Kavish_property_inferential

Configuration

List of the packages used and access to the data

```
#libraries ------------------------------------------------------------
library(tidyverse)
library(psych)
library(here)
library(knitr)

#data ------------------------------------------------------------------
daftdb <- here("data/daftdb.rds") %>%
  read_rds()
```

New data frame created

The original dataframe is stored in another dataframe, "dfdb" for work purposes.

```
(dfdb <- daftdb) %>%
  as_tibble()
```

```
## # A tibble: 37,487 x 8
##    index address      price  bathroom bedroom structure date       weblink
##    <dbl> <chr>        <chr>     <dbl>   <dbl> <chr>     <date>     <chr>
##  1     0 138 Church ~ "\x80~        1       3 Terraced~ 2020-03-09 /dublin/hous~
##  2     1 44 Shanglas~ "\x80~        2       3 Semi-Det~ 2020-03-09 /dublin/hous~
##  3     2 51 Grosveno~ "\x80~        2       4 Terraced~ 2020-03-09 none
##  4     3 3 Maolbuill~ "\x80~        1       3 End of T~ 2020-03-09 /dublin/hous~
##  5     4 117 Saint L~ "\x80~        1       4 Terraced~ 2020-03-09 /dublin/hous~
##  6     5 232 Charlem~ "\x80~        3       4 Semi-Det~ 2020-03-09 /dublin/hous~
##  7     6 19 Norseman~ "\x80~        1       2 Terraced~ 2020-03-09 /dublin/hous~
##  8     7 Shandon, 44~ "\x80~        1       3 Semi-Det~ 2020-03-09 /dublin/hous~
##  9     8 11 The Padd~ "\x80~        3       4 Detached~ 2020-03-09 /dublin/hous~
## 10     9 65 Annadale~ "\x80~        2       3 End of T~ 2020-03-09 /dublin/hous~
## # ... with 37,477 more rows
```

Filtering data

The rows are remove with houses either on auction or price not available. They were not replaced with 0's as this would skew the data.

```
dfdb <- dfdb %>%
  filter(str_detect(price, "Reserve") == "FALSE") %>%
  filter(str_detect(price, "AMV") == "FALSE") %>%
  filter(str_detect(price, "Price") == "FALSE") %>%
  print()
```

```
## # A tibble: 37,084 x 8
##    index address      price  bathroom bedroom structure date       weblink
```

```
##    <dbl> <chr>        <chr>     <dbl>   <dbl> <chr>     <date>     <chr>
##  1     0 138 Church ~ "\x80~        1       3 Terraced~ 2020-03-09 /dublin/hous~
##  2     1 44 Shanglas~ "\x80~        2       3 Semi-Det~ 2020-03-09 /dublin/hous~
##  3     2 51 Grosveno~ "\x80~        2       4 Terraced~ 2020-03-09 none
##  4     3 3 Maolbuill~ "\x80~        1       3 End of T~ 2020-03-09 /dublin/hous~
##  5     4 117 Saint L~ "\x80~        1       4 Terraced~ 2020-03-09 /dublin/hous~
##  6     5 232 Charlem~ "\x80~        3       4 Semi-Det~ 2020-03-09 /dublin/hous~
##  7     6 19 Norseman~ "\x80~        1       2 Terraced~ 2020-03-09 /dublin/hous~
##  8     7 Shandon, 44~ "\x80~        1       3 Semi-Det~ 2020-03-09 /dublin/hous~
##  9     8 11 The Padd~ "\x80~        3       4 Detached~ 2020-03-09 /dublin/hous~
## 10     9 65 Annadale~ "\x80~        2       3 End of T~ 2020-03-09 /dublin/hous~
## # ... with 37,074 more rows
```

