A guide to robust statistical methods in neuroscience: Figure 1

Rand R. Wilcox & Guillaume A. Rousselet

2023-03-02

Contents

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code. Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Dependencies

```
library(ggplot2)
library(cowplot)
library(tidyr)
library(tibble)
source("./code/Rallfun-v40.txt")
source("./code/theme_gar.txt")
```
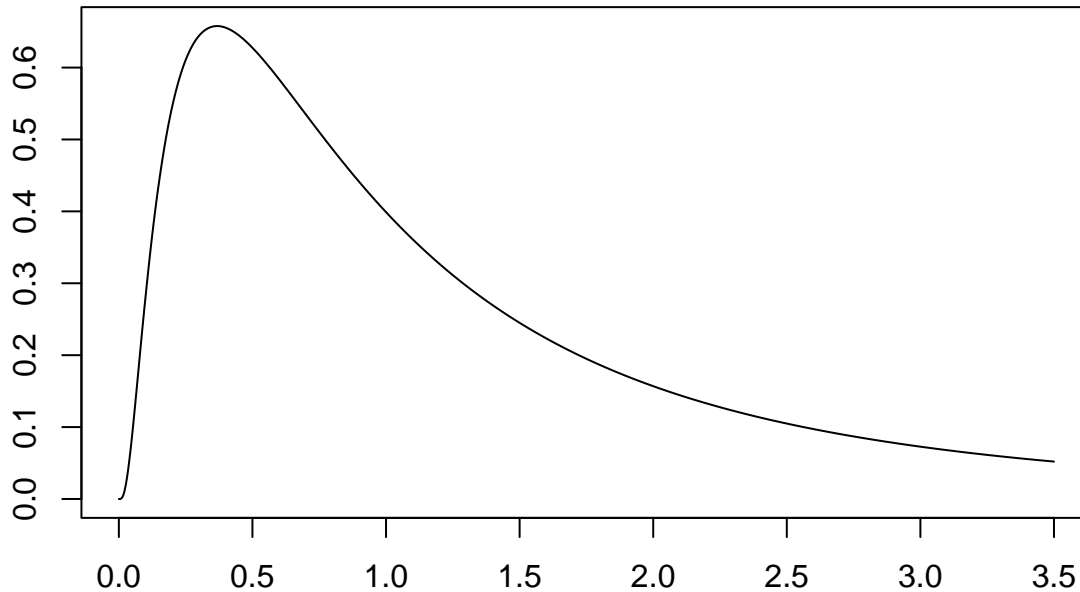
Make figure

Panel A: lognormal distribution

Get data

```
n <- 30
x <- c(0:3500)/1000
y <- dlnorm(x)
```

Base R figure

```
plot(x,y,type='n',xlab='',ylab='')
lines(x,y)
```


ggplot2 version

```
df <- tibble(x,y) # create data frame

pA <- ggplot(df, aes(x,y)) + theme_gar +
  geom_line(size=0.5) +
  scale_linetype_manual(values=c("solid")) +
  scale_color_manual(values=c('black'))+
  scale_x_continuous(breaks=seq(0,3.5,0.5)) +
  theme(axis.title.x = element_text(size = 18),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.title.y = element_text(size = 18),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank()) +
  labs(x = "Population values", y = "Density")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```

