Assignment 4 Data Wrangling

Nicholas Jacob and Zayne Mclaughlin

2024-09-05

1. Loading the housing data and calculating the age, ageSinceRemodel, and ageofGarage columns.

```r
housingData = read.csv("housingData-1.csv")

housingData <- housingData %>%
  dplyr::mutate(age = YrSold - YearBuilt, ageSinceRemodel = YrSold - YearRemodAdd, ageofGarage = YrSold
```

1.b.

Selecting only the numeric variables from the housing dataset

```r
housingNumeric <- housingData %>%
  dplyr::select(where(is.numeric))
```

1.c.

Selecting only the character variables from the housing dataset and converting them to factors.

```r
housingFactor <- housingData %>%
  dplyr::select(where(is.character))%>%
  mutate_all(factor)
```

1.d.

Using glimpse to check the structure of the new housingFactor and housingNumeric tibbles.

```r
glimpse(housingFactor)
```

```
## Rows: 1,000
## Columns: 38
## $ MSZoning     <fct> RL, RL, RL, RL, RL, RL, RL, RM, RL, RL, RL, RL, RL, RL, R~
## $ Alley        <fct> NA, NA, NA, NA, NA, NA, NA, Pave, NA, NA, NA, NA, NA, NA,~
## $ LotShape     <fct> IR1, IR1, IR1, IR1, Reg, Reg, Reg, IR1, Reg, IR1, IR1, Re~
## $ LandContour  <fct> Lvl, Low, Lvl, Bnk, Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, Lvl, Lv~
## $ LotConfig    <fct> CulDSac, Inside, other, Inside, Inside, Inside, Corner, C~
## $ LandSlope    <fct> Gtl, Mod, Gtl, Gtl, Gtl, Gtl, Gtl, Gtl, Mod, Gtl, Gtl, Gt~
## $ Neighborhood <fct> NAmes, ClearCr, Sawyer, Crawfor, NAmes, CollgCr, Sawyer, ~
## $ Condition1   <fct> Norm, Norm, Feedr, Norm, Norm, Norm, Norm, Norm, Norm, No~
## $ BldgType     <fct> 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fa~
## $ HouseStyle   <fct> 1Story, 1Story, 1Story, 2Story, 1Story, 2Story, 1Story, 2~
## $ RoofStyle    <fct> Gable, Gable, Gable, Gable, Gable, Gable, Hip, other, Gab~
## $ Exterior1st  <fct> Plywood, Wd Sdng, VinylSd, Wd Sdng, HdBoard, VinylSd, HdB~
## $ Exterior2nd  <fct> Plywood, Wd Sdng, VinylSd, Wd Sdng, HdBoard, VinylSd, HdB~
## $ MasVnrType   <fct> BrkFace, BrkCmn, None, None, None, BrkFace, None, None, B~
## $ ExterQual    <fct> Avg, Avg, Avg, Avg, Avg, AboveAvg, Avg, AboveAvg, Avg, Av~
## $ ExterCond    <fct> Avg, AboveAvg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Av~
## $ Foundation   <fct> CBlock, CBlock, CBlock, CBlock, CBlock, PConc, CBlock, ot~
## $ BsmtQual     <fct> Avg, Avg, Avg, Avg, Avg, AboveAvg, Avg, Avg, Avg, AboveAv~
## $ BsmtCond     <fct> Avg, Avg, Avg, Avg, Avg, Avg, Avg, BelowAvg, Avg, Avg, Av~
## $ BsmtExposure <fct> Mn, Av, No, No, No, No, No, No, Gd, No, No, Av, No, Gd, N~
## $ BsmtFinType1 <fct> BLQ, Rec, BLQ, BLQ, Rec, Unf, GLQ, Unf, GLQ, Unf, BLQ, GL~
## $ BsmtFinType2 <fct> Rec, Unf, Unf, Unf, Unf, Unf, Unf, Unf, LwQ, Unf, Unf, Un~
## $ Heating      <fct> GasA, GasA, GasA, GasA, GasA, GasA, GasA, other, GasA, Ga~
## $ HeatingQC    <fct> AboveAvg, BelowAvg, AboveAvg, BelowAvg, Avg, AboveAvg, Av~
## $ CentralAir   <fct> Y, Y, Y, Y, Y, Y, Y, N, Y, Y, Y, Y, Y, Y, N, N, Y, Y, Y, ~
## $ Electrical   <fct> SBrkr, SBrkr, SBrkr, SBrkr, SBrkr, SBrkr, SBrkr, SBrkr, S~
## $ KitchenQual  <fct> Avg, Avg, AboveAvg, AboveAvg, Avg, AboveAvg, Avg, AboveAv~
## $ Functional   <fct> Typ, Typ, Typ, Typ, Typ, Typ, Typ, Typ, Typ, Typ, Typ, Ty~
## $ FireplaceQu  <fct> BelowAvg, NA, NA, AboveAvg, AboveAvg, NA, NA, AboveAvg, N~
## $ GarageType   <fct> Attchd, Attchd, Attchd, Attchd, Attchd, Attchd, Detchd, D~
## $ GarageFinish <fct> RFn, Unf, RFn, Unf, RFn, RFn, Unf, Unf, RFn, Fin, RFn, Fi~
## $ GarageQual   <fct> Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Av~
## $ GarageCond   <fct> Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Avg, Av~
## $ PavedDrive   <fct> Y, N, Y, P, Y, Y, Y, N, Y, Y, Y, Y, Y, Y, Y, N, Y, N, Y, ~
## $ PoolQC       <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ Fence        <fct> MnPrv, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, MnPrv,~
## $ MiscFeature  <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ SaleType     <fct> WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, W~
```

```
glimpse(housingNumeric)
```

```
## Rows: 1,000
## Columns: 39
## $ Id              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
## $ MSSubClass      <int> 20, 20, 20, 70, 20, 60, 20, 70, 60, 60, 20, 120, 60, 2~
## $ LotFrontage     <int> NA, NA, 57, NA, 80, 72, 80, 65, 80, 93, 100, 43, 75, 8~
## $ LotArea         <int> 11000, 36500, 9764, 7500, 9200, 11317, 8480, 11700, 97~
## $ OverallQual     <int> 5, 5, 5, 6, 6, 7, 5, 7, 6, 6, 6, 7, 6, 6, 6, 4, 5, 6, ~
## $ OverallCond     <int> 6, 5, 7, 7, 6, 5, 6, 7, 6, 5, 5, 5, 6, 8, 4, 2, 5, 7, ~
## $ YearBuilt       <int> 1966, 1964, 1967, 1942, 1965, 2003, 1963, 1880, 1964, ~
## $ YearRemodAdd    <int> 1966, 1964, 2003, 1950, 1965, 2003, 1963, 2003, 1964, ~
## $ MasVnrArea      <int> 200, 621, 0, 0, 0, 101, 0, 0, 360, 318, 272, 16, 140, ~
## $ BsmtFinSF1      <int> 740, 812, 702, 547, 892, 0, 630, 0, 674, 0, 490, 16, 5~
## $ BsmtFinSF2      <int> 230, 0, 0, 0, 0, 0, 0, 0, 106, 0, 0, 0, 0, 0, 0, 0, 12~
## $ BsmtUnfSF       <int> 184, 812, 192, 224, 244, 840, 340, 1240, 0, 936, 935, ~
## $ TotalBsmtSF     <int> 1154, 1624, 894, 771, 1136, 840, 970, 1240, 780, 936, ~
## $ X1stFlrSF       <int> 1154, 1582, 894, 753, 1136, 840, 970, 1320, 798, 962, ~
## $ X2ndFlrSF       <int> 0, 0, 0, 741, 0, 828, 0, 1320, 813, 830, 0, 0, 728, 0,~
## $ LowQualFinSF    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ GrLivArea       <int> 1154, 1582, 894, 1494, 1136, 1668, 970, 2640, 1611, 17~
## $ BsmtFullBath    <int> 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, ~
## $ BsmtHalfBath    <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ~
## $ FullBath        <int> 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 1, ~
## $ HalfBath        <int> 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, ~
## $ BedroomAbvGr    <int> 3, 4, 3, 3, 3, 3, 2, 4, 4, 3, 3, 2, 3, 3, 4, 4, 2, 2, ~
## $ KitchenAbvGr    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, ~
## $ TotRmsAbvGrd    <int> 6, 7, 5, 7, 5, 8, 5, 8, 7, 8, 7, 7, 6, 6, 6, 8, 6, 5, ~
## $ Fireplaces      <int> 1, 0, 0, 2, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, ~
## $ GarageYrBlt     <int> 1966, 1964, 1967, 1942, 1965, 2003, 1996, 1950, 1964, ~
## $ GarageCars      <int> 2, 2, 2, 1, 1, 2, 2, 4, 2, 2, 2, 2, 2, 2, 1, 3, 2, 1, ~
## $ GarageArea      <int> 480, 390, 450, 213, 384, 500, 624, 864, 442, 451, 576,~
## $ WoodDeckSF      <int> 0, 168, 0, 0, 426, 144, 0, 181, 328, 0, 0, 143, 252, 2~
## $ OpenPorchSF     <int> 58, 198, 0, 0, 0, 68, 24, 0, 128, 0, 0, 20, 0, 0, 66, ~
## $ EncPorchSF      <int> 0, 0, 0, 224, 0, 0, 192, 386, 189, 0, 407, 0, 0, 0, 13~
## $ PoolArea        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ MiscVal         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ MoSold          <int> 11, 6, 5, 11, 7, 9, 7, 5, 6, 5, 7, 5, 7, 5, 5, 5, 4, 5~
## $ YrSold          <int> 2009, 2006, 2008, 2009, 2008, 2007, 2007, 2009, 2008, ~
## $ SalePrice       <int> 154000, 190000, 130000, 177500, 140000, 180000, 132500~
## $ age             <int> 43, 42, 41, 67, 43, 4, 44, 129, 44, 8, 44, 4, 32, 31, ~
## $ ageSinceRemodel <int> 43, 42, 5, 59, 43, 4, 44, 6, 44, 8, 44, 3, 32, 31, 60,~
## $ ageofGarage     <int> 43, 42, 41, 67, 43, 4, 11, 59, 44, 8, 44, 4, 32, 31, 9~
```

1.e.

