Lecture 2

Intro to Generalized Linear Models

Review – maximum likelihood estimation

$$\log\big(L(\Theta)\big) = \sum_{i=1}^{n} f(y_i; \Theta)$$

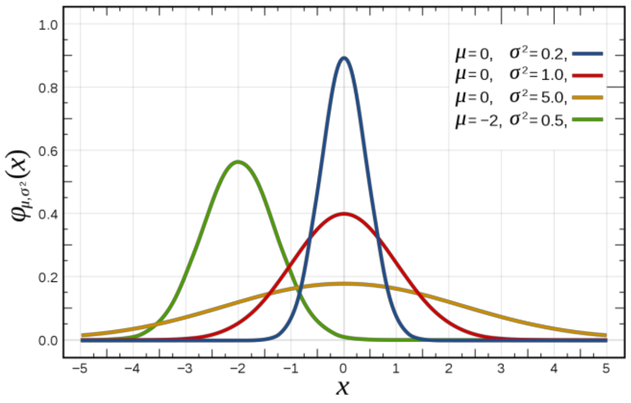
We want to find some value that maximizes log-likelihood

We previously looked at the form for the Gaussian and Poisson distributions

Common Probability Distributions

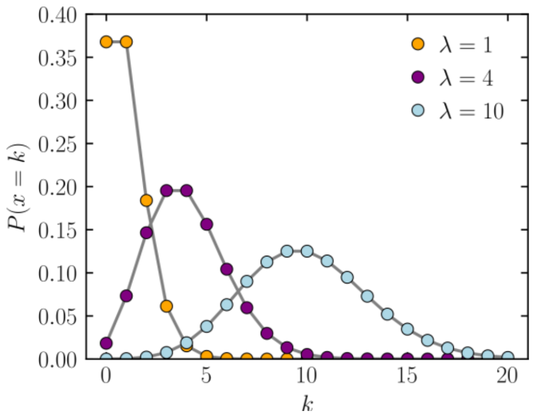
Lognormal

Gaussian Normal



- Parameters are mean and variance
- Real values
- Most physical properties and measurement errors
- Easy to add together

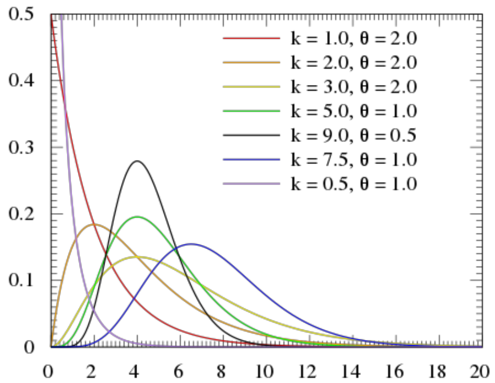
Poisson



- Parameter is the rate
- Discrete values
- Counts the number of events that occur in an interval given a **constant rate** and **independent events** – called a Poisson process

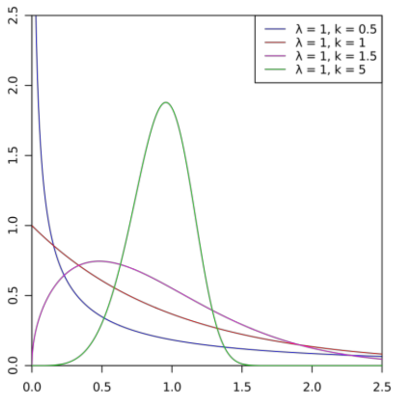
Common Probability Distributions

Gamma



- Parameters are shape (k) and scale (θ)
- Positive real values
- Measures time before k events occur in a Poisson process
- Case when k=1 is the **Exponential** distribution

