lab 4

Zeren Li

9/18/2019

Roadmap

- Compute se, t-stat, and confidence interval
- Heteroskedasticity
- Regression Diagnostics
- Non-linearity
  - Logged transformation regression
  - Bivariate quadratic equation

Assumptions of OLS

Assumption 1: The Error Term has Conditional Mean of Zero

Assumption 2: Independently and Identically Distributed Data

Assumption 3: Large Outliers are Unlikely

The Sampling Distribution of the OLS Estimator

- As $\hat{\beta}_0$ and $\hat{\beta}_1$ are computed from a sample, estimators themselves are random variables with a probability distribution — the so-called sampling distribution of the estimators — which describes the values they could take on over different samples.
- Sampling distribution can be complicated when the sample size is small and generally changes with the number of observations, n
- When Sample is sufficiently large, by the central limit theorem the *joint* sampling distribution of the estimators is well approximated by the bivariate normal distribution

$$E(\hat{\beta}_0) = \beta_0 \text{ and } E(\hat{\beta}_1) = \beta_1,$$

Hypothesis Tests

- Testing Hypotheses regarding regression coefficients
- Confidence intervals for regression coefficients

A general t-statistic has the form

$$t = \frac{\text{estimated value} - \text{hypothesized value}}{\text{standard error of the estimator}}.$$

For testing the hypothesis $H_0 : \beta_1 = \beta_{1,0}$, we need to perform the following steps:

1. Compute the standard error of $\hat{\beta}_1$, $sigma(\hat{\beta}_1)$

$$\hat{\sigma}_{\hat{\beta}_1} = \sqrt{\frac{\frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \overline{Y})^2}{\sum_{i=1}^{n}(X_i - \overline{X})^2}}$$

2. Compute the t-statistic

$$\frac{\hat{\beta}_1 - \beta_{1,0}}{\sigma(\hat{\beta}_1)} \sim t_{n-k-1}.$$

where k is the number of paramter, n is the number of observation.

For bivariate regression, $n - k - 1 = n - 2$.

3. Given a two sided alternative ($H_1 : \beta_1 \neq \beta_{1,0}$) we reject at the 5% level if $|t^{act}| > 1.96$ or, equivalently, if the p-value is less than 0.05.

Recall the definition of the p-value:

$$\begin{aligned} p\text{-value} &= \Pr_{H_0}\left[\left|\frac{\hat{\beta}_1 - \beta_{1,0}}{SE(\hat{\beta}_1)}\right| > \left|\frac{\hat{\beta}_1^{act} - \beta_{1,0}}{SE(\hat{\beta}_1)}\right|\right] \\ &= \Pr_{H_0}(|t| > |t^{act}|) \\ &\approx 2 \cdot \Phi(-|t^{act}|) \end{aligned}$$

The last transformation is due to the normal approximation for large samples.

Example: Returns to Sales Performance

Compute T-statistic:

```r
# y as dependent variable
y <- ceo$salary

# x as independent variable
x <- ceo$sales
n <- length(ceo$salary)

# beta1
beta1 = sum((y - mean(y)) * (x - mean(x))) / sum (((x - mean(x))^2))
beta1
```

```
## [1] 0.03669374
```

```r
# beta0
beta0 <- mean(y) - beta1 * mean(x)
beta0
```

```
## [1] 736.3552
```

```r
# predicted Y
y_hat <- beta1 * x + beta0
head(y_hat)
```

```
## [1]  963.8564  746.7395  742.5565  776.7183  749.2347 1433.5362
```

```r
# predicted u
u_hat = y - y_hat

# signma beta 1
denom = 1/(n-2) * sum(u_hat^2)
num =  sum( (x- mean(x))^2 )

sigma_beta1 = sqrt(denom/num)

# t statistic
t_test = (beta1 - 0)/sigma_beta1
t_test
```

```
## [1] 5.438338
```

Compute P-value:

```r
# pt() is the distribution function of t distribution
2*pt(-abs(t_test), df = n-2)
```

```
## [1] 1.788196e-07
```

