Python_Module6

September 20, 2020

0.1 Author: Joseph Vargovich

[2]:
```
#Import libraries
import pandas as pd
import requests
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from numpy import random as rand
from scipy.stats import norm
from scipy.stats import t
```

1 Exercise 1 - Work with the political candidate dataset.

[3]:
```
url = "https://raw.githubusercontent.com/dereksonderegger/444/master/data-raw/
 ↪Prez_Candidate_Birthdays"
prez = pd.read_csv(url)

#a. Re-code the Gender column to have Male and Female levels. Similarly conver␣
 ↪the party variable to be Democratic or Repulican
prez = prez.assign(Party = ['Democractic' if a == "D" else 'Republican' for a␣
 ↪in prez['Party']])

#b. Change Bernie Sanders' Party to "Independent"
prez.loc[10,'Party']="Independent"
prez.tail()
```

[3]:

	Candidate	Gender	Birthday	Party	AgeOnElection
6	Amy Klobucher	F	1960-05-25	Democractic	60
7	Elizabeth Warren	F	1949-06-22	Democractic	71
8	Donald Trump	M	1946-06-14	Republican	74
9	Joe Biden	M	1942-11-20	Democractic	77
10	Bernie Sanders	M	1941-09-08	Independent	79

2 Exercise 2 - Density function of X and conditionals for valid values of x.

2.0.1 a. Fill in the conditionals to correctly caluclate the density function.

[4]:
```
a = 4
b = 10
x = rand.randint(10)  # one random value between 0 and 10

if( x < a ):
    result = 0
elif( x <= b ):
    result = 1/(b-a) #Parenthesis are important here! Order of ops.
else:
  result = 0

print(round(result, 3))
```

```
0.167
```

2.0.2 b. Create conditionals with the "and" and "or" operators for the density function

[5]:
```
#i.
x = rand.randint(0,10)  # one random value between 0 and 10
if((a<=x) and (x<=b)): # AND (&) is necessary here as both bounds apply
    result <- 1/(b-a)
else:
    result <- 0

print(round(result, 3))

#ii.
x = rand.randint(0,10) # one random value between 0 and 10
if(x<a or b<x ): # OR is needed here as either condition would break the bounds␣
 ↪of the density function
    result = 0
else:
    result = 1/(b-a)

print(round(result, 3))

#iii.
x = rand.randint(0,10)  # one random value between 0 and 10
if( a<x and x<b ):
    result =  1/(b-a)
else:
    result = 0
```

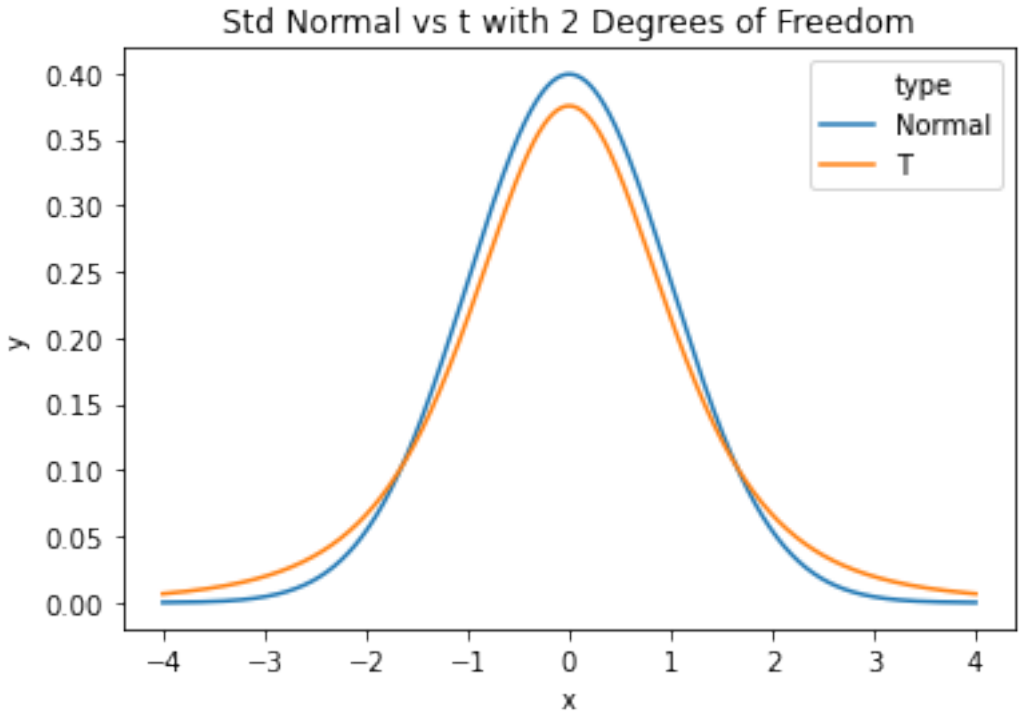
```
print(round(result, 3))
```

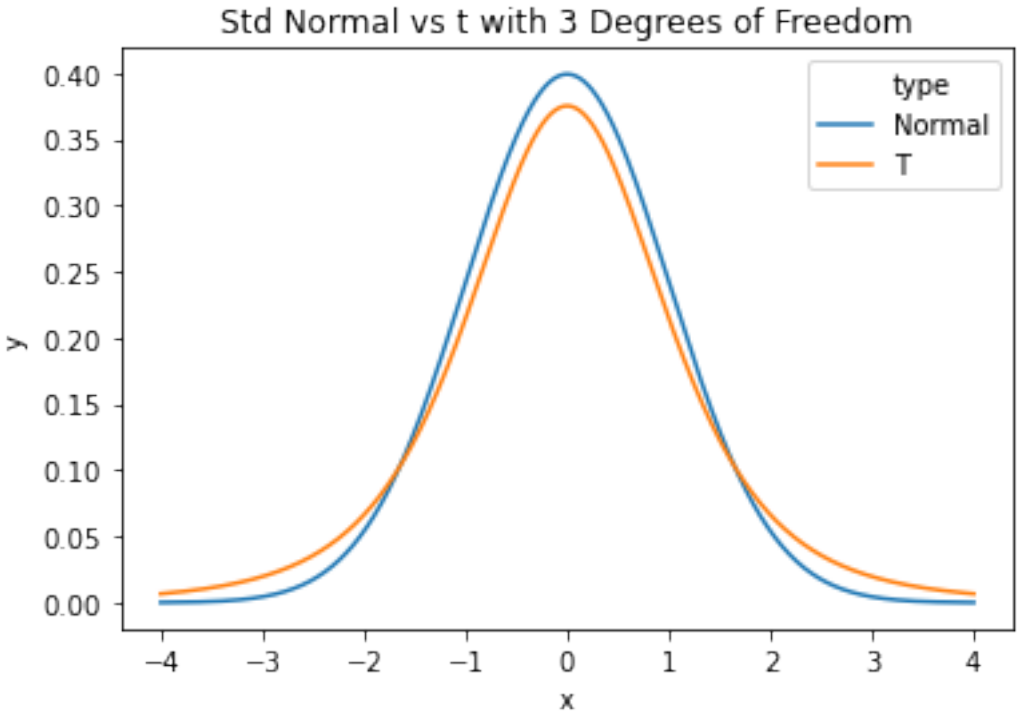
```
0.167
0.167
0.167
```

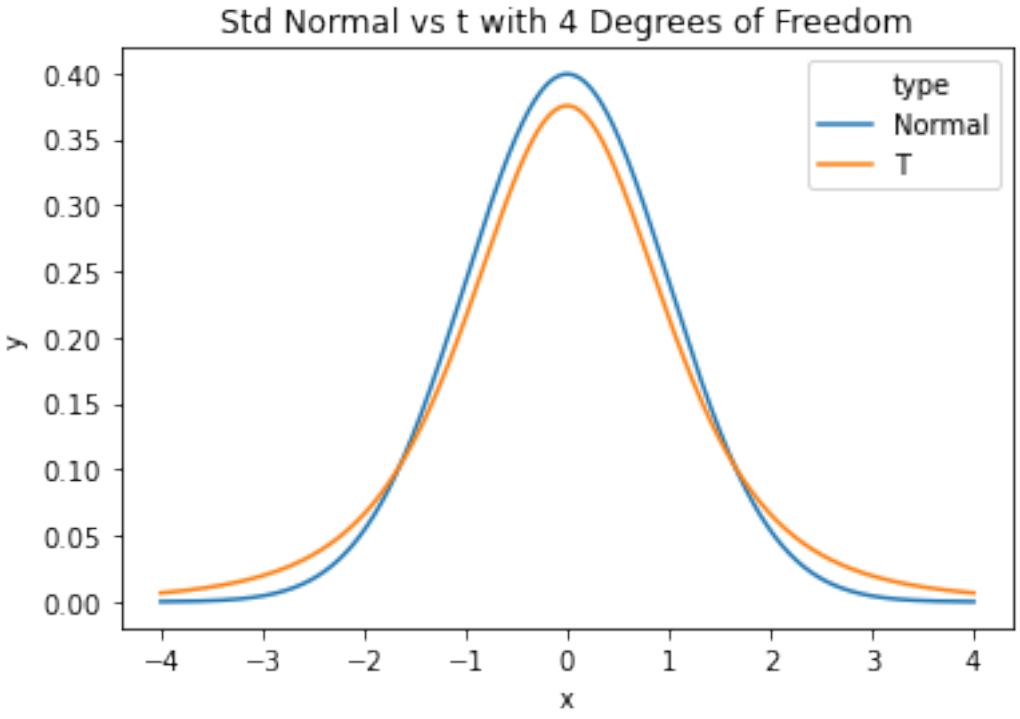
3 Exercise 3 - For loops to create multiple graphs concisely

