MULTIPLE AND LOGISTIC REGRESSION
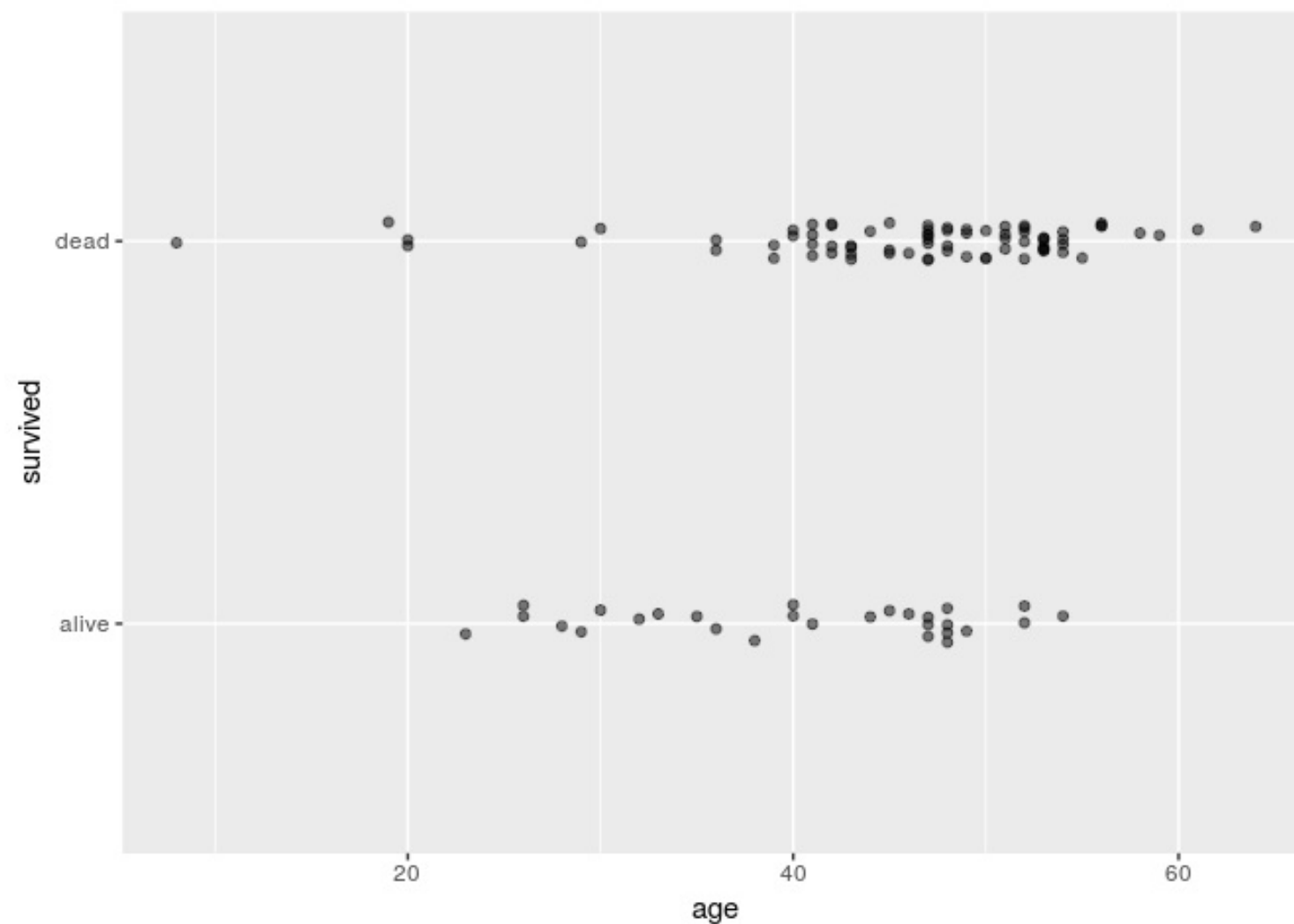
What is logistic regression?

