Southern University of Science And Technology
School of Life Sciences
BIO210 Biostatistics
Extra Reading Material

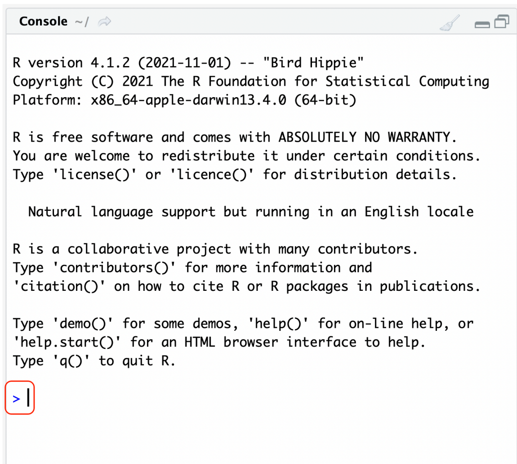
Fall, 2023 Lecture 20

1 Calculating Probability Using R

As we gradually introduce more and more statistic concepts and tests, we will need to do quite a few complicated calculations. Some of them cannot be computed by hand or regular calculator; others can, but there are just too many steps and they are error-prone. Therefore, we introduce how to use the programming language **R** for the calculation of probabilities and related stuff. You don't really need to know how to write scripts in **R**. For now, you can just use it as a calculator.

Go to the SUSTech mirror of **R**: https://mirrors.sustech.edu.cn/CRAN. Download and install the program based on your operating system.

If you open the program, you will probably see a window like shown in the screenshot on the right-handside. The ">" mark that I highlighted at the bottom is your "prompt", where you can type commands. Like I said before, for now, you don't really need to know how to write scripts in **R**. You just use it as a probability calculator. Later on, we will see how to perform statistical tests.



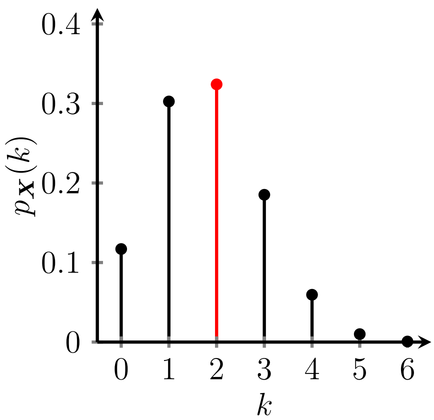
Since we already introduced how to calculate probability using **Excel**, we will use that as the comparison. Let's start with simple examples of three different distributions: $\boldsymbol{B(6, 0.3)}$, $\boldsymbol{Pois(2)}$ and $\boldsymbol{\mathcal{N}(1, 4)}$. In **R**, if you want the cumulative probability, the function always starts with "p"; if you want the probability mass/density, the function always starts with "d". For example, `pbinom`, `ppois`, and `pnorm` will give us the cumulative probability of the binomial, Poisson and normal distributions, respectively. They are like the `*.DIST` functions in **Excel** with the "cumulative" option set to "`TRUE` or `1`".

Similarly, the `dbinom`, `dpois` will give the exact probability of a binomial and Poisson distributions respectively, and `dnorm` will give you the probability density of a normal distribution. They are like the `*.DIST` functions in **Excel** with the "cumulative" option set to "`FALSE` or `0`".

If you want to do the opposite, that is, provide the cumulative probability and return the value of the random variable, the function always starts with "q", like `qbinom`, `qpois` and `qnorm`. They are like the `*.INV` functions in **Excel**.

Just to give you some visual explanation of typical calculations:

$\boldsymbol{B(6, 0.3)}$



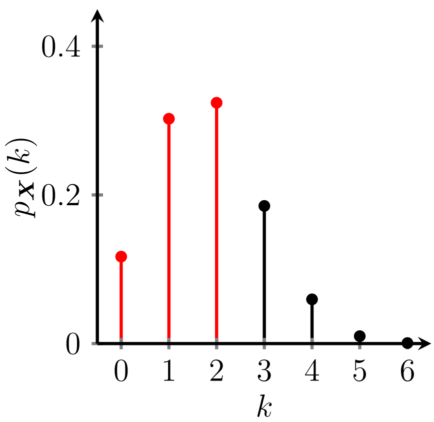
Excel:

`BINOM.DIST(2, 6, 0.3, FALSE)`

R:

`dbinom(x = 2, size = 6, prob = 0.3)`

$\boldsymbol{B(6, 0.3)}$



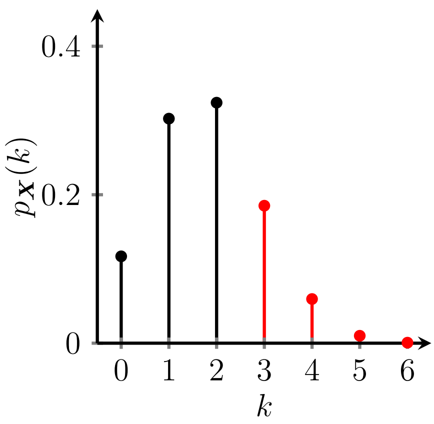
Excel:

`BINOM.DIST(2, 6, 0.3, TRUE)`

R:

```
pbinom(q = 2, size = 6, prob = 0.3,
       lower.tail = T)
```

$B(6, 0.3)$



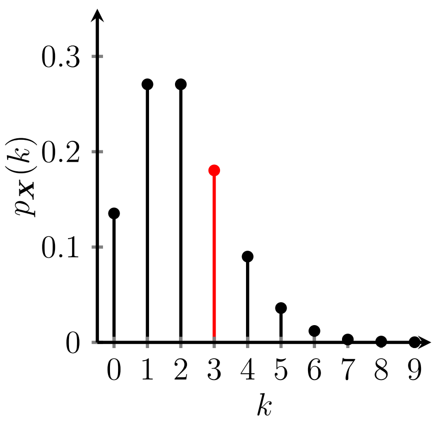
Excel:

`1 - BINOM.DIST(2, 6, 0.3, TRUE)`

R:

```
pbinom(q=2, size=6, prob=0.3,
        lower.tail=FALSE)
```

$Pois(2)$



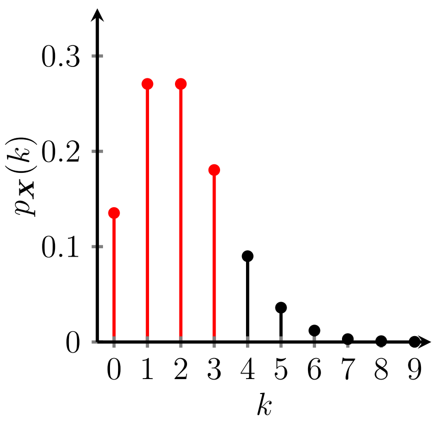
Excel:

`POISSON.DIST(3, 2, FALSE)`

R:

`dpois(x = 3, lambda = 2)`

$Pois(2)$

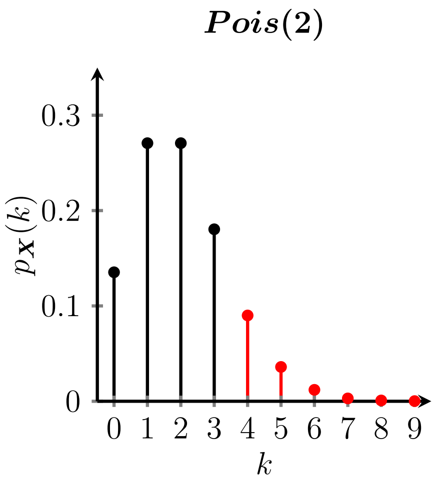


Excel:

`POISSON.DIST(3, 2, TRUE)`

R:

```
ppois(q = 3, lambda = 2,
       lower.tail = TRUE)
```

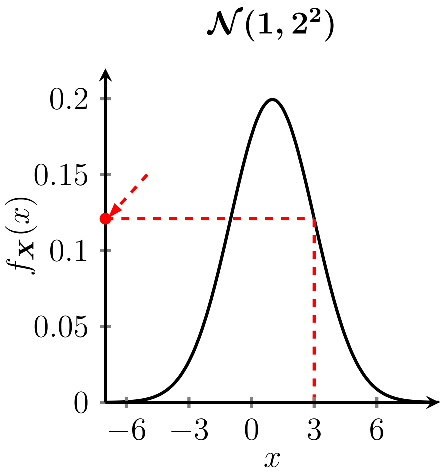


Excel:

```
1 - POISSON.DIST(3, 2, TRUE)
```

R:

```
ppois(q = 3, lambda = 2,
      lower.tail = FALSE)
```

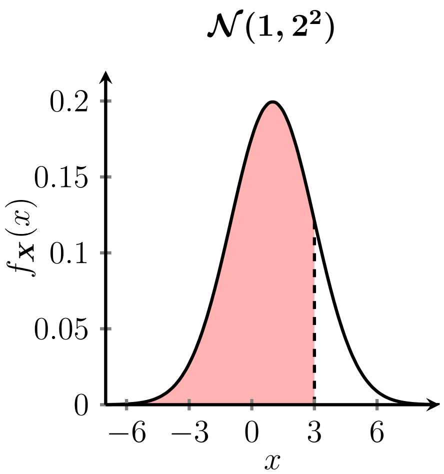


Excel:

```
NORM.DIST(3, 1, 2, FALSE)
```

R:

```
dnorm(x = 3, mean = 1, sd = 2)
```

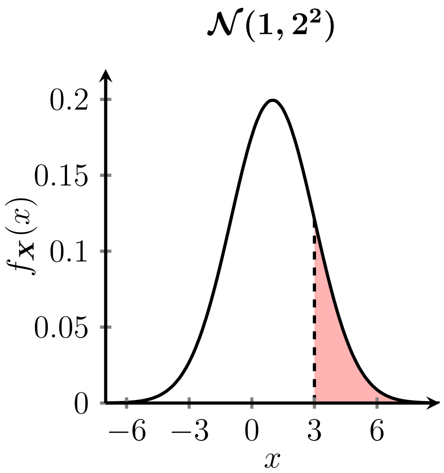


Excel:

```
NORM.DIST(3, 1, 2, TRUE)
```

R:

```
pnorm(q = 3, mean = 1, sd = 2,
      lower.tail = TRUE)
```



Excel:

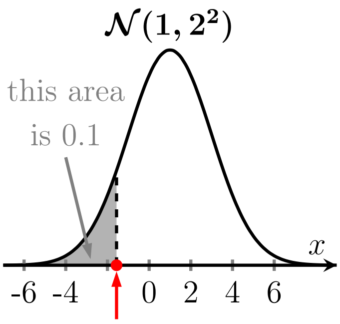
```
1 - NORM.DIST(3, 1, 2, TRUE)
```

R:

```
pnorm(q = 3, mean = 1, sd = 2,
      lower.tail = FALSE)
```

2 Find x with certain probability in a normal distribution

Again, using the function from **Excel** as a comparison, visually:

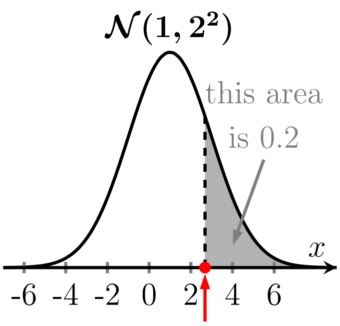


Excel:

```
NORM.INV(0.1, 1, 2)
```

R:

```
qnorm(p = 0.1, mean = 1, sd = 2,
      lower.tail = TRUE)
```



Excel:

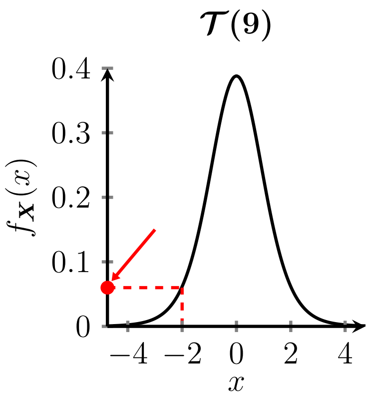
```
NORM.INV(0.8, 1, 2)
```

R:

```
qnorm(p = 0.2, mean = 1, sd = 2,
      lower.tail = FALSE)
```

3 More Complicated Distributions

For more complicated distributions that we will encounter later in the course, the principle is exactly the same.

