Chapter 1: Linear models

Big picture

This course builds on an understanding of the mechanics of linear models. Here we introduce some key topics that will facilitate future understanding hierarchical models.

Learning goals

- linear regression with `lm`
- intercepts, "categorical" effects
- varying model structure to estimate effects and standard errors
- interactions as variation in slope estimates for different groups
- centering input variables and intepreting resulting parameters
- assumptions and unarticulated priors
- understanding residual variance (Gaussian)
- understanding all of the above graphically
- understanding and plotting output of lm
- notation and linear algebra review: $X\beta$

Linear regression, ANOVA, ANCOVA, and multiple regression models are all species cases of general linear models (hereafter "linear models"). In all of these cases, we have observed some response variable y, which is potentially modeled as a function of some covariate(s) $x_1, x_2, ..., x_p$.

Model of the mean

If we have no covariates of interest, then we may be interested in estimating the population mean and variance of the random variable Y based on n observations, corresponding to the values $y_1, ..., y_n$. Here, capital letters indicate the random variable, and lowercase corresponds to realizations of that variable. This model is sometimes referred to as the "model of the mean".

```
# simulating a sample of y values from a normal distribution
y <- rnorm(20)
plot(y)
```

We have two parameters to estimate: the mean of Y, which we'll refer to as μ, and the variance of Y, which we'll refer to as σ^2. Here, and in general, we will use greek letters to refer to parameters. If Y is normally distributed, then we can assume that the realizations or samples y that we observe are also normally distributed: $y \sim N(\mu, \sigma^2)$. Here and elsewhere, the $\sim$ symbol represents that some quantity "is distributed as" something else (usually a probability distribution). You can also think of $\sim$ as meaning "is sampled from". A key concept here is that we are performing statistical inference, meaning we are trying to learn about (estimate) population-level parameters with sample data. In other words, we are not trying to learn about the sample mean $\bar{y}$ or sample variance of y. These can be calculated and treated as known once we have observed a particular collection of y values. The unknown quantities μ and σ^2 are the targets of inference.

Fitting this model (and linear models in general) is possible in R with the `lm` function. For this rather simple model, we can estimate the parameters as follows:

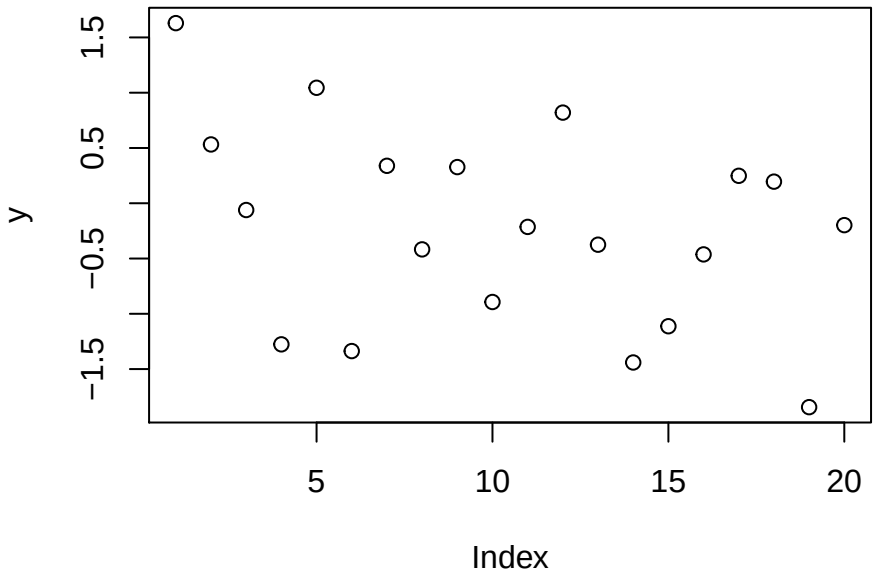

Figure 1: A set of observed y values, $n = 20$.

```r
# fitting a model of the mean with lm
m <- lm(y ~ 1)
summary(m)
```

```
##
## Call:
## lm(formula = y ~ 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.61972 -0.72338  0.01887  0.55496  1.85432
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2249     0.2024  -1.111     0.28
##
## Residual standard error: 0.905 on 19 degrees of freedom
```

The summary of our model object `m` provides a lot of information. For reasons that will become clear shortly, the estimated population mean is referred to as the "Intercept". Here, we get a point estimate for the population mean μ: -0.225 and an estimate of the residual standard deviation σ: 0.905, which we can square to get an estimate of the residual variance σ^2: 0.819.

Linear regression

Often, we are interested in estimating the mean of Y as a function of some other variable, say X. Simple linear regression assumes that y is again sampled from a normal distribution, but this time the mean or expected value of y is a function of x:

$$y_i \sim N(\mu_i, \sigma^2)$$

$$\mu_i = \alpha + \beta x_i$$

Here, subscripts indicate which particular value of y and x we're talking about. Specifically, we observe n pairs of values: $(y_i, x_i), ..., (y_n, x_n)$, with all x values known exactly. Linear regression models can equivalently be written as follows:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2)$$

Key assumptions here are that each of the error terms $\epsilon_1, ..., \epsilon_n$ are normally distributed around zero with some variance (i.e., the error terms are identically distributed), and that the value of ϵ_1 does not affect the value of any other ϵ (i.e., the errors are independent). This combination of assumptions is often referred to as "independent and identically distributed" or i.i.d. Equivalently, given some particular x_i and a set of linear regression parameters, the distribution of y_i is normal. A common misconception is that linear regression assumes the distribution of y is normal. This is wrong - linear regression assumes that the error terms are normally distributed. The assumption that the variance σ^2 is constant for all values of x is referred to as homoskedasticity. Rural readers may find it useful to think of skedasticity as the amount of "skedaddle" away from the regression line in the y values. If the variance is changing across values of x, then the assumption of homoskedasticity is violated and you've got a heteroskedasticity problem.

```
# simulate and plot x and y values
n <- 50
x <- runif(n)
alpha <- -2
beta <- 3
sigma <- .4
y <- rnorm(n, mean = alpha + beta * x, sd = sigma)
plot(x, y)

# add known mean function
lines(x = x, y = alpha + beta * x, col='blue')
legend('topleft',
       pch = c(1, NA), lty = c(NA, 1),
       col = c('black', 'blue'),
       legend = c('Observed data', 'E(y | x)'),
       bty = 'n')
```

The normality assumption means that the probability density of y is highest at the value $\alpha + \beta x$, where the regression line is, and falls off away from the line according to the normal probablity density. This graphically looks like a bell 'tube' along the regression line, adding a dimension along x to the classic bell 'curve'.

Model fitting

Linear regression parameters α, β, and σ^2 can be estimated with `lm`. The syntax is very similar to the previous model, except now we need to include our covariate `x` in the formula (the first argument to the `lm` function).

```
m <- lm(y ~ x)
summary(m)
```

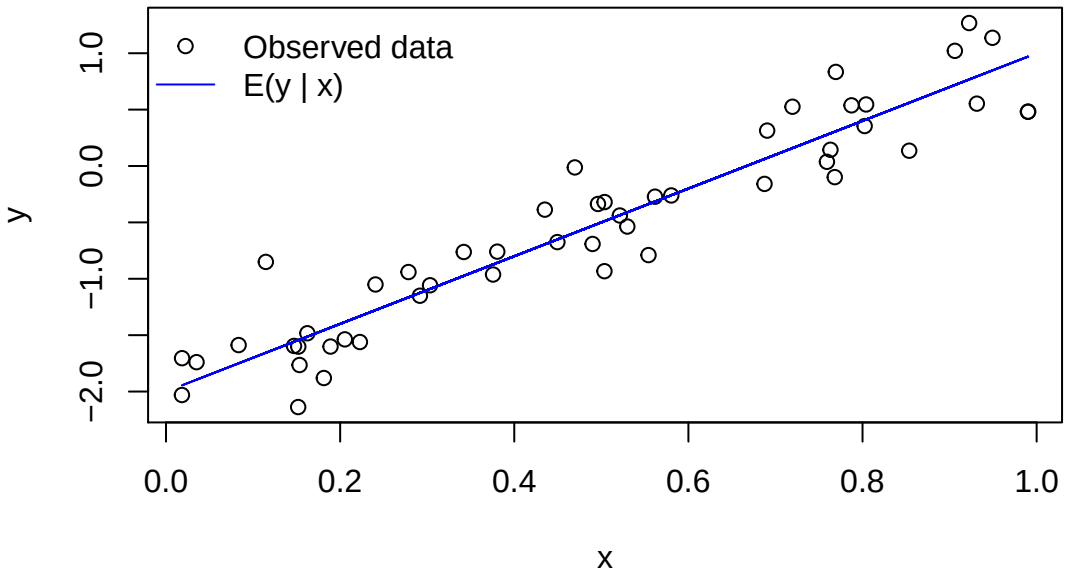

Figure 2: Simulated data from a linear regression model. The true expected value of y given x, E(y | x), is shown as a line.

