Contents

Chapter 1

Introduction

Description of the work. Prepare the reader for the following chapters.

You will cite literature here, typically, but

Chapter 2

First Chapter

2.1 To include a picture

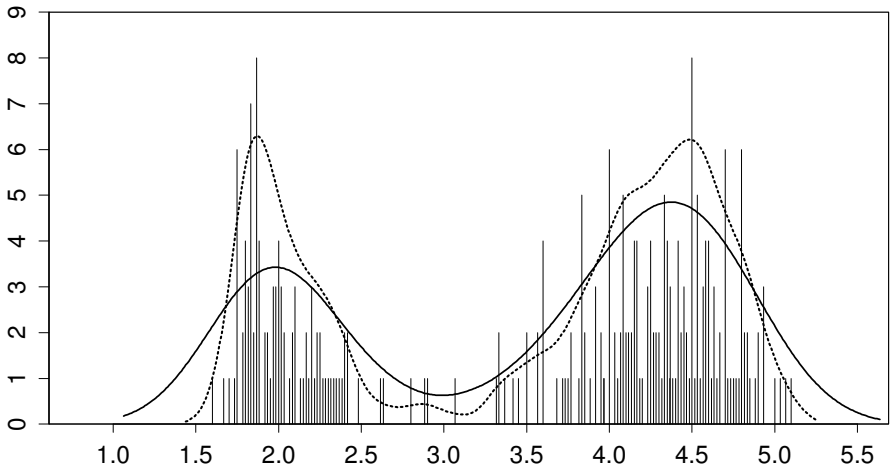

Figure 2.1: Old Faithful Geyser eruption lengths, $n = 272$; binned data and two (Gaussian) kernel density estimates ($\times 10$) with $h = h^* = .3348$ and $h = .1$ (dotted).

Or also with `includegraphics`:

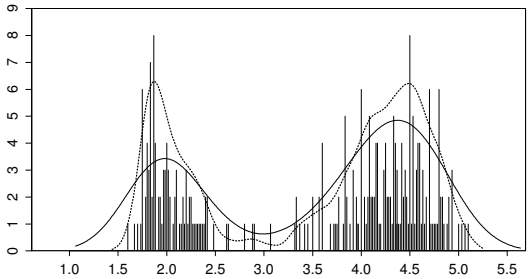

Figure 2.2: Old Faithful Geyser eruption lengths, $n = 272$; binned data and two (Gaussian) kernel density estimates ($\times 10$) with $h = h^* = .3348$ and $h = .1$ (dotted).

2.2 To make a proof

Proof. $1 + 1 = 2$ □

2.3 To include R code

See information in Appendix A.

2.4 Other information

Put a text between quotes: make sure to use nice quotes, such as “quote”.

Cite an article or book you refer shortly here, and then listed in the bibliography: **?**. Or mention that **?** (a person) or **?** (two persons) have already done quite a bit work.

Referencing a different part of your work: please refer to Appendix A.

Chapter 3

Summary

Summarize the presented work. Why is it useful to the research field or institute?

3.1 Future Work

Possible ways to extend the work.

Appendix A

Complementary information

Additional material. For example long mathematical derivations could be given in the appendix. Or you could include part of your code that is needed in printed form. You can add several Appendices to your thesis (as you can include several chapters in the main part of your work).

A.1 Including R code with verbatim

A simple (rather too simple, see A.2) way to include code or *R* output is to use `verbatim`. It just prints the text however it is (including all spaces, "strange" symbols,...) in a slightly different font.

```
## loading packages
library(RBGL)
library(Rgraphviz)
library(boot)

## global variables
X_MAX <- 150

   This allows me to put as many s  p a   c es   as I want.
I can also use \ and ‘ and & and all the rest that is usually only
accepted in the math mode.

I can also make as
                  many
             line
    breaks as
I want... and
             where I want.
```

But really recommended, much better is the following:

A.2 Including R code with the *listings* package

However, it is much nicer to use the *listings* package to include R code in your report. It allows you to number the lines, color the comments differently than the code, and so on. All the following is produced by simply writing `\lstinputlisting{Pictures/picture.R}` in your LaTeX "code":

```
## Example to generate an .pdf file with the function pdf.latex()
## Author: Sarah Gerster and Martin Maechler (UTF-8 Umlaute seem to fail here !?)
## Last revision: 16 Aug 2011

require("sfsmisc") # pdf.latex(), pdf.end(), etc

pdf.latex(file='test_plot.pdf') #, main=TRUE)
## no main=TRUE is needed to leave enough space for the plot title
## but see below

## make sure the legends are large enough
par(cex=1.5)

## Make sure your lines are "visible" enough. Otherwise your plot
## won't look very nicely in your text.
plot(-10:10, (-10:10)**2, type="l", lty=5,
     xlab="my_x", ylab="my_y",
     ## no main title: NOT recommended for figures in text which
     ## have a \caption{..}
     lwd=4, col='blue')
lines(-10:10, 0:20, type="p", lwd=4, pch=23,col='red')
legend(-3, 90, c("func1","func2"),lwd=4,col=c('blue', 'red'),
       lty=c(1,1),cex=1)
pdf.end() # starts the previewer (which refreshes itself;
          # at least on Linux at SfS
```

