# SEM with Continuous and Ordered Categorical Variables for Prostate Cancer

WANG Lijun

Department of Statistics, CUHK

April 10, 2019

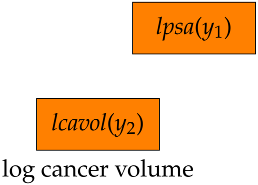
# Introduction

- Prostate Cancer: the second most common cancer in men worldwide(World Cancer Research Fund)
- Prostate-specific antigen (PSA):
  - monitor the progression of prostate cancer in men who had already been diagnosed with the disease.
  - increased availability of screening for PSA in men without symptoms of the disease.
- Dataset: Stamey et al. (1989) examined the relationship between the level of PSA and a number of clinical measures.
- Source: Friedman, Hastie, and Tibshirani (2001)

  https://web.stanford.edu/~hastie/ElemStatLearn/.

$\boxed{lpsa(y_1)}$

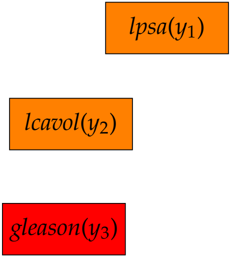
the log of prostate specific antigen (PSA)



$lpsa(y_1)$

$lcavol(y_2)$

log cancer volume

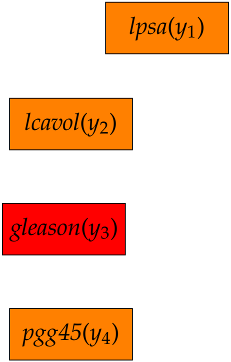


*lpsa*($y_1$)

*lcavol*($y_2$)

*gleason*($y_3$)

