# Group 5 HW 4

Vignesh Murugan and Tyler Brassfield

2024-09-22

## 1 Data Quality Report

(a) After loading the housing data into a data frame (or tibble) named housingData, run the code listed below to create three new variables. housingData <- housingData %>% dplyr::mutate(age = YrSold - YearBuilt, ageSinceRemodel = YrSold - YearRemodAdd,

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

---

(b) (2 points) Use the dplyr package to create a tibble named housingNumeric which contains all of the numeric variables from the original data. Please use the dplyr::select command along with the is.numeric function to complete this task.

```
housingNumeric <- housingData %>% select_if(is.numeric)
```

---

(c) (2 points) Use the dplyr package to create a tibble named housingFactor which contains all of the numeric variables from the original data. You can use dplyr::select command here or, if you like, consider the transmute command to simultaneously keep only the character variables and change all character variables to factors.

```
housingFactor <- housingData %>% transmute_if(is.character, as.factor)
```

---

(d) Try the glimpse command to take a look at your new tibbles.

```
glimpse(housingNumeric)
```

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

```
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~
```

---

(e) (4 points) Unfortunately, R does not have a method for extracting only Q1 or Q3. So, we will create our own user-defined functions to do this for us. Use the following code to create two new functions, Q1 and Q3, respectively. 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] } Briefly explain what these two new functions are doing.

```r
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]
}

# In the quantile function in R, when you compute quartiles of a data vector
# without specifying any specific probabilities,
# it returns a named vector of percentiles that typically include the following ->
# Index [1] - Minimum: It retrieves the minimum value of the data set.
# Index [2] - First Quartile: It retrieves the first quartile.
# Index [3] - Median: It retrieves the median of the data set.
# Index [4] - Third Quartile: It retrieves the third quartile.
# Index [5] - Maximum: It retrieves the maximum value of the data set.
# Q1 access the index 2, the first quartile and Q3 access the index 4, the third quartile.
```

---

(f) Next, we are going to create a new function that will apply several summary statistics to our data all at once. Create the new function myNumericSummary with the following code. 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)) } This code accepts a numerical vector x as an input parameter and then returns a vector where the first element is the length of the input vector (i.e., the number of observations), the second element is the number of unique values, the third is the number of missing values, the forth is the mean value of non-missing numerics, etc. Notice the use of our new functions Q1 and Q3.

```r
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))
}
```

---

(g) (8 points) Utitlize the dplyr::summarize command together with the new myNumericSummary function to apply the new function to every variable in the housingNumeric data set. You may need to look up some examples of how to use summmarize and the across() syntax from dplyr to do this efficiently. Save the results of this operation in a new tibble named numericSummary.

```r
numericSummary <- housingNumeric %>%
  summarize(across(everything(), myNumericSummary))
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
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~
```

---

(h) Next, column bind some labels to our summary statistics with the following code. numericSummary <- cbind( stat=c(“n”,“unique”,“missing”,“mean”,“min”,“Q1”,“median”,“Q3”,“max”,“sd”), numericSummary) If you glimpse the results, it should look something like Figure 3.

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

```
## Rows: 10
## Columns: 40
## $ stat            <chr> "n", "unique", "missing", "mean", "min", "Q1", "median~
## $ 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~
```

---

(i) While this is good data here, you need to perform a little trick on it so we can use the kable function and produce the table we want, i.e., need to "pivot" the data a couple of times. You also need to add a couple more computed values: percent missing and percent unique fields. Use the following code to accomplish this. numericSummaryFinal <- numericSummary %>% pivot_longer("Id":"ageofGarage", names_to = "variable", values_to = "value") %>% pivot_wider(names_from = stat, values_from = value) %>% mutate(missing_pct = 100*missing/n, unique_pct = 100*unique/n) %>% select(variable, n, missing, missing_pct, unique, unique_pct, everything()) and finally, produce the first part of the Data Quality report, library(knitr) options(digits=3) options(scipen=99) numericSummaryFinal %>% kable()

```r
numericSummaryFinal <- numericSummary %>%
  pivot_longer("Id":"ageofGarage", names_to = "variable", values_to = "value") %>%
  pivot_wider(names_from = stat, values_from = value) %>%
  mutate(missing_pct = 100*missing/n,
         unique_pct = 100*unique/n) %>%
  select(variable, n, missing, missing_pct, unique, unique_pct, everything())
library(knitr)
options(digits=3)
options(scipen=99)
numericSummaryFinal %>% kable()
```

| 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 |
| 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 |

(j) (30 points) Create the second part of the Data Quality report associated with the non-numeric data. See Figure 2 for a report excerpt. Note: R does not have functions for identifying the first, second, or least commmon modes. Use the code below to accomplish this. 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”) } } Note: R does not have functions for identifying the frequencies of the first, second, or least commmon modes. Use the code below to accomplish this.

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”) } }

```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")
  }
}

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

FactorSummary <- housingFactor %>%
  summarize(across(everything(), myFactorSummary))
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
glimpse(FactorSummary)
```

```
## Rows: 9
## Columns: 38
## $ MSZoning     <chr> "1000", "4", "0", "RL", "803", "RM", "151", "RH", "10"
## $ Alley        <chr> "1000", "3", "938", "Grvl", "40", "Pave", "22", "Pave", "~
## $ LotShape     <chr> "1000", "4", "0", "Reg", "633", "IR1", "330", "IR3", "7"
## $ LandContour  <chr> "1000", "4", "0", "Lvl", "905", "Bnk", "40", "Low", "26"
## $ LotConfig    <chr> "1000", "4", "0", "Inside", "711", "Corner", "179", "othe~
## $ LandSlope    <chr> "1000", "3", "0", "Gtl", "946", "Mod", "48", "Sev", "6"
## $ Neighborhood <chr> "1000", "18", "0", "NAmes", "167", "CollgCr", "113", "Tim~
## $ Condition1   <chr> "1000", "6", "0", "Norm", "871", "Feedr", "51", "PosA", "~
## $ BldgType     <chr> "1000", "5", "0", "1Fam", "837", "TwnhsE", "81", "2fmCon"~
## $ HouseStyle   <chr> "1000", "8", "0", "1Story", "488", "2Story", "310", "2.5F~
## $ RoofStyle    <chr> "1000", "3", "0", "Gable", "795", "Hip", "184", "other", ~
## $ Exterior1st  <chr> "1000", "8", "0", "VinylSd", "328", "HdBoard", "175", "Ce~
## $ Exterior2nd  <chr> "1000", "9", "0", "VinylSd", "320", "HdBoard", "159", "Br~
## $ MasVnrType   <chr> "1000", "5", "4", "None", "617", "BrkFace", "313", "BrkCm~
## $ ExterQual    <chr> "1000", "3", "0", "Avg", "657", "AboveAvg", "336", "Below~
## $ ExterCond    <chr> "1000", "3", "0", "Avg", "880", "AboveAvg", "103", "Below~
## $ Foundation   <chr> "1000", "4", "0", "CBlock", "463", "PConc", "414", "other~
## $ BsmtQual     <chr> "1000", "4", "31", "AboveAvg", "488", "Avg", "459", "Belo~
## $ BsmtCond     <chr> "1000", "4", "31", "Avg", "903", "AboveAvg", "37", "Below~
## $ BsmtExposure <chr> "1000", "5", "32", "No", "668", "Av", "140", "Mn", "76"
## $ BsmtFinType1 <chr> "1000", "7", "31", "GLQ", "273", "Unf", "265", "LwQ", "52"
## $ BsmtFinType2 <chr> "1000", "7", "32", "Unf", "853", "Rec", "36", "ALQ", "11"
## $ Heating      <chr> "1000", "2", "0", "GasA", "974", "other", "26", "other", ~
## $ HeatingQC    <chr> "1000", "3", "0", "AboveAvg", "664", "Avg", "300", "Below~
## $ CentralAir   <chr> "1000", "2", "0", "Y", "936", "N", "64", "N", "64"
## $ Electrical   <chr> "1000", "5", "1", "SBrkr", "908", "FuseA", "72", "FuseP",~
## $ KitchenQual  <chr> "1000", "3", "0", "Avg", "534", "AboveAvg", "439", "Below~
## $ Functional   <chr> "1000", "6", "0", "Typ", "924", "Min2", "26", "Maj2", "4"
## $ FireplaceQu  <chr> "1000", "4", "466", "AboveAvg", "250", "Avg", "240", "Bel~
## $ GarageType   <chr> "1000", "7", "53", "Attchd", "601", "Detchd", "280", "2Ty~
## $ GarageFinish <chr> "1000", "4", "53", "Unf", "434", "RFn", "291", "Fin", "22~
## $ GarageQual   <chr> "1000", "4", "53", "Avg", "907", "BelowAvg", "33", "Above~
## $ GarageCond   <chr> "1000", "4", "53", "Avg", "910", "BelowAvg", "31", "Above~
## $ PavedDrive   <chr> "1000", "3", "0", "Y", "912", "N", "62", "P", "26"
## $ PoolQC       <chr> "1000", "3", "998", "Fa", "1", "Gd", "1", "Fa", "1"
## $ Fence        <chr> "1000", "5", "805", "MnPrv", "108", "GdPrv", "40", "MnWw"~
## $ MiscFeature  <chr> "1000", "3", "966", "Shed", "32", "Othr", "2", "Othr", "2"
## $ SaleType     <chr> "1000", "2", "0", "WD", "971", "other", "29", "other", "2~
```

```
FactorSummary <-cbind(
  stat=c("n","unique","missing","most_common","most_common_count","2nd_most_common","2nd_most_common_cou
         "least_common","least_common_count"),
  FactorSummary)
glimpse(FactorSummary)
```

