Powerlifting Strength Comparison

Flash

12/05/2022

Data Prep for males >=20y

```
data <- read.csv("openpowerlifting.csv", header=TRUE)
```

```
d2 <- data[(data$Sex=="M")&(data$Equipment=="Raw")
           &(data$Event=="SBD")&(data$Age>=20),]
rm("data")
```

```
d2 <- d2[!is.na(d2$Best3BenchKg),]
d2 <- d2[!is.na(d2$Best3SquatKg),]
d2 <- d2[!is.na(d2$Best3DeadliftKg),]
d2 <- d2[!is.na(d2$TotalKg),]
d2 <- d2[!is.na(d2$BodyweightKg),]
d2 <- d2[(d2$Best3BenchKg>0)&(d2$Best3DeadliftKg>0)
         &(d2$Best3SquatKg>0)&(d2$TotalKg>0),]
```

```
d2$logBW <- log(d2$BodyweightKg)
d2$logBench <- log(d2$Best3BenchKg)
d2$logSquat <- log(d2$Best3SquatKg)
d2$logTotal <- log(d2$TotalKg)
d2$logDead <- log(d2$Best3DeadliftKg)
```

```
nrow(d2)
```

```
## [1] 111028
```

```
d2 <- d2[!duplicated(d2[,c('Name')]),]
nrow(d2)
```

```
## [1] 47003
```

```
d2$Rel_Total <- d2$TotalKg / d2$BodyweightKg
```

```
indices = sort(sample(nrow(d2), nrow(d2)*.8))
d2_train <- d2[indices,]
d2_test <- d2[-indices,]
```

```
library(rethinking)
```

```
library(MASS)
```

```
library(entropy)
```

Coloured Density Curves

If the density curves moves as the colour changes, that means there are different distributions of the score as the bodyweight changes. In other words, it isn't a fair score.

```
ColouredDensities <- function(score,bw){
  Weight_classes <- seq(55,135,by=5)
  color_vector <- hsv(seq(5/6,0,length.out=length(Weight_classes-1)),1,1)
  curve(dnorm(x,mean=mean(score),sd=sd(score)),xlim=c(-5,5),ylim=c(0,.5))

  for(i in 2:length(Weight_classes)){
    lines(density(score[exp(bw)>Weight_classes[i-1]
                    & exp(bw)<Weight_classes[i]]),col = color_vector[i-1])
  }

}
```

Wandering Schematic Plot

whuber, Measures of residuals heteroscedasticity, URL (version: 2021-06-09): https://stats.stackexchange.com/q/33033

```
WanderingSchematic <- function(n_bins,score,bw,title,yax_lab){
  n.bins <- n_bins
  bins <- cut(bw, quantile(bw,
                  probs = seq(0, 1, 1/n.bins)))
  b <- boxplot(score ~ bins, boxwex=1/2,
               main=title,
               xlab="Bodyweight", ylab=yax_lab,ylim=c(-5,5))
  colors <- hsv(seq(2/6, 1, 1/6))
  temp <- sapply(1:5, function(i) lines(lowess(1:n.bins,
                   b$stats[i,], f=.25),
        col=colors[i], lwd=2))
  abline(0.674,0)
  abline(-0.674,0)
}
```

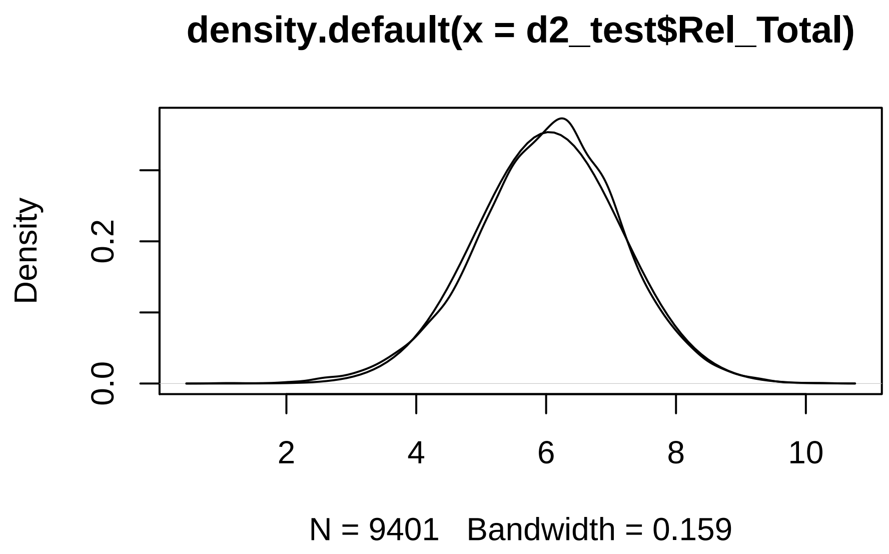
Relative Bodyweight

```
Relative_model <- quap(
  alist(
    Rel_Total ~ dnorm( mu , sigma ) ,
    mu <- dnorm( 1.5 , 1 ) ,
    sigma ~ dexp( 0.1 )

  ) , data=d2_train )
precis(Relative_model)
```

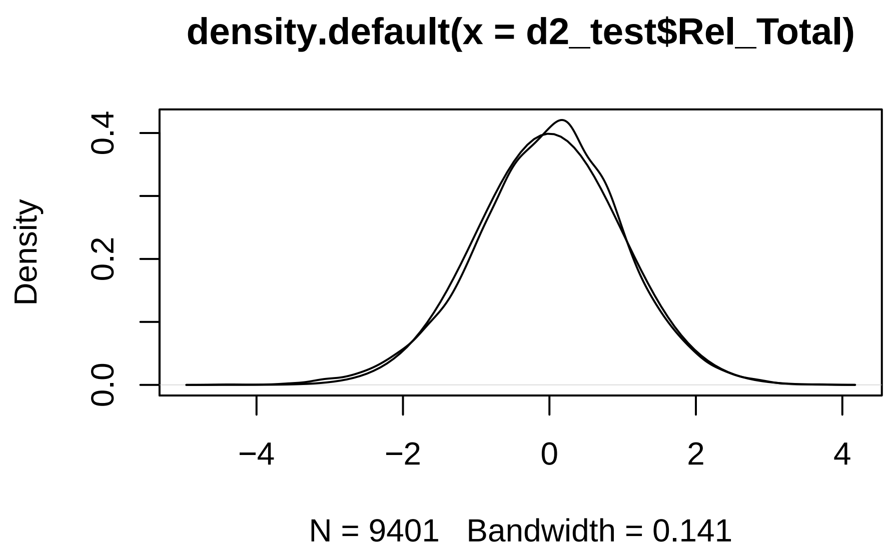
```
##           mean          sd     5.5%    94.5%
## mu    6.050769 0.005815505 6.041474 6.060063
## sigma 1.127717 0.004112048 1.121146 1.134289
```

```
post <- extract.samples( Relative_model )
map_mu <- mean(post$mu)
map_sigma <- mean(post$sigma)
plot(density(d2_test$Rel_Total))
curve(dnorm(x,map_mu,map_sigma),add=TRUE)
```



