# Computational Statistics 10.3: Non-linear functional forms




# Non-linear functional forms

## Overview

This lesson introduces a variety of non-linear functional forms specifications

## Objectives

After completing this module, students should be able to:

1. Estimate and interpret regressions using logarithmic transformations
2. Estimate and interpret regressions using quadratic and higher order terms

## Readings

Schumacker, Chapter 17.

# Non-linear vs linear relations

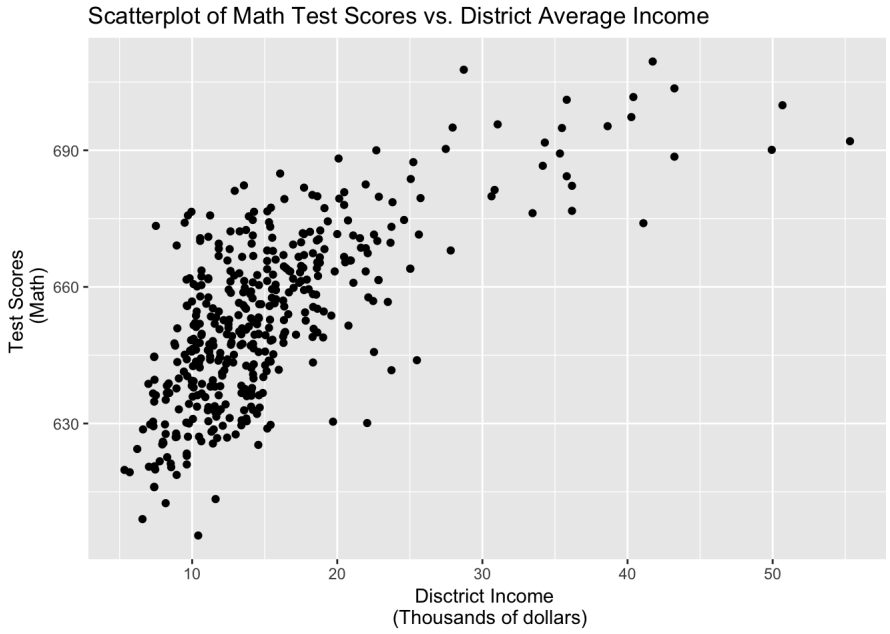
/

- Previously we have assumed that the regression function to be linear. That is, the effect of the independent variable ($x$) on the dependent variable ($y$) is constant over all the range of $x$.

- Sometimes the effect of $x$ on $y$ depends on the values of $x$. For example, we know that years of education has a positive effect on earnings. Also, we know that after you have a master or PhD degree, staying more years in school has no effect on earnings - in fact, some may argue that the effect may turn negative for very high levels of education as staying for a long time in school reduces the amount of accumulated job market experience -.

- The following graph was generated using the `CASchools` dataset from lesson 9.3 and it shows the relation between between the average income of a school district (measured in thousands of dollars) and the average math test scores.

```
# Loading data
suppressMessages(library(AER)) #Quietly loading AER Package
library(ggplot2)
data("CASchools") # Loading CASchools dataset

# Making scatterplot of income vs math
graph1 <- ggplot(data = CASchools, aes(x = income, y = math)) + geom_point()
graph1 <- graph1 + labs(title = "Scatterplot of Math Test Scores vs. District Average Income",
                        x = "Disctrict Income \n (Thousands of dollars)",
                        y = "Test Scores \n (Math)")
```




- The previous graph is evidence that sometimes we cannot guarantee that every increase in $x$ will cause the same $\beta_1$ effect on $y$ in all instances make us consider in what ways can we introduce non-linearity to a regression model.
    - The effect of income seems to cause a larger impact on test scores for lower values of income compared to when income is higher, in fact after about 30 thousands the effect seems to be zero.
    - This makes sense as kids that lives in extreme or severe poverty will benefit more from an increase in income than kids in middle or/and high income families.

# Polynomial functional forms

- A common non-linear functional form that can be easily introduced to a regression model is a polynomial.

**Quadratic Terms**

- For example, we can represent an **quadratic** relation in a regression model by adding $x^2$ to the regression:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

- The estimated parameters for $\beta_0, \beta_1$ and $\beta_2$ determine the shape of the quadratic relation, including whether it is convex or concave.

- Note that by simply raising the variable $x$ to the power of 2, we are allowing for $x$ to be related with $y$ in a non-linear way. Now, when you increase $x$ by one unit, the effect on $y$ is non-constant. Say that we want to know that's the effect on $y$ of increasing $x$ by one unit starting a some particular value $\bar{x}$,

$$\begin{aligned} y(x = \bar{x}) &= \beta_0 + \beta_1 \bar{x} + \beta_2 \bar{x}^2 + \epsilon && \text{(Value of } y \text{ at } x = \bar{x}) \\ y(x = \bar{x} + 1) &= \beta_0 + \beta_1 (\bar{x} + 1) + \beta_2 (\bar{x} + 1)^2 + \epsilon && \text{(Value of } y \text{ at } x = \bar{x} + 1) \end{aligned}$$

Now let's compute the difference between the values of $y$ ($\Delta y$) to have a measure of the effect,

$$\begin{aligned} \Delta(y) &= y(x = \bar{x} + 1) - y(x = \bar{x}) \\ &= \left(\beta_0 + \beta_1 (\bar{x} + 1) + \beta_2 (\bar{x} + 1)^2 + \epsilon\right) - \left(\beta_0 + \beta_1 \bar{x} + \beta_2 \bar{x}^2 + \epsilon\right) \\ &= (\beta_0 - \beta_0) + \beta_1 \left(\bar{x} + 1 - \bar{x}\right) + \beta_2 \left((\bar{x} + 1)^2 - \bar{x}\right) + (\epsilon - \epsilon) && \text{(Rearranging terms)} \\ &= \beta_1 + \beta_2 \left((\bar{x}^2 + 2\bar{x} + 1 - \bar{x}\right) && \text{(Expanding } (\bar{x} + 1)^2 = \bar{x}^2 + 2\bar{x} + 1) \\ &= \beta_1 + \beta_2 (\bar{x}^2 + \bar{x} + 1) \end{aligned}$$

- In the linear model the change in $y$ caused by an increase in one unit of $x$ was simply $\beta_1$, now it depends on the value of $x$ and the parameter $\beta_2$. That's what makes the relation non-linear

**Higher order Terms**

- To add higher order terms we can simply add the polynomial transformation for the higher order of $x$. For example, for the $kth$ degree polynomial the regression model will look like this:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots + \beta_k x^k + \epsilon$$

- The more polynomial terms the more complex and harder to interpret is the relation between $x$ and $y$. In practice is rare to see a regression model with a polynomial of degree $k \geq 4$.

