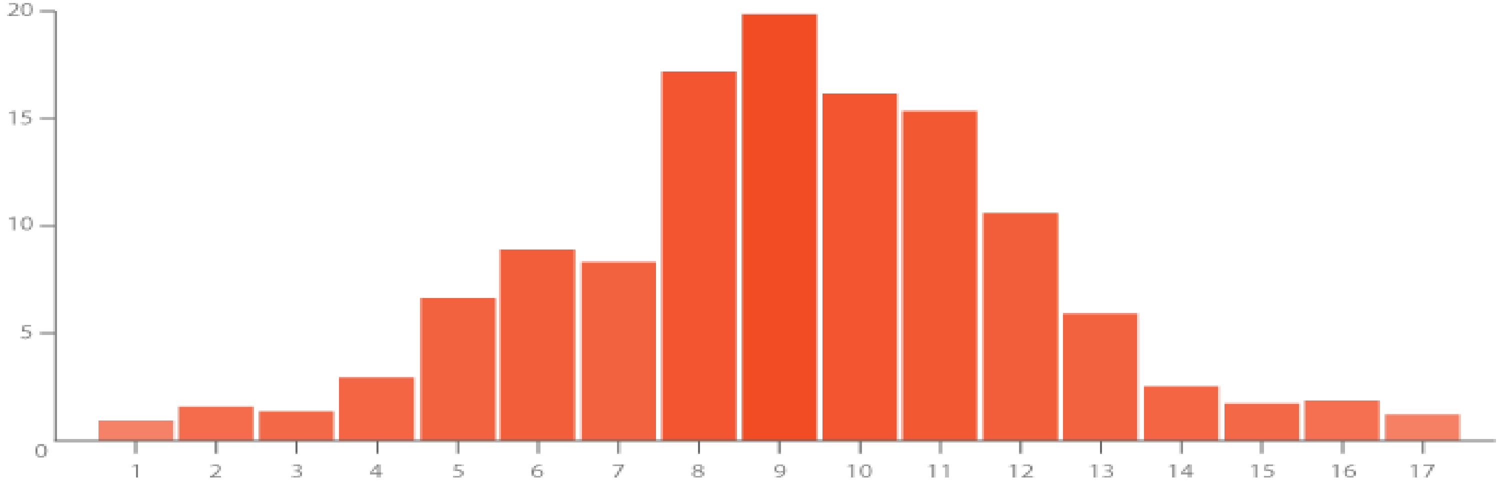
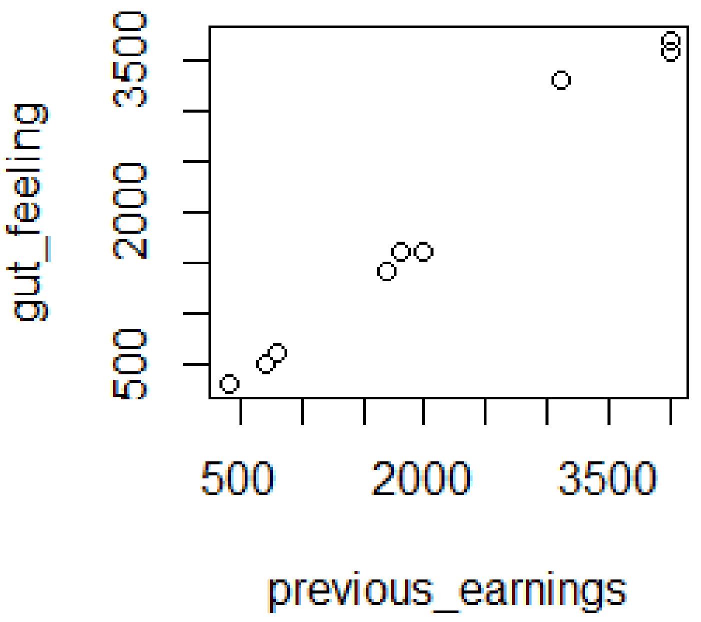
EXTENDING BAYES TO MAKE OPTIMAL DECISIONS