Normalize to Z Score

```
d2_test$Rel_Total <- ( d2_test$Rel_Total - map_mu) / map_sigma
plot(density(d2_test$Rel_Total))
curve(dnorm(x,0,1),add=TRUE)
```



Log Log / Allometric model

$$\ln(y) = \alpha + \beta \ln(x)$$

$$y = e^{\alpha + \beta \ln(x)}$$

$$y = e^{\alpha} e^{\beta \ln(x)}$$
$$y = e^{\alpha} x^{\beta}$$

Hence, the log transform of both variables is known as power law or allometric scaling

```
xbar <- mean(d2_train$logTotal)
loglogTotal_model <- quap(
  alist(
    logTotal ~ dnorm( mu , sigma ) ,
    mu <- a + b*( logBW - xbar ) ,
    a <- dnorm( 7 , 0.5) ,
    b <- dexp( 2 ) ,
    sigma ~ dexp( 3 )

  ) , data=d2_train )
precis(loglogTotal_model)
```

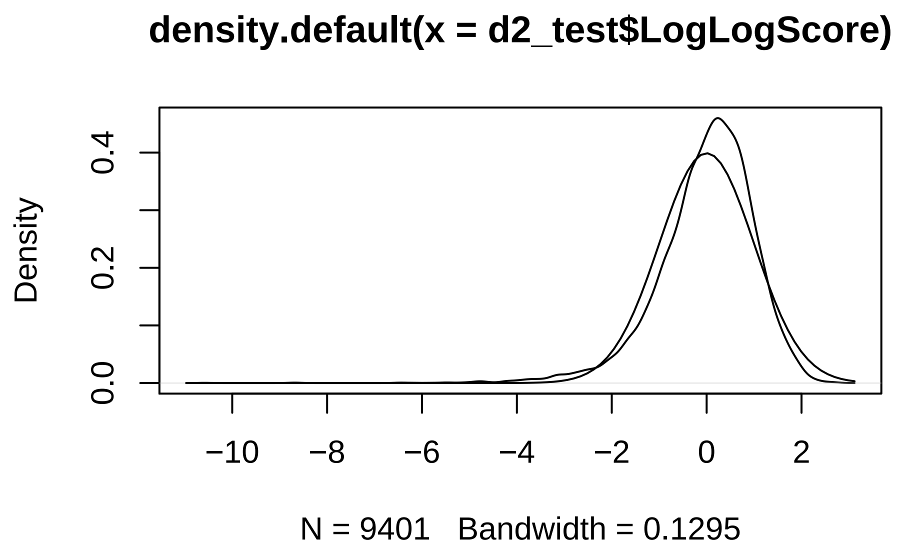
```
##              mean           sd      5.5%     94.5%
## a     7.2680484 0.0082972779 7.2547878 7.2813091
## b     0.5538484 0.0046288604 0.5464506 0.5612463
## sigma 0.1795228 0.0006545613 0.1784766 0.1805689
```

```
xbar <- mean(d2_train$logTotal)
post <- extract.samples( loglogTotal_model )
map_a <- mean(post$a)
map_b <- mean(post$b)
map_sigma <- mean(post$sigma)
map_means <- map_a + (map_b * (d2_test$logBW - xbar) )
```

Normalize to Z Score

```
d2_test$LogLogScore <- ( d2_test$logTotal - map_means ) / map_sigma
```

```
plot(density(d2_test$LogLogScore))
curve(dnorm(x,0,1),add=TRUE)
```



Varying Variance Model

```
xbar <- mean(d2_train$BodyweightKg)

varying_variance <- quap(
  alist(
    TotalKg ~ dnorm( mu , sigma ) ,
    mu <- a + b*( BodyweightKg - xbar ) ,
    a ~ dnorm( 500 , 40 ) ,
    b ~ dnorm( 3, 2 ) ,
    sigma <- d + c*(BodyweightKg - xbar) ,
    d ~ dnorm( 80, 5 ) ,
    c ~ dexp( 2 )
  ) , data=d2_train )
precis(varying_variance)
```

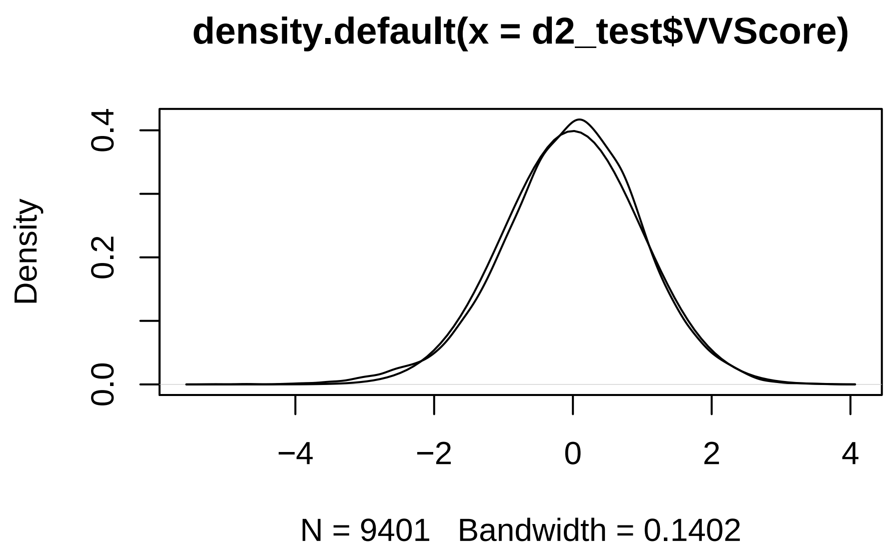
```
##          mean         sd        5.5%       94.5%
## a 546.3845380 0.47551725 545.6245696 547.1445064
## b   3.3580561 0.02766325   3.3138449   3.4022673
## d  91.3580120 0.33447878  90.8234503  91.8925737
## c   0.6396045 0.01740586   0.6117866   0.6674224
```

```
xbar <- mean(d2_train$BodyweightKg)
post <- extract.samples( varying_variance )
map_a <- mean(post$a)
map_b <- mean(post$b)
map_c <- mean(post$c)
map_d <- mean(post$d)
map_sigmas <- map_d + (map_c * ( d2_test$BodyweightKg - xbar) )
map_means <- map_a + (map_b * ( d2_test$BodyweightKg - xbar) )
```

Normalize to Z Score

```
d2_test$VVScore <- ( d2_test$TotalKg - map_means ) / (map_sigmas)
```

```
plot(density(d2_test$VVScore))
curve(dnorm(x,0,1),add=TRUE)
```