Gleason score, {6,7,8,9}



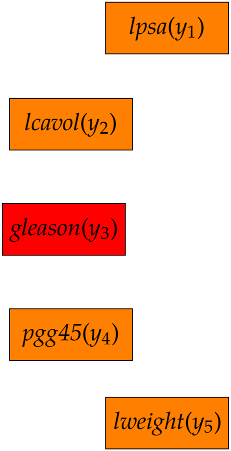
*lpsa*$(y_1)$

*lcavol*$(y_2)$

*gleason*$(y_3)$

*pgg45*$(y_4)$

percent of Gleason grade 4 or 5



$lpsa(y_1)$

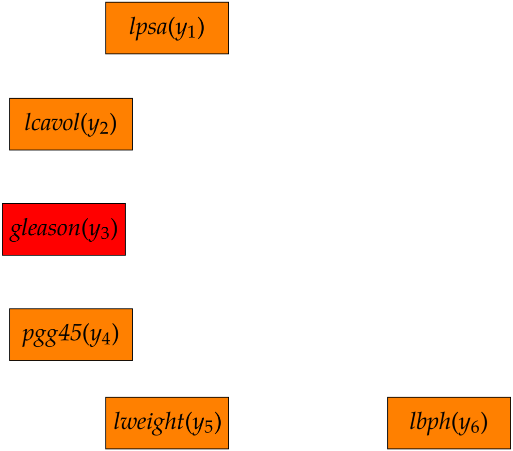
$lcavol(y_2)$

$gleason(y_3)$

$pgg45(y_4)$

$lweight(y_5)$

log prostate weight



$lpsa(y_1)$

$lcavol(y_2)$

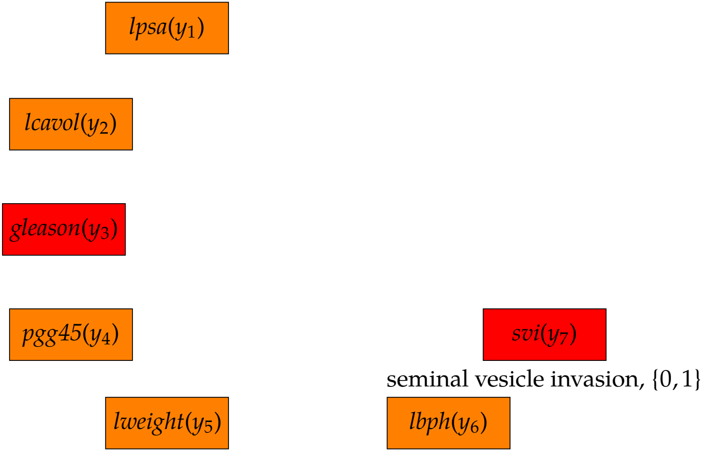
$gleason(y_3)$

$pgg45(y_4)$

$lweight(y_5)$

$lbph(y_6)$

log of benign prostatic hyperplasia



$lpsa(y_1)$

$lcavol(y_2)$

$gleason(y_3)$

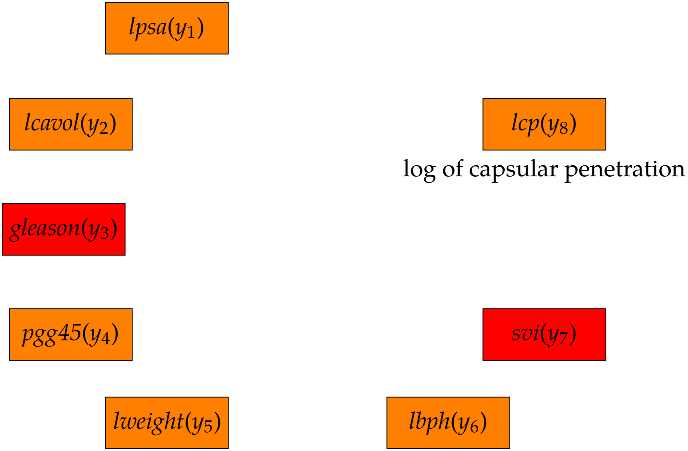
$pgg45(y_4)$

$svi(y_7)$

seminal vesicle invasion, $\{0,1\}$

$lweight(y_5)$

$lbph(y_6)$



$lpsa(y_1)$

$lcavol(y_2)$

$gleason(y_3)$

$pgg45(y_4)$

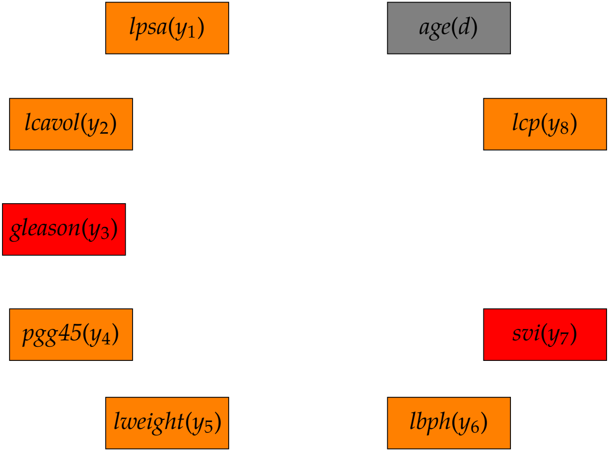
$lweight(y_5)$

$lbph(y_6)$

$svi(y_7)$

$lcp(y_8)$

log of capsular penetration



$lpsa(y_1)$

$age(d)$

$lcavol(y_2)$

$lcp(y_8)$

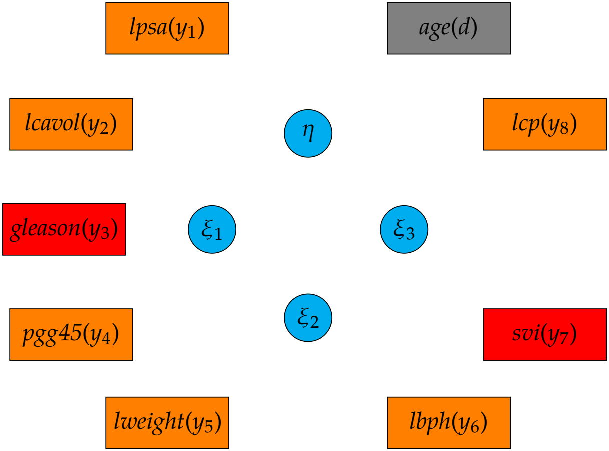
