# Many Models with tidyverse tools in R

*Casey Bates*

*2/7/2019*

## Motivation

Utilize `tidyverse` tools and the `broom` package to fit and tidy 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.

## Import and process the datasets

```
# This chunk includes the data processing steps to wrangle and clean the data
library(tidyverse) # For reading-in, wrangling, visualizing the data, and for pipe operator
library(readxl) # Tidyverse package for reading in Excel files
library(magrittr) # For the '%<>%' pipe operator

# Read-in townhomes and condo sales data
townhomes <- read_xlsx("2017-comparable-sales-condos-townhomes.xlsx")

# Read-in single family home sales data
single_family <- read_xlsx("2017-comparable-sales-single-family.xlsx")

# Check column names and formats
glimpse(townhomes)
```

```
## Observations: 606
## Variables: 14
## $ Account               <chr> "R007341", "R340686", "R045341", "R34080...
## $ `Parcel Number`       <chr> "239334430001", "239334350003", "2393343...
## $ Reception             <dbl> 852413, 871451, 861415, 872689, 875851, ...
## $ `Sale Date`           <chr> "8/8/2014", "12/11/2015", "4/14/2015", "...
## $ `Sale Price`          <dbl> 210000, 265000, 308500, 160000, 340000, ...
## $ `Situs Address`       <chr> "000133 SOPRIS AVE #A", "213 1/2 N 10TH ...
## $ Location              <chr> "CARBONDALE", "CARBONDALE", "CARBONDALE"...
## $ Classification        <chr> "Condo", "Condo", "Condo", "Condo", "Con...
## $ `Architectural Style` <chr> "CONDO", "CONDO", "CONDO", "CONDO", "CON...
## $ `Actual Year Built`   <dbl> 1974, 1980, 2008, 1981, 2007, 2007, 2007...
## $ Bedrooms              <dbl> 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2...
## $ Baths                 <dbl> 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 2.00...
## $ `Heated Area`         <dbl> 621, 732, 792, 800, 864, 864, 891, 918, ...
## $ Legal                 <chr> "Section: 34 Township: 7 Range: 88 DESC:...
```

```
glimpse(single_family)
```

```
## Observations: 1,369
## Variables: 13
## $ Account               <chr> "R340967", "R340073", "R112063", "R58014...
## $ `Parcel Number`       <chr> "239334401005", "239334200010", "2393351...
```

```
## $ Reception             <chr> "879240", "870778", "869383", "857328", ...
## $ `Sale Date`           <chr> "6/29/2016", "11/24/2015", "10/13/2015",...
## $ `Sale Price`          <dbl> 650000, 560000, 2750000, 630500, 2800000...
## $ `Situs Address`       <chr> "000066 N 2ND ST", "000276 10TH ST", "00...
## $ Location              <chr> "CARBONDALE", "CARBONDALE", "CARBONDALE"...
## $ `Architectural Style` <chr> "ONE STORY", "ONE STORY", "ONE STORY", "...
## $ `Year Built`          <dbl> 1970, 1971, 2002, 1999, 2008, 1994, 1993...
## $ Bedrooms              <dbl> 0, 1, 0, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2...
## $ Baths                 <dbl> 0.00, 1.00, 0.75, 1.00, 1.00, 1.00, 1.00...
## $ `Heated Area`         <dbl> 0, 480, 680, 710, 764, 804, 825, 957, 96...
## $ Legal                 <chr> "Section: 34 Township: 7 Range: 88 Subdi...
```

```
# 'sale_price' column was coerced to numeric and special characters '$' and ',' removed

# Replace spaces in column names with underscores ('_') and make names lowercase
# the package 'magrittr' is used for the pipe operator, this is loaded with the 'tidyverse' library
colnames(townhomes) %<>% str_replace_all("\\s", "_") %<>% tolower()
colnames(single_family) %<>% str_replace_all("\\s", "_") %<>% tolower()

# Rename column "actual_year_built" as "year_built" for consistency between datasets
townhomes <- townhomes %>%
  rename(year_built = actual_year_built)

# Add a "classifcation" column to the single_family dataset and set all values to "Single Family"
single_family <- single_family %>%
  mutate(classification = "Single Family")

# bind_rows() throws an error b/c 'reception' column is numeric in townhomes dataset; convert to charac
townhomes$reception <- as.character(townhomes$reception)

# Combine the datasets into one
# This dataset contains errors which we will see later
home_sales_errors <- bind_rows(single_family, townhomes)
glimpse(home_sales_errors)
```

```
## Observations: 1,975
## Variables: 14
## $ account             <chr> "R340967", "R340073", "R112063", "R580140"...
## $ parcel_number       <chr> "239334401005", "239334200010", "239335100...
## $ reception           <chr> "879240", "870778", "869383", "857328", "8...
## $ sale_date           <chr> "6/29/2016", "11/24/2015", "10/13/2015", "...
## $ sale_price          <dbl> 650000, 560000, 2750000, 630500, 2800000, ...
## $ situs_address       <chr> "000066 N 2ND ST", "000276 10TH ST", "0008...
## $ location            <chr> "CARBONDALE", "CARBONDALE", "CARBONDALE", ...
## $ architectural_style <chr> "ONE STORY", "ONE STORY", "ONE STORY", "ON...
## $ year_built          <dbl> 1970, 1971, 2002, 1999, 2008, 1994, 1993, ...
## $ bedrooms            <dbl> 0, 1, 0, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, ...
## $ baths               <dbl> 0.00, 1.00, 0.75, 1.00, 1.00, 1.00, 1.00, ...
## $ heated_area         <dbl> 0, 480, 680, 710, 764, 804, 825, 957, 960,...
## $ legal               <chr> "Section: 34 Township: 7 Range: 88 Subdivi...
## $ classification      <chr> "Single Family", "Single Family", "Single ...
```

```
# No new columns were created during the 'bind_rows()' process, indicating all colums align
# Remove single_family and townhomes dataframes because they are no longer needed
rm(single_family, townhomes)
```

```r
# unique(home_sales_errors$classification) reveals a "Garage Only" type. Drop this.
home_sales_errors <- home_sales_errors %>% filter(classification != "Garage Only")

# Rename the column "heated area" to "square_feet" for clarity, though technically not correct
home_sales_errors <- home_sales_errors %>%
  rename(square_feet = heated_area)
```

## Outline

- Part 1: Explore the dataset with `ggplot2`
  - Tidy wide datasets with `tidyr`
- Part 2: Fit and tidy many models with `purrr` and `broom` using:
  - `broom::tidy`
  - `broom::augment`
  - `broom::glance`