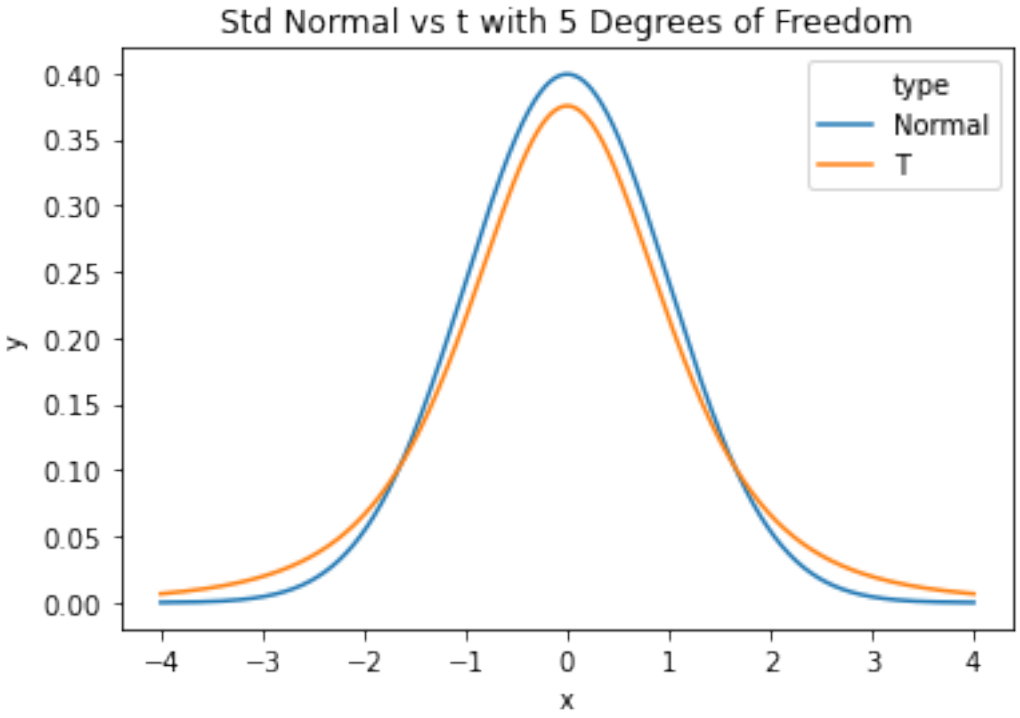
[6]:
```
### a. Use a for loop to create similar graphs for degrees of freedom 2,3,4 ...
 ↪, 29,30
```

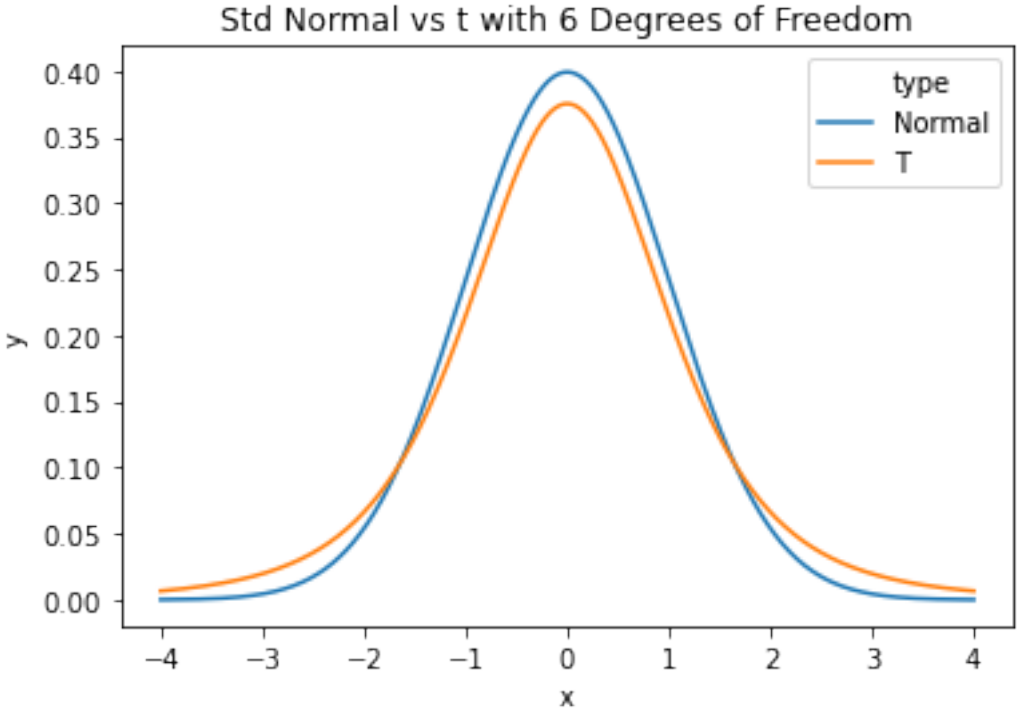
[7]:
```
df = 4 #Degrees of freedom
N = 1000 #Data points
x = np.linspace(start=-4, stop=4, num=N) #numpy's linspace is the equivalent of␣
 ↪R's seq function.
data = pd.read_csv("degreeFreedomGraph.csv")

#Create new graphs for each degree of freedom
for df in range(2,21):
    degFreedomPlot = sns.lineplot(x="x", y="y",
              hue="type",
              data=data)
    title = "Std Normal vs t with " + str(df) + " Degrees of Freedom"
    degFreedomPlot.set_title(title)
    plt.show()
```

