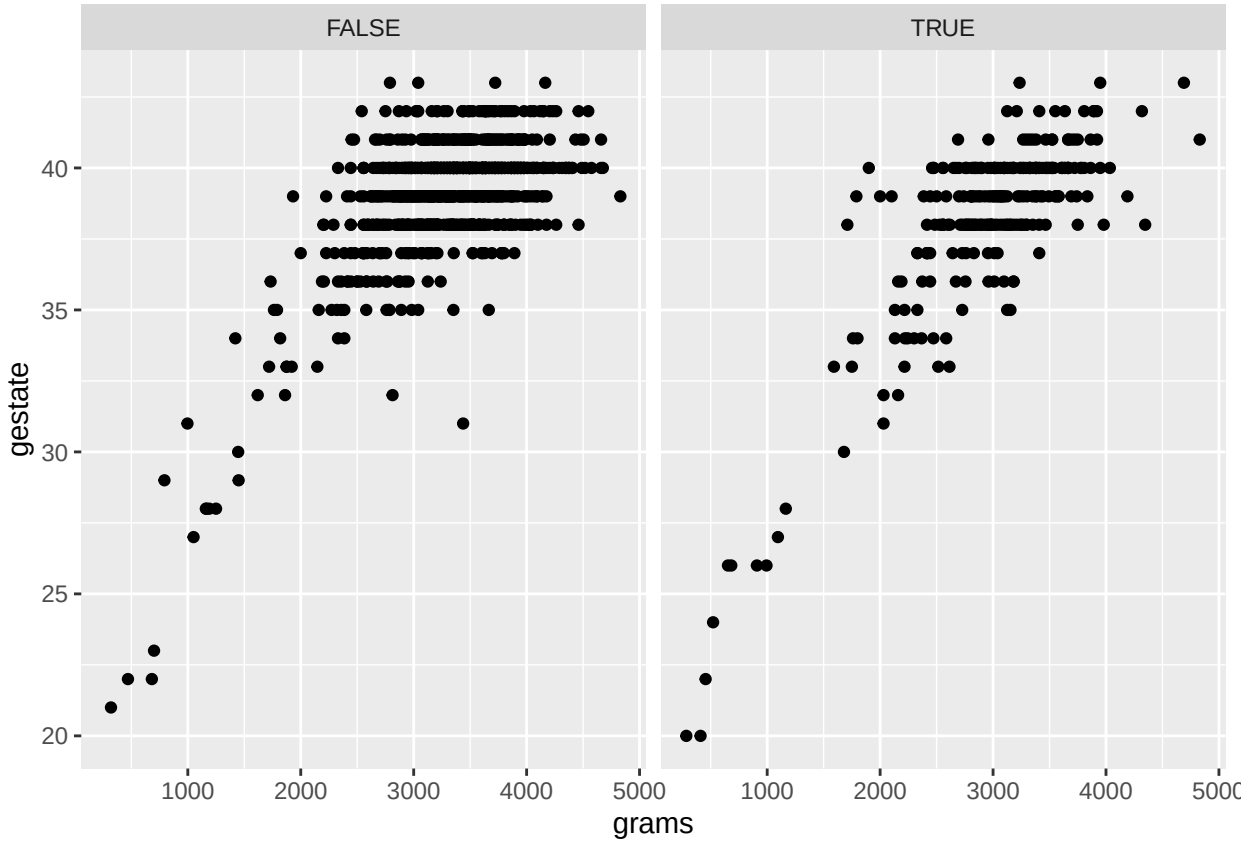
Assignment_5

Evan Z-G

2023-09-21

Chapter 5, Problem 2 Create two scatter plots of gestational length and birth weight, one for each smoking status

```
library(faraway)
data(phbirths)
#smoker <- phbirths %>%
#  group_by(smoke)
#smoke_yes <- filter(phbirths, smoke == TRUE)
#smoke_no <- filter(phbirths, smoke == FALSE)
ggplot(phbirths, aes(x=grams, y=gestate)) +
  geom_point() +
  facet_grid(cols = vars(smoke))
```

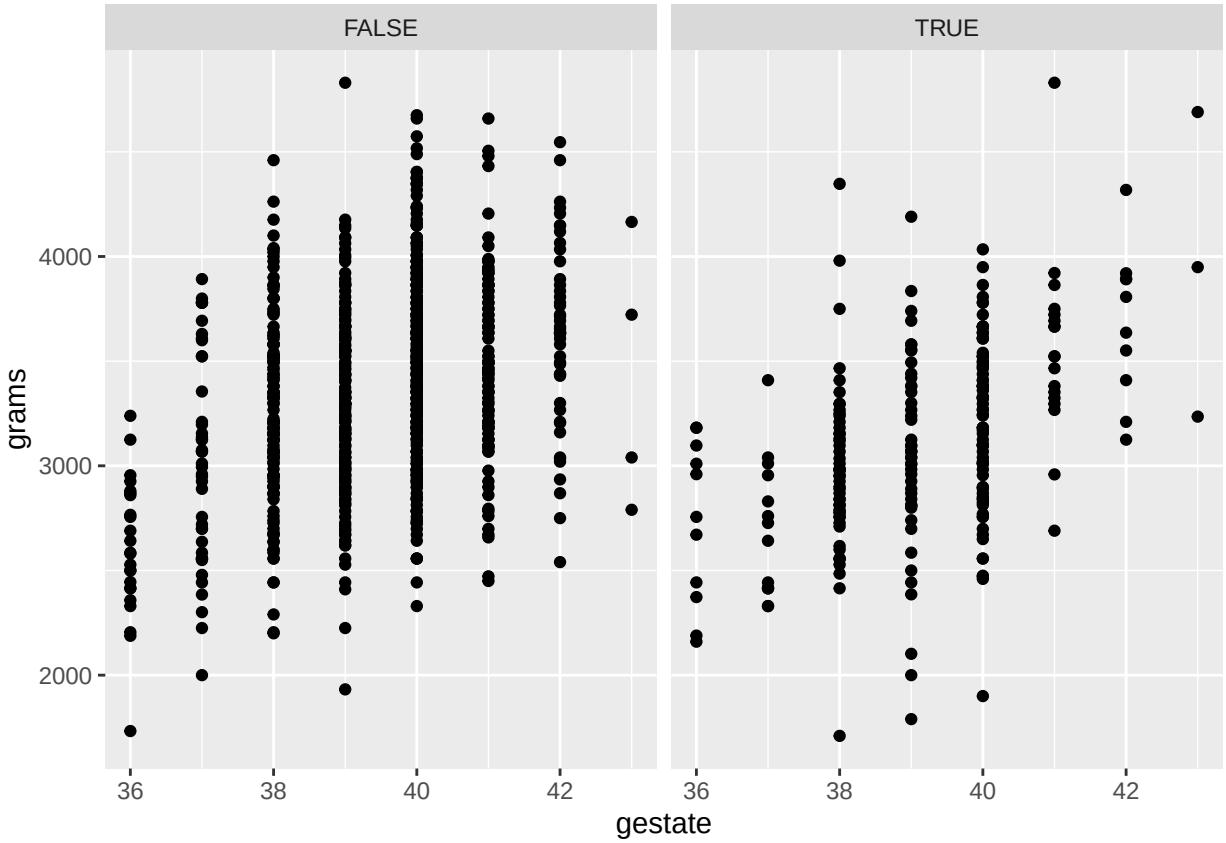



```
  #labs(x="Birth Weight (grams)", y="Gestational Length (weeks)")
```

Remove all the observations that are premature (less than 36 weeks). For the remainder of the problem, only

use these full-term babies.

```
mature <- phbirths %>% filter(., gestate > 35)

ggplot(mature, aes(y=grams, x=gestate)) +
  geom_point() +
  facet_grid(cols = vars(smoke))
```




```
head(mature)
```

```
##   black educ smoke gestate grams
## 1 FALSE    0  TRUE      40  2898
## 3 FALSE    2 FALSE      38  3977
## 4 FALSE    2  TRUE      37  3040
## 5 FALSE    2 FALSE      38  3523
## 6 FALSE    5  TRUE      40  3100
## 7  TRUE    6 FALSE      40  3670
```

Fit the quadratic model

```
model <- lm(grams ~ poly(gestate,2) * smoke, data=mature)
summary(model)
```

```
##
## Call:
## lm(formula = grams ~ poly(gestate, 2) * smoke, data = mature)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1433.51  -296.25   -12.25   291.68  1464.49
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  3364.02      15.45 217.751  < 2e-16 ***
## poly(gestate, 2)1            5770.90     504.11  11.448  < 2e-16 ***
## poly(gestate, 2)2           -2287.74     512.46  -4.464 8.92e-06 ***
## smokeTRUE                    -202.81      32.68  -6.206 7.85e-10 ***
## poly(gestate, 2)1:smokeTRUE  1813.07    1027.39   1.765 0.077904 .
## poly(gestate, 2)2:smokeTRUE  3654.80     988.42   3.698 0.000229 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 437.7 on 1033 degrees of freedom
## Multiple R-squared:  0.2108, Adjusted R-squared:  0.207
## F-statistic: 55.19 on 5 and 1033 DF,  p-value: < 2.2e-16
```

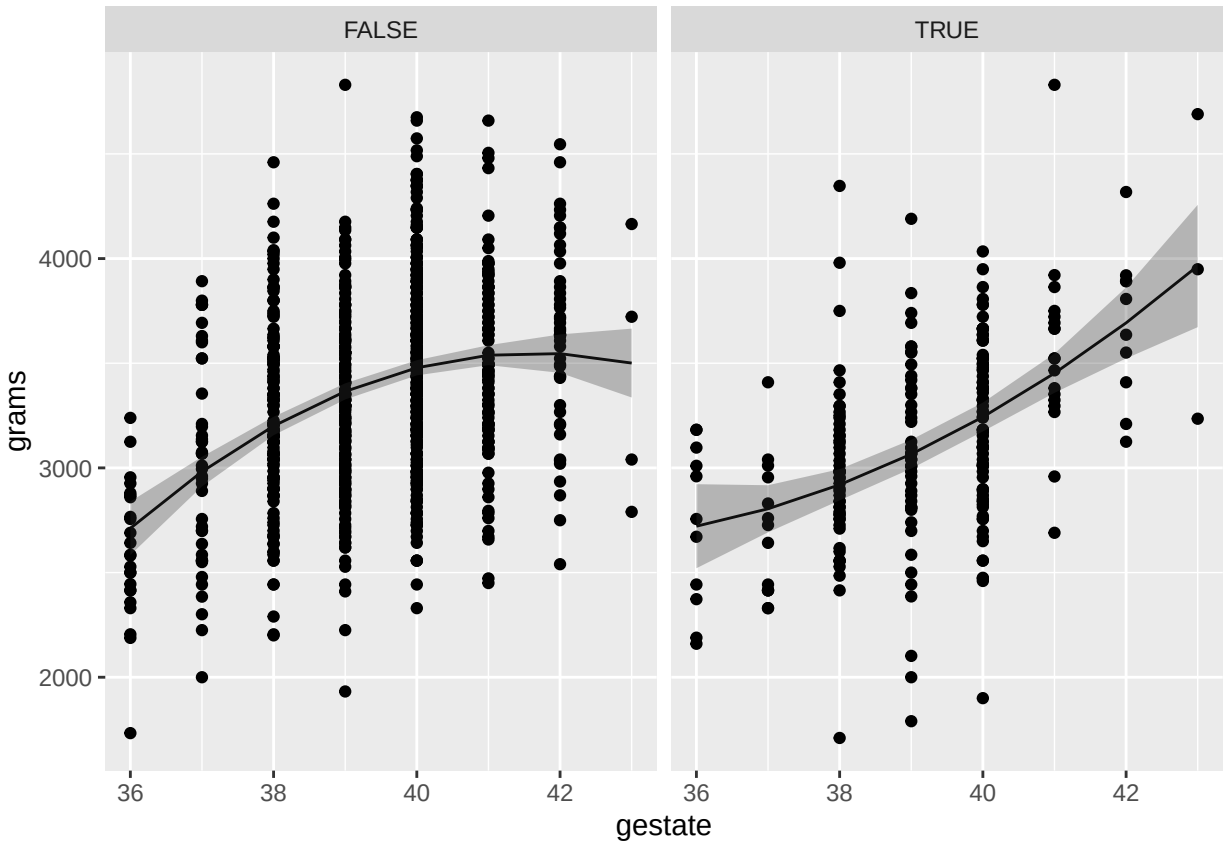
Add the model fitted values to the phbirths data frame along with the regression model confidence intervals.

```
mature <- mature %>%
  dplyr::select( -matches('fit'), -matches('lwr'), -matches('upr') ) %>%
  cbind( predict(model, newdata=., interval='confidence') )

head(mature)
```

```
##   black educ smoke gestate grams      fit      lwr      upr
## 1 FALSE    0  TRUE      40  2898 3243.132 3174.250 3312.013
## 3 FALSE    2 FALSE      38  3977 3200.173 3156.090 3244.256
## 4 FALSE    2  TRUE      37  3040 2804.668 2692.140 2917.196
## 5 FALSE    2 FALSE      38  3523 3200.173 3156.090 3244.256
## 6 FALSE    5  TRUE      40  3100 3243.132 3174.250 3312.013
## 7  TRUE    6 FALSE      40  3670 3478.249 3441.992 3514.507
```

On your two scatter plots from part (a), add layers for the model fits and ribbon of uncertainty for the model fits.

```
  ggplot(mature, aes(y=grams, x=gestate)) +
  geom_point() +
  facet_grid(cols = vars(smoke)) +
  geom_line( aes(y=fit) ) +
  geom_ribbon( aes( ymin=lwr, ymax=upr), alpha=.3 )
```



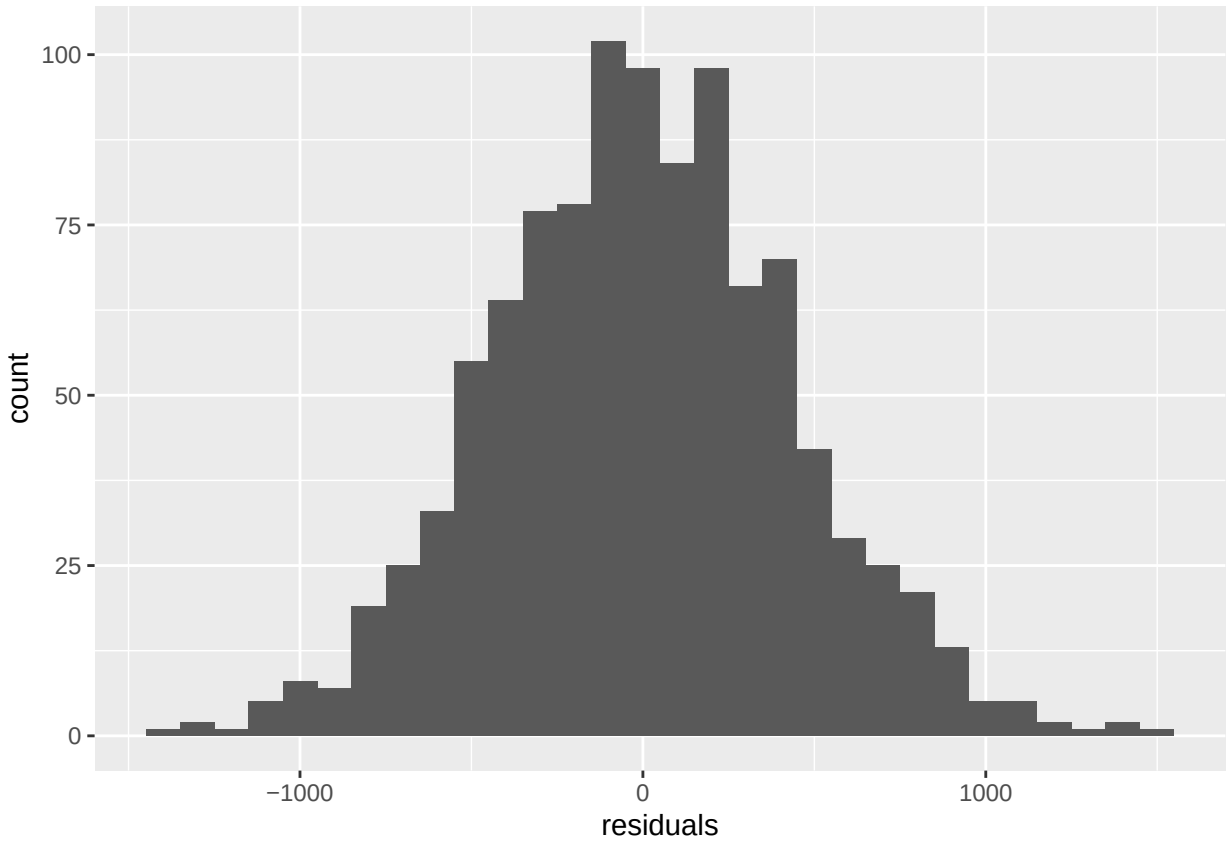
Create a column for the residuals in the phbirths data set using any of the following:

```
mature$residuals = resid(model)
```

Create a histogram of the residuals.

```
ggplot(mature, aes(x=residuals)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Chapter 6, Problem 2:

a.

```
a <- 4
b <- 10
x <- runif(n=1, 0,10)  # one random value between 0 and 10

if( x < a ){
  result <- 0     # Replace ???? with something appropriate!
}else if( x <= b ){
  result <- 1/(b-a)
}else{
  result <- 0
}
print(paste('x=',round(x,digits=3), '  result=', round(result,digits=3)))
```

```
## [1] "x= 3.355   result= 0"
```

b. i.

```
x <- runif(n=1, 0,10)  # one random value between 0 and 10
if( (a<=x) & (x<=b) ){
  result <- 1/(b-a)
}else{
  result <- 0
}
print(paste('x=',round(x,digits=3), '  result=', round(result,digits=3)))
```

```
## [1] "x= 8.306   result= 0.167"
```

b. ii.

