Confidence in Models

Author: Nicholas G Reich

This material is part of the **statsTeachR** *project*

Today's Lecture

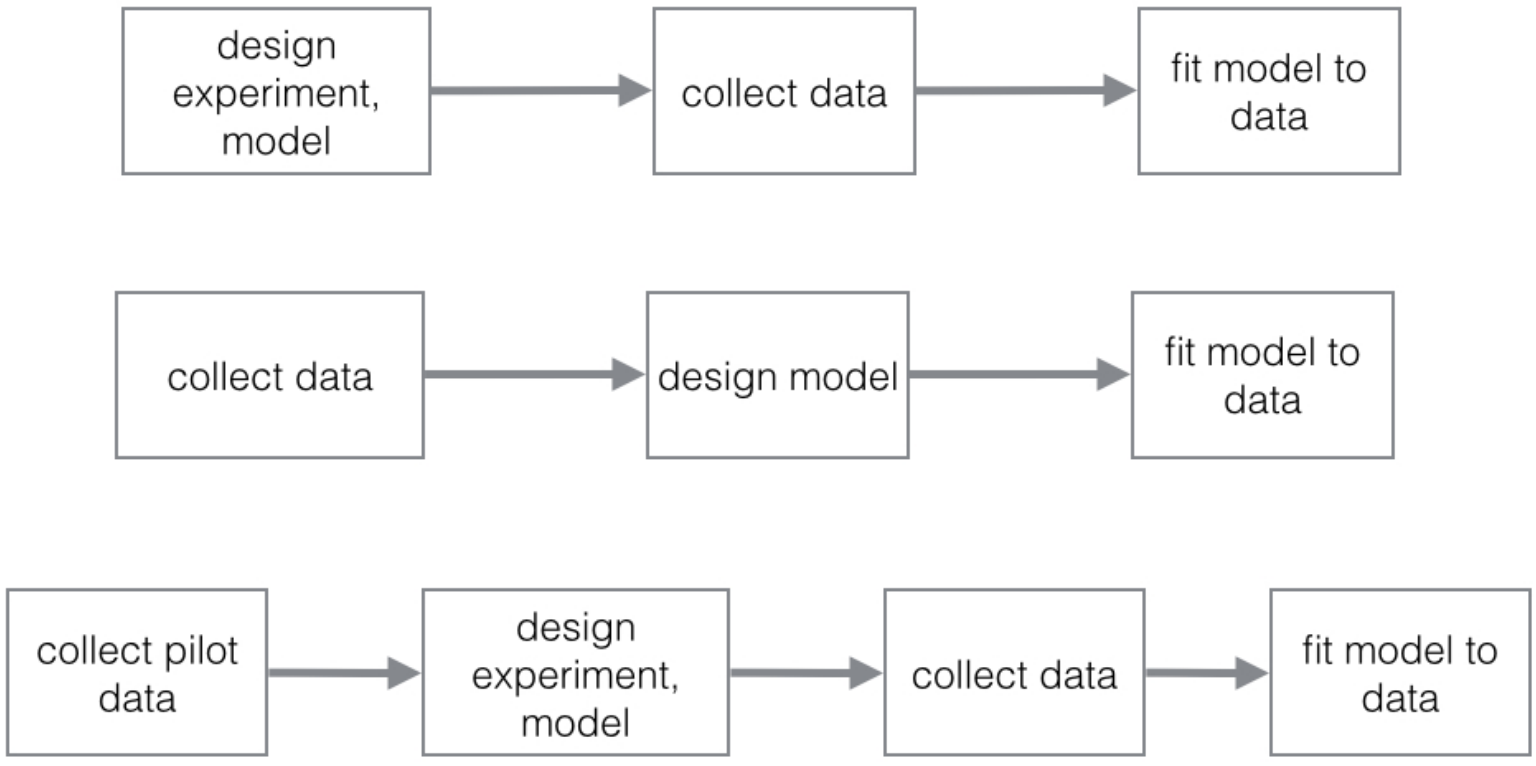
It aint what you dont know that gets you into trouble. Its what you know for sure that just aint so. -Mark Twain