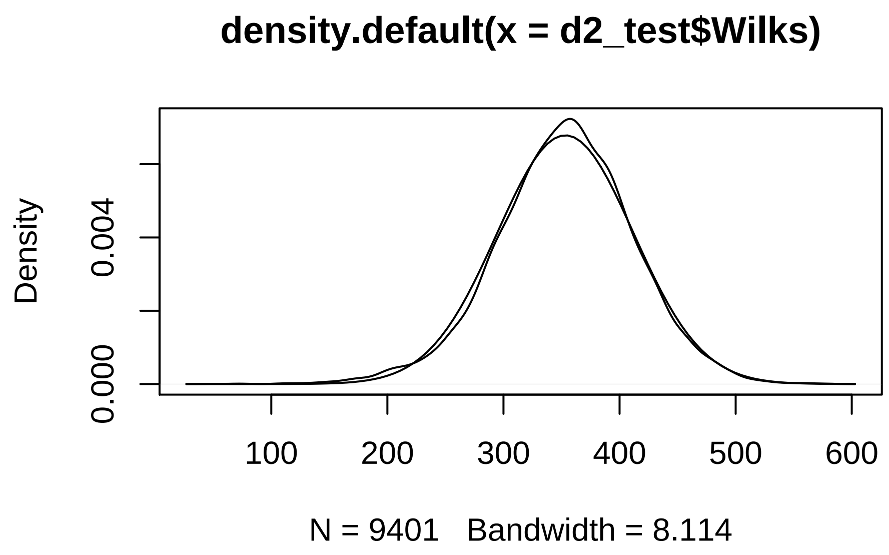
Wilks

```
Wilks_model <- quap(
  alist(
    Wilks ~ dnorm( mu , sigma ) ,
    mu <- dnorm( 300 , 50 ) ,
    sigma ~ dexp( 0.02 )

  ) , data=d2_train )
precis(Wilks_model)
```

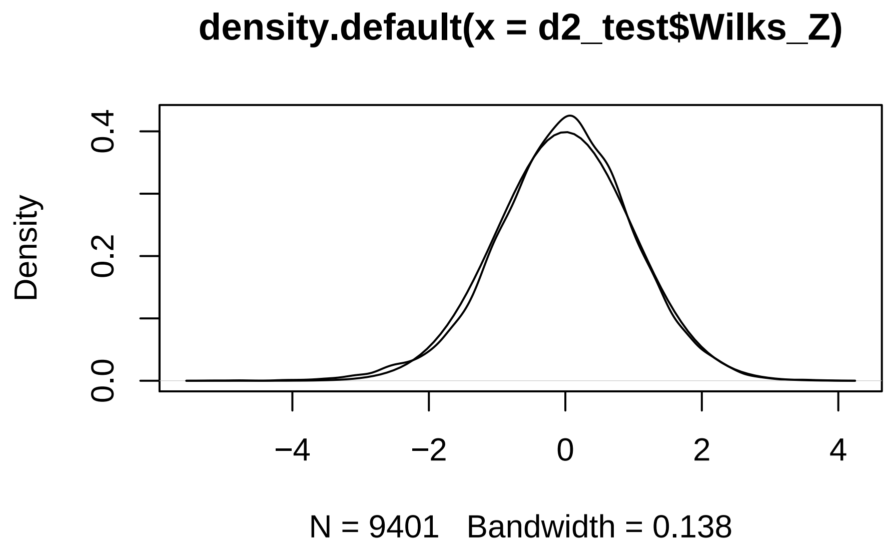
```
##             mean        sd      5.5%     94.5%
## mu    353.29754 0.3031112 352.81311 353.78197
## sigma  58.77809 0.2143286  58.43556  59.12063
```

```
post <- extract.samples( Wilks_model )
map_mu <- mean(post$mu)
map_sigma <- mean(post$sigma)
plot(density(d2_test$Wilks))
curve(dnorm(x,map_mu,map_sigma),add=TRUE)
```



## Normalize to Z Score

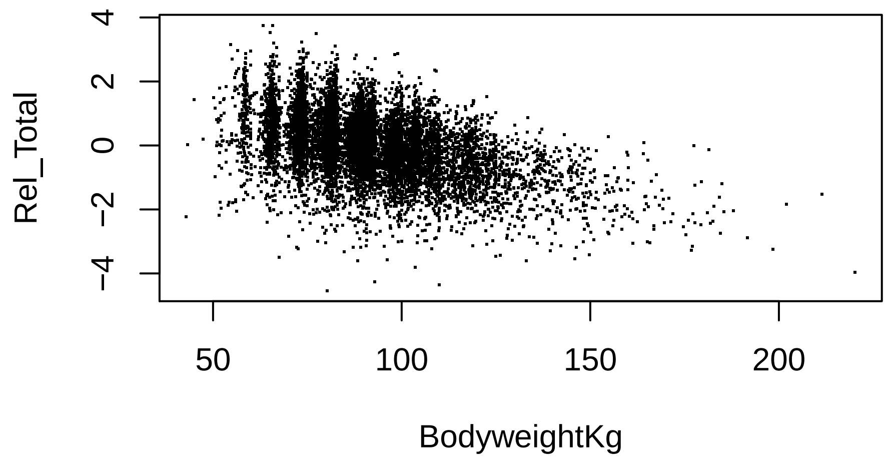
```
d2_test$Wilks_Z <- ( d2_test$Wilks - map_mu) / map_sigma
plot(density(d2_test$Wilks_Z))
curve(dnorm(x,0,1),add=TRUE)
```



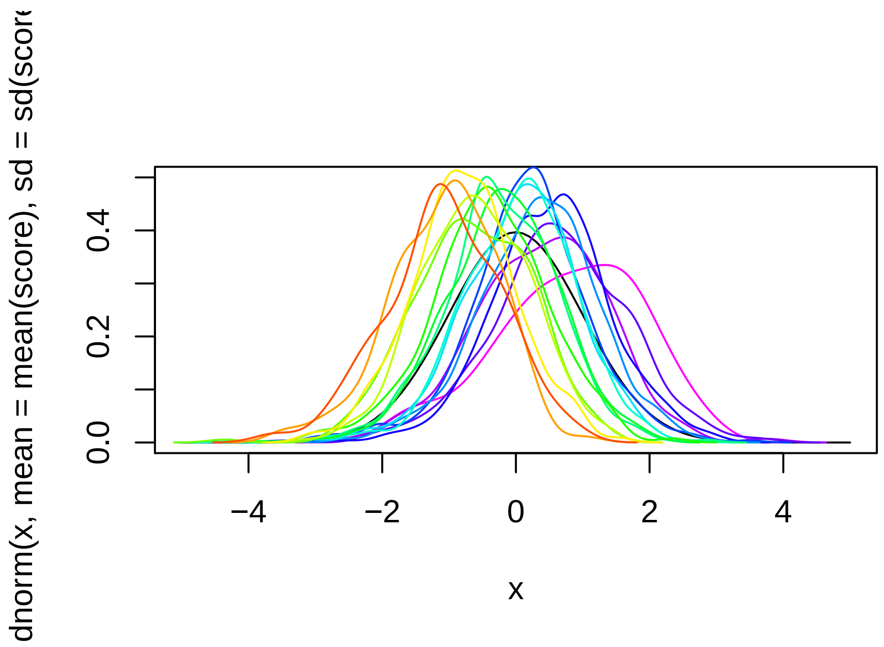
Plots Comparing Heteroskedasticity in Scores

