mtbe <- Winsorize(DATA$mtbe, val=quantile(DATA$mtbe, probs=c(0.01, 0.99), na.rm=T))
DATA$own <- Winsorize(DATA$own, val=quantile(DATA$own, probs=c(0.01, 0.99), na.rm=T))

# Given that I use log transforms, I will remove negative and zero observations
DATA$mtbe[DATA$mtbe <= 0] <- NA
DATA$own[DATA$own <= 0] <- NA
DATA<-na.omit(DATA)

# additional dummy variable
DATA$own.above.median <- DATA$own > median(DATA$own, na.rm=T)

# regressions model
fit1 <- (lm(mtbe ~ own, data = DATA))
fit2 <- (lm(log(mtbe) ~ log(own), data = DATA))
fit3 <- (lm(mtbe ~ own + I(own^2), data = DATA))
fit4 <- (lm(log(mtbe)~ own * own.above.median, data = DATA))

# make heteroskedasticity robust SEs
se1 <- coeftest(fit1, vcov = hccm)[,2]
se2 <- coeftest(fit2, vcov = hccm)[,2]
se3 <- coeftest(fit3, vcov = hccm)[,2]
se4 <- coeftest(fit4, vcov = hccm)[,2]
```

```
# stargazer output
stargazer(list(fit1, fit2, fit3, fit4),
  se = list(se1, se2, se3, se4),
  type="text",
  keep.stat=c("n", "rsq", "adj.rsq"),
  report=('vc*t'))

##
## =====
##                               Dependent variable:
##                               -----
##                               mtbe    log(mtbe)    mtbe    log(mtbe)
##                               (1)      (2)        (3)      (4)
## -----
## own                          4.438***          11.550***   36.798*
##                               t = 3.929          t = 3.835   t = 1.740
##
## log(own)                      0.038***
##                               t = 4.727
##
## I(own2)                      -26.255***
##                               t = -2.696
##
## own.above.median              0.126***
##                               t = 2.659
##
## own:own.above.median          -35.896*
##                               t = -1.697
##
## Constant                     2.714***   1.072***   2.630***   0.751***
##                               t = 41.144 t = 22.251 t = 37.512 t = 21.192
## -----
## Observations                 1,069      1,069      1,069      1,069
## R2                           0.021      0.025      0.029      0.026
## Adjusted R2                  0.020      0.024      0.027      0.023
## =====
## Note:                        *p<0.1; **p<0.05; ***p<0.01

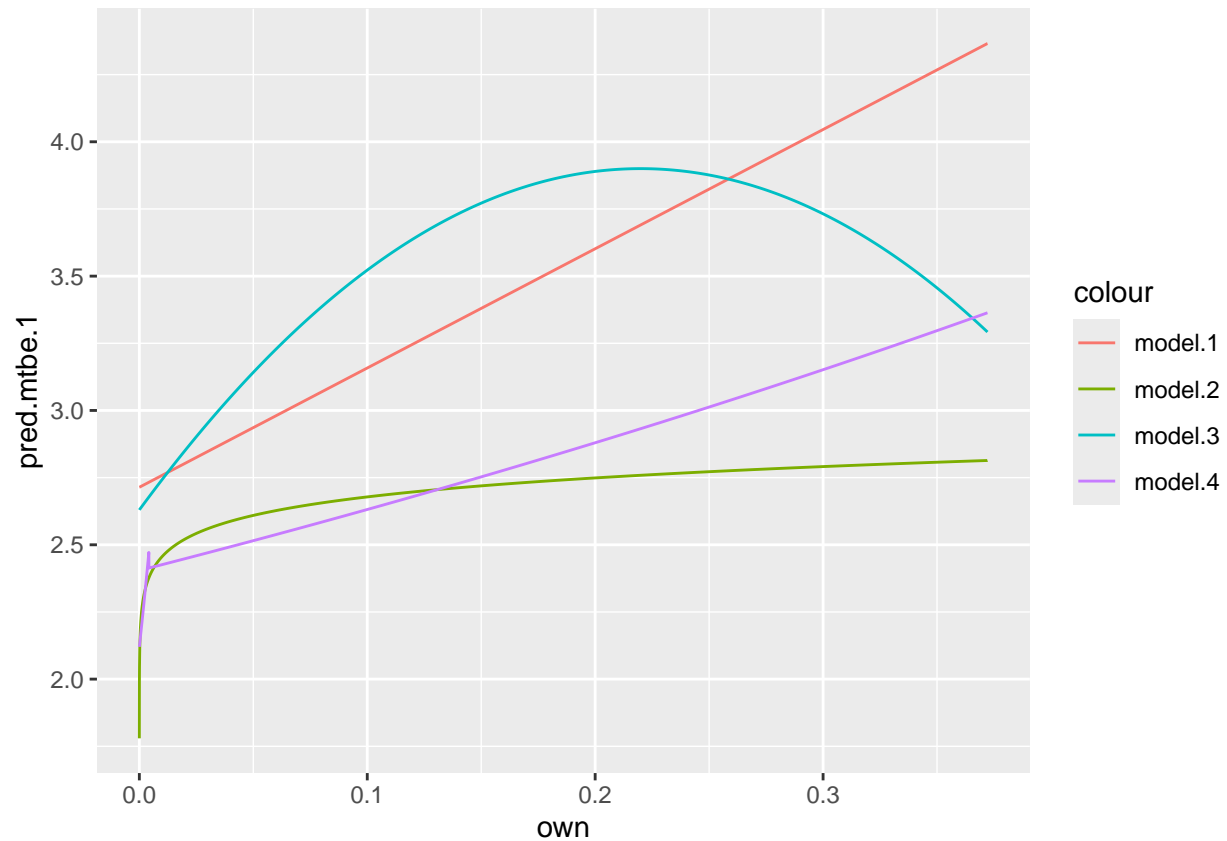
# make a data.frame
pred <- tibble(own = seq(min(DATA$own, na.rm=T), max(DATA$own, na.rm=T), length.out = 10000))

