Diffusion Limited Aggregation

Suraj Shourie

July 21, 2019

OBJECTIVE:

Introduce new particles at the boundary of M*M square matrix and initiate a random walk until the particle encounters another particle, thus building an aggregated structure of total N particles.

APPROACH:

1. My first approach was to create a brute force random walker which moves with an equal probability along the four directions. This worked well for small sample sizes but did not scale well to an increase in number of particles.
2. To reduce the run time and taking inspiration from mean reversion present in auto-regressive models, I created a bias in the random walk for center seeking moves. This drastically reduced the run-time and I could run the models for a larger N.

- Unexpectedly, a star like pattern emerged, as seen in the image below, especially for large number of particles. I believe that this pattern occurred because the algorithm used introduced a relatively high bias for center seeking moves. Thus the prevalence of particles along the diagonal and the two axes.

Figure 2: M =200, N=5,000 with bias towards the origin

3. The main problem affecting the time required to generate the DLA is that as the dimension of the bounding square (M) grows, the particles need to ‘walk’ a lot before they meet the aggregate. So to reduce the time taken for creating the DLA, I introduced the particles closer to the aggregate along the minimum bounding circle.

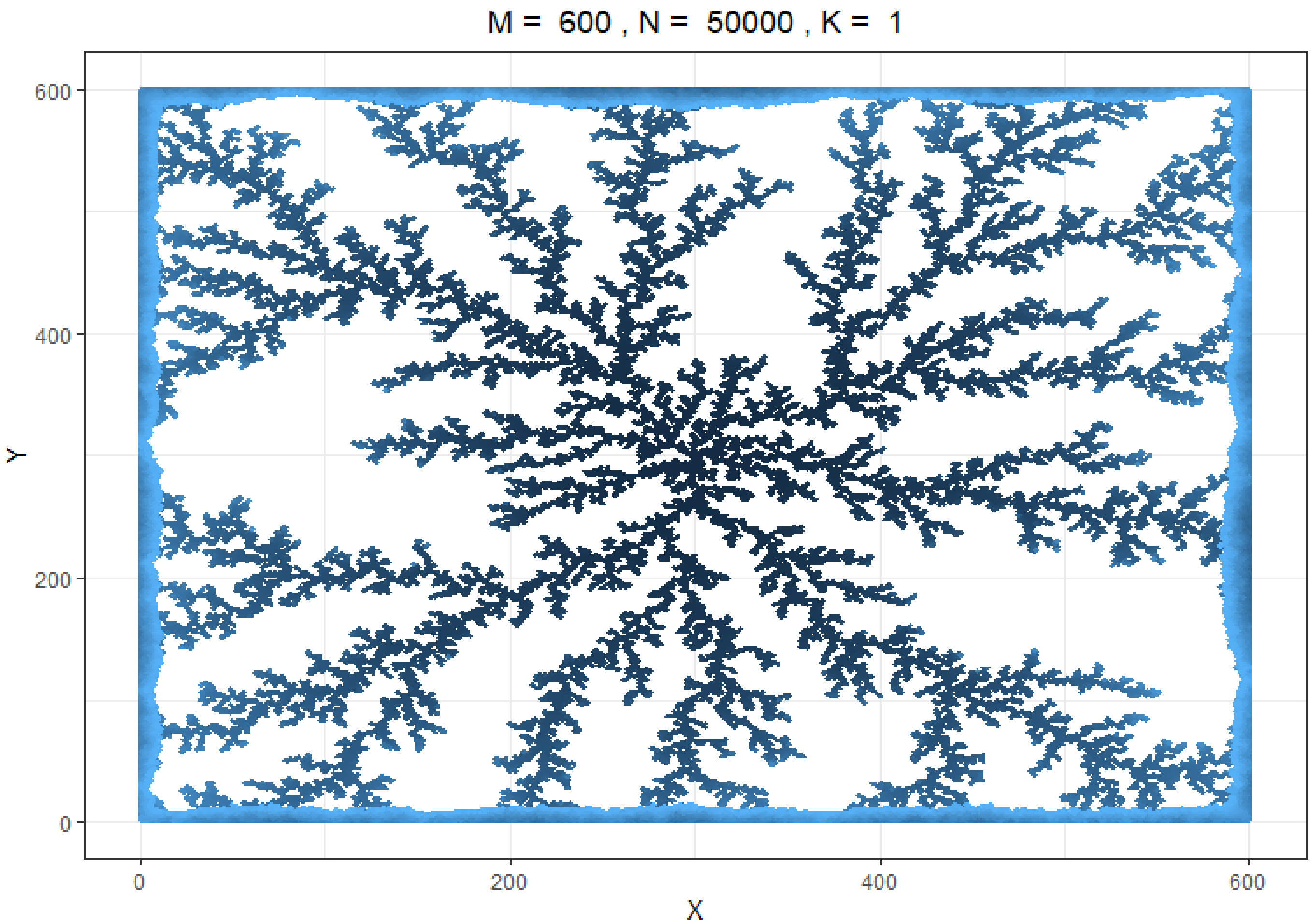


Figure 3: M =500, N=50,000

ESTIMATING STICKINESS:

1. Implemented stickiness as follows: Aggregation occurs if a new particle under Brownian motion comes in contact with another particle and if a random number falls under the given stickiness factor (K).

- Looking at the faceted plots below, it seems that for lower values of K the aggregations get denser, more spherical, and their surface area (or perimeter for 2D aggregations) is lower.
- I calculated Surface Area (hereafter denoted as SA) based on number of neighbors surrounding each cell using a vectorized approach.

