DATA 608 Meetup 9: Models and Uncertainty

George I. Hagstrom

Week Summary

- This week is about visualizing models and uncertainty
- Next week is about dimensionality reduction techniques
- Assignment: Make a model of house sale prices and tell a story about it
- Reading: Healy Visualization, Fundamentals of Data Visualization 16
- Bonus Readings: Marginal Effects Book: 2-3, 15

How to talk about models

- You have just fitted a model
- You have an array of model coefficients, accuracy measures, and summary statistics
- How do you convey insights from the model to a PM, supervisor, or even a colleague?

Model Coefficients Hard to Interpret

- Thornton dataset: How much does a monetary incentive impact the decision to get an HIV test?

```r
dat = get_dataset("thornton")
mod_thornton = glm(outcome ~ incentive * (agecat + distance),
    data = dat, family = binomial)
dat |> head()
```

```
# A data frame: 6 × 8
  village outcome distance amount incentive   age hiv2004 agecat
*   <int>   <int>    <dbl>  <dbl>     <int> <int>   <int> <fct>
1      43       1    0.549      0         0    14       0 <18
2     117       0    0.840      0         0    14       0 <18
3       2       0    3.34       0         0    15       0 <18
4       6       0    2.32       0         0    15       0 <18
5      11       0    1.39       0         0    15       0 <18
6      14       0    3.87       0         0    15       0 <18
```

Model Coefficients Hard to Interpret

- What do these numbers mean?
- Are log-odds-ratios intuitive to you?

```
            (Intercept)                incentive            agecat18 to 35
            -0.48616576               2.05760093                0.07937246
              agecat>35                 distance incentive:agecat18 to 35
             0.34522236              -0.18439675               -0.05850445
  incentive:agecat>35       incentive:distance
            -0.12468320               0.02304394
```

- Leads to poor decisions (like ignoring everything but sign)

Model Coefficients Hard to Interpret

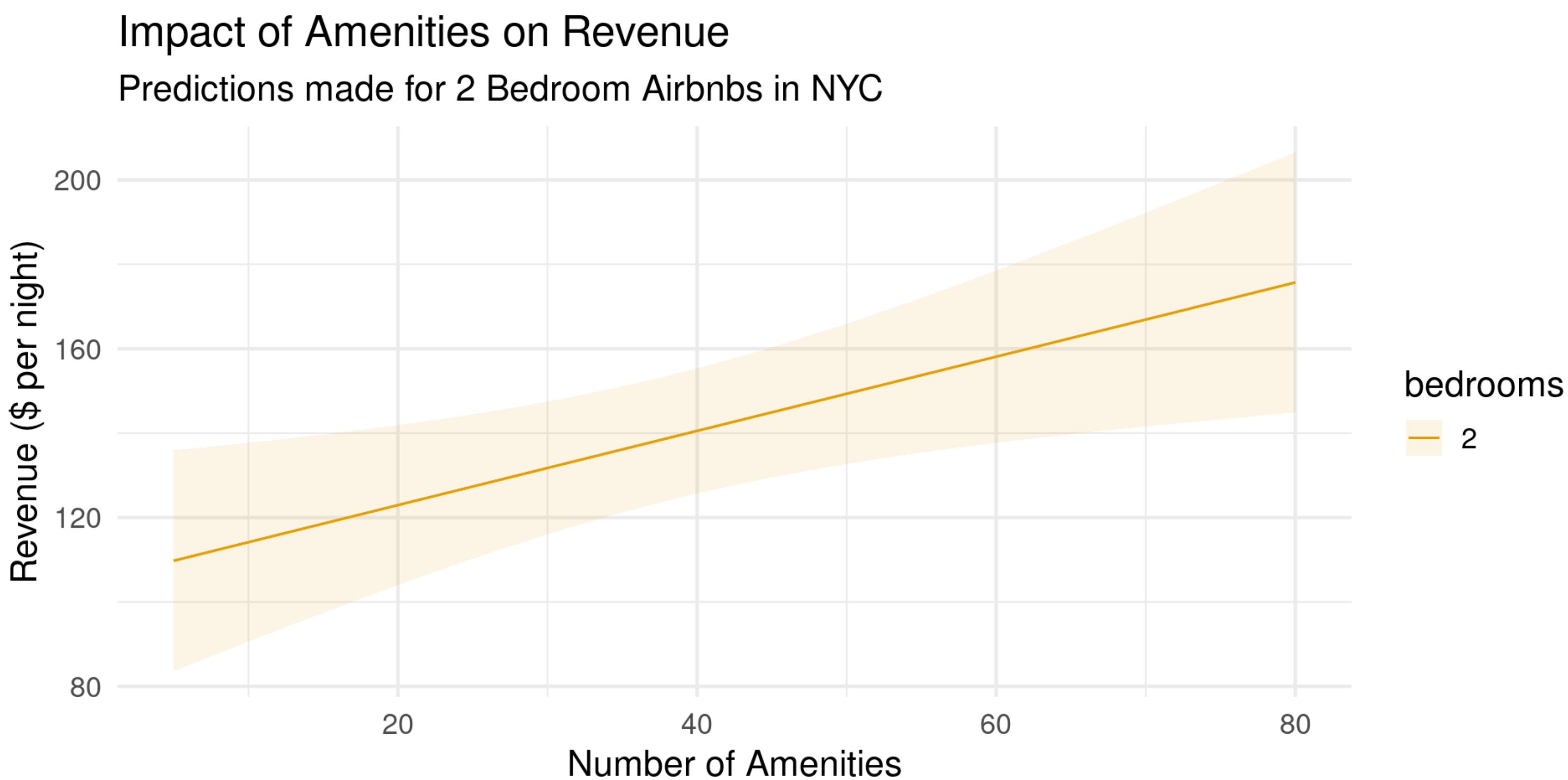
- This isn’t unique to logistic regression
- For any model but simple linear regression
- Need a better way to present model results

Alternative Workflow

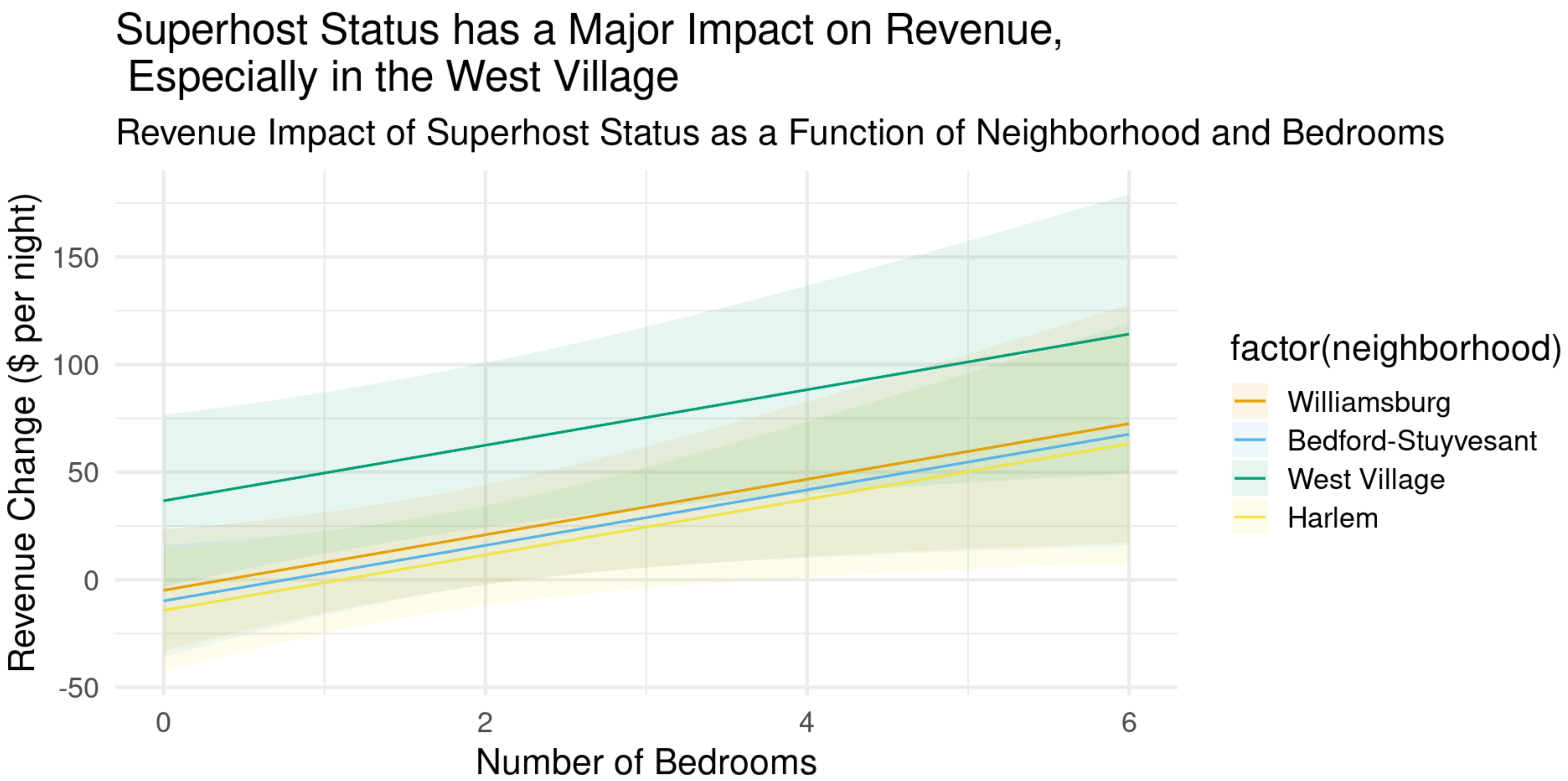
- Don’t focus on parameter estimates
- Instead transform them to quantities that shed light on your analytic questions
- Predictions, Comparisons, Slopes
- Always Show Uncertainty

