Joy of R

From Excel to Much More

Vincent D. Warmerdam - @fishnets88 - koaning.io - GoDataDriven

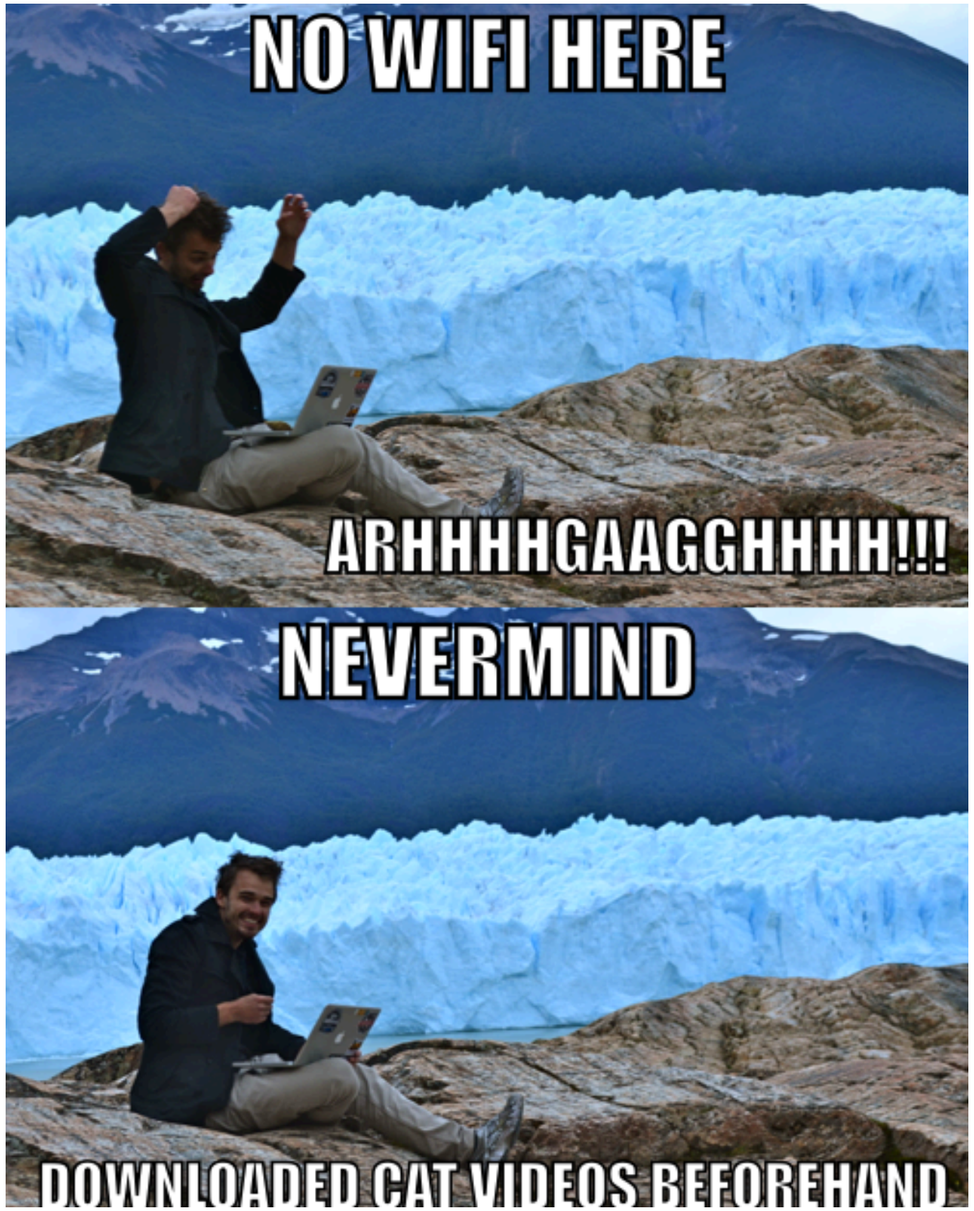
My Story

I was about to graduate econometrics 4 years ago.

- I worked for a business intelligence company.
- Wore a suit to work everyday and I clicked on a lot of buttons to make powerpoint slides. Such was the life of the post-student enterprise lifestyle.

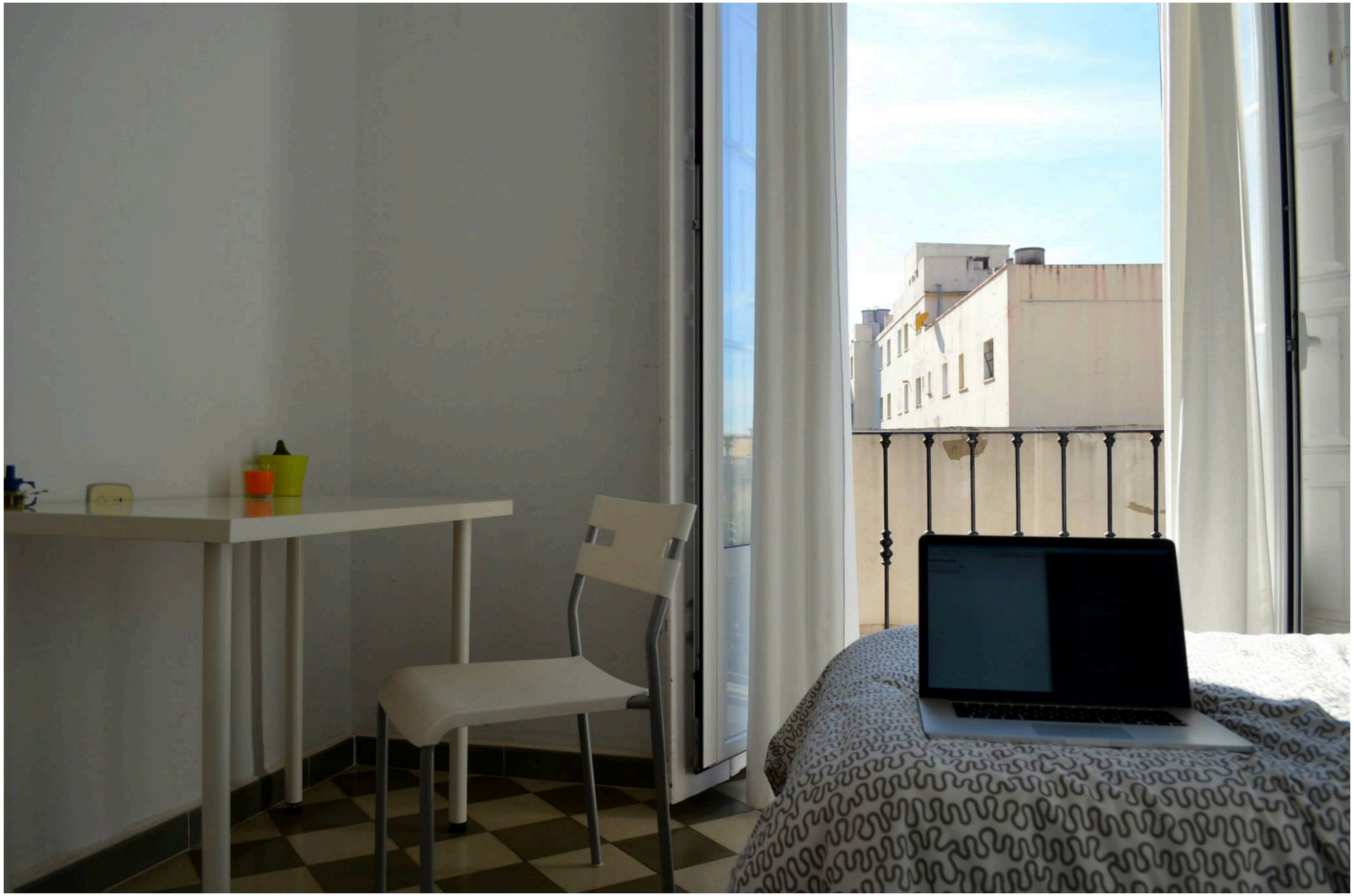
Then my boss asked me for a rather meaningless task ... this sort of changed my life from guy-in-suit to an internet meme.