Double-check with build-in function:

```r
# estimate the model
m1 <- lm(salary ~ sales, data = ceo)

# summary of regression
sum = summary(m1)
sum
```

```
##
## Call:
## lm(formula = salary ~ sales, data = ceo)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -735.4 -340.2 -125.7  236.5 4474.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.364e+02  4.738e+01  15.540  < 2e-16 ***
## sales       3.669e-02  6.747e-03   5.438 1.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 545 on 175 degrees of freedom
## Multiple R-squared:  0.1446, Adjusted R-squared:  0.1397
## F-statistic: 29.58 on 1 and 175 DF,  p-value: 1.788e-07
```

```r
# Estimate, SE, t value, and P value of coefficients
options(xtable.comment = FALSE)
xtable(sum$coefficients)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	736.36	47.38	15.54	0.00
sales	0.04	0.01	5.44	0.00

Confidence Intervals

The interval has a probability of 95% to contain the true value of β_i. So in 95% of all samples that could be drawn, the confidence interval will cover the true value of β_i.

$$CI_{\beta 1} = (\bar{\beta}_1 - t^* * \hat{\sigma}_{\beta_1}, \bar{X} + t^* * \hat{\sigma}_{\beta_1})$$

```
dof = n-2
critical_t <- qt(0.05/2, dof)

beta1 - critical_t*sigma_beta1
```

```
## [1] 0.05001016
```

```
beta1 + critical_t*sigma_beta1
```

```
## [1] 0.02337731
```

```
# coefficient
confint(m1)
```

```
##                   2.5 %       97.5 %
## (Intercept) 642.83695765 829.87346434
## sales         0.02337731   0.05001016
```

Heteroskedasticity and Homoskedasticity

All inference made in the previous discussion relies on the assumption that the error variance does not vary as regressor values change. But this will often not be the case in empirical applications.

- The error term of our regression model is homoskedastic if the variance of the conditional distribution of u_i given X_i, $Var(u_i|X_i = x)$, is constant *for all* observations in our sample:

$$\text{Var}(u_i|X_i = x) = \sigma^2 \ \forall \ i = 1, \dots, n.$$

- If instead there is dependence of the conditional variance of u_i on X_i, the error term is said to be heteroskedastic. We then write

$$\text{Var}(u_i|X_i = x) = \sigma_i^2 \ \forall \ i = 1, \dots, n.$$

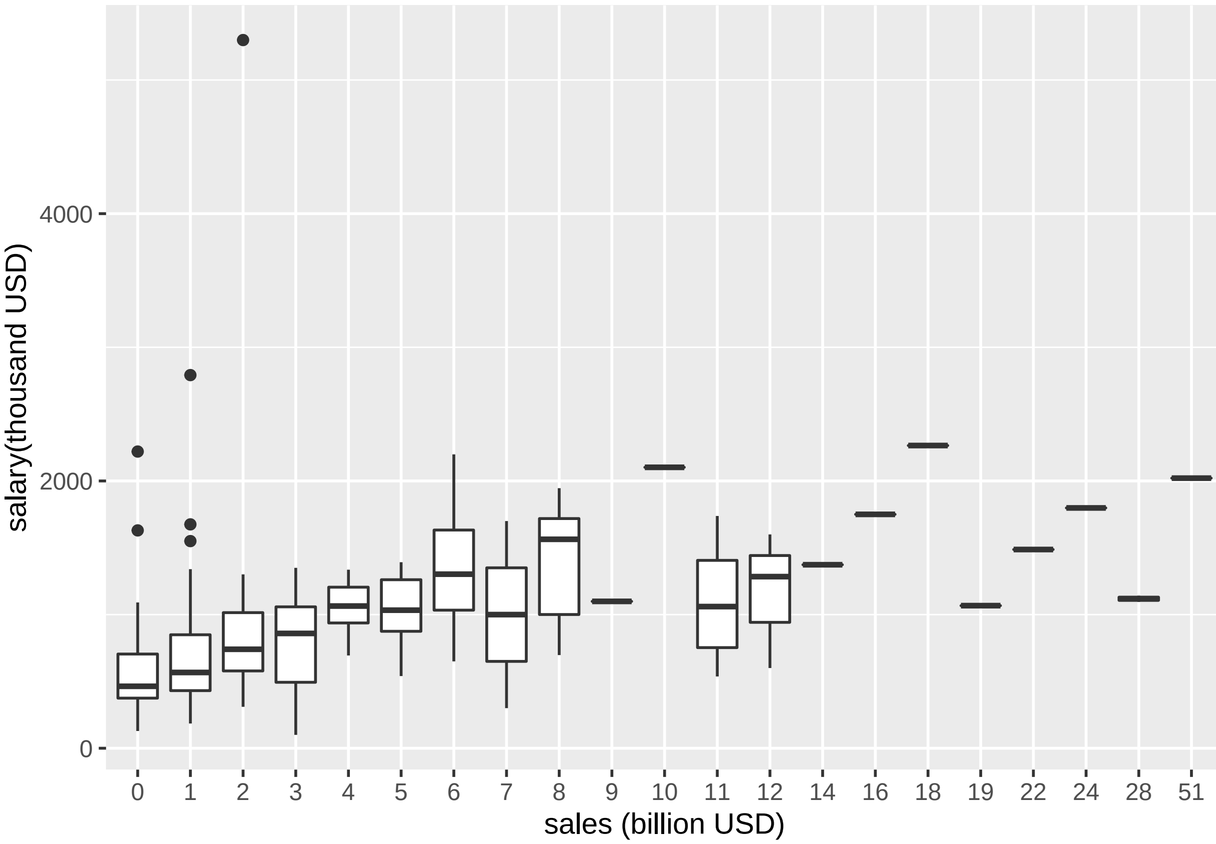
- Homoskedasticity is a *special case* of heteroskedasticity.

A better understanding of heteroskedasticity

$$salary_i = \beta_0 + \beta_1 \cdot sales_i + u_i.$$

- On average, CEO with higher sales earn more than their peers with lower sales -> an upward sloping regression line.
- It seems plausible that earnings of CEO with lower sales have a higher dispersion than those of CEO with higher sales.
- Some other factors matter for salary (relations with the board, Charisma, control of debt, etc.)