Ben Baumer

Instructor

A categorical response variable

```
ggplot(data = heartTr, aes(x = age, y = survived)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5)
```
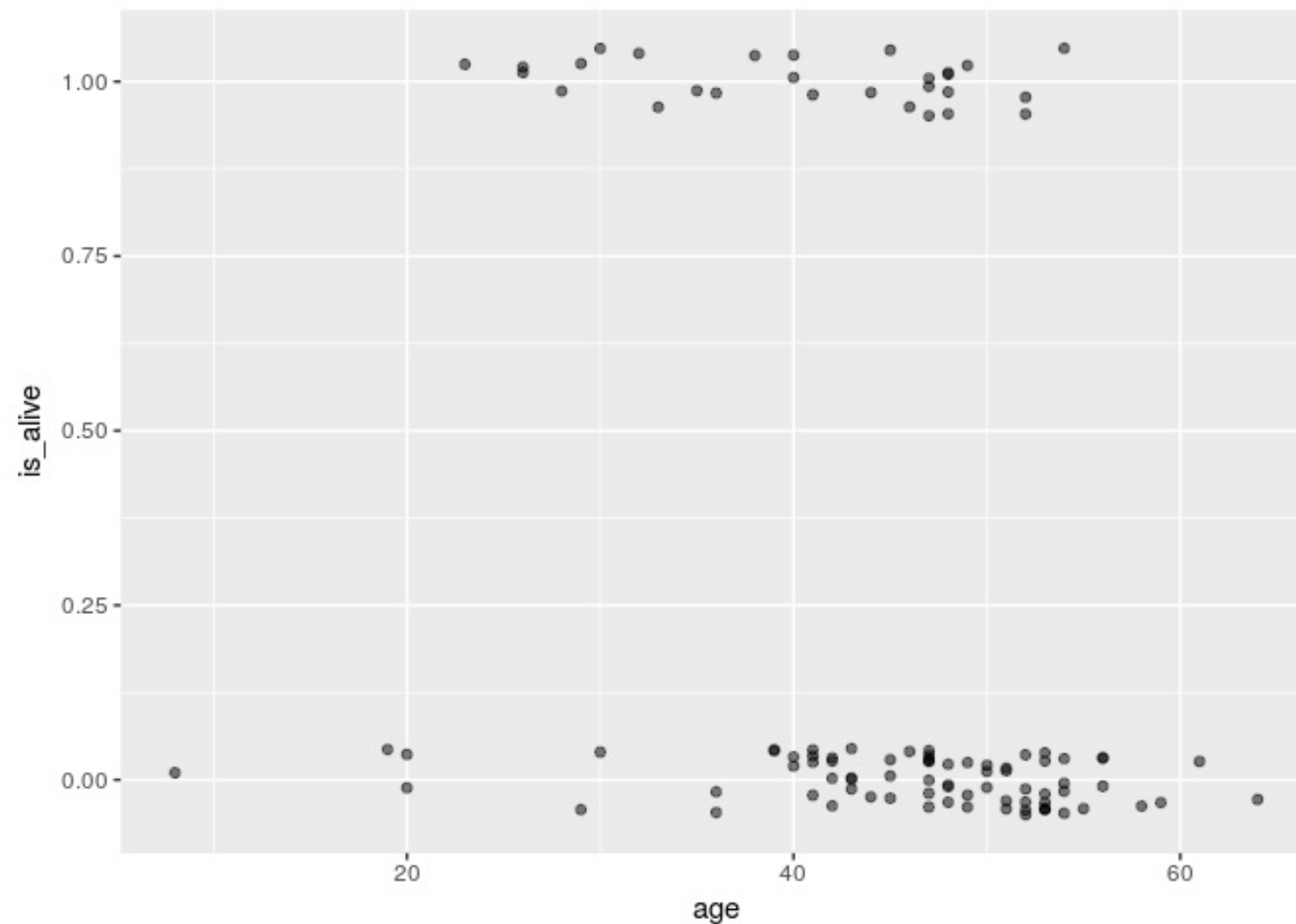
Making a binary variable

```
heartTr <- heartTr %>%
  mutate(is_alive = ifelse(survived == "alive", 1, 0))
```
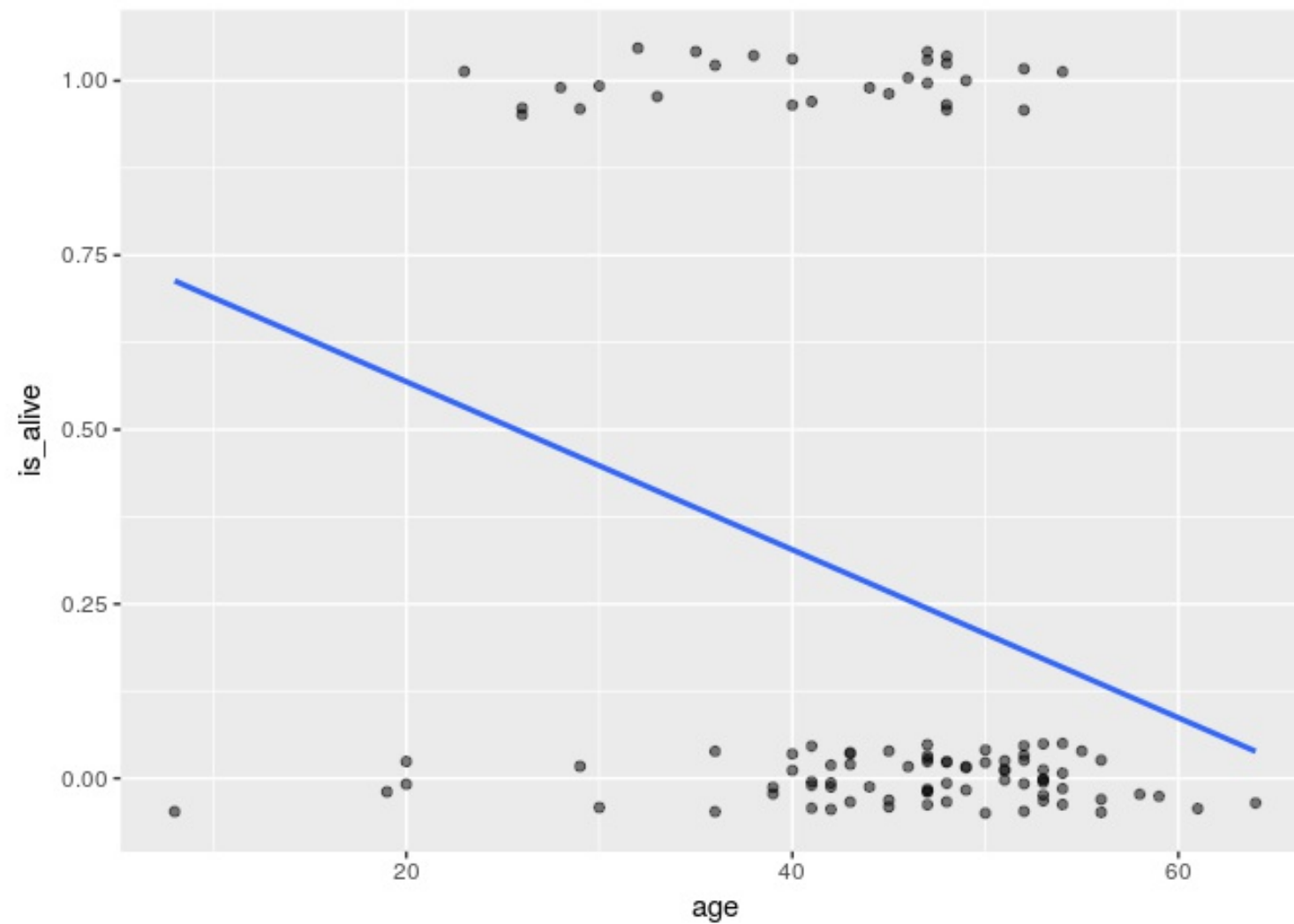
Visualizing a binary response

```
data_space <- ggplot(data = heartTr, aes(x = age, y = is_alive)) +
  geom_jitter(width = 0, height = 0.05, alpha = 0.5)
```

Regression with a binary response

```
data_space +
   geom_smooth(method = "lm", se = 0)
```

Limitations of regression

- Could make non-sensical predictions

- Binary response problematic

Generalized linear models

- generalization of multiple regression
  - model non-normal responses
- special case: logistic regression
  - models binary response
  - uses $logit$ link function
  - $logit(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \cdot x$

Fitting a GLM

```
glm(is_alive ~ age, data = heartTr, family = binomial)

binomial()

    ## Family: binomial
    ## Link function: logit
```

