COMPSCIX 415.2 Homework 7

Rajat Jain

July 19th, 2018

Contents

Exercises

Exercise 1

Load the train.csv dataset into R. How many observations and columns are there?

```
train_data <- read_csv("train.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Id = col_integer(),
##   MSSubClass = col_integer(),
##   LotFrontage = col_integer(),
##   LotArea = col_integer(),
##   OverallQual = col_integer(),
##   OverallCond = col_integer(),
##   YearBuilt = col_integer(),
##   YearRemodAdd = col_integer(),
##   MasVnrArea = col_integer(),
##   BsmtFinSF1 = col_integer(),
##   BsmtFinSF2 = col_integer(),
##   BsmtUnfSF = col_integer(),
##   TotalBsmtSF = col_integer(),
##   `1stFlrSF` = col_integer(),
##   `2ndFlrSF` = col_integer(),
##   LowQualFinSF = col_integer(),
##   GrLivArea = col_integer(),
##   BsmtFullBath = col_integer(),
##   BsmtHalfBath = col_integer(),
##   FullBath = col_integer()
##   # ... with 18 more columns
## )
```

```
## See spec(...) for full column specifications.
```

There are **1460** observations and **81** columns in `train.csv` data set.

Exercise 2

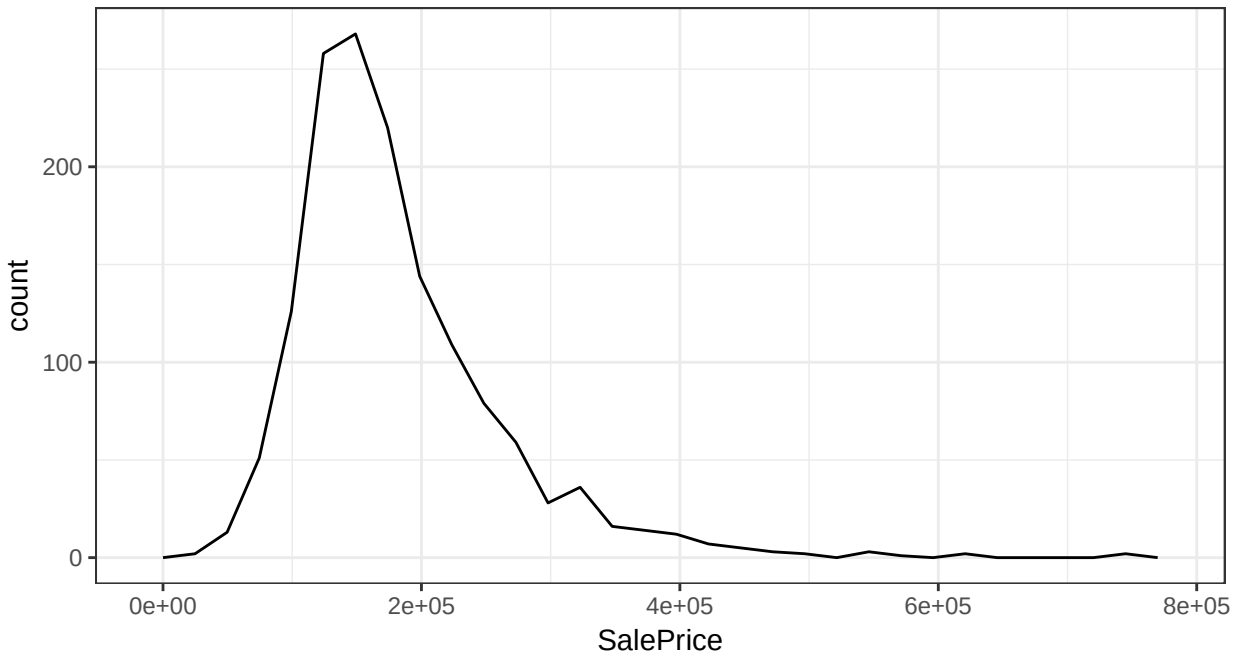
Normally at this point you would spend a few days on EDA, but for this homework we will do some very basic EDA and get right to fitting some linear regression models.

Our target will be `SalePrice`.