Today's central question

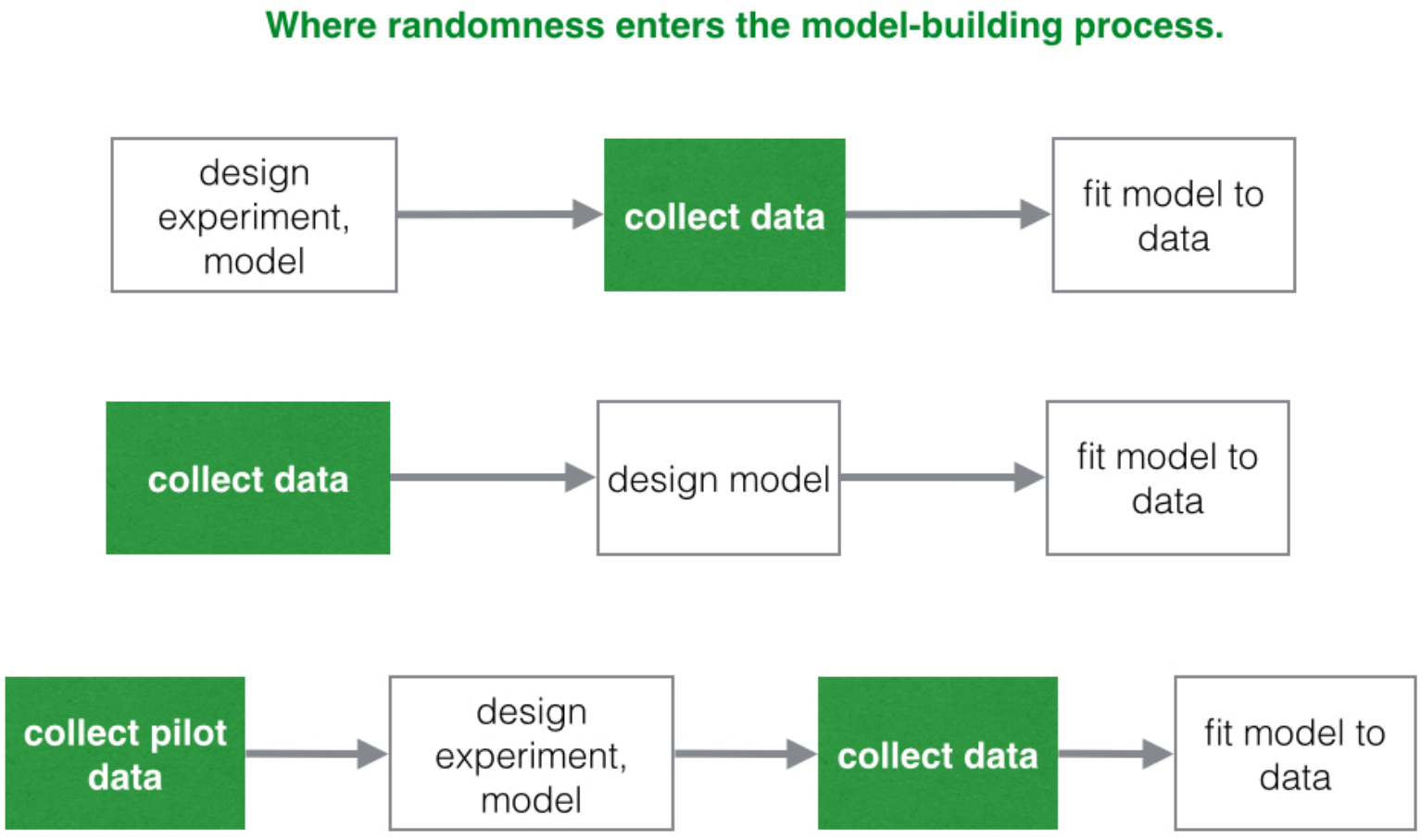
What do linear regression models tell us about what we know and do not know about a particular dataset?

Based loosely on Kaplan, Chapter 12.

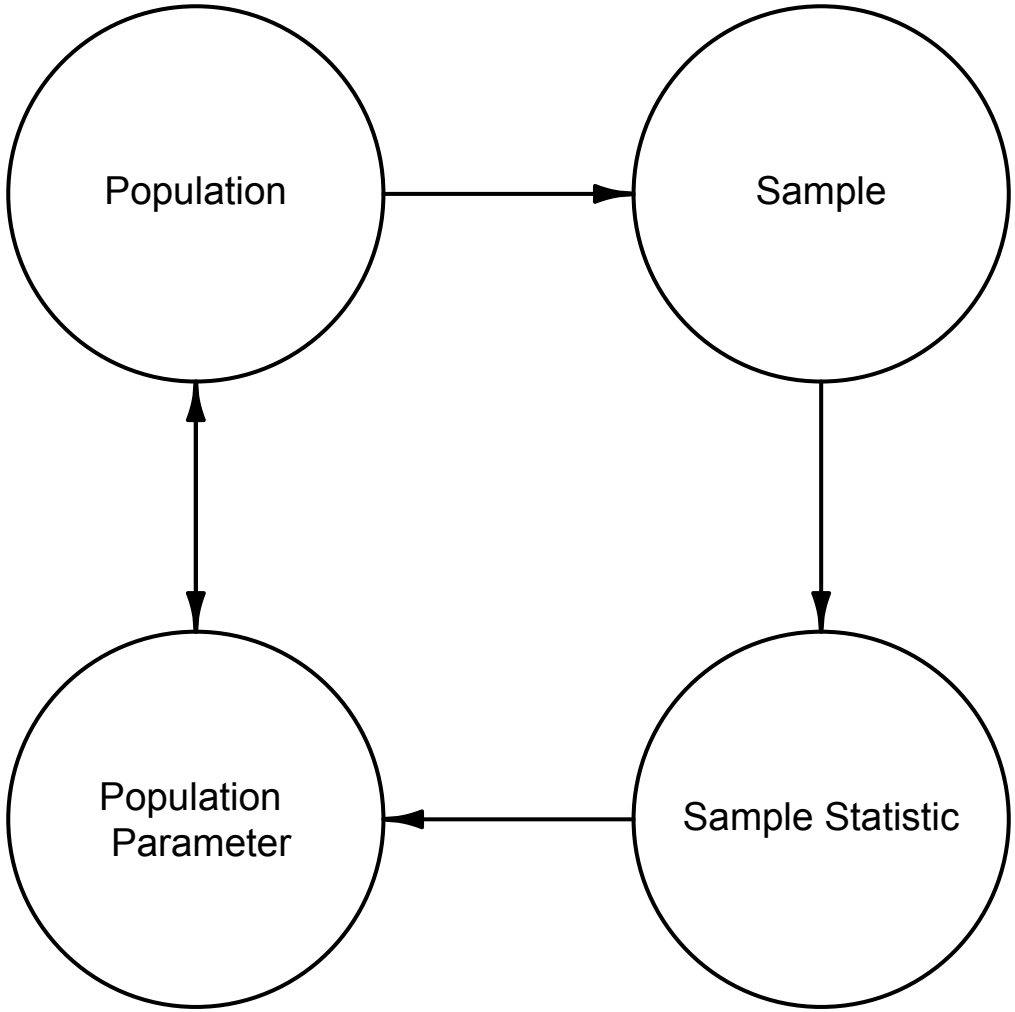
Process of building a statistical model



Process of building a statistical model



Circle of Life



How much will a sample tell us about the population

In practice we can very rarely sample the entire population of interest.
We can create a simple example of a population as a illustration.
E.g. 8636 running times for the Cherry Blossom Ten Mile race in Washington DC in 2005:

```
race <- mosaicData::TenMileRace
head(race)
```

```
##   state time  net age sex
## 1    VA 6060 5978  12   M
## 2    MD 4515 4457  13   M
## 3    VA 5026 4928  13   M
## 4    MD 4229 4229  14   M
## 5    MD 5293 5076  14   M
## 6    VA 6234 5968  14   M
```

A simple model for the race data

$$net \sim age + sex$$

or

$$net = \beta_0 + \beta_1 \cdot age + \beta_2 \cdot sex$$

Using all the data, i.e. the entire "population"

```
fm <- lm(net ~ age + sex, data=race)
summary(fm)$coef
```

```
##                Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 5339.15545 35.0486629 152.33550  0.000000e+00
## age           16.89362  0.9443776  17.88863  2.660668e-70
## sexM        -726.61948 20.0181263 -36.29808 1.281442e-268
```

Let's talk about SEs!

