Applied Statistical Regression
AS 2013 – Week 10

Marcel Dettling

Institute for Data Analysis and Process Design

Zurich University of Applied Sciences

marcel.dettling@zhaw.ch

http://stat.ethz.ch/~dettling

ETH Zürich, November 18, 2013

Collinearity = Correlated Predictors

If ≥ 2 predictors are strongly correlated, i.e. explain very similar aspects of the response, OLS estimation is difficult. The regression coefficients will be less precise, and the interpretation of the results is more difficult.

There is a need to recognize collinearity!

1) *Plot the correlation matrix of the predictors*
   ```
   plotcorr(cor(my.dat))
   ```

2) *Variance Inflation Factors*

$$Var(\hat{\beta}_k) = \sigma_E^2 \cdot \frac{1}{1-R_k^2} \cdot \frac{1}{\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)}$$

Applied Statistical Regression
AS 2013 – Week 10

How to Deal with Correlated Predictors?

1) **Generate new variables**

    → **see example on next slides**...

2) **Variable selection**

    Only use the relevant predictors, and omit the redundant ones. This often helps a lot. We will be discussing variable selection in detail.

3) **The Lasso and Ridge Regression**

    These are penalized OLS regression methods, which sparsely spend degrees of freedom. To be discussed later.

Example

Understanding how car drivers adjust their seat would greatly help engineers to design better cars. Thus, the measured

hipcenter = horizontal distance of hips to steering wheel

and tried to explain it with several predictors, namely:

Age	age in years
Weight	weight in pounds
HtShoes, Ht, Seated	height w/o, w/ shoes, seated height
Arm, Thigh, Leg	arm, thigh and leg length

We first fit a model with all these (correlated!) predictors

Example: Fit with All Predictors

```
> library(faraway); data(seatpos)
> summary(lm(hipcenter~., data=seatpos))
```

|	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	436.43213	166.57162	2.620	0.0138	*
Age	0.77572	0.57033	1.360	0.1843	
Weight	0.02631	0.33097	0.080	0.9372	
HtShoes	-2.69241	9.75304	-0.276	0.7845	
Ht	0.60134	10.12987	0.059	0.9531	
Seated	0.53375	3.76189	0.142	0.8882	
Arm	-1.32807	3.90020	-0.341	0.7359	
Thigh	-1.14312	2.66002	-0.430	0.6706	
Leg	-6.43905	4.71386	-1.366	0.1824	

```
Residual standard error: 37.72 on 29 degrees of freedom
Multiple R-squared: 0.6866, Adjusted R-squared: 0.6001
F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

Collinearity in the Seat Position Example

```
> vif(fit)
        Age        Weight       HtShoes             Ht
   1.997931      3.647030    307.429378     333.137832
     Seated           Arm         Thigh            Leg
   8.951054      4.496368      2.762886       6.694291
```
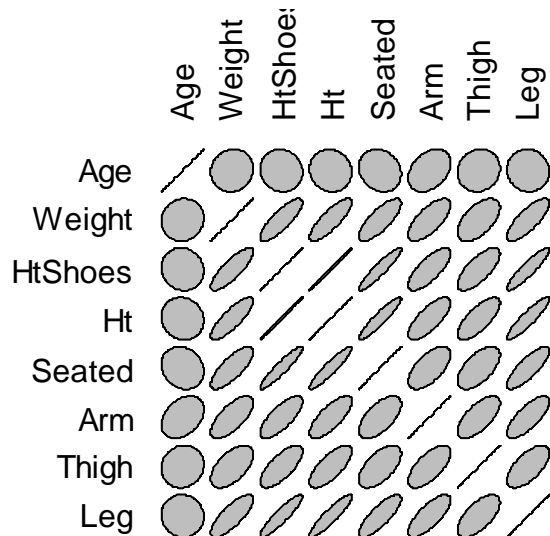

VIF ≥ 5 is critical, VIF ≥ 10 is dangerous. The observed values mean that the standard errors of the estimates are inflated by a factor up to about 18x.

Example: Generating New Variables

The body height is certainly a key predictors when it comes to the position of the driver seat. We leave this as it was, and change several of the other predictors:

```
age     <- Age
bmi     <- (Weight*0.454)/(Ht/100)^2
shoes   <- HtShoes-Ht
seated  <- Seated/Ht
arm     <- Arm/Ht
thigh   <- Thigh/Ht
leg     <- Leg/Ht
```
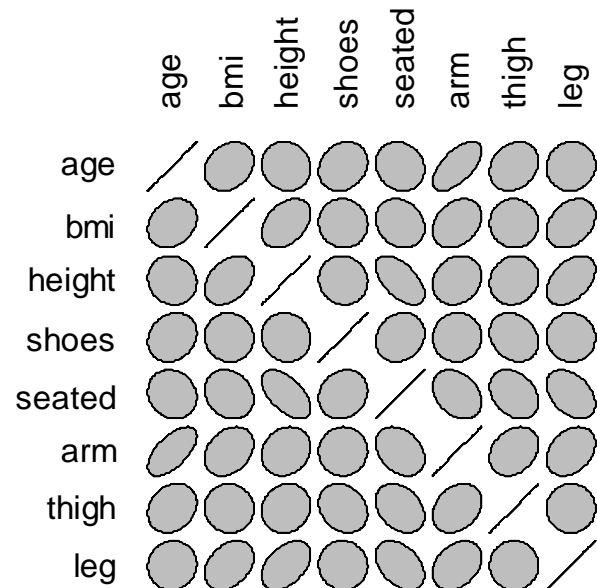
Does this solve the correlation problem...?

Example: New Correlation Matrix

```
> vif(fit00)
      age        bmi     height      shoes     seated
 1.994473   1.408055   1.968447   1.155285   1.851884
      arm      thigh        leg
 2.044727   1.284893   1.480397
```

Example: Fit with New Predictors

```
> summary(lm(hipc~., data=new.seatpos))

              Estimate Std. Error t value Pr(>|t|)
(Intercept) -632.0063    490.0451  -1.290    0.207
age           -0.7402      0.5697  -1.299    0.204
bmi           -0.4234      2.2622  -0.187    0.853
height         3.6521      0.7785   4.691 5.98e-05 ***
shoes          2.6964      9.8030   0.275    0.785
seated      -171.9495    631.3719  -0.272    0.787
arm          180.7123    655.9536   0.275    0.785
thigh        141.2007    443.8337   0.318    0.753
leg         1090.0111    806.1577   1.352    0.187
```

Residual standard error: 37.71 on 29 degrees of freedom
Multiple R-squared: 0.6867, Adjusted R-squared: 0.6002
F-statistic: 7.944 on 8 and 29 DF, p-value: 1.3e-05

Variable Selection: Why?

We want to fit a regression model…

Case 1: functional form and predictors exactly known
→ *estimation, test, confidence and prediction intervals*

Case 2: neither functional form nor the predictors are known
→ *explorative model search among potential predictors*

Case 3: we are interested in only 1 predictor, but want to correct for the effect of other covariates
→ *which covariates we need to correct for?*

Question in cases 2 & 3: WHICH PREDICTORS TO USE?

Variable Selection: Technical Aspects

We want to keep a model small, because of

1) Simplicity

→ *among several explanations, the simplest is the best*

2) Noise Reduction

→ *unnecessary predictors leads to less accuracy*

3) Collinearity

→ *removing excess predictors facilitates interpretation*

4) Prediction

→ *less variables, less effort for data collection*

Method or Process?

- **Variable selection is not a method!** The search for the best predictor set is an iterative process. It also involves *estimation*, *inference* and *model diagnostics*.

- For example, outliers and influential data points will not only change a particular model – they can even have an impact on the model we select. Also variable transformations will have an impact on the model that is selected.

- Some iteration and experimentation is often necessary for variable selection. *The ultimate aim is finding a model that is smaller, but as good or even better than the original one.*

Example: Mortality Data

