Many models with R `tidyverse` tools

Visualizing Home Sales Data, Garfield County, CO

Casey Bates

1/31/2019

Motivation

Utilize `tidyverse` tools and the `broom` package to generate numerous linear models to evaluate how well sale price is explained by square footage for home sales in Garfield County, Colorado. A dataset is publically available on the Garfield County Assessor website that contains 2 years of data from summer 2014 through summer 2016.

Outline

- Part 1: Exploring the dataset with `ggplot2`
  - Tidying wide datasets with `tidyr`
- Part 2: Many models with `purrr` and `broom`

Part 1: Exploring the dataset with `ggplot2`

Processing the data

- Import two Excel files:
  1. single family home sales, and
  2. condo & townhome sales
- Replace spaces in column names with underscore and make lowercase
- Rename some columns
- Add `classification` column to `single_family` dataset
  - Set all values to "Single Family"
- Use `bind_rows()` to combine the datasets into one
- Remove "Garage Only" observations

Glimpse of the data

```
glimpse(home_sales)
```

```
## Observations: 1,967
## Variables: 14
## $ account              <chr> "R340967", "R340073", "R112
## $ parcel_number        <chr> "239334401005", "2393342000
## $ reception            <chr> "879240", "870778", "869383
## $ sale_date            <chr> "6/29/2016", "11/24/2015", "
## $ sale_price           <dbl> 650000, 560000, 2750000, 63
## $ situs_address        <chr> "000066 N 2ND ST", "000276 1
## $ location             <chr> "CARBONDALE", "CARBONDALE",
## $ architectural_style  <chr> "ONE STORY", "ONE STORY", "O
## $ year_built           <dbl> 1970, 1971, 2002, 1999, 2008
## $ bedrooms             <dbl> 0, 1, 0, 1, 2, 1, 1, 1, 2, 2
## $ baths                <dbl> 0.00, 1.00, 0.75, 1.00, 1.00
## $ square_feet          <dbl> 0, 480, 680, 710, 764, 804,
## $ legal                <chr> "Section: 34 Township: 7 Ran
## $ classification       <chr> "Single Family", "Single Fam
```

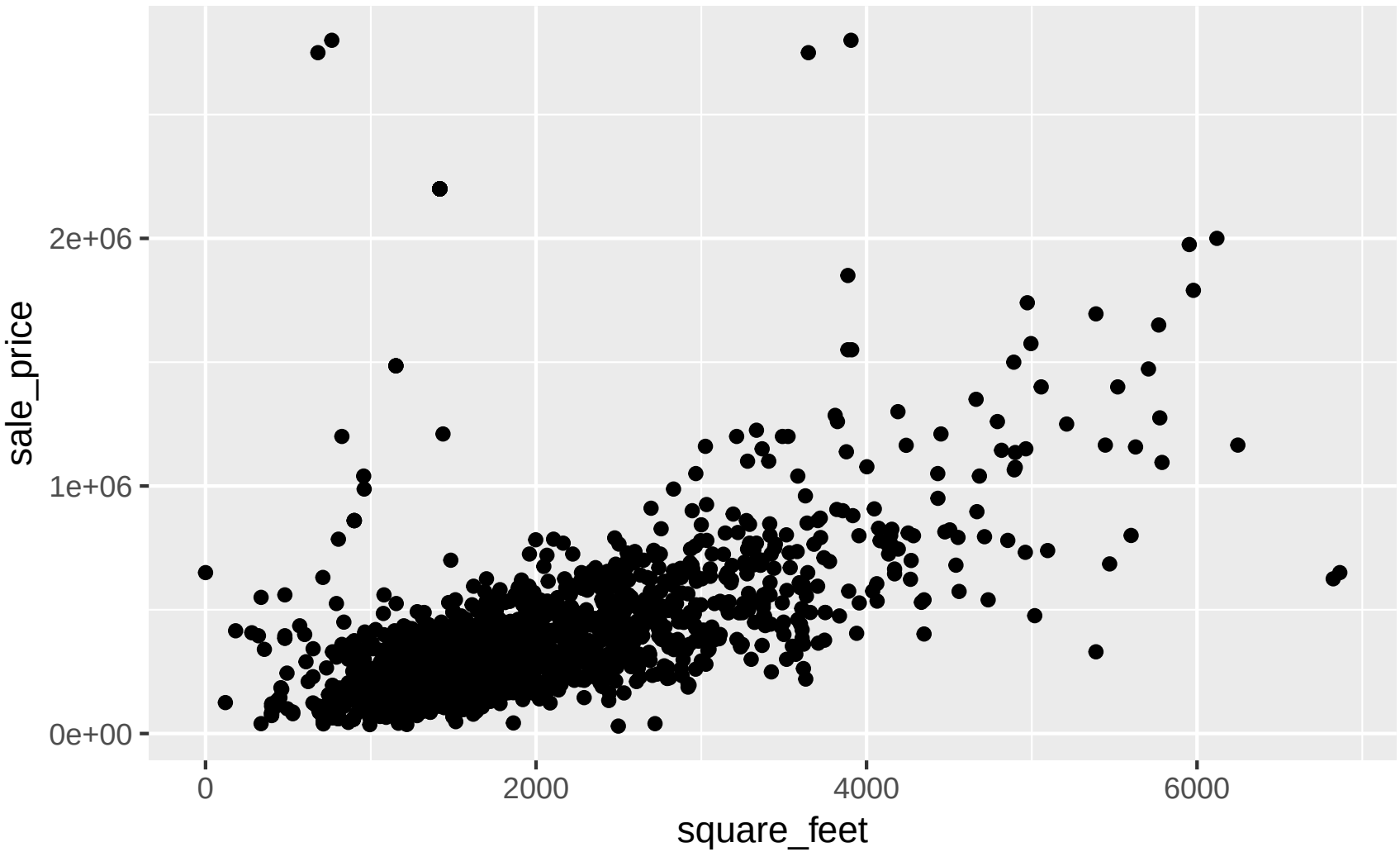
ggplot2 package in R

- Created by Hadley Wickham
- Built on the "Grammar of Graphics" principles
- Core **tidyverse** package
- Every ggplot2 plot has 3 key components:
  - **Data**
  - **Aesthetic mappings** between variables and visuals
  - Layer(s) to describe how to render each observation (usually created with a **geom** function)

