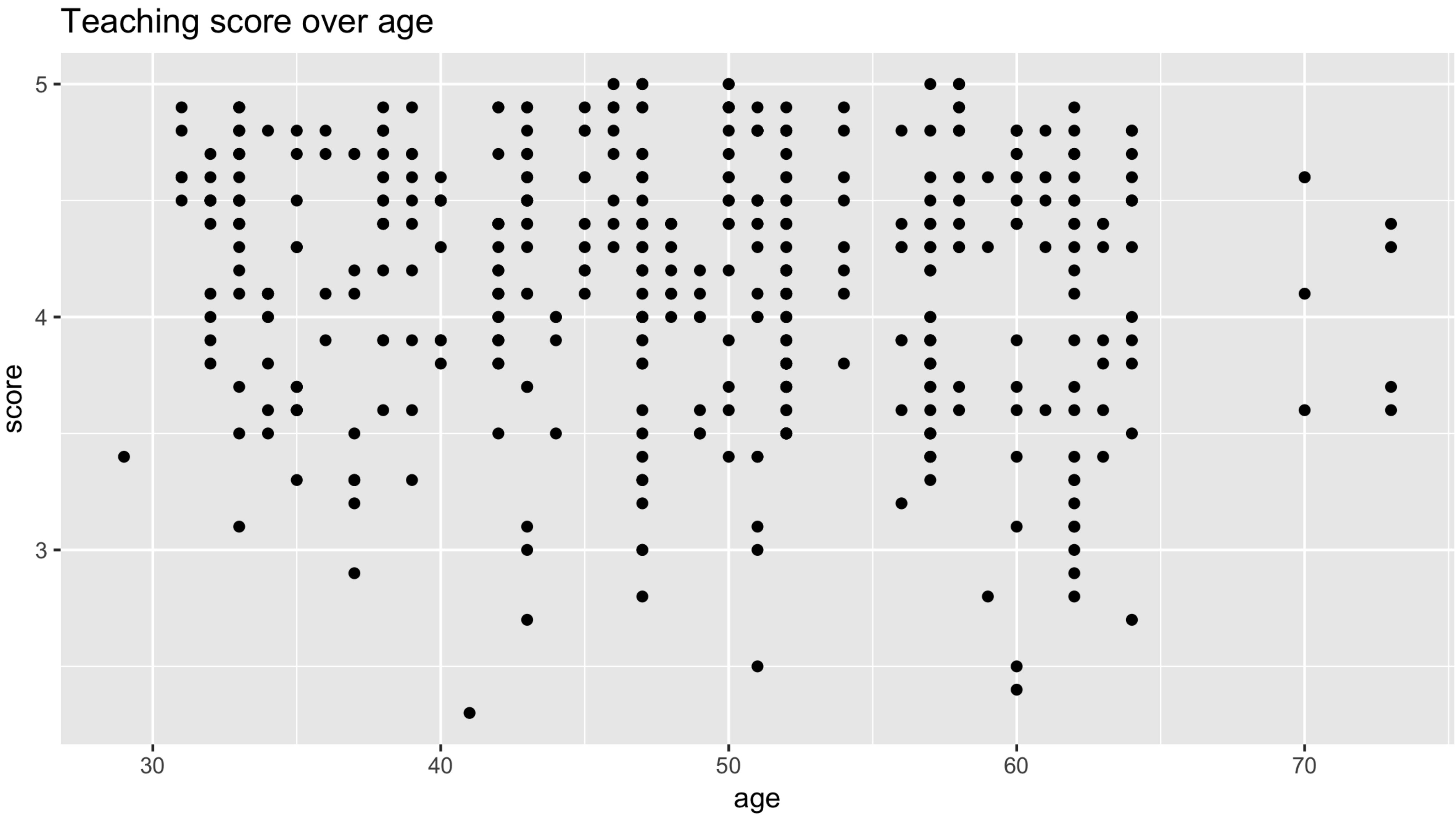
Explaining teaching score with age

MODELING WITH DATA IN THE TIDYVERSE

Albert Y. Kim

Assistant Professor of Statistical and Data Sciences

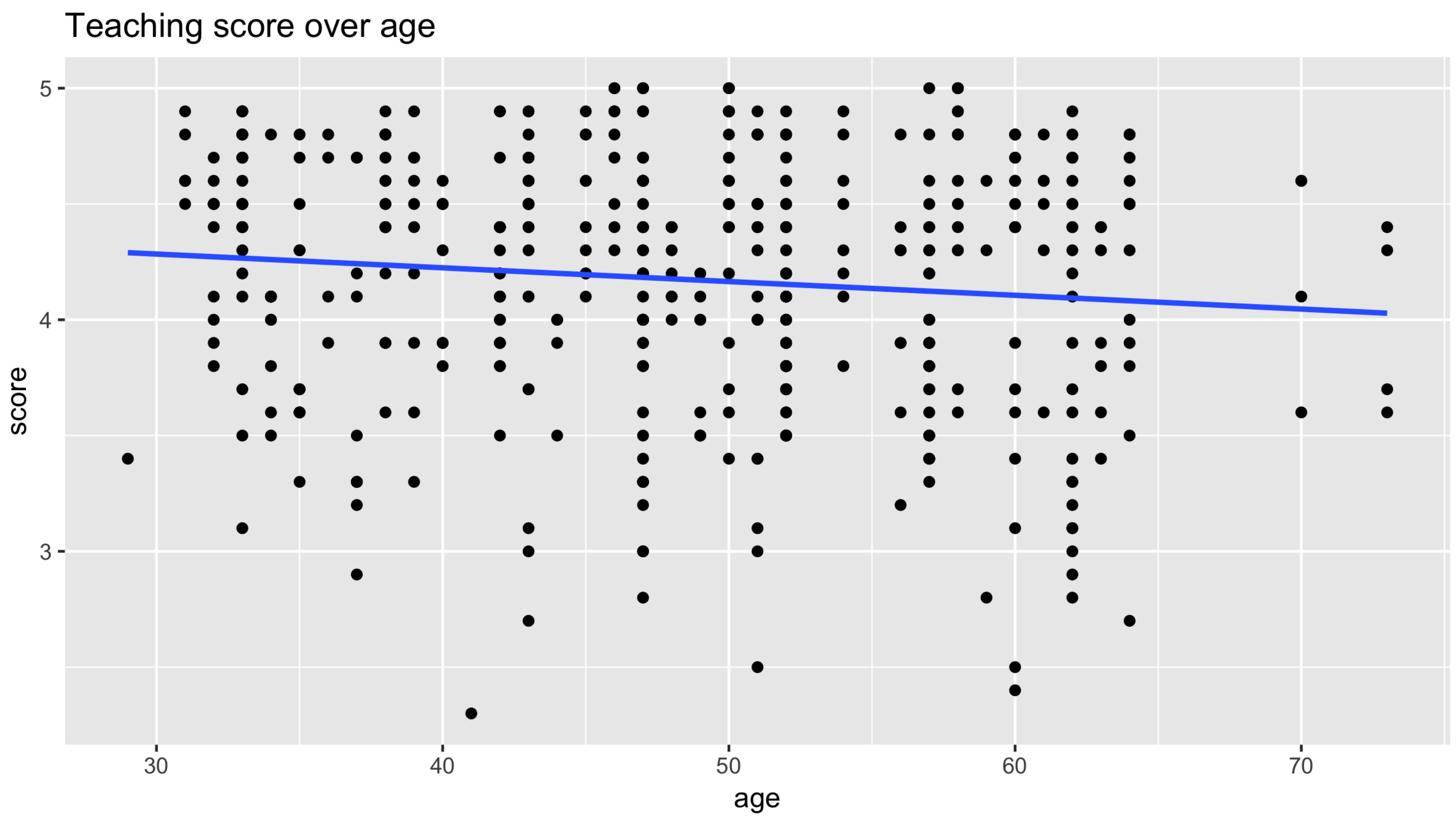
Refresher: Exploratory data visualization



