2017.09.22 (15:40~17:10)
日本心理学会第81回大会TWS
ベイジアンデータ解析入門

回帰分析を例に ベイジアンデータ解析 を体験してみる

広島大学大学院教育学研究科
平川　真

ベイジアン分析のステップ(p.24)

1) データの特定
2) モデルの定義 **(解釈可能な)モデルの作成**
3) パラメタの事前分布の設定
4) ベイズ推論を用いて、パラメタの値に確信度を再配分

ベイズ推定

5) 事後予測がデータを模倣できているかを確認

記述的妥当性のチェック

データの特定

被予測変数と予測変数を決める

どの変数を記述したいのか（被予測変数; y）

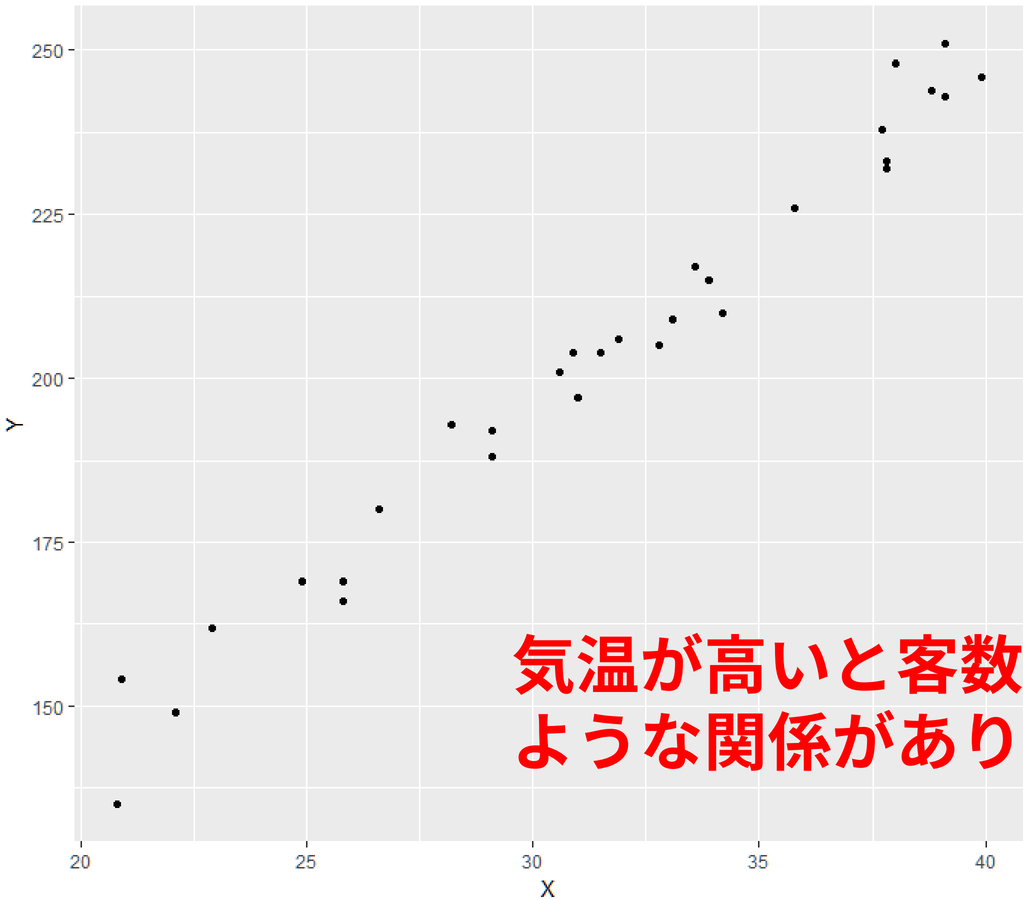
どの変数で記述したいのか（予測変数; x）

「アイス屋さんの客数を気温で予測する」

ということを考えてみる（架空データ）

データの確認

気温(x)と 客数(y)の30ポイントのデータ



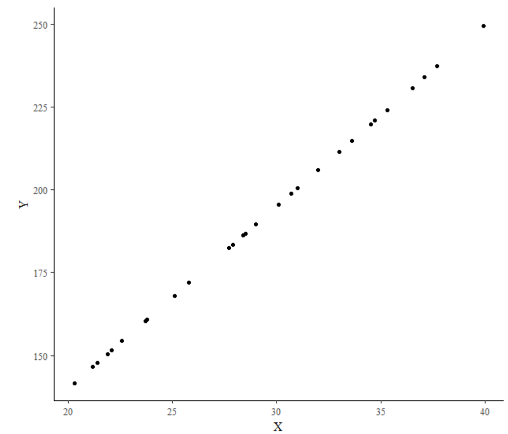
気温が高いと客数が多いような関係がありそう

	X	Y
1	25.8	169
2	35.8	226
3	28.2	193
4	37.7	238
5	38.8	244
6	20.9	154
7	30.6	201
8	37.8	232
9	31.0	197
10	29.1	188
11	39.1	251
12	29.1	192
13	33.6	217
14	31.5	204
15	22.1	149

モデルの定義

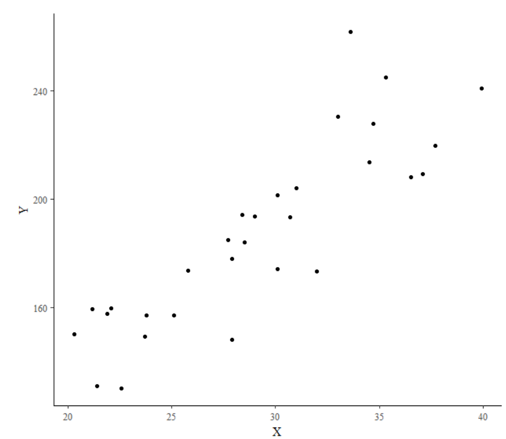
1) x と y の関係についてとりあえず線形関係を考える

- 現実世界の依存関係の多くは厳密には非線形かもしれないが、ほぼ線形で考えて問題ないことが多い (p. 434)



2) x と y の関係について確率的関係を考える

- xから予測できないyの変動が常にある
- yが予測変数の線形結合に完全に従うのではなく「ほぼ従う」と考える (p. 449)



回帰分析のモデル

√中心傾向 (μ) を x の線形結合で表現

√ μ 周辺に y が正規分布に従って発生する

解釈可能なモデル（赤字はパラメタ）

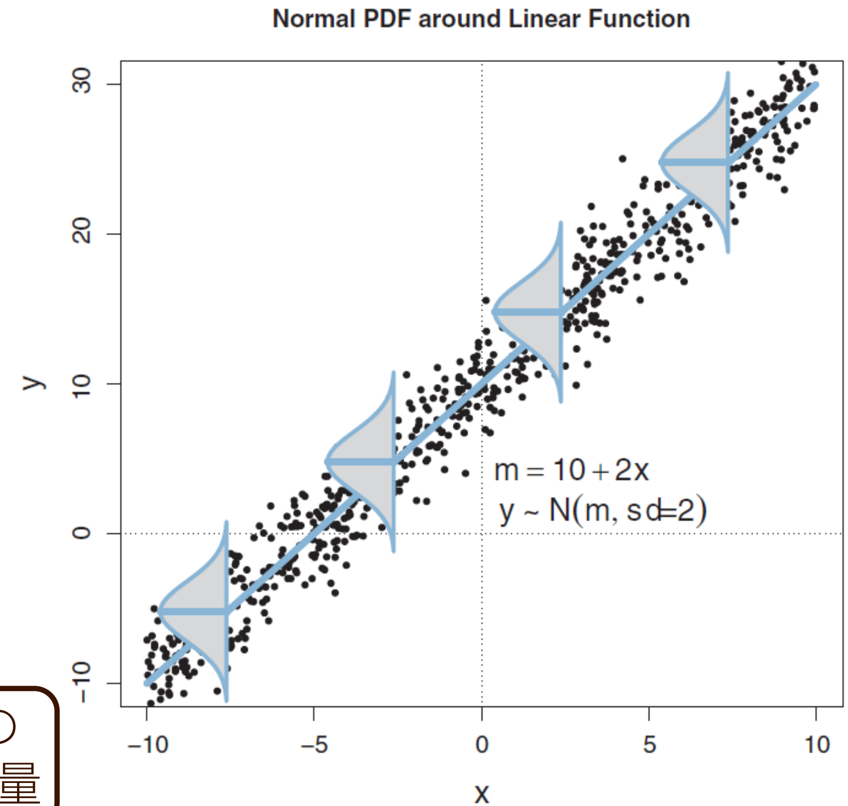
xが0のときの
予想されるyの値

$$\mu = \beta_0 + \beta_1 x$$

xが1増加したときの
予想されるyの変化量

$$y \sim normal(\mu, \sigma)$$

予想される周辺で
yが変動する程度



パラメタの事前分布の設定

無情報事前分布をつかう

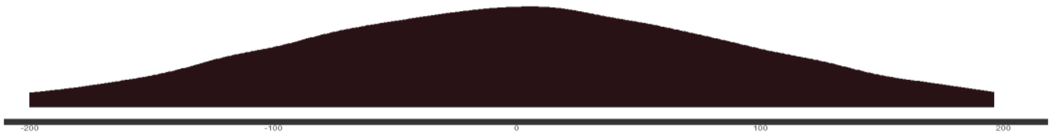
- パラメタについて事前情報がほとんどないことを示す

　＝データの情報を重視する

よく使う無情報事前分布

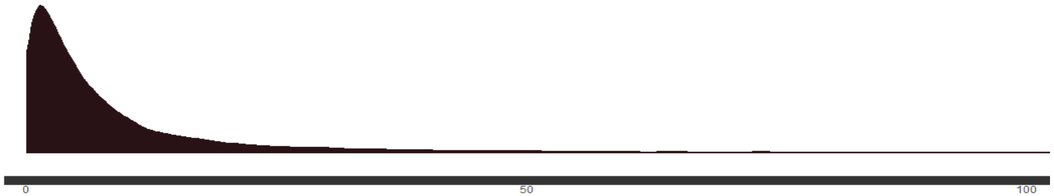
- 切片や回帰係数（-∞~∞の範囲）

$Normal(0, 100)$



- 標準偏差（0~∞の範囲）

$Cauchy(0, 5)_{I(0,\infty)}$



ベイジアン分析のステップ(p.24)

