Stat 344 Final

Trey Tipton

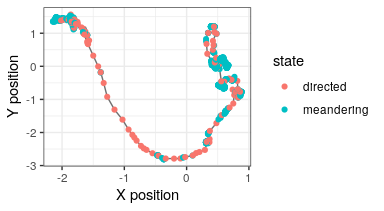
April 27, 2022

### 0. Academic Honesty Statement

I, Trey Tipton, affirm on my honor that this submission contains my own work and I did not get help from, or give help to, another person.

### 1. Circular Oxen

ox <- read.csv('https://sldr.netlify.app/data/ox.csv') |>  
 mutate(state = factor(state)) |>  
 na.omit()  
  
gf\_path(ys ~ xs, data = ox, color = 'grey44') |>  
 gf\_point(color = ~state) |>  
 gf\_labs(x = 'X position', y = 'Y position')



## A. von Mises MLE

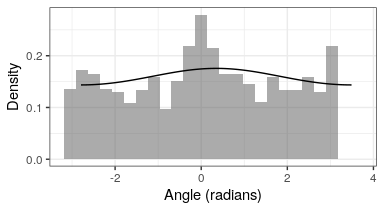
ll\_vm <- function(theta, x){  
 mu <- theta[1]  
 kappa <- theta[2]  
 if (kappa < 0) return(NA)  
 dvonmises(x, mu = mu, kappa = kappa, log = TRUE)  
}  
  
maxLik(ll\_vm, start = c(mu = 1, kappa = .1), x = ox$angle)

## Maximum Likelihood estimation  
## Newton-Raphson maximisation, 4 iterations  
## Return code 8: successive function values within relative tolerance limit (reltol)  
## Log-Likelihood: -2589.687 (2 free parameter(s))  
## Estimate(s): 0.349443 0.1005169

Using maximum-likelihood estimation to fit a von Mises distribution to the angle data, the fitted parameter estimates are as follows:

.

gf\_dhistogram(~angle, data = ox) %>%  
 gf\_labs(x = 'Angle (radians)', y = 'Density') %>%  
 gf\_dist('vonmises', mu = 0.349443, kappa = 0.1005169)



## B. von Mises - uniform

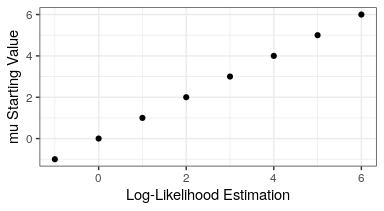
maxLik(ll\_vm, start = c(mu = -1, kappa = 0), fixed = 2, x = ox$angle)

## Maximum Likelihood estimation  
## Newton-Raphson maximisation, 1 iterations  
## Return code 1: gradient close to zero (gradtol)  
## Log-Likelihood: -2593.245 (1 free parameter(s))  
## Estimate(s): -1 0

ll\_unif <- function(theta, x){  
 mu <- theta[1]  
 dvonmises(x, mu = mu, kappa = 0, log = TRUE)  
}  
  
maxLik(ll\_unif, start = c(mu = 10), x = ox$angle)

## Maximum Likelihood estimation  
## Newton-Raphson maximisation, 1 iterations  
## Return code 1: gradient close to zero (gradtol)  
## Log-Likelihood: -2593.245 (1 free parameter(s))  
## Estimate(s): 10

mu\_starting\_val <- c(-1, 0, 1, 2, 3, 4, 5, 6)  
ll\_estimation <- c(-1, 0, 1, 2, 3, 4, 5, 6)  
  
gf\_point(mu\_starting\_val ~ ll\_estimation)%>%  
 gf\_labs(x = "Log-Likelihood Estimation", y = "mu Starting Value")



It appears that when is zero, the log-likelihood estimation for is just the starting value for . This is likely because when is zero, the distribution is uniform (perfectly circular), meaning that the mean () is fixed based on what starting value you use, because the maxLik function cannot optimize for when the data is circular.

## C. von Mises test

Hypotheses:

: = , is equal to zero and the distribution of the oxen movement angles is therefore uniform.