## Simulation with quadratic term

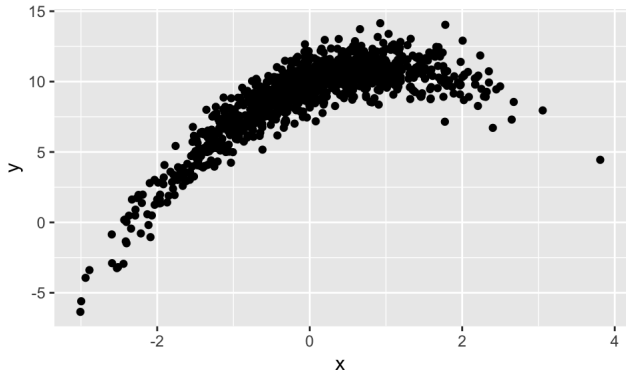
Let's use some simulated data before we turn back to the income and test scores example. In this example we are going to simulate a non-linear relation between $x$ and $y$ and try to estimate the parameters of the model. The goal is to compare how well linear regression represents the relation between $x$ and $y$ vs. a non-linear model.

- First, let's simulate some data

```
set.seed(1)
n <- 1000 # Number of observations
x <- rnorm(n) # Generating x
y <- 10 + 2*x - x^2 + rnorm(n) # Generating y using a quadratic function
df <- data.frame(y=y,x=x) # creating data frame
```

Now let's examine the data by making a scatterplot:

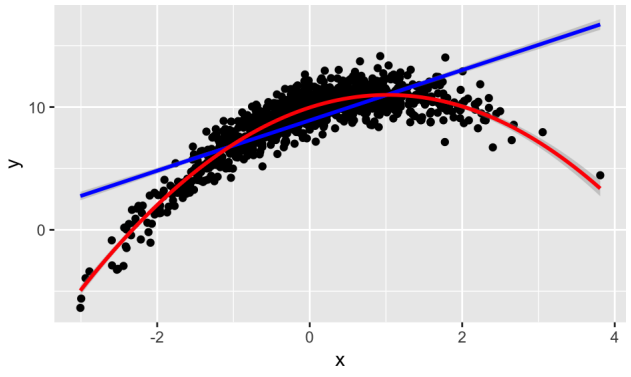
```
graph2 <- ggplot(data=df,aes(x=x,y=y)) + geom_point()
graph2
```




Clearly we have a **curvilinear relationship**.

Let's see what happens when we fit a linear regression vs a regression with a quadratic term:

```
# Adding linear regression to the graph
graph2 <- graph2 + geom_smooth(method = lm, formula = (y ~ x), color = "blue")
# Adding quadratic regression to the graph
graph2 <- graph2 + geom_smooth(method = lm, formula = (y ~ x + I(x^2)), color = "red")
graph2
```




- See how the quadratic model (red line) fits the data better than the linear model (blue line).
- Not only we get a better fit to the data with the quadratic model, but the linear model is incorrectly estimating the effect of $x$ on $y$ for very low and high values of $x$. Only on average values of $x$ is the linear regression a reasonable approximation.

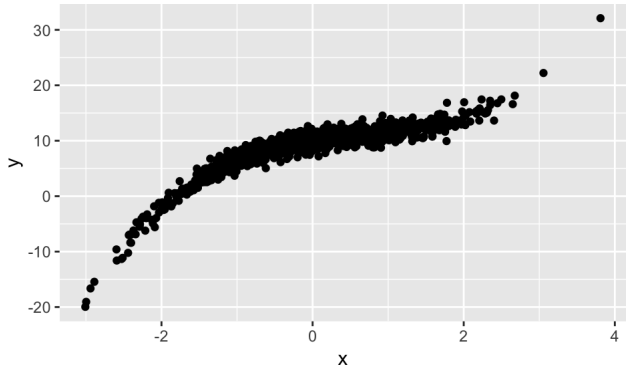
## Simulation with cubic term

Let's see how a cubic relation looks like:

```
set.seed(1)
n <- 1000 # Number of observations
x <- rnorm(n) # Generating x
y <- 10 + 2*x - x^2 + 0.5*x^3 + rnorm(n) # Generating y using a cubicfunction
df <- data.frame(y=y,x=x) # creating data frame
```

Now let's examine the data:

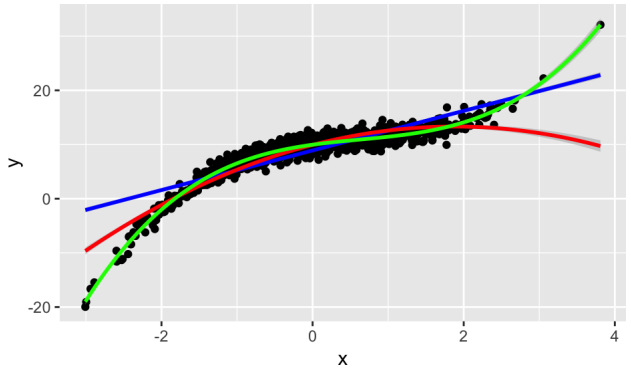
```
graph2 <- ggplot(data=df,aes(x=x,y=y)) + geom_point()
graph2
```




Again, we find a **curvilinear relationship**, but this time the data seems to have two distinct parts: first the effect of $x$ on $y$ is **diminishing** and then **increasing** – first, the slope is positive with decreasing returns to $x$, then the slope is still positive but increasing –