```
ceo %>%
  mutate(sales_scale = round(sales/1000) %>% as.factor()) %>%
# plot observations and add the regression line
  ggplot(.) +
  geom_boxplot( aes(x = sales_scale , y = salary )) +
  xlab("sales (billion USD)") +
  ylab("salary(thousand USD)")
```



Residual Analaysis

1. Residuals vs Fitted: shows if residuals have non-linear patterns.

The first plots the residuals versus the fitted values.

We are looking to see whether the residuals are spread uniformly across the line $y = 0$. If there is a U-shape, then that is evidence that there may be a variable "lurking" that we have not taken into account. It could be a variable that is related to the data that we did not collect, or it could be that our model should include a quadratic term.

2. Normal Q-Q: shows if residuals are normally distributed.

Ideally, the points would fall more or less along the line given in the plot. It takes some experience to know what is a reasonable departure from the line and what would indicate a problem.

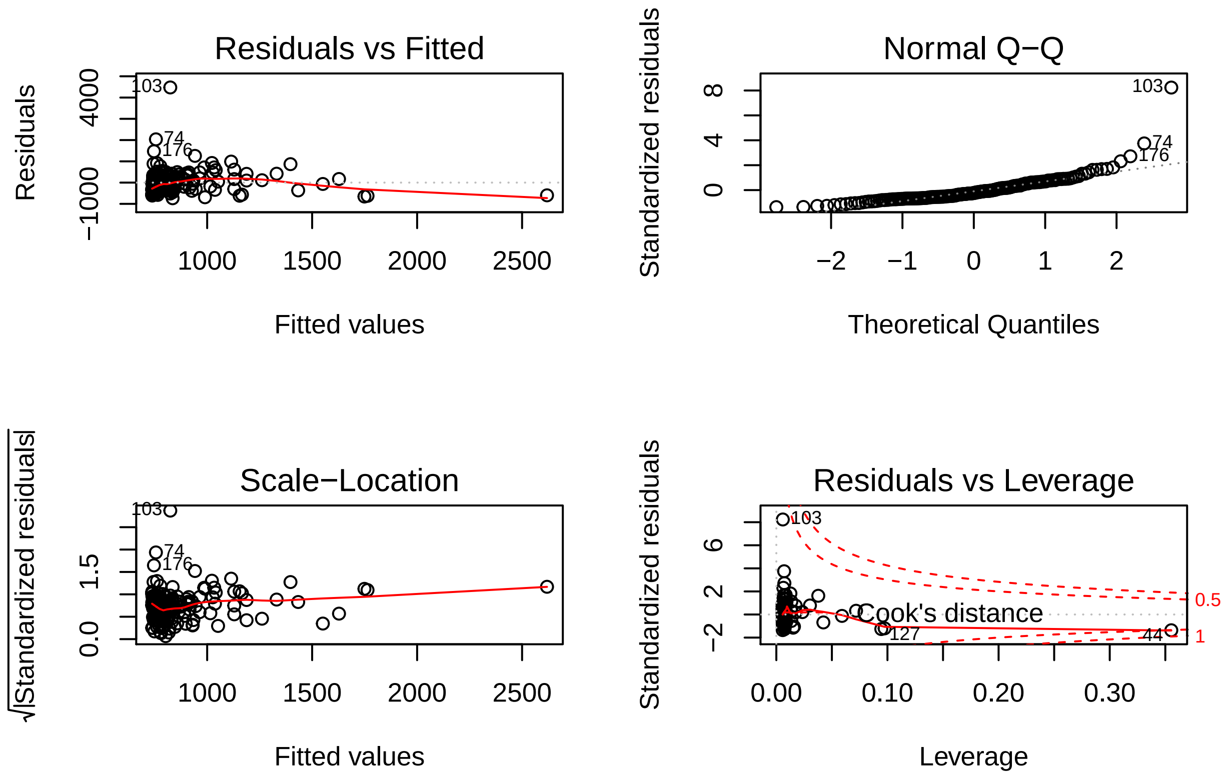
3. Scale-Location: shows if residuals are spread equally along with the ranges of predictors.

This is a plot that helps us to see whether the variance is constant across the fitted values. Many times, the variance will increase with the fitted value, in which case we would see an upward trend in this plot. We are looking to see that the line is more or less flat.

4. Residuals vs Leverage: helps us to find outliers.

Outliers are points that fit the model worse than the rest of the data. Outliers with x-coordinates in the middle of the data tend to have less of an impact on the final model than outliers toward the edge of the x-coordinates. Data that falls outside the red dashed lines are high-leverage outliers, meaning that they (may) have a large effect on the final model. You should consider removing the data and re-running in order to see how big the effect is. Or you could use robust methods (We may discuss this later this semster).