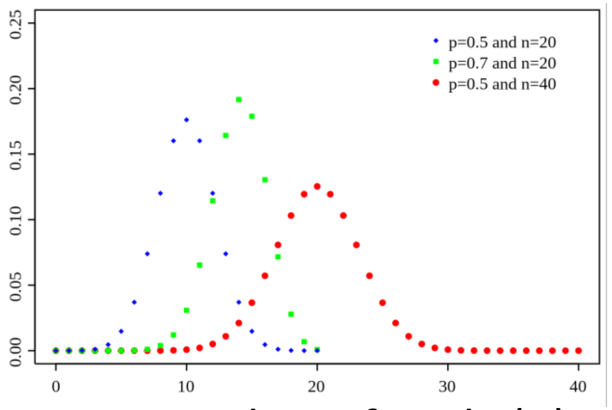
Weibull



- Parameters are scale (λ) and shape (k)
- Positive real values
- Measures a time-to-failure, when failure rate changes over time
  - K < 1 means failure rate decreases over time
  - K = 1 means failure rate is independent of time, this is also the **Exponential** distribution
  - K > 1 means failure rate increases over time

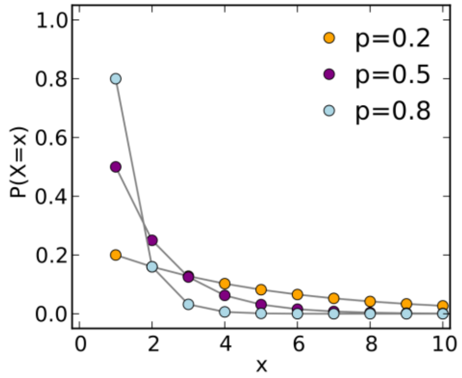
Common Probability Distributions

Binomial



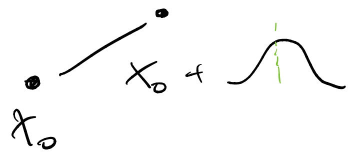
- Parameters are number of trials (n) and success probability (p)
- Positive discrete values
- Measures the probability of getting k successes from a Bernoulli process with success probability p

Geometric

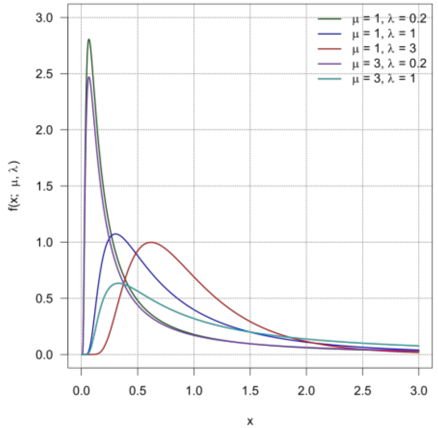


- Parameter is success probability (p)
- Positive discrete values
- Measures the number of Bernoulli trials needed to get one success

Common Probability Distributions

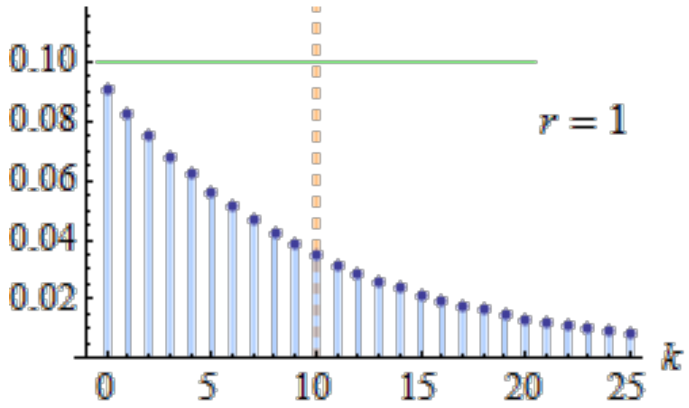


Inverse Gaussian



- Parameters are mean (μ) and shape (λ)
- Positive real values
- If the normal distribution is values in a random walk, the inverse Gaussian distribution is the number of steps taken to reach a given level

Negative Binomial



- Parameters are number of successes (r) and success probability (p)
- Measures the number of failures in a Bernoulli process before *r* successes

Comparing Probability Distributions

My cookie factory keeps having downtime. I want to understand this process better.