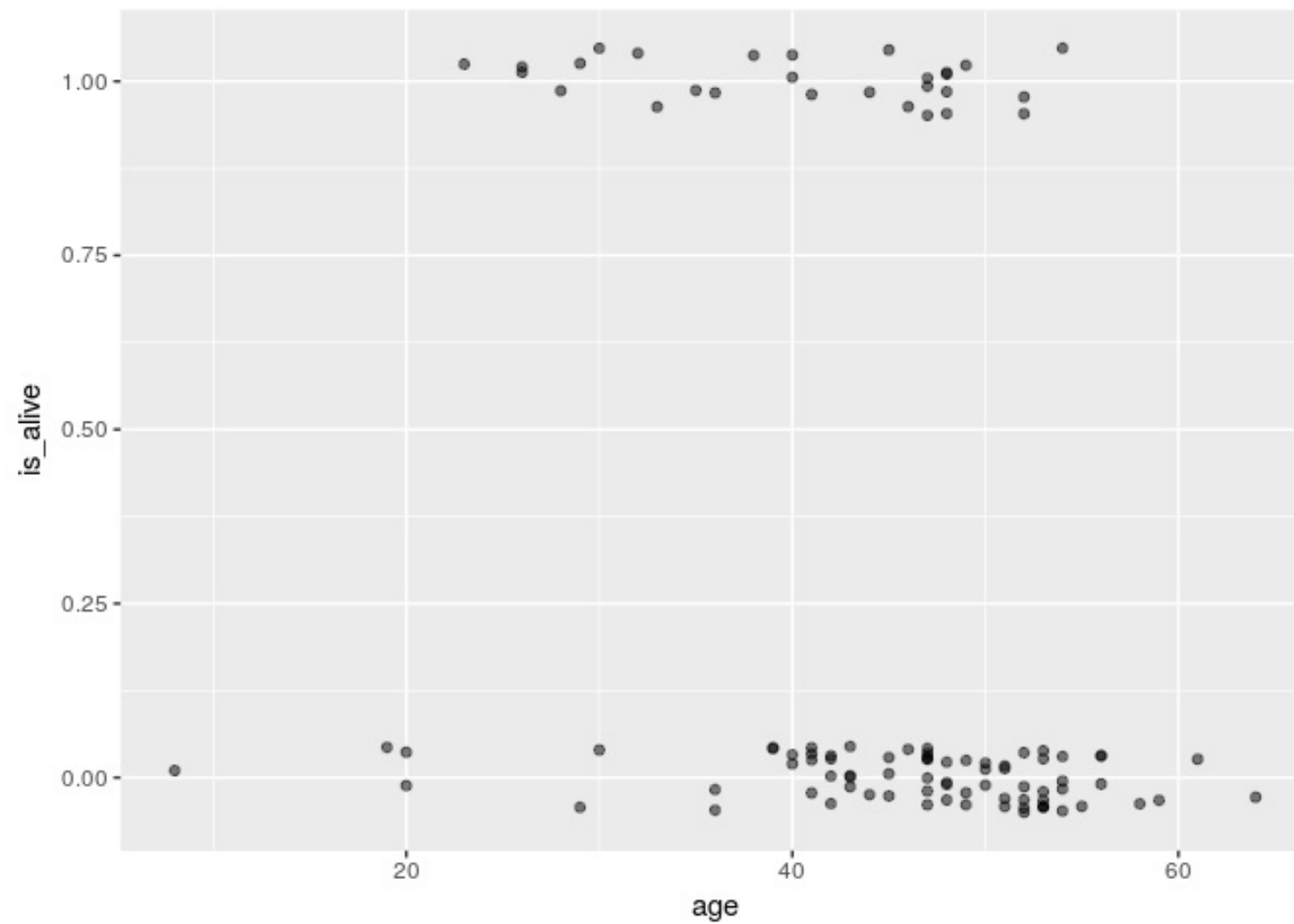MULTIPLE AND LOGISTIC REGRESSION

Let's practice!

MULTIPLE AND LOGISTIC REGRESSION

Visualizing logistic regression
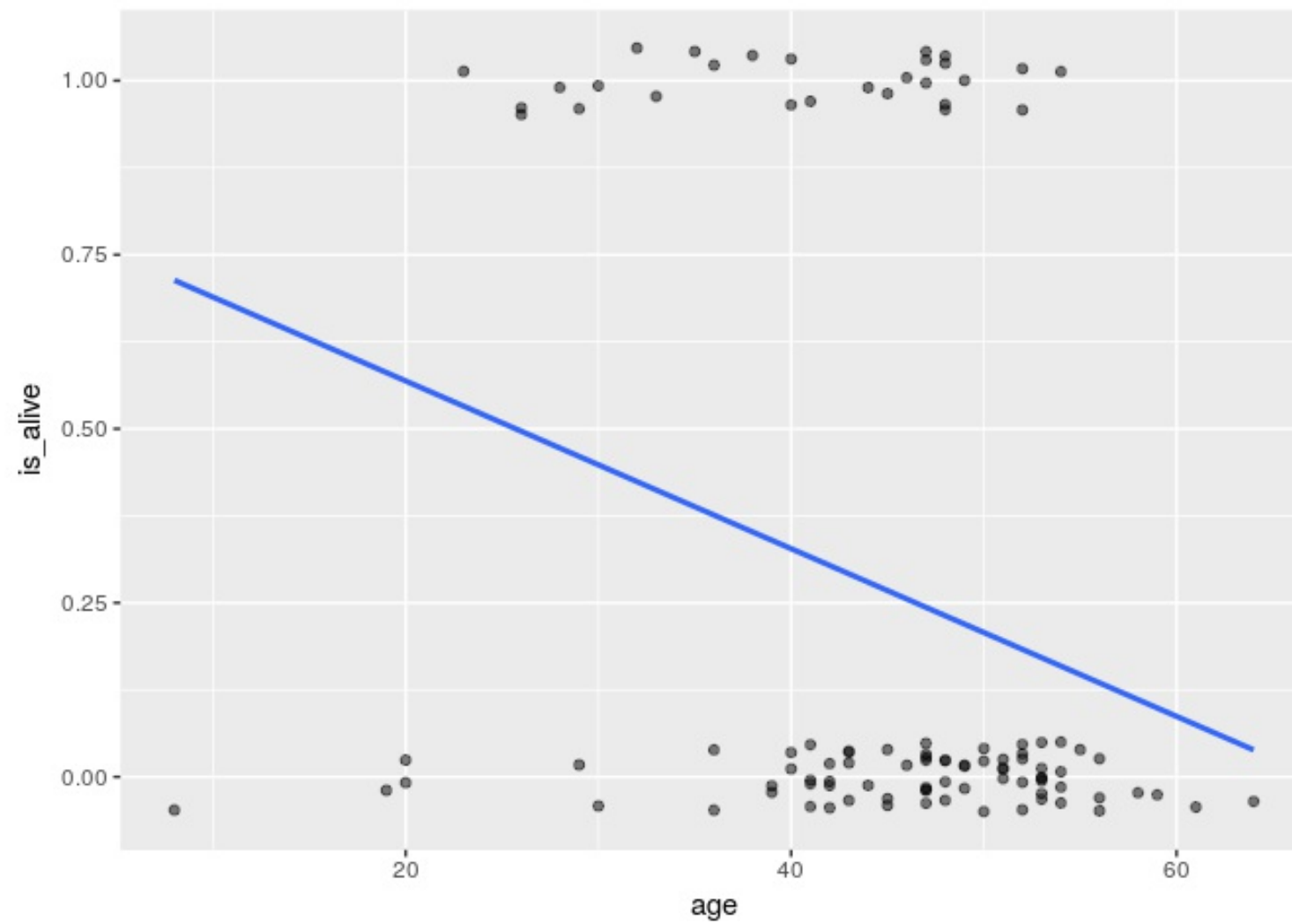
Ben Baumer

