Lesson 3:
Logistic Regression

This Lesson’s Goals

Learn about logistic regression

Make a figure for data from a logistic regression

Do a logistic regression in R

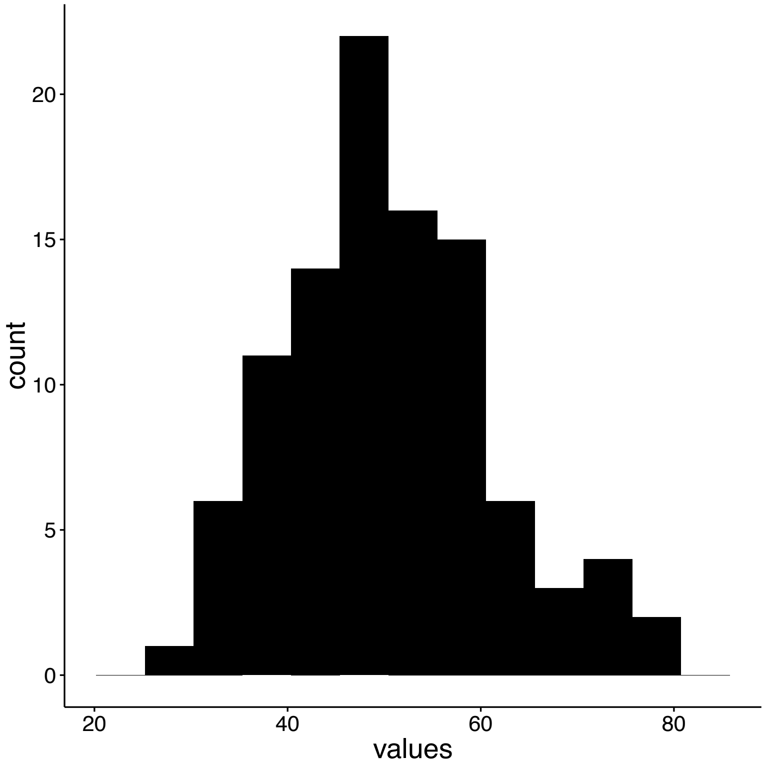
Summarise results in an R Markdown document

Math

linear regression

predict
continuous variables

talk about in regards to
mean and standard deviation

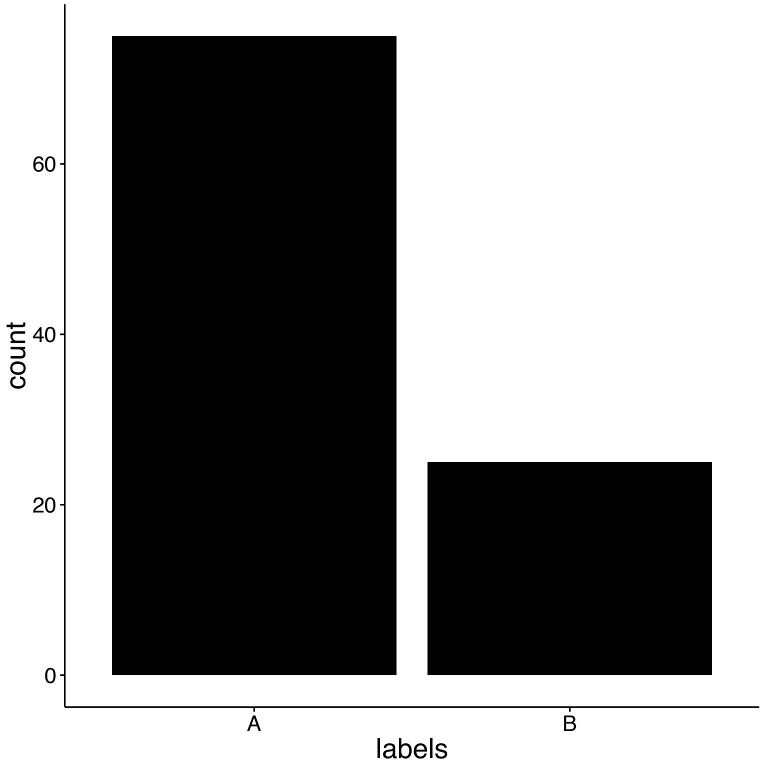


predict specific y-value
given specific x-value

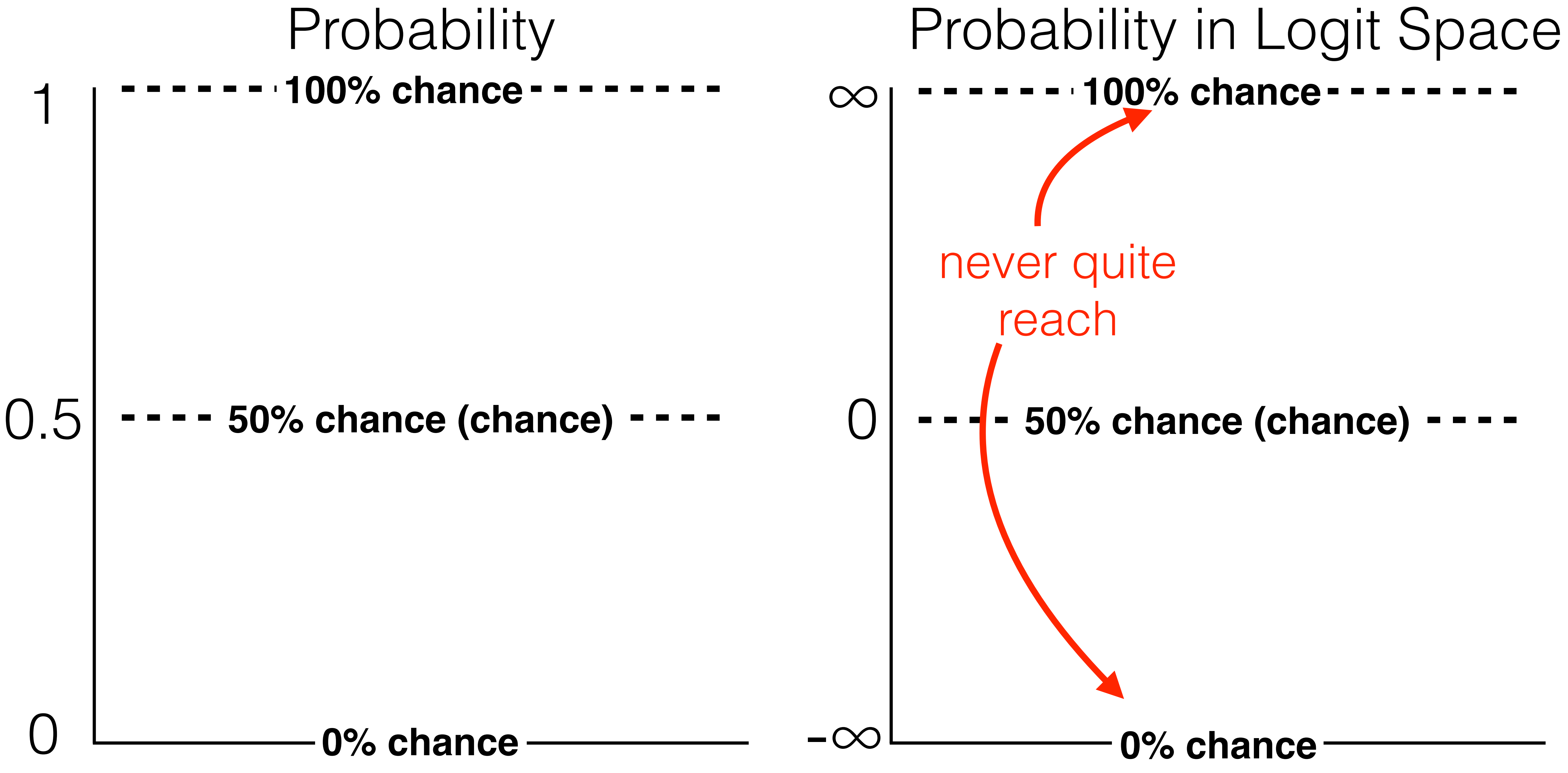
logistic regression

predict
categorical variables

talk about in regards to
counts



predict ***probability*** y-***level***
given specific x-value



Probability

1

0.5

0

100% chance

50% chance (chance)

0% chance

Probability in Logit Space

∞

0

-∞

100% chance

never quite reach

50% chance (chance)

0% chance

$$y_i = a + bx_i + e_i$$

$$\text{logit } p_i = a + bx_i$$

$$\boxed{\log[p/(1\text{-}p)]_i} = a + bx_i?$$

log odds

no error term

$$\boxed{log[p/(1-p)]_i} = \boxed{a} + \boxed{b}\boxed{x_i}$$

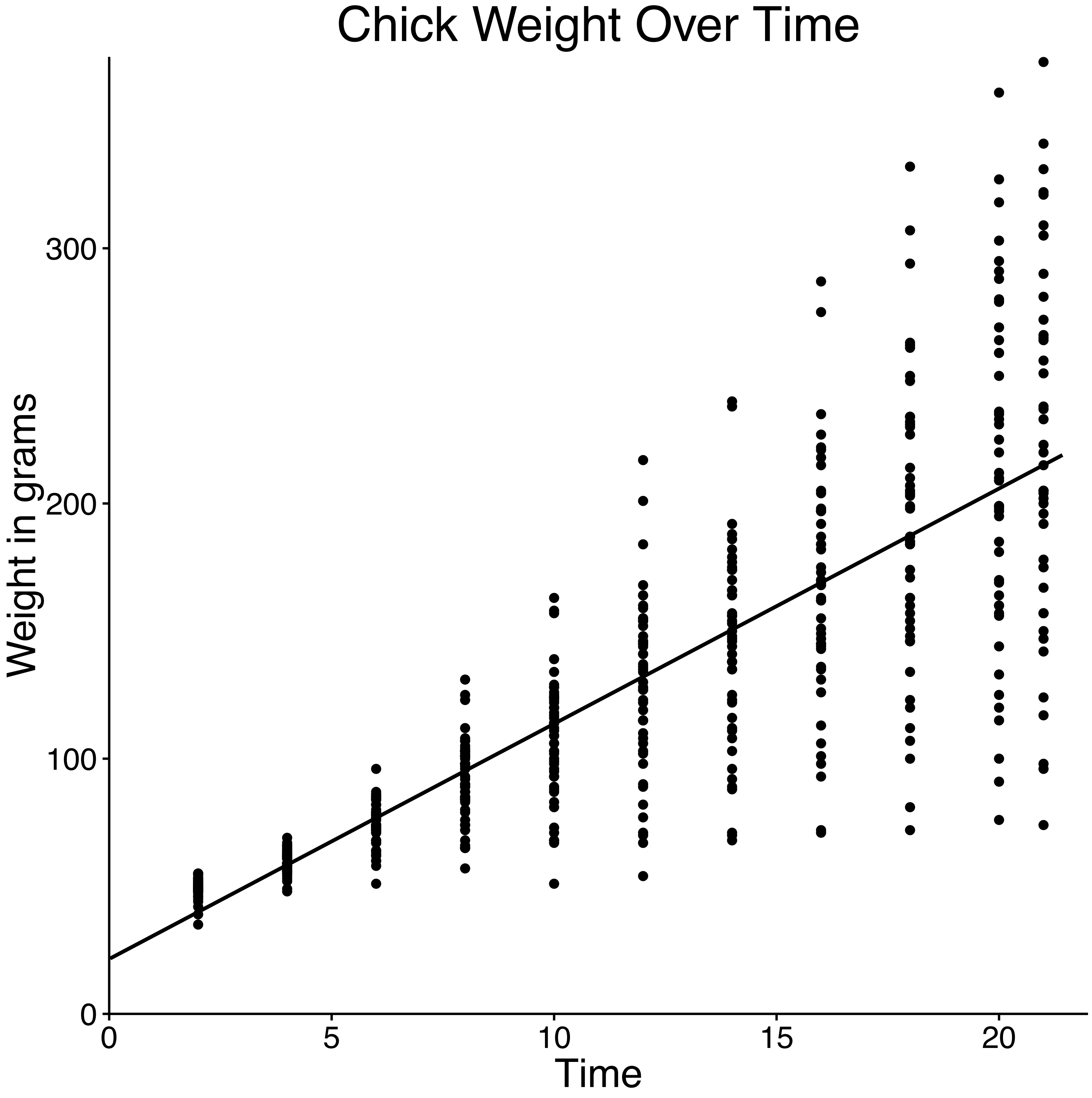
$log[p/(1-p)]_i$ = probability of specific y-level (F or T)
(dependent variable)

a = intercept

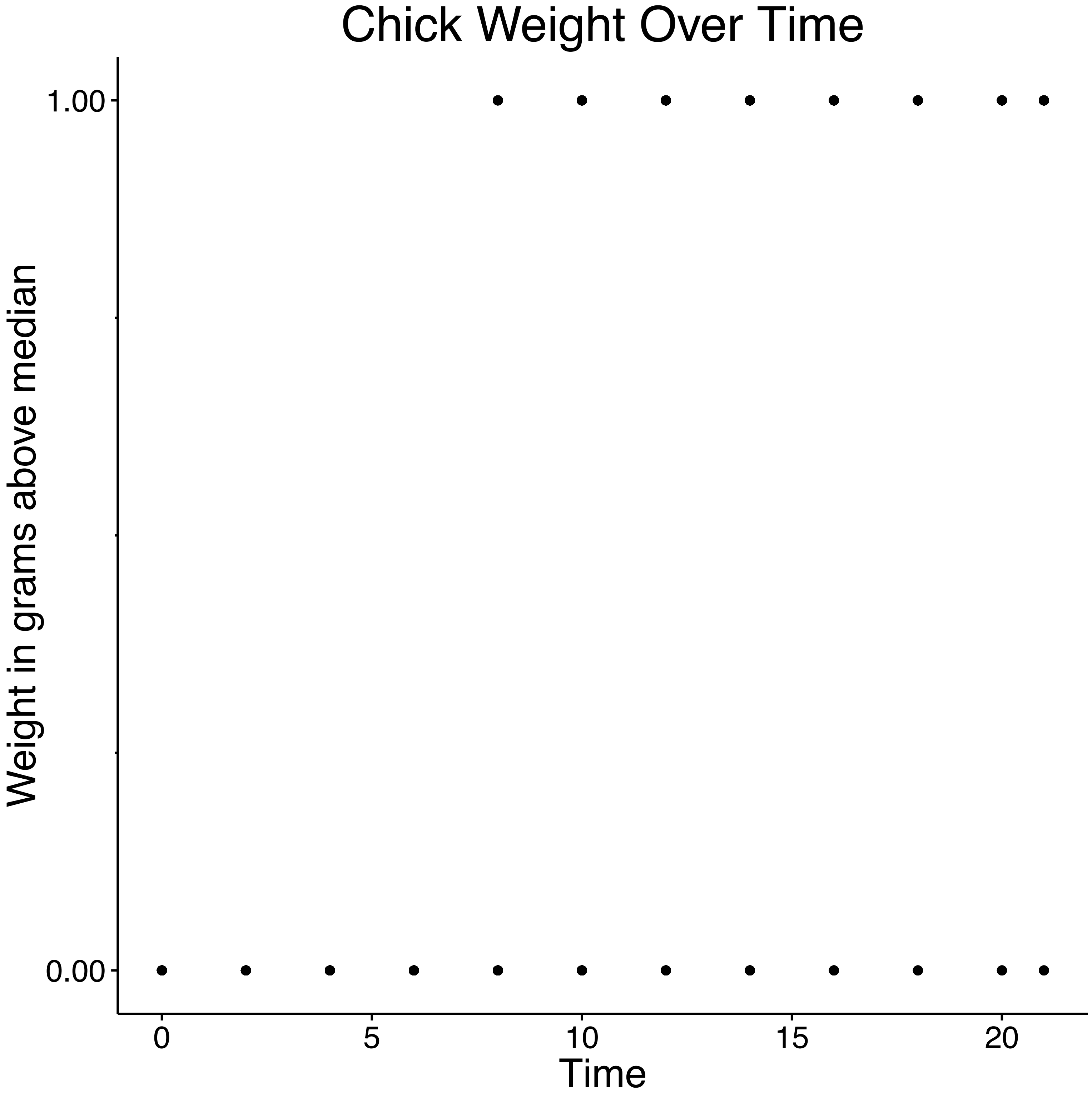
b = slope

x_i = specific x-values (independent variable)