Basic scatterplot

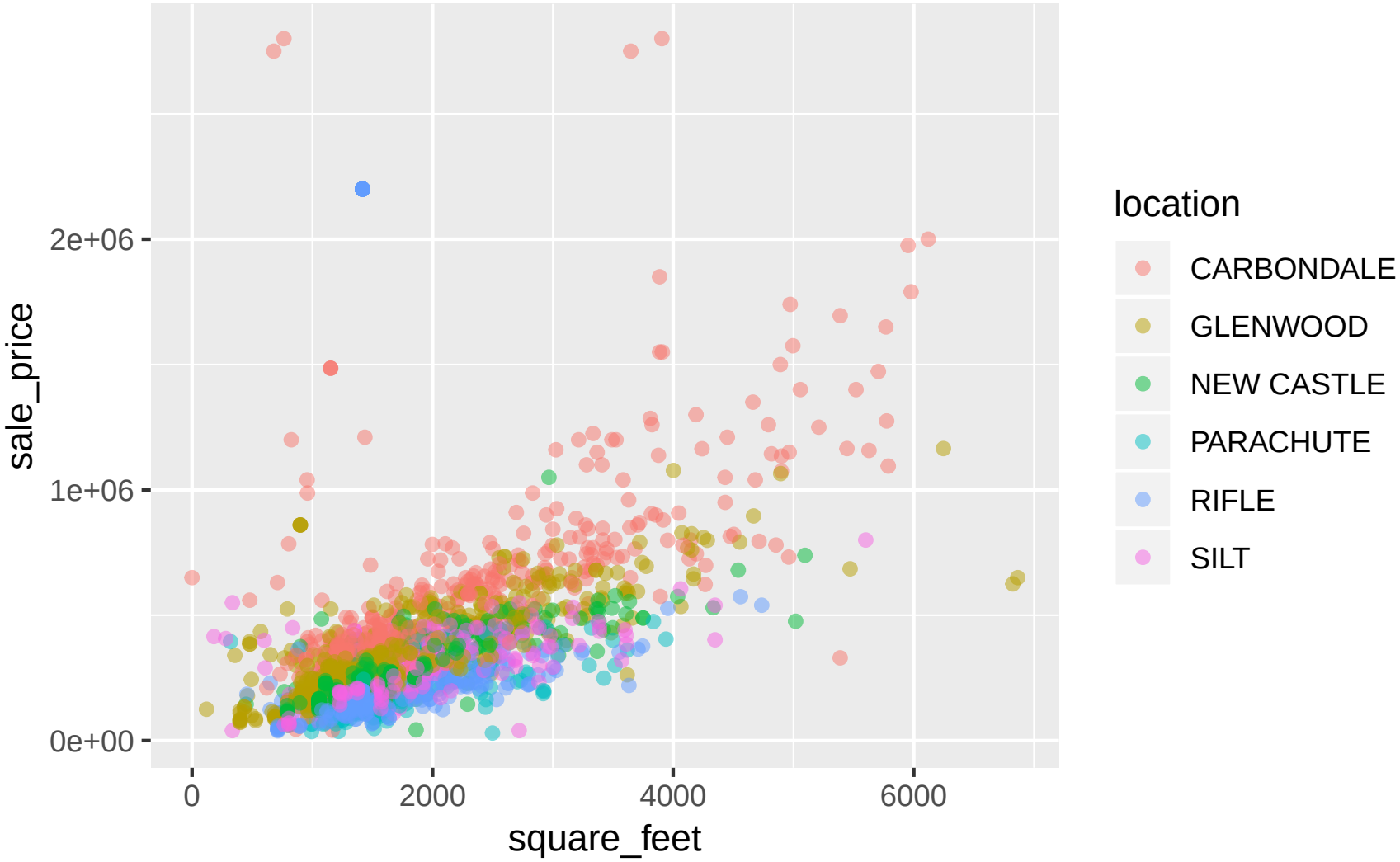
```
ggplot(data = home_sales, aes(x = square_feet, y = sale_pri
  geom_point()
```

```
## Warning: Removed 1 rows containing missing values (geom_
```



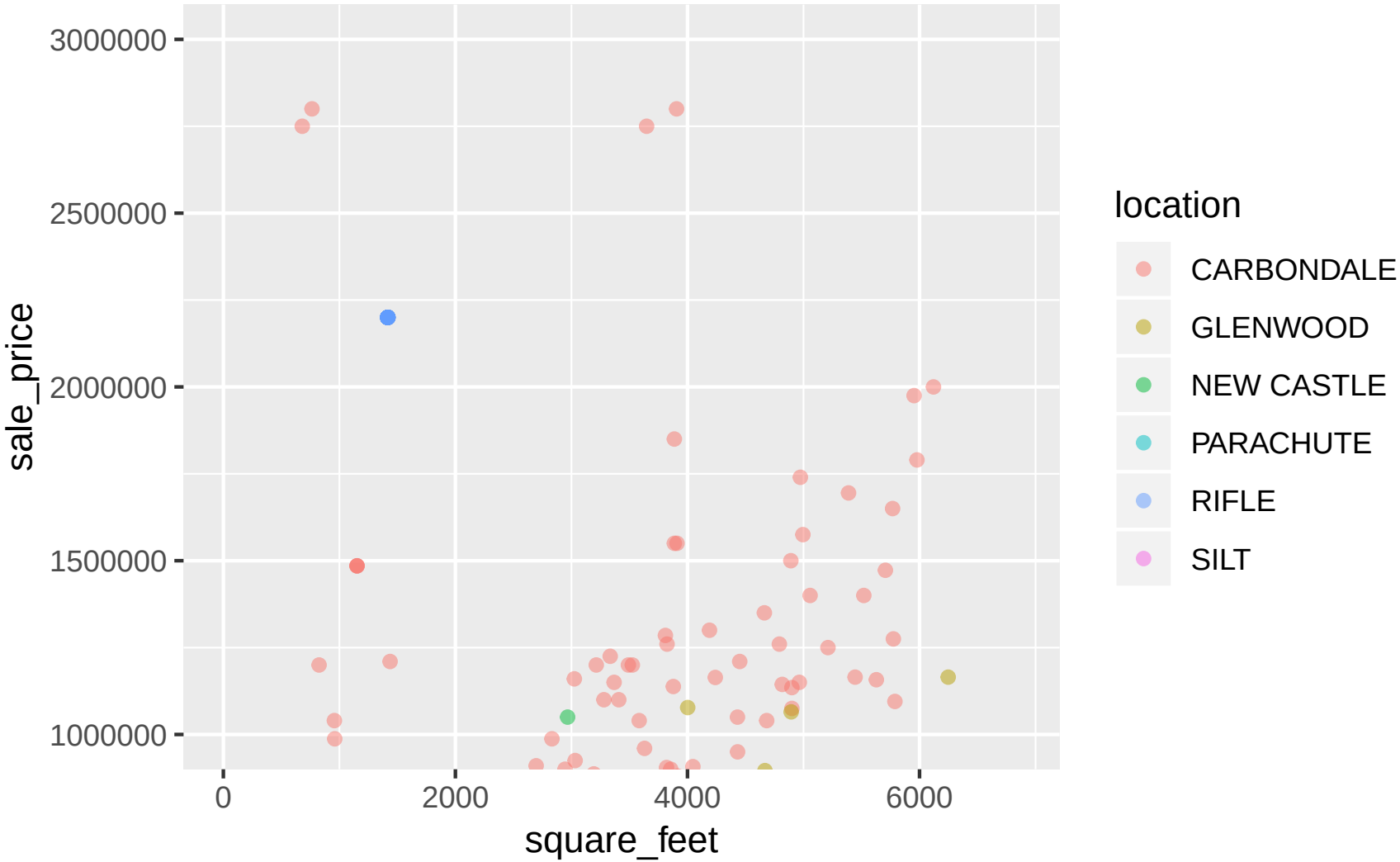
Transparency and color by location

```
ggplot(data = home_sales, aes(x = square_feet, y = sale_pri
  geom_point(alpha = 0.5)
```