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

```
FactorSummaryFinal <- FactorSummary %>%
  pivot_longer("MSZoning":"SaleType", names_to = "variable", values_to = "value") %>%
  pivot_wider(names_from = stat, values_from = value)
glimpse(FactorSummaryFinal)
```

```
## Rows: 38
## Columns: 10
## $ variable                <chr> "MSZoning", "Alley", "LotShape", "LandContour"~
## $ n                       <chr> "1000", "1000", "1000", "1000", "1000", "1000"~
## $ unique                  <chr> "4", "3", "4", "4", "4", "3", "18", "6", "5", ~
## $ missing                 <chr> "0", "938", "0", "0", "0", "0", "0", "0", "0",~
## $ most_common             <chr> "RL", "Grvl", "Reg", "Lvl", "Inside", "Gtl", "~
## $ most_common_count       <chr> "803", "40", "633", "905", "711", "946", "167"~
## $ ‘2nd_most_common‘       <chr> "RM", "Pave", "IR1", "Bnk", "Corner", "Mod", "~
## $ ‘2nd_most_common_count‘ <chr> "151", "22", "330", "40", "179", "48", "113", ~
## $ least_common            <chr> "RH", "Pave", "IR3", "Low", "other", "Sev", "T~
## $ least_common_count      <chr> "10", "22", "7", "26", "38", "6", "20", "7", "~
```

```
FactorSummaryFinal$n <- as.numeric(FactorSummaryFinal$n)
FactorSummaryFinal$unique <- as.numeric(FactorSummaryFinal$unique)
FactorSummaryFinal$missing <- as.numeric(FactorSummaryFinal$missing)
FactorSummaryFinal <- FactorSummaryFinal %>%
  mutate(missing_pct = 100*missing/n,
         unique_pct = 100*unique/n) %>%
  select(variable, n, missing, missing_pct, unique, unique_pct, everything())
library(knitr)
options(digits=3)
options(scipen=99)
FactorSummaryFinal %>% kable()
```

| variable | n | missing | missing_pct | unique | unique_pct | most_common | most_common_count | 2nd_most_common | 2nd_most_common_count | least_common | least_common_count |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MSZoning | 1000 | 0 | 0.0 | 4 | 0.4 | RL | 803 | RM | 151 | RH | 10 |
| Alley | 1000 | 938 | 93.8 | 3 | 0.3 | Grvl | 40 | Pave | 22 | Pave | 22 |
| LotShape | 1000 | 0 | 0.0 | 4 | 0.4 | Reg | 633 | IR1 | 330 | IR3 | 7 |
| LandContour | 1000 | 0 | 0.0 | 4 | 0.4 | Lvl | 905 | Bnk | 40 | Low | 26 |
| LotConfig | 1000 | 0 | 0.0 | 4 | 0.4 | Inside | 711 | Corner | 179 | other | 38 |
| LandSlope | 1000 | 0 | 0.0 | 3 | 0.3 | Gtl | 946 | Mod | 48 | Sev | 6 |
| Neighborhood | 1000 | 0 | 0.0 | 18 | 1.8 | NAmes | 167 | CollgCr | 113 | Timber | 20 |
| Condition1 | 1000 | 0 | 0.0 | 6 | 0.6 | Norm | 871 | Feedr | 51 | PosA | 7 |
| BldgType | 1000 | 0 | 0.0 | 5 | 0.5 | 1Fam | 837 | TwnhsE | 81 | 2fmCon | 20 |
| HouseStyle | 1000 | 0 | 0.0 | 8 | 0.8 | 1Story | 488 | 2Story | 310 | 2.5Fin | 5 |
| RoofStyle | 1000 | 0 | 0.0 | 3 | 0.3 | Gable | 795 | Hip | 184 | other | 21 |
| Exterior1st | 1000 | 0 | 0.0 | 8 | 0.8 | VinylSd | 328 | HdBoard | 175 | CemntBd | 36 |
| Exterior2nd | 1000 | 0 | 0.0 | 9 | 0.9 | VinylSd | 320 | HdBoard | 159 | BrkFace | 24 |
| MasVnrType | 1000 | 4 | 0.4 | 5 | 0.5 | None | 617 | BrkFace | 313 | BrkCmn | 8 |
| ExterQual | 1000 | 0 | 0.0 | 3 | 0.3 | Avg | 657 | AboveAvg | 336 | BelowAvg | 7 |
| ExterCond | 1000 | 0 | 0.0 | 3 | 0.3 | Avg | 880 | AboveAvg | 103 | BelowAvg | 17 |

| variable | n | missing | missing_pct | unique | unique_pct | most_common | most_common_count | 2nd_most_common | 2nd_most_common_count | least_common | least_common_count |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Foundation | 1000 | 0 | 0.0 | 4 | 0.4 | CBlock | 463 | PConc | 414 | other | 27 |
| BsmtQual | 1000 | 31 | 3.1 | 4 | 0.4 | AboveAvg | 488 | Avg | 459 | BelowAvg | 22 |
| BsmtCond | 1000 | 31 | 3.1 | 4 | 0.4 | Avg | 903 | AboveAvg | 37 | BelowAvg | 29 |
| BsmtExposure | 1000 | 32 | 3.2 | 5 | 0.5 | No | 668 | Av | 140 | Mn | 76 |
| BsmtFinType1 | 1000 | 31 | 3.1 | 7 | 0.7 | GLQ | 273 | Unf | 265 | LwQ | 52 |
| BsmtFinType2 | 1000 | 32 | 3.2 | 7 | 0.7 | Unf | 853 | Rec | 36 | ALQ | 11 |
| Heating | 1000 | 0 | 0.0 | 2 | 0.2 | GasA | 974 | other | 26 | other | 26 |
| HeatingQC | 1000 | 0 | 0.0 | 3 | 0.3 | AboveAvg | 664 | Avg | 300 | BelowAvg | 36 |
| CentralAir | 1000 | 0 | 0.0 | 2 | 0.2 | Y | 936 | N | 64 | N | 64 |
| Electrical | 1000 | 1 | 0.1 | 5 | 0.5 | SBrkr | 908 | FuseA | 72 | FuseP | 2 |
| KitchenQual | 1000 | 0 | 0.0 | 3 | 0.3 | Avg | 534 | AboveAvg | 439 | BelowAvg | 27 |
| Functional | 1000 | 0 | 0.0 | 6 | 0.6 | Typ | 924 | Min2 | 26 | Maj2 | 4 |
| FireplaceQu | 1000 | 466 | 46.6 | 4 | 0.4 | AboveAvg | 250 | Avg | 240 | BelowAvg | 44 |
| GarageType | 1000 | 53 | 5.3 | 7 | 0.7 | Attchd | 601 | Detchd | 280 | 2Types | 3 |
| GarageFinish | 1000 | 53 | 5.3 | 4 | 0.4 | Unf | 434 | RFn | 291 | Fin | 222 |
| GarageQual | 1000 | 53 | 5.3 | 4 | 0.4 | Avg | 907 | BelowAvg | 33 | AboveAvg | 7 |
| GarageCond | 1000 | 53 | 5.3 | 4 | 0.4 | Avg | 910 | BelowAvg | 31 | AboveAvg | 6 |
| PavedDrive | 1000 | 0 | 0.0 | 3 | 0.3 | Y | 912 | N | 62 | P | 26 |
| PoolQC | 1000 | 998 | 99.8 | 3 | 0.3 | Fa | 1 | Gd | 1 | Fa | 1 |
| Fence | 1000 | 805 | 80.5 | 5 | 0.5 | MnPrv | 108 | GdPrv | 40 | MnWw | 8 |
| MiscFeature | 1000 | 966 | 96.6 | 3 | 0.3 | Shed | 32 | Othr | 2 | Othr | 2 |
| SaleType | 1000 | 0 | 0.0 | 2 | 0.2 | WD | 971 | other | 29 | other | 29 |

---

## 2 Transformations

(a) (8 points) Via visual inspection, identify two numeric variables that are highly skewed (e.g., not symmetric and far from normally distributed).Use a transformation method (e.g., ladder of powers or boxcox transformation) to transform these variables to be more normally distributed. Show visual depictions of distributions before/after transformations

```
glimpse(housingNumeric)
```