Data Wrangling

The price column will be cleaned and converted into numeric type. Further, a new column is created specifying the Dublin areas of the properties**(Dublin 1,3,5,. . . .)**.

```
#replaced any string or commas ------------------------------------------------------------
dfdb$price <- str_replace_all(dfdb$price, "[a-z,A-Z]","")

#euro sign removed-------------------------------------------------------------------------
dfdb$price <- str_sub(dfdb$price,2)

#conversion time---------------------------------------------------------------------------
dfdb$price=as.double(dfdb$price)
class(dfdb$price) #successful
```

```
## [1] "numeric"
```

```
#separating areas and adding "Co. Dublin" in place of NAs ---------------------------
dfdb <- dfdb %>%
  mutate(dublin_code = str_extract(address, "Dublin [0-9]+")) %>%
  mutate(dublin_code = as.factor(dublin_code)) %>%
  mutate(dublin_code = fct_explicit_na(dublin_code, na_level = "Co. Dublin")) %>%
  print()
```

```
## # A tibble: 37,084 x 9
##    index address  price bathroom bedroom structure date       weblink
##    <dbl> <chr>    <dbl>    <dbl>   <dbl> <chr>     <date>     <chr>
##  1     0 138 Ch~ 350000        1       3 Terraced~ 2020-03-09 /dubli~
##  2     1 44 Sha~ 445000        2       3 Semi-Det~ 2020-03-09 /dubli~
##  3     2 51 Gro~ 595000        2       4 Terraced~ 2020-03-09 none
##  4     3 3 Maol~ 375000        1       3 End of T~ 2020-03-09 /dubli~
##  5     4 117 Sa~ 845000        1       4 Terraced~ 2020-03-09 /dubli~
##  6     5 232 Ch~ 550000        3       4 Semi-Det~ 2020-03-09 /dubli~
##  7     6 19 Nor~ 350000        1       2 Terraced~ 2020-03-09 /dubli~
##  8     7 Shando~ 495000        1       3 Semi-Det~ 2020-03-09 /dubli~
##  9     8 11 The~ 795000        3       4 Detached~ 2020-03-09 /dubli~
## 10     9 65 Ann~ 400000        2       3 End of T~ 2020-03-09 /dubli~
## # ... with 37,074 more rows, and 1 more variable: dublin_code <fct>
```

Statistical analysis

Descriptive analysis and inferential analysis will be done in the following sections.

Descriptive

Starting with correlation.

```
#correlation between number of bathrooms and bedrooms vs price of the house -------------
cor.test(dfdb$bathroom, dfdb$price) #p-value < 0.05 : statistically significant
```

```
##
##  Pearson's product-moment correlation
##
## data:  dfdb$bathroom and dfdb$price
## t = 108.57, df = 37082, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4833567 0.4988031
## sample estimates:
##       cor
## 0.4911185
```

```
cor.test(dfdb$bedroom, dfdb$price) #p-value < 0.05 : statistically significant
```

```
##
##  Pearson's product-moment correlation
##
## data:  dfdb$bedroom and dfdb$price
## t = 112.88, df = 37082, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4980837 0.5132342
## sample estimates:
##       cor
## 0.5056979
```

The p-values in both the cases were found to be lesser than 0.05, hence there exist a correlation between the prices of properties and the number of bathrooms and bedrooms in the house.

Also, both the values show a positive correlation.

Further, a new table was created which stored difference in prices, number of bedrooms and bathrooms, and the original prices. It also contains 'r n_distinct(dfdb$address)' unique addresses and its respective Dublin area code.

```
#calculating difference in prices ----------------------------------------------------
diffdb <- dfdb %>%
  group_by(address, dublin_code) %>%
  summarise(price_diff = max(price) - min(price), bath=max(bathroom),
            bed=max(bedroom), price=max(price)) %>%
  ungroup()
```

Several new columns will be added in order to conduct inferential analysis on the dataset in the table **diffdb**

```
#creating new column to determine high and low-priced houses ---------------------
summary(diffdb$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  125000  325000  395000  466937  525000 3600000
```

```
diffdb$price_cat <- ifelse(diffdb$price > 400000,'High price','Low price')
#another way of coding
```

```
#diffdb$price_cat <- ifelse(diffdb$price > 400000,1,0) #1=High, 0 = Low
diffdb <- diffdb %>%
  mutate(price_cat = as.factor(price_cat))

#creating new columns to determine higher and lower no. of bedrooms and bathrooms -----
summary(diffdb$bed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   3.000   3.253   4.000  14.000
```

```
diffdb$bed_cat <- ifelse(diffdb$bed > 3,'Higher bedrooms','Lower bedrooms')
diffdb <- diffdb %>%
  mutate(bed_cat = as.factor(bed_cat))

summary(diffdb$bath)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   1.853   2.000   9.000
```

```
diffdb$bath_cat <- ifelse(diffdb$bath > 2,'Higher bathrooms','Lower bathrooms')
diffdb <- diffdb %>%
  mutate(bath_cat = as.factor(bath_cat))

#In real world, we ought to filter out houses with 0 bathrooms and 0 bedrooms
#Technically, this won't be possible!
```

Inferential Tests

Relevant T-test and ANOVA will be conducted.

Conducting Independent Sample T-test price_diff vs house pricing ———————— If the difference in prices is dependent on higher or lower house prices **Null hypothesis - H0 Alternate hypothesis - H1**

H0 - There is no effect on difference in prices due to house pricing H1 - There is an effect on difference in prices due to house pricing

```
t.test(diffdb$price_diff ~ diffdb$price_cat)
```

```
##
##  Welch Two Sample t-test
##
## data:  diffdb$price_diff by diffdb$price_cat
## t = 3.6556, df = 549.8, p-value = 0.0002812
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   658.6604 2188.6301
## sample estimates:
## mean in group High price  mean in group Low price
##                1764.9667                 341.3215
```

Since the p-value (0.0002812) < 0.05, then null-hypothesis is rejected.

