Introduction to R & RStudio:

deck 10: Modeling

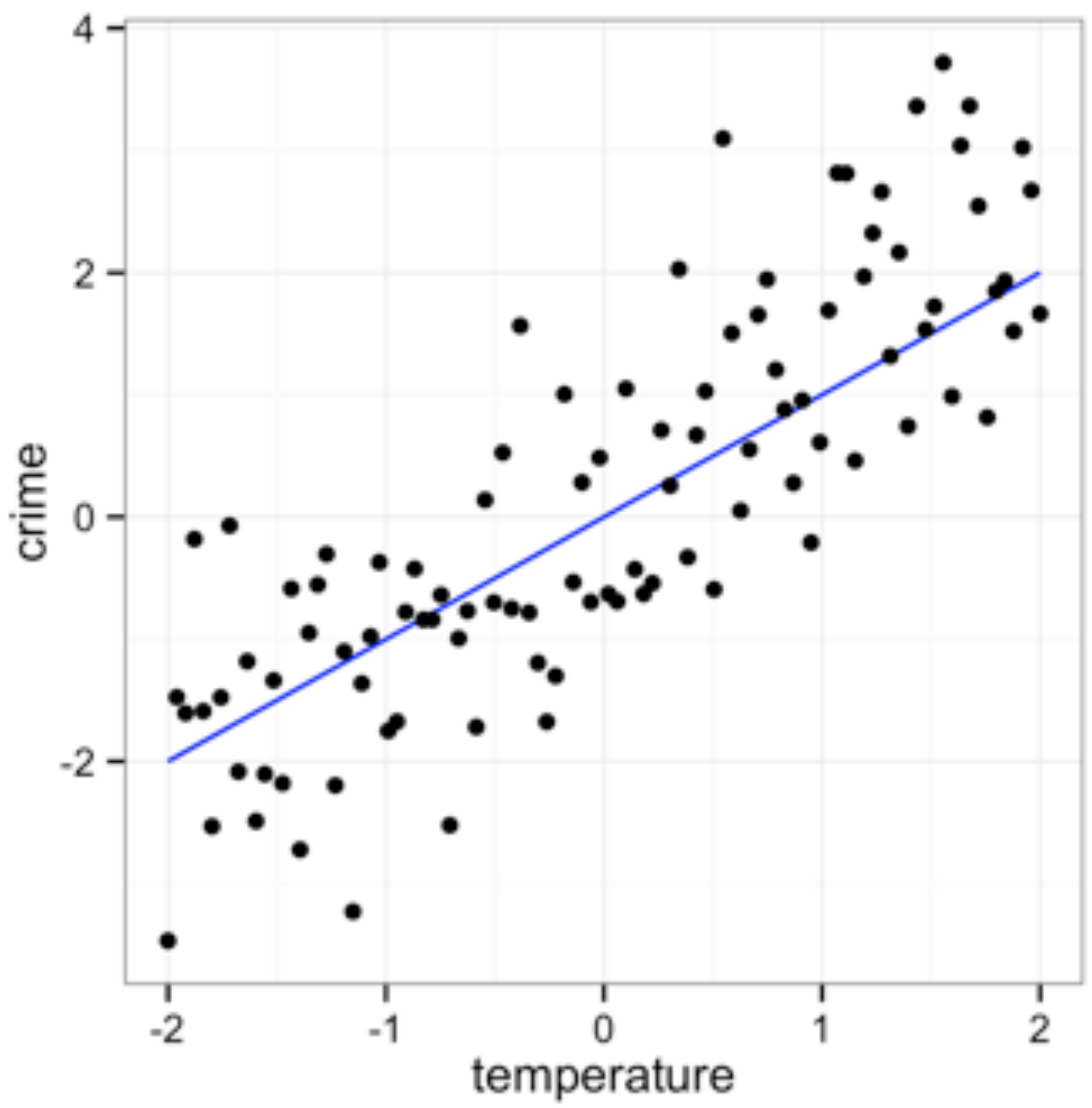
Amelia McNamara
Visiting Assistant Professor of Statistical and Data Sciences
Smith College

