These materials adapted by Amelia McNamara from the RStudio CC BY-SA materials Introduction to R (2014) and Master the Tidyverse (2017).

Introduction to R & RStudio:

deck 10: Modeling

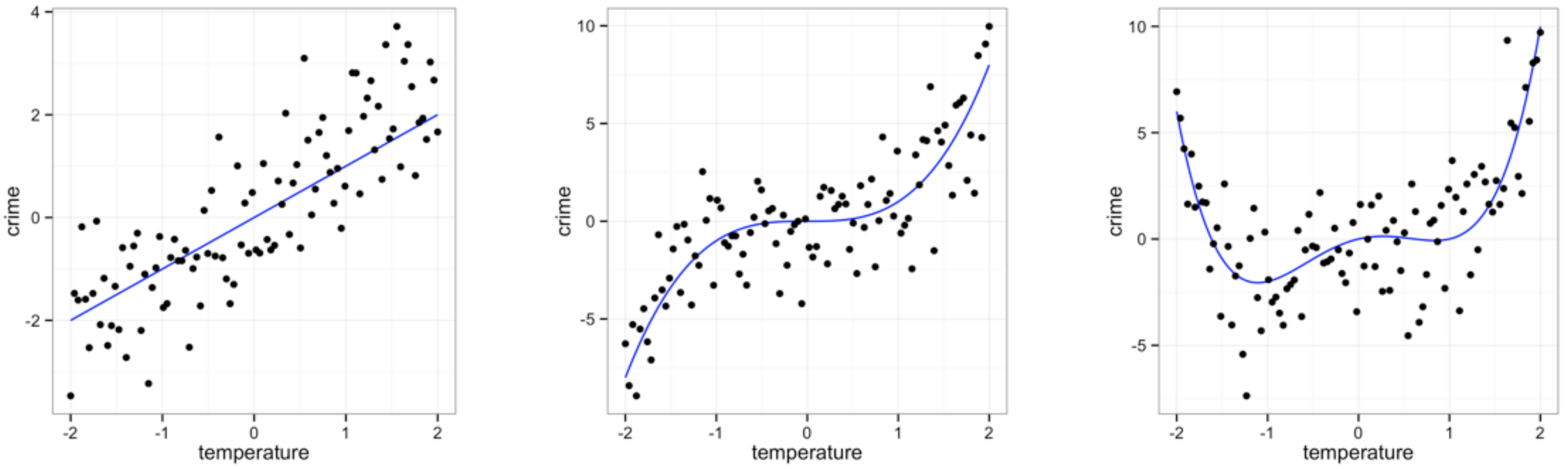
Amelia McNamara
Visiting Assistant Professor of Statistical and Data Sciences
Smith College

January 2018

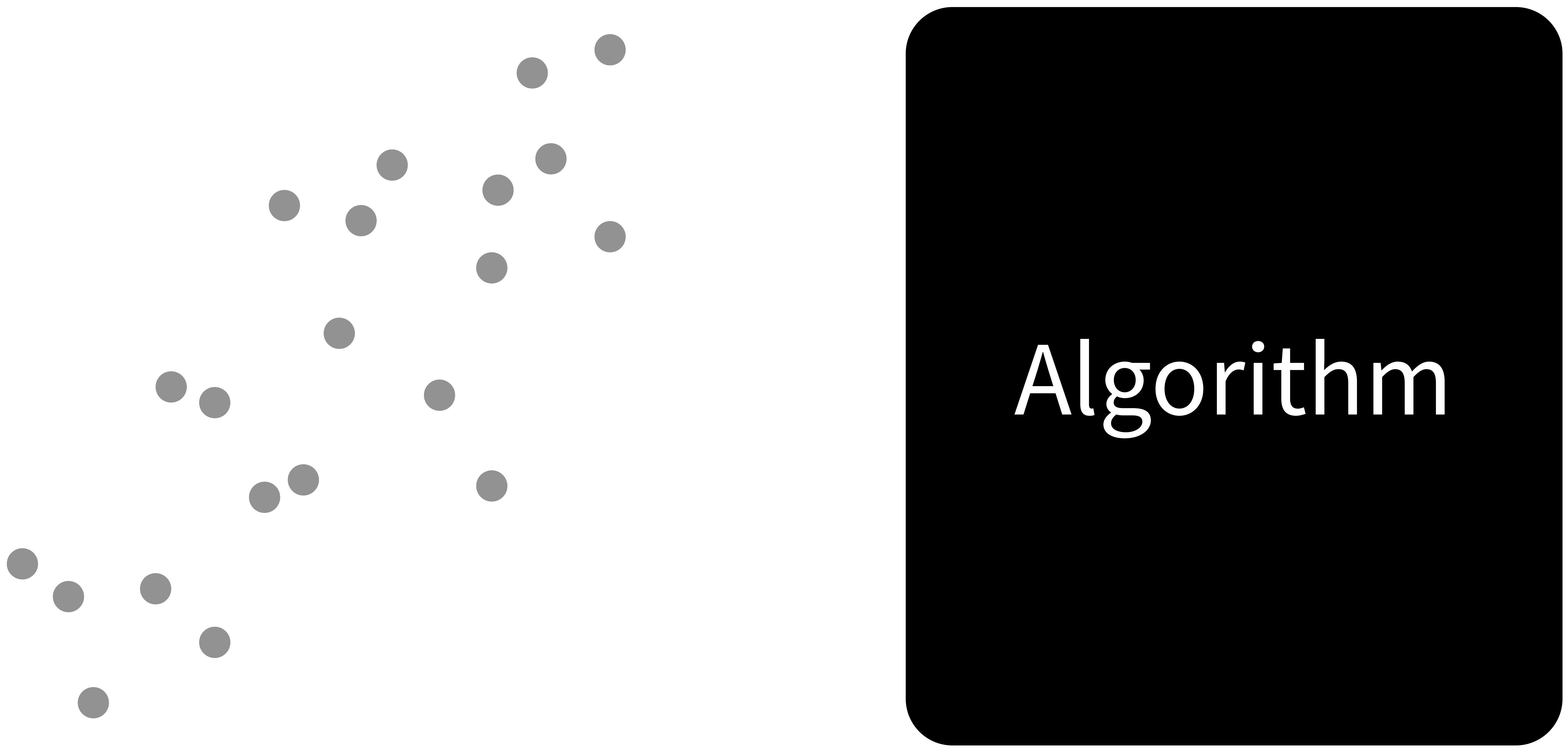
The basics

Models

A low dimensional description of a higher dimensional data set.



