

# Analysis of Directional Data

Brett Presnell

2015-09-14

# Examples

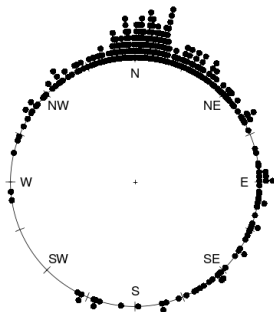
Wish to analyze data in which response is a “direction”:

- ▶ 2d directional data are called *circular* data
- ▶ 3d directional data are called *spherical* data
- ▶ not all “directional” data are directions in the usual sense
- ▶ “directional” data may also arise in higher dimensions

# Wind Directions

- ▶ Recorded at Col de la Roa, Italian Alps
- ▶  $n = 310$  (first 40 listed below)
- ▶ Radians, clockwise from north
- ▶ Source: Agostinelli (CSDA 2007); also R package `circular`

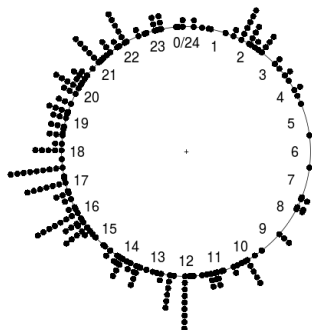
6.23	1.03	0.15	0.72	2.20
0.46	0.63	1.45	0.37	1.95
0.08	0.15	0.33	0.09	0.09
6.23	0.05	6.14	6.28	6.17
6.24	6.02	6.14	6.25	0.01
5.38	5.30	5.63	0.77	1.34
6.14	0.22	6.23	2.33	3.61
0.49	6.12	0.01	0.00	0.46



# Arrival Times at an ICU

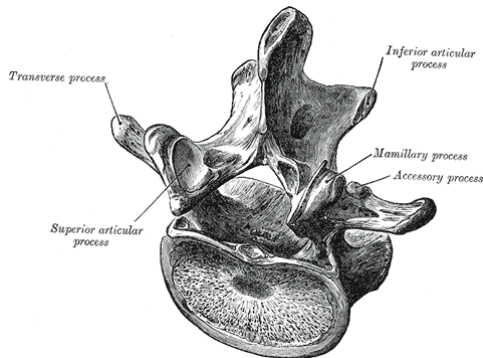
- ▶ 24-hour clock times (format hrs.mins)
- ▶  $n = 254$  (first 32 listed below)
- ▶ Source: Cox & Lewis (1966); also Fisher (1993) and R package `circular`

11.00	17.00	23.15	10.00
12.00	8.45	16.00	10.00
15.30	20.20	4.00	12.00
2.20	12.00	5.30	7.30
12.00	16.00	16.00	1.30
11.05	16.00	19.00	17.45
20.20	21.00	12.00	12.00
18.00	22.00	22.00	22.05

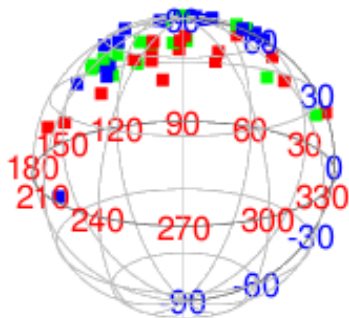


# Primate Vertebrae

- ▶ Orientation of left superior facet of last lumbar vertebra in humans, gorillas, and chimpanzees
- ▶ Source: Keifer (2005 UF Anthropology MA Thesis)



**Figure :** Human lumbar vertebra with right superior facet labelled as superior articulate process.



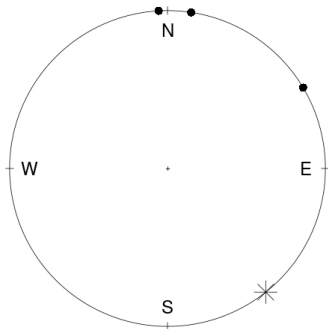
**Figure :** Orientation of left superior facets for samples of 18 chimpanzees (red), 16 gorillas (green) and 19 humans (blue).