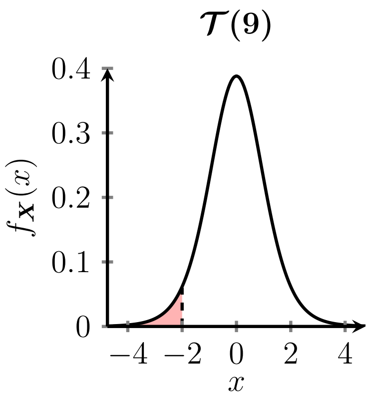


Excel:

`T.DIST(-2, 9, FALSE)`

R:

`dt(x = -2, df = 9)`

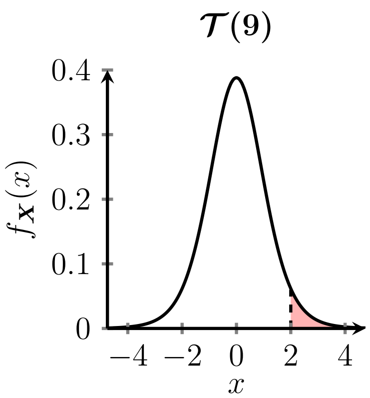


Excel:

`T.DIST(-2, 9, TRUE)`

R:

`pt(q = -2, df = 9, lower.tail = TRUE)`

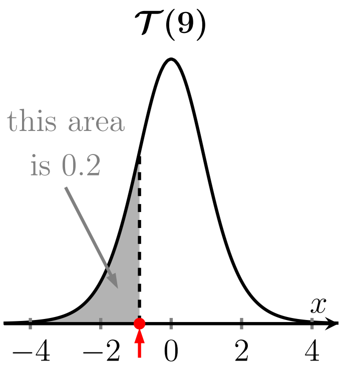


Excel:

`1 - T.DIST(2, 9, TRUE)`

R:

`pt(q = -2, df = 9, lower.tail = FALSE)`

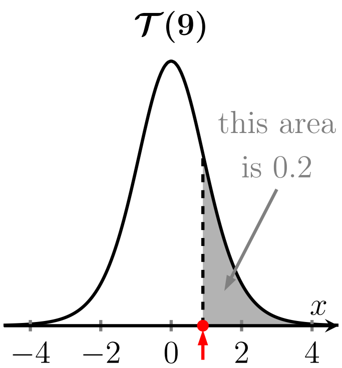


Excel:

`T.INV(0.2, 9)`

R:

`qt(p = 0.2, df = 9, lower.tail = TRUE)`

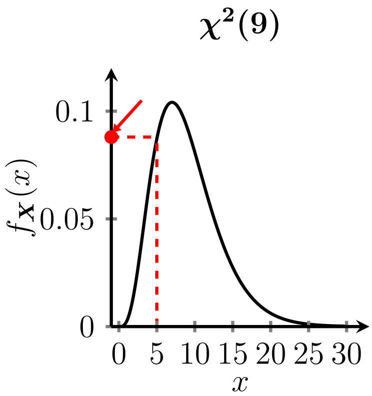


Excel:

`T.INV(0.8, 9)`

R:

`qt(p = 0.2, df = 9, lower.tail = FALSE)`

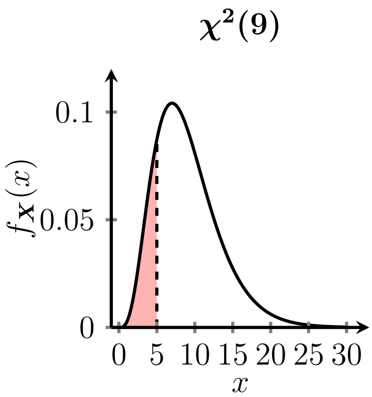


Excel:

`CHISQ.DIST(5, 9, FALSE)`

R:

`dchisq(x = 5, df = 9)`

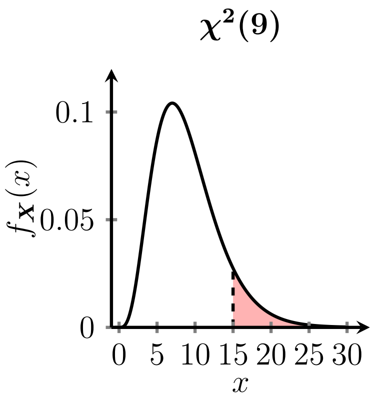


Excel:

`CHISQ.DIST(5, 9, TRUE)`

R:

`pchisq(q = 5, df = 9, lower.tail = TRUE)`

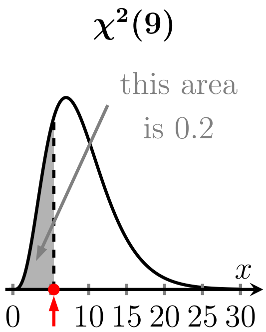


Excel:

`1 - CHISQ.DIST(15, 9, TRUE)`

R:

`pchisq(q = 15, df = 9, lower.tail = FALSE)`

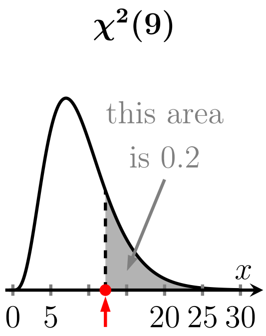


Excel:

`CHISQ.INV(0.2, 9)`

R:

`qchisq(p = 0.2, df = 9, lower.tail = TRUE)`

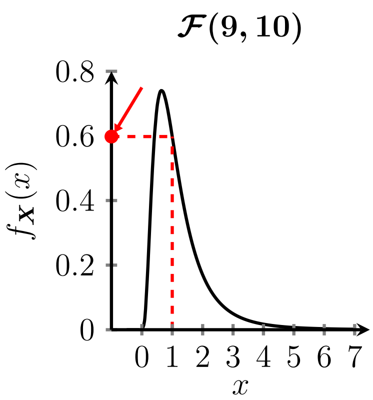


Excel:

```
CHISQ.INV(0.8, 9)
```

R:

```
qchisq(p = 0.2, df = 9, lower.tail = FALSE)
```

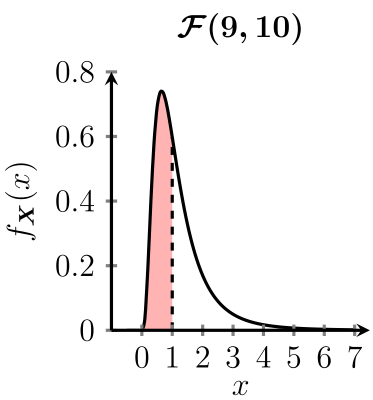


Excel:

```
F.DIST(1, 9, 10, FALSE)
```

R:

```
df(x = 1, df1 = 9, df2 = 10)
```



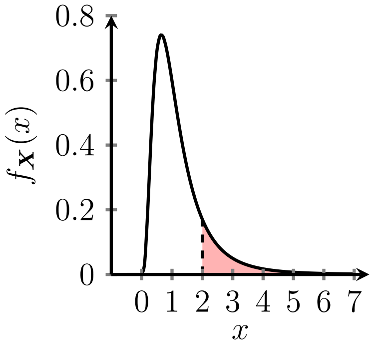
Excel:

```
F.DIST(1, 9, 10, TRUE)
```

R:

```
pf(q = 1, df1 = 9, df2 = 10,
   lower.tail = TRUE)
```

$\mathcal{F}(9, 10)$



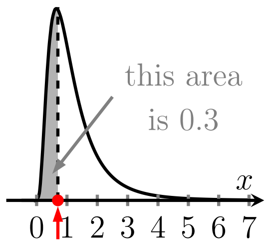
Excel:

```
1 - F.DIST(2, 9, 10, TRUE)
```

R:

```
pf(q = 2, df1 = 9, df2 = 10,
   lower.tail = FALSE)
```

$\mathcal{F}(9, 10)$



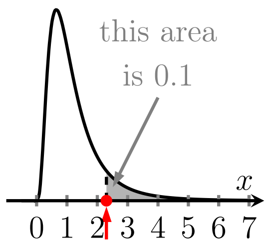
Excel:

```
F.INV(0.3, 9, 10)
```

R:

```
qf(p = 0.3, df1 = 9, df2 = 10, lower.tail = TRUE)
```

$\mathcal{F}(9, 10)$



Excel:

```
F.INV(0.9, 9, 10)
```

R:

```
qf(p = 0.1, df1 = 9, df2 = 10, lower.tail = FALSE)
```