- **Visualize the distribution of `SalePrice`.**

Here is a plot of frequency distribuion of `SalePrice`.

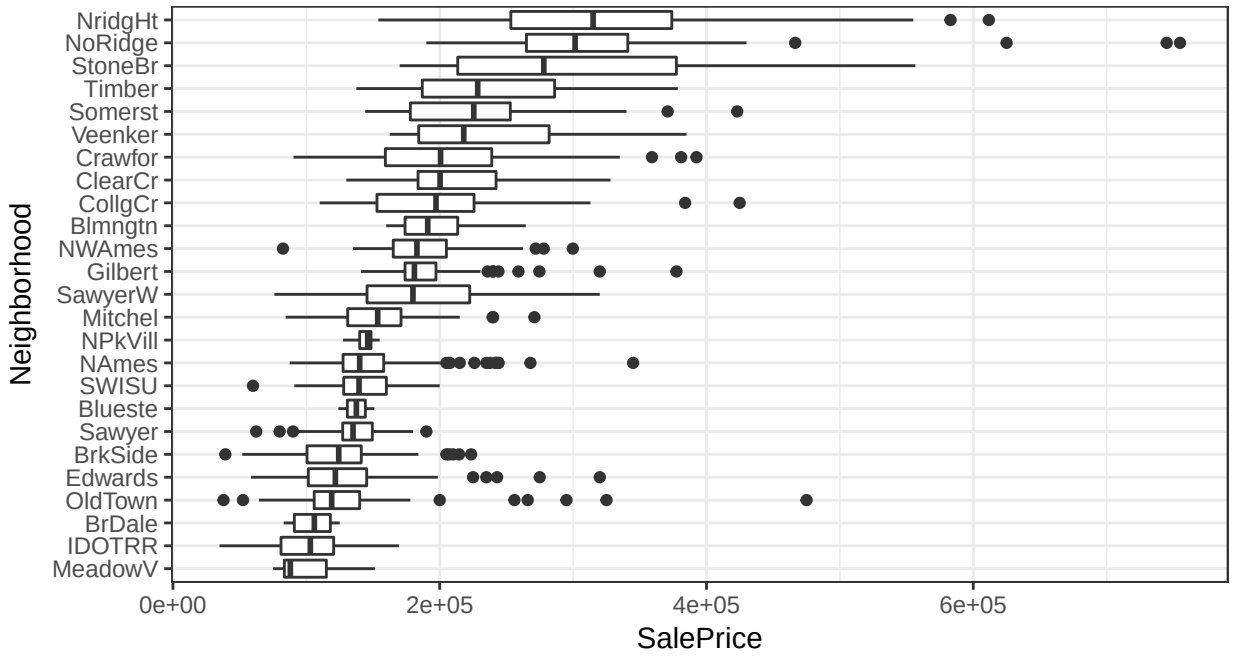
```
ggplot(data = train_data, aes(x = SalePrice)) +
  geom_freqpoly() + theme_bw()
```



The distribution is skewed towards right. This means that most `SalePrice` are with-in $300K with some exceptions.

