

Stan [WIP]

# ¿Qué es Stan?

- ▶ Stan es un lenguaje escrito en C++ para realizar Inferencia Bayesiana en utilizando probabilísticos.
- ▶ El programa es de código abierto, todo lo relacionado a Stan se puede encontrar en <http://mc-stan.org/> junto con un manual de usuario instrucciones de uso.
- ▶ La primera versión, Stan 1.0, se lanzó en 2012 y actualmente el programa ya está en la versión XXX
- ▶ Se puede llamar a Stan desde R usando el paquete `{rstan}`, o desde Python usando el paquete `pystan`. Stan cuenta también cuenta con interfaces a otros lenguajes.

# Instalación de Stan y RStan

- ▶ En Windows (o Mac), escriba en la consola R:  
`install.packages("rstan", repositorios =  
c("https://mc-stan.org/r-packages/", getOption("repos")))`
- ▶ En Linux recomiendo instalar vía terminal `sudo`  
`add-apt-repository ppa:c2d4u.team/c2d4u4.0+`  
`sudo apt install -no-install-recomienda`  
`r-cran-rstan`
- ▶ Cualquier problema de instalación verifique:  
<https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>

# Elementos de un programa de Stan

En Stan definimos un modelo probabilístico utilizando el Idioma Stan.

- ▶ El modelo se define intuitivamente pero debe seguir algunas normas.
- ▶ Un programa en Stan se define en bloques, son:
  - ▶ Datos.
  - ▶ Parámetros.
  - ▶ Parámetros transformados.
  - ▶ Modelo.

# Datos

```
data {  
  int<lower=1> N; // Cantidad de observaciones  
  int<lower=1> K; // Cantidad de variables  
  matrix[N, K] X; // Matriz de diseño  
  vector[N] y;    // Vector de respuestas  
}
```

- ▶ En este bloque definimos qué datos, y de qué tipo, serán utilizado en el modelo.
- ▶ Tenemos que declarar el tipo y dimensión de los datos.
- ▶ Además de los datos, también puede haber otras constantes, tamaño de la muestra y número de predictores.
- ▶ Los tipos más comunes son: int, real, matriz, vector

# Parámetros

```
parameters {  
  vector[K] beta;  
  real<lower=0> sigma;  
}
```

- ▶ Los parámetros del modelo a estimar se declaran en el bloque de parámetros.
- ▶ Debemos imponer las restricciones necesarias a los parámetros.
- ▶ Los parámetros generalmente se definen como vectores (números reales) o uno real.

# Modelo

```
model {  
  vector[N] mu;  
  mu = X * beta;  
  
  // priors  
  beta ~ normal(0, 10);  
  sigma ~ cauchy(0, 5);  
  
  // likelihood  
  y ~ normal(mu, sigma);  
}
```

- ▶ El bloque modelo es donde definimos las distribuciones previas y la verosimilitud del modelo.
- ▶ También podemos declarar algunas variables que no son de nuestro interés pero facilita la escritura del modelo.

- ▶ Stan tiene muchas distribuciones de probabilidad ya definidas. Pero también existe la posibilidad de que el usuario defina su propia distribución.
- ▶ Es muy recomendable escribir el modelo de forma matricial.

**TODO** agregar ejemplos, parametrizaciones, y enlaces a la documentación



# Datos de tipo escalar

Enteros y reales sin restricciones:

```
int N;  
real theta;
```

Enteros y reales con restricciones:

```
int<lower = 1> N;  
real<lower = 0> sigma;  
real<lower = -1, upper = 1> rho;
```

# Vectores y matrices

Los vectores en Stan son de tipo columna y se declaran junto con la dimensión:

```
vector[3] u;
```

Tambien pueden tener restricciones

```
vector<lower = 0>[3] u;
```

Las matrices se declaran junto a su número de filas y columnas

```
matrix[3, 3] A;  
matrix<upper=0>[3, 4] B;
```

# Tipos especiales de matrices y vectores

Vector unitario simplex (suma de elementos es igual a 1)

```
simplex[5] theta;
```

Vector ordenado o positivo y ordenado

```
ordered[5] c;  
positive_ordered[5] d;
```

Vector fila

```
row_vector[3] u;
```

Matriz de covarianza

```
cov_matrix[K] Omega;
```

Matriz de correlación

```
corr_matrix[3] Sigma;
```

# Ejemplos

- ▶ Modelo Bernoulli
- ▶ Modelo binomial
- ▶ Modelo Poisson
- ▶ Modelo con *prior* uniforme
- ▶ Modelo de regresión lineal
- ▶ Modelo de regresión logística