Models

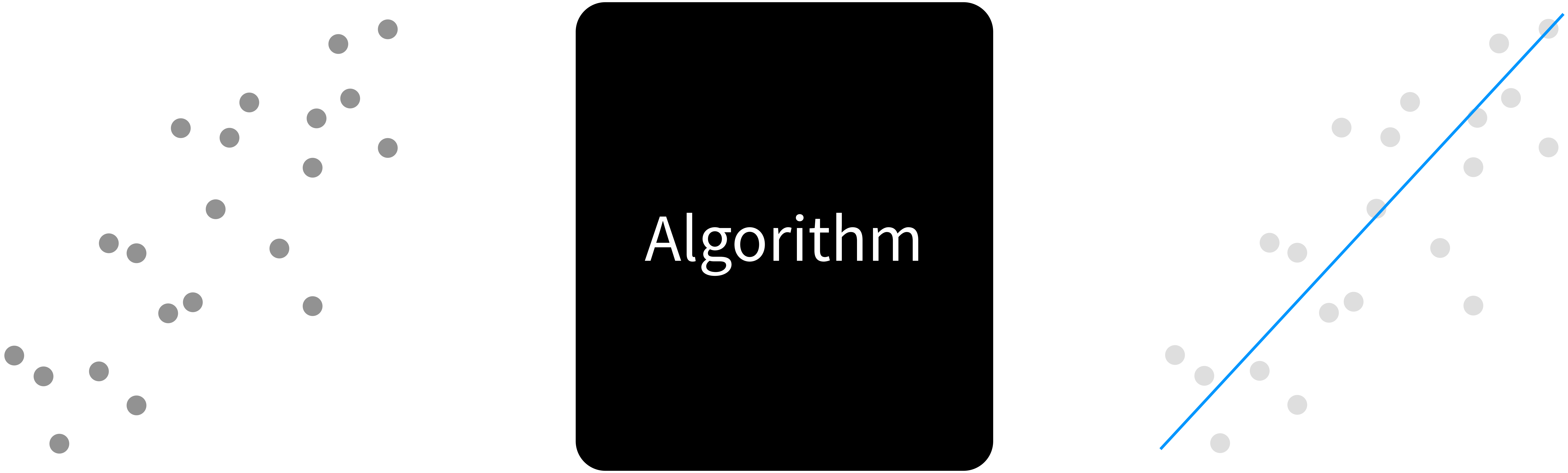


Data

Model Function

Models

What is the model function?

