**Loading up packages**

I’m definitely in love with “magick” right now 

library(tidyverse) ## for pretty much everything...

library(magick) ## I'm now a magick fan!!!

library(scales) ## Handy when it comes to scaling, but I also love show\_col function

library(patchwork) ## put ggplot side by side easily :)

## Let's just set some of my favourite number.

phi <- (1 + sqrt(5)) / 2

golden\_angle <- pi\*(3-sqrt(5))

**Function To Draw Artwork**

Instead of creating data frame, then plot with different parameters, I’ve created function so that I can tweak some of parameters to create different art.

* n : changes number of points to use (number of rows in data frame to plot)
* u : I’m making art by drawing lines by connecting dots, but skipping “u” numbers of dots in between.
* v : Useful to set if you want to create rotating animation
* angle: using golden angle will produce nice phyllotactic spiral, but we can use different angle to produce different art

create\_art <- function(n=1800,u=5,v=0,angle=golden\_angle,colors="#ffffff",...){

my\_colours <- colors ## default is using one colour, white, but I can use vector of colours too.

## Create data frame first using those parameters specified above

df <- tibble(

idx = c(0:(n-1)), ## you can increase the number here to use more lines.

t = seq(0,2\*pi,length.out=n), ## since I used 0 to 1800 above, need to add 1

r = sqrt(idx), ## radius

x = r\*cos(angle\*idx),

y = r\*sin(angle\*idx),

color\_angle = atan2(y=y,x=x) ## get angle between x-axos and the vector from the origin to x,y

)

## In case you specified m2>m then change

v <- ifelse(v%

ggplot(aes(x=x,y=y,color=color\_angle)) +

geom\_path(data= . %>% filter(idx%%u==v), ## only use partial data to connect the dots

lineend="round", linejoin="mitre", linemitre=3,

aes(size=idx, alpha=idx)) +

coord\_fixed() +

theme\_void() +

scale\_alpha\_continuous(guide="none", range=c(0,1), trans="sqrt")+

scale\_size\_continuous(guide="none", range=c(10,0), trans="sqrt") +

scale\_color\_gradientn(guide="none",

colors=my\_colours) +

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

expand\_limits(x=c(-max\_r,max\_r),y=c(-max\_r,max\_r))

my\_art + annotate(geom="text", x=Inf,y=-Inf,

label=str\_glue('n: {n} | u: {u} | v: {v} | angle: {round(angle,3)} radian'),

family="Roboto Condensed", color="#ffffffae",

hjust=1,vjust=-1)

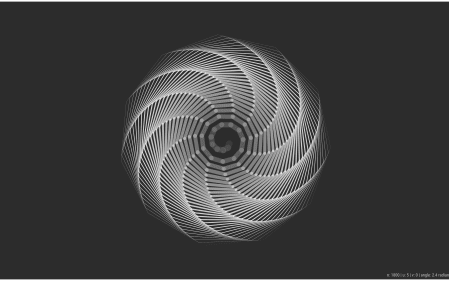
}

## Using all default value to plot!

create\_art() +

## using below just to make canvas wider to fit blog post nicer?

geom\_blank(data=data.frame(), aes(x=c(-phi\*50,phi\*50),y=c(-50,50), color=c(0,0)))



**Variation of Art by Tweaking Some Parameters**

Arranging plot side by side is very easy with package [patchwork](https://github.com/thomasp85/patchwork)!

## changing up the angle

create\_art(n=360,angle=sqrt(2)) +

create\_art(n=360,angle=sqrt(3)) +

create\_art(n=360,angle=sqrt(5)) +

create\_art(n=360,angle=pi/7) +

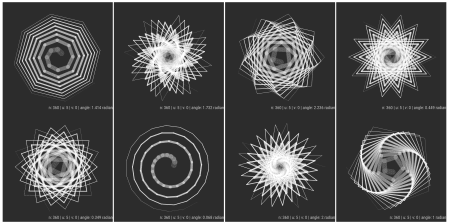
create\_art(n=360,angle=pi/9) +

create\_art(n=360,angle=pi/46) +

create\_art(n=360,angle=2) +

create\_art(n=360,angle=1) +

plot\_layout(ncol = 4)



## changing up the u value - number of points to skip in order to connect the points

create\_art(n=360,u=5) +

create\_art(n=360,u=11) +

create\_art(n=360,u=13) +

create\_art(n=360,u=6) +

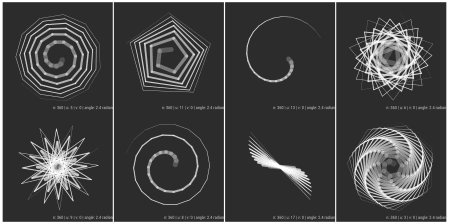
create\_art(n=360,u=9) +

create\_art(n=360,u=8) +

create\_art(n=360,u=17) +

create\_art(n=360,u=3) +

plot\_layout(ncol = 4)