Let's see what happens when we fit a linear regression vs a regression with a quadratic and cubic models:

```
graph2 <- graph2 + geom_smooth(method = lm, formula = (y ~ x), color = "blue")
graph2 <- graph2 + geom_smooth(method = lm, formula = (y ~ x + I(x^2)), color = "red")
graph2 <- graph2 + geom_smooth(method = lm, formula = (y ~ x + I(x^2) + I(x^3)), color = "green")
graph2
```




- From the previous graph is easy to see that the cubic regression (green line) is the best fit to the data. The linear model is again a bad estimation of the effect of $x$ on $y$ for very high and low values of $x$, and only a good approximation for average values of $x$. The quadratic model is a better fit than the linear models for low values of $x$, but is worse for high values of $x$. Note how incorrectly specifying the number of higher order terms can lead **mispecification bias**.
- **Mispecification bias** using a linear model to estimate the effect of $x$ on $y$ is an issue when you are really interested in estimating the effect of $x$ on $y$ for very low and/or high values of $x$, the linear model generally is a good approximation of the effect $x$ on $y$ for average values of $x$. Then, we should be cautious when using a linear regression model at the extreme values of $x$.

## Specifying higher order terms in the `lm` command

We can add quadratic and higher order terms into a regression model in at least three different ways:

- **Manually**: Simply create a new variable using $x^2$, something like this `xsq <- x^2`. Then you can add `xsq` to the `lm` command like any other variable, e.g. `lm(y ~ x + xsq)`. This is also true for higher order polynomials, if you want to add a cubic term ($x^3$) you can simply save the variable `xcu <- x^3` and add it to the regression, e.g. `lm(y ~ x + xsq + xcu)`

- **`I()` command**: Instead of creating a variable everytime time you need to add a higher order term, we can use the `I()` command. This command allows you to do a transformation to a variable before running a regression. Then, you will add `I(x^2)` to the `lm` command like this `lm(y ~ x + I(x^2))`, which means: first transform `x` to `x^2` and then run the regression.

- **`poly()` command**: This functions adds all the terms of a polynomial of degree `k` for a specific variable. For example, `poly(x,2)` will generate the following polynomial: `x + x^2`, which can be added to `lm` directly, e.g. `lm(y ~ poly(x,2))` .This is can save you some time when you are estimating regressions with high degrees terms (I consider high after 3 or 4).

## Estimating and interpreting higher order terms

When reading the output of a regression with higher order terms we have to answer the following questions:

**Is the relation between $x$ and $y$ non-linear?**

- If the coefficient on the $x^k$ term is not significant - assuming the regression does not suffer from multicollinearity or OVB -, that means that the $k$ degree polynomial transformation of $x$ has effect on $y$. And the relation is non-linear.
- If all the higher order polynomial terms are statistically equal to zero, then we can't reject the hypothesis that the relation is linear.

**Interpreting the sign of a higher order term**

- A polynomial of degree $k$ (with $k$ higher order terms), have $k-1$ regions in which the slope is **diminishing** or **increasing**. For example, a quadratic polynomial ($k=2$) has only ($k-1=2-1=1$) region, so the effect will be diminishing or increasing in all the domain of the

/

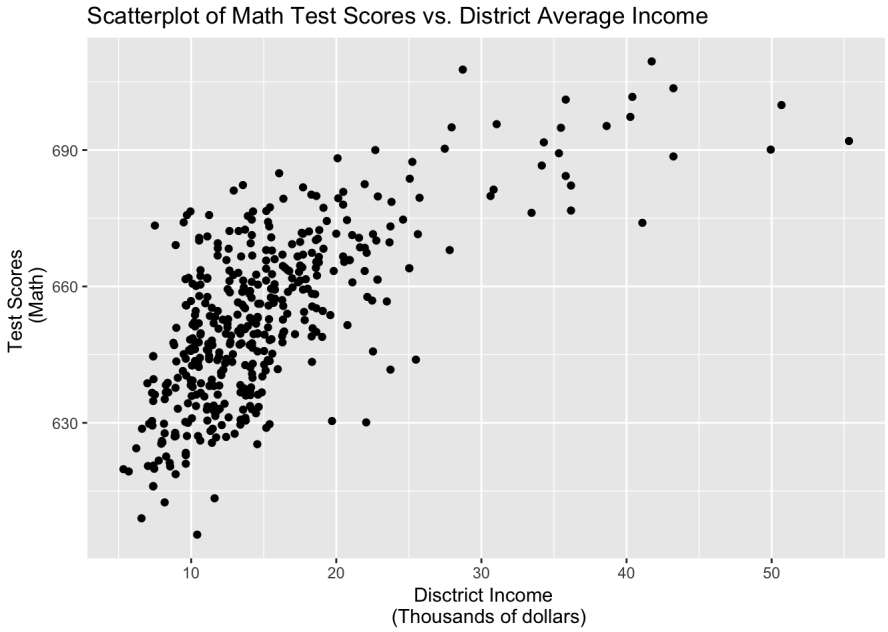
function. On the other hand, a cubic polynomial ($k = 3$) has ($k - 1 = 3 - 1 = 2$) regions, so the effect of $x$ on $y$ can be (1) first diminishing and then increasing, (2) first increasing and then diminishing, (3) always increasing, or, (4) always decreasing.