## `tidyverse` packages used

### Importing

- `readr`; `readxl`

### Wrangle

- `dplyr`; `tidyr`; `stringr`; `tibble`

### Visualize

- `ggplot2`

### Program

- `purrr`; `magrittr`

### Model

- `broom`; `modelr`

## 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_errors)
```

```
## Observations: 1,967
## Variables: 14
## $ account             <chr> "R340967", "R340073", "R112063", "R580140"...
## $ parcel_number       <chr> "239334401005", "239334200010", "239335100...
## $ reception           <chr> "879240", "870778", "869383", "857328", "8...
## $ sale_date           <chr> "6/29/2016", "11/24/2015", "10/13/2015", "...
## $ sale_price          <dbl> 650000, 560000, 2750000, 630500, 2800000, ...
## $ situs_address       <chr> "000066 N 2ND ST", "000276 10TH ST", "0008...
## $ location            <chr> "CARBONDALE", "CARBONDALE", "CARBONDALE", ...
## $ architectural_style <chr> "ONE STORY", "ONE STORY", "ONE STORY", "ON...
## $ year_built          <dbl> 1970, 1971, 2002, 1999, 2008, 1994, 1993, ...
## $ bedrooms            <dbl> 0, 1, 0, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 2, ...
## $ baths               <dbl> 0.00, 1.00, 0.75, 1.00, 1.00, 1.00, 1.00, ...
## $ square_feet         <dbl> 0, 480, 680, 710, 764, 804, 825, 957, 960,...
## $ legal               <chr> "Section: 34 Township: 7 Range: 88 Subdivi...
## $ classification      <chr> "Single Family", "Single Family", "Single ...
```

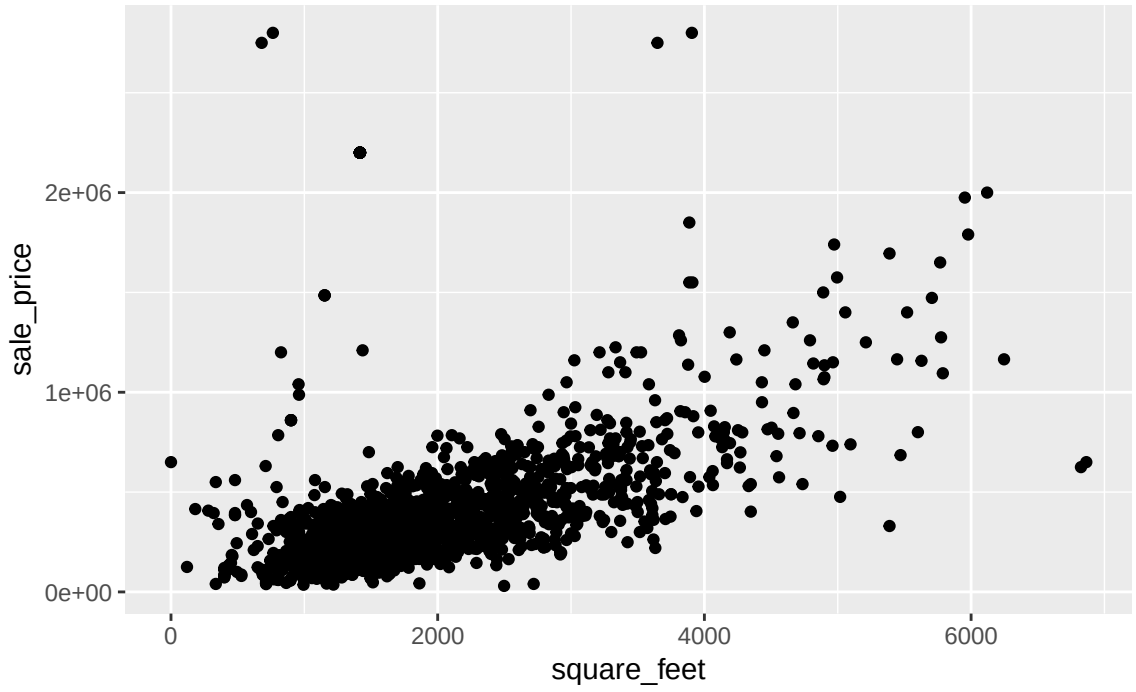
## ggplot2 package in R

- 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

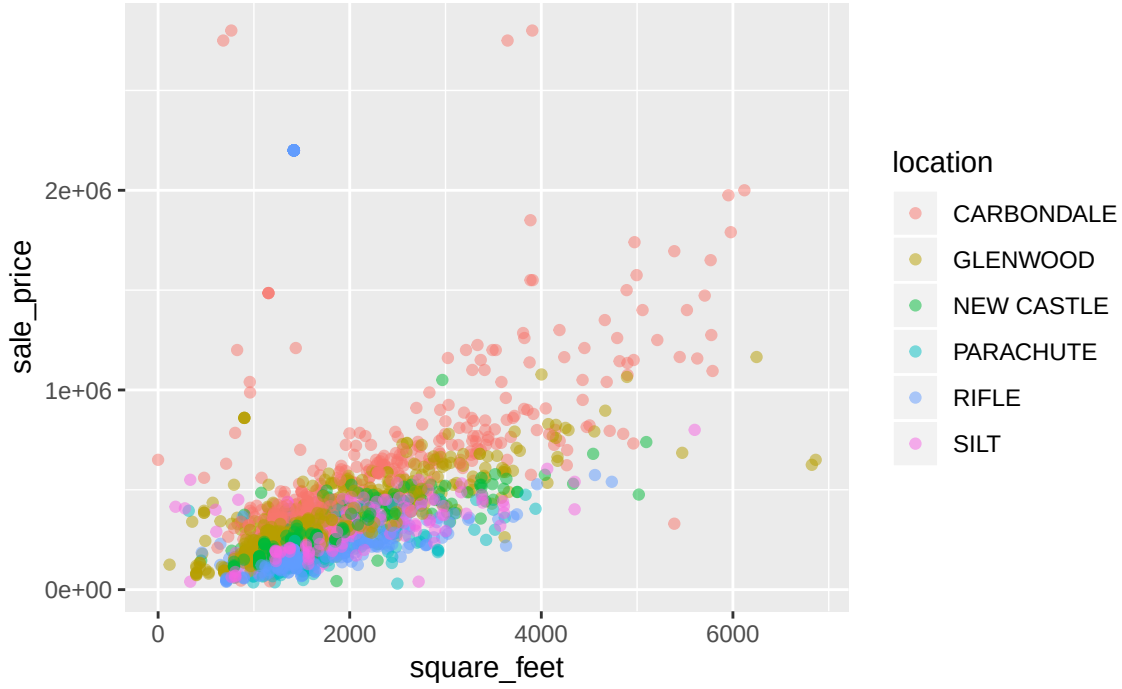
```
ggplot(data = home_sales_errors, aes(x = square_feet, y = sale_price)) +
  geom_point()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



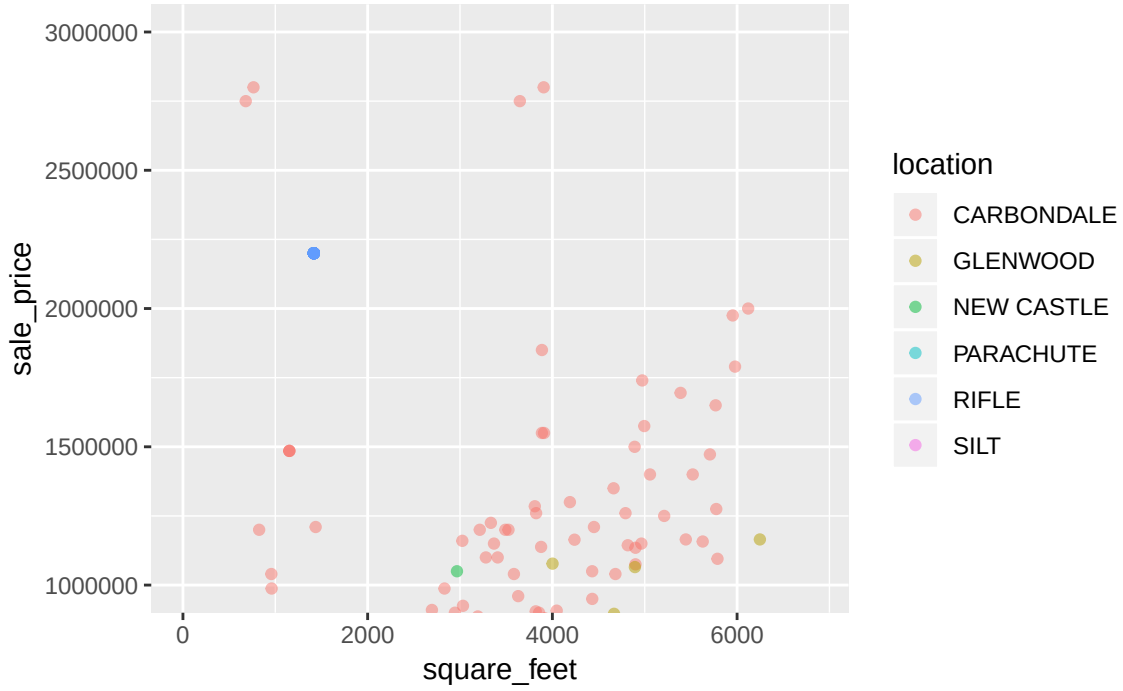

### Transparency and color by location

```
ggplot(data = home_sales_errors,
       aes(x = square_feet, y = sale_price, color = location)) +
  geom_point(alpha = 0.5)
```



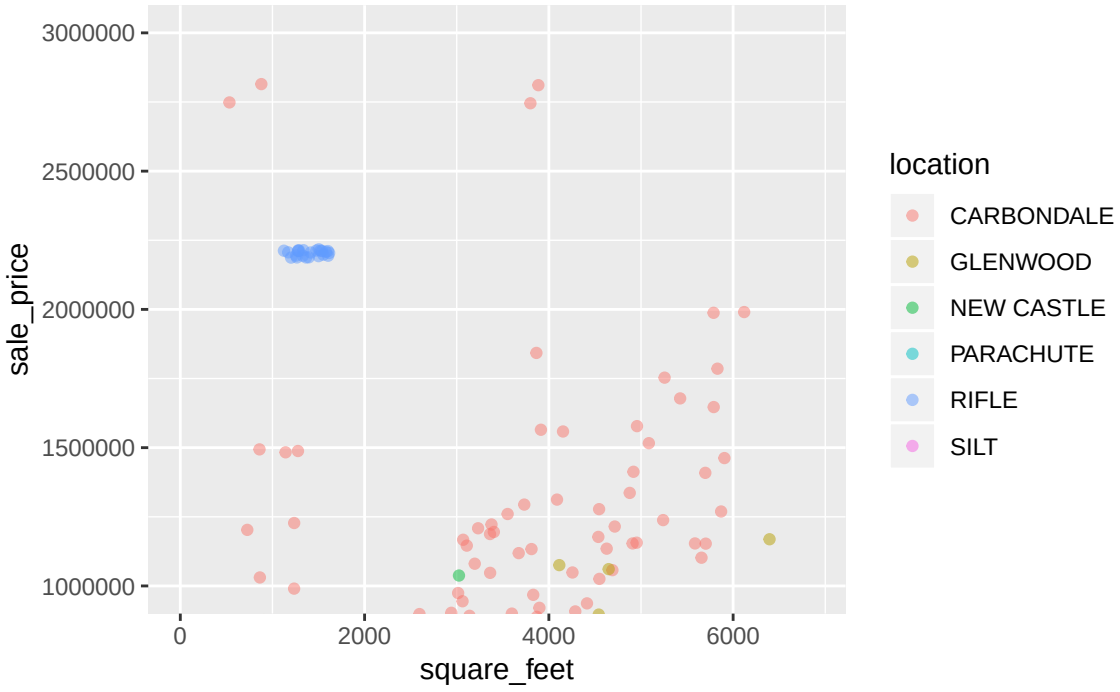
### Zooming into sales above $1M

```
ggplot(data = home_sales_errors,
       aes(x = square_feet, y = sale_price, color = location)) +
  geom_point(alpha = 0.5) +
  coord_cartesian(ylim = c(1000000, 3000000))
```