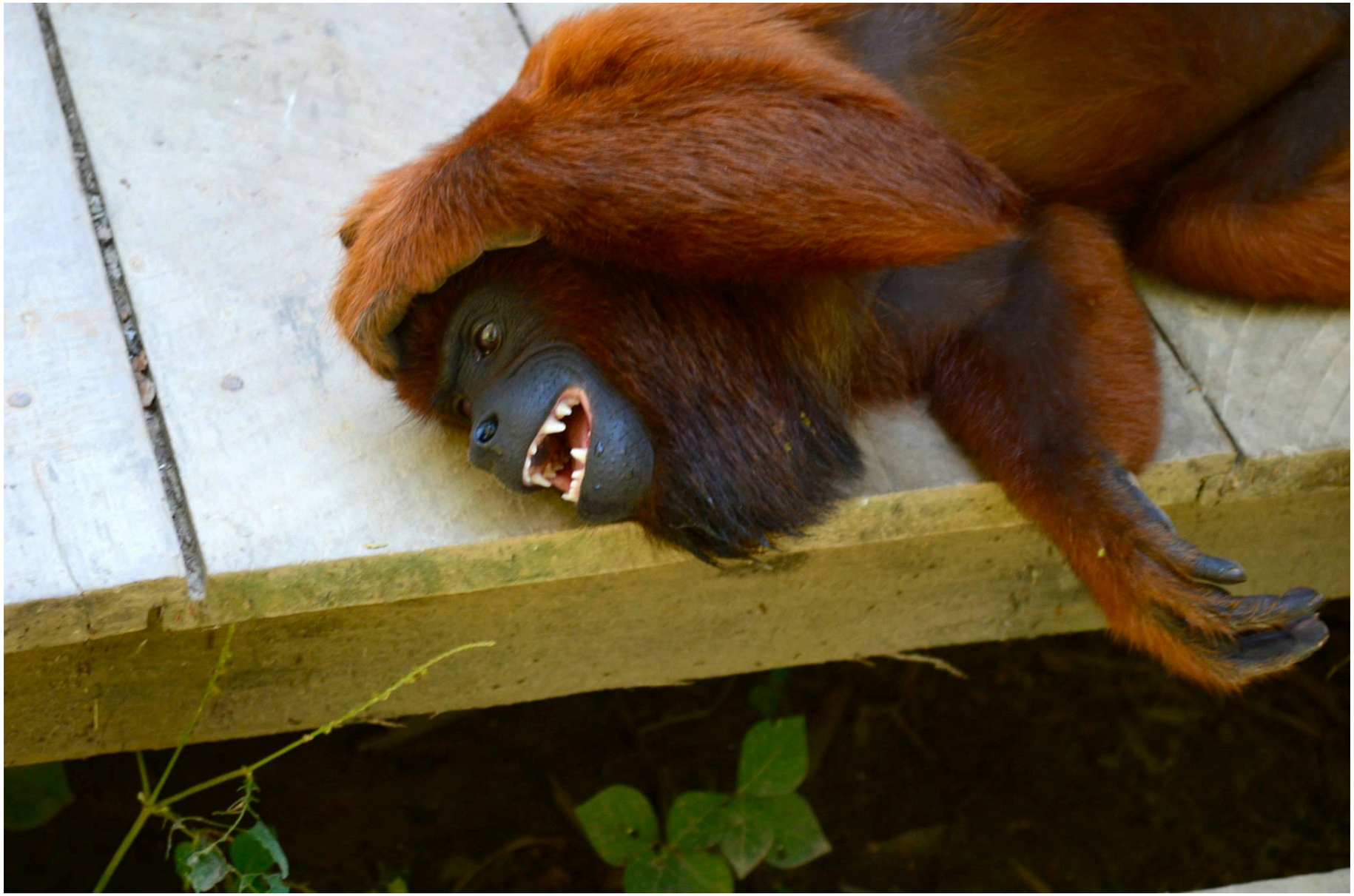

THE WIFI SUCKS HERE
WANT MONEY BACK


NO WIFI HERE
ARHHHHGAAGGHHHH!!!
NEVERMIND
DOWNLOADED CAT VIDEOS BEFOREHAND

"As much as I love cat videos, who actually downloads them and maintains a stash?"


Basics of
Qualitative
Research

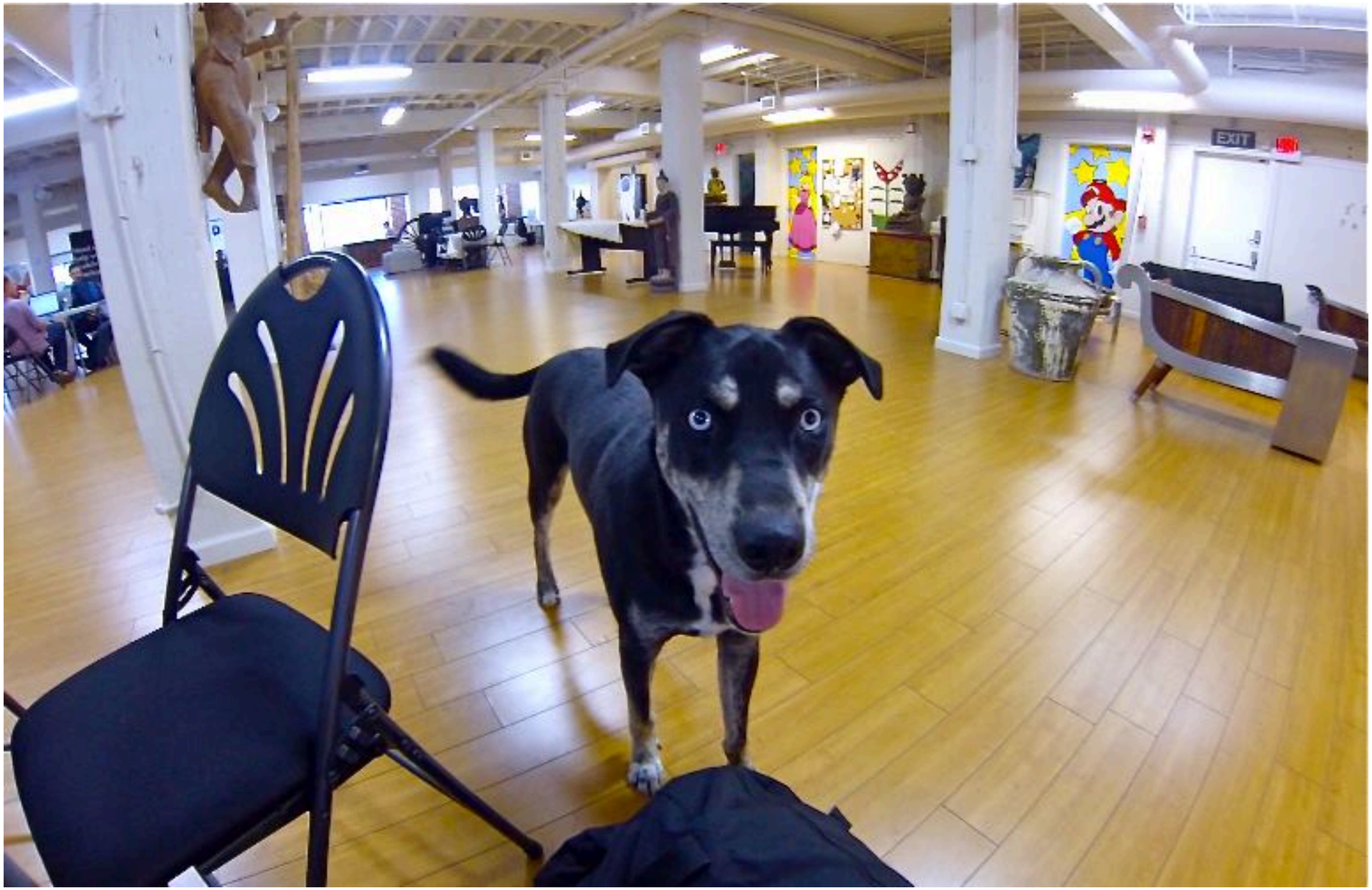




Vincent D. Warmerdam - @fishnets88 - koaning.io - GoDataDriven


Wi Fi
¡Estamos conectados!
En esta zona el internet es GRATIS.
mira flores
en tu corazón


ETERNAL WAVE


EXIT


FORCLAZ 50 ultralight

Fast Forward a While

Vincent D. Warmerdam - @fishnets88 - koaning.io - GoDataDriven

I'm also this guy.

Europe

Mango Business Solutions Ltd
UK

Mango is a leading data science and data analytics company offering customized solutions, training, consultancy, application development and support.

Click for Description

Vincent Warmerdam
Amsterdam, Netherlands

Vincent D. Warmerdam is a data scientist at GoDataDriven, the leading big data science consultancy firm in the Netherlands.

Click for Biography

Vincent Guyader
France

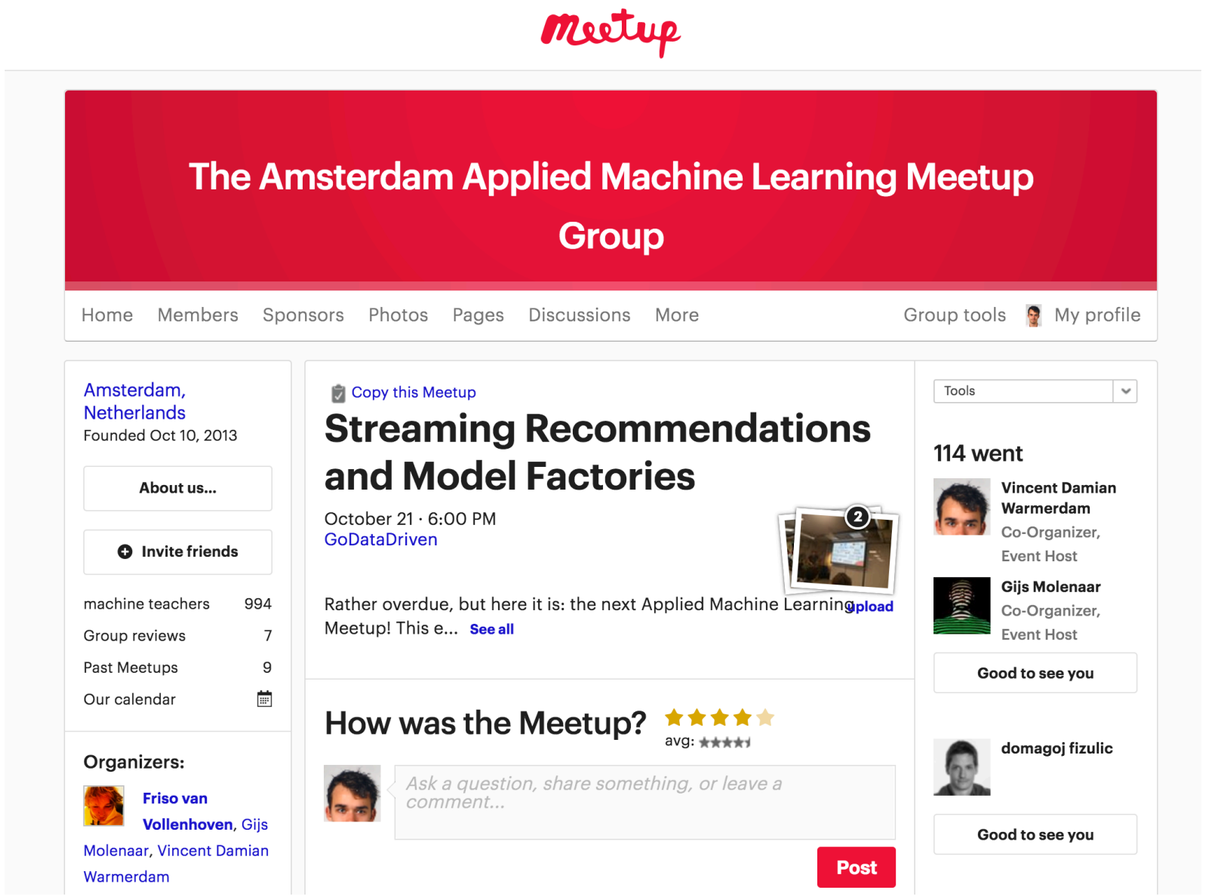
Vincent Guyader is a a french statistician and President of ThinkR, a company which helps businesses migrate to R. He teaches R in finance, insurance, marketing, and life sciences.

I'm also this guy.

