R Session 6

Some Functions That Support Fitting

William A. Cooper

Research Aviation Facility, Earth Observing Laboratory
National Center for Atmospheric Research

Presentation on 11/21/2014

COMMON USES FOR FITTING AND REGRESSION

1. Represent some variable in terms of other measurements
   (a) PCors as functions of Mach number, angle of attack, etc.
   (b) Angle of attack as a function of ADIFR, QCF, etc.
   (c) Nusselt number as a function of Reynolds number for King Probe
   (d) Recovery factor as a function of Mach number
2. Relationship between measurements (e.g., intercomparison)
3. Determining the best use of overconstrained measurements (e.g., 4-beam LAMS)
4. Finding best-fit smoothed distributions, e.g., for drop or particle size measurements.
5. Fitting to CCN measurements to find $\{C, k\}$ in $N = C\,S^k$.
6. Fitting circles to 2D images (best estimate of raindrop sizes).

THE R FUNCTION *lm* ()

Meets most of the preceding needs

- "linear model" – linear in the coefficients. These are OK:
  $(V = A + Bx^3 + C\ln(x) + D\sin y + E\,f(x,y,z)$
- R call: F <- *lm*
  (V~I(x^3)+I(ln(x))+sin(y)+I(f(x,y,z))+exp(x))
- not linear: $V = a\exp(bx)$ – but $v' = \ln V = a' + bx$ is linear
- If you use *lm* (V~A*B) this will be interpreted as
  V~A+B+AB so use I( ), even for x^2: I(x^2)

Simple example:

Fitting to a speed run to find $\alpha = c_0 + c_1$(ADIFR/QCF)

Method: $\alpha_{Ref} = \theta - w_p/V \sim$ I(ADIFR/QCF)

comment re terminology

you fit an equation to data (not fit data)

USE DEEPWAVE SPEED RUN

R code to get speed-run data:

```
require(Ranadu, quietly = TRUE, warn.conflicts=FALSE)
## Loading required package:  methods
Project <- "DEEPWAVE"
Flight <- "rf15"          # this was the flight with cal maneuvers
fname = sprintf("%s%s/%s%s.nc", DataDirectory (),
                Project, Project, Flight)
VarNames <- c("TASX", "ADIFR", "PITCH", "QCF", "GGVSPDB")
D1 <- getNetCDF (fname, VarNames, F=15)
r <- c(setRange (D1$Time, 32100, 32900),      # 12,500 ft
       setRange (D1$Time, 41500, 42300),      # FL200
       setRange (D1$Time, 50100, 51100))      # FL 300
Flight <- "rf11"          # this had  cal maneuvers at 40K ft
fname = sprintf("%s%s/%s%s.nc", DataDirectory (),
                Project, Project, Flight)
D2 <- getNetCDF (fname, VarNames, Start=103000, End=104000, F=11)
## construct data.frame with speed-run data
DSR <- merge (D1[r, ], D2, all=TRUE)
```

FIT TO DSR DATA

Construct reference, then fit to it

```
DSR <- DSR[(DSR$TASX > 130), ]      # eliminate very slow measurements
Cradeg <- pi / 180
DSR["AOAREF"] <- DSR$PITCH - (DSR$GGVSPDB / DSR$TASX) / Cradeg
FitSR <- lm (AOAREF ~ I(ADIFR/QCF), data=DSR, na.action=na.omit)
```

FIT TO DSR DATA

Construct reference, then fit to it

```
DSR <- DSR[(DSR$TASX > 130), ]      # eliminate very slow measurements
Cradeg <- pi / 180
DSR["AOAREF"] <- DSR$PITCH - (DSR$GGVSPDB / DSR$TASX) / Cradeg
FitSR <- lm (AOAREF ~ I(ADIFR/QCF), data=DSR, na.action=na.omit)
```

