

# MSDS 6371 Project Analysis 2

Duy Nguyen

4/8/2022

```
####Import Data####  
getwd()
```

```
## [1] "C:/Users/dnguy/OneDrive/Desktop/epicduy.github.io/house-prediction"
```

```
train <- read.csv("train.csv")  
test <- read.csv("test.csv")
```

```
####Wrangling Data####  
df_train = train  
df_test = test  
  
dim(df_train)
```

```
## [1] 1460 81
```

```
dim(df_test)
```

```
## [1] 1459 80
```

```
####train: Dealing with NAs####  
prop.table(table(is.na(df_train)))
```

```
##  
##      FALSE      TRUE  
## 0.94110435 0.05889565
```

```
colSums(is.na(df_train))
```

```
##      Id  MSSubClass  MSZoning  LotFrontage  LotArea  
##      0           0           0           259           0  
##  Street      Alley  LotShape  LandContour  Utilities  
##      0      1369           0           0           0  
##  LotConfig  LandSlope  Neighborhood  Condition1  Condition2  
##      0           0           0           0           0  
##  BldgType  HouseStyle  OverallQual  OverallCond  YearBuilt  
##      0           0           0           0           0  
##  YearRemodAdd  RoofStyle  RoofMatl  Exterior1st  Exterior2nd
```

```
##          0          0          0          0          0
##   MasVnrType   MasVnrArea   ExterQual   ExterCond   Foundation
##          8          8          0          0          0
##   BsmtQual     BsmtCond   BsmtExposure   BsmtFinType1   BsmtFinSF1
##          37          37          38          37          0
##   BsmtFinType2   BsmtFinSF2   BsmtUnfSF   TotalBsmtSF   Heating
##          38          0          0          0          0
##   HeatingQC     CentralAir   Electrical   X1stFlrSF   X2ndFlrSF
##          0          0          1          0          0
##   LowQualFinSF   GrLivArea   BsmtFullBath   BsmtHalfBath   FullBath
##          0          0          0          0          0
##   HalfBath     BedroomAbvGr   KitchenAbvGr   KitchenQual   TotRmsAbvGrd
##          0          0          0          0          0
##   Functional   Fireplaces   FireplaceQu   GarageType   GarageYrBlt
##          0          0          690          81          81
##   GarageFinish   GarageCars   GarageArea   GarageQual   GarageCond
##          81          0          0          81          81
##   PavedDrive     WoodDeckSF   OpenPorchSF   EnclosedPorch   X3SsnPorch
##          0          0          0          0          0
##   ScreenPorch     PoolArea     PoolQC          Fence   MiscFeature
##          0          0          1453          1179          1406
##   MiscVal         MoSold       YrSold       SaleType   SaleCondition
##          0          0          0          0          0
##   SalePrice
##          0
```

```
names(df_train)[sapply(df_train, anyNA)]
```

```
## [1] "LotFrontage" "Alley" "MasVnrType" "MasVnrArea" "BsmtQual"
## [6] "BsmtCond" "BsmtExposure" "BsmtFinType1" "BsmtFinType2" "Electrical"
## [11] "FireplaceQu" "GarageType" "GarageYrBlt" "GarageFinish" "GarageQual"
## [16] "GarageCond" "PoolQC" "Fence" "MiscFeature"
```

```
#We will now go down the list
```

```
#LotFrontage
```

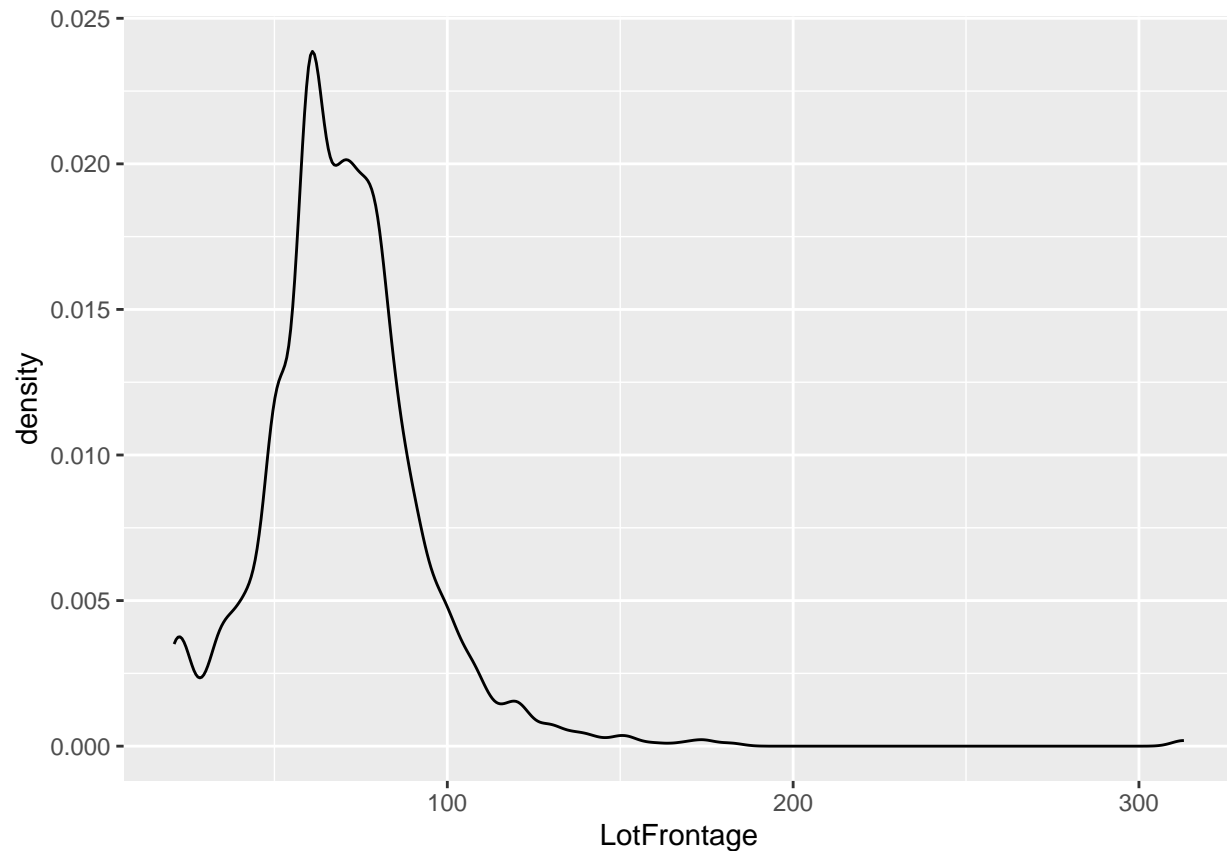
```
prop.table(table(df_train$LotFrontage, useNA = "ifany"))
```

```
##
##          21          24          30          32          33          34
## 0.0157534247 0.0130136986 0.0041095890 0.0034246575 0.0006849315 0.0068493151
##          35          36          37          38          39          40
## 0.0061643836 0.0041095890 0.0034246575 0.0006849315 0.0006849315 0.0082191781
##          41          42          43          44          45          46
## 0.0041095890 0.0027397260 0.0082191781 0.0061643836 0.0020547945 0.0006849315
##          47          48          49          50          51          52
## 0.0034246575 0.0041095890 0.0027397260 0.0390410959 0.0102739726 0.0095890411
##          53          54          55          56          57          58
## 0.0068493151 0.0041095890 0.0116438356 0.0034246575 0.0082191781 0.0047945205
##          59          60          61          62          63          64
## 0.0089041096 0.0979452055 0.0054794521 0.0061643836 0.0116438356 0.0130136986
##          65          66          67          68          69          70
## 0.0301369863 0.0102739726 0.0082191781 0.0130136986 0.0075342466 0.0479452055
```

```
##          71          72          73          74          75          76
## 0.0082191781 0.0116438356 0.0123287671 0.0102739726 0.0363013699 0.0075342466
##          77          78          79          80          81          82
## 0.0061643836 0.0171232877 0.0116438356 0.0472602740 0.0041095890 0.0082191781
##          83          84          85          86          87          88
## 0.0034246575 0.0061643836 0.0273972603 0.0068493151 0.0034246575 0.0068493151
##          89          90          91          92          93          94
## 0.0041095890 0.0157534247 0.0041095890 0.0068493151 0.0054794521 0.0041095890
##          95          96          97          98          99         100
## 0.0047945205 0.0054794521 0.0013698630 0.0054794521 0.0020547945 0.0109589041
##          101         102         103         104         105         106
## 0.0013698630 0.0027397260 0.0020547945 0.0020547945 0.0041095890 0.0006849315
##          107         108         109         110         111         112
## 0.0047945205 0.0020547945 0.0013698630 0.0041095890 0.0006849315 0.0006849315
##          114         115         116         118         120         121
## 0.0013698630 0.0013698630 0.0013698630 0.0013698630 0.0047945205 0.0013698630
##          122         124         128         129         130         134
## 0.0013698630 0.0013698630 0.0006849315 0.0013698630 0.0020547945 0.0013698630
##          137         138         140         141         144         149
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          150         152         153         160         168         174
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0013698630
##          182         313         <NA>
## 0.0006849315 0.0013698630 0.1773972603
```

```
ggplot(df_train, aes(x = LotFrontage)) + geom_density()
```

```
## Warning: Removed 259 rows containing non-finite values ('stat_density()').
```



```
#LotFrontage is heavily right skewed, therefore its NAs are imputed with its median
df_train$LotFrontage[is.na(df_train$LotFrontage)] <- mean(df_train$LotFrontage, na.rm=TRUE)
```

```
#Alley
prop.table(table(df_train$Alley, useNA = "ifany"))
```

```
##
##      Grvl      Pave      <NA>
## 0.03424658 0.02808219 0.93767123
```

```
df_train$Alley[is.na(df_train$Alley)] <- "None"
#test$Alley[is.na(test$Alley)] <- "None"

#MasVnrArea
prop.table(table(df_train$MasVnrArea, useNA = "ifany"))
```

```
##
##      0      1      11      14      16      18
## 0.5897260274 0.0013698630 0.0006849315 0.0006849315 0.0047945205 0.0013698630
##      22      24      27      28      30      31
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0013698630 0.0006849315
##      32      34      36      38      40      41
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0027397260 0.0006849315
##      42      44      45      46      48      50
## 0.0020547945 0.0020547945 0.0013698630 0.0006849315 0.0006849315 0.0020547945
```

##	51	53	54	56	57	60
##	0.0006849315	0.0006849315	0.0013698630	0.0006849315	0.0006849315	0.0013698630
##	63	64	65	66	67	68
##	0.0006849315	0.0006849315	0.0006849315	0.0013698630	0.0006849315	0.0013698630
##	70	72	74	75	76	80
##	0.0020547945	0.0054794521	0.0020547945	0.0013698630	0.0027397260	0.0041095890
##	81	82	84	85	86	88
##	0.0006849315	0.0020547945	0.0034246575	0.0020547945	0.0006849315	0.0006849315
##	89	90	92	94	95	96
##	0.0006849315	0.0020547945	0.0013698630	0.0013698630	0.0013698630	0.0006849315
##	97	98	99	100	101	102
##	0.0006849315	0.0013698630	0.0020547945	0.0027397260	0.0013698630	0.0006849315
##	104	105	106	108	109	110
##	0.0020547945	0.0013698630	0.0041095890	0.0054794521	0.0006849315	0.0020547945
##	112	113	114	115	116	117
##	0.0013698630	0.0013698630	0.0006849315	0.0006849315	0.0020547945	0.0013698630
##	119	120	122	123	125	126
##	0.0006849315	0.0047945205	0.0006849315	0.0006849315	0.0013698630	0.0006849315
##	127	128	130	132	135	136
##	0.0006849315	0.0020547945	0.0020547945	0.0034246575	0.0013698630	0.0020547945
##	137	138	140	142	143	144
##	0.0006849315	0.0006849315	0.0013698630	0.0006849315	0.0013698630	0.0013698630
##	145	146	147	148	149	150
##	0.0020547945	0.0006849315	0.0013698630	0.0020547945	0.0013698630	0.0006849315
##	151	153	154	156	157	158
##	0.0006849315	0.0013698630	0.0006849315	0.0006849315	0.0013698630	0.0013698630
##	160	161	162	163	164	165
##	0.0027397260	0.0006849315	0.0013698630	0.0006849315	0.0013698630	0.0006849315
##	166	167	168	169	170	171
##	0.0020547945	0.0006849315	0.0027397260	0.0020547945	0.0034246575	0.0013698630
##	172	174	175	176	178	180
##	0.0013698630	0.0020547945	0.0006849315	0.0020547945	0.0027397260	0.0054794521
##	182	183	184	186	188	189
##	0.0013698630	0.0027397260	0.0013698630	0.0020547945	0.0006849315	0.0013698630
##	192	194	196	200	202	203
##	0.0013698630	0.0006849315	0.0027397260	0.0041095890	0.0006849315	0.0013698630
##	204	205	206	207	208	209
##	0.0006849315	0.0013698630	0.0013698630	0.0006849315	0.0020547945	0.0006849315
##	210	212	215	216	218	219
##	0.0027397260	0.0013698630	0.0013698630	0.0027397260	0.0006849315	0.0006849315
##	220	223	224	225	226	228
##	0.0027397260	0.0006849315	0.0006849315	0.0006849315	0.0020547945	0.0006849315
##	230	232	233	234	236	237
##	0.0006849315	0.0013698630	0.0013698630	0.0006849315	0.0013698630	0.0006849315
##	238	240	243	244	245	246
##	0.0013698630	0.0020547945	0.0013698630	0.0006849315	0.0013698630	0.0027397260
##	247	248	250	252	254	255
##	0.0006849315	0.0006849315	0.0013698630	0.0027397260	0.0006849315	0.0006849315
##	256	258	259	260	261	262
##	0.0020547945	0.0006849315	0.0006849315	0.0013698630	0.0006849315	0.0006849315
##	263	266	268	270	272	274
##	0.0006849315	0.0013698630	0.0027397260	0.0027397260	0.0020547945	0.0006849315
##	275	278	280	281	284	285
##	0.0006849315	0.0006849315	0.0006849315	0.0013698630	0.0013698630	0.0013698630

```

##          286          287          288          289          290          292
## 0.0006849315 0.0006849315 0.0027397260 0.0013698630 0.0006849315 0.0006849315
##          293          294          295          296          297          298
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0013698630
##          299          300          302          304          305          306
## 0.0006849315 0.0027397260 0.0013698630 0.0013698630 0.0013698630 0.0013698630
##          309          310          312          315          318          320
## 0.0006849315 0.0006849315 0.0020547945 0.0006849315 0.0013698630 0.0034246575
##          324          328          333          335          336          337
## 0.0006849315 0.0013698630 0.0006849315 0.0013698630 0.0027397260 0.0006849315
##          338          340          342          344          348          350
## 0.0013698630 0.0041095890 0.0006849315 0.0013698630 0.0006849315 0.0020547945
##          351          359          360          361          362          365
## 0.0013698630 0.0006849315 0.0034246575 0.0006849315 0.0013698630 0.0006849315
##          366          368          370          375          376          378
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          380          381          387          388          391          396
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          399          408          410          412          415          420
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0020547945
##          423          424          425          426          428          432
## 0.0013698630 0.0013698630 0.0013698630 0.0006849315 0.0006849315 0.0006849315
##          435          436          438          442          443          448
## 0.0006849315 0.0006849315 0.0006849315 0.0013698630 0.0006849315 0.0006849315
##          450          451          452          456          459          464
## 0.0006849315 0.0006849315 0.0006849315 0.0027397260 0.0006849315 0.0006849315
##          466          468          472          473          479          480
## 0.0006849315 0.0006849315 0.0013698630 0.0006849315 0.0006849315 0.0013698630
##          481          491          500          506          510          513
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0013698630
##          528          530          541          554          562          564
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          567          571          573          576          579          584
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          594          600          603          604          616          621
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          630          632          640          650          651          653
## 0.0006849315 0.0006849315 0.0006849315 0.0013698630 0.0006849315 0.0006849315
##          660          664          673          705          731          748
## 0.0013698630 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          760          762          766          768          772          788
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          796          816          860          870          894          921
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          922          975          1031          1047          1115          1129
## 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
##          1170          1378          1600          <NA>
## 0.0006849315 0.0006849315 0.0006849315 0.0054794521

```

```

#MasVnrType
prop.table(table(df_train$MasVnrType, useNA = "ifany"))

```

```

##
##      BrkCmn      BrkFace      None      Stone      <NA>

```

```
## 0.010273973 0.304794521 0.591780822 0.087671233 0.005479452
```

```
#An NA MasVnrArea seems to tie with an NA MasVnrType, therefore its NAs are set to 0  
df_train$MasVnrType[is.na(df_train$MasVnrType)] <- "None"  
df_train$MasVnrArea[is.na(df_train$MasVnrArea)] <- 0
```

```
#####  
table(df_train$BsmtQual, useNA = "ifany")
```

```
##  
## Ex Fa Gd TA <NA>  
## 121 35 618 649 37
```

```
table(df_train$BsmtCond, useNA = "ifany")
```

```
##  
## Fa Gd Po TA <NA>  
## 45 65 2 1311 37
```

```
#BsmtQual  
prop.table(table(df_train$BsmtQual, useNA = "ifany"))
```

```
##  
## Ex Fa Gd TA <NA>  
## 0.08287671 0.02397260 0.42328767 0.44452055 0.02534247
```

```
df_train$BsmtQual[is.na(df_train$BsmtQual)] <- "None"
```

```
#BsmtCond  
prop.table(table(df_train$BsmtCond, useNA = "ifany"))
```

```
##  
## Fa Gd Po TA <NA>  
## 0.030821918 0.044520548 0.001369863 0.897945205 0.025342466
```

```
df_train$BsmtCond[is.na(df_train$BsmtCond)] <- "None"
```

```
#####
```

```
#BsmtExposure  
prop.table(table(df_train$BsmtExposure, useNA = "ifany"))
```

```
##  
## Av Gd Mn No <NA>  
## 0.15136986 0.09178082 0.07808219 0.65273973 0.02602740
```

```
df_train$BsmtExposure[is.na(df_train$BsmtExposure)] <- "None"
```

```
#BsmtFinType1  
prop.table(table(df_train$BsmtFinType1, useNA = "ifany"))
```

```
##
##      ALQ      BLQ      GLQ      LwQ      Rec      Unf      <NA>
## 0.15068493 0.10136986 0.28630137 0.05068493 0.09109589 0.29452055 0.02534247
```

```
df_train$BsmtFinType1[is.na(df_train$BsmtFinType1)] <- "None"
```

```
#BsmtFinType2
```

```
prop.table(table(df_train$BsmtFinType2, useNA = "ifany"))
```

```
##
##      ALQ      BLQ      GLQ      LwQ      Rec      Unf
## 0.013013699 0.022602740 0.009589041 0.031506849 0.036986301 0.860273973
##      <NA>
## 0.026027397
```

```
df_train$BsmtFinType2[is.na(df_train$BsmtFinType2)] <- "None"
```

```
#Electrical
```

```
prop.table(table(df_train$Electrical, useNA = "ifany"))
```

```
##
##      FuseA      FuseF      FuseP      Mix      SBrkr      <NA>
## 0.0643835616 0.0184931507 0.0020547945 0.0006849315 0.9136986301 0.0006849315
```

```
df_train$Electrical[is.na(df_train$Electrical)] <- "None"
```

```
#FireplaceQu
```

```
prop.table(table(df_train$FireplaceQu, useNA = "ifany"))
```

```
##
##      Ex      Fa      Gd      Po      TA      <NA>
## 0.01643836 0.02260274 0.26027397 0.01369863 0.21438356 0.47260274
```

```
df_train$FireplaceQu[is.na(df_train$FireplaceQu)] <- "None"
```

```
#GarageType
```

```
prop.table(table(df_train$GarageType, useNA = "ifany"))
```

```
##
##      2Types      Attchd      Basement      BuiltIn      CarPort      Detchd
## 0.004109589 0.595890411 0.013013699 0.060273973 0.006164384 0.265068493
##      <NA>
## 0.055479452
```

```
df_train$GarageType[is.na(df_train$GarageType)] <- "None"
```

```
#GarageYrBlt
```

```
prop.table(table(df_train$GarageYrBlt, useNA = "ifany"))
```



```
##
##      1900      1906      1908      1910      1914      1915
## 0.0006849315 0.0006849315 0.0006849315 0.0020547945 0.0013698630 0.0013698630
##      1916      1918      1920      1921      1922      1923
## 0.0034246575 0.0013698630 0.0095890411 0.0020547945 0.0034246575 0.0020547945
##      1924      1925      1926      1927      1928      1929
## 0.0020547945 0.0068493151 0.0041095890 0.0006849315 0.0027397260 0.0013698630
##      1930      1931      1932      1933      1934      1935
## 0.0054794521 0.0027397260 0.0020547945 0.0006849315 0.0013698630 0.0027397260
##      1936      1937      1938      1939      1940      1941
## 0.0034246575 0.0013698630 0.0020547945 0.0061643836 0.0095890411 0.0068493151
##      1942      1945      1946      1947      1948      1949
## 0.0013698630 0.0027397260 0.0027397260 0.0013698630 0.0075342466 0.0054794521
##      1950      1951      1952      1953      1954      1955
## 0.0164383562 0.0041095890 0.0020547945 0.0082191781 0.0130136986 0.0089041096
##      1956      1957      1958      1959      1960      1961
## 0.0109589041 0.0136986301 0.0143835616 0.0116438356 0.0130136986 0.0089041096
##      1962      1963      1964      1965      1966      1967
## 0.0143835616 0.0109589041 0.0123287671 0.0143835616 0.0143835616 0.0102739726
##      1968      1969      1970      1971      1972      1973
## 0.0178082192 0.0102739726 0.0136986301 0.0089041096 0.0095890411 0.0095890411
##      1974      1975      1976      1977      1978      1979
## 0.0123287671 0.0061643836 0.0198630137 0.0239726027 0.0130136986 0.0102739726
##      1980      1981      1982      1983      1984      1985
## 0.0102739726 0.0068493151 0.0027397260 0.0047945205 0.0054794521 0.0068493151
##      1986      1987      1988      1989      1990      1991
## 0.0041095890 0.0075342466 0.0095890411 0.0068493151 0.0109589041 0.0061643836
##      1992      1993      1994      1995      1996      1997
## 0.0089041096 0.0150684932 0.0123287671 0.0123287671 0.0136986301 0.0130136986
##      1998      1999      2000      2001      2002      2003
## 0.0212328767 0.0205479452 0.0184931507 0.0136986301 0.0178082192 0.0342465753
##      2004      2005      2006      2007      2008      2009
## 0.0363013699 0.0445205479 0.0404109589 0.0335616438 0.0198630137 0.0143835616
##      2010      <NA>
## 0.0020547945 0.0554794521
```

```
#An NA Garage Year Built seems to tie with an NA Garage Type, therefore its NAs are set to 0
#No need to address GarageType NAs since the variable is not numeric
df_train$GarageYrBlt[is.na(df_train$GarageYrBlt)] <- 0
```

```
#GarageFinish
prop.table(table(df_train$GarageFinish, useNA = "ifany"))
```

```
##
##      Fin      RFn      Unf      <NA>
## 0.24109589 0.28904110 0.41438356 0.05547945
```

```
df_train$GarageFinish[is.na(df_train$GarageFinish)] <- "None"
```

```
#####
table(df_train$GarageQual, useNA = "ifany")
```

```
##
```

```
##      Ex      Fa      Gd      Po      TA <NA>
##      3      48      14      3 1311      81
```

```
table(df_train$GarageCond, useNA = "ifany")
```

```
##
##      Ex      Fa      Gd      Po      TA <NA>
##      2      35      9      7 1326      81
```

```
#GarageQual
prop.table(table(df_train$GarageQual, useNA = "ifany"))
```

```
##
##              Ex              Fa              Gd              Po              TA              <NA>
## 0.002054795 0.032876712 0.009589041 0.002054795 0.897945205 0.055479452
```

```
df_train$GarageQual[is.na(df_train$GarageQual)] <- "None"
```

```
#GarageCond
prop.table(table(df_train$GarageCond, useNA = "ifany"))
```

```
##
##              Ex              Fa              Gd              Po              TA              <NA>
## 0.001369863 0.023972603 0.006164384 0.004794521 0.908219178 0.055479452
```

```
df_train$GarageCond[is.na(df_train$GarageCond)] <- "None"
#####
```

```
#PoolQC
prop.table(table(df_train$PoolQC, useNA = "ifany"))
```

```
##
##              Ex              Fa              Gd              <NA>
## 0.001369863 0.001369863 0.002054795 0.995205479
```

```
df_train$PoolQC[is.na(df_train$PoolQC)] <- "None"
```

```
#Fence
prop.table(table(df_train$Fence, useNA = "ifany"))
```

```
##
##      GdPrv      GdWo      MnPrv      MnWw      <NA>
## 0.040410959 0.036986301 0.107534247 0.007534247 0.807534247
```

```
df_train$Fence[is.na(df_train$Fence)] <- "None"
```

```
#MiscFeature
prop.table(table(df_train$MiscFeature, useNA = "ifany"))
```

```
##
##          Gar2          Othr          Shed          TenC          <NA>
## 0.0013698630 0.0013698630 0.0335616438 0.0006849315 0.9630136986
```

```
df_train$MiscFeature[is.na(df_train$MiscFeature)] <- "None"

colSums(is.na(df_train))
```

```
##          Id  MSSubClass  MSZoning  LotFrontage  LotArea
##          0           0           0           0           0
##      Street      Alley      LotShape  LandContour  Utilities
##          0           0           0           0           0
##      LotConfig  LandSlope  Neighborhood  Condition1  Condition2
##          0           0           0           0           0
##      BldgType  HouseStyle  OverallQual  OverallCond  YearBuilt
##          0           0           0           0           0
##  YearRemodAdd  RoofStyle  RoofMatl  Exterior1st  Exterior2nd
##          0           0           0           0           0
##      MasVnrType  MasVnrArea  ExterQual  ExterCond  Foundation
##          0           0           0           0           0
##      BsmtQual  BsmtCond  BsmtExposure  BsmtFinType1  BsmtFinSF1
##          0           0           0           0           0
##  BsmtFinType2  BsmtFinSF2  BsmtUnfSF  TotalBsmtSF  Heating
##          0           0           0           0           0
##      HeatingQC  CentralAir  Electrical  X1stFlrSF  X2ndFlrSF
##          0           0           0           0           0
##  LowQualFinSF  GrLivArea  BsmtFullBath  BsmtHalfBath  FullBath
##          0           0           0           0           0
##      HalfBath  BedroomAbvGr  KitchenAbvGr  KitchenQual  TotRmsAbvGrd
##          0           0           0           0           0
##      Functional  Fireplaces  FireplaceQu  GarageType  GarageYrBlt
##          0           0           0           0           0
##  GarageFinish  GarageCars  GarageArea  GarageQual  GarageCond
##          0           0           0           0           0
##      PavedDrive  WoodDeckSF  OpenPorchSF  EnclosedPorch  X3SsnPorch
##          0           0           0           0           0
##      ScreenPorch  PoolArea  PoolQC  Fence  MiscFeature
##          0           0           0           0           0
##      MiscVal  MoSold  YrSold  SaleType  SaleCondition
##          0           0           0           0           0
##      SalePrice
##          0
```

```
####train: Factoring columns according to data_description.txt####
```

```
#Seeking out numeric and non-numeric columns
str(df_train)
```

```
## 'data.frame': 1460 obs. of 81 variables:
## $ Id : int 1 2 3 4 5 6 7 8 9 10 ...
## $ MSSubClass : int 60 20 60 70 60 50 20 60 50 190 ...
## $ MSZoning : chr "RL" "RL" "RL" "RL" ...
## $ LotFrontage : num 65 80 68 60 84 ...
```

```

## $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
## $ Street       : chr  "Pave" "Pave" "Pave" "Pave" ...
## $ Alley        : chr  "None" "None" "None" "None" ...
## $ LotShape     : chr  "Reg" "Reg" "IR1" "IR1" ...
## $ LandContour  : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
## $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
## $ LotConfig    : chr  "Inside" "FR2" "Inside" "Corner" ...
## $ LandSlope    : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
## $ Neighborhood : chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
## $ Condition1   : chr  "Norm" "Feedr" "Norm" "Norm" ...
## $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
## $ BldgType     : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
## $ HouseStyle   : chr  "2Story" "1Story" "2Story" "2Story" ...
## $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
## $ OverallCond  : chr  "5" "8" "5" "5" ...
## $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
## $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
## $ RoofStyle    : chr  "Gable" "Gable" "Gable" "Gable" ...
## $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
## $ Exterior1st  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
## $ Exterior2nd  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
## $ MasVnrType   : chr  "BrkFace" "None" "BrkFace" "None" ...
## $ MasVnrArea   : num  196 0 162 0 350 0 186 240 0 0 ...
## $ ExterQual    : chr  "Gd" "TA" "Gd" "TA" ...
## $ ExterCond    : chr  "TA" "TA" "TA" "TA" ...
## $ Foundation   : chr  "PConc" "CBlock" "PConc" "BrkTil" ...
## $ BsmtQual     : chr  "Gd" "Gd" "Gd" "TA" ...
## $ BsmtCond     : chr  "TA" "TA" "TA" "Gd" ...
## $ BsmtExposure : chr  "No" "Gd" "Mn" "No" ...
## $ BsmtFinType1 : chr  "GLQ" "ALQ" "GLQ" "ALQ" ...
## $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
## $ BsmtFinType2 : chr  "Unf" "Unf" "Unf" "Unf" ...
## $ BsmtFinSF2   : int  0 0 0 0 0 0 32 0 0 ...
## $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
## $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
## $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
## $ HeatingQC    : chr  "Ex" "Ex" "Ex" "Gd" ...
## $ CentralAir   : chr  "Y" "Y" "Y" "Y" ...
## $ Electrical   : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
## $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
## $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
## $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 ...
## $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
## $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
## $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
## $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
## $ KitchenQual  : chr  "Gd" "TA" "Gd" "Gd" ...
## $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
## $ Functional   : chr  "Typ" "Typ" "Typ" "Typ" ...
## $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
## $ FireplaceQu  : chr  "None" "TA" "TA" "Gd" ...