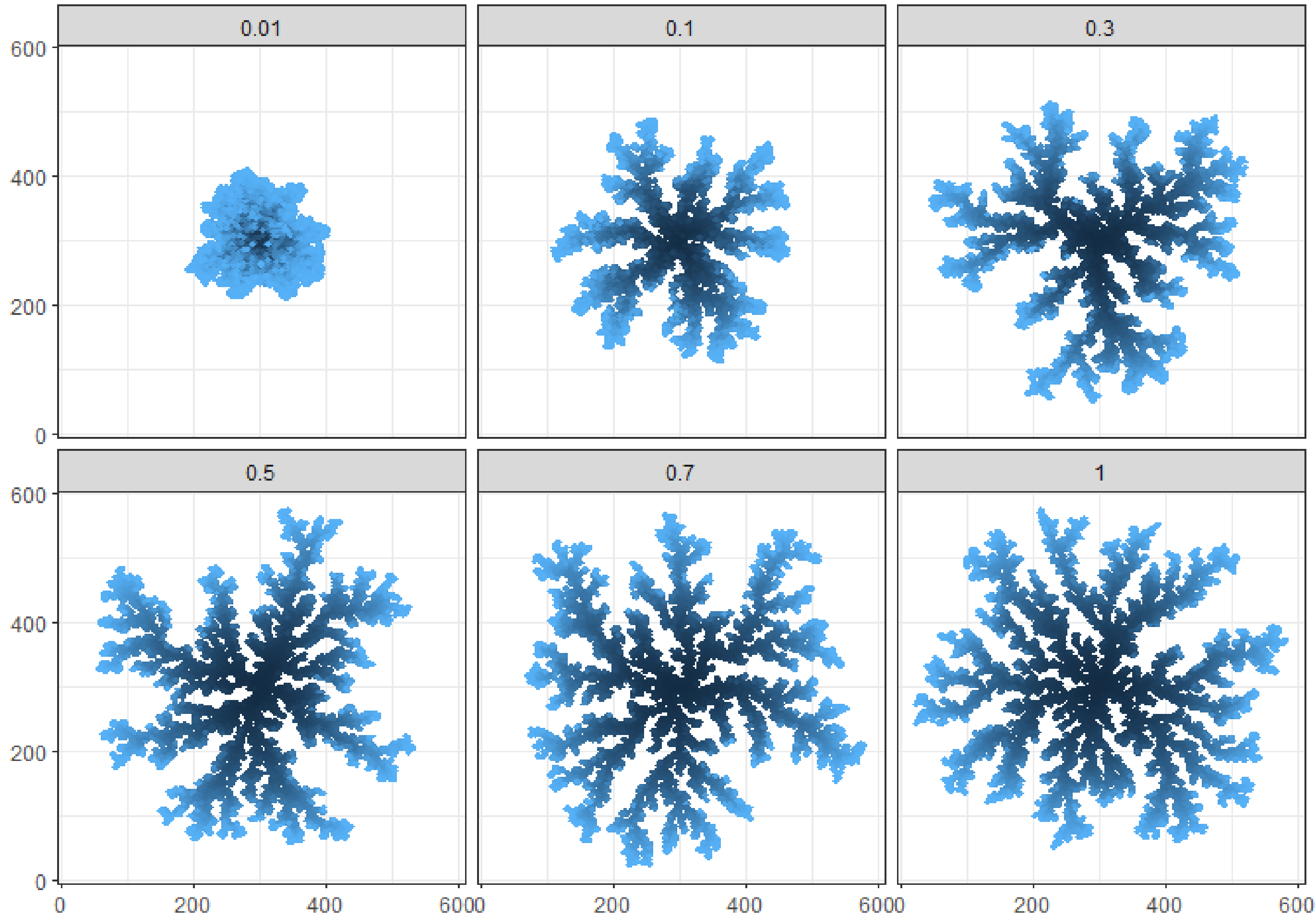


Figure 4: M=200, N=3000 for different values of K

2. There are three independent variables M (size of matrix), N (number of particles), and SA (Surface Area) with K (Stickiness Parameter) as the dependent variable.

- Intuitively, if M is large enough, the size of matrix (or how far the boundary from which the particles originate is from the central particle) shouldn't make a difference in the shape and surface area of the aggregates.
- To confirm this, I plotted Surface Area vs K for different values of M (100,300, 500) for the same number of particles (N = 1000) and discarded M as a dependent variable from the model.