The following functions create a Q_1 and Q_3 function that takes a vector x and an optional `na.rm` which by default has been set to true. Then it calls the `quantile` function and extracts the second and the fourth element from the five number summary giving just the Q_1 and Q_3.

```
Q1<-function(x,na.rm=TRUE) {
quantile(x,na.rm=na.rm)[2]
}
Q3<-function(x,na.rm=TRUE) {
quantile(x,na.rm=na.rm)[4]
}
```

1.f. Here is a function that will do the numeric computations for us.

```
myNumericSummary <- function(x){
  c(length(x), n_distinct(x), sum(is.na(x)), mean(x, na.rm=TRUE),
  min(x,na.rm=TRUE), Q1(x,na.rm=TRUE), median(x,na.rm=TRUE), Q3(x,na.rm=TRUE),
  max(x,na.rm=TRUE), sd(x,na.rm=TRUE))
}
```

1.g.

I think this is what was intended. I did get this to work with `summarise_all` as well without the additional commands of `across` and `where`

```
numericSummary <- housingNumeric %>%
  summarise_all(myNumericSummary) #Applying the custom myNumericSummary function to all variables in th

glimpse(numericSummary)
```

```
## Rows: 10
## Columns: 39
## $ Id              <dbl> 1000.0000, 1000.0000, 0.0000, 500.5000, 1.0000, 250.75~
## $ MSSubClass      <dbl> 1000.00000, 13.00000, 0.00000, 57.18500, 20.00000, 20.~
## $ LotFrontage     <dbl> 1000.00000, 102.00000, 207.00000, 68.74527, 21.00000, ~
## $ LotArea         <dbl> 1000.000, 760.000, 0.000, 10424.881, 1477.000, 7500.00~
## $ OverallQual     <dbl> 1000.000000, 10.000000, 0.000000, 5.979000, 1.000000, ~
## $ OverallCond     <dbl> 1000.000000, 8.000000, 0.000000, 5.638000, 2.000000, 5~
## $ YearBuilt       <dbl> 1000.00000, 108.00000, 0.00000, 1969.83600, 1875.00000~
## $ YearRemodAdd    <dbl> 1000.00000, 61.00000, 0.00000, 1984.10800, 1950.00000,~
## $ MasVnrArea      <dbl> 1000.00000, 249.00000, 4.00000, 95.41767, 0.00000, 0.0~
## $ BsmtFinSF1      <dbl> 1000.0000, 490.0000, 0.0000, 438.6860, 0.0000, 0.0000,~
## $ BsmtFinSF2      <dbl> 1000.000, 107.000, 0.000, 44.296, 0.000, 0.000, 0.000,~
## $ BsmtUnfSF       <dbl> 1000.0000, 598.0000, 0.0000, 535.0780, 0.0000, 208.000~
## $ TotalBsmtSF     <dbl> 1000.0000, 549.0000, 0.0000, 1018.0600, 0.0000, 793.00~
## $ X1stFlrSF       <dbl> 1000.0000, 581.0000, 0.0000, 1131.2510, 334.0000, 868.~
## $ X2ndFlrSF       <dbl> 1000.0000, 306.0000, 0.0000, 346.2790, 0.0000, 0.0000,~
## $ LowQualFinSF    <dbl> 1000.00000, 15.00000, 0.00000, 4.99100, 0.00000, 0.000~
## $ GrLivArea       <dbl> 1000.000, 664.000, 0.000, 1482.521, 334.000, 1110.750,~
## $ BsmtFullBath    <dbl> 1000.0000000, 3.0000000, 0.0000000, 0.4270000, 0.00000~
## $ BsmtHalfBath    <dbl> 1000.0000000, 2.0000000, 0.0000000, 0.0590000, 0.00000~
## $ FullBath        <dbl> 1000.0000000, 4.0000000, 0.0000000, 1.5290000, 0.00000~
## $ HalfBath        <dbl> 1000.0000000, 3.0000000, 0.0000000, 0.3840000, 0.00000~
## $ BedroomAbvGr    <dbl> 1000.0000000, 7.0000000, 0.0000000, 2.8650000, 0.00000~
## $ KitchenAbvGr    <dbl> 1000.0000000, 3.0000000, 0.0000000, 1.0410000, 1.00000~
## $ TotRmsAbvGrd    <dbl> 1000.000000, 11.000000, 0.000000, 6.410000, 2.000000, ~
## $ Fireplaces      <dbl> 1000.0000000, 4.0000000, 0.0000000, 0.6180000, 0.00000~
## $ GarageYrBlt     <dbl> 1000.00000, 94.00000, 53.00000, 1976.93770, 1906.00000~
## $ GarageCars      <dbl> 1000.0000000, 5.0000000, 0.0000000, 1.7200000, 0.00000~
## $ GarageArea      <dbl> 1000.0000, 353.0000, 0.0000, 458.3290, 0.0000, 318.750~
## $ WoodDeckSF      <dbl> 1000.0000, 226.0000, 0.0000, 94.5550, 0.0000, 0.0000, ~
## $ OpenPorchSF     <dbl> 1000.00000, 169.00000, 0.00000, 43.61000, 0.00000, 0.0~
## $ EncPorchSF      <dbl> 1000.0000, 122.0000, 0.0000, 40.6410, 0.0000, 0.0000, ~
## $ PoolArea        <dbl> 1000.00000, 3.00000, 0.00000, 1.22400, 0.00000, 0.0000~
## $ MiscVal         <dbl> 1000.0000, 14.0000, 0.0000, 27.2100, 0.0000, 0.0000, 0~
## $ MoSold          <dbl> 1000.000000, 12.000000, 0.000000, 6.207000, 1.000000, ~
## $ YrSold          <dbl> 1000.00000, 5.00000, 0.00000, 2007.91900, 2006.00000, ~
## $ SalePrice       <dbl> 1000.00, 477.00, 0.00, 174560.61, 39300.00, 130000.00,~
## $ age             <dbl> 1000.00000, 115.00000, 0.00000, 38.08300, 1.00000, 10.~
## $ ageSinceRemodel <dbl> 1000.00000, 61.00000, 0.00000, 23.81100, 0.00000, 6.00~
## $ ageofGarage     <dbl> 1000.00000, 97.00000, 53.00000, 30.97254, 0.00000, 9.0~
```

1.h.

Adding descriptive stat names as the first column to the numeric summary table.

```r
numericSummary <-cbind(stat=c("n","unique","missing","mean","min","Q1","median",
                              "Q3","max","sd"),numericSummary)
```

1.i.

```r
numericSummaryFinal <- numericSummary %>%
  tidyr::pivot_longer("Id":"ageofGarage", names_to = "variable", values_to = "value") %>%
  tidyr::pivot_wider(names_from = stat, values_from = value) %>%#Pivoting the numeric summary table to
  dplyr::mutate(missing_pct = 100*missing/n,
         unique_pct = 100*unique/n) %>%
  dplyr::select(variable, n, missing, missing_pct, unique, unique_pct, everything())
#Adding percentage calculations for missing and unique values.

options(digits=3)#limit the number of digits in the table
options(scipen=99)
numericSummaryFinal %>% kable()#display table nicely when knitted
```

variable	n	missing	missing_pct	unique	unique_pct	mean	min	Q1	median	Q3	max	sd
Id	1000	0	0.0	1000	100.0	500.500	1	251	500	750.2	1000	288.819
MSSubClass	1000	0	0.0	13	1.3	57.185	20	20	50	70.0	190	41.875
LotFrontage	1000	207	20.7	102	10.2	68.745	21	58	68	80.0	313	23.198
LotArea	1000	0	0.0	760	76.0	10424.881	1477	7500	9422	11423.5	215245	9940.619
OverallQual	1000	0	0.0	10	1.0	5.979	1	5	6	7.0	10	1.310
OverallCond	1000	0	0.0	8	0.8	5.638	2	5	5	6.0	9	1.114
YearBuilt	1000	0	0.0	108	10.8	1969.836	1875	1954	1971	1998.0	2009	29.119