Instructor

The data space

```
data_space
```

Regression

```
data_space +
    geom_smooth(method = "lm", se = 0)
```

Using geom_smooth()

```r
data_space +
  geom_smooth(method = "lm", se = 0) +
  geom_smooth(method = "glm", se = 0, color = "red",
              method.args = list(family = "binomial"))
```
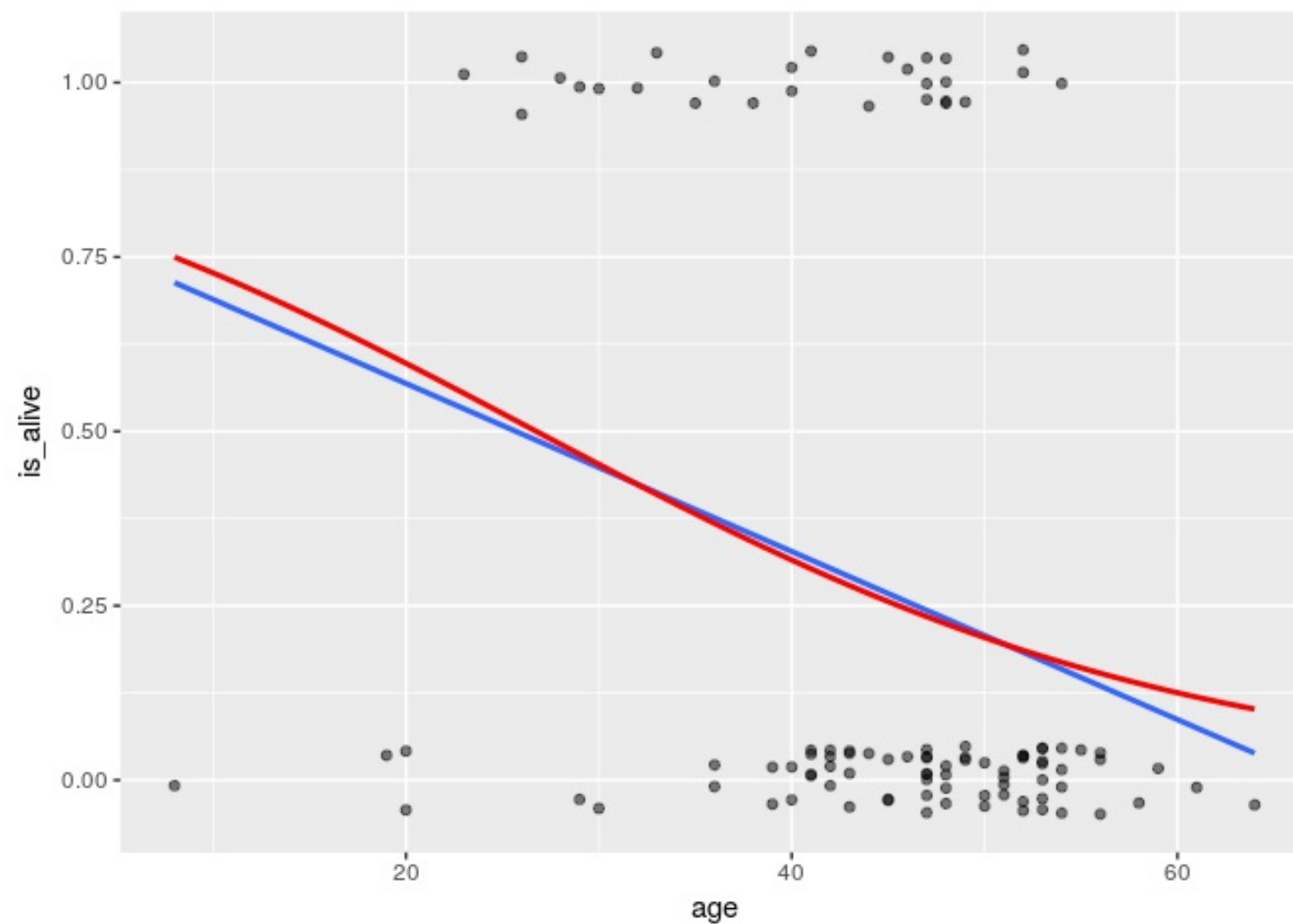
Using bins