I want to know if the likelihood of a failure is dependent on the time since the last failure.

Start Time
8/24/2023 18:54
8/25/2023 11:11
8/25/2023 12:34
8/25/2023 12:53
8/25/2023 14:02
8/25/2023 14:46
8/27/2023 23:44
8/28/2023 0:01
8/28/2023 2:29
8/28/2023 4:28
8/28/2023 5:19
8/28/2023 5:30
8/28/2023 7:30
8/28/2023 8:34
8/28/2023 9:13
8/28/2023 10:22
8/28/2023 10:38
8/24/2023 18:54
8/25/2023 11:11
8/25/2023 12:34

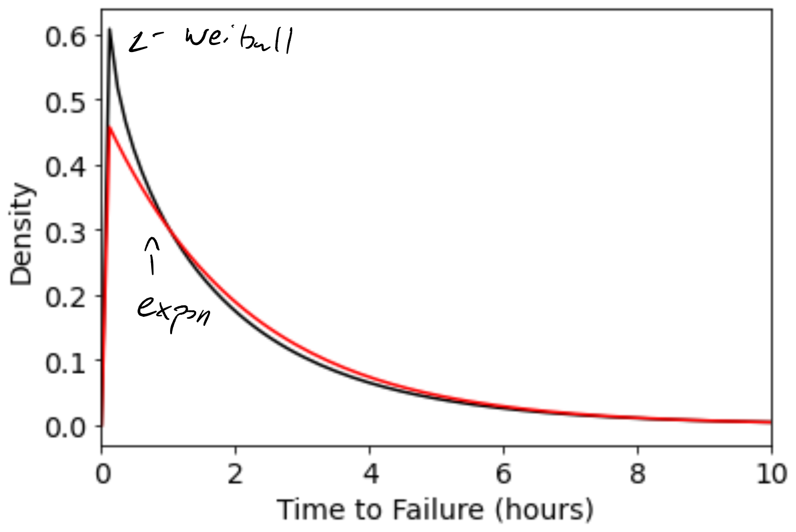
from scipy.stats import weibull_min, expon

Comparing Probability Distributions

```
# ttf is a numpy array of time to failure
weibull_dist = weibull_min.fit(ttf)
exponential_dist = expon.fit(ttf)

expon_loglik = np.sum([np.log(expon(*exponential_dist).pdf(x)) for x in ttf])
weibull_loglik = np.sum([np.log(weibull_min(*weibull_dist).pdf(x)) for x in ttf])

print(expon_loglik, weibull_loglik)
>>> -1413.542   -1407.631
```




```
print(Weibull_dist)
>>> (0.921326, 0.061388, 1.911398)

# Shape (k) is 0.921
# Location is 0.061
# Scale (lambda) is 1.911
```

$f(x)$

$f(x-L)$

Extending MLE – the Exponential Family

$$f(x|\theta) = e^{\frac{x*\theta - b(\theta)}{a(\phi)} + c(x,\theta)}$$

- θ is the **canonical parameter** of the distribution
- ϕ is the **diffusion parameter** of the distribution

Examples of the Exponential Family

$$f(x|\theta) = e^{\frac{x*\theta - b(\theta)}{a(\phi)} + c(x,\theta)}$$

For a normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x_j - \mu_0\right)^2}{2\sigma^2}\right)$$

- Canonical parameter (θ) is μ
- Diffusion parameter (ϕ) is σ^2
- $a(\phi) = \sigma^2$
- $b(\theta) = \frac{\mu^2}{2}$
- $c(x, \phi) = -\frac{x^2}{2\phi} - \log\sqrt{2\pi\phi}$