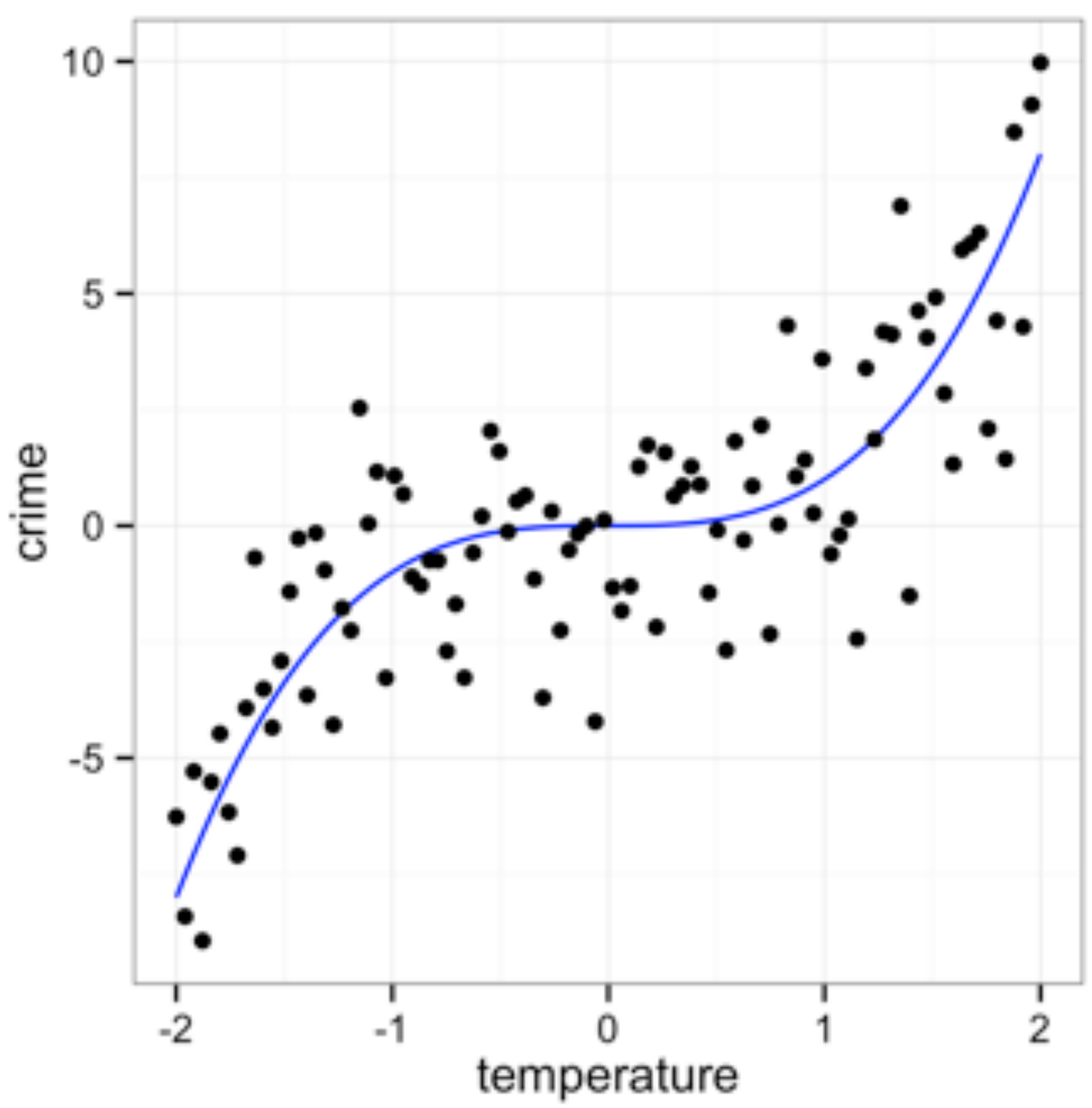
January 2018

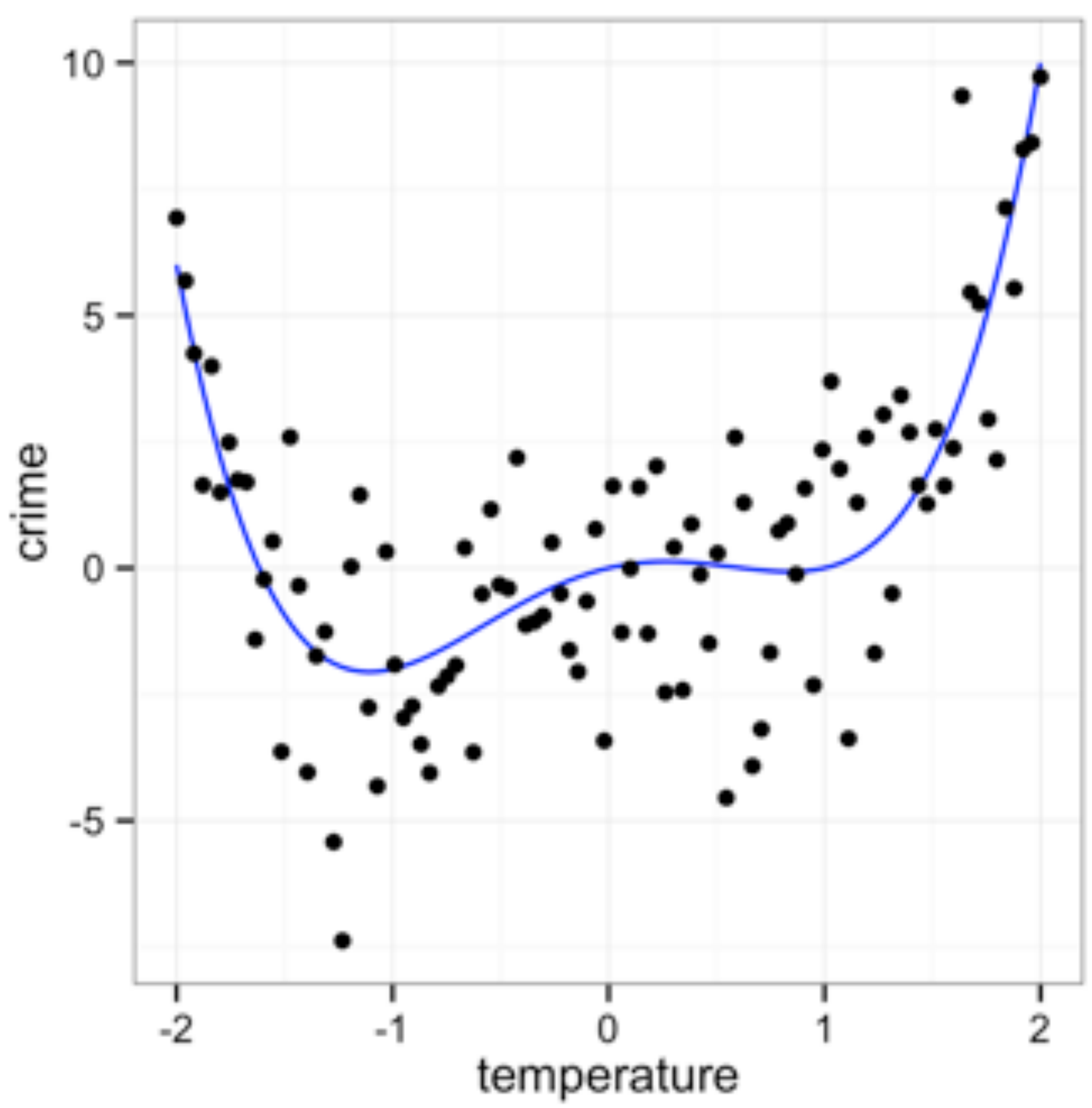
The basics

Models

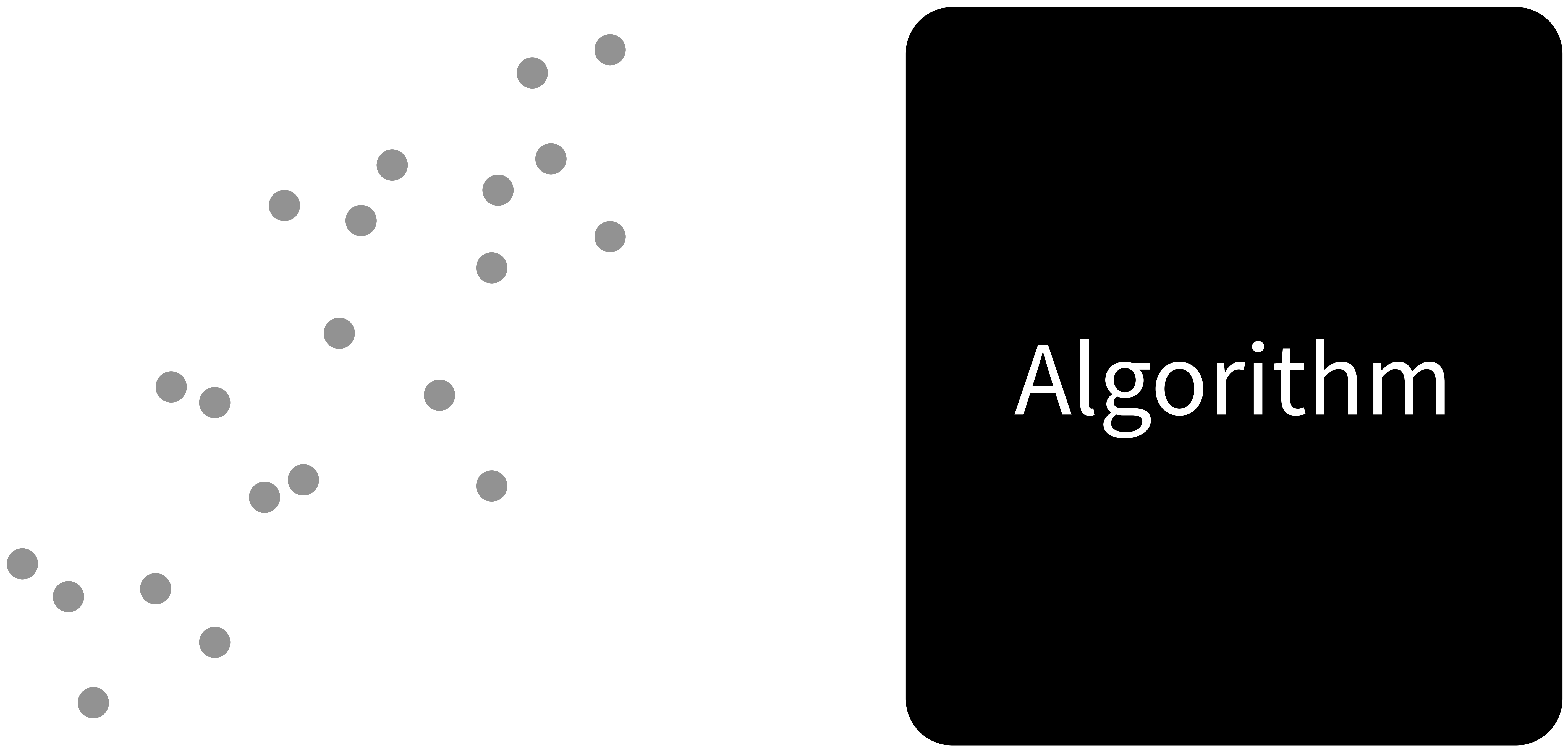
A low dimensional description of a higher dimensional data set.







