Geographical Ecology

Z620: Quantitative Biodiversity, Indiana University

February 13, 2015

OVERVIEW

In this exercise, we will add a spatial context to the concepts of alpha (α) and beta (β) by considering how diversity can vary across distance and area. We will re-emphasize the importance of the site-by-species matrix and also emphasize core concepts like aggregation and spatial autocorrelation. We will discuss spatially implicit and spatially explicit patterns and stress the importance of spatial scale. Throughout this exercise, we will introduce you to new R functions and packages, and cover new rudimentary programming skills.

After completing this exercise you will know how to:

1. Quantify, compare, and interpret the distribution of abundance and richness across space.
2. Examine whether spatial patterns are highly dependent on scale.
3. Examine how geographic distance influences similarity of communities and environments.
4. Use control structures such as `loops` to control how R operates on variables.

1.) SETUP

Retrieve and Set Your Working Directory

```r
rm(list=ls())
getwd()
setwd("~/GitHub/QuantitativeBiodiversity/Assignments/GeographicalEcology")
```

Load Packages

As in previous exercises, we will use the `vegan` package. We will also use the `BiodiversityR` package, which contains additional diversity estimators and related functions.

```r
library("vegan")
library("BiodiversityR")
```

We will also use package developed in R for geographical information systems (GIS). These packages will allow us to generate maps, project diversity data onto environmental layers, and conduct geographical analyses.

```r
#install.packages("RgoogleMaps")
library(RgoogleMaps) # overlays of data onto a Google map.
```

Question 1: Identify two ways that the library() function differs from require().

> ***Answer 1***:

Load Data

We will analyze environmental and bacterial community data from a survey of shallow ponds found east of Bloomington, Indiana. These ponds are scattered throughout Brown County State Park, Yellowood State Forest, and Hoosier National Forest. In the 1940s, Maury Reeves of the Indiana Department of Natural Resources began constructing refuge ponds for wildlife. In the summer of 2013, we visited approximately 50 of these ponds and recorded their geographic locations using a GPS unit; 'GPS' is the acronym for Global Positioning System. The ponds were sampled for water chemistry, physical properties, and bacterial community composition.

In this exercise will work with the site-by-species matrix for this dataset as well as accounting for the environmental differences and geography.

Let's begin by loading an image of one of these ponds.

```r
#![pond](./data/pond.JPG)
```

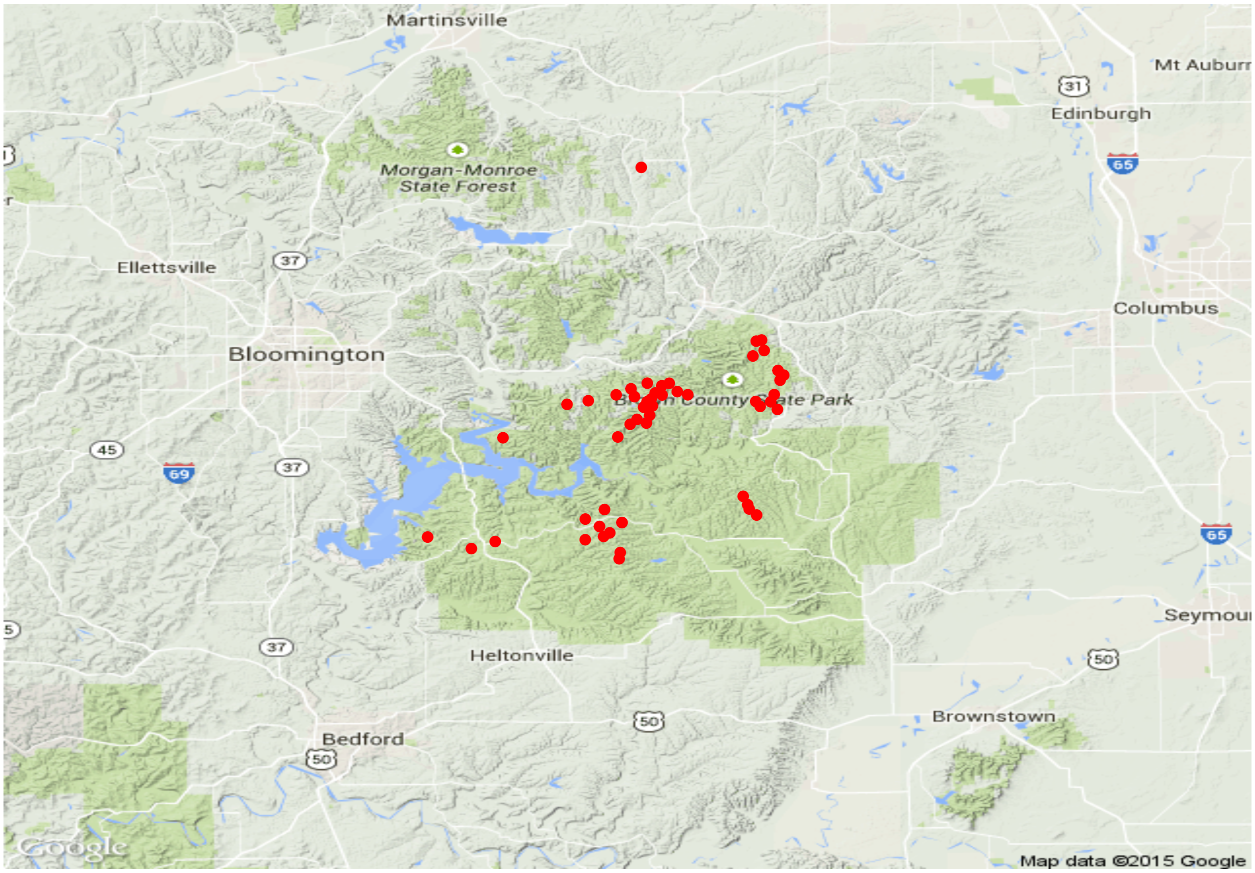
Question 2: The image of this pond should provoke thoughts of the processes, mechanisms, and dynamics driving biodiversity of various communities and populations. As a thought exercise, speculate on what you would expect to find in terms of the biodiversity of this pond, and why.

> ***Answer 2***:

2.) MAKE A MAP

In this section of the exercise, we are going to use R to generate a map of the refuge ponds using **Google Maps**. First, we will retrieve a map of Indiana using the `GetMap` function in the package `RgoogleMaps`. Then, we will generate a map in RStudio that is centered on Brown County, Indiana (39.1 degrees latitude, -86.3 degrees longitude).

```r
newmap <- GetMap(center = c(39.1,-86.3), zoom = 10, destfile = "Brown_Co_map.png", maptype="terrain")
PlotOnStaticMap(newmap, zoom = 10, cex = 2, col='blue') # Plot map in RStudio

Ponds <- read.table(file="BrownCoData/20130801_PondDataMod.csv", head=TRUE, sep=",")
lats <- as.numeric(Ponds[, 3]) # latitudes (north and south)
lons <- as.numeric(Ponds[, 4]) # longitudes (east and west)
PlotOnStaticMap(newmap, lats, lons, cex=1, pch=20, col='red', add = TRUE)
```



```
# Use the help() function to learn about GetMap and PlotOnStaticMap.
# Adjust `zoom`, `cex`, and `pch` to see how those "keyword arguments" influence the plot.
```

Question 3: Based on the map you generated, what do you notice about the spatial distribution of the refuge ponds?

> ***Answer 3***:

3.) PRIMARY CONCEPTS

Concept 1: Spatial aggregation

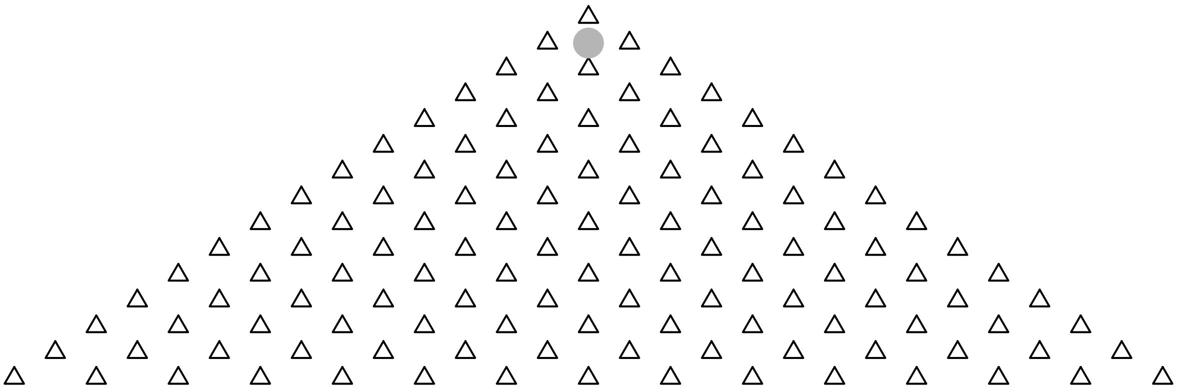
Spatial aggregation refers to the tendency for things to cluster in space, i.e. to be under-dispersed. There are many ecological reasons to expect individuals to be aggregated. Resources and environmental conditions are often aggregated and patchy. Individuals closely associate for protection and reproductive opportunities, and respond to sociobiological pressures to form groups, family units, and colonies.

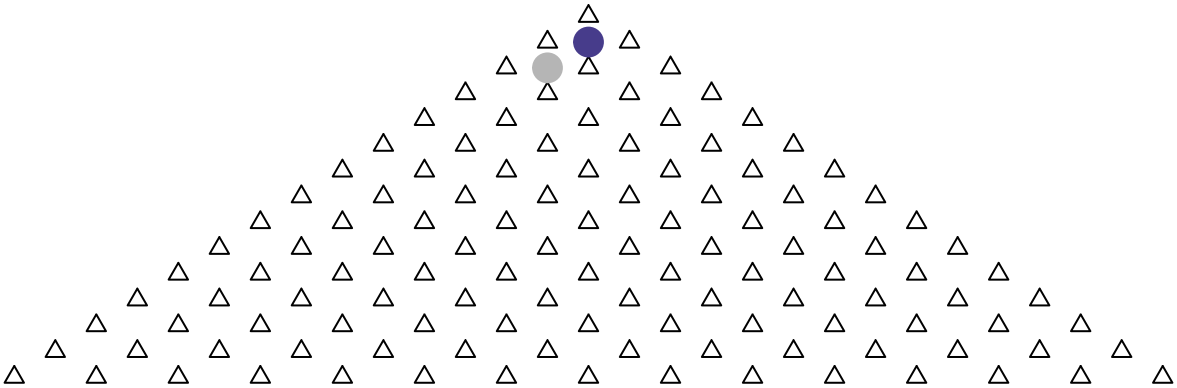
However, stochastic models also predict aggregation based on various scenarios of random movement and placement. Consider a scenario where individuals enter an environment from a central upstream location and then move downward, veering randomly left and right until they arrive at at downstream edge. Run the following code to see a demonstration of this scenario.

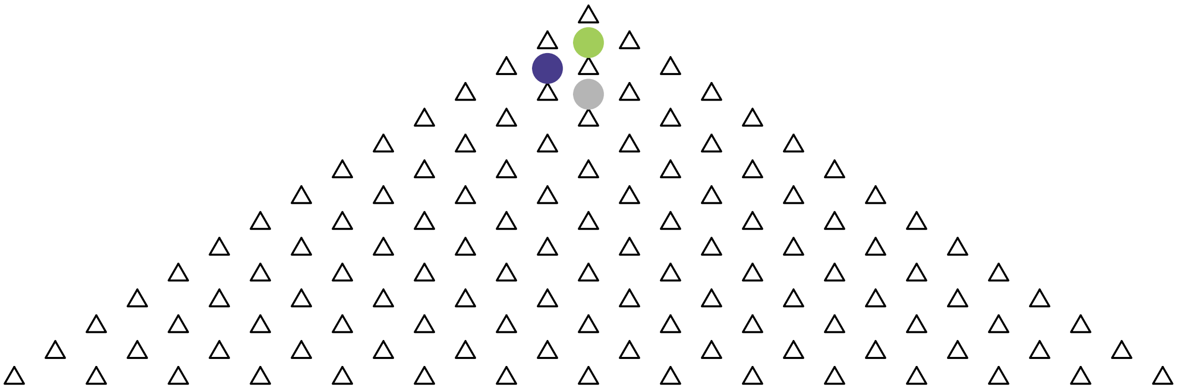
```
#install.packages("animation") # install if necessary
library(animation)
```

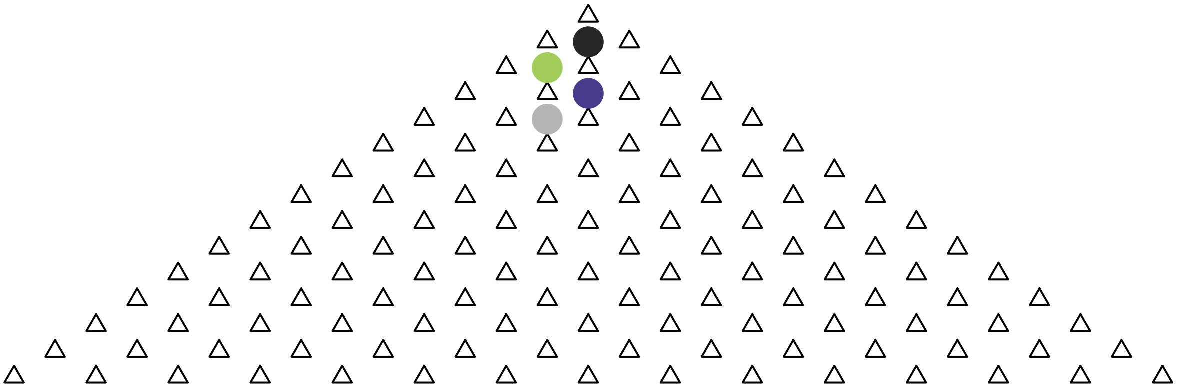
```r
ani.options(interval = 0.1, nmax = 200)
quincunx()
```

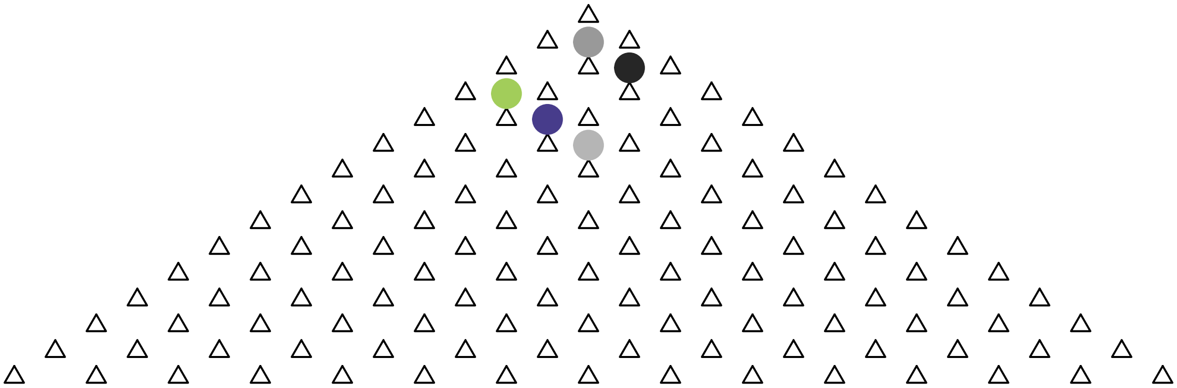
```
## Warning in quincunx(): It's strongly recommended that ani.options(nmax =
## balls + layers -2)
```