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

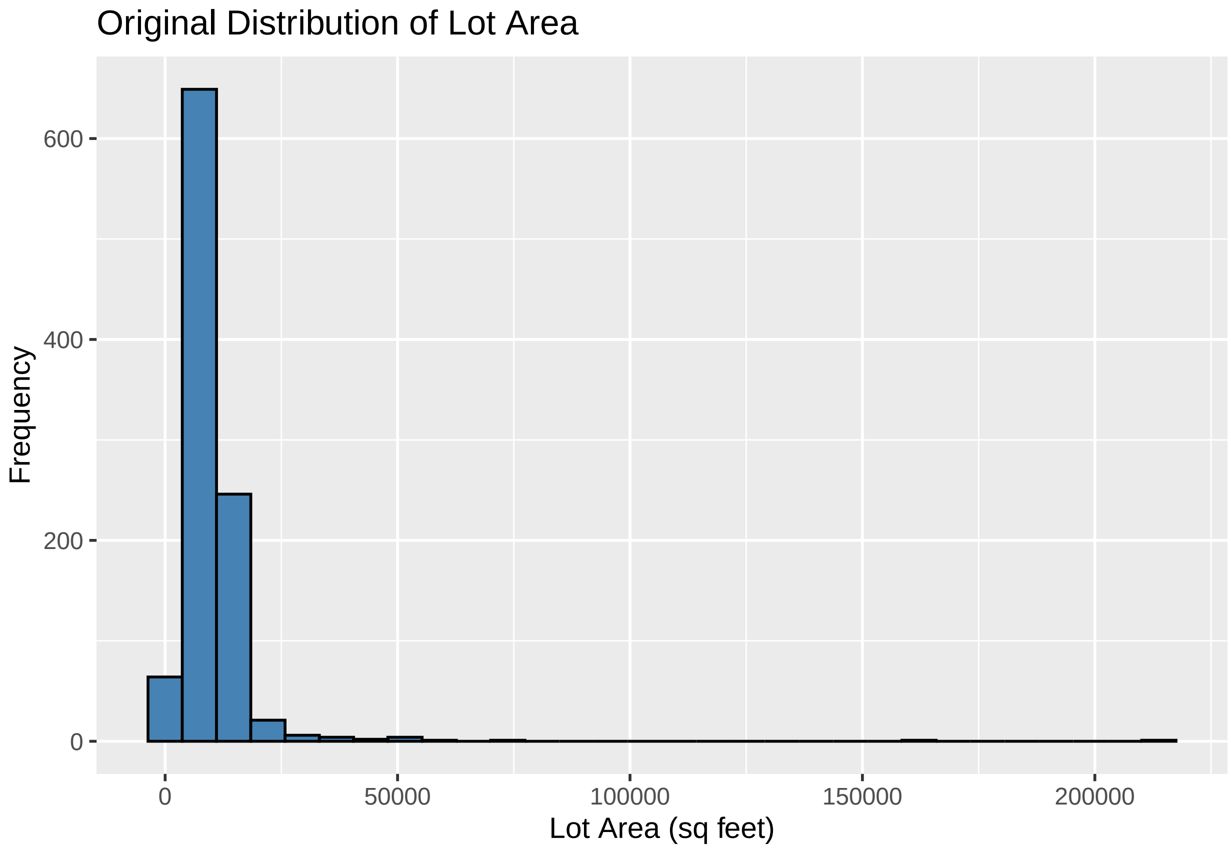
```
# Lot Area Tranformation
BoxCoxTrans(housingNumeric$LotArea)
```

```
## Box-Cox Transformation
##
## 1000 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1477    7500    9422   10425   11424  215245
##
## Largest/Smallest: 146
## Sample Skewness: 12.9
##
## Estimated Lambda: 0
## With fudge factor, Lambda = 0 will be used for transformations
```

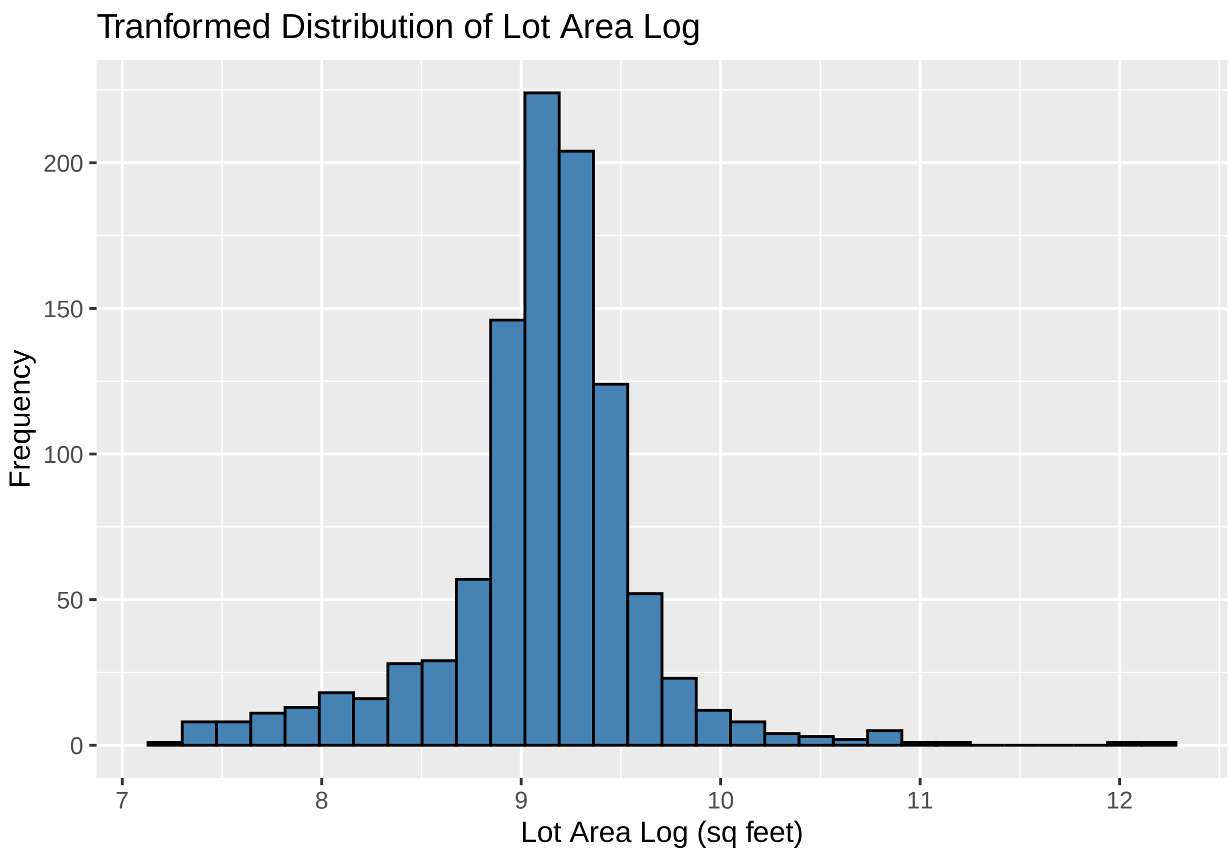
```
housingNumeric$LotAreaLog <- log(housingNumeric$LotArea)
BoxCoxTrans(housingNumeric$LotAreaLog)
```

```
## Box-Cox Transformation
##
## 1000 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.30    8.92    9.15    9.10    9.34   12.28
##
## Largest/Smallest: 1.68
## Sample Skewness: -0.129
##
## Estimated Lambda: 1.5
```

```r
# Plotting before transformation
ggplot(housingNumeric, aes(x = LotArea)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  ggtitle("Original Distribution of Lot Area") +
  xlab("Lot Area (sq feet)") +
  ylab("Frequency")
```



```r
# Plotting after transformation
ggplot(housingNumeric, aes(x = LotAreaLog)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  ggtitle("Tranformed Distribution of Lot Area Log") +
  xlab("Lot Area Log (sq feet)") +
  ylab("Frequency")
```



```r
# OpenPorchSF Transformation
BoxCoxTrans(housingNumeric$OpenPorchSF+1)
```

