

# Simple Regression Model

Thach Pham

14/11/2018

The experiment uses R Core Team (2018) and some packages developed by Wickham et al. (2018), Allaire et al. (2018), Hothorn et al. (2018), Hlavac (2018), Wickham (2017), and Shea (2018).

```
library(wooldridge)
library(tidyverse)
library(stargazer)
library(lmtest)
```

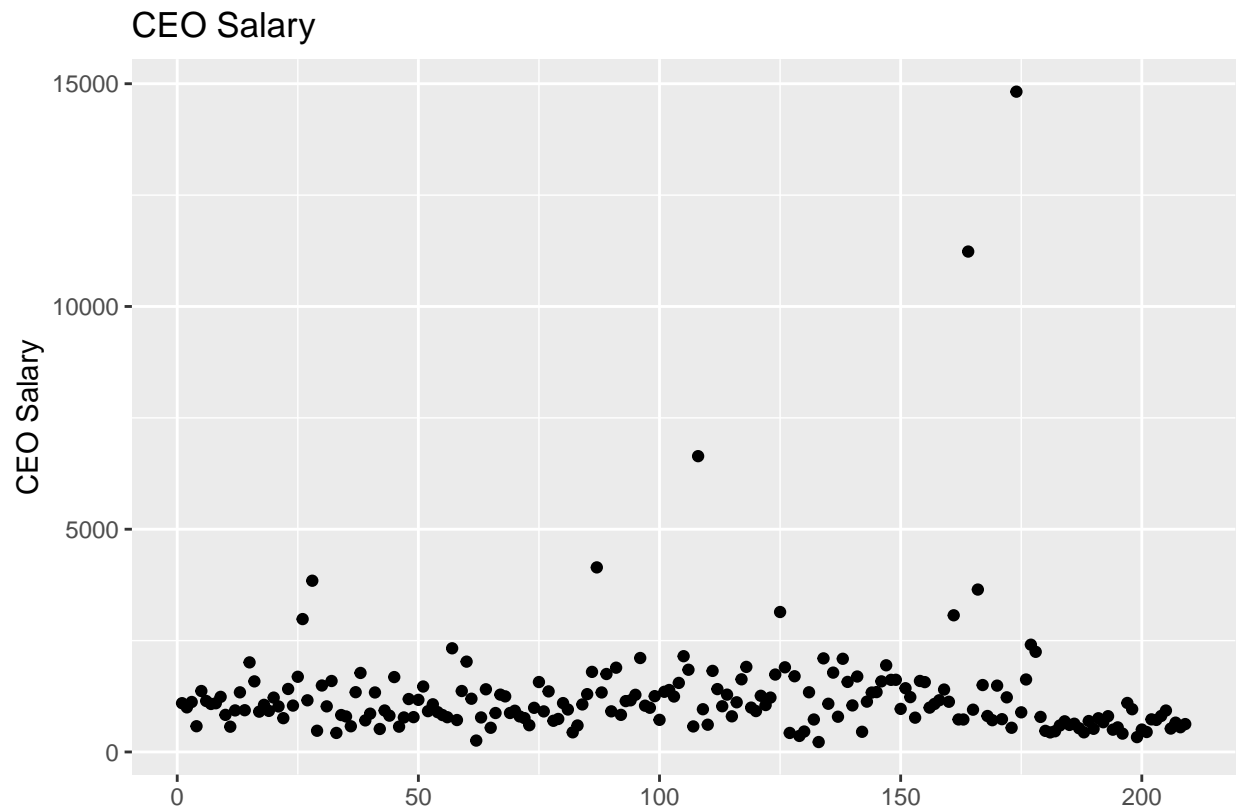
## Descriptive Statistics of ceosal1 data

```
# Descriptive Statistics
data("ceosal1")
desc.ceosal1 <- select(ceosal1, -c(indus, finance, consprod, utility))
stargazer(desc.ceosal1, type = "text", style = "qje", title = "Descriptive Statistics",
           digits = 4, out = "SR0/ceosal1-sum.doc")
```

```
##
## Descriptive Statistics
## -----
## Statistic   N      Mean      St. Dev.      Min      Pctl(25) Pctl(75)      Max
## =====
## salary      209  1,281.1200  1,372.3450     223       736       1,407     14,822
## pcsalary    209   13.2823   32.6339      -61        -1        20        212
## sales       209  6,923.7930 10,633.2700     175     2,210.3     7,177     97,650
## roe         209   17.1842    8.5185       0.5000    12.4000    20.0000    56.3000
## pcroe       209   10.8005    97.2194     -98.9000   -21.2000   19.5000    977.0000
## ros         209   61.8038    68.1771      -58        21        81        418
## lsalary     209    6.9504    0.5664       5.4072     6.6012     7.2492     9.6039
## lsales      209    8.2923    1.0132       5.1659     7.7009     8.8786    11.4891
## =====
```