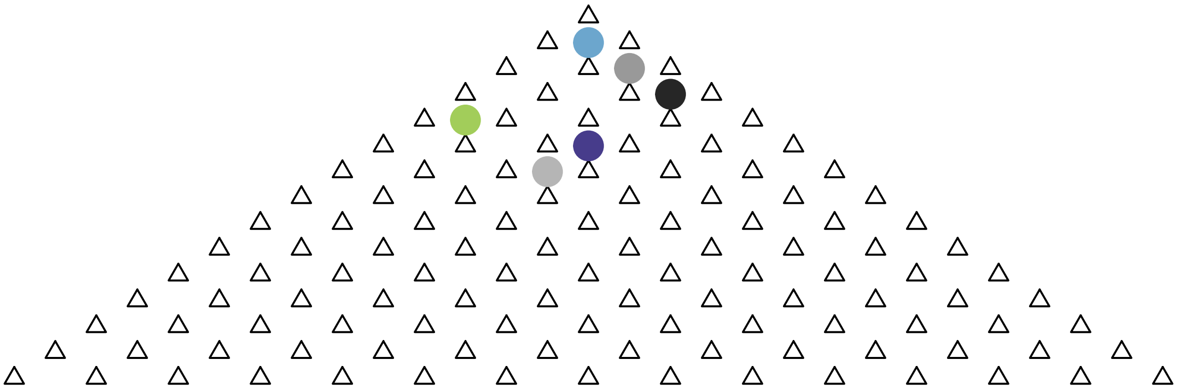


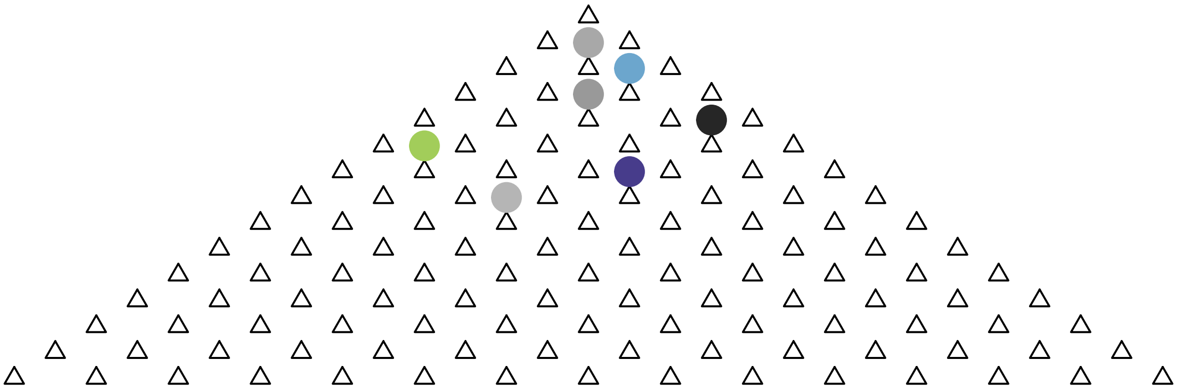


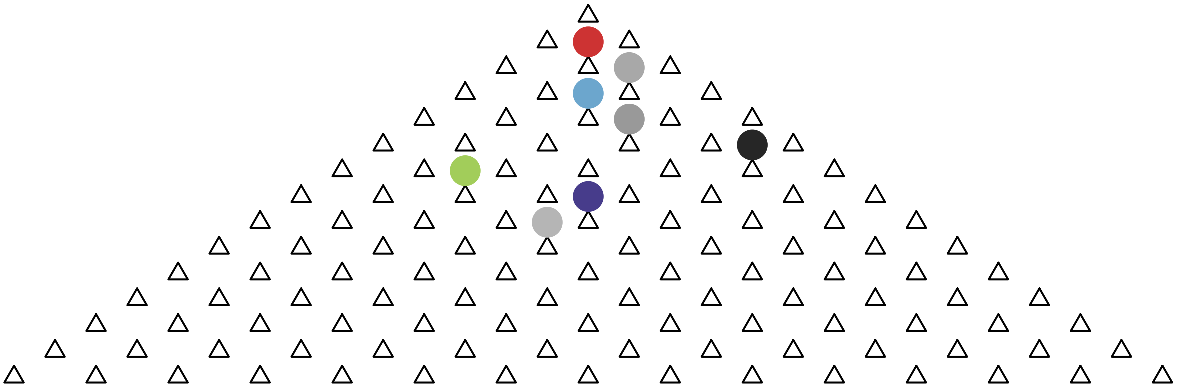


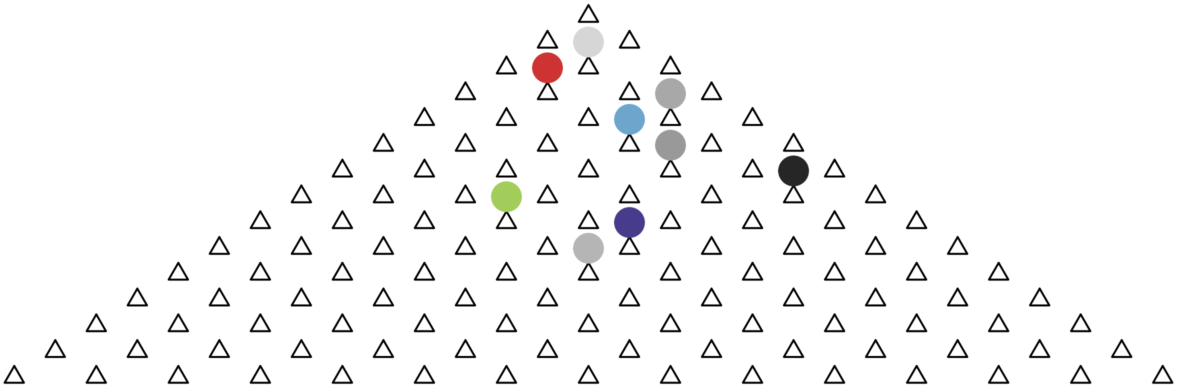


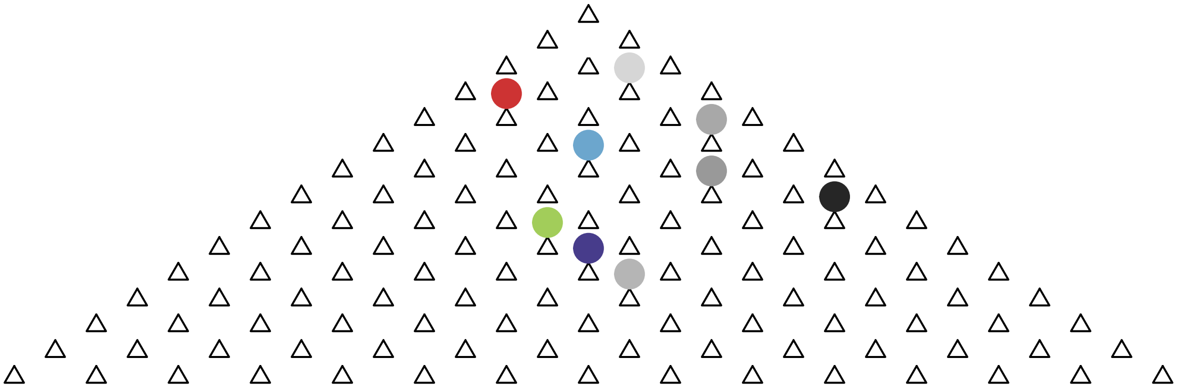


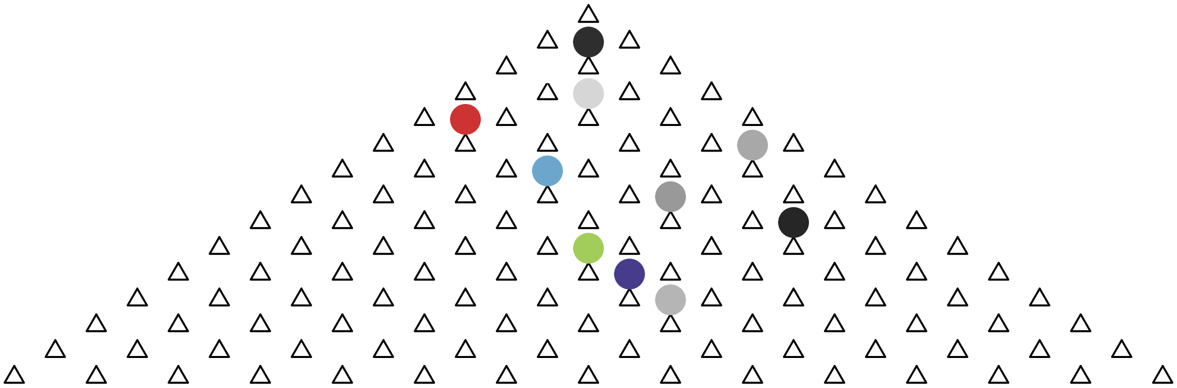


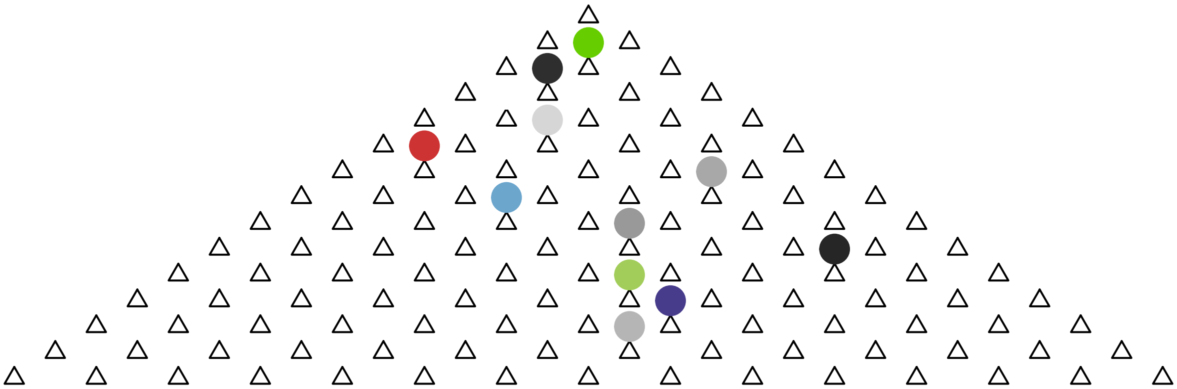


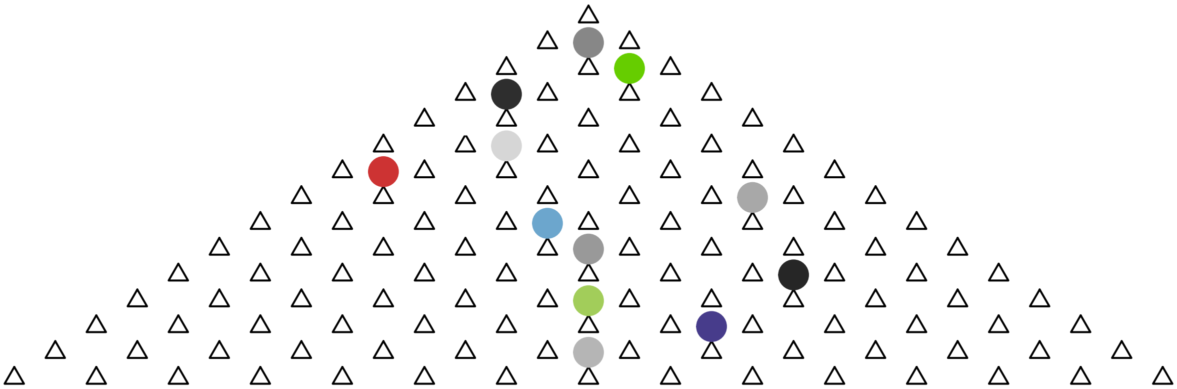


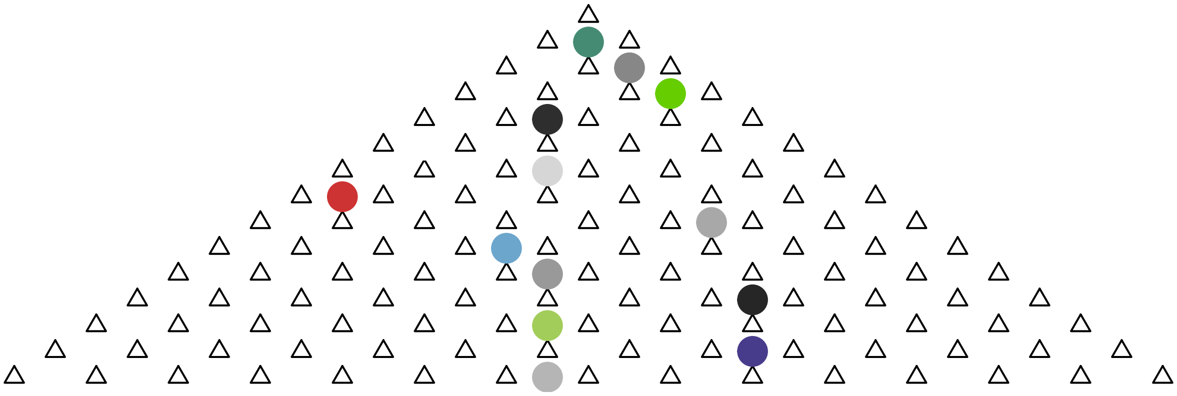


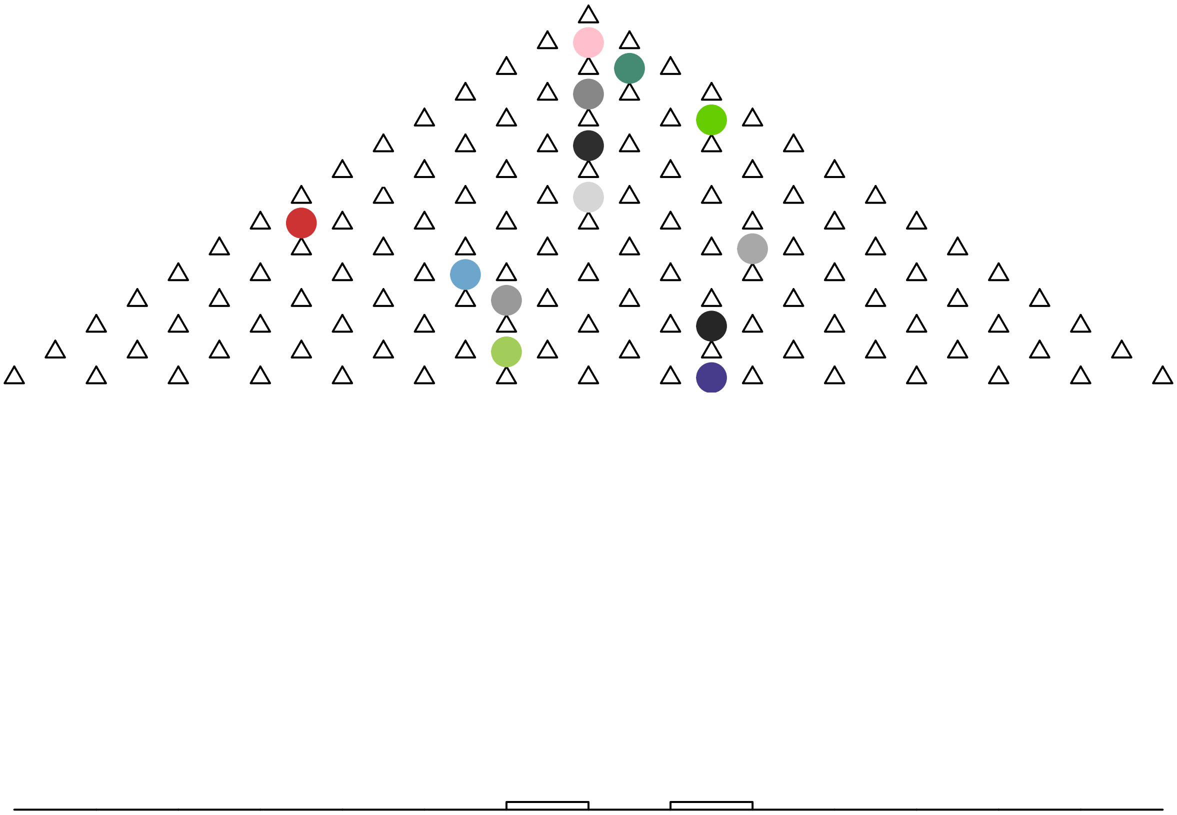


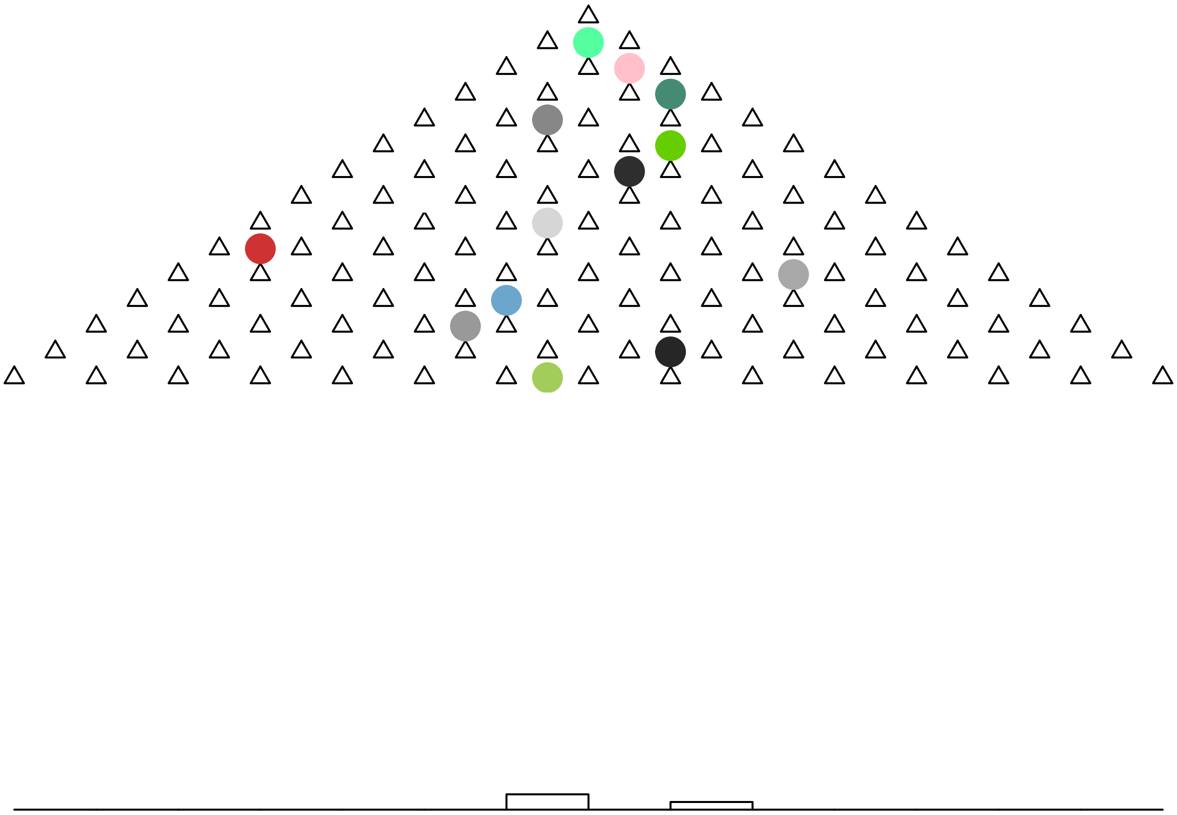


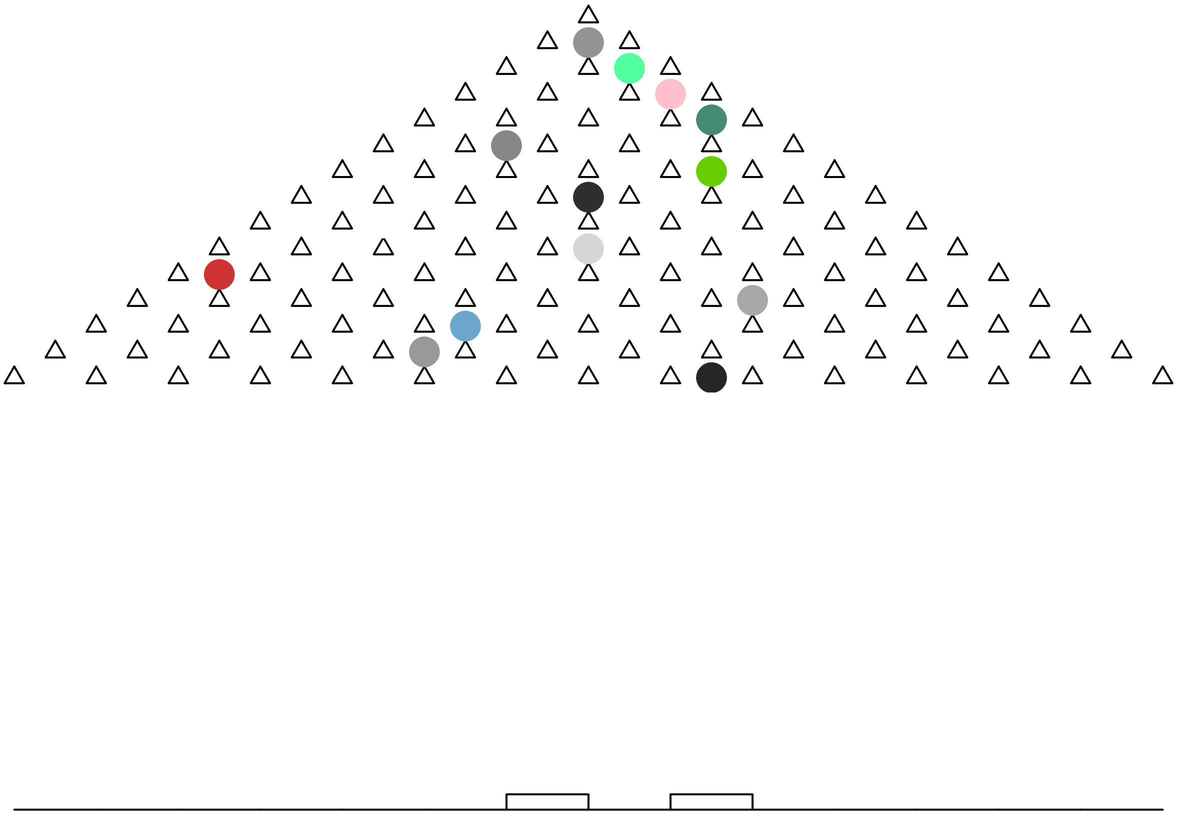


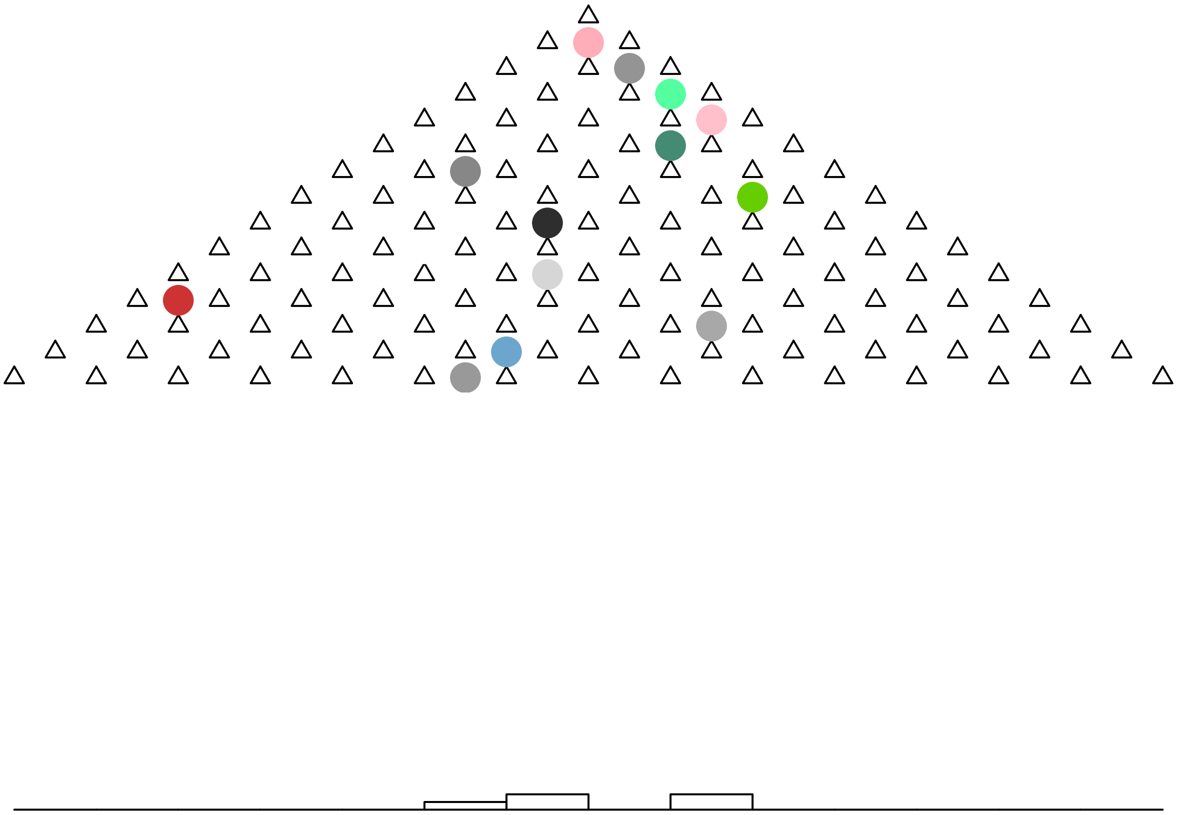


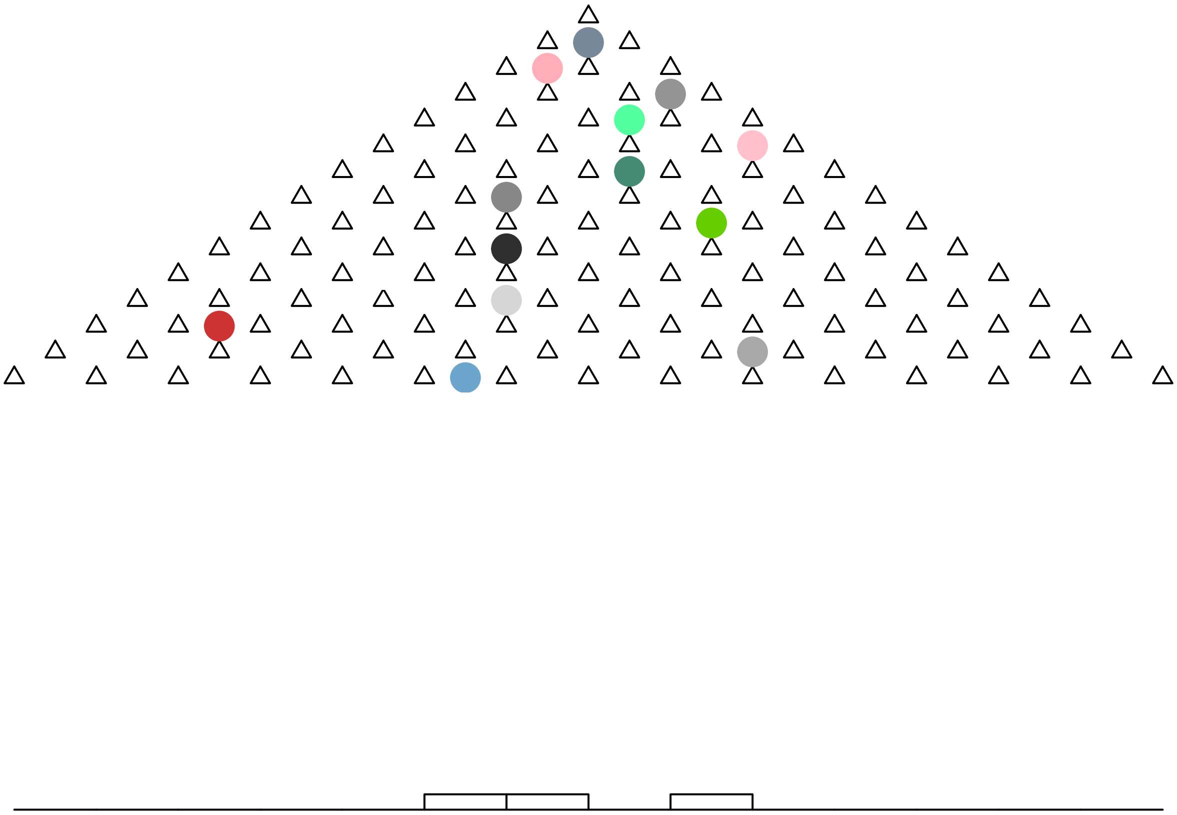


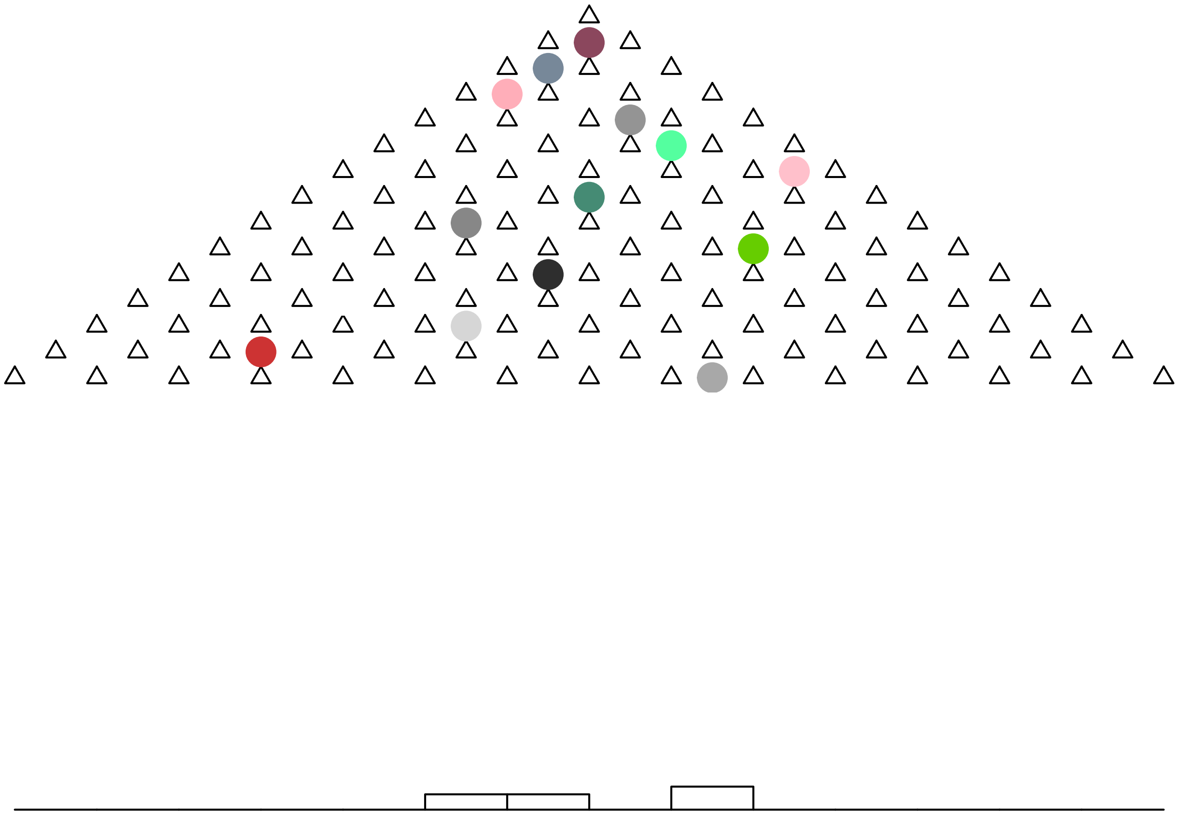


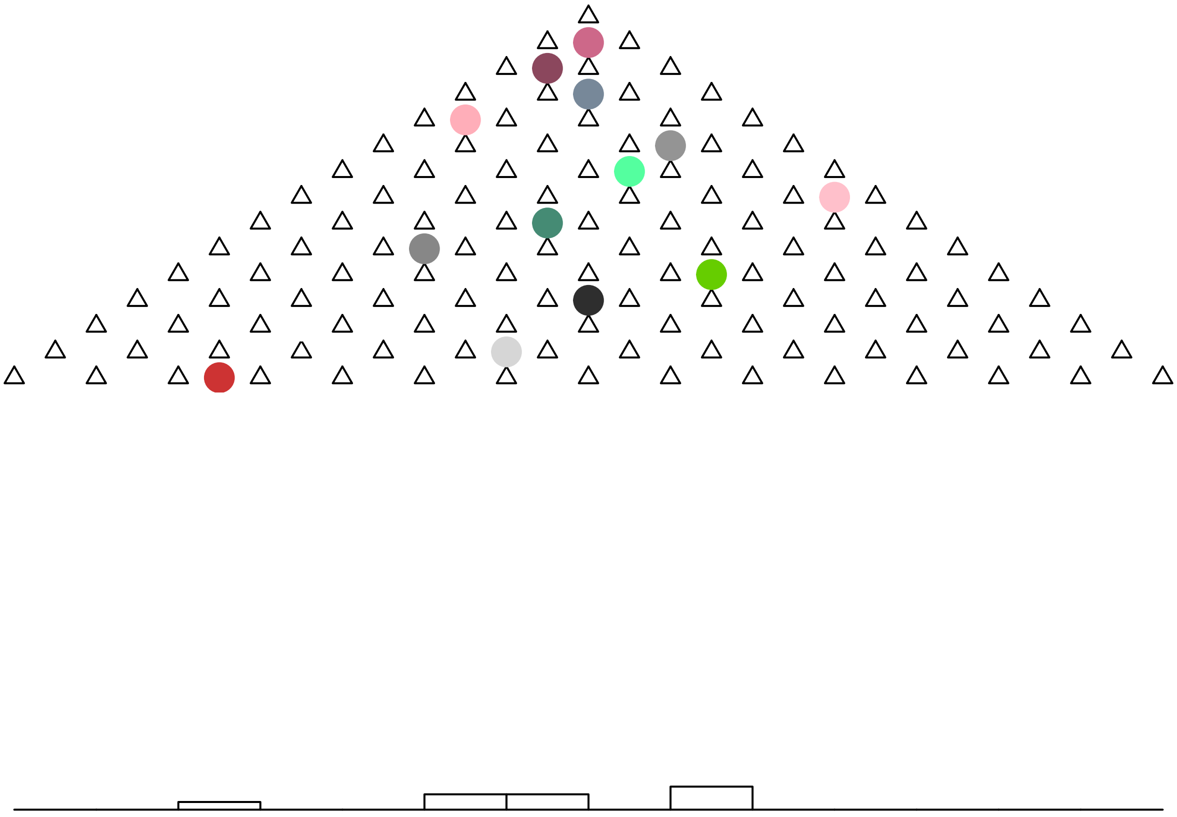


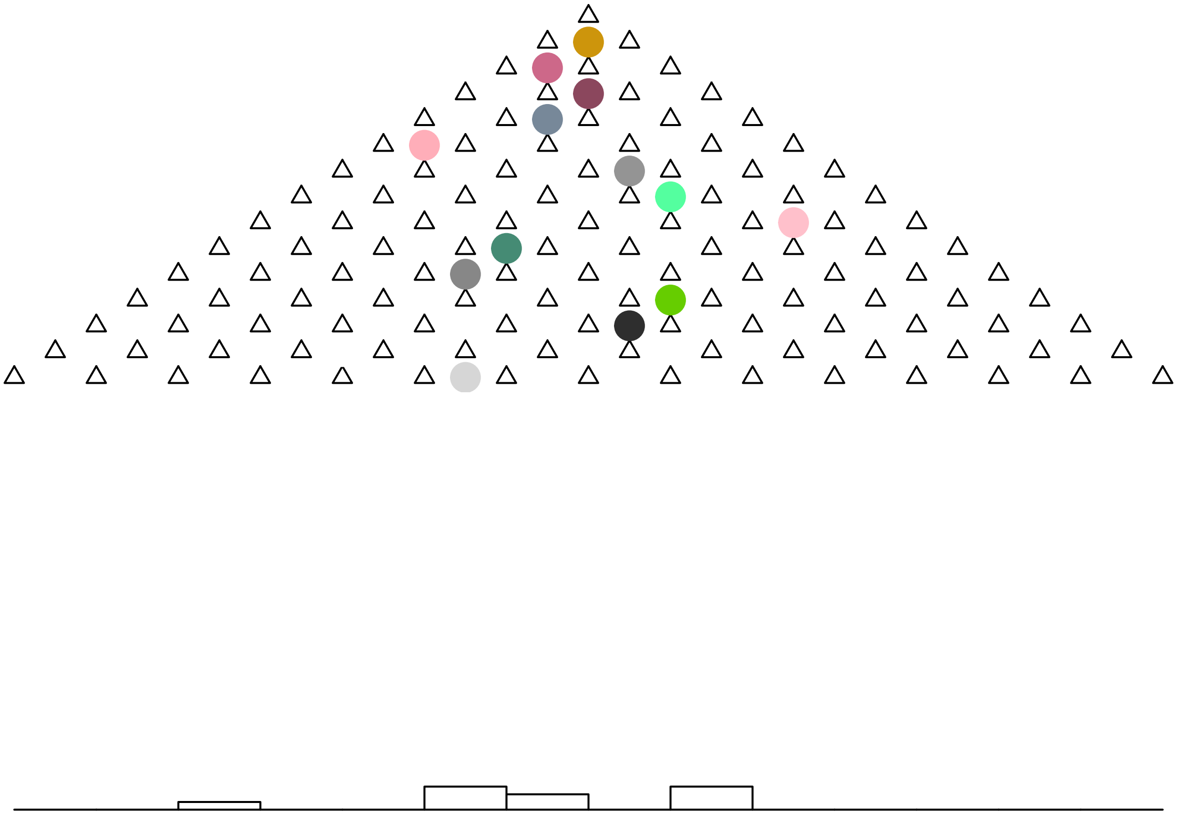


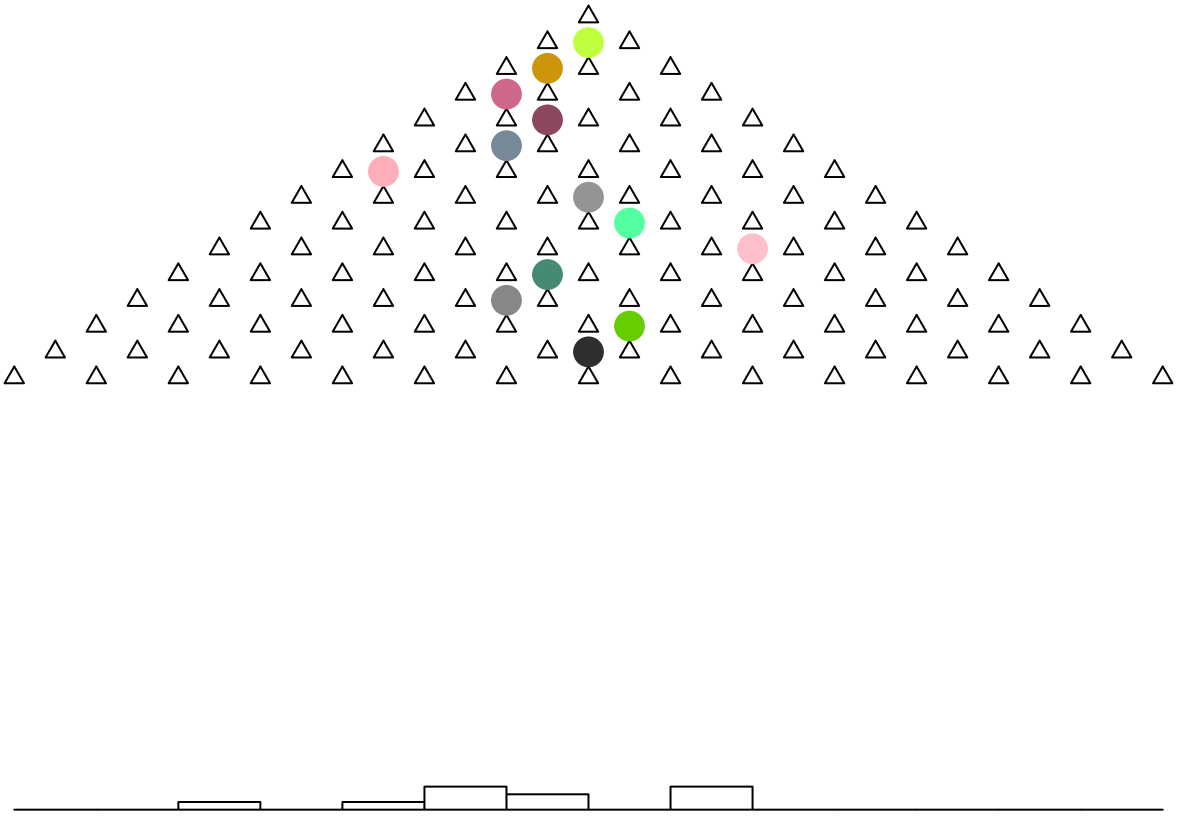


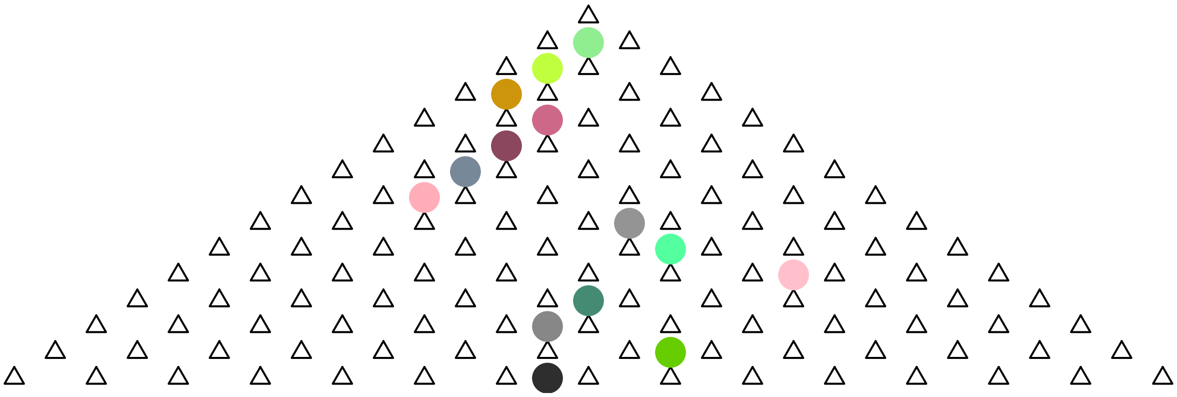


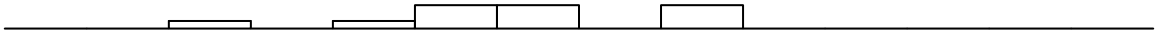


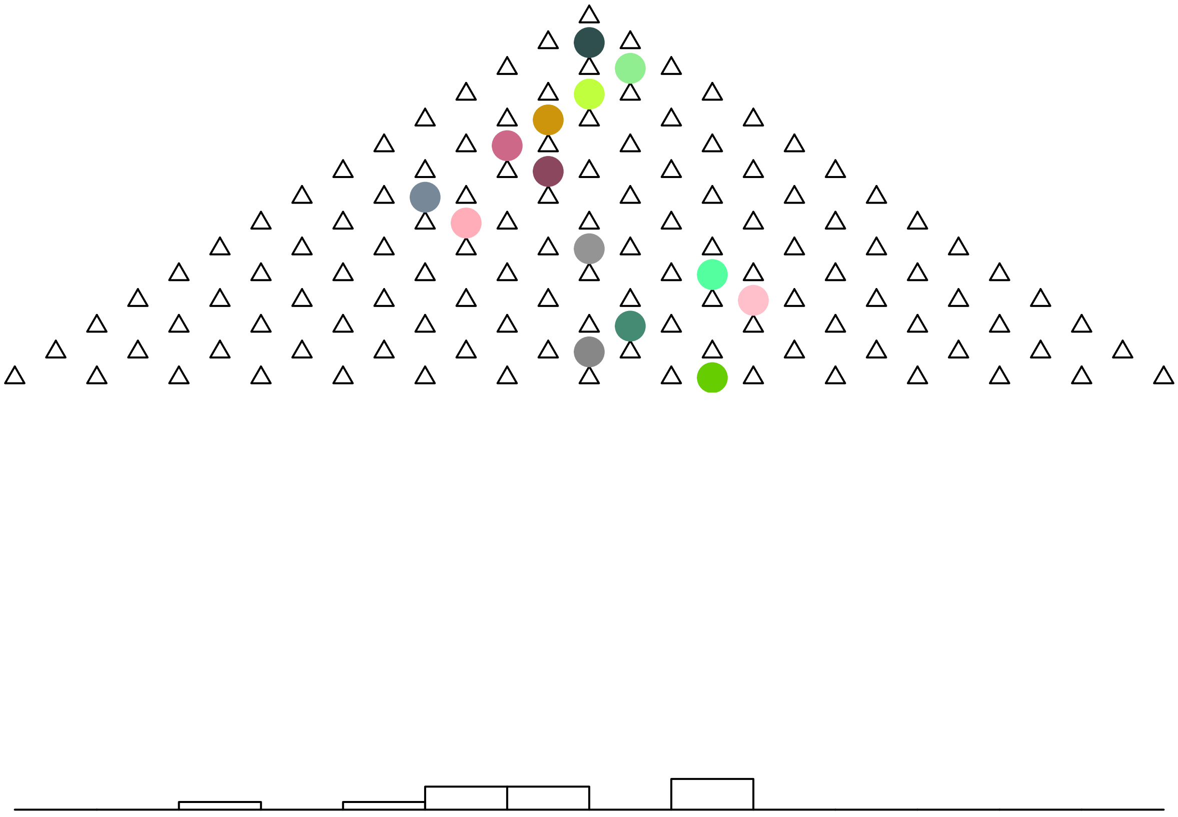


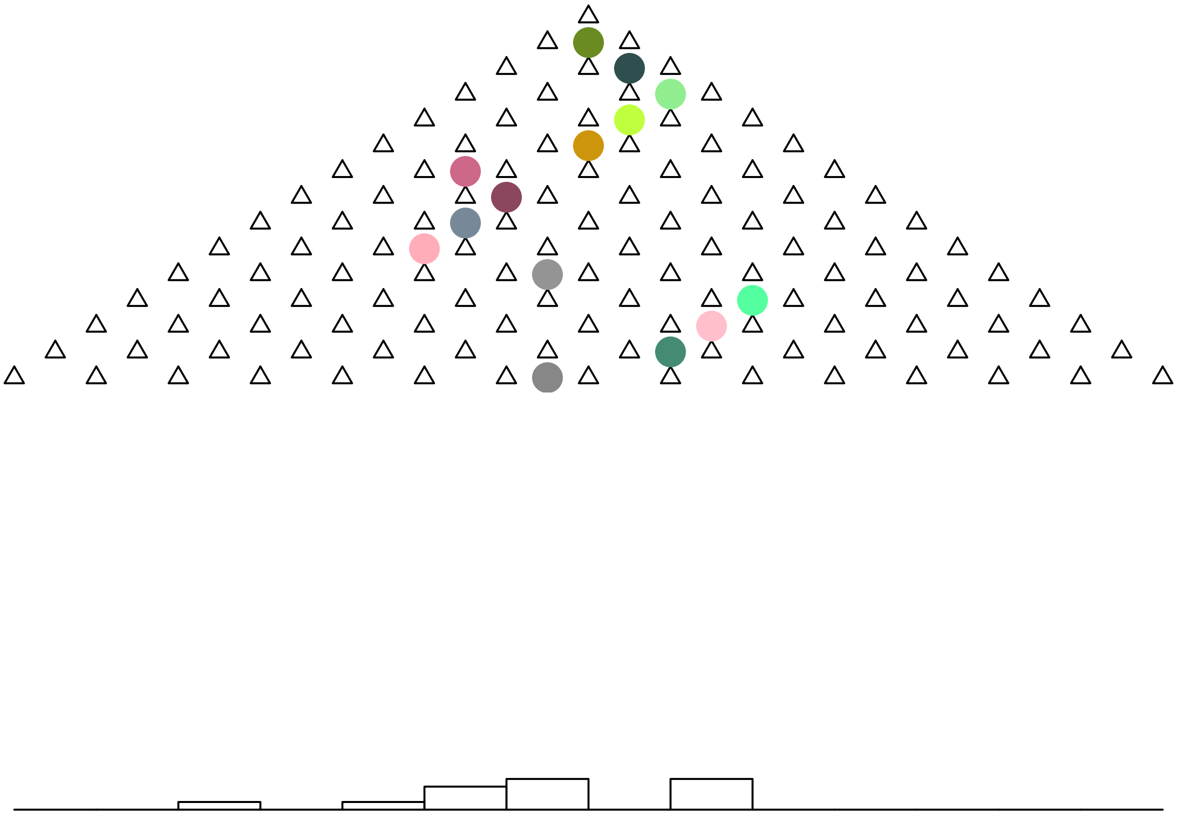


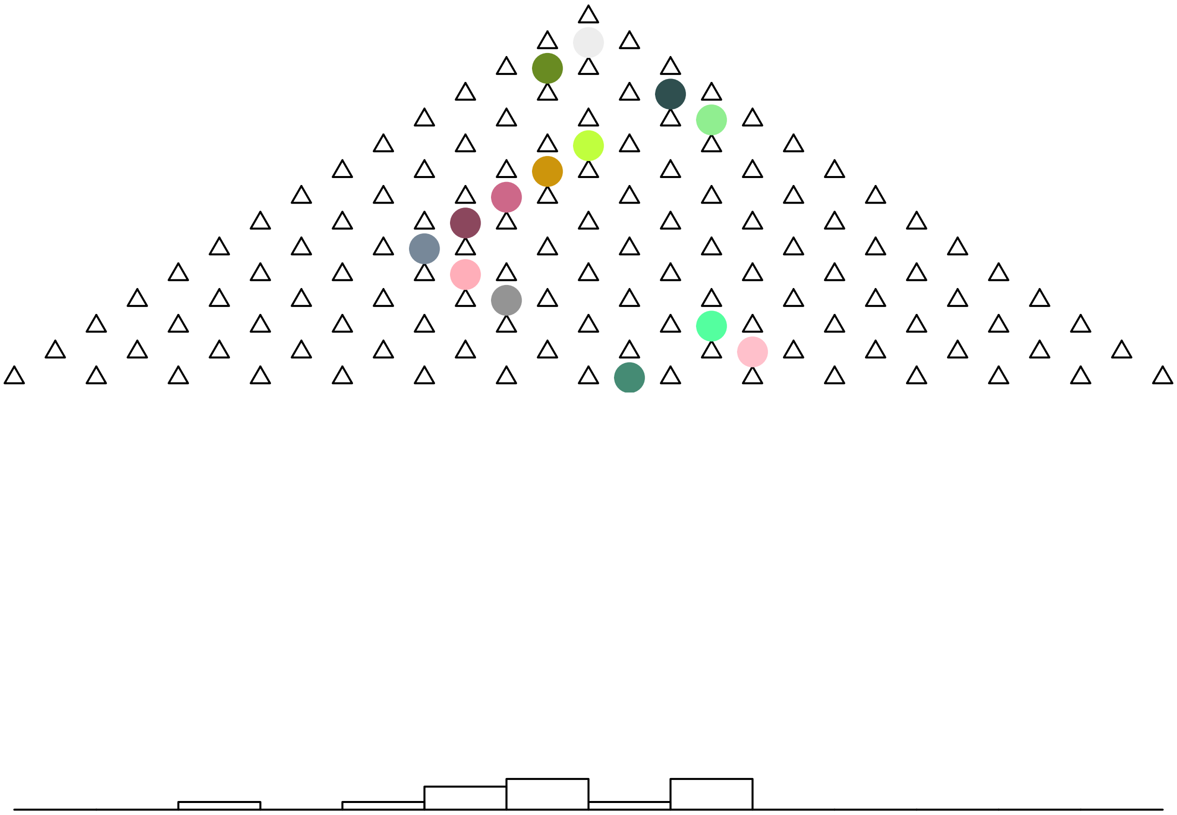


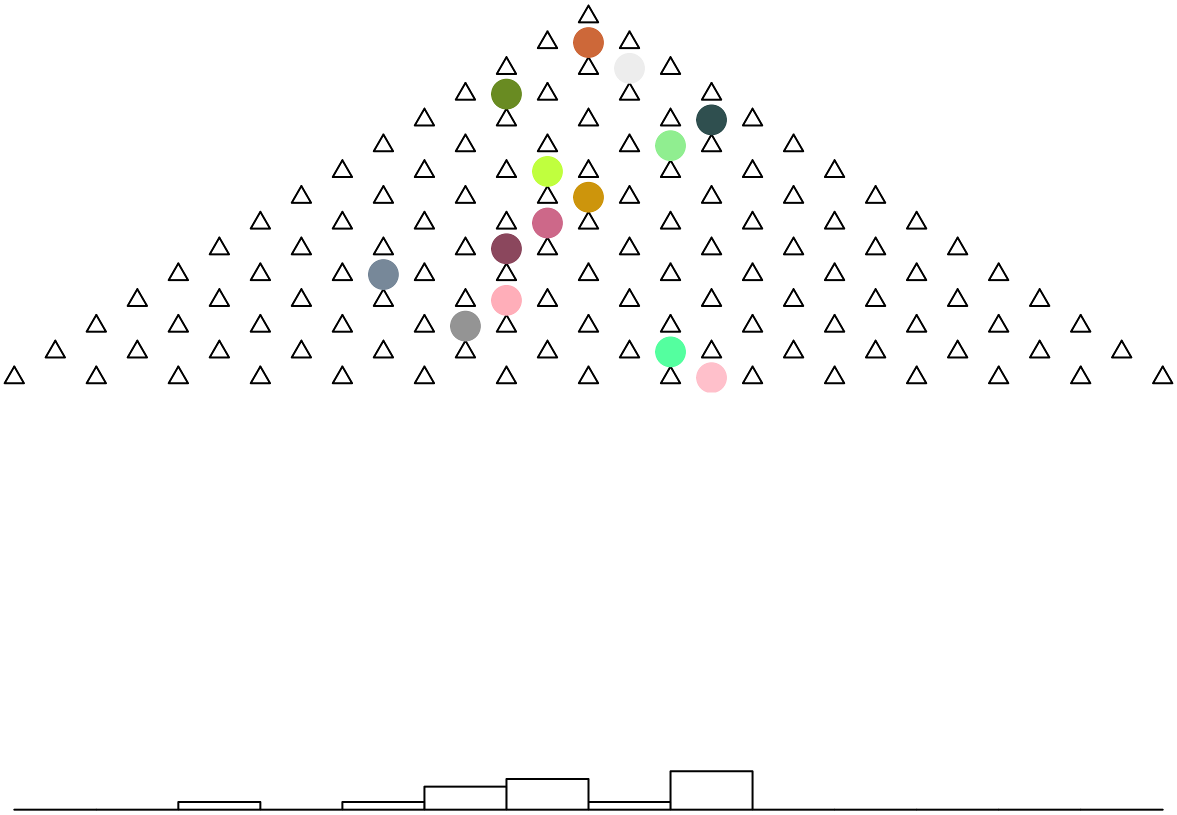


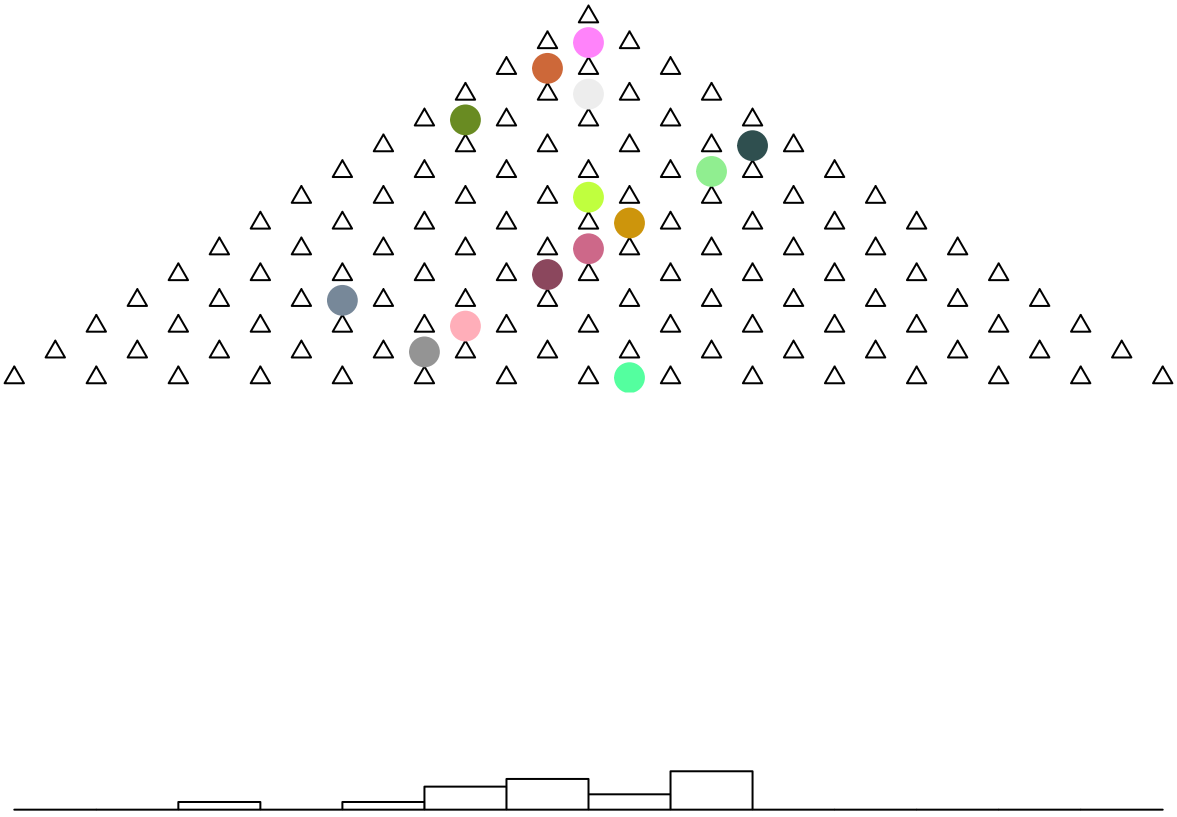


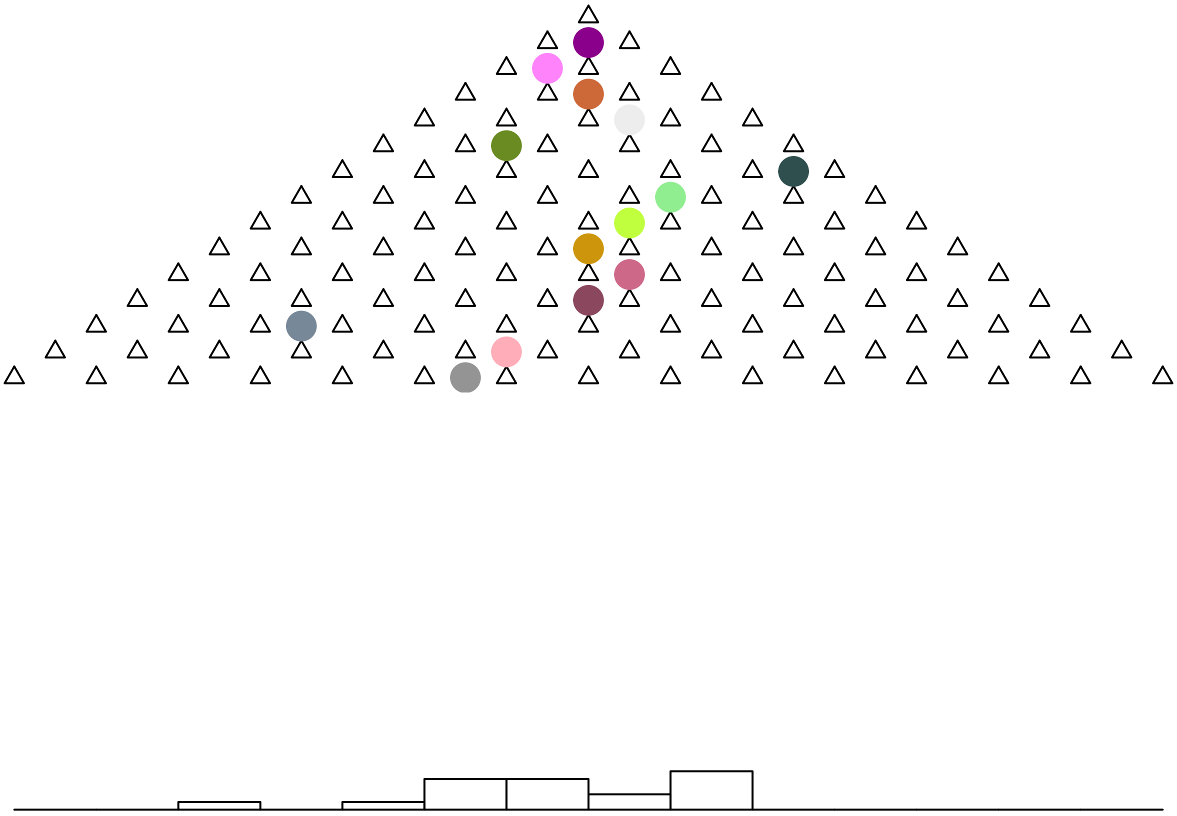


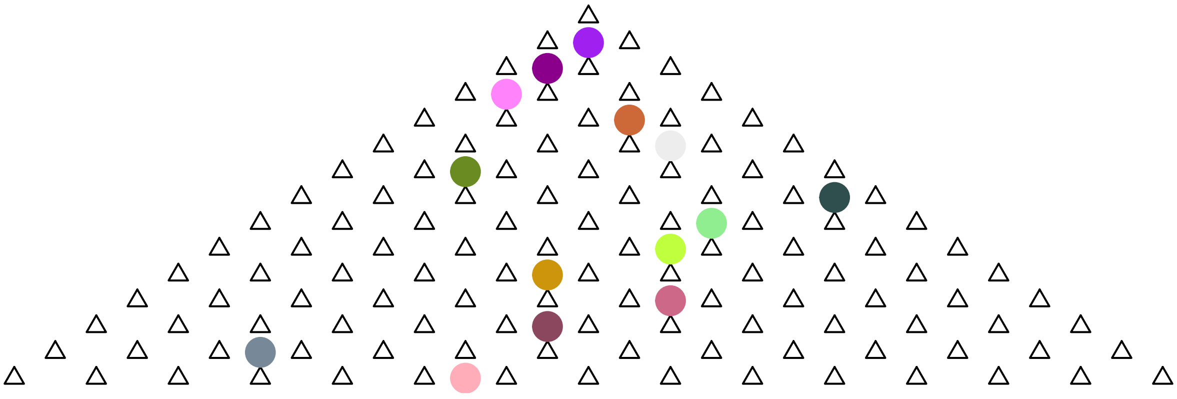


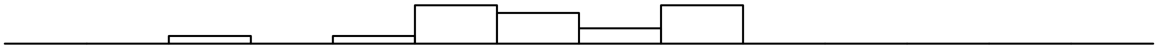


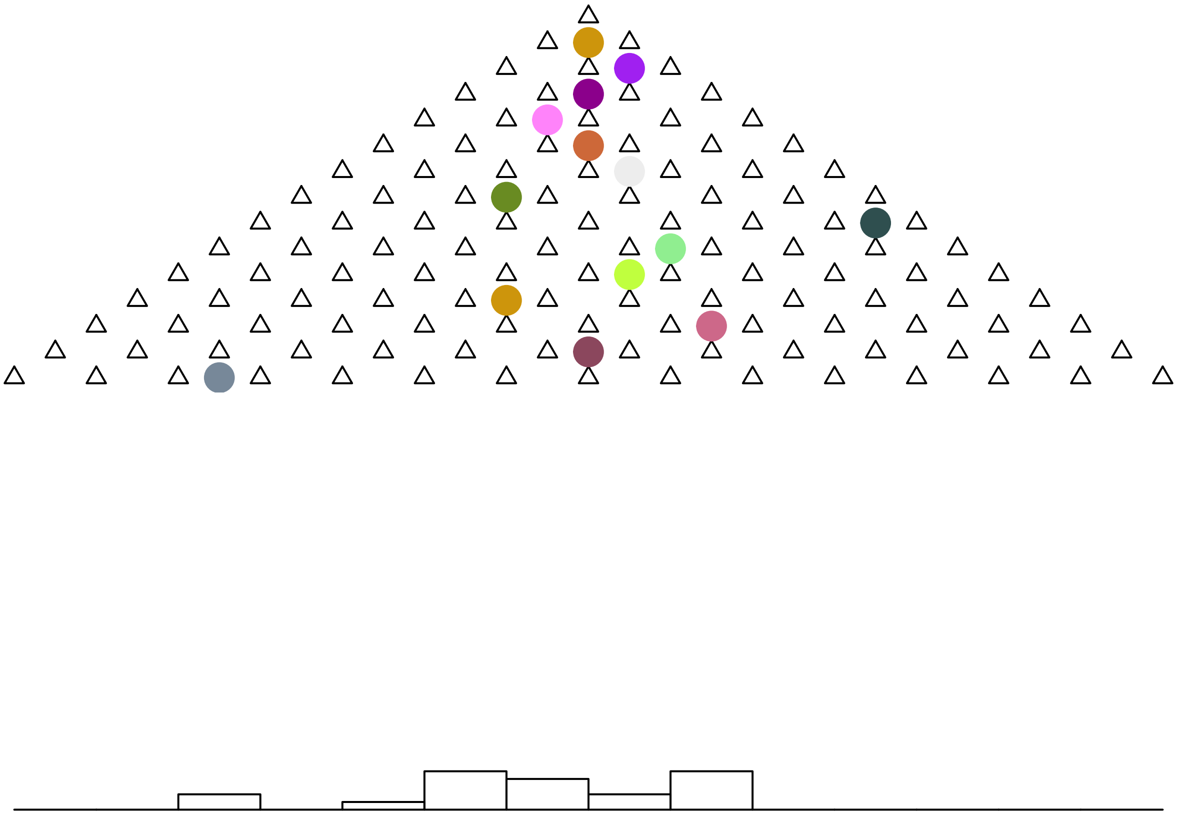


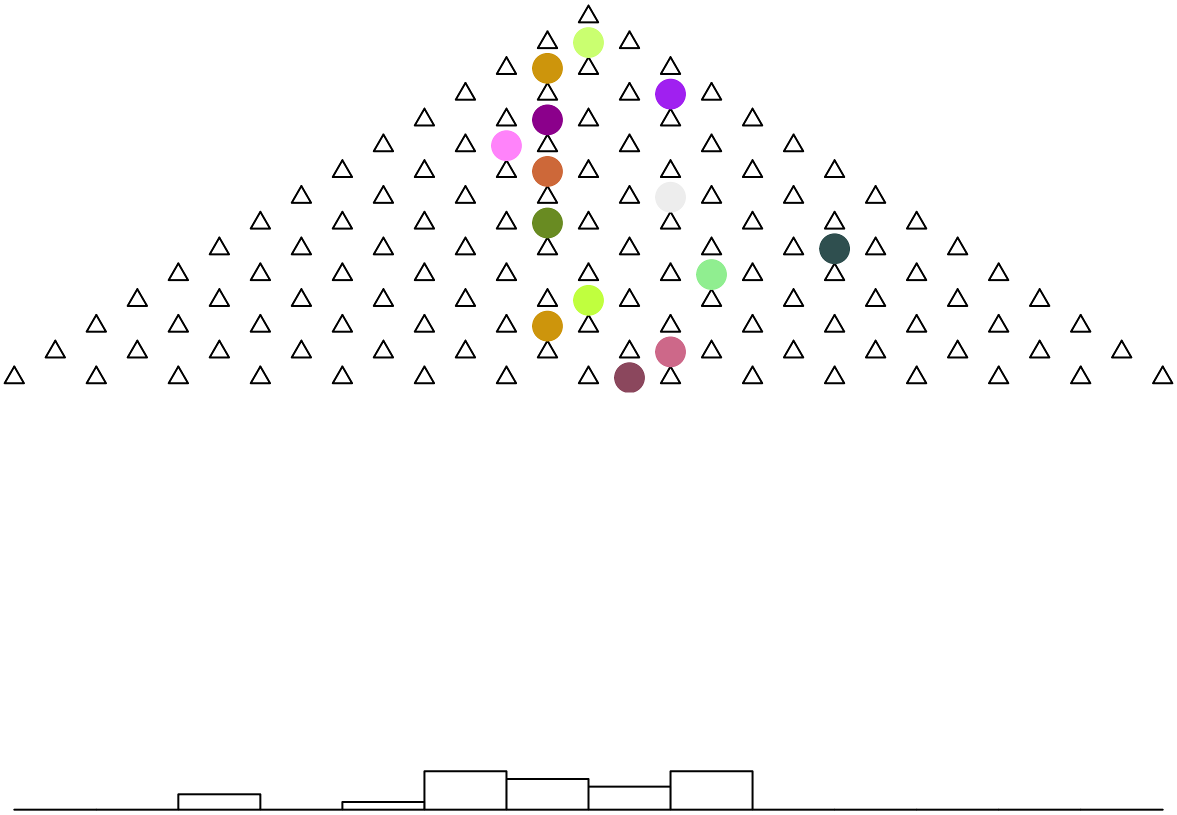


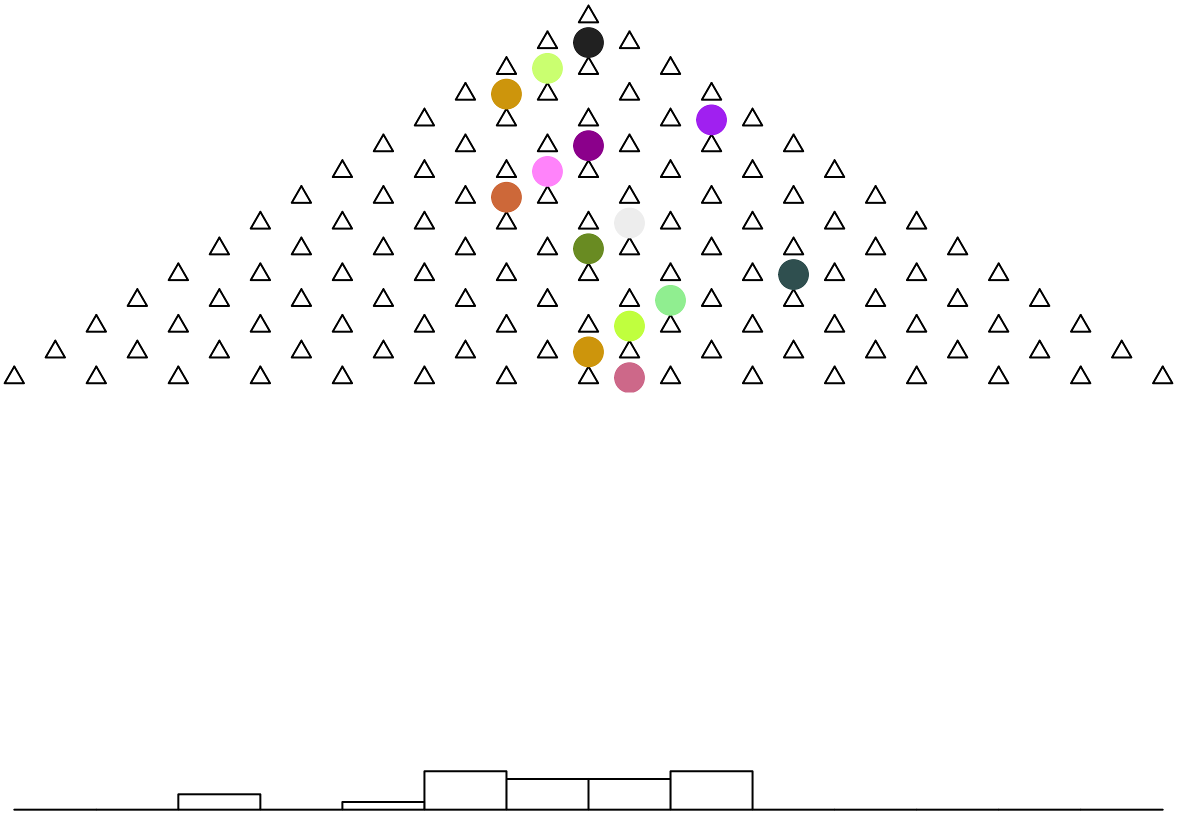


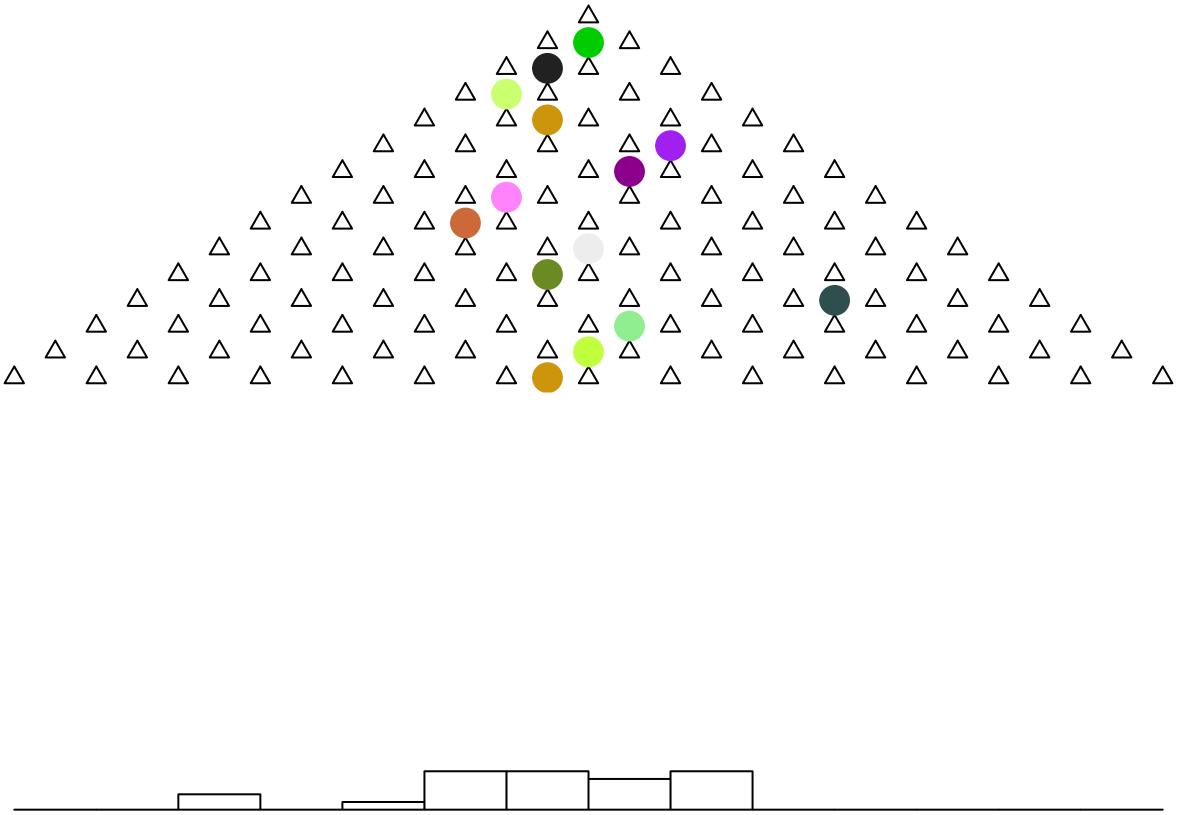


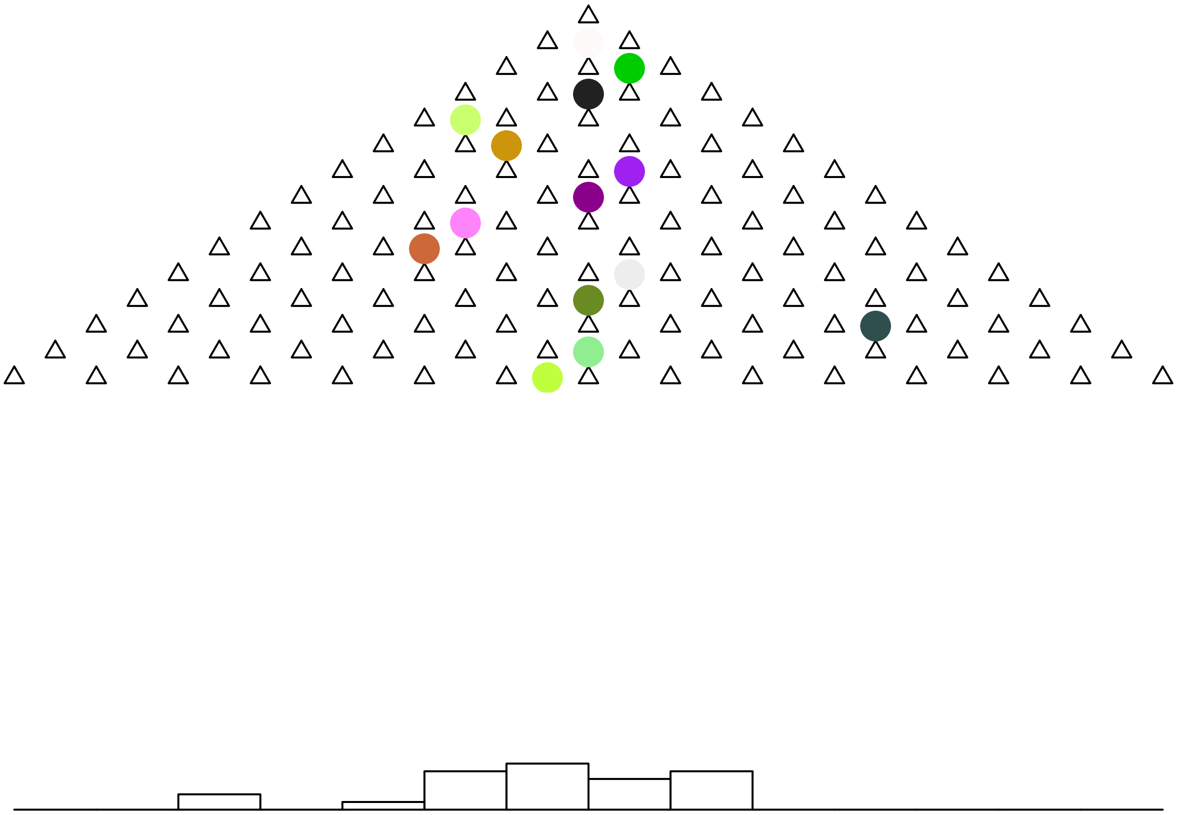


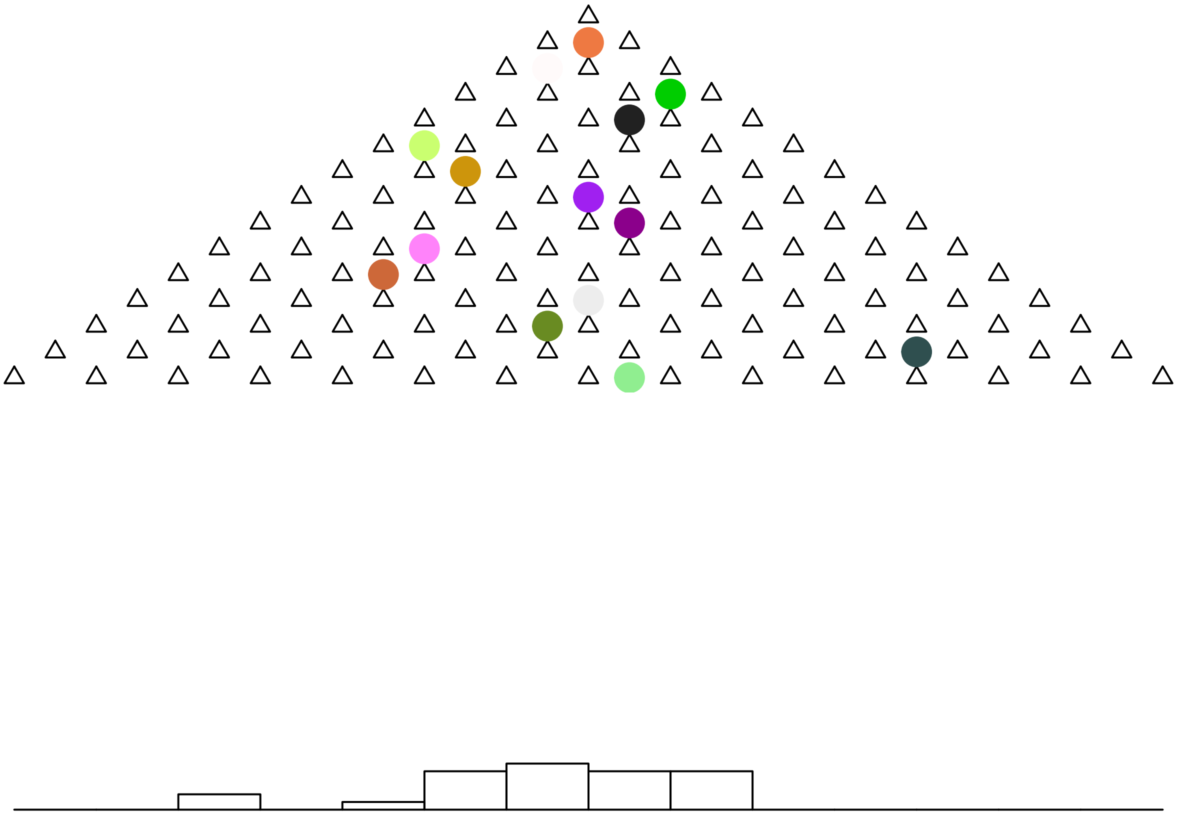


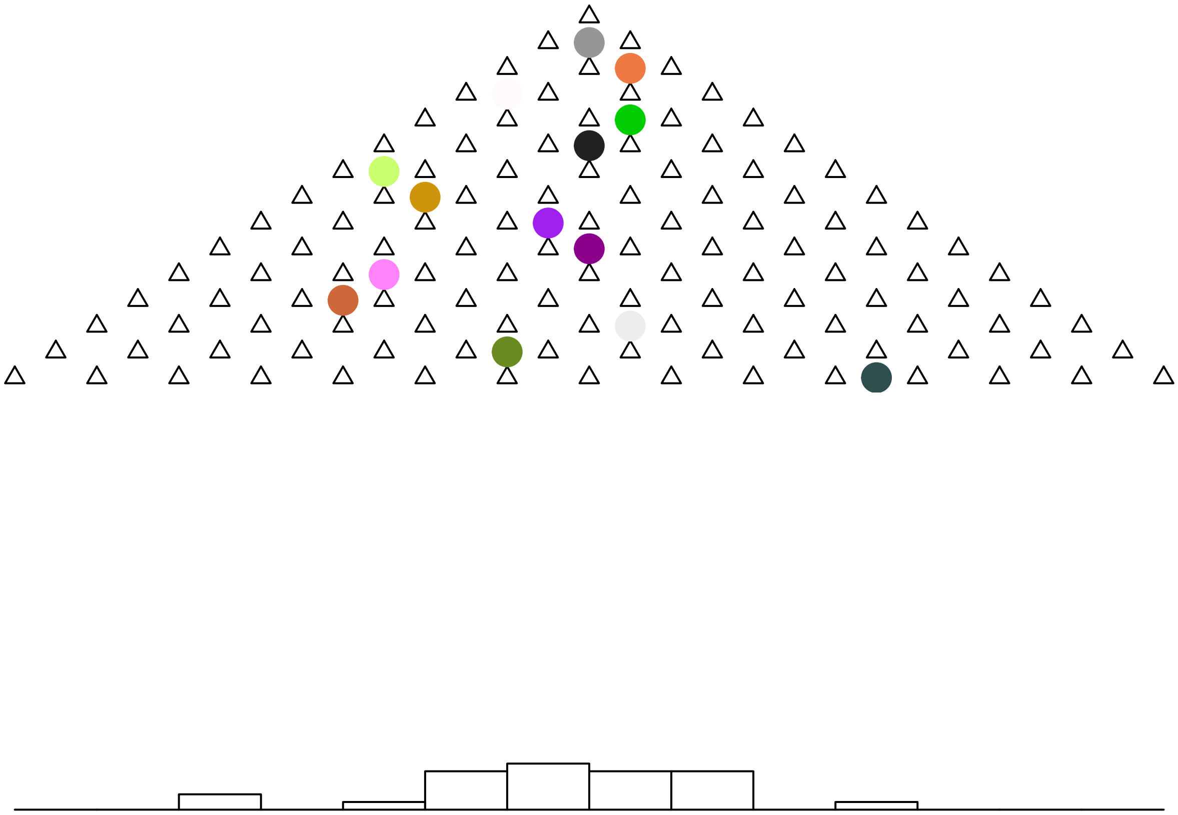


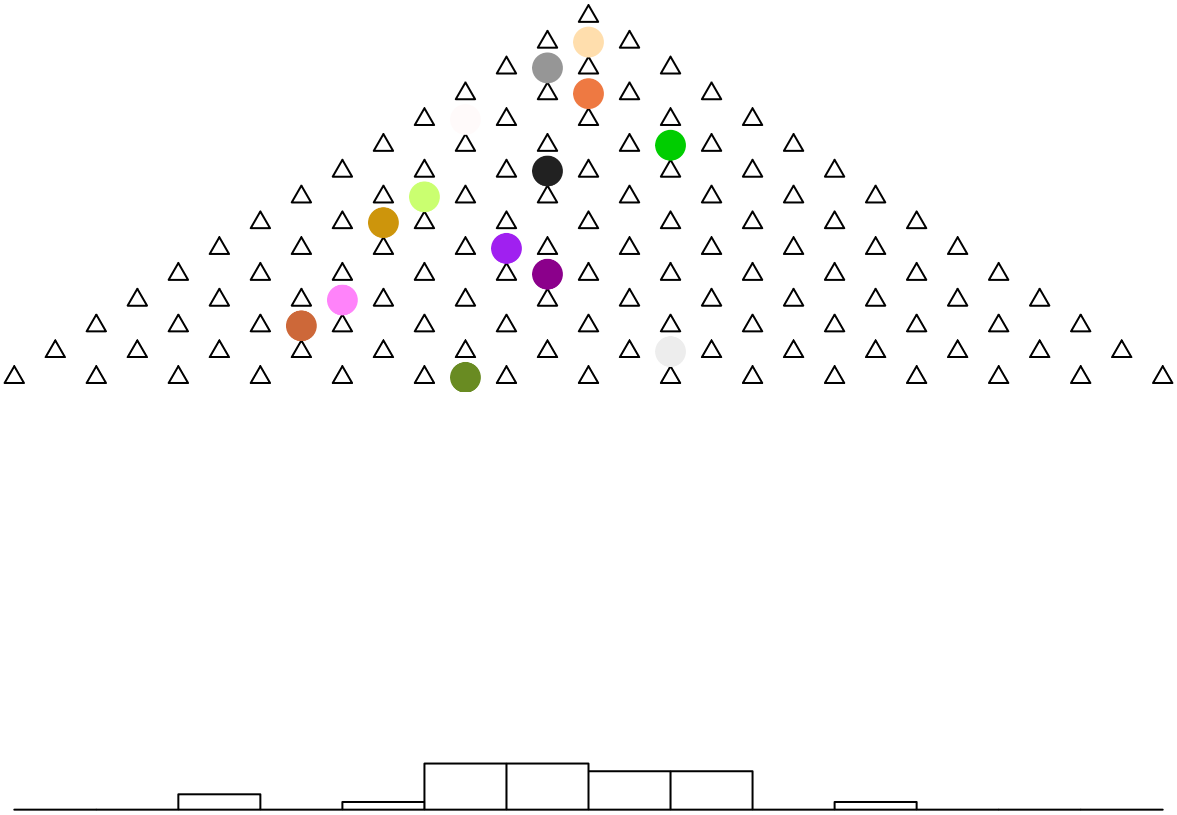


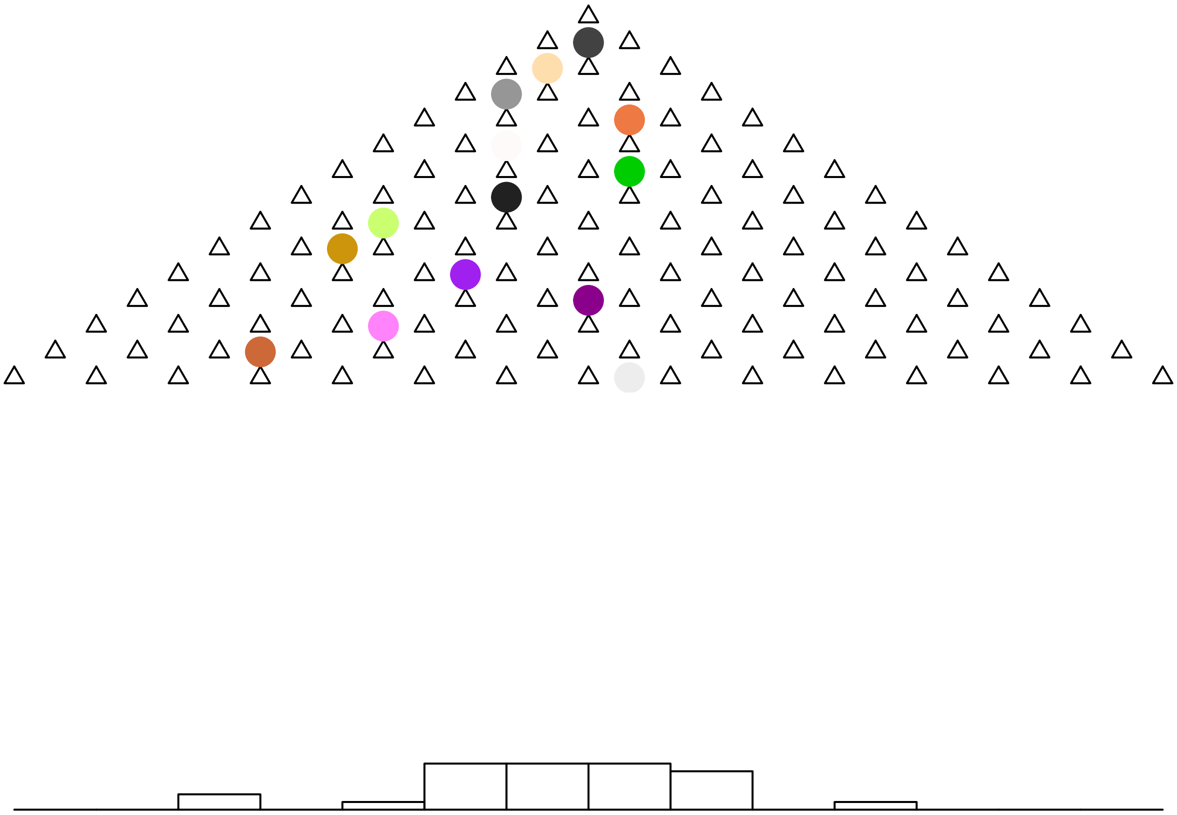


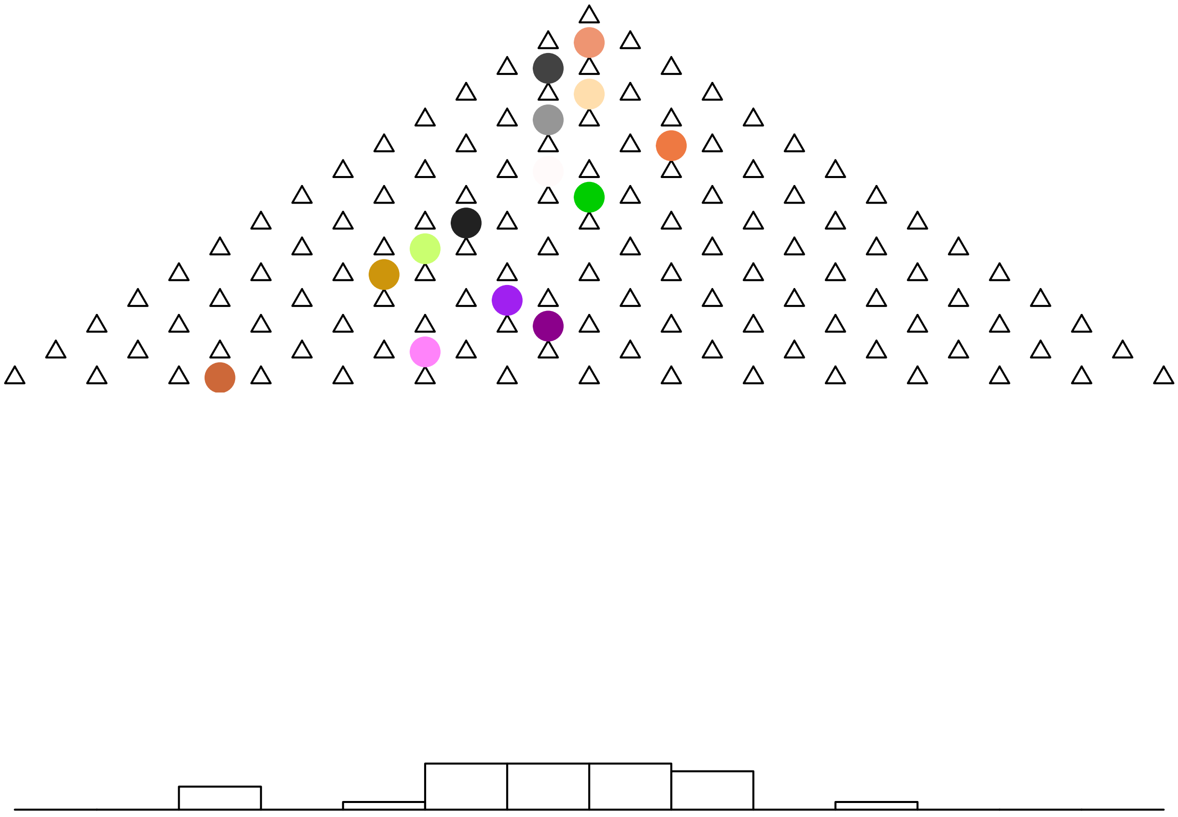


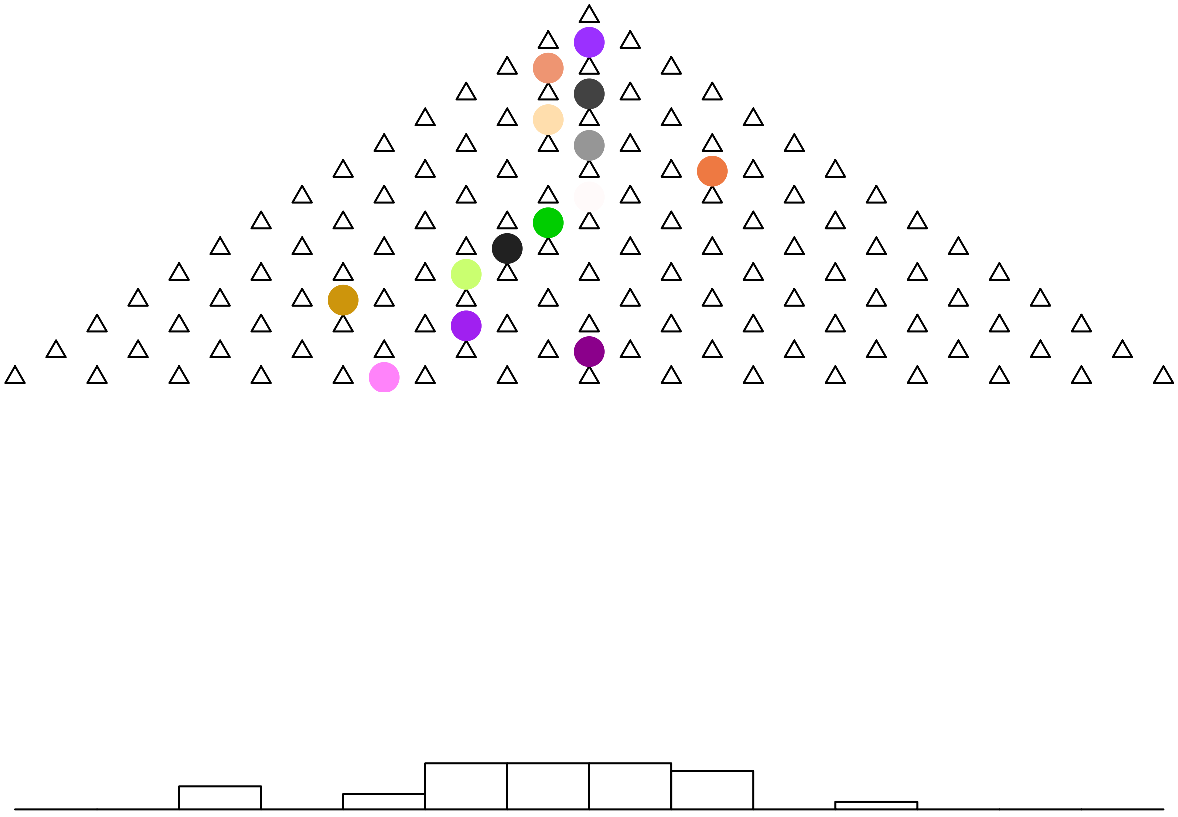


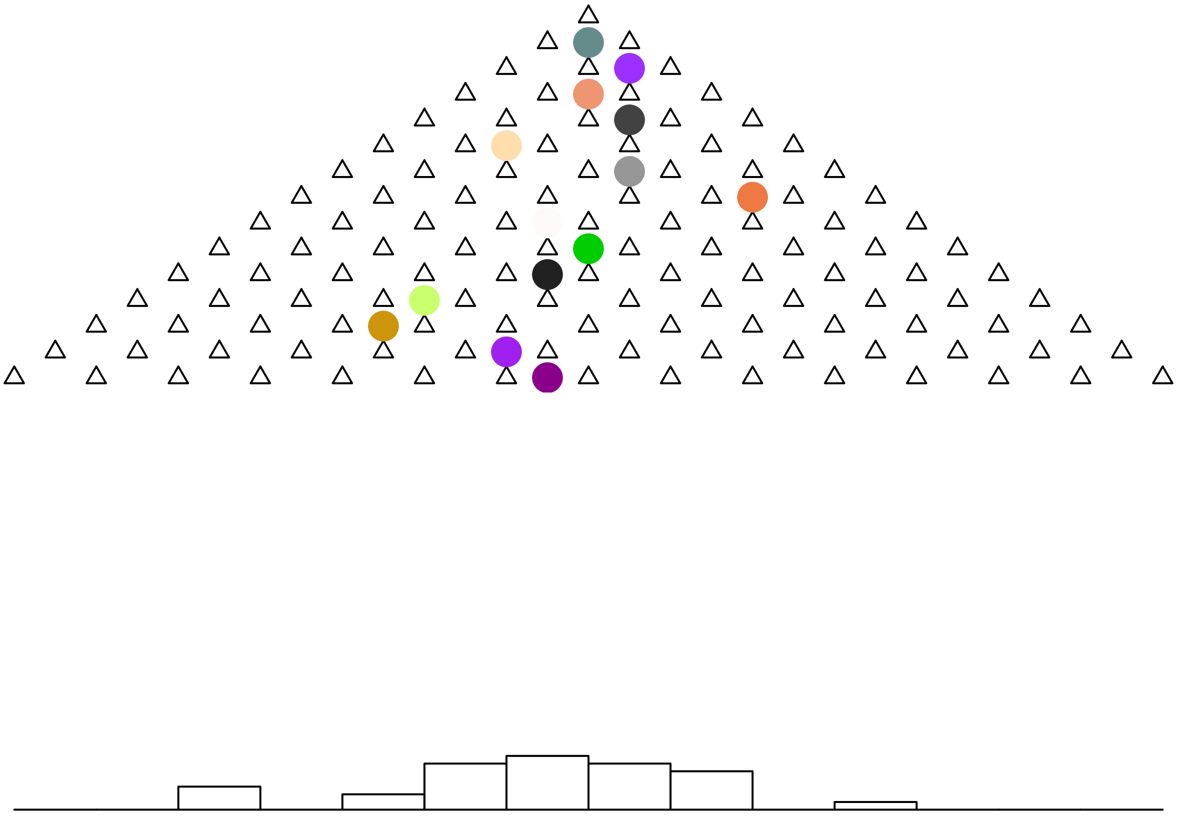


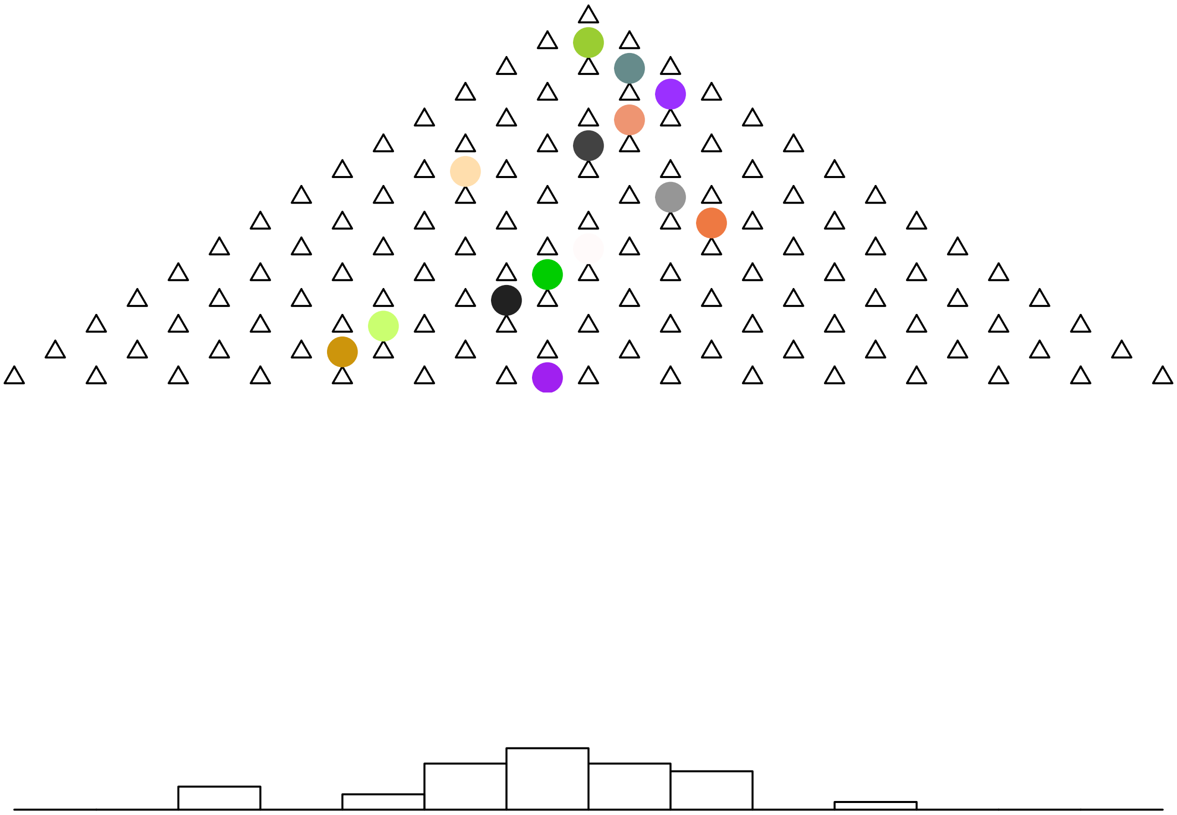


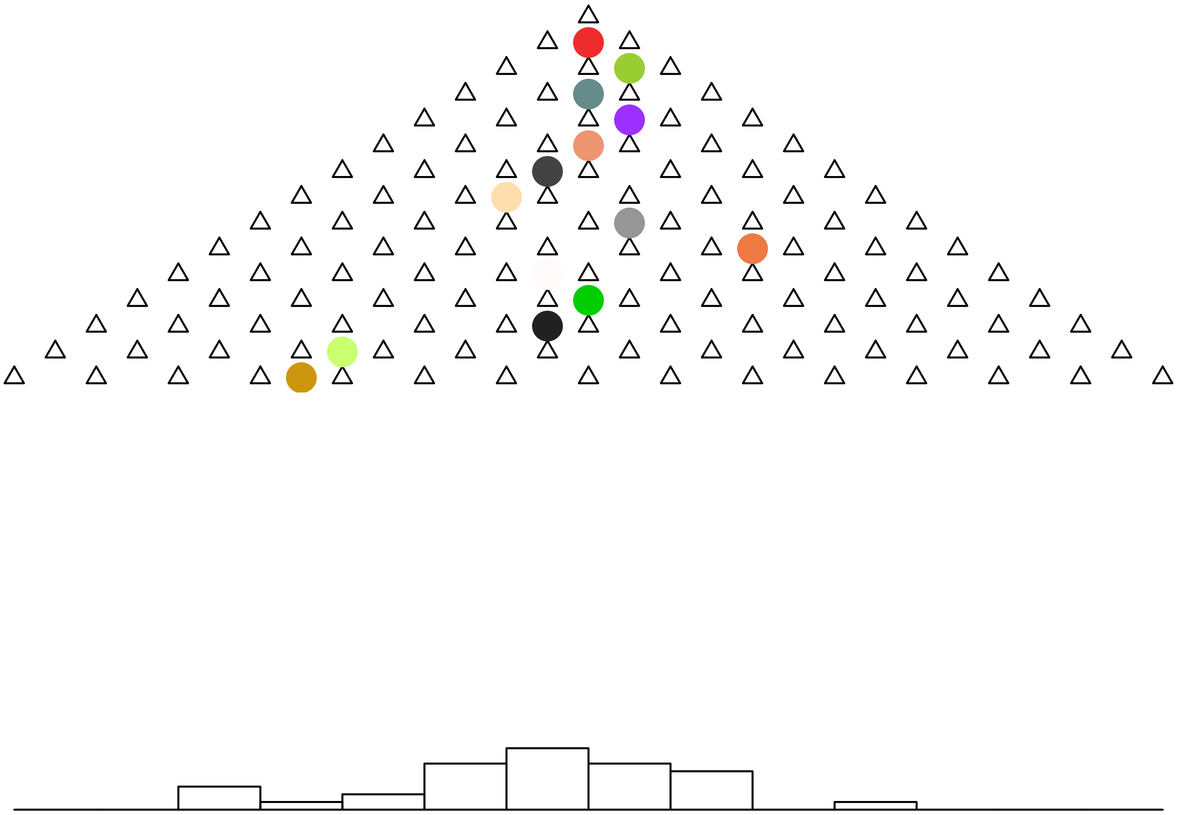


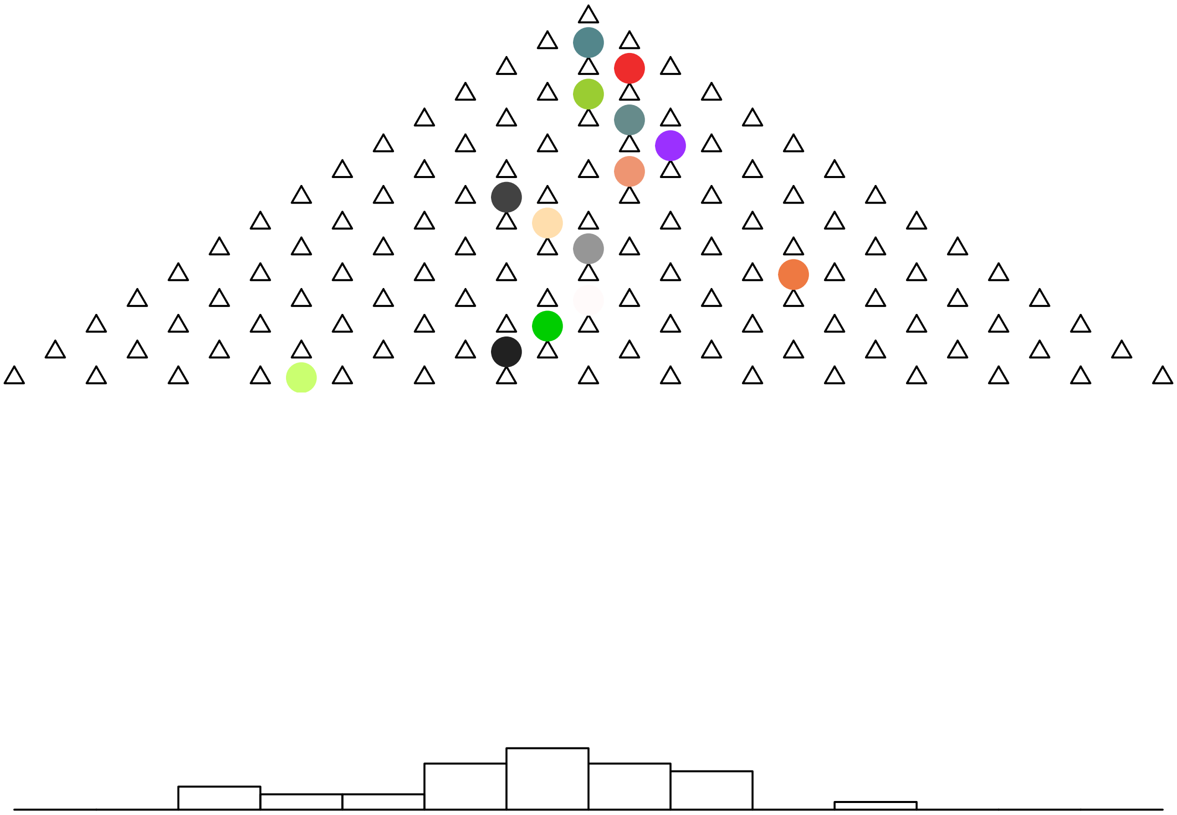


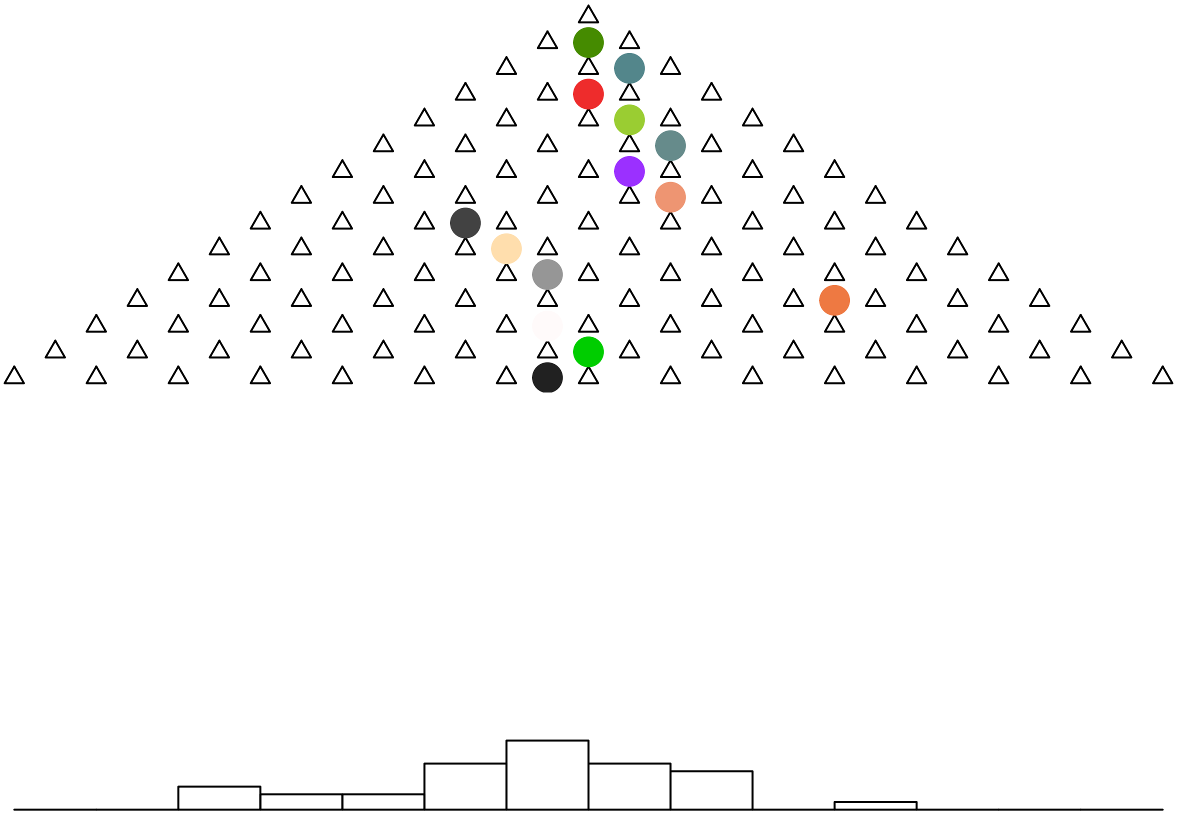


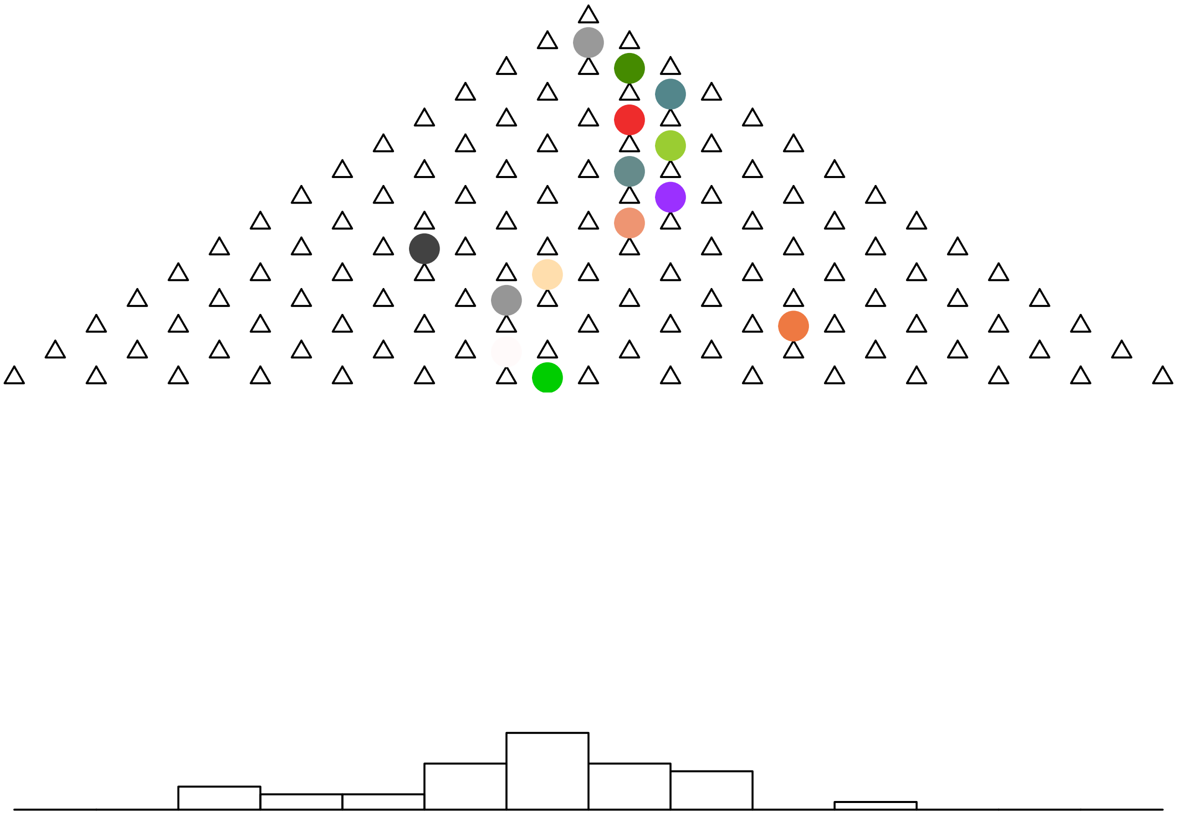


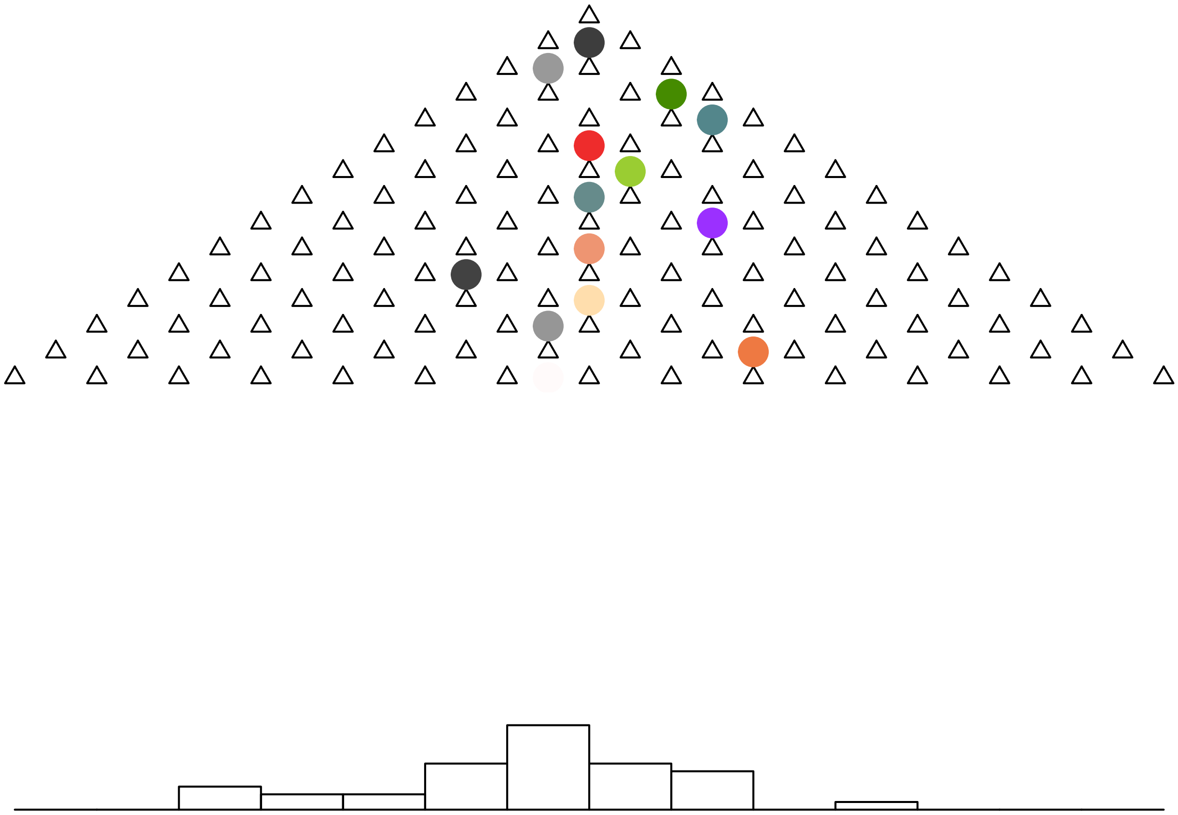


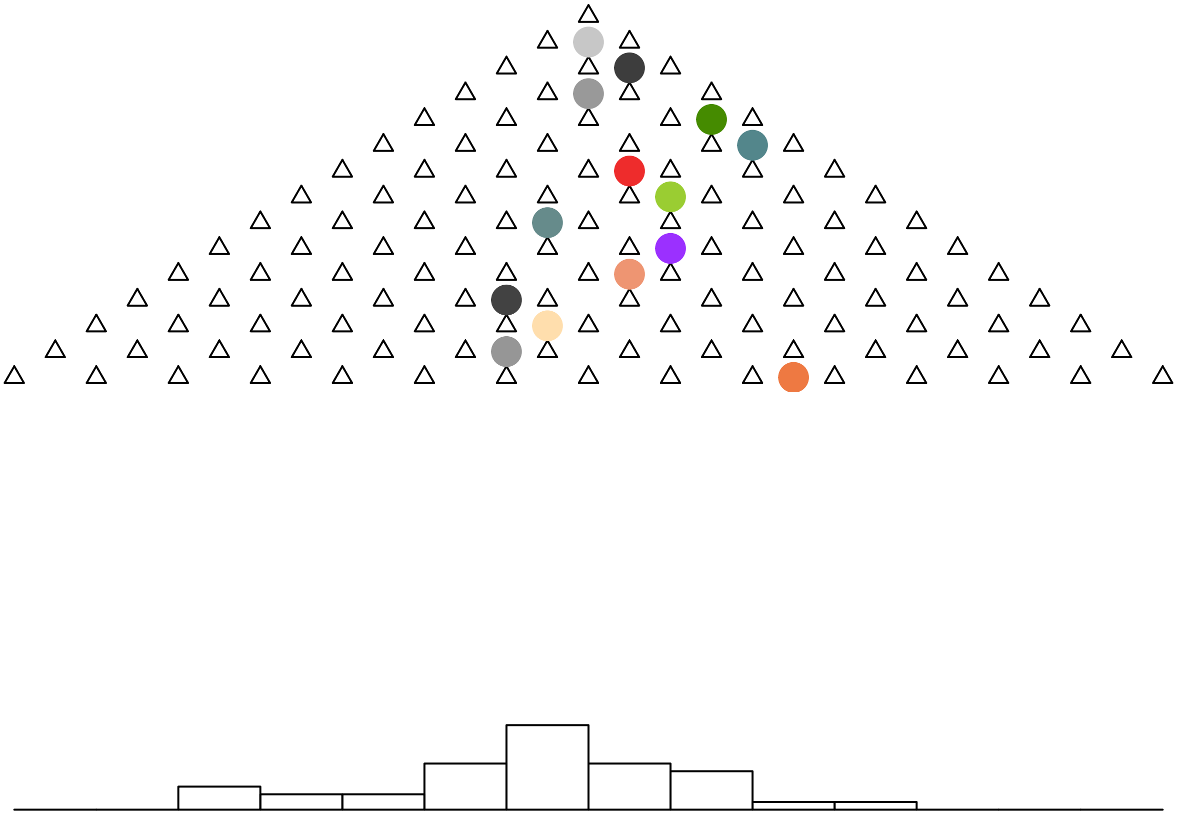


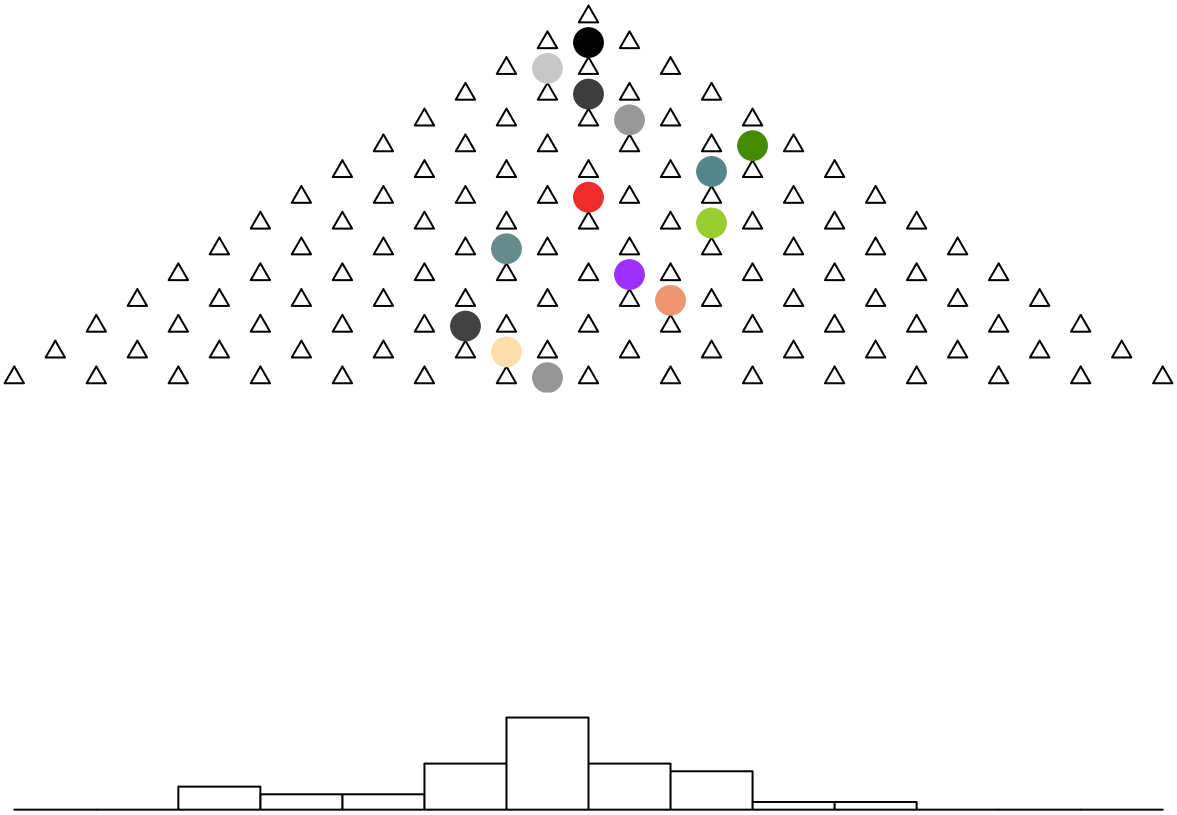


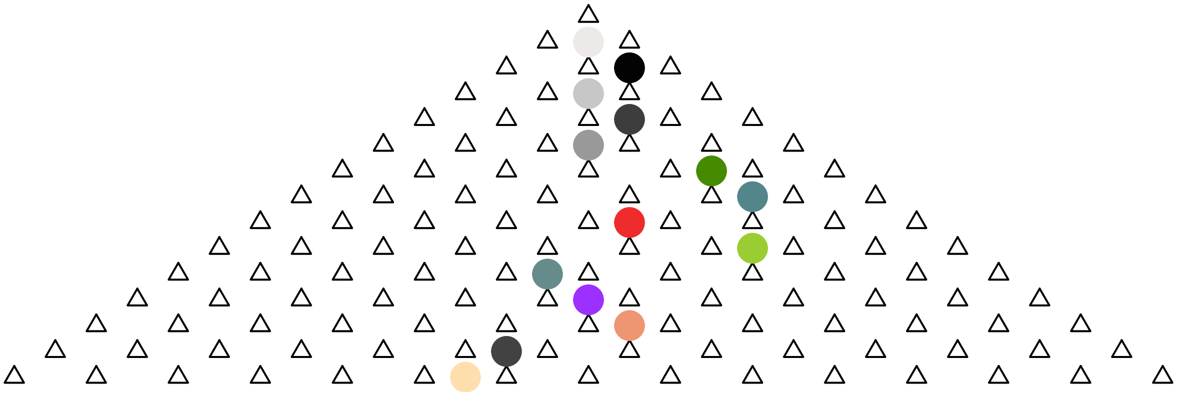


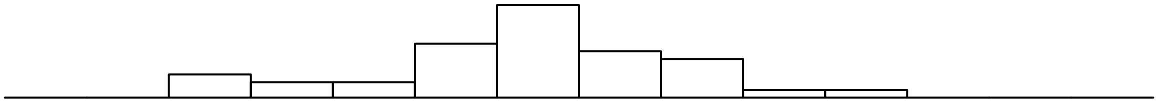


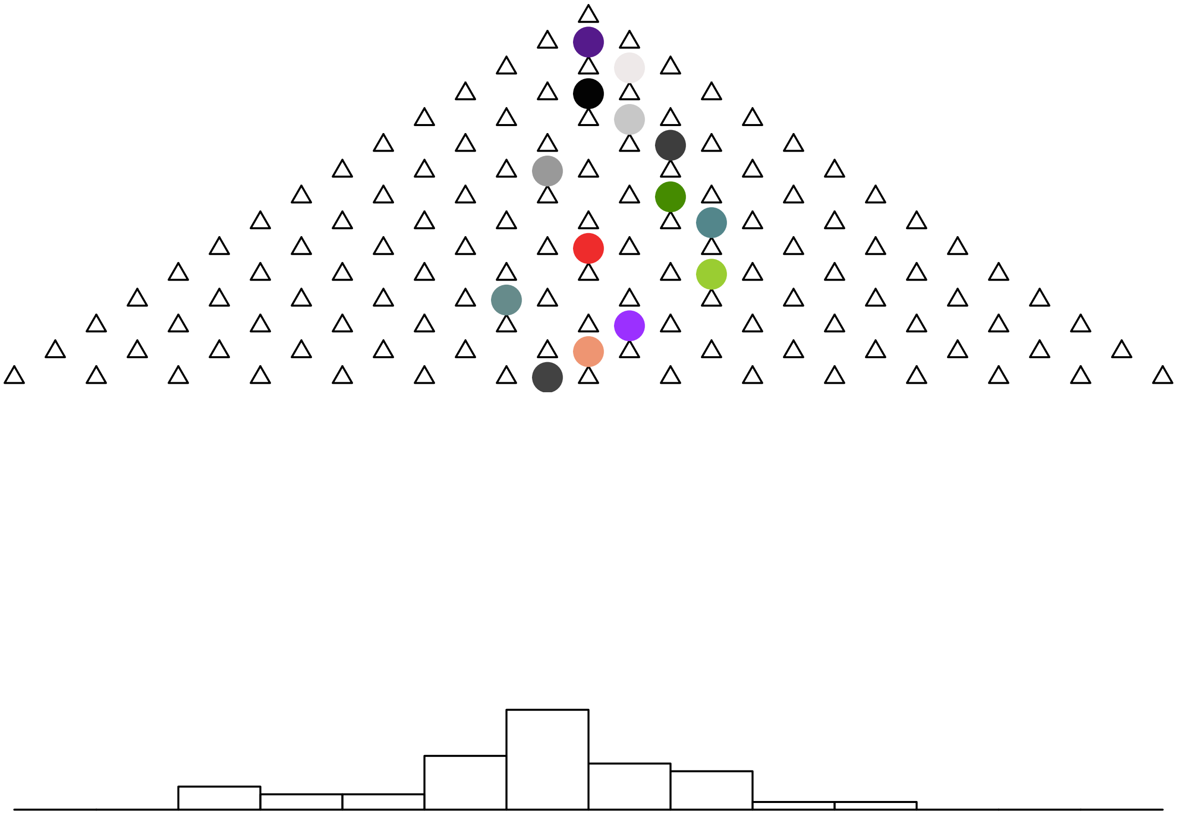


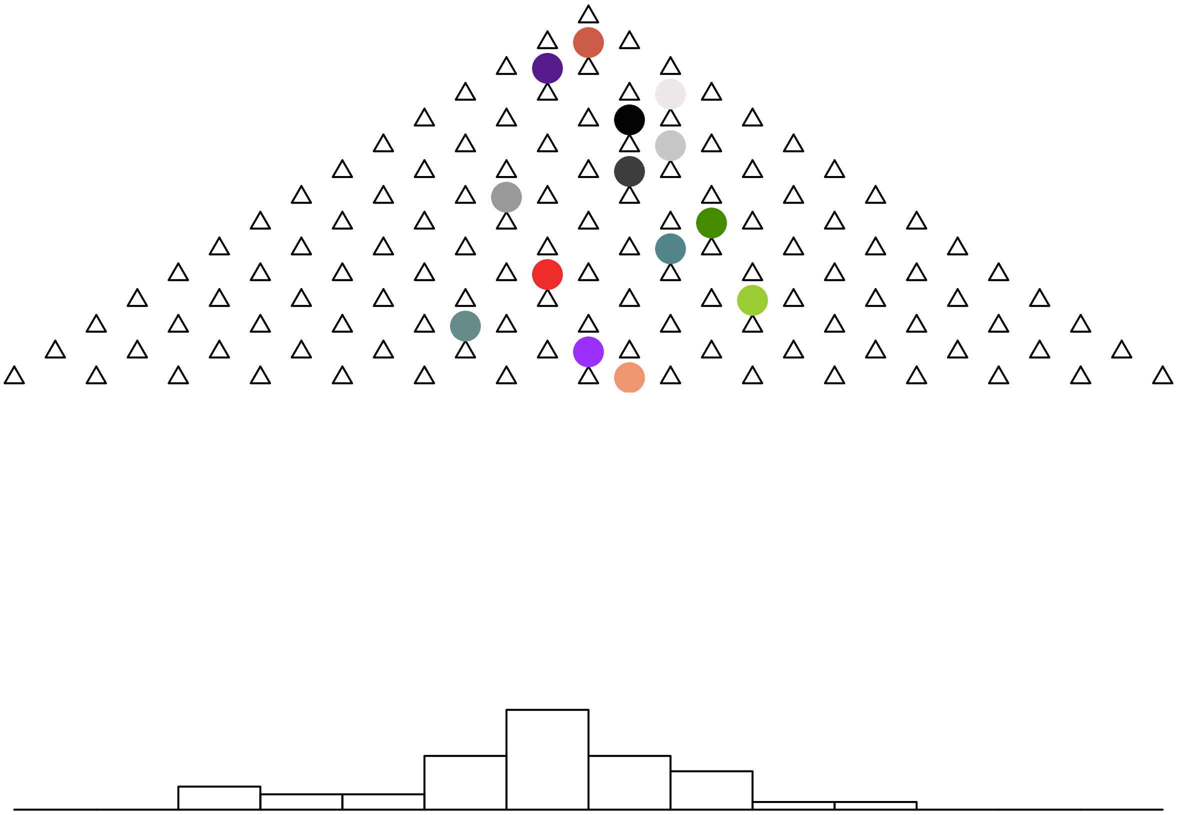


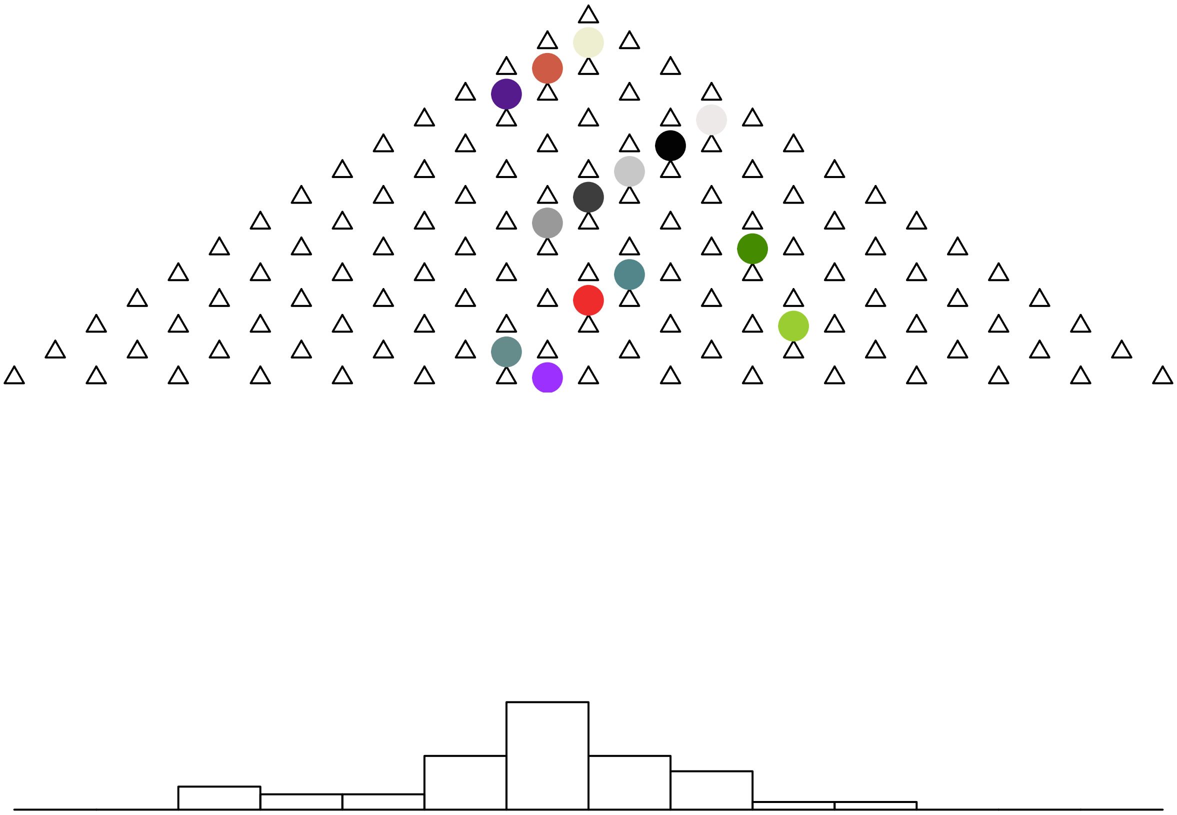


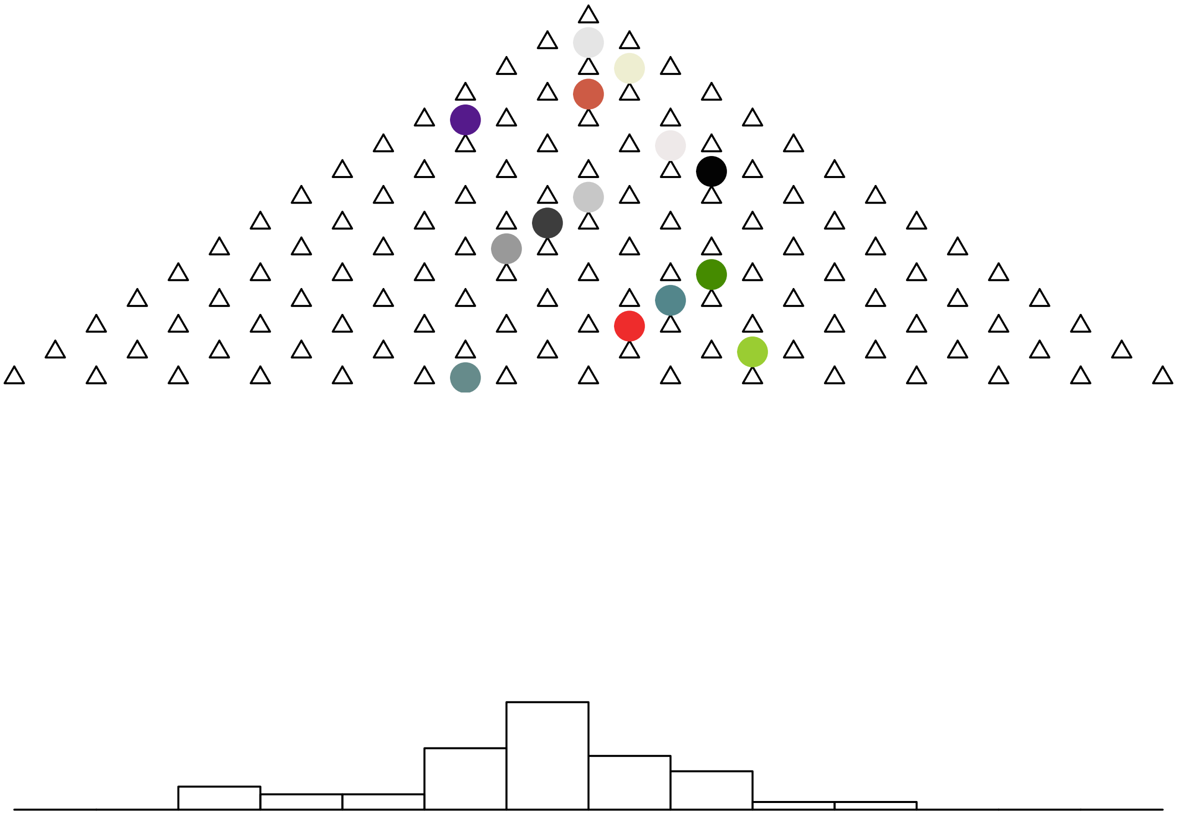


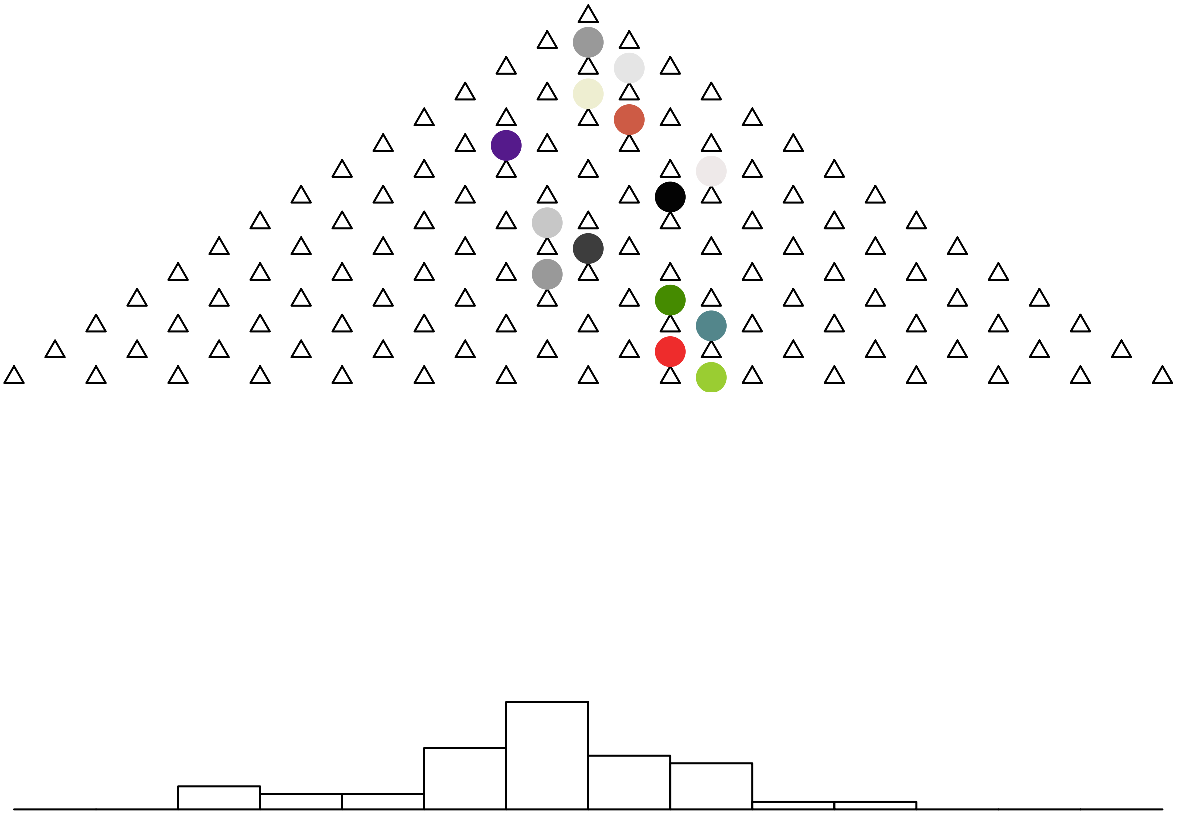


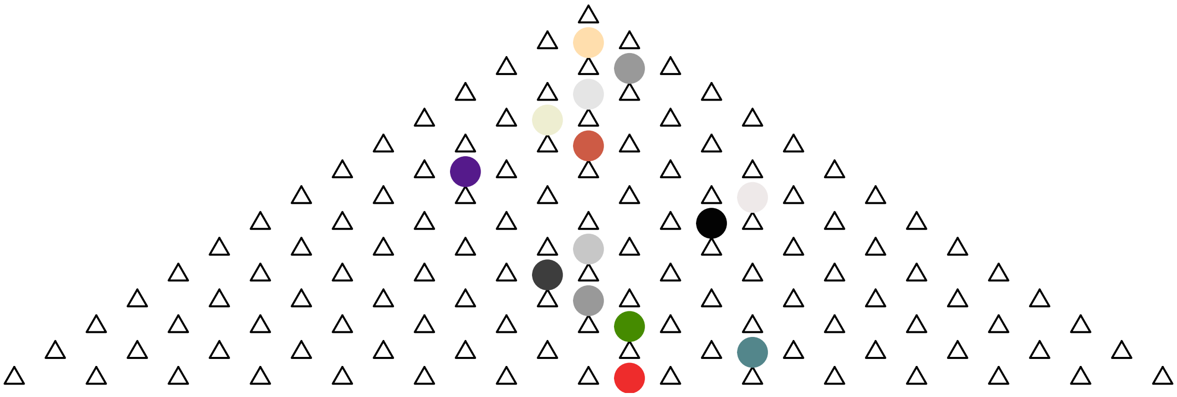


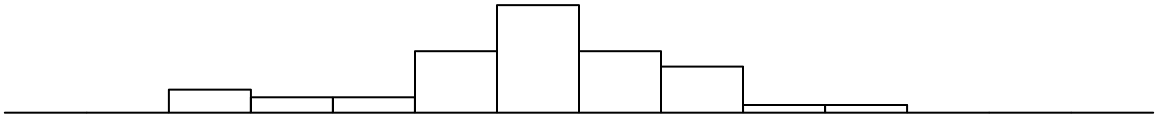


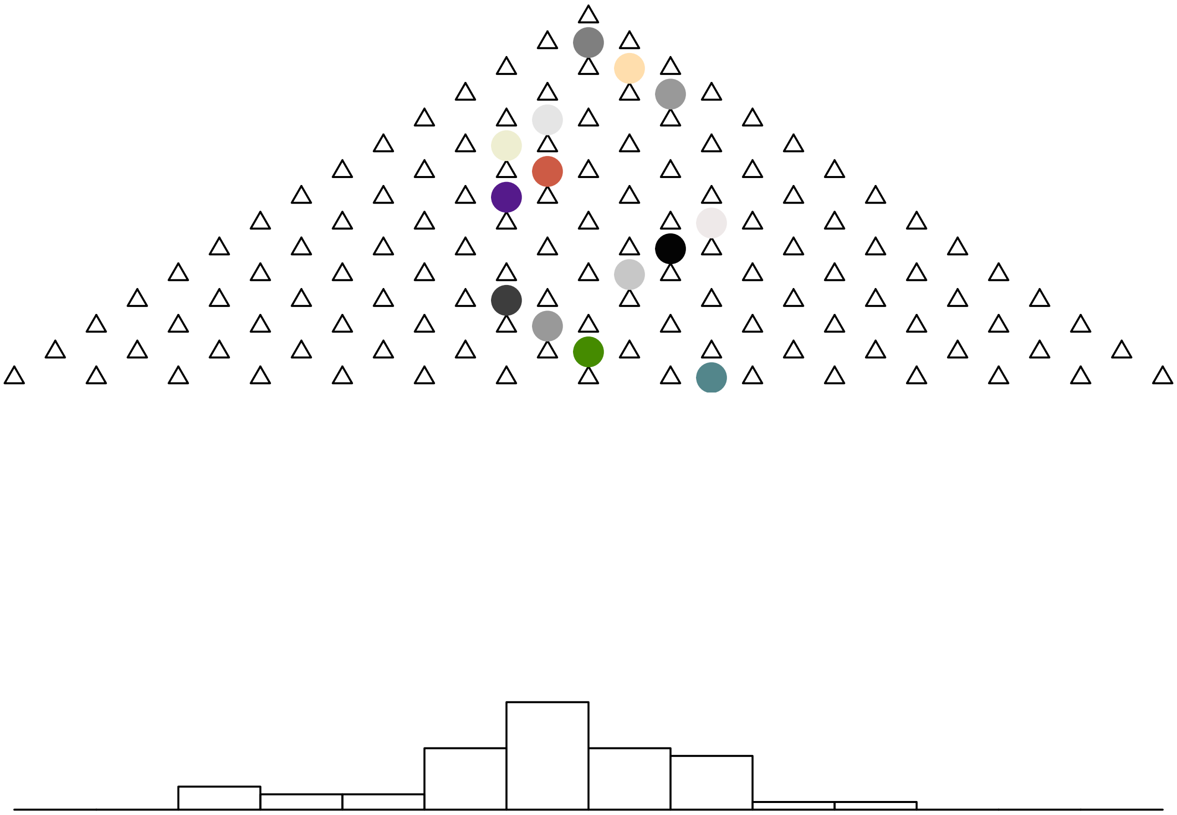


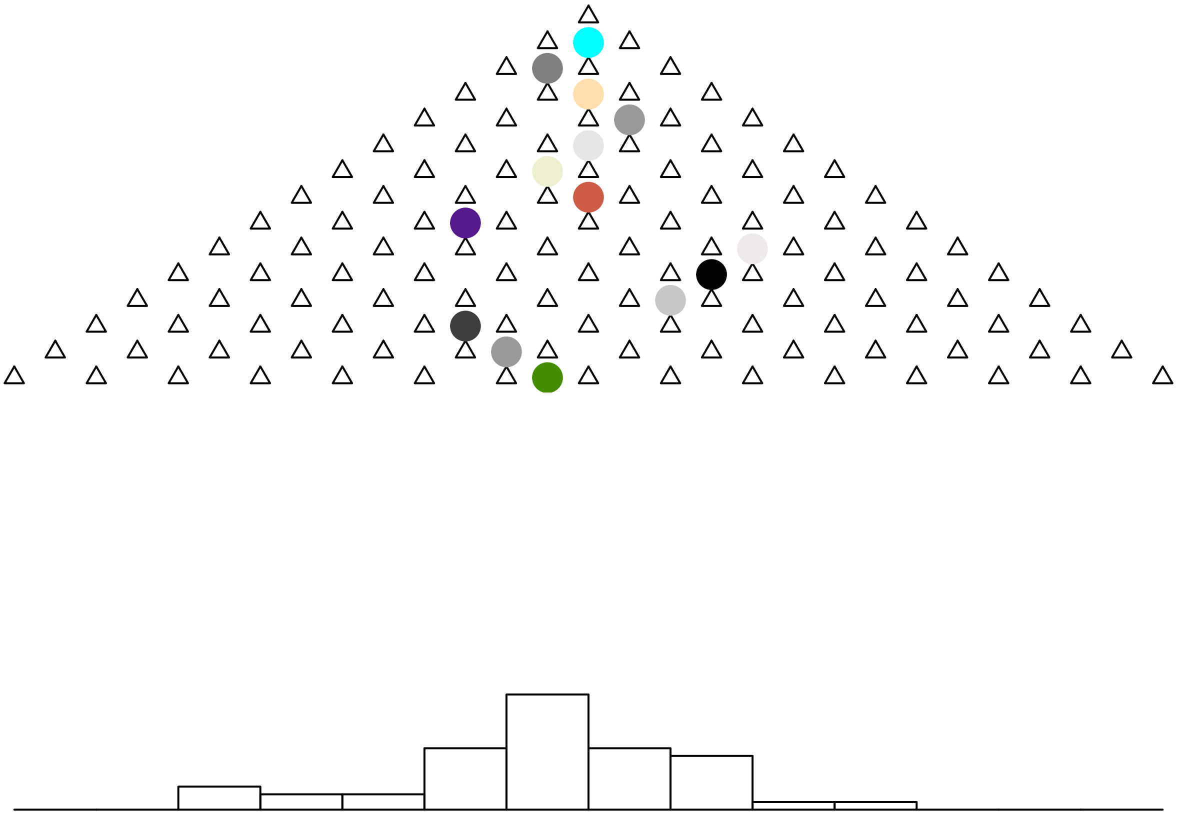


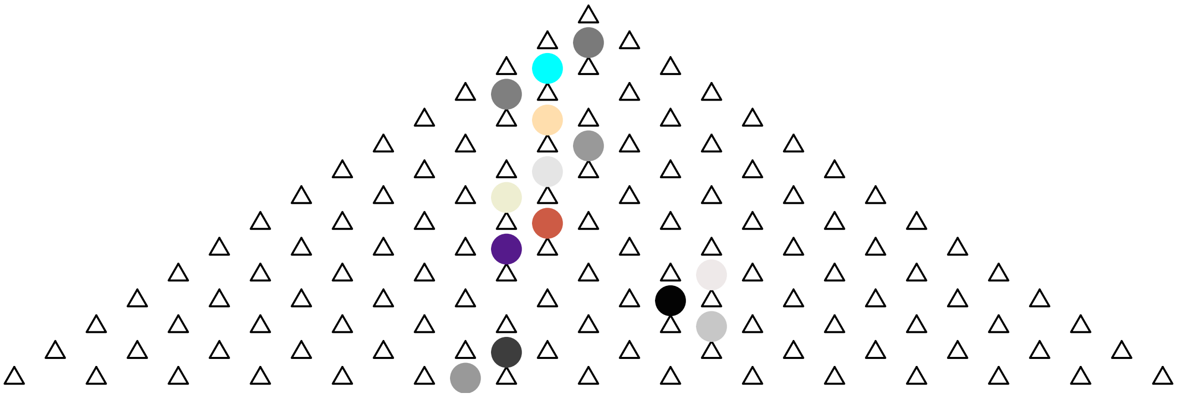


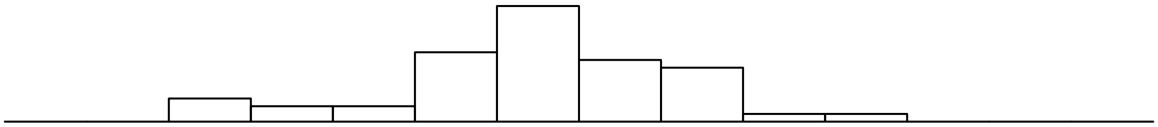


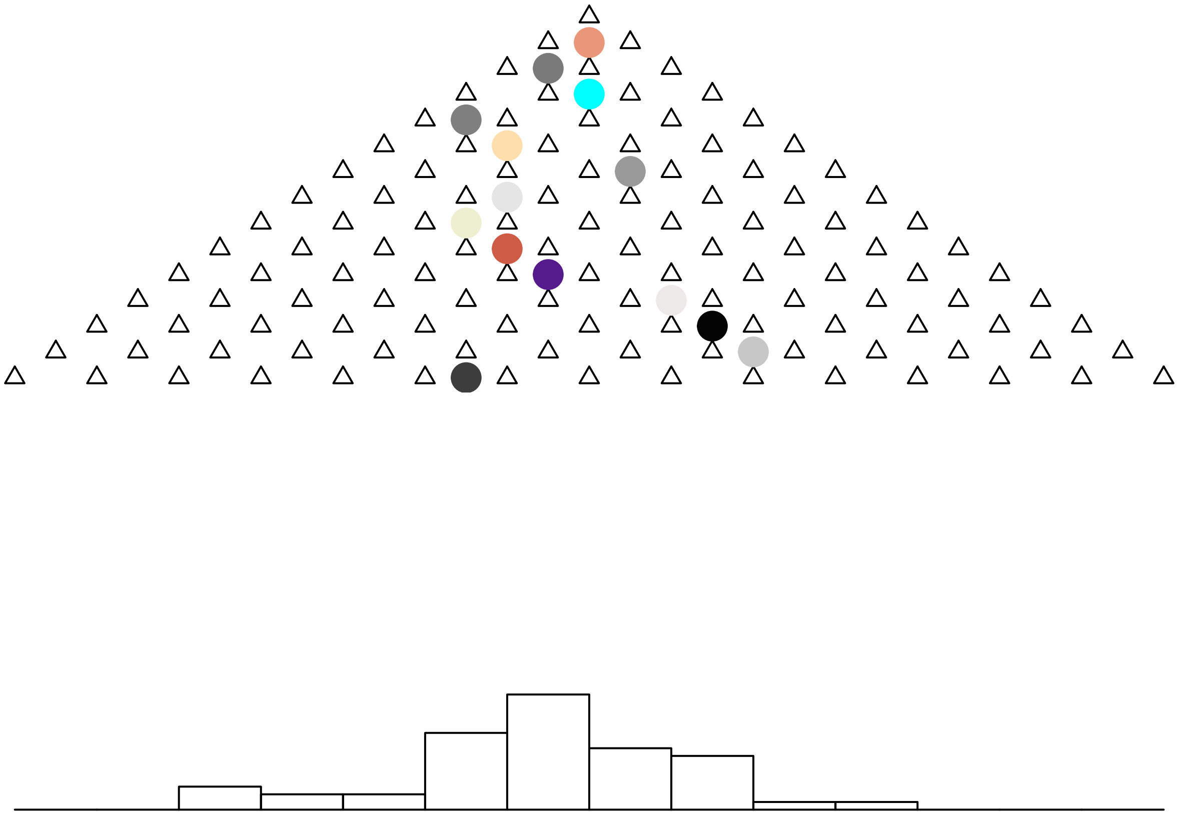


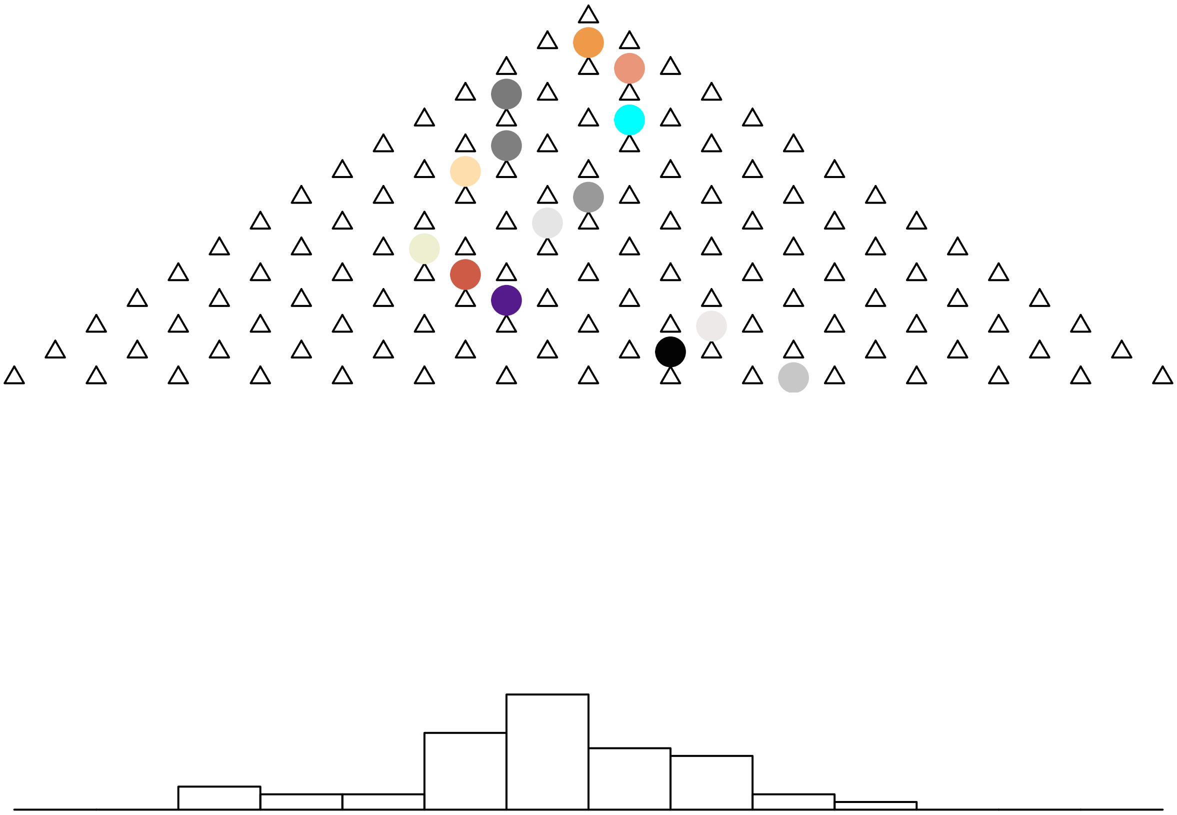


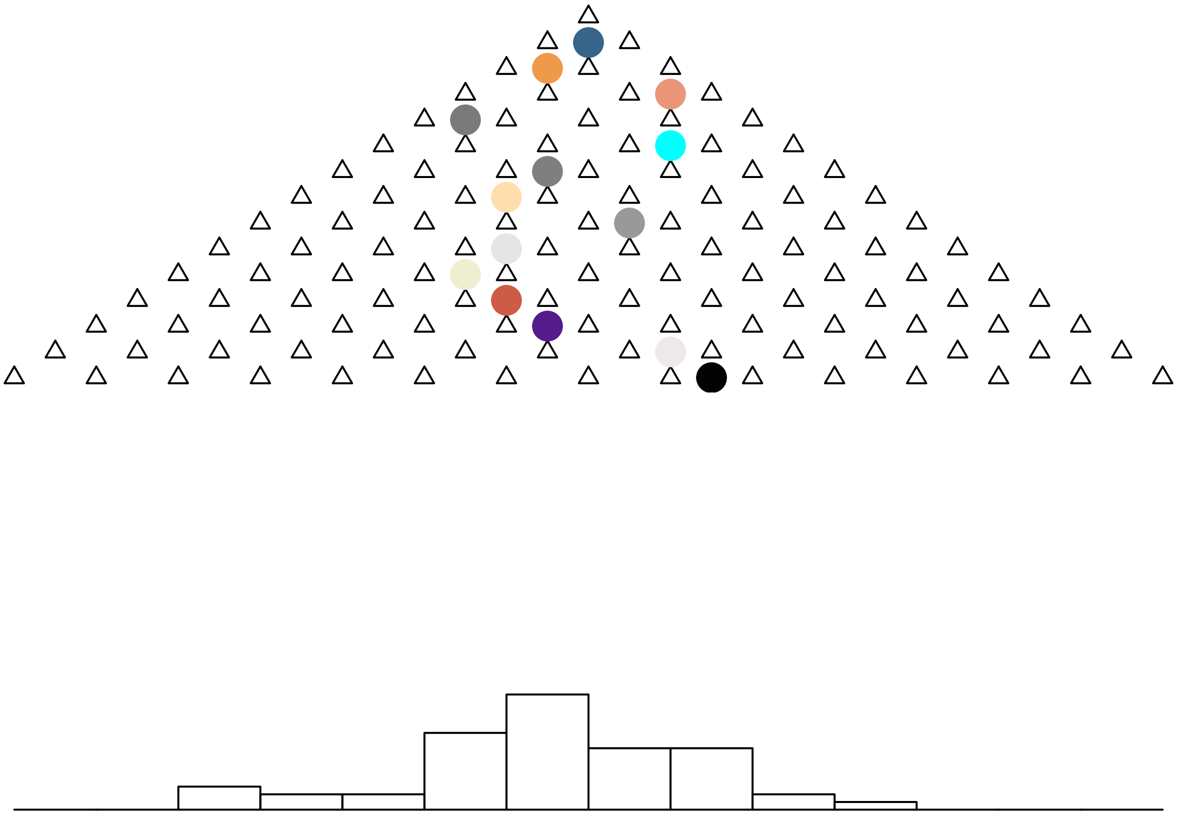


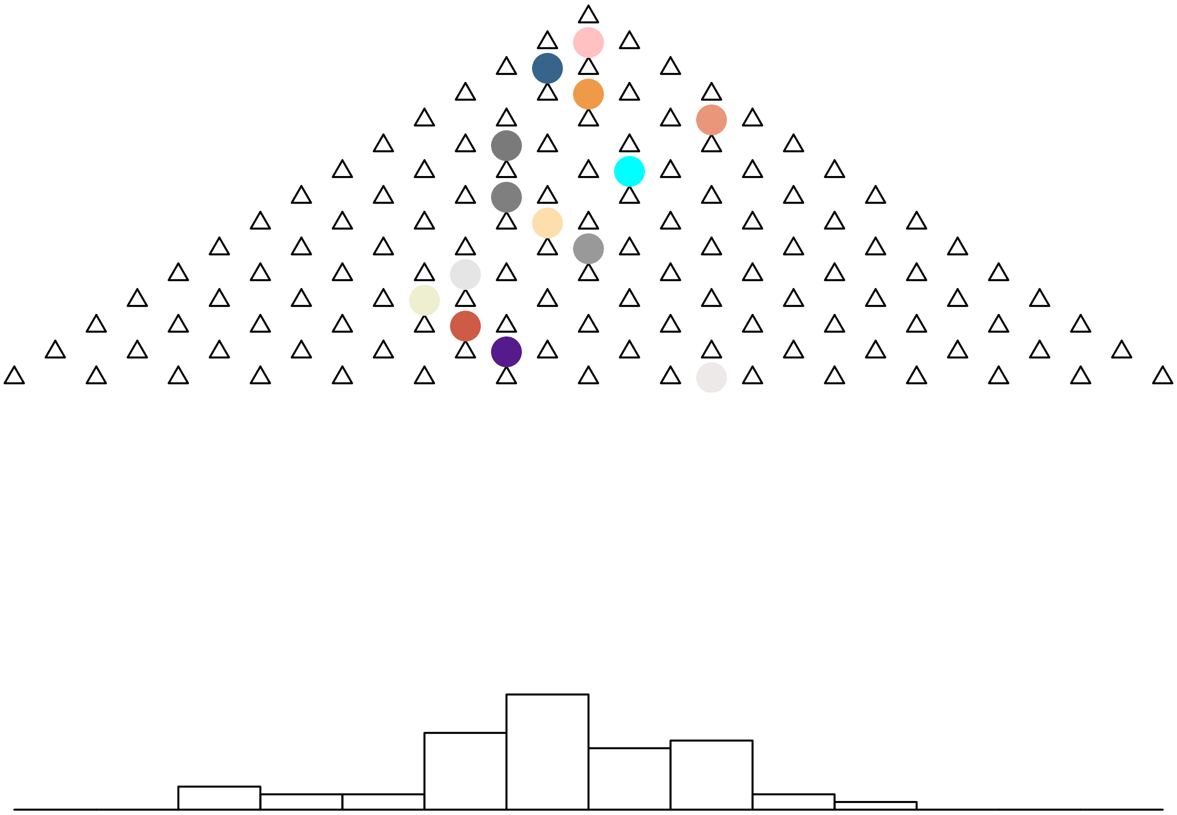


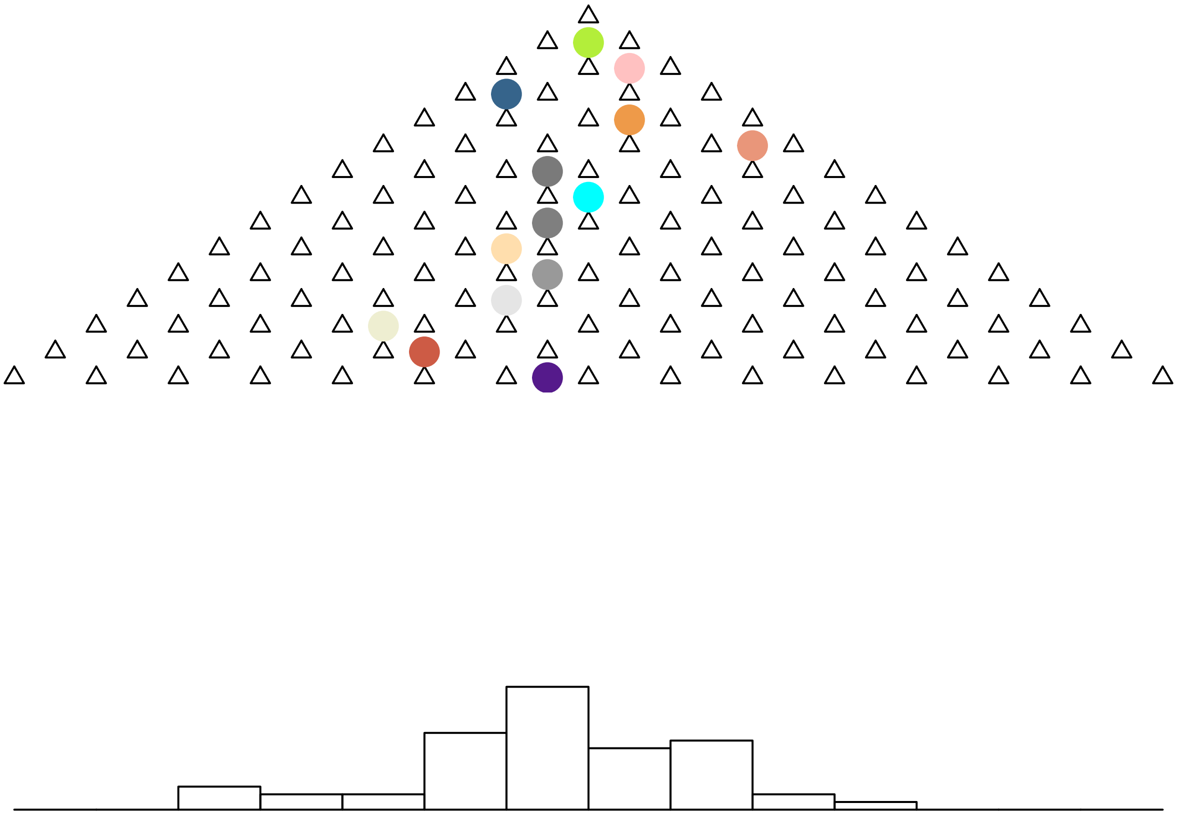


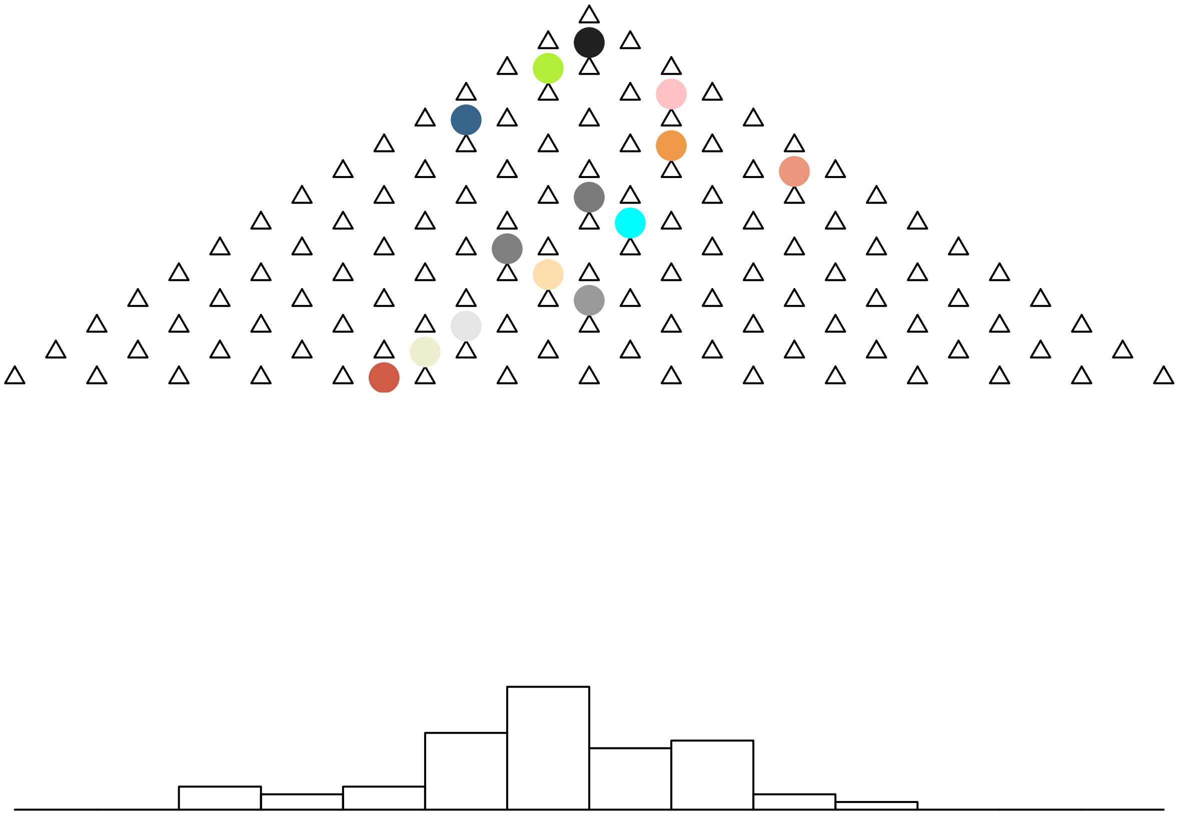


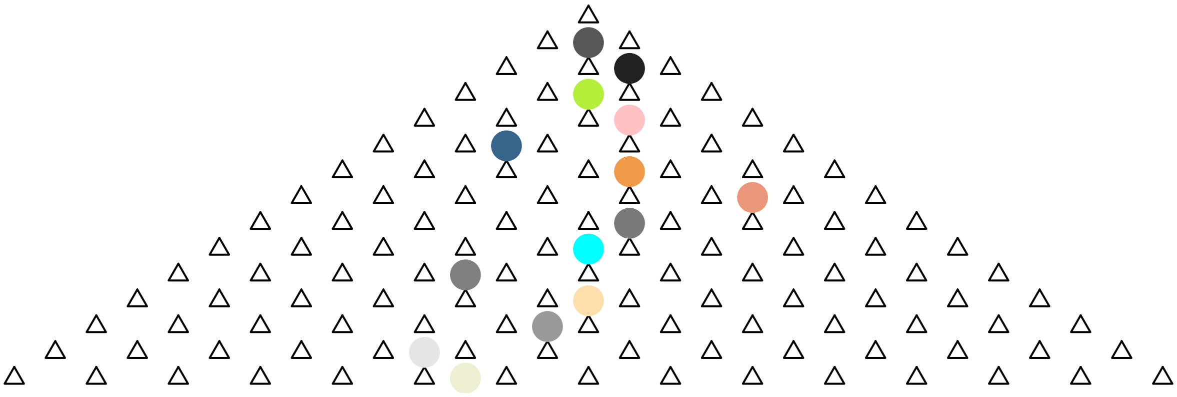


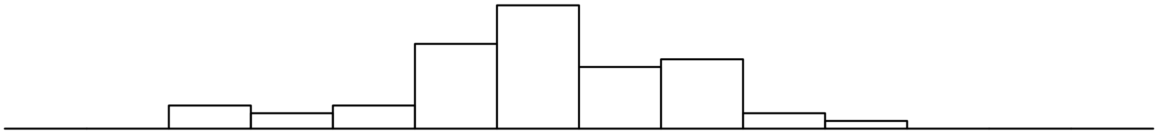


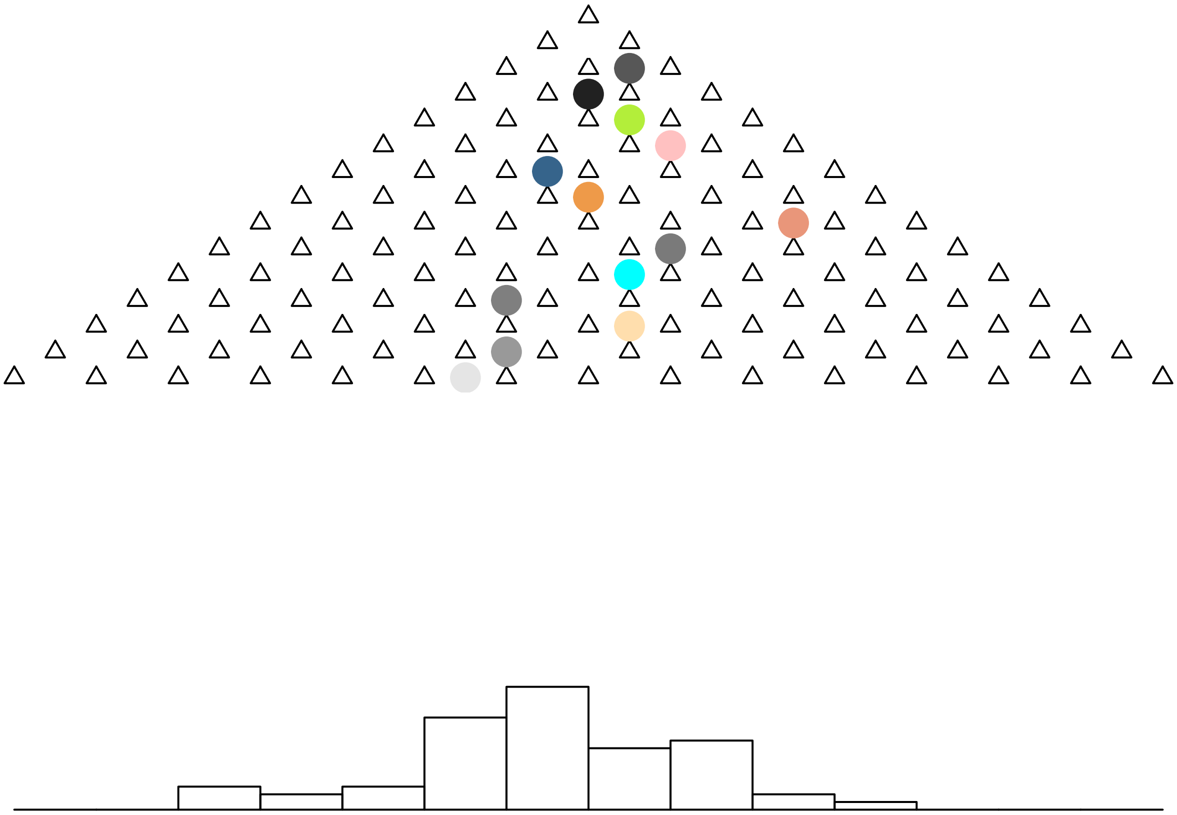


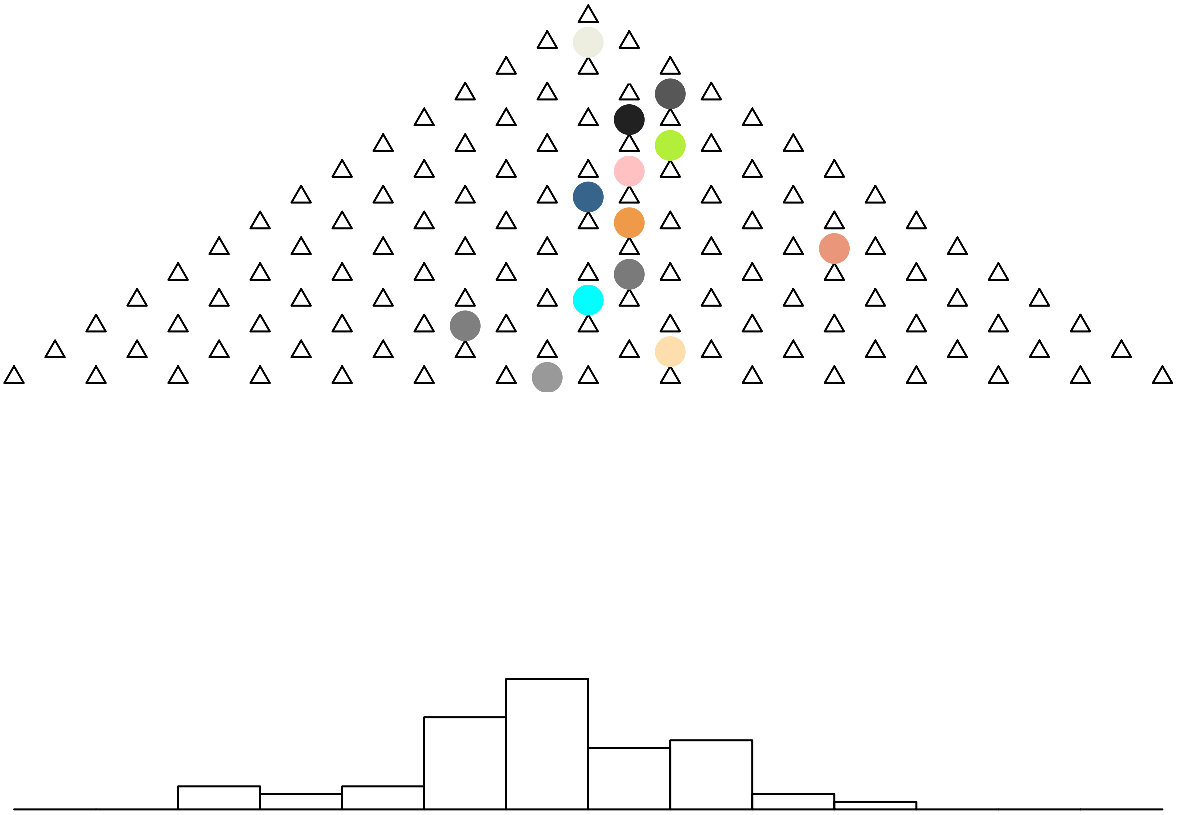


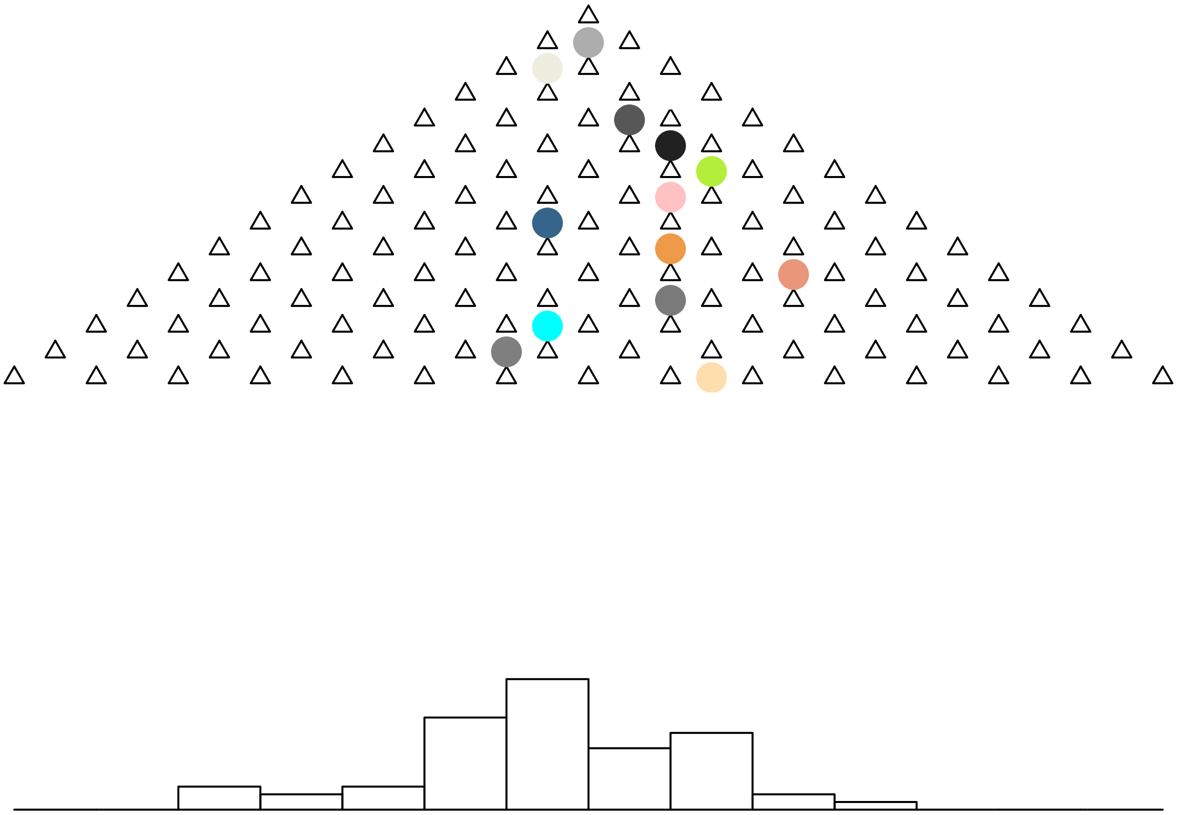


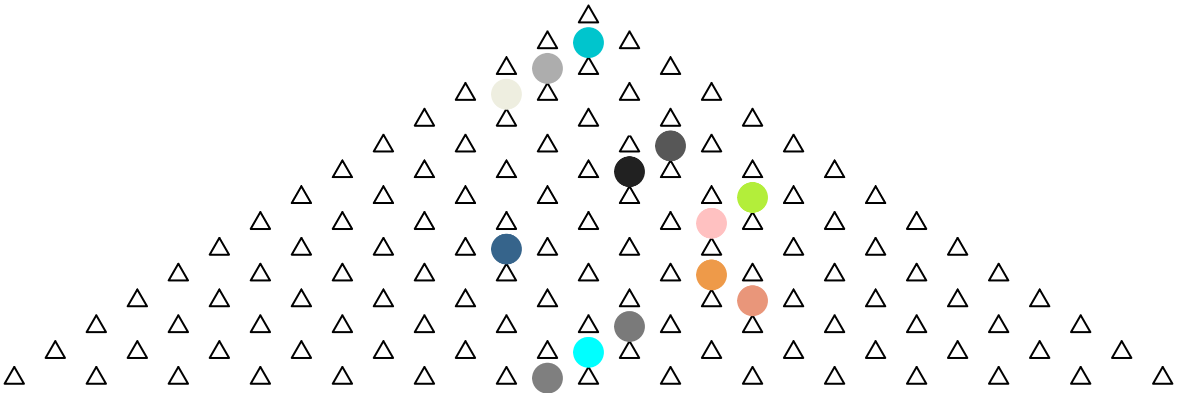


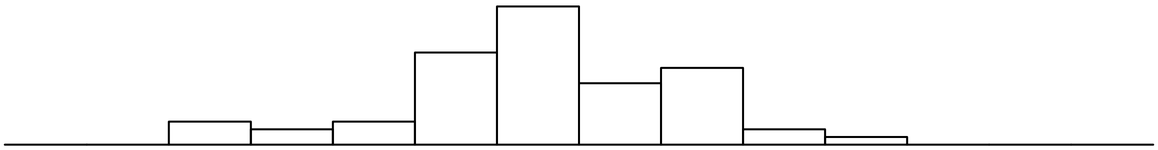


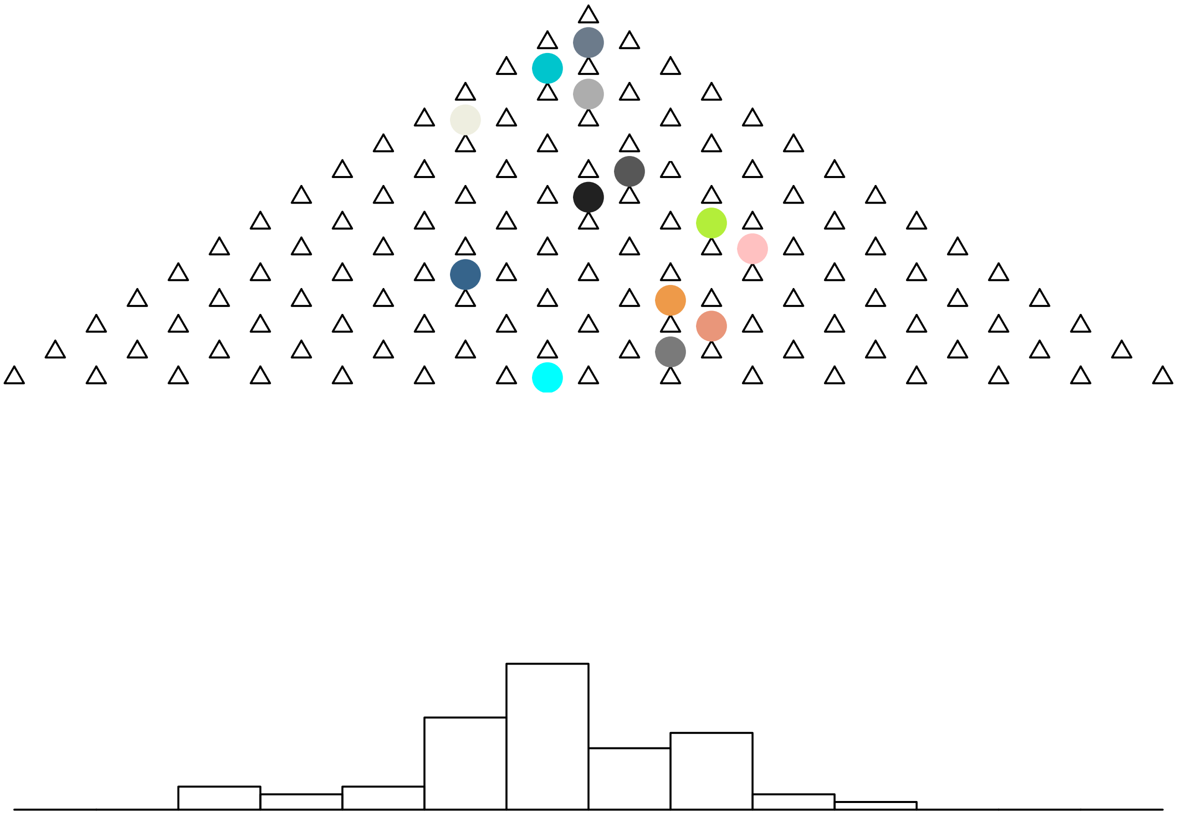


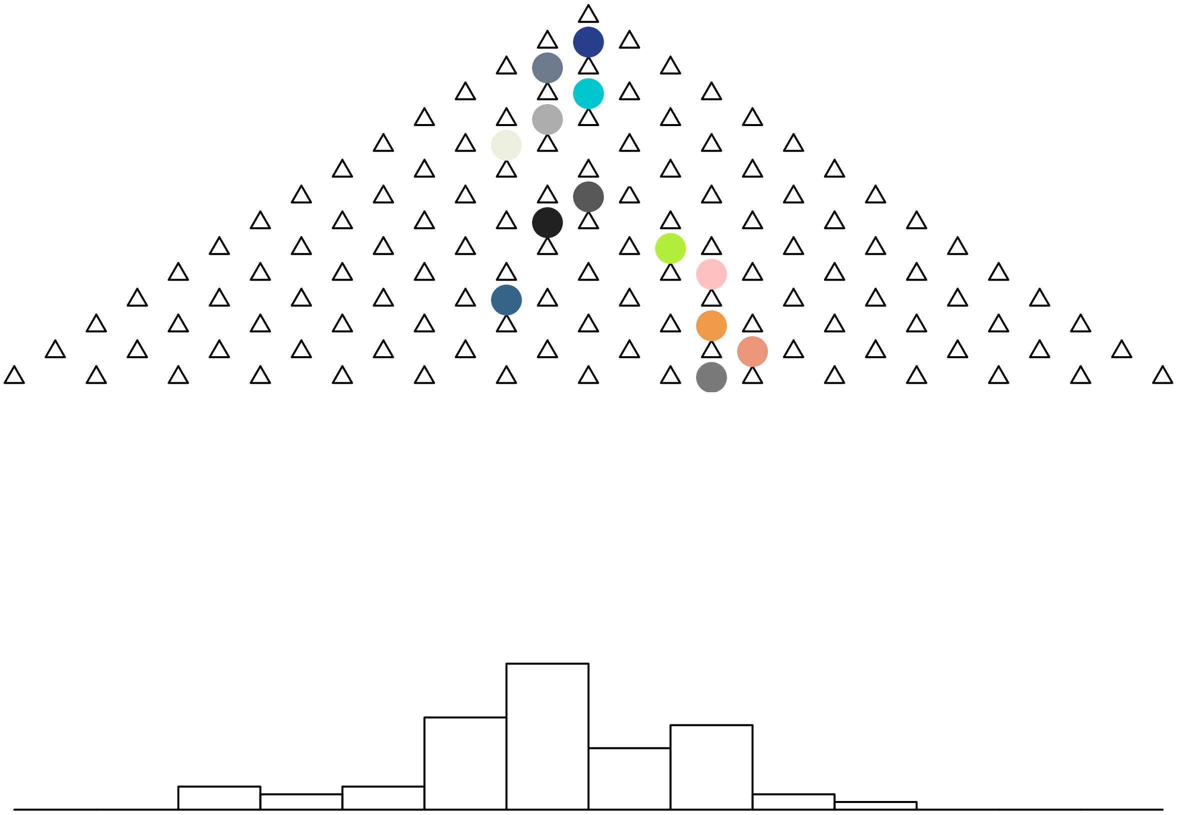


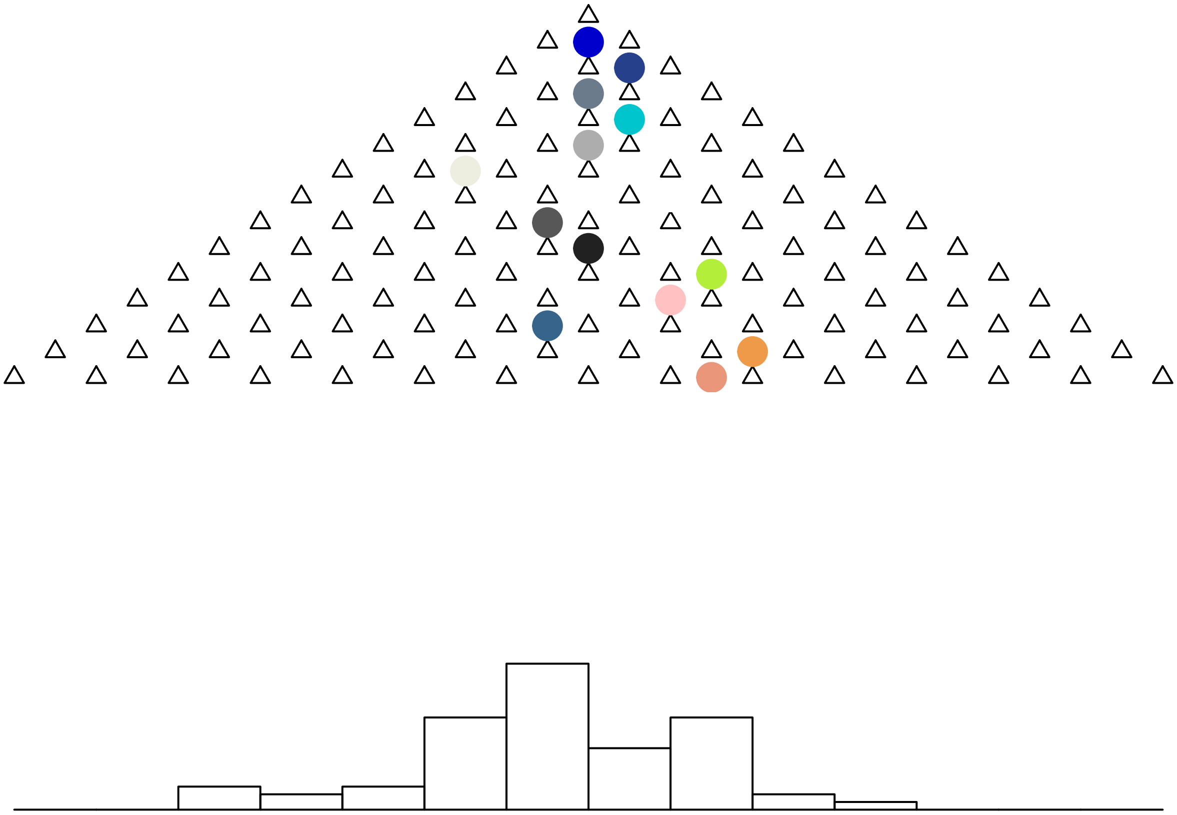


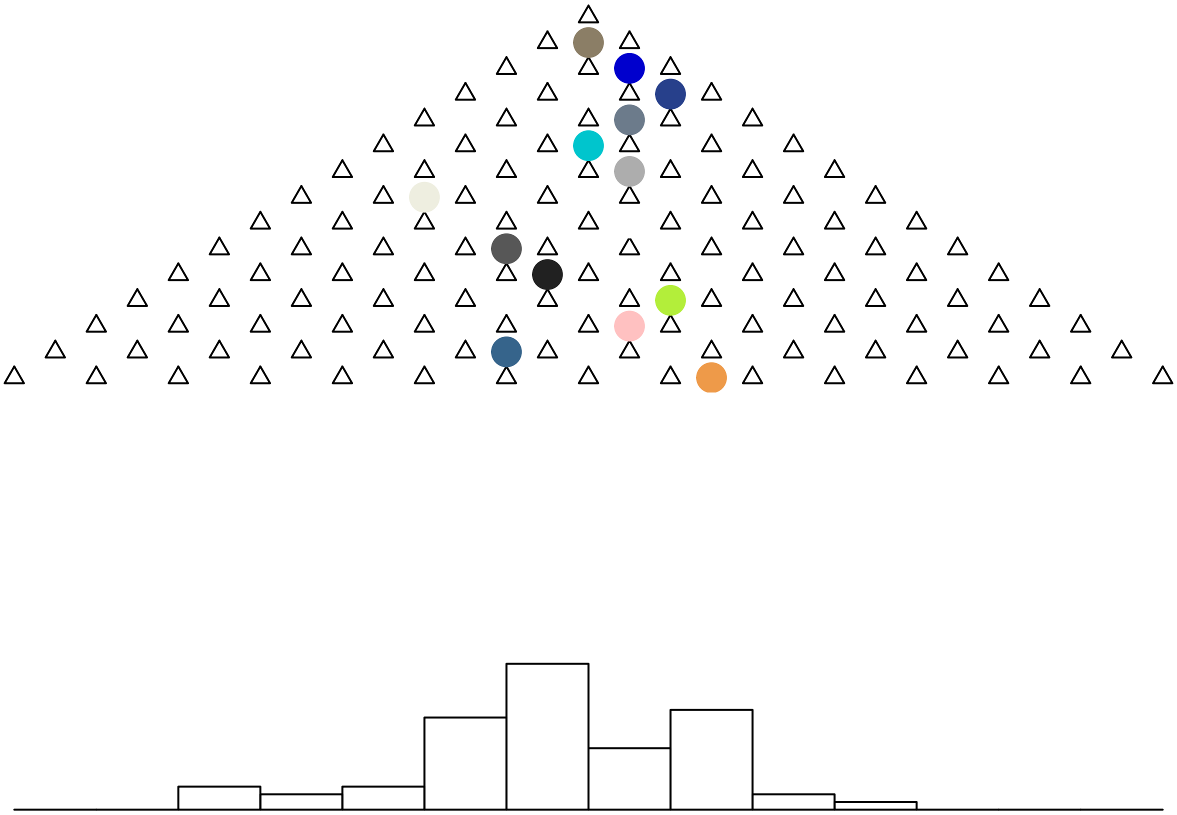


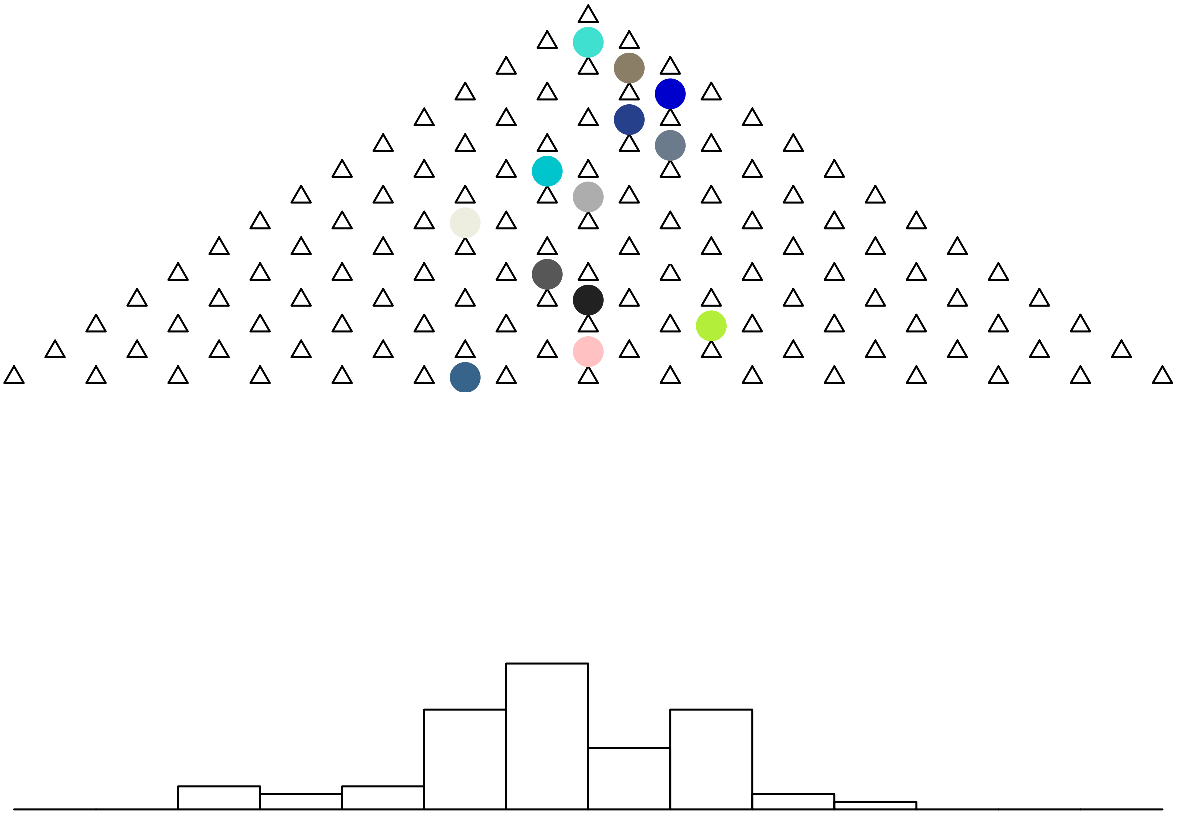


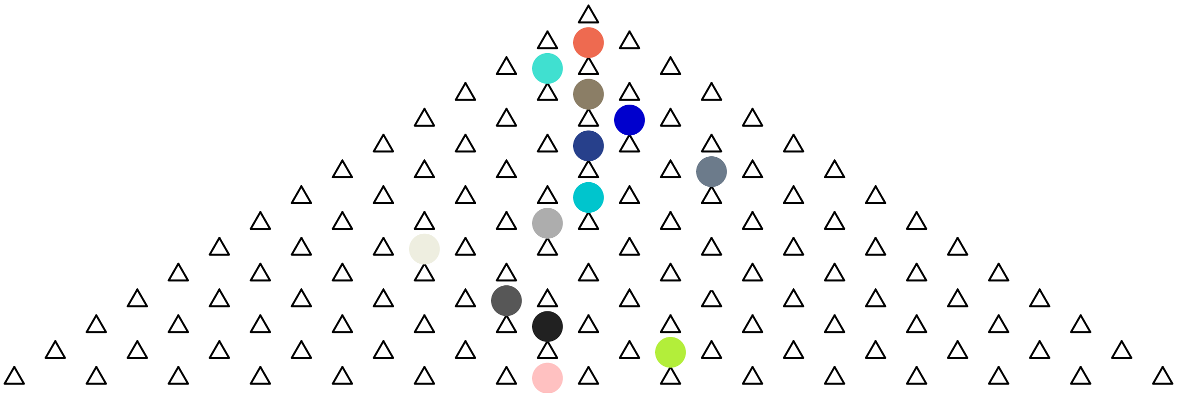


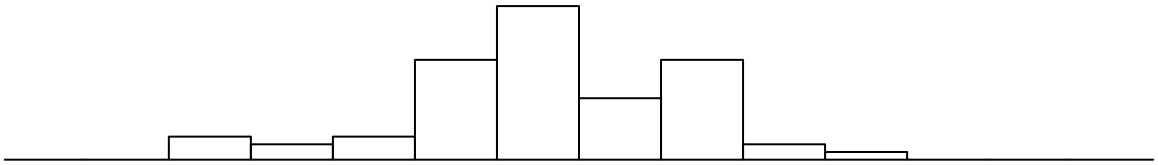


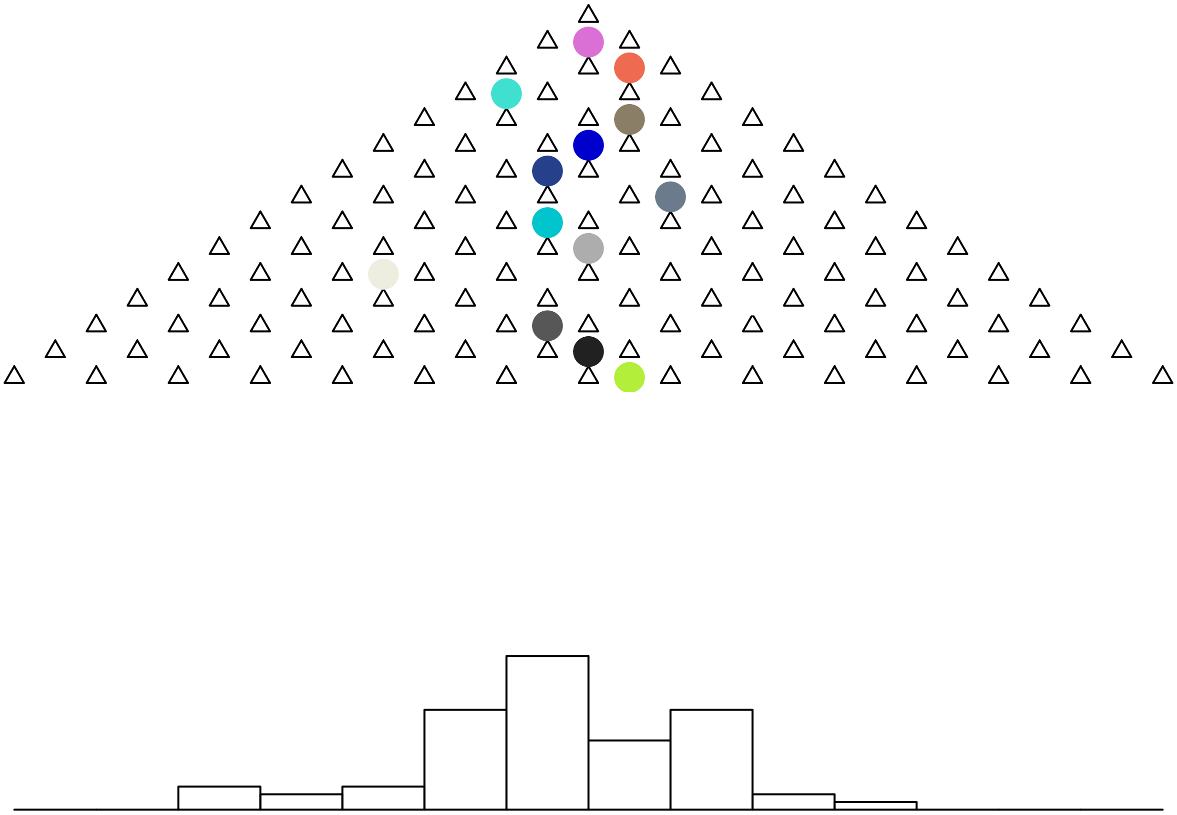


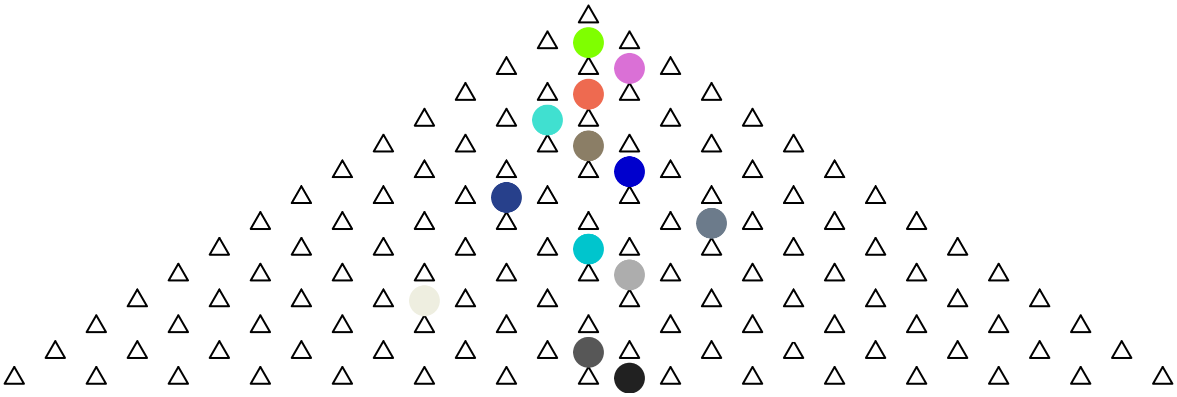


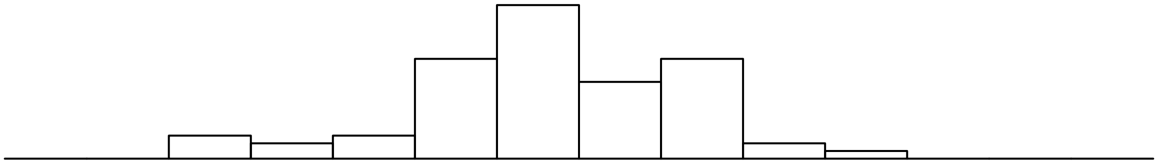


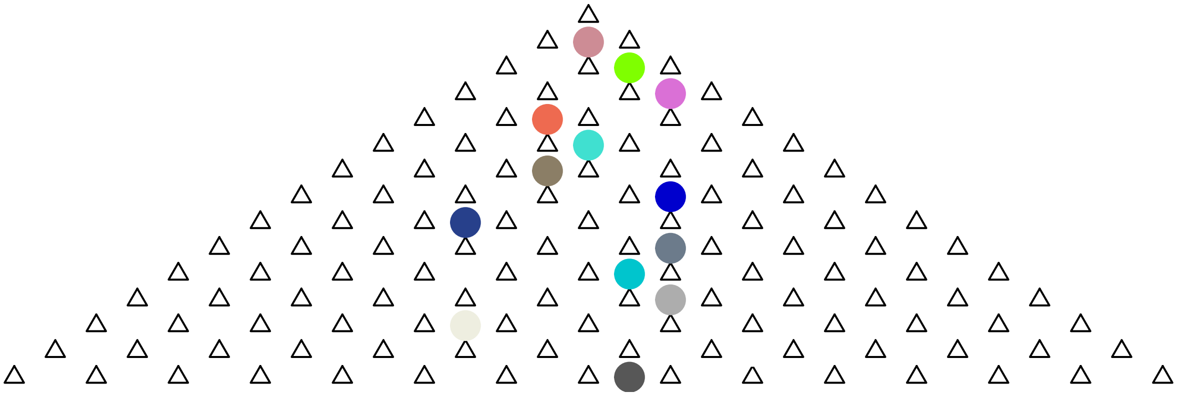


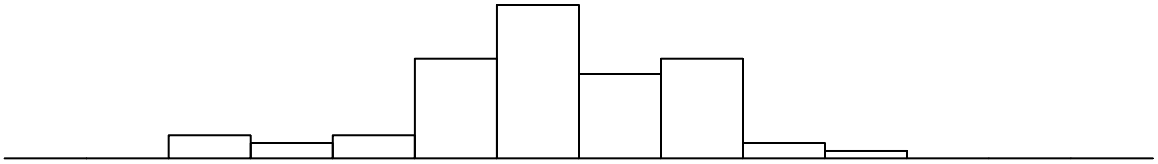


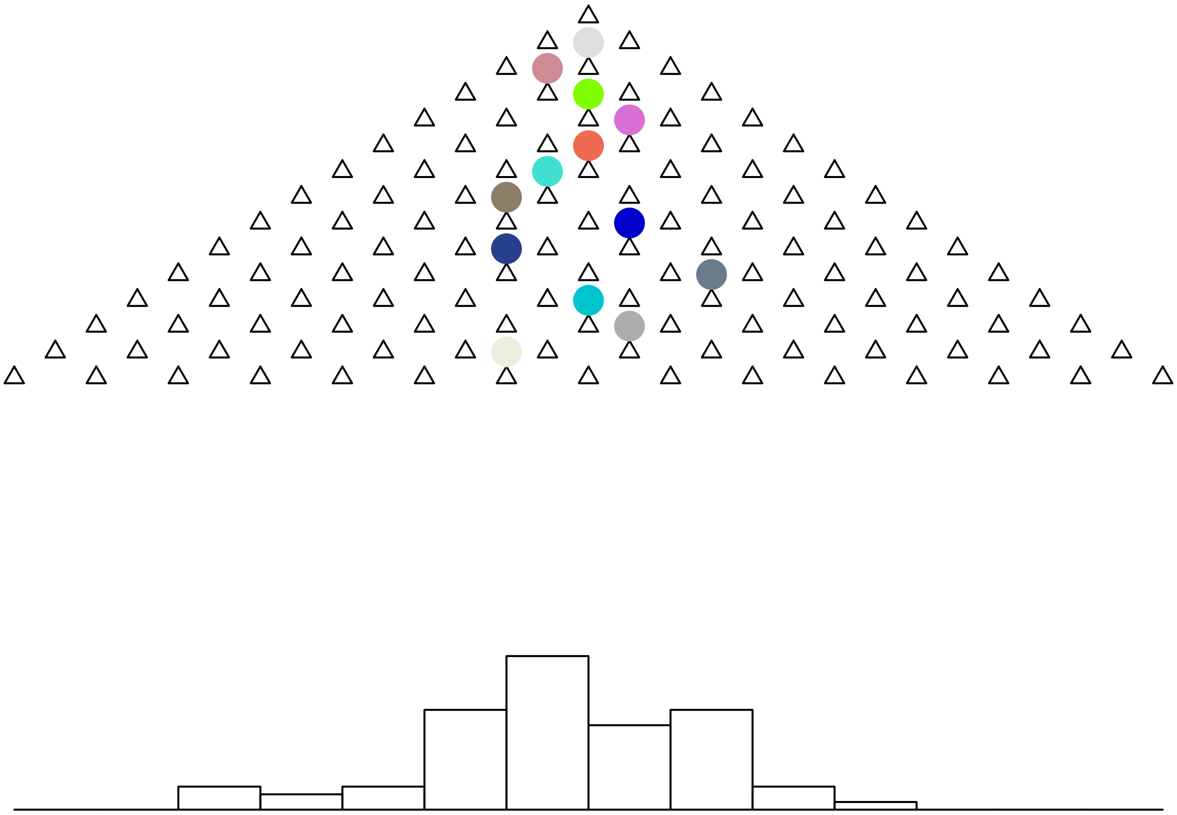


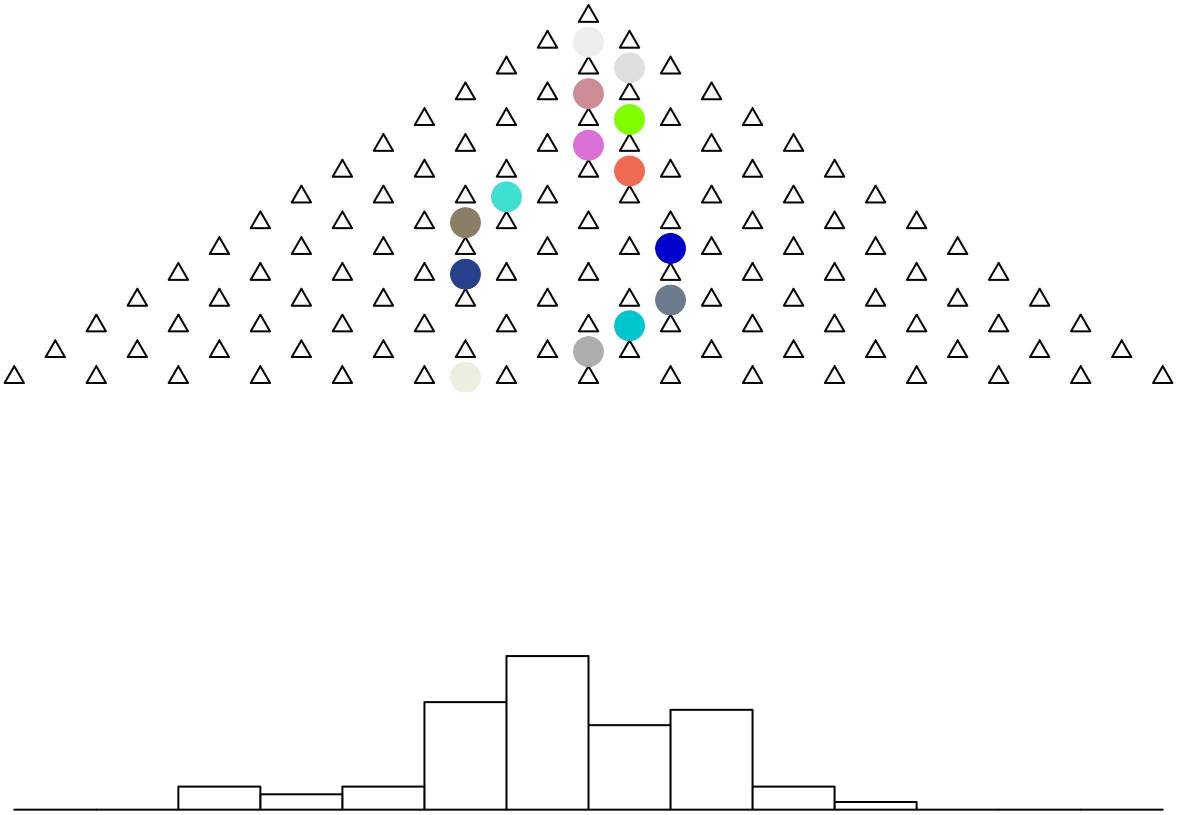


