Intro to R for Statistics

2020/10/09

Welcome to the Adv Stats course

Labs:

- 4/5 of them
- You have to submit only 2
- You'll prepare a simple report for each
- Groups of 2 students
- You should submit them before the deadline
- to: **adv.statistics.2020@gmail.com**

We are going to use R

Why R?

- open source
- a lot of **Stats libraries**

Also:

- nice and easy plots (`ggplot2`)
- complete environment to do reproducible science (`RMarkdown` + Git)
- "easy"-to-build prototypes (Shiny)

Basic data types

- double/numeric
- integer
- logical
- character

```
pi * 2^2 # numeric/double
```

```
## [1] 12.56637
```

```
(TRUE && (2 < 1)) || TRUE # logical (TRUE or FALSE)
```

```
## [1] TRUE
```

```
word1 <- "hello"
word2 <- "world"
paste(word1, word2) # concatenate
```

```
## [1] "hello world"
```

Use `class(object)` to get the data type of `object`

Missing values

Denoted with NA

```
missing <- NA
missing
```

```
## [1] NA
```

```
missing == NA # don't do this
```

```
## [1] NA
```

```
is.na(missing)
```

```
## [1] TRUE
```

More data types: vector

```
vector1<- c(1, 5, 9, -1, 4)
vector1
```

```
## [1]  1  5  9 -1  4
```

```
seq(1, 2, 0.5)
```

```
## [1] 1.0 1.5 2.0
```

```
rep(5, 7)
```

```
## [1] 5 5 5 5 5 5 5
```

Remember that indexing starts from 1

More data types: vector1=1, 5, 9, -1, 4

```
vector1[c(1,2)]
```

```
## [1] 1 5
```

```
c(vector1,c(1,2))
```

```
## [1]  1  5  9 -1  4  1  2
```

```
vector1[-2]
```

```
## [1]  1  9 -1  4
```

```
vector1 < 5
```

```
## [1]  TRUE FALSE FALSE  TRUE  TRUE
```

```
sum(vector1 < 5)
```

```
## [1] 3
```

More data types: lists

can contain any R object

```
my_list <- list(name="John", age=35, children=c("Jane", "Hannah"))
my_list$name
```

```
## [1] "John"
```

```
my_list[[2]]
```

```
## [1] 35
```

```
str(my_list)
```

```
## List of 3
##  $ name    : chr "John"
##  $ age     : num 35
##  $ children: chr [1:2] "Jane" "Hannah"
```

Some more syntax

if, for, while

```
if (guard) {
  do something
} else {
  do another thing
}
```

functions

```
sum_2 <- function (number) {
  return(number + 2)
}
sum_2(3)
```

```
## [1] 5
```

Datasets

- the class can be `data.frame` (r base) or `tibble`(tidyverse)
- function to read csv `read.csv` (r base) or `read_csv` (readr from tidyverse)
- to copy-paste from unfriendly documents (pdf or website) you can try the library `datapasta`

Simulations

```
sample(c(0, 1), 10, replace=TRUE)
```

```
##  [1] 1 1 0 0 0 1 0 1 0 0
```

```
sample(c(0, 1), 10, replace=TRUE, prob=c(0.7, 0.3))
```

```
##  [1] 1 0 0 1 0 0 0 1 0 1
```

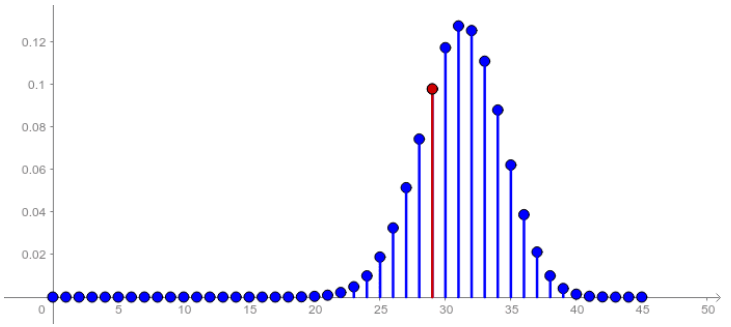
For reproducible analysis:

```
set.seed(0906)
sample(c(0,1),5,replace=TRUE)
```

