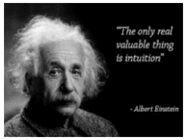
# Transformation #5: Predicting Count Data

YC(✘) – Y is not continuous, but counts

EV(✘) - The error variance is not constant



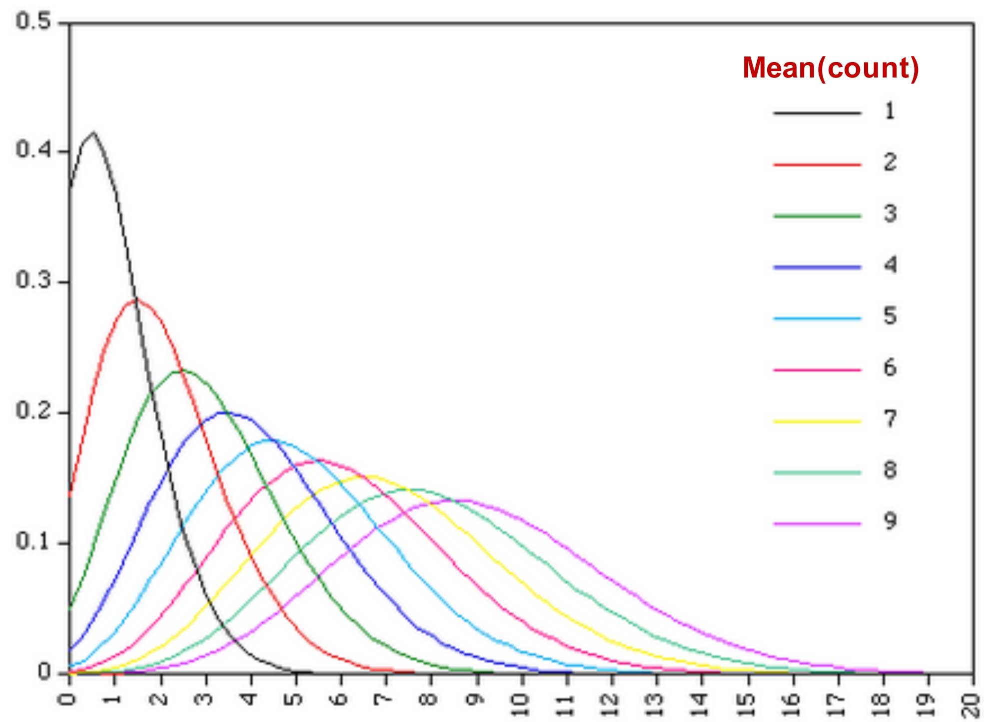



# Count Data Models: Intuition

- Predicting counts is a **common problem** in analytics – e.g., number of students enrolled, number of customers in a store, number of votes in an election, number of days to sell a house
- It is **not uncommon** to see predictive models for count using OLS
- But this is **incorrect** for a number of reasons, including
  - Counts are **discrete**, not continuous (i.e., no decimals)
  - Counts are **positive** – i.e., can't be less than 0 (i.e., data is **"truncated"** at 0.
  - The **distribution** of count data is **not normal**
  - The **error variance** is **not constant** – relatively low near 0 and increasing as counts get larger
  - In many count data sets, there is a **disproportionate** amount of **0's**

# The Poisson Distribution

- The **Normal** distribution: is **symmetrical** around the mean; it's **tails** extend **indefinitely** at both ends; and it is **continuous**.
- The **Poisson** distribution: is bounded at **0**; is **asymmetrical**; **varies** in **shape** as the mean increases (it approaches a normal distribution when counts have very wide ranges)
- As it turns out, **count data** follow a **Poisson** distribution
- Notice in the diagram how the **shape** of the distribution curve **changes** with the **count mean** and how it becomes **more normal** with **large** means.



# Count Data Models: Details

- **Log-transforming** the count data (i.e., outcome variable) and assuming a **Poisson distribution** for the counts yields more suitable regression models to predict count data:

$$Log(y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \varepsilon$$

- This type of regression is called **"Poisson"** model
- The **interpretation** of the coefficients is the same as the **Log-Linear** model explained before:

**x ↑ 1 unit → y ↑ 100*$\beta_1$% (i.e., $\beta_1$ fraction)**

- This model can be estimated with **OLS**
- But it is customary to estimate it using a **"Generalized Linear Model"** (GLM) regression using **Maximum Likelihood Estimation** (MLE), which we explain later in the chapter.
- It sounds complicated, but as we will explain later (see the binary logistic lecture), it can be fit in **R** using the ***glm()*** function by specifying the attribute ***family = poisson(link = "log")***

# R Tips

```
glm.count.fit = glm(y~x1+x2+etc.,
                    data=dataName,
                    family=poisson(link="log"))
```

→ Predicting count data outcome `y` using the `glm()`; `family=poisson` is the distribution used for count data and `link="log"` is the link function – i.e., the function used to transform the predicted variable `y`.

Kogod School
of
Business