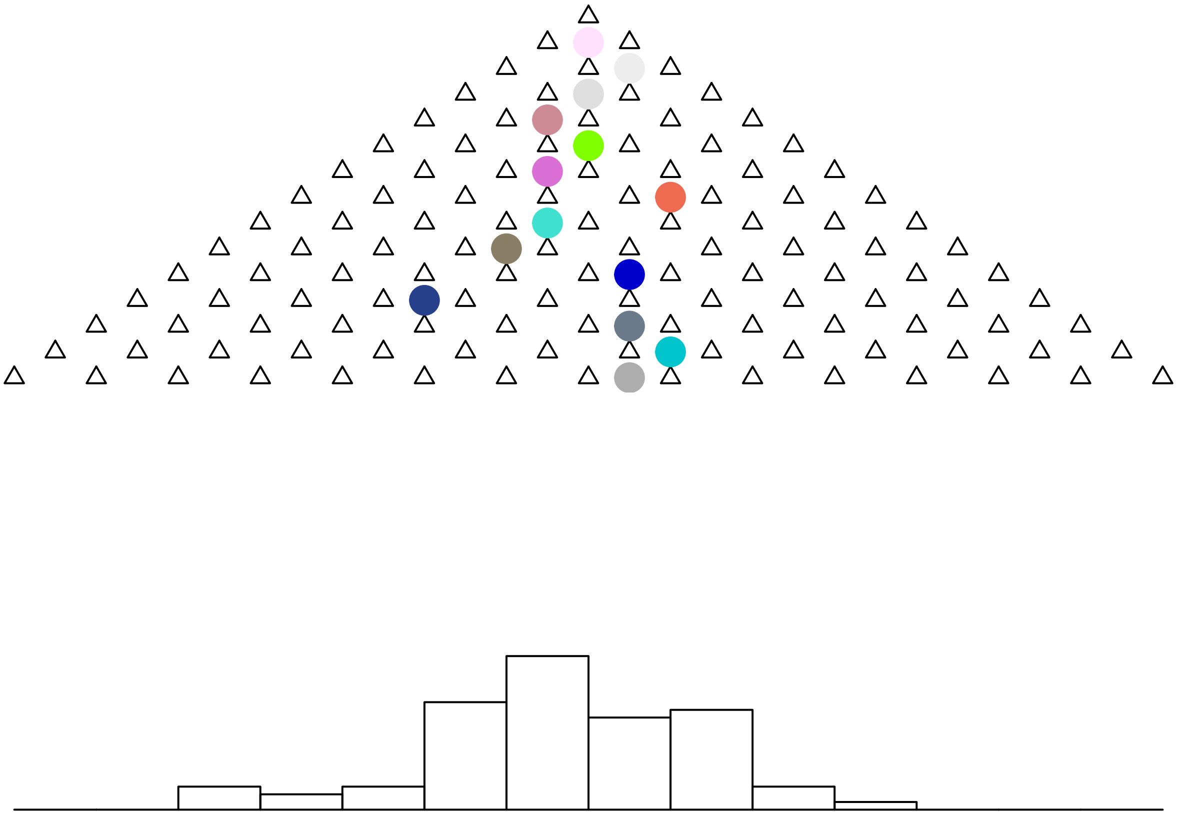


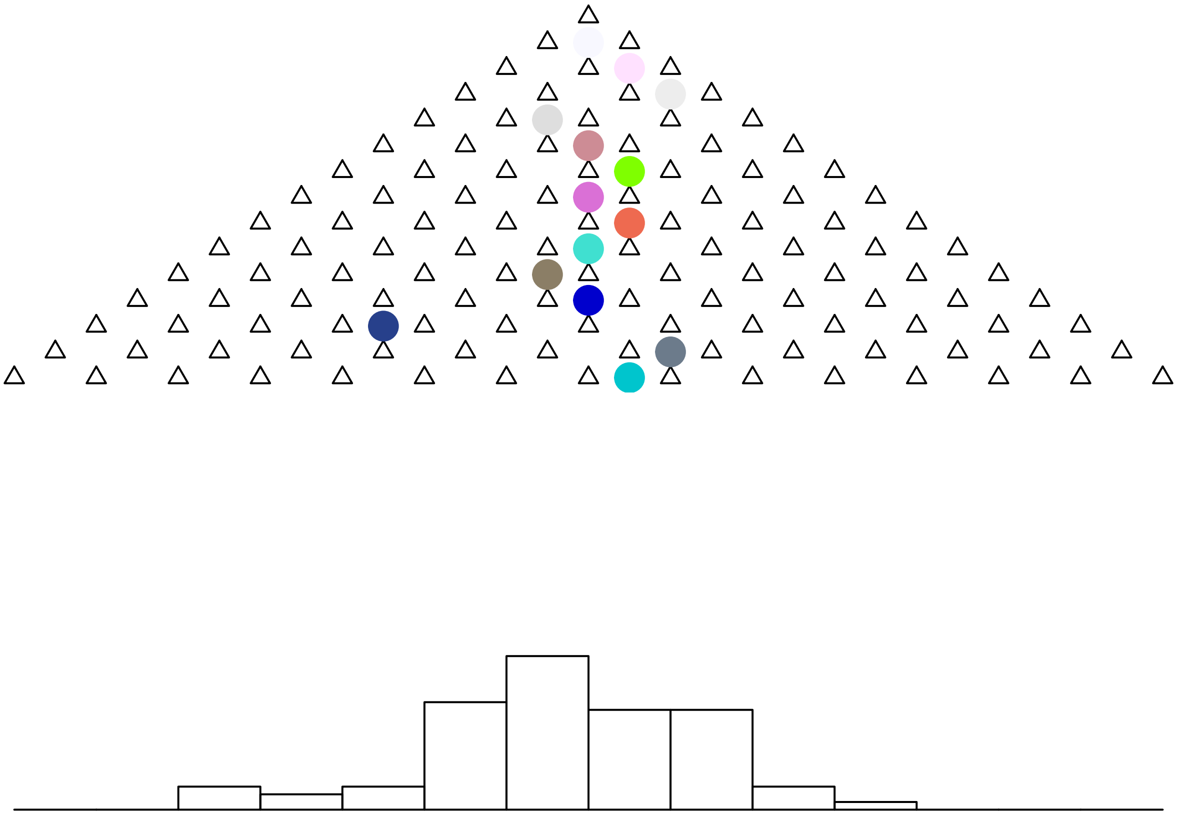


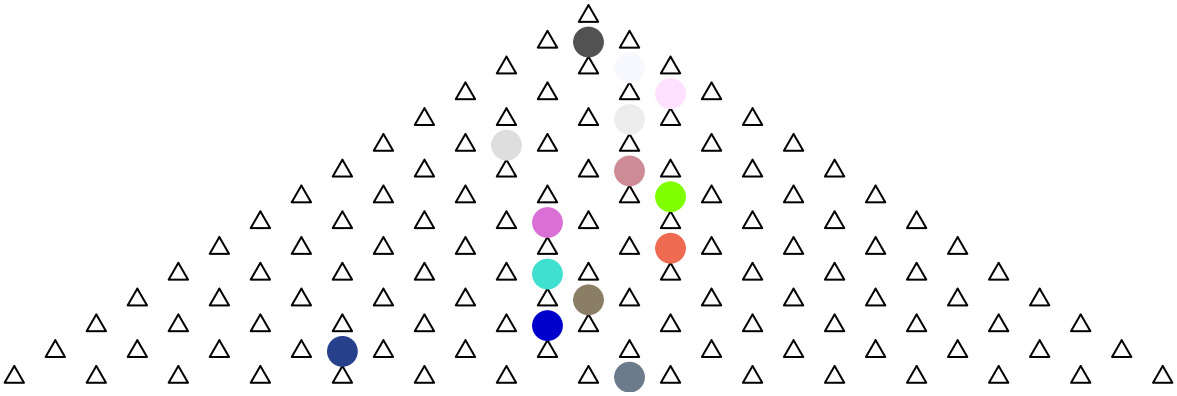


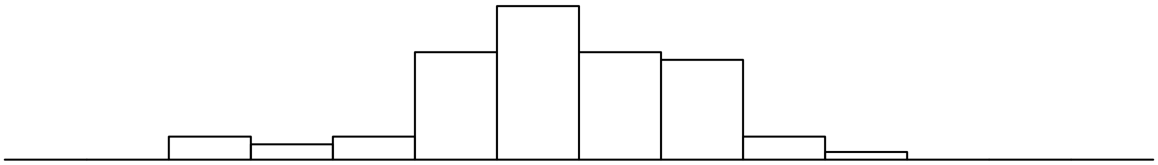


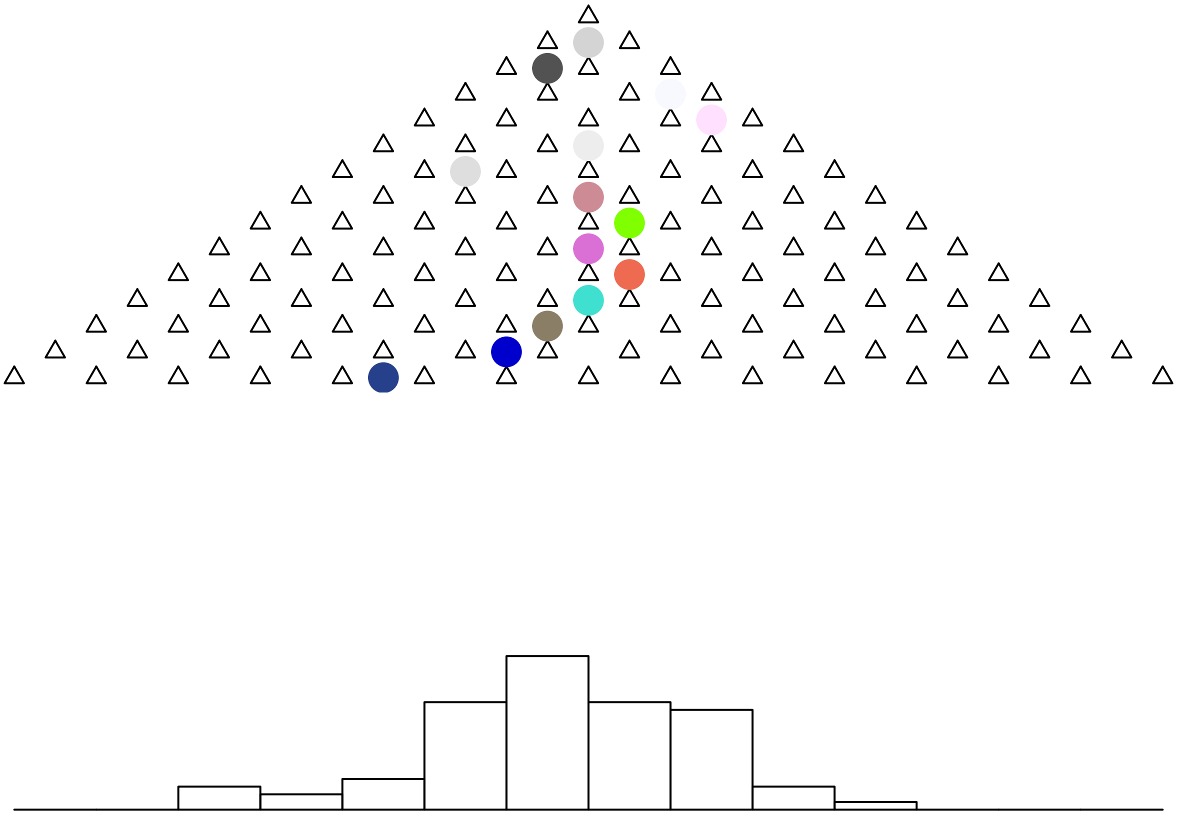


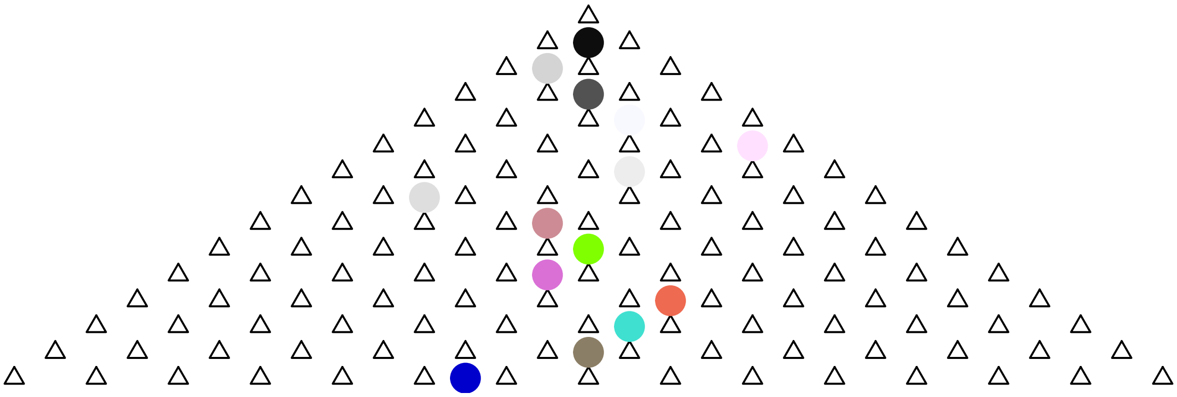


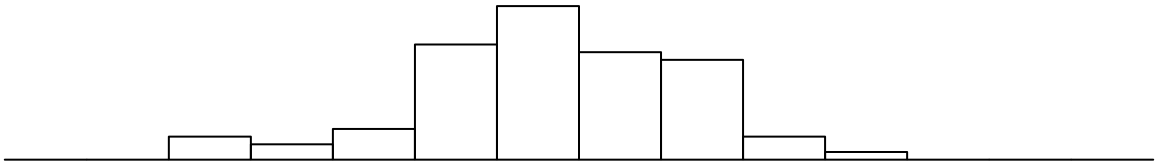


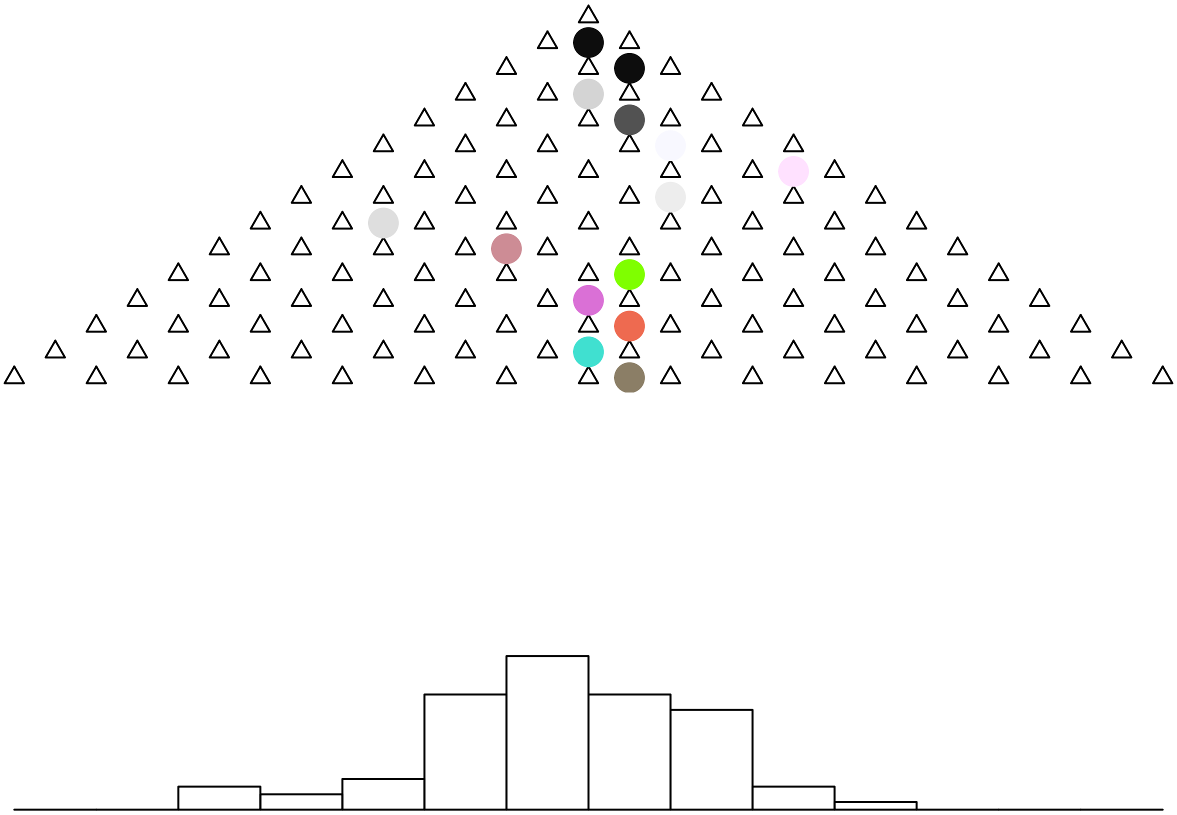


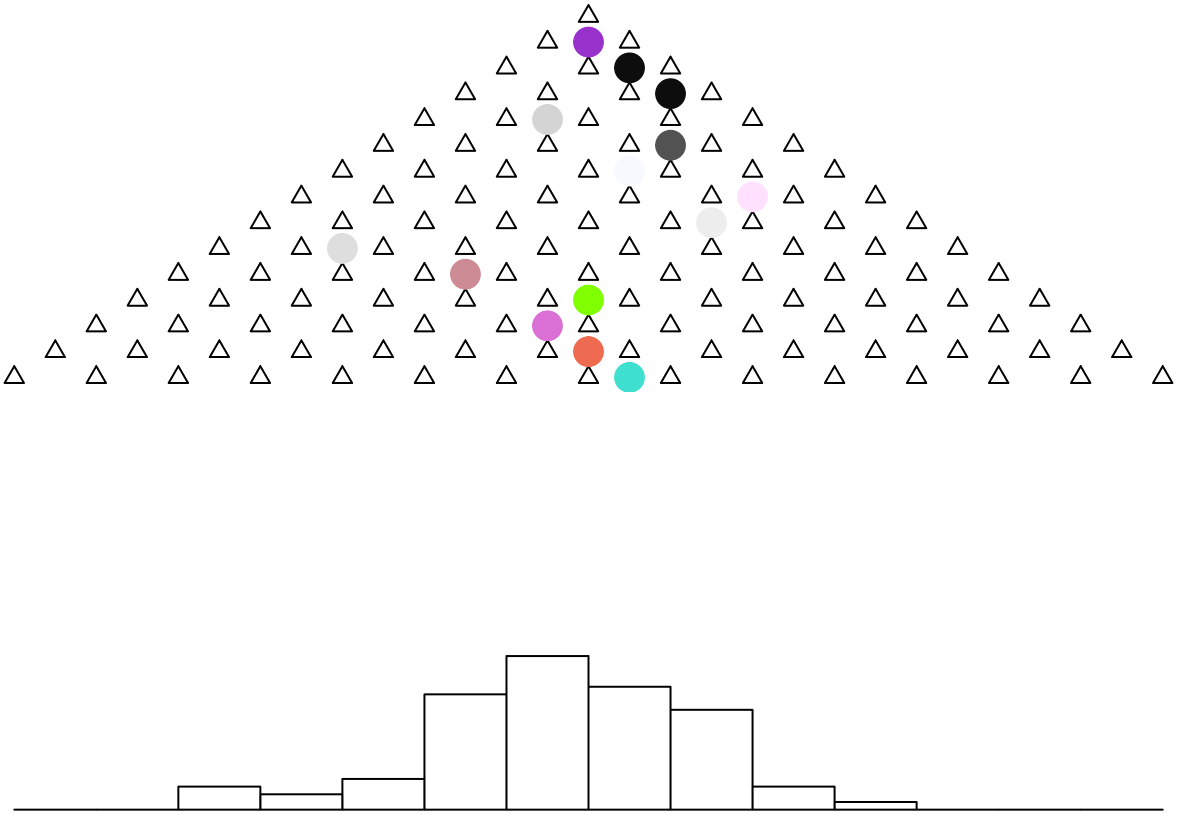


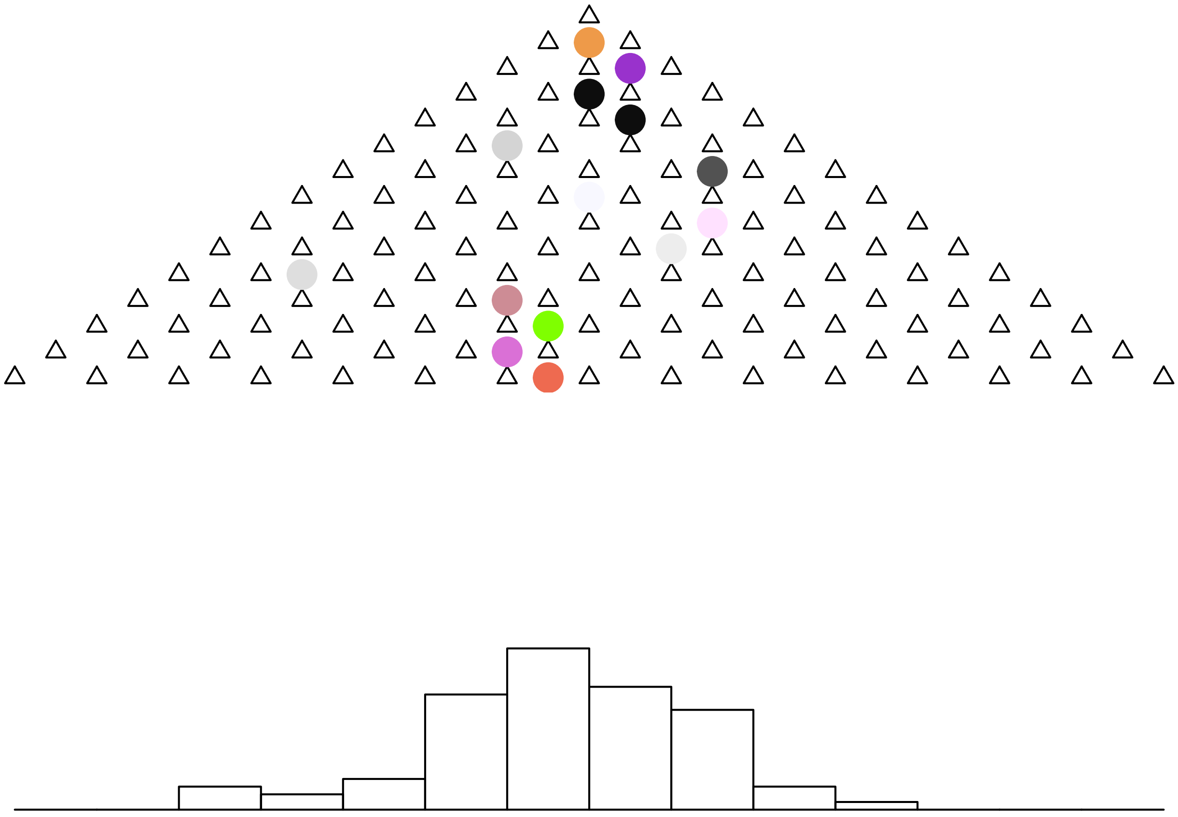


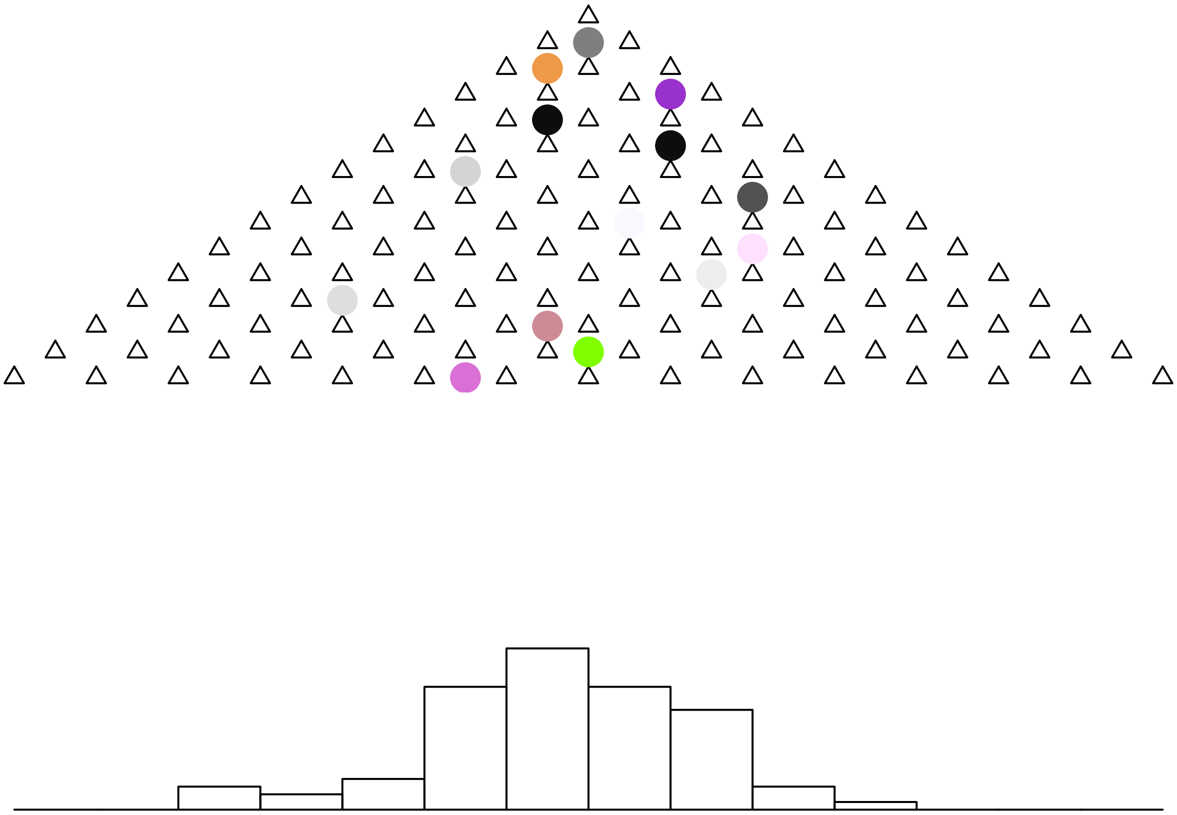


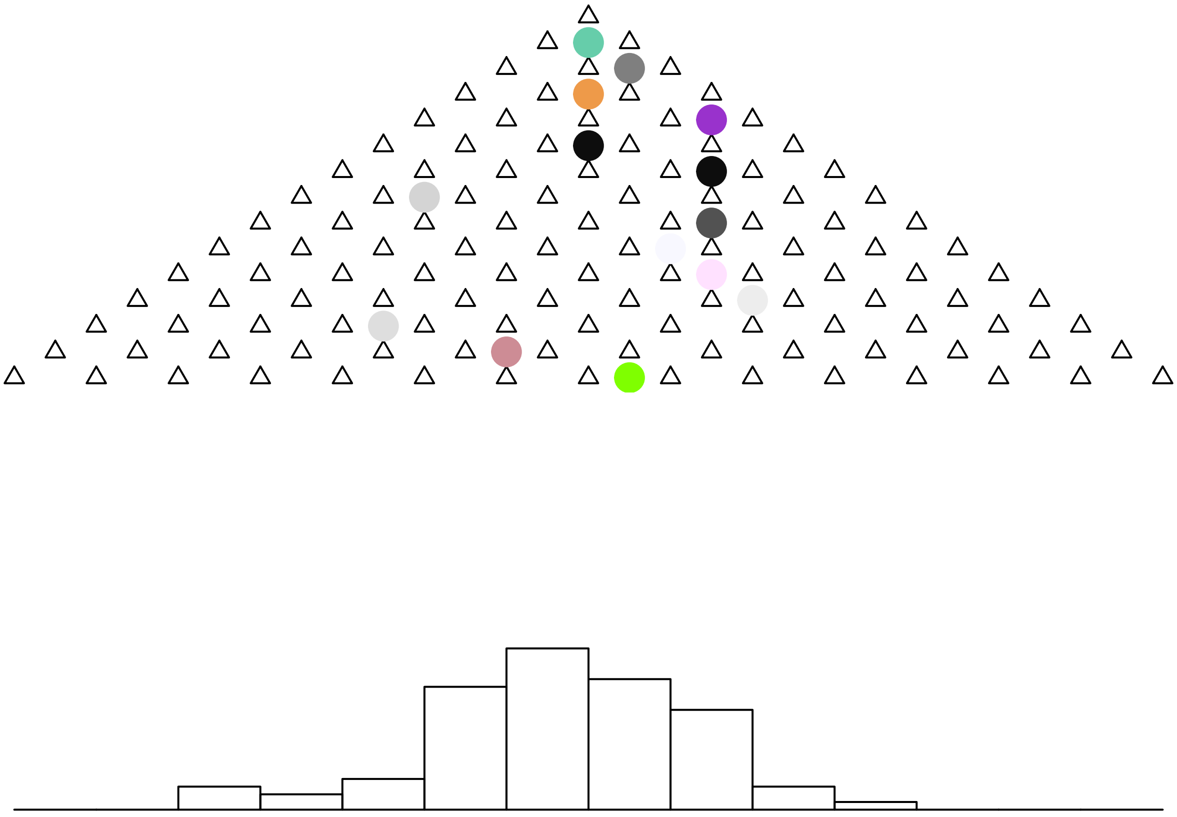


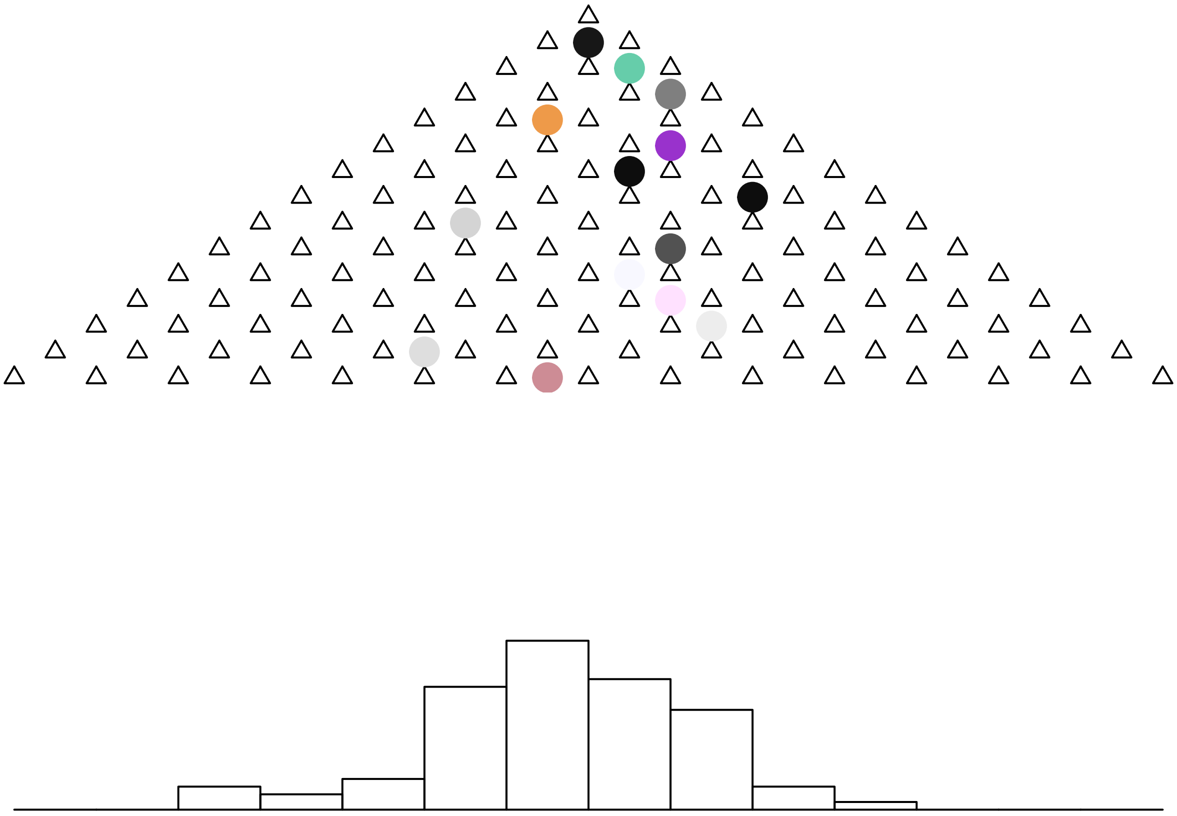


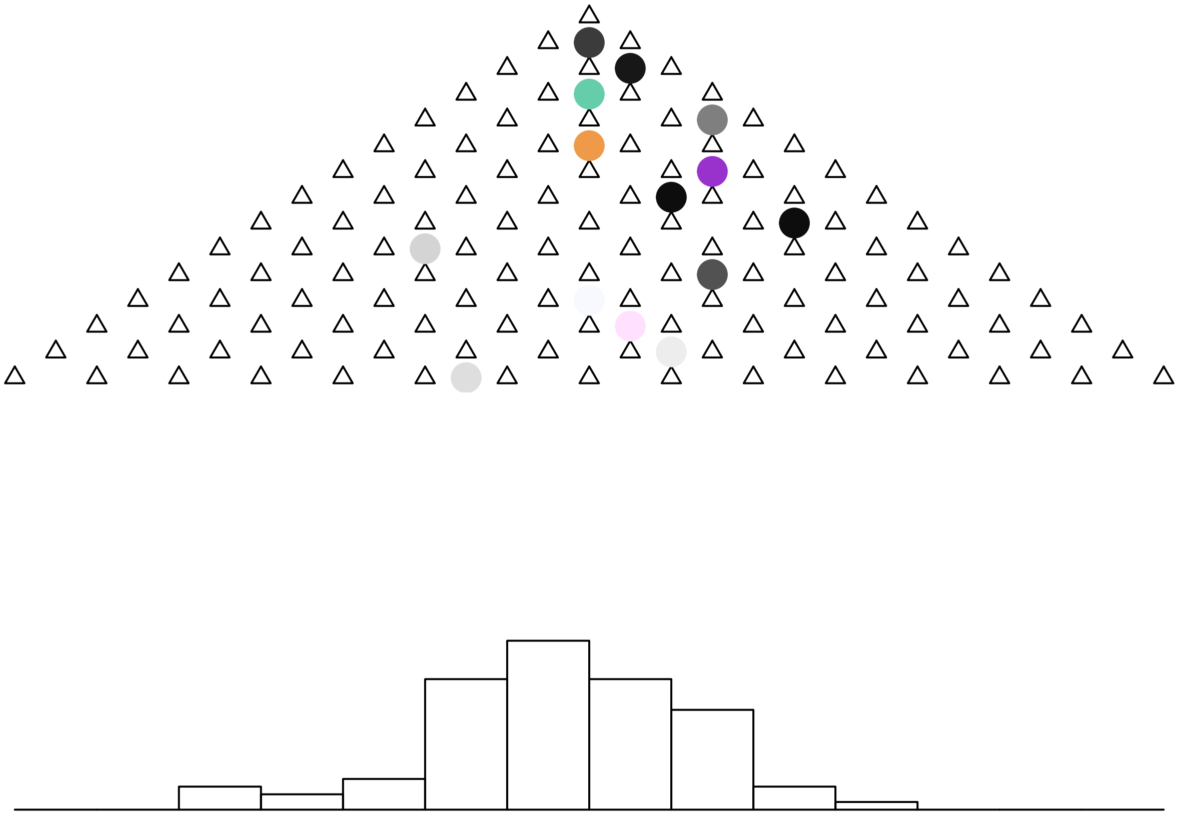


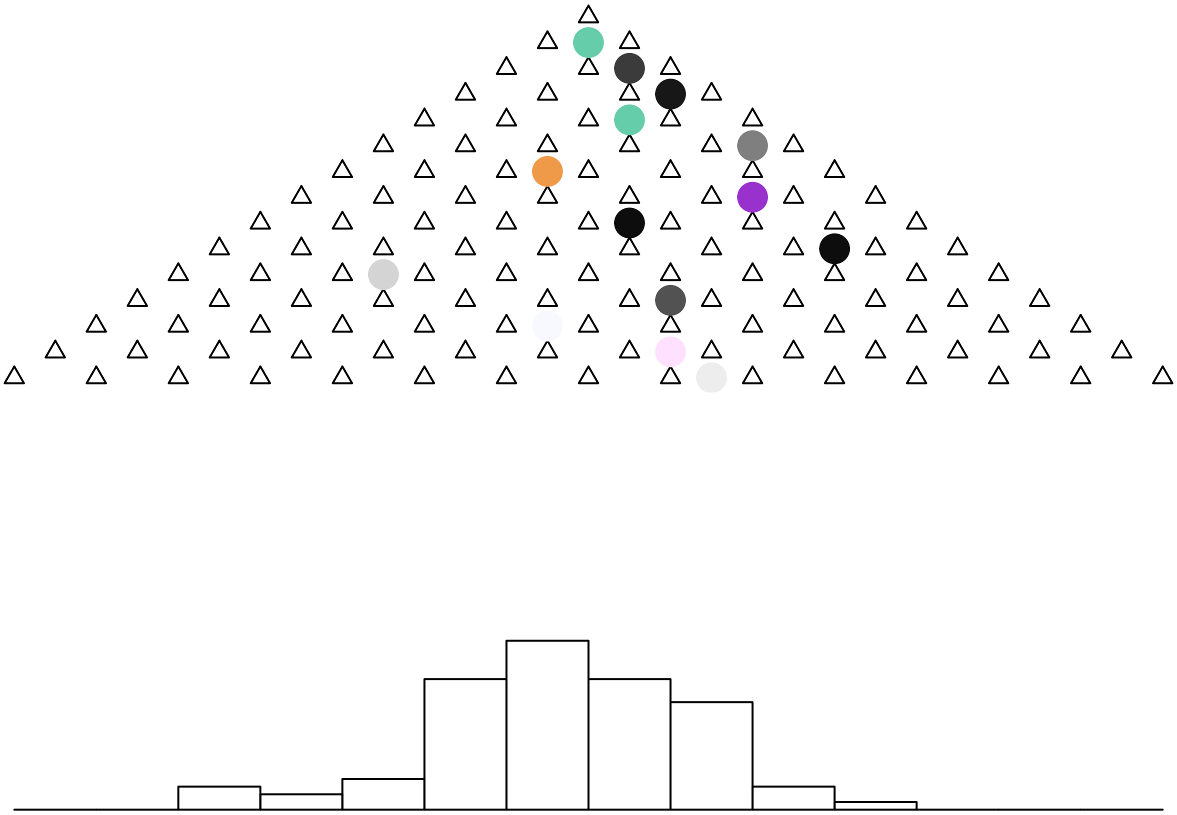


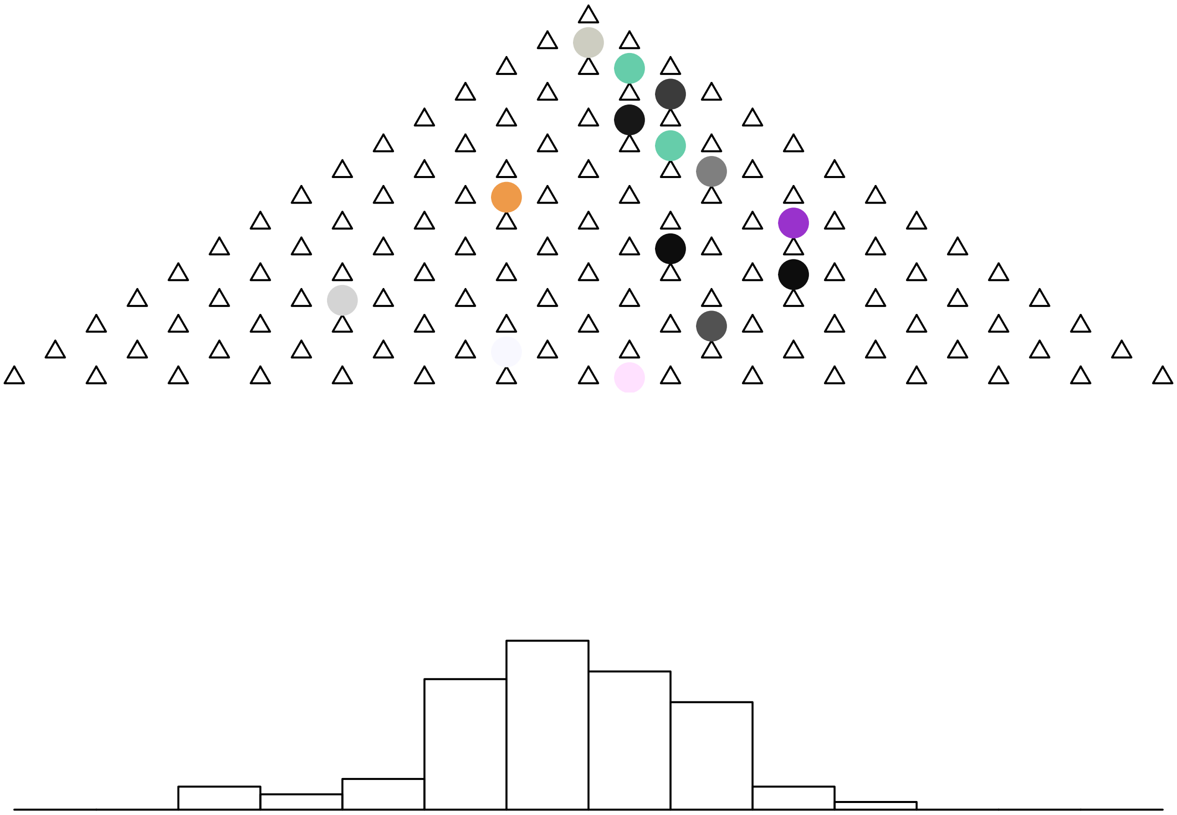


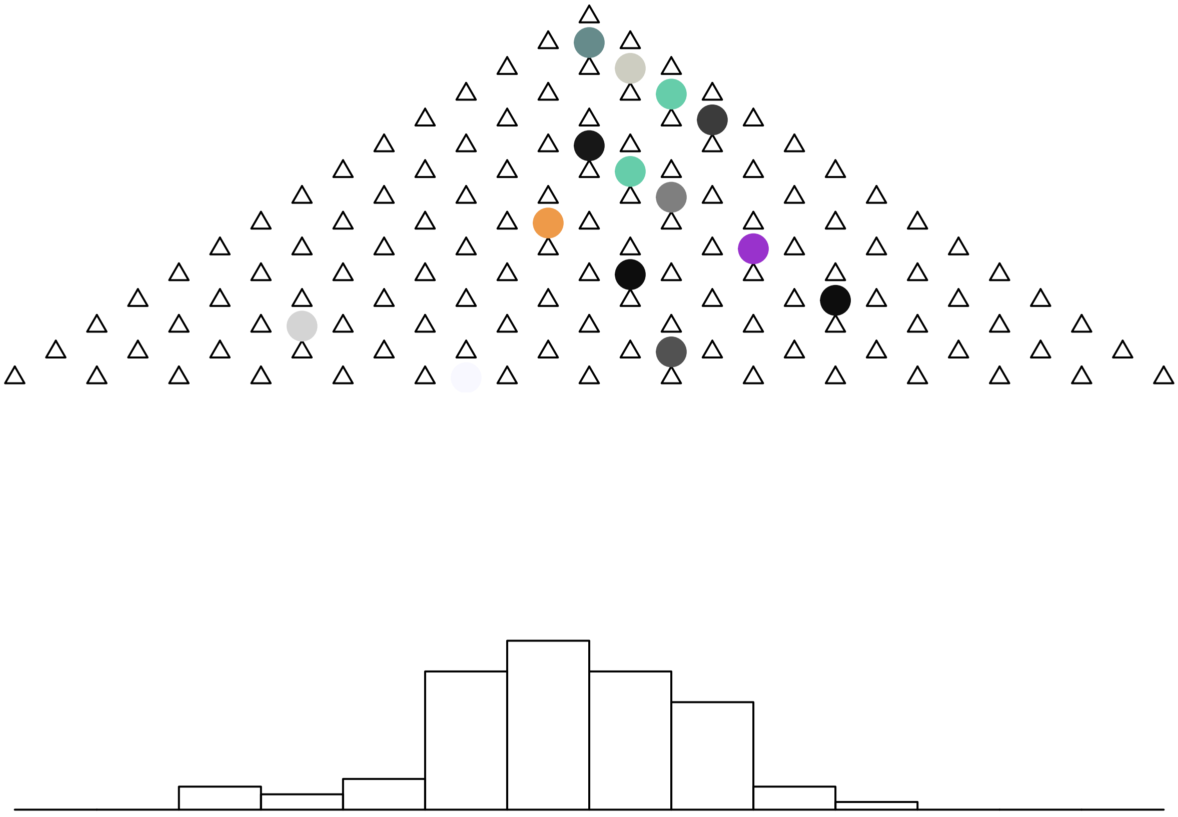


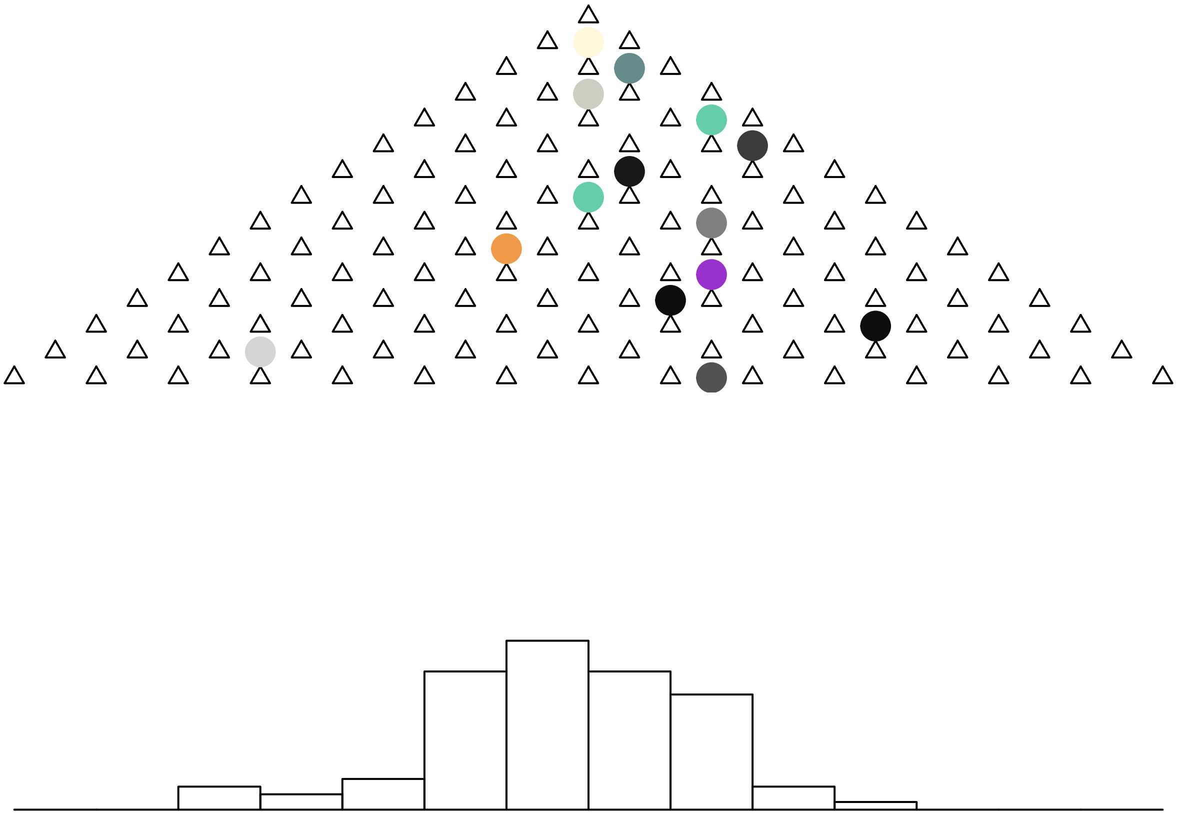


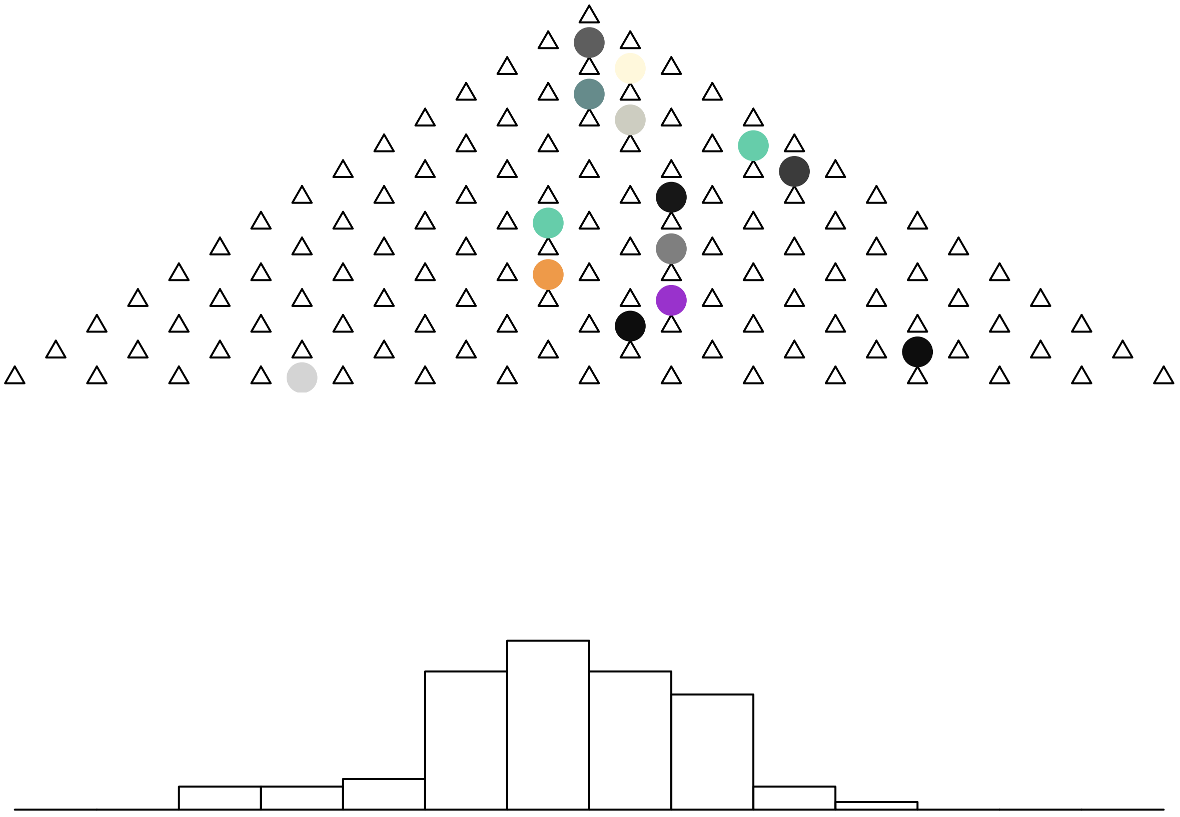


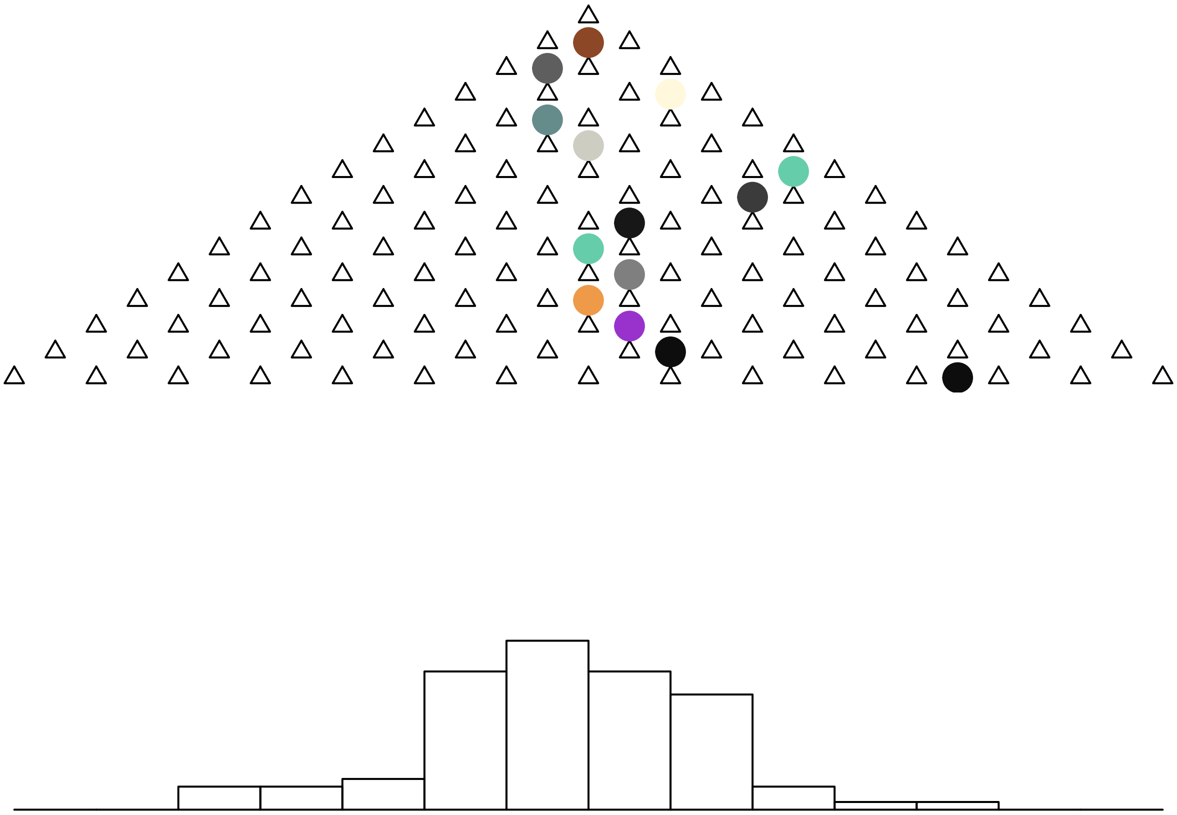


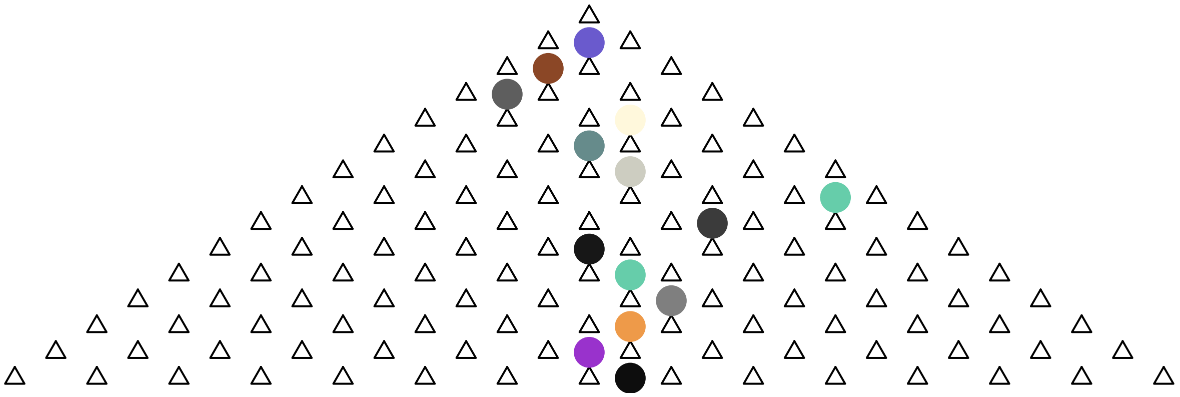


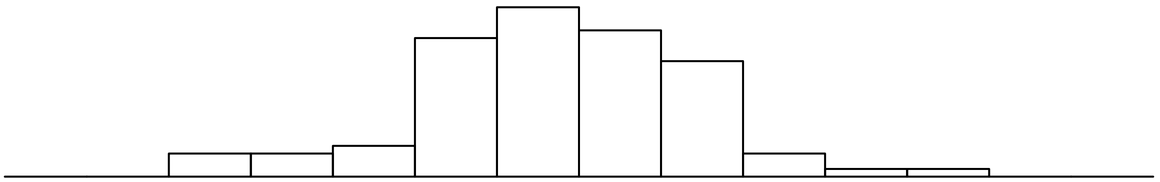


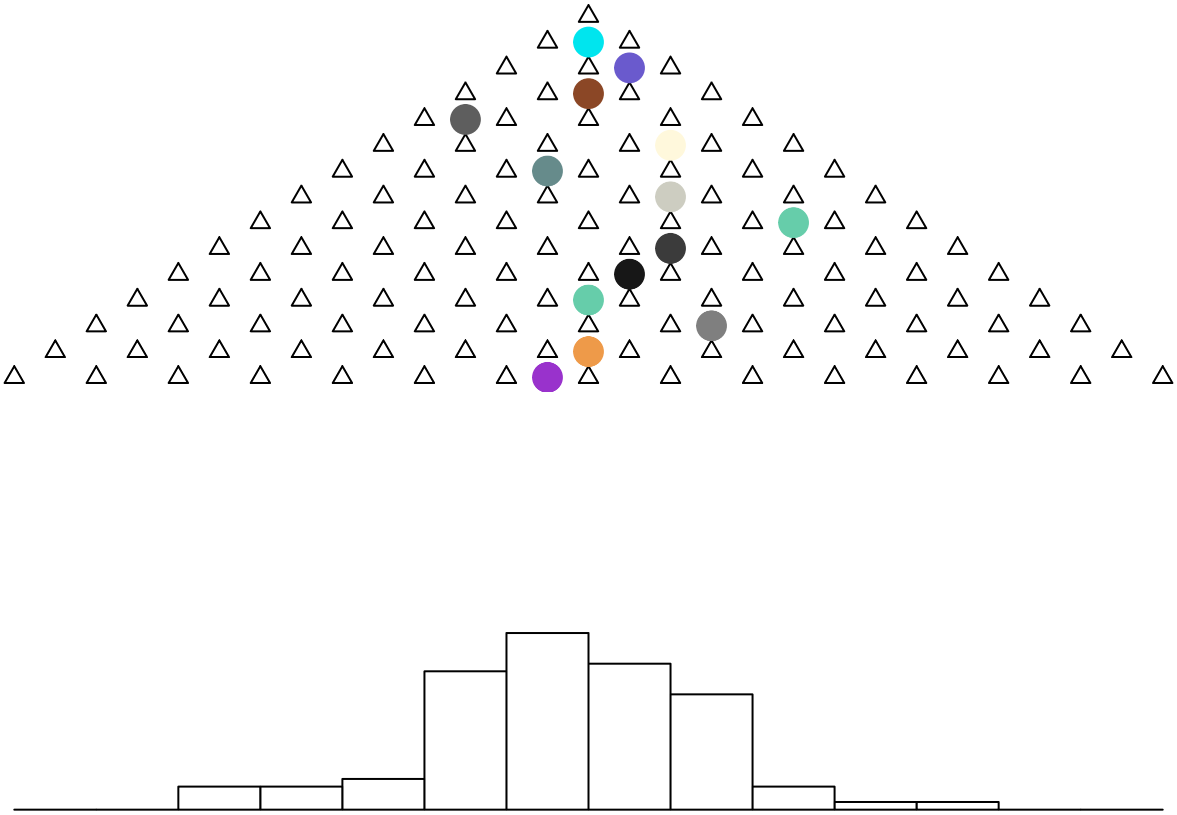


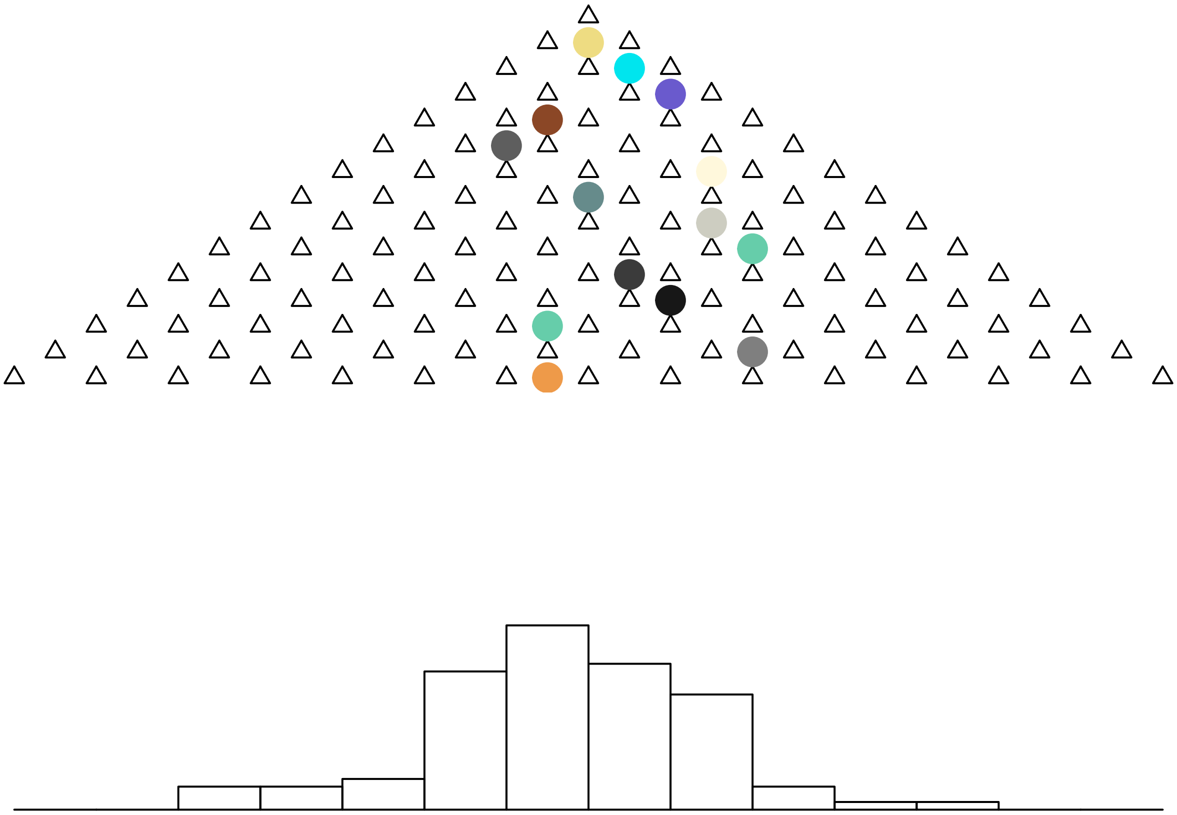


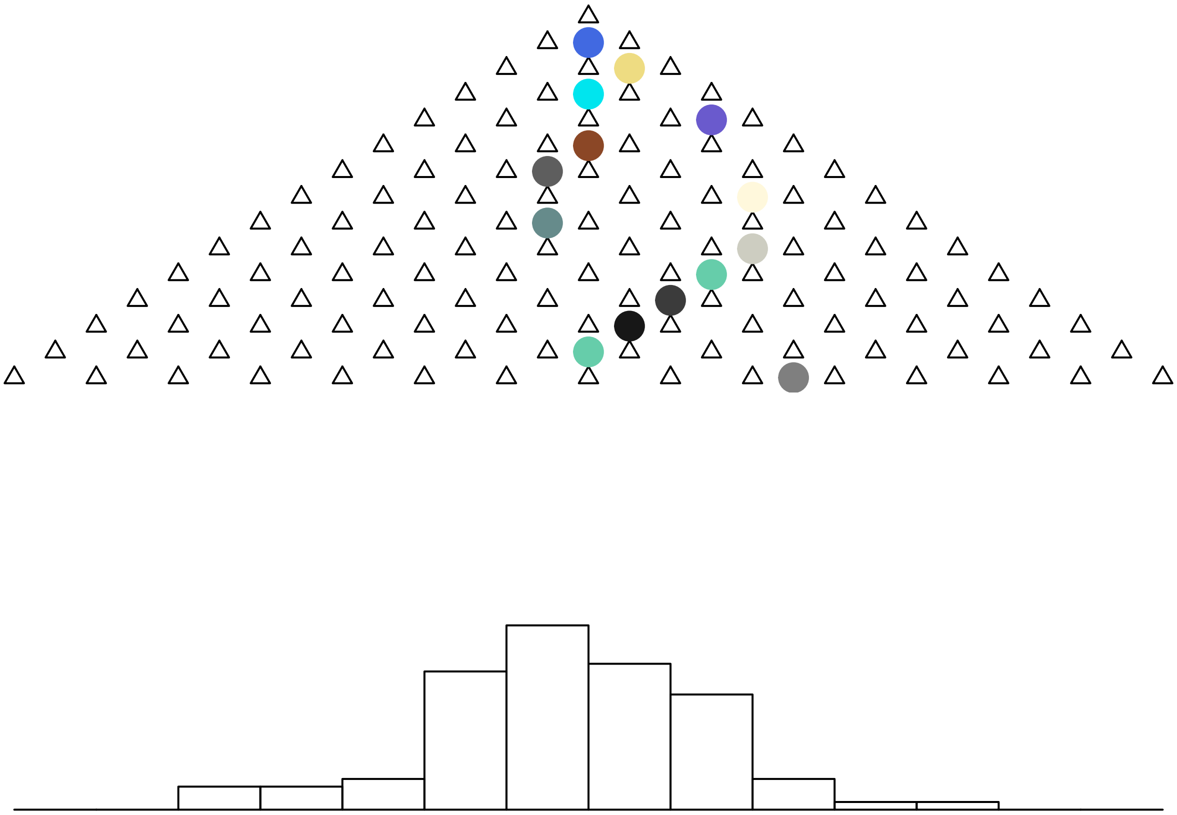


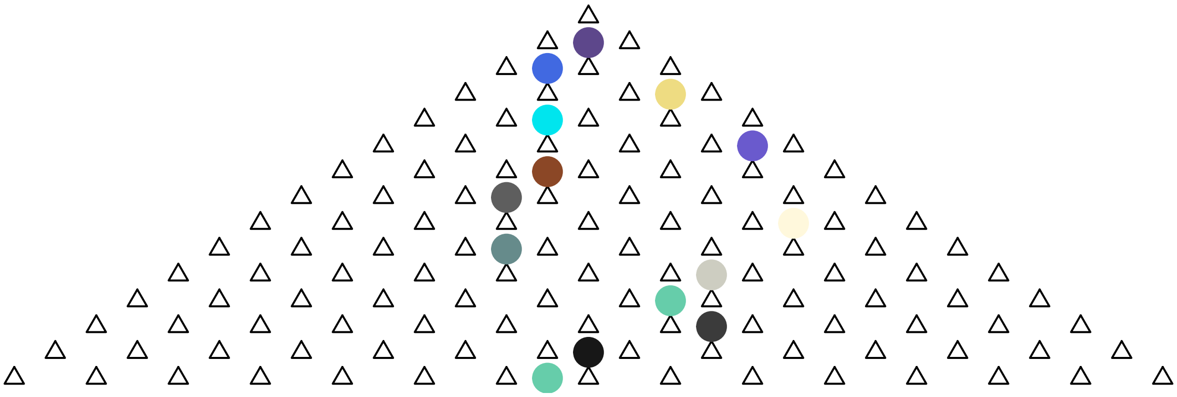


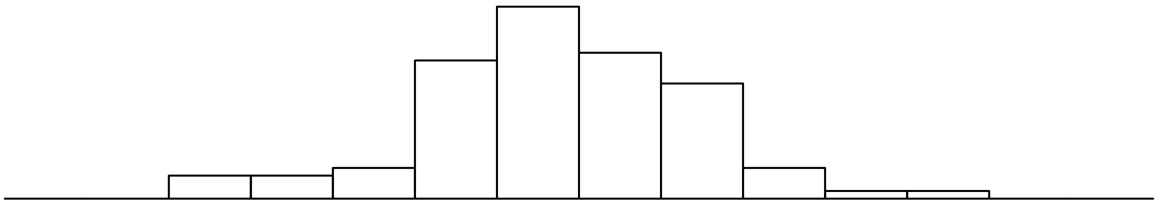


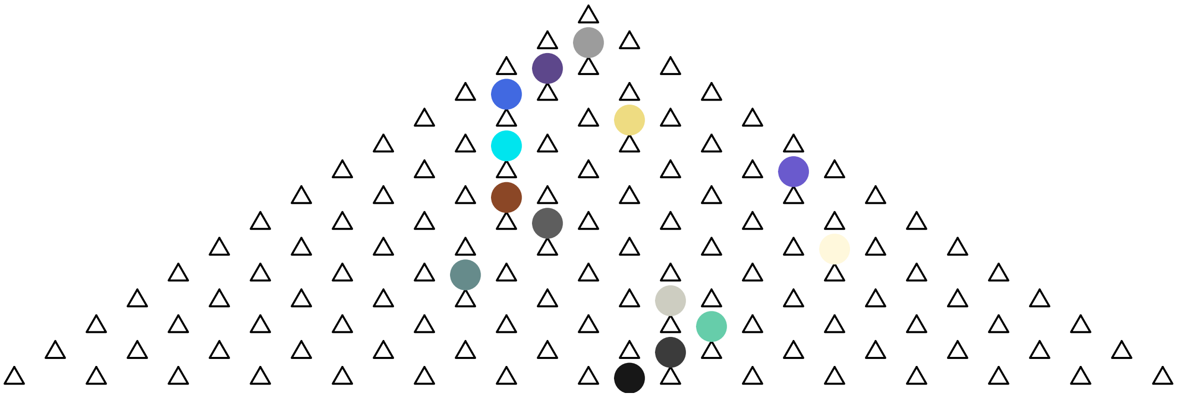


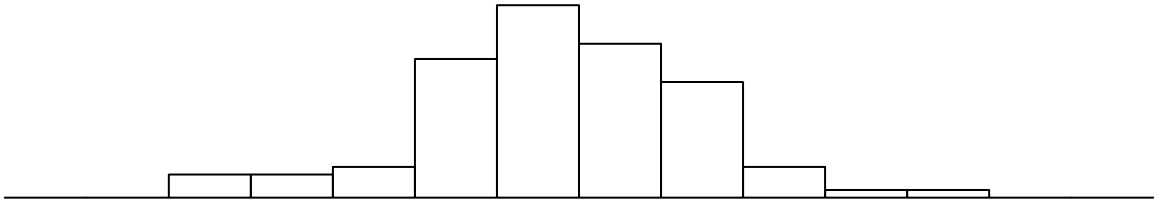


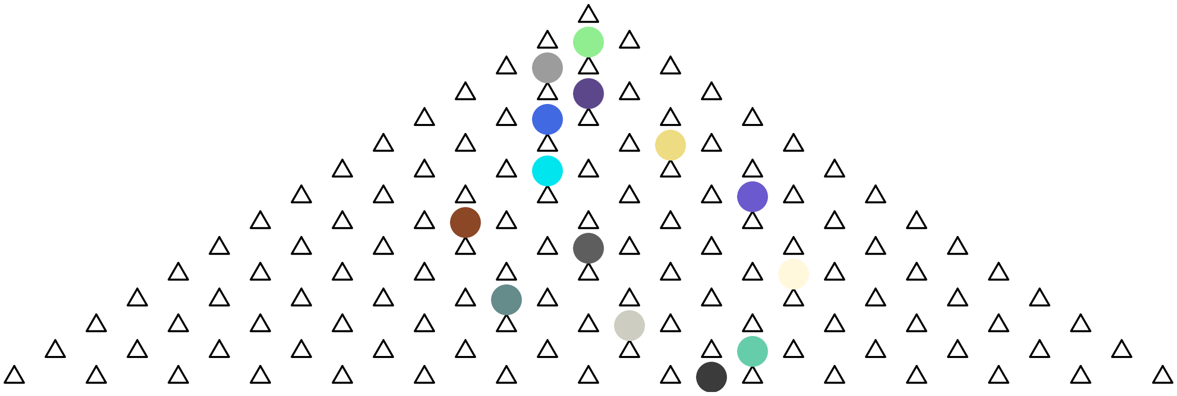


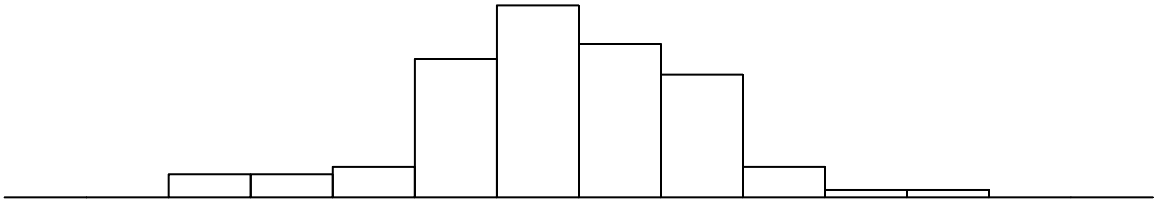


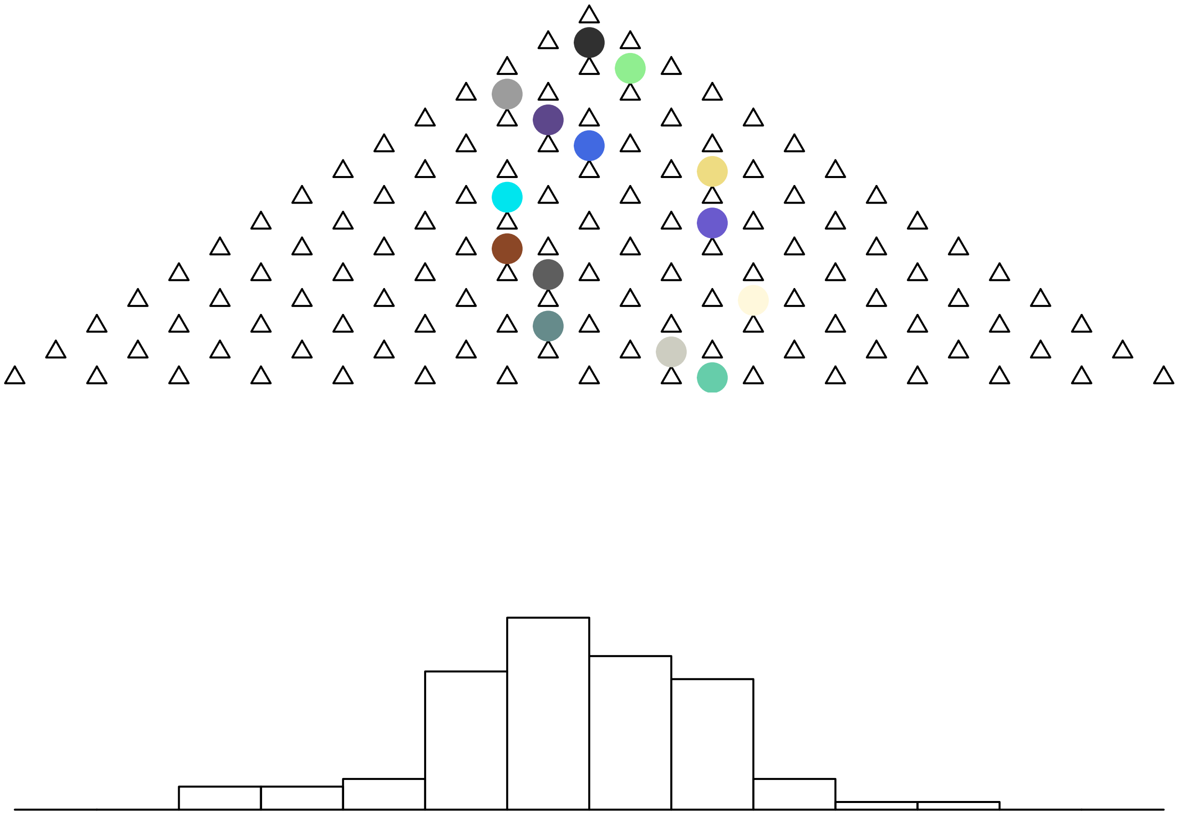


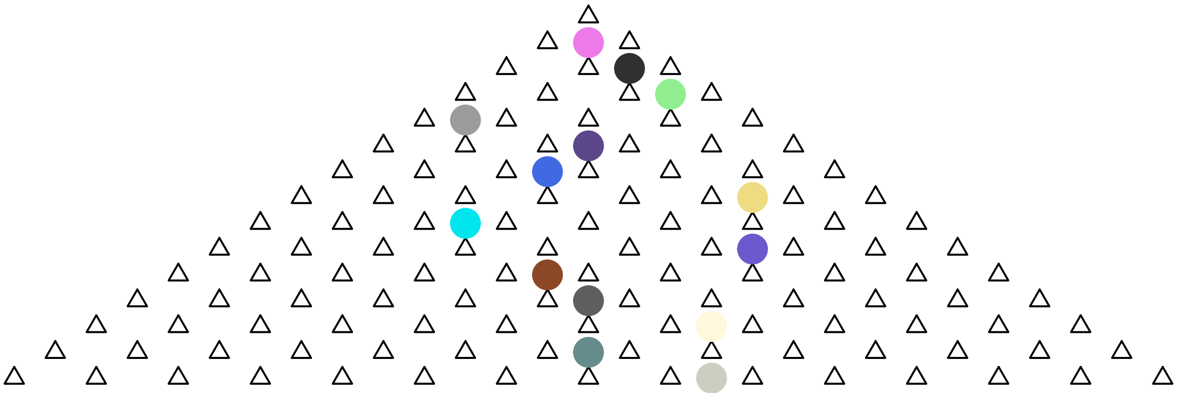


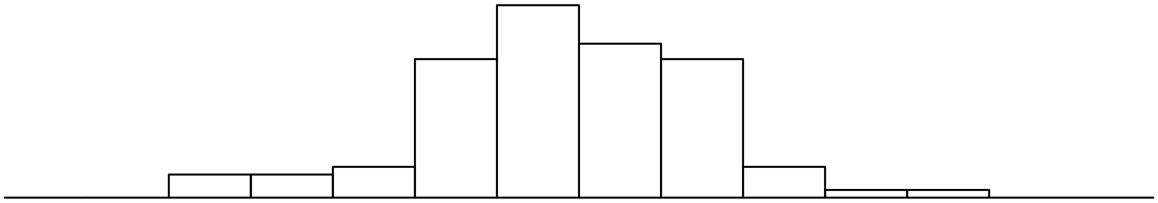


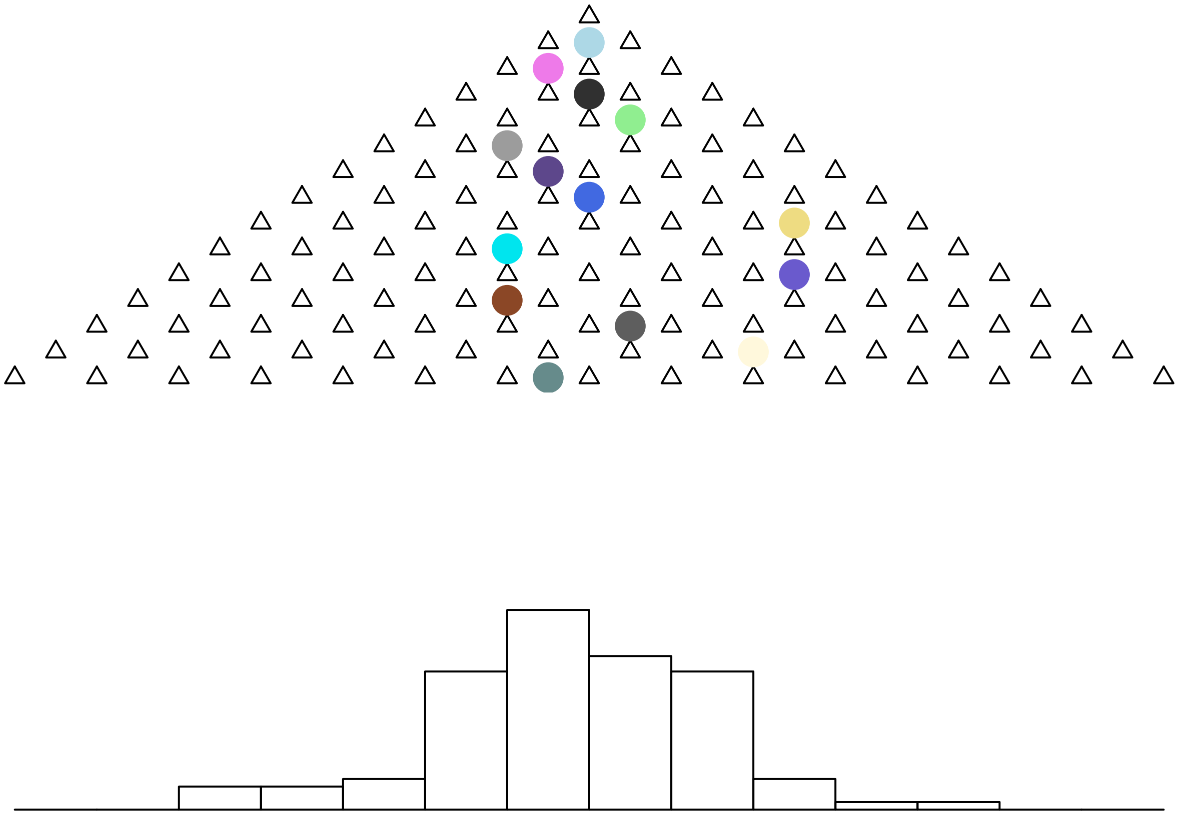


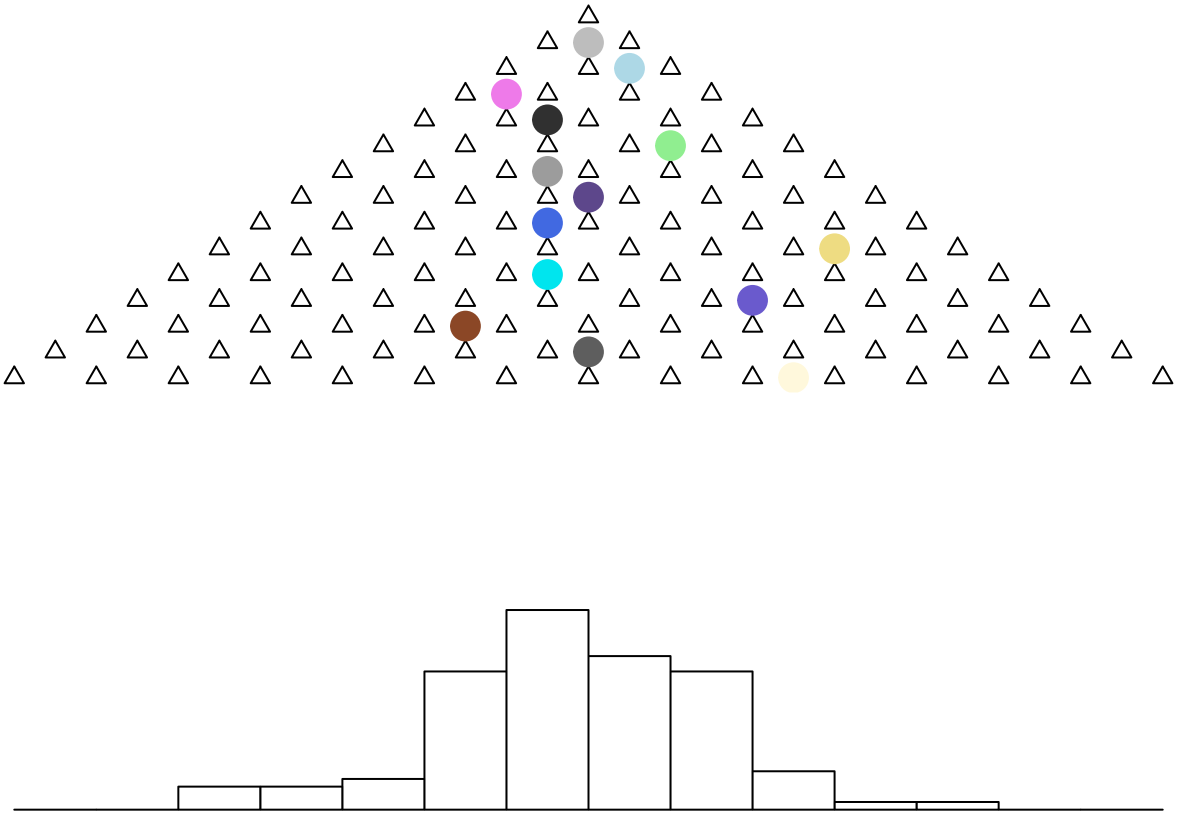


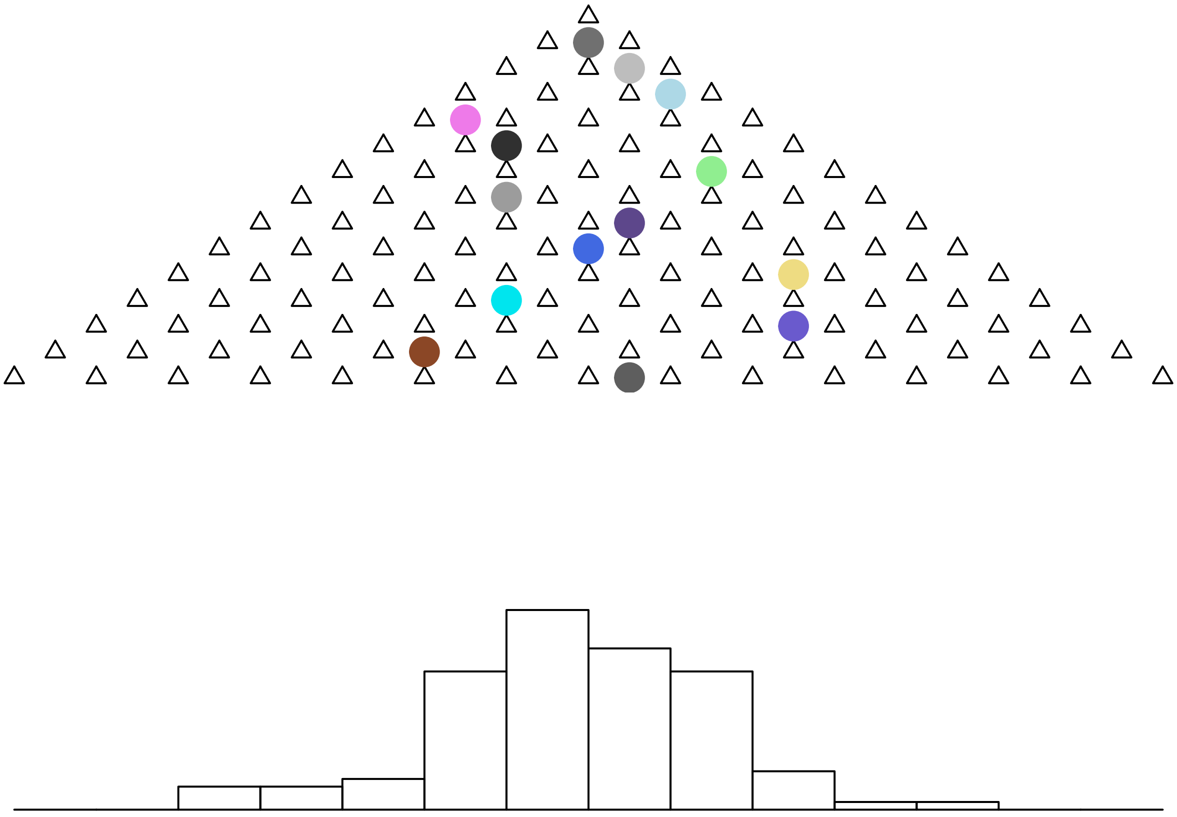


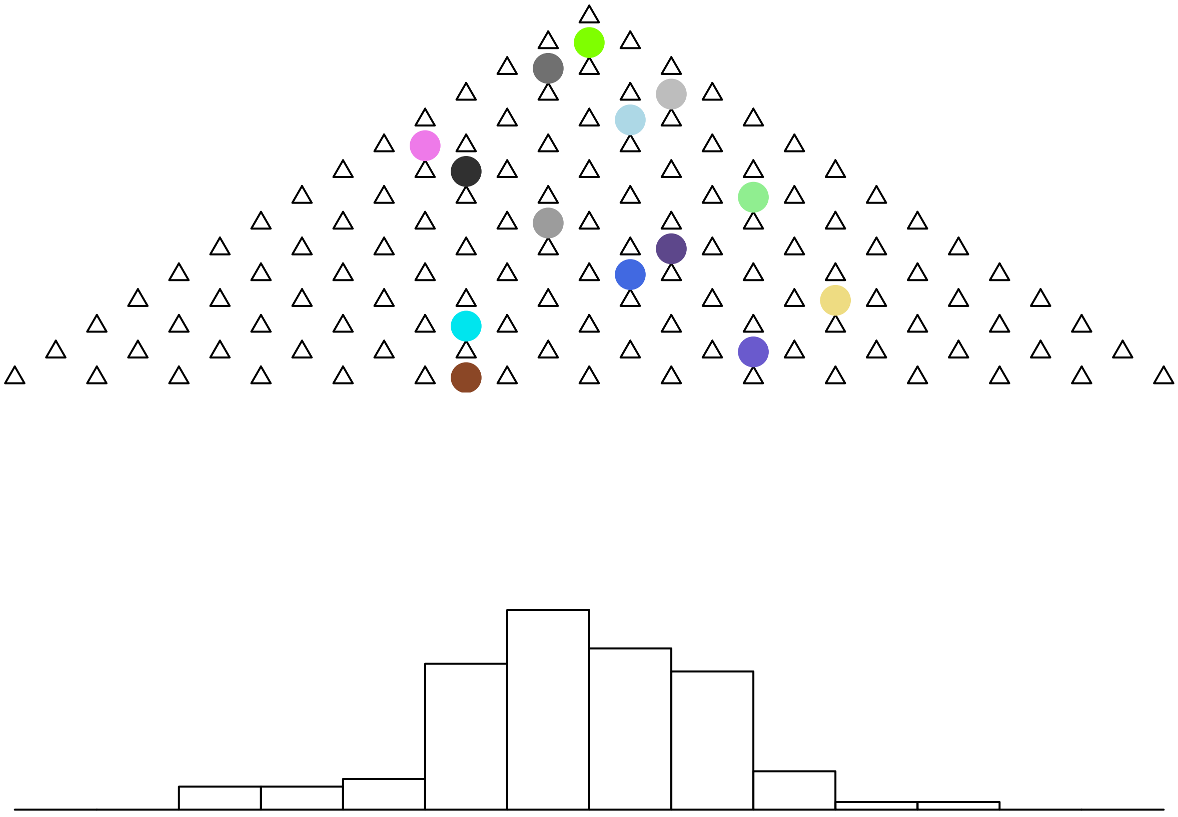


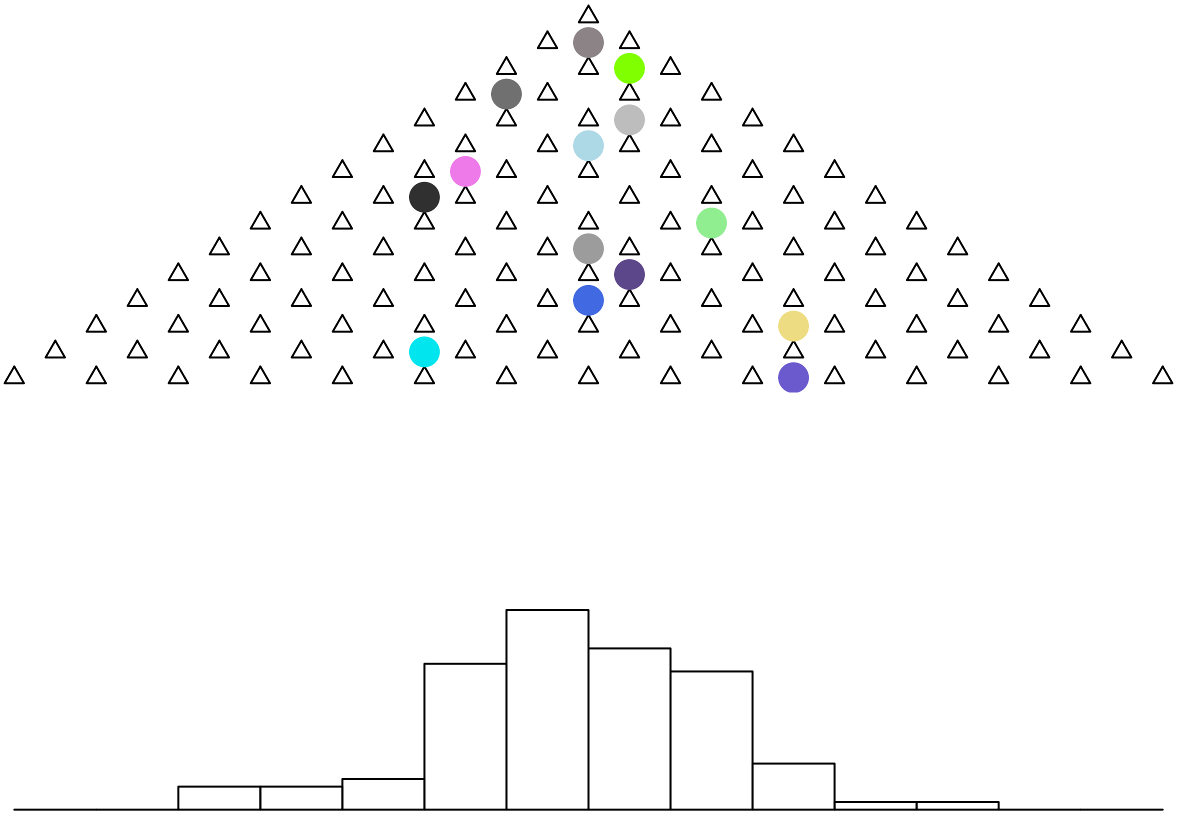


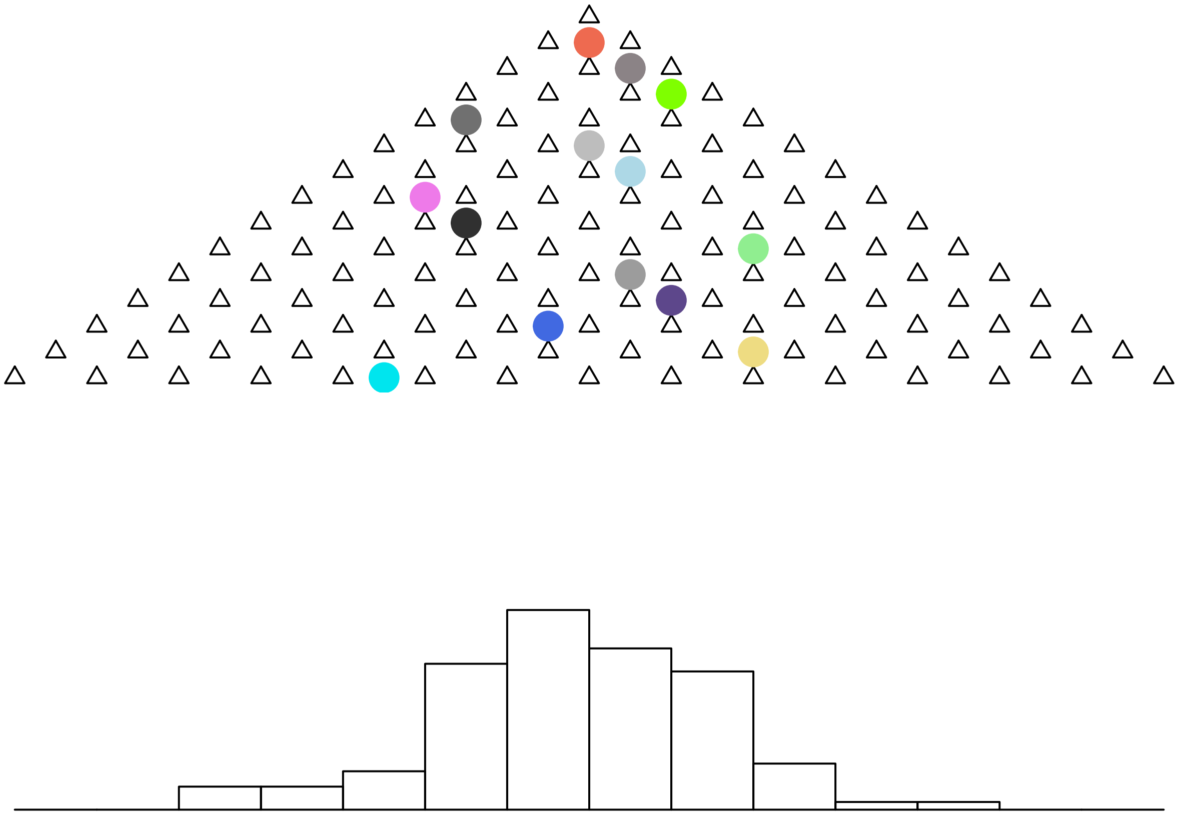


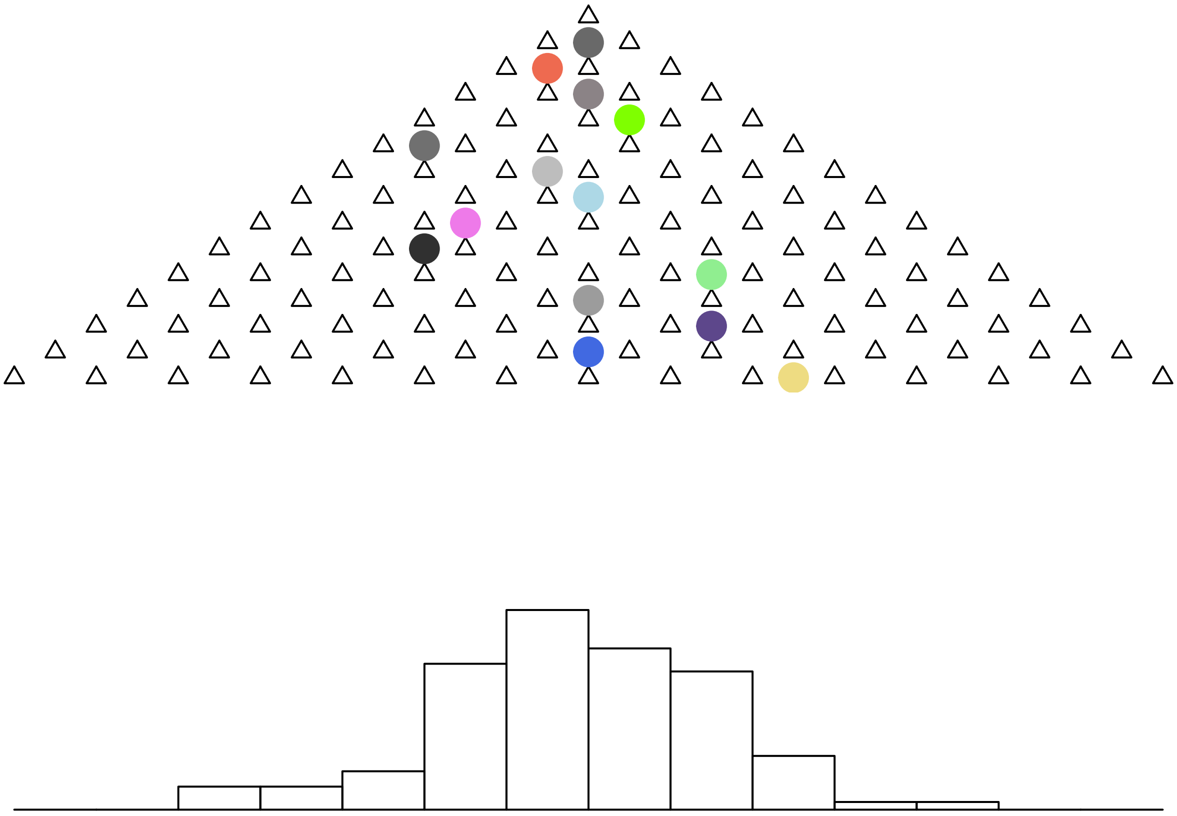


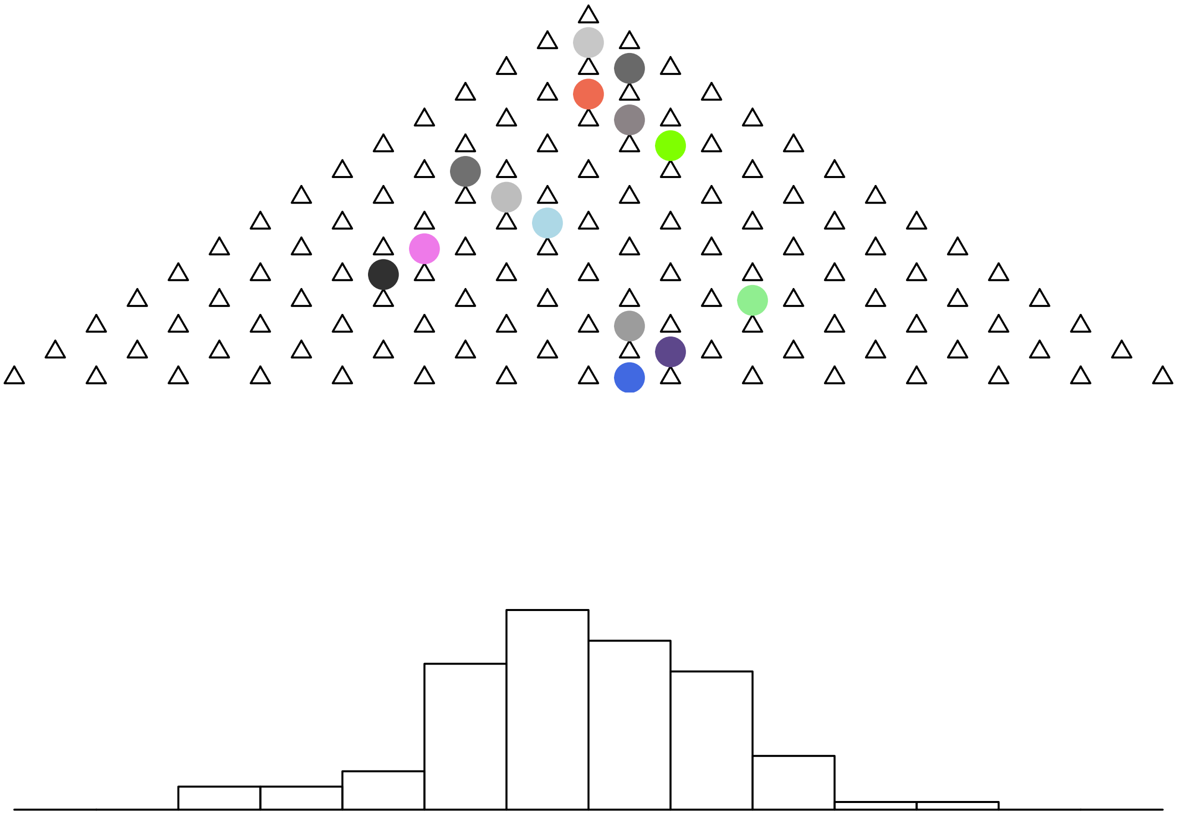


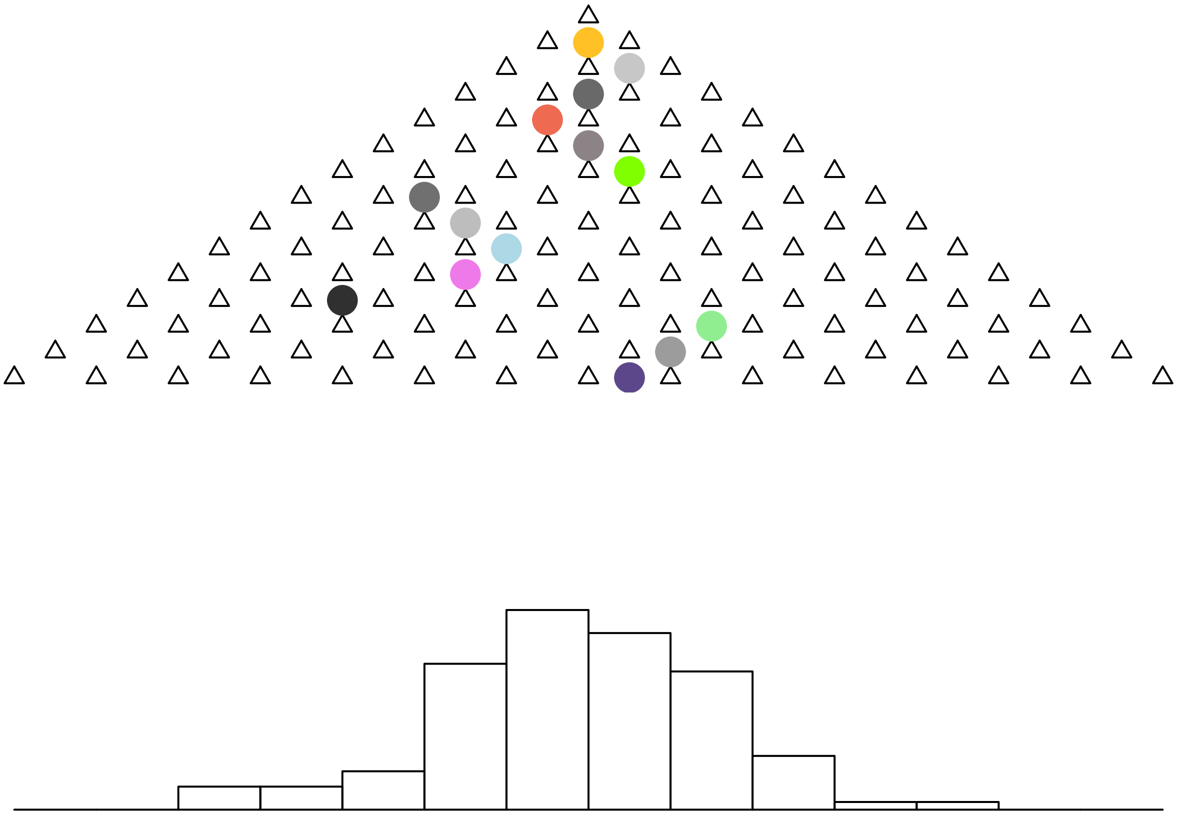


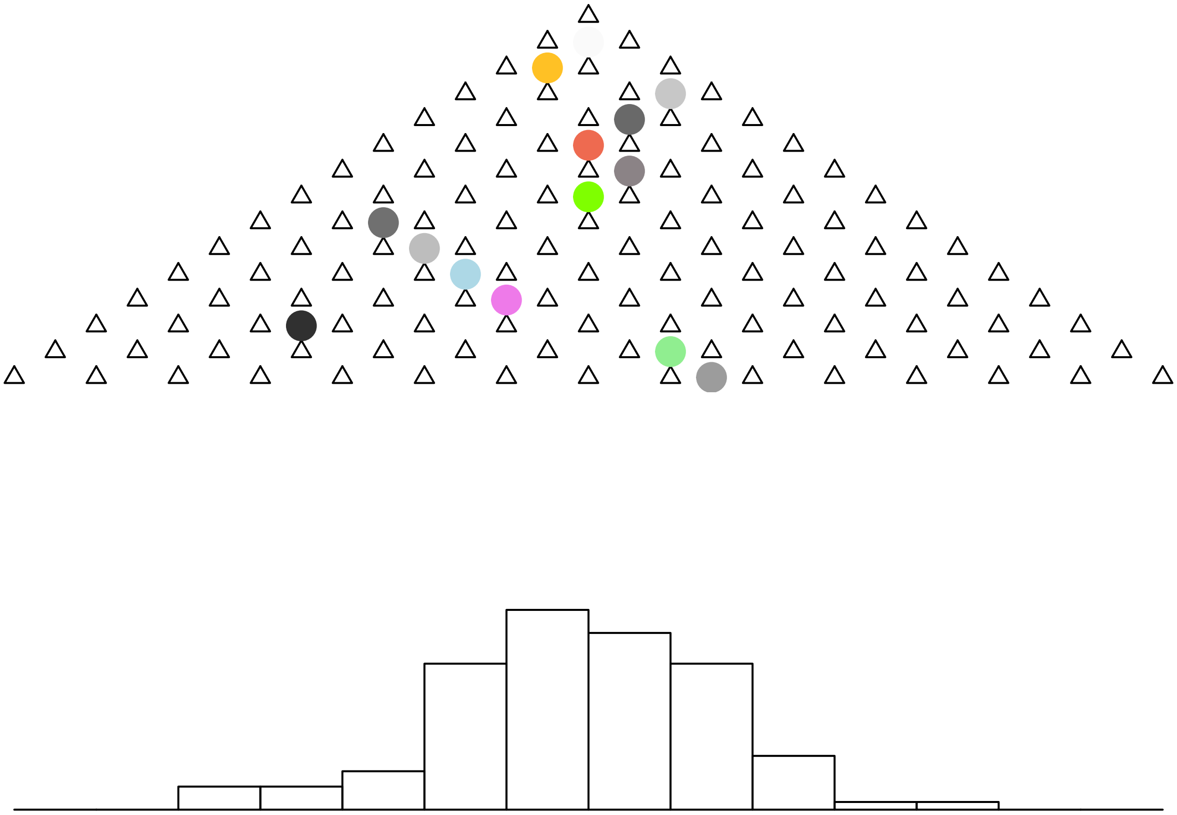


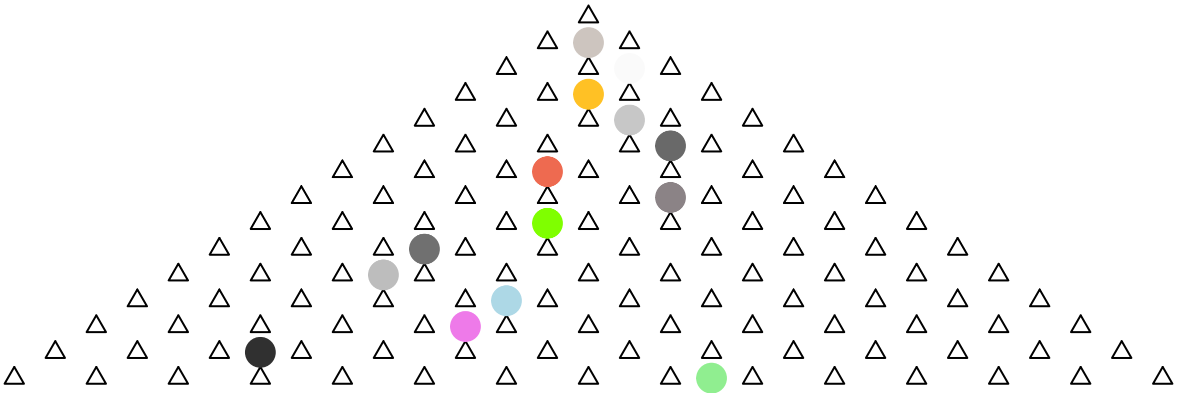


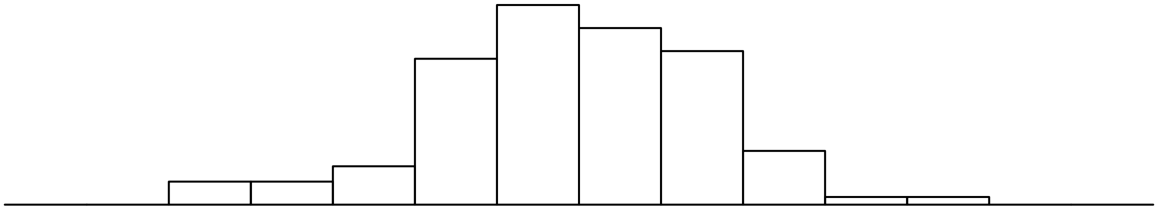


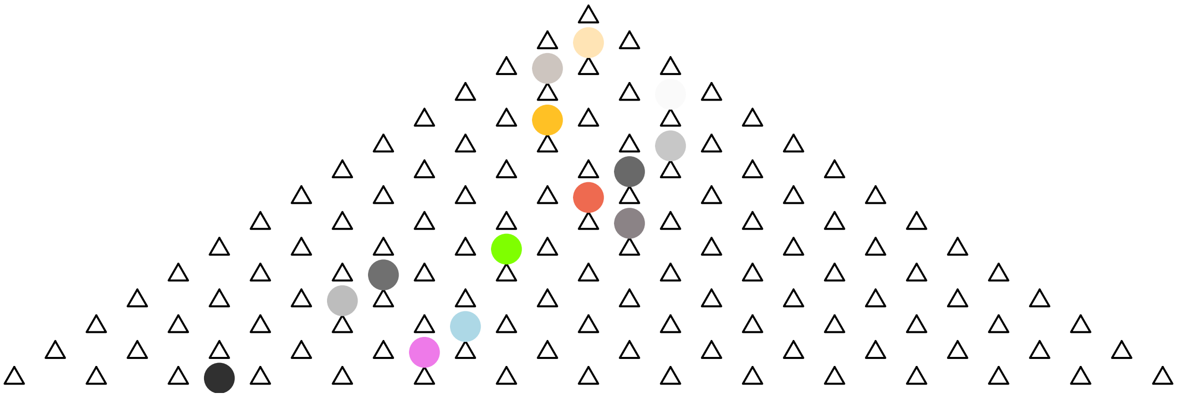


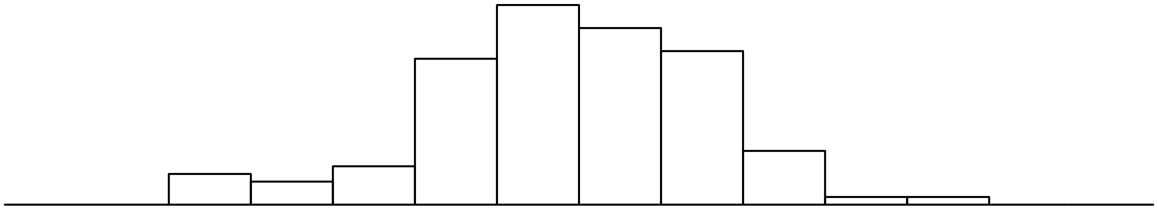


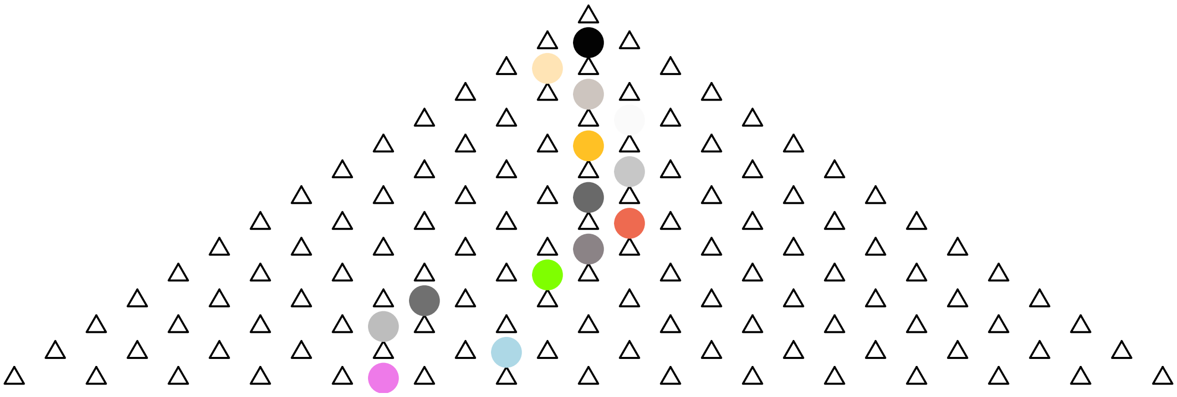


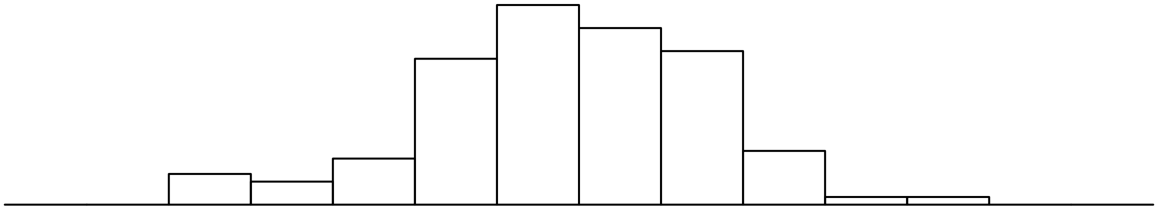


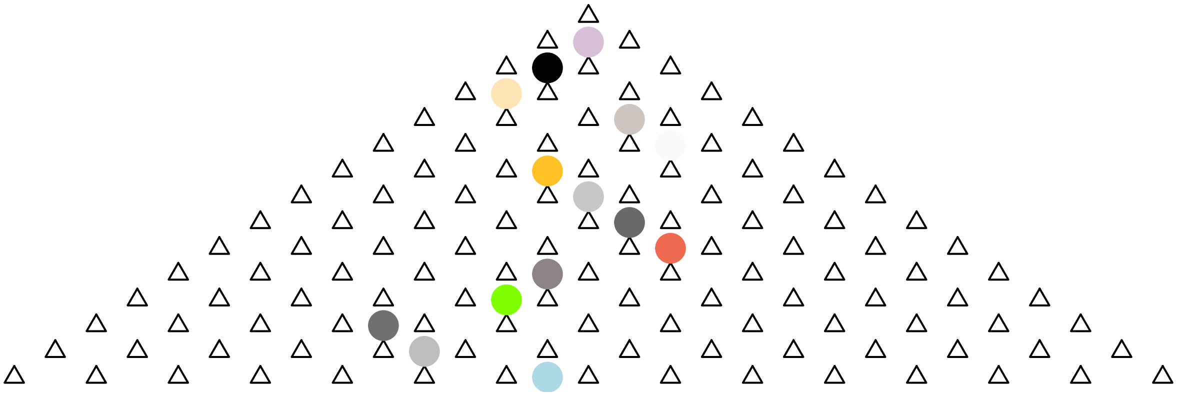


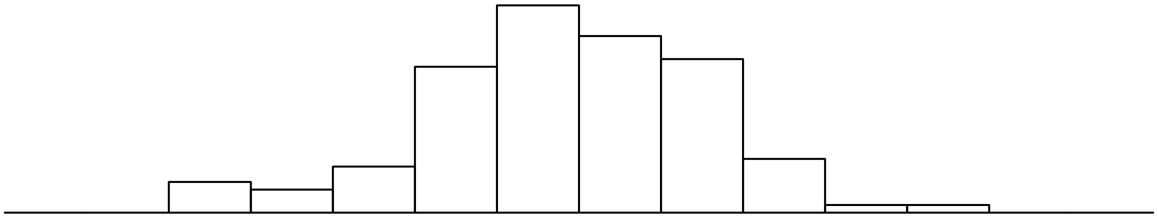


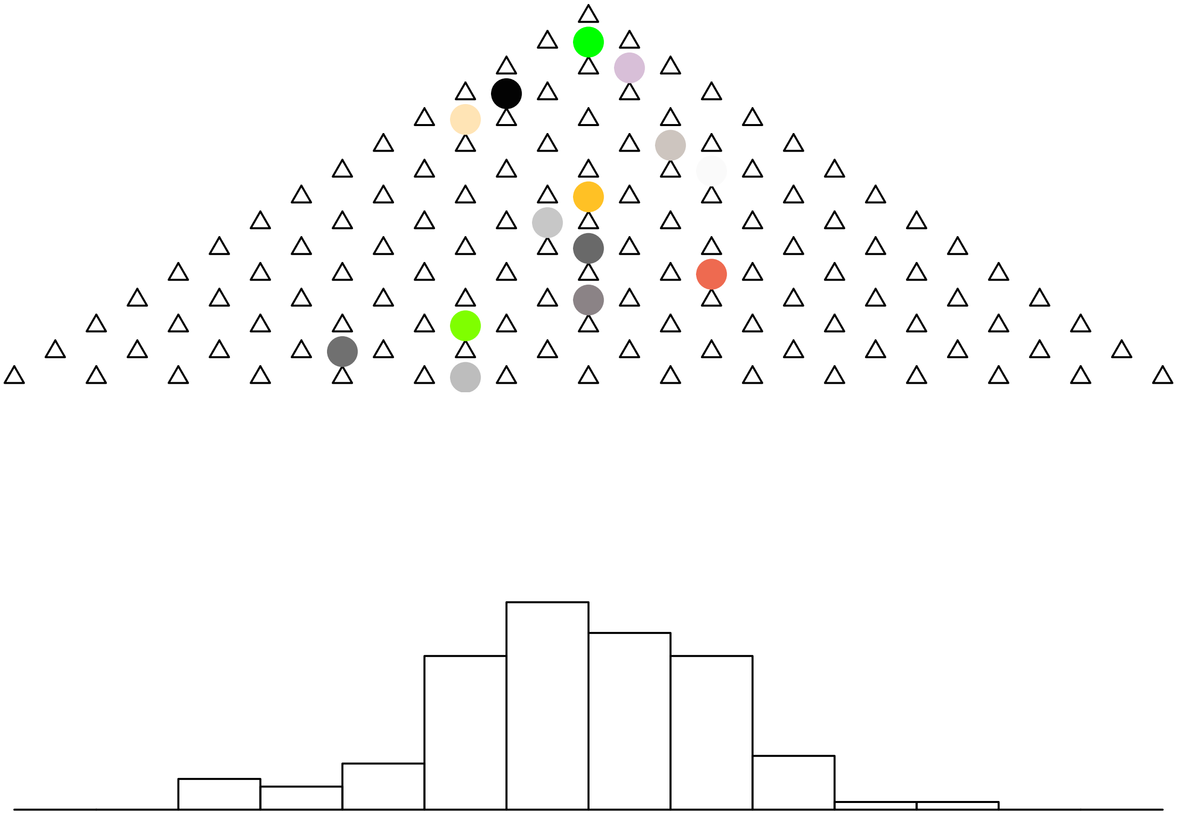


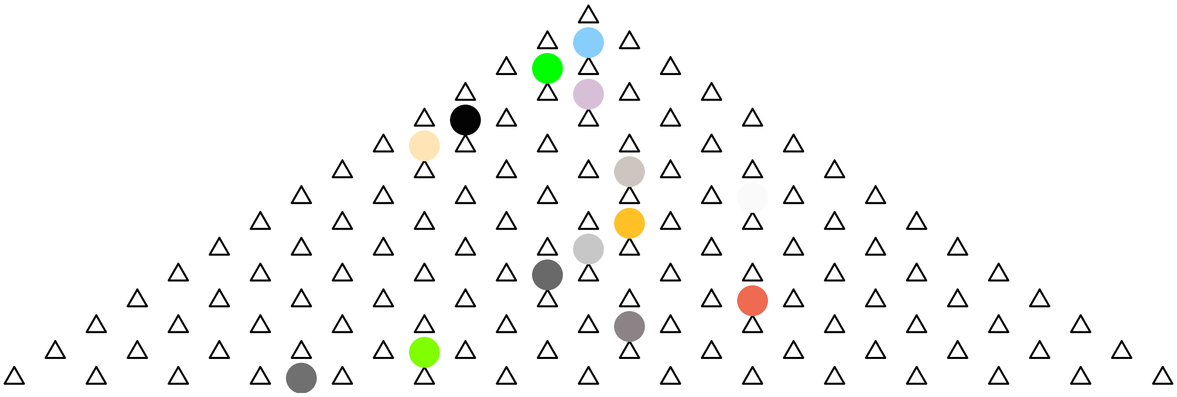


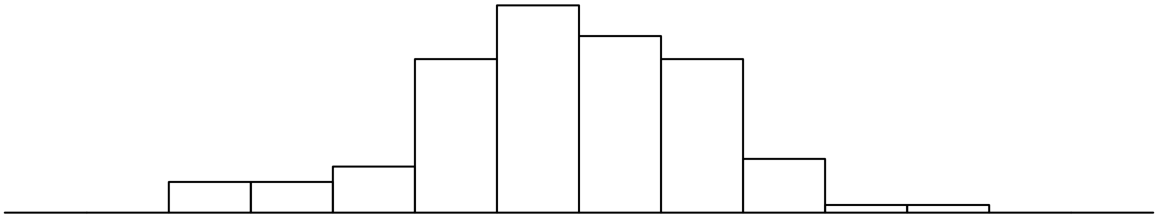


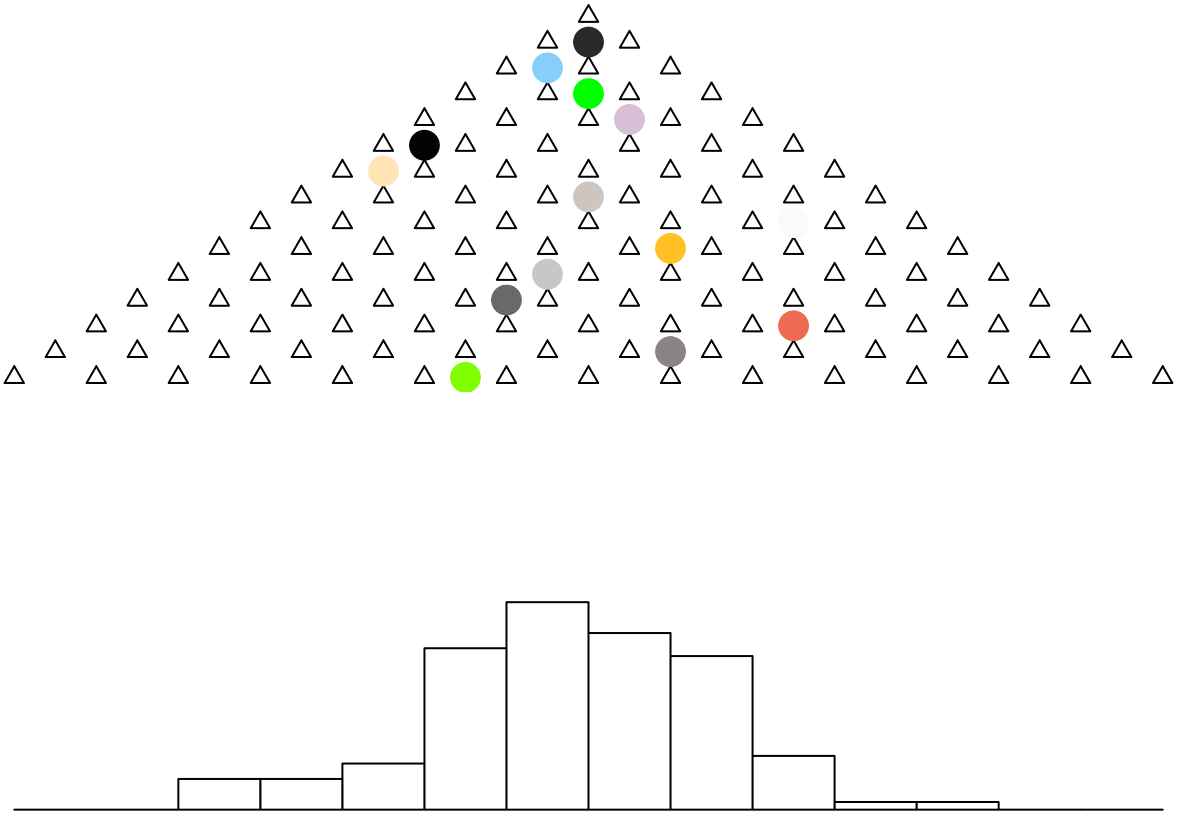


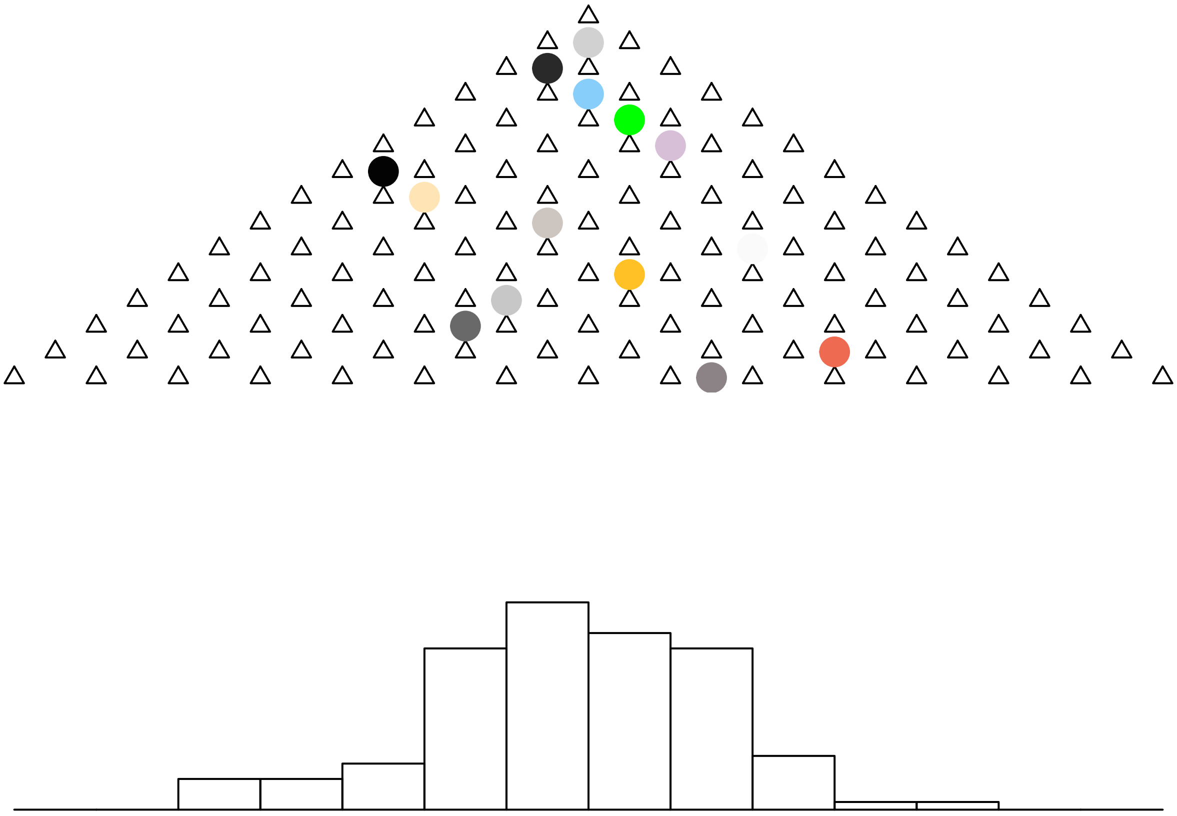


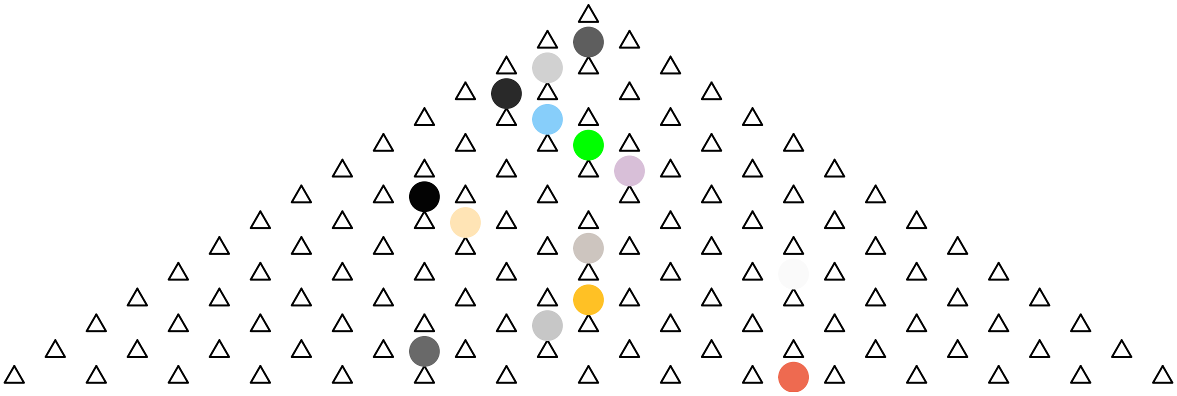


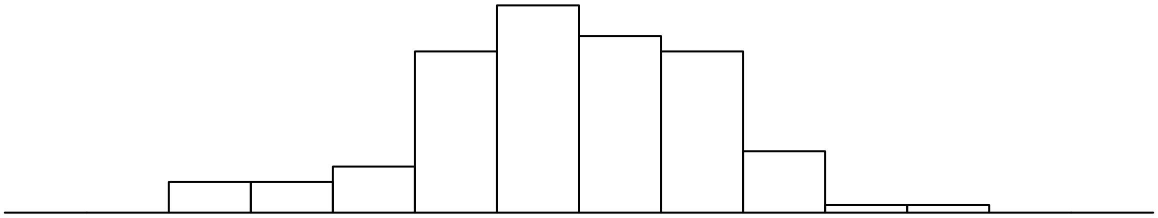


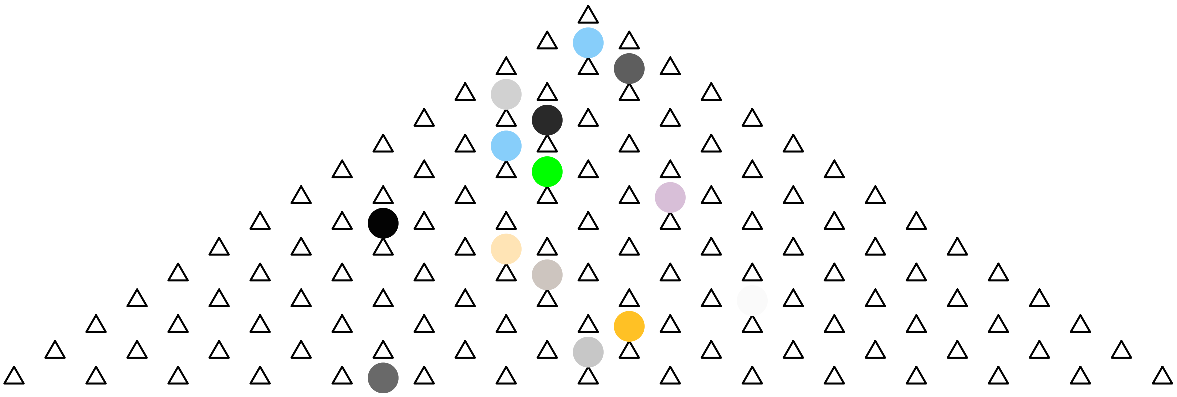


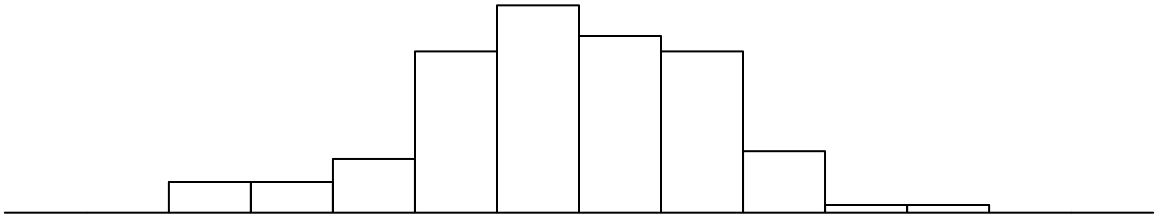


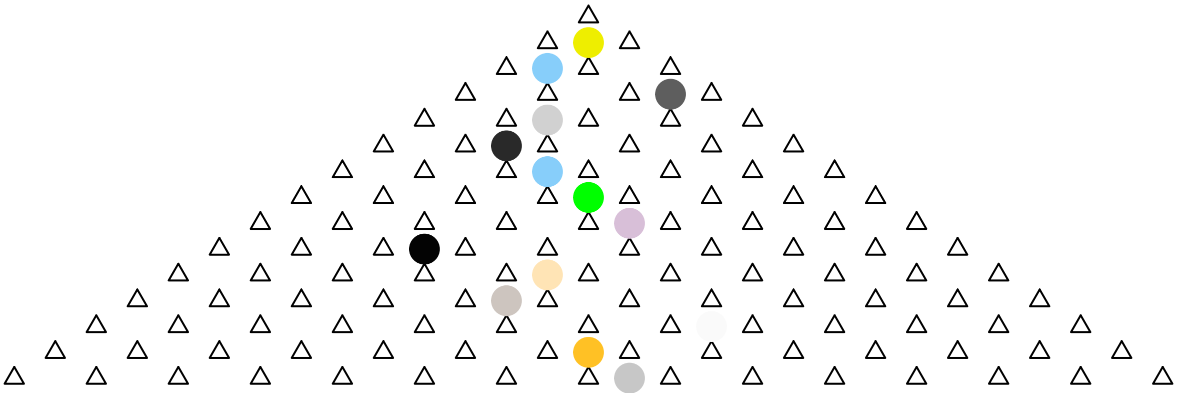


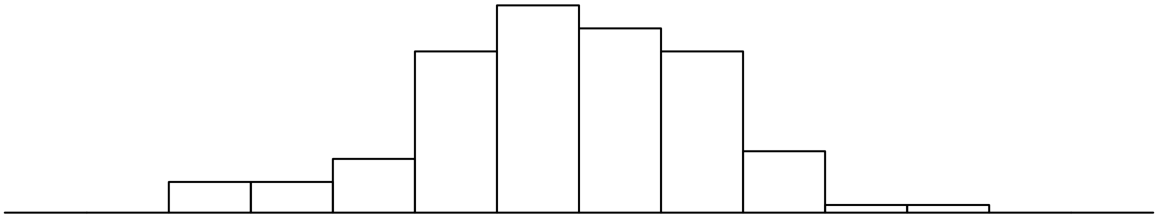


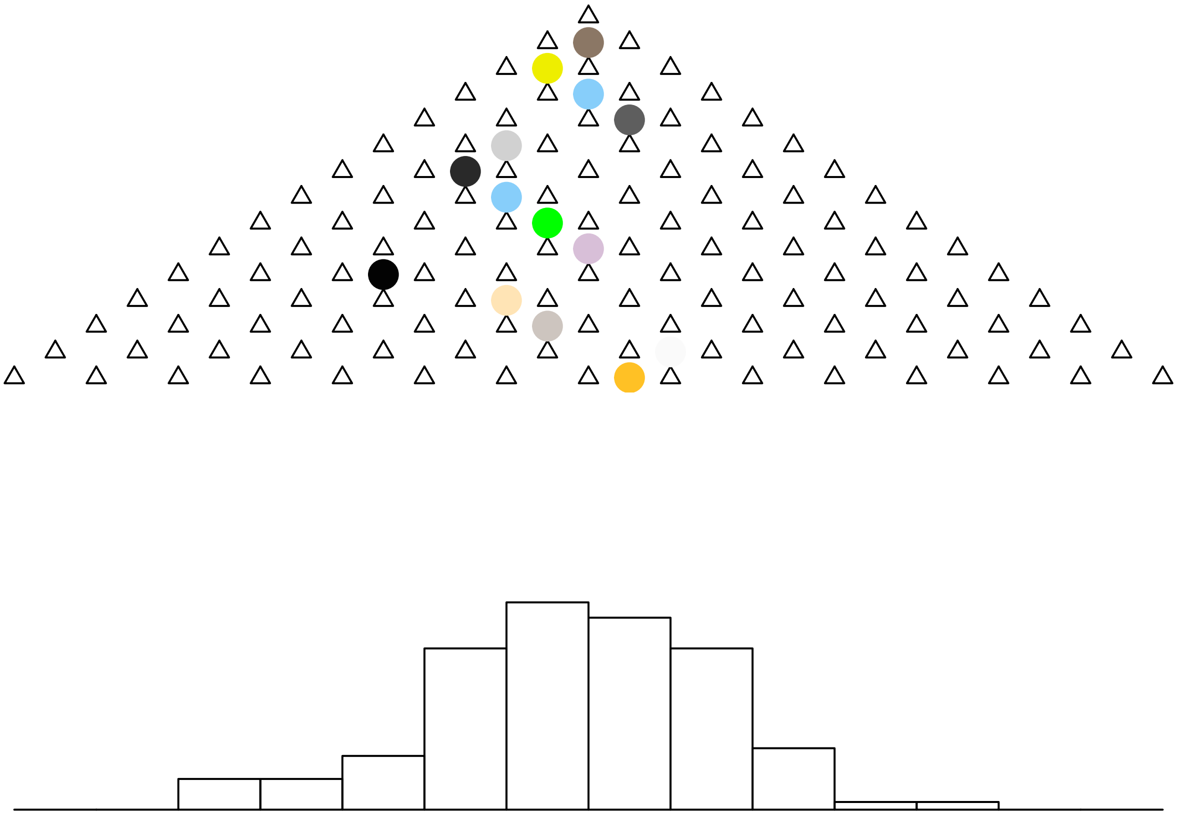


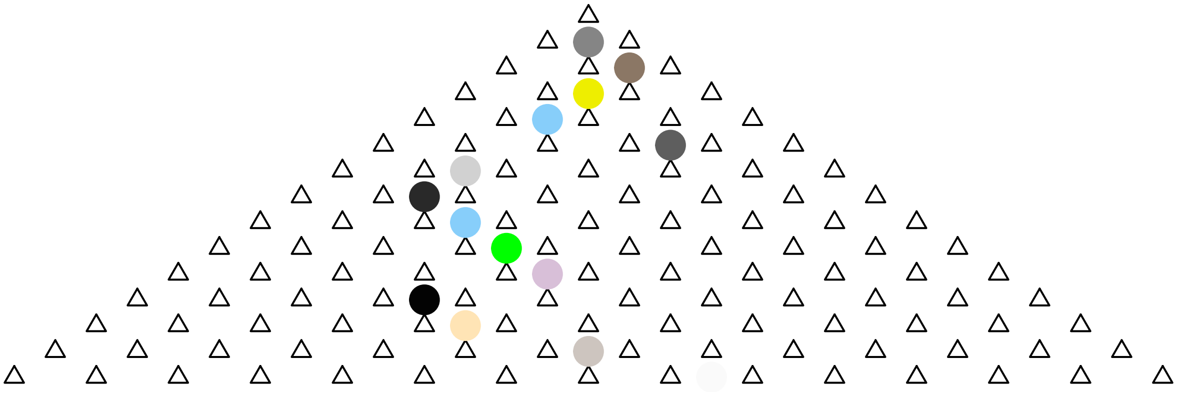


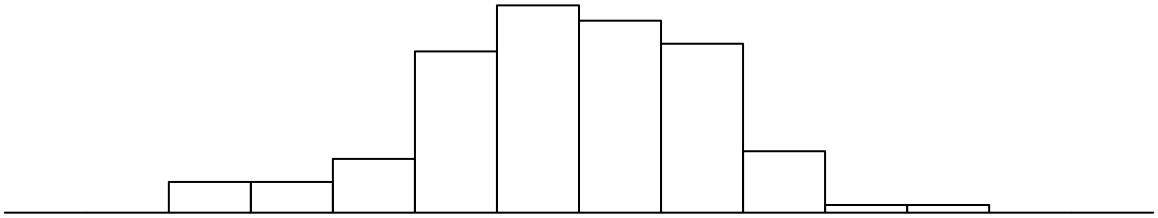


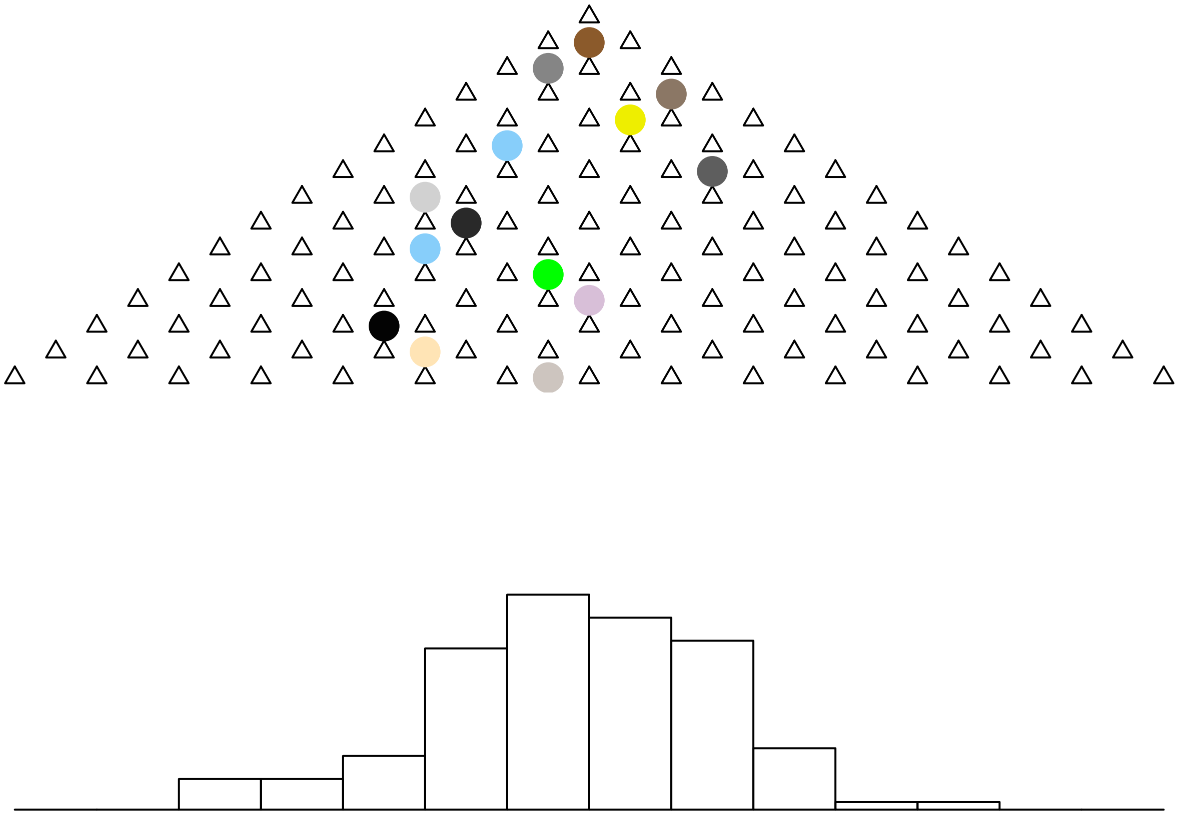


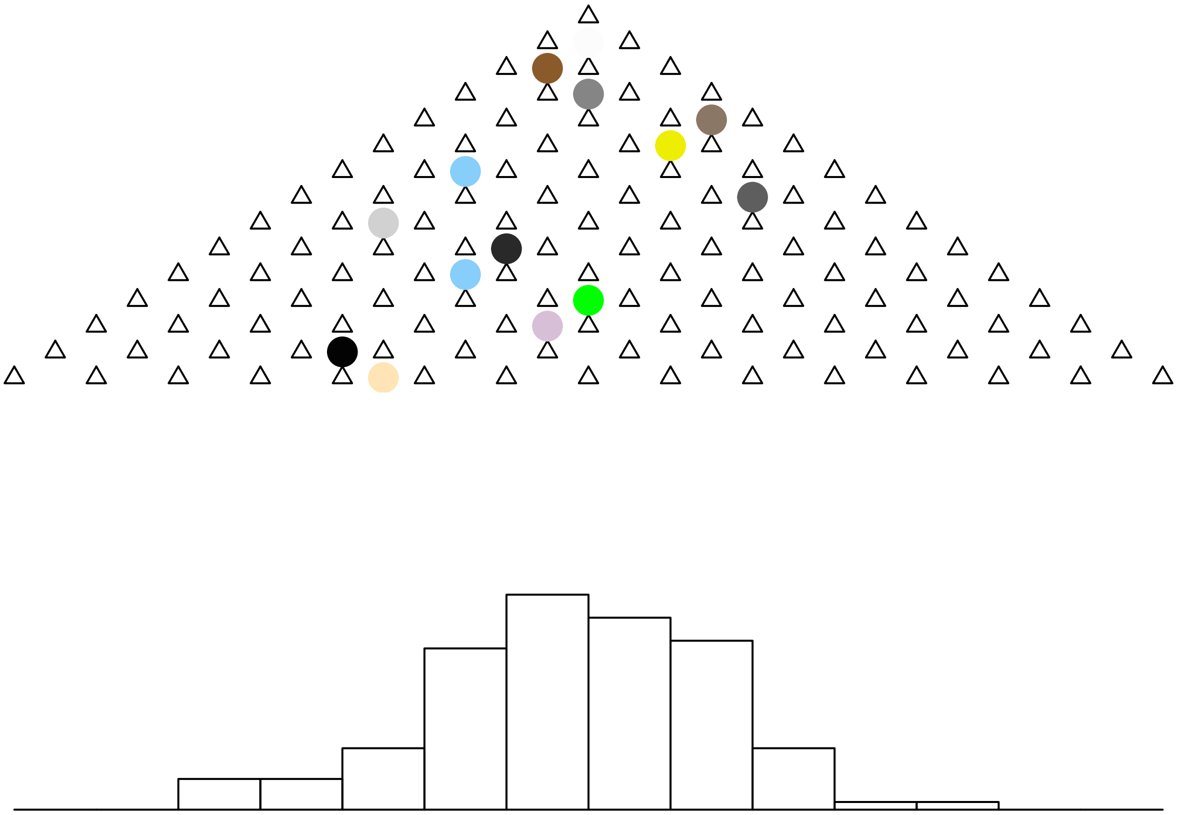


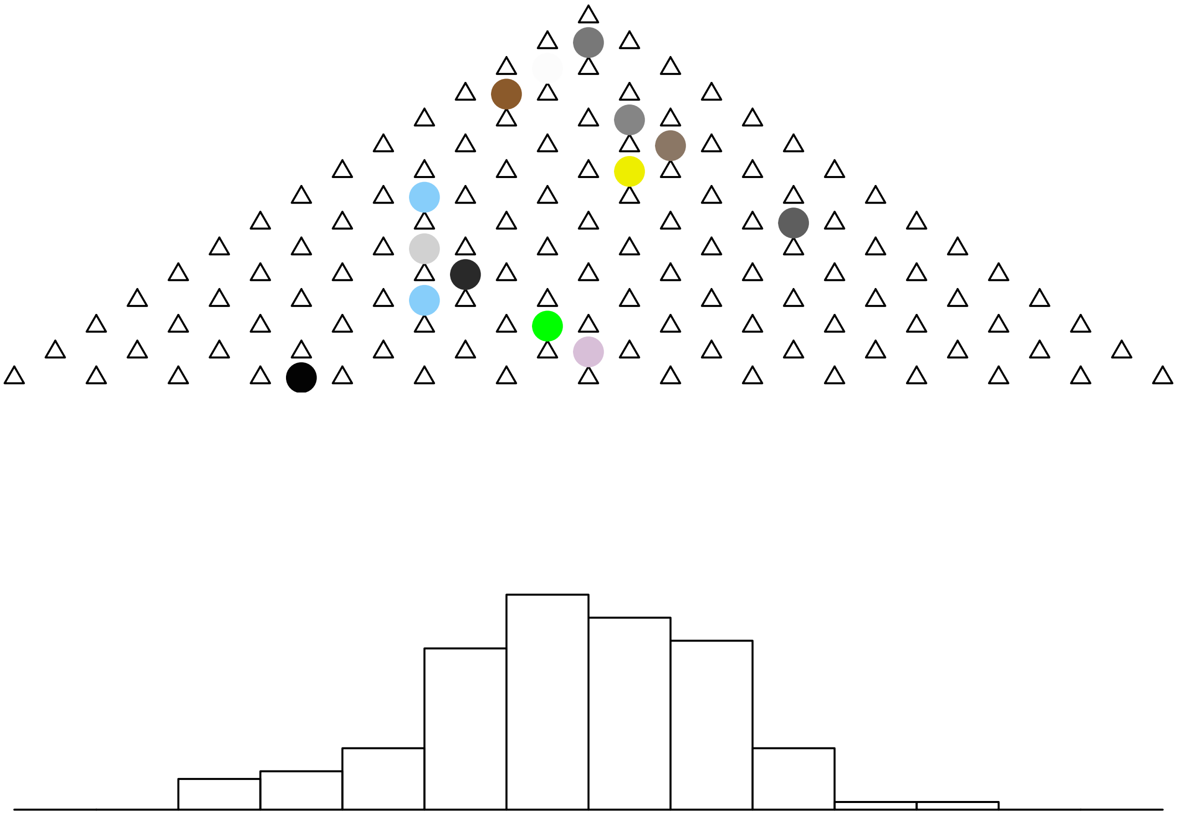


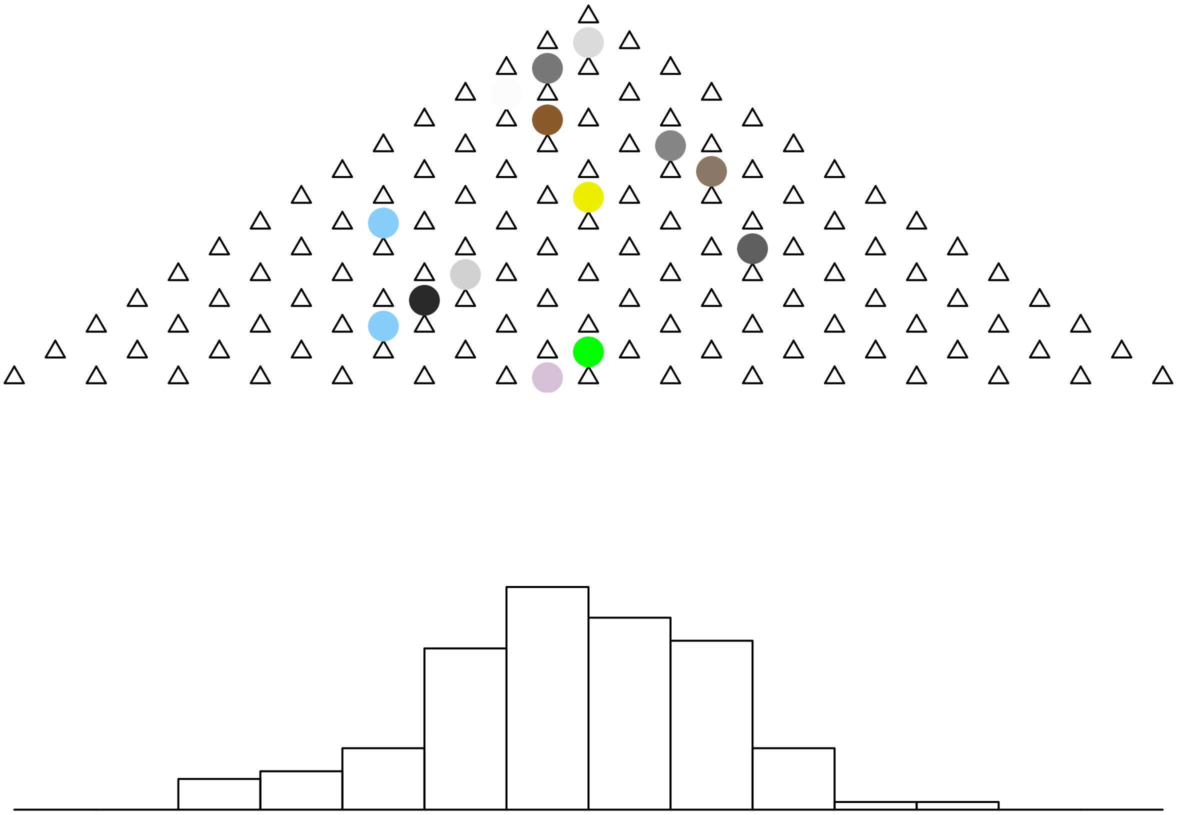


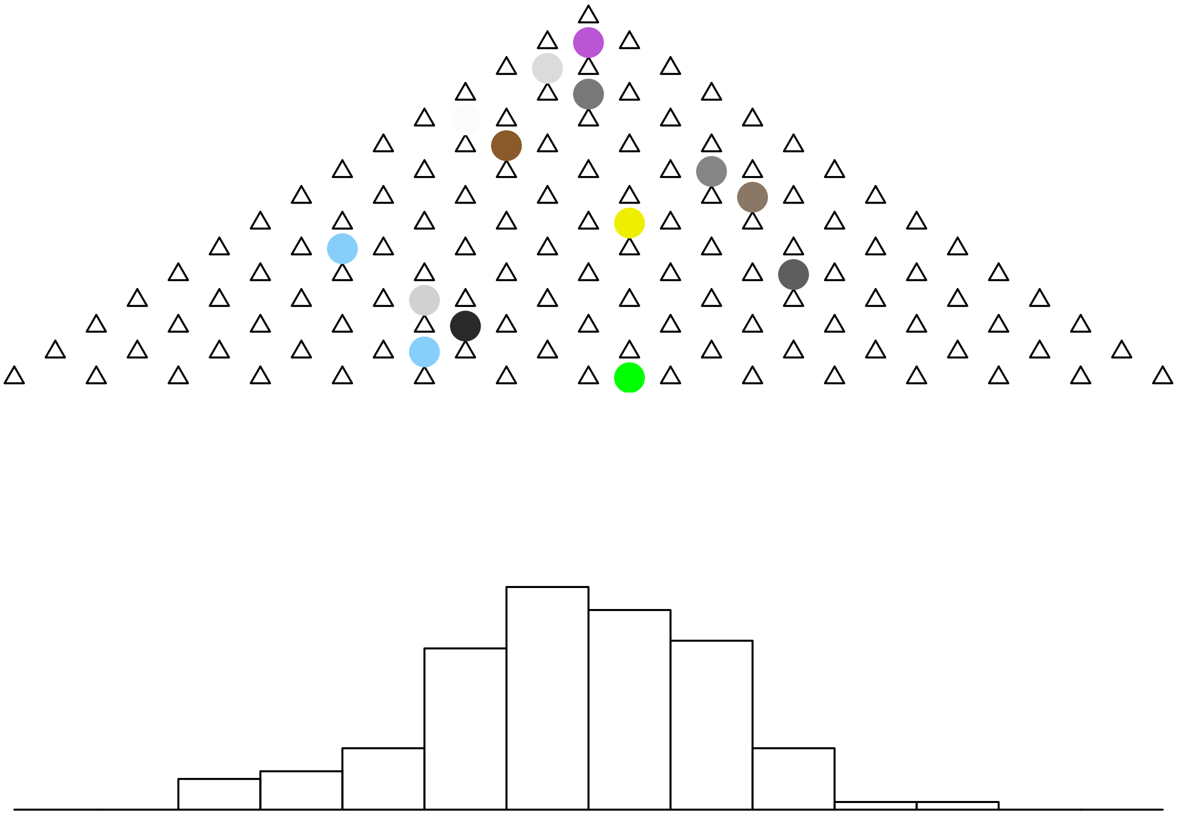


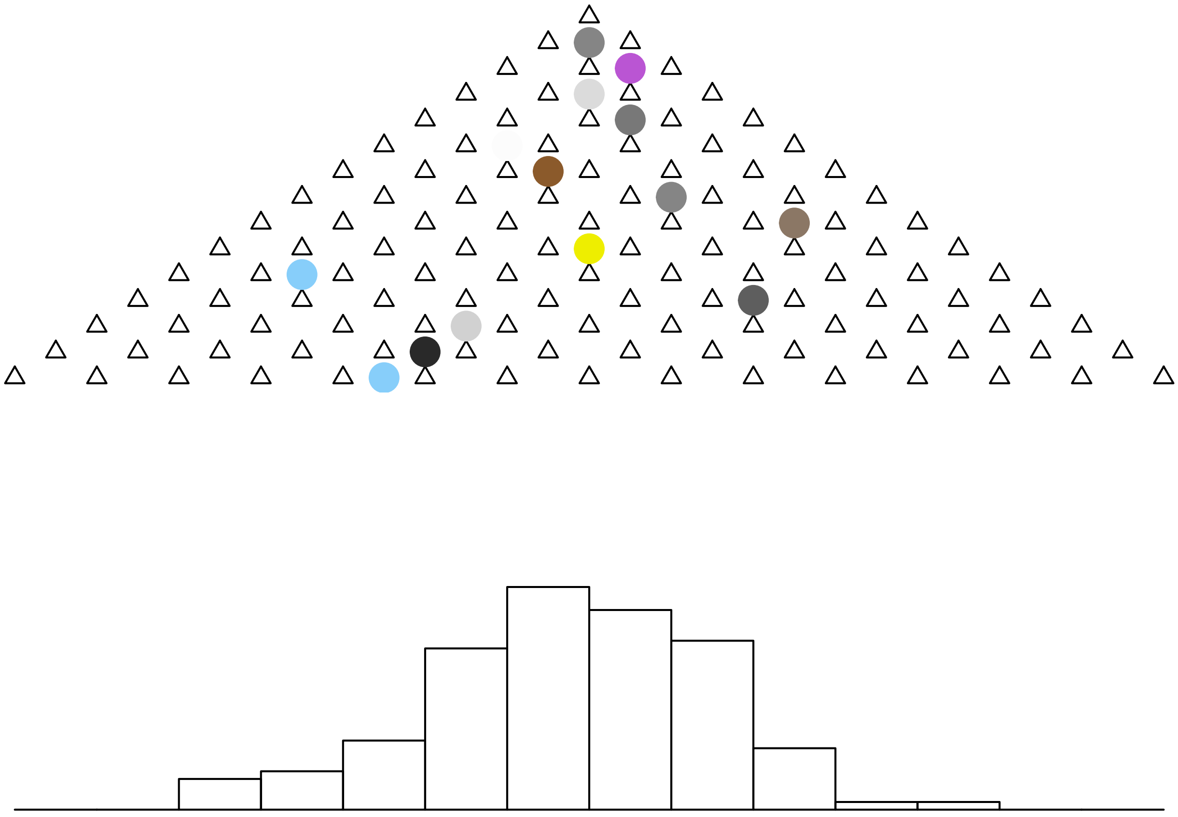


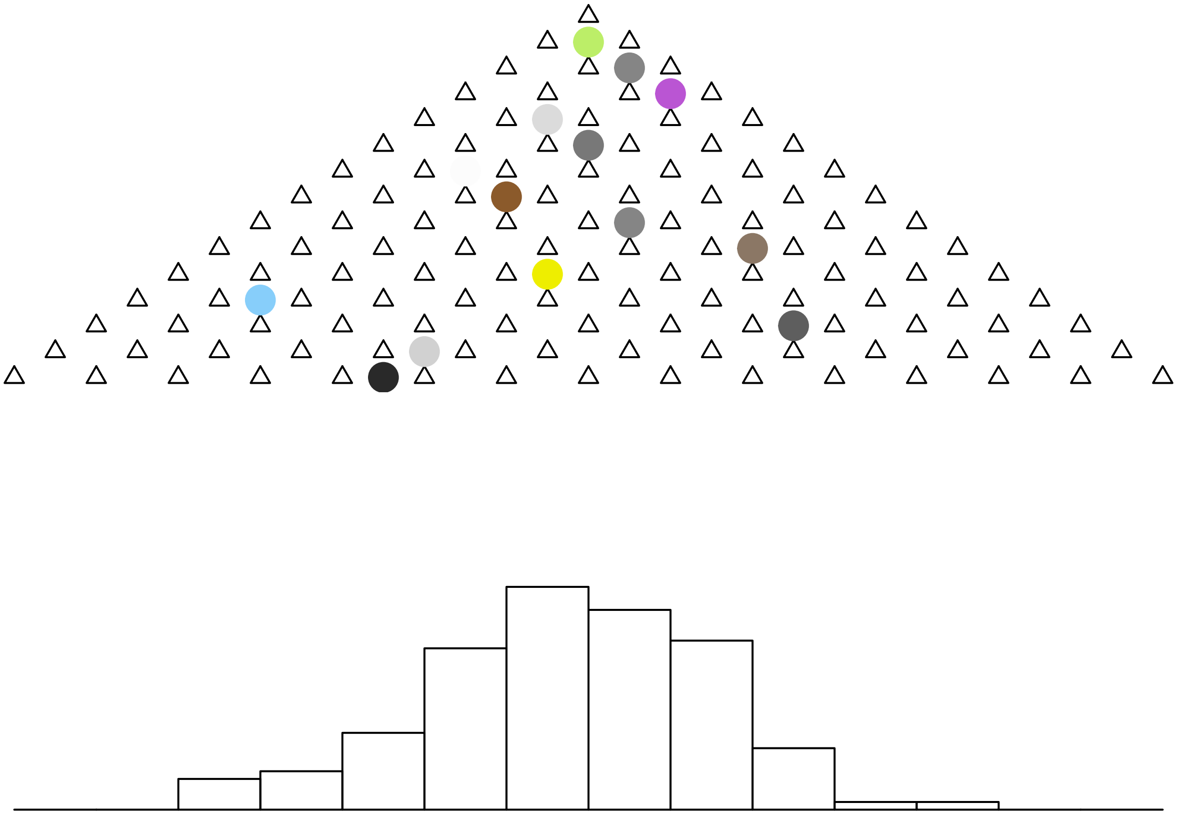


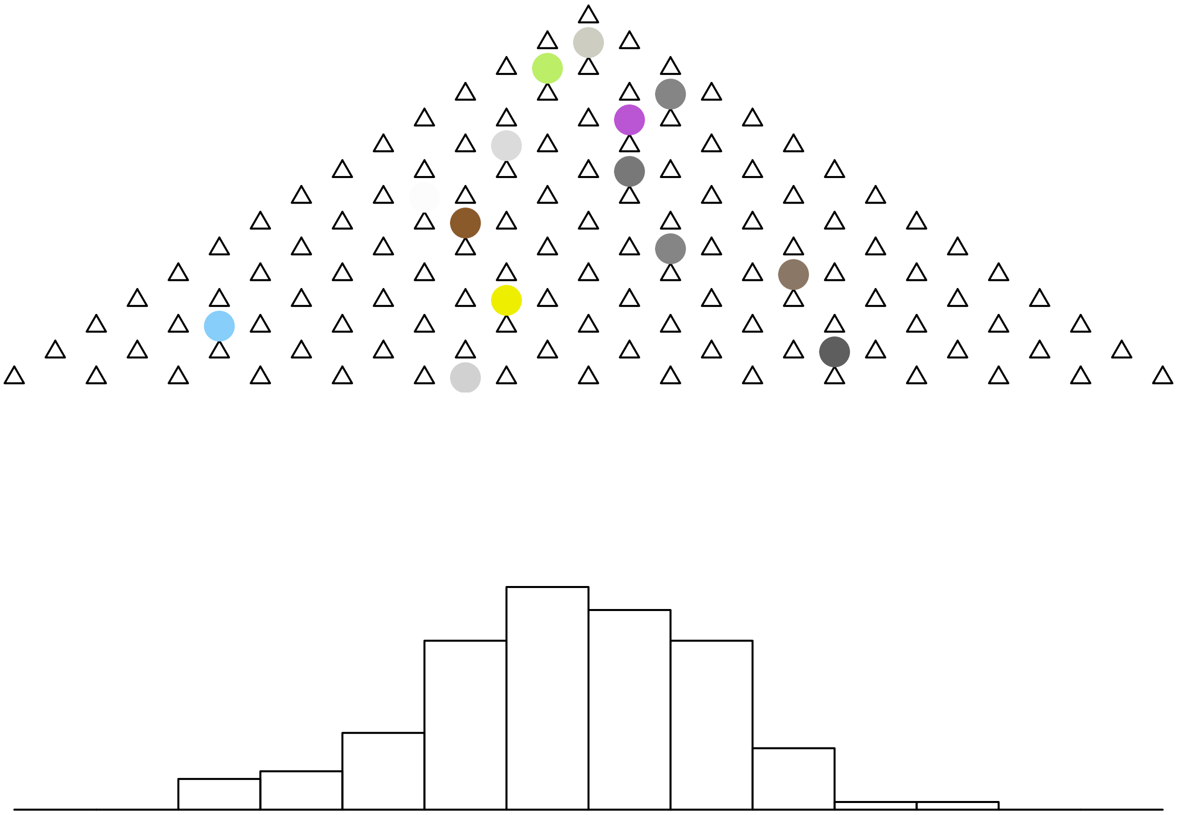


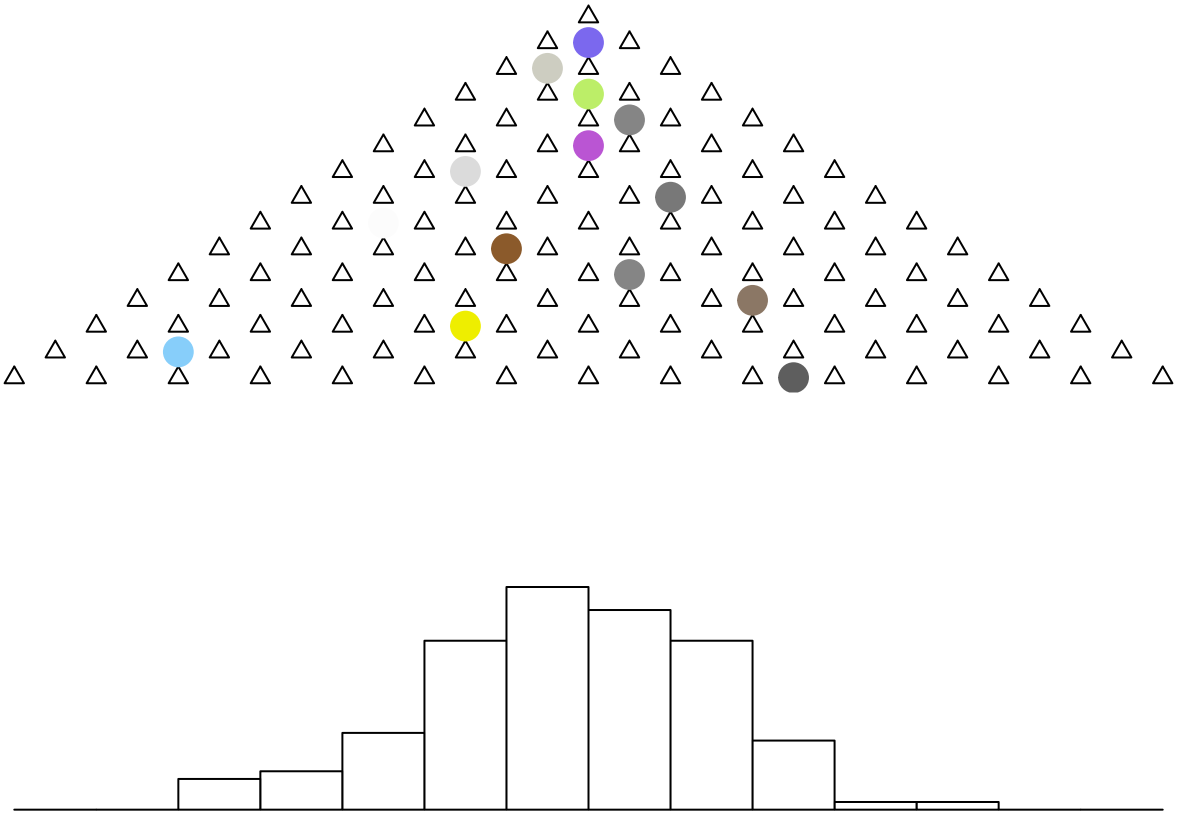


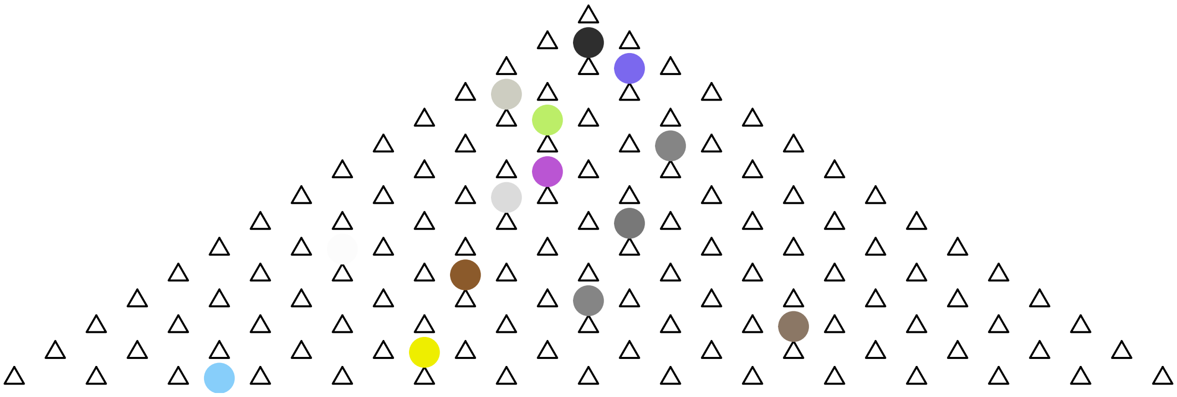


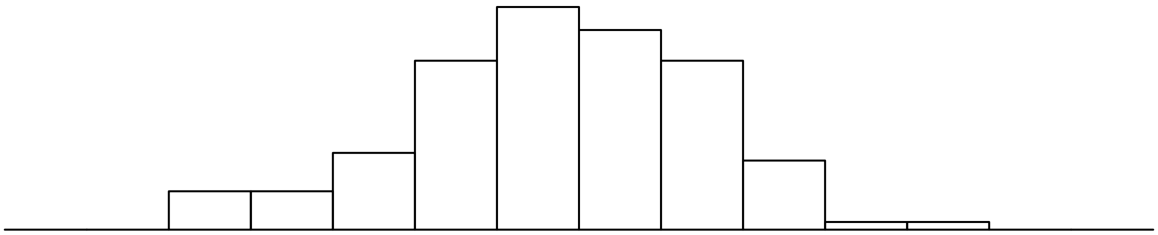


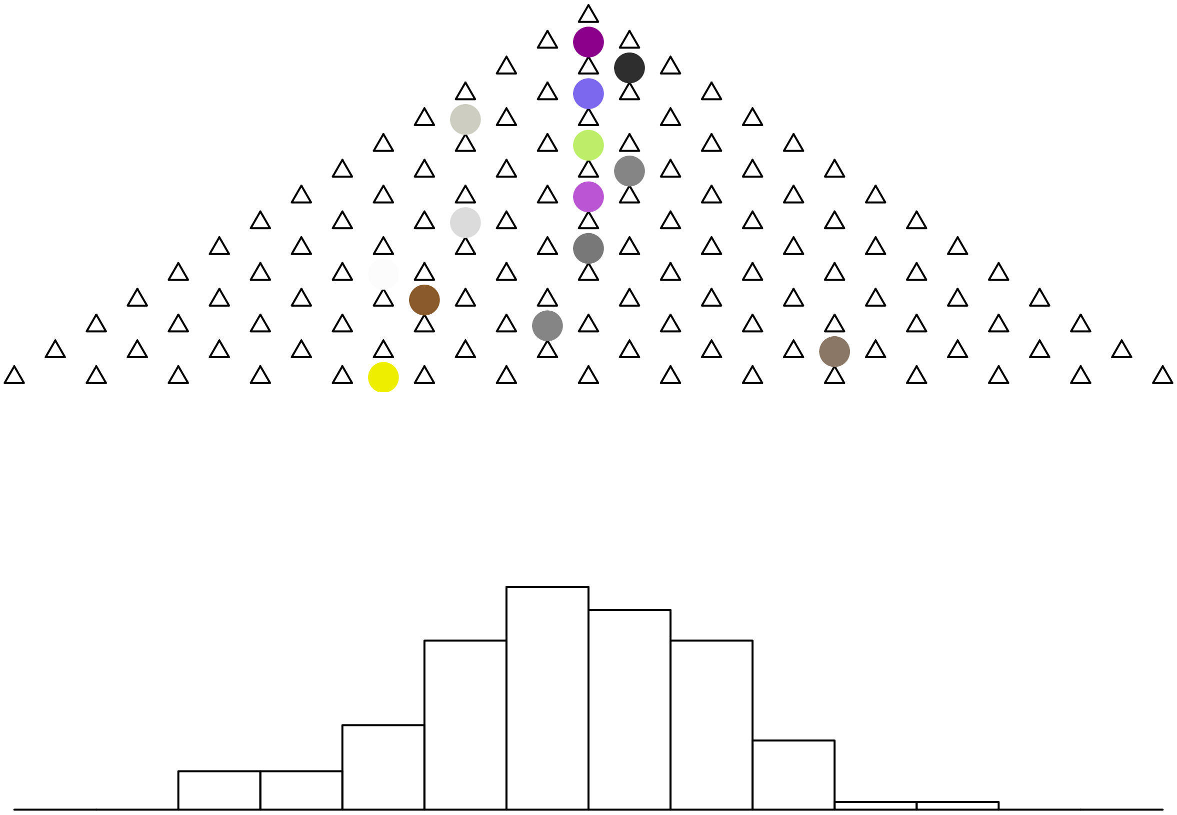


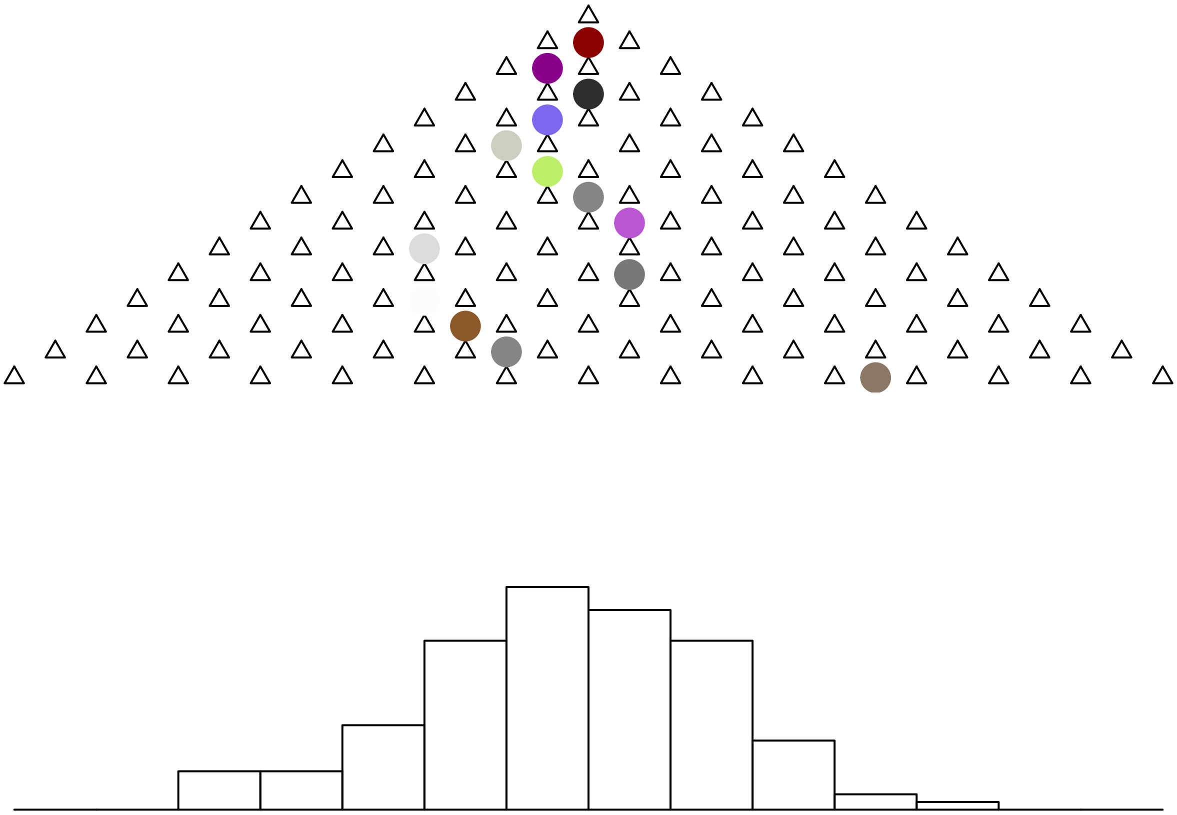


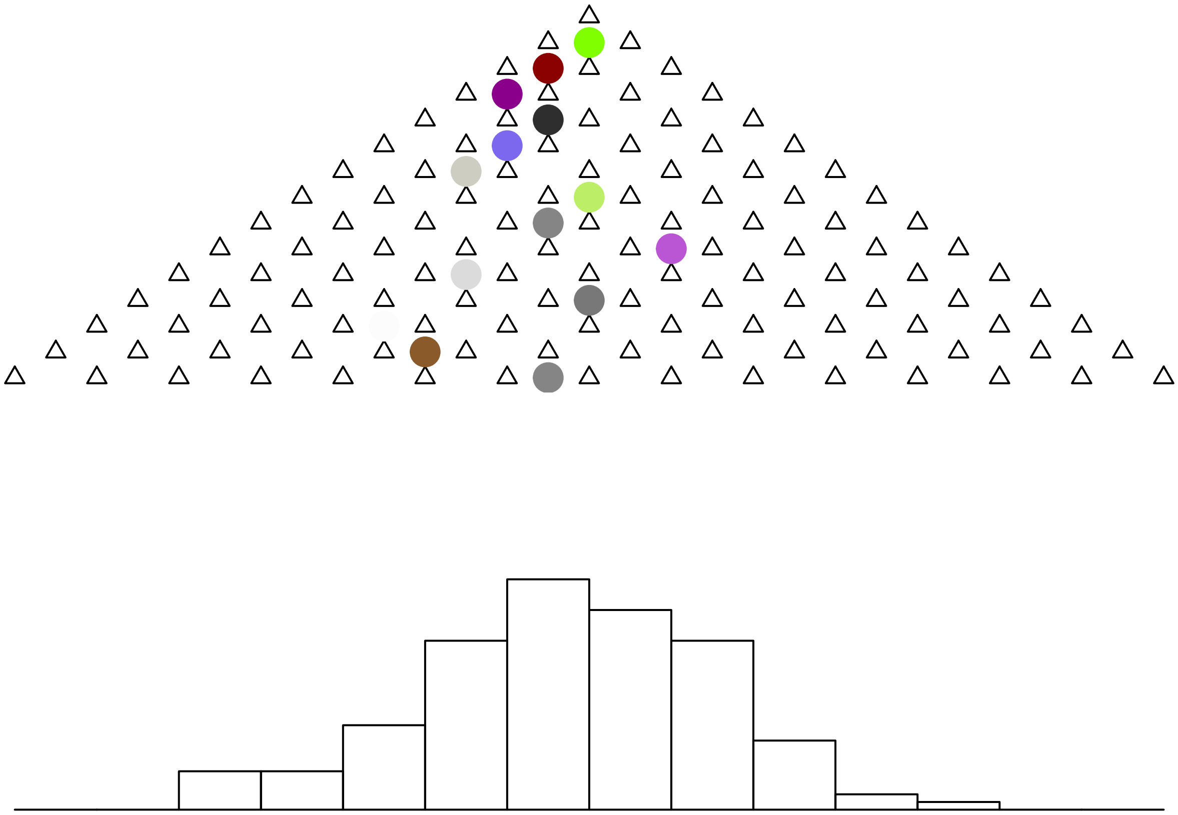


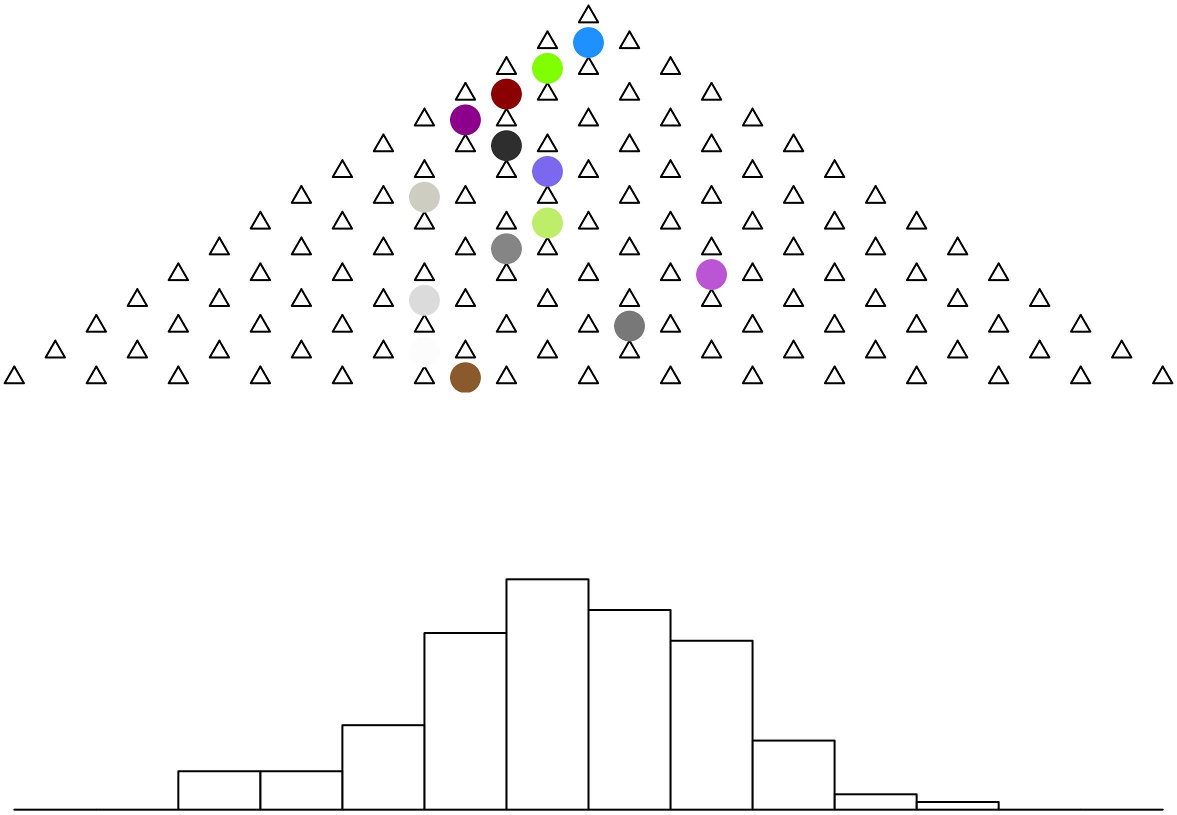


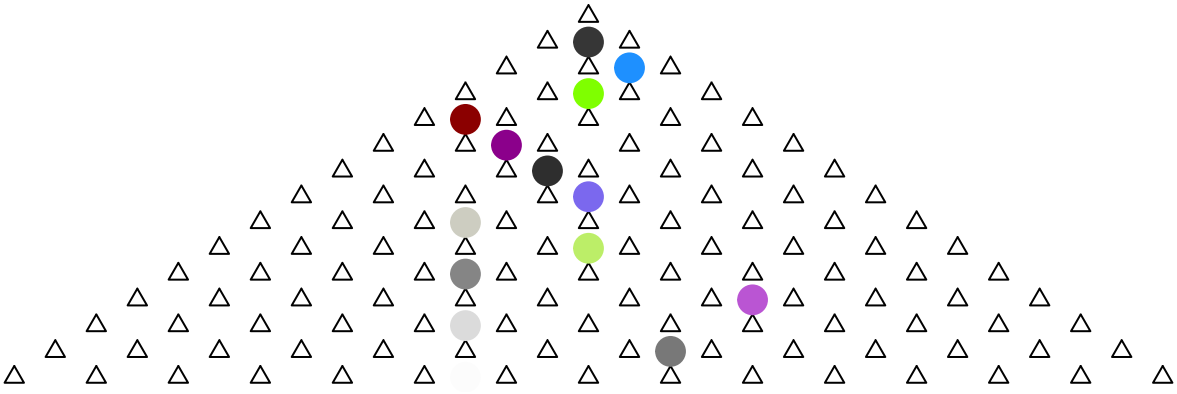


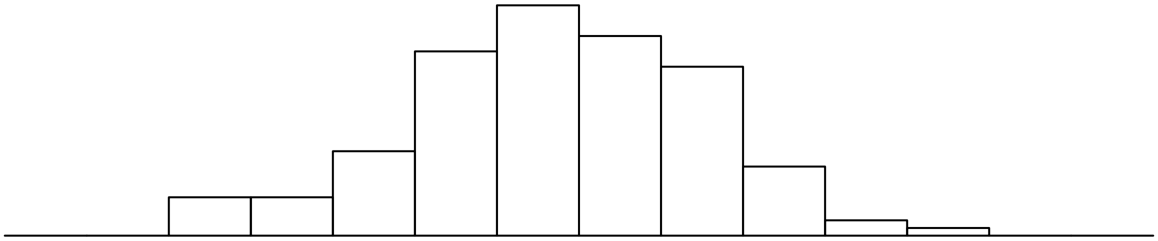


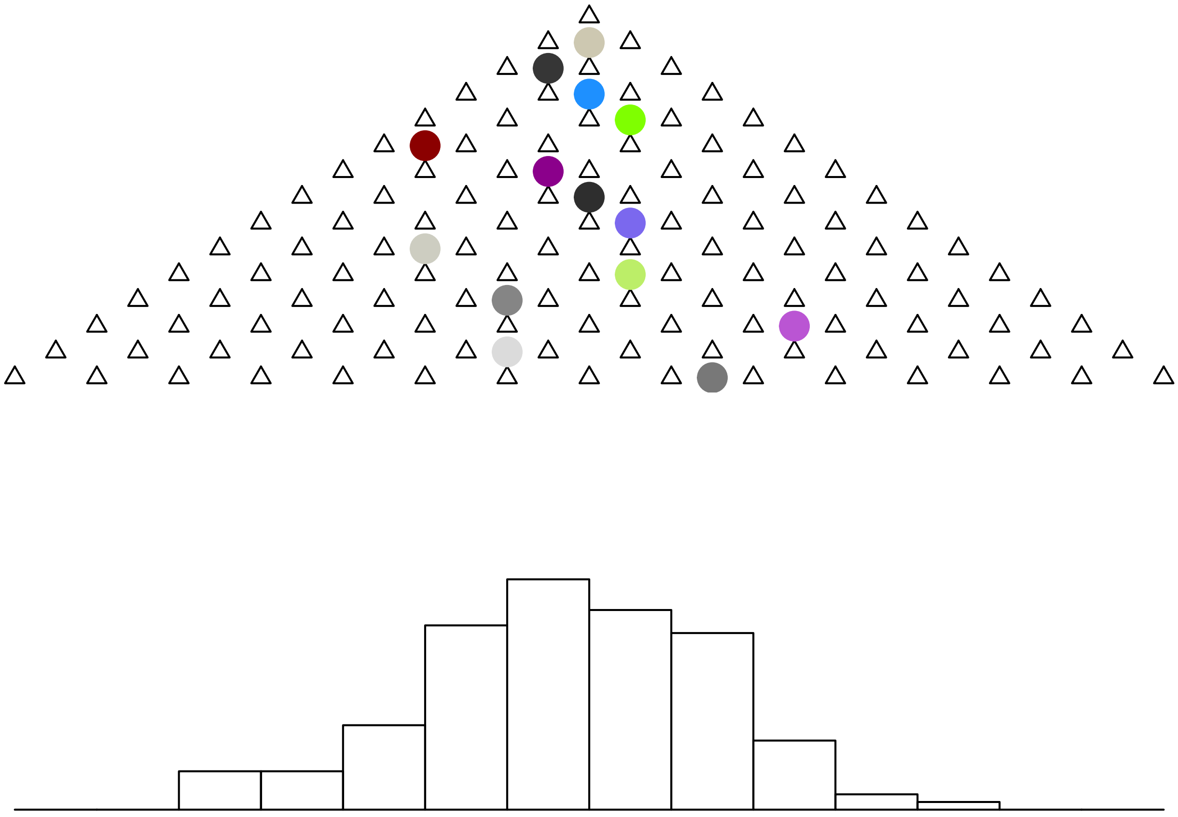


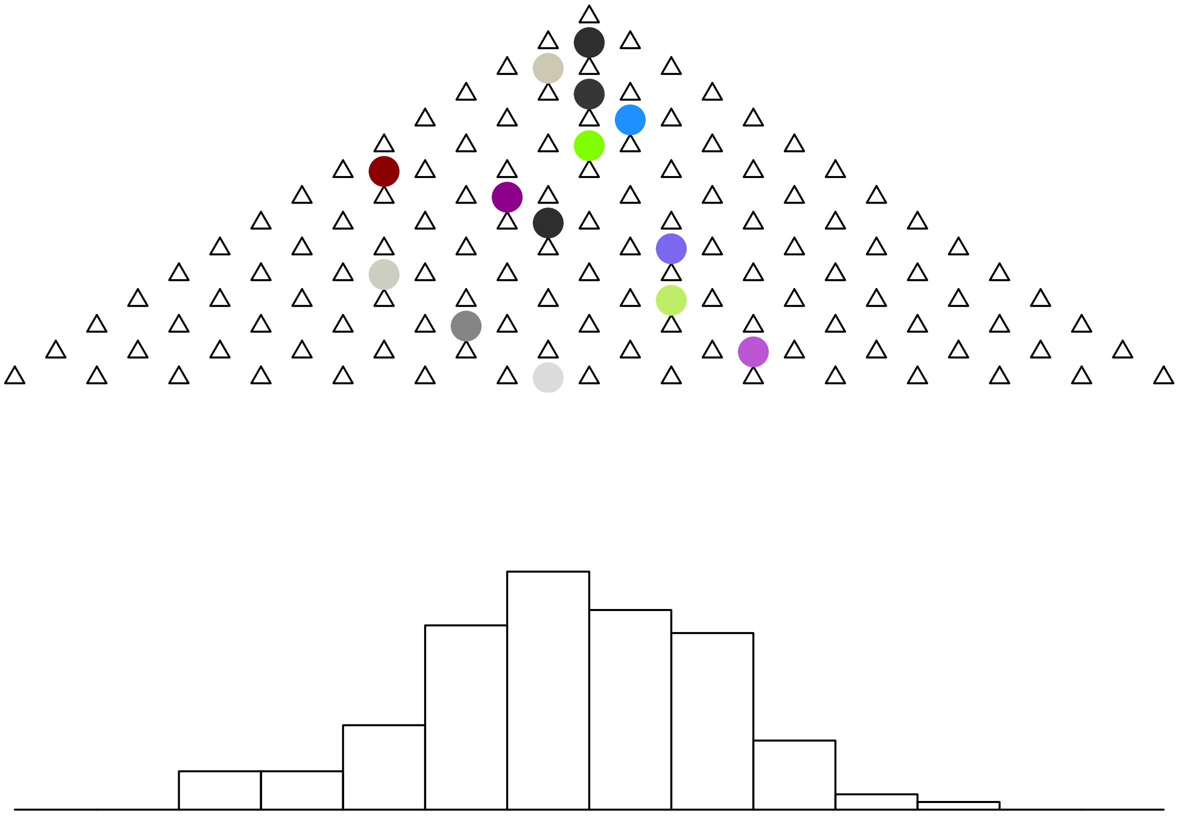


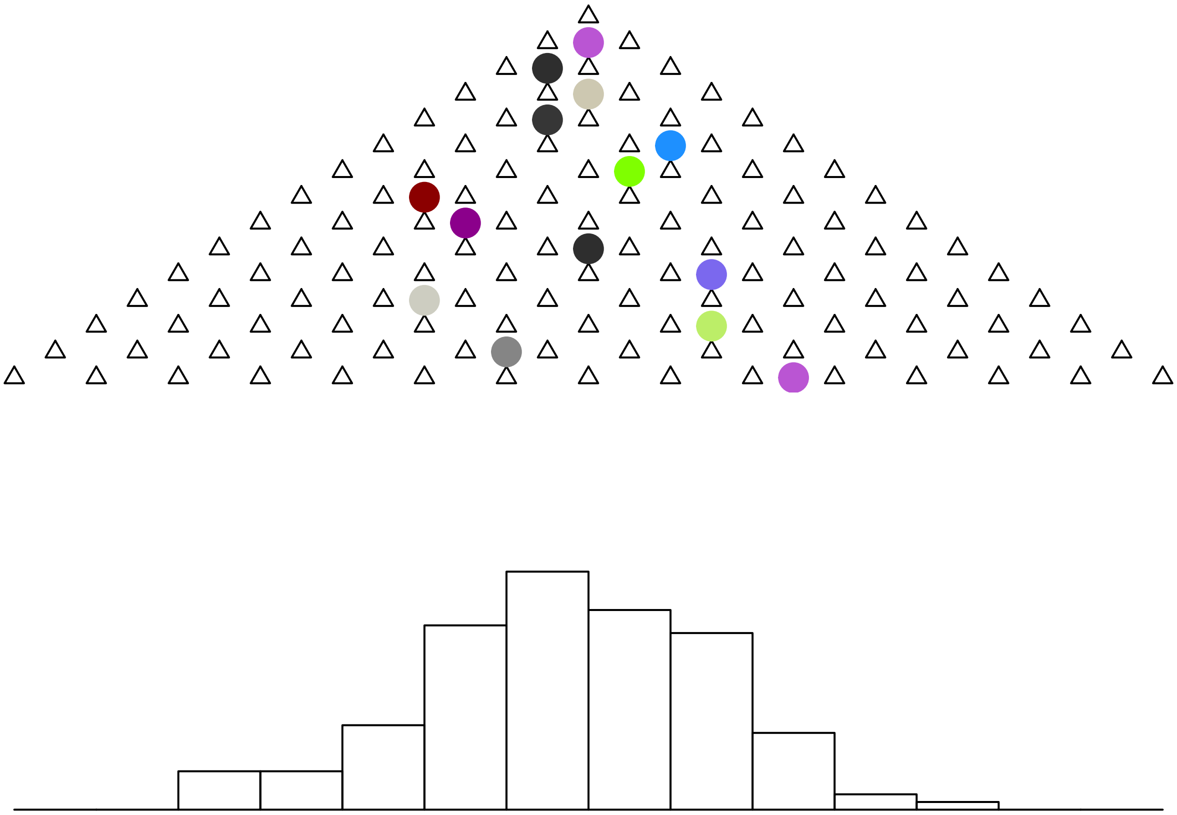


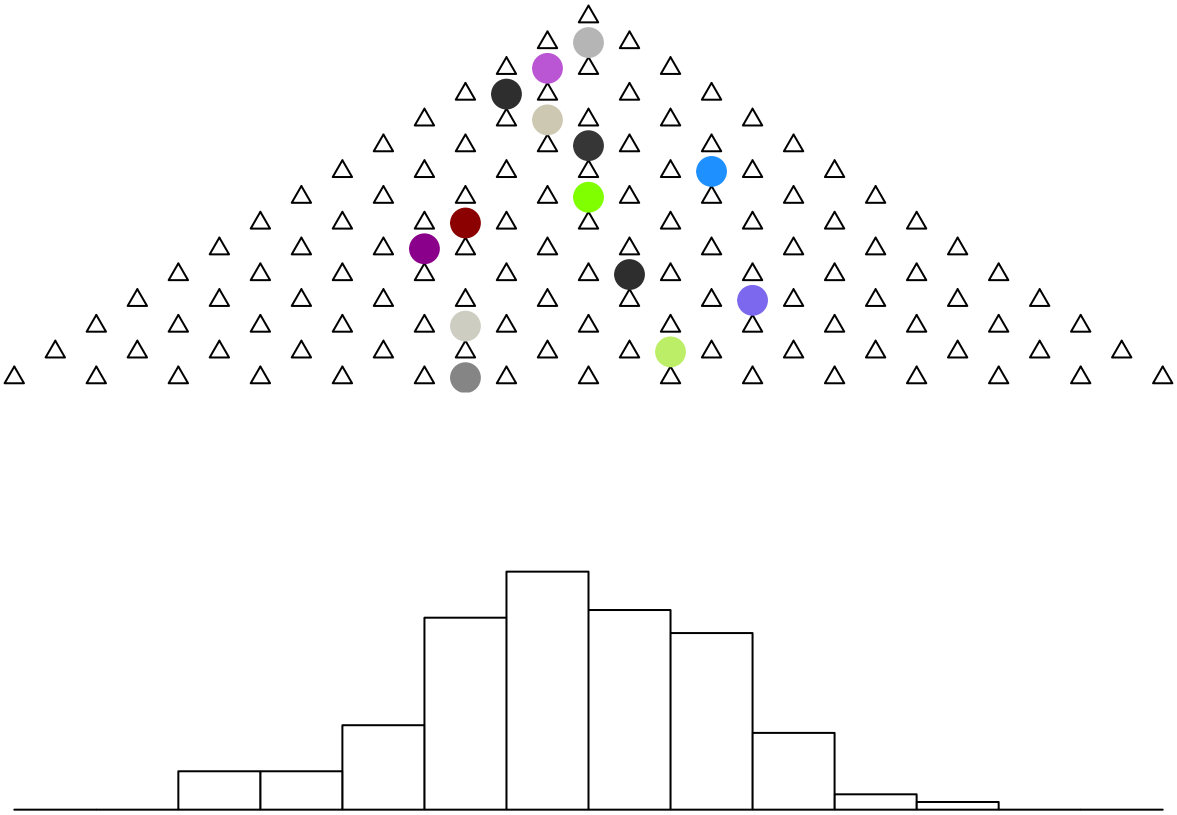


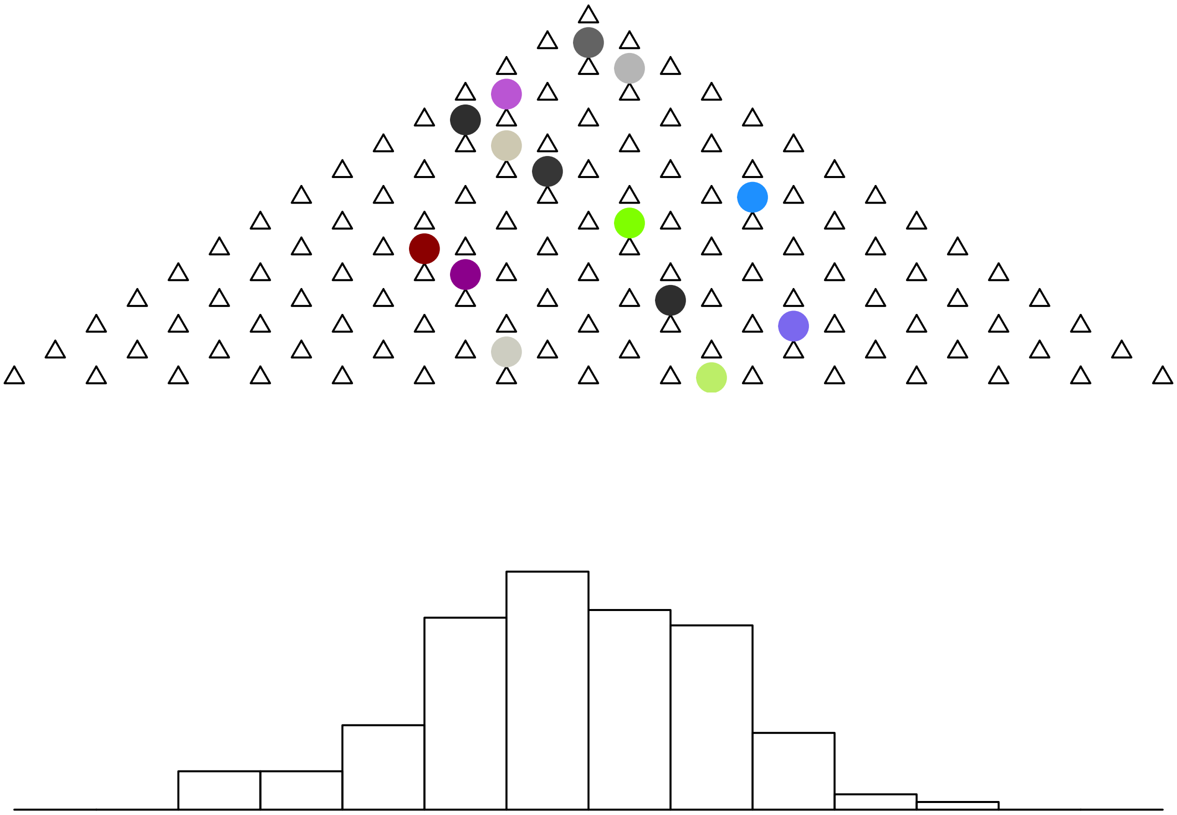