variable	n	missing	missing_pct	unique	unique_pct	mean	min	Q1	median	Q3	max	sd
YearRemodAdd	1000	0	0.0	61	6.1	1984.108	1950	1967	1992	2002.0	2010	20.116
MasVnrArea	1000	4	0.4	249	24.9	95.418	0	0	0	146.2	1600	177.318
BsmtFinSF1	1000	0	0.0	490	49.0	438.686	0	0	400	700.0	1880	405.837
BsmtFinSF2	1000	0	0.0	107	10.7	44.296	0	0	0	0.0	1127	150.493
BsmtUnfSF	1000	0	0.0	598	59.8	535.078	0	208	441	779.2	2153	417.944
TotalBsmtSF	1000	0	0.0	549	54.9	1018.060	0	793	962	1223.5	3206	403.641
X1stFlrSF	1000	0	0.0	581	58.1	1131.251	334	868	1060	1327.2	3228	350.862
X2ndFlrSF	1000	0	0.0	306	30.6	346.279	0	0	0	735.0	1872	426.395
LowQualFinSF	1000	0	0.0	15	1.5	4.991	0	0	0	0.0	528	45.295
GrLivArea	1000	0	0.0	664	66.4	1482.521	334	1111	1442	1735.0	4316	490.566
BsmtFullBath	1000	0	0.0	3	0.3	0.427	0	0	0	1.0	2	0.509
BsmtHalfBath	1000	0	0.0	2	0.2	0.059	0	0	0	0.0	1	0.236
FullBath	1000	0	0.0	4	0.4	1.529	0	1	2	2.0	3	0.531
HalfBath	1000	0	0.0	3	0.3	0.384	0	0	0	1.0	2	0.501
BedroomAbvGr	1000	0	0.0	7	0.7	2.865	0	2	3	3.0	6	0.791
KitchenAbvGr	1000	0	0.0	3	0.3	1.041	1	1	1	1.0	3	0.203
TotRmsAbvGrd	1000	0	0.0	11	1.1	6.410	2	5	6	7.0	12	1.562
Fireplaces	1000	0	0.0	4	0.4	0.618	0	0	1	1.0	3	0.642
GarageYrBlt	1000	53	5.3	94	9.4	1976.938	1906	1960	1977	1999.0	2009	23.592
GarageCars	1000	0	0.0	5	0.5	1.720	0	1	2	2.0	4	0.714
GarageArea	1000	0	0.0	353	35.3	458.329	0	319	470	572.0	1356	197.780
WoodDeckSF	1000	0	0.0	226	22.6	94.555	0	0	0	168.0	857	127.144
OpenPorchSF	1000	0	0.0	169	16.9	43.610	0	0	22	64.0	547	61.915
EncPorchSF	1000	0	0.0	122	12.2	40.641	0	0	0	0.0	508	82.139
PoolArea	1000	0	0.0	3	0.3	1.224	0	0	0	0.0	648	27.403
MiscVal	1000	0	0.0	14	1.4	27.210	0	0	0	0.0	3500	190.707
MoSold	1000	0	0.0	12	1.2	6.207	1	4	6	8.0	12	2.626
YrSold	1000	0	0.0	5	0.5	2007.919	2006	2007	2008	2009.0	2010	1.318
SalePrice	1000	0	0.0	477	47.7	174560.607	39300	130000	160000	205000.0	755000	69329.319
age	1000	0	0.0	115	11.5	38.083	1	10	37	55.0	135	29.109
ageSinceRemodel	1000	0	0.0	61	6.1	23.811	0	6	16	41.2	60	20.033
ageofGarage	1000	53	5.3	97	9.7	30.973	0	9	30	48.0	102	23.563

1.j.

Helper functions to compute the modes of a vector, including the first, second, and least common modes, as well as their frequencies.

```r
getmodes <- function(v,type=1) {
tbl <- table(v)
m1<-which.max(tbl)
if (type==1) {
return (names(m1)) #1st mode
}
else if (type==2) {
return (names(which.max(tbl[-m1]))) #2nd mode
}
else if (type==-1) {
return (names(which.min(tbl))) #least common mode
}
else {
stop("Invalid type selected")
}
}
getmodesCnt <- function(v,type=1) {
tbl <- table(v)
m1<-which.max(tbl)
if (type==1) {
return (max(tbl)) #1st mode freq
}
else if (type==2) {
return (max(tbl[-m1])) #2nd mode freq
}
else if (type==-1) {
return (min(tbl)) #least common freq
}
else {
stop("Invalid type selected")
}
}
```

Next, I'll package all the functions (that don't rely on another computation) together

```
myCategoricalSummary <- function(x){
  c(length(x),sum(is.na(x)), n_distinct(x),
  getmodes(x,1),getmodesCnt(x,1),getmodes(x,2),getmodesCnt(x,2),getmodes(x,-1),getmodesCnt(x,-1))
}

myCategoricalSummary(housingData$MSZoning)
```

```
## [1] "1000" "0"    "4"    "RL"   "803"  "RM"   "151"  "RH"   "10"
```

Test looks good. Now we `summarize` all the data.

```
factorSummary <- housingFactor %>%
  summarise_all(myCategoricalSummary)

factorSummary<-cbind(stat=c("n","missing","unique",
                             "1st mode", "first_mode_freq","2nd mode",
                            "second_mode_freq","least common","least common freq"),
                     factorSummary) #add titles becarefule to omit spaces for ones we need again.
glimpse(factorSummary)
```

```
## Rows: 9
## Columns: 39
## $ stat         <chr> "n", "missing", "unique", "1st mode", "first_mode_freq", ~
## $ MSZoning     <chr> "1000", "0", "4", "RL", "803", "RM", "151", "RH", "10"
## $ Alley        <chr> "1000", "938", "3", "Grvl", "40", "Pave", "22", "Pave", "~
## $ LotShape     <chr> "1000", "0", "4", "Reg", "633", "IR1", "330", "IR3", "7"
## $ LandContour  <chr> "1000", "0", "4", "Lvl", "905", "Bnk", "40", "Low", "26"
## $ LotConfig    <chr> "1000", "0", "4", "Inside", "711", "Corner", "179", "othe~
## $ LandSlope    <chr> "1000", "0", "3", "Gtl", "946", "Mod", "48", "Sev", "6"
## $ Neighborhood <chr> "1000", "0", "18", "NAmes", "167", "CollgCr", "113", "Tim~
## $ Condition1   <chr> "1000", "0", "6", "Norm", "871", "Feedr", "51", "PosA", "~
## $ BldgType     <chr> "1000", "0", "5", "1Fam", "837", "TwnhsE", "81", "2fmCon"~
## $ HouseStyle   <chr> "1000", "0", "8", "1Story", "488", "2Story", "310", "2.5F~
## $ RoofStyle    <chr> "1000", "0", "3", "Gable", "795", "Hip", "184", "other", ~
## $ Exterior1st  <chr> "1000", "0", "8", "VinylSd", "328", "HdBoard", "175", "Ce~
## $ Exterior2nd  <chr> "1000", "0", "9", "VinylSd", "320", "HdBoard", "159", "Br~
## $ MasVnrType   <chr> "1000", "4", "5", "None", "617", "BrkFace", "313", "BrkCm~
## $ ExterQual    <chr> "1000", "0", "3", "Avg", "657", "AboveAvg", "336", "Below~
## $ ExterCond    <chr> "1000", "0", "3", "Avg", "880", "AboveAvg", "103", "Below~
## $ Foundation   <chr> "1000", "0", "4", "CBlock", "463", "PConc", "414", "other~
## $ BsmtQual     <chr> "1000", "31", "4", "AboveAvg", "488", "Avg", "459", "Belo~
## $ BsmtCond     <chr> "1000", "31", "4", "Avg", "903", "AboveAvg", "37", "Below~
## $ BsmtExposure <chr> "1000", "32", "5", "No", "668", "Av", "140", "Mn", "76"
## $ BsmtFinType1 <chr> "1000", "31", "7", "GLQ", "273", "Unf", "265", "LwQ", "52"
## $ BsmtFinType2 <chr> "1000", "32", "7", "Unf", "853", "Rec", "36", "ALQ", "11"
## $ Heating      <chr> "1000", "0", "2", "GasA", "974", "other", "26", "other", ~
## $ HeatingQC    <chr> "1000", "0", "3", "AboveAvg", "664", "Avg", "300", "Below~
## $ CentralAir   <chr> "1000", "0", "2", "Y", "936", "N", "64", "N", "64"
## $ Electrical   <chr> "1000", "1", "5", "SBrkr", "908", "FuseA", "72", "FuseP",~
## $ KitchenQual  <chr> "1000", "0", "3", "Avg", "534", "AboveAvg", "439", "Below~
## $ Functional   <chr> "1000", "0", "6", "Typ", "924", "Min2", "26", "Maj2", "4"
## $ FireplaceQu  <chr> "1000", "466", "4", "AboveAvg", "250", "Avg", "240", "Bel~
## $ GarageType   <chr> "1000", "53", "7", "Attchd", "601", "Detchd", "280", "2Ty~
## $ GarageFinish <chr> "1000", "53", "4", "Unf", "434", "RFn", "291", "Fin", "22~
## $ GarageQual   <chr> "1000", "53", "4", "Avg", "907", "BelowAvg", "33", "Above~
## $ GarageCond   <chr> "1000", "53", "4", "Avg", "910", "BelowAvg", "31", "Above~
## $ PavedDrive   <chr> "1000", "0", "3", "Y", "912", "N", "62", "P", "26"
## $ PoolQC       <chr> "1000", "998", "3", "Fa", "1", "Gd", "1", "Fa", "1"
## $ Fence        <chr> "1000", "805", "5", "MnPrv", "108", "GdPrv", "40", "MnWw"~
## $ MiscFeature  <chr> "1000", "966", "3", "Shed", "32", "Othr", "2", "Othr", "2"
## $ SaleType     <chr> "1000", "0", "2", "WD", "971", "other", "29", "other", "2~
```

```r
factorSummaryFinal <- factorSummary %>%
  tidyr::pivot_longer("MSZoning":"SaleType", names_to = "variable", values_to = "value") %>%
  tidyr::pivot_wider(names_from = stat, values_from = value) %>%
  dplyr::mutate(missing_pct = 100*as.numeric(missing)/as.numeric(n),#compute missing_pct
         unique_pct = 100*as.numeric(unique)/as.numeric(n), #unique percent
         freqRatio = as.numeric(first_mode_freq)/as.numeric(second_mode_freq)) %>% #freqRatio as define
  dplyr::select(variable, n, missing, missing_pct, unique, unique_pct,freqRatio, everything())

options(digits=3)
options(scipen=99)
factorSummaryFinal %>% kable() #display nicely
```