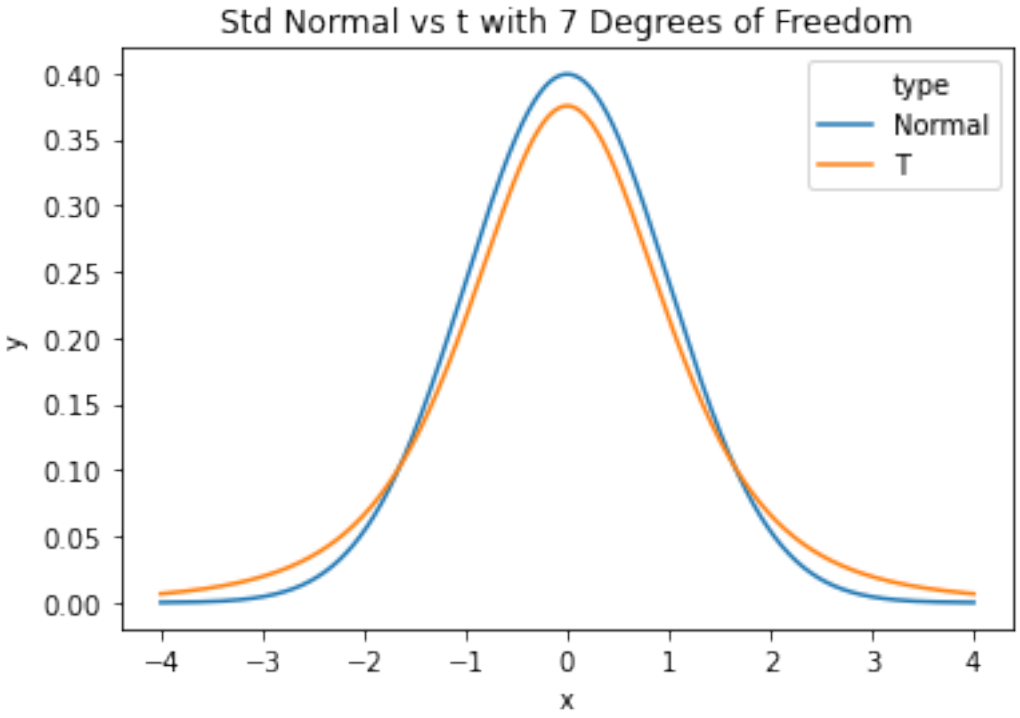


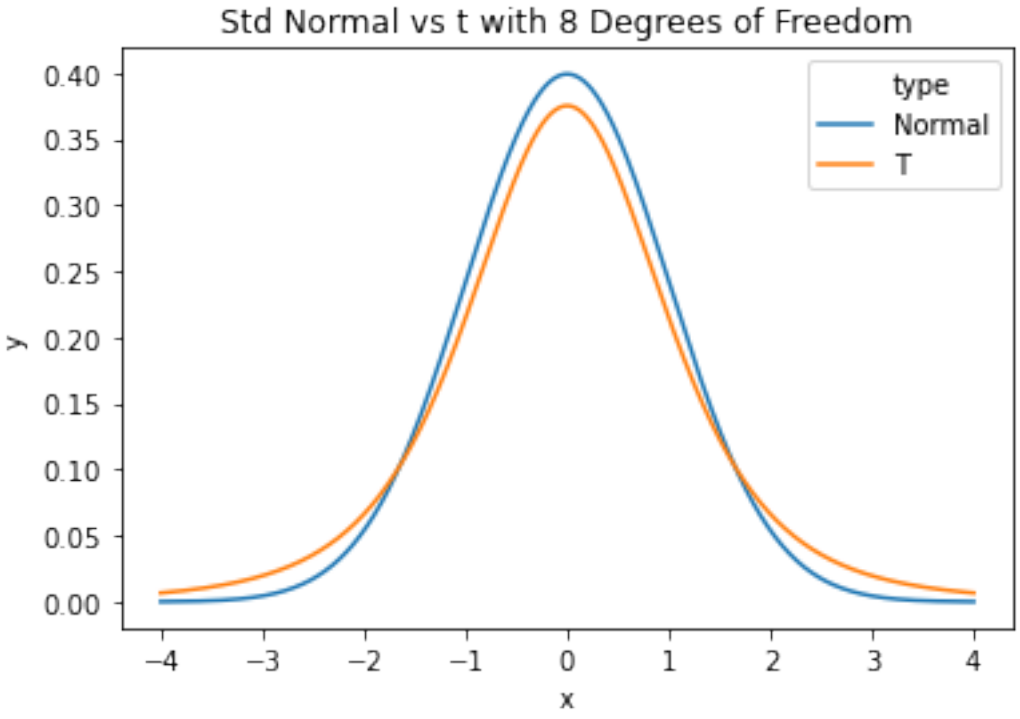

Std Normal vs t with 3 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

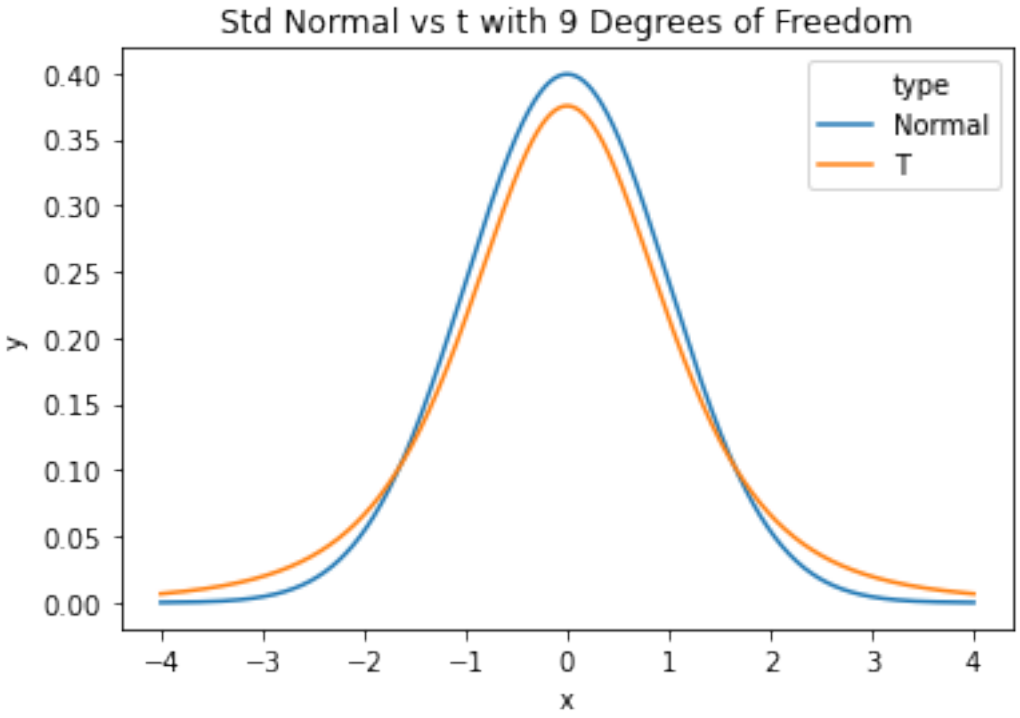

Std Normal vs t with 4 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

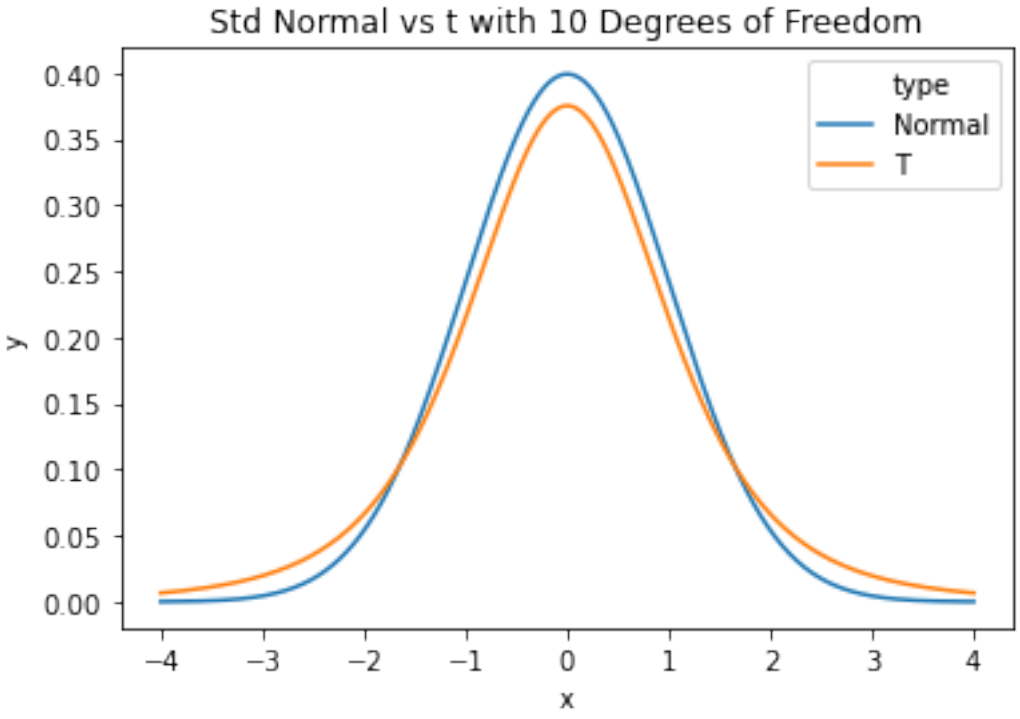

Std Normal vs t with 5 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

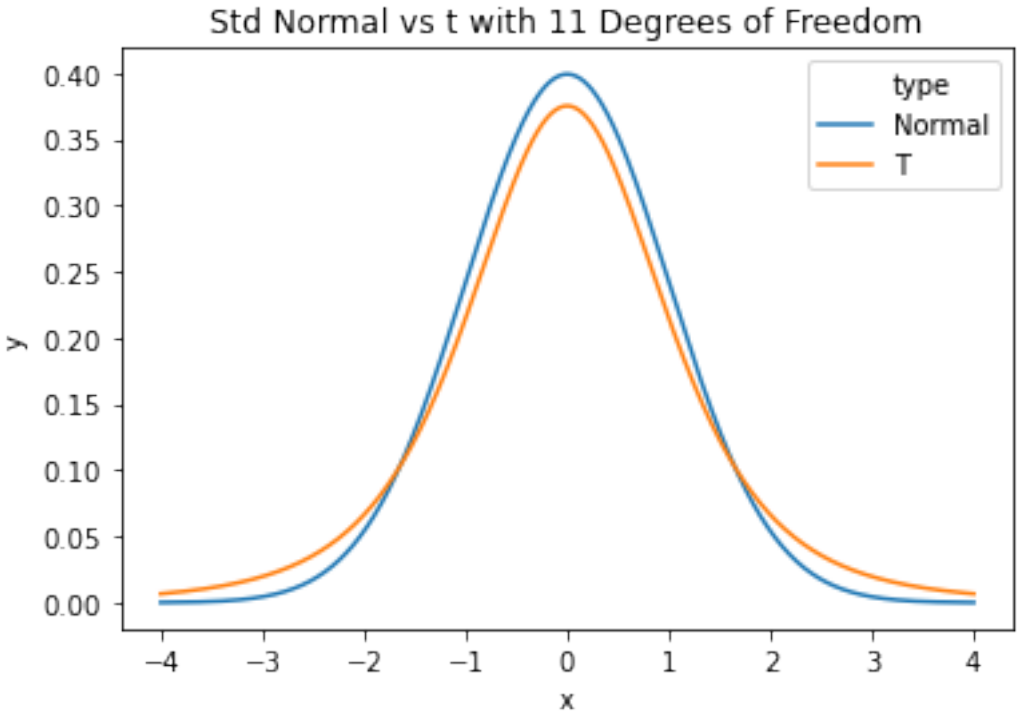

Std Normal vs t with 6 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

