

1 Models

1.1 SVMALD

$$\begin{bmatrix} Y_{1,t+1} - Y_{1,t} \\ Y_{2,t+1} - Y_{1,t} \\ V_{1,t+1} - V_{1,t} \\ V_{2,t+1} - V_{2,t} \end{bmatrix} = \begin{bmatrix} \mu_1 \Delta \\ \mu_2 \Delta \\ \kappa_1(\theta_1 - V_{1,t})\Delta \\ \kappa_2(\theta_2 - V_{2,t})\Delta \end{bmatrix} + \sqrt{\Delta} \Sigma_t^{\frac{1}{2}} \begin{bmatrix} \epsilon_{y_1,t+1} \\ \epsilon_{y_2,t+1} \\ \epsilon_{v_1,t+1} \\ \epsilon_{v_2,t+1} \end{bmatrix} + \begin{bmatrix} N_{t+1}(M_{y_1,t+1}\xi_{y_1,t+1} + M_{c,t+1}\xi_{y_1,c,t+1}) \\ N_{t+1}(M_{y_2,t+1}\xi_{y_2,t+1} + M_{c,t+1}\xi_{y_2,c,t+1}) \\ 0 \\ 0 \end{bmatrix}, \quad (1)$$

$$\text{where } \Sigma_t = \begin{bmatrix} V_{1,t} & \rho_y \sqrt{V_{1,t}V_{2,t}} & \rho_1 \sigma_{v,1} V_{1,t} & 0 \\ \rho_y \sqrt{V_{1,t}V_{2,t}} & V_{2,t} & 0 & \rho_2 \sigma_{v,2} V_{2,t} \\ \rho_1 \sigma_{v,1} V_{1,t} & 0 & \sigma_{v,1}^2 V_{1,t} & \rho_v \sigma_{v,1} \sigma_{v,2} \sqrt{V_{1,t}V_{2,t}} \\ 0 & \rho_2 \sigma_{v,2} V_{2,t} & \rho_v \sigma_{v,1} \sigma_{v,2} \sqrt{V_{1,t}V_{2,t}} & \sigma_{v,2}^2 V_{2,t} \end{bmatrix},$$

$$\epsilon_{k,t+1} \stackrel{iid}{\sim} N(0, 1), k = \{y_1, y_2, v_1, v_2\}, \xi_{y_1,t+1} \stackrel{iid}{\sim} \mathcal{AL}_1(\mu_{y_1}, \nu_{y_1}^2), \xi_{y_2,t+1} \stackrel{iid}{\sim} \mathcal{AL}_1(\mu_{y_2}, \nu_{y_2}^2), \boldsymbol{\xi}_{c,t+1} = [\xi_{y_1,c,t+1}, \xi_{y_2,c,t+1}]' \stackrel{iid}{\sim} \mathcal{AL}_2(\boldsymbol{\mu}_c, \boldsymbol{\nu}_c), [M_{y_1,t+1}, M_{y_2,t+1}, M_{c,t+1} \sim \text{Multinomial}(1; \frac{\lambda_{y_1}}{\lambda}, \frac{\lambda_{y_2}}{\lambda}, \frac{\lambda_c}{\lambda})], \text{ and } N_{t+1} \sim \text{Bernoulli}(\lambda\Delta)$$

1.2 SVLD

$$\begin{bmatrix} Y_{1,t+1} - Y_{1,t} \\ Y_{2,t+1} - Y_{1,t} \\ V_{1,t+1} - V_{1,t} \\ V_{2,t+1} - V_{2,t} \end{bmatrix} = \begin{bmatrix} \mu_1 \Delta \\ \mu_2 \Delta \\ \kappa_1(\theta_1 - V_{1,t})\Delta \\ \kappa_2(\theta_2 - V_{2,t})\Delta \end{bmatrix} + \sqrt{\Delta} \Sigma_t^{\frac{1}{2}} \begin{bmatrix} \epsilon_{y_1,t+1} \\ \epsilon_{y_2,t+1} \\ \epsilon_{v_1,t+1} \\ \epsilon_{v_2,t+1} \end{bmatrix} + \begin{bmatrix} N_{t+1}(M_{y_1,t+1}\xi_{y_1,t+1} + M_{c,t+1}\xi_{y_1,c,t+1}) \\ N_{t+1}(M_{y_2,t+1}\xi_{y_2,t+1} + M_{c,t+1}\xi_{y_2,c,t+1}) \\ 0 \\ 0 \end{bmatrix}, \quad (2)$$

