Thursday

1. The nonexistence of a multivariate PIT

For any continuous random variable \( X \) with cdf \( F_X(x) \), if you plug in \( X \) into its own cdf you get a new random variable \( T = F_X(X) \) which takes values in \([0, 1]\). Now (assuming \( F_X^{-1} \) exists), the proof of the following statement is quite simple, and you saw it in class last Tuesday:

\[ T \sim \text{Unif}[0, 1]. \]

\( T \) is called the probability integral transform and \( X \) is said to be “transformed” into the uniform variable \( T \). In some intuitive sense, this is sensible; since cdfs are nondecreasing, if for samples \( X', X'' \) from \( X \) we have \( X' \leq X'' \), we correspondingly have \( U' \leq U'' \), and vice versa. It is an order preserving transformation. (Though this is easy if \( F \) is continuous and strictly monotone, you can use the pseudoinverse definition from class if it isn’t.)

Precisely because of this, it intuitively should not be true that a multivariate cdf has a PID \( T \) with a uniform distribution, because there is no sensible way to order a set of elements in \( \mathbb{R}^n \) if \( n > 1 \). After all, if there were an easy sensible way to define \( F_X^{-1} \), you could convert between a multivariate “world” in \( \mathbb{R}^n \) and a univariate “world” in \( \mathbb{R} \) and then use the ordering on \( \mathbb{R} \) to order \( \mathbb{R}^n \! \).

The exercise is to show explicitly for just one example that a PIT doesn’t exist.

Suppose \( X_1, X_2 \) are iid Unif[0, 1]. Then (per Math 180A) \( F_{X_1, X_2}(x_1, x_2) = q(x_1)q(x_2) \), where \( q(x) = 0 \) if \( x \leq 0 \), \( q(x) = 1 \) if \( x \geq 1 \), and \( q(x) = x \) for \( 0 < x < 1 \).

This just simplifies to \( F_{X_1, X_2}(x_1, x_2) = x_1x_2 \) on the unit square \([0, 1] \times [0, 1]\).

Now show that \( T = F_{X_1, X_2}(X_1, X_2) \) is not Unif[0, 1], though certainly it does take values in that interval. \( \text{Hint:} \) This should be an easy problem. Use some basic properties of the uniform distribution, e.g. its expectation, and remember that \( X_1 \) and \( X_2 \) take values outside the unit square only with probability 0.

(Originally there was a typo here that said \( F_{X_1, X_2}(x_1, x_2) = 1_{[0,1]}(x_1)1_{[0,1]}(x_2) \).

\( \text{Solution.} \) One quick way to show that it’s not uniform is to take its expectation. \( \mathbb{E}T = \mathbb{E}F_{X_1, X_2}(X_1, X_2) \). We don’t need to write it with the \( q \)'s because \((X_1, X_2)\), being standard
uniform, don’t take values outside of the unit square. So it is “certain” (probability 1) that

\[ \mathbb{E} F_{X_1, X_2}(X_1, X_2) = \mathbb{E} X_1 X_2 \]

Then \( \mathbb{E} X_1 X_2 = \mathbb{E} X_1 \mathbb{E} X_2 \), by independence. But this is equal to \( \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} \neq \mathbb{E} U = \frac{1}{2} \), for some \( U \sim \text{Unif}[0, 1] \).

**Programming**

Sometimes the homework asks you to produce several plots, for the purposes of exploration. I will sometimes omit those—compare with the assignment text file to see the omissions.

1. **qqnorm commands with normal data.**

   You’re supposed to create MULTIPLE versions of these plots to get a sense for the random variation but I won’t. Also I won’t plot all of them.

   Here’s `qqnorm(16)`:

   ![Normal Q-Q Plot](image)
Whereas (to get a sense for the effect of asymptotics) here’s \texttt{qqnorm}(16000):

![Normal Q-Q Plot](image)

For small data, the plots are \textit{approximating} a straight line, but they are quite unreliably straight.

