

1 Basics

This is a basic Sweave document. We can do arithmetic.

```
> 1 + 1
```

```
[1] 2
```

We can save a value in a variable and output it in text.

```
> a <- 2 * pi
```

```
> b <- 4
```

The numeric value of 2π , rounded to 4 digits, is 6.2832.

By default, both the input and output of R commands are shown. Either or both can be suppressed, as will be demonstrated in future sections.

2 Tables

We can fit a linear model on the Chatterjee-Price Attitude Data. The R help file describes the data as follows.

From a survey of the clerical employees of a large financial organization, the data are aggregated from the questionnaires of the approximately 35 employees for each of 30 (randomly selected) departments. The numbers give the percent proportion of favourable responses to seven questions in each department.

This data file has 30 rows and 7 columns. We can use the `xtable` library to output the first few rows of the data. In the source file for this document, notice that for the chunk loading this library, both input and output are hidden (using `echo=FALSE` and `results=hide`, respectively). And in the chunk that creates the table, we tell Sweave that the results are already LaTeX code using `results=tex`.

	rating	complaints	privileges	learning	raises	critical	advance
1	43.00	51.00	30.00	39.00	61.00	92.00	45.00
2	63.00	64.00	51.00	54.00	63.00	73.00	47.00
3	71.00	70.00	68.00	69.00	76.00	86.00	48.00
4	61.00	63.00	45.00	47.00	54.00	84.00	35.00
5	81.00	78.00	56.00	66.00	71.00	83.00	47.00
6	43.00	55.00	49.00	44.00	54.00	49.00	34.00

We can fit a basic linear model and output a summary.

```
> summary(fm1 <- lm(rating ~ ., data = attitude))
```

```
Call:
lm(formula = rating ~ ., data = attitude)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-10.9418  -4.3555   0.3158   5.5425  11.5990
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.78708   11.58926   0.931 0.361634
complaints   0.61319    0.16098   3.809 0.000903 ***
privileges  -0.07305    0.13572  -0.538 0.595594
learning     0.32033    0.16852   1.901 0.069925 .
raises       0.08173    0.22148   0.369 0.715480
critical     0.03838    0.14700   0.261 0.796334
advance     -0.21706    0.17821  -1.218 0.235577
---
```

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

```
Residual standard error: 7.068 on 23 degrees of freedom
Multiple R-squared:  0.7326,    Adjusted R-squared:  0.6628
F-statistic: 10.5 on 6 and 23 DF,  p-value: 1.240e-05
```

Or we can output the table of coefficients automatically in LaTeX format using `xtable`.