HERE IS THE RESULT SUMMARIZED FROM FitSR:

```
summary (FitSR)


##
## Call:
## lm(formula = AOAREF ~ I(ADIFR/QCF), data = DSR, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50691 -0.06594  0.01899  0.07371  0.41546
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.396942   0.005256   836.6   <2e-16 ***
## I(ADIFR/QCF) 21.001024   0.064391   326.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1211 on 2017 degrees of freedom
## Multiple R-squared:  0.9814, Adjusted R-squared:  0.9814
## F-statistic: 1.064e+05 on 1 and 2017 DF,  p-value: < 2.2e-16
```

FIT TO DSR DATA

Construct reference, then fit to it

```
DSR <- DSR[(DSR$TASX > 130), ]      # eliminate very slow measurements
Cradeg <- pi / 180
DSR["AOAREF"] <- DSR$PITCH - (DSR$GGVSPDB / DSR$TASX) / Cradeg
FitSR <- lm (AOAREF ~ I(ADIFR/QCF), data=DSR, na.action=na.omit)
```

HERE IS THE RESULT SUMMARIZED FROM FitSR:

```
summary (FitSR)


##
## Call:
## lm(formula = AOAREF ~ I(ADIFR/QCF), data = DSR, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50691 -0.06594  0.01899  0.07371  0.41546
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.396942   0.005256   836.6   <2e-16 ***
## I(ADIFR/QCF) 21.001024   0.064391   326.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1211 on 2017 degrees of freedom
## Multiple R-squared:  0.9814, Adjusted R-squared:  0.9814
## F-statistic: 1.064e+05 on 1 and 2017 DF,  p-value: < 2.2e-16
```

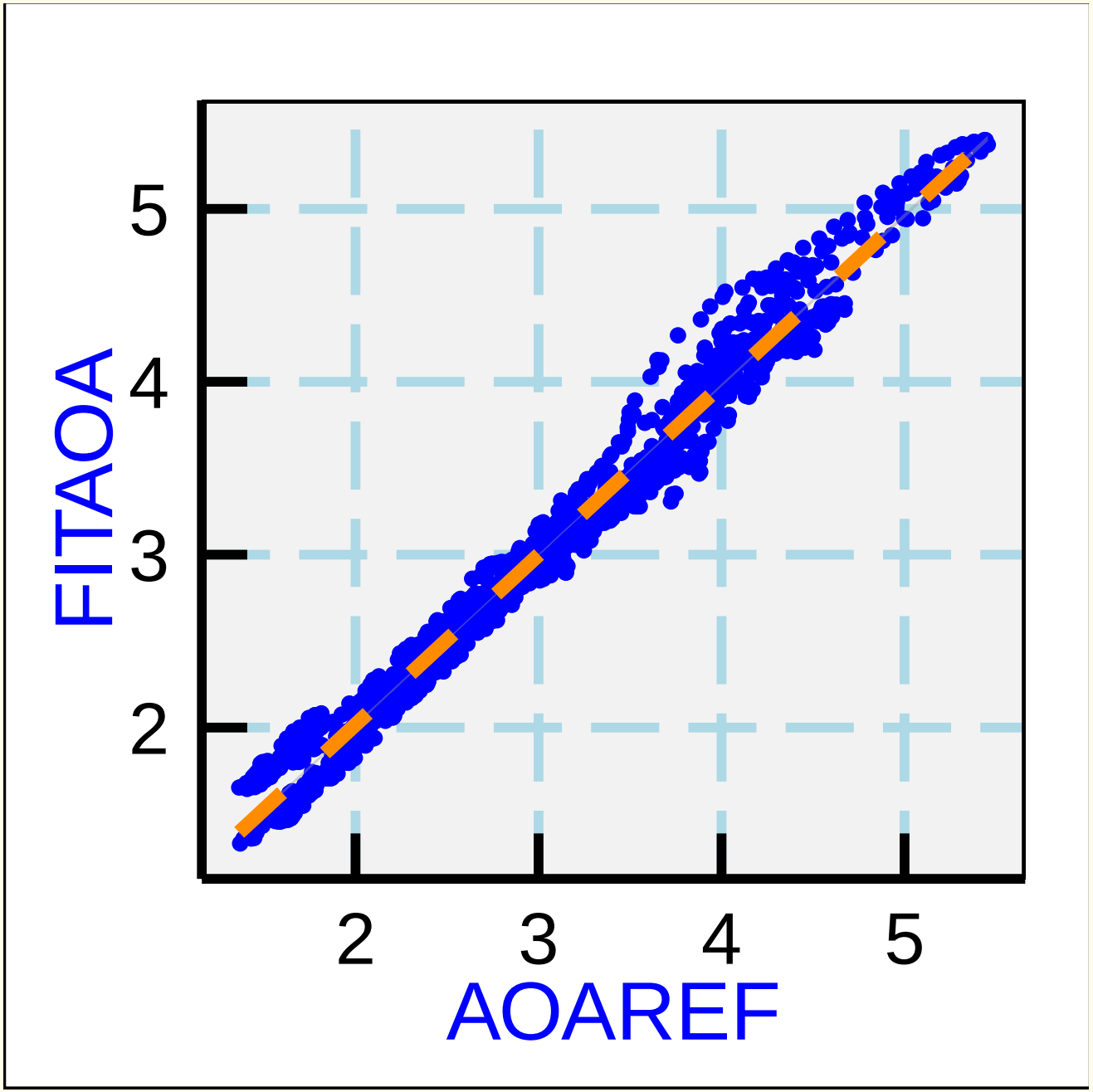
Result: if RATIO=ADIFR / QCR,
$\alpha = 4.397 + 21.00 * \text{RATIO}$
residual error 0.12°

