16 - Linear Regression

ST 560 | Fall 2017
University of Alabama

16-regression.pdf

Contents

1 Goals of Regression

There are two primary goals for using regression analysis

1. Inference about parameters
   - How does advertising budget affect sales?
   - Estimate effect of X on Y, controlling for other explanatory factors.
2. Prediction
   - How accurately can sales be predicted given a certain sales budget?
   - Use whatever it takes (transformations, new variables, etc.) to get better predictions.

These are different goals and drive potentially different model specifications.

1.1 Modeling in general

Models are a way to summarize data. Linear regression models, in particular, are a family of models that impose a *linear* structure between the predictor and response variables.

See the RDS Models chapter of the textbook for some good information on modelling basics.

The free book An Introduction to Statistical Learning

> This book provides an introduction to statistical learning methods. It is aimed for upper level undergraduate students, masters students and Ph.D. students in the non-mathematical sciences. The book also contains a number of R labs with detailed explanations on how to implement the various methods in real life settings, and should be a valuable resource for a practicing data scientist.

2 Simple Linear Regression

2.1 Advertising Data

Consider some advertising data:

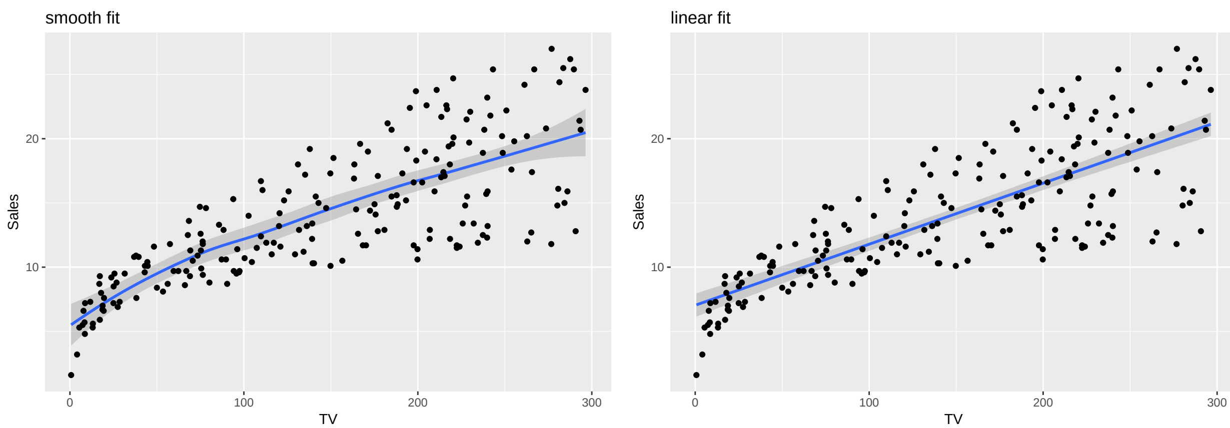
```
#- load advertising data (drop 1st column of row names)
advert = read_csv("http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv") %>%
  select(-1)         # remove first column of rownames
summary(advert)
#>        TV             Radio          Newspaper          Sales
#>  Min.   :  0.70   Min.   : 0.000   Min.   :  0.30   Min.   : 1.60
#>  1st Qu.: 74.38   1st Qu.: 9.975   1st Qu.: 12.75   1st Qu.:10.38
#>  Median :149.75   Median :22.900   Median : 25.75   Median :12.90
#>  Mean   :147.04   Mean   :23.264   Mean   : 30.55   Mean   :14.02
#>  3rd Qu.:218.82   3rd Qu.:36.525   3rd Qu.: 45.10   3rd Qu.:17.40
#>  Max.   :296.40   Max.   :49.600   Max.   :114.00   Max.   :27.00
```

These data give the sales of a product (in thousands of units) under advertising budgets (in thousands of dollars) of TV, Radio, and Newspaper. This was most likely *observational* data (not experimental) which limits the conclusions we can make from modeling.

