Linear Models - Start to Finish

Cody Flagg

June 24th, 2014

Strategy:

1) Explore Data Structure
2) Explore Data Patterns (EDA)
3) Fitting Linear Model(s) 3b) Assessing Linear Models
4) Comparing Multiple Models
5) Examining Variable Importance and Model Predictions 5a) Predict New, or original, data to understand model results 5b) Examine scaled model parameters

This example is from a book on Mixed Effects Modeling in R. The specific data example is described below.

Note: a "GAM" is a "generalized additive model", a special case of a linear regression. We will analyze this dataset with a simple multiple linear regression (MLR) rather than a GAM to start with, as GAMs are a bit more complicated. The most basic difference between a GAM and an MLR is that MLRs essentially fit "straight lines" to the predictor variables and can use categorical data, whereas a GAM can fit non-linear curves to predictor variables and cannot use categorical data.

The Goal: Explore which variables drive species richness, and by how much.

(from page 63 of Zuur's "Mixed Effects Models..." book)

In this section, we show how confusing GAM becomes if you ignore this step of avoiding correlated explanatory variables. We use a plant vegetation data set for illustration. Sikkink et al. (2007) analysed grassland data from a monitoring programme from two temperate communities in Montana, USA: Yellowstone National Park and National Bison Range. The aim of the study was to determine whether the biodiversity of these bunchgrass communities changed over time and if they did, whether the changes in biodiversity relate to specific environmental factors. Here, we use the Yellowstone National Park data. Sikkink et al. (2007) quantified biodiversity using species richness to summarise the large number of species: ninety species were identified in the study. Richness is defined as the different number of species per site. The data were measured in eight different transects and each transect was measured repeatedly over time with time intervals of about four to ten years. For the moment, we ignore the temporal aspects of the data. And, instead of using all 20 or so explanatory variables, we use only those explanatory variables that Sikkink et al. (2007) identified as important. Figure 3.18 shows a scatterplot of all the variables used in this section. The response variable is species richness for the 64 observations, and the explanatory variables are rock content (ROCK), litter content (LITTER), bare soil (BARESOIL), rainfall in the fall (FallPrec), and maximum temperature in the spring (SprTmax). The correlation between ROCK and LITTER is reasonably high with a Pearson correlation of -0.7.

```
# data source:
# Alain Zuur Book: http://www.highstat.com/book2.htm
# the "~" indicates a relative path i.e. a folder path that is potentially shared across computers, but
veg <- read.table("~/GitHub/AvalonSoilProject/code_challenges/Vegetation.txt", header = T)
```

1) Explore Data Structure

- What are the variable names?
- What are some summary statistics?

- Are there any structural issues e.g. missing values, NA's, mixed negative and positive values, categorical variables??

```
names(veg)
```

```
## [1] "SAMPLEYR" "Time"     "Transect" "Richness" "ROCK"     "LITTER"
## [7] "BARESOIL" "FallPrec" "SprTmax"
```

```
summary(veg)
```

```
##     SAMPLEYR         Time         Transect       Richness
##  Min.   :1958   Min.   :1.00   Min.   :1.00   Min.   : 5.000
##  1st Qu.:1966   1st Qu.:2.75   1st Qu.:2.75   1st Qu.: 8.000
##  Median :1978   Median :4.50   Median :4.50   Median :10.000
##  Mean   :1978   Mean   :4.50   Mean   :4.50   Mean   : 9.966
##  3rd Qu.:1990   3rd Qu.:6.25   3rd Qu.:6.25   3rd Qu.:12.000
##  Max.   :2002   Max.   :8.00   Max.   :8.00   Max.   :18.000
##                                               NA's   :6
##       ROCK           LITTER         BARESOIL         FallPrec
##  Min.   : 0.00   Min.   : 5.00   Min.   : 0.000   Min.   :  9.90
##  1st Qu.: 7.25   1st Qu.:17.00   1st Qu.: 7.875   1st Qu.: 32.00
##  Median :18.50   Median :23.00   Median :16.250   Median : 43.43
##  Mean   :20.99   Mean   :22.85   Mean   :17.595   Mean   : 55.95
##  3rd Qu.:27.00   3rd Qu.:28.75   3rd Qu.:27.000   3rd Qu.: 72.64
##  Max.   :59.00   Max.   :51.00   Max.   :42.000   Max.   :153.41
##  NA's   :6       NA's   :6       NA's   :6        NA's   :6
##     SprTmax
##  Min.   : 8.91
##  1st Qu.:10.59
##  Median :11.74
##  Mean   :12.12
##  3rd Qu.:13.27
##  Max.   :16.92
##  NA's   :6
```

- Now, we can remove NA's
- Not always a good idea, as this will remove ANY ROW with an NA, so you could be tossing out rows that still have data in other columns.
- In this case, they are missed sampling years.

```
veg <- na.omit(veg) # remove NA's, mostly years with no data
```

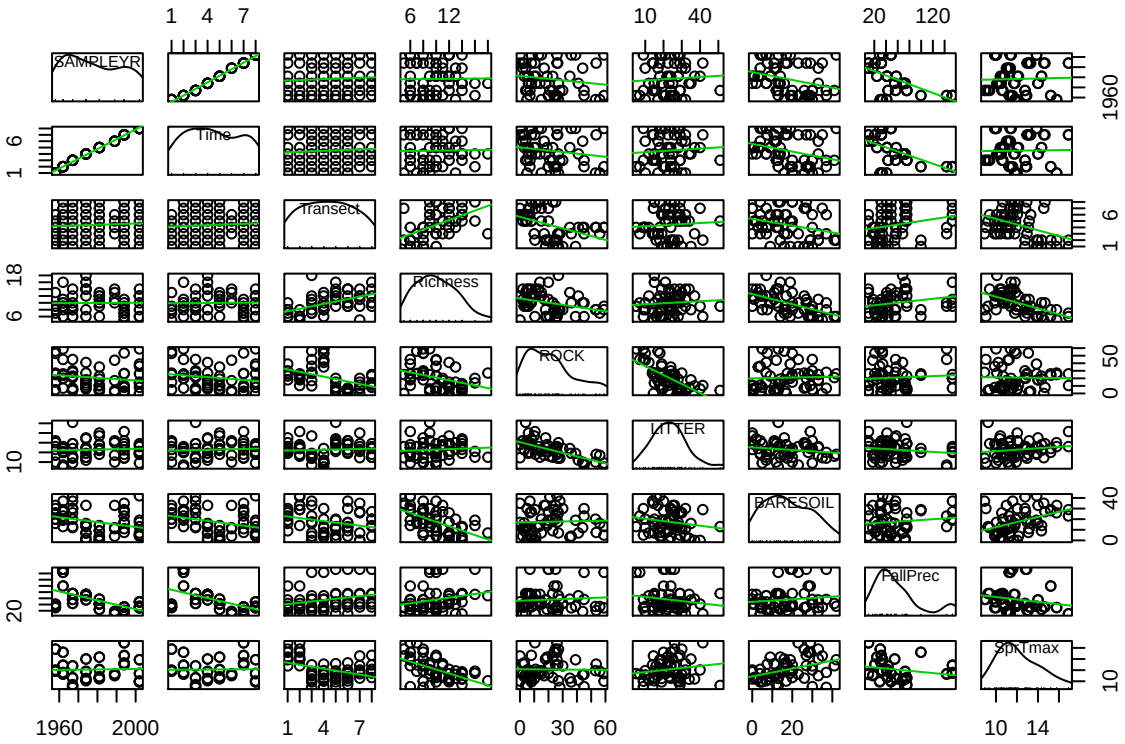
2) Exploratory Data Analysis (EDA): foundations of building new variables and models