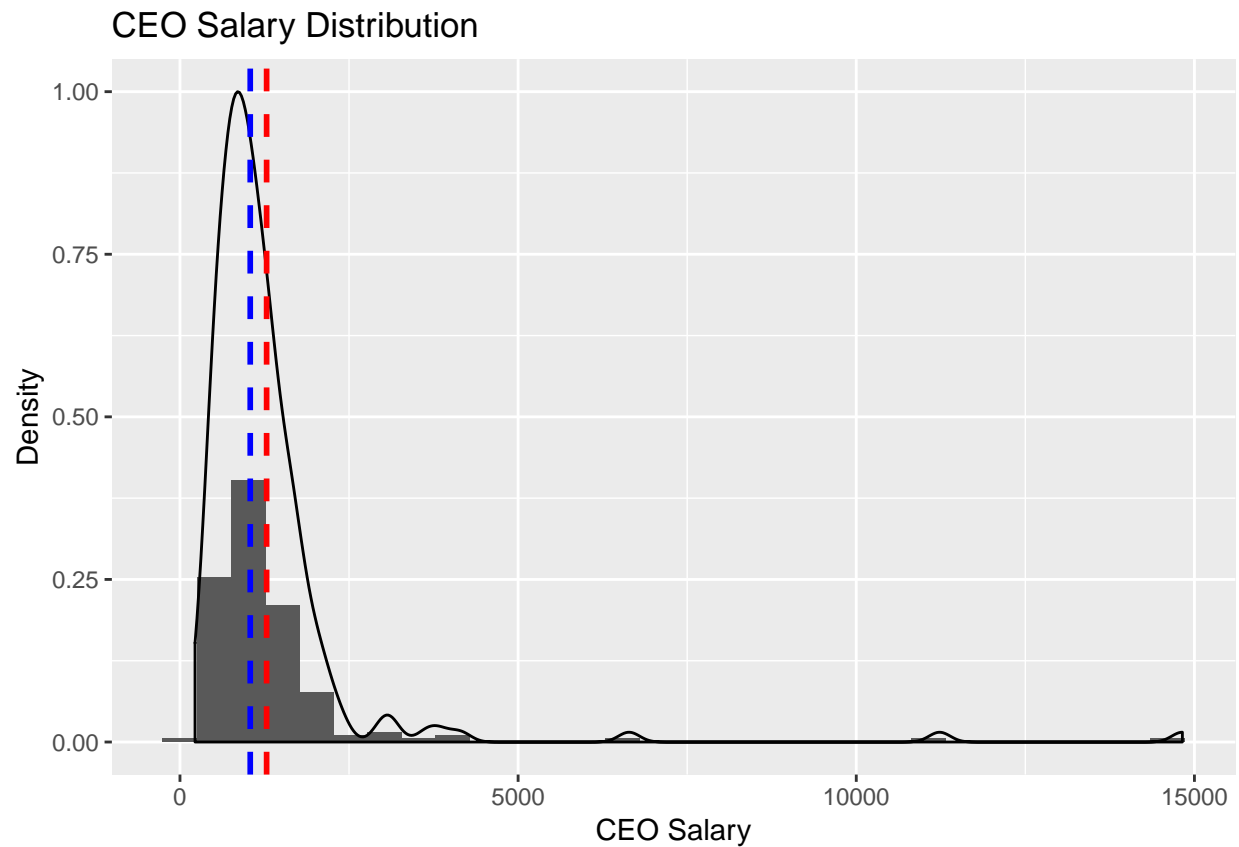
## Salary Plot and Distribution

```
# Salary Plot
(salary <- ggplot(desc.ceosal1, aes(x = 1:nrow(desc.ceosal1), y = salary)) +
  geom_point() +
  theme(legend.position = "none") +
  labs(x = "", y = "CEO Salary", title = "CEO Salary"))
```



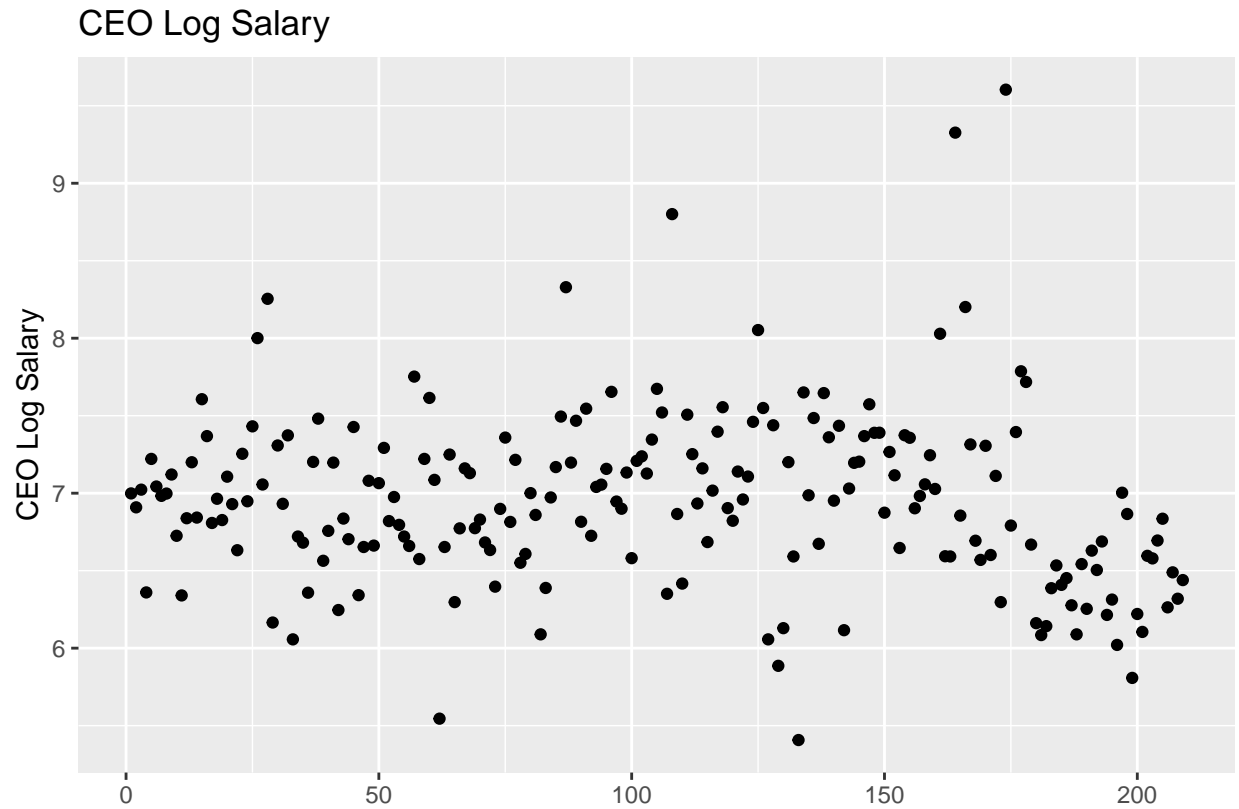
```
ggsave("SRO/CEO Salary.png", salary)

# Salary Distribution
(salary.dist <- ggplot(desc.ceosal1, aes(salary)) +
  geom_histogram(aes(y = ..count../sum(..count..))) +
  theme(legend.position = "none") +
  labs(x = "CEO Salary", y = "Density", title = "CEO Salary Distribution") +
  geom_density(aes(y = ..scaled..)) +
  geom_vline(aes(xintercept = mean(salary)), color = "red", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(salary)), color = "blue", linetype = "dashed", size = 1))
```



