Quantile Regression

DS-6030 | Spring 2026

quantile-regression.pdf

Table of contents

1 From Means to Quantiles

1.1 Motivating Example

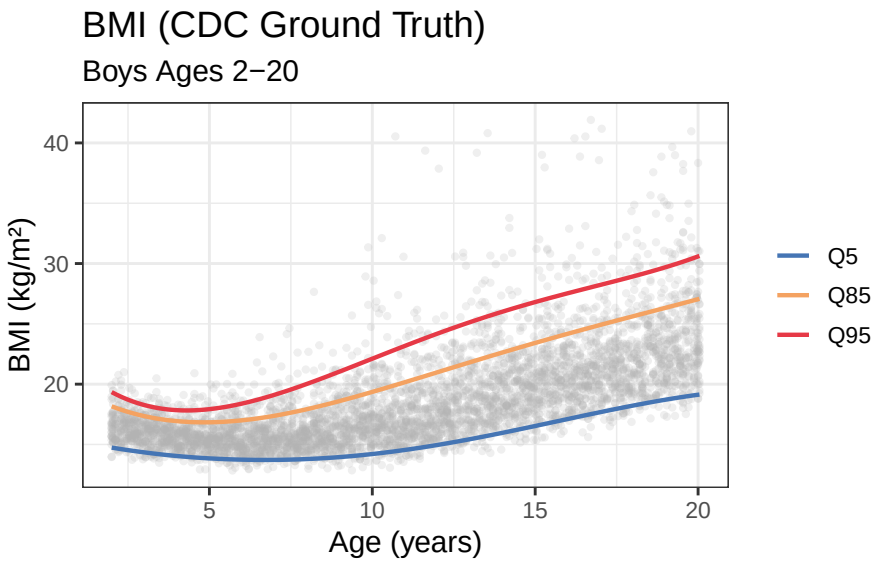
- When you fit a regression model and predict $\hat{y}$ for a new observation, what exactly are you predicting?
- The standard answer is: the **conditional mean** $E[Y \mid X = x]$. That is, the *average* outcome across all individuals who share the same covariate values x.
- This is useful, but often not what we actually want to know. Consider a pediatrician assessing a 10-year-old boy with a BMI of 22. The *mean/avg* BMI for a 10-year-old boy is around 17.
  - 22 is above average, but is it a problem?
- The CDC doesn't define childhood obesity or nourishment using the mean. It uses **percentile thresholds**:

Category	Definition
Underweight	Below the 5th percentile
Healthy weight	5th to 85th percentile
Overweight	85th to 95th percentile
Obese	95th percentile and above

- To use these thresholds, the pediatrician doesn't need to know the *mean BMI* at age 10. They need to know the **5th, 85th, and 95th percentiles** of BMI at age 10.
  - *Quantiles* are percentiles written in decimal form: The 5th percentile is the 0.05 quantile, etc.
- The mean tells you where the **average** child is. The quantiles tells you where **a particular child** is positioned in the distribution. This is the information that actually drives clinical decisions.

1.2 Setup and Data

Consider a sample of BMI and age observations. These are for boys aged 2-20. The data has two variables: `age_years` and `bmi`.



Your Turn #1 : Observations

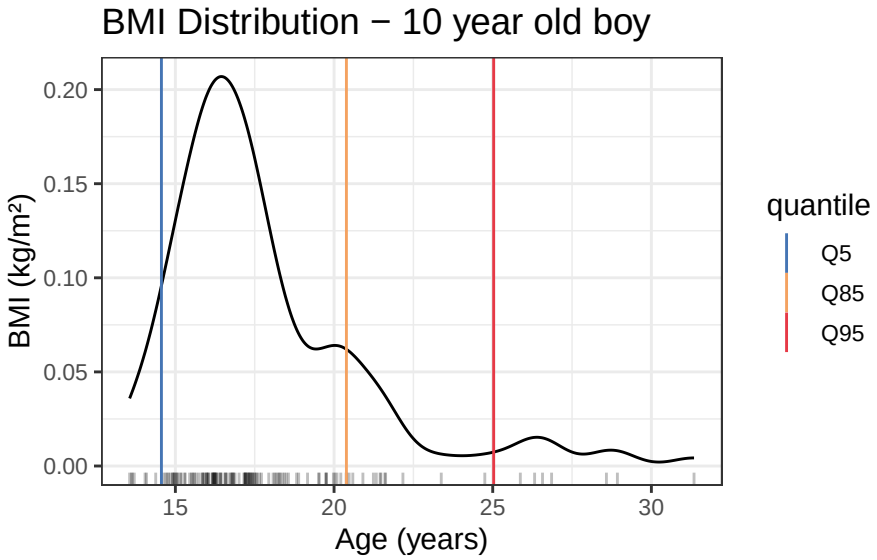
What patterns do you see?