What to look for:

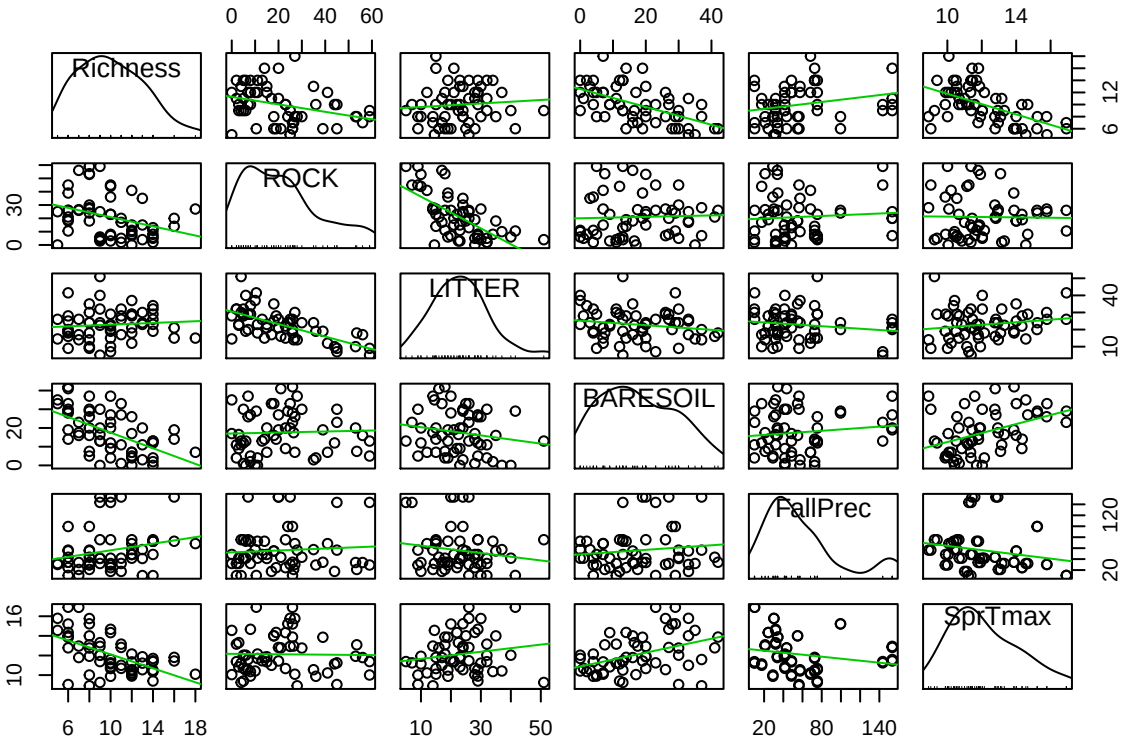
- Variables with dense, tightly clustered distributions - these need transforming
- Variables with lots of variation - these are sometimes better for prediction than variables with low variance

- Variables that are collinear - these variables should never be in the same model (they will 'inflate' the importance of a variable)
- Outliers
- Are there distinct groups or clouds of datapoints? This may imply another unseen physical structure to the data i.e. data derived from different ecosystems or treatments
- Are there lots of zeros or perhaps ordinal (ranked) rather than continuous data?
- Are observations independent? i.e. is there a time, spatial, or relational component driving variations?

```
library(car)

# do all variables - have to use "as.formula" to leave quotes out - can also use to copy-paste a nicer
all_var = as.formula(paste("~", paste(names(veg),collapse = "+")))

# can print 'all_var' and copy-paste the text as input for a model without having to re-type all of the

# plot this
scatterplotMatrix(all_var, data = veg, reg.line = lm, smoother = FALSE)
```




```
scatterplotMatrix(~Richness + ROCK + LITTER + BARESOIL + FallPrec + SprTmax, data = veg, smoother = FALS
```



Conclusions from EDA:

- ROCK and LITTER are highly correlated, they should not be included together in a single model.
- BARESOIL and SprTmax appear to have some correlation, but the scatter seems much larger.
- Several outliers appear at the edges of quite a few scatterplots.

Build a Model

Model 1 - one variable regression

```
# How strongly does spring max temperature drive species richness?
m1.temp <- lm(Richness ~ SprTmax, data = veg) # fit the model
summary(m1.temp) # examine the summary statistics
```

