SIMPLE PLOTS IN R

PLOTTING WITH R

When we analyze data, the first thing we should do is **look** at it: for each variable,

- what are the most common values?
- how much variability is present?
- are there any unusual observations?

Producing graphics for data analysis is relatively simple. Producing graphics for publication is **relatively more complex** and requires a great deal of tweaking to achieve the desired appearance.

`R` provides a number of functions for visualizing data; the table on the next page summarizes a few important plot types.

[E. Ghasshim, P. Boily, *Basics of R for Data Analysis*, Data Science Report Series, DAL Blog]

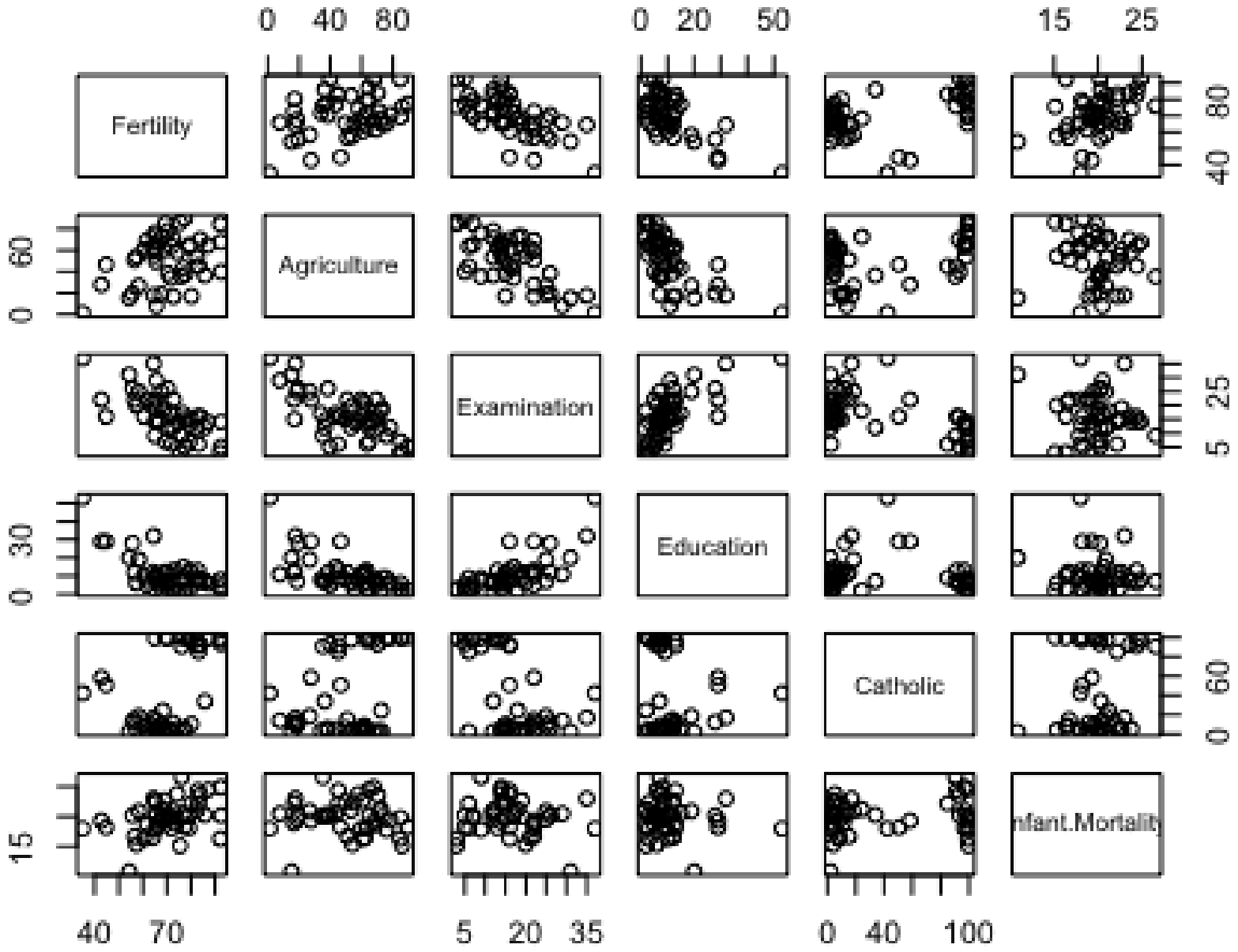
BASE R PLOTTING FUNCTIONALITY

Function	Graph type
`plot()`	Scatter plots and various others
`barplot()`	Bar plot (including stacked and grouped bar plots)
`hist()`	Histograms and (relative) frequency diagrams
`curve()`	Curves of mathematical expressions
`pie()`	Pie charts (for less scientific uses)
`boxplot()`	Box-and-whisker plots

EXAMPLE: SWISS DATASET

We start with a built-in R dataset called `swiss`.

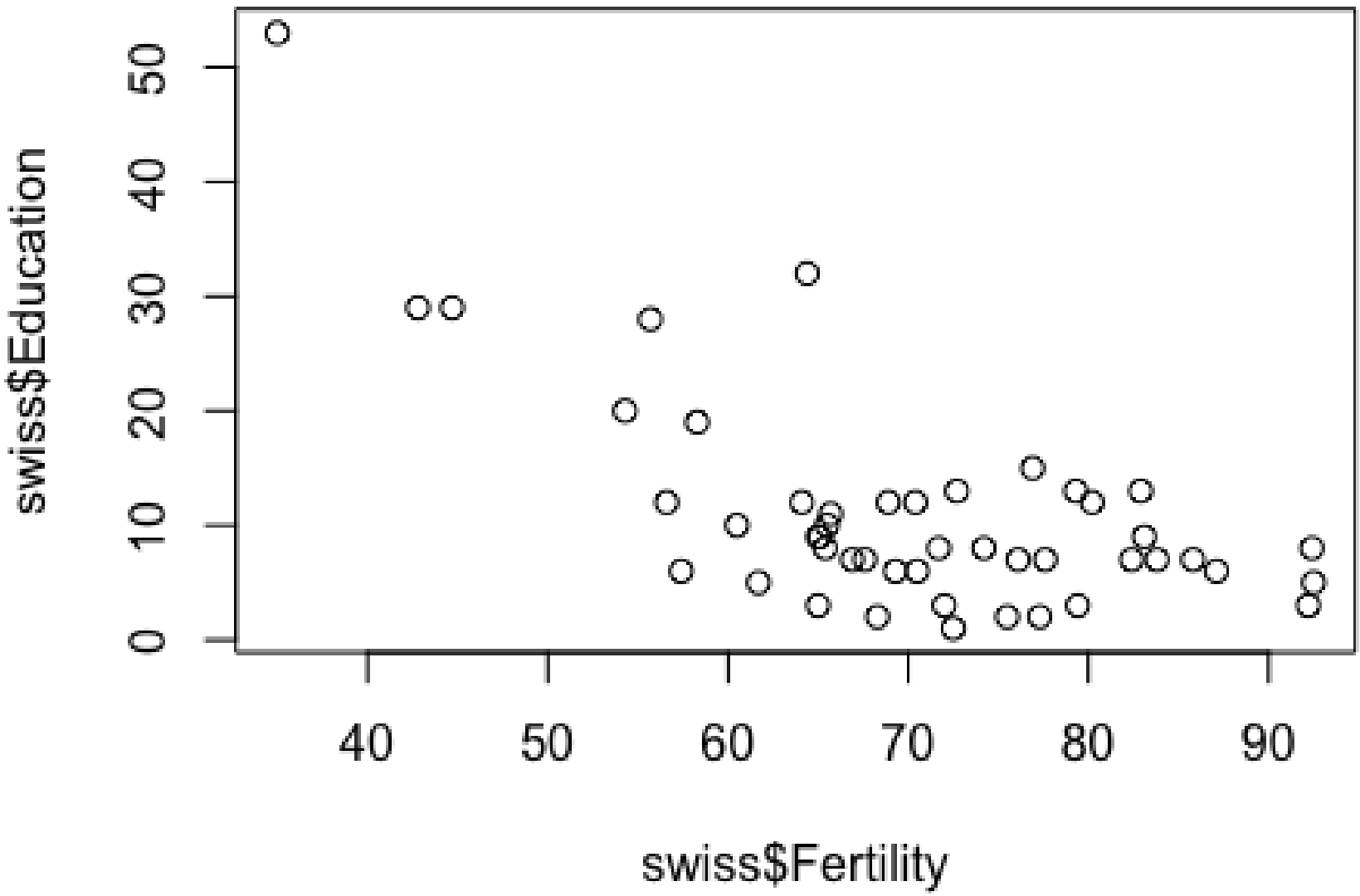
```
> str(swiss) # structure of the swiss dataset
## 'data.frame':    47 obs. of  6 variables:
##  $ Fertility       : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9
...
##  $ Agriculture     : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...
##  $ Examination     : int  15 6 5 12 17 9 16 14 12 16 ...
##  $ Education       : int  12 9 5 7 15 7 7 8 7 13 ...
##  $ Catholic        : num  9.96 84.84 93.4 33.77 5.16 ...
##  $ Infant.Mortality: num  22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...

> pairs(swiss) # scatter plot matrix for the swiss dataset
```


Fertility
Agriculture
Examination
Education
Catholic
Infant.Mortality
0 40 80
0 20 50
15 25
0 60
5 25
0 30
0 60
15
40 70
5 20 35
0 40 100

EXAMPLE: SWISS DATASET

Let's focus on one specific pair: `Fertility` vs. `Education`

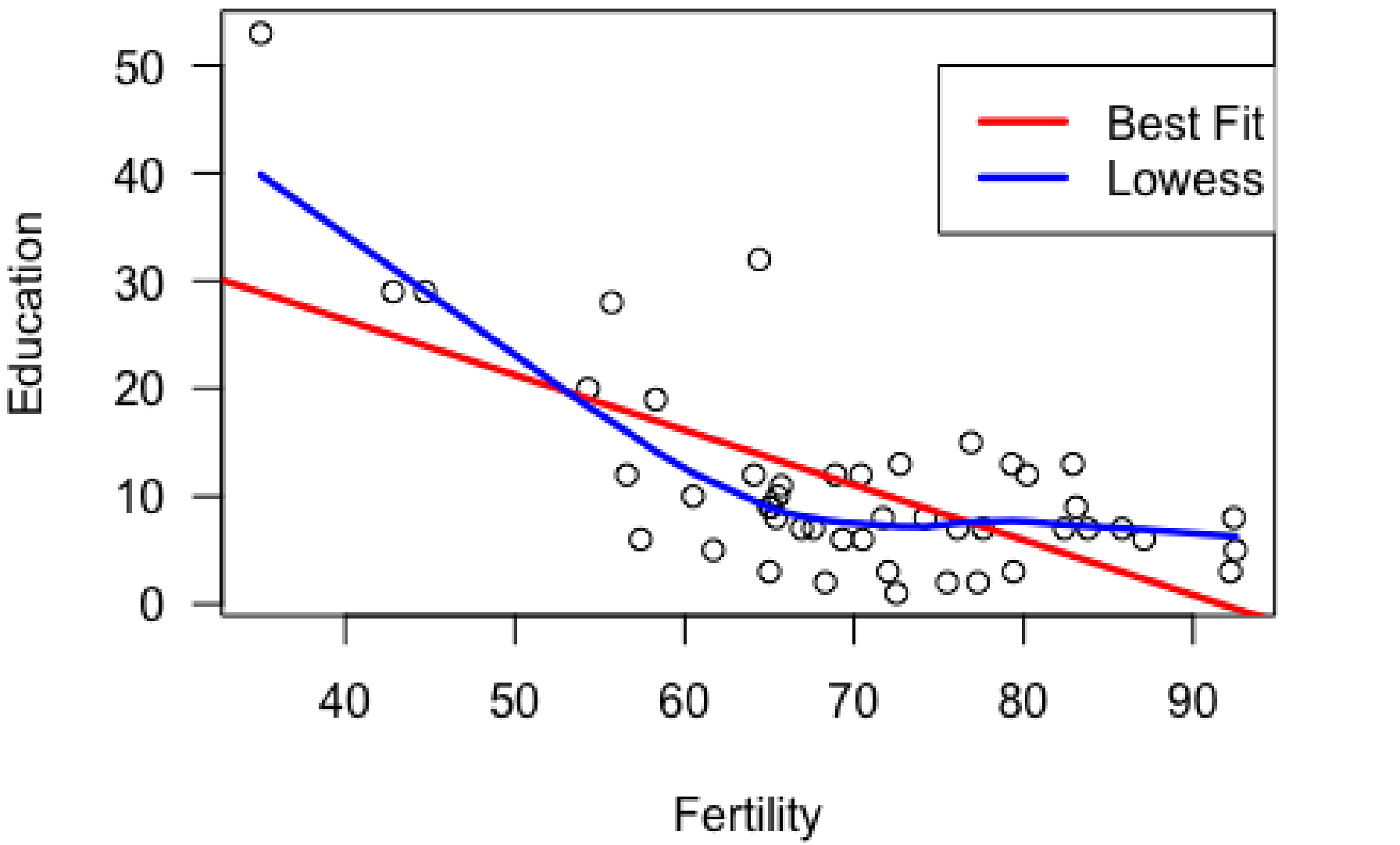
```
# raw plot
> plot(swiss$Fertility, swiss$Education)
```


swiss$Education
0
10
20
30
40
50
40
50
60
70
80
90
swiss$Fertility

EXAMPLE: SWISS DATASET

The plot can be prettified and made more informative:

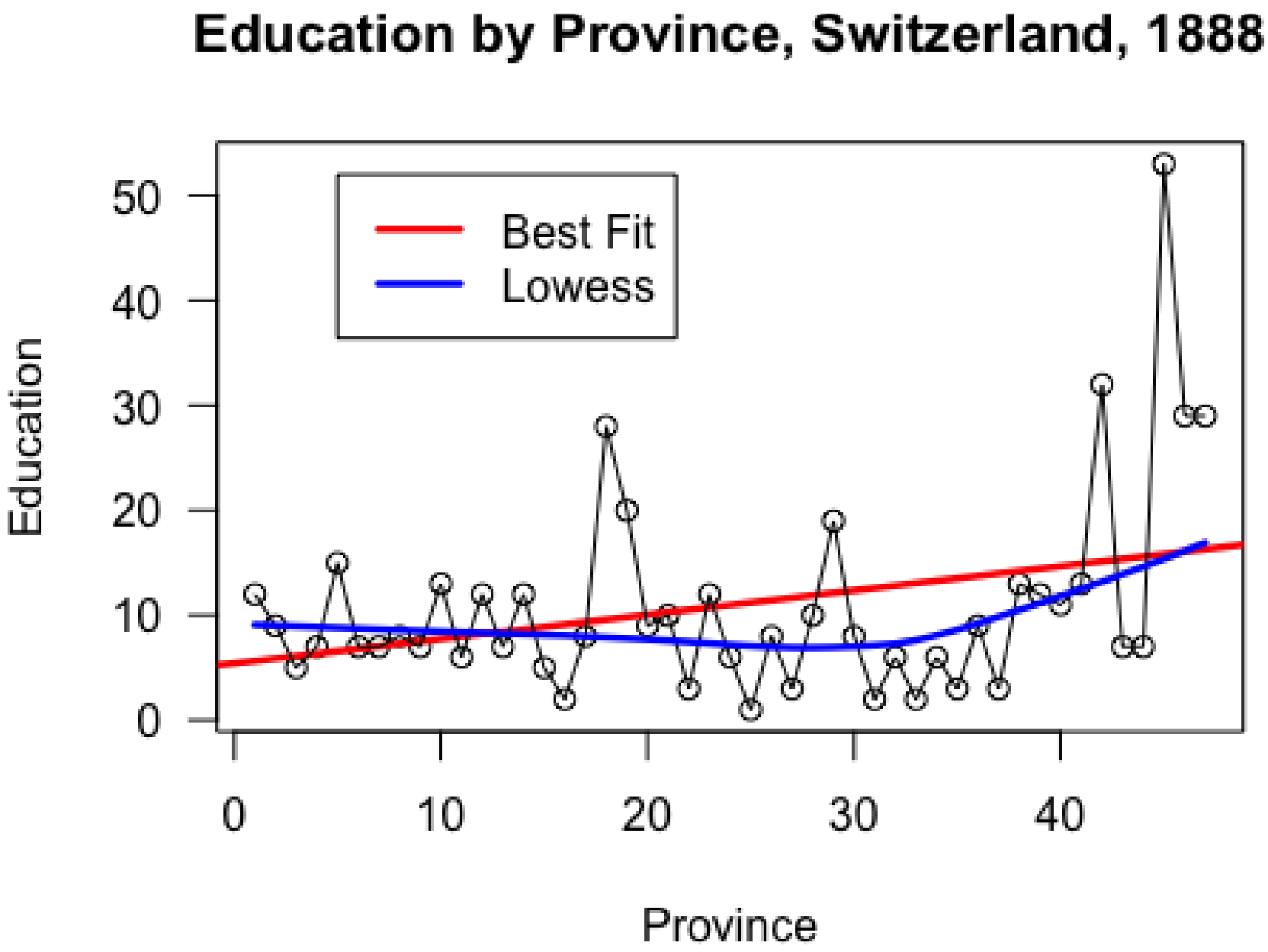
```
# add a title and axis labels
> plot(swiss$Fertility, swiss$Education, xlab="Fertility", ylab="Education",
      main="Education vs Fertility (by province), Switzerland, 1888", las=1)

# add the line of best fit (in red)
> abline(lm(swiss$Education~swiss$Fertility), col="red", lwd=2.5)

# add the smoothing lowess curve (in blue)
> lines(lowess(swiss$Fertility,swiss$Education), col="blue", lwd=2.5)

# add a legend
> legend(75,50, c("Best Fit","Lowess"), lty=c(1,1), lwd=c(2.5,2.5),
      col=c("red","blue"))
```


Education vs Fertility (by province), Switzerland, 18
Education
Fertility
Best Fit
Lowess
0
10
20
30
40
50
40
50
60
70
80
90

EXAMPLE: SWISS DATASET

Compare that graph with the one found on the next page:

```
> plot(swiss$Education, xlab="Province", ylab="Education", main="Education by
      Province, Switzerland, 1888", las=1)

> abline(lm(swiss$Education~row(swiss)[,1]), col="red", lwd=2.5)

> lines(swiss$Education)

> lines(lowess(row(swiss)[,1],swiss$Education), col="blue", lwd=2.5)

> legend(5,52, c("Best Fit","Lowess"), lty=c(1,1), lwd=c(2.5,2.5),
      col=c("red","blue"))
```


Education by Province, Switzerland, 1888
Best Fit
Lowess
Education
Province
0
10
20
30
40
50
0
10
20
30
40

EXAMPLE: SWISS DATASET

So we can get an actual graph here, but … why doesn't it actually make sense to produce that *specific* graph?

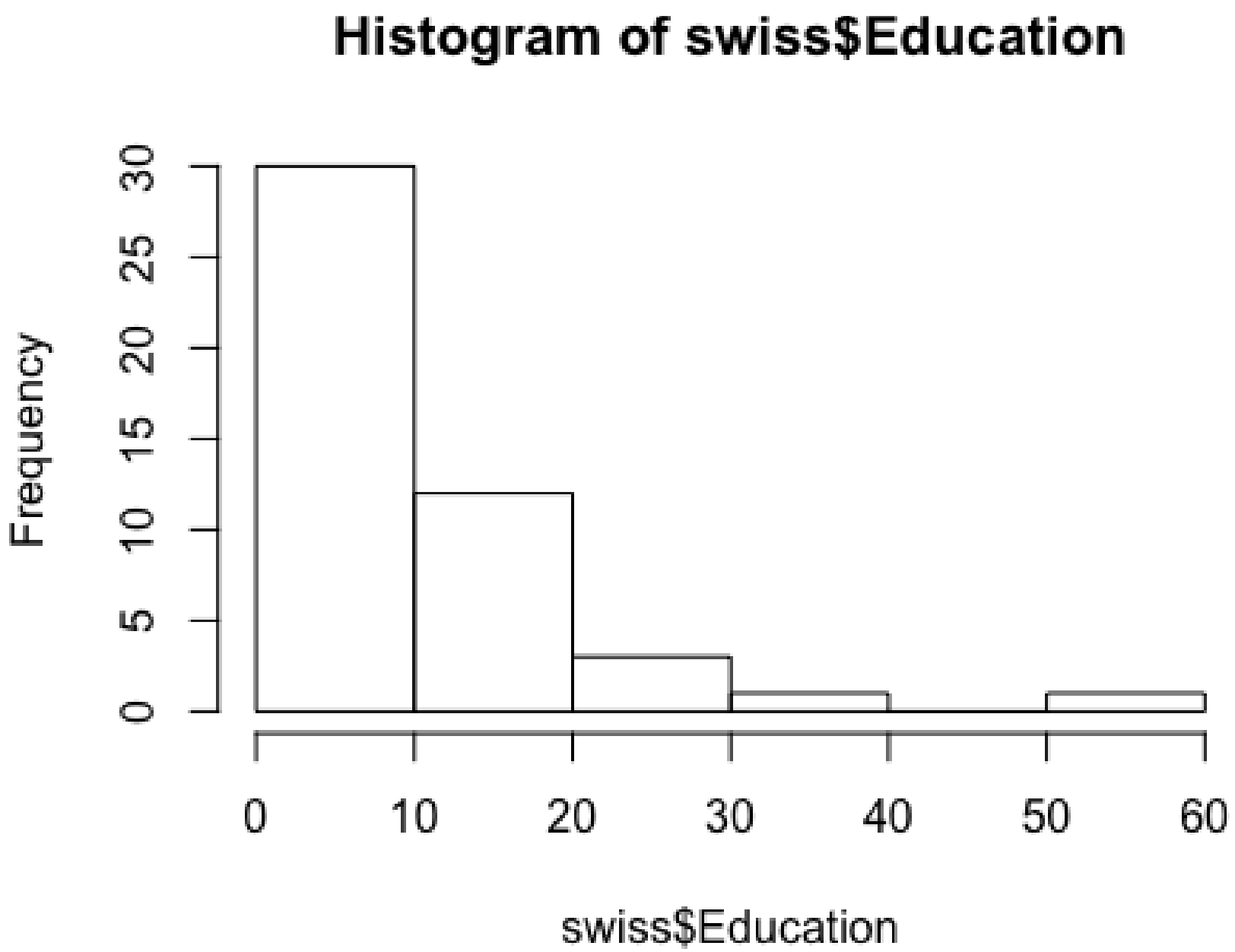
What do the curves **mean**?

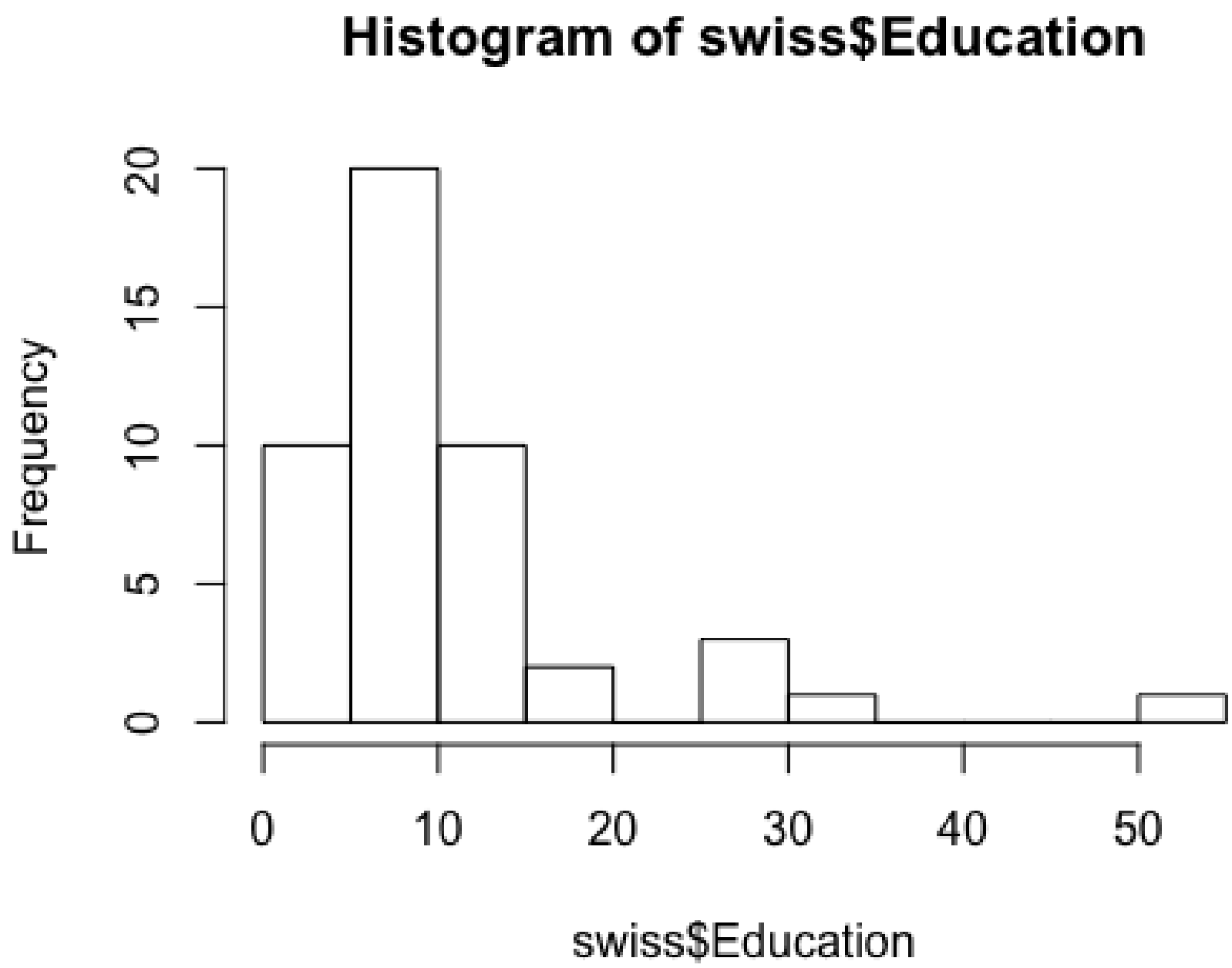
Take-Away: the fact that `R` can produce a graph doesn't guarantee that it will be useful or meaningful in any way.