2 Quantile Regression

2.1 What Is a Quantile?

The τ-th quantile of a distribution is the value q such that a fraction τ of the mass falls below q.

- $\tau = 0.5$: the **median** - half the distribution is below
- $\tau = 0.1$: the 10th percentile - 10% of the distribution is below
- $\tau = 0.9$: the 90th percentile - 90% of the distribution is below



quantile	value	num_less	num_greater
Q5	14.56	7	129
Q50	16.94	68	68
Q85	20.39	115	21
Q95	25.03	129	7

2.1.1 Quantile Regression

Quantile regression extends this to a regression setting. We build models to predict the *quantile* of Y **conditional on** $X = x$.

- That is, instead of modeling for the mean/avg $E[Y \mid X = x]$ like we do in regular regression, in quantile regression we model $Q_\tau(Y \mid X = x)$ which is the τ-th conditional quantile of Y given $X = x$.

The parameter $\tau \in (0, 1)$ is something *you choose*. Running a quantile regression at $\tau = 0.1$ gives us a curve such that, at each value of x, approximately 10% of observations fall below the curve.

2.2 The Pinball Loss

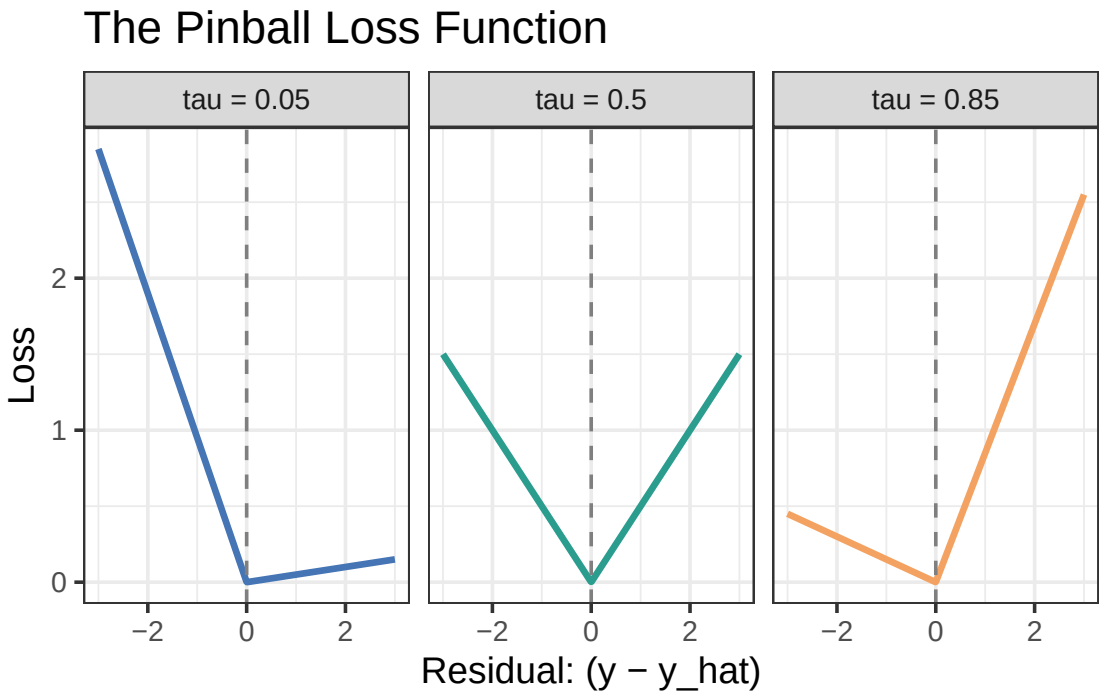
How do we estimate a conditional quantile? We need a loss function. Recall that OLS regression minimizes the **squared error** loss, which targets the mean:

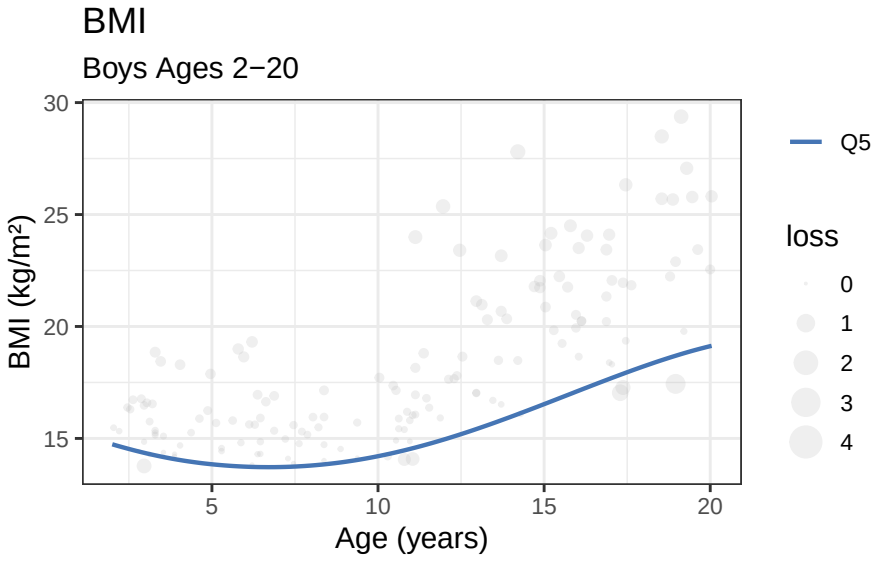
$$\hat{\mu}(x) = \arg\min_f \; E\left[(Y - f(X))^2\right]$$

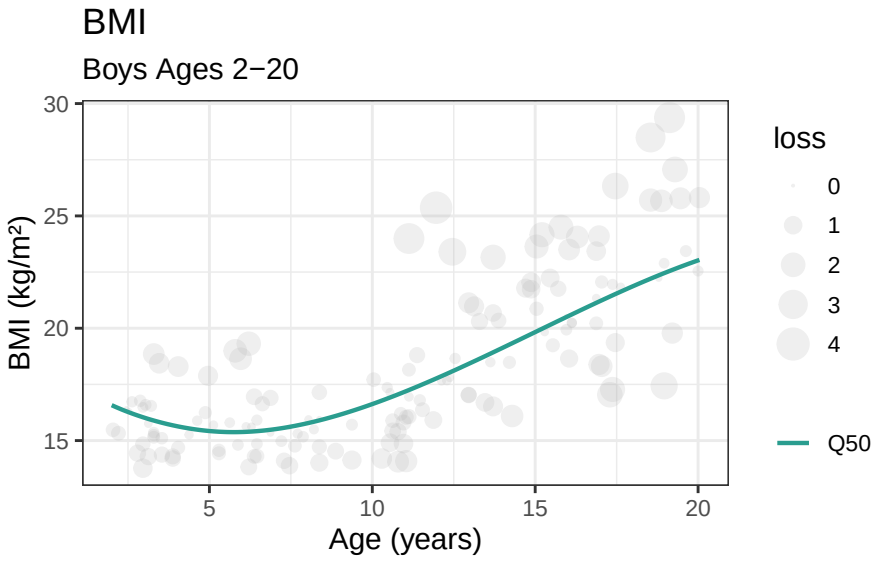
To target a quantile instead, we use the **pinball loss**:

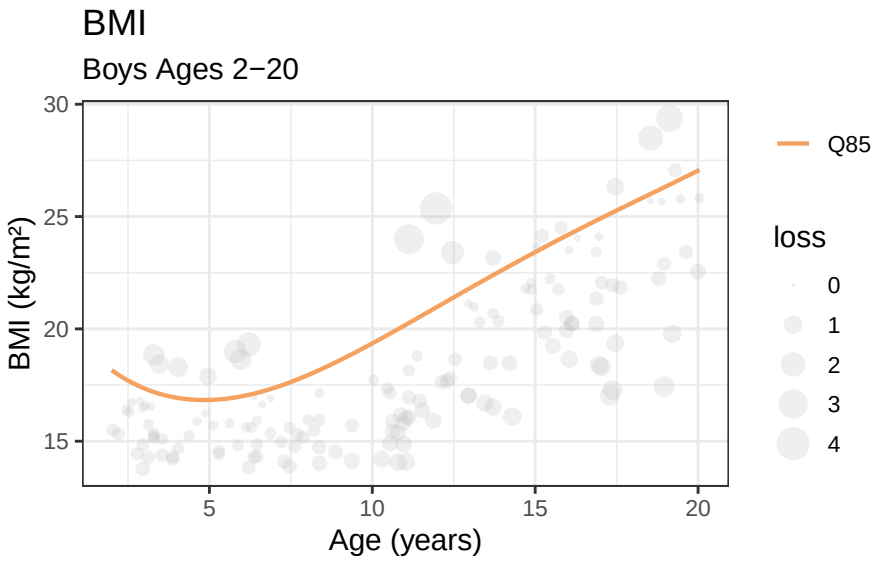
$$L_\tau(y, \hat{y}) = \begin{cases} \tau \, |y - \hat{y}| & \text{if } y \geq \hat{y} \\ (1 - \tau) \, |y - \hat{y}| & \text{if } y < \hat{y} \end{cases}$$