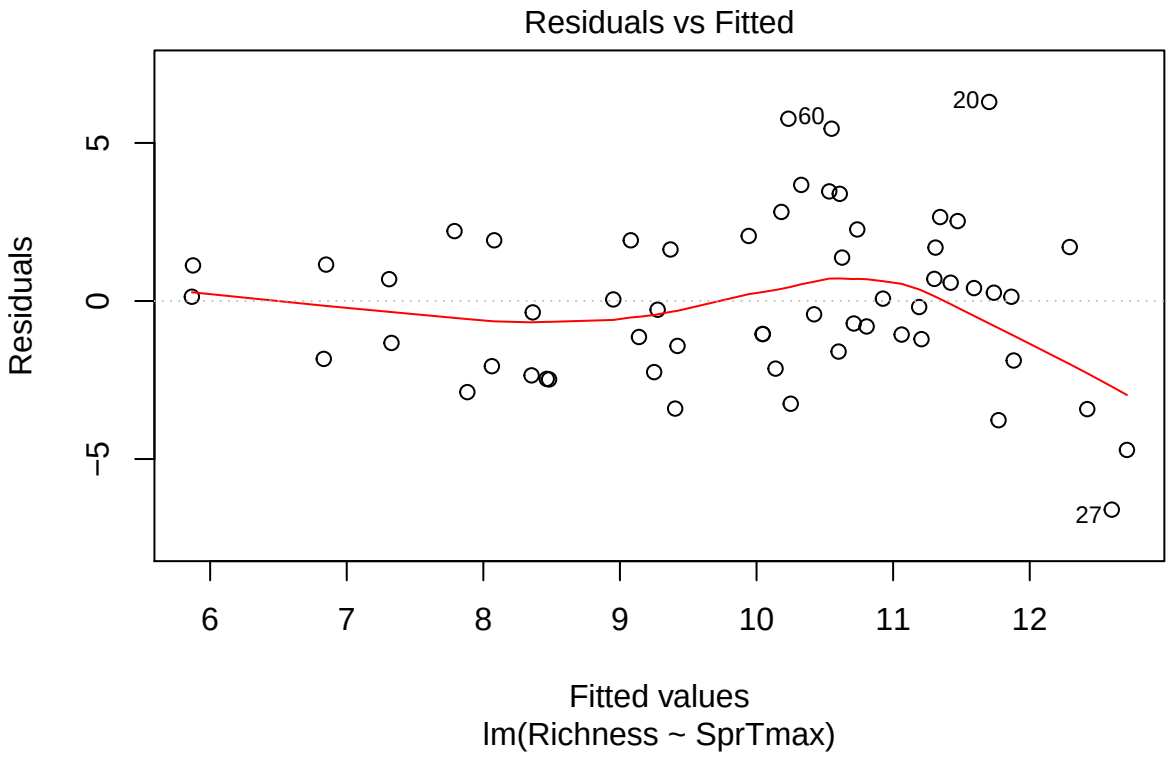
```
##
## Call:
## lm(formula = Richness ~ SprTmax, data = veg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6006 -1.7740 -0.0709  1.7028  6.2968
##
## Coefficients:
```