1) データの特定
2) モデルの定義　　(解釈可能な)モデルの作成
3) パラメタの事前分布の設定

4) ベイズ推論を用いて、パラメタの値に確信度を再配分

ベイズ推定

5) 事後予測がデータを模倣できているかを確認

記述的妥当性のチェック

モデルをコードに

example3.stan(*モデルブロックのみ)

モデルの記述

$\mu = \beta_0 + \beta_1 x$

$y \sim normal(\mu, \sigma)$

事前分布の設定

$\beta_0 \sim normal\ (0, 100)$

$\beta_1 \sim normal\ (0, 100)$

$\sigma \sim\ cauchy\ (0, 5)$

```
model{
  //モデルの記述
  real mu[N];
  for(n in 1:N){
    mu[n] = beta0+beta1*x[n];
    y[n] ~ normal(mu[n], sigma);
  }
  //事前分布の設定
  beta0 ~ normal(0, 100);
  beta1 ~ normal(0, 100);
  sigma ~ cauchy(0, 5);
}
```

そのまま

Stanコード解説

```
data{
  int N;//人数(整数)
  real y[N];//被予測変数(N個の配列(実数))
  real x[N];//予測変数(N個の配列(実数))
}

parameters{
  real beta0;//切片(実数)
  real beta1;//回帰係数(実数)
  real <lower=0> sigma;//正規分布の標準偏差(下限0)(実数)
}

model{
  //モデルの記述
  real mu[N];
  for(n in 1:N){
    mu[n] = beta0+beta1*x[n];
    y[n] ~ normal(mu[n], sigma);
  }
  //事前分布の設定
  beta0 ~ normal(0, 100);
  beta1 ~ normal(0, 100);
  sigma ~ cauchy(0, 5);
}
```

渡すデータを宣言

- N人分の y と x を渡すよ

モデルで使うパラメタを宣言

- $beta0, beta1, sigma$ というパラメタを使うよ
- $sigma$は正の値だよ

実行

```
#実行コード--------------
model<-stan_model("example3.stan")

data<-list(N=nrow(dat), y=dat$Y, x=dat$X)

fit<-sampling(model,
              data=data,
              chains = 4,
              iter = 2000,
              warmup = 1000,
              seed=123)
```

←先ほどのモデルをコンパイル

↑渡すデータをリスト形式で作成

←MCMCサンプリングの実行

chains:
MCMCサンプル列を何本発生させるか
Iter:
MCMCサンプルを何個だすか
warmup:
MCMCサンプルの初めのほうを何個除去するか

とりあえずみてみる

```
> fit
Inference for Stan model: reg.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

         mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
beta0   32.21    0.17 5.44  21.63  28.70  32.15  35.67  43.25  1031    1
beta1    5.42    0.01 0.17   5.08   5.31   5.42   5.53   5.76  1042    1
sigma    5.13    0.02 0.70   4.02   4.62   5.06   5.53   6.80  1488    1
lp__   -62.74    0.04 1.34 -66.27 -63.30 -62.37 -61.77 -61.24  1050    1

Samples were drawn using NUTS(diag_e) at Tue Sep 19 14:14:15 2017.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
>
```

←MCMCの設定

パラメタの要約

パラメタの事後分布をみる前に

MCMCの代表性をチェックする

チェイン内の値は事後分布を代表していなければならない。

チェインの任意の初期値に過度に影響をうけるべきでなく、

一部に留まることなく事後分布の範囲を十分に探索すべきである（p.181）

みためによるチェック：

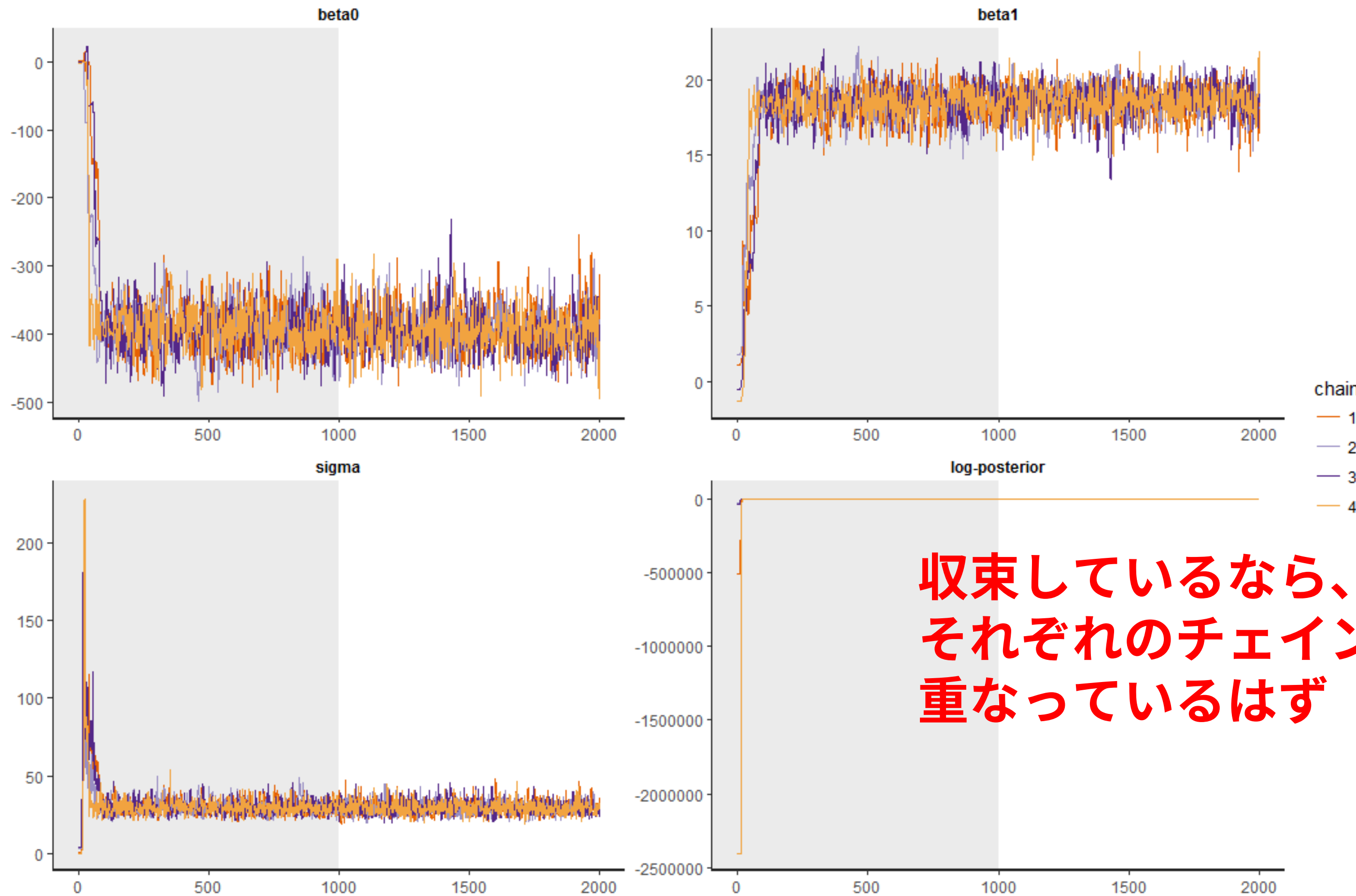
トレースプロット、確率密度プロット

数値によるチェック：

Gelman-Rubin統計量

トレースプロットの確認

```
##トレースプロット
stan_trace(fit,
           pars=c("beta0", "beta1", "sigma", "lp__"),
           inc_warmup = T)
```

