Lecture 12: Advanced regression tools

Skidmore College, MA 251

Goals

- What happens when linear or logistic regression don't look right?
- Spline terms

Regression problems/assumptions

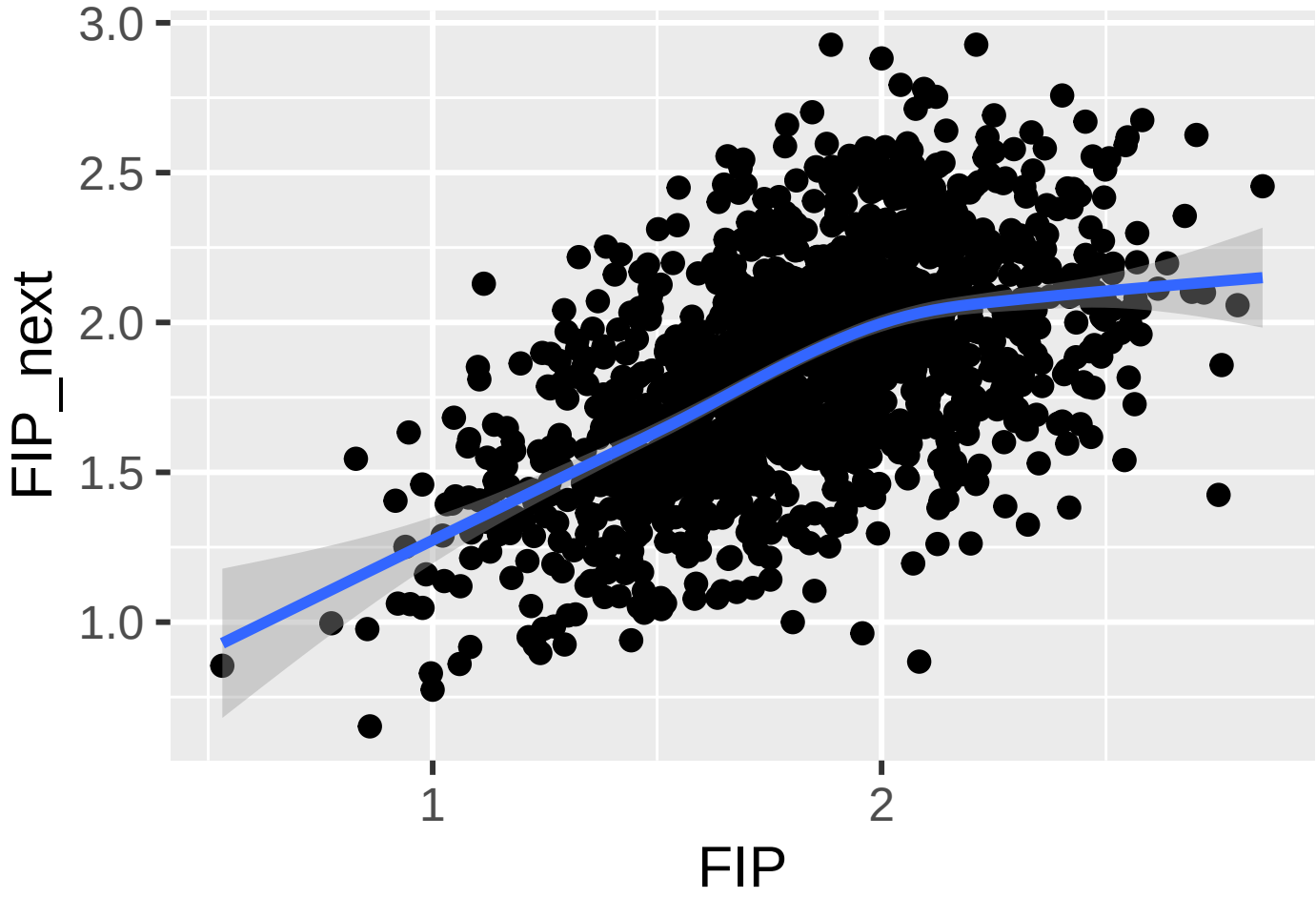
1. Residuals
2. Collinearity
3. Lack of fit
4. Which model is best?
5. Poor prediction