Jonas Kristoffer Lindeløv

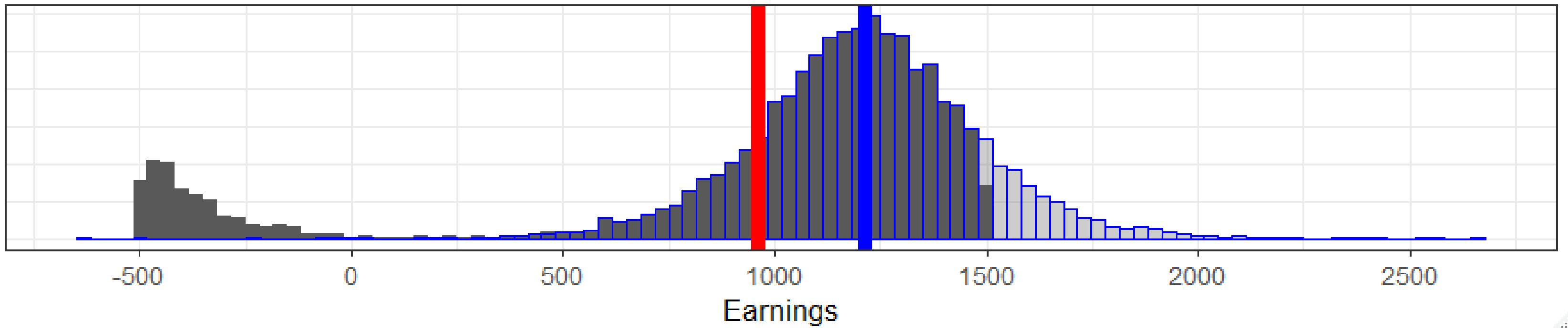
Email: jonas@hum.aau.dk
Twitter: @jonaslindeloev
Blog: lindeloev.net

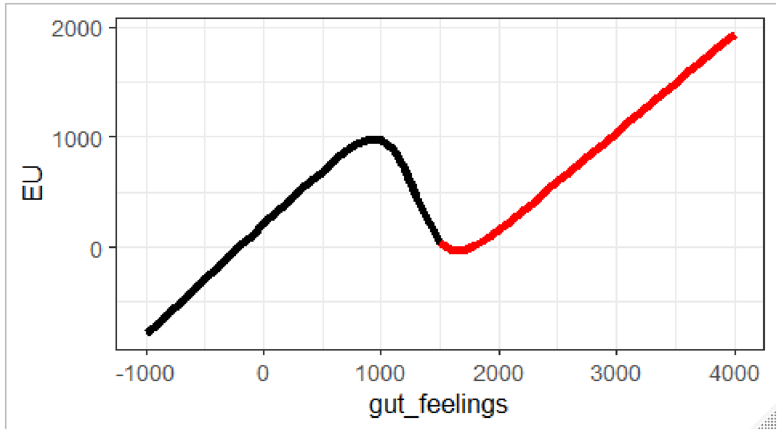

20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

EXAMPLE: EVADING TAXES




```
# Infer how parameter(s) predict 'previous_earnings'
fit = stan_glm(previous_earnings ~ gut_feeling, ...)

# Prediction distribution for a particular predictor value
predictions = posterior_predict(fit, data.frame(gut_feeling=1000))

# Utility function; abrupt change at x > 1500
utility = function(x) ifelse(x < 1500, x, x*0.9 - 18500*0.1)

# Plot the utility distribution, the Expected Utility (EU), and predictions
plot(density(utility(predictions)))
abline(v = mean(utility(predictions)))
```