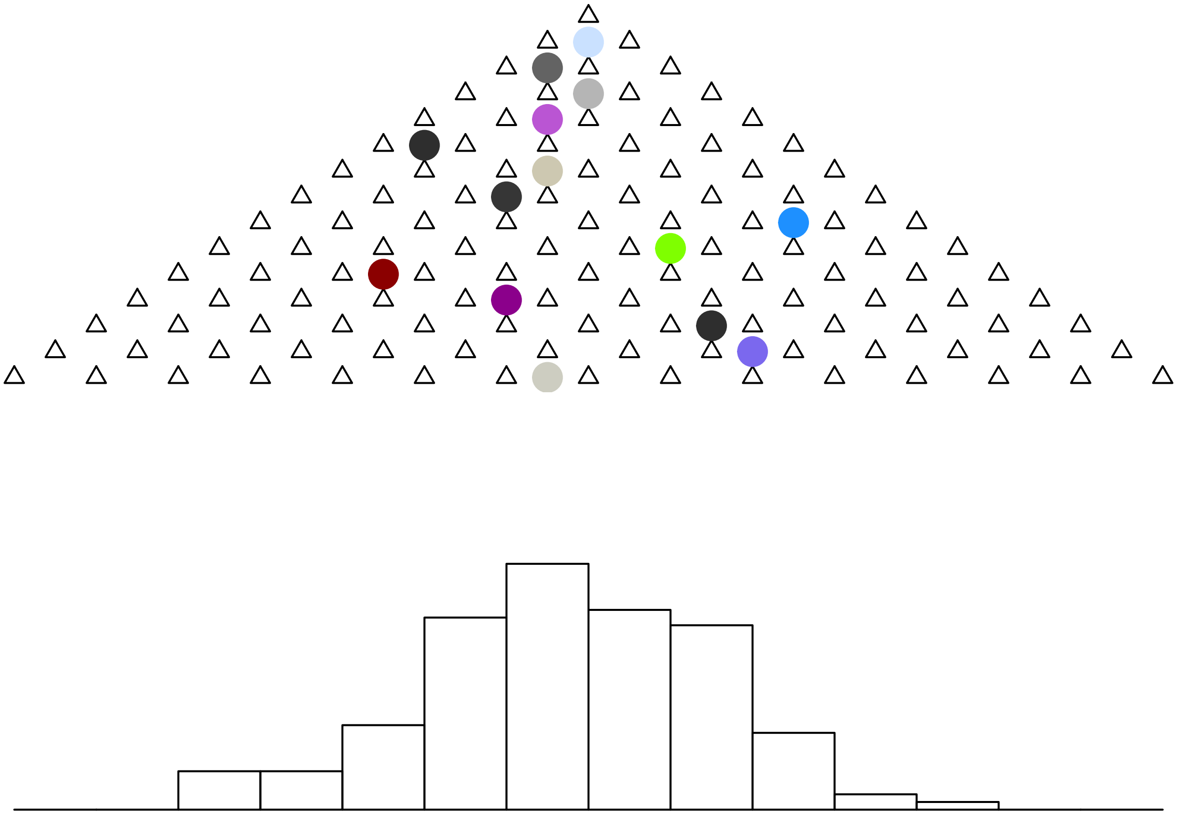


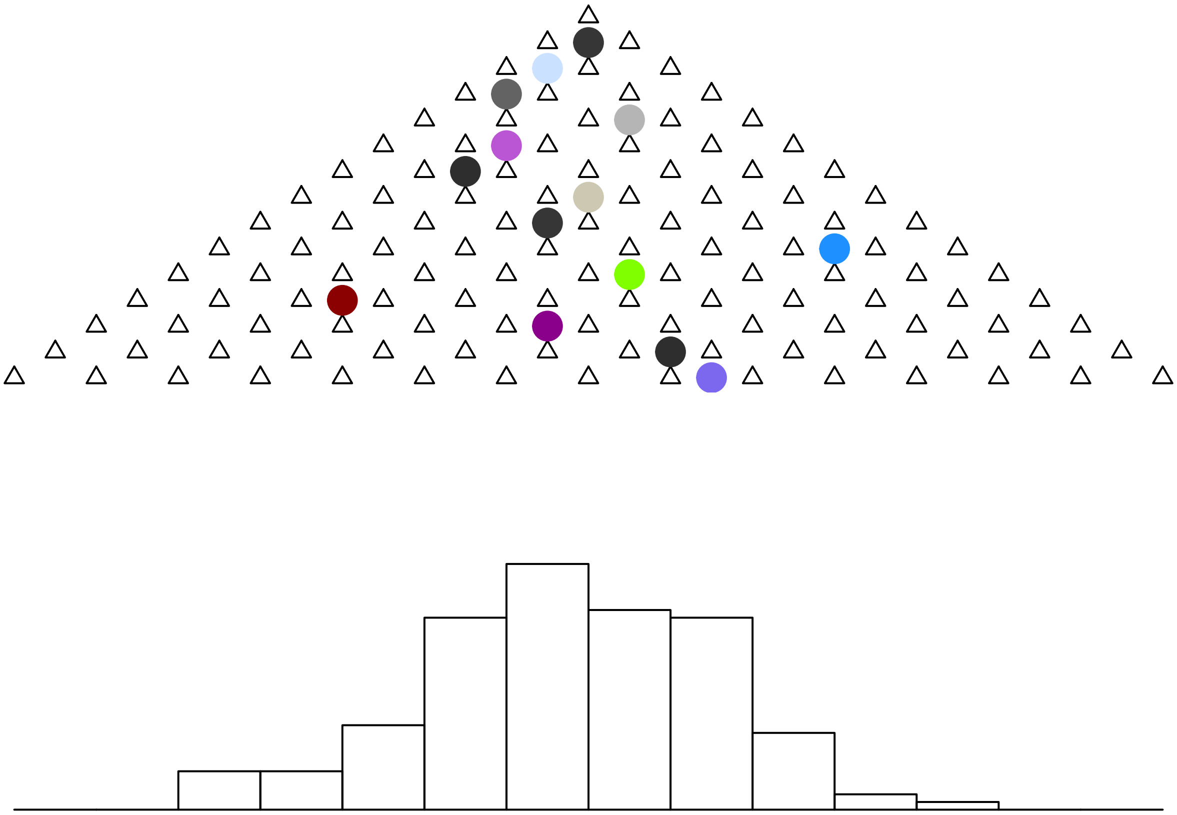


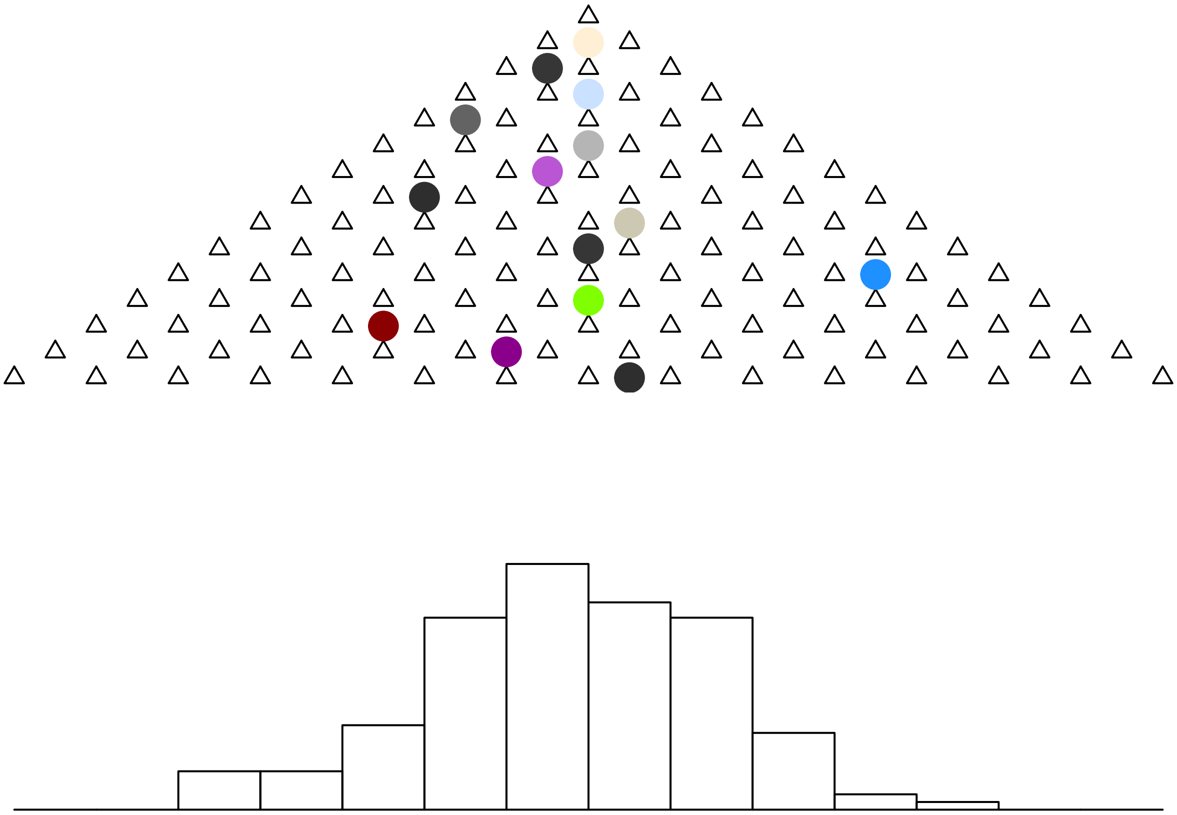


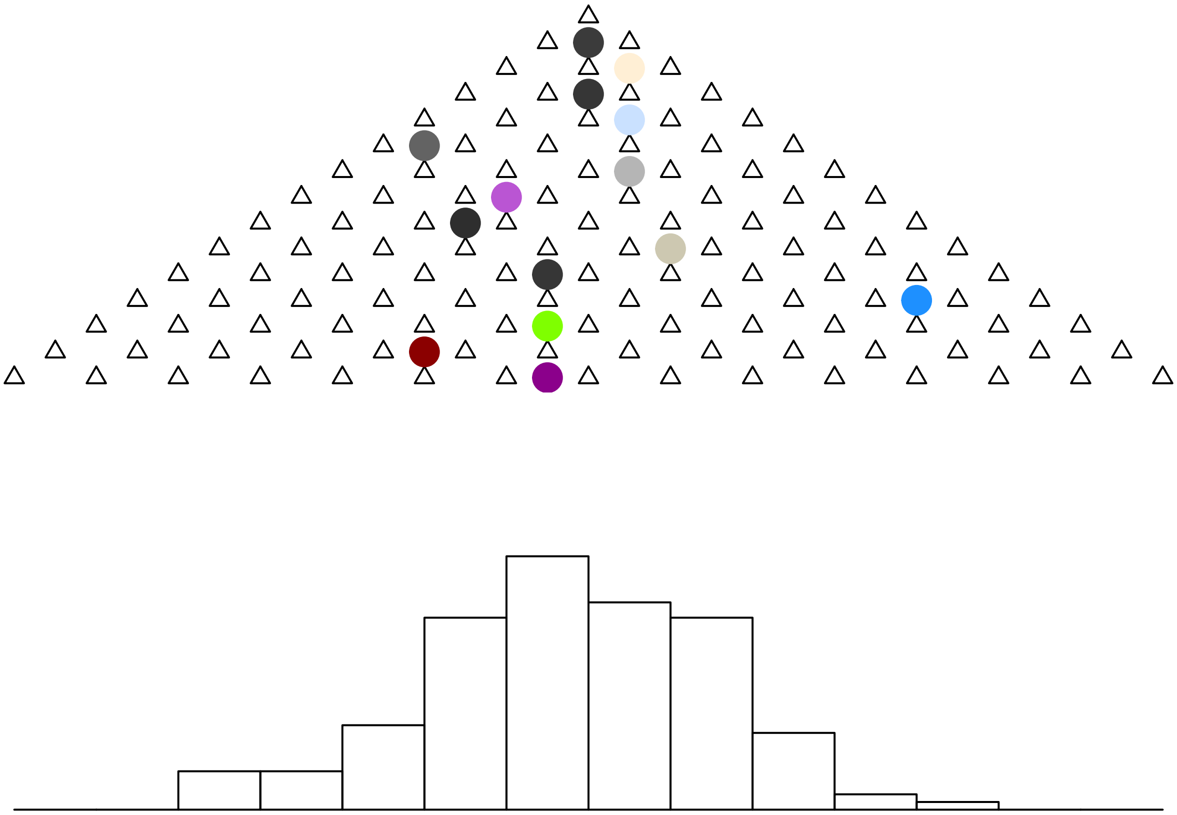


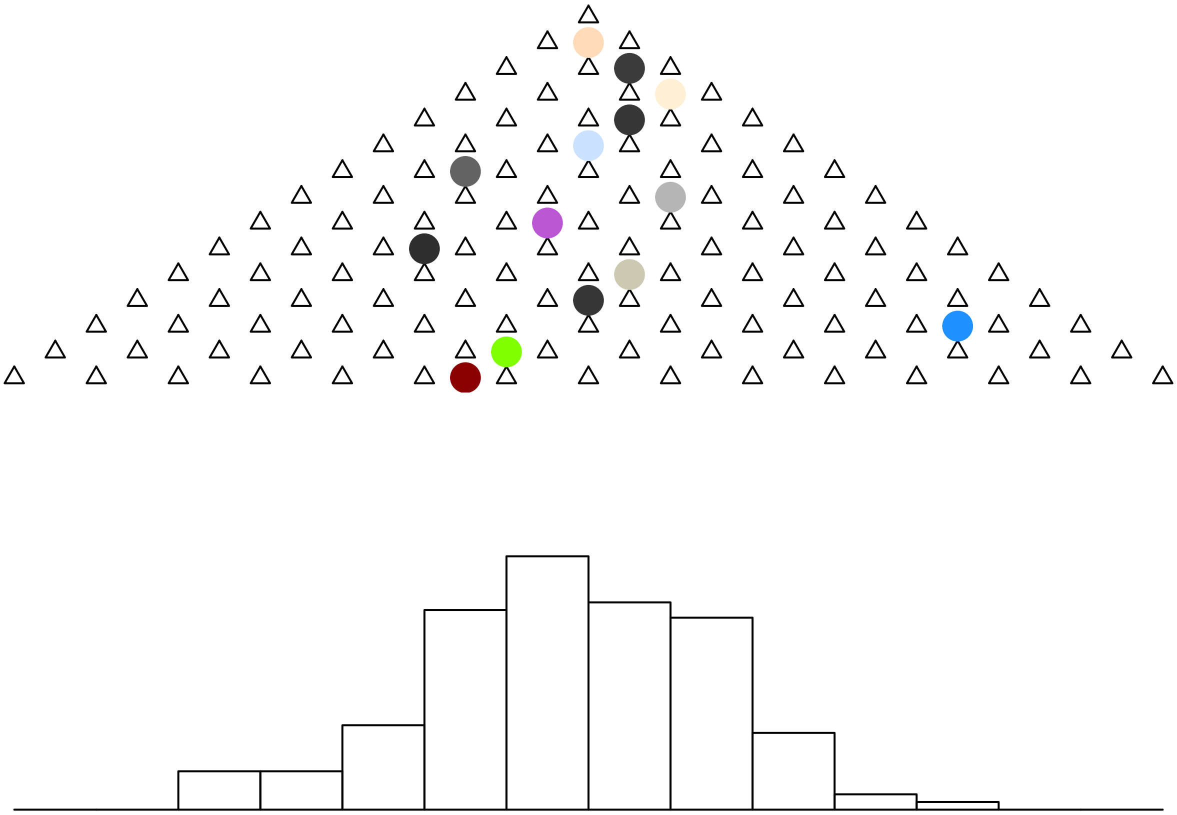


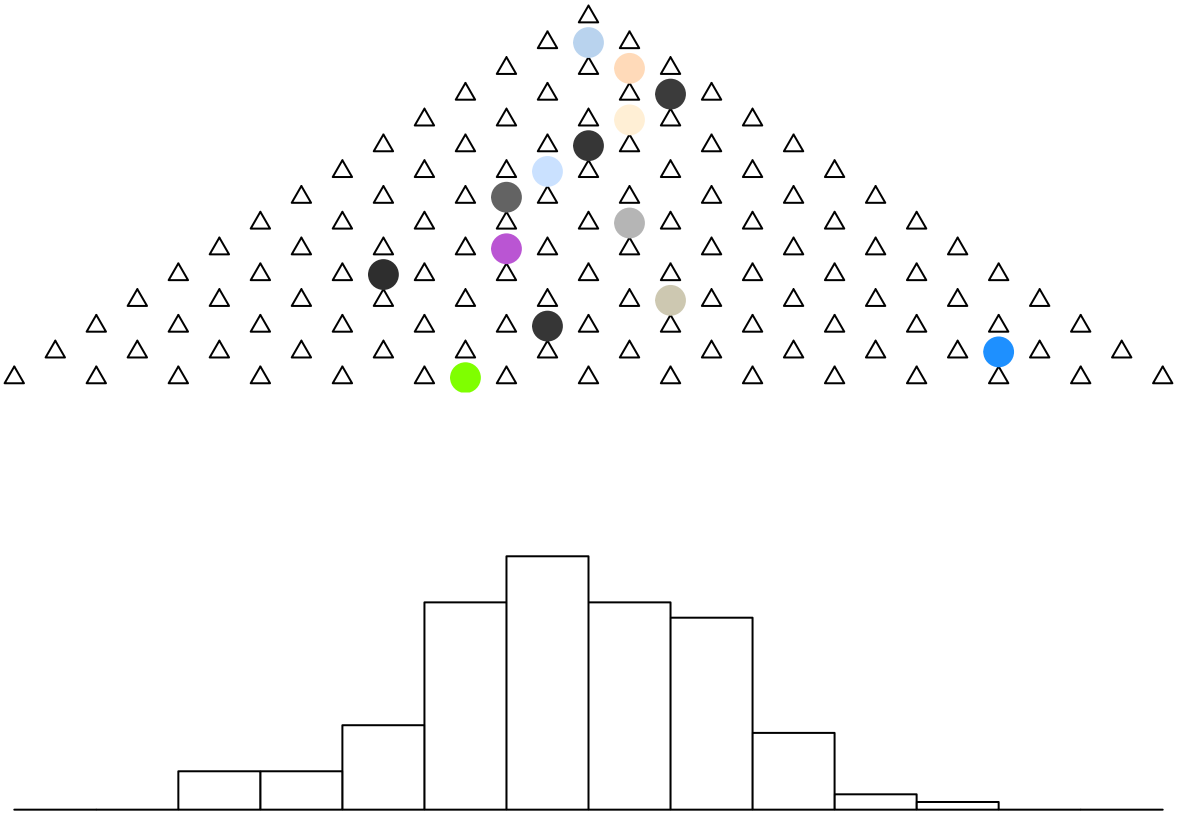


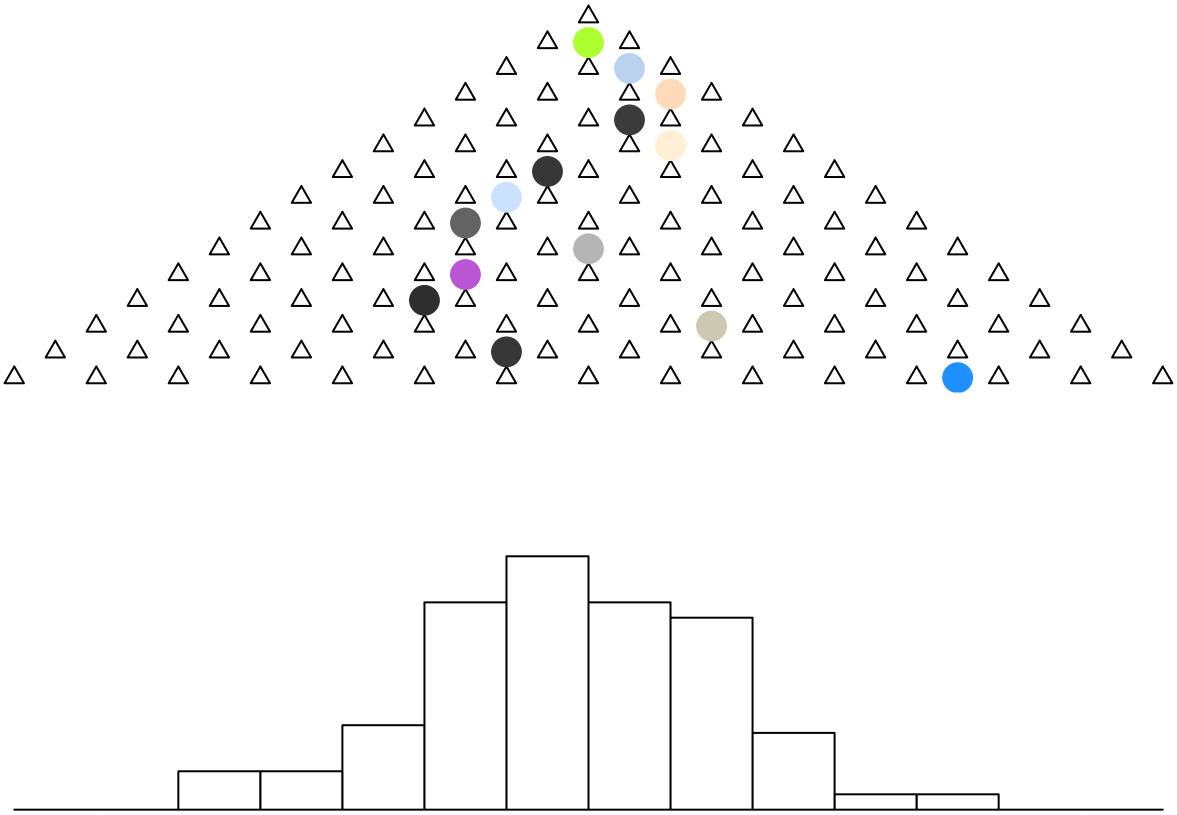


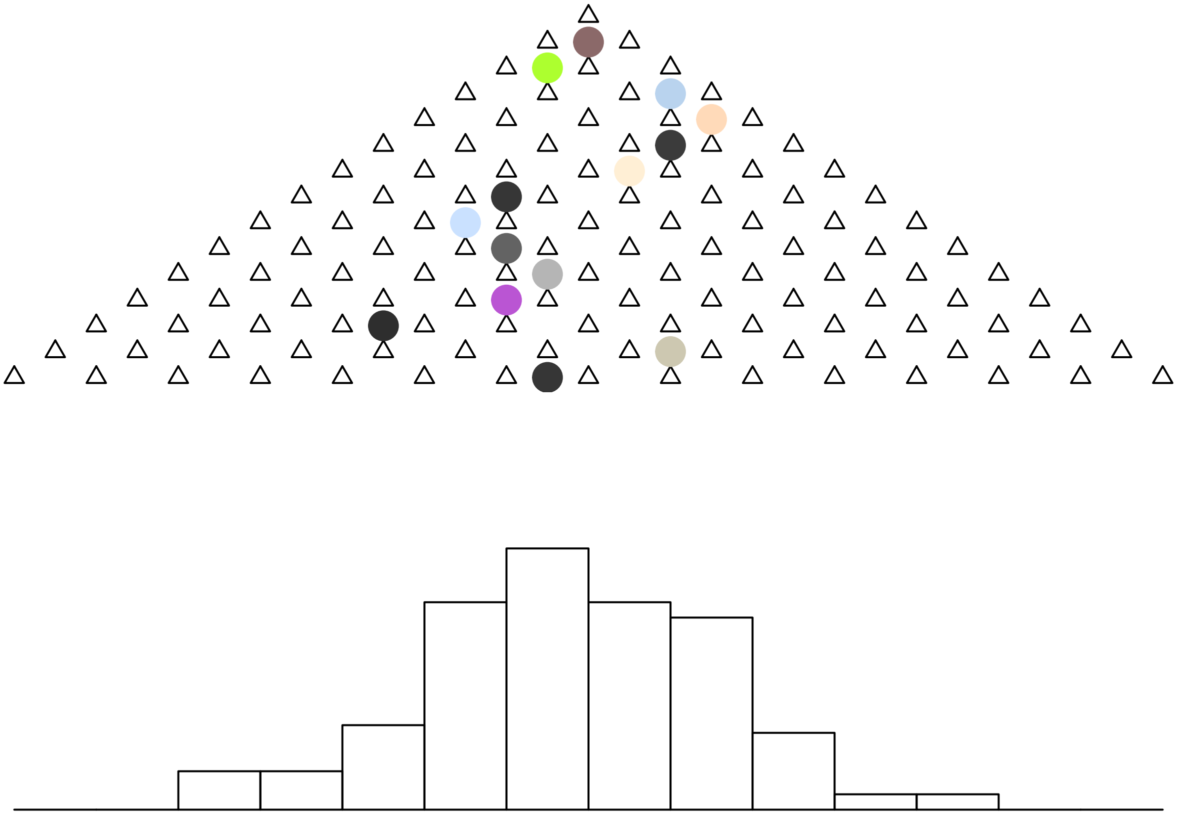


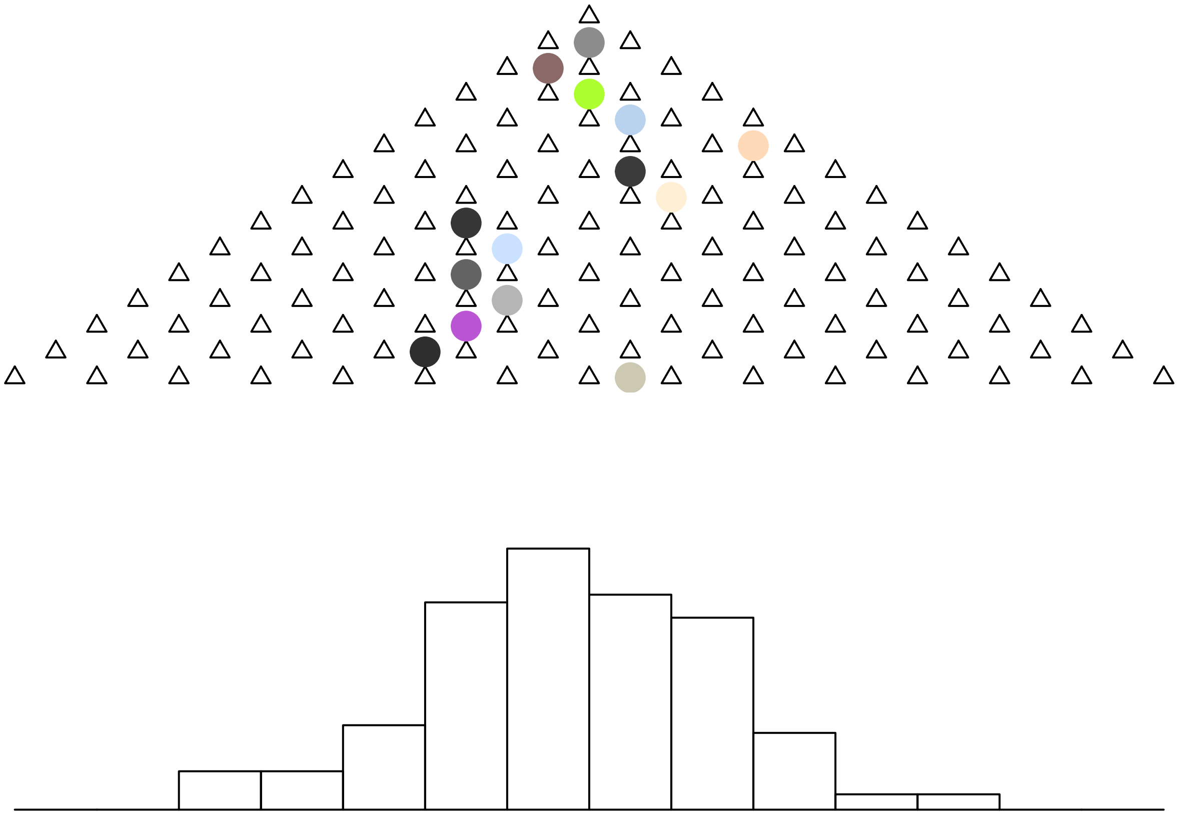


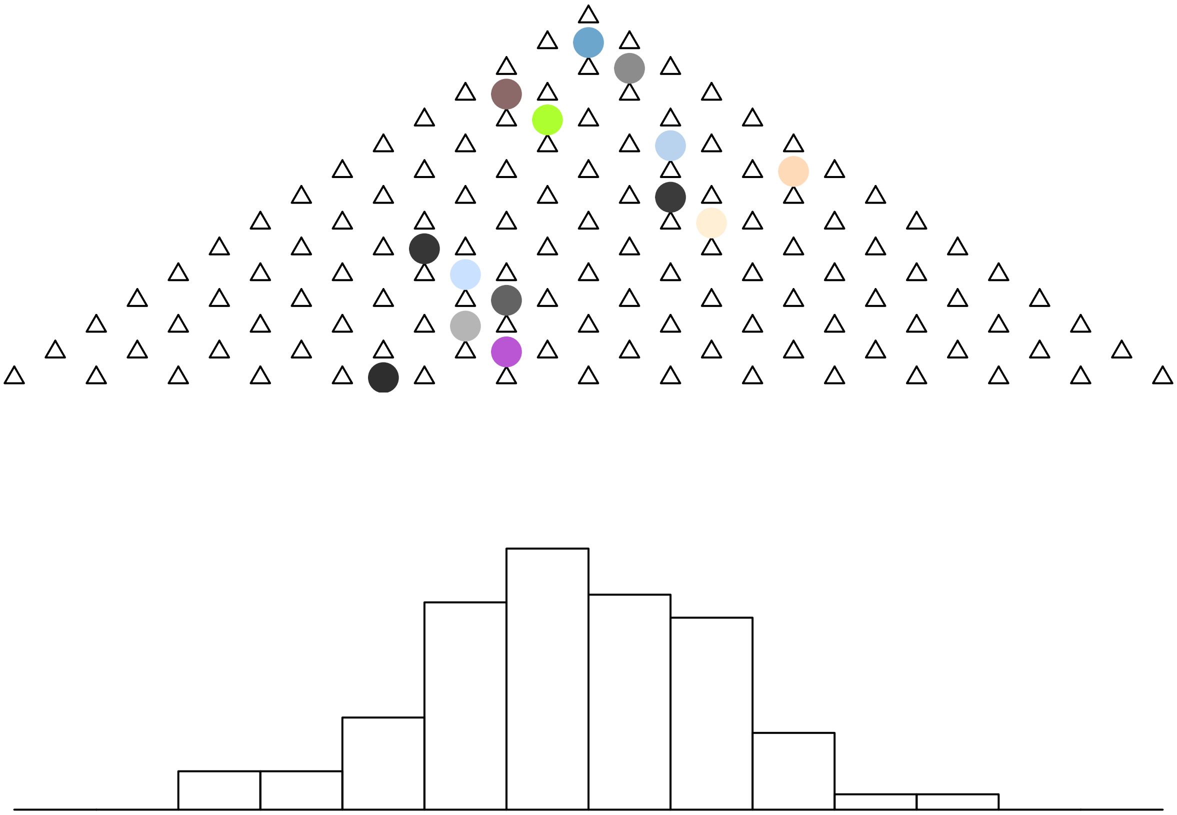


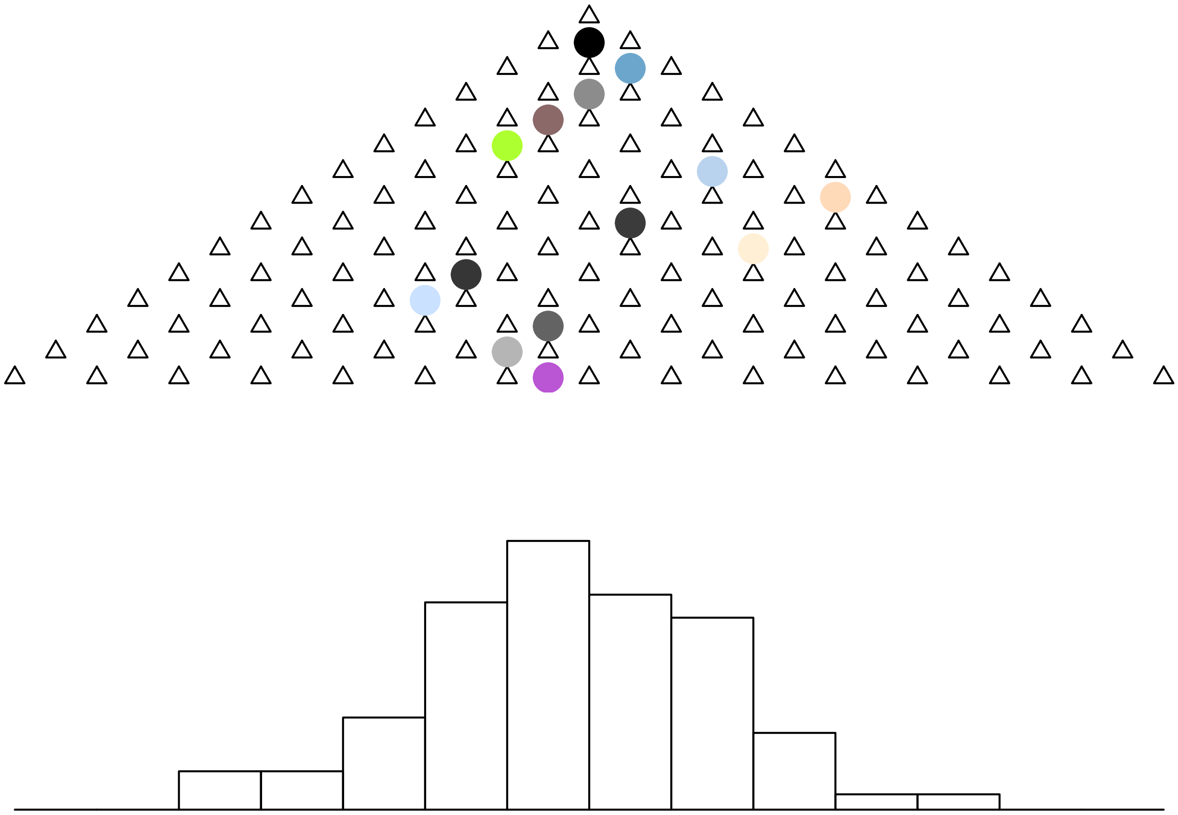


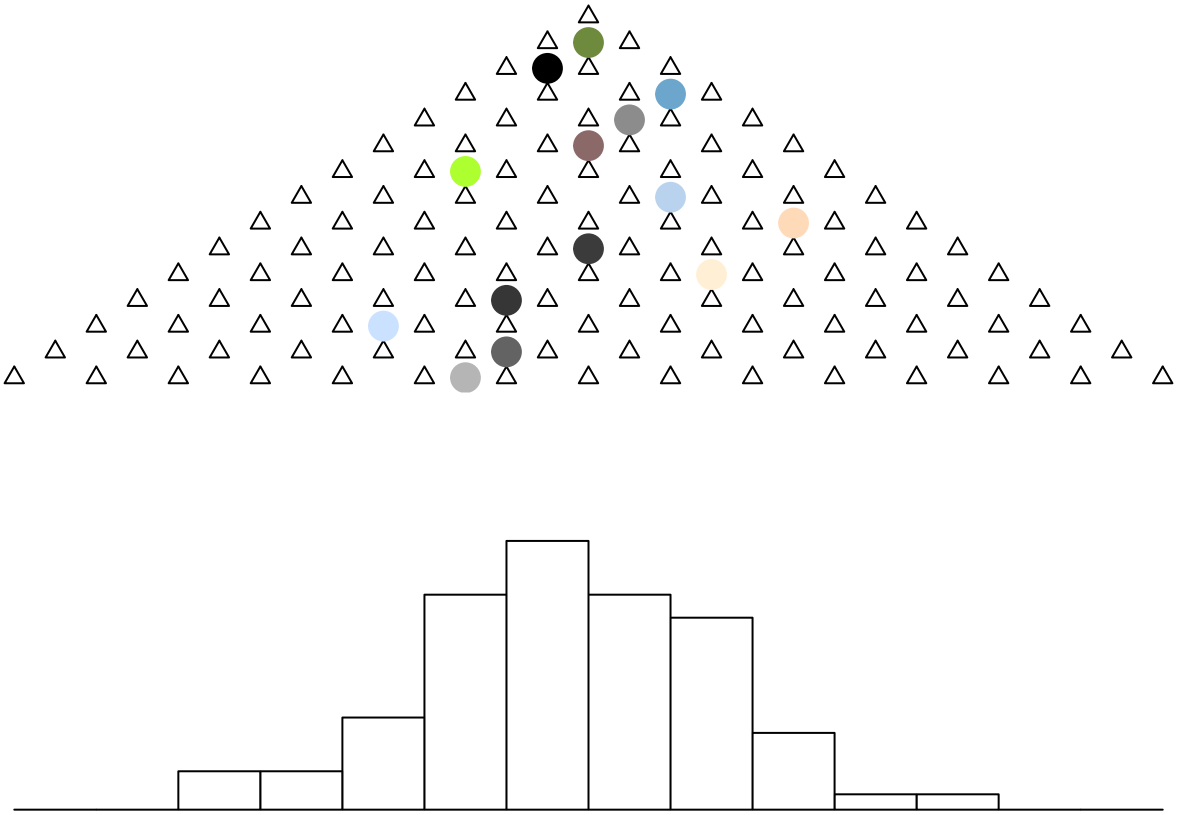


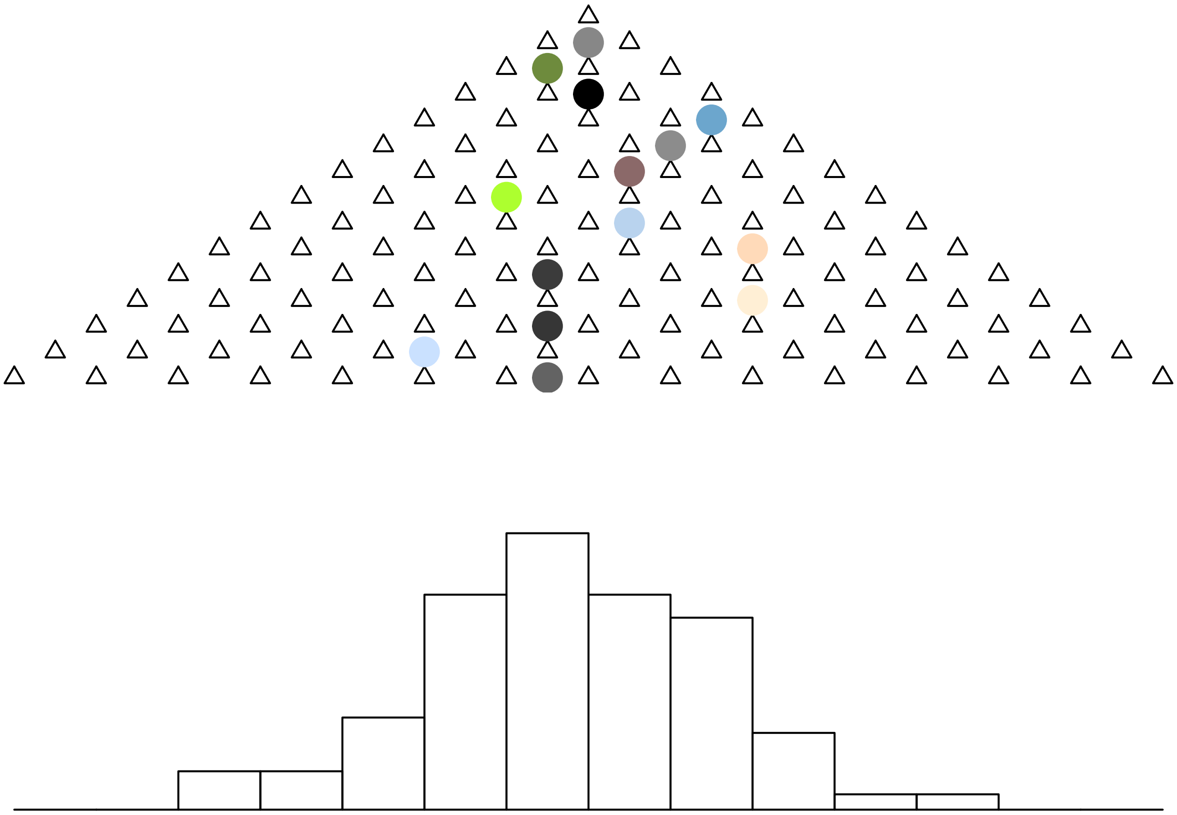


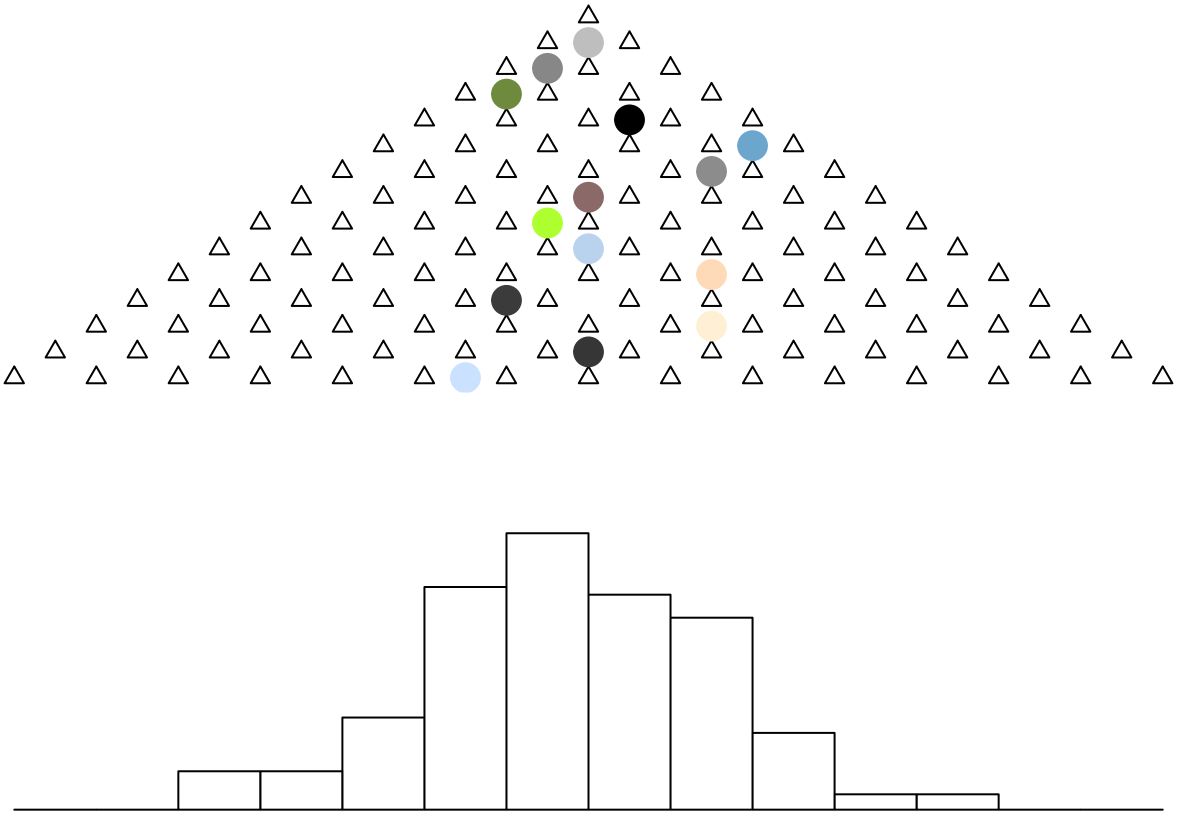


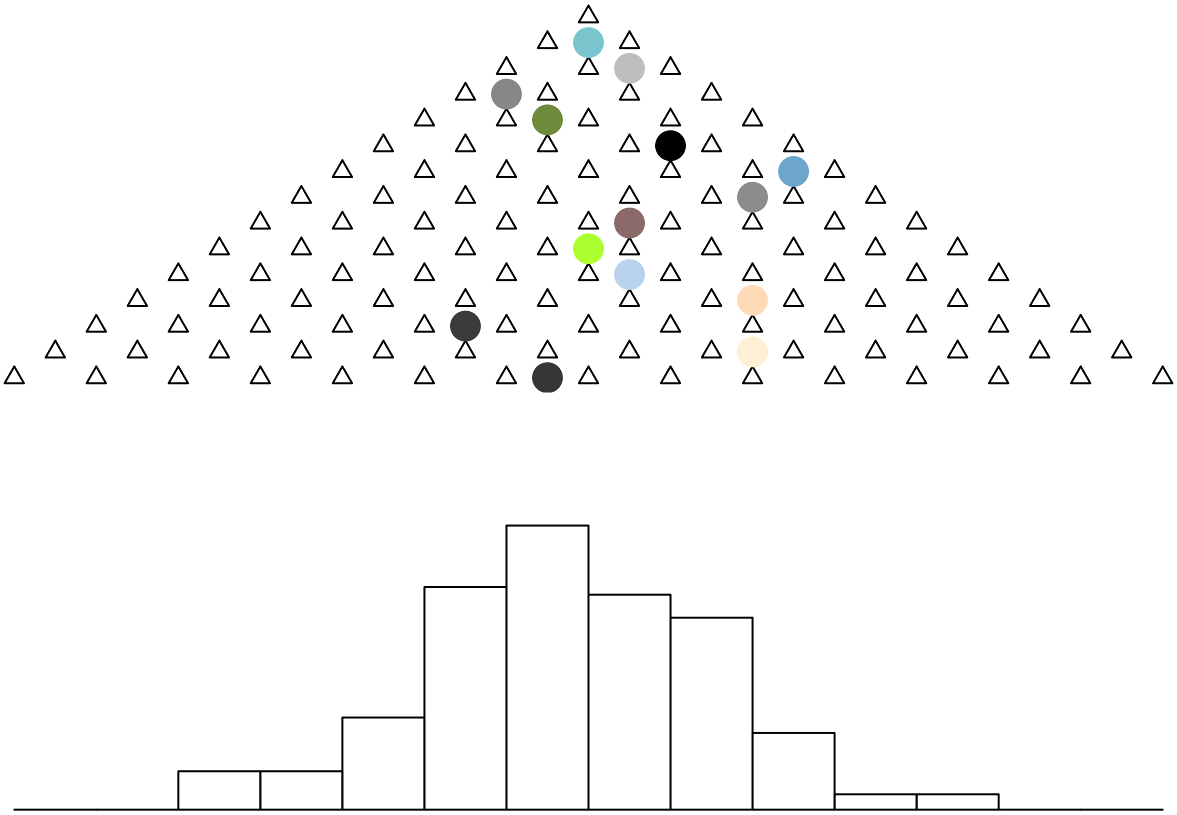


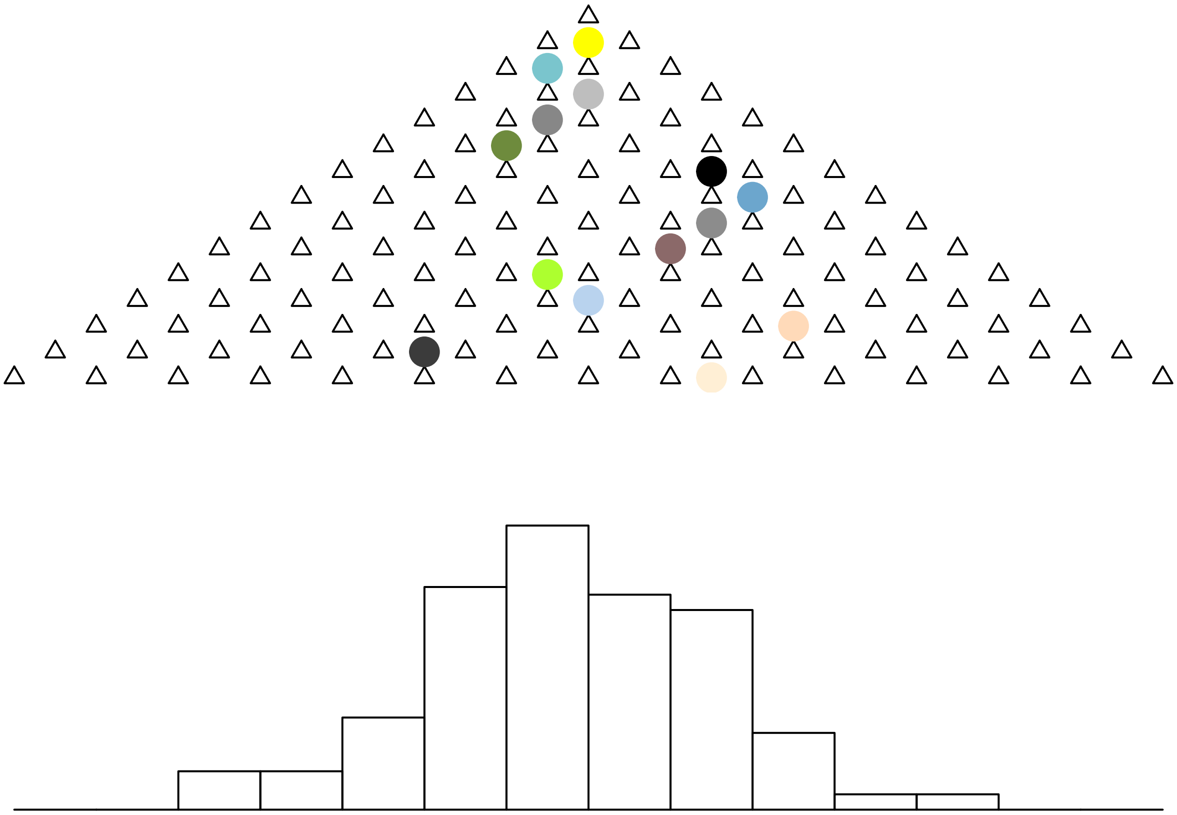


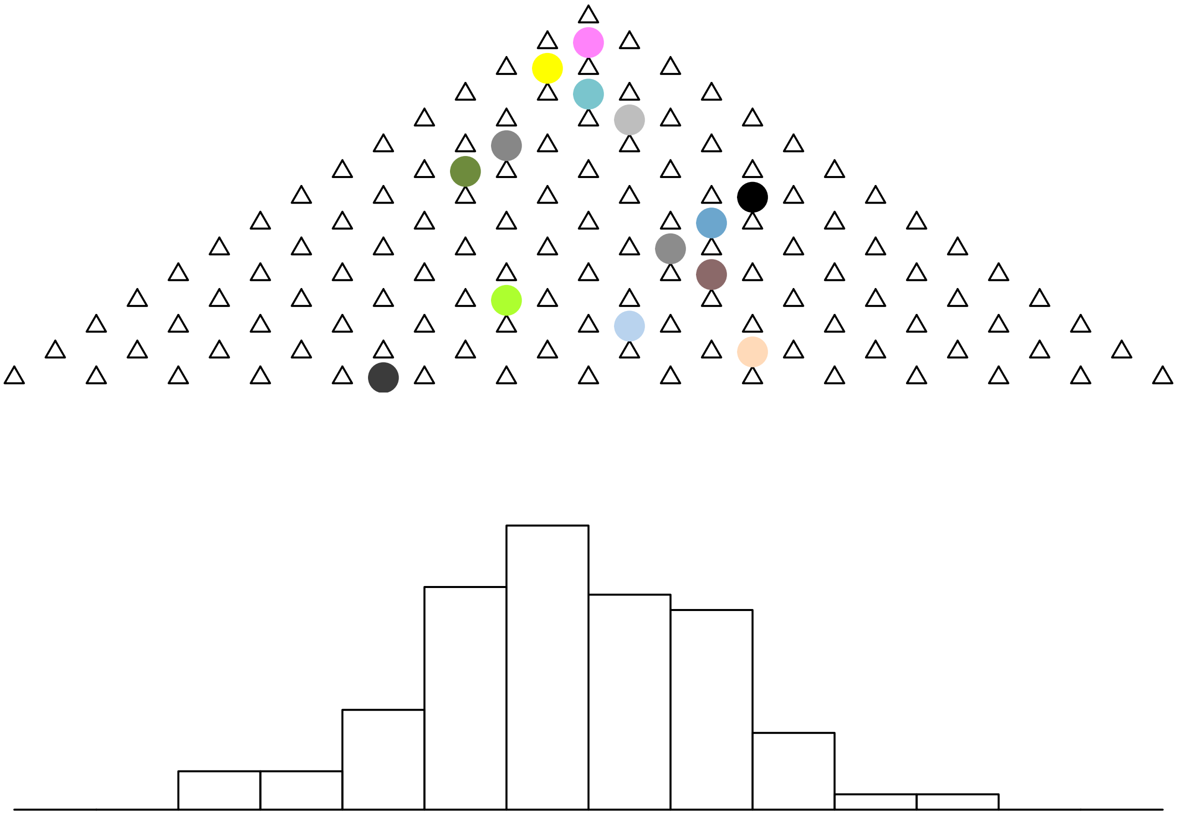


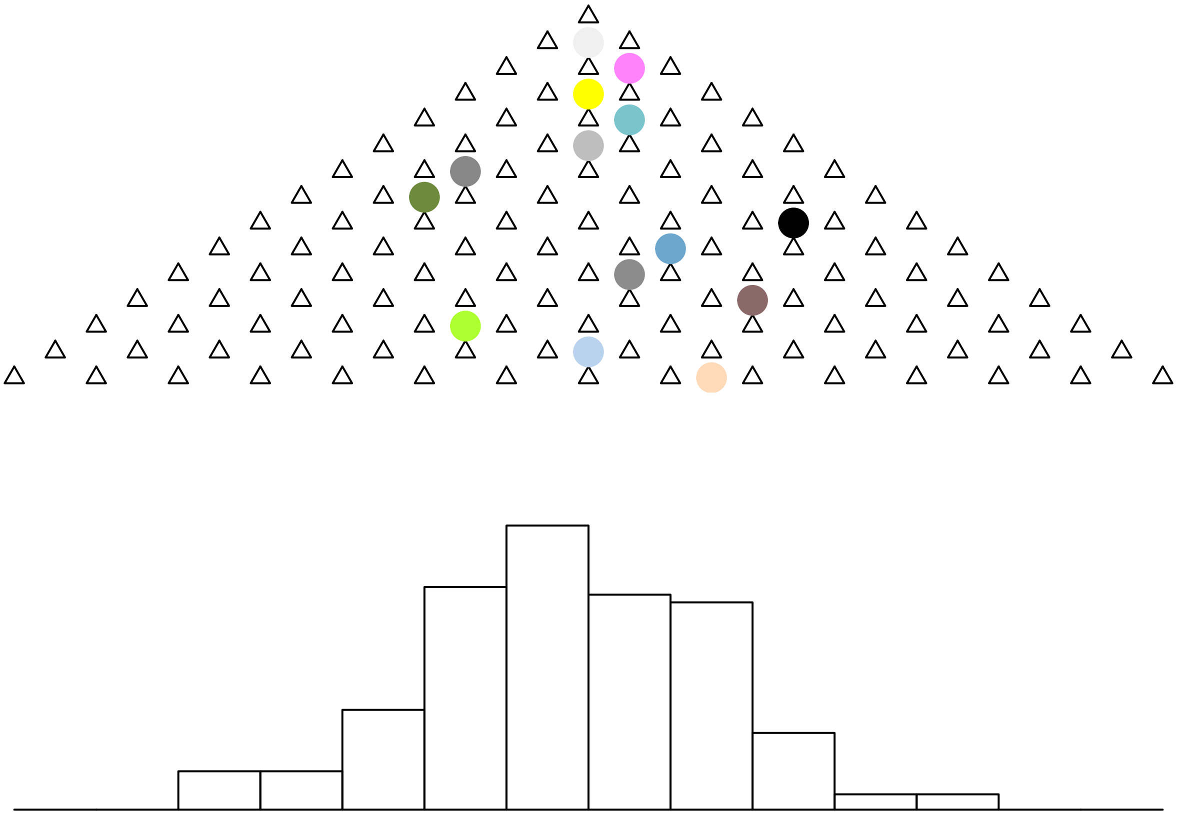


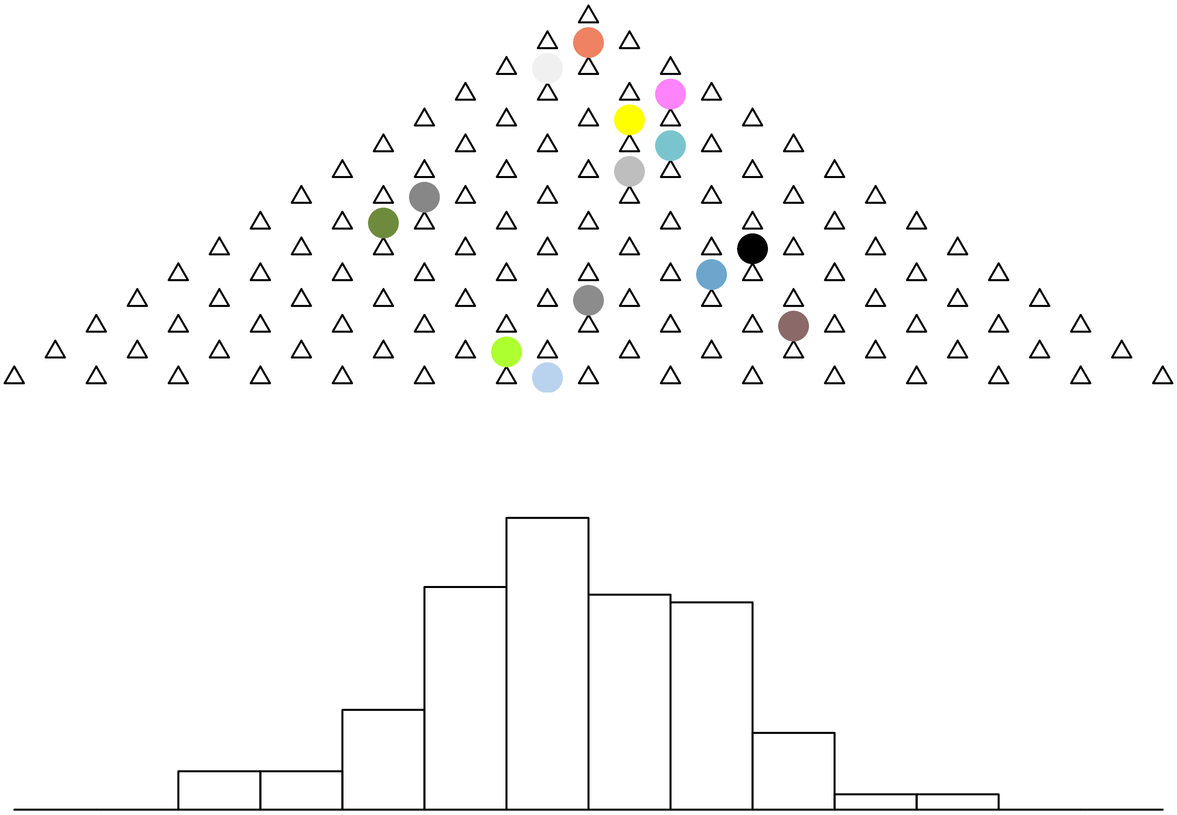


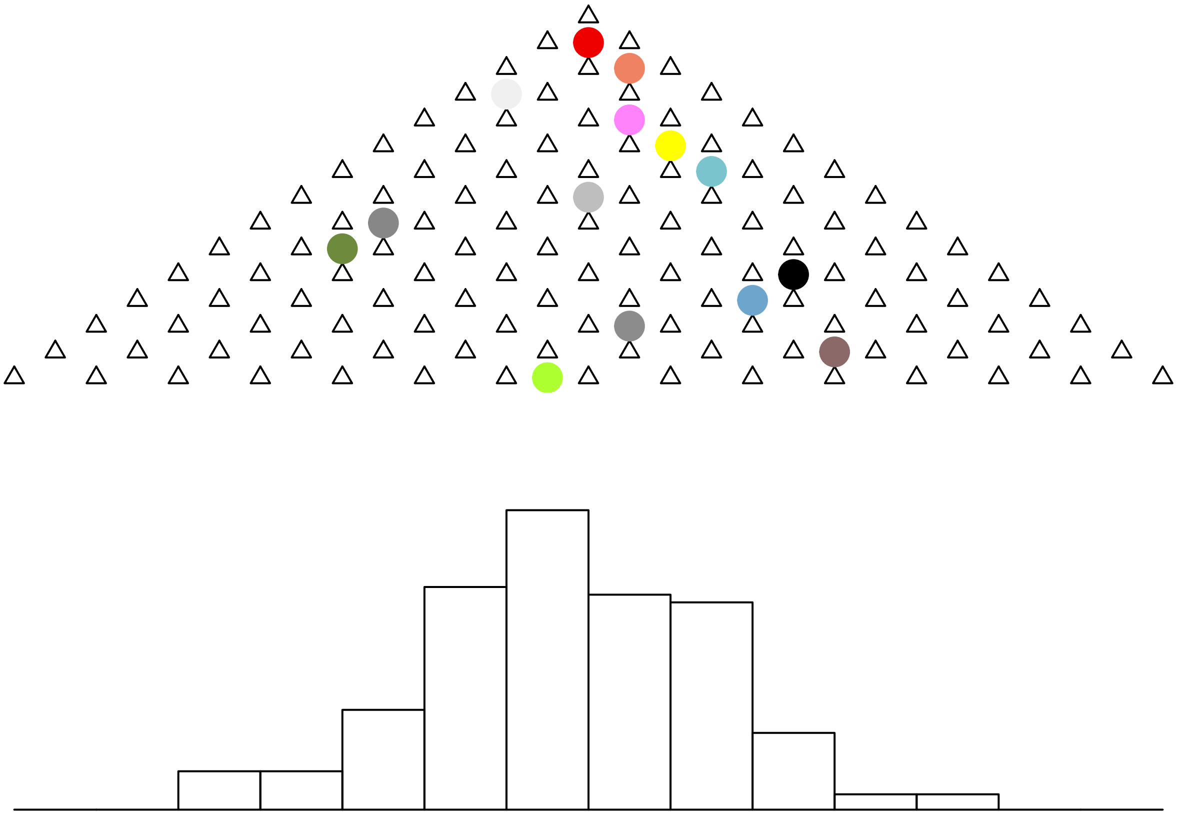


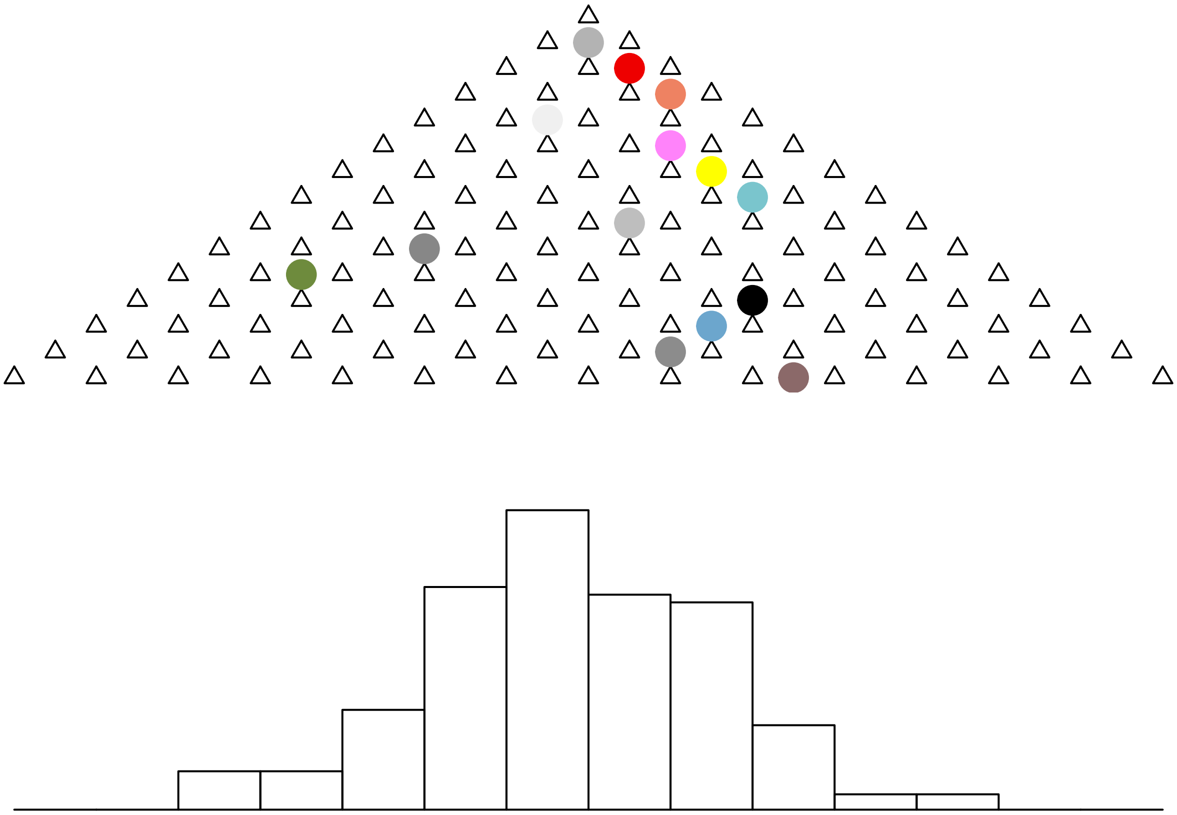


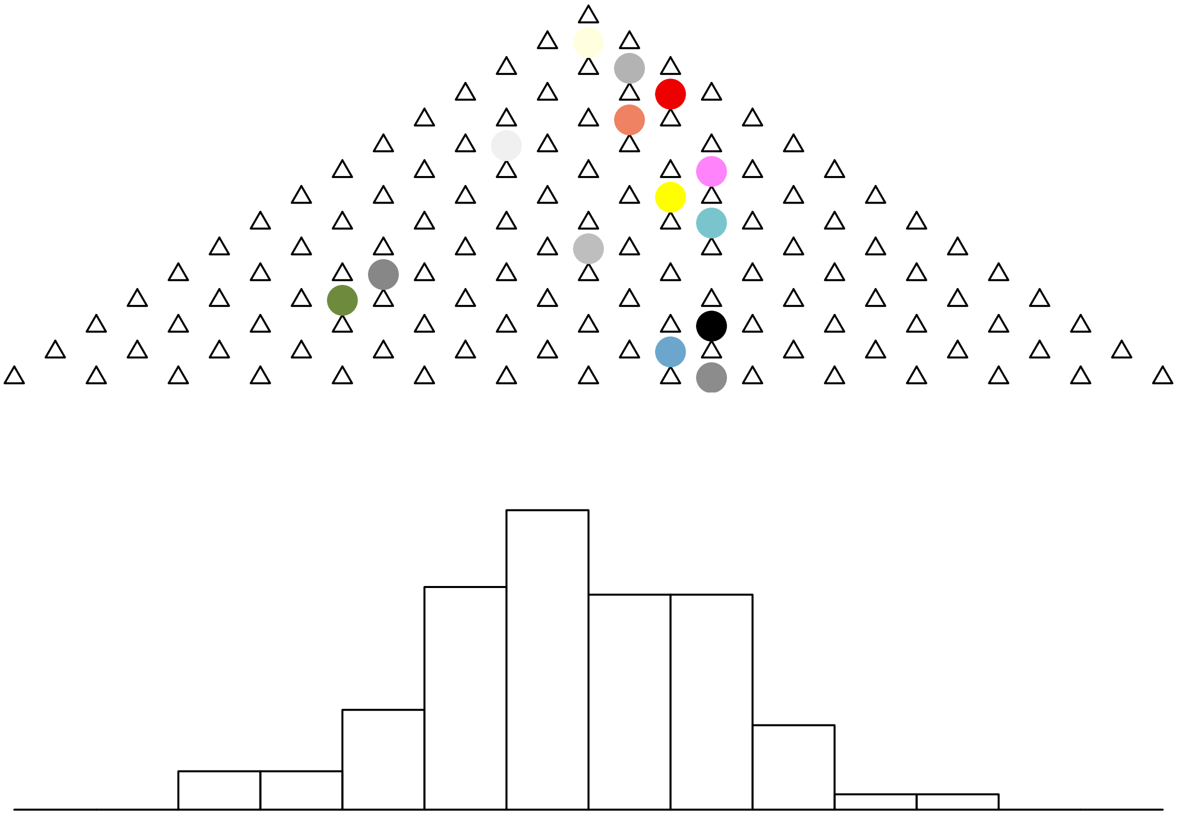


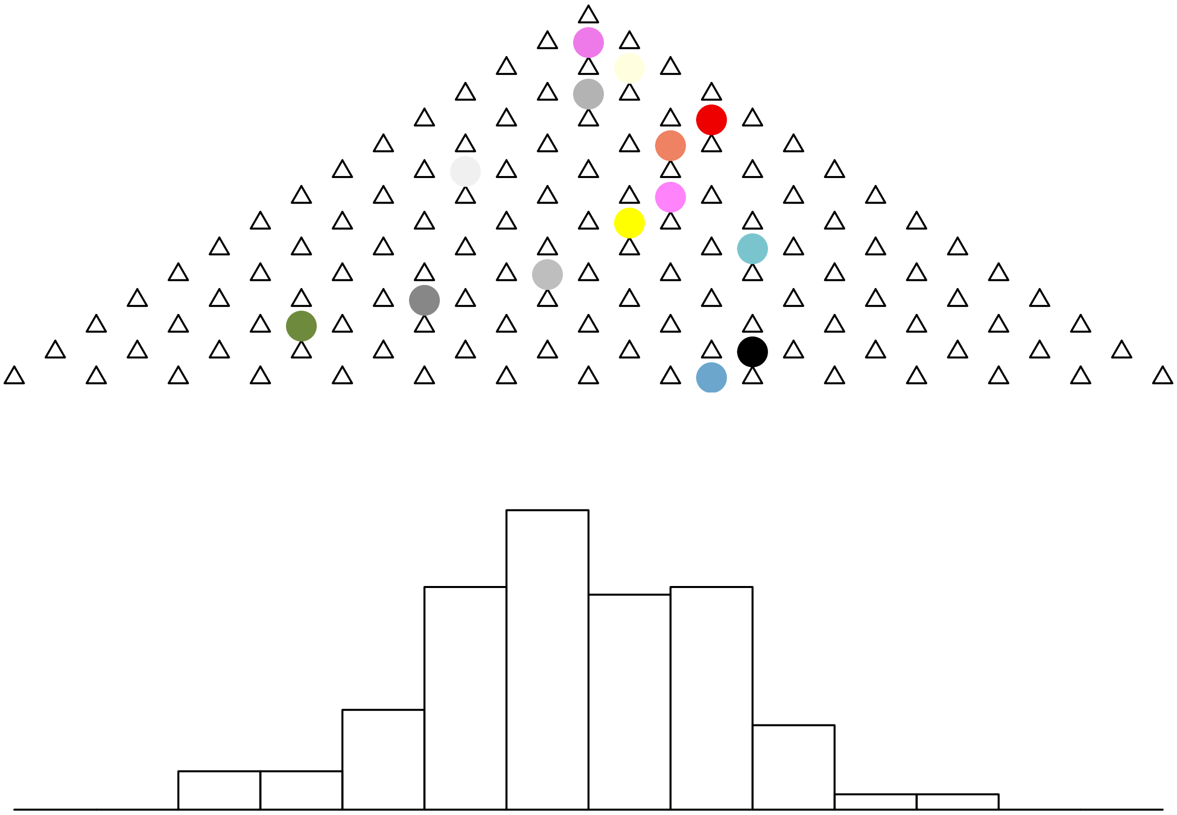


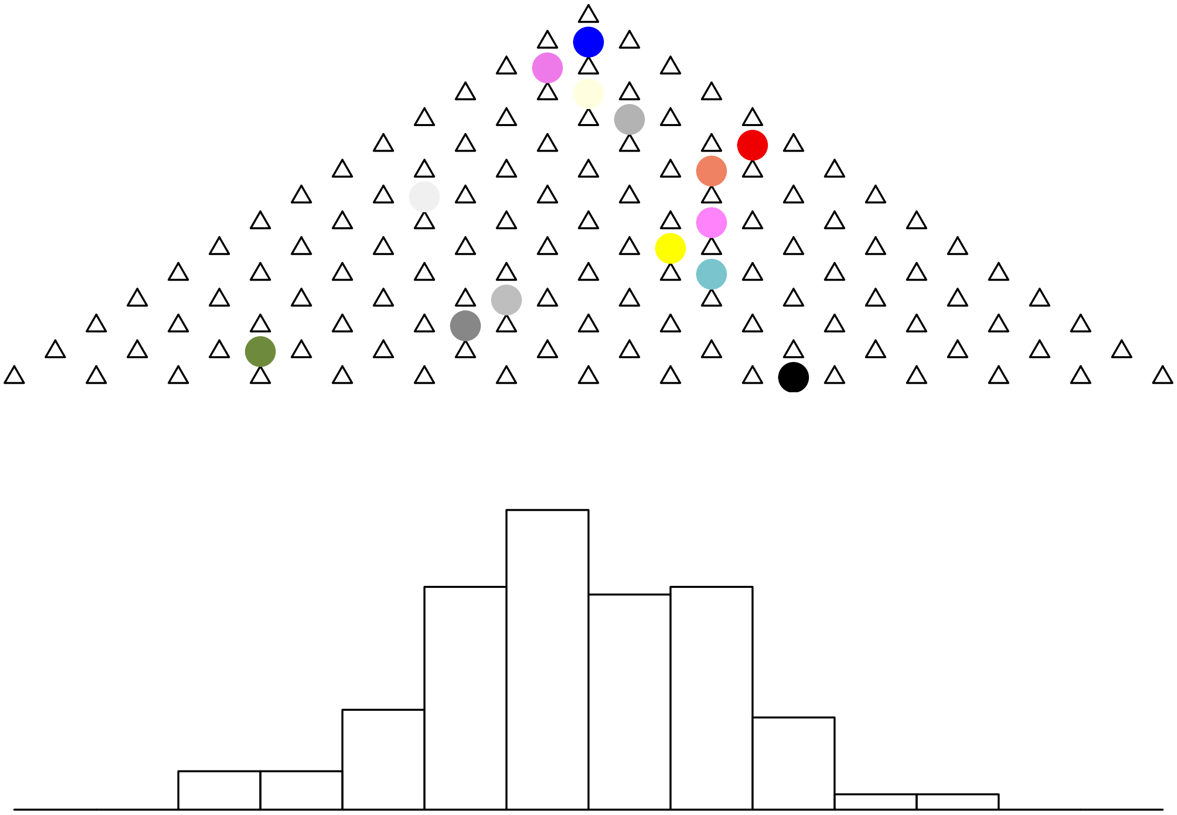


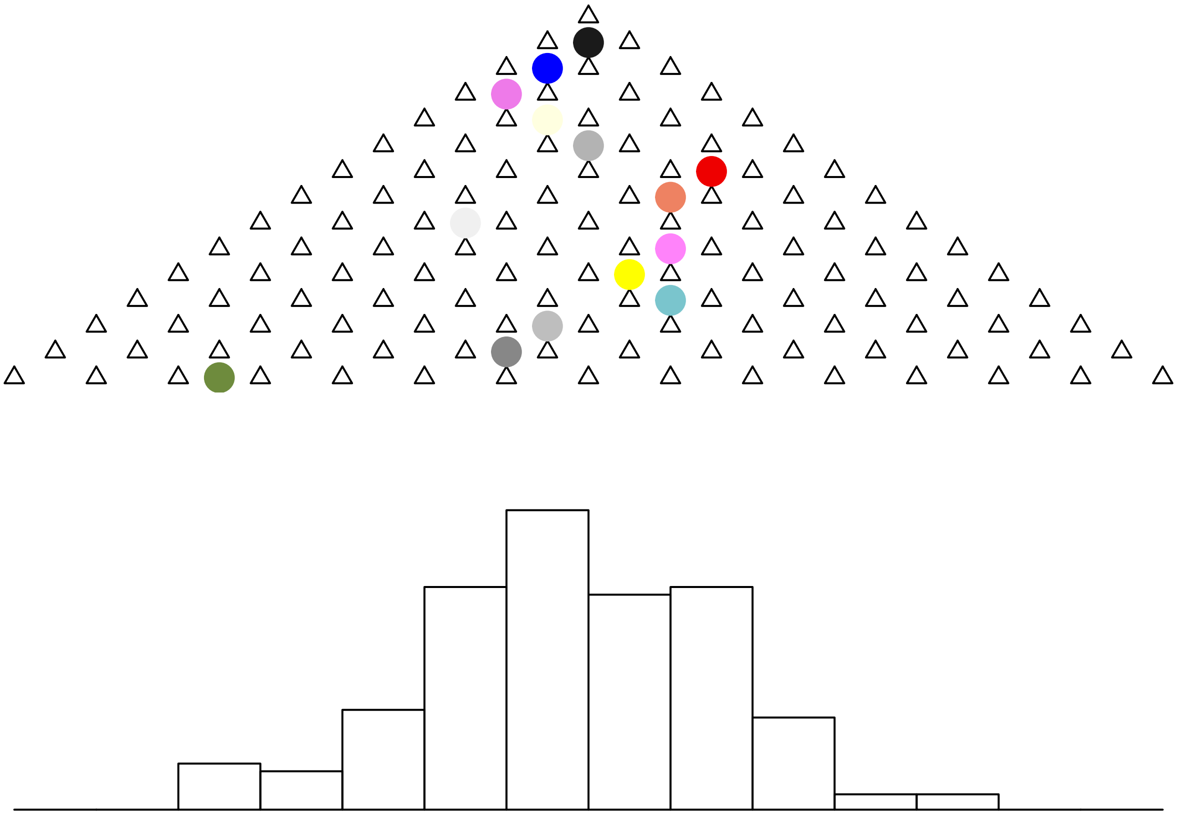


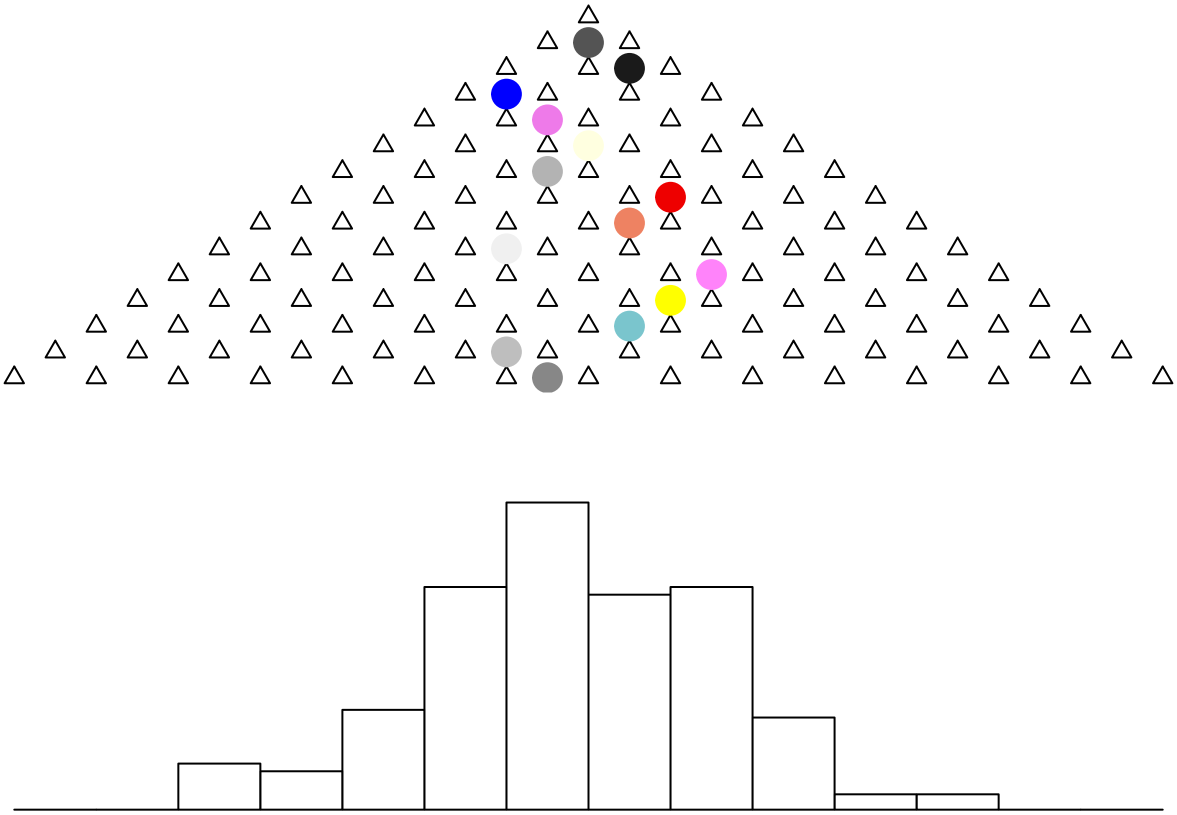


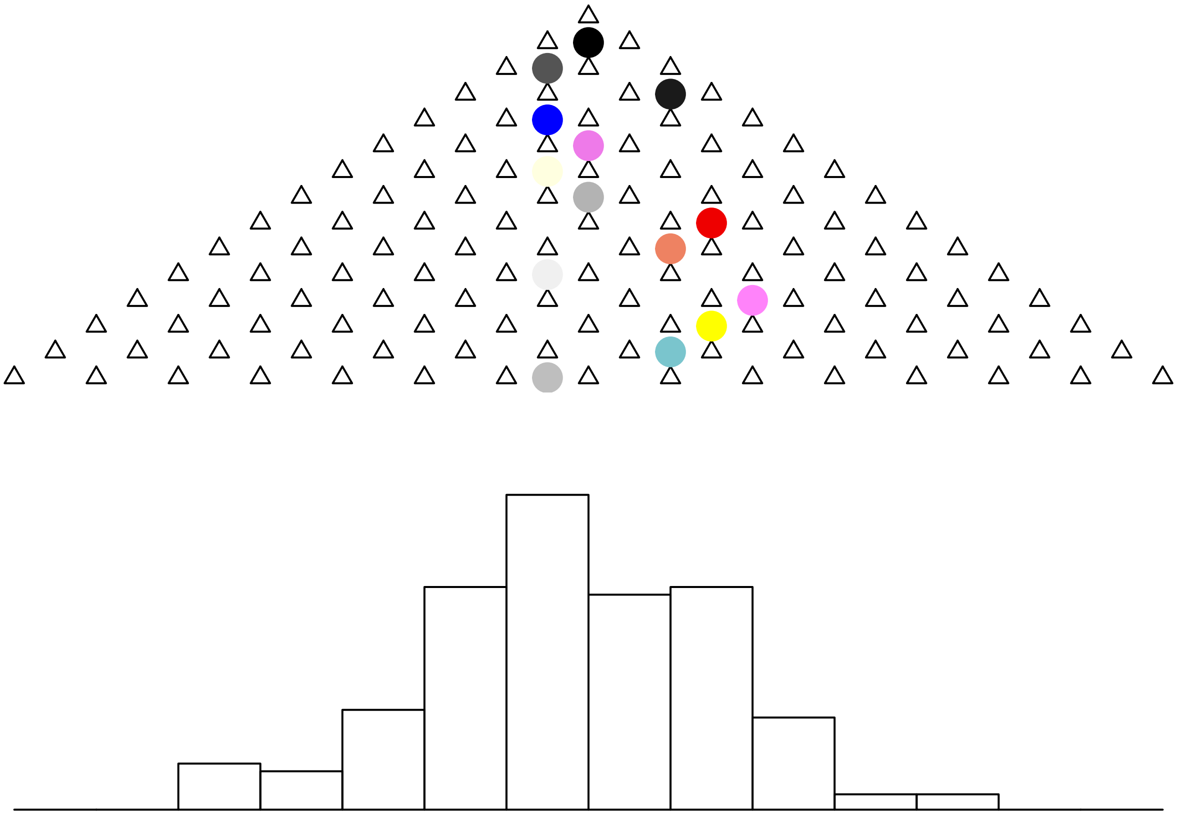


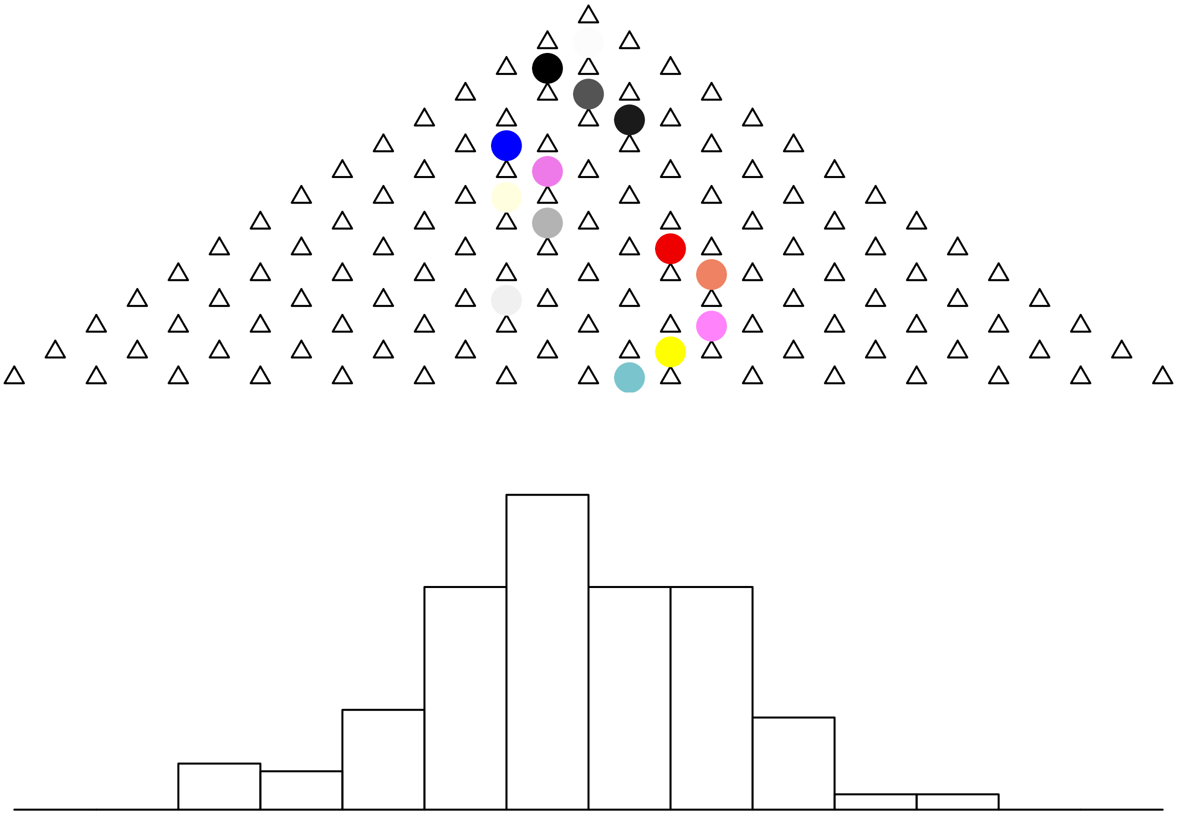


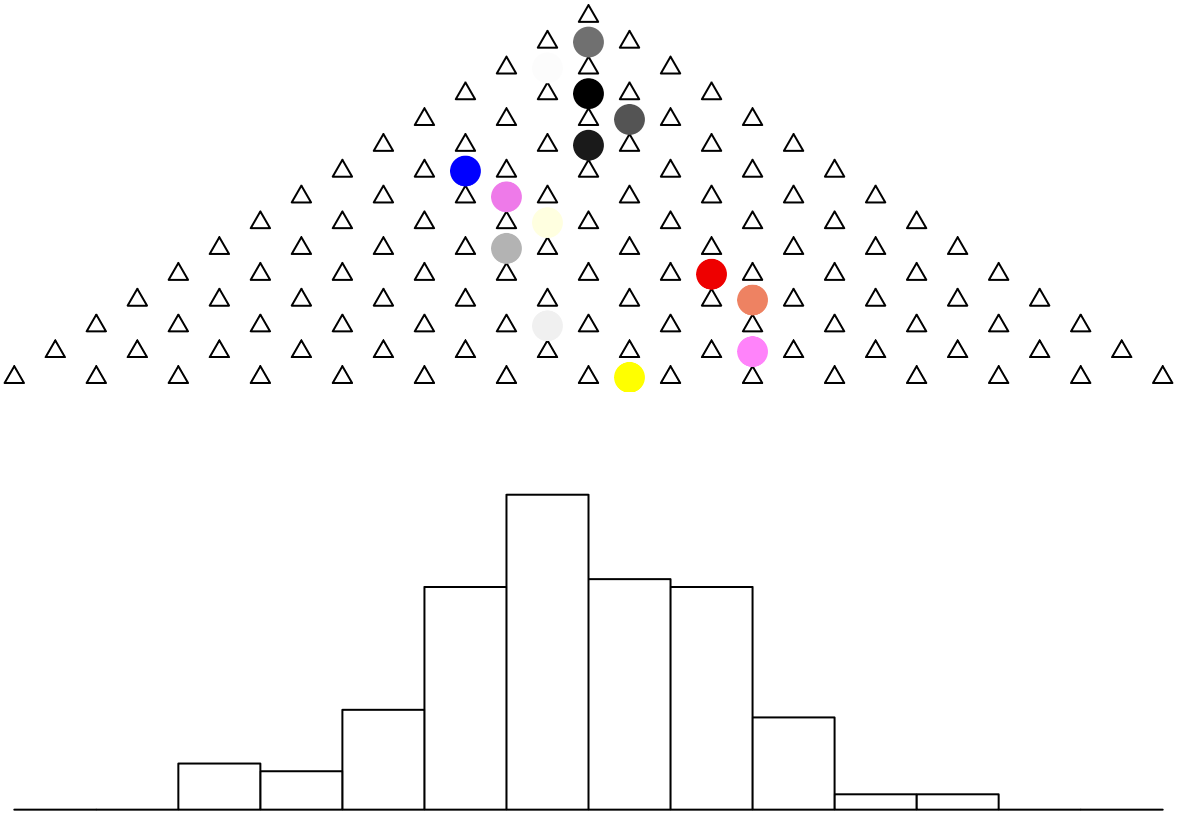


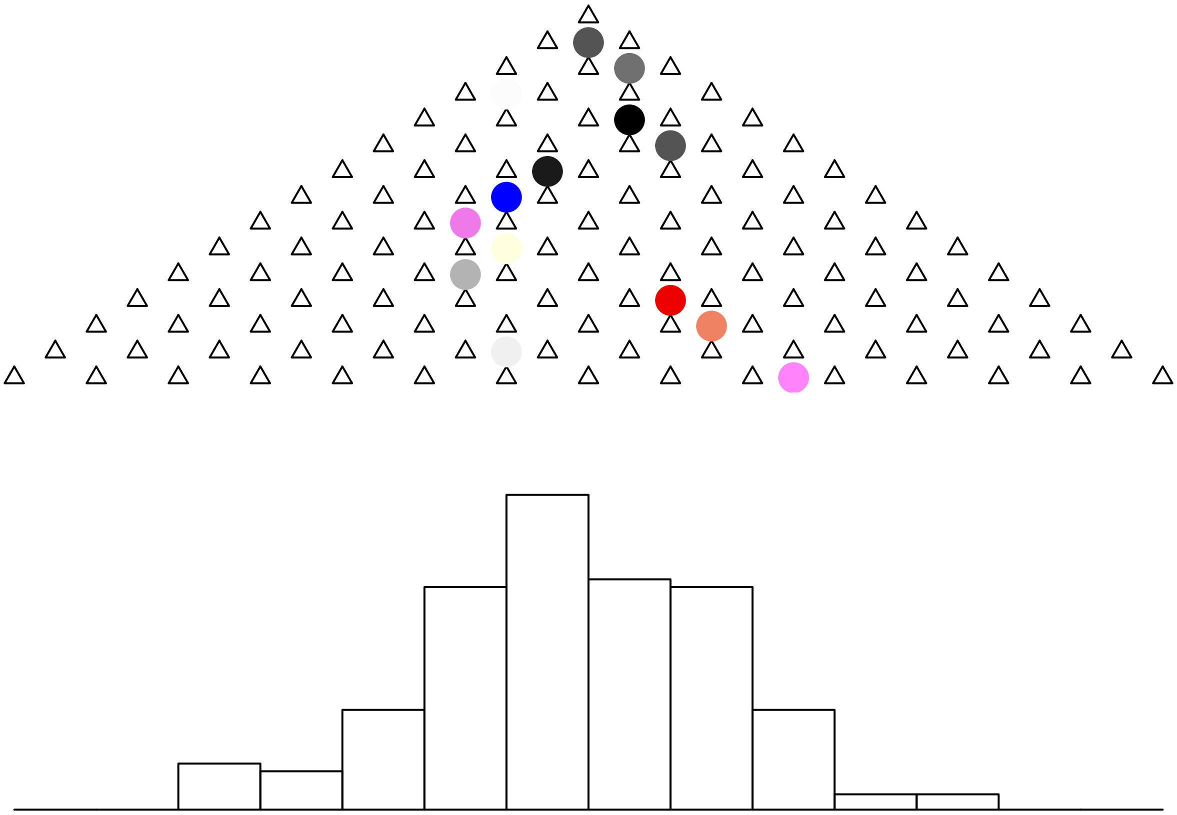


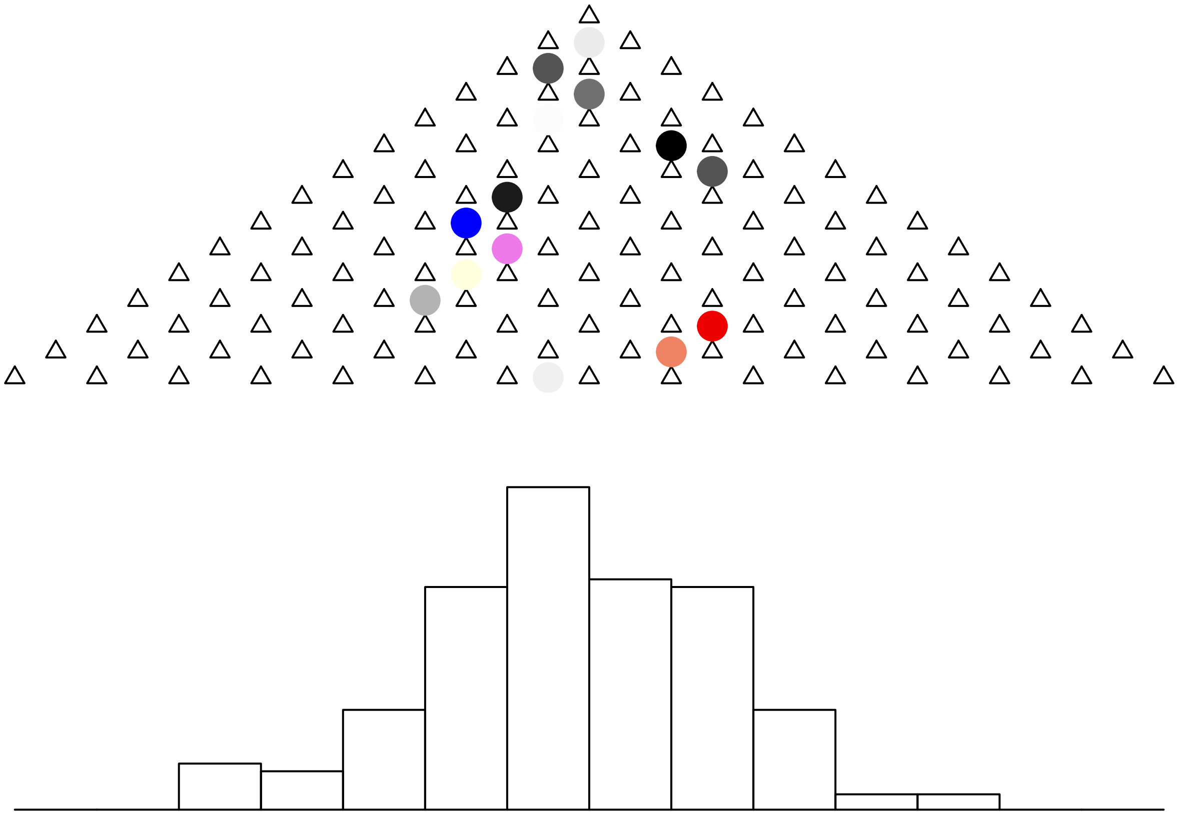


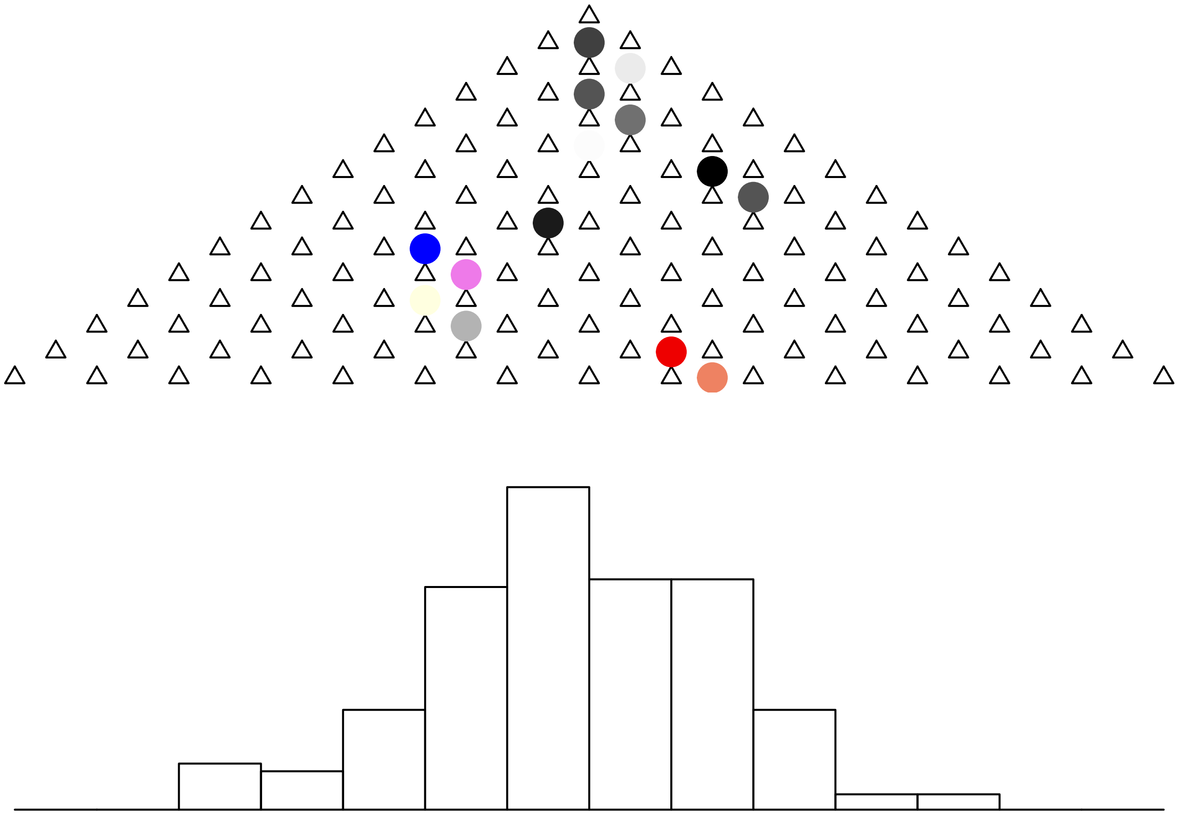


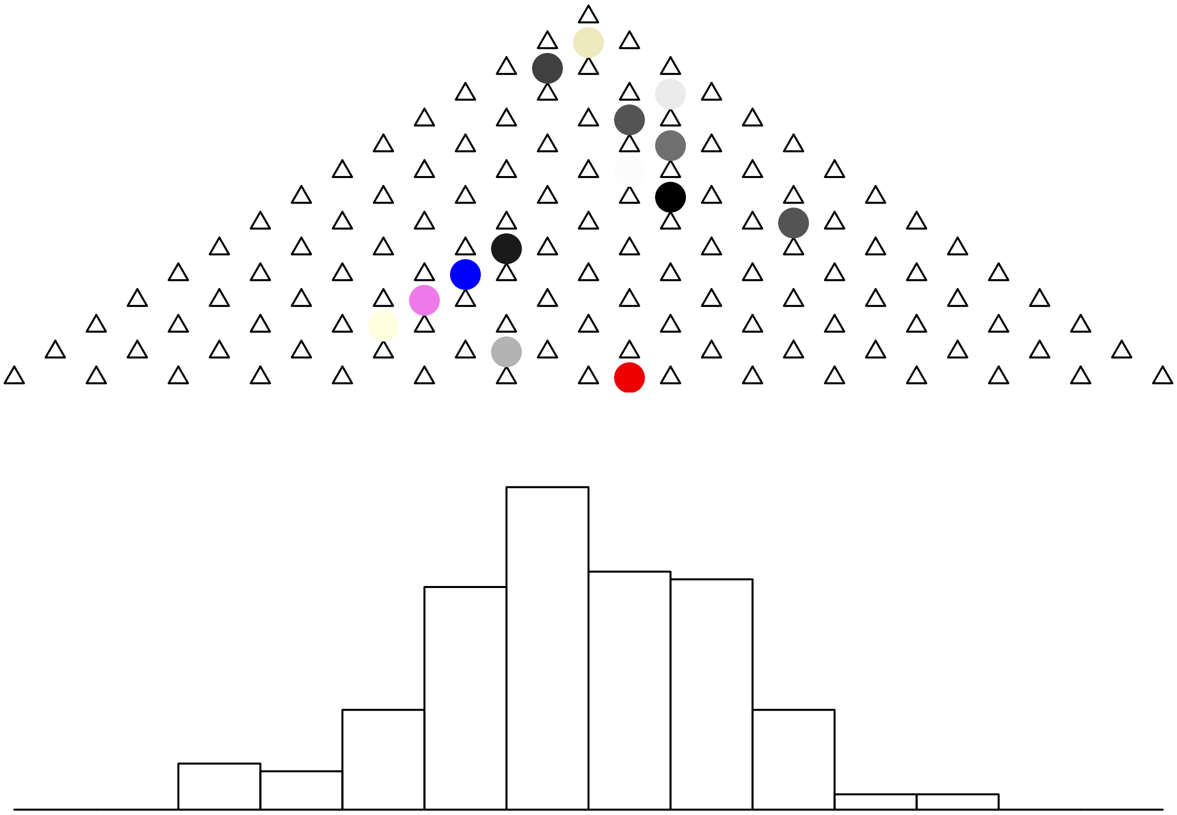


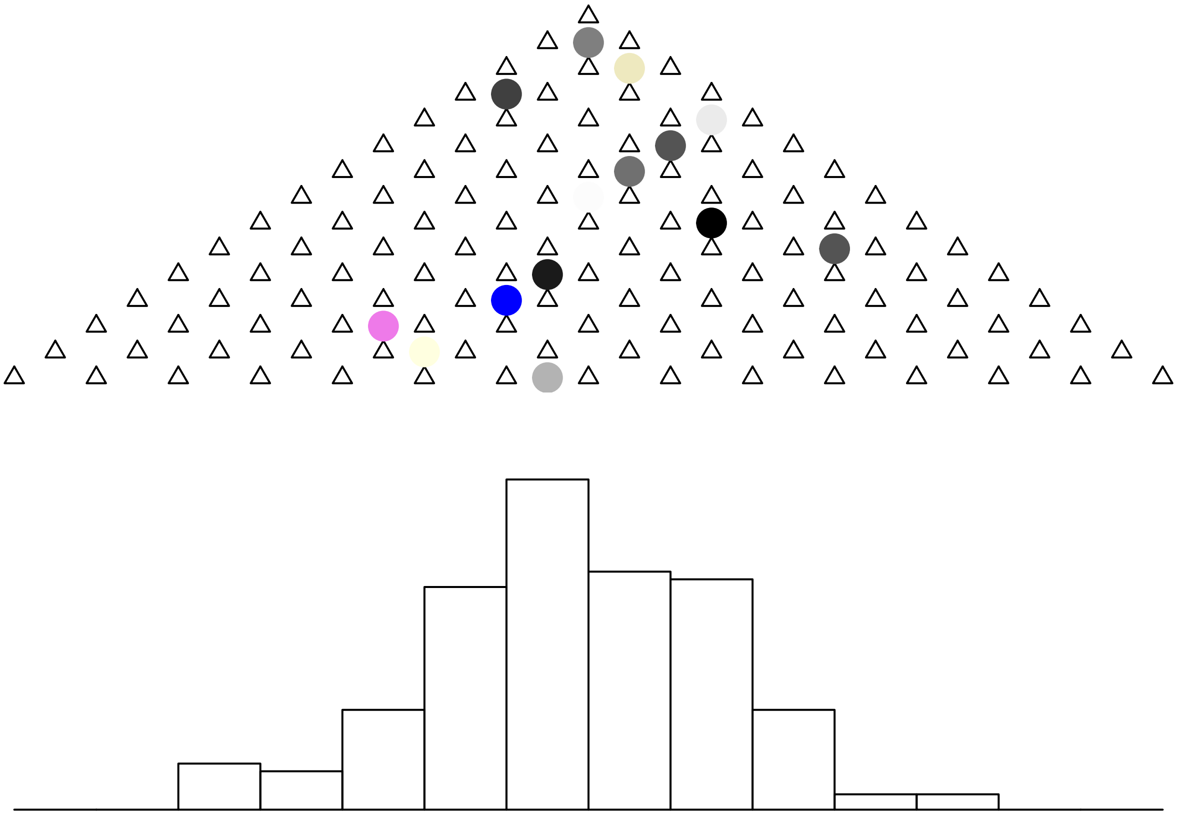


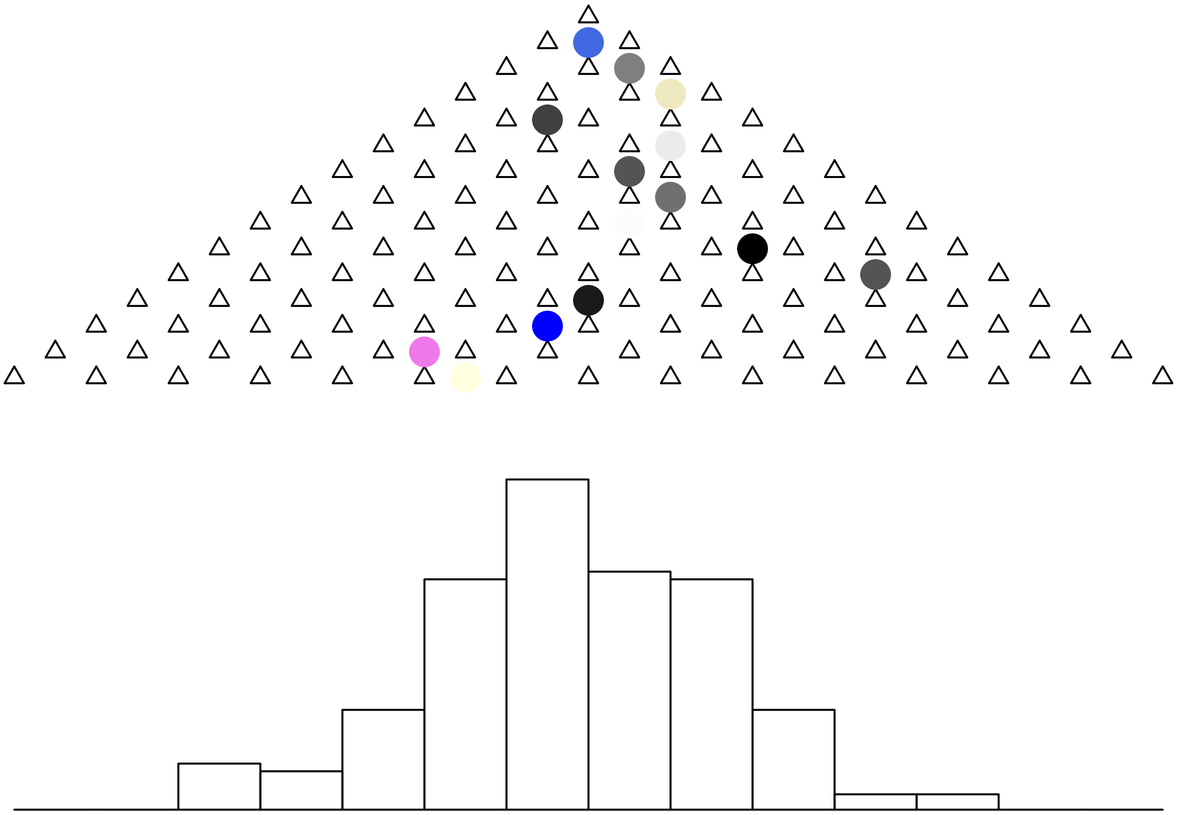


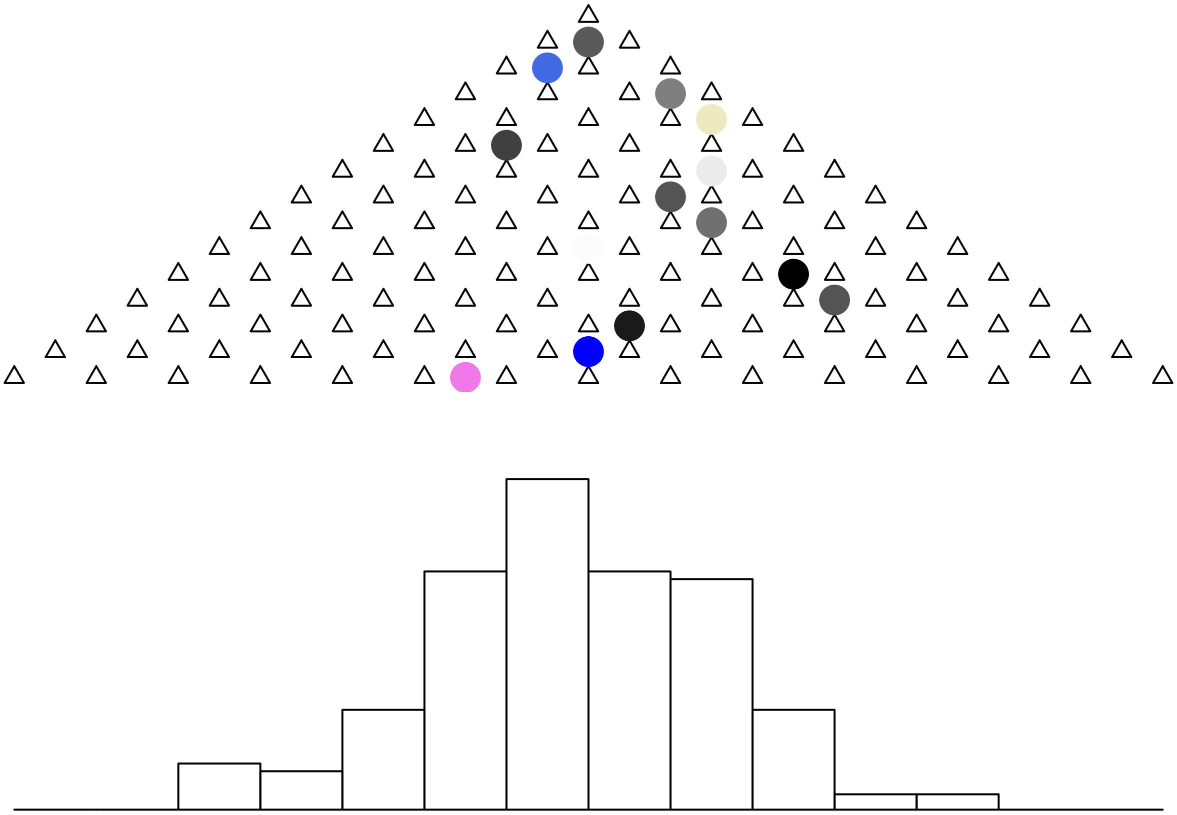


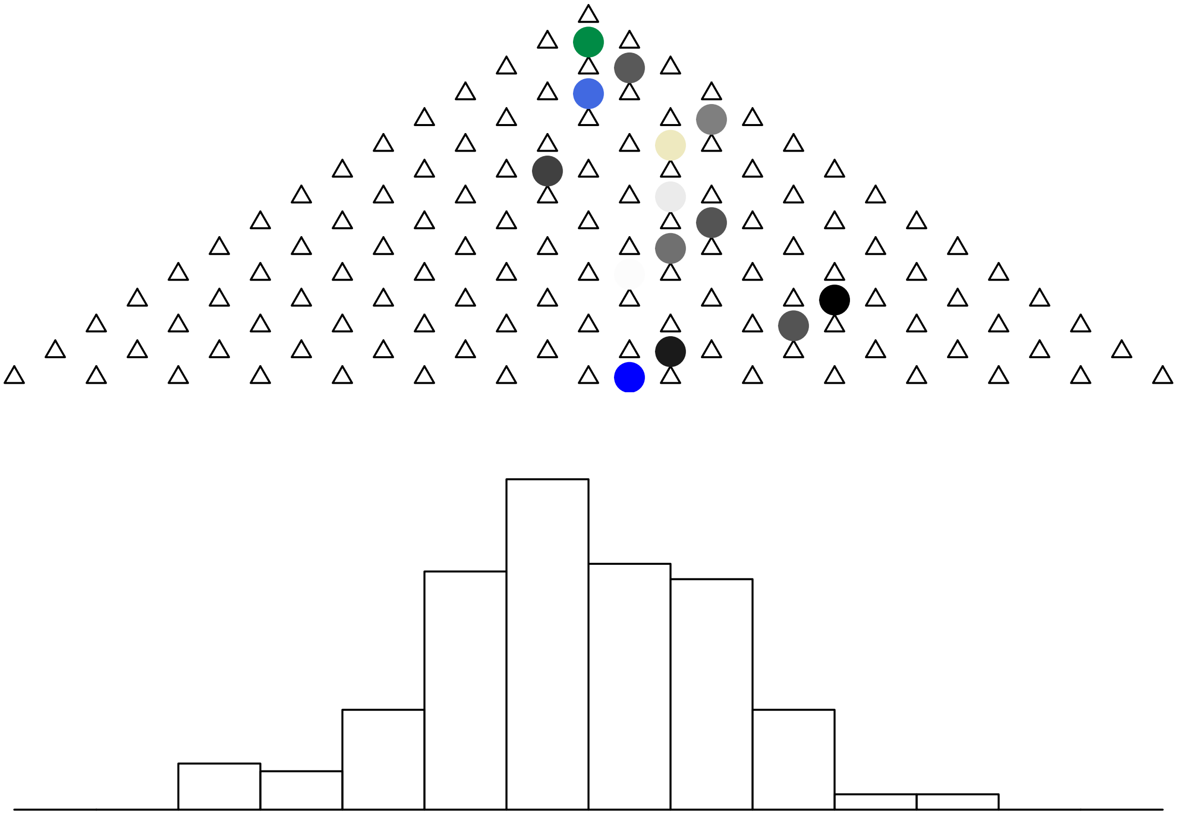


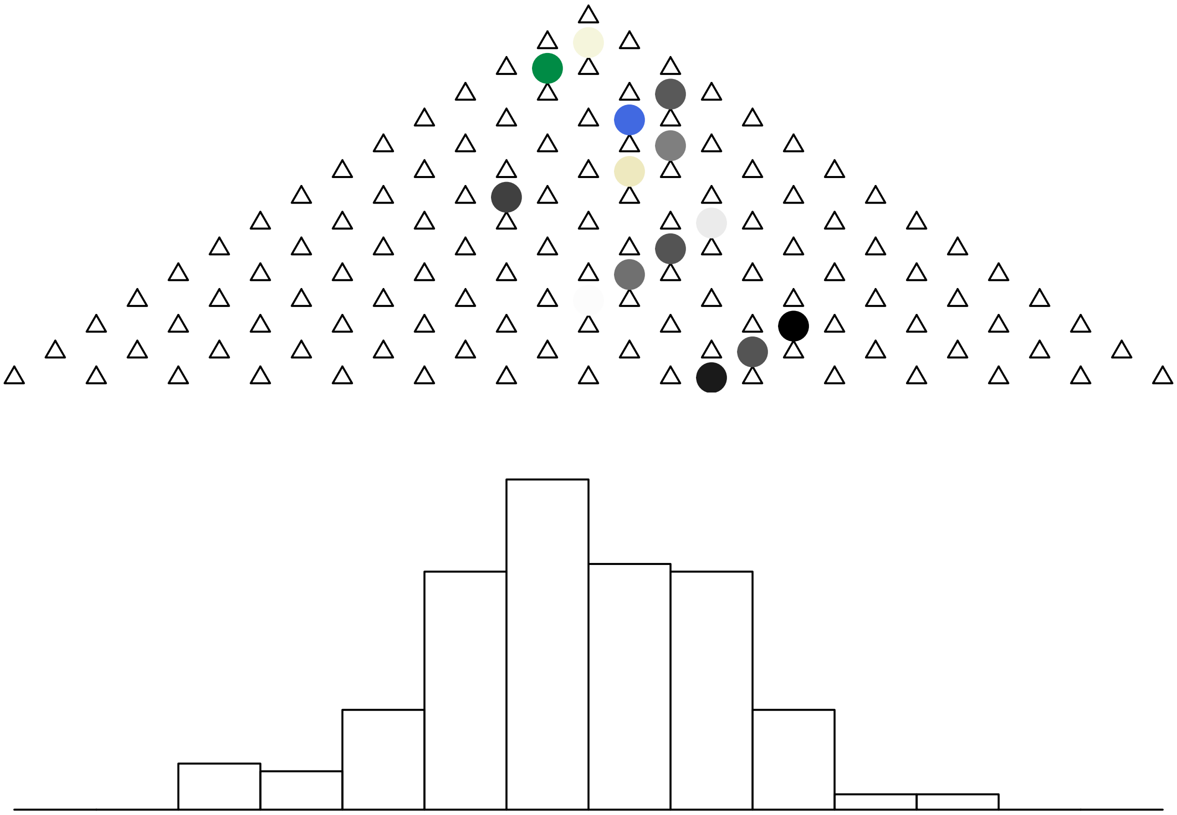


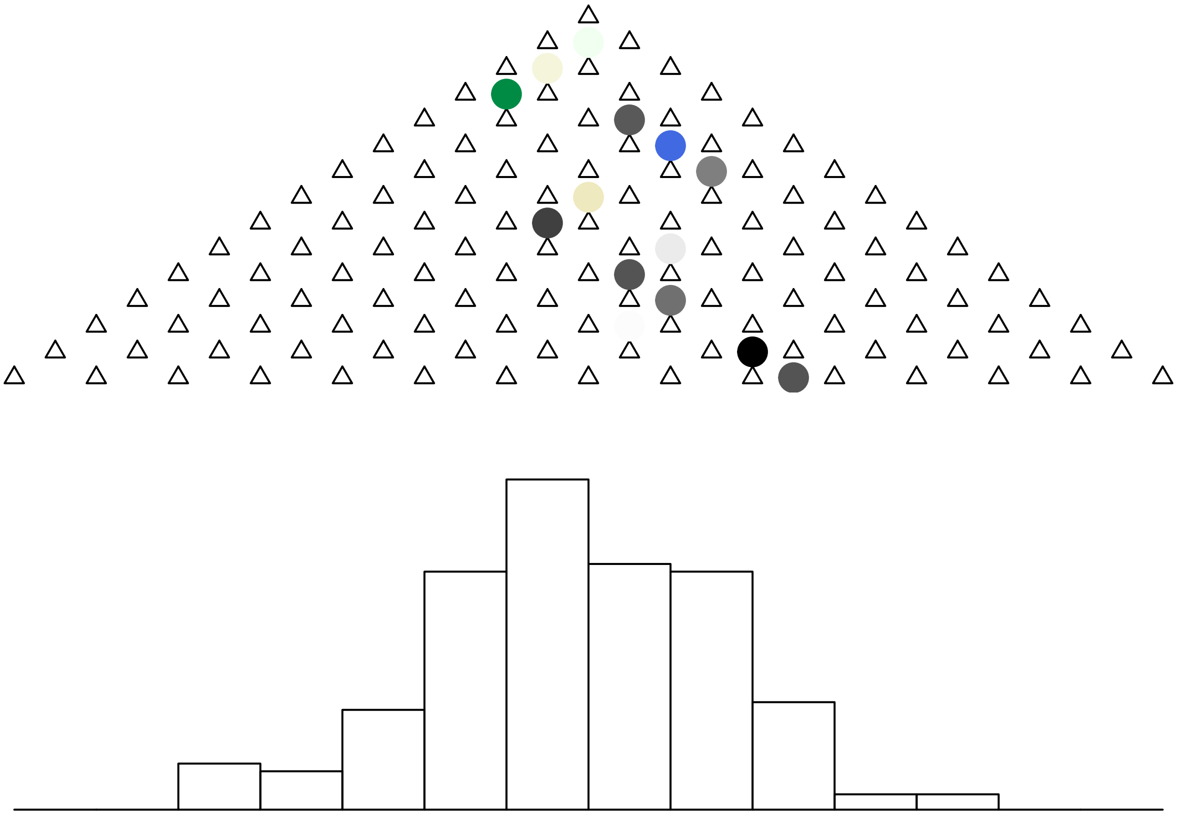


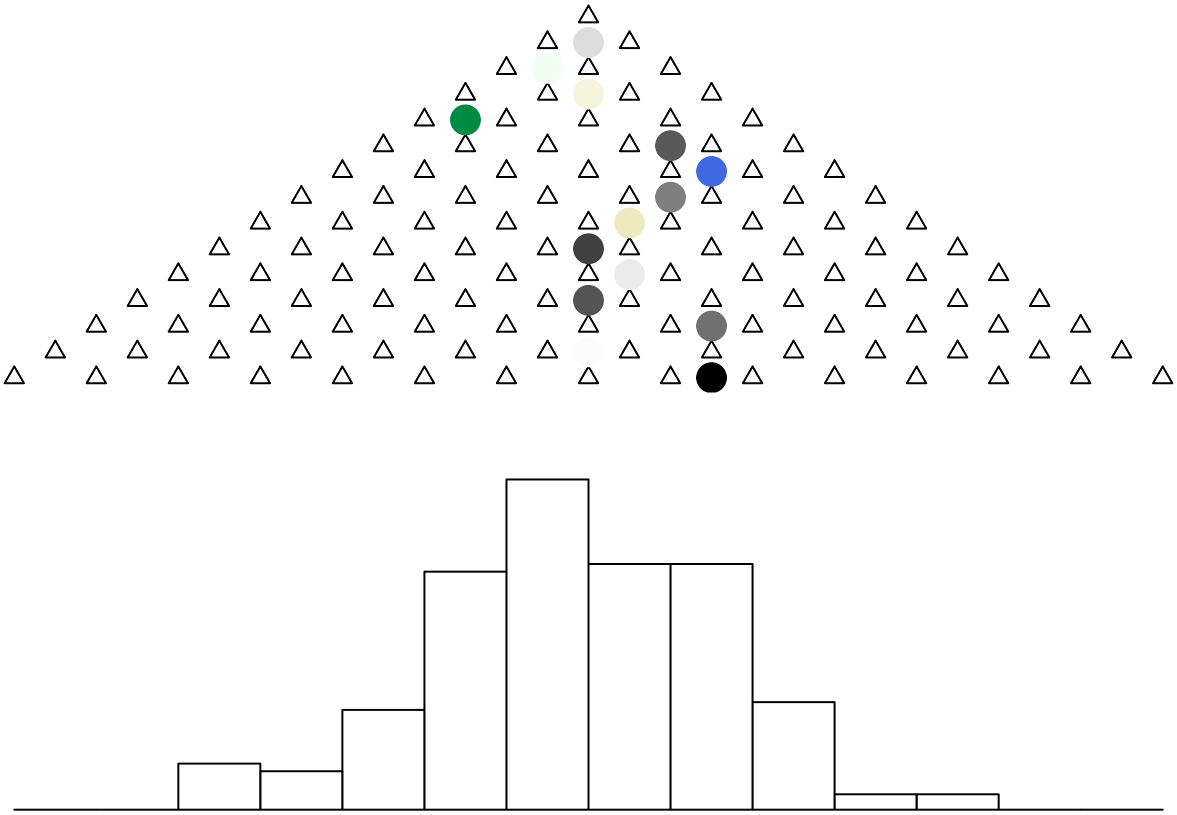


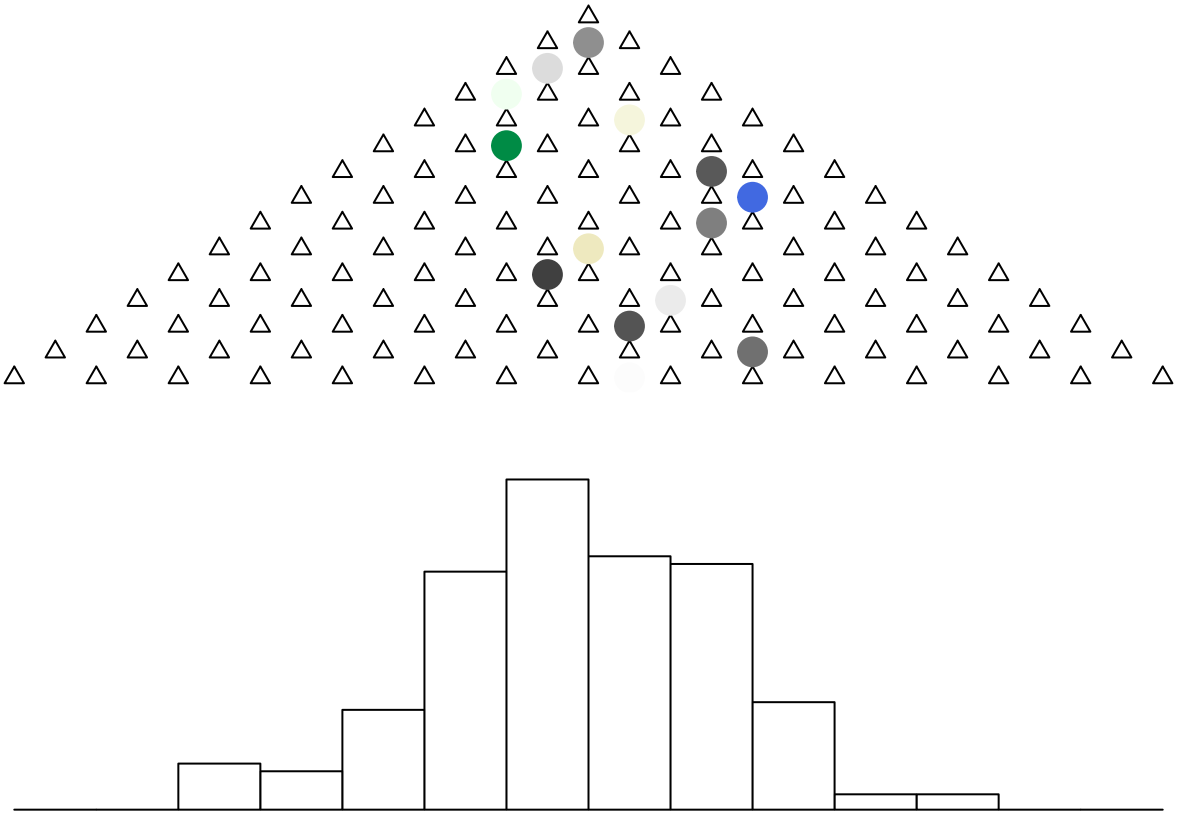


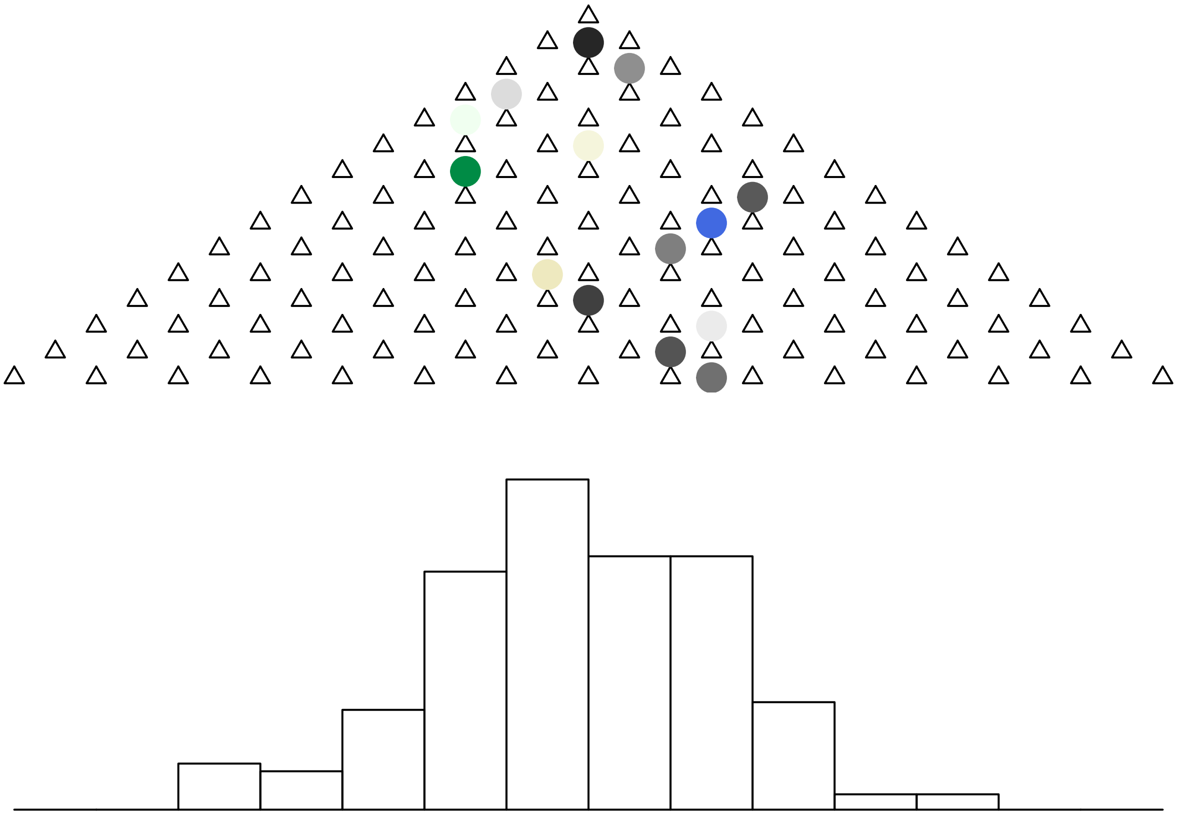


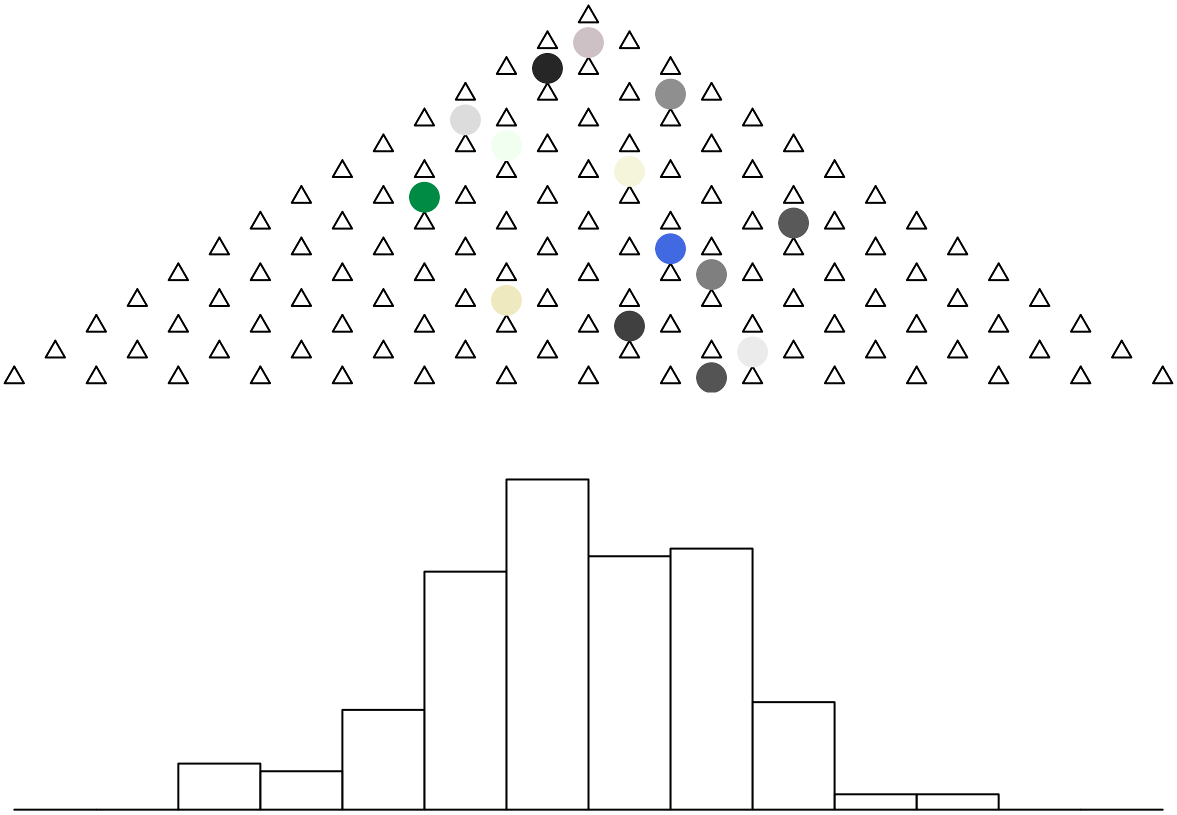


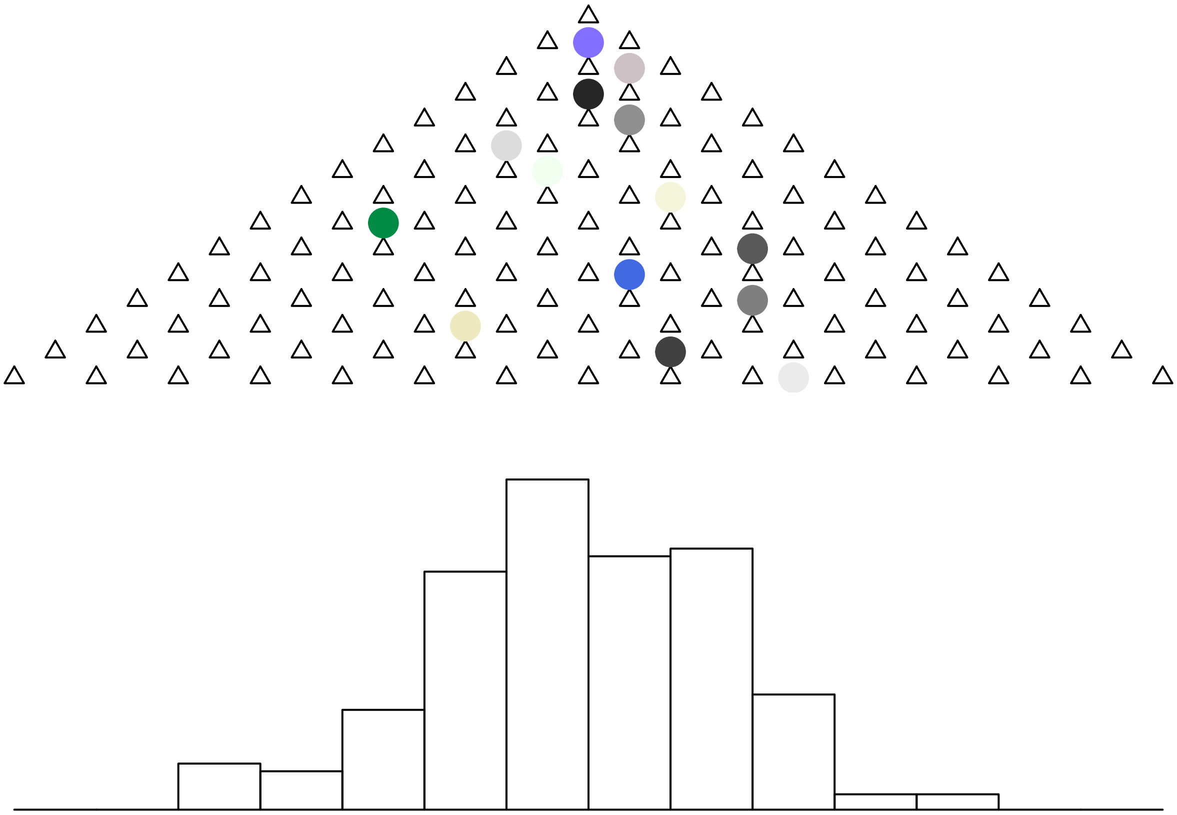


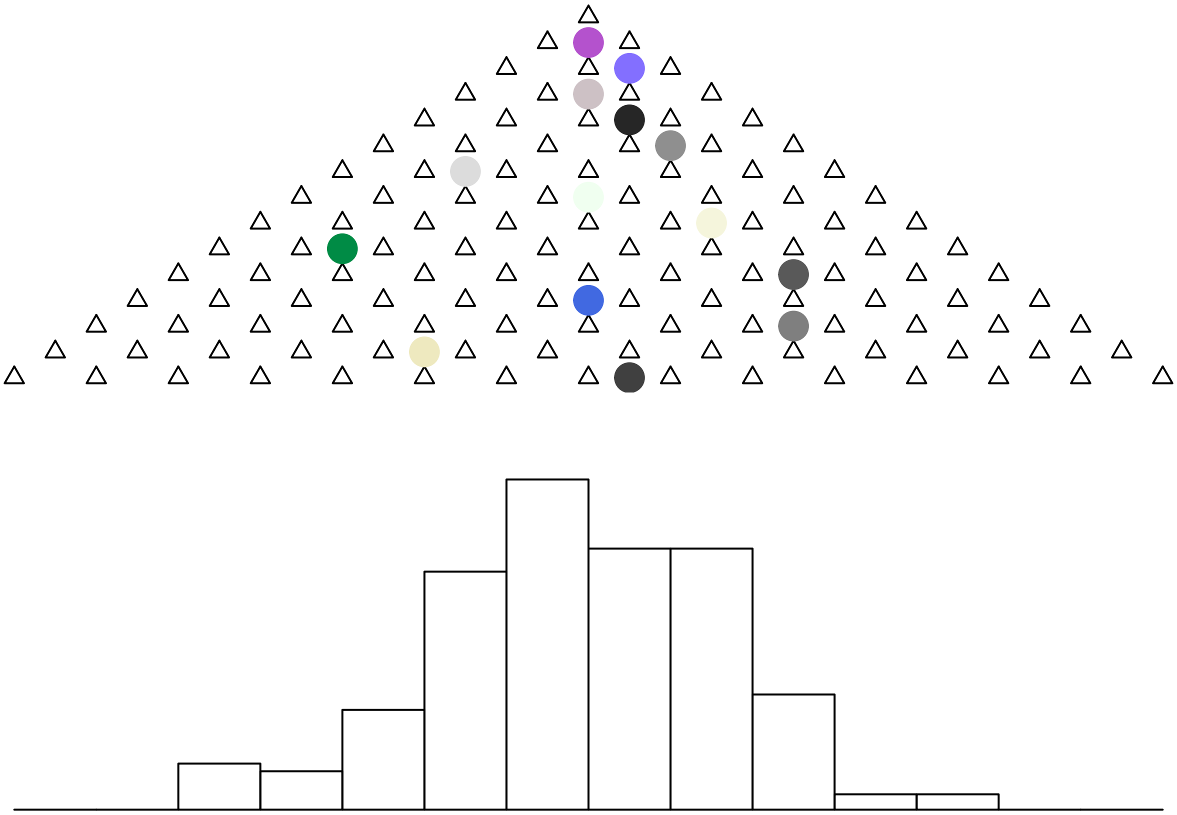


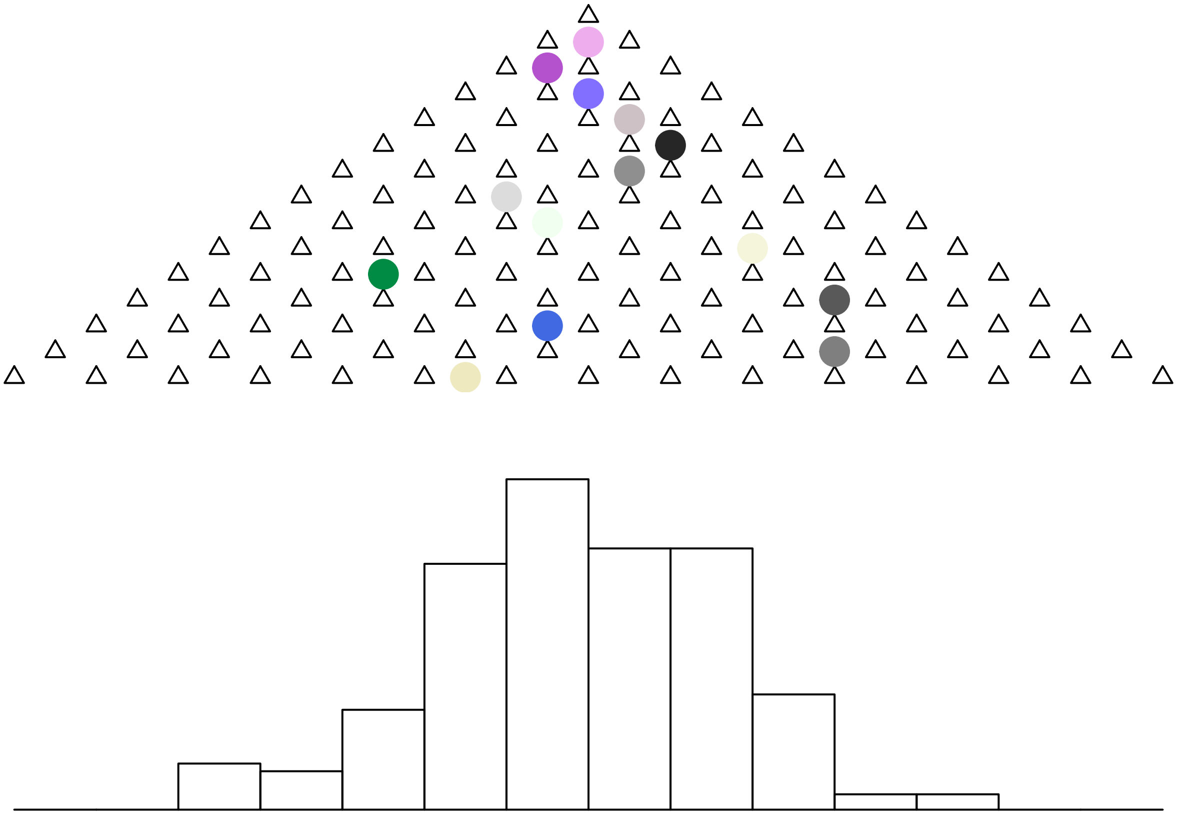


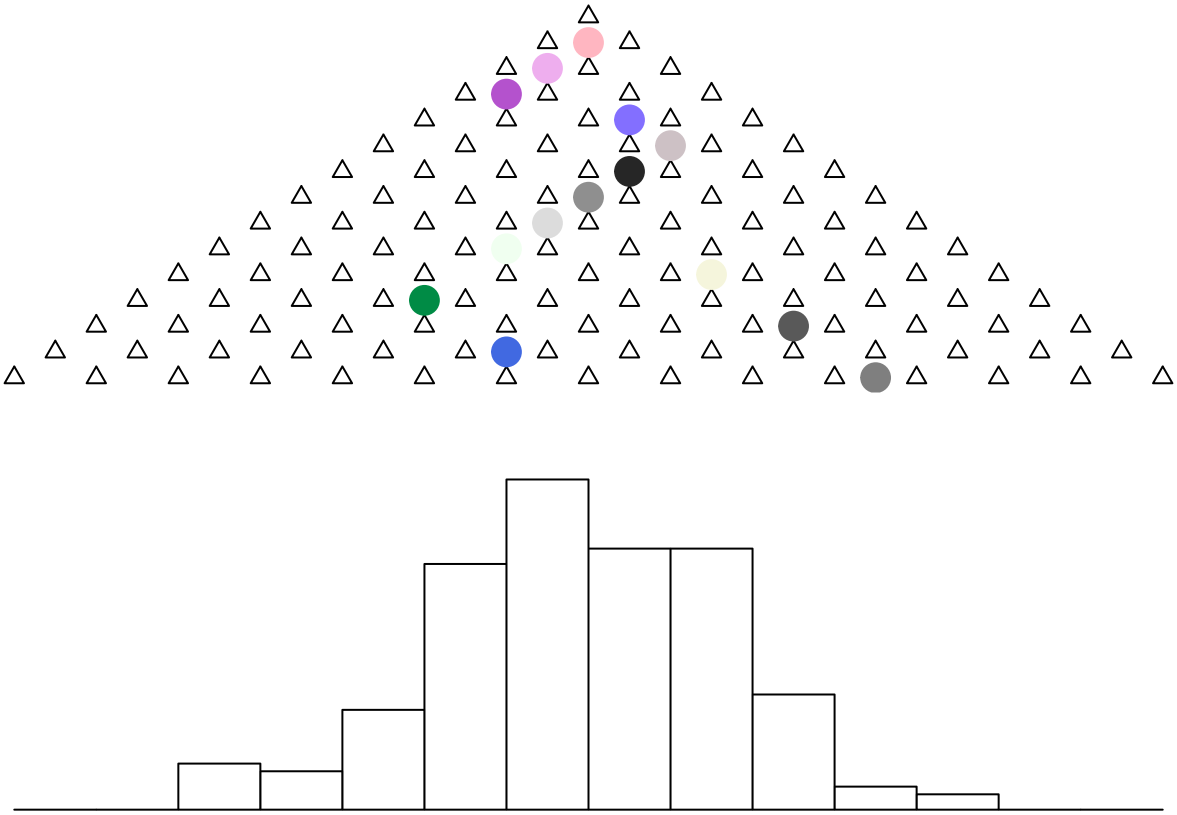












This example is referred to as the Galton Board or Bean Machine, and demonstrates how the randomly descending individuals aggregate in a way that approaches a normal distribution.

Question 4: Can you think of other ways in which random movement/placement could produce spatial aggregation? Why might it be important to consider that aggregation can occur in scenarios where each individual movement cannot be reliably predicted, but where the general outcome can?

Answer 4:

Question 5: It may be of no surprise that spatial aggregation is common. But, under what scenarios would you expect a uniform distribution, e.g., nearly equal distances between all individuals?

Answer 5:

Concept 2: Spatially explicit and implicit

In the previous demonstration, many individuals aggregated around the center of the x-axis. Consequently, our knowledge of where the aggregation occurred was **explicit**. But, because we did not track the location of each individual, our knowledge of where any individual ended up, was **implicit**. That is, the pattern of aggregation implies that an individual drawn at random will have likely ended up somewhere near the center.

In short, if the location of some unit (e.g. individuals, patches) is known and is examined or modeled, then our analysis or model is spatially explicit with regards to that unit. While our Galton Board, was spatially implicit with regards to individuals, we were able to track which locations were becoming aggregated and so, was spatially explicit in regards to "patches".

In accounting for space and location in your own research, you'll need to know whether you are dealing with a spatially implicit or explicit problem/hypothesis and whether you need a spatially explicit or implicit analysis/model.