# Butterfly Migrations

- ▶ Direction of travel observed for 2649 migrating butterflies in Florida
- ▶ Source: Thomas J Walker, University of Florida, Dept of Entomology and Nematology
- ▶ Other variables:
  - ▶ site: 23 locations in Florida
  - ▶ observer: Thomas Walker (tw) or James J. Whitesell (jw)
  - ▶ species: cloudless sulphur (cs), gulf fritillary (gf), long-tailed skipper (lt)
  - ▶ distance to coast (km)
  - ▶ date and time of observation
  - ▶ percentage of sky free of clouds
  - ▶ quality of sunlight: (b)right, (h)aze, (o)bstructed, (p)artly obstructed
  - ▶ presence/absence and direction (N, NE, E, SE, S, SW, W, NW) of wind
  - ▶ temperature

# Why is the Analysis of Directional Data Different?

- ▶ First three observations from the wind directions data: 6.23, 1.03, 0.15
- ▶ The mean of these three numbers is 2.47
- ▶ What do you think?





# Graphical Display of Circular Data (in R)

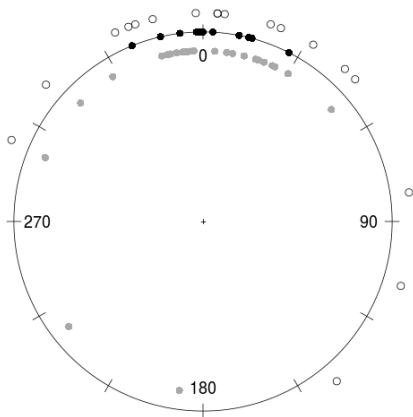
- ▶ Have already seen simple dot plots for circular data, e.g., for the wind data:

```
1  windc <- circular(wind, type="angles", units="radians",
2                    template="geographics")
3  require("circular")
4  par(mar=c(0,0,0,0)+0.1, oma=c(0,0,0,0)+0.1)
5  plot(windc, cex=1.5, axes=FALSE,
6       bin=360, stack=TRUE, sep=0.035, shrink=1.3)
7  axis.circular(at=circular(seq(0, (7/4)*pi, pi/4),
8                           template="geographics"),
9              labels=c("N", "NE", "E", "SE", "S", "SW", "W", "NW"),
10             cex=1.4)
11  ticks.circular(circular(seq(0, (15/8)*pi, pi/8)),
12                zero=pi/2, rotation="clock",
13                tcl=0.075)
```

- ▶ and for the ICU data:

```
1  ## Note that pch=17 does not work properly here.
2  par(mar=c(0,0,0,0)+0.1, oma=c(0,0,0,0)+0.1)
3  plot(fisherB1c, cex=1.5, axes=TRUE,
4       bin=360, stack=TRUE, sep=0.035, shrink=1.3)
```

- ▶ and one more ...



**Figure :** Walking directions of long-legged desert ants under three different experimental conditions:

```
1  par(mar=c(0,0,0,0)+0.1, oma=c(0,0,0,0)+0.1)
2  plot(fisherB10c$set1, units="degrees", zero=pi/2,
3        rotation="clock", pch=16, cex=1.5)
4  ticks.circular(circular(seq(0, (11/6)*pi, pi/6)),
5                  zero=pi/2, rotation="clock", tcl=0.075)
6  points(fisherB10c$set2, zero=pi/2,
7          rotation="clock", pch=16, col="darkgrey",
8          next.points=-0.1, cex=1.5)
9  points(fisherB10c$set3, zero=pi/2,
10         rotation="clock", pch=1,
11         next.points=0.1, cex=1.5)
```

# Circular Histograms

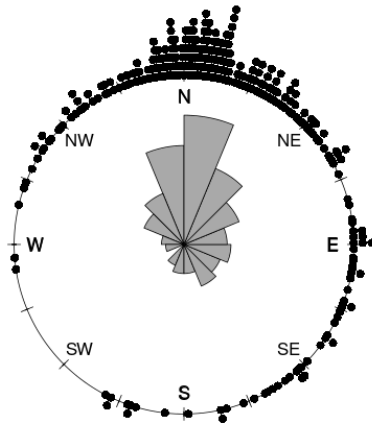
- ▶ Circular histograms exist (see Fisher and Mardia and Jupp)  
but is there a ready-made function in R?