We can start by examining the relationship between the `TV` budget and `Sales` using scatterplots with a *linear* fit:

```
#- left (smooth)
ggplot(advert, aes(TV, Sales)) + geom_smooth() +
  geom_point()  + ggtitle("smooth fit")

#- right (linear)
ggplot(advert, aes(TV, Sales)) + geom_smooth(method="lm") +
  geom_point()  + ggtitle("linear fit")
```



2.2 Simple (univariate) Linear Regression Model

A *simple* linear regression model is one with a single explanatory variable

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- β_0 is the intercept
- β_1 is the slope
- We will use training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ to estimate the model parameters. (this is the same data used to make a scatterplot)

2.3 Estimating model parameters (coefficients)

In the advertising data, let's consider how `Sales` are related to the `TV` budget. The linear model is

$$\text{sales} = \beta_0 + \beta_1 \times TV + \epsilon$$

and the fitted, predictive model is

$$\widehat{\text{sales}} = \hat{\beta}_0 + \hat{\beta}_1 \times TV$$

and we wish to find estimate the parameters, $\hat{\beta}_0, \hat{\beta}_1$ such that $\widehat{\text{sales}}$ is close to the actual sales for any given value of `TV` budget.

2.4 Using `lm()` for fitting linear regression models

In R, the `lm()` function creates (and estimates) a *linear model*.

```
lm.TV = lm(Sales~TV, data=advert)
```

Notice a few things:

- This produces the `lm` object `lm.TV`. This is basically a list, but structured so it can be used easily in other functions.
- The formula interface `Sales~TV` makes `Sales` the response/dependent variable and `TV` the predictor/independent variable
- The `data=advert` provides the data

We can do lots with the `lm.TV` object:

```
summary(lm.TV)      # gives a summary of the linear model
#>
#> Call:
#> lm(formula = Sales ~ TV, data = advert)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -8.3860 -1.9545 -0.1913  2.0671  7.2124
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
#> TV          0.047537   0.002691   17.67   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 3.259 on 198 degrees of freedom
#> Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
#> F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
summary(lm.TV)$r.squared  # R squared value
#> [1] 0.6118751
coef(lm.TV)      # model coefficients (the betas)
#> (Intercept)          TV
#>  7.03259355  0.04753664
confint(lm.TV, level=0.95)  # 95% confidence interval of coefficients
#>                  2.5 %     97.5 %
#> (Intercept) 6.12971927 7.93546783
#> TV          0.04223072 0.05284256
```

So we see the *fitted* linear model is:

$$\widehat{\text{sales}} = 7.03 + 0.048 \times TV$$

2.5 Using `predict()` to make predictions

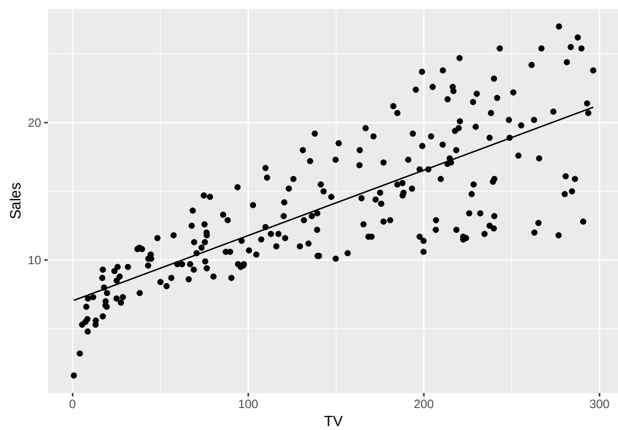
Once we have a model, the `predict()` function will give predictions. We have to pass in the model and the data for making predictions (as a data frame)

```
est.sales = predict(lm.TV, newdata = data.frame(TV = advert$TV))

advert2 = advert %>% mutate(est.sales)
```

Now we can replicate the `geom_smooth(method='lm')` call

```
ggplot(advert2, aes(x=TV)) +
  geom_point(aes(y=Sales)) +
  geom_line(aes(y=est.sales))
```