$gleason(y_3)$

$pgg45(y_4)$

$svi(y_7)$

$lweight(y_5)$

$lbph(y_6)$



$lpsa(y_1)$

$age(d)$

$lcavol(y_2)$

$\eta$

$lcp(y_8)$

$gleason(y_3)$

$\xi_1$

$\xi_3$

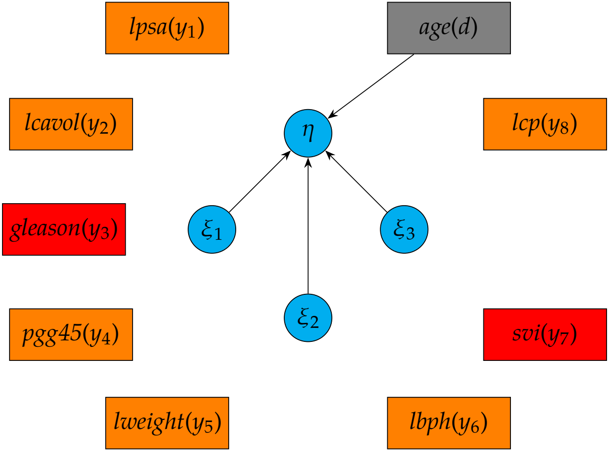
$\xi_2$

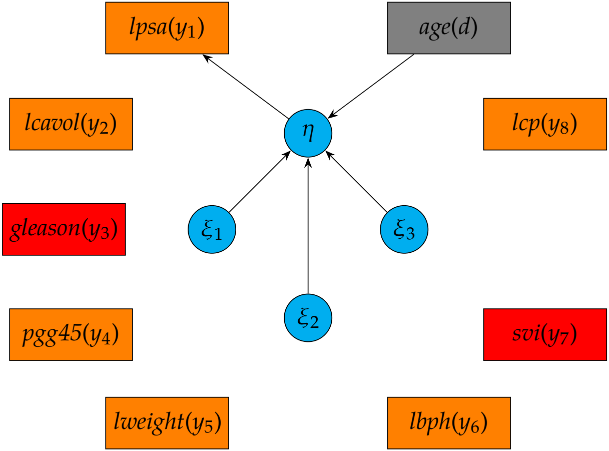
$pgg45(y_4)$

$svi(y_7)$

$lweight(y_5)$

$lbph(y_6)$


$lpsa(y_1)$
$age(d)$
$lcavol(y_2)$
$\eta$
$lcp(y_8)$
$gleason(y_3)$
$\xi_1$
$\xi_3$
$\xi_2$
$pgg45(y_4)$
$svi(y_7)$
$lweight(y_5)$
$lbph(y_6)$



$lpsa(y_1)$

$age(d)$

$lcavol(y_2)$

$\eta$

$lcp(y_8)$

$gleason(y_3)$

$\xi_1$

$\xi_3$

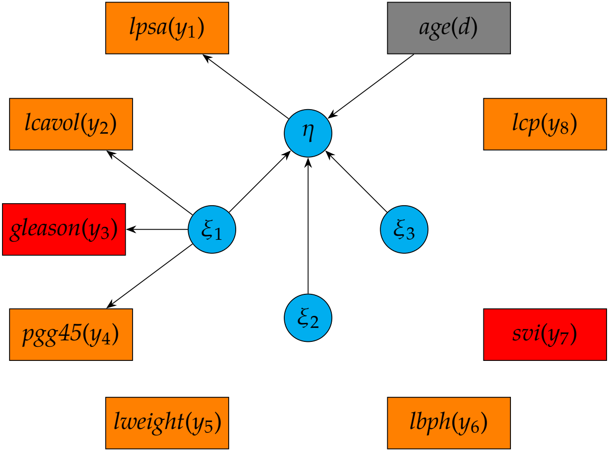
$pgg45(y_4)$

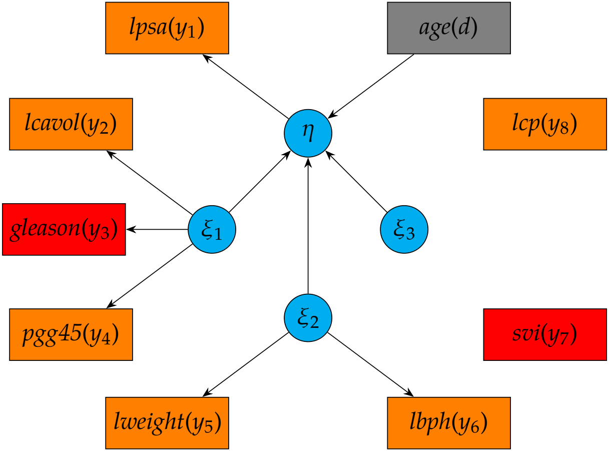
$\xi_2$

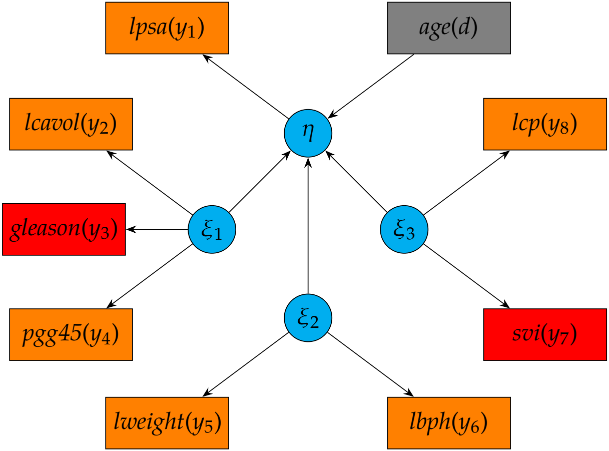
$svi(y_7)$

$lweight(y_5)$

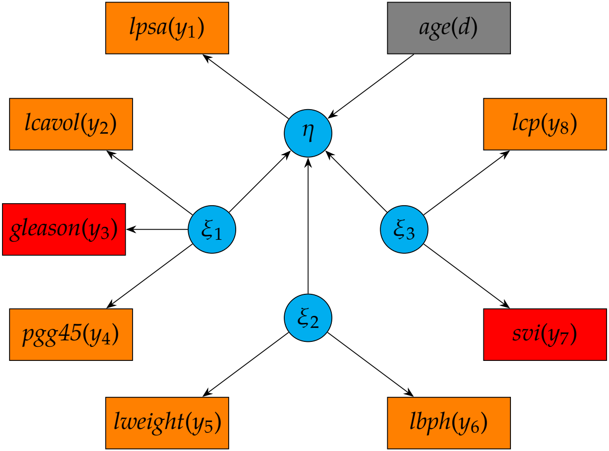
$lbph(y_6)$


$lpsa(y_1)$
$age(d)$
$lcavol(y_2)$
$\eta$
$lcp(y_8)$
$gleason(y_3)$
$\xi_1$
$\xi_3$
$\xi_2$
$pgg45(y_4)$
$svi(y_7)$
$lweight(y_5)$
$lbph(y_6)$