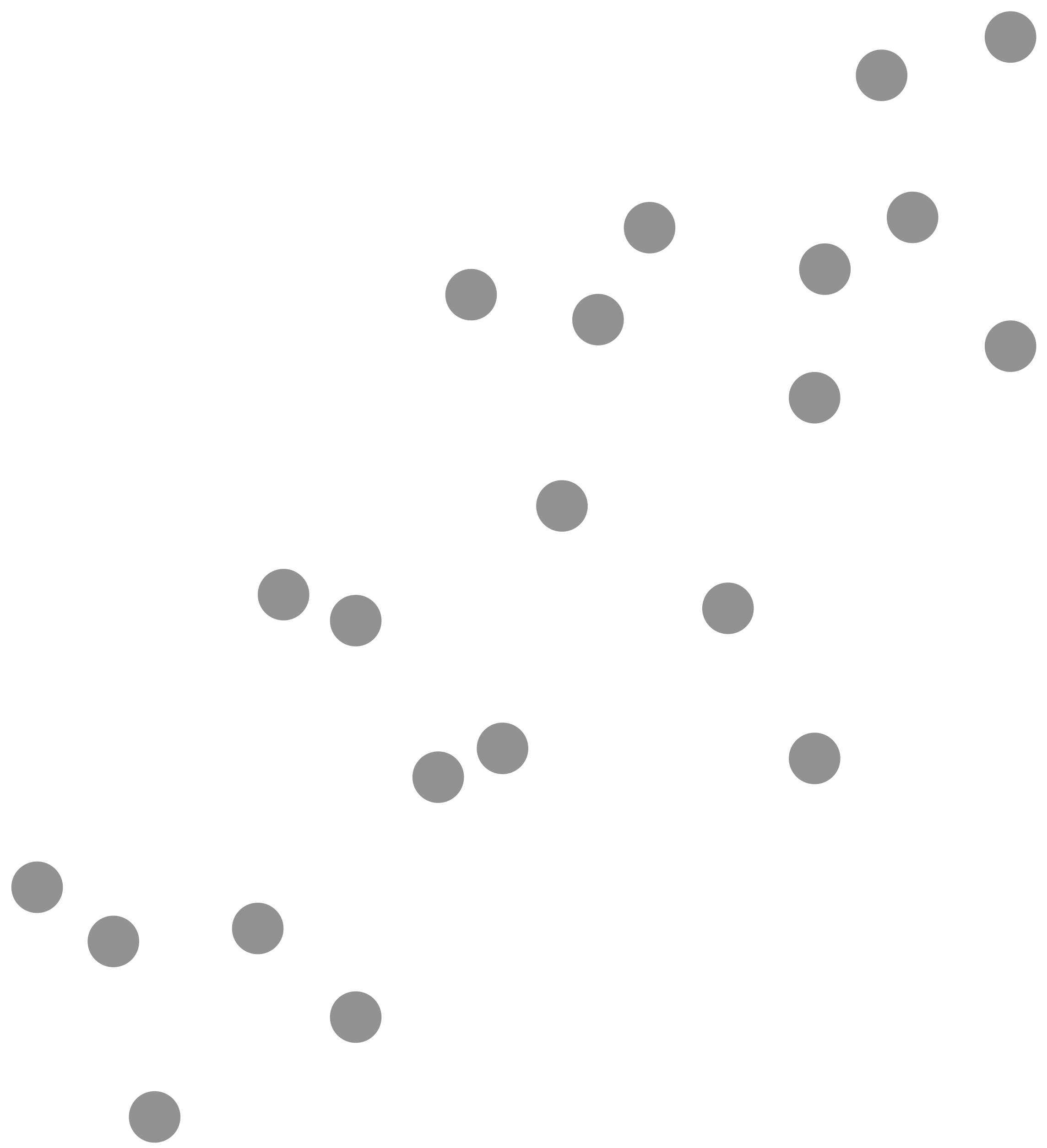


Data

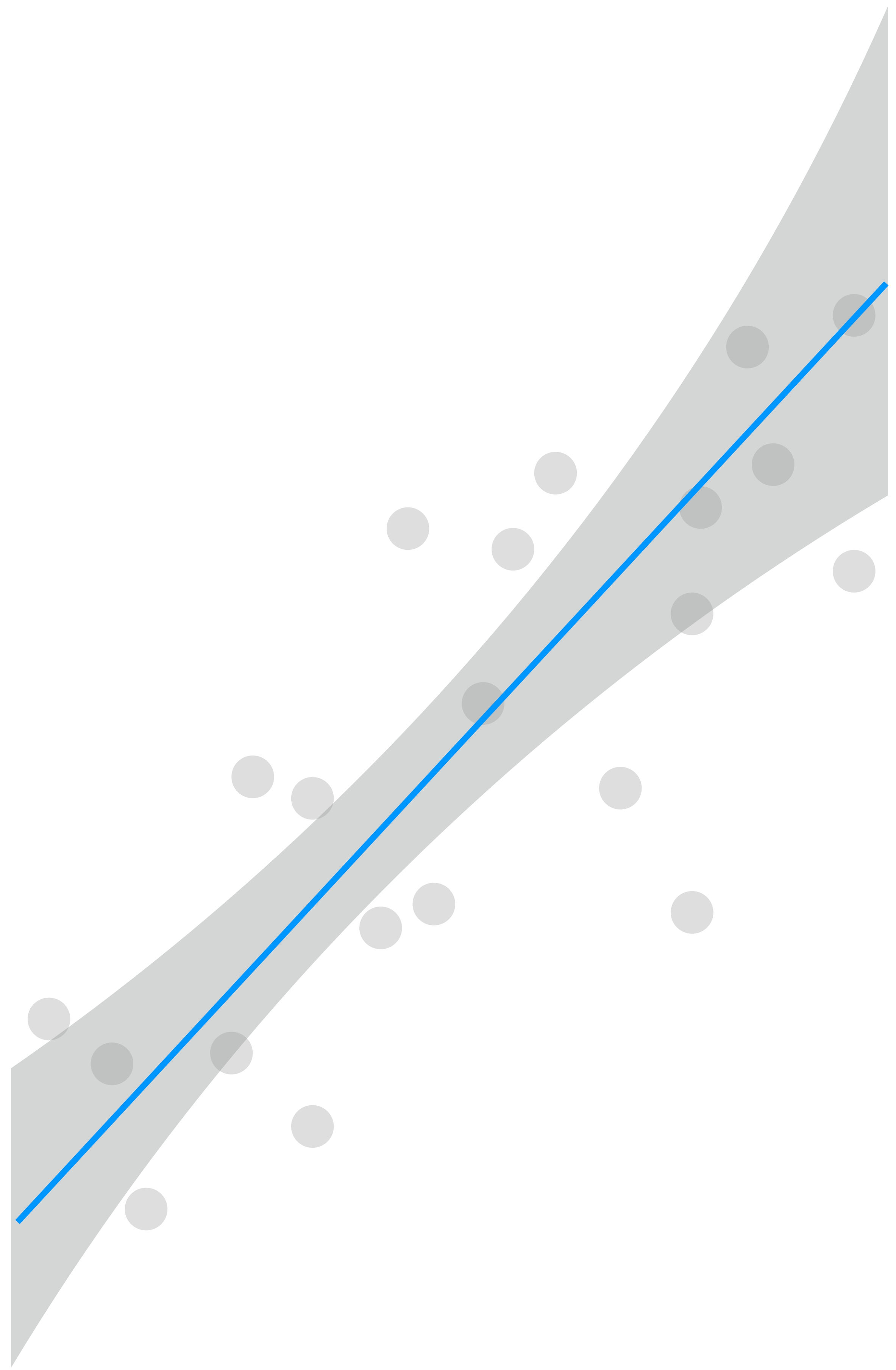
Model Function

Models

What uncertainty is associated with it?