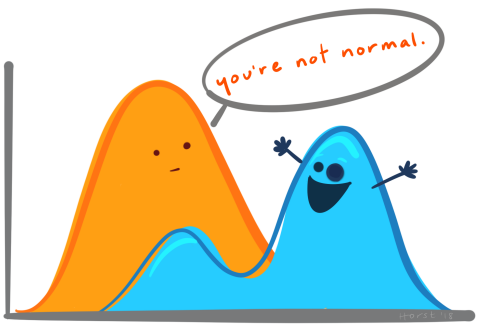
```
## [1] 0 0 0 1 0
```

Famous variables

- Binomial (coins)



- Gaussian (almost everything)



Artwork by @allison_horst

Famous variables

Binomial Bi(10,1/2):

```
dbinom(5,10,0.5)    # puntual prob
```

```
## [1] 0.2460938
```

```
pbinom(2,10,0.5)    # cummulative prob
```

```
## [1] 0.0546875
```

```
qbinom(0.5,10,0.5)  # quantile
```

```
## [1] 5
```

```
rbinom(20,10,0.5)   # random samples
```

```
##  [1] 6 3 5 3 5 4 3 3 5 7 4 3 6 4 6 2 5 6 8 5
```

also: *geom, *dnbinom, *norm, *gamma, *beta, etc.

Approximating probabilities

Because of law of large numbers:

P("event") is approximated by f_n if n is big enough

f_n: relative frequence of appeareance of "event" in n repetitions

(MONTE CARLO simulations)

EX 1

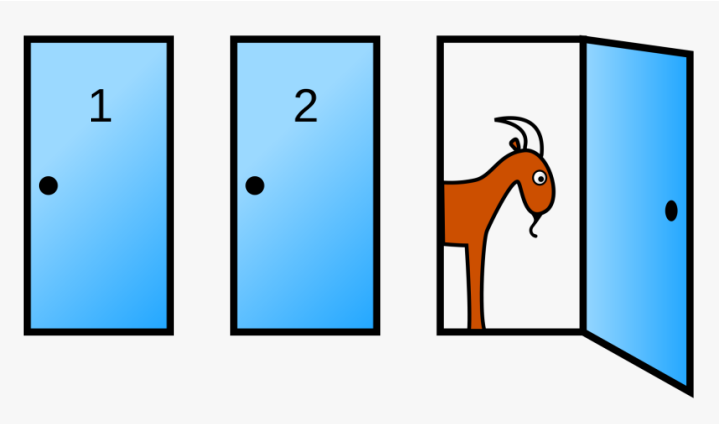
We repetedly toss a biased coin with probability 0.4 of showing head (H) and 0.6 of showing tail (T). In this sequence, a group is a consecutive sequence of tosses from the same side of the coin. For example, the groups from HTTTHTHTT are:

(H)(TTT)(H)(T)(H)(TT)

Give a decent approximate answer for the following questions:

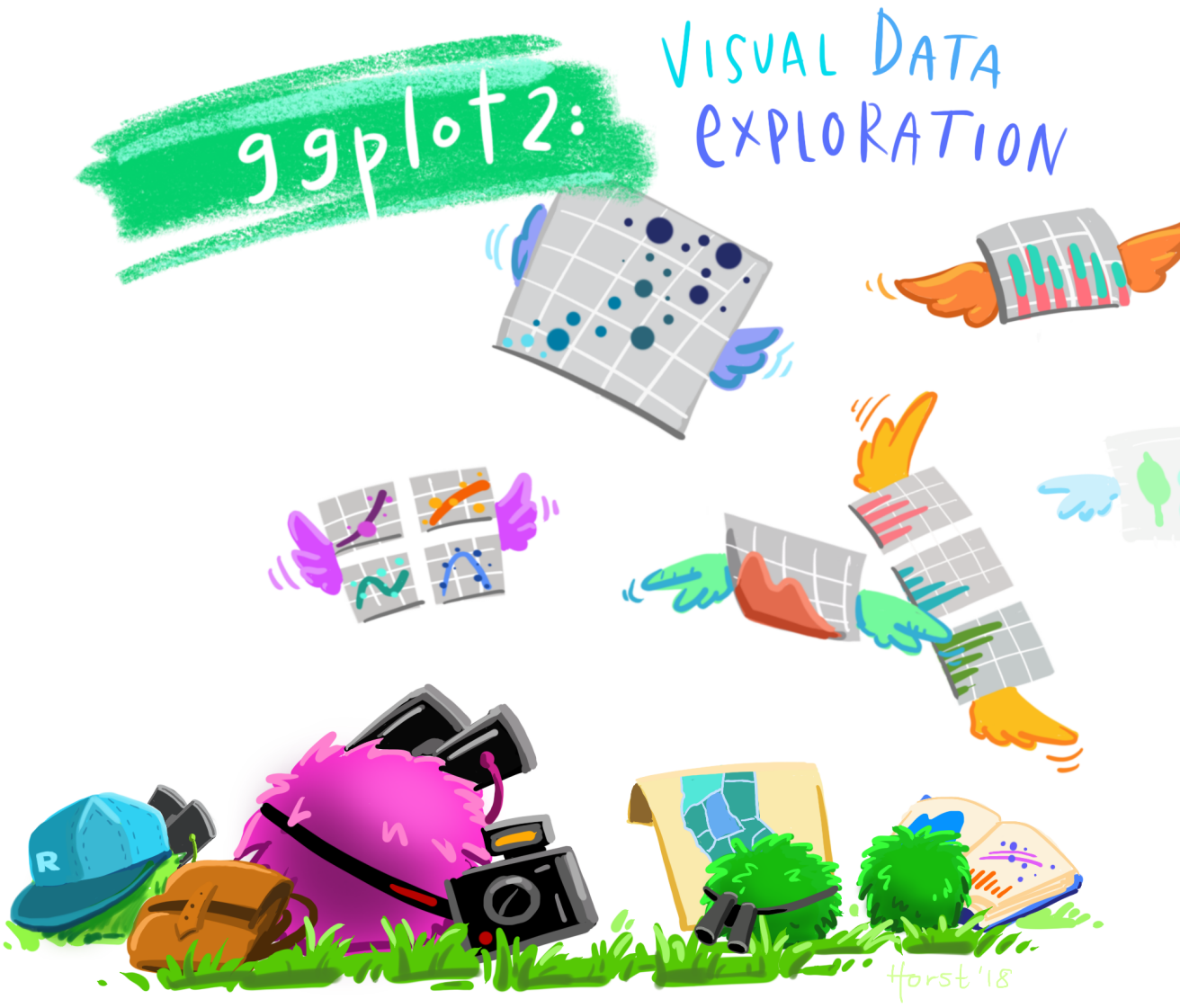
1. ¿Which is the probability of exceding 5 groups in a sequence of 10 tosses?
2. ¿Which is the expected number of groups in a sequence of 10 tosses?