data_binned_space
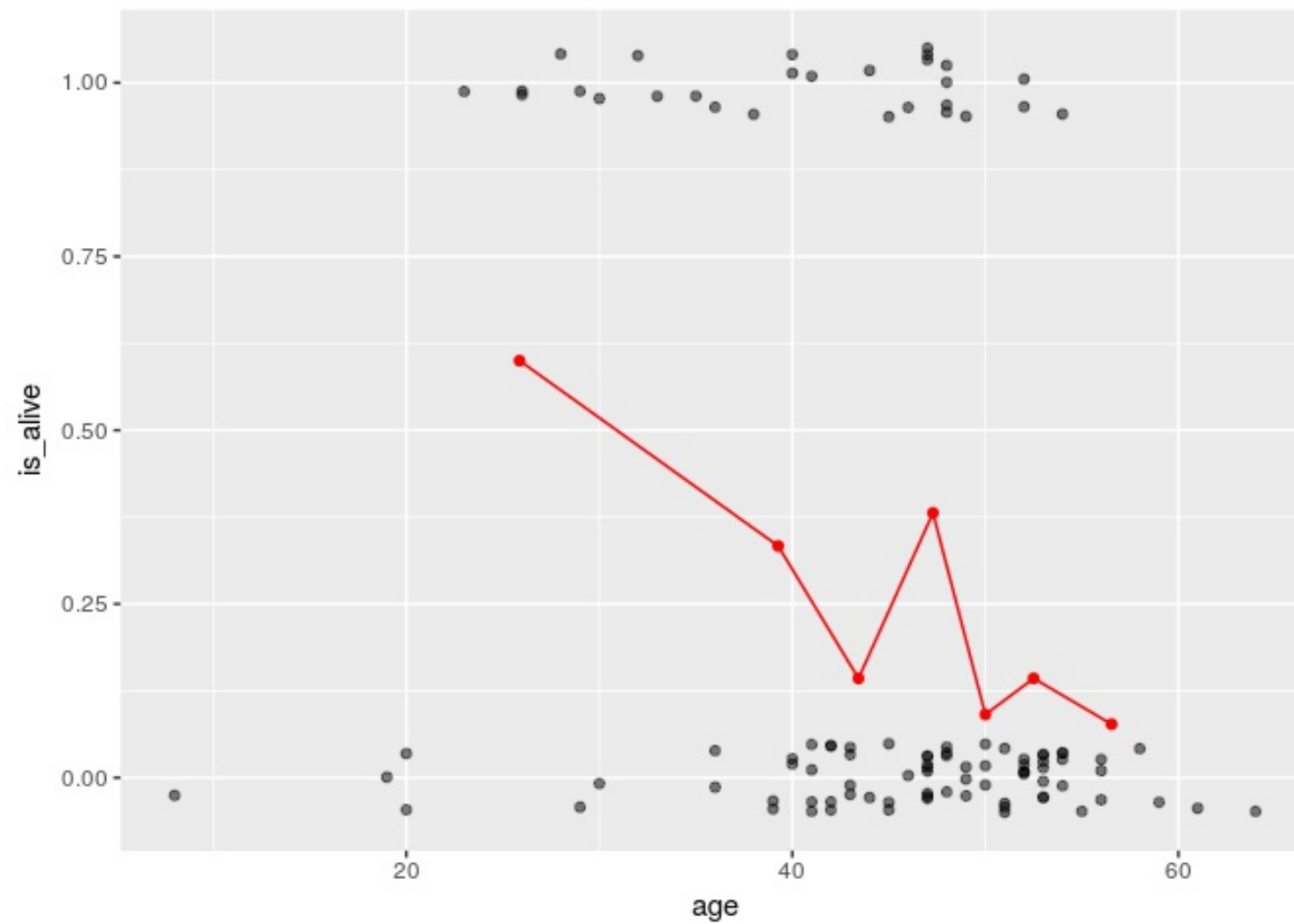
Adding the model to the binned plot

```
data_binned_space +
  geom_line(data = augment(mod, type.predict = "response"),
            aes(y = .fitted), color = "blue")
```
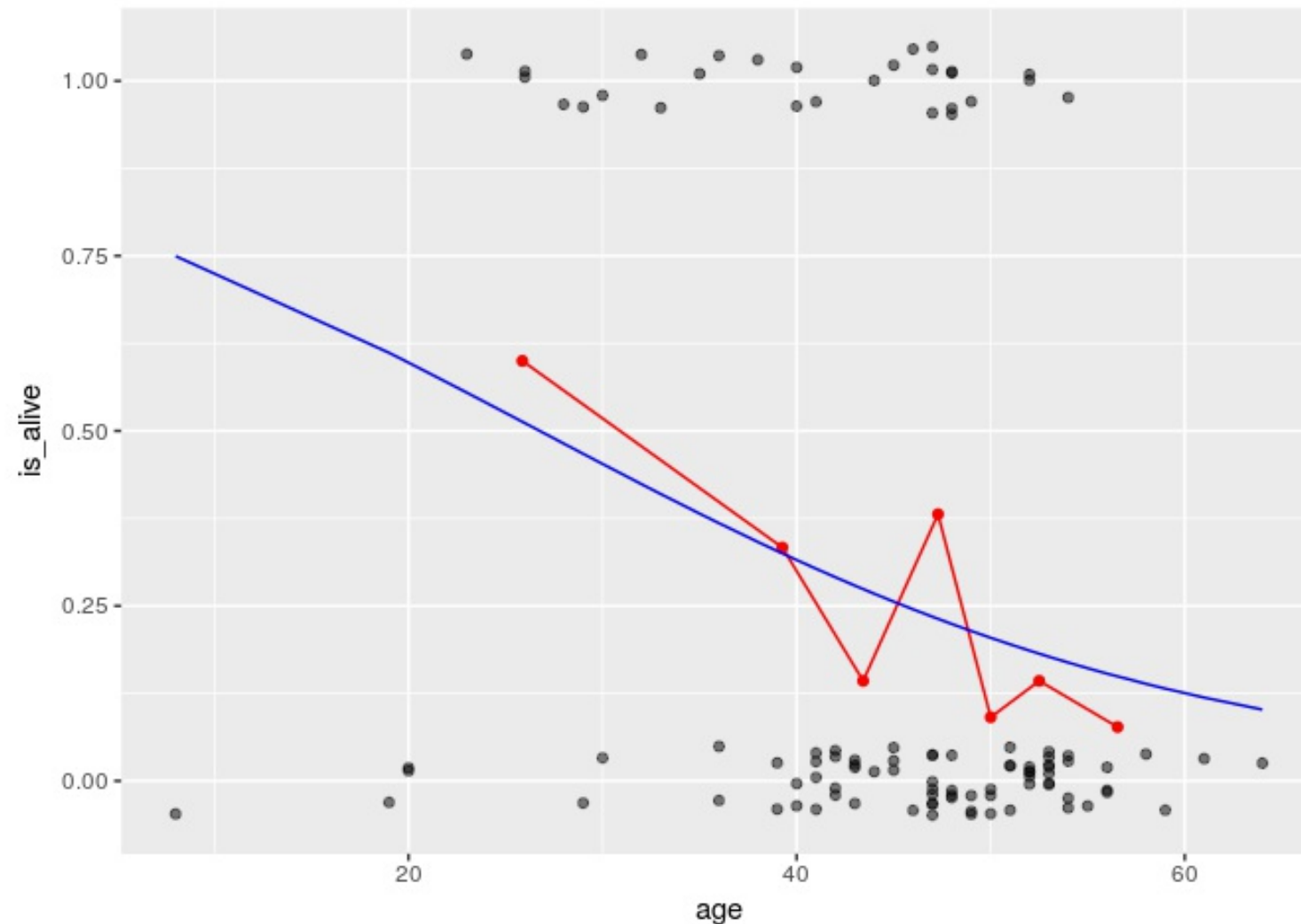
MULTIPLE AND LOGISTIC REGRESSION

Let's practice!

MULTIPLE AND LOGISTIC REGRESSION

Three scales approach to interpretation

Ben Baumer

Instructor

Probability scale

$$\hat{y} = \frac{\exp\left(\hat{\beta}_0 + \hat{\beta}_1 \cdot x\right)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 \cdot x)}$$

```
heartTr_plus <- mod %>%
  augment(type.predict = "response") %>%
  mutate(y_hat = .fitted)
```

