COMPREHENSIVE DATA EXPLORATION WITH PYTHON

Pedro Marcelino (http://pmarcelino.com) - February 2017

Other Kernels: Data analysis and feature extraction with Python (https://www.kaggle.com/pmarcelino/data-analysis-and-feature-extraction-with-python)

'The most difficult thing in life is to know yourself'

This quote belongs to Thales of Miletus. Thales was a Greek/Phonecian philosopher, mathematician and astronomer, which is recognised as the first individual in Western civilisation known to have entertained and engaged in scientific thought (source: https://en.wikipedia.org/wiki/Thales (https://en.wikipedia.org/wiki/Thales))

I wouldn't say that knowing your data is the most difficult thing in data science, but it is time-consuming. Therefore, it's easy to overlook this initial step and jump too soon into the water.

So I tried to learn how to swim before jumping into the water. Based on Hair et al. (2013) (https://amzn.to/2JuDmvo), chapter 'Examining your data', I did my best to follow a comprehensive, but not exhaustive, analysis of the data. I'm far from reporting a rigorous study in this kernel, but I hope that it can be useful for the community, so I'm sharing how I applied some of those data analysis principles to this problem.

Despite the strange names I gave to the chapters, what we are doing in this kernel is something like:

1. **Understand the problem**. We'll look at each variable and do a philosophical analysis about their meaning and importance for this problem.
2. **Univariable study**. We'll just focus on the dependent variable ('SalePrice') and try to know a little bit more about it.
3. **Multivariate study**. We'll try to understand how the dependent variable and independent variables relate.
4. **Basic cleaning**. We'll clean the dataset and handle the missing data, outliers and categorical variables.

5. **Test assumptions**. We'll check if our data meets the assumptions required by most multivariate techniques.

Now, it's time to have fun!

In [1]:
```
#invite people for the Kaggle party
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import norm
from sklearn.preprocessing import StandardScaler
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [2]:
```
#bring in the six packs
df_train = pd.read_csv('../input/train.csv')
```

In [3]:
```
#check the decoration
df_train.columns
```

Out[3]:
```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Str
eet',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'Bldg
Type',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'Year
RemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnr
Type',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQua
```

```
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQua
l',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Hea
ting',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrS
F',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath',
'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'Ga
rageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'Ga
rageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'Pool
QC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleTy
pe',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

1. So... What can we expect?

In order to understand our data, we can look at each variable and try to understand their meaning and relevance to this problem. I know this is time-consuming, but it will give us the flavour of our dataset.

In order to have some discipline in our analysis, we can create an Excel spreadsheet with the following columns:

- **Variable** - Variable name.
- **Type** - Identification of the variables' type. There are two possible values for this field: 'numerical' or 'categorical'. By 'numerical' we mean variables for which the values are numbers, and by 'categorical' we mean variables for which the values are categories.
- **Segment** - Identification of the variables' segment. We can define three possible segments: building, space or location. When we say 'building', we mean a variable that relates to the physical characteristics of the building (e.g. 'OverallQual'). When we say 'space', we mean a

physical characteristics of the building (e.g. 'OverallQual'). When we say 'space', we mean a variable that reports space properties of the house (e.g. 'TotalBsmtSF'). Finally, when we say a 'location', we mean a variable that gives information about the place where the house is located (e.g. 'Neighborhood').

- **Expectation** - Our expectation about the variable influence in 'SalePrice'. We can use a categorical scale with 'High', 'Medium' and 'Low' as possible values.
- **Conclusion** - Our conclusions about the importance of the variable, after we give a quick look at the data. We can keep with the same categorical scale as in 'Expectation'.
- **Comments** - Any general comments that occured to us.

While 'Type' and 'Segment' is just for possible future reference, the column 'Expectation' is important because it will help us develop a 'sixth sense'. To fill this column, we should read the description of all the variables and, one by one, ask ourselves:

- Do we think about this variable when we are buying a house? (e.g. When we think about the house of our dreams, do we care about its 'Masonry veneer type'?).
- If so, how important would this variable be? (e.g. What is the impact of having 'Excellent' material on the exterior instead of 'Poor'? And of having 'Excellent' instead of 'Good'?).
- Is this information already described in any other variable? (e.g. If 'LandContour' gives the flatness of the property, do we really need to know the 'LandSlope'?).

