RFinalHYu

Haozhe (Jerry) Yu

2023-04-27

Data Import

Read in Raw Data

Read in the raw data directly form the url.

```
rawhouse <- read.csv("https://www4.stat.ncsu.edu/~online/ST308/Data/hyu23_house.csv")
```

Data Subsetting

Create a tibble from the read in data table with the following modifications:

1. Remove any observations where
    - the `SaleType` variable takes the value "Other" or
    - the `BedroomAbvGr` variable takes on a value less than or equal to 2
2. Create a new variable with a name of your choosing that is the `SalePrice` variable divided by 100000.
3. The `GarageArea` and `MSZoning` variables are removed

```
House <- rawhouse %>%
         filter(SaleType != "Other") %>%
         filter(BedroomAbvGr > 2) %>%
         mutate(SalePrice100k = SalePrice/100000) %>%
         select(-GarageArea, -MSZoning)
```

Now print out the first 10 observations and first 6 variables of House.

```
House %>%
  select(SalePrice, BsmtUnfSF, OverallQual, OpenPorchSF, BedroomAbvGr, YrSold) %>%
  slice(1:10) %>%
  kable()
```

SalePrice	BsmtUnfSF	OverallQual	OpenPorchSF	BedroomAbvGr	YrSold
208500	150	7	61	3	2008
181500	284	6	0	3	2007
223500	434	7	42	3	2008
140000	540	7	35	3	2006
250000	490	8	84	4	2008
307000	317	8	57	3	2007
200000	216	7	204	3	2009
279500	1494	7	33	3	2007
159000	468	5	102	3	2008
139000	525	5	0	3	2009

Output Creation Steps

Contingency Tables

Create a 2 way contingency table between `BsmtFinType2` and `LotShap`.

```
contingency <- table(House$BsmtFinType2,
      House$LotShap)
contingency
```

```
##
##         IR1 IR2or3 Reg
##   Other  29      4  68
##   Unf   229     24 361
```

The upper most value of 29 is the number of observations where `BsmtFinType2` equals "Other" and `LotShap` equals "IR1".

Numeric Summaries

Create the tibble `ssatshouse` storing the following summary statistics:

1. sample meal
2. sample standard deviation
3. sample 1st quartile
4. sample 3rd quartile

For the following variables:

1. SalePrice
2. BsmtUnfSF
3. OverallQual

At every level of the `BsmtCond` variable. Then use `kable()` to print out `ssatshouse`.