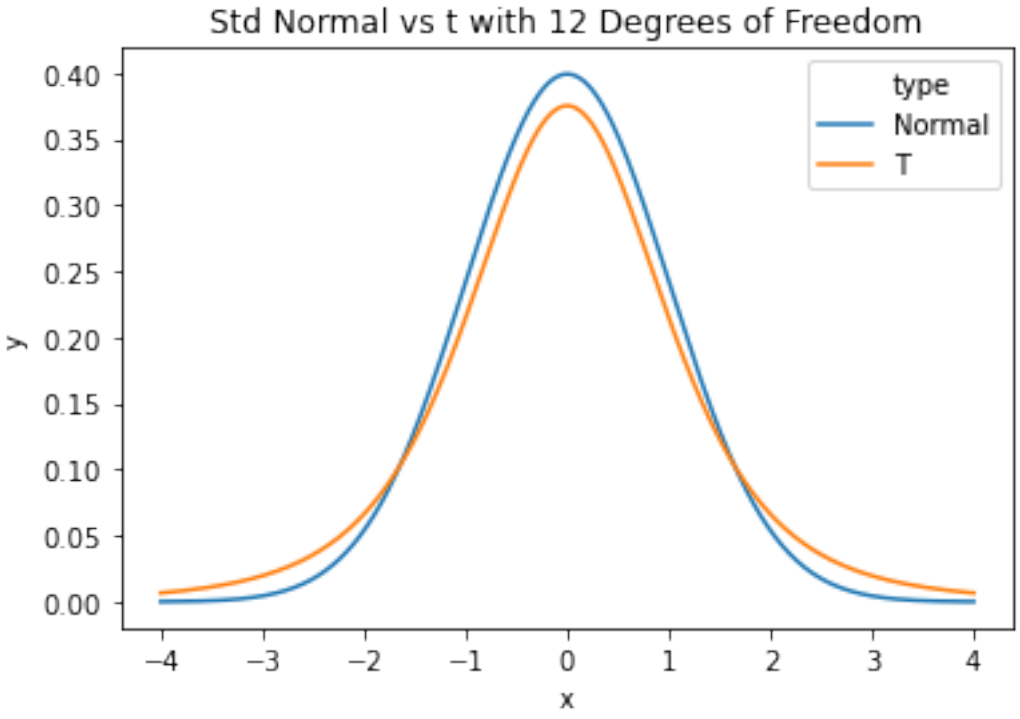

Std Normal vs t with 7 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

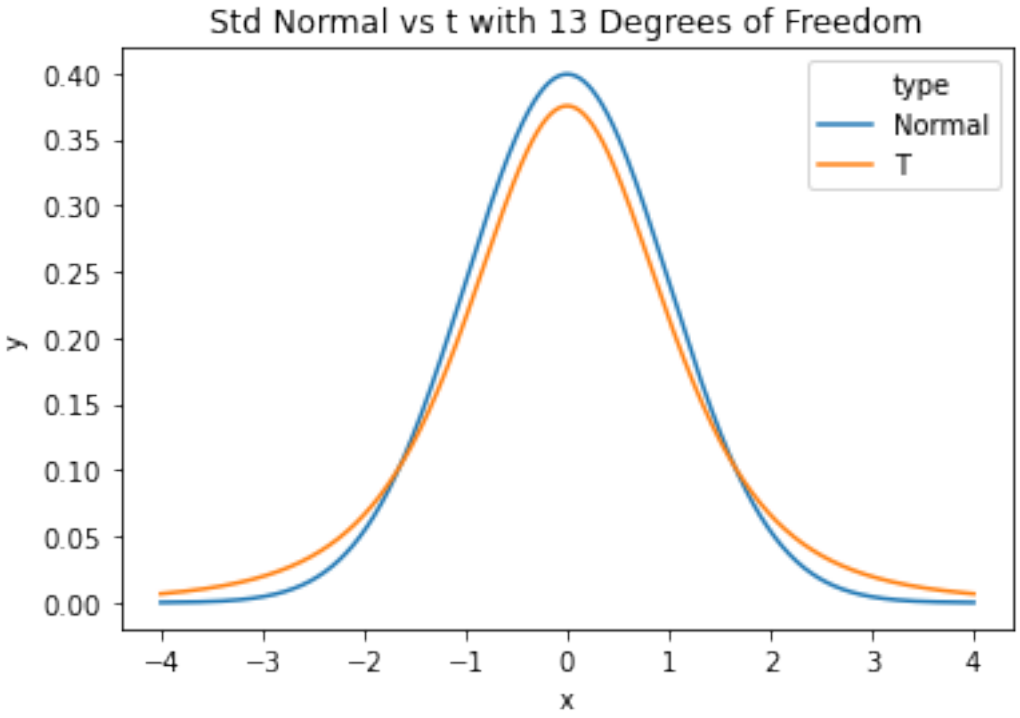

Std Normal vs t with 8 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

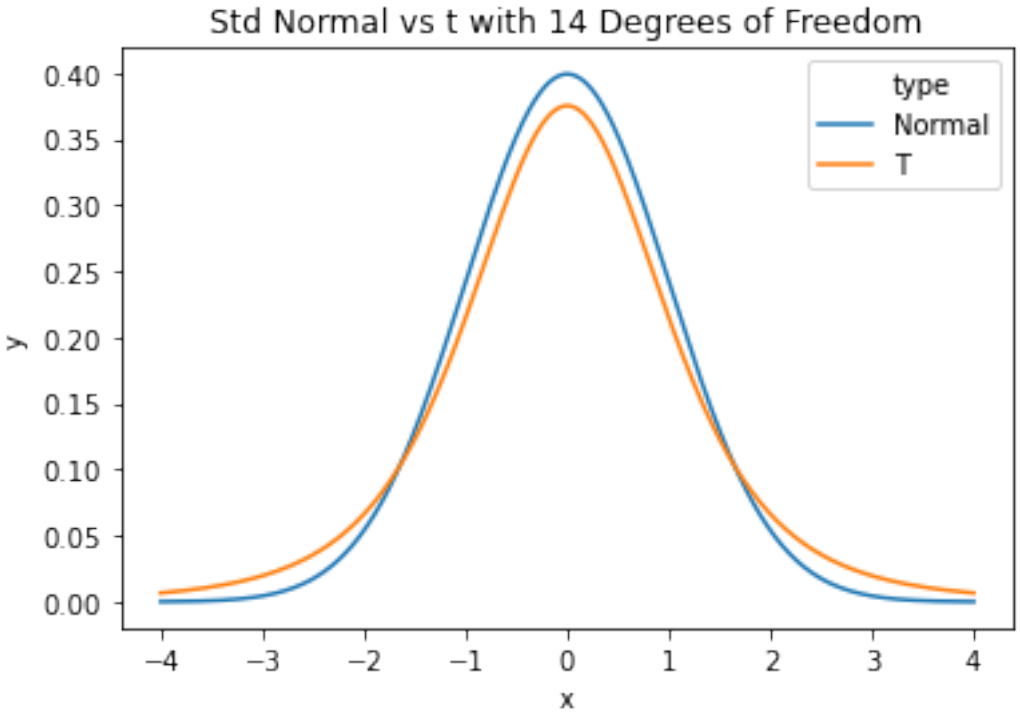

Std Normal vs t with 9 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

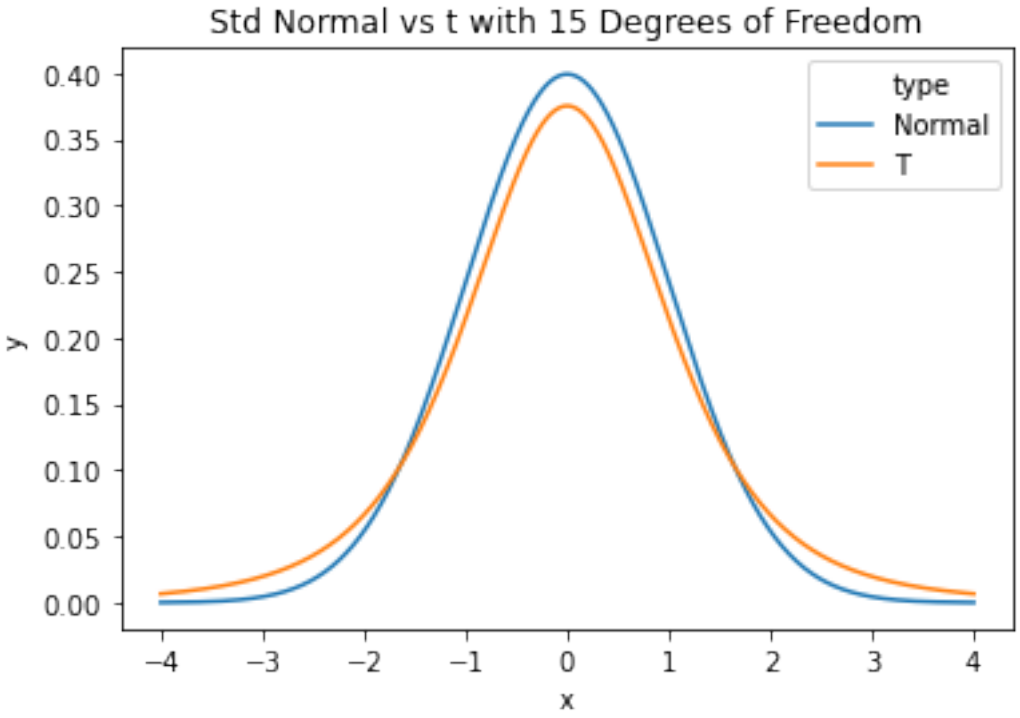

Std Normal vs t with 10 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

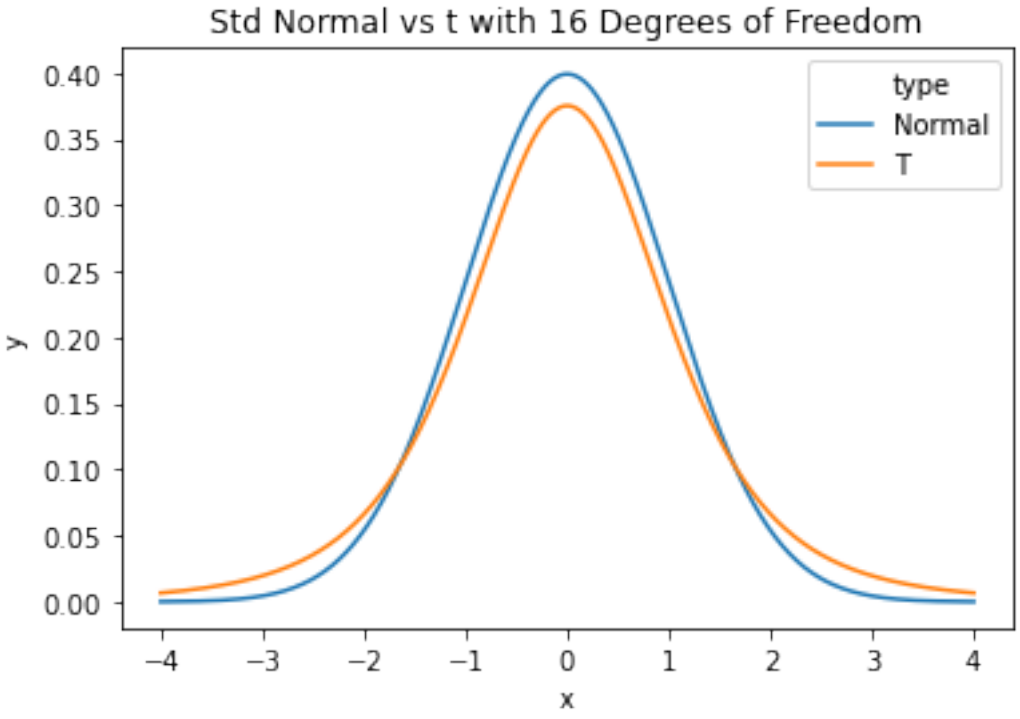