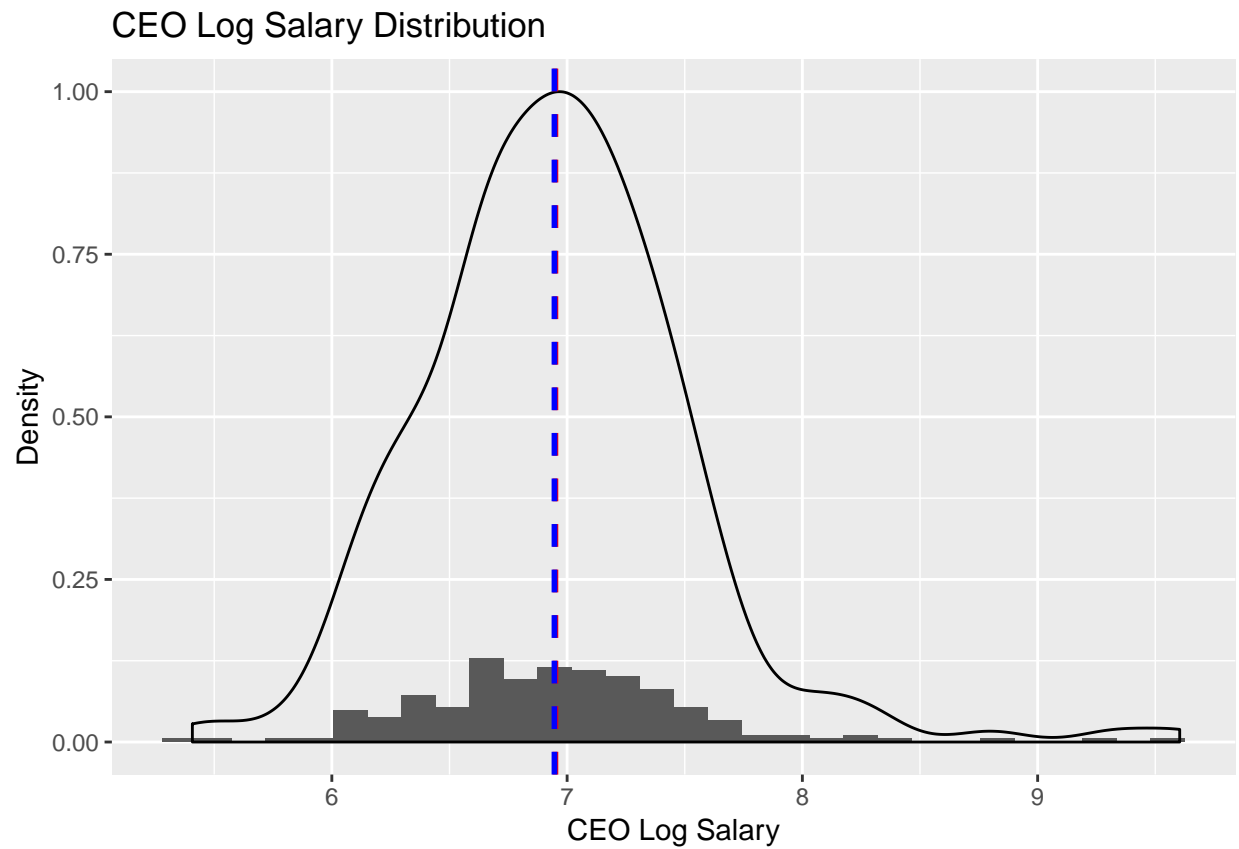
```
ggsave("SRO/CEO Salary Distribution.png", salary.dist)

# Logarithm of Salary
(log.salary <- ggplot(desc.ceosal1, aes(x = 1:nrow(desc.ceosal1), y = lsalary)) +
  geom_point() +
  theme(legend.position = "none") +
  labs(x = "", y = "CEO Log Salary", title = "CEO Log Salary"))
```



```
ggsave("SRO/CEO Log Salary.png", log.salary)

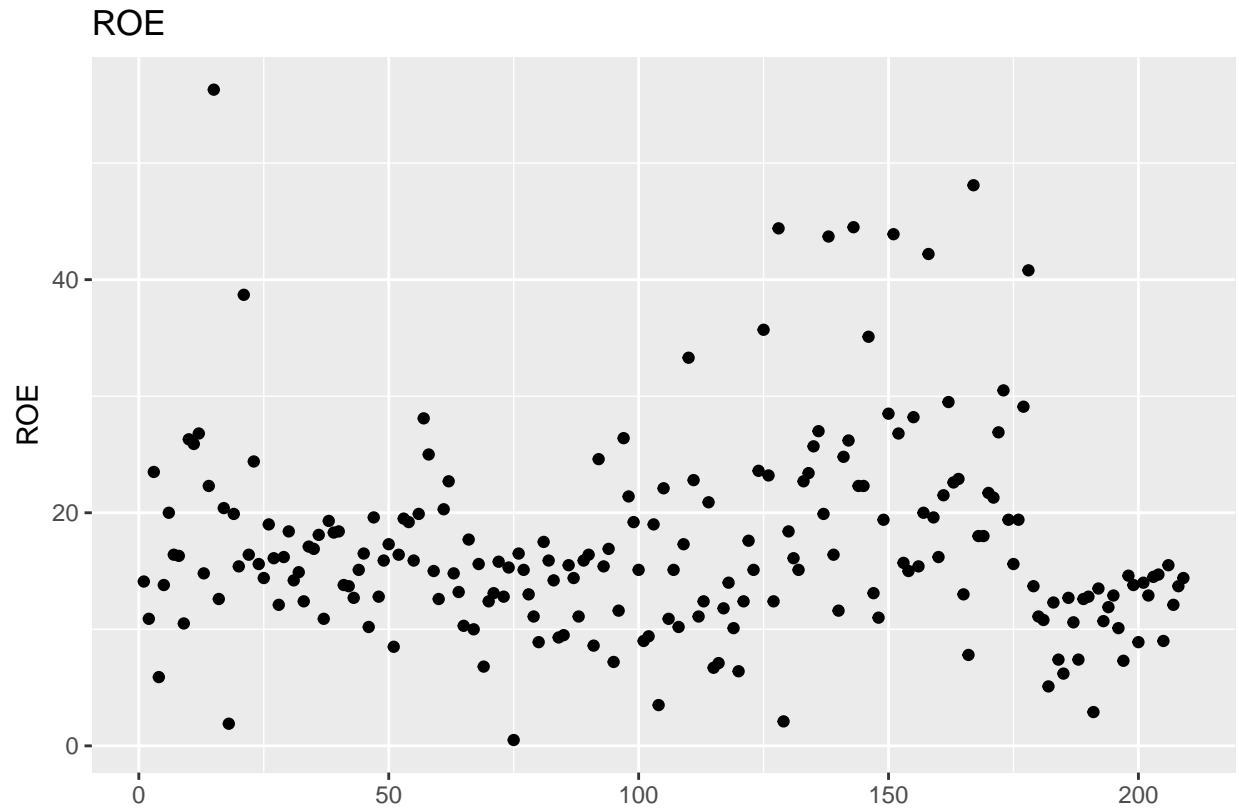
# Log Salary Distribution
(log.salary.dist <- ggplot(desc.ceosal1, aes(lsalary)) +
  geom_histogram(aes(y = ..count../sum(..count..))) +
  theme(legend.position = "none") +
  labs(x = "CEO Log Salary", y = "Density", title = "CEO Log Salary Distribution") +
  geom_density(aes(y = ..scaled..)) +
  geom_vline(aes(xintercept = mean(lsalary)), color = "red", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(lsalary)), color = "blue", linetype = "dashed", size = 1))
```



```
ggsave("SRO/CEO Log Salary Distribution.png", log.salary.dist)
```