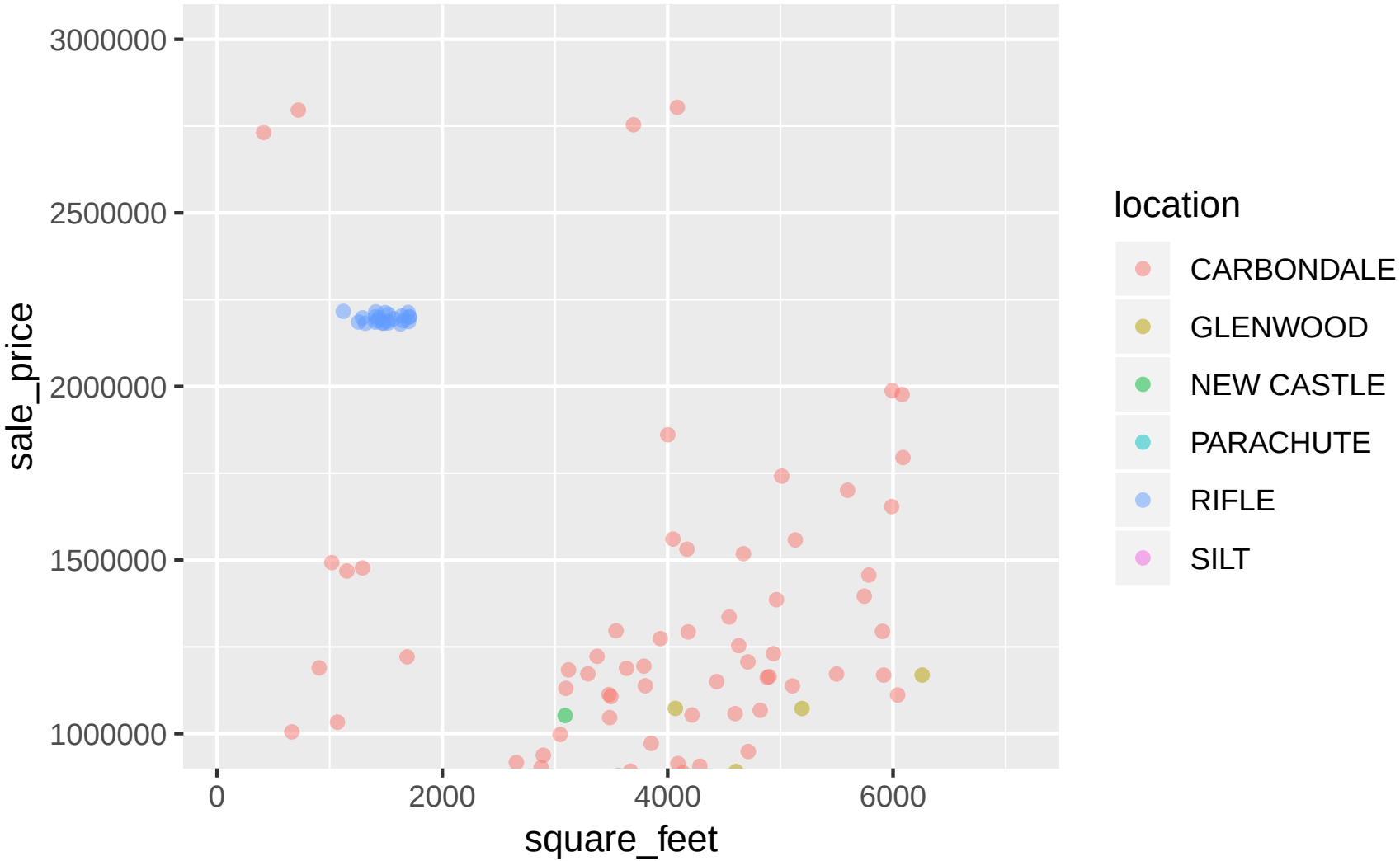
Zooming into sales above $1M

```
ggplot(data = home_sales, aes(x = square_feet, y = sale_pri
  geom_point(alpha = 0.5) +
  coord_cartesian(ylim = c(1000000, 3000000))
```



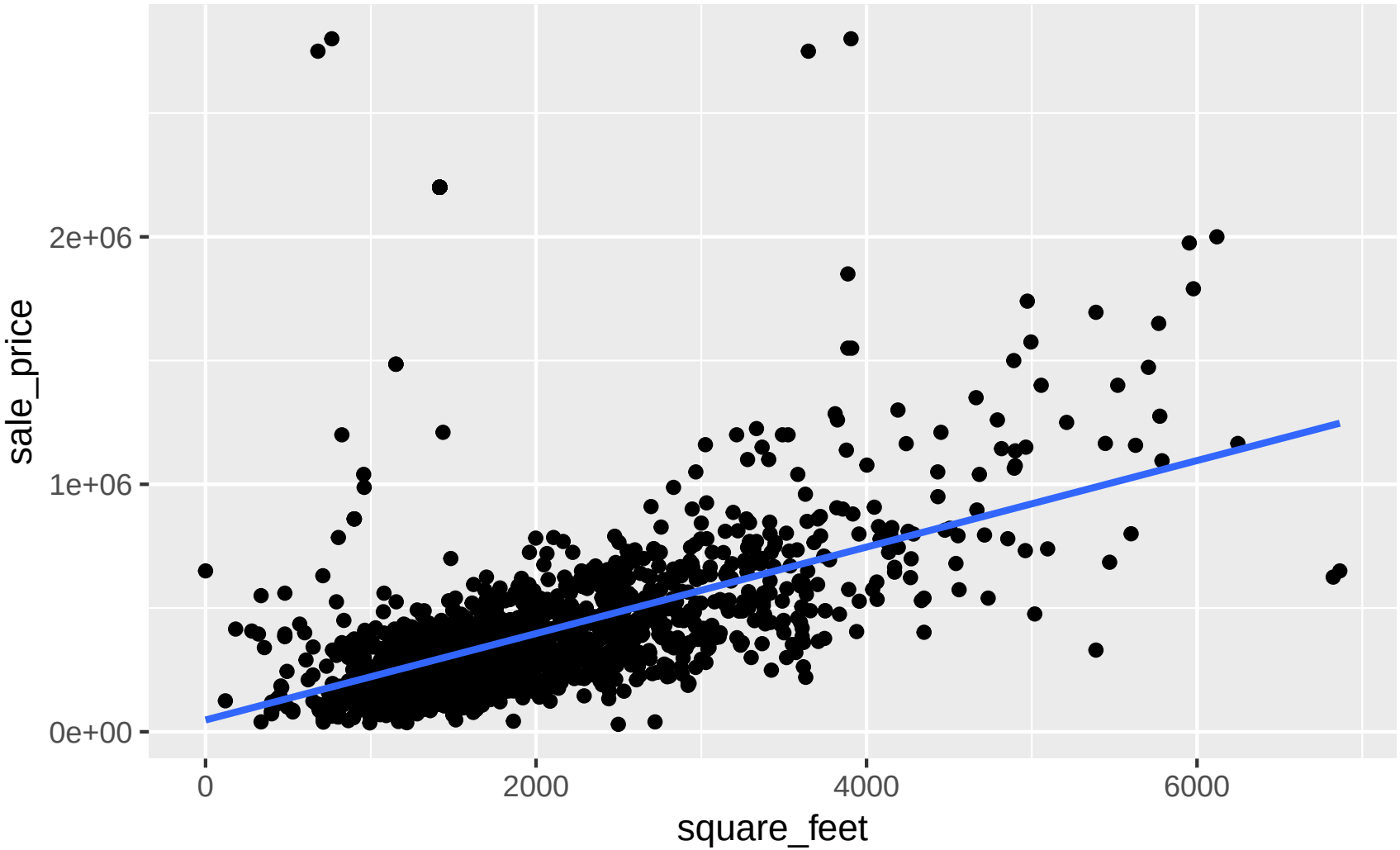
Add random noise with jitter

```
ggplot(data = home_sales, aes(x = square_feet, y = sale_pri
  geom_jitter(alpha = 0.5, width = 300, height = 20000) +
  coord_cartesian(ylim = c(1000000, 3000000))
```