Std Normal vs t with 11 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

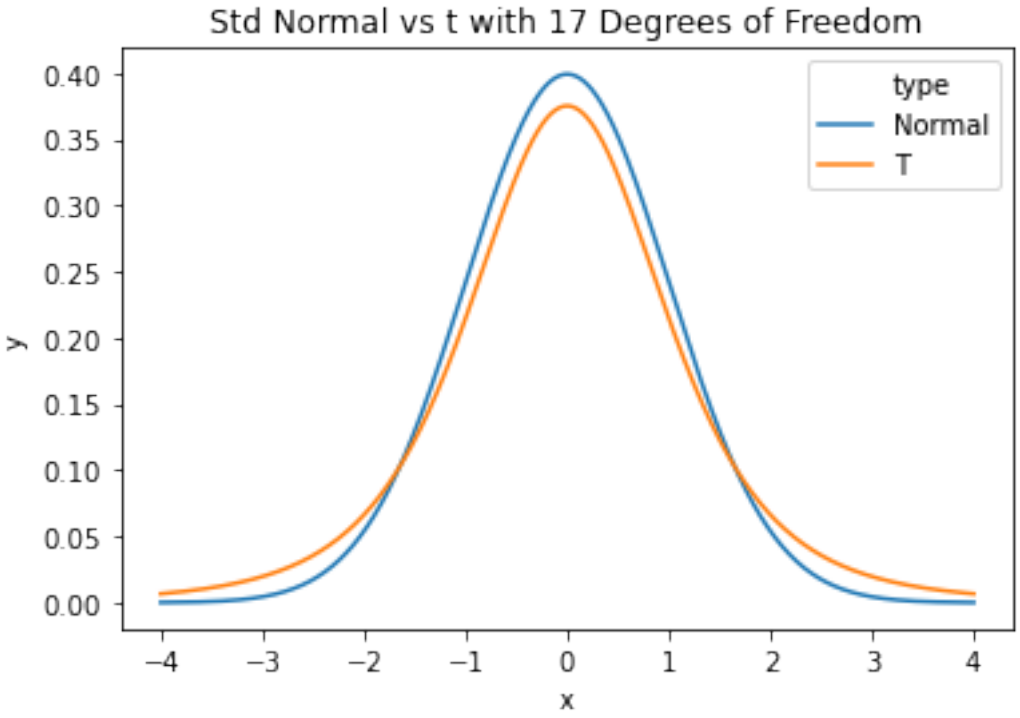

Std Normal vs t with 12 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

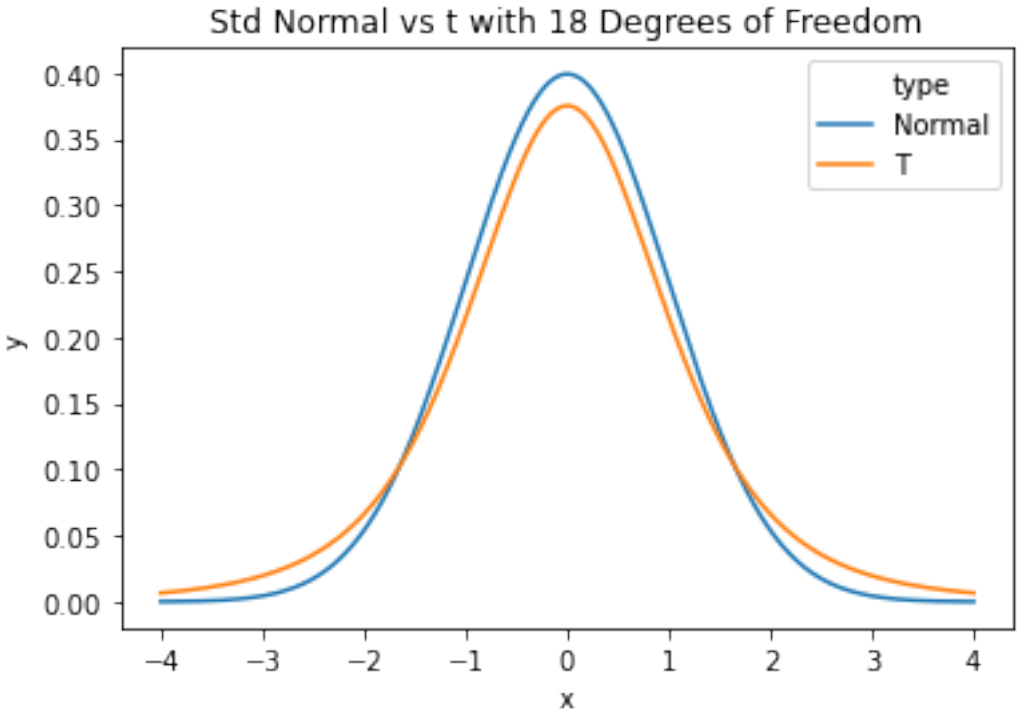

Std Normal vs t with 13 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

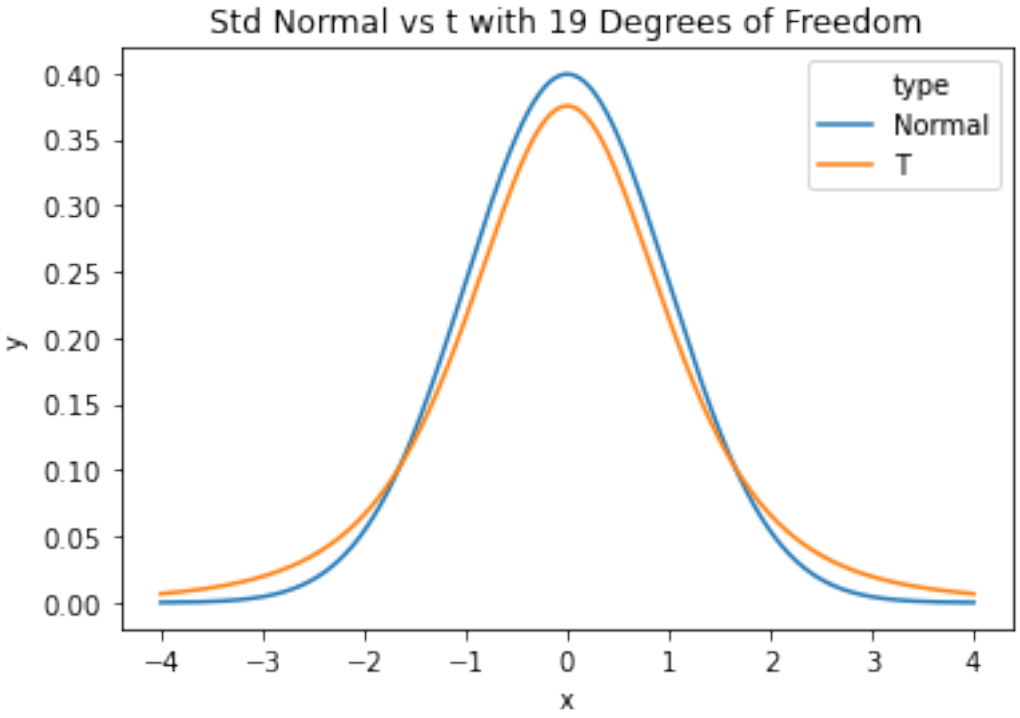

Std Normal vs t with 14 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

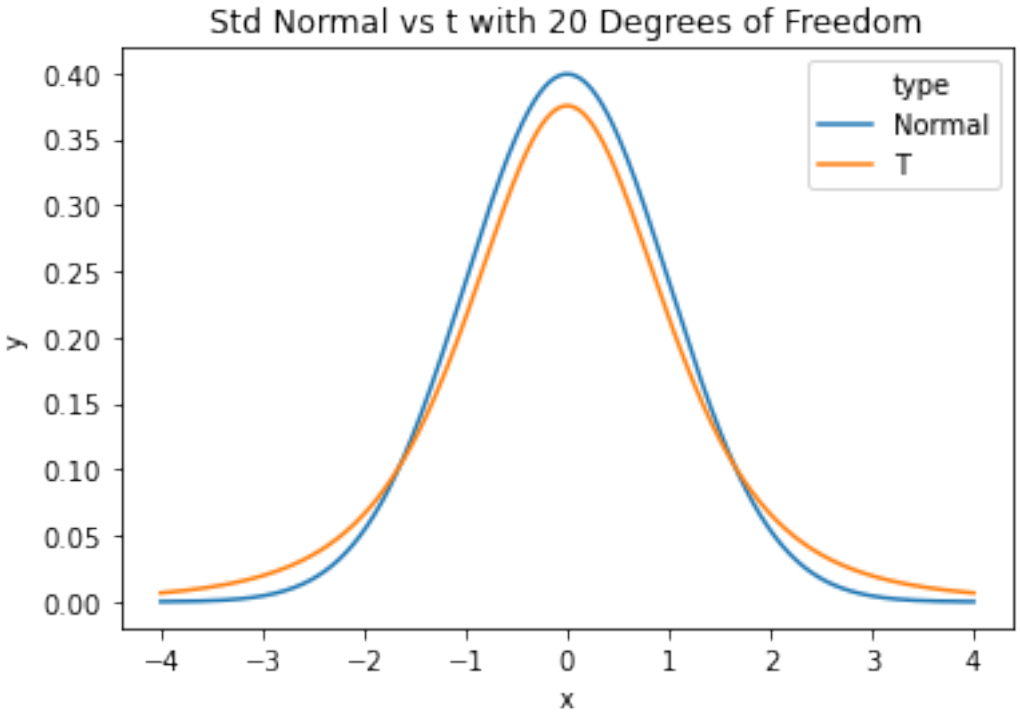

Std Normal vs t with 15 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x