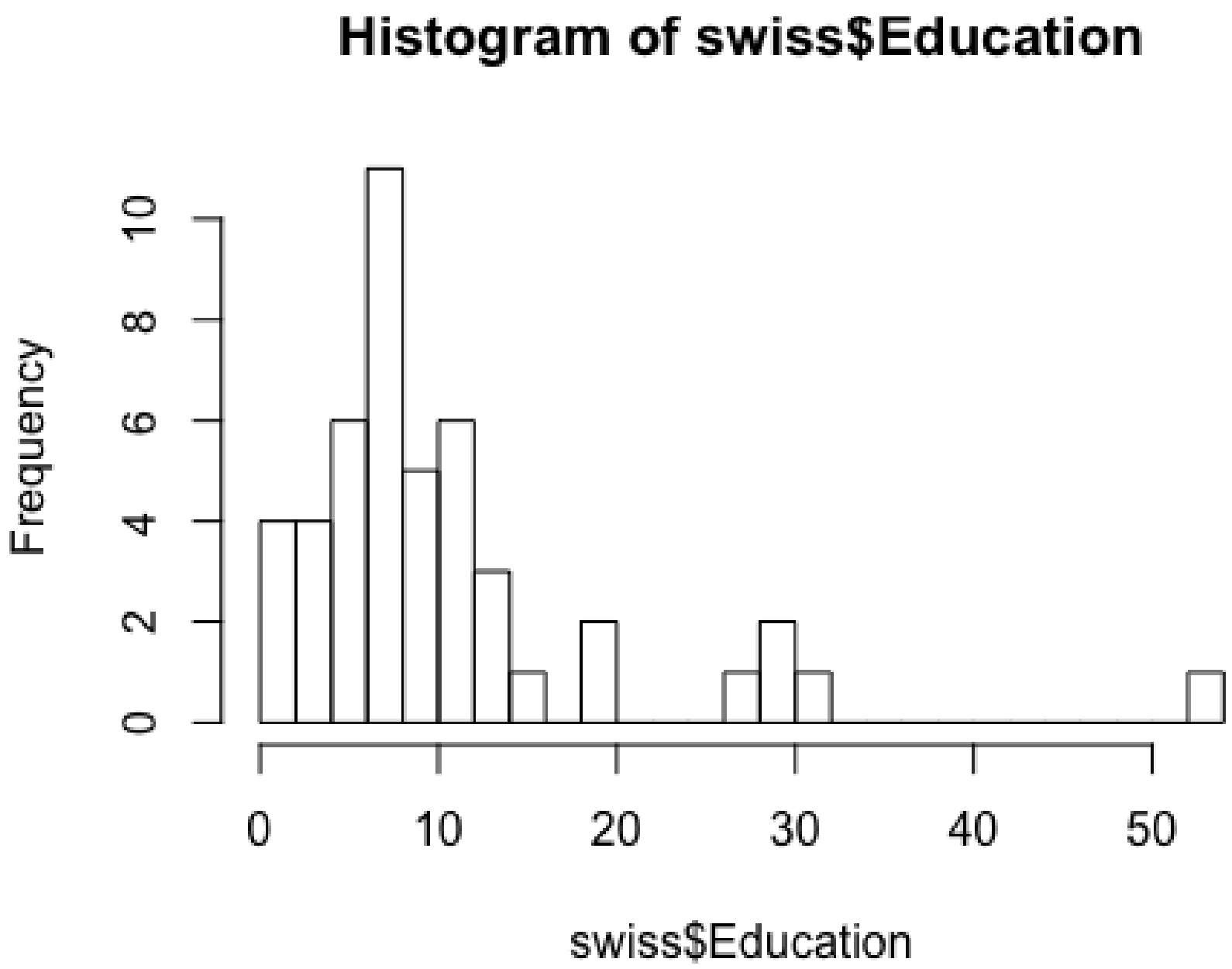
EXAMPLE: SWISS DATASET

What does the distribution of the `Education` variable look like?

```
## Histogram/Bar Charts
> hist(swiss$Education)   # default number of bins
> hist(swiss$Education, breaks=10)   # with 10 bins
> hist(swiss$Education, breaks=20)   # with 20 bins
```


Histogram of swiss$Education
Frequency
0
5
10
15
20
25
30
0
10
20
30
40
50
60
swiss$Education


Histogram of swiss$Education
Frequency
0
5
10
15
20
0
10
20
30
40
50
swiss$Education

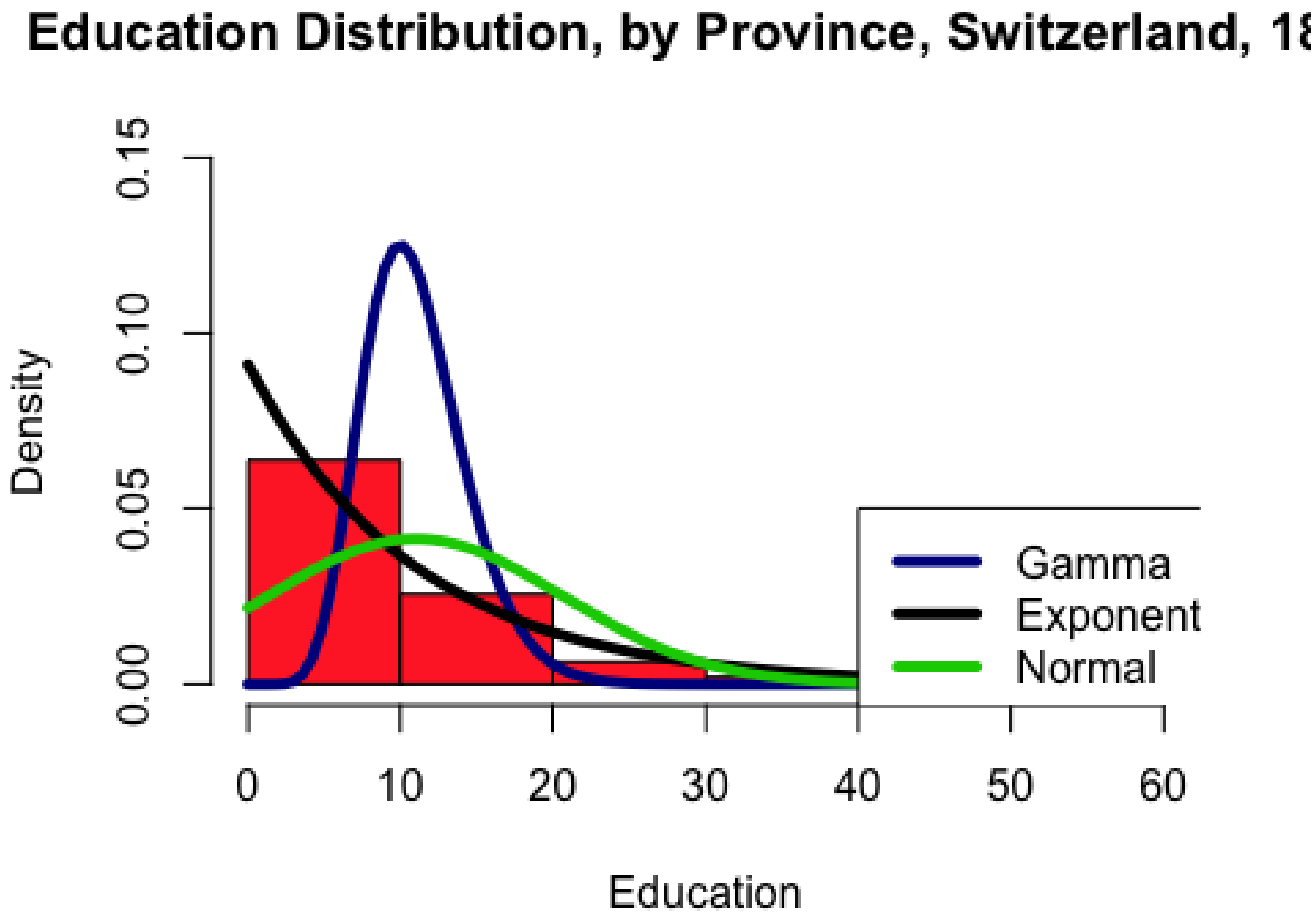

Histogram of swiss$Education
Frequency
0
2
4
6
8
10
0
10
20
30
40
50
swiss$Education

EXAMPLE: SWISS DATASET

The distribution pattern is distincly different with 10 and 20 bins. Don't get carried away: too many bins may end up masking trends if the dataset isn't large enough.

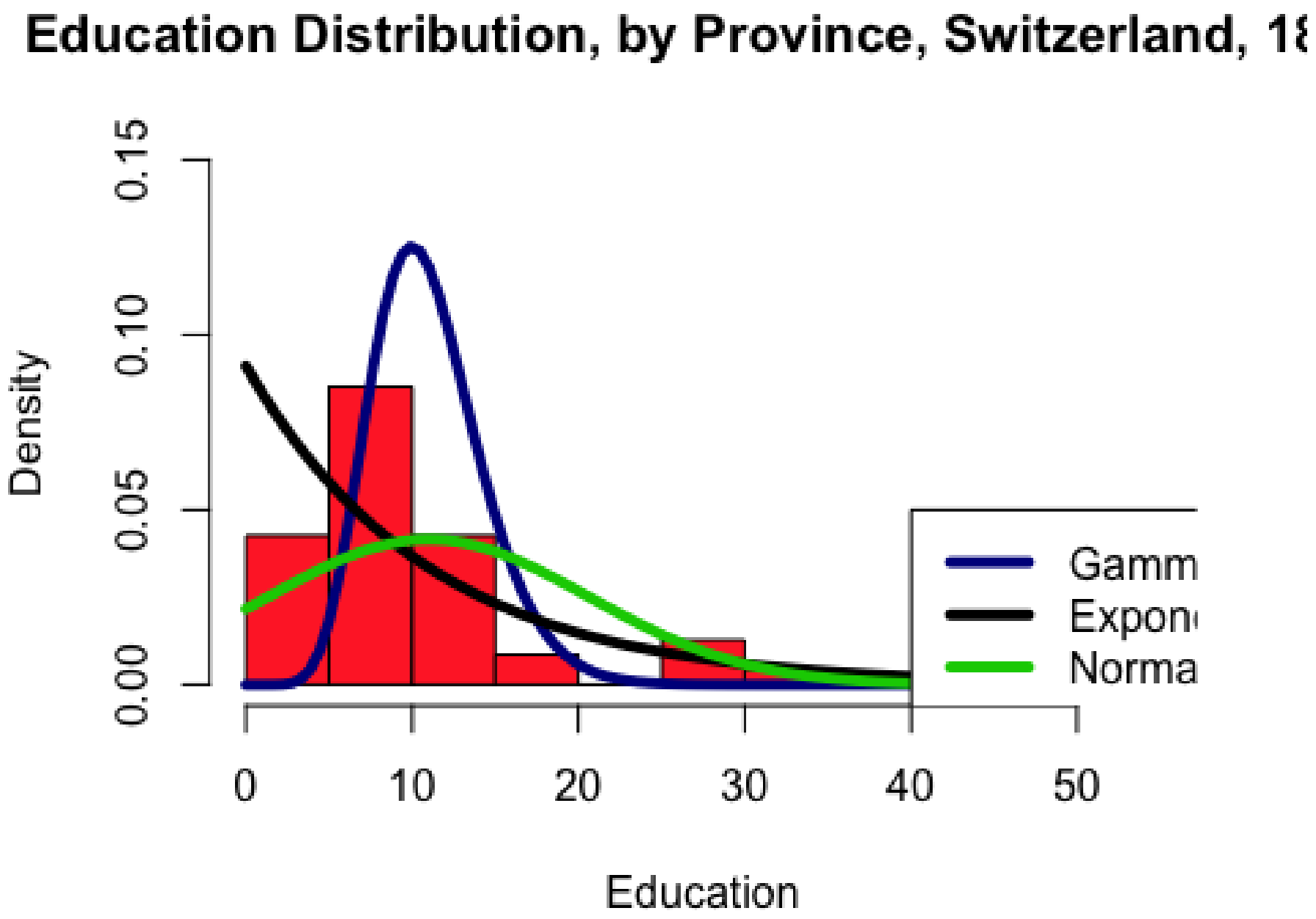
We can look for best fits for various parametric distributions (gamma, exp, normal):

```
> hist(swiss$Education, freq=FALSE, xlab="Education",main="Education Distribution,
   by Province, Switzerland, 1888", col="firebrick1", ylim=c(0,0.15))

> curve(dgamma(x,shape=mean(swiss$Education)),add=TRUE, col="darkblue", lwd=4)

> curve(dexp(x,rate=1/mean(swiss$Education)),add=TRUE, col="black", lwd=4)

> curve(dnorm(x,mean=mean(swiss$Education),sd=sd(swiss$Education)), add=TRUE,
      col="green3", lwd=4)

> legend(40,0.05, c("Gamma","Exponential","Normal"), lty=c(1,1), lwd=c(4,4),
      col=c("darkblue","black", "green3"))
```


Education Distribution, by Province, Switzerland, 18
Density
0.00
0.05
0.10
0.15
0
10
20
30
40
50
60
Education
Gamma
Exponent
Normal

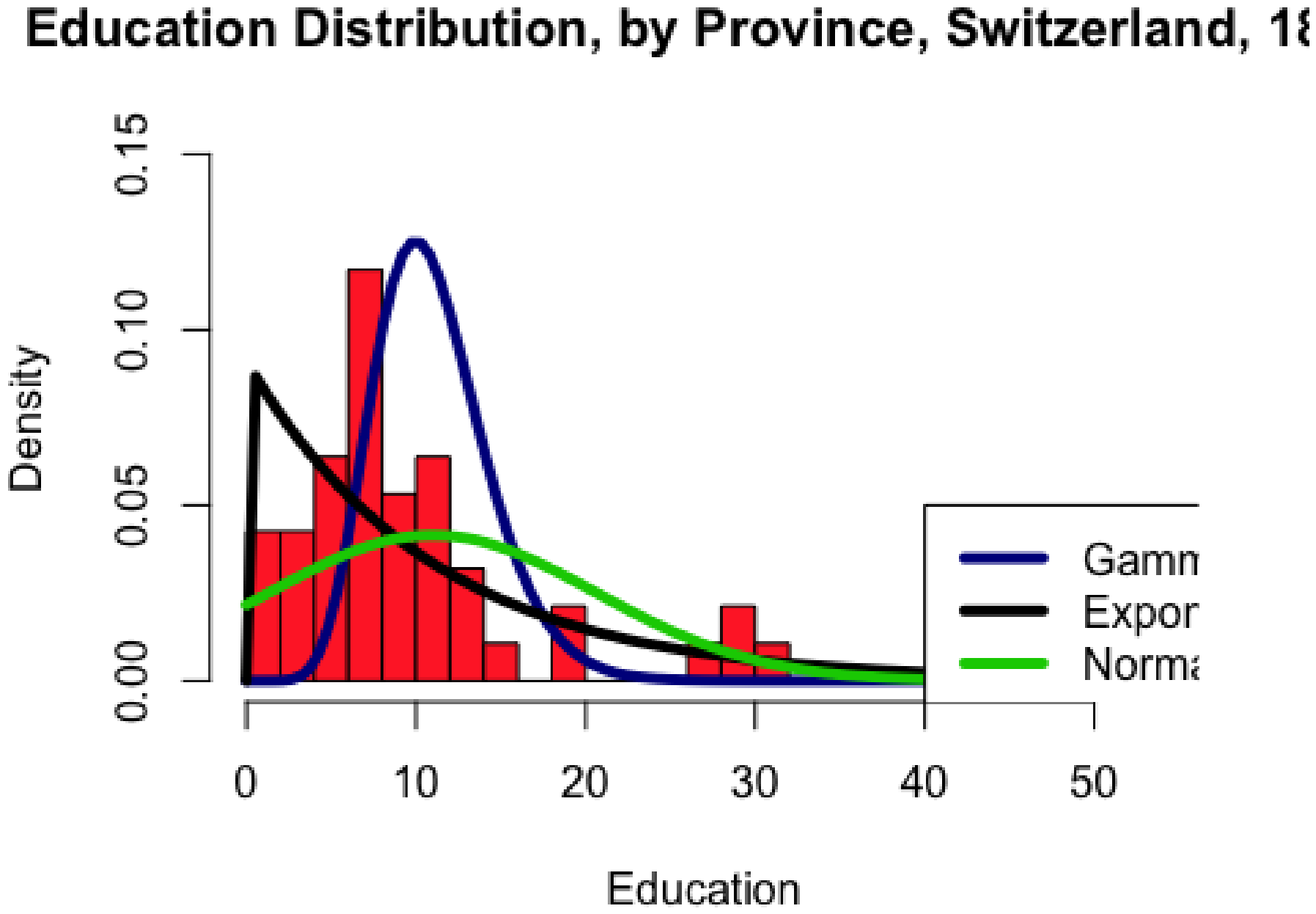
EXAMPLE: SWISS DATASET

```
> hist(swiss$Education, breaks=10, freq=FALSE, xlab="Education",
      main="Education Distribution, by Province, Switzerland, 1888",
      col="firebrick1", ylim=c(0,0.15))

> curve(dgamma(x,shape=mean(swiss$Education)),add=TRUE, col="darkblue", lwd=4)

> curve(dexp(x,rate=1/mean(swiss$Education)),add=TRUE, col="black", lwd=4)

> curve(dnorm(x,mean=mean(swiss$Education),sd=sd(swiss$Education)), add=TRUE,
      col="green3", lwd=4)

> legend(40,0.05, c("Gamma","Exponential","Normal"), lty=c(1,1), lwd=c(4,4),
      col=c("darkblue","black", "green3"))
```