# add the above median variable
pred$own.above.median <- pred$own > median(DATA$own, na.rm=T)

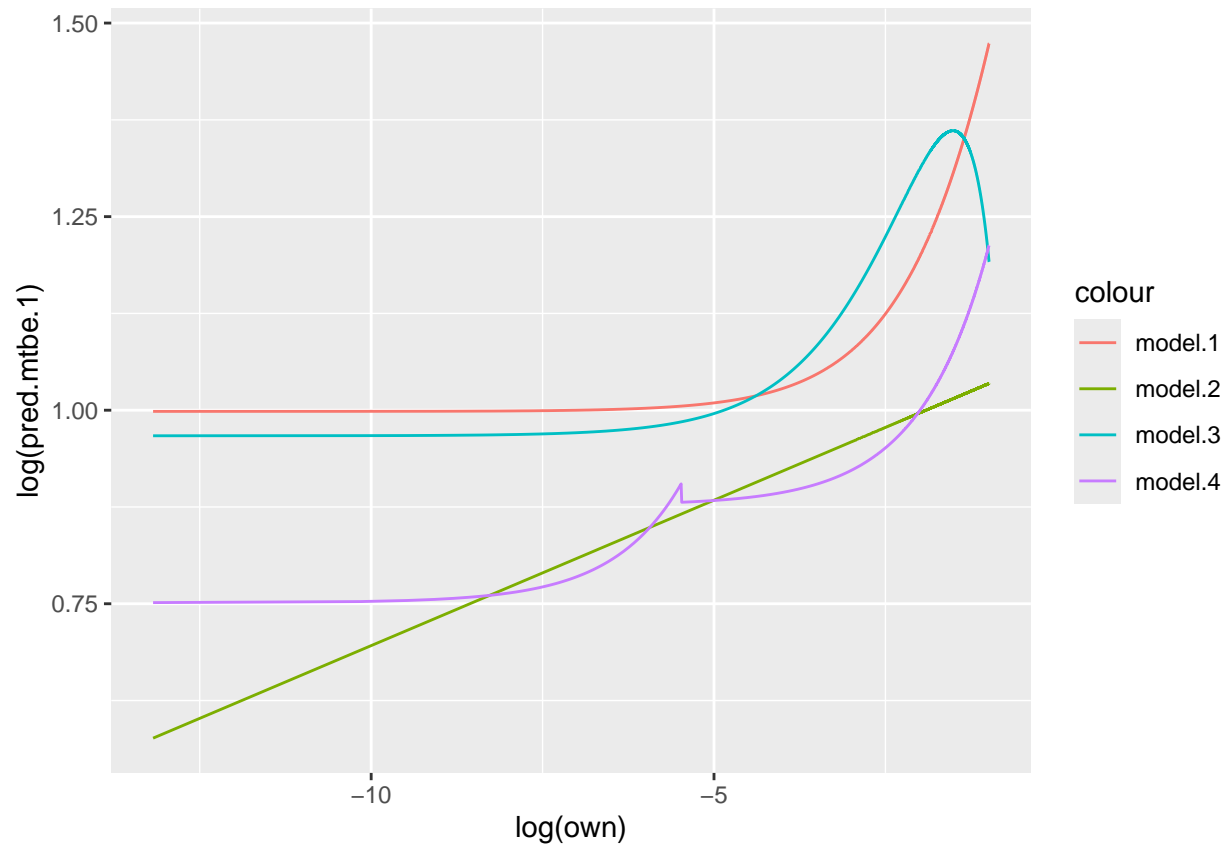
# add prediction from each model
pred$pred.mtbe.1 <- predict(fit1, pred)
pred$pred.mtbe.2 <- exp(predict(fit2, pred)) # i need to take exponent due to the log(y) in fit
pred$pred.mtbe.3 <- predict(fit3, pred)
pred$pred.mtbe.4 <- exp(predict(fit4, pred))