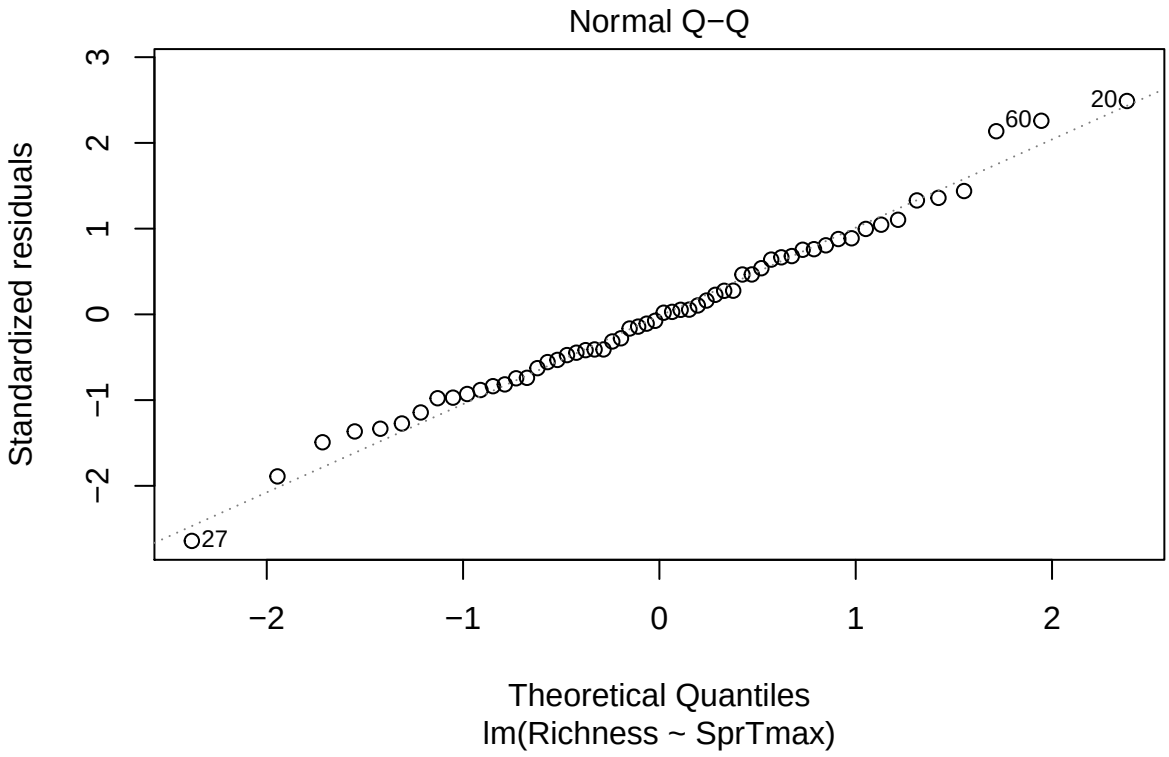
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.3266     2.1240   9.570 2.19e-13 ***
## SprTmax      -0.8546     0.1730  -4.941 7.39e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.577 on 56 degrees of freedom
## Multiple R-squared:  0.3036, Adjusted R-squared:  0.2912
## F-statistic: 24.42 on 1 and 56 DF,  p-value: 7.39e-06
```

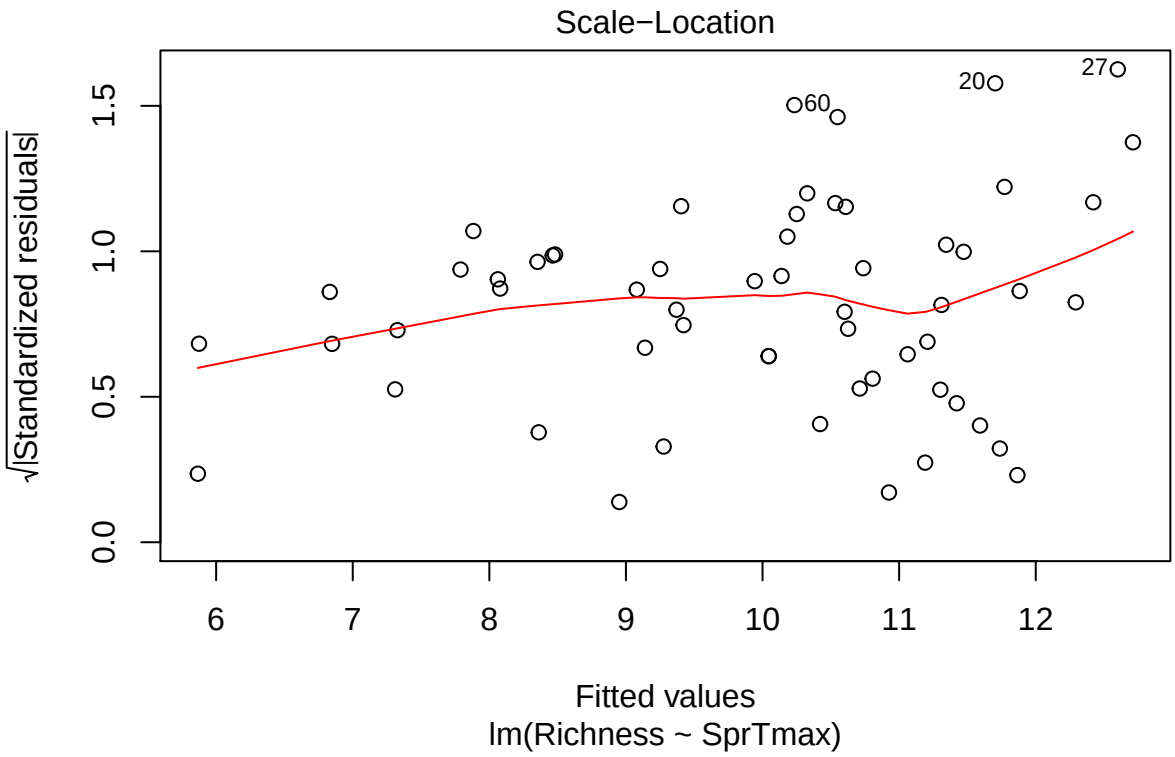
Model 1 - Model Diagnostics

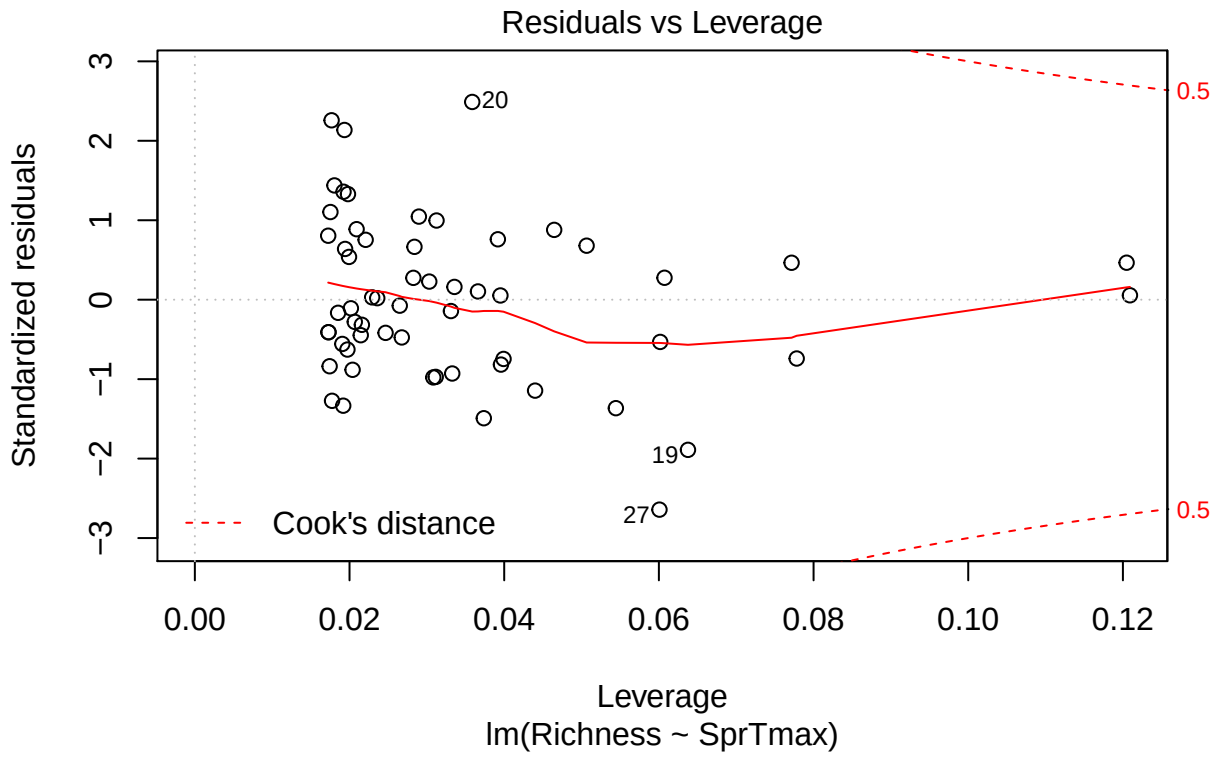
Things to Look For * *Residuals* should be homogeneous or without strong patterns (i.e. the amount of variance should be similar across all values) * This plot is good for examining the independence of the data (Qian, pg. 137) * *Normal Q-Q* plot should show most points falling along the 1-1 line. If there is large deviation at the tails, you may need to transform variables. * This plot is good for checking that the residuals are normally distributed (Qian, pg. 137) * *Scale-Location* shows how large squared residuals are for a specific value; numbered points highlight potential outliers. * *Residuals vs. Leverage* is similar to the last plot; points that fall outside of Cook's distance have too much leverage as outliers and may need to be removed. * **Fitted Values Minus Mean vs. Residuals-Fitted** rfsplot() in R * Qian pg.139, figure 5.9

```
plot(m1.temp)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Richness ~ SprTmax)
60
20
27
5
0
−5
6
7
8
9
10
11
12


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Richness ~ SprTmax)
60
20
27
3
2
1
0
−1
−2
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
6
7
8
9
10
11
12
60
20
27
Fitted values
lm(Richness ~ SprTmax)