Education Distribution, by Province, Switzerland, 18
Density
0.00
0.05
0.10
0.15
0
10
20
30
40
50
Education
Gamm
Expon
Norma

EXAMPLE: SWISS DATASET

```
> hist(swiss$Education, breaks=20, freq=FALSE, xlab="Education",
      main="Education Distribution, by Province, Switzerland, 1888",
      col="firebrick1", ylim=c(0,0.15))

> curve(dgamma(x,shape=mean(swiss$Education)),add=TRUE, col="darkblue", lwd=4)

> curve(dexp(x,rate=1/mean(swiss$Education)),add=TRUE, col="black", lwd=4)

> curve(dnorm(x,mean=mean(swiss$Education),sd=sd(swiss$Education)), add=TRUE,
      col="green3", lwd=4)

> legend(40,0.05, c("Gamma","Exponential","Normal"), lty=c(1,1), lwd=c(4,4),
      col=c("darkblue","black", "green3"))
```


Education Distribution, by Province, Switzerland, 18
Density
0.00
0.05
0.10
0.15
0
10
20
30
40
50
Education
Gamn
Expor
Norma

EXAMPLE: SWISS DATASET

With a small number of bins, the exponential distribution seems like a good fit.

With a larger number of bins, neither of the three families seems particularly well-advised.

Let's try out another classic dataset. Remember that our purpose is to show what can be done with base R graphing capability: it's more than adequate for exploration.

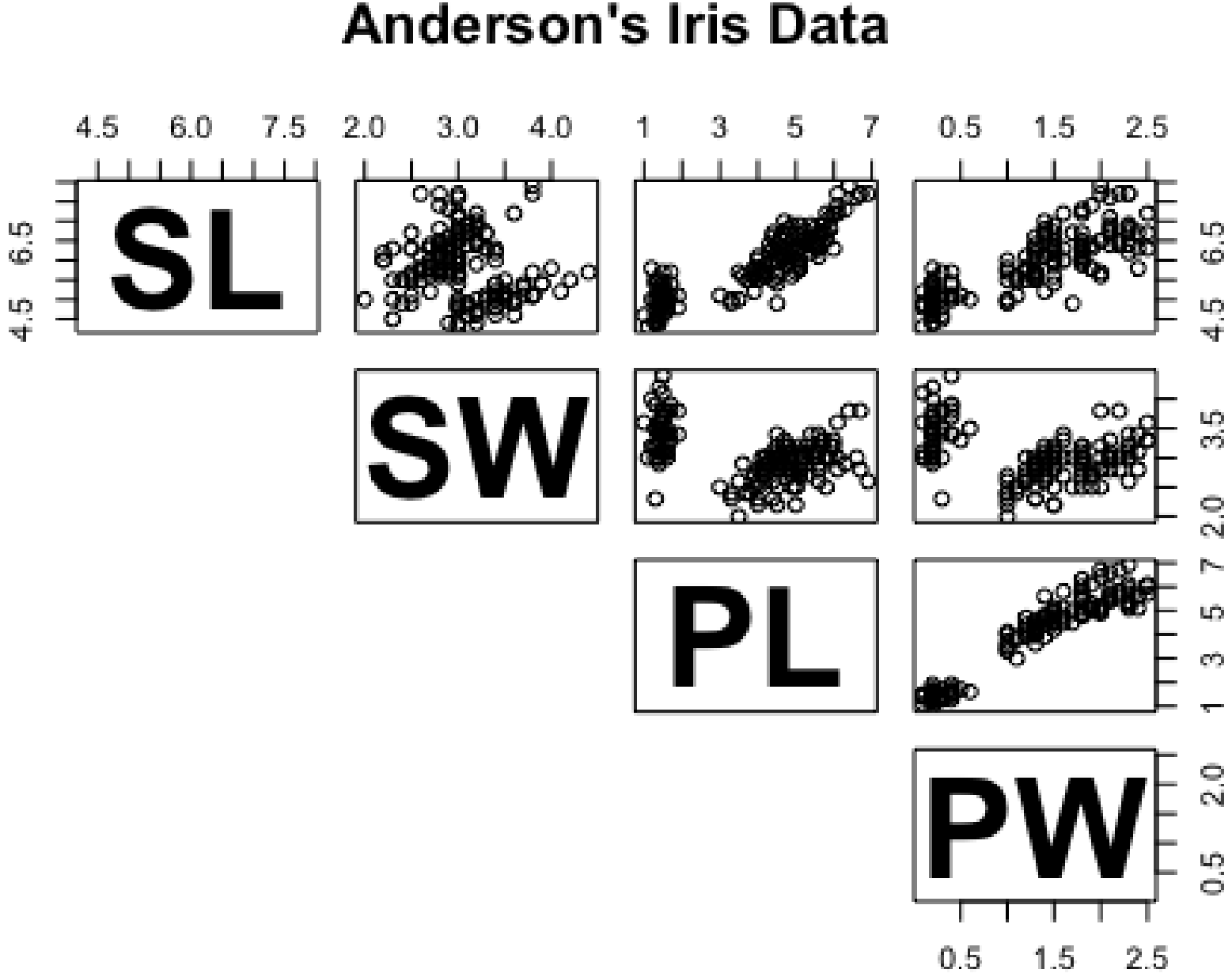
EXAMPLE: IRIS DATASET

Let's do the same thing for the built-in `iris` dataset.

```
> str(iris) # structure of the dataset
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..:  1 1 1 1 1 ...

> summary(iris) # information on the distributions for each feature
##    Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

EXAMPLE: IRIS DATASET

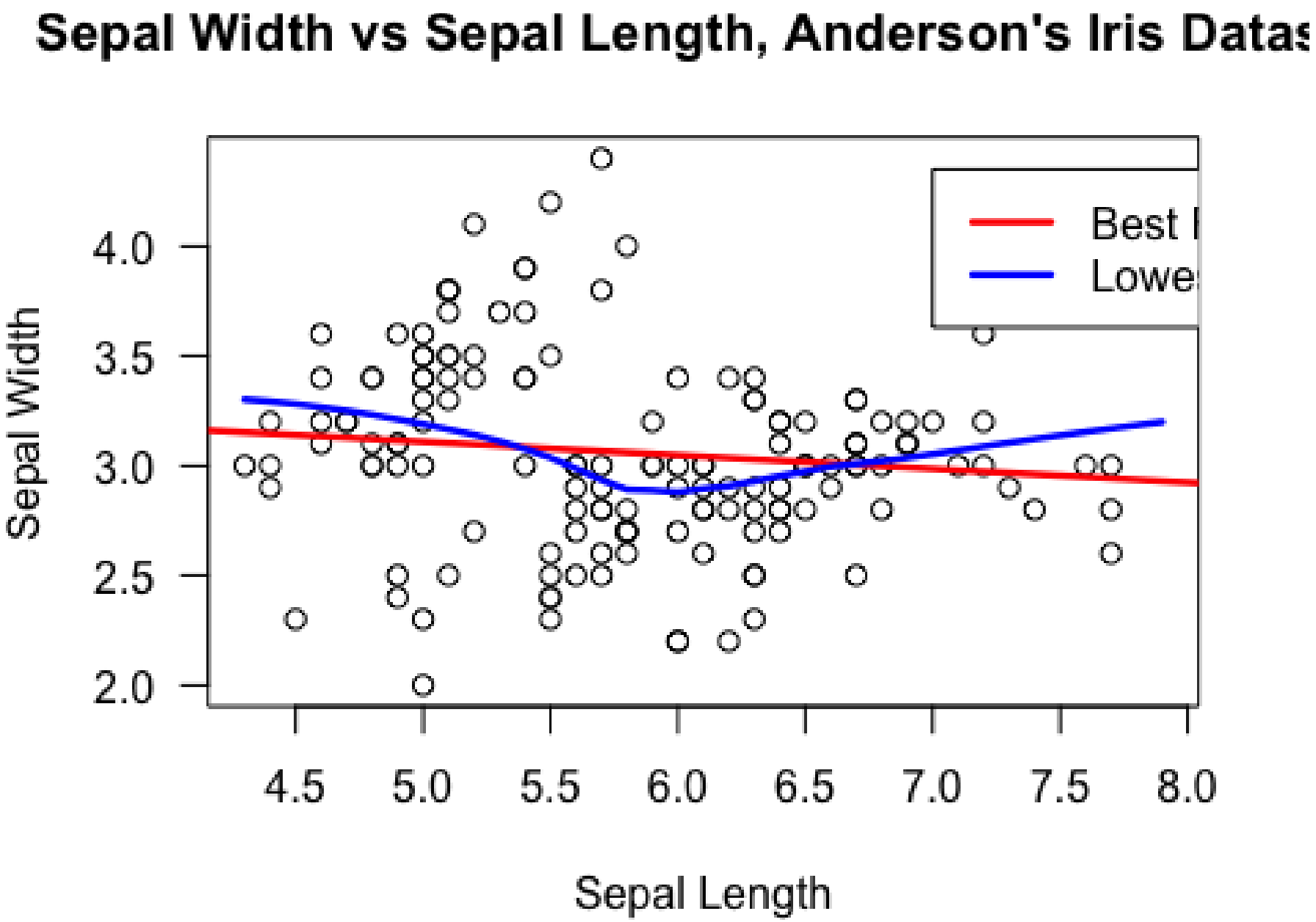
```
# information on the dataset itself
> ?iris

# scatter plot matrix on which the lower panel has
been removed due to redundancy

> pairs(iris[1:4], main = "Anderson's Iris Data",
      pch = 21, lower.panel=NULL,
      labels=c("SL","SW","PL","PW"), font.labels=2,
      cex.labels=4.5)
```


Anderson's Iris Data
SL
SW
PL
PW
4.5
6.0
7.5
2.0
3.0
4.0
1
3
5
7
0.5
1.5
2.5
4.5
6.5
2.0
3.5
1
3
5
7
0.5
1.5
2.5

EXAMPLE: IRIS DATASET

We can compare the sepal width and length variables in a manner similar to what we did with the `swiss` dataset.

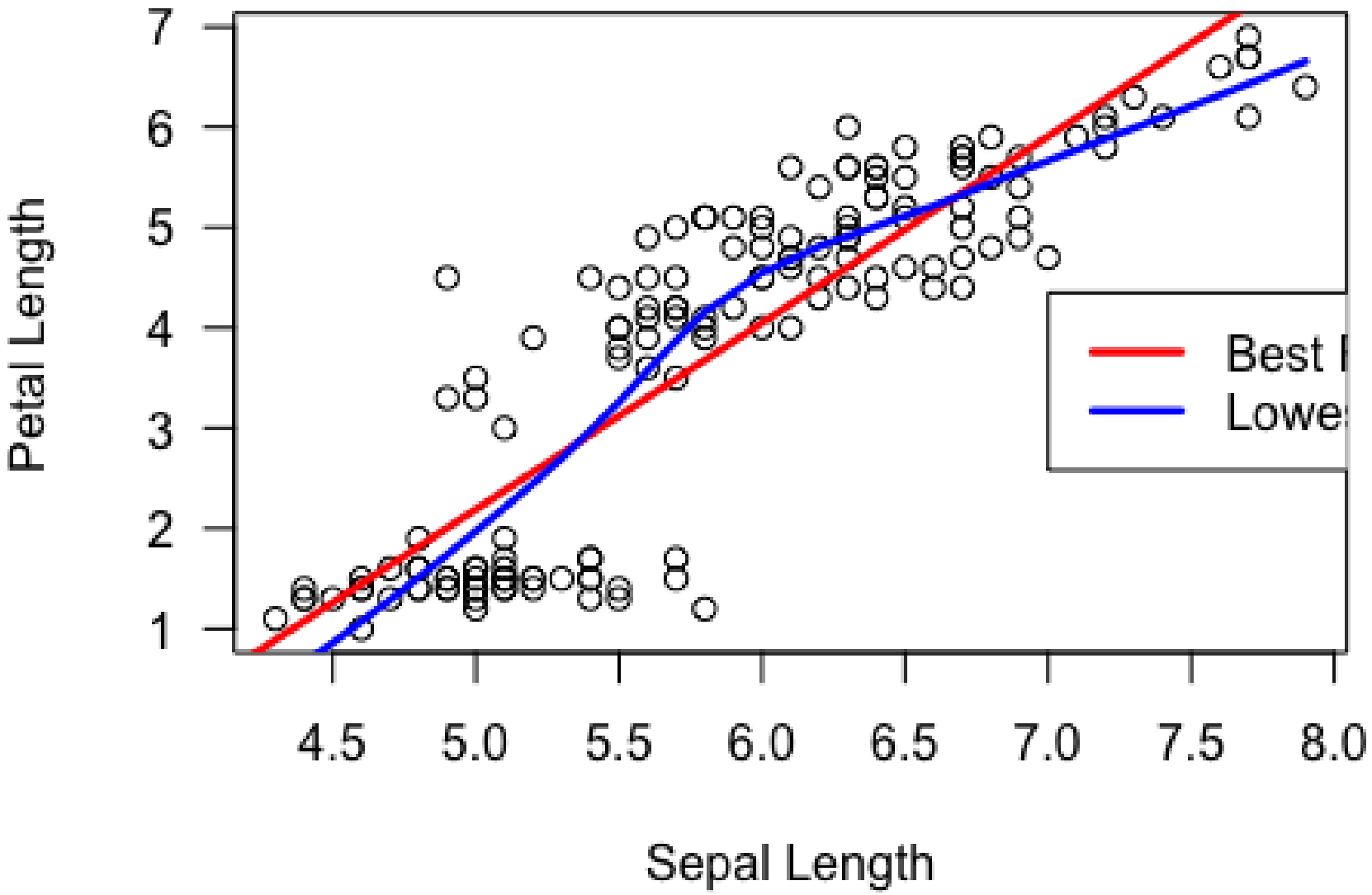
```
## Iris 1
> plot(iris$Sepal.Length, iris$Sepal.Width, xlab="Sepal Length",
      ylab="Sepal Width", main="Sepal Width vs Sepal Length, Anderson's Iris
      Dataset", las=1, bg=c("yellow","black","green")[unclass(iris$Species)])