# Rose Diagrams

- ▶ Invented by Florence Nightingale (elected first female member of the Royal Statistical Society in 1859; honorary member of ASA)
- ▶ Nightingale's rose in R (see also this post and the R graph catalog)
- ▶ Note that radii of segments are proportional to *square root* of the frequencies (counts), so that areas are proportional to frequencies. Is this the right thing to do?
- ▶ Rose diagrams suffer from the same problems as histograms. The impression conveyed may depend strongly on:
  - ▶ the binwidth of the cells
  - ▶ the choice of starting point for the bins

# Adding a Rose Diagram to the Plot of Wind Directions

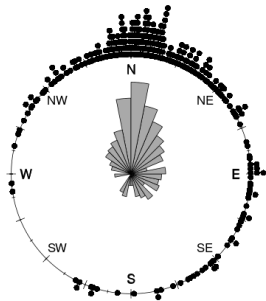
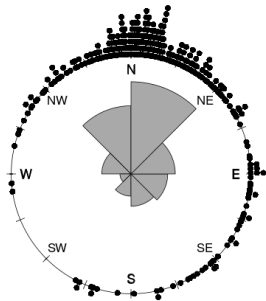
```
1 rose.diag(windc, bins=16, col="darkgrey",  
2           cex=1.5, prop=1.35, add=TRUE)
```



**Figure :** Wind direction data with rose diagram with segment areas are proportional to counts (segment radii are proportional to square roots of counts).



# Changing the Binwidth



# Changing the Radii

- I think that the default “radii proportional to counts” is generally best, but this is not always obvious. The scale certainly makes a big difference however.

```
1 rose.diag(windc, bins=16, col="darkgrey",  
2         radii.scale="linear",  
3         cex=1.5, prop=2.4, add=TRUE)
```

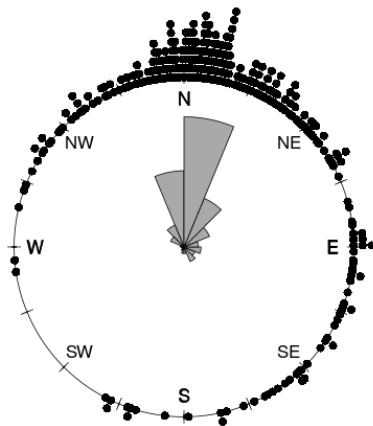


Figure : Wind direction data with rose diagram (segment radii proportional to counts).

# Kernel Density Estimates

```
1 lines(density.circular(windc, bw=40), lwd=2, lty=1)
```

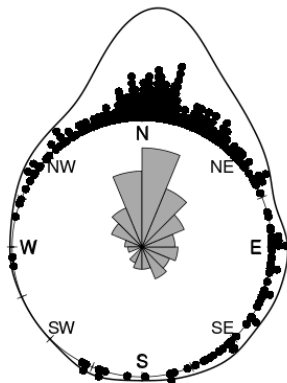


Figure : Wind direction data with rose diagram and kernel density estimate.

# Spherical Data

- ▶ Are there any canned routines for plotting spherical data in R?