LOOK AT THE RESULTS: (orange line is the fit)

Code to generate a plot:

```
DSR["FITAOA"] <- coef(FitSR)[1]+coef(FitSR)[2]*DSR$ADIFR/DSR$QCF
ggplot (data=DSR, aes(x=AOAREF, y=FITAOA)) +
  geom_point (pch=20, color='blue') +
  geom_smooth (method="lm", color='darkorange', lwd=1.5, lty=2) +
  theme_WAC ()
```

LOOK AT THE RESULTS: (orange line is the fit)



OTHER RESULTS:

PCOR functions based on LAMS:

$$\frac{\Delta p}{p} = a_0 + a_1 \frac{q}{p} + a_2 M^3 + a_3 \frac{\alpha}{a_r}$$

Δp provided by LAMS
Parameterized representation of Δp then allows use of the LAMS result when LAMS.
is not present or not operational

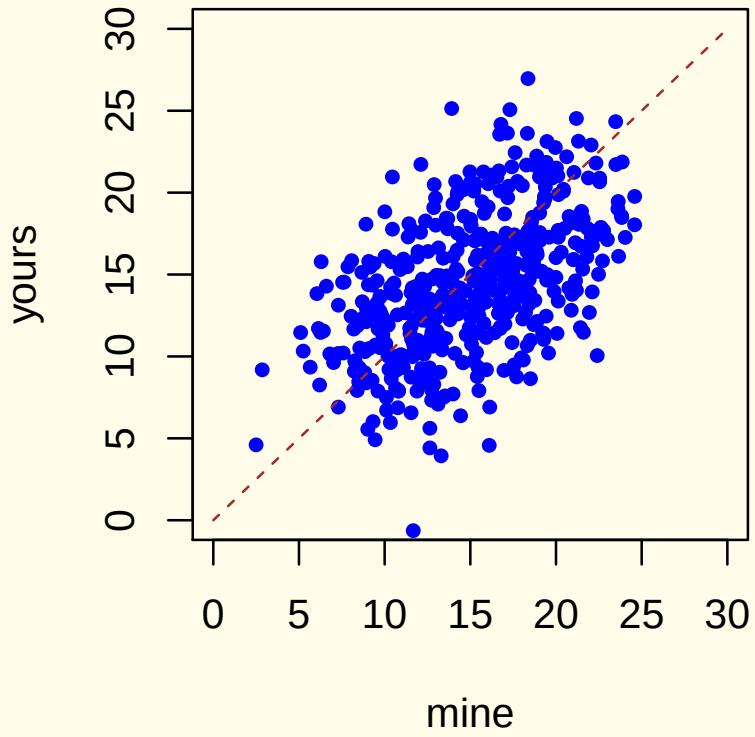
Circle maneuvers: Find wind and TAS from ground track alone