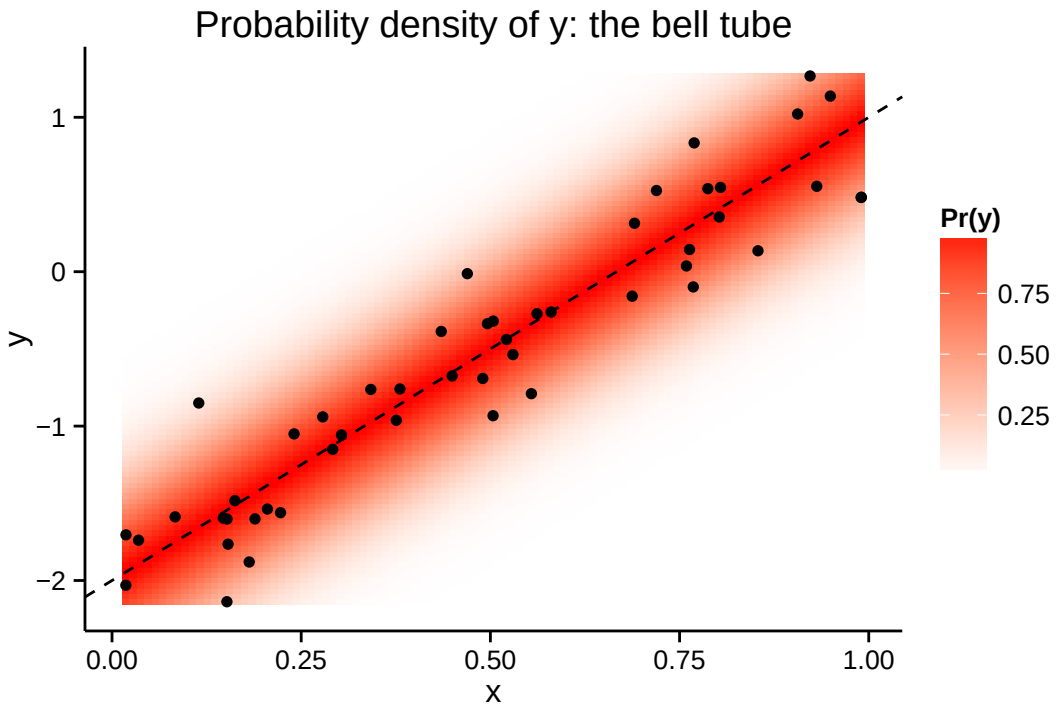

Figure 3: Graphical depiction of the linear regression normality assumption. The probability density of y is shown in color. Higher probabilities are shown as more intense colors, and regions with low probabilities are lighter.

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61678 -0.20203 -0.01286  0.20039  0.77915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.96539    0.08404  -23.39   <2e-16 ***
## x            2.92841    0.14866   19.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3058 on 48 degrees of freedom
## Multiple R-squared:  0.8899, Adjusted R-squared:  0.8876
## F-statistic:   388 on 1 and 48 DF,  p-value: < 2.2e-16
```

The point estimate for the parameter α is called "(Intercept)". This is because our estimate for α is the y-intercept of the estimated regression line when $x = 0$ (recall that $y_i = \alpha + \beta x_i + \epsilon_i$). The estimate for β is called "x", because it is a coefficient associated with the variable "x" in this model. This parameter is often referred to as the "slope", because it represents the increase in the expected value of y for a one unit increase in x (the rise in y over run in x). Point estimates for the standard deviation and variance of ϵ can be extracted as before (`summary(m)$sigma` and `summary(m)$sigma^2`).

Centering and scaling covariates

Often, it's a good idea to "center" covariates so that they have a mean of zero ($\bar{x} = 0$). This is achieved by subtracting the sample mean of a covariate from the vector of covariate values ($x - \bar{x}$). It's also useful to additionally scale covariates so that they are all on a common and unitless scale. While many will divide each covariate by its standard deviation, Gelman and Hill (pg. 57) recommend dividing by twice the standard deviation (s_x) so that binary covariates are transformed from $x \in \{0, 1\}$ to $x_t \in \{-0.5, 0.5\}$, where x_t is the transformed covariate: $x_t = \frac{x - \bar{x}}{2s_x}$. If covariates are not centered and scaled, then it is common to observe correlations between estimated slopes and intercepts.

So, we expect that in this case, the estimates for the intercept and slope must be negatively correlated. This is bourne out in the confidence region for our estimates of α and β. Usually, people inspect univariate confidence intervals for parameters, e.g.,

```
confint(m)
```

```
##                 2.5 %    97.5 %
## (Intercept) -3.6866135 -2.357159
## x            0.8354427  1.210665
```

This is misleading because our estimates for these parameters are correlated. For any given value of the intercept, there are only certain values of the slope that are supported. To assess this possibility, we might also be interested in the bivariate confidence ellipse for these two parameters. We can evaluate this quantity graphically as follows with some help from the `car` package:

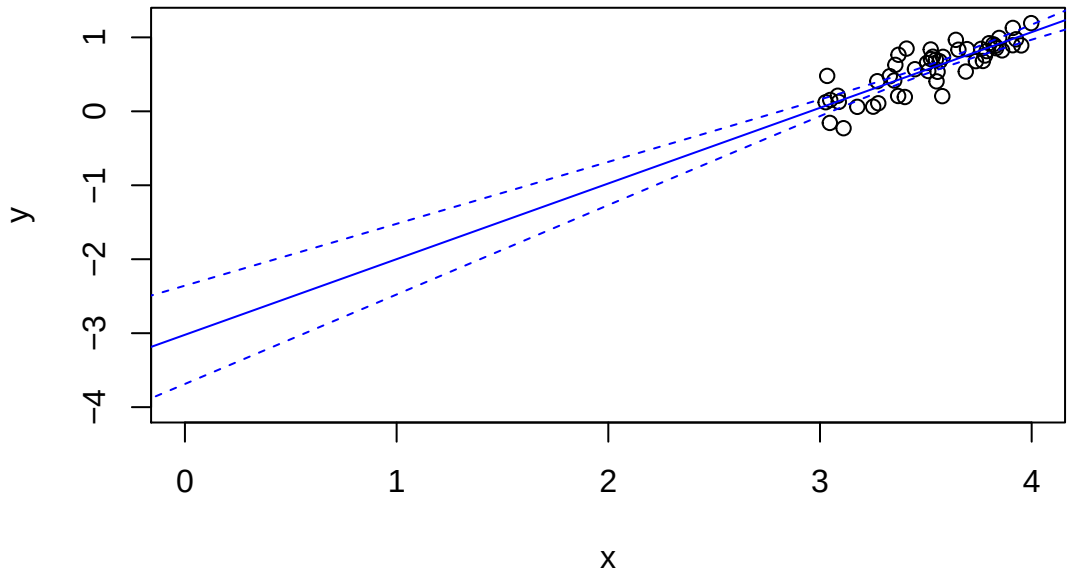

Figure 4: Linear regression line of best fit with 95% confidence intervals for the line. Notice that if the slope is lower (the upper dashed line) then the intercept necessarily goes up, and if the slope is higher (the lower dashed line), the intercept must decrease.

```
library(car)
confidenceEllipse(m)
```

This is not great. We want to be able to directly use the univariate confidence intervals. Our problem can be solved by centering x:

Now there is no serious correlation in the estimates and we are free to use the univariate confidence intervals without needing to consider the joint distribution of the slope and intercept. This trick helps with interpretation, but it will also prove useful later in the course in the context of Markov chain Monte Carlo (MCMC) sampling.

Checking assumptions

We have assumed that the distribution of error terms is normally distributed, and this assumption is worth checking. Below, we plot a histogram of the residuals (another name for the ϵ parameters) along with a superimposed normal probability density so that we can check normality.