```
x <- runif(n=1, 0,10)  # one random value between 0 and 10
if( (x<a) | (b<x) ){
  result <- 0
}else{
  result <- 1/(b-a)
}
print(paste('x=',round(x,digits=3), '  result=', round(result,digits=3)))
```

```
## [1] "x= 5.879   result= 0.167"
```

b. iii.

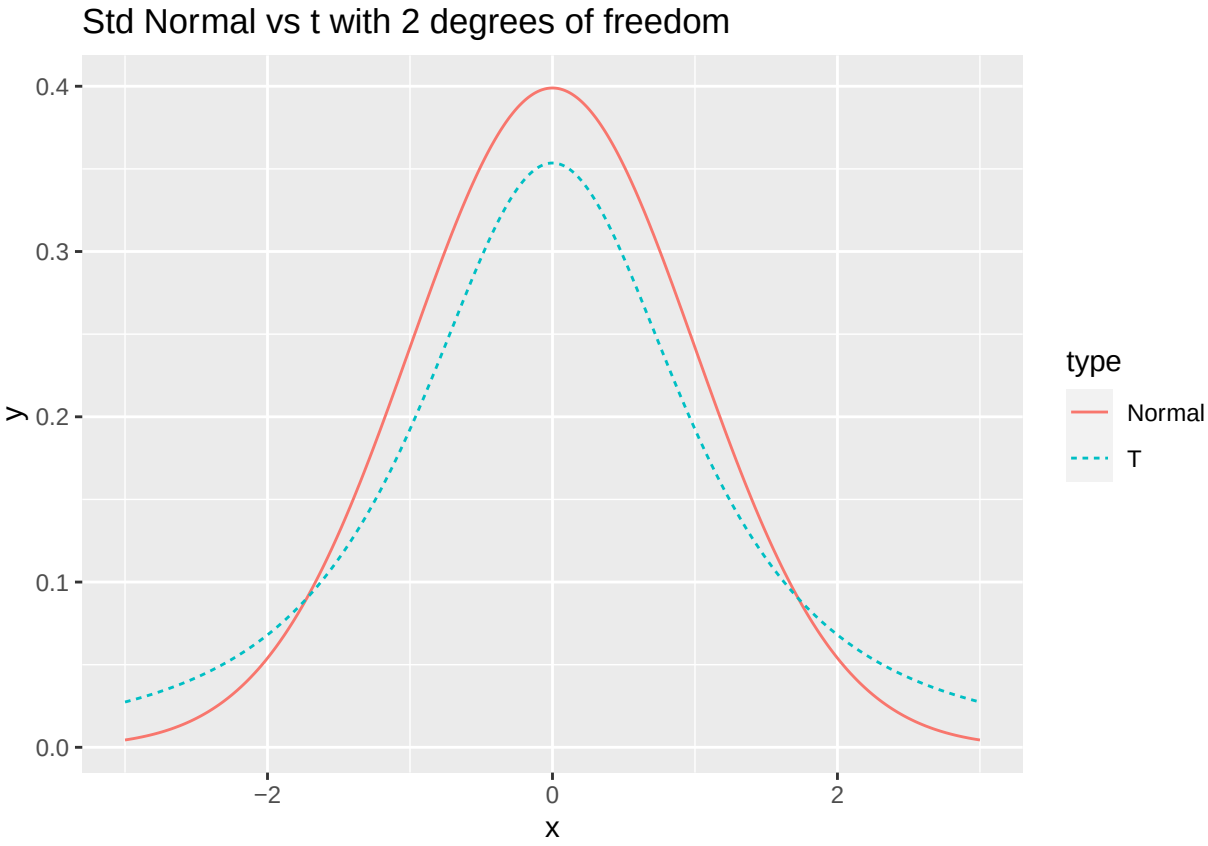
```
x <- runif(n=1, 0,10)  # one random value between 0 and 10
result <- ifelse( a<=x & x<=b, 1/(b-a), 0 )
print(paste('x=',round(x,digits=3), '  result=', round(result,digits=3)))
```

```
## [1] "x= 5.749   result= 0.167"
```

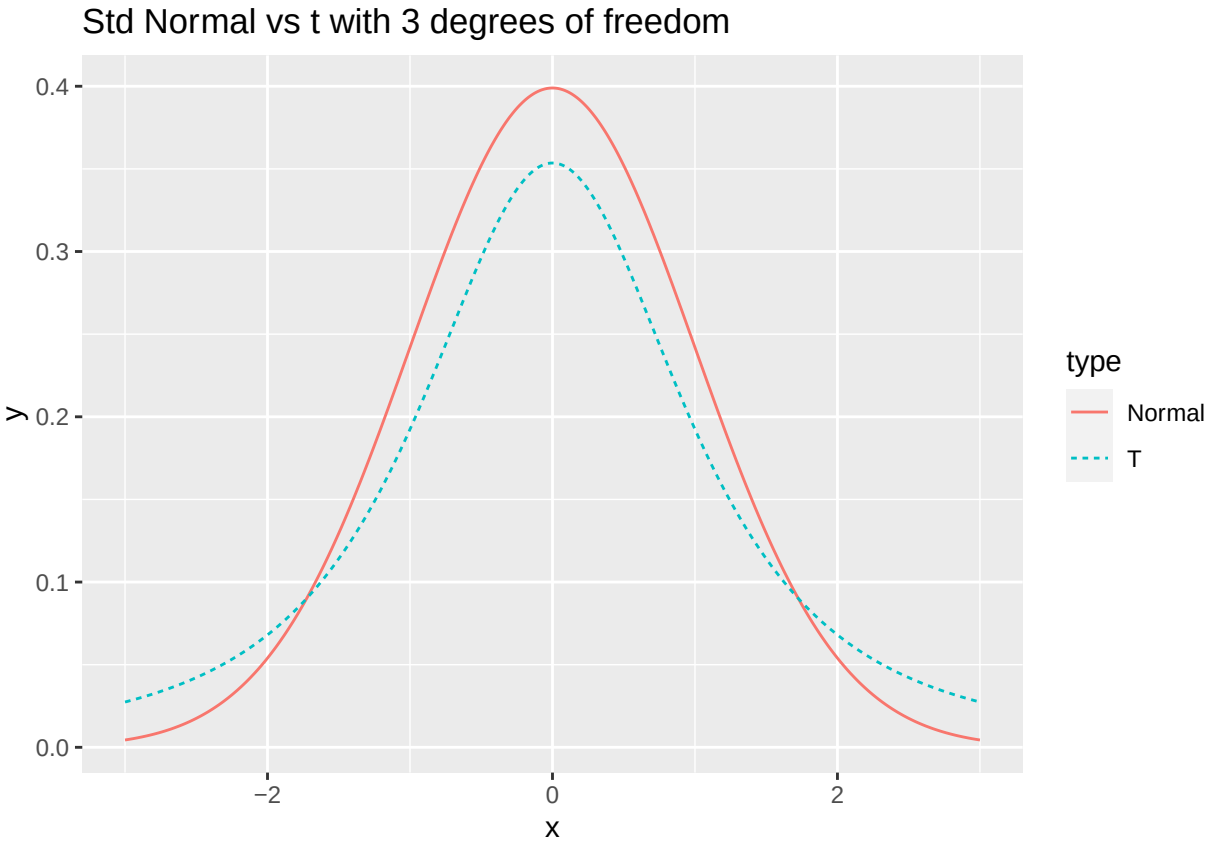
3.a.

```
library(ggplot2)
N <- 1000
df <- 2
x.grid <- seq(-3, 3, length=N)
data <- data.frame(
  x = c(x.grid, x.grid),
  y = c(dnorm(x.grid), dt(x.grid, df)),
  type = c( rep('Normal',N), rep('T',N) ) )



for ( df in 2:30 ){
  # print out current value of df
  print( paste("In loop and df is now:", df) )


  # make a nice graph
  myplot <- ggplot(data, aes(x=x, y=y, color=type, linetype=type)) +
  geom_line() +
  labs(title = paste('Std Normal vs t with', df, 'degrees of freedom'))

  # actually print the nice graph we made
  print(myplot)
  }
```

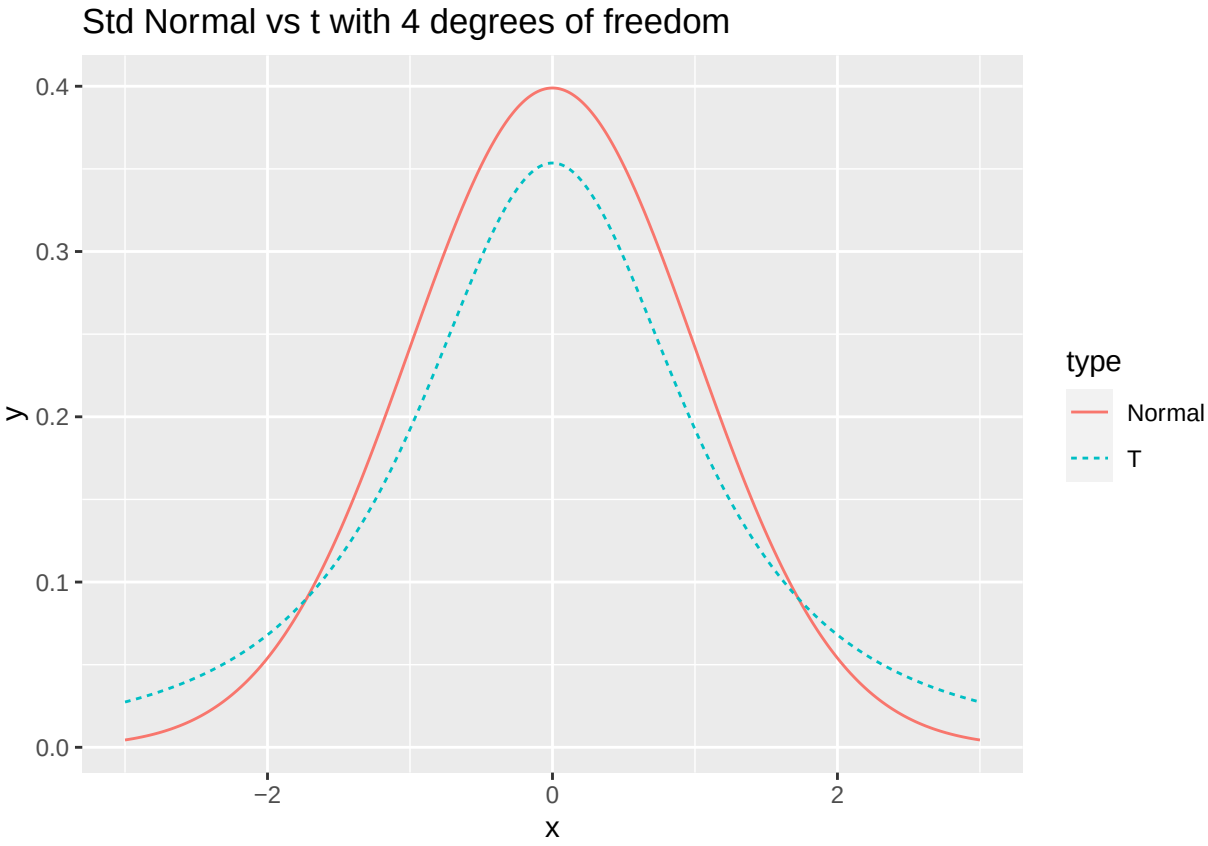
```
## [1] "In loop and df is now: 2"
```



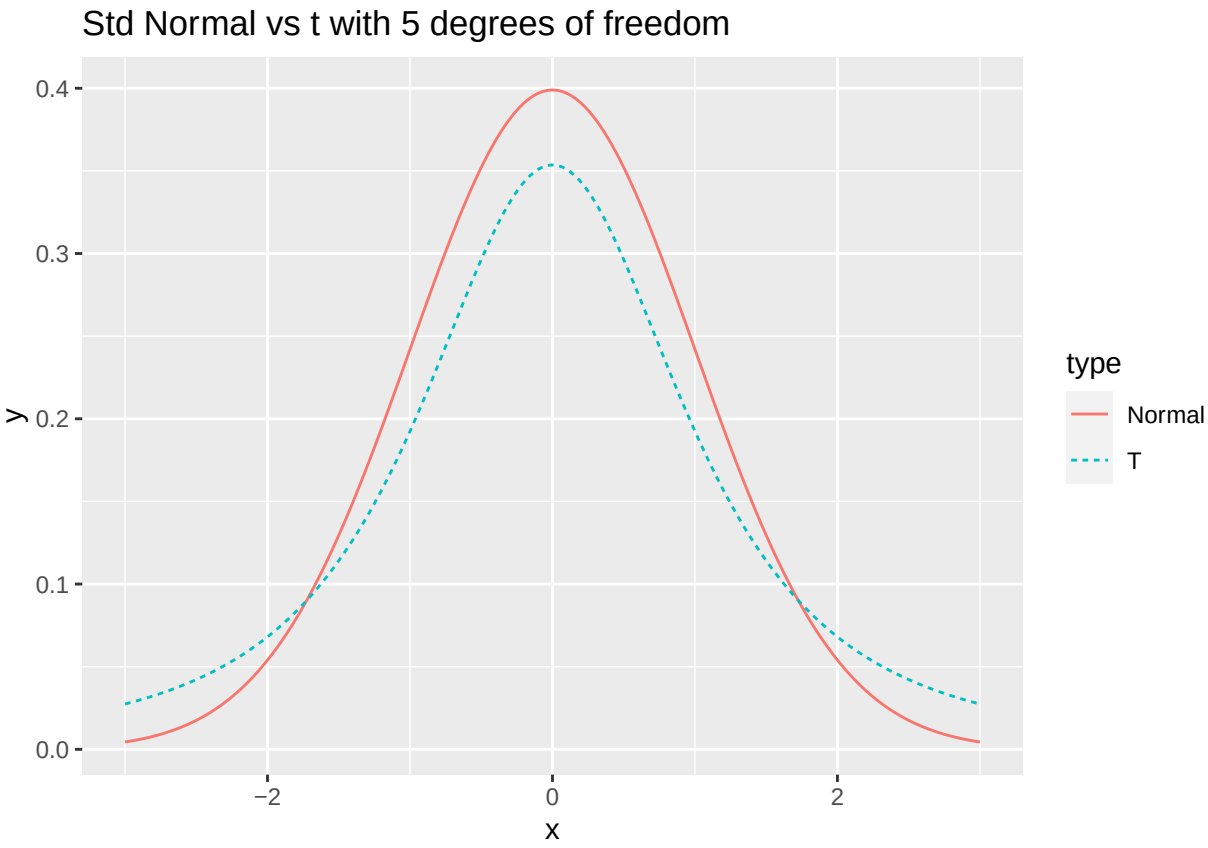

```
## [1] "In loop and df is now: 3"
```



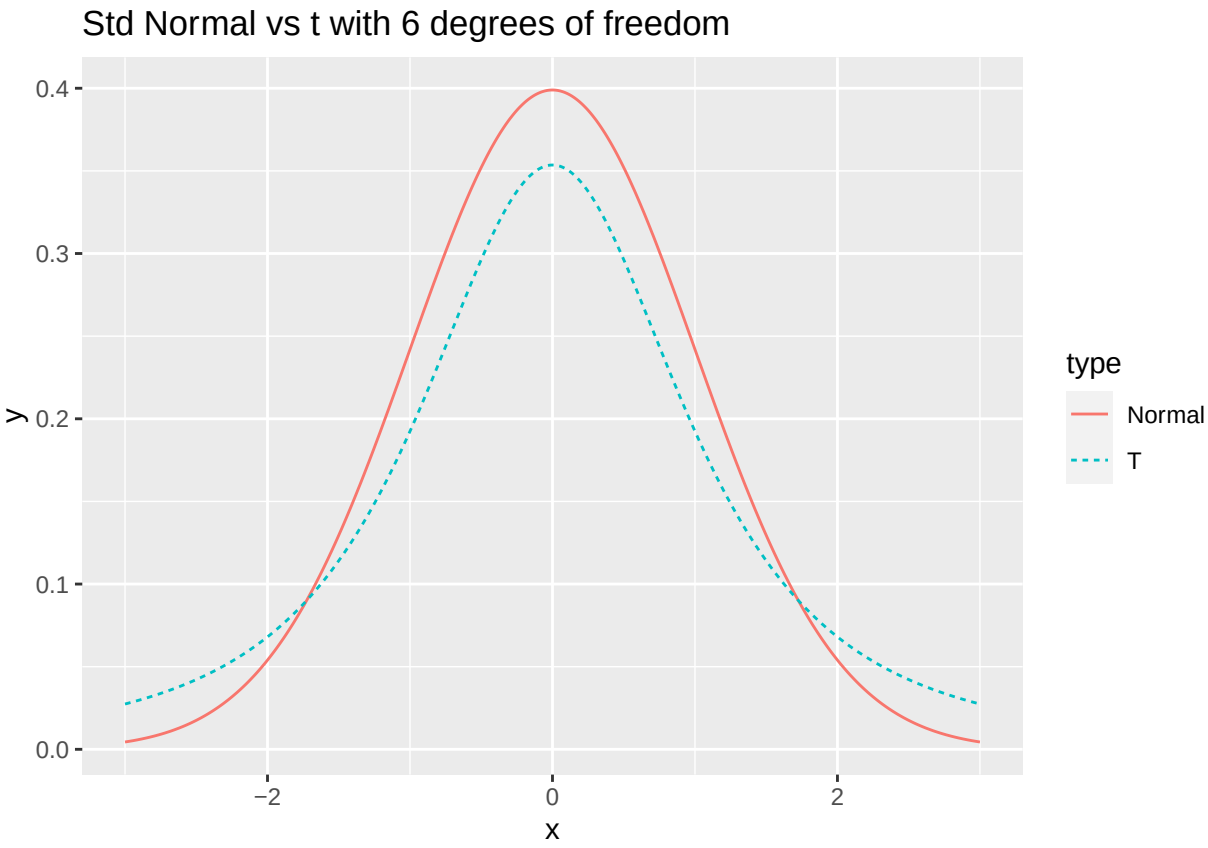