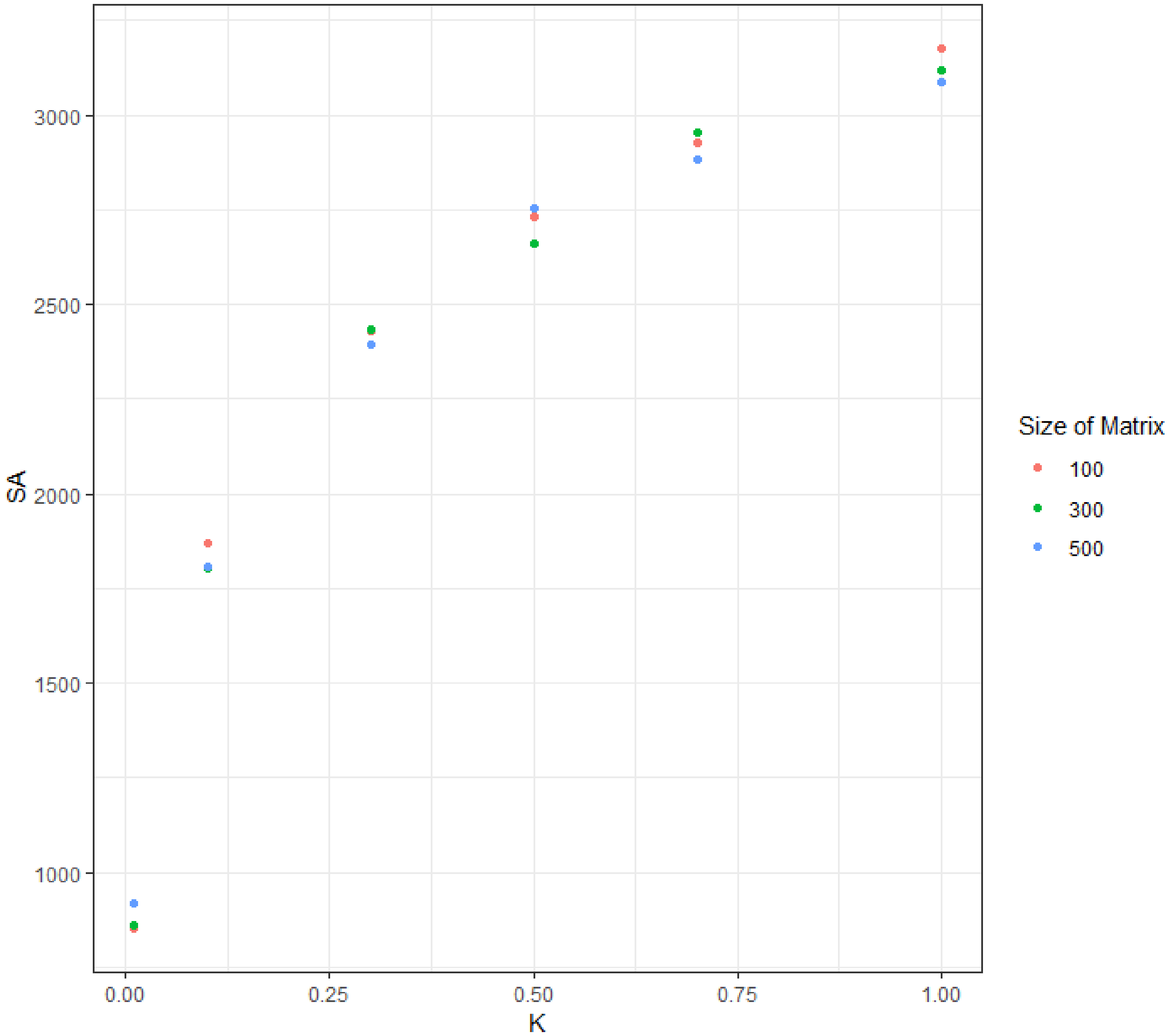