Algorithm

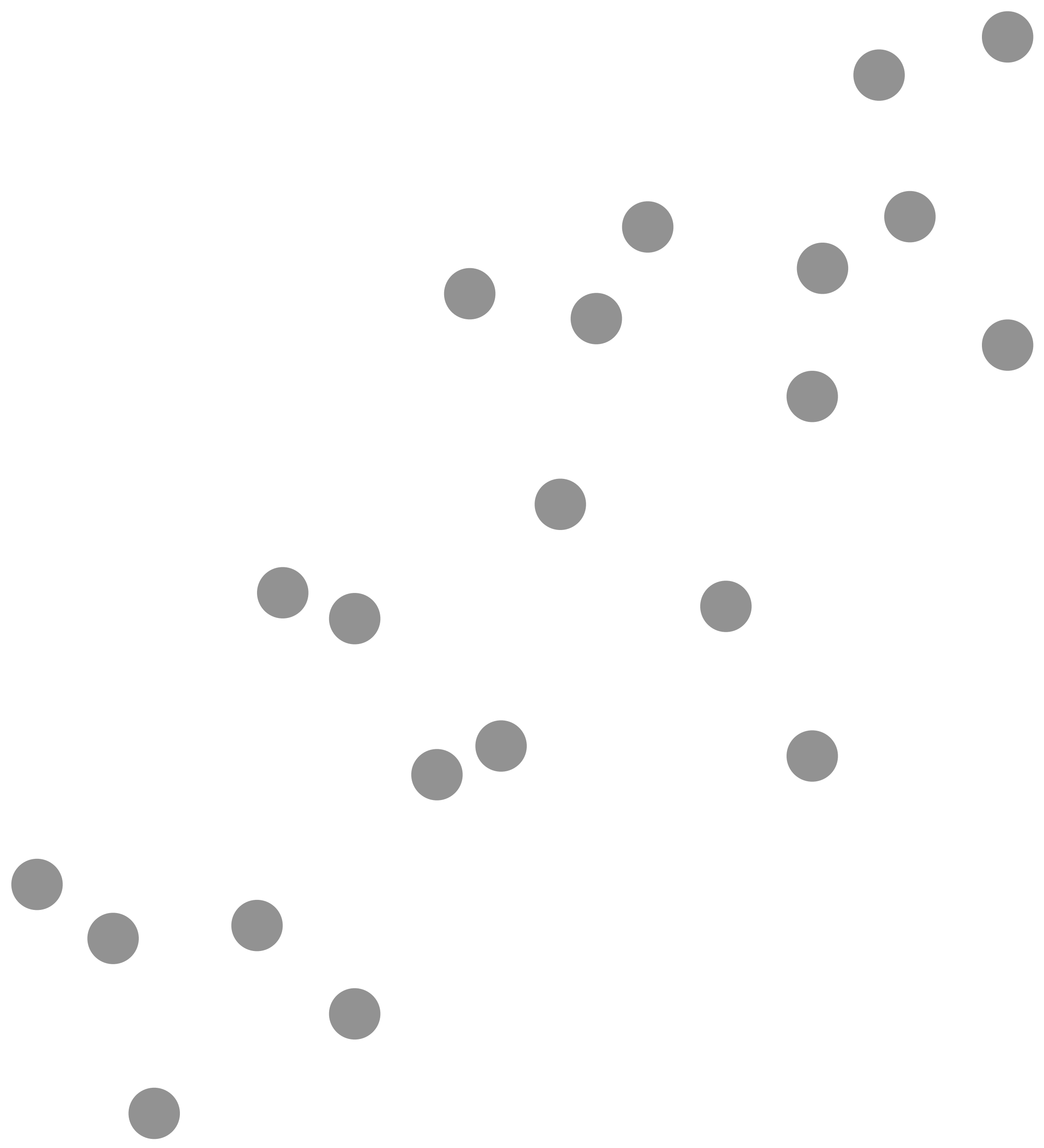


Data

Model Function

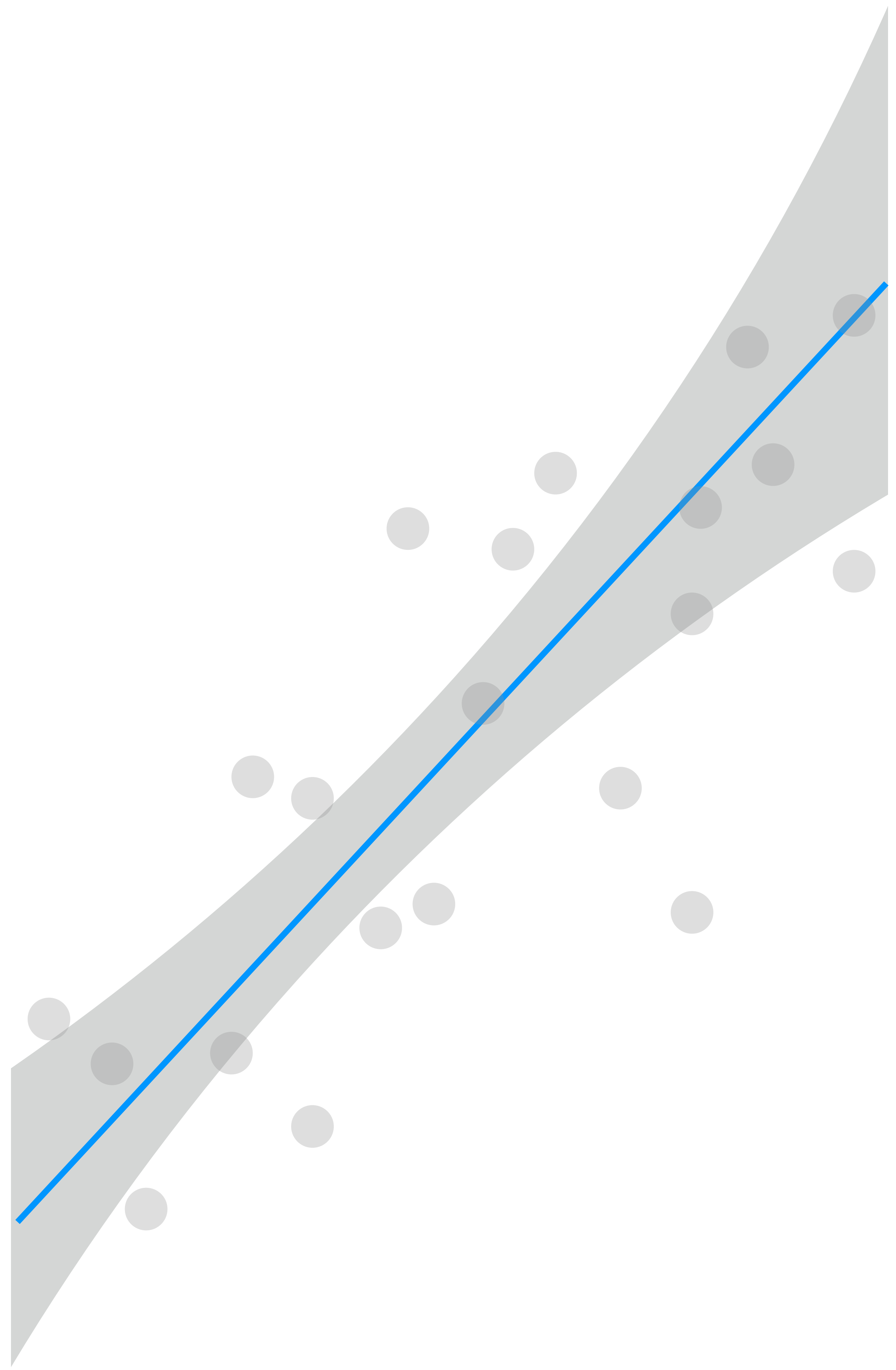
Models

How "good" is the model?