Models



Data

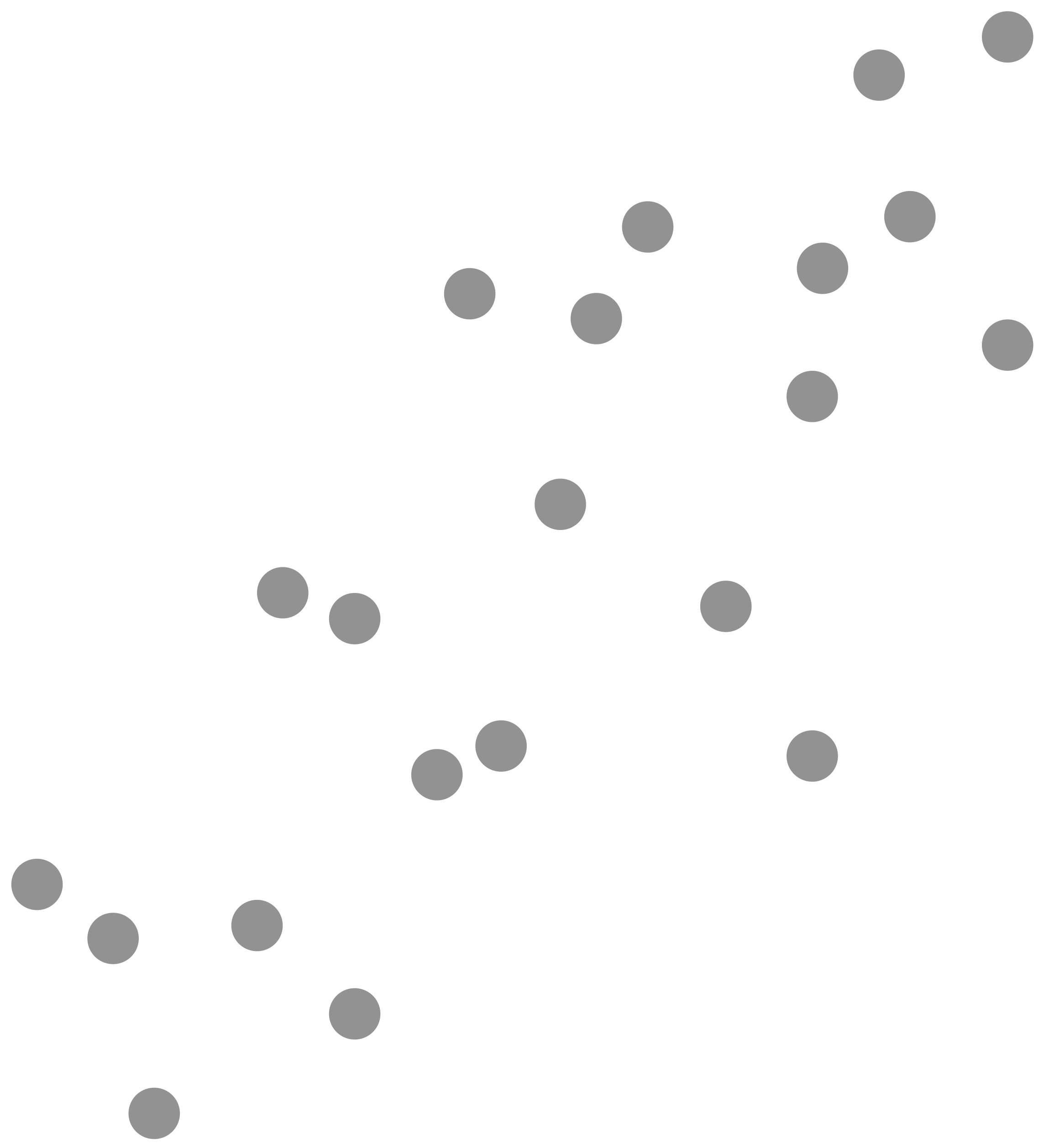
Model Function

Models

What is the **model function**?

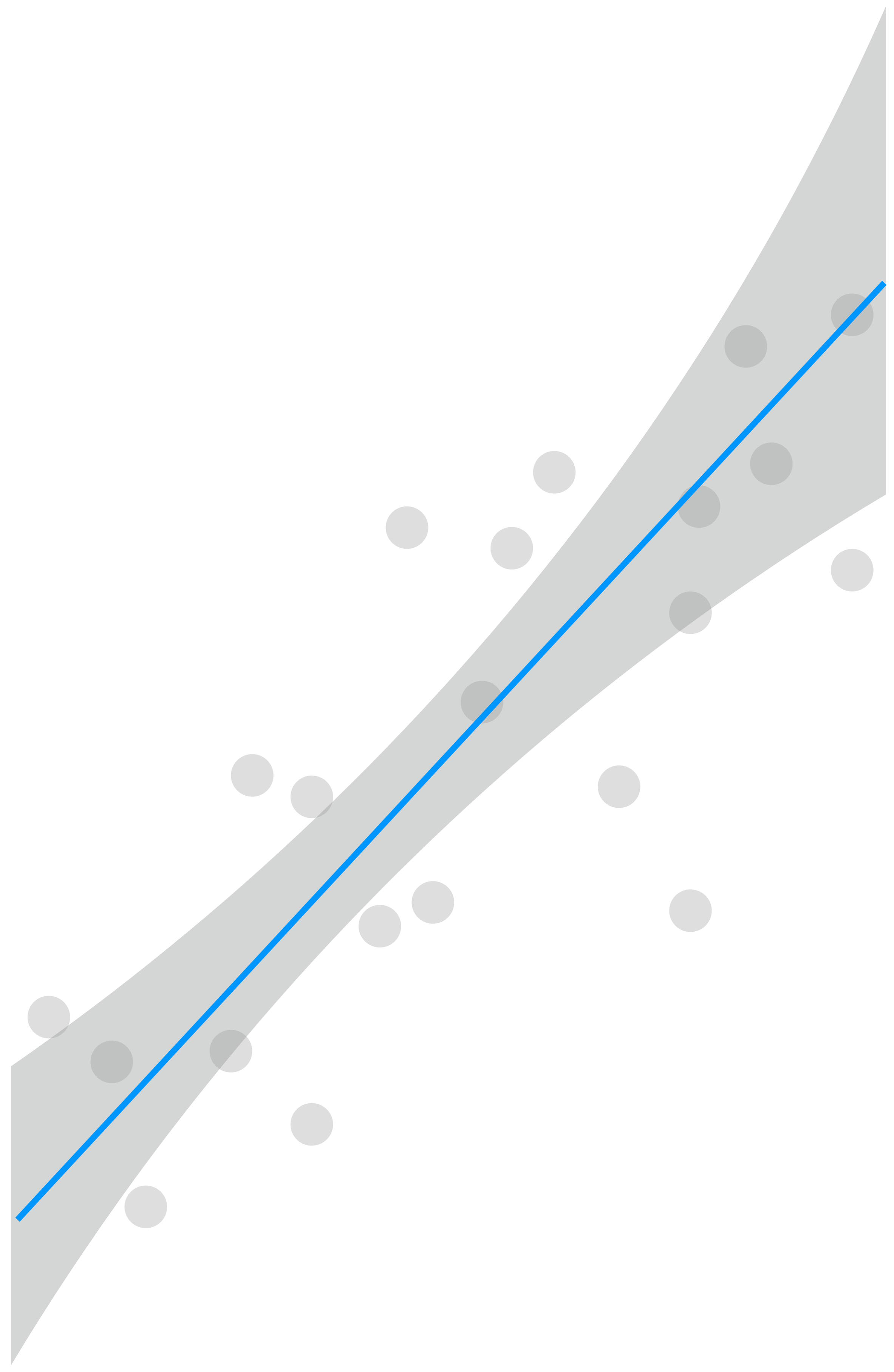
Models

What uncertainty is associated with it?