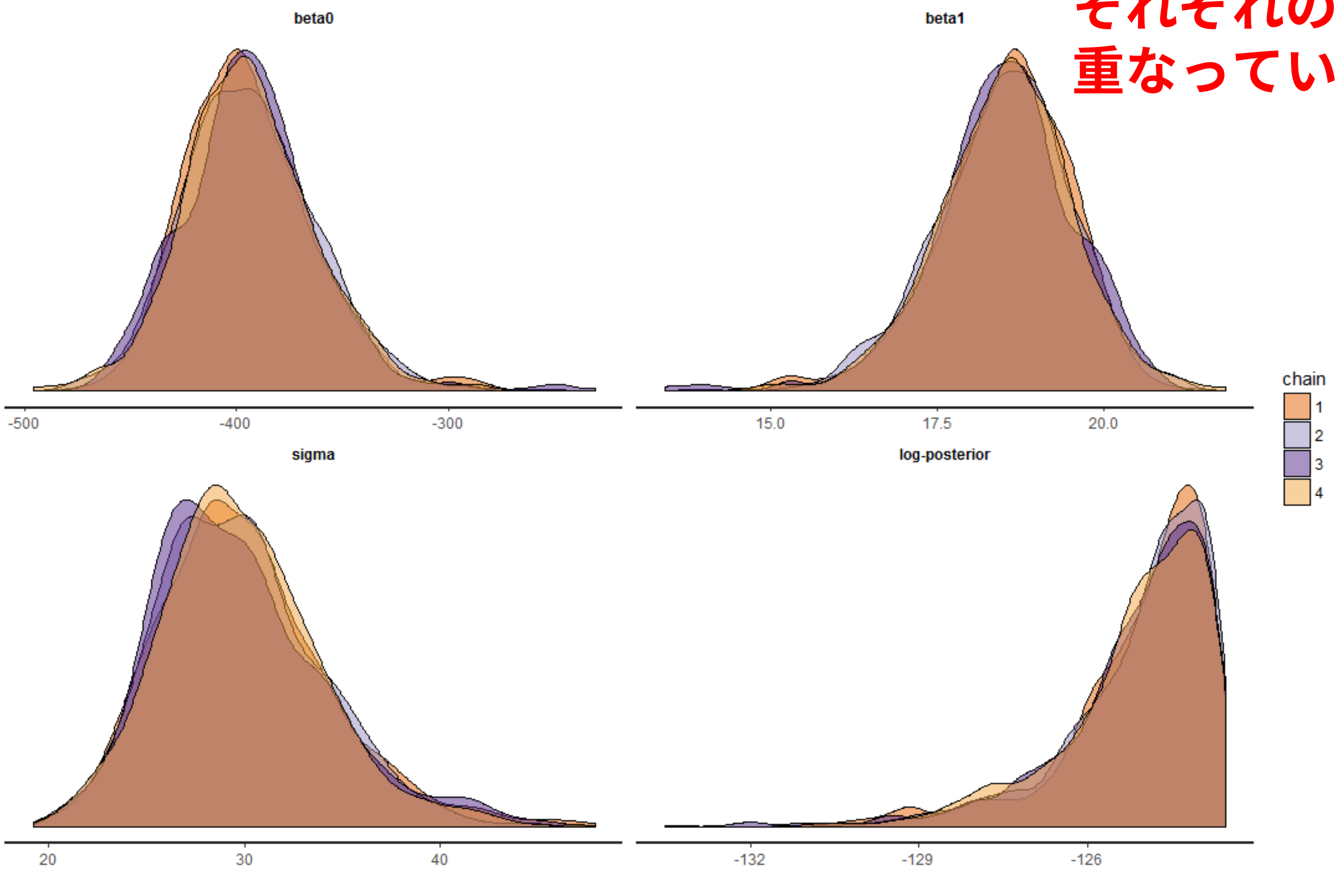


収束しているなら、
それぞれのチェインのサンプルが
重なっているはず

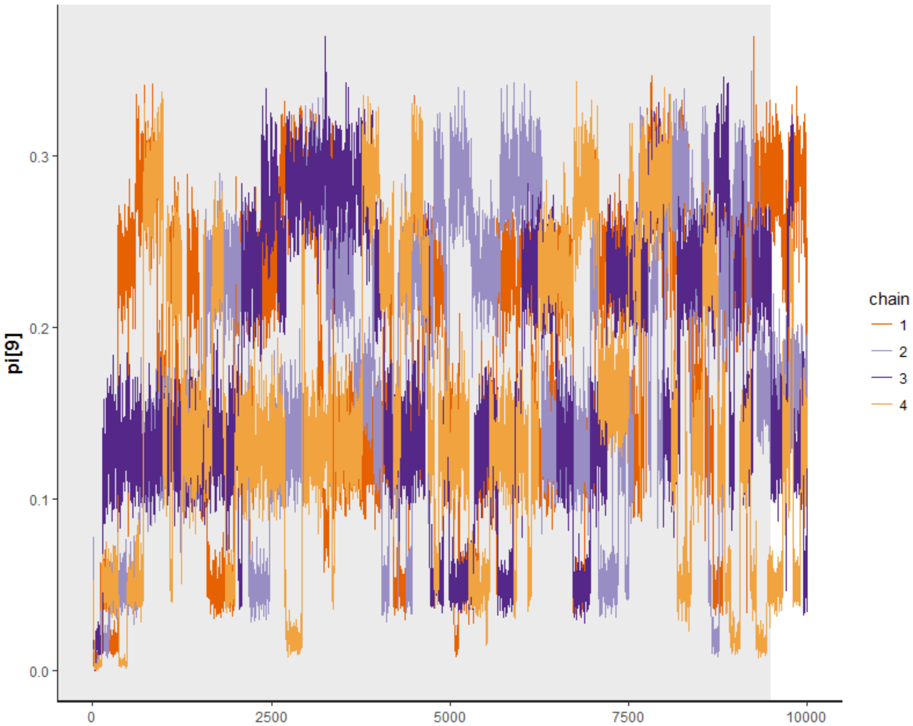
確率密度プロットの確認

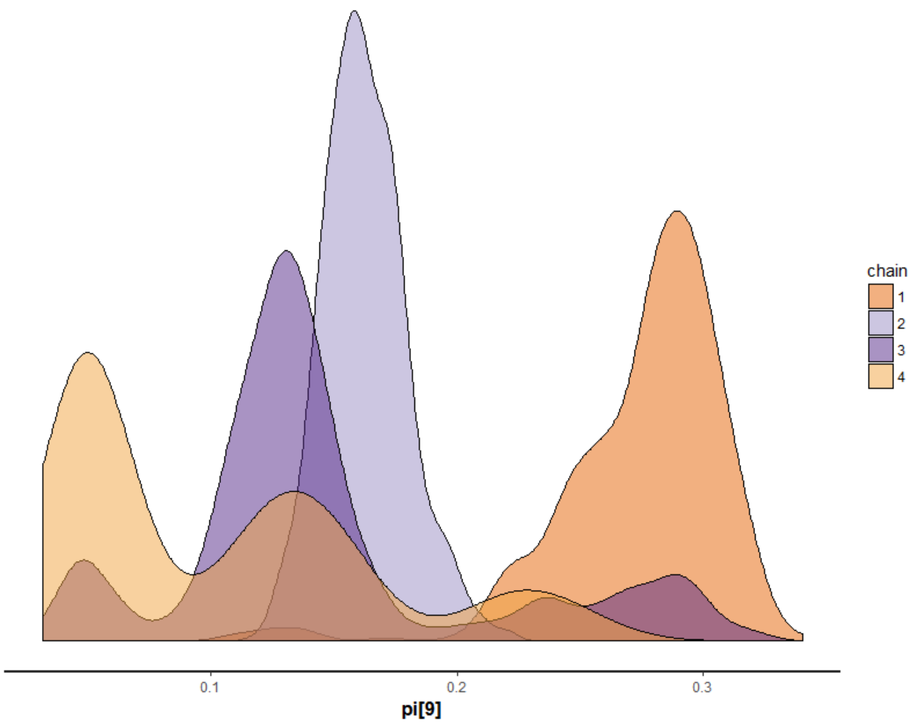
```
##確率密度プロット
stan_dens(fit,
          pars=c("beta0", "beta1", "sigma", "lp__"),
          inc_warmup = F,
          separate_chains = T)
```

収束しているなら、
それぞれのチェインのサンプルが
重なっているはず



だめなMCMC





それぞれのチェインのサンプルが重なっていない

Gelman-Rubin統計量の確認

チェイン内の分散に対してチェイン間の分散がどれくらい大きいか、の指標

完全に収束した場合に1.0となり、乖離したチェインがあれば1.0以上の値になる

```
> fit
Inference for Stan model: reg.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

         mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
beta0   32.21    0.17 5.44  21.63  28.70  32.15  35.67  43.25  1031    1
beta1    5.42    0.01 0.17   5.08   5.31   5.42   5.53   5.76  1042    1
sigma    5.13    0.02 0.70   4.02   4.62   5.06   5.53   6.80  1488    1
lp__   -62.74    0.04 1.34 -66.27 -63.30 -62.37 -61.77 -61.24  1050    1

Samples were drawn using NUTS(diag_e) at Tue Sep 19 14:14:15 2017.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
>
```

1.1以上＝NG

パラメタの事後分布をみる前に

MCMCの**正確性**をチェックする

推定を正確で安定したものとするために、チェインは十分なサイズであるべきである。特に、(中央値や最頻値などの) 中心傾向の推定や95% HDIの限界は、分析を繰り返した際に大きく異なるべきではない (p. 181)