Relative Bodyweight

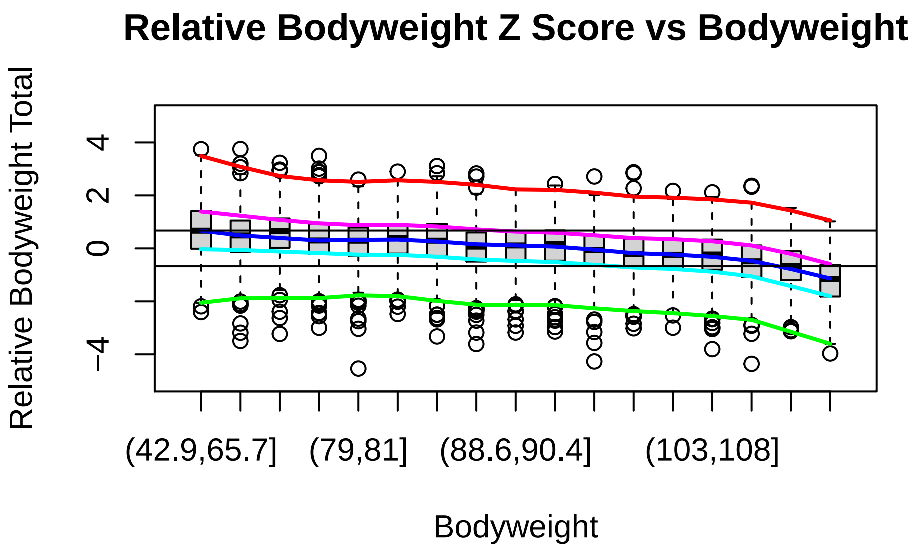
```
#par(mfrow=c(3,1))
plot(Rel_Total~BodyweightKg,data=d2_test,pch=".",cex=0.001)
```



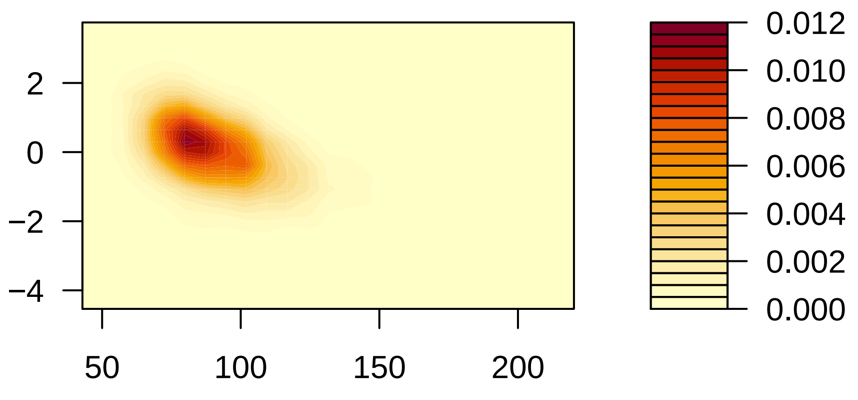
```
ColouredDensities( d2_test$Rel_Total,d2_test$logBW)
```



```
WanderingSchematic(17,d2_test$Rel_Total, d2_test$BodyweightKg ,
                   "Relative Bodyweight Z Score vs Bodyweight",
                   "Relative Bodyweight Total")
```

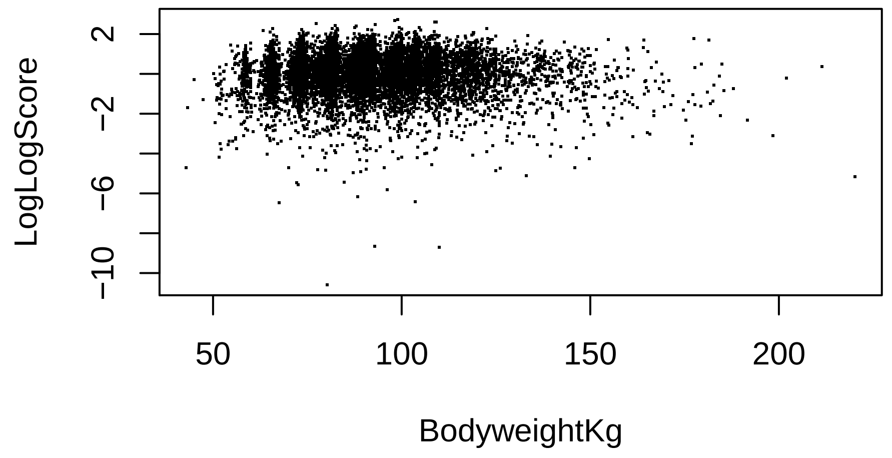


```
filled.contour(kde2d(d2_test$BodyweightKg,d2_test$Rel_Total))
```

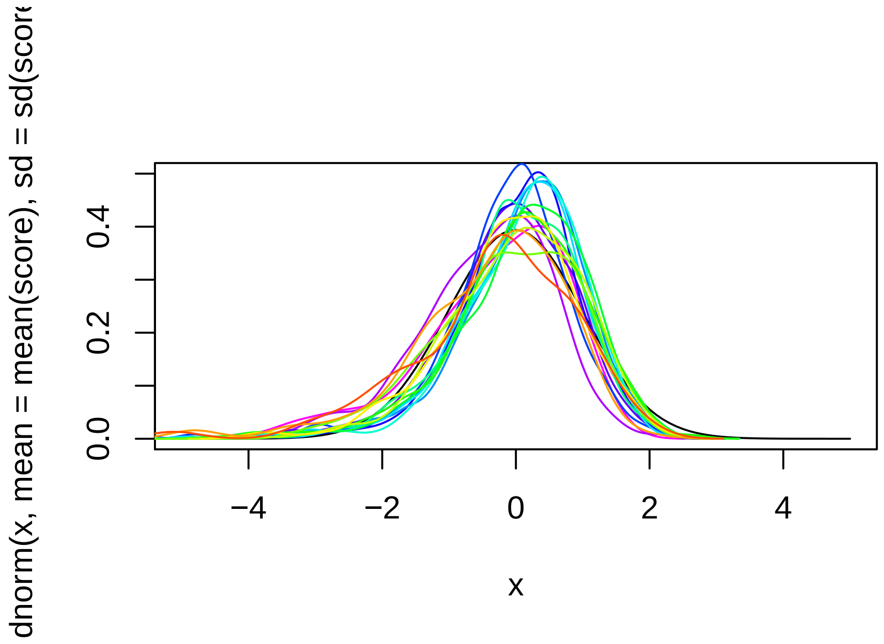


Log Log

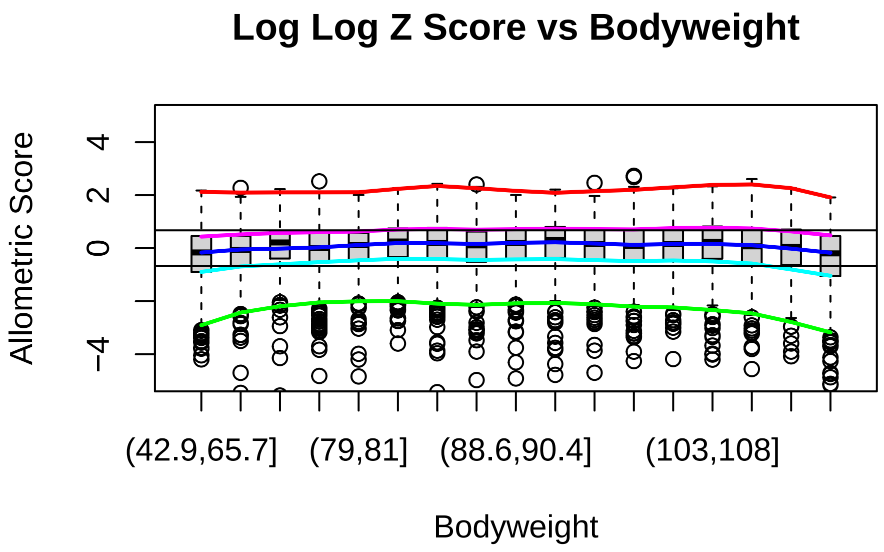
```
#par(mfrow=c(2,2))
plot(LogLogScore~BodyweightKg,data=d2_test,pch=".",cex=0.001)
```