Regression line

```r
# Code to create scatterplot
ggplot(evals, aes(x = age, y = score)) +
  geom_point() +
  labs(x = "age", y = "score",
       title = "Teaching score over age")
# Add a "best-fitting" line
ggplot(evals, aes(x = age, y = score)) +
  geom_point() +
  labs(x = "age", y = "score",
       title = "Teaching score over age") +
  geom_smooth(method = "lm", se = FALSE)
```

Regression line



Refresher: Modeling in general

- **Truth:** Assumed model is $y = f(\vec{x}) + \epsilon$
- **Goal:** Given y and $\vec{x}$, fit a model $\hat{f}(\vec{x})$ that *approximates* $f(\vec{x})$, where $\hat{y} = \hat{f}(\vec{x})$ is the *fitted/predicted* value for the *observed* value y

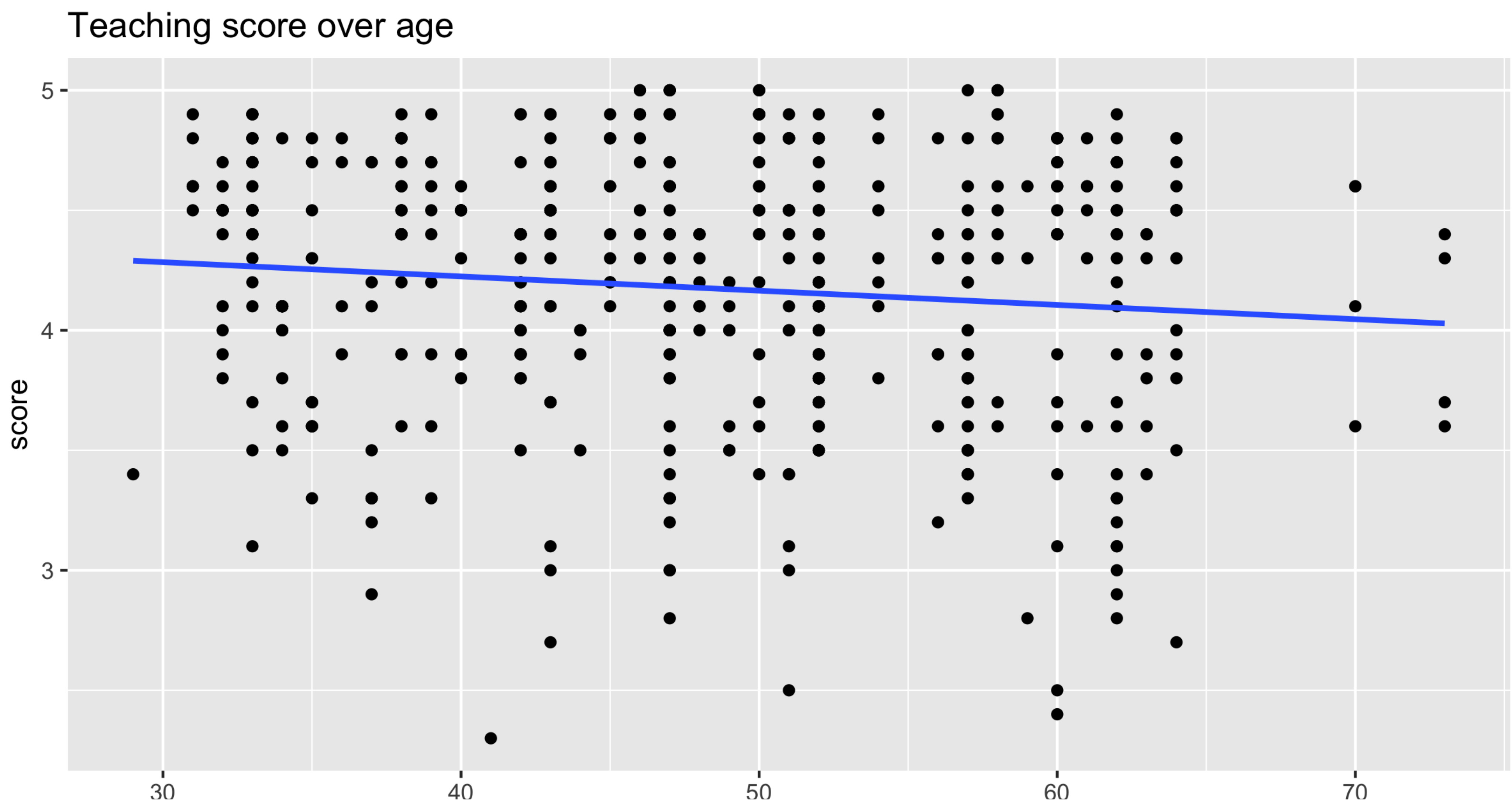
Modeling with basic linear regression

- **Truth:**
  - Assume $f(x) = \beta_0 + \beta_1 \cdot x$
  - *Observed* value $y = f(x) + \epsilon = \beta_0 + \beta_1 \cdot x + \epsilon$
- **Fitted:**
  - Assume $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$
  - *Fitted/predicted* value $\hat{y} = \hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$