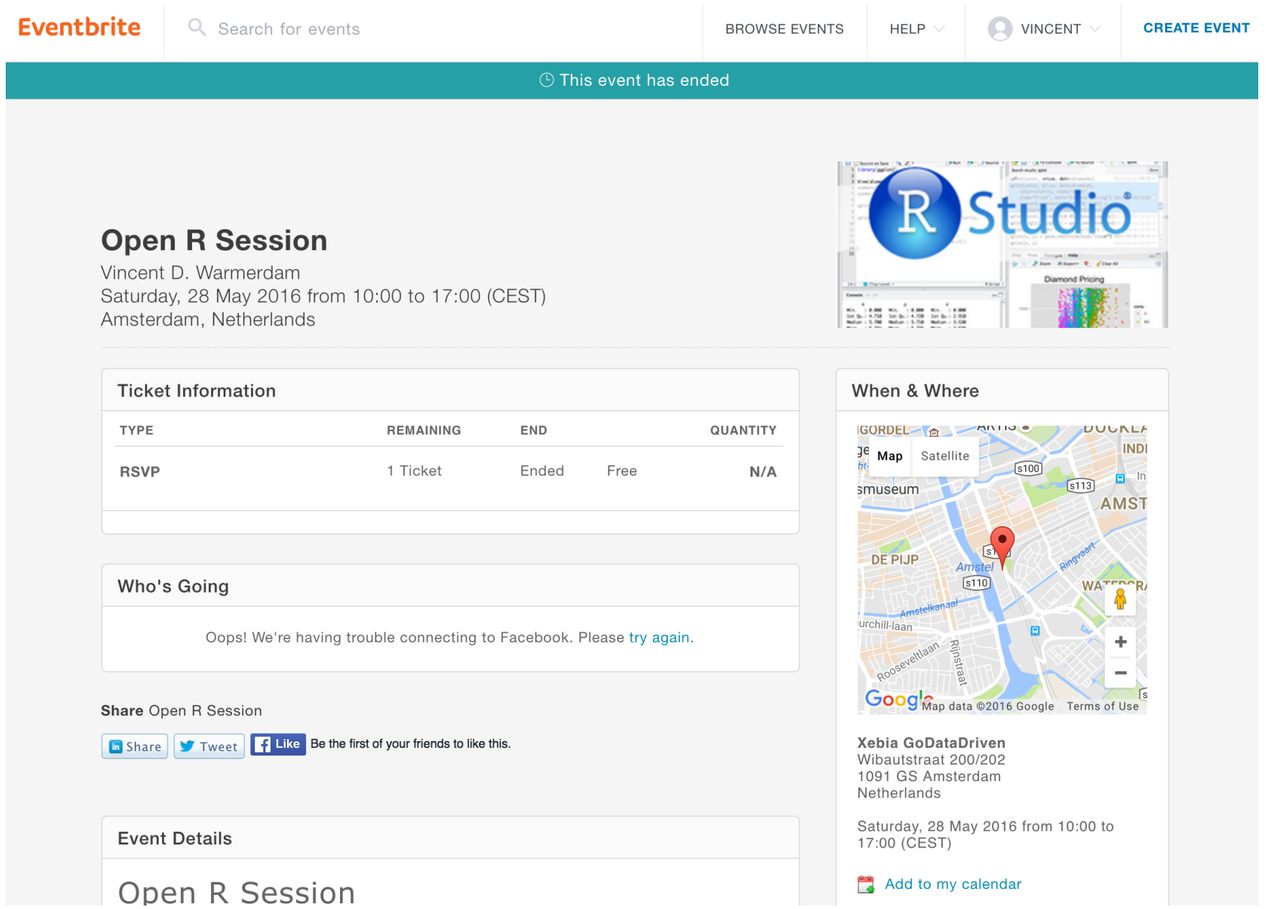
PyData

Amsterdam 2016

I'm also this guy.



I'm also this guy.



I'm also this guy.

python, javascript, R, dplyr, ggplot, ditto, scala, mongo, html5, bootstrap, git, sublime, d3, leaflet, sawk, pandas, feebas, numpy, scikit, nltk, crebase, juypter, onyx, lodash, mesos, docker, django, flask, neo4j, vulpix, selenium, node-webkit, hadoop, hoopa, impala, spark, azurill, ansible, hadoop, mapreduce.

I'm also this guy.

python, javascript, R, dplyr, ggplot, ditto, scala, mongo, html5, bootstrap, git, sublime, d3, leaflet, sawk, pandas, feebas, numpy, scikit, nltk, crebase, juypter, onyx, lodash, mesos, docker, django, flask, neo4j, vulpix, selenium, node-webkit, hadoop, hoopa, impala, spark, azurill, ansible, hadoop, mapreduce.

Can you recognize which ones are pokemon?

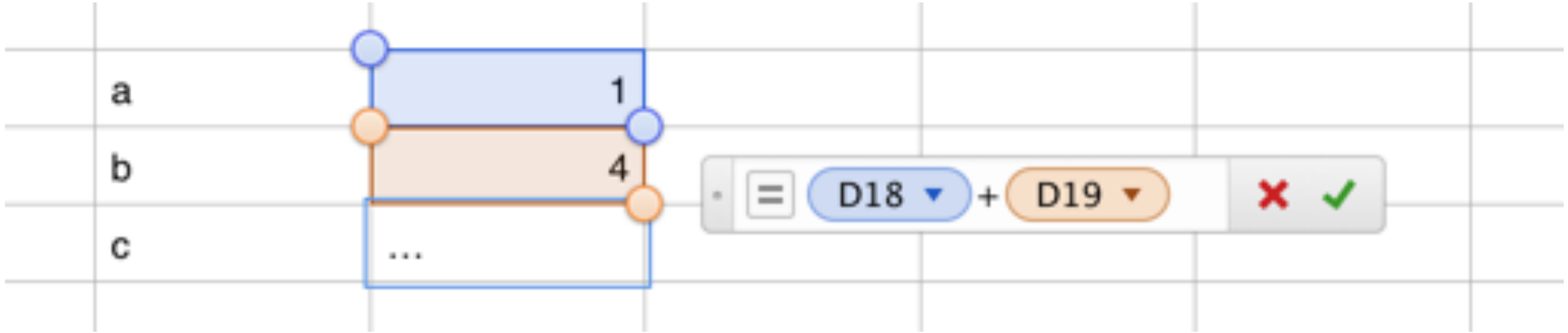
Long Story Short

my name is vincent
i do open source data
ask me anything
today i'll teach you R

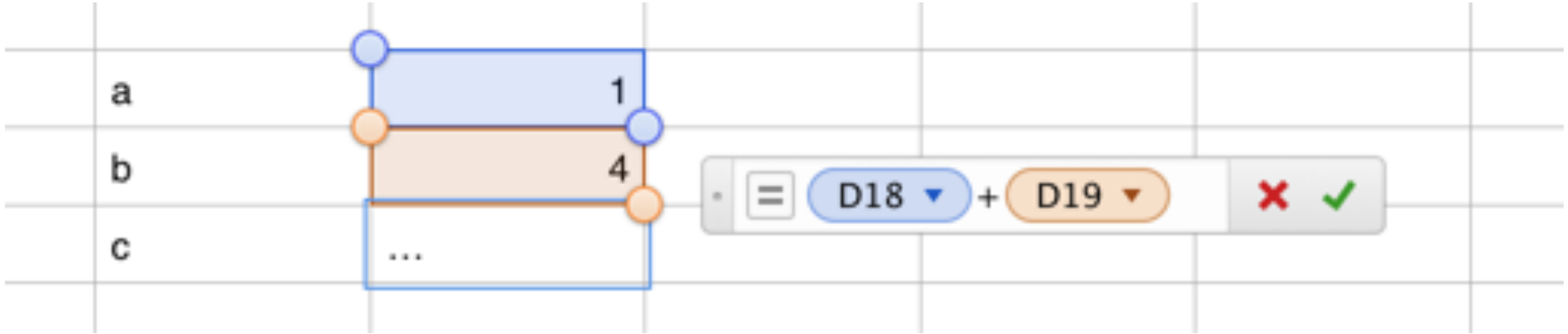
Today

- The morning will be a dry intro to programming.
- The afternoon will be fun!
- ... with visualisation
- ... with data wrangling
- ... with reproducible results
- ... with a borrel at the end

Excel Workflow



Excel Workflow



In R:

```
c <- a + b
```

Excel Workflow

Excel Workflow

In R:

```
c <- max(a, b)
```

Let's now make some code

I'll demonstrate code, you guys try to repeat what I am doing.

A Use Case

Farmer Fred

We're going to do an ABCD test based on different diets for different chickens.

We'll pretend to be consultants, except that we're obviously much than most McKinsey people at what we do because we can actually code.

We'll use the dataset `ChickWeight`.

We have data.

Now what?

We have data.

Why not just look at it?

Guess.

What does this code do?

```
> head(ChickWeight)
  weight Time Chick Diet
1     42    0     1    1
2     51    2     1    1
3     59    4     1    1
4     64    6     1    1
5     76    8     1    1
6     93   10     1    1
```

Guess.

What does this code do?

```
> summary(ChickWeight)
```

Guess.

What does this code do?

```
> summary(ChickWeight)
     weight           Time           Chick     Diet   
 Min.   : 35.0   Min.   : 0.00   13     : 12   1:220  
 1st Qu.: 63.0   1st Qu.: 4.00   9      : 12   2:120  
 Median :103.0   Median :10.00   20     : 12   3:120  
 Mean   :121.8   Mean   :10.72   10     : 12   4:118  
 3rd Qu.:163.8   3rd Qu.:16.00   17     : 12          
 Max.   :373.0   Max.   :21.00   19     : 12          
                                 (Other):506          
```

Feeling.