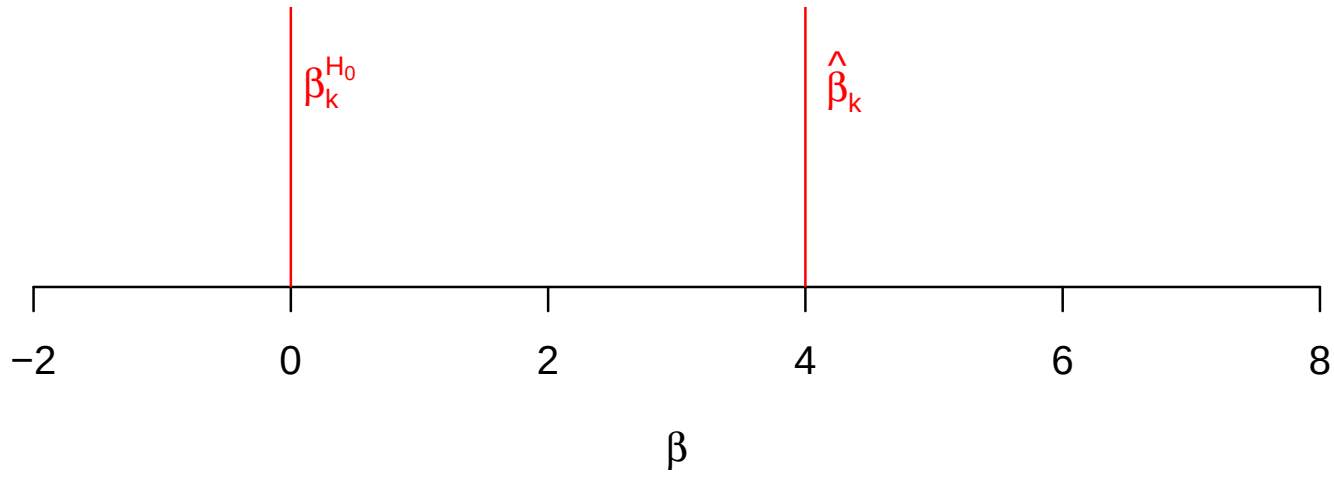
- We can use "statistical inference" to gauge our uncertainty about our estimated βs.
- Intuitively, we want to estimate how much uncertainty we expect to have about each β in our model.
- Out of the box, R gives you p-values to test hypothses of the form: $H_0 : \beta_k = 0$.
- The more uncertainty we have about a specific $\hat{\beta}_k$, the less likely we are to reject the null hypothesis.

```
summary(fm)$coef
```

```
##                Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 5339.15545 35.0486629 152.33550  0.000000e+00
## age           16.89362  0.9443776  17.88863  2.660668e-70
## sexM        -726.61948 20.0181263 -36.29808 1.281442e-268
```

Hypothesis testing for $\hat{\beta}_k$

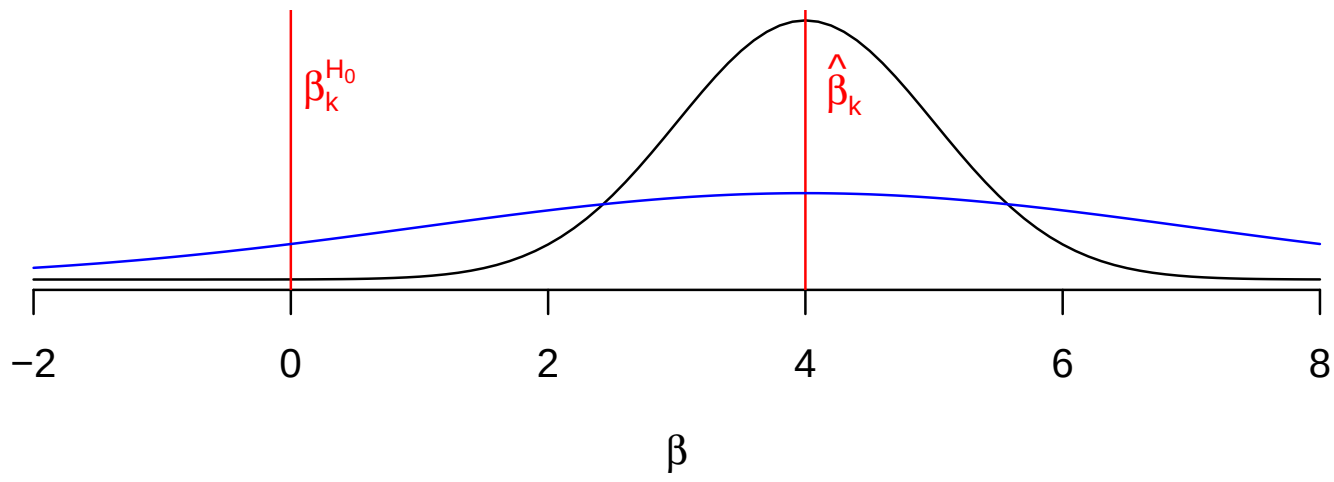
Null hypothesis (H_0): $\beta_k = 0$



We need a measure of uncertainty about our point-estimates to evaluate "statistical significance", which is different from "practical significance".

