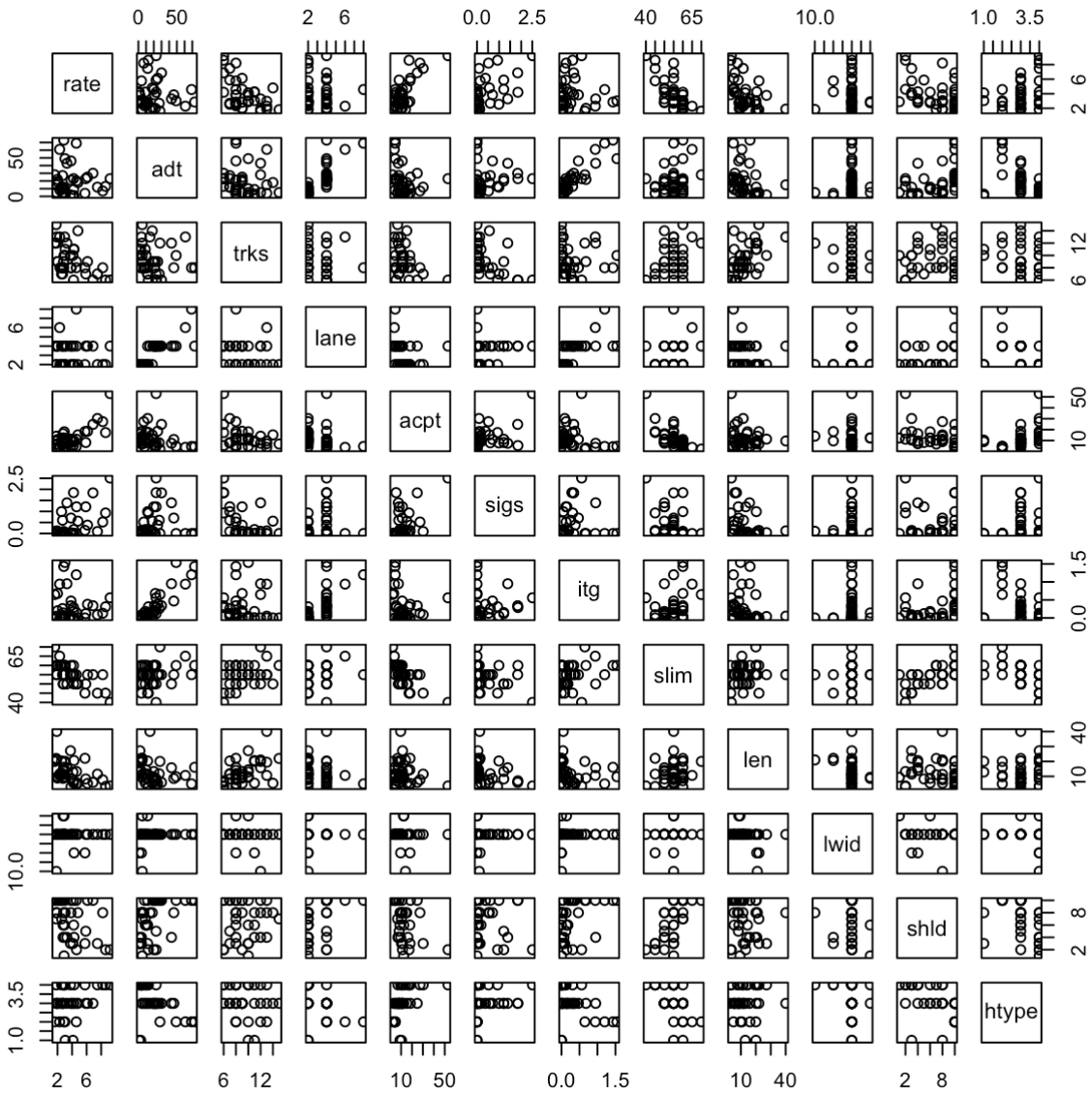
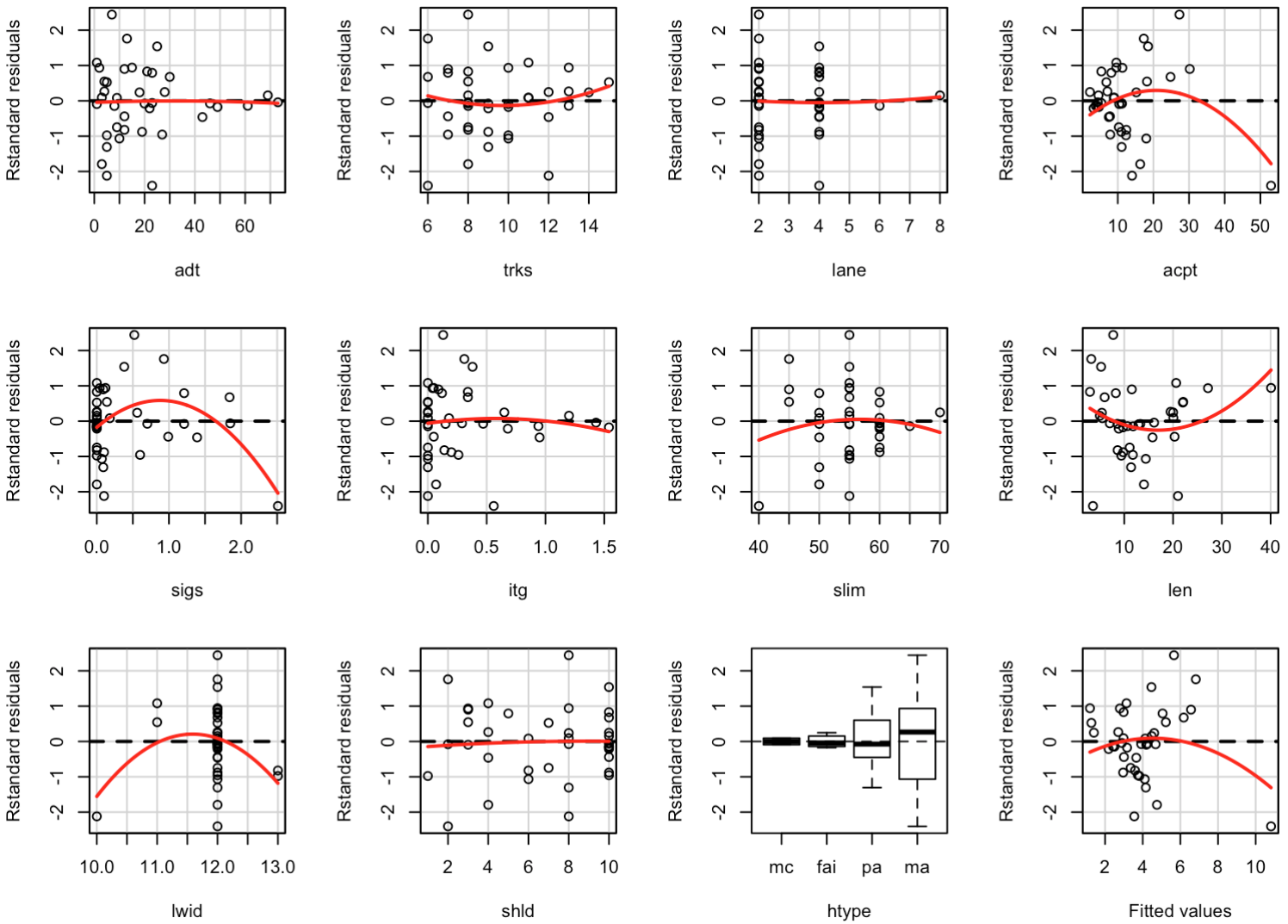
MLR: Model Selection in R