Data

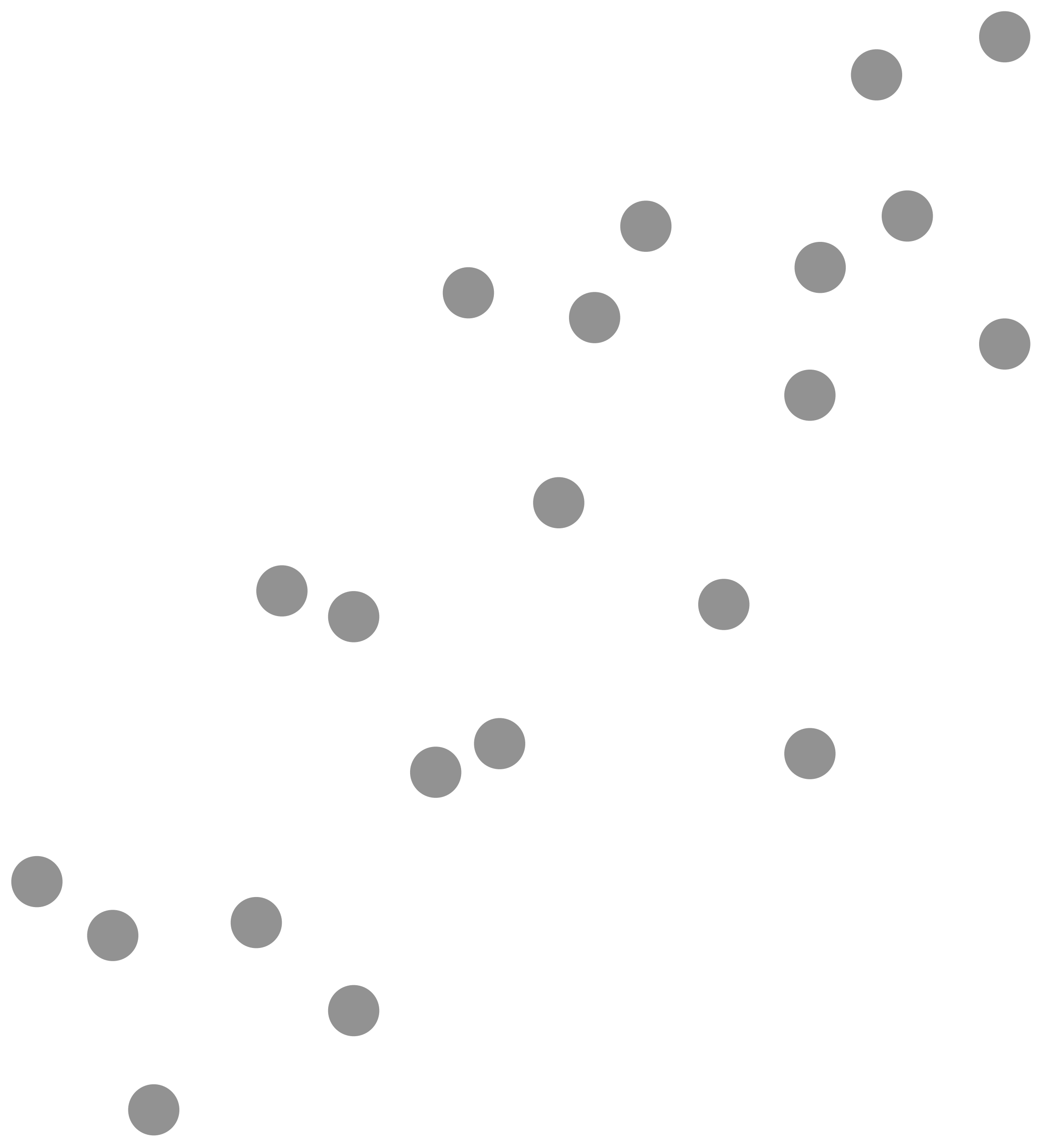
Algorithm



Model Function

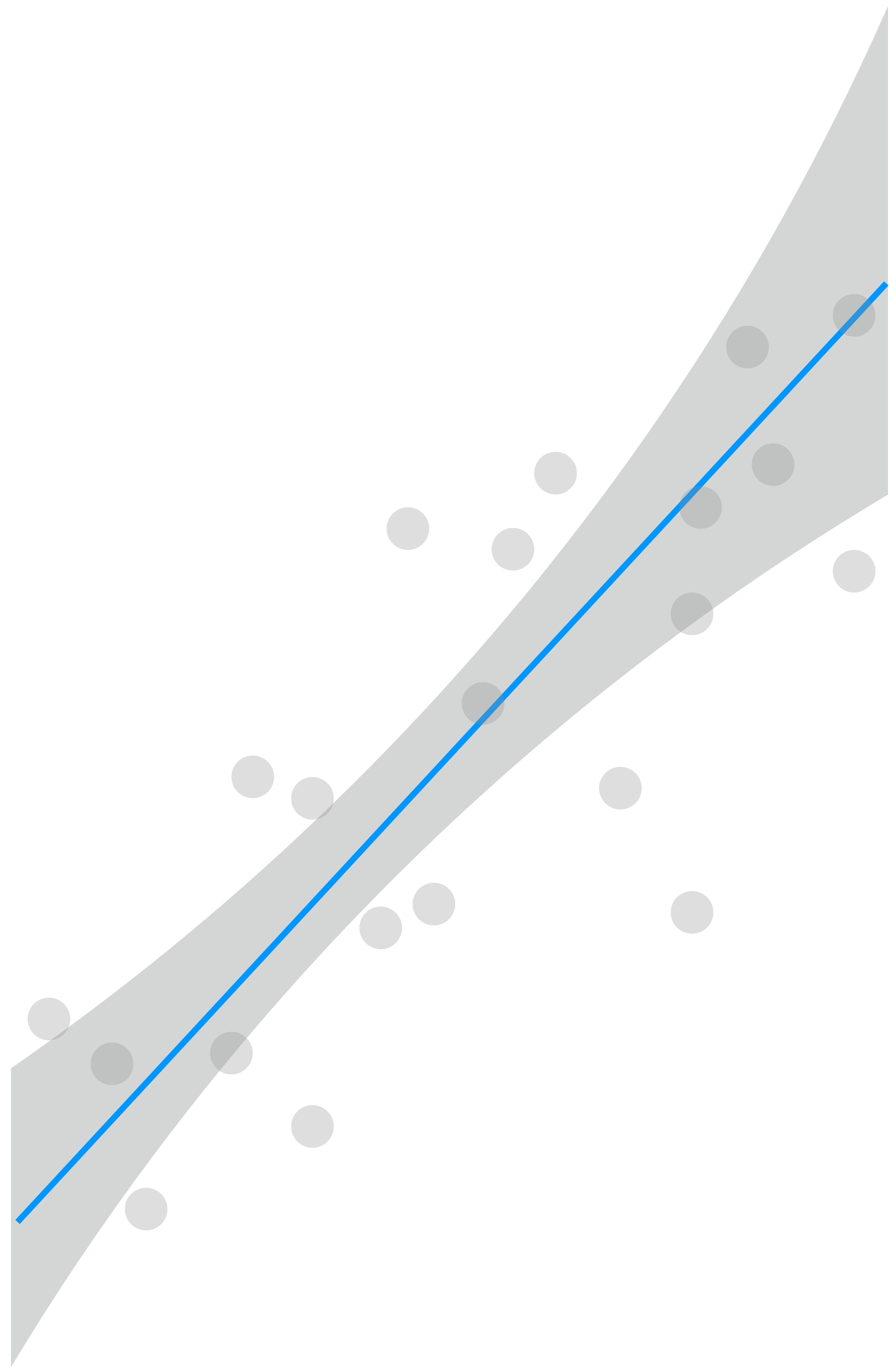
Models

How "good" is the model?