Ex:

```
library(tidyverse); library(Lahman); options(digits = 4)
Pitching <- Pitching %>%
  filter(yearID >= 2000, yearID <= 2014, BFP >= 500) %>%
  mutate(K_rate = SO/BFP,
         BB_rate = BB/BFP,
         HR_rate = HR/BFP,
         FIP = ((13*HR) + 5*(H - HR)  + 3*(BB + HBP) - 2*SO)/(IPouts)) %>%
  arrange(playerID, yearID) %>%
  group_by(playerID) %>%
  mutate(FIP_next = lead(FIP)) %>%
  filter(!is.na(FIP_next))
```

Baseball example

```
ggplot(Pitching, aes(FIP, FIP_next)) +
  geom_point() +
  geom_smooth()
```



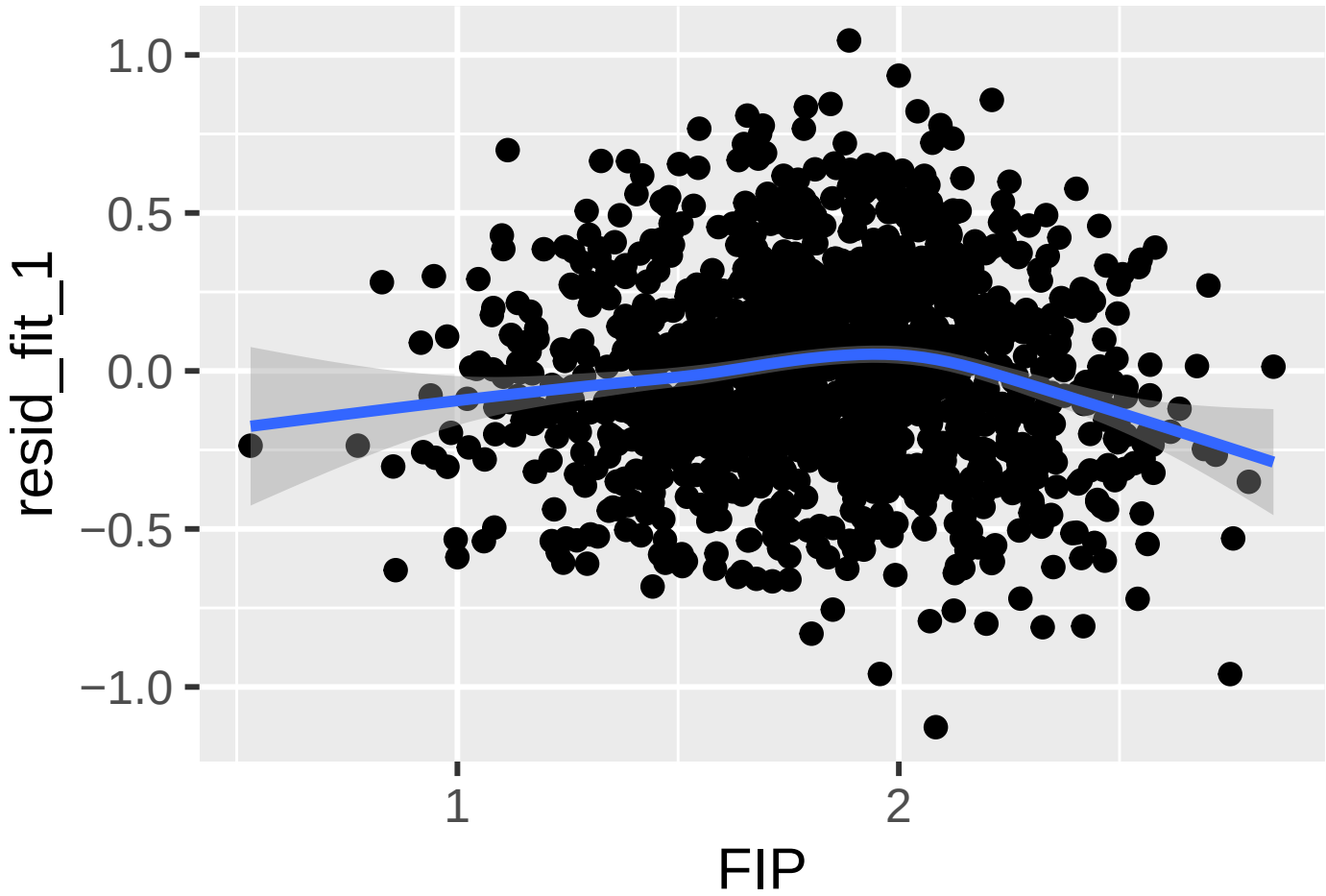
Baseball example

```
library(broom)
fit_pitcher_1 <- lm(FIP_next ~ FIP, data = Pitching)
tidy(fit_pitcher_1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    0.781    0.0459      17.0 8.23e- 59
## 2 FIP            0.583    0.0246      23.6 3.24e-103
```