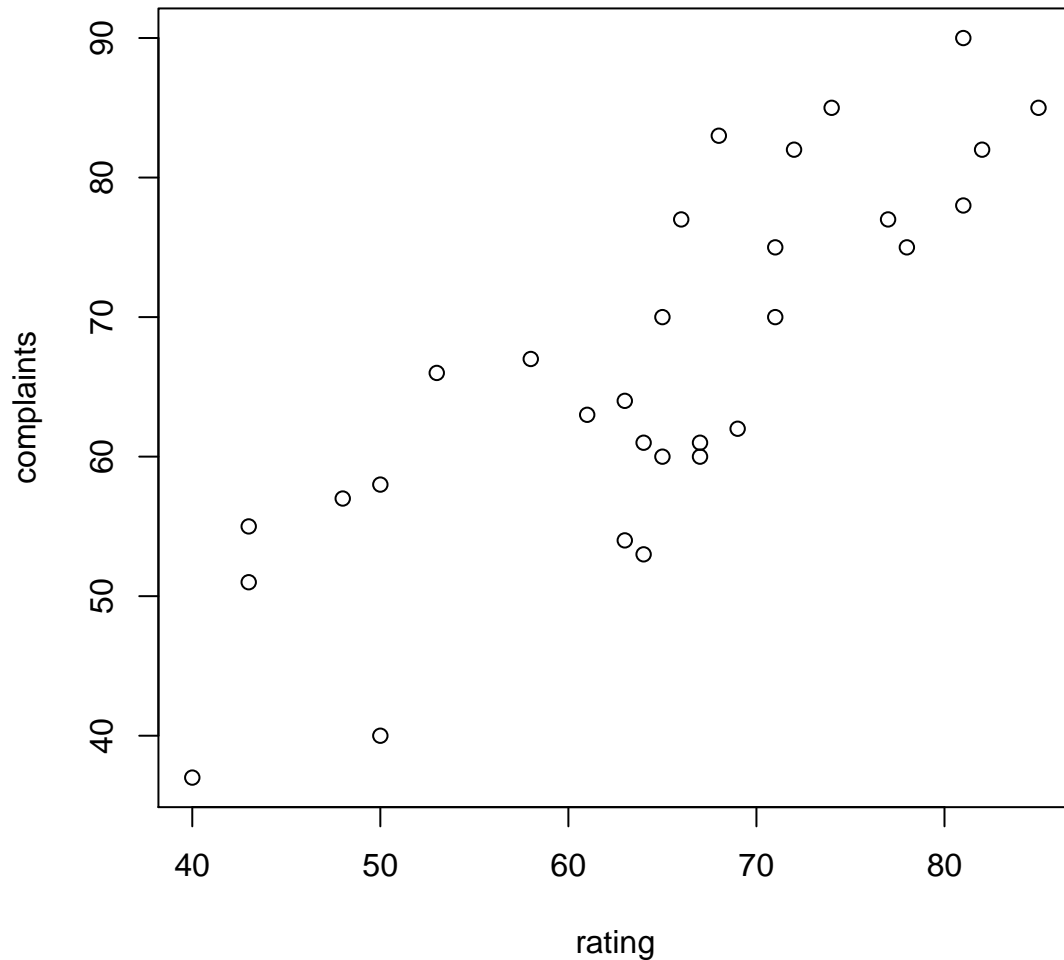
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.7871	11.5893	0.93	0.3616
complaints	0.6132	0.1610	3.81	0.0009
privileges	-0.0731	0.1357	-0.54	0.5956
learning	0.3203	0.1685	1.90	0.0699
raises	0.0817	0.2215	0.37	0.7155
critical	0.0384	0.1470	0.26	0.7963
advance	-0.2171	0.1782	-1.22	0.2356

3 Plots

This is one of the places I think Sweave really shines. No more trying to keep separately keep track of the both the code that created a figure and the figure itself! When you have many figures, that gets hairy fast.

You can make a simple picture just by putting plotting code in an Sweave chunk.

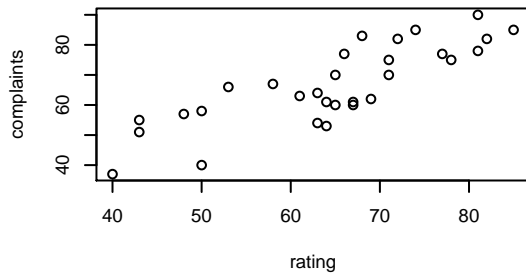
```
> with(attitude, plot(rating, complaints))
```



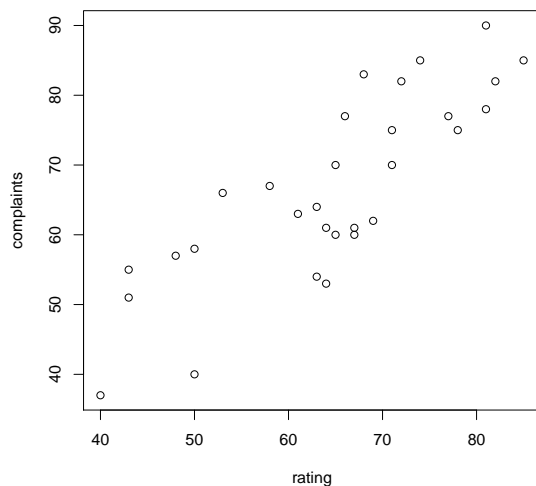
3.1 Scaling graphics

I prefer to scale pictures by having R output them in a different size, this way the text on all the pictures is always the appropriate height. To do this, use the `nogin` document option (at the beginning of your Sweave file); this tells Sweave not to scale all figures to be 80% of the width of the page, which is otherwise the default. Then use `height=` and `width=` in calling the Sweave chunk to change the height and width. Here's the same plot but with height 2 and width 3. I've also used the `cex` parameter option in R to change the text and point size.

```
> par(cex = 0.6)
> with(attitude, plot(rating, complaints))
> par(cex = 1)
```



