Session 05

Robust and Resistant Regression

V&R § 6.5, p. 156 ff

A radical change of view

- In most data analysis contexts the data are considered the gold standard and models to describe the process giving rise to it are largely speculative.
- In some situations, however, the model is considered to be confidently known but the data may be questionable.
- In these cases we would like to use an estimation method that *retains high efficiency*, but is *not seriously damaged* if a fair proportion of the data is either in error, or wrongly included.

Strategies

- M estimators. Use a proxy score function with re-descending influence function.
  - Like MLE, but with a special likelihood ('M') designed to downweight observations which are inconsistent with the 'core of good data points', according to the assumed model.
  - t-robust estimators: use a t-distribution, low d.f.
- LTS estimators. Minimise the sum of squares of the smallest residuals in absolute size, typically the smallest half
- LQS estimators: Minimise some quantile of the squared residuals, typically the median (LMS).

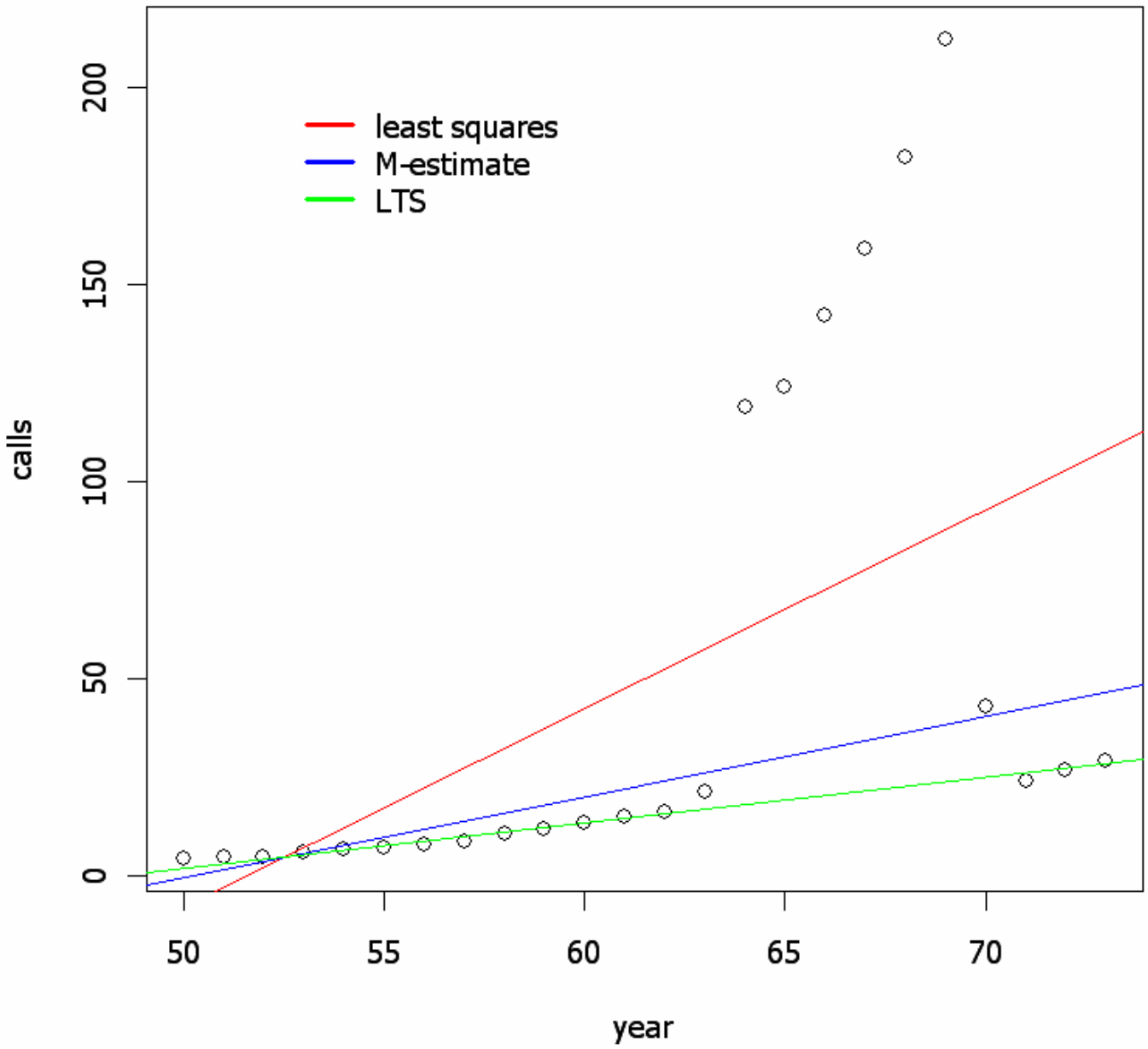
Some software

- **MASS** package:
  - **rlm** – M-estimator with sensible defaults
  - **lqs** – Least trimmed squares
  - **ltsreg** – front end to lqs(…, method = "lts")
  - **lmsreg** – front end to lqs(…, method = "lms")

- **robust** package:
  - **lmRob** – variable algorithm, like **rlm**
  - **glmRob** – limited options, problems?