Probability scale plot

```
ggplot(heartTr_plus, aes(x = age, y = y_hat)) +
  geom_point() + geom_line() +
  scale_y_continuous("Probability of being alive", limits = c(0, 1))
```
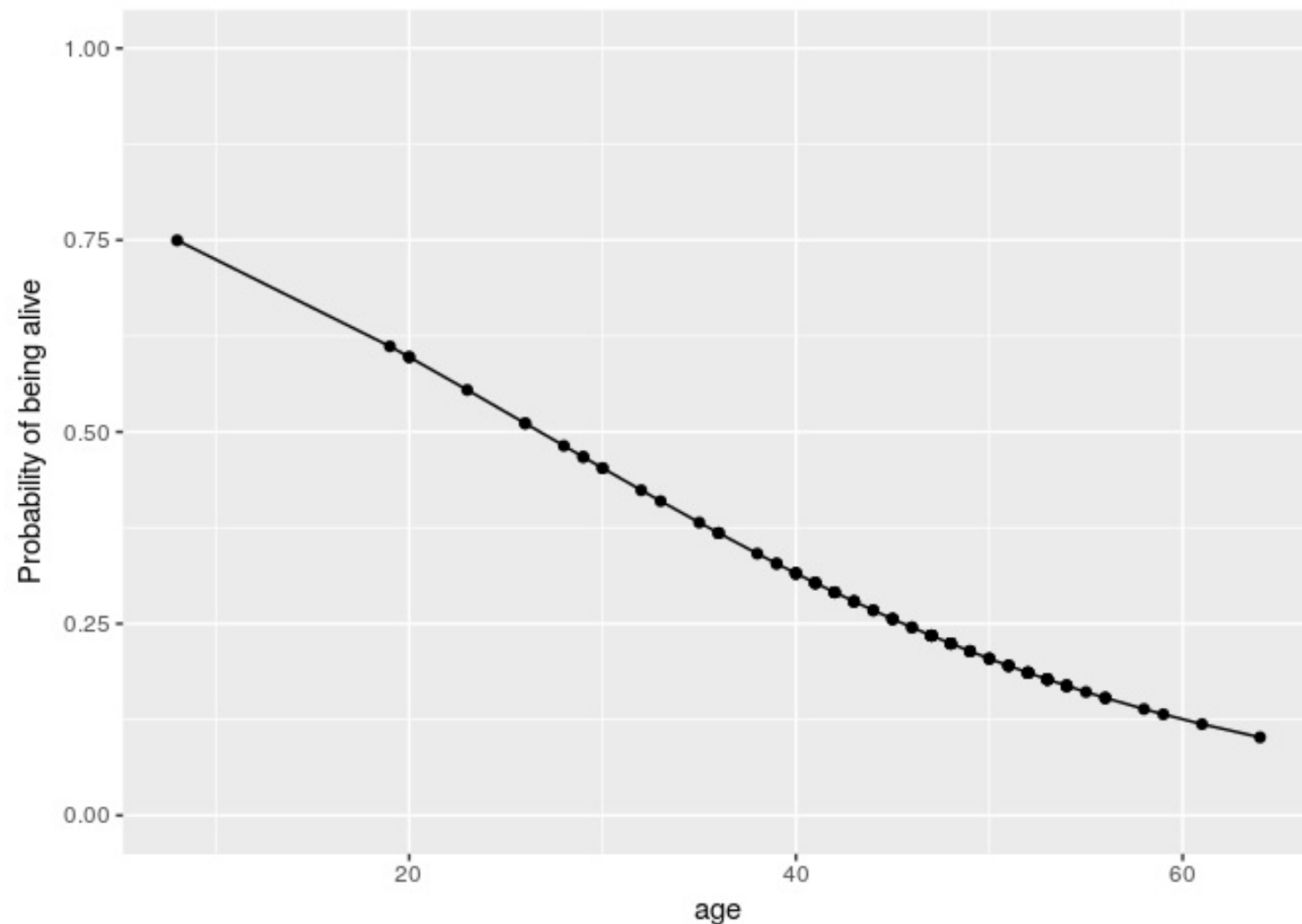
Odds scale

$$odds(\hat{y}) = \frac{\hat{y}}{1 - \hat{y}} = \exp\left(\hat{\beta}_0 + \hat{\beta}_1 \cdot x\right)$$

```
heartTr_plus <- heartTr_plus %>%
  mutate(odds_hat = y_hat / (1 - y_hat))
```

Odds scale plot

```
ggplot(heartTr_plus, aes(x = age, y = odds_hat)) +
   geom_point() + geom_line() +
   scale_y_continuous("Odds of being alive")
```
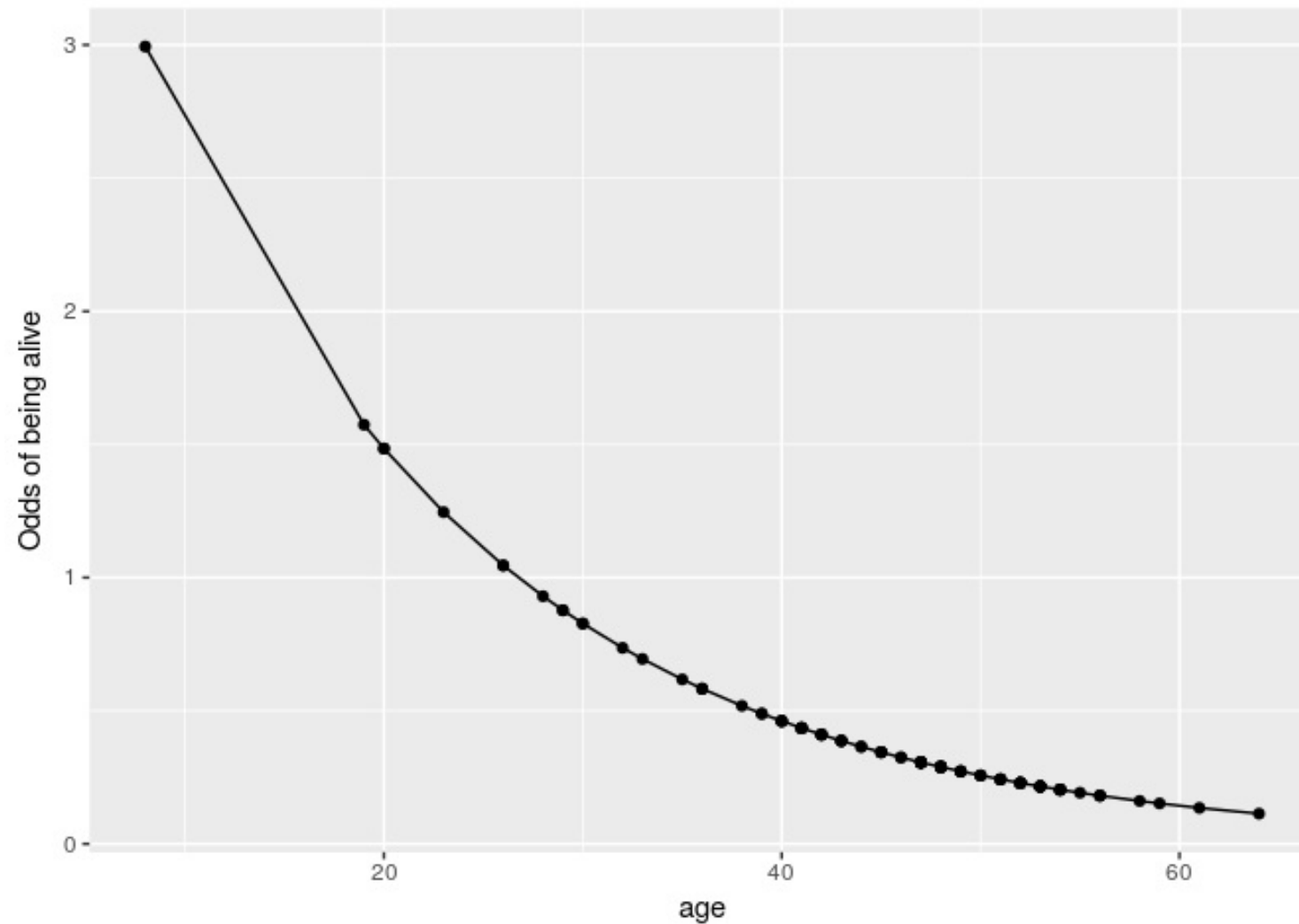
Log-odds scale

$$logit(\hat{y}) = \log\left[\frac{\hat{y}}{1 - \hat{y}}\right] = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$$

```
heartTr_plus <- heartTr_plus %>%
  mutate(log_odds_hat = log(odds_hat))
```