```
## [1] "In loop and df is now: 4"
```



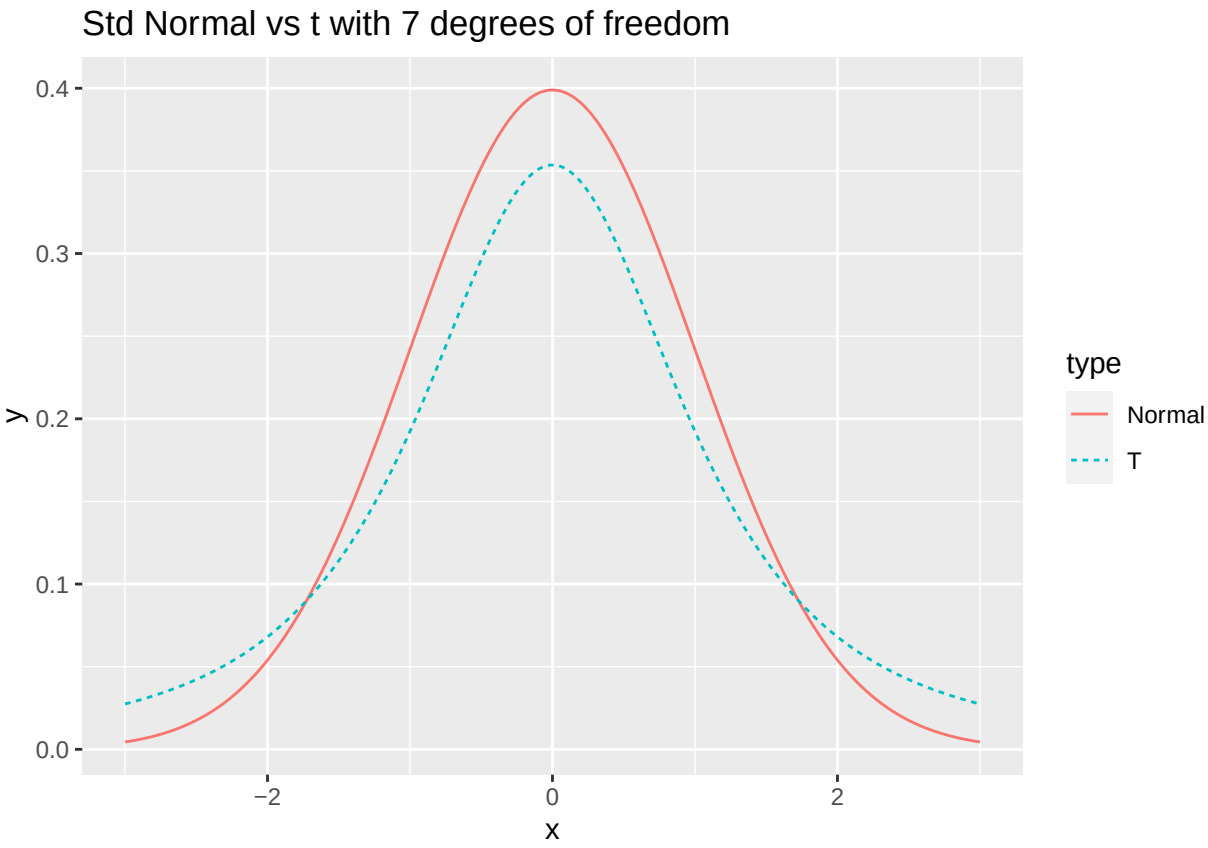

```
## [1] "In loop and df is now: 5"
```



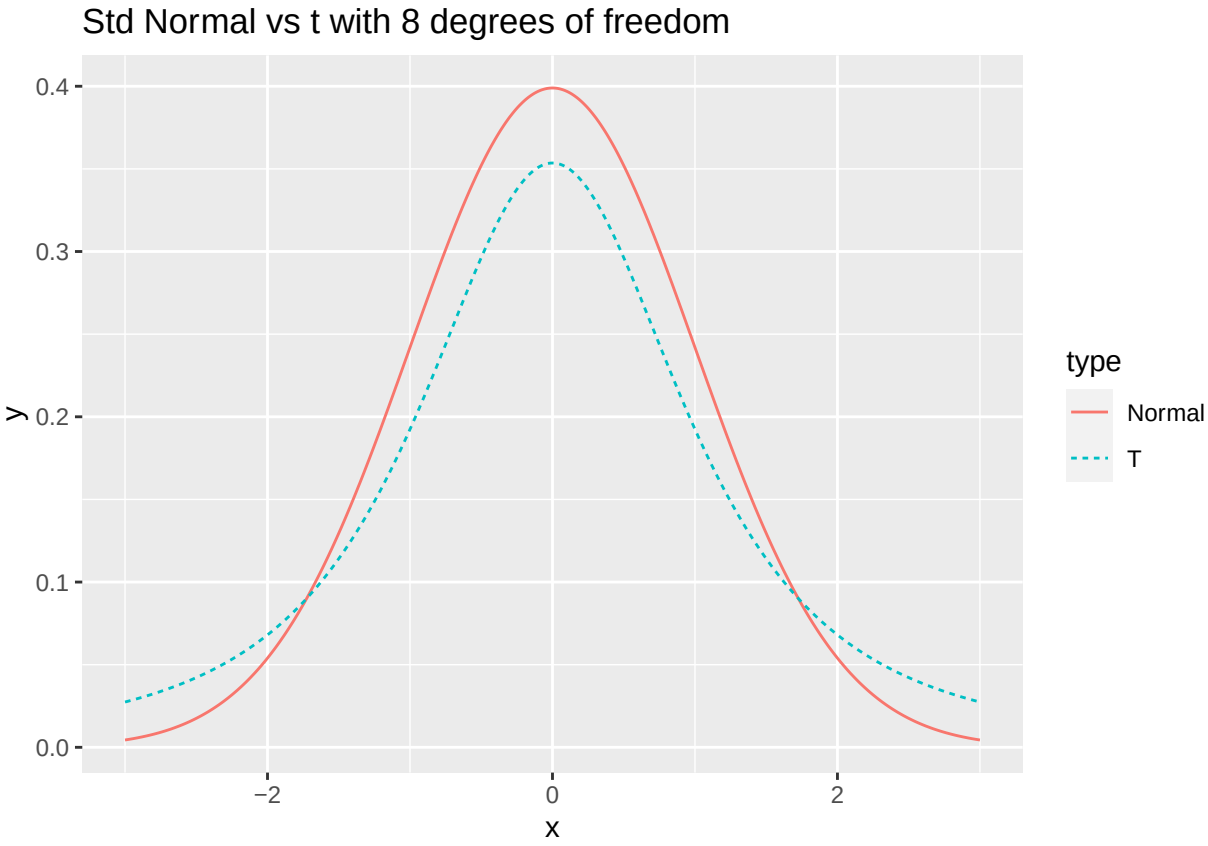

```
## [1] "In loop and df is now: 6"
```



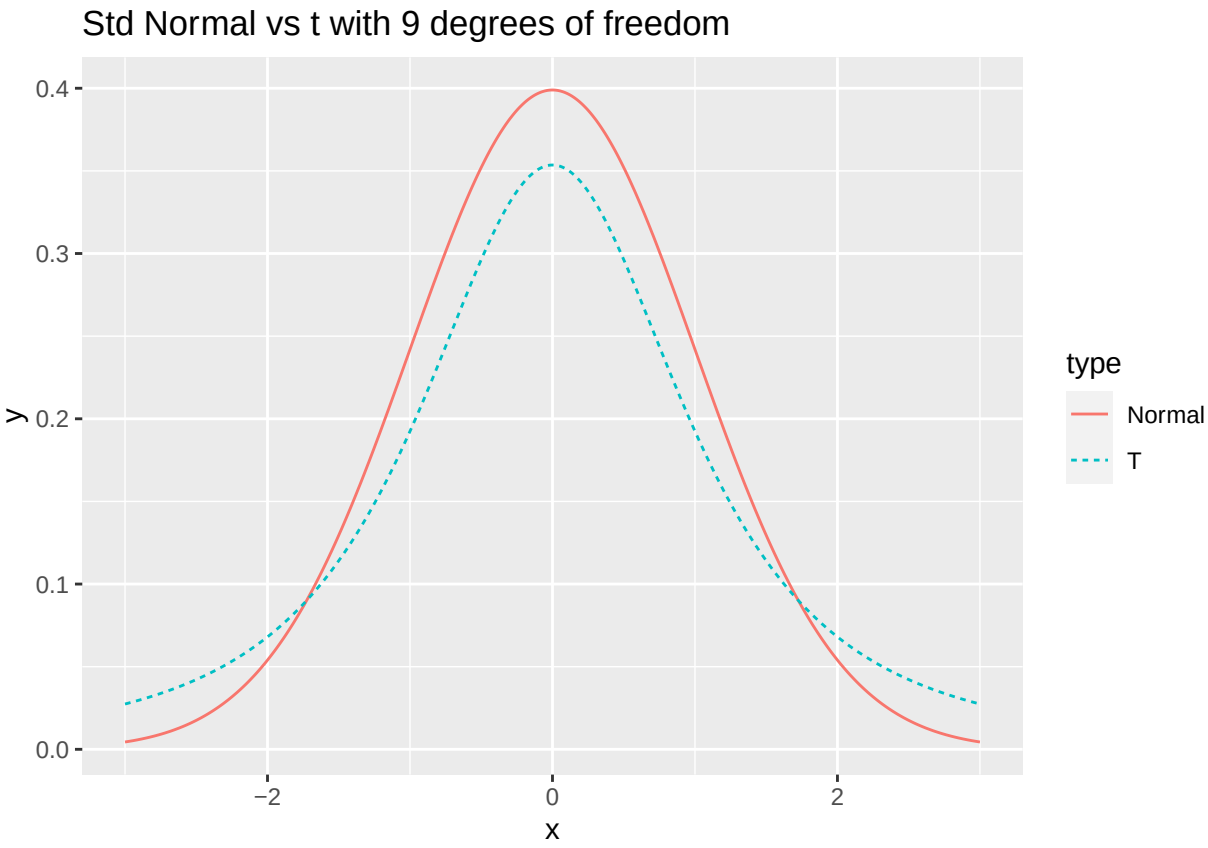

```
## [1] "In loop and df is now: 7"
```



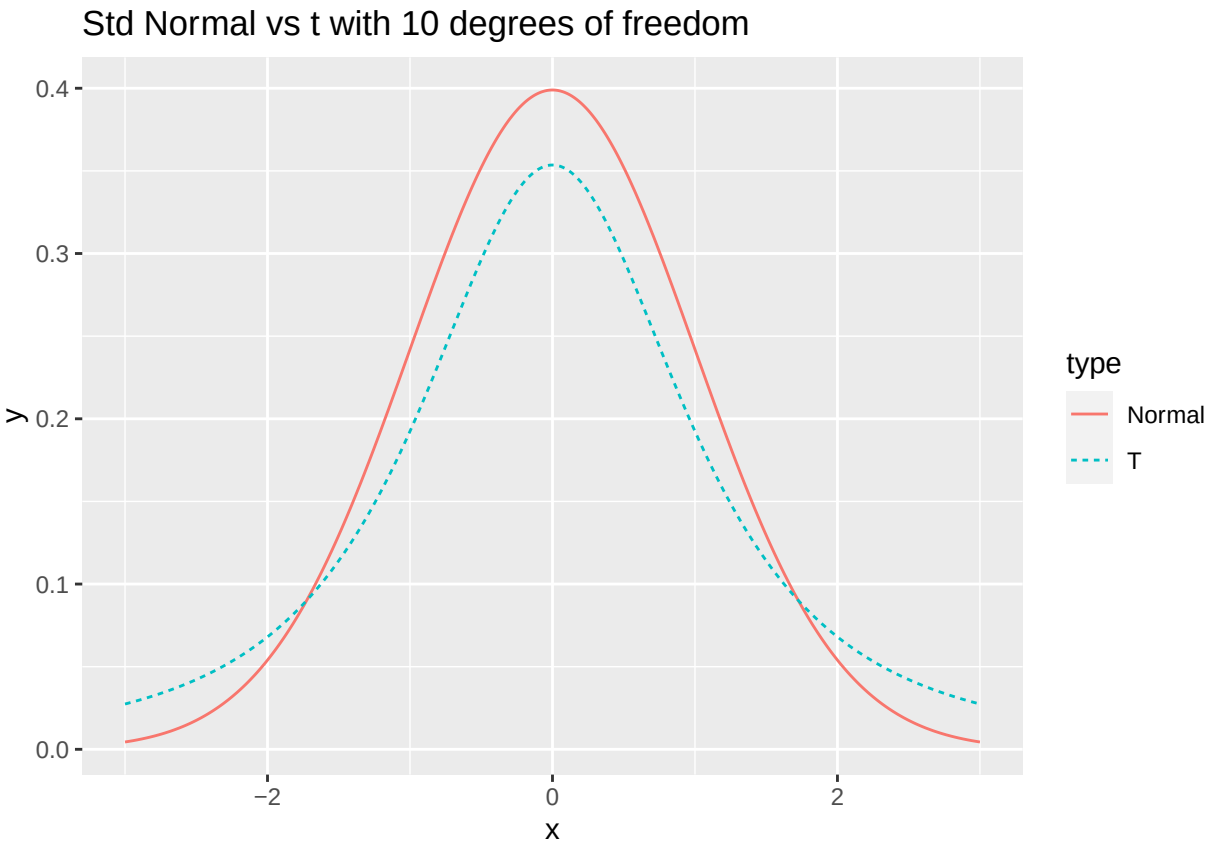

```
## [1] "In loop and df is now: 8"
```



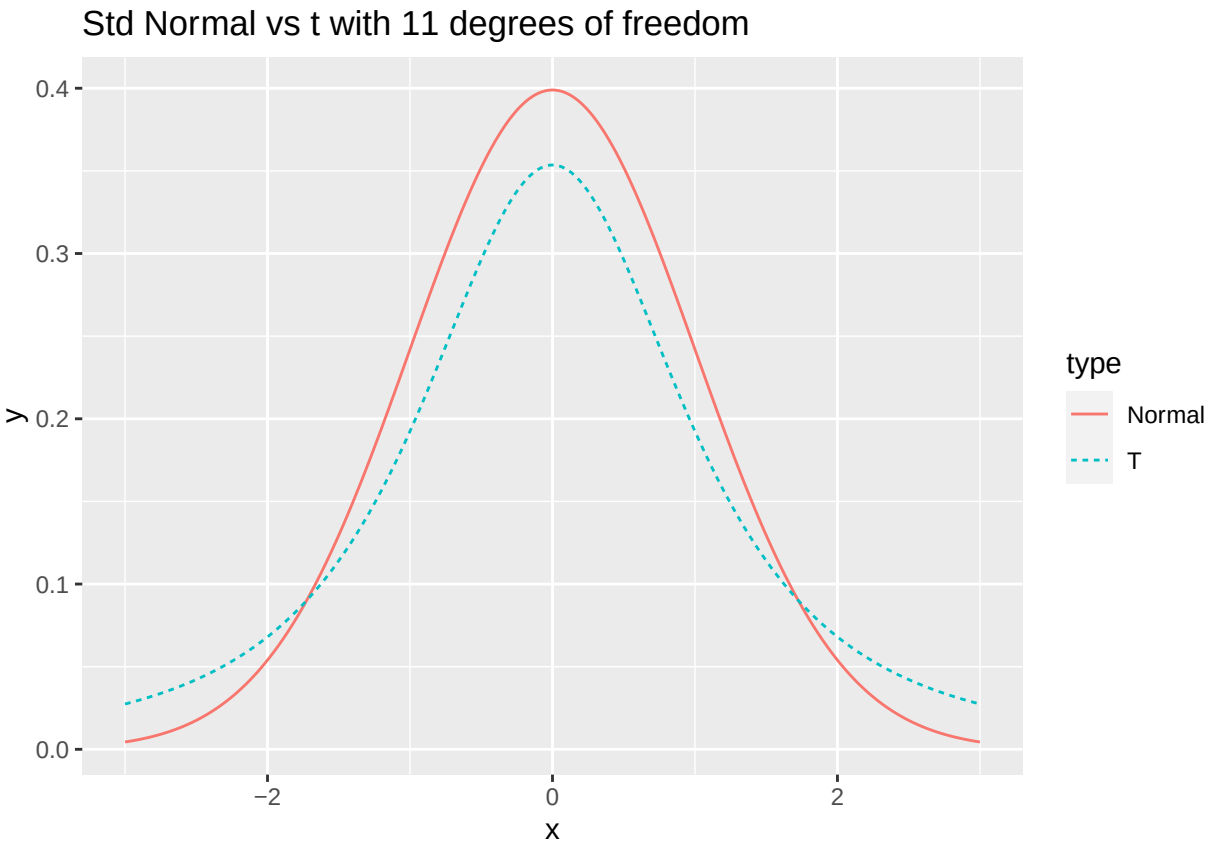

```
## [1] "In loop and df is now: 9"
```



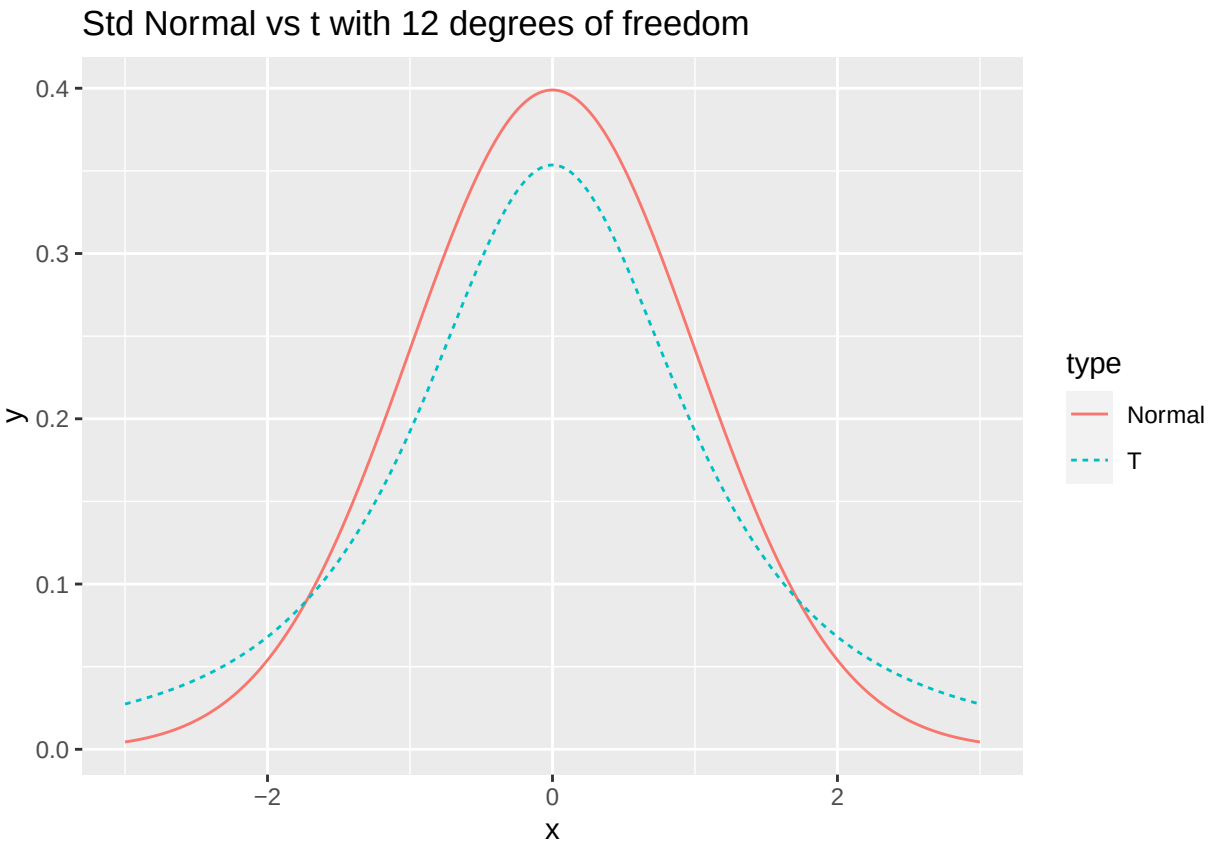