> abline(lm(iris$Sepal.Width~iris$Sepal.Length), col="red", lwd=2.5)

> lines(lowess(iris$Sepal.Length,iris$Sepal.Width), col="blue", lwd=2.5)

> legend(7,4.35, c("Best Fit","Lowess"), lty=c(1,1), lwd=c(2.5,2.5),
      col=c("red","blue"))
```


Sepal Width vs Sepal Length, Anderson's Iris Datas
Best
Lowe
Sepal Width
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Sepal Length

EXAMPLE: IRIS DATASET

There does not seem to be a very strong relationship between these variables. What can we say about sepal length and petal length?

```
## Iris 2
> plot(iris$Sepal.Length, iris$Petal.Length, xlab="Sepal Length",
      ylab="Petal Length", main="Sepal Width vs Petal Length, Anderson's Iris
      Dataset", las=1)

> abline(lm(iris$Petal.Length~iris$Sepal.Length), col="red", lwd=2.5)

> lines(lowess(iris$Sepal.Length,iris$Petal.Length), col="blue", lwd=2.5)

> legend(7,4.35, c("Best Fit","Lowess"), lty=c(1,1), lwd=c(2.5,2.5),
      col=c("red","blue"))
```

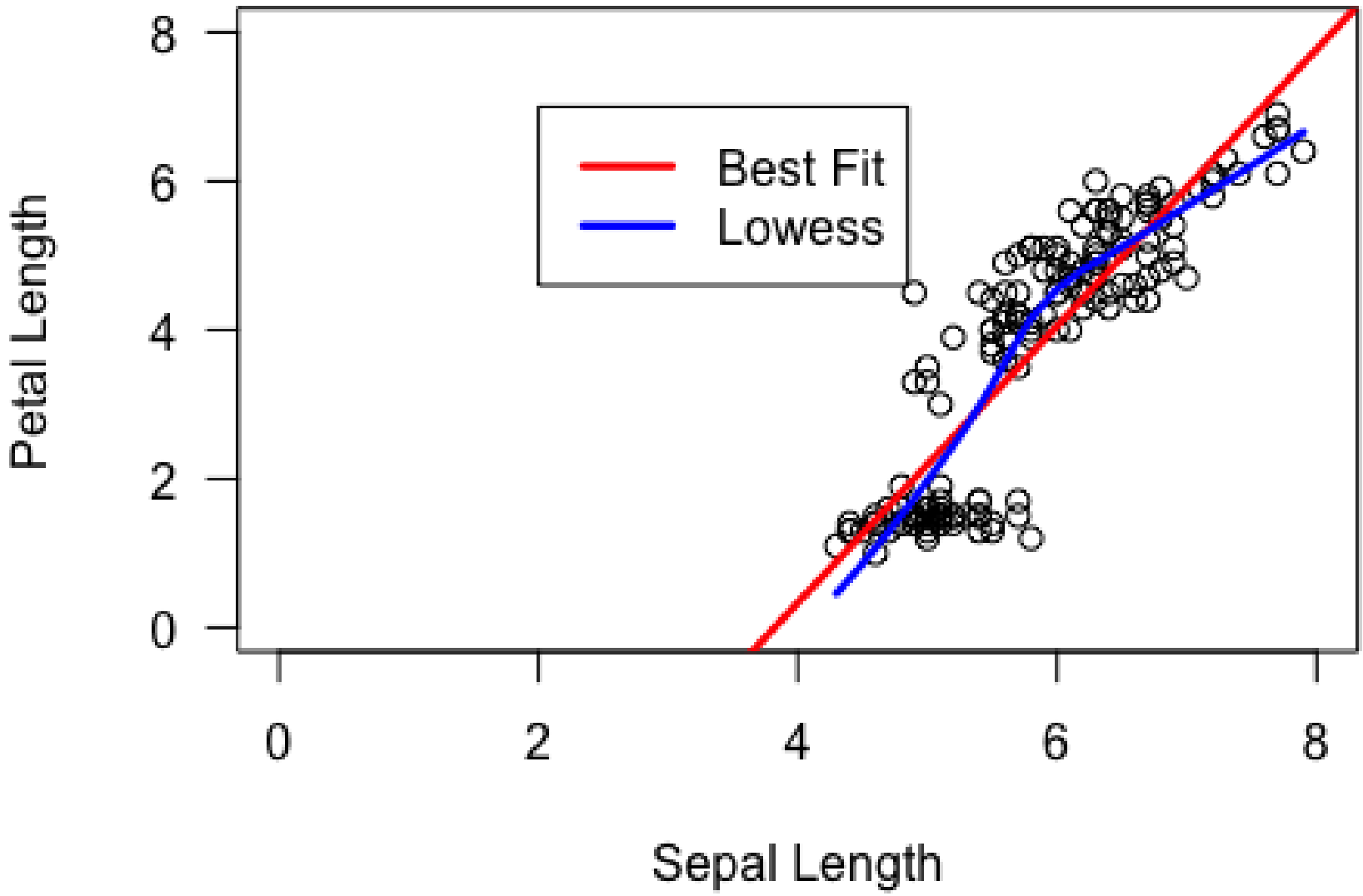

Sepal Width vs Petal Length, Anderson's Iris Datas
Petal Length
Sepal Length
1
2
3
4
5
6
7
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Best
Lowe

EXAMPLE: IRIS DATASET

Visually, the relationship is striking: the line seems to have a slope of 1! But notice that the axes are unevenly scaled, and have been cutoff away from the origin. The following graph gives a better idea of the situation.

```
## Iris 3
> plot(iris$Sepal.Length, iris$Petal.Length, xlab="Sepal Length",
      ylab="Petal Length", main="Sepal Width vs Petal Length, Anderson's Iris
      Dataset", xlim=c(0,8), ylim=c(0,8), las=1)

> abline(lm(iris$Petal.Length~iris$Sepal.Length), col="red", lwd=2.5)

> lines(lowess(iris$Sepal.Length,iris$Petal.Length), col="blue", lwd=2.5)

> legend(2,7, c("Best Fit","Lowess"), lty=c(1,1), lwd=c(2.5,2.5),
      col=c("red","blue"))
```

Sepal Width vs Petal Length, Anderson's Iris Datas

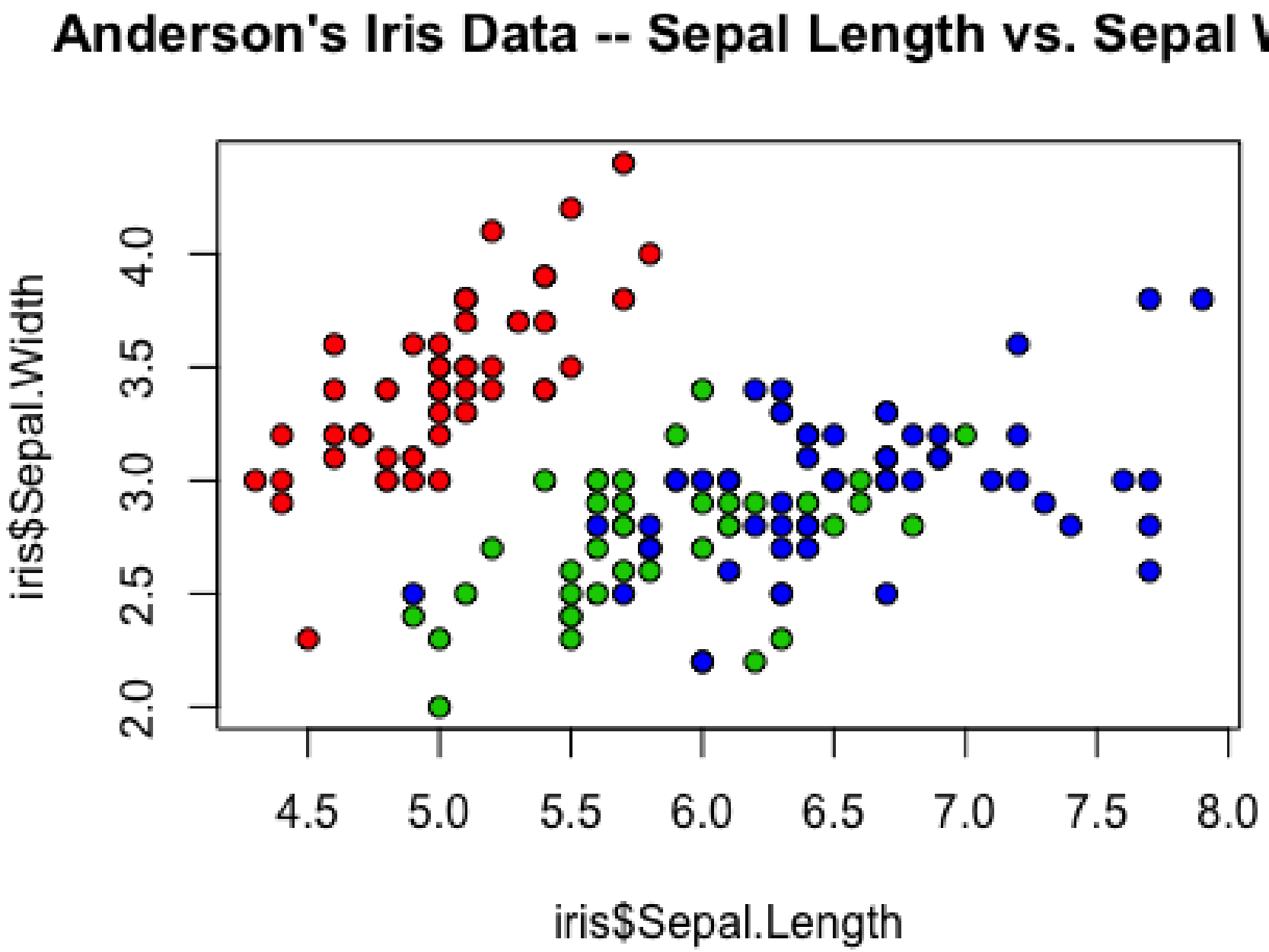


EXAMPLE: IRIS DATASET

A relationship is still present, but it is **affine**, not linear as could have been guessed by naively looking at the original graph.

Colour can also be used to highlight various data elements:

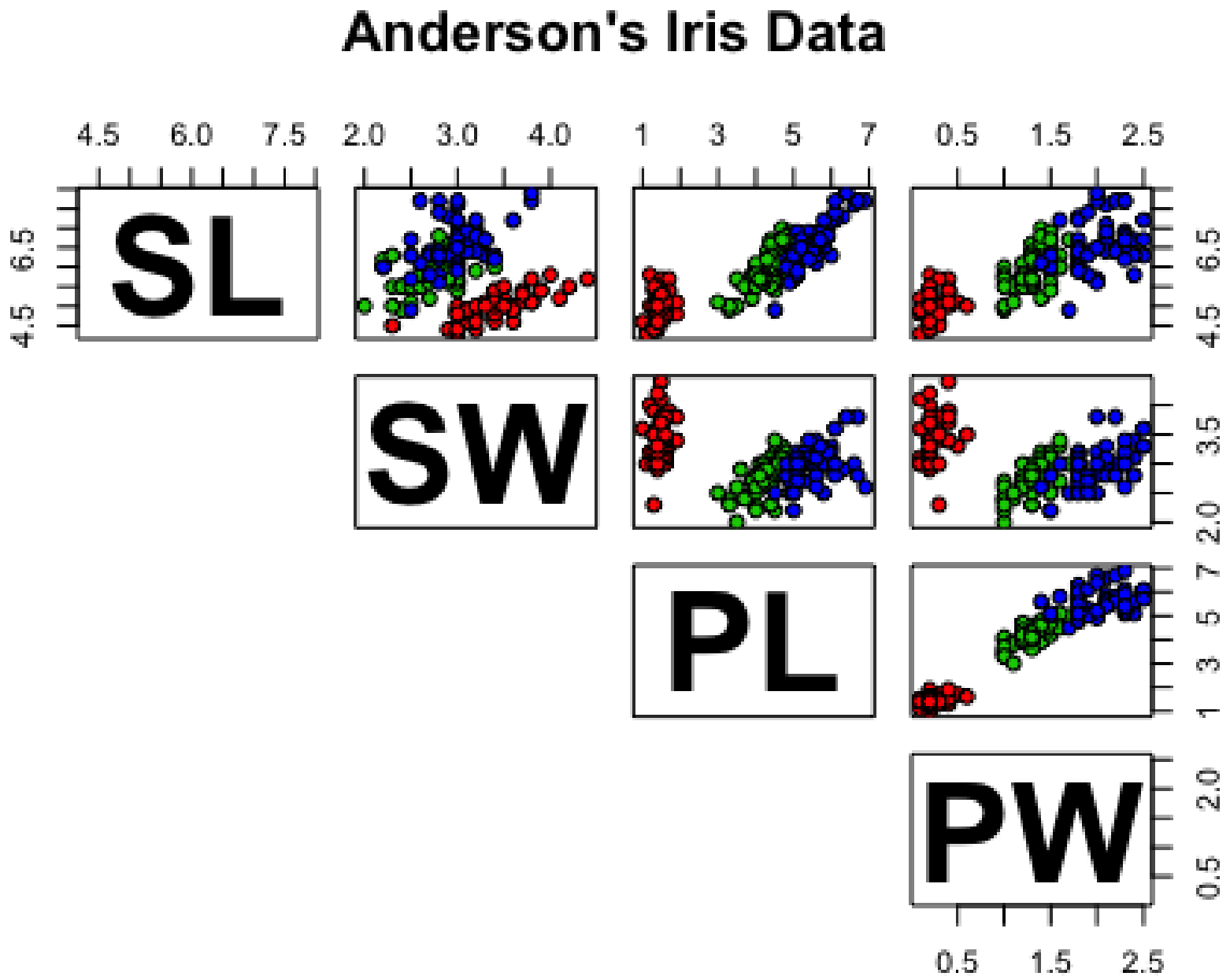
```
# colour each observation differently according to its species
> plot(iris$Sepal.Length, iris$Sepal.Width, pch=21,
      bg=c("red","green3","blue")[unclass(iris$Species)],
      main="Anderson's Iris Data -- Sepal Length vs. Sepal
      Width")
```


Anderson's Iris Data -- Sepal Length vs. Sepal Wid
iris$Sepal.Width
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
iris$Sepal.Length

EXAMPLE: IRIS DATASET

This can be done on all scatterplots concurrently using `pairs`.