- The sign of the coefficient of the $x^k$ term determines if the effect of $x$ on $y$ is **diminishing** or **increasing** for the $k - 1$ part of the function.
    - If the parameter for the $k$ coefficient is positive, the effect of $x$ on $y$ is increasing in the $k - 1$ part of the function.
    - If the parameter for the $k$ coefficient is negative, the effect of $x$ on $y$ is diminishing in the $k - 1$ part of the function.
- For example, a quadratic regression (k=2) has only $k - 1 = 2 - 1 = 1$ region where the effect of $x$ on $y$ is non-linear
    - If the parameter of the quadratic term is positive, it means that the effect is globally increasing. In this case the function is **convex**
    - If the parameter is negative, it means that the effect is globally diminishing. In this case the function is **concave**
    - In the first simulation, the quadratic term was negative, and we can see how the value of the slope is diminishing in for all values of $x$ , making the relation **concave**.
- Second example, consider a cubic regression (k=3), then we have $k - 2 = 3 - 1 = 2$ regions where the effect of $x$ on $y$ is non-linear:
    - If the parameter of the quadratic term is positive, it means that the effect is increasing in the $k - 1 = 2 - 1 = 1$ part of the function (at relative low values of x).
    - If the parameter of the quadratic term is negative, it means that the effect is diminishing in the $k - 1 = 2 - 1 = 1$ part of the function (at relative low values of x).
    - If the parameter of the cubic term is positive, it means that the effect is increasing in the $k - 1 = 3 - 1 = 2$ part of the function (at relative high values of x).
    - If the parameter of the cubic term is negative, it means that the effect is diminishing in the $k - 1 = 3 - 1 = 2$ part of the function (at relative high values of x).
    - In the second simulation, the quadratic term was negative and the cubic term is positive, and we can see how the value of the slope is initially diminishing (for low values of $x$) and the increasing (for high values of $x$).
- You can see how the interpretation can get very complex if $k \geq 4$.

## Real world example: Fitting a polynomial (1/2)

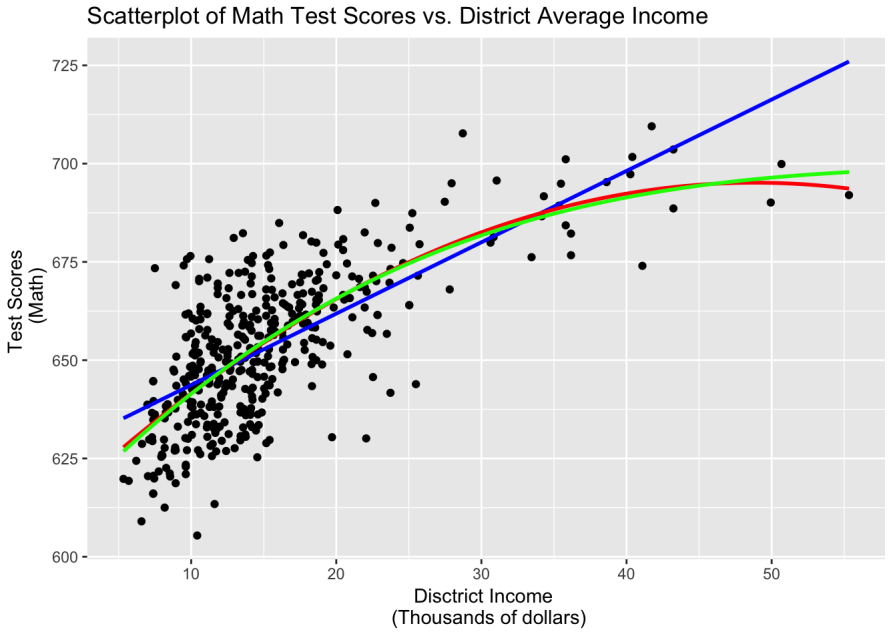
Let's illustrate the use of higher order terms using the `CASchools` dataset. Let's take a look at the data again:

```
graph1
```




- Because the effect of $x$ on $y$ seems to be **diminishing** for all the range of $x$, it looks like a quadratic regression is a better fit than a linear function. To be sure, we are going to estimate three regressions and compare the results. Let's take a look at the graph and then we'll interpret the regressions:

```
# Adding linear model in blue
graph1 <- graph1 + geom_smooth(method = lm, formula = (y ~ x), color = "blue", se = FALSE)
# Adding quadratic model in red
graph1 <- graph1 + geom_smooth(method = lm, formula = (y ~ x + I(x^2)), color = "red", se = FALSE)
# Adding cubic model in green
graph1 <- graph1 + geom_smooth(method = lm, formula = (y ~ x + I(x^2) + I(x^3)), color = "green", se = FALSE)
graph1
```




- Again, the linear regression model seems to be a bad fit.
- Note how the cubic and quadratic fits are almost identical (green and red lines), that's probably because the cubic term is not statistically significant. We can verify this by looking at the regression output

## Real world example: Fitting a polynomial (2/2)

Let's estimate the three different regression models:

```
# Estimating linear model
reg1 <- lm(math ~ income, data = CASchools)
# Estimating quadratic model
reg2 <- lm(math ~ income + I(income^2), data = CASchools)
# Estimating cubic model
reg3 <- lm(math ~ income + I(income^2) + I(income^3), data = CASchools)

#Using stargazer for output
suppressMessages(library(stargazer))
stargazer(list(reg1, reg2, reg3), type = "html")
```

| | *Dependent variable:* | | |
|---|---|---|---|
| | math | | |
| | (1) | (2) | (3) |
| income | 1.815*** | 3.473*** | 3.976*** |
| | (0.091) | (0.310) | (0.877) |
| I(income2) | | -0.036*** | -0.059 |
| | | (0.006) | (0.038) |
| I(income3) | | | 0.0003 |
| | | | (0.0005) |
| Constant | 625.539*** | 610.346*** | 607.230*** |
| | (1.536) | (3.103) | (5.951) |
| Observations | 420 | 420 | 420 |
| $R^2$ | 0.489 | 0.525 | 0.525 |
| Adjusted $R^2$ | 0.488 | 0.522 | 0.522 |
| Residual Std. Error | 13.420 (df = 418) | 12.962 (df = 417) | 12.972 (df = 416) |
| F Statistic | 400.257*** (df = 1; 418) | 230.063*** (df = 2; 417) | 153.272*** (df = 3; 416) |
| *Note:* | | | *p<0.1;* ***p<0.05;*** p<0.01 |

**Interpretation:**

- Because the quadratic term in equation (2) is statistically significant we reject the hypothesis that `income` has no quadratic effect on `math`. Therefore, the relation is not linear and we can discard the ouput of regression (1).
- Because the cubic term in equation (3) is statistically insignificant we can't reject the hypothesis that `income` has no cubic effect on `math`. Therefore, we can discard regression (3).