EX 2



Approximate Monty Hall probabilities of winning for the 2 possible strategies

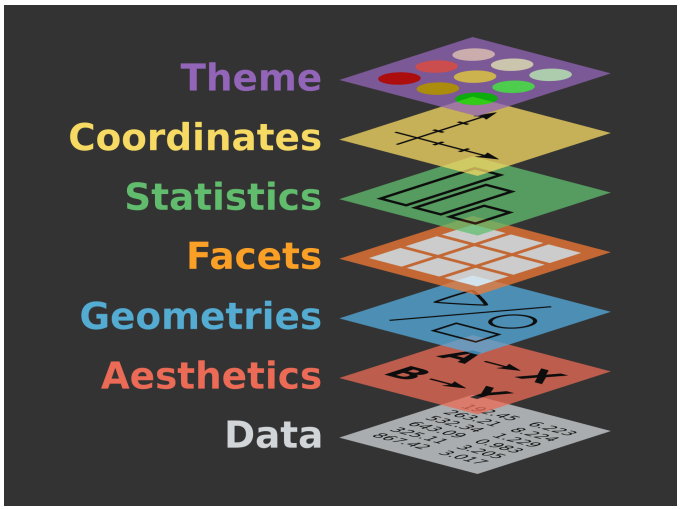
Change the rules and recompute the approximated probabilities


ggplot2:
VISUAL DATA EXPLORATION
R
Horst '18

ggplot2

Library to create plots in R

Graphs are linked to a dataset and aesthetics of the graphic are linked to variables



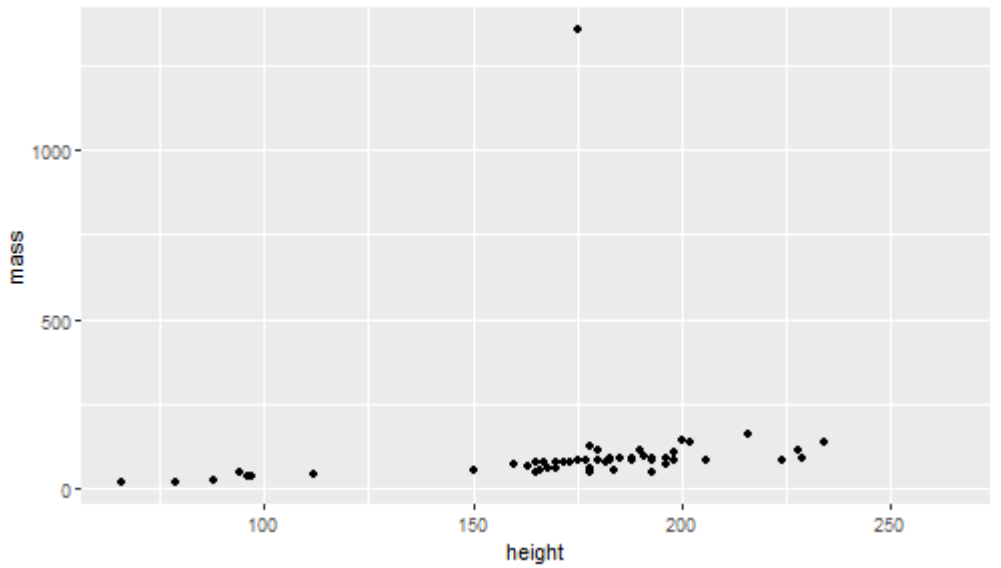
```
ggplot(dataset, aes(x = var1, y = var2)) +
  geom_*() +
  theme_*()
```

```
dplyr::starwars
```

```
## # A tibble: 87 x 13
##    name  height  mass hair_color skin_color eye_color birth_year gender
##    <chr>  <int> <dbl> <chr>      <chr>      <chr>           <dbl> <chr>
##  1 Luke…    172    77 blond      fair       blue             19   male
##  2 C-3PO    167    75 <NA>       gold       yellow          112   <NA>
##  3 R2-D2     96    32 <NA>       white, bl… red              33   <NA>
##  4 Dart…    202   136 none       white      yellow           41.9 male
##  5 Leia…    150    49 brown      light      brown            19   female
##  6 Owen…    178   120 brown, gr… light      blue             52   male
##  7 Beru…    165    75 brown      light      blue             47   female
##  8 R5-D4     97    32 <NA>       white, red red              NA   <NA>
##  9 Bigg…    183    84 black      light      brown            24   male
## 10 Obi-…    182    77 auburn, w… fair       blue-gray        57   male
## # … with 77 more rows, and 5 more variables: homeworld <chr>, species <chr
## #   films <list>, vehicles <list>, starships <list>
```