Predictions

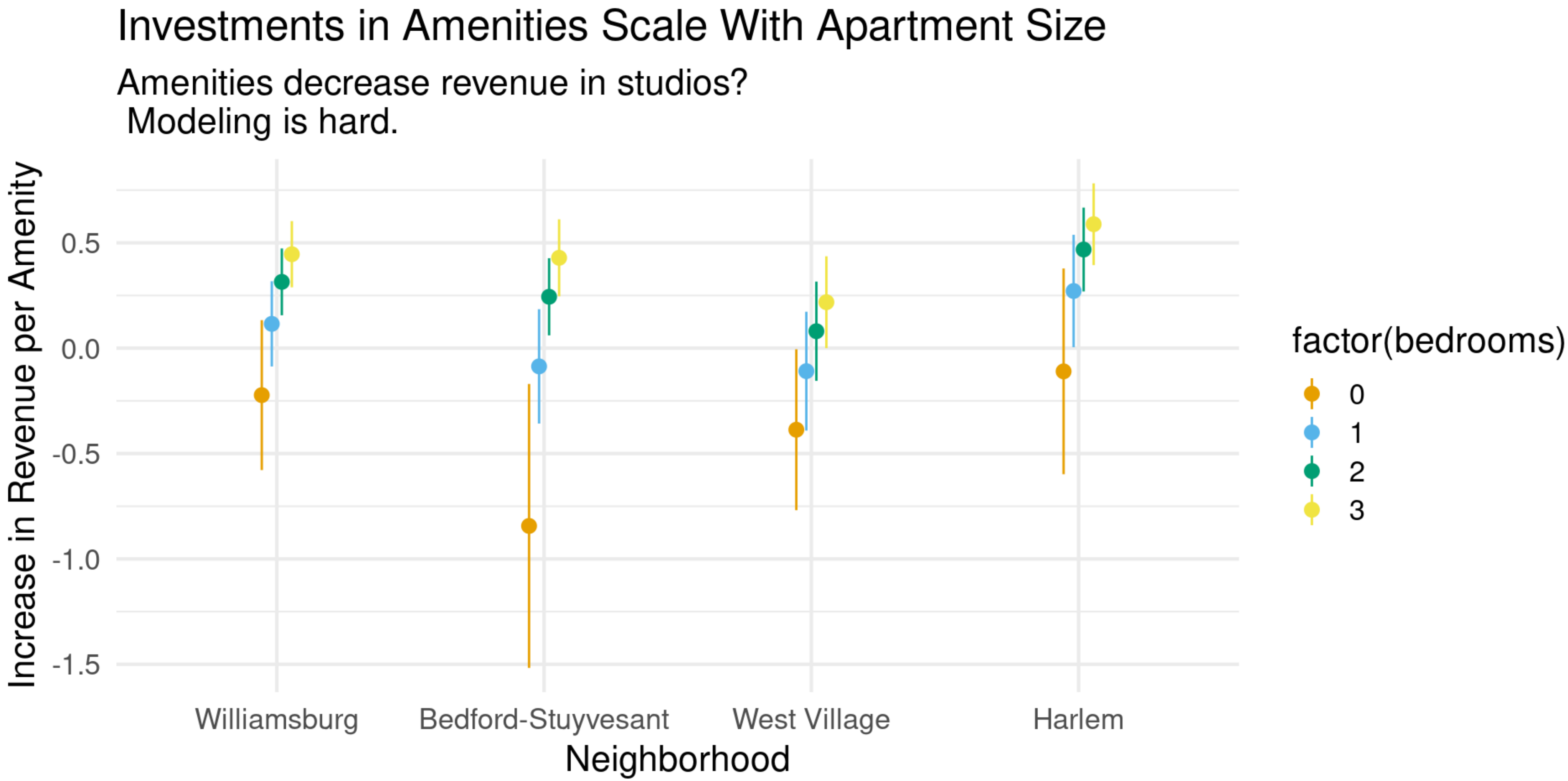
- A prediction shows what your model says would happen for units with hypothetical predictor values



Counterfactual Comparisons



Slopes



Investments in Amenities Scale With Apartment Size
Amenities decrease revenue in studios?
Modeling is hard.
Increase in Revenue per Amenity
0.5
0.0
-0.5
-1.0
-1.5
Williamsburg
Bedford-Stuyvesant
West Village
Harlem
Neighborhood
factor(bedrooms)
0
1
2
3

Look inside a model object

```r
ames = read_csv("/home/georgehagstrom/work/Teaching/DATA608/DataStory5/trai
mod = lm(SalePrice ~ YearBuilt + BedroomAbvGr + `1stFlrSF` +
             `2ndFlrSF` + GarageArea + Neighborhood,ames)
mod
```

```
Call:
lm(formula = SalePrice ~ YearBuilt + BedroomAbvGr + `1stFlrSF` + 
    `2ndFlrSF` + GarageArea + Neighborhood, data = ames)

Coefficients:
        (Intercept)            YearBuilt         BedroomAbvGr  
         -906678.44               477.60             -7057.40  
         `1stFlrSF`           `2ndFlrSF`           GarageArea  
              94.96                69.22                42.01  
NeighborhoodBlueste   NeighborhoodBrDale  NeighborhoodBrkSide  
          -15845.51            -18830.59              8491.97  
NeighborhoodClearCr  NeighborhoodCollgCr  NeighborhoodCrawfor  
           24509.12             14684.76             37778.60  
NeighborhoodEdwards  NeighborhoodGilbert   NeighborhoodIDOTRR  
```

summary for top level info

```r
mod |> summary()
```

```
Call:
lm(formula = SalePrice ~ YearBuilt + BedroomAbvGr + `1stFlrSF` +
    `2ndFlrSF` + GarageArea + Neighborhood, data = ames)

Residuals:
    Min      1Q  Median      3Q     Max
-428522  -16684     328   15778  269785

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -9.067e+05  1.440e+05  -6.297 4.03e-10 ***
YearBuilt            4.776e+02  7.191e+01   6.642 4.39e-11 ***
BedroomAbvGr        -7.057e+03  1.599e+03  -4.412 1.10e-05 ***
`1stFlrSF`           9.496e+01  3.591e+00  26.443  < 2e-16 ***
```

`str` for all details

```r
mod |> str()
```

```
List of 13
 $ coefficients : Named num [1:30] -906678.4 477.6 -7057.4 95 69.2 ...
  ..- attr(*, "names")= chr [1:30] "(Intercept)" "YearBuilt" "BedroomAbvGr"
"`1stFlrSF`" ...
 $ residuals    : Named num [1:1460] 1614 -21747 8140 -55088 -58200 ...
  ..- attr(*, "names")= chr [1:1460] "1" "2" "3" "4" ...
 $ effects      : Named num [1:1460] -6912989 -1586708 -624098 -1370096
-1285383 ...
  ..- attr(*, "names")= chr [1:1460] "(Intercept)" "YearBuilt" "BedroomAbvGr"
"`1stFlrSF`" ...
 $ rank         : int 30
 $ fitted.values: Named num [1:1460] 206886 203247 215360 195088 308200 ...
  ..- attr(*, "names")= chr [1:1460] "1" "2" "3" "4" ...
 $ assign       : int [1:30] 0 1 2 3 4 5 6 6 6 6 ...
 $ qr           :List of 5
```

Tidying Model Objects

- `broom` is part of `tidymodels` family and contains function which tidy the model for you
- `tidy()` creates tibble of model components:

```
mod |> tidy(conf.int = TRUE)
```