### Add random noise with jitter

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




### High sale price observations

```
home_sales_errors %>% arrange(desc(sale_price)) %>% select(c("sale_price", "location", "classification"
```

```
## # A tibble: 30 x 5
##    sale_price location   classification bedrooms square_feet
##         <dbl> <chr>      <chr>             <dbl>       <dbl>
##  1    2800000 CARBONDALE Single Family         2         764
##  2    2800000 CARBONDALE Single Family         2        3906
##  3    2750000 CARBONDALE Single Family         0         680
##  4    2750000 CARBONDALE Single Family         3        3648
##  5    2200000 RIFLE      Townhome              3        1417
##  6    2200000 RIFLE      Townhome              3        1417
##  7    2200000 RIFLE      Townhome              3        1417
##  8    2200000 RIFLE      Townhome              3        1417
##  9    2200000 RIFLE      Townhome              3        1417
## 10    2200000 RIFLE      Townhome              3        1417
## # ... with 20 more rows
```

### Remove erroneous high sale price observations

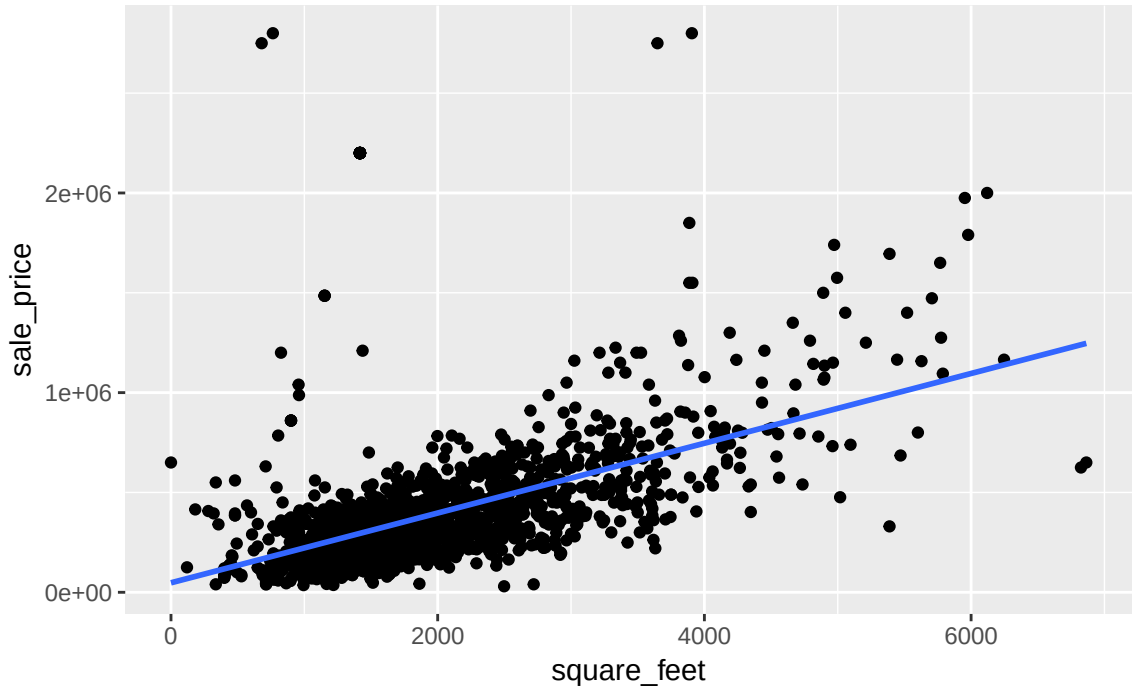
```
home_sales_fixed <- home_sales_errors %>% filter(location != "RIFLE" | classification != "Townhome" | sa
home_sales_fixed %>% filter(location == "RIFLE" & classification == "Townhome") %>% arrange(desc(sale_pr
```

```
## # A tibble: 10 x 14
##    account parcel_number reception sale_date sale_price situs_address
##    <chr>   <chr>         <chr>     <chr>          <dbl> <chr>
##  1 R009182 217710207012  873859    2/17/2016     285000 001471 FIR CT
##  2 R009185 217710207015  863141    5/22/2015     249000 001485 FIR CT
```

```
##  3 R083245 217704359004  877783    5/27/2016     199500 846 W 24TH ST
##  4 R083237 217704358001  878258    6/10/2016     199000 820 W 24TH ST
##  5 R083241 217704358005  876194    4/20/2016     199000 828 W 24TH ST
##  6 R083242 217704359001  877203    5/13/2016     199000 840 W 24TH ST
##  7 R044597 217704351005  878910    6/24/2016     199000 000718 W 24T~
##  8 R009189 217710207019  862189    4/30/2015     196500 001498 FIR CT
##  9 R083243 217704359002  878983    6/28/2016     195000 842 W 24TH ST
## 10 R083244 217704359003  878120    6/7/2016      195000 844 W 24TH ST
## # ... with 8 more variables: location <chr>, architectural_style <chr>,
## #   year_built <dbl>, bedrooms <dbl>, baths <dbl>, square_feet <dbl>,
## #   legal <chr>, classification <chr>
```

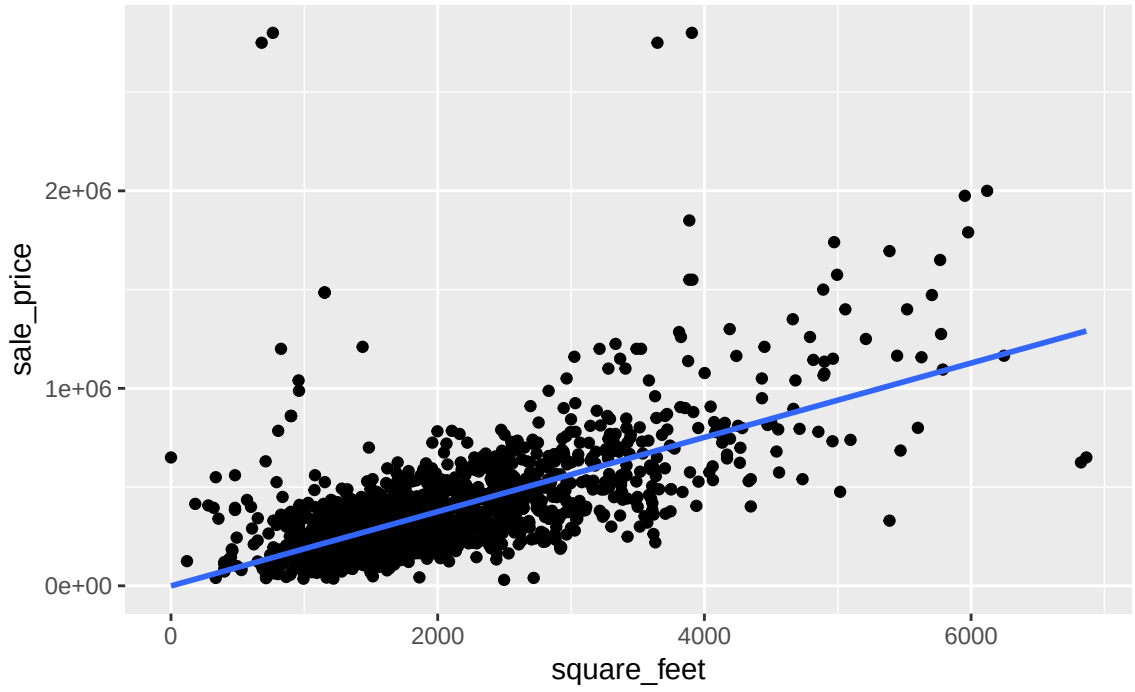
### Linear model: sale price vs. square ft.

```
ggplot(data = home_sales_errors, aes(x = square_feet, y = sale_price)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to lm for the linear model
```