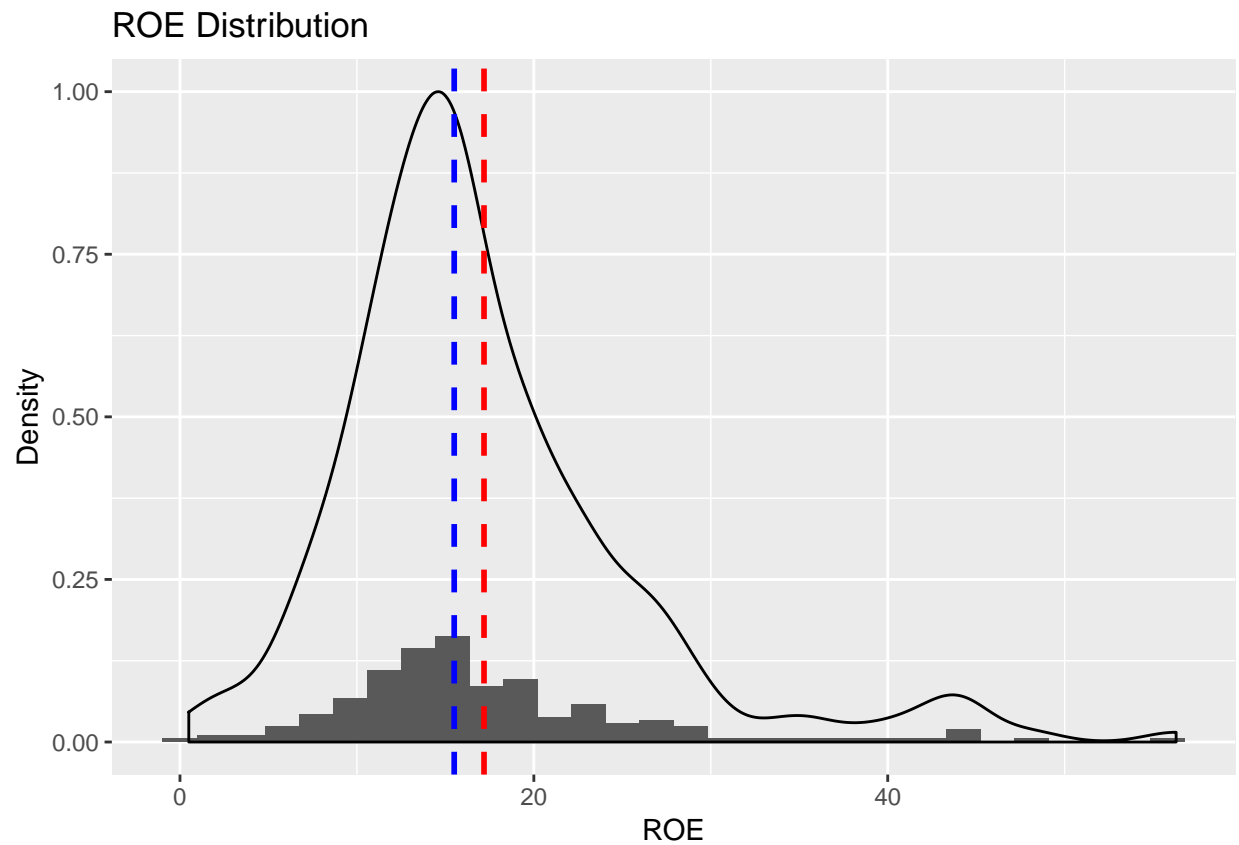
## ROE Plot and Distribution

```
# ROE Plot
(roe <- ggplot(desc.ceosal1, aes(x = 1:nrow(desc.ceosal1), y = roe)) +
  geom_point() +
  theme(legend.position = "none") +
  labs(x = "", y = "ROE", title = "ROE"))
```



```
ggsave("SRO/ROE.png", roe)

# ROE Distribution
(roe.dist <- ggplot(desc.ceosal1, aes(roe)) +
  geom_histogram(aes(y = ..count../sum(..count..))) +
  theme(legend.position = "none") +
  labs(x = "ROE", y = "Density", title = "ROE Distribution") +
  geom_density(aes(y = ..scaled..)) +
  geom_vline(aes(xintercept = mean(roe)), color = "red", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = median(roe)), color = "blue", linetype = "dashed", size = 1))
```



```
ggsave("SR0/ROE Distribution.png", roe.dist)
```

