

# 1 Some Distributions

## Binomial and Negative Binomial Distributions

The binomial and negative binomial distributions arise in very similar situations. In each case

- The random process consists of sub-processes called **trials**.
- Each trial has **two possible outcomes** (generically called *success* and *failure*).
- The **probability of success** for each trial is the same ( $\pi$ ).
- Each trial is **independent** of all the others.

**Binomial:**  $X \sim \text{Binom}(n, \pi)$

- The total number of trials ( $n$ ) is known in advance
- $X$  = the number of success
- pmf:  $f(x) = \binom{n}{x} \pi^x (1 - \pi)^{n-x} = \text{dbinom}(x, n, \text{prob})$

**Negative Binomial:**  $Y \sim \text{NBinom}(s, \pi)$

- The number of successes ( $s$ ) is known in advance
- $Y$  = the number of failures
- pmf:  $f(x) = \binom{s+x-1}{s} \pi^s (1 - \pi)^x = \text{dnbinom}(x, \text{size}, \text{prob})$
- If  $s = 1$  the distribution is also called a **geometric distribution**

## Poisson and Exponential Distributions

The Poisson and Exponential distributions form another interesting pair. This time the situation is that we are watching for occurrences of some event such that

- occurrences are independent of each other
- the average rate of occurrences remains constant

**Poisson:**  $X \sim \text{Pois}(\lambda)$

- watch for a **fixed amount of time**
- $\lambda$  = average number of occurrences *per time watched*
- $X$  = **number of occurrences** (discrete)
- pmf:  $e^{-\lambda} \frac{\lambda^x}{x!} = \text{dpois}(x, \text{lambda})$

**Exponential:**  $Y \sim \text{Exp}(\lambda)$

- watch until there is an occurrence
- $\lambda$  = average number of occurrences *per unit time*
- $Y$  = **time until an occurrence** (continuous)
- pdf:  $\lambda e^{-\lambda x} = \text{dexp}(x, \text{rate})$

## 2 Working with Distributions in R

### cdfs (cumulative distribution functions)

Often it is convenient to work with the proportion of a distribution that is **less than or equal to** a given value  $a$ . This is computed using

- $F(x) = \int_{-\infty}^x f(t) dt$  for continuous distributions, and
- $F(x) = \sum_{y \leq x} f(y)$  for discrete distributions.

These functions are called **cumulative distribution functions** (abbreviated cdf) and are usually denoted with capital letters matching the lower case letters used for the pdf or pmf.

### Proportion notation

It will be handy to use the following notation for distributions:

- $P(X = 5)$ : The proportion of the distribution of  $X$  that equals 5.
- $P(X \leq 5)$ : The proportion of the distribution of  $X$  that is less than or equal to 5.
- etc.

### pmfs, pdfs, and cdfs in R

R includes pmf or pdf and cdf functions for many important distributions. The pdf and pmf functions all begin with the letter **d** (for density) even if the distribution is discrete. The cdfs all begin with the letter **p** (this will make more sense later). For example: `dbinom()`, `pbinom()`, `dnbinom()`, `pnbinom()`, `dpois()`, `ppois()`, `dexp()`, `pexp()`, `dunif()`, `punif()`. The arguments to these functions need not be named if they are provided in the standard order, but I've used names for all but the first argument below to make it clear what each argument represents.

### Examples

1. Suppose  $X \sim \text{Exp}(\lambda = 2)$ .

a) What proportion of the distribution is less than 1?

A.  $P(X \leq 1) =$

```
| pexp(1,rate=2)
|[1] 0.8646647
```

b) What proportion of the distribution is greater than 1?

A.  $P(X \geq 1) =$

```
| 1 - pexp(1,rate=2)
|[1] 0.1353353
```

c) What proportion of the distribution is between 1 and 4?

A.  $P(1 \leq X \leq 4) =$

```
| pexp(4,rate=2) - pexp(1,rate=2)
|[1] 0.1349998
```

2. We have to be a bit more careful when working with discrete distributions because the proportion less than or equal to a value may not be the same as the proportion less than or equal to that value.

Suppose, for example, that  $X \sim \text{Binom}(100, .5)$ .

- a) What proportion of the distribution is less than or equal to 40?  
 A.  $P(X \leq 40) =$   

```
|> pbinom(40,size=100,prob=.5)
[1] 0.02844397
```
- b) What proportion of the distribution is greater than or equal to 40?  
 A.  $P(X \geq 40) = 1 - P(X \leq 39) =$   

```
|> 1 - pbinom(39,size=100,prob=.5)
[1] 0.9823999
```
- c) What proportion is between 40 and 60?  
 A.  $P(40 \leq X \leq 60) = P(X \leq 60) - P(X \leq 39) =$   

```
|> pbinom(60,size=100,prob=.5) - pbinom(39,size=100,prob=.5)
[1] 0.9647998
```

## Functions, Sums, and Numerical Integracion in R

Most of the distributions we will use in this class have built-in functions in R. But if you need to work with a new distribution, the following can be very useful.

1. We can define functions in R. For example, if a pdf is  $f(x) = 3x^2$  on  $[0, 1]$ , we might define the function  

```
|> f <- function(x) { return(3 * x^2) }
```
2. We can do numerical integration using the `integrate()` function.

```
|> integrate(f,0,1)
1 with absolute error < 1.1e-14
```

Note that because we did not define the function correctly outside of the interval  $[0, 1]$ , we have to make our limits be 0 and 1 rather than  $-\infty$  and  $\infty$ . We can fix this if we like:

```
|> f = function(x) { (x >= 0) * (x <= 1) * 3 * x^2 }
> f(-1)
[1] 0
> f(3)
[1] 0
> integrate(f,0,1)
1 with absolute error < 1.1e-14
> integrate(f, -Inf, Inf)
1 with absolute error < 7.2e-08
```

3. For discrete distributions, sums are more interesting than integrals. Here we define our function in a fancier way so that it gives a value of 0 where it should for a pmf.

```
|> g <- function(x) { return( (x %in% 1:4) * x/10 ) }
> g(0:10)
[1] 0.0 0.1 0.2 0.3 0.4 0.0 0.0 0.0 0.0 0.0 0.0
> sum(g(0:10))
[1] 1
```

## How did that go again?

- `args(pbinom)` will display the arguments the `pbinom()` function takes. (On a Mac this displays at the bottom of the screen as soon as you type `pbinom(.)`)
- `?pbinom` will display a help page for `pbinom()`.
- `apropos('bin')` will list all function that have `bin` as part of their name.
- `example(pbinom)` will show some examples of how to use `pbinom()`. This can be a very good way to get to know a new function.

**Some Problems**

Use R as much as you can for these problems.

## 3 Measures of Center Using R

### 3.1 Working with Data

#### Loading Data from Packages

As you would expect, a statistics programming environment like R is good at working with data. The most common way to store data in R is in a format called a **data frame**. It is arranged with variables in columns and observational units in rows. There are a number of data sets that come with R by default, and many more that are available in add-on packages. To load a data set from a package is very easy, for example:

```
| data(faithful)
```

will load a data containing information about eruptions of Old Faithful geyser in Yellowstone National Park, storing the data in a data frame named `faithful`. The following code let's us see what the variables are called and look at the first few rows of the data set.

```
| > data(faithful)
| > names(faithful)
| [1] "eruptions" "waiting"
| > faithful[1:5,]
|   eruptions waiting
| 1    3.600      79
| 2    1.800      54
| 3    3.333      74
| 4    2.283      62
| 5    4.533      85
```

I have made a package called `DevFar2` that contains all the data sets from the CD that comes with the book. This package is NOT installed in the engineering labs, but you can download it to your own machine with the following command

```
| install.packages('DevFar2', repos='http://www.calvin.edu/~rpruim/R', type='source')
```

Data sets are named with an 'e' for examples or an 'x' for exercises. So, to get the data for Exercise 2.1.2, you can do the following:

```
| > require(DevFar2)
| > data(x2.1.2)
| > x2.1.2
|   C1
| 1 15.0
| 2 13.0
| 3 18.0
| 4 14.5
| 5 12.0
| 6 11.0
| 7  8.9
| 8  8.0
```

On Macs and PCs there are also GUI menu items to help with installing and loading packages.

#### Using Your Own Data

Of course, you can also read your own data from properly configured files. The simplest format is csv – comma separated values – which Excel is happy to produce for you if you have your data in an Excel spreadsheet. Just be sure that you have your spreadsheet arranged with rows corresponding to observational units and columns corresponding to variables **with no extra junk in the spreadsheet**. The top row of your spreadsheet should contain the names of the variables. (It is possible to work with unnamed columns, but by default R expects

the first row of a csv file to contain variable names, and it is a good idea to name your variables anyway.)

Here's how you read your data:

```
mydata <- read.csv("/some/path/to/myDataFile.csv"); # Mac/Linux style path
```

That's all there is to it.

## Mean and Median

It is easy to get R to calculate medians and means (including trimmed means):

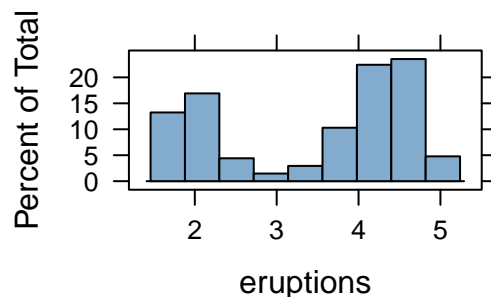
```
> median(faithful$eruptions)
[1] 4
> mean(faithful$eruptions)
[1] 3.487783
> mean(faithful$eruptions,trim=.10) # 10% trimmed mean
[1] 3.529807
```

The \$ operator is used to access just a single variable from a data frame.

## Histograms

A good rule of thumb is never to put much stock in numerical summaries until you have looked at a plot. Let's make a histogram:

```
> require(lattice)
> histogram(~eruptions, data=faithful)
```



From this we see that the mean and median are not good measures of a “typical eruption time”. The distribution is bimodal and the mean and median fall in the valley between the two mountains.

## 3.2 Discrete Distributions

We can also use R to investigate distributions. For example, consider the following discrete distribution:

value of $X$	1	2	3	4
probability	.1	.2	.3	.4

We can compute the mean using the definition:

$$\mu_X = E(X) = \sum_x x P(X = x).$$

```
> vals = 1:4           # short hand for c(1,2,3,4)
> probs = c(.1,.2,.3,.4) # c() is used to paste numbers into a vector
> vals * probs        # componentwise products
[1] 0.1 0.4 0.9 1.6
> sum(vals * probs)   # sum of component wise products is the mean
[1] 3
```

### 3.3 Continuous Distributions

The mean of a continuous distribution is just as easy. This time, we first have to define a pdf function. Consider  $k(x) = x^2$  on  $[0, 3]$ . It's the kernel of a pdf. Let's find the pdf and compute the mean of the distribution using the definition

$$\mu_X = E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

```
> k <- function(x) { x^2 }
> k(0:4)                # quick sanity check
[1] 0 1 4 9 16
> integrate(k,0,3)
9 with absolute error < 1e-13
> integrate(k,0,3)$value # get value as a number (w/out tolerance)
[1] 9
> f <- function(x) { x^2 / integrate(k,0,3)$value }
> xf <- function(x) {x * f(x)}
> integrate(xf,0,3)
2.25 with absolute error < 2.5e-14
```

### 3.4 A GUI for R?

R was designed primarily to be a programming language. For sophisticated statistical analysis this has many benefits, including making it possible to write your own functions to automate common tasks and providing a record of exactly how an analysis was done. R commands can be stored in files and re-run at a later time. Particularly useful commands can be placed into a package.

The downside is that it takes some learning to get going. If you have done some programming before (especially if it was in C, C++, java, perl, or some language with a similar syntax), R is quite easy to learn. Even if you have not programmed before, it is really not too hard to learn commands as you have need of them.

Nevertheless for beginners it is sometimes nice to have drop down menus to guide you. The `Rcmdr` package provides just such menus for the most frequently encountered topics in an introductory course. As an added bonus, it displays the actual commands being executed, so you can learn them as you go along. I'll be showing the actual R commands in class and in handouts since it is much easier to record than describing menus and mouse clicks. But if you want to give `Rcmdr` a try, just type

```
|require(Rcmdr)
```

One thing that is mostly missing from `Rcmdr` is `lattice` graphics. This is unfortunate because compared to the basic graphics that come with R `lattice` graphics generally look better, provide more control over layout, and have some extra features that are very nice for exploring a data set graphically. I've written a plug-in package to add some `lattice` functionality to `Rcmdr`. You can get the extra menu items using

```
|install.packages('RcmdrPlugin.Calvin', repos='http://www.calvin.edu/~rpruim/R', type='source')
|require(RcmdrPlugin.Calvin) # this can also be selected from within Rcmdr
```

You still won't find everything R can do in `Rcmdr`, but many useful things are there.

## Exercises

**1** A manufacturer orders parts from a supplier. Before accepting a large shipment of parts, the manufacturer tests 10 parts selected at random from the shipment. If any of the parts is found to be of substandard quality, the shipment is rejected. If all 10 sampled parts are of good quality, the entire shipment is accepted. (This is sometimes called **acceptance sampling**.)

- a) If 1% of a the shipment are substandard, how likely is the manufacturer to accept the shipment?
- b) Repeat the previous question for different percentages ( 5%, 10%, 15%, 20%, and 25%) of substandard parts.
- c) Now repeat the analysis assuming that the manufacturer is more lax and allows at most one substandard part in the sample.
- d) Repeat the analysis once more assuming that the manufacturer will accept shipments if there are at most 2 substandard parts in the sample.
- e) Put your results onto a graph. On the horizontal axis put the percentage of substandard parts in the shipment. On the vertical axis, put the proportion of such shipments that are accepted. Use different colors or symbols to distinguish the three acceptance protocols (at most 0, 1, or 2 substandard parts).

**2** A manufacturing plant keeps track of how many days they have gone without an injury to an employee and posts this on a sign near the employee entrance. Over the last several years, the company has averaged 10 injuries per year. The plant runs 24-hours a day, 365 days per year. (Ignore Leap Days to keep this simple.) As you walk into the plant, you notice that the sign says “10 days without an injuring, and counting...”.

- a) How likely is the plant to make it to 30 days without an injury?
- b) How likely is the plant to make it to 50 days without an injury?
- c) How likely is the plant to make it to 100 days without an injury?

**3** During an August meteor shower, meteors come at an average rate of 20 per hour.

- a) What is the probability of seeing exactly 20 meteors if you watch for one hour?
- b) What is the probability of seeing between 15 and 20 meteors?
- c) How likely are you to wait longer than 5 minutes until you see a meteor?