```
hist(resid(m), breaks = 20, freq = F,
     main = 'Histogram of model residuals')
curve_x <- seq(min(resid(m)), max(resid(m)), .01)
lines(curve_x, dnorm(curve_x, 0, summary(m)$sigma))
```

Even when the assumption of normality is correct, it is not always obvious that the residuals are normally distributed. Another useful plot for assessing normality of errors is a quantile-quantile or Q-Q plot. If the residuals do not deviate much from normality, then the points in a Q-Q plot won't deviate much from the dashed one-to-one line. If points lie above or below the line, then the residual is larger or smaller, respectively, than expected based on a normal distribution.

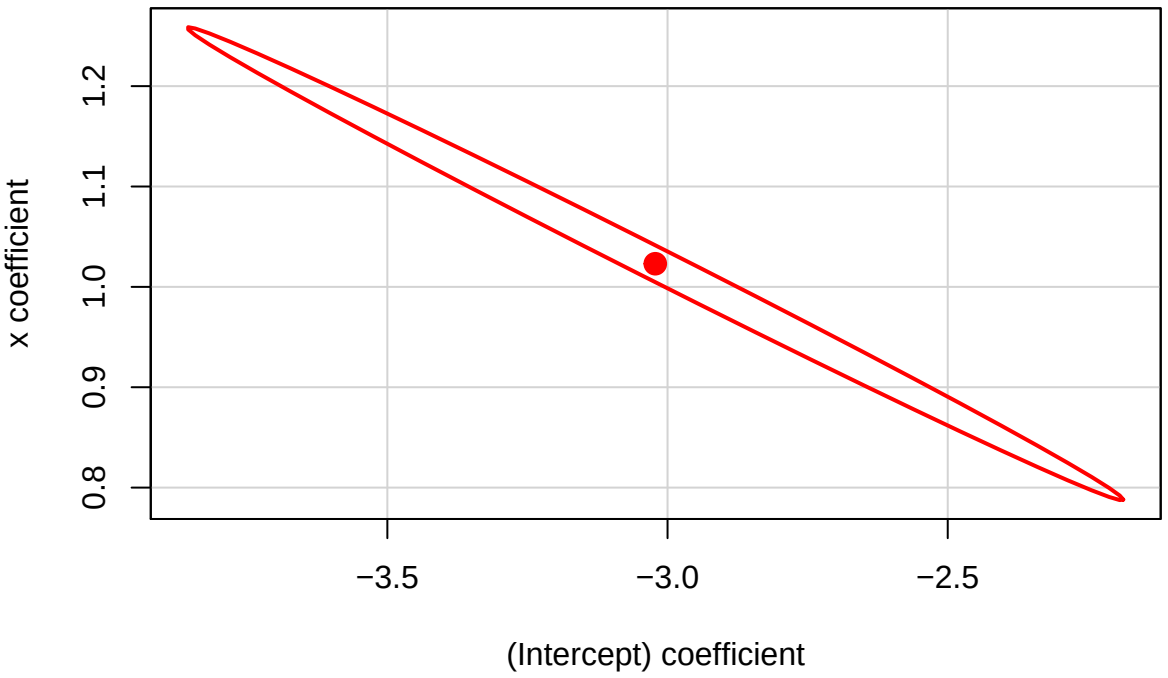

Figure 5: The bivariate confidence region for the slope and intercept. Correlation implies that for any particular value of alpha or beta, only a small subset of the values of the other parameter are supported. This information is not available by considering the univariate confidence intervals alone.

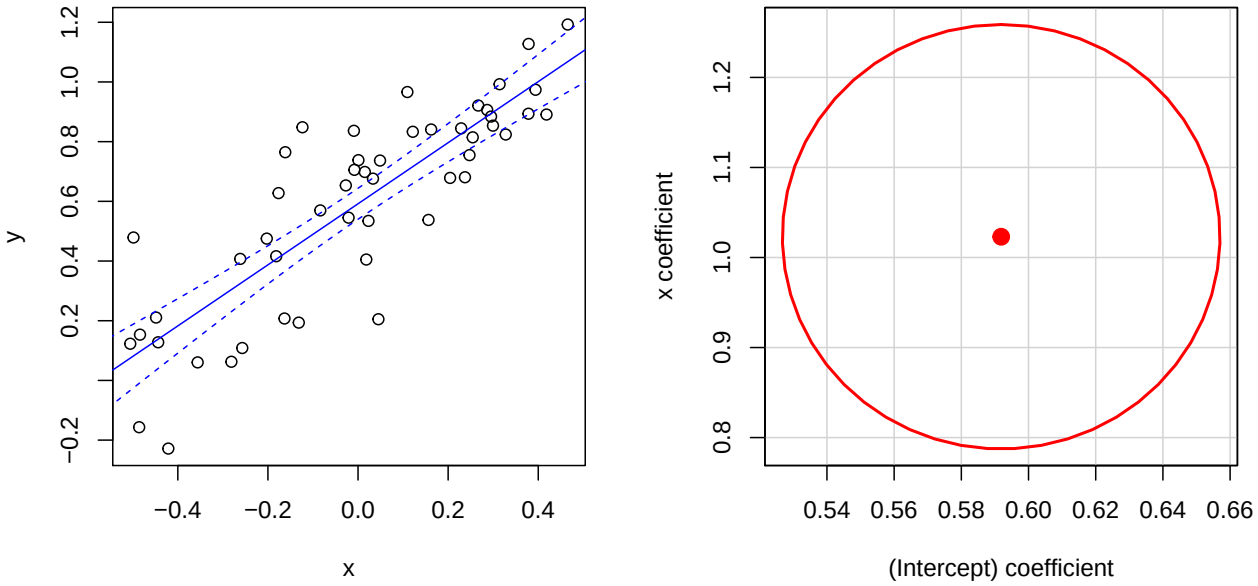

Figure 6: New line of best fit with confidence ellipses for the slope and intercept after centering the covariate x

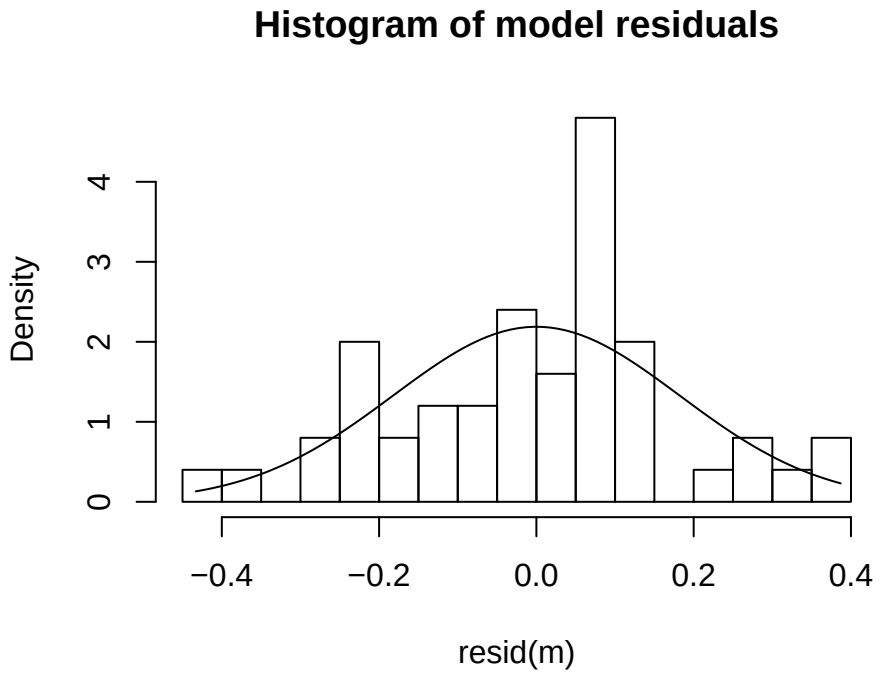

Figure 7: Simulated data from a linear regression model.

```
plot(m, 2)
```

To assess heteroskedasticity, it is useful to inspect a plot of the residuals vs. fitted values, e.g. `plot(m, 1)`. If it seems as though the spread or variance of residuals varies across the range of fitted values, then it may be worth worrying about homoskedasticity and trying some transformations to fix the problem.

Analysis of variance

Sometimes, the covariate of interest is not continuous but instead categorical (e.g., "chocolate", "strawberry", or "vanilla"). We might again wonder whether the mean of a random variable Y depends on the value of this covariate. However, we cannot really estimate a meaningful "slope" parameter, because in this case x is not continuous. Instead, we might formulate the model as follows:

$$y_i \sim N(\alpha_{j[i]}, \sigma^2)$$

Where α_j is the mean of group j, and we have J groups total. The notation $\alpha_{j[i]}$ represents the notion that the i^{th} observation corresponds to group j, and we are going to assume that all observations in the j^{th} group have the same mean, α_j. The above model is perfectly legitimate, and our parameters to estimate are the group means $\alpha_1, ..., \alpha_J$ and the residual variance σ^2. This parameterization is called the "means" parameterization, and though it is perhaps easier to understand than the following alternative, it is less often used.

This model is usually parameterized not in terms of the group means, but rather in terms of an intercept (corresponding to the mean of one "reference" group), and deviations from the intercept (differences between a group of interest and the intercept). For instance, in R, the group whose mean is the intercept (the "reference" group) will be the group whose name comes first alphabetically. Either way, we will estimate the same number of parameters. So if our groups are "chocolate", "strawberry", and "vanilla", R will assign the

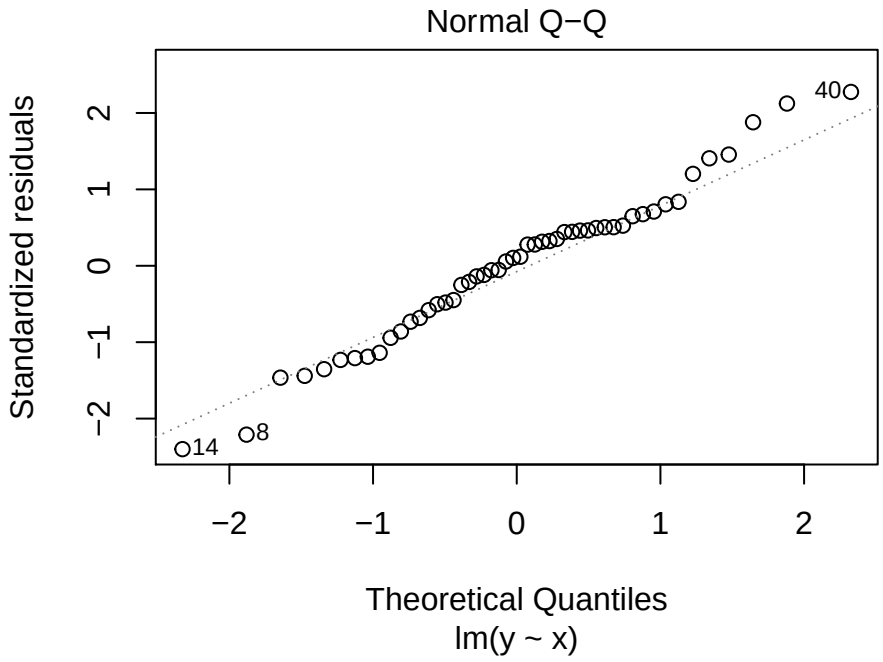

Figure 8: A quantile-quantile plot to assess normality of residuals.

group "chocolate" to be the intercept, and provide 2 more coefficient estimates for the difference between the estimated group mean of strawberry vs. chocoloate, and vanilla vs. chocolate.

This parameterization can be written as

$$y_i \overset{iid}{\sim} N(\mu_0 + \beta_{j[i]}, \sigma^2)$$

where μ_0 is the "intercept" or mean of the reference group, and β_j represents the difference in the population mean of group j compared to the reference group (if j is the reference group, the $\beta_j = 0$). Traditionally this model is called simple one-way analysis of variance, but we view it simply as another special case of a linear model.

The following example illustrates some data simulation, visualization, and parameter estimation in this context. Specifically, we assess 60 humans for their taste response to three flavors of iced cream. We want to extrapolate from our sample to the broader population of all ice cream eating humans to learn whether in general people think ice cream tastiness varies as a function of flavor.