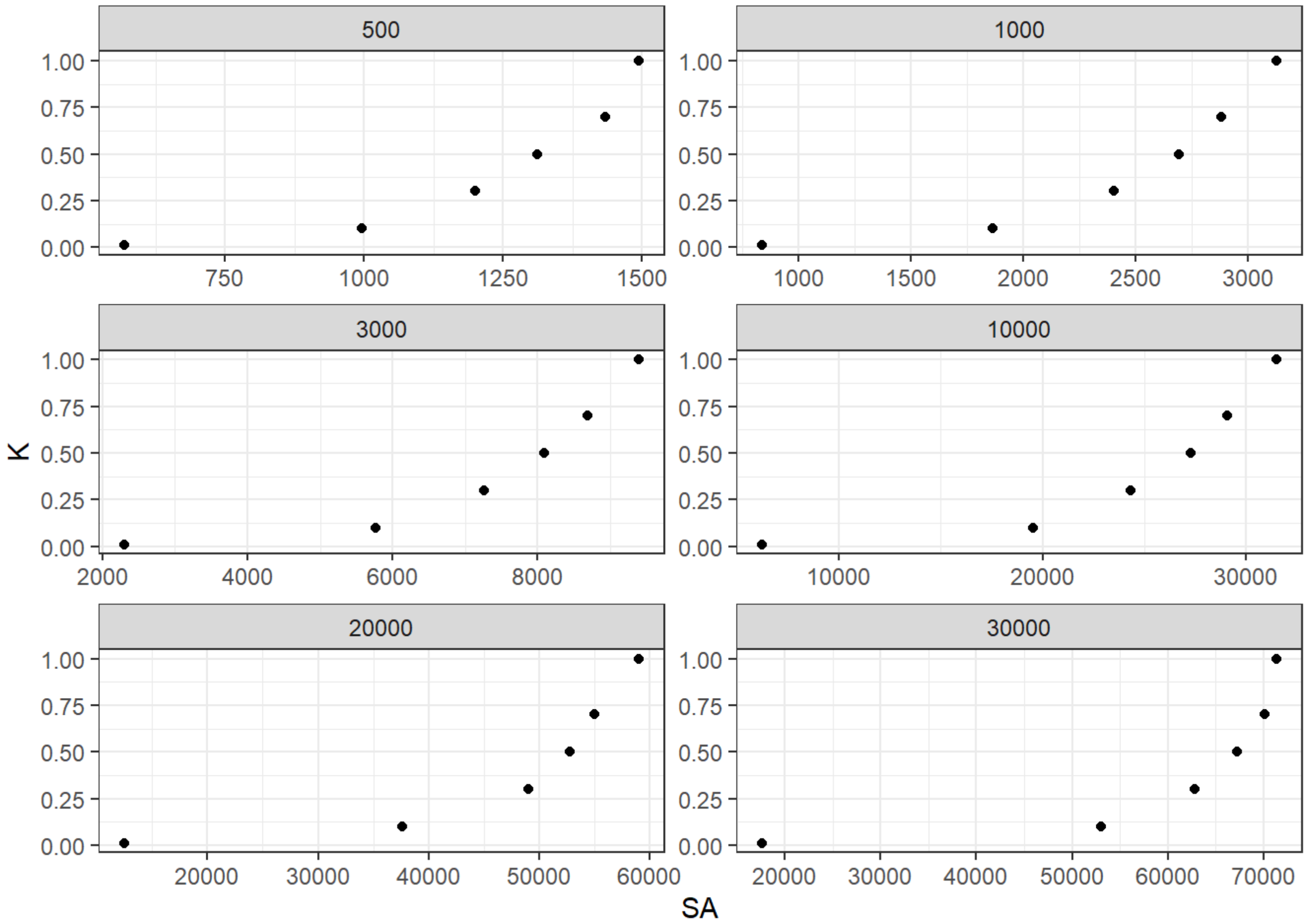