$lpsa(y_1)$
$age(d)$
$lcavol(y_2)$
$\eta$
$lcp(y_8)$
$gleason(y_3)$
$\xi_1$
$\xi_3$
$pgg45(y_4)$
$\xi_2$
$svi(y_7)$
$lweight(y_5)$
$lbph(y_6)$


$lpsa(y_1)$
$age(d)$
$lcavol(y_2)$
$\eta$
$lcp(y_8)$
$gleason(y_3)$
$\xi_1$
$\xi_3$
$pgg45(y_4)$
$\xi_2$
$svi(y_7)$
$lweight(y_5)$
$lbph(y_6)$

# Path Diagram



# Measurement and Structural Equation

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{\Lambda}\boldsymbol{\omega} + \boldsymbol{\varepsilon} \tag{1}$$

$$\eta = bd + \boldsymbol{\Gamma}\boldsymbol{\xi} + \delta\,, \tag{2}$$

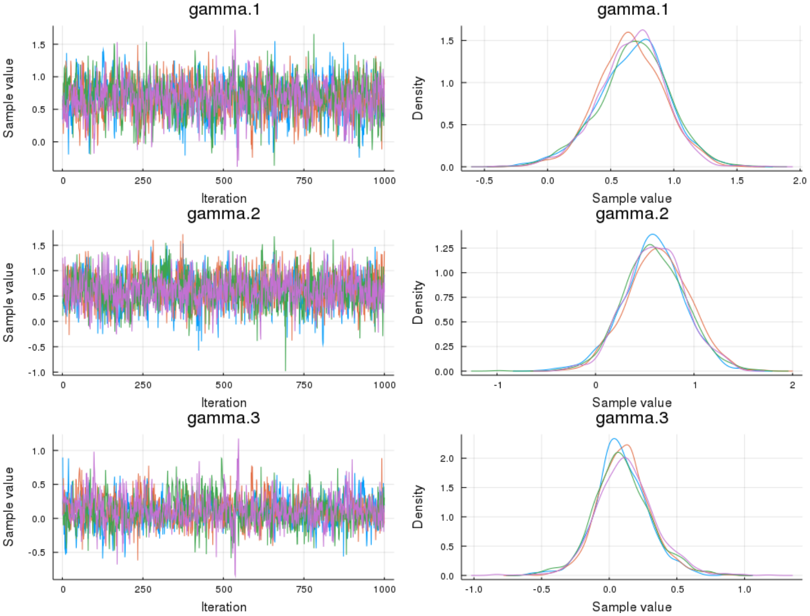
where

- $\boldsymbol{y} = [y_1, y_2, y_3^*, y_4, y_5, y_6, y_7^*, y_8]'$, where the unobservable $y_3^*, y_7^*$ are related to the ordered categorical variable $y_3, y_7$ via a set of thresholds.
- $\boldsymbol{\omega} = [\eta, \boldsymbol{\xi}']'$, and $\boldsymbol{\xi} = [\xi_1, \xi_2, \xi_3]' \sim N(\mathbf{0}, \boldsymbol{\Phi})$.
- $\boldsymbol{\Lambda}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \lambda_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \lambda_4 \end{bmatrix}$
- $\boldsymbol{\Gamma} = [\gamma_1, \gamma_2, \gamma_3]$
- $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \boldsymbol{\Psi}_\epsilon), \delta \sim N(0, \psi_\delta)$.