Hypothesis testing for $\hat{\beta}_k$

Null hypothesis (H_0): $\beta_k = 0$



Sampling distributions measure our uncertainty. But we have to come up with ways to approximate them.

Factors influencing our uncertainty about $\hat{\beta}_k$

How do each of these factors influence uncertainty about $\hat{\beta}_k$

- Increased sample size:
- Increased variability in y;
- Increased variability in x:
- Bias in your sampling of observations:

Sampling distribution terminology

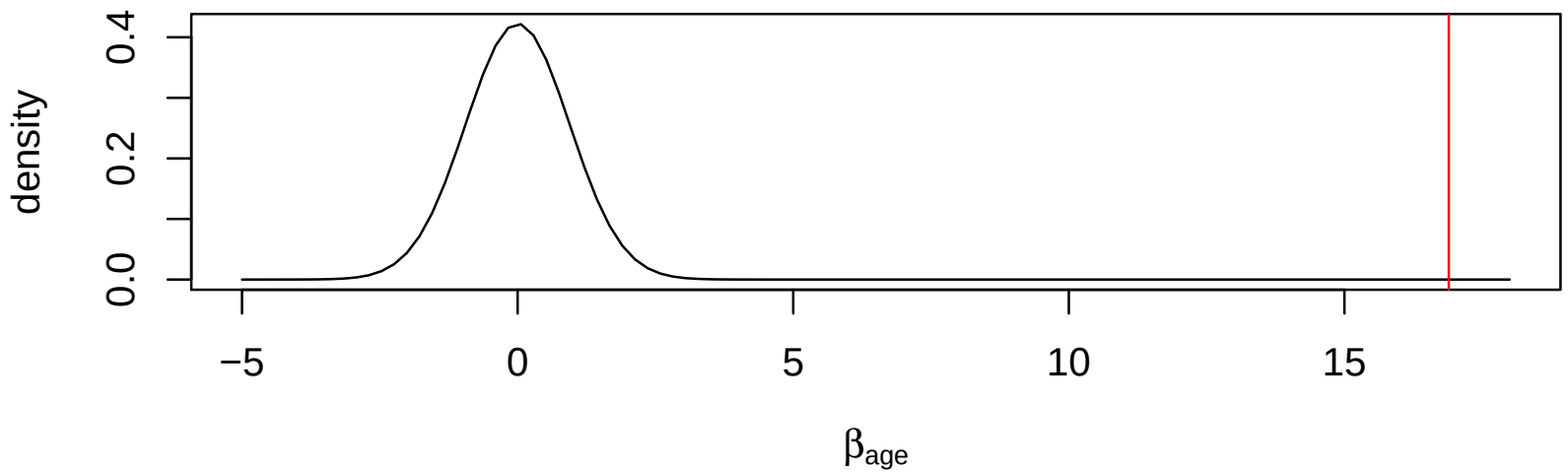
Really important vocabulary!

- **sampling distribution**: the distribution of an estimated parameter, reflecting the randomness of the sampling (data collection) process.
- **standard error**: the standard deviation of a sampling distribution, measures the precision of our estimate or the amount of information we have about the parameter.
- **margin of error**: the half-width of the confidence interval
- **point estimate**: the exact numerical value that represents our best guess at the true parameter value. (In regression, this is the least-squares estimate of our β.)
- **p-value**: the probability of observing a value of as or more extreme as what you did observe in your data, assuming the null hypothesis is true.

Standard inference about β_k in R

Assuming $H_0 : \beta_{age} = 0$ is true we can use the estimated SE to approximate the sampling distribution as a t-distribution. We see that if H_0 were true, our observed $\hat{\beta}_{age}$ would be VERY unlikely.

```
##                Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 5339.15545 35.0486629 152.33550  0.000000e+00
## age           16.89362  0.9443776  17.88863  2.660668e-70
## sexM        -726.61948 20.0181263 -36.29808 1.281442e-268
```



Different inference techniques

Standard inference

Uses a mathematical approximation to the sampling distribution that gets more reliable with larger sample sizes. In many practical data analysis situations, this standard inference procedure works just fine.

Permutation-based inference

Uses numerical/simulated approximation to the sampling distribution under the null hypothesis. This is what we did for the Lady Tasting Tea example.

Bootstrap inference

Similar to permutation based inference but it does not permute the data and simulate when H_0 is true. Instead, it resamples your data to estimate the standard error.

Permutation and bootstrap inference can be particularly useful when you have a procedure that does not have a tidy, closed-form solution, unlike regression which does.

What if our dataset was only a fraction of the pop'n?

$$net = \beta_0 + \beta_1 \cdot age + \beta_2 \cdot sex$$

This is the model fit to the entire population

```
coef(fm)

## (Intercept)         age        sexM
##  5339.15545    16.89362  -726.61948
```

But what if it was just fit to a subsample of the population?
[See activity...]