Data

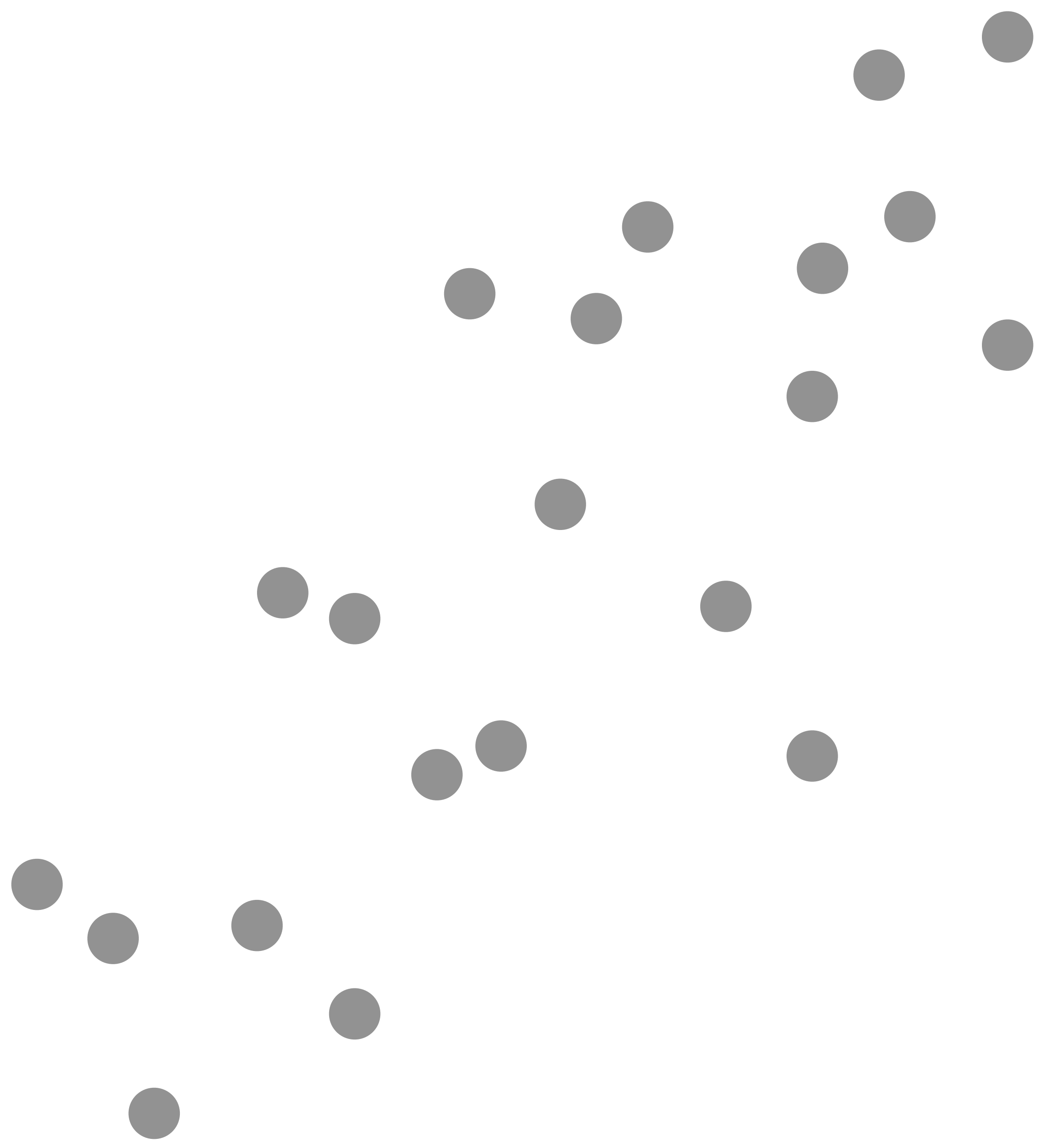
Algorithm



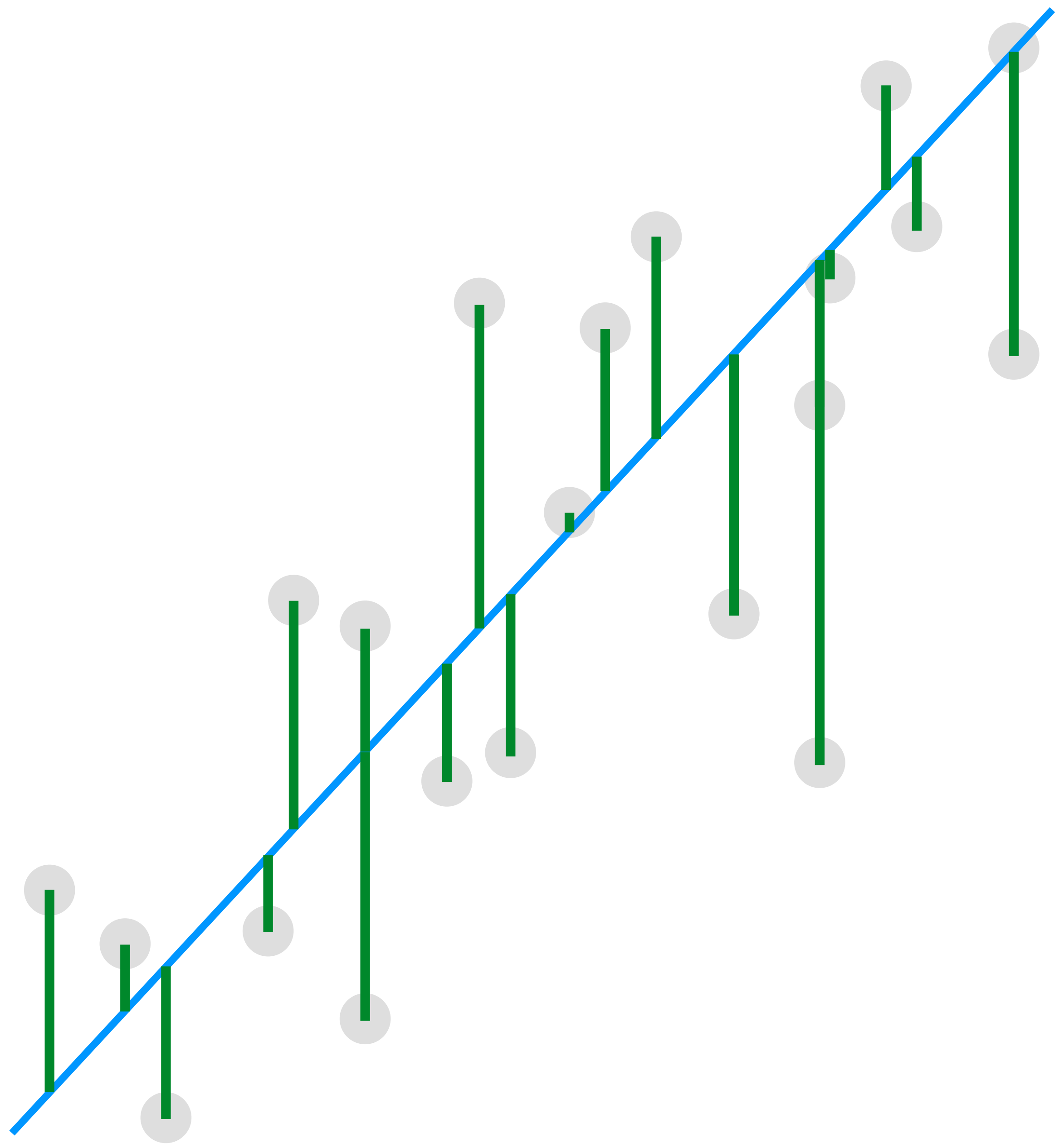
Model Function

Models

What are the residuals?



Algorithm



Data

Model Function

Models

What are the predictions?