An very bad example: the phones data

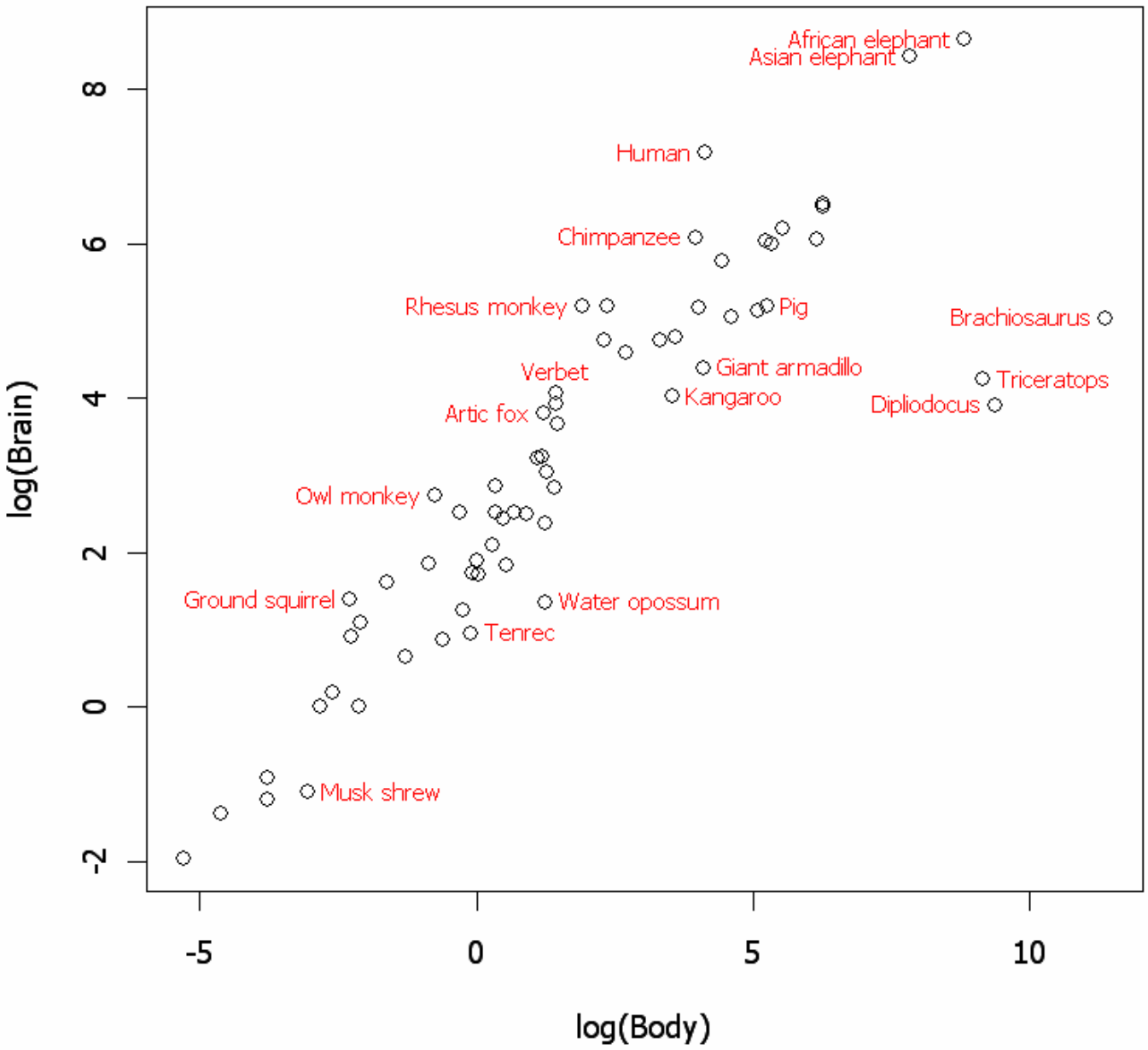
```
phones.lm <- lm(calls ~ year, data = phones)
phones.rlm <- rlm(calls ~ year, phones, maxit=50)
phones.lqs <- lqs(calls ~ year, phones)

with(phones, plot(year, calls))
abline(coef(phones.lm), col = "red")
abline(coef(phones.rlm), col = "blue")
abline(coef(phones.lqs), col = "green")

legend(52.5, 200,
  col = c("red", "blue", "green"),
  lwd = 2, bty = "n", legend =
  c("least squares", "M-estimate", "LTS"))
```


least squares
M-estimate
LTS
calls
year
0
50
100
150
200
50
55
60
65
70

A second artificial example: animals

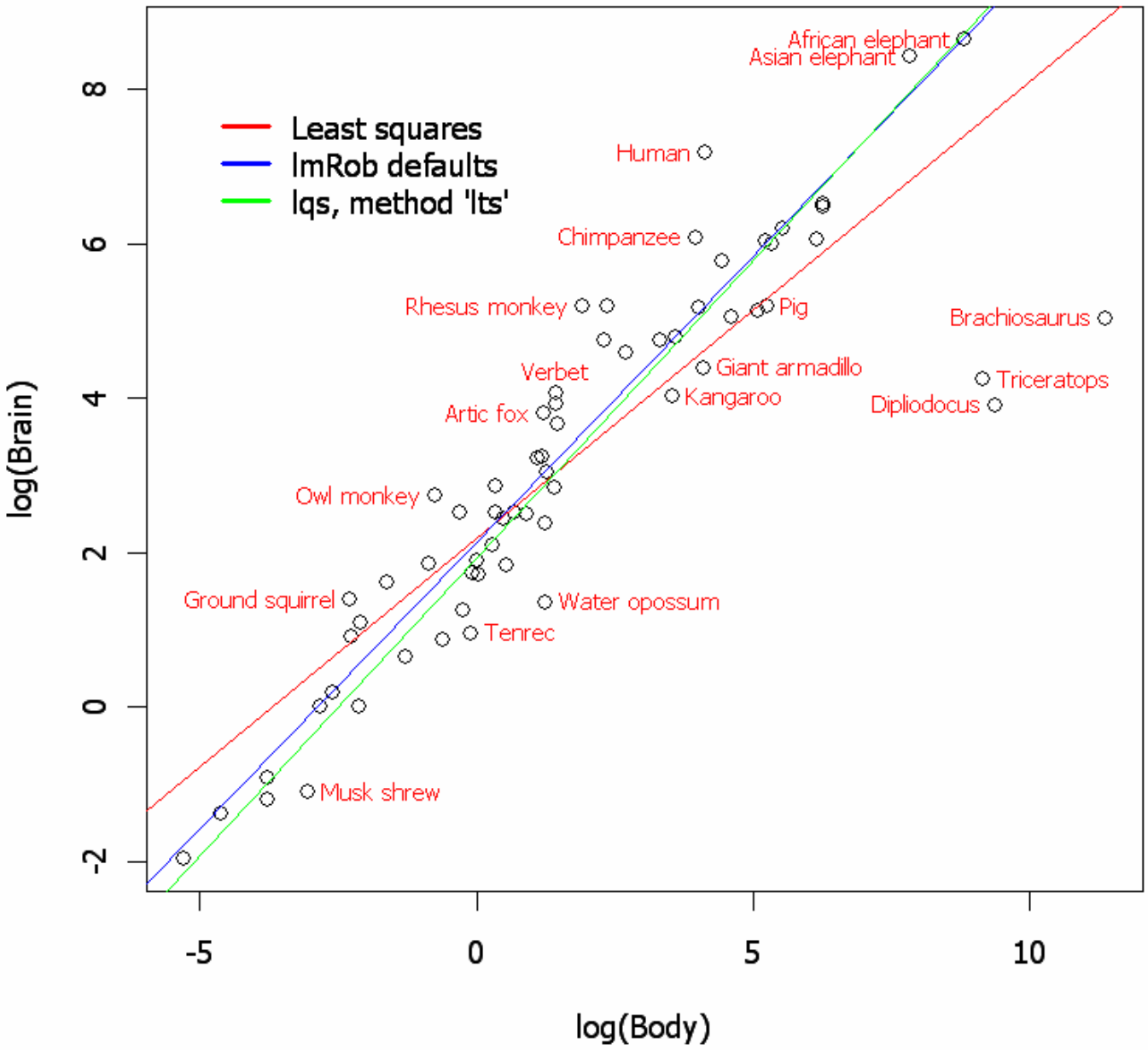
- The ‘animals’ data set is an extension of the one given in the MASS library. It contains average body (kg) and brain (gm) sizes for many modern mammal species, including human, but is corrupted by three dinosaurs.

```
with(animals, {
  plot(log(Body), log(Brain))
  identify(log(Body), log(Brain), Name,
     col = "red", cex = 0.75)
})
```


