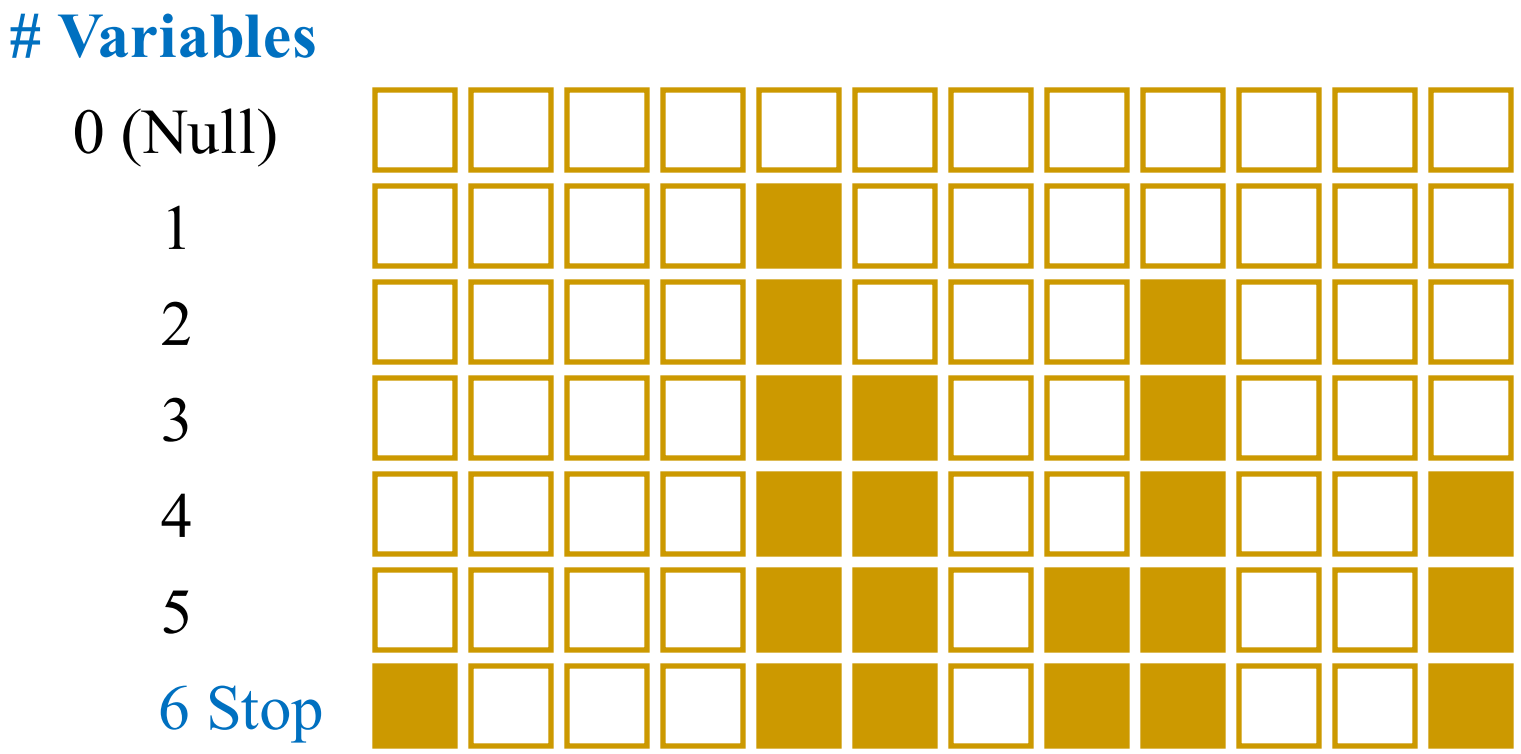
Step Methods

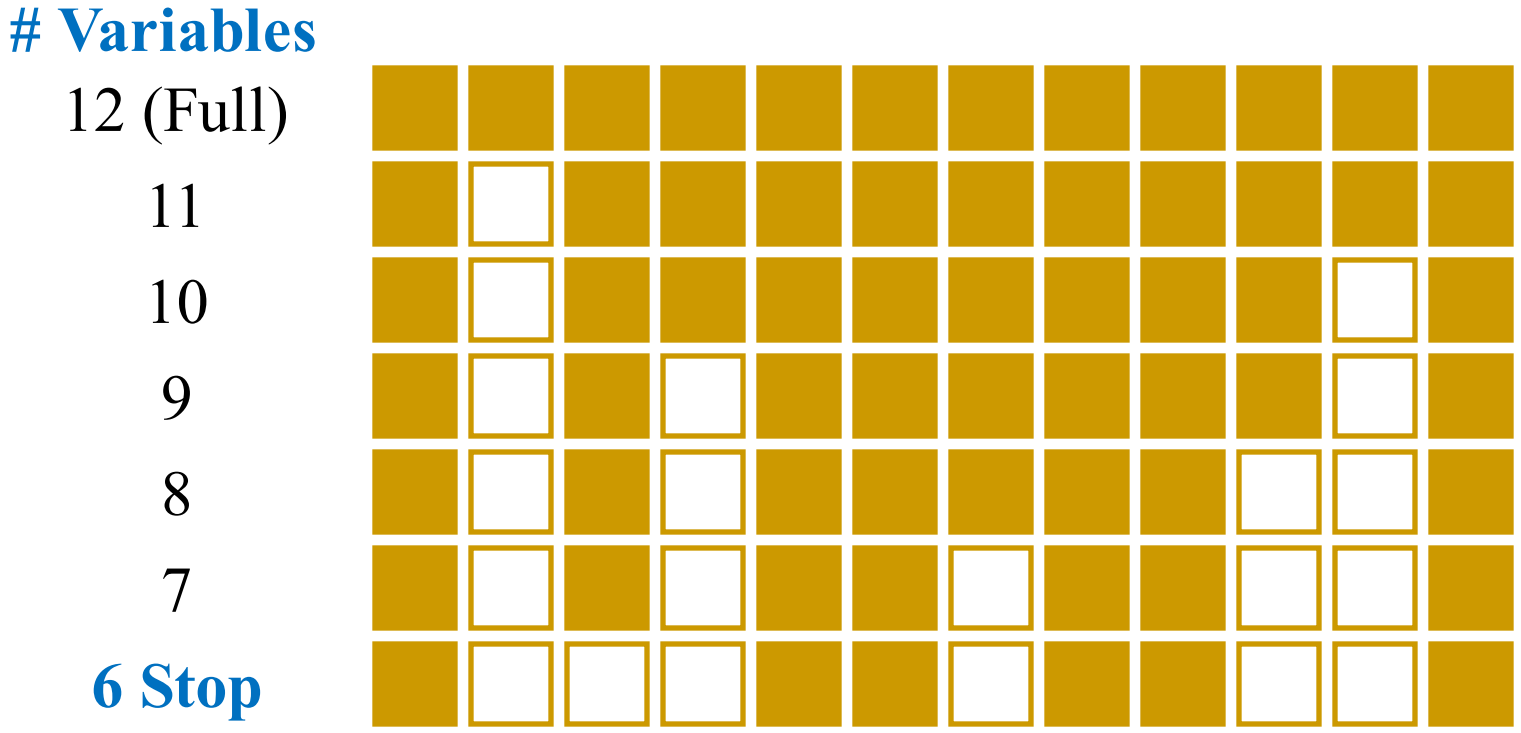
Forward Selection

- **Start** with the **Null** model; then try all **single-variable** models; select the **most significant variable**.
- Try each of the remaining variables, 1 at time; select the **most significant** variable, and so on.
- **Stop** when **no** more variables are **significant** when added to the model
- **Problem:** this method **starts** with the **most biased** model → as you add variables, some of the variables in the model may lose significance