# Mean Direction and Mean Resultant Length

- First three observations from the wind directions data:

theta	x	y
6.23	-0.06	1.00
1.03	0.86	0.51
0.15	0.15	0.99

- resultant (sum of direction vectors):  $(0.952, 2.5)$
- mean vector:  $(\bar{x}, \bar{y}) = (0.317, 0.833)$
- resultant length (Euclidean norm of resultant):  $R = 2.675$
- mean resultant length:  $\bar{R} = 0.892$
- mean direction:  $(\bar{x}, \bar{y})/\bar{R} = (0.356, 0.934)$
- $\tilde{\theta} = 0.364$

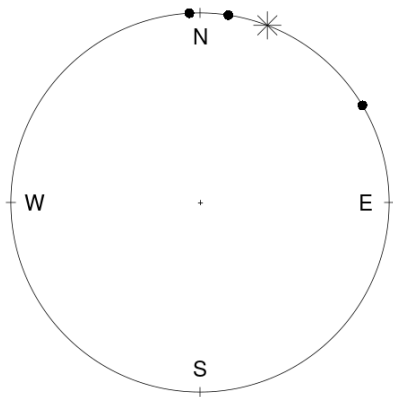


Figure : First three observations from the wind directions data and their sample mean direction.



# Generating Random Points on the Sphere

- ▶ Wish to generate a random “direction” in  $d$ -dimensions; i.e., an observation from the uniform distribution in the  $d - 1$  sphere.
- ▶ Usual way: let  $X \sim N_d(0, I)$  and return  $U = X/||X||$ .
- ▶ An alternative rejection sampler:
  - ▶ Repeat until  $||X|| \leq 1$ 
    - ▶ Let  $X$  be uniformly distributed on the cube  $[-1,1]^d$
  - ▶ Return  $U = X/||X||$
- ▶ What is the acceptance rate for the rejection sampler:
  - ▶ Volume of the  $d - 1$  sphere is  $\pi^{d/2}/\Gamma(d/2 + 1)$
  - ▶ Volume of  $[-1,1]^d$  is  $2^d$
  - ▶ Acceptance rate is  $(\pi^{1/2}/2)^d/\Gamma(d/2 + 1)$
  - ▶ Curse of dimensionality

dimension	2	3	4	5	6	7	8	9	10
accept rate (%)	79	52	31	16	8	4	2	1	0

```

1  runifSphere <- function(n, dimension, method=c("norm", "cube", "slownorm")) {
2      method <- match.arg(method)
3      if (method=="norm") {
4          u <- matrix(rnorm(n*dimension), ncol=dimension)
5          u <- sweep(u, 1, sqrt(apply(u*u, 1, sum)), "/")
6      } else if (method=="slownorm") {
7          u <- matrix(nrow=n, ncol=dimension)
8          for (i in 1:n) {
9              x <- rnorm(dimension)
10             xnorm <- sqrt(sum(x^2))
11             u[i,] <- x/xnorm
12         }
13     } else {
14         u <- matrix(nrow=n, ncol=dimension)
15         for (i in 1:n) {
16             x <- runif(dimension, -1, 1)
17             xnorm <- sqrt(sum(x^2))
18             while (xnorm > 1) {
19                 x <- runif(dimension, -1, 1)
20                 xnorm <- sqrt(sum(x^2))
21             }
22             u[i,] <- x/xnorm
23         }
24     }
25     u
26 }

```

## Easy fix for Borel's paradox in 3-d

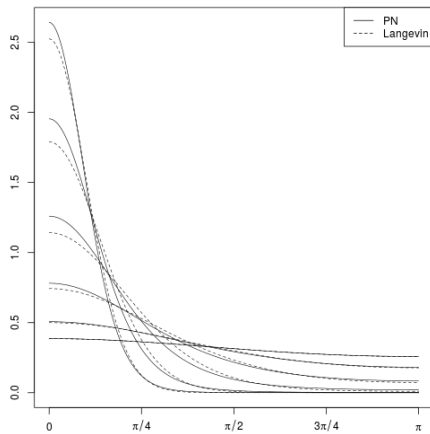
Take longitude  $\phi \sim U(0, 2\pi)$  independent of latitude  $\theta = \arcsin(2U - 1)$ ,  $U \sim U(0, 1)$ .