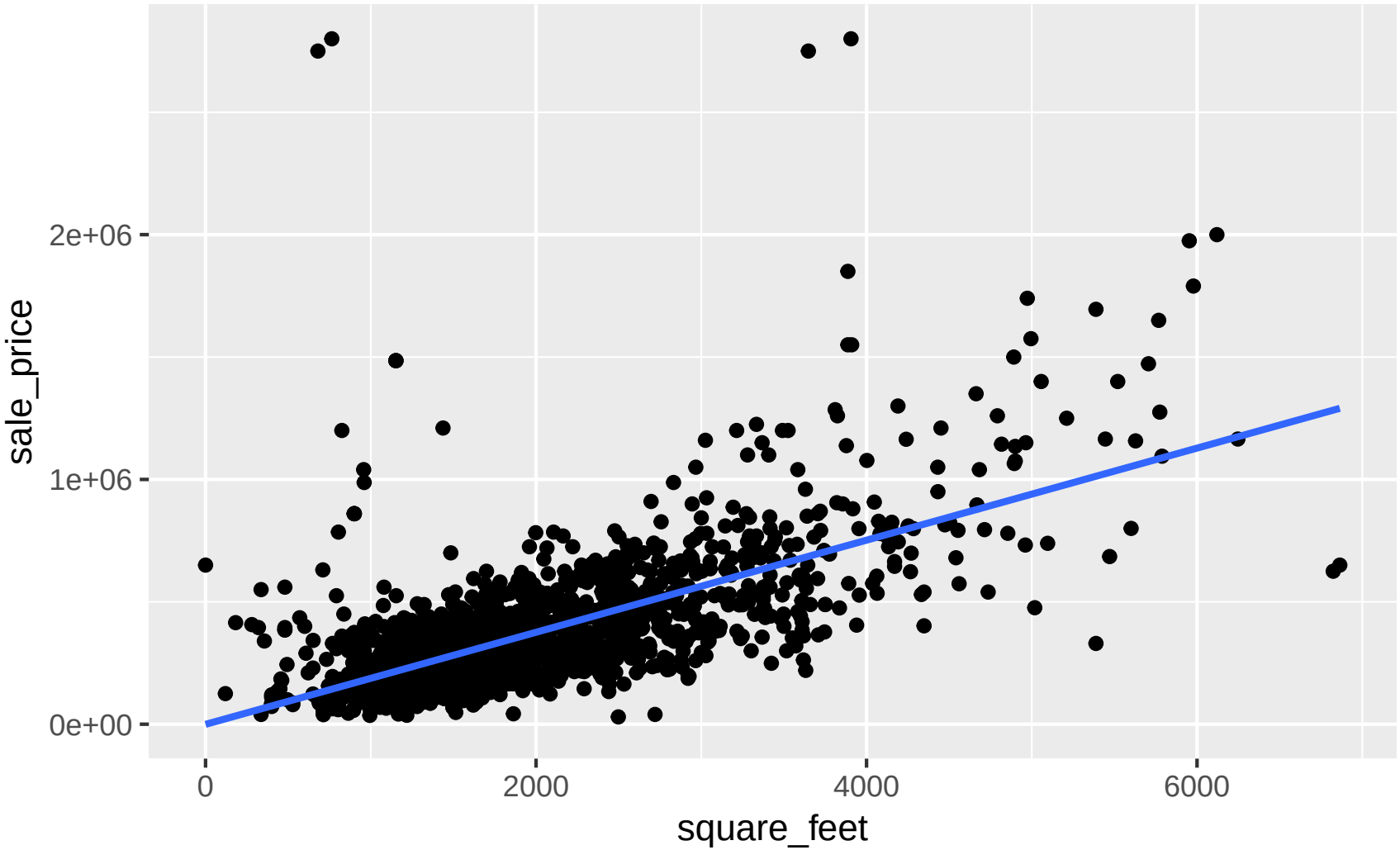
Linear model: sale price vs. square ft.

```
ggplot(data = home_sales, aes(x = square_feet, y = sale_pri
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to lm
```