**Creating Animation GIF!!!**

I’ve just discovered that I can utizile image\_graph function in magick! I couldn’t figure out if there’s way to plot 2 gifs next to each other, so that both are showing side by side. I also coulnd’t figure out how to plot multiple images using map function but not displaying [[1]] [[2]] [[3]] in output…

## Create Rotating Animation

#imgs\_1 <- image\_graph(width=600, height=600)

#params <- tibble(n=1800,u=44,v=seq(0,u-1,by=2),angle=golden\_angle)

#params %>% pwalk(.,~create\_art(.) %>% print())

#dev.off()

## Smoother animation

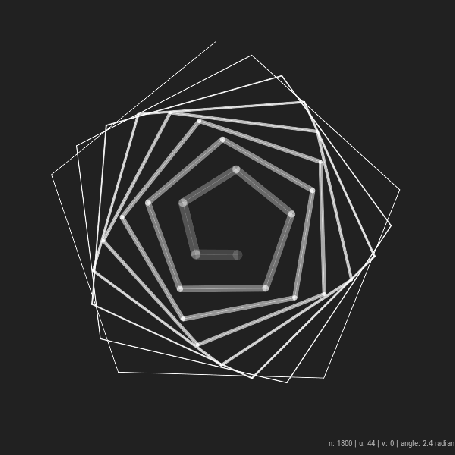
#imgs\_2 <- image\_graph(width=600, height=600)

#params <- tibble(n=1800,u=22,v=seq(0,u-1,by=1),angle=-golden\_angle)

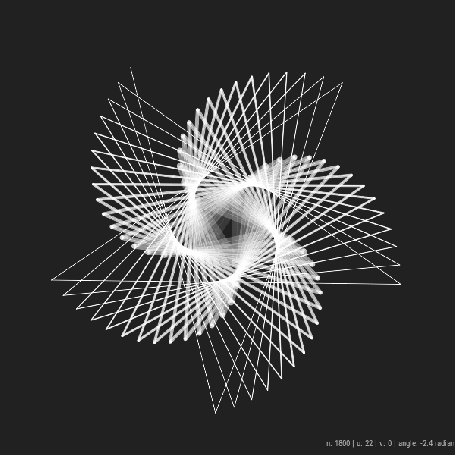
#params %>% pmap(.,create\_art)

#dev.off()

image\_animate(imgs\_1,fps=10)



image\_animate(imgs\_2,fps=10)



**Plotting Image Side by Side with image\_append**

While I wasn’t sure how to put 2 animation gifs next to each other, I was able to put images side by side using image\_append function!

## I'm commenting out below, because I couldn't figure out how to run it silently...

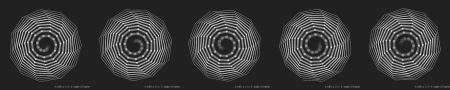
#imgs\_3 <- image\_graph(width=400, height=400)

#params <- tibble(n=640,u=5,v=seq(0,u-1,by=1),angle=golden\_angle)

#params %>% pmap(.,create\_art)

#dev.off()

image\_join(imgs\_3) %>% image\_append()



#imgs\_4 <- image\_graph(width=400, height=400)

#params <- tibble(n=640,u=10,v=seq(0,u-1,by=2),angle=c(sqrt(2),sqrt(3),sqrt(4),sqrt(5),sqrt(6)))

#params %>% pmap(.,create\_art)

#dev.off()

image\_join(imgs\_4) %>% image\_append()

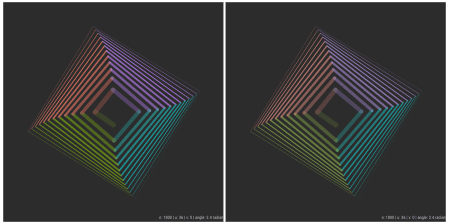


**Generating Art with Different Colour Palette!**

## Finally You can also add colours to it.

create\_art(u=36,colors=hue\_pal()(4)) +

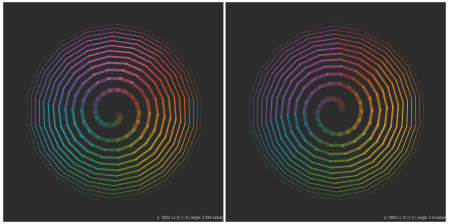
create\_art(u=36,colors=hue\_pal(c=60)(4))



## Using different palette

create\_art(u=8,colors=ggthemes::tableau\_color\_pal("Hue Circle")(19), angle=2\*pi-golden\_angle) +

create\_art(u=8,colors=ggthemes::tableau\_color\_pal("Classic Cyclic")(13))



## One Last one!

create\_art(n=3600,u=18, colors=ggthemes::canva\_pal("Fun and cheerful")(4)) +

create\_art(n=3600,u=18, colors=ggthemes::canva\_pal()(4), angle=-golden\_angle)