Data

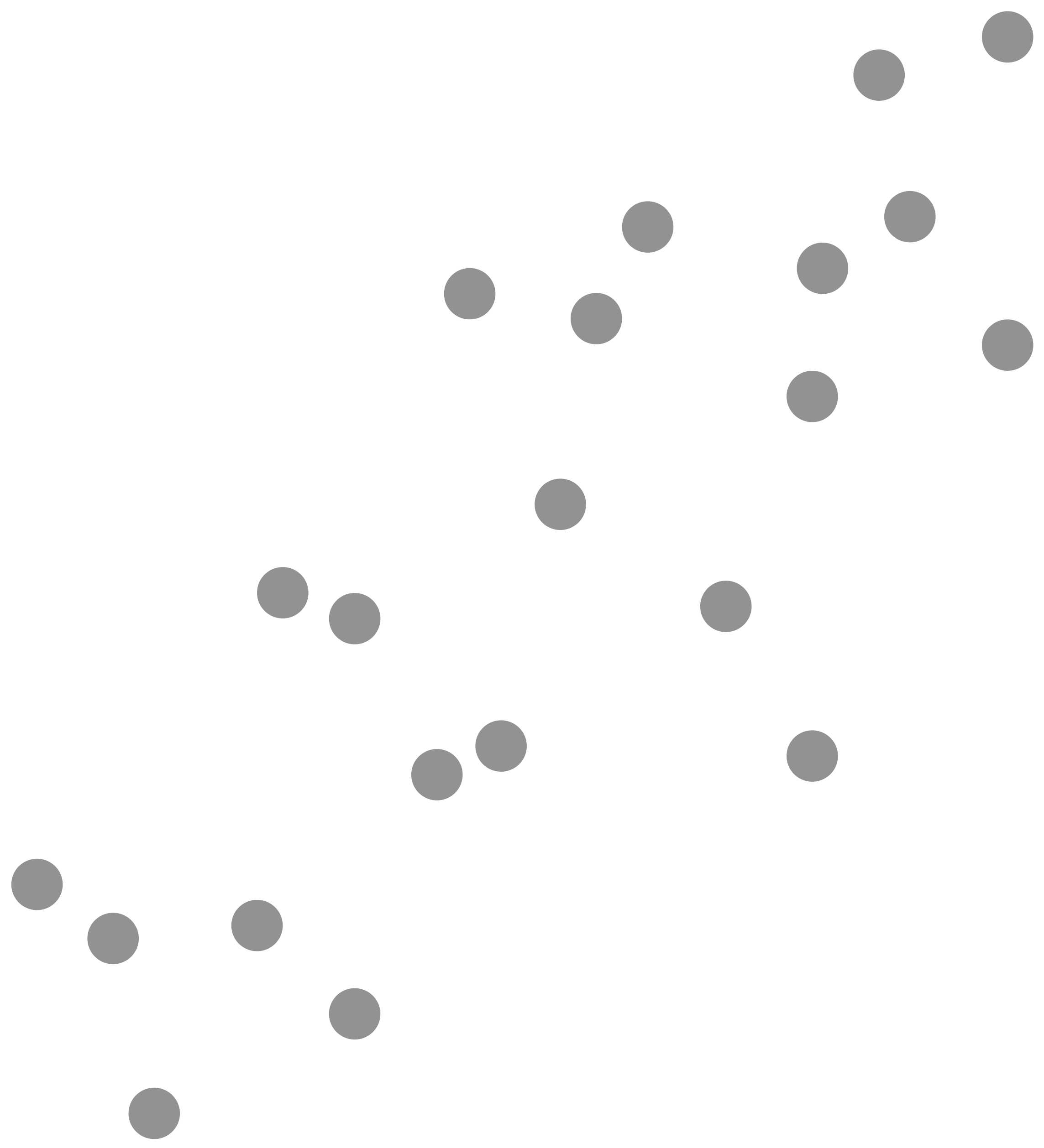
Algorithm



Model Function

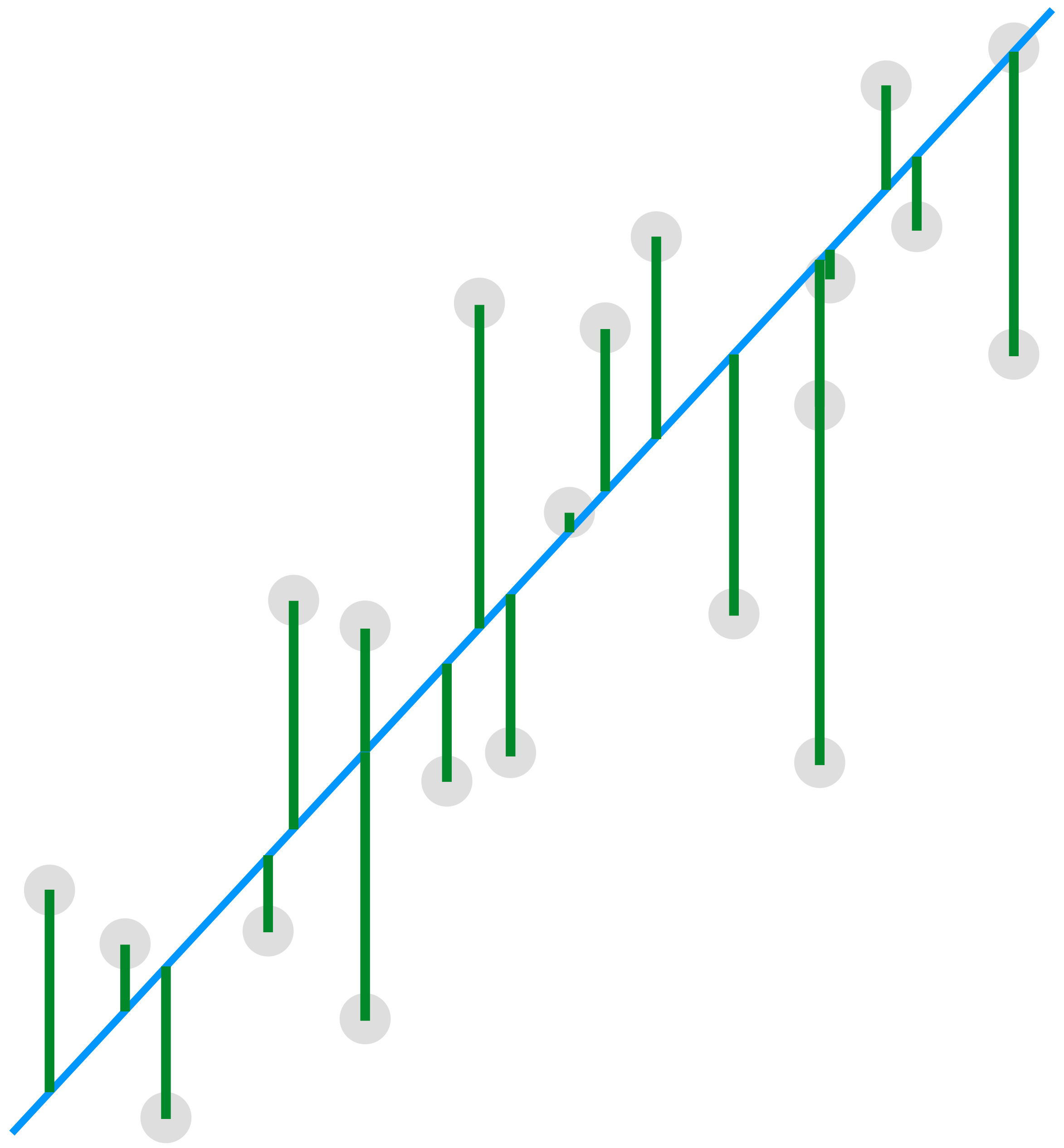
Models

What are the residuals?



Data

Algorithm



Model Function

Models

What are the predictions?

(Popular) **modeling functions in R**

function	**package**	**fits**
lm()	stats	linear models
glm()	stats	generalized linear models
gam()	mgcv	generalized additive models
glmnet()	glmnet	penalized linear models
rlm()	MASS	robust linear models
rpart()	rpart	trees
randomForest()	randomForest	random forests
xgboost()	xgboost	gradient boosting machines

(Popular) modeling functions in R

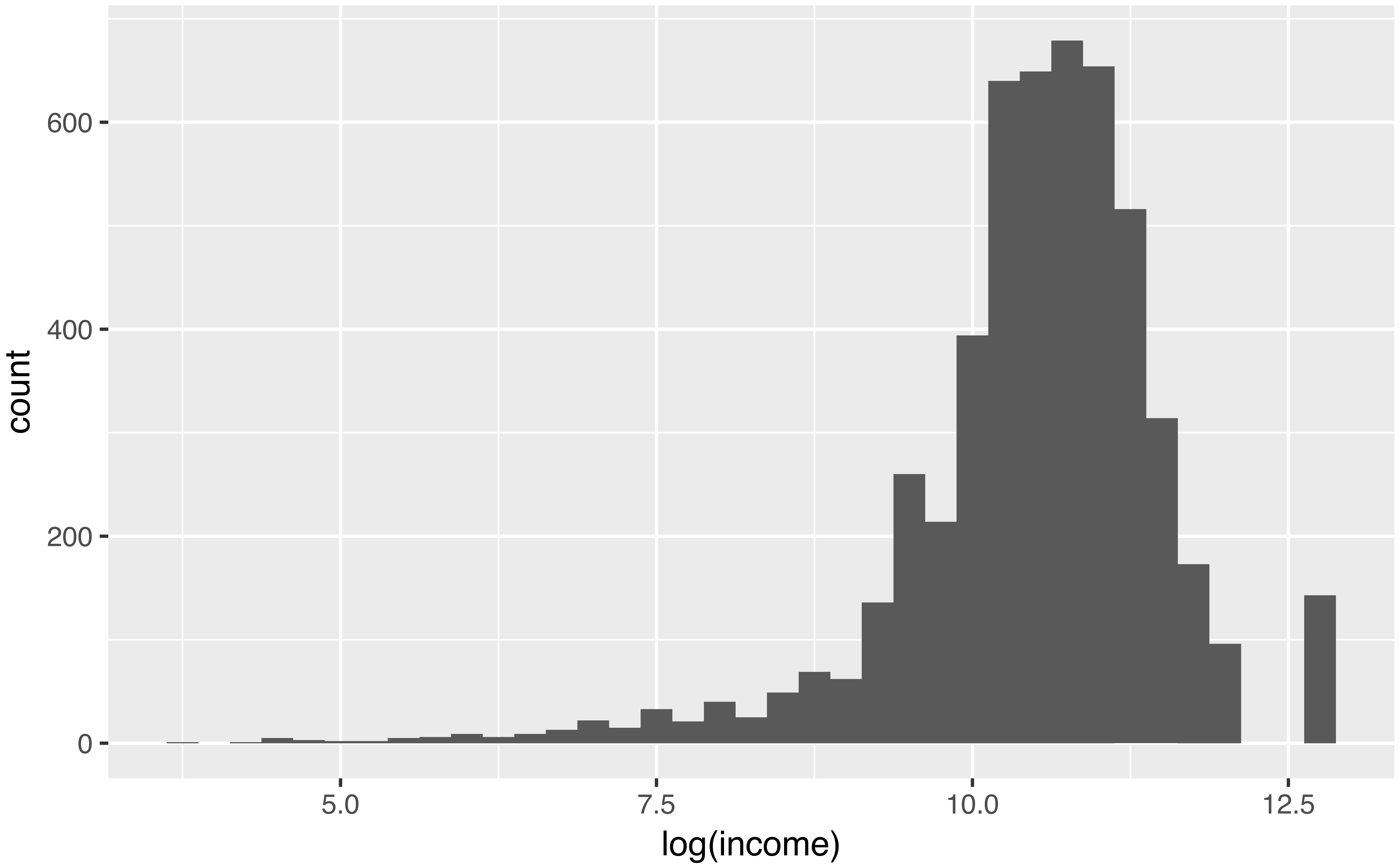
function	package	fits
lm()	**stats**	**linear models**
glm()	stats	generalized linear models
gam()	mgcv	generalized additive models
glmnet()	glmnet	penalized linear models
rlm()	MASS	robust linear models
rpart()	rpart	trees
randomForest()	randomForest	random forests
xgboost()	xgboost	gradient boosting machines

```
wages
```

income <int>	height <dbl>	weight <int>	age <int>	marital <fctr>	sex <fctr>	education <int>	afqt <dbl>
19000	60	155	53	married	female	13	6.841
35000	70	156	51	married	female	10	49.444
105000	65	195	52	married	male	16	99.393
40000	63	197	54	married	female	14	44.022
75000	66	190	49	married	male	14	59.683
102000	68	200	49	divorced	female	18	98.798
0	74	225	48	married	male	16	82.260
70000	64	160	54	divorced	female	12	50.283
60000	69	162	55	divorced	male	12	89.669
150000	69	194	54	divorced	male	13	95.977

1–10 of 7,006 rows Previous 1 2 3 4 5 6 ... 100 Next

```r
wages %>%
  ggplot(aes(log(income)) + geom_histogram(binwidth = 0.25)
```



lm()

lm()