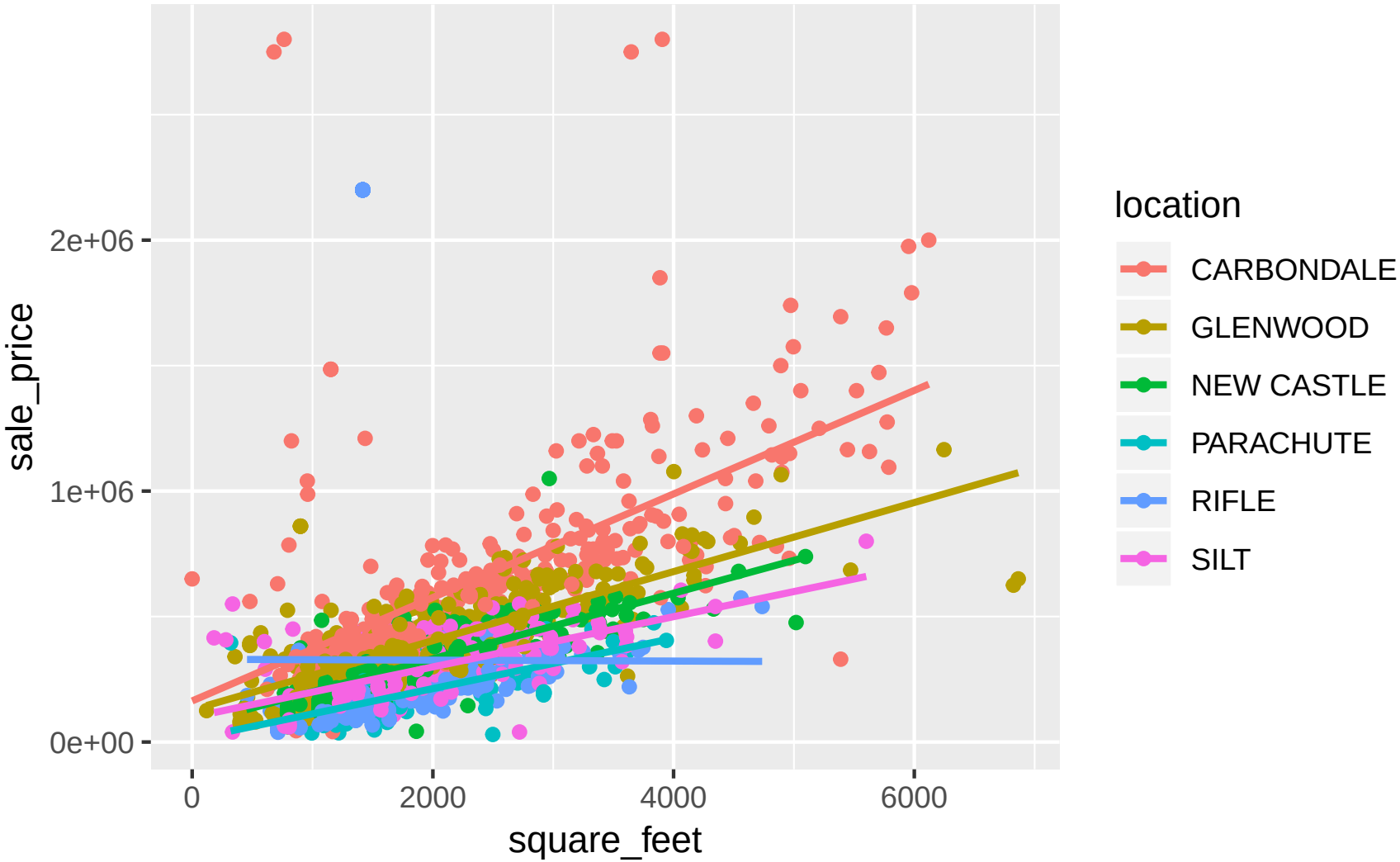
Linear model: sale price vs. square ft.

```
ggplot(data = home_sales_fix, aes(x = square_feet, y = sale
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to lm
```



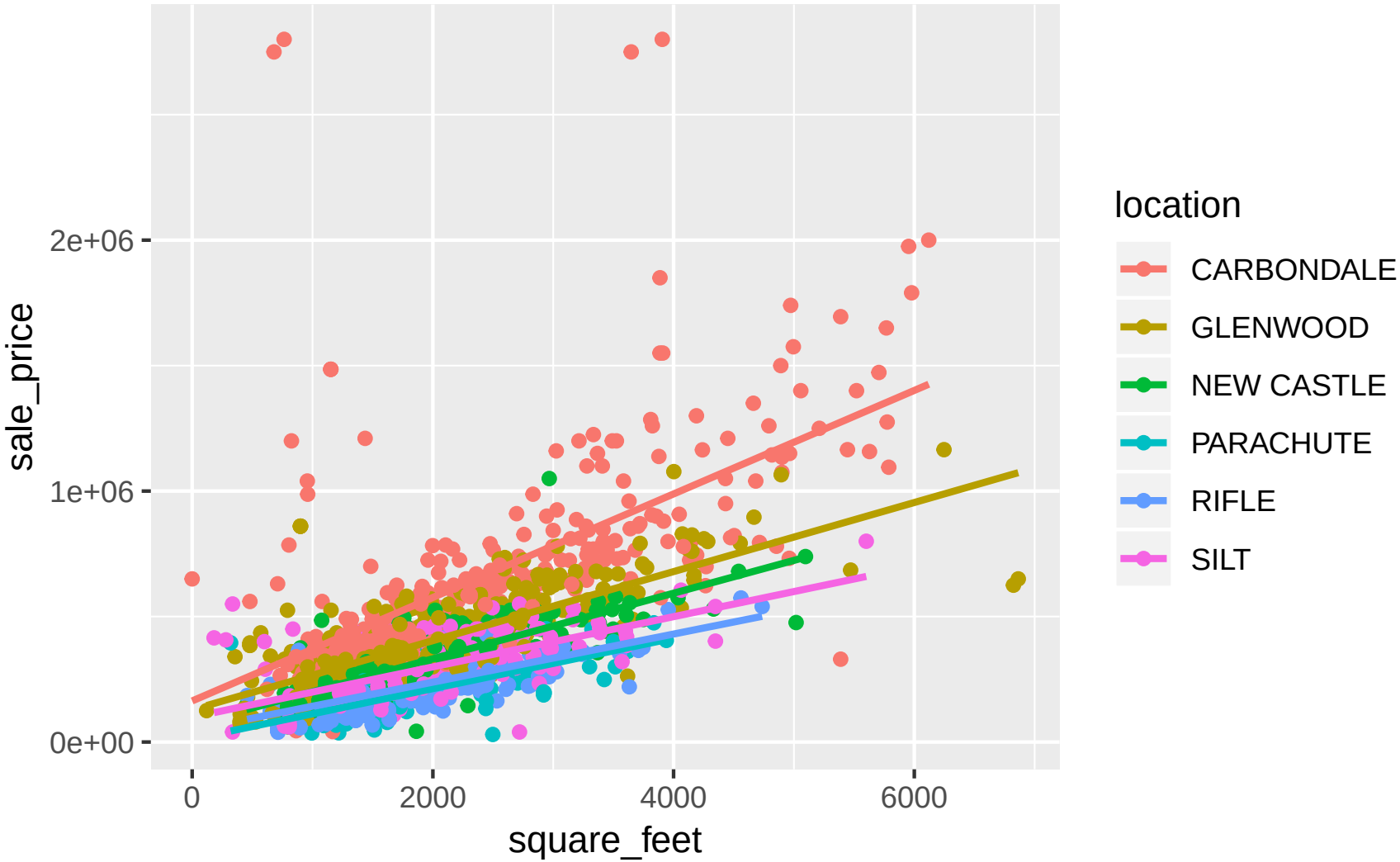
Linear model: sale price vs. square feet

```
ggplot(data = home_sales, aes(x = square_feet, y = sale_pri
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to lm
```



Linear model with errors removed

```
ggplot(data = home_sales_fix, aes(x = square_feet, y = sale
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to l
```