Std Normal vs t with 16 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x


Std Normal vs t with 17 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

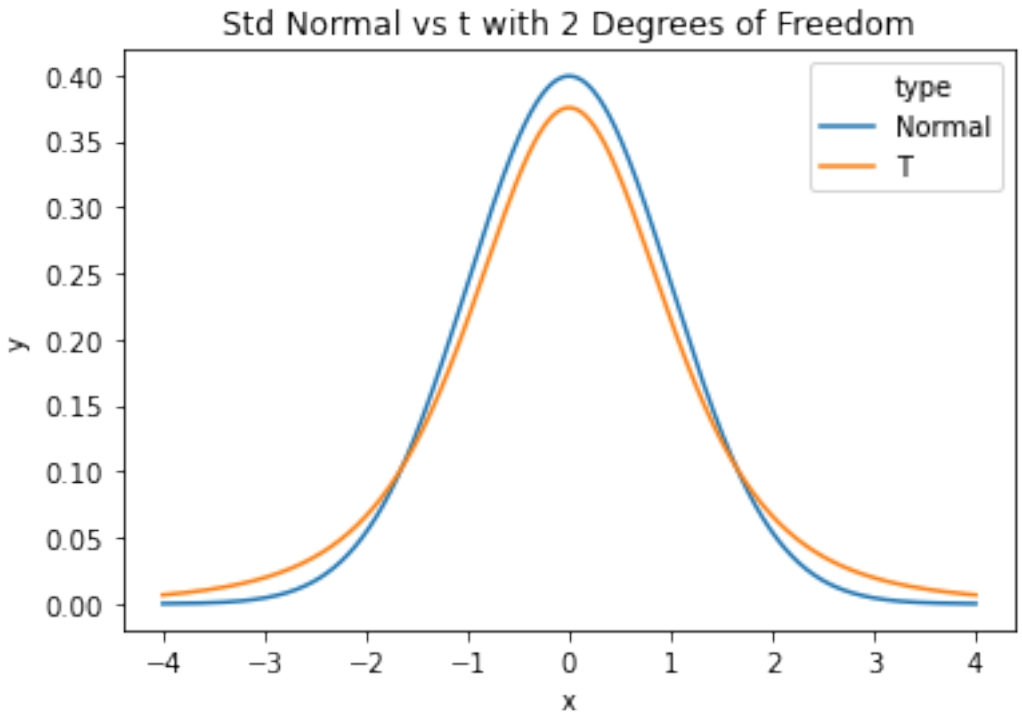

Std Normal vs t with 18 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

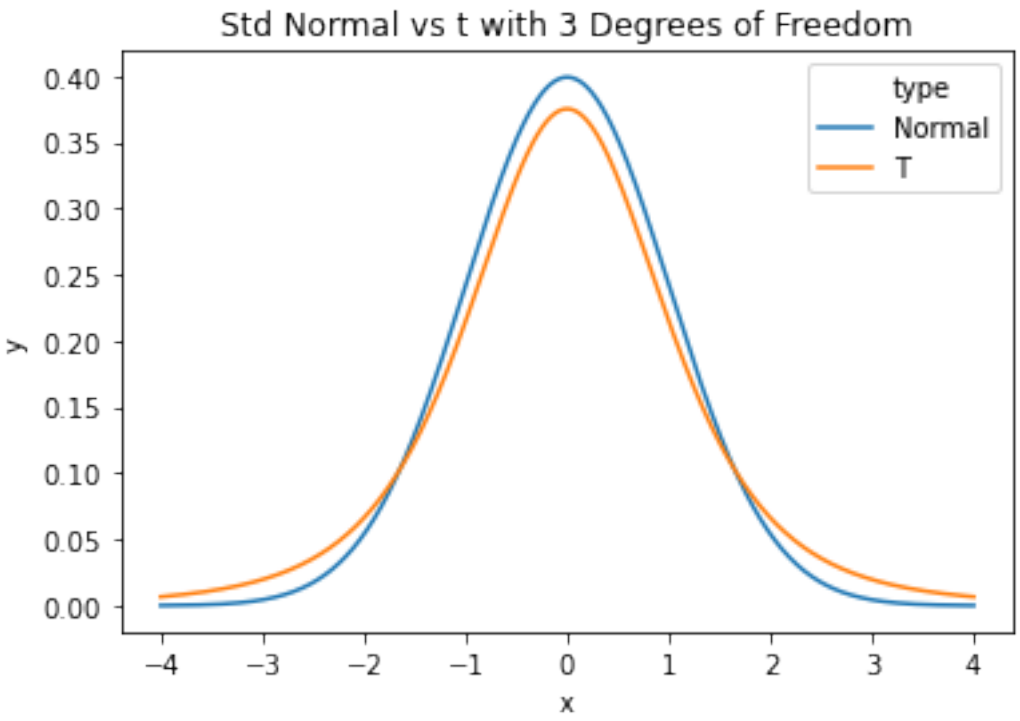

Std Normal vs t with 19 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

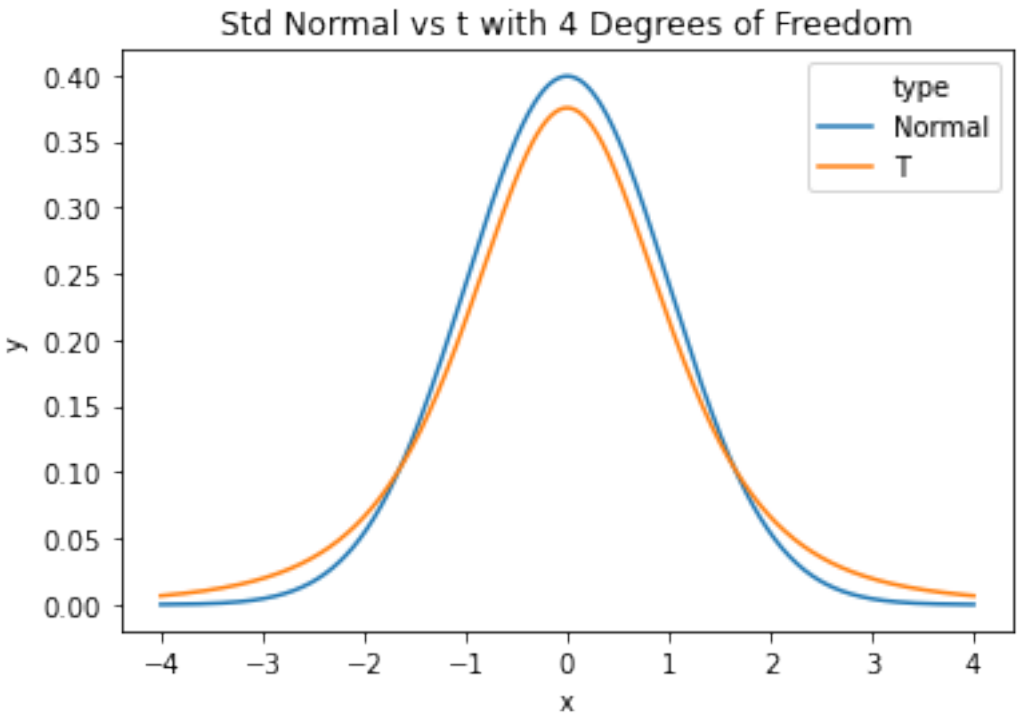

Std Normal vs t with 20 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

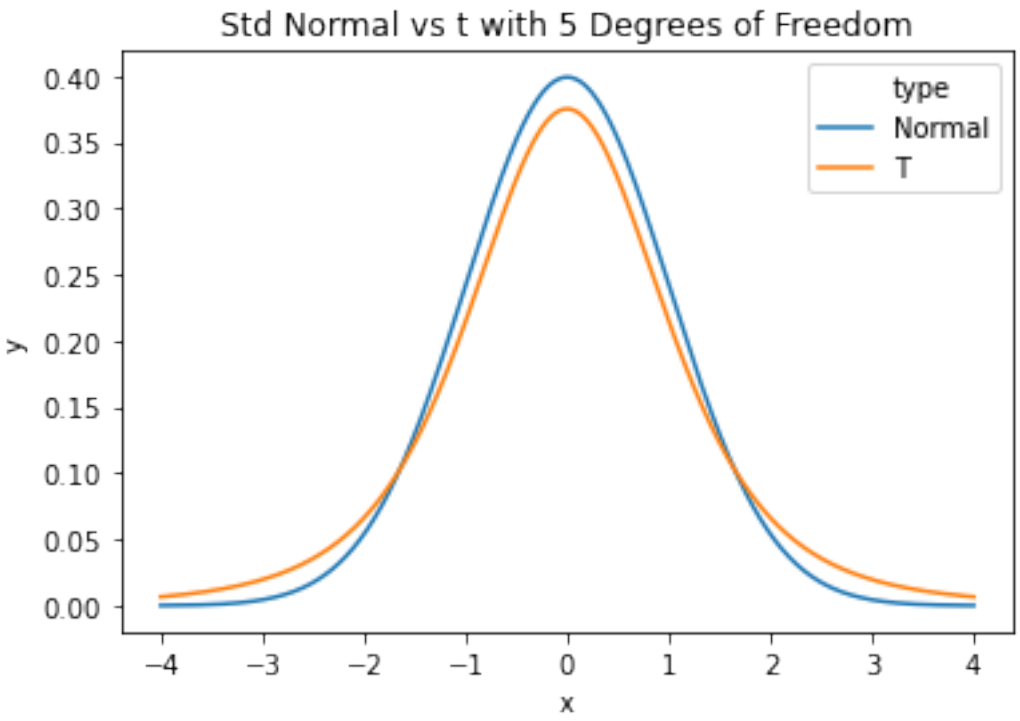