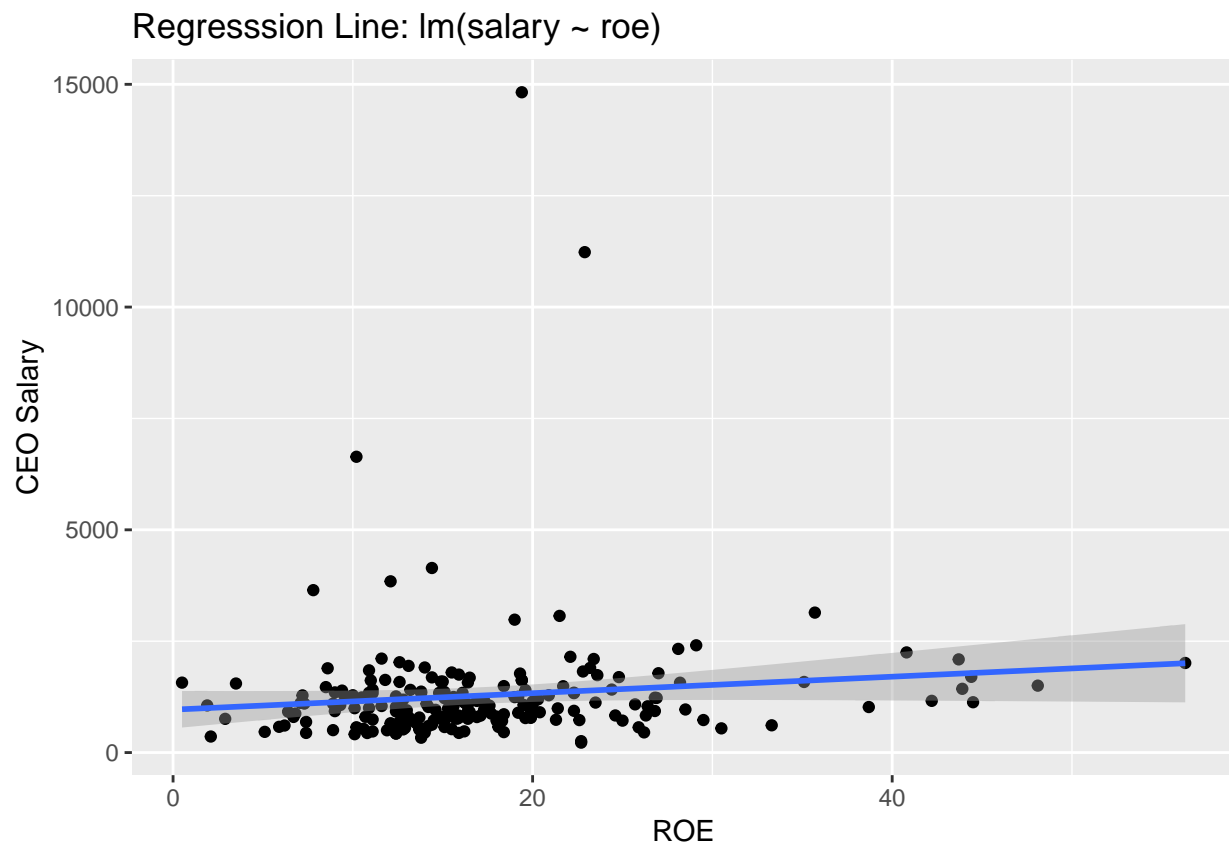
## Simple OLS Estimations

```
# Salary vs ROE
model1 <- lm(salary ~ roe, data = desc.ceosal1)
stargazer(model1, title = "Model Estimation: lm(salary ~ roe)", type = "text",
           style = "qje", out = "SR0/Model 1.doc")
```

```
##
## Model Estimation: lm(salary ~ roe)
## =====
##                                salary
## -----
## roe                            18.501*
##                               (11.123)
##
## Constant                       963.191***
##                               (213.240)
##
## N                               209
## R2                             0.013
```

```
## Adjusted R2                0.008
## Residual Std. Error        1,366.555 (df = 207)
## F Statistic                 2.767* (df = 1; 207)
## =====
## Notes:                      ***Significant at the 1 percent level.
##                             **Significant at the 5 percent level.
##                             *Significant at the 10 percent level.
```

```
(regline.model11 <- ggplot(desc.ceosal1, aes(x = roe, y = salary)) +
  geom_point() +
  labs(x = "ROE", y = "CEO Salary", title = "Regression Line: lm(salary ~ roe)") +
  geom_smooth(method = "lm"))
```



```
ggsave("SR0/Regression Line 1.png", regline.model11)
```