```
> summary(fit.orig)
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept) 1496.4915    572.7205   2.613  0.01224 *
JanTemp        -2.4479      0.8808  -2.779  0.00798 **
...
Dens           11.9490     16.1836   0.738  0.46423
NonWhite      326.6757     62.9092   5.193 5.09e-06 ***
WhiteCollar  -146.3477    112.5510  -1.300  0.20028
...
---
Residual standard error: 34.23 on 44 degrees of freedom
Multiple R-squared: 0.7719,Adjusted R-squared: 0.6994
F-statistic: 10.64 on 14 and 44 DF,  p-value: 6.508e-10
```

Note: due to space constraints, this is only part of the output.

Backward Elimination with p-Values

Aim: Reducing the regression model such that the remaining predictors show a significant relation to the response.

How: We start with the full model and then exclude the least significant predictor in a step-by-step manner, as long as its p-value is greater than $\alpha_{crit} = 0.05$.

In R:
```
> fit <- update(fit, . ~ . - RelHum)
> summary(fit)
```

→   *Re-fit the model after each exclusion!*

→   *Wording:* **Backward Elimination with p-Values**

→   For prediction, one also uses $\alpha_{crit} = 0.10 / 0.15 / 0.20$

Example: Final Result

```
> ft09 <- update(ft08, .~.-WhiteCollar); summary(ft09)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 992.2069     79.6994  12.449  < 2e-16 ***
JanTemp      -2.1304      0.5017  -4.246 8.80e-05 ***
Rain          1.8122      0.5066   3.577 0.000752 ***
Educ        -16.4207      6.1202  -2.683 0.009710 **
NonWhite    268.2564     38.8832   6.899 6.56e-09 ***
NOx          18.3230      4.3960   4.168 0.000114 ***
---
Residual standard error: 33.47 on 53 degrees of freedom
Multiple R-squared: 0.7373, Adjusted R-squared: 0.7125
F-statistic: 29.75 on 5 and 53 DF,  p-value: 2.931e-14
```

→ 9 predictors are eliminated, 5 remain in the final model.

Interpretation of the Result

- The remaining predictors are now "more significant" than before. This is almost always the case. Do not overestimate the importance of these predictors!

- Collinearity among the predictors is usually at the root of this observation. The predictive power is first spread out among several predictors, then it becomes concentrated.

- **Important**: the removed variables can still be related to the response. If we run a simple linear regression, they can even be significant. In the multiple linear model however, there are other, better, more informative predictors.

Alternatives to Backward Elimination

Backward elimination that is based on p-values requires laborious handwork (*in R*) and has a few disadvantages...

- When the principal goal is prediction, then the resulting models are often too small, i.e. there are bigger models which yield a more accurate prognosis.

- From a (theoretical) mathematical viewpoint variable selection via the AIC/BIC criterions is more suitable.

- In a step-by-step backward elimination, the best model is often missed. Evaluating more models can be very beneficial for finding *the best one*...

The AIC/BIC Criteria

Aim: Judging the quality of a regression model

→ *Gauging Goodness-of-Fit vs. The Number of Predictors*

AIC-Criterion:

$$AIC \quad = \quad -2\max(\log likelihood) + 2p$$
$$= \quad const + n\log(RSS / n) + 2p$$

BIC-Criterion:

$$BIC \quad = \quad -2\max(\log likelihood) + p\log n$$
$$= \quad const + n\log(RSS / n) + p\log n$$

Backward Elimination with AIC/BIC

Aim: Reducing the regression model such that the remaining predictors are *necessary* for describing the response.

How: We start with the full model and then in a step-by-step manner exclude the predictor that leads to the biggest improvement in AIC/BIC.

In R:
```
> fit.aic <- step(fit, dir="backward", k=2)

> fit.bic <- step(fit, dir="backward", k=log(59))
```

→ *The variable selection stops when AIC/BIC cannot be improved anymore. There is neither a need nor a guarantee that the selected predictors are significant.*

Example: Models with AIC/BIC