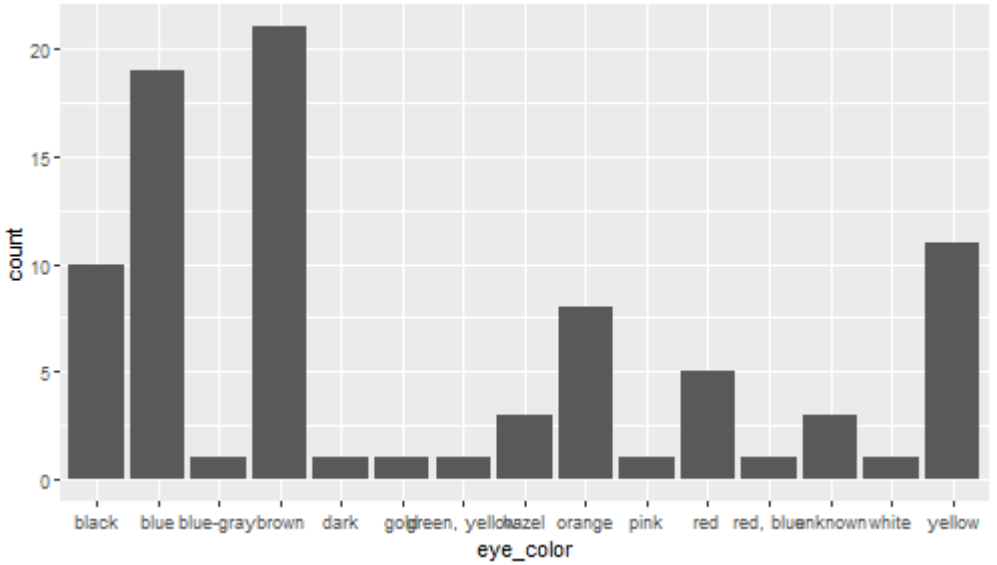
ggplot2: scatter plot

```
ggplot(dplyr::starwars, aes(x = height, y = mass)) +
  geom_point()
```



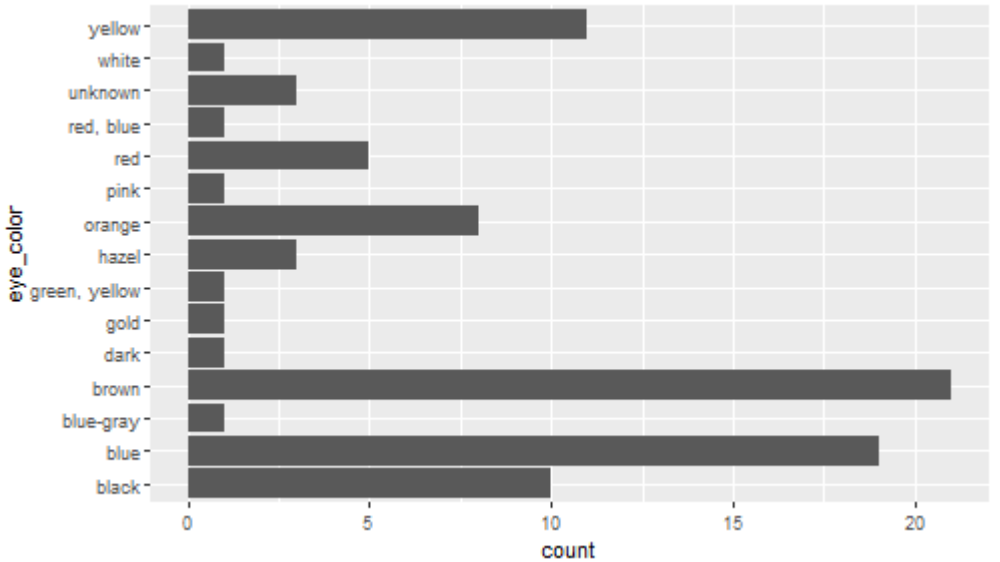
ggplot2: bar plot

```
ggplot(dplyr::starwars, aes(x = eye_color)) +
  geom_bar()
```