3.0.1 b. Use a for loop to create similar graphs for degrees of freedom 2,3,4,5,10,15,20 25,30

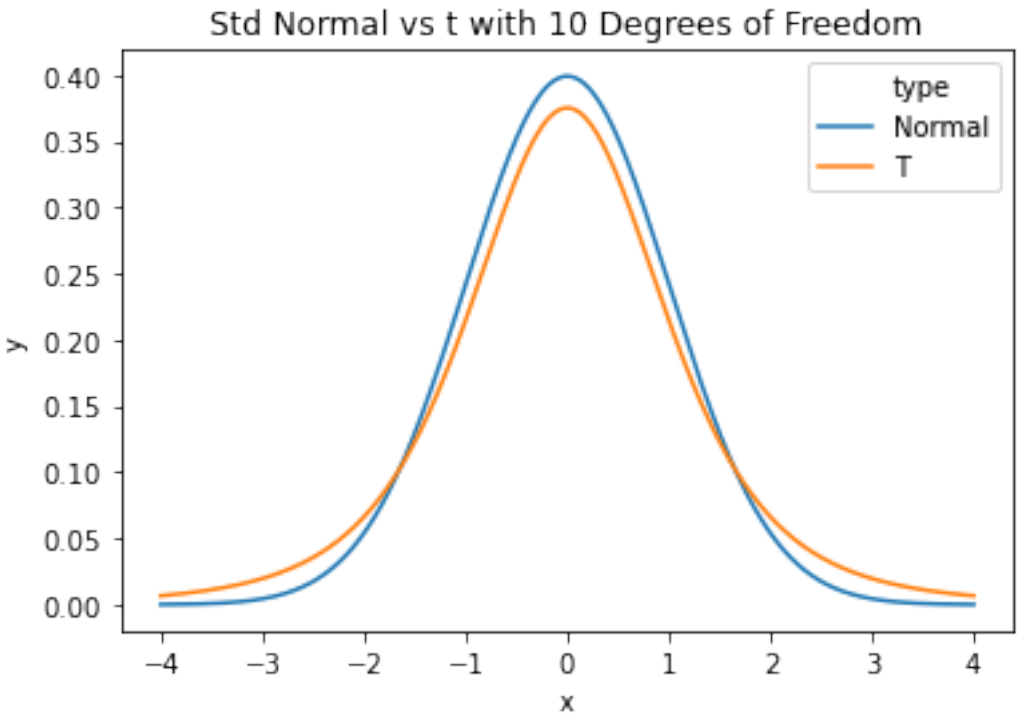
[8]:
```
# b. Just create graphs for df in a specified vector.a
dfVec = [2,3,4,5,10,15,20,25,30]

#Create and print each graph for each df value in the list.
for df in dfVec:
    degFreedomPlot = sns.lineplot(x="x", y="y",
             hue="type",
             data=data)
    title = "Std Normal vs t with " + str(df) + " Degrees of Freedom"
    degFreedomPlot.set_title(title)
    plt.show()
```