which is written here as a function of the *absolute prediction error* $|y - \hat{y}|$.









2.3 The Median as a Special Case

At $\tau = 0.5$, the pinball loss simplifies to:

$$L_{0.5}(y, \hat{y}) = \frac{1}{2}|y - \hat{y}|$$

which is proportional to the absolute error. So:

$$\hat{Q}_{0.5}(Y \mid X = x) = \arg\min_{f} \; E\left[|Y - f(X)|\right]$$

The **median regression** minimizes MAE. This is more robust to outliers than mean regression (which minimizes MSE), because absolute error grows linearly rather than quadratically with prediction errors.

3 Fitting Quantile Regression Models

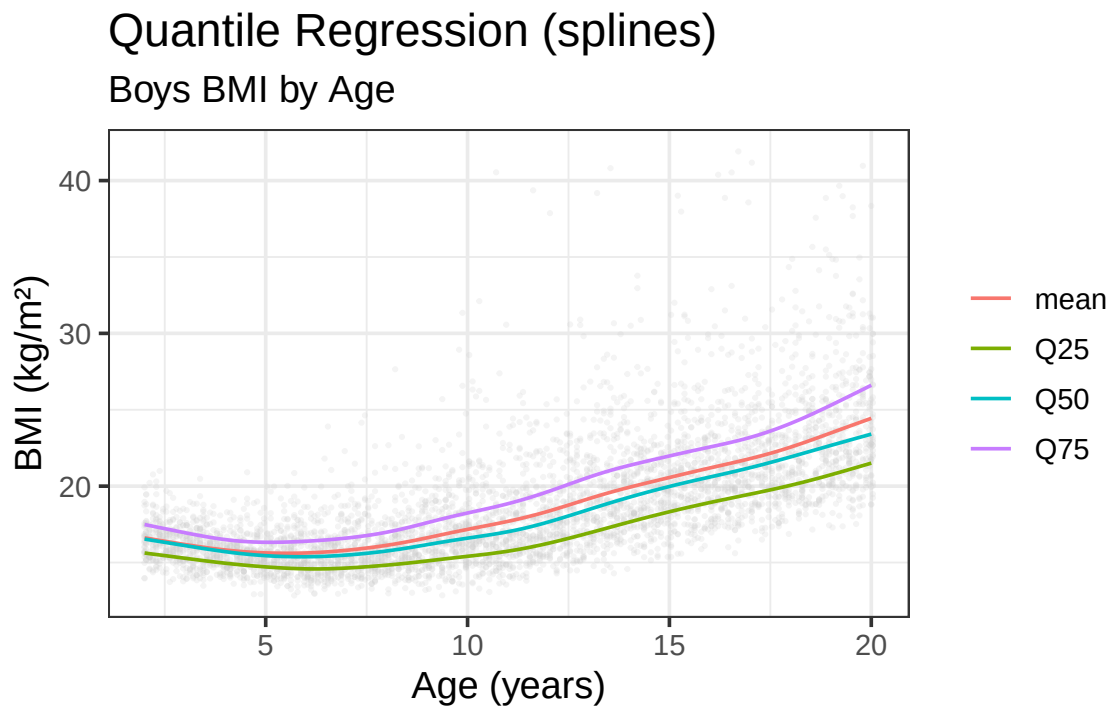
3.1 Splines

In R, the `qgam` package fits smooth quantile curves using the notation of `mgcv::gam()`. For example,

```r
library(qgam)

fit_qgam = qgam::mqgam(
  bmi ~ s(age_years, k = 20),   # smooth function of age_years
  data = growth_bmi,            # observed data
  qu   = c(0.25, 0.50, 0.75)    # vector of quantiles
)
```