After this daunting exercise, we can filter the spreadsheet and look carefully to the variables with 'High' 'Expectation'. Then, we can rush into some scatter plots between those variables and 'SalePrice', filling in the 'Conclusion' column which is just the correction of our expectations.

I went through this process and concluded that the following variables can play an important role in this problem:

- OverallQual (which is a variable that I don't like because I don't know how it was computed; a funny exercise would be to predict 'OverallQual' using all the other variables available).
- YearBuilt.
- TotalBsmtSF.
- GrLivArea.

I ended up with two 'building' variables ('OverallQual' and 'YearBuilt') and two 'space' variables ('TotalBsmtSF' and 'GrLivArea'). This might be a little bit unexpected as it goes against the real estate mantra that all that matters is 'location, location and location'. It is possible that this quick data examination process was a bit harsh for categorical variables. For example, I expected the

'Neighborhood' variable to be more relevant, but after the data examination I ended up excluding it. Maybe this is related to the use of scatter plots instead of boxplots, which are more suitable for categorical variables visualization. The way we visualize data often influences our conclusions.

However, the main point of this exercise was to think a little about our data and expectactions, so I think we achieved our goal. Now it's time for 'a little less conversation, a little more action please'. Let's **shake it!**

2. First things first: analysing 'SalePrice'

'SalePrice' is the reason of our quest. It's like when we're going to a party. We always have a reason to be there. Usually, women are that reason. (disclaimer: adapt it to men, dancing or alcohol, according to your preferences)

Using the women analogy, let's build a little story, the story of 'How we met 'SalePrice''.

Everything started in our Kaggle party, when we were looking for a dance partner. After a while searching in the dance floor, we saw a girl, near the bar, using dance shoes. That's a sign that she's there to dance. We spend much time doing predictive modelling and participating in analytics competitions, so talking with girls is not one of our super powers. Even so, we gave it a try:

'Hi, I'm Kaggly! And you? 'SalePrice'? What a beautiful name! You know 'SalePrice', could you give me some data about you? I just developed a model to calculate the probability of a successful relationship between two people. I'd like to apply it to us!'

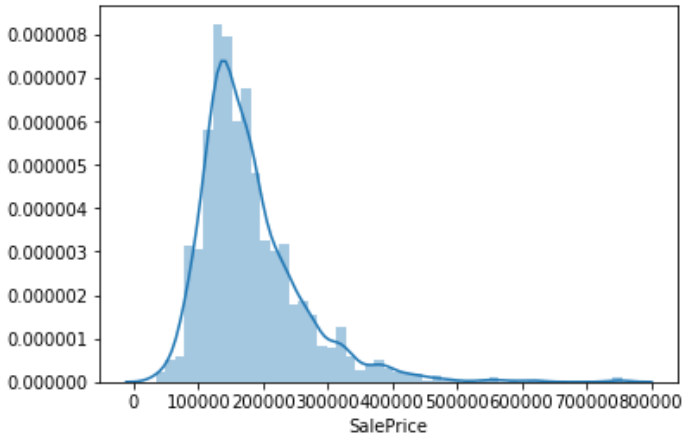
In [4]:
```
#descriptive statistics summary
df_train['SalePrice'].describe()
```

Out[4]:
```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
```

```
max       755000.000000
Name: SalePrice, dtype: float64
```

'Very well... It seems that your minimum price is larger than zero. Excellent! You don't have one of those personal traits that would destroy my model! Do you have any picture that you can send me? I don't know... like, you in the beach... or maybe a selfie in the gym?'

In [5]:
```
#histogram
sns.distplot(df_train['SalePrice']);
```



'Ah! I see you that you use seaborn makeup when you're going out... That's so elegant! I also see that you:

- ***Deviate from the normal distribution.***
- ***Have appreciable positive skewness.***
- ***Show peakedness.***

This is getting interesting! 'SalePrice', could you give me your body measures?'