$$\text{where } \Sigma_t = \begin{bmatrix} V_{1,t} & \rho_y \sqrt{V_{1,t}V_{2,t}} & \rho_1 \sigma_{v,1} V_{1,t} & 0 \\ \rho_y \sqrt{V_{1,t}V_{2,t}} & V_{2,t} & 0 & \rho_2 \sigma_{v,2} V_{2,t} \\ \rho_1 \sigma_{v,1} V_{1,t} & 0 & \sigma_{v,1}^2 V_{1,t} & \rho_v \sigma_{v,1} \sigma_{v,2} \sqrt{V_{1,t}V_{2,t}} \\ 0 & \rho_2 \sigma_{v,2} V_{2,t} & \rho_v \sigma_{v,1} \sigma_{v,2} \sqrt{V_{1,t}V_{2,t}} & \sigma_{v,2}^2 V_{2,t} \end{bmatrix},$$

$$\epsilon_{k,t+1} \stackrel{iid}{\sim} N(0, 1), k = \{y_1, y_2, v_1, v_2\}, \xi_{y_1,t+1} \stackrel{iid}{\sim} \mathcal{AL}_1(0, \nu_{y_1}^2), \xi_{y_2,t+1} \stackrel{iid}{\sim} \mathcal{AL}_1(0, \nu_{y_2}^2), \boldsymbol{\xi}_{c,t+1} = [\xi_{y_1,c,t+1}, \xi_{y_2,c,t+1}]' \stackrel{iid}{\sim} \mathcal{AL}_2(\mathbf{0}, \boldsymbol{\nu}_c), [M_{y_1,t+1}, M_{y_2,t+1}, M_{c,t+1} \sim \text{Multinomial}(1; \frac{\lambda_{y_1}}{\lambda}, \frac{\lambda_{y_2}}{\lambda}, \frac{\lambda_c}{\lambda})], \text{ and } N_{t+1} \sim \text{Bernoulli}(\lambda\Delta)$$

1.3 SVMVN

$$\begin{bmatrix} Y_{1,t+1} - Y_{1,t} \\ Y_{2,t+1} - Y_{1,t} \\ V_{1,t+1} - V_{1,t} \\ V_{2,t+1} - V_{2,t} \end{bmatrix} = \begin{bmatrix} \mu_1 \Delta \\ \mu_2 \Delta \\ \kappa_1(\theta_1 - V_{1,t})\Delta \\ \kappa_2(\theta_2 - V_{2,t})\Delta \end{bmatrix} + \sqrt{\Delta} \Sigma_t^{\frac{1}{2}} \begin{bmatrix} \epsilon_{y_1,t+1} \\ \epsilon_{y_2,t+1} \\ \epsilon_{v_1,t+1} \\ \epsilon_{v_2,t+1} \end{bmatrix} + \begin{bmatrix} N_{t+1}(M_{y_1,t+1}\xi_{y_1,t+1} + M_{c,t+1}\xi_{y_1,c,t+1}) \\ N_{t+1}(M_{y_2,t+1}\xi_{y_2,t+1} + M_{c,t+1}\xi_{y_2,c,t+1}) \\ 0 \\ 0 \end{bmatrix}, \quad (3)$$