- **Visualize the covariation between `SalePrice` and `Neighborhood`.**

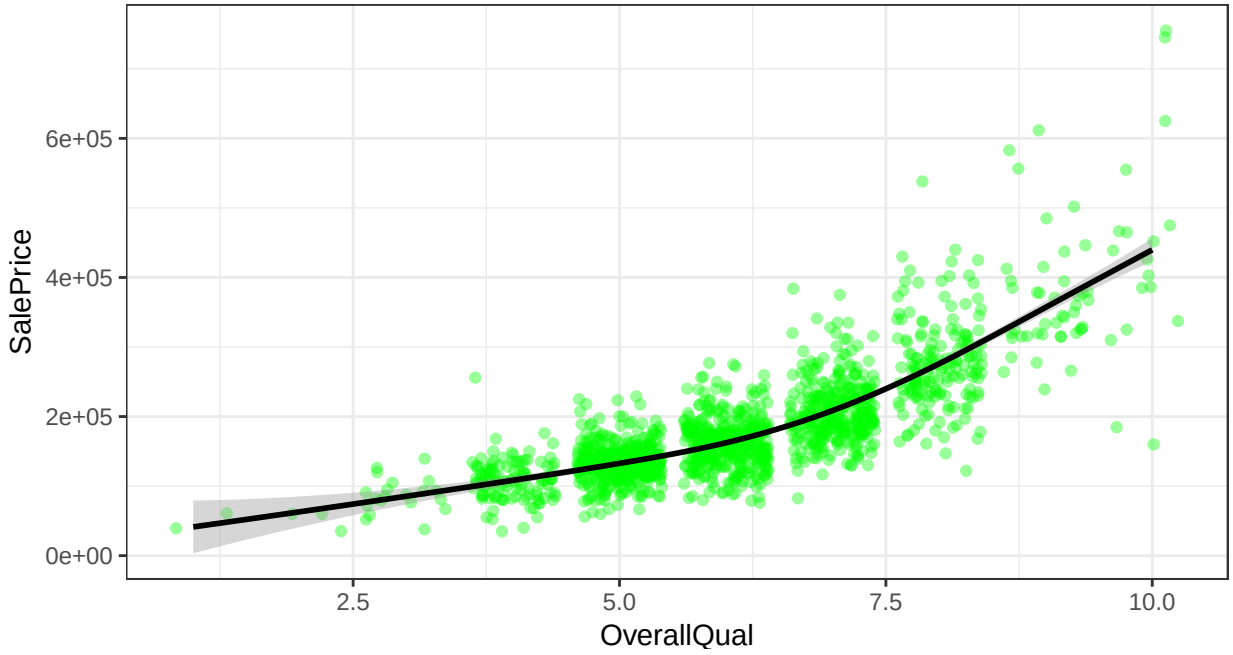
```
ggplot(data = train_data, aes(x = reorder(Neighborhood, SalePrice, FUN = median), y = SalePrice)) +
  geom_boxplot() + xlab("Neighborhood") +
  coord_flip() + theme_bw()
```



The plot above shows that median `SalePrice` vary a bit by `Neighborhood`. However, the correlation doesn't seem to be a very strong one.

- **Visualize the covariation between `SalePrice` and `OverallQual`.**

```
ggplot(data = train_data, aes(x = OverallQual, y = SalePrice)) +
  geom_point(position = "jitter", color = "green", alpha = 0.4) +
  geom_smooth(color = "black") + theme_bw()
```



`SalePrice` show a strong positive covariation with `OverallQual`.

Exercise 3

Our target is called `SalePrice`. First, we can fit a simple regression model consisting of only the intercept (the average of `SalePrice`). Fit the model and then use the `broom` package to

Here is the code to fit a simple regression model consisting of only the intercept:

```
model <- lm(formula = SalePrice ~ 1, data = train_data)
```

- **take a look at the coefficient,**

```
tidy(model)
```

```
## # A tibble: 1 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  180921.     2079.      87.0      0.
```

Coefficient (Intercept) is 180921.2.

- **compare the coefficient to the average value of `SalePrice`, and**

```
mean(train_data$SalePrice)
```

```
## [1] 180921.2
```

Coefficient is equal to the average value of `SalePrice`.

- **take a look at the R-squared.**

```
glance(model)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared  sigma statistic p.value    df  logLik    AIC
##       <dbl>         <dbl>  <dbl>     <dbl>   <dbl> <int>   <dbl>  <dbl>
## 1        0.            0. 79443.        NA      NA     1 -18544. 37092.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

R-squared value is 0. It means that this model does not explain any variability in the data.

Exercise 4

Now fit a linear regression model using `GrLivArea`, `OverallQual`, and `Neighborhood` as the features. Don't forget to look at data_description.txt to understand what these variables mean. Ask yourself these questions before fitting the model:

- **What kind of relationship will these features have with our target?**

These features should have a direct positive relationship with our target.

- **Can the relationship be estimated linearly?**

Yes. Although not sure about `Neighborhood`, as it is a categorical feature.

- **Are these good features, given the problem we are trying to solve?**

Yes, it seems so based on the EDA above.

Here is the code to fit the multiple linear regression model:

```
linear <- lm(formula = SalePrice ~ GrLivArea + OverallQual + Neighborhood, data = train_data)
```

After fitting the model, output the coefficients and the R-squared using the `broom` package.