In [6]:
```
#skewness and kurtosis
print("Skewness: %f" % df_train['SalePrice'].skew())
print("Kurtosis: %f" % df_train['SalePrice'].kurt())
```

```
Skewness: 1.882876
Kurtosis: 6.536282
```

'Amazing! If my love calculator is correct, our success probability is 97.834657%. I think we should meet again! Please, keep my number and give me a call if you're free next Friday. See you in a while, crocodile!'

'SalePrice', her buddies and her interests

It is military wisdom to choose the terrain where you will fight. As soon as 'SalePrice' walked away, we went to Facebook. Yes, now this is getting serious. Notice that this is not stalking. It's just an intense research of an individual, if you know what I mean.

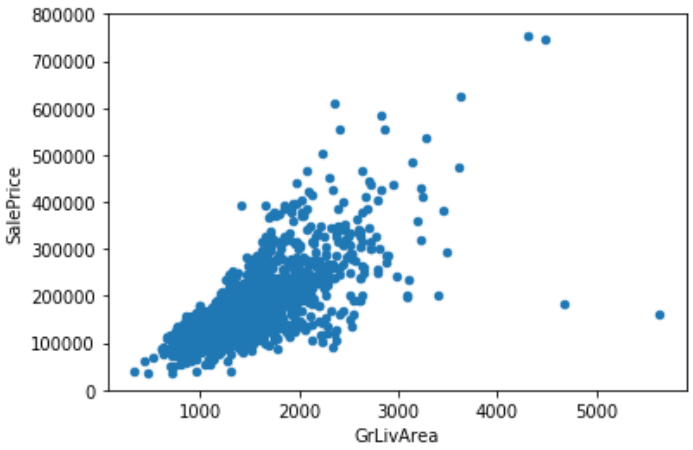
According to her profile, we have some common friends. Besides Chuck Norris, we both know 'GrLivArea' and 'TotalBsmtSF'. Moreover, we also have common interests such as 'OverallQual' and 'YearBuilt'. This looks promising!

To take the most out of our research, we will start by looking carefully at the profiles of our common friends and later we will focus on our common interests.

Relationship with numerical variables

In [7]:
```
#scatter plot grlivarea/saleprice
var = 'GrLivArea'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
```

```
data.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```

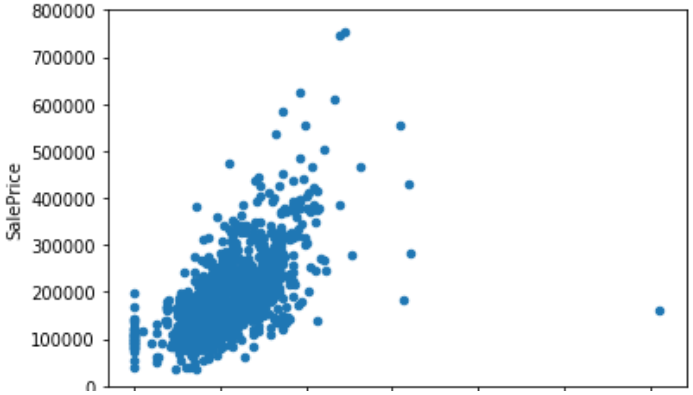


Hmmm... It seems that 'SalePrice' and 'GrLivArea' are really old friends, with a ***linear relationship.***

And what about 'TotalBsmtSF'?

In [8]:

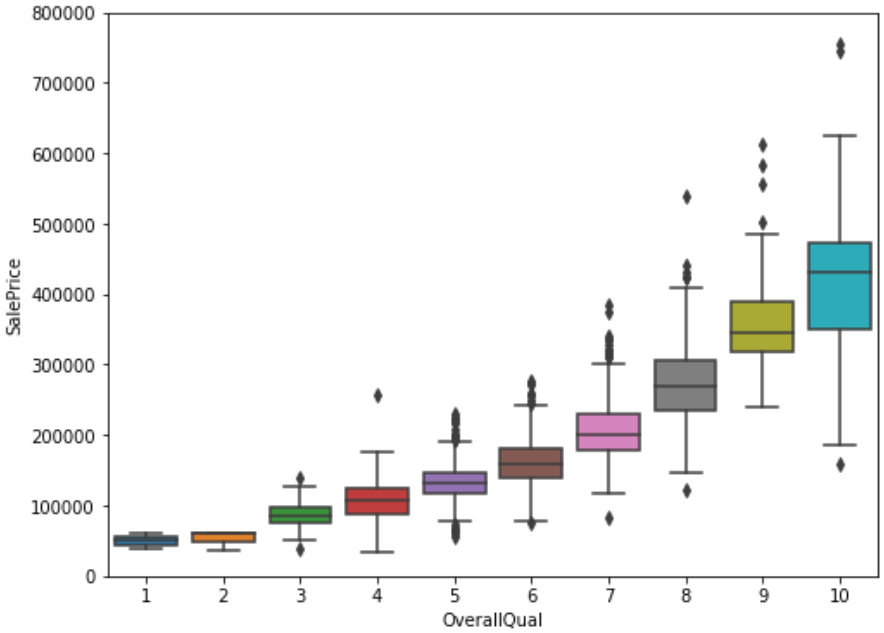
```
#scatter plot totalbsmtsf/saleprice
var = 'TotalBsmtSF'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```