```

```
## $ GarageType : chr "Attchd" "Attchd" "Attchd" "Detchd" ...
## $ GarageYrBlt : num 2003 1976 2001 1998 2000 ...
## $ GarageFinish : chr "RFn" "RFn" "RFn" "Unf" ...
## $ GarageCars : int 2 2 2 3 3 2 2 2 1 ...
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...
## $ GarageQual : chr "TA" "TA" "TA" "TA" ...
## $ GarageCond : chr "TA" "TA" "TA" "TA" ...
## $ PavedDrive : chr "Y" "Y" "Y" "Y" ...
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...
## $ EnclosedPorch : int 0 0 0 272 0 0 0 228 205 0 ...
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolQC : chr "None" "None" "None" "None" ...
## $ Fence : chr "None" "None" "None" "None" ...
## $ MiscFeature : chr "None" "None" "None" "None" ...
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...
## $ MoSold : int 2 5 9 2 12 10 8 11 4 1 ...
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
## $ SaleType : chr "WD" "WD" "WD" "WD" ...
## $ SaleCondition: chr "Normal" "Normal" "Normal" "Abnorml" ...
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

```
names(df_train)[sapply(df_train, is.numeric)]
```

```
## [1] "Id" "MSSubClass" "LotFrontage" "LotArea"
## [5] "OverallQual" "YearBuilt" "YearRemodAdd" "MasVnrArea"
## [9] "BsmtFinSF1" "BsmtFinSF2" "BsmtUnfSF" "TotalBsmtSF"
## [13] "X1stFlrSF" "X2ndFlrSF" "LowQualFinSF" "GrLivArea"
## [17] "BsmtFullBath" "BsmtHalfBath" "FullBath" "HalfBath"
## [21] "BedroomAbvGr" "KitchenAbvGr" "TotRmsAbvGrd" "Fireplaces"
## [25] "GarageYrBlt" "GarageCars" "GarageArea" "WoodDeckSF"
## [29] "OpenPorchSF" "EnclosedPorch" "X3SsnPorch" "ScreenPorch"
## [33] "PoolArea" "MiscVal" "MoSold" "YrSold"
## [37] "SalePrice"
```

```
names(df_train)[sapply(df_train, is.character)]
```

```
## [1] "MSZoning" "Street" "Alley" "LotShape"
## [5] "LandContour" "Utilities" "LotConfig" "LandSlope"
## [9] "Neighborhood" "Condition1" "Condition2" "BldgType"
## [13] "HouseStyle" "OverallCond" "RoofStyle" "RoofMatl"
## [17] "Exterior1st" "Exterior2nd" "MasVnrType" "ExterQual"
## [21] "ExterCond" "Foundation" "BsmtQual" "BsmtCond"
## [25] "BsmtExposure" "BsmtFinType1" "BsmtFinType2" "Heating"
## [29] "HeatingQC" "CentralAir" "Electrical" "KitchenQual"
## [33] "Functional" "FireplaceQu" "GarageType" "GarageFinish"
## [37] "GarageQual" "GarageCond" "PavedDrive" "PoolQC"
## [41] "Fence" "MiscFeature" "SaleType" "SaleCondition"
```

```

#Character variables into factors
df_train[sapply(df_train, is.character)] <- lapply(df_train[sapply(df_train, is.character)], as.factor)
#This also applies for columns with both chars and nums, which is still
#consistent with the actual meaning of such columns that are meant for factoring

#Sanity check
str(df_train[sapply(df_train, is.character)])

```

```
## 'data.frame':    1460 obs. of  0 variables
```

```
str(df_train[sapply(df_train, is.numeric)])
```

```

## 'data.frame':    1460 obs. of  37 variables:
## $ Id          : int  1 2 3 4 5 6 7 8 9 10 ...
## $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
## $ LotFrontage  : num  65 80 68 60 84 ...
## $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
## $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
## $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
## $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
## $ MasVnrArea   : num  196 0 162 0 350 0 186 240 0 0 ...
## $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
## $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
## $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
## $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
## $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
## $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
## $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
## $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
## $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
## $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
## $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
## $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
## $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
## $ GarageYrBlt  : num  2003 1976 2001 1998 2000 ...
## $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
## $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
## $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
## $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
## $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
## $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
## $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
## $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
## $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
## $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
## $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
## $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

```

```
#Numeric columns at this point mistakenly includes obscure factor-able columns
#like MSSubClass, OverallQual, OverallCond
#Let's try and fix that
```

```
#MSSubClass
```

```
df_train$MSSubClass <- as.factor(df_train$MSSubClass)
str(df_train$MSSubClass)
```

```
## Factor w/ 15 levels "20","30","40",...: 6 1 6 7 6 5 1 6 5 15 ...
```

```
table(df_train$MSSubClass)
```

```
##
## 20 30 40 45 50 60 70 75 80 85 90 120 160 180 190
## 536 69 4 12 144 299 60 16 58 20 52 87 63 10 30
```

```
#OverallQual
```

```
df_train$OverallQual <- as.factor(df_train$OverallQual)
str(df_train$OverallQual)
```

```
## Factor w/ 10 levels "1","2","3","4",...: 7 6 7 7 8 5 8 7 7 5 ...
```

```
#OverallCond
```

```
df_train$OverallCond <- as.factor(df_train$OverallCond)
str(df_train$OverallCond)
```

```
## Factor w/ 10 levels "1","2","3","4",...: 5 8 5 5 5 5 5 6 5 6 ...
```

```
#Sanity check 2
```

```
str(df_train[sapply(df_train, is.numeric)])
```

```
## 'data.frame': 1460 obs. of 35 variables:
## $ Id : int 1 2 3 4 5 6 7 8 9 10 ...
## $ LotFrontage : num 65 80 68 60 84 ...
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
## $ YearBuilt : int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
## $ YearRemodAdd : int 2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
## $ MasVnrArea : num 196 0 162 0 350 0 186 240 0 0 ...
## $ BsmtFinSF1 : int 706 978 486 216 655 732 1369 859 0 851 ...
## $ BsmtFinSF2 : int 0 0 0 0 0 0 0 32 0 0 ...
## $ BsmtUnfSF : int 150 284 434 540 490 64 317 216 952 140 ...
## $ TotalBsmtSF : int 856 1262 920 756 1145 796 1686 1107 952 991 ...
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
## $ BsmtFullBath : int 1 0 1 1 1 1 1 0 1 ...
## $ BsmtHalfBath : int 0 1 0 0 0 0 0 0 0 ...
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...
## $ HalfBath : int 1 0 1 0 1 1 0 1 0 0 ...
```

```
## $ BedroomAbvGr : int 3 3 3 3 4 1 3 3 2 2 ...
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 2 2 ...
## $ TotRmsAbvGrd : int 8 6 6 7 9 5 7 7 8 5 ...
## $ Fireplaces   : int 0 1 1 1 1 0 1 2 2 2 ...
## $ GarageYrBlt  : num 2003 1976 2001 1998 2000 ...
## $ GarageCars   : int 2 2 2 3 3 2 2 2 2 1 ...
## $ GarageArea   : int 548 460 608 642 836 480 636 484 468 205 ...
## $ WoodDeckSF   : int 0 298 0 0 192 40 255 235 90 0 ...
## $ OpenPorchSF  : int 61 0 42 35 84 30 57 204 0 4 ...
## $ EnclosedPorch: int 0 0 0 272 0 0 0 228 205 0 ...
## $ X3SsnPorch   : int 0 0 0 0 0 320 0 0 0 0 ...
## $ ScreenPorch  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolArea     : int 0 0 0 0 0 0 0 0 0 0 ...
## $ MiscVal      : int 0 0 0 0 0 700 0 350 0 0 ...
## $ MoSold       : int 2 5 9 2 12 10 8 11 4 1 ...
## $ YrSold       : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
## $ SalePrice    : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

```
str(df_train[sapply(df_train, is.factor)])
```

```
## 'data.frame': 1460 obs. of 46 variables:
## $ MSSubClass : Factor w/ 15 levels "20","30","40",...: 6 1 6 7 6 5 1 6 5 15 ...
## $ MSZoning : Factor w/ 5 levels "C (all)","FV",...: 4 4 4 4 4 4 4 4 5 4 ...
## $ Street : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 ...
## $ Alley : Factor w/ 3 levels "Grvl","None",...: 2 2 2 2 2 2 2 2 2 ...
## $ LotShape : Factor w/ 4 levels "IR1","IR2","IR3",...: 4 4 1 1 1 1 1 4 1 4 ...
## $ LandContour : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 4 4 4 4 4 4 ...
## $ Utilities : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
## $ LotConfig : Factor w/ 5 levels "Corner","CulDSac",...: 5 3 5 1 3 5 5 1 5 1 ...
## $ LandSlope : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",...: 6 25 6 7 14 12 21 17 18 4 ...
## $ Condition1 : Factor w/ 9 levels "Artery","Feedr",...: 3 2 3 3 3 3 3 5 1 ...
## $ Condition2 : Factor w/ 8 levels "Artery","Feedr",...: 3 3 3 3 3 3 3 3 1 ...
## $ BldgType : Factor w/ 5 levels "1Fam","2fmCon",...: 1 1 1 1 1 1 1 1 1 2 ...
## $ HouseStyle : Factor w/ 8 levels "1.5Fin","1.5Unf",...: 6 3 6 6 6 1 3 6 1 2 ...
## $ OverallQual : Factor w/ 10 levels "1","2","3","4",...: 7 6 7 7 8 5 8 7 7 5 ...
## $ OverallCond : Factor w/ 10 levels "1","2","3","4",...: 5 8 5 5 5 5 5 6 5 6 ...
## $ RoofStyle : Factor w/ 6 levels "Flat","Gable",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ RoofMatl : Factor w/ 8 levels "ClyTile","CompShg",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ Exterior1st : Factor w/ 15 levels "AsbShng","AsphShn",...: 13 9 13 14 13 13 13 7 4 9 ...
## $ Exterior2nd : Factor w/ 16 levels "AsbShng","AsphShn",...: 14 9 14 16 14 14 14 7 16 9 ...
## $ MasVnrType : Factor w/ 4 levels "BrkCmn","BrkFace",...: 2 3 2 3 2 3 4 4 3 3 ...
## $ ExterQual : Factor w/ 4 levels "Ex","Fa","Gd",...: 3 4 3 4 3 4 3 4 4 4 ...
## $ ExterCond : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ Foundation : Factor w/ 6 levels "BrkTil","CBlock",...: 3 2 3 1 3 6 3 2 1 1 ...
## $ BsmtQual : Factor w/ 5 levels "Ex","Fa","Gd",...: 3 3 3 5 3 3 1 3 5 5 ...
## $ BsmtCond : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 2 5 5 5 5 5 5 ...
## $ BsmtExposure : Factor w/ 5 levels "Av","Gd","Mn",...: 4 2 3 4 1 4 1 3 4 4 ...
## $ BsmtFinType1 : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 3 1 3 1 3 3 3 1 7 3 ...
## $ BsmtFinType2 : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 7 7 7 7 7 7 7 2 7 7 ...
## $ Heating : Factor w/ 6 levels "Floor","GasA",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ HeatingQC : Factor w/ 5 levels "Ex","Fa","Gd",...: 1 1 1 3 1 1 1 1 3 1 ...
## $ CentralAir : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
## $ Electrical : Factor w/ 6 levels "FuseA","FuseF",...: 6 6 6 6 6 6 6 6 2 6 ...
```



```
## $ KitchenQual : Factor w/ 4 levels "Ex","Fa","Gd",...: 3 4 3 3 3 4 3 4 4 4 ...
## $ Functional : Factor w/ 7 levels "Maj1","Maj2",...: 7 7 7 7 7 7 7 3 7 ...
## $ FireplaceQu : Factor w/ 6 levels "Ex","Fa","Gd",...: 4 6 6 3 6 4 3 6 6 6 ...
## $ GarageType : Factor w/ 7 levels "2Types","Attchd",...: 2 2 2 6 2 2 2 2 6 2 ...
## $ GarageFinish : Factor w/ 4 levels "Fin","None","RFn",...: 3 3 3 4 3 4 3 3 4 3 ...
## $ GarageQual : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 2 3 ...
## $ GarageCond : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 6 6 ...
## $ PavedDrive : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
## $ PoolQC : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 4 4 4 4 4 4 4 ...
## $ Fence : Factor w/ 5 levels "GdPrv","GdWo",...: 5 5 5 5 5 3 5 5 5 5 ...
## $ MiscFeature : Factor w/ 5 levels "Gar2","None",...: 2 2 2 2 2 4 2 4 2 2 ...
## $ SaleType : Factor w/ 9 levels "COD","Con","ConLD",...: 9 9 9 9 9 9 9 9 9 9 ...
## $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",...: 5 5 5 1 5 5 5 5 1 5 ...
```

```
####train: Removing Columns####
```

```
#Redundant means a value representing over 99% of the data in a column
```

```
#Utilities
```

```
prop.table(table(df_train$Utilities, useNA = "ifany"))#["1"]
```

```
##
## AllPub NoSeWa
## 0.9993150685 0.0006849315
```

```
#PoolArea
```

```
prop.table(table(df_train$PoolArea, useNA = "ifany"))
```

```
##
## 0 480 512 519 555 576
## 0.9952054795 0.0006849315 0.0006849315 0.0006849315 0.0006849315 0.0006849315
## 648 738
## 0.0006849315 0.0006849315
```

```
df_train[,c("Utilities", "PoolArea", "GrLivArea", "TotalBsmtSF")] <- list(NULL)
```

```
####train: EDA####
```

```
model0 = lm(log(SalePrice)~., data = df_train)
paste(summary(model0)$r.squared, " | ", summary(model0)$adj.r.squared)
```

```
## [1] "0.947047289700748 | 0.93458255349144"
```

```
# R-Squared = 0.944680793010884
#Adjusted R-Squared = 0.932797066613555"
```

```
#Residuals QQ Plot
```

```
residuals = resid(model0)
p1 = ggplot(df_train, aes(sample = residuals)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  labs(title = "QQ Plot of Residuals", x = "Theoretical Quantile", y = "Actual Quantile")
```

```

#Residuals Histogram
p2 = ggplot(df_train, aes(residuals)) +
  geom_histogram(aes(y = ..density..), bins = 30) +
  geom_density(alpha = .2, color = "red", fill = "azure") +
  labs(title = "Histogram of Residuals", x = "Residuals", y = "Density")

#Cook's Distance Plot
library(lindia)
p3 = gg_cooksd(model0)

#Neighborhood vs RStudent
stdres2 <- rstandard(model0)
p4 = ggplot(df_train, aes(as.factor(Neighborhood), stdres2)) +
  geom_boxplot() +
  labs(title = "RStudent Boxplot", x = "Neighborhood", y = "RStudent")

#Standardized Residuals Plot
p5 = ggplot(df_train, aes(x = seq(stdres2), y = stdres2)) +
  geom_point() +
  geom_hline(yintercept = 3, color = "red") +
  geom_hline(yintercept = -3, color = "red") +
  labs(title = "Prediction vs RStudent", x = "Predicted Value", y = "RStudent")

#DFFITS
p6 = ggplot(df_train, aes(x = seq(dffits(model0)), y = dffits(model0))) +
  geom_point() +
  geom_hline(color="red", yintercept=0) +
  labs(title = "DFFITS", x = "Observation Number", y = "DFFITS")
ylim(-5,5)

```

```

## <ScaleContinuousPosition>
## Range:
## Limits: -5 -- 5

```

```

grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 3)

```

```

## Warning: The dot-dot notation ('..density..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(density)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

```

## Warning: Removed 27 rows containing missing values ('geom_point()').

```

```

## Warning: Removed 27 rows containing missing values ('geom_segment()').

```

```

## Warning: Removed 27 rows containing missing values ('geom_text()').

```

```

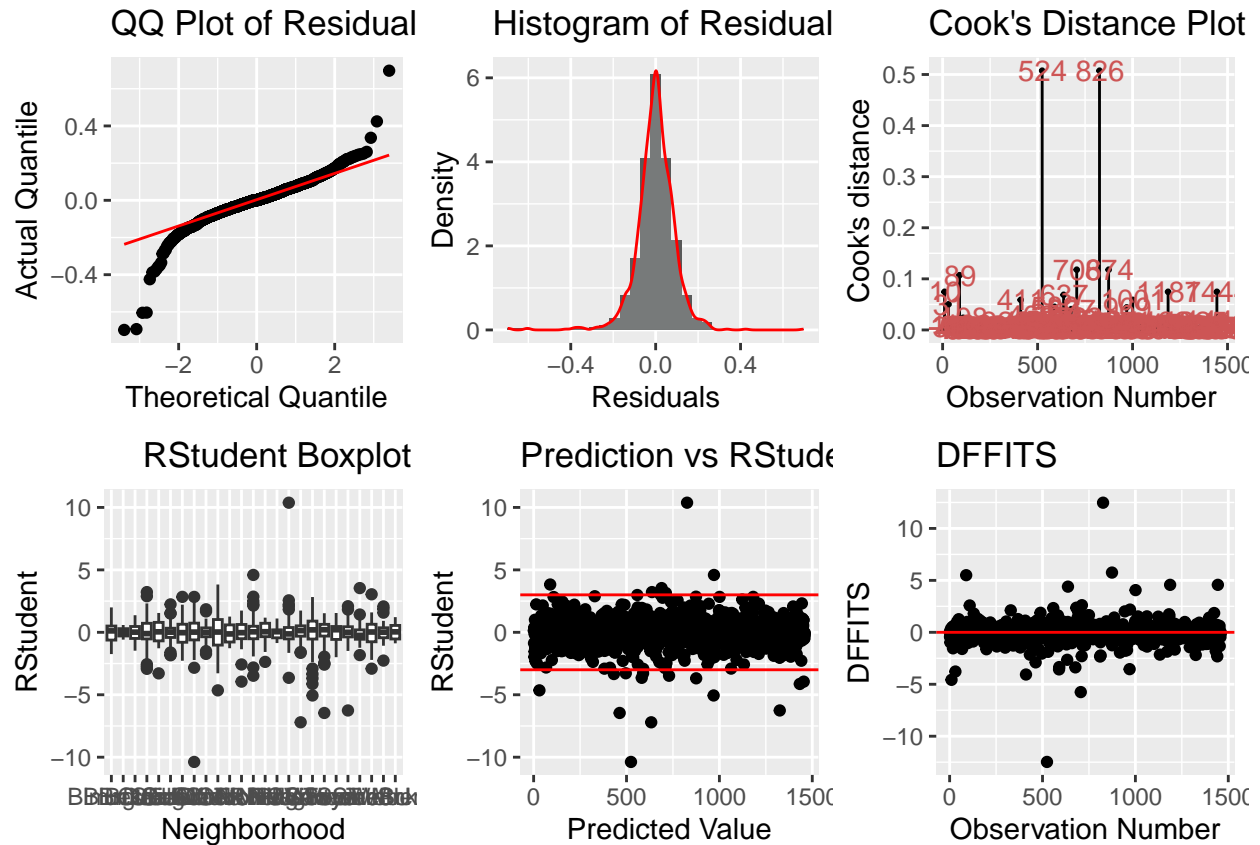
## Warning: Removed 27 rows containing non-finite values ('stat_boxplot()').

```