Model Table from Tidy

```
# A tibble: 30 × 7
   term                 estimate std.error statistic   p.value conf.low conf.high
   <chr>                   <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
 1 (Intercept)           -9.07e5 143987.      -6.30   4.03e- 10 -1.19e6 -624229.
 2 YearBuilt              4.78e2     71.9      6.64   4.39e- 11  3.37e2     619.
 3 BedroomAbvGr          -7.06e3   1599.      -4.41   1.10e-  5 -1.02e4   -3920.
 4 `1stFlrSF`             9.50e1      3.59    26.4    8.95e-126  8.79e1     102.
 5 `2ndFlrSF`             6.92e1      3.31    20.9    5.41e- 85  6.27e1      75.7
 6 GarageArea             4.20e1      6.35     6.61   5.29e- 11  2.96e1      54.5
 7 NeighborhoodBlueste   -1.58e4  28337.      -0.559  5.76e-  1 -7.14e4   39741.
 8 NeighborhoodBrDale    -1.88e4  13551.      -1.39   1.65e-  1 -4.54e4    7752.
 9 NeighborhoodBrkSide    8.49e3  11653.       0.729  4.66e-  1 -1.44e4   31351.
10 NeighborhoodClearCr    2.45e4  11954.       2.05   4.05e-  2  1.06e3   47959.
11 NeighborhoodCollgCr    1.47e4   9803.       1.50   1.34e-  1 -4.54e3   33914.
12 NeighborhoodCrawfor    3.78e4  11506.       3.28   1.05e-  3  1.52e4   60350.
13 NeighborhoodEdwards   -1.35e4  10551.      -1.28   1.99e-  1 -3.42e4    7154.
14 NeighborhoodGilbert    1.02e4  10354.       0.987  3.24e-  1 -1.01e4   30525.
15 NeighborhoodIDOTRR    -8.01e3  12357.      -0.649  5.17e-  1 -3.23e4   16225.
16 NeighborhoodMeadowV   -2.27e4  13277.      -1.71   8.79e-  2 -4.87e4    3372.
17 NeighborhoodMitchel   -4.47e3  10843.      -0.413  6.80e-  1 -2.57e4   16797.
18 NeighborhoodNAmes     -1.50e3  10110.      -0.148  8.82e-  1 -2.13e4   18335.
19 NeighborhoodNoRidge    7.12e4  11251.       6.33   3.36e- 10  4.91e4   93240.
```

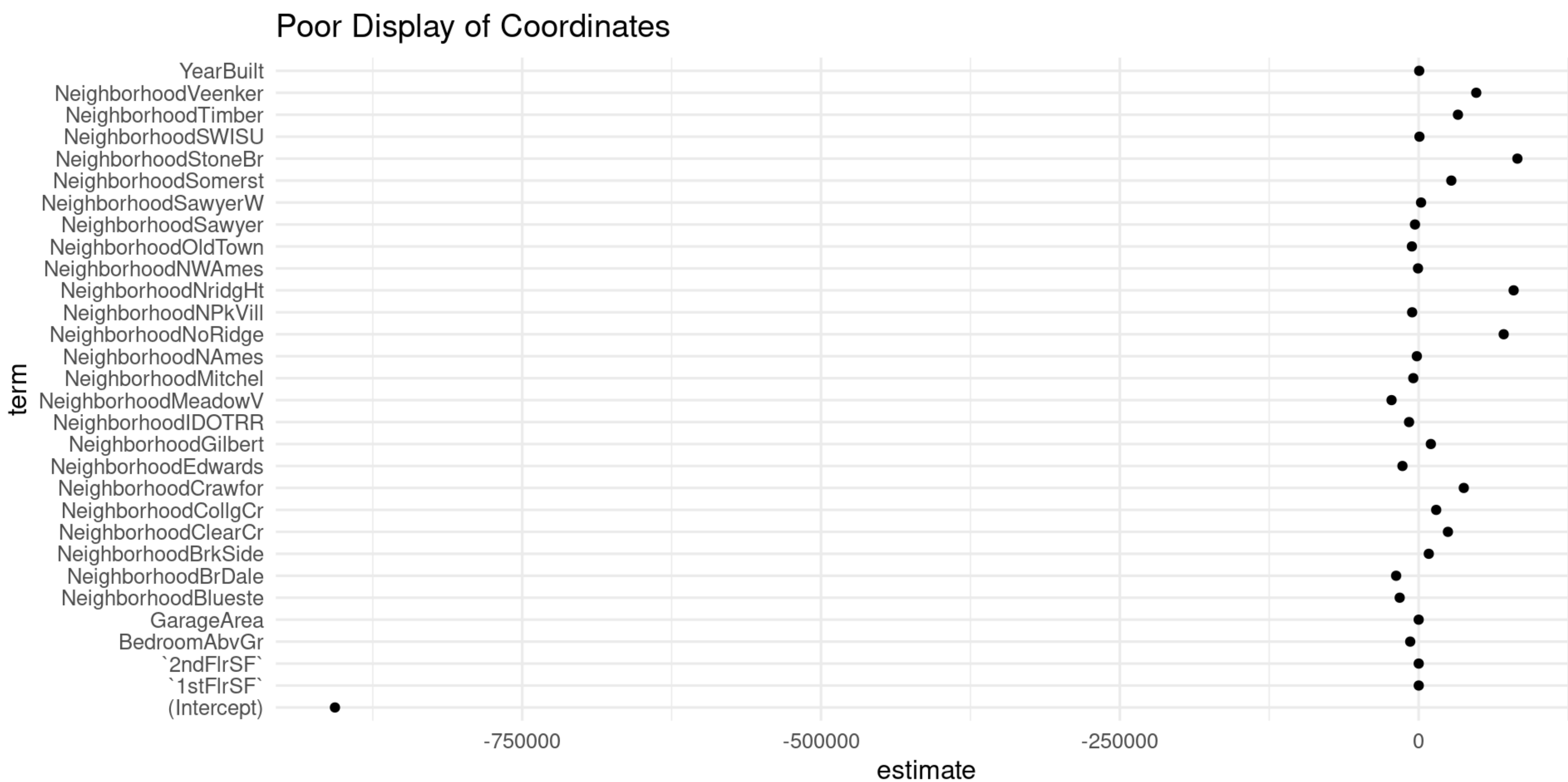
Plotting Coefficients

- `tidy` lets us make plots of coordinates

```
tidy_ames = tidy(mod)
tidy_ames |> ggplot(mapping = aes(x = term,
                                  y = estimate)) +
  geom_point() +
  coord_flip() +
  theme_minimal(base_size=12) +
  labs(title="Poor Display of Coordinates")
```

Plotting Coefficients

- `tidy` lets us make plots of coordinates



Select Neighborhood, Clean Names

```
tidy_ames |> filter(str_detect(term,"Neighborhood")) |>
  mutate(term = str_remove_all(term,"Neighborhood")) |>
  ggplot(mapping = aes(x = term,
                       y = estimate)) +
  geom_point() +
  coord_flip() +
  theme_minimal(base_size=12) +
  labs(title="Neighborhood Effect on Housing Price in Ames",
       subtitle= "Still Needs Uncertainties, to be sorted",
       y="Neighborhood",
       x = "Price Effect (USD)")
```

Select Neighborhood, Clean Names

Sort and add Uncertainty

```
tidy_ames |> filter(str_detect(term,"Neighborhood")) |>
  mutate(term = str_remove_all(term,"Neighborhood")) |>
  ggplot(mapping = aes(x = reorder(term,estimate),
                       y = estimate,
                       ymin = estimate - std.error,
                       ymax = estimate + std.error)) +
  geom_pointrange() +
  coord_flip() +
  theme_minimal(base_size=12) +
  labs(title="Neighborhood Effect on Housing Price in Ames", subtitle = "Er
       y = "Price Effect (USD)",
       x = "Neighborhood")
```

Sort and add Uncertainty

Augment adds fits to data

```r
ames_fit = mod |> augment(ames)
ames_fit |> select(SalePrice:.std.resid)
```

```
# A tibble: 1,460 × 7
   SalePrice .fitted  .resid    .hat .sigma     .cooksd .std.resid
       <dbl>   <dbl>   <dbl>   <dbl>  <dbl>       <dbl>      <dbl>
 1    208500 206886.   1614. 0.00851 37688. 0.000000530     0.0430
 2    181500 203247. -21747. 0.0932  37683. 0.00126        -0.606
 3    223500 215360.   8140. 0.00842 37687. 0.0000133       0.217
 4    140000 195088. -55088. 0.0254  37659. 0.00190        -1.48
 5    250000 308200. -58200. 0.0276  37655. 0.00232        -1.57
 6    143000 168576. -25576. 0.0295  37682. 0.000482       -0.689
 7    307000 244182.  62818. 0.0145  37651. 0.00139         1.68
 8    200000 207423.  -7423. 0.0166  37687. 0.0000222      -0.199
 9    129900 164566. -34666. 0.0117  37676. 0.000338       -0.926
10    118000 124647.  -6647. 0.0185  37687. 0.0000199      -0.178
# ℹ 1,450 more rows
```