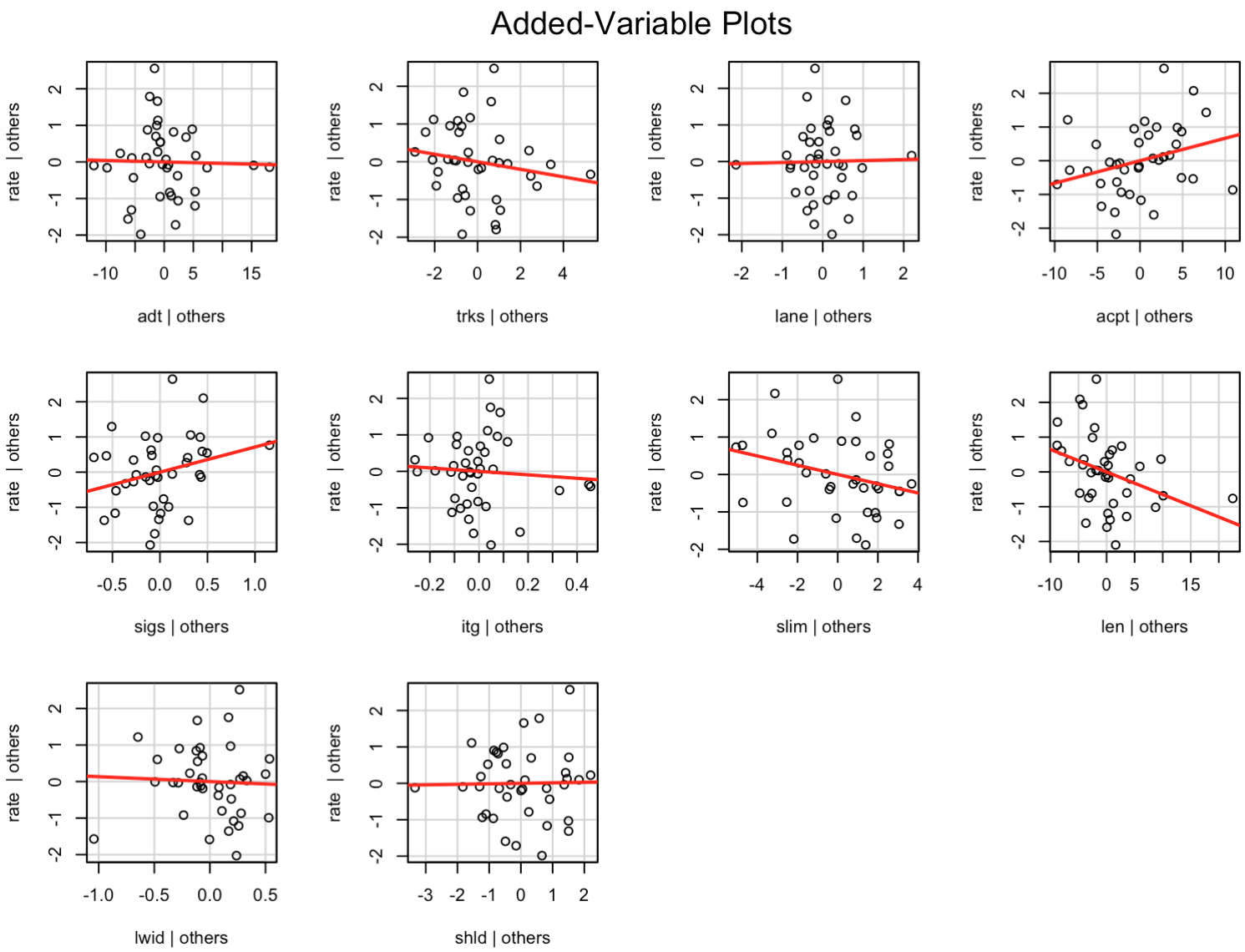
Math 430, Winter 2017

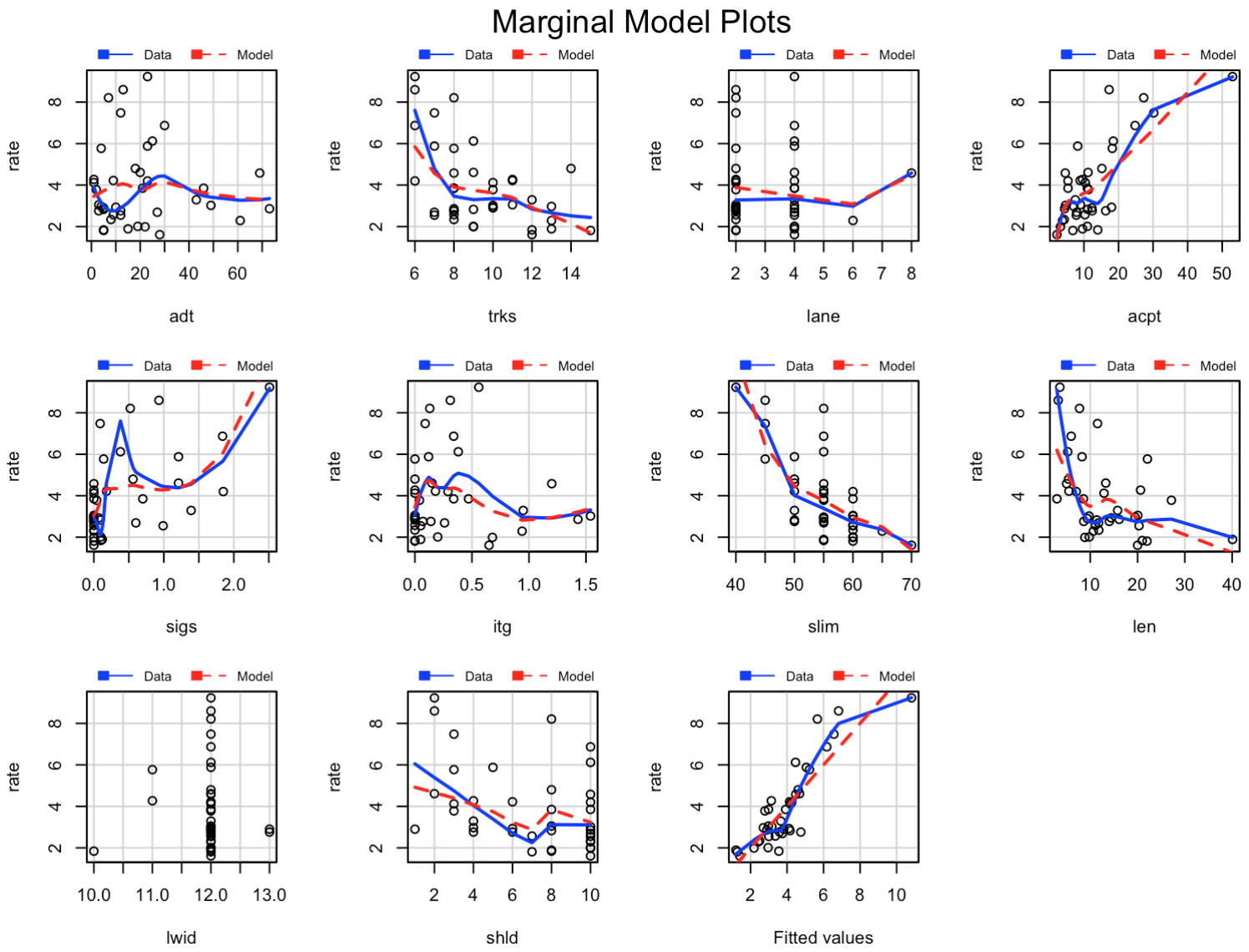
Highway accident data

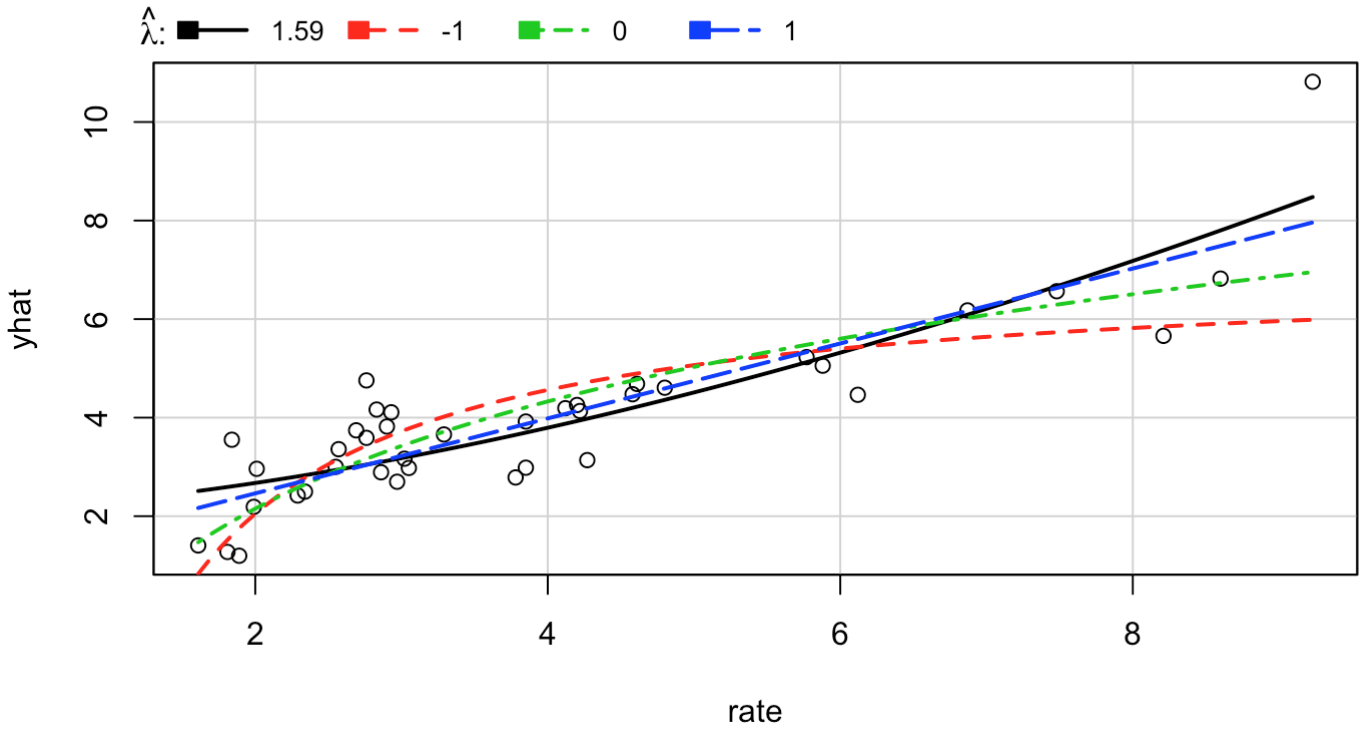
Variable	Description
adt	average daily traffic count (thousands)
trks	truck volume as a percent of the total volume
lane	total number of lanes of traffic
acpt	number of access points per mile
sigs	number of signalized interchanges per mile
itg	number of freeway-type interchanges per mile
slim	speed limit
len	length of the Highway segment (miles)
lwid	lane width (feet)
shld	width in feet of outer shoulder on the roadway
htype	type of roadway/funding source
rate	accident rate per million vehicle miles