Baseball example

```
Pitching$resid_fit_1 <- fit_pitcher_1$residuals
ggplot(Pitching, aes(FIP, resid_fit_1)) +
  geom_point() +
  geom_smooth()
```



Baseball example

```
Pitching <- Pitching %>%
  mutate(FIP_sq = FIP^2)


fit_pitcher_2 <- lm(FIP_next ~ FIP + FIP_sq, data = Pitching)
tidy(fit_pitcher_2)
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -0.138    0.167     -0.828 4.08e- 1
## 2 FIP            1.64     0.186      8.80  4.38e-18
## 3 FIP_sq        -0.292    0.0510    -5.72  1.33e- 8
```

```
AIC(fit_pitcher_1)
```

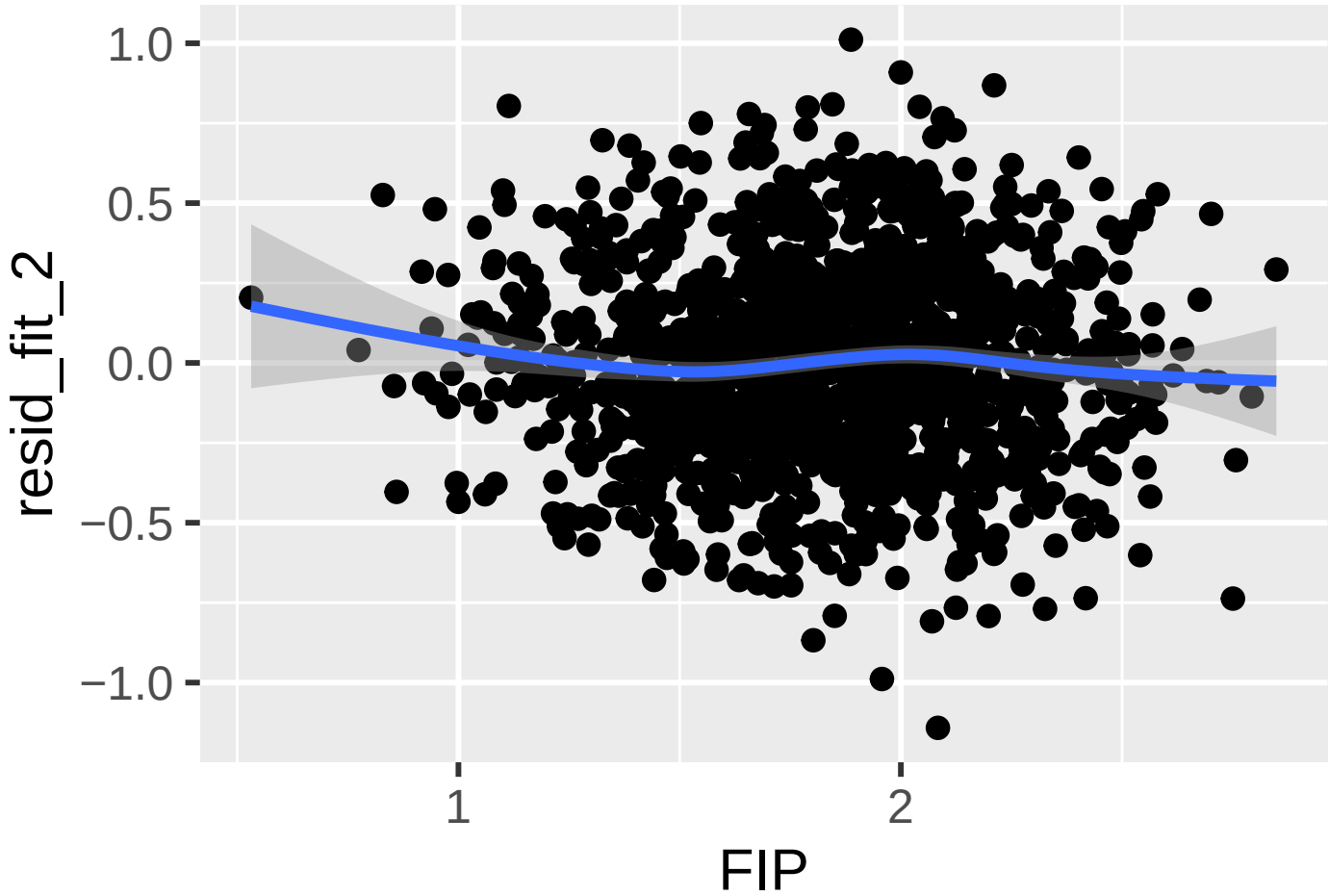
```
## [1] 680.8
```

```
AIC(fit_pitcher_2)
```

```
## [1] 650.5
```