For a Poisson distribution:

$$f(x) = \frac{\lambda^k e^{-\lambda}}{k!}$$

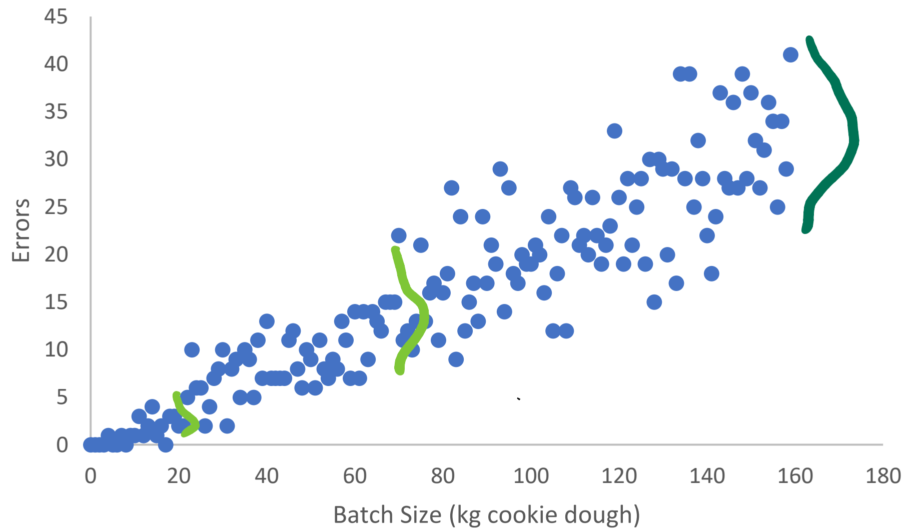
- Canonical parameter (θ) is $\log(\lambda)$
- Diffusion parameter (ϕ) is 1
- $a(\phi) = 1$
- $\boxed{b(\theta) = \lambda = e^\theta}$
- $c(x, \phi) = -\log\sqrt{x!}$

Motivating example

My cookie company is struggling with high error rates in cookie production.

I have measured the error rate from different cookie batch sizes.

How can I use this information to predict error rate in the future?



Predictive Modelling with Independent Variables

independent variable

• Inclusion of **covariate** information allows us to reduce the number of parameters in the model, while providing some predictive power.

• At a minimum, we seek estimators with

1. low bias.
2. small variance (*i.e.* we seek efficiency).
3. consistency: the estimator converges to the true parameter in probability when sample size goes to infinity.

Poisson Regression

errors

- Since we are modelling a rate of discrete occurrences (~~sales~~), we can assume it follows a Poisson distribution

- Our response variable is dependent on an independent variable; we want to understand this relationship

Poisson Regression

- We can begin by fitting a **different** distribution to **each** data point
- Taking a distribution from the exponential family

$$f(y|\theta) = e^{\frac{y*\theta - b(\theta)}{a(\phi)} + c(y,\theta)}$$

- Gives us the log-likelihood

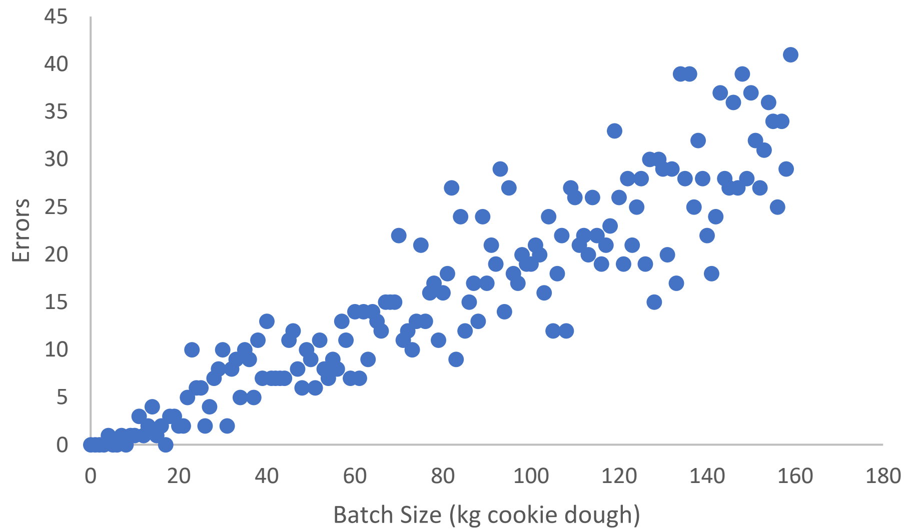
$$l(\theta) = {y\theta - b(\theta)}\Big/{a(\phi)} + c(y,\phi)$$