Question 6: In considering our microbial data from the refuge ponds, what is our data (and probably our questions) spatially explicit and implicit in regards to?

> ***Answer 6***:

Concept 3: Spatial autocorrelation

Aggregation is one of the most common features of any natural system. Another is spatial autocorrelation, that is, the degree to which a set of spatial features and their associated data values tend to be clustered together in space (positive spatial autocorrelation) or dispersed (negative spatial autocorrelation). Tobler's first law of geography is a formulation of the concept of spatial autocorrelation made by the geographer Waldo Tobler (1930-). This law states that "Everything is related to everything else, but near things are more related than distant things."

When analyzing spatial data, it is important to check for autocorrelation. If there is evidence of spatial autocorrelation, then one of the underlying assumptions of your analysis may be violated and your results may be invalidated.

Addressing spatial autocorrelation leads to more robust and replicable results.
Analysis of spatial autocorrelation can be broken down into steps: detecting, describing, and adjusting/predicting.

i. Detecting spatial autocorrelation *Moran's I* and the *Mantel test* are commonly-used indicators of spatial autocorrelation. Both will indicate if your spatial autocorrelation is positive or negative and provide a p-value for the level of autocorrelation. Both *Moran's I* and the *Mantel test* test against the null that there is no spatial autocorrelation. Moran's I does this with a correlation that is weighted by inverse distances; the Mantel test examines the correlation between two distance matrices and generating a null distribution for this correlation by randomly permuting one of the matrices.

Let's calculate **Moran's I** for one of our environmental variables

```r
#install.packages("ape")
library(ape)

# first, generate a distance matrix,
ponds.dists <- as.matrix( dist( cbind(lons, lats)))

# then take inverse of the matrix values
ponds.dists.inv <- 1/ponds.dists

# then replace the diagonal entries with zeros
diag(ponds.dists.inv) <- 0

# Let's check what we've done
ponds.dists.inv[1:5, 1:5]
```

```
##          1        2         3         4         5
## 1  0.00000 22.00459  32.60652  43.33236  36.58834
## 2 22.00459  0.00000  47.64274  37.05327  37.91190
## 3 32.60652 47.64274   0.00000 128.07792 183.46400
## 4 43.33236 37.05327 128.07792   0.00000 212.41760
## 5 36.58834 37.91190 183.46400 212.41760   0.00000
```

```
# We can now calculate Moran's I using the command Moran.I.
ponds.dists.inv[is.infinite(ponds.dists.inv)] <- 0
Moran.I(Ponds$Temp, ponds.dists.inv)
```

```
## $observed
## [1] 0.1440198
##
## $expected
## [1] -0.01960784
##
## $sd
## [1] 0.0362343
##
## $p.value
## [1] 6.307167e-06
```

Next, let's calculate **Mantel's test** for the same environmental variable

```
#install.packages("ade4")
library(ade4)
```

```
##
## Attaching package: 'ade4'
##
## The following object is masked from 'package:vegan':
##
##     cca
```

```
ponds.dists <- dist( cbind(lons, lats))
env.dists <- dist(Ponds$Temp)

# Checking the upper left corner of our results
as.matrix(ponds.dists)[1:5, 1:5]
```

```
##            1          2           3           4           5
## 1 0.00000000 0.04544507 0.030668714 0.023077442 0.027331115
## 2 0.04544507 0.00000000 0.020989556 0.026988175 0.026376945
## 3 0.03066871 0.02098956 0.000000000 0.007807747 0.005450661
## 4 0.02307744 0.02698817 0.007807747 0.000000000 0.004707708
## 5 0.02733111 0.02637694 0.005450661 0.004707708 0.000000000
```

```
as.matrix(ponds.dists)[1:5, 1:5]
```

```