Data

Model Function

(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

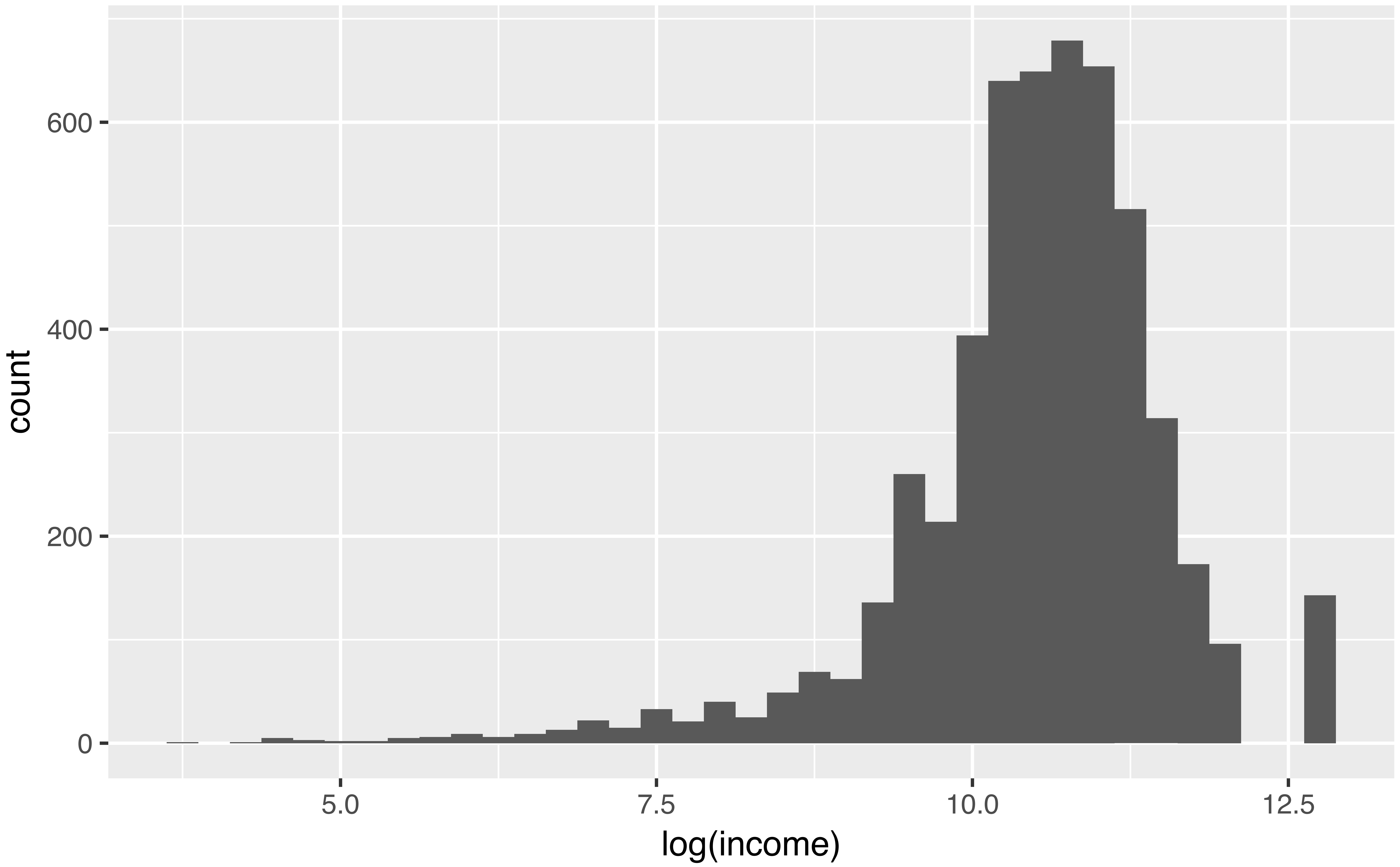
function	package	fits
lm()	**stats**	**linear models**
glm()	stats	generalized linear models
gam()	mgcv	generalized additive models
glmnet()	glmnet	penalized linear models
rlm()	MASS	robust linear models
rpart()	rpart	trees
randomForest()	randomForest	random forests
xgboost()	xgboost	gradient boosting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

```
wages
```

income	height	weight	age	marital	sex	education	afqt
<int>	<dbl>	<int>	<int>	<fctr>	<fctr>	<int>	<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 1

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

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

02:00

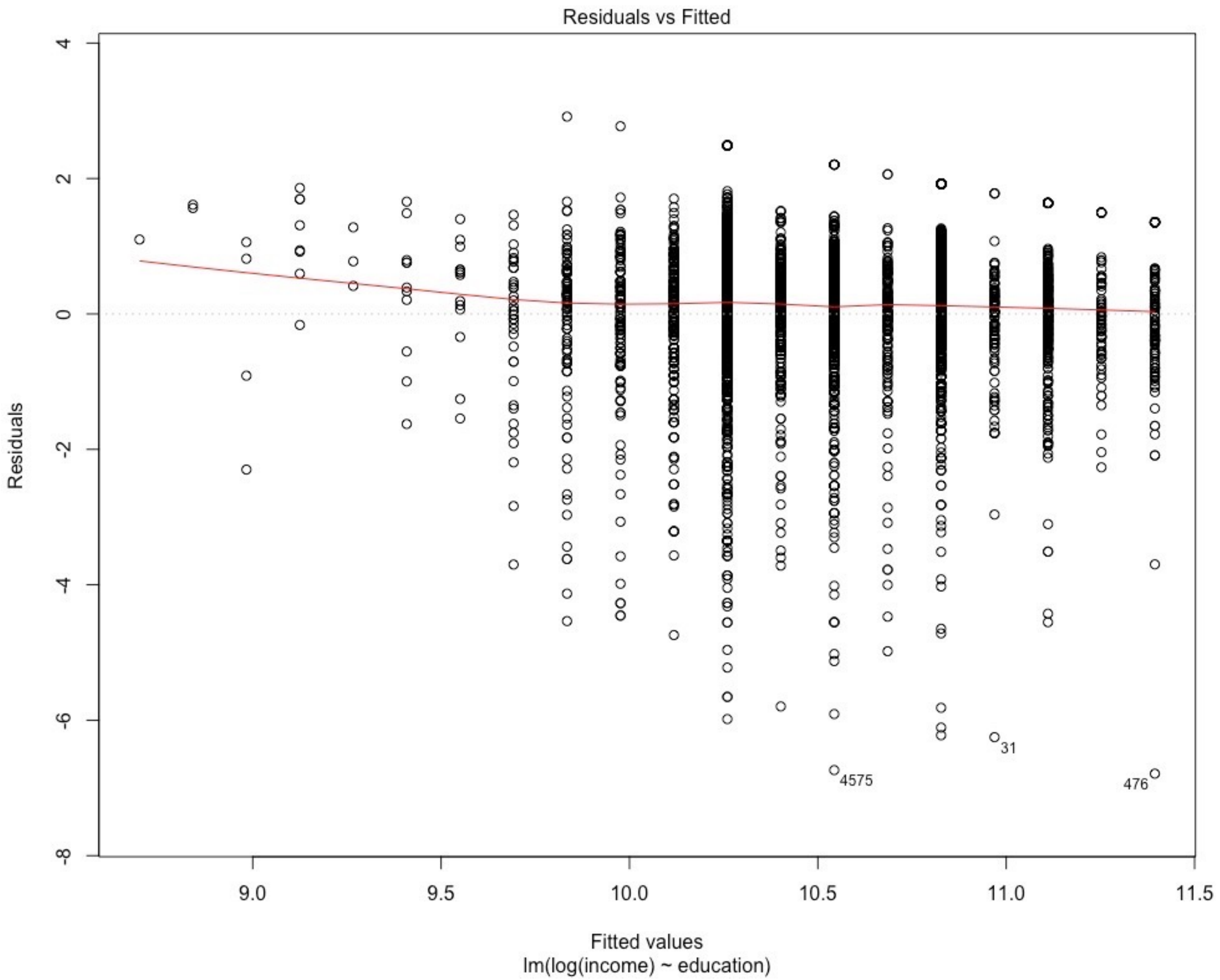
```r
mod_e <- lm(log(income) ~ education, data = wages)
mod_e
##  Call:
##  lm(formula = log(income) ~ education, data = wages)
##
## Coefficients:
## (Intercept)    education
##      8.5577       0.1418
class(mod_e)
## "lm"
```

```
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
```