`purrr` and `broom`: work with many models

- Use `gapminder` data on GDP and life expectancy

```r
library(gapminder)
gapminder
```

```
# A tibble: 1,704 × 6
   country     continent  year lifeExp      pop gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
 1 Afghanistan Asia       1952    28.8  8425333      779.
 2 Afghanistan Asia       1957    30.3  9240934      821.
 3 Afghanistan Asia       1962    32.0 10267083      853.
 4 Afghanistan Asia       1967    34.0 11537966      836.
 5 Afghanistan Asia       1972    36.1 13079460      740.
 6 Afghanistan Asia       1977    38.4 14880372      786.
 7 Afghanistan Asia       1982    39.9 12881816      978.
 8 Afghanistan Asia       1987    40.8 13867957      852.
 9 Afghanistan Asia       1992    41.7 16317921      649.
10 Afghanistan Asia       1997    41.8 22227415      635.
# ℹ 1,694 more rows
```

`nest` creates a tibble of tibbles

```r
nested_gap = gapminder |>
  group_by(continent,year) |>
  nest()
nested_gap
```

```
# A tibble: 60 × 3
# Groups:   continent, year [60]
   continent  year data
   <fct>     <int> <list>
 1 Asia       1952 <tibble [33 × 4]>
 2 Asia       1957 <tibble [33 × 4]>
 3 Asia       1962 <tibble [33 × 4]>
 4 Asia       1967 <tibble [33 × 4]>
 5 Asia       1972 <tibble [33 × 4]>
 6 Asia       1977 <tibble [33 × 4]>
 7 Asia       1982 <tibble [33 × 4]>
 8 Asia       1987 <tibble [33 × 4]>
 9 Asia       1992 <tibble [33 × 4]>
10 Asia       1997 <tibble [33 × 4]>
# ℹ 50 more rows
```

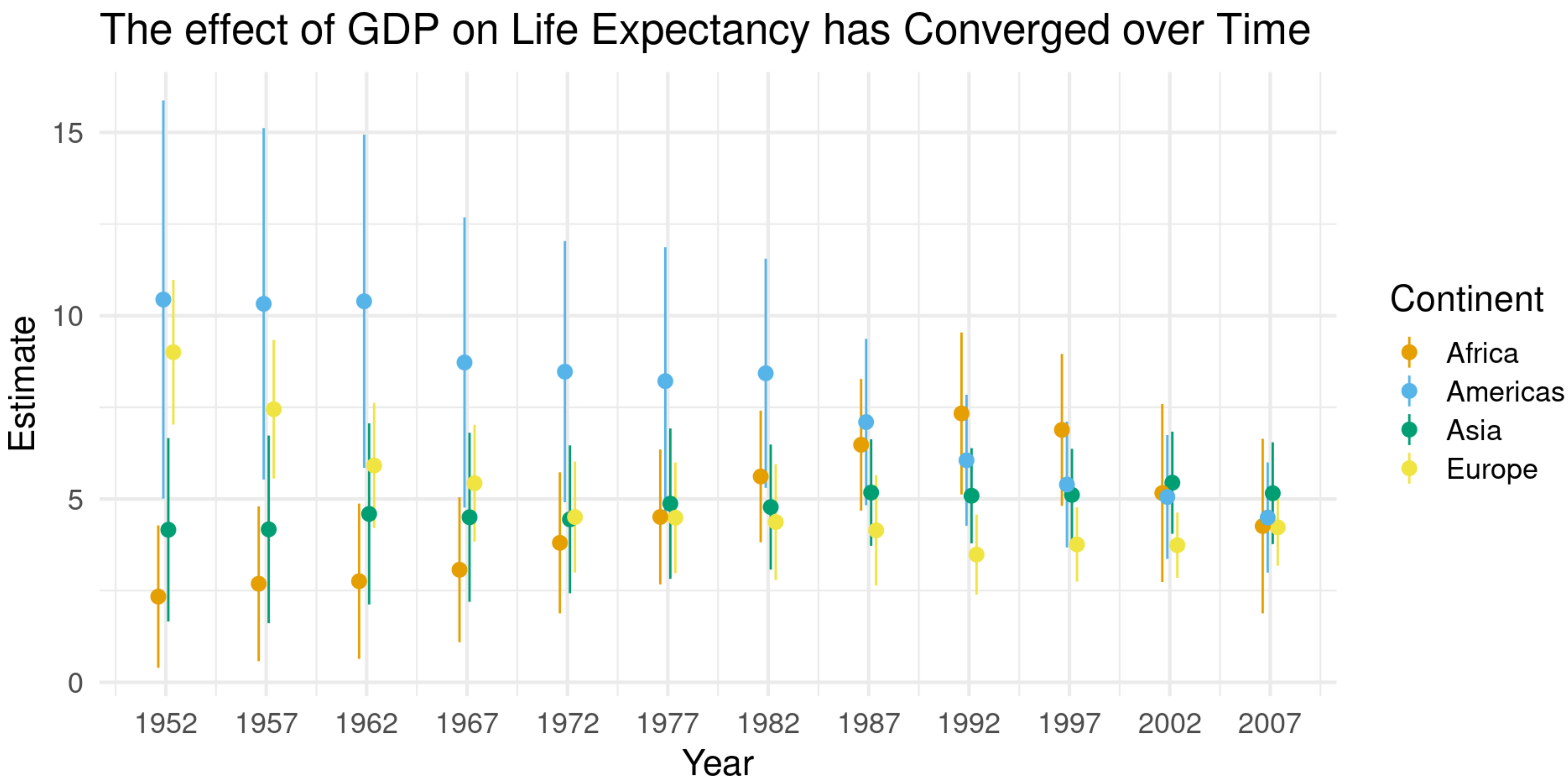
Use map to fit models

```
fit_ols <- function(df) {
    lm(lifeExp ~ log(gdpPercap), data = df)
}

out_tidy = nested_gap |>
    mutate(model = map(data, fit_ols),
           tidied = map(model, tidy))  |>
    unnest(tidied, .drop = TRUE)  |>
    filter(term != "(Intercept)" &
           continent != "Oceania")

out_tidy
```

Use map to fit models

```
# A tibble: 15 × 9
# Groups:   continent, year [15]
   continent  year data      model  term    estimate std.error statistic  p.value
   <fct>     <int> <list>    <list> <chr>      <dbl>     <dbl>     <dbl>    <dbl>
 1 Asia       1952 <tibble> <lm>   log(gd…     4.16     1.25       3.33 2.28e- 3
 2 Asia       1957 <tibble> <lm>   log(gd…     4.17     1.28       3.26 2.71e- 3
 3 Asia       1962 <tibble> <lm>   log(gd…     4.59     1.24       3.72 7.94e- 4
 4 Asia       1967 <tibble> <lm>   log(gd…     4.50     1.15       3.90 4.77e- 4
 5 Asia       1972 <tibble> <lm>   log(gd…     4.44     1.01       4.41 1.16e- 4
 6 Asia       1977 <tibble> <lm>   log(gd…     4.87     1.03       4.75 4.42e- 5
 7 Asia       1982 <tibble> <lm>   log(gd…     4.78     0.852      5.61 3.77e- 6
 8 Asia       1987 <tibble> <lm>   log(gd…     5.17     0.727      7.12 5.31e- 8
 9 Asia       1992 <tibble> <lm>   log(gd…     5.09     0.649      7.84 7.60e- 9
10 Asia       1997 <tibble> <lm>   log(gd…     5.11     0.628      8.15 3.35e- 9
11 Asia       2002 <tibble> <lm>   log(gd…     5.44     0.696      7.82 8.05e- 9
12 Asia       2007 <tibble> <lm>   log(gd…     5.16     0.694      7.43 2.26e- 8
13 Europe     1952 <tibble> <lm>   log(gd…     9.00     0.987      9.12 7.04e-10
14 Europe     1957 <tibble> <lm>   log(gd…     7.45     0.945      7.88 1.39e- 8
15 Europe     1962 <tibble> <lm>   log(gd…     5.91     0.853      6.93 1.56e- 7
```