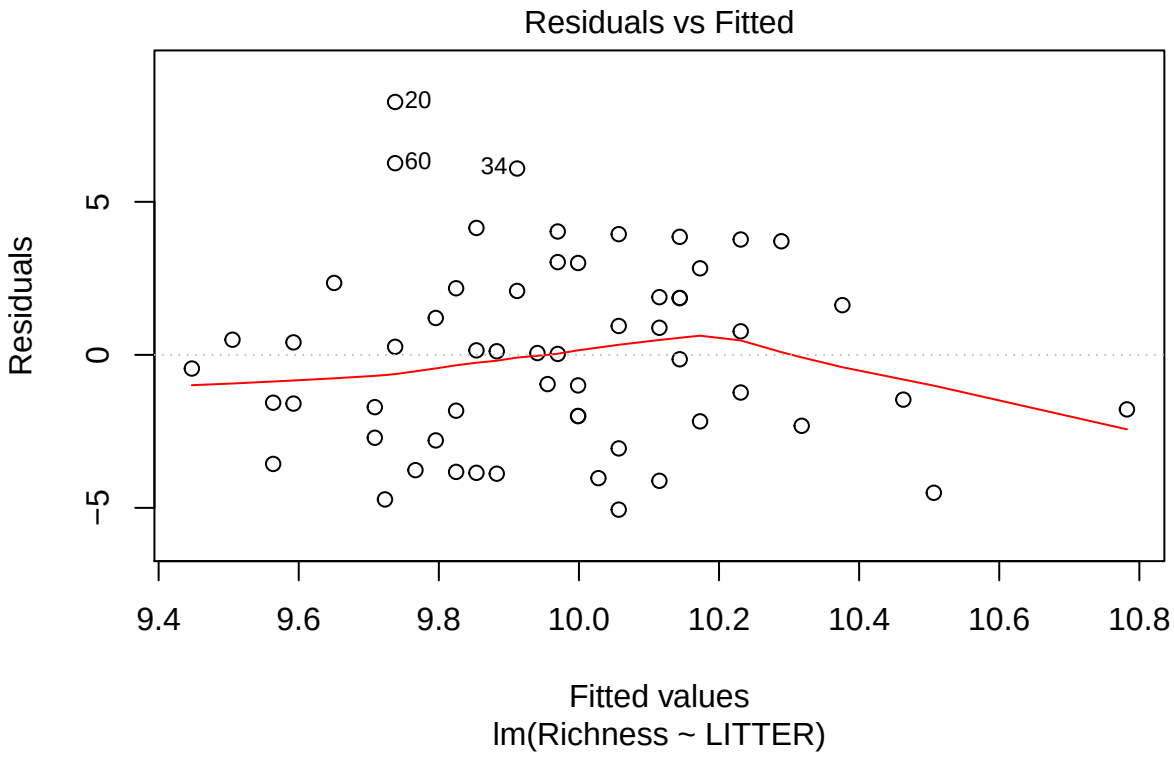


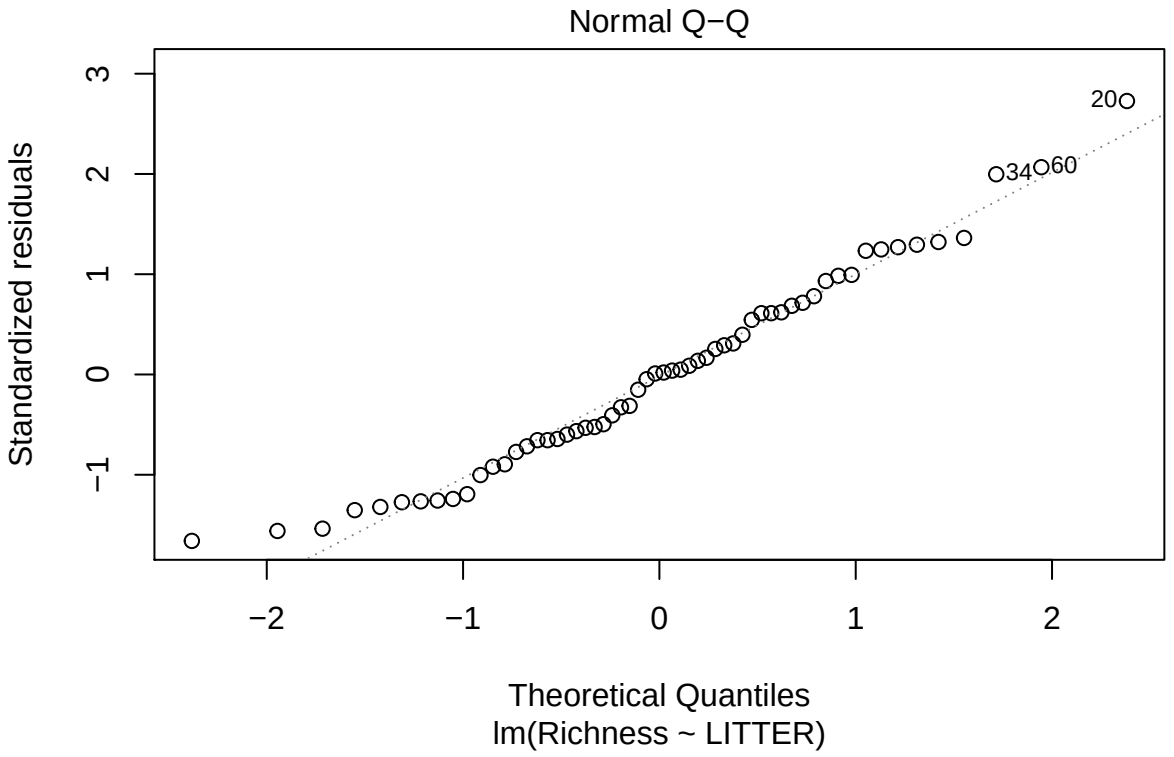
Model 2 - Build another single-variable model for comparison purposes

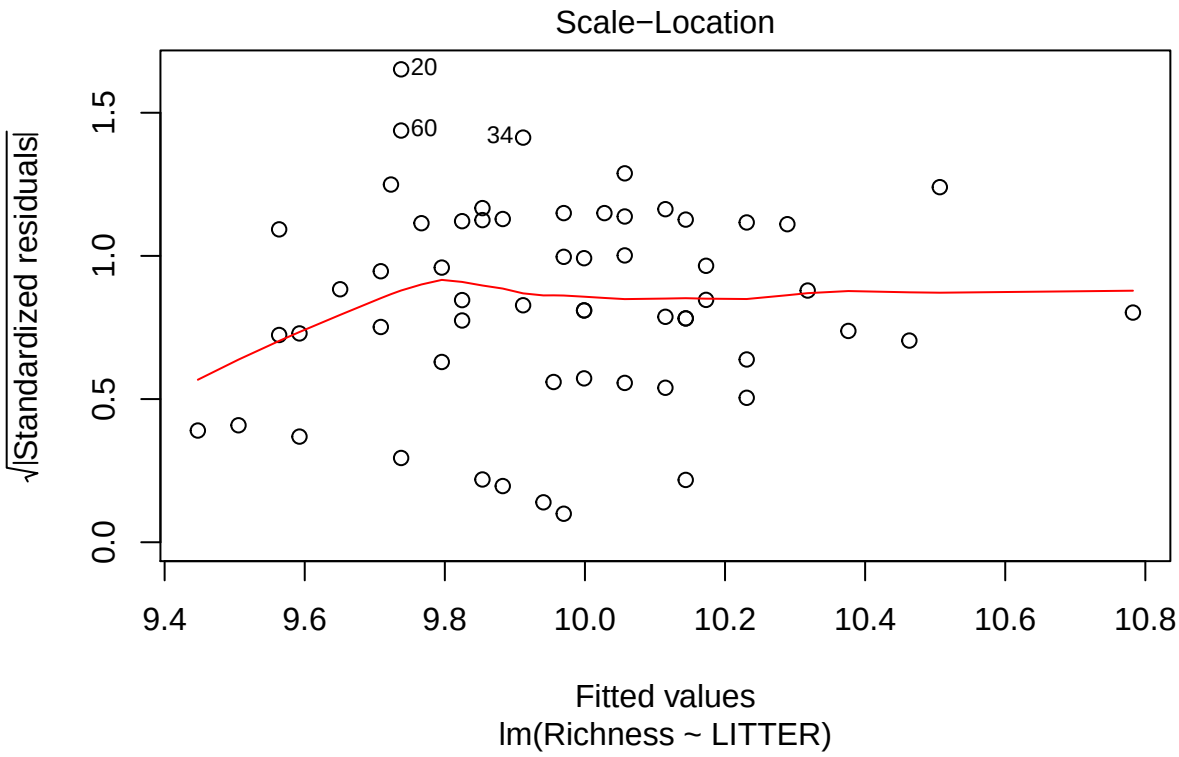
```
m1.litter <- lm(Richness ~ LITTER, data = veg)
summary(m1.litter)
```