/

- The fact that $\beta_2 < 0$ means that the effect of `income` on `math` is diminishing, i.e. for higher values of `income`, the effect of income on `math` is smaller than for lower values.

## Predicted effects using non-linear models

Let's see how the three models predict an increase in `income` from 10 to 20 and from 40 to 50 (recall that income is measured in thousands of dollars).

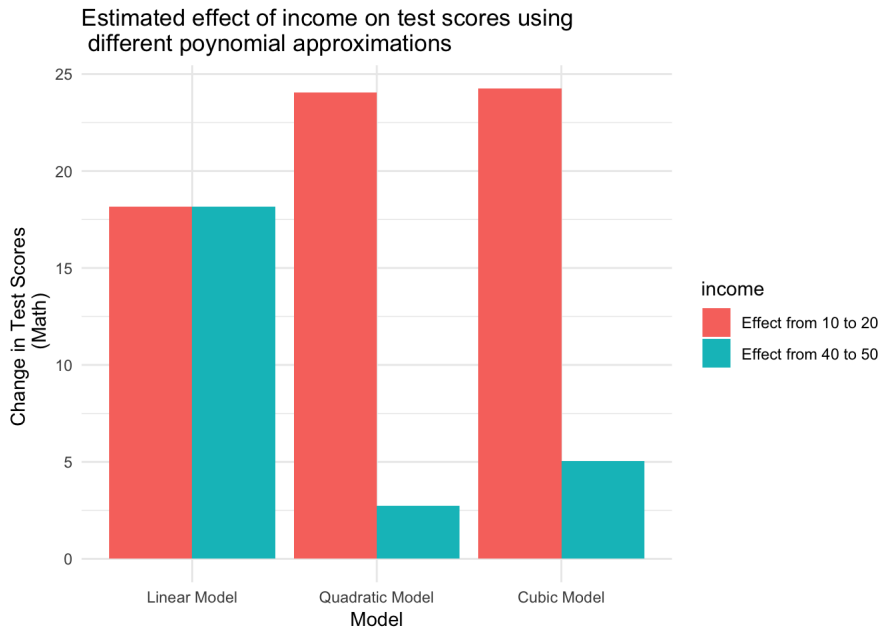
```
# Linear model approximation
lin_y <- predict(reg1, data.frame(income = c(10, 20, 40, 50)))
lin_deltaY_lowIncome <- lin_y[2] - lin_y[1]
lin_deltaY_highIncome <- lin_y[4] - lin_y[3]
linApprox <- c(lin_deltaY_lowIncome, lin_deltaY_highIncome)

# Quadratic model approximation
qua_y <- predict(reg2, data.frame(income = c(10, 20, 40, 50)))
qua_deltaY_lowIncome <- qua_y[2] - qua_y[1]
qua_deltaY_highIncome <- qua_y[4] - qua_y[3]
quaApprox <- c(qua_deltaY_lowIncome, qua_deltaY_highIncome)

# Cubic model approximation
cub_y <- predict(reg3, data.frame(income = c(10, 20, 40, 50)))
cub_deltaY_lowIncome <- cub_y[2] - cub_y[1]
cub_deltaY_highIncome <- cub_y[4] - cub_y[3]
cubApprox <- c(cub_deltaY_lowIncome, cub_deltaY_highIncome)

approxTable <- cbind(linApprox, quaApprox, cubApprox)
approxTable <- as.data.frame(approxTable)

names(approxTable) <- c("Linear Model",
                        "Quadratic Model",
                        "Cubic Model")
rownames(approxTable) <- c("Effect from 10 to 20",
                           "Effect from 40 to 50")

stargazer(approxTable, summary = FALSE, header = FALSE,  type = "html")
```

| | Linear Model | Quadratic Model | Cubic Model |
|---|---|---|---|
| Effect from 10 to 20 | 18.152 | 24.061 | 24.245 |
| Effect from 40 to 50 | 18.152 | 2.731 | 5.037 |

- Note how the linear model predicts a constant effect when changing from 10 to 20 and from 40 to 50.
- Both the quadratic and cubic model are able to capture the fact that income has a greater impact on test scores for lower values of income (between 10 and 20) than for higher values of income (between 40 and 50).

Let's take a look at the same result with a graph:

```
graphData <- data.frame( model = rep(colnames(approxTable), each = 2),
                         income = rep(rownames(approxTable), 3),
                         effect = c(linApprox, quaApprox, cubApprox) , stringsAsFactors = TRUE)
effectsGraph <- ggplot(data = graphData, aes(x = reorder(model, rep(c(1,2,3),2)), y = effect, fill = income)) + geom_bar(stat="identity", positi
                                  x = "Model",
                                  y = "Change in Test Scores \n (Math)")
effectsGraph + theme_minimal()
```




- Now is easy to see how using the linear model to make predictions for high values of $x$ is inappropiate.

## Good practices when specifying polynomial regressions

/

This are some good practices when dealing with polynomial regressions:

- Always inspect the scatter plot of $y$ on $x$. If for different regions of $x$, the effect on $y$ seems to be drastically different a linear regression is probably not a good fit and you should consider using a non-linear function.
- If the change in the slope is the same (either only increasing or diminishing for all values of $x$) then a quadratic term is probably enough to capture the non-linearity of the relationship. If there are more than one region in which the slope is increasing/diminishing then you should consider adding higher order terms. Is rare to add more than $k \geq 4$.
- Consecutively enter each next higher polynomial and run regressions at each stage. That is, if start with a linear model, then quadratic, then cubic, etc. Once the next higher polynomial has no longer a significant effect; drop that polynomial and use the previous model.
- Use prior research and theory to consider why the relation may not be linear.
- Lower order terms do not need to be significant (often that happens due to inherent multicollinearity). For example, in a $k = 4$ polynomial, the $k = 2$ and $k = 4$ terms will be correlated with each other and that will cause the standard errors to be inflated due to multicollinearity, and in some cases we won't be able to reject the null that the parameter for the term $k = 2$ is insignificant. This is not a sign of incorrect specification, just the curse of multicollinearity with polynomials. You will only drop a polynomial term if the higher order term is not significant; in the previous example, only if parameter of $k = 4$ is insignificant.