$$\text{where } \Sigma_t = \begin{bmatrix} V_{1,t} & \rho_y \sqrt{V_{1,t}V_{2,t}} & \rho_1 \sigma_{v,1} V_{1,t} & 0 \\ \rho_y \sqrt{V_{1,t}V_{2,t}} & V_{2,t} & 0 & \rho_2 \sigma_{v,2} V_{2,t} \\ \rho_1 \sigma_{v,1} V_{1,t} & 0 & \sigma_{v,1}^2 V_{1,t} & \rho_v \sigma_{v,1} \sigma_{v,2} \sqrt{V_{1,t}V_{2,t}} \\ 0 & \rho_2 \sigma_{v,2} V_{2,t} & \rho_v \sigma_{v,1} \sigma_{v,2} \sqrt{V_{1,t}V_{2,t}} & \sigma_{v,2}^2 V_{2,t} \end{bmatrix},$$

$$\epsilon_{k,t+1} \stackrel{iid}{\sim} N(0, 1), k = \{y_1, y_2, v_1, v_2\}, \xi_{y_1,t+1} \stackrel{iid}{\sim} N(\mu_{y_1}, \nu_{y_1}^2), \xi_{y_2,t+1} \stackrel{iid}{\sim} N(\mu_{y_2}, \nu_{y_2}^2), \boldsymbol{\xi}_{c,t+1} = [\xi_{y_1,c,t+1}, \xi_{y_2,c,t+1}]' \stackrel{iid}{\sim} MVN(\boldsymbol{\mu}_c, \boldsymbol{\nu}_c), [M_{y_1,t+1}, M_{y_2,t+1}, M_{c,t+1} \sim \text{Multinomial}(1; \frac{\lambda_{y_1}}{\lambda}, \frac{\lambda_{y_2}}{\lambda}, \frac{\lambda_c}{\lambda})], \text{ and } N_{t+1} \sim \text{Bernoulli}(\lambda\Delta)$$

1.4 SVIND

$$\begin{bmatrix} Y_{t+1} - Y_t \\ V_{t+1} - V_t \end{bmatrix} = \begin{bmatrix} \mu \Delta \\ \kappa(\theta - V_t)\Delta \end{bmatrix} + \sqrt{V_t} \Delta \begin{bmatrix} 1 & 0 \\ \rho_{\sigma_v} & \sqrt{1 - \rho^2} \sigma_v \end{bmatrix} \begin{bmatrix} \epsilon_{y,t+1} \\ \epsilon_{v,t+1} \end{bmatrix} + \begin{bmatrix} N_{y,t+1} \xi_{y,t+1} \\ 0 \end{bmatrix}, \quad (4)$$

$$\epsilon_{k,t+1} \stackrel{iid}{\sim} N(0, 1), k = \{y, v\}, \xi_{y,t+1} \stackrel{iid}{\sim} \mathcal{AL}_1(\mu_y, \nu_y^2), N_{y,t+1} \sim \text{Bernoulli}(\lambda\Delta)$$

