Class Activity 5: Quiz [15 minutes]

1. List the assumptions implied by the ***linear regression model*** specification

$$y_i \sim \mathcal{N}(\mathbf{x}_i^\top \boldsymbol{\beta}, \sigma^2), i = 1, \cdots, n$$

2. Rewrite the above expression as a single sample from a ***multivariate normal distribution*** using the ***multivariate random variable*** $\mathbf{y}_{n\times 1}$, ***design matrix*** $X_{n\times p}$, and ***parameters*** $\boldsymbol{\beta}_{p\times 1}$ and $\boldsymbol{\Sigma}_{n\times n} = \sigma^2 I_{n\times n}$ (and include all the dimensions in your expression)
3. Write down the mathematical expression of the PDF of the above ***linear regression model*** as a ***multivariate normal distribution*** in terms of $\boldsymbol{\Sigma}_{n\times n}$ (instead of $\sigma^2 I_{n\times n}$) (and feel free to look up the expression of the pdf online if needed)
4. What family of ***priors*** would be ***conjugate*** for the ***multivariate parameter*** $\boldsymbol{\beta}$ for this ***linear regression model***?
5. What mathmematical form would a ***conjugate prior*** for the ***covariance matrix*** $\boldsymbol{\Sigma}$ (as opposed to σ^2 as in $\sigma^2 I$) be proportional to for this ***linear regression model***?

   Hint:

$$(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) = \text{tr}\left((\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right) = \text{tr}\left((\mathbf{y}-\mathbf{X}\boldsymbol{\beta})(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1}\right)$$

Class Activity 5: Solutions [15 minutes]

1. Assumptions of ***Linear Regression***

   A. indepedendent error terms B. normally distributed error terms C.homoskedastic error terms

   D. linear form and $\mathbf{x_i}$ having no randomness (measured without error)
2. ***Multivariate Normal Distribution*** specification of muliple linear regression

   $\mathbf{y}_{n\times 1} \sim \mathcal{MVN}(\mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \boldsymbol{\Sigma}_{n\times n} = \sigma^2 I_{n\times n})$
3. For ***positive definite*** Σ

$$p(\mathbf{y}_{n\times 1}) = (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2}\ \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right) \propto \exp\left(-\frac{1}{2}(\boldsymbol{\beta}^\top\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y}\right)$$

$$\propto \exp\left(-\frac{1}{2}\left(\boldsymbol{\beta}-(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}\right)^\top\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\left(\boldsymbol{\beta}-(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y})\right)\right) \quad \underbrace{\mathbf{y} = \hat{\mathbf{y}} + \hat{\boldsymbol{\epsilon}} \quad \text{and} \quad \mathbf{X}^\top \overbrace{\Sigma^{-1}}^{\sigma^{-2}I}\hat{\boldsymbol{\epsilon}}}_{\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \implies \mathbf{X}^\top\hat{\mathbf{y}} = \mathbf{X}^\top\mathbf{y}} = \mathbf{0}$$

4. The ***conjugate prior*** for $\boldsymbol{\beta}$ would be $\boldsymbol{\beta} \sim \mathcal{MVN}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta)$

Class Activity 5: Solutions [5 minutes]

1. Assumptions of ***Linear Regression***

   A. indepedendent error terms B. normally distributed error terms C.homoskedastic error terms

   D. linear form and $\mathbf{x_i}$ having no randomness (measured without error)
2. ***Multivariate Normal Distribution*** specification of muliple linear regression

   $\mathbf{y}_{n\times 1} \sim \mathcal{MVN}(\mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \boldsymbol{\Sigma}_{n\times n} = \sigma^2 I_{n\times n})$
3. For ***positive definite*** Σ

$$p(\mathbf{y}_{n\times 1}) = (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2}\ \exp\left(-\frac{1}{2}\text{tr}\left((\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right)\right)$$

$$\propto \ \det(\boldsymbol{\Sigma})^{-1/2}\ \exp\left(-\frac{1}{2}\text{tr}\left((\mathbf{y}-\mathbf{X}\boldsymbol{\beta})(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1}\right)\right)$$