```

## Warning: Removed 27 rows containing missing values ('geom_point()').
## Removed 27 rows containing missing values ('geom_point()').

```

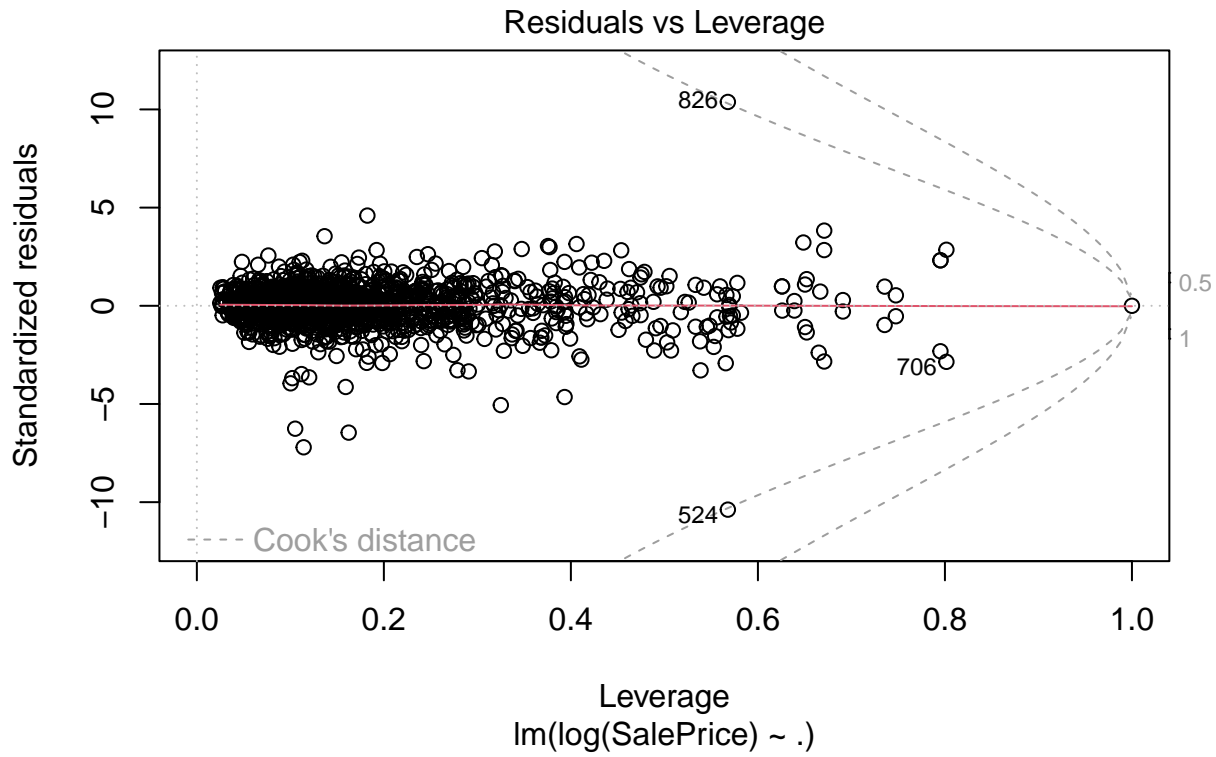


```
#Standardized Residuals vs Leverage
plot(model0, which = 5)
```

```
## Warning: not plotting observations with leverage one:
## 11, 121, 186, 251, 272, 326, 333, 347, 376, 399, 534, 584, 667, 811, 822, 949, 1004, 1012, 1188, 1...

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```



```
####train: Modeling with outliers####
```

```
#Identify influential points with Cook's D > 0.1
```

```
as.numeric(names(cooks.distance(model0))[(cooks.distance(model0) > 0.1)])
```

```
## [1] NA 89 NA NA NA NA NA NA NA NA NA 524 NA NA NA 706 NA NA 826
## [20] 874 NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
#And they are 89, 524, 826
```

```
paste(summary(model0)$r.squared, " | ", summary(model0)$adj.r.squared)
```

```
## [1] "0.947047289700748 | 0.93458255349144"
```

```
#Start removing outliers one by one and a combo of outliers
```

```
#And find the most desired R-Squared and Adj R-Squared
```

```
A2Data01 = df_train[-89,]
```

```
model01 <- lm(log(SalePrice)~., data = A2Data01)
```

```
paste(summary(model01)$r.squared, " | ", summary(model01)$adj.r.squared)
```

```
## [1] "0.947602261263939 | 0.93525770925663"
```

```
A2Data02 = df_train[-524,]
model02 <- lm(log(SalePrice)~., data = A2Data02)
paste(summary(model02)$r.squared, " | ", summary(model02)$adj.r.squared)
```

```
## [1] "0.951875528976546 | 0.94053772987102"
```

```
A2Data03 = df_train[-826,]
model03 <- lm(log(SalePrice)~., data = A2Data03)
paste(summary(model03)$r.squared, " | ", summary(model03)$adj.r.squared)
```

```
## [1] "0.951732377578162 | 0.940360852973695"
```

```
A2Data04 = df_train[-c(89, 524, 826),]
model04 <- lm(log(SalePrice)~., data = A2Data04)
paste(summary(model04)$r.squared, " | ", summary(model04)$adj.r.squared)
```

```
## [1] "0.952273180152351 | 0.941060008737763"
```

```
A2Data05 = df_train[-c(89, 524),]
model05 <- lm(log(SalePrice)~., data = A2Data05)
paste(summary(model05)$r.squared, " | ", summary(model05)$adj.r.squared)
```

```
## [1] "0.952417038132603 | 0.94119730666599"
```

```
A2Data06 = df_train[-c(89, 826),]
model06 <- lm(log(SalePrice)~., data = A2Data06)
paste(summary(model06)$r.squared, " | ", summary(model06)$adj.r.squared)
```

```
## [1] "0.952275359817096 | 0.941022221589066"
```

```
A2Data07 = df_train[-c(524, 826),]
model07 <- lm(log(SalePrice)~., data = A2Data07)
paste(summary(model07)$r.squared, " | ", summary(model07)$adj.r.squared)
```

```
## [1] "0.951730157619736 | 0.9403990166542"
```

```
#The most desired is A2Data05 and model05
A2Data = A2Data05
model1 = model05
paste(summary(model1)$r.squared, " | ", summary(model1)$adj.r.squared)
```

```
## [1] "0.952417038132603 | 0.94119730666599"
```

```
# R-Squared = 0.952373329054498
#Adjusted R-Squared = 0.941193169857969
```

```
####test: Dealing with NAs####
table(is.na(df_test))
```

```
##
## FALSE TRUE
## 109720 7000
```

```
colSums(is.na(df_test))
```

```
##      Id  MSSubClass  MSZoning  LotFrontage  LotArea
##      0         0         4         227         0
##      Street      Alley      LotShape  LandContour  Utilities
##      0         1352         0         0         2
##      LotConfig  LandSlope  Neighborhood  Condition1  Condition2
##      0         0         0         0         0
##      BldgType  HouseStyle  OverallQual  OverallCond  YearBuilt
##      0         0         0         0         0
##      YearRemodAdd  RoofStyle  RoofMatl  Exterior1st  Exterior2nd
##      0         0         0         1         1
##      MasVnrType  MasVnrArea  ExterQual  ExterCond  Foundation
##      16         15         0         0         0
##      BsmtQual  BsmtCond  BsmtExposure  BsmtFinType1  BsmtFinSF1
##      44         45         44         42         1
##      BsmtFinType2  BsmtFinSF2  BsmtUnfSF  TotalBsmtSF  Heating
##      42         1         1         1         0
##      HeatingQC  CentralAir  Electrical  X1stFlrSF  X2ndFlrSF
##      0         0         0         0         0
##      LowQualFinSF  GrLivArea  BsmtFullBath  BsmtHalfBath  FullBath
##      0         0         2         2         0
##      HalfBath  BedroomAbvGr  KitchenAbvGr  KitchenQual  TotRmsAbvGrd
##      0         0         0         1         0
##      Functional  Fireplaces  FireplaceQu  GarageType  GarageYrBlt
##      2         0         730         76         78
##      GarageFinish  GarageCars  GarageArea  GarageQual  GarageCond
##      78         1         1         78         78
##      PavedDrive  WoodDeckSF  OpenPorchSF  EnclosedPorch  X3SsnPorch
##      0         0         0         0         0
##      ScreenPorch  PoolArea  PoolQC  Fence  MiscFeature
##      0         0         1456         1169         1408
##      MiscVal  MoSold  YrSold  SaleType  SaleCondition
##      0         0         0         1         0
```

```
names(df_test)[sapply(df_test, anyNA)]
```

```
## [1] "MSZoning" "LotFrontage" "Alley" "Utilities" "Exterior1st"
## [6] "Exterior2nd" "MasVnrType" "MasVnrArea" "BsmtQual" "BsmtCond"
## [11] "BsmtExposure" "BsmtFinType1" "BsmtFinSF1" "BsmtFinType2" "BsmtFinSF2"
## [16] "BsmtUnfSF" "TotalBsmtSF" "BsmtFullBath" "BsmtHalfBath" "KitchenQual"
## [21] "Functional" "FireplaceQu" "GarageType" "GarageYrBlt" "GarageFinish"
## [26] "GarageCars" "GarageArea" "GarageQual" "GarageCond" "PoolQC"
## [31] "Fence" "MiscFeature" "SaleType"
```

```
str(df_test)
```

```
## 'data.frame':    1459 obs. of  80 variables:
## $ Id             : int  1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 ...
## $ MSSubClass     : int  20 20 60 60 120 60 20 60 20 20 ...
## $ MSZoning       : chr   "RH" "RL" "RL" "RL" ...
## $ LotFrontage    : int  80 81 74 78 43 75 NA 63 85 70 ...
## $ LotArea        : int 11622 14267 13830 9978 5005 10000 7980 8402 10176 8400 ...
## $ Street         : chr   "Pave" "Pave" "Pave" "Pave" ...
## $ Alley          : chr   NA NA NA NA ...
## $ LotShape       : chr   "Reg" "IR1" "IR1" "IR1" ...
## $ LandContour    : chr   "Lvl" "Lvl" "Lvl" "Lvl" ...
## $ Utilities      : chr   "AllPub" "AllPub" "AllPub" "AllPub" ...
## $ LotConfig      : chr   "Inside" "Corner" "Inside" "Inside" ...
## $ LandSlope      : chr   "Gtl" "Gtl" "Gtl" "Gtl" ...
## $ Neighborhood   : chr   "Names" "Names" "Gilbert" "Gilbert" ...
## $ Condition1     : chr   "Feedr" "Norm" "Norm" "Norm" ...
## $ Condition2     : chr   "Norm" "Norm" "Norm" "Norm" ...
## $ BldgType       : chr   "1Fam" "1Fam" "1Fam" "1Fam" ...
## $ HouseStyle     : chr   "1Story" "1Story" "2Story" "2Story" ...
## $ OverallQual    : int  5 6 5 6 8 6 6 6 7 4 ...
## $ OverallCond    : int  6 6 5 6 5 5 7 5 5 5 ...
## $ YearBuilt      : int  1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...
## $ YearRemodAdd   : int  1961 1958 1998 1998 1992 1994 2007 1998 1990 1970 ...
## $ RoofStyle      : chr   "Gable" "Hip" "Gable" "Gable" ...
## $ RoofMatl       : chr   "CompShg" "CompShg" "CompShg" "CompShg" ...
## $ Exterior1st    : chr   "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...
## $ Exterior2nd    : chr   "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...
## $ MasVnrType     : chr   "None" "BrkFace" "None" "BrkFace" ...
## $ MasVnrArea     : int  0 108 0 20 0 0 0 0 0 0 ...
## $ ExterQual      : chr   "TA" "TA" "TA" "TA" ...
## $ ExterCond      : chr   "TA" "TA" "TA" "TA" ...
## $ Foundation     : chr   "CBlock" "CBlock" "PConc" "PConc" ...
## $ BsmtQual       : chr   "TA" "TA" "Gd" "TA" ...
## $ BsmtCond       : chr   "TA" "TA" "TA" "TA" ...
## $ BsmtExposure   : chr   "No" "No" "No" "No" ...
## $ BsmtFinType1   : chr   "Rec" "ALQ" "GLQ" "GLQ" ...
## $ BsmtFinSF1     : int  468 923 791 602 263 0 935 0 637 804 ...
## $ BsmtFinType2   : chr   "LwQ" "Unf" "Unf" "Unf" ...
## $ BsmtFinSF2     : int  144 0 0 0 0 0 0 0 0 78 ...
## $ BsmtUnfSF      : int  270 406 137 324 1017 763 233 789 663 0 ...
## $ TotalBsmtSF    : int  882 1329 928 926 1280 763 1168 789 1300 882 ...
## $ Heating        : chr   "GasA" "GasA" "GasA" "GasA" ...
## $ HeatingQC      : chr   "TA" "TA" "Gd" "Ex" ...
## $ CentralAir     : chr   "Y" "Y" "Y" "Y" ...
## $ Electrical     : chr   "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
## $ X1stFlrSF      : int  896 1329 928 926 1280 763 1187 789 1341 882 ...
## $ X2ndFlrSF      : int  0 0 701 678 0 892 0 676 0 0 ...
## $ LowQualFinSF   : int  0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea      : int  896 1329 1629 1604 1280 1655 1187 1465 1341 882 ...
## $ BsmtFullBath   : int  0 0 0 0 0 0 1 0 1 1 ...
## $ BsmtHalfBath   : int  0 0 0 0 0 0 0 0 0 0 ...
## $ FullBath       : int  1 1 2 2 2 2 2 2 1 1 ...
```

```
## $ HalfBath      : int  0 1 1 1 0 1 0 1 1 0 ...
## $ BedroomAbvGr : int  2 3 3 3 2 3 3 3 2 2 ...
## $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 1 1 ...
## $ KitchenQual   : chr  "TA" "Gd" "TA" "Gd" ...
## $ TotRmsAbvGrd  : int  5 6 6 7 5 7 6 7 5 4 ...
## $ Functional    : chr  "Typ" "Typ" "Typ" "Typ" ...
## $ Fireplaces    : int  0 0 1 1 0 1 0 1 1 0 ...
## $ FireplaceQu   : chr  NA NA "TA" "Gd" ...
## $ GarageType     : chr  "Attchd" "Attchd" "Attchd" "Attchd" ...
## $ GarageYrBlt   : int  1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...
## $ GarageFinish  : chr  "Unf" "Unf" "Fin" "Fin" ...
## $ GarageCars    : int  1 1 2 2 2 2 2 2 2 2 ...
## $ GarageArea    : int  730 312 482 470 506 440 420 393 506 525 ...
## $ GarageQual    : chr  "TA" "TA" "TA" "TA" ...
## $ GarageCond    : chr  "TA" "TA" "TA" "TA" ...
## $ PavedDrive    : chr  "Y" "Y" "Y" "Y" ...
## $ WoodDeckSF    : int  140 393 212 360 0 157 483 0 192 240 ...
## $ OpenPorchSF   : int  0 36 34 36 82 84 21 75 0 0 ...
## $ EnclosedPorch : int  0 0 0 0 0 0 0 0 0 0 ...
## $ X3SsnPorch    : int  0 0 0 0 0 0 0 0 0 0 ...
## $ ScreenPorch   : int  120 0 0 0 144 0 0 0 0 0 ...
## $ PoolArea      : int  0 0 0 0 0 0 0 0 0 0 ...
## $ PoolQC        : chr  NA NA NA NA ...
## $ Fence         : chr  "MnPrv" NA "MnPrv" NA ...
## $ MiscFeature   : chr  NA "Gar2" NA NA ...
## $ MiscVal       : int  0 12500 0 0 0 0 500 0 0 0 ...
## $ MoSold        : int  6 6 3 6 1 4 3 5 2 4 ...
## $ YrSold        : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
## $ SaleType      : chr  "WD" "WD" "WD" "WD" ...
## $ SaleCondition : chr  "Normal" "Normal" "Normal" "Normal" ...
```

```
#We will now go down the list
```

```
#MSZoning
prop.table(table(df_test$MSZoning, useNA = "ifany"))
```

```
##
##      C (all)      FV      RH      RL      RM      <NA>
## 0.010281014 0.050719671 0.006854010 0.763536669 0.165867032 0.002741604
```

```
df_test$MSZoning[is.na(df_test$MSZoning)] <- "None"
prop.table(table(df_test$MSZoning, useNA = "ifany"))
```

```
##
##      C (all)      FV      None      RH      RL      RM
## 0.010281014 0.050719671 0.002741604 0.006854010 0.763536669 0.165867032
```

```
#LotFrontage
prop.table(table(df_test$LotFrontage, useNA = "ifany"))
```

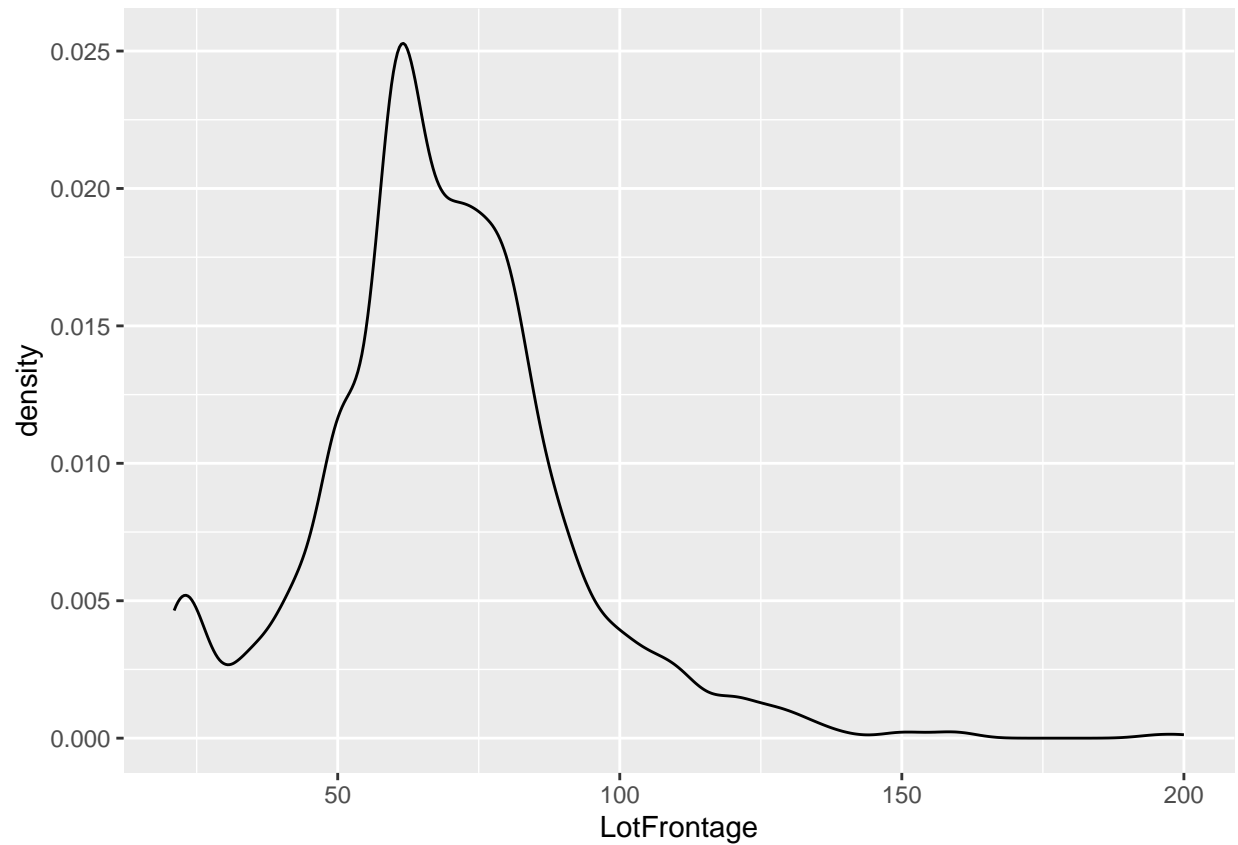
```
##
##      21      22      24      25      26      28
```