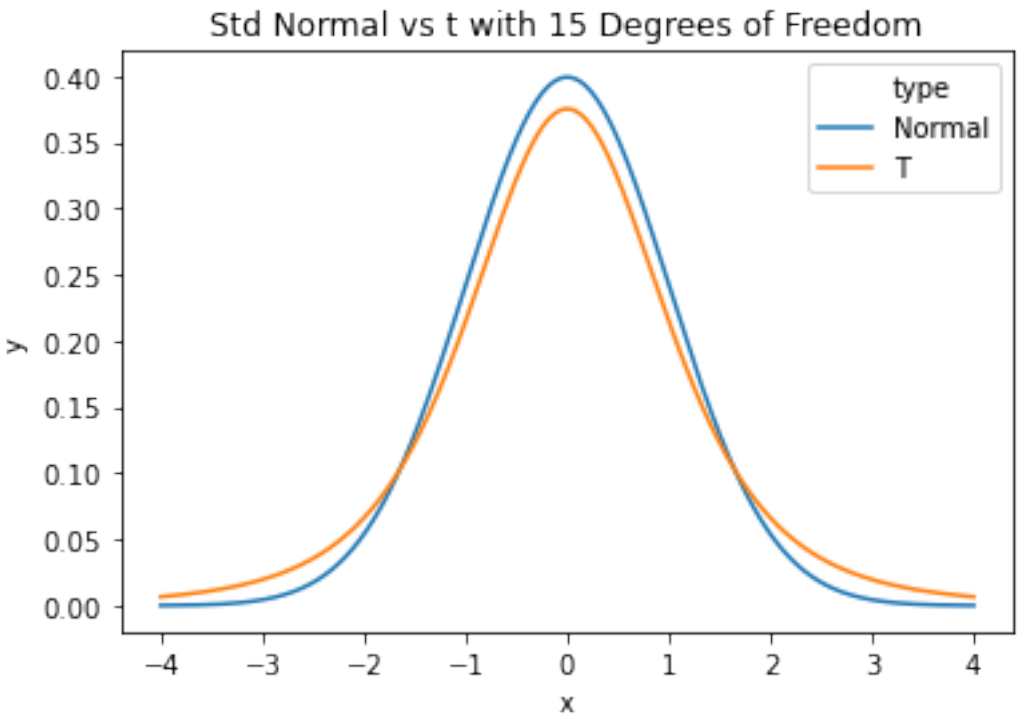


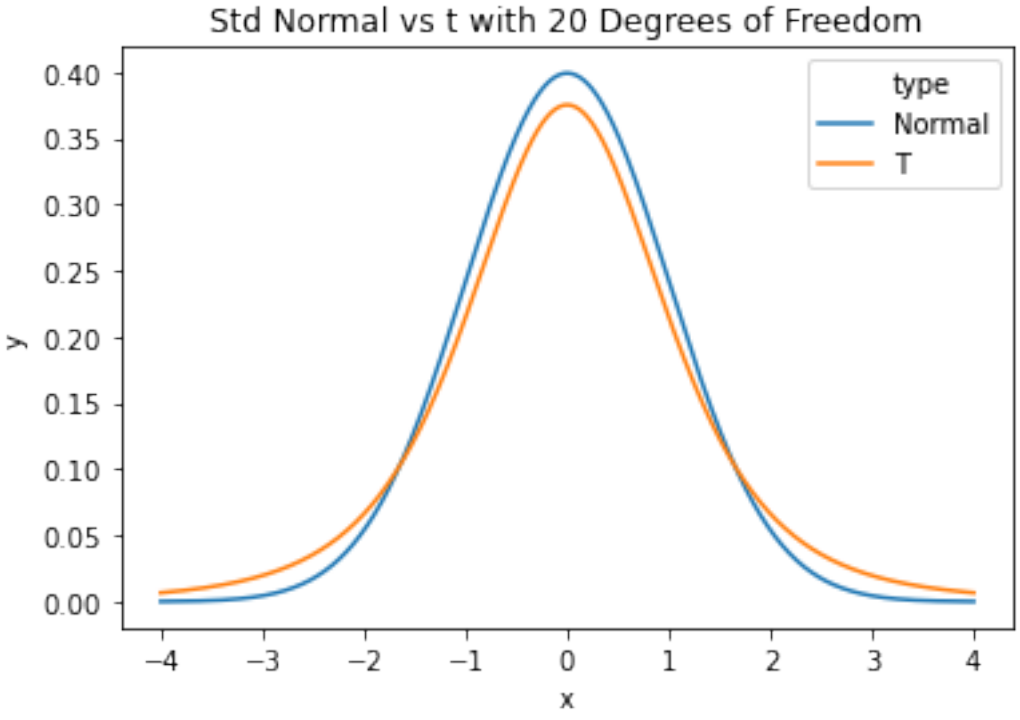

Std Normal vs t with 3 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

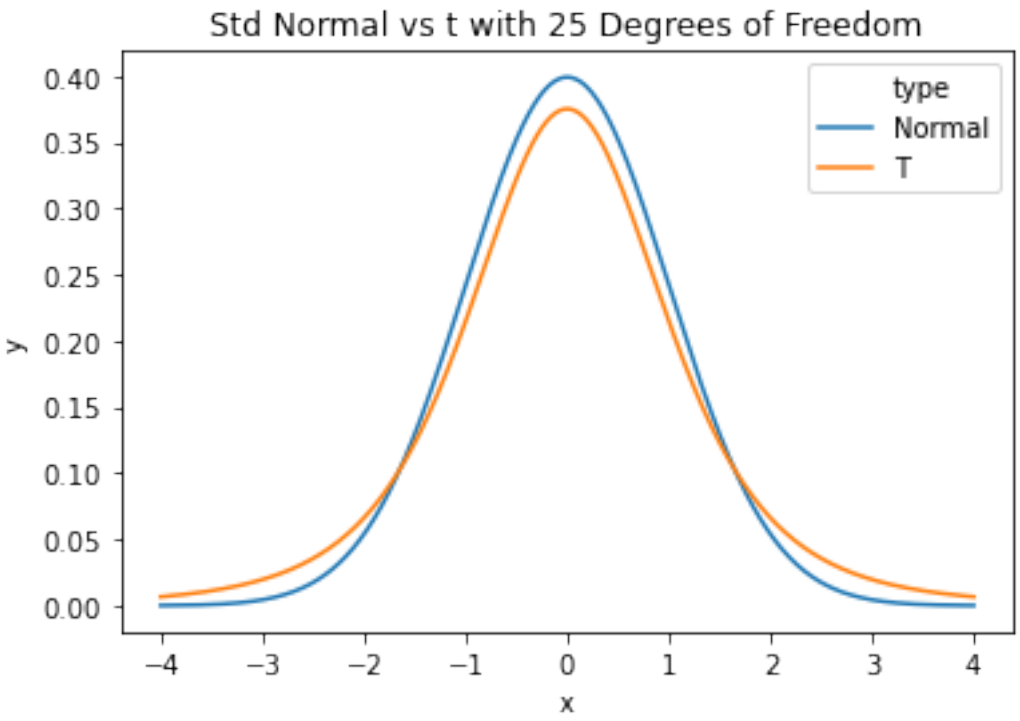

Std Normal vs t with 4 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x

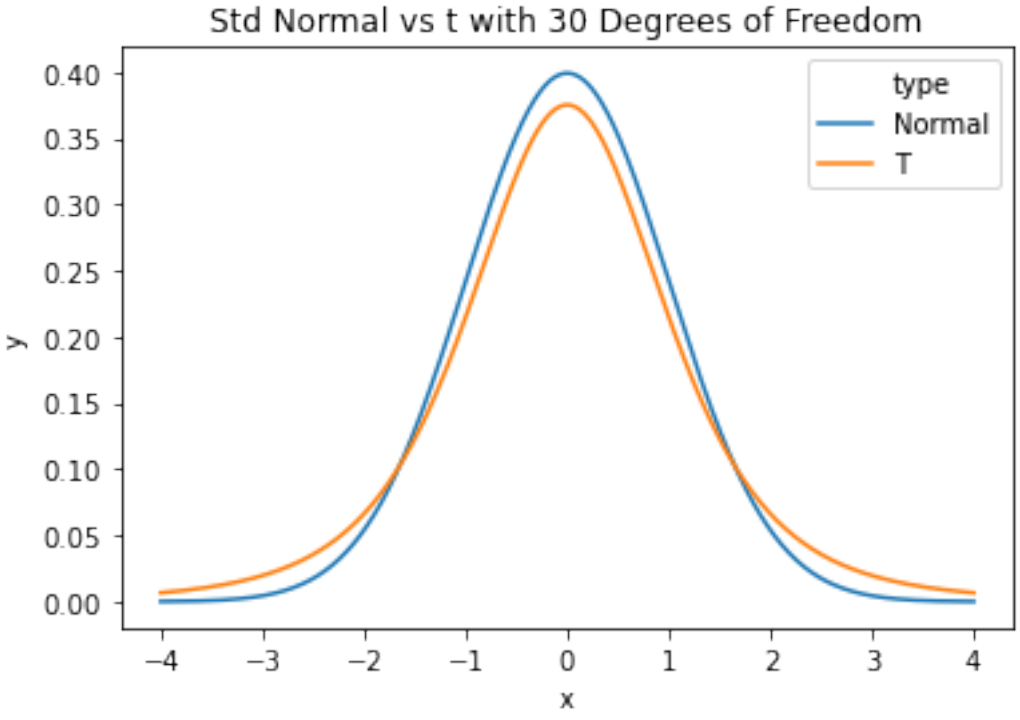

Std Normal vs t with 5 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x


Std Normal vs t with 10 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x


Std Normal vs t with 15 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x


Std Normal vs t with 20 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x


Std Normal vs t with 25 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x


Std Normal vs t with 30 Degrees of Freedom
type
Normal
T
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
y
−4
−3
−2
−1
0
1
2
3
4
x