or `\lstinputlisting{/u/maechler/R/Pkgs/sfsmisc/R/misc/ellipse.R}` :

```
ellipsePoints ← function(a,b, alpha = 0, loc = c(0,0), n = 201,
                         keep.ab.order = FALSE)
{
    ## Purpose: ellipse points,radially equispaced, given geometric par.s
    ## ----------------------------------------------------------------------
    ## Arguments: a, b : length of half axes in (x,y) direction
    ##            alpha: angle (in degrees) for rotation
    ##            loc  : center of ellipse
    ##            n    : number of points
    ## ----------------------------------------------------------------------
    ## Author: Martin Maechler, Date: 19 Mar 2002

    stopifnot(is.numeric(a), is.numeric(b))
    reorder ← a < b && keep.ab.order
    B ← min(a,b)
    A ← max(a,b)
    ## B <= A
    d2 ← (A-B)*(A+B) ## = A^2 - B^2
    phi ← 2*pi*seq(0,1, len = n)
    sp ← sin(phi)
    cp ← cos(phi)
    r ← a*b / sqrt(B^2 + d2 * sp^2)
    xy ← r * if(reorder) cbind(sp, cp) else cbind(cp, sp)
    ## xy are the ellipse points for alpha = 0 and loc = (0,0)
    al ← alpha * pi/180
    ca ← cos(al)
    sa ← sin(al)
    xy %*% rbind(c(ca, sa), c(-sa, ca)) + cbind(rep(loc[1],n),
                                                rep(loc[2],n))
}
```

A.3 Using Sweave (or knitr) to include R code (and more) in your report

The easiest (and most elegant) way to include R code and its output (and have all your figures up to date with your report) is to use Sweave—or the knitr R package with even more possibilities.

Search the web to find lots of intro material on how to use Sweave or knitr (on Wikipedia).

Appendix B

Yet another appendix....

B.1 Description

Something details.

Something else other definition.

B.2 Tables

Refer to Table B.1 to see a left justified table with caption on top.

Table B.1: Results.

Student	**Grade**
Marie	6
Alain	5.5
Josette	4.5
Pierre	5

Appendix C

2nd Appendix: More sophisticated R code listing

Chapter-wise listing of parts of R code, using

- `firstline=n1`
- `lastline=n2`
- `title=<text>`

e.g., for the first example below

```
\lstinputlisting[firstline=1,lastline=32,
                 title= \texttt{read\_irwls\_fn.R}]{../RCode/read_irwls_fn.R}
```

C.1 Chapter 5

```
ellipsePoints ← function(a,b, alpha = 0, loc = c(0,0), n = 201,
                         keep.ab.order = FALSE)
{
    ## Purpose: ellipse points,radially equispaced, given geometric par.s
    ## ----------------------------------------------------------------------
    ## Arguments: a, b : length of half axes in (x,y) direction
    ##            alpha: angle (in degrees) for rotation
    ##            loc  : center of ellipse
    ##            n    : number of points
    ## ----------------------------------------------------------------------
    ## Author: Martin Maechler, Date: 19 Mar 2002

    stopifnot(is.numeric(a), is.numeric(b))
    reorder ← a < b && keep.ab.order
    B ← min(a,b)
    A ← max(a,b)
    ## B <= A
    d2 ← (A-B)*(A+B) ## = A^2 - B^2
    phi ← 2*pi*seq(0,1, len = n)
    sp ← sin(phi)
    cp ← cos(phi)
    r ← a*b / sqrt(B^2 + d2 * sp^2)
    xy ← r * if(reorder) cbind(sp, cp) else cbind(cp, sp)
    ## xy are the ellipse points for alpha = 0 and loc = (0,0)
    al ← alpha * pi/180
    ca ← cos(al)
    sa ← sin(al)
```

```
    xy %*% rbind(c(ca, sa), c(-sa, ca)) + cbind(rep(loc[1],n),
                                                rep(loc[2],n))
}
```

read_irwls_fn.R

```
ellipsePoints ← function(a,b, alpha = 0, loc = c(0,0), n = 201,
                         keep.ab.order = FALSE)
{
    ## Purpose: ellipse points,radially equispaced, given geometric par.s
    ## ----------------------------------------------------------------------
    ## Arguments: a, b : length of half axes in (x,y) direction
    ##            alpha: angle (in degrees) for rotation
    ##            loc  : center of ellipse
    ##            n    : number of points
    ## ----------------------------------------------------------------------
    ## Author: Martin Maechler, Date: 19 Mar 2002

    stopifnot(is.numeric(a), is.numeric(b))
    reorder ← a < b && keep.ab.order
    B ← min(a,b)
    A ← max(a,b)
    ## B <= A
    d2 ← (A-B)*(A+B) ## = A^2 - B^2
    phi ← 2*pi*seq(0,1, len = n)
    sp ← sin(phi)
    cp ← cos(phi)
    r ← a*b / sqrt(B^2 + d2 * sp^2)
    xy ← r * if(reorder) cbind(sp, cp) else cbind(cp, sp)
    ## xy are the ellipse points for alpha = 0 and loc = (0,0)
    al ← alpha * pi/180
    ca ← cos(al)
    sa ← sin(al)
    xy %*% rbind(c(ca, sa), c(-sa, ca)) + cbind(rep(loc[1],n),
                                                rep(loc[2],n))
}
```

plot.psi.R