: = , is greater than zero and the oxen movement angles fit better to some sort of von Mises distribution.

where = <, >,

= { | = 0}, and

= { | > 0}}

Test Stat:

Using the maximum likelihood estimations from the previous question, we got two values, one from the von Mises MLE and one from the uniform MLE (von Mises but with fixed at zero):

von Mises Log-Likelihood: -2589.687

Uniform Log-Likelihood: -2593.245

We use those for our W test statistic:

w <- 2\*(-2589.687 - -2593.245 )  
1 - pchisq(w, 1)

## [1] 0.007639898

With a low p-value of 0.00764, we reject the null hypothesis that a uniform distribution generates the data and that is zero. Therefore, it is likely that is greater than zero and a von Mises distribution fits the data better, since we do not have enough evidence to say that .

## D. Goodness!

Hypotheses:

{}, where = <, >, A von Mises distribution generates the angle data for the ox data set.

A von Mises distribution does not generate the angle data for the ox data set.

Test Statistic:

=

min(ox$angle)

## [1] -3.140357

max(ox$angle)

## [1] 3.141593

angle\_breaks <- c(-pi+.000001, -(3\*pi)/4, -pi/2, -pi/4, 0, pi/4, pi/2, (3\*pi)/4, pi)  
ox <- ox %>%  
 mutate(binned\_angles = cut(angle, breaks = angle\_breaks))  
  
tally(~binned\_angles, data = ox)

## binned\_angles  
## (-3.14,-2.36] (-2.36,-1.57] (-1.57,-0.785] (-0.785,0] (0,0.785]   
## 169 158 138 222 223   
## (0.785,1.57] (1.57,2.36] (2.36,3.14]   
## 145 175 181

count\_dat <- data.frame(tally(~binned\_angles, data = ox))  
count\_dat

## binned\_angles Freq  
## 1 (-3.14,-2.36] 169  
## 2 (-2.36,-1.57] 158  
## 3 (-1.57,-0.785] 138  
## 4 (-0.785,0] 222  
## 5 (0,0.785] 223  
## 6 (0.785,1.57] 145  
## 7 (1.57,2.36] 175  
## 8 (2.36,3.14] 181

count\_dat <- count\_dat %>%  
 mutate(probs = diff(pvonmises(angle\_breaks, mu = 0.349443, kappa = 0.1005169,   
 from = pi+.000000001)), e = sum(count\_dat$Freq) \* probs)  
count\_dat$e

## [1] 159.5294 164.6831 176.7215 189.1231 194.0184 187.9976 175.2307 163.6960

G\_angle <- 2 \* sum( count\_dat$Freq \* log( count\_dat$Freq / count\_dat$e))  
G\_angle

## [1] 31.99943

1 - pchisq(G\_angle, df = nrow(count\_dat) - 1 - 2)

## [1] 5.942794e-06

pearson\_angle <- sum(((count\_dat$Freq - count\_dat$e)^2) / count\_dat$e)  
  
1 - pchisq(pearson\_angle, df = 2)

## [1] 1.831628e-07

Both of the p-values we computed from the Likelihood Ratio Test Statistic and the Pearson chi-square test stat are very low at 5.942794e-06 and 1.831628e-07, therefore we reject the null hypothesis that a von Mises distribution is a good fit to the data. This means that although it is a better fit than a uniform distribution, we do not have enough evidence to say that the angle data are formed by a von Mises distribution.

### 2. More Oxen

## A. Regression - Fit

ox\_lm <- glm(state ~ altitude + time, data = ox, family = binomial(link = 'logit'))  
msummary(ox\_lm)

## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.36518 0.17224 7.926 2.27e-15 \*\*\*  
## altitude 0.00825 0.00117 7.048 1.81e-12 \*\*\*  
## timenight 0.82901 0.26471 3.132 0.00174 \*\*   
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 664.15 on 1410 degrees of freedom  
## Residual deviance: 597.05 on 1408 degrees of freedom  
## AIC: 603.05  
##   
## Number of Fisher Scoring iterations: 6

Model Equation:

$$logit(p\_{meandering}) = log \bigg{(} \frac{p\_{meandering}}{(1-p\_{meandering})} \bigg{)} = 1.36518 + 0.00825x\_{altitude} + 0.82901x\_{time}$$

$$y\_{state} \sim {\sf Binom}(1, p\_{meandering})$$

where is the probability that an ox is meandering, and = 2 if it is nighttime and 1 if daytime.

## B. Ox at night

p <- 1/(1 + exp(-(1.36518 + 0.82901\*2)))  
p

## [1] 0.9536113

The probability that an ox would be meandering when at sea level ( = 0) and at midnight ( = 2) is 0.9536113.