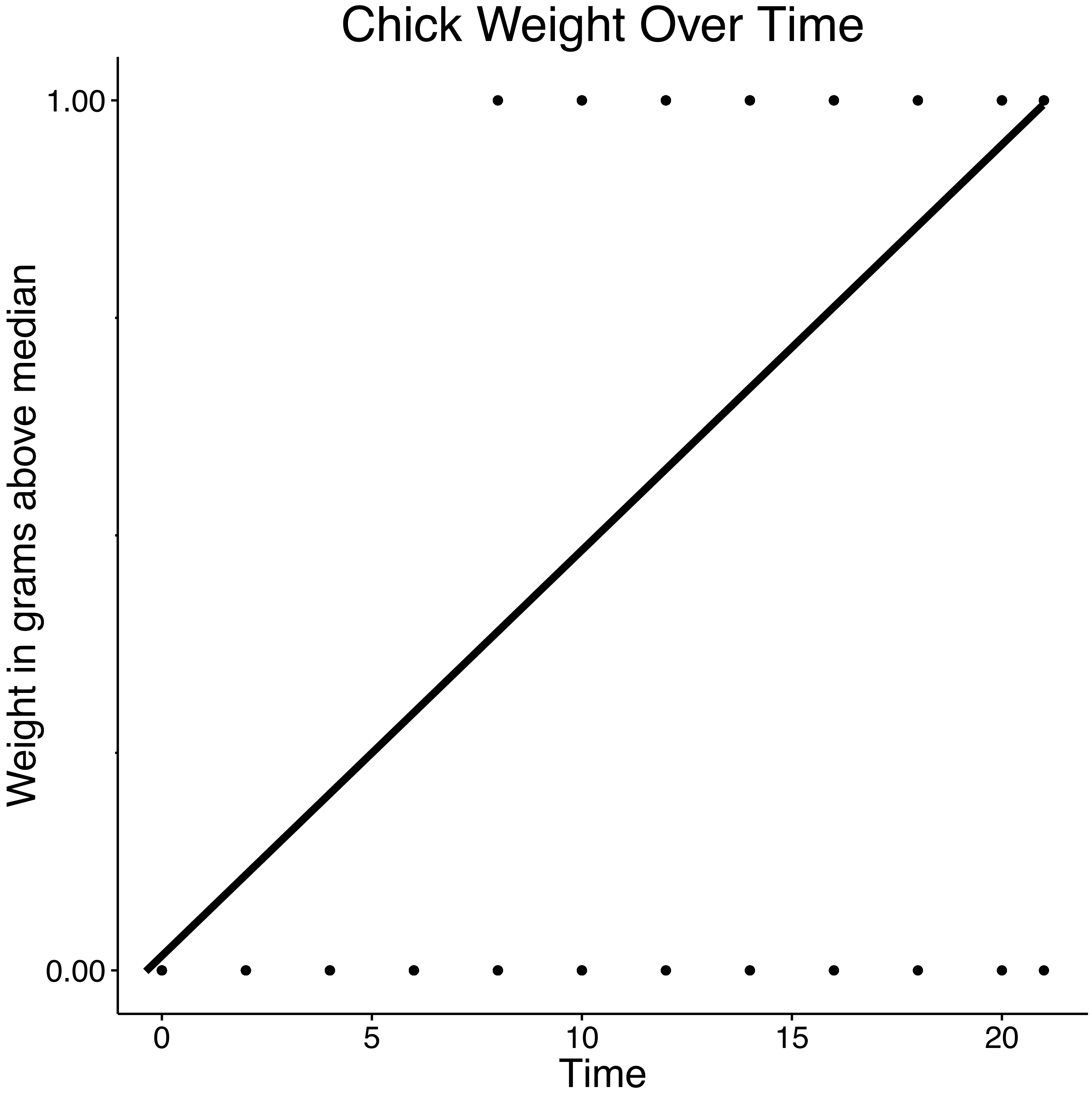
$$y_i = a + bx_i + e_i$$


Chick Weight Over Time
Weight in grams
Time
0
100
200
300
0
5
10
15
20

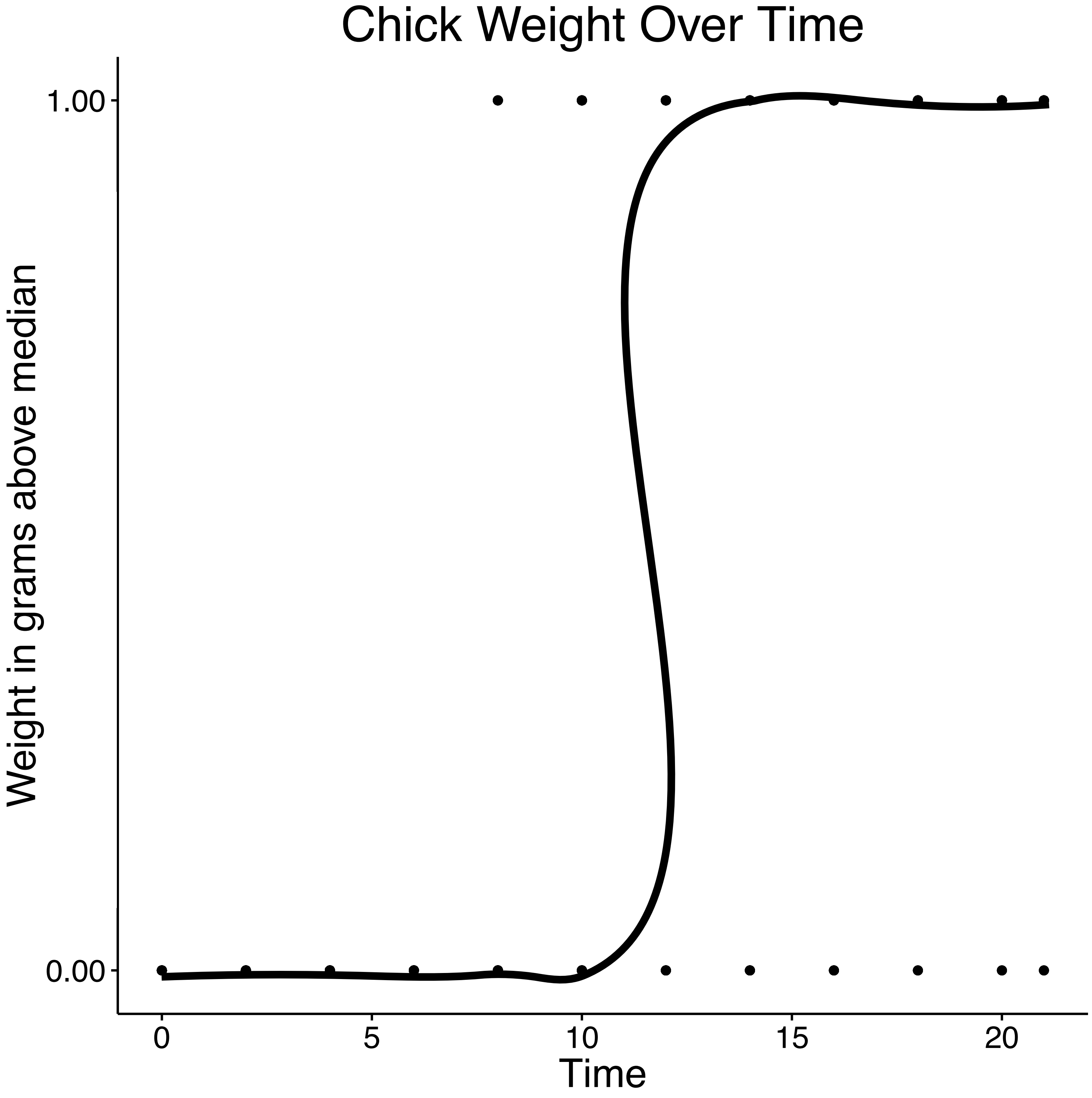
$$\log[p/(1-p)]_i = a + bx_i$$



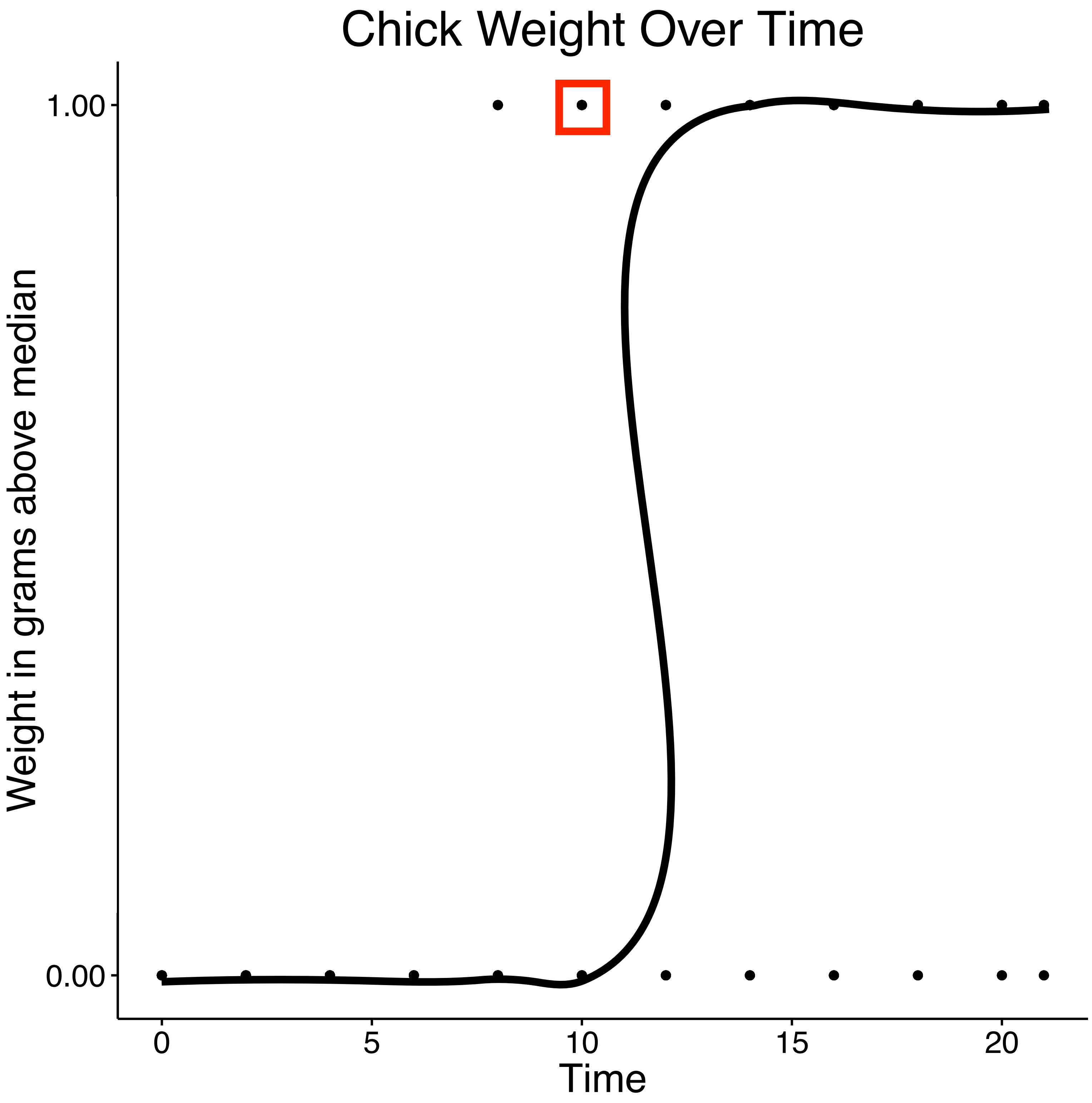
$$\log[p/(1-p)]_i = a + bx_i$$



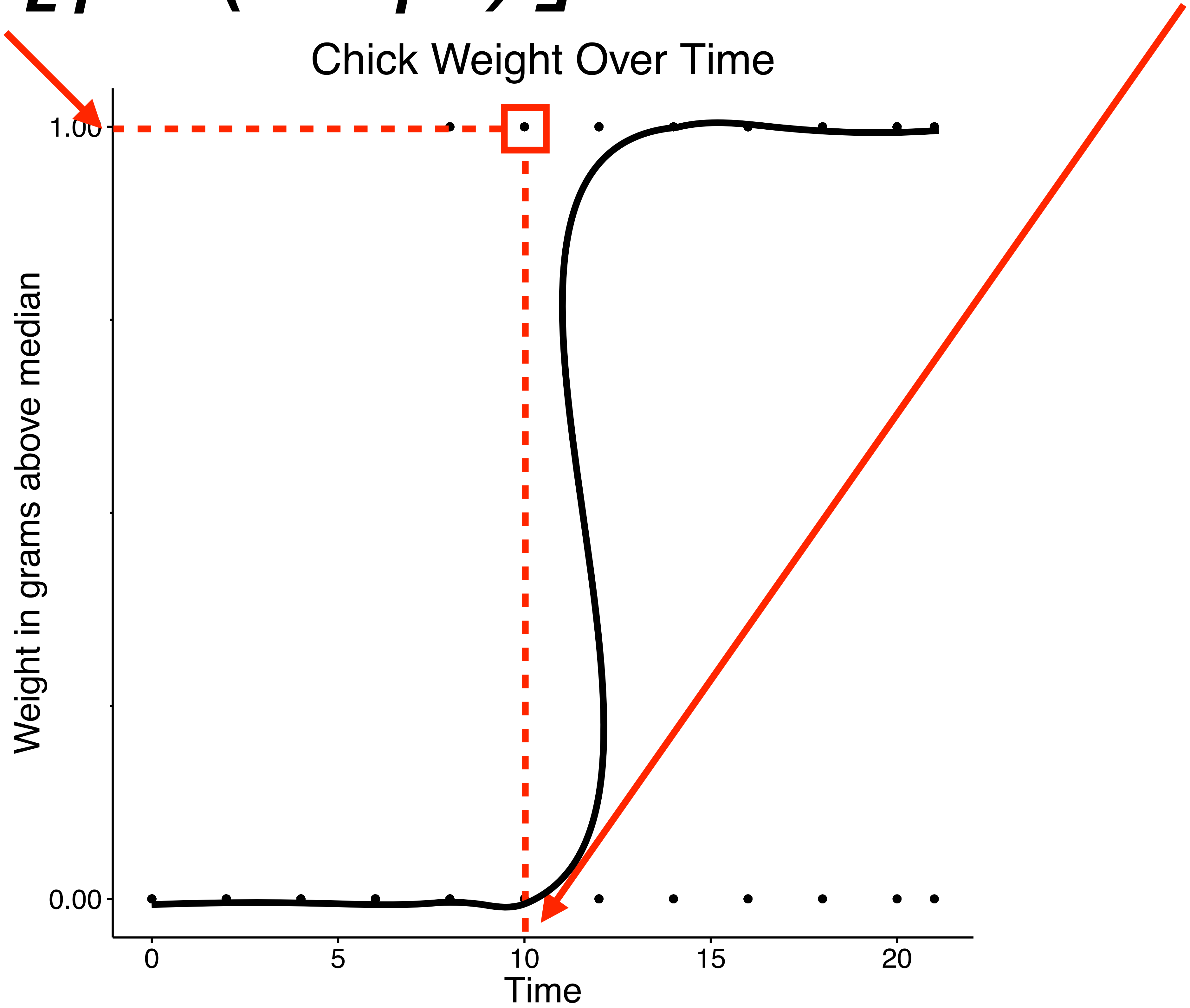
$$log[p/(1-p)]_i = a + bx_i$$



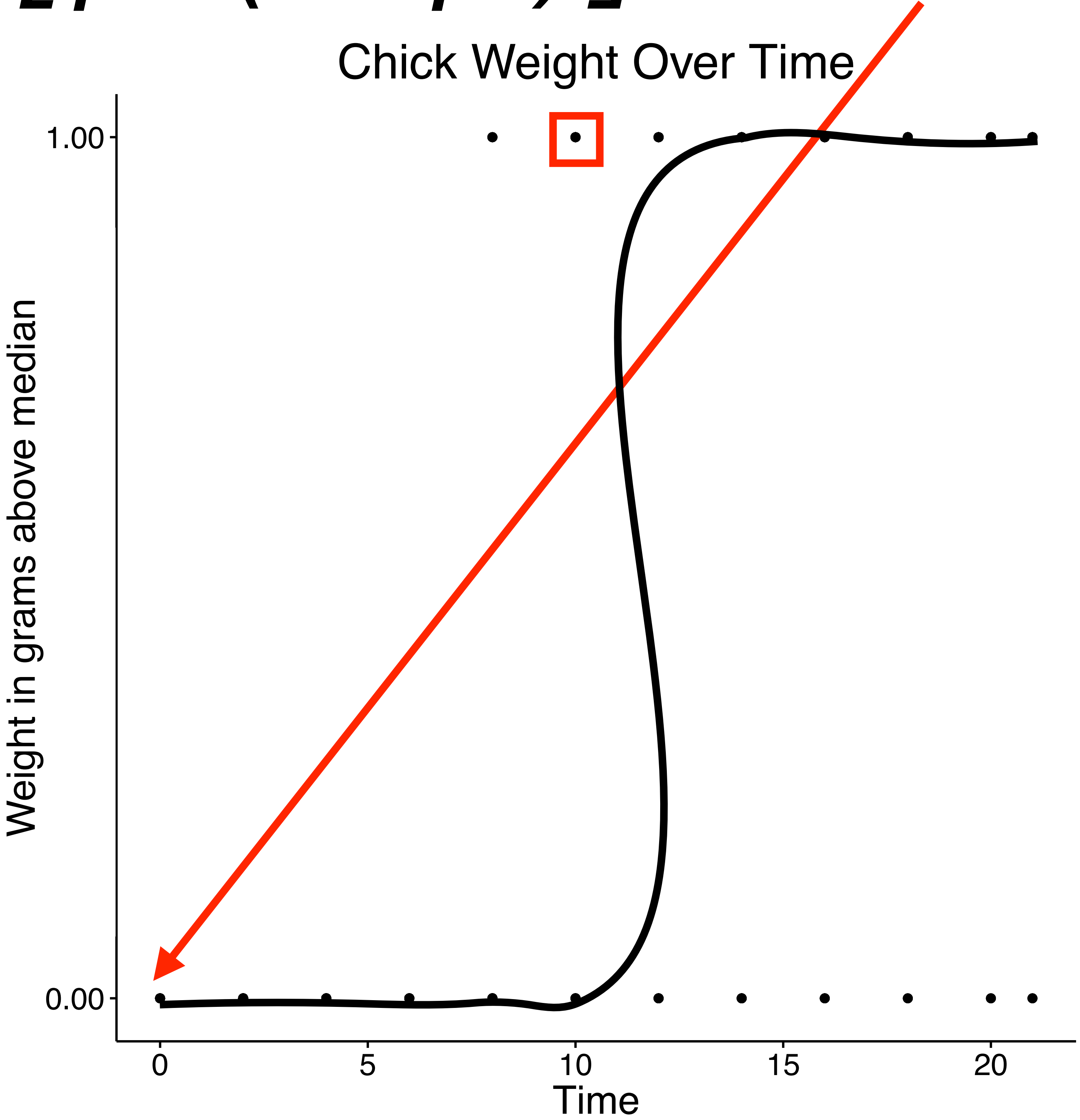
$$\log[p/(1-p)]_i = a + bx_i$$



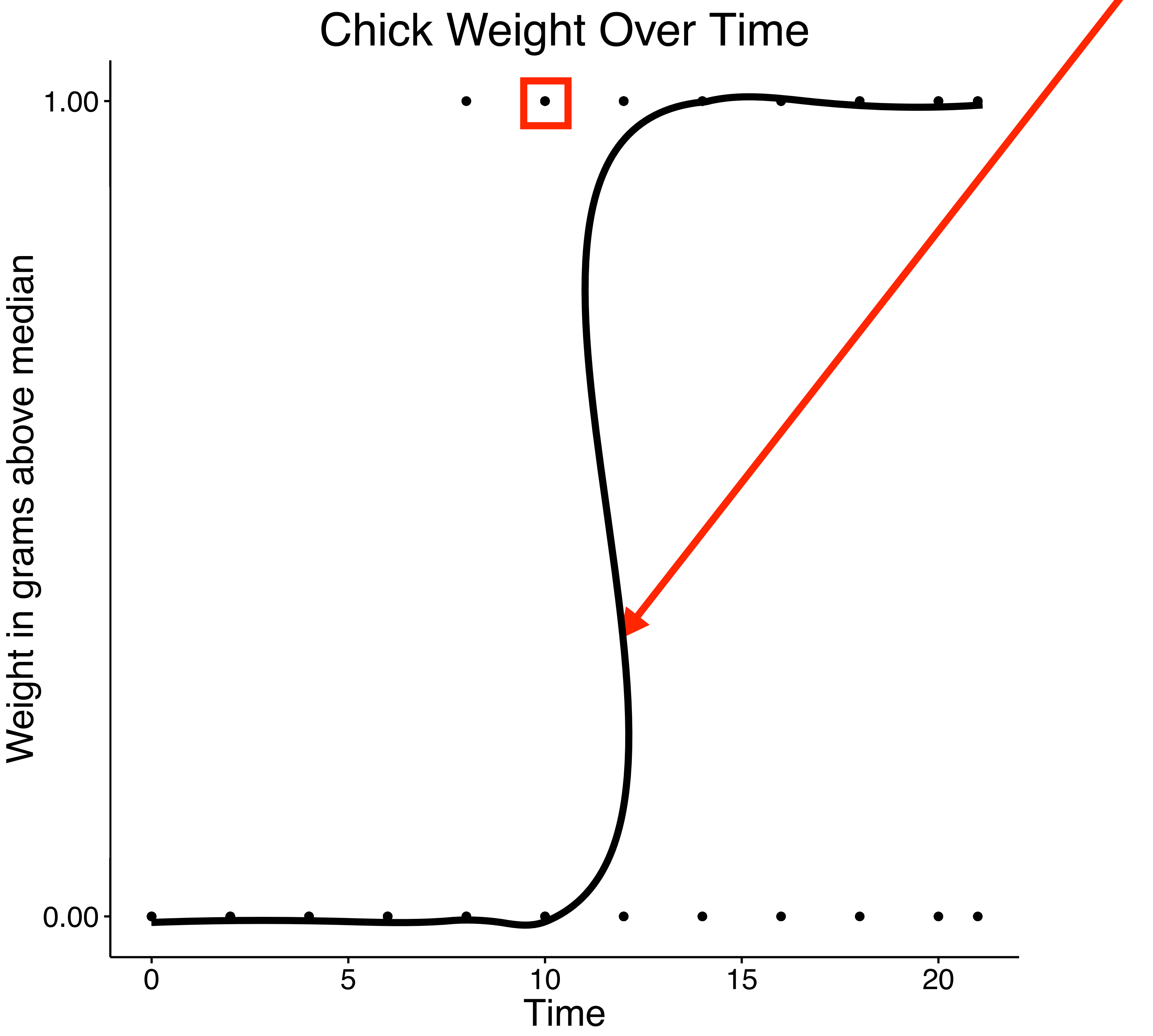
$$log[p/(1-p)]_i = a + bx_i$$



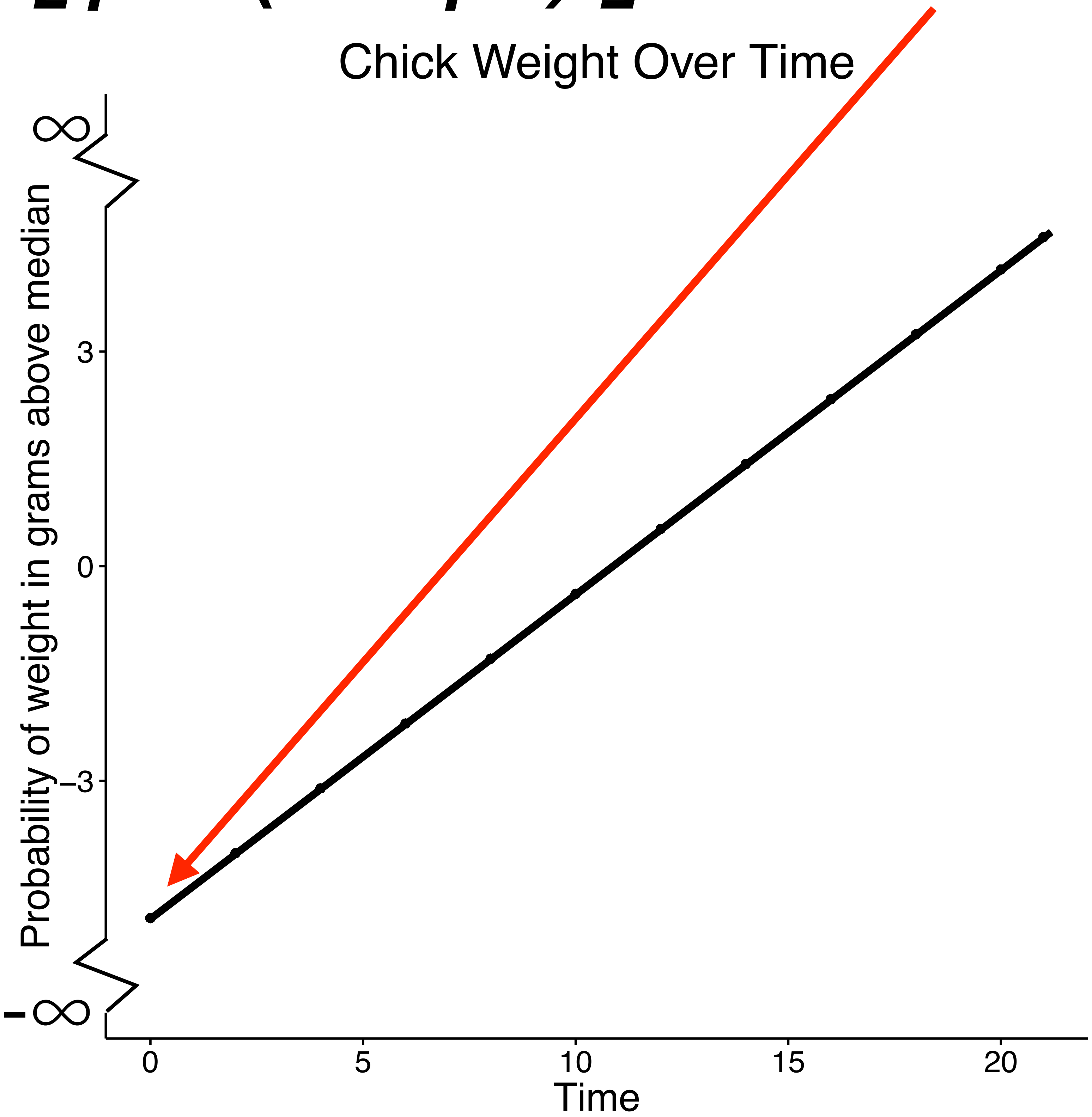
$$log[p/(1-p)]_i = a + bx_i$$