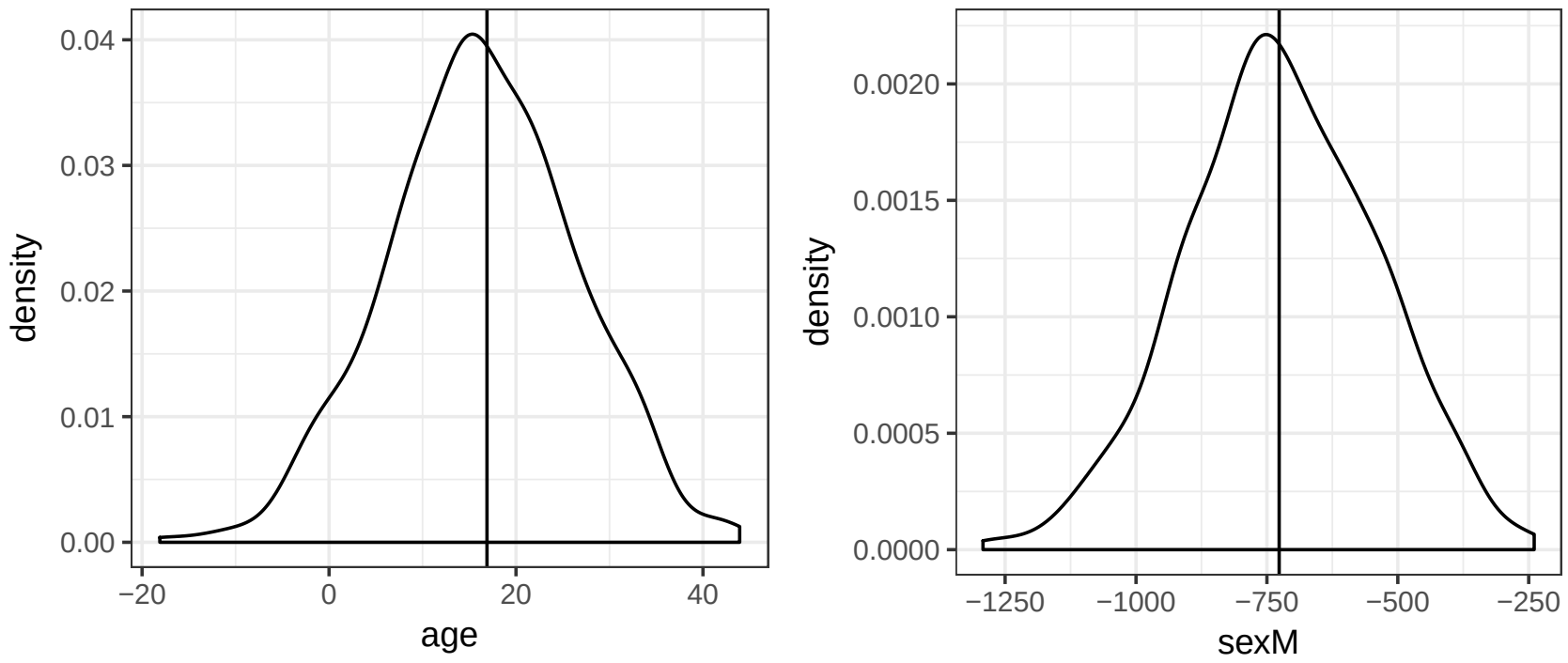
What if our dataset was only a fraction of the pop'n?

$$net = \beta_0 + \beta_1 \cdot age + \beta_2 \cdot sex$$

```
library(mosaic)
s <- do(500) * lm(net ~ age + sex, data=sample(race, 100))
head(s[,1:5])

##   Intercept      age      sexM    sigma r.squared
## 1  5115.484 21.23172 -752.0379 868.1021 0.1609840
## 2  5500.219 18.01911 -998.9722 978.5163 0.1917701
## 3  4974.495 27.71411 -557.4720 750.1912 0.2065647
## 4  5384.336 16.93553 -764.5308 695.6753 0.2604681
## 5  5128.338 24.31173 -927.9286 870.9974 0.2634031
## 6  4701.265 29.01096 -818.3494 915.7264 0.1918550
```

The sampling distribution of the βs

```
library(gridExtra)
p1 <- ggplot(s) + geom_density(aes(x=age)) +
    geom_vline(xintercept=coef(fm)["age"])
p2 <- ggplot(s) + geom_density(aes(x=sexM)) +
    geom_vline(xintercept=coef(fm)["sexM"])
grid.arrange(p1, p2, nrow=1)
```



The standard error depends on...

The quality of the data

- If your data collection process involves a measurement process that contains a lot of error (just noise, not biased observations on average), how will that impact the standard errors?
- In the setting of the race, what measurement procedures might lead to less or more error?

The standard error depends on...

The quality of the model

Models with lower residual error tend to have lower standard errors than ones with larger residual error.