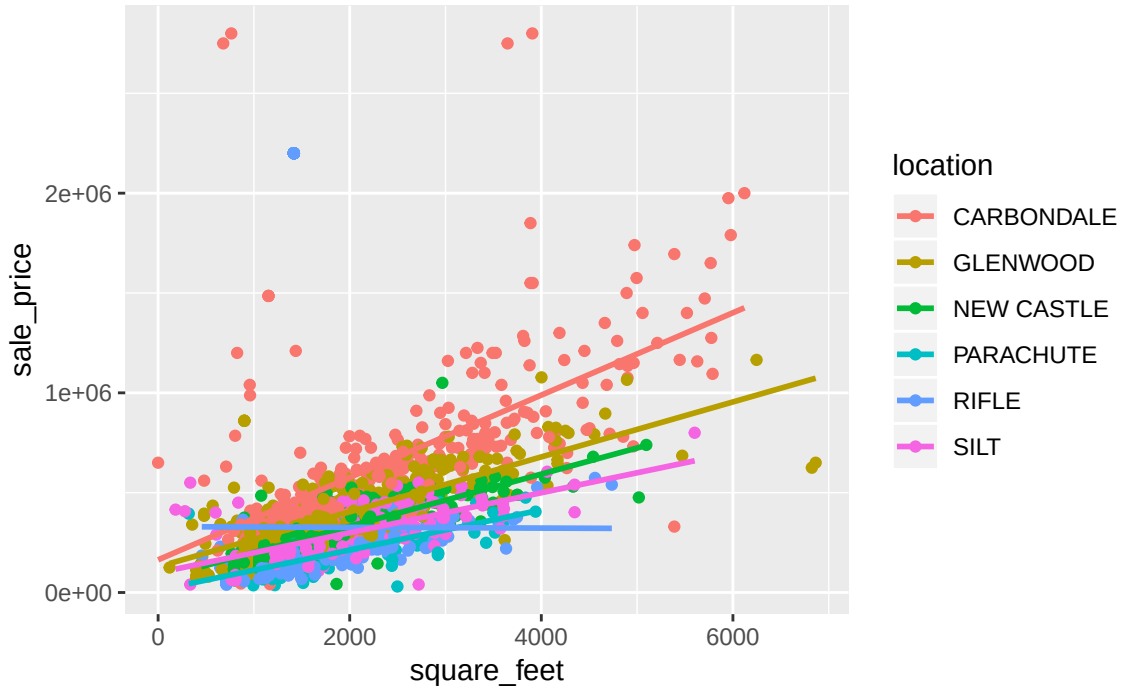

### Linear model: sale price vs. square ft.

```
ggplot(data = home_sales_fixed, aes(x = square_feet, y = sale_price)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to lm for the linear model
```



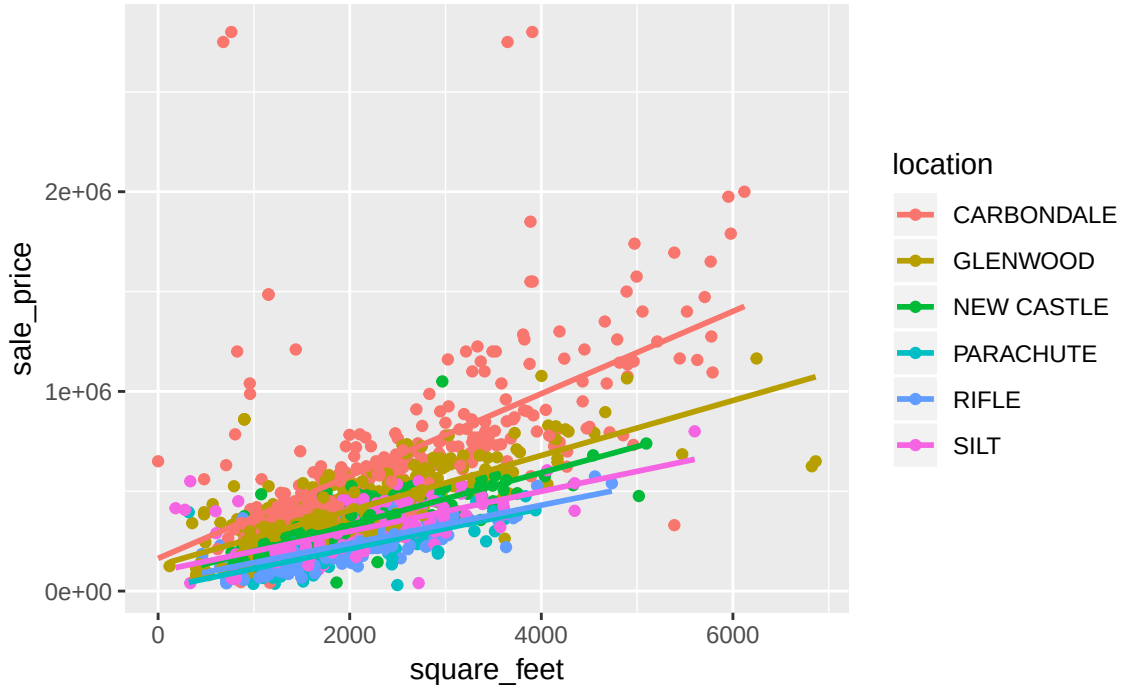

### Linear model: sale price vs. square feet

```
ggplot(data = home_sales_errors,
       aes(x = square_feet, y = sale_price, color = location)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to lm for the linear model
```



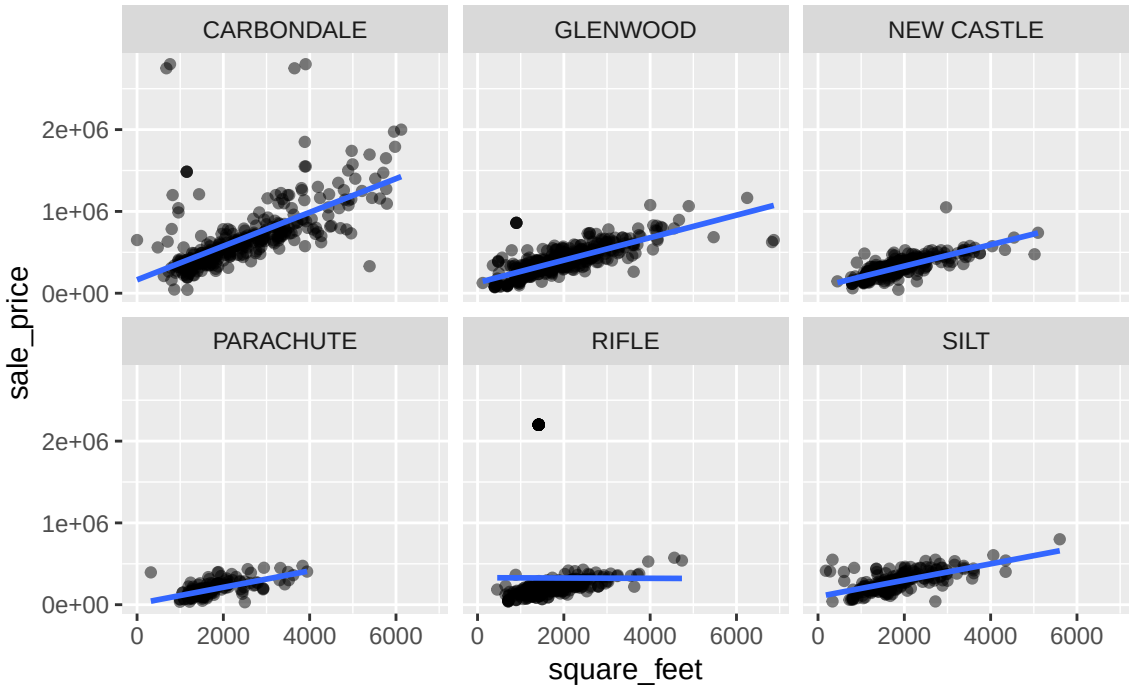
### Linear model with errors removed

```
ggplot(data = home_sales_fixed,
       aes(x = square_feet, y = sale_price, color = location)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) # Method set to lm for the linear model
```