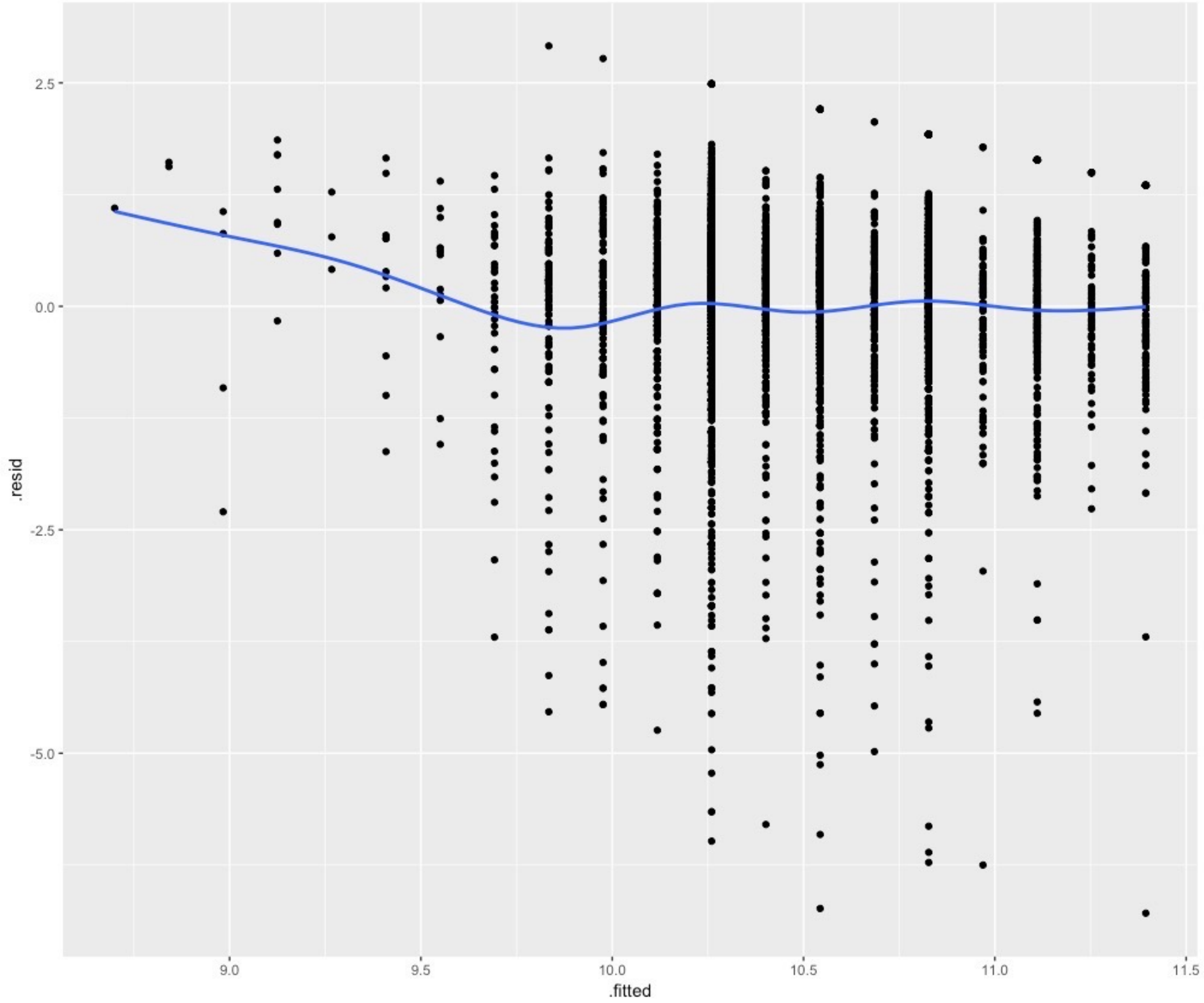
```
names(mod_e)
 [1] "coefficients"  "residuals"     "effects"
 [4] "rank"          "fitted.values" "assign"
 [7] "qr"            "df.residual"   "na.action"
[10] "xlevels"       "call"          "terms"
[13] "model"
mod_e$coefficients
(Intercept)   education
  8.5576906   0.1418404
```