```
# simulate and visualize data
n <- 60
x <- rep(c("chocolate", "strawberry", "vanilla"), length.out = n)
x <- factor(x)
sigma <- 1
mu_y <- c(chocolate = 3.352, strawberry = .93, vanilla = 1.5)
y <- rnorm(n, mu_y[x], sigma)

library(ggplot2)
ggplot(data.frame(x, y), aes(x, y)) +
  geom_jitter(position = position_jitter(width=.1)) +
  xlab('Group') +
  ylab('Tastiness')
```

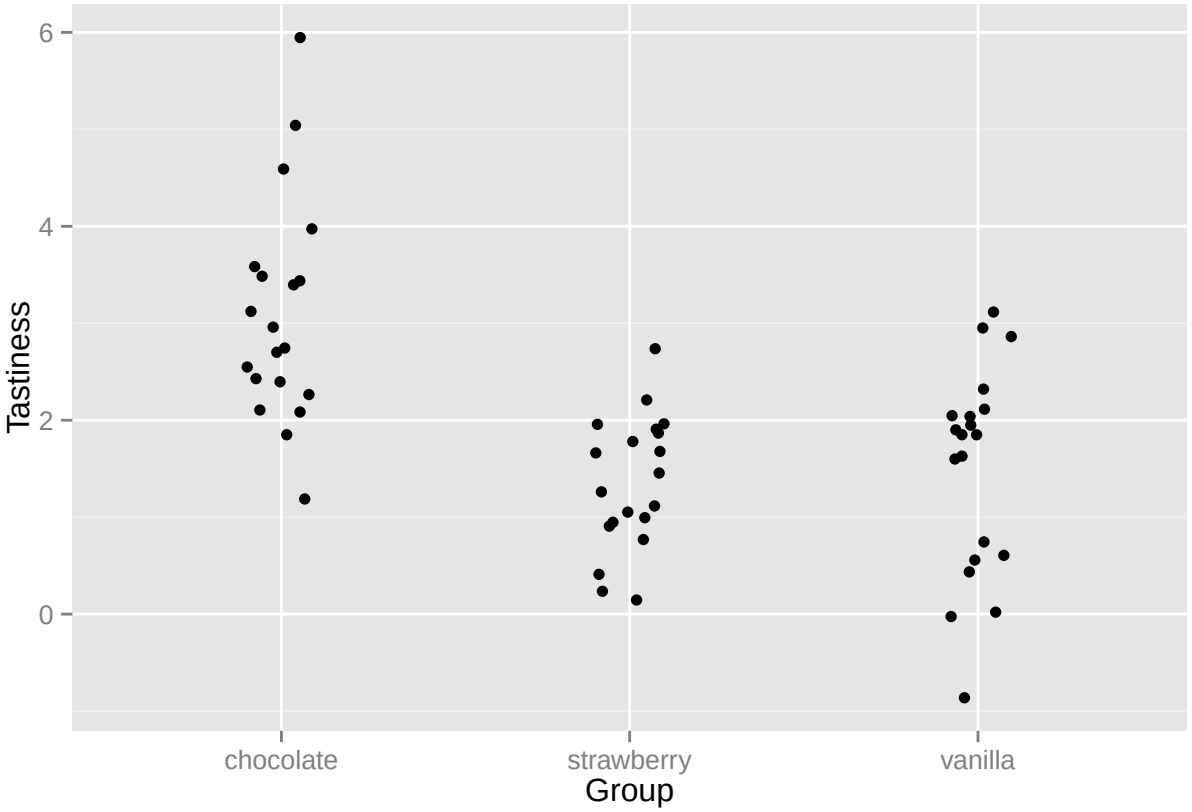

Figure 9: Simulated data on tastiness across three groups. Each point is an observation

Model fitting

We can estimate our parameters with the `lm` function (this should be a strong hint that there are not huge differences between linear regression and ANOVA). The syntax is exactly the same as with linear regression. The only difference is that our input `x` is not numeric, it's a character vector.

```
m <- lm(y ~ x)
summary(m)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3479 -0.7073  0.1083  0.5537  2.8540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.092      0.222  13.928  < 2e-16 ***
## xstrawberry   -1.740      0.314  -5.540 8.04e-07 ***
## xvanilla      -1.607      0.314  -5.118 3.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9929 on 57 degrees of freedom
## Multiple R-squared:  0.4003, Adjusted R-squared:  0.3792
## F-statistic: 19.02 on 2 and 57 DF,  p-value: 4.693e-07
```

Because chocolate comes first alphabetically, it is the reference group and the "(Intercept)" estimate corresponds to the estimate of the group-level mean for chocolate. The other two estimates are contrasts between the other groups and this reference group, i.e. "xstrawberry" is the estimated difference between the group mean for strawberry and the reference group.

If we wish instead to use a means paramaterization, we need to supress the intercept term in our model as follows:

```
m <- lm(y ~ 0 + x)
summary(m)
```

```
##
## Call:
## lm(formula = y ~ 0 + x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3479 -0.7073  0.1083  0.5537  2.8540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## xchocolate     3.092      0.222  13.928  < 2e-16 ***
## xstrawberry    1.353      0.222   6.093 1.01e-07 ***
## xvanilla       1.485      0.222   6.690 1.05e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9929 on 57 degrees of freedom
## Multiple R-squared:  0.8288, Adjusted R-squared:  0.8197
## F-statistic: 91.95 on 3 and 57 DF,  p-value: < 2.2e-16
```

Arguably, this approach is more useful because it simplifies the construction of confidence intervals for the group means:

```
confint(m)
```

```
##                 2.5 %   97.5 %
## xchocolate  2.6477338 3.536922
## xstrawberry 0.9082129 1.797401
## xvanilla    1.0406475 1.929835
```

Checking assumptions

Our assumptions in this simple one way ANOVA context are identical to our assumptions with linear regression. Specifically, we assumed that our errors are independently and identically distributed, and that the variance is constant for each group (homoskedasticity). The built in `plot` method for `lm` objects is designed to return diagnostic plots that help to check these assumptions.

```
par(mfrow=c(2, 2))
plot(m)
```

General linear models

We have covered a few special cases of general linear models, which are usually written as follows:

$$y \overset{iid}{\sim} N(X\beta, \sigma^2)$$

Where y is a vector consisting of n observations, X is a "design" matrix with n rows and p columns, and β is a vector of p parameters. There are multivariate general linear models (e.g., MANOVA) where the response variable is a matrix and a covariance matrix is used in place of a scalar variance parameter, but we'll stick to univariate models for simplicity. The key point here is that the producct of X and β provides the mean of the normal distribution from which y is drawn. From this perspective, the difference between the model of the mean, linear regression, ANOVA, etc., lies in the structure of X and subsequent interpretation of the parameters β. This is a very powerful idea that unites many superficially disparate approaches. It also is the reason that these models are considered "linear", even though a regression line might by quite non-linear (e.g., polynomial regression). These models are linear in their parameters, meaning that our expected value for the response y is a **linear combination** (formal notion) of the parameters. If a vector of expected values for y in some model cannot be represented as $X\beta$, then it is not a linear model.

In the model of the mean, X is an n by 1 matrix, with each element equal to 1 (i.e. a vector of ones). With linear regression, X's first column is all ones (corresponding to the intercept parameter), and the second column contains the values of the covariate x. In ANOVA, the design matrix X will differ between the means and effects parameterizations. With a means parameterization, the entries in column j will equal one if observation (row) i is in group j, and entries are zero otherwise. If you are not comfortable with matrix multiplication, it's worth investing some effort so that you can understand why $X\beta$ is such a powerful construct.

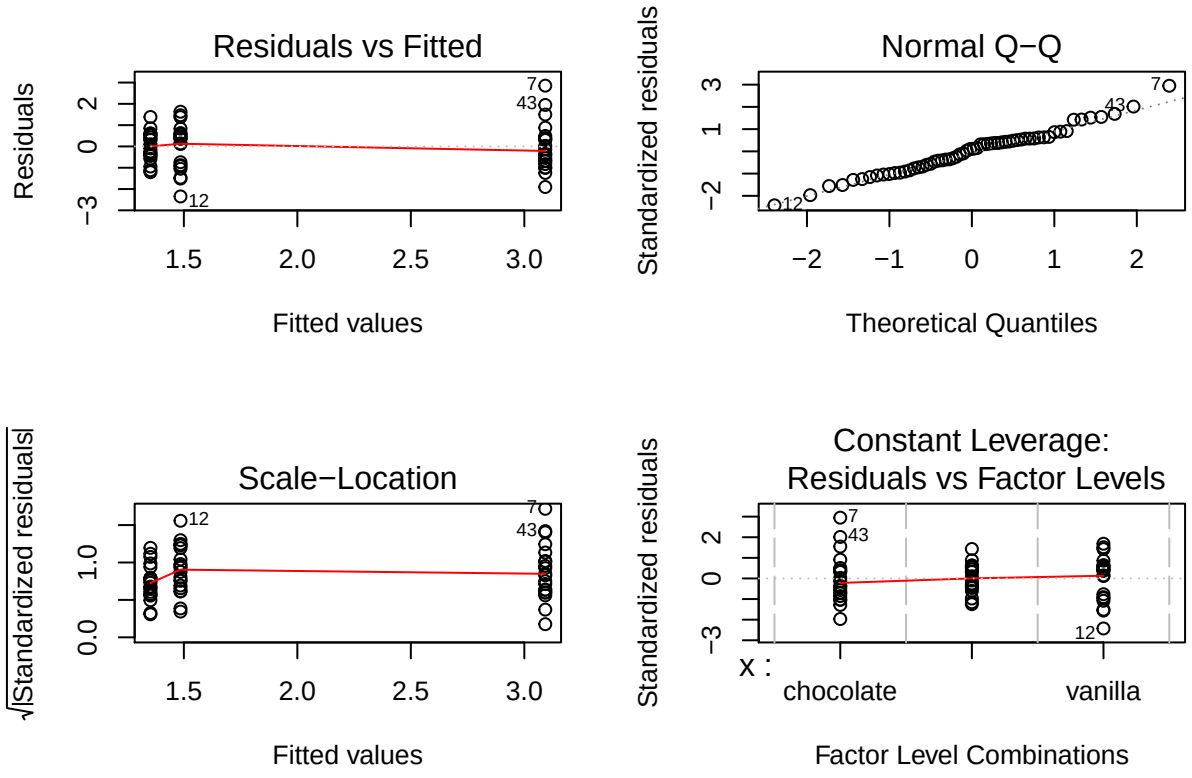

Figure 10: Default diagnostic plots for `lm` objects.

Can you figure out the structure of X with R's default effects parameterization? You can check your work with `model.matrix(m)`, where `m` is a model that you've fitted with `lm`.

Interactions between covariates

Often, the effect of one covariate depends on the value of another covariate. This is referred to as "interaction" between the covariates. Interactions can exist between two or more continuous and/or nominal covariates. These situations have special names in the classical statistics literature. For example, models with interactions between nominal covariates fall under "factorial ANOVA", those with interactions between a continuous and a nominal covariate are referred to as "analysis of covariance (ANCOVA)". Here we prefer to consider these all as special cases of general linear models.

Interactions between two continuous covariates

Here we demonstrate simulation and estimation for a model with an interaction between two continuous covariates. Notice that in the simulation, we have exploited the $X\beta$ construct to generate a vector of expected values for y.