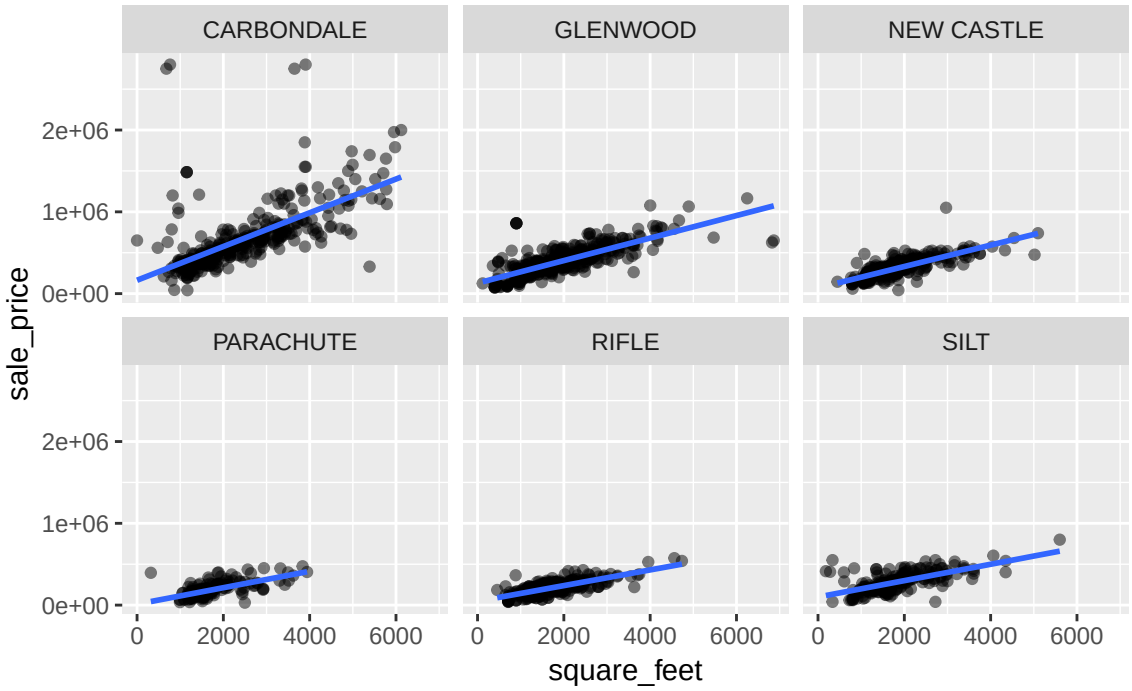

### Linear model facetted by location

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



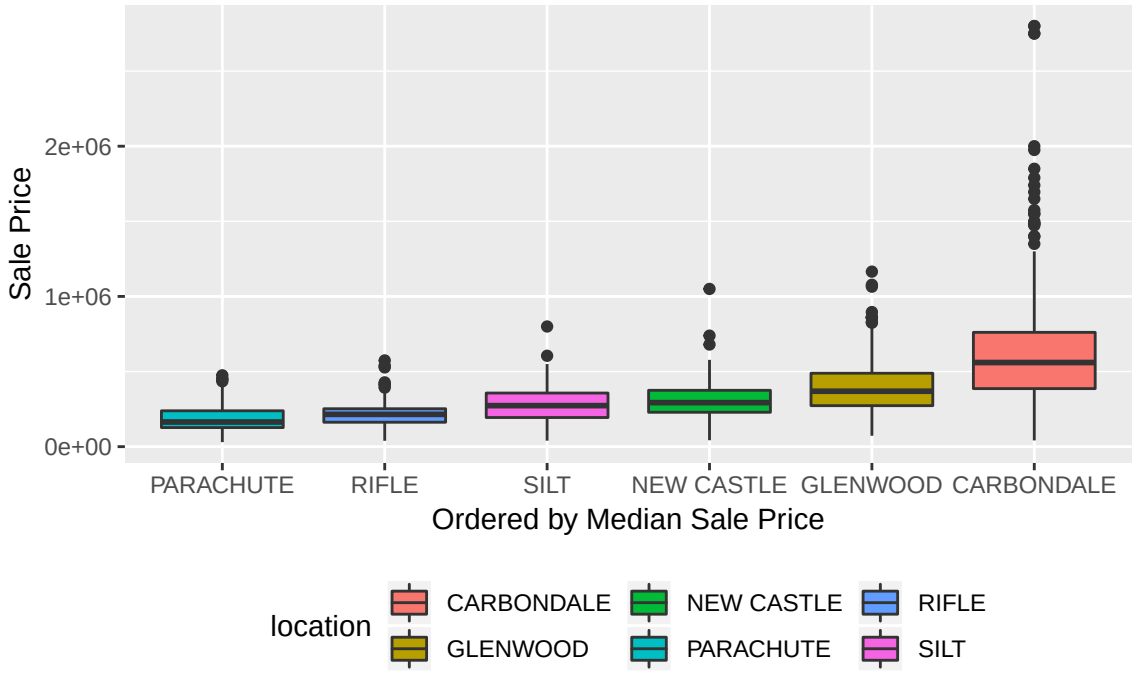

### Linear model facetted by location

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



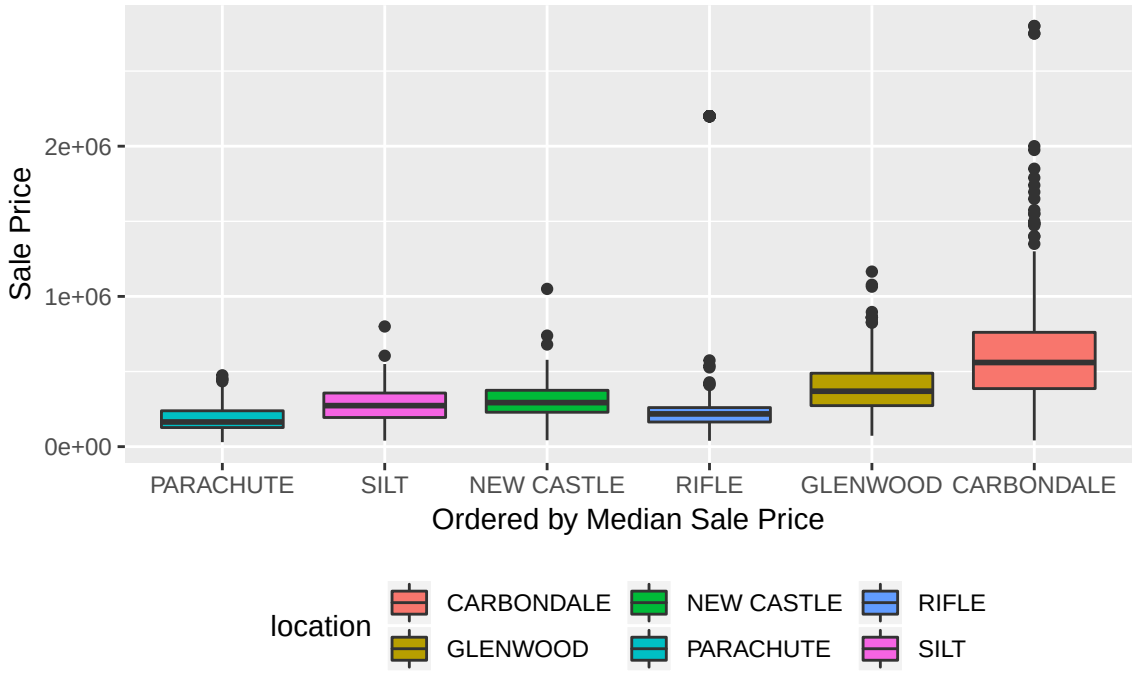
### Boxplots ordered by median sale price

```
ggplot(home_sales_fixed,
       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_errors,
       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")
```




### Import seattle rain gauge dataset

```
seattle_rain <- read_csv("Observed_Monthly_Rain_Gauge_Accumulations_Oct_2002_to_May_2017.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   RG01 = col_double(),
##   RG02 = col_double(),
##   RG03 = col_double(),
##   RG04 = col_double(),
##   RG05 = col_double(),
##   RG07 = col_double(),
##   RG08 = col_double(),
##   RG09 = col_double(),
##   RG10_30 = col_double(),
##   RG11 = col_double(),
##   RG12 = col_double(),
##   RG14 = col_double(),
##   RG15 = col_double(),
##   RG16 = col_double(),
##   RG17 = col_double(),
##   RG18 = col_double(),
##   RG20_25 = col_double()
## )
```

### Tidying wide datasets

```
seattle_rain # Display Monthly Rain Gauge Accumulations for Seattle
```

```
## # A tibble: 175 x 18
##    Date   RG01  RG02  RG03  RG04  RG05  RG07  RG08  RG09 RG10_30  RG11
##    <chr> <dbl> <dbl> <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    3.25  2.38
##  2 12/3~  4.31  1.4  5.46   4.8  1.99   5.06 2.48   2.35    6.48  4.95
##  3 01/3~  6.55  7.35 5.84   6.48 7.57   4.47 7.39   7.31    5.42  6.58
##  4 02/2~  1.61  1.81 1.7    1.49 1.11   1.5  1.56   1.73    1.18  1.37
##  5 03/3~  5.01  5.88 3.12   5.01 5.09   5.15 5.14   5.01    5.68  4.01
##  6 04/3~  2.27  3.15 2.69   2.56 2.2    2.49 2.5    1.45    1.78  2.34
##  7 05/3~  0.91  1.49 1.51   1.4  0.43   1.59 0.98   0.93    0.83  1.45
##  8 06/3~  0.49  0.89 0.4    0.34 0.570  0.94 0.75   0.79    0.33  0.82
##  9 07/3~  0.12  0.18 0.16   0.51 0.17   0.89 0.21   0.06    0.14  0.2
## 10 08/3~  0.33  0.46 0.290  0.26 0.37   1.33 0.570  0.22    0.68  0.3
## # ... with 165 more rows, and 7 more variables: RG12 <dbl>, RG14 <dbl>,
## #   RG15 <dbl>, RG16 <dbl>, RG17 <dbl>, RG18 <dbl>, RG20_25 <dbl>
```

### gather columns RG01 through RG20_25

```
library(tidyr)
seattle_rain_tall <- seattle_rain %>%
  gather(RG01:RG20_25, key = rain_gauge, value = precip_inches)
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) # Same result as before!
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
```

### . . . or you could do it in SQL :)

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

## Part 2: Fit and tidy many models with `purrr` and `broom`

- `broom::tidy(model)`
  - Returns 1 row for each coefficient
  - Columns present info about variability or estimates
- `broom::augment(model, data)`
  - Returns 1 row for each row in the data
  - Adds residuals, influence statistics
- `broom::glance(model)`
  - Returns 1 row for each model
  - Each column represents a model summary (quality and/or complexity)

`broom` can be used with many built-in statistical functions and popular packages.