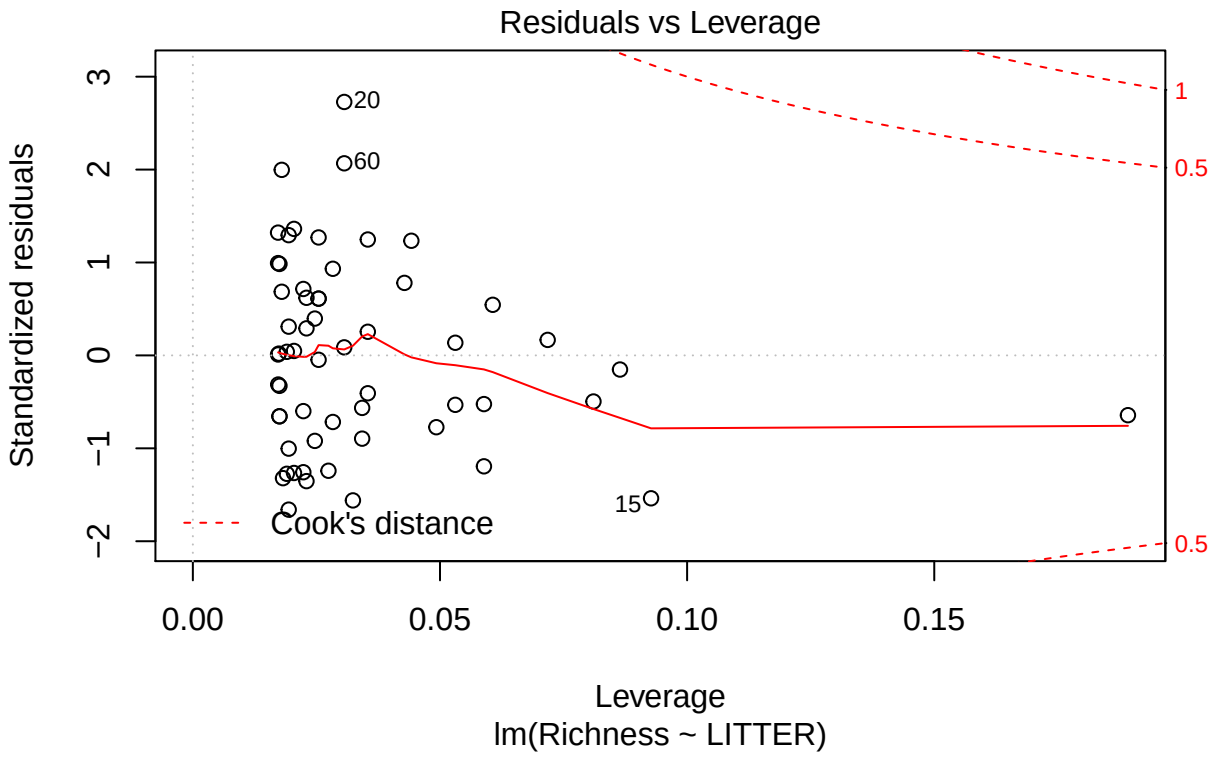
```
##
## Call:
## lm(formula = Richness ~ LITTER, data = veg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0568 -2.1294  0.0447  2.0375  8.2624
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.30232    1.11181   8.367 1.93e-11 ***
## LITTER       0.02902    0.04532   0.640    0.525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.077 on 56 degrees of freedom
## Multiple R-squared:  0.007267,   Adjusted R-squared:  -0.01046
## F-statistic: 0.4099 on 1 and 56 DF,  p-value: 0.5246
```

```r
plot(m1.litter)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Richness ~ LITTER)
20
60
34
5
0
−5
9.4
9.6
9.8
10.0
10.2
10.4
10.6
10.8


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Richness ~ LITTER)
20
34
60
3
2
1
0
−1
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
20
60
34
9.4
9.6
9.8
10.0
10.2
10.4
10.6
10.8
Fitted values
lm(Richness ~ LITTER)



Now compare the two models - a few ways to do this

Method 1: Akaike's Information Criterion (AIC)

AIC is used to compare models based on their complexity (i.e. the number of parameters) and their overall fit (i.e. the residuals or leftover variance). More complex models are penalized more than simpler models. AIC produces a single value that can be used to compare across models. Te AIC value itself is subject and does not matter on its own, it must be compared to other similar models where a lower AIC value means a "better" model.

AIC = $2k$ - 2ln*(model likelihood)

Where,

k = the number of parameters/predictors ln = natural log likelihood = the maximum likelihood value of the model

AIC can be applied to Linear regression:

AIC = 2k + n Log(RSS/n)

Where,

n = the number of observations RSS = the residual sum of squares