4. The ***conjugate prior*** for $\boldsymbol{\beta}$ would be $\boldsymbol{\beta} \sim \mathcal{MVN}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta)$
5. The ***conjugate prior*** for $\boldsymbol{\Sigma}$ would be $p(\boldsymbol{\Sigma}) \propto \det(\boldsymbol{\Sigma})^{-\upsilon/2}\exp\left(-\frac{1}{2}\text{tr}\left(\Psi\boldsymbol{\Sigma}^{-1}\right)\right)$

   an [Inverse-Wishart distribution (https://en.wikipedia.org/wiki/Inverse-Wishart_distribution)](https://en.wikipedia.org/wiki/Inverse-Wishart_distribution) and it is ***conjugate*** since [determinants multiply (https://proofwiki.org/wiki/Determinant_of_Matrix_Product)](https://proofwiki.org/wiki/Determinant_of_Matrix_Product) and [traces add (https://proofwiki.org/wiki/Trace_of_Sum_of_Matrices_is_Sum_of_Traces#:~:text=let%20A%2BB%20denote%20the,denotes%20the%20trace%20of%20A.)](https://proofwiki.org/wiki/Trace_of_Sum_of_Matrices_is_Sum_of_Traces#:~:text=let%20A%2BB%20denote%20the,denotes%20the%20trace%20of%20A.)

Bayesian Linear Regression:
Multivariate Normal Distributions [10 minutes]

$$\mathbf{y}_{n\times 1} \sim \mathcal{MVN}(\mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \boldsymbol{\Sigma}_{n\times n} = \sigma^2 I_{n\times n})$$

$$p(\mathbf{y}|\boldsymbol{\beta}, \sigma, \mathbf{X}) = (2\pi)^{-n/2}\ \cancel{\det(\boldsymbol{\Sigma})^{-1/2}}\ \frac{1}{\sigma^{\frac{n}{2}}}\exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top\ \cancel{\boldsymbol{\Sigma}^{-1}}\ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right)$$

$$\boldsymbol{\beta} \sim \mathcal{MVN}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta) \quad or\ ?$$

$$\sigma \sim exponential(\lambda) \quad p(\sigma) = \lambda e^{-\lambda\sigma}1_{[0,\infty]}(\sigma)$$

$$or \quad \sigma \sim \text{HalfNormal}(\mu_\sigma, \sigma_\sigma), \sigma \sim \text{InverseGamma}(\alpha, \beta), \sigma \sim \text{TruncatedNormal}(\mu_\sigma, \sigma_\sigma, a, b) \quad or\ ?$$

In [1]:
```python
import pymc as pm; import numpy as np; n,p=100,10; X,y=np.zeros((n,p)),np.ones((n,1))
with pm.Model() as MLR:
    betas = pm.MvNormal('betas', mu=np.zeros((p,1)), cov=np.eye(p), shape=(p,1))
    sigma = pm.TruncatedNormal('sigma', mu=1, sigma=1, lower=0) # it's just a half normal, actually
    y = pm.Normal('y', mu=pm.math.dot(X, betas), sigma=sigma, observed=y)
    # y = pm.MvNormal('y', mu=X@betas, cov=sigma**2*np.eye(n), shape=(n,1), observed=y)

with MLR:
    idata = pm.sample()
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [betas, sigma]
```

100.00% [8000/8000 00:11<00:00 Sampling 4 chains, 0 divergences]

```
Sampling 4 chains for 1_000 tune and 1_000 draw iterations (4_000 + 4_000 draws total) took 11 seconds.
```