## Workflow with `broom`

`tidyr::nest()` %>% `purrr::map()` %>% `tidyr::unnest()`

### Typical 4-step process

1. `nest()` dataset by categorical variable

2. Fit models to nested lists with `map()`
3. Apply `broom::tidy`, `broom::augment`, and/or `broom::glance` to each nested model
4. `unnest()` to tidy dataframe

*The 4-step process can also be applied to other packages and functions such as `modelr::add_residuals`*

## Why `broom`?

**While model inputs usually require tidy inputs, such attention to detail doesn't carry over to model outputs. Outputs such as predictions and estimated coefficients aren't always tidy. This makes it more difficult to combine results from multiple models.** For example, in R, the default representation of model coefficients is not tidy because it does not have an explicit variable that records the variable name for each estimate, they are instead recorded as row names. In R, row names must be unique, so combining coefficients from many models (e.g., from bootstrap resamples, or subgroups) requires workarounds to avoid losing important information. **This knocks you out of the flow of analysis and makes it harder to combine the results from multiple models. I'm not currently aware of any packages that resolve this problem.**

Hadley Wickham

*Emphasis added by David Robinson, author of the `broom` package, in this post.*

## Typical linear model output

```
carbondale <- home_sales_fixed %>%
  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 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 309200 on 376 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.402,  Adjusted R-squared:  0.4005
## F-statistic: 252.8 on 1 and 376 DF,  p-value: < 2.2e-16
```

## Set number of significant digits for display in tibbles

```
# You cannot set trailing zeros
options(pillar.sigfig = 4)
getOption("pillar.sigfig")
```

```
## [1] 4
```

### Transform model output into tidy data frame with `broom::tidy()`

```
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.       4.761 2.752e- 6
## 2 square_feet    206.2     12.97    15.90  6.664e-44
```

### `bind_rows` to combine models

```
carbondale <- home_sales_fixed %>%
  filter(location == "CARBONDALE")
model1 <- lm(sale_price ~ square_feet, data = carbondale)
glenwood <- home_sales_fixed %>%
  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.        4.761 2.752e- 6
## 2 square_feet    206.2    12.97     15.90  6.664e-44
## 3 (Intercept) 128578.  11400.       11.28  4.421e-26
## 4 square_feet    137.6     5.363    25.66  2.235e-89
```

### Step 1: `nest` by location

```
home_sales_fixed_models <- home_sales_fixed %>%
  nest(-location)
home_sales_fixed_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]>
```

### Alternate syntax used by Wickham

```
home_sales_fixed %>%
  group_by(location) %>%
  nest()
```

```
## # 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_fixed_models$data[[3]]
```

```
## # A tibble: 327 x 13
##    account parcel_number reception sale_date sale_price situs_address
##    <chr>   <chr>         <chr>     <chr>          <dbl> <chr>
##  1 R044215 212331109026  860557    3/16/2015     145000 000176 N 4TH~
##  2 R380139 212331107012  868155    9/16/2015     195000 000146 N 2ND~
##  3 R043923 212331110031  866949    8/14/2015     162000 000161 N 4TH~
##  4 R380402 212331226003  875582    4/4/2016      205000 000222 N 7TH~
##  5 R040070 212325300002  872566    1/13/2016     375000 000570 137 C~
##  6 R380215 212331223016  861672    4/17/2015     176900 000640 W MAI~
##  7 R015027 218106400058  858833    1/30/2015     255000 005341 214 C~
##  8 R170410 218332101001  872474    1/13/2016     485000 008149 312 C~
##  9 R005504 212330320001  866365    8/4/2015      268500 000601 LARIA~
## 10 R005511 212330320008  878740    6/23/2016     299000 000623 LARIA~
## # ... with 317 more rows, and 7 more variables: architectural_style <chr>,
## #   year_built <dbl>, bedrooms <dbl>, baths <dbl>, square_feet <dbl>,
## #   legal <chr>, classification <chr>
```

### Unnesting returns to original

```
unnest(home_sales_fixed_models)
```

```
## # A tibble: 1,944 x 14
##    location account parcel_number reception sale_date sale_price
##    <chr>    <chr>   <chr>         <chr>     <chr>          <dbl>
##  1 CARBOND~ R340967 239334401005  879240    6/29/2016     650000
##  2 CARBOND~ R340073 239334200010  870778    11/24/20~     560000
##  3 CARBOND~ R112063 239335100057  869383    10/13/20~    2750000
##  4 CARBOND~ R580140 239334366004  857328    12/18/20~     630500
##  5 CARBOND~ R043949 239120300276  853541    9/12/2014    2800000
##  6 CARBOND~ R011301 239120300057  868354    9/21/2015     785000
##  7 CARBOND~ R011415 239325100148  861479    4/10/2015    1200000
##  8 CARBOND~ R005930 246304125013  875431    3/24/2016    1040000
```

```
##  9 CARBOND~ R040419 246303100026  865422    7/13/2015     340000
## 10 CARBOND~ R041666 239334268001  868224    9/15/2015     395000
## # ... with 1,934 more rows, and 8 more variables: situs_address <chr>,
## #   architectural_style <chr>, year_built <dbl>, bedrooms <dbl>,
## #   baths <dbl>, square_feet <dbl>, legal <chr>, classification <chr>
```

### Step 2: map() to fit lm to each data frame

```
library(purrr)
home_sales_fixed_models <- home_sales_fixed %>%
  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_fixed_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_fixed_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_fixed_models <- home_sales_fixed %>%
  nest(-location) %>%
  mutate(models = map(data, ~lm(sale_price ~ square_feet, .))) %>%
  mutate(tidied = map(models, tidy))
home_sales_fixed_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_fixed_models$tidied[[3]]
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  70046.  11041.        6.344 7.489e-10
## 2 square_feet    130.7     5.684    23.00  3.651e-70
```

### Step 4: unnest to tidy table of coefficients

```
location_coeffs <- home_sales_fixed %>%
  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   p.value
##    <chr>      <chr>           <dbl>     <dbl>     <dbl>     <dbl>
##  1 CARBONDALE (Intercept) 163668.   34376.       4.761  2.752e- 6
##  2 CARBONDALE square_feet    206.2     12.97    15.90   6.664e-44
##  3 GLENWOOD   (Intercept) 128578.   11400.      11.28   4.421e-26
##  4 GLENWOOD   square_feet    137.6      5.363   25.66   2.235e-89
##  5 NEW CASTLE (Intercept)  70046.   11041.       6.344  7.489e-10
##  6 NEW CASTLE square_feet    130.7      5.684   23.00   3.651e-70
##  7 PARACHUTE  (Intercept)  11651.   14526.       0.8021 4.235e- 1
##  8 PARACHUTE  square_feet    100.5      8.071   12.46   9.753e-27
##  9 RIFLE      (Intercept)  46757.    8363.       5.591  4.409e- 8
## 10 RIFLE      square_feet     95.86     4.635   20.68   1.357e-63
## 11 SILT       (Intercept)  98978.   15629.       6.333  1.308e- 9
## 12 SILT       square_feet    100.1      7.875   12.71   3.296e-28
```

### Location slopes

```
location_slopes <-  location_coeffs %>%
  filter(term == "square_feet")%>%
  arrange(estimate)
location_slopes
```

```
## # A tibble: 6 x 6
##   location   term        estimate std.error statistic   p.value
##   <chr>      <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 RIFLE      square_feet    95.86     4.635     20.68 1.357e-63
```

```
## 2 SILT       square_feet   100.1      7.875     12.71 3.296e-28
## 3 PARACHUTE  square_feet   100.5      8.071     12.46 9.753e-27
## 4 NEW CASTLE square_feet   130.7      5.684     23.00 3.651e-70
## 5 GLENWOOD   square_feet   137.6      5.363     25.66 2.235e-89
## 6 CARBONDALE square_feet   206.2     12.97      15.90 6.664e-44
```

### Easily add `p.adjust()` column

```
location_slopes <-  location_coeffs %>%
  filter(term == "square_feet")%>%
  mutate(p.adjusted = p.adjust(p.value)) %>%
  arrange(estimate)
location_slopes
```

```
## # A tibble: 6 x 7
##   location   term        estimate std.error statistic   p.value p.adjusted
##   <chr>      <chr>          <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
## 1 RIFLE      square_feet    95.86     4.635     20.68 1.357e-63  5.427e-63
## 2 SILT       square_feet   100.1      7.875     12.71 3.296e-28  6.593e-28
## 3 PARACHUTE  square_feet   100.5      8.071     12.46 9.753e-27  9.753e-27
## 4 NEW CASTLE square_feet   130.7      5.684     23.00 3.651e-70  1.825e-69
## 5 GLENWOOD   square_feet   137.6      5.363     25.66 2.235e-89  1.341e-88
## 6 CARBONDALE square_feet   206.2     12.97      15.90 6.664e-44  1.999e-43
```

### Apply this process/workflow to other packages

Before we move on to `broom:augment` and `broom::glance()`, let's apply this process to `modelr::add_residuals()`.