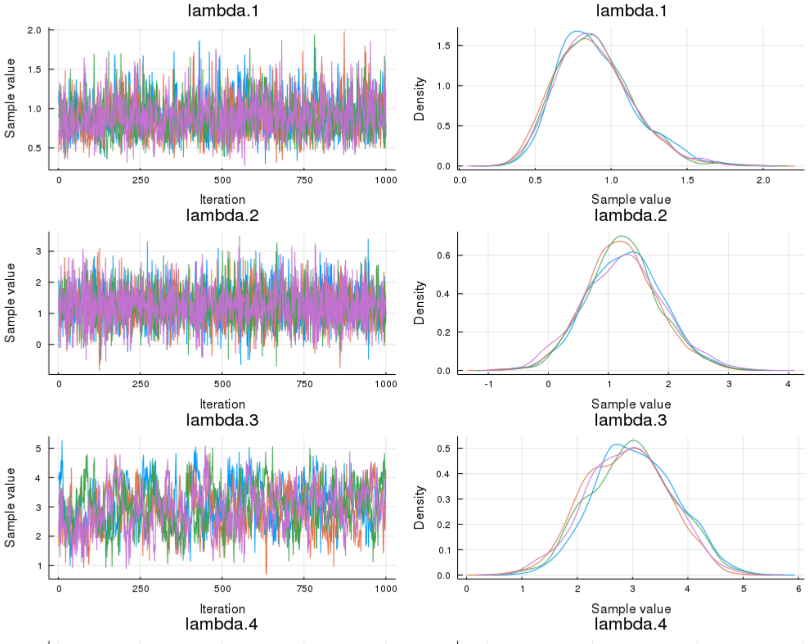
# Parameters

- priors:
  - hyperparameters for $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ are $\{0.9, 0.7, 0.9, 0.7\}$.
  - hyperparameters for $\{b, \gamma_1, \gamma_2, \gamma_3\}$ are $\{0.5, 0.4, 0.2, 0.5\}$.
  - hyperparameters for $\mathbf{\Phi}$ are $\rho_0 = 4$ and $\mathbf{R}_0^{-1} = \mathbf{I}$.
  - $\alpha_{0\epsilon k} = \alpha_{0\delta} = 9$ and $\beta_{0\epsilon k} = \beta_{0\delta} = 4$.
- sampling:
  - method: Hamiltonian Monte Carlo (Stan Interface for Julia)
  - number of iterations: 1000 (burn in) + 1000
  - number of chains: 4, each with different initial values

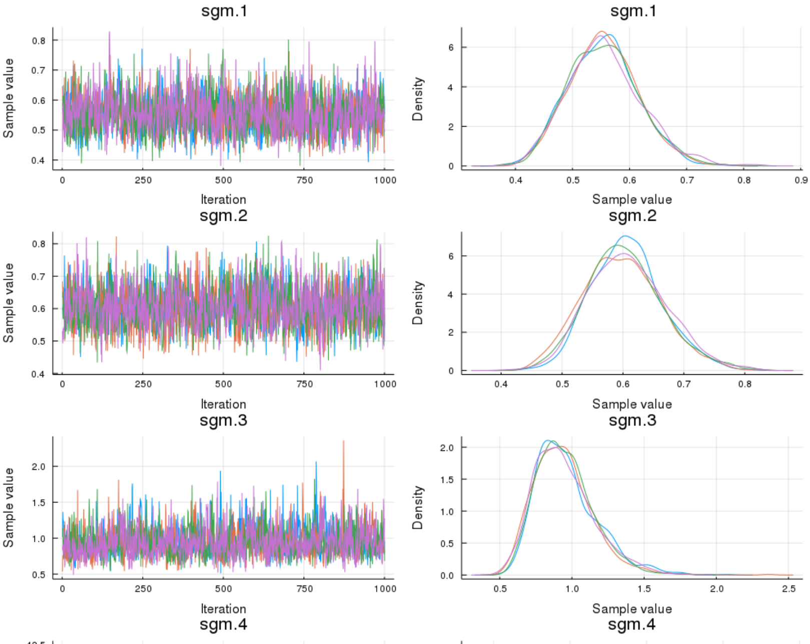
# Convergence ($\Gamma$)



# Convergence (Λ)



# Convergence ($\Psi_\epsilon$)



# Results

| | Mean | SD | SE | MCSE | | Mean | SD | SE | MCSE |
|---|---|---|---|---|---|---|---|---|---|
| b | 0.0088 | 0.0101 | 0.0002 | 0.001 | mu.8 | -0.2386 | 0.1443 | 0.0023 | 0.004 |
| c0 | 1.4981 | 1.1569 | 0.0183 | 0.0792 | phx.1.1 | 1.0871 | 0.2046 | 0.0032 | 0.0039 |
| c.1 | -0.402 | 1.0013 | 0.0158 | 0.0182 | phx.1.2 | 0.0668 | 0.0546 | 0.0009 | 0.0017 |
| c.2 | 2.0835 | 1.1226 | 0.0178 | 0.0244 | phx.1.3 | 1.453 | 0.7031 | 0.0111 | 0.0808 |
| c.3 | 2.2658 | 1.1455 | 0.0181 | 0.0254 | phx.2.1 | 0.0668 | 0.0546 | 0.0009 | 0.0017 |
| gamma.1 | 0.671 | 0.2705 | 0.0043 | 0.0083 | phx.2.2 | 0.1168 | 0.029 | 0.0005 | 0.0009 |
| gamma.2 | 0.6099 | 0.3108 | 0.0049 | 0.0088 | phx.2.3 | 0.0601 | 0.0931 | 0.0015 | 0.0037 |
| gamma.3 | 0.1062 | 0.2062 | 0.0033 | 0.0068 | phx.3.1 | 1.453 | 0.7031 | 0.0111 | 0.0808 |
| lambda.1 | 0.8925 | 0.249 | 0.0039 | 0.0062 | phx.3.2 | 0.0601 | 0.0931 | 0.0015 | 0.0037 |
| lambda.2 | 1.2511 | 0.6235 | 0.0099 | 0.0083 | phx.3.3 | 2.8512 | 3.1856 | 0.0504 | 0.391 |
| lambda.3 | 2.9679 | 0.7407 | 0.0117 | 0.0425 | sgd | 0.5515 | 0.0591 | 0.0009 | 0.0013 |
| lambda.4 | 0.8845 | 0.3198 | 0.0051 | 0.0331 | sgm.1 | 0.5549 | 0.0619 | 0.001 | 0.0013 |
| mu.1 | 1.852 | 0.6424 | 0.0102 | 0.0625 | sgm.2 | 0.6039 | 0.0621 | 0.001 | 0.0016 |
| mu.2 | 1.2855 | 0.1219 | 0.0019 | 0.0035 | sgm.3 | 0.9498 | 0.2057 | 0.0033 | 0.007 |
| mu.3 | 0.0296 | 0.9847 | 0.0156 | 0.0181 | sgm.4 | 32.4829 | 2.2894 | 0.0362 | 0.0279 |
| mu.4 | 2.0747 | 0.9671 | 0.0153 | 0.0146 | sgm.5 | 0.4218 | 0.0334 | 0.0005 | 0.0006 |
| mu.5 | 3.6116 | 0.0552 | 0.0009 | 0.001 | sgm.6 | 0.9644 | 0.1749 | 0.0028 | 0.009 |
| mu.6 | 0.0705 | 0.1417 | 0.0022 | 0.0026 | sgm.7 | 0.6703 | 0.1117 | 0.0018 | 0.0034 |
| mu.7 | 0.008 | 0.9626 | 0.0152 | 0.0138 | sgm.8 | 0.73 | 0.0896 | 0.0014 | 0.0043 |