```
AIC(m1.temp, m1.litter)
```

```
##           df      AIC
## m1.temp    3 278.3587
## m1.litter  3 298.9236
```

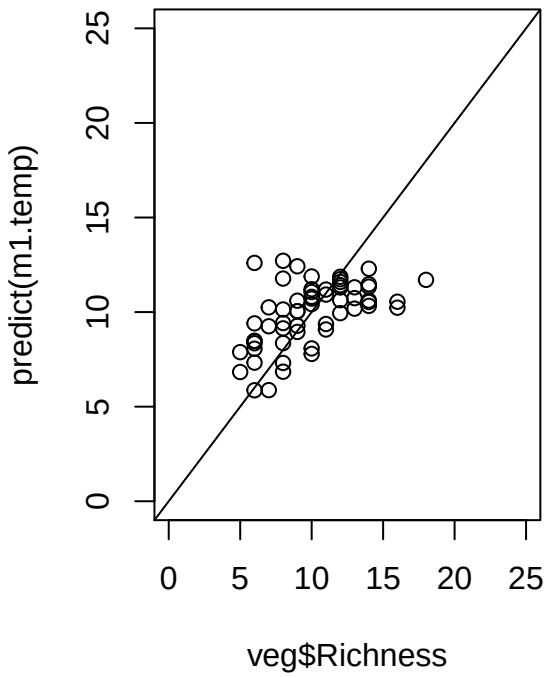
Method 2: Examining Model Predictions

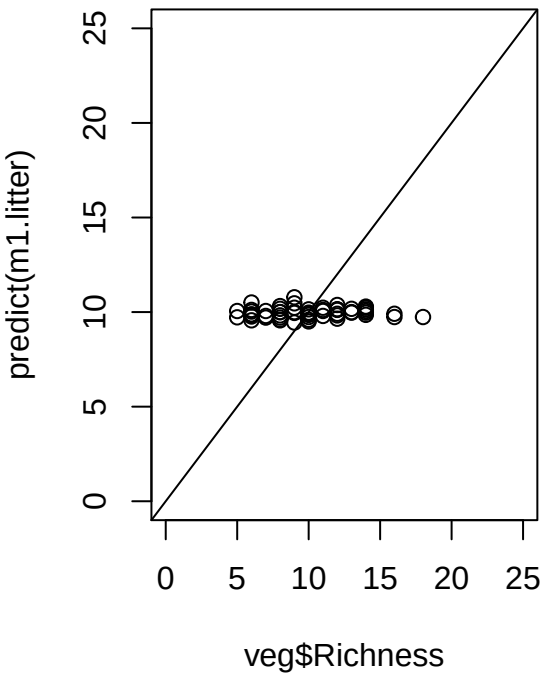
- Can our model predict our own data? If not, it's probably not a good model.
- All model objects in R should be able to produce predictions with predict()
- Also called "fitted values" i.e. predictions of your response variable
- Using predict() with no newdata will produce predictions with the original data

```
par(mfrow=c(1,2))
# plot the temperature model
predict(m1.temp) # gives the predicted values
```

```
##         1         2         3         4         5         7         8
##  6.848856  7.327458  9.421343  8.361581  8.079548  5.874559  8.481232
##         9        10        11        12        13        15        16
##  6.831763  7.310366  9.404250  8.353034  8.062454  5.866012  8.464139
##        17        18        19        20        21        22        23
## 10.250350 10.711860 12.711733 11.703250 11.421216 10.925521  9.250414
##        24        25        26        27        28        29        30
## 11.882726 10.139246 10.600755 12.600629 11.592146 11.310112 10.805871
##        31        32        33        34        35        36        37
##  9.139309 11.771621 10.181979 10.549477 12.421153 11.472496 11.190462
##        38        39        40        41        42        43        44
## 10.737499  9.079484 11.865633 10.045235 10.421280 12.292957 11.344298
##        45        46        47        48        50        51        52
## 11.062265 10.609302  8.951287 11.737436  9.370064 11.301566 10.327269
##        53        55        56        58        59        60        61
## 10.045235  7.882978 10.626395  9.276053 11.207555 10.233257  9.942678
##        63        64
##  7.788968 10.532384
```

```
plot(predict(m1.temp)~veg$Richness, xlim=c(0,25), ylim=c(0,25)) # plot against observed richness values
abline(0,1) # the 1:1 line

# plot the litter model
plot(predict(m1.litter)~veg$Richness, xlim=c(0,25), ylim=c(0,25))
abline(0,1)
```





Assessing Variable Importance

- Build a multiple regression model (i.e. more than one predictor variable)

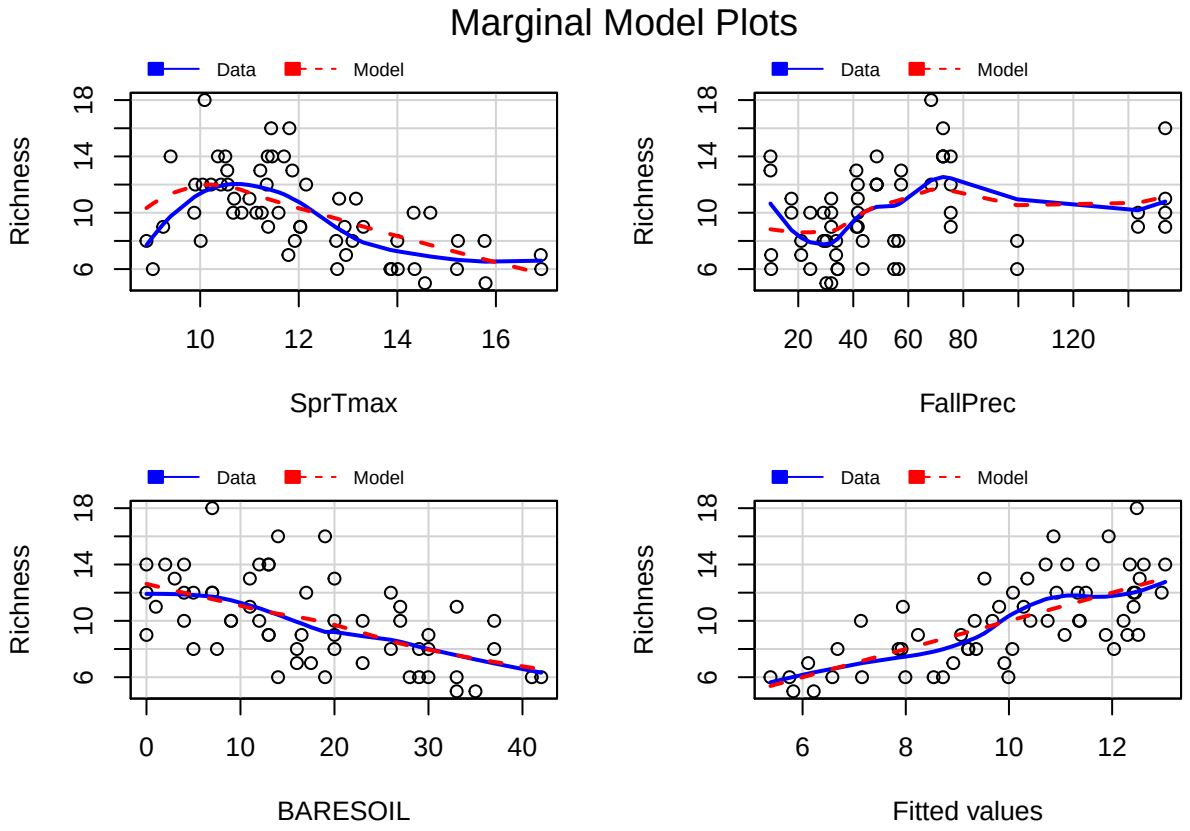
```
#
m2.all <- lm(Richness ~ SprTmax + FallPrec + BARESOIL, data = veg)
summary(m2.all)
```