```
n <- 50
x1 <- rnorm(n)
x2 <- rnorm(n)
beta <- c(.5, 1, -1, 2)
sigma <- 1
X <- matrix(c(rep(1, n), x1, x2, x1 * x2), nrow=n)
mu_y <- X %*% beta
```

```
y <- rnorm(n, mu_y, sigma)
m <- lm(y ~ x1 * x2)
summary(m)
```

```
##
## Call:
## lm(formula = y ~ x1 * x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20086 -0.68151 -0.03498  0.46790  2.52323
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5732     0.1434   3.998 0.000229 ***
## x1            0.9957     0.1335   7.458 1.88e-09 ***
## x2           -0.9086     0.1751  -5.188 4.66e-06 ***
## x1:x2         2.2055     0.1672  13.191  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9663 on 46 degrees of freedom
## Multiple R-squared:  0.8261, Adjusted R-squared:  0.8148
## F-statistic: 72.86 on 3 and 46 DF,  p-value: < 2.2e-16
```

Visualizing these models is tricky, because we are in 3d space (with dimensions x_1, x_2, and y), but contour plots can be effective and leverage peoples' understanding of topographic maps.

```
# visualizing the results in terms of the linear predictor
lo <- 40
x1seq <- seq(min(x1), max(x1), length.out = lo)
x2seq <- seq(min(x2), max(x2), length.out = lo)
g <- expand.grid(x1=x1seq, x2=x2seq)
g$e_y <- beta[1] + beta[2] * g$x1 + beta[3] * g$x2 + beta[4] * g$x1 * g$x2
ggplot(g, aes(x=x1, y=x2)) +
  geom_tile(aes(fill=e_y)) +
  stat_contour(aes(z=e_y), col='grey') +
  scale_fill_gradient2() +
  geom_point(data=data.frame(x1, x2))
```

Alternatively, you might check out the `effects` package:

```
library(effects)
plot(allEffects(m))
```

Interactions between two categorical covariates

Here we demonstrate interaction between two categorical covariates, using the `diamonds` dataset which is in the `ggplot2` package.t We are interested in the relationship between diamond price, cut quality, and color.

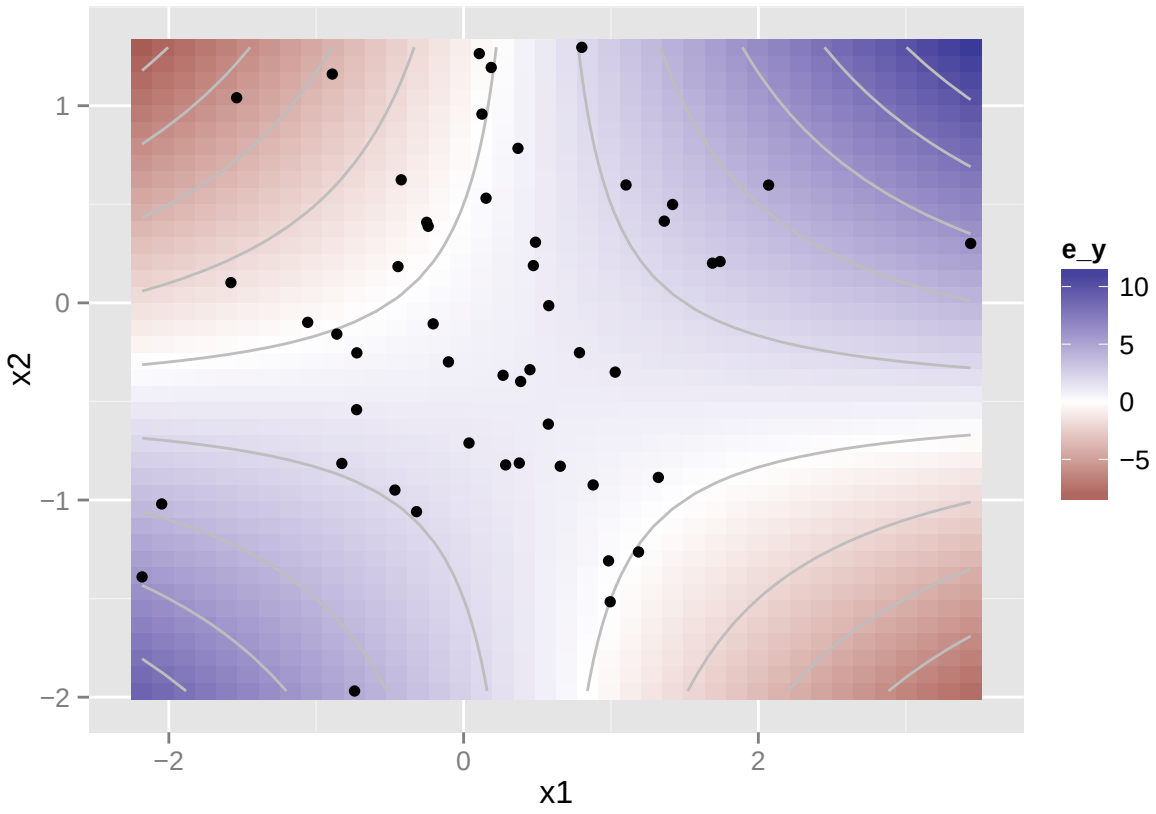

Figure 11:

```
str(ToothGrowth)
```

```
## 'data.frame':    60 obs. of  3 variables:
##  $ len : num  4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
##  $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
##  $ dose: num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

```
ToothGrowth$dose <- factor(ToothGrowth$dose)
ggplot(ToothGrowth, aes(x=interaction(dose, supp), y=len)) +
  geom_point()
```

In general, visualizing the raw data is a good idea. However, we might also be interested in a table with group-wise summaries, such as the sample means, standard deviations, and sample sizes.

```
library(dplyr)
ToothGrowth %>%
  group_by(dose, supp) %>%
  summarize(mean = mean(len),
            sd = sd(len),
            n = n())
```

```
## Source: local data frame [6 x 5]
## Groups: dose [?]
```