```r
par(mfrow=c(2,2))
plot(m1, ask=F)
```



Log Transformation

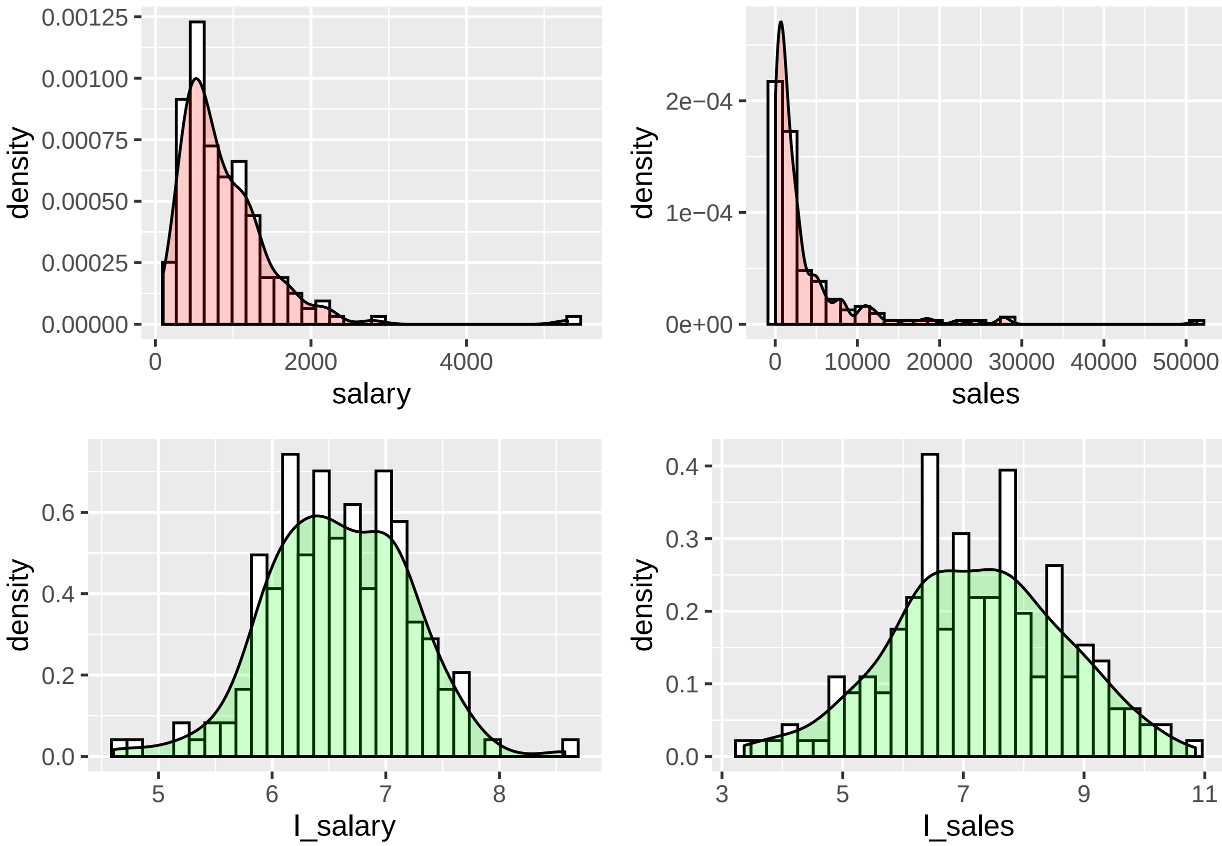
level-level, $y = \beta_1 x + \beta_0$, a 1 unit change in x results in a β_1 unit change in y

level-log, $y = \beta_1 ln(x) + \beta_0$, a 1% change in x results in a $\beta_1/100$ unit change in y

