

## R

## effects

---

### 基本的な使い方

効果のあるものすべて `allEffects`

```
plot(allEffects(model))
```

```
plot(allEffects(model), multiline=T)
```

- ・ 複数の線を重ねて描く

```
plot(allEffects(model), multiline=T, confint = list(style = "auto"))
```

- ・ 信頼区間も加える ( , confint = list(style = "auto") )

```
plot(allEffects(model),  
      multiline=T,  
      confint = list(style = "auto"),  
      colors= c("black", "gray"))
```

- ・ 白黒にする colors= c("black", "gray")

### 発展

予測変数の効果の可視化

- ・ 出来上がったモデルにおいて、特定の説明変数がどのように結果を予測するかを可視化

```
plot(predictorEffects(model))
```

```
plot(predictorEffects(model), multiline=T)
```

- ・ 複数の線を重ねて描く

条件のレベルの設定 `xlevels=list( 変数 =c())`

```
plot(predictorEffects( モデル , xlevels=list( 変数 =c(3, 5, 10, 15, 20))))
```

軸の工夫 `axes=list()`

(Predictor Effects Graphics Gallery; pp. 15-17.)

- ・ 説明変数が対数変換されている場合そのままだと、線がカーブする
  - ・ x 軸自体を対数変換したメモリにすれば、直線になる

```
x=list( 変数名 =list(transform=list(trans=log, inverse=exp)))
```

- ・ x 軸のメモリを指定する

```
ticks=list(at=c(2000, 5000, 10000, 20000))
```

- ・ x 軸の範囲を指定する

```
lim=c(1900, 21000)
```

- ・ x 軸のラベルを指定する

```
lab="変数名, log-scale"
```

### 特定の変数の効果

```
plot(Effect(c("変数", "変数"), model))
```

```
plot(Effect(c("変数", "変数"), model), multiline=T)
```

- ・ 複数の線を重ねて描く

要因がたくさんある場合に、X 軸に来る要因を指定する

### 参考例

<https://yuzar-blog.netlify.app/posts/2021-01-01-how-to-visualize-models-their-assumptions-and-post-hocs/>