# Comparison of Projected Normal and Langevin Distributions

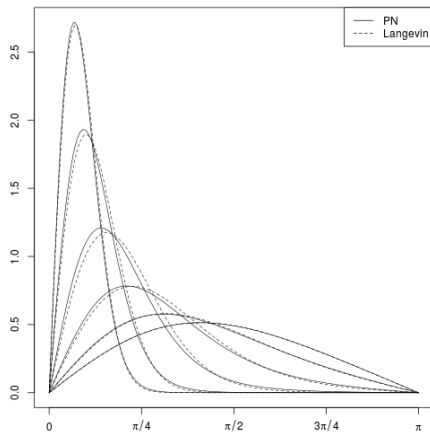
One way that we might compare the  $(\mu, \kappa)$  and  $(\gamma\mu, I)$  distributions by choosing  $\kappa$  and  $\gamma$  to give the same mean resultant lengths and comparing the densities of the cosine of the angle  $\theta$  between  $U$  and  $\mu$ .

Of course matching mean resultant lengths is not necessarily the best way to compare these families of distributions.

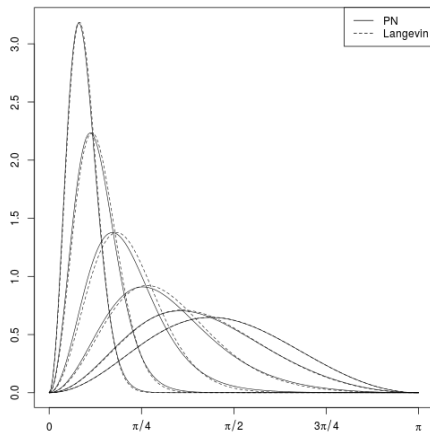
$$d = 2$$



$$d = 3$$



$$d = 4$$



# Gould's Model

A.k.a., the barber pole model.



# Gould's Model: Likelihood

Calculate the (profile) log-likelihood for Gould (1969 Biometrics) model for simple (single predictor) regression with an intercept. For fixed “slope”  $\beta$ , this function “profiles out” (maximizes over) the “intercept” term and optionally the concentration parameter  $\kappa$ .

```
1 loglklhd.gould <- function(beta, theta, x, do.kappa=FALSE) {
2   res <- sapply(beta,
3     function(b, th, x) {
4       sqrt(sum(cos(th - b*x))^2
5         + sum(sin(th - b*x))^2)
6     },
7     th=theta, x=x)
8   if (do.kappa) {
9     n <- length(theta)
10    kappa <- sapply(res/n, imr1LvMF, dimen=2)
11    res <- n*log(constLvMF(kappa, dimen=2)) + kappa*res
12  }
13  res
14 }
```

# Gould's Model with Equally Spaced X

```
1  alpha <- 0
2  beta <- 1
3  kappa = 2.5
4  x <- seq(-1, 1, length=10)
5  mu <- as.circular((alpha + beta*x) %%(2*pi))
6  theta <- as.circular(mu + rvonmises(length(mu), mu=0, kappa=kappa))
7  period <- 2*pi/(min(diff(sort(x)))) # Useful only for lattice x
8  nperiods <- 1
9  curve(loglklhd.gould(beta, theta, x, do.kappa), xname="beta",
10        xlim=beta + nperiods*period*c(-1.125,1.125), n=nperiods*200,
11        xlab=expression(beta),
12        ylab="Log-Likelihood")
13  abline(v = beta + ((-nperiods):nperiods)*period, lty=3) # for lattice x
```

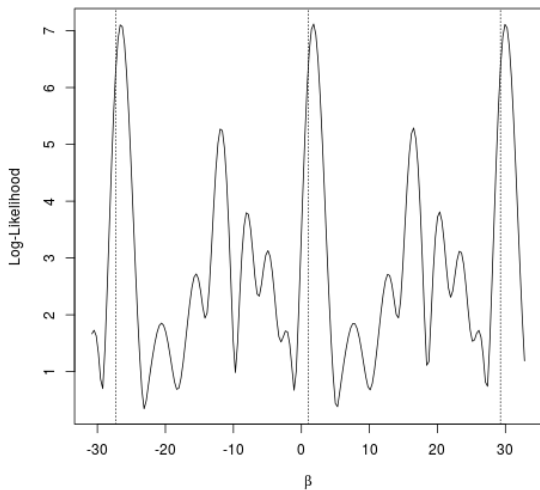


Figure : Gould's model log-likelihood with  $n=10$  equally-spaced  $x$ 's;  $\kappa$  not profiled out.