```
# Coefficients
tidy(linear)
```

```
## # A tibble: 27 x 5
##    term                estimate std.error statistic  p.value
##    <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)         -34829.   11541.       -3.02 2.59e- 3
##  2 GrLivArea               55.6      2.50     22.2  2.38e-94
##  3 OverallQual          20951.    1162.       18.0  1.24e-65
##  4 NeighborhoodBlueste -30753.   27697.       -1.11 2.67e- 1
##  5 NeighborhoodBrDale  -43359.   12979.       -3.34 8.57e- 4
##  6 NeighborhoodBrkSide -13025.   10450.       -1.25 2.13e- 1
##  7 NeighborhoodClearCr  24576.   11570.        2.12 3.38e- 2
##  8 NeighborhoodCollgCr  11414.    9497.        1.20 2.30e- 1
##  9 NeighborhoodCrawfor  14444.   10502.        1.38 1.69e- 1
## 10 NeighborhoodEdwards -17843.    9986.       -1.79 7.42e- 2
## # ... with 17 more rows
```

```
# R-squared
glance(linear)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared  sigma statistic p.value    df  logLik    AIC
## *     <dbl>         <dbl>  <dbl>     <dbl>   <dbl> <int>   <dbl>  <dbl>
## 1     0.787         0.783 37009.      203.      0.    27 -17416. 34887.
## # ... with 3 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>
```

Answer these questions:

- **How would you interpret the coefficients on `GrLivArea` and `OverallQual`?**

Controlling for all other features, if the `GrLivArea` increases by 1 unit, the `SalePrice` increases by $55.56 on average. Similarly, keeping all other features fixed, a unit increase in `OverallQual` increases the `SalePrice` on average by $20,951.42.

- **How would you interpret the coefficient on `NeighborhoodBrkSide`?**

`SalePrice` in `NeighborhoodBrkSide` are on average $13,025.45 lower relative to those in `NeighborhoodBlmngtn` while controlling for all other features.

- **Are the features *significant*?**

`GrLivArea` and `OverallQual` both have p-values < 0.05, so statistically speaking they are significant. However, of the variables generated from categorical `Neighborhood`, the following have p-value > 0.05 - `NeighborhoodBlueste`, `NeighborhoodBrkSide`, `NeighborhoodCollgCr`, `NeighborhoodCrawfor`, `NeighborhoodEdwards`, `NeighborhoodGilbert`, `NeighborhoodMeadowV`, `NeighborhoodMitchel`, `NeighborhoodNAmes`, `NeighborhoodNPkVill`, `NeighborhoodNWAmes`, `NeighborhoodSawyer`, `NeighborhoodSawyerW`, `NeighborhoodSomerst`. So we can say that `Neighborhood` is not statistically significant.

- **Are the features *practically significant*?**

Yes, the features are practically significant. None of the intercept is close to 0 and it makes sense for the price to change with living area and rating of overall material and finishing of the house.

- **Is the model a good fit?**

Our adjusted-R-squared value for this model is 0.78, which is pretty high. Hence we can say that the model is a good fit and a lot of variability in the data is explained by this model.

Exercise 6

One downside of the linear model is that it is sensitive to unusual values because the distance incorporates a squared term. Fit a linear model to the simulated data below (use `y` as the target and `x` as the feature), and look at the resulting coefficients and R-squared. Rerun it about 5-6 times to generate different simulated datasets. What do you notice about the model's coefficient on `x` and the R-squared values?