```
# scatterplot matrix with species membership
> pairs(iris[1:4], main = "Anderson's Iris Data", pch = 21,
      bg = c("red", "green3", "blue")[unclass(iris$Species)],
      lower.panel=NULL, labels=c("SL","SW","PL","PW"),
      font.labels=2, cex.labels=4.5)
```


Anderson's Iris Data
SL
SW
PL
PW
4.5
6.0
7.5
2.0
3.0
4.0
1
3
5
7
0.5
1.5
2.5
4.5
6.5
2.0
3.5
1
3
5
7
0.5
1.5
2.5

EXAMPLE: IRIS DATASET

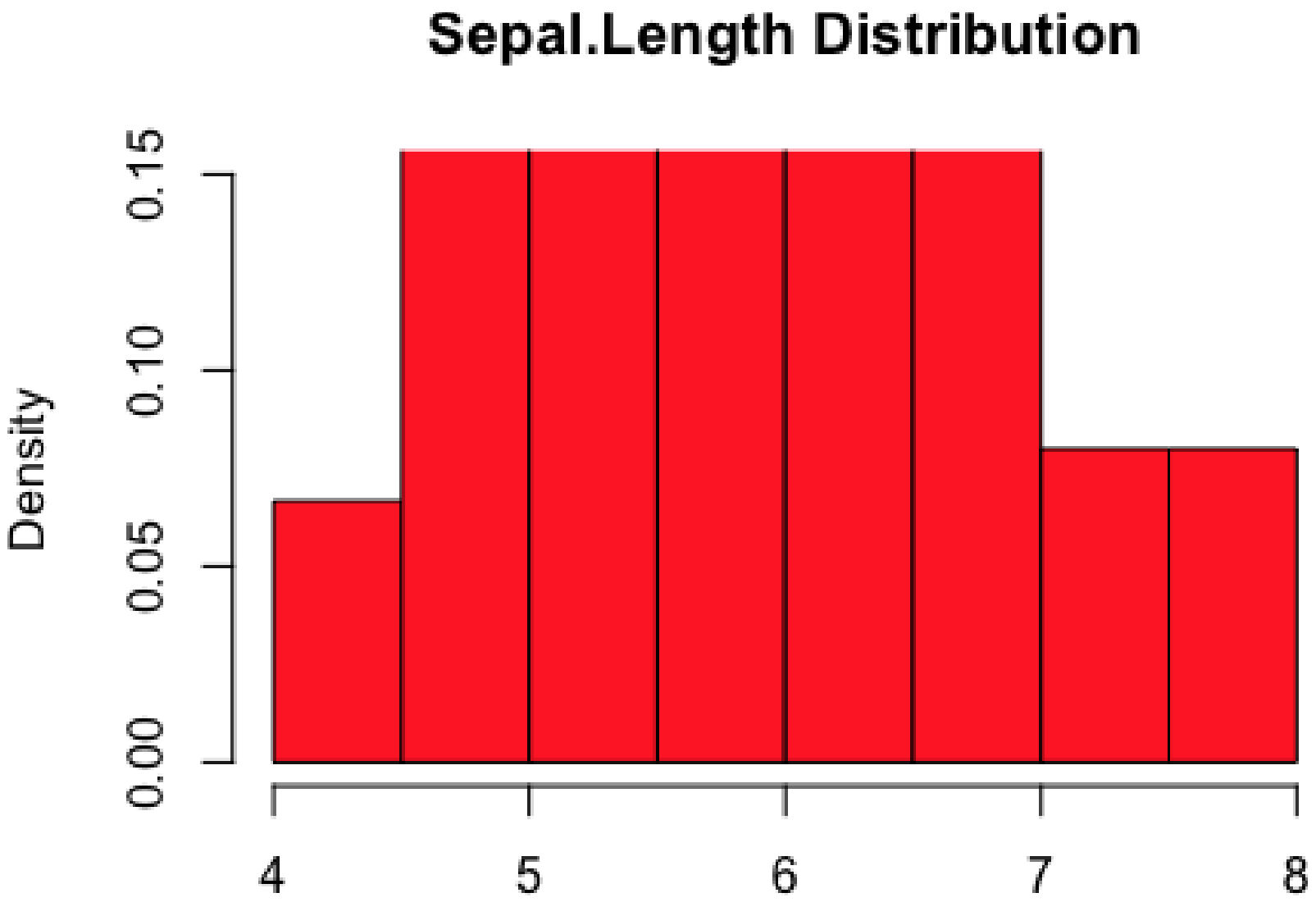
The redundancy in the scatter plot matrix can be used to display other data elements as well: `GGally` allows for **feature distributions** in the diagonal entries, and **correlations between pairs of variables** and **density plots** in the redundant panels (among other things).

Be weary of utilizing too many colours or features in these plots, however. They can easily become too difficult to read to provide any meaningful insights.

EXAMPLE: IRIS DATASET

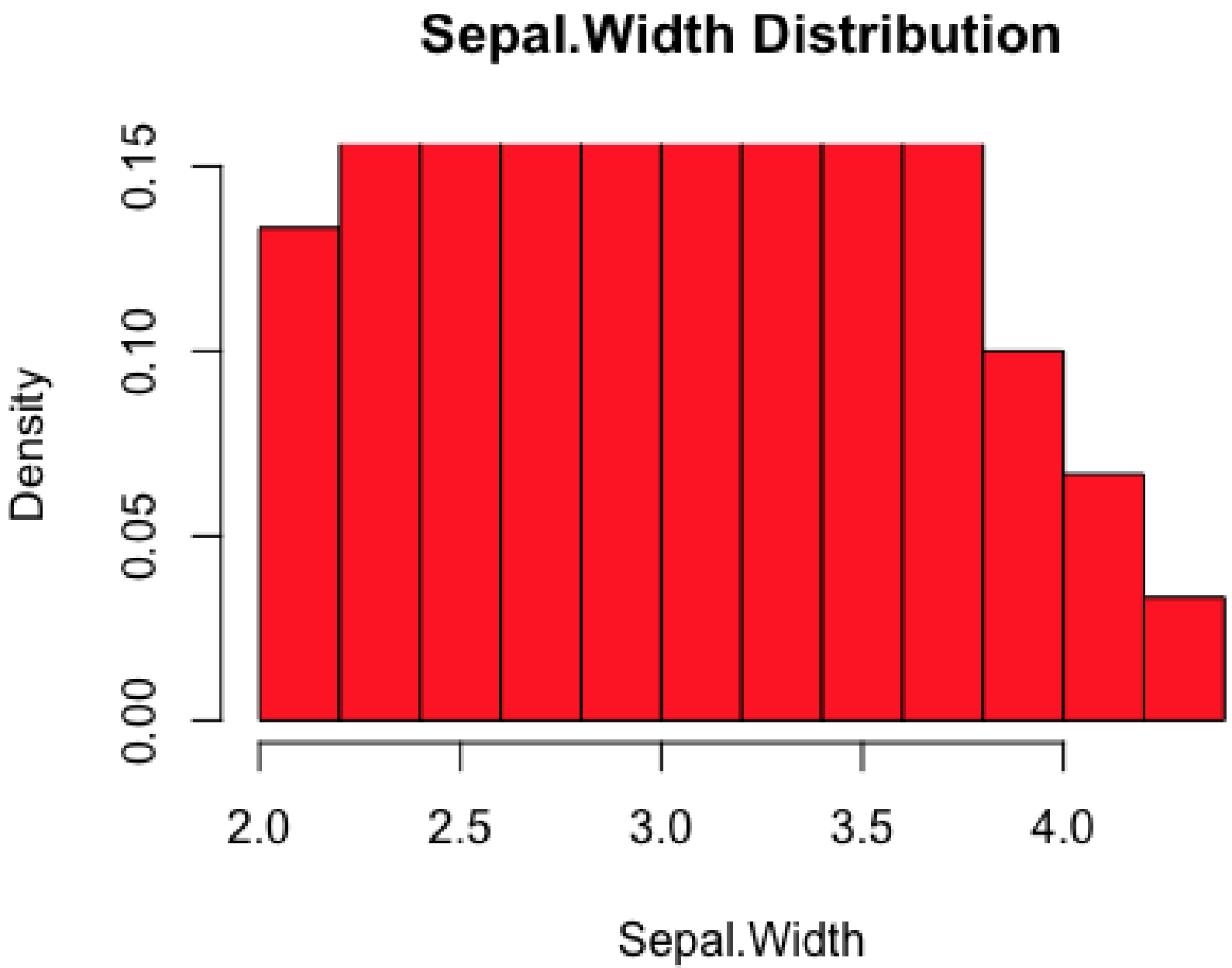
Can you figure out what is happening with these visualizations of the `iris` dataset?

```
> hist(iris$Sepal.Length, freq=FALSE,
     xlab="Sepal.Length",
     main="Sepal.Length Distribution",
     col="firebrick1",
     ylim=c(0,0.15))
```


Sepal.Length Distribution
Density
0.00
0.05
0.10
0.15
4
5
6
7
8

EXAMPLE: IRIS DATASET

```
# what happens if we replace freq=FALSE with freq=TRUE?

# Another feature
> hist(iris$Sepal.Width, freq=FALSE,
     xlab="Sepal.Width",
     main="Sepal.Width Distribution",
     col="firebrick1",
     ylim=c(0,0.15))
```

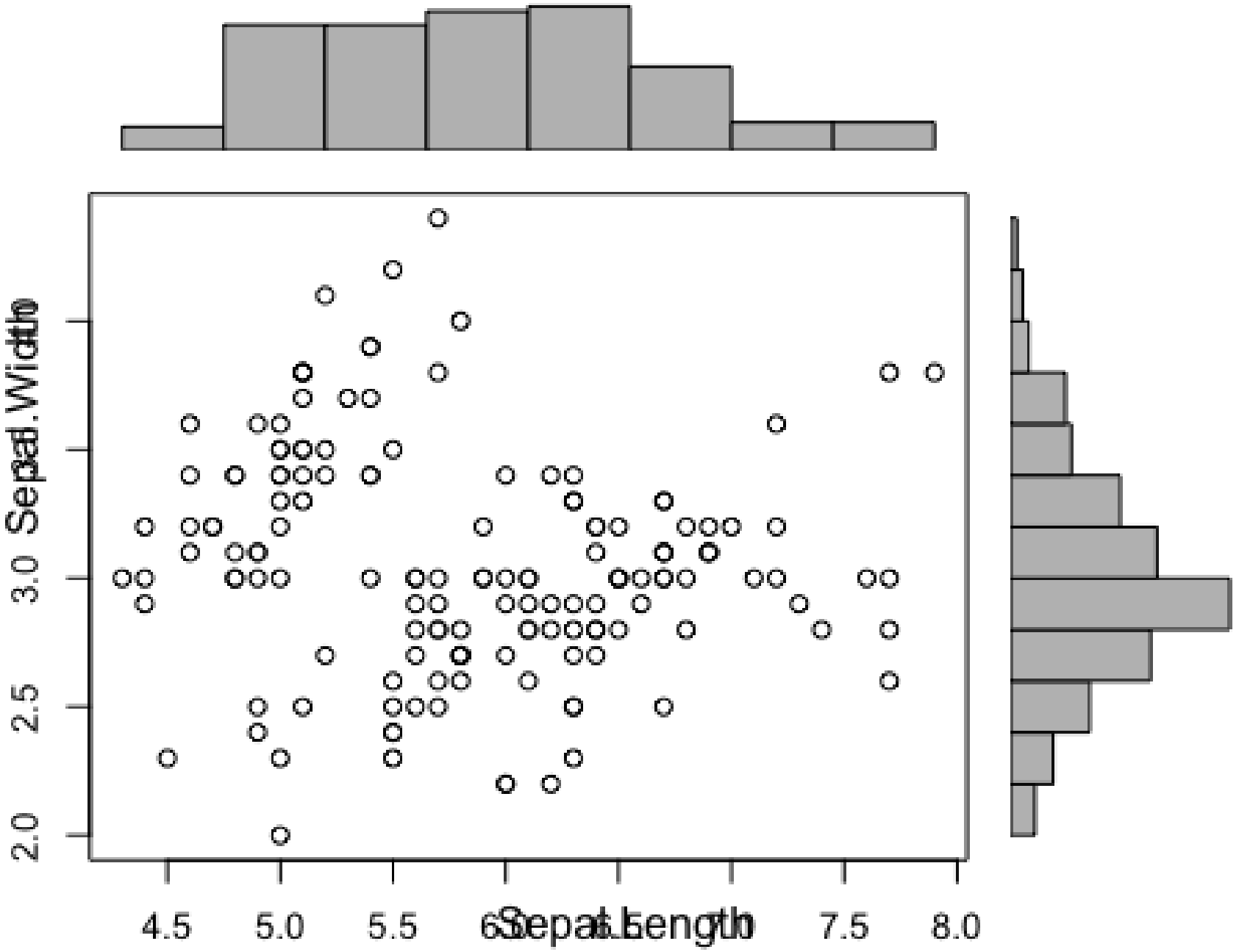

Sepal.Width Distribution
Density
0.00
0.05
0.10
0.15
2.0
2.5
3.0
3.5
4.0
Sepal.Width

EXAMPLE: IRIS DATASET

Histograms (1D data representations) can be combined with scatterplots (2D data representations) to provide **marginal** information.

This chart uses the user-defined `scatterhist` function (see notebook).