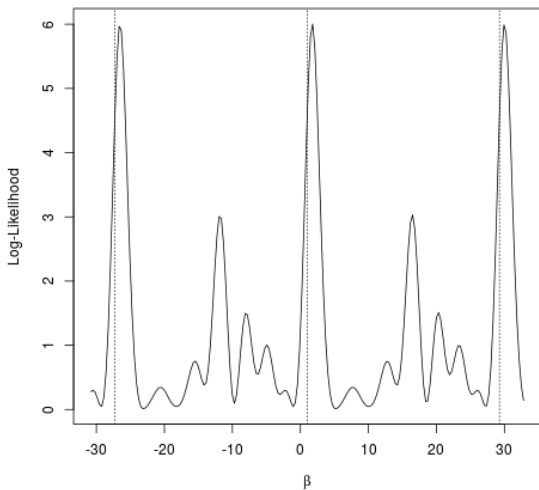


Figure : Gould's model log-likelihood with  $n=10$  equally-spaced  $x$ 's;  $\kappa$  profiled out.

# Gould's Model with Random X: Data Generation

```
1  alpha <- 0
2  beta <- 1
3  kappa = 2.5
4  x <- rnorm(10)
5  mu <- as.circular((alpha + beta*x) %% (2*pi))
6  theta <- as.circular(mu + rvonmises(length(mu), mu=0, kappa=kappa))
```

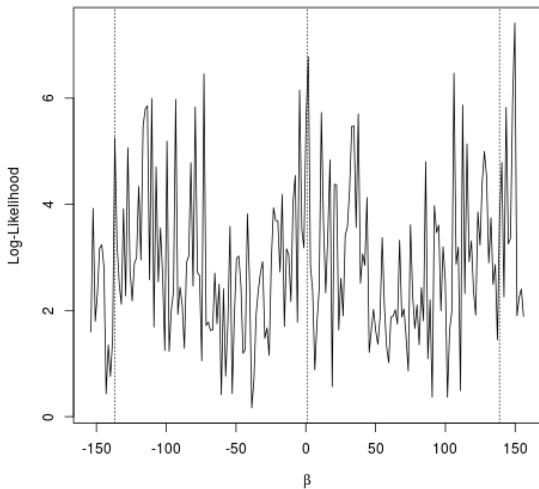


Figure : Gould's model log-likelihood with  $n=10$  random normal  $x$ 's;  $\kappa$  not profiled out.

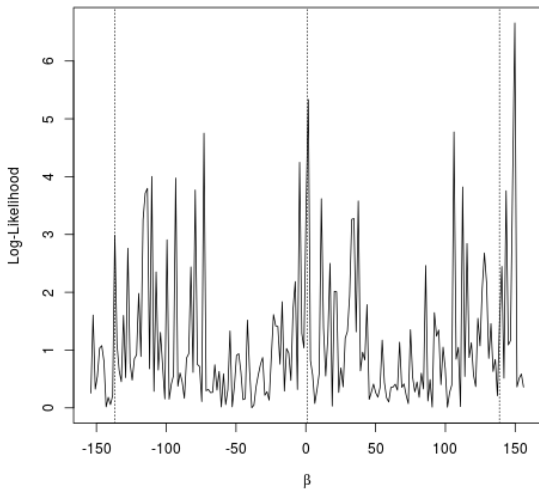


Figure : Gould's model log-likelihood with  $n=10$  random normal  $x$ 's;  $\kappa$  profiled out.