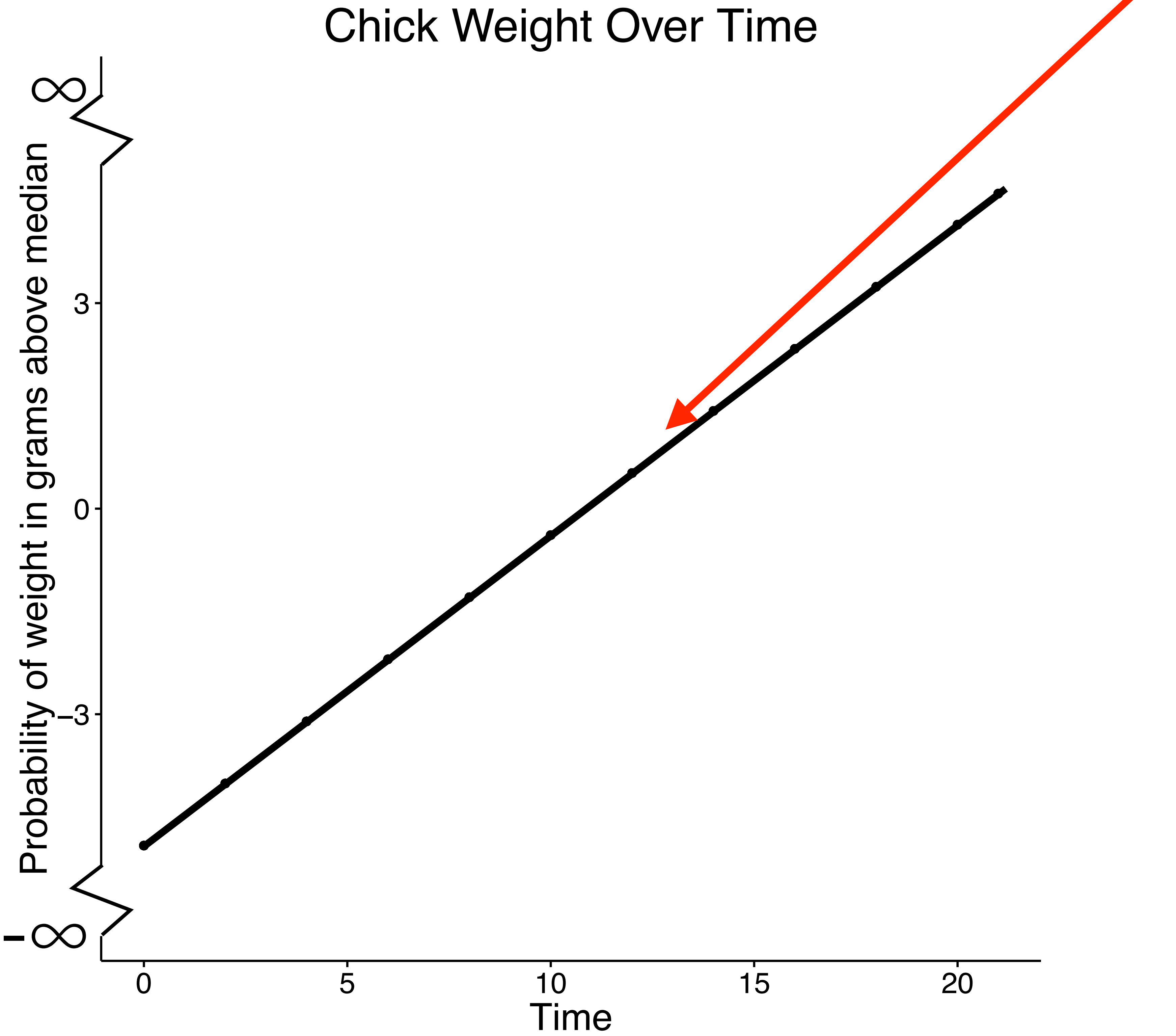
$$log[p/(1-p)]_i = a + bx_i$$



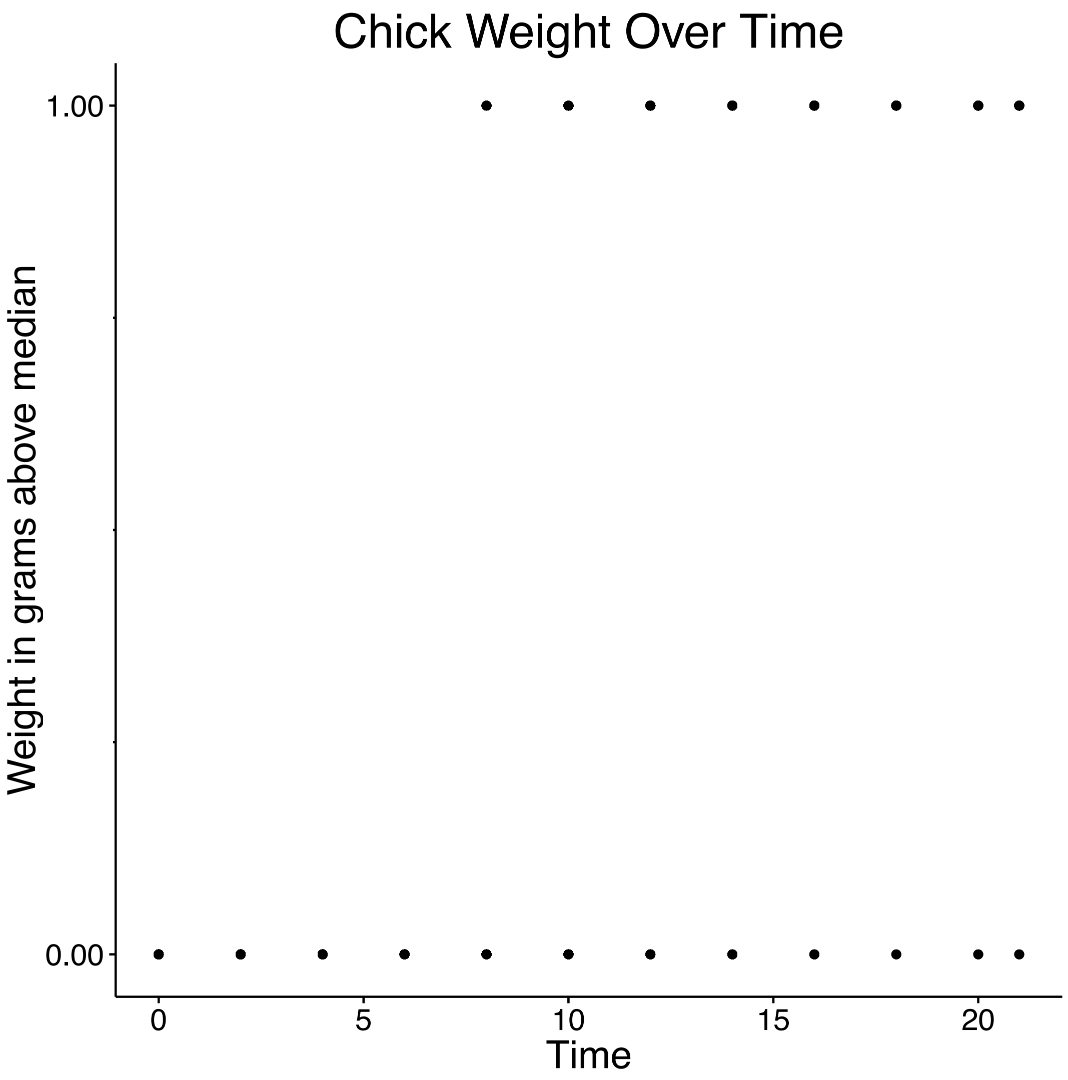
$$\log[p/(1-p)]_i = a + bx_i$$



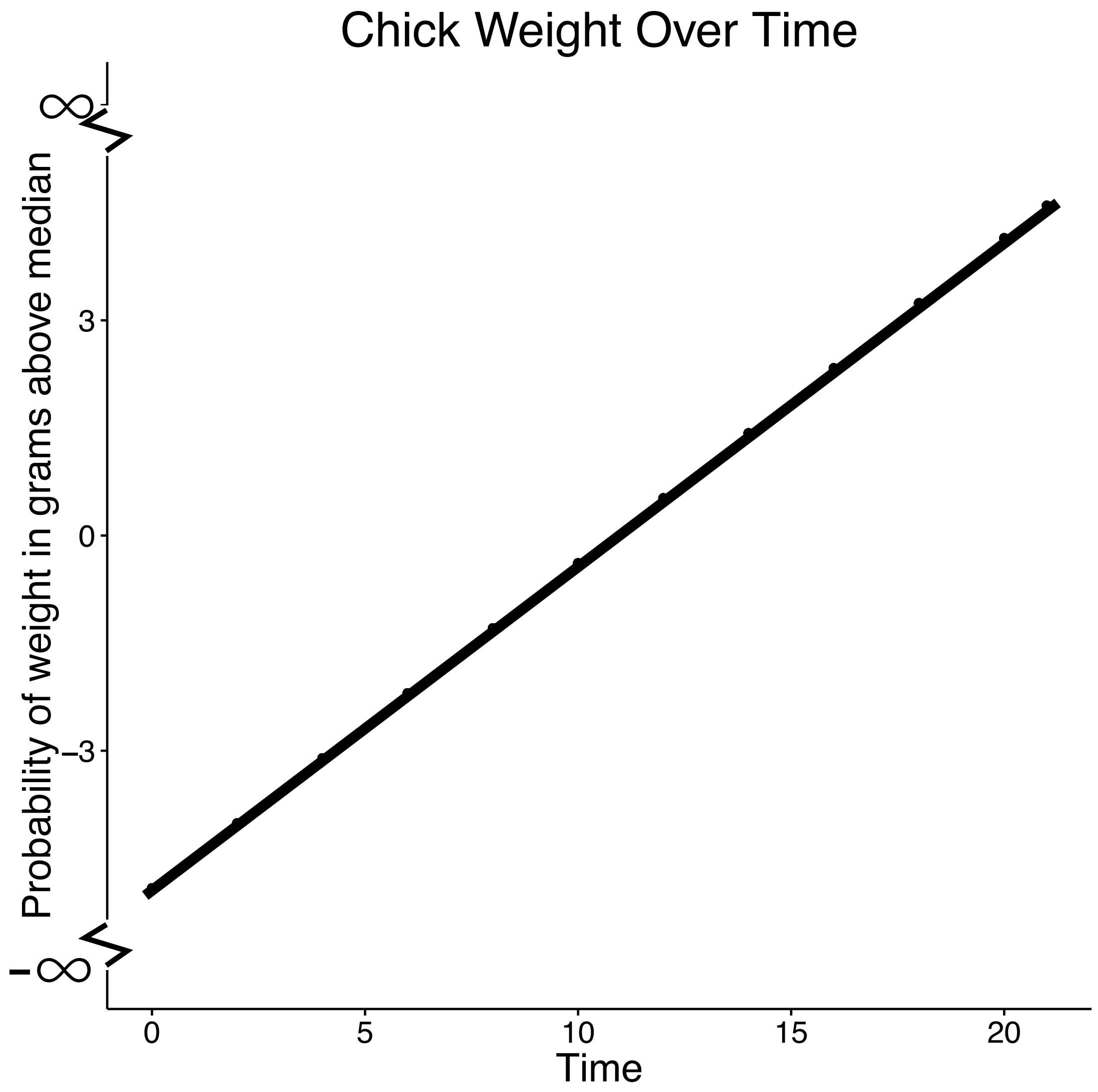
$$\log[p/(1-p)]_i = a + bx_i$$



$$log[p/(1-p)]_i = a + bx_i$$

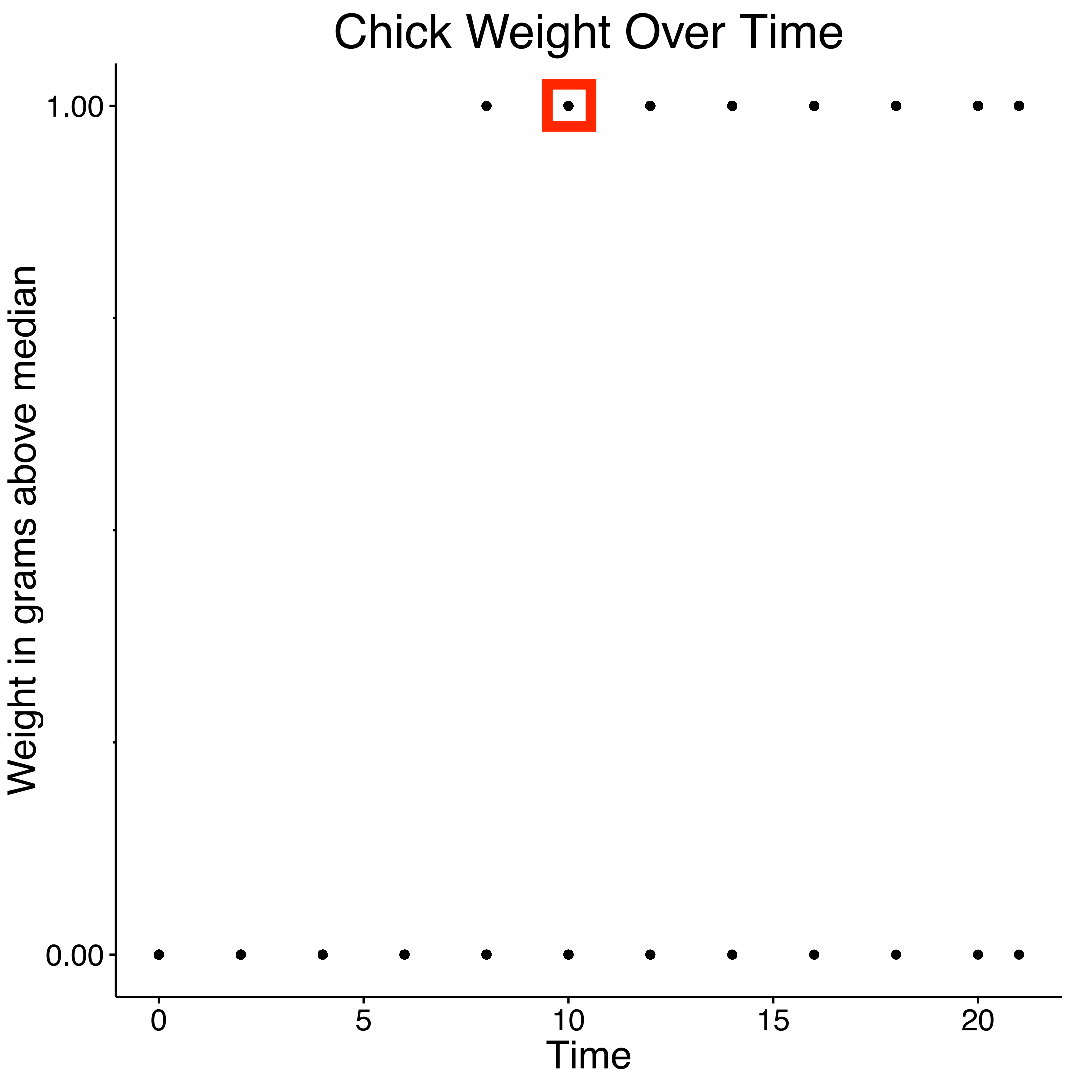


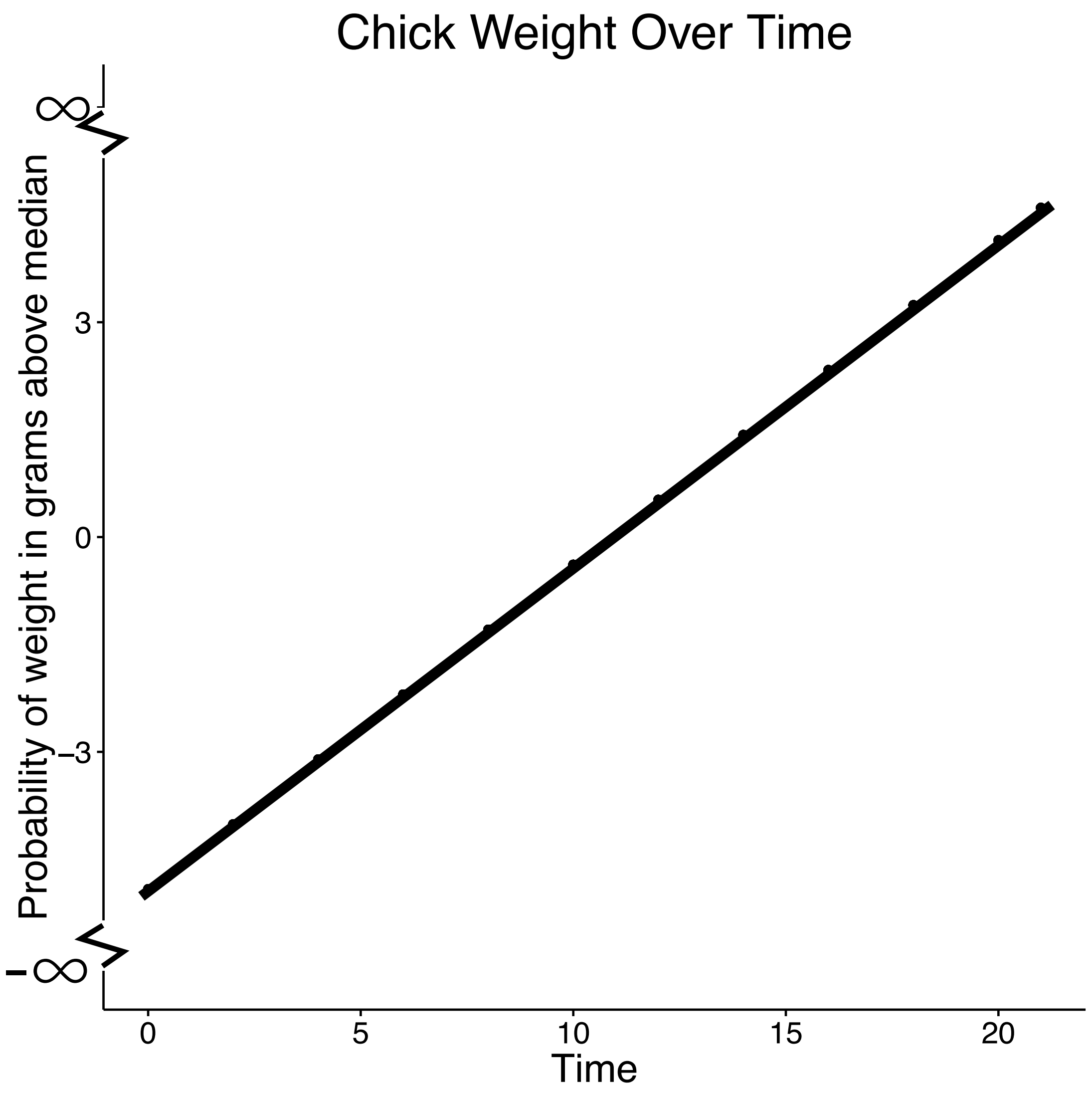
Chick Weight Over Time



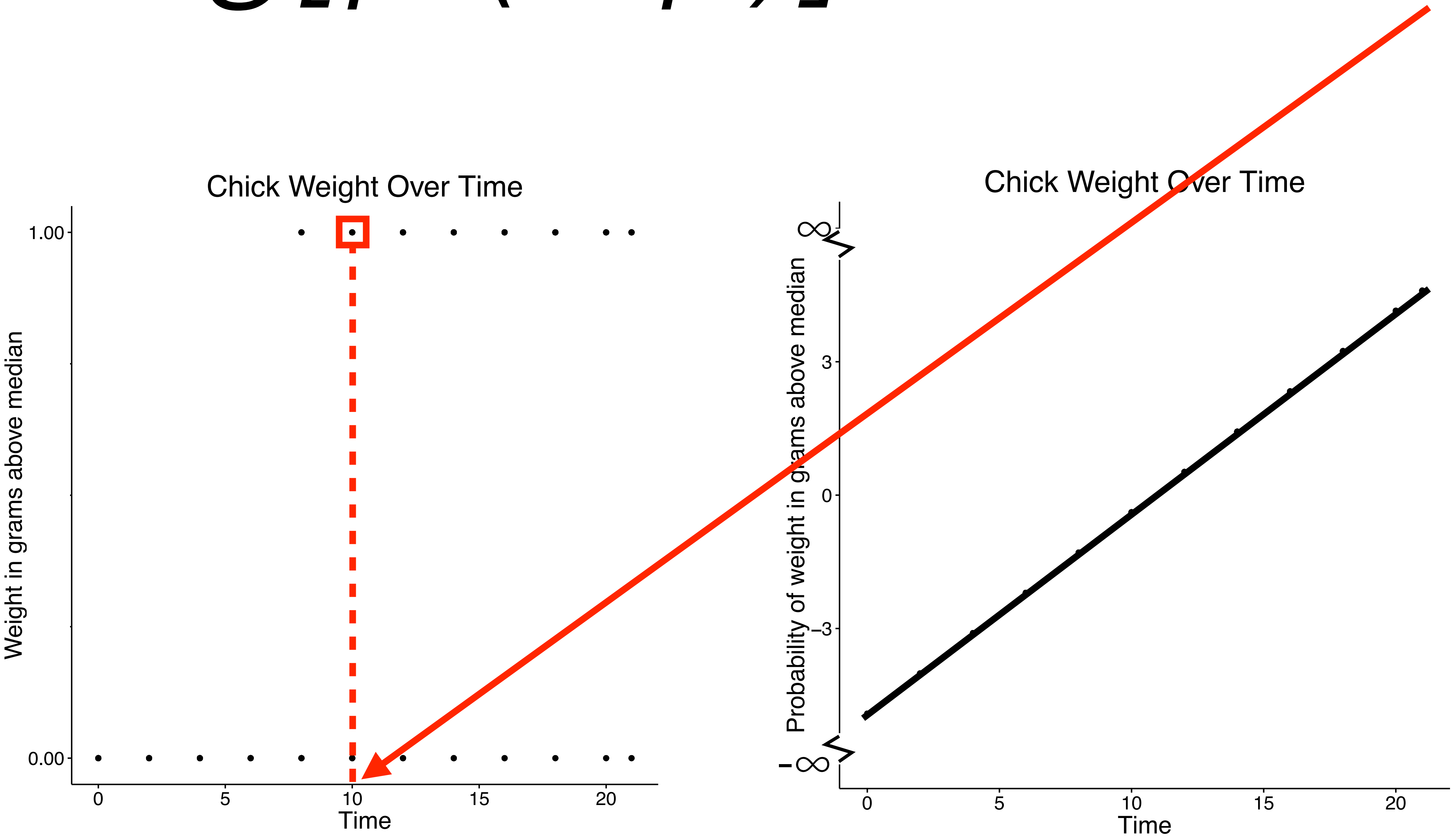
Chick Weight Over Time

$$log[p/(1-p)]_i = a + bx_i$$

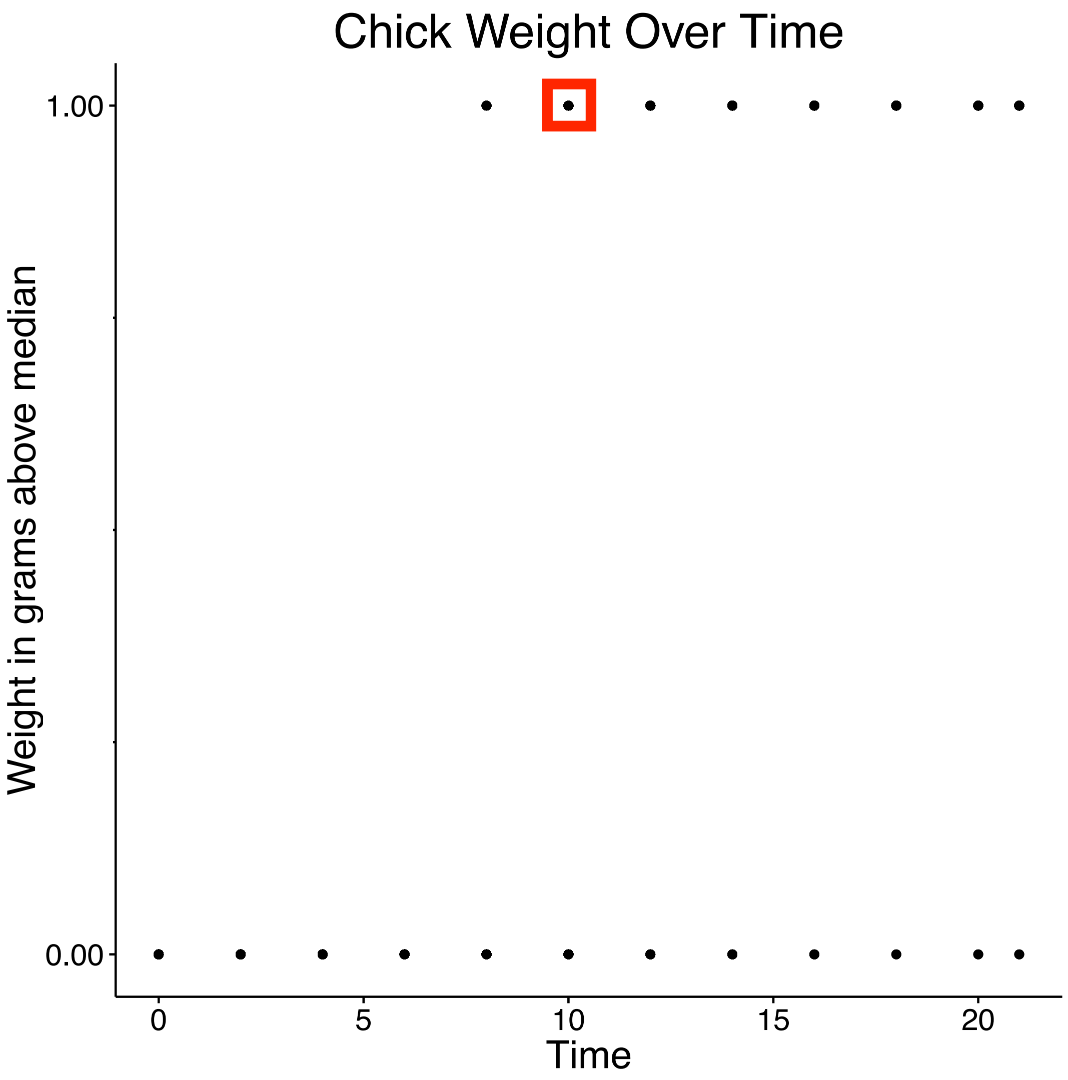


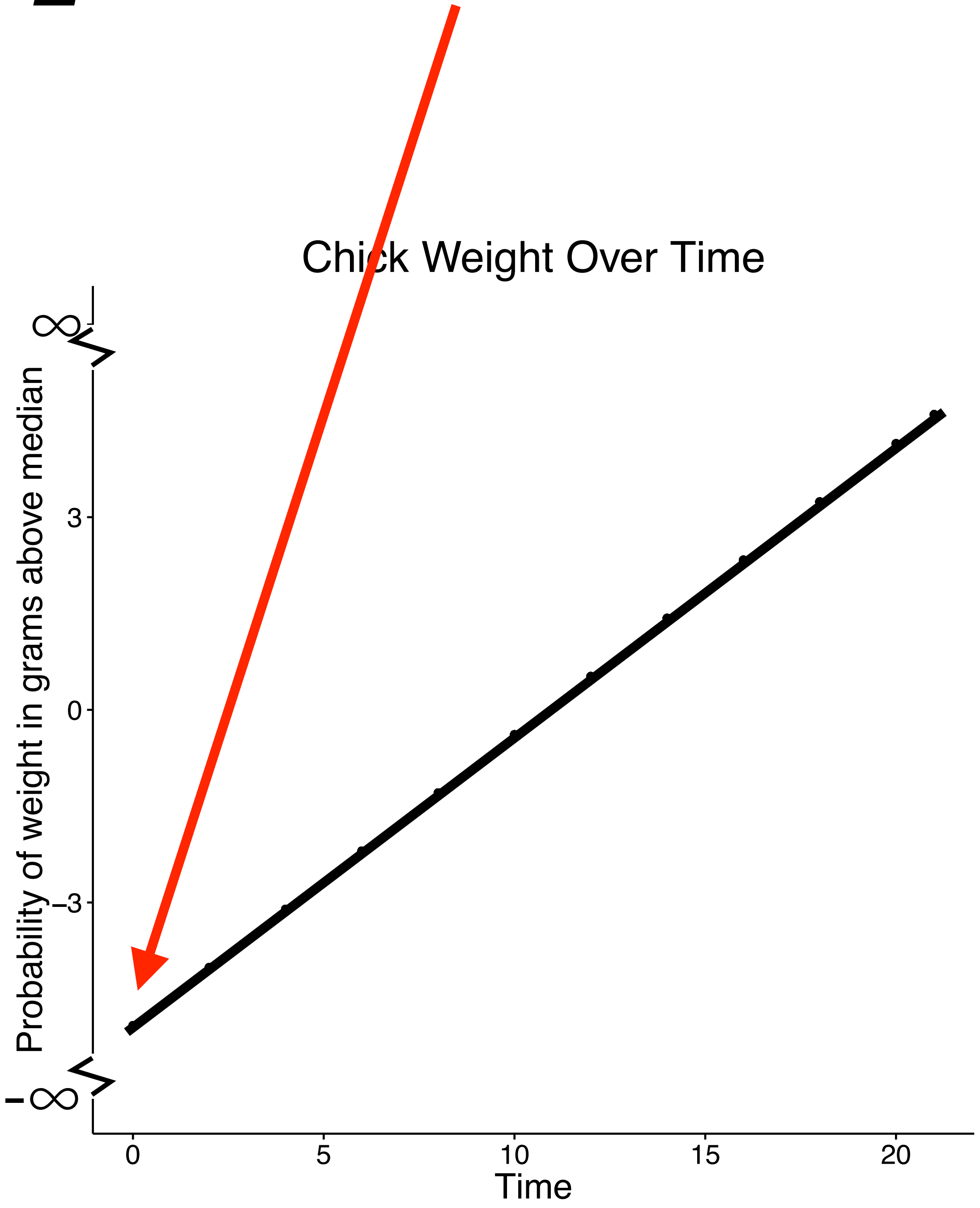


$$\log[p/(1-p)]_i = a + bx_i$$