Log-odds plot

```
ggplot(heartTr_plus, aes(x = age, y = log_odds_hat)) +
  geom_point() + geom_line() +
  scale_y_continuous("Log(odds) of being alive")
```
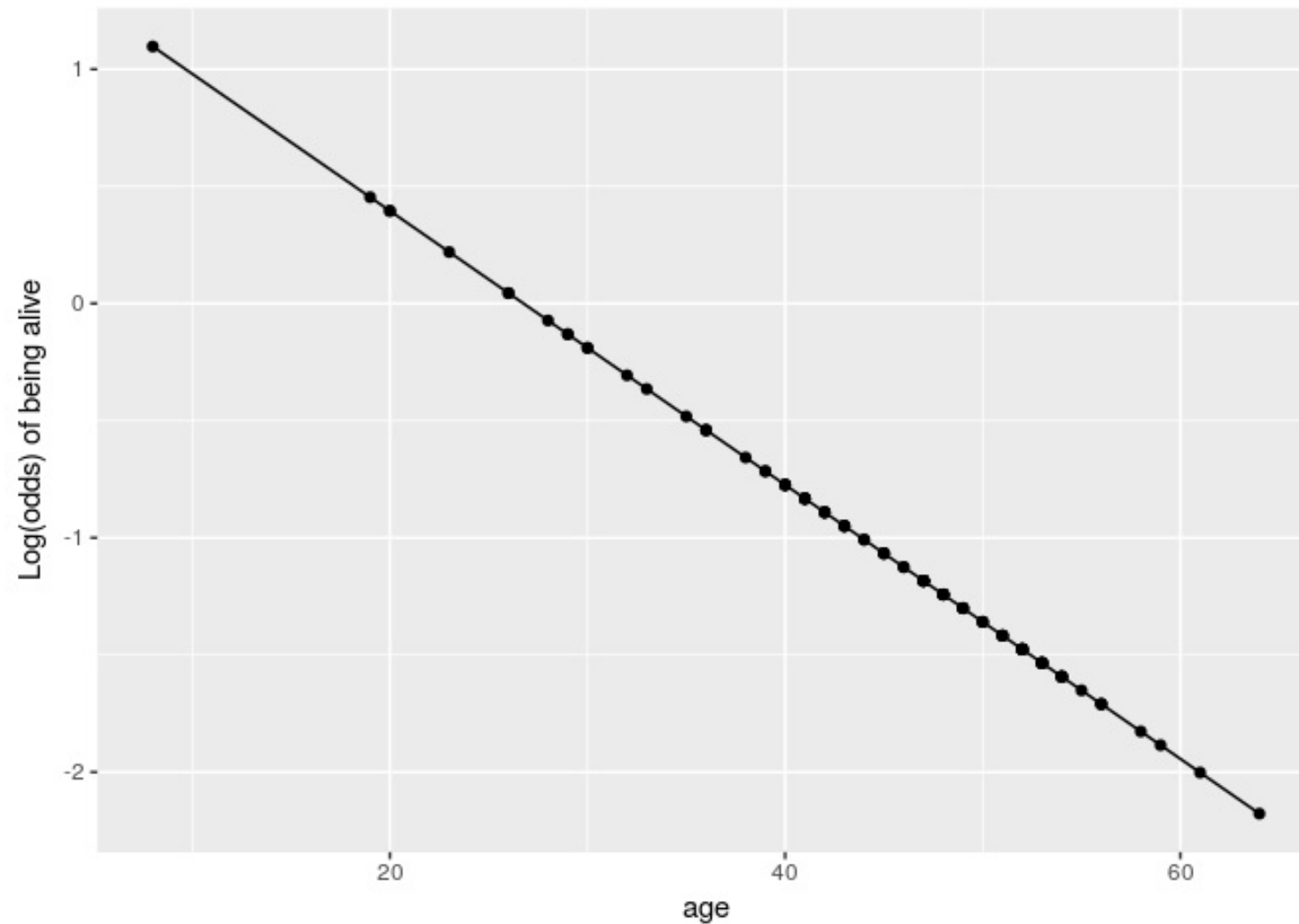
Comparison

- Probability scale

  - scale: intuitive, easy to interpret

  - function: non-linear, hard to interpret

- Odds scale

  - scale: harder to interpret

  - function: exponential, harder to interpret

- Log-odds scale

  - scale: impossible to interpret

  - function: linear, easy to interpret

Odds ratios

$$OR = \frac{odds(\hat{y}|x+1)}{odds(\hat{y}|x)} = \frac{\exp\left(\hat{\beta}_0 + \hat{\beta}_1 \cdot (x+1)\right)}{\exp\left(\hat{\beta}_0 + \hat{\beta}_1 \cdot x\right)} = \exp \beta_1$$

```
exp(coef(mod))

(Intercept)           age
  4.7797050     0.9432099
```

MULTIPLE AND LOGISTIC REGRESSION

Let's practice!