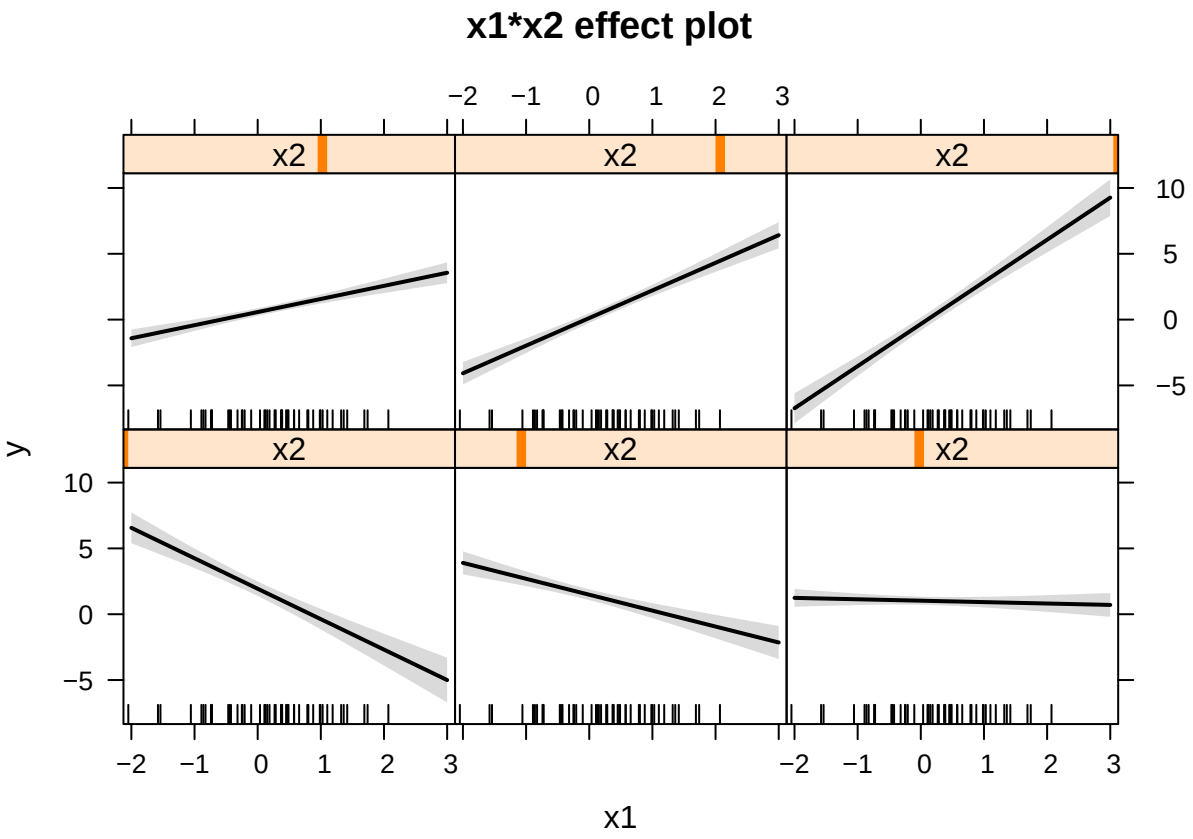

Figure 12:

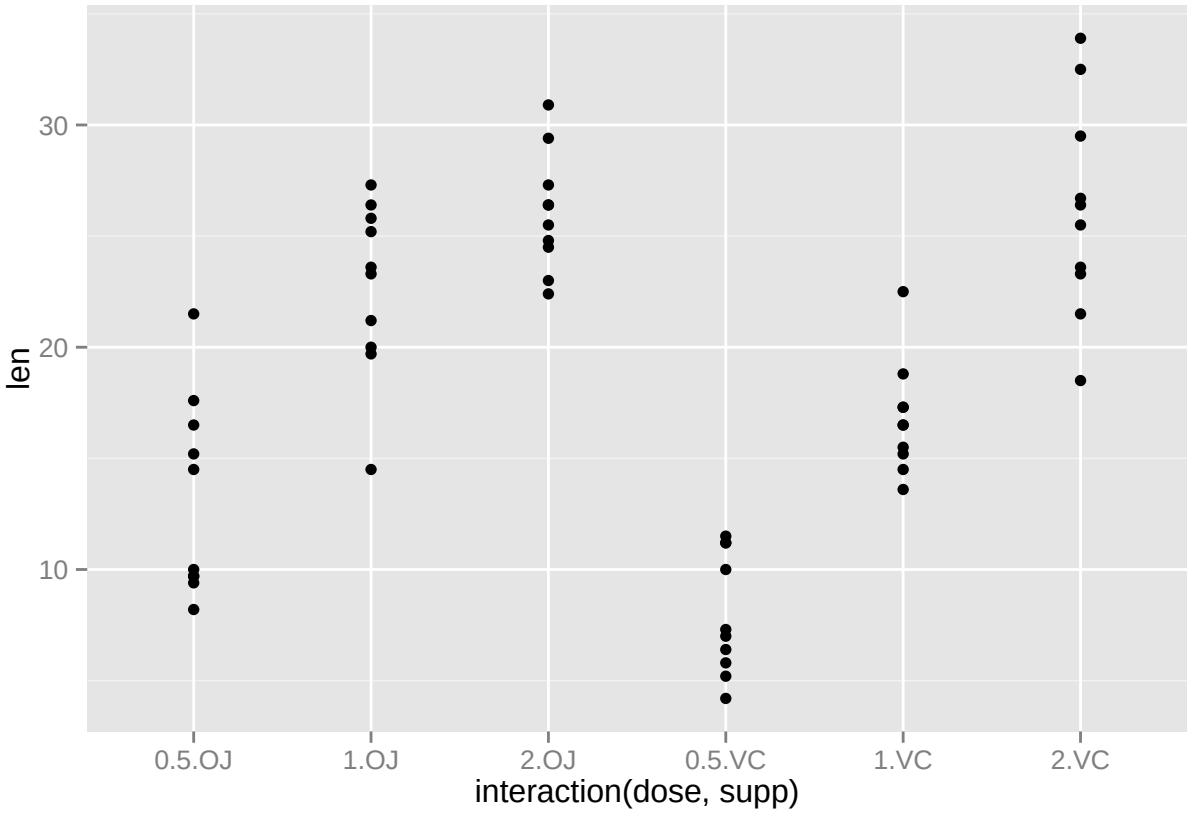

Figure 13:

```
##
##     dose   supp  mean       sd     n
##   (fctr) (fctr) (dbl)    (dbl) (int)
## 1    0.5     OJ 13.23 4.459709    10
## 2    0.5     VC  7.98 2.746634    10
## 3      1     OJ 22.70 3.910953    10
## 4      1     VC 16.77 2.515309    10
## 5      2     OJ 26.06 2.655058    10
## 6      2     VC 26.14 4.797731    10
```

We can construct a model to estimate the effect of dose, supplement, and their interaction.

```
m <- lm(len ~ dose * supp, data = ToothGrowth)
summary(m)
```

```
##
## Call:
## lm(formula = len ~ dose * supp, data = ToothGrowth)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.20  -2.72  -0.27   2.65   8.27
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.230      1.148  11.521 3.60e-16 ***
## dose1           9.470      1.624   5.831 3.18e-07 ***
## dose2          12.830      1.624   7.900 1.43e-10 ***
## suppVC         -5.250      1.624  -3.233  0.00209 **
## dose1:suppVC   -0.680      2.297  -0.296  0.76831
## dose2:suppVC    5.330      2.297   2.321  0.02411 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.631 on 54 degrees of freedom
## Multiple R-squared:  0.7937, Adjusted R-squared:  0.7746
## F-statistic: 41.56 on 5 and 54 DF,  p-value: < 2.2e-16
```

This summary gives the effects-parameterization version of the summary. The "(Intercept)" refers to the combination of factor levels that occur first alphabetically: in this case, a dose of 0.5 with the "OJ" supplement. The coefficients for `dose1` and `dose2` represent estimated contrasts for these two groups relative to the intercept. The coefficient for `suppVC` represents the contrast between the "VC" and "OJ" levels of supplement when the dose is 0.5. The interaction terms represent the difference in the effect of VC for `dose1` and `dose2` relative to a dose of 0.5. None of this is particularly intuitive, but this information can be gleaned by inspecting the design matrix X produced by `lm` in the process of fitting the model. Inspecting the design matrix along with the dataset to gives a better sense for how X relates to the factor levels:

```
cbind(model.matrix(m), ToothGrowth)
```

```
##    (Intercept) dose1 dose2 suppVC dose1:suppVC dose2:suppVC  len supp dose
## 1            1     0     0      1            0            0  4.2   VC  0.5
## 2            1     0     0      1            0            0 11.5   VC  0.5
```

```
## 3          1     0     0     1          0          0  7.3   VC  0.5
## 4          1     0     0     1          0          0  5.8   VC  0.5
## 5          1     0     0     1          0          0  6.4   VC  0.5
## 6          1     0     0     1          0          0 10.0   VC  0.5
## 7          1     0     0     1          0          0 11.2   VC  0.5
## 8          1     0     0     1          0          0 11.2   VC  0.5
## 9          1     0     0     1          0          0  5.2   VC  0.5
## 10         1     0     0     1          0          0  7.0   VC  0.5
## 11         1     1     0     1          1          0 16.5   VC    1
## 12         1     1     0     1          1          0 16.5   VC    1
## 13         1     1     0     1          1          0 15.2   VC    1
## 14         1     1     0     1          1          0 17.3   VC    1
## 15         1     1     0     1          1          0 22.5   VC    1
## 16         1     1     0     1          1          0 17.3   VC    1
## 17         1     1     0     1          1          0 13.6   VC    1
## 18         1     1     0     1          1          0 14.5   VC    1
## 19         1     1     0     1          1          0 18.8   VC    1
## 20         1     1     0     1          1          0 15.5   VC    1
## 21         1     0     1     1          0          1 23.6   VC    2
## 22         1     0     1     1          0          1 18.5   VC    2
## 23         1     0     1     1          0          1 33.9   VC    2
## 24         1     0     1     1          0          1 25.5   VC    2
## 25         1     0     1     1          0          1 26.4   VC    2
## 26         1     0     1     1          0          1 32.5   VC    2
## 27         1     0     1     1          0          1 26.7   VC    2
## 28         1     0     1     1          0          1 21.5   VC    2
## 29         1     0     1     1          0          1 23.3   VC    2
## 30         1     0     1     1          0          1 29.5   VC    2
## 31         1     0     0     0          0          0 15.2   OJ  0.5
## 32         1     0     0     0          0          0 21.5   OJ  0.5
## 33         1     0     0     0          0          0 17.6   OJ  0.5
## 34         1     0     0     0          0          0  9.7   OJ  0.5
## 35         1     0     0     0          0          0 14.5   OJ  0.5
## 36         1     0     0     0          0          0 10.0   OJ  0.5
## 37         1     0     0     0          0          0  8.2   OJ  0.5
## 38         1     0     0     0          0          0  9.4   OJ  0.5
## 39         1     0     0     0          0          0 16.5   OJ  0.5
## 40         1     0     0     0          0          0  9.7   OJ  0.5
## 41         1     1     0     0          0          0 19.7   OJ    1
## 42         1     1     0     0          0          0 23.3   OJ    1
## 43         1     1     0     0          0          0 23.6   OJ    1
## 44         1     1     0     0          0          0 26.4   OJ    1
## 45         1     1     0     0          0          0 20.0   OJ    1
## 46         1     1     0     0          0          0 25.2   OJ    1
## 47         1     1     0     0          0          0 25.8   OJ    1
## 48         1     1     0     0          0          0 21.2   OJ    1
## 49         1     1     0     0          0          0 14.5   OJ    1
## 50         1     1     0     0          0          0 27.3   OJ    1
## 51         1     0     1     0          0          0 25.5   OJ    2
## 52         1     0     1     0          0          0 26.4   OJ    2
## 53         1     0     1     0          0          0 22.4   OJ    2
## 54         1     0     1     0          0          0 24.5   OJ    2
## 55         1     0     1     0          0          0 24.8   OJ    2
## 56         1     0     1     0          0          0 30.9   OJ    2
```

```
## 57           1     0     1      0            0            0 26.4   OJ    2
## 58           1     0     1      0            0            0 27.3   OJ    2
## 59           1     0     1      0            0            0 29.4   OJ    2
## 60           1     0     1      0            0            0 23.0   OJ    2
```