Homework 5: Part I

1. Go get data from kaggle.com and do a ***Bayesian Linear Regression*** analysis

```python
import pymc as pm; import numpy as np
n,p=100,10; X,y=np.zeros((n,p)),np.ones((n,1))
# Replace this made up data with your data set from kaggle...
with pm.Model() as MLR:
    betas = pm.MvNormal('betas', mu=np.zeros((p,1)), cov=np.eye(p), shape=(p,1))
    sigma = pm.TruncatedNormal('sigma', mu=1, sigma=1, lower=0) # half normal
    y = pm.Normal('y', mu=pm.math.dot(X, betas), sigma=sigma, observed=y)

with MLR:
    idata = pm.sample()
```

2. Choose ***prior*** that are sensible: certainly you might change the ***hyperparameters***, and perhaps you can experiment with different distributional families for `sigma` ...
3. [Optional] Assess the performance of the MCMC and note any issues or warnings
   A. Traceplots, inference (credible) intervals, effective sample sizes, energy plots, warnings and other notes... just the usual stuff they do [here (https://www.pymc.io/projects/docs/en/stable/learn/core_notebooks/pymc_overview.html#pymc-overview)](https://www.pymc.io/projects/docs/en/stable/learn/core_notebooks/pymc_overview.html#pymc-overview)
4. [Optional] Perform ***Multiple Linear Regression*** diagnostics... residual plots, etc.

Bayesian Linear Regression?
Geneal Σ Instead of $\sigma^2 I$ [10 minutes]

$$\begin{aligned}
\mathbf{y}_{n\times 1} &\sim \mathcal{MVN}(\mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \boldsymbol{\Sigma}_{n\times n} = \sigma^2 I_{n\times n})\\
p(\mathbf{y}|\boldsymbol{\beta},\sigma,\mathbf{X}) &= (2\pi)^{-n/2}\,\cancel{\det(\boldsymbol{\Sigma})^{-1/2}}\,\frac{1}{\sigma^{\frac{n}{2}}}\exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top\,\cancel{\boldsymbol{\Sigma}^{-1}}\,(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right)
\end{aligned}$$

$$\begin{aligned}
\mathbf{y}_{n\times 1} &\sim \mathcal{MVN}(\mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \boldsymbol{\Sigma}_{n\times n} \cancel{= \sigma^2 I_{n\times n}})\\
p(\mathbf{y}|\boldsymbol{\beta},\sigma,\mathbf{X}) &= (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right)
\end{aligned}$$

$$\boldsymbol{\beta} \sim \mathcal{MVN}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_\beta) \quad \textit{or}\ ?$$

$$p(\boldsymbol{\Sigma}) \propto \det(\boldsymbol{\Sigma})^{-\upsilon/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left(\Psi\boldsymbol{\Sigma}^{-1}\right)\right) \quad \textit{or}\ ?$$

Do we forsee any problems with what we're going to do here?

Conjugate Multivariate Normal Priors [15 minutes]

$$\begin{aligned}
p(\mathbf{y}|\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{X}) &= (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\beta)^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\beta)\right)\\
&\propto \exp\left(\beta^\top\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y} - \frac{1}{2}\beta^\top\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\beta\right)\\
&\propto \exp\left(-\frac{1}{2}\left(\beta - (\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y}\right)^\top\left[\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\right]\left(\beta - (\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y}\right)\right)\\
p(\boldsymbol{\beta}) \propto 1 \Longrightarrow p(\boldsymbol{\beta}|\boldsymbol{\Sigma},\mathbf{X},\mathbf{y}) &= \mathcal{MVN}\left(E[\boldsymbol{\beta}] = (\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y}, \mathrm{Var}[\boldsymbol{\beta}] = \left[\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\right]^{-1}\right)\\
&\textit{or} \quad \text{to use a conjugate family of priors...}\\
p(\boldsymbol{\beta}) &= \mathcal{MVN}\left(E[\boldsymbol{\beta}] = \boldsymbol{\beta}_0, \mathrm{Var}[\boldsymbol{\beta}] = \boldsymbol{\Sigma}_\beta\right) \quad \text{so}\\
p(\boldsymbol{\beta}|\boldsymbol{\Sigma},\mathbf{X},\mathbf{y}) &= \mathcal{MVN}\left(E[\boldsymbol{\beta}|\boldsymbol{\Sigma},\mathbf{X},\mathbf{y}] = \mathrm{Var}[\boldsymbol{\beta}|\boldsymbol{\Sigma},\mathbf{X},\mathbf{y}]^{-1}\left(\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y} + \boldsymbol{\Sigma}_\beta^{-1}\boldsymbol{\beta}_0\right), \mathrm{Var}[\boldsymbol{\beta}|\boldsymbol{\Sigma},\mathbf{X},\mathbf{y}] = \left[\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\right]^{-1} + \boldsymbol{\Sigma}_\beta^{-1}\right)
\end{aligned}$$