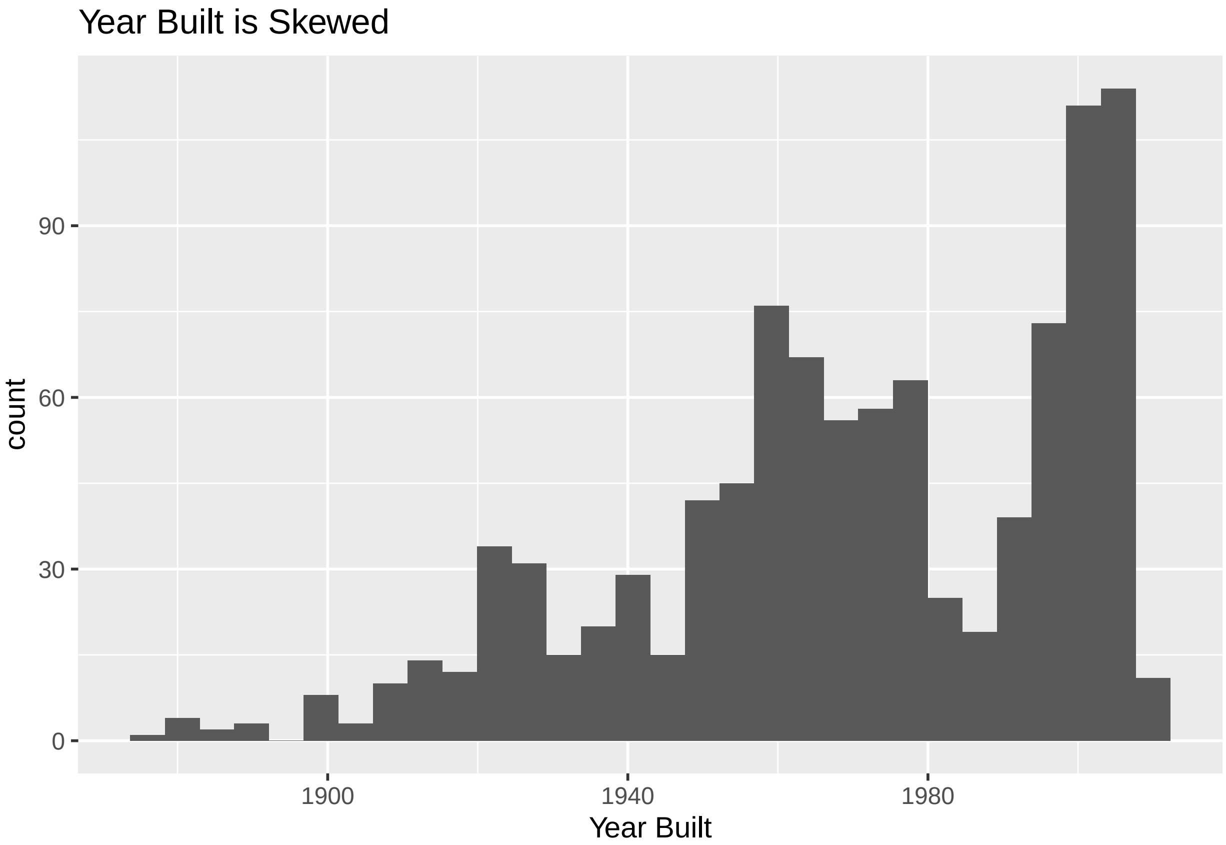
variable	n	missing	missing_pct	unique	unique_pct	freqRatio	1st mode	first_mode_freq	2nd mode	second_mode_freq	least common	least common freq
MSZoning	1000	0	0.0	4	0.4	5.32	RL	803	RM	151	RH	10
Alley	1000	938	93.8	3	0.3	1.82	Grvl	40	Pave	22	Pave	22
LotShape	1000	0	0.0	4	0.4	1.92	Reg	633	IR1	330	IR3	7
LandContour	1000	0	0.0	4	0.4	22.62	Lvl	905	Bnk	40	Low	26
LotConfig	1000	0	0.0	4	0.4	3.97	Inside	711	Corner	179	other	38
LandSlope	1000	0	0.0	3	0.3	19.71	Gtl	946	Mod	48	Sev	6
Neighborhood	1000	0	0.0	18	1.8	1.48	NAmes	167	CollgCr	113	Timber	20
Condition1	1000	0	0.0	6	0.6	17.08	Norm	871	Feedr	51	PosA	7
BldgType	1000	0	0.0	5	0.5	10.33	1Fam	837	TwnhsE	81	2fmCon	20
HouseStyle	1000	0	0.0	8	0.8	1.57	1Story	488	2Story	310	2.5Fin	5
RoofStyle	1000	0	0.0	3	0.3	4.32	Gable	795	Hip	184	other	21
Exterior1st	1000	0	0.0	8	0.8	1.87	VinylSd	328	HdBoard	175	CemntBd	36

											least	least
							1st		2nd		com-	common
variable	n	missing	missing_pct	unique	unique_pct	freqRatio	mode	first_mode_freq	mode	second_mode_freq	mon	freq
Exterior2nd	1000	0	0.0	9	0.9	2.01	VinylSd	320	HdBoard	159	BrkFace	24
MasVnrType	1000	4	0.4	5	0.5	1.97	None	617	BrkFace	313	BrkCmn	8
ExterQual	1000	0	0.0	3	0.3	1.96	Avg	657	AboveAvg	336	BelowAvg	7
ExterCond	1000	0	0.0	3	0.3	8.54	Avg	880	AboveAvg	103	BelowAvg	17
Foundation	1000	0	0.0	4	0.4	1.12	CBlock	463	PConc	414	other	27
BsmtQual	1000	31	3.1	4	0.4	1.06	AboveAvg	488	Avg	459	BelowAvg	22
BsmtCond	1000	31	3.1	4	0.4	24.41	Avg	903	AboveAvg	37	BelowAvg	29
BsmtExposure	1000	32	3.2	5	0.5	4.77	No	668	Av	140	Mn	76
BsmtFinType1	1000	31	3.1	7	0.7	1.03	GLQ	273	Unf	265	LwQ	52
BsmtFinType2	1000	32	3.2	7	0.7	23.69	Unf	853	Rec	36	ALQ	11
Heating	1000	0	0.0	2	0.2	37.46	GasA	974	other	26	other	26
HeatingQC	1000	0	0.0	3	0.3	2.21	AboveAvg	664	Avg	300	BelowAvg	36
CentralAir	1000	0	0.0	2	0.2	14.62	Y	936	N	64	N	64
Electrical	1000	1	0.1	5	0.5	12.61	SBrkr	908	FuseA	72	FuseP	2
KitchenQual	1000	0	0.0	3	0.3	1.22	Avg	534	AboveAvg	439	BelowAvg	27
Functional	1000	0	0.0	6	0.6	35.54	Typ	924	Min2	26	Maj2	4
FireplaceQu	1000	466	46.6	4	0.4	1.04	AboveAvg	250	Avg	240	BelowAvg	44
GarageType	1000	53	5.3	7	0.7	2.15	Attchd	601	Detchd	280	2Types	3
GarageFinish	1000	53	5.3	4	0.4	1.49	Unf	434	RFn	291	Fin	222
GarageQual	1000	53	5.3	4	0.4	27.48	Avg	907	BelowAvg	33	AboveAvg	7
GarageCond	1000	53	5.3	4	0.4	29.36	Avg	910	BelowAvg	31	AboveAvg	6
PavedDrive	1000	0	0.0	3	0.3	14.71	Y	912	N	62	P	26
PoolQC	1000	998	99.8	3	0.3	1.00	Fa	1	Gd	1	Fa	1
Fence	1000	805	80.5	5	0.5	2.70	MnPrv	108	GdPrv	40	MnWw	8
MiscFeature	1000	966	96.6	3	0.3	16.00	Shed	32	Othr	2	Othr	2
SaleType	1000	0	0.0	2	0.2	33.48	WD	971	other	29	other	29

2.a.