rate
adt
trks
lane
acpt
sigs
itg
slim
len
lwid
shld
htype


Rstandard residuals
adt
trks
lane
acpt
sigs
itg
slim
len
lwid
shld
htype
mc
fai
pa
ma
Fitted values

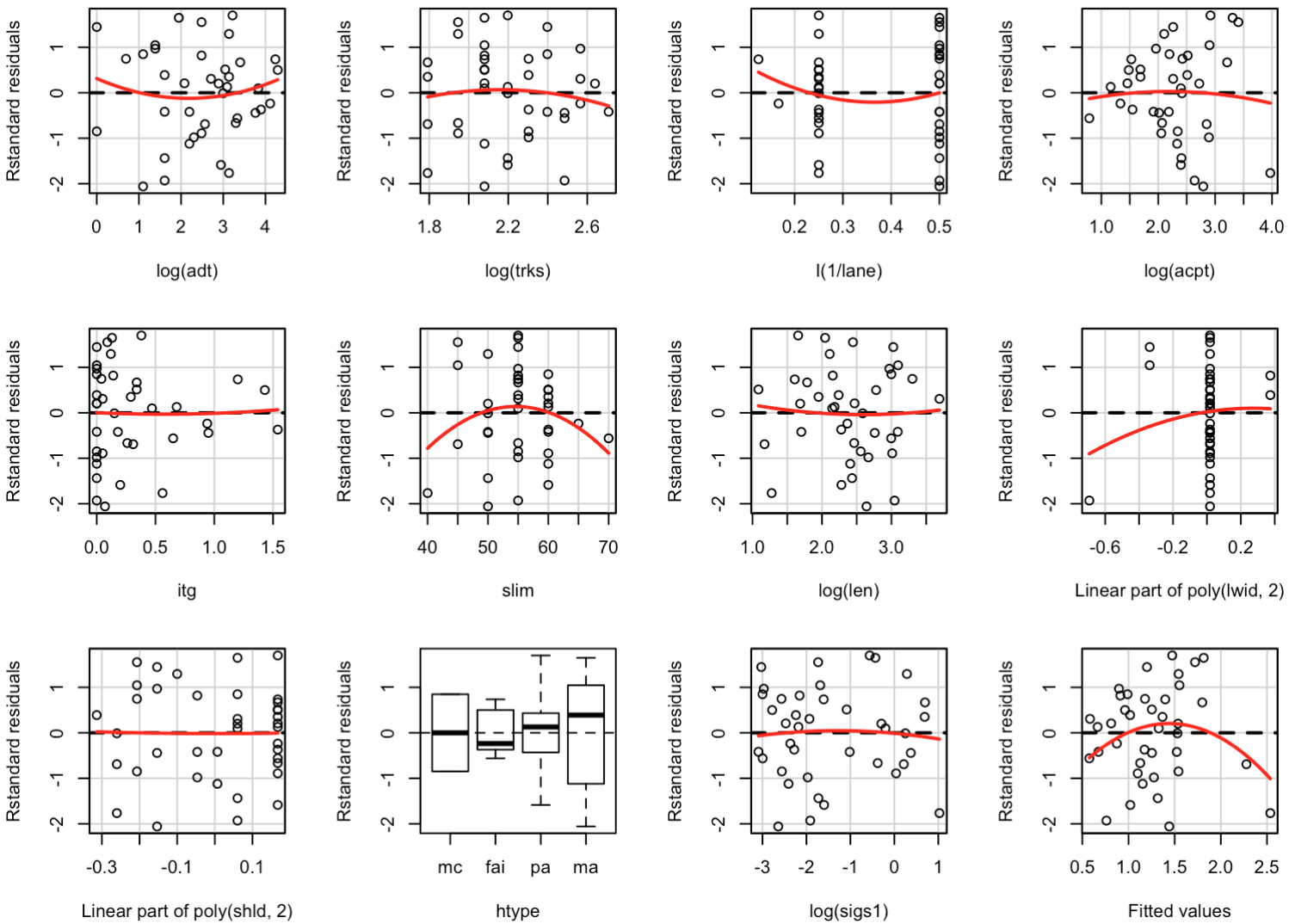

Added-Variable Plots
rate | others
adt | others
trks | others
lane | others
acpt | others
sigs | others
itg | others
slim | others
len | others
lwid | others
shld | others