Figure 4: Surface Area vs Stickiness for 1,000 particles and different values of M

3. Next, I tried to see what relation exists between K and the dependent variables of N and SA

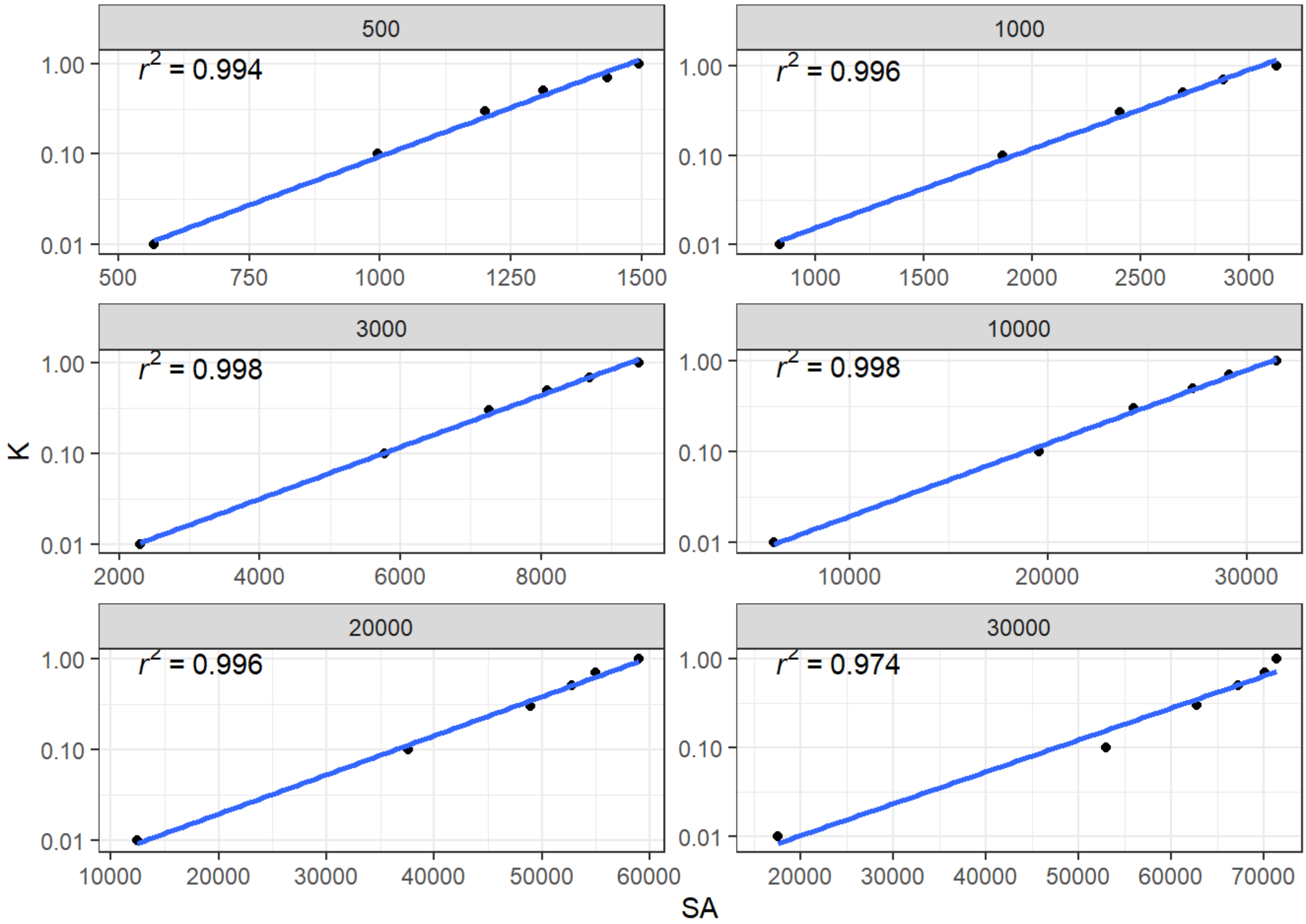
- Firstly, checking if a direct relationship exists between SA and K for different N.

```
#plotted SA vs Stickiness for different N
df2  %>%  ggplot(aes(x=SA, y=K )) + geom_point() + theme_bw() + labs(shape ="#Particles") + f
acet_wrap(vars(N), scales ="free", ncol=2)
```