```
## [1] "In loop and df is now: 10"
```



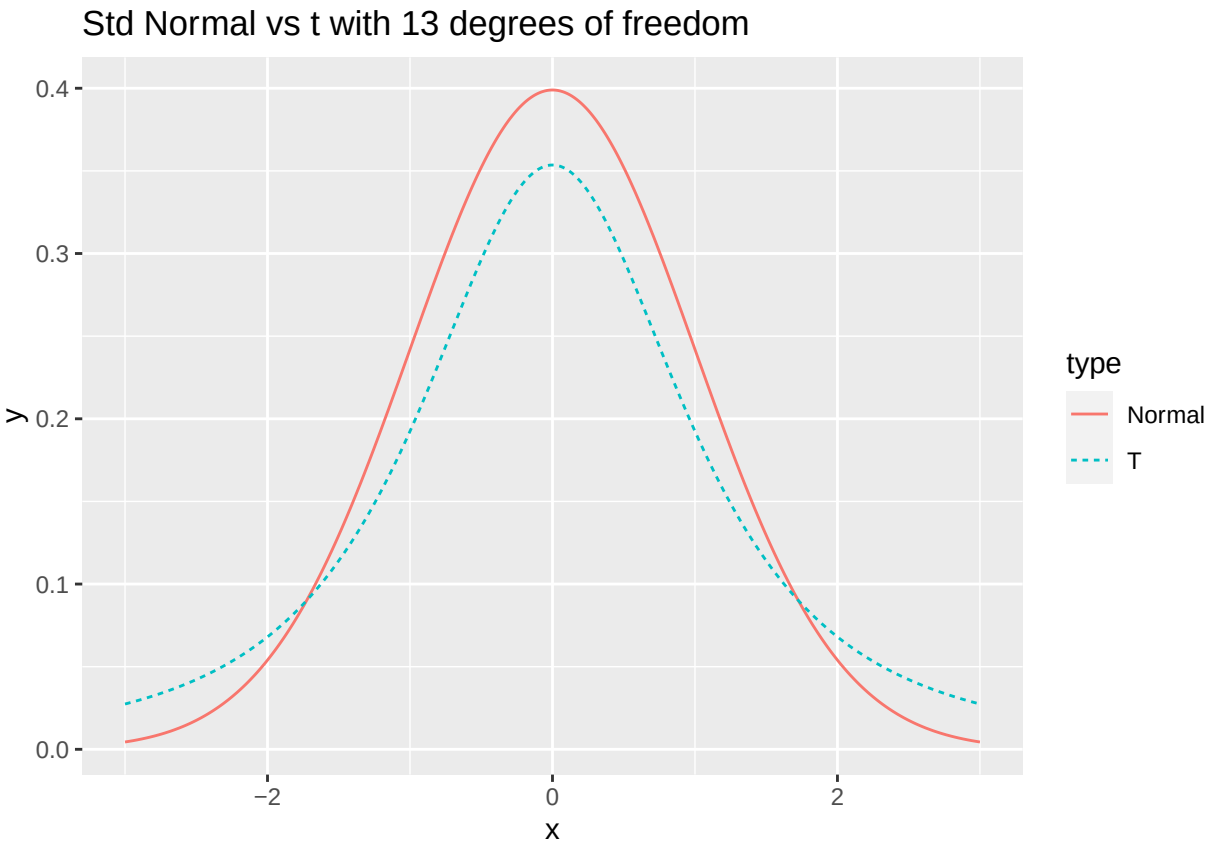

```
## [1] "In loop and df is now: 11"
```



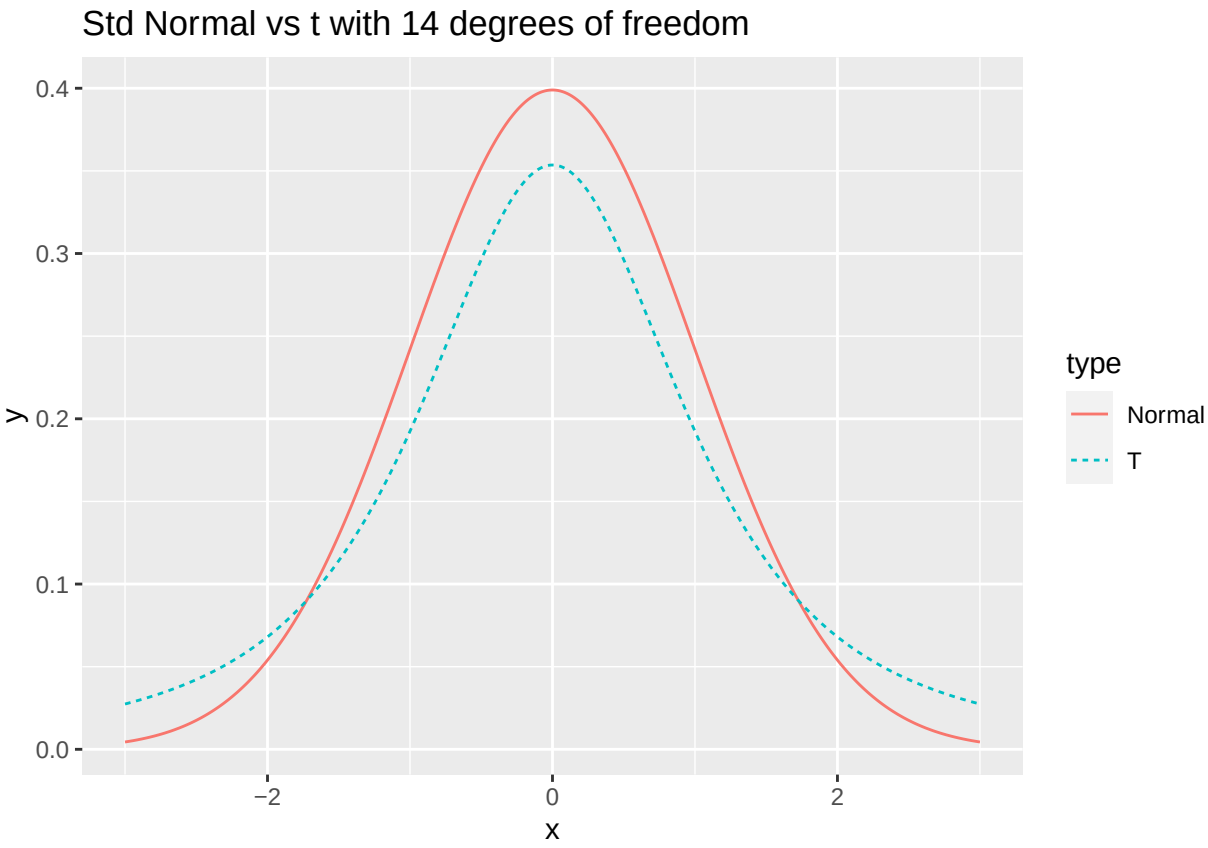

```
## [1] "In loop and df is now: 12"
```



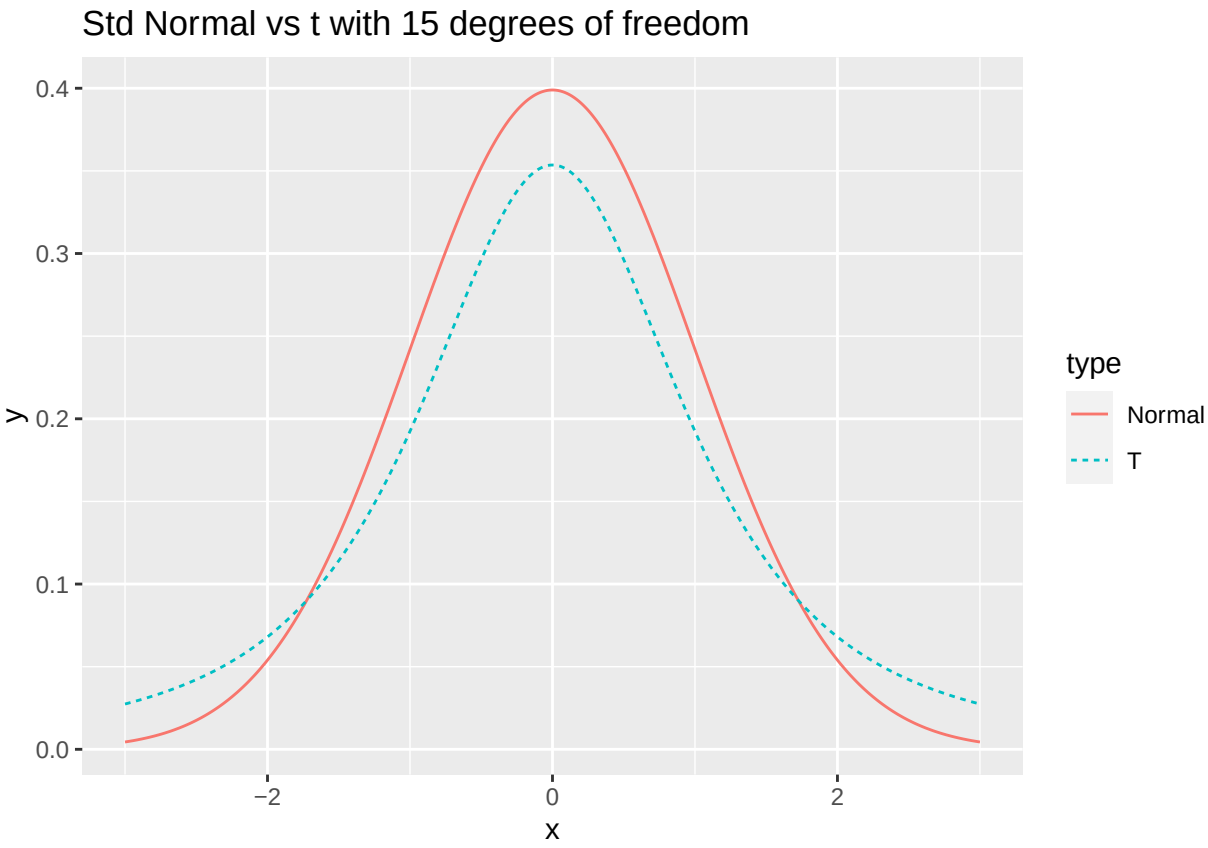

```
## [1] "In loop and df is now: 13"
```



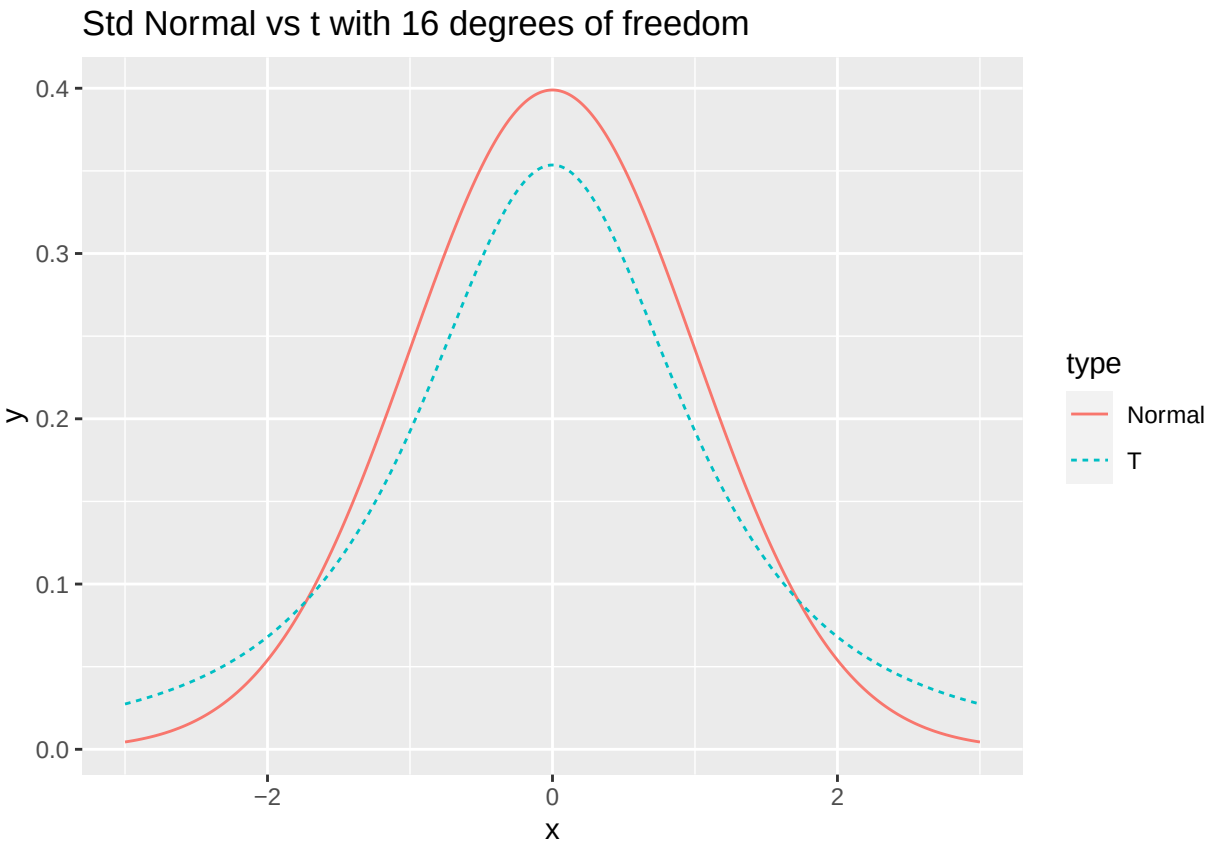

```
## [1] "In loop and df is now: 14"
```



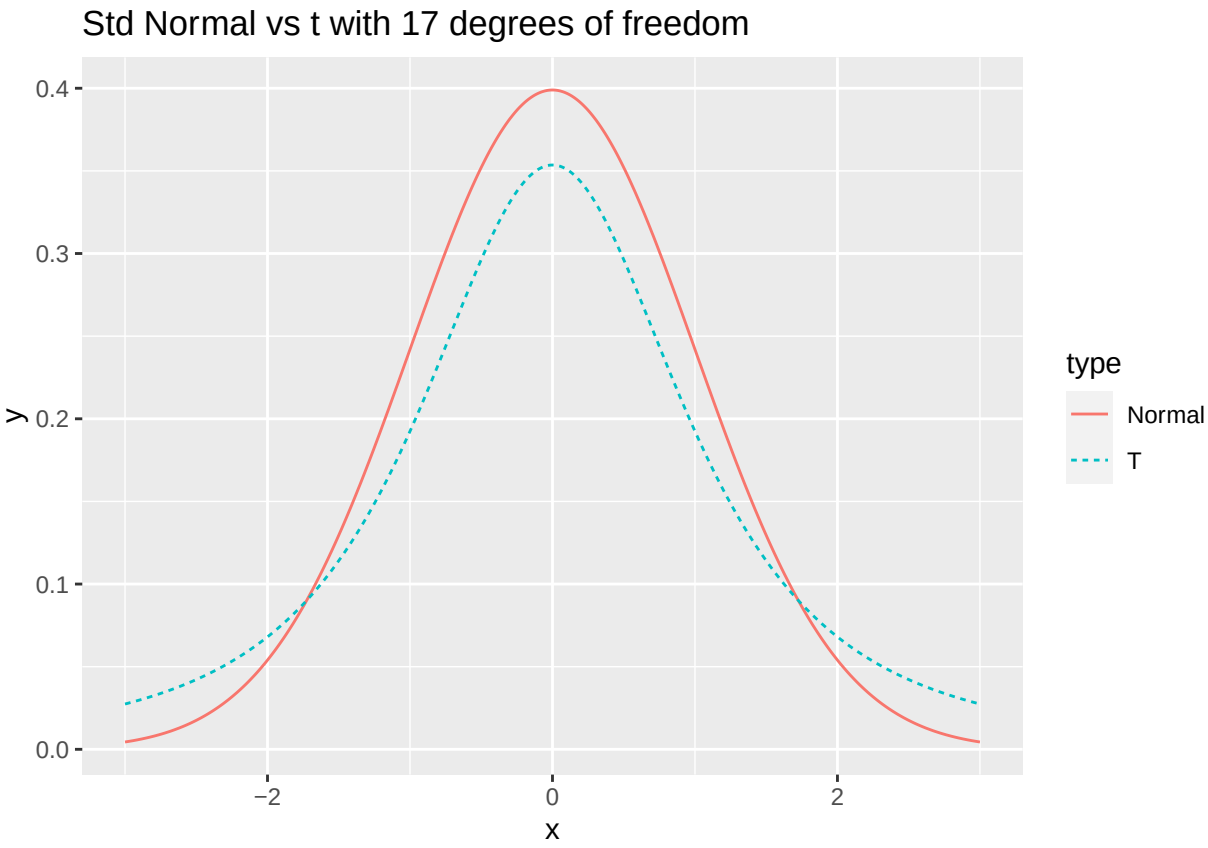

