exercise_8.2_FigueroaHolly

Holly Figueroa

5/3/2021

Packages: QuantPsych Rcmdr

```
knitr::opts_chunk$set(echo = TRUE)
```

#i
Explain any transformations or modifications you made to the dataset

```
#load housing df
housing_wk8 <- read_xlsx("hello-world/assignment_06_07_files_FigueroaHolly/week-6-housing.xlsx")

#Data changes
colnames(housing_wk8)[c(1,2)]<-c("sale_date", "sale_price")
colnames(housing_wk8)
```

```
##  [1] "sale_date"                "sale_price"
##  [3] "sale_reason"              "sale_instrument"
##  [5] "sale_warning"             "sitetype"
##  [7] "addr_full"                "zip5"
##  [9] "ctyname"                  "postalctyn"
## [11] "lon"                      "lat"
## [13] "building_grade"           "square_feet_total_living"
## [15] "bedrooms"                 "bath_full_count"
## [17] "bath_half_count"          "bath_3qtr_count"
## [19] "year_built"               "year_renovated"
## [21] "current_zoning"           "sq_ft_lot"
## [23] "prop_type"                "present_use"
```

#ii
Create two variables; one that will contain the variables Sale Price and Square Foot of Lot (same variables used from previous assignment on simple regression) and one that will contain Sale Price and several additional predictors of your choice. Explain the basis for your additional predictor selections.

```
housing_sp_lot <- (housing_wk8)[c(2,22)]
head(housing_sp_lot)
```

```
## # A tibble: 6 x 2
##   sale_price sq_ft_lot
##        <dbl>     <dbl>
## 1     698000      6635
## 2     649990      5570
## 3     572500      8444
## 4     420000      9600
## 5     369900      7526
## 6     184667      7280
```

```
housing_sp_vars <- (housing_wk8)[c(2,13,14,15,16,22)]
head(housing_sp_vars)
```

```
## # A tibble: 6 x 6
##   sale_price building_grade square_feet_tota~ bedrooms bath_full_count sq_ft_lot
##        <dbl>          <dbl>             <dbl>    <dbl>           <dbl>     <dbl>
## 1     698000              9              2810        4               2      6635
## 2     649990              9              2880        4               2      5570
## 3     572500              8              2770        4               1      8444
## 4     420000              8              1620        3               1      9600
## 5     369900              7              1440        3               1      7526
## 6     184667              7              4160        4               2      7280
```

#iii Execute a summary() function on two variables defined in the previous step to compare the model results. What are the R2 and Adjusted R2 statistics? Explain what these results tell you about the overall model. Did the inclusion of the additional predictors help explain any large variations found in Sale Price?

```
housing_lm_1 <- lm(sale_price ~ sq_ft_lot, data = housing_sp_lot)
housing_lm_2 <- lm(sale_price ~ building_grade
                   + square_feet_total_living + bedrooms
                   + bath_full_count + sq_ft_lot, data = housing_sp_vars)

summary(housing_lm_1)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = housing_sp_lot)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2016064  -194842   -63293    91565  3735109
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.418e+05  3.800e+03  168.90   <2e-16 ***
## sq_ft_lot   8.510e-01  6.217e-02   13.69   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 401500 on 12863 degrees of freedom
## Multiple R-squared:  0.01435,    Adjusted R-squared:  0.01428
## F-statistic: 187.3 on 1 and 12863 DF,  p-value: < 2.2e-16
```

```
summary(housing_lm_2)
```

```
##
## Call:
```

```
## lm(formula = sale_price ~ building_grade + square_feet_total_living +
##     bedrooms + bath_full_count + sq_ft_lot, data = housing_sp_vars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1915140  -113296   -42511    40683  3755382
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -3.726e+04  3.148e+04  -1.184   0.2366
## building_grade            3.781e+04  4.435e+03   8.526  < 2e-16 ***
## square_feet_total_living  1.493e+02  5.912e+00  25.255  < 2e-16 ***
## bedrooms                 -1.794e+04  4.494e+03  -3.991 6.61e-05 ***
## bath_full_count           3.705e+04  5.738e+03   6.456 1.11e-10 ***
## sq_ft_lot                 1.360e-01  5.771e-02   2.356   0.0185 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358000 on 12859 degrees of freedom
## Multiple R-squared:  0.2166, Adjusted R-squared:  0.2163
## F-statistic: 711.2 on 5 and 12859 DF,  p-value: < 2.2e-16
```