Use Map to Fit Models



Marginal Effects Package

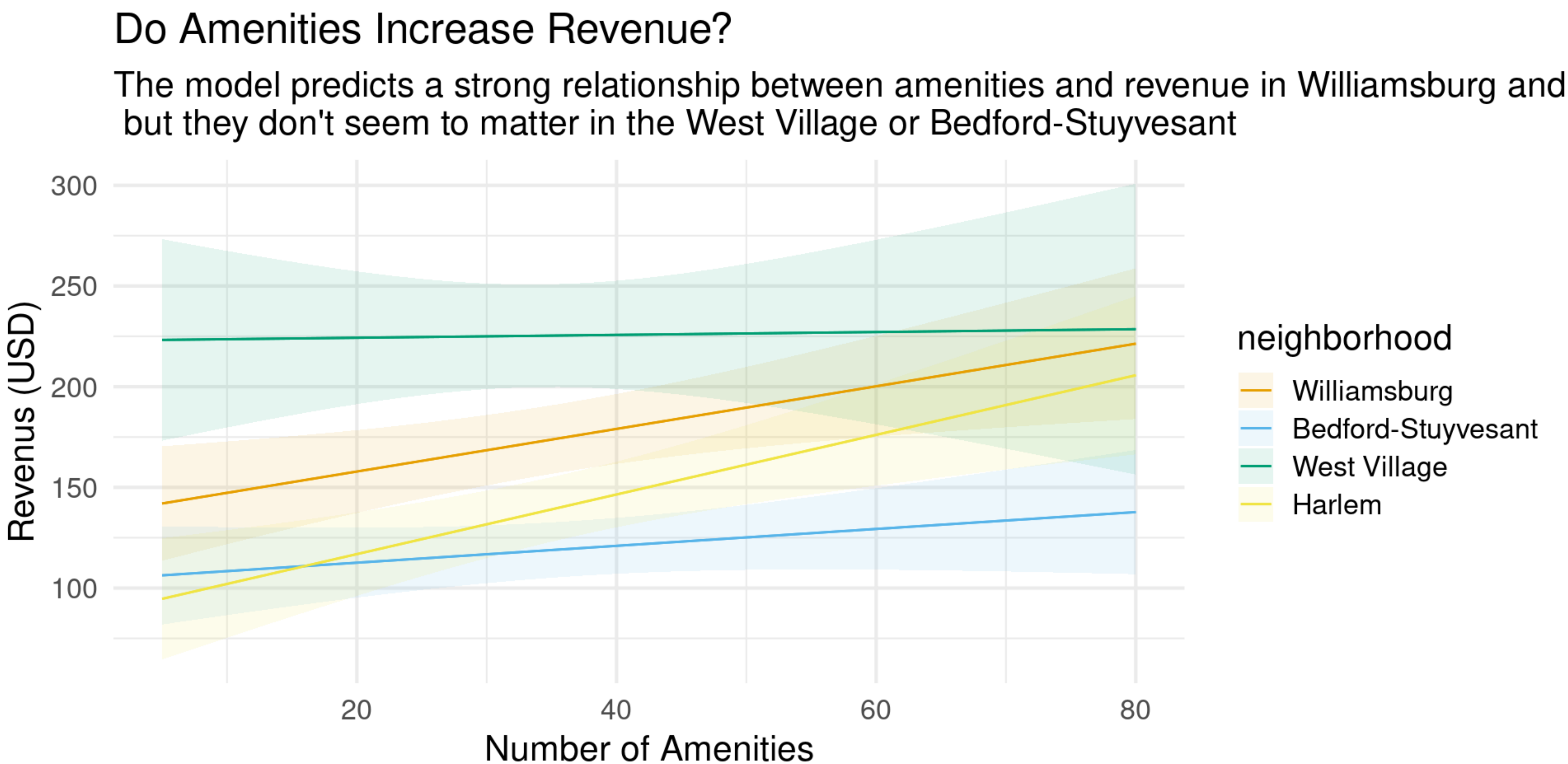
- `marginaleffects` is a powerful package in both `python` and R for model interpretation
- Comes with a book Marginal Effects
- Several alternatives exist (detailed there)
- Can be used with nearly any type of model, from statistical, to Bayesian, to Machine Learning

Plotting Predictions

- `plot_predictions` function (also `predictions`)
- Specify `grid` of data on which to predict
- Do this with `condition`

```r
mod = lm(revenue ~ . + host_is_superhost*bedrooms +
              host_is_superhost*neighborhood +
              num_amenities*neighborhood +
              num_amenities*bedrooms   ,
            data = nyc_bnb_slim)

mod |> plot_predictions(condition = c("num_amenities", "neighborhood")) +
  theme_minimal(base_size = 16) +
  labs(title = "Amenities increase revenues",
       x="Number of Amenities",
       y="Revenus (USD)")
```

Plotting Predictions



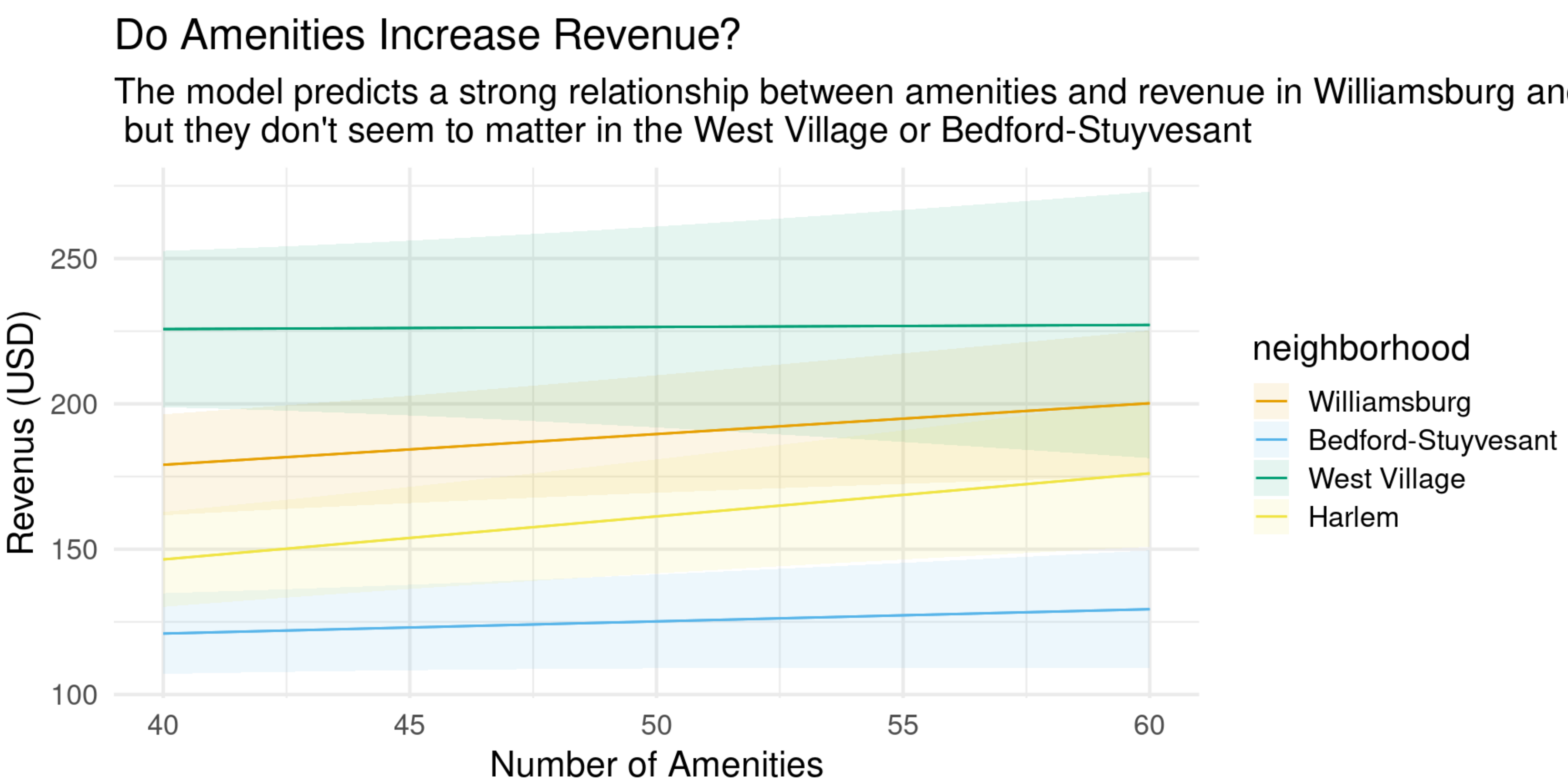
Changing the Conditions

- Can provide a range of conditions

```
mod |> plot_predictions(condition =
            list("num_amenities" = 40:60,
              "neighborhood")) +
  theme_minimal(base_size = 16) +
  labs(title = "Do Amenities Increase Revenue?",
       subtitle = "The model predicts a strong relationship between ameniti
       x="Number of Amenities",
       y="Revenus (USD)")
```