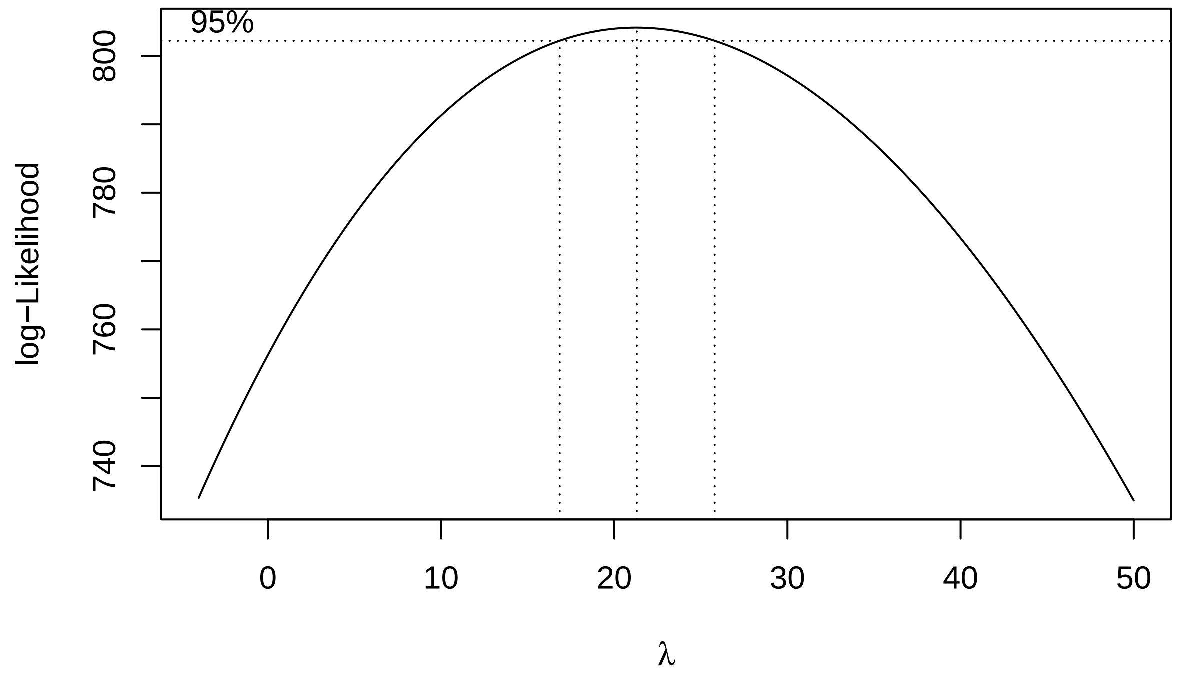
I notice two examples of skewed data, `YearBuilt` and `SalePrice`. The skews are in opposite directions so that is fun! First I try the Year built. Here is the visualization.

```r
yb <- ggplot(data = housingData, aes(x = YearBuilt)) +
  geom_histogram() +
  ggtitle("Year Built is Skewed") +
  xlab("Year Built")
yb
```



Very bimodal and skewed to the current time. Apply the `boxcox` function.

```
b<-boxcox(lm(housingData$YearBuilt~1), lambda = seq(-4,50,1/10)) #tweaked the limits until found the co
```



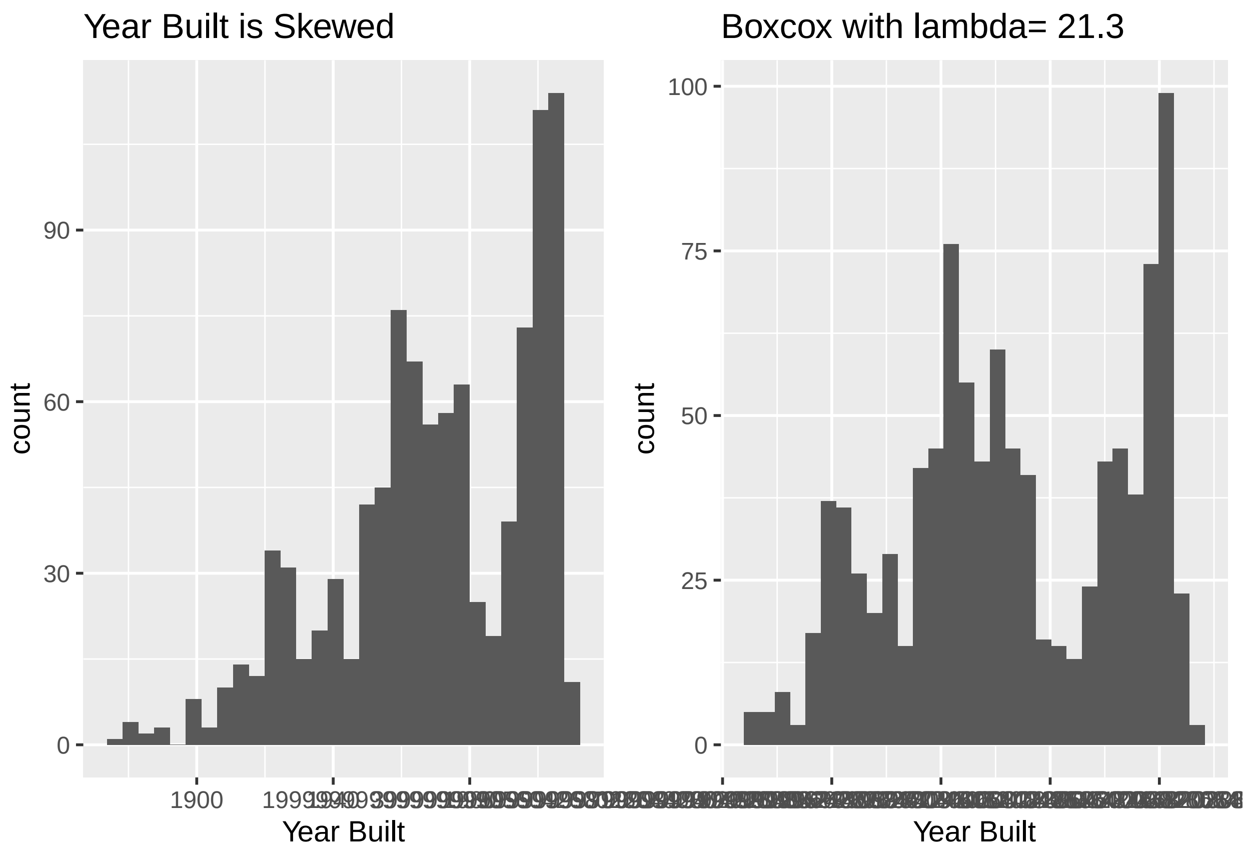
```r
lambda <- b$x[which.max(b$y)] #find the max
lambda
```

```
## [1] 21.3
```

This is the optimal λ.

```r
mytitle <- paste("Boxcox with lambda=", lambda)
ybm <- housingData %>%
  mutate(YearBuiltMod = (YearBuilt^lambda-1)/lambda) %>%
  ggplot( aes(x = YearBuiltMod)) +
  geom_histogram() +
  ggtitle(mytitle) +
  xlab("Year Built")

grid.arrange(yb,ybm, nrow = 1)
```