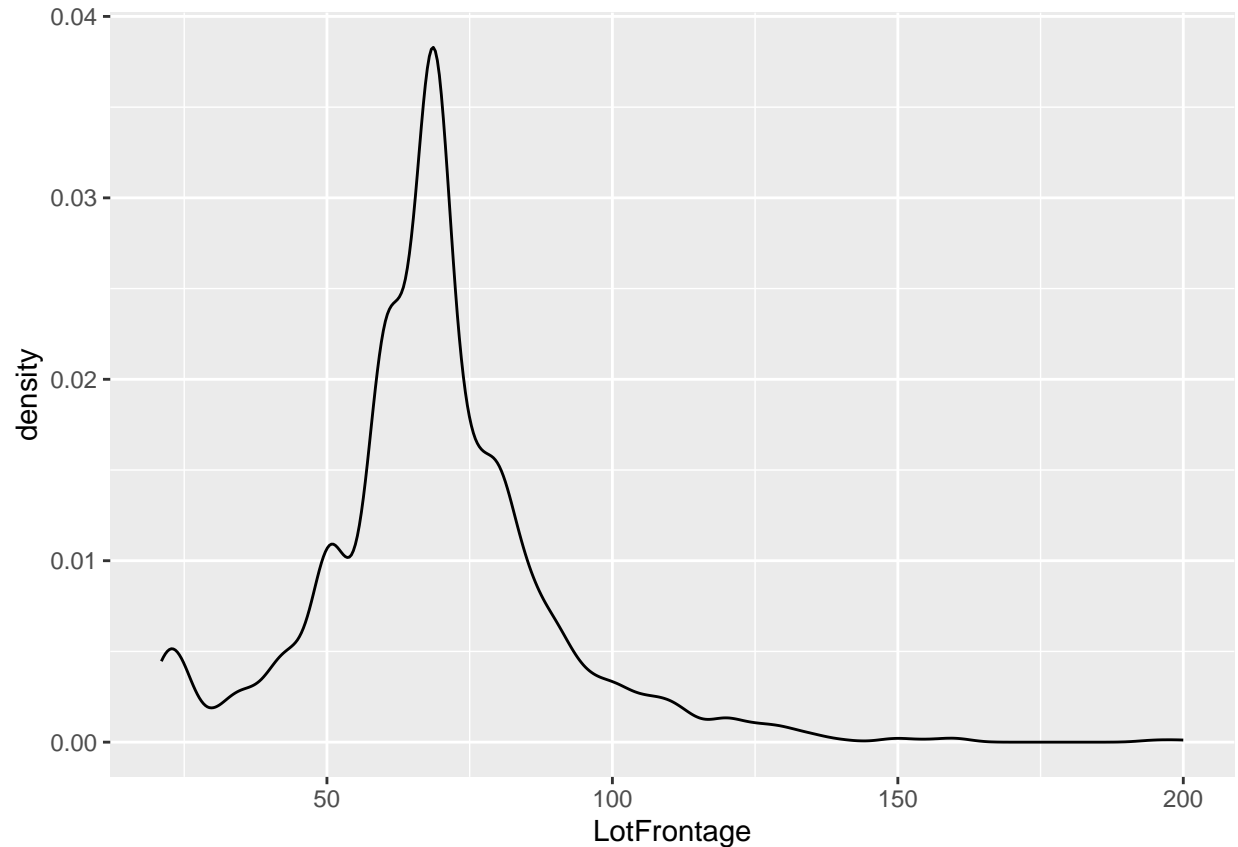
```
## 0.018505826 0.000685401 0.020562029 0.000685401 0.002056203 0.000685401
##          30          31          32          33          34          35
## 0.003427005 0.000685401 0.002056203 0.001370802 0.004112406 0.006854010
##          36          37          38          39          40          41
## 0.001370802 0.001370802 0.002056203 0.003427005 0.002741604 0.005483208
##          42          43          44          45          46          47
## 0.005483208 0.007539411 0.006854010 0.004797807 0.001370802 0.002056203
##          48          49          50          51          52          53
## 0.006854010 0.000685401 0.041124058 0.008224812 0.009595613 0.009595613
##          54          55          56          57          58          59
## 0.002741604 0.004797807 0.009595613 0.009595613 0.006854010 0.009595613
##          60          61          62          63          64          65
## 0.091158328 0.006168609 0.015078821 0.020562029 0.016449623 0.033584647
##          66          67          68          69          70          71
## 0.007539411 0.006854010 0.017135024 0.006168609 0.043180260 0.004797807
##          72          73          74          75          76          77
## 0.015078821 0.010281014 0.016449623 0.035640850 0.009595613 0.004112406
##          78          79          80          81          82          83
## 0.014393420 0.007539411 0.046607265 0.008224812 0.010966415 0.005483208
##          84          85          86          87          88          89
## 0.006168609 0.024674435 0.003427005 0.004112406 0.008224812 0.002741604
##          90          91          92          93          94          95
## 0.015764222 0.005483208 0.003427005 0.003427005 0.004112406 0.005483208
##          96          97          98          99         100         101
## 0.004112406 0.002056203 0.002741604 0.002056203 0.008224812 0.000685401
##         102         103         104         105         106         107
## 0.004112406 0.002056203 0.001370802 0.004112406 0.002056203 0.002741604
##         108         109         110         112         113         114
## 0.002056203 0.001370802 0.006168609 0.002056203 0.002056203 0.002056203
##         115         117         118         119         120         121
## 0.000685401 0.000685401 0.001370802 0.000685401 0.004797807 0.000685401
##         123         124         125         126         128         129
## 0.000685401 0.001370802 0.002056203 0.000685401 0.001370802 0.001370802
##         130         131         133         134         135         136
## 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##         140         149         150         155         160         195
## 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##         200         <NA>
## 0.000685401 0.155586018
```

```
ggplot(df_test, aes(x = LotFrontage)) + geom_density()
```

```
## Warning: Removed 227 rows containing non-finite values ('stat_density()').
```



```
#LotFrontage is heavily right skewed, therefore its NAs are imputed with its median  
df_test$LotFrontage[is.na(df_test$LotFrontage)] <- mean(df_test$LotFrontage, na.rm=TRUE)  
ggplot(df_test, aes(x = LotFrontage)) + geom_density()
```



```
#Alley
prop.table(table(df_test$Alley, useNA = "ifany"))
```

```
##
##      Grv1      Pave      <NA>
## 0.04797807 0.02535984 0.92666210
```

```
df_test$Alley[is.na(df_test$Alley)] <- "None"
```

```
#Utilities
prop.table(table(df_test$Utilities, useNA = "ifany"))
```

```
##
##      AllPub      <NA>
## 0.998629198 0.001370802
```

```
df_test$Utilities[is.na(df_test$Utilities)] <- "None"
```

```
#Exterior1st
prop.table(table(df_test$Exterior1st, useNA = "ifany"))
```

```
##
##      AsbShng      AsphShn      BrkComm      BrkFace      CBlock      CemntBd
## 0.016449623 0.000685401 0.002741604 0.025359836 0.000685401 0.044551062
```

```
##      HdBoard      MetalSd      Plywood      Stucco      VinylSd      Wd Sdng
## 0.150788211 0.157642221 0.077450308 0.012337217 0.349554489 0.140507197
##      WdShng      <NA>
## 0.020562029 0.000685401
```

```
df_test$Exterior1st[is.na(df_test$Exterior1st)] <- "None"
```

```
#Exterior2nd
```

```
prop.table(table(df_test$Exterior2nd, useNA = "ifany"))
```

```
##
##      AsbShng      AsphShn      Brk Cmn      BrkFace      CBlock      CmentBd
## 0.012337217 0.000685401 0.010281014 0.015078821 0.001370802 0.045236463
##      HdBoard      ImStucc      MetalSd      Plywood      Stone      Stucco
## 0.136394791 0.003427005 0.159698424 0.087731323 0.000685401 0.014393420
##      VinylSd      Wd Sdng      Wd Shng      <NA>
## 0.349554489 0.132967786 0.029472241 0.000685401
```

```
df_test$Exterior2nd[is.na(df_test$Exterior2nd)] <- "None"
```

```
#MasVnrArea
```

```
prop.table(table(df_test$MasVnrArea, useNA = "ifany"))
```

```
##
##      0      1      3      14      16      18
## 0.601096642 0.000685401 0.000685401 0.002056203 0.002741604 0.000685401
##      20      22      23      24      28      30
## 0.002741604 0.000685401 0.002741604 0.000685401 0.000685401 0.001370802
##      32      36      38      39      40      41
## 0.002056203 0.000685401 0.000685401 0.000685401 0.002741604 0.001370802
##      44      45      47      50      51      52
## 0.002741604 0.000685401 0.000685401 0.002741604 0.001370802 0.002056203
##      53      54      56      58      60      62
## 0.000685401 0.001370802 0.000685401 0.001370802 0.003427005 0.000685401
##      65      67      68      69      70      72
## 0.000685401 0.000685401 0.002056203 0.000685401 0.000685401 0.002056203
##      74      76      80      82      84      85
## 0.000685401 0.002056203 0.002056203 0.001370802 0.001370802 0.000685401
##      86      87      88      89      90      91
## 0.001370802 0.000685401 0.002741604 0.000685401 0.002056203 0.000685401
##      94      95      96      98      99      100
## 0.001370802 0.000685401 0.002056203 0.002056203 0.000685401 0.000685401
##      101      102      104      106      108      112
## 0.000685401 0.000685401 0.000685401 0.000685401 0.002056203 0.002741604
##      113      114      115      118      119      120
## 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401 0.005483208
##      121      122      123      124      125      126
## 0.000685401 0.001370802 0.001370802 0.000685401 0.000685401 0.002056203
##      128      130      132      134      135      136
## 0.004112406 0.002056203 0.002056203 0.001370802 0.000685401 0.001370802
##      138      140      141      142      143      144
## 0.000685401 0.003427005 0.000685401 0.000685401 0.002741604 0.006168609
##      145      146      148      149      150      153
```

##	0.002056203	0.000685401	0.001370802	0.001370802	0.002741604	0.000685401
##	156	157	158	160	161	162
##	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802	0.002056203
##	163	164	165	166	168	170
##	0.000685401	0.003427005	0.001370802	0.000685401	0.000685401	0.002056203
##	172	174	176	177	178	179
##	0.002056203	0.002741604	0.000685401	0.000685401	0.002741604	0.000685401
##	180	182	184	186	187	188
##	0.002741604	0.002056203	0.000685401	0.002741604	0.000685401	0.001370802
##	189	190	192	194	196	197
##	0.000685401	0.002056203	0.001370802	0.002741604	0.003427005	0.000685401
##	198	199	200	202	203	204
##	0.004112406	0.000685401	0.004797807	0.000685401	0.003427005	0.000685401
##	205	206	209	210	212	214
##	0.000685401	0.002056203	0.000685401	0.003427005	0.001370802	0.000685401
##	215	216	217	218	221	222
##	0.000685401	0.005483208	0.000685401	0.001370802	0.000685401	0.000685401
##	226	227	228	229	230	232
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.002741604
##	234	235	236	238	240	242
##	0.000685401	0.000685401	0.000685401	0.001370802	0.002741604	0.002741604
##	244	246	248	250	251	252
##	0.000685401	0.001370802	0.002056203	0.001370802	0.000685401	0.002056203
##	253	254	256	257	258	259
##	0.000685401	0.000685401	0.003427005	0.000685401	0.000685401	0.000685401
##	260	261	264	265	268	270
##	0.003427005	0.000685401	0.002056203	0.001370802	0.000685401	0.002056203
##	272	275	276	278	279	280
##	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401	0.002056203
##	283	284	285	286	288	289
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	290	291	292	294	295	296
##	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	298	300	302	304	305	306
##	0.000685401	0.002056203	0.004112406	0.000685401	0.000685401	0.002741604
##	308	309	310	320	322	323
##	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401	0.000685401
##	327	332	340	342	352	353
##	0.000685401	0.000685401	0.002741604	0.000685401	0.001370802	0.000685401
##	355	356	359	360	364	365
##	0.000685401	0.001370802	0.000685401	0.001370802	0.001370802	0.000685401
##	366	368	371	372	378	379
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	380	382	383	385	394	397
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	400	402	405	406	410	418
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	420	422	423	425	430	432
##	0.002741604	0.001370802	0.000685401	0.000685401	0.001370802	0.000685401
##	434	440	442	444	450	456
##	0.000685401	0.000685401	0.000685401	0.000685401	0.002056203	0.002056203
##	466	468	470	472	473	480
##	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802	0.001370802
##	492	495	500	501	502	504

```
## 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401 0.004112406
##      506      509      510      513      514      515
## 0.000685401 0.000685401 0.000685401 0.002056203 0.000685401 0.000685401
##      518      519      522      525      526      532
## 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401
##      549      550      554      567      568      572
## 0.000685401 0.000685401 0.001370802 0.000685401 0.001370802 0.000685401
##      600      615      621      632      634      647
## 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      652      657      662      668      674      680
## 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##      692      710      714      724      726      730
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      734      738      754      771      877      886
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      902      945      970      1050      1095      1110
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      1159      1224      1290      <NA>
## 0.000685401 0.001370802 0.000685401 0.010281014
```

```
#MasVnrType
```

```
prop.table(table(df_test$MasVnrType, useNA = "ifany"))
```

```
##
##      BrkCmn      BrkFace      None      Stone      <NA>
## 0.00685401 0.29746402 0.60178204 0.08293352 0.01096642
```

```
#An NA MasVnrArea seems to tie with an NA MasVnrType, therefore its NAs are set to 0
```

```
df_test$MasVnrType[is.na(df_test$MasVnrType)] <- "None"
```

```
df_test$MasVnrArea[is.na(df_test$MasVnrArea)] <- 0
```

```
#####
```

```
table(df_test$BsmtQual, useNA = "ifany")
```

```
##
##      Ex      Fa      Gd      TA <NA>
## 137      53      591      634      44
```

```
table(df_test$BsmtCond, useNA = "ifany")
```

```
##
##      Fa      Gd      Po      TA <NA>
## 59      57      3 1295      45
```

```
#BsmtQual
```

```
prop.table(table(df_test$BsmtQual, useNA = "ifany"))
```

```
##
##      Ex      Fa      Gd      TA      <NA>
## 0.09389993 0.03632625 0.40507197 0.43454421 0.03015764
```

```
df_test$BsmtQual[is.na(df_test$BsmtQual)] <- "None"
```

```
#BsmtCond
```

```
prop.table(table(df_test$BsmtCond, useNA = "ifany"))
```

```
##
##          Fa          Gd          Po          TA          <NA>
## 0.040438657 0.039067855 0.002056203 0.887594243 0.030843043
```

```
df_test$BsmtCond[is.na(df_test$BsmtCond)] <- "None"
```

```
#####
```

```
#BsmtExposure
```

```
prop.table(table(df_test$BsmtExposure, useNA = "ifany"))
```

```
##
##          Av          Gd          Mn          No          <NA>
## 0.13502399 0.09732694 0.08567512 0.65181631 0.03015764
```

```
df_test$BsmtExposure[is.na(df_test$BsmtExposure)] <- "None"
```

```
#BsmtFinType1
```

```
prop.table(table(df_test$BsmtFinType1, useNA = "ifany"))
```

```
##
##          ALQ          BLQ          GLQ          LwQ          Rec          Unf          <NA>
## 0.14324880 0.08293352 0.29540781 0.05483208 0.10623715 0.28855380 0.02878684
```

```
df_test$BsmtFinType1[is.na(df_test$BsmtFinType1)] <- "None"
```

```
#BsmtFinSF1
```

```
prop.table(table(df_test$BsmtFinSF1, useNA = "ifany"))
```

```
##
##          0          16          20          24          28          32
## 0.316655243 0.003427005 0.002056203 0.010281014 0.001370802 0.000685401
##          36          40          42          48          51          52
## 0.002056203 0.001370802 0.000685401 0.001370802 0.000685401 0.001370802
##          53          54          55          56          60          65
## 0.000685401 0.001370802 0.000685401 0.001370802 0.002741604 0.000685401
##          68          70          73          76          77          78
## 0.002056203 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401
##          80          81          85          88          96          100
## 0.002056203 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          104         110         113         114         116         119
## 0.002056203 0.002056203 0.000685401 0.002056203 0.000685401 0.002056203
##          120         121         122         126         128         130
## 0.002056203 0.001370802 0.000685401 0.000685401 0.000685401 0.002056203
##          132         133         134         138         140         143
## 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401
##          144         148         149         150         154         155
```

##	0.002056203	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	156	162	168	169	172	173
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	175	176	181	186	188	189
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802
##	190	192	194	196	198	200
##	0.001370802	0.002056203	0.000685401	0.001370802	0.001370802	0.000685401
##	201	203	205	207	208	210
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	215	216	220	221	224	225
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	228	230	234	236	238	240
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	241	242	244	246	247	248
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	249	250	252	254	256	257
##	0.000685401	0.001370802	0.002741604	0.000685401	0.000685401	0.001370802
##	258	259	260	261	262	263
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	264	267	271	273	274	276
##	0.001370802	0.001370802	0.000685401	0.000685401	0.001370802	0.004112406
##	278	279	280	281	282	283
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	284	286	288	292	294	299
##	0.001370802	0.000685401	0.003427005	0.001370802	0.001370802	0.002056203
##	300	301	305	306	308	310
##	0.003427005	0.000685401	0.001370802	0.001370802	0.001370802	0.001370802
##	311	312	314	315	316	317
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	318	319	320	321	324	326
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	329	330	331	332	335	336
##	0.001370802	0.002741604	0.000685401	0.000685401	0.000685401	0.001370802
##	337	338	339	341	342	343
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.001370802
##	346	347	350	351	353	354
##	0.000685401	0.000685401	0.001370802	0.000685401	0.002056203	0.000685401
##	355	358	360	361	362	363
##	0.000685401	0.000685401	0.002056203	0.000685401	0.000685401	0.000685401
##	366	368	370	371	372	373
##	0.000685401	0.002741604	0.000685401	0.000685401	0.001370802	0.000685401
##	374	375	376	378	379	380
##	0.000685401	0.002741604	0.000685401	0.000685401	0.001370802	0.000685401
##	381	382	384	385	386	388
##	0.000685401	0.000685401	0.002741604	0.000685401	0.000685401	0.000685401
##	389	390	393	397	399	400
##	0.000685401	0.001370802	0.000685401	0.001370802	0.000685401	0.000685401
##	402	403	406	408	410	414
##	0.000685401	0.000685401	0.001370802	0.002056203	0.001370802	0.001370802
##	415	416	417	420	421	424
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	425	426	432	433	434	435
##	0.000685401	0.000685401	0.002056203	0.000685401	0.000685401	0.002056203
##	437	438	441	445	448	450



##	0.001370802	0.000685401	0.001370802	0.001370802	0.000685401	0.001370802
##	452	453	454	455	456	457
##	0.000685401	0.001370802	0.000685401	0.000685401	0.002741604	0.000685401
##	458	460	462	466	467	468
##	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401	0.002741604
##	469	471	472	474	475	476
##	0.000685401	0.000685401	0.001370802	0.002056203	0.000685401	0.002741604
##	480	481	483	484	485	486
##	0.002056203	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	488	489	490	491	492	494
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.002056203
##	496	497	500	501	502	503
##	0.000685401	0.000685401	0.002741604	0.000685401	0.000685401	0.000685401
##	504	505	506	507	509	510
##	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	512	513	514	516	520	521
##	0.000685401	0.000685401	0.000685401	0.001370802	0.002056203	0.000685401
##	522	526	527	528	531	532
##	0.001370802	0.000685401	0.002056203	0.002056203	0.001370802	0.001370802
##	533	534	535	536	537	539
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	540	544	546	547	548	549
##	0.000685401	0.002741604	0.001370802	0.001370802	0.001370802	0.002056203
##	550	552	553	554	557	560
##	0.002056203	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	564	565	566	568	574	575
##	0.000685401	0.000685401	0.000685401	0.000685401	0.002056203	0.000685401
##	576	577	578	579	580	583
##	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	585	587	588	590	594	595
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	596	599	600	601	602	603
##	0.000685401	0.000685401	0.002741604	0.000685401	0.004112406	0.000685401
##	606	608	609	611	612	615
##	0.001370802	0.000685401	0.000685401	0.000685401	0.002056203	0.000685401
##	617	621	622	623	624	625
##	0.000685401	0.001370802	0.000685401	0.000685401	0.002741604	0.002056203
##	626	630	632	633	634	636
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	637	638	639	644	646	647
##	0.002056203	0.001370802	0.000685401	0.001370802	0.000685401	0.000685401
##	648	651	652	654	656	658
##	0.001370802	0.000685401	0.000685401	0.001370802	0.000685401	0.001370802
##	659	660	662	663	666	668
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	669	670	671	672	673	678
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	679	681	682	683	685	687
##	0.000685401	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401
##	688	691	697	698	699	700
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.002741604
##	701	704	705	706	708	709
##	0.000685401	0.002056203	0.000685401	0.000685401	0.000685401	0.000685401
##	710	712	717	718	720	722

##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	725	726	727	728	729	732
##	0.000685401	0.000685401	0.001370802	0.000685401	0.001370802	0.001370802
##	734	736	737	739	741	744
##	0.002056203	0.001370802	0.000685401	0.000685401	0.001370802	0.001370802
##	745	747	748	749	755	758
##	0.001370802	0.001370802	0.000685401	0.000685401	0.002056203	0.003427005
##	760	762	764	766	767	769
##	0.001370802	0.001370802	0.000685401	0.001370802	0.001370802	0.001370802
##	770	771	773	774	775	776
##	0.001370802	0.000685401	0.001370802	0.000685401	0.001370802	0.001370802
##	778	779	780	781	782	784
##	0.001370802	0.002056203	0.002056203	0.002056203	0.000685401	0.002056203
##	785	787	788	791	792	793
##	0.001370802	0.000685401	0.001370802	0.001370802	0.000685401	0.001370802
##	794	796	797	799	800	803
##	0.000685401	0.001370802	0.000685401	0.001370802	0.002056203	0.000685401
##	804	809	810	811	812	813
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	816	819	820	824	826	828
##	0.000685401	0.000685401	0.002056203	0.000685401	0.000685401	0.000685401
##	832	833	836	838	841	844
##	0.001370802	0.000685401	0.001370802	0.000685401	0.001370802	0.000685401
##	846	847	848	850	851	853
##	0.001370802	0.001370802	0.000685401	0.001370802	0.002056203	0.000685401
##	856	860	864	865	866	870
##	0.002056203	0.000685401	0.002741604	0.000685401	0.000685401	0.002056203
##	871	872	873	876	880	881
##	0.000685401	0.002056203	0.000685401	0.001370802	0.000685401	0.000685401
##	888	890	893	894	901	902
##	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802	0.001370802
##	903	904	908	909	910	913
##	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802	0.001370802
##	914	915	918	920	923	924
##	0.000685401	0.002741604	0.000685401	0.001370802	0.001370802	0.000685401
##	925	926	930	931	935	936
##	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401	0.001370802
##	937	938	939	941	944	949
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	950	951	952	953	954	955
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	958	960	962	964	967	968
##	0.000685401	0.000685401	0.002056203	0.002056203	0.000685401	0.000685401
##	973	976	983	986	988	994
##	0.000685401	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401
##	996	998	1000	1001	1004	1005
##	0.001370802	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	1010	1011	1012	1015	1018	1021
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1023	1026	1027	1029	1030	1032
##	0.002056203	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	1033	1034	1035	1036	1037	1038
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1044	1047	1048	1051	1053	1059

```
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802
##          1070          1073          1075          1078          1080          1082
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1084          1087          1090          1092          1094          1096
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1098          1101          1104          1110          1111          1112
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1115          1116          1118          1122          1127          1129
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1136          1137          1141          1142          1148          1149
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1151          1152          1153          1173          1178          1181
## 0.000685401 0.001370802 0.000685401 0.001370802 0.000685401 0.000685401
##          1182          1188          1191          1194          1196          1198
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1200          1204          1206          1218          1220          1223
## 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401 0.000685401
##          1224          1225          1230          1232          1234          1239
## 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401
##          1243          1246          1249          1252          1259          1262
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1271          1290          1294          1298          1300          1304
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1308          1309          1312          1319          1328          1329
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1333          1337          1341          1346          1350          1359
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1360          1369          1373          1375          1383          1386
## 0.001370802 0.000685401 0.001370802 0.000685401 0.001370802 0.000685401
##          1387          1392          1412          1414          1416          1420
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1430          1433          1455          1474          1476          1478
## 0.001370802 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##          1500          1505          1518          1531          1538          1557
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1558          1562          1564          1571          1572          1573
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802
##          1576          1593          1632          1640          1660          1682
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1684          1728          1732          1733          1758          1812
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1836          1965          1972          2085          2146          2158
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          2257          2288          4010          <NA>
## 0.000685401 0.000685401 0.000685401 0.000685401
```

```
df_test$BsmtFinSF1[is.na(df_test$BsmtFinSF1)] <- 0
```

```
#BsmtFinType2
```

```
prop.table(table(df_test$BsmtFinType2, useNA = "ifany"))
```

```
##
##          ALQ          BLQ          GLQ          LwQ          Rec          Unf          <NA>
## 0.02261823 0.02398903 0.01370802 0.02810144 0.03495545 0.84784099 0.02878684
```

```
df_test$BsmtFinType2[is.na(df_test$BsmtFinType2)] <- "None"
```

```
#BsmtFinSF2
```

```
prop.table(table(df_test$BsmtFinSF2, useNA = "ifany"))
```

```
##
##      0      6      12      38      40      42
## 0.875942426 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802
##      46      48      52      60      63      66
## 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401
##      68      72      76      78      80      92
## 0.000685401 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401
##      95     102     105     108     110     113
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     116     120     121     127     128     136
## 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     138     144     147     153     154     156
## 0.000685401 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401
##     159     162     167     168     174     186
## 0.001370802 0.002056203 0.000685401 0.001370802 0.000685401 0.000685401
##     196     201     202     206     210     216
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     227     240     243     247     250     252
## 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401 0.001370802
##     259     262     263     264     270     273
## 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##     276     278     281     284     286     288
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802
##     294     297     308     319     321     337
## 0.002056203 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     344     350     351     354     360     364
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     373     382     387     393     398     400
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     402     404     417     419     432     435
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802
##     442     448     449     450     453     456
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     465     474     483     488     492     495
## 0.000685401 0.000685401 0.002056203 0.000685401 0.000685401 0.000685401
##     497     507     512     522     529     530
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     539     543     555     590     596     604
## 0.000685401 0.000685401 0.000685401 0.001370802 0.001370802 0.000685401
##     613     619     620     622     624     670
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     679     684     688     691     694     722
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     723     748     750     755     761     774
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     799     811     826     829     831     841
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     842     850     852     859     873     875
```

```
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      884      891      904      912      915      955
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      956      981      982     1020     1037     1039
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     1073     1083     1164     1393     1526     <NA>
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
```

```
df_test$BsmtFinSF2[is.na(df_test$BsmtFinSF2)] <- 0
```

```
#BsmtUnfSF
```

```
prop.table(table(df_test$BsmtUnfSF, useNA = "ifany"))
```

```
##
##      0      17      20      22      25      27
## 0.084304318 0.000685401 0.000685401 0.000685401 0.002056203 0.000685401
##      28      30      33      34      35      36
## 0.000685401 0.003427005 0.000685401 0.000685401 0.000685401 0.000685401
##      42      45      46      48      52      53
## 0.000685401 0.000685401 0.002056203 0.001370802 0.001370802 0.000685401
##      54      55      56      57      58      60
## 0.001370802 0.000685401 0.001370802 0.000685401 0.000685401 0.000685401
##      63      72      75      76      78      79
## 0.000685401 0.002056203 0.000685401 0.001370802 0.000685401 0.000685401
##      80      83      86      88      89      90
## 0.001370802 0.001370802 0.000685401 0.001370802 0.001370802 0.001370802
##      91      93      94      95      96      98
## 0.000685401 0.001370802 0.001370802 0.000685401 0.002056203 0.002056203
##      99     100     102     104     106     107
## 0.000685401 0.004797807 0.000685401 0.000685401 0.000685401 0.000685401
##     108     109     110     111     112     113
## 0.002056203 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##     115     117     118     120     122     123
## 0.000685401 0.000685401 0.000685401 0.004112406 0.000685401 0.000685401
##     125     126     127     128     129     130
## 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802
##     131     132     135     136     137     138
## 0.000685401 0.002056203 0.000685401 0.000685401 0.001370802 0.002056203
##     140     141     142     143     144     147
## 0.001370802 0.000685401 0.001370802 0.002056203 0.002056203 0.000685401
##     150     153     156     158     160     161
## 0.003427005 0.000685401 0.000685401 0.001370802 0.002741604 0.000685401
##     162     163     164     165     166     167
## 0.002056203 0.001370802 0.000685401 0.000685401 0.002741604 0.000685401
##     168     170     171     172     173     174
## 0.001370802 0.000685401 0.001370802 0.002056203 0.000685401 0.000685401
##     175     176     177     179     180     181
## 0.000685401 0.001370802 0.000685401 0.000685401 0.001370802 0.000685401
##     182     183     185     186     187     188
## 0.001370802 0.000685401 0.000685401 0.003427005 0.000685401 0.001370802
##     189     190     191     192     193     194
## 0.000685401 0.002741604 0.001370802 0.000685401 0.000685401 0.000685401
##     195     196     197     198     200     201
## 0.002741604 0.000685401 0.001370802 0.000685401 0.002741604 0.002056203
```