Plotting models

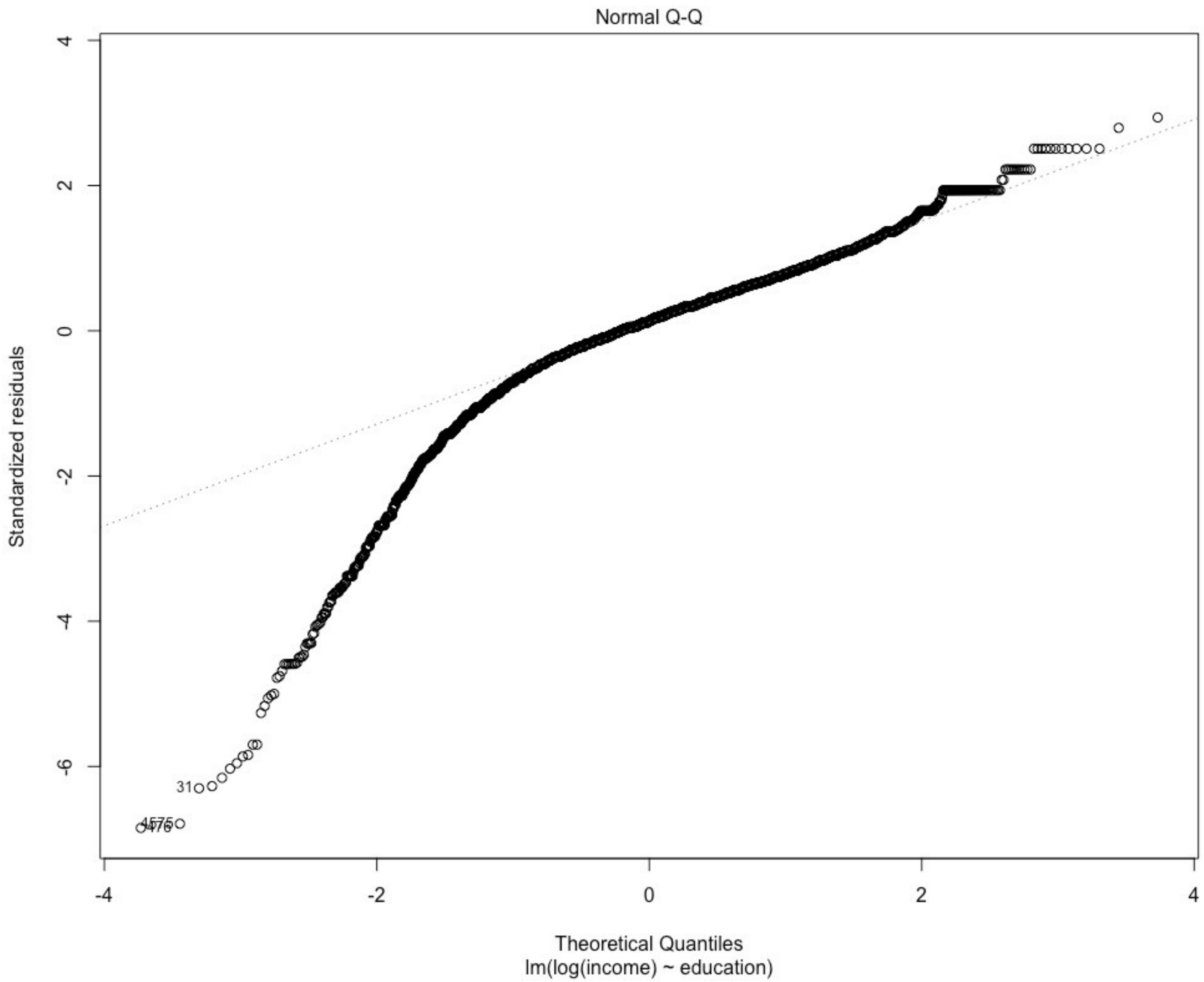
```
plot(mod_e, which=1)
```



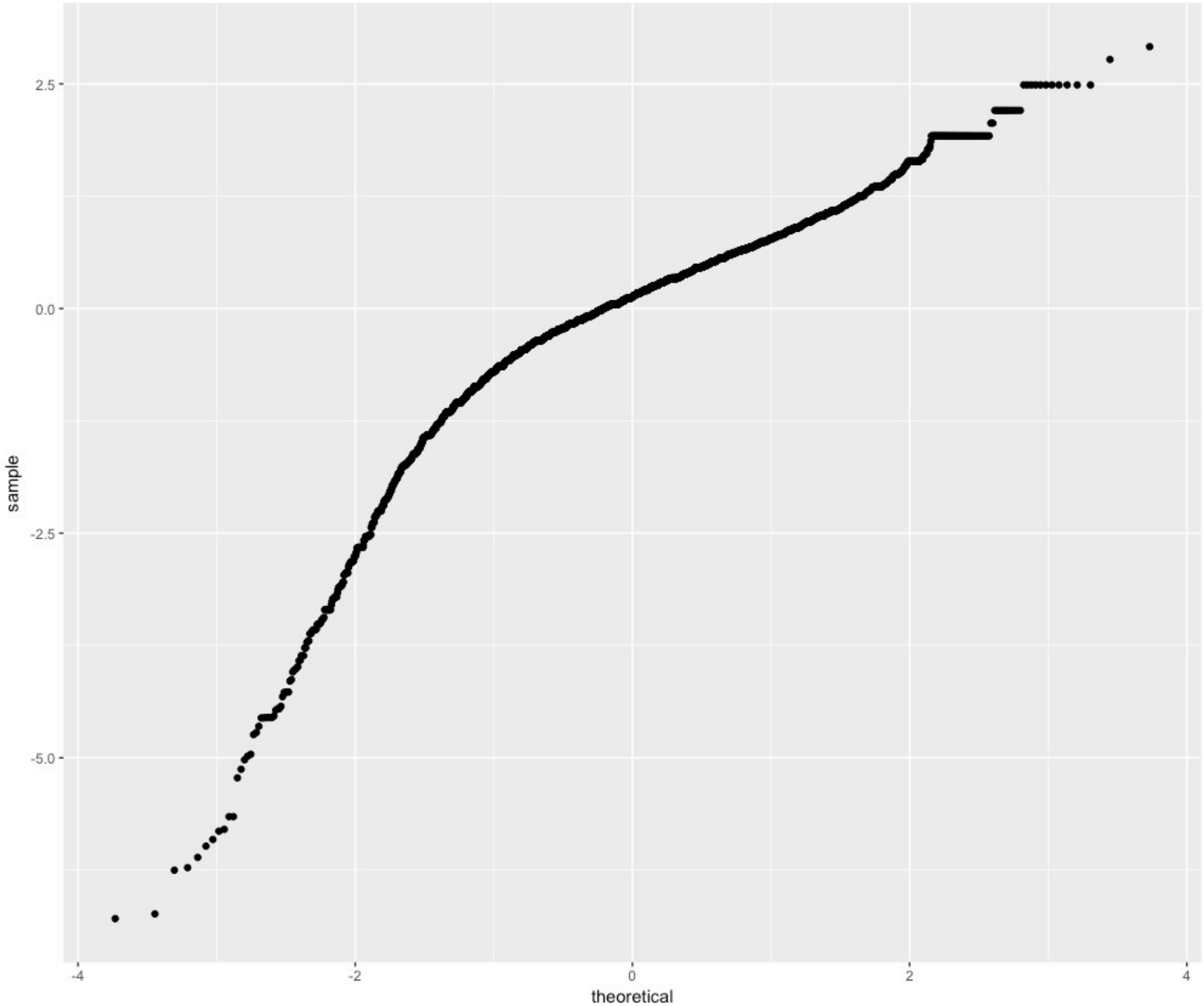
```
ggplot(mod_e, aes(x=.fitted, y=.resid)) + geom_point() +
geom_smooth(se = FALSE)
```



```
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 <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>	p.value <dbl>	df <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**

```r
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 2

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 3

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

03:00

```
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 4

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

03:00

```
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:

$$\log(\text{income}) = 8.25 + 0.15 * \text{education} + 0 * \text{height}$$

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:

$$\log(\text{income}) = 8.25 + 0.15 * \text{education} + 0 * \text{height} - 0.46$$

```
##          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:

$$\log(\text{income}) = 8.25 + 0.15 * \text{education} + 0 * \text{height}$$

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:

$$\log(\text{income}) = 8.25 + 0.15 * \text{education} + 0 * \text{height} - 0.46$$