Project\_DataCleaning

Thibeaux

2023-06-30

# Data Upload

Full2016 <- read.csv(file.choose(), header = TRUE, fill = TRUE)  
Full2017 <- read.csv(file.choose(), header = TRUE, fill = TRUE)

# Data Cleaning

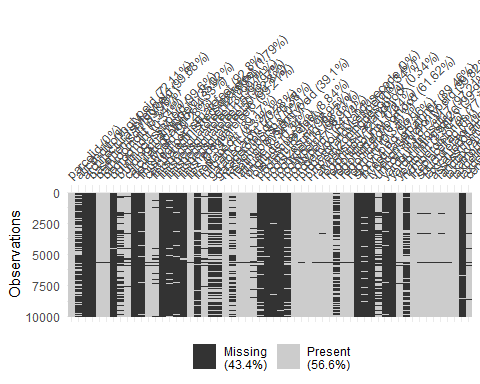
## Take Samples

sample\_size = 10000  
set.seed(1)  
  
# 2016 File  
idxs2016 = sample(1:nrow(Full2016),sample\_size,replace=F)  
subsample2016 = Full2016[idxs2016,]  
  
# 2017 File  
idxs2017 = sample(1:nrow(Full2017),sample\_size,replace=F)  
subsample2017 = Full2017[idxs2017,]

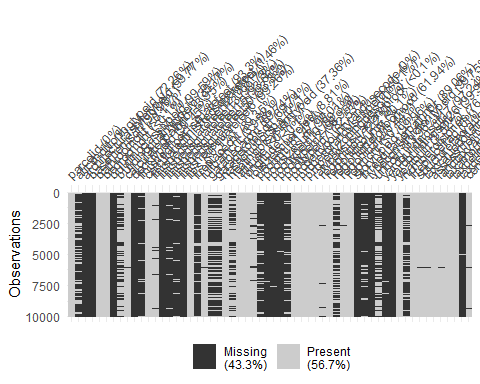
## Checking location of NAs

# Regular NAs  
vis\_miss(subsample2016)

## Warning: `gather\_()` was deprecated in tidyr 1.2.0.  
## ℹ Please use `gather()` instead.  
## ℹ The deprecated feature was likely used in the visdat package.  
## Please report the issue at <]8;;https://github.com/ropensci/visdat/issueshttps://github.com/ropensci/visdat/issues]8;;>.



vis\_miss(subsample2017)



# Closer look at data that may have blank values  
  
# Column 23  
subsample2016$hashottuborspa <- as.factor(subsample2016$hashottuborspa)  
summary(subsample2016$hashottuborspa)

## true   
## 9761 239

# 98% of these values are blank - since I'm not sure if that means "false" or just missing data, this column would not be helpful.  
  
# Column 33  
subsample2016$propertycountylandusecode <- as.factor(subsample2016$propertycountylandusecode)  
summary(subsample2016$propertycountylandusecode)

## 0 0100 0101 0103 0104 0108 0109 010C 010D 010E 010F 010G 010H 010M 010V   
## 37 24 3884 787 8 53 2 18 816 238 151 8 6 2 3 9   
## 0110 0118 012C 0131 01DC 01HC 0200 0201 020E 0300 0301 0304 030V 0400 0401 040D   
## 3 1 32 2 14 9 333 5 1 109 2 1 3 101 1 1   
## 040V 0700 070P 1 100V 1014 10C 1110 1111 1112 1117 1118 1128 1129 1200 1201   
## 1 6 73 185 3 1 1 141 434 2 5 37 31 126 8 1   
## 1202 120C 1210 122 1222 128 1310 1321 135 1410 1421 1432 34 38 71 73   
## 1 1 40 1736 5 2 1 2 10 1 2 1 399 14 15 27   
## 96   
## 24

# There are less than 1% blank, so we'll keep this as a categorical variable.  
  
# Column 33 Change to Factor  
Full2016$propertycountylandusecode <- as.factor(Full2016$propertycountylandusecode)  
  
Full2017$propertycountylandusecode <- as.factor(Full2017$propertycountylandusecode)  
  
# Column 35  
subsample2016$propertyzoningdesc <- as.factor(subsample2016$propertyzoningdesc)  
summary(subsample2016$propertyzoningdesc)