##	204	206	207	208	210	212
##	0.002056203	0.002056203	0.001370802	0.000685401	0.002056203	0.002056203
##	213	214	215	216	217	218
##	0.002056203	0.000685401	0.001370802	0.004112406	0.000685401	0.001370802
##	219	220	221	222	223	224
##	0.001370802	0.002056203	0.000685401	0.002056203	0.000685401	0.002741604
##	225	226	228	229	231	232
##	0.000685401	0.003427005	0.003427005	0.001370802	0.000685401	0.002741604
##	233	234	235	236	237	238
##	0.000685401	0.000685401	0.002056203	0.000685401	0.000685401	0.001370802
##	240	242	243	245	247	248
##	0.003427005	0.001370802	0.000685401	0.001370802	0.000685401	0.001370802
##	249	250	251	252	253	254
##	0.000685401	0.002741604	0.000685401	0.002741604	0.001370802	0.001370802
##	255	258	260	261	262	263
##	0.000685401	0.000685401	0.000685401	0.002741604	0.000685401	0.001370802
##	264	265	268	269	270	271
##	0.002741604	0.002056203	0.001370802	0.000685401	0.002741604	0.000685401
##	272	273	275	276	277	278
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.002056203
##	279	280	282	284	285	286
##	0.000685401	0.002056203	0.002741604	0.000685401	0.001370802	0.001370802
##	288	290	292	293	294	295
##	0.000685401	0.000685401	0.002056203	0.000685401	0.004112406	0.001370802
##	296	297	298	299	300	301
##	0.000685401	0.002056203	0.000685401	0.000685401	0.002056203	0.001370802
##	304	306	307	308	311	312
##	0.000685401	0.003427005	0.000685401	0.002056203	0.000685401	0.001370802
##	313	316	318	319	320	321
##	0.000685401	0.002056203	0.000685401	0.001370802	0.000685401	0.000685401
##	322	323	324	325	326	327
##	0.003427005	0.000685401	0.002056203	0.002056203	0.000685401	0.000685401
##	328	329	330	331	332	333
##	0.001370802	0.002056203	0.001370802	0.000685401	0.001370802	0.000685401
##	335	336	338	339	340	341
##	0.000685401	0.002056203	0.000685401	0.001370802	0.000685401	0.001370802
##	342	345	346	348	350	352
##	0.002056203	0.001370802	0.002741604	0.004797807	0.000685401	0.000685401
##	354	355	356	357	358	359
##	0.000685401	0.000685401	0.002741604	0.000685401	0.000685401	0.000685401
##	360	362	363	365	366	367
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	369	370	371	372	375	378
##	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	379	380	381	382	383	384
##	0.001370802	0.000685401	0.001370802	0.000685401	0.000685401	0.007539411
##	385	386	388	390	392	393
##	0.000685401	0.002056203	0.000685401	0.000685401	0.002056203	0.000685401
##	394	395	396	397	398	399
##	0.000685401	0.001370802	0.003427005	0.001370802	0.001370802	0.001370802
##	400	402	403	404	405	406
##	0.001370802	0.000685401	0.001370802	0.002056203	0.003427005	0.000685401
##	407	408	410	411	412	415
##	0.000685401	0.002056203	0.000685401	0.000685401	0.000685401	0.001370802

##	416	420	422	423	424	425
##	0.002741604	0.000685401	0.002056203	0.001370802	0.000685401	0.000685401
##	426	427	430	431	432	433
##	0.001370802	0.001370802	0.001370802	0.000685401	0.002741604	0.000685401
##	434	436	437	438	439	440
##	0.001370802	0.001370802	0.001370802	0.000685401	0.000685401	0.001370802
##	441	443	444	448	449	450
##	0.000685401	0.001370802	0.000685401	0.002741604	0.000685401	0.002056203
##	451	453	454	455	456	457
##	0.000685401	0.000685401	0.000685401	0.002741604	0.003427005	0.000685401
##	459	460	461	462	466	467
##	0.000685401	0.002056203	0.000685401	0.002056203	0.001370802	0.001370802
##	469	470	471	472	474	475
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.001370802
##	476	477	479	480	481	482
##	0.002741604	0.000685401	0.000685401	0.004797807	0.000685401	0.002056203
##	483	484	486	487	490	491
##	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401	0.002056203
##	494	496	497	498	500	501
##	0.001370802	0.002056203	0.001370802	0.002056203	0.000685401	0.000685401
##	502	503	504	505	507	508
##	0.000685401	0.002056203	0.001370802	0.000685401	0.001370802	0.000685401
##	510	513	516	517	520	521
##	0.000685401	0.002056203	0.001370802	0.000685401	0.001370802	0.000685401
##	522	523	525	526	527	529
##	0.000685401	0.000685401	0.003427005	0.000685401	0.000685401	0.000685401
##	530	532	533	534	535	536
##	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401	0.001370802
##	538	540	541	543	545	546
##	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802	0.003427005
##	547	548	549	550	552	554
##	0.000685401	0.001370802	0.000685401	0.001370802	0.002056203	0.000685401
##	555	556	559	560	561	564
##	0.000685401	0.000685401	0.000685401	0.002056203	0.000685401	0.002056203
##	565	569	570	572	575	576
##	0.000685401	0.002056203	0.002056203	0.002741604	0.001370802	0.002741604
##	577	579	581	583	584	585
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.003427005
##	587	588	590	591	592	593
##	0.000685401	0.000685401	0.002741604	0.000685401	0.001370802	0.000685401
##	594	595	596	598	599	600
##	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401	0.003427005
##	604	605	608	610	613	615
##	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401	0.000685401
##	616	618	621	624	628	630
##	0.002056203	0.001370802	0.000685401	0.005483208	0.000685401	0.002741604
##	631	632	634	635	636	637
##	0.000685401	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401
##	638	639	640	641	643	644
##	0.000685401	0.001370802	0.000685401	0.002056203	0.000685401	0.001370802
##	646	647	649	655	656	657
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	659	660	661	662	663	666
##	0.000685401	0.001370802	0.000685401	0.001370802	0.001370802	0.001370802

##	667	671	672	674	675	676
##	0.000685401	0.000685401	0.004797807	0.000685401	0.001370802	0.002741604
##	677	678	679	680	683	684
##	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	686	689	692	697	698	704
##	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802
##	707	710	713	715	716	717
##	0.000685401	0.000685401	0.001370802	0.000685401	0.001370802	0.000685401
##	720	722	723	724	725	727
##	0.001370802	0.000685401	0.002056203	0.000685401	0.002056203	0.000685401
##	728	729	736	738	739	740
##	0.003427005	0.000685401	0.001370802	0.004797807	0.000685401	0.000685401
##	741	742	744	745	747	749
##	0.002056203	0.000685401	0.002056203	0.000685401	0.000685401	0.000685401
##	750	752	756	760	763	764
##	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802	0.000685401
##	765	767	768	769	770	771
##	0.000685401	0.000685401	0.003427005	0.000685401	0.000685401	0.000685401
##	778	780	781	782	784	785
##	0.001370802	0.002741604	0.001370802	0.001370802	0.004112406	0.002056203
##	788	789	792	794	796	797
##	0.000685401	0.001370802	0.000685401	0.000685401	0.002056203	0.000685401
##	798	801	802	808	809	810
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802
##	815	816	817	818	819	821
##	0.002056203	0.004112406	0.001370802	0.000685401	0.000685401	0.001370802
##	825	827	828	830	831	832
##	0.001370802	0.000685401	0.000685401	0.000685401	0.001370802	0.003427005
##	835	836	840	844	845	846
##	0.001370802	0.000685401	0.002741604	0.000685401	0.000685401	0.001370802
##	847	848	849	850	851	852
##	0.002056203	0.001370802	0.001370802	0.002056203	0.002056203	0.000685401
##	856	858	860	861	863	864
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	868	869	872	873	879	880
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	884	888	889	891	892	894
##	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	896	897	898	901	903	908
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	910	912	916	917	918	925
##	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802	0.002056203
##	928	929	930	931	932	935
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	936	938	939	941	944	945
##	0.002056203	0.002741604	0.000685401	0.001370802	0.000685401	0.001370802
##	949	951	952	957	960	963
##	0.000685401	0.000685401	0.001370802	0.000685401	0.002056203	0.000685401
##	966	967	969	970	971	972
##	0.000685401	0.001370802	0.000685401	0.001370802	0.000685401	0.000685401
##	974	982	992	996	998	1008
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	1012	1017	1018	1020	1022	1026
##	0.000685401	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401



##	1028	1035	1040	1041	1043	1046
##	0.000685401	0.000685401	0.002741604	0.000685401	0.000685401	0.001370802
##	1048	1052	1054	1057	1058	1064
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	1065	1066	1068	1074	1075	1077
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	1081	1082	1084	1085	1087	1088
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1090	1092	1093	1095	1100	1105
##	0.000685401	0.000685401	0.001370802	0.000685401	0.002741604	0.000685401
##	1108	1114	1115	1117	1118	1126
##	0.000685401	0.002056203	0.000685401	0.000685401	0.000685401	0.000685401
##	1128	1129	1131	1140	1143	1144
##	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802	0.000685401
##	1146	1150	1152	1153	1158	1162
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	1168	1173	1179	1180	1181	1191
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1195	1198	1204	1209	1211	1214
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	1216	1218	1226	1228	1242	1246
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	1248	1250	1251	1254	1257	1258
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1262	1266	1270	1272	1280	1298
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1300	1302	1308	1313	1314	1318
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	1323	1324	1326	1327	1328	1330
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1332	1335	1337	1339	1341	1344
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	1346	1347	1348	1369	1376	1390
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1400	1402	1404	1405	1411	1421
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1430	1436	1439	1444	1450	1451
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	1453	1461	1466	1468	1473	1474
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	1486	1488	1491	1495	1496	1503
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	1504	1508	1510	1519	1526	1527
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1528	1530	1544	1550	1556	1558
##	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	1559	1560	1568	1574	1582	1584
##	0.000685401	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401
##	1588	1590	1594	1595	1598	1602
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	1604	1614	1615	1618	1619	1625
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1629	1632	1640	1643	1660	1664
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401

```
##          1678          1680          1685          1689          1694          1696
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1704          1706          1721          1726          1728          1736
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          1765          1794          1800          1802          1824          1836
## 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##          1851          1866          1921          1958          1967          2062
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          2140          <NA>
## 0.000685401 0.000685401
```

```
df_test$BsmtUnfSF[is.na(df_test$BsmtUnfSF)] <- 0

#TotalBsmtSF
prop.table(table(df_test$TotalBsmtSF, useNA = "ifany"))
```

```
##
##          0          160          173          192          216          240
## 0.028101439 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##          245          264          279          297          301          346
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802
##          348          352          356          370          381          384
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.008224812
##          385          392          396          405          407          409
## 0.000685401 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401
##          416          423          430          432          448          450
## 0.001370802 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##          451          456          462          468          480          481
## 0.000685401 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401
##          483          484          492          494          498          502
## 0.004797807 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          504          516          520          525          526          528
## 0.000685401 0.002056203 0.001370802 0.002741604 0.000685401 0.001370802
##          530          531          533          536          546          547
## 0.002056203 0.000685401 0.000685401 0.000685401 0.008224812 0.002056203
##          550          552          554          560          561          565
## 0.000685401 0.000685401 0.000685401 0.001370802 0.001370802 0.000685401
##          569          570          572          576          583          585
## 0.000685401 0.001370802 0.003427005 0.002056203 0.000685401 0.002741604
##          592          596          600          608          610          616
## 0.001370802 0.000685401 0.006168609 0.002741604 0.000685401 0.001370802
##          621          624          628          629          630          631
## 0.000685401 0.006854010 0.000685401 0.000685401 0.002741604 0.000685401
##          635          636          637          641          644          651
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##          660          662          663          666          671          672
## 0.002056203 0.000685401 0.001370802 0.001370802 0.000685401 0.008224812
##          675          676          677          678          680          684
## 0.001370802 0.003427005 0.000685401 0.000685401 0.002741604 0.001370802
##          686          687          689          690          691          697
## 0.001370802 0.000685401 0.001370802 0.000685401 0.001370802 0.000685401
##          698          699          704          707          709          712
## 0.002056203 0.000685401 0.002056203 0.002056203 0.000685401 0.001370802
##          713          715          716          720          723          725
```

##	0.000685401	0.001370802	0.001370802	0.006168609	0.000685401	0.002741604
##	728	732	734	736	738	739
##	0.005483208	0.001370802	0.000685401	0.000685401	0.005483208	0.000685401
##	741	744	745	747	750	752
##	0.002056203	0.002741604	0.000685401	0.001370802	0.000685401	0.001370802
##	754	755	756	760	763	764
##	0.001370802	0.000685401	0.005483208	0.000685401	0.000685401	0.000685401
##	765	768	770	771	779	780
##	0.001370802	0.008224812	0.000685401	0.000685401	0.000685401	0.005483208
##	781	782	784	789	792	793
##	0.001370802	0.002741604	0.004797807	0.001370802	0.000685401	0.001370802
##	794	795	796	797	798	799
##	0.000685401	0.000685401	0.003427005	0.000685401	0.001370802	0.000685401
##	801	804	806	808	810	812
##	0.001370802	0.001370802	0.000685401	0.002056203	0.000685401	0.001370802
##	813	816	817	818	819	821
##	0.000685401	0.006854010	0.000685401	0.000685401	0.000685401	0.000685401
##	822	824	825	826	827	828
##	0.000685401	0.000685401	0.002741604	0.000685401	0.001370802	0.001370802
##	830	831	832	833	835	836
##	0.000685401	0.001370802	0.004797807	0.001370802	0.003427005	0.003427005
##	837	840	842	844	845	846
##	0.000685401	0.004112406	0.000685401	0.000685401	0.002741604	0.000685401
##	847	848	850	851	852	853
##	0.002056203	0.003427005	0.001370802	0.001370802	0.000685401	0.000685401
##	854	855	856	858	859	860
##	0.001370802	0.003427005	0.001370802	0.002056203	0.000685401	0.001370802
##	861	864	868	869	870	872
##	0.002056203	0.026730637	0.001370802	0.000685401	0.001370802	0.001370802
##	876	878	879	880	882	884
##	0.003427005	0.000685401	0.000685401	0.000685401	0.002741604	0.002741604
##	886	888	890	891	892	894
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.004112406
##	895	896	900	901	902	903
##	0.000685401	0.002056203	0.001370802	0.002056203	0.000685401	0.000685401
##	904	907	908	910	911	912
##	0.000685401	0.000685401	0.000685401	0.002056203	0.000685401	0.007539411
##	913	914	915	916	917	918
##	0.000685401	0.002741604	0.000685401	0.000685401	0.000685401	0.000685401
##	920	922	923	924	925	926
##	0.002741604	0.000685401	0.002056203	0.000685401	0.003427005	0.003427005
##	928	929	930	931	932	936
##	0.002741604	0.001370802	0.002741604	0.000685401	0.001370802	0.005483208
##	938	941	942	943	944	945
##	0.002056203	0.002056203	0.000685401	0.001370802	0.000685401	0.002741604
##	946	948	949	951	952	954
##	0.000685401	0.001370802	0.002056203	0.002056203	0.002741604	0.000685401
##	955	957	960	964	966	967
##	0.001370802	0.001370802	0.008910212	0.000685401	0.001370802	0.002741604
##	969	970	972	973	975	976
##	0.000685401	0.002056203	0.003427005	0.001370802	0.001370802	0.002741604
##	979	980	981	982	984	985
##	0.000685401	0.001370802	0.001370802	0.002741604	0.003427005	0.000685401
##	988	989	990	992	994	995

##	0.006854010	0.000685401	0.002056203	0.002056203	0.000685401	0.001370802
##	996	998	1000	1001	1004	1005
##	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401	0.002056203
##	1008	1014	1015	1020	1022	1024
##	0.008224812	0.000685401	0.001370802	0.001370802	0.002741604	0.002056203
##	1025	1026	1027	1029	1031	1032
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	1035	1036	1037	1038	1040	1041
##	0.000685401	0.000685401	0.001370802	0.000685401	0.007539411	0.000685401
##	1042	1043	1044	1045	1046	1047
##	0.000685401	0.001370802	0.001370802	0.001370802	0.000685401	0.000685401
##	1048	1049	1050	1051	1052	1053
##	0.001370802	0.001370802	0.000685401	0.001370802	0.002056203	0.000685401
##	1054	1055	1056	1057	1058	1059
##	0.003427005	0.001370802	0.001370802	0.001370802	0.002056203	0.000685401
##	1060	1061	1062	1064	1065	1066
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	1067	1068	1069	1070	1072	1073
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.003427005
##	1074	1075	1076	1077	1078	1079
##	0.000685401	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401
##	1080	1081	1082	1083	1084	1088
##	0.002056203	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1089	1090	1091	1092	1093	1094
##	0.000685401	0.002056203	0.000685401	0.001370802	0.001370802	0.002056203
##	1097	1100	1103	1104	1105	1107
##	0.000685401	0.004112406	0.000685401	0.002056203	0.002741604	0.001370802
##	1108	1112	1113	1114	1116	1117
##	0.001370802	0.000685401	0.000685401	0.002741604	0.000685401	0.001370802
##	1118	1121	1122	1124	1127	1128
##	0.000685401	0.000685401	0.001370802	0.002056203	0.000685401	0.000685401
##	1129	1130	1131	1135	1138	1140
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.001370802
##	1141	1142	1143	1144	1145	1146
##	0.002056203	0.001370802	0.002056203	0.000685401	0.001370802	0.000685401
##	1147	1148	1149	1150	1151	1152
##	0.000685401	0.001370802	0.000685401	0.002056203	0.001370802	0.002056203
##	1153	1158	1160	1162	1165	1166
##	0.001370802	0.002056203	0.002056203	0.000685401	0.000685401	0.000685401
##	1168	1169	1170	1172	1173	1174
##	0.004797807	0.001370802	0.000685401	0.000685401	0.001370802	0.000685401
##	1175	1176	1177	1178	1179	1180
##	0.001370802	0.002741604	0.000685401	0.000685401	0.001370802	0.000685401
##	1181	1182	1187	1188	1189	1190
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	1191	1195	1196	1200	1202	1204
##	0.001370802	0.000685401	0.000685401	0.002741604	0.000685401	0.001370802
##	1206	1208	1209	1211	1212	1214
##	0.000685401	0.002056203	0.001370802	0.000685401	0.001370802	0.000685401
##	1216	1217	1218	1220	1222	1223
##	0.002741604	0.000685401	0.002741604	0.002056203	0.000685401	0.000685401
##	1224	1226	1228	1230	1231	1232
##	0.001370802	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	1235	1236	1240	1242	1243	1244