**4** On average, 10 phone calls arrive at a customer service call center per hour. Fran works a 4-hour shift answering the initial calls and transferring them to the appropriate customer service agent. (Her company has not yet invested in one of those “For this kind of help, press 1; for this other type of help, press 2 ...” messaging systems.)

- a) How likely is it that she receives 30 or more calls during her shift?
- b) How likely is it that she receives 50 or more calls during her shift?
- c) How likely is Fran to go 10 minutes without a call?

**5** Are there any reasons to be concerned about the models used in the problems above? Each model makes certain assumptions about the situation. Identify any places where you think these assumptions are (or may

be) unreasonable.

**6** The kernel of a continuous distribution on  $[0, 2]$  is  $x^3$ .

**a)** Determine the pdf.

**b)** What  $P(X \leq 1)$ ?

## Some Solutions

1 Here is a table of probabilities of accepting the shipment. The R code to generate the values (and more accurate results) appears below.

	pfail = 0.01	pfail = 0.05	pfail = 0.1	pfail = 0.15	pfail = 0.2	pfail = 0.25
max bad = 0:	0.90	0.60	0.35	0.20	0.11	0.06
max bad = 1:	1.00	0.91	0.74	0.54	0.38	0.24
max bad = 2:	1.00	0.99	0.93	0.82	0.68	0.53

```
> pfail <- c(.01,.05,.10,.15,.20,.25);
> results <- outer(0:2,pfail, function(x,p) {pbinom(x,10,p)});
> row.names(results) <- paste("max bad = ",0:2,",",sep="");
> colnames(results) <- paste('pfail = ',pfail,sep="");
> print(results);
      pfail = 0.01 pfail = 0.05 pfail = 0.1 pfail = 0.15 pfail = 0.2
max bad = 0:    0.9043821    0.5987369    0.3486784    0.1968744    0.1073742
max bad = 1:    0.9957338    0.9138616    0.7360989    0.5442998    0.3758096
max bad = 2:    0.9998862    0.9884964    0.9298092    0.8201965    0.6777995
      pfail = 0.25
max bad = 0:    0.05631351
max bad = 1:    0.24402523
max bad = 2:    0.52559280
```

2 Here are two possible solutions. The first is a somewhat better model because it allows (with small probability) for multiple accidents on the same day. But since this probability is small, the two models give very similar results.

```
# modeling using exponential with a rate of 10/365 per day
> 1 - pexp(c(20,40,90), 10/365);
[1] 0.57813654 0.33424186 0.08494482
# modeling as a binomial with a probability of 10/365 of an accident any day
> dbinom(0,c(20,40,90),10/365);
[1] 0.5737329 0.3291695 0.0820718
```

### 3

```
> dpois(20,20);          # P(X=20)          X ~ Pois(20)
[1] 0.08883532
> ppois(20,20) - ppois(14,20); # P(15 <= X <= 20)  X ~ Pois(20)
[1] 0.4542283
> 1-pexp(5,rate=20);     # P(Y <= 20)          Y ~ Exp(20)
[1] 0
```

### 4

```
> 1 - dpois(29,40)      # 1 - P(X <= 29)      X ~ Pois(40)
[1] 0.9861509
> 1 - dpois(49,40)      # 1 - P(X <= 29)      X ~ Pois(40)
[1] 0.9778662
> dpois(0,10/6)         # P(X = 0)           X ~ Pois(10/6)   [per 10 minutes]
[1] 0.1888756
> 1 - pexp(10,10/60)    # 1 - P(Y <= 10)     Y ~ Exp(10/60)   [minutes]
[1] 0.1888756
> 1 - pexp(10/60,10)    # 1 - P(Y <= 10/60)  Y ~ Exp(10)      [hours]
[1] 0.1888756
```

**5** Examples of potential problems:

- At the call center, call may come with higher frequency at certain times during the shift, so the rate might not be constant of the time watched.
- Injuries might not be independent. If there is a big accident, multiple people might be injured all at once.

**6**

```
> f <- function(x) { x^3 }
> integrate(f,0,2);
4 with absolute error < 4.4e-14
> g <- function(x) { x^3 / integrate(f,0,2)$value }
> integrate(g,0,2);
1 with absolute error < 1.1e-14
> integrate(g,0,1);
0.0625 with absolute error < 7e-16
```

## 4 Some Additional Distributions

### 4.1 Quantiles

Quantiles are basically percentiles described with an arbitrary proportion. The 80th percentile, for example, is the 0.8-quantile, and has the property that 80% of the distribution is below it. The median is the 0.5-quantile. Quantiles can be found in R using functions that begin with the letter ‘q’. For the normal distribution, the function is `qnorm()`.

### 4.2 Lognormal Distributions

$X$  has a lognormal distribution if  $Y = \ln(X)$  has a normal distribution. (So  $X = e^Y$ ). Lognormal distributions are usually described by specifying the mean and standard deviation of  $Y$  rather than of  $X$ . Although you could work with lognormal distributions by transforming to normal distributions, R provides `dlognorm()`, `plognorm()` and `qlognorm()` for working with the lognormal distributions directly.

### 4.3 Weibull and Gamma Distributions

The Weibull and Gamma distributions are generalizations of the exponential distribution. The kernels for these distributions have the following basic forms:

<u>distribution</u>	<u>kernel</u>	<u>kernel (alt)</u>	<u>range of parameters</u>
Weibull	$x^{\alpha-1}e^{-(\lambda x)^\alpha}$	$x^{\alpha-1}e^{-(x/\beta)^\alpha}$	$\alpha > 0, \beta > 0, \lambda > 0$
Gamma	$x^{\alpha-1}e^{-\lambda x}$	$x^{\alpha-1}e^{-x/\beta}$	$\alpha > 0, \beta > 0, \lambda > 0$

Some comments:

- The first of these is integrable using substitution and the second by parts, but we will primarily using the R functions `dweibull()`, `pweibull()`, `qweibull()`, `dgamma()`, `pgamma()`, and `qgamma()` when working with these distributions.
- The exponential distributions are a special case of the Weibull (when  $\alpha = 1$ ) and Gamma (also when  $\alpha = 1$ ) distributions.
- The parameters  $\alpha$  and  $\beta$  are often called the shape and scale parameters and  $\lambda = \frac{1}{\beta}$  is called the rate parameter. R allows you to select a Gamma distribution using either  $\alpha$  (**shape**) and  $\beta$  (**scale**) or  $\alpha$  and  $\lambda$  (**rate**). The Weibull must be described using **shape** and **scale**.
- The Weibull distribution is often used to model time to failure. The wide variety of shapes of these distributions can model such features as early failure and wear.

### 4.4 Beta Distributions

The Beta distributions, like the uniform distributions, model values on a fixed finite range. If we restrict the range to be  $[0, 1]$ , Beta distributions model distributions of percentages.

<u>distribution</u>	<u>kernel</u>	<u>range of parameters</u>
Beta	$x^{\alpha-1}(1-x)^{\beta-1}$	$\alpha > 0, \beta > 0$

The uniform distribution is obtained by setting  $\alpha = \beta = 1$ .

## 4.5 The Gamma function

The pdfs, means, and variances of the Weibull, Gamma, and Beta distributions mention the **gamma function** (denoted  $\Gamma()$  in mathematics and called `gamma()` in R). This function is defined for all positive real numbers (actually, it can be defined for all complex numbers except for 0 and the negative integers), is continuous on its domain, and behaves like a shifted factorial function on the positive integers:

$$\Gamma(n) = (n - 1)! \text{ for all positive integers } n.$$

## 4.6 Summary of Important Distributions

Here is a summary of the main distributions we have encountered. This table will be provided on exams.

distribution	mass or density function	mean	variance
Poisson	$\text{dpois}(x, \lambda) = e^{-\lambda} \frac{\lambda^x}{x!}$	$\lambda$	$\lambda$
Binomial	$\text{dbinom}(x, n, \pi) = \binom{n}{x} \pi^x (1 - \pi)^{n-x}$	$n\pi$	$n\pi(1 - \pi)$
Geometric	$\text{dgeom}(x, \pi) = \pi(1 - \pi)^x$	$\frac{1}{\pi} - 1$	$\frac{1 - \pi}{\pi^2}$
Neg. Binomial	$\text{dnbinom}(x, \text{size}=s, \text{prob}=\pi) = \binom{x+n-1}{s} \pi^s (1 - \pi)^{x-s}$	$\frac{s}{\pi} - s$	$\frac{s(1 - \pi)}{\pi^2}$
Uniform	$\text{dunif}(x, a, b) = \frac{1}{b - a}$ on $[a, b]$	$\frac{b + a}{2}$	$\frac{(b - a)^2}{12}$
Std. normal	$\text{dnorm}(x, 0, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$	0	1
Normal	$\text{dnorm}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$	$\mu$	$\sigma^2$
Lognormal	$\text{dlnorm}(x, \text{meanlog}=\mu, \text{sdlog}=\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-[\ln(x)-\mu]^2/2\sigma^2}$	$e^{\mu+\sigma^2/2}$	$e^{2\mu+\sigma^2}(e^{\sigma^2} - 1)$
Exponential	$\text{dexp}(x, \lambda) = \lambda e^{-\lambda x}$	$1/\lambda$	$1/\lambda^2$
Gamma	$\text{dgamma}(x, \alpha, \text{rate}=\lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$	$\alpha/\lambda$	$\alpha/\lambda^2$
Weibull	$\text{dweibull}(x, \text{shape}=\alpha, \text{scale}=\beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha}$	$\beta\Gamma(1 + \frac{1}{\alpha})$	$\beta^2 \left[ \Gamma(1 + \frac{2}{\alpha}) - \left[ \Gamma(1 + \frac{1}{\alpha}) \right]^2 \right]$
Beta	$\text{dbeta}(x, \text{shape1}=\alpha, \text{shape2}=\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}$	$\frac{\alpha}{\alpha + \beta}$	$\frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$

## 5 Quantiles, Boxplots, and Quantile Plots

### 5.1 Quantiles

#### 5.1.1 Quantiles of Continuous Distributions

Quantiles are a generalization of percentiles. The  $q$ -quantile of a distribution  $X$  is the value  $q$  such that

$$P(X \leq q) = p$$

For a continuous distribution,  $x$  is determined by solving an integral equation:

$$\int_{-\infty}^q f(x) dx = p$$

where  $p$  is known and  $q$  is to be determined.

R provides functions to solve this equation for you when the distribution is any of the special distributions we have encountered. For example, to find the median of a  $\text{Exp}(3)$  distribution, we can use

```
> qexp(.5,3)
[1] 0.2310491
```

#### 5.1.2 Quantiles Discrete Distributions

Quantiles can also be computed for discrete distributions, but there is a wrinkle: For some values of  $p$  there may be no value  $q$  satisfying

$$P(X \leq q) = p .$$

In this case  $q$  is chosen to be the smallest value such that

$$P(X \leq q) \geq p .$$

```
> p <- qpois(.3,10); p;      # find the .30-quantile
[1] 8
> ppois(p,10)              # MORE than 30% is below (so this is the quantile)
[1] 0.3328197
> ppois(p-1,10)           # LESS than 30% is below (so this is not the quantile)
[1] 0.2202206
```

#### 5.1.3 Data Quantiles

We can also compute quantiles from data sets, but we have issues here that are similar to those that arose for discrete distributions. A number of different solutions to this problem have been proposed. (R provides 9 different quantile algorithms.) Whatever the details of the algorithm it will be the case that:

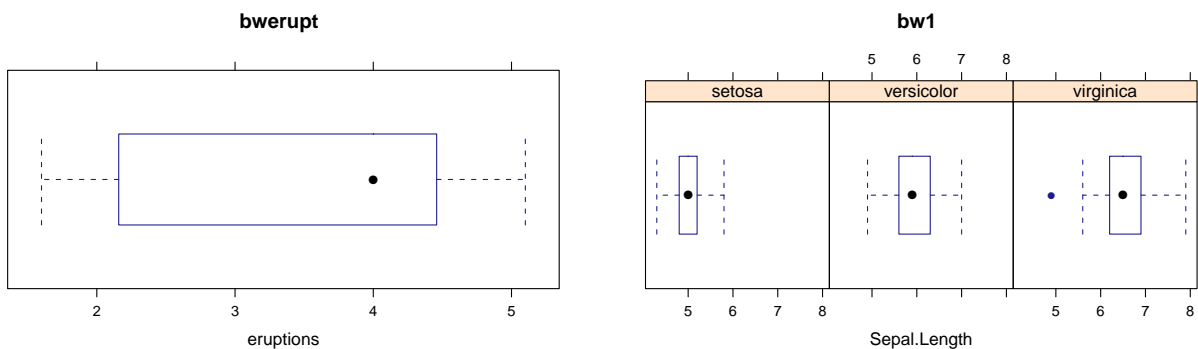
- $P(X \leq q) \approx p$  .
- The approximation is better when the data sets are larger.

## 5.2 Quartiles and Boxplots

The 0.0-, 0.25-, 0.50-, 0.75-, and 1.0-quantiles are called the **quartiles** because they delineate the quarters of the distribution. The five quartiles taken together are often called the **five number summary**.

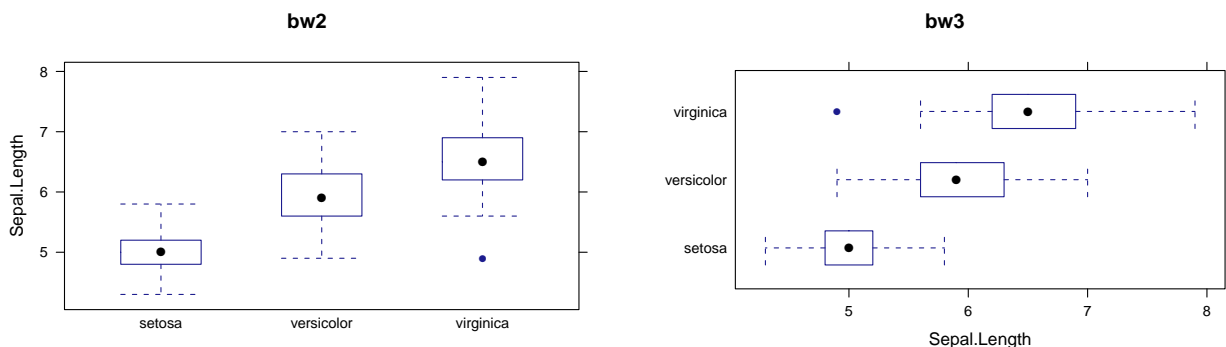
A boxplot is a way of representing the five number summary graphically. The box shows the location of the central 50% of the data. The “whiskers” show the outer two quartiles.

```
> data(faithful);
> quantile(faithful$eruptions);      # gives quartiles by default
  0%    25%   50%   75%   100%
1.60000 2.16275 4.00000 4.45425 5.10000
> fivenum(faithful$eruptions);      # bwplot actually uses this version
[1] 1.60000 2.1585 4.00000 4.4585 5.10000
> bwerupt <- bwplot(~eruptions,data=faithful,main="bwerupt");
```



It is especially useful to compare boxplots side by side.