Changing the Conditions

- Can provide a range of conditions

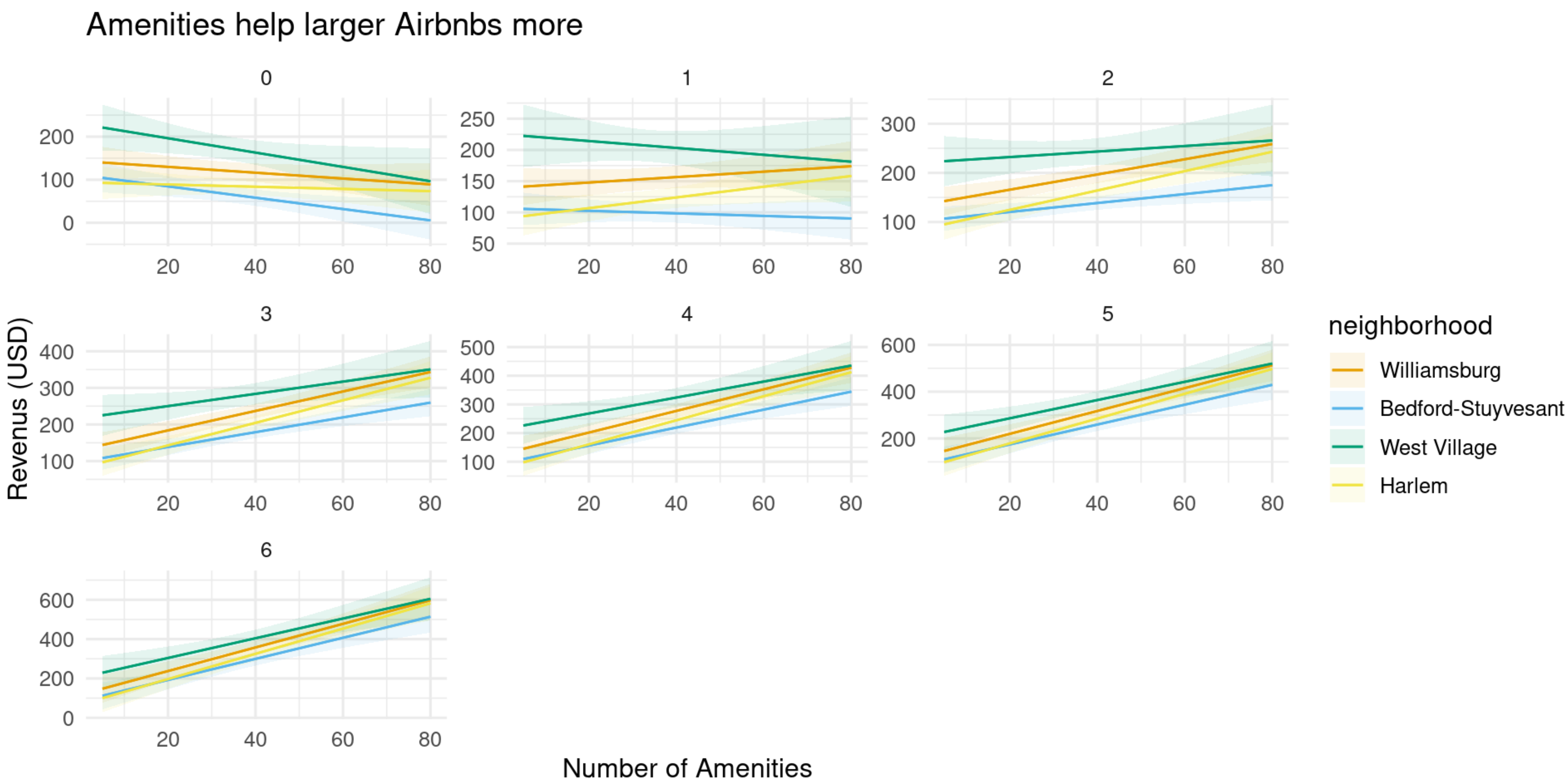


Changing the Conditions

- Can split into subplots

```
mod |> plot_predictions(condition =
             list("num_amenities",
               "neighborhood",
               "bedrooms" = unique)) +
  theme_minimal(base_size = 16) +
  labs(title = "Do Amenities Increase Revenue?",
       x="Number of Amenities",
       y="Revenus (USD)")
```

Changing the Conditions

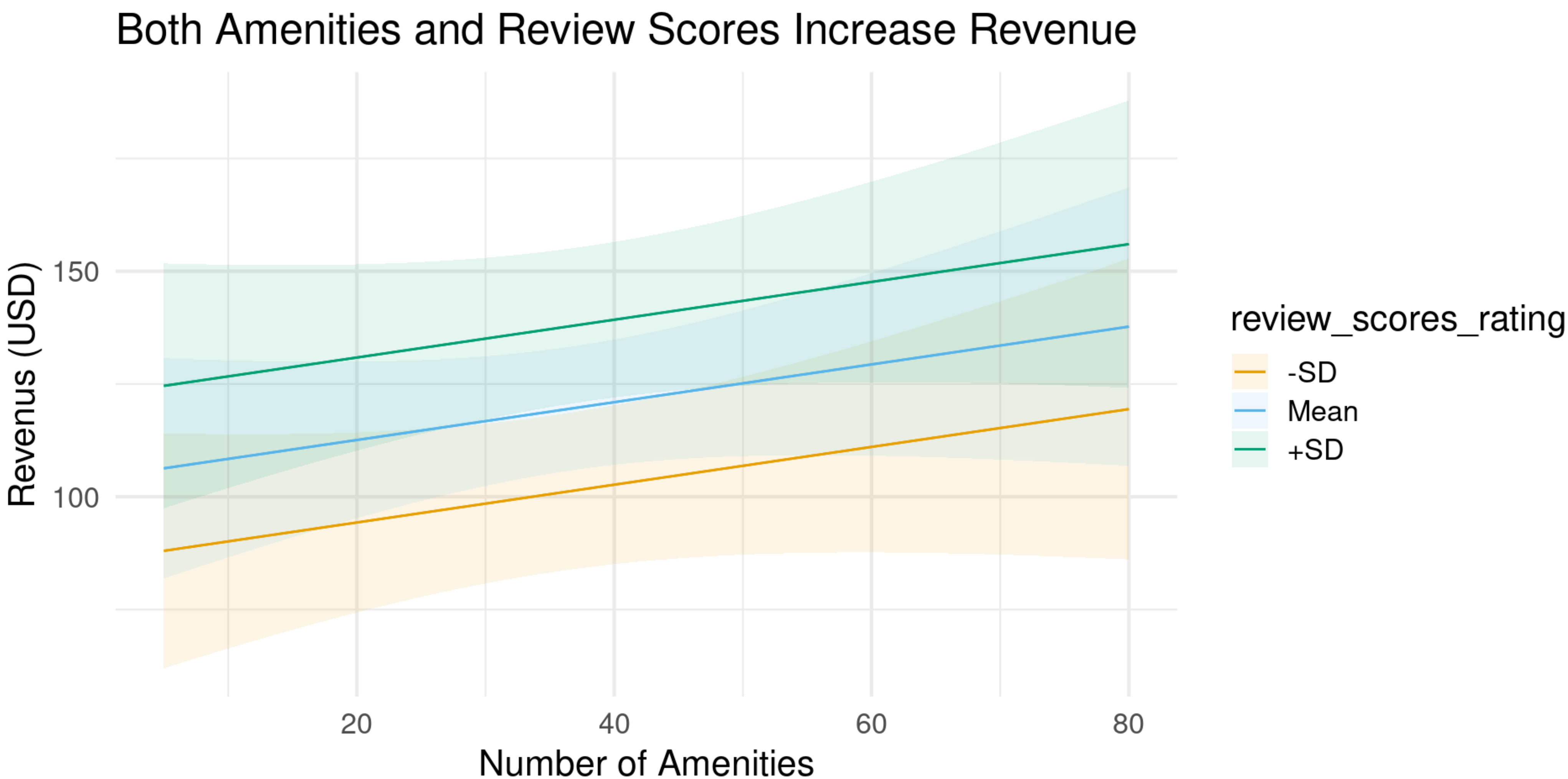


Adding Cases

- “threenum” keyword plots $\text{mean} \pm \sigma$
- Other options include “minmax”, “quartile”, check ?plot_predictions

```r
mod |> plot_predictions(condition =
              list("num_amenities",
                   "review_scores_rating" = "threenum")) +
  theme_minimal(base_size = 16) +
  labs(title = "Do Amenities Increase Revenue?",
       subtitle = "The model predicts a strong relationship between ameniti
       x="Number of Amenities",
       y="Revenus (USD)")
```