```
pA
```
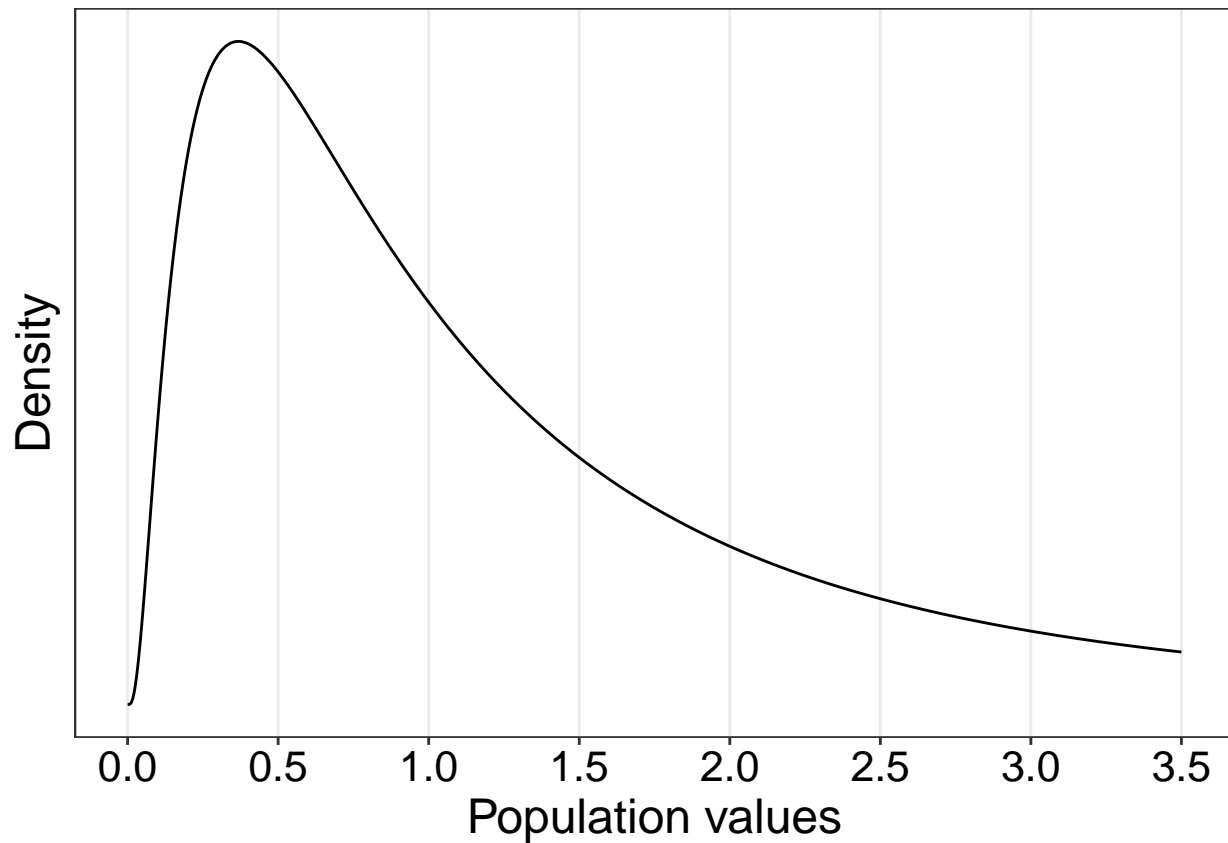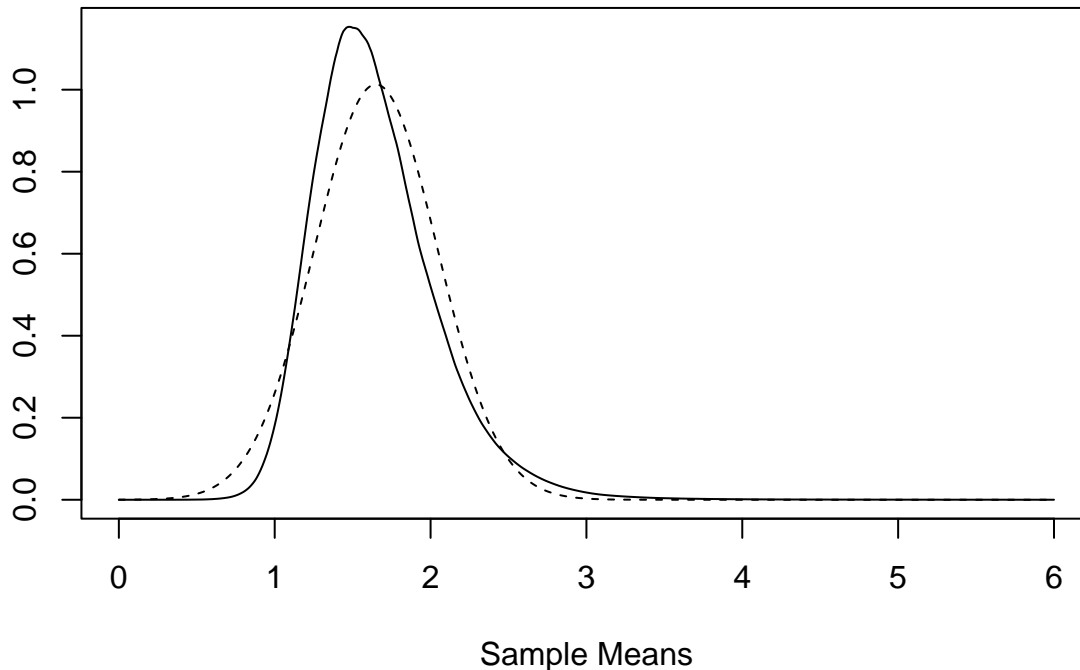
Panel B: distribution of sample means

Get data