##	0.000685401	0.000685401	0.000685401	0.002056203	0.000685401	0.001370802
##	1246	1248	1250	1251	1254	1256
##	0.001370802	0.002056203	0.000685401	0.000685401	0.000685401	0.001370802
##	1257	1259	1260	1265	1266	1267
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1270	1272	1273	1277	1278	1280
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.002056203
##	1282	1284	1286	1288	1290	1292
##	0.000685401	0.000685401	0.000685401	0.002056203	0.001370802	0.000685401
##	1295	1296	1298	1300	1302	1306
##	0.000685401	0.001370802	0.000685401	0.001370802	0.002741604	0.000685401
##	1308	1310	1312	1313	1317	1319
##	0.000685401	0.001370802	0.001370802	0.002741604	0.000685401	0.000685401
##	1324	1325	1326	1329	1330	1331
##	0.001370802	0.000685401	0.001370802	0.001370802	0.000685401	0.000685401
##	1332	1334	1335	1336	1337	1338
##	0.000685401	0.001370802	0.000685401	0.001370802	0.001370802	0.000685401
##	1339	1341	1342	1344	1347	1348
##	0.000685401	0.001370802	0.002741604	0.002056203	0.000685401	0.001370802
##	1350	1351	1352	1357	1358	1362
##	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1363	1365	1367	1368	1369	1370
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	1372	1376	1377	1378	1380	1381
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1382	1388	1389	1390	1393	1395
##	0.000685401	0.000685401	0.000685401	0.002056203	0.000685401	0.001370802
##	1396	1398	1400	1401	1402	1405
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.002741604
##	1406	1408	1409	1414	1415	1417
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1418	1419	1420	1422	1425	1426
##	0.001370802	0.000685401	0.002741604	0.001370802	0.000685401	0.000685401
##	1427	1430	1431	1432	1433	1434
##	0.001370802	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	1436	1438	1441	1444	1445	1449
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	1450	1451	1453	1454	1455	1460
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1461	1462	1463	1468	1479	1480
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401
##	1487	1488	1489	1491	1492	1494
##	0.000685401	0.002056203	0.000685401	0.001370802	0.001370802	0.001370802
##	1495	1498	1501	1504	1508	1509
##	0.000685401	0.000685401	0.000685401	0.001370802	0.001370802	0.000685401
##	1510	1511	1512	1518	1519	1524
##	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401	0.002056203
##	1528	1529	1530	1531	1538	1540
##	0.002056203	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401
##	1542	1544	1546	1550	1553	1554
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	1555	1556	1557	1560	1561	1562
##	0.000685401	0.001370802	0.000685401	0.002741604	0.000685401	0.000685401
##	1564	1568	1569	1570	1573	1574

```
## 0.000685401 0.001370802 0.000685401 0.000685401 0.001370802 0.000685401
##      1577      1578      1580      1582      1584      1587
## 0.000685401 0.001370802 0.000685401 0.002056203 0.000685401 0.001370802
##      1590      1592      1594      1595      1596      1598
## 0.000685401 0.000685401 0.002741604 0.001370802 0.001370802 0.000685401
##      1602      1603      1604      1606      1612      1614
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.002741604
##      1615      1617      1618      1620      1621      1625
## 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401 0.002056203
##      1629      1632      1641      1642      1643      1645
## 0.000685401 0.001370802 0.000685401 0.000685401 0.001370802 0.000685401
##      1652      1660      1664      1666      1670      1671
## 0.001370802 0.000685401 0.001370802 0.001370802 0.000685401 0.000685401
##      1673      1675      1678      1679      1680      1685
## 0.000685401 0.000685401 0.000685401 0.000685401 0.002056203 0.000685401
##      1686      1689      1694      1696      1698      1700
## 0.000685401 0.000685401 0.001370802 0.001370802 0.000685401 0.000685401
##      1702      1704      1705      1706      1710      1712
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      1714      1720      1721      1726      1728      1733
## 0.000685401 0.000685401 0.001370802 0.000685401 0.002056203 0.000685401
##      1736      1738      1739      1740      1742      1748
## 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401
##      1751      1760      1765      1774      1776      1778
## 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##      1779      1780      1782      1790      1792      1800
## 0.000685401 0.000685401 0.001370802 0.000685401 0.000685401 0.001370802
##      1802      1803      1822      1824      1829      1832
## 0.000685401 0.000685401 0.001370802 0.001370802 0.000685401 0.000685401
##      1836      1838      1840      1848      1850      1851
## 0.002056203 0.002056203 0.000685401 0.000685401 0.001370802 0.000685401
##      1858      1865      1866      1868      1884      1898
## 0.000685401 0.000685401 0.000685401 0.000685401 0.002056203 0.000685401
##      1910      1921      1922      1934      1949      1950
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      1958      1964      1966      1967      1978      1982
## 0.000685401 0.000685401 0.000685401 0.000685401 0.001370802 0.000685401
##      1994      2002      2014      2020      2024      2036
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      2048      2062      2108      2140      2171      2190
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      2200      2208      2220      2271      2320      2418
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      2452      2458      2461      2492      2535      2552
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      2630      2660      2846      5095      <NA>
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
```

```
df_test$TotalBsmtSF[is.na(df_test$TotalBsmtSF)] <- 0
```

```
#BsmtFullBath
```

```
prop.table(table(df_test$BsmtFullBath, useNA = "ifany"))
```

```
##
```

```
##           0           1           2           3           <NA>
## 0.581905415 0.400274160 0.015764222 0.000685401 0.001370802
```

```
df_test$BsmtFullBath[is.na(df_test$BsmtFullBath)] <- 0
```

```
#BsmtHalfBath
```

```
prop.table(table(df_test$BsmtHalfBath, useNA = "ifany"))
```

```
##
##           0           1           2           <NA>
## 0.934886909 0.062371487 0.001370802 0.001370802
```

```
df_test$BsmtHalfBath[is.na(df_test$BsmtHalfBath)] <- 0
```

```
#KitchenQual
```

```
prop.table(table(df_test$KitchenQual, useNA = "ifany"))
```

```
##
##           Ex           Fa           Gd           TA           <NA>
## 0.071967101 0.021247430 0.387251542 0.518848526 0.000685401
```

```
df_test$KitchenQual[is.na(df_test$KitchenQual)] <- "None"
```

```
#Functional
```

```
prop.table(table(df_test$Functional, useNA = "ifany"))
```

```
##
##           Maj1           Maj2           Min1           Min2           Mod           Sev
## 0.003427005 0.002741604 0.023303633 0.024674435 0.013708019 0.000685401
##           Typ           <NA>
## 0.930089102 0.001370802
```

```
df_test$Functional[is.na(df_test$Functional)] <- "None"
```

```
#FireplaceQu
```

```
prop.table(table(df_test$FireplaceQu, useNA = "ifany"))
```

```
##
##           Ex           Fa           Gd           Po           TA           <NA>
## 0.01302262 0.02810144 0.24948595 0.01782042 0.19122687 0.50034270
```

```
df_test$FireplaceQu[is.na(df_test$FireplaceQu)] <- "None"
```

```
#GarageType
```

```
prop.table(table(df_test$GarageType, useNA = "ifany"))
```

```
##
##           2Types           Attchd           Basement           BuiltIn           CarPort           Detchd
## 0.011651816 0.584647019 0.011651816 0.067169294 0.004112406 0.268677176
##           <NA>
## 0.052090473
```

```
df_test$GarageType[is.na(df_test$GarageType)] <- "None"
```

```
#GarageYrBlt
```

```
prop.table(table(df_test$GarageYrBlt, useNA = "ifany"))
```

```
##
##      1895      1896      1900      1910      1915      1916
## 0.000685401 0.000685401 0.003427005 0.004797807 0.003427005 0.000685401
##      1917      1918      1919      1920      1921      1922
## 0.001370802 0.000685401 0.000685401 0.013022618 0.001370802 0.002056203
##      1923      1924      1925      1926      1927      1928
## 0.002056203 0.003427005 0.003427005 0.006168609 0.002741604 0.002056203
##      1930      1932      1934      1935      1936      1937
## 0.013022618 0.000685401 0.001370802 0.002741604 0.001370802 0.002741604
##      1938      1939      1940      1941      1942      1943
## 0.005483208 0.008224812 0.007539411 0.002741604 0.002741604 0.000685401
##      1945      1946      1947      1948      1949      1950
## 0.004112406 0.003427005 0.002056203 0.005483208 0.004112406 0.018505826
##      1951      1952      1953      1954      1955      1956
## 0.007539411 0.008910212 0.007539411 0.012337217 0.007539411 0.017135024
##      1957      1958      1959      1960      1961      1962
## 0.009595613 0.014393420 0.013022618 0.012337217 0.012337217 0.009595613
##      1963      1964      1965      1966      1967      1968
## 0.012337217 0.011651816 0.008910212 0.012337217 0.014393420 0.015078821
##      1969      1970      1971      1972      1973      1974
## 0.011651816 0.008224812 0.007539411 0.008910212 0.010281014 0.011651816
##      1975      1976      1977      1978      1979      1980
## 0.013022618 0.014393420 0.021247430 0.015078821 0.013708019 0.011651816
##      1981      1982      1983      1984      1985      1986
## 0.003427005 0.003427005 0.002741604 0.007539411 0.005483208 0.004112406
##      1987      1988      1989      1990      1991      1992
## 0.004797807 0.004112406 0.006168609 0.006854010 0.005483208 0.009595613
##      1993      1994      1995      1996      1997      1998
## 0.018505826 0.014393420 0.011651816 0.013708019 0.017135024 0.018505826
##      1999      2000      2001      2002      2003      2004
## 0.016449623 0.019191227 0.014393420 0.018505826 0.028786840 0.031528444
##      2005      2006      2007      2008      2009      2010
## 0.052775874 0.038382454 0.045236463 0.021932831 0.005483208 0.001370802
##      2207      <NA>
## 0.000685401 0.053461275
```

```
#An NA Garage Year Built seems to tie with an NA Garage Type, therefore its NAs are set to 0
#No need to address GarageType NAs since the variable is not numeric
```

```
df_test$GarageYrBlt[is.na(df_test$GarageYrBlt)] <- 0
```

```
#GarageFinish
```

```
prop.table(table(df_test$GarageFinish, useNA = "ifany"))
```

```
##
##      Fin      RFn      Unf      <NA>
## 0.25154215 0.26662097 0.42837560 0.05346127
```



```
df_test$GarageFinish[is.na(df_test$GarageFinish)] <- "None"
```

```
#GarageCars
```

```
prop.table(table(df_test$GarageCars, useNA = "ifany"))
```

```
##
##           0           1           2           3           4           5
## 0.052090473 0.278958191 0.527758739 0.132282385 0.007539411 0.000685401
##      <NA>
## 0.000685401
```

```
df_test$GarageCars[is.na(df_test$GarageCars)] <- 0
```

```
#GarageArea
```

```
prop.table(table(df_test$GarageArea, useNA = "ifany"))
```

```
##
##           0           100           160           162           164           180
## 0.052090473 0.000685401 0.000685401 0.001370802 0.000685401 0.004797807
##           184           185           195           200           205           207
## 0.000685401 0.000685401 0.002056203 0.004797807 0.000685401 0.000685401
##           209           210           215           216           217           220
## 0.000685401 0.000685401 0.002056203 0.007539411 0.000685401 0.000685401
##           224           225           226           228           230           231
## 0.001370802 0.002056203 0.000685401 0.001370802 0.001370802 0.002056203
##           234           240           242           246           249           250
## 0.000685401 0.021247430 0.000685401 0.001370802 0.000685401 0.002056203
##           252           253           256           257           258           260
## 0.004112406 0.002741604 0.002741604 0.000685401 0.000685401 0.001370802
##           263           264           265           266           267           270
## 0.000685401 0.018505826 0.000685401 0.000685401 0.000685401 0.002741604
##           272           273           275           276           280           281
## 0.000685401 0.000685401 0.001370802 0.001370802 0.009595613 0.000685401
##           283           286           287           288           292           293
## 0.000685401 0.007539411 0.000685401 0.015764222 0.000685401 0.000685401
##           294           295           297           299           300           301
## 0.004112406 0.000685401 0.004797807 0.002056203 0.004112406 0.000685401
##           303           304           305           307           308           310
## 0.000685401 0.000685401 0.001370802 0.000685401 0.019191227 0.001370802
##           311           312           313           315           316           317
## 0.001370802 0.006854010 0.001370802 0.003427005 0.000685401 0.000685401
##           318           319           320           322           323           324
## 0.001370802 0.002741604 0.003427005 0.001370802 0.000685401 0.001370802
##           326           330           331           332           336           338
## 0.000685401 0.002056203 0.001370802 0.000685401 0.011651816 0.000685401
##           340           342           343           344           345           350
## 0.000685401 0.002056203 0.000685401 0.000685401 0.000685401 0.002056203
##           351           352           353           355           356           357
## 0.001370802 0.003427005 0.000685401 0.001370802 0.000685401 0.001370802
##           360           363           364           365           366           368
## 0.003427005 0.000685401 0.000685401 0.001370802 0.001370802 0.001370802
##           369           370           371           372           374           378
```

##	0.000685401	0.002056203	0.001370802	0.000685401	0.000685401	0.000685401
##	379	380	384	386	388	390
##	0.002056203	0.000685401	0.008224812	0.000685401	0.002056203	0.005483208
##	392	393	394	396	397	398
##	0.002741604	0.002741604	0.001370802	0.002056203	0.000685401	0.001370802
##	399	400	401	402	403	404
##	0.001370802	0.022618232	0.000685401	0.000685401	0.000685401	0.000685401
##	408	409	410	412	416	418
##	0.000685401	0.000685401	0.002741604	0.000685401	0.001370802	0.001370802
##	420	423	427	428	429	430
##	0.006854010	0.000685401	0.000685401	0.000685401	0.001370802	0.002741604
##	431	432	433	434	435	436
##	0.002056203	0.002056203	0.002056203	0.002741604	0.001370802	0.001370802
##	437	438	440	441	442	443
##	0.002741604	0.001370802	0.032213845	0.004797807	0.000685401	0.000685401
##	444	449	450	451	452	453
##	0.001370802	0.000685401	0.003427005	0.002056203	0.002056203	0.000685401
##	454	456	460	461	462	463
##	0.000685401	0.000685401	0.004797807	0.002741604	0.008910212	0.000685401
##	464	465	466	467	468	469
##	0.001370802	0.000685401	0.000685401	0.002056203	0.000685401	0.000685401
##	470	471	472	473	474	476
##	0.004797807	0.000685401	0.003427005	0.004797807	0.001370802	0.000685401
##	477	478	479	480	481	482
##	0.002056203	0.005483208	0.000685401	0.020562029	0.000685401	0.004797807
##	483	484	485	486	487	488
##	0.002056203	0.023303633	0.000685401	0.002056203	0.000685401	0.001370802
##	489	490	492	494	495	496
##	0.001370802	0.002056203	0.002056203	0.000685401	0.008910212	0.001370802
##	497	498	499	501	502	504
##	0.000685401	0.002056203	0.001370802	0.001370802	0.004112406	0.006168609
##	506	509	510	511	512	513
##	0.009595613	0.000685401	0.001370802	0.002056203	0.003427005	0.000685401
##	515	516	517	518	520	521
##	0.002741604	0.001370802	0.003427005	0.000685401	0.003427005	0.002056203
##	522	523	524	525	527	528
##	0.002056203	0.000685401	0.002056203	0.007539411	0.002056203	0.021932831
##	529	530	531	532	534	535
##	0.002741604	0.002741604	0.001370802	0.000685401	0.002056203	0.000685401
##	539	540	541	542	543	544
##	0.002056203	0.002741604	0.001370802	0.004797807	0.000685401	0.000685401
##	545	546	549	550	551	552
##	0.001370802	0.001370802	0.000685401	0.006168609	0.002056203	0.002741604
##	554	555	556	557	558	559
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	560	561	564	565	566	567
##	0.001370802	0.001370802	0.003427005	0.000685401	0.000685401	0.000685401
##	568	569	570	571	572	574
##	0.000685401	0.000685401	0.002056203	0.000685401	0.003427005	0.000685401
##	575	576	577	578	579	580
##	0.002741604	0.034270048	0.000685401	0.004112406	0.001370802	0.002741604
##	581	582	583	584	585	586
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802
##	588	590	591	592	594	596

##	0.004112406	0.000685401	0.002056203	0.000685401	0.000685401	0.002056203
##	597	598	599	600	605	608
##	0.001370802	0.002056203	0.000685401	0.001370802	0.000685401	0.002056203
##	609	610	612	614	615	616
##	0.000685401	0.003427005	0.002056203	0.000685401	0.001370802	0.004112406
##	617	618	619	620	621	622
##	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	623	624	625	626	627	628
##	0.000685401	0.0006854010	0.002741604	0.001370802	0.002056203	0.001370802
##	630	631	632	636	638	640
##	0.000685401	0.001370802	0.002056203	0.001370802	0.001370802	0.001370802
##	641	642	644	646	647	648
##	0.000685401	0.002056203	0.001370802	0.000685401	0.000685401	0.000685401
##	649	650	656	658	660	662
##	0.002056203	0.002056203	0.001370802	0.002056203	0.003427005	0.000685401
##	663	666	668	670	672	674
##	0.000685401	0.000685401	0.001370802	0.002741604	0.005483208	0.000685401
##	675	676	678	682	683	684
##	0.000685401	0.002056203	0.002056203	0.000685401	0.000685401	0.000685401
##	685	686	687	688	690	691
##	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	692	698	700	701	704	706
##	0.002056203	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401
##	711	713	714	715	720	721
##	0.000685401	0.000685401	0.000685401	0.000685401	0.004112406	0.000685401
##	722	724	725	726	728	729
##	0.000685401	0.001370802	0.001370802	0.000685401	0.001370802	0.000685401
##	730	732	736	738	741	744
##	0.002741604	0.001370802	0.000685401	0.000685401	0.000685401	0.000685401
##	746	747	748	751	754	756
##	0.001370802	0.000685401	0.000685401	0.001370802	0.001370802	0.001370802
##	758	760	762	768	773	774
##	0.001370802	0.000685401	0.001370802	0.001370802	0.000685401	0.001370802
##	776	780	782	783	784	786
##	0.000685401	0.001370802	0.000685401	0.000685401	0.003427005	0.002056203
##	787	788	789	791	792	795
##	0.000685401	0.001370802	0.000685401	0.000685401	0.001370802	0.002056203
##	796	803	806	810	811	814
##	0.000685401	0.000685401	0.000685401	0.002056203	0.000685401	0.001370802
##	815	816	818	820	828	831
##	0.000685401	0.002056203	0.000685401	0.002056203	0.001370802	0.000685401
##	834	836	843	844	845	846
##	0.000685401	0.000685401	0.000685401	0.001370802	0.000685401	0.001370802
##	848	850	851	852	856	859
##	0.000685401	0.002741604	0.000685401	0.001370802	0.000685401	0.000685401
##	864	868	869	870	871	874
##	0.002741604	0.000685401	0.000685401	0.002056203	0.000685401	0.001370802
##	876	878	880	885	886	888
##	0.000685401	0.001370802	0.002741604	0.001370802	0.000685401	0.000685401
##	892	894	896	898	899	900
##	0.000685401	0.000685401	0.001370802	0.000685401	0.000685401	0.000685401
##	904	905	907	912	916	920
##	0.000685401	0.000685401	0.000685401	0.000685401	0.000685401	0.001370802
##	925	927	928	932	933	938

```
## 0.001370802 0.000685401 0.001370802 0.002056203 0.000685401 0.001370802
##      944      949      958      959      962      963
## 0.001370802 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      972      984      1003      1008      1017      1040
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      1041      1085      1092      1105      1110      1138
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      1150      1154      1174      1184      1200      1231
## 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      1314      1348      1488      <NA>
## 0.000685401 0.000685401 0.000685401 0.000685401
```

```
df_test$GarageArea[is.na(df_test$GarageArea)] <- 0
```

```
#####
table(df_test$GarageQual, useNA = "ifany")
```

```
##
##   Fa   Gd   Po   TA <NA>
##   76   10    2 1293    78
```

```
table(df_test$GarageCond, useNA = "ifany")
```

```
##
##   Ex   Fa   Gd   Po   TA <NA>
##    1   39    6    7 1328    78
```

```
#GarageQual
prop.table(table(df_test$GarageQual, useNA = "ifany"))
```

```
##
##           Fa           Gd           Po           TA           <NA>
## 0.052090473 0.006854010 0.001370802 0.886223441 0.053461275
```

```
df_test$GarageQual[is.na(df_test$GarageQual)] <- "None"
```

```
#GarageCond
prop.table(table(df_test$GarageCond, useNA = "ifany"))
```

```
##
##           Ex           Fa           Gd           Po           TA           <NA>
## 0.000685401 0.026730637 0.004112406 0.004797807 0.910212474 0.053461275
```

```
df_test$GarageCond[is.na(df_test$GarageCond)] <- "None"
#####
```

```
#PoolQC
prop.table(table(df_test$PoolQC, useNA = "ifany"))
```

```
##
##           Ex           Gd           <NA>
## 0.001370802 0.000685401 0.997943797
```

```
df_test$PoolQC[is.na(df_test$PoolQC)] <- "None"
```

```
#Fence
```

```
prop.table(table(df_test$Fence, useNA = "ifany"))
```

```
##
```

```
##      GdPrv      GdWo      MnPrv      MnWw      <NA>
```

```
## 0.040438657 0.039753256 0.117888965 0.000685401 0.801233722
```

```
df_test$Fence[is.na(df_test$Fence)] <- "None"
```

```
#MiscFeature
```

```
prop.table(table(df_test$MiscFeature, useNA = "ifany"))
```

```
##
```

```
##      Gar2      Othr      Shed      <NA>
```

```
## 0.002056203 0.001370802 0.031528444 0.965044551
```

```
df_test$MiscFeature[is.na(df_test$MiscFeature)] <- "None"
```

```
#SaleType
```

```
prop.table(table(df_test$SaleType, useNA = "ifany"))
```

```
##
```

```
##      COD      Con      ConLD      ConLI      ConLw      CWD
```

```
## 0.030157642 0.002056203 0.011651816 0.002741604 0.002056203 0.005483208
```

```
##      New      Oth      WD      <NA>
```

```
## 0.080191912 0.002741604 0.862234407 0.000685401
```

```
df_test$SaleType[is.na(df_test$SaleType)] <- "Oth"
```

```
colSums(is.na(df_test))
```

```
##      Id      MSSubClass      MSZoning      LotFrontage      LotArea
##      0      0      0      0      0
##      Street      Alley      LotShape      LandContour      Utilities
##      0      0      0      0      0
##      LotConfig      LandSlope      Neighborhood      Condition1      Condition2
##      0      0      0      0      0
##      BldgType      HouseStyle      OverallQual      OverallCond      YearBuilt
##      0      0      0      0      0
##      YearRemodAdd      RoofStyle      RoofMatl      Exterior1st      Exterior2nd
##      0      0      0      0      0
##      MasVnrType      MasVnrArea      ExterQual      ExterCond      Foundation
##      0      0      0      0      0
##      BsmtQual      BsmtCond      BsmtExposure      BsmtFinType1      BsmtFinSF1
##      0      0      0      0      0
##      BsmtFinType2      BsmtFinSF2      BsmtUnfSF      TotalBsmtSF      Heating
##      0      0      0      0      0
##      HeatingQC      CentralAir      Electrical      X1stFlrSF      X2ndFlrSF
##      0      0      0      0      0
```

```
## LowQualFinSF      GrLivArea BsmtFullBath BsmtHalfBath      FullBath
##           0           0           0           0           0
##      HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd
##           0           0           0           0           0
##      Functional   Fireplaces   FireplaceQu   GarageType   GarageYrBlt
##           0           0           0           0           0
## GarageFinish   GarageCars   GarageArea   GarageQual   GarageCond
##           0           0           0           0           0
##      PavedDrive   WoodDeckSF   OpenPorchSF EnclosedPorch   X3SsnPorch
##           0           0           0           0           0
##      ScreenPorch   PoolArea           PoolQC           Fence   MiscFeature
##           0           0           0           0           0
##      MiscVal      MoSold      YrSold      SaleType SaleCondition
##           0           0           0           0           0
```

```
table(is.na(df_test))
```

```
##
## FALSE
## 116720
```

```
####test: Factoring columns according to data_description.txt####
```

```
#Seeking out numeric and non-numeric columns
str(df_test)
```

```
## 'data.frame':    1459 obs. of  80 variables:
## $ Id             : int  1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 ...
## $ MSSubClass     : int  20 20 60 60 120 60 20 60 20 20 ...
## $ MSZoning       : chr  "RH" "RL" "RL" "RL" ...
## $ LotFrontage    : num  80 81 74 78 43 ...
## $ LotArea        : int  11622 14267 13830 9978 5005 10000 7980 8402 10176 8400 ...
## $ Street         : chr  "Pave" "Pave" "Pave" "Pave" ...
## $ Alley          : chr  "None" "None" "None" "None" ...
## $ LotShape       : chr  "Reg" "IR1" "IR1" "IR1" ...
## $ LandContour    : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
## $ Utilities      : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
## $ LotConfig      : chr  "Inside" "Corner" "Inside" "Inside" ...
## $ LandSlope      : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
## $ Neighborhood   : chr  "Names" "Names" "Gilbert" "Gilbert" ...
## $ Condition1     : chr  "Feedr" "Norm" "Norm" "Norm" ...
## $ Condition2     : chr  "Norm" "Norm" "Norm" "Norm" ...
## $ BldgType       : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
## $ HouseStyle     : chr  "1Story" "1Story" "2Story" "2Story" ...
## $ OverallQual    : int  5 6 5 6 8 6 6 6 7 4 ...
## $ OverallCond    : int  6 6 5 6 5 5 7 5 5 5 ...
## $ YearBuilt      : int  1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...
## $ YearRemodAdd   : int  1961 1958 1998 1998 1992 1994 2007 1998 1990 1970 ...
## $ RoofStyle      : chr  "Gable" "Hip" "Gable" "Gable" ...
## $ RoofMatl       : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
## $ Exterior1st    : chr  "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...
## $ Exterior2nd    : chr  "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...
## $ MasVnrType     : chr  "None" "BrkFace" "None" "BrkFace" ...
```

```

## $ MasVnrArea : num 0 108 0 20 0 0 0 0 0 0 ...
## $ ExterQual : chr "TA" "TA" "TA" "TA" ...
## $ ExterCond : chr "TA" "TA" "TA" "TA" ...
## $ Foundation : chr "CBlock" "CBlock" "PConc" "PConc" ...
## $ BsmtQual : chr "TA" "TA" "Gd" "TA" ...
## $ BsmtCond : chr "TA" "TA" "TA" "TA" ...
## $ BsmtExposure : chr "No" "No" "No" "No" ...
## $ BsmtFinType1 : chr "Rec" "ALQ" "GLQ" "GLQ" ...
## $ BsmtFinSF1 : num 468 923 791 602 263 0 935 0 637 804 ...
## $ BsmtFinType2 : chr "LwQ" "Unf" "Unf" "Unf" ...
## $ BsmtFinSF2 : num 144 0 0 0 0 0 0 0 78 ...
## $ BsmtUnfSF : num 270 406 137 324 1017 ...
## $ TotalBsmtSF : num 882 1329 928 926 1280 ...
## $ Heating : chr "GasA" "GasA" "GasA" "GasA" ...
## $ HeatingQC : chr "TA" "TA" "Gd" "Ex" ...
## $ CentralAir : chr "Y" "Y" "Y" "Y" ...
## $ Electrical : chr "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
## $ X1stFlrSF : int 896 1329 928 926 1280 763 1187 789 1341 882 ...
## $ X2ndFlrSF : int 0 0 701 678 0 892 0 676 0 0 ...
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea : int 896 1329 1629 1604 1280 1655 1187 1465 1341 882 ...
## $ BsmtFullBath : num 0 0 0 0 0 0 1 0 1 1 ...
## $ BsmtHalfBath : num 0 0 0 0 0 0 0 0 0 0 ...
## $ FullBath : int 1 1 2 2 2 2 2 2 1 1 ...
## $ HalfBath : int 0 1 1 1 0 1 0 1 1 0 ...
## $ BedroomAbvGr : int 2 3 3 3 2 3 3 3 2 2 ...
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 1 1 ...
## $ KitchenQual : chr "TA" "Gd" "TA" "Gd" ...
## $ TotRmsAbvGrd : int 5 6 6 7 5 7 6 7 5 4 ...
## $ Functional : chr "Typ" "Typ" "Typ" "Typ" ...
## $ Fireplaces : int 0 0 1 1 0 1 0 1 1 0 ...
## $ FireplaceQu : chr "None" "None" "TA" "Gd" ...
## $ GarageType : chr "Attchd" "Attchd" "Attchd" "Attchd" ...
## $ GarageYrBlt : num 1961 1958 1997 1998 1992 ...
## $ GarageFinish : chr "Unf" "Unf" "Fin" "Fin" ...
## $ GarageCars : num 1 1 2 2 2 2 2 2 2 ...
## $ GarageArea : num 730 312 482 470 506 440 420 393 506 525 ...
## $ GarageQual : chr "TA" "TA" "TA" "TA" ...
## $ GarageCond : chr "TA" "TA" "TA" "TA" ...
## $ PavedDrive : chr "Y" "Y" "Y" "Y" ...
## $ WoodDeckSF : int 140 393 212 360 0 157 483 0 192 240 ...
## $ OpenPorchSF : int 0 36 34 36 82 84 21 75 0 0 ...
## $ EnclosedPorch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ X3SsnPorch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ScreenPorch : int 120 0 0 0 144 0 0 0 0 0 ...
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PoolQC : chr "None" "None" "None" "None" ...
## $ Fence : chr "MnPrv" "None" "MnPrv" "None" ...
## $ MiscFeature : chr "None" "Gar2" "None" "None" ...
## $ MiscVal : int 0 12500 0 0 0 0 500 0 0 0 ...
## $ MoSold : int 6 6 3 6 1 4 3 5 2 4 ...
## $ YrSold : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
## $ SaleType : chr "WD" "WD" "WD" "WD" ...
## $ SaleCondition : chr "Normal" "Normal" "Normal" "Normal" ...