Fit a linear model to data

```
lm(log(income) ~ education, data = wages)
```

A formula that describes the model equation

The data set

formulas

Formula only needs to include the response and predictors

$$y = \alpha + \beta x + \epsilon$$

y ~ x

Your Turn

Fit the model below and then examine the output. What does it look like?

```
mod_e <- lm(log(income) ~ education, data = wages)
```

02:00

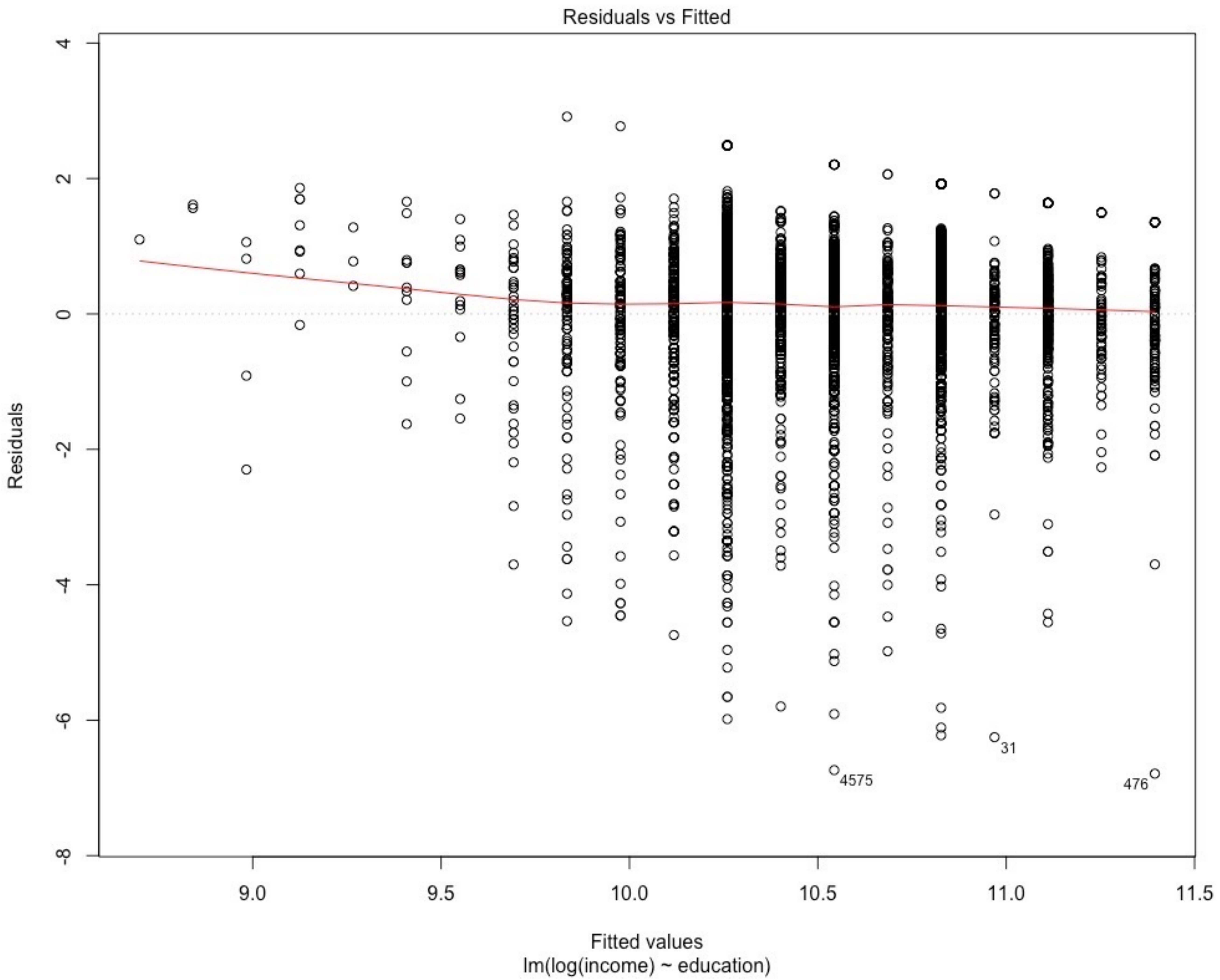
```r
mod_e <- lm(log(income) ~ education, data = wages)
mod_e
##  Call:
##  lm(formula = log(income) ~ education, data = wages)
##
## Coefficients:
## (Intercept)    education
##      8.5577       0.1418
class(mod_e)
## "lm"
```

```
summary(mod_e)
Call:
lm(formula = log(income) ~ education, data = wages)

Residuals:
    Min      1Q  Median      3Q     Max
-6.7893 -0.3563  0.1328  0.5798  2.9136

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.557691   0.073260  116.81   <2e-16 ***
education   0.141840   0.005305   26.74   <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9923 on 5262 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.1196,Adjusted R-squared:  0.1195
F-statistic:   715 on 1 and 5262 DF,  p-value: < 2.2e-16
```

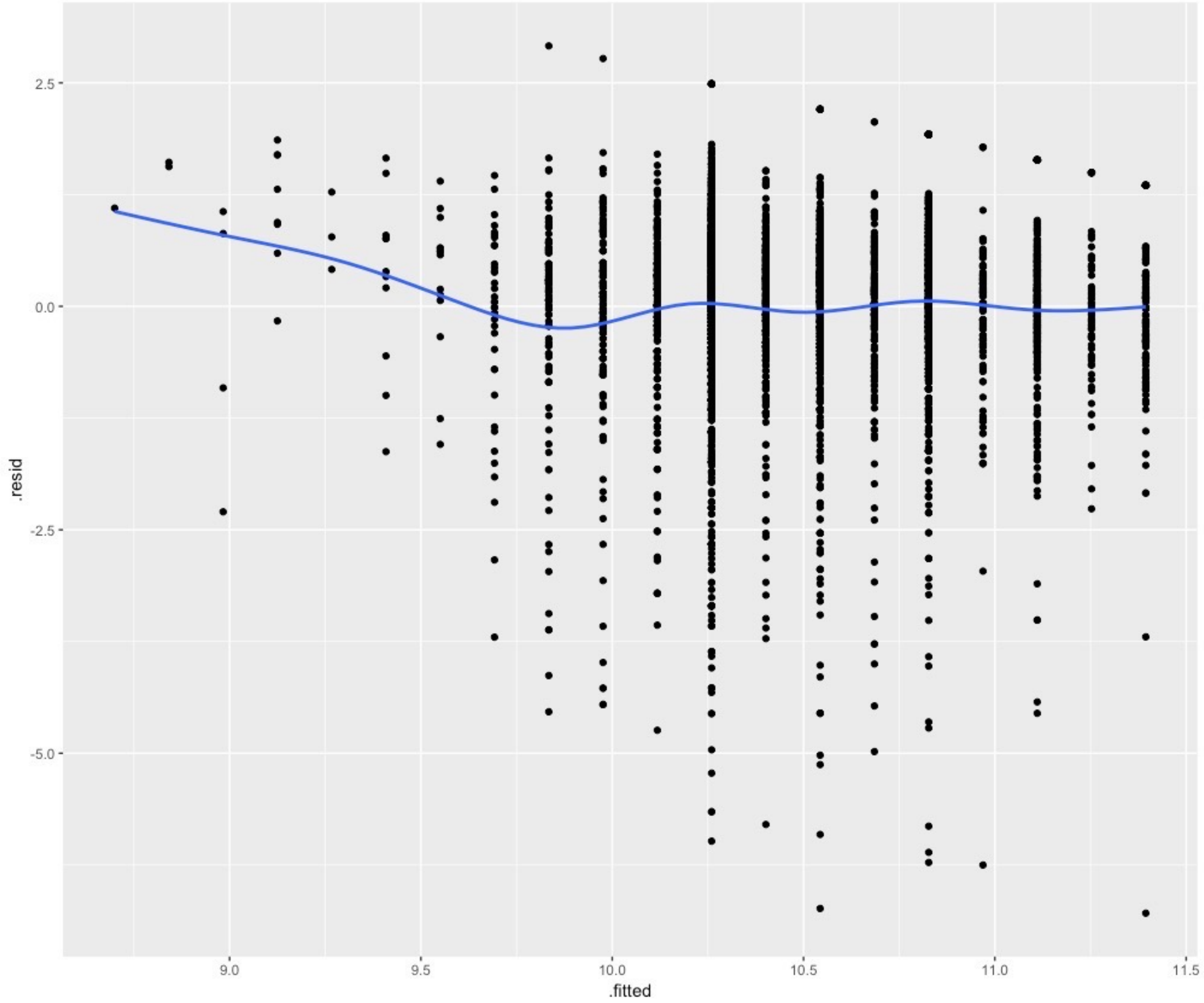
```
names(mod_e)
 [1] "coefficients"  "residuals"     "effects"
 [4] "rank"          "fitted.values" "assign"
 [7] "qr"            "df.residual"   "na.action"
[10] "xlevels"       "call"          "terms"
[13] "model"
mod_e$coefficients
(Intercept)   education
  8.5576906   0.1418404
```