African elephant
Asian elephant
Human
Chimpanzee
Rhesus monkey
Pig
Brachiosaurus
Giant armadillo
Verbet
Triceratops
Kangaroo
Dipliodocus
Artic fox
Owl monkey
Ground squirrel
Water opossum
Tenrec
Musk shrew
log(Brain)
log(Body)
8
6
4
2
0
-2
-5
0
5
10

Compare different methods

```
require(robust)
BB.lm <- lm(log(Brain) ~ log(Body), animals)
BB.lmRob <- lmRob(log(Brain) ~ log(Body), animals)
BB.lqs <- lqs(log(Brain) ~ log(Body), animals)

abline(coef(BB.lm), col = "red")
abline(coef(BB.lmRob), col = "blue")
abline(coef(BB.lqs), col = "green")

legend(-5, 8,  c("Least squares", "lmRob defaults",
   "lqs, method 'lts'"),
   col = c("red", "blue", "green"), lwd = 2, bty = "n")
```

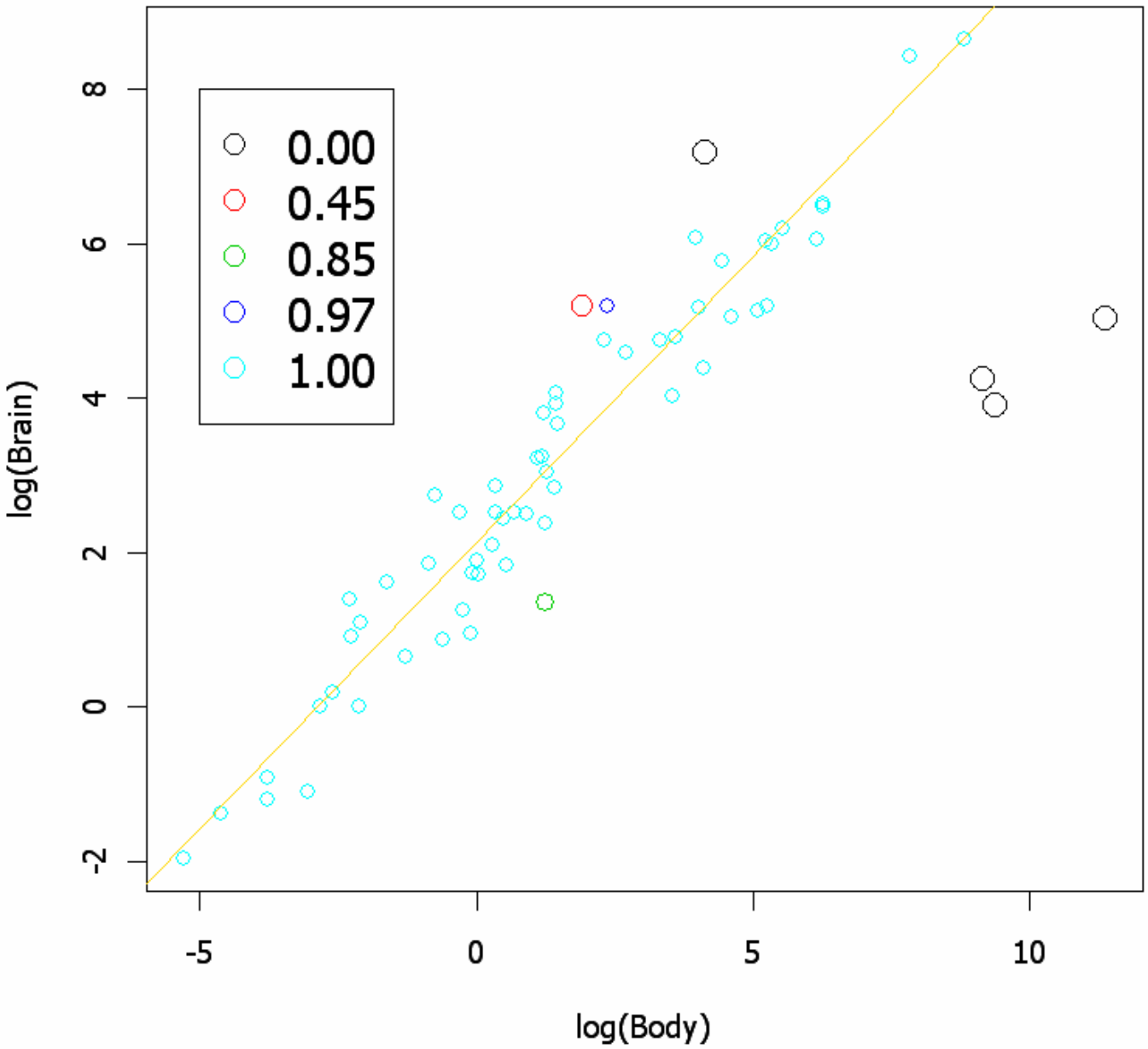

Least squares
lmRob defaults
lqs, method 'lts'
African elephant
Asian elephant
Human
Chimpanzee
Rhesus monkey
Pig
Brachiosaurus
Giant armadillo
Triceratops
Verbet
Kangaroo
Dipliodocus
Artic fox
Owl monkey
Ground squirrel
Water opossum
Tenrec
Musk shrew
log(Brain)
log(Body)
8
6
4
2
0
-2
-5
0
5
10

M-estimator weights

- The M-estimators do an iteratively weighted least squares fit. The weights themselves can be used to see what the algorithm considers to be the corrupt data.

```
wt <- format(round(BB.lmRob$M.weights, 2))
uwt <- unique(sort(wt))
wt <- match(wt, uwt)
cx <- 2 - wt/max(wt)

with(animals,  plot(log(Body), log(Brain), col = wt, cex =
   cx))
abline(BB.lmRob, col = "gold")
legend(-5, 8, uwt, pch=1, col = 1:length(uwt), cex = 1.5)
```

- Dinosaurs out, Humans out, Elephants in, Water Opossum 'mostly' in!