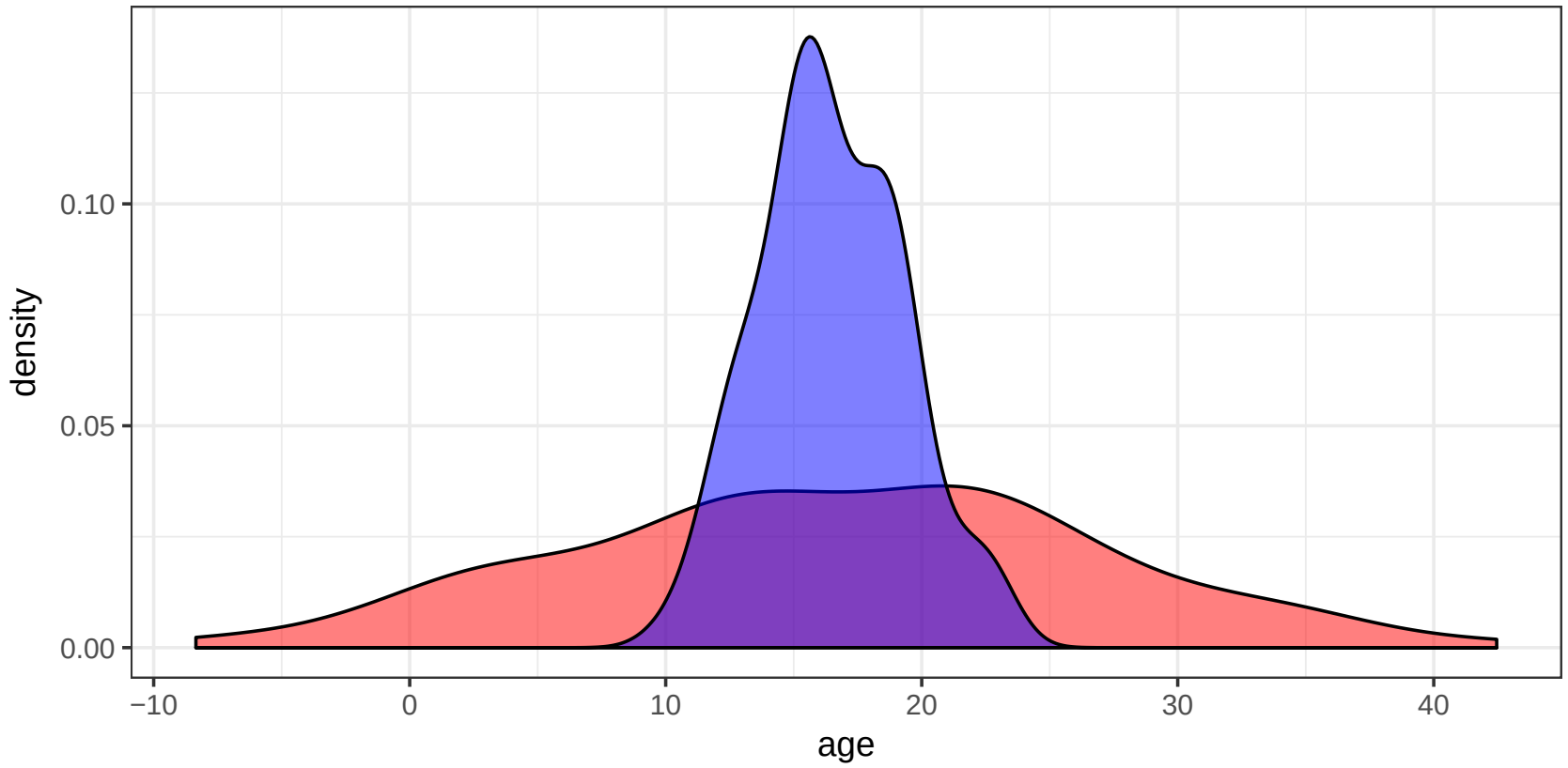
The standard error depends on...

The sample size

As the sample size increases, what happens to the standard error?

The standard error and sample size

```
s100 <- do(100) * lm(net ~ age + sex, data=sample(race, 100))
s1000 <- do(100) * lm(net ~ age + sex, data=sample(race, 1000))
ggplot() + geom_density(aes(x=age), fill="red", alpha=.5, data=s100) +
    geom_density(aes(x=age), fill="blue", alpha=.5, data=s1000)
```



The standard error and sample size (con't)

The formula for the standard error is proportional to $1/\sqrt{n}$. This is kind of a slow decrease: "to make the standard error 10 times smaller you need to make the dataset 100 times larger"!

And now, back to our true sample

In reality, we don't have the luxury of measuring the entire population!

- We can use information in the original sample to make a good guess at what the sampling distribution is (see Kaplan Ch 5.2).
- The guess is based on an approximation that has good properties when the assumptions of our model aren't broken.

```
fm <- lm(net ~ age + sex, data=race)
summary(fm)$coef
```

```
##                Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 5339.15545 35.0486629 152.33550  0.000000e+00
## age           16.89362  0.9443776  17.88863  2.660668e-70
## sexM        -726.61948 20.0181263 -36.29808 1.281442e-268
```

Confidence Interval

A confidence interval summarizes our uncertainty about a point estimate.

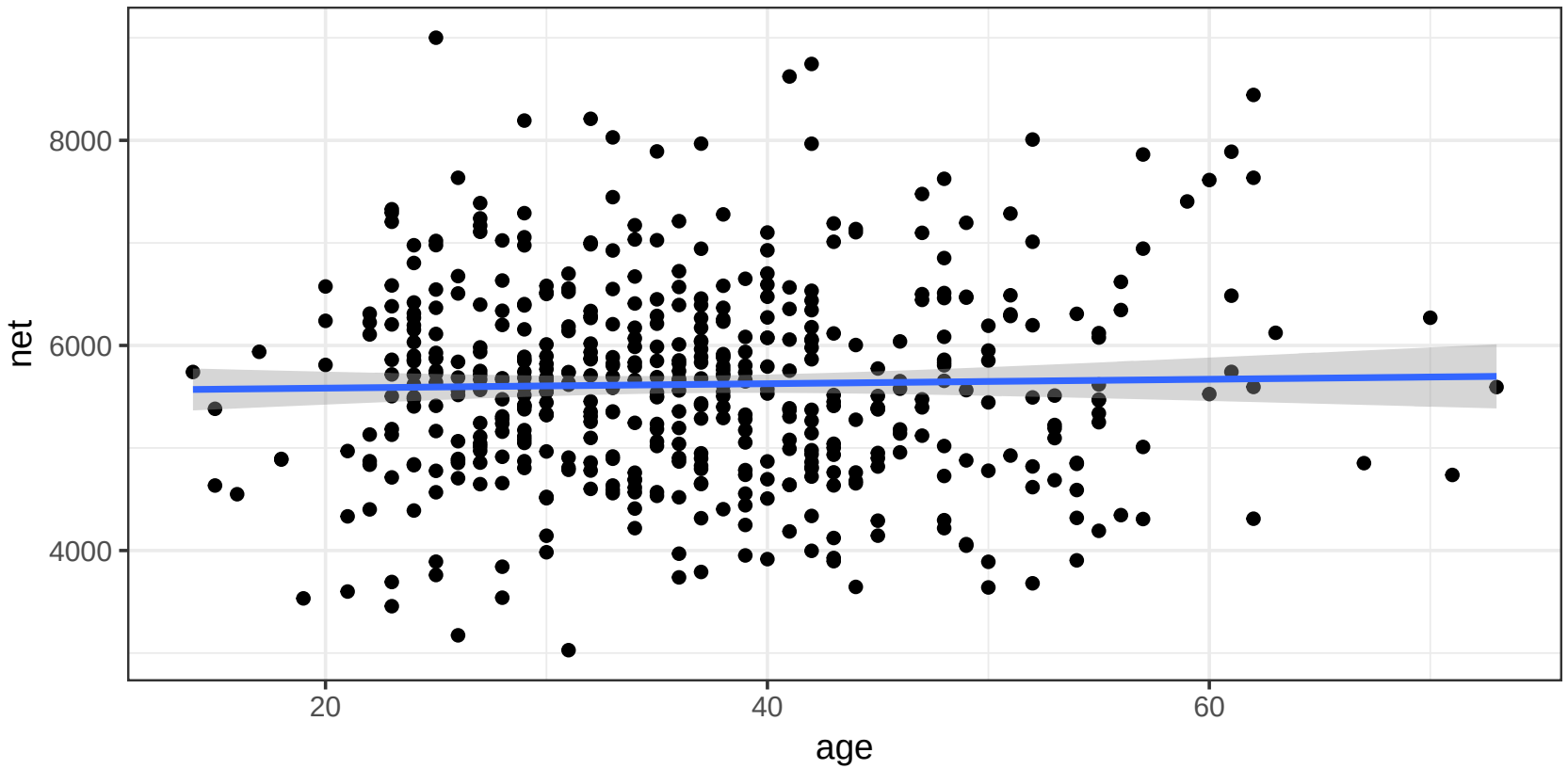
- For example: "our analysis suggests that the age coefficient in the model is 17 $\pm$ 2, with 95% confidence."
- More precisely, we could do the calculation as: 16.9 $\pm$ 2*0.94.
- We multiply the standard error by two because this approximates a 95% coverage interval of the sampling distribution.
- NOTE: when your sample size is very small (e.g. $n < 20$) the multiplier of 2 is misleading, and larger values should be used. See, e.g. Table 12.1 in Kaplan.

```
summary(fm)$coef["age",]

##     Estimate   Std. Error      t value     Pr(>|t|)
## 1.689362e+01 9.443776e-01 1.788863e+01 2.660668e-70
```

