1 Syntax

<table>
<thead>
<tr>
<th>format</th>
<th>start</th>
<th>end</th>
<th>inline</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnw</td>
<td>&lt;&lt;*&gt;&gt;=</td>
<td>@</td>
<td>$expr{x}</td>
<td>TeX</td>
</tr>
<tr>
<td>Rmd</td>
<td><code>(r *)</code></td>
<td>```</td>
<td><code>r x</code></td>
<td>MD</td>
</tr>
<tr>
<td>Rhtml</td>
<td><code>{r --begin.rcode * end.rcode--}</code></td>
<td><code>{r --rinline x--}</code></td>
<td>HTML</td>
<td></td>
</tr>
<tr>
<td>Rst</td>
<td><code>.. (r *)</code></td>
<td><code>..</code></td>
<td><code>r:x</code></td>
<td>reST</td>
</tr>
<tr>
<td>Rtex</td>
<td><code>% begin.rcode * % end.rcode</code></td>
<td><code>%rinline(x)</code></td>
<td>TeX</td>
<td></td>
</tr>
<tr>
<td>Rasciidoc</td>
<td><code>// begin.rcode * // end.rcode</code></td>
<td><code>+r x+</code></td>
<td>AsciiDoc</td>
<td></td>
</tr>
<tr>
<td>Rtextile</td>
<td>### begin.rcode * ### end.rcode</td>
<td>#r x@</td>
<td>Textile</td>
<td></td>
</tr>
<tr>
<td>brew</td>
<td>% begin.rcode * % end.rcode</td>
<td>%rinline(x)</td>
<td>TeX</td>
<td></td>
</tr>
</tbody>
</table>

* denotes local chunk options, e.g. `<label, eval=FALSE>=`; x denotes inline R code, e.g. `r 1+2` (MD stands for [Markdown](https://en.wikipedia.org/wiki/Markdown)).

2 Minimal Examples

2.1 Sweave (*.Rnw)

```Rnw
\documentclass{article}
\begin{document}

Below is a code chunk.  
```
<<foo, echo=TRUE>>=

z = 1+1
plot(cars)
```

The value of z is \$z\$.

\end{document}
```

2.2 R Markdown (*.Rmd)

```
Hi _markdown_!

```r
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
`