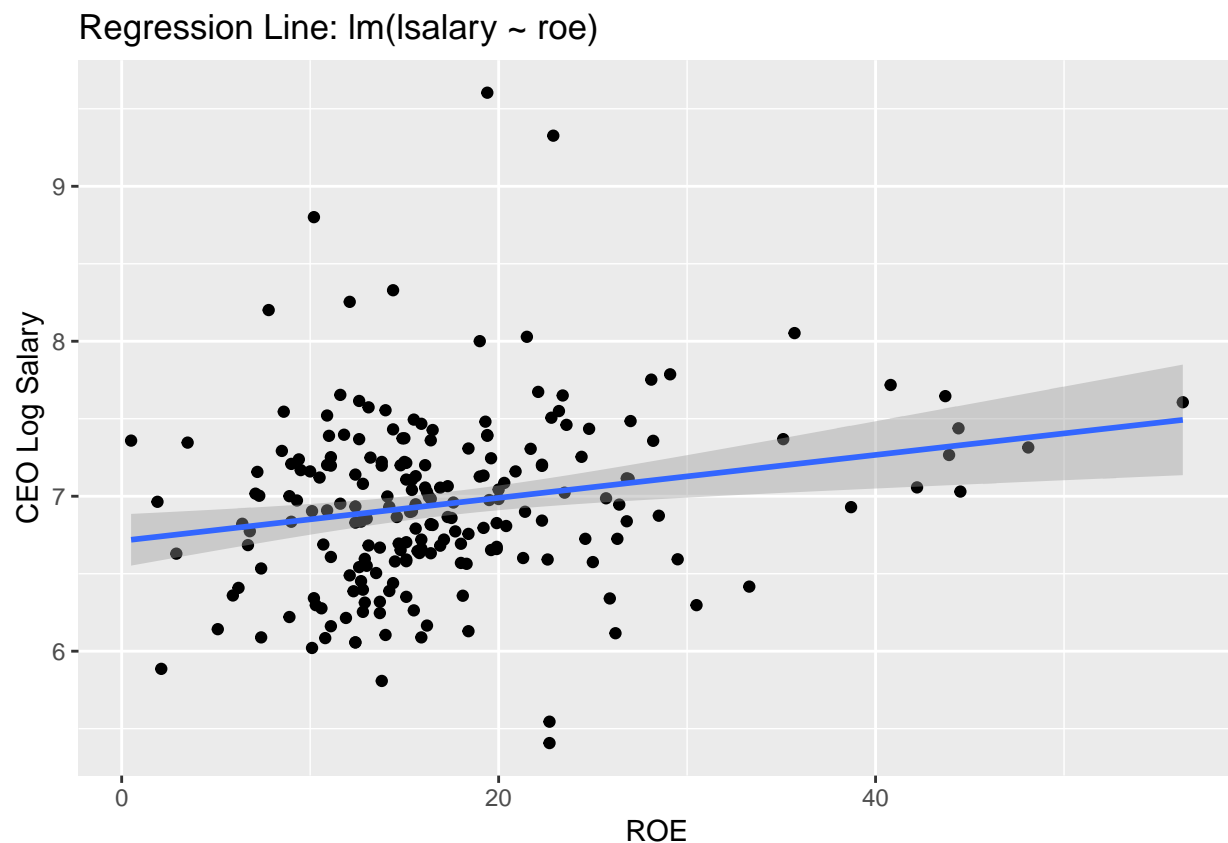
```
# Log Salary vs ROE
model2 <- lm(lsalary ~ roe, data = desc.ceosal1)
stargazer(model2, title = "Model Estimation: lm(lsalary ~ roe)", type = "text",
  style = "qje", out = "SR0/Model 2.doc")
```

```
##
## Model Estimation: lm(lsalary ~ roe)
## =====
##                                lsalary
## -----
```



```
## roe                                0.014***
##                                  (0.005)
##
## Constant                          6.712***
##                                  (0.087)
##
## N                                  209
## R2                                0.043
## Adjusted R2                       0.039
## Residual Std. Error               0.555 (df = 207)
## F Statistic                       9.408*** (df = 1; 207)
## =====
## Notes:                            ***Significant at the 1 percent level.
##                                **Significant at the 5 percent level.
##                                *Significant at the 10 percent level.
```

```
(regline.model2 <- ggplot(desc.ceosal1, aes(x = roe, y = lsalary)) +
  geom_point() +
  labs(x = "ROE", y = "CEO Log Salary", title = "Regression Line: lm(lsalary ~ roe)") +
  geom_smooth(method = "lm"))
```