```
AIC:              Estimate Std. Error t value Pr(>|t|)
(Intercept) 1035.5384       85.1924  12.155  < 2e-16 ***
JanTemp        -2.0188        0.5043  -4.003 0.000200 ***
Rain            1.9637        0.5146   3.816 0.000363 ***
Educ          -11.7708        6.9613  -1.691 0.096842 .
NonWhite      261.5379       38.8830   6.726 1.35e-08 ***
WhiteCollar  -139.2913      102.0379  -1.365 0.178101
NOx            19.4440        4.4372   4.382 5.73e-05 ***


BIC:              Estimate Std. Error t value Pr(>|t|)
(Intercept)  992.2069       79.6994  12.449  < 2e-16 ***
JanTemp        -2.1304        0.5017  -4.246 8.80e-05 ***
Rain            1.8122        0.5066   3.577 0.000752 ***
Educ          -16.4207        6.1202  -2.683 0.009710 **
NonWhite      268.2564       38.8832   6.899 6.56e-09 ***
NOx            18.3230        4.3960   4.168 0.000114 ***
```
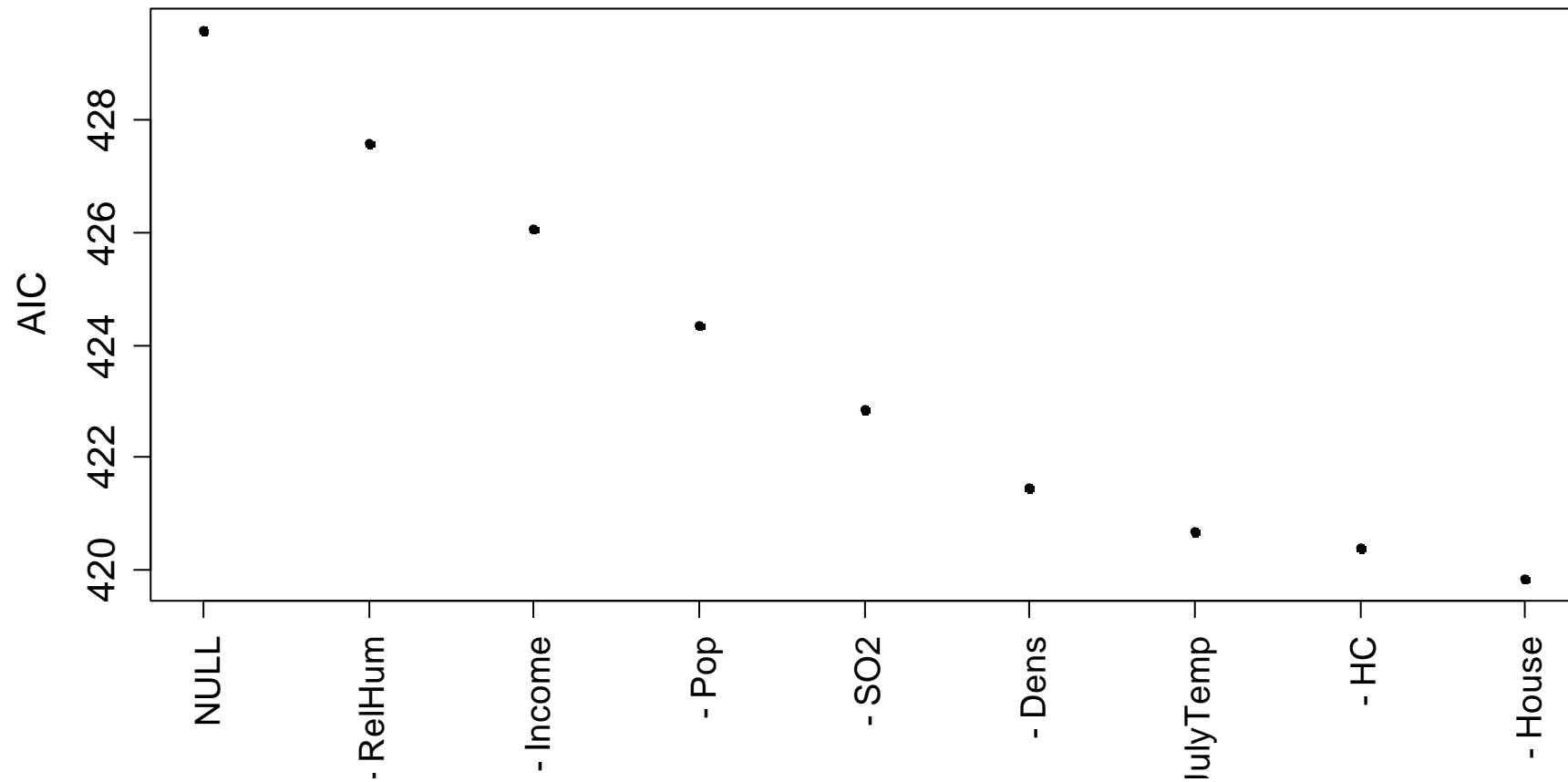
Visualization of Variable Selection