For a single variable regression, like our first model, "housing_lm_1" the R2 statistic is the square of the correlation between sale price and lot size. Specifically, it represents how much variability in sale price can be accounted for by lot size. The adjusted R sqaured represents the same measure in regards to sale price, but it reflects how it would be expected to change when applied to more/new data. When both values are close, it indicates that the model will generalize well. The first model indicates that variability in lot size explains almost none of the changes sale price, at approx 0.01 for both R2 and adjusted R2. The second model has much higher values compared to the first. The second model indicates that the predicting variables account for roughly 21% of the variabiliy found in sale price. Because the both values for R2 and adjusted R2 are nearly identical, we can also conclude that the second model has good generalizability.

#iv
Considering the parameters of the multiple regression model you have created. What are the standardized betas for each parameter and what do the values indicate?

```
lm.beta(housing_lm_1)
```

```
## sq_ft_lot
## 0.1198122
```

```
lm.beta(housing_lm_2)
```

```
##           building_grade square_feet_total_living                 bedrooms
##               0.10216546               0.36546358              -0.03886233
##          bath_full_count                sq_ft_lot
##               0.05961918               0.01914434
```

The standardized betas allow variables of different scales to be compared by showing the impact of each variable in terms of standard deviation. The ouput for the standardized betas for the linear model "housing_lm_1" indicate that with every standard devation increase in size of the property's lot size by square feet, there is an increase in the sale price by 0.11 standard deviations. The ouput for the model "housing_lm_2" indicates how each variable similarly impacts the sale price. The analysis shows that the variable with the most predictive impact is square feet of the total living space, where an increase by one standard deviation of this variable is found with an increase of .37 standard deviations.

#v
Calculate the confidence intervals for the parameters in your model and explain what the results indicate.

```
confint(housing_lm_1)
```

```
##                    2.5 %       97.5 %
## (Intercept) 6.343730e+05 6.492698e+05
## sq_ft_lot   7.291208e-01 9.728641e-01
```

```
confint(housing_lm_2)
```

```
##                                  2.5 %        97.5 %
## (Intercept)              -9.896334e+04 24443.2970173
## building_grade            2.911905e+04 46504.0252857
## square_feet_total_living  1.377186e+02   160.8951316
## bedrooms                 -2.674661e+04 -9127.6129384
## bath_full_count           2.579848e+04 48291.8566980
## sq_ft_lot                 2.284809e-02     0.2491056
```

The confidence intervals caluculated for "housing_lm_1" are based on the *b* values of our linear model which estimate change from the mean sale price given our parameter of lot size. The smaller the range between the two values 7.29 and 9.73 adds confidence that the predictor's *b* value is close to the population's real *b* value. The confidence intervals calculated for "housing_lm_2" highlight a serious concern as the boundaries listed, -1.39 and 4.41 are not only farther apart than the rest, but they cross zero. This means that in some cases the number of bedrooms sometimes means an increase in sale_price and at other times, means a decrease in sale price. This is a fundamental problem of lacking directionality. For predictor to be useful in our current model, their relationship to the desired output "sale price" should at least be consistent. All other variables appear to have acceptable confidence boundaries for *b* as they all appear to have a short range and consistent direction.

#vi
Assess the improvement of the new model compared to your original model (simple regression model) by testing whether this change is significant by performing an analysis of variance.

```
anova(housing_lm_1, housing_lm_2)
```

```
## Analysis of Variance Table
##
## Model 1: sale_price ~ sq_ft_lot
## Model 2: sale_price ~ building_grade + square_feet_total_living + bedrooms +
##     bath_full_count + sq_ft_lot
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12859 1.6479e+15  4 4.2548e+14 830.03 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> The analysis shows that the second model, “housing_lm_2”, at 4 degrees of freedom did perform significantly better with a p value well below 0.001.

#vii
Perform casewise diagnostics to identify outliers and/or influential cases, storing each function’s output in a dataframe assigned to a unique variable name.