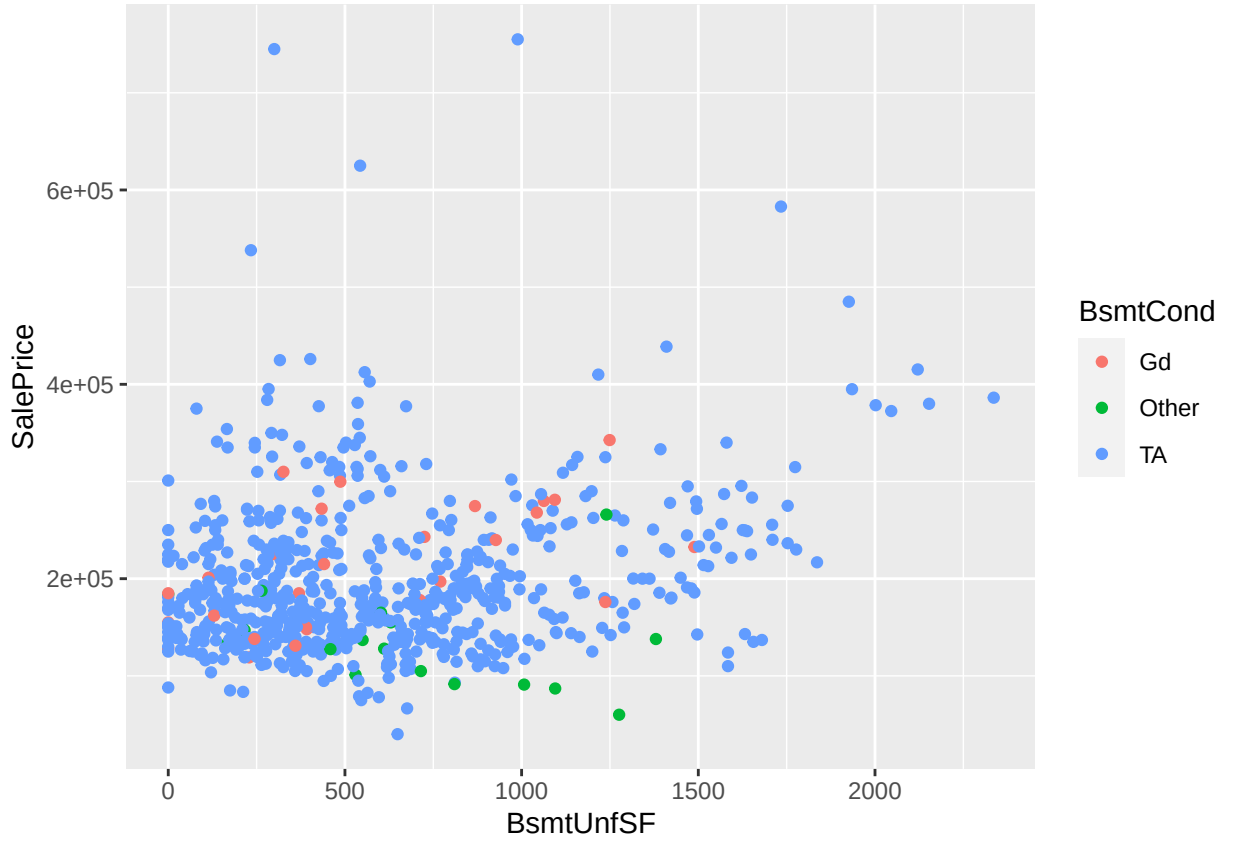
```
ssatshouse <- House %>%
              group_by(BsmtCond) %>%
              reframe(
                across(c(SalePrice,BsmtUnfSF,OverallQual),
                       list(mean = ~mean(., na.rm = TRUE),
                            sd = ~sd(., na.rm = TRUE),
                            q1 = ~quantile(., 0.25, na.rm = TRUE),
                            q3 = ~quantile(., 0.75, na.rm = TRUE)
                            ),
                           .names = "{col}_{fn}"
                         )
                        )

kable(ssatshouse)
```

BsmtCond	SalePrice_mean	SalePrice_sd	SalePrice_q1	SalePrice_q3	BsmtUnfSF_mean	BsmtUnfSF_sd	Bs
.	118417.2	17470.96	110625	128108.0	0.0000	0.0000	
Gd	209888.6	63106.75	153725	269000.0	568.8214	418.0385	
Other	135534.1	48534.55	101000	155000.0	679.5882	408.5473	
TA	196140.7	79025.24	142125	228837.5	597.7128	441.9203	

Scatterplots

Create a scatter plot using `SalePrice` on the y-axis and `BsmtUnfSF` on the x-axis. Color the points by the `BsmtCond` variable.

```
filterhouse <- filter(House, BsmtCond != ".")
scatter <- ggplot(data=filterhouse, aes(x = BsmtUnfSF, y = SalePrice, color=BsmtCond),) +
  geom_point(na.rm = TRUE)
scatter
```