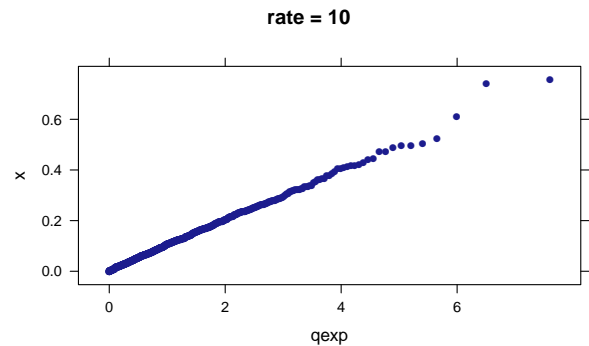
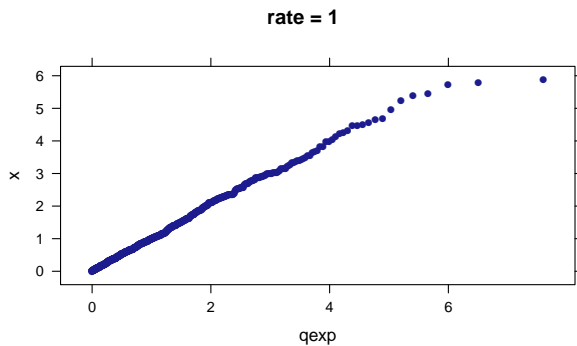
```
> data(iris);
> bw1 <- bwplot(~Sepal.Length|Species,data=iris,main="bw1")
> bw2 <- bwplot(Sepal.Length~Species,data=iris,main="bw2")
> bw3 <- bwplot(Species~Sepal.Length,data=iris,main="bw3")
```



## 5.3 Quantile-Quantile Plots

Quantile plots are scatter plots that plot the quantiles of a data distribution against the theoretical quantiles from a model distribution. If the data were “perfect”, the data and theoretical quantiles would be the same and the resulting plot would fall along a line with slope 1 and intercept 0 since the two coordinates of each point would be the same. Of course, real data is never perfect, so we should expect a bit of noise in the plot, even when the data come from a known distribution. Here are two example quantile-quantile plots for exponential data.

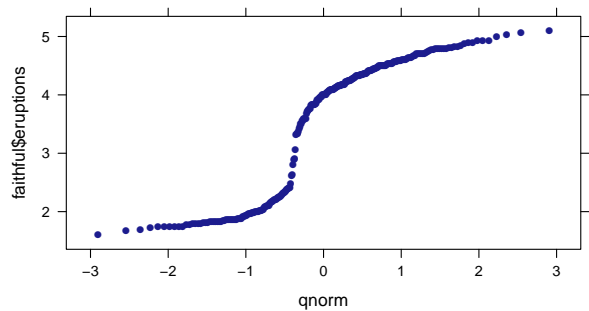
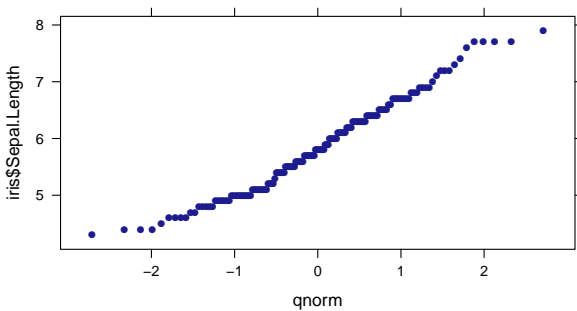
```
> x <- rexp(1000);
> qq1 <- qqmath(~x, distribution=qexp, main="rate = 1");
> x <- rexp(1000,rate=10);
> qq2 <- qqmath(~x, distribution=qexp, rate=10, main="rate = 10");
```



In fact, if the data come from a distribution that is a linear transformation of the comparison theoretical distribution, the resulting quantile plot will still fall along a line – the slope and intercept will change depending on the linear transformation involved.

Since all normal distributions are linear transformations of the standard normal distribution, this means that we can use quantile-quantile plots to see if a data set has an approximately normal distribution without needing to know the mean and standard deviation. Because testing for normality is so common, that is the default distribution used by `qqmath()`.

```
> qq3 <- qqmath(iris$Sepal.Length);
> qq4 <- qqmath(faithful$eruptions);
```



In this case, the sepal lengths look pretty good, but the eruption times do not. (We should have expected this since we have already observed that the distribution is bimodal.)

### Calibration

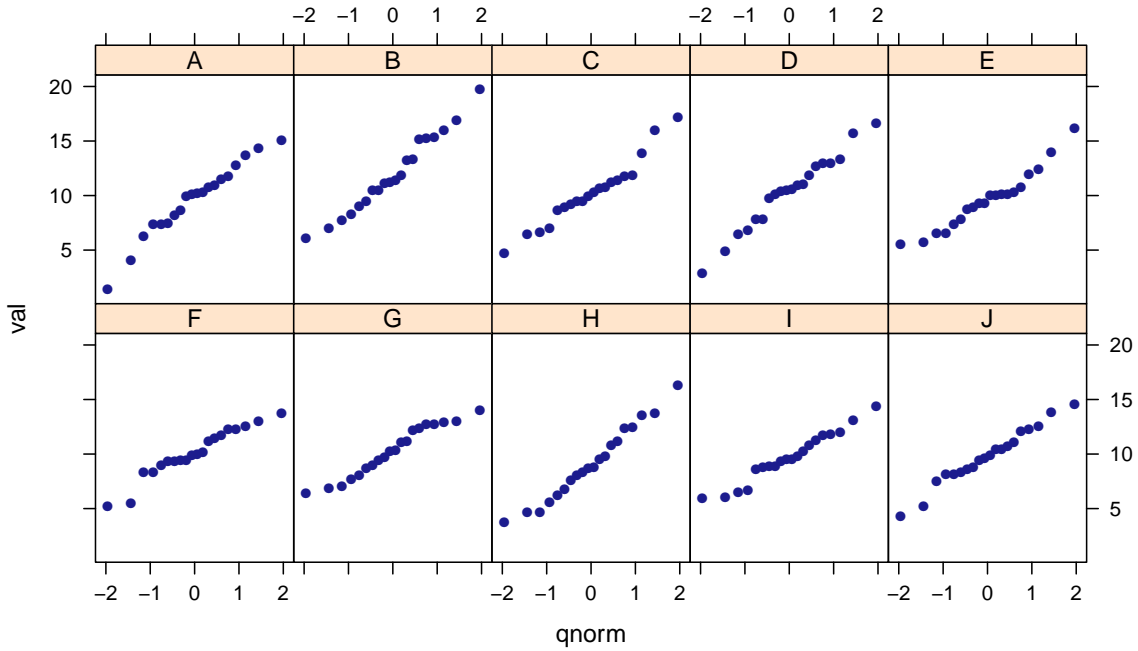
It is useful to generate some quantile-quantile plots from known distributions to get a feel for how much noise can be expected. The amount of noise is more pronounced in smaller data sets and in the tails of the distribution (where the data values are expected to vary more from sample to sample).

Here are some calibration plots using samples of size 20 and 50.

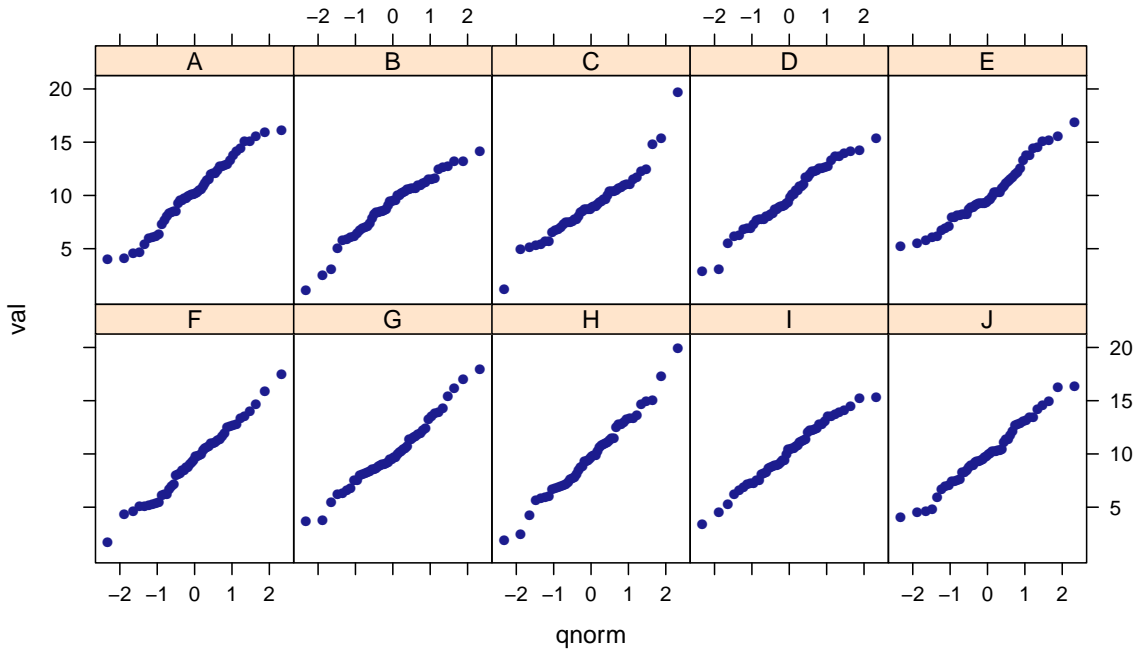
```
> n <- 10; size <- 20
> ddd <- data.frame(val = rnorm(n*size,mean=10,sd=3),
+                   set=rep(toupper(letters[1:n]),each=size));
```

```
> qq20 <- qqmath(~val|set,ddd,as.table=TRUE,main="sample size = 20");
>
> n <- 10; size <- 50
> ddd <- data.frame(val = rnorm(n*size,mean=10,sd=3),
+                   set=rep(toupper(letters[1:n]),each=size));
> qq50 <- qqmath(~val|set,ddd,as.table=TRUE,main="sample size = 50");
```

sample size = 20



sample size = 50



## 6 The Correlation Coefficient

### 6.1 Definition

The correlation coefficient  $r$  is a number that attempts to summarize the **strength** and **direction** of a **linear** association between two **quantitative** variables. The correlation coefficient is defined by the sum of the product of  $z$ -scores (with a scaling factor):

$$r = \frac{1}{n-1} \sum_{i=1}^n \underbrace{\frac{x_i - \bar{x}}{s_x}}_{z\text{-score}} \cdot \underbrace{\frac{y_i - \bar{y}}{s_y}}_{z\text{-score}}$$

We can interpret  $r$  in terms of a dot product of the following two vectors:

$$\mathbf{a} = \left\langle \frac{x_1 - \bar{x}}{s_x}, \frac{x_2 - \bar{x}}{s_x}, \dots, \frac{x_n - \bar{x}}{s_x} \right\rangle \quad \mathbf{b} = \left\langle \frac{y_1 - \bar{y}}{s_y}, \frac{y_2 - \bar{y}}{s_y}, \dots, \frac{y_n - \bar{y}}{s_y} \right\rangle$$

Using these vectors, we see that

$$r = \frac{\mathbf{a} \cdot \mathbf{b}}{n-1} = \frac{|\mathbf{a}| \cdot |\mathbf{b}| \cos \theta}{n-1} = \frac{s_x \sqrt{n-1}}{s_x} \frac{s_y \sqrt{n-1}}{s_y} \cos \theta \frac{1}{n-1} = \cos \theta$$

where  $\theta$  is the angle between  $\mathbf{a}$  and  $\mathbf{b}$ . This explains the scaling factor of  $1/(n-1)$ .

### 6.2 Properties

From the definition and this interpretation as a dot product, we get the following properties of  $r$ :

1.  $r$  is unitless.

Because we divide by the standard deviations, it does not matter what units we measure the two quantitative variables with.

2.  $r$  is symmetric in  $x$  and  $y$ , so it doesn't matter which variable has which role.
3.  $-1 \leq r \leq 1$ .

4.  $|r| = 1$  only if all the points  $\langle x_i, y_i \rangle$  fall exactly on a line. The line will have positive slope if  $r = 1$  and negative slope if  $r = -1$ .

It is important to note that  $r$  attempts to measure only *linear* associations. Other patterns may exist, and when they do,  $r$  may not be a good way to measure the association. For this reason, **always look at a plot as well as the correlation coefficient if you can.**

### 6.3 Computing in R

R provides the function `cor()` for calculating the correlation coefficient of two variables.

```
> require(MASS)
Loading required package: MASS
> data(geyser)
> cor(geyser$duration,geyser$waiting)
[1] -0.644623
```

## 7 Regression – Least Squares Best Fit

The correlation coefficient is a numerical measure of the strength and direction of a linear relationship between two variables. Now we want a way of finding an equation for the line that “fits the data best” and allows us to predict one variable (the **response** or dependent variable typically denoted  $y$ ) from the other (the **predictor** or independent variable, typically denoted  $x$ ). Our lines will be described by equations of the form

$$\hat{y} = b_0 + b_1x ,$$

and our goal is to choose the “best”  $\mathbf{b} = \langle b_0, b_1 \rangle$ . We will measure how good our line is using the method of **least squares**:

$$\text{quality of fit} = S(\mathbf{b}) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - (b_0 + b_1x_i))^2$$

A lower sum of squares indicates a better fit, so we want to make  $S$  as small as possible. Here we are using  $\hat{\mathbf{y}}(\mathbf{b})$  to indicate the fit vector if we use  $\mathbf{b}$  for our parameter estimates. We can minimize

$$S = \sum (y_i - b_0 - b_1x_i)^2$$

by determining where the two partial derivatives vanish.

$$\begin{aligned} 0 = \frac{\partial S}{\partial b_0} &= \sum 2(y_i - b_0 - b_1x_i)(-1) \\ &= (-2)[n\bar{y} - nb_0 - nb_1\bar{x}] \\ &= (-2n)[\bar{y} - b_0 - b_1\bar{x}] \end{aligned} \tag{1}$$

$$\begin{aligned} 0 = \frac{\partial S}{\partial b_1} &= \sum 2(y_i - b_0 - b_1x_i)(-x_i) \\ &= (-2) \sum [x_i y_i - b_0 x_i - b_1 x_i^2] \end{aligned} \tag{2}$$

From (1) we see that

$$\bar{y} = b_0 - b_1\bar{x} ,$$

so the point  $\langle \bar{x}, \bar{y} \rangle$  is always on the least squares regression line, and

$$b_0 = \bar{y} - b_1\bar{x} . \tag{3}$$

Substituting (3) into (2), we see that  $S$  is minimized when

$$\begin{aligned} 0 &= \sum [x_i y_i - (\bar{y} - b_1\bar{x})x_i - b_1 x_i^2] \\ &= \sum [x_i y_i - \bar{y}x_i - b_1\bar{x}x_i - b_1 x_i^2] , \end{aligned}$$

so

$$\begin{aligned} b_1 &= \frac{\sum (y_i - \bar{y})x_i}{\sum x_i^2 - \sum x_i\bar{x}} = \frac{\sum (y_i - \bar{y})x_i}{\sum (x_i - \bar{x})x_i} \\ &= \frac{\sum (y_i - \bar{y})(x_i - \bar{x})}{\sum (x_i - \bar{x})(x_i - \bar{x})} \end{aligned} \tag{4}$$

$$= \frac{\sum (y_i - \bar{y})(x_i - \bar{x})}{\sum (x_i - \bar{x})^2} = \frac{S_{xy}}{S_{xx}} \tag{5}$$

$$= \frac{1}{n-1} \sum_i \left( \frac{x_i - \bar{x}}{s_x} \cdot \frac{y_i - \bar{y}}{s_y} \right) \cdot \frac{s_y}{s_x} \tag{6}$$

$$= r \frac{s_y}{s_x} . \tag{7}$$

Equation (4) deserves some comment. In the numerator we have subtracted

$$\sum (y_i - \bar{y})(\bar{x}) = \bar{x} \sum (y_i - \bar{y}) = \bar{x} (n\bar{y} - n\bar{y}) = 0,$$

so the numerator is unchanged. The argument for the denominator is the same.