## LAR1 LAR3 LBR1N LAR2 LARS LARD1.5   
## 3356 918 211 175 169 164 145   
## LARD2 SCUR2 TORR-LO LARA LCR1YY LARE11 LARE15   
## 120 105 98 79 72 60 58   
## CARS\* LKR1YY LCR1\* PDR1\* LAR4 PRSF\* LARD3   
## 57 50 48 46 45 45 44   
## NOR1YY LAC2 LCA106 BPR1\* BUR1YY PSR6 SCUR3   
## 44 43 39 38 38 37 37   
## ALRPD\* GLR1YY LBR2N LCA11\* LCR2\* CORL\* LKR1\*   
## 34 34 34 34 33 30 30   
## CERS5000 LCA16000\* LCR175 BUR1\* LCR3\* GAR1 LCA25\*   
## 29 29 28 27 27 26 26   
## LARE9 MNRS SCSP DOR105 SMR2\* PSR4 LCRA06   
## 25 25 25 24 24 23 22   
## LCRA7000\* NOR1\* SCUR1 MNR1YY PSR1 WCR1YY LCA1\*   
## 22 22 22 21 21 21 20   
## GDR1 LARE40 LCA21\* LCA22\* POR17200\* SGR1YY ARR1YY   
## 19 19 19 19 19 19 18   
## LAR5 LCR2YY LMR1\* GLR4YY MPR1YY BFR1\* BUR4\*   
## 18 18 18 17 17 16 16   
## GLR4\* HAR1YY INR1YY LCR106 LCR16000\* RPRS10000\* SMR1\*   
## 16 16 16 16 16 16 16   
## WCPCD1\* WDR4\* ALR1YY DUR16500\* LCR17500\* LNR2YY RBR-1   
## 16 16 15 15 15 15 15   
## SLR1YY GLR1\* LBR1S LBR3S LCR110000\* MPR1\* SGR3YY   
## 15 14 14 14 14 14 14   
## WHR1YY DOR15000\* INR2YY LARD6 TCR172 WCR1\* CLRS10000\*   
## 14 13 13 13 13 13 12   
## EMR3\* (Other)   
## 12 2248

# Half the values in the sample are either blank or (other), so don't think this is helpful.  
  
# Column 50  
subsample2016$fireplaceflag <- as.factor(subsample2016$fireplaceflag)  
summary(subsample2016$fireplaceflag)

## true   
## 9984 16

# 99% of these values are blank - since I'm not sure if that means "false" or just missing data, this column would not be helpful.  
  
# Column 56  
subsample2016$taxdelinquencyflag <- as.factor(subsample2016$taxdelinquencyflag)  
summary(subsample2016$taxdelinquencyflag)

## Y   
## 9797 203

# 98% of these are "other" so this column is not helpful either  
  
# Column 58  
subsample2016$censustractandblock <- as.factor(subsample2016$censustractandblock)  
summary(subsample2016$censustractandblock)

## 60372622001000 60379201081015 60379203312000 60590423244002 61110058021001   
## 6 5 5 5 5   
## 60371393031002 60371395041000 60372360002001 60372679012002 60372754002003   
## 4 4 4 4 4   
## 60372964011008 60374086253009 60375719001010 60376213012001 60377025023000   
## 4 4 4 4 4   
## 60378005041029 60379006092006 60379200431012 60590015062004 60590320471000   
## 4 4 4 4 4   
## 60590320481000 60590320571003 60590423121003 60590524203033 60590626411030   
## 4 4 4 4 4   
## 60590881061004 60590881062002 60590885011006 61110036051026 61110059012007   
## 4 4 4 4 4   
## 61110074033002 60371112052003 60371132111023 60371154031010 60371272202008   
## 4 3 3 3 3   
## 60371392003001 60371417002000 60371436031017 60371437001018 60371862022001   
## 3 3 3 3 3   
## 60371897021004 60371990002024 60372013021000 60372015013005 60372073012008   
## 3 3 3 3 3   
## 60372079002009 60372753022013 60372766032001 60372933011003 60372933011004   
## 3 3 3 3 3   
## 60374004031006 60374033162002 60374033171001 60374049011009 60374063001014   
## 3 3 3 3 3   
## 60374078011000 60374300032001 60374307212004 60374325002001 60374828001032   
## 3 3 3 3 3   
## 60375507002003 60375548022002 60376007031014 60376010023006 60376203033001   
## 3 3 3 3 3   
## 60376508003032 60376513022000 60376703243018 60377003004003 60378003323003   
## 3 3 3 3 3   
## 60378005041038 60379102073011 60379200281007 60379200312000 60379200351012   
## 3 3 3 3 3   
## 60379200381000 60379201081000 60379203302001 60379203311002 60379203381016   
## 3 3 3 3 3   
## 60590218163014 60590219203000 60590219231005 60590320342000 60590320343001   
## 3 3 3 3 3   
## 60590320482001 60590320494000 60590320571024 60590320582009 60590320592003   
## 3 3 3 3 3   
## 60590421081000 60590423204019 60590423232009 60590423322003 60590524183002   
## 3 3 3 3 3   
## 60590524204010 60590524241007 60590524253000 (Other) NA's   
## 3 3 3 9424 245