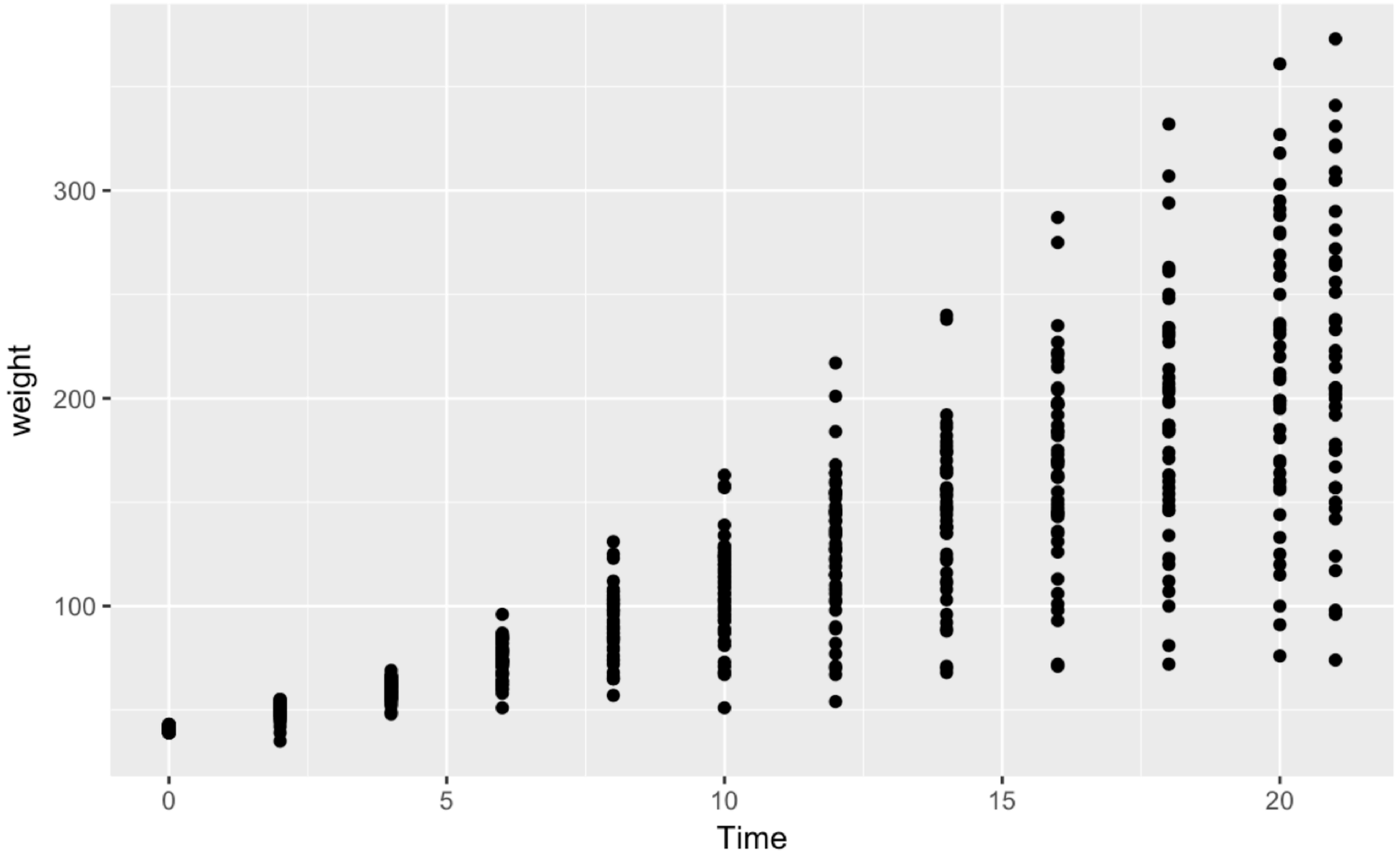
We now know what we can expect from the data.

Let's now actually look at it by visualising it.

Guess

What does this code do?

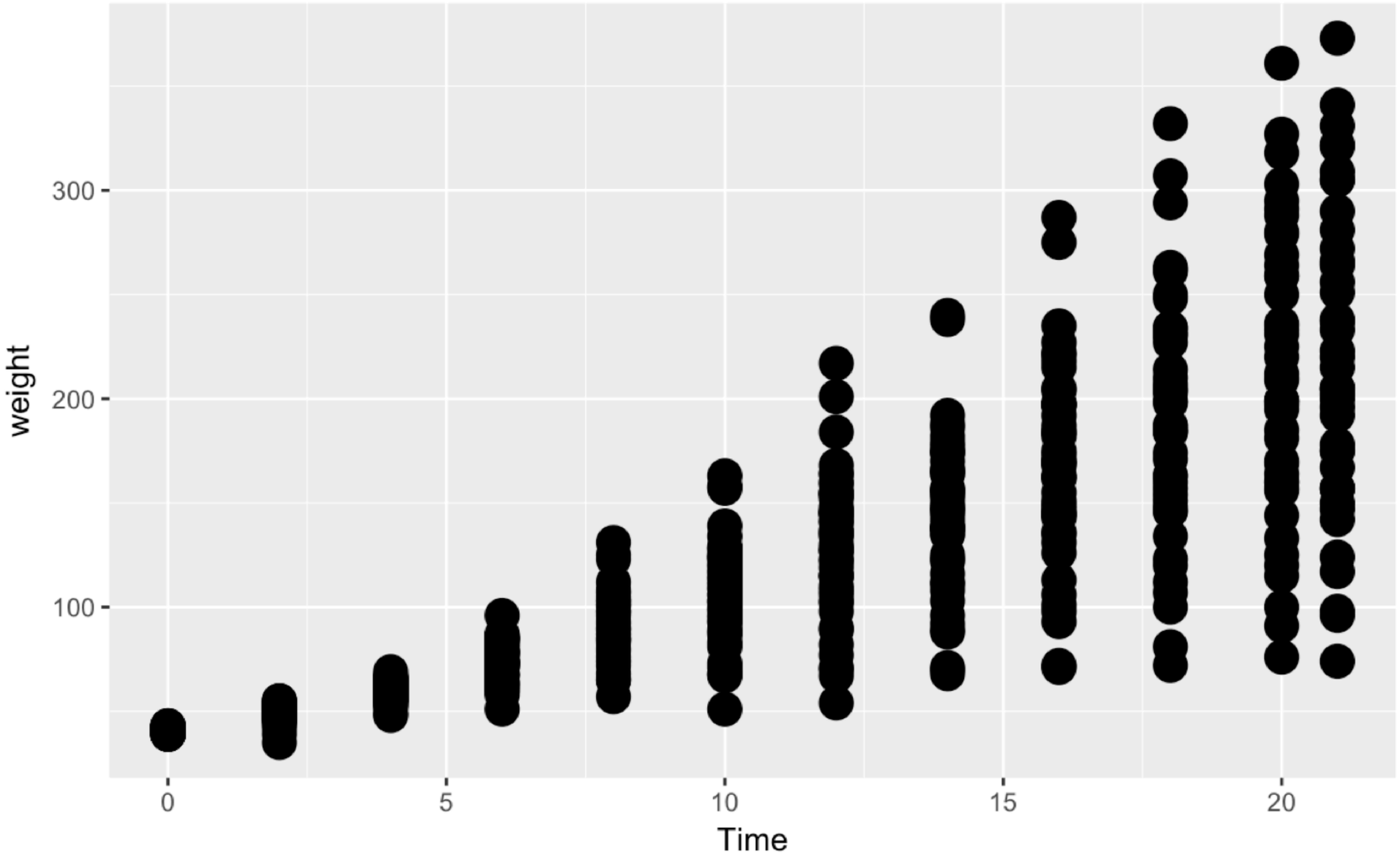
```
p <- ggplot()
p + geom_point(
  data=ChickWeight,
  aes(x=Time, y=weight)
)
```


weight
300
200
100
0
5
10
15
20
Time

Guess

What does this code do?

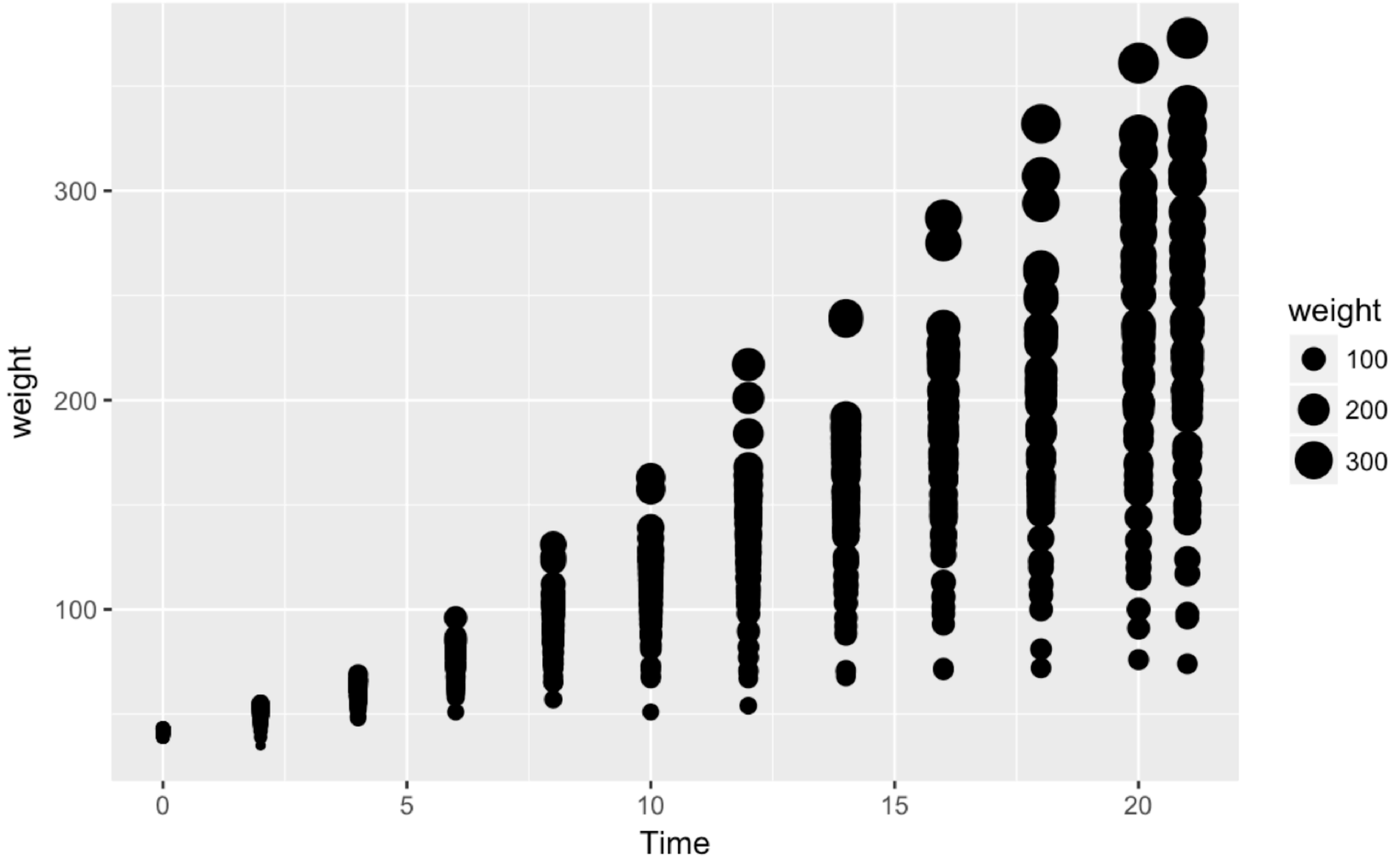
```
p <- ggplot()
p + geom_point(
  data=ChickWeight,
  aes(x=Time, y=weight),
  size = 5
)
```


weight
300
200
100
0
5
10
15
20
Time

Guess

What does this code do?