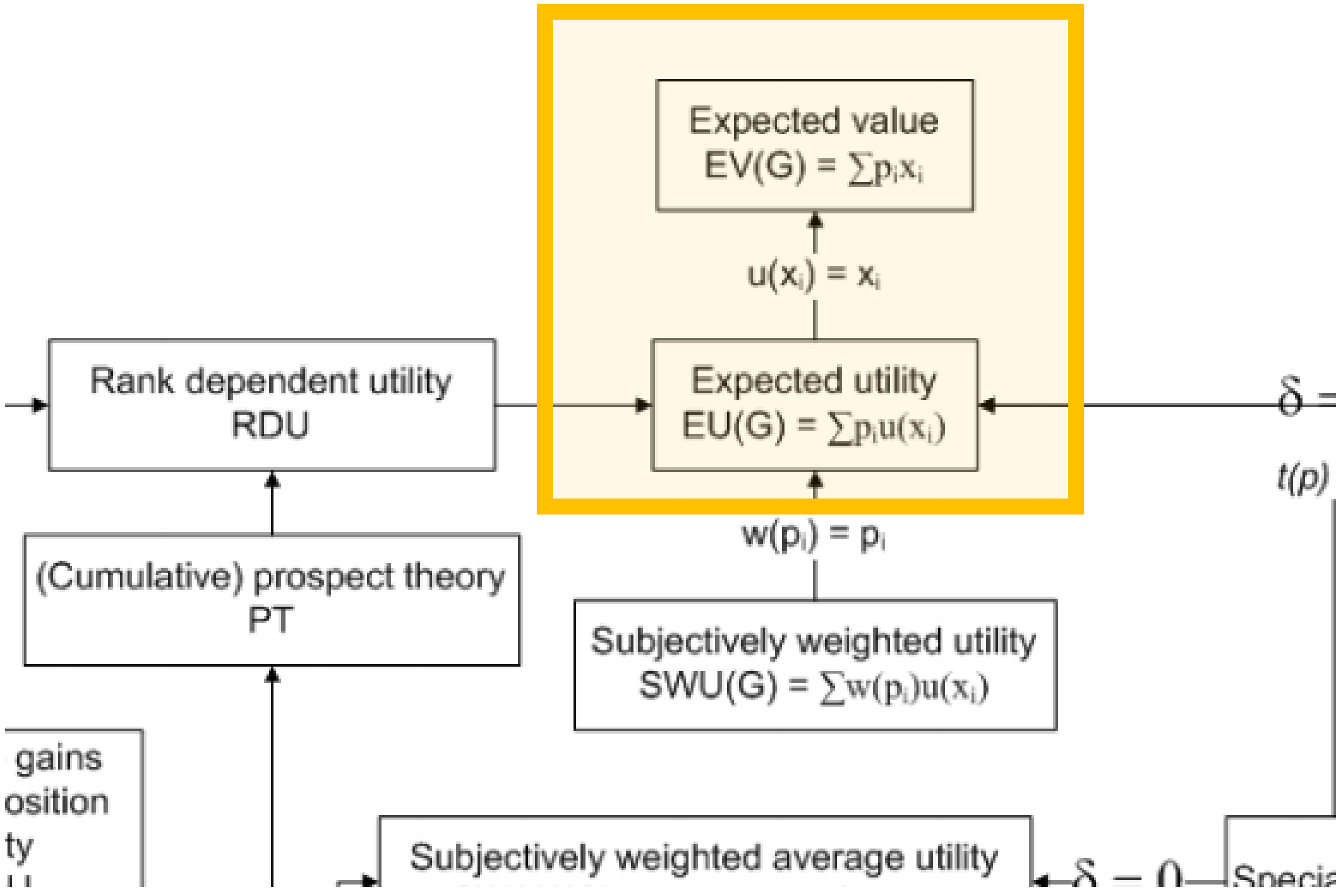




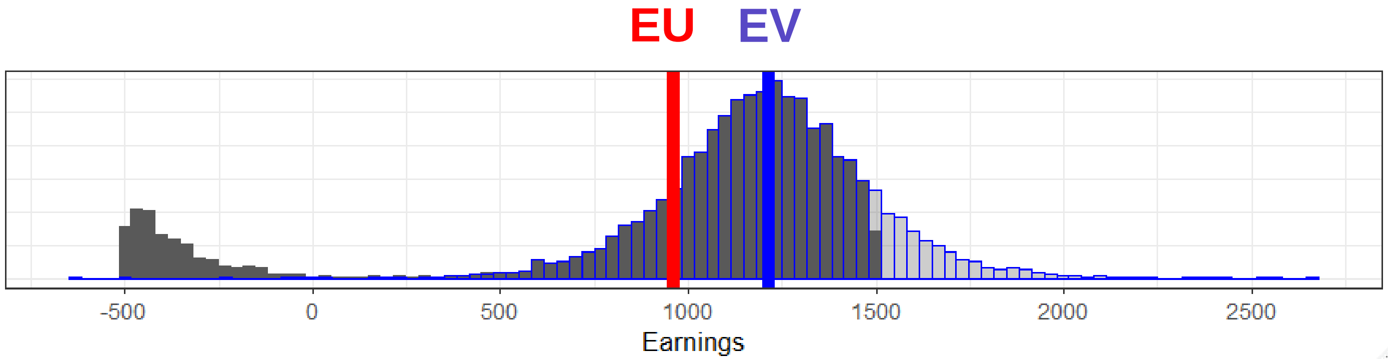
UTILITY THEORY

- Seven axioms:
  - Three by Kolmogorov (or five)
  - Four by Von Neumann–Morgenstern
- Utility theory is about outcomes (x_i), not inferred predictors.
- However, we can use predictors to predict hypothetical outcomes.
- Insight: science is value-neutral. All outcomes are valued equally (i.e., squared loss).