Back to regression line

Equation for fitted blue regression line:

$$\hat{y} = \hat{f}(\vec{x}) = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$$



Computing slope and intercept of regression line

Using the formula form `y ~ x`:

```r
# Fit regression model using formula of form: y ~ x
model_score_1 <- lm(score ~ age, data = evals)
# Output contents
model_score_1
```

```
Call:
lm(formula = score ~ age, data = evals)

Coefficients:
(Intercept)          age
   4.461932    -0.005938
```

Computing slope and intercept of regression line

Using the formula form `y ~ x`, which is akin to $\hat{y} = \hat{f}(\vec{x})$

```
# Fit regression model using formula of form: y ~ x
model_score_1 <- lm(score ~ age, data = evals)

# Output regression table using wrapper function:
get_regression_table(model_score_1)
```

```
# A tibble: 2 x 7
  term      estimate std_error statistic p_value...
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>...
1 intercept    4.46      0.127     35.2    0...
2 age         -0.006     0.003     -2.31   0.021...
```

Let's practice!

MODELING WITH DATA IN THE TIDYVERSE

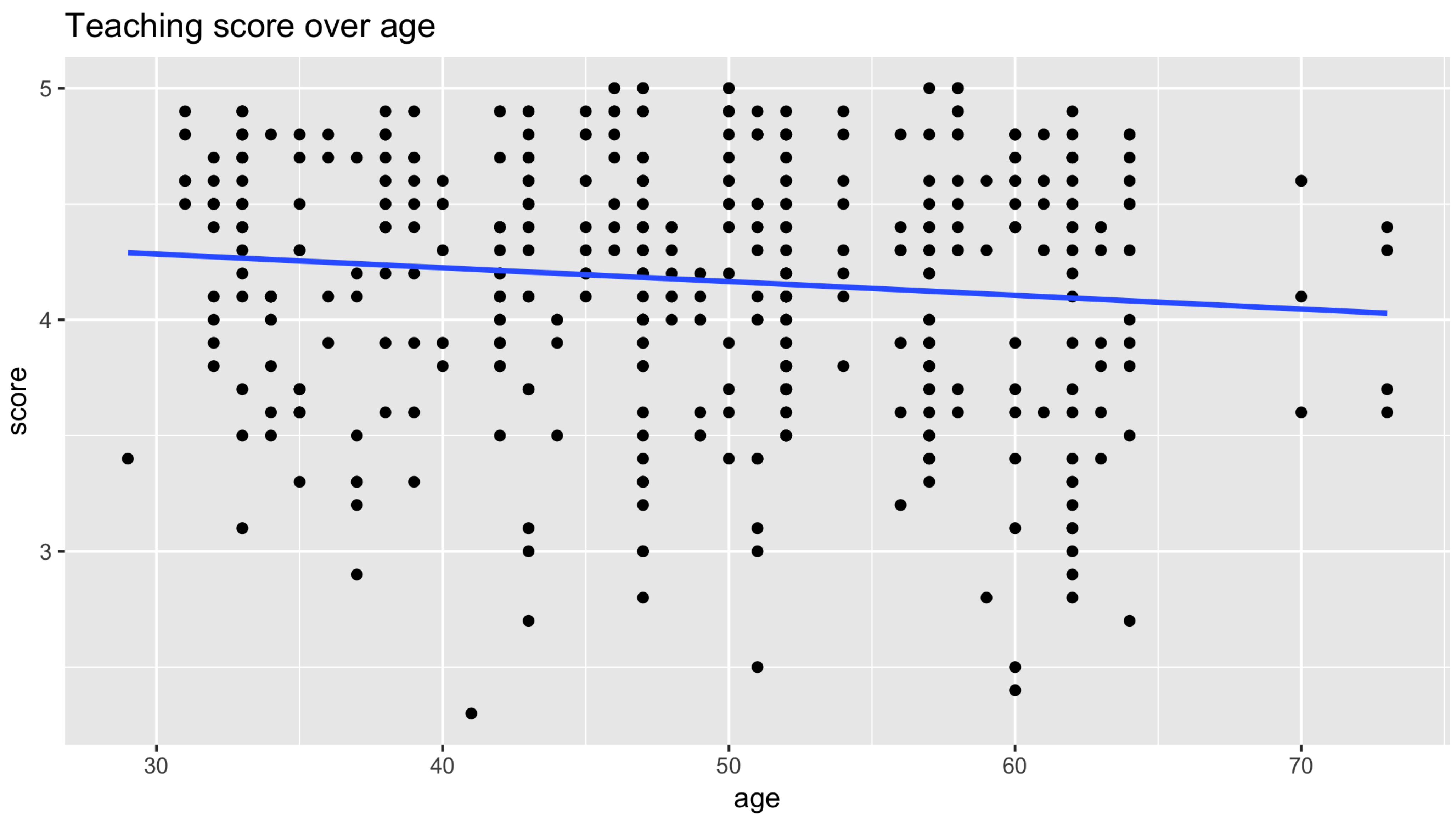
Predicting teaching score using age

MODELING WITH DATA IN THE TIDYVERSE

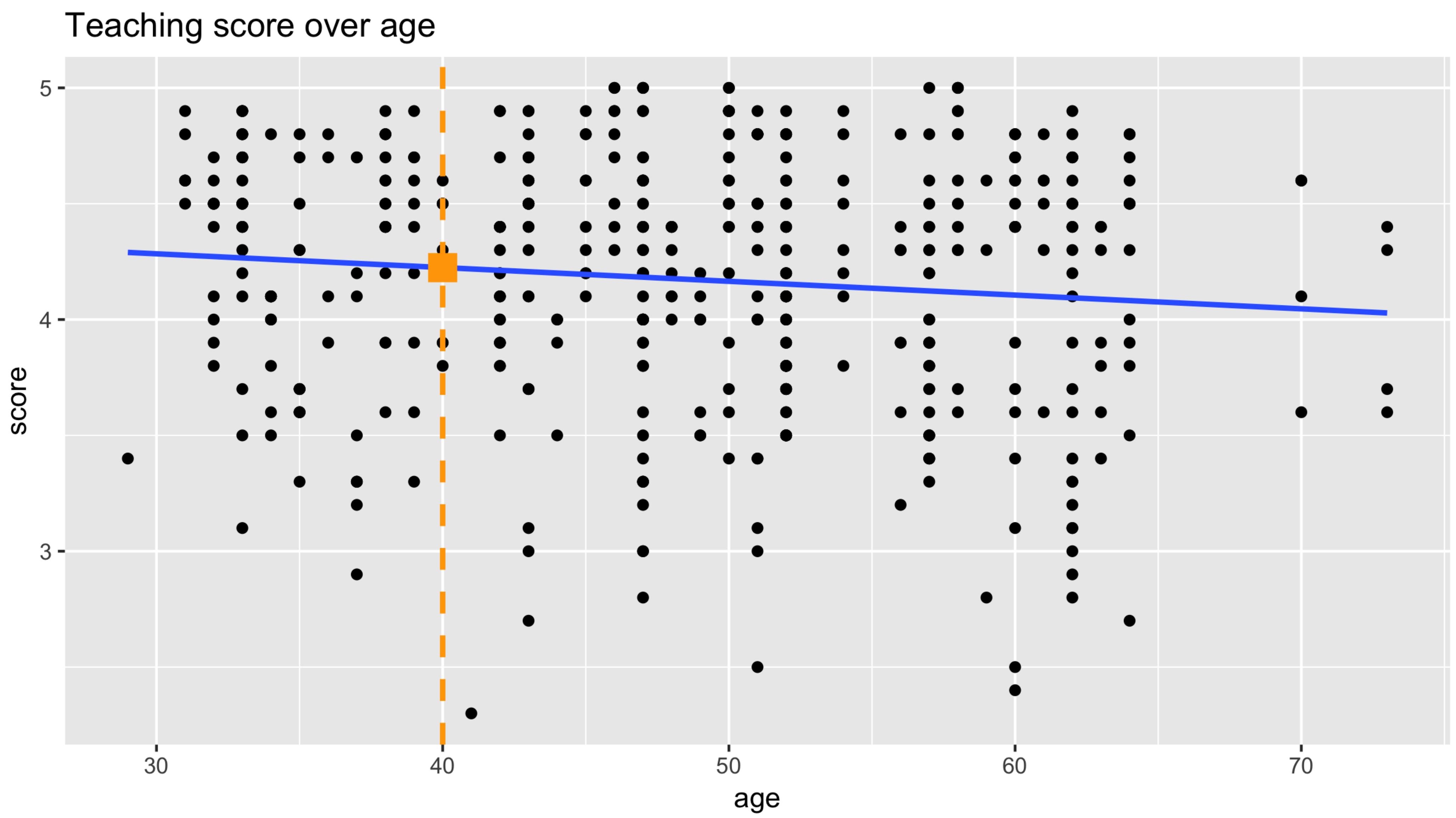
Albert Y. Kim

Assistant Professor of Statistical and Data Sciences

Refresher: Regression line



New instructor prediction



Refresher: Regression table

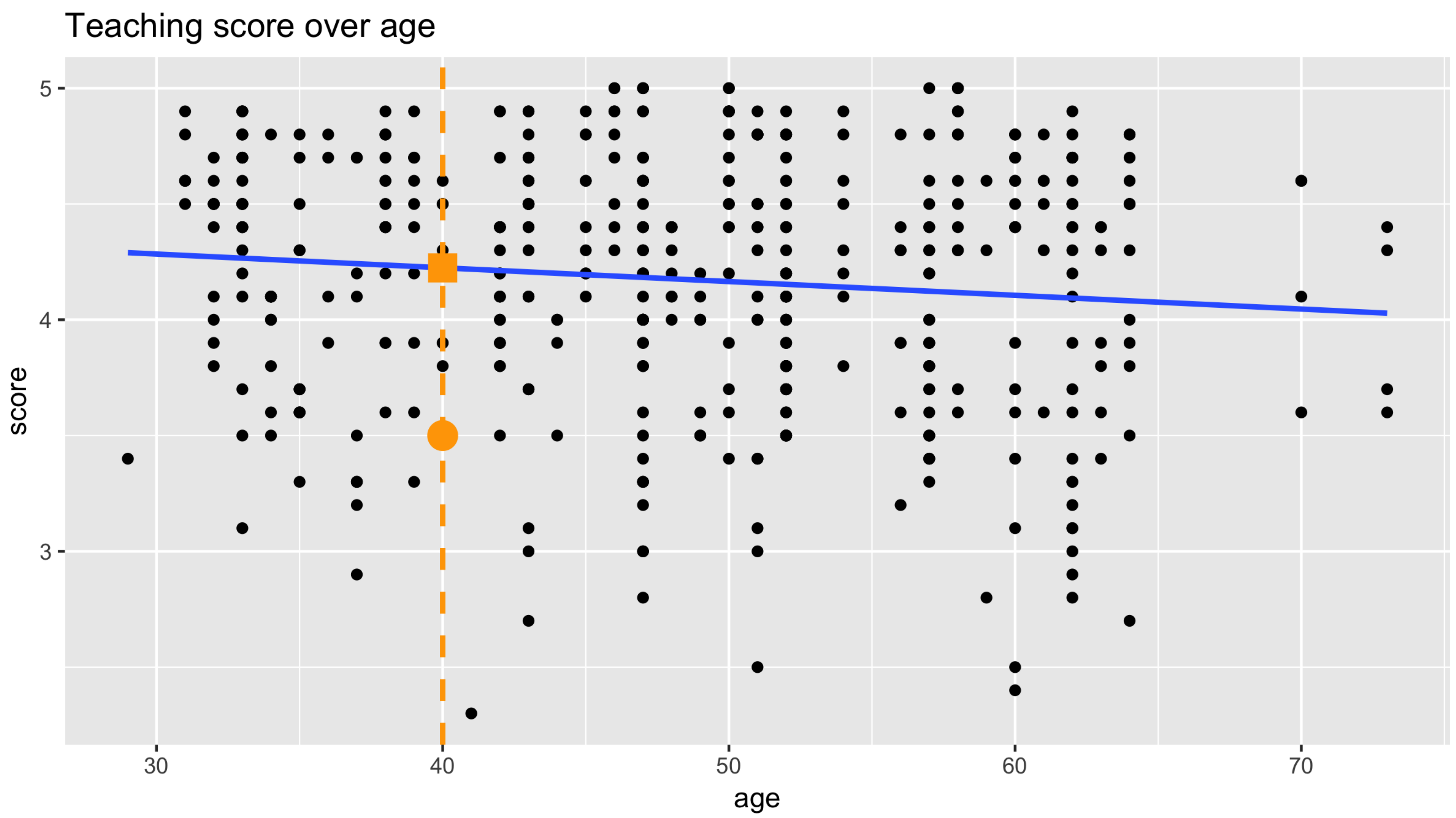
```r
library(ggplot2)
library(dplyr)
library(moderndive)

# Fit regression model using formula of form: y ~ x
model_score_1 <- lm(score ~ age, data = evals)

# Output regression table using wrapper function
get_regression_table(model_score_1)
```

```
# A tibble: 2 x 7
  term      estimate std_error statistic p_value lower_ci...
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>...
1 intercept    4.46      0.127     35.2    0        4.21...
2 age         -0.006     0.003     -2.31   0.021   -0.011...
```