```
## [1] "In loop and df is now: 15"
```



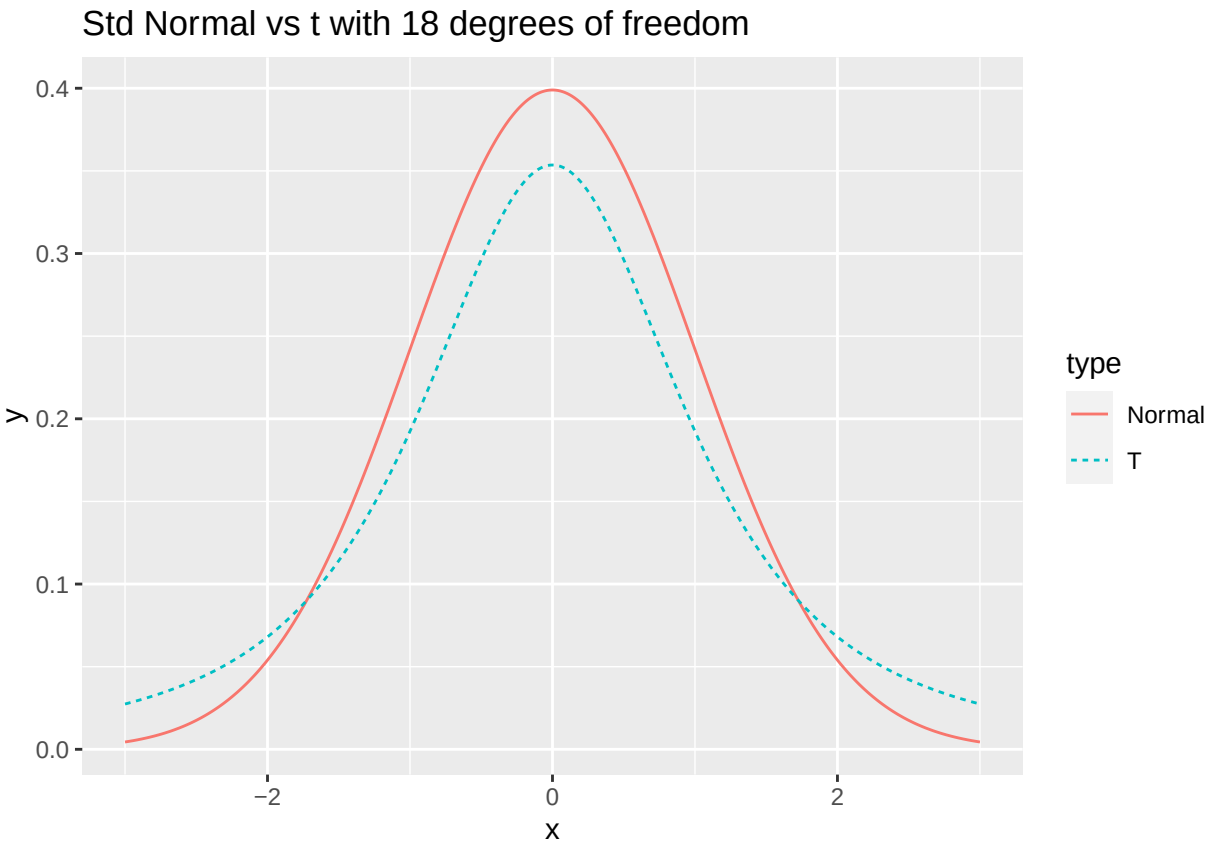

```
## [1] "In loop and df is now: 16"
```



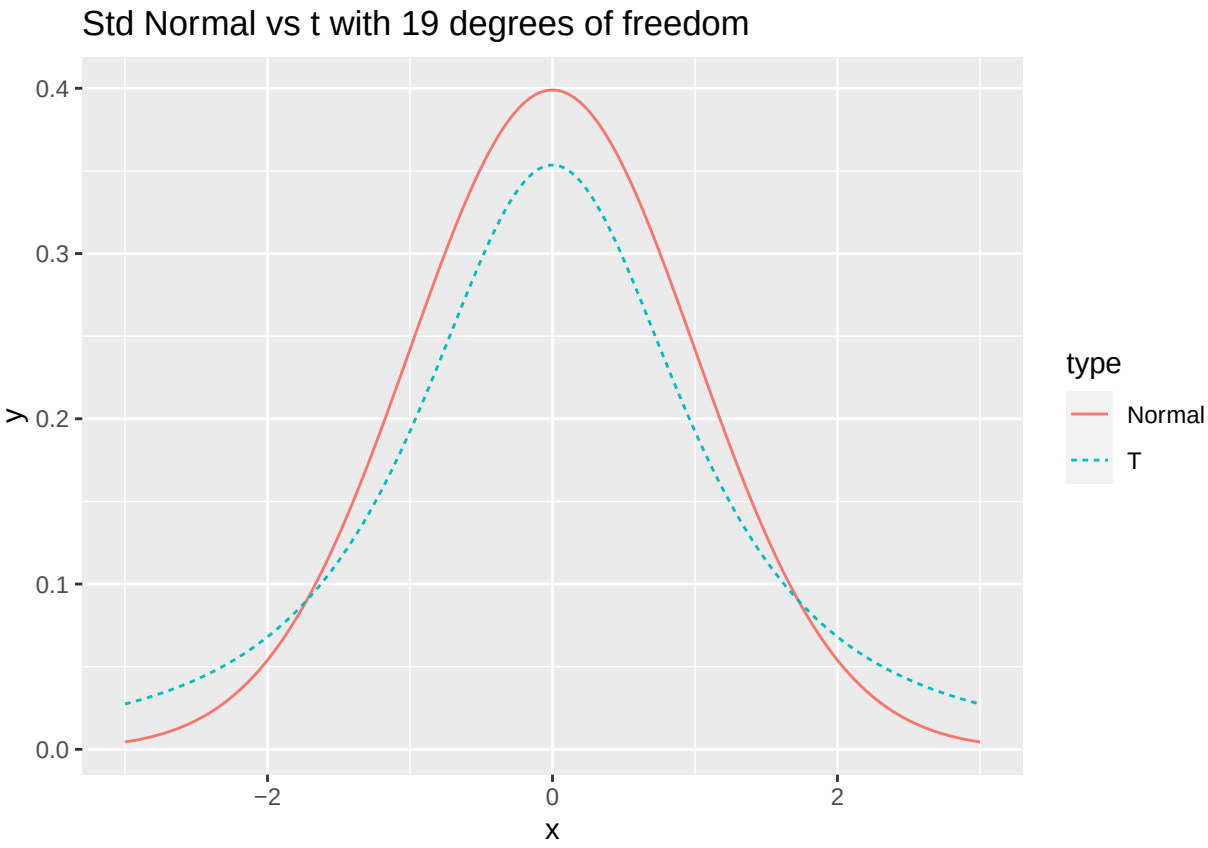

```
## [1] "In loop and df is now: 17"
```



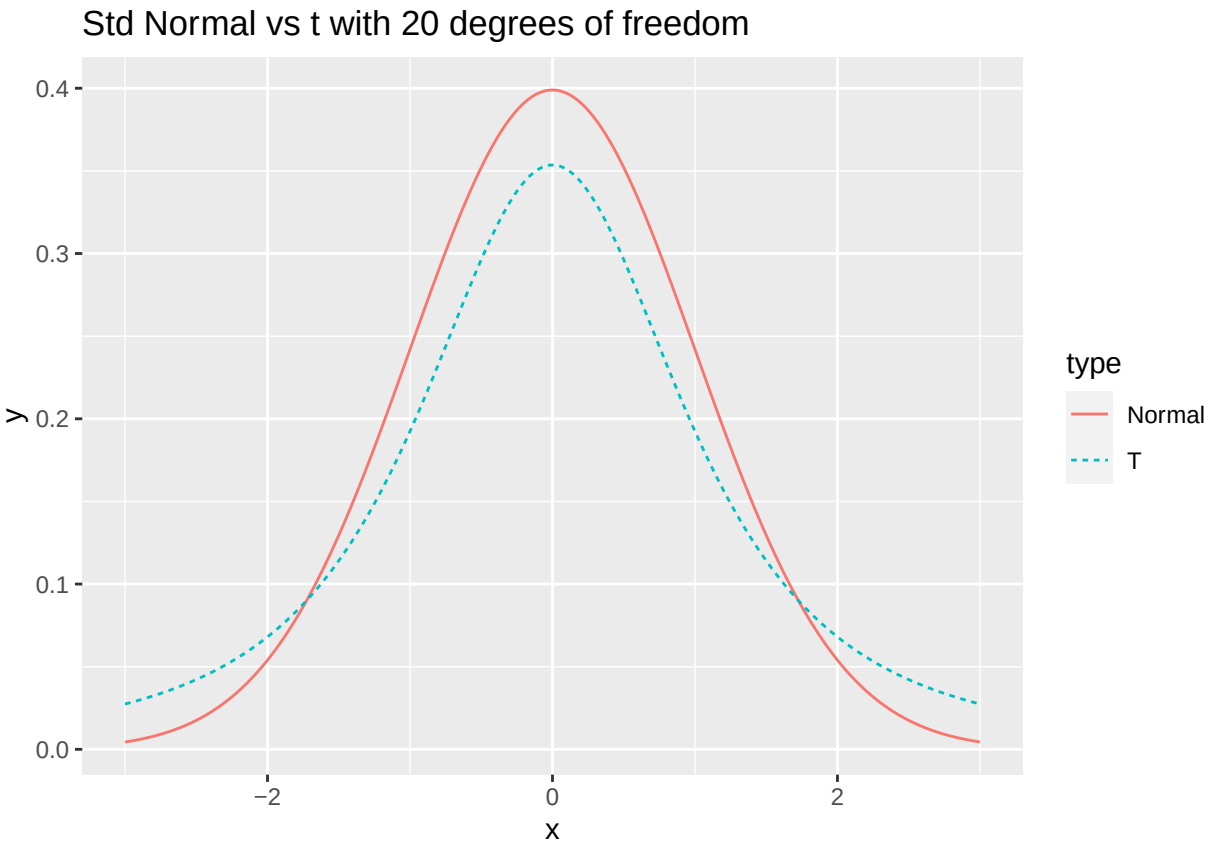

```
## [1] "In loop and df is now: 18"
```



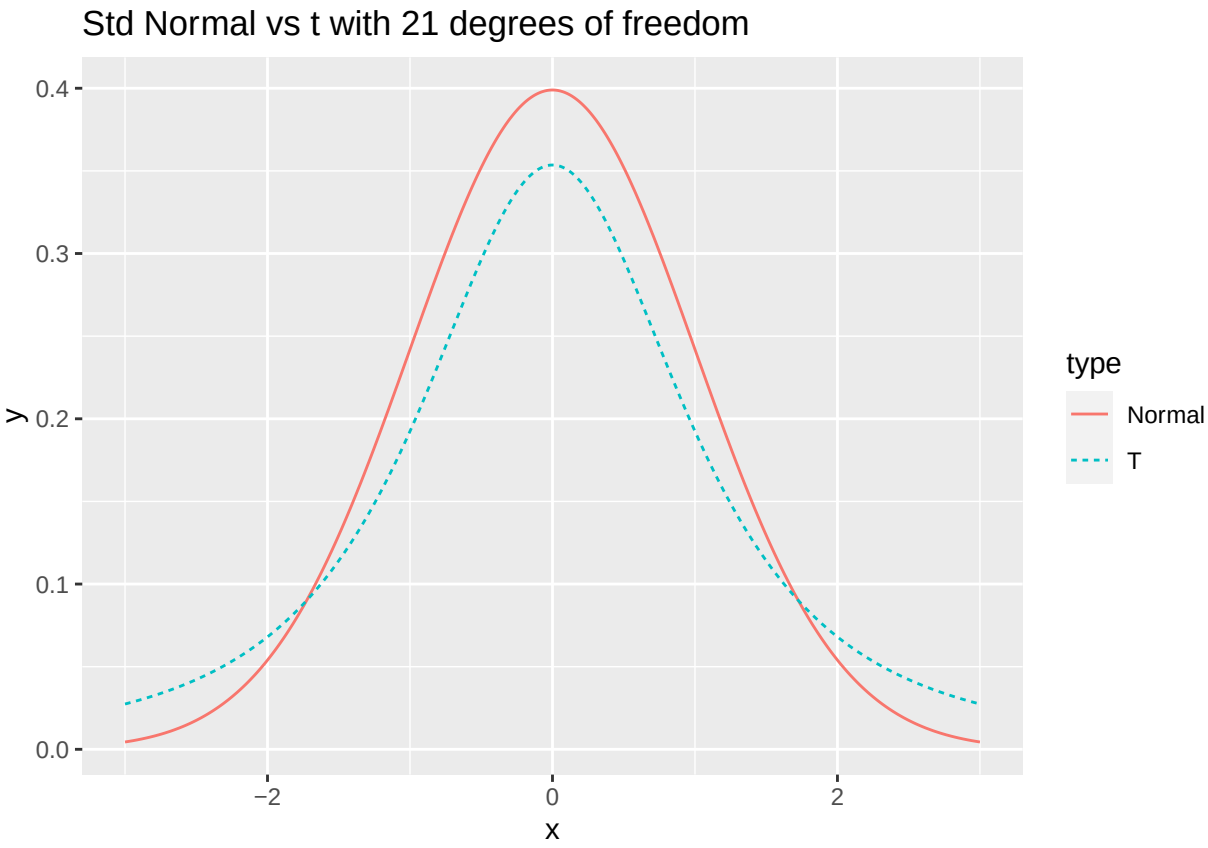

```
## [1] "In loop and df is now: 19"
```



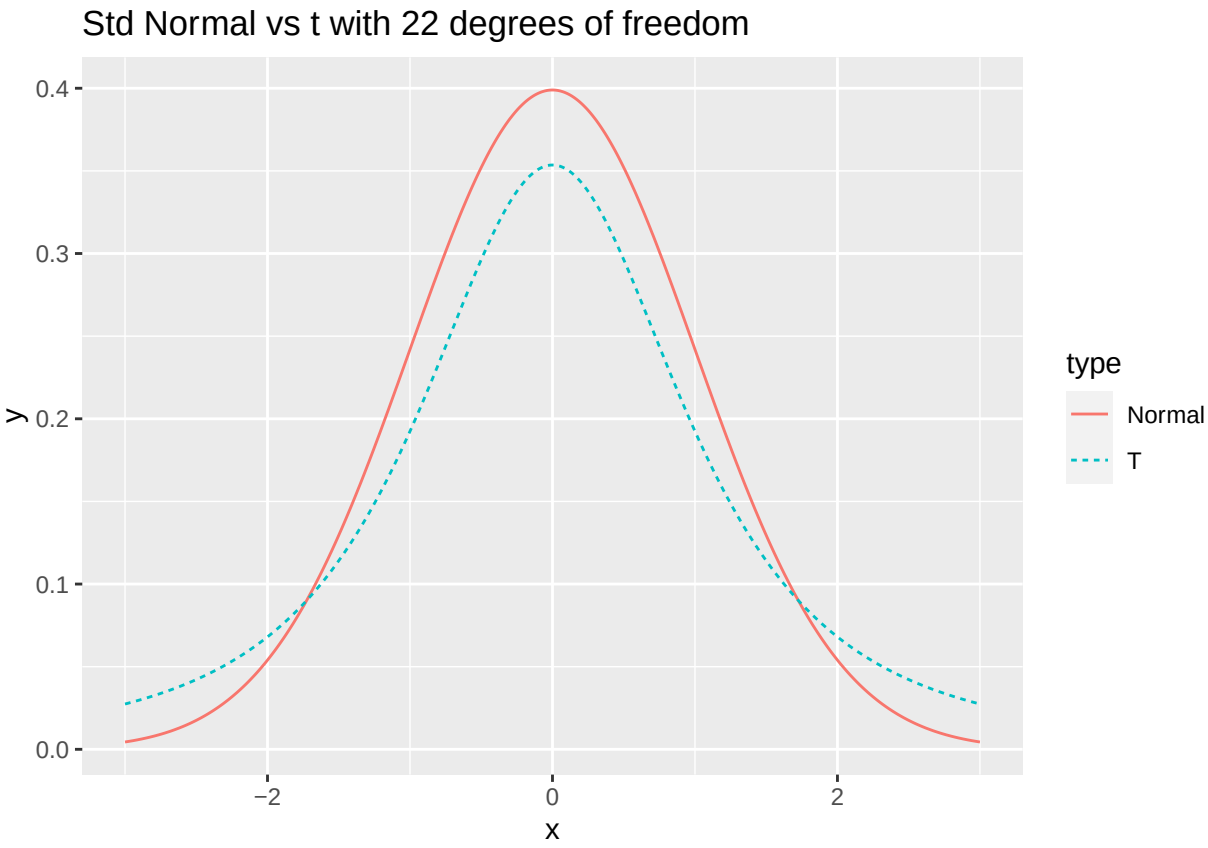

```
## [1] "In loop and df is now: 20"
```



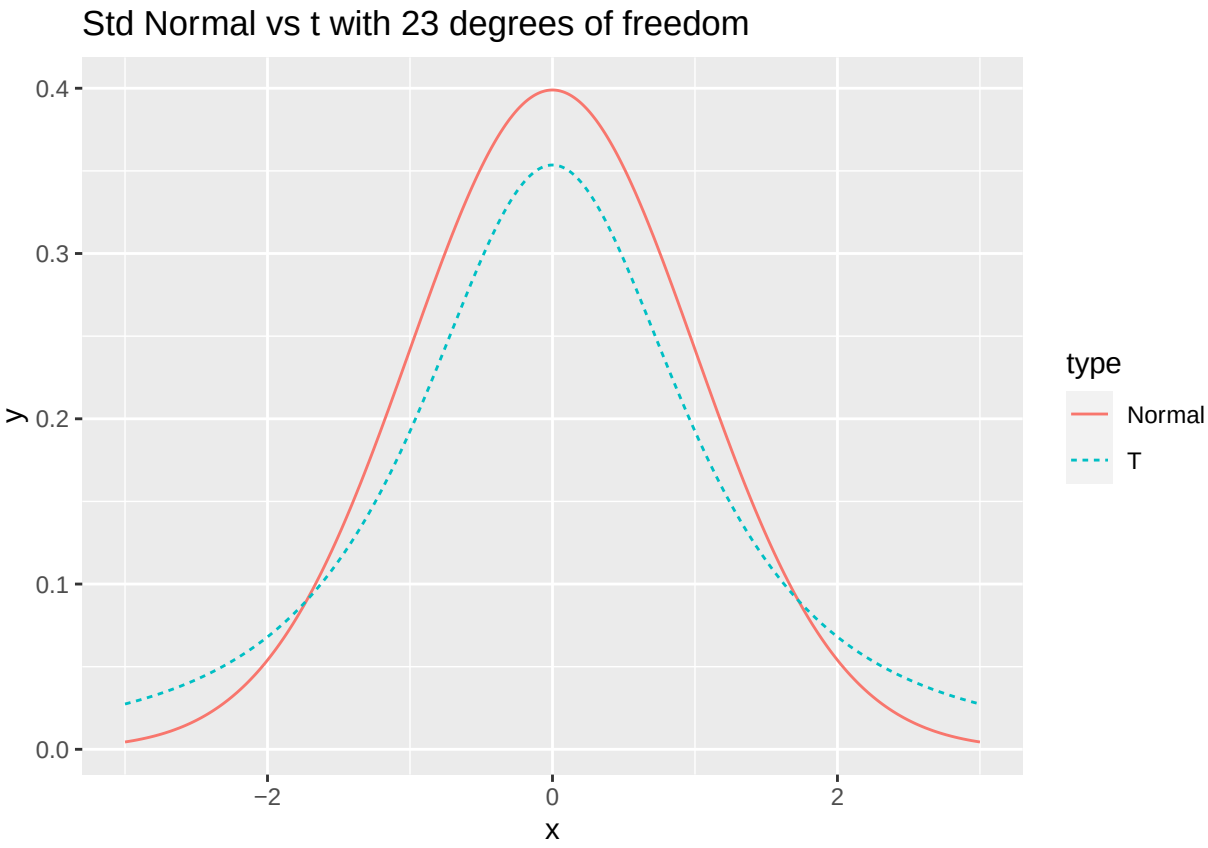