$$\text{Look familiar?} \Downarrow \qquad p(\theta|x,\theta_0,\tau,\phi) = \mathrm{N}\left(\overbrace{\frac{(\tau\theta_0 + \phi\sum_{i=1}^n x_i)}{(\tau + n\phi)}}^{\textbf{mean}}, \overbrace{\tau + n\phi}^{\textbf{precision}}\right)$$

Homework 5: Part II

Answer the following with respect to $p(\boldsymbol{\beta}|\boldsymbol{\Sigma}, \mathbf{X}, \mathbf{y})$ on the previous slide

1. Rewrite $p(\boldsymbol{\beta}|\boldsymbol{\Sigma}, \mathbf{X}, \mathbf{y})$ in terms of σ^2 (no longer using Σ) if $\Sigma = \sigma^2 I$
2. What is $E[\boldsymbol{\beta}|\boldsymbol{\Sigma}, \mathbf{X}, \mathbf{y}]$?
3. What ***hyperparameters*** values (legal or illegal) would make $E[\boldsymbol{\beta}|\boldsymbol{\Sigma}, \mathbf{X}, \mathbf{y}] = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$?
4. What ***hyperparameters*** values (legal or illegal) would make $E[\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}|\boldsymbol{\Sigma}, \mathbf{X}, \mathbf{y}] = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$?
5. What is $\text{Var}[\boldsymbol{\beta}|\boldsymbol{\Sigma}, \mathbf{X}, \mathbf{y}]$?

Inverse-Wishart Conjugate Priors for Σ [15 minutes]

$$\begin{aligned}
p(\mathbf{y}|\boldsymbol{\beta}, \Sigma, \mathbf{X}) &= (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\beta)^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\beta)\right)\\
&= (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}\text{tr}\left((\mathbf{y}-\mathbf{X}\beta)^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\beta)\right)\right)\\
&= (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}\text{tr}\left((\mathbf{y}-\mathbf{X}\beta)(\mathbf{y}-\mathbf{X}\beta)^\top\boldsymbol{\Sigma}^{-1}\right)\right)\\
p(\boldsymbol{\Sigma}) \propto 1 &\Longrightarrow \boldsymbol{\Sigma}|\boldsymbol{\beta}, \mathbf{X}, \mathbf{y} \sim \mathcal{W}^{-1}\left(\boldsymbol{\Psi} = (\mathbf{y}-\mathbf{X}\beta)(\mathbf{y}-\mathbf{X}\beta)^\top, \nu = -n\right)
\end{aligned}$$