  $u(x_i) = x_i$



(From the vignette to the 'pt' package.)



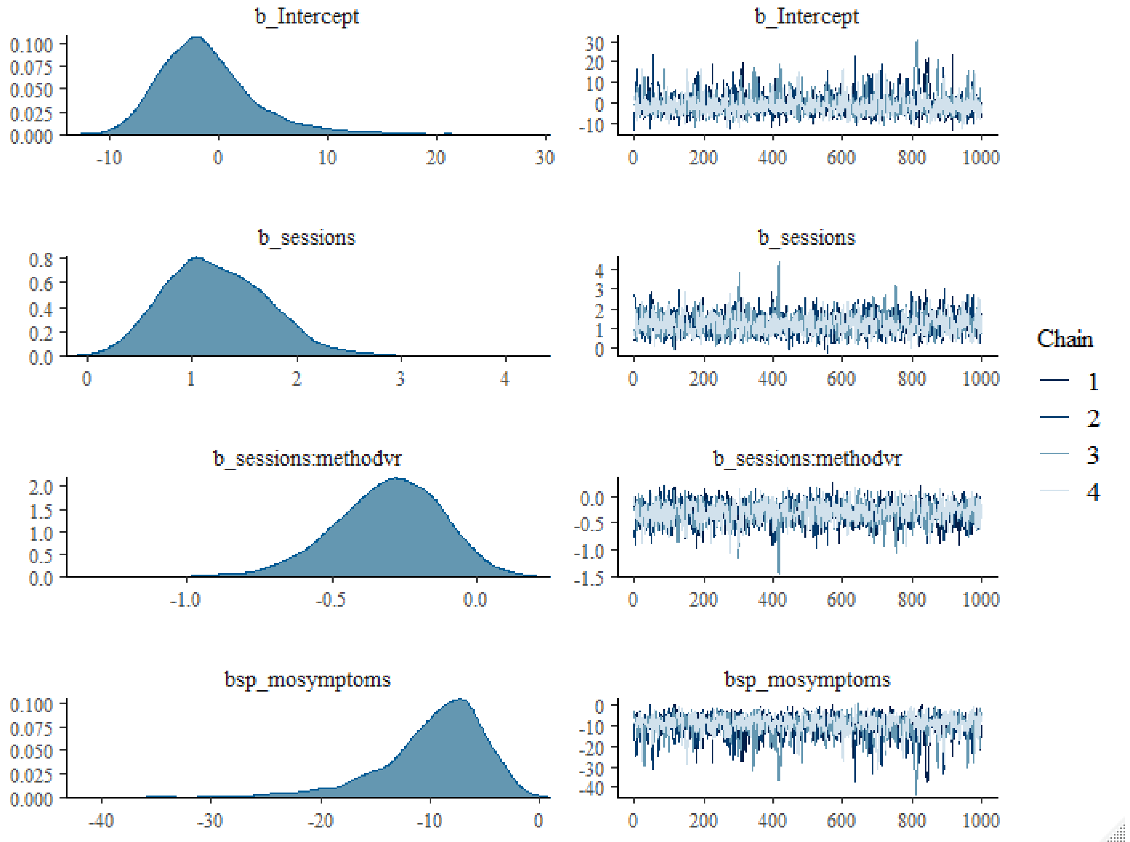
EXAMPLE: PREDICTING OPTIMAL THERAPY LENGTH

id	success	symptoms	method	sessions
1	1	2	vr	14
2	0	4	face	6
3	1	5	face	9
4	1	3	face	10
5	1	4	vr	11
…	…	…	…	…

1. INFER PREDICTORS

- In general: if you can get a posterior distribution, you can calculate utility.
    - GLMM (brms)
    - SEM (blavaan)
- Use priors on predictors to reflect other sources of ”data”.
    - Previous literature.
    - Subjective hunches.

```
# Infer predictors
library('brms')

formula=success ~ sessions + sessions:method + mo(symptoms)

priors = c(
  set_prior('normal(1, 1)', coef='sessions'),
  set_prior('normal(1, 1)', coef='sessions:methodvr')
)
fit = brm(formula, data=df, prior=priors, family=bernoulli)
```



2. DEFINE UTILITIES

- Make a loss function to reflect the positive or negative utility given predictors.