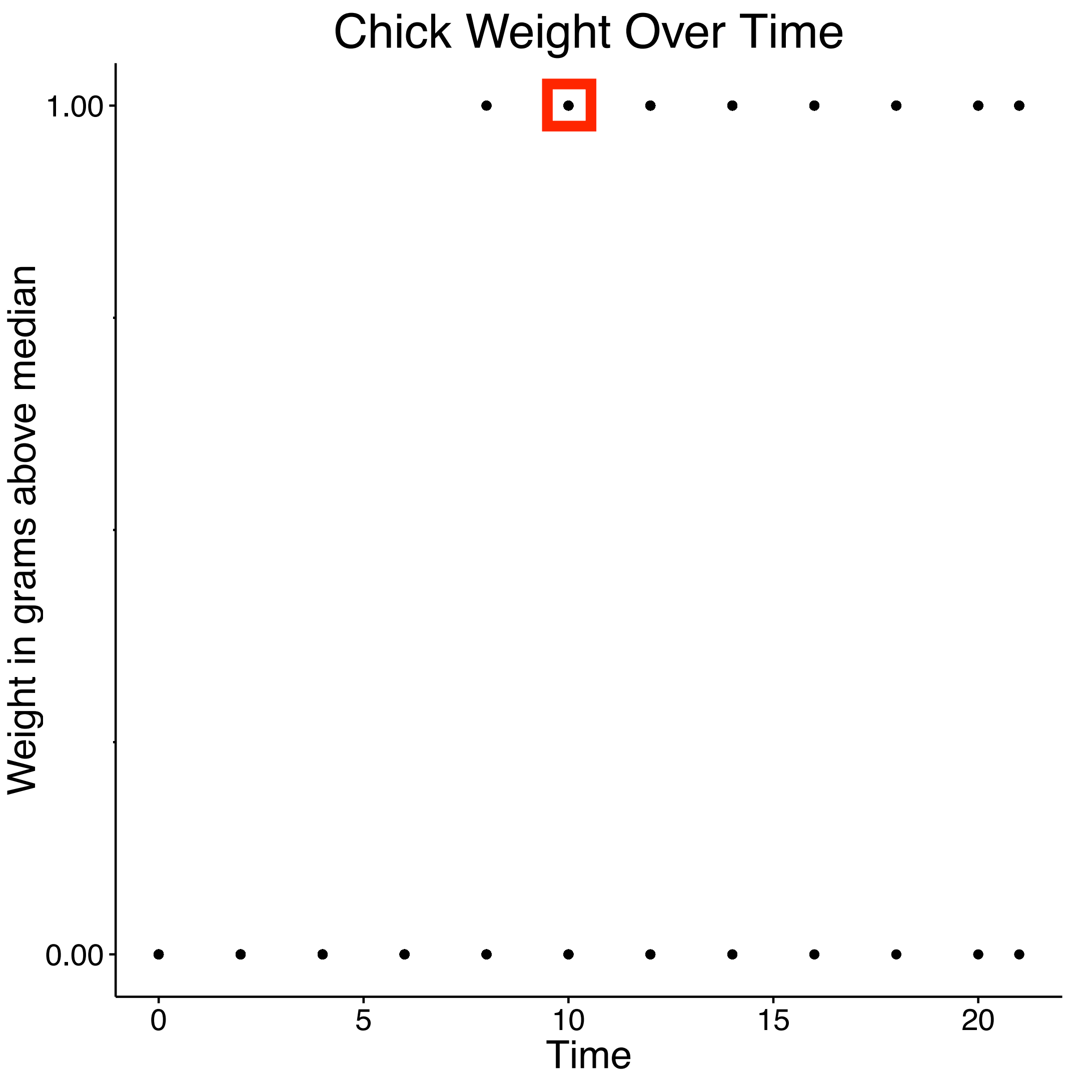


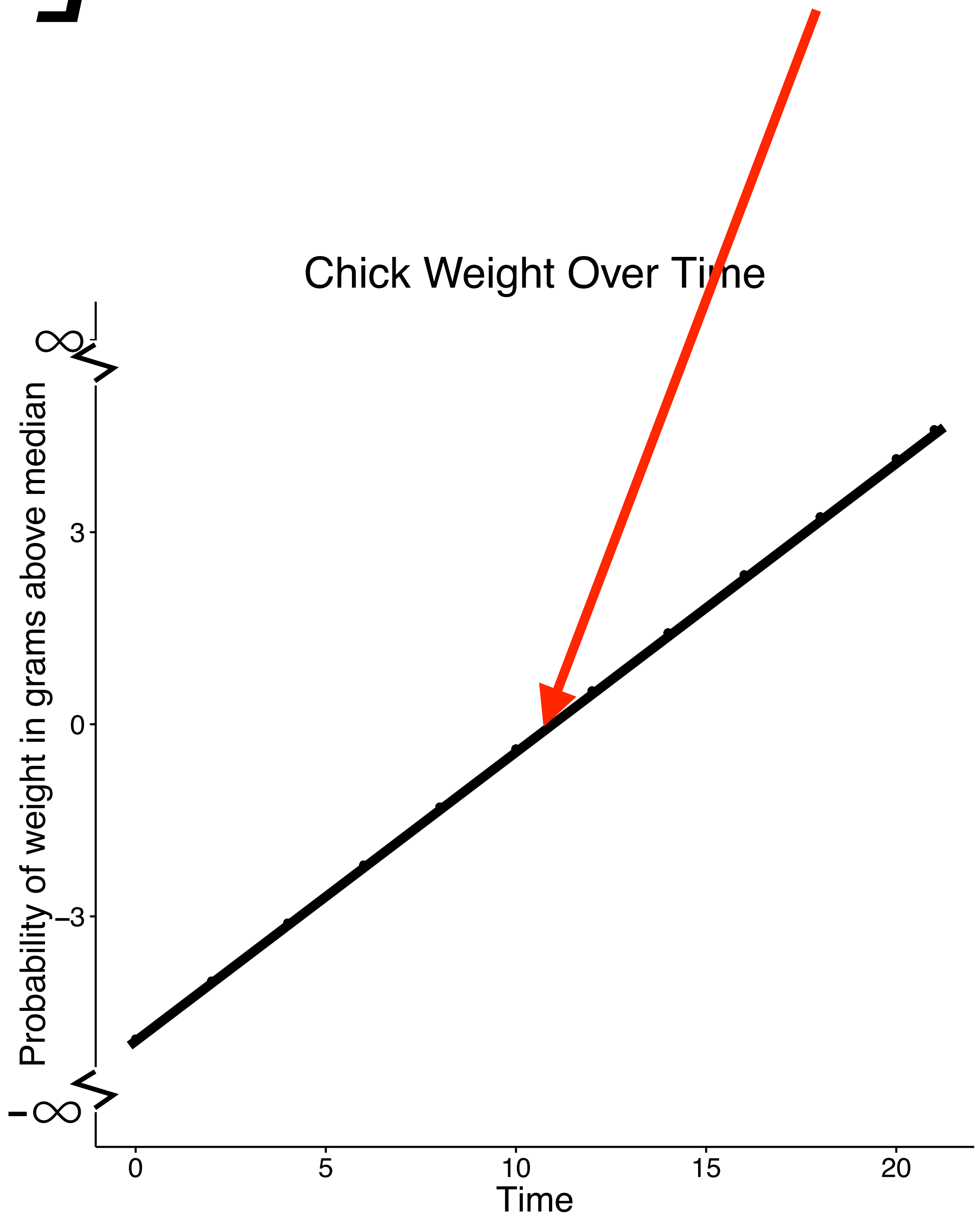
$$\log[p/(1-p)]_i = a + bx_i$$



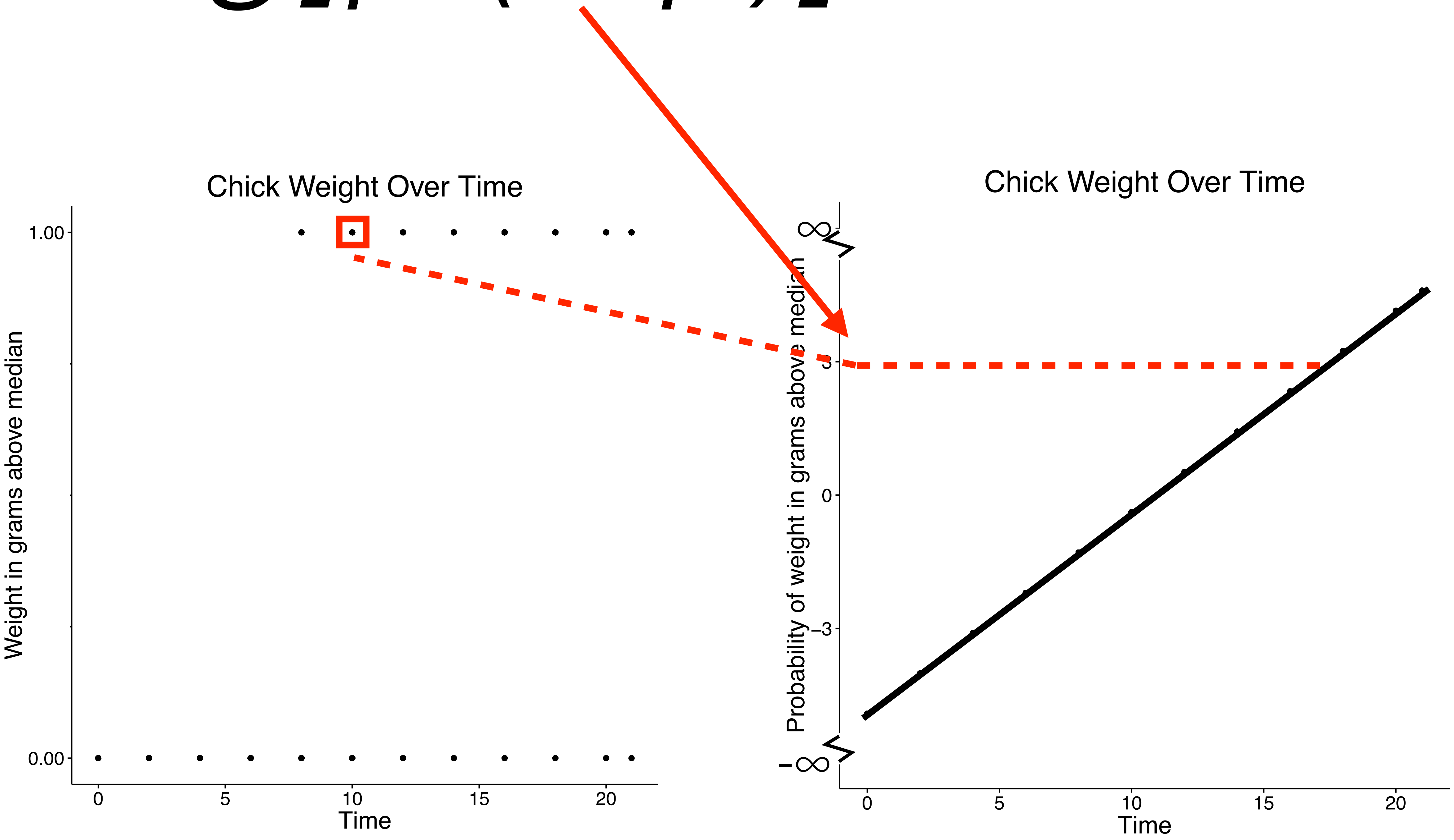


$$log[p/(1-p)]_i = a + bx_i$$



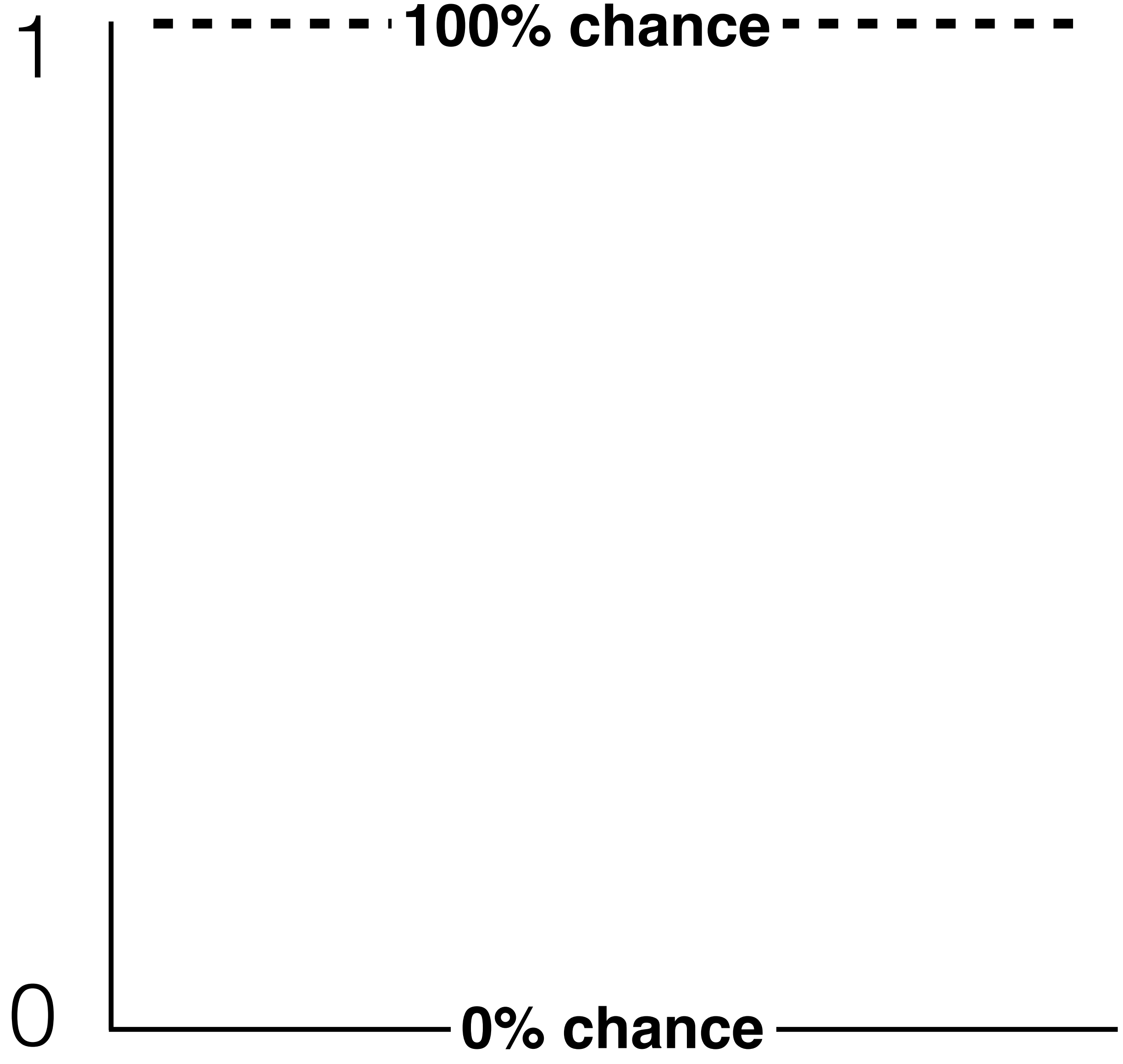


$$\log[p/(1-p)]_i = a + bx_i$$



INPUT

Probability



1

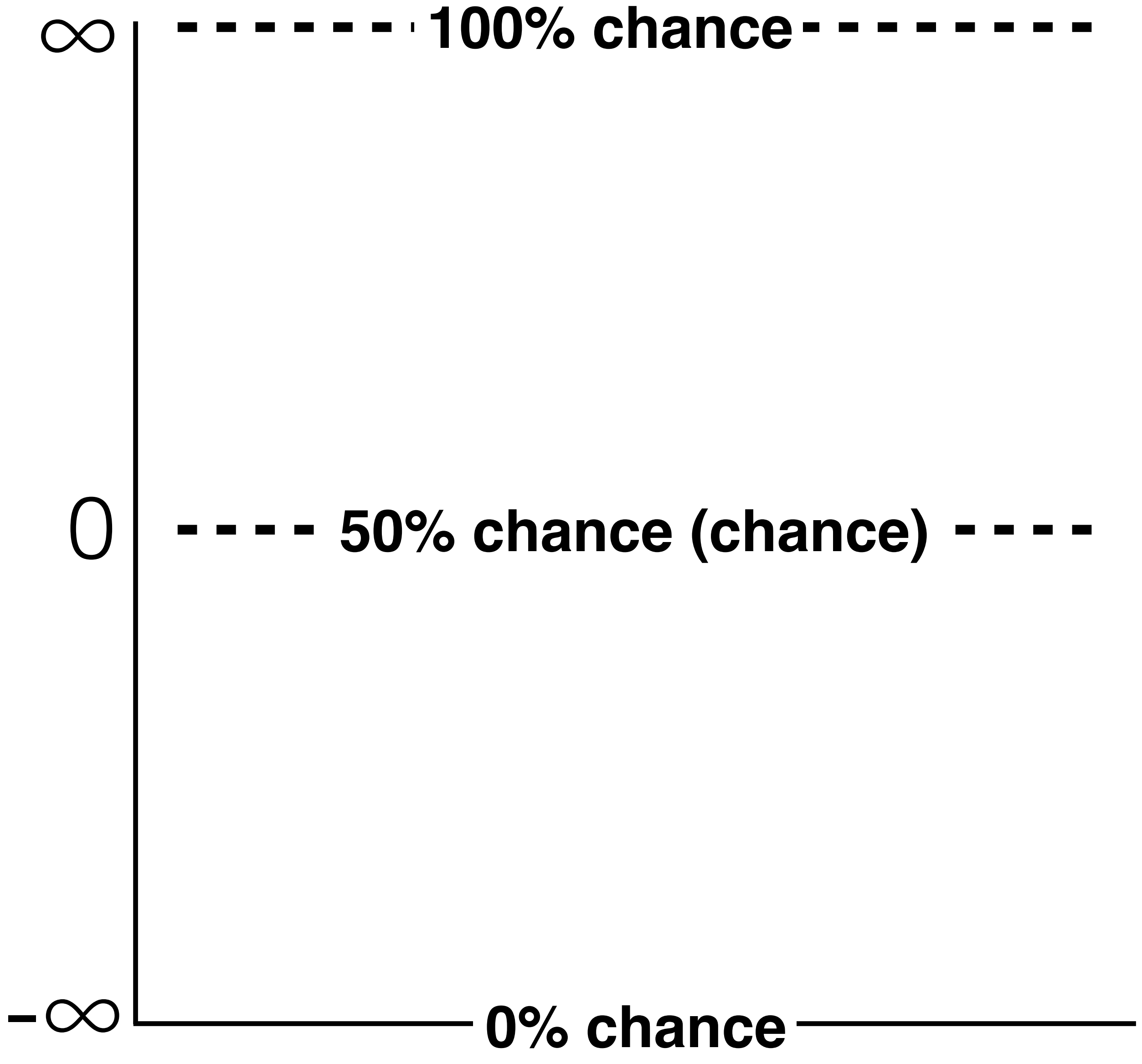
100% chance

0% chance

0

MODEL COEFFIECIENTS

Probability in Logit Space



∞

100% chance

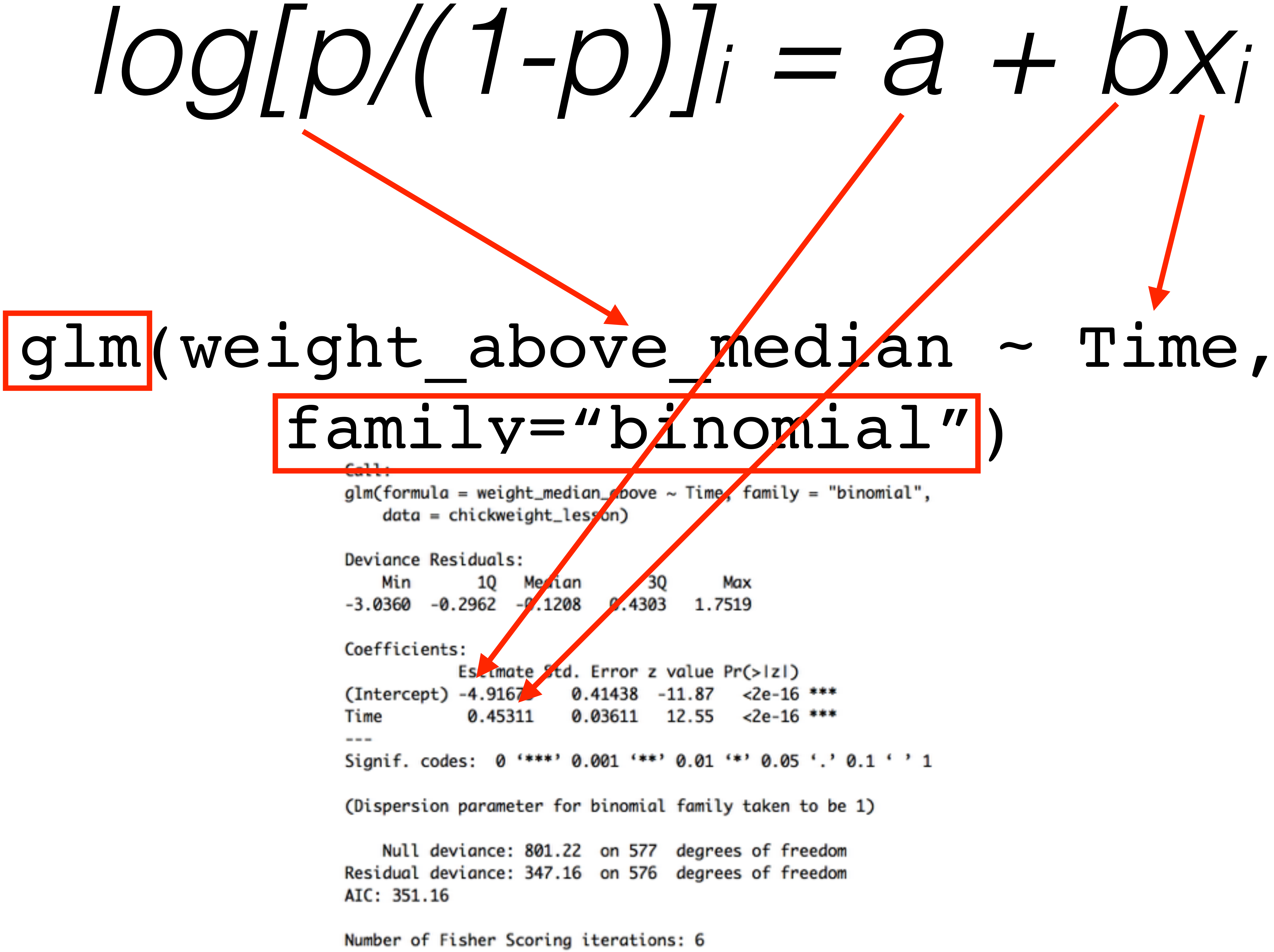
0

50% chance (chance)

−∞

0% chance

R Code



$$\log[p/(1-p)]_i = a + bx_i$$

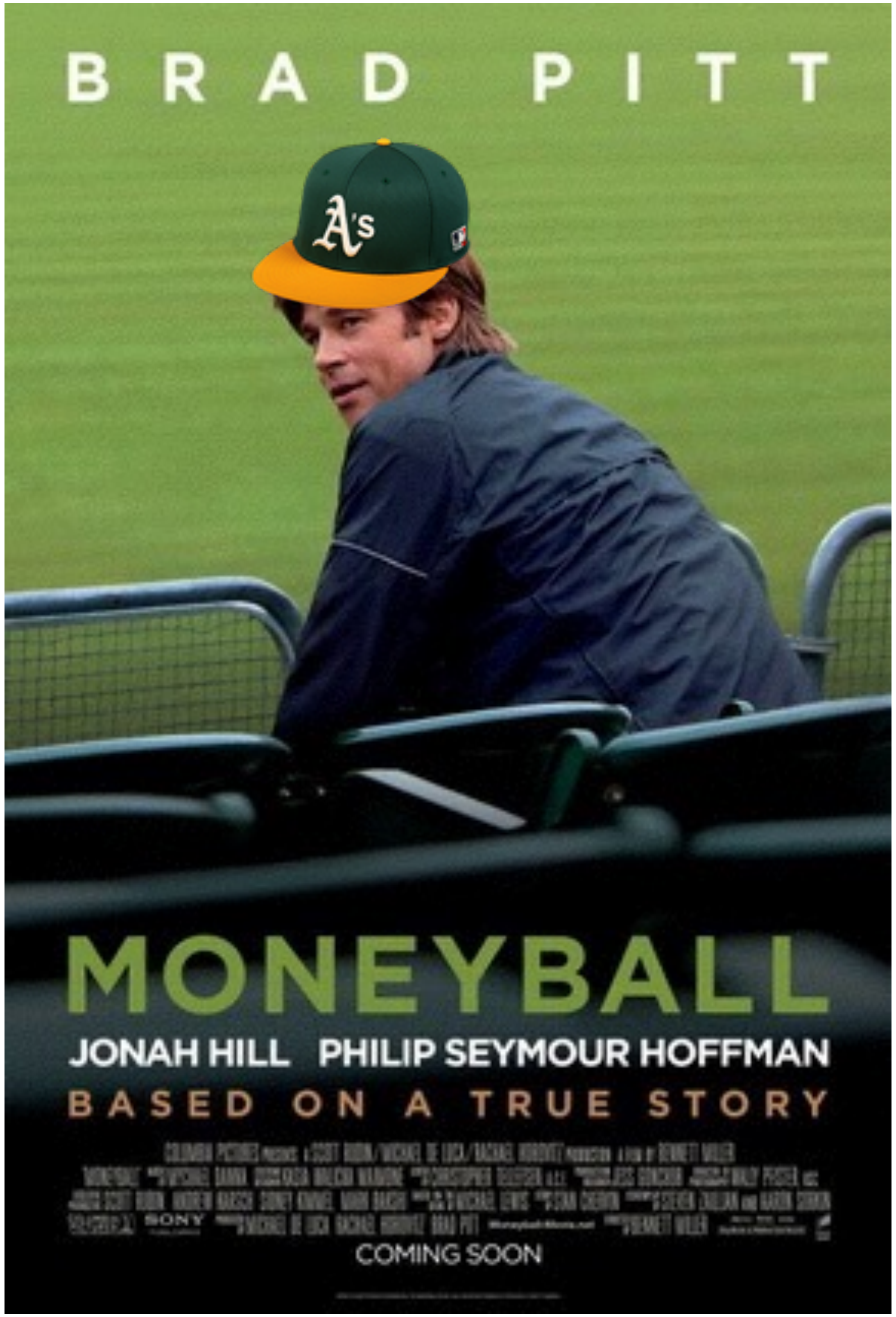
```
glm(weight_above_median ~ Time,
    family="binomial")
```

```
Call:
glm(formula = weight_median_above ~ Time, family = "binomial",
    data = chickweight_lesson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.0360  -0.2962  -0.1208   0.4303   1.7519

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.91678    0.41438  -11.87   <2e-16 ***
Time         0.45311    0.03611   12.55   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 801.22  on 577  degrees of freedom
Residual deviance: 347.16  on 576  degrees of freedom
AIC: 351.16

Number of Fisher Scoring iterations: 6
```