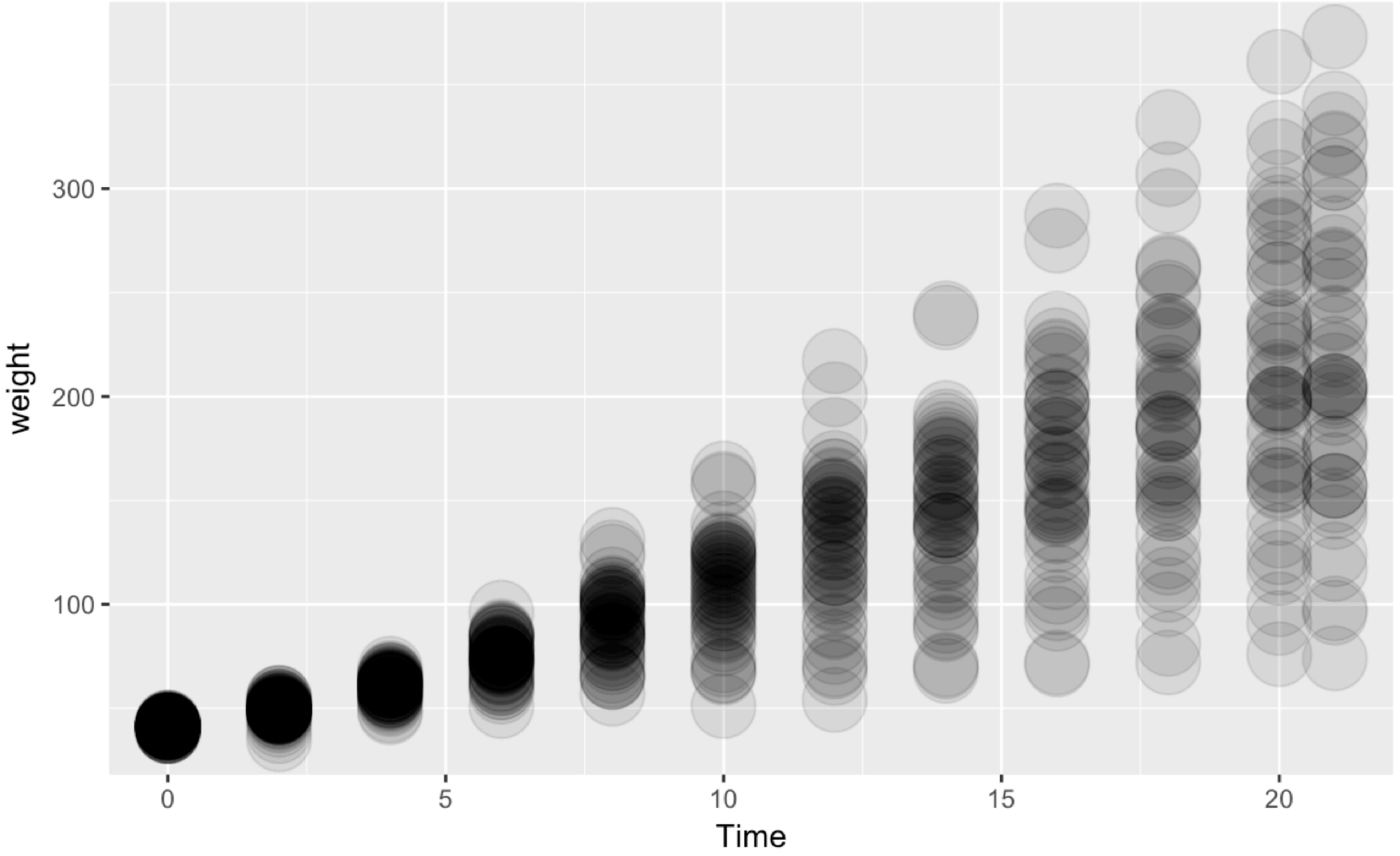
```
p <- ggplot()
p + geom_point(
  data=ChickWeight,
  aes(x=Time, y=weight, size = weight)
)
```


300
200
100
weight
0
5
10
15
20
Time
weight
100
200
300

Guess

What does this code do?

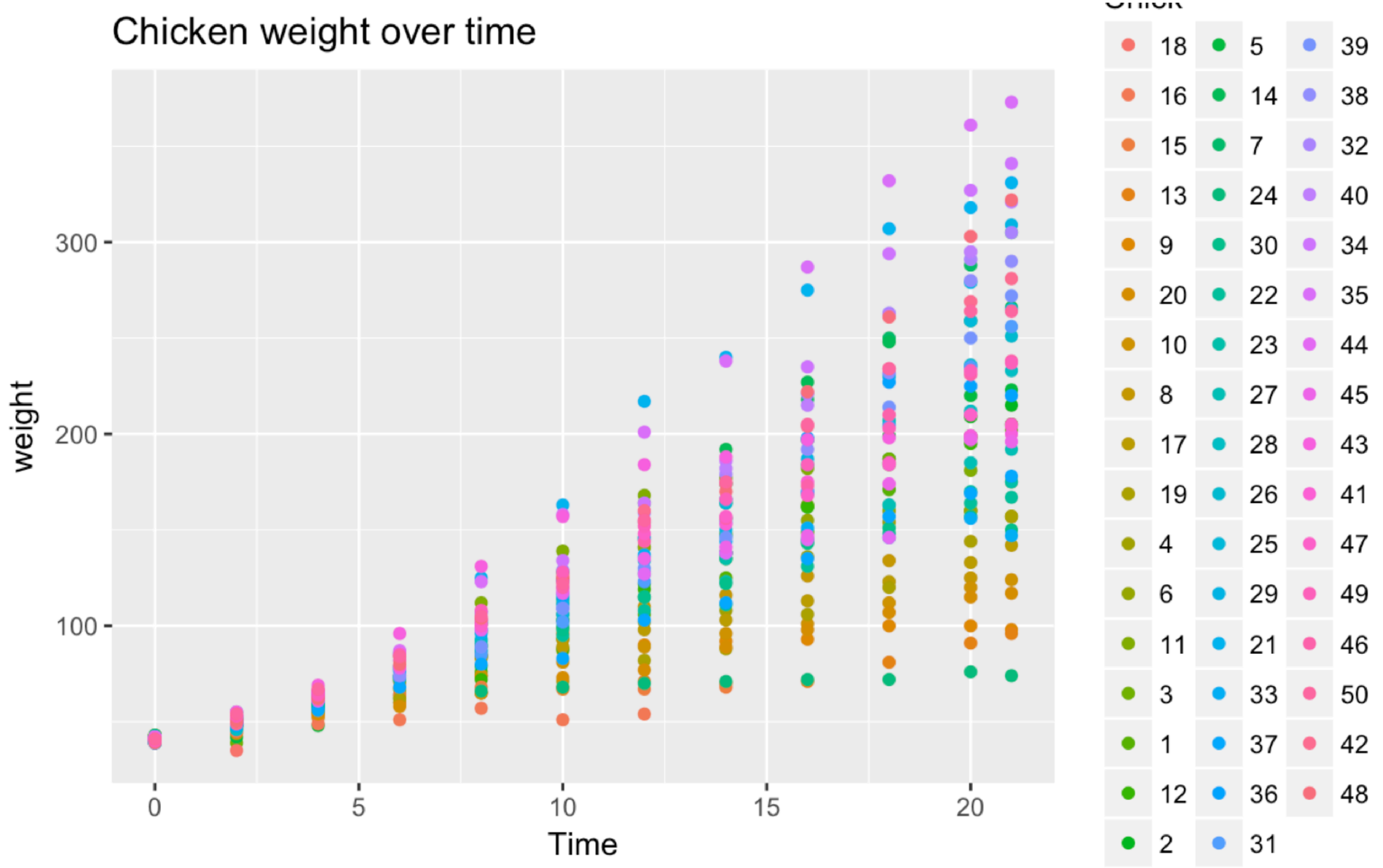
```
p <- ggplot()
p + geom_point(
  data=ChickWeight,
  aes(x=Time, y=weight),
  size = 10, alpha = 0.1
)
```


weight
300
200
100
0
5
10
15
20
Time

Guess

What does this code do?