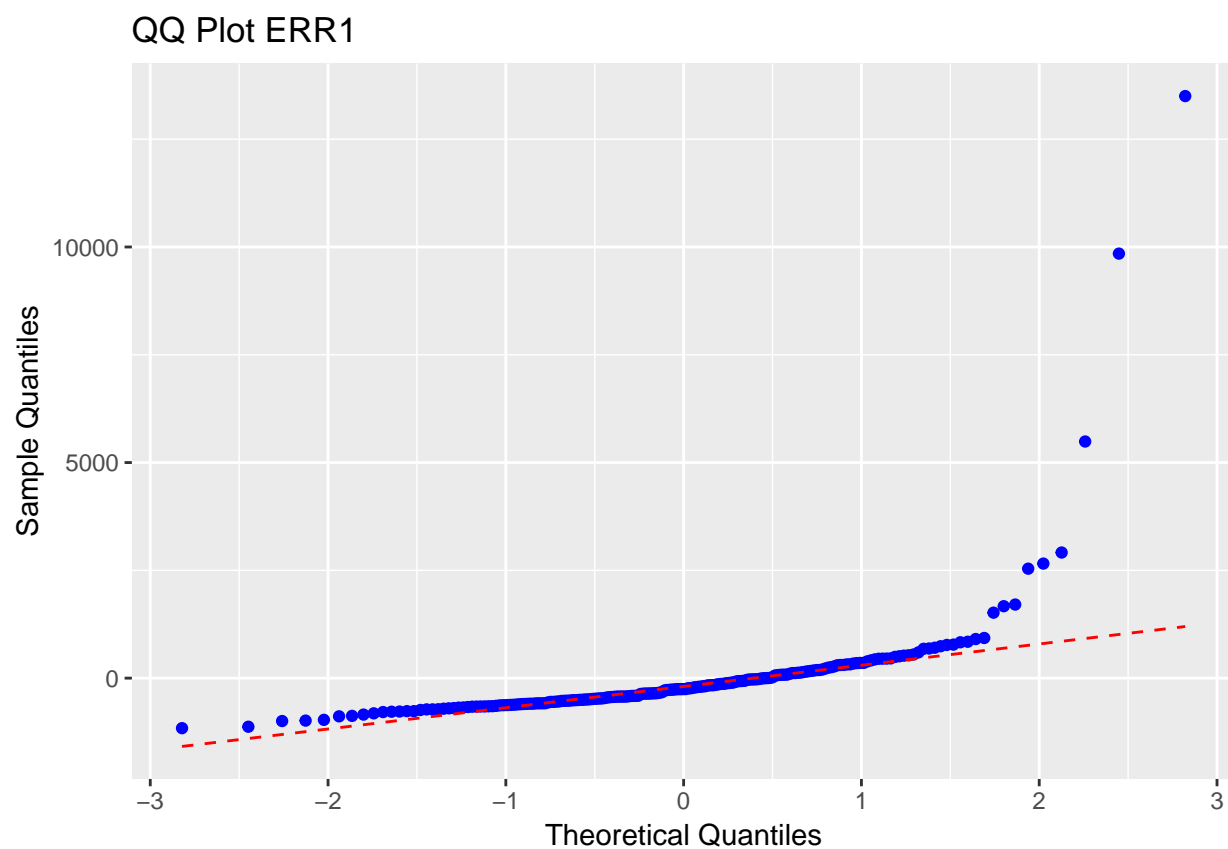
```
ggsave("SRO/Regression Line 2.png", regline.model2)
```

## Checking Normal Distribution of Error Terms

```
# Checking normal distribution of error term model 1
shapiro.test(residuals(model1))
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(model1)
## W = 0.41717, p-value < 2.2e-16
```

```
(error.model1 <- ggplot(model1, aes(sample = model1$residuals)) +
  stat_qq(col = "blue") +
  stat_qq_line(col = "red", lty = 2) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles", title = "QQ Plot ERR1"))
```