```
> ds = iris

> with(ds, scatterhist(iris$Sepal.Length, iris$Sepal.Width,
      xlab="Sepal.Length", ylab="Sepal.Width"))
```


Sepal.Width
2.0
2.5
3.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Sepal.Length

EXAMPLE: CMA & CA

Bubble charts are a neat way to show at least 3 variables on the same 2D display. The location of the bubbles' centre takes care of 2 variables: size, colour, and shape of bubbles can also be added to represent different data elements.

For this example, we'll look at demographic data regarding Canada's CMA and CA in 2011 (from StatsCan).

```
# import the data
> can.2011=read.csv("../Data/Canada2011.csv", head=TRUE)
```

EXAMPLE: CMA & CA

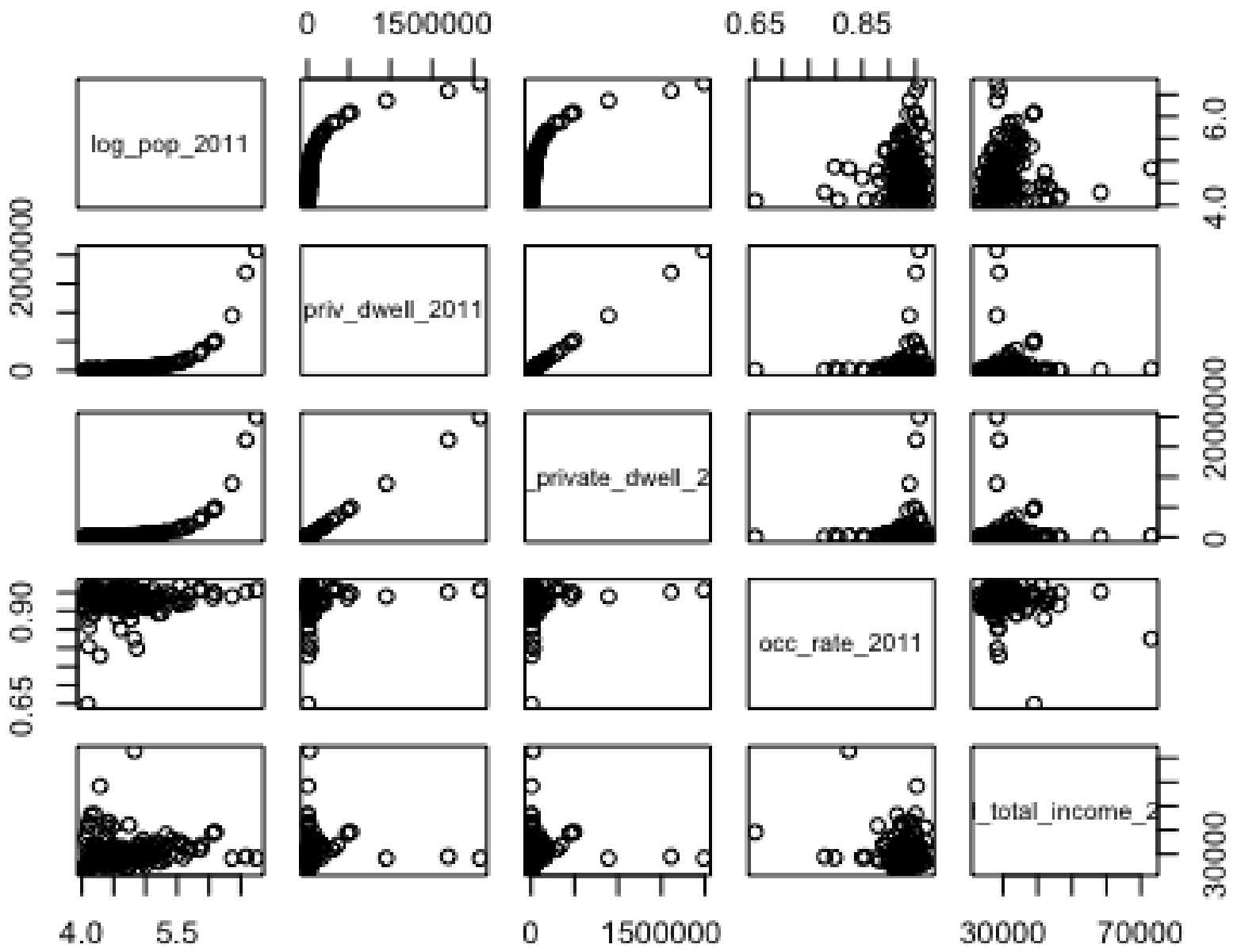
```
# take a look at the structure of the data
> str(can.2011)
## 'data.frame':    147 obs. of  12 variables:
##  $ Geographic.code       : int  1 5 10 15 105 110 205 210 215 220 ...
##  $ Geographic.name       : Factor w/ 147 levels "Abbotsford - Mission",..
##  $ Province              : Factor w/ 12 levels "AB","BC","MB",..: 5 5 5 5 ...
##  $ Region                : Factor w/ 6 levels "Atlantic","British Columbia"
##  $ Type                  : Factor w/ 2 levels "CA","CMA": 2 1 1 1 1 1 2 ...
##  $ pop_2011              : int  196966 10871 13725 27202 64487 16488 ...
##  $ log_pop_2011          : num  5.29 4.04 4.14 4.43 4.81 ...
##  $ pop_rank_2011         : int  20 147 128 94 52 120 13 97 67 78 ...
##  $ priv_dwell_2011       : int  84542 4601 6134 11697 28864 7323 ...
##  $ occ_private_dwell_2011: int  78960 4218 5723 11110 26192 19492 15256 ...
##  $ occ_rate_2011         : num  0.934 0.917 0.933 0.95 0.907 ...
##  $ med_total_income_2011 : int  33420 24700 26920 27430 30110 ...
```

EXAMPLE: CMA & CA

```
# provide a distribution information for features 3 to 12, allowing for up to 13
factors in the categorical distributions
> summary(can.2011[,3:12], maxsum=13)
##  Province                Region    Type        pop_2011
##  AB:18    Atlantic          :18   CA :114   Min.   :  10871
##  BC:25    British Columbia:25   CMA: 33   1st Qu.:  18429
##  MB: 5    North             : 2             Median :  40077
##  NB: 7    Ontario           :43             Mean   : 186632
##  NL: 4    Prairies          :31             3rd Qu.:  98388
##  NS: 5    Quebec            :28             Max.   :5583064
##  NW: 1
##  ON:43
etc.
```

Let's see what the dataset looks like in the scatterplot framework for 5 variables.

```
> pairs(can.2011[,c(7,9,10,11,12)])
```

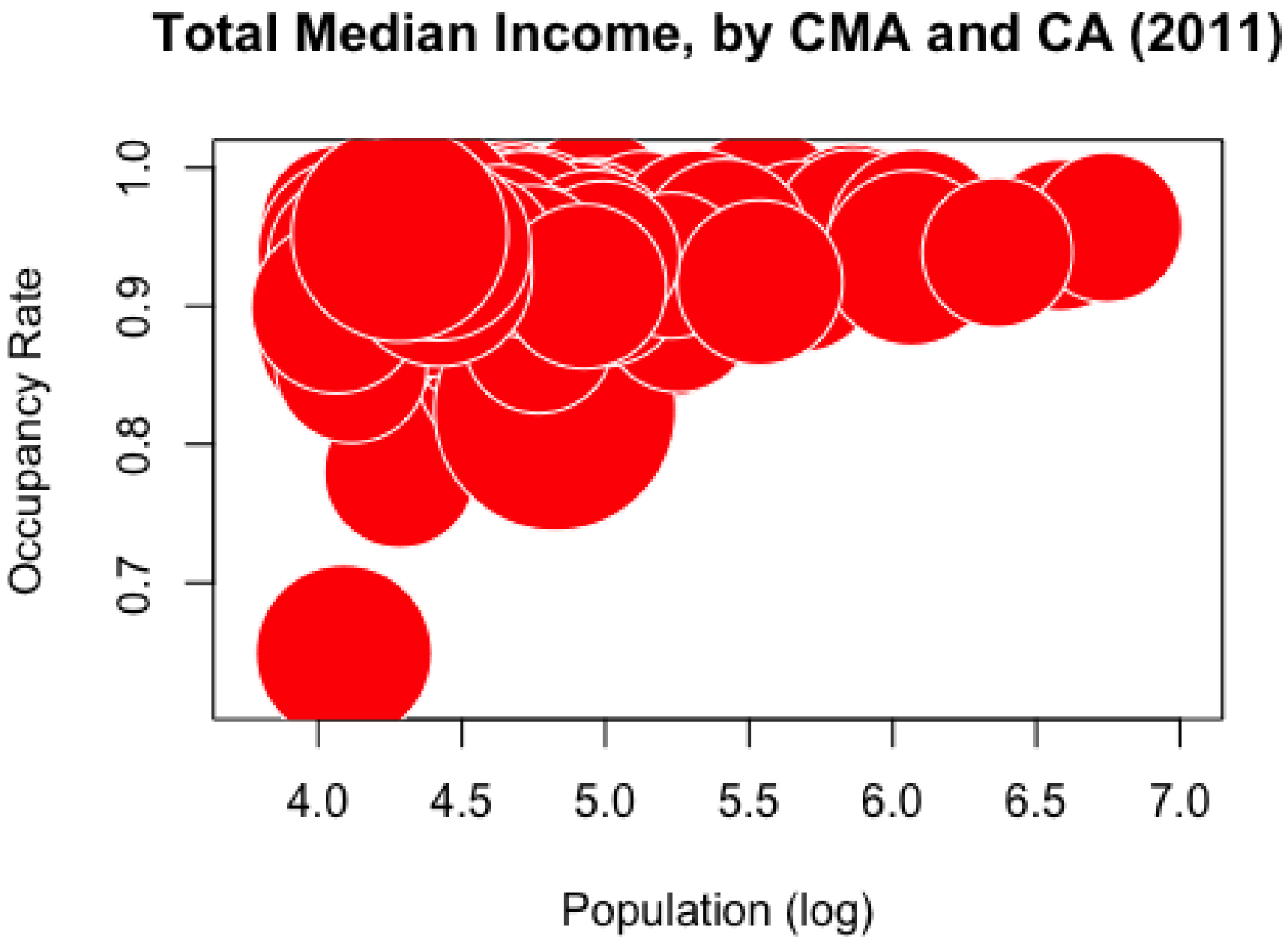

log_pop_2011
priv_dwell_2011
_private_dwell_2
occ_rate_2011
_total_income_2
0
1500000
0.65
0.85
4.0
6.0
0
2000000
0
2000000
0.65
0.90
4.0
5.5
0
1500000
30000
70000
30000

EXAMPLE: CMA & CA

It's ... not that interesting.

Can we deduct anything useful with bubble charts? We use median income as the radius for the bubbles, and focus on occupancy rates and population.