```
## [1] "In loop and df is now: 21"
```



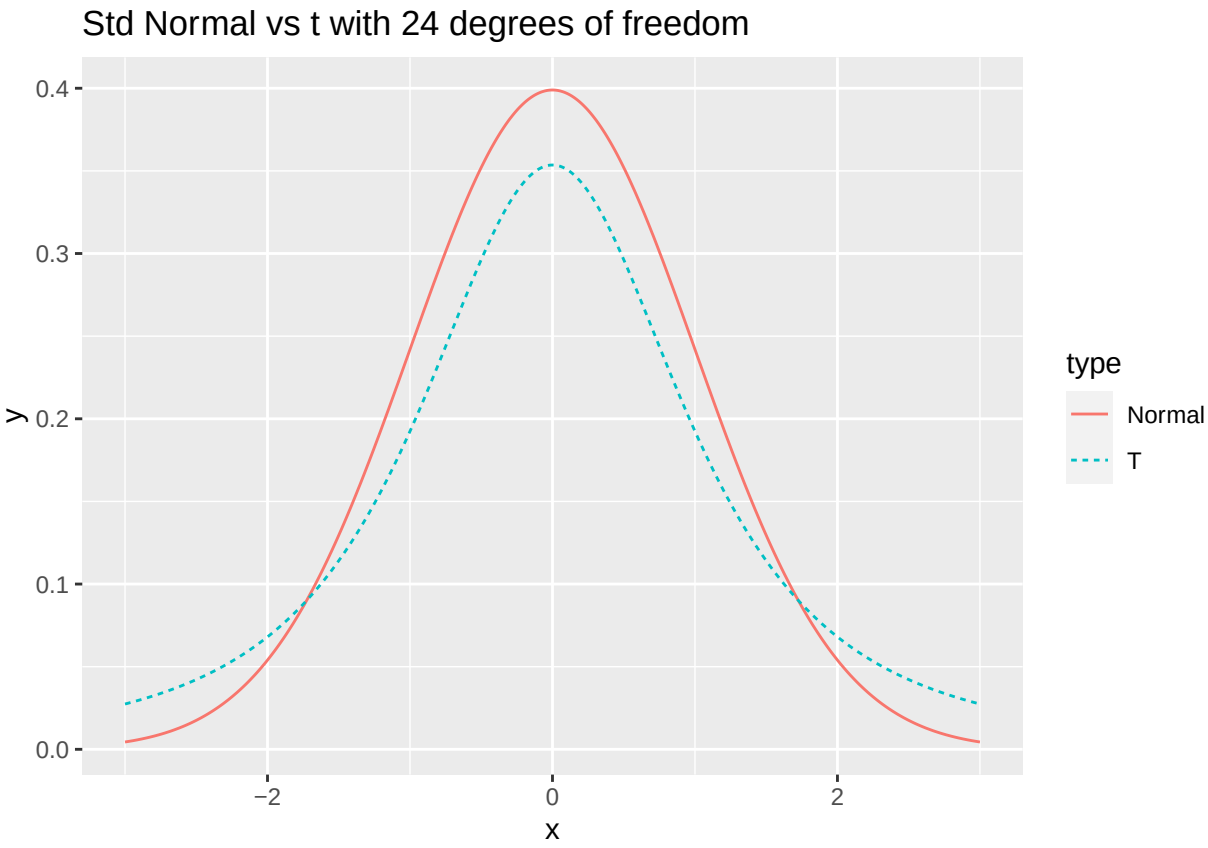

```
## [1] "In loop and df is now: 22"
```



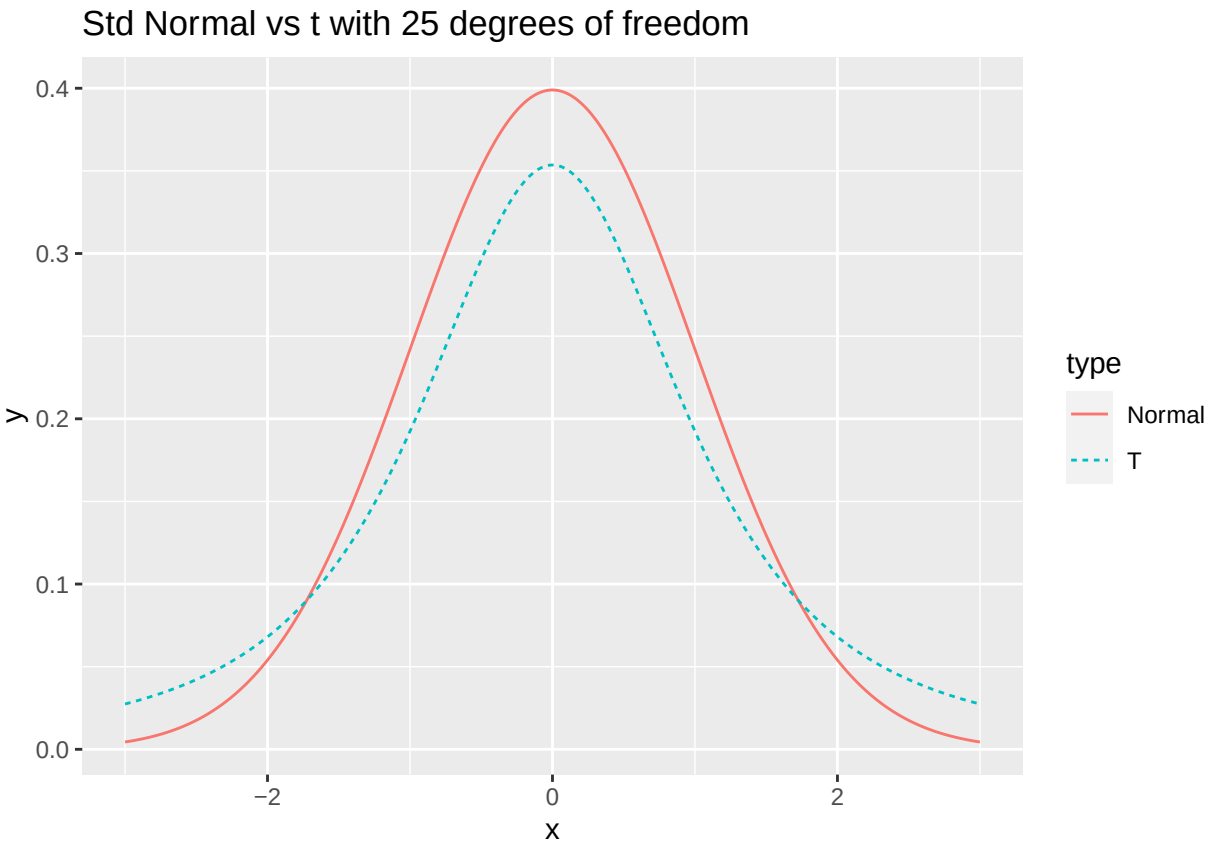

```
## [1] "In loop and df is now: 23"
```



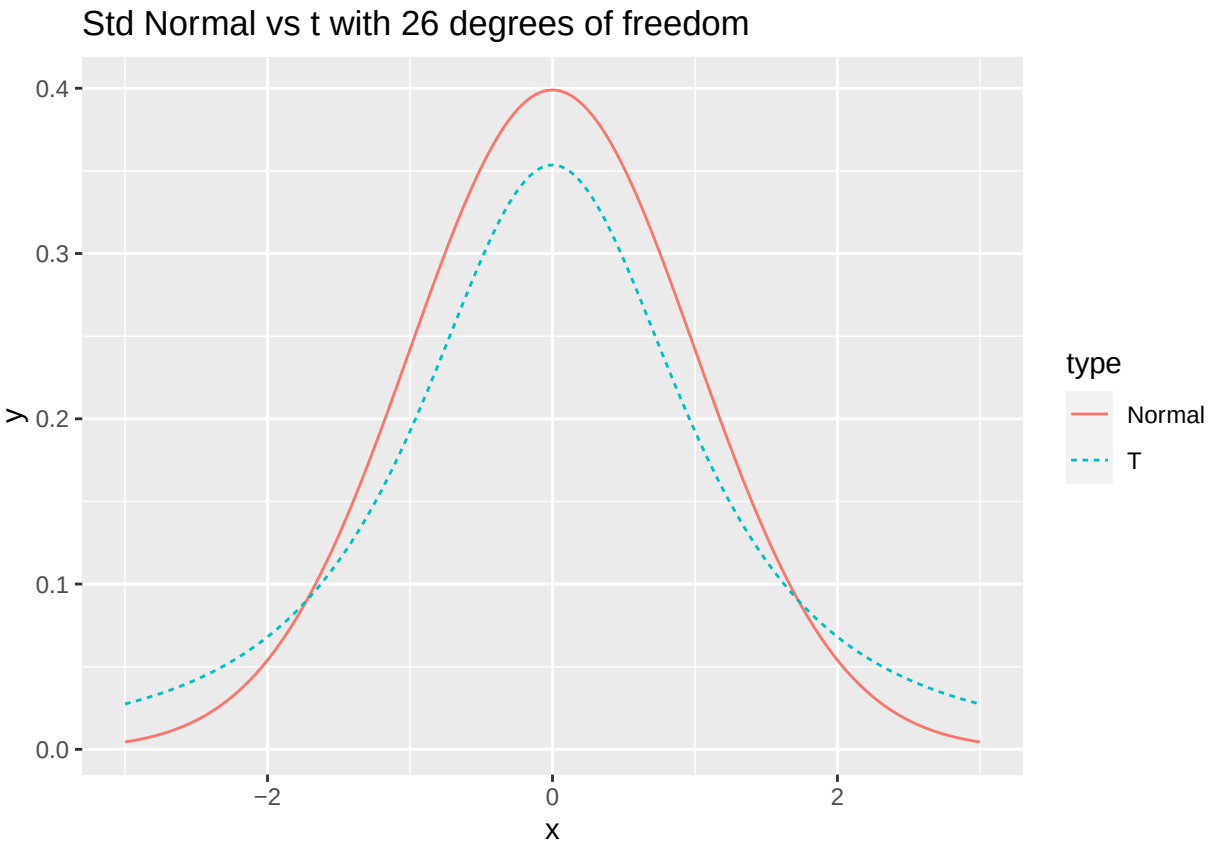

```
## [1] "In loop and df is now: 24"
```



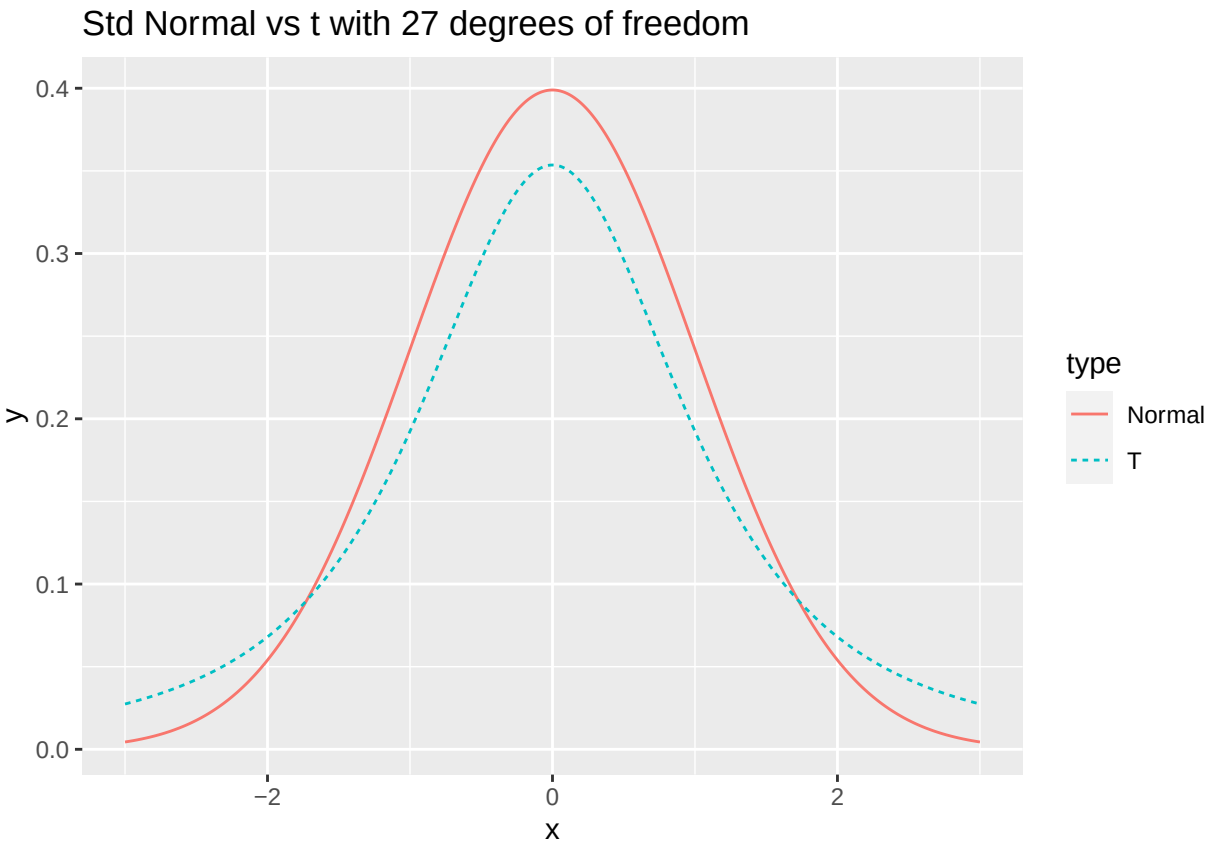

```
## [1] "In loop and df is now: 25"
```



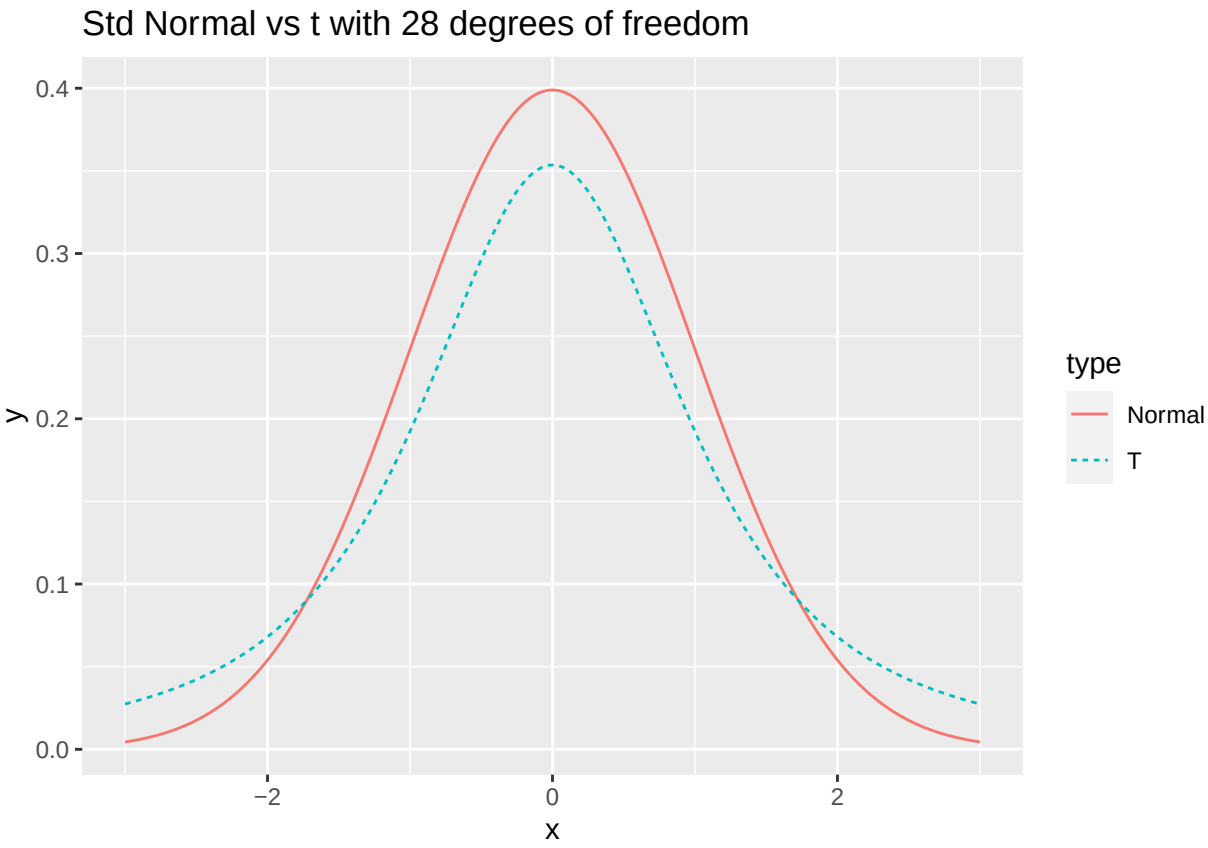

```
## [1] "In loop and df is now: 26"
```



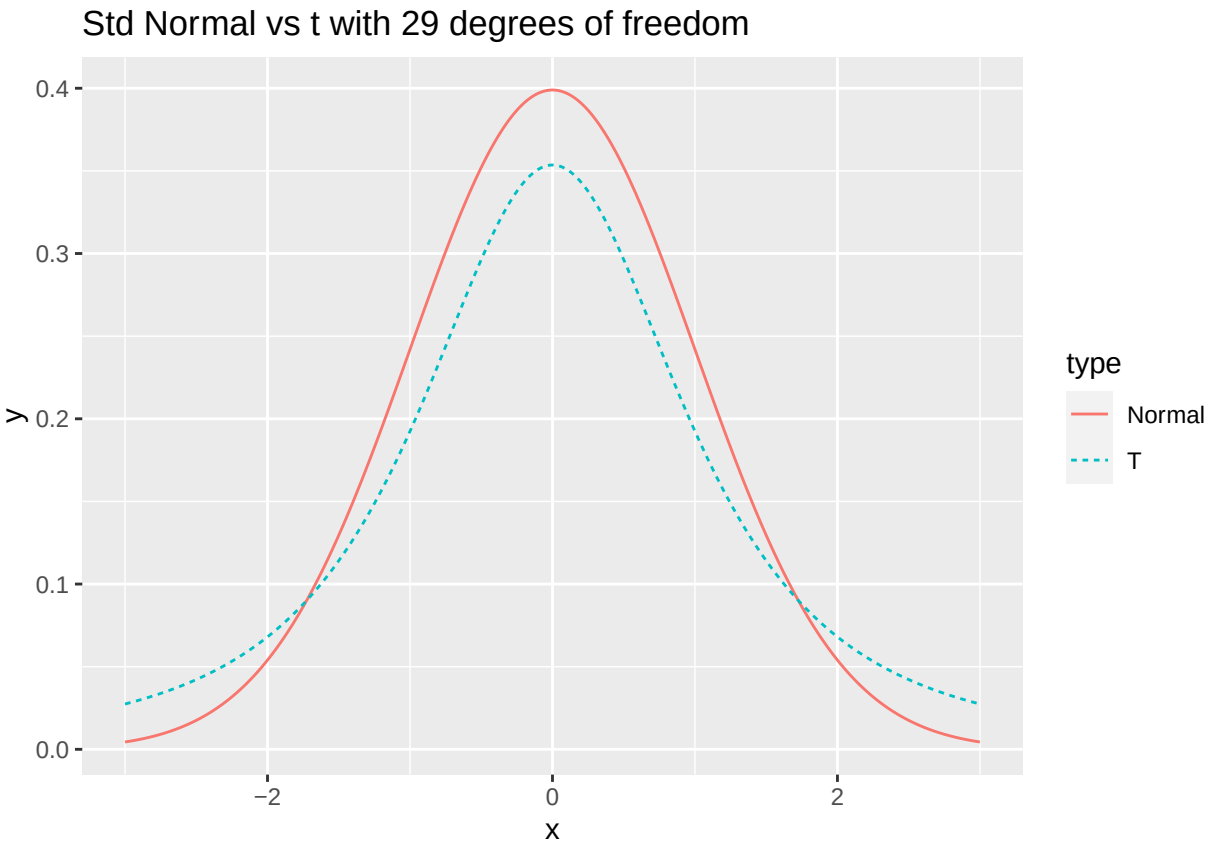