# Interpretation

- All the factor loading $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ estimates 0.89, 1.25, 2.96, 0.88 are high, indicating strong associations between each latent variable and their corresponding indicators.
- The latent variables can be interpreted as
  - $\eta$: the level of prostate specific antigen
  - $\xi_1$: the status of cancer cells.
  - $\xi_2$: physical measurements of prostate.
  - $\xi_3$: living environment of cells.
- The estimated structural equation is

$$\eta = 0.0088d + 0.671\xi_1 + 0.6099\xi_2 + 0.1062\xi_3\,,$$

thus, the status of cancer cells ($\xi_1$) has most important effect on the level of PSA, and the physical measurements of prostate ($\xi_2$) has slightly less important effect, while the living environment ($\xi_3$) is not very important.

# Model Comparison (I)

Motivation: there might be some interactions between cancer cells ($\xi_1$) and prostate ($\xi_2$), then propose a nonlinear SEM,

$$M_0 : \eta = bd + \gamma_1 \xi_1 + \gamma_2 \xi_2 + \gamma_3 \xi_3 + \gamma_4 \xi_1 \xi_2 + \delta\,,$$

use the following linking model to compare it with the current model,

$$M_t : \eta = bd + \gamma_1 \xi_1 + \gamma_2 \xi_2 + \gamma_3 \xi_3 + (1-t)\gamma_4 \xi_1 \xi_2 + \delta\,,$$

where $M_1$ corresponds to the current model.

Result: $\widehat{\log B_{10}} = 0.4834$, which implies that the interaction $\xi_1 \xi_2$ doesn't make much difference.

# Model Comparison (II)

Motivation: there might be some interactions between cancer cells ($\xi_1$) and environment ($\xi_3$), then propose a nonlinear SEM,

$$M_0' : \eta = bd + \gamma_1 \xi_1 + \gamma_2 \xi_2 + \gamma_3 \xi_3 + \gamma_4 \xi_1 \xi_3 + \delta \,,$$

use the following linking model to compare it with the current model,

$$M_t' : \eta = bd + \gamma_1 \xi_1 + \gamma_2 \xi_2 + \gamma_3 \xi_3 + (1-t)\gamma_4 \xi_1 \xi_3 + \delta \,,$$

where $M_1'$ corresponds to the current model.
Result: $\widehat{\log B_{10}'} = 2.0830$, which slightly supports the current model.

# Model Comparison (III)

Motivation: *age* might not make much difference because prostate cancer is only for old people because the small coefficient $b = 0.0088$.

$$M_0'' : \eta = \gamma_1 \xi_1 + \gamma_2 \xi_2 + \gamma_3 \xi_3 + \delta \, ,$$

use the following linking model to compare it with the current model,

$$M_t'' : \eta = tbd + \gamma_1 \xi_1 + \gamma_2 \xi_2 + \gamma_3 \xi_3 + \delta \, ,$$

where $M_1''$ corresponds to the current model.
Result: $\widehat{\log B_{10}''} = -0.6812$, which implies that $M_0''$ is better.

# Sensitivity Analysis

Focus on $\hat{\mathbf{\Gamma}}$ with different hyperparameters in the priors of $\mathbf{\Gamma}$.