1. This isn't linear: Want to minimize deviations of the actual flight track from that determined by three parameters, TAS (assumed constant) and two components of horizontal wind. See below in connection with "nlm"

REGRESSION

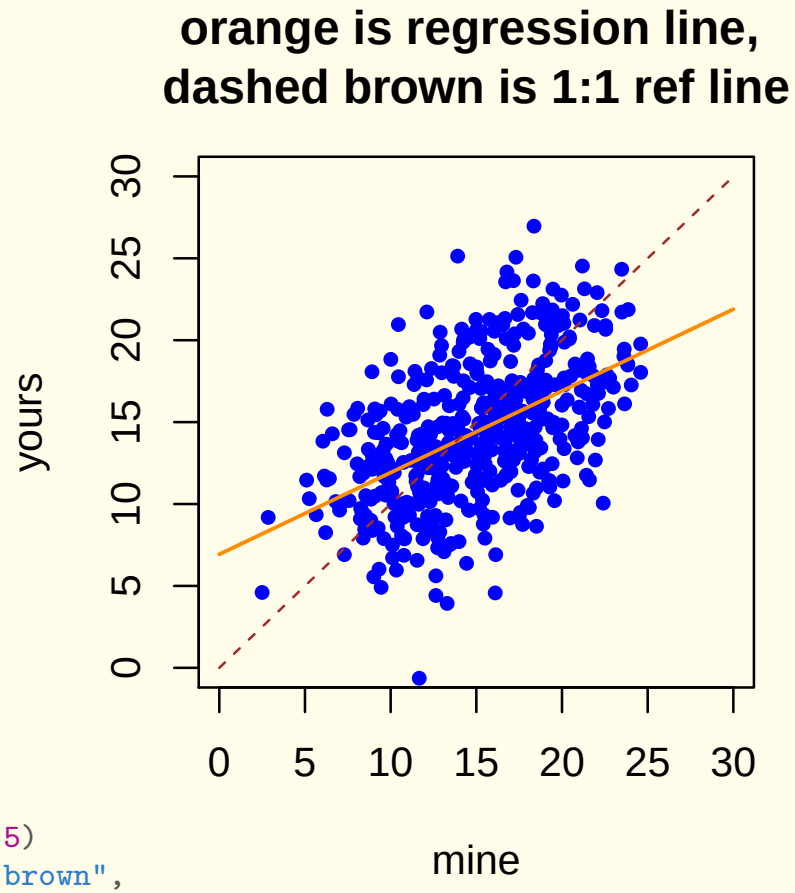
- Will use this as an introduction to the Deming fit
- Generate an appropriate data.frame:

```
## choose a random value x between 0
## and 10 assume that two instruments
## measure x, with the same errors e
x <- runif(500, 10, 20)
mine <- x + rnorm(500, 0, 3)
yours <- x + rnorm(500, 0, 3)
```



REGRESSION: yours = function (mine)