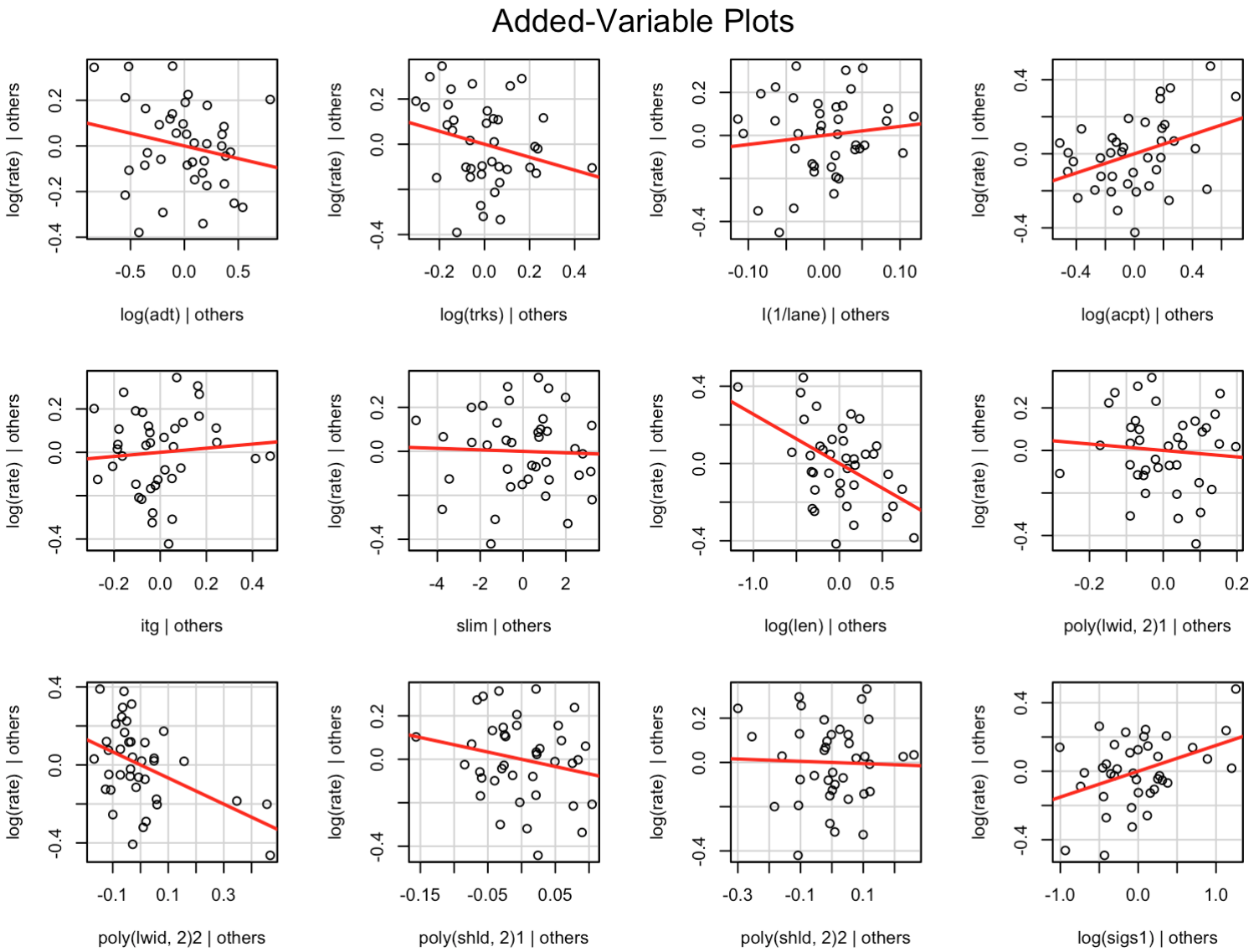

Marginal Model Plots
Data
Model
rate
adt
trks
lane
acpt
sigs
itg
slim
len
lwid
shld
Fitted values