2.6 More univariate models

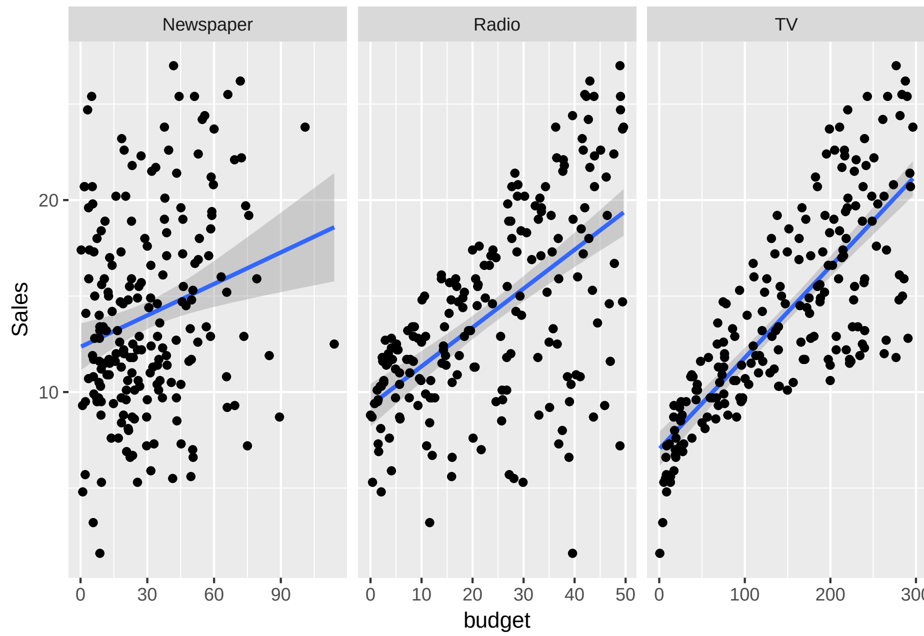
Just change the name of the predictor variable to get the models for the other available predictors

```
lm.TV = lm(Sales~TV, data=advert)
lm.Radio = lm(Sales~Radio, data=advert)
lm.Newspaper = lm(Sales~Newspaper, data=advert)
```

And if we wanted to plot all three, we could use faceting if we first convert the data into long form

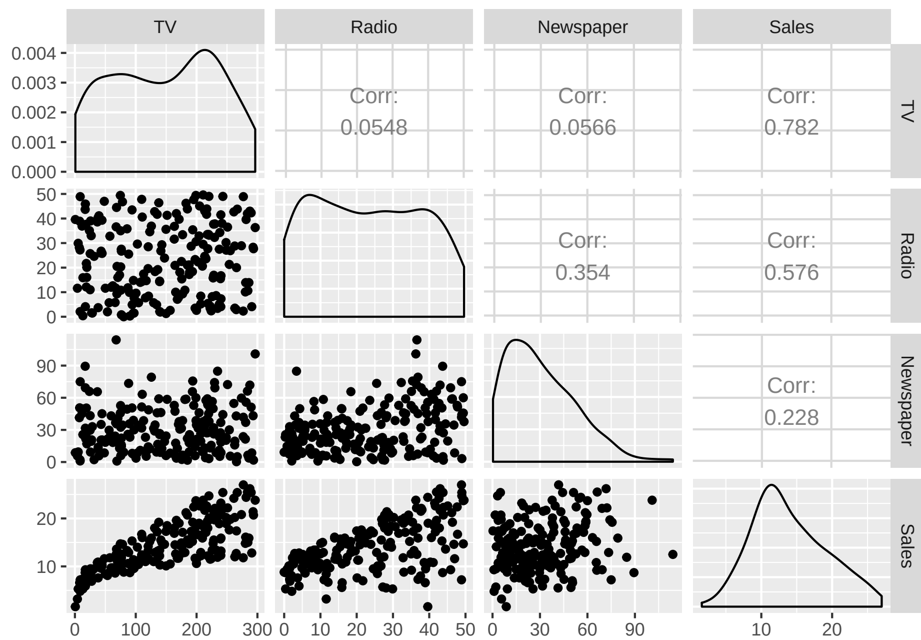
```
advert_long = gather(advert,
                     key="channel",
                     value="budget",
                     -Sales)

ggplot(advert_long, aes(x=budget, y=Sales)) +
  geom_smooth(method="lm") + geom_point() +
  facet_wrap(~channel, scales="free_x" )
```



2.7 Multivariate Considerations

```
library(GGally)
ggpairs(advert)       # also see pairs(advert) for a base R version
```



See the http://ggobi.github.io/ggally/ help page for more fun examples of cool plots.