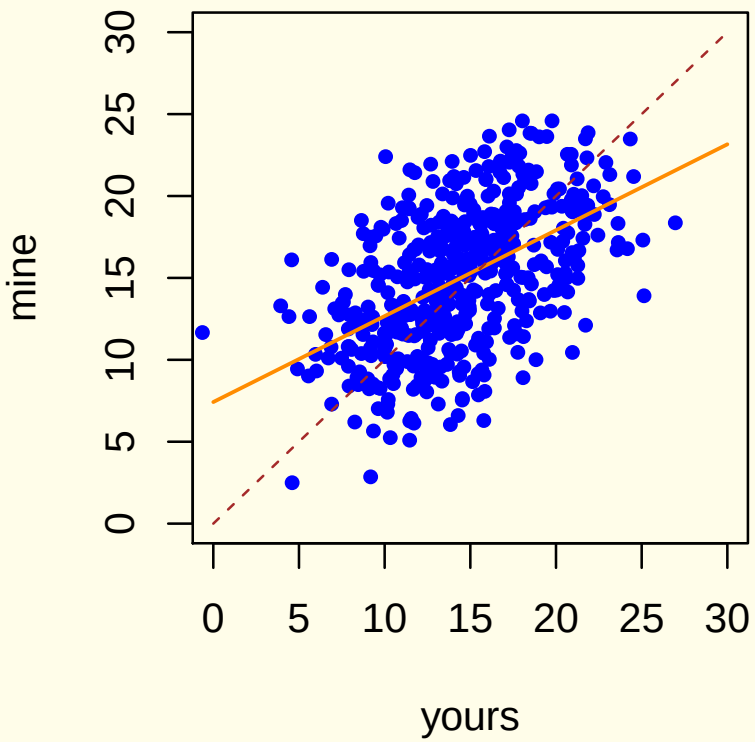
- I fit, yours (mine):
- "your measurements are too high at low values and too low at high values (relative to my correct values)"




```
plot(mine, yours, pch = 20, col = "blue",
    xlim = c(0, 30), ylim = c(0, 30))
Fm1 <- lm(yours ~ mine)
xp <- c(0, 30)
yp <- coef(Fm1)[1] + coef(Fm1)[2] *
    xp
lines(xp, yp, col = "darkorange", lwd = 1.5)
lines(c(0, 30), c(0, 30), lty = 2, col = "brown",
    lwd = 1)
title("orange is regression line, \ndashed brown is 1:1 ref line")
```

REGRESSION: mine = function (yours)

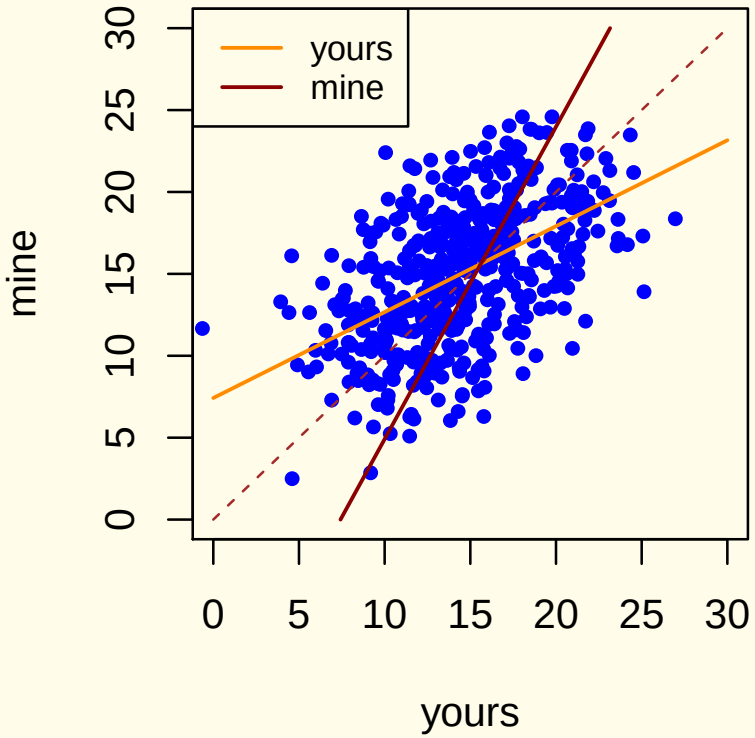
- but you fit using mine (yours):
- you also say: "No, your measurements are too high at low values and too low at high values (relative to my correct values)"
- limit for vanishing correlation gives perpendicular lines