Plotting models

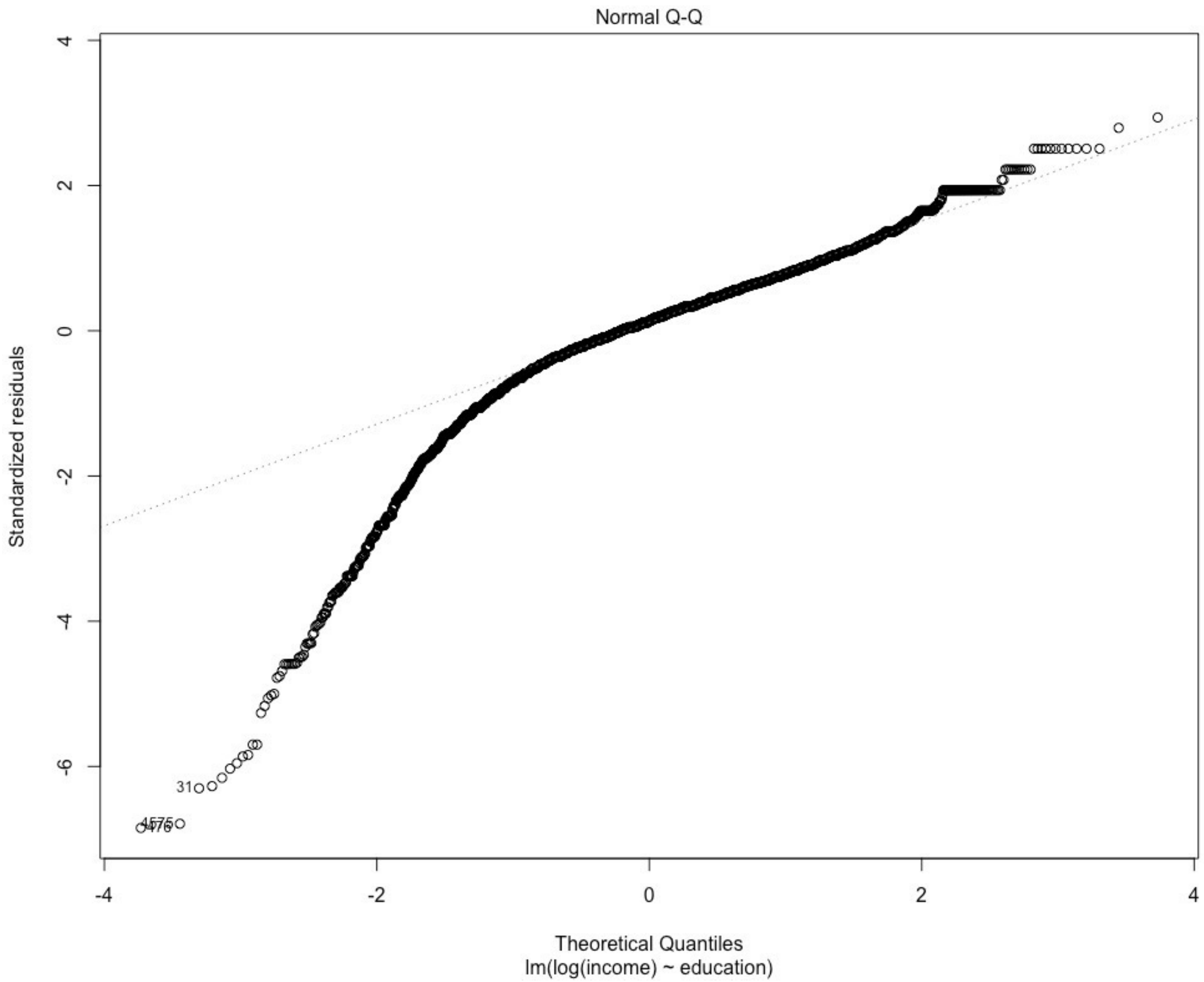
```
plot(mod_e, which=1)
```



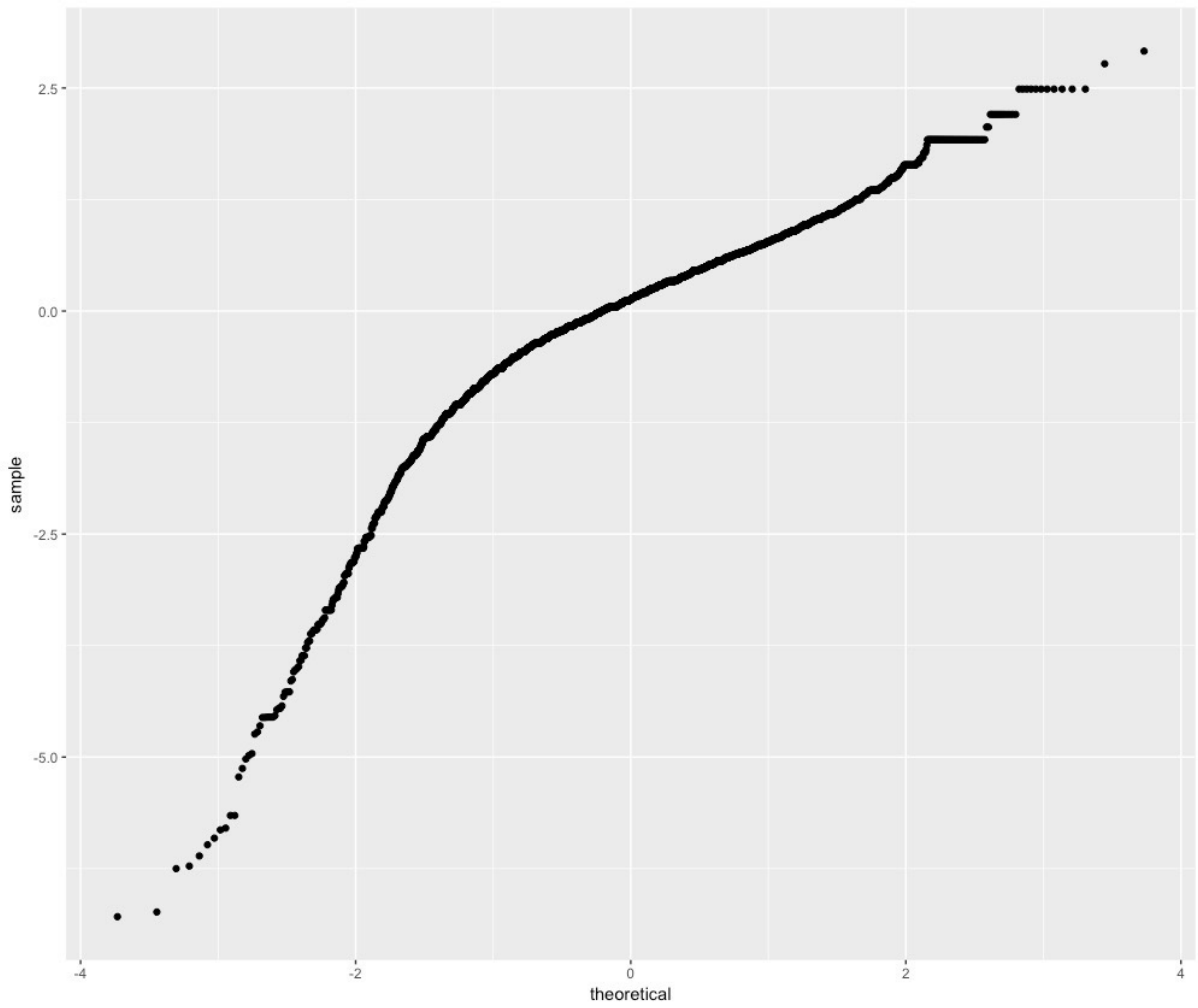
```
ggplot(mod_e, aes(x=.fitted, y=.resid)) + geom_point() +
geom_smooth(se = FALSE)
```



```
plot(mod_e, which=2)
```



```
ggplot(mod_e, aes(sample = .resid)) + geom_qq()
```



broom

broom

Turns model output into data frames

```r
# install.packages("tidyverse")
library(broom)
```

broom

Broom includes three functions which work for most types of models (and can be extended to more):

1. **tidy()** - returns model coefficients, stats
2. **glance()** - returns model diagnostics
3. **augment()** - returns predictions, residuals, and other raw values

tidy()

Returns useful **model output** as a data frame

```
mod_e %>% tidy()
```

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	8.5576906	0.073259622	116.81320	0.000000e+00
education	0.1418404	0.005304577	26.73924	8.408952e-148

2 rows

glance

Returns common **model diagnostics** as a data frame

```
mod_e %>% glance()
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int	
0.1196233	0.119456	0.9923358	714.987	8.408952e-148	2	-

1 row | 1-10 of 11 columns

augment()

Returns data frame of **model output related to original data points**