# 94% of these are "other" so this column is not helpful either

## Select columns

These columns do not have more than 9% NAs or blanks: 1, 5, 6, 9, 12:13, 18, 20, 25:27, 33:34, 36:38, 40:41, 48, 51:55

subset2016 <- select(Full2016, 1, 5, 6, 9, 12:13, 18, 20, 25:27, 33:34, 36:38, 40:41, 48, 51:55)  
subset2017 <- select(Full2017, 1, 5, 6, 9, 12:13, 18, 20, 25:27, 33:34, 36:38, 40:41, 48, 51:55)  
  
summary(subset2016)

## parcelid bathroomcnt bedroomcnt calculatedbathnbr  
## Min. : 10711725 Min. : 0.000 Min. : 0.000 Min. : 1.0   
## 1st Qu.: 11643707 1st Qu.: 2.000 1st Qu.: 2.000 1st Qu.: 2.0   
## Median : 12545094 Median : 2.000 Median : 3.000 Median : 2.0   
## Mean : 13325858 Mean : 2.209 Mean : 3.089 Mean : 2.3   
## 3rd Qu.: 14097122 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.: 3.0   
## Max. :169601949 Max. :20.000 Max. :20.000 Max. :20.0   
## NA's :11462 NA's :11450 NA's :128912   
## calculatedfinishedsquarefeet finishedsquarefeet12 fips   
## Min. : 1 Min. : 1 Min. :6037   
## 1st Qu.: 1213 1st Qu.: 1196 1st Qu.:6037   
## Median : 1572 Median : 1539 Median :6037   
## Mean : 1827 Mean : 1760 Mean :6048   
## 3rd Qu.: 2136 3rd Qu.: 2070 3rd Qu.:6059   
## Max. :952576 Max. :290345 Max. :6111   
## NA's :55565 NA's :276033 NA's :11437   
## fullbathcnt latitude longitude lotsizesquarefeet   
## Min. : 1.00 Min. :33324388 Min. :-119475780 Min. : 100   
## 1st Qu.: 2.00 1st Qu.:33827685 1st Qu.:-118392983 1st Qu.: 5688   
## Median : 2.00 Median :34008249 Median :-118172540 Median : 7000   
## Mean : 2.24 Mean :34001469 Mean :-118201934 Mean : 22823   
## 3rd Qu.: 3.00 3rd Qu.:34161860 3rd Qu.:-117949468 3rd Qu.: 9898   
## Max. :20.00 Max. :34819650 Max. :-117554316 Max. :328263808   
## NA's :128912 NA's :11437 NA's :11437 NA's :276099   
## propertycountylandusecode propertylandusetypeid rawcensustractandblock  
## 0100 :1153896 Min. : 31 Min. :60371011   
## 122 : 522145 1st Qu.:261 1st Qu.:60373203   
## 0101 : 247494 Median :261 Median :60375712   
## 010C : 225410 Mean :260 Mean :60483450   
## 1111 : 126491 3rd Qu.:261 3rd Qu.:60590423   
## 34 : 123249 Max. :275 Max. :61110091   
## (Other): 586532 NA's :11437 NA's :11437   
## regionidcity regionidcounty regionidzip roomcnt   
## Min. : 3491 Min. :1286 Min. : 95982 Min. : 0.000   
## 1st Qu.: 12447 1st Qu.:2061 1st Qu.: 96180 1st Qu.: 0.000   
## Median : 25218 Median :3101 Median : 96377 Median : 0.000   
## Mean : 34993 Mean :2570 Mean : 96553 Mean : 1.475   
## 3rd Qu.: 45457 3rd Qu.:3101 3rd Qu.: 96974 3rd Qu.: 0.000   
## Max. :396556 Max. :3101 Max. :399675 Max. :96.000   
## NA's :62845 NA's :11437 NA's :13980 NA's :11475   
## yearbuilt structuretaxvaluedollarcnt taxvaluedollarcnt assessmentyear   
## Min. :1801 Min. : 1 Min. : 1 Min. :2000   
## 1st Qu.:1950 1st Qu.: 74800 1st Qu.: 179675 1st Qu.:2015   
## Median :1963 Median : 122590 Median : 306086 Median :2015   
## Mean :1964 Mean : 170884 Mean : 420479 Mean :2015   
## 3rd Qu.:1981 3rd Qu.: 196889 3rd Qu.: 488000 3rd Qu.:2015   
## Max. :2015 Max. :251486000 Max. :282786000 Max. :2016   
## NA's :59928 NA's :54982 NA's :42550 NA's :11439   
## landtaxvaluedollarcnt taxamount   
## Min. : 1 Min. : 1   
## 1st Qu.: 74836 1st Qu.: 2461   
## Median : 167042 Median : 3992   
## Mean : 252478 Mean : 5378   
## 3rd Qu.: 306918 3rd Qu.: 6201   
## Max. :90246219 Max. :3458861   
## NA's :67733 NA's :31250