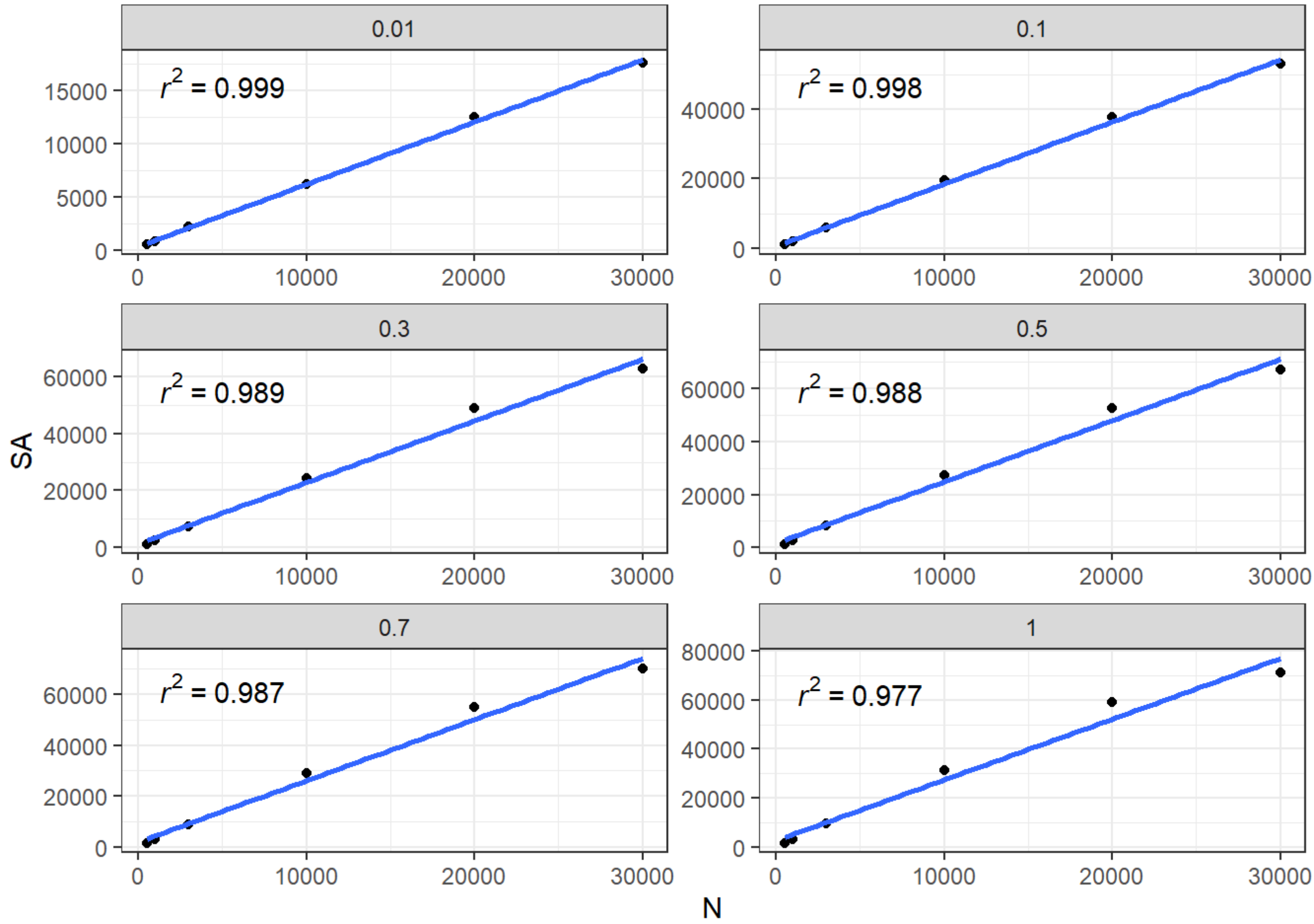
- It seems that a relationship exists between Surface Area and the stickiness parameter. Taking log scale for the variable K in the vertical axis, and checking the R^2 of the best fit lines.

```
#Plotted SA vs Log Stickiness for different N
df2  %>%  ggplot(aes(x=SA, y=K )) + geom_point() + theme_bw() + scale_y_log10() +
  labs(shape ="#Particles") + facet_wrap(vars(N), scales ="free", ncol=2) + stat_smooth(metho
d = "lm", se =FALSE) +
   stat_smooth_func(geom="text",method="lm",hjust=0,parse=TRUE)
```