log-level, $ln(y) = \beta_1 x + \beta_0$, a 1 unit change in x results in a $\beta_1 * 100$ unit change in y

log-log, $ln(y) = \beta_1 ln(x) + \beta_0$, a 1% change in x results in a β_1% change in y

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



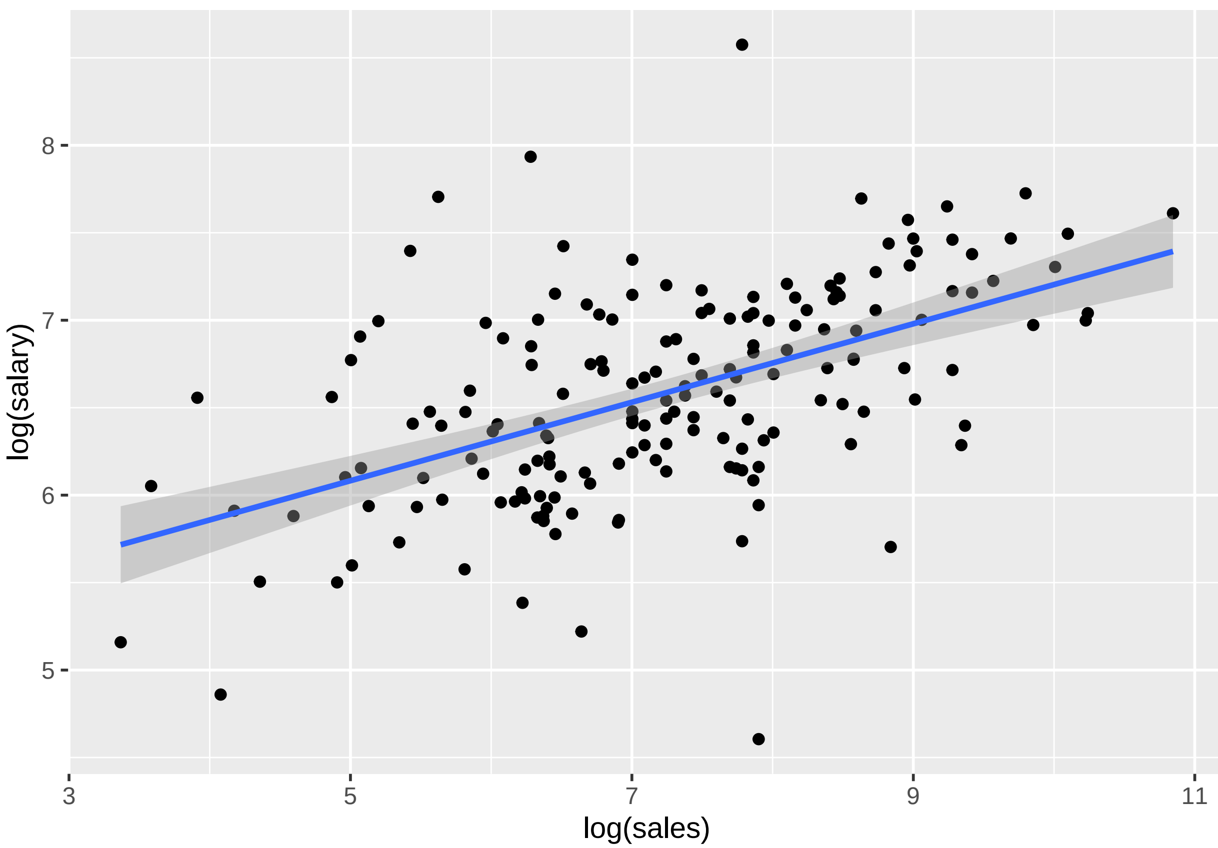