The main points of all this algebra are

1. The least squares regression line is characterized by two nice properties:
  - a) The point  $\langle \bar{x}, \bar{y} \rangle$  is on the regression line.
  - b) The slope is  $r \frac{s_y}{s_x}$ .
2. The regression line can be computed automatically very efficiently from data. We will typically let R perform the calculations for us.

## 7.1 Using `lm()` in R

Least squares regression lines are special case of what statisticians call a linear model. For this reason, the R function for computing the least squares regression line is called `lm()`.

Example 3.7

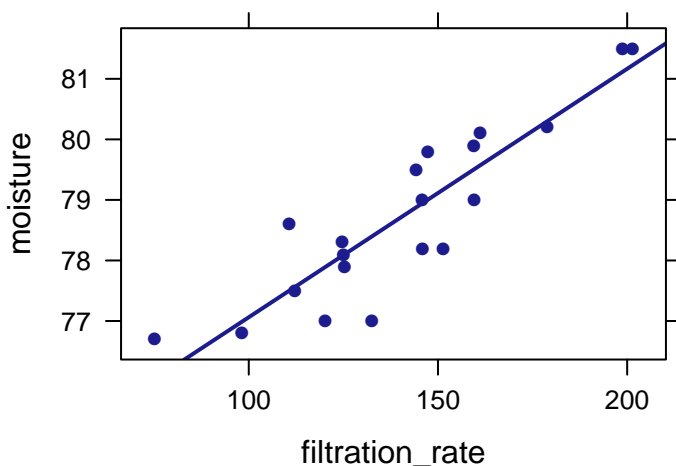
```
> require(DevFar2)
> data(e3.7)
> names(e3.7) <- c('filtration_rate', 'moisture') # replace generic names
> plot <- xyplot(moisture~filtration_rate, data=e3.7, type=c('p', 'r'))
> lm(moisture~filtration_rate, data=e3.7)
```

Call:

```
lm(formula = moisture ~ filtration_rate, data = e3.7)
```

Coefficients:

(Intercept)	filtration_rate
72.95855	0.04103



So how well does this line fit the data? There are several ways we might answer this question.

1. Look at the scatter plot.

We could give a qualitative answer based on looking at the plot. In this case there does seem to be a linear trend, but there is also a fair amount variability about the regression line.

2. Look at residuals.

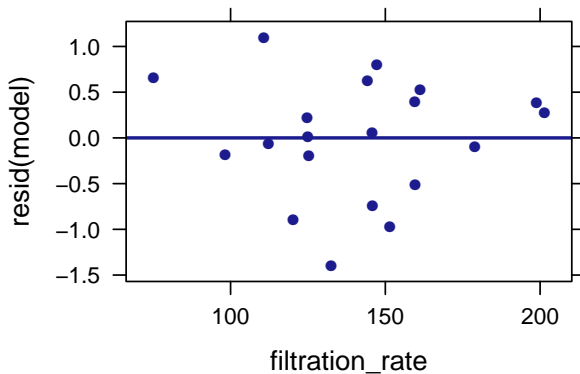
A **residual** is the difference between an observed response ( $y$ ) value and a the predicted value based on the regression line. That is

$$i\text{th residual} = e_i = y_i - \hat{y}_i = y_i - (b_0 + b_1x_i)$$

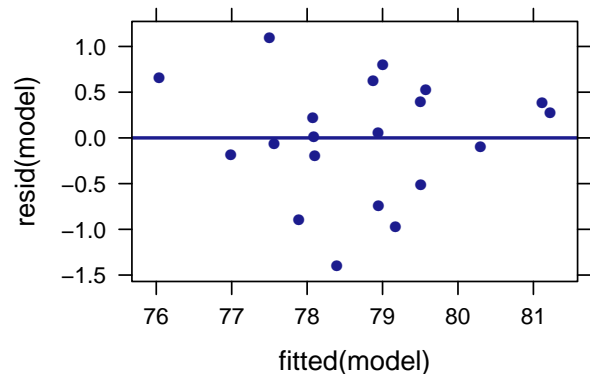
If we save the output of `lm()`, there are several functions we can apply, including `resid()` and `fitted()` which return the residuals and the fitted values for our model. Using this, we can generate two variations on the **residual plot** theme. **residual plot**.

```
> model <- lm(moisture~filtration_rate,data=e3.7)
> plot1 <- xyplot(resid(model)~filtration_rate,data=e3.7,type=c('p','r'),
+               main="Residuals vs. Predictor")
> plot2 <- xyplot(resid(model)~fitted(model),data=e3.7,type=c('p','r'),
+               main="Residuals vs. Fitted values ")
```

**Residuals vs. Predictor**



**Residuals vs. Fitted values**



Notice that the pictures are the same except for the labeling along the horizontal axis. You may find the first plot more natural, but the second is much more useful when we consider models with more than one predictor. Even with more predictors, there is still just one fitted value for each observation, so we can still make a two-dimensional plot of residuals vs. fitted values.

The residuals have some interesting properties:

- a) The mean of the residuals is always 0.

$$\sum_{i=1}^n e_i = \sum_{i=1}^n y_i - \hat{y}_i = \sum_{i=1}^n y_i - b_0 - b_1x_i = n\bar{y} - nb_0 - nb_1\bar{x} = n(\bar{y} - b_0 - b_1\bar{x}) = 0$$

because the point  $\langle \bar{x}, \bar{y} \rangle$  is on the regression line.

A good residual plot is one where the points are scattered “randomly” about the horizontal line at 0. Any patterns or trends in the residuals are an indication that there may be problems with our model.

- b) The **residual vector**  $\mathbf{e}_i = \mathbf{y} - \hat{\mathbf{y}}$  is orthogonal (perpendicular) to the **effect vector**  $\hat{\mathbf{y}} - \bar{\mathbf{y}}$ .
- c) Because  $\mathbf{y} - \hat{\mathbf{y}} \perp \hat{\mathbf{y}} - \bar{\mathbf{y}}$ ,

$$|\mathbf{y} - \bar{\mathbf{y}}|^2 = |\mathbf{y} - \hat{\mathbf{y}}|^2 + |\hat{\mathbf{y}} - \bar{\mathbf{y}}|^2 .$$

This is a simple application of the Pythagorean Theorem since clearly

$$y - \hat{y} + \hat{y} - \bar{y} = y - \bar{y}.$$

3. Look at the coefficient of determinism ( $r^2$ ).

The Pythagorean identity above has an interesting connection to the correlation coefficient  $r$ :

$$\underbrace{|y - \bar{y}|^2}_{SST} = \underbrace{|y - \hat{y}|^2}_{SSE} + \underbrace{|\hat{y} - \bar{y}|^2}_{SSM}$$

$$r^2 = \frac{SSM}{SST} = 1 - \frac{SSE}{SST}$$

For this reason  $r^2$  is often interpreted as the **proportion of the variability in the response that is explained by the linear model**. Note that  $SST = (n - 1)s_y^2$ . When  $r^2$  is large, the linear model “explains” most of the variation contributing to the variance of in the response values ( $y$ ).

We can obtain the values of  $SSM$  and  $SSE$  from R as follows.

```
> anova(model)
Analysis of Variance Table

Response: moisture
          Df Sum Sq Mean Sq F value    Pr(>F)
filtration_rate  1 31.860  31.860  71.973 1.052e-07 ***
Residuals      18  7.968   0.443
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> SSM = 31.860
> SSE = 7.968
> SST = SSM + SSE; SST
[1] 39.828
> SSM/SST;
[1] 0.7999397
```

In this case  $SST = SSM + SSE = 31.860 + 7.968 = 39.828$ , and  $r^2 = 0.7999$ .

As you might guess, there is an even easier way to get  $r^2$ :

```
> summary(model)

Call:
lm(formula = moisture ~ filtration_rate, data = e3.7)

Residuals:
    Min       1Q   Median       3Q      Max
-1.39552 -0.27694  0.03548  0.42913  1.09901

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  72.958547   0.697528  104.596 < 2e-16 ***
filtration_rate  0.041034   0.004837   8.484 1.05e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6653 on 18 degrees of freedom
Multiple R-squared:  0.7999,    Adjusted R-squared:  0.7888
F-statistic: 71.97 on 1 and 18 DF,  p-value: 1.052e-07
```

The following computations show that this does indeed agree with the values we could have gotten using `cor()`.

```
> cor(e3.7$filtration_rate,e3.7$moisture)
[1] 0.8943937
> cor(e3.7$filtration_rate,e3.7$moisture)^2
[1] 0.7999401
```

4. Look at  $s_e$ , the **residual standard error** (also called standard error about the regression line).

$$s_e = \sqrt{\frac{SSE}{n-2}}$$

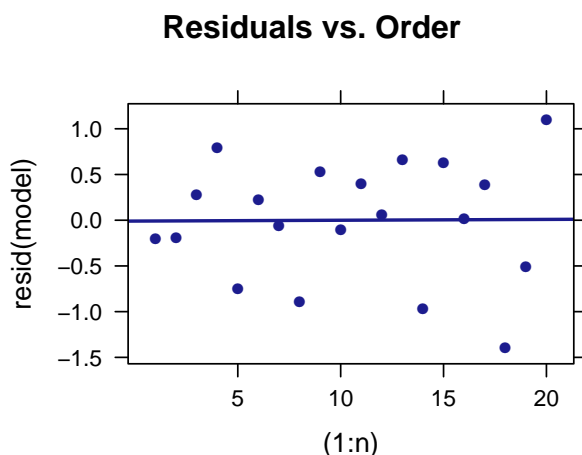
is a measure of the amount of variability about the regression line in the units of the response variable. It can be interpreted in much the same way that the standard deviation is interpreted for a single quantitative variable.

The residual standard error appears on the summary output above (the value is 0.6653). Given this value, we would expect the majority of measured responses to be less than 0.6653 of the prediction given from the regression line, but also a non-negligible proportion to be outside that range. (If the residuals were normally distributed, for example, we would expect 68% within that range and 32% outside that range.) If that level of accuracy is sufficient for our purposes, we can be quite happy with our model. If more accuracy is needed, we need to search for a better model.

5. Look at a residual time series.

If the order of our data are meaningful (for example, if they are in the order the data were collected), it can be informative to look at a plot of the residuals in order. Trends in such a plot could indicate that something is changing over time (the researchers may be getting better or worse at measuring, a machine or instrument may be wearing or failing, some part of the system may get bumped out of calibration, etc.) Such trends would become apparent in residual time series. Each of these

```
> n <- length(resid(model))      # how many residuals were there?
> plot3 <- xyplot(resid(model)~(1:n),data=e3.7,type=c('p','r'),
+               main="Residuals vs. Order")
```



6. Look for outliers and influential observations.

Regression can be strongly affected by just one or a small number of observations that do not fit the overall pattern of the rest of the data. Whenever such values are in the data set, interpreting the results can be very challenging. A number of different techniques have been developed that make regression less sensitive to outliers and influential observations. Such techniques are often referred to as **robust regression**.

Outliers and Influential observations can be detected by looking at the plots or by “leave-one-out analysis”. In leave-one-out analysis, regression models are made with each observation removed from the data set. If the regression line changes dramatically when one or a small number of values are removed from the data set, then those observations are very influential to the overall model.

## 7.2 If a Line Doesn't Fit, Don't Fit a Line

Software will fit a least squares regression line to any data set – even when it is not appropriate. There are a number of things that can be done when a simple linear model is not the best description of the relationship between two variables. The simplest of these alternatives is to

- apply a transformation to one or both variables,
- make a simple linear fit between the transformed variables.

But how does one choose a transformation? There are at least two important considerations:

1. An a priori expectation about the type of relationship one should expect.

Often theoretical reasons or previous experience lead us to expect a certain type of relationship, and we can choose transformations accordingly.

2. The shape of the scatterplot might suggest a transformation.

The textbook has a nice description of how one might find a good **power transformation**. This often works well when the relationship is monotonic.

R can fit models to transformed variables, but there are a couple of wrinkles you might not expect in the way these models are described to R. Logarithmic transformations are straightforward:

```
> data(e3.10)
> summary(lm(log(y)~log(x),e3.10))

Call:
lm(formula = log(y) ~ log(x), data = e3.10)

Residuals:
    Min       1Q   Median       3Q      Max
-0.012613 -0.006097 -0.001804  0.003166  0.018169

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.24685    0.15389  -14.60 1.93e-09 ***
log(x)       1.71361    0.03688   46.47 7.74e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.009523 on 13 degrees of freedom
Multiple R-squared:  0.994,    Adjusted R-squared:  0.9936
F-statistic:  2159 on 1 and 13 DF,  p-value: 7.741e-16
```

But transformations involving addition, subtraction, multiplication, division, and exponentiation require a bit of special coding because the normal arithmetic symbols are used to represent something else in the context of linear models.

```
> data(e3.12)
> summary(lm(y~I(1/x),e3.12))

Call:
lm(formula = y ~ I(1/x), data = e3.12)
```

## Residuals:

Min	1Q	Median	3Q	Max
-0.8042	-0.4681	-0.4157	-0.0358	2.0788

## Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.4797	0.5786	0.829	0.439
I(1/x)	83.1226	6.6573	12.486	1.61e-05 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.062 on 6 degrees of freedom

Multiple R-squared: 0.9629, Adjusted R-squared: 0.9568

F-statistic: 155.9 on 1 and 6 DF, p-value: 1.613e-05

## 8 Joint Distributions and Distributions of Sums

### 8.1 Joint and Marginal Mass Functions

Just as the pmf of a discrete random variable  $X$  must describe the probability of each value of  $X$ , the **joint pmf** of  $X$  and  $Y$  must describe the probability of any *combination* of values that  $X$  and  $Y$  could have. We could describe the joint pmf of random variables  $X$  and  $Y$  using a table like the one in the following example.