```
housing_sp_vars$standardized.residuals<-rstandard(housing_lm_2)
housing_sp_vars$studentized.residuals<-rstudent(housing_lm_2)
housing_sp_vars$cooks.distance<-cooks.distance(housing_lm_2)
housing_sp_vars$dfbeta<-dfbeta(housing_lm_2)
housing_sp_vars$dffit<-dffits(housing_lm_2)
housing_sp_vars$leverage<-hatvalues(housing_lm_2)
housing_sp_vars$covariance.ratios<-covratio(housing_lm_2)
```

#viii Calculate the standardized residuals using the appropriate command, specifying those that are +-2, storing the results of large residuals in a variable you create.

```
housing_sp_vars$large.residual<-
  housing_sp_vars$standardized.residuals > 2 | housing_sp_vars$standardized.residuals < -2
```

#ix Use the appropriate function to show the sum of large residuals.

```
sum(housing_sp_vars$large.residual)
```

```
## [1] 322
```

#x
Which specific variables have large residuals (only cases that evaluate as TRUE)?

```
large_res<-housing_sp_vars%>%
  filter(large.residual == 1)
```

#xii
Investigate further by calculating the leverage, cooks distance, and covariance ratios. Comment on all cases that are problematic.

```
#calculate average leverage for comparison with 4 parameters
avg_leverage = (4+1)/12865
avg_leverage
```

```
## [1] 0.0003886514
```

```
#calculate limit(s) leverage should not exceed
leverage_limit= avg_leverage*2
leverage_limit
```

```
## [1] 0.0007773028
```

```
leverage_limit3 = avg_leverage*3
leverage_limit3
```

```
## [1] 0.001165954
```

```
#get count of samples over leverage limit
large_res%>%
  filter(leverage > leverage_limit)%>%
  nrow()
```

```
## [1] 111
```

```
large_res%>%
  filter(leverage > leverage_limit3)%>%
  nrow()
```

```
## [1] 87
```

> Here the calculated average leverage for our variables has been determined and doubled to create a boundary for our data. When this boundary is applied to the dataframe of large residuals, we find that 111 cases go beyond this boundary. Even if we loosen the criteria, changing the limit to 3 times the average leverage, we still find 87 cases that exceed it. These cases now include data that not only demonstrates unusually large residuals, but they are also calculated to be having a disproportionate influence on our model's outcome compared to other cases.

```
#Search for cook's distance values that exceed 1.0
large_res%>%
  filter(cooks.distance > 1)%>%
  nrow()
```

```
## [1] 0
```

```
#Calculate upper CVR boundary
upper_cvr = 1 + (3*(4+1)/12865)
upper_cvr
```

```
## [1] 1.001166
```

```
#Calculate lower CVR boundary
lower_cvr = 1 - (3*(4+1)/12865)
lower_cvr
```

```
## [1] 0.998834
```

```
#Check for values outside of CVR boundaries
large_res%>%
  filter(covariance.ratios > upper_cvr)%>%
  nrow()
```

```
## [1] 20
```

```
large_res%>%
  filter(covariance.ratios < lower_cvr)%>%
  nrow()
```

```
## [1] 250
```

```
#Percentage of sample with residuals over (+/-)2
nrow(housing_wk8)
```

```
## [1] 12865
```

```
nrow(large_res)
```

```
## [1] 322
```

```
322/12865*100
```

```
## [1] 2.502915
```

> When reviewing the data with large residuals, no concerning data is found for cook's distance as none of the values meet or exceed 1. After calculating the average leverage as 0.00038, we find that around 124 examples that are more than twice this value, causing concern for linear model. The target for covariance ratios was found to be between .998 and 1.001.

#xiii
Perform the necessary calculations to assess the assumption of independence and state if the condition is met or not.

#xiv
Perform the necessary calculations to assess the assumption of no multicollinearity and state if the condition is met or not.

#xv
Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

#xvi
Overall, is this regression model unbiased? If an unbiased regression model, what does this tell us about the sample vs. the entire population model?

Submission Instructions

You can either save your work in your own repository and submit a link to GitHub or you can submit a PDF of your R Markdown files to the assignment link. Make sure you do not just submit an R Markdown file – it needs to either be PDF or a GitHub link.