##            1          2           3           4           5
## 1 0.00000000 0.04544507 0.030668714 0.023077442 0.027331115
## 2 0.04544507 0.00000000 0.020989556 0.026988175 0.026376945
## 3 0.03066871 0.02098956 0.000000000 0.007807747 0.005450661
## 4 0.02307744 0.02698817 0.007807747 0.000000000 0.004707708
## 5 0.02733111 0.02637694 0.005450661 0.004707708 0.000000000
```

```
#ponds.dists.inv[is.infinite(ponds.dists.inv)] <- 0
mantel.rtest(ponds.dists, env.dists, nrepet = 9999)
```

```
## Warning in is.euclid(m1): Zero distance(s)
```

```
## Warning in is.euclid(m2): Zero distance(s)
```

```
## Warning in is.euclid(distmat): Zero distance(s)
```

```
## Monte-Carlo test
## Observation: 0.2005394
## Call: mantel.rtest(m1 = ponds.dists, m2 = env.dists, nrepet = 9999)
## Based on 9999 replicates
## Simulated p-value: 0.002
```

ii. Describing and visualizing autocorrelation These pages demonstrate how to generate a variogram for your data. A variogram (or semi-variogram) gives you a sense of the degree and range of spatial autocorrelation in your data and how it changes over distances.

A measure of the dispersion of all observations that fall below the mean or target value of a data set. Semivariance is an average of the squared deviations of values that are less than the mean. Semivariance is similar to variance; however, it only considers observations below the mean.

```
#install.packages("geoR")
library(geoR)
```

```
## Loading required package: sp
## Loading required package: MASS
## --------------------------------------------------------------
##  Analysis of geostatistical data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-4.1 (built on 2012-06-29) is now loaded
## --------------------------------------------------------------
```

```
dists <- dist(cbind(lats, lons))
summary(dists)
```

```
## Class: dist
## Distance matrix by lower triangle : d21, d22, ..., d2n, d32, ...
## Size: 52
## Labels:
## call: dist(x = cbind(lats, lons))
## method: euclidean
```

```
## Warning in is.euclid(object): Zero distance(s)
```

```
## Euclidean matrix (Gower 1966): TRUE
```

```
breaks = seq(0, 1.5, l = 11)
v1 <- variog(coords = cbind(lats,lons), data = Ponds$Temp, breaks = breaks)
```

```
## variog: computing omnidirectional variogram
## variog: co-locatted data found, adding one bin at the origin
```

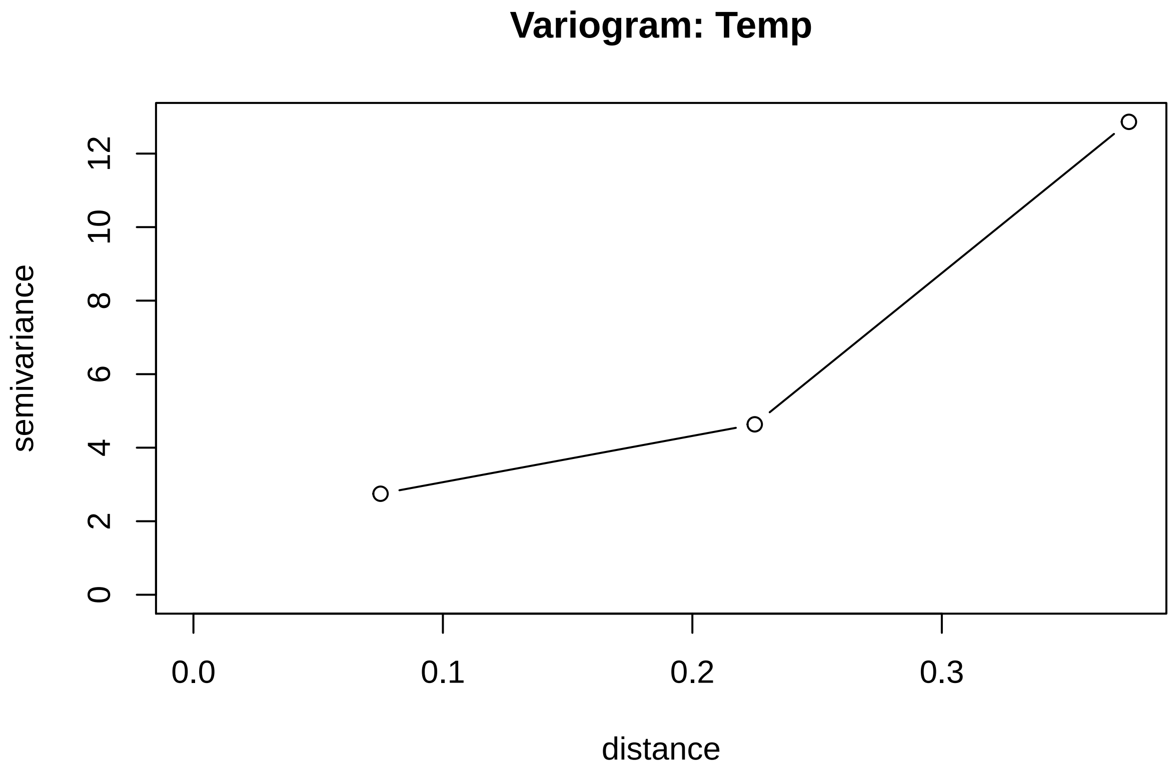
```
v1.summary <- cbind(c(1:10), v1$v, v1$n)
```

```
## Warning in cbind(c(1:10), v1$v, v1$n): number of rows of result is not a
## multiple of vector length (arg 2)
```

```
colnames(v1.summary) <- c("lag", "semi-variance", "# of pairs")
v1.summary
```

```
##       lag semi-variance # of pairs
##  [1,]   1      2.747398       1055
##  [2,]   2      4.635415        265
##  [3,]   3     12.863710          5
##  [4,]   4      2.747398       1055
##  [5,]   5      4.635415        265
##  [6,]   6     12.863710          5
##  [7,]   7      2.747398       1055
##  [8,]   8      4.635415        265
##  [9,]   9     12.863710          5
## [10,]  10      2.747398       1055
```

```
plot(v1, type = "b", main = "Variogram: Temp")
```



iii. Adjusting for (or predicting with) autocorrelation Here, we will find the empirical variogram that is closest to what you observe in your data and using this theoretical variogram to predict your outcome at unobserved locations.

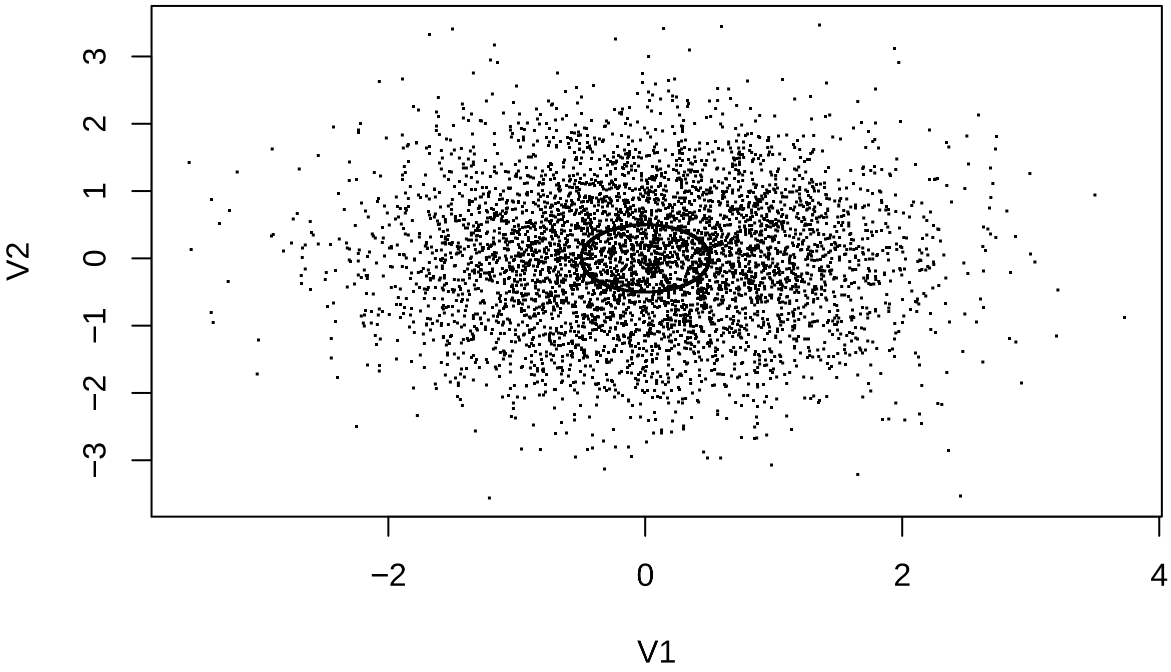
Concept 4: Scale dependence

i. extent and grain

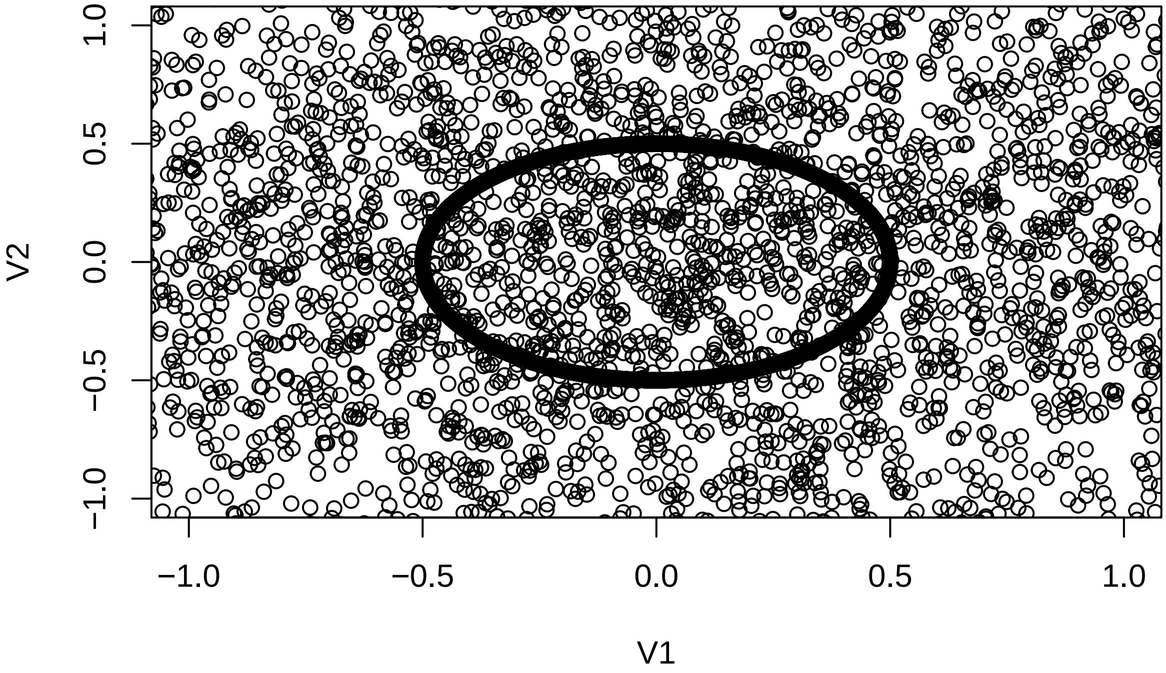
ii. How interpretation changes with extent

Here, we generate a single sample of data based on a random draw from a normal distribution (aka Gaussian distribution, bell curve). While all the points were randomly drawn from the same distribution, zooming in and out to different extents reveals what we would very likely call different patterns of aggregation.