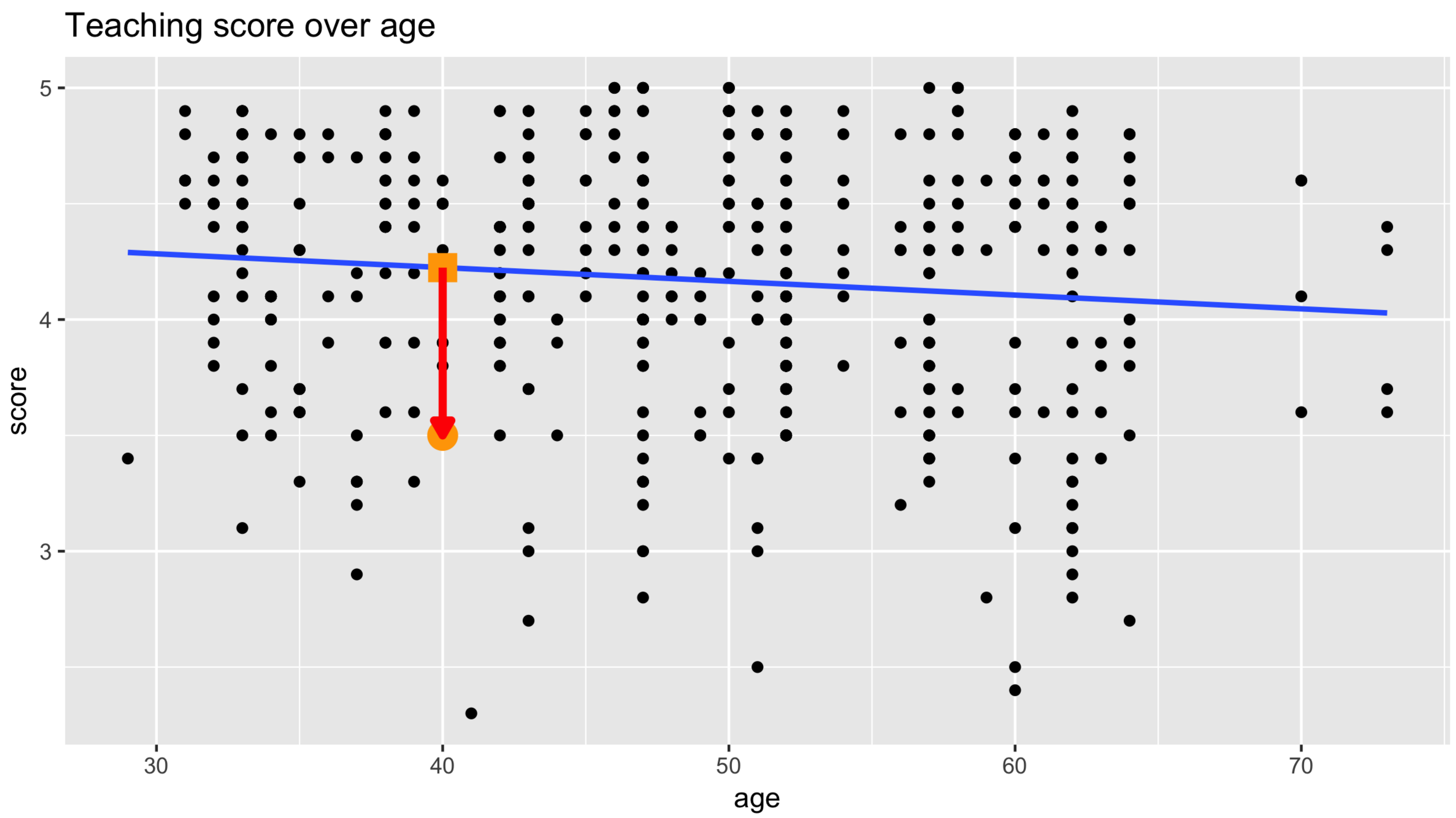
Predicted value

- Predictive regression models in general:
  $\hat{y} = \hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$
- Our predictive model: $\widehat{\text{score}} = 4.46 - 0.006 \cdot \text{age}$
- Our prediction: $4.46 - 0.006 \cdot 40 = 4.22$