```
> radius.med.income.2011<-sqrt(can.2011$med_total_income_2011/pi)

> symbols(can.2011$log_pop_2011, can.2011$occ_rate_2011,
      circles=radius.med.income.2011, inches=0.45, fg="white", bg="red",
      xlab="Population (log)", ylab="Occupancy Rate")

> title("Total Median Income, by CMA and CA (2011)")
```

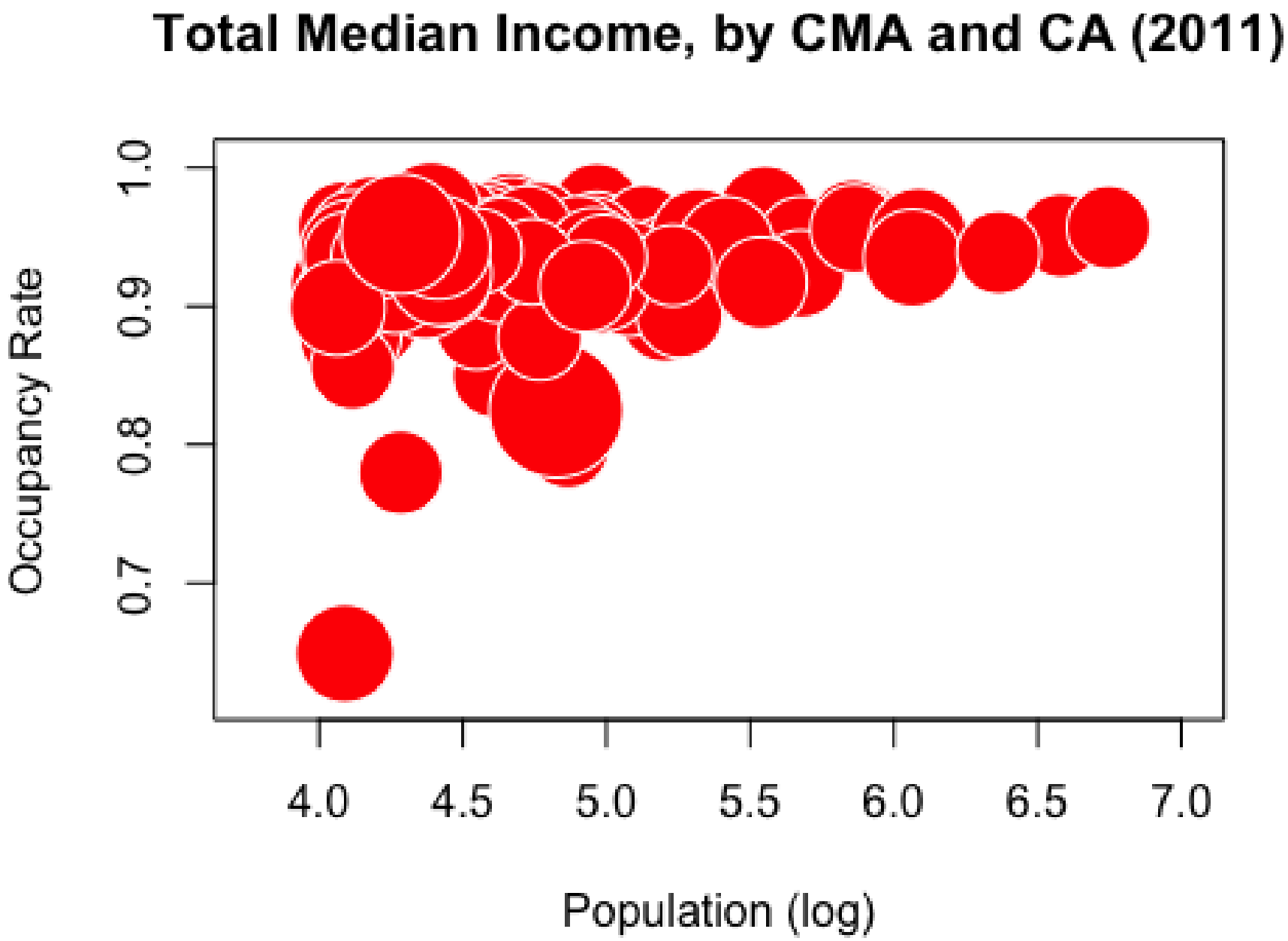

Total Median Income, by CMA and CA (2011)
Occupancy Rate
0.7
0.8
0.9
1.0
4.0
4.5
5.0
5.5
6.0
6.5
7.0
Population (log)

EXAMPLE: CMA & CA

Clearly, an increase in population seems to be associated with (and not necessarily a cause of) a rise in occupancy rates.

But the median income seems to have very little correlation with either of the other two variables. Perhaps such a correlation is hidden by the default unit used to draw the bubbles? Let's shrink it from 0.45 to 0.25 and see if anything pops out.

```
> symbols(can.2011$log_pop_2011, can.2011$occ_rate_2011,
      circles=radius.med.income.2011, inches=0.25, fg="white", bg="red",
      xlab="Population (log)", ylab="Occupancy Rate")

> title("Total Median Income, by CMA and CA (2011)")
```

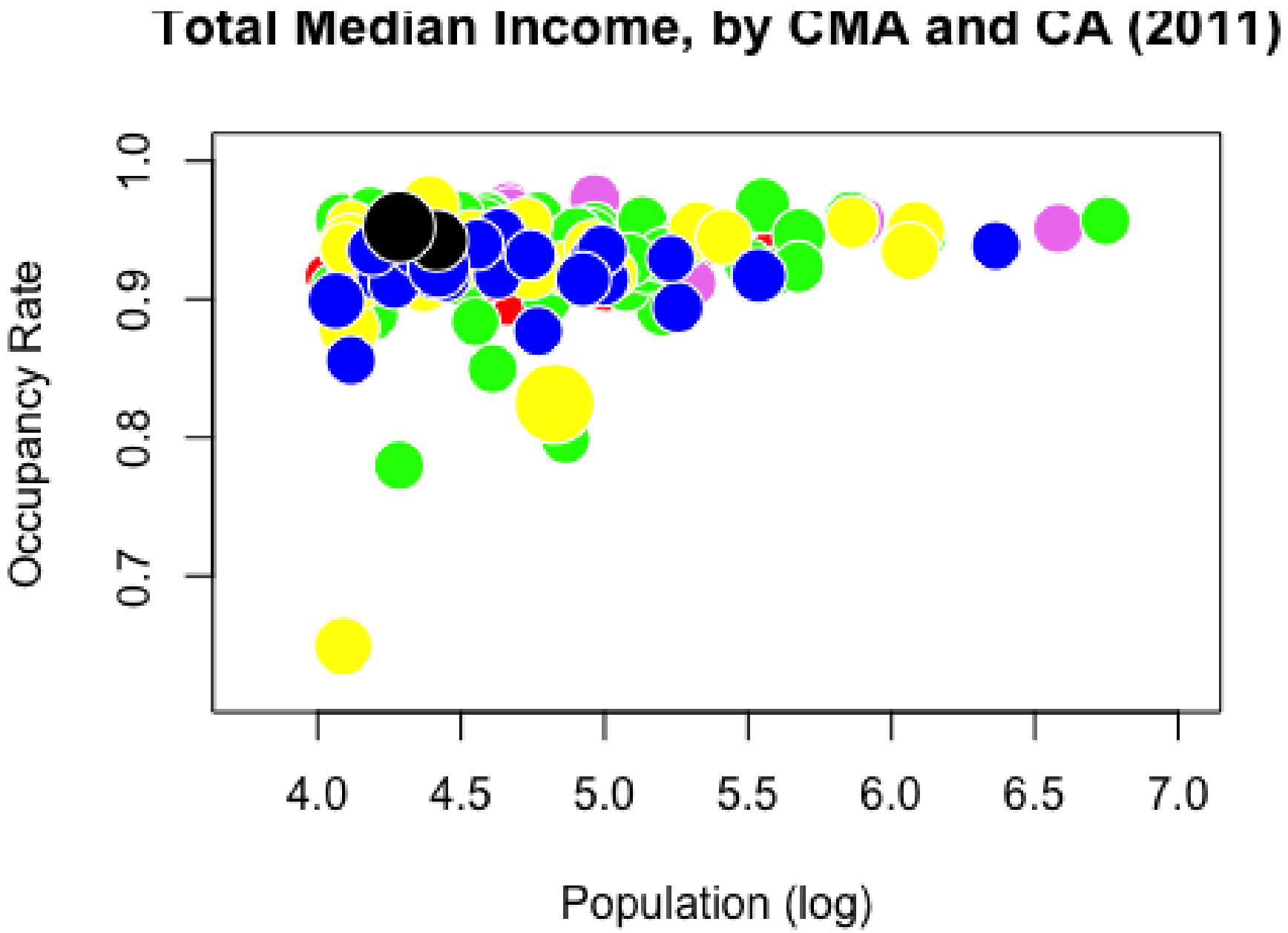

Total Median Income, by CMA and CA (2011)
Occupancy Rate
0.7
0.8
0.9
1.0
4.0
4.5
5.0
5.5
6.0
6.5
7.0
Population (log)

EXAMPLE: CMA & CA

Nothing doing.

But surely there would be a relationship in these quantities if we included the CMA/CA's region?

```
> symbols(can.2011$log_pop_2011, can.2011$occ_rate_2011,
      circles=radius.med.income.2011, inches=0.15, fg="white",
      bg=c("red","blue","black","green","yellow","violet")[can.2011$Region],
      xlab="Population (log)", ylab="Occupancy Rate")

> title("Total Median Income, by CMA and CA (2011)")
```


Total Median Income, by CMA and CA (2011)
Occupancy Rate
0.7
0.8
0.9
1.0
4.0
4.5
5.0
5.5
6.0
6.5
7.0
Population (log)

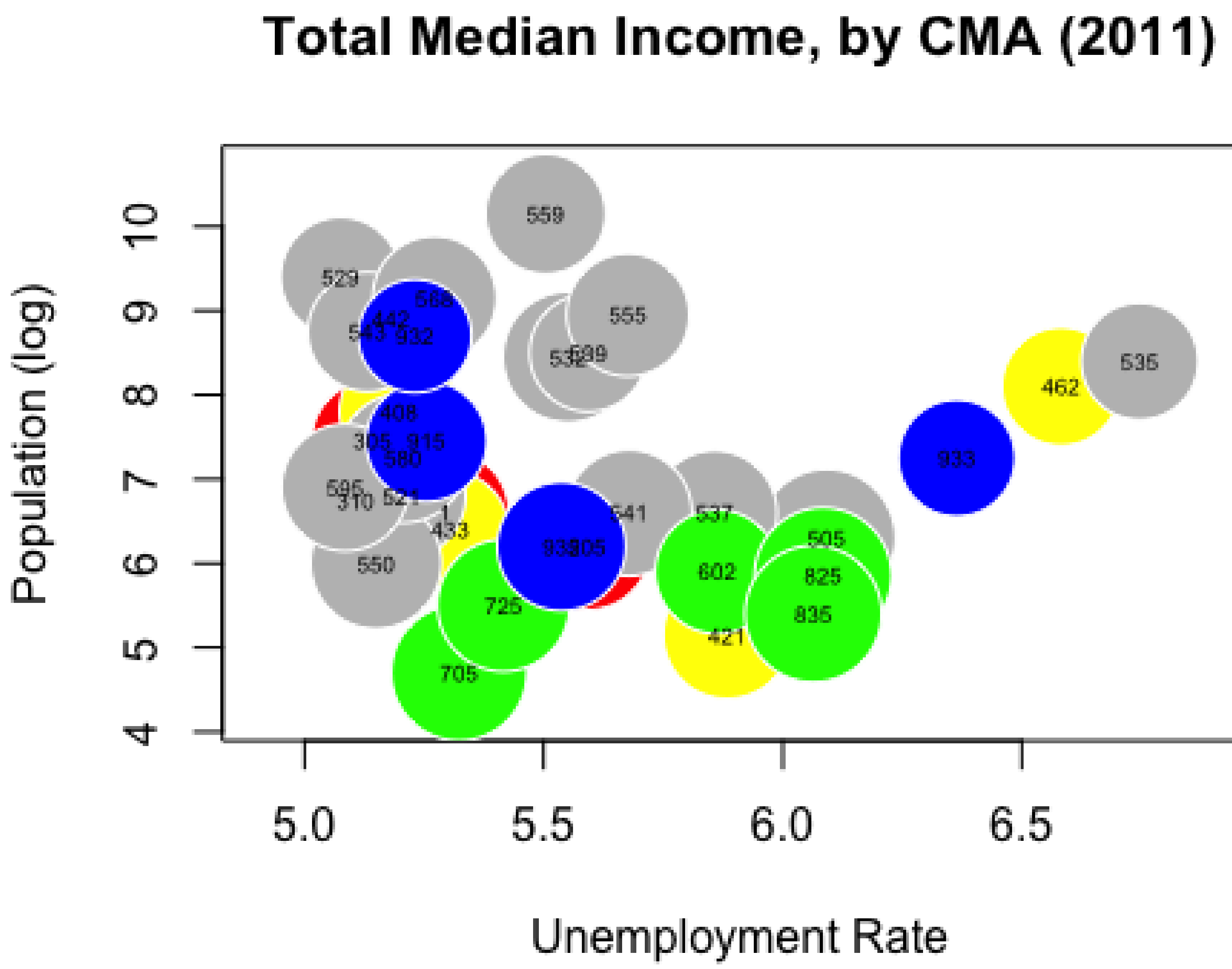
EXAMPLE: CMA & CA

Perhaps the CA distort the full picture (since they are smaller and more numerous). Let's stick to CMAs instead.

```
> can.2011.CMA=read.csv("../Data/Canada2011_CMA.csv", head=TRUE)
```

We 'll look at population and unemployment.

```
> radius.med.income.2011.CMA<-sqrt(can.2011.CMA$med_total_income_2011/pi)

> symbols(can.2011.CMA$log_pop_2011, can.2011.CMA$med_unemployment_2011,
      circles=radius.med.income.2011.CMA, inches=0.25, fg="white",
      bg=c("red","blue","gray","green","yellow")[can.2011.CMA$Region],
      ylab="Population (log)", xlab="Unemployment Rate")

> title("Total Median Income, by CMA (2011)")

> text(can.2011.CMA$log_pop_2011, can.2011.CMA$med_unemployment_2011,
      can.2011.CMA$Geographic.code, cex=0.5)
```


Total Median Income, by CMA (2011)
Population (log)
Unemployment Rate
4
5
6
7
8
9
10
5.0
5.5
6.0
6.5
559
529
588
442
543
932
555
599
462
535
408
305
915
580
933
310
521
433
541
537
505
550
602
825
725
835
421
705

EXAMPLE: CMA & CA

Part of the issue is that median income seems to be roughly uniform among CMAs.

What if we used rank statistics instead? Switch the radius to population rank, say?

```
> radius.pop.rank.2011.CMA<-sqrt(can.2011.CMA$pop_rank_2011/pi)

> symbols(can.2011.CMA$med_total_income_2011, can.2011.CMA$med_unemployment_2011,
      circles=radius.pop.rank.2011.CMA, inches=0.25, fg="white",
      bg=c("red","blue","gray","green","yellow")[can.2011.CMA$Region],
      ylab="Median Income", xlab="Unemployment Rate")

> title("Population Rank, by CMA and CA (2011)")

> text(can.2011.CMA$med_total_income_2011, can.2011.CMA$med_unemployment_2011,
      can.2011.CMA$Geographic.code, cex=0.5)
```

Sometimes, nothing useful comes out of a dataset. That's life.

Population Rank, by CMA and CA (2011)