Adding Cases



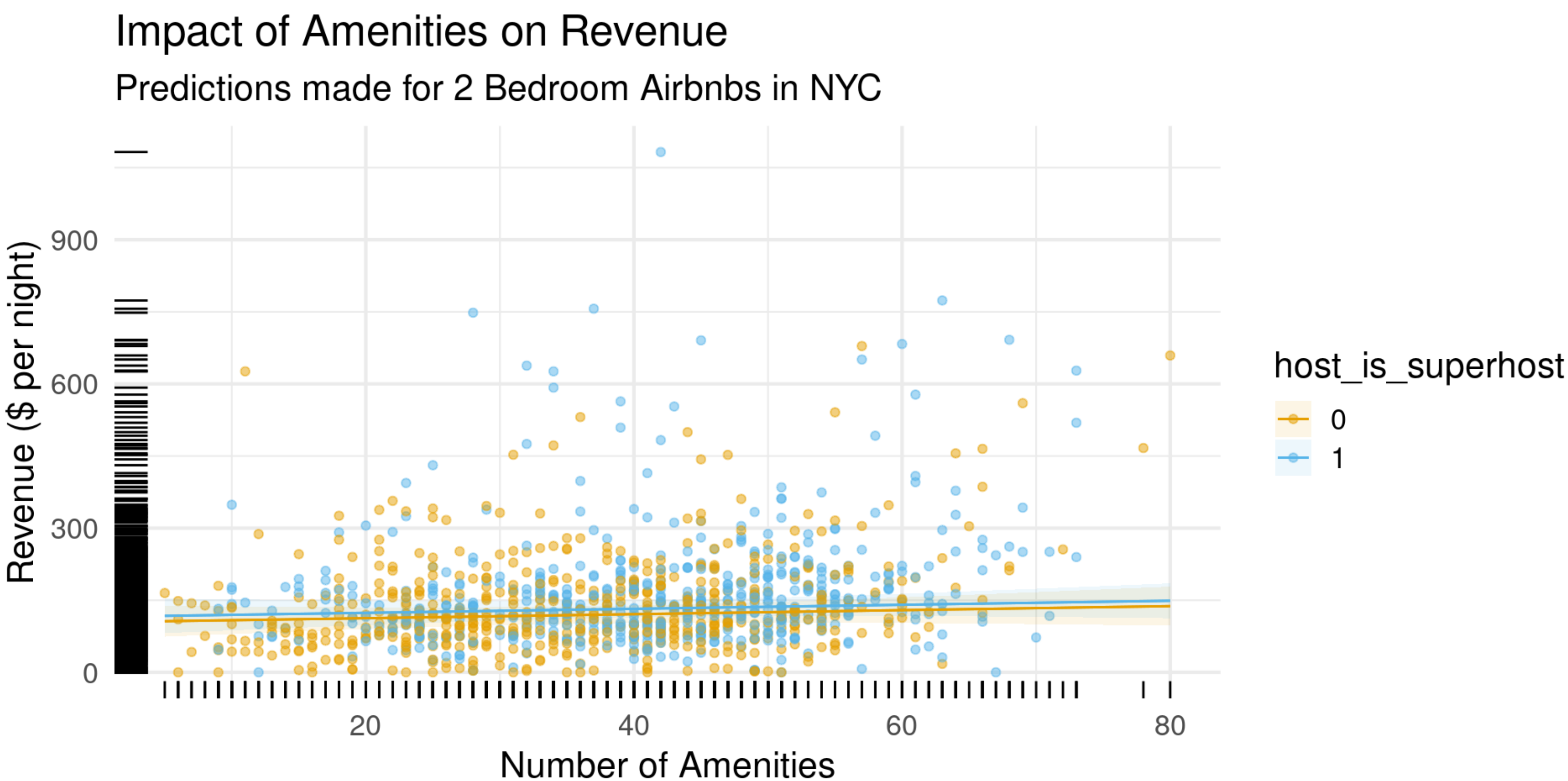
Show Your Data

- If you can, plot your data along with predictions

```
plot_predictions(
    mod,conf_level = .99,
    condition = list(
        "num_amenities",
        "host_is_superhost"), points = .5, rug = TRUE) +
  theme_minimal(base_size=16) +
  labs(title = "Impact of Amenities on Revenue",
       subtitle = "Predictions made for 2 Bedroom Airbnbs in NYC", x = "Num
```

Show Your Data

- If you can, plot your data along with predictions

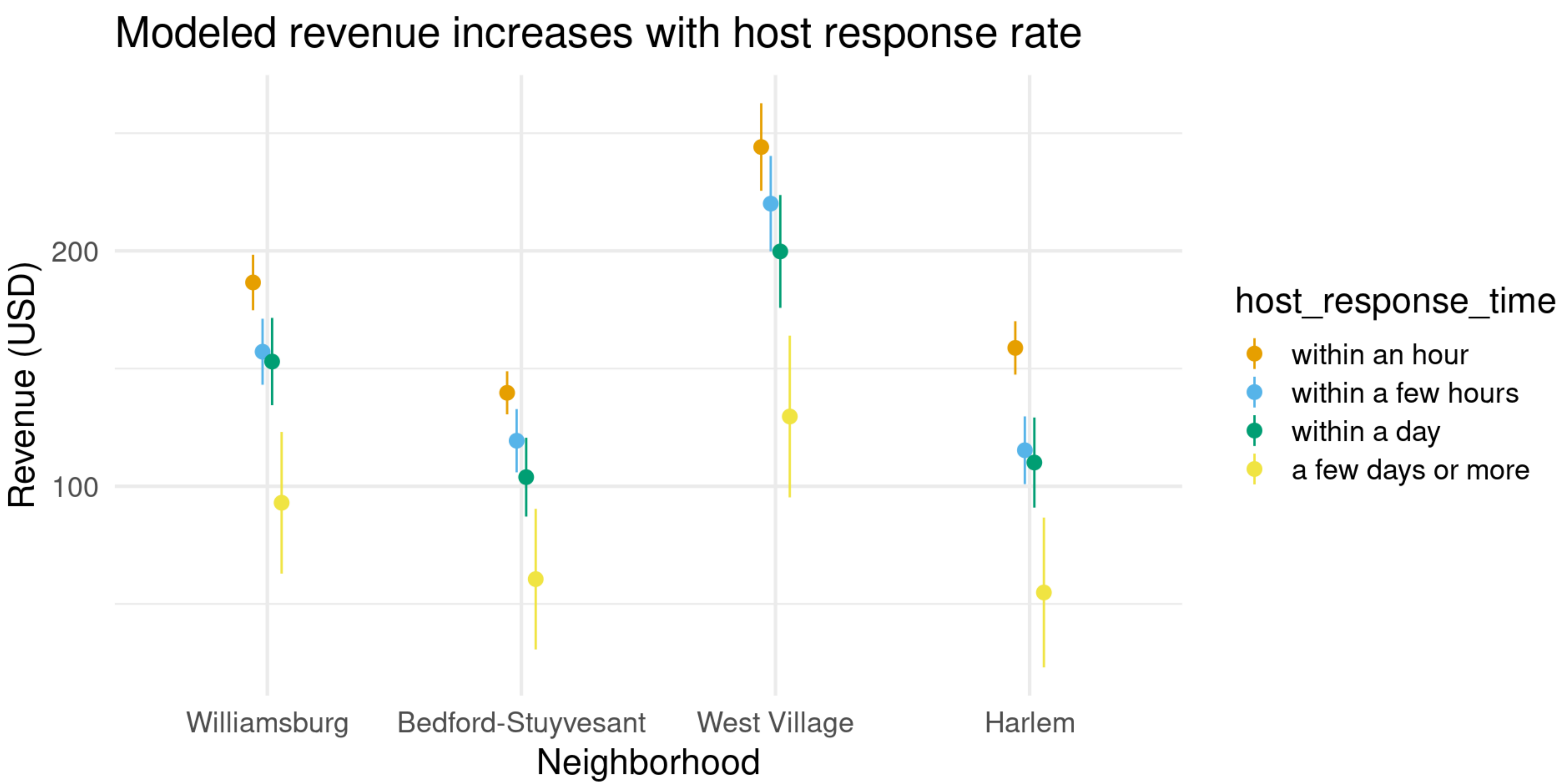


Marginal Predictions

- Marginal prediction is averaged over a grouping
- Use by keyword to specify aggregation

```
mod |> plot_predictions(by = c("neighborhood", "host_response_time")) +
  theme_minimal(base_size=16) +
  labs(title = "Modeled revenue increases with host response rate",
       y = "Revenue (USD)",
       x = "Neighborhood")
```

Marginal Predictions



Plotting Comparisons

- Comparisons with `plot_comparisons`
- Similar syntax to `plot_predictions`
- Very useful for categorical outcomes

Thornton Revisited

- Thornton dataset: How much does a monetary incentive impact the decision to get an HIV test?