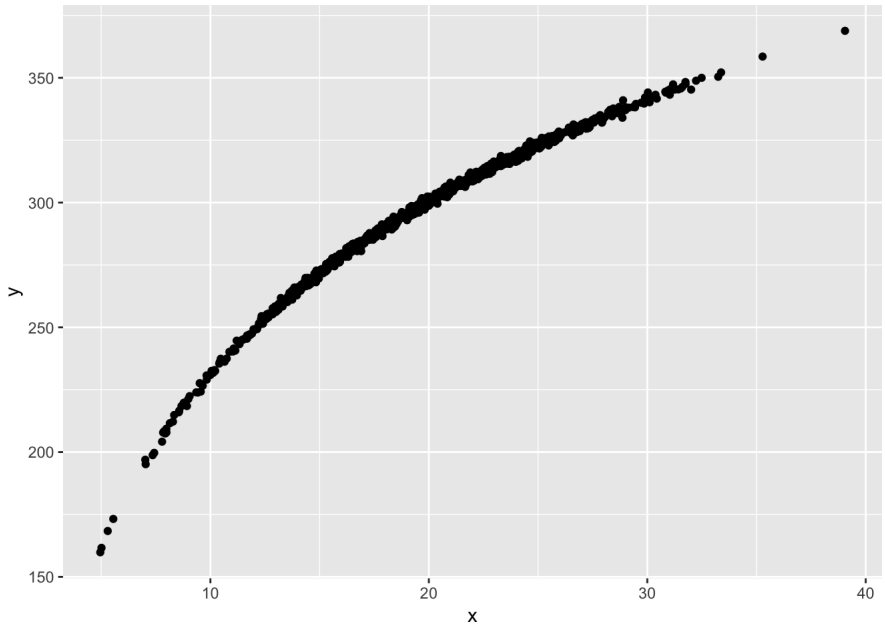
# Natural Logarithms in Regression Models

- Another way to specify a non-linear relation is to use the natural logarithm function $\ln(\dot{})$. We can transform either $y$ or $x$ to using the natural logarithm before estimating the regression.
- Logarithms convert changes in variables into percentage changes, and many relationships are naturally expressed in terms of percentages. Additionally, a logarithm transformation can change the scale in which a variable is measured, this is very convenient when there are a few high values outliers in the data - e.g. in the previous example, note how the income variable includes fewer high value observations –

**Exponential function and natural logarithm**

- The exponential function and its inverse, the natural logarithm, are important functional forms in the analysis of nonlinear regressions. The **exponential function** of $x$ is $e^x$ (`exp(x)` in `R`), where $e$ is the constant $2.71828...$. The **natural logarithm** is the inverse of the exponential function; that is, the natural logarithm is the function for which $x = \ln(e^x)$.
- This is how the natural logarithm function looks like:

```
set.seed(1)
n <- 1000 # Number of observations
x <- rnorm(n, mean = 20, sd = 5) # Generating x
y <- 1 + 100*log(x) + rnorm(n, mean = 0, sd = 1) # Generating y using a log function
df <- data.frame(y=y,x=x) # creating data frame
ggplot(df, aes(y=y,x=x)) + geom_point()
```




- Note that the effect of $x$ on $y$ is nonlinear and diminishing.

## Properties of the natural logarithm function

- The natural logarithm function has the following useful properties:

$$\begin{aligned}
\ln(1) &= 0 \\
\ln(0) &= -\infty \\
\ln(e) &= 1 \\
\ln(ab) &= \ln(a) + \ln(b) \\
\ln(a/b) &= \ln(a) - \ln(b) \\
\ln(1/a) &= \ln(1) - \ln(a) = -\ln(a) \\
\ln(a^b) &= b\ln(a) \\
\ln(e^a) &= a\ln(e) = a
\end{aligned}$$

/

And the function is only defined for $x > 0$. Be careful with this, as applying a the natural logarithm to a variable with 0 or negative values will return $-\infty$.

## Regression models using natural logarithm

There are three potential non-linear specifications using logarithms:

- **Case 1:** $x$ is in logarithms, $y$ is not:

$$y = \beta_0 + \beta_1 \ln(x) + \epsilon$$

This is often called the **linear-log** model.

- When $x$ increase by one unit, the effect on $y$ is:

$$\begin{aligned} y(x = \bar{x}) &= \beta_0 + \beta_1 \ln(\bar{x}) + \epsilon \\ y(x = \bar{x} + 1) &= \beta_0 + \beta_1 \ln(\bar{x} + 1) + \epsilon \\ \Delta(y) = y(x = \bar{x} + 1) - y(x = \bar{x}) &= \beta_0 + \beta_1 \ln(\bar{x} + 1) + \epsilon - (\beta_0 + \beta_1 \ln(\bar{x}) + \epsilon) \\ &= \beta_1 (\ln(\bar{x} + 1) - \ln(\bar{x})) \\ &= \beta_1 \ln(\frac{\bar{x} + 1}{\bar{x}}) \end{aligned}$$

- Similar to the polynomial specification, we can see how the change in $y$ depends on the value of $x$. In this particular case, note that the effect of $x$ on $y$ tends to zero as $x$ increase.

$$\lim_{x \to \infty} \Delta(y) = \lim_{x \to \infty} \ln(\frac{\bar{x} + 1}{\bar{x}}) = \ln(1) = 0$$

- **Case 2:** $y$ is in logarithms, $x$ is not:

$$\ln(y) = \beta_0 + \beta_1 x + \epsilon$$

This is often called the **log-linear** model.