Lab

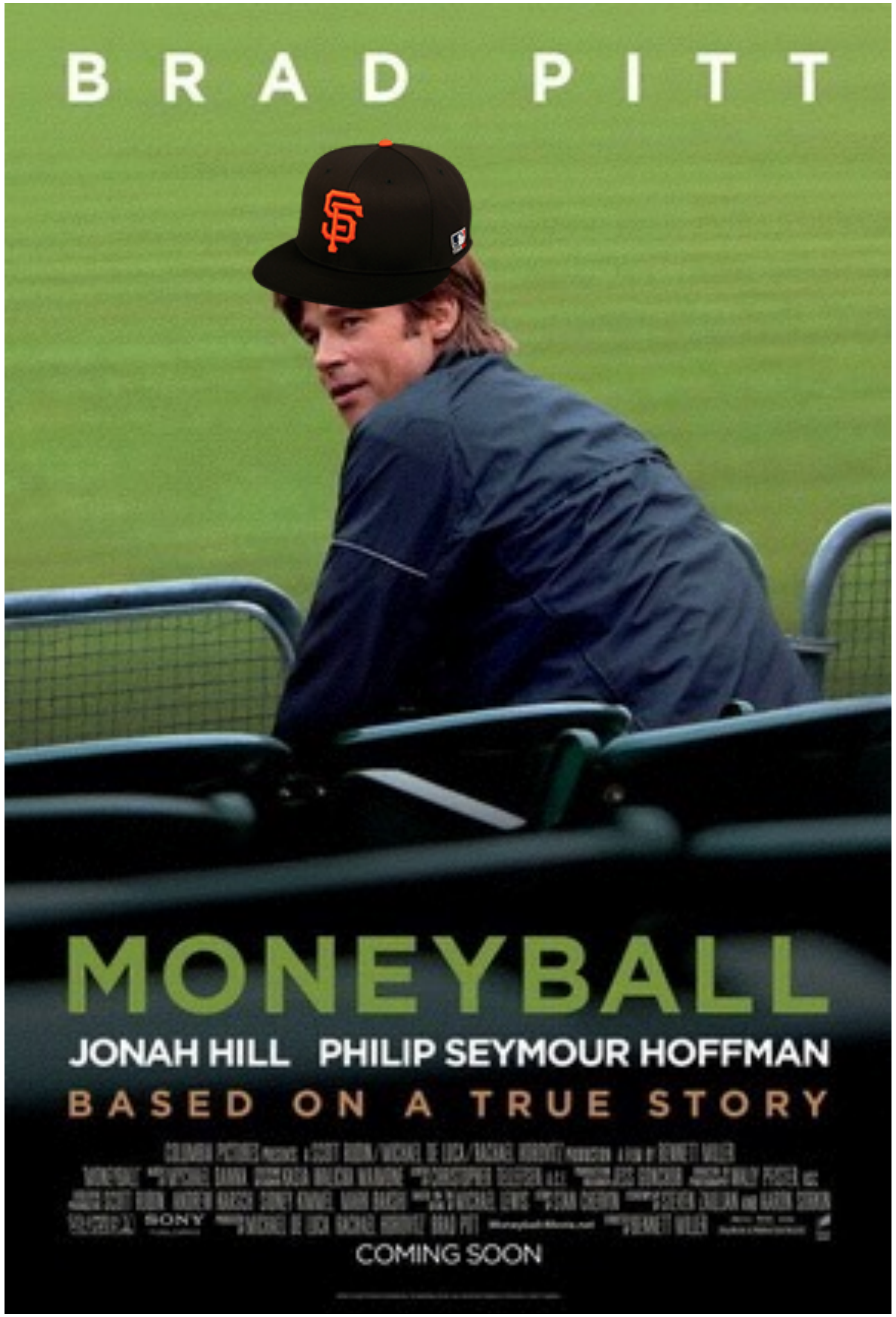
Data set: The San Francisco Giants 2010 Baseball Season

source: retrosheet.org


BRAD PITT
A's
MONEYBALL
JONAH HILL PHILIP SEYMOUR HOFFMAN
BASED ON A TRUE STORY
SONY
COMING SOON

A's

R


BRAD PITT
MONEYBALL
JONAH HILL PHILIP SEYMOUR HOFFMAN
BASED ON A TRUE STORY
SONY
COMING SOON

R

Data set: The San Francisco Giants
2010 Baseball Season

Full Season: Did the Giants win more games before or after the All-Star break?

Buster Posey: Are the Giant’s more likely to win in games where Buster Posey was walked at least once?

Full Season

logit p_i	=	win or loss
a	=	? - from model
b	=	? - from model
x_i	=	All Star break

Buster Posey

logit p_i	=	win or loss
a	=	? - from model
b	=	? - from model
x_i	=	walked

source: retrosheet.org

dplyr

```
data_clean = data
```

dplyr

```
data_clean = data %>%
```

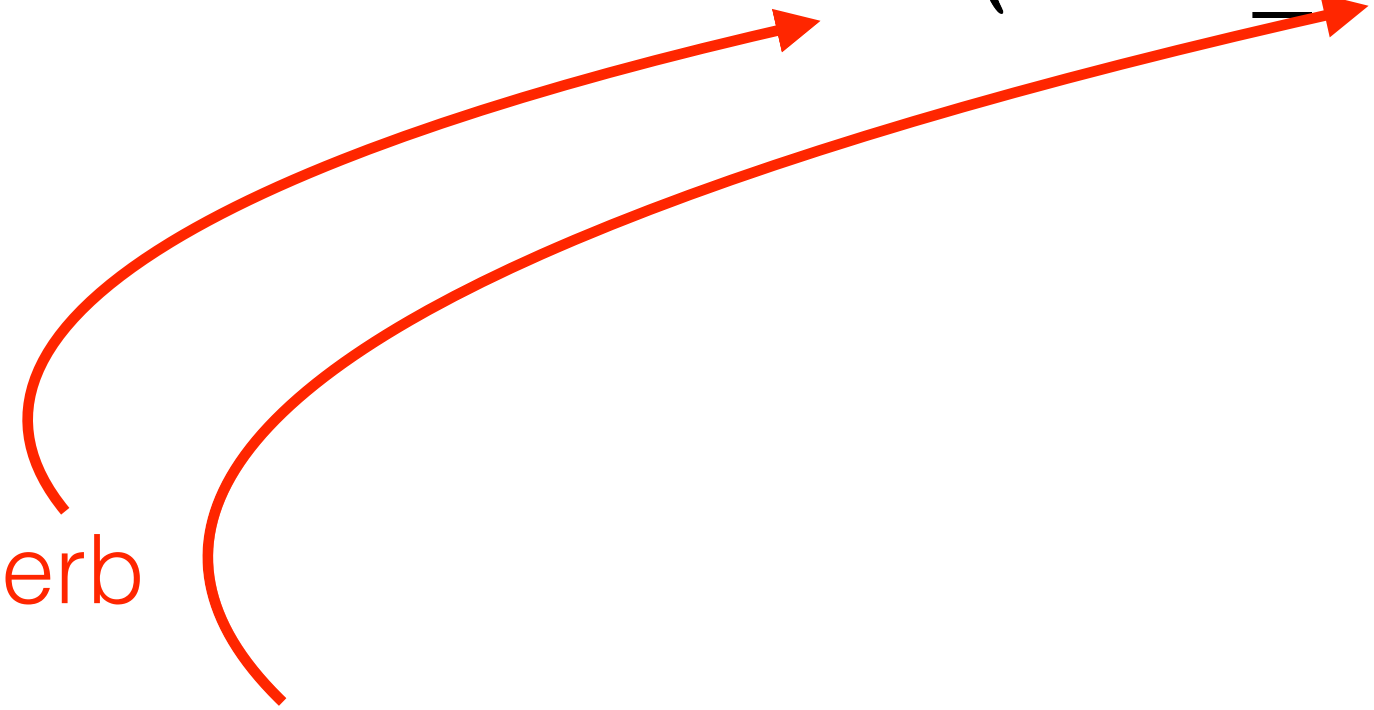
dplyr

```
data_clean = data %>%
             mutate(
                                             )
```

verb

dplyr

```
data_clean = data %>%
             mutate(home_visitor =

                                             )
```



verb

new variable

dplyr

```
data_clean = data %>%
             mutate(home_visitor =
                    ifelse(
                                            ))
```

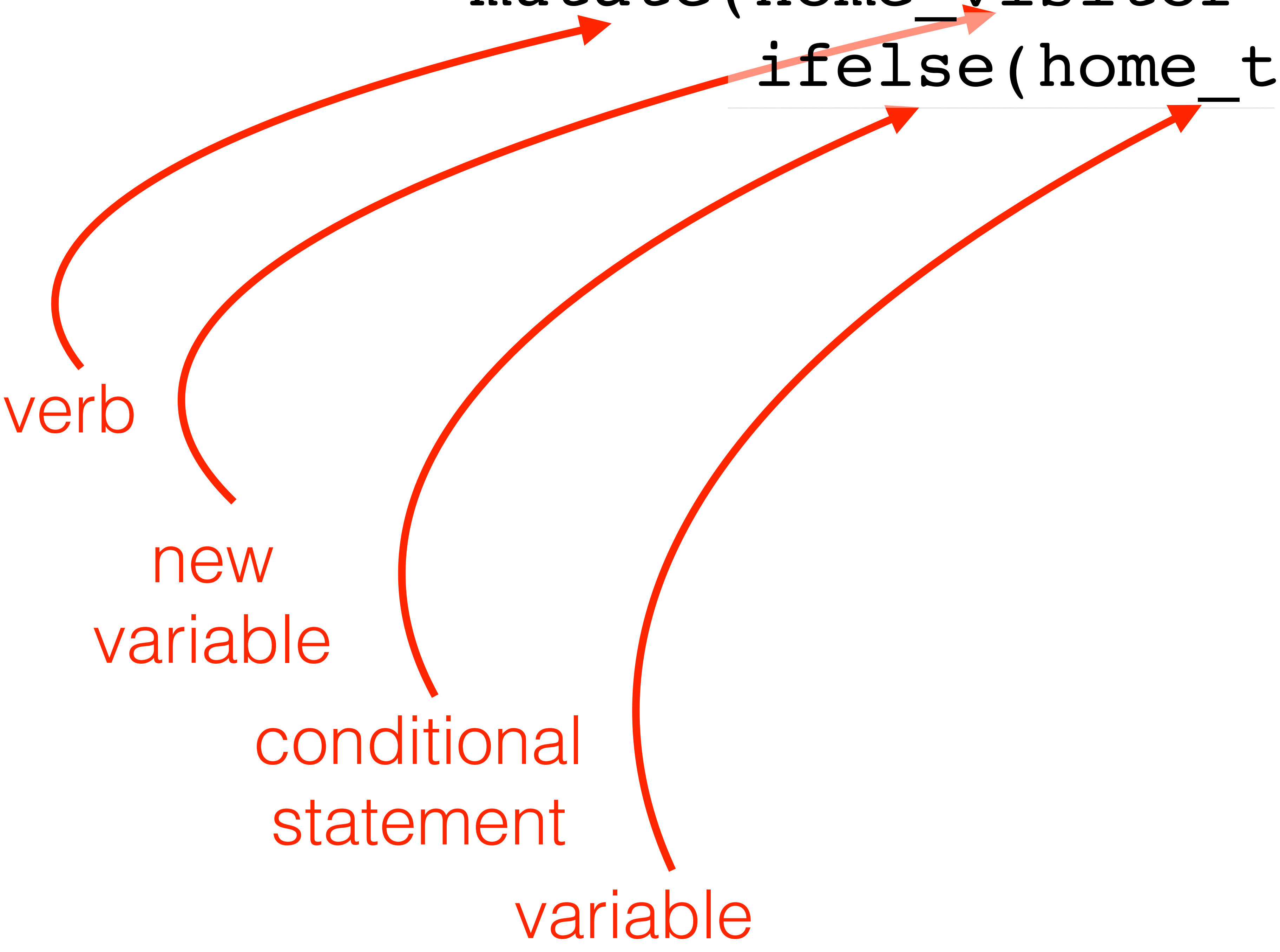
verb

new variable

conditional statement

dplyr

```
data_clean = data %>%
              mutate(home_visitor =
                     ifelse(home_team
                                                   ))
```



verb

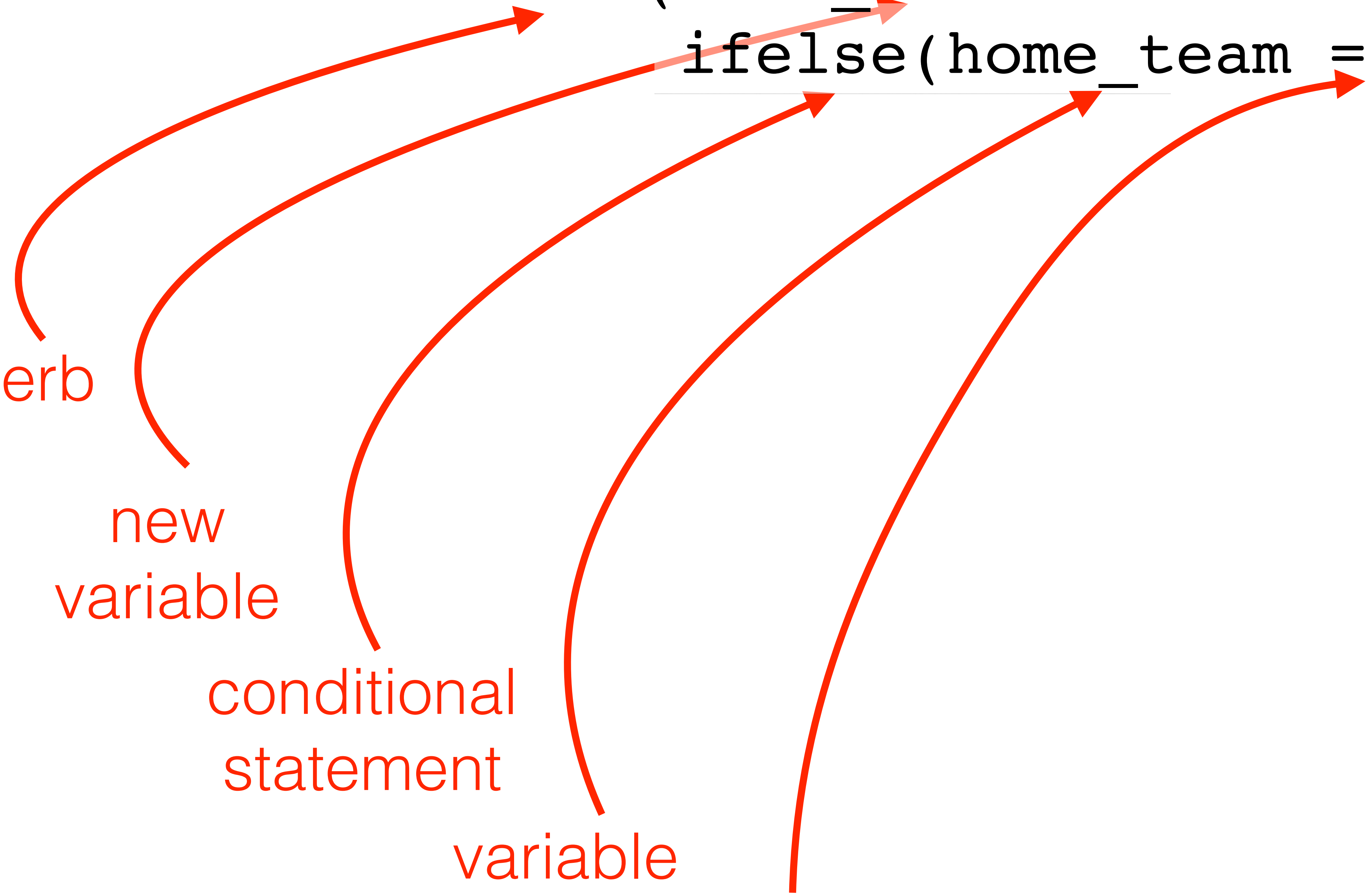
new variable

conditional statement

variable

dplyr

```
data_clean = data %>%
             mutate(home_visitor =
                    ifelse(home_team ==
                                                ))
```



verb

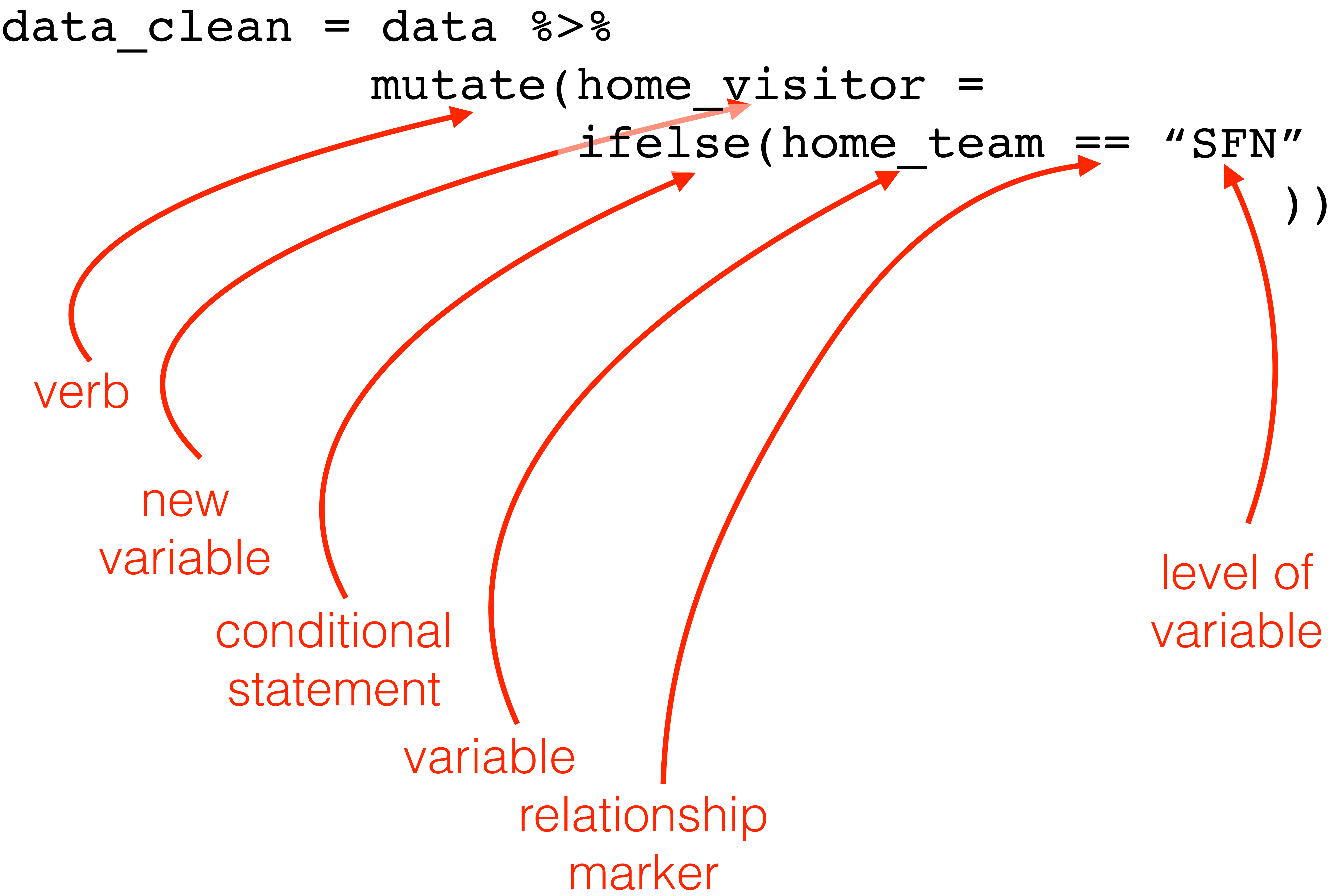
new
variable

conditional
statement

variable

relationship
marker

dplyr



```
data_clean = data %>%
             mutate(home_visitor =
                    ifelse(home_team == “SFN”
                                            ))
```