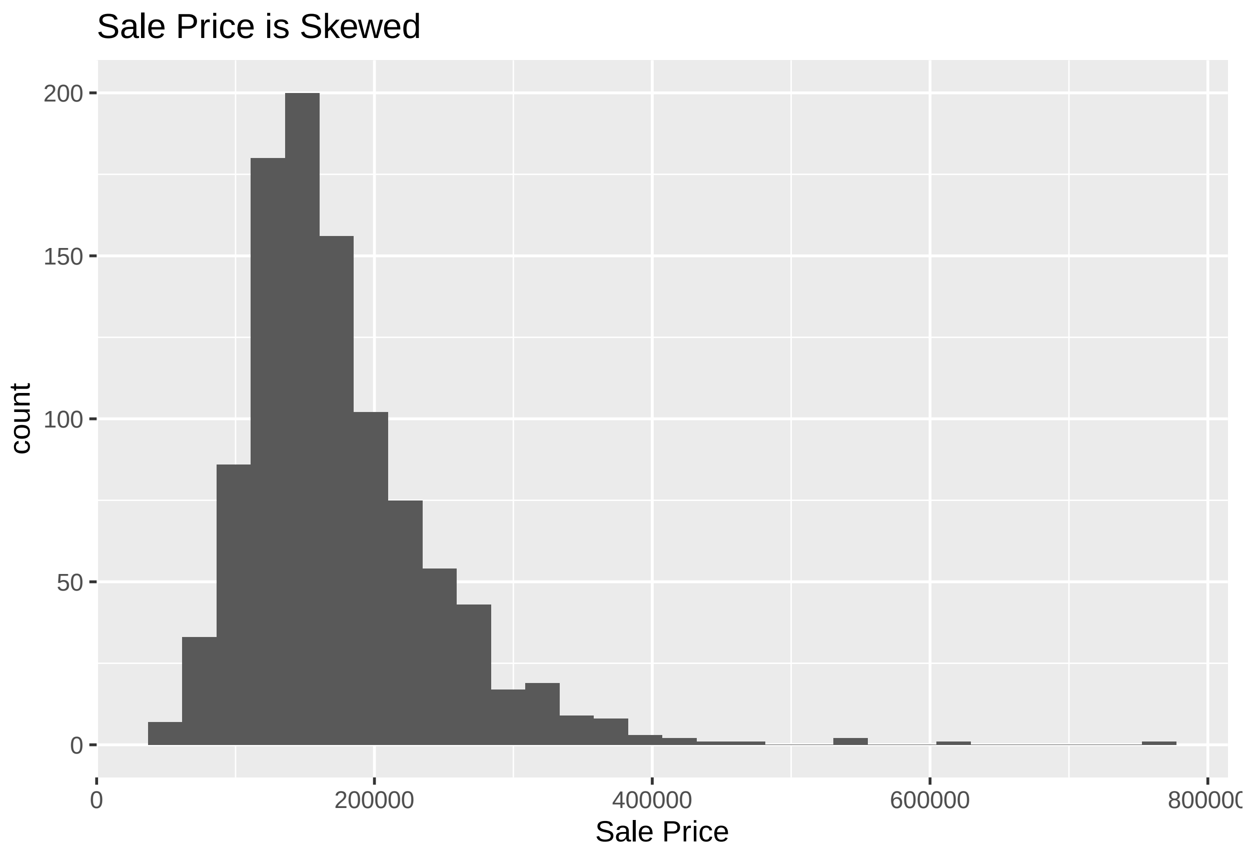


This is much better but by no means perfect. Looks like the year built was quite a difficult column due to the bi-modal distribution.

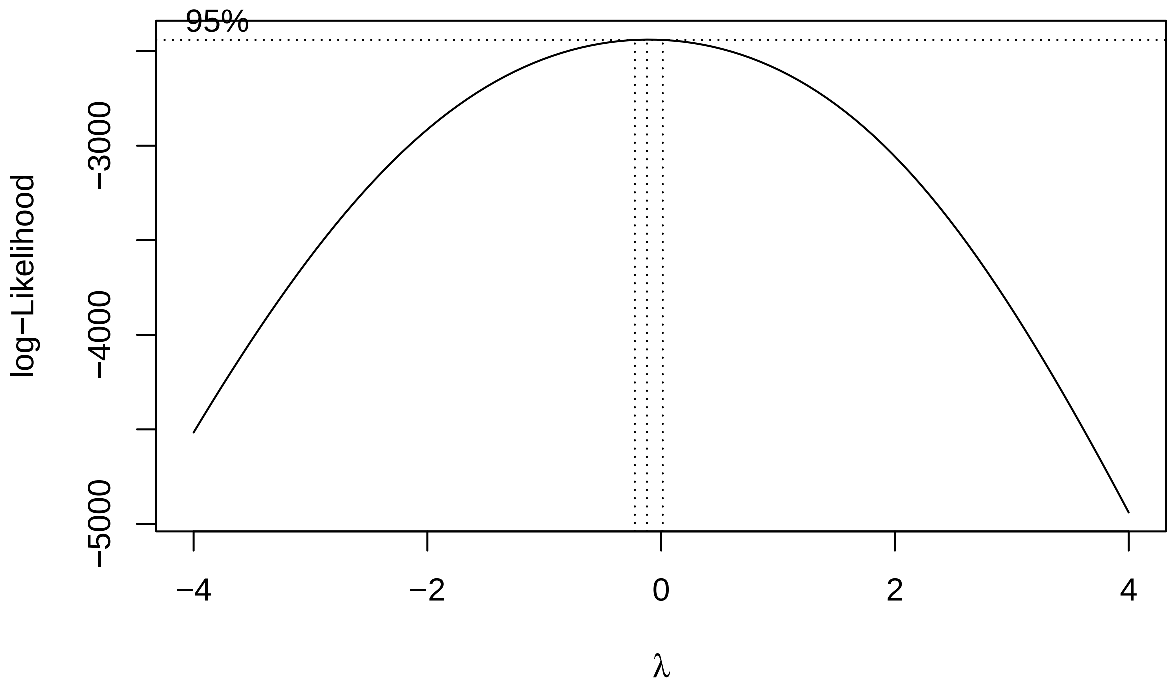
I am going to look at the Sale Price. These are notorious for not being normal.

```
sp <- ggplot(data = housingData, aes(x = SalePrice)) +
  geom_histogram() +
  ggtitle("Sale Price is Skewed") +
  xlab("Sale Price")

sp
```



```
b<-boxcox(lm(housingData$SalePrice~1), lambda = seq(-4,4,1/10))
```

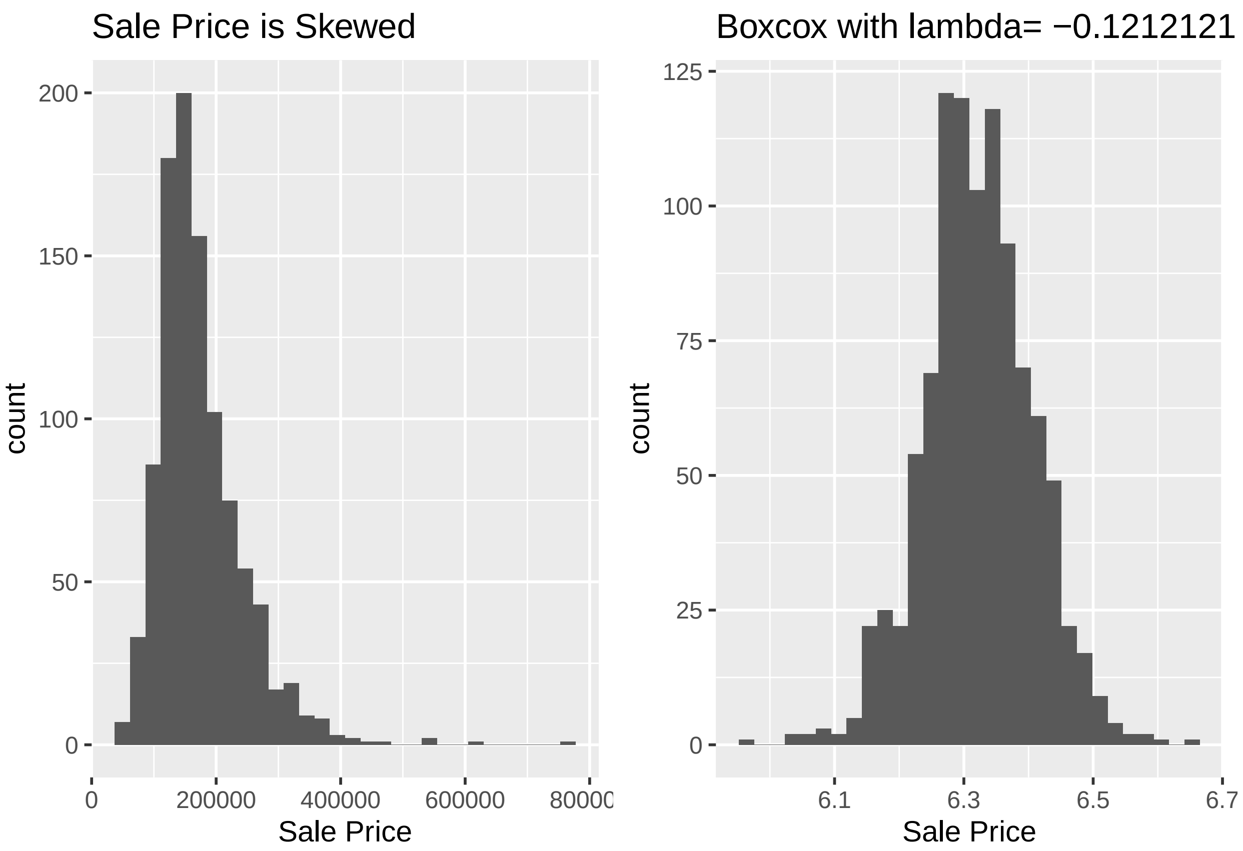


```
lambda <- b$x[which.max(b$y)]
lambda
```

```
## [1] -0.121
```

```
mytitle <- paste("Boxcox with lambda=", lambda)
spm <- housingData %>%
  mutate(SalePriceMod = (SalePrice^lambda -1)/lambda) %>%
  ggplot( aes(x = SalePriceMod)) +
  geom_histogram() +
  ggtitle(mytitle) +
  xlab("Sale Price")

grid.arrange(sp,spm, nrow = 1)
```



That looks much better. Quite normal and ready for analysis.