## C. Ox at night again

odds\_night <- 0.9536113/(1 - 0.9536113)  
odds\_night

## [1] 20.55697

p\_day <- 1/(1 + exp(-(1.36518 + 0.82901)))  
odds\_day <- p\_day/(1-p\_day)  
p\_day

## [1] 0.8997266

odds\_day

## [1] 8.97273

st.err <- sqrt(diag(vcov(ox\_lm)))  
st.err

## (Intercept) altitude timenight   
## 0.172243372 0.001170412 0.264705334

odds <- -(odds\_night - odds\_day)  
  
odds

## [1] -11.58424

odds + c(-1, 1)\*st.err[3]\*qnorm(0.975)

## [1] -12.10306 -11.06543

The odds of meandering for the sea-level ox decreases by about 11.58424 as time advances from midnight to daytime. From this we can create a confidence interval: we are 95% confidence that the change in odds of meandering at sea-level from midnight to daytime is contained in the interval (-12.10306, -11.06543).

### 3.Yet more oxen

## A. Fit a model

ox <- ox %>%  
 mutate(time2 = ifelse(time=="day",1,0))  
  
  
ll\_mod <- function(theta, x){  
 beta0 <- theta[1]  
 beta1 <- theta[2]  
 beta2 <- theta[3]  
 beta3 <- theta[4]  
 beta4 <- theta[5]  
 sigma <- theta[6]  
 resids <- ox$step - (beta0 + beta1\*ox$time2 + beta2\*ox$temp + beta3\*ox$altitude + beta4\*ox$xs)  
 if (sigma < 0) return(NA)  
 dnorm(resids, mean = 0, sd = sigma, log = TRUE)  
}  
  
  
maxLik(logLik = ll\_mod, start = c(beta0 = 1, beta1 = 1, beta2 = -1, beta3 = 0, beta4 = -1, sigma = 1), x = ox)

## Maximum Likelihood estimation  
## Newton-Raphson maximisation, 46 iterations  
## Return code 8: successive function values within relative tolerance limit (reltol)  
## Log-Likelihood: -78591044 (6 free parameter(s))  
## Estimate(s): 144.2272 64.61116 -6.31048 -0.7636456 -41.22386 0.9229477

where is ox’s step length and is zero if it is night, and 1 if it is day.

## B. Select

ox\_model <- lm(step ~ time2 + temp + altitude + xs, data = ox)  
msummary(ox\_model)

## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 144.1810 24.3082 5.931 3.78e-09 \*\*\*  
## time2 54.4010 16.8966 3.220 0.001313 \*\*   
## temp -6.8724 1.2666 -5.426 6.78e-08 \*\*\*  
## altitude -0.3951 0.1231 -3.208 0.001366 \*\*   
## xs -44.6570 13.1955 -3.384 0.000733 \*\*\*  
##   
## Residual standard error: 300.6 on 1406 degrees of freedom  
## Multiple R-squared: 0.06172, Adjusted R-squared: 0.05905   
## F-statistic: 23.12 on 4 and 1406 DF, p-value: < 2.2e-16

Backwards Step-wise Selection: we will do an Anova test on the full model to see which predictor produces the largest large p-value, and continue doing so until all p-values are small.

car::Anova(ox\_model)

## Anova Table (Type II tests)  
##   
## Response: step  
## Sum Sq Df F value Pr(>F)   
## time2 936788 1 10.366 0.0013129 \*\*   
## temp 2660658 1 29.442 6.778e-08 \*\*\*  
## altitude 930095 1 10.292 0.0013662 \*\*   
## xs 1035040 1 11.453 0.0007333 \*\*\*  
## Residuals 127061446 1406   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

From the full model, there is no large p-values, so each predictor plays a significant role in the model. The best model includes time, temp, altitude, and xs as predictors for step length.

## C. Predict

new\_data <- data.frame(altitude = 0, time2 = 0, xs = .5, temp = -10)  
conf\_int <- predict(ox\_model, newdata = new\_data,   
 interval = 'prediction',  
 level = 0.95)  
conf\_int

## fit lwr upr  
## 1 190.5765 -402.2231 783.376

A 95% prediction interval interval for the distance (meters) moved in the next hour for an ox at sea level, at midnight, when it was -10 degrees, and at an xs position of 0.5 is (-402.2231 , 783.376). Since we know that the ox can only move a positive distance, we can predict with 95% confidence that the ox will move between 0 and 783.376 meters in the next hour.