チェックする指標：

有効サンプルサイズ(ESS)、自己相関

有効サンプルサイズの確認

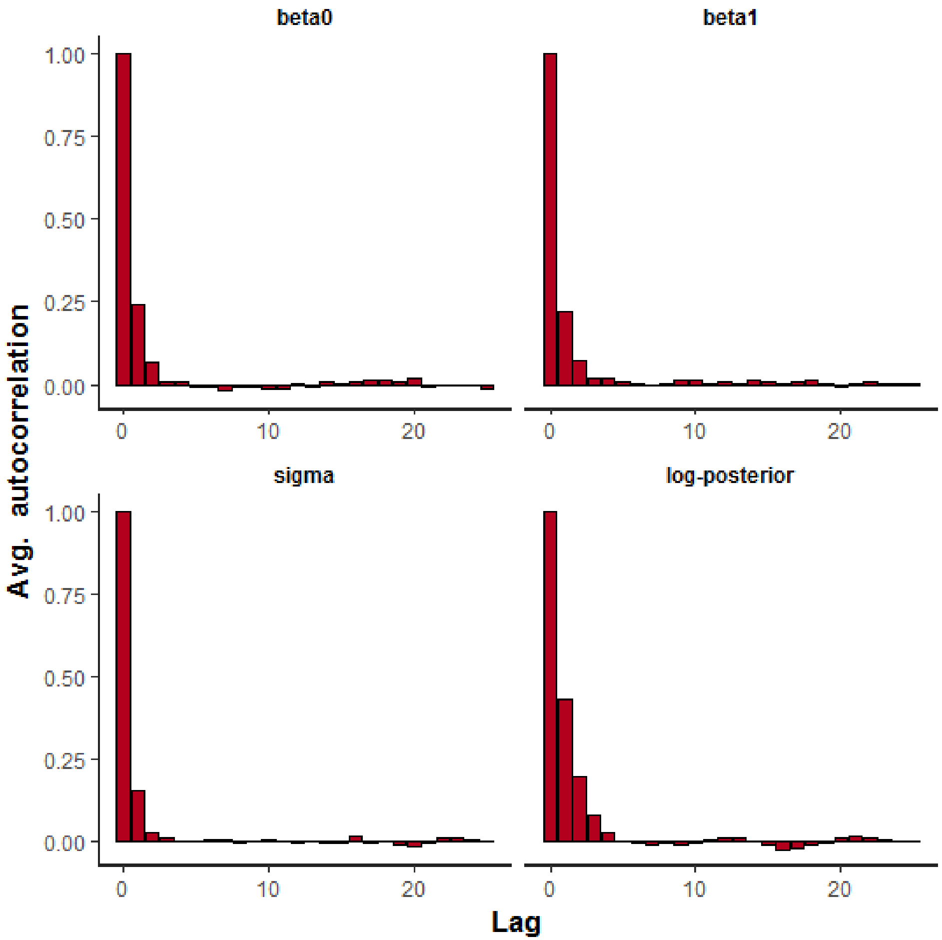
チェインの中に独立した情報がどれくらいあるかの指標

```
> fit
Inference for Stan model: reg.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

         mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
beta0   32.21    0.17 5.44  21.63  28.70  32.15  35.67  43.25  1031    1
beta1    5.42    0.01 0.17   5.08   5.31   5.42   5.53   5.76  1042    1
sigma    5.13    0.02 0.70   4.02   4.62   5.06   5.53   6.80  1488    1
lp__   -62.74    0.04 1.34 -66.27 -63.30 -62.37 -61.77 -61.24  1050    1

Samples were drawn using NUTS(diag_e) at Tue Sep 19 14:14:15 2017.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
> 
```

$$\mathrm{ESS} = N \Bigg/ \left(1 + 2\sum_{k=1}^{\infty} \mathrm{ACF}(k)\right)$$

自己相関の確認

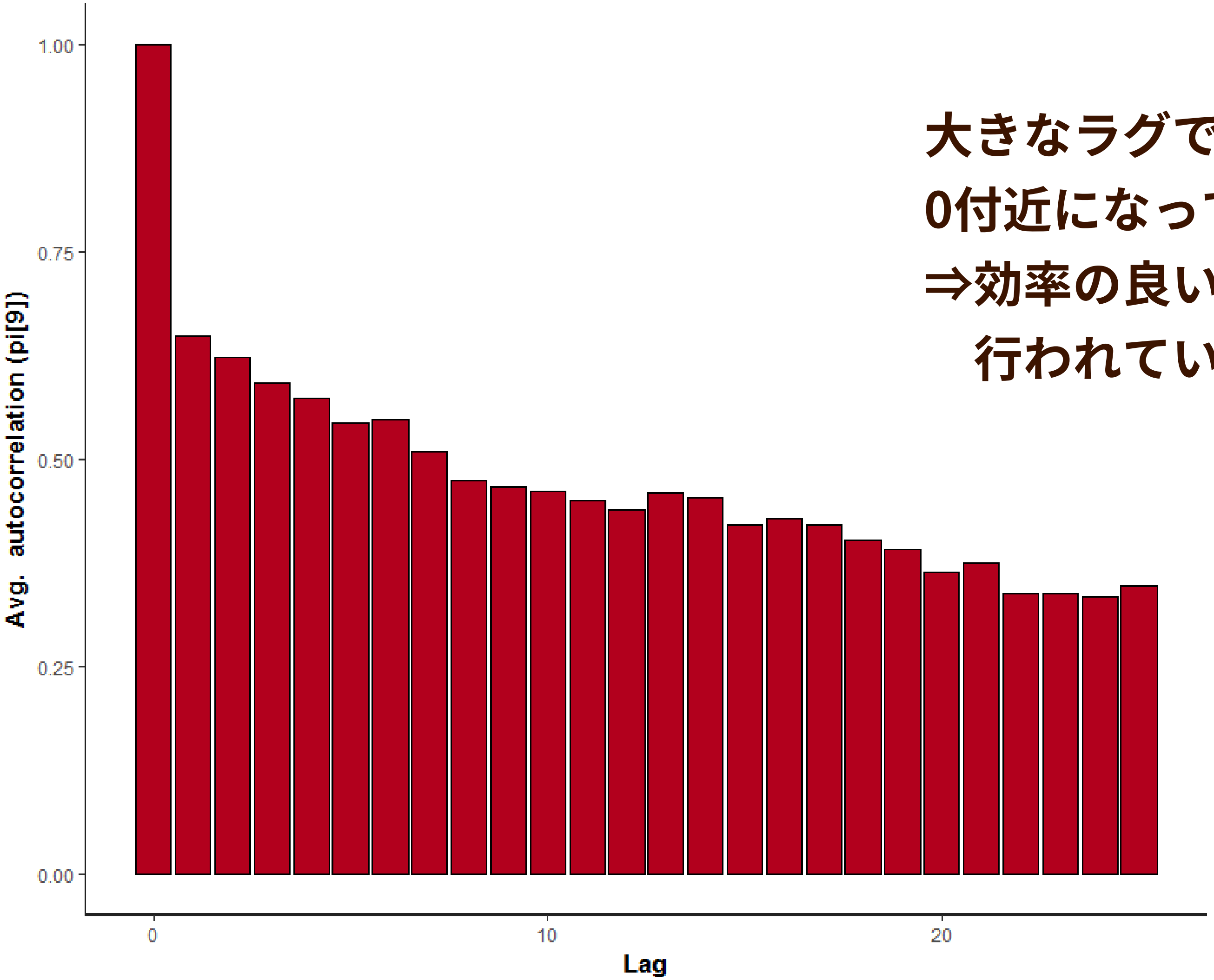
```
##自己相関
stan_ac(fit,
        pars=c("beta0", "beta1", "sigma", "lp__"),
        inc_warmup = F)
```



kステップ前の値との相関

離れているステップの値とは相関しないはず

自己相関が高い場合



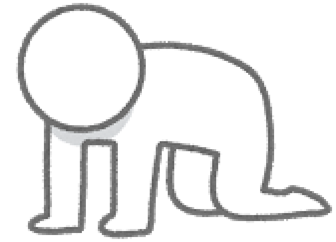
大きなラグでも相関が
0付近になっていない
⇒効率の良いサンプリングが
　行われていない可能性あり

どれくらいESSがあると良いのか

扱いたい事後分布による。(中略)

1つの簡単なガイドラインとしては、95%HDIの限界を正確で安定した妥当な推定の為に推奨されるESSの値は10,000である。これは、単に慣習上の経験に基づくヒューリスティックであり、必須のものではない。HDIの限界の正確性が実用上重要でなければ、ESSが小さくても十分である場合もある。p. 187

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
beta0	32.21	0.17	5.44	21.63	28.70	32.15	35.67	43.25	1031	1
beta1	5.42	0.01	0.17	5.08	5.31	5.42	5.53	5.76	1042	1
sigma	5.13	0.02	0.70	4.02	4.62	5.06	5.53	6.80	1488	1
lp__	-62.74	0.04	1.34	-66.27	-63.30	-62.37	-61.77	-61.24	1050	1



