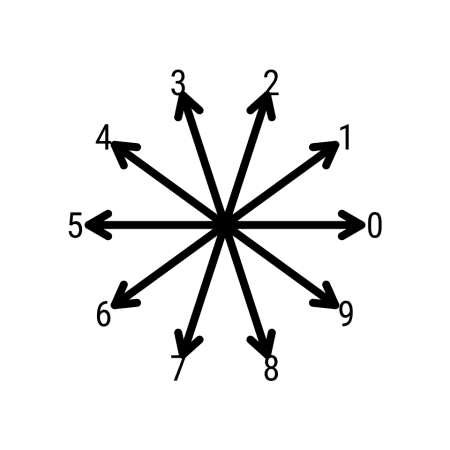
There are so many beautiful “π” arts everywhere, and I wanted to practice ggplot2 by mimicing those arts [further more](https://chichacha.netlify.com/2018/10/13/visual-art-with-pi-using-ggplot2-circlize/). Another pi art caught my eye is random walk of pi digits. Here’s one of examples in [WIRED magazine](https://www.wired.com/2012/06/a-random-walk-with-pi/).

For random walk to work, I’ve assigned direction to “walk” depending on digits 0-9.



I’ve prepared data frame as below, so I can use ggplot2 to plot

## pi\_df contains first 100K digits , each digit is stored in one row.

df\_walk <- pi\_df %>%

mutate(pos = row\_number()-1, ## assign position

dig= as.numeric(dig)) %>%

select(pos, dig) %>%

mutate(angle\_rad = 2\*pi/10\*dig, ## using current digit determine direction to move

angle\_deg = circular::deg(angle\_rad), ## I just like to see number in degree...

move\_x = cos(angle\_rad), ## how much to move in x direction

move\_y = sin(angle\_rad), ## how much to move in y direction

last\_x = replace\_na(lag(move\_x),0), ## position of last x, set origin as 0

last\_y = replace\_na(lag(move\_y),0), ## position of last y, set origin as 0

cumsum\_x = cumsum(move\_x), ## walkig == adding up all steps in x

cumsum\_y = cumsum(move\_y), ## walking == adding up all steps in y

cumsum\_x\_lag = cumsum(last\_x),

cumsum\_y\_lag = cumsum(last\_y))

df\_walk %>% head(n=5) %>% knitr::kable()

| **pos** | **dig** | **angle\_rad** | **angle\_deg** | **move\_x** | **move\_y** | **last\_x** | **last\_y** | **cumsum\_x** | **cumsum\_y** | **cumsum\_x\_lag** | **cumsum\_y\_lag** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 3 | 1.8849556 | 108 | -0.309017 | 0.9510565 | 0.000000 | 0.0000000 | -0.309017 | 0.9510565 | 0.000000 | 0.0000000 |
| 1 | 1 | 0.6283185 | 36 | 0.809017 | 0.5877853 | -0.309017 | 0.9510565 | 0.500000 | 1.5388418 | -0.309017 | 0.9510565 |
| 2 | 4 | 2.5132741 | 144 | -0.809017 | 0.5877853 | 0.809017 | 0.5877853 | -0.309017 | 2.1266270 | 0.500000 | 1.5388418 |
| 3 | 1 | 0.6283185 | 36 | 0.809017 | 0.5877853 | -0.809017 | 0.5877853 | 0.500000 | 2.7144123 | -0.309017 | 2.1266270 |
| 4 | 5 | 3.1415927 | 180 | -1.000000 | 0.0000000 | 0.809017 | 0.5877853 | -0.500000 | 2.7144123 | 0.500000 | 2.7144123 |

Now to visualize the random walk, I’ve used below script to visualize first 1000, first 10000 and first 100000 digits of pi.

n\_steps <- 1000

## Random Walk of Pi

df\_walk %>%

filter(pos < n\_steps) %>%

ggplot(aes(x=cumsum\_x, y=cumsum\_y, color=pos)) +

geom\_segment(size=0.5, aes(xend=cumsum\_x\_lag, yend=cumsum\_y\_lag)) +

geom\_point(size=0.8) +

theme\_void(base\_family="Roboto Condensed") +

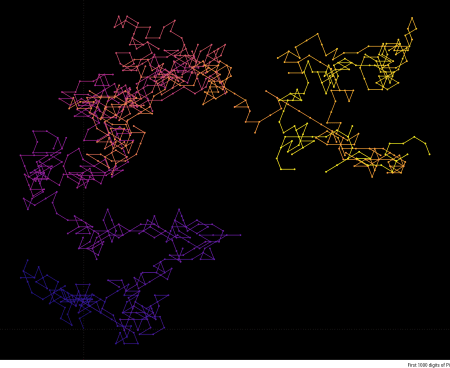
theme(panel.background=element\_rect(fill="#000000")) +

scale\_color\_viridis\_c(option="plasma", guide="none") +

labs(caption=paste("First",n\_steps,"digits of Pi")) +

geom\_hline(yintercept=0, color="#ffffff30", linetype=3) +

geom\_vline(xintercept=0, color="#ffffff30", linetype=3)



#ggsave(str\_c("output/random\_walk\_pi\_",n\_steps,".png"), width=11, height=7)

n\_steps <- 10000

## Random Walk of Pi

df\_walk %>%

filter(pos < n\_steps) %>%

ggplot(aes(x=cumsum\_x, y=cumsum\_y, color=pos)) +

geom\_segment(size=0.1, aes(xend=cumsum\_x\_lag, yend=cumsum\_y\_lag)) +

geom\_point(size=0.01) +

theme\_void(base\_family="Roboto Condensed") +

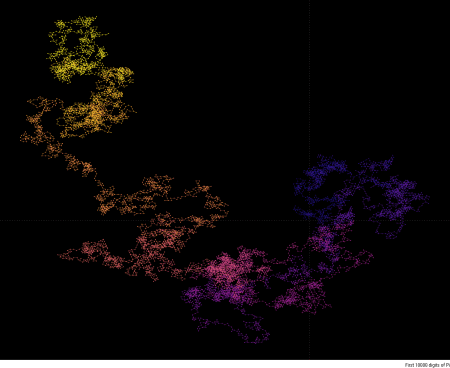
theme(panel.background=element\_rect(fill="#000000")) +

scale\_color\_viridis\_c(option="plasma", guide="none") +

labs(caption=paste("First",n\_steps,"digits of Pi")) +

geom\_hline(yintercept=0, color="#ffffff30", linetype=3) +

geom\_vline(xintercept=0, color="#ffffff30", linetype=3)



n\_steps <- 100000

## Random Walk of Pi

df\_walk %>%

filter(pos < n\_steps) %>%

ggplot(aes(x=cumsum\_x, y=cumsum\_y, color=pos)) +

geom\_segment(size=0.1, aes(xend=cumsum\_x\_lag, yend=cumsum\_y\_lag)) +

geom\_point(size=0.005) +

theme\_void(base\_family="Roboto Condensed") +

theme(panel.background=element\_rect(fill="#000000")) +

scale\_color\_viridis\_c(option="plasma", guide="none") +

labs(caption=paste("First 100000 digits of Pi")) +

geom\_hline(yintercept=0, color="#ffffff30", linetype=3) +

geom\_vline(xintercept=0, color="#ffffff30", linetype=3)