verb

new variable

conditional statement

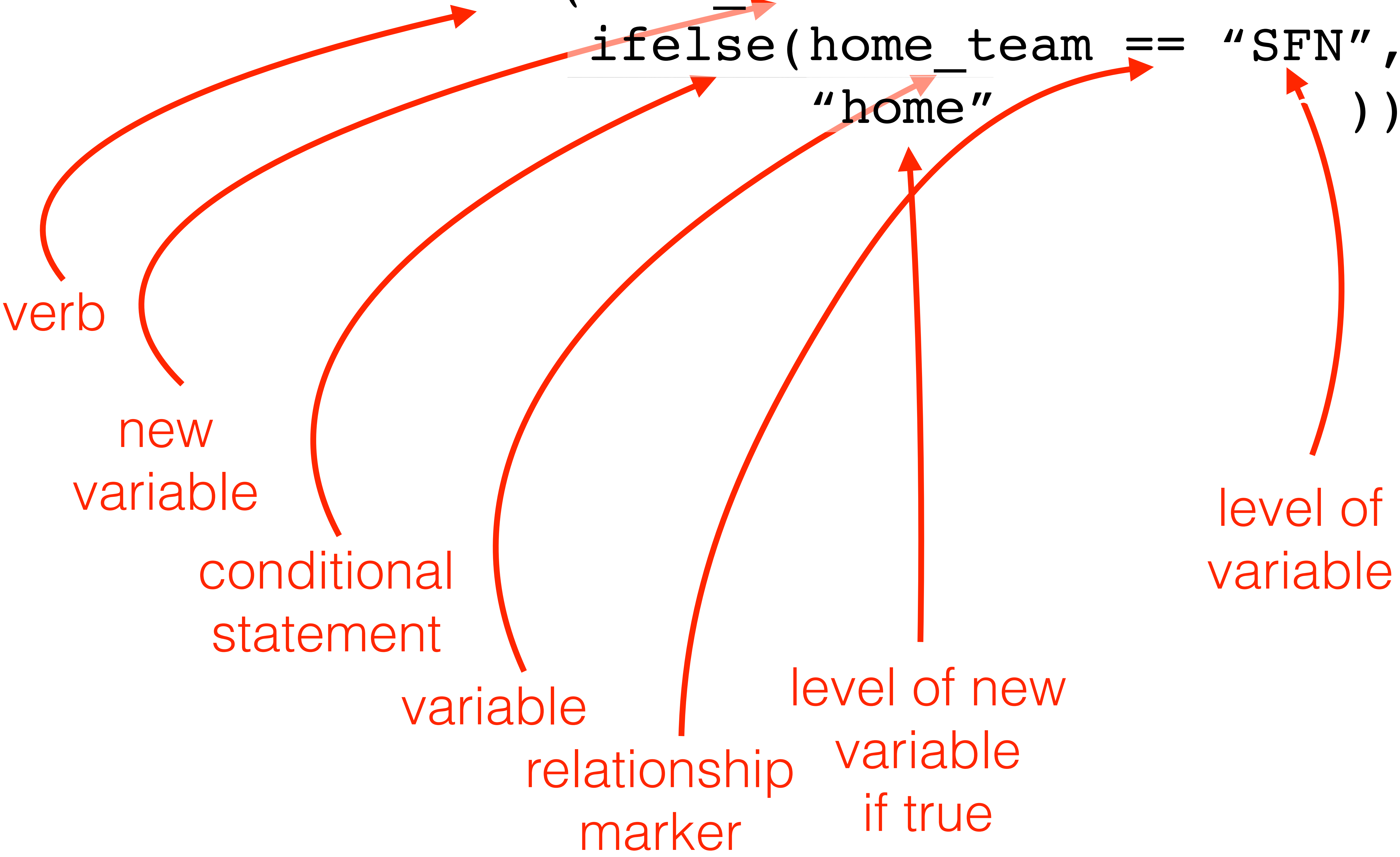
variable

relationship marker

level of variable

dplyr

```
data_clean = data %>%
             mutate(home_visitor =
                    ifelse(home_team == “SFN”,
                           “home”                ))
```



verb

new variable

conditional statement

variable

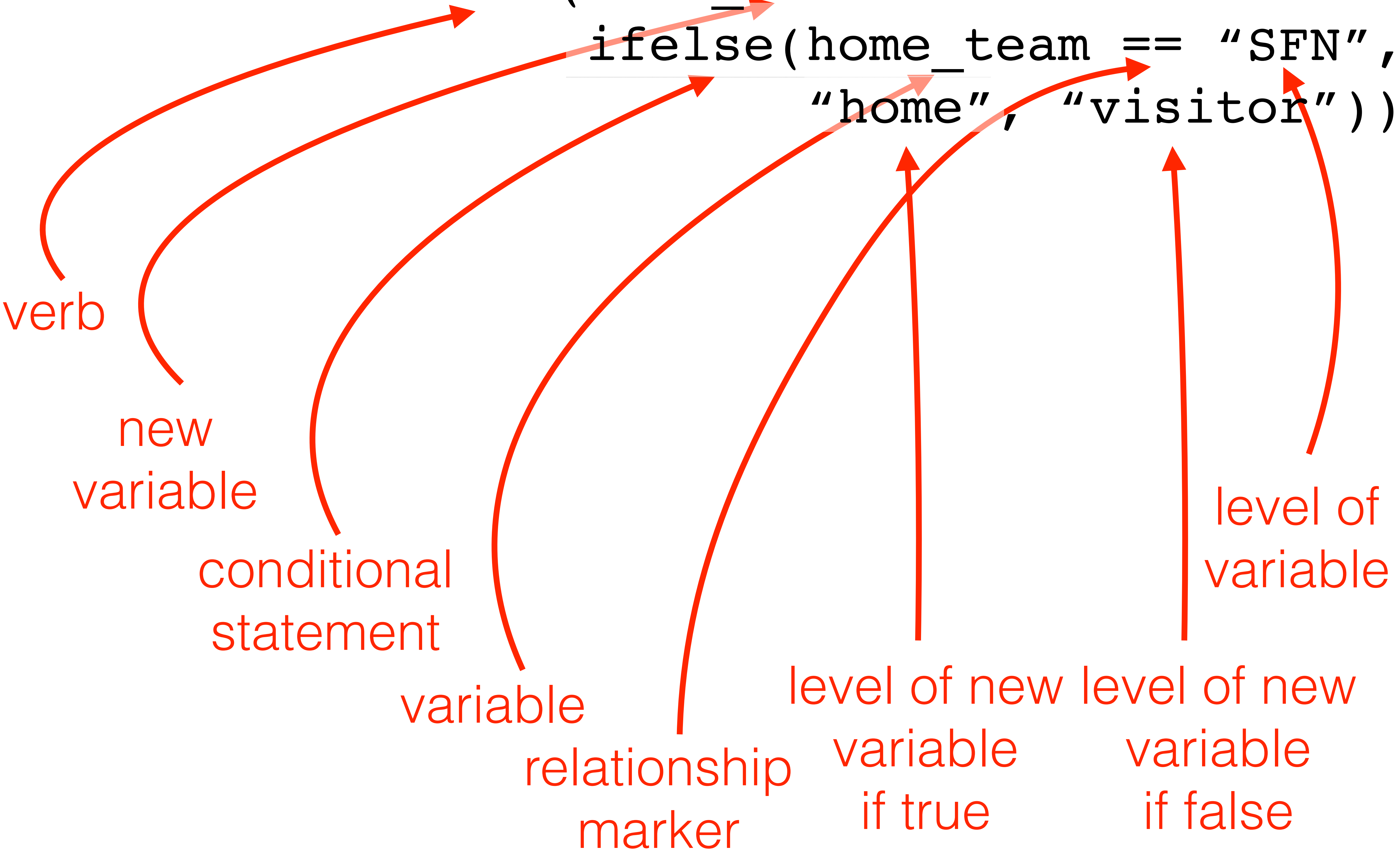
relationship marker

level of new variable if true

level of variable

dplyr

```
data_clean = data %>%
                  mutate(home_visitor =
                            ifelse(home_team == “SFN”,
                                      “home”, “visitor”))
```



verb

new variable

conditional statement

variable

relationship marker

level of new variable if true

level of new variable if false

level of variable

dplyr

```
data_posey_clean = data_posey
```

dplyr

```
data_posey_clean = data_posey %>%
```

dplyr

```
data_posey_clean = data_posey %>%
                    inner_join(          )
```

two table
verb

dplyr

```
data_posey_clean = data_posey %>%
                    inner_join(data_clean)
```

two table verb

data frame

dplyr

```
data_posey_clean = data_posey %>%
                     inner_join(data_clean)
```

two table verb

data frame

```
data_posey  +  data_clean  = data_posey_clean
```

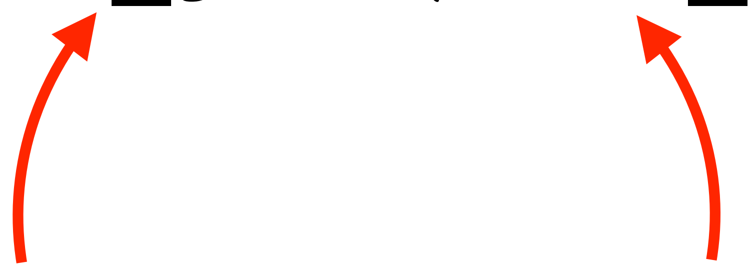
date	opponent
20100529	ARI
20100530	ARI
20100531	COL
20100601	COL

date	day_of_week
20100405	Mon
20100406	Tue
20100529	Sat
20100530	Sun

date	opponent	day_of_week
20100529	ARI	Sat
20100530	ARI	Sun

dplyr

```
data_posey_clean = data_posey %>%
                     inner_join(data_clean)
```



two table verb

data frame

data_posey + data_clean = data_posey_clean

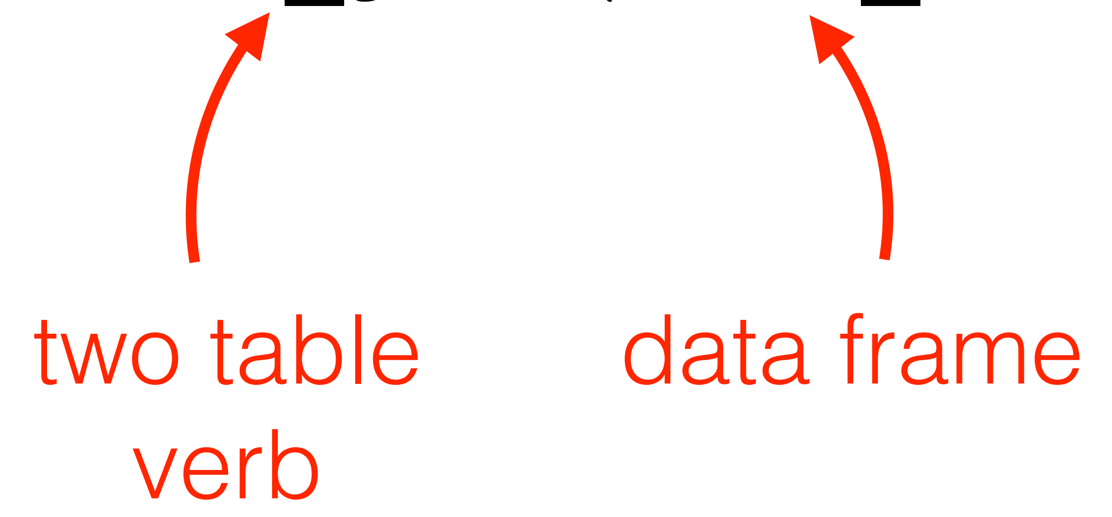
date	opponent
20100529	ARI
20100530	ARI
20100531	COL
20100601	COL

date	day_of_week
20100405	Mon
20100406	Tue
20100529	Sat
20100530	Sun

date	opponent	day_of_week
20100529	ARI	Sat
20100530	ARI	Sun

dplyr

```
data_posey_clean = data_posey %>%
                     inner_join(data_clean)
```



two table verb

data frame

```
data_posey    +    data_clean    =  data_posey_clean
```

date	at_bats
20100529	4
20100530	5
20100531	3
20100601	4

date	day_of_week
20100405	Mon
20100406	Tue
20100529	Sat
20100530	Sun

date	opponent	day_of_week
20100529	ARI	Sat
20100530	ARI	Sun

dplyr

```
data_figs_sum = data_figs
```

dplyr

```
data_figs_sum = data_figs %>%
```

dplyr

```
data_figs_sum = data_figs %>%
                group_by(             )
```

verb

dplyr

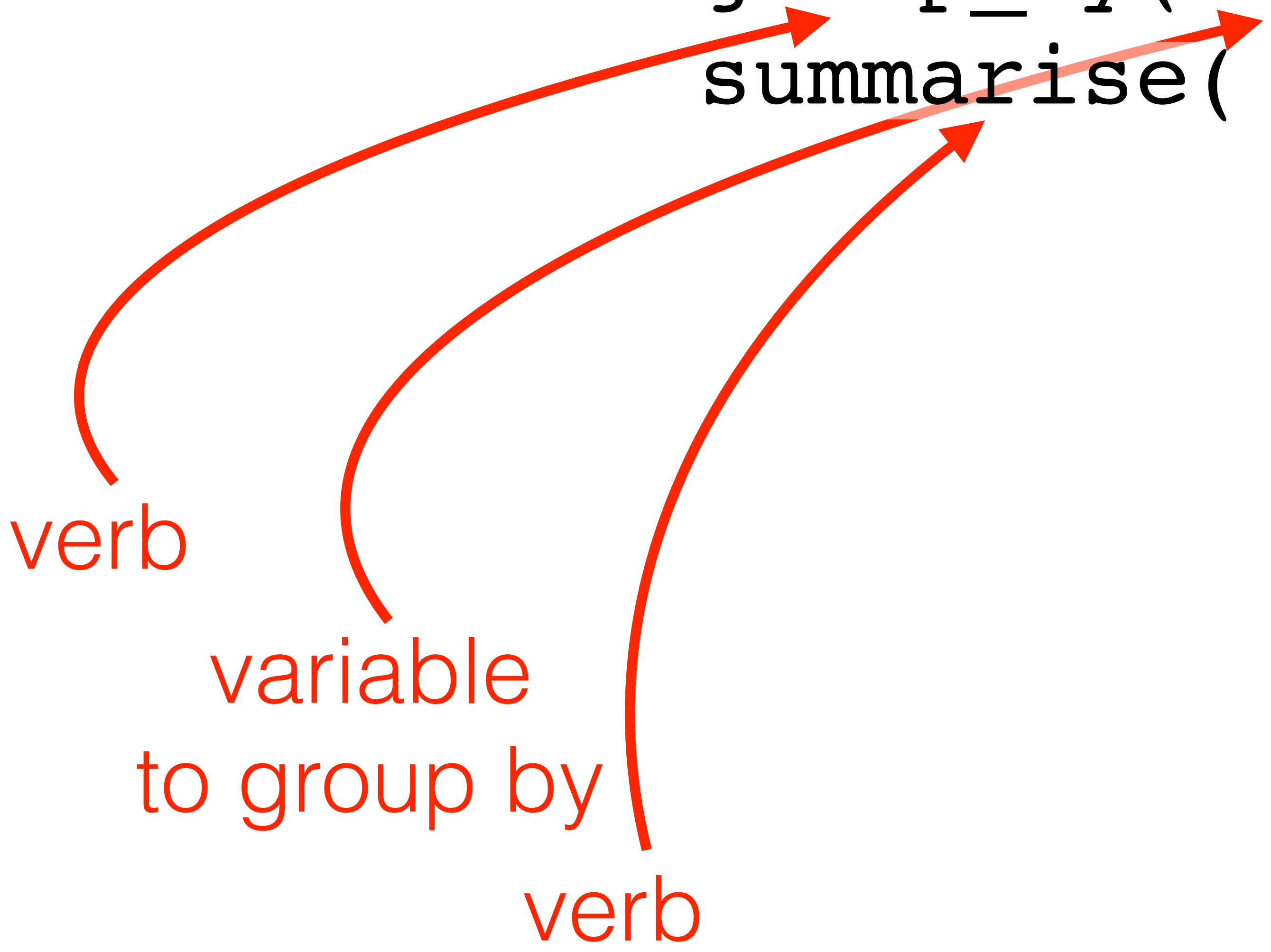
```
data_figs_sum = data_figs %>%
                group_by(allstar_break)
```

verb

variable
to group by

dplyr

```
data_figs_sum = data_figs %>%
                group_by(allstar_break) %>%
                summarise(
                                         )
```



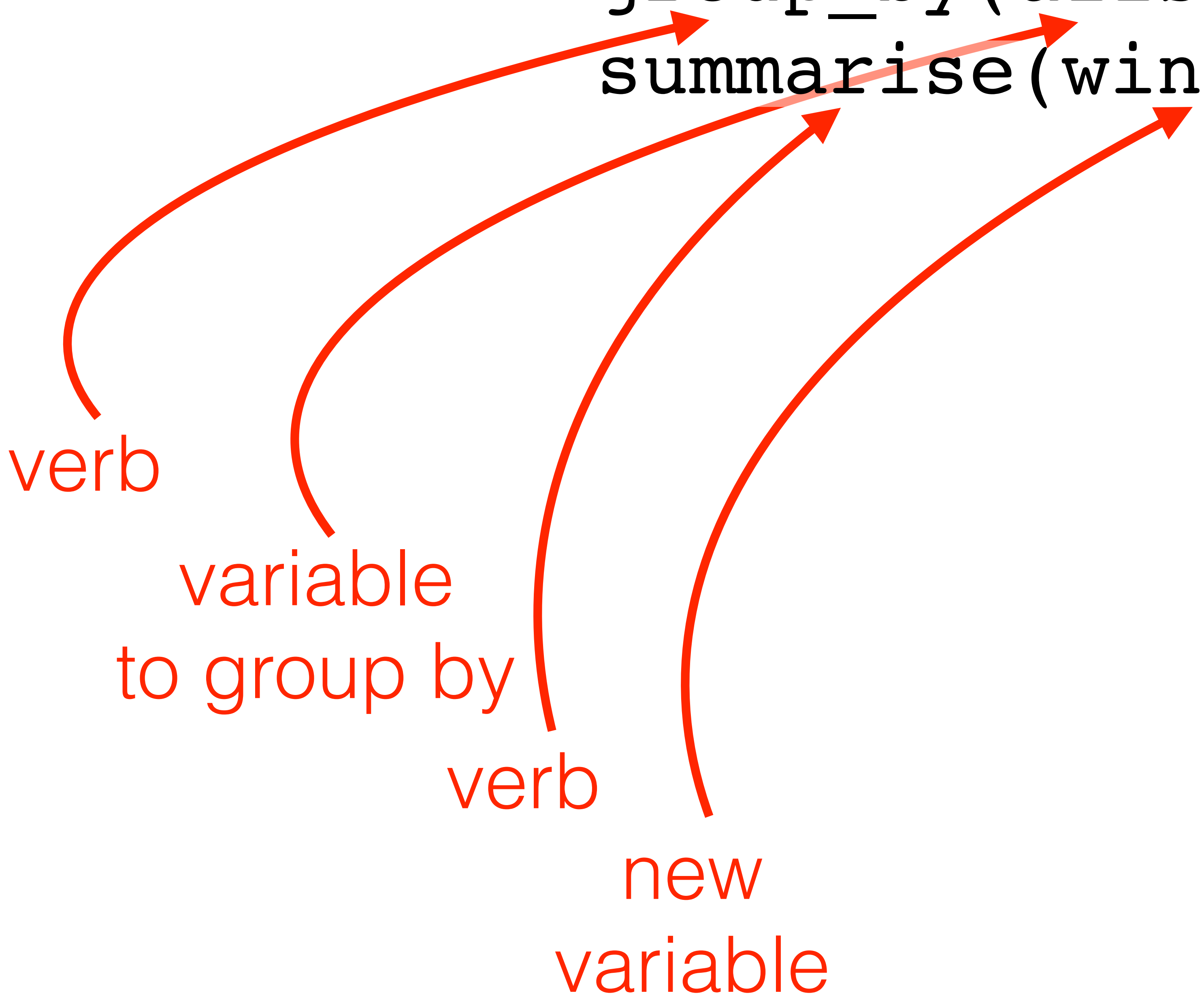
verb

variable
to group by

verb

dplyr

```
data_figs_sum = data_figs %>%
                group_by(allstar_break) %>%
                summarise(wins_perc
                                         )
```