2 Empirical study: S&P and BTC jointly

2.1 Results: Joint Jumps

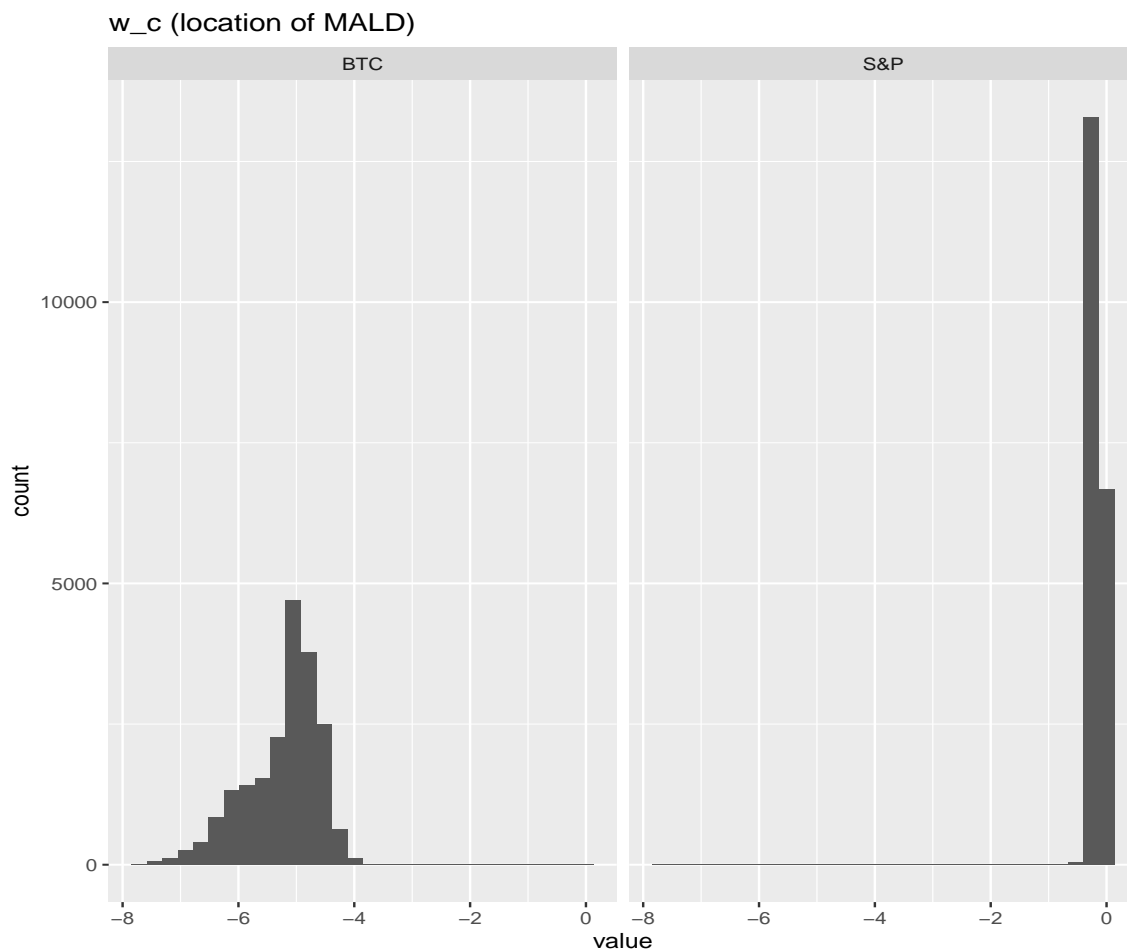
```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##      filter, lag
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
## List of 13
## $ lambda : num [1:4] 0.097 0.0367 0.085 0.7813
## $ sigma_v : num [1:2] 0.436 0.212
## $ sigma_c : num [1:2] 1.074 0.174
## $ rhoc : num -0.104
## $ xi_cw : num [1:2] -5.236 -0.166
## $ xi_y1eta: num 0.839
## $ xi_y1w : num 4.91
## $ xi_y2eta: num 0.347
## $ xi_y2w : num -0.668
## $ phi : num [1:2] 0.985 0.976
## $ theta : num [1:2] 6.96 1.12
## $ mu : num [1:2] 0.165 0.0547
## $ rho : num [1:4] 0.00665 0.07259 0.45863 -0.70195
```

#Related to joint jumps:

```
keepsBTCSP$xi_cw %>%  
  as.data.frame() %>%  
  melt() %>%  
  mutate(variable = factor(variable, levels = c("V1", "V2"), labels = c("BTC", "S&P"))) %>%  
  ggplot() +  
  geom_histogram(aes(x = value)) +  
  facet_grid(~variable)+  
  ggtitle("w_c (location of MALD)")
```

No id variables; using all as measure variables

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.