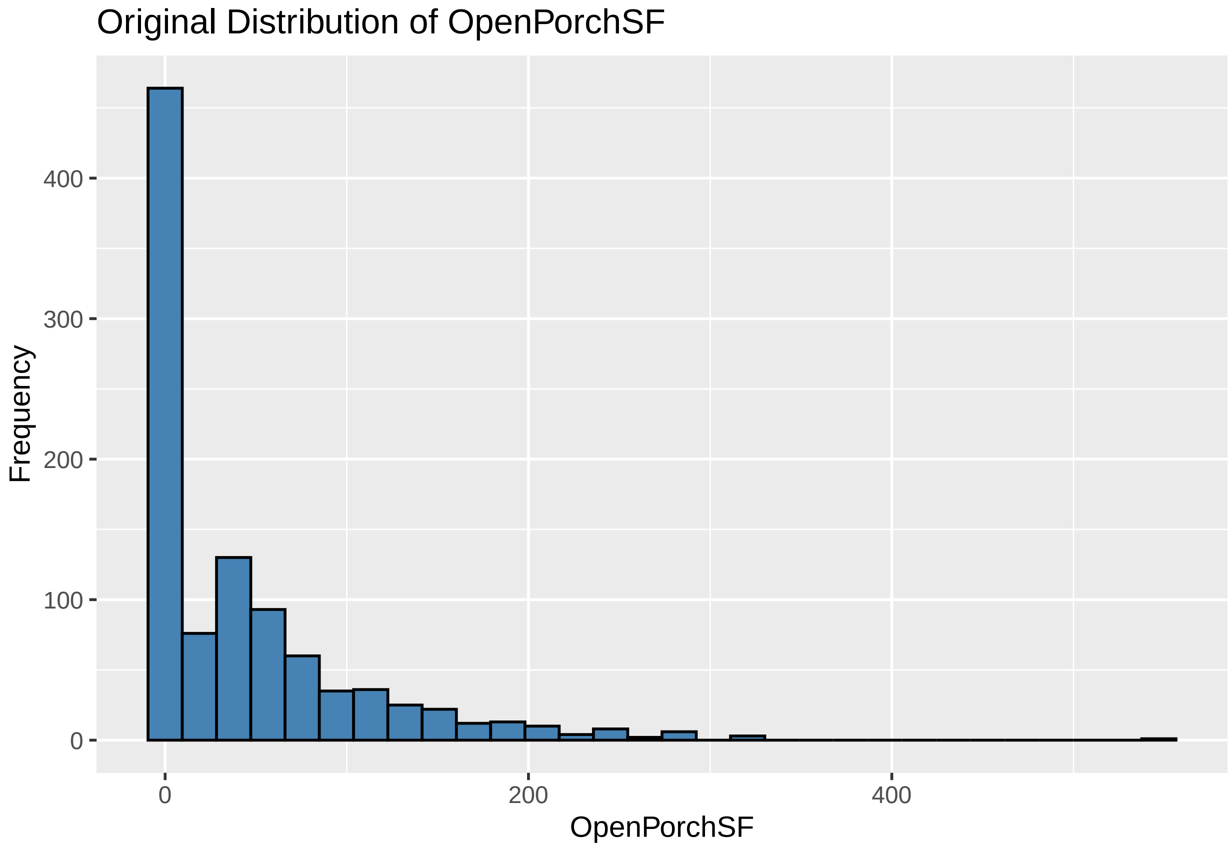
```
## Box-Cox Transformation
##
## 1000 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1      23      45      65     548
##
## Largest/Smallest: 548
## Sample Skewness: 2.16
##
## Estimated Lambda: 0
## With fudge factor, Lambda = 0 will be used for transformations
```

```r
housingNumeric$TransformedOpenPorchSF <- log(housingNumeric$OpenPorchSF)
BoxCoxTrans(housingNumeric$TransformedOpenPorchSF+1)
```

```
## Box-Cox Transformation
##
## 1000 data points used to estimate Lambda
##
## Input data summary:
```

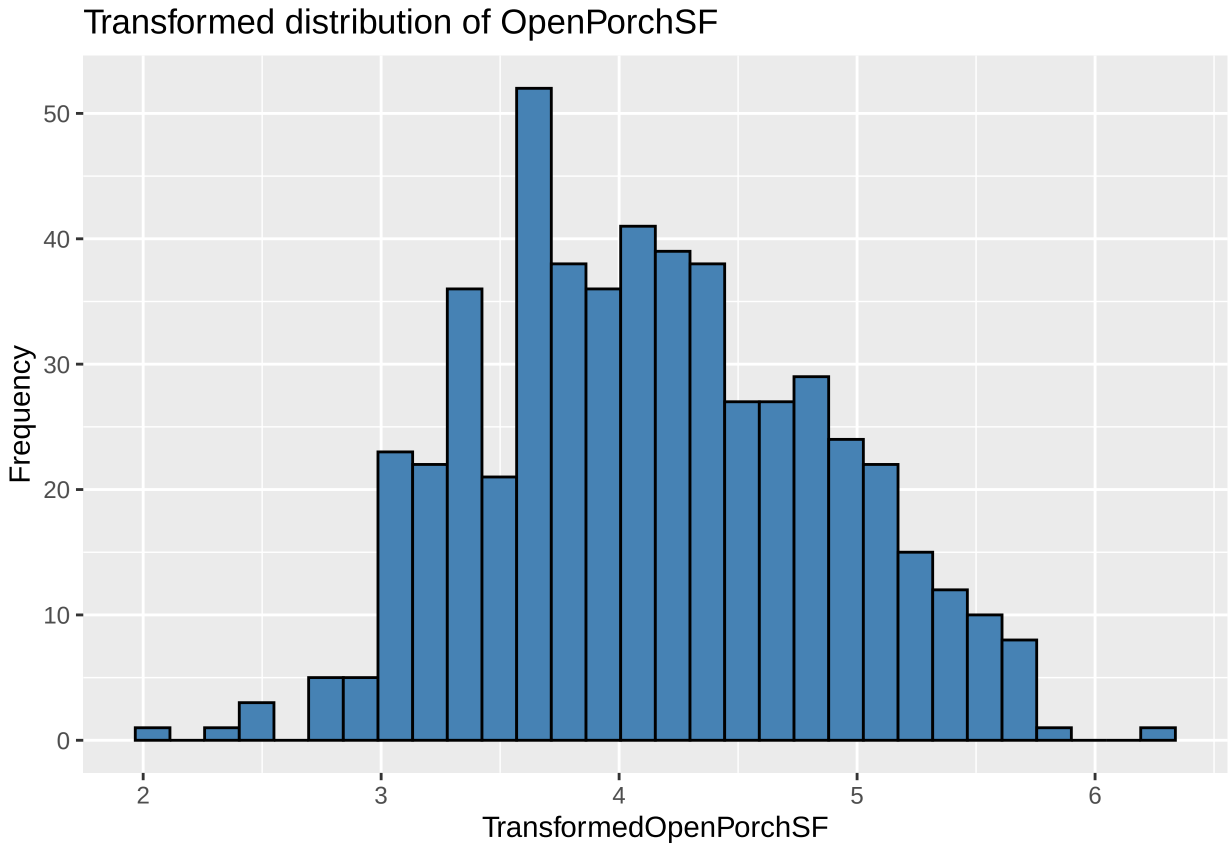
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -Inf    -Inf    4.09    -Inf    5.16    7.30
##
## Lambda could not be estimated; no transformation is applied
```

```
ggplot(housingNumeric, aes(x = OpenPorchSF)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  ggtitle("Original Distribution of OpenPorchSF") +
  xlab("OpenPorchSF") +
  ylab("Frequency")
```



```
ggplot(housingNumeric, aes(x = TransformedOpenPorchSF)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black") +
  ggtitle("Transformed distribution of OpenPorchSF") +
  xlab("TransformedOpenPorchSF") +
  ylab("Frequency")
```

```
## Warning: Removed 463 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



(b) (20 points) The variable LotFrontage has several missing values. Impute the missing values using:

i. mean value imputation
ii. regression with error
iii. predictive mean matching (Use the mice package and optionally see https://datascienceplus.com/imputing-missing-data-with-r-mice-package/ for help )
iv. For all of the above show visual depictions of how the data was transformed (e.g., histogram or density plots )

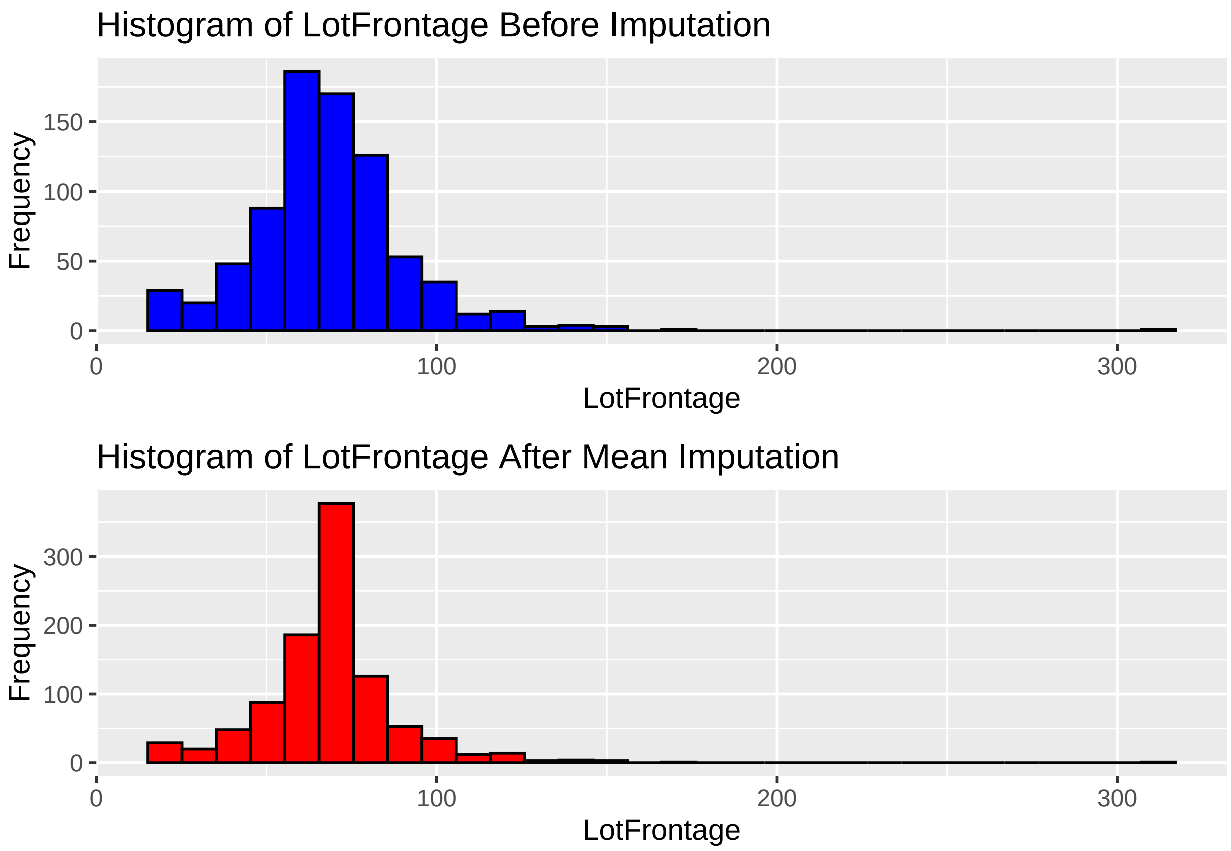
```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# Mean imputation
mean_value <- mean(housingData$LotFrontage, na.rm = TRUE)
housingData$LotFrontage_mean <- ifelse(is.na(housingData$LotFrontage), mean_value, housingData$LotFronta
# Create histogram of LotFrontage before imputation
p1 <- ggplot(housingData, aes(x = LotFrontage)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
```

```r
  ggtitle("Histogram of LotFrontage Before Imputation") +
  xlab("LotFrontage") +
  ylab("Frequency")
# Create histogram of LotFrontage after imputation
p2 <- ggplot(housingData, aes(x = LotFrontage_mean)) +
  geom_histogram(bins = 30, fill = "red", color = "black") +
  ggtitle("Histogram of LotFrontage After Mean Imputation") +
  xlab("LotFrontage") +
  ylab("Frequency")
# Arrange the plots side by side
grid.arrange(p1, p2, nrow = 2)
```