```
##
## Call:
## lm(formula = Richness ~ SprTmax + FallPrec + BARESOIL, data = veg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0363 -1.3713  0.0048  1.1617  5.5224
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.029634   2.102132   8.101 6.78e-11 ***
## SprTmax     -0.492307   0.173243  -2.842 0.006318 **
## FallPrec     0.018481   0.008145   2.269 0.027291 *
## BARESOIL    -0.121040   0.029377  -4.120 0.000131 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.26 on 54 degrees of freedom
## Multiple R-squared:  0.4835, Adjusted R-squared:  0.4548
## F-statistic: 16.85 on 3 and 54 DF,  p-value: 7.529e-08
```

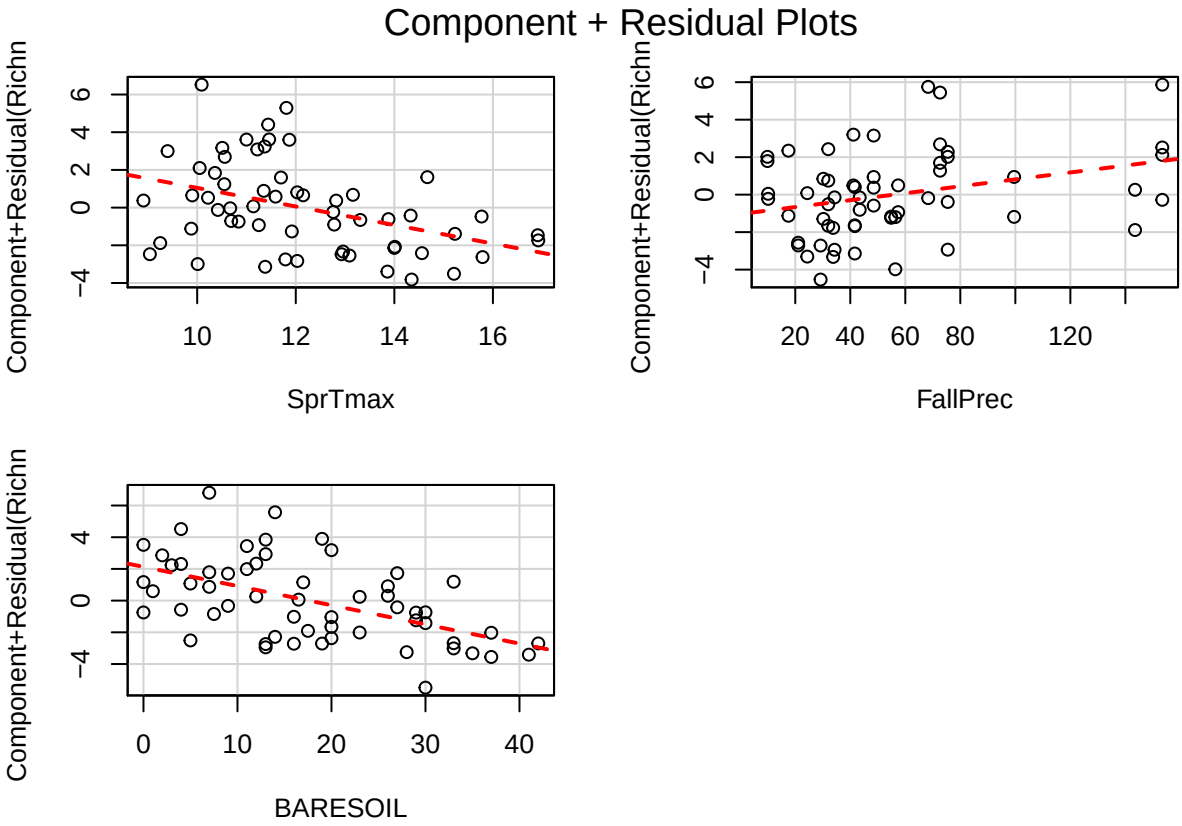
- Various diagnostic prediction plots can show you how much of an effect a predictor will have on the response
- **Marginal Plots** show you how each predictor influences the response *on its own* i.e. without the other variables present

```
car::marginalModelPlots(m2.all) #only marginal plots
```



- **Conditional Plots** show you how much each predictor influences the response *while accounting for* the other predictors

```
# visualize effects - this looks similar to a "partial regression/residual plot"
# http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/partregr.htm
car::crPlots(m2.all, smoother = NULL) # component+residual plots i.e. partial residuals
```




```
##### Other useful functions
# car::avPlots(m2.all) # conditional plots i.e. added-variable plots
# car::influencePlot(m2.all)
# car::leveragePlot(m2.all)
# car::mcPlot()
```

- Finally, you can visualize the effects of each variable by looking at the model predictions from different, perhaps new, data
- Spring Max Temp looks to be a pretty influential variable, what happens to Richness if the max temp reaches 20 degrees C in the future?

```
par(mfrow=c(1,1))
# create a new data frame with an expanded temperature range
# keep fall precip and baresoil constant, or else we have to make multiple graphs
nd = with(veg, expand.grid(SprTmax = seq(8, 22, by = 2),
                           FallPrec = mean(FallPrec),
                           BARESOIL = mean(BARESOIL)))


# make predictions with "new" data
temp_pred = predict(m2.all, newdata = nd, se.fit = TRUE)

# append to previous data frame
pred1 = data.frame(SprTmax = nd$SprTmax, Richness = temp_pred$fit, Richness.se = temp_pred$se.fit)
```

```r
# now plot the predicted richness with 95% confidence intervals
plot(Richness ~ SprTmax, data = pred1, type = "l") # the predicted richness
upp = pred1$Richness + 1.92*pred1$Richness.se # Upper conf interval, 1.92 is the t-value for a 95% conf
lwr = pred1$Richness - 1.92*pred1$Richness.se # lower conf interval
lines(upp ~ pred1$SprTmax, lty = 2, col = "red") # add the upper line
lines(lwr ~ pred1$SprTmax, lty = 2, col = "red") # add the lower line
```