**Example 8.1.** The joint distribution of  $X$  and  $Y$  is described by the following table.

		value of $X$		
		1	2	3
value of $Y$	1	.17	.15	.08
	2	.00	.10	.10
	3	.08	.20	.12

We can now calculate the probabilities of a number of events:

- $P(X = 2) = .15 + .10 + .20 = .45$
- $P(Y = 2) = .00 + .10 + .10 = .20$
- $P(X = Y) = .17 + .10 + .12 = .39$
- $P(X > Y) = .15 + .08 + .10 = .33$

The first two probabilities show that we can recover the pmf's for  $X$  and for  $Y$  from the joint distribution. These are known as the **marginal distributions** of  $X$  and  $Y$  because they can be obtained by summing across rows or down columns, and the natural place to record those values is in the margins of the table.

		value of $X$			total
		1	2	3	
value of $Y$	1	.17	.15	.08	.40
	2	.00	.10	.10	.20
	3	.08	.20	.12	.40
total		.25	.45	.30	1.00

Of course, a table like the one in Example 8.1 is really just a description of a function, so our formal definitions are the following.

**Definition 8.2.** The *joint pmf* of a pair of discrete random variables  $\langle X, Y \rangle$  is a function  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  such that

$$f(x, y) = P(X = x \text{ and } Y = y)$$

for all  $x$  and  $y$ .

**Definition 8.3.** If  $f$  is the joint pmf for  $X$  and  $Y$ , then the *marginal distributions* of  $X$  and  $Y$  are

- $f_X(x) = \sum_y f(x, y)$ , and
- $f_Y(y) = \sum_x f(x, y)$ .

Similar definitions are possible for continuous distributions, we just replace mass functions with density functions and sums with integrals.

## 8.2 Independent Random Variables

An important (and generally easier) special case of joint distributions is the distribution of independent random variables.

**Definition 8.4.** Random variables  $X$  and  $Y$  are *independent* if for every  $x$  and  $y$

$$f(x, y) = f_X(x) \cdot f_Y(y) ,$$

that is, if

$$P(X = x \text{ and } Y = y) = P(X = x) \cdot P(Y = y) .$$

**Example 8.5.** The variables  $X$  and  $Y$  from Example 8.1 are not independent. This can be seen by observing that

- $P(X = 2) = .45$ ,
- $P(Y = 2) = .20$ , but
- $P(X = 2 \text{ and } Y = 2) = .10 \neq .45 \cdot .20 = .09$ ;

or that

- $P(X = 1) = .25$ ,
- $P(Y = 2) = .20$ , but
- $P(X = 1 \text{ and } Y = 2) = .00 \neq .25 \cdot .20$ .

Note that the fact that

- $P(X = 3) = .30$ ,
- $P(Y = 3) = .40$ , and
- $P(X = 3 \text{ and } Y = 3) = .12 = .30 \cdot .40$

is not enough to make the variables independent.

## 8.3 Sums (and other combinations) of random variables

If  $X$  and  $Y$  are jointly distributed random variables, we can define sums, differences, products, and other combinations of these variables. For example, if you roll two standard dice in a game like monopoly, you are primarily interested in the sum of the two values:  $S = X + Y$ , where  $X$  and  $Y$  are the point counts on the first and second die.

It is straightforward to determine the mass function for such a combination from the joint mass function of the variables involved.

**Example 8.6.** Continuing with Example 8.1, we can determine the pmf for  $X + Y$ . The possible values of  $X + Y$  are from 2 to 6. You should check that the table below is formed by adding the appropriate cell probabilities from the table on page 26.

value of $X + Y$		2		3		4		5		6	
probability		.17		.15		.26		.30		.12	

## 8.4 More Than Two Variables

All the definitions of the previous sections can be extended to handle the case of  $k \geq 3$  random variables. A joint pmf is then a function  $f : \mathbb{R}^k \rightarrow \mathbb{R}$  such that

$$f(x_1, \dots, x_k) = P(X_1 = x_1 \text{ and } \dots \text{ and } X_k = x_k) .$$

Marginal distributions (for any subset of the variables) are obtained by summing over all values of the other variables.

The only definition that is somewhat tricky is the definition of independence.

**Definition 8.7.** Let  $X_1, \dots, X_k$  be  $k$  jointly distributed discrete random variables with joint pmf  $f : \mathbb{R}^k \rightarrow \mathbb{R}$ . Then

1.  $X_1, \dots, X_k$  are *pairwise independent* if  $X_i$  and  $X_j$  are independent whenever  $i \neq j$ .
2.  $X_1, \dots, X_k$  are *independent* if the joint pmf factors:

$$f(x_1, \dots, x_k) = f_{X_1}(x_1) \cdot f_{X_2}(x_2) \cdots f_{X_k}(x_k).$$

Independence is stronger than pairwise independence.

## 8.5 The Mean and Variance of a Sum

Example 8.6 is especially important because a number of important situations can be expressed as sums of random variables.

- A binomial random variable is the sum of Bernoulli random variables.

If  $X \sim \text{Binom}(n, \pi)$ , then  $X = X_1 + X_2 + \cdots + X_n$  where

$$X_i = \begin{cases} 0 & \text{if the } i\text{th trial is a failure,} \\ 1 & \text{if the } i\text{th trial is a success.} \end{cases}$$

Each  $X_i \sim \text{Binom}(1, \pi)$ , and the  $X_i$ 's are independent by the definition of the binomial distributions.

- A negative binomial random variable is the sum of geometric random variables.

If  $X \sim \text{NBinom}(r, \pi)$ , then  $X = X_1 + X_2 + \cdots + X_r$  where each  $X_i \sim \text{NBinom}(1, \pi)$ .  $X_i$  is the number of failures between the success  $i - 1$  and success  $i$ . Once again, the  $X_i$ 's are independent of each other since the trials are independent.

- When we compute a sample mean, we first add all of the data values.

If we have an independent random sample from some population, then the sample mean is a random variable and

$$\bar{X} = \frac{1}{n} (X_1 + X_2 + \cdots + X_n),$$

where  $X_i$  is the value of the  $i$ th individual in the sample. This means that once we understand sums of random variables, we can reduce the study of sample means to the study of a sample of size 1.

For these reasons and more, the distribution of sums of random variables will arise again and again. Fortunately, the next theorem makes dealing with these sums very convenient.

**Theorem 8.8.** Let  $X$  and  $Y$  be discrete random variables. Then

1.  $E(X + Y) = E(X) + E(Y)$ ,
2.  $E(X \cdot Y) = E(X) \cdot E(Y)$ , provided  $X$  and  $Y$  are independent, and
3.  $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$ , provided  $X$  and  $Y$  are independent.

*Proof.* All three parts of this theorem are proved by algebraic manipulation of the definitions involved. For the second and third parts, it will be important that  $X$  and  $Y$  be independent – otherwise the argument fails. Independence is not required for the first part.

Let  $f$  and  $g$  be the (marginal) pmfs of  $X$  and  $Y$ , respectively, and let  $h$  be the joint pmf. Then

$$E(X + Y) = \sum_{x,y} (x + y)P(X = x \& Y = y) \quad (8)$$

$$= \sum_{x,y} x h(x, y) + y h(x, y) \quad (9)$$

$$= \sum_x \sum_y x h(x, y) + \sum_y \sum_x y h(x, y) \quad (10)$$

$$= \sum_x x \sum_y h(x, y) + \sum_y y \sum_x h(x, y) \quad (11)$$

$$= \sum_x x \cdot P(X = x) + \sum_y y \cdot P(Y = y) \quad (12)$$

$$= E(X) + E(Y) \quad (13)$$

Similarly, for part (2),

$$E(XY) = \sum_{x,y} xyP(X = x \& Y = y) \quad (14)$$

$$= \sum_{x,y} xyf(x)g(y) \quad (15)$$

$$= \sum_x \sum_y xy \cdot f(x)g(y) \quad (16)$$

$$= \sum_x xf(x) \sum_y y \cdot g(y) \quad (17)$$

$$= \sum_x xf(x) E(Y) \quad (18)$$

$$= E(X) \cdot E(Y) \quad (19)$$

In the argument above, we needed independence to conclude that 14 and 15 are equivalent.

For the variance, we use part (2) of the theorem and our short-cut method for calculating the variance.

$$\text{Var}(X + Y) = E((X + Y)^2) - [E(X + Y)]^2 \quad (20)$$

$$= E(X^2 + 2XY + Y^2) - [E(X) + E(Y)]^2 \quad (21)$$

$$= E(X^2) + E(2XY) + E(Y^2) - [(E(X))^2 + 2E(X)E(Y) + (E(Y))^2] \quad (22)$$

$$= E(X^2) + 2E(X)E(Y) + E(Y^2) - [E(X)^2 + 2E(X)E(Y) + E(Y)^2] \quad (23)$$

$$= E(X^2) + E(Y^2) - E(X)^2 - E(Y)^2 \quad (24)$$

$$= E(X^2) - E(X)^2 + E(Y^2) - E(Y)^2 \quad (25)$$

$$= \text{Var}(X) + \text{Var}(Y) \quad (26)$$

□

Induction allows us to easily extend these results from sums of two random variables to sums of three or more random variables.

**Example 8.9.** Returning to Example 8.1, notice that

$$E(X) = 1(0.25) + 2(.45) + 3(.30) = 2.05$$

$$E(Y) = 1(0.40) + 2(.20) + 3(.40) = 2.0$$

$$E(X + Y) = 2(.17) + 3(.15) + 4(.26) + 5(.40) + 6(.12) = 4.05$$

We can apply these tools to compute the means and variances of binomial distributions.

**Theorem 8.10.** *If  $X \sim \text{Binom}(n, \pi)$ , then  $E(X) = n\pi$  and  $\text{Var}(X) = n\pi(1 - \pi)$ .*

*Proof.* If  $X \sim \text{Binom}(n, \pi)$ , then we can write  $X = X_1 + X_2 + \cdots + X_n$  where  $X_i \sim \text{Binom}(1, \pi)$ . We already know that  $E(X_i) = \pi$ , and  $\text{Var}(X_i) = \pi(1 - \pi)$ , and the  $X_i$ 's are independent. So

$$E(X) = \sum_{i=1}^n E(X_i) = n\pi, \text{ and}$$

$$\text{Var}(X) = \sum_{i=1}^n \text{Var}(X_i) = n\pi(1 - \pi).$$

□

## 8.6 The Distribution of a Sum

We now have methods for computing the mean and variance of a sum of independent random variables, but these methods do not tell us anything about the shape of that distribution. There is, however, one situation where the shape of the distribution is also easily obtainable.

**Theorem 8.11.** *Let  $X$  and  $Y$  be normal random variables, then  $X + Y$  is also a normal random variable.*

Be warned that this property does not hold for most random variables.

**Example 8.12.** Let  $X \sim \text{Norm}(100, 20)$  and  $Y \sim \text{Norm}(80, 15)$  be independent random variables. Then  $X + Y \sim \text{Norm}(180, \sqrt{20^2 + 15^2}) = \text{Norm}(180, 25)$ . Similarly,  $X - Y = X + (-Y) \sim \text{Norm}(20, 15)$  because  $-Y \sim \text{Norm}(-80, 15)$ .

This latter distribution can be used to calculate  $P(Y > X) = P(X - Y < 0)$ :

```
| pnorm(0, 20, 25)
|[1] 0.2118554
```

## 8.7 Covariance

Let's take another look at the proof of Theorem 8.8. If  $X$  and  $Y$  are independent, then  $E(XY) = E(X)E(Y)$ , so those terms cancel in (23). But what if  $X$  and  $Y$  are not independent? Then we get the following equation:

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2E(XY) - 2E(X)E(Y).$$

Notice that if  $X = Y$  (i.e.,  $P(X = Y) = 1$ ), then

$$E(XY) - E(X)E(Y) = E(X^2) - E(X)^2 = \text{Var}(X).$$

This leads us to make the following definition and theorem.

**Definition 8.13.** Let  $X$  and  $Y$  be jointly distributed random variables, then the *covariance* of  $X$  and  $Y$  is defined by

$$\text{Cov}(X, Y) = E(XY) - E(X)E(Y).$$

An immediate consequence of this definition is the following lemma.

**Lemma 8.14.** *Let  $X$  and  $Y$  be jointly distributed random variables (possibly dependent), then*

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y).$$

□

Lemma 8.14 can be generalized to more than two random variables. You are asked to do this in Exercise 9.

The intuition for the name *covariance* – besides its connection to the variance – is easier to explain using the following lemma.

**Lemma 8.15.** *Let  $X$  and  $Y$  be jointly distributed random variables, then*

$$\text{Cov}(X, Y) = \mathbf{E} [(X - \mu_X)(Y - \mu_Y)] .$$

*Proof.* Exercise 10. □

The expression  $(X - \mu_X)(Y - \mu_Y)$  is positive when  $X$  and  $Y$  are either both greater than their means or both less than their means; it is negative when one is larger and the other smaller. So  $\text{Cov}(X, Y)$  will be positive if  $X$  and  $Y$  usually large together or small together – that is, if they *vary together*. Similarly,  $\text{Cov}(X, Y)$  will be negative when  $X$  and  $Y$  vary in opposite directions. And  $\text{Cov}(X, Y)$  will be near 0 when large values of  $X$  occur about equally with large or small values of  $Y$ . In particular, as we have already seen,

**Lemma 8.16.** *If  $X$  and  $Y$  are independent, then  $\text{Cov}(X, Y) = 0$ .* □

Unfortunately, the converse of Lemma 8.16 is not true. (See Exercise 11.) Computation of covariance is often aided by the following lemma.