```r
n <- 30
set.seed(45)
v=NA
nsim <- 50000
# Distribution of sample means from lognormal population
for(i in 1:nsim){
  v[i] <- mean(rlnorm(n))
}
# Distributions of sample means from normal population
m1 <- mean(v)
SD1 <- sd(v)
x <- c(0:600 / 100)
val1 <- dnorm(x, m1, SD1)
# fval <- akerd(v, xlab = 'Sample Mean', pyhat = TRUE, pts = x)
# save(fval, file = "./data/fig01B.RData")
```

Base R figure

```r
xlab <- 'Sample Means'
ylab <- ""
```

```r
load(file = "./data/fig01B.RData")
plot(x, fval, type="n",ylab=ylab,xlab=xlab)
lines(x, fval, col="black")
lines(x, val1, lty=2)
```


ggplot2 version

```r
df <- tibble(x,`Lognormal`=fval,`Normal`=val1) # create data frame
df <- tidyr::gather(df,Distribution,y,2:3)

# make plot
df$Distribution <- as.factor(df$Distribution)

pB <- ggplot(df, aes(x,y, group=Distribution)) + theme_gar +
  geom_line(aes(linetype=Distribution, colour=Distribution, size=Distribution)) +
  scale_size_manual(values=c(1,0.5)) +
  scale_linetype_manual(values=c("solid","solid")) +
  scale_colour_manual(values=c('black','grey50')) +
  scale_x_continuous(breaks=seq(0,6,1)) +
  coord_cartesian(xlim = c(0, 5)) +
  theme(axis.title = element_text(size = 18),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        legend.key.width = unit(1.5,"cm"),
        legend.position = c(0.7,0.8),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank()) +
  labs(x = "Sample Means", y = "Density",
       linetype="n=30", colour="n=30", size="n=30")
```
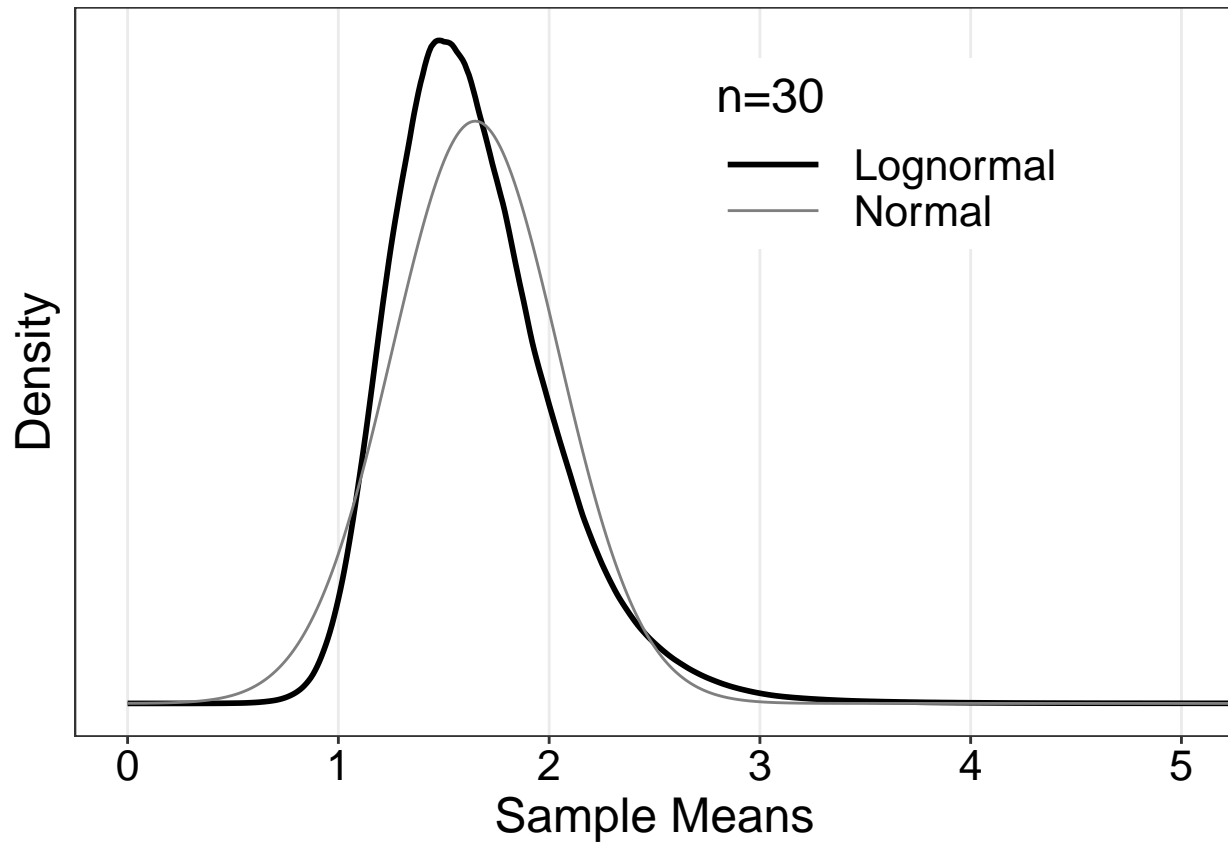
Panel C: T dist. n=30

Get data