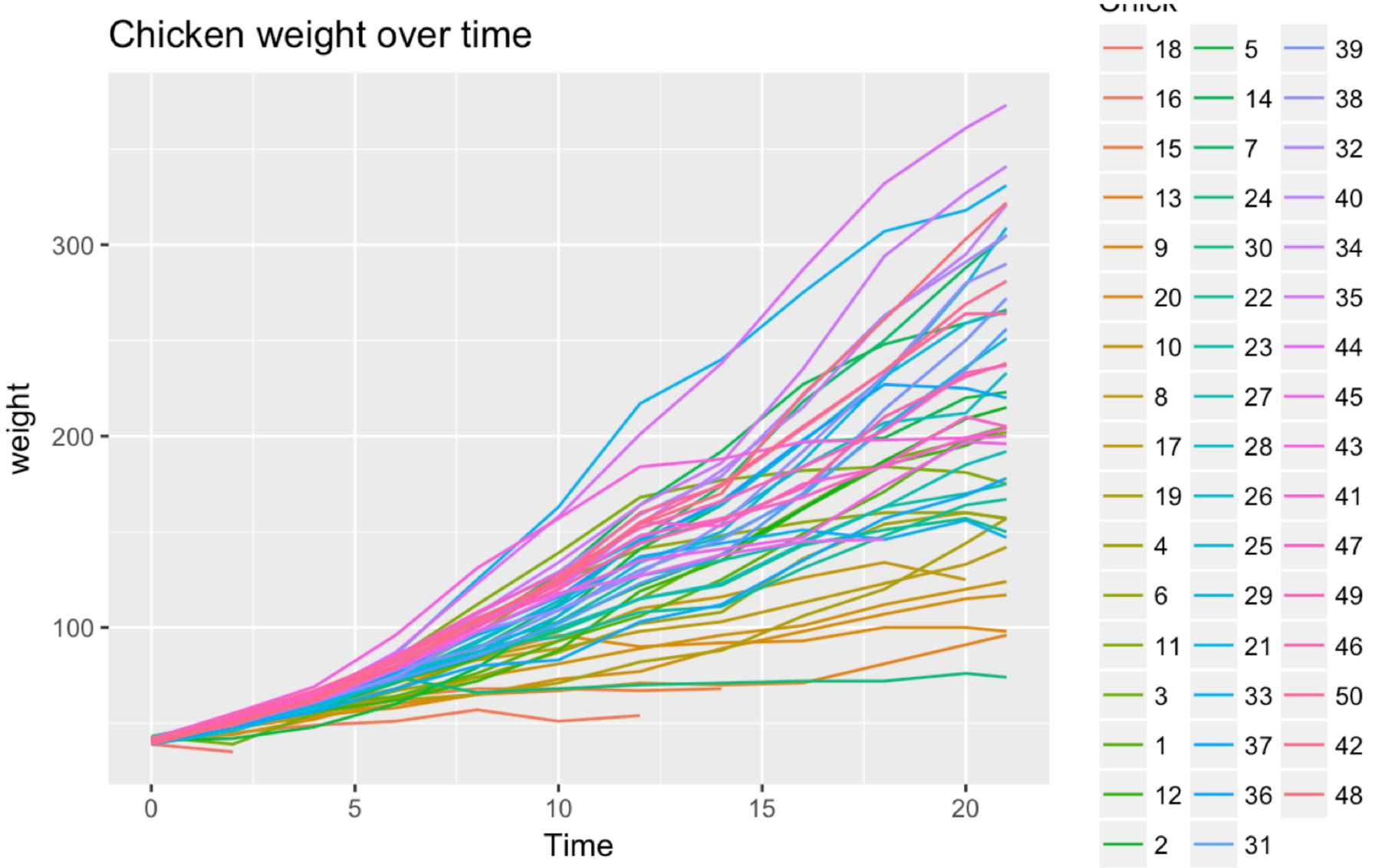
```
p <- ggplot()
p <- p + geom_point(
  data=ChickWeight,
  aes(x=Time, y=weight, colour=Chick)
)
p + ggtitle("Chicken weight over time")
```


Chicken weight over time
weight
300
200
100
0
5
10
15
20
Time
18
5
39
16
14
38
15
7
32
13
24
40
9
30
34
20
22
35
10
23
44
8
27
45
17
28
43
19
26
41
4
25
47
6
29
49
11
21
46
3
33
50
1
37
42
12
36
48
2
31

Guess

What does this code do?

```
p <- ggplot()
p <- p + geom_line(
  data=ChickWeight,
  aes(x=Time, y=weight, colour=Chick)
)
p + ggtitle("Chicken weight over time")
```


Chicken weight over time
weight
300
200
100
0
5
10
15
20
Time
Chick
18
16
15
13
9
20
10
8
17
19
4
6
11
3
1
12
2
5
14
7
24
30
22
23
27
28
26
25
29
21
33
37
36
31
39
38
32
40
34
35
44
45
43
41
47
49
46
50
42
48

Guess

What does this code do?

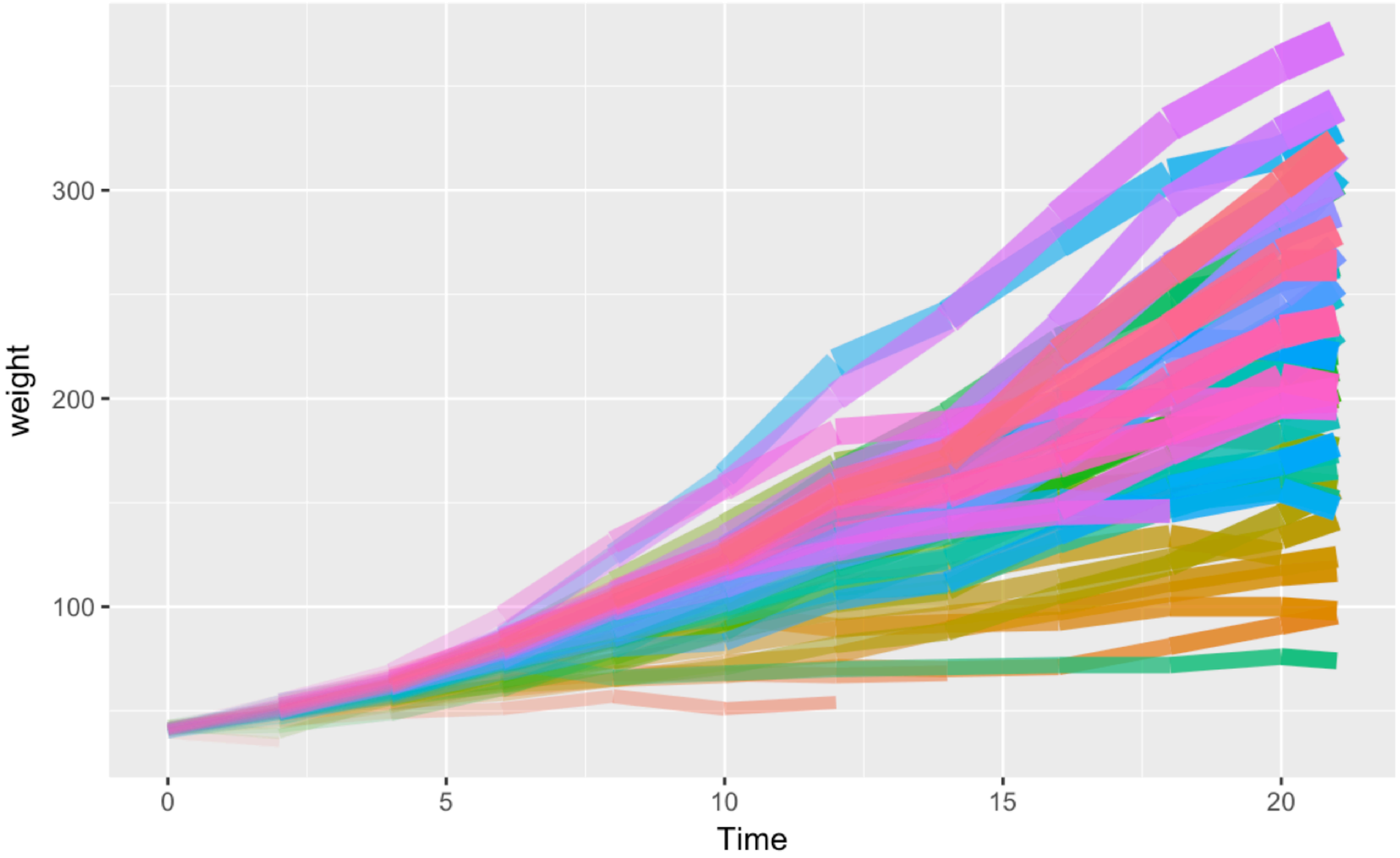
```
p <- ggplot() +
    geom_line(
        data=ChickWeight,
        aes(x=Time, y=weight, colour=Chick)) +
        theme(legend.position="none")

p + ggtitle("Chicken weight over time",
    subtitle = "Note! Some chickens seem to die prematurely!")
```


Chicken weight over time
Note! Each line represents a chicken, some chickens seem to die prematurely.
weight
300
200
100
0
5
10
15
20
Time

Guess

What does this code do?

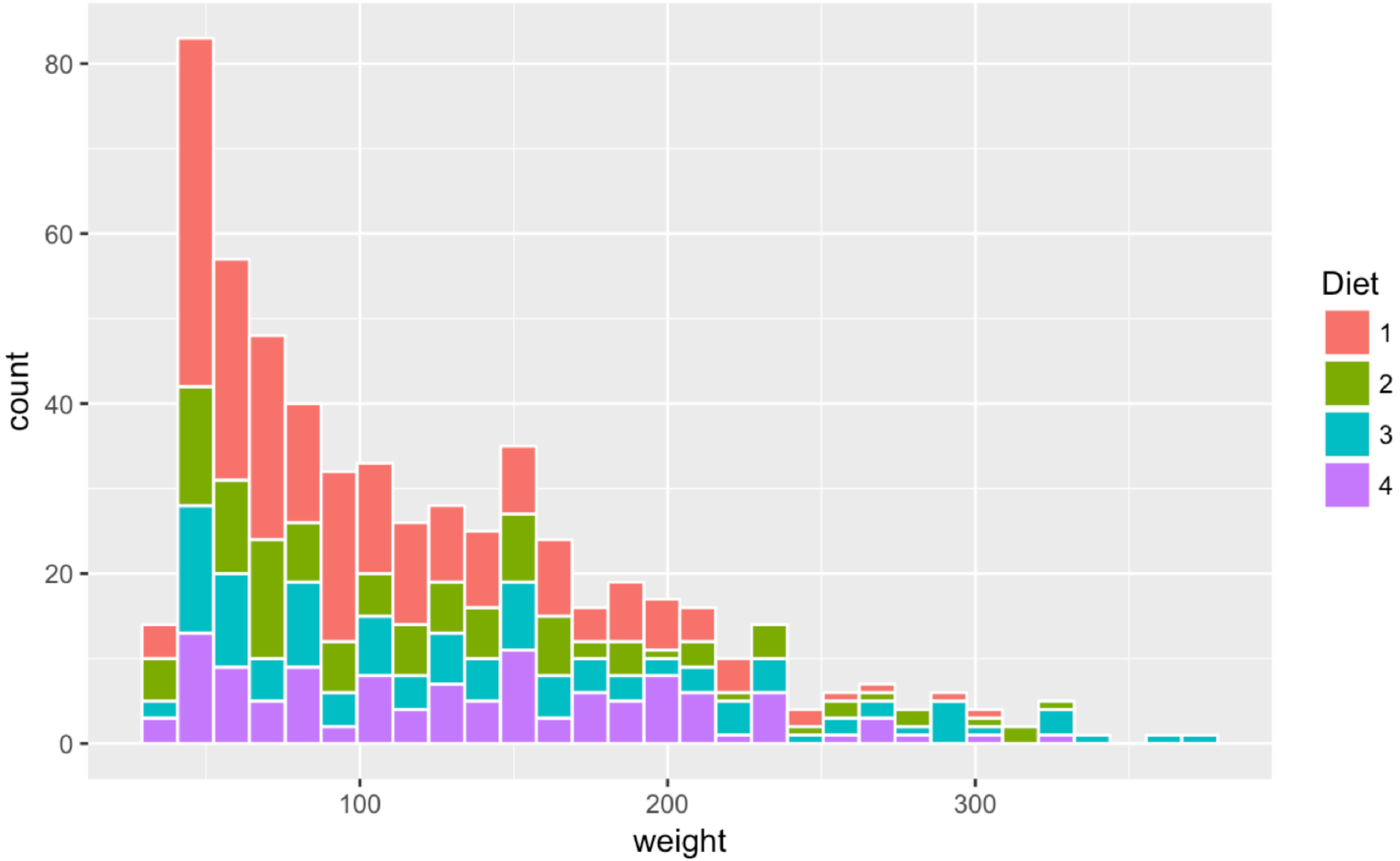
```
ggplot() +
  geom_line(data=ChickWeight,
            aes(x=Time, y=weight,
                colour=Chick, alpha=Time,
                size=weight)) +
  theme(legend.position="none")
```


weight
300
200
100
0
5
10
15
20
Time

Guess

What does this code do?