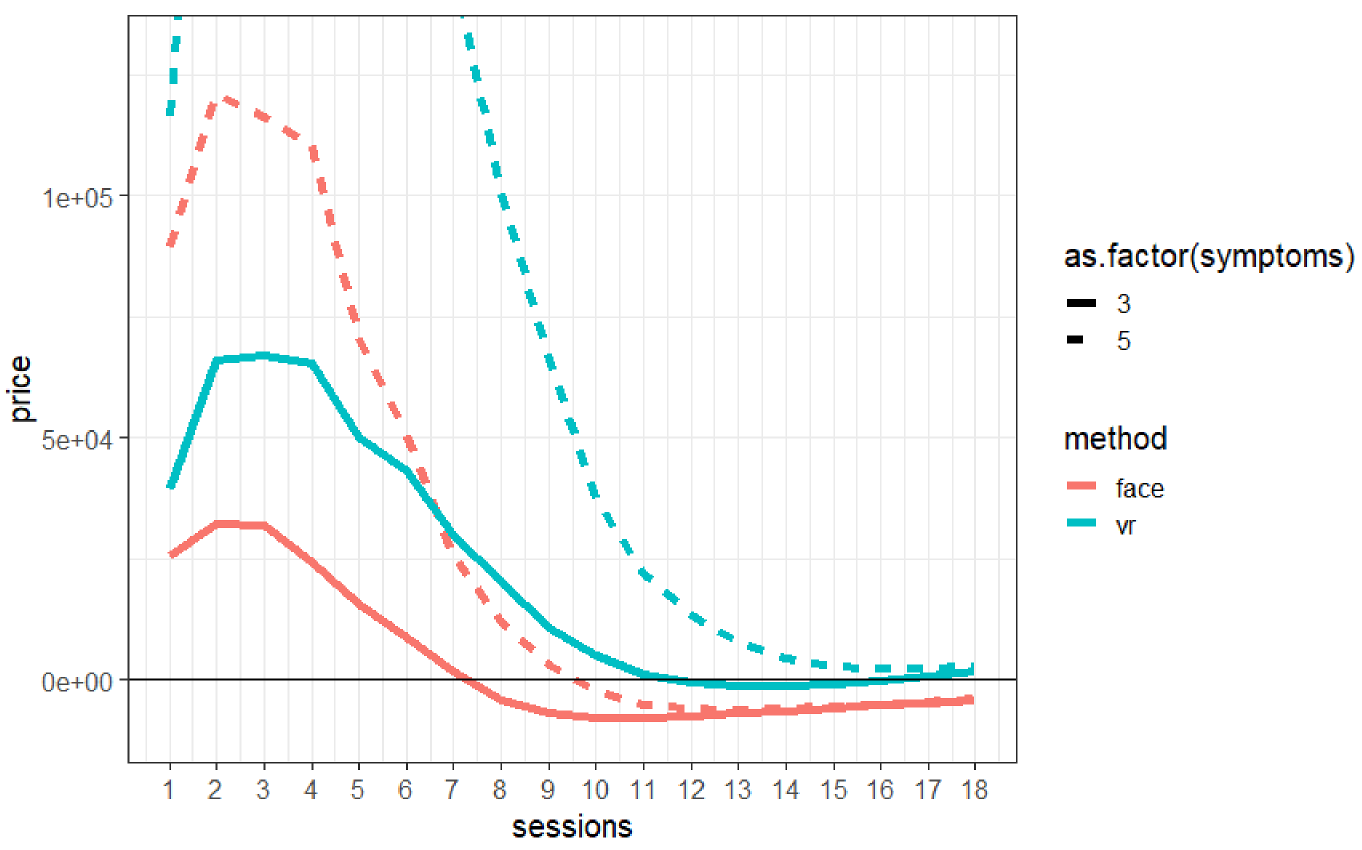
```
# Utility in terms of money
price_session = list(vr=600, face=900)  # Negative
outcome = c(success=15000, loss=-4000)  # Positive

cost = function(sessions, method, symptoms) {
  # Make the prediction
  new = data.frame(sessions, method, symptoms)
  prob_success = predict(fit, newdata=new)[1]

  # Add loss (cost of treatment)
  therapy_price = price_session[[method]] * sessions
  therapy_gain = prob_success * outcome['success'] +
                 (1-prob_success)* outcome['loss']

  # Gain should outweigh loss per treatment success:
  therapy_price * (1 / prob_success) - therapy_gain
}
```