NoNA2016 <- na.omit(subset2016)  
  
summary(subset2017)

## parcelid bathroomcnt bedroomcnt calculatedbathnbr  
## Min. : 10711725 Min. : 0.000 Min. : 0.000 Min. : 1.0   
## 1st Qu.: 11643707 1st Qu.: 2.000 1st Qu.: 2.000 1st Qu.: 2.0   
## Median : 12545094 Median : 2.000 Median : 3.000 Median : 2.0   
## Mean : 13325858 Mean : 2.216 Mean : 3.093 Mean : 2.3   
## 3rd Qu.: 14097122 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.: 3.0   
## Max. :169601949 Max. :32.000 Max. :25.000 Max. :32.0   
## NA's :2957 NA's :2945 NA's :117156   
## calculatedfinishedsquarefeet finishedsquarefeet12 fips   
## Min. : 1 Min. : 1 Min. :6037   
## 1st Qu.: 1215 1st Qu.: 1198 1st Qu.:6037   
## Median : 1574 Median : 1542 Median :6037   
## Mean : 1832 Mean : 1764 Mean :6048   
## 3rd Qu.: 2140 3rd Qu.: 2075 3rd Qu.:6059   
## Max. :952576 Max. :427079 Max. :6111   
## NA's :45097 NA's :264431 NA's :2932   
## fullbathcnt latitude longitude lotsizesquarefeet   
## Min. : 1.00 Min. :33324388 Min. :-119475780 Min. : 100   
## 1st Qu.: 2.00 1st Qu.:33827400 1st Qu.:-118392899 1st Qu.: 5683   
## Median : 2.00 Median :34008074 Median :-118172091 Median : 7000   
## Mean : 2.25 Mean :34001306 Mean :-118201663 Mean : 22604   
## 3rd Qu.: 3.00 3rd Qu.:34161856 3rd Qu.:-117948920 3rd Qu.: 9893   
## Max. :32.00 Max. :34819650 Max. :-117554316 Max. :371000512   
## NA's :117156 NA's :2932 NA's :2932 NA's :272706   
## propertycountylandusecode propertylandusetypeid rawcensustractandblock  
## 0100 :1153332 Min. : 31.0 Min. :60371011   
## 122 : 522127 1st Qu.:261.0 1st Qu.:60374002   
## 0101 : 248386 Median :261.0 Median :60375713   
## 010C : 225950 Mean :260.1 Mean :60483665   
## 1111 : 127012 3rd Qu.:261.0 3rd Qu.:60590423   
## 34 : 123249 Max. :279.0 Max. :61110091   
## (Other): 585161 NA's :2932 NA's :2932   
## regionidcity regionidcounty regionidzip roomcnt   
## Min. : 3491 Min. :1286 Min. : 95982 Min. : 0.000   
## 1st Qu.: 12447 1st Qu.:1286 1st Qu.: 96180 1st Qu.: 0.000   
## Median : 25218 Median :3101 Median : 96377 Median : 0.000   
## Mean : 34988 Mean :2569 Mean : 96553 Mean : 1.474   
## 3rd Qu.: 45457 3rd Qu.:3101 3rd Qu.: 96974 3rd Qu.: 0.000   
## Max. :396556 Max. :3101 Max. :399675 Max. :96.000   
## NA's :62128 NA's :2932 NA's :12714 NA's :2969   
## yearbuilt structuretaxvaluedollarcnt taxvaluedollarcnt assessmentyear  
## Min. :1801 Min. : 1 Min. : 1 Min. :2000   
## 1st Qu.:1950 1st Qu.: 77666 1st Qu.: 188220 1st Qu.:2016   
## Median :1963 Median : 127066 Median : 321161 Median :2016   
## Mean :1964 Mean : 178143 Mean : 443528 Mean :2016   
## 3rd Qu.:1981 3rd Qu.: 204000 3rd Qu.: 514072 3rd Qu.:2016   
## Max. :2016 Max. :255321161 Max. :319622473 Max. :2016   
## NA's :47833 NA's :46464 NA's :34266 NA's :2933   
## landtaxvaluedollarcnt taxamount   
## Min. : 1 Min. : 0   
## 1st Qu.: 79700 1st Qu.: 2469   
## Median : 176619 Median : 4008   
## Mean : 268456 Mean : 5409   
## 3rd Qu.: 326100 3rd Qu.: 6230   
## Max. :94011079 Max. :3823176   
## NA's :59926 NA's :22752