Linear model facetted by location

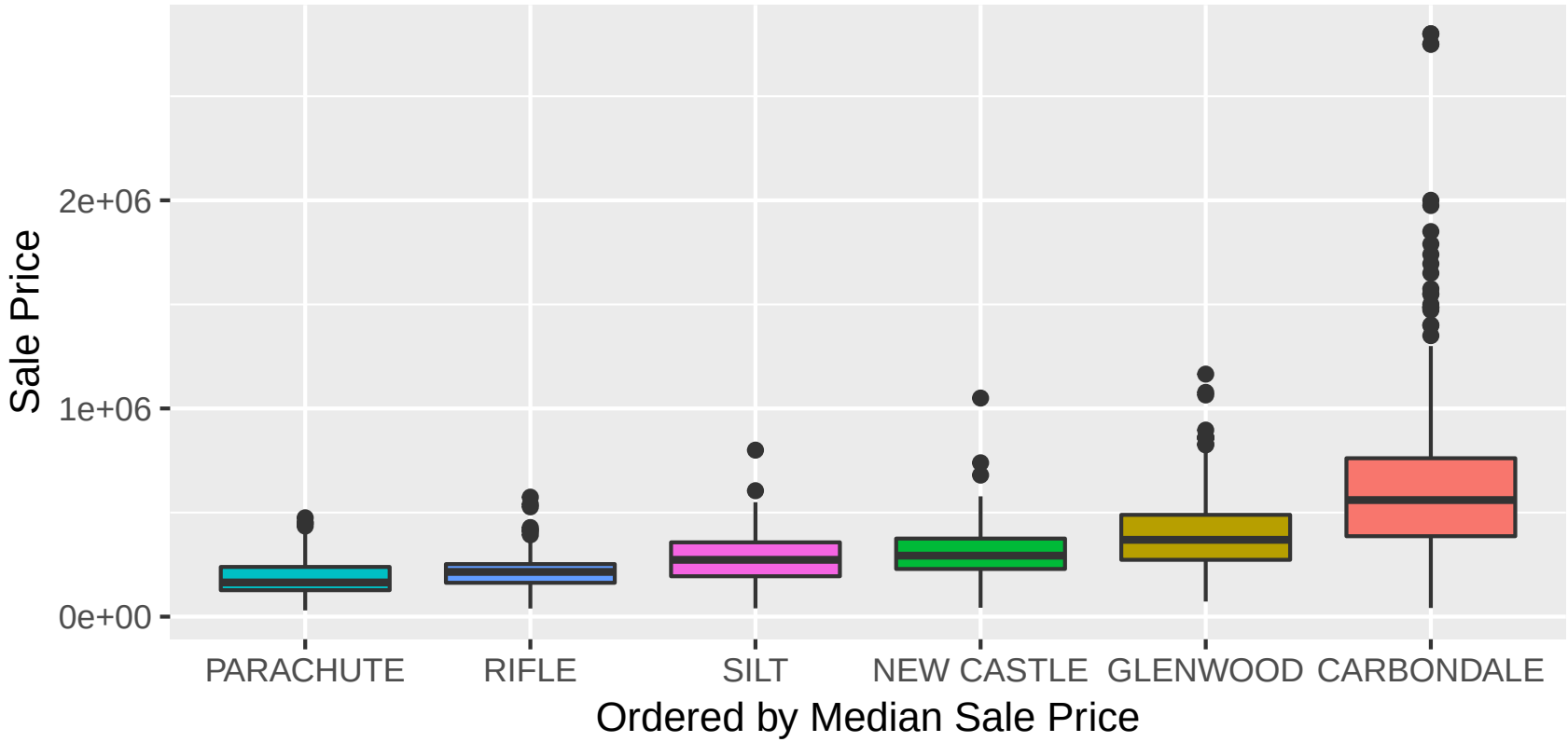
```
ggplot(home_sales, aes(square_feet, sale_price, color = loc
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~ location)
```

Linear model facetted by location

```
ggplot(home_sales_fix, aes(square_feet, sale_price, color =
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~ location)
```

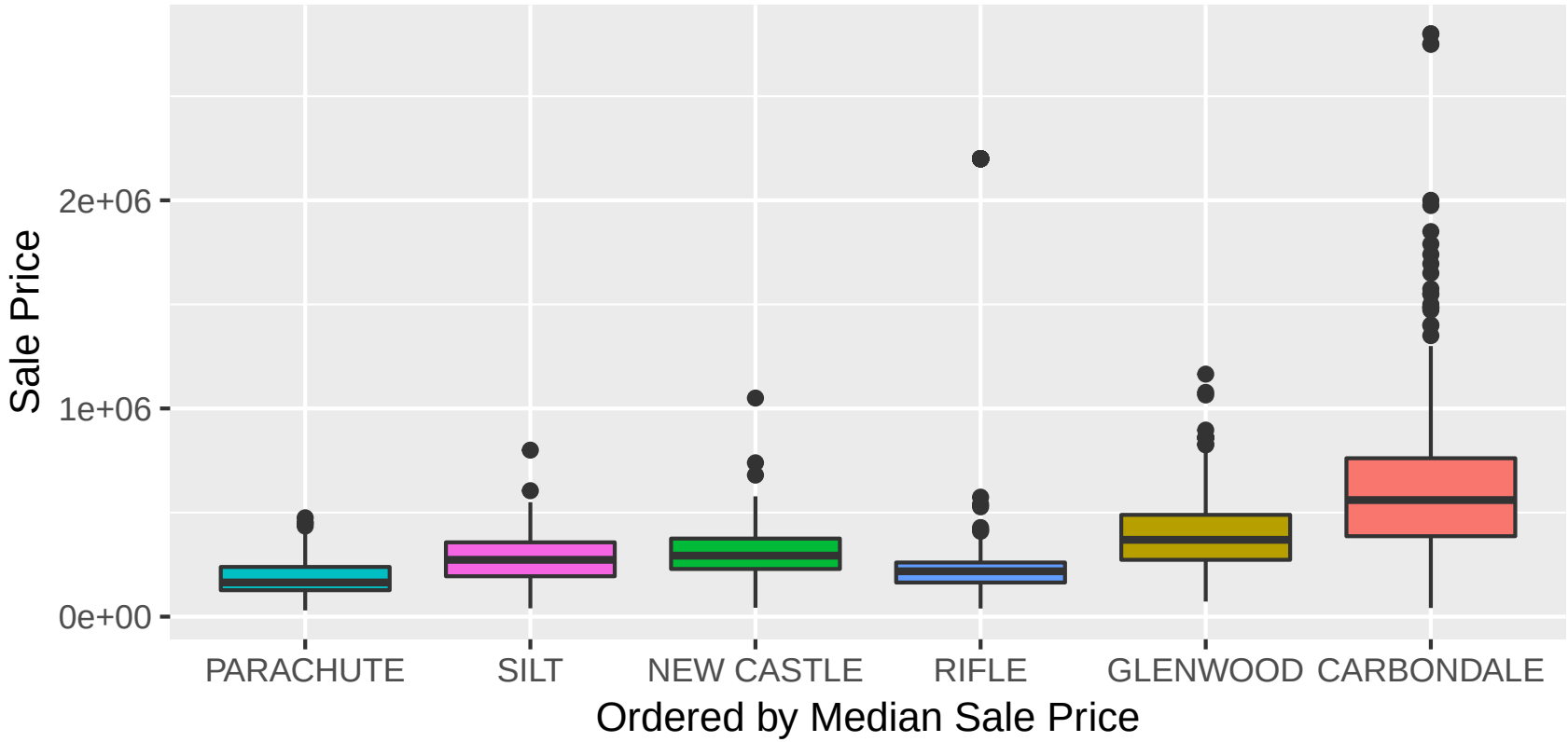
Boxplots ordered by median sale price

```
ggplot(home_sales_fix,
       aes(x = reorder(location, sale_price, fun = median),
           y = sale_price, fill = location)) +
  geom_boxplot() + theme(legend.position = "bottom") +
  labs(x = "Ordered by Median Sale Price", y = "Sale Price"
```