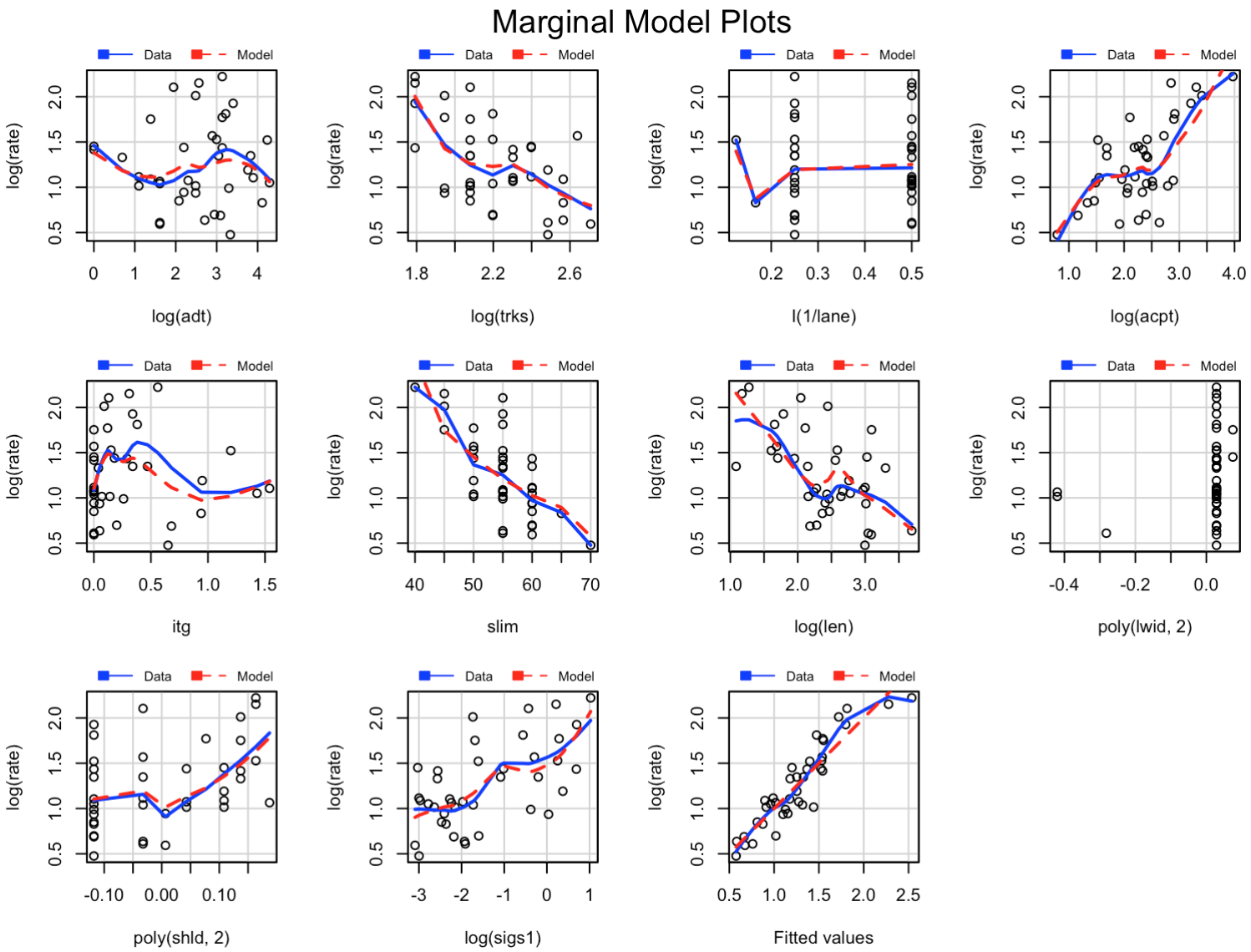



```
##      lambda      RSS
## 1  1.594529 26.41210
## 2 -1.000000 45.59093
## 3  0.000000 33.75636
## 4  1.000000 27.29810
```

```
Highway <- mutate(Highway, sigs1 = (sigs * len + 1)/len)
full_mod_tform <- lm(log(rate) ~ log(adt) + log(trks) + I(1/lane) + log(acpt) +
    itg + slim + log(len) + poly(lwid, 2) + poly(shld, 2) + htype + log(sigs1),
    data = Highway)
```


Rstandard residuals
log(adt)
log(trks)
I(1/lane)
log(acpt)
itg
slim
log(len)
Linear part of poly(lwid, 2)
Linear part of poly(shld, 2)
htype
mc
fai
pa
ma
log(sigs1)
Fitted values


Added-Variable Plots
log(rate) | others
log(adt) | others
log(trks) | others
I(1/lane) | others
log(acpt) | others
itg | others
slim | others
log(len) | others
poly(lwid, 2)1 | others
poly(lwid, 2)2 | others
poly(shld, 2)1 | others
poly(shld, 2)2 | others
log(sigs1) | others


Marginal Model Plots
Data
Model
log(rate)
log(adt)
log(trks)
l(1/lane)
log(acpt)
itg
slim
log(len)
poly(lwid, 2)
poly(shld, 2)
log(sigs1)
Fitted values

The `step` command

Backward elimination

```
belim <- step(full_mod_tform, scope = list(lower = ~ 1), direction = "backward")
```

```
broom::tidy(belim)
```

```
##              term    estimate  std.error  statistic      p.value
## 1     (Intercept)  3.24639448 0.75119775  4.3216244 0.0001764473
## 2        log(adt) -0.14429407 0.07746273 -1.8627547 0.0730195963
## 3       log(acpt)  0.18987179 0.10707212  1.7733075 0.0870567742
## 4            slim -0.02011261 0.01007683 -1.9959263 0.0557516497
## 5        log(len) -0.25644916 0.07871784 -3.2578279 0.0029403083
## 6  poly(lwid, 2)1  0.13688282 0.25106602  0.5452065 0.5899285279
## 7  poly(lwid, 2)2 -0.60177023 0.23510121 -2.5596220 0.0161662281
## 8        htypefai  0.33059140 0.33000676  1.0017716 0.3250331856
## 9         htypepa -0.21786065 0.21955592 -0.9922786 0.3295598277
## 10        htypema -0.06105924 0.18951707 -0.3221833 0.7497070874
## 11     log(sigs1)  0.17789568 0.05689946  3.1264916 0.0040983118
```

Forward selection

```
null_mod <- lm(log(rate) ~ 1, data = Highway)
fselect <- step(null_mod, scope = list(lower = ~ 1,
upper = ~ log(adt) + log(trks) + I(1/lane) + log(acpt) + itg + slim + log(len) + poly(lwid, 2) + poly(s
direction = "forward")
```

```
broom::tidy(fselect)
```

```
##              term     estimate  std.error   statistic     p.value
## 1     (Intercept)  2.122284499 0.95534397  2.22148730 0.034283129
## 2            slim -0.001240547 0.01467532 -0.08453287 0.933213647
## 3        log(len) -0.313267858 0.08812125 -3.55496377 0.001318954
## 4       log(acpt)  0.290282436 0.10291806  2.82051992 0.008560433
## 5  poly(lwid, 2)1 -0.340597367 0.27175343 -1.25333238 0.220095829
## 6  poly(lwid, 2)2 -0.778909771 0.25408606 -3.06553523 0.004666478
## 7  poly(shld, 2)1 -0.917057215 0.42976253 -2.13386963 0.041437529
## 8  poly(shld, 2)2 -0.013503844 0.30087592 -0.04488177 0.964509178
## 9       log(trks) -0.342058129 0.20960980 -1.63188044 0.113518148
## 10            itg  0.153598077 0.12288303  1.24995348 0.221309869
```