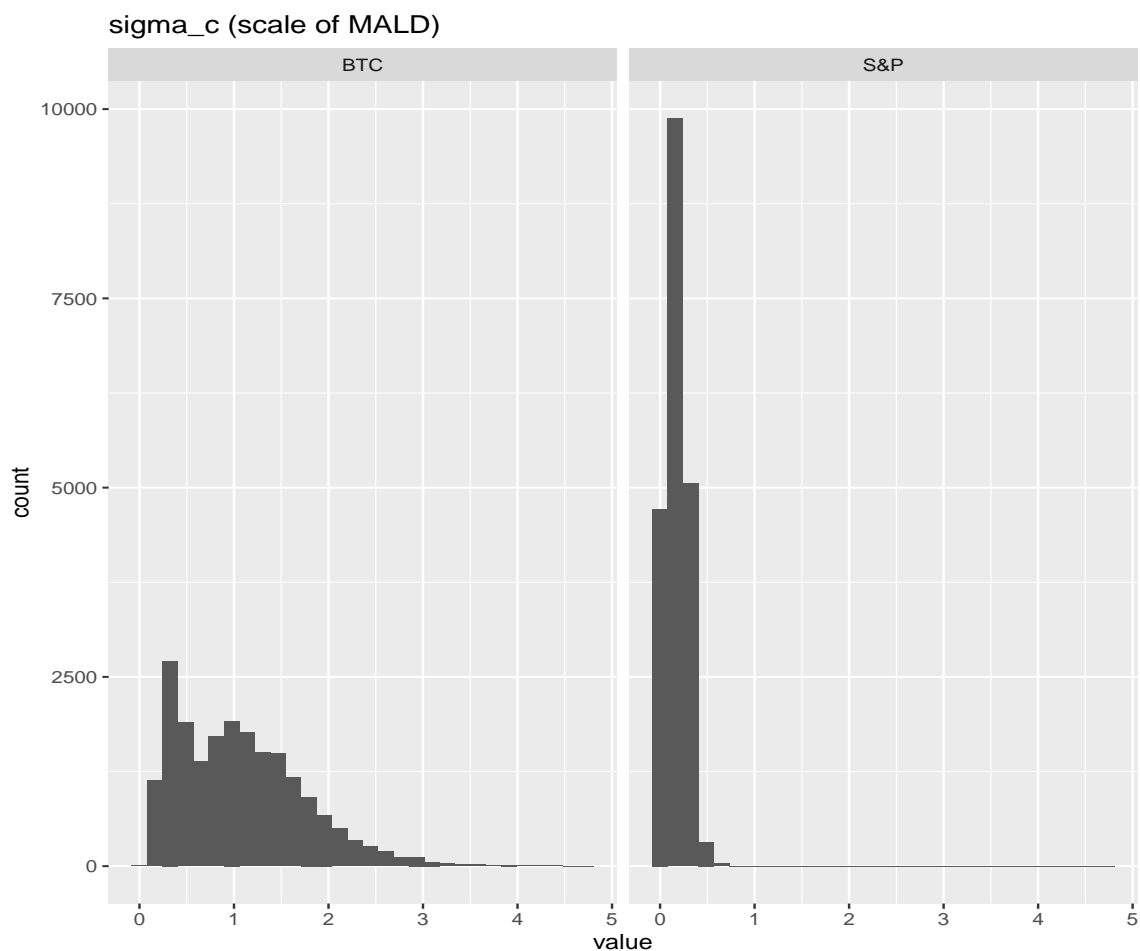
```

keepsBTCSP$sigma_c %>%
  as.data.frame() %>%
  melt() %>%
  mutate(variable = factor(variable, levels = c("V1", "V2"), labels = c("BTC", "S&P"))) %>%
  ggplot() +
  geom_histogram(aes(x = value)) +
  facet_grid(~variable) +
  ggtitle("sigma_c (scale of MALD)")