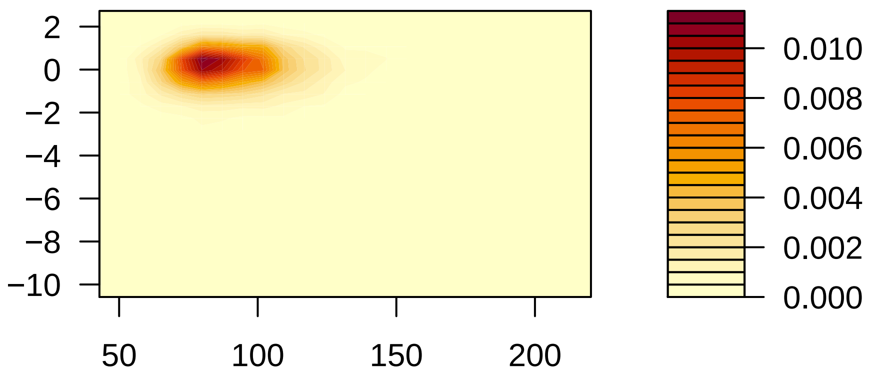
```
ColouredDensities( d2_test$LogLogScore,d2_test$logBW)
```



```
WanderingSchematic(17,d2_test$LogLogScore, d2_test$BodyweightKg ,
                   "Log Log Z Score vs Bodyweight", "Allometric Score")
```

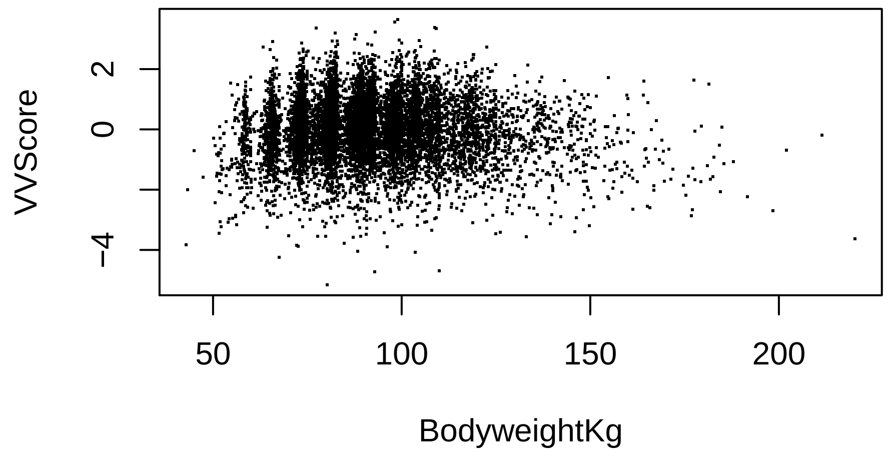


```
filled.contour(kde2d(d2_test$BodyweightKg,d2_test$LogLogScore))
```

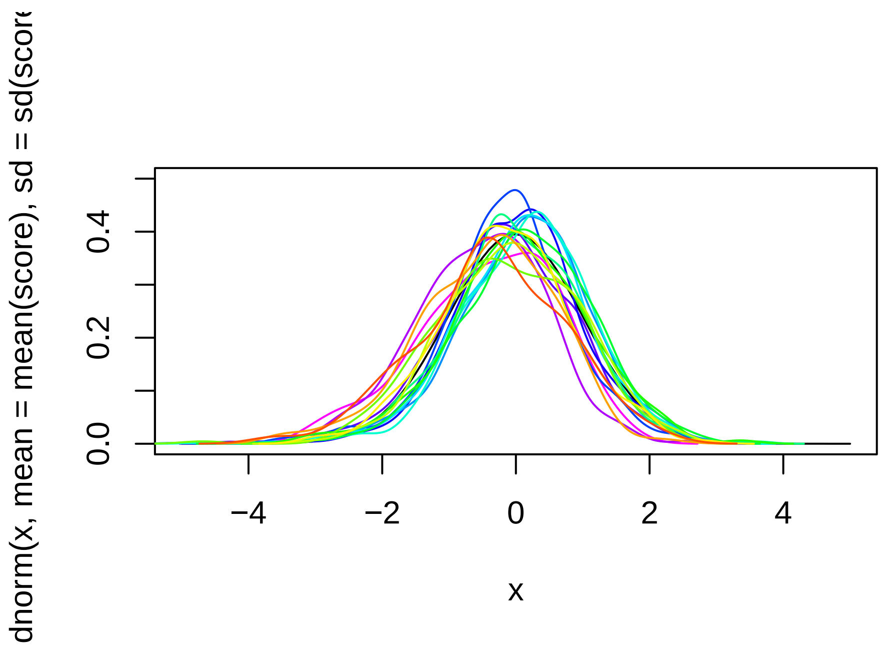


Varying Variance

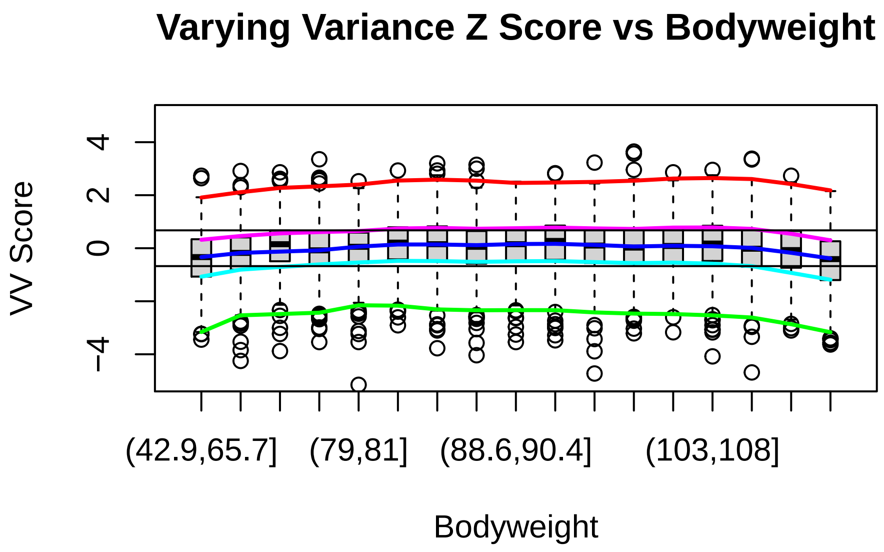
```
#par(mfrow=c(2,2))
plot(VVScore~BodyweightKg,data=d2_test,pch=".",cex=0.001)
```