**Lemma 8.17.** *Let  $X$ , and  $Y$  be jointly distributed random variables. Then*

1.  $\text{Cov}(X, Y) = \text{Cov}(Y, X)$ .
2.  $\text{Cov}(a + X, Y) = \text{Cov}(X, Y)$ .
3.  $\text{Cov}(aX, bY) = ab \text{Cov}(X, Y)$ .
4.  $\text{Cov}(X, Y + Z) = \text{Cov}(X, Y) + \text{Cov}(X, Z)$ .
5.  $\text{Cov}(aW + bX, cY + dZ) = ac \text{Cov}(W, Y) + ad \text{Cov}(W, Z) + bc \text{Cov}(X, Y) + bd \text{Cov}(X, Z)$ .

*Proof.* Each of these can be proved by straightforward algebraic manipulations using the previously established properties of expected value. See Exercise 12. □

Lemma 8.17 allows us to calculate the variance of sums of more than two random variables.

**Lemma 8.18.** *Let  $X_1, X_2, \dots, X_k$  be jointly distributed random variables. Then*

$$\begin{aligned} \text{Var}(X_1 + X_2 + \dots + X_k) &= \sum_i \text{Var}(X_i) + 2 \sum_{i < j} \text{Cov}(X_i, X_j) \\ &= \sum_{i, j} \text{Cov}(X_i, X_j) . \end{aligned}$$

*Proof.* Exercise 9. □

Lemma 8.15 also motivates the definition of a “unitless” version of covariance, called the **correlation coefficient**. The correlation coefficient for a pair of random variables behaves very much like the correlation coefficient we used with data in the context of regression.

**Definition 8.19.** The **correlation coefficient**  $\rho$  of two random variables  $X$  and  $Y$  is defined by

$$\rho = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X) \cdot \text{Var}(Y)}} .$$

If we let  $\sigma_{XY} = \text{Cov}(X, Y)$ , then this is equivalent to

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} .$$

**Lemma 8.20.** *Let  $\rho$  be the correlation coefficient of random variables  $X$  and  $Y$ . Then*

$$-1 \leq \rho \leq 1 .$$

*Proof.*

$$\begin{aligned} 0 \leq \operatorname{Var}\left(\frac{X}{\sigma_X} + \frac{Y}{\sigma_Y}\right) &= \operatorname{Var}\left(\frac{X}{\sigma_X}\right) + \operatorname{Var}\left(\frac{Y}{\sigma_Y}\right) + 2 \operatorname{Cov}\left(\frac{X}{\sigma_X} + \frac{Y}{\sigma_Y}\right) \\ &= \frac{1}{\sigma_X^2} \operatorname{Var}(X) + \frac{1}{\sigma_Y^2} \operatorname{Var}(Y) + \frac{2}{\sigma_X \sigma_Y} \operatorname{Cov}(X, Y) \\ &= 1 + 1 + 2\rho \end{aligned}$$

from which it follows that  $\rho \geq -1$ .

The other inequality is proved similarly by considering  $\operatorname{Var}\left(\frac{X}{\sigma_X} - \frac{Y}{\sigma_Y}\right)$ . □

**Lemma 8.21.** *Let  $X$  and  $Y$  be jointly distributed variables such that  $\rho = \pm 1$ . Then there are constants  $a$  and  $b$  such that*

$$P(Y = a + bX) = 1 .$$

*Proof.* If  $\rho = -1$ , then  $\operatorname{Var}\left(\frac{X}{\sigma_X} + \frac{Y}{\sigma_Y}\right) = 0$ , so for some constant  $c$ ,

$$P\left(\frac{X}{\sigma_X} + \frac{Y}{\sigma_Y} = c\right) = 1 .$$

Algebraic rearrangement gives

$$P\left(Y = c\sigma_Y - \frac{\sigma_Y}{\sigma_X}X\right) = 1 .$$

Note that the slope of this linear transformation is negative.

The proof in the case that  $\rho = 1$  is similar. When  $\rho = 1$  the slope will be positive. □

## 8.8 Summary of some important facts about expected value and variance

Let  $X$  and  $Y$  be random variables and let  $a$  and  $b$  be real numbers. Then

### expected value rule

1)  $E(aX) = a E(X)$

2)  $E(X + b) = E(X) + b$

3)  $E(X + Y) = E(X) + E(Y)$

### variance rule

$\operatorname{Var}(aX) = a^2 \operatorname{Var}(X)$

$\operatorname{Var}(X + b) = \operatorname{Var}(X)$

$\operatorname{Var}(X + Y) = \operatorname{Var}(X) + \operatorname{Var}(Y)$  provided  $X$  and  $Y$  are independent.

## 8.9 Exercises

**7** Find an example of three random variables that are pairwise independent but not independent.

**8** The expected value and variance of a sum of 3 or more independent random variables are also easy to compute.

a) Show that for any random variables  $X$ ,  $Y$ , and  $Z$ ,

$$E(X + Y + Z) = E(X) + E(Y) + E(Z) .$$

b) Show that for independent random variables  $X$ ,  $Y$ , and  $Z$ ,

$$\text{Var}(X + Y + Z) = \text{Var}(X) + \text{Var}(Y) + \text{Var}(Z) .$$

Similar results hold for any number of random variables.

[Hint: Use what you already know about sums of two random variables.]

**9**

- a) Determine the formula for  $\text{Var}(X+Y+Z)$  in the general case where the variables may not be independent. Show that your formula is correct.
- b) Generalize this to sums of  $k$  random variables  $X = X_1 + X_2 + \cdots + X_k$ .

**10** Prove Lemma 8.15.

**11** Describe a joint distribution of random variables  $X$  and  $Y$  such that  $\text{Cov}(X, Y) = 0$ , but  $X$  and  $Y$  are not independent.

**12** Prove Lemma 8.17.

**13** Suppose  $X \sim \text{Norm}(20, 3)$  and  $Y \sim \text{Norm}(30, 4)$  and that  $X$  and  $Y$  are independent. Determine the following probabilities.

- a)  $P(X \geq 30)$
- b)  $P(X \geq Y)$
- c)  $P(2X \geq 30)$
- d)  $P(3X \geq 2Y)$

**14** Suppose  $X_1, X_2, X_3$ , and  $X_4$  are independent random variables and that each has mean 10 and standard deviation 2.

- a) What is  $E(X_1 + X_2 + X_3 + X_4)$ ?
- b) What is  $E(\frac{X_1+X_2+X_3+X_4}{4})$ ?
- c) What is  $\text{Var}(\frac{X_1+X_2+X_3+X_4}{4})$ ?

**15** Let  $X$  and  $Y$  be as in Example 8.1.

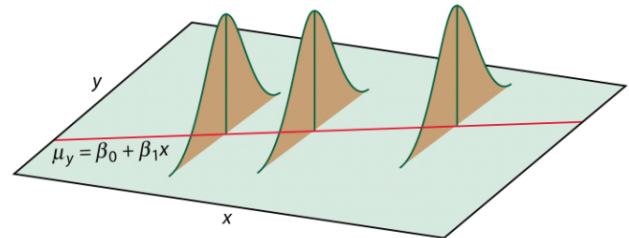
- a) Compute  $\text{Cov}(X, Y)$ .
- b) Compute the correlation coefficient  $\rho$ .

## 9 Regression Revisited

### 9.1 The simple linear regression model

The model for simple linear regression is

$$\underbrace{Y_i}_{\text{data}} = \underbrace{\beta_0 + \beta_1 x_i}_{\text{fit}} + \underbrace{\epsilon_i}_{\text{error}} \quad \epsilon_i \sim N(0, \sigma)$$



The assumptions for the linear regression model are

- 1.
- 2.
- 3.

### 9.2 Estimating the parameters

We determine estimates for  $b_1$  and  $b_0$  for  $\beta_1$  and  $\beta_0$  by minimizing the sum of the squares of the residuals. As we have seen, the least squares regression line satisfies

- $b_1 = r \frac{s_y}{s_x}$ .
- $\langle \bar{x}, \bar{y} \rangle$  is always on the regression line (so  $b_0 = \bar{y} - b_1 \bar{x}$ ).

To emphasize that we use  $b_1$  and  $b_0$  are used to estimate values of the response variable  $y$  we write

$$\hat{y} = b_1 x + b_0$$

(The “hat” on top indicates an estimate. We can also write  $\hat{\beta}_1$  and  $\hat{\beta}_0$  instead of  $b_1$  and  $b_0$ .)

We now have our least squares estimates for  $\beta_0$  and  $\beta_1$ , what about for the other parameter in the model, namely  $\sigma^2$ ?

$$\hat{\sigma}^2 = s^2 = \frac{SSE}{n-2} = \frac{\sum_i (y_i - \hat{y})^2}{n-2}$$

Why  $n - 2$ ?

- This is the correct denominator to make the estimate unbiased:  $E(\hat{\sigma}^2) = \sigma^2$ .
- This is the correct degrees of freedom since (roughly) we lose a degree of freedom for estimating each of  $\beta_0$  and  $\beta_1$ .

As usual, we will let  $s = \hat{\sigma} = \sqrt{\hat{\sigma}^2}$ .

### 9.3 R and regression – the lm() command

**Example.** We want to predict ACT scores from SAT scores. We sample scores from 60 students who have taken both tests.

```
> xyplot(ACT~SAT,act,panel=panel.lm)
> act.lm <- lm(ACT~SAT,data=act)
> plot(act.lm)
> anova(act.lm)
Analysis of Variance Table
```

```
Response: ACT
      Df Sum Sq Mean Sq F value    Pr(>F)
SAT     1  874.37   874.37  116.16 1.796e-15 ***
Residuals 58  436.56     7.53
```

```
> act[47,]
  Student SAT ACT
47      47 420  21
> actAdj <- act[-47,]; lm.actAdj <- lm(ACT~SAT,actAdj)
> summary(lm.actAdj)
```

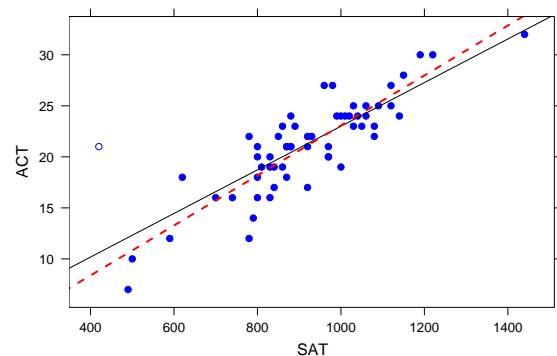
```
      Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.427642   1.691981  -0.844   0.402
SAT           0.024498   0.001807  13.556 <2e-16 ***
```

```
Residual standard error: 2.333 on 57 degrees of freedom
Multiple R-Squared: 0.7632, Adjusted R-squared: 0.7591
F-statistic: 183.8 on 1 and 57 DF, p-value: < 2.2e-16
```

```
> summary(act.lm)
```

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.626282   1.844230   0.882   0.382
SAT           0.021374   0.001983  10.778 1.80e-15 ***
```

```
Residual standard error: 2.744 on 58 degrees of freedom
Multiple R-Squared: 0.667, Adjusted R-squared: 0.6612
F-statistic: 116.2 on 1 and 58 DF, p-value: 1.796e-15
```



### 9.4 Since the model is based on normal distributions and we don't know $\sigma$ ...

1. Regression is going to be sensitive to **outliers**. Outliers with especially large or small values of the independent variable are especially **influential**.
2. We can check if the model is reasonable by looking at our **residuals**:
  - a) Use histograms or (better) normal quantile plots to check overall normality. We are looking for a roughly bell-shaped histogram or a roughly linear normal quantile plot.
  - b) Plots of
    - residuals vs  $x$ , or
    - residuals vs. order, or
    - residuals vs. fit      note: fit =
 indicate whether the standard deviation appears to remain constant throughout. We are looking to NOT see any clear pattern in these plots. A pattern would indicate something other than randomness is influencing the residuals.
3. We can do **inference** for  $\beta_0$ ,  $\beta_1$ , etc. using the **t distributions**, we just need to know the corresponding  $eSE$  and degrees of freedom.

## 9.5 Inference for Regression

Four inference situations:

- $\beta_0$  (usually uninteresting),
- $\beta_1$  (and model utility),
- predicting the mean response for a given value of the explanatory variable,
- predicting an individual response for a given value of the explanatory variable.

In each case

$$\frac{\text{estimate} - \text{parameter}}{eSE} \sim T(n - 2)$$

and confidence intervals have the form

$$\text{estimate} \pm t_* eSE$$

We won't ever compute these standard errors by hand, but here are the formulas. Note that the degrees of freedom is  $n - 2$ .

<u>parameter</u>	<u>estimate</u>	<u>eSE</u>	<u>df</u>
$\beta_0$	$\hat{\beta}_0 = b_0$	$eSE_{b_0} = s \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2}}$	$n - 2$
$\beta_1$	$\hat{\beta}_1 = b_1$	$eSE_{b_1} = \frac{s}{\sqrt{\sum(x_i - \bar{x})^2}}$	$n - 2$
$\hat{y}$ (mean response)	$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x^*$	$eSE = s \sqrt{\frac{1}{n} + \frac{x^* - \bar{x}}{\sum(x_i - \bar{x})^2}}$	$n - 2$
$\hat{y}$ (individual response)	$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x^*$	$eSE = s \sqrt{1 + \frac{1}{n} + \frac{x^* - \bar{x}}{\sum(x_i - \bar{x})^2}}$	$n - 2$

The estimated standard errors for  $\beta_1$  and  $\beta_0$  are easy to spot in the R output.

The standard error for confidence and prediction intervals depend on the value of the predictor  $x$ , so things are a bit more complicated, but R provides a function that does all the heavy lifting for us.

### 9.5.1 Confidence Intervals vs. Prediction Intervals

Recall that our goal was to make predictions of  $y$  from  $x$ . As you would probably expect, such predictions will also be described using confidence intervals. Actually there are two kinds of predictions:

1. Confidence intervals for the mean response
2. Prediction intervals are confidence intervals for a future observation.

Notice that for predictions (confidence intervals and prediction intervals), the standard errors depend on  $x^*$  (the  $x$  value for which you want a prediction made), so it is not possible for the computer to tell you what  $eSE$  is until you decide what prediction you want to make. R can do the whole thing for you, but there is no option in Rcmdr to do it. Here is how R makes prediction and confidence intervals of SAT scores corresponding to an ACT score of 25.

## 9.6 Some Examples

### 9.6.1 Example 11.5 – strength vs. carbonation depth

```

> data(e11.7)
> e11.7 -> steelbars
> names(steelbars) <- c('carbdepth','strength')
> steel.model <- lm(strength ~ carbdepth, steelbars)
> summary(steel.model)

Call:
lm(formula = strength ~ carbdepth, data = steelbars)

Residuals:
    Min       1Q   Median       3Q      Max
-4.1317 -2.0043 -0.7488  2.1366  5.1439