十分な数を得る

iter = 2000 の結果

```
> fit
Inference for Stan model: reg.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

        mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
beta0  32.21    0.17 5.44  21.63  28.70  32.15  35.67  43.25  1031    1
beta1   5.42    0.01 0.17   5.08   5.31   5.42   5.53   5.76  1042    1
sigma   5.13    0.02 0.70   4.02   4.62   5.06   5.53   6.80  1488    1
lp__  -62.74    0.04 1.34 -66.27 -63.30 -62.37 -61.77 -61.24  1050    1
```

$$MCSE = SD/\sqrt{ESS}$$

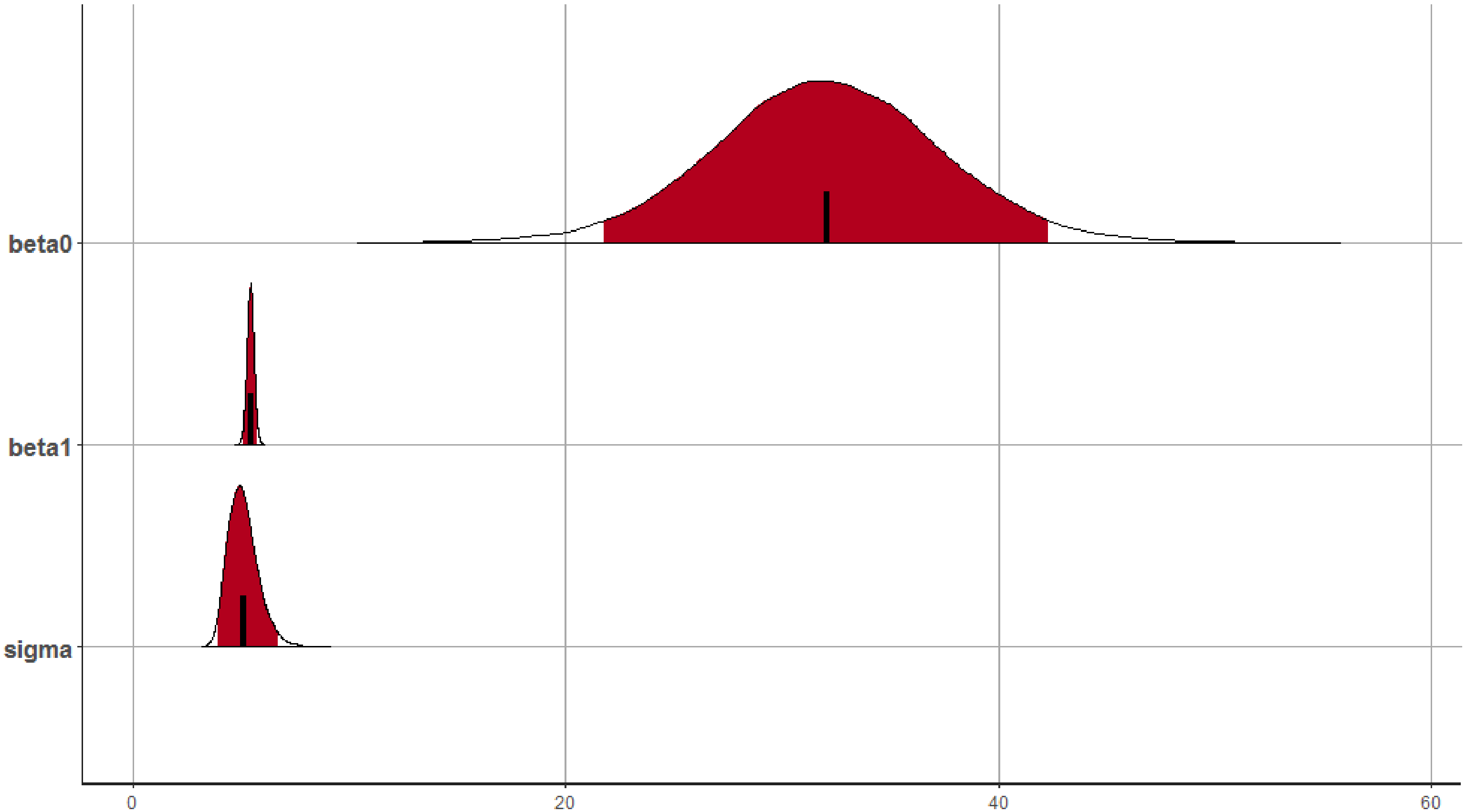
iter = 10000 の結果

```
> fit2
Inference for Stan model: reg.
4 chains, each with iter=10000; warmup=1000; thin=1;
post-warmup draws per chain=9000, total post-warmup draws=36000.

        mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
beta0  32.06    0.05 5.21  21.76  28.65  32.03  35.50  42.30 11169    1
beta1   5.43    0.00 0.16   5.11   5.32   5.43   5.53   5.75 11210    1
sigma   5.09    0.01 0.70   3.95   4.59   5.02   5.50   6.68 13906    1
lp__  -62.69    0.01 1.29 -66.04 -63.26 -62.35 -61.76 -61.24 10168    1
```

事後分布を代表するサンプルを十分な数、得ることができた

パラメタの事後分布をみる

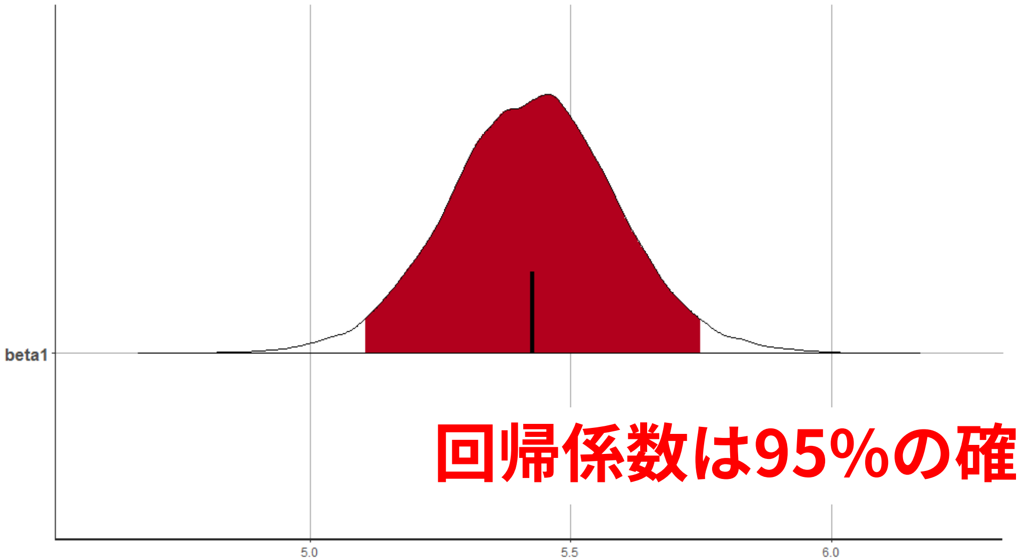
```
stan_plot(fit2,
          point_est = "mean",
          ci_level = 0.95,
          outer_level = 1.00,
          show_density = T)
```



パラメタの事後分布をみる

```r
print(fit2,
      pars=c("beta0", "beta1", "sigma"),
      probs = c(0.025, 0.975),
      digit=2)
```

```
       mean se_mean   sd  2.5% 97.5% n_eff Rhat
beta0 32.06    0.05 5.21 21.76 42.30 11169    1
beta1  5.43    0.00 0.16  5.11  5.75 11210    1
sigma  5.09    0.01 0.70  3.95  6.68 13906    1
```



回帰係数は95%の確率で5.11~5.75の範囲！！

MCMCサンプルをとりだしてみる

```r
##MCMCサンプルを取り出してみる-----
MCMC_sample <- rstan::extract(fit2)
MCMC_sample$beta1
```