Baseball example

```
Pitching$resid_fit_2 <- fit_pitcher_2$residuals
ggplot(Pitching, aes(FIP, resid_fit_2)) +
  geom_point() +
  geom_smooth()
```



Spline terms

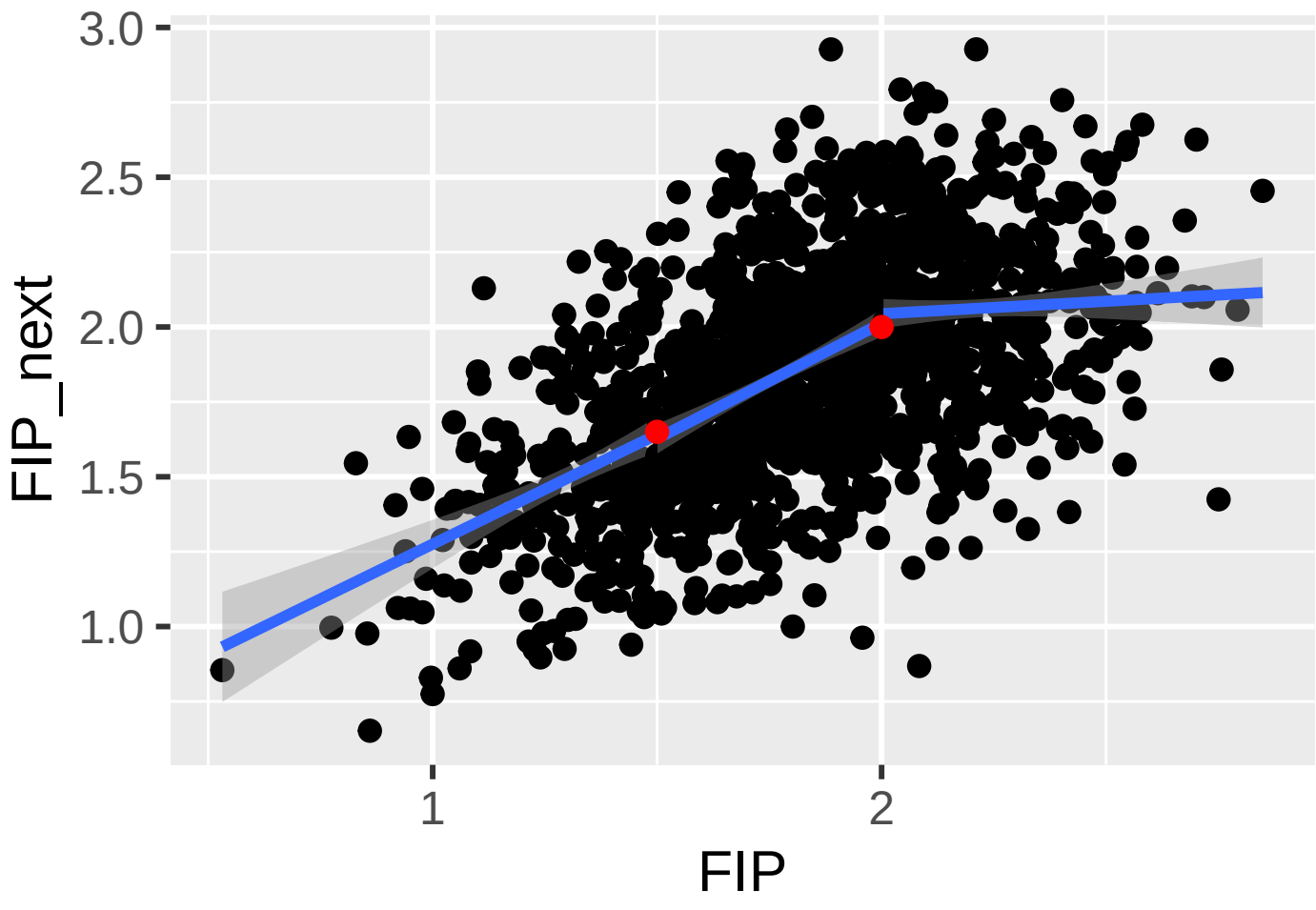
Linear spline: A linear spline is a continuous function formed by connecting linear segments. The points where the segments connect are called the knots of the spline.