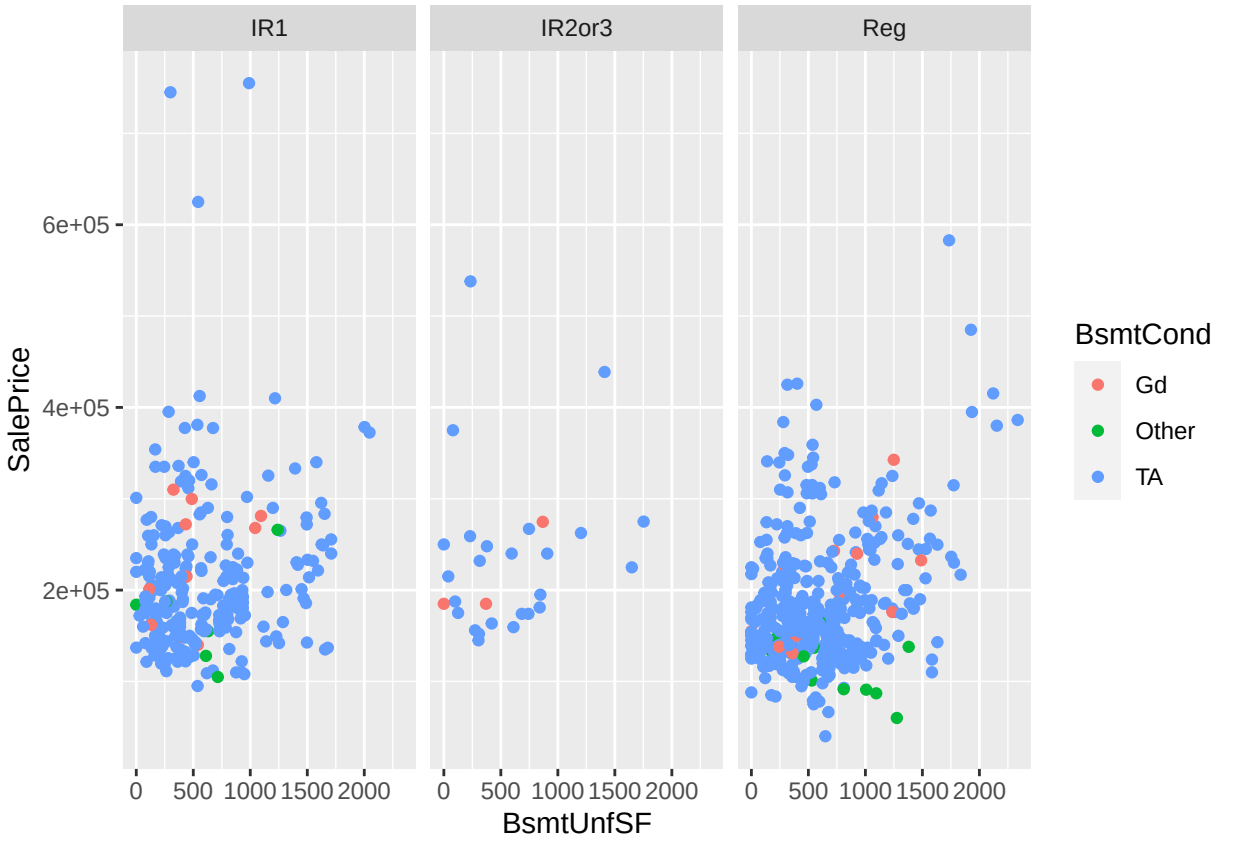


There seems to be a weak positive correlation between `SalePrice` and `BsmtUnfSF`, a code of "gd' seemed to positively correlate sale price, while a code of"other seemed to negatively correlate with sale price.

Panel Scatter

Create the same plot as above at every level of the `LotShape` variable.

```
scatterfacet <- ggplot(data=filterhouse, aes(x = BsmtUnfSF, y = SalePrice, color=BsmtCond)) +
  geom_point() +
  facet_wrap(~LotShape)
scatterfacet
```



Regression

Create a multiple linear regression between `SalePrice` as the response variable and `BsmtUnfSF` and `OverallQual` as predictors, sans interactions. The do the following things with the model: 1.) Inspect the `summary()` of the model discuss if any parameters are significant 2.) USe `predict()` to predict a future SalePrice at two different combinations of `BsmtUnfSF` and `OverallQual`.

```
#create multiple linear regression model
mlrhouse <- lm(SalePrice ~ BsmtUnfSF + OverallQual, data=House, na.action = na.omit)

#structure
summary(mlrhouse)
```

```
##
## Call:
## lm(formula = SalePrice ~ BsmtUnfSF + OverallQual, data = House,
##     na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -186263  -27407   -2426   21013  384172
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.062e+05  9.144e+03 -11.619   <2e-16 ***
```

```
## BsmtUnfSF   -1.665e+00  4.366e+00  -0.381    0.703
## OverallQual  4.787e+04  1.503e+03  31.849   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48880 on 712 degrees of freedom
## Multiple R-squared:  0.6123, Adjusted R-squared:  0.6112
## F-statistic: 562.3 on 2 and 712 DF,  p-value: < 2.2e-16
```

```
#predict
housepredict <- data.frame(BsmtUnfSF = c(787, 80), OverallQual = c(7, 8))
predictionshouse <- predict(mlrhouse, housepredict)
predictionshouse
```

```
##        1        2
## 227548.7 276597.6
```

In the multiple regression model, the p value for `BsmtUnfSF` was 0.703, meaning that it was not significant at a standard alpha of p<0.05, while the p value for `OverallQual` was <2e-16, meaning that it was highly significant at a standard alpha of p<0.05.

For my 2 values, the first data point had `BsmtUnfSF`= 787 and `OverallQual` = 7, while the second had `BsmtUnfSF`= 80 and `OverallQual` = 8. The predicted values were 2.2754872×10^5 and 2.7659759×10^5, respectively.