Prediction error



Prediction error



Residuals as model errors

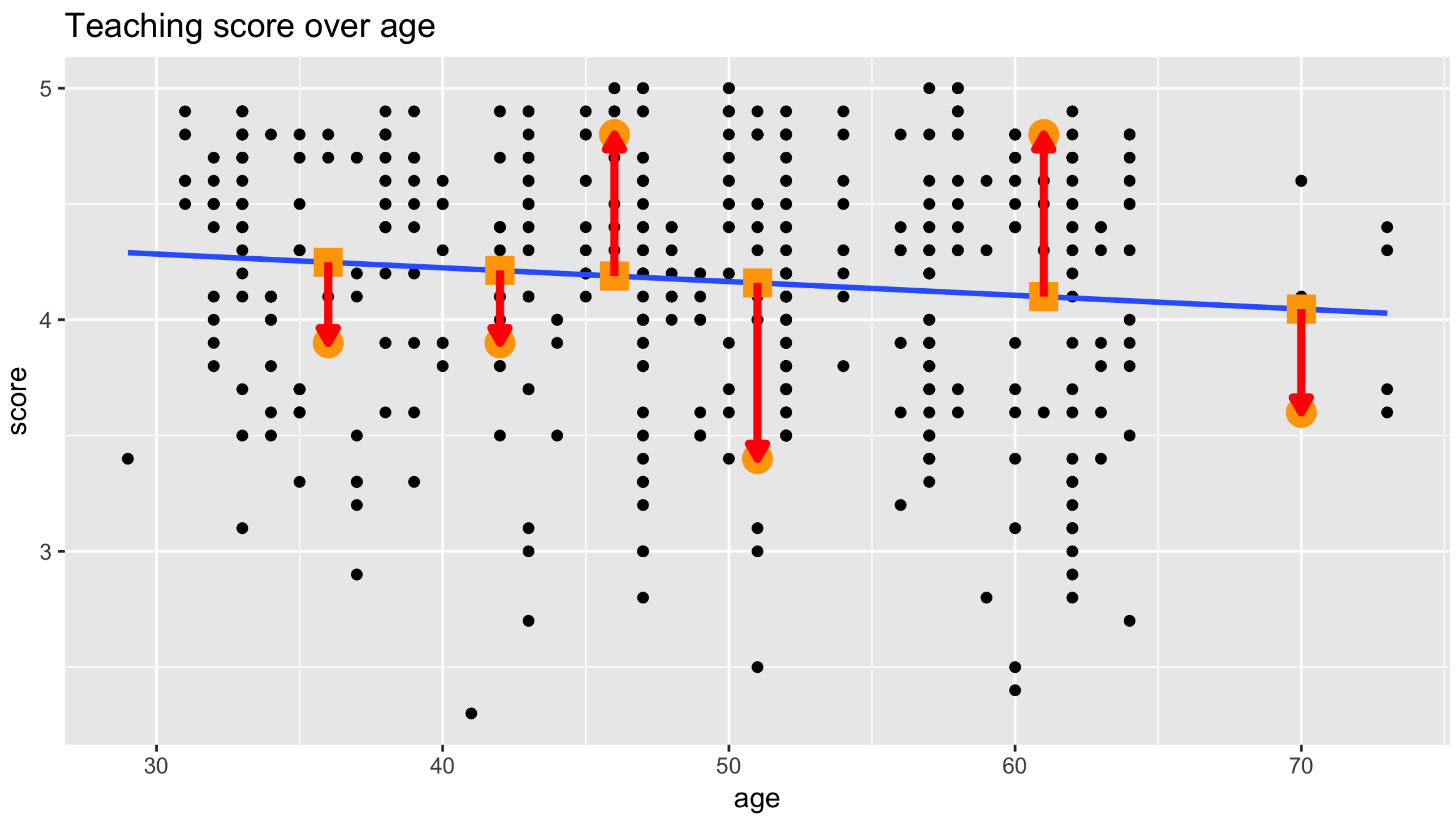
- Residual = $y - \hat{y}$
- Corresponds to ϵ from $y = f(\vec{x}) + \epsilon$
- For our example instructor: $y - \hat{y} = 3.5 - 4.22 = -0.72$
- In linear regression, they are on average 0.

Computing all predicted values

```r
# Fit regression model using formula of form: y ~ x
model_score_1 <- lm(score ~ age, data = evals)
# Get information on each point
get_regression_points(model_score_1)
```

```
# A tibble: 463 x 5
      ID score   age score_hat residual
   <int> <dbl> <dbl>     <dbl>    <dbl>
 1     1   4.7    36      4.25    0.452
 2     2   4.1    36      4.25   -0.148
 3     3   3.9    36      4.25   -0.348
 4     4   4.8    36      4.25    0.552
 5     5   4.6    59      4.11    0.488
```

"Best fitting" regression line



Let's practice!