Boxplots ordered by median sale price

```
ggplot(home_sales,
       aes(x = reorder(location, sale_price, fun = median),
           y = sale_price, fill = location)) +
  geom_boxplot() + theme(legend.position = "bottom") +
  labs(x = "Ordered by Median Sale Price", y = "Sale Price")
```



Tidying wide datasets

```
seattle_rain # Display Monthly Rain Gauge Accumulations fo
```

```
## # A tibble: 175 x 18
##    Date   RG01  RG02  RG03  RG04  RG05  RG07  RG08  RG0
##    <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 11/3~  2.43  3.36 2.88   2.48 0.78   2.49 2.57   2.93
##  2 12/3~  4.31  1.4  5.46   4.8  1.99   5.06 2.48   2.35
##  3 01/3~  6.55  7.35 5.84   6.48 7.57   4.47 7.39   7.31
##  4 02/2~  1.61  1.81 1.7    1.49 1.11   1.5  1.56   1.73
##  5 03/3~  5.01  5.88 3.12   5.01 5.09   5.15 5.14   5.01
##  6 04/3~  2.27  3.15 2.69   2.56 2.2    2.49 2.5    1.45
##  7 05/3~  0.91  1.49 1.51   1.4  0.43   1.59 0.98   0.93
##  8 06/3~  0.49  0.89 0.4    0.34 0.570  0.94 0.75   0.79
##  9 07/3~  0.12  0.18 0.16   0.51 0.17   0.89 0.21   0.06
## 10 08/3~  0.33  0.46 0.290  0.26 0.37   1.33 0.570  0.22
## # ... with 165 more rows, and 7 more variables: RG12 <db
## #   RG15 <dbl>, RG16 <dbl>, RG17 <dbl>, RG18 <dbl>, RG20
```

gather columns RG01 through RG20_25

```
library(tidyr)
seattle_rain_tall <- seattle_rain %>%
  gather(RG01:RG20_25, key = rain_gauge, value = precip_inc
seattle_rain_tall
```

```
## # A tibble: 2,975 x 3
##    Date       rain_gauge precip_inches
##    <chr>      <chr>                <dbl>
##  1 11/30/2002 RG01                  2.43
##  2 12/31/2002 RG01                  4.31
##  3 01/31/2003 RG01                  6.55
##  4 02/28/2003 RG01                  1.61
##  5 03/31/2003 RG01                  5.01
##  6 04/30/2003 RG01                  2.27
##  7 05/31/2003 RG01                  0.91
##  8 06/30/2003 RG01                  0.49
##  9 07/31/2003 RG01                  0.12
## 10 08/31/2003 RG01                  0.33
## # ... with 2,965 more rows
```

gather all columns *except* Date

```
library(tidyr)
seattle_rain_tall <- seattle_rain %>%
  gather(key = rain_gauge, value = precip_inches, -Date) #
seattle_rain_tall
```

```
## # A tibble: 2,975 x 3
##    Date       rain_gauge precip_inches
##    <chr>      <chr>               <dbl>
##  1 11/30/2002 RG01                 2.43
##  2 12/31/2002 RG01                 4.31
##  3 01/31/2003 RG01                 6.55
##  4 02/28/2003 RG01                 1.61
##  5 03/31/2003 RG01                 5.01
##  6 04/30/2003 RG01                 2.27
##  7 05/31/2003 RG01                 0.91
##  8 06/30/2003 RG01                 0.49
##  9 07/31/2003 RG01                 0.12
## 10 08/31/2003 RG01                 0.33
## # ... with 2,965 more rows
```

Gathering in SQL

```
SELECT date, 'rg01' as rain_gauge, rg01 as precip_inches FR
SELECT date, 'rg02' as rain_gauge, rg02 as precip_inches FR
SELECT date, 'rg03' as rain_gauge, rg03 as precip_inches FR
SELECT date, 'rg04' as rain_gauge, rg04 as precip_inches FR
SELECT date, 'rg05' as rain_gauge, rg05 as precip_inches FR
SELECT date, 'rg07' as rain_gauge, rg07 as precip_inches FR
SELECT date, 'rg08' as rain_gauge, rg08 as precip_inches FR
SELECT date, 'rg09' as rain_gauge, rg09 as precip_inches FR
SELECT date, 'rg10_30' as rain_gauge, rg10_30 as precip_inc
SELECT date, 'rg11' as rain_gauge, rg11 as precip_inches FR
SELECT date, 'rg12' as rain_gauge, rg12 as precip_inches FR
SELECT date, 'rg14' as rain_gauge, rg14 as precip_inches FR
SELECT date, 'rg15' as rain_gauge, rg15 as precip_inches FR
SELECT date, 'rg16' as rain_gauge, rg16 as precip_inches FR
SELECT date, 'rg17' as rain_gauge, rg17 as precip_inches FR
SELECT date, 'rg18' as rain_gauge, rg18 as precip_inches FR
SELECT date, 'rg20_25' as rain_gauge, rg20_25 as precip_inc
ORDER BY rain_gauge, date;
```

Part 2: Many models with `purrr` and `broom`

Linear model for Carbondale

```r
carbondale <- home_sales_fix %>%
  filter(location == "CARBONDALE")
model <- lm(sale_price ~ square_feet, data = carbondale)
summary(model)
```

```
##
## Call:
## lm(formula = sale_price ~ square_feet, data = carbondale
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -944784 -117066  -61120   10307 2478788
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 163668.19   34376.15   4.761 2.75e-06 ***
## square_feet    206.21      12.97  15.900  < 2e-16 ***
## ---
```