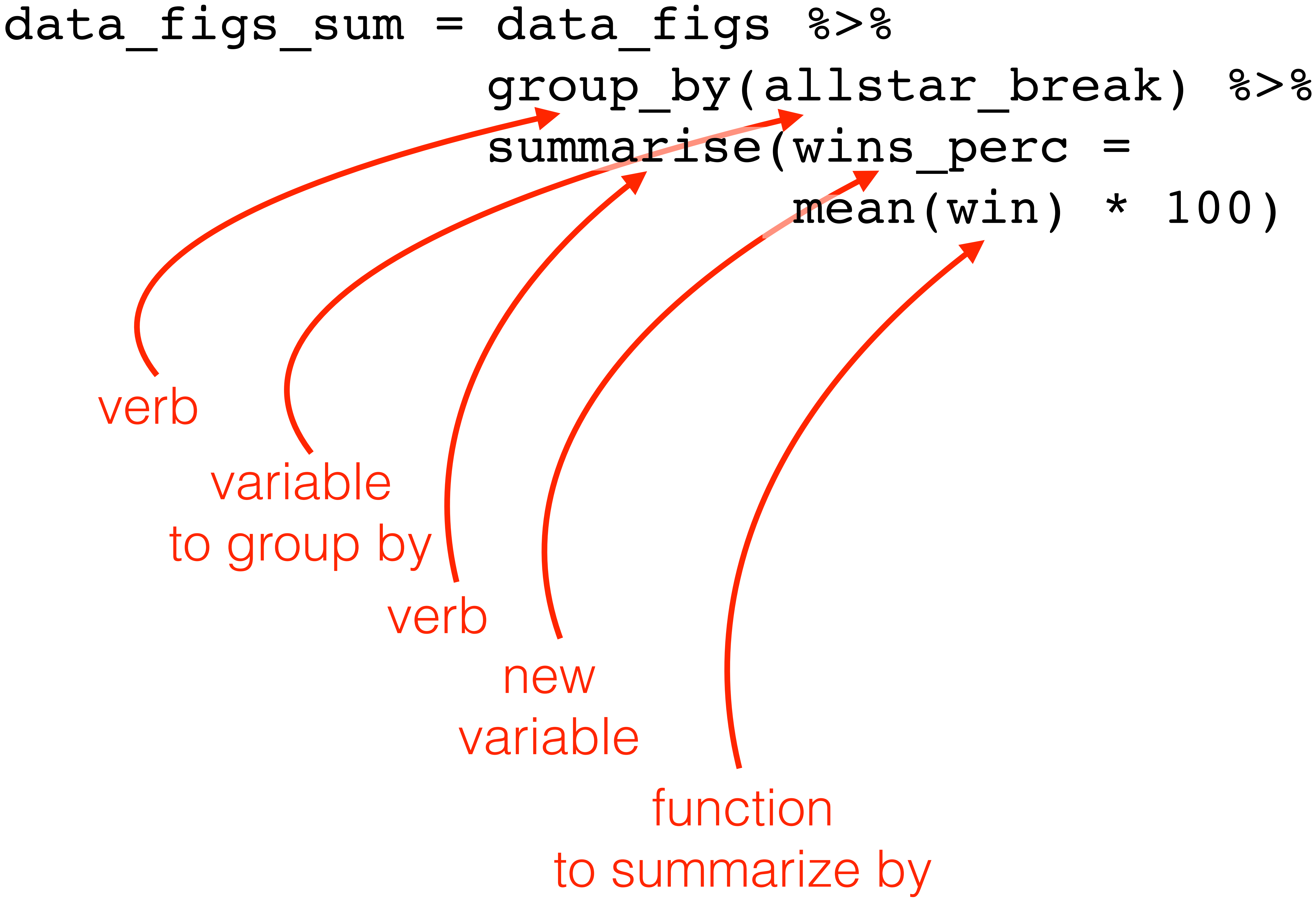
verb

variable
to group by

verb

new
variable

dplyr



```
data_figs_sum = data_figs %>%
                group_by(allstar_break) %>%
                summarise(wins_perc =
                          mean(win) * 100)
```

verb

variable
to group by

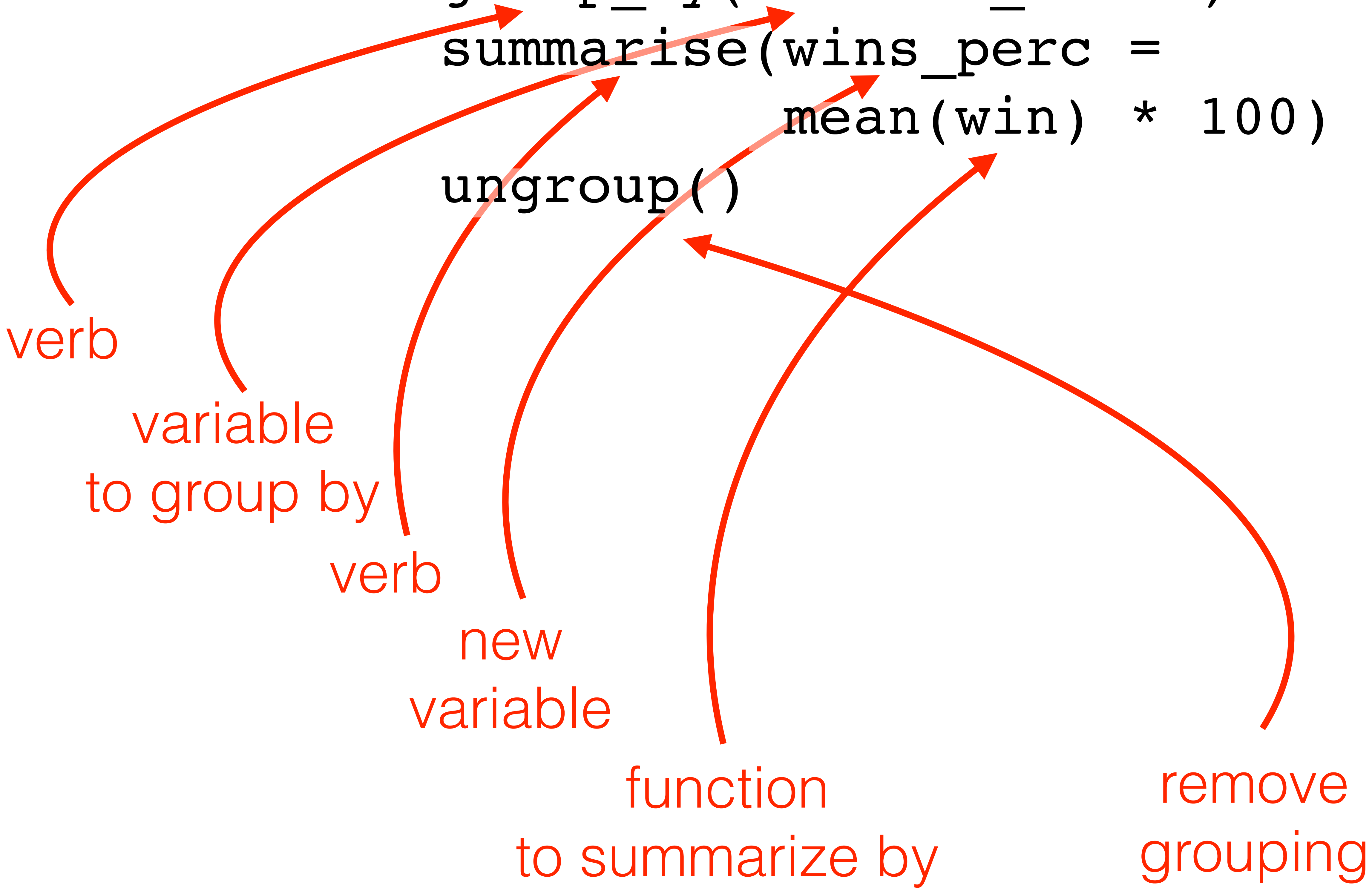
verb

new
variable

function
to summarize by

dplyr

```
data_figs_sum = data_figs %>%
                group_by(allstar_break) %>%
                summarise(wins_perc =
                          mean(win) * 100) %>%
                ungroup()
```



verb

variable
to group by

verb

new
variable

function
to summarize by

remove
grouping

ggplot2

```
allstar.plot = ggplot(data_figs_sum,
                      aes(x = allstar_break,
                          y = wins_perc))
```

ggplot2

```
allstar.plot = ggplot(data_figs_sum,
                      aes(x = allstar_break,
                          y = wins_perc)) +
```

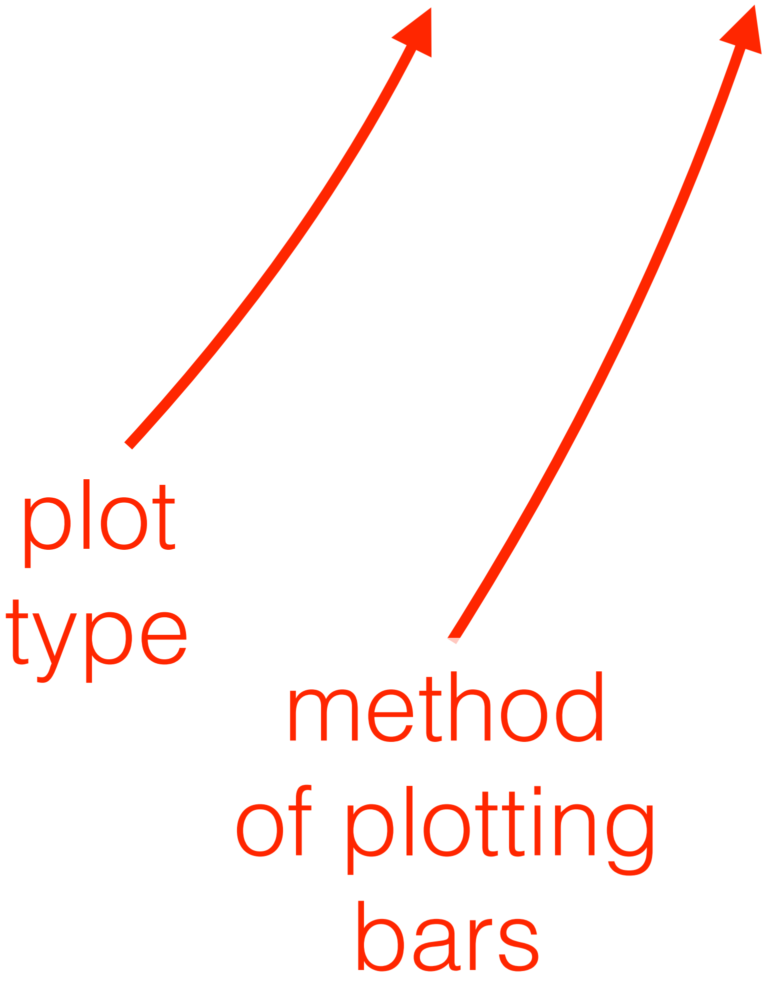
ggplot2

```
allstar.plot = ggplot(data_figs_sum,
                      aes(x = allstar_break,
                          y = wins_perc)) +
                geom_bar(                )
```

plot
type

ggplot2

```
allstar.plot = ggplot(data_figs_sum,
                      aes(x = allstar_break,
                          y = wins_perc)) +
                geom_bar(stat             )
```

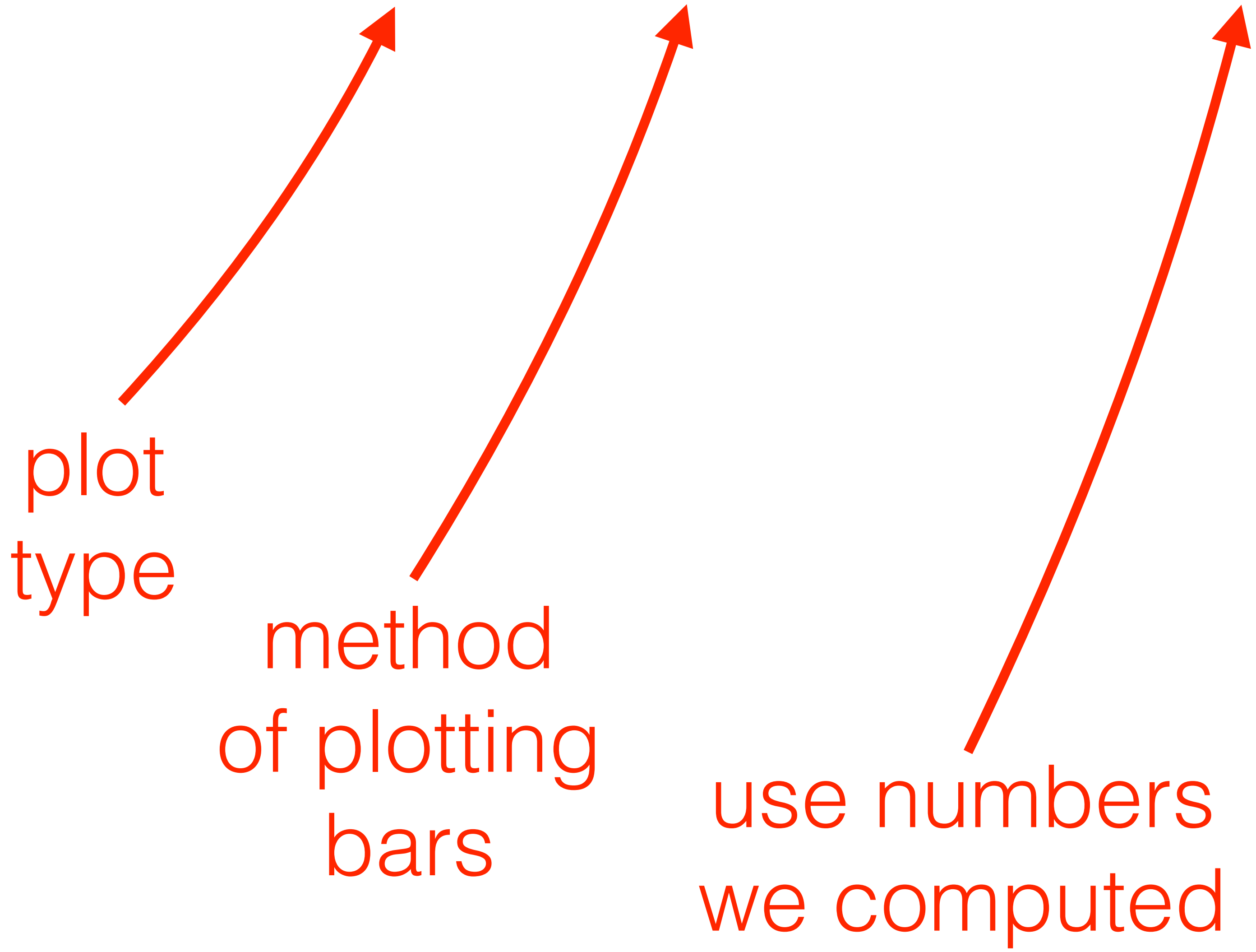


plot type

method of plotting bars

ggplot2

```
allstar.plot = ggplot(data_figs_sum,
                      aes(x = allstar_break,
                          y = wins_perc)) +
                geom_bar(stat = “identity”)
```

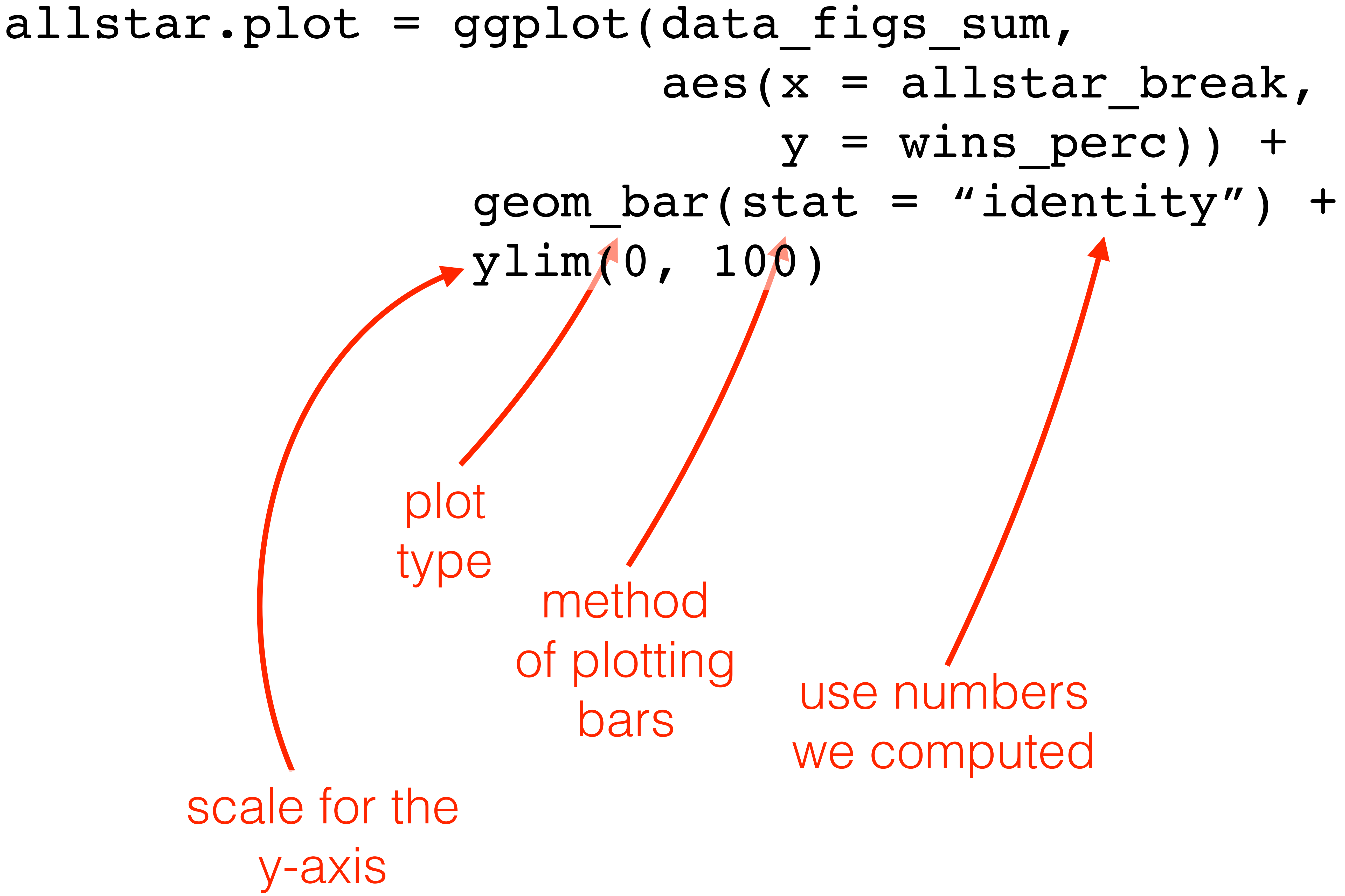


plot type

method of plotting bars

use numbers we computed

ggplot2



```
allstar.plot = ggplot(data_figs_sum,
                          aes(x = allstar_break,
                                 y = wins_perc)) +
               geom_bar(stat = “identity”) +
               ylim(0, 100)
```

plot type

method of plotting bars

use numbers we computed

scale for the y-axis