```

```
names(df_test)[sapply(df_test, is.numeric)]
```

```
## [1] "Id"           "MSSubClass"    "LotFrontage"   "LotArea"
## [5] "OverallQual"  "OverallCond"   "YearBuilt"     "YearRemodAdd"
## [9] "MasVnrArea"   "BsmtFinSF1"    "BsmtFinSF2"    "BsmtUnfSF"
## [13] "TotalBsmtSF"  "X1stFlrSF"     "X2ndFlrSF"     "LowQualFinSF"
## [17] "GrLivArea"    "BsmtFullBath"  "BsmtHalfBath"  "FullBath"
## [21] "HalfBath"     "BedroomAbvGr"  "KitchenAbvGr"  "TotRmsAbvGrd"
## [25] "Fireplaces"   "GarageYrBlt"   "GarageCars"    "GarageArea"
## [29] "WoodDeckSF"   "OpenPorchSF"   "EnclosedPorch" "X3SsnPorch"
## [33] "ScreenPorch"  "PoolArea"      "MiscVal"       "MoSold"
## [37] "YrSold"
```

```
names(df_test)[sapply(df_test, is.character)]
```

```
## [1] "MSZoning"      "Street"        "Alley"         "LotShape"
## [5] "LandContour"   "Utilities"     "LotConfig"     "LandSlope"
## [9] "Neighborhood"  "Condition1"    "Condition2"    "BldgType"
## [13] "HouseStyle"    "RoofStyle"     "RoofMatl"      "Exterior1st"
## [17] "Exterior2nd"   "MasVnrType"    "ExterQual"     "ExterCond"
## [21] "Foundation"    "BsmtQual"      "BsmtCond"      "BsmtExposure"
## [25] "BsmtFinType1"  "BsmtFinType2"  "Heating"       "HeatingQC"
## [29] "CentralAir"    "Electrical"    "KitchenQual"   "Functional"
## [33] "FireplaceQu"   "GarageType"    "GarageFinish"  "GarageQual"
## [37] "GarageCond"    "PavedDrive"    "PoolQC"        "Fence"
## [41] "MiscFeature"   "SaleType"      "SaleCondition"
```

```
#Character variables into factors
```

```
df_test[sapply(df_test, is.character)] <- lapply(df_test[sapply(df_test, is.character)], as.factor)
```

```
#This also applies for columns with both chars and nums, which is still
```

```
#consistent with the actual meaning of such columns that are meant for factoring
```

```
#Sanity check
```

```
str(df_test[sapply(df_test, is.character)])
```

```
## 'data.frame': 1459 obs. of 0 variables
```

```
str(df_test[sapply(df_test, is.numeric)])
```

```
## 'data.frame': 1459 obs. of 37 variables:
## $ Id : int 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 ...
## $ MSSubClass : int 20 20 60 60 120 60 20 60 20 20 ...
## $ LotFrontage : num 80 81 74 78 43 ...
## $ LotArea : int 11622 14267 13830 9978 5005 10000 7980 8402 10176 8400 ...
## $ OverallQual : int 5 6 5 6 8 6 6 7 4 ...
## $ OverallCond : int 6 6 5 6 5 5 7 5 5 5 ...
## $ YearBuilt : int 1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...
## $ YearRemodAdd : int 1961 1958 1998 1998 1992 1994 2007 1998 1990 1970 ...
## $ MasVnrArea : num 0 108 0 20 0 0 0 0 0 0 ...
## $ BsmtFinSF1 : num 468 923 791 602 263 0 935 0 637 804 ...
## $ BsmtFinSF2 : num 144 0 0 0 0 0 0 0 0 78 ...
```



```
## $ BsmtUnfSF : num 270 406 137 324 1017 ...
## $ TotalBsmtSF : num 882 1329 928 926 1280 ...
## $ X1stFlrSF : int 896 1329 928 926 1280 763 1187 789 1341 882 ...
## $ X2ndFlrSF : int 0 0 701 678 0 892 0 676 0 0 ...
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea : int 896 1329 1629 1604 1280 1655 1187 1465 1341 882 ...
## $ BsmtFullBath : num 0 0 0 0 0 0 1 0 1 1 ...
## $ BsmtHalfBath : num 0 0 0 0 0 0 0 0 0 0 ...
## $ FullBath : int 1 1 2 2 2 2 2 1 1 ...
## $ HalfBath : int 0 1 1 1 0 1 0 1 1 0 ...
## $ BedroomAbvGr : int 2 3 3 3 2 3 3 2 2 ...
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 1 ...
## $ TotRmsAbvGrd : int 5 6 6 7 5 7 6 7 5 4 ...
## $ Fireplaces : int 0 0 1 1 0 1 0 1 1 0 ...
## $ GarageYrBlt : num 1961 1958 1997 1998 1992 ...
## $ GarageCars : num 1 1 2 2 2 2 2 2 2 ...
## $ GarageArea : num 730 312 482 470 506 440 420 393 506 525 ...
## $ WoodDeckSF : int 140 393 212 360 0 157 483 0 192 240 ...
## $ OpenPorchSF : int 0 36 34 36 82 84 21 75 0 0 ...
## $ EnclosedPorch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ X3SsnPorch : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ScreenPorch : int 120 0 0 0 144 0 0 0 0 0 ...
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...
## $ MiscVal : int 0 12500 0 0 0 0 500 0 0 0 ...
## $ MoSold : int 6 6 3 6 1 4 3 5 2 4 ...
## $ YrSold : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
```

```
str(df_test[sapply(df_test, is.factor)])
```

```
## 'data.frame': 1459 obs. of 43 variables:
## $ MSZoning : Factor w/ 6 levels "C (all)","FV",...: 4 5 5 5 5 5 5 5 5 5 ...
## $ Street : Factor w/ 2 levels "Grv1","Pave": 2 2 2 2 2 2 2 2 2 2 ...
## $ Alley : Factor w/ 3 levels "Grv1","None",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ LotShape : Factor w/ 4 levels "IR1","IR2","IR3",...: 4 1 1 1 1 1 1 1 4 4 ...
## $ LandContour : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 2 4 4 4 4 4 ...
## $ Utilities : Factor w/ 2 levels "AllPub","None": 1 1 1 1 1 1 1 1 1 1 ...
## $ LotConfig : Factor w/ 5 levels "Corner","CulDSac",...: 5 1 5 5 5 1 5 5 5 1 ...
## $ LandSlope : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",...: 13 13 9 9 22 9 9 9 13 ...
## $ Condition1 : Factor w/ 9 levels "Artery","Feedr",...: 2 3 3 3 3 3 3 3 3 ...
## $ Condition2 : Factor w/ 5 levels "Artery","Feedr",...: 3 3 3 3 3 3 3 3 3 ...
## $ BldgType : Factor w/ 5 levels "1Fam","2fmCon",...: 1 1 1 1 5 1 1 1 1 1 ...
## $ HouseStyle : Factor w/ 7 levels "1.5Fin","1.5Unf",...: 3 3 5 5 3 5 3 5 3 3 ...
## $ RoofStyle : Factor w/ 6 levels "Flat","Gable",...: 2 4 2 2 2 2 2 2 2 2 ...
## $ RoofMatl : Factor w/ 4 levels "CompShg","Tar&Grv",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Exterior1st : Factor w/ 14 levels "AsbShng","AsphShn",...: 12 13 12 12 7 7 7 12 7 10 ...
## $ Exterior2nd : Factor w/ 16 levels "AsbShng","AsphShn",...: 14 15 14 14 7 7 7 14 7 11 ...
## $ MasVnrType : Factor w/ 4 levels "BrkCmn","BrkFace",...: 3 2 3 2 3 3 3 3 3 3 ...
## $ ExterQual : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 4 3 4 4 4 4 4 ...
## $ ExterCond : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 3 5 5 5 ...
## $ Foundation : Factor w/ 6 levels "BrkTil","CBBlock",...: 2 2 3 3 3 3 3 3 2 2 ...
## $ BsmtQual : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 5 3 3 3 3 5 5 ...
## $ BsmtCond : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ BsmtExposure : Factor w/ 5 levels "Av","Gd","Mn",...: 4 4 4 4 4 4 4 4 2 4 ...
```

```
## $ BsmtFinType1 : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 6 1 3 3 1 7 1 7 3 1 ...
## $ BsmtFinType2 : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 4 7 7 7 7 7 7 7 6 ...
## $ Heating      : Factor w/ 4 levels "GasA","GasW",...: 1 1 1 1 1 1 1 1 1 ...
## $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 1 1 3 1 3 3 5 ...
## $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 ...
## $ Electrical   : Factor w/ 4 levels "FuseA","FuseF",...: 4 4 4 4 4 4 4 4 4 ...
## $ KitchenQual  : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 3 5 3 3 5 5 5 3 5 ...
## $ Functional   : Factor w/ 8 levels "Maj1","Maj2",...: 8 8 8 8 8 8 8 8 8 ...
## $ FireplaceQu  : Factor w/ 6 levels "Ex","Fa","Gd",...: 4 4 6 3 4 6 4 3 5 4 ...
## $ GarageType   : Factor w/ 7 levels "2Types","Attchd",...: 2 2 2 2 2 2 2 2 2 ...
## $ GarageFinish : Factor w/ 4 levels "Fin","None","RFn",...: 4 4 1 1 3 1 1 1 4 1 ...
## $ GarageQual   : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 5 5 5 5 5 5 ...
## $ GarageCond   : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 6 ...
## $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 ...
## $ PoolQC       : Factor w/ 3 levels "Ex","Gd","None": 3 3 3 3 3 3 3 3 3 ...
## $ Fence        : Factor w/ 5 levels "GdPrv","GdWo",...: 3 5 3 5 5 5 1 5 5 3 ...
## $ MiscFeature  : Factor w/ 4 levels "Gar2","None",...: 2 1 2 2 2 2 4 2 2 2 ...
## $ SaleType     : Factor w/ 9 levels "COD","Con","ConLD",...: 9 9 9 9 9 9 9 9 9 ...
## $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",...: 5 5 5 5 5 5 5 5 5 ...
```

```
#Numeric columns at this point mistakenly includes obscure factor-able columns
#like MSSubClass, OverallQual, OverallCond
#Let's try and fix that
```

```
#MSSubClass
```

```
df_test$MSSubClass <- as.factor(df_test$MSSubClass)
str(df_test$MSSubClass)
```

```
## Factor w/ 16 levels "20","30","40",...: 1 1 6 6 12 6 1 6 1 1 ...
```

```
table(df_test$MSSubClass)
```

```
##
## 20 30 40 45 50 60 70 75 80 85 90 120 150 160 180 190
## 543 70 2 6 143 276 68 7 60 28 57 95 1 65 7 31
```

```
#OverallQual
```

```
df_test$OverallQual <- as.factor(df_test$OverallQual)
str(df_test$OverallQual)
```

```
## Factor w/ 10 levels "1","2","3","4",...: 5 6 5 6 8 6 6 6 7 4 ...
```

```
#OverallCond
```

```
df_test$OverallCond <- as.factor(df_test$OverallCond)
str(df_test$OverallCond)
```

```
## Factor w/ 9 levels "1","2","3","4",...: 6 6 5 6 5 5 7 5 5 ...
```

```
#Sanity check 2
```

```
str(df_test[sapply(df_test, is.numeric)])
```

```
## 'data.frame':    1459 obs. of  34 variables:
## $ Id             : int  1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 ...
## $ LotFrontage    : num  80 81 74 78 43 ...
## $ LotArea        : int  11622 14267 13830 9978 5005 10000 7980 8402 10176 8400 ...
## $ YearBuilt      : int  1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...
## $ YearRemodAdd   : int  1961 1958 1998 1998 1992 1994 2007 1998 1990 1970 ...
## $ MasVnrArea     : num  0 108 0 20 0 0 0 0 0 0 ...
## $ BsmtFinSF1     : num  468 923 791 602 263 0 935 0 637 804 ...
## $ BsmtFinSF2     : num  144 0 0 0 0 0 0 0 0 78 ...
## $ BsmtUnfSF      : num  270 406 137 324 1017 ...
## $ TotalBsmtSF    : num  882 1329 928 926 1280 ...
## $ X1stFlrSF      : int  896 1329 928 926 1280 763 1187 789 1341 882 ...
## $ X2ndFlrSF      : int  0 0 701 678 0 892 0 676 0 0 ...
## $ LowQualFinSF   : int  0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea      : int  896 1329 1629 1604 1280 1655 1187 1465 1341 882 ...
## $ BsmtFullBath   : num  0 0 0 0 0 0 1 0 1 1 ...
## $ BsmtHalfBath   : num  0 0 0 0 0 0 0 0 0 0 ...
## $ FullBath       : int  1 1 2 2 2 2 2 2 1 1 ...
## $ HalfBath       : int  0 1 1 1 0 1 0 1 1 0 ...
## $ BedroomAbvGr   : int  2 3 3 3 2 3 3 3 2 2 ...
## $ KitchenAbvGr   : int  1 1 1 1 1 1 1 1 1 1 ...
## $ TotRmsAbvGrd   : int  5 6 6 7 5 7 6 7 5 4 ...
## $ Fireplaces     : int  0 0 1 1 0 1 0 1 1 0 ...
## $ GarageYrBlt    : num  1961 1958 1997 1998 1992 ...
## $ GarageCars     : num  1 1 2 2 2 2 2 2 2 2 ...
## $ GarageArea     : num  730 312 482 470 506 440 420 393 506 525 ...
## $ WoodDeckSF     : int  140 393 212 360 0 157 483 0 192 240 ...
## $ OpenPorchSF    : int  0 36 34 36 82 84 21 75 0 0 ...
## $ EnclosedPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
## $ X3SsnPorch     : int  0 0 0 0 0 0 0 0 0 0 ...
## $ ScreenPorch    : int  120 0 0 0 144 0 0 0 0 0 ...
## $ PoolArea       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ MiscVal        : int  0 12500 0 0 0 0 500 0 0 0 ...
## $ MoSold         : int  6 6 3 6 1 4 3 5 2 4 ...
## $ YrSold         : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
```

```
str(df_test[sapply(df_test, is.factor)])
```

```
## 'data.frame':    1459 obs. of  46 variables:
## $ MSSubClass     : Factor w/ 16 levels "20","30","40",...: 1 1 6 6 12 6 1 6 1 1 ...
## $ MSZoning       : Factor w/ 6 levels "C (all)","FV",...: 4 5 5 5 5 5 5 5 5 5 ...
## $ Street         : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
## $ Alley         : Factor w/ 3 levels "Grvl","None",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ LotShape       : Factor w/ 4 levels "IR1","IR2","IR3",...: 4 1 1 1 1 1 1 1 4 4 ...
## $ LandContour    : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 2 4 4 4 4 4 ...
## $ Utilities     : Factor w/ 2 levels "AllPub","None": 1 1 1 1 1 1 1 1 1 1 ...
## $ LotConfig     : Factor w/ 5 levels "Corner","CulDSac",...: 5 1 5 5 5 1 5 5 5 1 ...
## $ LandSlope     : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood  : Factor w/ 25 levels "Blmngtn","Blueste",...: 13 13 9 9 22 9 9 9 13 ...
## $ Condition1    : Factor w/ 9 levels "Artery","Feedr",...: 2 3 3 3 3 3 3 3 3 ...
## $ Condition2    : Factor w/ 5 levels "Artery","Feedr",...: 3 3 3 3 3 3 3 3 3 ...
## $ BldgType      : Factor w/ 5 levels "1fam","2fmCon",...: 1 1 1 1 5 1 1 1 1 1 ...
## $ HouseStyle    : Factor w/ 7 levels "1.5Fin","1.5Unf",...: 3 3 5 5 3 5 3 5 3 3 ...
## $ OverallQual   : Factor w/ 10 levels "1","2","3","4",...: 5 6 5 6 8 6 6 6 7 4 ...
```

```

## $ OverallCond : Factor w/ 9 levels "1","2","3","4",...: 6 6 5 6 5 5 7 5 5 5 ...
## $ RoofStyle   : Factor w/ 6 levels "Flat","Gable",...: 2 4 2 2 2 2 2 2 2 2 ...
## $ RoofMatl    : Factor w/ 4 levels "CompShg","Tar&Grv",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Exterior1st : Factor w/ 14 levels "AsbShng","AsphShn",...: 12 13 12 12 7 7 7 12 7 10 ...
## $ Exterior2nd : Factor w/ 16 levels "AsbShng","AsphShn",...: 14 15 14 14 7 7 7 14 7 11 ...
## $ MasVnrType  : Factor w/ 4 levels "BrkCmn","BrkFace",...: 3 2 3 2 3 3 3 3 3 3 ...
## $ ExterQual   : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 4 3 4 4 4 4 4 ...
## $ ExterCond   : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 3 5 5 5 ...
## $ Foundation  : Factor w/ 6 levels "BrkTil","CBlock",...: 2 2 3 3 3 3 3 3 3 2 ...
## $ BsmtQual    : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 5 3 3 3 3 3 5 ...
## $ BsmtCond    : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ BsmtExposure : Factor w/ 5 levels "Av","Gd","Mn",...: 4 4 4 4 4 4 4 4 2 4 ...
## $ BsmtFinType1 : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 6 1 3 3 1 7 1 7 3 1 ...
## $ BsmtFinType2 : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 4 7 7 7 7 7 7 7 7 6 ...
## $ Heating     : Factor w/ 4 levels "GasA","GasW",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ HeatingQC   : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 1 1 3 1 3 3 5 ...
## $ CentralAir  : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
## $ Electrical  : Factor w/ 4 levels "FuseA","FuseF",...: 4 4 4 4 4 4 4 4 4 4 ...
## $ KitchenQual : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 3 5 3 3 5 5 5 3 5 ...
## $ Functional  : Factor w/ 8 levels "Maj1","Maj2",...: 8 8 8 8 8 8 8 8 8 8 ...
## $ FireplaceQu : Factor w/ 6 levels "Ex","Fa","Gd",...: 4 4 6 3 4 6 4 3 5 4 ...
## $ GarageType  : Factor w/ 7 levels "2Types","Attchd",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ GarageFinish : Factor w/ 4 levels "Fin","None","RFn",...: 4 4 1 1 3 1 1 1 4 1 ...
## $ GarageQual  : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ GarageCond  : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 6 6 ...
## $ PavedDrive  : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
## $ PoolQC      : Factor w/ 3 levels "Ex","Gd","None": 3 3 3 3 3 3 3 3 3 3 ...
## $ Fence       : Factor w/ 5 levels "GdPrv","GdWo",...: 3 5 3 5 5 5 1 5 5 3 ...
## $ MiscFeature  : Factor w/ 4 levels "Gar2","None",...: 2 1 2 2 2 2 4 2 2 2 ...
## $ SaleType     : Factor w/ 9 levels "COD","Con","ConLD",...: 9 9 9 9 9 9 9 9 9 9 ...
## $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",...: 5 5 5 5 5 5 5 5 5 5 ...

```

```
str(df_test)
```

```

## 'data.frame':    1459 obs. of  80 variables:
## $ Id           : int  1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 ...
## $ MSSubClass   : Factor w/ 16 levels "20","30","40",...: 1 1 6 6 12 6 1 6 1 1 ...
## $ MSZoning     : Factor w/ 6 levels "C (all)","FV",...: 4 5 5 5 5 5 5 5 5 5 ...
## $ LotFrontage  : num  80 81 74 78 43 ...
## $ LotArea      : int  11622 14267 13830 9978 5005 10000 7980 8402 10176 8400 ...
## $ Street       : Factor w/ 2 levels "Grv1","Pave": 2 2 2 2 2 2 2 2 2 2 ...
## $ Alley        : Factor w/ 3 levels "Grv1","None",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",...: 4 1 1 1 1 1 1 1 4 4 ...
## $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 2 4 4 4 4 4 ...
## $ Utilities    : Factor w/ 2 levels "AllPub","None": 1 1 1 1 1 1 1 1 1 1 ...
## $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",...: 5 1 5 5 5 1 5 5 5 1 ...
## $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",...: 13 13 9 9 22 9 9 9 9 13 ...
## $ Condition1   : Factor w/ 9 levels "Artery","Feedr",...: 2 3 3 3 3 3 3 3 3 3 ...
## $ Condition2   : Factor w/ 5 levels "Artery","Feedr",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ BldgType     : Factor w/ 5 levels "1fam","2fmCon",...: 1 1 1 1 5 1 1 1 1 1 ...
## $ HouseStyle   : Factor w/ 7 levels "1.5Fin","1.5Unf",...: 3 3 5 5 3 5 3 5 3 3 ...
## $ OverallQual  : Factor w/ 10 levels "1","2","3","4",...: 5 6 5 6 8 6 6 6 7 4 ...
## $ OverallCond  : Factor w/ 9 levels "1","2","3","4",...: 6 6 5 6 5 5 7 5 5 5 ...