In python, it takes a bit more effort but still doable:

```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import SplineTransformer
from sklearn.linear_model import QuantileRegressor
from sklearn.pipeline import Pipeline

X = growth_bmi[["age_years"]]
y = growth_bmi.bmi

quantiles = [0.25, 0.50, 0.75]

fits = {
    tau: Pipeline([
        ("spline", SplineTransformer(n_knots=20, degree=3)),
        ("qr",     QuantileRegressor(quantile=tau, alpha=0))
    ]).fit(X, y)
    for tau in quantiles
}
```

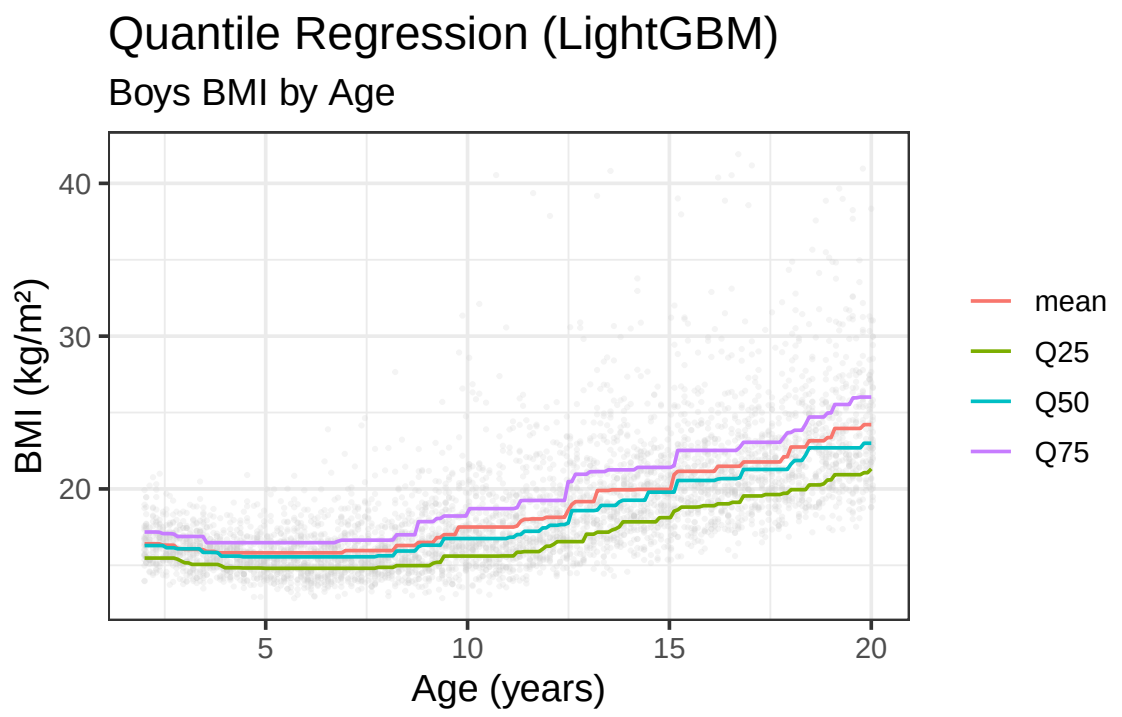


3.2 Boosted Trees with Pinball Loss

Boosted trees are a great option too. Since we only have one predictor, `age_years` I'll use stumps. This cuts down on the requirement to tuning tree depth related hyperparameters. We should be tuning either `trees` or `learn_rate` *for each* τ, but for illustration here, I just set it to something that produced decent visual pattern.

```
library(tidymodels)
library(bonsai)

# Helper function: fit one quantile model at tau
fit_quantile <- function(tau, data) {
  boost_tree() |>
    set_mode("regression") |>
    set_engine(
      "lightgbm",
      objective = "quantile",
      alpha     = tau,
      verbose   = -1
    ) |>
    set_args(
      tree_depth = 1,         # stumps
      trees      = 1000,
      learn_rate = 0.01
      # leaving other tuning parameters at default values
    ) |>
    fit(
      bmi ~ age_years,
      data = data
    )
}

#: fit model for each tau
taus = c(0.25, 0.50, 0.75)
fits = map(taus, \(tau) fit_quantile(tau, data = growth_bmi)) |>
  set_names(paste0("Q", taus * 100))