Stepwise selection

```
step_hwy <- step(null_mod, scope = list(lower = ~ 1,
upper =  ~ log(adt) + log(trks) + I(1/lane) + log(acpt) + itg + slim + log(len) + poly(lwid, 2) + poly(
direction = "both")
```

```
broom::tidy(step_hwy)
```

```
##             term    estimate  std.error   statistic      p.value
## 1    (Intercept)  2.05731539 0.55797123  3.68713528 0.0008951161
## 2       log(len) -0.31583627 0.08133725 -3.88304606 0.0005260574
## 3      log(acpt)  0.29490540 0.08573062  3.43990740 0.0017315283
## 4 poly(lwid, 2)1 -0.34766034 0.25427502 -1.36726107 0.1817012875
## 5 poly(lwid, 2)2 -0.78312449 0.24498872 -3.19657372 0.0032673201
## 6 poly(shld, 2)1 -0.94261102 0.30037603 -3.13810336 0.0037965769
## 7 poly(shld, 2)2 -0.02018253 0.28547297 -0.07069857 0.9441068560
## 8      log(trks) -0.34589714 0.20121625 -1.71903182 0.0959128224
## 9            itg  0.15622255 0.11691234  1.33623655 0.1915196786
```

Using BIC rather than AIC

```
belim_bic <- step(full_mod_tform, scope = list(lower = ~ 1), direction = "backward",
                  k = log(nrow(Highway)))
```

```
broom::tidy(belim_bic)
```

```
##              term    estimate  std.error  statistic      p.value
## 1     (Intercept)  3.24639448 0.75119775  4.3216244 0.0001764473
## 2        log(adt) -0.14429407 0.07746273 -1.8627547 0.0730195963
## 3       log(acpt)  0.18987179 0.10707212  1.7733075 0.0870567742
## 4            slim -0.02011261 0.01007683 -1.9959263 0.0557516497
## 5        log(len) -0.25644916 0.07871784 -3.2578279 0.0029403083
## 6  poly(lwid, 2)1  0.13688282 0.25106602  0.5452065 0.5899285279
## 7  poly(lwid, 2)2 -0.60177023 0.23510121 -2.5596220 0.0161662281
## 8        htypefai  0.33059140 0.33000676  1.0017716 0.3250331856
## 9         htypepa -0.21786065 0.21955592 -0.9922786 0.3295598277
## 10        htypema -0.06105924 0.18951707 -0.3221833 0.7497070874
## 11     log(sigs1)  0.17789568 0.05689946  3.1264916 0.0040983118
```

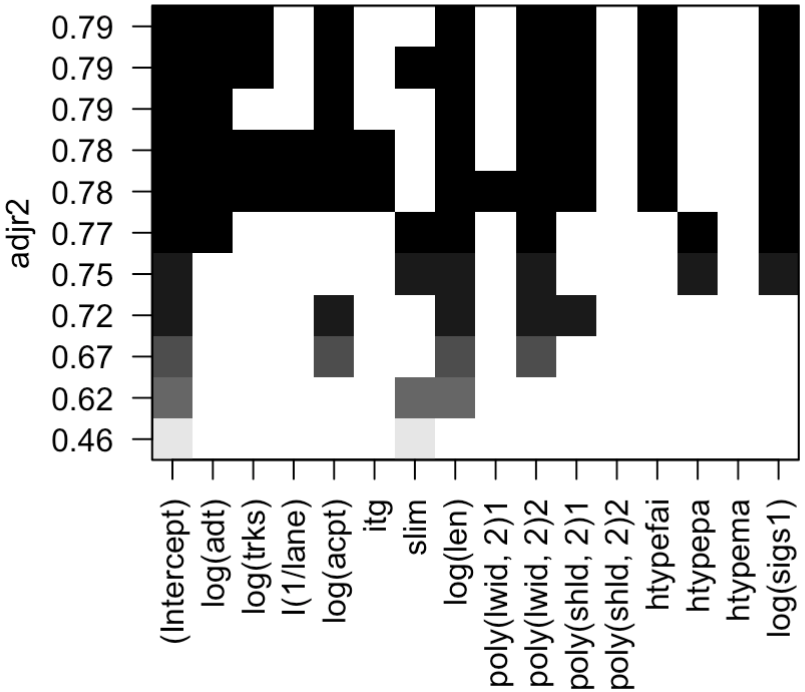
The `regsubsets` command

All subsets in R

```
library(leaps)
regfit_full <- regsubsets(log(rate) ~ log(adt) + log(trks) + I(1/lane) + log(acpt) +
    itg + slim + log(len) + poly(lwid, 2) + poly(shld, 2) + htype + log(sigs1),
    data = Highway, method = "exhaustive", nvmax = 11, nbest = 1)
reg_summary <- summary(regfit_full)
```

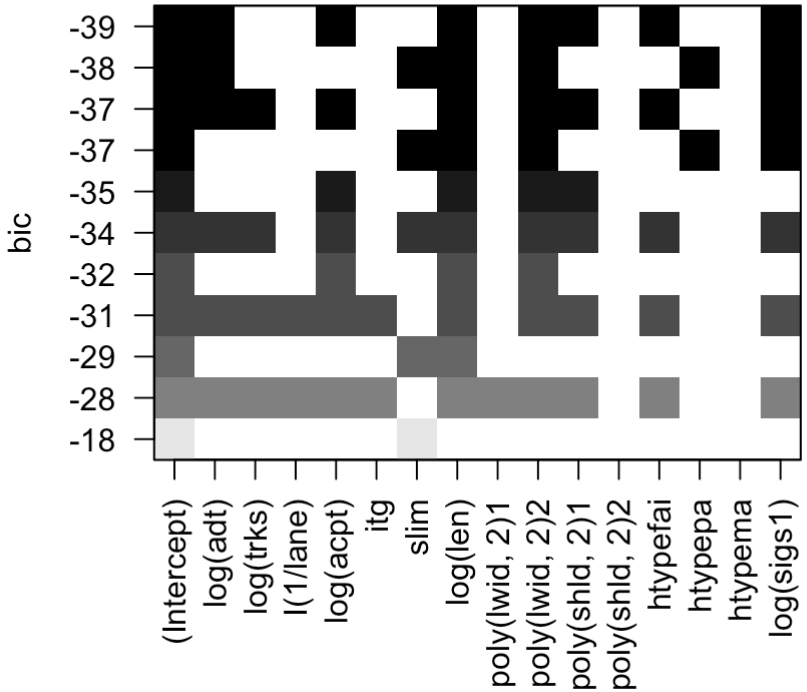
Investigating the results

```
plot(regfit_full, scale = "adjr2")
```