```

```

## $ YearBuilt      : int  1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...
## $ YearRemodAdd   : int  1961 1958 1998 1998 1992 1994 2007 1998 1990 1970 ...
## $ RoofStyle      : Factor w/ 6 levels "Flat","Gable",...: 2 4 2 2 2 2 2 2 2 ...
## $ RoofMatl       : Factor w/ 4 levels "CompShg","Tar&Grv",...: 1 1 1 1 1 1 1 1 1 ...
## $ Exterior1st    : Factor w/ 14 levels "AsbShng","AsphShn",...: 12 13 12 12 7 7 7 12 7 10 ...
## $ Exterior2nd    : Factor w/ 16 levels "AsbShng","AsphShn",...: 14 15 14 14 7 7 7 14 7 11 ...
## $ MasVnrType     : Factor w/ 4 levels "BrkCmn","BrkFace",...: 3 2 3 2 3 3 3 3 3 ...
## $ MasVnrArea     : num  0 108 0 20 0 0 0 0 0 ...
## $ ExterQual      : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 4 3 4 4 4 4 ...
## $ ExterCond      : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 3 5 5 ...
## $ Foundation     : Factor w/ 6 levels "BrkTil","CBlock",...: 2 2 3 3 3 3 3 3 2 ...
## $ BsmtQual       : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 5 3 3 3 3 5 ...
## $ BsmtCond       : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 5 5 5 5 5 5 ...
## $ BsmtExposure   : Factor w/ 5 levels "Av","Gd","Mn",...: 4 4 4 4 4 4 4 2 4 ...
## $ BsmtFinType1   : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 6 1 3 3 1 7 1 7 3 1 ...
## $ BsmtFinSF1     : num  468 923 791 602 263 0 935 0 637 804 ...
## $ BsmtFinType2   : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 4 7 7 7 7 7 7 7 6 ...
## $ BsmtFinSF2     : num  144 0 0 0 0 0 0 0 0 78 ...
## $ BsmtUnfSF      : num  270 406 137 324 1017 ...
## $ TotalBsmtSF    : num  882 1329 928 926 1280 ...
## $ Heating        : Factor w/ 4 levels "GasA","GasW",...: 1 1 1 1 1 1 1 1 1 ...
## $ HeatingQC      : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 1 1 3 1 3 3 5 ...
## $ CentralAir     : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 ...
## $ Electrical     : Factor w/ 4 levels "FuseA","FuseF",...: 4 4 4 4 4 4 4 4 4 ...
## $ X1stFlrSF      : int  896 1329 928 926 1280 763 1187 789 1341 882 ...
## $ X2ndFlrSF      : int  0 0 701 678 0 892 0 676 0 0 ...
## $ LowQualFinSF   : int  0 0 0 0 0 0 0 0 0 0 ...
## $ GrLivArea      : int  896 1329 1629 1604 1280 1655 1187 1465 1341 882 ...
## $ BsmtFullBath   : num  0 0 0 0 0 0 1 0 1 1 ...
## $ BsmtHalfBath   : num  0 0 0 0 0 0 0 0 0 0 ...
## $ FullBath       : int  1 1 2 2 2 2 2 2 1 1 ...
## $ HalfBath       : int  0 1 1 1 0 1 0 1 1 0 ...
## $ BedroomAbvGr   : int  2 3 3 3 2 3 3 3 2 2 ...
## $ KitchenAbvGr   : int  1 1 1 1 1 1 1 1 1 1 ...
## $ KitchenQual    : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 3 5 3 3 5 5 5 3 5 ...
## $ TotRmsAbvGrd   : int  5 6 6 7 5 7 6 7 5 4 ...
## $ Functional     : Factor w/ 8 levels "Maj1","Maj2",...: 8 8 8 8 8 8 8 8 8 ...
## $ Fireplaces     : int  0 0 1 1 0 1 0 1 1 0 ...
## $ FireplaceQu    : Factor w/ 6 levels "Ex","Fa","Gd",...: 4 4 6 3 4 6 4 3 5 4 ...
## $ GarageType     : Factor w/ 7 levels "2Types","Attchd",...: 2 2 2 2 2 2 2 2 2 ...
## $ GarageYrBlt    : num  1961 1958 1997 1998 1992 ...
## $ GarageFinish   : Factor w/ 4 levels "Fin","None","RFn",...: 4 4 1 1 3 1 1 1 4 1 ...
## $ GarageCars     : num  1 1 2 2 2 2 2 2 2 ...
## $ GarageArea     : num  730 312 482 470 506 440 420 393 506 525 ...
## $ GarageQual     : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 5 5 5 5 5 5 ...
## $ GarageCond     : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 6 ...
## $ PavedDrive     : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 ...
## $ WoodDeckSF     : int  140 393 212 360 0 157 483 0 192 240 ...
## $ OpenPorchSF    : int  0 36 34 36 82 84 21 75 0 0 ...
## $ EnclosedPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
## $ X3SsnPorch     : int  0 0 0 0 0 0 0 0 0 0 ...
## $ ScreenPorch    : int  120 0 0 0 144 0 0 0 0 0 ...
## $ PoolArea       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ PoolQC         : Factor w/ 3 levels "Ex","Gd","None": 3 3 3 3 3 3 3 3 3 ...

```

```
## $ Fence      : Factor w/ 5 levels "GdPrv","GdWo",...: 3 5 3 5 5 5 1 5 5 3 ...
## $ MiscFeature : Factor w/ 4 levels "Gar2","None",...: 2 1 2 2 2 2 4 2 2 2 ...
## $ MiscVal     : int  0 12500 0 0 0 0 500 0 0 0 ...
## $ MoSold      : int   6 6 3 6 1 4 3 5 2 4 ...
## $ YrSold      : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
## $ SaleType    : Factor w/ 9 levels "COD","Con","ConLD",...: 9 9 9 9 9 9 9 9 9 ...
## $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",...: 5 5 5 5 5 5 5 5 5 ...
```

```
####test: Removing Columns####
```

```
#Redundant means a value representing over 99% of the data in a column
```

```
#Utilities
```

```
prop.table(table(df_test$Utilities, useNA = "ifany"))#["1"]
```

```
##
##      AllPub      None
## 0.998629198 0.001370802
```

```
#PoolArea
```

```
prop.table(table(df_test$PoolArea, useNA = "ifany"))
```

```
##
##      0      144      228      368      444      561
## 0.995887594 0.000685401 0.000685401 0.000685401 0.000685401 0.000685401
##      800
## 0.000685401
```

```
df_test[,c("Utilities", "PoolArea", "GrLivArea", "TotalBsmtSF")] <- list(NULL)
```

```
####Row-bind the train dataset with the test dataset####
```

```
#Make new isTrain column with 0/1
```

```
A2Data$isTrain = TRUE
df_test$isTrain = FALSE
```

```
dim(A2Data)
```

```
## [1] 1458 78
```

```
dim(df_test)
```

```
## [1] 1459 77
```

```
#Fill in all NAs for SalePrice in df_test
```

```
df_test$SalePrice = rep(NA,1459)
str(df_test$SalePrice)
```

```
## logi [1:1459] NA NA NA NA NA NA ...
```

```
# Combining the train and test set
A2Data_final = rbind(A2Data,df_test)
```

```
#df_test$MSZoning <- factor(df_test$MSZoning, levels = levels(A2Data$MSZoning))
```

```
#table(A2Data$MSZoning, useNA = "ifany")
#table(df_test$MSZoning, useNA = "ifany")
```

```
A2Data = A2Data_final[A2Data_final$isTrain == TRUE,]
df_test = A2Data_final[A2Data_final$isTrain == FALSE,]
```

```
dim(A2Data)
```

```
## [1] 1458 78
```

```
dim(df_test)
```

```
## [1] 1459 78
```

```
str(df_test)
```

```
## 'data.frame': 1459 obs. of 78 variables:
## $ Id : int 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 ...
## $ MSSubClass : Factor w/ 16 levels "20","30","40",...: 1 1 6 6 12 6 1 6 1 1 ...
## $ MSZoning : Factor w/ 6 levels "C (all)","FV",...: 3 4 4 4 4 4 4 4 4 4 ...
## $ LotFrontage : num 80 81 74 78 43 ...
## $ LotArea : int 11622 14267 13830 9978 5005 10000 7980 8402 10176 8400 ...
## $ Street : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
## $ Alley : Factor w/ 3 levels "Grvl","None",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ LotShape : Factor w/ 4 levels "IR1","IR2","IR3",...: 4 1 1 1 1 1 1 1 4 4 ...
## $ LandContour : Factor w/ 4 levels "Bnk","HLS","Low",...: 4 4 4 4 2 4 4 4 4 4 ...
## $ LotConfig : Factor w/ 5 levels "Corner","CulDSac",...: 5 1 5 5 5 1 5 5 5 1 ...
## $ LandSlope : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
## $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",...: 13 13 9 9 22 9 9 9 13 ...
## $ Condition1 : Factor w/ 9 levels "Artery","Feedr",...: 2 3 3 3 3 3 3 3 3 ...
## $ Condition2 : Factor w/ 8 levels "Artery","Feedr",...: 3 3 3 3 3 3 3 3 ...
## $ BldgType : Factor w/ 5 levels "1Fam","2fmCon",...: 1 1 1 1 5 1 1 1 1 1 ...
## $ HouseStyle : Factor w/ 8 levels "1.5Fin","1.5Unf",...: 3 3 6 6 3 6 3 6 3 3 ...
## $ OverallQual : Factor w/ 10 levels "1","2","3","4",...: 5 6 5 6 8 6 6 6 7 4 ...
## $ OverallCond : Factor w/ 10 levels "1","2","3","4",...: 6 6 5 6 5 5 7 5 5 5 ...
## $ YearBuilt : int 1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...
## $ YearRemodAdd : int 1961 1958 1998 1998 1992 1994 2007 1998 1990 1970 ...
## $ RoofStyle : Factor w/ 6 levels "Flat","Gable",...: 2 4 2 2 2 2 2 2 2 2 ...
## $ RoofMatl : Factor w/ 8 levels "ClyTile","CompShg",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ Exterior1st : Factor w/ 16 levels "AsbShng","AsphShn",...: 13 14 13 13 7 7 7 13 7 10 ...
## $ Exterior2nd : Factor w/ 17 levels "AsbShng","AsphShn",...: 14 15 14 14 7 7 7 14 7 11 ...
## $ MasVnrType : Factor w/ 4 levels "BrkCmn","BrkFace",...: 3 2 3 2 3 3 3 3 3 3 ...
## $ MasVnrArea : num 0 108 0 20 0 0 0 0 0 0 ...
## $ ExterQual : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 4 3 4 4 4 4 4 ...
## $ ExterCond : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 5 5 5 5 3 5 5 5 ...
## $ Foundation : Factor w/ 6 levels "BrkTil","CBBlock",...: 2 2 3 3 3 3 3 3 3 2 ...
## $ BsmtQual : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 5 3 3 3 3 3 5 ...
```

```

## $ BsmtCond      : Factor w/ 5 levels "Fa","Gd","None",...: 5 5 5 5 5 5 5 5 5 5 ...
## $ BsmtExposure  : Factor w/ 5 levels "Av","Gd","Mn",...: 4 4 4 4 4 4 4 4 2 4 ...
## $ BsmtFinType1  : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 6 1 3 3 1 7 1 7 3 1 ...
## $ BsmtFinSF1    : num  468 923 791 602 263 0 935 0 637 804 ...
## $ BsmtFinType2  : Factor w/ 7 levels "ALQ","BLQ","GLQ",...: 4 7 7 7 7 7 7 7 6 ...
## $ BsmtFinSF2    : num  144 0 0 0 0 0 0 0 78 ...
## $ BsmtUnfSF     : num  270 406 137 324 1017 ...
## $ Heating       : Factor w/ 6 levels "Floor","GasA",...: 2 2 2 2 2 2 2 2 2 ...
## $ HeatingQC     : Factor w/ 5 levels "Ex","Fa","Gd",...: 5 5 3 1 1 3 1 3 3 5 ...
## $ CentralAir    : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 ...
## $ Electrical    : Factor w/ 6 levels "FuseA","FuseF",...: 6 6 6 6 6 6 6 6 6 ...
## $ X1stFlrSF     : int   896 1329 928 926 1280 763 1187 789 1341 882 ...
## $ X2ndFlrSF     : int    0 0 701 678 0 892 0 676 0 0 ...
## $ LowQualFinSF  : int    0 0 0 0 0 0 0 0 0 0 ...
## $ BsmtFullBath  : num    0 0 0 0 0 0 1 0 1 1 ...
## $ BsmtHalfBath  : num    0 0 0 0 0 0 0 0 0 0 ...
## $ FullBath      : int    1 1 2 2 2 2 2 2 1 1 ...
## $ HalfBath      : int    0 1 1 1 0 1 0 1 1 0 ...
## $ BedroomAbvGr : int    2 3 3 3 2 3 3 3 2 2 ...
## $ KitchenAbvGr  : int    1 1 1 1 1 1 1 1 1 1 ...
## $ KitchenQual   : Factor w/ 5 levels "Ex","Fa","Gd",...: 4 3 4 3 3 4 4 4 3 4 ...
## $ TotRmsAbvGrd : int    5 6 6 7 5 7 6 7 5 4 ...
## $ Functional    : Factor w/ 8 levels "Maj1","Maj2",...: 7 7 7 7 7 7 7 7 7 ...
## $ Fireplaces    : int    0 0 1 1 0 1 0 1 1 0 ...
## $ FireplaceQu   : Factor w/ 6 levels "Ex","Fa","Gd",...: 4 4 6 3 4 6 4 3 5 4 ...
## $ GarageType    : Factor w/ 7 levels "2Types","Attchd",...: 2 2 2 2 2 2 2 2 2 ...
## $ GarageYrBlt   : num   1961 1958 1997 1998 1992 ...
## $ GarageFinish  : Factor w/ 4 levels "Fin","None","RFn",...: 4 4 1 1 3 1 1 1 4 1 ...
## $ GarageCars    : num    1 1 2 2 2 2 2 2 2 ...
## $ GarageArea    : num   730 312 482 470 506 440 420 393 506 525 ...
## $ GarageQual    : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 6 ...
## $ GarageCond    : Factor w/ 6 levels "Ex","Fa","Gd",...: 6 6 6 6 6 6 6 6 6 ...
## $ PavedDrive    : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 ...
## $ WoodDeckSF    : int   140 393 212 360 0 157 483 0 192 240 ...
## $ OpenPorchSF   : int    0 36 34 36 82 84 21 75 0 0 ...
## $ EnclosedPorch : int    0 0 0 0 0 0 0 0 0 0 ...
## $ X3SsnPorch    : int    0 0 0 0 0 0 0 0 0 0 ...
## $ ScreenPorch   : int   120 0 0 0 144 0 0 0 0 0 ...
## $ PoolQC        : Factor w/ 4 levels "Ex","Fa","Gd",...: 4 4 4 4 4 4 4 4 4 ...
## $ Fence         : Factor w/ 5 levels "GdPrv","GdWo",...: 3 5 3 5 5 5 1 5 5 3 ...
## $ MiscFeature    : Factor w/ 5 levels "Gar2","None",...: 2 1 2 2 2 2 4 2 2 2 ...
## $ MiscVal       : int    0 12500 0 0 0 0 500 0 0 0 ...
## $ MoSold        : int    6 6 3 6 1 4 3 5 2 4 ...
## $ YrSold        : int   2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
## $ SaleType      : Factor w/ 9 levels "COD","Con","ConLD",...: 9 9 9 9 9 9 9 9 9 ...
## $ SaleCondition : Factor w/ 6 levels "Abnorml","AdjLand",...: 5 5 5 5 5 5 5 5 5 ...
## $ SalePrice     : int   NA NA NA NA NA NA NA NA NA NA ...
## $ isTrain       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...

```

```

# #MSZoning
# df_test<-A2Data[!(A2Data$MSZoning == "None")]
# prop.table(table(df_test$MSZoning, useNA = "ifany"))
#
# #MSSubClass

```



```
# df_test<-A2Data[!(A2Data$MSSubClass == 150)]
# prop.table(table(df_test$MSSubClass, useNA = "ifany"))
#
# #Functional
# df_test<-A2Data[!(A2Data$Functional == "None")]
# prop.table(table(df_test$Functional, useNA = "ifany"))

####Training a 70% partition with the train dataset for the test dataset####
set.seed(760397)
index = createDataPartition(y = A2Data$SalePrice, p = 0.7, list = FALSE)
train = A2Data[index, ]
test = A2Data[-index, ]
```

## Forward Selection

```
#forward.vars = stepAIC(model1, direction = "backward", trace = FALSE)
#forward.vars$anova

model_forward = lm(log(SalePrice) ~ MSSubClass + MSZoning + LotFrontage + LotArea +
  Street + LotConfig + LandSlope + Neighborhood + Condition1 +
  Condition2 + OverallQual + OverallCond + YearBuilt + YearRemodAdd +
  RoofStyle + RoofMatl + Exterior1st + MasVnrType + ExterCond +
  Foundation + BsmtExposure + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +
  Heating + HeatingQC + CentralAir + X1stFlrSF + X2ndFlrSF +
  BsmtFullBath + FullBath + HalfBath + KitchenAbvGr + KitchenQual +
  Functional + Fireplaces + GarageCars + GarageArea + GarageQual +
  GarageCond + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch +
  ScreenPorch + PoolQC + Fence + SaleCondition,
  data = A2Data)
paste(summary(model_forward)$r.squared, " | ", summary(model_forward)$adj.r.squared)

## [1] "0.94856656184831 | 0.941086069664299"

#           R-Squared = 0.948000722728401
# Adjusted R-Squared = 0.940717568869546
press(model_forward)
```

```
## [1] Inf
```

```
#####

# model_forward = lm(log(SalePrice)~OverallQual+Neighborhood+TotRmsAbvGrd+
#           GarageArea+BsmFinType1+X1stFlrSF+X2ndFlrSF+RoofMatl+
#           OverallCond+YearBuilt+SaleCondition+BsmFinSF1+BldgType+
#           LotArea+Functional+ScreenPorch+CentralAir+Condition1+
#           KitchenQual+BsmExposure+HeatingQC+Heating+LandSlope+
#           Fireplaces+BsmFullBath+Street+WoodDeckSF+Foundation+
#           LotConfig+GarageCars+PoolQC+HalfBath+FullBath+KitchenAbvGr+
#           BsmUnfSF+BsmFinSF2+LotFrontage+YearRemodAdd+GarageQual+
```

```

#           Electrical+EnclosedPorch+SaleType+OpenPorchSF+X3SsnPorch+
#           GarageCond,
#           data = A2Data)
# summary(model_forward)
# paste(summary(model_forward)$r.squared, " | ", summary(model_forward)$adj.r.squared)
# #           R-Squared = 0.948000722728401
# #Adjusted R-Squared = 0.940717568869546
# press(model_forward)

#predict(model_forward, newdata = df_test)

```

```

#Doing the prediction on partitioned test set
prediction = predict(model_forward, newdata = df_test)

```

```

#Performing inverse log transform
value = exp(prediction)

```

```

# Checking the RMSE of the model
model_RMSE = rmse(df_test$SalePrice,value)
model_RMSE

```

```

str(df_test$Id)
str(df_test$SalePrice)

```

```

# Comparing the model predicted values with observed values
table = data.frame(Id = df_test$Id, SalePrice = value)
dim(table)

```

```

write.csv(table,file="C:/Users/dnguy/Desktop/3 Statistical Foundations/Unit 14 & 15 Project/kaggle_forw

```

## Creating the model using backward variable selection

Multiple R-Squared=0.9188 Adjusted R-Squared=0.9162 Kaggle RMSE=0.13700 CV Press=19.45764

```

# Applying backward variable selection method
backward.var=stepAIC(fit.analysis2.model,direction = "backward")
backward.var$anova

# Generating the model with the variables generated by backward variable selection method
back.model=lm(log(SalePrice) ~ MSZoning + log(LotArea) + Street + Alley + LotShape +
              LandContour + Condition2 + BldgType + OverallQual + OverallCond +
              YearBuilt + YearRemodAdd + Exterior1st + Exterior2nd + MasVnrType +
              ExterQual + ExterCond + Foundation + BsmtQual + BsmtExposure +
              log(BsmtFinSF1) + log(BsmtFinSF2) + log(BsmtUnfSF) + HeatingQC + CentralAir +
              X1stFlrSF + X2ndFlrSF + LowQualFinSF + BsmtFullBath + FullBath +
              HalfBath + KitchenAbvGr + KitchenQual + Functional + log(Fireplaces) +
              FireplaceQu + GarageCars + GarageArea + GarageCond + PavedDrive +
              log(WoodDeckSF) + log(EnclosedPorch) + ScreenPorch + MiscFeature +
              YrSold + SaleCondition,data=fit.analysis2)
summary(back.model)

# Calculating the CV Press of the backward linear regression model

```

```

ols_press(backward.model)

#Doing the prediction on partitioned test set
prediction=predict(back.model,newdata=test.data)
prediction

#Performing inverse log transform
value=2.718^prediction
value

# Checking the RMSE of the model
rmse.model=rmse(test.data$SalePrice,value)
rmse.model

# Comparing the model predicted values with observed values
table=data.frame(Id=test.data$Id,ObsSalePrice=test.data$SalePrice,PredSalePrice=value)
table

# Predictions on the Original Test Set
predictiontest=predict(back.model,newdata=test)
predictiontest

#Performing inverse log transform
pred_value=2.718^predictiontest
pred_value

# Putting the predicted values in a dataframe
output.df_train=data.frame(Id=test$Id, SalePrice=pred_value)
head(output.df_train)
dim(output.df_train)
table(is.na(output.df_train))

# Putting the dataframe in a csv to submit on the kaggle to check the Score
write.csv(output.df_train,file="C:/Users/ARTH PATEL/Desktop/MSDS@SMU/6371-LSA/Stats Project/kaggle_subm

```

## Creating the model using stepwise variable selection

Multiple R-Squared=0.9188 Adjusted R-Squared=0.9162 Kaggle RMSE=0.13926 CV Press=19.0847

```

# Applying stepwise variable selection method
stepwise.var=stepAIC(fit.analysis2.model,direction = "both")
stepwise.var$anova

# Generating the model with the variables generated by stepwise variable selection method
step.model=lm(log(SalePrice) ~ MSZoning + log(LotArea) + Street + Alley + LotShape +
  LandContour + Condition2 + BldgType + OverallQual + OverallCond +
  YearBuilt + YearRemodAdd + Exterior1st + Exterior2nd + MasVnrType +
  ExterQual + ExterCond + Foundation + BsmtQual + BsmtExposure +
  log(BsmtFinSF1) + log(BsmtFinSF2) + log(BsmtUnfSF) + HeatingQC + CentralAir +
  X1stFlrSF + X2ndFlrSF + LowQualFinSF + BsmtFullBath + FullBath +
  HalfBath + KitchenAbvGr + KitchenQual + Functional + log(Fireplaces) +
  FireplaceQu + GarageCars + GarageArea + GarageCond + PavedDrive +

```

```

        log(WoodDeckSF) + log(EnclosedPorch) + ScreenPorch + MiscFeature +
        YrSold + SaleCondition,data=fit.analysis2)
summary(step.model)

# Calculating the CV Press of the backward linear regression model
ols_press(stepwise.model)

#Doing the prediction on partitioned test set
prediction=predict(step.model,newdata=test.data)
prediction

#Performing inverse log transform
value=2.718^prediction
value

# Checking the RMSE of the model
rmse.model=rmse(test.data$SalePrice,value)
rmse.model

# Comparing the model predicted values with observed values
table=data.frame(Id=test.data$Id,ObsSalePrice=test.data$SalePrice,PredSalePrice=value)
table

# Predictions on the Original Test Set
predictiontest=predict(step.model,newdata=test)
predictiontest

#Performing inverse log transform
pred_value=2.718^predictiontest
pred_value

# Putting the predicted values in a dataframe
output.df_train=data.frame(Id=test$Id, SalePrice=pred_value)
head(output.df_train)
dim(output.df_train)
table(is.na(output.df_train))

# Putting the dataframe in a csv to submit on the kaggle to check the Score
write.csv(output.df_train,file="C:/Users/ARTH PATEL/Desktop/MSDS@SMU/6371-LSA/Stats Project/kaggle_subm

```

## Creating the model using custom variable selection

Multiple R-Squared=0.8596 Adjusted R-Squared=0.8568 Kaggle RMSE=0.13926 CV Press=34.379

```

# Generating the custom model
custom.model=lm(log(SalePrice) ~ Neighborhood+GarageCars+SaleCondition+RoofStyle+CentralAir+Fireplaces+
        X1stFlrSF*X2ndFlrSF+ScreenPorch+BsmFinSF1*BsmFinSF2+KitchenQual+BsmFinFullBath*BsmFinHalf
        YearBuilt+PoolQC+HouseStyle+LotArea+BsmFinType2*BsmFinType1+BsmFinType1+Electrical
        Electrical*CentralAir+GarageFinish+GarageYrBlt+GarageType,data=fit.analysis2)
summary(custom.model)

# Calculating the CV Press of the custom linear regression model

```

```

ols_press(custom.model)

#Doing the prediction on partitioned test set
prediction=predict(custom.model,newdata=test.data)
prediction

#Performing inverse log transform
value=2.718^prediction
value

# Checking the RMSE of the model
rmse.model=rmse(test.data$SalePrice,value)
rmse.model

# Comparing the model predicted values with observed values
table=data.frame(Id=test.data$Id,ObsSalePrice=test.data$SalePrice,PredSalePrice=value)
table

# Predictions on the Original Test Set
predictiontest=predict(custom.model,newdata=test)
predictiontest

#Performing inverse log transform
pred_value=2.718^predictiontest
pred_value

# Putting the predicted values in a dataframe
output.df_train=data.frame(Id=test$Id, SalePrice=pred_value)
head(output.df_train)
dim(output.df_train)
table(is.na(output.df_train))

# Putting the dataframe in a csv to submit on the kaggle to check the Score
write.csv(output.df_train, file="C:/Users/dnguy/Desktop/3 Statistical Foundations/Unit 14 & 15 Project/

####Analysis 2 EDA####
plot_histogram(A2Data[,2:79])

```