- Prior: {0.4, 0.2, 0.5}
- Prior I: {0.1, 0.1, 0.1}
- Prior II: {0.9, 0.9, 0.9}
- Prior III: {0.1, 0.1, 0.9}

| | $\hat{\gamma}_1(\hat{\sigma}_1)$ | $\hat{\gamma}_2(\hat{\sigma}_2)$ | $\hat{\gamma}_3(\hat{\sigma}_3)$ |
|---|---|---|---|
| Prior | 0.6710(0.2705) | 0.6099(0.3108) | 0.1062(0.2062) |
| Prior I | 0.6558(0.2747) | 0.5920(0.3133) | 0.1089(0.2040) |
| Prior II | 0.6920(0.2650) | 0.8339(0.3164) | 0.1107(0.2171) |
| Prior III | 0.5181(0.2925) | 0.6054(0.3155) | 0.2252(0.2258) |

Result: The results are robust to different prior inputs for $\mathbf{\Gamma}$. Specifically, $\xi_1$ and $\xi_2$ again have approximately equal effects on the level of PSA and $\xi_3$ is still not important even for the highest prior (case III).

# Conclusion

By the following techniques,

1. SEM with Continuous and Ordered Categorical Variables
2. Model Comparisons
3. Sensitivity Analysis

we obtain reasonable (expected) conclusions:

- the status of cancer cells and the physical measurements of prostate have approximately equal effects on PSA.
- the living environment is much less important for PSA.
- the *age* is not important at all, and even can be omitted in the model.

# Source code
# *model.stan*

```
/**********************************************
* Stan Program for the final project of STAT 5020
*
* author: WANG Lijun <ljwang@link.cuhk.edu.hk>
* date: April 9, 2019
*
***********************************************/
data {
    int<lower=1> N;
    vector[6] Y[N];
    int<lower=0, upper=9> Z[N, 2];
    int<lower=0> age[N];
}
transformed data {
    vector[3] zero = rep_vector(0, 3);
    cov_matrix[3] Phi0 = diag_matrix(rep_vector(1, 3));
}
parameters {
    vector[8] mu;
    vector[4] lambda;
    vector[3] gamma;
    vector<lower=0.0>[8] sgm2;
    real<lower=0.0> sgd2;
    cov_matrix[3] phx;
    vector[N] eta;
```

# Source code
# *model.stan* (Cont'd)

```
    vector[3] xi[N];
    ordered[3] c;
    real c0;
    real b;
}
transformed parameters {
    vector[8] u[N];
    vector[N] nu;
    vector[8] sgm = sqrt(sgm2);
    real sgd = sqrt(sgd2);

    for (i in 1:N){
        nu[i] = b * age[i] + gamma[1] * xi[i, 1] + gamma[2] * xi[i, 2] + gamma[3] *
              xi[i, 3];
        u[i, 1] = mu[1] + eta[i];
        u[i, 2] = mu[2] + xi[i, 1];
        u[i, 3] = mu[3] + lambda[1] * xi[i, 1];
        u[i, 4] = mu[4] + lambda[2] * xi[i, 1];
        u[i, 5] = mu[5] + xi[i, 2];
        u[i, 6] = mu[6] + lambda[3] * xi[i, 2];
        u[i, 7] = mu[7] + xi[i, 3];
        u[i, 8] = mu[8] + lambda[4] * xi[i, 3];
    }
}
model {
    vector[4] theta;
```

# Source code
# *model.stan* (Cont’d)

```
vector[2] theta0;

// prior
mu ~ normal(0, 1);
lambda[1] ~ normal(0.9, sgm[1]);
lambda[2] ~ normal(0.7, sgm[2]);
lambda[3] ~ normal(0.9, sgm[3]);
lambda[4] ~ normal(0.7, sgm[4]);
gamma[1] ~ normal(0.4, sgd);
gamma[2] ~ normal(0.2, sgd);
gamma[3] ~ normal(0.5, sgd);
b ~ normal(0.5, sgd);

sgm2 ~ inv_gamma(9, 4);
sgd2 ~ inv_gamma(9, 4);

phx ~ inv_wishart(4, Phi0);

for (i in 1:N) {
    eta[i] ~ normal(nu[i], sgd);
    xi[i] ~ multi_normal(zero, phx);
}

// likelihood
for (i in 1:N) {
    theta[1] = Phi((c[1] - u[i, 3]) / sgm[3]);
```