NoNA2017 <- na.omit(subset2017)

## Compare samples from different seeds

Method from: <https://www.kdnuggets.com/2019/05/sample-huge-dataset-machine-learning.html>

This article explains how choosing a small, representative dataset from a large population can improve model training reliability.

# Seed 1 for 2016 data  
sample\_size = 10000  
set.seed(1)  
idxs = sample(1:nrow(NoNA2016),sample\_size,replace=F)  
subsample = NoNA2016[idxs,]  
pvalues = list()  
for (col in names(NoNA2016)) {  
 if (class(NoNA2016[,col]) %in% c("numeric","integer")) {  
 # Numeric variable. Using Kolmogorov-Smirnov test  
   
 pvalues[[col]] = ks.test(subsample[[col]],NoNA2016[[col]])$p.value  
   
 } else {  
 # Categorical variable. Using Pearson's Chi-square test  
   
 probs = table(NoNA2016[[col]])/nrow(NoNA2016)  
 pvalues[[col]] = chisq.test(table(subsample[[col]]),p=probs)$p.value  
   
 }  
}

## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties

## Warning in chisq.test(table(subsample[[col]]), p = probs): Chi-squared  
## approximation may be incorrect

## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties

pvalues

## $parcelid  
## [1] 0.8398666  
##   
## $bathroomcnt  
## [1] 0.9656554  
##   
## $bedroomcnt  
## [1] 0.9970891  
##   
## $calculatedbathnbr  
## [1] 0.9656554  
##   
## $calculatedfinishedsquarefeet  
## [1] 0.9990852  
##   
## $finishedsquarefeet12  
## [1] 0.9990852  
##   
## $fips  
## [1] 0.9999797  
##   
## $fullbathcnt  
## [1] 0.9934855  
##   
## $latitude  
## [1] 0.8654928  
##   
## $longitude  
## [1] 0.4719448  
##   
## $lotsizesquarefeet  
## [1] 0.03614913  
##   
## $propertycountylandusecode  
## [1] NaN  
##   
## $propertylandusetypeid  
## [1] 1  
##   
## $rawcensustractandblock  
## [1] 0.9540183  
##   
## $regionidcity  
## [1] 0.8347834  
##   
## $regionidcounty  
## [1] 0.9999797  
##   
## $regionidzip  
## [1] 0.9115249  
##   
## $roomcnt  
## [1] 1  
##   
## $yearbuilt  
## [1] 0.7462459  
##   
## $structuretaxvaluedollarcnt  
## [1] 0.1477397  
##   
## $taxvaluedollarcnt  
## [1] 0.0173452  
##   
## $assessmentyear  
## [1] 1  
##   
## $landtaxvaluedollarcnt  
## [1] 0.01124671  
##   
## $taxamount  
## [1] 0.01561739