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 27.18294    1.65135   16.461 1.88e-11 ***
carbdepth   -0.29756    0.04116   -7.229 2.01e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.864 on 16 degrees of freedom
Multiple R-squared:  0.7656,    Adjusted R-squared:  0.7509
F-statistic: 52.25 on 1 and 16 DF,  p-value: 2.013e-06

> anova(steel.model)
Analysis of Variance Table

Response: strength
      Df Sum Sq Mean Sq F value    Pr(>F)
carbdepth  1  428.62   428.62   52.253 2.013e-06 ***
Residuals 16  131.24     8.20
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> predict(steel.model, newdata=data.frame(carbdepth=45))
1
13.79268
> predict(steel.model, newdata=data.frame(carbdepth=45),interval='confidence')
      fit      lwr      upr
1 13.79268 12.18525 15.40011
> predict(steel.model, newdata=data.frame(carbdepth=45),interval='prediction')
      fit      lwr      upr
1 13.79268  7.512036 20.07333
> predict(steel.model, newdata=data.frame(carbdepth=45),interval='confidence', level=.99)
      fit      lwr      upr
1 13.79268 11.57799 16.00738
> predict(steel.model, newdata=data.frame(carbdepth=45),interval='prediction', level=.99)
      fit      lwr      upr
1 13.79268  5.13928 22.44608
> predict(steel.model, newdata=steelbars,interval='confidence')
      fit      lwr      upr
1  24.802446 21.924677 27.68022
2  22.719518 20.352145 25.08689
3  22.273176 20.008694 24.53766
4  21.231712 19.194769 23.26865

```

```

5 21.231712 19.194769 23.26865
6 19.000002 17.362911 20.63709
7 18.256099 16.713130 19.79907
8 18.256099 16.713130 19.79907
9 16.768293 15.330346 18.20624
10 15.875609 14.439428 17.31179
11 15.280487 13.819192 16.74178
12 13.792681 12.185254 15.40011
13 12.304875 10.457440 14.15231
14 12.304875 10.457440 14.15231
15 10.817069 8.666962 12.96718
16 10.817069 8.666962 12.96718
17 9.626824 7.205036 12.04861
18 7.841456 4.980495 10.70242

```

### 9.6.2 Example 11.2 – mortar dry density vs. mortar air content

```

> data(e11.2)
> e11.2 -> bricks
> names(bricks) <- c('drydensity','air')
> bricks.model <- lm(air ~ drydensity,bricks)
> summary(bricks.model)

Call:
lm(formula = air ~ drydensity, data = bricks)

Residuals:
    Min       1Q   Median       3Q      Max
-4.1834 -1.2176 -0.8351  1.4913  6.9273

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 126.2489     2.2544  56.001 < 2e-16 ***
drydensity  -0.9176     0.1460  -6.286 2.81e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.941 on 13 degrees of freedom
Multiple R-squared:  0.7524,    Adjusted R-squared:  0.7334
F-statistic: 39.51 on 1 and 13 DF,  p-value: 2.808e-05

> anova(bricks.model)
Analysis of Variance Table

Response: air
          Df Sum Sq Mean Sq F value    Pr(>F)
drydensity  1 341.73  341.73  39.508 2.808e-05 ***
Residuals  13 112.44    8.65
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

## 9.6.3 Example 11.3 – abrasive wear loss vs. retained austentite content (%)

Note: This data set does not exactly match the values listed in the book.

```
> data(e11.5)
> e11.5 -> alloy
> names(alloy) <- c('austentite','wear')
> alloy.model <- lm(wear ~ austentite,alloy)
> summary(alloy.model)

Call:
lm(formula = wear ~ austentite, data = alloy)

Residuals:
    Min       1Q   Median       3Q      Max
-0.217417 -0.148373  0.001107  0.114513  0.541629

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.772492   0.094782   8.150 6.84e-07 ***
austentite   0.007809   0.001902   4.105 0.000936 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1985 on 15 degrees of freedom
Multiple R-squared:  0.5291,    Adjusted R-squared:  0.4977
F-statistic: 16.85 on 1 and 15 DF,  p-value: 0.0009362

> anova(alloy.model)
Analysis of Variance Table

Response: wear
      Df Sum Sq Mean Sq F value    Pr(>F)
austentite  1  0.66427  0.66427  16.853 0.0009362 ***
Residuals 15  0.59123  0.03942
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 9.7 Where does regression go from here?

- Regression with transformed data
- More than one explanatory variable
- Categorical explanatory variables
- Logistic regression  
(allows for categorical *response*)
- Robust regression

## 9.7.1 Regression with transformed data

Reasons to transform data:

- better fit
- better residual behavior
- theoretical model

We have already looked at ways to choose transformations.

## 10 Multiple Regression Revisited

Multiple regression handles more than one explanatory variable. A typical model with  $k$  explanatory variables (often called **predictors** or **regressors**) has the form

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

where  $\varepsilon \sim \text{Norm}(0, \sigma)$ . This model is called the **general additive model**. The model has  $k$  predictors and  $p = k + 1$  parameters.

### 10.1 Higher order terms and interaction

Many interesting regression models are formed by using transformations or combinations of explanatory variables as predictors. Suppose we have two predictors. Here are some possible models

- First-order model (same as above):  $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$
- Second-order, no interaction:  $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \varepsilon$
- First-order, plus interaction:  $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \varepsilon$
- Complete Second-order model:  $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \varepsilon$

### 10.2 Interpreting the parameters

- $\beta_i$  for  $i > 0$  can be thought of as adjustments to the baseline affect given by  $\beta_0$ . (This is especially useful when the predictors are categorical and the intercept has a natural interpretation.)
- When there are no interaction or higher order terms in the model, the parameter  $\beta_i$  can be interpreted as the amount we expect the response to change if we increase  $x_i$  by 1 and *leave all other predictors fixed*. The effects due to different predictors are **additive** and do not depend on the values of the other predictors. Graphically this gives us parallel lines or planes.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

- With higher order terms in the model, the dependence of the response on one predictor (with all other predictors fixed) may not be linear.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_2$$

- With interaction terms, the model is no longer additive: the effect of changing one predictor may depend on the values of the other predictors. Graphically, our lines or curves are no longer parallel.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$$

### 10.3 Fitting the model

The model can again be fit using least squares (or maximum likelihood) estimators.<sup>1</sup> As you can imagine, the formulas for estimated standard errors become quite complicated, but statistical software will easily output this information for you, so we will focus on using the output from R and interpreting the results.

**Example.** Let's fit a first-order additive model to the data in Example 11.12.

```
> data(e11.12); e11.12 -> wire;
> wire.model <- lm(Strength ~ Force + Power + Temperature + Time, data=wire);
> summary(wire.model);
```

Call:

```
lm(formula = Strength ~ Force + Power + Temperature + Time, data = wire)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-11.0900  -1.7608  -0.3067   2.4392   7.5933
```

Coefficients:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -37.47667   13.09964  -2.861  0.00841 **
Force         0.21167    0.21057   1.005  0.32444
Power         0.49833    0.07019   7.100 1.93e-07 ***
Temperature   0.12967    0.04211   3.079 0.00499 **
Time          0.25833    0.21057   1.227 0.23132
```

---

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

Residual standard error: 5.158 on 25 degrees of freedom

Multiple R-squared: 0.714, Adjusted R-squared: 0.6682

F-statistic: 15.6 on 4 and 25 DF, p-value: 1.592e-06

### 10.4 Model Utility Test

So how good is this model? There are several things we can look at to answer this question.

1.  $R^2 = 71.4\%$

Recall that

$$R^2 = \frac{SSM}{SST} = 1 - \frac{SSE}{SST}$$

where

$$SST = \sum (y_i - \bar{y})^2 = \text{total sum of squares}$$

$$SSM = \sum (\hat{y}_i - \bar{y})^2 = \text{model sum of squares}$$

$$SSE = \sum (y_i - \hat{y}_i)^2 = \text{residual sum of squares}$$

$$SST = SSM + SSE$$

$SSM$  is a measure of how much variation the model explains ( $SSM$  is big when the values of  $\hat{y}$  are quite different from the average  $\bar{y}$ ), and  $SSE$  is a measure of of unexplained variation since it measure how different the data value  $y_i$  is from the prediction of the model  $\hat{y}$ .

In our case, 71.4% of the variation in strength is explained by our model.

<sup>1</sup>If there are  $k$  predictors ( $k + 1$  parameters including  $\beta_0$ ), then using the least squares approach will lead to a system of  $k + 1$  equations in  $k + 1$  unknowns, so standard linear algebra can be used to find the estimates.

2. adjusted  $R^2 = 0.6682$

Every time we add another term to the model,  $R^2$  will go up. (In theory it could stay the same, but it can never go down.) This is because the larger model has more flexibility to fit the data. This adjusted measure takes into account the size (number of parameters) in the model and makes it easier to compare models with different numbers of parameters.

$$\text{adjusted } R^2 = 1 - \frac{n-1}{n-p} \frac{SSE}{SST}$$

where  $p$  is the number of parameters in the model and  $n$  is the number of observations in our sample. In our case, there are 5 parameters in the model (the coefficients for our four predictors plus the constant  $\beta_0$ ) and  $n = 30$ , so

$$\text{adjusted } R^2 = 1 - \frac{30-1}{30-5} (1 - R^2) = 1 - \frac{29}{25} (.286) = .6682$$

3. The **model utility test** tests the null hypothesis

$$H_0 : \beta_1 = \beta_2 = \beta_3 = \cdots = \beta_k = 0$$

verses the alternative that at least one of these  $\beta_i$ 's is not 0.

The test statistic used for this test is

$$F = \frac{SSM/k}{SSE/(n-k-1)} = \frac{MSM}{MSE} = \frac{R^2/k}{(1-R^2)/[n-k-1]}$$

This p-value also appears in our output. It is based on a distribution called the F-distribution. F-distributions have two parameters called the numerator degrees of freedom ( $k$ , in this case) and the denominator degrees of freedom ( $n - k - 1$ ). Notice these add to  $n - 1$ , the total degrees of freedom.

The small p-value indicates that this model is “better than nothing”, that is, it does a better job of predicting than just using the overall mean of the response. Another way to say this is that we are comparing our model

$$y = \beta_0 + \beta_1 \text{force} + \beta_2 \text{power} + \beta_3 \text{temp} + \beta_4 \text{time} + \varepsilon$$

to the very simple model

$$y = \beta_0 + \varepsilon$$

and our model does much better than would be expected just by random chance.

If you look back at our simple linear models, you will see that this model utility test is done there too. In that case it is testing

$$H_0 : \beta_1 = 0$$

So it is equivalent to the  $t$ -test testing the same hypothesis; in fact,  $t^2 = F$ .

## 10.5 Tests (and intervals) for individual parameters

There are 5 p-values listed in the Coefficients table. They correspond to tests with null hypothesis

$$H_0 : \beta_i = 0$$

and should be interpreted as testing whether the coefficient on one term is 0 *in a model where all the other terms are present*. From these it would appear that time and force are not as important to the model as power and temperature. We can also use the information in that part of the table to give confidence intervals for each parameter.

## 10.6 Comparing nested models

Two models are said to be **nested** if the smaller model (the one with fewer parameters) is obtained from the larger model by setting some of the parameters to 0. In our case, since force and time seemed to be less important, we could drop them from the model.

```
> wire.model2 <- lm(Strength ~ Power + Temperature, data=wire);
> summary(wire.model2);

Call:
lm(formula = Strength ~ Power + Temperature, data = wire)

Residuals:
    Min       1Q   Median       3Q      Max
-11.3233  -2.8067  -0.8483   3.1892   9.4600

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -24.90167    10.07207  -2.472  0.02001 *
Power         0.49833     0.07086   7.033 1.47e-07 ***
Temperature  0.12967     0.04251   3.050 0.00508 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 5.207 on 27 degrees of freedom
Multiple R-squared:  0.6852,    Adjusted R-squared:  0.6619
F-statistic: 29.38 on 2 and 27 DF,  p-value: 1.674e-07
```

We see that  $R^2$  decreases (it always will), but the adjusted  $R^2$  values are quite close.

There is a formal hypothesis test to test whether the smaller model will do:

- $H_0$ :  $\beta_i = 0$  for all  $i > l$  (the last  $k - l$  parameters in some ordering)
  - if  $H_0$  is true, then the “reduced” model is correct.
- $H_a$ :  $\beta_i \neq 0$  for at least one  $i > l$ .
  - if  $H_a$  is true, then the “full” model is better because at least some of the extra parameters are not 0.
- Test statistic:  $F = \frac{MSE_{\text{diff}}}{MSE_{\text{full}}}$

The idea of this test is to take the unexplained variation from the reduced model and split it into two pieces: the portion explained by the full model but not by the reduced model ( $SSE_{\text{diff}} = SSE_{\text{reduced}} - SSE_{\text{full}}$ ) and the portion unexplained even in the full model ( $SSE_{\text{full}}$ ). The degrees of freedom for the numerator is the difference in the number of parameters for the two models. The degrees of freedom for the denominator is the residual degrees of freedom for the full model.

```
> anova(wire.model2, wire.model);
Analysis of Variance Table

Model 1: Strength ~ Power + Temperature
Model 2: Strength ~ Force + Power + Temperature + Time
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      27 732.04
2      25 665.12  2     66.92 1.2577 0.3017
```

In this case the p-value is not small enough to reject the null hypothesis. Our data are consistent with the smaller model being just as good as the larger one.

When we have multiple predictors, there are often many possible models that we could fit. The use of the adjusted  $R^2$  and model comparison tests are two ways to help us select an appropriate model. Of course, we should also do our usual diagnostics and look at plots of residuals to check that the model assumptions (normality, constant variance, linearity) appear to be satisfied for the models we select.