# plot with linear scale (you do not need to use ggplot!)
ggplot(data = pred,
  aes(x = own)) +
```

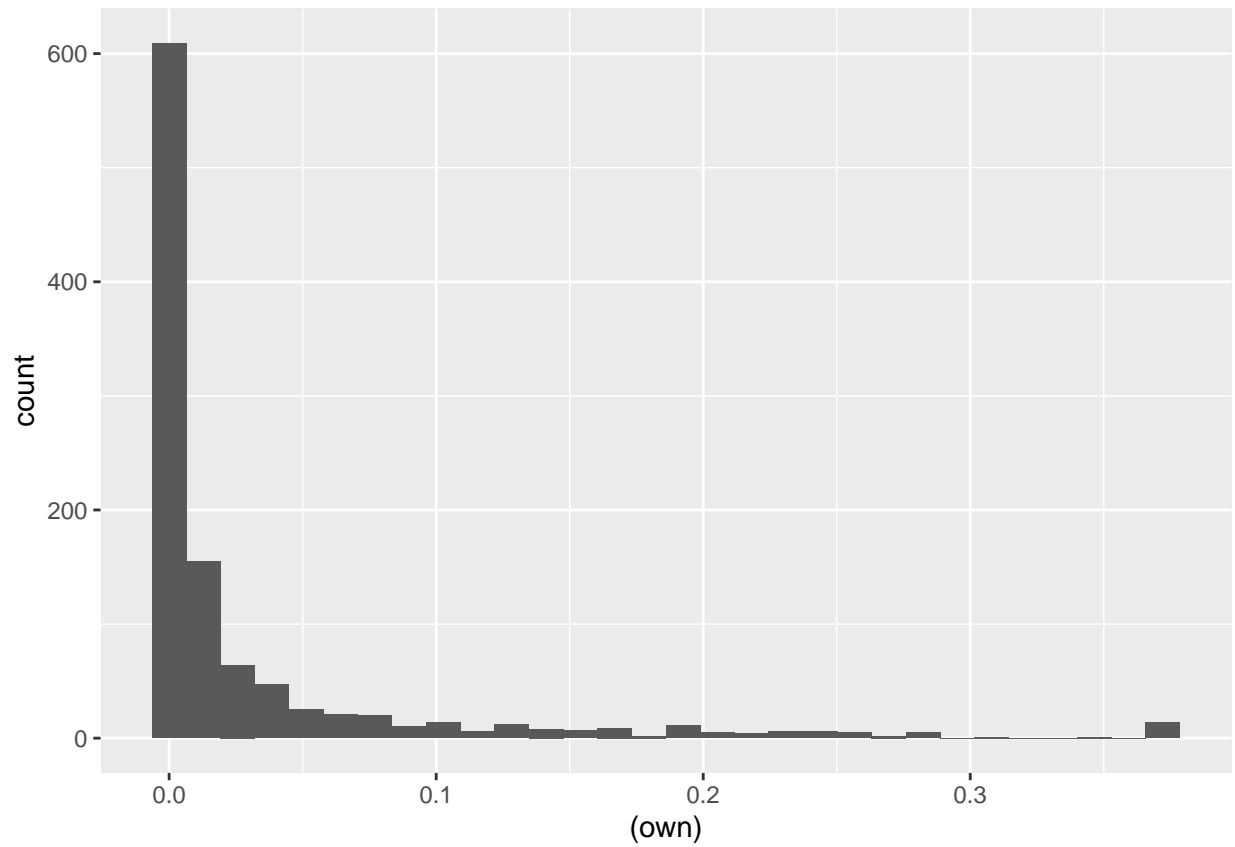
```
geom_line(aes(y = pred.mtbe.1, colour = "model.1")) +
geom_line(aes(y = pred.mtbe.2, colour = "model.2")) +
geom_line(aes(y = pred.mtbe.3, colour = "model.3")) +
geom_line(aes(y = pred.mtbe.4, colour = "model.4"))
```



```
# plot with log scale
ggplot(data = pred,
  aes(x = log(own))) +
  geom_line(aes(y = log(pred.mtbe.1), colour = "model.1")) +
  geom_line(aes(y = log(pred.mtbe.2), colour = "model.2")) +
  geom_line(aes(y = log(pred.mtbe.3), colour = "model.3")) +
  geom_line(aes(y = log(pred.mtbe.4), colour = "model.4"))
```



```
# histogram linear scale
ggplot(data = DATA,
  aes(x = (own))) +
  geom_histogram()
```



```
# histogram log scale  
ggplot(data = DATA,  
  aes(x = log(own))) +  
  geom_histogram()
```