MULTIPLE AND LOGISTIC REGRESSION

Using a logistic model

Ben Baumer

Instructor

Learning from a model

```
mod <- glm(is_alive ~ age + transplant, data = heartTr, family = binomial)

exp(coef(mod))

    ##          (Intercept)                     age transplanttreatment
    ##            2.6461676               0.9265153           6.1914009
```

# Using augment()

```
# log-odds scale
augment(mod)

   ##       is_alive age transplant     .fitted    .se.fit     .resid       .hat
   ## 1           0   53    control -3.0720949 0.7196746 -0.3009421 0.02191525
   ## 2           0   43    control -2.3088482 0.5992811 -0.4352986 0.02952903
   ## 3           0   52    control -2.9957702 0.7044109 -0.3123727 0.02250241
   ## 4           0   52    control -2.9957702 0.7044109 -0.3123727 0.02250241
   ## 5           0   54    control -3.1484196 0.7355066 -0.2899116 0.02134668
   ## 6           0   36    control -1.7745756 0.5704650 -0.5596850 0.04033929
   ## 7           0   47    control -2.6141469 0.6379934 -0.3759601 0.02587839
   ## 8           0   41  treatment -0.3330375 0.2810663 -1.0396433 0.01921191
   ## 9           0   47    control -2.6141469 0.6379934 -0.3759601 0.02587839
   ## 10          0   51    control -2.9194456 0.6897533 -0.3242157 0.02311200
```

Making probabilistic predictions

```
# probability scale
augment(mod, type.predict = "response")

   ##       is_alive age transplant     .fitted      .se.fit      .resid        .hat
   ## 1           0  53    control 0.04427310 0.03045159 -0.3009421 0.02191525
   ## 2           0  43    control 0.09039280 0.04927406 -0.4352986 0.02952903
   ## 3           0  52    control 0.04761733 0.03194498 -0.3123727 0.02250241
   ## 4           0  52    control 0.04761733 0.03194498 -0.3123727 0.02250241
   ## 5           0  54    control 0.04115360 0.02902308 -0.2899116 0.02134668
   ## 6           0  36    control 0.14497423 0.07071297 -0.5596850 0.04033929
   ## 7           0  47    control 0.06823348 0.04056214 -0.3759601 0.02587839
   ## 8           0  41  treatment 0.41750173 0.06835365 -1.0396433 0.01921191
   ## 9           0  47    control 0.06823348 0.04056214 -0.3759601 0.02587839
   ## 10          0  51    control 0.05120063 0.03350761 -0.3242157 0.02311200
```
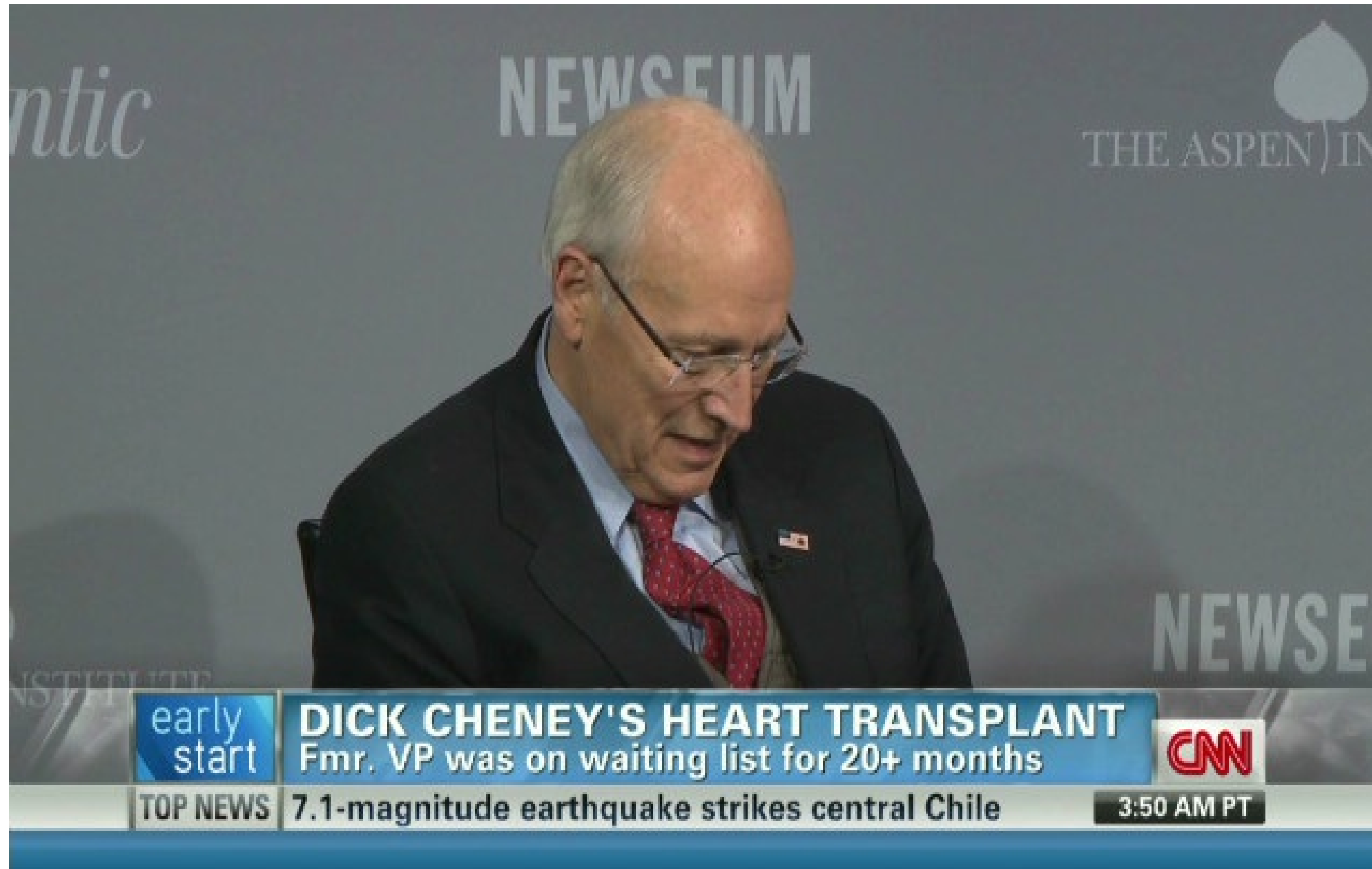
Out-of-sample predictions

```r
cheney <- data.frame(age = 71, transplant = "treatment")

augment(mod, newdata = cheney, type.predict = "response")

    ##   age transplant    .fitted    .se.fit
    ## 1  71  treatment 0.06768681 0.04572512
```

Making binary predictions

```
mod_plus <- augment(mod, type.predict = "response") %>%
  mutate(alive_hat = round(.fitted))

mod_plus %>%
  select(is_alive, age, transplant, .fitted, alive_hat)

   ##      is_alive age transplant    .fitted alive_hat
   ## 1           0  53    control 0.04427310         0
   ## 2           0  43    control 0.09039280         0
   ## 3           0  52    control 0.04761733         0
   ## 4           0  52    control 0.04761733         0
   ## 5           0  54    control 0.04115360         0
   ## 6           0  36    control 0.14497423         0
   ## 7           0  47    control 0.06823348         0
   ## 8           0  41  treatment 0.41750173         0
   ## 9           0  47    control 0.06823348         0
   ## 10          0  51    control 0.05120063         0
```

Confusion matrix

```
mod_plus %>%
  select(is_alive, alive_hat) %>%
  table()

  ##          alive_hat
  ## is_alive  0  1
  ##        0 71  4
  ##        1 20  8
```

MULTIPLE AND LOGISTIC REGRESSION

Let's practice!