```
> plot(fit.aic$anova$AIC, ...)
```

Entwicklung des AIC-Kriteriums

AIC or BIC?

Usually, both criteria lead to similar models. BIC penalizes bigger models harder, with factor $\log n$ instead of factor 2.

→ *"BIC models" tend to be smaller than "AIC models"!*

Rule of the thumb for criterion choice:

- **BIC** is used when we are after a small model that is easy to interpret, i.e. in cases where understanding the predictor-response relation is the primary goal.

- **AIC** is used when the principal aim is the prediction of future observations. In these cases, small out-of-sample error is key, but neither the number or meaning of the predictors.

Alternative Search Heuristics

Forward Selection

1) Start with an empty model, i.e. only the intercept, but no predictors. The fitted value is the mean of the responses.

2) In a step-by-step manner, the predictor which leads to the best AIC/BIC value is added to the model.

3) Adding predictor variables is repeated until the AIC/BIC value can no longer be improved.

R: `> fit.aic <- step(fit, dir="forward", k=2)`

→ Forward Selection is used with big datasets, where backward elimination is too time consuming.

Alternative Search Heuristics

Stepwise Model Search

- This is an alternation of forward and backward steps. We can either start with the full model (1. step is backwards) or with the empty model (1. step is forward).

- In each forward step, all predictors can be added, also these that were excluded before. In each backward step, any of the predictors can be kicked out of the model (again).

→ Similar to Backward Elimination resp. Forward Search

→ Not much more time consuming, but more exhaustive

→ Recommended!

Stepwise Model Search in R

Starting with an empty model:

```
> null <- lm(Mortality ~ 1, data=mortality)
> all  <- lm(Mortality ~ ., data=mortality)
> fit  <- step(null, scope=list(upper=all))
```

Starting with the full model:

```
> fit  <- step(all, direction="both", k=2)
```

Note:

Argument `scope=...` allows specifying arbitrary minimal and maximal models for both cases. Then some predictors can be added or be removed from the model.

Alternative Search Heuristics

All Subsets Regression

- When m predictors are present, there are in fact 2^m different models that could be tried for finding the best one.

- In cases where m is small (i.e. $m \approx 10-20$) all submodels (up to a certain size) can be tried and evaluated by computing the AIC/BIC criterion.

→ Complete search, but enormous computing time needed

→ Yields a good solution, but not the causal model either

→ Recommended for small dataset where it is feasible

→ R implementation with function `leaps()`

All Subsets Regression in R

R commands:

```
> library(leaps)
> out <- regsubsets(Mortality~., nbest=1,
                      data=mortality, nvmax=14)
> summary(out)
> plot(out)
```
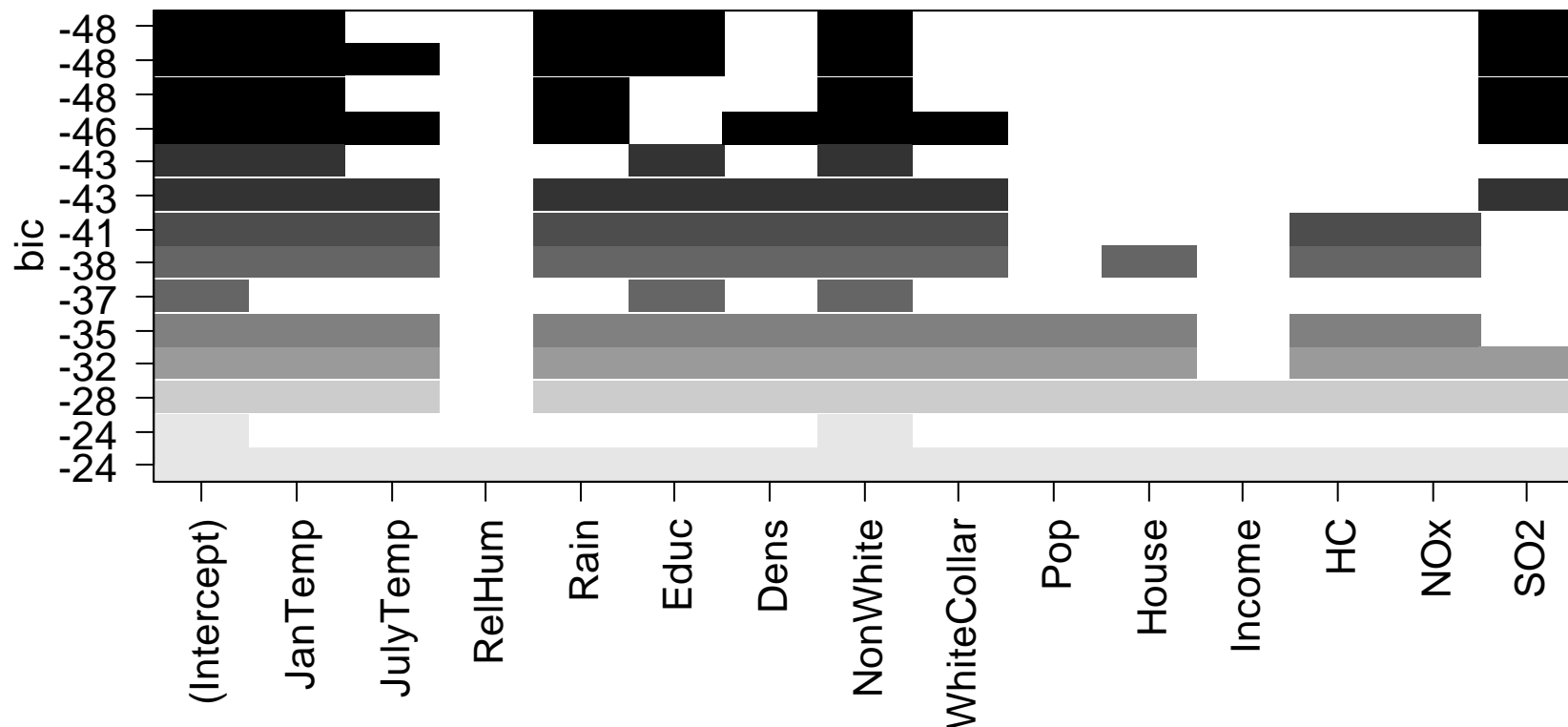
Note:

The procedure starts with the empty model and for each number of predictors, identifies the `nbest=1` models. By typing `~.` in the formula, all predictors are allowed. The maximum model size that is search can be determined with `nvmax=14`.

Visualization of All Subsets Selection

BIC-Modellevaluation nach All Subsets Regression

Final Remarks

- Each search heuristics yields a different *"best model"*.

- If we had another data sample from the same population and would repeat the variable selection using the same heuristic, the result might turn out to be different.

- The *"best model"* has the character of a random variable.

How to deal with that in practice?

We should not only consider the one "best model" according to a particular search heuristic, but a number of alternative model with similar performance (if they exist).

Interactions and Categorical Input

Models with Interactions

Do not remove main effect terms if there are interactions with these predictors contained in the model.

Categorical Input

- If a single dummy coefficient is non-significant, we cannot just kick this term out of the model, but we have to test the entire block of indicator variables.

- When we work manually and testing based, this will be done with a partial F-test. When working criterion based, `step()` does the right thing