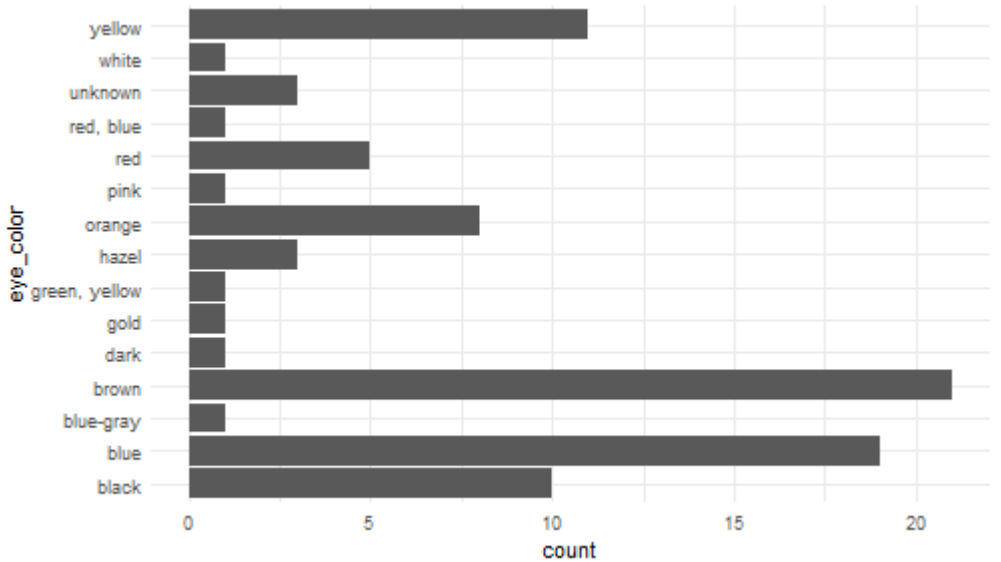
ggplot2: bar plot

```
ggplot(dplyr::starwars, aes(x = eye_color)) +
  geom_bar() +
  coord_flip()
```



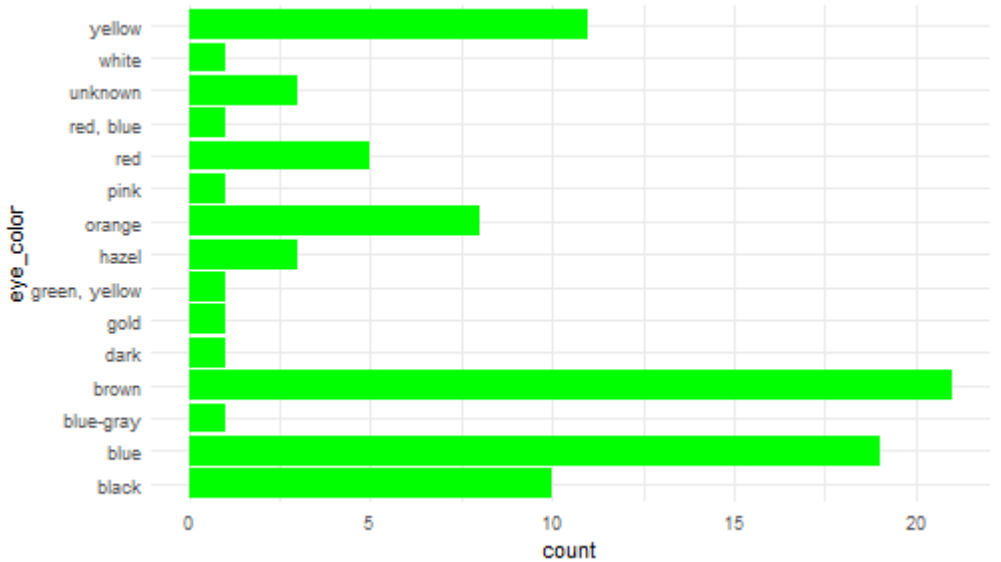
ggplot2: bar plot

```
ggplot(dplyr::starwars, aes(x = eye_color)) +
  geom_bar() +
  coord_flip() +
  theme_minimal()
```