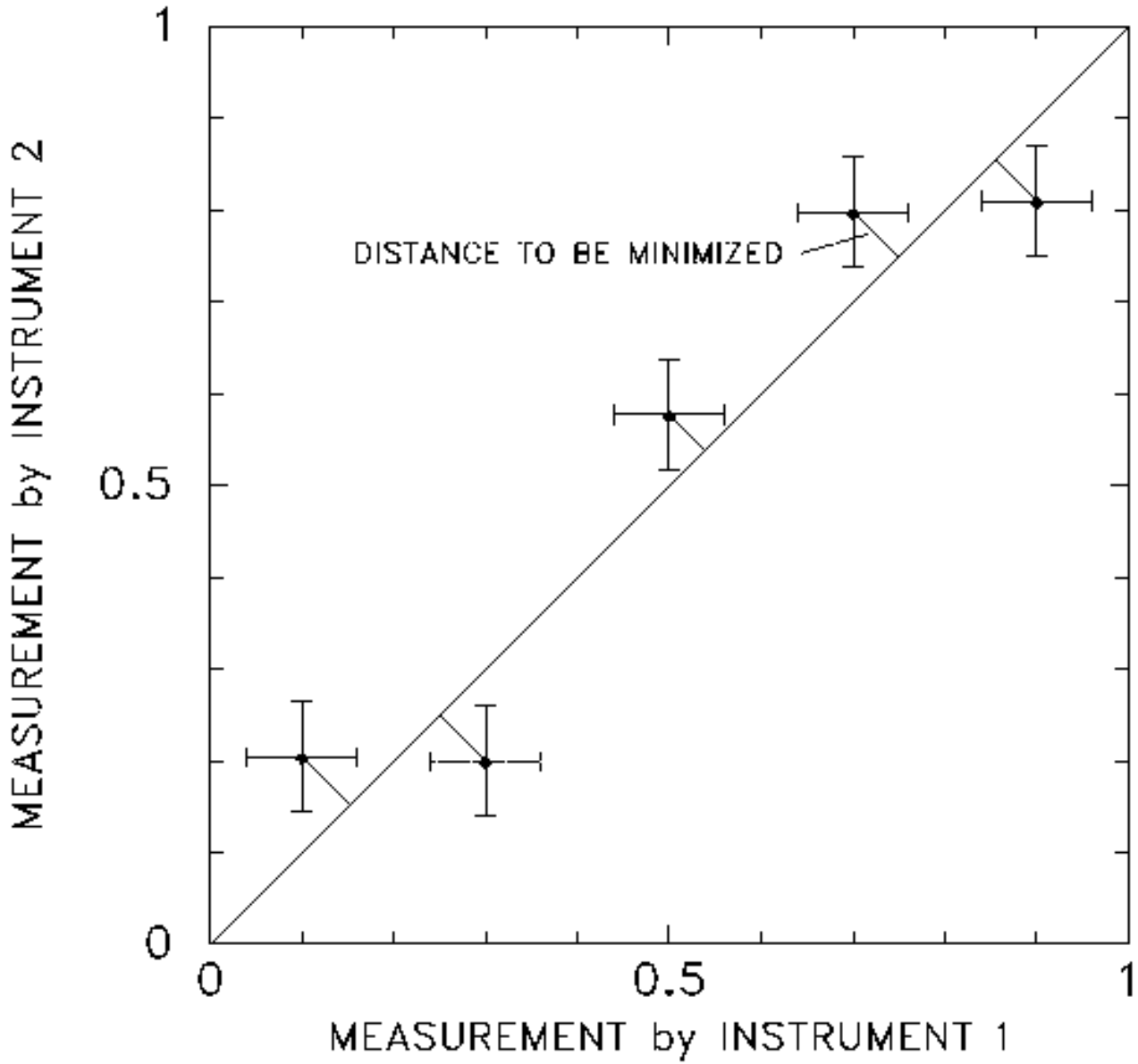

```
Fm2 <- lm(mine ~ yours)
plot(yours, mine, pch = 20, col = "blue",
    xlim = c(0, 30), ylim = c(0, 30))
xp <- c(0, 30)
yp <- coef(Fm2)[1] + coef(Fm2)[2] *
    xp
lines(xp, yp, col = "darkorange", lwd = 1.5)
lines(c(0, 30), c(0, 30), lty = 2, col = "brown",
    lwd = 1)
# title ('orange is regression line,
# \ndashed brown is 1:1 ref line')
```

WHAT CAUSES THIS APPARENT PARADOX?