2. \textbf{one or two \texttt{qqplot(qnorm((1:1000)/1001),data) commands, why are they the same as \texttt{qqnorm}?}}

I won’t do the plots—they work like \texttt{qqnorm}. The syntax of \texttt{qqplot} takes in the “theoretical distribution” in the first argument (x-axis) and the dataset to compare against in the second (y-axis). The QQplot will plot the quantiles of one dataset against those of the other. In this case the “theoretical distribution” really are the true theoretical quantiles of the normal distribution, so it works like \texttt{qqnorm}, which plots against theoretical normal quantiles.

3. \textbf{Are QQ Plots a reliable check to see if data really is normal?}

Yes, but probably not for small data. There’s a bit of random variation where you don’t know where it really isn’t a straight line, even if it will become one asymptotically. And it might not! Not to mention that distributions in nature are never exactly normal, though they can get close, which may cause the random variation to look worse than it really is for small data.

Note that \texttt{qqnorm} doesn’t check whether data follows a standard normal distribution, but any normal, because the linearity of a \texttt{qqplot} is unaffected by the data \(X_i\) (on either axis) undergoing a affine transformation. That’s a fancy way of saying that a set of new data points on the y axis \(X_i' = a X_i + b\) for some \(b\) and \(a \neq 0\) will still look properly linear against the x axis. (This is really just a statement of function composition. if \(f\) and \(g\) are both lines then \(f \circ g\) is a line.)

4. \texttt{qqplot commands with L1,L2,L3}
Here’s the definition of L1, L2, L3:

L1 <- exp(rnorm(1000, mean=2))
L2 <- exp(rnorm(1000, mean = 3))
L3 <- exp(rnorm(1000, mean=2, sd=2))

These are the exponentials of a normal, so their logs are normal. That’s lognormal data. It’s good to look at their histograms.
Here’s hist(L1):

![Histogram of L1](image1)

Here’s hist(L2):

![Histogram of L2](image2)
And `hist(L3)`:

Notice that the probability mass is pushed more and more to the left. The result is that the right tail (you could say the region to the right of the median) is longer than the left. This is called a right (positive) skew. The more probability mass is on the left, the longer the right tail, and the more positive the skew. L3 is skewed farthest to the right. (Skew can be quantified through a calculation called the skewness—it relates to $E(X^3)$ for some random variable $X$.)

Before we get into the QQplots, it’s helpful to do the following commands:

```r
n = 2  # rows
m = 2  # columns
nf <- layout(matrix(c(n,m), n,m,byrow=T), c(1,1), c(1,1), T)
layout.show(nf)
```

This will let you plot multiple plots at once (in RStudio, not sure about other places.)
Here’s `qqplot(L1,L2)`:

![qqplot(L1,L2)](image1)

Here’s `qqplot(L2,L3)`:

![qqplot(L2,L3)](image2)
And `qqplot(L1,L3)`:

Now for the interpretations.

`qqplot(L1,L2)` is basically straight looking. This is saying that L1 and L2 have nearly the same shape, as you saw in the histogram. There are a few data at the top because the probability that events happen out there is small but nonnegligible (the tails aren't that thin).

`qqplot(L2,L3)` seems to curve up (imagine data filling in the gap between the two rightmost points). What is this saying? Before the median is reached, a lot of data is bunched up together in the left “tail”—but much more in L3 than in L2. In the distance it takes L2 to move from the 0th percentile to the median, L3 has barely moved at all. Then the opposite happens once you pass the median. To get from 50th to the 100th percentile, you must move along L3 a great distance compared to from 0th to 50th. Why does this occur? Because the left “tail” is much shorter in compared to the right tail in L3 than in L2, so there is more density in a smaller space before the median in L3. In other words (as explained above) L3 is more right skewed than L1. (Evidently the effect of increasing standard deviation is quite powerful in this regard.)

`qqplot(L1,L3)` is the same story as `qqplot(L2,L3)`, but more pronounced.

On average, this sort of rightward bulging U shape occurs when the y axis is right skewed.