# Mean model as baseline: standard MSE objective
fit_mean =
   boost_tree() |>
    set_mode("regression") |>
    set_engine(
      "lightgbm",
      objective = "regression", # MSE
      verbose   = -1
    ) |>
    set_args(
      tree_depth = 1,         # stumps
      trees      = 1000,
      learn_rate = 0.01
      # leaving other tuning parameters at default values
    ) |>
    fit(
      bmi ~ age_years,
      data = growth_bmi
    )

# Predict on age grid
pred_df_bt = grid_df |>
  mutate(
    Q25  = predict(fits$Q25,  grid_df)$.pred,
```

```
    Q50  = predict(fits$Q50,  grid_df)$.pred,
    Q75  = predict(fits$Q75,  grid_df)$.pred,
    mean = predict(fit_mean,  grid_df)$.pred
)
```



3.3 Quantile Random Forests

Another useful option for predicting quantiles is to use **quantile regression forests** (Meinshausen, 2006).

The basic approach is actually very simple (and doesn't use the pinball loss).

Training

1. Fit a random forest using the usual MSE based splitting
2. Keep all OOB observations and their associated leaf node

Prediction

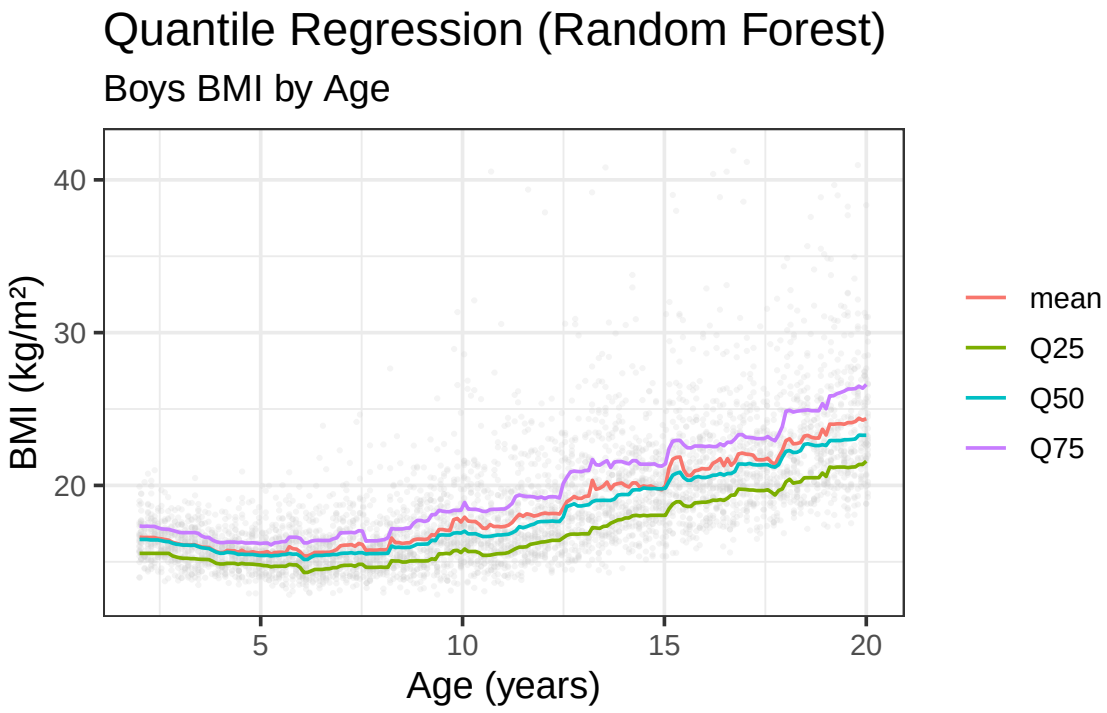
1. Put the new X down each tree.
2. Collect all the training observations that share that terminal node across all trees. These are the *neighbors* of X, weighted by how often they co-occur with X
3. The full empirical distribution of those neighbors' y-values is the estimated conditional distribution $\hat{F}(y \mid X = x)$. Read off whatever quantile(s) you want from that empirical distribution.

Notes

- Unlike the other models, this approach only requires fitting one model but can provide any quantile.