3 Multiple Linear Regression

3.1 Linear Regression

The standard general form for linear regression is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots, +\beta_p X_p + \epsilon$$

- Y is the response or dependent variable
- $X_1, X_2, \ldots, X_p$ are called the p explanatory, independent, or predictor variables
- the greek letter ϵ (epsilon) is the random error variable

Training data is used to estimate the model *parameters* or *coefficients*.

$$\left[\begin{array}{cccccc|c} x_{11} & x_{12} & \cdot & \cdot & \cdot & x_{1p} & y_1 \\ x_{21} & x_{22} & \cdot & \cdot & \cdot & x_{2p} & y_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & x_{n2} & \cdot & \cdot & \cdot & x_{np} & y_n \end{array}\right]$$

Producing the predictive model:

$$\hat{y}(x_1, x_2, \ldots, x_p) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots, +\hat{\beta}_p x_p$$

- where $\hat{\beta}_j$ are the weights assigned to each variable
- these weights are the values the minimize the residual sum of squares (RSS) for predicting the training data

3.2 Multiple Components

Consider the advertising sales model that uses all three predictors

$$\text{sales} = \beta_0 + \beta_1 \times (\text{TV}) + \beta_2 \times (\text{radio}) + \beta_3 \times (\text{newspaper}) + \text{error}$$

In R, the formula would be `Sales ~ TV + Radio + Newspaper` (the order of the predictor variables does not matter).

```
lm.all = lm(Sales ~ TV + Radio + Newspaper, data=advert)
summary(lm.all)
#>
#> Call:
#> lm(formula = Sales ~ TV + Radio + Newspaper, data = advert)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -8.8277 -0.8908  0.2418  1.1893  2.8292
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
#> TV           0.045765   0.001395  32.809   <2e-16 ***
#> Radio        0.188530   0.008611  21.893   <2e-16 ***
#> Newspaper   -0.001037   0.005871  -0.177     0.86
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 1.686 on 196 degrees of freedom
#> Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
#> F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

Notice that we have a big increase in the R^2 and reduction in RSE indicating that this model with all three terms does better at fitting the training data than the models with only a single predictor.

However, notice that the p-value for the `Newspaper` coefficient is not small (and no significance stars). Maybe `Newspaper` is not very helpful once `TV` and `Radio` are in the model?

Let's consider using only these two variables (just remove `Newspaper`)

```
lm.TVRadio = lm(Sales ~ TV + Radio, data=advert)
summary(lm.TVRadio)
#>
#> Call:
#> lm(formula = Sales ~ TV + Radio, data = advert)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -8.7977 -0.8752  0.2422  1.1708  2.8328
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  2.92110    0.29449   9.919   <2e-16 ***
#> TV           0.04575    0.00139  32.909   <2e-16 ***
```

```
#> Radio        0.18799    0.00804  23.382   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 1.681 on 197 degrees of freedom
#> Multiple R-squared:  0.8972, Adjusted R-squared:  0.8962
#> F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

3.2.1 Adjusted R^2

We want to compare a model with four estimated parameters (`lm.all`) to a model with only three estimated parameters (`lm.TVRadio`). By adding an additional parameter the R^2 will necessarily increase. It is better to consider the RSE or adjusted R^2.

The adjusted R^2 penalizes (reduces) the R^2 to account for the number of parameters that need to be estimated

$$R^2_{\text{adj}} = 1 - \frac{RSS}{TSS}\left(\frac{n-1}{n-p-1}\right)$$

The larger the adjusted R^2, R^2_{adj}, the better the model.

- The adjusted R^2 for `lm.all` = `0.8956` and for `lm.TVRadio` = `0.8962`
- Because the model `lm.TVRadio` has the (slightly) larger R^2_{adj} it provides a reason to prefer it over the full model.
- When the values are this close, choose the simpler (less coefficients to estimate) model
  - What if we had a confidence interval for `R^2_{adj}`?
  - It is an estimate of a population parameter, so we can get a confidence interval, or test to see if the values differ between models
- Perhaps the company will stop spending money on `Newspaper` advertising? Should they?
  - If we believe our regression model is true, then yes. But for such observational data, proceed with caution. Check assumptions and try more models before making such a decision.

4 Extending the Linear Model

4.1 Removing the Additive Structure

We have found that the best model so far is the one that uses `TV` and `Radio` to predict the value of `Sales`.