5. `qqnorm(L1)`. explain what you see and use left and right tails in normal and lognormal in your explanation.
The right tail of L1 is *significantly* longer than its left tail, compared to the comparison between the right and left tails of the normal, which are symmetrical. L1 is very right skewed compared to the normal (which is symmetric and has zero skewness.)

6. **Are QQ plots a reliable check to see if data is not normal?**
   
   Personally, I think so, even with smaller data. If you have small data but you see a shape like these bows, then there’s no way that could’ve just occurred with natural random variation!

7. **qqnorm commands with t data**
   
   Rather than use 1000 data points as assigned, I use 10000 so we can see it clearer. I also add a qqline, which shows the deviation from normality (it’s plotted through the data points corresponding to the 25th and 75th percentiles. I could’ve used a qqline before as well, but it wasn’t as necessary.)
Here’s `qqnorm(rt(10000,4), pch = ".")` with `qqline(rt(10000,4))`:

![Normal Q-Q Plot](image)

And `qqnorm(rt(10000,2), pch = ".")` with the line:

![Normal Q-Q Plot](image)

In the first plot, the shape says that more extreme data is more likely in the sample data than in the theoretical—in other words, the sample has heavier tails. The second plot just shows even heavier tails, because the sample data is even farther from the qqline. That makes sense because it’s *t* against normal, and the lower the degrees of freedom the heavier-tailed it is.

(However I think this kind of explanation only works between symmetric distributions. I won’t actually plot this but suppose you were to plot a $\chi^2_7$ on the x-axis and a $\chi^2_2$ on
the y axis. What you will see is a plot very much like the QQ plot of L3 against L1 above. Imagine a qqline through the 25th and 75th percentile—then we see, to the right of the median, that the data goes high up so doesn’t $\chi^2_2$ have a heavier tail than $\chi^2_7$? In fact this is false; the pdf of the former is strictly beneath the pdf of the latter if you go far enough to the right. What is true is that the skewness decreases with increasing degrees of freedom—the right tail is heavier compared to the left in $\chi^2_2$ than in $\chi^2_7$. )

(So, we see that the skewness, the “asymmetry” is also an independent, important player in the shape of the QQ Plot. I wonder whether, depending on your level on intuition you may be able to decide what curvature is due to skewness versus what is due to heavy tails—I’m not really confident even I could do that. At least we can say that symmetric curving like this problem, behind and in front of the median, means heavy tails. Meanwhile, opposite curvature like with between the L1 and L3’s from before, or the chi squares I didn’t plot, is at least partly due to skew.)

8. **estimate the variance of t with 4 d.f. and with 2 d.f. Does something go wrong in the 2 case? why or why not?**

If we run var(rt(n,4)) with higher and higher values of $n$ we get a closer and closer approximation to the standard deviation for $t_4$—it is 2.007 when I do it for $n = 999999$. Per the formula for the $t$ standard deviation, $4/(4 - 2) = 2$.

What about var(rt(n,2))? If you used various different values for $n$, including very large ones, you would get an essentially random distribution of values. What gives? You see that the formula $df/(df - 2)$ for the variance is not defined and indeed the variance of $t_2$ is infinite. Therefore in the formula for sample variance $1/(n - 1) \sum (X_i - \bar{X})^2$ we’d get the $X_i$’s have a high probability to get far from $\bar{X}$, so that the summand is too big to be controlled even by division by $n - 1$.

9. **qqplot with t data. explain what you see. use the tail thicknesses of $t_2$ and $t_4$ in your explanation.**

```r
qqplot(qt((1:999)/1000,4),rt(1000,2))
qqplot(qt((1:999)/1000,2),rt(1000,2))
```

I won’t plot these; you should see that in the first plot, the theoretical quantiles are the $t_4$ so you’ll get your old heavy tail QQ plot. As of course you should because 2 degrees of freedom is heavier tailed than 4. In the second one, you’re plotting a sample of $t_2$ against its own quantiles so it should be nearly straight.