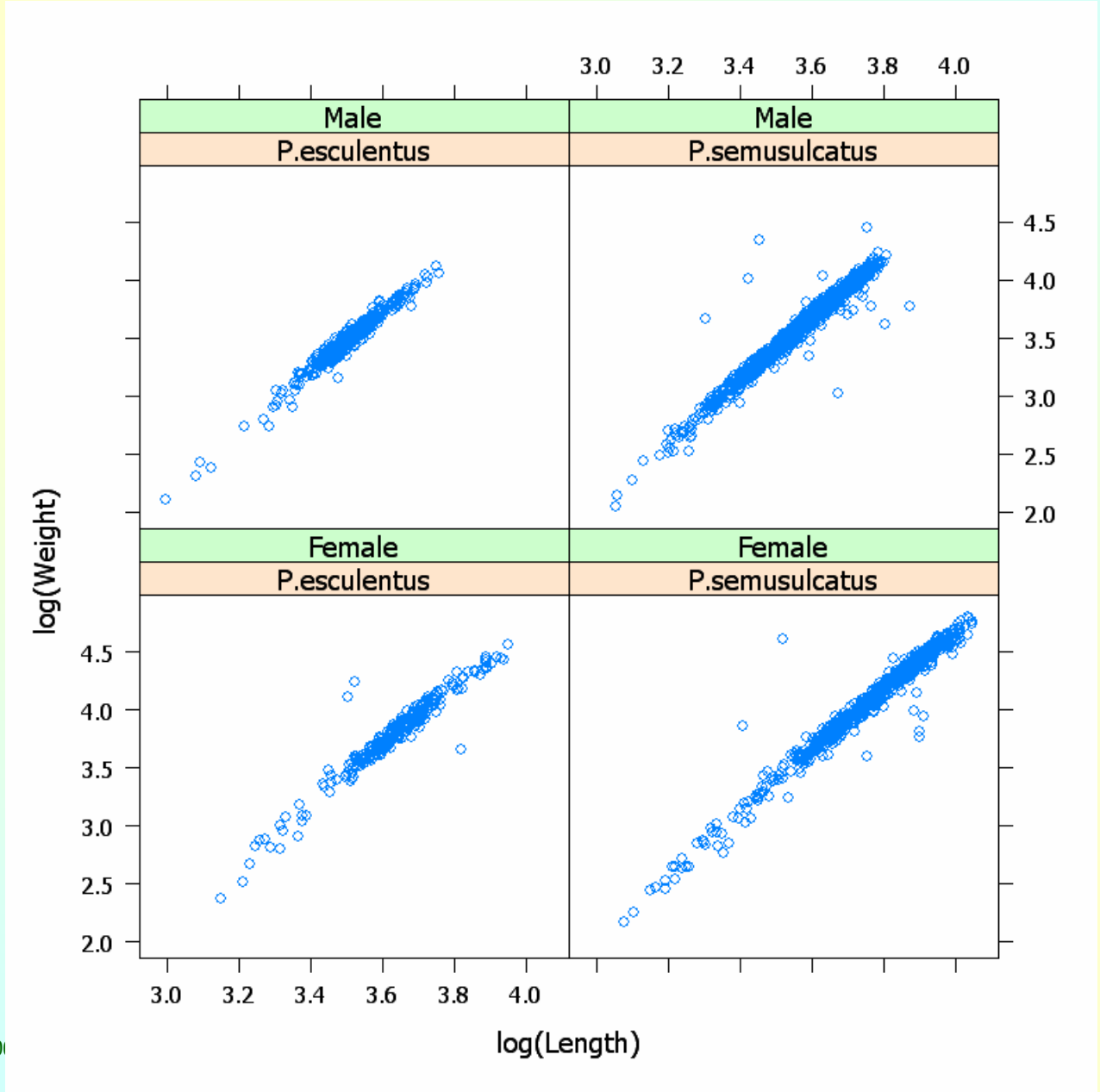
A real example: length-weight relationships

- The data consists of a sample of tiger prawns for which carapace length and weight are measured.
- There are two species, and males and females of both.
- The problem is to estimate the length-weight relationship curve
- Most morphometric relationships are linear on a log-log scale, that is the natural model is

$$W = \exp(\alpha + \beta \log L + \varepsilon)$$

- The form appears reasonable, but there are clearly a few very serious outliers, which are ‘clearly’ data errors.

```
xyplot(log(Weight) ~ log(Length) | Species*Sex, LWData)
```



```
LWData <- transform(LWData,
   SS = factor(paste(substring(Species, 1, 5),
                     substring(Sex,     1, 1), sep="-")))
require(nlme)
prawns.lm1 <- lmList(log(Weight) ~ log(Length) | SS, LWData,
                                 pool = FALSE)
round(summary(prawns.lm1)$coef, 4)
, , (Intercept)
        Estimate Std. Error  t value Pr(>|t|)
P.esc-F  -5.2690     0.1573 -33.5044        0
P.esc-M  -5.5458     0.0913 -60.7373        0
P.sem-F  -5.5358     0.0786 -70.4157        0
P.sem-M  -5.8169     0.0722 -80.5634        0

, , log(Length)
        Estimate Std. Error  t value Pr(>|t|)
P.esc-F   2.4920     0.0432  57.6254        0
P.esc-M   2.5748     0.0261  98.7613        0
P.sem-F   2.5512     0.0209 122.3265        0
P.sem-M   2.6348     0.0203 130.0823        0
```

- A second method for separate regressions is as follows:

```
prawns.lm2 <- lm(log(Weight) ~ SS/log(Length) - 1, LWData)
round(summary(prawns.lm2)$coef, 4)
```

	Estimate	Std. Error	t value	Pr(>|t|)
SSP.esc-F	-5.2690	0.1398	-37.6932	0
SSP.esc-M	-5.5458	0.1592	-34.8318	0
SSP.sem-F	-5.5358	0.0694	-79.7355	0
SSP.sem-M	-5.8169	0.0726	-80.1139	0
SSP.esc-F:log(Length)	2.4920	0.0384	64.8298	0
SSP.esc-M:log(Length)	2.5748	0.0455	56.6380	0
SSP.sem-F:log(Length)	2.5512	0.0184	138.5169	0
SSP.sem-M:log(Length)	2.6348	0.0204	129.3565	0

- Pay particular attention to the standard errors as they govern the confidence intervals for the predictions.