Transform model into data frame with broom

```
library(broom)
tidy(model)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  163668.   34376.       4.76 2.75e- 6
## 2 square_feet     206.      13.0     15.9  6.66e-44
```

bind_rows to combine models

```
carbondale <- home_sales_fix %>%
  filter(location == "CARBONDALE")
model1 <- lm(sale_price ~ square_feet, data = carbondale)
glenwood <- home_sales_fix %>%
  filter(location == "GLENWOOD")
model2 <- lm(sale_price ~ square_feet, data = glenwood)
bind_rows(tidy(model1), tidy(model2))
```

```
## # A tibble: 4 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  163668.   34376.        4.76 2.75e- 6
## 2 square_feet     206.      13.0      15.9  6.66e-44
## 3 (Intercept)  128578.   11400.       11.3  4.42e-26
## 4 square_feet     138.       5.36     25.7  2.24e-89
```

Step 1: nest by location

```
home_sales_fix_models <- home_sales_fix %>%
  nest(-location)
home_sales_fix_models
```

```
## # A tibble: 6 x 2
##   location   data
##   <chr>      <list>
## 1 CARBONDALE <tibble [379 x 13]>
## 2 GLENWOOD   <tibble [441 x 13]>
## 3 NEW CASTLE <tibble [327 x 13]>
## 4 PARACHUTE  <tibble [202 x 13]>
## 5 RIFLE      <tibble [370 x 13]>
## 6 SILT       <tibble [225 x 13]>
```

Examine the data for New Castle

```
home_sales_fix_models$data[[3]]
```

```
## # A tibble: 327 x 13
##    account parcel_number reception sale_date sale_price
##    <chr>   <chr>         <chr>     <chr>           <dbl>
##  1 R044215 212331109026  860557    3/16/2015      145000
##  2 R380139 212331107012  868155    9/16/2015      195000
##  3 R043923 212331110031  866949    8/14/2015      162000
##  4 R380402 212331226003  875582    4/4/2016       205000
##  5 R040070 212325300002  872566    1/13/2016      375000
##  6 R380215 212331223016  861672    4/17/2015      176900
##  7 R015027 218106400058  858833    1/30/2015      255000
##  8 R170410 218332101001  872474    1/13/2016      485000
##  9 R005504 212330320001  866365    8/4/2015       268500
## 10 R005511 212330320008  878740    6/23/2016      299000
## # ... with 317 more rows, and 7 more variables: architec
## #   year_built <dbl>, bedrooms <dbl>, baths <dbl>, squar
## #   legal <chr>, classification <chr>
```

Unnesting returns to original

```
unnest(home_sales_fix_models)
```

```
## # A tibble: 1,944 x 14
##    location account parcel_number reception sale_date sa
##    <chr>    <chr>   <chr>         <chr>     <chr>
##  1 CARBOND~ R340967 239334401005  879240    6/29/2016
##  2 CARBOND~ R340073 239334200010  870778    11/24/20~
##  3 CARBOND~ R112063 239335100057  869383    10/13/20~
##  4 CARBOND~ R580140 239334366004  857328    12/18/20~
##  5 CARBOND~ R043949 239120300276  853541    9/12/2014
##  6 CARBOND~ R011301 239120300057  868354    9/21/2015
##  7 CARBOND~ R011415 239325100148  861479    4/10/2015
##  8 CARBOND~ R005930 246304125013  875431    3/24/2016
##  9 CARBOND~ R040419 246303100026  865422    7/13/2015
## 10 CARBOND~ R041666 239334268001  868224    9/15/2015
## # ... with 1,934 more rows, and 8 more variables: situs_
## #   architectural_style <chr>, year_built <dbl>, bedroom
## #   baths <dbl>, square_feet <dbl>, legal <chr>, classif
```

Step 2: map() to fit lm to each dataset

```
library(purrr)
home_sales_fix_models <- home_sales_fix %>%
  nest(-location) %>%
  mutate(models = map(data, ~lm(sale_price ~ square_feet,
  # data has been passed into lm through map function
  # dot "." is used for data in the lm call
home_sales_fix_models
```

```
## # A tibble: 6 x 3
##   location   data                models
##   <chr>      <list>              <list>
## 1 CARBONDALE <tibble [379 x 13]> <S3: lm>
## 2 GLENWOOD   <tibble [441 x 13]> <S3: lm>
## 3 NEW CASTLE <tibble [327 x 13]> <S3: lm>
## 4 PARACHUTE  <tibble [202 x 13]> <S3: lm>
## 5 RIFLE      <tibble [370 x 13]> <S3: lm>
## 6 SILT       <tibble [225 x 13]> <S3: lm>
```

Examine the model for New Castle

```
home_sales_fix_models$models[[3]]
```

```
##
## Call:
## lm(formula = sale_price ~ square_feet, data = .)
##
## Coefficients:
## (Intercept)  square_feet
##     70045.8        130.7
```

Step 3: Use map() to tidy each model

```
home_sales_fix_models <- home_sales_fix %>%
  nest(-location) %>%
  mutate(models = map(data, ~lm(sale_price ~ square_feet,
  mutate(tidied = map(models, tidy))
home_sales_fix_models
```

```
## # A tibble: 6 x 4
##   location   data                 models   tidied
##   <chr>      <list>               <list>   <list>
## 1 CARBONDALE <tibble [379 x 13]> <S3: lm> <tibble [2 x 5
## 2 GLENWOOD   <tibble [441 x 13]> <S3: lm> <tibble [2 x 5
## 3 NEW CASTLE <tibble [327 x 13]> <S3: lm> <tibble [2 x 5
## 4 PARACHUTE  <tibble [202 x 13]> <S3: lm> <tibble [2 x 5
## 5 RIFLE      <tibble [370 x 13]> <S3: lm> <tibble [2 x 5
## 6 SILT       <tibble [225 x 13]> <S3: lm> <tibble [2 x 5
```

Examine tidy model for New Castle

```
home_sales_fix_models$tidied[[3]]
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   70046.  11041.         6.34 7.49e-10
## 2 square_feet     131.      5.68      23.0  3.65e-70
```

Step 4: unnest to tidy table of coefficients

```
location_coeffs <- home_sales_fix %>%
  nest(-location) %>%
  mutate(models = map(data, ~lm(sale_price ~ square_feet,
  mutate(tidied = map(models, tidy)) %>%
  unnest(tidied)
location_coeffs
```

```
## # A tibble: 12 x 6
##    location   term        estimate std.error statistic
##    <chr>      <chr>          <dbl>     <dbl>     <dbl>
##  1 CARBONDALE (Intercept) 163668.   34376.       4.76
##  2 CARBONDALE square_feet    206.      13.0     15.9
##  3 GLENWOOD   (Intercept) 128578.   11400.      11.3
##  4 GLENWOOD   square_feet    138.       5.36    25.7
##  5 NEW CASTLE (Intercept)  70046.   11041.       6.34
##  6 NEW CASTLE square_feet    131.       5.68    23.0
##  7 PARACHUTE  (Intercept)  11651.   14526.       0.802
##  8 PARACHUTE  square_feet    101.       8.07    12.5
##  9 RIFLE      (Intercept)  46757.    8363.       5.59
```