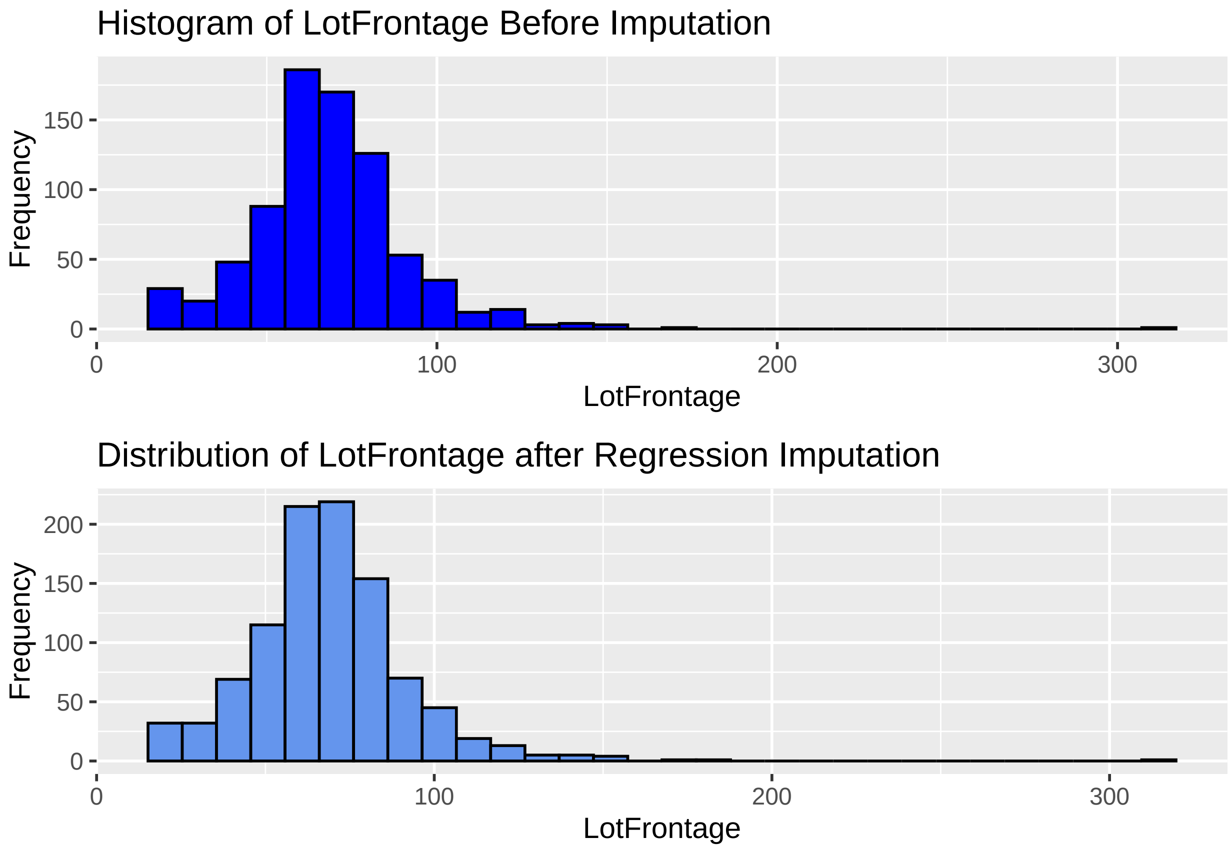
```
## Warning: Removed 207 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



```r
# Check if 'LotArea' has missing values and impute if necessary
housingData$LotArea[is.na(housingData$LotArea)] <- mean(housingData$LotArea, na.rm = TRUE)
# Fit a linear model to predict 'LotFrontage' using 'LotArea'
model <- lm(LotFrontage ~ LotArea, data = housingData, na.action = na.exclude)
# Make predictions for the full dataset
predicted_values <- predict(model, newdata = housingData)
# Calculate the residuals and their standard deviation
residuals <- resid(model)
std_error <- sd(residuals, na.rm = TRUE)
```

```
# Impute missing values in 'LotFrontage'
missing_indices <- is.na(housingData$LotFrontage)
housingData$LotFrontage_reg <- housingData$LotFrontage
# Add normally distributed noise based on the residual standard deviation
housingData$LotFrontage_reg[missing_indices] <- predicted_values[missing_indices] +
  rnorm(sum(missing_indices), mean = 0, sd = std_error)
# Visualization using a histogram to see the distribution after imputation
p3 <- ggplot(housingData, aes(x = LotFrontage_reg)) +
  geom_histogram(bins = 30, fill = "cornflowerblue", color = "black") +
  ggtitle("Distribution of LotFrontage after Regression Imputation") +
  xlab("LotFrontage") +
  ylab("Frequency")
# Arrange the plots side by side
grid.arrange(p1, p3, nrow = 2)
```

```
## Warning: Removed 207 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



```
library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
```

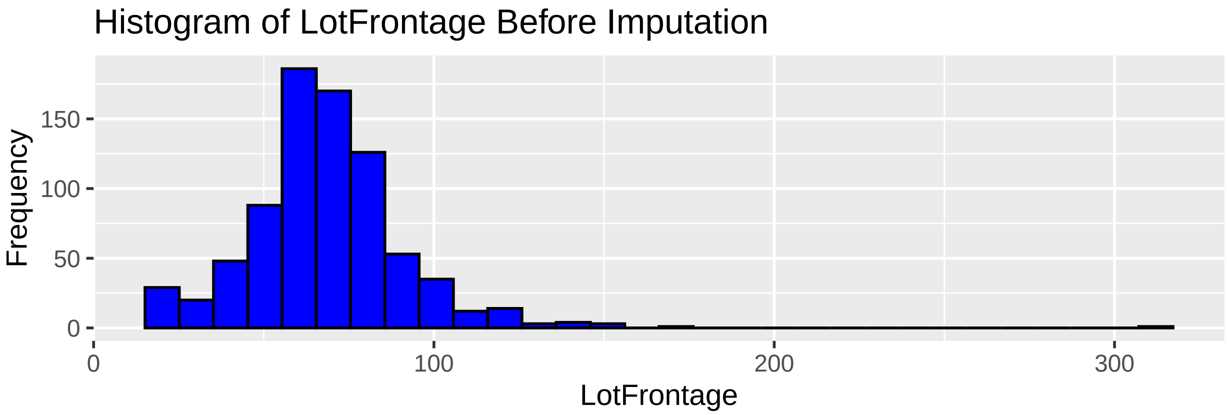
```
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

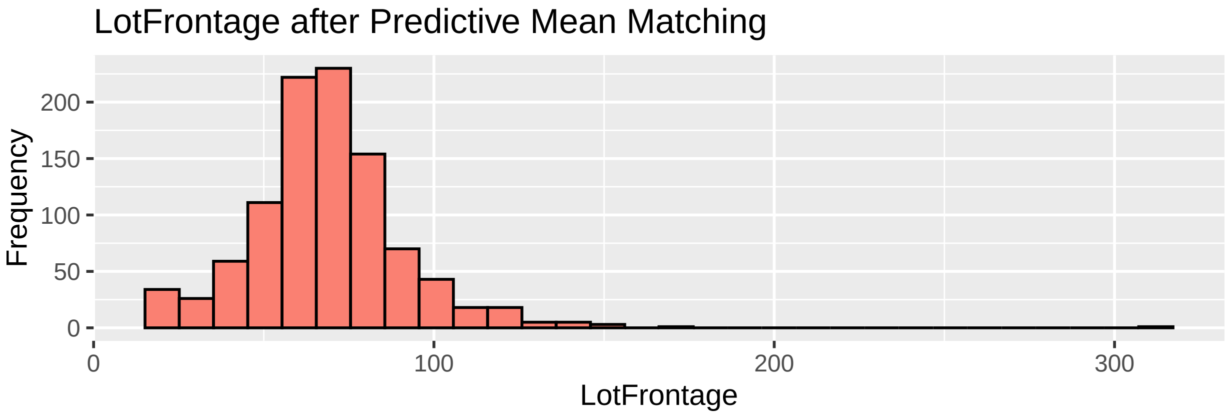
```
mice_mod <- mice(housingData[, c("LotFrontage", "LotArea")], method = 'pmm', m = 5, seed = 123)
```