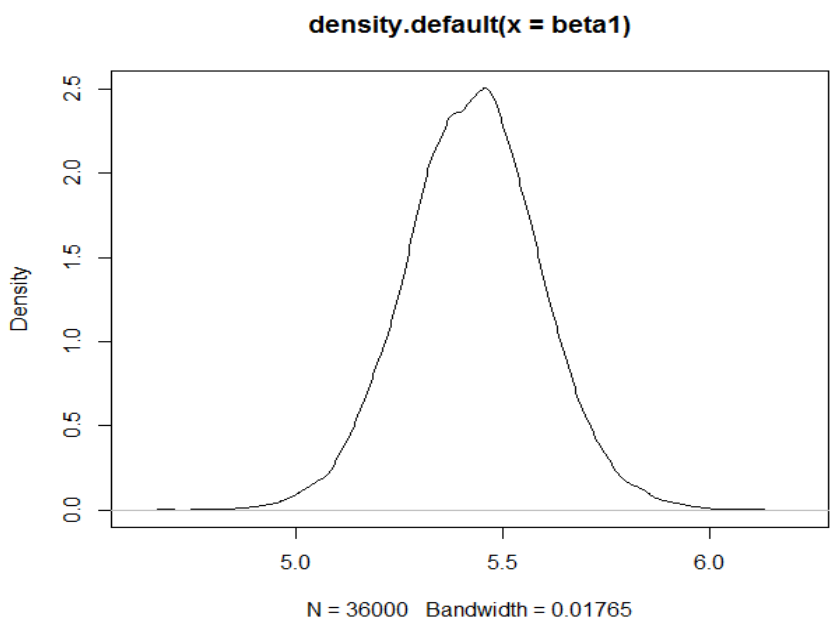
```
[9526] 5.327744 5.391297 5.586168 5.228040 5.471859 5.396478 5.394000 5.298967 5.294479 5.407322 5.313322 5.690541 5.242444 5.492485 5.442045
[9541] 5.523062 5.682807 5.403977 5.711528 5.238148 5.021745 5.267758 5.368967 5.606367 5.442238 5.484823 5.457237 5.406047 5.181691 5.221145
[9556] 5.583856 5.418137 5.346719 5.403742 5.694464 5.406850 5.197044 5.503226 5.574650 5.647990 5.588874 5.374763 5.257702 5.696040 5.684942
[9571] 5.292002 5.497614 5.287017 5.407800 5.450511 5.591223 5.341848 5.308522 5.349519 5.429579 5.488189 5.331035 5.423699 5.459995 5.270441
[9586] 5.443112 5.551823 5.357891 5.393292 5.719813 5.737147 5.423611 5.582843 5.359094 5.195234 5.448931 5.540705 5.467804 5.511697 5.569983
[9601] 5.216079 5.545751 5.253380 5.227147 5.248379 5.649595 5.310313 5.421093 5.489024 5.485219 5.685252 5.710785 5.270703 5.509406 5.260835
[9616] 5.541055 5.682500 5.472370 5.539820 5.485763 5.549721 5.216008 5.196411 5.535062 5.506433 5.378822 5.367976 5.205392 5.232530 5.193947
[9631] 5.339119 5.381403 5.340438 5.462128 5.303274 5.323616 5.630579 5.448144 5.440452 5.419324 5.491012 5.549345 5.396453 5.606915 5.090921
[9646] 5.148990 5.402892 5.372041 5.395216 5.639929 5.445213 5.583051 5.123423 5.570712 5.346336 5.525440 5.289096 5.124714 5.498917 5.326757
[9661] 5.537477 5.463039 5.158288 5.623456 5.533672 5.516235 5.459725 5.509476 5.336971 5.176089 5.364578 5.616866 5.317989 5.415214 5.211840
[9676] 5.329596 5.740530 5.086448 5.248351 5.623909 5.439858 5.323970 5.401471 5.440994 5.471716 5.466710 5.478588 5.445794 5.333914 5.545963
[9691] 5.394366 5.362815 5.275089 5.504674 5.347213 5.384583 5.382437 5.359517 5.575699 5.424004 5.631942 4.971791 5.614748 5.327558 5.488903
[9706] 5.636495 5.411853 5.023058 5.427178 5.522153 5.115228 5.256378 5.594736 5.457374 5.379871 5.329675 5.569590 5.409419 5.320595 5.414298
[9721] 5.624036 5.473210 5.284501 5.448425 5.604199 5.380861 5.336508 5.404221 5.392213 5.689517 5.158517 5.612915 5.261411 5.623937 5.571804
[9736] 5.415824 5.586629 5.250031 5.475339 5.350601 5.355794 5.497493 5.402543 5.412105 5.165743 5.586504 5.296861 5.425583 5.464363 5.634242
[9751] 5.457049 5.218538 5.347443 5.444074 5.571790 5.259725 5.321866 5.541890 5.166652 5.535494 5.342726 5.540269 5.659826 5.423026 5.380296
[9766] 5.327957 5.500350 5.564712 5.535970 5.332730 5.330612 5.372054 5.470599 5.415579 5.311598 5.800682 5.295659 5.395054 5.450714 5.433114
[9781] 5.305030 5.426745 5.371906 5.427851 5.488430 5.158702 5.351487 5.220493 5.598603 5.041680 5.501504 5.451044 5.264861 5.418514 5.681993
[9796] 5.322028 5.576613 5.659707 5.391264 5.304711 5.195193 5.396485 5.355377 5.334470 5.506894 5.400300 5.391822 5.531689 5.466296 5.477733
[9811] 5.524075 5.592017 5.495238 5.273194 5.541097 5.487424 5.432141 5.634036 5.827715 5.419672 5.464317 5.254784 5.374201 5.299846 5.628546
[9826] 5.197037 5.519877 5.481440 5.311706 5.789387 4.884573 5.271566 5.451288 5.567992 5.401861 5.285810 5.528694 5.515528 5.784244 5.235852
[9841] 5.427705 5.366086 5.485043 5.468828 5.048683 5.327874 5.198257 5.300303 5.203722 5.604651 5.232679 5.270381 5.843844 5.376533 5.586029
[9856] 5.252775 5.521540 5.410832 5.518144 5.355337 5.500081 5.489055 5.263331 5.372635 5.483577 5.387697 5.474512 5.789923 5.508983 5.547019
[9871] 5.532555 5.504337 5.323353 5.714781 5.427752 5.374196 5.429243 5.521466 5.320753 5.516128 5.462477 5.226265 5.346915 5.403695 5.655206
[9886] 5.744851 5.515232 5.417890 5.457810 5.277921 5.407784 5.322643 5.427137 5.191031 5.561998 5.410702 5.612071 5.364951 5.361724 5.556714
[9901] 5.405357 5.214694 5.215783 5.276046 5.122227 5.609517 5.339299 5.505984 5.051742 5.309609 5.577188 5.100382 5.210931 5.338170 5.242857
[9916] 5.494320 5.279839 5.420686 5.467431 5.226570 5.342503 5.173700 5.474954 5.120192 5.388047 5.071685 5.558050 5.497036 5.433397 5.807631
[9931] 5.542141 5.468080 5.644989 5.640255 5.700194 5.470304 5.632964 5.448672 5.296340 5.495016 5.428788 5.423055 5.428169 5.288128 5.549563
[9946] 5.617458 5.291532 5.362761 5.408573 5.496875 5.338166 5.409951 5.492119 5.373749 5.303897 5.581111 5.321083 5.247136 5.557733 5.434447
[9961] 5.255783 5.318330 5.405285 5.288522 5.246765 5.430772 5.535952 5.485694 5.048507 5.220742 5.716676 5.308802 5.403158 5.511092 5.421033
[9976] 5.279330 5.619901 5.209563 5.625805 5.387189 5.577692 5.610974
[9991] 5.383406 5.442135 5.792463 5.508121 5.384278 5.265811 5.295682
 [ reached getOption("max.print") -- omitted 26000 entries ]
>
```

36000個のMCMCサンプル

MCMCサンプルをとりだしてみる

```
> beta1 <- MCMC_sample$beta1
> print(
+   quantile(beta1, probs = c(0.025, 0.975)),
+   digit = 3
+ )
 2.5% 97.5%
 5.11  5.75
```

とりだした36000個のMCMCサンプルの
2.5%タイル点と97.5%タイル点を求める
＝回帰係数の95%確信区間を求める



とりだした36000個のMCMCサンプルの
密度をプロット
＝回帰係数の事後分布を描く

MCMCサンプルを自由につかう

回帰係数が5を超える確率が知りたい！

```
> a <- 5
> sum(ifelse(beta1 > a, 1, 0))/length(beta1)
[1] 0.9942222
```

とりだした36000個のMCMCサンプルのうち
5を超えた個数を数えて、36000でわる

99.4%！！

```
> a <- 6
> sum(ifelse(beta1 > a, 1, 0))/length(beta1)
[1] 0.0004166667
```

```
> a <- 5.5
> sum(ifelse(beta1 > a, 1, 0))/length(beta1)
[1] 0.3228889
```

MCMCサンプルを自由につかう

気温が30度のときの客数の95%範囲が知りたい！

```
> beta0<-MCMC_sample$beta0
> beta1<-MCMC_sample$beta1
> sigma<-MCMC_sample$sigma
>
> x<-30
> y<-rnorm(n = 36000,
+          mean = beta0 + beta1*x,
+          sd = sigma)
>
> round(quantile(y, probs=c(0.025, 0.975)))
 2.5% 97.5%
  185   205
> |
```

各パラメタのMCMCサンプルをとりだして格納

回帰モデルに従ったｙ (乱数) を36000個発生

発生させたｙの2.5%, 97.5%
タイル点をもとめる

185人～205人！！

↓発生させたｙ(36000個の一部)

```
[9796] 197.1240 188.6008 186.0957 198.9072 194.2556 197.3051
[9811] 195.6576 194.9754 202.9250 196.9239 188.3343 192.4133
[9826] 192.1186 196.4582 200.4046 200.3447 197.9343 192.3935
[9841] 197.2240 189.7236 189.3067 200.1799 184.5545 199.9318
[9856] 191.0731 197.8907 185.9613 200.5236 198.9759 194.7952
[9871] 196.0838 203.1068 200.9442 199.6343 207.9483 191.8382
[9886] 199.6671 197.0766 193.4163 186.5562 195.1757 189.1155
[9901] 202.8738 188.2106 197.6509 198.8228 183.9102 191.8424
[9916] 199.9673 195.6112 192.8284 194.2560 195.5894 194.5207
[9931] 194.7325 196.0280 198.8894 185.8372 200.1046 191.5518
[9946] 197.0613 197.3704 187.1656 195.1825 192.5136 196.9820
[9961] 192.7017 190.0687 195.9860 208.9759 203.0380 192.5204
[9976] 193.3814 195.4453 190.2607 196.1808 195.8417 188.0084
[9991] 195.7850 189.6549 190.3989 199.9954 193.1358 205.6664
[ reached getoption("max.print") -- omitted 26000 entries ]
> |
```

ベイジアン分析のステップ(p.24)