# Source code
# *model.stan* (Cont'd)

```
        theta[2] = Phi((c[2] - u[i, 3]) / sgm[3]) - Phi((c[1] - u[i, 3]) / sgm[3]);
        theta[3] = Phi((c[3] - u[i, 3]) / sgm[3]) - Phi((c[2] - u[i, 3]) / sgm[3]);
        theta[4] = 1 - Phi((c[3] - u[i, 3]) / sgm[3]);
        Y[i, 1] ~ normal(u[i, 1], sgm[1]);
        Y[i, 2] ~ normal(u[i, 2], sgm[2]);
        Z[i, 1] - 5 ~ categorical(theta);
        Y[i, 3] ~ normal(u[i, 4], sgm[4]);
        Y[i, 4] ~ normal(u[i, 5], sgm[5]);
        Y[i, 5] ~ normal(u[i, 6], sgm[6]);
        theta0[1] = Phi((c0 - u[i, 7]) / sgm[7]);
        theta0[2] = 1 - theta0[1];
        Z[i, 2] + 1 ~ categorical(theta0);
        Y[i, 6] ~ normal(u[i, 8], sgm[8]);
    }
}
/* only for model comparisons
generated quantities {
    real U = 0;
    for (i in 1:N) {
        U -= (eta[i] - b*age[i] - gamma[1]*xi[i,1] - gamma[2]*xi[i,2] - gamma[3]*xi[i
              ,3] - gamma[4]*(1-t)*xi[i,1]*xi[i,2] ) * (gamma[4] * xi[i, 1] * xi[i
              ,2]) / sgd2;
    }
}
*/
```

# Source code
# *sem.jl*

```julia
## ##################################################
## Julia Program for the final project of STAT 5020
##
## author: WANG Lijun <ljwang@link.cuhk.edu.hk>
## date: April 9, 2019
##
## ##################################################
using CmdStan
using DelimitedFiles
data = readdlm("prostate.data", ',')
Y = data[2:end, [9, 1, 8, 2, 4, 6]]
age = data[2:end, 3]
Z = data[2:end, [7, 5]]
N = size(Y)[1]

inputdata = Dict("N" => N, "Y" => Y, "Z" => Z, "age" => age)
monitor = vcat("mu.".*string.(1:8),
                "lambda.".*string.(1:4),
                "gamma.".*string.(1:3),
                "sgm.".*string.(1:8), "sgd",
                "phx.1.".*string.(1:3), "phx.2.".*string.(1:3), "phx.3.".*string
                      .(1:3),
                "b",
                "c.".*string.(1:3), "c0")
```

# Source code
# *sem.jl* (Cont’d)

```
# run
model = Stanmodel(model = read(open("model.stan"), String), monitors = monitor)
rc, sim, cnames = stan(model, inputdata)

## ############################################
## save traceplots
## ############################################
using MCMCChains
using StatsPlots
p1 = plot(sim[vcat("sgm.".*string.(1:4))])
savefig(p1, "sgm1to4.png")
p2 = plot(sim[vcat("sgm.".*string.(5:8))])
savefig(p2, "sgm5to8.png")
p3 = plot(sim[vcat("lambda.".*string.(1:4))])
savefig(p3, "lambda.png")
p4 = plot(sim[vcat("gamma.".*string.(1:3))])
savefig(p4, "gamma.png")
p5 = plot(sim[["phx.1.1","phx.1.2","phx.1.3","phx.2.2","phx.2.3", "phx.3.3"]])
savefig(p5, "phx.png")
p6 = plot(sim[vcat("b","c.".*string.(1:3), "c0")])
savefig(p6, "bc.png")
p7 = plot(sim["sgd"])
savefig(p7, "sgd.png")

## ############################################
```

# Source code
# *sem.jl* (Cont'd)

```
## calculate logBF
## ############################################
using StatsBase
function logBF(model, data; nt::Int=10)
    U = ones(nt+1)
    for s = 1:(nt+1)
        data["t"] = 1/nt*(s-1)
        rc, sim, cnames = stan(model, data, summary = false)
        U[s] = mean(sim[:,:,1])
    end
    res = 0
    for s = 1:nt
        res += ( U[s] + U[s+1] ) * (1 / nt) / 2
    end
    return res
end

## ############################################
## model comparisons
## ############################################

# comparison I
model_c = Stanmodel(model = read(open("model_c.stan"), String), output_format=:array,
      monitors = ["U"], nchains = 1)
res = ones(10)
for i=1:10
```

# Source code
# *sem.jl* (Cont'd)

```
    res[i] = logBF(model_c, inputdata)
end

# comparison II
model_c13 = Stanmodel(model = read(open("model_c13.stan"), String), output_format=:
      array, monitors = ["U"], nchains = 1)
res = ones(10)
for i=1:10
    res[i] = logBF(model_c13, inputdata)
end

# comparison III
model_cd = Stanmodel(model = read(open("model_cd.stan"), String), output_format=:
      array, monitors = ["U"], nchains = 1)
res = ones(10)
for i = 1:10
    res[i] = logBF(model_cd, inputdata)
end

## #############################################
## sensitivity analysis
## #############################################
model_s1 = Stanmodel(model = read(open("model_s1.stan"), String), monitors = monitor,
       nchains = 4)
rc, sim, cnames = stan(model_s1, inputdata)
```

# Source code *sem.jl* (Cont’d)

```
model_s2 = Stanmodel(model = read(open("model_s2.stan"), String), monitors = monitor,
      nchains = 4)
rc, sim, cnames = stan(model_s2, inputdata)
model_s3 = Stanmodel(model = read(open("model_s3.stan"), String), monitors = monitor,
      nchains = 4)
rc, sim, cnames = stan(model_s3, inputdata)
```

*Thank You!*