- Differentiating with respect to θ and finding the root gives us

$$l'(\theta) = \frac{y - b'(\theta)}{a(\phi)} \qquad \hat{\theta} = {b'}^{-1}(y)$$

Poisson Regression

- Find $\hat{\theta}$ for every data point
- This gives us a unique distribution for each data point
- Such a model is called a **saturated model**



Poisson Regression

- Assume our data follows a Poisson distribution with a canonical parameter (θ) **linearly dependent** on our batch size (x)

log-likelihood

$$l(\theta) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)$$

↓ substitute

Log-likelihood for point i →

$$l_i = y_i \beta x_i - b(\beta x_i) + c(y, \phi)$$

↓ differentiate

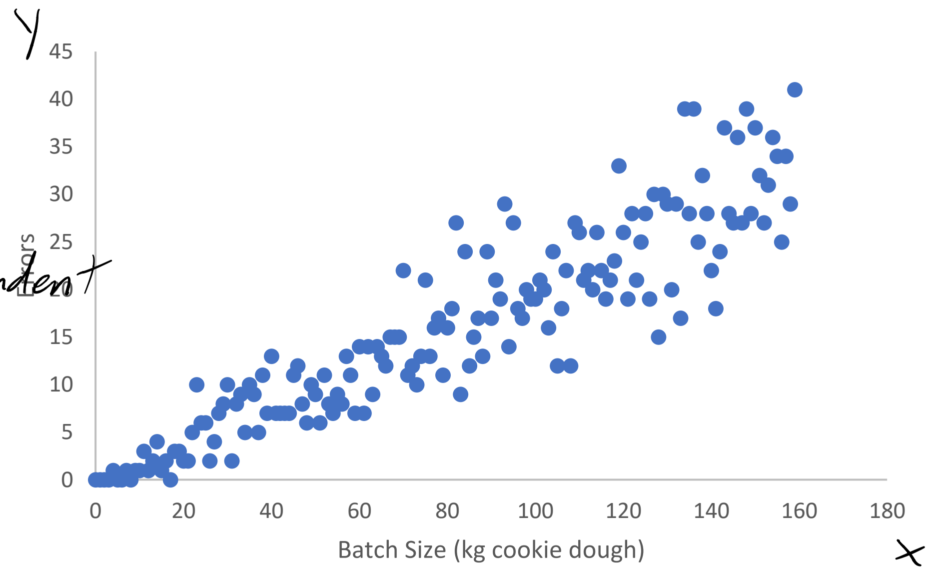
$$\frac{\partial l_i}{\partial \beta} = y_i x_i - b'(\beta x_i) x_i$$

independent

$$\theta = \beta x$$

canonical parameter

scalar



Poisson Regression

- Take the derivation with respect to β

All exponential family distributions are the same to here

$$\frac{\partial l_i}{\partial \beta} = y_i x_i - b'(\beta x_i) x_i$$

- Set the derivative to zero

$$0 = y_i x_i - e^{\beta x_i} x_i$$

- Rearrange the equation to solve

$$\hat{\beta} = \log(y_i) / x_i$$

$$b(\theta) = e^{\theta}$$

$$\frac{d}{dx} e^{x} = e^{x}$$

$$\frac{d}{d\beta} e^{\beta x} = e^{\beta x}$$

$$b(\theta) = \frac{\mu^2}{2}$$

$$b'(\theta) = \frac{2\mu}{2} = \mu$$

Poisson Regression

Instead of individual values of β, we want to come up with a single value that best represents all our data. We have some equations to help us:

$$\theta = \beta * \cancel{x} \qquad\qquad \hat{\beta} = \boxed{\log(y_i)\,/x_i}$$

$$a(\phi) = 1$$
$$b(\theta) = \lambda = e^{\theta}$$
$$c(x, \phi) = -\log \sqrt{x!}$$

$$l_i = y_i \beta x_i - b(\beta x_i) + c(y, \phi)$$

Substituting values into the log-likelihood, we can derive the equation:

$$\boxed{l(\vec{y}) = \sum_{i=1}^{n} y_i * \beta * x_i - e^{\beta * x_i} - \log(y_i!)}$$

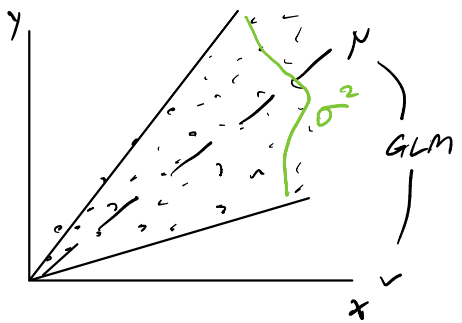
Finding $argmax\big(l(\vec{y})\big)$ is not trivial and is done numerically.

A full derivation and associated R code is available at https://statomics.github.io/SGA2019/assets/poissonIRWLS-implemented.html

Generalized Linear Model

- Generalized Linear Models (GLMs) predict a distribution in response to independent variables

- A GLM has three properties
  - A value *y* (dependent variable) is generated from a distribution
  - The mean of the distribution depends on some independent variables *X*
  - The link between *X* and the mean μ is called a **link function**



A closer look at link functions

- A **link function** is function that relates the independent variable to the mean of the distribution

- A **canonical link function** is derived from the distribution's density function
  - Link functions exist for every distribution in the exponential family
  - You do not have to use the canonical link function, but it's a good start

- When using a distribution with canonical parameter θ the link function has the form $\theta = b(\mu)$

parameter mean independent

$$\theta \xleftarrow[b(\mu)]{} \mu \xleftarrow[\text{link}]{} \beta x$$

Canonical Link Functions

Distribution	Uses	Canonical Link Function
Normal	Data with linear responses	μ
Poisson	Counts of occurrences within time/space	$\log(\mu)$
Exponential	Time between events	$-\frac{1}{\mu}$
Gamma	Sum of exponential response variables	$-\frac{1}{\mu}$
Binomial	Count of 1s in a series of [0,1] trials	$\log\left(\frac{\mu}{n-\mu}\right)$

Completing the motivating example

sm.GLM("Errors ~ Batch Size", data=df)

link = log(μ)