MODELING WITH DATA IN THE TIDYVERSE

Explaining teaching score with gender

MODELING WITH DATA IN THE TIDYVERSE

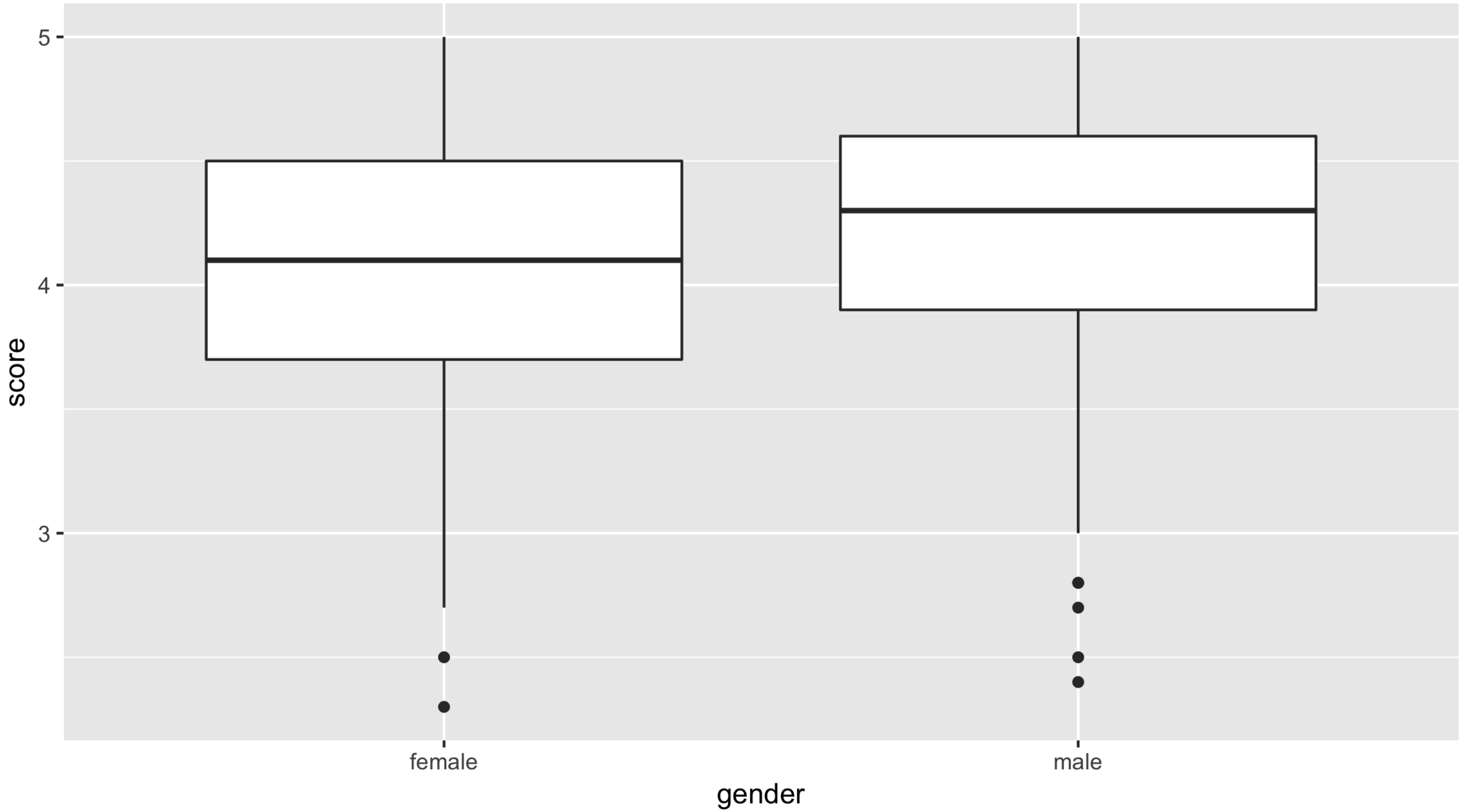
Albert Y. Kim

Assistant Professor of Statistical and Data Sciences

Exploratory data visualization

```
library(ggplot2)
library(dplyr)
library(moderndive)

ggplot(evals, aes(x = gender, y = score)) +
  geom_boxplot() +
  labs(x = "gender", y = "score")
```

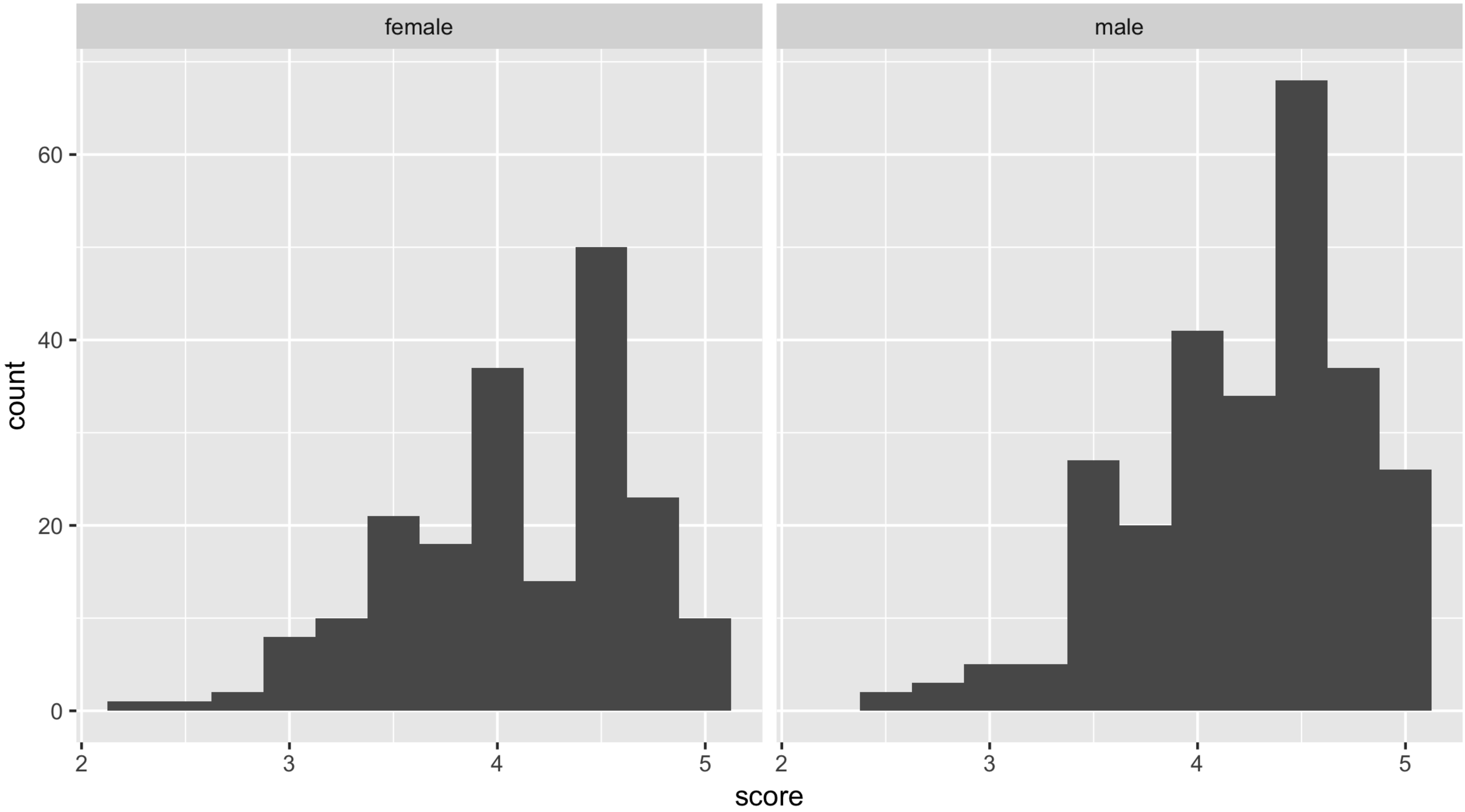
Boxplot of score over gender



Facetted histogram

```r
library(ggplot2)
library(dplyr)
library(moderndive)

ggplot(evals, aes(x = score)) +
  geom_histogram(binwidth = 0.25) +
  facet_wrap(~gender) +
  labs(x = "gender", y = "score")
```

Facetted histogram



Fitting a regression model

```
# Fit regression model
model_score_3 <- lm(score ~ gender, data = evals)

# Get regression table
get_regression_table(model_score_3)
```

```
# A tibble: 2 x 7
  term       estimate std_error statistic p_value...
  <chr>         <dbl>     <dbl>     <dbl>   <dbl>...
1 intercept     4.09      0.039    106.     0...
2 gendermale    0.142     0.051      2.78   0.006...
```

Fitting a regression model

```r
# Compute group means based on gender
evals %>%
  group_by(gender) %>%
  summarize(avg_score = mean(score))
```

```
# A tibble: 2 x 2
  gender avg_score
  <fct>      <dbl>
1 female      4.09
2 male        4.23
```

A different categorical explanatory variable: rank

```
evals %>%
  group_by(rank) %>%
  summarize(n = n())
```

```
# A tibble: 3 x 2
  rank             n
  <fct>        <int>
1 teaching       102
2 tenure track   108
3 tenured        253
```

Let's practice!