- When $x$ increase by one unit, the effect on $y$ is a bit trickier to find because $y$ is expressed in $\ln(\cdot)$. In order to find the effect of $x$ on $y$, we need to get the values of $y$ using the exponential function:

$$\begin{aligned} \ln(y(x = \bar{x})) &= \beta_0 + \beta_1 \bar{x} + \epsilon \\ e^{\ln(y(x = \bar{x}))} &= e^{\beta_0 + \beta_1 \bar{x} + \epsilon} \\ y(x = \bar{x}) &= e^{\beta_0 + \beta_1 \bar{x} + \epsilon} \\ y(x = \bar{x} + 1) &= e^{\beta_0 + \beta_1 (\bar{x} + 1) + \epsilon} \end{aligned}$$

Then,

$$\begin{aligned} \Delta(y) = y(x = \bar{x} + 1) - y(x = \bar{x}) &= e^{\beta_0 + \beta_1 (\bar{x} + 1) + \epsilon} - e^{\beta_0 + \beta_1 \bar{x} + \epsilon} \\ &= e^{\beta_0 + \beta_1 \bar{x} + \epsilon} (e^{\beta_1} - 1) \end{aligned}$$

- Again, we can see how the change in $y$ depends on the value of $x$. Note that for the effect of $x$ on $y$ tends to infinity or minus infinity (depends on $\beta_1$) as $x$ increase.

$$\lim_{\bar{x} \to \infty} \Delta(y) = \lim_{\bar{x} \to \infty} e^{\beta_0 + \beta_1 (\bar{x}) + \epsilon} (e^{\beta_1} - 1) = \begin{cases} \infty & (e^{\beta_1} - 1) > 1 \\ -\infty & (e^{\beta_1} - 1) < 1 \end{cases}$$

In any case, the absolute value of the effect of $x$ on $y$ is **increasing**.

- **Case 3:** both $x$ and $y$ are in logarithms

$$\ln(y) = \beta_0 + \beta_1 \ln(x) + \epsilon$$

This is often called the log-log model.

- When $x$ increase by one unit, the effect on $y$ is:

$$\begin{aligned} \ln(y(x = \bar{x})) &= \beta_0 + \beta_1 \ln(\bar{x}) + \epsilon \\ e^{\ln(y(x = \bar{x}))} &= e^{\beta_0 + \beta_1 \ln(\bar{x}) + \epsilon} \\ y(x = \bar{x}) &= e^{\beta_0 + \beta_1 + \epsilon} (\bar{x}) \\ y(x = \bar{x}) &= e^{\beta_0 + \beta_1 + \epsilon} (\bar{x}) \\ y(x = \bar{x} + 1) &= e^{\beta_0 + \beta_1 + \epsilon} (\bar{x} + 1) \end{aligned}$$

Then,

$$\begin{aligned} \Delta(y) = y(x = \bar{x} + 1) - y(x = \bar{x}) &= e^{\beta_0 + \beta_1 + \epsilon} (\bar{x} + 1) - e^{\beta_0 + \beta_1 + \epsilon} (\bar{x}) \\ &= e^{\beta_0 + \beta_1 + \epsilon} \end{aligned}$$

- In this case the estimated effect of $x$ and $y$ is constant (independent of the values of $x$), but only after applying natural logarithm function to each variable. When a relation between to variables is made linear after applying a log transformation is often called **log-linearization**.

## Specifying a regression model with natural logarithm

- If the effect of $x$ on $y$ (the slope) seems to converge to zero (**diminhishing effect**) as $x$ increase then the linear-log model is the appropriate appropriate functional form.

/

- If the effect of $x$ on $y$ seems to approximate to positive or negative infinity (**increasing effect**) as $x$ increase then the linear-log model is the appropriate functional form.
- If the effect of $x$ on $y$ seems to be constant after applying a log transformation then then the log-log model is the appropriate functional form.

## Estimation with `R`

We can add logarithmic terms to a regression in two ways:

- **Manually**: Simply create a new variable based on $\ln(x)$ or $\ln(y)$, something like this `logx <- log(x)`. Then you can add `logx` to the `lm` command like any other variable, e.g. `lm(y ~ logx)` for the linear log model.
- **`I()` command**: Instead of creating a variable you `I()` command, that allows you to do a transformation to a variable before running the regression. Then, you will add `I(log(x))` to the `lm` command like this `lm(y ~ I(log(x)))`, which means: first transform `x` to `log(x)` and then run the regression.

## Interpretation of estimated parameter

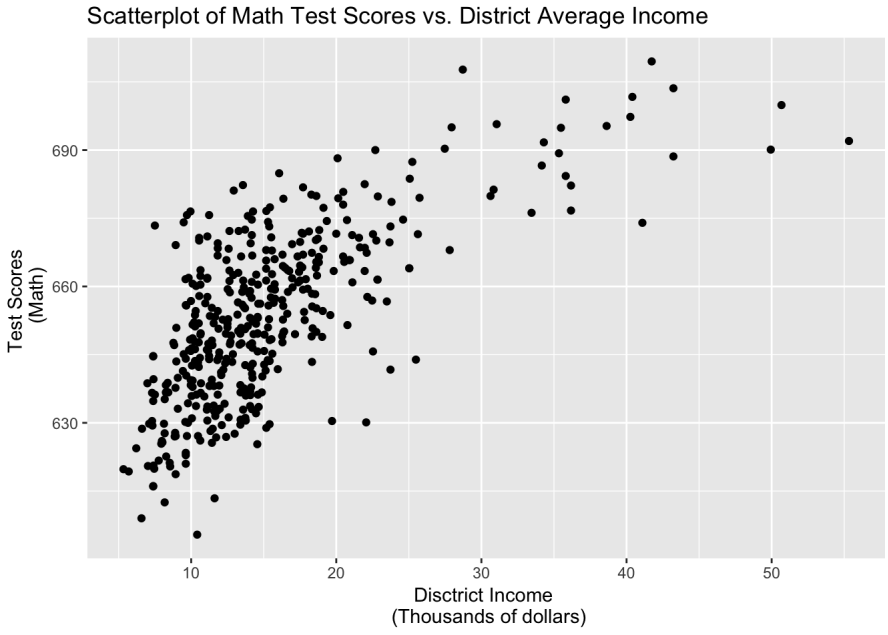
Because the non-linear transformation are often complex, the interpretation of $\beta_1$ is not as straightforward as for other models.The following table summarizes the interpretation of $\beta_1$ for the three natural log transformations:

| Case | Model Name | Regression Specification | Interpretation of $\beta_1$ |
|---|---|---|---|
| 1 | Linear-Log | $y = \beta_0 + \beta_1 \ln(x) + \epsilon$ | A 1% change in $x$ is associated with a change in $y$ of $0.01 \times \beta_1$ |
| 2 | Log-Linear | $\ln(y) = \beta_0 + \beta_1 x + \epsilon$ | A change in $x$ by one unit is associated with a $100 \times \beta_1\%$ change in $y$ |
| 3 | Log-Log | $\ln(y) = \beta_0 + \beta_1 \ln(x) + \epsilon$ | A 1% change in $x$ is associated with a $\beta_1\%$ change in $y$ |

## Real world example: Fitting a model with logarithms (1/2)

Let's illustrate the use of natural logarithms using the `CASchools` dataset. Let's take a look at the data again:

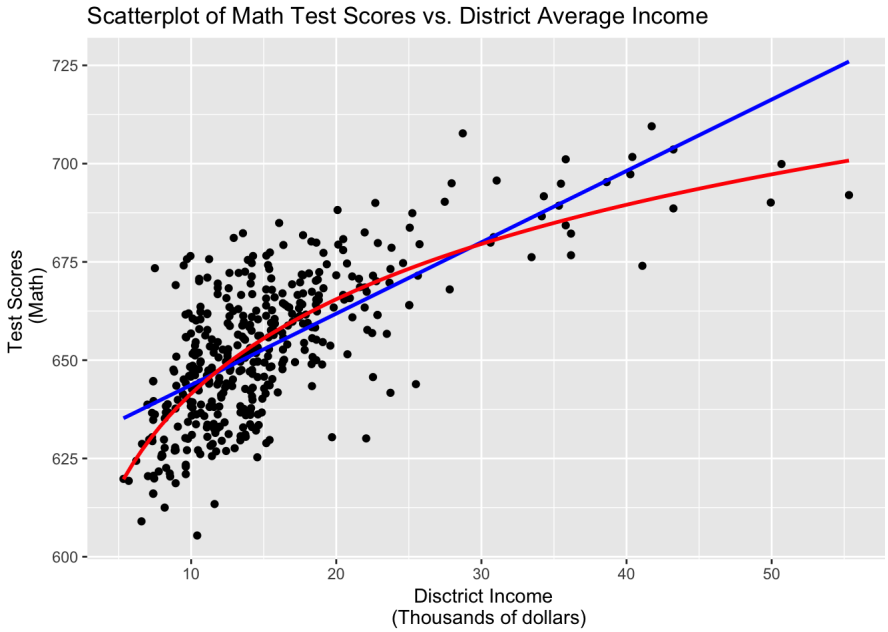
```
graph1 <- ggplot(data = CASchools, aes(x = income, y = math)) + geom_point()
graph1 <- graph1 + labs(title = "Scatterplot of Math Test Scores vs. District Average Income",
                        x = "Disctrict Income \n (Thousands of dollars)",
                        y = "Test Scores \n (Math)")
graph1
```




- Because the effect of $x$ on $y$ seems to be **diminishing** for all the range of $x$, it looks like a linear-log regression is a better fit than a linear function.

/

```
graph1 <- graph1 + geom_smooth(method = lm, formula = (y ~ x), color = "blue", se = FALSE)
graph1 <- graph1 + geom_smooth(method = lm, formula = (y ~ I(log(x))), color = "red", se = FALSE)
graph1
```




- Again, the linear regression model seems to be a bad fit.
- Note how the linear-log model is a better fit than the linear model.

## Real world example: Fitting a model with logarithms (2/2)

Now let's estimate the regression models, I'll estimate all possible specifications using logs just to illustrate the process:

```
reg1 <- lm(math ~ income, data = CASchools) #linear model
reg2 <- lm(math ~ I(log(income)), data = CASchools) #linear-log model
reg3 <- lm(I(log(math)) ~ income, data = CASchools) #log-linear model
reg4 <- lm(I(log(math)) ~ I(log(income)), data = CASchools) #log-log model

#Using stargazer for output
suppressMessages(library(stargazer))
stargazer(list(reg1, reg2, reg3, reg4), type = "html")
```

| | *Dependent variable:* | | | |
|---|---|---|---|---|
| | math | | I(log(math)) | |
| | (1) | (2) | (3) | (4) |
| income | 1.815*** | | 0.003*** | |
| | (0.091) | | (0.0001) | |
| I(log(income)) | | 34.664*** | | 0.053*** |
| | | (1.610) | | (0.002) |
| Constant | 625.539*** | 561.661*** | 6.440*** | 6.342*** |
| | (1.536) | (4.304) | (0.002) | (0.007) |
| Observations | 420 | 420 | 420 | 420 |
| $R^2$ | 0.489 | 0.526 | 0.481 | 0.522 |
| Adjusted $R^2$ | 0.488 | 0.525 | 0.480 | 0.521 |
| Residual Std. Error (df = 418) | 13.420 | 12.928 | 0.021 | 0.020 |
| F Statistic (df = 1; 418) | 400.257*** | 463.806*** | 387.338*** | 456.883*** |
| *Note:* | | | | *p<0.1; **p<0.05;** p<0.01 |

**Interpretation of $\beta_1$**

- **Linear Model**: A unit increase in `income` cause a 1.815 increase in `math`.
- **Linear-Log Model**: A 1% increase in `income` cause a change in `math` of $0.01 \times 34.664 = 0.346$.
- **Log-Linear Model**: A unit increase in `income` cause a $(100 \times 0.003)\% = 0.3\%$ change in `math`.

- **Log-Log Model**: A 1% increase in `income` cause a $0.053\%$ change in `math`.

- All the models reject the null that `income` is not related with `math`.