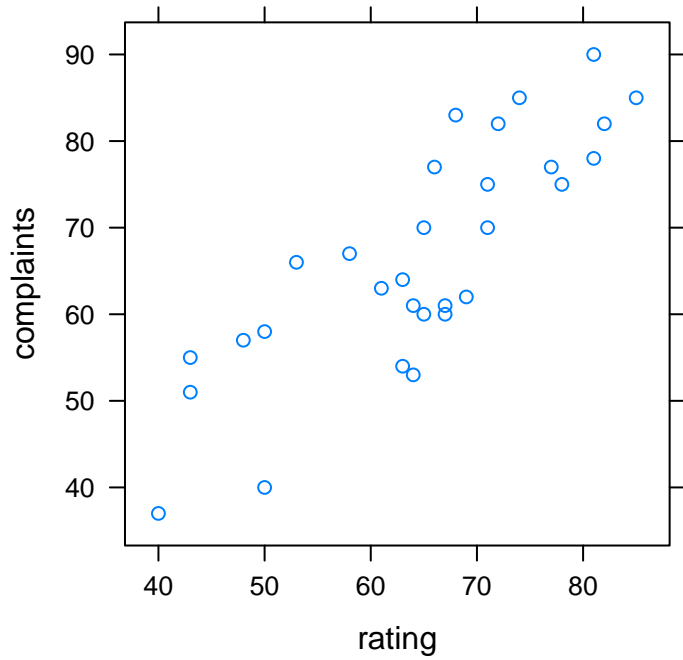
An alternative way to have Sweave create the figure but not include it; you then use `includegraphics` as usual to include the figure. To do this easily, you'll need to name the chunk; this chunk I named `myfig`; look in the source to see how. The name of the created file to include is the name of the Sweave document file (here `myfile`), followed by a hyphen and then the name of the chunk, so here it's `myfile-myfig`. In this case, I've created the first figure again and scaled it by 50%.



3.2 Lattice

A note about using the `lattice` library; because of the way that library creates figures, you must wrap the plot command (or its output) in an additional plot command to have success, as follows.

```
> library(lattice)
> myplot <- xyplot(complaints ~ rating, data = attitude)
> plot(myplot)
```



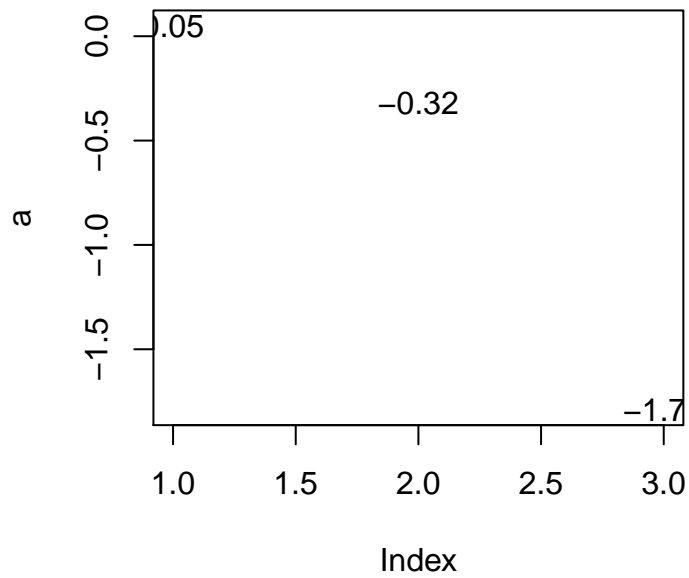
3.3 Randomness in plots

And, a note about using random numbers in plots; Sweave *re-executes* code chunks to create figures, so if you're not careful, you can get behaviour like this:

```
> a <- round(rnorm(3), 2)
> a
```

```
[1] 0.04 0.18 1.32
```

```
> plot(a, type = "n")
> text(1:3, a, a)
```



The values in the plot are different from the values in the output!

Instead, put the random number generation and the plotting in two different chunks, like this:

First chunk:

```
> a <- round(rnorm(3), 2)
```

```
> a
```

```
[1] 0.22 -0.15 1.05
```

Second chunk:

```
> plot(a, type = "n")
```

```
> text(1:3, a, a)
```