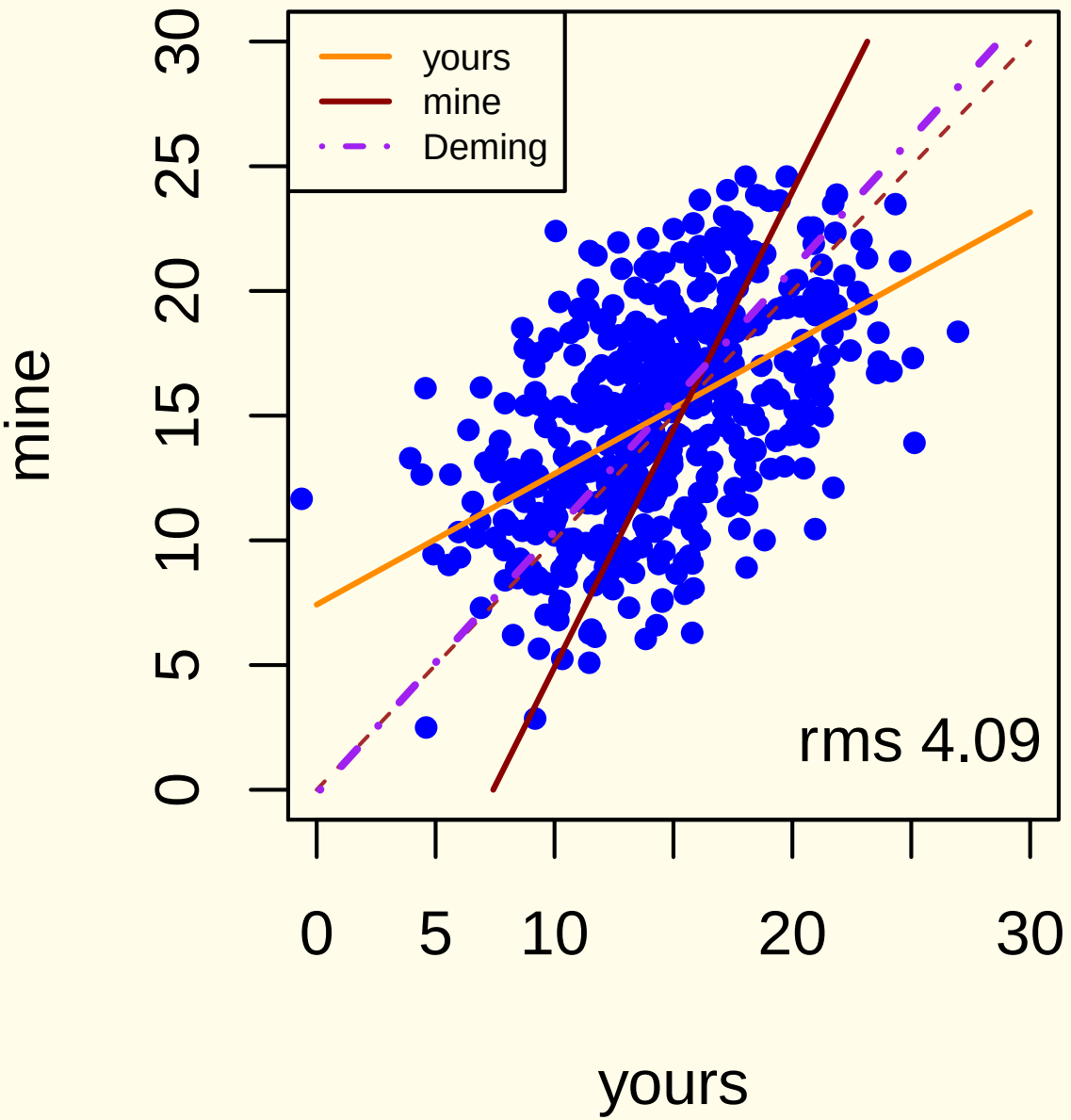
- Each fit is trying to minimize the distance from the points to a line by measuring the distance along an axis.
- To treat each equally, the fit should minimize the perpendicular distance from the line to the points.
- This is called a "Deming fit" or "Deming regression".
- *Never* use ordinary regression to compare measurements!