Confidence in predictions

Confidence intervals are not appropriate for making predictions about individual data-points!

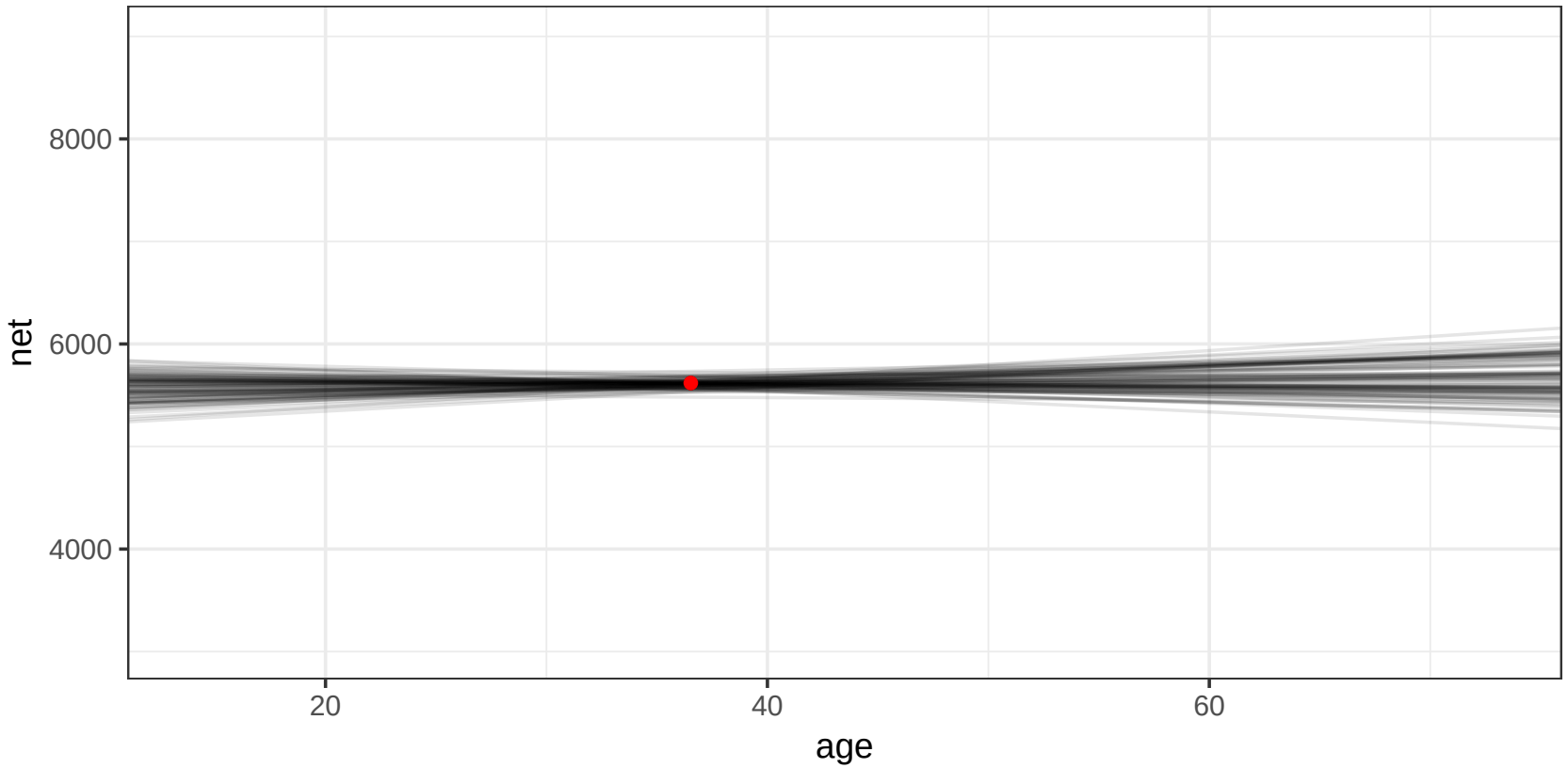
```
r <- sample(race, 500)
qplot(age, net, data=r) + geom_smooth(method="lm")
```



E.g. 95% of 60-year-olds will not have times within $\pm$ 200 of the predicted value ($\sim$ 5900).