2.b.i.

Looking to `LotFrontage`, we see lots of missing values.

```
missing <- is.na(housingData$LotFrontage) #find the missing values
sum(missing) #give a total
```

```
## [1] 207
```

We impute first by replacing it with the mean.

```
avg <- mean(housingData$LotFrontage, na.rm = TRUE) #get the mean before imputing
housingData <- housingData %>%
  mutate(LFMean = if_else(is.na(LotFrontage), avg,LotFrontage)) #create a new column imputed with the m
```

b.ii.

Now we'll impute with a linear regression and some error depending on that regression. Since we are trying to predict something about the lot, I keep only variables with information about the outside of the house. I could not use `Alley` because it did not have enough levels to fit the linear model. I could not use `LotShape` due to some shapes not being in the training data. I could not use `Fence` either due to many missing values.

```
names(housingData)[c(4,5,8,9)] #find the names of variables that will work
```

```
## [1] "LotFrontage" "LotArea"     "LandContour" "LotConfig"
```

```
fit <- lm(LotFrontage ~ ., data=housingData[,names(housingData)[c(4,5,8,9)]]) #do linear fit
summary(fit)
```

```
## 
## Call:
## lm(formula = LotFrontage ~ ., data = housingData[, names(housingData)[c(4, 
##     5, 8, 9)]])
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -115.31  -11.03   -0.33   10.11  219.58 
## 
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)    
## (Intercept)       73.18464    3.99768   18.31 < 0.0000000000000002 ***
## LotArea            0.00103    0.00009   11.48 < 0.0000000000000002 ***
## LandContourHLS     6.57561    5.33736    1.23                0.218    
## LandContourLow   -15.37132    7.66387   -2.01                0.045 *  
## LandContourLvl    -1.66336    3.56385   -0.47                0.641    
## LotConfigCulDSac -30.01658    3.97942   -7.54     0.00000000000013 ***
## LotConfigInside  -14.91123    1.95984   -7.61     0.00000000000008 ***
## LotConfigother   -21.95523    4.33070   -5.07     0.00000049743063 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 20.5 on 785 degrees of freedom
##   (207 observations deleted due to missingness)
## Multiple R-squared:  0.224,  Adjusted R-squared:  0.217 
## F-statistic: 32.4 on 7 and 785 DF,  p-value: <0.0000000000000002
```

We see that the fit is decent with most of the variables that we have used showing significance.

Now we make predictions and impute.

```r
pred <- predict(fit,housingData[,names(housingData)[c(4,5,8,9)]] ) #create predictions
se <- summary(fit)[[6]] #standard error of the fit
housingData <- housingData%>%
  mutate(LFLM = if_else(is.na(LotFrontage),pred + rnorm(length(pred),0,se),LotFrontage)) #add the new c
```

b.iii.

We use the `mice` package for predictive mean matching. Here I used LotArea and SalePrice to build the model. This package would not work with any missing values.

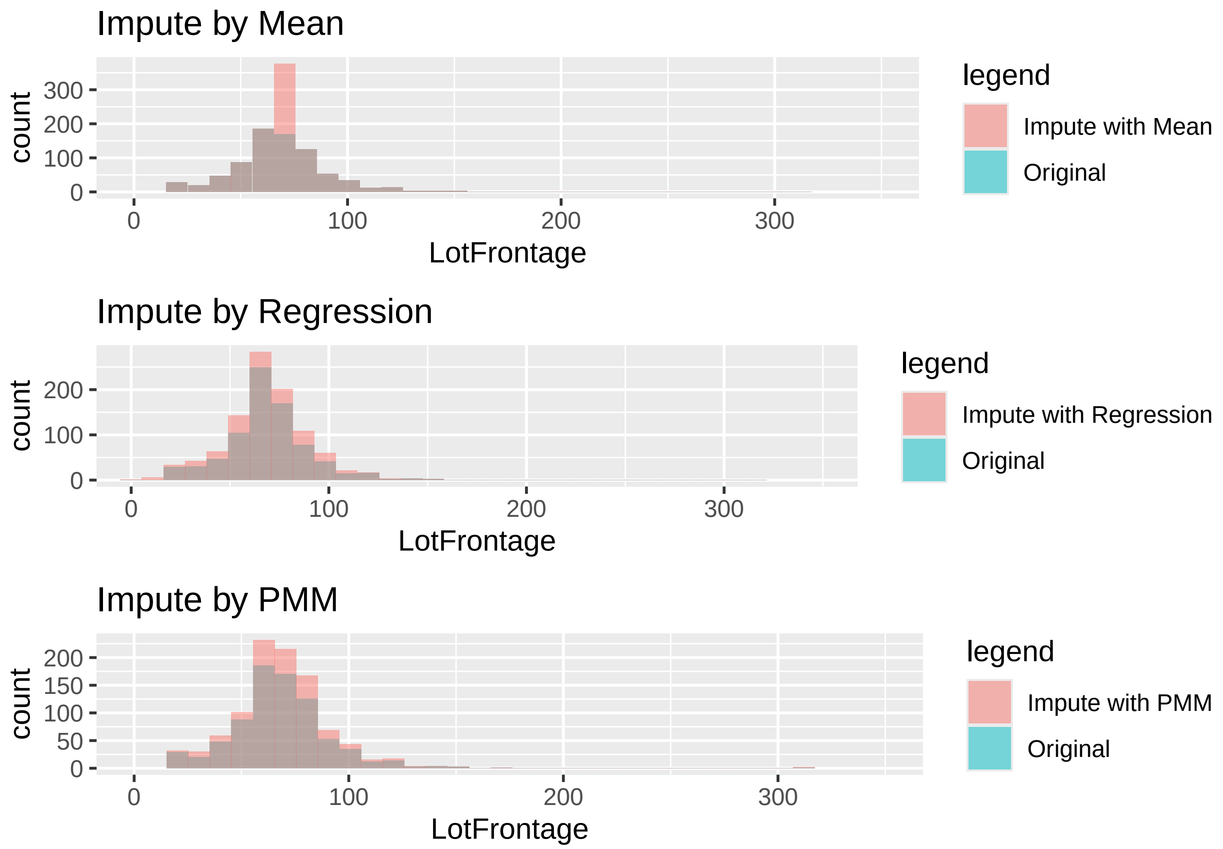
```r
housingData$LFPMM <- housingData$LotFrontage #create a new column with all the values
housingData[missing,"LFPMM"] =  mice.impute.pmm(housingData$LotFrontage,!missing,housingData[,names(hous
```

b.iv.

Time to visualize.

```r
colors <- c("Original" = "blue", "Impute with Mean" = "yellow", "Impute with Regression" = "red", "Impu
g1 <- ggplot(housingData)+
  geom_histogram(aes(x = LotFrontage, fill = "Original"),alpha = 0.5) +
  geom_histogram(aes(x = LFMean, fill = "Impute with Mean"),alpha = 0.5) +
  labs(title = "Impute by Mean",  fill = "legend")+
  scale_color_manual(values = colors) +
  coord_cartesian(xlim = c(0,350))
g2 <- ggplot(housingData)+
  geom_histogram(aes(x = LotFrontage, fill = "Original"),alpha = 0.5) +
  geom_histogram(aes(x = LFLM, fill = "Impute with Regression"),alpha = 0.5) +
  labs(title = "Impute by Regression",  fill = "legend")+
  scale_color_manual(values = colors) +
  coord_cartesian(xlim = c(0,350))
g3 <- ggplot(housingData)+
  geom_histogram(aes(x = LotFrontage, fill = "Original"),alpha = 0.5) +
  geom_histogram(aes(x = LFPMM, fill = "Impute with PMM"),alpha = 0.5) +
  labs(title = "Impute by PMM",  fill = "legend")+
  scale_color_manual(values = colors) +
  coord_cartesian(xlim = c(0,350))

grid.arrange(g1,g2,g3)
```



We see some of what was expected. The mean imputation really returns that same value a lot. The regression imputation is better and the pmm seems best.

2.c. Create 5 levels for the variable `Exterior1st`. Here is the original counts sorted.

```
housingData %>%
  dplyr::count(Exterior1st, sort = TRUE) #somewhere I masked the dplyr count function
```

```
##   Exterior1st   n
## 1     VinylSd 328
## 2     HdBoard 175
## 3     MetalSd 153
## 4     Wd Sdng 141
## 5     Plywood  73
## 6       other  52
## 7     BrkFace  42
## 8     CemntBd  36
```

```
housingData %>%
  mutate(Exterior1st = fct_lump(Exterior1st,n=4)) %>% #this will create 4 categories with the 5th being
  dplyr::count(Exterior1st, sort = TRUE)
```

```
##   Exterior1st   n
## 1     VinylSd 328
## 2       Other 203
## 3     HdBoard 175
## 4     MetalSd 153
## 5     Wd Sdng 141
```

2.d.i.