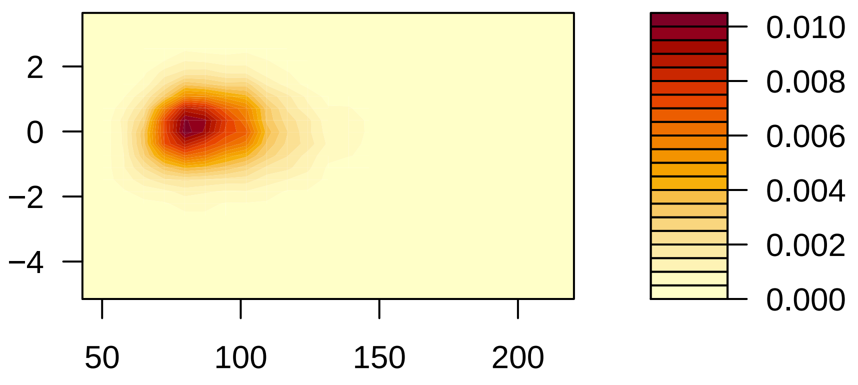
```
ColouredDensities( d2_test$VVScore,d2_test$logBW)
```



```
WanderingSchematic(17,d2_test$VVScore, d2_test$BodyweightKg ,
                   "Varying Variance Z Score vs Bodyweight", "VV Score")
```

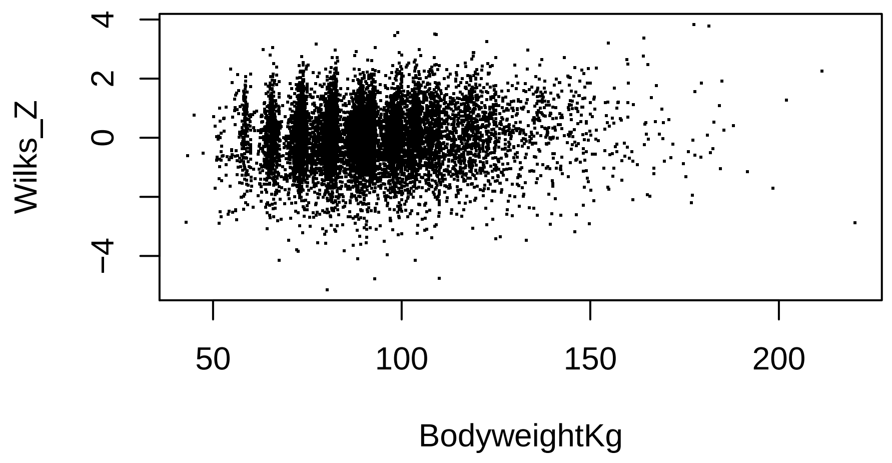


```
filled.contour(kde2d(d2_test$BodyweightKg,d2_test$VVScore))
```

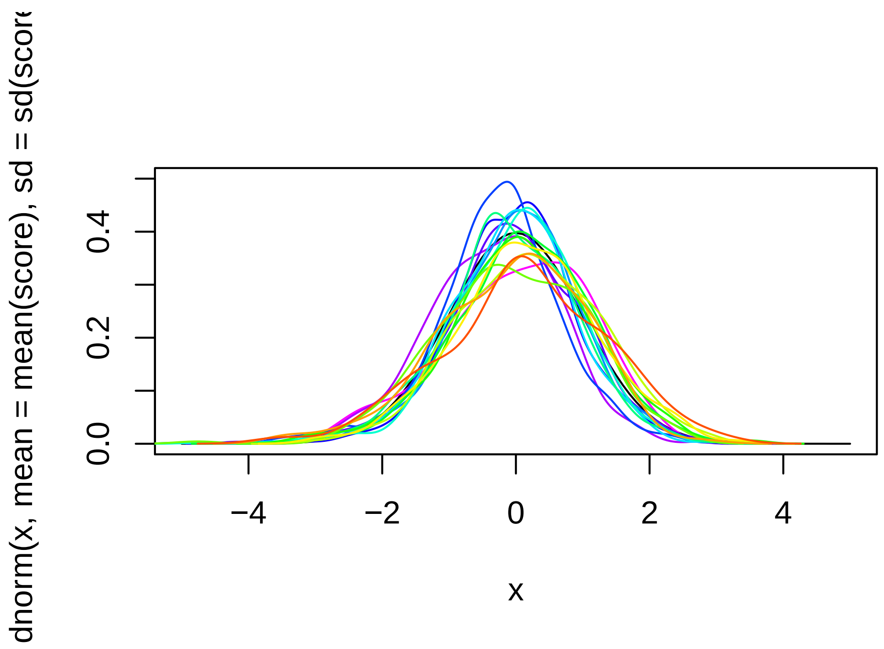


Wilks

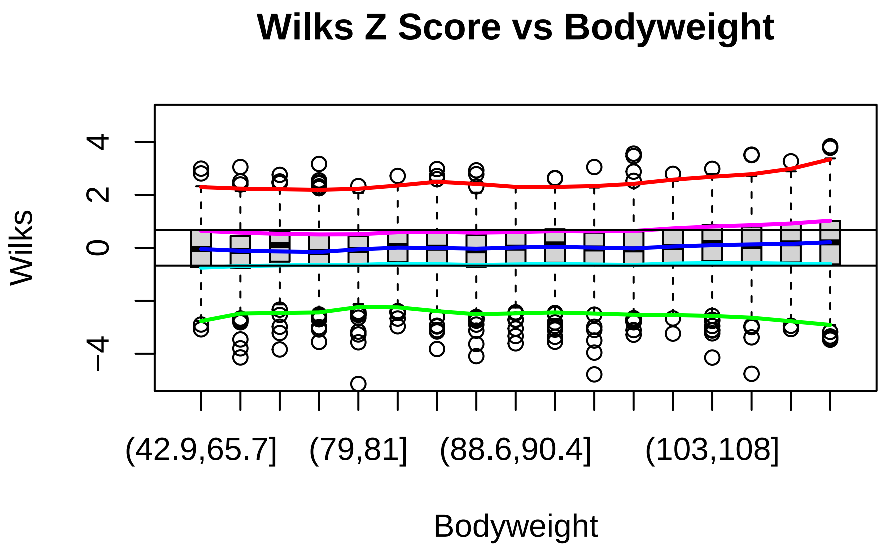
```
#par(mfrow=c(2,2))
plot(Wilks_Z~BodyweightKg,data=d2_test,pch=".",cex=0.001)
```



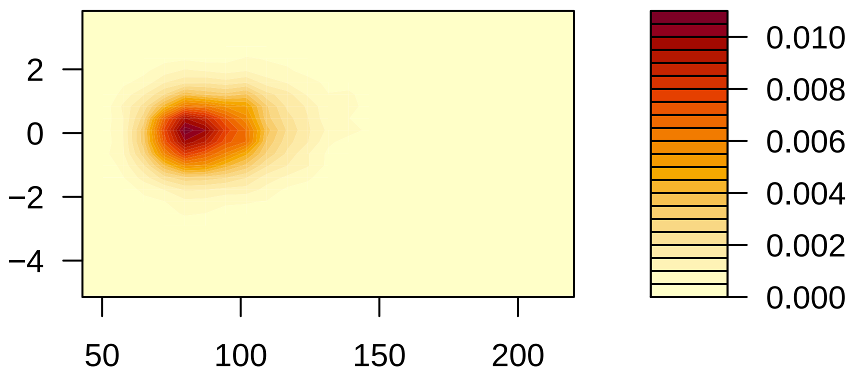
```
ColouredDensities( d2_test$Wilks_Z,d2_test$logBW)
```



```
WanderingSchematic(17,d2_test$Wilks_Z, d2_test$BodyweightKg ,
                   "Wilks Z Score vs Bodyweight", "Wilks")
```



```
filled.contour(kde2d(d2_test$BodyweightKg,d2_test$Wilks_Z))
```