```
sim1a <- tibble(
  x = rep(1:10, each = 3),
  y = x * 1.5 + 6 + rt(length(x), df = 2)
)
```

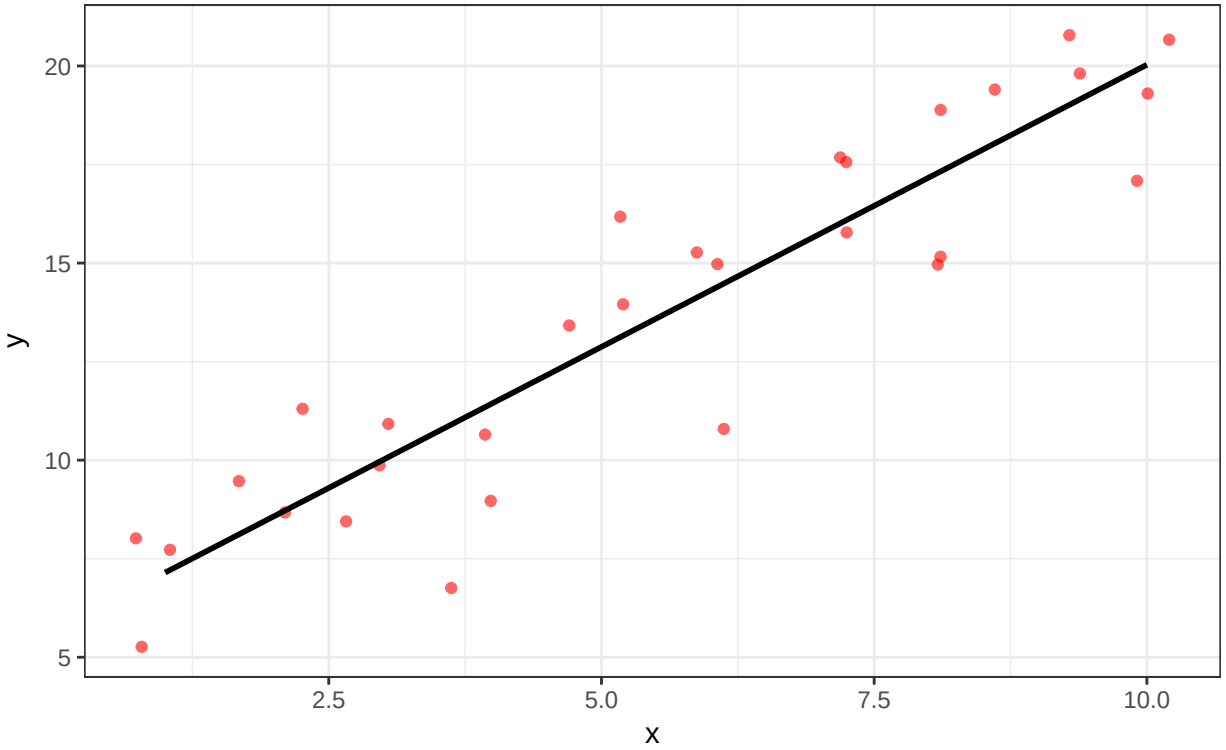
```
N <- 10

coeff <- rep(NA, N)
r2rd <- rep(NA, N)

for(i in 1:N) {
  sim_data <- tibble(
    x = rep(1:10, each = 3),
    y = x * 1.5 + 6 + rt(length(x), df = 2)
  )

  m <- lm(formula = y ~ x, data = sim_data)
  td <- tidy(m)
  coeff[i] <- td %>% filter(term == 'x') %>% select(estimate) %>% .[[1]]

  gl <- glance(m)
  r2rd[i] <- gl %>% select(r.squared) %>% .[[1]]
}

results <- tibble(
  coefficient.x = coeff,
  r.squared = r2rd
)

results
```

```
## # A tibble: 10 x 2
##    coefficient.x r.squared
##            <dbl>     <dbl>
##  1          1.30     0.757
##  2          1.36     0.906
##  3          1.42     0.868
##  4          1.39     0.804
##  5          1.49     0.907
##  6          1.44     0.699
##  7         0.642    0.0608
##  8          1.78     0.830
##  9          1.32     0.758
## 10          1.43     0.826
```

Model's coefficient on `x` and the R-squared values vary greately with each iteration, although the values of `y` in simulation data is generated as a linear function of `x`.

But the random term being added to it generates some unusual values as shown in the plot below.

```
ggplot(sim_data, aes(x = x, y = y)) +
  geom_point(position = "jitter", color = "red", alpha = 0.6) +
  stat_smooth(method = "lm", color = "black", se = FALSE) +
  theme_bw()
```



The points which are far away from the fit line, have great impact on the slope.