### 10.6.1 Relationship to 1-parameter tests

If we compare two models, one our (larger) model of interest and the other a model with one parameter removed, we get another way to do our test for a single parameter. These tests are equivalent, and  $F = t^2$ .

```
> wire.model3 <- lm(Strength ~ Force + Power + Temperature, data=wire); # no Time
> anova(wire.model3,wire.model);
Analysis of Variance Table

Model 1: Strength ~ Force + Power + Temperature
Model 2: Strength ~ Force + Power + Temperature + Time
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      26 705.16
2      25 665.12  1    40.04 1.5051 0.2313
> summary(wire.model);

Call:
lm(formula = Strength ~ Force + Power + Temperature + Time, data = wire)

Residuals:
    Min       1Q   Median       3Q      Max
-11.0900  -1.7608  -0.3067   2.4392   7.5933

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -37.47667    13.09964  -2.861  0.00841 **
Force         0.21167     0.21057   1.005  0.32444
Power         0.49833     0.07019   7.100 1.93e-07 ***
Temperature   0.12967     0.04211   3.079  0.00499 **
Time          0.25833     0.21057   1.227  0.23132
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 5.158 on 25 degrees of freedom
Multiple R-squared:  0.714,    Adjusted R-squared:  0.6682
F-statistic: 15.6 on 4 and 25 DF,  p-value: 1.592e-06
```

## 10.7 Can we do even better?

Now let's consider a model that includes not only our four predictors (force, power, temperature, and time) but also the six interaction terms. So  $k = 10$  and  $p = 11$  for this model. R provides a shortcut for specifying this model:

```
> wire.model.interaction <- lm(Strength ~ (Force + Power + Temperature + Time)^2 , data=wire);
> summary(wire.model.interaction);

Call:
lm(formula = Strength ~ (Force + Power + Temperature + Time)^2,
    data = wire)
```

```

Residuals:
    Min       1Q   Median       3Q      Max
-10.607  -1.874  -0.440   2.406   7.593

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -18.426667 108.812510  -0.169  0.8673
Force        -1.238333   2.459616  -0.503  0.6204
Power         0.118333   0.908220   0.130  0.8977
Temperature   0.791167   0.455672   1.736  0.0987
Time         -4.299167   2.817633  -1.526  0.1435
Force:Power    0.024000   0.015952   1.505  0.1489
Force:Temperature -0.009300  0.009571  -0.972  0.3434
Force:Time     0.075500   0.047855   1.578  0.1311
Power:Temperature -0.004667  0.003190  -1.463  0.1599
Power:Time     0.023667   0.015952   1.484  0.1543
Temperature:Time 0.000700   0.009571   0.073  0.9425
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.786 on 19 degrees of freedom
Multiple R-squared:  0.8129,    Adjusted R-squared:  0.7144
F-statistic: 8.253 on 10 and 19 DF,  p-value: 4.932e-05

```

Note:

- $R^2$  has increased to 0.813.
- A fairer comparison is to look at adjusted  $R^2$ . The gains are modest here.
- None of the individual parameters has a significant p-value.

This doesn't mean that none of these predictors is important. It may just mean that there is some redundancy among the predictors and that none of them adds much predictive power *when all the others are in the model*.

Given all that, it seems wise to check if we gain anything by having interaction terms. We can do that with the following model comparison test.

```

> wire.model.interaction <- lm(Strength ~ (Force + Power + Temperature + Time)^2 , data=wire);
> wire.model <- lm(Strength ~ (Force + Power + Temperature + Time) , data=wire);
> anova(wire.model.interaction,wire.model);
Analysis of Variance Table

Model 1: Strength ~ (Force + Power + Temperature + Time)^2
Model 2: Strength ~ (Force + Power + Temperature + Time)
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     19  435.12
2     25  665.12 -6   -230.00 1.6738 0.1819

```

The p-value is not small enough to reject the null hypothesis (which says that none of the interaction terms is significantly different from 0). So there is no compelling reason to put interaction terms into this model.

In Devore&Farnum, they do a similar thing adding all the second order terms as well.

## 10.8 Another example: cement strength

In a different book by Devore, an experiment is described where the strength of cement is tested under different “recipes”. In particular, the amount of water and of limestone is varied.

```
> require(Devore6)
> data(xmp13.13); xmp13.13 -> cement
> names(cement)[1] <- 'limestone'
> names(cement)[2] <- 'water'
> model1 <-
+   lm(strength~limestone+water,cement);
> model2 <-
+   lm(strength~limestone*water,cement);
> model3 <-
+   lm(strength~limestone + limestone:water,cement);
> model4 <-
+   lm(strength~limestone + water + I(water^2),cement);

> summary(model1);
Call:
lm(formula = strength ~ limestone + water, data = cement)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  84.8167    12.2415   6.929 0.000448 ***
limestone     0.1643     0.1431   1.148 0.294673
water       -79.6667    20.0349  -3.976 0.007313 **

Residual standard error: 3.47 on 6 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.6541
F-statistic: 8.565 on 2 and 6 DF,  p-value: 0.01746

> summary(model2);
Call:
lm(formula = strength ~ limestone * water, data = cement)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)     6.217    30.304   0.205  0.8455
limestone       5.779     2.079   2.779  0.0389 *
water           41.2667    408.2325  -0.101  0.923
I(water^2)     -32.0000    339.7026  -0.094  0.929

Residual standard error: 3.798 on 5 degrees of freedom
Multiple R-squared:  0.7411,    Adjusted R-squared:  0.5857
F-statistic:  4.77 on 3 and 5 DF,  p-value: 0.06272

> summary(model3);
Call:
lm(formula = strength ~ limestone + limestone:water, data = cement)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.0167    1.6200  22.850 4.6e-07 ***
limestone     3.7478    0.5863   6.392 0.00069 ***
limestone:water -5.9725    0.9628  -6.203 0.00081 ***

Residual standard error: 2.423 on 5 degrees of freedom
Multiple R-squared:  0.8946,    Adjusted R-squared:  0.8314
F-statistic: 14.15 on 3 and 5 DF,  p-value: 0.00706

> summary(model4);
Call:
lm(formula = strength ~ limestone + water + I(water^2), data = cement)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  73.4033    121.8991   0.602  0.573
limestone     0.1643     0.1566   1.049  0.342
water       -41.2667    408.2325  -0.101  0.923
I(water^2)     -32.0000    339.7026  -0.094  0.929

Residual standard error: 2.43 on 6 degrees of freedom
Multiple R-squared:  0.8728,    Adjusted R-squared:  0.8304
F-statistic: 20.58 on 2 and 6 DF,  p-value: 0.002058
```

The interesting thing to note here is that in the original model it looks like water is a good predictor and limestone is not, but when we add an interaction term, limestone is a good predictor and water is no longer needed. *Whether a predictor is useful can depend on what other predictors are in the model.*

## 10.9 Diagnostics

The same sorts of diagnostic checks can be done for multiple regression as for simple linear regression. Most of these involve looking at the residuals. As before `plot(lm(...))` will make several types of residual plots.

Consideration of the normality and homoscedasticity (equal variance) assumptions of linear models should also be a part of selecting a model.

## 10.10 The final choice

So how do we choose a model? There are no hard and fast rules, but here are some things that play a role:

- A priori theory.

Some models are chosen by there is some scientific theory that predicts a relationship of a certain form. Statistics is used to find the most likely parameters in a model of this form. If there are competing theories, we can fit multiple models and see which seems to fit better.

- Previous experience.

Models that have worked well in other similar situations may work well again.

- The data.

Especially in new situations, we may only have the data to go on. Regression diagnostics, adjusted  $r^2$ , various hypothesis tests, and other methods like the commonly used information criteria AIC and BIC can help us choose between models. *In general, it is good to choose the simplest model that works well.*

There are a number of methods that have been proposed to automate the proces of searching through many models to find the “best” one. One commonly used one is called stepwise regression. Stepwise regression works by repeatedly dropping or adding a single term from the model until there are no such single parameter changes that improve the model (based on some criterion; AIC is the default in R.) The function `step()` will do this in R.

If the number of parameters is small enough, one could try all possible subsets of the parameters. This could find a “better” model than the one found by stepwise regression.

### 10.10.1 AIC: Aikeke's Information Criterion

$$AIC = 2k + n \ln(RSS/n)$$

where  $k$  is the number of parameters and RSS is the residual sum of squares (SSE). Smaller is better. There are theoretical reasons for this particular formula (base don likelihood methods), but notice that the first addend increases with each parameter in the model and the second decreases; so this forms a kind of balance between the expected gains for adding new parameters against the costs of complication and potential for over-fitting. The scale of AIC is only meaningful relative to a fixed data set.

There are several other criteria that have been proposed.

## 10.11 More regression examples

**Example.** Where regression got its name.

In the 1890's Karl Pearson gathered data on over 1100 British families. Among other things he measured the heights of parents and children. The analysis below comes from his data is for heights of mothers and daughters. Can you see why this data (and analysis) led him to coin the phrase “regression toward the mean”?

<pre>&gt; require(alr3) &gt; data(heights) &gt; xyplot(Dheight~Mheight,heights) &gt; summary(lm(Dheight~Mheight,heights)) \columnbreak Coefficients:       Estimate Std. Error t value Pr(&gt; t )</pre>	<pre>(Intercept) 29.91744    1.62247    18.44 &lt;2e-16 *** Mheight      0.54175    0.02596    20.87 &lt;2e-16 ***  Residual standard error: 2.266 on 1373 degrees of freedom Multiple R-Squared:  0.2408,    Adjusted R-squared:  0.2402 F-statistic: 435.5 on 1 and 1373 DF,  p-value: &lt; 2.2e-16</pre>
--	---

**Example.** Rats were given a dose of a drug proportional to their body weight. The rats were then slaughtered and the amount of drug in the liver and the weight of the liver were measured.

```

> require(alr3); data(rat)
> summary(lm(y~BodyWt*LiverWt, rat))

Call:
lm(formula = y ~ BodyWt * LiverWt, data = rat)

Residuals:
    Min       1Q   Median       3Q      Max
-0.133986 -0.043370 -0.007227  0.036389  0.184029

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.573822   1.468416   1.072   0.301
BodyWt        -0.007759   0.008568  -0.906   0.379
LiverWt       -0.177329   0.198202  -0.895   0.385
BodyWt:LiverWt 0.001095   0.001138   0.962   0.351

Residual standard error: 0.09193 on 15 degrees of freedom
Multiple R-Squared:  0.1001,    Adjusted R-squared:  -0.07985
F-statistic: 0.5563 on 3 and 15 DF,  p-value: 0.6519

> plot(lm(y~BodyWt*LiverWt, rat))

```

None of the individual parameters looks significantly non-zero. What is the interpretation?

## 10.12 Categorical predictors

We can even handle categorical predictors! We do this by introducing **dummy variables** (less pejoratively called **indicator variables**). If a variable  $v$  has only two possible values –  $A$  and  $B$  – we can build an indicator variable for  $v$  as follows:

$$x = \begin{cases} 1 & v = B \\ 0 & v \neq B \end{cases}$$

That is, we simply code the possibilities as 0 and 1.

If we have more than two possible values (levels), we introduce multiple dummy variables<sup>2</sup> (one less than the number of levels). For a variable with three levels ( $A$ ,  $B$  and  $C$ ), one standard encoding (and the one that R uses by default) is

$$x_1 = \begin{cases} 1 & v = B \\ 0 & v \neq B \end{cases} \quad x_2 = \begin{cases} 1 & v = C \\ 0 & v \neq C \end{cases}$$

We don't need to have 3 dummy variables, since the intercept term captures the effect of one of the levels. R conveniently takes care of the recoding for us.

<sup>2</sup>There are models that do other things. Coding with a single variable with values 0, 1, 2 is possible, for example, but is a very different model. In this alternative there is an implied order among the groups and the effect of moving from category 0 to category 1 is the same size as the effect of moving from category 1 to category 2. This model should only be used if these assumptions make sense.

**Example.** Home field advantage?

A professor from the University from Minnesota ran tests to see if adjusting the air conditioning in the Metrodome could affect the distance a batted ball travels. Notice that Cond is categorical and indicates whether there was a (artificial) headwind or tailwind.

```
> require(alr3); data(domedata)
> summary(domedata)
```

Cond	Velocity	Angle
Head:19	Min. :149.3	Min. :48.30
Tail:15	1st Qu.:154.1	1st Qu.:49.50
	Median :155.5	Median :50.00
	Mean :155.2	Mean :49.98
	3rd Qu.:156.3	3rd Qu.:50.60
	Max. :160.9	Max. :51.00

```
\dots
> lm.dome01 <- lm(
  Dist~Velocity+Angle+BallWt+BallDia+Cond,
  data=domedata)
```

```
> summary(lm.dome01)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 181.7443    335.6959   0.541  0.59252
Velocity      1.7284     0.5433    3.181  0.00357 **
Angle        -1.6014     1.7995   -0.890  0.38110
BallWt       -3.9862     2.6697   -1.493  0.14659
BallDia     190.3715    62.5115   3.045  0.00502 **
CondTail      7.6705     2.4593    3.119  0.00418 **

Residual standard error: 6.805 on 28 degrees of freedom
Multiple R-Squared: 0.5917,      Adjusted R-squared: 0.5188
F-statistic: 8.115 on 5 and 28 DF,  p-value: 7.81e-05
```

It looks like the angle and ball weight don't matter much. Velocity and direction of wind do (this makes sense). So does ball diameter; that's a bit unfortunate. Let's do some model comparisons.

```
> step(lm.dome01,direction="both")
Start:  AIC= 135.8
  Dist ~ Velocity + Angle + BallWt + BallDia + Cond
```

	Df	Sum of Sq	RSS	AIC
- Angle	1	36.67	1333.24	134.75
<none>			1296.57	135.80
- BallWt	1	103.24	1399.81	136.40
- BallDia	1	429.46	1726.03	143.53
- Cond	1	450.46	1747.03	143.94
- Velocity	1	468.68	1765.26	144.29

```
Step:  AIC= 134.75
  Dist ~ Velocity + BallWt + BallDia + Cond
```

	Df	Sum of Sq	RSS	AIC
<none>			1333.24	134.75
- BallWt	1	111.05	1444.30	135.47
+ Angle	1	36.67	1296.57	135.80
- BallDia	1	408.92	1742.16	141.84
- Velocity	1	481.14	1814.38	143.22
- Cond	1	499.48	1832.72	143.56

```
Call:
lm(formula = Dist ~ Velocity + BallWt + BallDia + Cond,
    data = domedata)

Coefficients:
(Intercept)      Velocity      BallWt      BallDia
  133.824         1.750        -4.127        184.842
CondTail
   7.990
```

```
> lm.dome02 <- lm(Dist~Velocity+Cond,data=domedata)
> summary(lm.dome02)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -27.9887     83.9460  -0.333  0.7411
Velocity      2.4388     0.5404    4.513 8.63e-05 ***
CondTail      6.5118     2.5935    2.511  0.0175 *

Residual standard error: 7.497 on 31 degrees of freedom
Multiple R-Squared: 0.4513,      Adjusted R-squared: 0.4159
F-statistic: 12.75 on 2 and 31 DF,  p-value: 9.118e-05

> anova(lm.dome02,lm.dome01)
Analysis of Variance Table

Model 1: Dist ~ Velocity + Cond
Model 2: Dist ~ Velocity + Angle + BallWt + BallDia + Cond
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     31 1742.45
2     28 1296.57  3    445.88 3.2096 0.03812 *
```

```
> lm.dome03 <- lm(Dist~Velocity+Cond+BallDia,data=domedata)
> anova(lm.dome02,lm.dome03)
Analysis of Variance Table

Model 1: Dist ~ Velocity + Cond
Model 2: Dist ~ Velocity + Cond + BallDia
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     31 1742.45
2     30 1444.30  1    298.15 6.193 0.01860 *
```