### `mutate()` [add] a residuals column

```
library(modelr)
home_sales_fixed_models <- home_sales_fixed_models %>%
  # Add 'resid' column to data frames in 'data' list column with mutate
  # Call add_residuals() with each data-model pair
  mutate(data = map2(data, models, add_residuals))
home_sales_fixed_models
```

```
## # A tibble: 6 x 4
##   location   data               models   tidied
##   <chr>      <list>             <list>   <list>
## 1 CARBONDALE <tibble [379 x 14]> <S3: lm> <tibble [2 x 5]>
## 2 GLENWOOD   <tibble [441 x 14]> <S3: lm> <tibble [2 x 5]>
## 3 NEW CASTLE <tibble [327 x 14]> <S3: lm> <tibble [2 x 5]>
## 4 PARACHUTE  <tibble [202 x 14]> <S3: lm> <tibble [2 x 5]>
## 5 RIFLE      <tibble [370 x 14]> <S3: lm> <tibble [2 x 5]>
## 6 SILT       <tibble [225 x 14]> <S3: lm> <tibble [2 x 5]>
```

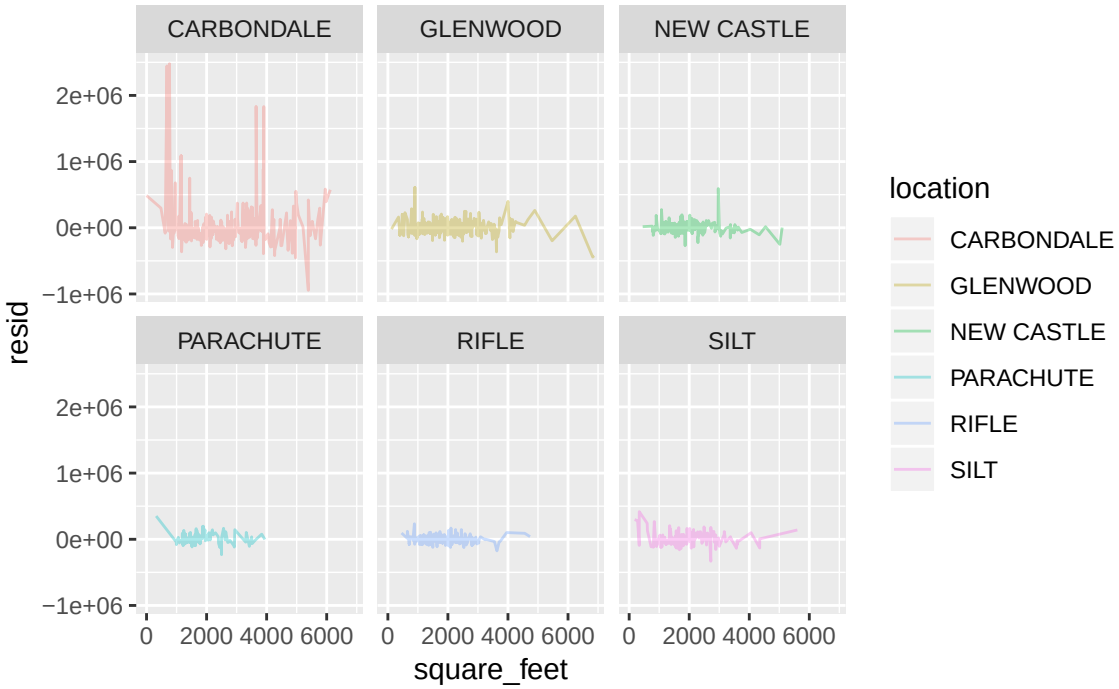
### Examine residuals for New Castle

```
home_sales_fixed_models$data[[3]] %>%
  # Select a few columns...
  select(sale_price, square_feet, resid)
```

```
## # A tibble: 327 x 3
##    sale_price square_feet    resid
##         <dbl>       <dbl>    <dbl>
##  1     145000         450   16124.
##  2     195000         768   24551.
##  3     162000         800  -12632.
##  4     205000         864   22001.
##  5     375000         900  187294.
##  6     176900         947  -16950.
##  7     255000        1064   45854.
##  8     485000        1076  274285.
##  9     268500        1148   48372.
## 10     299000        1148   78872.
## # ... with 317 more rows
```

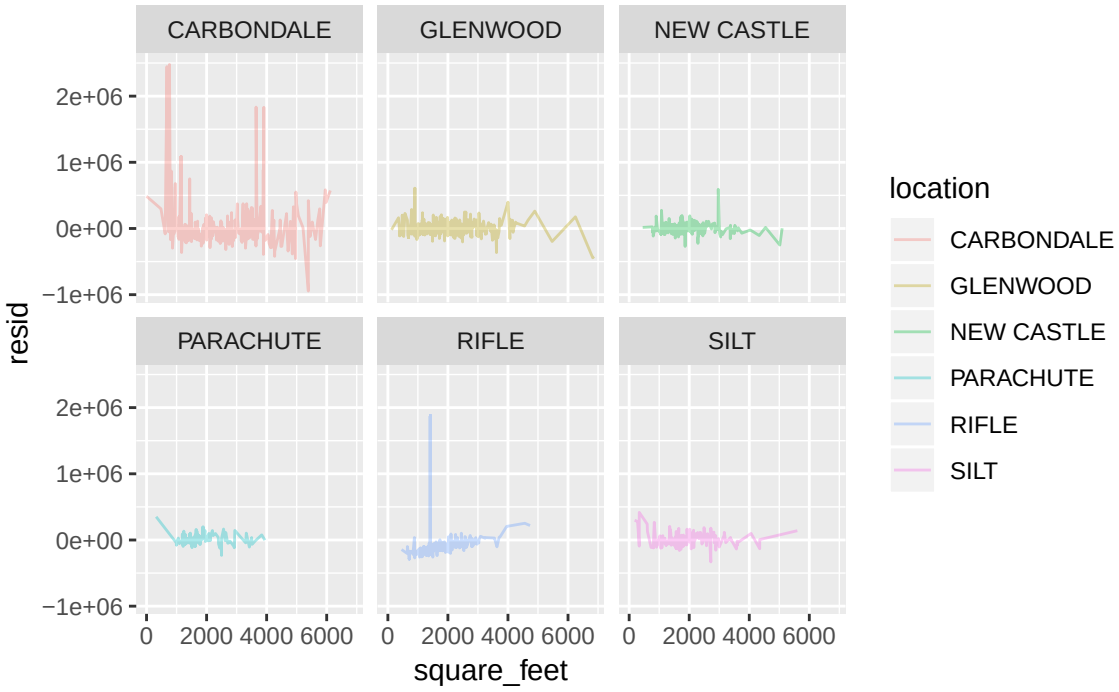
### Facet by location

```
resids <- unnest(home_sales_fixed_models, data)
resids %>%
  ggplot(aes(square_feet, resid, color = location)) +
  geom_line(alpha = 1 / 3) +
  facet_wrap(~ location)
```



### Pipe it all together!

```
home_sales_errors %>% # Data frame with the errors
  nest(-location) %>% # Step 1
  mutate(models = map(data, ~lm(sale_price ~ square_feet, .))) %>% # Step 2
  mutate(data = map2(data, models, add_residuals)) %>% # Step 3
  unnest(data) %>% #Step 4
  ggplot(aes(square_feet, resid, color = location)) + #Step 5 - plot!
  geom_line(alpha = 1 / 3) +
  facet_wrap(~ location)
```




### `broom::augment()` the results

```
home_sales_fixed_models <- home_sales_fixed_models %>%
  mutate(augmented = map(models, broom::augment))
home_sales_fixed_models
```

```
## # A tibble: 6 x 5
##   location   data              models   tidied          augmented
##   <chr>      <list>            <list>   <list>          <list>
## 1 CARBONDALE <tibble [379 x 14~ <S3: lm> <tibble [2 x 5~ <tibble [378 x 1~
## 2 GLENWOOD   <tibble [441 x 14~ <S3: lm> <tibble [2 x 5~ <tibble [441 x 9~
## 3 NEW CASTLE <tibble [327 x 14~ <S3: lm> <tibble [2 x 5~ <tibble [327 x 9~
## 4 PARACHUTE  <tibble [202 x 14~ <S3: lm> <tibble [2 x 5~ <tibble [202 x 9~
## 5 RIFLE      <tibble [370 x 14~ <S3: lm> <tibble [2 x 5~ <tibble [370 x 9~
## 6 SILT       <tibble [225 x 14~ <S3: lm> <tibble [2 x 5~ <tibble [225 x 9~
```