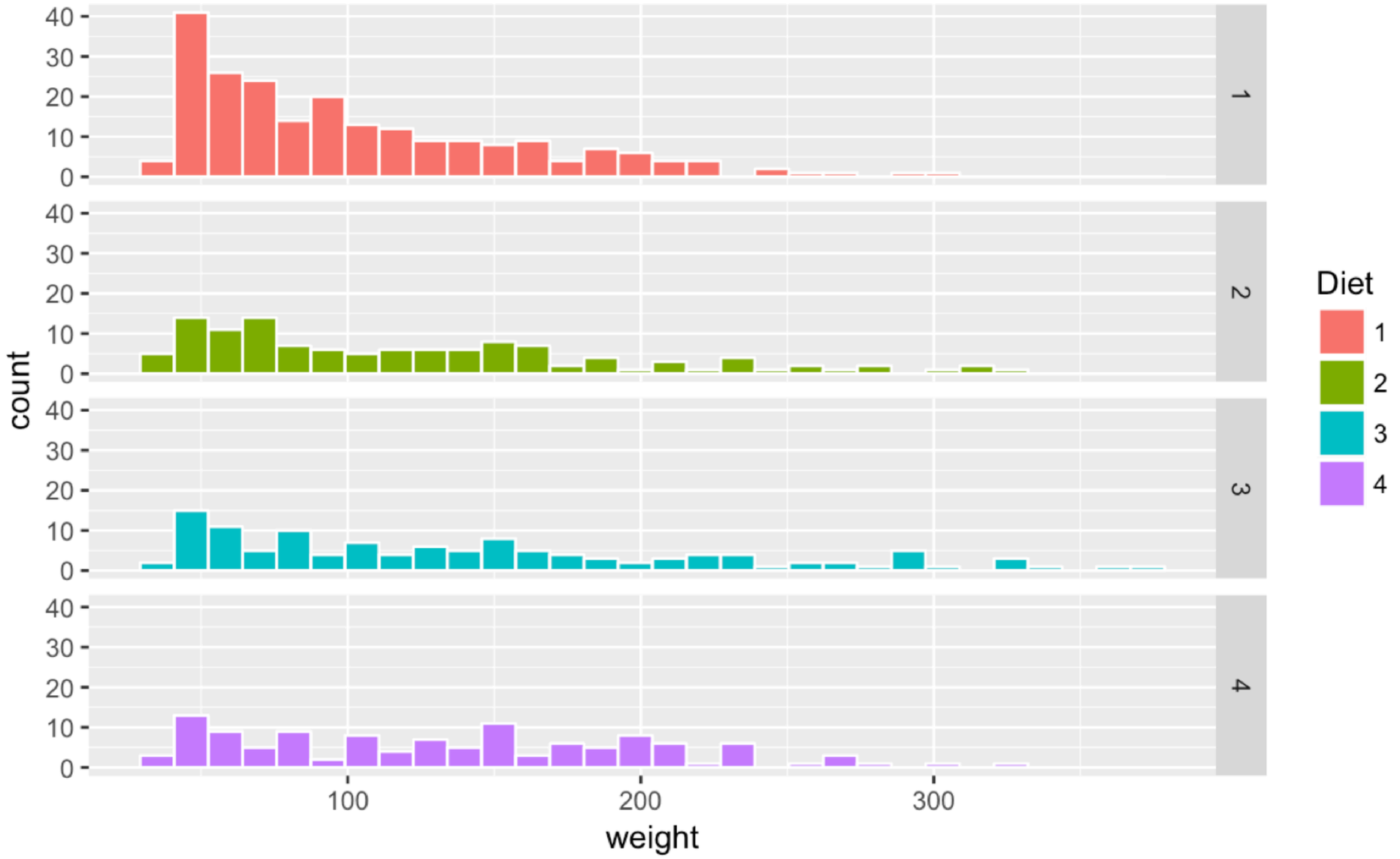
```
p <- ggplot()
p + geom_histogram(
  data=ChickWeight,
  aes(x=weight, fill=Diet),
  colour="white"
)
```


80
60
40
20
0
count
100
200
300
weight
Diet
1
2
3
4

Guess

What does this code do?

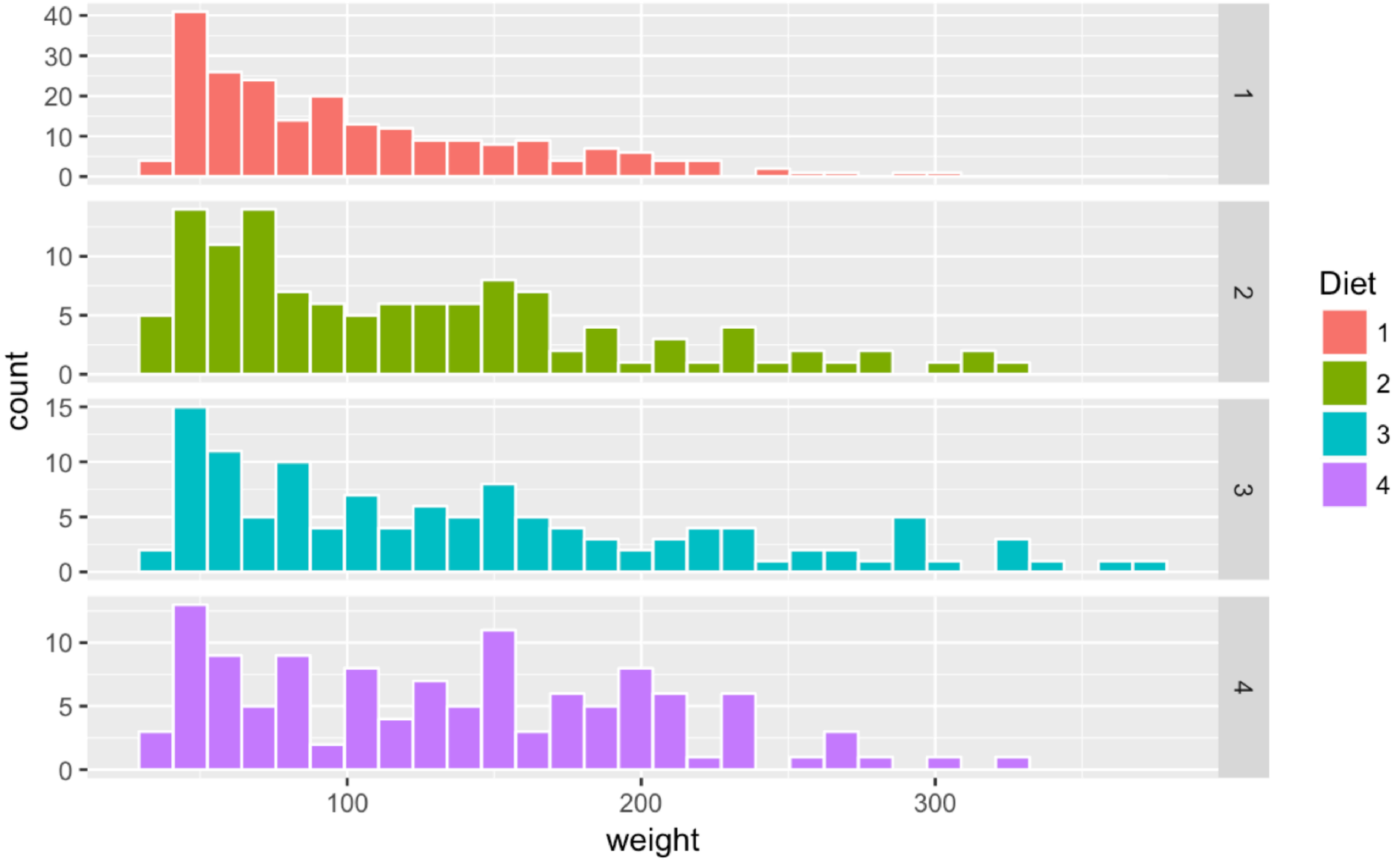
```
p <- ggplot()
p <- p + geom_histogram(
  data=ChickWeight,
  aes(x=weight, fill=Diet),
  colour="white"
)
p + facet_grid(Diet ~ .)
```


count
40
30
20
10
0
1
2
3
4
100
200
300
weight
Diet
1
2
3
4

Guess

What does this code do?