```
##
##  iter imp variable
##   1   1  LotFrontage
##   1   2  LotFrontage
##   1   3  LotFrontage
##   1   4  LotFrontage
##   1   5  LotFrontage
##   2   1  LotFrontage
##   2   2  LotFrontage
##   2   3  LotFrontage
##   2   4  LotFrontage
##   2   5  LotFrontage
##   3   1  LotFrontage
##   3   2  LotFrontage
##   3   3  LotFrontage
##   3   4  LotFrontage
##   3   5  LotFrontage
##   4   1  LotFrontage
##   4   2  LotFrontage
##   4   3  LotFrontage
##   4   4  LotFrontage
##   4   5  LotFrontage
##   5   1  LotFrontage
##   5   2  LotFrontage
##   5   3  LotFrontage
##   5   4  LotFrontage
##   5   5  LotFrontage
```

```
# Perform the imputation
imputed_data <- complete(mice_mod, 1)  # We use the first completed dataset for simplicity
# Imputed Data Histogram
p4 <- ggplot(imputed_data, aes(x = LotFrontage)) +
  geom_histogram(bins = 30, fill = "salmon", color = "black") +
  ggtitle("LotFrontage after Predictive Mean Matching") +
  xlab("LotFrontage") + ylab("Frequency")
# Arrange the plots side by side
grid.arrange(p1, p4, nrow = 2)
```

```
## Warning: Removed 207 rows containing non-finite outside the scale range
## (`stat_bin()`).
```





---

(c) (10 points) Use the forcats package to do just that: Collapse the factor levels in the Exterior1st down to only five levels – the first four levels should be the most frequent levels and all other levels should be collapsed into a single "Other" level.