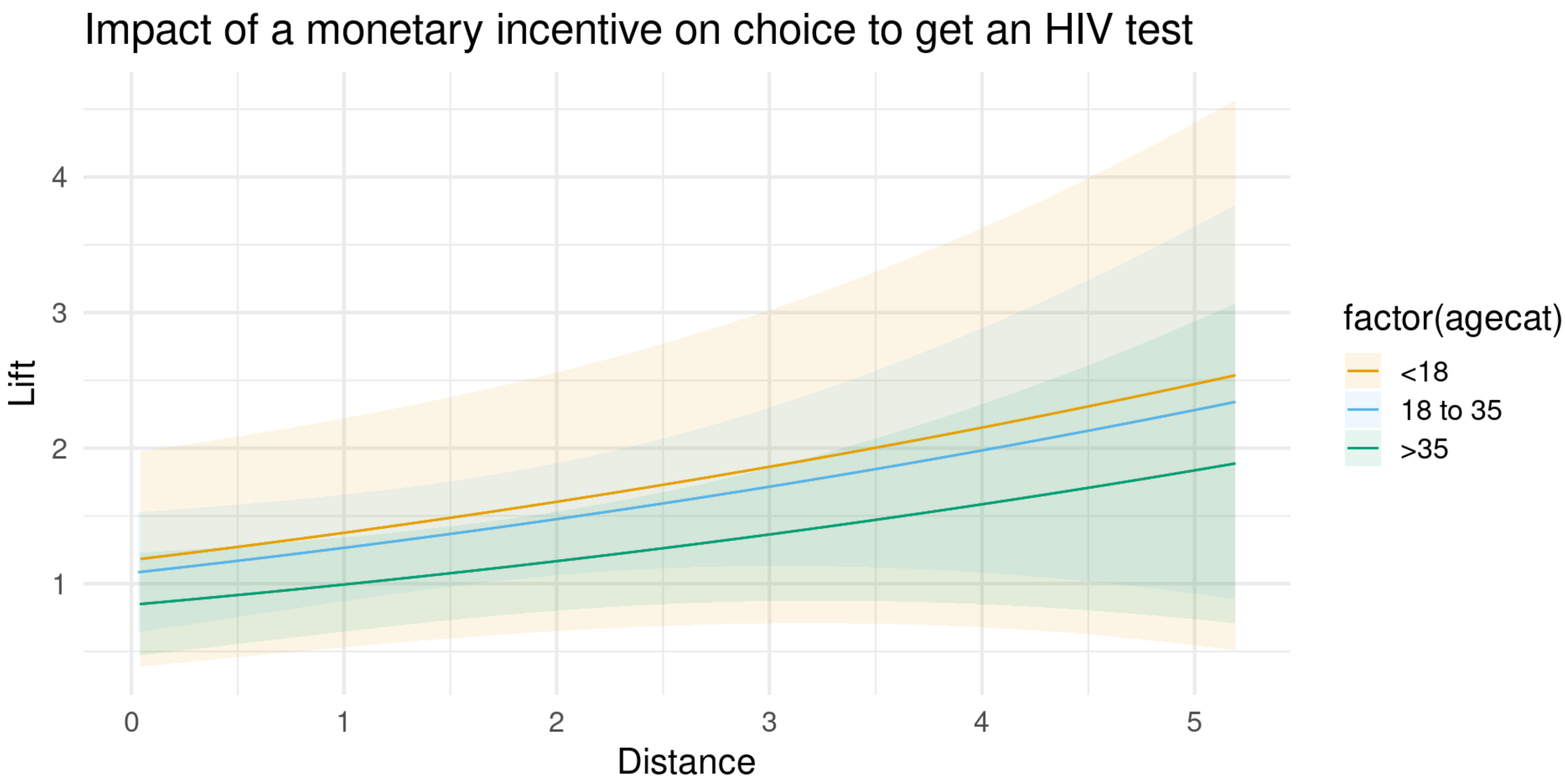
```
Call:
lm(formula = revenue ~ . + host_is_superhost * bedrooms + host_is_superhost *
    neighborhood + num_amenities * neighborhood + num_amenities *
    bedrooms, data = nyc_bnb_slim)

Residuals:
    Min      1Q  Median      3Q     Max
-282.29  -50.44   -7.98   40.73  654.75

Coefficients: (1 not defined because of singularities)
                                                   Estimate Std. Error t value
(Intercept)                                      -156.33072  126.44123  -1.236
neighborhoodBedford-Stuyvesant                    -32.49483   20.36231  -1.596
neighborhoodWest Village                           86.16582   32.33182   2.665
```

Comparisons

- `variable` is what is compared across
- `by` provides additional levels

```
plot_comparisons(mod_thornton,
  variables = c("incentive"),
  by = c("agecat"),
  comparison = "lift") +
  theme_minimal(base_size=16) +
  labs(title = "Impact of a monetary incentive on choice to get an HIV test
```

Comparisons



Comparison Functions

- `difference`, `ratio`, `odds`, arbitrary functions possible

$$\hat{Y}_{X=b} - \hat{Y}_{X=a} \qquad \text{Difference}$$

$$\frac{\hat{Y}_{X=b}}{\hat{Y}_{X=a}} \qquad \text{Ratio}$$

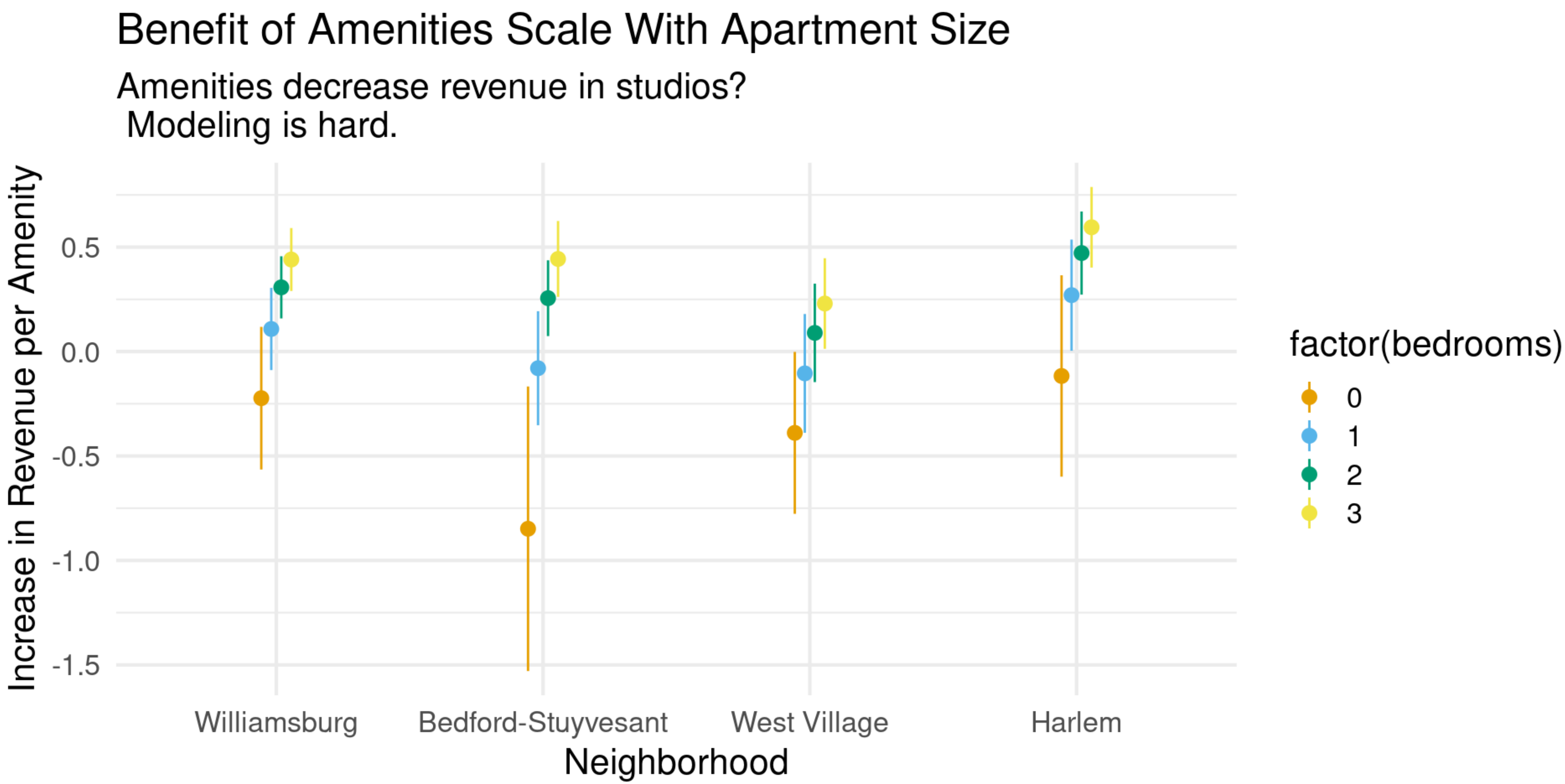
$$\frac{\hat{Y}_{X=b} - \hat{Y}_{X=a}}{\hat{Y}_{X=a}} \qquad \text{Lift}$$

Slopes

- The slope is the sensitivity of outcome to a predictor
- `eyex` keyword computes elasticity

```
plot_slopes(mod,
  variables = "num_amenities",
  slope = "eyex",
  condition = list("neighborhood", "bedrooms"= 0:3)) +
  theme_minimal(base_size=16) +
  labs(title="Investments in Amenities Scale With Apartment Size",
       subtitle= "Decrease in revenue for studios suggests either a flaw in
```

Slopes


Benefit of Amenities Scale With Apartment Size
Amenities decrease revenue in studios?
Modeling is hard.
Increase in Revenue per Amenity
0.5
0.0
-0.5
-1.0
-1.5
Williamsburg
Bedford-Stuyvesant
West Village
Harlem
Neighborhood
factor(bedrooms)
0
1
2
3

Thanks!