```
n <- 30
tv <- NA
set.seed(45)
mpop <- exp(0.5) # mean of the standard lognormal distribution

for(i in 1:nsim){
  tv[i] <- trimci(rlnorm(n) - mpop, tr=0, pr=FALSE)$test.stat
}

x <- c(-800:400) / 100
# tdis <- akerd(tv,xlab='T',pyhat=TRUE,plotit=FALSE,pts=x)
# save(tdis, file = "./data/fig01C.RData")
DT <- dt(x, df=n-1) # normal population
# DT[x<(-3)] <- NA
```

Base R figure

```
load(file = "./data/fig01C.RData")
plot(c(x, x), c(tdis,DT), xlab='T', type='n', ylab='')
lines(x, DT)
lines(x, tdis, lty=2)
```
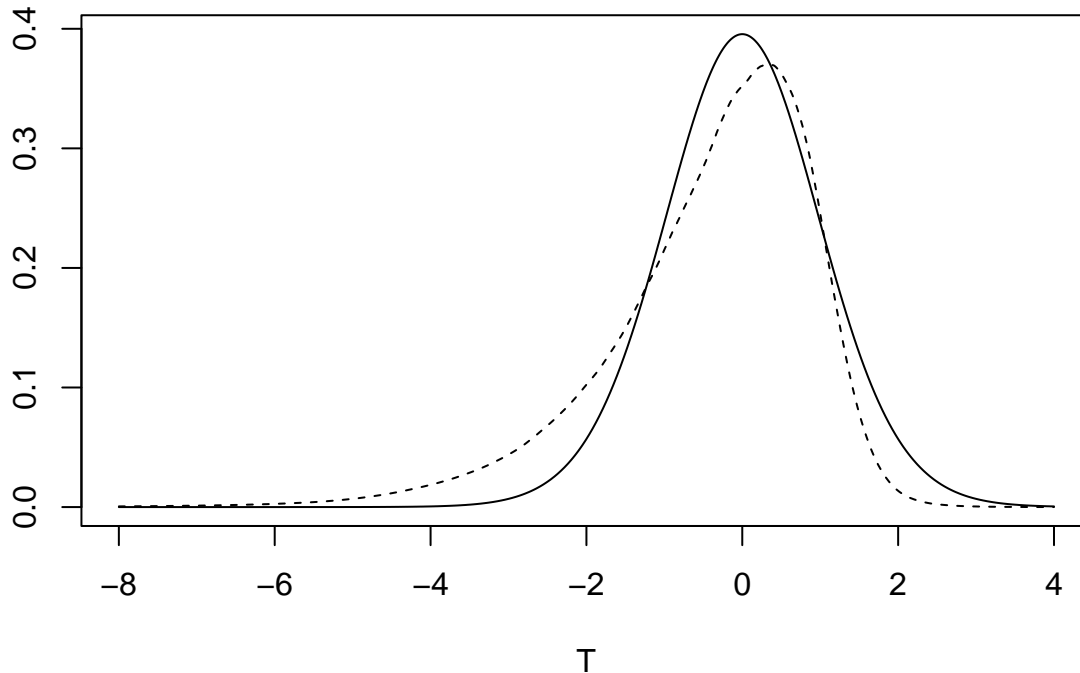