```r
# baseline model
m1 <- lm(salary ~ sales, ceo_logged)

# model with logged independent variable(s)
m2 <- lm(salary ~ l_sales, ceo_logged)

# model with logged depdent variable
m3 <- lm(l_salary ~ sales, ceo_logged)

# model with logged depdent variable
# and logged independent variable(s)
m4 <- lm(l_salary ~ l_sales, ceo_logged)
```

```r
# export regression table
stargazer(m1,m2,m3,m4, header = F, type = "text")
```

```
##
## ========================================================================
##                                           Dependent variable:
##                                   --------------------------------------
##                                        salary              l_salary
##                                     (1)        (2)        (3)        (4)
## ------------------------------------------------------------------------
## sales                             0.037***              0.00004***
##                                   (0.007)               (0.00001)
##
## l_sales                                     177.149***             0.224***
##                                              (27.976)               (0.027)
```

```
##
## Constant                         736.355*** -415.105**  6.439***  4.961***
##                                   (47.384)  (206.204)   (0.048)    (0.200)
##
## ----------------------------------------------------------------------------
## Observations                        177         177        177        177
## R2                                 0.145       0.186      0.168      0.281
## Adjusted R2                        0.140       0.182      0.163      0.277
## Residual Std. Error (df = 175)  545.009     531.513      0.554      0.515
## F Statistic (df = 1; 175)      29.576***  40.096***  35.327***  68.345***
## ============================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

```
# plot observations and add the regression line
ggplot(ceo, aes(x = log(sales), y = log(salary) )) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x )
```



Quadratic Regression

$$salary_i = \beta_0 + \beta_1 \cdot sales_i + \beta_2 \cdot sales_i^2 + u_i.$$

```
q_ceo <- ceo %>%
 mutate(
   # transform to billion USD
   sales = sales/1000,
   sales_2 = sales^2)
```

```r
# fit a quadratic regression
q_m <- lm(salary ~ sales_2 + sales, q_ceo  )

# summary of regression
summary(q_m)
```

```
##
## Call:
## lm(formula = salary ~ sales_2 + sales, data = q_ceo)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -795.7 -305.1 -103.2  234.9 4467.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 674.1354    52.9038  12.743  < 2e-16 ***
## sales_2      -0.9577     0.3829  -2.501   0.0133 *
## sales        67.7043    14.0697   4.812 3.23e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 537 on 174 degrees of freedom
## Multiple R-squared:  0.1743, Adjusted R-squared:  0.1648
## F-statistic: 18.36 on 2 and 174 DF,  p-value: 5.831e-08
```

```r
# coefficients
coef(q_m)
```

```
## (Intercept)     sales_2       sales
## 674.1354494  -0.9576503  67.7042527
```

```r
# predicted value of y
# by hand
y_predict_byhand <- coef(q_m)[1] + coef(q_m)[2] * q_ceo$sales_2 + coef(q_m)[3] * q_ceo$sales

# use predict()
y_predict <- predict(q_m)

# double check two outputs
data.frame(y_predict_byhand, y_predict) %>% head()
```

```
##   y_predict_byhand y_predict
## 1        1057.0897 1057.0897
## 2         693.2191  693.2191
## 3         685.5501  685.5501
## 4         747.4514  747.4514
## 5         697.7817  697.7817
## 6        1614.8045 1614.8045
```

$$salary_i = 674.14 + 67.70 \cdot sales_i - 0.96 \cdot sales_i^2 + u_i.$$

```r
ggplot(q_ceo, aes(x = sales, y = salary)) +
  geom_point() +
```

```r
stat_smooth(method = "lm", formula = y ~ x + I(x^2), size = 1) +
xlab("sales (billion USD)")
```