```
as.factor(housingData$Exterior1st)
```

```
##   [1] Plywood Wd Sdng VinylSd Wd Sdng HdBoard VinylSd HdBoard other   HdBoard
##  [10] VinylSd HdBoard VinylSd HdBoard MetalSd Wd Sdng other   HdBoard other
##  [19] Wd Sdng other   VinylSd other   VinylSd VinylSd VinylSd VinylSd Wd Sdng
##  [28] VinylSd MetalSd VinylSd Plywood HdBoard VinylSd VinylSd VinylSd VinylSd
##  [37] HdBoard CemntBd HdBoard HdBoard Plywood CemntBd HdBoard VinylSd HdBoard
##  [46] HdBoard VinylSd MetalSd VinylSd VinylSd VinylSd VinylSd Wd Sdng VinylSd
##  [55] VinylSd VinylSd MetalSd Plywood VinylSd Wd Sdng HdBoard VinylSd VinylSd
##  [64] VinylSd VinylSd MetalSd VinylSd VinylSd VinylSd BrkFace HdBoard VinylSd
##  [73] Wd Sdng HdBoard VinylSd VinylSd HdBoard MetalSd Wd Sdng other   HdBoard
##  [82] VinylSd MetalSd MetalSd VinylSd Wd Sdng VinylSd VinylSd MetalSd HdBoard
##  [91] MetalSd Wd Sdng VinylSd VinylSd Wd Sdng HdBoard MetalSd VinylSd Plywood
## [100] MetalSd MetalSd HdBoard HdBoard MetalSd Wd Sdng Wd Sdng HdBoard VinylSd
## [109] VinylSd Wd Sdng other   VinylSd MetalSd VinylSd CemntBd VinylSd VinylSd
## [118] HdBoard Wd Sdng Wd Sdng HdBoard VinylSd HdBoard Wd Sdng MetalSd HdBoard
## [127] other   Wd Sdng HdBoard VinylSd HdBoard Wd Sdng Wd Sdng BrkFace HdBoard
## [136] Plywood HdBoard Wd Sdng MetalSd HdBoard Wd Sdng HdBoard Plywood MetalSd
## [145] BrkFace VinylSd HdBoard MetalSd other   VinylSd VinylSd VinylSd Plywood
##  [154] Plywood HdBoard CemntBd MetalSd VinylSd HdBoard HdBoard VinylSd VinylSd
##  [163] HdBoard Wd Sdng VinylSd other   VinylSd HdBoard Wd Sdng BrkFace HdBoard
##  [172] Wd Sdng Plywood VinylSd CemntBd Plywood other   Wd Sdng Wd Sdng CemntBd
##  [181] VinylSd VinylSd VinylSd VinylSd HdBoard Wd Sdng CemntBd VinylSd HdBoard
##  [190] VinylSd VinylSd MetalSd Plywood VinylSd Plywood Plywood VinylSd VinylSd
##  [199] Wd Sdng MetalSd MetalSd MetalSd Wd Sdng Plywood other   VinylSd Plywood
##  [208] HdBoard VinylSd VinylSd VinylSd MetalSd Plywood CemntBd VinylSd Wd Sdng
##  [217] VinylSd MetalSd VinylSd Plywood Plywood VinylSd Wd Sdng Plywood MetalSd
##  [226] HdBoard VinylSd MetalSd VinylSd HdBoard MetalSd HdBoard Wd Sdng VinylSd
##  [235] MetalSd VinylSd HdBoard other   VinylSd VinylSd MetalSd VinylSd HdBoard
##  [244] HdBoard HdBoard BrkFace MetalSd Wd Sdng VinylSd MetalSd VinylSd VinylSd
##  [253] HdBoard CemntBd MetalSd Plywood VinylSd HdBoard VinylSd VinylSd Wd Sdng
##  [262] VinylSd VinylSd HdBoard Wd Sdng Plywood VinylSd HdBoard HdBoard Wd Sdng
##  [271] HdBoard CemntBd other   Plywood MetalSd MetalSd Wd Sdng VinylSd MetalSd
##  [280] VinylSd Plywood HdBoard MetalSd Plywood VinylSd MetalSd HdBoard BrkFace
##  [289] MetalSd VinylSd Plywood Plywood Wd Sdng HdBoard Wd Sdng Wd Sdng other  
##  [298] MetalSd VinylSd VinylSd VinylSd BrkFace HdBoard HdBoard Wd Sdng BrkFace
##  [307] HdBoard HdBoard BrkFace VinylSd Wd Sdng MetalSd VinylSd VinylSd MetalSd
##  [316] other   HdBoard other   Plywood VinylSd Wd Sdng CemntBd CemntBd BrkFace
##  [325] MetalSd HdBoard VinylSd MetalSd other   Wd Sdng other   HdBoard Plywood
##  [334] HdBoard VinylSd BrkFace Plywood VinylSd CemntBd VinylSd VinylSd MetalSd
##  [343] MetalSd other   VinylSd other   HdBoard CemntBd VinylSd VinylSd HdBoard
##  [352] CemntBd CemntBd MetalSd Plywood Wd Sdng HdBoard VinylSd Wd Sdng Wd Sdng
##  [361] BrkFace Plywood Plywood Wd Sdng MetalSd VinylSd HdBoard VinylSd MetalSd
##  [370] MetalSd HdBoard HdBoard Wd Sdng VinylSd MetalSd other   other   CemntBd
##  [379] HdBoard HdBoard VinylSd VinylSd VinylSd Wd Sdng VinylSd VinylSd MetalSd
##  [388] HdBoard HdBoard MetalSd MetalSd VinylSd VinylSd BrkFace MetalSd Plywood
##  [397] VinylSd Plywood CemntBd VinylSd Wd Sdng MetalSd Wd Sdng VinylSd MetalSd
##  [406] Plywood VinylSd Wd Sdng VinylSd Wd Sdng VinylSd MetalSd VinylSd Wd Sdng
##  [415] HdBoard VinylSd MetalSd CemntBd MetalSd Plywood MetalSd VinylSd CemntBd
##  [424] MetalSd HdBoard MetalSd HdBoard MetalSd VinylSd HdBoard Plywood VinylSd
##  [433] MetalSd Plywood MetalSd VinylSd MetalSd VinylSd MetalSd VinylSd VinylSd
##  [442] VinylSd HdBoard Wd Sdng BrkFace HdBoard VinylSd VinylSd VinylSd VinylSd
##  [451] MetalSd HdBoard HdBoard MetalSd other   Wd Sdng MetalSd VinylSd VinylSd
##  [460] VinylSd VinylSd MetalSd VinylSd CemntBd BrkFace VinylSd MetalSd VinylSd
##  [469] HdBoard MetalSd HdBoard VinylSd HdBoard other   Plywood VinylSd CemntBd
##  [478] Wd Sdng MetalSd VinylSd HdBoard BrkFace VinylSd VinylSd Wd Sdng Wd Sdng
##  [487] MetalSd Wd Sdng VinylSd VinylSd Plywood VinylSd MetalSd MetalSd Wd Sdng
##  [496] VinylSd VinylSd HdBoard Plywood BrkFace Wd Sdng VinylSd Wd Sdng Plywood
##  [505] VinylSd other   other   VinylSd HdBoard VinylSd Plywood MetalSd Plywood
##  [514] HdBoard Plywood VinylSd Wd Sdng VinylSd VinylSd VinylSd VinylSd VinylSd
##  [523] VinylSd VinylSd VinylSd BrkFace Plywood HdBoard MetalSd Wd Sdng VinylSd
##  [532] VinylSd VinylSd CemntBd BrkFace VinylSd Plywood other   Wd Sdng Wd Sdng
##  [541] HdBoard BrkFace Plywood MetalSd VinylSd HdBoard other   Wd Sdng VinylSd
##  [550] VinylSd HdBoard CemntBd VinylSd VinylSd HdBoard HdBoard MetalSd VinylSd
##  [559] HdBoard Plywood HdBoard VinylSd HdBoard VinylSd HdBoard MetalSd MetalSd
##  [568] MetalSd MetalSd VinylSd HdBoard MetalSd Wd Sdng HdBoard VinylSd HdBoard
##  [577] Plywood MetalSd Wd Sdng HdBoard MetalSd VinylSd VinylSd other   VinylSd
##  [586] VinylSd VinylSd MetalSd Plywood VinylSd VinylSd VinylSd VinylSd VinylSd
##  [595] VinylSd VinylSd MetalSd BrkFace Wd Sdng MetalSd MetalSd VinylSd CemntBd
##  [604] HdBoard HdBoard CemntBd MetalSd HdBoard VinylSd MetalSd other   VinylSd
##  [613] Wd Sdng Wd Sdng MetalSd BrkFace Wd Sdng HdBoard Plywood other   VinylSd
##  [622] MetalSd Plywood MetalSd VinylSd VinylSd HdBoard VinylSd other   VinylSd
##  [631] MetalSd VinylSd VinylSd CemntBd MetalSd VinylSd VinylSd VinylSd BrkFace
##  [640] HdBoard VinylSd Wd Sdng HdBoard VinylSd VinylSd VinylSd VinylSd HdBoard
##  [649] HdBoard other   VinylSd VinylSd MetalSd Wd Sdng HdBoard MetalSd VinylSd
##  [658] VinylSd VinylSd VinylSd Wd Sdng VinylSd VinylSd other   MetalSd Wd Sdng
##  [667] VinylSd VinylSd CemntBd VinylSd VinylSd VinylSd BrkFace VinylSd VinylSd
##  [676] HdBoard Plywood CemntBd Plywood HdBoard Wd Sdng VinylSd VinylSd HdBoard
##  [685] MetalSd VinylSd VinylSd VinylSd VinylSd HdBoard Wd Sdng Wd Sdng VinylSd
##  [694] VinylSd Plywood VinylSd VinylSd VinylSd HdBoard MetalSd HdBoard VinylSd
##  [703] Wd Sdng HdBoard MetalSd Plywood MetalSd Wd Sdng MetalSd VinylSd HdBoard
##  [712] BrkFace Plywood MetalSd MetalSd HdBoard HdBoard BrkFace CemntBd Wd Sdng
##  [721] Wd Sdng Wd Sdng Wd Sdng Wd Sdng VinylSd HdBoard VinylSd VinylSd Wd Sdng
##  [730] BrkFace MetalSd MetalSd MetalSd Wd Sdng MetalSd HdBoard HdBoard Wd Sdng
##  [739] HdBoard HdBoard Plywood VinylSd VinylSd BrkFace Plywood VinylSd Wd Sdng
##  [748] MetalSd CemntBd BrkFace VinylSd VinylSd MetalSd other   MetalSd BrkFace
##  [757] Wd Sdng other   VinylSd other   Wd Sdng BrkFace Wd Sdng MetalSd MetalSd
##  [766] HdBoard VinylSd VinylSd Wd Sdng VinylSd HdBoard Wd Sdng HdBoard MetalSd
##  [775] Wd Sdng VinylSd VinylSd HdBoard Wd Sdng MetalSd Wd Sdng Wd Sdng VinylSd
##  [784] Wd Sdng VinylSd Wd Sdng HdBoard Wd Sdng VinylSd Wd Sdng Wd Sdng HdBoard
##  [793] VinylSd VinylSd VinylSd VinylSd VinylSd VinylSd MetalSd MetalSd MetalSd
##  [802] other   VinylSd VinylSd VinylSd other   Wd Sdng Plywood Wd Sdng Plywood
##  [811] other   HdBoard VinylSd VinylSd MetalSd Wd Sdng Plywood BrkFace VinylSd
##  [820] Wd Sdng VinylSd Wd Sdng VinylSd Wd Sdng MetalSd VinylSd HdBoard VinylSd
##  [829] Wd Sdng VinylSd HdBoard VinylSd Plywood MetalSd Wd Sdng VinylSd VinylSd
##  [838] HdBoard HdBoard VinylSd VinylSd VinylSd BrkFace Wd Sdng other   HdBoard
##  [847] MetalSd Wd Sdng MetalSd HdBoard VinylSd Wd Sdng other   HdBoard VinylSd
##  [856] other   HdBoard HdBoard HdBoard BrkFace Plywood MetalSd CemntBd Plywood
##  [865] MetalSd HdBoard VinylSd HdBoard Wd Sdng VinylSd BrkFace VinylSd BrkFace
##  [874] VinylSd VinylSd MetalSd VinylSd VinylSd VinylSd MetalSd HdBoard BrkFace
##  [883] other   HdBoard other   VinylSd VinylSd HdBoard MetalSd MetalSd VinylSd
##  [892] HdBoard HdBoard MetalSd MetalSd Plywood MetalSd Wd Sdng HdBoard VinylSd
##  [901] VinylSd VinylSd other   other   VinylSd Wd Sdng VinylSd Wd Sdng Wd Sdng
##  [910] MetalSd Wd Sdng VinylSd BrkFace VinylSd MetalSd VinylSd VinylSd VinylSd
##  [919] other   BrkFace Plywood HdBoard MetalSd Wd Sdng VinylSd Wd Sdng Plywood
##  [928] VinylSd HdBoard Wd Sdng Wd Sdng Wd Sdng other   HdBoard HdBoard VinylSd
##  [937] MetalSd HdBoard other   Plywood VinylSd VinylSd VinylSd BrkFace CemntBd
##  [946] VinylSd HdBoard Wd Sdng MetalSd Wd Sdng VinylSd HdBoard Wd Sdng MetalSd
##  [955] CemntBd MetalSd VinylSd VinylSd BrkFace HdBoard CemntBd VinylSd HdBoard
##  [964] HdBoard VinylSd MetalSd Wd Sdng MetalSd CemntBd VinylSd MetalSd Plywood
##  [973] HdBoard MetalSd HdBoard MetalSd MetalSd VinylSd HdBoard VinylSd MetalSd
##  [982] VinylSd VinylSd VinylSd Wd Sdng BrkFace HdBoard VinylSd HdBoard Wd Sdng
##  [991] Wd Sdng HdBoard HdBoard other   HdBoard Wd Sdng Plywood Wd Sdng HdBoard
## [1000] Wd Sdng
## Levels: BrkFace CemntBd HdBoard MetalSd other Plywood VinylSd Wd Sdng
```

```
fct_count(housingData$Exterior1st, sort = T)
```

```
## # A tibble: 8 x 2
##   f           n
##   <fct>   <int>
## 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
```

```
fct_unique(housingData$Exterior1st)
```

```
## [1] BrkFace CemntBd HdBoard MetalSd other   Plywood VinylSd Wd Sdng
## Levels: BrkFace CemntBd HdBoard MetalSd other Plywood VinylSd Wd Sdng
```

```
# Collapse factor levels
housingData$Exterior1st <- fct_lump_n(housingData$Exterior1st, n = 4)
# Check the changes
table(housingData$Exterior1st)
```