```
mod_e %>% augment()
```

.rownames <chr>	log.income. <dbl>	education <int>	.fitted <dbl>	.se.fit <dbl>	.resid <dbl>	.hat <dbl>	.sigma <dbl>	.c
1	9.852194	13	10.401615	0.01400504	-0.549421141	0.0001991827	0.9924012	3.05413
2	10.463103	10	9.976094	0.02335067	0.487009048	0.0005537086	0.9924074	6.67558
3	11.561716	16	10.827137	0.01880219	0.734579123	0.0003590043	0.9923784	9.84331
4	10.596635	14	10.543456	0.01386811	0.053178965	0.0001953068	0.9924299	2.80556
5	11.225243	14	10.543456	0.01386811	0.681787624	0.0001953068	0.9923856	4.61145
6	11.532728	18	11.110817	0.02719979	0.421910848	0.0007513008	0.9924131	6.80081
7	11.156251	12	10.259775	0.01600734	0.896475490	0.0002602083	0.9923532	1.06237
8	11.002100	12	10.259775	0.01600734	0.742324811	0.0002602083	0.9923774	7.28429
9	11.918391	13	10.401615	0.01400504	1.516775174	0.0001991827	0.9922098	2.32766
10	11.652687	16	10.827137	0.01880219	0.825550901	0.0003590043	0.9923648	1.24323

augment()

Returns data frame of **model output related to original data points**

```
mod_e %>% augment(data = wages)
```

Adds the original wages data set to the output

Your Turn

Model **log(income)** against **height**. Then use broom and dplyr functions to extract:

1. The coefficient estimates and their related statistics

2. The adj.r.squared and p.value for the overall model

05:00

multivariate regression

To fit multiple predictors,
add multiple variables to the formula:

```
log(income) ~ education + height
```

Your Turn

Model **log(income)** against **education** *and* **height**. Do the coefficients change?

03:00

```r
mod_eh <- lm(log(income) ~ education + height, data =
wages)


mod_eh %>%
  tidy()
##          term   estimate   std.error statistic       p.value
## 1 (Intercept) 5.34837618 0.231320415  23.12107 1.002503e-112
## 2   education 0.13871285 0.005205245  26.64867 7.120134e-147
## 3      height 0.04830864 0.003309870  14.59533  2.504935e-47
```

Your Turn

Model **log(income)** against **education** and **height** and **sex**. Can you interpret the coefficients?

03:00

```r
mod_ehs <- lm(log(income) ~ education + height + sex, data =
wages)

mod_ehs %>%
  tidy()
## 		term 	estimate 	std.error 	statistic 	p.value
## 1 (Intercept)  8.250422260 0.334703051  24.649976 4.681336e-127
## 2   education  0.147983063 0.005196676  28.476486 5.164290e-166
## 3      height  0.006726614 0.004792698   1.403513  1.605229e-01
## 4   sexfemale -0.461747002 0.038941592 -11.857425  5.022841e-32
```

What does this mean?

Where is sexmale?

```
##          term     estimate   std.error  statistic       p.value
## 1 (Intercept)  8.250422260 0.334703051  24.649976 4.681336e-127
## 2   education  0.147983063 0.005196676  28.476486 5.164290e-166
## 3      height  0.006726614 0.004792698   1.403513  1.605229e-01
## 4   sexfemale -0.461747002 0.038941592 -11.857425  5.022841e-32
```

For factors, R treats the first level as the baseline level, e.g. the mean log(income) for a male is:

log(income) = 8.25 + 0.15 * education + 0 * height

Each additional level gets a coefficient that acts as an *adjustment* between the baseline level and the additional level, e.g. the mean income for a female is:

log(income) = 8.25 + 0.15 * education + 0 * height - 0.46

```
##          term     estimate   std.error  statistic       p.value
## 1 (Intercept)  8.250422260 0.334703051  24.649976 4.681336e-127
## 2   education  0.147983063 0.005196676  28.476486 5.164290e-166
## 3      height  0.006726614 0.004792698   1.403513  1.605229e-01
## 4   sexfemale -0.461747002 0.038941592 -11.857425  5.022841e-32
```

For factors, R treats the first level as the baseline level, e.g. the mean log(income) for a male is:

log(income) = 8.25 + 0.15 * education + 0 * height

Each additional level gets a coefficient that acts as an ***adjustment*** between the baseline level and the additional level, e.g. the mean income for a female is:

log(income) = 8.25 + 0.15 * education + 0 * height - 0.46

logistic regression

(Popular) **modeling functions in R**

function	package	fits
lm()	stats	linear models
glm()	stats	generalized linear models
gam()	mgcv	generalized additive models
glmnet()	glmnet	penalized linear models
rlm()	MASS	robust linear models
rpart()	rpart	trees
randomForest()	randomForest	random forests
xgboost()	xgboost	gradient boosting machines

(Popular) modeling functions in R

function	package	fits
lm()	stats	linear models
glm()	stats	generalized linear models
gam()	mgcv	generalized additive models
glmnet()	glmnet	penalized linear models
rlm()	MASS	robust linear models
rpart()	rpart	trees
randomForest()	randomForest	random forests
xgboost()	xgboost	gradient boosting machines

Bechdel— predicting pass/fail

Filter

	year	imdb	title	test	clean_test	binary	budget	domgross	intgross
1	2013	tt1711425	21 & Over	notalk	notalk	FAIL	13000000	25682380	4219576
2	2012	tt1343727	Dredd 3D	ok-disagree	ok	PASS	45000000	13414714	4086899
3	2013	tt2024544	12 Years a Slave	notalk-disagree	notalk	FAIL	20000000	53107035	15860703
4	2013	tt1272878	2 Guns	notalk	notalk	FAIL	61000000	75612460	13249301
5	2013	tt0453562	42	men	men	FAIL	40000000	95020213	9502021
6	2013	tt1335975	47 Ronin	men	men	FAIL	225000000	38362475	14580384
7	2013	tt1606378	A Good Day to Die Hard	notalk	notalk	FAIL	92000000	67349198	30424919
8	2013	tt2194499	About Time	ok-disagree	ok	PASS	12000000	15323921	8732474
9	2013	tt1814621	Admission	ok	ok	PASS	13000000	18007317	1800731
10	2013	tt1815862	After Earth	notalk	notalk	FAIL	130000000	60522097	24437319
11	2013	tt1800241	American Hustle	ok-disagree	ok	PASS	40000000	148430908	24948490
12	2013	tt1322269	August: Osage County	ok	ok	PASS	25000000	37304874	5030487
13	2013	tt1559547	Beautiful Creatures	ok	ok	PASS	50000000	19452138	5594067
14	2013	tt2334873	Blue Jasmine	ok-disagree	ok	PASS	18000000	33345833	6844783
15	2013	tt1535109	Captain Phillips	notalk	notalk	FAIL	55000000	107136417	21874357
16	2013	tt1939659	Carrie	ok	ok	PASS	30000000	35266619	8500165
17	2013	tt1985966	Cloudy with a Chance of Meatballs 2	nowomen-disagree	nowomen	FAIL	78000000	119640264	27172544
18	2013	tt1690953	Despicable Me 2	ok	ok	PASS	76000000	368065385	97076600

Showing 1 to 19 of 1,794 entries

Logistic regression takes 0/1 outcomes, so we have to mutate our data

```
bechdel <- bechdel %>%
    mutate(pass = if_else(binary == "PASS", 0, 1))
```

`glm()` uses the same formula syntax as `lm()`

```
mod_pass <- glm(pass~budget, data=bechdel, family=binomial)
```

```
summary(mod_pass)

Call:
glm(formula = pass ~ budget, family = binomial, data = bechdel)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9312  -1.2088   0.9046   1.1221   1.1955

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.248e-02  6.606e-02  -0.643     0.52
budget       5.797e-09  1.083e-09   5.354  8.6e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2467.3  on 1793  degrees of freedom
Residual deviance: 2436.2  on 1792  degrees of freedom
AIC: 2440.2

Number of Fisher Scoring iterations: 4
```

glm() works with all the same functions we've just been exploring

Your Turn

Model **pass** against **budget, year** and **domgross_2013**.

tidy() the model output.

03:00

```
mod_pass2 <- glm(pass~budget+year+domgross_2013,
data=bechdel, family=binomial)

mod_pass2 %>%
  tidy()
           term      estimate    std.error  statistic      p.value
1   (Intercept)  5.490589e+01 1.263961e+01  4.3439550 1.399402e-05
2        budget  6.816246e-09 1.294625e-09  5.2650353 1.401624e-07
3          year -2.747601e-02 6.312823e-03 -4.3524124 1.346477e-05
4 domgross_2013  3.289601e-10 4.958123e-10  0.6634771 5.070250e-01
```

Other modeling functions work much the same way

function	package	fits
lm()	stats	linear models
glm()	stats	generalized linear models
gam()	mgcv	generalized additive models
glmnet()	glmnet	penalized linear models
rlm()	MASS	robust linear models
rpart()	rpart	trees
randomForest()	randomForest	random forests
xgboost()	xgboost	gradient boosting machines