# Seed 2  
sample\_size = 10000  
set.seed(2)  
idxs = sample(1:nrow(NoNA2016),sample\_size,replace=F)  
subsample = NoNA2016[idxs,]  
pvalues = list()  
for (col in names(NoNA2016)) {  
 if (class(NoNA2016[,col]) %in% c("numeric","integer")) {  
 # Numeric variable. Using Kolmogorov-Smirnov test  
   
 pvalues[[col]] = ks.test(subsample[[col]],NoNA2016[[col]])$p.value  
   
 } else {  
 # Categorical variable. Using Pearson's Chi-square test  
   
 probs = table(NoNA2016[[col]])/nrow(NoNA2016)  
 pvalues[[col]] = chisq.test(table(subsample[[col]]),p=probs)$p.value  
   
 }  
}

## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties

## Warning in chisq.test(table(subsample[[col]]), p = probs): Chi-squared  
## approximation may be incorrect

## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties  
  
## Warning in ks.test.default(subsample[[col]], NoNA2016[[col]]): p-value will be  
## approximate in the presence of ties

pvalues

## $parcelid  
## [1] 0.6926251  
##   
## $bathroomcnt  
## [1] 0.9971033  
##   
## $bedroomcnt  
## [1] 0.4245304  
##   
## $calculatedbathnbr  
## [1] 0.9971033  
##   
## $calculatedfinishedsquarefeet  
## [1] 0.6365861  
##   
## $finishedsquarefeet12  
## [1] 0.6365861  
##   
## $fips  
## [1] 0.9999724  
##   
## $fullbathcnt  
## [1] 0.9998912  
##   
## $latitude  
## [1] 0.5279419  
##   
## $longitude  
## [1] 0.783989  
##   
## $lotsizesquarefeet  
## [1] 0.7611358  
##   
## $propertycountylandusecode  
## [1] NaN  
##   
## $propertylandusetypeid  
## [1] 0.9999961  
##   
## $rawcensustractandblock  
## [1] 0.84986  
##   
## $regionidcity  
## [1] 0.9996583  
##   
## $regionidcounty  
## [1] 1  
##   
## $regionidzip  
## [1] 0.8695746  
##   
## $roomcnt  
## [1] 0.9918689  
##   
## $yearbuilt  
## [1] 0.8433741  
##   
## $structuretaxvaluedollarcnt  
## [1] 0.6043258  
##   
## $taxvaluedollarcnt  
## [1] 0.7384443  
##   
## $assessmentyear  
## [1] 1  
##   
## $landtaxvaluedollarcnt  
## [1] 0.7563114  
##   
## $taxamount  
## [1] 0.9951607

# Save sample at seed 2  
# Commented out since it has already written: write.csv(subsample, "~/DS7330/cleansubsample2016.csv", row.names=TRUE)

# Seed 1 for 2017 data  
sample\_size = 10000  
set.seed(1)  
idxs = sample(1:nrow(NoNA2017),sample\_size,replace=F)  
subsample2017 = NoNA2017[idxs,]  
pvalues = list()  
for (col in names(NoNA2017)) {  
 if (class(NoNA2017[,col]) %in% c("numeric","integer")) {  
 # Numeric variable. Using Kolmogorov-Smirnov test  
   
 pvalues[[col]] = ks.test(subsample2017[[col]],NoNA2017[[col]])$p.value  
   
 } else {  
 # Categorical variable. Using Pearson's Chi-square test  
   
 probs = table(NoNA2017[[col]])/nrow(NoNA2017)  
 pvalues[[col]] = chisq.test(table(subsample2017[[col]]),p=probs)$p.value  
   