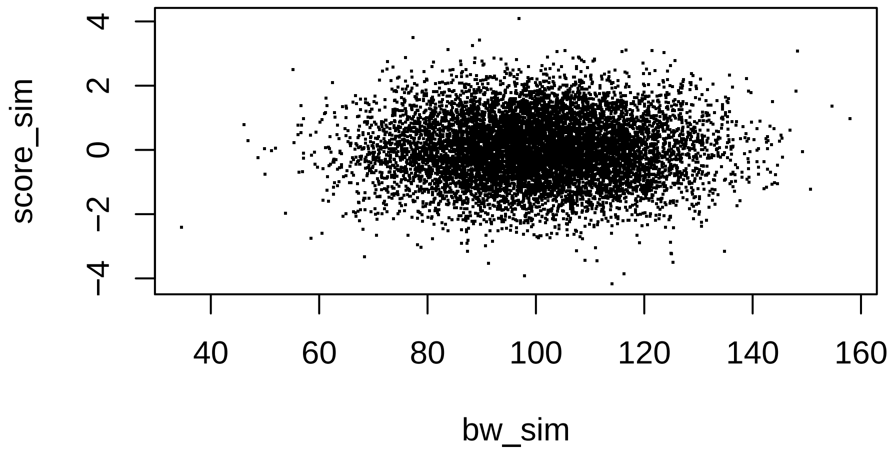
Fair Score

Question: what does a truly fair score look like?

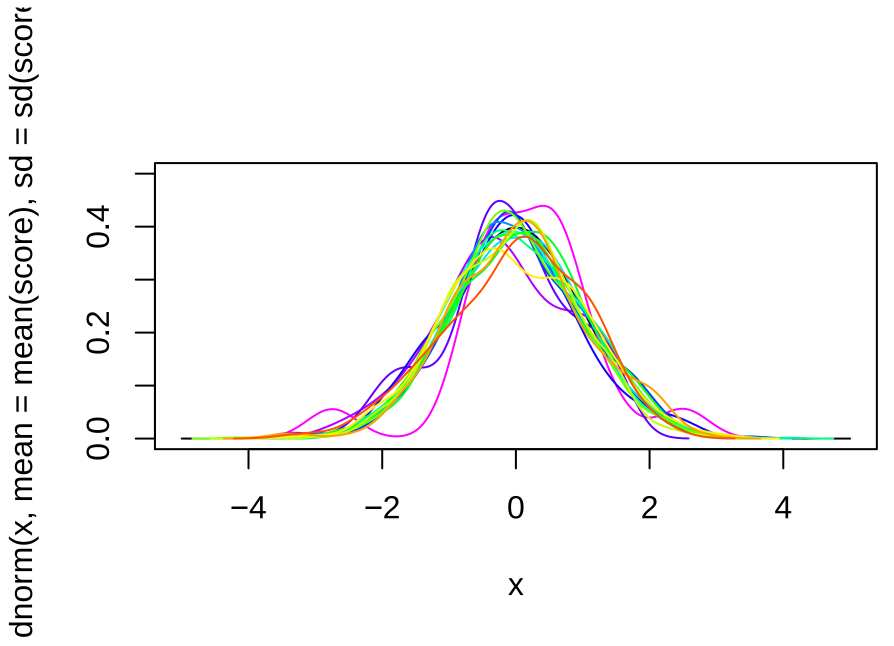
Simulation:

```r
n_sim <- nrow(d2_test)
bw_sim <- rnorm(n_sim,100,15)
score_sim <- rnorm(n_sim,0,1) #standard normal independent of bodyweight
```

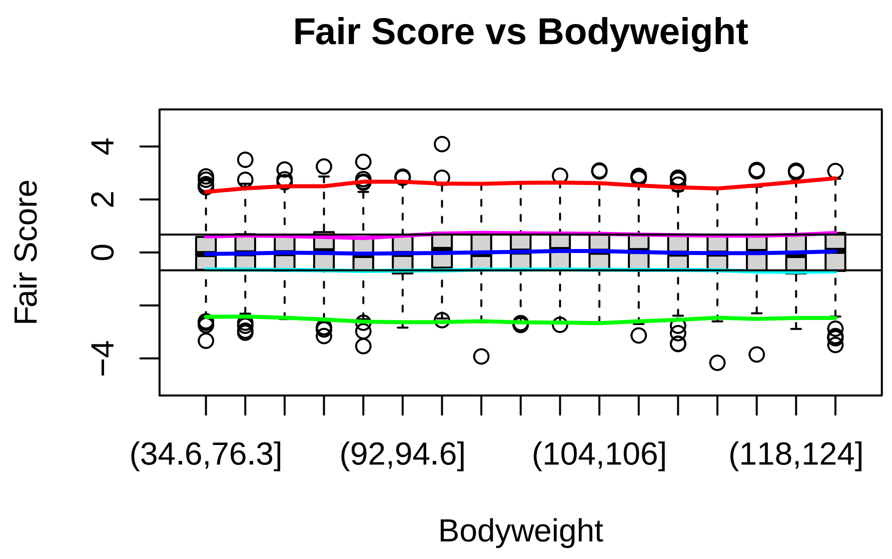
```r
#par(mfrow=c(2,2))
plot(score_sim~bw_sim,pch=".",cex=0.001)
```



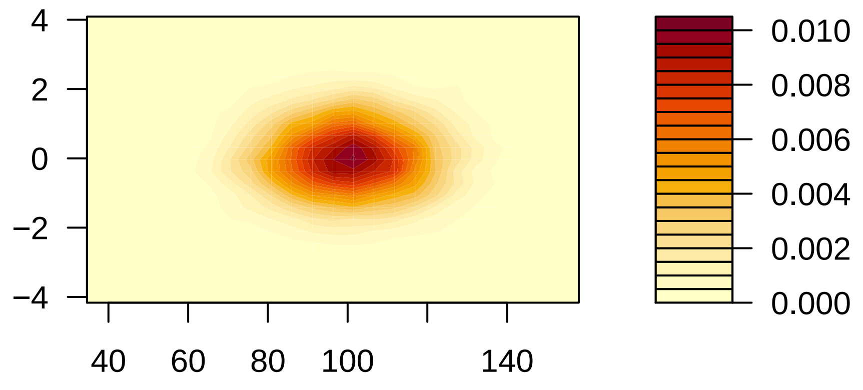
```
ColouredDensities( score_sim,log(bw_sim))
```



```
WanderingSchematic(17,score_sim, bw_sim ,
                   "Fair Score vs Bodyweight", "Fair Score")
```



```
filled.contour(kde2d(bw_sim,score_sim))
```