The robust alternative

```
prawns.lmR <- lmRob(log(Weight) ~ SS/log(Length) - 1, LWData)
round(summary(prawns.lmR, cor = F)$coef, 4)
```

	Value	Std. Error	t value	Pr(>|t|)
SSP.esc-F	-5.1375	0.0904	-56.8543	0
SSP.esc-M	-5.4123	0.0948	-57.0805	0
SSP.sem-F	-5.6007	0.0432	-129.6468	0
SSP.sem-M	-5.9924	0.0420	-142.8019	0
SSP.esc-F:log(Length)	2.4564	0.0248	99.0861	0
SSP.esc-M:log(Length)	2.5373	0.0271	93.7844	0
SSP.sem-F:log(Length)	2.5702	0.0114	224.7061	0
SSP.sem-M:log(Length)	2.6847	0.0118	228.0719	0

```
Residual standard error: 0.0442379 on 2245 degrees of freedom
Multiple R-Squared: 0.818486

Test for Bias:
            statistic   p-value
M-estimate   12.07414 0.1479263
LS-estimate 676.06889 0.0000000
```

Test for bias

- Can be done directly

```
test.lmRob(prawns.lmR)
             statistic   p-value
M-estimate    12.07414 0.1479263
LS-estimate 676.06889 0.0000000
```

- Message:
  - The least squares are very biased
  - The M-estimates are probably not!

Yohai, V., Stahel, W. A., and Zamar, R. H. (1991). A procedure for robust estimation and inference in linear regression, in Stahel, W. A. and Weisberg, S. W., Eds., *Directions in robust statistics and diagnostics, Part II*. Springer-Verlag.

Which values have been downweighted?

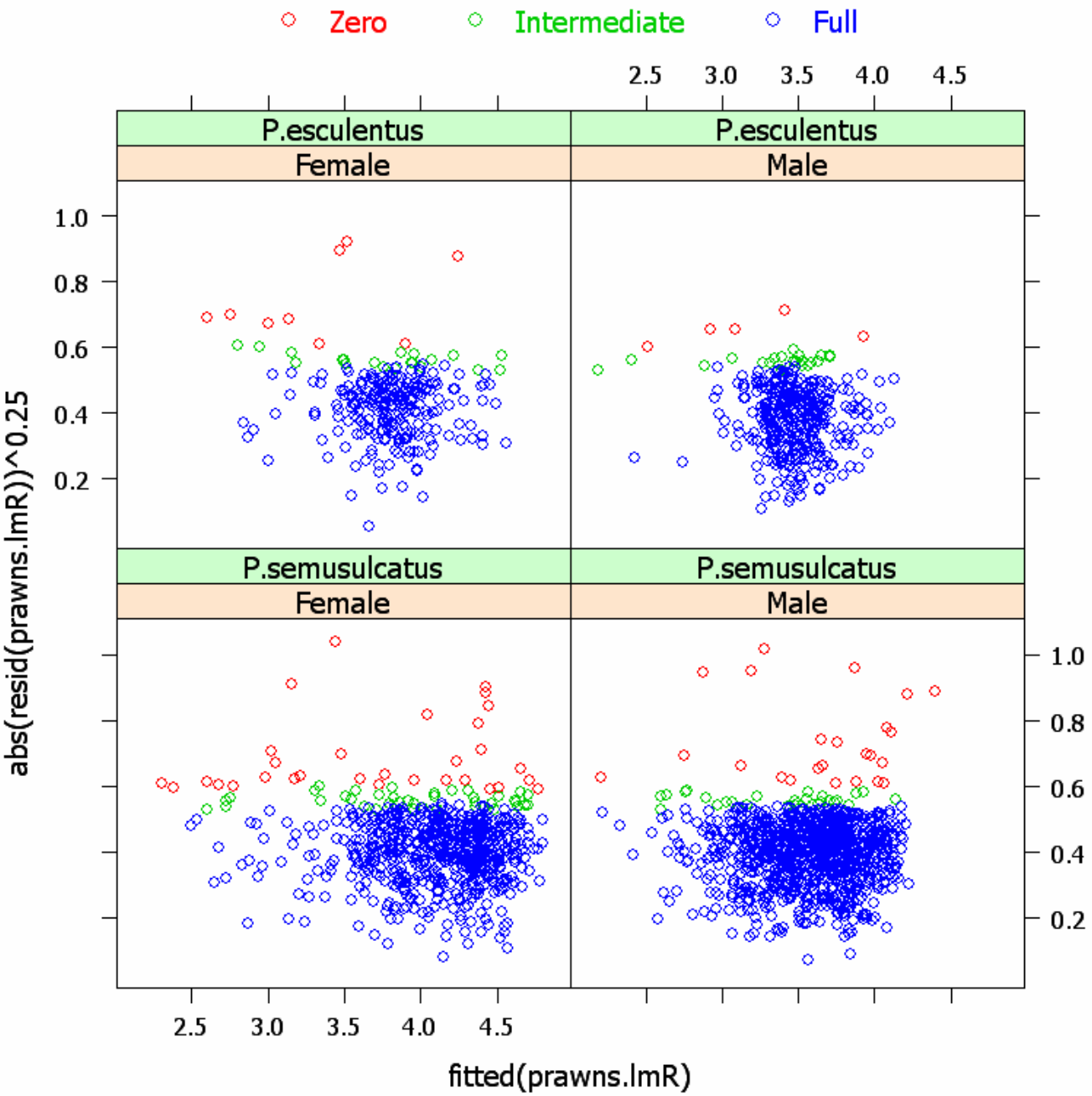
- The standard errors have been approximately halved.
- Which observations are considered suspect by the procedure.
- Difficult to appreciate because of the large sample size, but we can confirm that they correspond to the outlying residuals, at least:

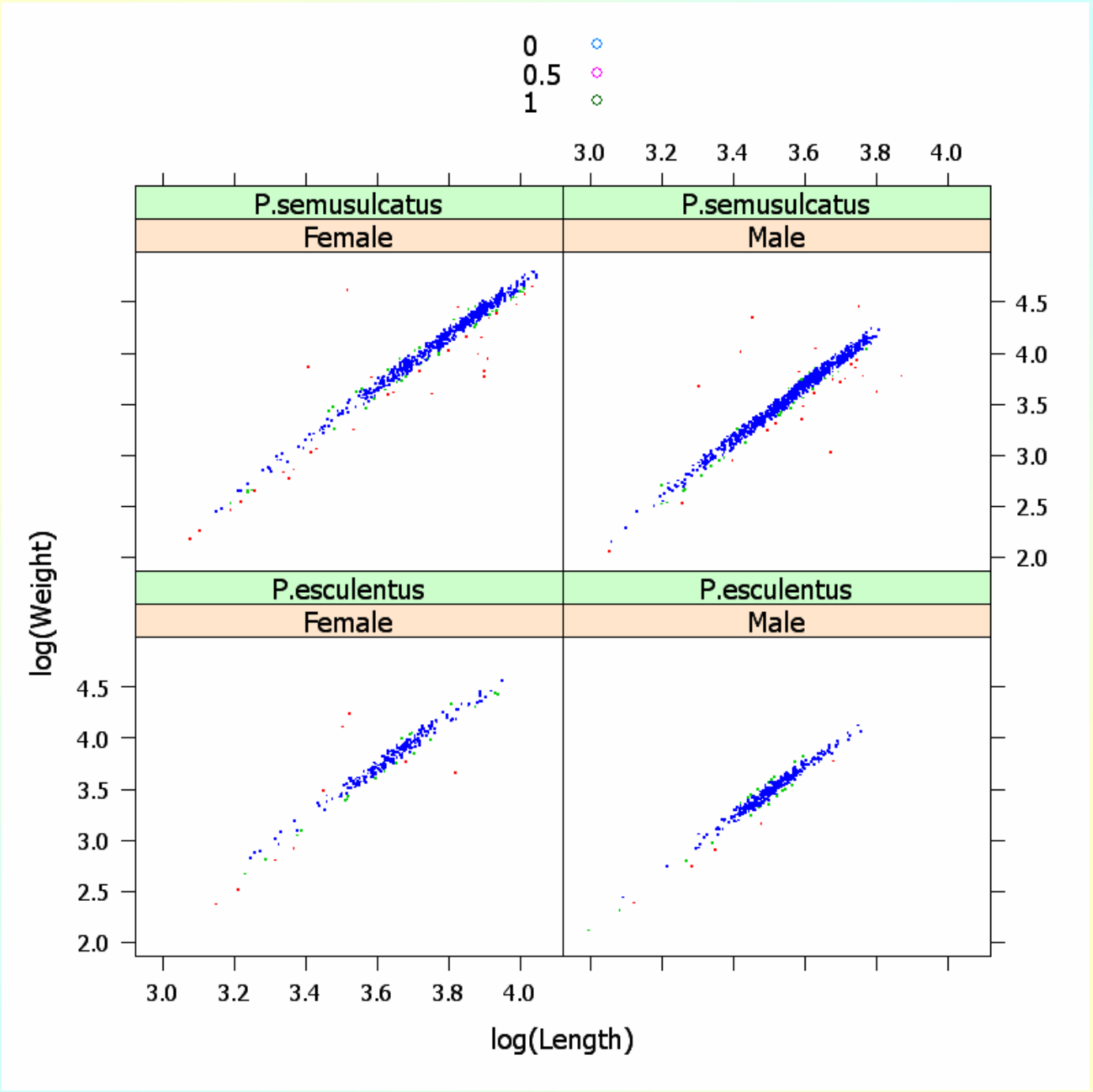
```
wt <- prawns.lmR$M.weights
wtf <- factor(ifelse(wt == 0, 0,
  ifelse(wt < 1, 0.5, 1)))
table(wtf)
```

```
wtf
 0  0.5    1
70  110 2073
```

A display of the results

```
palette("default")  # restore default colours
xyplot(abs(resid(prawns.lmR))^0.25 ~
   fitted(prawns.lmR) | Sex*Species,
   LWData, groups = wtf, as.table = T,  col = 2:4,
   key = list(columns = 3, background = 0, points
   = list(pch = 1, col = 2:4),
   text = list(c("Zero", "Intermediate", "Full"),
   col = 2:4)))
# a more conventional display
xyplot(log(Weight) ~ log(Length) | Sex*Species,
   LWData, groups = wtf,
  pch = ".", col = 2:4,cex = 1.5,auto.key = TRUE)
```


Zero
Intermediate
Full
P.esculentus
Female
P.esculentus
Male
P.semusulcatus
Female
P.semusulcatus
Male
abs(resid(prawns.lmR))^0.25
fitted(prawns.lmR)
1.0
0.8
0.6
0.4
0.2
2.5
3.0
3.5
4.0
4.5


0
0.5
1
P.semusulcatus
Female
P.semusulcatus
Male
P.esculentus
Female
P.esculentus
Male
log(Weight)
log(Length)
3.0
3.2
3.4
3.6
3.8
4.0
2.0
2.5
3.0
3.5
4.0
4.5

Using the t-distribution

- The t-distribution is "normal in the middle" but has very fat tails for low degrees of freedom
- An old suggestion has been to use the t-distribution itself with small (e.g. 5) degrees of freedom to give a robust alternative likelihood to the Normal
- One advantage is that the method easily adapts to non-linear regressions as well.
- The function **`tRob`** (given here), and methods, is used like **`lmRob`** or **`lm`**

- K. L. Lange, R. J. A. Little and J. M. G. Taylor. (1989) "Robust Statistical Modeling Using the t Distribution". *Journal of the American Statistical Association,* **84**, 881-896.

