Data Cleaning & Analysis Spec

28 August 2018 09:35 AM

<u>Data Cleaning -</u>

Qualitative work

Determine what needs to be re-categorized (i.e. appears as quantitative, but is qualitative rank)
Merge and/or simplify existing variables, create new variable
Define data sets into quantitative vs qualitative
Transform qualitative into dummy/encoding

Identify & delete duplicate entries
Calculate & plot %missing entries bar chart -> exclude variable with missing entries > x%

<u>Test Normality - *Razaq to code*</u>

1. Dependent - Test Normality
    a. Plot histogram and run normality test for dependent
    - Include descriptive stats
2. Independent - Test Normality
    a. Plot histogram matrices for features

```
f = pd.melt(train, value_vars=quantitative)
g = sns.FacetGrid(f, col="variable",  col_wrap=2, sharex=False, sharey=False)
g = g.map(sns.distplot, "value")
```

    b. Plot probability plots (if you can do a matrix?)
    c. Normality test

```
test_normality = lambda x: stats.shapiro(x.fillna(0))[1] < 0.01
normal = pd.DataFrame(train[quantitative])
normal = normal.apply(test_normality)
print(not normal.any())
```

<u>Outlier handling & collinearity - *Razaq to code*</u>

Dendogram ***-> indication of collinearity***

Plot correlation matrix -> qualitative & quantitative classification -

```
k = 10 #number of variables for heatmap
cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index
cm = np.corrcoef(df_train[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'size': 10},
yticklabels=cols.values, xticklabels=cols.values)
plt.show()
```

Plot multiple scatters ***-> test for collinearity , homoscedasticity, & outliers -quantitative data only***

```
sns.set()
cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'TotalBsmtSF', 'FullBath', 'YearBuilt']
sns.pairplot(df_train[cols], size = 2.5)
plt.show()
```

Plot multiple box plot ***-> test for collinearity , homoscedasticity, & outliers - qualitative only***

```
for c in qualitative:
  train[c] = train[c].astype('category')
  if train[c].isnull().any():
    train[c] = train[c].cat.add_categories(['MISSING'])
    train[c] = train[c].fillna('MISSING')
def boxplot(x, y, **kwargs):
    sns.boxplot(x=x, y=y)
    x=plt.xticks(rotation=90)
f = pd.melt(train, id_vars=['SalePrice'], value_vars=qualitative)
g = sns.FacetGrid(f, col="variable",  col_wrap=2, sharex=False, sharey=False, size=5)
g = g.map(boxplot, "value", "SalePrice"
```

Depending on above, direct tests for collinearity, homoscedasticity, autocorrelation

Transformation - ZW, SJ, FF

Outlier removal & transformations