 }  
}

## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties

## Warning in chisq.test(table(subsample2017[[col]]), p = probs): Chi-squared  
## approximation may be incorrect

## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties

pvalues

## $parcelid  
## [1] 0.05071853  
##   
## $bathroomcnt  
## [1] 0.9999313  
##   
## $bedroomcnt  
## [1] 0.9996271  
##   
## $calculatedbathnbr  
## [1] 0.9999313  
##   
## $calculatedfinishedsquarefeet  
## [1] 0.5510696  
##   
## $finishedsquarefeet12  
## [1] 0.5510696  
##   
## $fips  
## [1] 0.9874461  
##   
## $fullbathcnt  
## [1] 0.9999254  
##   
## $latitude  
## [1] 0.640601  
##   
## $longitude  
## [1] 0.9829539  
##   
## $lotsizesquarefeet  
## [1] 0.8124603  
##   
## $propertycountylandusecode  
## [1] NaN  
##   
## $propertylandusetypeid  
## [1] 1  
##   
## $rawcensustractandblock  
## [1] 0.6357926  
##   
## $regionidcity  
## [1] 0.8618068  
##   
## $regionidcounty  
## [1] 0.9999996  
##   
## $regionidzip  
## [1] 0.8378729  
##   
## $roomcnt  
## [1] 0.9640767  
##   
## $yearbuilt  
## [1] 0.9159423  
##   
## $structuretaxvaluedollarcnt  
## [1] 0.9464951  
##   
## $taxvaluedollarcnt  
## [1] 0.396871  
##   
## $assessmentyear  
## [1] 1  
##   
## $landtaxvaluedollarcnt  
## [1] 0.2450761  
##   
## $taxamount  
## [1] 0.5185373

# Seed 2  
sample\_size = 10000  
set.seed(2)  
idxs = sample(1:nrow(NoNA2017),sample\_size,replace=F)  
subsample2017 = NoNA2017[idxs,]  
pvalues = list()  
for (col in names(NoNA2017)) {  
 if (class(NoNA2017[,col]) %in% c("numeric","integer")) {  
 # Numeric variable. Using Kolmogorov-Smirnov test  
   
 pvalues[[col]] = ks.test(subsample2017[[col]],NoNA2017[[col]])$p.value  
   
 } else {  
 # Categorical variable. Using Pearson's Chi-square test  
   
 probs = table(NoNA2017[[col]])/nrow(NoNA2017)  
 pvalues[[col]] = chisq.test(table(subsample2017[[col]]),p=probs)$p.value  
   
 }  
}

## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties

## Warning in chisq.test(table(subsample2017[[col]]), p = probs): Chi-squared  
## approximation may be incorrect

## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties  
  
## Warning in ks.test.default(subsample2017[[col]], NoNA2017[[col]]): p-value will  
## be approximate in the presence of ties

pvalues

## $parcelid  
## [1] 0.9498489  
##   
## $bathroomcnt  
## [1] 0.7236865  
##   
## $bedroomcnt  
## [1] 1  
##   
## $calculatedbathnbr  
## [1] 0.7236865  
##   
## $calculatedfinishedsquarefeet  
## [1] 0.3190295  
##   
## $finishedsquarefeet12  
## [1] 0.3190295  
##   
## $fips  
## [1] 0.9999969  
##   
## $fullbathcnt  
## [1] 0.7221509  
##   
## $latitude  
## [1] 0.5990702  
##   
## $longitude  
## [1] 0.9099151  
##   
## $lotsizesquarefeet  
## [1] 0.7928329  
##   
## $propertycountylandusecode  
## [1] NaN  
##   
## $propertylandusetypeid  
## [1] 0.9980481  
##   
## $rawcensustractandblock  
## [1] 0.9455285  
##   
## $regionidcity  
## [1] 0.9800262  
##   
## $regionidcounty  
## [1] 0.9999969  
##   
## $regionidzip  
## [1] 0.7454741  
##   
## $roomcnt  
## [1] 0.9999988  
##   
## $yearbuilt  
## [1] 0.7418883  
##   
## $structuretaxvaluedollarcnt  
## [1] 0.4602944  
##   
## $taxvaluedollarcnt  
## [1] 0.9116177  
##   
## $assessmentyear  
## [1] 1  
##   
## $landtaxvaluedollarcnt  
## [1] 0.6708195  
##   
## $taxamount  
## [1] 0.9844505

# Save sample at seed 2  
# Commented out since it has already written: write.csv(subsample2017, "~/DS7330/cleansubsample2017.csv", row.names=TRUE)