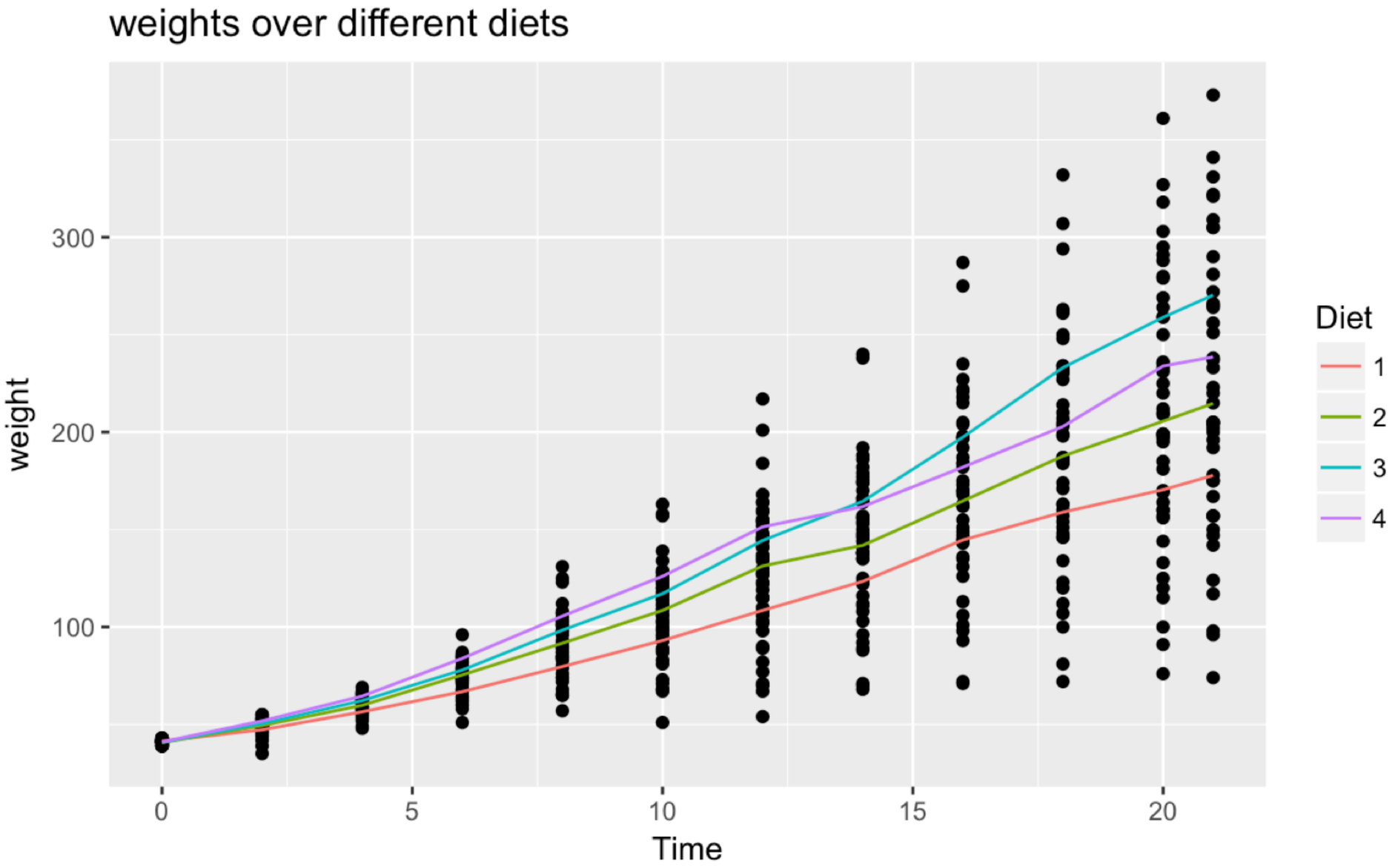
```
p <- ggplot()
p <- p + geom_histogram(
  data=ChickWeight,
  aes(x=weight, fill=Diet),
  colour="white"
)
p + facet_grid(Diet ~ ., scales = "free_y")
```


count
weight
Diet
1
2
3
4
40
30
20
10
0
15
5
100
200
300

Plan

Guess what does this code do?

```
agg <- ChickWeight %>%
  group_by(Diet, Time) %>%
  summarise(m = mean(weight))

ggplot() +
  geom_point(data=ChickWeight, aes(Time, weight)) +
  geom_line(data=agg, aes(Time, m, colour=Diet)) +
  ggtitle("weights over different diets")
```


weights over different diets
weight
300
200
100
0
5
10
15
20
Time
Diet
1
2
3
4

These Dead Chickens
Can you try to find them?

Vincent D. Warmerdam - @fishnets88 - koaning.io - GoDataDriven

Verbs

Just like **ggplot** offers verbs for plotting **dplyr** gives us verbs for data wrangling. We'll demonstrate a few examples.

Guess

What does this code do?

```
ChickWeight %>%
    select(weight, Time) %>%
    head()
```

	weight	Time
1	42	0
2	51	2
3	59	4
4	64	6
5	76	8
6	93	10

Guess

What does this code do?

```
ChickWeight %>%
    select(weight, Time) %>%
    filter(weight < 50) %>%
    head(10)
```

	weight	Time
1	42	0
2	40	0
3	49	2
4	43	0
5	39	2
6	42	0
7	49	2
8	41	0
9	42	2
10	48	4

Guess

What does this code do?

```
ChickWeight %>%
    select(-Chick) %>%
    filter(weight < 40) %>%
    head()
```

	weight	Time	Diet
1	39	2	1
2	39	0	1
3	35	2	1
4	39	0	2
5	39	0	2
6	39	0	2

Guess

What does this code do?

```
ChickWeight %>%
    select(-Chick) %>%
    filter(weight > 50) %>%
    filter(weight < 100) %>%
    head()
```

	weight	Time	Diet
1	51	2	1
2	59	4	1
3	64	6	1
4	76	8	1
5	93	10	1
6	58	4	1

Guess

What does this code do?

```
ChickWeight %>%
    select(-Chick) %>%
    filter(weight > 50, weight < 100) %>%
    head()
```

	weight	Time	Diet
1	51	2	1
2	59	4	1
3	64	6	1
4	76	8	1
5	93	10	1
6	58	4	1

Guess

What does this code do?

```
ChickWeight %>%
    filter(Time < 12) %>%
    summary()
```

```
     weight           Time              Chick     Diet
 Min.   : 35.00   Min.   : 0.000   16     :  6   1:116
 1st Qu.: 49.00   1st Qu.: 2.000   15     :  6   2: 60
 Median : 63.50   Median : 4.000   13     :  6   3: 60
 Mean   : 70.43   Mean   : 4.973   9      :  6   4: 60
 3rd Qu.: 86.00   3rd Qu.: 8.000   20     :  6
 Max.   :163.00   Max.   :10.000   10     :  6
                                   (Other):260
```

Guess

What does this code do?

```
ChickWeight %>%
    arrange(weight) %>%
    head()
```

	weight	Time	Chick	Diet
1	35	2	18	1
2	39	2	3	1
3	39	0	18	1
4	39	0	27	2
5	39	0	28	2
6	39	0	29	2

Guess

What does this code do?

```
ChickWeight %>%
    head() %>%
    arrange(-weight)
```

	weight	Time	Chick	Diet
1	93	10	1	1
2	76	8	1	1
3	64	6	1	1
4	59	4	1	1
5	51	2	1	1
6	42	0	1	1

Guess

What does this code do?

```
ChickWeight %>%
    summarise(number_rows = n())
```

```
     n
1  578
```

Guess

What does this code do?

```
ChickWeight %>%
    group_by(Time) %>%
    summarise(number_rows = n())
```

	Time	n
	<dbl>	<int>
1	0	50
2	2	50
3	4	49
4	6	49
5	8	49
6	10	49
7	12	49
8	14	48
9	16	47
10	18	47
11	20	46
12	21	45

Guess

What does this code do?

```
ChickWeight %>%
    group_by(Time) %>%
    summarise(number_rows = n(), m = mean(weight), v = var(weight))
```

	Time	n	m	v
	<dbl>	<int>	<dbl>	<dbl>
1	0	50	41.06000	1.282041
2	2	50	49.22000	13.603673
3	4	49	59.95918	20.206633
4	6	49	74.30612	81.216837
5	8	49	91.24490	263.730442
6	10	49	107.83673	575.389456
7	12	49	129.24490	1164.147109
8	14	48	143.81250	1466.921543
9	16	47	168.08511	2199.992599
10	18	47	190.19149	3294.158187
11	20	46	209.71739	4423.807246
12	21	45	218.68889	5113.719192

Guess

What does this code do?

```
ChickWeight %>%
    mutate(weight2 = weight * 2) %>%
    head()
```

	weight	Time	Chick	Diet	weight2
1	42	0	1	1	84
2	51	2	1	1	102
3	59	4	1	1	118
4	64	6	1	1	128
5	76	8	1	1	152
6	93	10	1	1	186

Guess

What does this code do?

```
ChickWeight %>%
    mutate(weight = weight * 2) %>%
    head()
```

	weight	Time	Chick	Diet
1	84	0	1	1
2	102	2	1	1
3	118	4	1	1
4	128	6	1	1
5	152	8	1	1
6	186	10	1	1

Guess

What does this code do?

```
ChickWeight %>%
    group_by(Chick) %>%
    mutate(r = row_number()) %>%
    ungroup() %>%
    filter(r < 5) %>%
    select(Time, Chick, r) %>%
    head(11)
```

	Time	Chick	r
	<dbl>	<ord>	<int>
1	0	1	1
2	2	1	2
3	4	1	3
4	6	1	4
5	0	2	1
6	2	2	2
7	4	2	3
8	6	2	4
9	0	3	1
10	2	3	2
11	4	3	3

Guess

What does this code do?

```
ChickWeight %>%
    select(Chick, Diet) %>%
    distinct()
```

	Chick	Diet
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1
11	11	1
12	12	1
13	13	1
14	14	1
15	15	1

...

Guess

What does this code do?

```
ChickWeight %>%
    sample_n(3)
```

```
> ChickWeight %>% sample_n(3)
    weight Time Chick Diet
524    120   10    46    4
478    126   10    42    4
506    146   18    44    4

> ChickWeight %>% sample_n(3)
    weight Time Chick Diet
573    155   12    50    4
388    341   21    34    3
343     62    4    31    3
```

Guess

What does this code do?

```
ChickWeight %>% View()
```

Assignment!
Find the fattest chicken.

Hint; use `summarise` and `group_by`.

Assignment!
Find the dead chickens.

Hint; use `summarise` and `group_by`.

Assignment!

When do chickens grow the most?

Hint; google the `lag` function from dplyr.

Modelling

Let's make a simple regression model. You can do that via;

```
mod <- lm(weight ~ Time + Diet, data=ChickWeight)
summary(mod)
```

or

```
lm(weight ~ Time + Diet, data=ChickWeight) %>%
    summary()
```

Markdown Demo

Let's now combine everything into a nice document.

A document we can share around or even host on a website, like an internal blog.

Shiny Demo