---

an Inverse-Wishart distribution (https://en.wikipedia.org/wiki/Inverse-Wishart_distribution)

$$\begin{aligned}
\Longrightarrow p(\boldsymbol{\Sigma}) &= \frac{\det(\boldsymbol{\Psi})^{\nu/2}}{2^{\nu n/2}\Gamma_n(\frac{\nu}{2})}\det(\boldsymbol{\Sigma})^{-(\nu+n+1)/2}e^{-\frac{1}{2}\text{tr}(\boldsymbol{\Psi}\boldsymbol{\Sigma}^{-1})}\\
&\textit{or} \quad \text{to use a conjugate family of priors...}\\
p(\boldsymbol{\Sigma}) &= \mathcal{W}^{-1}\left(\boldsymbol{\Psi} = \boldsymbol{\Psi}_0, \nu = n+1\right) \quad \text{so}\\
\boldsymbol{\Sigma}|\boldsymbol{\beta}, \mathbf{X}, \mathbf{y} &\sim \mathcal{W}^{-1}\left(\boldsymbol{\Psi} = \boldsymbol{\Psi}_0 + (\mathbf{y}-\mathbf{X}\beta)(\mathbf{y}-\mathbf{X}\beta)^\top, \nu = n+2\right)
\end{aligned}$$

since determinants multiply (https://proofwiki.org/wiki/Determinant_of_Matrix_Product) and traces add (https://proofwiki.org/wiki/Trace_of_Sum_of_Matrices_is_Sum_of_Traces#:~:text=let%20A%2BB%20denote%20the,denotes%20the%20trace%20of%20A.)

Inverse-Wishart Distributions [10 minutes]

In [2]:
```python
import numpy as np; from scipy import stats
p = 2; Psi = np.eye(p) # 2x2 identity
try:
    stats.invwishart(df=-p, scale=Psi)
except ValueError as error:
    print(error)
```

```
Degrees of freedom must be greater than the dimension of scale matrix minus 1.
```

So $p(\boldsymbol{\Sigma}) \propto 1$ is an ***improper prior*** that results in an ***imporoper posterior***

If $p(\boldsymbol{\Sigma}) = \mathcal{W}^{-1}\left(\boldsymbol{\Psi} = \boldsymbol{\Psi}_0, \nu = n+1\right)$

In [25]:
```python
                                    # +1 comes from the multivariate normal distribution
myIWD = stats.invwishart(df=p+2, scale=Psi); myIWD.rvs(1) # p-1 also won't work
```

Out[25]:
```
array([[0.34511785, 0.37631635],
       [0.37631635, 0.5903704 ]])
```

$$E[\boldsymbol{\Sigma}] = \frac{\boldsymbol{\Psi}}{\nu - p - 1} \text{ for } \boldsymbol{\Sigma} \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}, \nu) \text{ with } \nu > p+1$$

That's why we made the "interesting" choice of $\nu = n+1$ for the ***conjugate prior*** specification above

In [20]:
```python
                               # df=p+2 won't work...
myIWD = stats.invwishart(df=p+2, scale=Psi); myIWD.rvs(size=100000).mean(axis=0)
```

Out[20]:
```
array([[0.96682326, 0.00990798],
       [0.00990798, 0.97517561]])
```

The LKJ (instead of the Inverse-Wishart) Distribution [18 minutes]

The ***covariance matrix*** $\boldsymbol{\Sigma}_{p\times p} = \mathbf{DRD} = \mathbf{DLL}^\top\mathbf{D}$ for

- $\mathbf{D} = \text{diag}(\sigma)$ the ***diagonal matrix*** of ***standard deviations***
- $\mathbf{R}$ the ***correlation matrix*** with all ***diagonal values*** equal to 1, and
- $\mathbf{L}$ the ***lower diagonal of the Cholesky decomposition*** of $\mathbf{R}$