- Because we are getting a distribution from each leaf node, we don't let the trees get too deep (unlike the usual setting where we use deep trees to minimize bias).

In R, `ranger` supports this natively via `quantreg = TRUE` at fit time and `type = "quantiles"` at predict time.



4 Confidence Intervals vs. Prediction Intervals

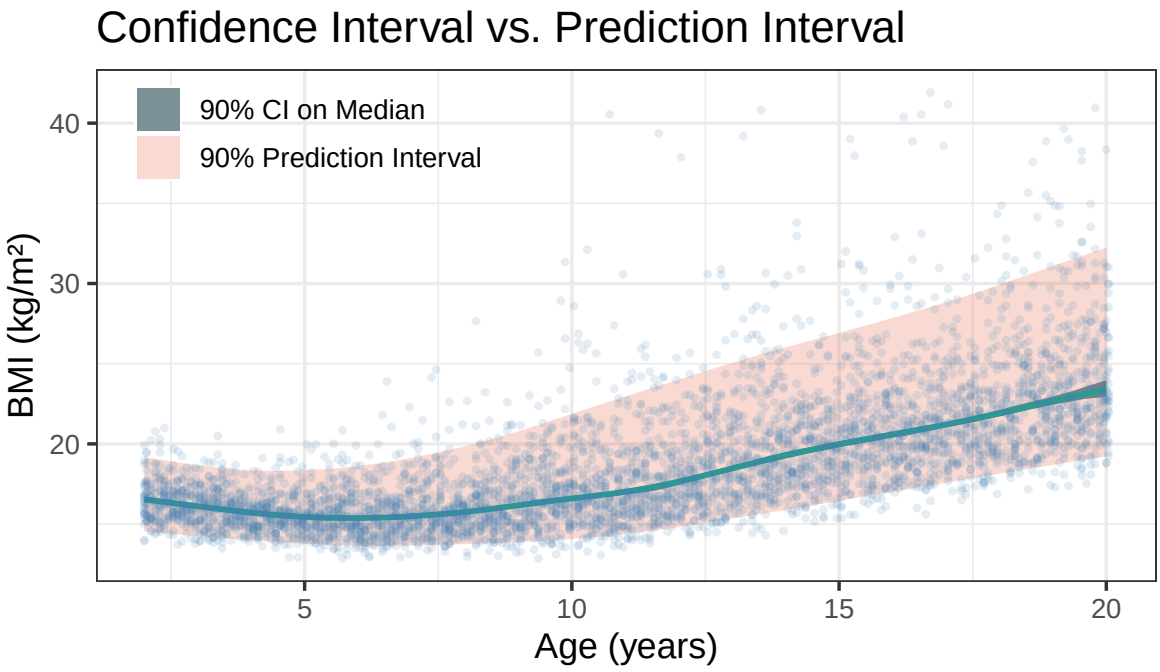
Confidence and Prediction intervals are interval estimates, but they answer **different questions**:

	Confidence Interval	Prediction Interval
Target	A population parameter (e.g., the mean curve)	A single new observation
Question	What is the true mean function?	Where will the next child's height fall?
Shrinks with more data?	Yes, proportional to $1/\sqrt{n}$	Can a little, but bounded by irreducible noise
Accounts for	Estimation uncertainty	Estimation uncertainty + data variability

A confidence interval on $E[Y \mid X = x]$ answers: *"If I collected many datasets and fit models to each one, what is the the mean curve?"* Confidence intervals gets narrower as sample size grows because we become more certain about the mean.

A prediction interval answers: *"If I bring in one new child aged 10, between what BMIs should I expect them to fall into?"* Prediction intervals **do not** shrink to zero no matter how much data we have, individual children still vary.

Sketch of Confidence Intervals and Prediction Intervals


Confidence Interval vs. Prediction Interval
90% CI on Median
90% Prediction Interval
BMI (kg/m²)
40
30
20
5
10
15
20
Age (years)