```
# hd(tv,.025)  -3.704622
# hd(tv,.975)   1.455312
# mean(tv<=qt(.025,29))   .111
```

ggplot2 version

```
df <- tibble(x,`Lognormal`=tdis,`Normal`=DT) # create data frame
df <- tidyr::gather(df,Distribution,y,2:3)

# make plot
df$Distribution <- as.factor(df$Distribution)

pC <- ggplot(df, aes(x,y, group=Distribution)) + theme_gar +
  geom_line(aes(linetype=Distribution, colour=Distribution, size=Distribution)) +
  scale_size_manual(values=c(1,0.5)) +
  scale_linetype_manual(values=c("solid","solid")) +
  scale_color_manual(values=c('black','grey50')) +
  scale_x_continuous(breaks=seq(-8,4,1)) +
  coord_cartesian(xlim = c(-6, 3.5)) +
  theme(axis.title = element_text(size = 18),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        legend.key.width = unit(1.5,"cm"),
```
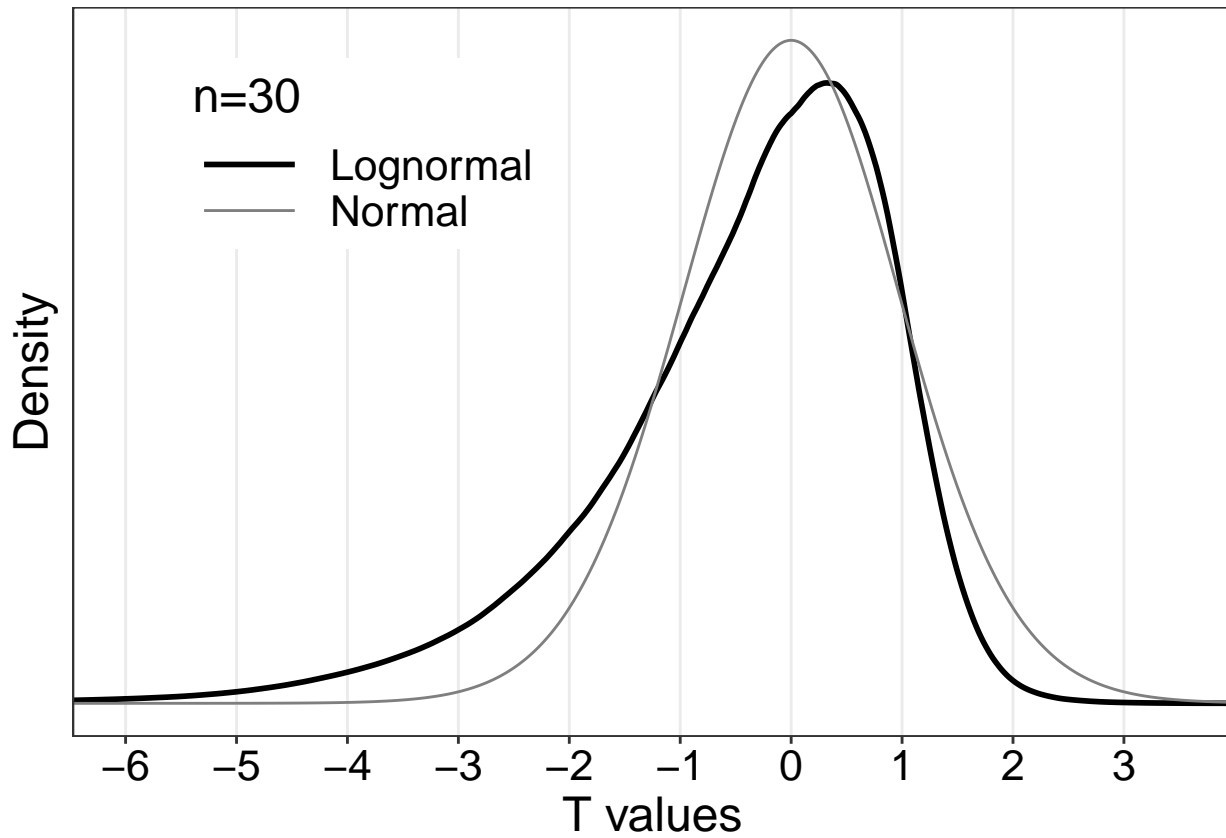
```
        legend.position = c(0.25,0.80),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank()) +
  labs(x = "T values", y = "Density",
       linetype="n=30", colour="n=30", size="n=30")
pC
```
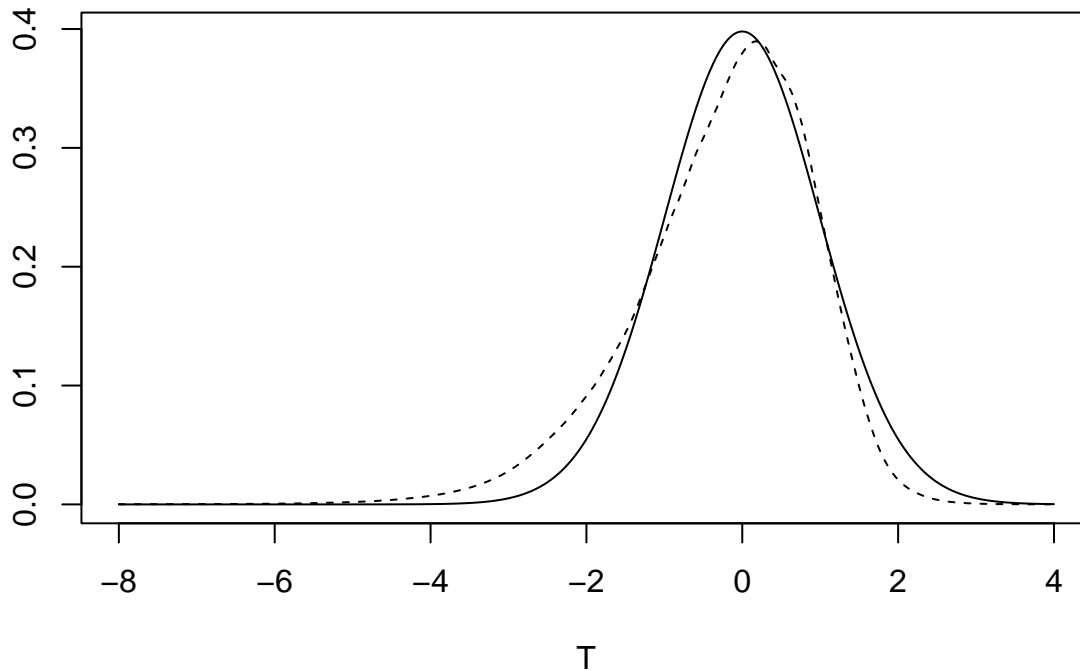

Panel D: T dist. n=100

Get data