1) データの特定
2) モデルの定義 **(解釈可能な)モデルの作成**
3) パラメタの事前分布の設定

4) ベイズ推論を用いて、パラメタの値に確信度を再配分

ベイズ推定

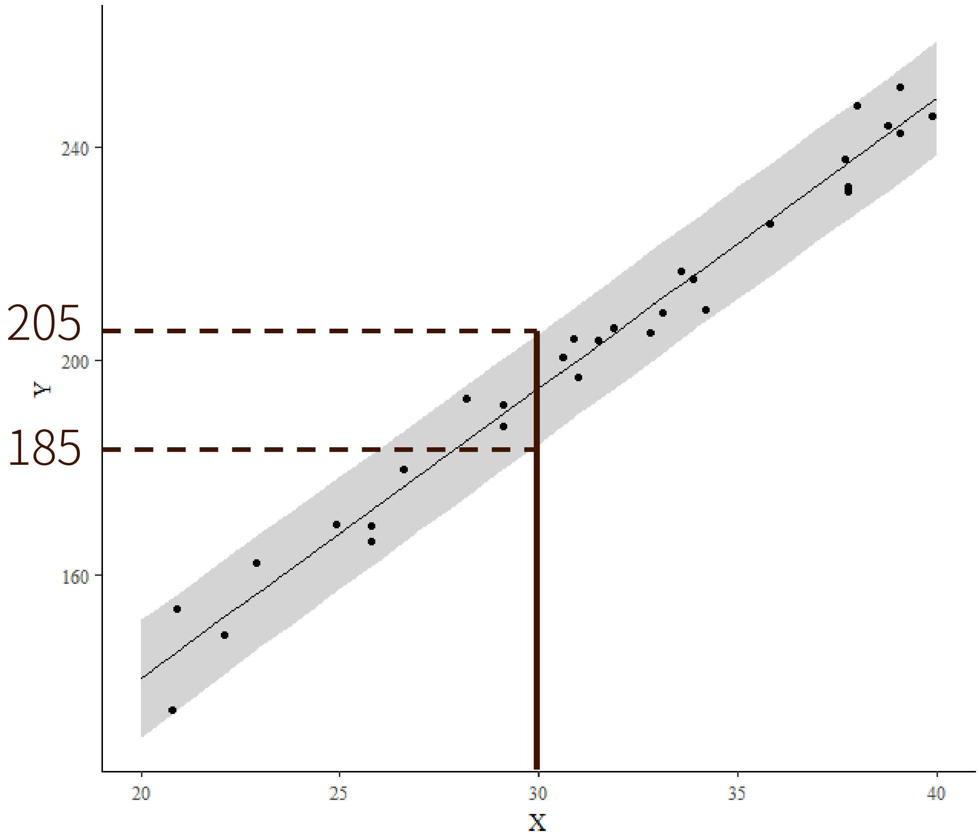
5) 事後予測がデータを模倣できているかを確認

記述的妥当性のチェック

事後予測チェック

事後予測がデータを模倣できているかを確認

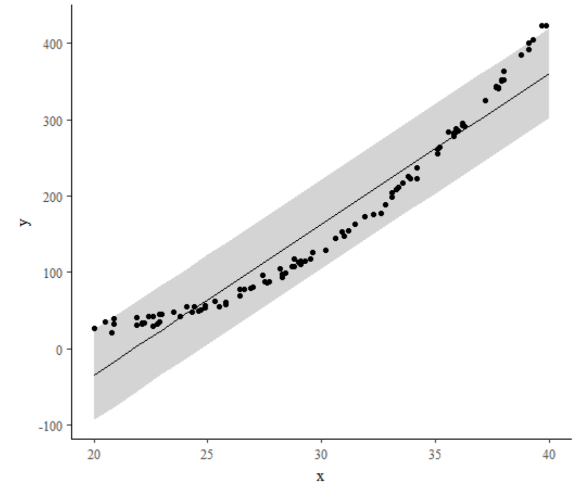
モデルから発生させたデータの95%範囲（グレー部分）とデータをプロット



事後予測分布とデータに

一貫したずれがある場合

⇒モデルを修正する必要あり



一般化線形モデル

一般化線形モデル (GLM) の枠組み

✓中心傾向 (μ) を x の線形結合で表現

✓ μ 周辺に y が「ある分布」に従って発生

e.g., 回帰分析の表現

$$\mu = \beta_0 + \beta_1 x$$

$$y \sim normal(\mu, \sigma)$$

↑

yが正規分布に従って発生すると考える

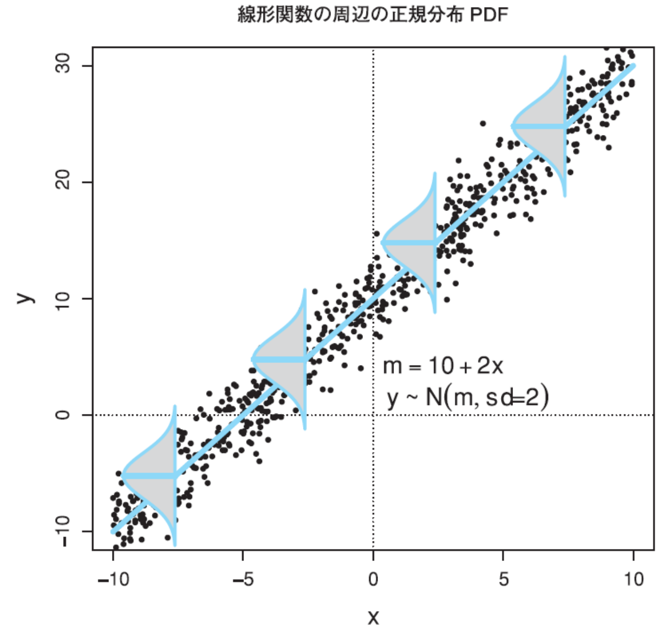


図15.9 (p. 453)

GLMの形式的表現 (p. 452)

$\mu = f(lin(x), [\text{パラメタ}])$

$y \sim \mathrm{pdf}(\mu, [\text{パラメタ}])$

✓中心傾向 (μ) を x の線形結合で表現

✓μ 周辺に y が「ある分布」に従って発生

表15. 2

被予測変数 y スケールタイプ	典型的なノイズ分布 $y \sim \mathbf{pdf}(\mu, [\text{パラメータ}])$	典型的逆リンク関数 $\mu = f(\mathbf{lin}(x), [\text{パラメータ}])$
量的	$y \sim \mathrm{normal}(\mu, \sigma)$	$\mu = \mathrm{lin}(x)$
2 値	$y \sim \mathrm{bernoulli}(\mu)$	$\mu = \mathrm{logistic}(\mathrm{lin}(x))$
名義	$y \sim \mathrm{categorical}(\ldots, \mu_k, \ldots)$	$\mu_k = \dfrac{\exp(\mathrm{lin}_k(x))}{\sum_c \exp(\mathrm{lin}_c(x))}$
順序	$y \sim \mathrm{categorical}(\ldots, \mu_k, \ldots)$	$\mu_k = \Phi((\theta_k - \mathrm{lin}(x))/\sigma) - \Phi((\theta_{k-1} - \mathrm{lin}(x))/\sigma)$
カウント	$y \sim \mathrm{poisson}(\mu)$	$\mu = \exp(\mathrm{lin}(x))$

値 μ は予測されたデータの中心傾向（平均である必要はない）．予測変数は x，$\mathrm{lin}(x)$ は表 15.1 で示されているような x の線形関数.

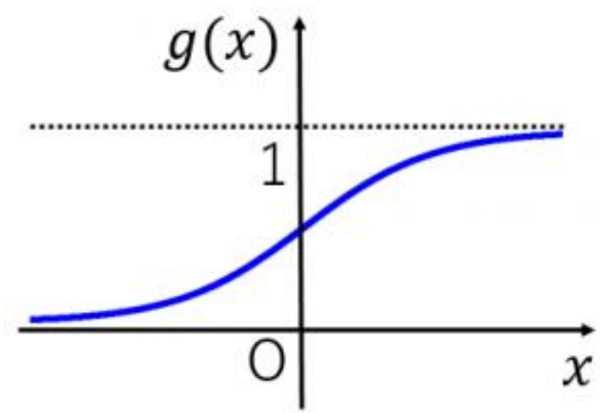
* pdf: 確率密度関数(*probability density function*)

(例) ロジスティック回帰の場合

被予測変数 y スケールタイプ	典型的なノイズ分布 $y \sim \mathrm{pdf}(\mu, [パラメータ])$	典型的逆リンク関数 $\mu = f\,(\mathrm{lin}(x), [パラメータ])$
2 値	$y \sim \mathrm{bernoulli}(\mu)$	$\mu = \mathrm{logistic}\,(\mathrm{lin}(x))$

$$\mathrm{logistic}(x) = \frac{1}{(1 + \exp(-x))}$$

```
model{
  //モデルの記述
  real mu[N];
  for(n in 1:N){
    mu[n] = inv_logit(beta0+beta1*x[n]);
    y[n] ~ bernoulli(mu[n]);
    }
  //事前分布の設定
  beta0 ~ normal(0, 100);
  beta1 ~ normal(0, 100);
}
```