Ex:

```
p_linear_spline <- ggplot() +
  geom_point(data = filter(Pitching, FIP < 1.5), aes(FIP, FIP_next)) +
  geom_smooth(data = filter(Pitching, FIP < 1.5), aes(FIP, FIP_next),
              method = "lm") +
  geom_point(data = filter(Pitching, FIP < 2, FIP > 1.5), aes(FIP, FIP_next)) +
  geom_smooth(data = filter(Pitching, FIP < 2, FIP > 1.5), aes(FIP, FIP_next),
              method = "lm") +
  geom_point(data = filter(Pitching, FIP > 2), aes(FIP, FIP_next)) +
  geom_smooth(data = filter(Pitching, FIP > 2), aes(FIP, FIP_next),
              method = "lm") +
  annotate("point", x = 1.5, y = 1.65, colour = "red") +
  annotate("point", x = 2, y = 2, colour = "red")
```

Spline terms

```
p_linear_spline
```



Spline terms

A spline of degree D is a function formed by connecting polynomial segments of degree D so that:

1. The function is continuous
2. The function has D ??? 1 continuous derivatives, and
3. The Dth derivative is constant between knots.

We'll analyze a slightly more complex version, called B-splines, defined by

- k number of knots
- degree of the polynomial in between the knots

In R, the `ns(X, k)` fits cubic splines for variable `X` with `k` knots

Spline terms

```
library(splines)
fit_pitcher_3 <- lm(FIP_next ~ ns(FIP, 4), data = Pitching)
tidy(fit_pitcher_3)
```