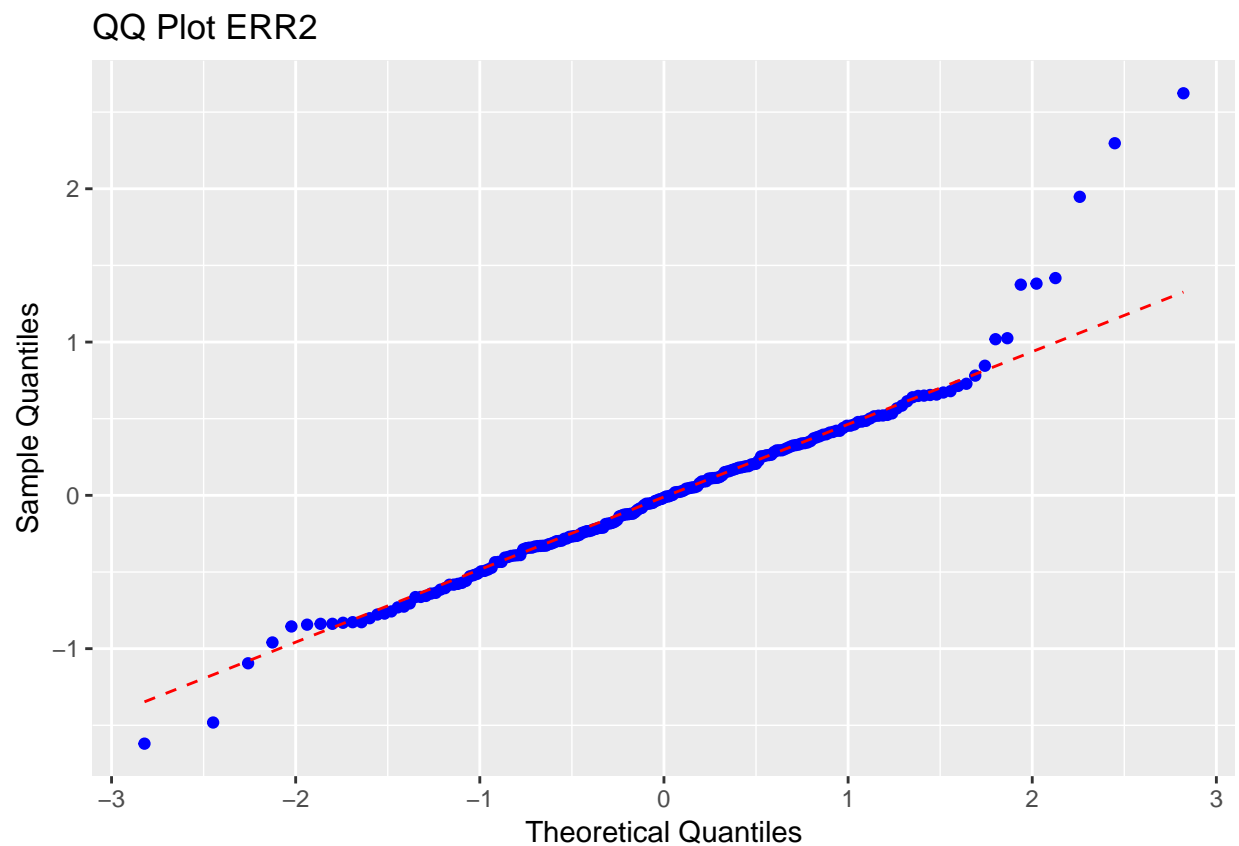
```
ggsave("SRO/QQ Plot ERR1.png", error.model1)
```

```
# Checking normal distribution of error term model 2
shapiro.test(residuals(model2))
```

```
##
## Shapiro-Wilk normality test
```

```
##
## data: residuals(model2)
## W = 0.94288, p-value = 2.464e-07
```

```
(error.model2 <- ggplot(model2, aes(sample = model2$residuals)) +
  stat_qq(col = "blue") +
  stat_qq_line(col = "red", lty = 2) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles", title = "QQ Plot ERR2"))
```



```
ggsave("SRO/QQ Plot ERR2.png", error.model2)
```

## Test of Heteroskedasticity

```
bptest(model1)
```

```
##
## studentized Breusch-Pagan test
##
## data: model1
## BP = 0.17205, df = 1, p-value = 0.6783
```

```
bptest(model2)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: model2  
## BP = 0.022814, df = 1, p-value = 0.8799
```

## References

- Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. 2018. *Rmarkdown: Dynamic Documents for R*. <https://CRAN.R-project.org/package=rmarkdown>.
- Hlavac, Marek. 2018. *Stargazer: Well-Formatted Regression and Summary Statistics Tables*. <https://CRAN.R-project.org/package=stargazer>.
- Hothorn, Torsten, Achim Zeileis, Richard W. Farebrother, and Clint Cummins. 2018. *Lmtest: Testing Linear Regression Models*. <https://CRAN.R-project.org/package=lmtest>.
- R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Shea, Justin M. 2018. *Wooldridge: 111 Data Sets from “Introductory Econometrics: A Modern Approach, 6e” by Jeffrey M. Wooldridge*. <https://CRAN.R-project.org/package=wooldridge>.
- Wickham, Hadley. 2017. *Tidyverse: Easily Install and Load the ‘Tidyverse’*. <https://CRAN.R-project.org/package=tidyverse>.
- Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, and Kara Woo. 2018. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2>.