Conducting Independent Sample T-test: price_diff vs no. of bedrooms ———————— If the difference in prices is dependent on no. of bedrooms

H0 - There is no effect on difference in prices due to no. of bedrooms H1 - There is an effect on difference in prices due to no. of bedrooms

```
t.test(diffdb$price_diff ~ diffdb$bed_cat)
```

```
##
##  Welch Two Sample t-test
##
## data:  diffdb$price_diff by diffdb$bed_cat
## t = 3.1279, df = 326.65, p-value = 0.001919
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   631.4688 2772.0165
## sample estimates:
## mean in group Higher bedrooms  mean in group Lower bedrooms
##                     2207.0175                      505.2749
```

Since the p-value (0.001919) < 0.05, then null-hypothesis is rejected.

Conducting Independent Sample T-test: price_diff vs no. of bathrooms ———— If the difference in prices is dependent on no. of bathrooms

H0 - There is no effect on difference in prices due to no. of bathrooms H1 - There is an effect on difference in prices due to no. of bathrooms

```
t.test(diffdb$price_diff ~ diffdb$bath_cat)
```

```
##
##  Welch Two Sample t-test
##
## data:  diffdb$price_diff by diffdb$bath_cat
## t = -0.14687, df = 320.13, p-value = 0.8833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -944.9422  813.6582
## sample estimates:
## mean in group Higher bathrooms  mean in group Lower bathrooms
##                       959.3909                      1025.0329
```

Since the p-value (0.8833) > 0.05, then null-hypothesis is accepted.

Conducting One-way ANOVA test (more then 2 levels) ———————————— If the difference in prices is dependent on different areas in Dublin

H0 - There is no effect on difference in prices due to different areas in Dublin H1 - There is an effect on difference in prices due to different areas in Dublin

```
anova_one_way <- aov(diffdb$price_diff ~ diffdb$dublin_code, data = diffdb)
summary(anova_one_way)
```

```
##                      Df      Sum Sq  Mean Sq F value Pr(>F)
## diffdb$dublin_code    9   655104529 72789392   2.196 0.0203 *
## Residuals           948 31424670515 33148387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the p-value (0.0203) < 0.05, then null-hypothesis is rejected.

```r
fit1 <- lm(price_diff ~ price + bed + bath + dublin_code, data = diffdb)
summary(fit1)
```

Conducting multiple regression with several levels in a categorical variable (dublin_code)

```
##
## Call:
## lm(formula = price_diff ~ price + bed + bath + dublin_code, data = diffdb)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14919  -1318   -976   -428  53855
##
## Coefficients:
##                             Estimate     Std. Error t value Pr(>|t|)
## (Intercept)            10874.8967596   3437.1372857   3.164 0.001606 **
## price                      0.0007796      0.0008522   0.915 0.360547
## bed                      678.8749457    226.6691519   2.995 0.002816 **
## bath                    -561.1657359    227.2279091  -2.470 0.013702 *
## dublin_codeDublin 11  -11561.1468538   3367.0869742  -3.434 0.000622 ***
## dublin_codeDublin 13  -10812.9130088   3377.4340667  -3.202 0.001413 **
## dublin_codeDublin 15  -10117.1749902   3787.3889641  -2.671 0.007686 **
## dublin_codeDublin 17  -12229.5905342   3845.2555333  -3.180 0.001518 **
## dublin_codeDublin 3   -11461.0718836   3358.1880766  -3.413 0.000670 ***
## dublin_codeDublin 5   -11468.7439409   3364.8564516  -3.408 0.000681 ***
## dublin_codeDublin 7   -12060.3966422   3368.2488706  -3.581 0.000360 ***
## dublin_codeDublin 9   -10994.6027913   3363.4545604  -3.269 0.001119 **
## dublin_codeCo. Dublin -12915.0761664   4081.5139762  -3.164 0.001604 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5729 on 945 degrees of freedom
## Multiple R-squared:  0.03308,    Adjusted R-squared:  0.0208
## F-statistic: 2.694 on 12 and 945 DF,  p-value: 0.001405
```

This implies the equation is like this -

price_diff = 10874.90 + 0.00078(price) + 678.87(bedroom) - 561.17(bathroom) -11561.15(Dublin 11) - 10812.91(Dublin 13) - 10117.18(Dublin 15) - 12229.59(Dublin 17) - 11461.07(Dublin 3) - 11468.74(Dublin 5) - 12060.40(Dublin 7) - 10994.60(Dublin 9) - 12915.08(Co. Dublin)

If all the Dublin code values will be *equal to zero*, then for **Dublin 1**, price_diff = 10874.90 + 0.00078(price) + 678.87(bedroom) - 561.17(bathroom)

Further, by looking at the p-values, we can determine which variable will be statiscally significant for the variable price_diff.

```r
print("Thank you!")
```

```
## [1] "Thank you!"
```