```
# generate the data
set.seed(20111105)
x = rbind(matrix(rnorm(10000), ncol = 2), local({
  r = runif(10000, 0, 2 * pi)
  0.5 * cbind(sin(r), cos(r))
}))

x = as.data.frame(x[sample(nrow(x)), ])

plot(x, pch = ".") # The distribution of points is clearly aggregated around the center of the graph (i
```



```
plot(x, xlim = c(-1, 1), ylim = c(-1, 1)) # Zooming in, our first guess would not be that these points
```



7.) PRIMARY SPATIAL PATTERNS

Spatially implicit patterns of aggregation and diversity in the site-by-species matrix

i. Spatial abundance distribution

ii. Abundance-occupancy relationship

iii. Occupancy-frequency distribution

Spatially explicit patterns of beta-diversity

i. Species-level: Do near sites have more similar (relative) abundances than far sites?

ii. Distance-decay relationship: Do near communities have greater similarity than far communities?

Cumulative patterns

i. Species-area relationship Do we accumulate species with greater area just because there is more area or because greater area means more niches?

8.) ENTER, THE ENVIRONMENT

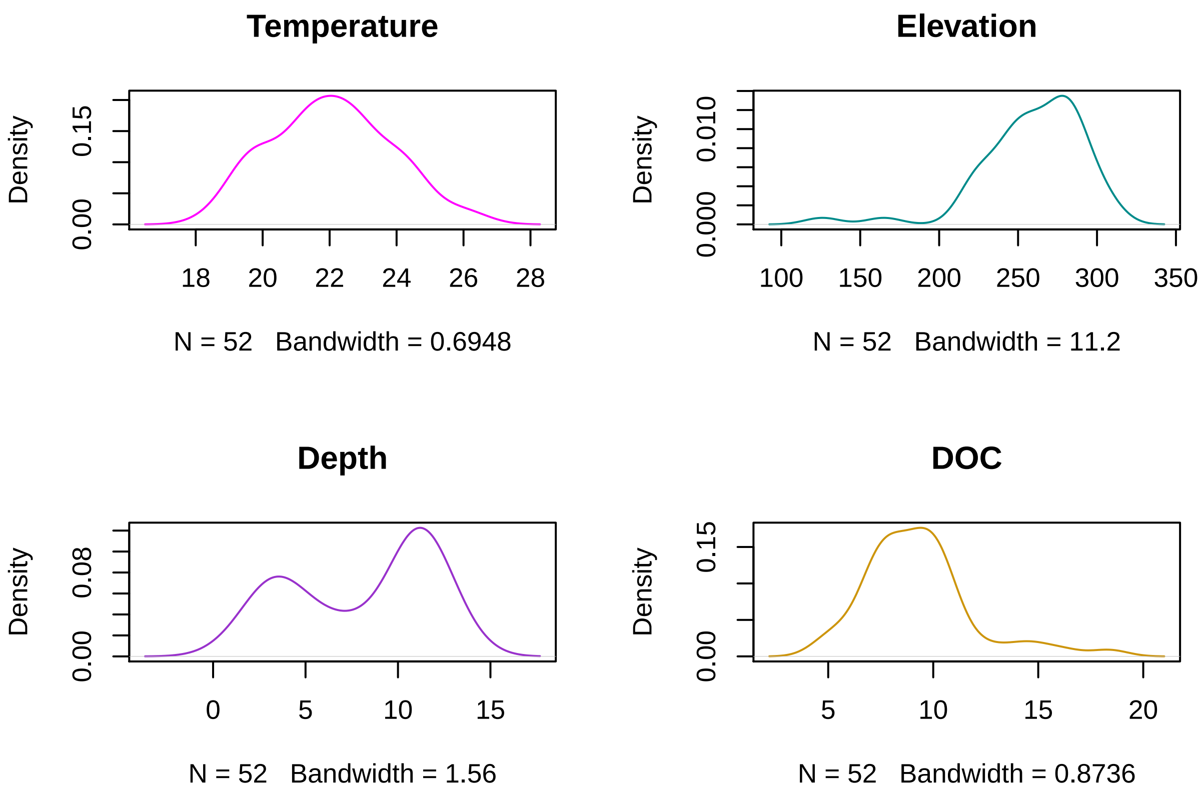
i. Graphically examine environmental data

It is good practice to look at the underlying distribution of your data, before conducting formal statistical tests. By conducting a simple graphical analysis, we can ask whether our data appear to be normally distributed, highly skewed, multi-modal (i.e. two distinct distributions of data where we might have only expected one).

Let's begin with a simple graphical exploration of the environmental data collected from the refuge ponds. We will use **kernel density curves**, which are analogous to histograms, but are more appropriate for continuous data because they avoid the arbitrary creation of bins (i.e. bars). Kernel density curves attempt to account for uncertainty and sampling error, i.e., in revealing the probability that a randomly drawn data point will take a value within a particular range.

Include units, e.g., degrees C, meters = elevation?, feet!?!? = depth, mg/L = DOC

```
EnvData <- Ponds[5:21]
par(mfrow=c(2, 2))

data <- as.numeric(EnvData[, "Temp"])
plot(density(data), col = 'magenta', main = "Temperature")

data <- as.numeric(EnvData[, "Elevation"])
plot(density(data), col = 'DarkCyan', main = "Elevation")

data <- as.numeric(EnvData[, "Depth"])
plot(density(data), col = 'darkorchid', main = "Depth")

data <- as.numeric(EnvData[, "DOC"])
plot(density(data), col = 'darkgoldenrod3', main = "DOC")
```



Question 3: In the density() function, what does "bandwidth", i.e., "bw" mean?

> ***Answer 3***:

Question 4: Describe the kernel density plots (think of them as smoothed histograms) of the environmental variables in the refuge ponds.

> ***Answer 4***:

ii. Graphically examine taxonomic data

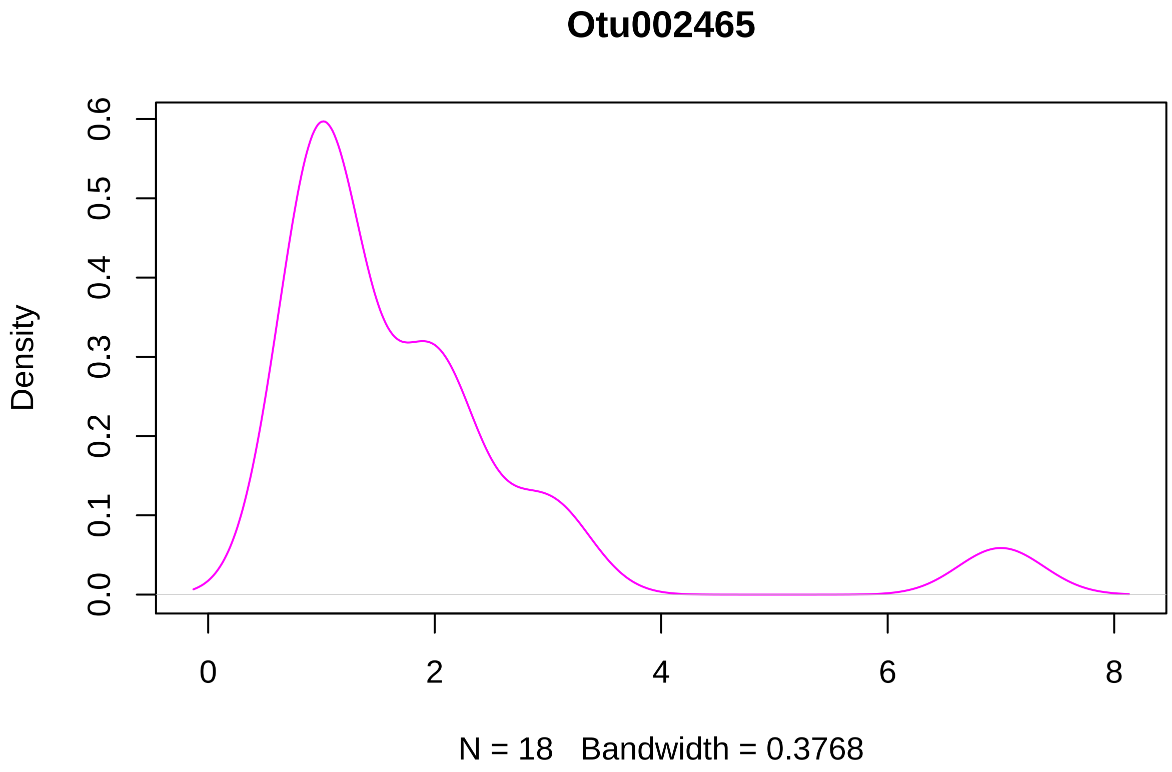
Load the site-by-OTU matrix for samples taken from ponds in and around Brown County, Indiana. **perhaps we need to describe the biological data more above? Bacterial 16S rRNA genes, etc.**

```r
# Site by OTU matrix
OTUs <- read.csv(file="BrownCoData/SiteBySpecies.csv", head=TRUE, sep=",")
otu.names <- names(OTUs)
OTUs <- as.data.frame(OTUs[-1]) # remove first column (site names)
Site.N <- as.vector(rowSums(OTUs)) # no. reads
```

Now, let's construct kernel-density curves for the distribution of abundance among individual OTU's across the refuge pond dataset.

Introducing loops

```
ad <- c(0,0)
otu <- 1
while (length(ad) <= 10){
  otu <- sample(1:length(OTUs), 1)

  ad <- OTUs[, otu]
  ad = as.vector(t(x = ad))
  ad = ad[ad > 0]

  }

plot(density(ad), col = 'magenta', main = otu.names[otu])
```