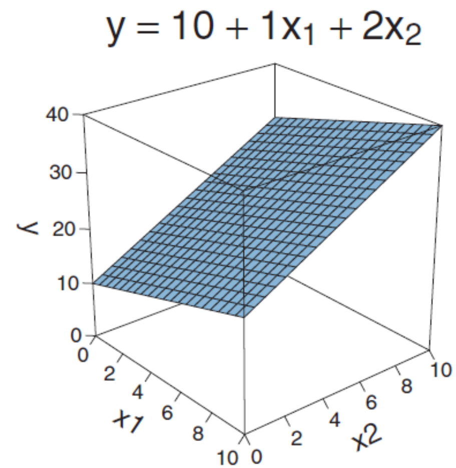
https://mathwords.net/logitkansu

線形関数の作り方

複数の x を考えたい場合：とりあえず加法結合

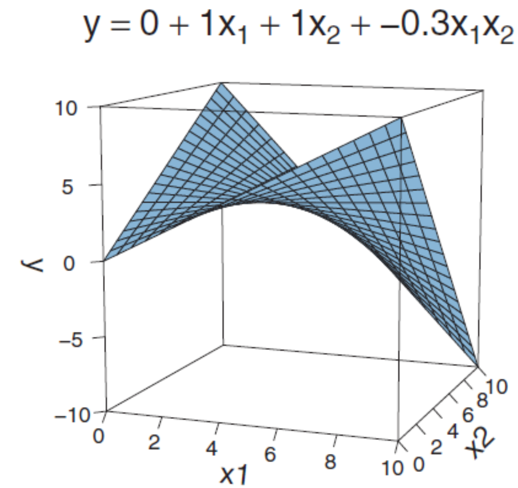
$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

$$= \beta_0 + \sum_k \beta_k x_k$$



交互作用を考えたい場合: 掛け算の項をたす

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2$$



線形関数の作り方

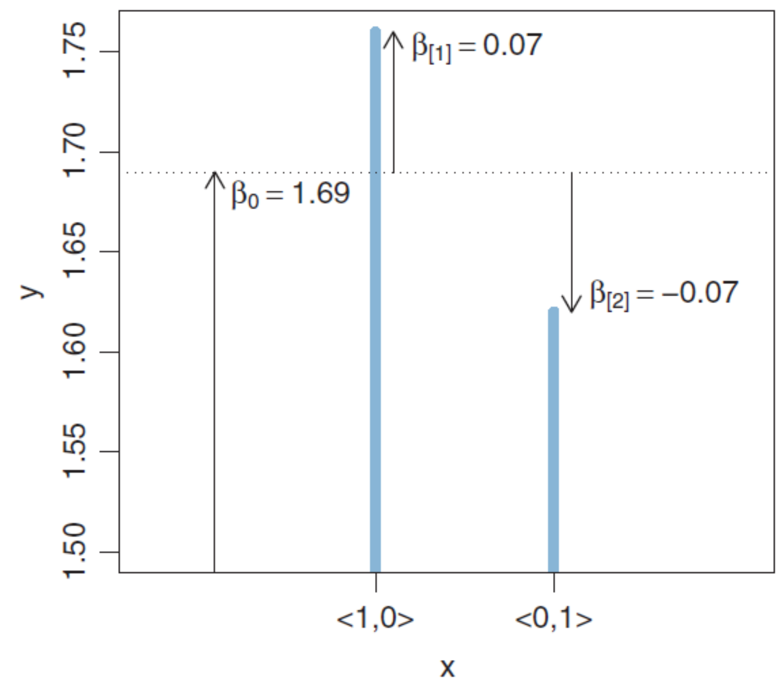
予測変数が名義変数の場合: 各水準の効果を考えるとよい

xが水準1の場合の
yの変化量
↓

$$y = \beta_0 + \beta_{[1]} x_{[1]} + \cdots + \beta_{[J]} x_{[J]}$$
$$= \beta_0 + \vec{\beta} \cdot \vec{x}$$

制約→ $$\sum_{j=1}^{J} \beta_{[j]} = 0$$

$$y = 1.69 + < 0.07, -0.07 > \cdot \vec{x}$$



予測変数の種類に応じた線形関数

表15. 1

<table>
<tr><th colspan="6">予測変数 x のスケールタイプ</th></tr>
<tr><th></th><th></th><th colspan="2">量的</th><th colspan="2">名義的</th></tr>
<tr><th>単一の群</th><th>2 つの群</th><th>単一の予測変数</th><th>複数の予測変数</th><th>単一の因子</th><th>複数の因子</th></tr>
<tr><td>β_0</td><td>$\beta_{x=1}$
$\beta_{x=2}$</td><td>β_0
$+\beta_1 x$</td><td>β_0
$+\sum_k \beta_k x_k$
$+\sum_{j,k} \beta_{j\times k} x_j x_k$
$+\left[\begin{array}{c}\text{より高次の}\\\text{交互作用}\end{array}\right]$</td><td>$\beta_0$
$+\vec{\beta}\cdot\vec{x}$</td><td>β_0
$+\sum_k \vec{\beta}_k \cdot \vec{x}_k$
$+\sum_{j,k} \vec{\beta}_{j\times k} \cdot \vec{x}_{j\times k}$
$+\left[\begin{array}{c}\text{より高次の}\\\text{交互作用}\end{array}\right]$</td></tr>
</table>

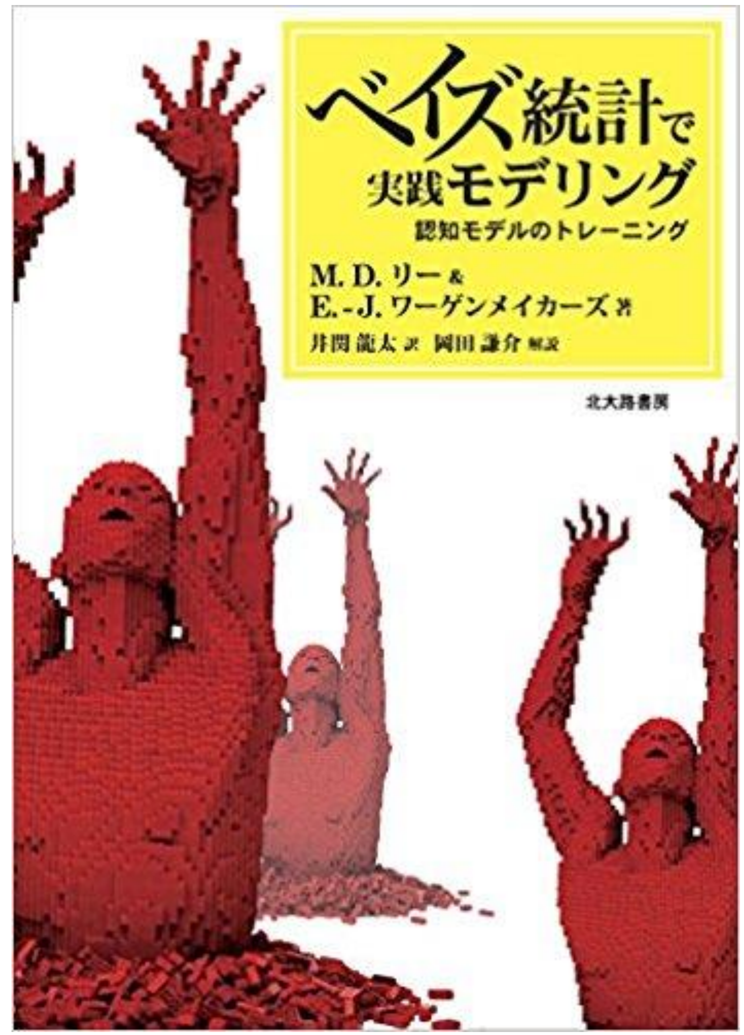
本書で対応している章

<table>
<tr><th rowspan="3">被予測変数 y のスケールタイプ</th><th colspan="6">予測変数 x のスケールタイプ</th></tr>
<tr><th colspan="2"></th><th colspan="2">量的</th><th colspan="2">名義</th></tr>
<tr><th>単一の群</th><th>2つの群</th><th>単一の予測変数</th><th>複数予測変数</th><th>単一の因子</th><th>複数の因子</th></tr>
<tr><td>量的</td><td colspan="2">第16章</td><td>第17章</td><td>第18章</td><td>第19章</td><td>第20章</td></tr>
<tr><td>2値</td><td colspan="2">第6章〜第9章</td><td colspan="4">第21章</td></tr>
<tr><td>名義</td><td colspan="6">第22章</td></tr>
<tr><td>順序</td><td colspan="6">第23章</td></tr>
<tr><td>カウント</td><td colspan="6">第24章</td></tr>
</table>

今すぐ欲しい！

Enjoy!

ベイズ統計モデリング
R, JAGS, Stanによるチュートリアル
原著第2版
著
John K. Kruschke
監訳
前田和寛 小杉考司
訳
前田和寛 小杉考司 井関龍太
井上和哉 鬼田崇作 紀ノ定保礼
国里愛彦 坂本次郎 杣取恵太
高田菜美 竹林由武 徳岡 大
難波修史 西田若葉 平川 真
福屋いずみ 武藤杏里 山根嵩史
横山仁史
Doing Bayesian Data Analysis
A Tutorial with R, JAGS, and Stan
2nd ed.
p(θ|D)
p(D|θ)
p(θ)
p(D)
共立出版



ベイズ統計で実践モデリング
認知モデルのトレーニング
M. D. リー &
E.-J. ワーゲンメイカーズ 著
井関龍太 訳 岡田謙介 解説
北大路書房