A simple demonstration

- A function to fit the t-distribution in a simple case

```
fitT <- local({
  obj <- function(par, X) {
    mu <- par[1]
    sg <- exp(par[2])
    -sum(dt((X-mu)/sg, 3, log=TRUE) - par[2])
  }
  function(x, nu = 3) {
    p <- optim(c(0,0), obj, method = "BFGS",
                    X = x)$par
    c(mu = p[1], sg = exp(p[2])*sqrt(nu/(nu-2)))
  }
})
```

Generate and record the data

```
x <- rnorm(30)
sink("x.txt")
print(round(sort(x), 2))
sink()

th <- theta <- NULL
mx <- max(x)
x <- sort(x)[-length(x)]
```

The main loop

```
for(x0 in mx + 0:20/2) {
  y <- c(x, x0)
  th <- rbind(th, c(x0=x0, fitT(y, 5)))
  theta <- rbind(theta, c(x0=x0, mu = mean(y), sg
  = sd(y)))
}
row.names(th) <- paste("T", 1:nrow(th))
row.names(theta) <- paste("N", 1:nrow(theta))
est <- rbind(data.frame(th,    Est = "t"),
             data.frame(theta, Est = "N"))
png(filename = "Sim_%03d.png", height = 600,
  width = 800)
```

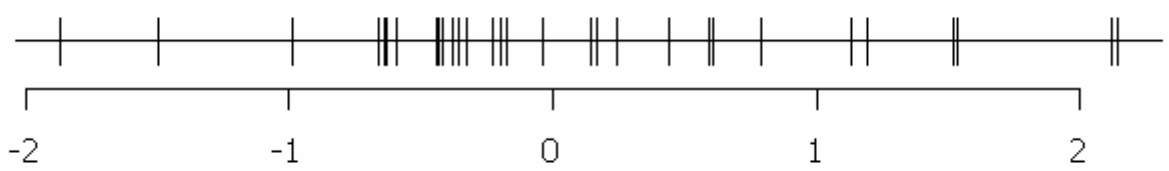
Present the results

```
print(xyplot(sg ~ mu, est, subset = Est == "t",
   type = "b", xlab = expression(hat(mu)),
  ylab = expression(hat(sigma)), pch=15))

print(xyplot(sg ~ mu, est, groups = Est, type =
   "b", xlab = expression(hat(mu)),
  ylab = expression(hat(sigma)), pch=15))

dev.off()
```

The data – file x.txt

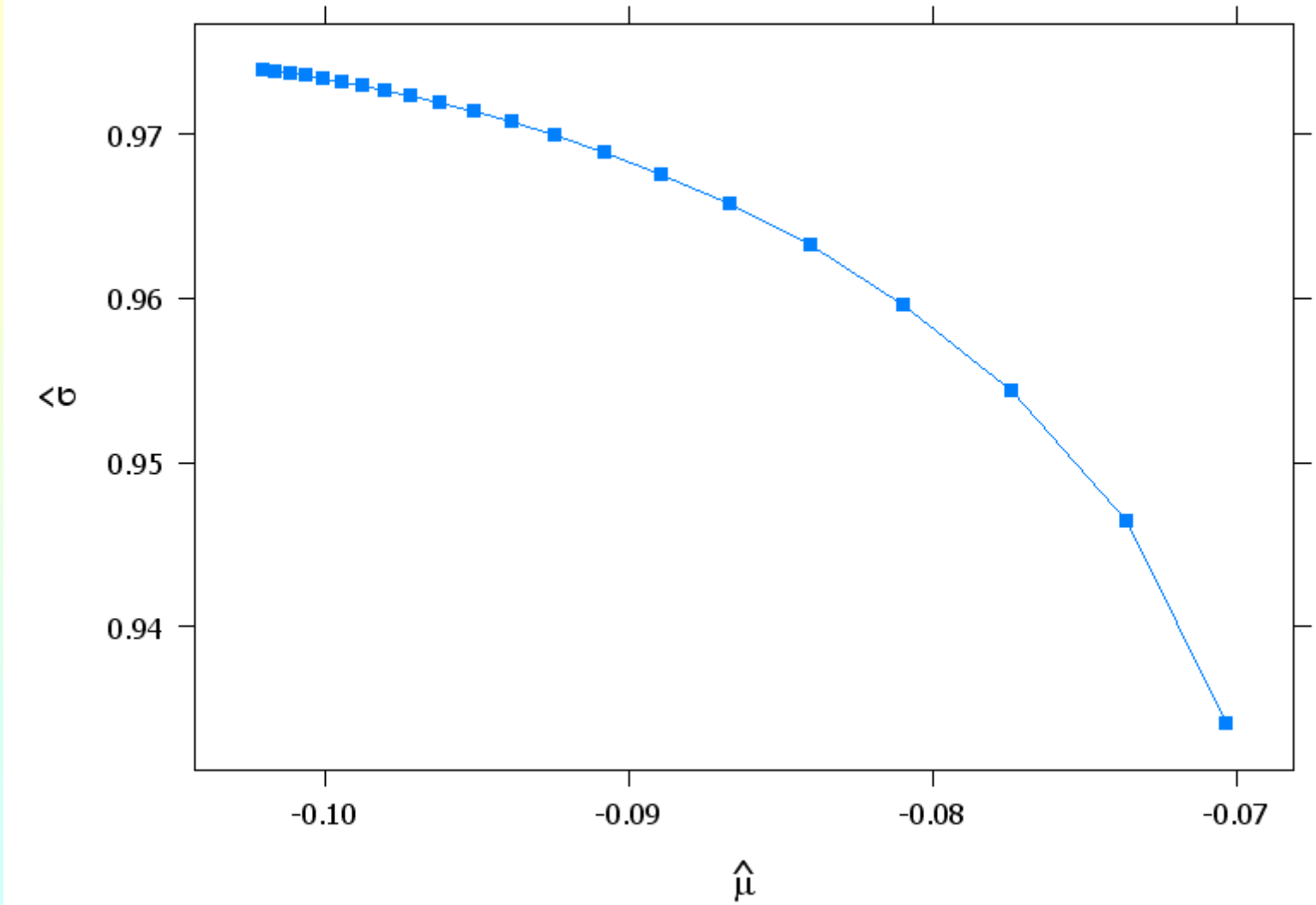
```
-1.82 -1.57 -1.09 -0.96 -0.82 -0.80
-0.79 -0.60 -0.58 -0.55 -0.51 -0.47
-0.46 -0.21 -0.18  0.00  0.09  0.17
 0.27  0.28  0.28  0.47  0.62  0.83
 0.95  1.06  1.06  1.17  1.49  1.55
```