Specifically, the least squares model is:

$$\widehat{\text{sales}} = 2.921 + 0.046 \times (\text{TV}) + 0.188 \times (\text{radio})$$

- So a one unit increase in `TV` would suggest a 0.046 unit increase in `Sales`, no matter the budget allocated to `Radio`
- But what if spending money on `Radio` advertising actually increases the effectiveness of the `TV` advertising?
  - So `TV` effects should increase as `Radio` increases

  - E.g., spending 1/2 of a $100,000 budget on `TV` and `Radio` may increase `Sales` more than allocating the entire amount to only `TV` or only `Radio`
  - In marketing, this is the *synergy* effect. In statistics, this is known as an interaction effect.

4.1.1 Interaction Effect

Consider the linear regression model with two variables and an interaction effect

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$$

This model relaxes the additive structure, while maintaining the linear structure. Consider the equation re-written

$$\begin{aligned} Y &= \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2 + \epsilon \\ &= \beta_0 + \tilde{\beta}_1 X_1 + \beta_2 X_2 + \epsilon \end{aligned}$$

where $\tilde{\beta}_1 = (\beta_1 + \beta_3 X_2)$.

- Since $\tilde{\beta}_1$ changes with X_2, the effect of X_1 on Y is no longer constant.
  - Adjusting X_2 will change the impact of X_1 on Y

In R, use the notation `X_1:X_2` to include an interaction effect:

```
lm.synergy = lm(Sales ~ TV + Radio + TV:Radio, data=advert)
summary(lm.synergy)
#>
#> Call:
#> lm(formula = Sales ~ TV + Radio + TV:Radio, data = advert)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -6.3366 -0.4028  0.1831  0.5948  1.5246
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 6.750e+00  2.479e-01  27.233   <2e-16 ***
#> TV          1.910e-02  1.504e-03  12.699   <2e-16 ***
#> Radio       2.886e-02  8.905e-03   3.241   0.0014 **
#> TV:Radio    1.086e-03  5.242e-05  20.727   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.9435 on 196 degrees of freedom
#> Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
#> F-statistic:  1963 on 3 and 196 DF,  p-value: < 2.2e-16
```

Your Turn #1

1. Do you think the addition of an interaction helped? Why?

4.1.2 Predicted values in R

In R, the `predict()` function will return the predicted values from a fitted regression model. Besides the model, the function needs the X values (the `newdata` argument) for making predictions.
- Type `?predict.lm` to read the help pages
- The `object` argument is the `lm` model
- The `newdata` must be a `data.frame`

```
# predict the Sales for a budget with TV = 50 ($50,000) and Radio = 20 ($20,000)
predict(lm.TVRadio, newdata=data.frame(TV=50,Radio=20))
#>        1
#> 8.968725
```

Your Turn #2

1. Suppose the company has a total advertising budget of $100,000. What is the predicted `Sales` for the following scenarios using the `lm.TVRadio` model
   a. All budget to `TV`
   b. All budget to `Radio`
   c. Half budget to each
2. Suppose the company has a total advertising budget of $100,000. What is the predicted `Sales` for the following scenarios using the `lm.synergy` (interaction) model
   a. All budget to `TV`
   b. All budget to `Radio`
   c. Half budget to each
3. How would you recommend investment?

4.2 Transforming Variables

In R, it is easy to manipulate the models. Here we can try some common transformations

```r
#- Transforming predictors
lm(Sales ~ log(TV) + Radio , data=advert)
#>
#> Call:
#> lm(formula = Sales ~ log(TV) + Radio, data = advert)
#>
#> Coefficients:
#> (Intercept)      log(TV)        Radio
#>     -9.1343       3.9338       0.2054

lm(Sales ~ log(TV) + sqrt(Radio) , data=advert)
#>
#> Call:
#> lm(formula = Sales ~ log(TV) + sqrt(Radio), data = advert)
#>
#> Coefficients:
#> (Intercept)      log(TV)  sqrt(Radio)
#>     -11.659        3.901        1.662

#- Transforming Response variable
lm(log(Sales) ~ TV + Radio , data=advert)
#>
#> Call:
#> lm(formula = log(Sales) ~ TV + Radio, data = advert)
#>
#> Coefficients:
#> (Intercept)           TV        Radio
#>    1.745078     0.003673     0.011985
#  Warning: if you transform the response variable, you can no longer
#   compare to other non-transformed models using Rsq, etc.
```