```
## [1] "In loop and df is now: 27"
```



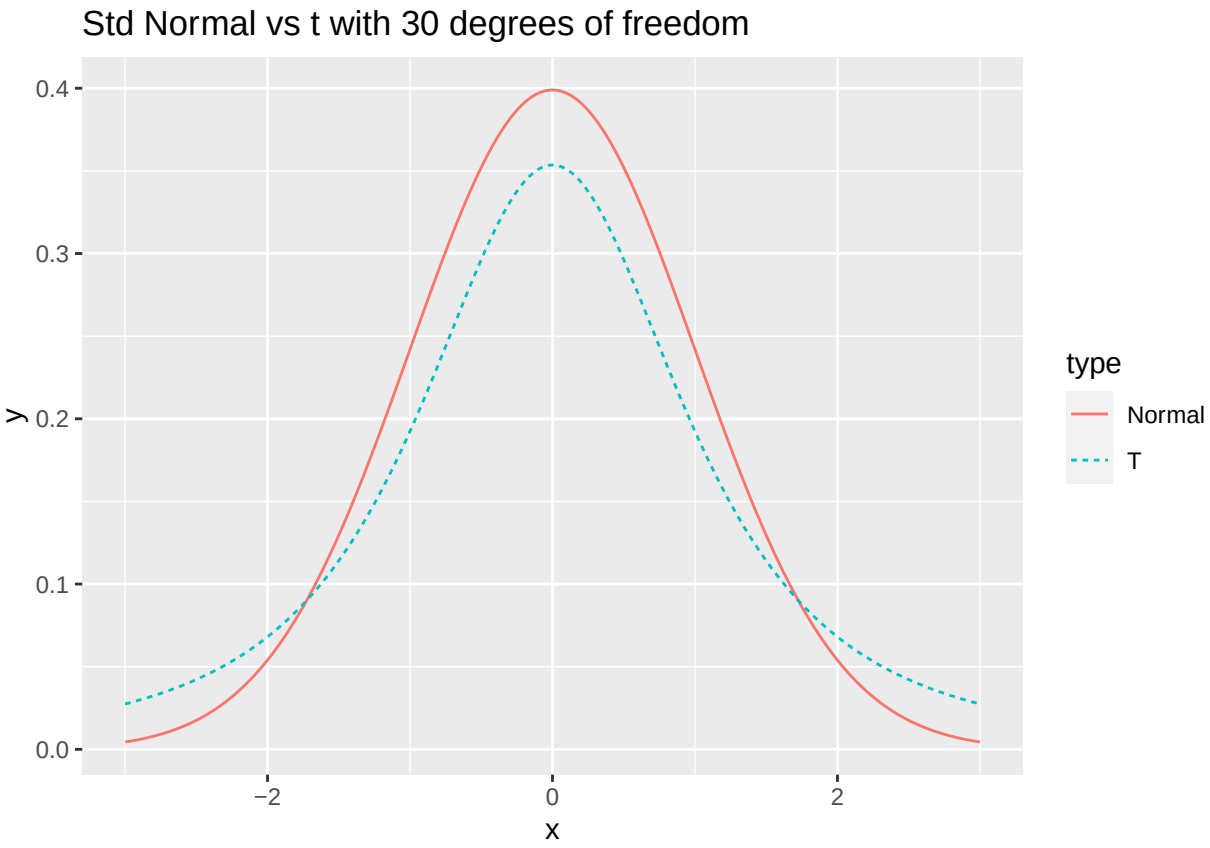

```
## [1] "In loop and df is now: 28"
```




```
## [1] "In loop and df is now: 29"
```



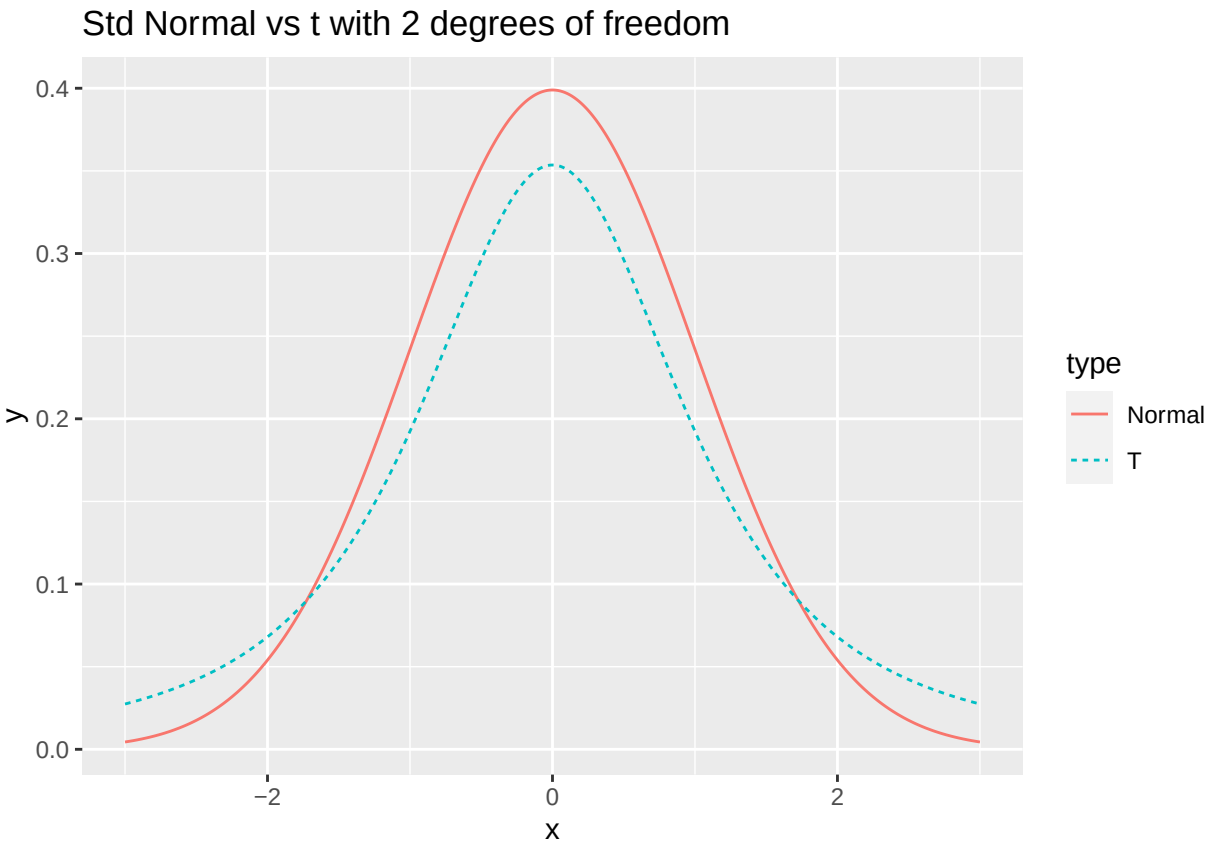

```
## [1] "In loop and df is now: 30"
```



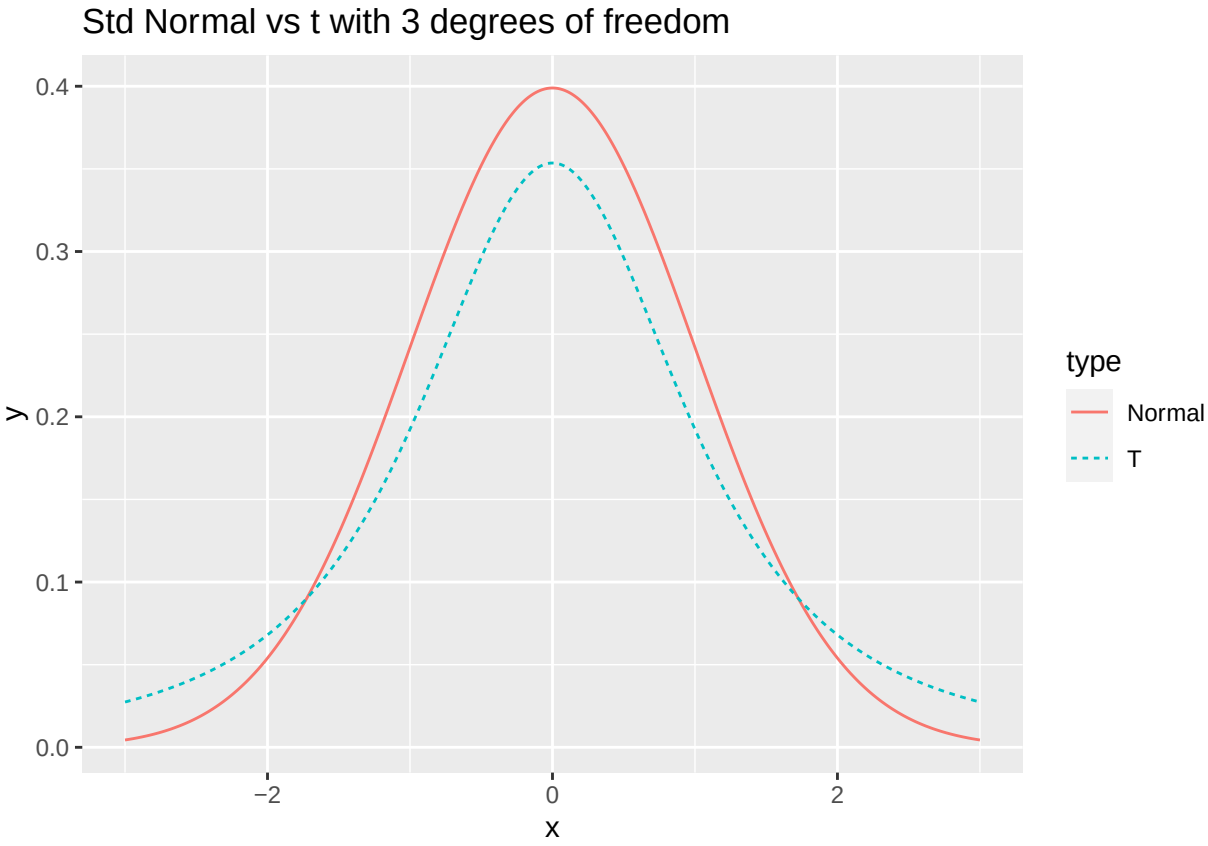
3. b.

```
library(ggplot2)
N <- 1000
df <- 2
x.grid <- seq(-3, 3, length=N)
data <- data.frame(
  x = c(x.grid, x.grid),
  y = c(dnorm(x.grid), dt(x.grid, df)),
  type = c( rep('Normal',N), rep('T',N) ) )


for( df in c(2,3,4,5,10,15,20,25,30) ){
  # print out current value of df
  print( paste("In loop and df is now:", df) )

  # make a nice graph
  myplot <- ggplot(data, aes(x=x, y=y, color=type, linetype=type)) +
  geom_line() +
  labs(title = paste('Std Normal vs t with', df, 'degrees of freedom'))

  # actually print the nice graph we made
  print(myplot)
  }
```

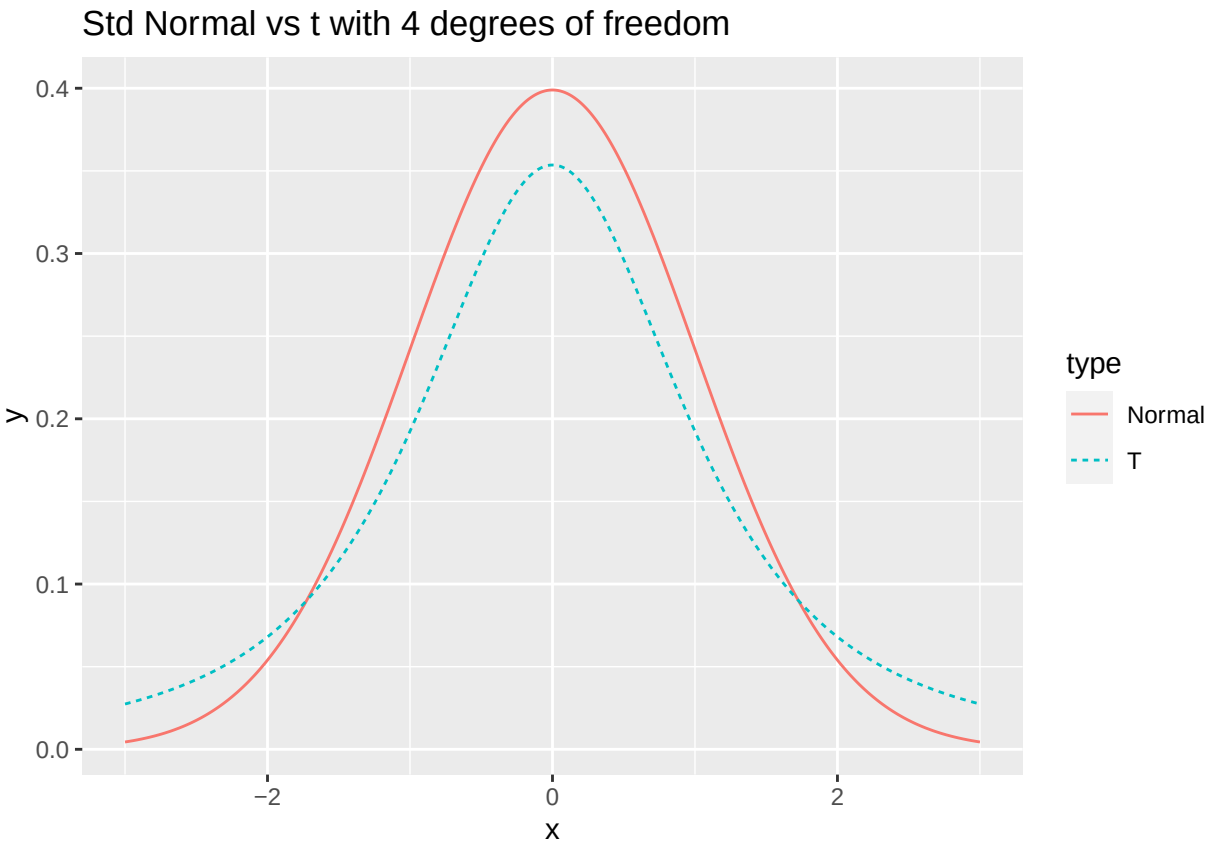
```
## [1] "In loop and df is now: 2"
```



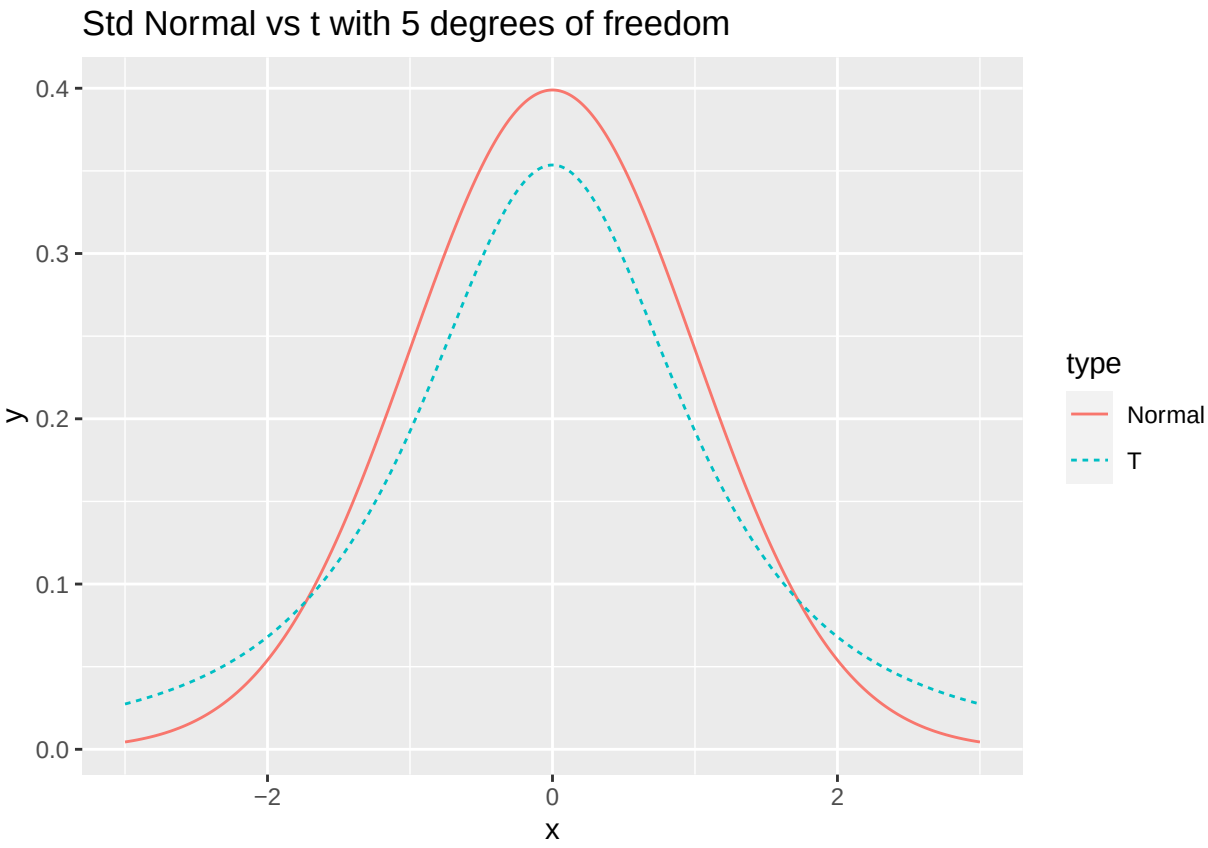