The ***LKJ (Lewandowski-Kurowicka-Joe) prior*** is simpler than the ***Inverse-Wishart*** and is simple to evaluate (https://mc-stan.org/docs/functions-reference/cholesky-lkj-correlation-distribution.html)

$$p(\mathbf{R}) \propto \det(\mathbf{R})^{\eta-1} \quad f^{-1}(\mathbf{L}) = \mathbf{R} = \mathbf{L}\mathbf{L}^\top \quad J = \frac{d f^{-1}(\mathbf{L})}{d\mathbf{L}} = \frac{d\mathbf{L}\mathbf{L}^\top}{d\mathbf{L}} = \frac{d\mathbf{R}}{d\mathbf{L}} \quad J_{ij} = \frac{d\mathbf{r}_i}{d\mathbf{l}_j}$$

$$p(\mathbf{L}) = \det(\mathbf{L}\mathbf{L}^\top)^{\eta-1}\det(J) = \overbrace{\left(\prod_{k=1}^p \mathbf{L}_{kk}\right)^{\eta-1}}^{\det(\mathbf{L})^{\eta-1}} \overbrace{\left(\prod_{k=1}^p \mathbf{L}_{kk}^\top\right)^{\eta-1}}^{\det(\mathbf{L}^\top)^{\eta-1}} \overbrace{\left(\prod_{k=1}^p \mathbf{L}_{kk}^{p-k}\right)}^{\det(J)} = \prod_{k=\cancel{1}2}^p \mathbf{L}_{kk}^{p-k+2(\eta-1)}$$

and provides efficient computation of $(2\pi)^{-k/2}\det(\mathbf{\Sigma})^{-1/2}\ \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\beta)^\top\mathbf{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\beta)\right)$ since

- $\det(\mathbf{\Sigma}) = \det(\mathbf{L}\mathbf{L}^\mathbf{T}) = \prod_{k=1}^p \mathbf{L}_{kk}^2$ (as above) and $(\mathbf{y}-\boldsymbol{\mu})^\top\mathbf{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu}) = \epsilon^\top\mathbf{L}^{-\top}\mathbf{L}^{-1}\epsilon = (\mathbf{L}^{-1}\epsilon)^\top(\mathbf{L}^{-1}\epsilon) = \mathbf{x}^\top\mathbf{x}$ where $\mathbf{x}$ can be efficiently solved for based on ***lower triangular backwards substitution*** $\mathbf{L} \underset{\mathbf{L}^{-1}\epsilon}{x} = \epsilon$

The LKJ (instead of the Inverse-Wishart) Distribution [12 minutes]

$p(\mathbf{\Sigma}) = p(\sigma)p(\mathbf{R})$ and the $\eta = 1$ ***hyperparameter*** specifies a uniform distribution on ***correlation matrices***

$$p(\mathbf{R}) \propto \det(\mathbf{R}_{p\times p})^{\eta-1} \qquad \eta = 1 \text{ gives proper posteriors since } p(\mathbf{R}) \propto 1 \text{ is not the same as } p(\mathbf{\Sigma}) \propto 1$$

- The absolute ***determinant*** is the product of the ***singular values***
  (and the ***determinant*** is positive for ***positive definite matrices***)
- For ***correlation matrices*** the ***determinant*** is largest when all ***singular values*** (which sum to p)
  are equal to 1 which happens when all off-diagonal correlations are 0

Increasing $\eta \to \infty$ thus favors ***correlation matrices*** with smaller magnitudes of component correlations

In [30]:
```
import pymc as pm # https://www.pymc.io/projects/examples/en/latest/case_studies/LKJ.html
# https://www.pymc.io/projects/docs/en/stable/api/distributions/generated/pymc.LKJCholeskyCov.html
with pm.Model() as LKJ:
    packed_L = pm.LKJCholeskyCov("packed_L", n=2, eta=2.0,
                                 sd_dist=pm.Exponential.dist(1.0, shape=2), compute_corr=False)
packed_L.eval()
```

Out[30]:
```
array([ 0.15250025, -0.15400333,  1.12802046])
```

In [31]:
```
with LKJ:
    L = pm.expand_packed_triangular(2, packed_L)
    Sigma = L.dot(L.T)
Sigma.eval()#.shape
```

Out[31]:
```
array([[ 0.02325633, -0.02348555],
       [-0.02348555,  1.29614719]])
```