```

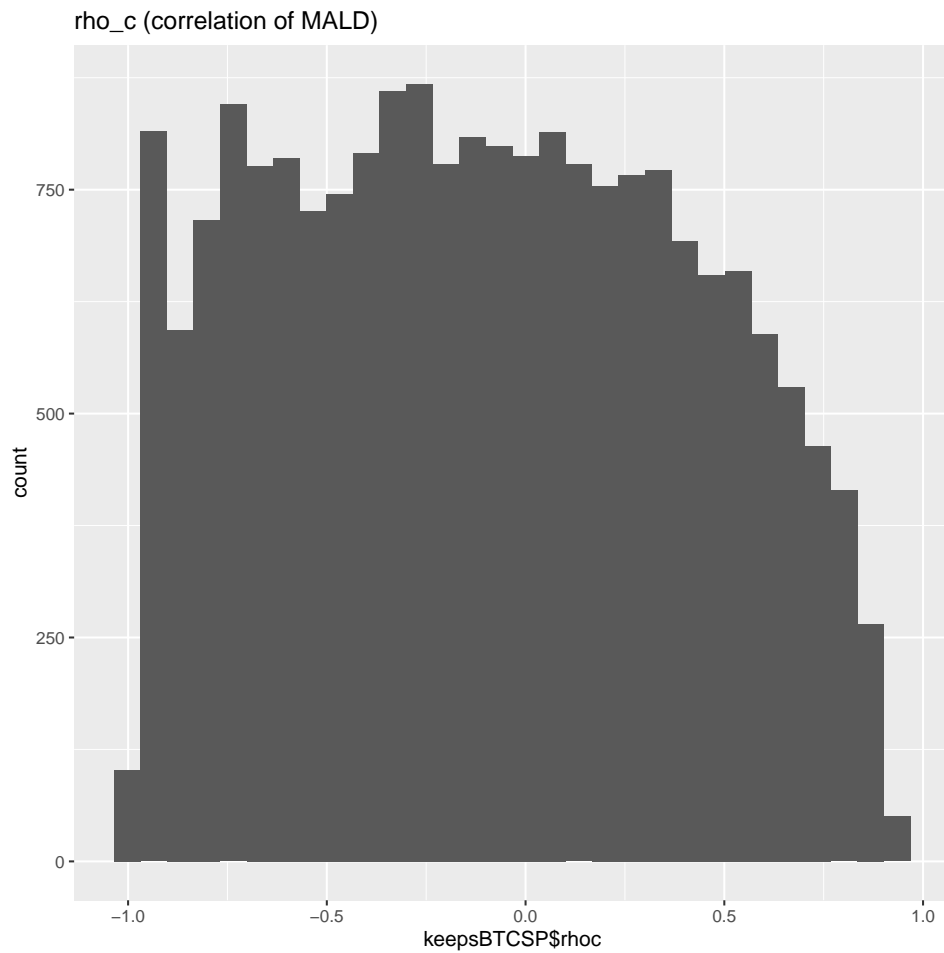
No id variables; using all as measure variables

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.



```
ggplot() +  
  geom_histogram(aes(x = keepsBTCSP$rhoc)) +  
  ggtitle("rho_c (correlation of MALD)")
```

'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.