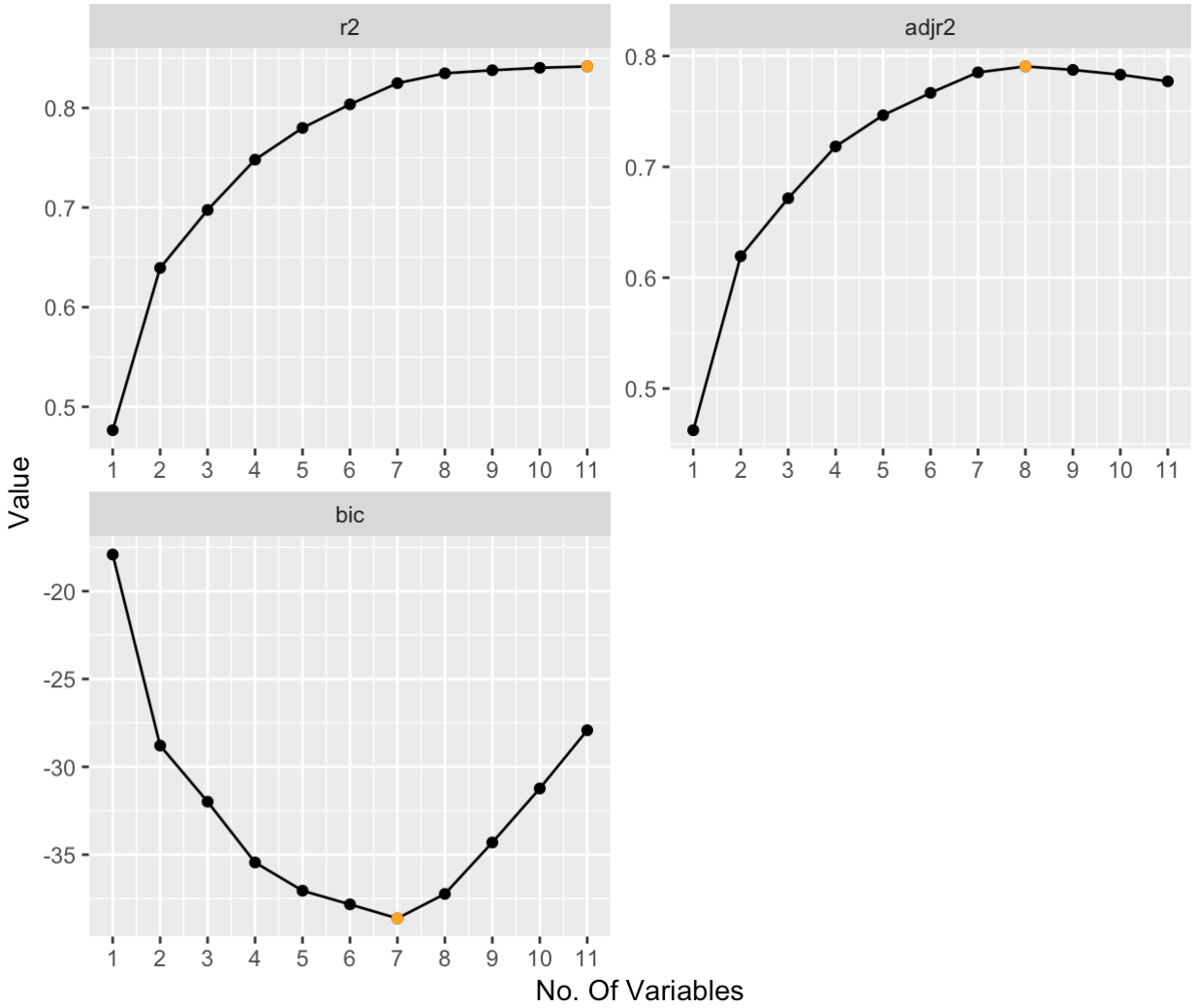


Investigating the results

```
plot(regfit_full, scale = "bic")
```



Another plot option



Extracting goodness-of-fit measures

```
broom::glance(step_hwy)
```

```
##   r.squared adj.r.squared     sigma statistic      p.value df   logLik
## 1 0.7962987     0.7419783 0.2351582   14.6593 1.865432e-08  9 6.229883
##        AIC      BIC deviance df.residual
## 1 7.540235 24.17585 1.658981          30
```

Extract R^2_{adj}

```
broom::glance(step_hwy)$adj.r.squared
```

```
## [1] 0.7419783
```

Calculate AIC

```
# The first number is equiv. d.f., the second is AIC
extractAIC(step_hwy, k = 2)
```

```
## [1]    9.000 -105.137
```

Calculate AICc

```
n <- nrow(Highway)
nslope <- length(step_hwy$coefficients) - 1
extractAIC(step_hwy, k = 2) + 2 * (nslope + 1) * (nslope + 2) / (n - nslope - 1)
```

```
## [1]  15.00000 -99.13697
```

Calculate BIC

```
extractAIC(step_hwy, k = log(n))
```

```
## [1]   9.00000 -90.16492
```

Training and test data sets

```
# Select rows for a training data set
train_id <- sample(1:nrow(df), size = round((2/3) * nrow(df)))

# Create the training and test data sets
train <- df[train_id,]
test  <- df[-train_id,]
```