Bayesian Multivariate Normal Inference: the MVN-LKJ model *as opposed to Bayesian Linear Regression* [8 minutes]

$$\mathbf{y}_{n\times 1} \sim \mathcal{MVN}(\mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \mathbf{\Sigma}_{n\times n} = \sigma^2 I_{n\times n})$$
$$p(\mathbf{y}|\boldsymbol{\beta},\sigma,\mathbf{X}) = (2\pi)^{-n/2}\cancel{\det(\mathbf{\Sigma})^{-1/2}}\frac{1}{\sigma^{\frac{n}{2}}}\exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top\cancel{\mathbf{\Sigma}^{-1}}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right)$$

$$\mathbf{y}_{n\times 1} \sim \mathcal{MVN}(\mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \mathbf{\Sigma}_{n\times n} \cancel{= \sigma^2 I_{n\times n}})$$
$$p(\mathbf{y}|\boldsymbol{\beta},\sigma,\mathbf{X}) = (2\pi)^{-n/2}\det(\mathbf{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top\mathbf{\Sigma}^{-1}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right)$$

$$\mathbf{y}_i \sim \mathcal{MVN}(\boldsymbol{\mu}, \mathbf{\Sigma} = \mathbf{DRD})$$
$$p(\mathbf{y}|\boldsymbol{\beta},\sigma,\mathbf{X}) = (2\pi)^{-p/2}\det(\mathbf{\Sigma})^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top\mathbf{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right)$$
$$\boldsymbol{\beta} \sim \mathcal{MVN}(\boldsymbol{\beta}_0, \mathbf{\Sigma}_\beta)$$
$$\mathbf{R} \sim \mathcal{LKJ}(\eta) \qquad\qquad p(\mathbf{R}) \propto \det(\mathbf{R}_{p\times p})^{\eta-1}$$
$$\sigma_i \sim exponential(\lambda) \quad \text{and} \quad \mathbf{D} = \text{diag}(\boldsymbol{\sigma}) \qquad\qquad p(\sigma_i) = \lambda e^{-\lambda\sigma_i}1_{[0,\infty]}(\sigma_i)$$

$$(2\pi)^{-p/2}\det(\mathbf{DRD})^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top(\mathbf{DRD})^{-1}(\mathbf{y}-\boldsymbol{\mu})\right) \times \det(\mathbf{R}_{p\times p})^{\eta-1} \times \prod_{i=1}^p \lambda e^{-\lambda\sigma_i}1_{[0,\infty]}(\sigma_i)$$

Bayesian Multivariate Normal Inference: the MVN-LKJ model *as opposed to Bayesian Linear Regression* [12 minutes]

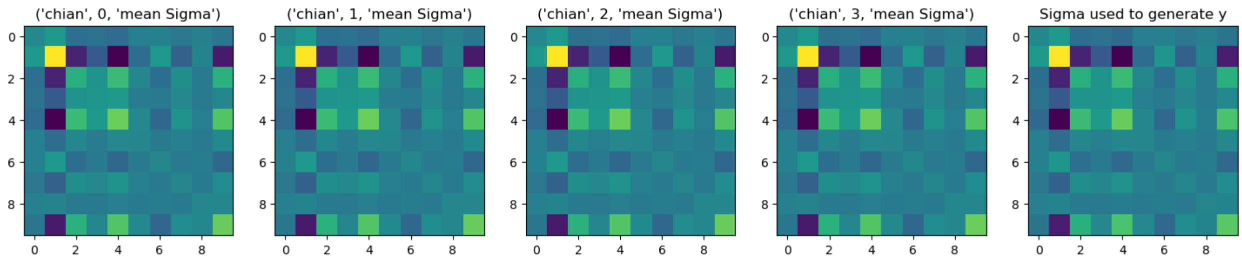
In [32]:
```python
import numpy as np; from scipy import stats; import pymc as pm; p=10; Psi=np.eye(p); a_cov = stats.invwishart(df=p+2, scale=Psi).rvs()
n=1000; y=stats.multivariate_normal(mean=np.zeros(p), cov=a_cov).rvs(size=n)
with pm.Model() as MNV_LKJ:
    packed_L = pm.LKJCholeskyCov("packed_L", n=p, eta=2.0, sd_dist=pm.Exponential.dist(1.0, shape=2), compute_corr=False)
    L = pm.expand_packed_triangular(p, packed_L); Sigma = pm.Deterministic('Sigma', L.dot(L.T))
    mu = pm.MvNormal('mu', mu=np.array(0), cov=np.eye(p), shape=p);
    y = pm.MvNormal('y', mu=mu, cov=Sigma, shape=(n,1), observed=y)
    idata = pm.sample()
```

```
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Multiprocess sampling (4 chains in 4 jobs)
NUTS: [packed_L, mu]
```