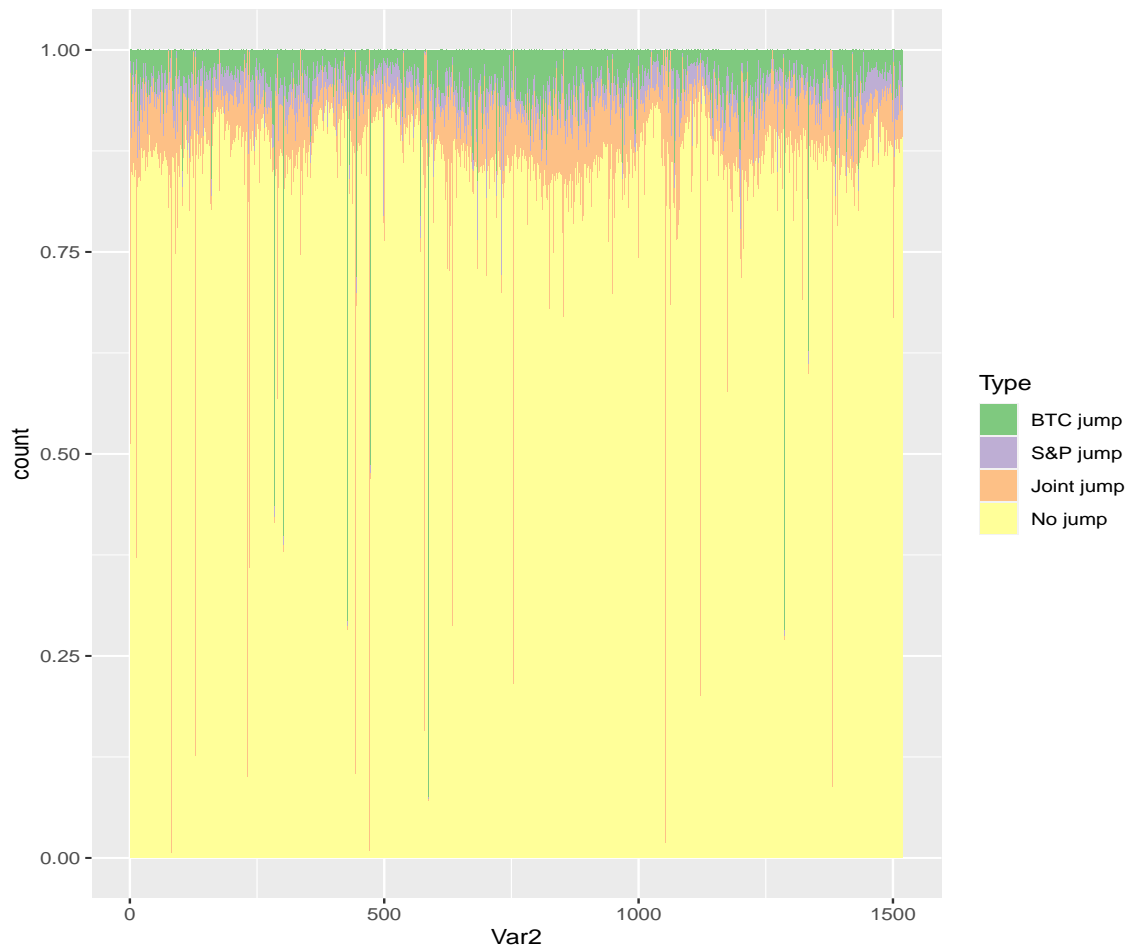


```

keepsBTCSP$delta %>%
  melt() %>%
  mutate(value = factor(value, levels = c(0:3),
                        labels = c("BTC jump", "S&P jump", "Joint jump", "No jump"))) %>%

  ggplot() +
  geom_bar(aes(x = Var2, fill = value), position = "fill") +
  scale_fill_brewer("Type", type = "qual")

```



#it's worth noting the MVN plot looks almost identical - same patterns

	SVMALD	SVLD	SVMVN	SVIND
Diffuse priors	9813.312	9796.879	9818.368	10953.52
Fulop-style priors	9827.731		9714.333	10884.23

Table 1: Resulting DIC from runs with diffuse priors and runs with more informative priors.

2.2 DIC

$$DIC_7 = -4E_{\theta, \mathbf{z} | \mathbf{y}}(\ln(p(\mathbf{y} | \theta, \mathbf{z}))) + 2\ln(p(\mathbf{y} | \hat{\theta}, \hat{\mathbf{z}})).$$

2.3 Posterior predictive p-values

First, a lineup to motivate. Posterior predictive p-values operate on the assumption that $p(y^* | Y) = \int p(y^* | \theta)p(\theta | Y)d\theta$ should be consistent with what we observed in Y .

