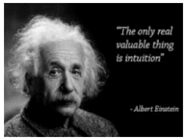
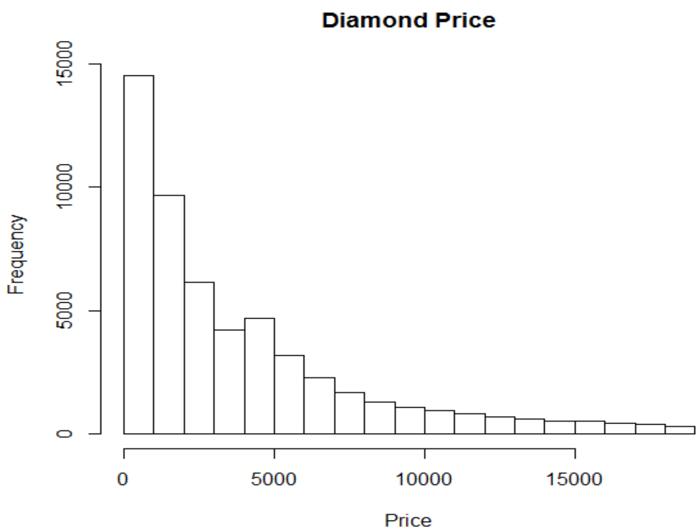
# Transformation #4:
## Log Models

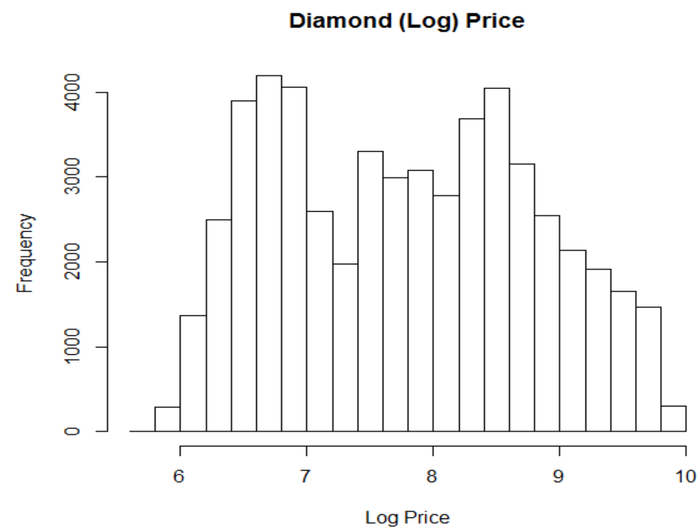





# Log Models: Intuition

- Log transformations are very useful when the data is skewed because the logged data becomes more normally distributed
- You can inspect the **QQ-Plot** and **Histogram** of the variable to evaluate normality
- You **cannot log** variables that have **negative values** (there are no logs for negative values)
- Useful with **truncated data** (e.g., count data – truncated at 0)
- Useful to model **elasticity (%) effects**



# Log Models: Interpreting Effects

A model $y = \beta_0 + \beta_1 x + \varepsilon$ has 4 possible linear-log models (more than one x variable can be logged):

| Dependent Variable | Independent Variable | |
|---|---|---|
| | x | Log(x) |
| y | **Linear Model**<br>$y = \beta_0 + \beta_1 x + \varepsilon$<br>x ↑ 1 unit → y ↑ $\beta_1$ units | **Linear-Log Model**<br>$y = \beta_0 + \beta_1 Log(x) + \varepsilon$<br>1% (1/100) ↑ in x → y ↑ $\beta_1$/100 units |
| Log(y) | **Log-Linear Model**<br>$Log(y) = \beta_0 + \beta_1 X + \varepsilon$<br>x ↑ 1 unit → y ↑ 100*$\beta_1$% (i.e., $\beta_1$ fraction) | **Log-Log (Elasticity) Model**<br>$Log(y) = \beta_0 + \beta_1 \mathrm{Log}(x) + \varepsilon$<br>1% ↑ x → y ↑ $\beta_1$% |

# R Tips

`lm.log.linear.fit=lm(log(y)~x1+x2+etc.,`
`data=dataName)` → Log-Linear model

`lm.linear.log.fit=lm(y~log(x1)+x2+etc.,`
`data=dataName)` → Linear-Log model
with 1 predictor `x1` logged

`lm.log.log.fit=lm(log(y)~log(x1)+log(x2)+etc.,`
`data=dataName)` → Log-Log (Elasticity)
model with all predictors logged

KOGOD SCHOOL
of
BUSINESS