sessions	method	symptoms	price
1	vr	3	39.525
2	vr	3	66.062
3	vr	3	67.102
4	vr	3	65.240
5	vr	3	49.954
1	vr	3	39.525

3. MAKE A DECISION

- Maximum gain or minimum loss:
  - Eye-ball maxima and minima in plots.
  - optim or similar functions (but beware local extremums)

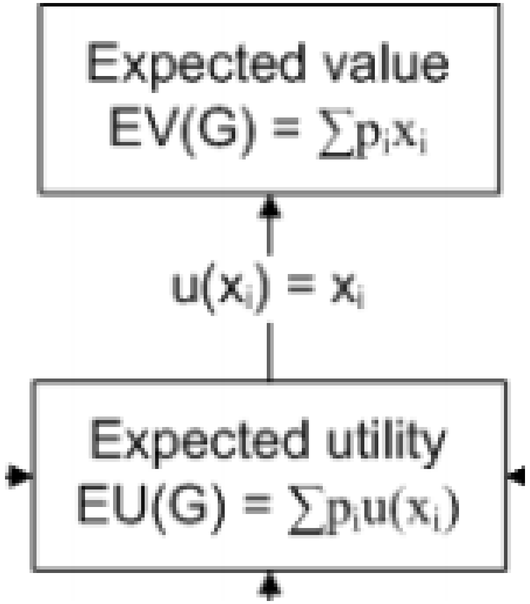
Method 1: Eye-ball plots

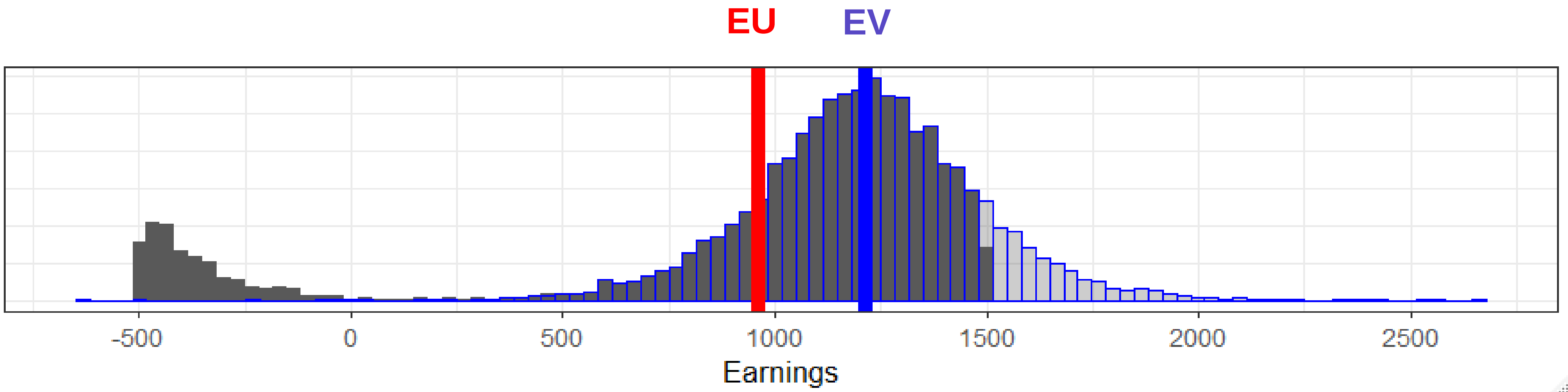


Method 2: Optimization functions

```
> optim(9, function(x) -cost(x, method='face', symptoms=5))
[1] 12.75

$value
[1] -2381.112
```

SUMMARY


Expected value
EV(G) = ∑p_i x_i
u(x_i) = x_i
Expected utility
EU(G) = ∑p_i u(x_i)


EU
EV
-500
0
500
1000
1500
2000
2500
Earnings