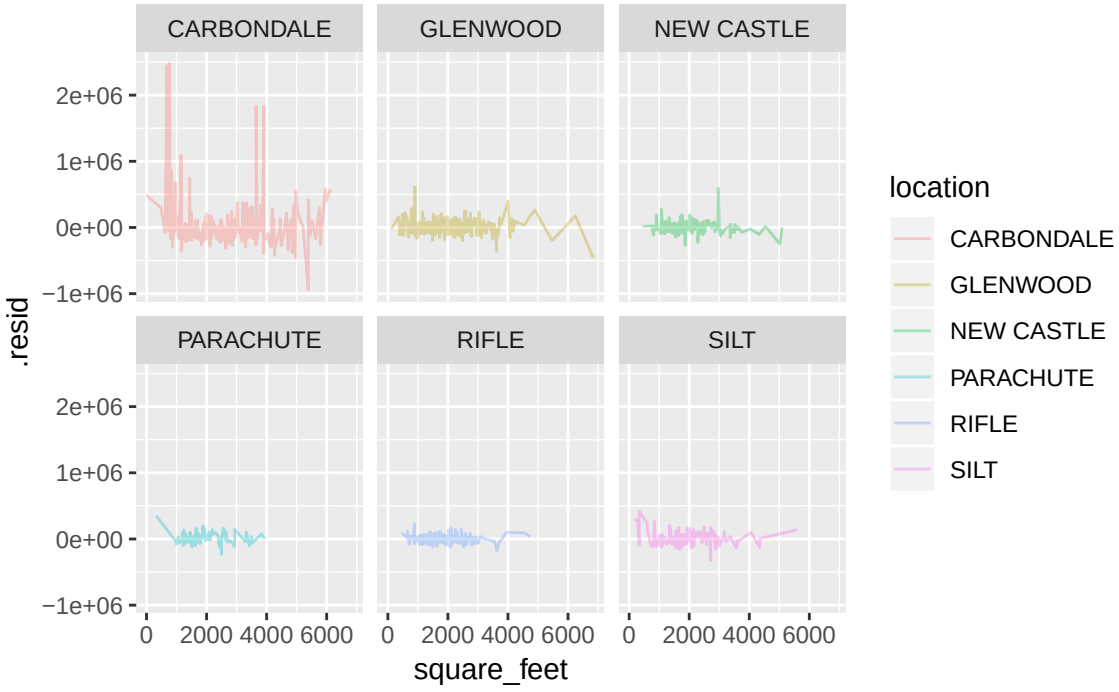
### glimpse at the list column augmented

```
augment_results <- home_sales_fixed_models %>%
  unnest(augmented)
glimpse(augment_results)
```

```
## Observations: 1,943
## Variables: 11
## $ location    <chr> "CARBONDALE", "CARBONDALE", "CARBONDALE", "CARBOND...
## $ .rownames   <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",...
## $ sale_price  <dbl> 650000, 560000, 2750000, 630500, 2800000, 785000, ...
## $ square_feet <dbl> 0, 480, 680, 710, 764, 804, 825, 957, 960, 960, 96...
## $ .fitted     <dbl> 163668.2, 262648.5, 303890.3, 310076.6, 321211.9, ...
## $ .se.fit     <dbl> 34376.15, 29000.74, 26869.49, 26556.90, 25999.44, ...
## $ .resid      <dbl> 486331.813, 297351.469, 2446109.659, 320423.387, 2...
## $ .hat        <dbl> 0.012358274, 0.008795523, 0.007550265, 0.007375614...
## $ .sigma      <dbl> 308606.7, 309255.3, 282488.0, 309193.9, 281735.7, ...
## $ .cooksd     <dbl> 1.566887e-02, 4.138936e-03, 2.398338e-01, 4.018742...
## $ .std.resid  <dbl> 1.58253989, 0.96585106, 7.94041795, 1.04004813, 8....
```

### Plot the residuals from augmented

```
augment_results %>%
ggplot(aes(square_feet, .resid, color = location)) +
  geom_line(alpha = 1 / 3) +
  facet_wrap(~ location)
```



### broom::glance() at model summaries

```
home_sales_fixed_models <- home_sales_fixed_models %>%
  mutate(glanced = map(models, broom::glance))
home_sales_fixed_models
```

```
## # A tibble: 6 x 6
##   location   data          models  tidied      augmented      glanced
##   <chr>      <list>        <list>  <list>      <list>         <list>
## 1 CARBONDALE <tibble [379~ <S3: l~ <tibble [2~ <tibble [378 ~ <tibble [1 ~
## 2 GLENWOOD   <tibble [441~ <S3: l~ <tibble [2~ <tibble [441 ~ <tibble [1 ~
## 3 NEW CASTLE <tibble [327~ <S3: l~ <tibble [2~ <tibble [327 ~ <tibble [1 ~
## 4 PARACHUTE  <tibble [202~ <S3: l~ <tibble [2~ <tibble [202 ~ <tibble [1 ~
## 5 RIFLE      <tibble [370~ <S3: l~ <tibble [2~ <tibble [370 ~ <tibble [1 ~
## 6 SILT       <tibble [225~ <S3: l~ <tibble [2~ <tibble [225 ~ <tibble [1 ~
```

### glance() results

```
glance_results <- home_sales_fixed_models %>%
  unnest(glanced)
glance_results
```

```
## # A tibble: 6 x 16
##   location data  models tidied augmented r.squared adj.r.squared   sigma
##   <chr>    <lis> <list> <list> <list>        <dbl>         <dbl>   <dbl>
## 1 CARBOND~ <tib~ <S3: ~ <tibb~ <tibble ~    0.4020        0.4005 309228.
## 2 GLENWOOD <tib~ <S3: ~ <tibb~ <tibble ~    0.6000        0.5991 111523.
## 3 NEW CAS~ <tib~ <S3: ~ <tibb~ <tibble ~    0.6195        0.6183  73936.
## 4 PARACHU~ <tib~ <S3: ~ <tibb~ <tibble ~    0.4369        0.4341  64836.
## 5 RIFLE    <tib~ <S3: ~ <tibb~ <tibble ~    0.5375        0.5362  52336.
## 6 SILT     <tib~ <S3: ~ <tibb~ <tibble ~    0.4202        0.4176  89680.
## # ... with 8 more variables: statistic <dbl>, p.value <dbl>, df <int>,
## #   logLik <dbl>, AIC <dbl>, BIC <dbl>, deviance <dbl>, df.residual <int>
```

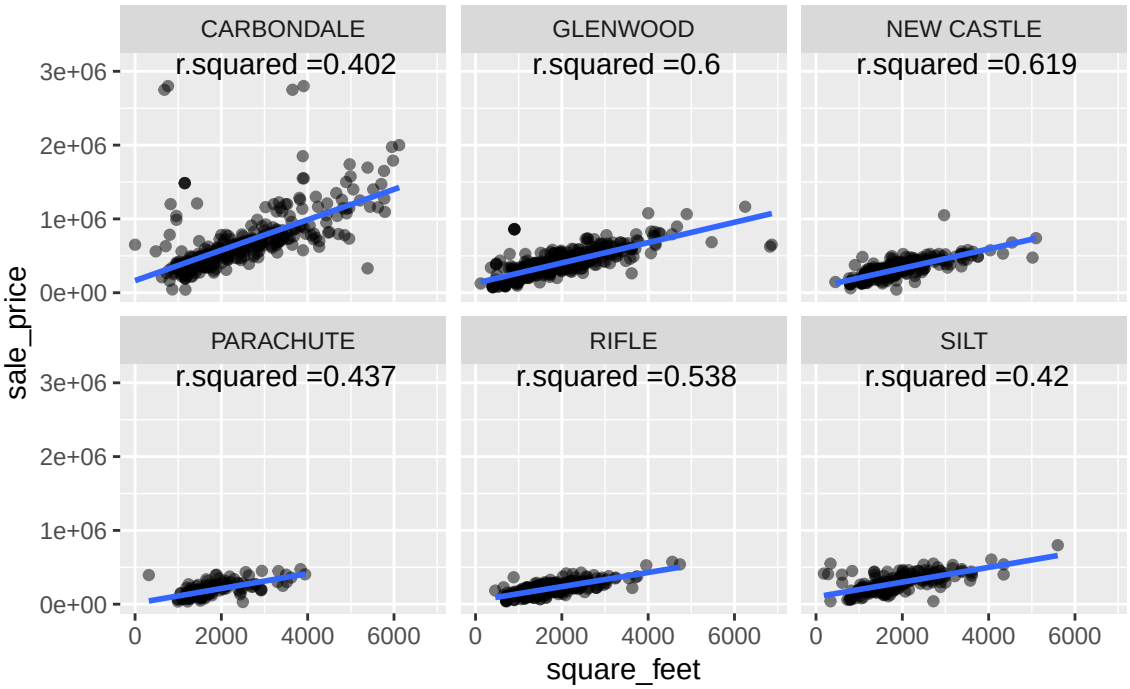
### glance() results without list columns

```
glance_results <- home_sales_fixed_models %>%
  # Supress nested columns with ".drop = TRUE"
  unnest(glanced, .drop = TRUE)
glance_results
```

```
## # A tibble: 6 x 12
##   location r.squared adj.r.squared   sigma statistic   p.value    df logLik
##   <chr>        <dbl>         <dbl>   <dbl>     <dbl>     <dbl> <int>  <dbl>
## 1 CARBOND~    0.4020        0.4005 309228.     252.8 6.664e-44     2 -5314.
## 2 GLENWOOD    0.6000        0.5991 111523.     658.4 2.235e-89     2 -5750.
## 3 NEW CAS~    0.6195        0.6183  73936.     529.1 3.651e-70     2 -4129.
## 4 PARACHU~    0.4369        0.4341  64836.     155.2 9.753e-27     2 -2524.
## 5 RIFLE       0.5375        0.5362  52336.     427.7 1.357e-63     2 -4544.
## 6 SILT        0.4202        0.4176  89680.     161.6 3.296e-28     2 -2884.
## # ... with 4 more variables: AIC <dbl>, BIC <dbl>, deviance <dbl>,
## #   df.residual <int>
```

## Add r.squared values to your plot

```
ggplot(home_sales_fixed, aes(square_feet, sale_price)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  geom_text(data = glance_results, aes(x = 3500, y = 3100000,
                label = paste0("r.squared =", round(r.squared, 3)))) +
  facet_wrap(~ location)
```




## `tidyverse` packages used

### Importing

- `readr`; `readxl`

### Wrangle

- `dplyr`; `tidyr`; `stringr`; `tibble`

### Visualize

- `ggplot2`

### Program

- `purrr`; `magrittr`

### Model

- `broom`; `modelr`

## Resources

- `broom` and `dplyr` vignette

- `broom` intro by David Robinson
- Exploratory Data Analysis in R by David Robinson on DataCamp
- R for Data Science by Hadley Wickham
- ggplot2 book by Hadley Wickham

## Questions?

casey.bates@erm.com

Thank you!