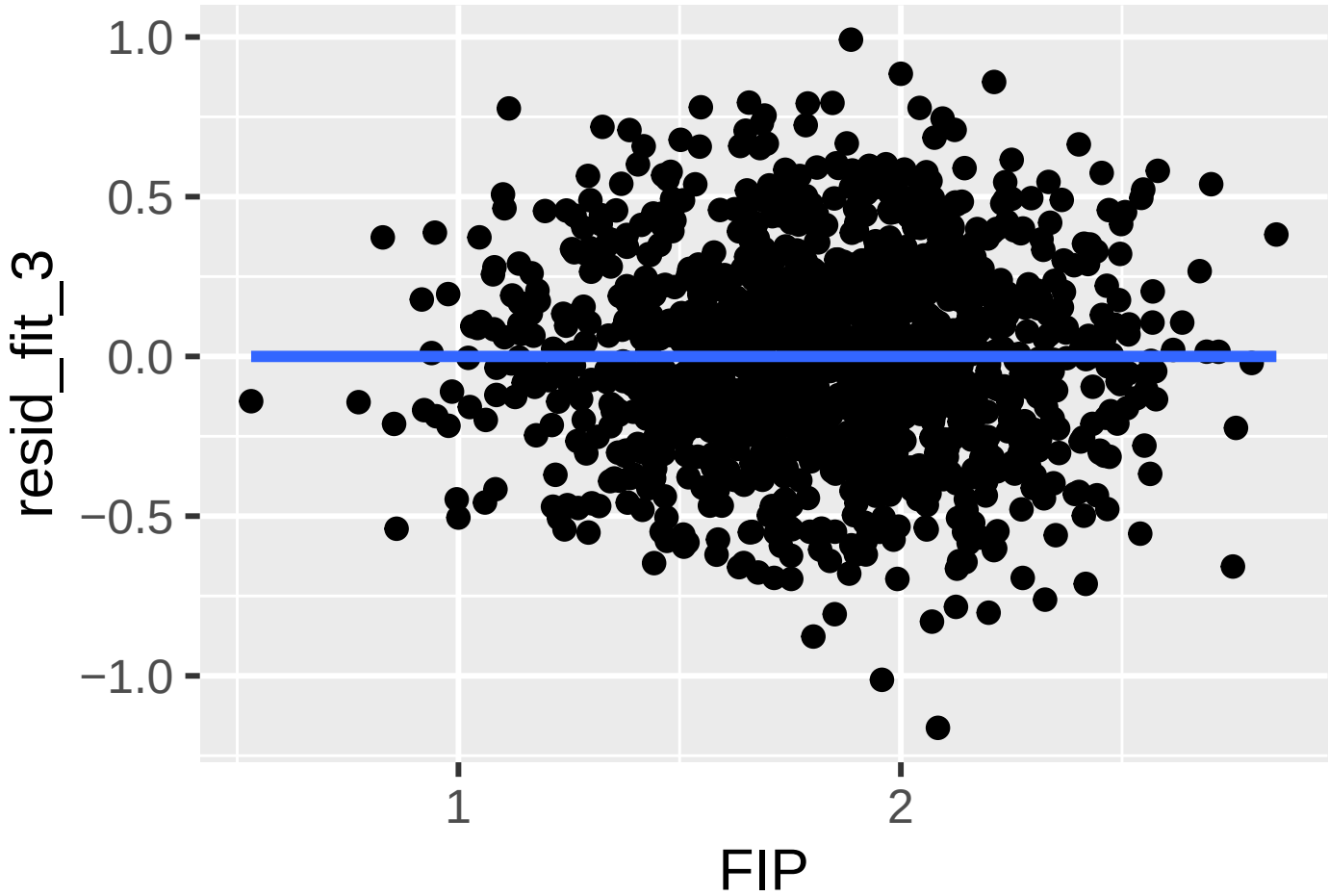
```
## # A tibble: 5 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    0.995    0.127       7.81 1.15e-14
## 2 ns(FIP, 4)1    0.920    0.120       7.65 3.92e-14
## 3 ns(FIP, 4)2    0.957    0.0835     11.5  4.96e-29
## 4 ns(FIP, 4)3    1.52     0.274       5.55 3.52e- 8
## 5 ns(FIP, 4)4    0.844    0.0988      8.53 3.86e-17
```

```
AIC(fit_pitcher_3)
```

```
## [1] 644.6
```

Spline terms

```
Pitching$resid_fit_3 <- fit_pitcher_3$residuals
ggplot(Pitching, aes(FIP, resid_fit_3)) +
  geom_point() +
  geom_smooth()
```



Alternative models

Different numbers of knots

```
fit_pitcher_3_i <- lm(FIP_next ~ ns(FIP, 2), data = Pitching)
fit_pitcher_3_ii <- lm(FIP_next ~ ns(FIP, 5), data = Pitching)
fit_pitcher_3_iii <- lm(FIP_next ~ ns(FIP, 10), data = Pitching)
AIC(fit_pitcher_3_i)
```

```
## [1] 649.7
```

```
AIC(fit_pitcher_3_ii)
```

```
## [1] 644.8
```

```
AIC(fit_pitcher_3_iii)
```

```
## [1] 647.2
```

Fitted terms

```
Pitching$FIP_next_hat_1 <- predict(fit_pitcher_1, Pitching) ## Linear model
Pitching$FIP_next_hat_2 <- predict(fit_pitcher_2, Pitching) ## Quadratic term
Pitching$FIP_next_hat_3 <- predict(fit_pitcher_3, Pitching) ## Spline term

Pitching %>% filter(playerID == "silvaca01", yearID == 2006) %>%
  select(playerID, yearID, teamID, FIP,
         FIP_next, FIP_next_hat_1, FIP_next_hat_2,
         FIP_next_hat_3)
```

```
## # A tibble: 1 x 8
## # Groups:   playerID [1]
##   playerID yearID teamID   FIP FIP_next FIP_next_hat_1 FIP_next_hat_2
##   <chr>     <int> <fct>  <dbl>    <dbl>          <dbl>          <dbl>
## 1 silvaca~   2006 MIN     2.79     2.06           2.41           2.16
## # ... with 1 more variable: FIP_next_hat_3 <dbl>
```

For a primer on splines, see
https://cran.r-project.org/web/packages/crs/vignettes/spline_primer.pdf

Additional thoughts

Why spline terms?

Why not spline terms?

What problems have we still not accounted for?