100.00% [8000/8000 00:41<00:00 Sampling 4 chains, 0 divergences]

```
Sampling 4 chains for 1_000 tune and 1_000 draw iterations (4_000 + 4_000 draws total) took 42 seconds.
```

In [33]:
```python
import matplotlib.pyplot as plt; fig,ax = plt.subplots(1,5,figsize=(18,4)); ax[-1].imshow(a_cov); ax[0].set_title(("Sigma used to genera
for chain in range(4):
    ax[chain].imshow(idata.posterior['Sigma'].mean(axis=1)[chain+1]); ax[chain].set_title(("chian",chain,"mean Sigma"))
```



Homework 5: Part III

1. Go get data from kaggle.com and perform inference for a ***Bayesian Multivariate Normal Model***

```python
import numpy as np; from scipy import stats
p=10; Psi=np.eye(p); a_cov = stats.invwishart(df=p+2, scale=Psi).rvs(1)
n=1000; y=stats.multivariate_normal(mean=np.zeros(p), cov=a_cov).rvs(size=n)
# Replace this made up data with your data set from kaggle...

with pm.Model() as MNV_LKJ:
    packed_L = pm.LKJCholeskyCov("packed_L", n=p, eta=2.0,
                                 sd_dist=pm.Exponential.dist(1.0, shape=2), compute_corr=False)
    L = pm.expand_packed_triangular(p, packed_L)
    # Sigma = pm.Deterministic('Sigma', L.dot(L.T)) # Don't use a covariance matrix parameterization
    mu = pm.MvNormal('mu', mu=np.array(0), cov=np.eye(p), shape=p);
    # y = pm.MvNormal('y', mu=mu, cov=Sigma, shape=(n,1), observed=y)
    # Figure out how to parameterize this with a Cholesky factor to improve computational efficiency
with MNV_LKJ
    idata = pm.sample()
```

2. As indicated above, don't use a covariance matrix parameterization and instead figure out how to parameterize this with a ***Cholesky factor*** to improve computational efficiency. The ***Cholesky***-based formulation allows general $O(n^3)$ $\det(\mathbf{\Sigma})$ to be computed using a simple $O(n)$ product and general $O(n^3)$ $\mathbf{\Sigma}^{-1}$ to be instead evaluated with $O(n^2)$ ***backward substitution***.
3. Specify ***priors*** that work: certainly you'll likely need to change the ***prior hyperparameters*** for $\boldsymbol{\mu}$ (`mu`) and $\mathbf{R}$ (`packed_L`)...
   A. And you could consider adjusting the ***prior*** for $\boldsymbol{\sigma}$ using `sd_dist` ...
4. [Optional] Assess the performance of the MCMC and note any issues
   A. Traceplots, inference (credible) intervals, effective sample sizes, energy plots, warnings and other notes... just the usual stuff they do [here (https://www.pymc.io/projects/docs/en/stable/learn/core_notebooks/pymc_overview.html#pymc-overview)](https://www.pymc.io/projects/docs/en/stable/learn/core_notebooks/pymc_overview.html#pymc-overview)