Often, researchers want to know if interactions need to be included in the model. From a null hypothesis significance testing perspective, we can evaluate the 'significance' of the interaction term as follows:

```
anova(m)
```

```
## Analysis of Variance Table
##
## Response: len
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## dose       2 2426.43 1213.22  92.000 < 2.2e-16 ***
## supp       1  205.35  205.35  15.572 0.0002312 ***
## dose:supp  2  108.32   54.16   4.107 0.0218603 *
## Residuals 54  712.11   13.19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We find that the interaction between dose and supplement is statistically significant, meaning that if we assume that there is no interaction, it is unlikely to observe data that are as or more extreme as what we have observed over the course of infinitely many replicated experiments that will probably never occur. Although this is far from intuitive, this approach has been widely used. We will introduce a more streamlined procedure in chapter 3 that 1) does not assume that the effect is zero to begin with, and 2) does not necessarily invoke a hypothetical infinite number of replicated realizations of the data, conditional on one particular parameter value. An alternative approach would be to use information theoretics to decide whether the interaction is warranted:

```
m2 <- lm(len ~ dose + supp, data = ToothGrowth)
AIC(m, m2)
```

```
##    df      AIC
## m   7 332.7056
## m2  5 337.2013
```

In the past decade following Burnham and Anderson's book on the topic, ecologists have leaned heavily on Akaike's information criterion (AIC), which is a relative measure of model quality (balancing goodness of fit with model complexity). Here we see that the original model `m` with interaction has a lower AIC value, and is therefore better supported. AIC can be considered to be similar to cross validation, approximating the ability of a model to predict future data.

Being somewhat lazy, we might again choose to plot the results of this model using the `effects` package.

```
plot(allEffects(m))
```

This is less than satisfying, as it does not show any data. All we see is model output. If the model is crap, then the output and these plots are also crap. But, evaluating the crappiness of the model is difficult when there are no data shown. Ideally, the data can be shown along with the estimated group means and some indication of uncertainty. If we weren't quite so lazy, we could use the `predict` function to obtain confidence intervals for the means of each group.

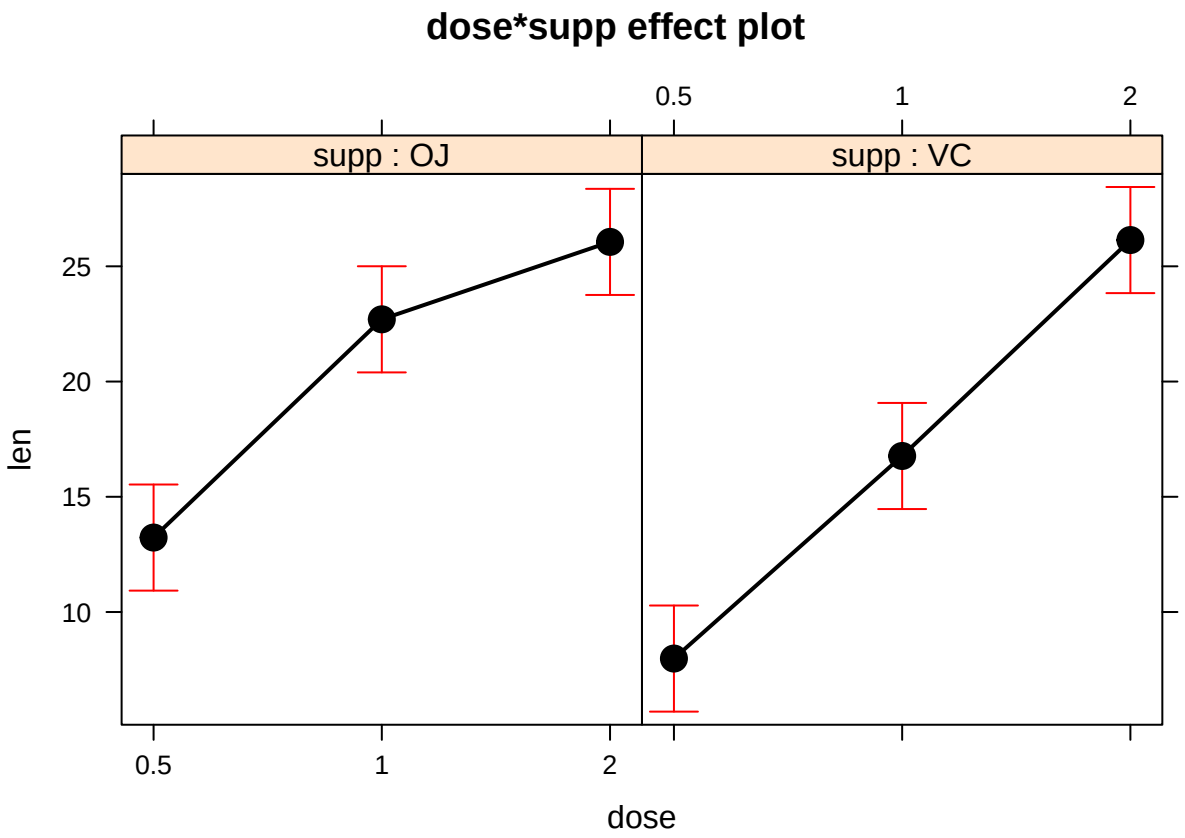

Figure 14:

```
# construct a new data frame for predictions
g <- expand.grid(supp = levels(ToothGrowth$supp),
                 dose = levels(ToothGrowth$dose))
p <- predict(m, g, interval = 'confidence', type='response')
predictions <- cbind(g, data.frame(p))

ggplot(ToothGrowth, aes(x=interaction(dose, supp), y=len)) +
  geom_segment(data=predictions,
               aes(y=lwr, yend=upr,
                   xend=interaction(dose, supp)), col='red') +
  geom_point(data=predictions, aes(y=fit), color='red', size=2, shape=2) +
  geom_jitter(position = position_jitter(width=.1), shape=1) +
  ylab("Length")
```

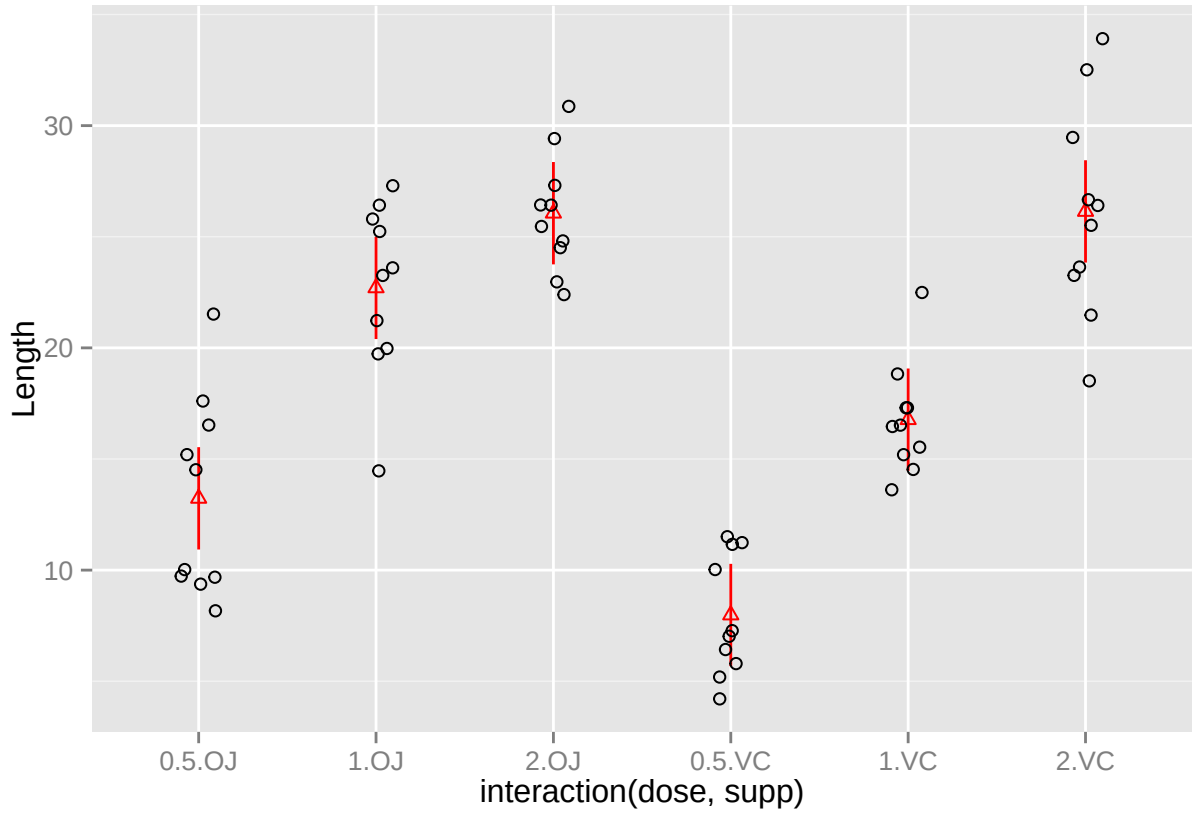

Figure 15:

This plot is nice because we can observe the data along with the model output. This makes it easier for readers to understand how the model relates to, fits, and does not fit the data. If you wish to obscure the data, you could make a bar plot with error pars to represent the standard errors. Although "dynamite" plots are common, we shall not include one here and we strongly recommend that you never produce such a plot (more here).

Interactions between continuous and categorical covariates

Sometimes, we're interested in interactions between continuous or numeric covariates and another covariates with discrete categorical levels. Again, this falls under the broad class of models used in analysis of covariance

(ANCOVA).