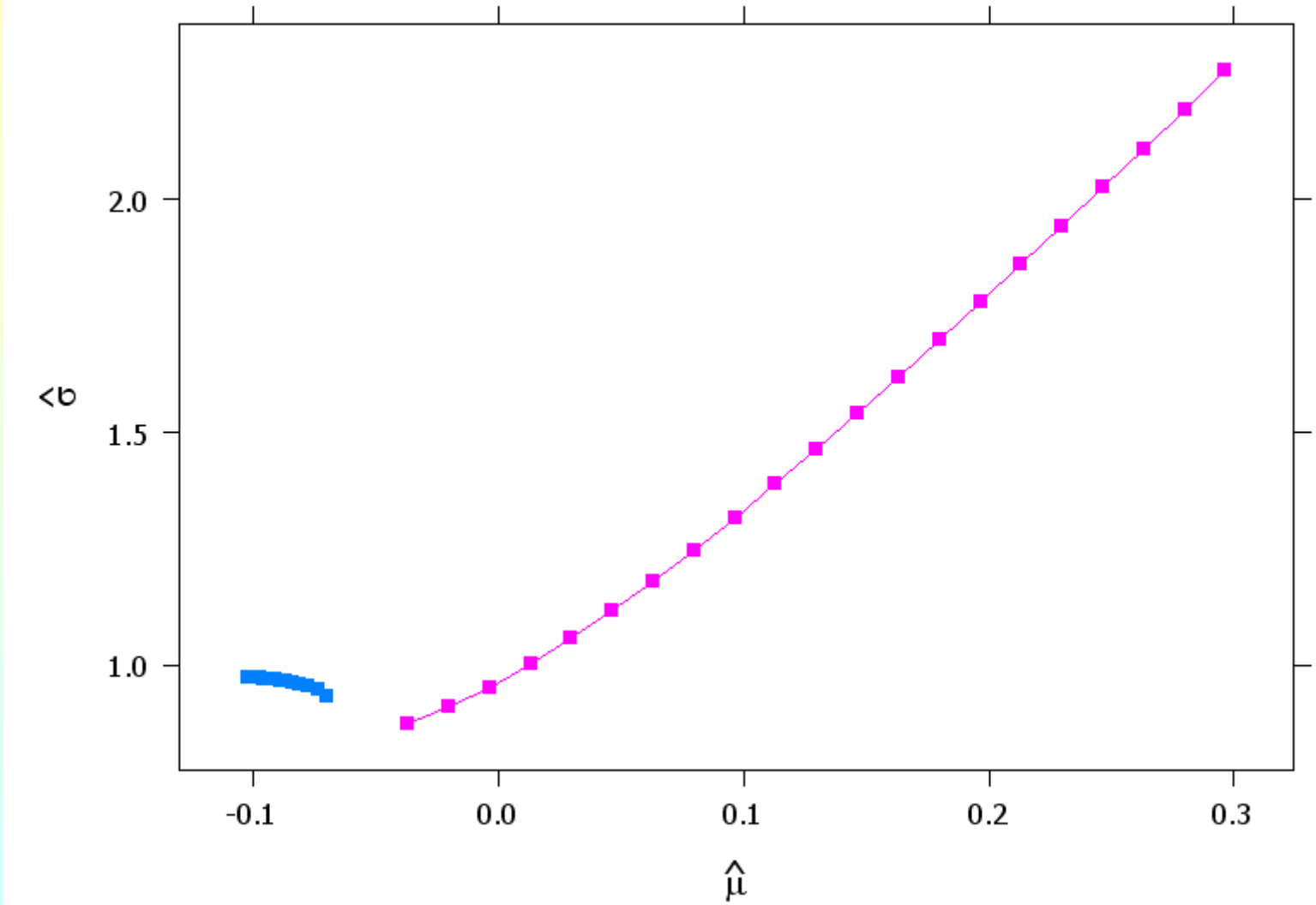
A N(0,1) sample of size 30



sample mean = 0.0909
sample sd = 0.9687

```
-1.87 -1.50 -0.99 -0.66 -0.64 -0.63 -0.60 -0.44 -0.43 -0.42
-0.38 -0.36 -0.33 -0.23 -0.20 -0.17 -0.04  0.15  0.16  0.24
 0.44  0.59  0.61  0.79  1.13  1.19  1.52  1.54  2.12  2.14
```





A quick comparison

```
prawns.tRob <- tRob(log(Weight) ~ SS/log(Length) - 1, LWData)
summary(prawns.tRob)

Call:
tRob(formula = log(Weight) ~ SS/log(Length) - 1, data =
   LWData)

Parameter estimates:
                       Parameter         SE
SSP.esc-F              -5.284156 0.10095249
SSP.esc-M              -5.450040 0.09914016
SSP.sem-F              -5.647413 0.04572155
SSP.sem-M              -5.987124 0.04289421
SSP.esc-F:log(Length)   2.496342 0.02772038
SSP.esc-M:log(Length)   2.548039 0.02827280
SSP.sem-F:log(Length)   2.582238 0.01209744
SSP.sem-M:log(Length)   2.682941 0.01203543
log(sigma)             -3.212614 0.01901083
```

Three methods

```
cbind(Least_sq = coef(prawns.lm2),
      Robust = coef(prawns.lmR),
      T_robust = coef(prawns.tRob))
```

	Least_sq	Robust	T_robust
SSP.esc-F	-5.269048	-5.137539	-5.284156
SSP.esc-M	-5.545795	-5.412274	-5.450040
SSP.sem-F	-5.535809	-5.600741	-5.647413
SSP.sem-M	-5.816874	-5.992377	-5.987124
SSP.esc-F:log(Length)	2.492025	2.456420	2.496342
SSP.esc-M:log(Length)	2.574800	2.537290	2.548039
SSP.sem-F:log(Length)	2.551181	2.570196	2.582238
SSP.sem-M:log(Length)	2.634783	2.684658	2.682941

Final notes

- Robust methods are only appropriate if, a priori, the model can be reliably identified but the data cannot be considered reliable
- The main effect of using a robust method may be to provide reliable estimates of confidence intervals and error variance rather than coefficient estimates
- A robust method is always a compromise between efficiency and breakdown point. The better methods, such as “MM”, can be computationally expensive.
- With re-descending M-estimators it can be difficult to find a good starting value. Try several to be sure.
- The t-robust alternative is useful for some situations.