THE DEMING FIT:



OPTIONS: Ranadu function DemingFit.R, Package mcr, mc.deming() (and others)



NON-LINEAR OPTIMIZATION

Routine *nlm* ()

- Given a function of a vector of variables, finds the values of those variables that minimize the function.
- I used this, for example, to find wind and TAS from the circle maneuvers, varying the three variables TAS, east and north component of the wind, to find the minimum deviation from the GPS ground track.
- This can also be used for finding a maximum, so can be used with the method of maximum likelihood.

AN ADDITIONAL VERY USEFUL FUNCTION: nleqslv

Part of a package also called nleqslv [require(nleqslv)]

Example of use:

```
A <- nleqslv (FirstGuess, fn, jac=NULL,
     method=''Newton'', additionalArguments)
## To find the dew point that corresponds to a
## specified vapor pressure E, first define a
## function that approaches zero as its argument
## DP approaches the dewpoint that corresponds to
## the the equilibrium vapor pressure E:
MKerror <- function (DP, VaporPressure) {
  return (MurphyKoop (DP) - VaporPressure)
}
## use the function with nleqslv to find the answer
DewPoint <- nleqslv (-10., fn=MKerror, jac=NULL,
            method=''Newton'', E)$x
```

FITTING VIA MAXIMUM LIKELIHOOD

Motivation

- Sometimes fitting gets really convoluted with correlated errors, difficult statistics (e.g., Poisson), non-linear functions, or other impediments to standard methods.
- In that case, a brute-force method for minimizing chi-square or maximizing likelihood may save the day.
- Example: A CCN counter produces a set of counted particles $N_i$at stepped supersaturations (SS_i), with known sample volume V_i for each count. It is desirable to fit these measurements to a conventional form $N_i(SS_i) = C\,SS_i^k$, where C and k are constants. However, especially at low supersaturation, there may be small numbers of detected particles, so it is necessary to consider Poisson statistics when evaluating confidence intervals for the counts. But a count of 5 in a given channel does not mean that the Poisson distribution with mean 5 is appropriate, because the observed 5 may come from a true distribution with mean 4, for example.

MATH re MAXIMUM LIKELIHOOD

Basic premise:

- Best estimate of parameters is that giving the highest probability that the actual measurements would be made.
- To find parameters $\{a\}$ (e.g, C and k for the CCN distribution), express the probability of making the actual observations $\{x\}$ (e.g., $\{N_1, N_2, \ldots\}$ given values $\{a\}$ for the parameters: $\phi(x_i; \{a\})$. The probability must be normalized.
- The "likelihood" is the product of probabilities of all the observations:

$$\mathscr{L}(a) = \prod_i \phi(x_i; \{a\}) \text{ and } \mathscr{W} = \log \mathscr{L}(\{a\}) = \sum_i \log(\phi(x_i; \{a\}))$$

- The estimated values of the parameters $\{a\}$ are then the values that lead to the maximum value of $\mathscr{W}$.

APPLICATION TO CCN PROBLEM

1. Assume values for C and k
2. For each observation N_i, determine the probability of making that observation given that the value expected from the assumed parameters is $N_i^* = V_i\, C(SS_i)^k$ where V is the volume sampled. That is given by the Poisson distribution:

$$\phi^{Poisson}(N_i; N_i^*) = \frac{(N_i^*)^{N_i} e^{-N_i^*}}{N_i!}$$

3. Vary C and k over a grid of values to search for the maximum $\mathscr{W}$, or use the R function "nlm ()" to search for the maximum.

But, in R, also consider glm ()

- Like lm() but can represent distributions other than Gaussian, including Poisson. It can perform this CCN fit directly
- Useful to consider when fitting to counted measurements
- Must get the statistics to represent actual counts.

HOW THIS FITS IN THE "SESSIONS"

The Plan:

1. Introduction to R and esp. to RStudio
2. The data.frame and other variables
3. Basic math operations; vector operations
4. Packages, including 'Ranadu'
5. Constructing plots
6. Fit procedures; solving non-linear equations
7. Reproducible Analyses using R and knitr
8. Specific examples of application for RAF tasks

Next: knitr

- embedded documentation
- using LaTeX without knowing LaTeX
- using Markdown to get documents formatted for web pages
- "Reproducible Analysis" is the goal!