```
x1 <- rnorm(n)
x2 <- factor(sample(c('A', 'B'), n, replace=TRUE))

# generate slopes and intercepts for the first and second groups
a <- rnorm(2)
b <- rnorm(2)
sigma <- .4

X <- matrix(c(ifelse(x2 == 'A', 1, 0),
              ifelse(x2 == 'B', 1, 0),
              ifelse(x2 == 'A', x1, 0),
              ifelse(x2 == 'B', x1, 0)
            ), nrow=n)

mu_y <- X %*% c(a, b)
y <- rnorm(n, mu_y, sigma)
plot(x1, y, col=x2, pch=19)
legend('topright', col=1:2, legend=c('Group A', 'Group B'), pch=19)
```

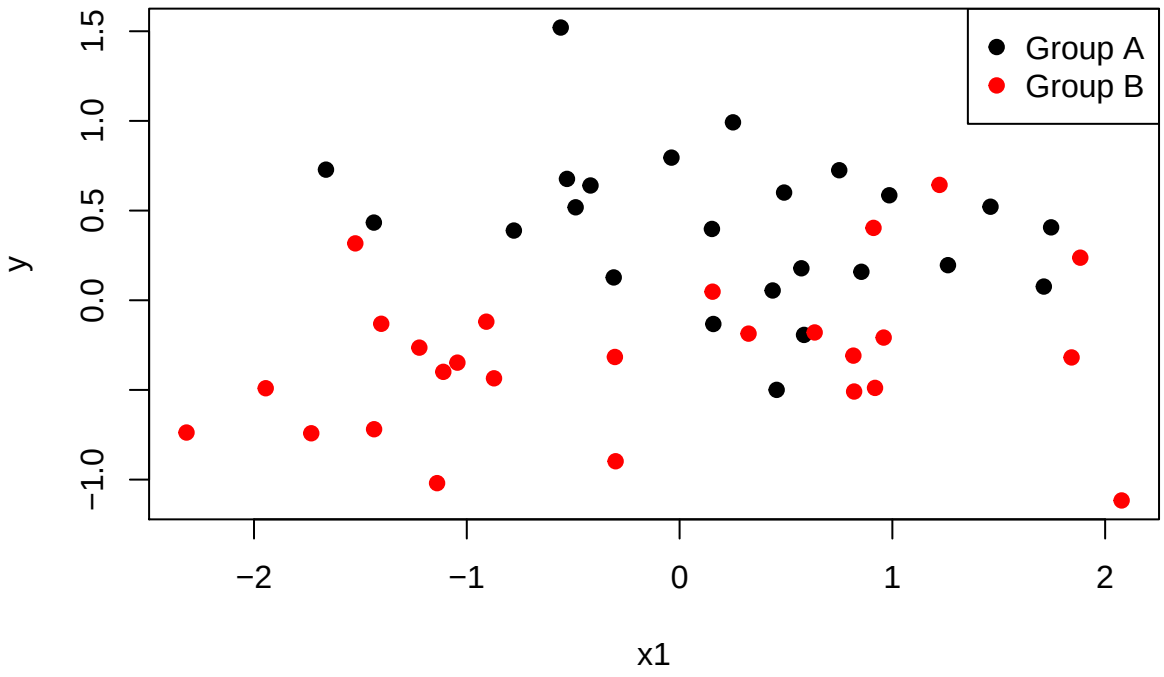

Figure 16:

Here the intercepts and slopes are allowed to vary for two groups. We can fit a model with an interaction between these covariates. The intercepts and slopes are estimated separately for the two groups.

```
m <- lm(y ~ x1 * x2)
summary(m)
```

```
##
## Call:
```

```
## lm(formula = y ~ x1 * x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98899 -0.23586 -0.00216  0.25239  0.98912
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.44763    0.08757   5.112 6.03e-06 ***
## x1          -0.15007    0.09610  -1.562   0.1252
## x2B         -0.75103    0.12003  -6.257 1.19e-07 ***
## x1:x2B       0.23486    0.11553   2.033   0.0479 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4144 on 46 degrees of freedom
## Multiple R-squared:  0.4833, Adjusted R-squared:  0.4496
## F-statistic: 14.34 on 3 and 46 DF,  p-value: 9.935e-07
```

Let's plot the lines of best fit along with the data.

```
plot(x1, y, col=x2, pch=19)
legend('topright', col=1:2, legend=c('Group A', 'Group B'), pch=19)
abline(coef(m)[1], coef(m)[2])
abline(coef(m)[1] + coef(m)[3], coef(m)[2] + coef(m)[4], col='red')
```

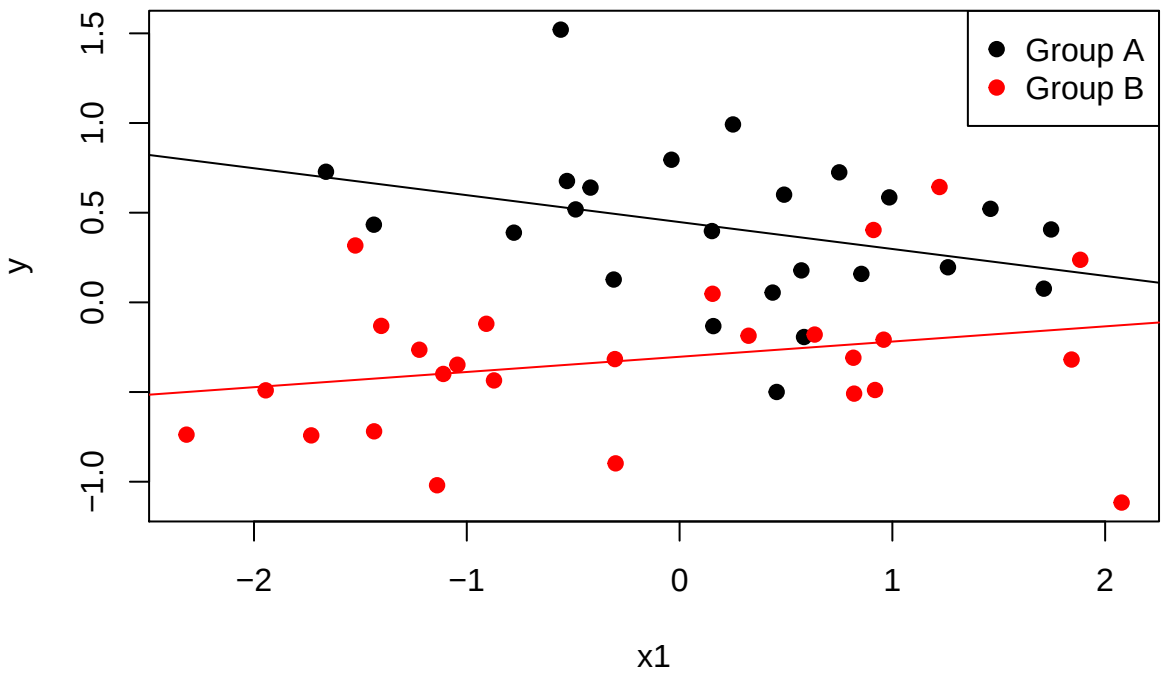

Figure 17:

The `abline` function, used above, adds lines to plots based on a y-intercept (first argument) and a slope (second argument). Do you understand why the particular coefficients that we used as inputs provide the desired intercepts and slopes for each group?

Further reading

Schielzeth, H. 2010. Simple means to improve the interpretability of regression coefficients. Methods in Ecology and Evolution 1:103–113.

Enqvist, L. 2005. The mistreatment of covariate interaction terms in linear model analyses of behavioural and evolutionary ecology studies. Animal Behaviour 70:967–971.

Gelman and Hill. 2009. *Data analysis using regression and multilevel/hierarchical models*. Chapter 3-4.