MODELING WITH DATA IN THE TIDYVERSE

Predicting teaching score using gender

MODELING WITH DATA IN THE TIDYVERSE

Albert Y. Kim

Assistant Professor of Statistical and Data Sciences

Group means as predictions

```r
library(ggplot2)
library(dplyr)
library(moderndive)

evals %>%
  group_by(gender) %>%
  summarize(mean_score = mean(score), sd_score = sd(score))
```

```
# A tibble: 2 x 3
  gender mean_score sd_score
  <fct>       <dbl>    <dbl>
1 female       4.09    0.564
2 male         4.23    0.522
```

Computing all predicted values and residuals

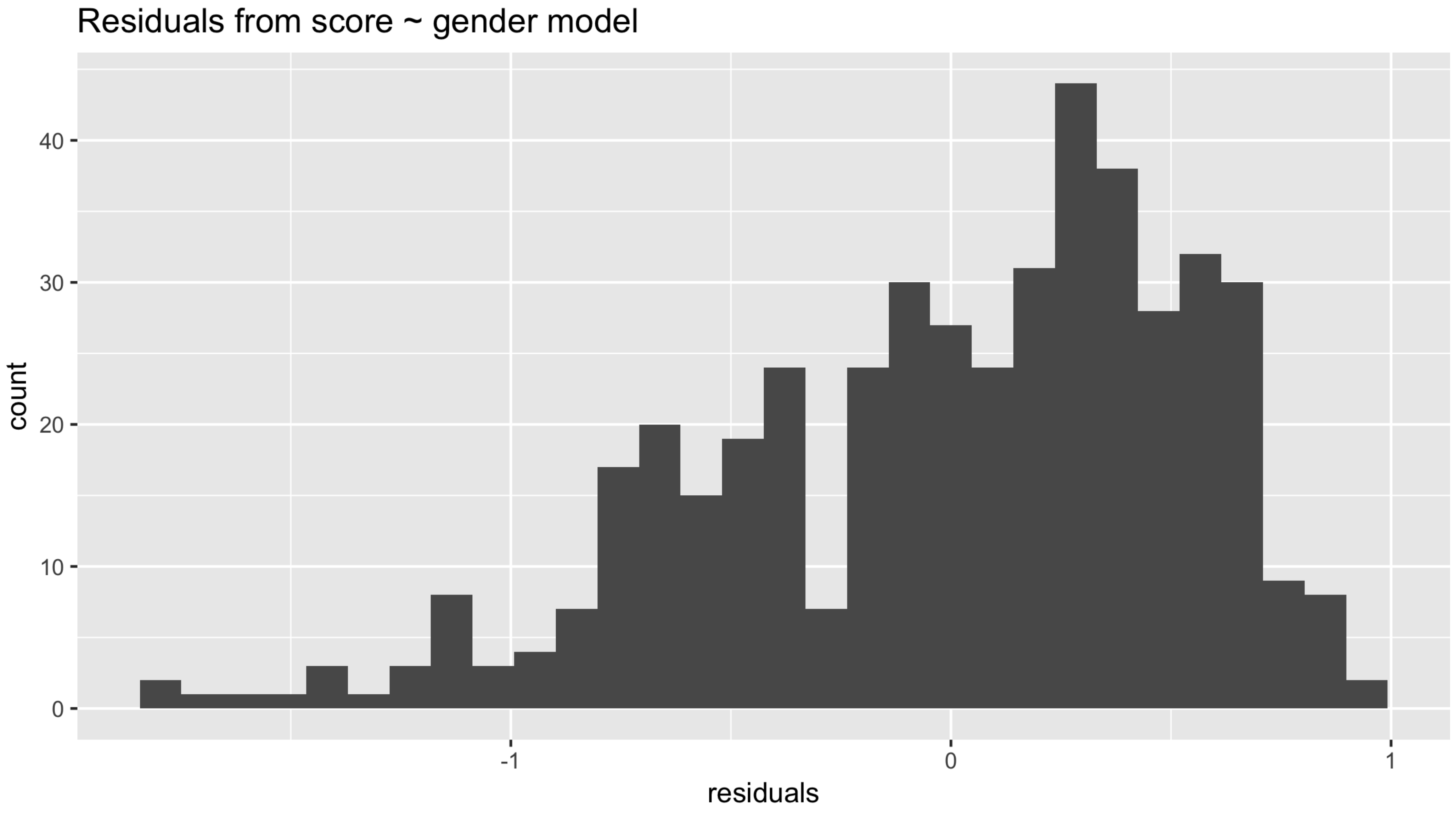
```r
# Fit regression model:
model_score_3 <- lm(score ~ gender, data = evals)

# Get information on each point
get_regression_points(model_score_3)
```

```
# A tibble: 463 x 5
      ID score gender score_hat residual
   <int> <dbl> <fct>      <dbl>    <dbl>
 1     1   4.7 female      4.09    0.607
 2     2   4.1 female      4.09    0.007
 3     3   3.9 female      4.09   -0.193
 4     4   4.8 female      4.09    0.707
 5     5   4.6 male        4.23    0.366
 6     6   4.3 male        4.23    0.066
```

Histogram of residuals

```
# Fit regression model
model_score_3 <- lm(score ~ gender, data = evals)

# Get regression points
model_score_3_points <- get_regression_points(model_score_3)
model_score_3_points
# Plot residuals
ggplot(model_score_3_points, aes(x = residual)) +
  geom_histogram() +
  labs(x = "residuals",
       title = "Residuals from score ~ gender model")
```

Histogram of residuals



Let's practice!

MODELING WITH DATA IN THE TIDYVERSE