```
tv <- NA
n <- 100
set.seed(45)
for(i in 1:nsim){
  tv[i]=trimci(rlnorm(n)-sqrt(exp(1)), tr=0, pr=FALSE)$test.stat
}
x <- c(-800:400)/100
# tdis <- akerd(tv,xlab='T',pyhat=TRUE,plotit=FALSE,pts=x)
# save(tdis, file = "./data/fig01D.RData")
DT <- dt(x, df=n-1)
# DT[x<(-3)] <- NA
```

Base R figure

```r
load(file = "./data/fig01D.RData")
plot(c(x, x), c(tdis,DT), xlab='T', type='n', ylab='')
lines(x, DT)
lines(x, tdis, lty=2)
```



```r
#  hd(tv,.025)  -2.794811
# hd(tv,.975)  1.579935
# Type I = .082 based on 10,000 reps
```

ggplot2 version

```r
df <- tibble(x,`Lognormal`=tdis,`Normal`=DT) # create data frame
df <- tidyr::gather(df,Distribution,y,2:3)

# make plot
df$Distribution <- as.factor(df$Distribution)

pD <- ggplot(df, aes(x,y, group=Distribution)) + theme_gar +
  geom_line(aes(linetype=Distribution, colour=Distribution, size=Distribution)) +
  scale_size_manual(values=c(1,0.5)) +
  scale_linetype_manual(values=c("solid","solid")) +
  scale_color_manual(values=c('black','grey50')) +
  scale_x_continuous(breaks=seq(-8,3,1)) +
  coord_cartesian(xlim = c(-6, 3)) +
  theme(axis.title = element_text(size = 18),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        legend.key.width = unit(1.5,"cm"),
```
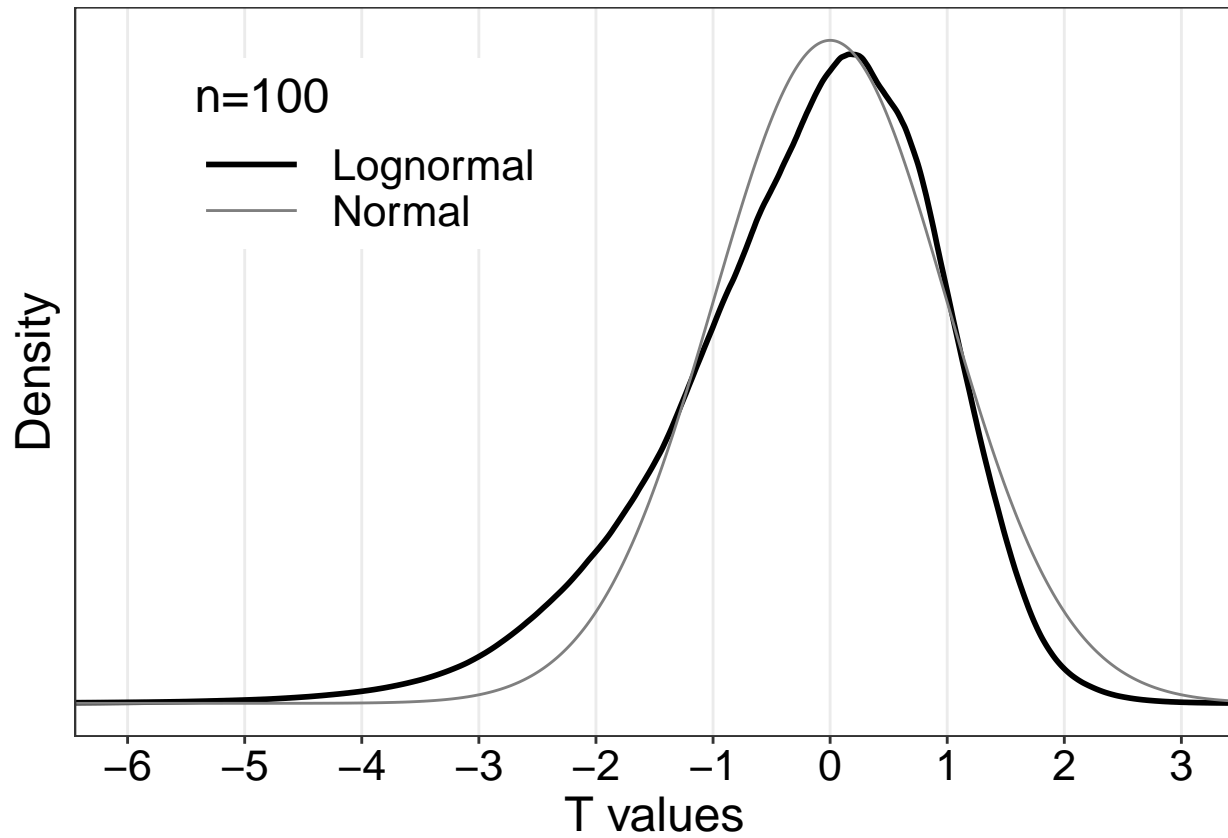
```
        legend.position = c(0.25,0.80),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank()) +
  labs(x = "T values", y = "Density",
       linetype="n=100", colour="n=100", size="n=100")
pD
```


Combine panels into one figure

```
cowplot::plot_grid(pA, pB, pC, pD,
                   labels=c("A", "B", "C", "D"),
                   ncol = 2,
                   nrow = 2,
                   rel_widths = c(1, 1, 1, 1),
                   label_size = 20,
                   hjust = -0.5,
                   scale=.95,
                   align = "h")

# save figure
ggsave(filename='./figures/figure1.pdf',width=10,height=7)
```