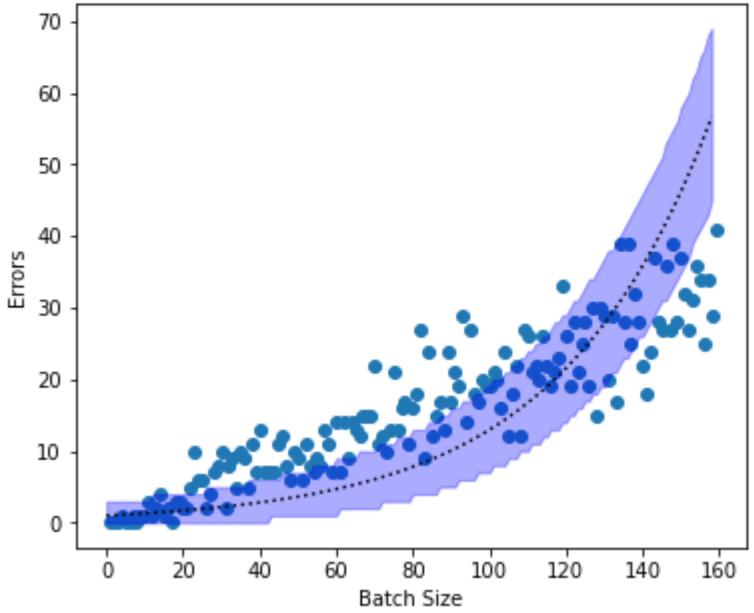
$f(x) = \log(\mu)$

$\mu = e^{f(x)}$

```
import statsmodels.api as sm
import pandas as pd
from scipy.stats import poisson

df = pd.read_excel("lecture2_figures.xlsx“)
poisson_glm = sm.GLM(df['Errors'],  <- endog = y
          distribution        df['Batch Size’], <- exog = X
                        family=sm.families.Poisson())
results = poisson_glm.fit()

predictions = results.predict()
low_bar = [poisson.ppf(0.05,x) for x in predictions]
high_bar = [poisson.ppf(0.95,x) for x in predictions]
```



results.summary() <-> summary(results)

Py R

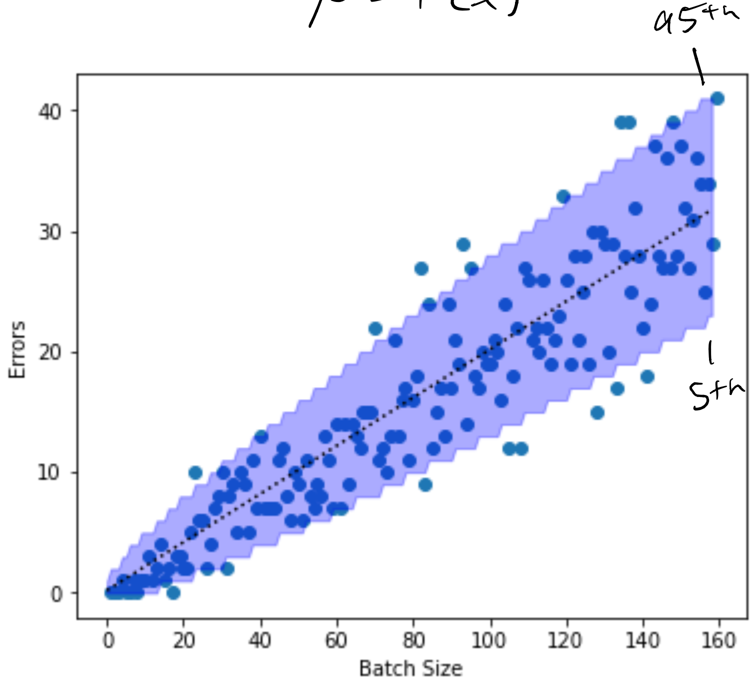
Completing the motivating example

link = μ
f(x) = μ
μ = f(x)

```
import statsmodels.api as sm
import pandas as pd
from statsmodels.genmod.families.links import Identity
from scipy.stats import poisson

df = pd.read_excel("lecture2_figures.xlsx“)
poisson_glm = sm.GLM(df['Errors'],
                      df['Batch Size’],
       family=sm.families.Poisson(link=Identity())
       )
results = poisson_glm.fit()

predictions = results.predict()
low_bar = [poisson.ppf(0.05,x) for x in predictions]
high_bar = [poisson.ppf(0.95,x) for x in predictions]
```



Completing the motivating example

```
import statsmodels.api as sm
import pandas as pd
from statsmodels.genmod.families.links import Identity
from scipy.stats import poisson

df = pd.read_excel("lecture2_figures.xlsx“)
poisson_glm = sm.GLM(df['Errors'],
                      df['Batch Size’],
       family=sm.families.Poisson(link=Identity())
       )
results = poisson_glm.fit()

predictions = results.predict()
low_bar = [poisson.ppf(0.05,x) for x in predictions]
high_bar = [poisson.ppf(0.95,x) for x in predictions]
```