As you can see, sampled abundance for a given OTU is often aggregated, revealing many sites where the OTU is relatively rare and many where it is relatively more common. **In fact, this uneven spatial distribution of abundance is common occurrence in ecological systems.** This spatially implicit distribution of abundance is often referred to as the species spatial abundance distribution (SSAD).

I think the concept here needs to be more developed. Maybe we'll talk about SSAD in lecture. I'm afraid most students will have never heard of SSAD. Need to explain. X-axis is not labeled, so not sure what we're looking at. Also, perhaps this is an opportunity to ask a question as opposed to telling them what they're seeing. learning opp

** Should SSAD be it's own section as opposed to be nestled in with "aggregation"? Seems like theory or concept is a little burried**

Question 1: In the site-by-species matrix, each row represents a site and each column represents an OTU. If the SSAD is generated by considering all rows for a single column, then what do we obtain when we consider all columns for a given row? Have we examined this sort of data structure before? If so, elaborate?

See comment above about introducing a new core concept (i.e., SSAD). Need to teach them what this is first!

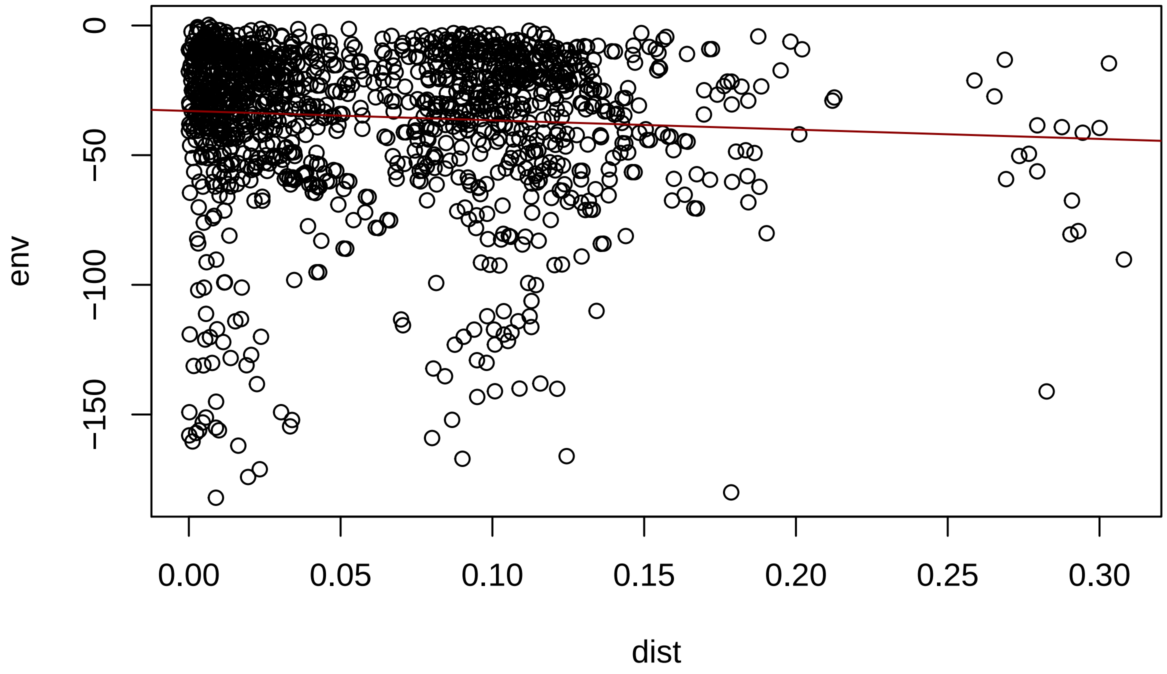
> ***Answer 1***:

5) THE DISTANCE-DECAY RELATIONSHIP

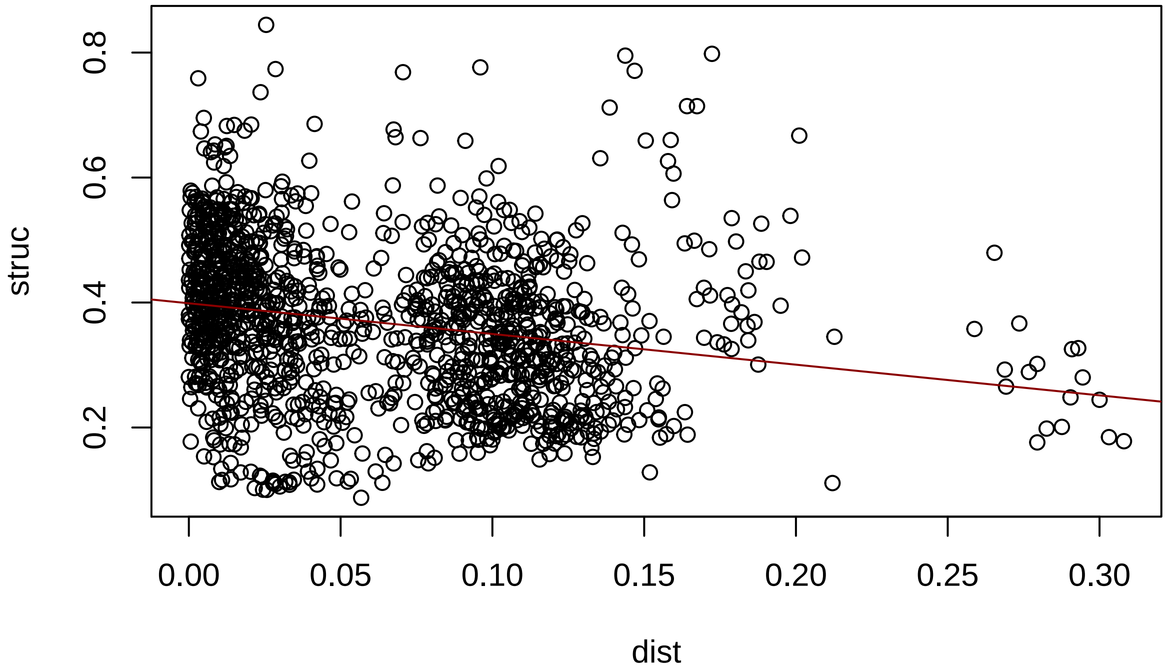
First law of geography…spatial autocorrelation…decrease in similarity with distance… **Lot more text and development needed* **Lots of code for them to type out**

```
#install.packages("simba")
library(simba) #vs. require()???#
```

```
# calculate the similarity (Bray-Curtis) between the plots
struc.dist <- 1 - vegdist(OTUs)
# calculate geographical distance between plots
coord.dist <- dist(as.matrix(lats,lons))

# transform environmental data to numeric types
temp <- as.numeric(EnvData[, "Temp"])
elev <- as.numeric(EnvData[, "Elevation"])
depth <- as.numeric(EnvData[, "Depth"])
doc <- as.numeric(EnvData[, "DOC"])

# calculate the distance (Euclidean) between the plots regarding environmental variables
env.dist <- 1 - vegdist(cbind(temp, elev, depth, doc), "euclidean")

# transform all distance matrices into list format:
struc.dist.ls <- liste(struc.dist, entry="struc")
env.dist.ls <- liste(env.dist, entry="env")
coord.dist.ls <- liste(coord.dist, entry="dist")

# create a data frame containg plot information, geographical distance,
# similarity of environment, and similarity of community
df <- data.frame(coord.dist.ls, env.dist.ls[,3], struc.dist.ls[,3])
names(df)[4:5] <- c("env", "struc")
attach(df)
#df <- subset(df, struc != 0)

# plot Distance-Decay relationships with regression lines in red
par(mfrow=c(1, 1))
plot(dist, env)
abline(lm(env ~ dist), col="red4")
```



```
par(mfrow=c(1, 1))
plot(dist, struc)
abline(lm(struc ~ dist), col="red4")
```



```
# Is the slope significantly different?
res <- diffslope(dist, env, dist, struc)
res
```

```
##
## Is difference in slope significant?
## Significance is based on 1000 permutations
##
## Call:
## diffslope(x1 = dist, y1 = env, x2 = dist, y2 = struc)
##
## Difference in Slope: -35.44
## Significance: 0.012
##
## Empirical upper confidence limits of r:
##   90%   95% 97.5%   99%
##  20.8  27.6  31.8  35.6
```

6) THE SPECIES-AREA RELATIONSHIP

The species-area relationship (SAR) describes the rate at which we discover or accumulate species with increasing area. The general relationship of the SAR … and is of the form S=c*A^z … which was first predicted by Arrhenius (1921)…wow, that's old.

Random accmulation of sites

Accumulation of sites by proximity

```
OTU.obs <- function(x = ""){
  rowSums(x > 0) * 1
  }
```

By topography we mean physical properties that contribute to the geographical complexity of an environment, e.g., elevational changes.