```
##
## HdBoard MetalSd VinylSd Wd Sdng   Other
##     175     153     328     141     203
```

---

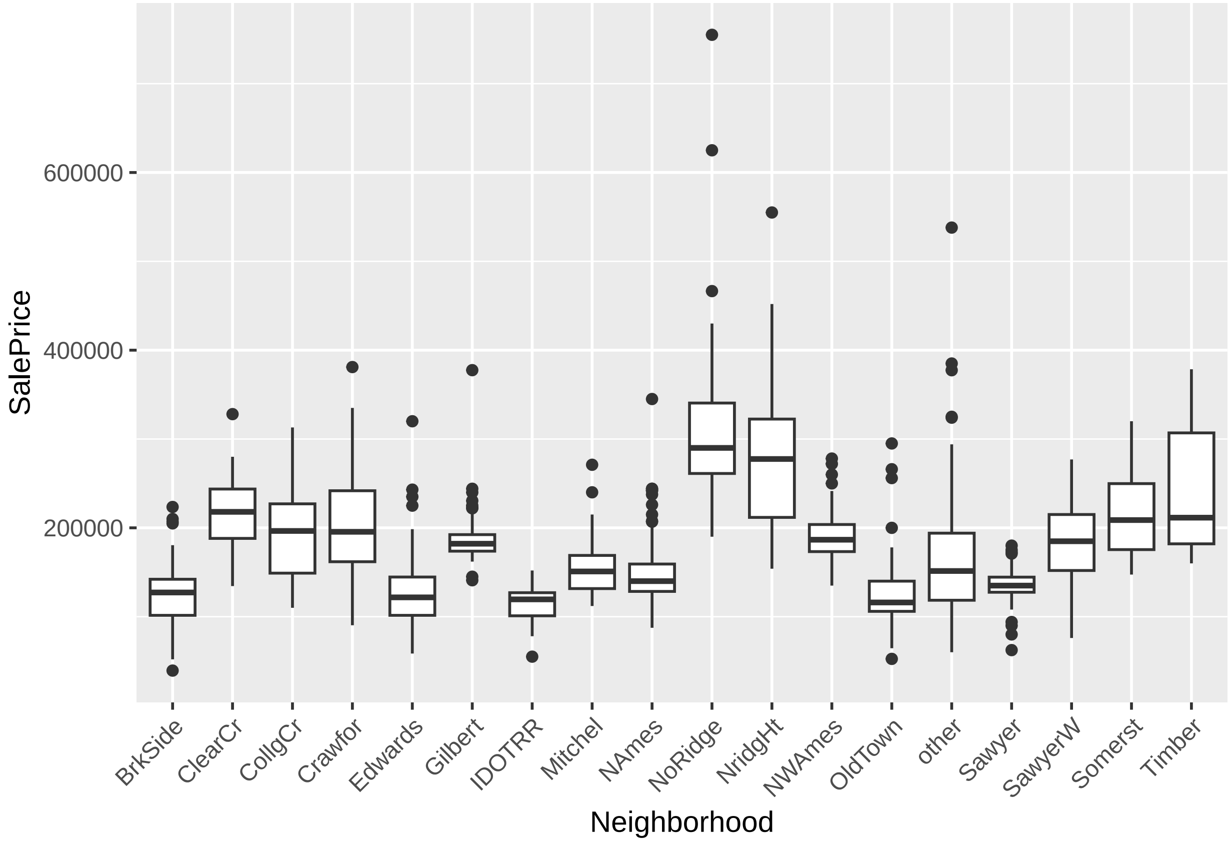
(d) (16 points) More fun with factors

i. Use tidyverse packages to compute the average SalePrice for each Neighborhood factor level.
ii. Create a parallel boxplot chart of this data, i.e., a boxplot associated with the sale prices for homes in each of the 18 neighborhoods.
iii. You should notice that there is a lot of variation in price by neighborhood. Using forcats re-order the factor levels of the Neighborhood variable in descending order of the median price per neighborhood (i.e., the neighborhood with the highest median price is NoRidge, the next highest median is NridgHt, etc., so NoRidge should be the first level and NridgHt should be the second factor level, etc.)
iv. If you have done re-ordering correctly, you should be able to produce a parallel boxplot of neighborhoods and sales prices in descending order (see Figure 5). Note: R orders values in graphs according to the ordering of the factors.

```
# Average SalePrice for each Neighborhood factor level
average_prices <- housingData %>%
  group_by(Neighborhood) %>%
  summarise(AverageSalePrice = mean(SalePrice, na.rm = TRUE))
print(average_prices)
```

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

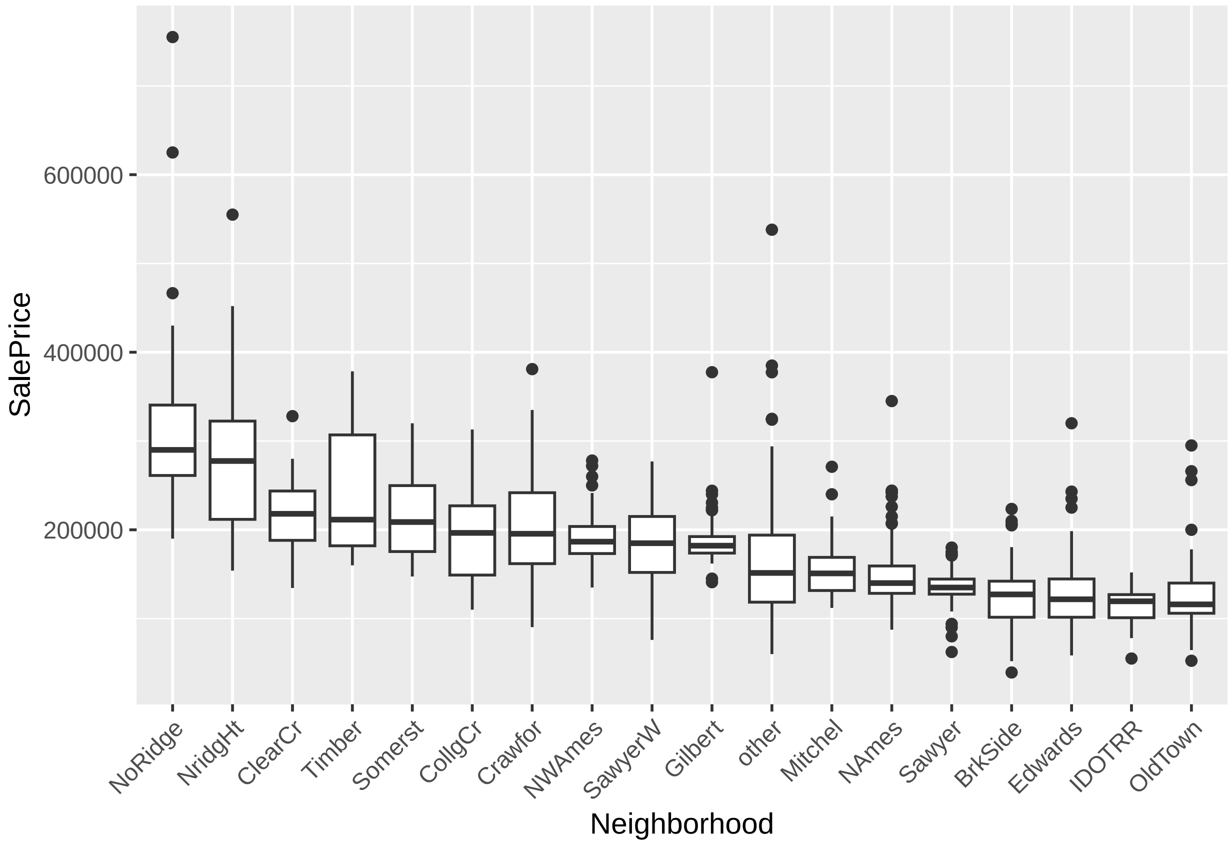
```
# Parallel Boxplots before Sorting
housingData %>%
  ggplot(aes(x = Neighborhood, y = SalePrice)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x labels for better visibility
```



```
housingData <- housingData %>%
  mutate(Neighborhood = fct_reorder(Neighborhood, SalePrice, median, .desc = TRUE))
# Verify the new order
housingData %>%
  group_by(Neighborhood) %>%
  summarise(MedianSalePrice = median(SalePrice, na.rm = TRUE))
```

```
## # A tibble: 18 x 2
##    Neighborhood MedianSalePrice
##    <fct>                  <dbl>
##   1 NoRidge               290000
##   2 NridgHt               277500
##   3 ClearCr               218000
##   4 Timber                211450
##   5 Somerst               208750
##   6 CollgCr               196500
##   7 Crawfor               195550
##   8 NWAmes                186625
##   9 SawyerW               184900
##  10 Gilbert               182100
##  11 other                 151400
##  12 Mitchel               150900
##  13 NAmes                 140000
##  14 Sawyer                135000
##  15 BrkSide               127250
##  16 Edwards               121750
##  17 IDOTRR                119500
##  18 OldTown               116000
```

```
# Parallel Boxplots after Sorting
housingData %>%
  ggplot(aes(x = Neighborhood, y = SalePrice)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x labels for better visibility
```


SalePrice
NoRidge
NridgHt
ClearCr
Timber
Somerst
CollgCr
Crawfor
NWAmes
SawyerW
Gilbert
other
Mitchel
NAmes
Sawyer
BrkSide
Edwards
IDOTRR
OldTown
Neighborhood