## Fisher-Lee Model: Likelihood

Calculate the (profile) log-likelihood for the Fisher-Lee (1992 Biometrics) model. For fixed “slope”  $\beta$ , this function “profiles out” (maximizes over) the “intercept” term and optionally the concentration parameter  $\kappa$ . Computing this with biggish matrix multiplies instead of using `apply()` or looping.

```
1 logklhdFisherLee <- function(beta, theta, X, do.kappa=FALSE) {
2   n <- length(theta)
3   nbeta <- dim(beta)[2]
4   if (dim(X)[1] != n) {
5     stop("Number of rows of X must equal length of theta.")
6   }
7   if (dim(beta)[1] != dim(X)[2]) {
8     stop("Number of rows of beta must equal number of columns of X")
9   }
10  dev <- theta - 2*atan(X %*% beta)
11  res <- sqrt(apply(cos(dev), 2, sum)^2
12             + apply(sin(dev), 2, sum)^2)
13  if (do.kappa) {
14    kappa <- sapply(res/n, imrLvMF, dimen=2)
15    res <- n*log(constLvMF(kappa, dimen=2)) + kappa*res
16  }
17  res
18 }
```



# Fisher-Lee Model with Random X: Data Generation

Note that Fisher recommends centering the  $x$  values before fitting the model. Here, to be certain that the model whose likelihood we plot is equivalent to the data generating model, we will center the  $x$  values before generating the responses.

```
1  alpha <- 0
2  beta <- 1
3  kappa = 2.5
4  x <- rnorm(10)
5  x <- x - mean(x)
6  mu <- as.circular(alpha + 2*atan(beta*x))
7  theta <- as.circular(mu + rvonmises(length(mu), mu=0, kappa=kappa))
```

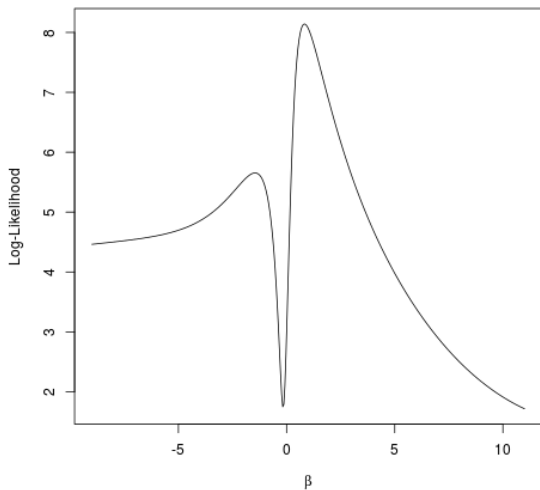


Figure : Fisher-Lee model log-likelihood with  $n=10$  random normal  $x$ 's;  $\kappa$  not profiled out.

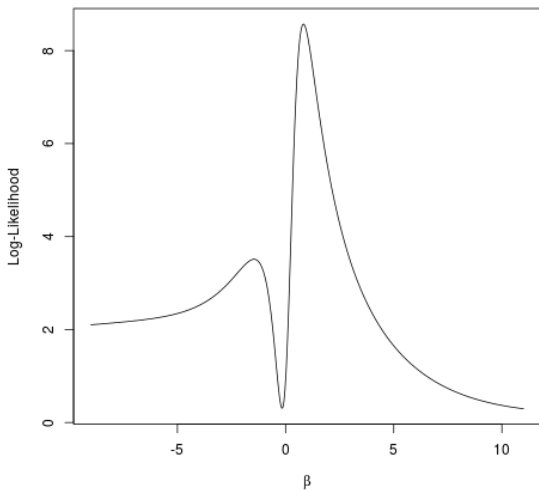


Figure : Fisher-Lee model log-likelihood with  $n=10$  random normal  $x$ 's;  $\kappa$  profiled out.