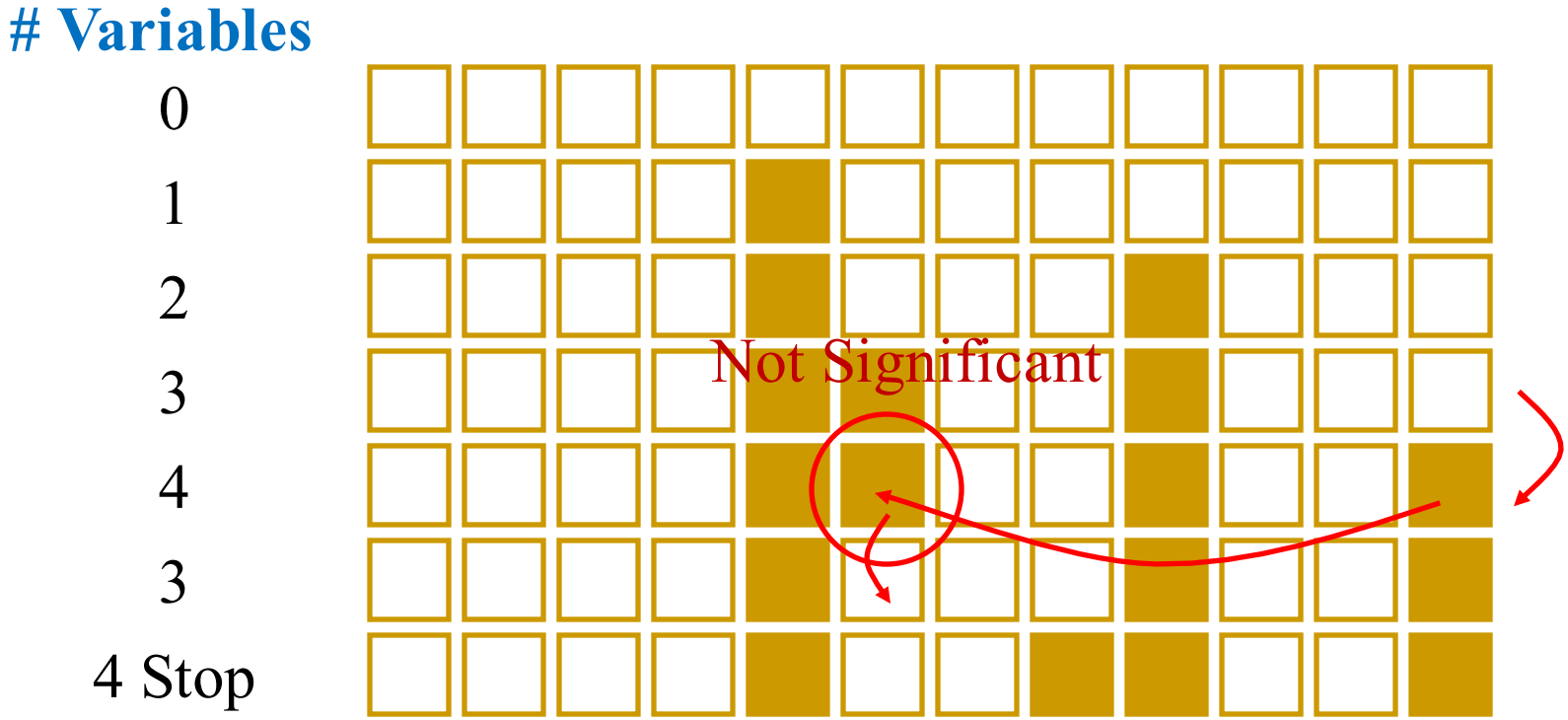
Backwards Elimination

- **Start** with the **Full** model; then remove the **least significant variable**; then the **next least significant**, and so on.
- **Stop** when **no** more of the variables left out become **significant** when there are **no more non-significant** variables to remove.
- **Problem:** this method **starts** with the **over-identified** model and may suffer from **dimensionality** issues → as you remove variables, some variables that were not significant may become **significant if added** again



Stepwise

- Works like **Forward** selection, but you can take a backward step to remove variables that become non-significant when other variables are added.
- **Stop** when **no** more variables become significant if added to the model and there are no non-significant variables to remove
- **Problem:** similar to Forward and Backwards methods, but it is **often preferred** due to the flexibility. One can **vary** the **p-value** for inclusion or removal of variables, thus giving more control of the selection process



R Tips

`regsubsets(){leaps}` → Can be used for forward/backward selection

```
regfit.fwd=regsubsets(y~x1+x2+etc.,
            data=dataName,nvmax=xx,
            method="forward")
```
or `method="backward")`

`step(){stat}` → Another function to do step selection with either `lm()` or `glm()` objects

`lm.null=lm(y~1, data=dataName)` → Define the null model with only the intercept

`lm.full=lm(Salary ~., data=dataName)` → Then define the full model with all variables

```
lm.fwd=step(lm.null,
        scope=list(lower=lm.null, upper=lm.full),
        direction="forward", test="F")
```
→ Forward selection from null to full model with F-tests → Use:

`direction="backward"` → For backward elimination; or

`direction="both"` → For stepwise

KOGOD SCHOOL
of
BUSINESS