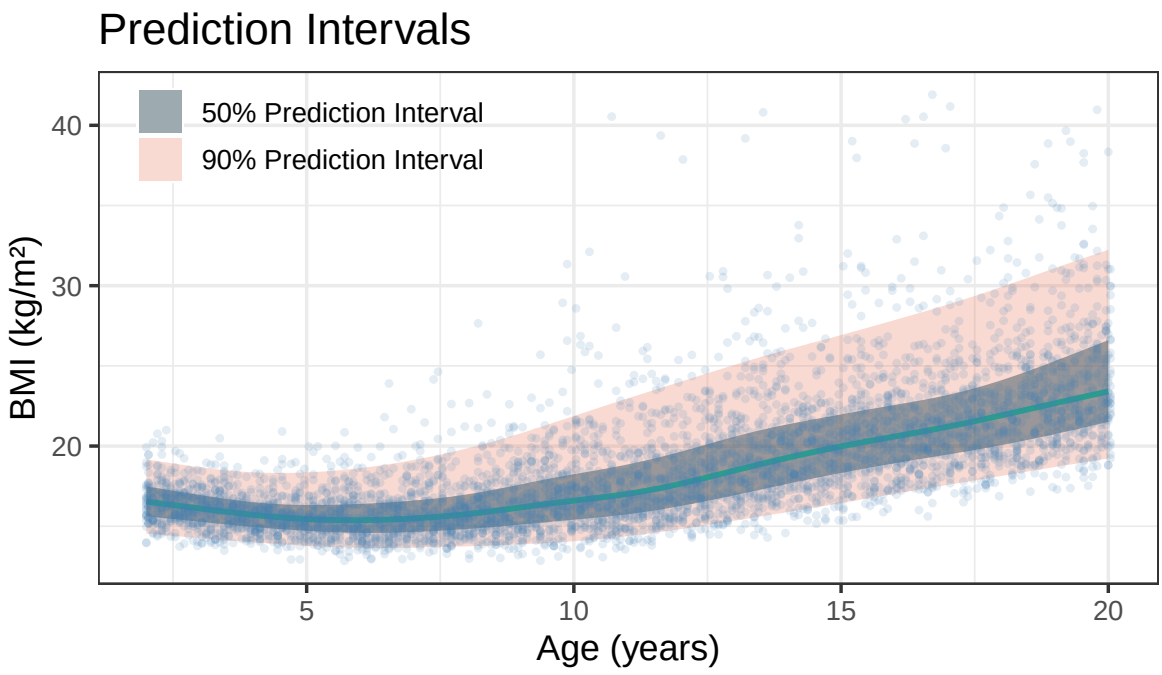
5 Quantile-Based Prediction Intervals

To build a prediction interval with nominal coverage $1 - \alpha$, we fit the model at two quantiles:

$$\text{PI} = \left[\hat{Q}_{\alpha/2}(Y \mid X = x),\ \hat{Q}_{1-\alpha/2}(Y \mid X = x)\right]$$

- For a 50% PI: fit at $\tau = 0.25$ and $\tau = 0.75$
- For a 90% PI: fit at $\tau = 0.05$ and $\tau = 0.95$
- For a 95% PI: fit at $\tau = 0.025$ and $\tau = 0.975$

By construction, approximately $1 - \alpha$ of observations should fall inside the interval.



5.1 Checking Empirical Coverage

We want to verify that approximately 90% of observations fall inside the interval. We check this by predicting at each observed age **in hold-out/test data**.

Predict the quantiles on the test data. Here is a sample:

age_years	bmi	q05	q25	q50	q75	q95
15.542	16.68	16.75	18.66	20.32	22.30	27.42
4.708	16.11	13.84	14.77	15.50	16.34	18.33
2.542	18.89	14.39	15.44	16.30	17.19	18.89
7.458	15.69	13.70	14.72	15.59	16.75	19.44
8.542	16.18	13.81	14.97	15.97	17.30	20.34
7.458	16.49	13.70	14.72	15.59	16.75	19.44

Then check to see what proportion are in the desired prediction intervals. This value is called the *coverage* of the interval.

n	coverage_50	coverage_90
1000	0.552	0.92

---

6 Summary

The main ideas from today:

- **Quantile regression** estimates any conditional quantile $Q_\tau(Y \mid X = x)$ by minimizing the pinball loss
- The **pinball loss** is asymmetric: the slope on each side is controlled by τ, pushing the model toward the target quantile
- At $\tau = 0.5$, the pinball loss reduces to **MAE**, and the model estimates the **conditional median**. This is more robust to outliers than the mean (but again, everything depends on which loss function is right for the application).
- **Prediction intervals** are not the same as confidence intervals. PIs capture where individual observations will fall, not where the mean function is.
- Fitting at $\tau = \alpha/2$ and $\tau = 1 - \alpha/2$ gives a $(1 - \alpha)$ prediction interval that adapts to **heteroscedasticity** (the width varies with x)