4. Now, checking for a relationship between SA and N for different K.

```
df2  %>%  ggplot(aes(x=N, y=SA )) + geom_point() + theme_bw()  +labs(shape ="Stickiness") +
  facet_wrap(vars(K), scales ="free", ncol=2) + stat_smooth(method = "lm", se =FALSE) +
  stat_smooth_func(geom="text",method="lm",hjust=0,parse=TRUE)
```



5. Based on the above charts, it seems that SA is linearly proportional with N and has logarithmic relation with K. I tried to model log(K) as a function of SA/N using linear regression. Checking the ANOVA table and the R-squared for regression model:

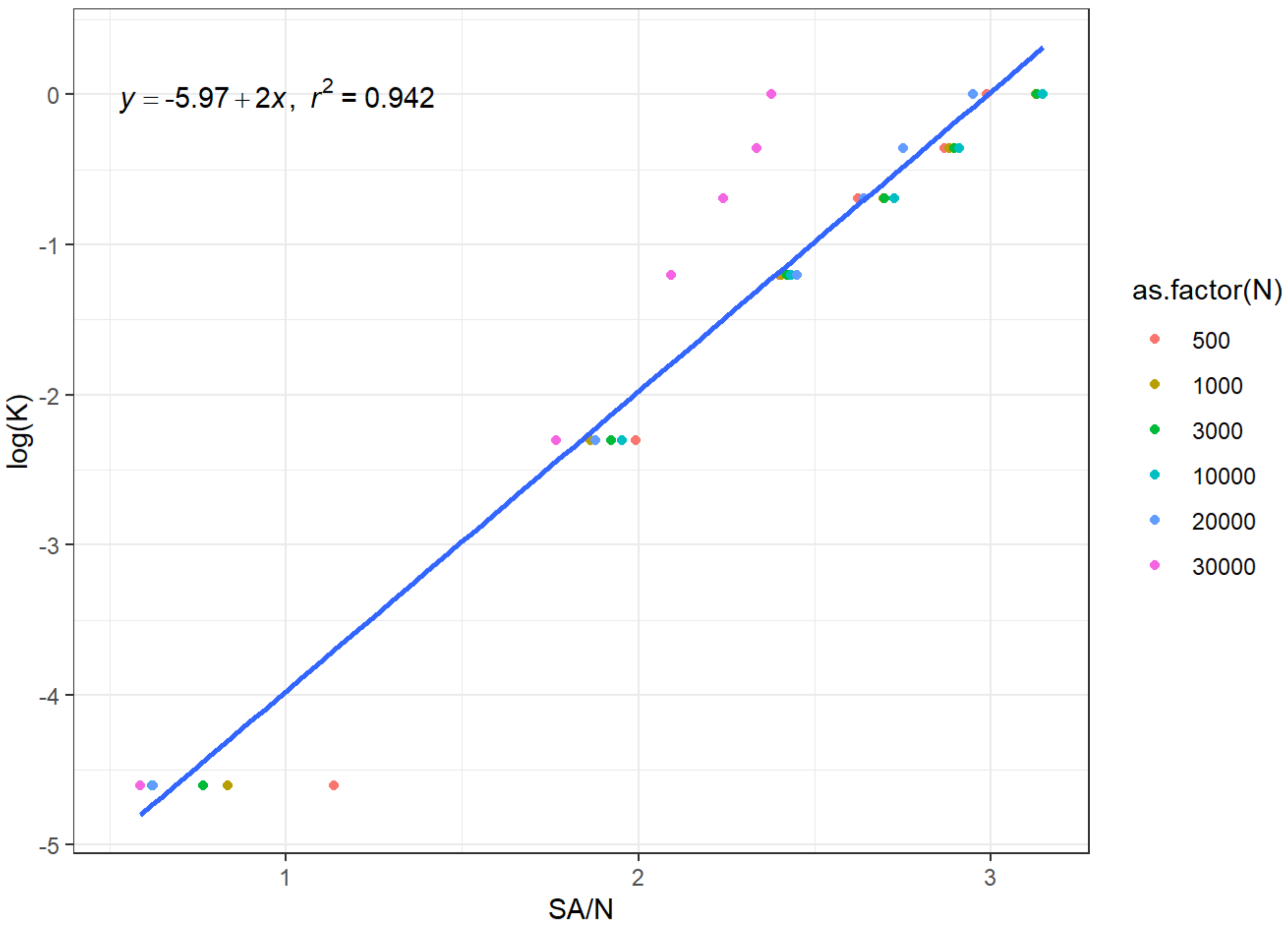
```
print(anova(lm( log(K) ~ `SurfaceArea/N`, df2)))
```

```
## Analysis of Variance Table
##
## Response: log(K)
##                 Df Sum Sq Mean Sq F value    Pr(>F)
## `SurfaceArea/N`  1 82.415  82.415  554.77 < 2.2e-16 ***
## Residuals       34  5.051   0.149
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## [1] "R-Squared for the model is:  0.942"
```

The equation for the model is then:

```
df2 %>%  ggplot(aes(y=log(K), x=SA/N )) + geom_point(aes(col = as.factor(N))) +theme_bw() +
  stat_smooth(method = "lm", se =FALSE) + stat_smooth_func2(geom="text",method="lm",hjust=0,p
arse=TRUE)
```


$y = -5.97 + 2x,\ r^2 = 0.942$
log(K)
SA/N
0
-1
-2
-3
-4
-5
1
2
3
as.factor(N)
500
1000
3000
10000
20000
30000