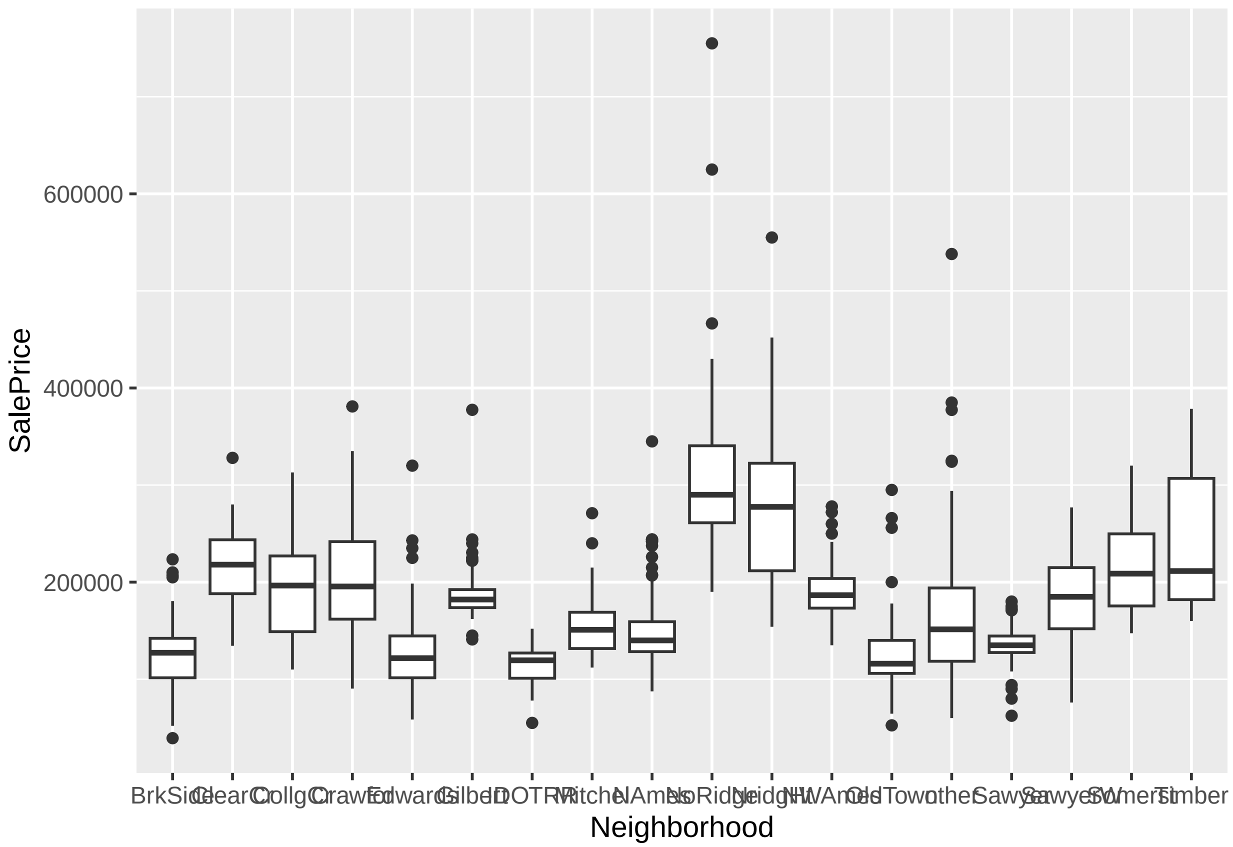
We use `dplyr` again for this noticing that some of the functions need to be called with package name.

```
housingData %>%
  dplyr::group_by(Neighborhood) %>%
  dplyr::summarise(average = mean(SalePrice)) %>%
  arrange(desc(average))
```

```
## # A tibble: 18 x 2
##    Neighborhood average
##    <chr>          <dbl>
##  1 NoRidge      328794.
##  2 NridgHt      283057.
##  3 Timber       241940
##  4 ClearCr      218265.
##  5 Somerst      211678.
##  6 Crawfor      209766.
##  7 CollgCr      194942.
##  8 NWAmes       191823.
##  9 Gilbert      189466.
## 10 SawyerW      183971.
## 11 other        170248.
## 12 Mitchel      154788.
## 13 NAmes        146669.
## 14 Sawyer       134708.
## 15 Edwards      128772.
## 16 OldTown      126023.
## 17 BrkSide      124844.
## 18 IDOTRR       114319.
```

2.d.ii. Create a boxplot of the saleprice with neighborhoods.

```
ggplot(housingData, aes(y = SalePrice, x = Neighborhood)) +
  geom_boxplot()
```



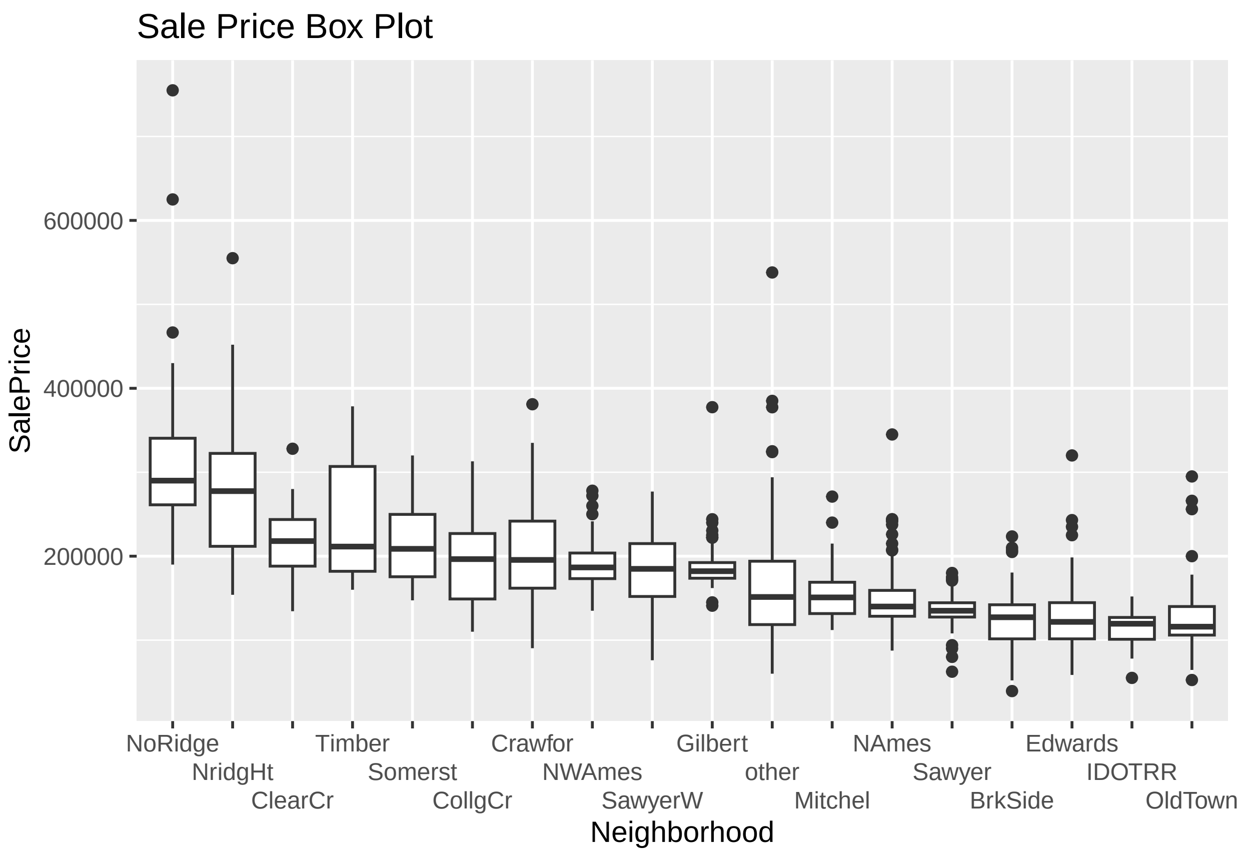
2.d.iii.

```
housingData <- housingData %>%
  mutate(Neighborhood = factor(Neighborhood)) %>% #turn the data into a factor
  mutate(Neighborhood = fct_reorder(Neighborhood,SalePrice, .desc = TRUE)) #reorder the data uses media
```

2.d.iv.

Since the data has been reordered, we only need to call ggplot.

```
housingData %>%
  ggplot(aes(x= Neighborhood, y = SalePrice)) +
  geom_boxplot() +
  labs(title = "Sale Price Box Plot", x = "Neighborhood")+
  scale_x_discrete(guide = guide_axis(n.dodge=3)) #just to get the variables to dodge
```


Sale Price Box Plot
SalePrice
600000
400000
200000
NoRidge
NridgHt
ClearCr
Timber
Somerst
CollgCr
Crawfor
NWAmes
SawyerW
Gilbert
other
Mitchel
NAmes
Sawyer
BrkSide
Edwards
IDOTRR
OldTown
Neighborhood