# Blue Periwinkles

```
1  periwinkles <- read.table(datafile("periwinkle.txt"), header=TRUE)
```

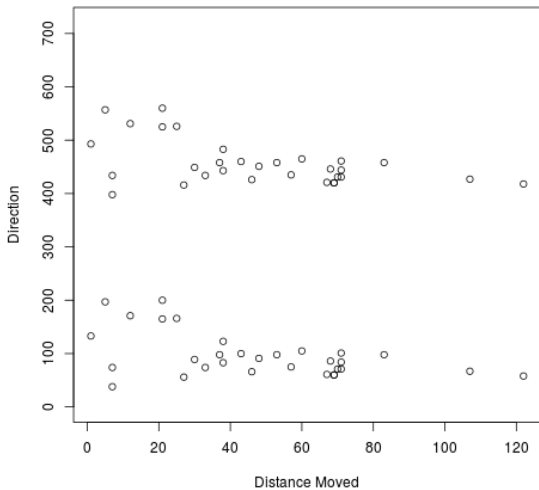


Figure : Direction and distance moved by 31 small blue periwinkles.

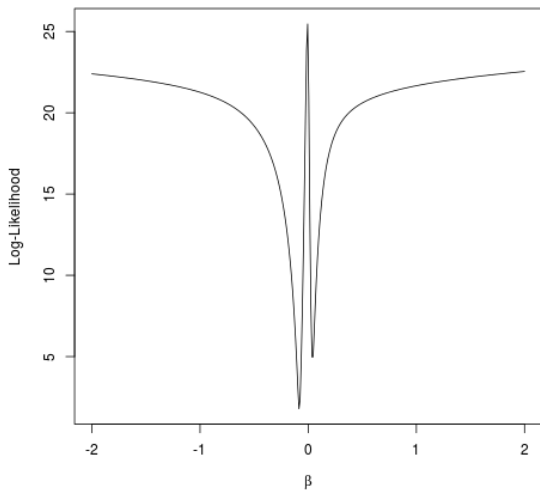


Figure : Fisher-Lee model log-likelihood for periwinkle data;  $\kappa$  not profiled out.

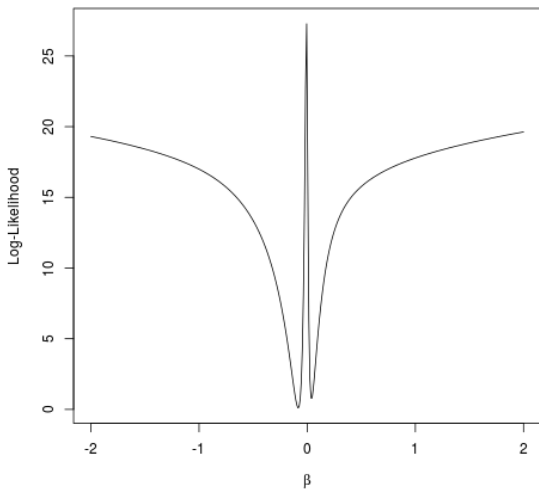
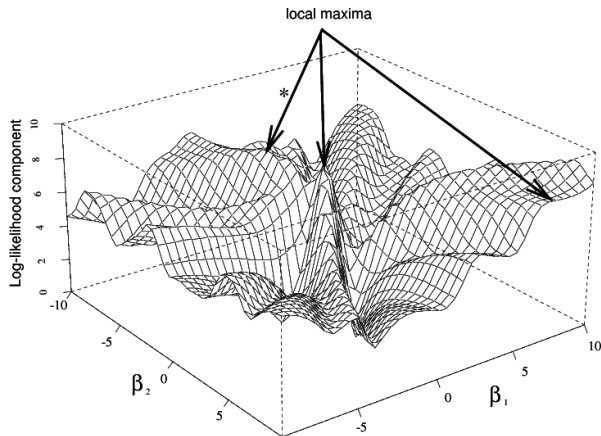


Figure : Fisher-Lee model log-likelihood for periwinkle data;  $\kappa$  profiled out.

# Fisher-Lee Model with Two Predictors



**Figure :** Fisher-Lee profile log-likelihood for a simulated data set with  $n = 10$ ,  $\kappa = 1.0$ ,  $\beta_1 = 0.1$ , and  $\beta_2 = 0.1$ . The global maximum is indicated by an asterisk.