Confidence in predictions

Confidence intervals for regression coefficients represent the uncertainty in the coefficient, but not in the predictions at certain, fixed values. Recall that the line has to pass through the point $(\bar{x}, \bar{y})$. Small changes in slope/intercept will have minimal changes to where the line passes near that fulcrum, and larger changes at the fringes.



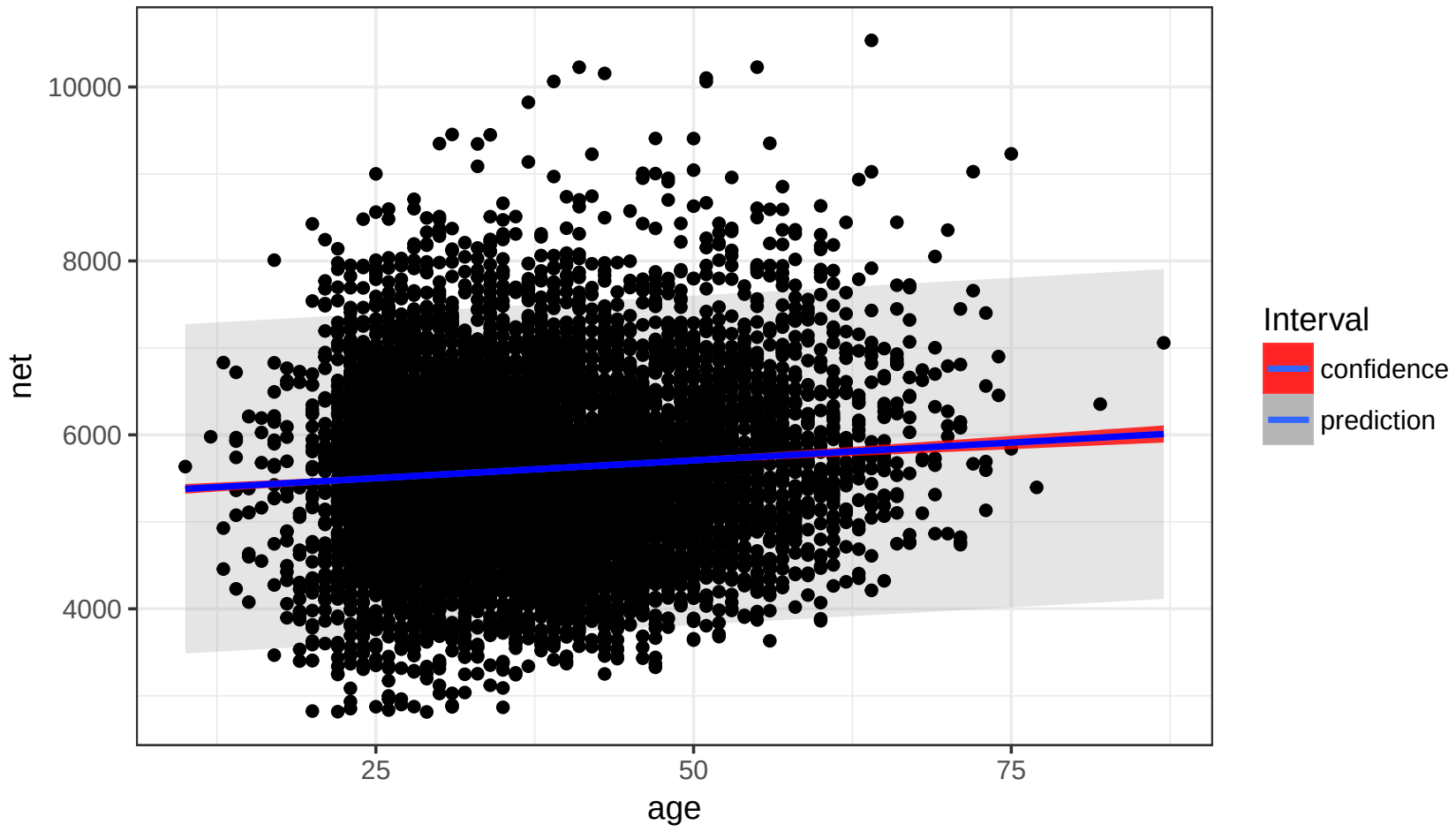
Making predictions

```
head(predict(fm, interval="confidence"))

##        fit      lwr      upr
## 1 4815.259 4757.616 4872.903
## 2 4832.153 4776.141 4888.165
## 3 4832.153 4776.141 4888.165
## 4 4849.047 4794.652 4903.442
## 5 4849.047 4794.652 4903.442
## 6 4849.047 4794.652 4903.442

head(predict(fm, interval="prediction"))

##        fit      lwr      upr
## 1 4815.259 3050.770 6579.749
## 2 4832.153 3067.716 6596.590
## 3 4832.153 3067.716 6596.590
## 4 4849.047 3084.660 6613.433
## 5 4849.047 3084.660 6613.433
## 6 4849.047 3084.660 6613.433
```

Making predictions

```
predict(fm, newdata = data.frame(age=c(20, 50, 60),
                                 sex=c("M", "M", "M")),
        interval="prediction")

##        fit      lwr      upr
## 1 4950.408 3186.285 6714.532
## 2 5457.217 3693.359 7221.075
## 3 5626.153 3861.995 7390.312
```

Prediction vs. confidence interval, race data



Today's key topics

- Sampling distributions
- Standard error
- Confidence errors and intervals for coefficients
- Prediction intervals for future observations