4.2.1 Formula Specification in R

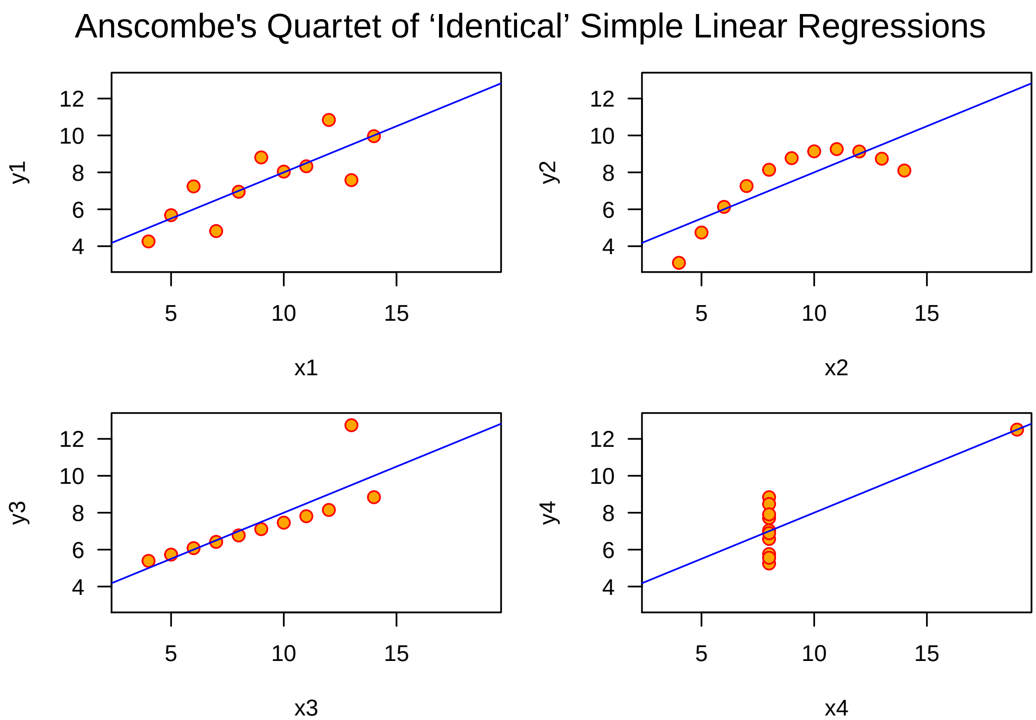
R provides a flexible formula interface for trying different model specifications. Here is a good resource http://faculty.chicagobooth.edu/richard.hahn/teaching/FormulaNotation.pdf

5 Regression Diagnostics

5.1 Topics

1. Checking for non-linearity
2. Correlation of error terms
3. Non-constant variance of error terms
4. Outliers
5. High Leverage points
6. Collinearity

5.1.1 Anscombe's Quartet



5.2 In-sample vs. Out-of-Sample

Use a hold-out set or cross-validation to assess the performance of a model on future data

6 Logistic Regression

6.1 Logistic Regression

In R, use the `glm()` function with the `family = binomial` setting.

```
library(openintro)
data(email)
g <- glm(spam ~ to_multiple + winner + format, data=email,
         family = binomial)
summary(g)
#>
#> Call:
#> glm(formula = spam ~ to_multiple + winner + format, family = binomial,
#>     data = email)
#>
#> Deviance Residuals:
#>     Min       1Q   Median       3Q      Max
#> -1.3122  -0.3536  -0.3536  -0.3536   3.2057
#>
```

```
#> Coefficients:
#>              Estimate Std. Error z value Pr(>|z|)    
#> (Intercept) -1.18678    0.08229 -14.423  < 2e-16 ***
#> to_multiple -2.39135    0.30149  -7.932 2.16e-15 ***
#> winneryes    1.49826    0.29817   5.025 5.04e-07 ***
#> format      -1.55416    0.11571 -13.431  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> (Dispersion parameter for binomial family taken to be 1)
#> 
#>     Null deviance: 2437.2  on 3920  degrees of freedom
#> Residual deviance: 2168.6  on 3917  degrees of freedom
#> AIC: 2176.6
#> 
#> Number of Fisher Scoring iterations: 6
```