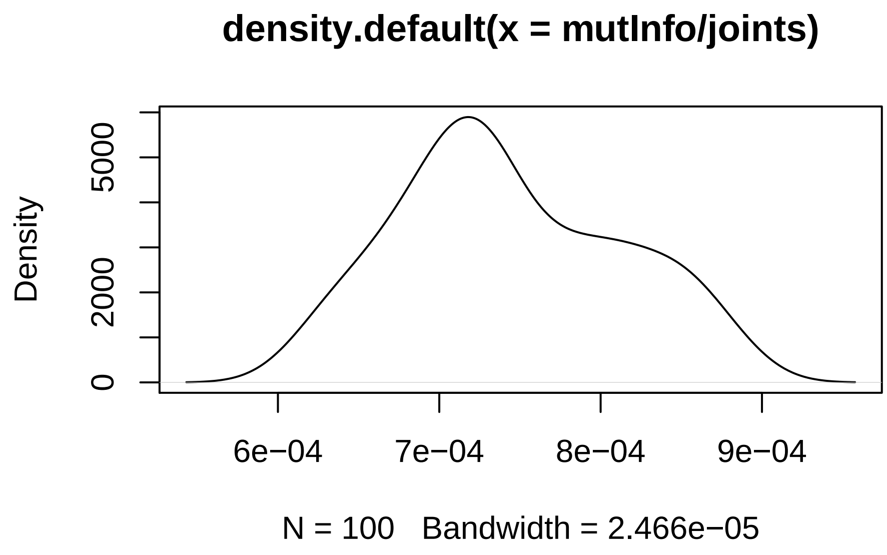
The most telling graph is probably the Wandering Schematic which is much better than any of the real scores used.

Under the null hypothesis that score is independent of bodyweight, what is the distribution of mutual information and information quality ratio?

```
sim_trials <- 100
mutInfo <- c()
joints <- c()
```

```
for(i in 1:sim_trials){
  #if(i%%floor(sim_trials/10)==0) print(i/sim_trials)
  n_sim <- nrow(d2_test)
  bw_sim <- rnorm(n_sim,mean(d2_test$BodyweightKg),sd(d2_test$BodyweightKg))
  score_sim <- rnorm(n_sim,0,1)
  bins <- floor(sqrt(n_sim))
  joint <- kde2d(bw_sim,score_sim,n=95)$z
  mutInfo <- append(mutInfo,mi.plugin(joint))
  joints <- append(joints,entropy(joint))

}
plot(density(mutInfo/joints))
```



```
quantile(mutInfo/joints)
```

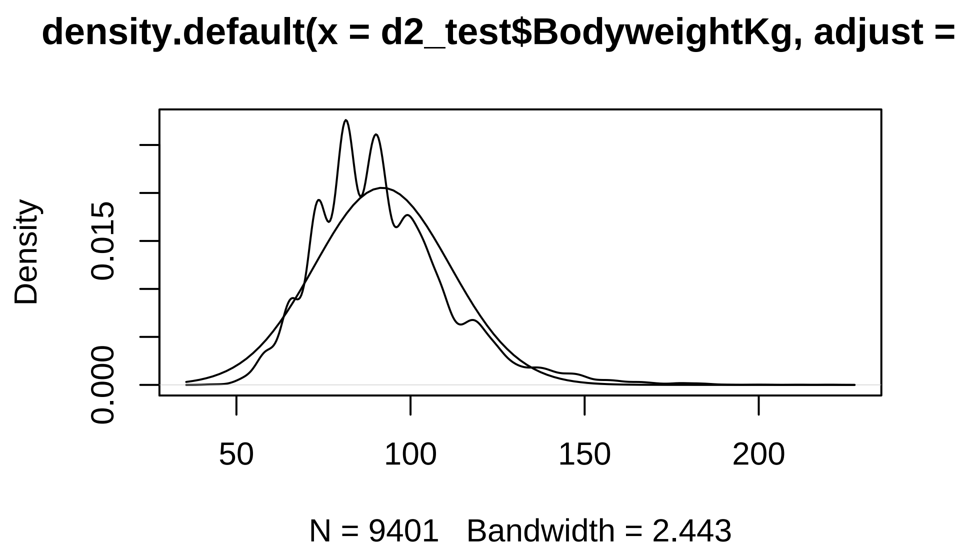
```
##           0%          25%          50%          75%         100% 
## 0.0006171895 0.0006946367 0.0007297699 0.0007989231 0.0008837246
```

```
HPDI(mutInfo/joints)
```

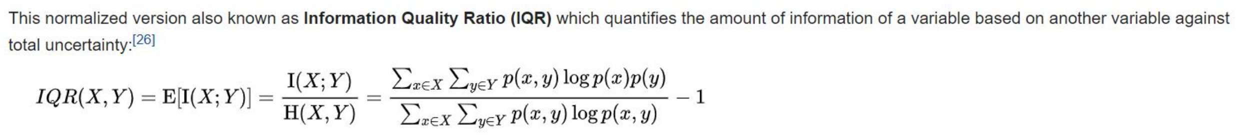
```
##        |0.89        0.89| 
## 0.0006366567 0.0008533794
```

Is the normality assumption in the simulation true?

```
plot(density(d2_test$BodyweightKg,adjust=1))
curve(dnorm(x,mean(d2_test$BodyweightKg),sd(d2_test$BodyweightKg)),add=TRUE)
```



Score comparison using mutual information


This normalized version also known as **Information Quality Ratio (IQR)** which quantifies the amount of information of a variable based on another variable against total uncertainty:[26]

$$IQR(X,Y) = \mathrm{E}[\mathrm{I}(X;Y)] = \frac{\mathrm{I}(X;Y)}{\mathrm{H}(X,Y)} = \frac{\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x)p(y)}{\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x,y)} - 1$$

Figure 1: According to wikipedia, dividing by the joint entropy is a normalized version of MI known as IQR

As shown by the simulation and a priori reasoning, if the score is fair then the MI or IQR should approach 0

```
rel_joint <- kde2d(d2_test$BodyweightKg,d2_test$Rel_Total,n=95)$z
LL_joint <- kde2d(d2_test$BodyweightKg,d2_test$LogLogScore,n=95)$z
VV_joint <- kde2d(d2_test$BodyweightKg,d2_test$VVScore,n=95)$z
Wilks_joint <- kde2d(d2_test$BodyweightKg,d2_test$Wilks_Z,n=95)$z
mi.plugin(rel_joint) / entropy(rel_joint)
```

```
## [1] 0.01775084
```

```
mi.plugin(LL_joint) / entropy(LL_joint)
```

```
## [1] 0.003149953
```

```
mi.plugin(VV_joint) / entropy(VV_joint)
```

```
## [1] 0.003344473
```

```
mi.plugin(Wilks_joint) / entropy(Wilks_joint)
```

```
## [1] 0.002427462
```

```
HPDI(mutInfo/joints)
```

```
##        |0.89        0.89|
## 0.0006366567 0.0008533794
```

All scores are well above the 89% interval for the information quality ratio. It appears that Wilks is slightly better than Allometric or varying variance scores. Relative Bodyweight Score is of course horrendous.