ggplot2: bar plot

```
ggplot(dplyr::starwars, aes(x = eye_color)) +
  geom_bar(fill = "green") +
  coord_flip() +
  theme_minimal()
```

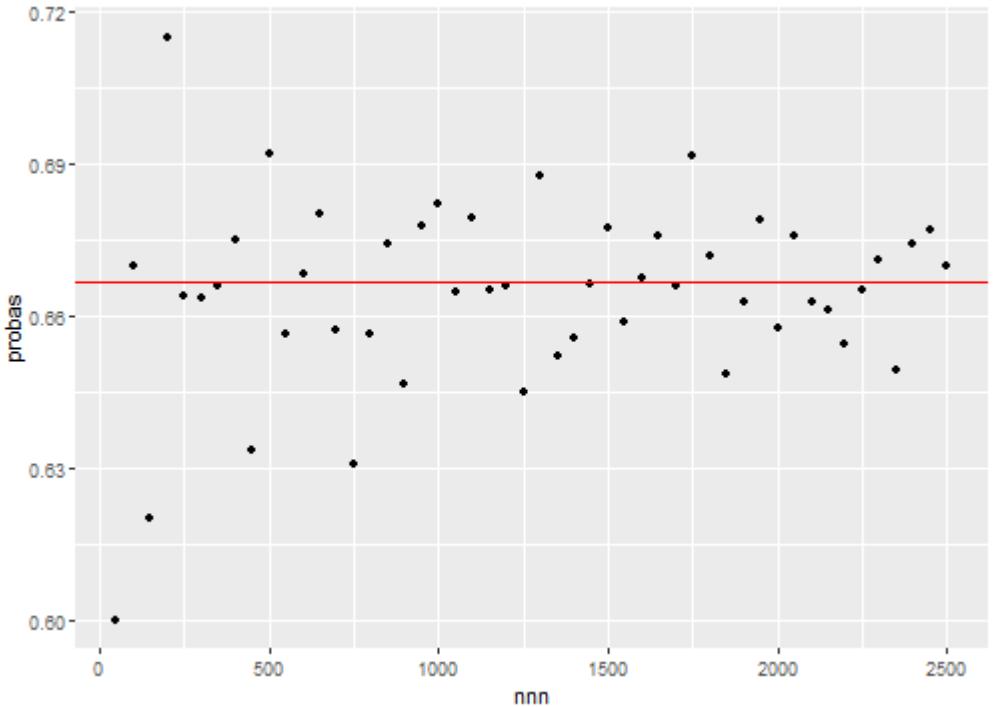


EX 3

- Generate $n = 10$ gaussian samples with mean 3 and sd 0.5
- Plot histogram
- Plot the theoretical distribution on the same plot
- Change n

EX 4

Reproduce the following graph that shows the estimated probabilities of winning in the Monty Hall Game with the changing strategy as function of the number of repetitions of the Monte Carlo simulations. The theoretical probability of winning is plotted in red.



Artwork by @allison_horst

For your reports

You will use **RMarkdown** (.Rmd extension)

- Code chunks
- Text with Markdown mark-up
- LaTeX formules
- Can build .pdf, .html, .doc, etc.

Next course

- We'll solve the excercises in the "Applications" slides
- We'll empirically verify the estimator properties with simulations