```
## [1] "In loop and df is now: 3"
```



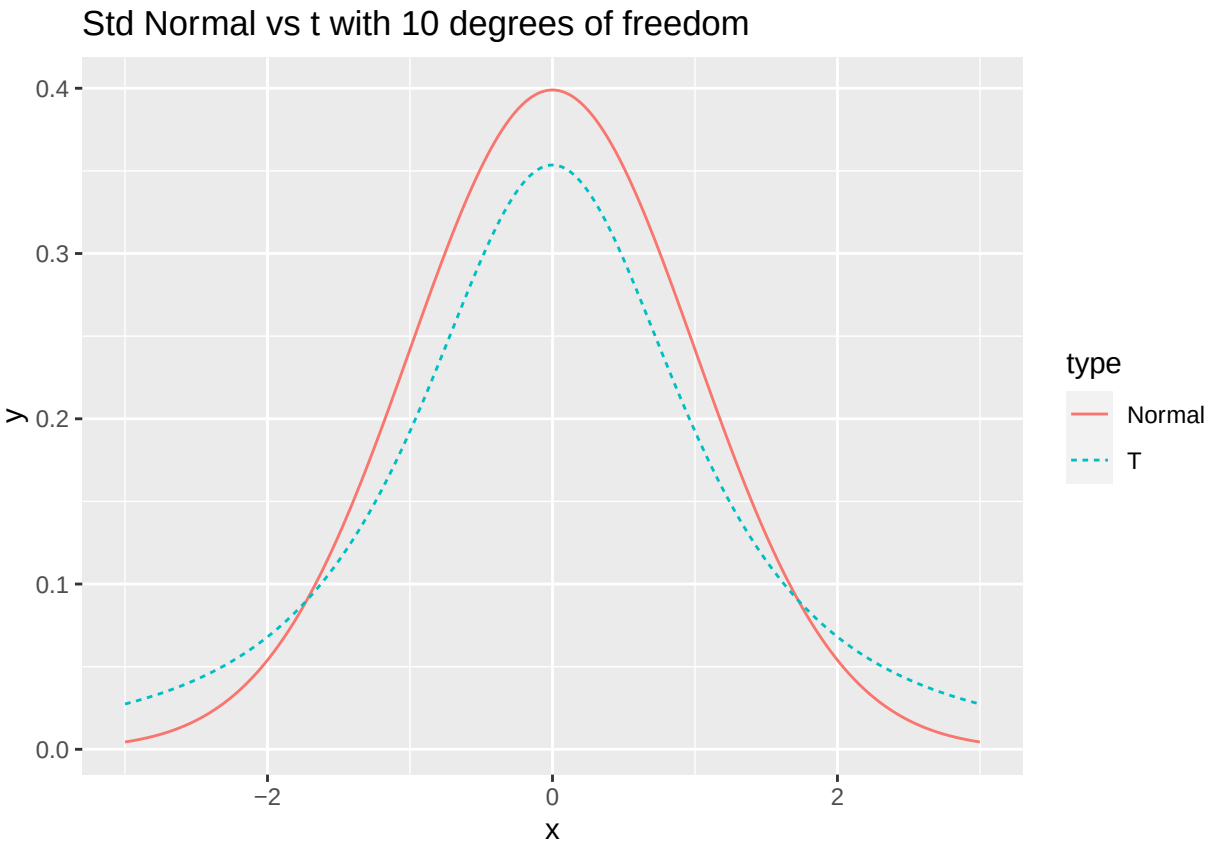

```
## [1] "In loop and df is now: 4"
```



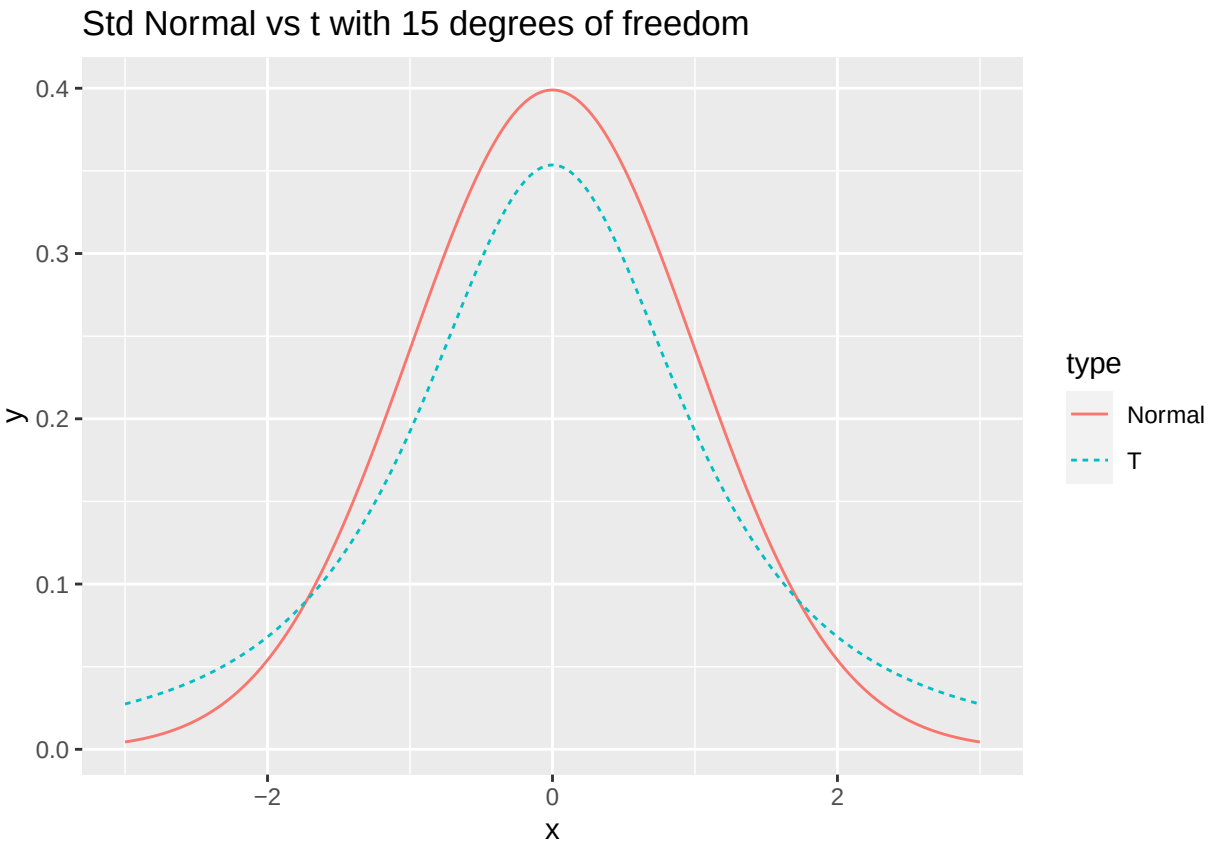

```
## [1] "In loop and df is now: 5"
```



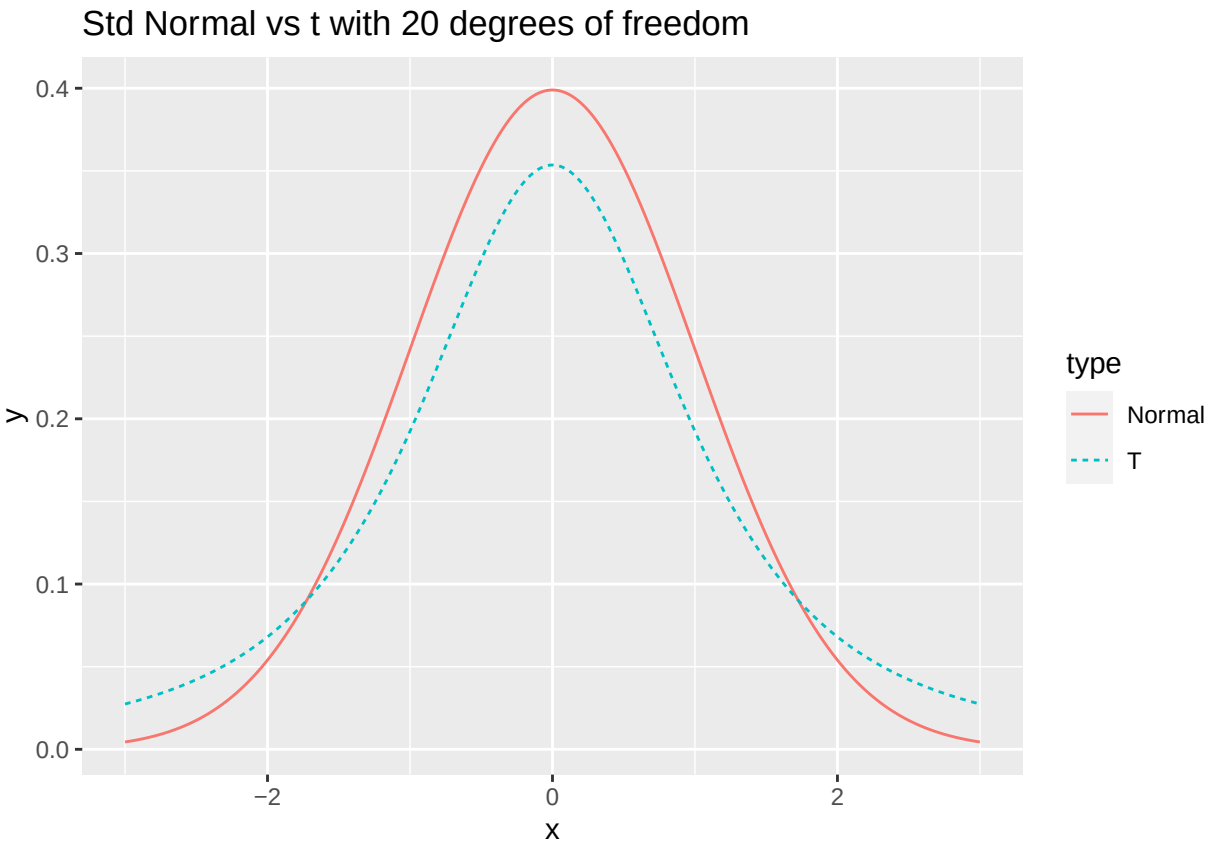

```
## [1] "In loop and df is now: 10"
```



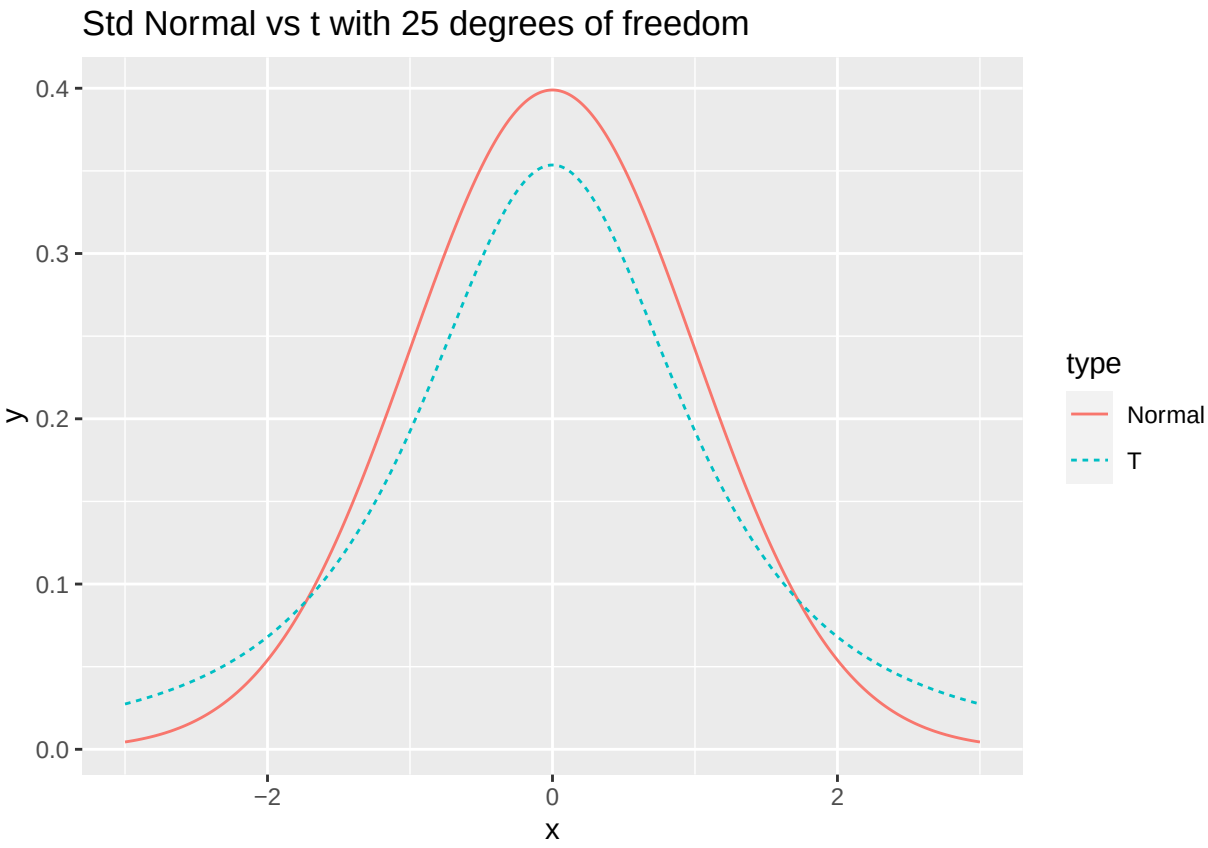

```
## [1] "In loop and df is now: 15"
```



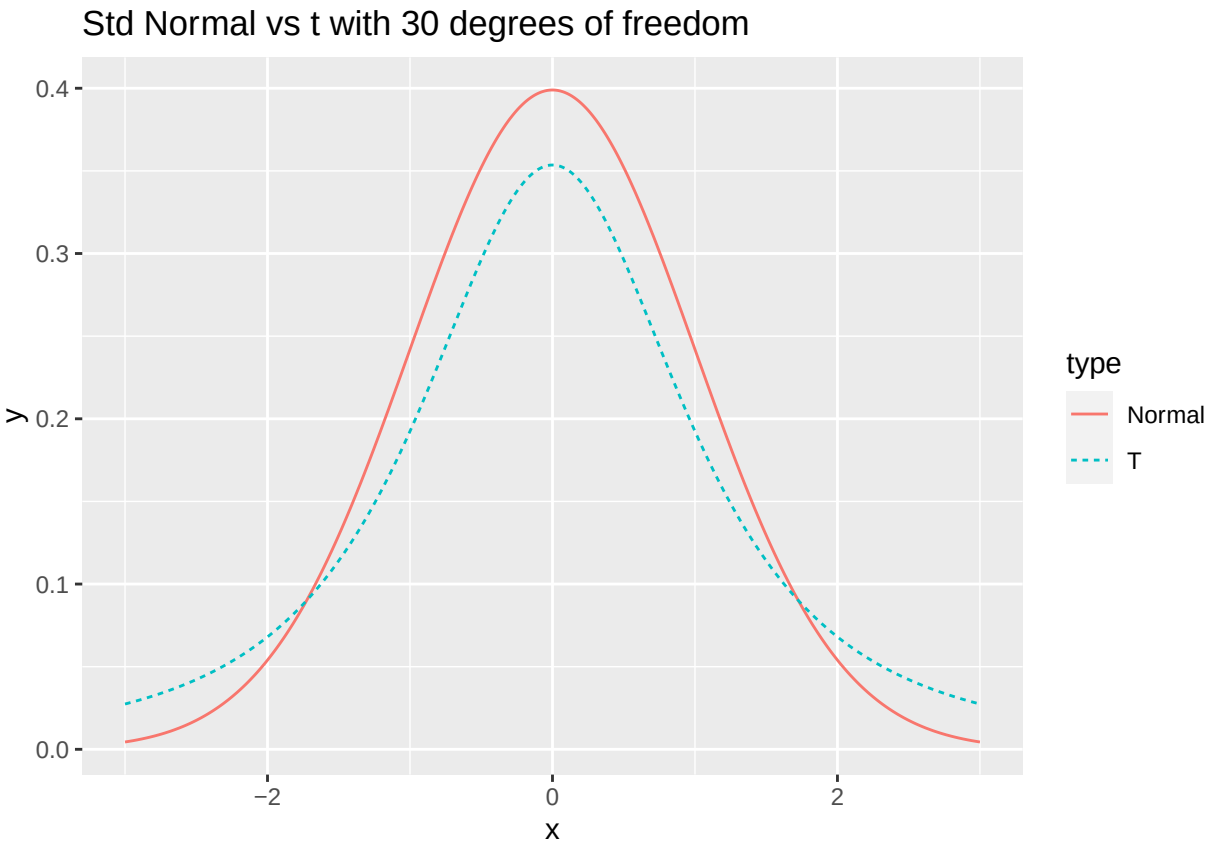

```
## [1] "In loop and df is now: 20"
```




```
## [1] "In loop and df is now: 25"
```




```
## [1] "In loop and df is now: 30"
```



4. a.

We can simulate rolling two 6-sided dice using the sample() function with the replace=TRUE option. Read the help file on sample() to see how to sample from the numbers

```
throw <- sample(1:6, size=2, replace = TRUE, prob = NULL)

throw
```

```
## [1] 3 6
```

```
str(throw)
```

```
##  int [1:2] 3 6
```

4. b.

```
throws <- NULL
print(throws)
```

```
## NULL
```

```
for( i in 1:24 ){
  throws[i] <- sample(1:6, size=2, replace = TRUE, prob = NULL) %>% sum()
print(throws[i])
game <- any( throws == 12 ) # Gives a TRUE/FALSE value
}
```

```
## [1] 6
## [1] 5
## [1] 7
## [1] 7
```

```
## [1] 4
## [1] 5
## [1] 5
## [1] 5
## [1] 8
## [1] 6
## [1] 10
## [1] 7
## [1] 8
## [1] 4
## [1] 6
## [1] 4
## [1] 9
## [1] 9
## [1] 5
## [1] 8
## [1] 6
## [1] 4
## [1] 8
## [1] 9
```

```
throws
```

```
##  [1]  6  5  7  7  4  5  5  5  8  6 10  7  8  4  6  4  9  9  5  8  6  4  8  9
```

```
print(game)
```

```
## [1] FALSE
```