*'TotalBsmtSF' is also a great friend of 'SalePrice' but this seems a much more emotional relationship! Everything is ok and suddenly, in a **strong linear (exponential?)** reaction, everything changes. Moreover, it's clear that sometimes 'TotalBsmtSF' closes in itself and gives zero credit to 'SalePrice'.*

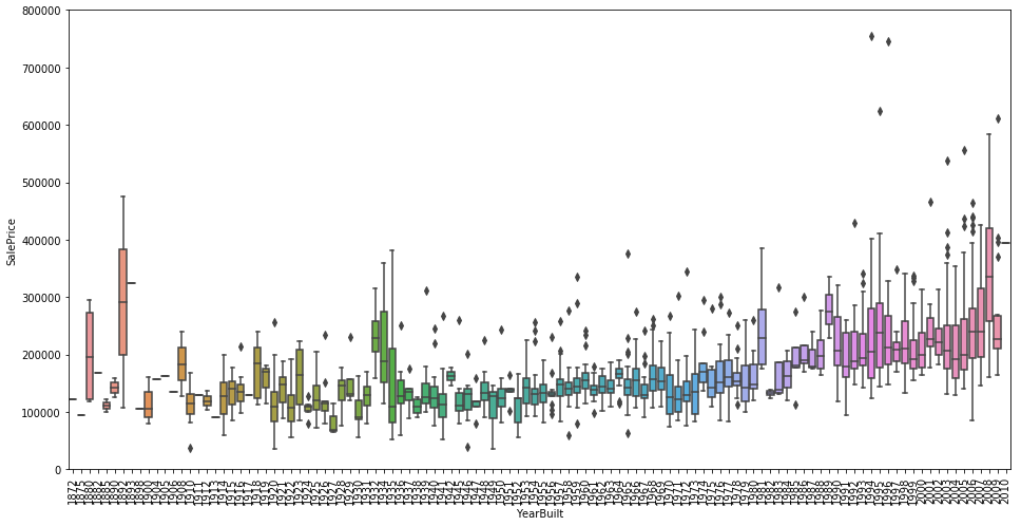
Relationship with categorical features

In [9]:
```
#box plot overallqual/saleprice
var = 'OverallQual'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);
```



Like all the pretty girls, 'SalePrice' enjoys 'OverallQual'. Note to self: consider whether McDonald's is suitable for the first date.

In [10]:
```
var = 'YearBuilt'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
f, ax = plt.subplots(figsize=(16, 8))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);
plt.xticks(rotation=90);
```



Although it's not a strong tendency, I'd say that 'SalePrice' is more prone to spend more money in new stuff than in old relics.

Note: we don't know if 'SalePrice' is in constant prices. Constant prices try to remove the effect of inflation. If 'SalePrice' is not in constant prices, it should be, so than prices are comparable over the years.

In summary

Stories aside, we can conclude that:

- 'GrLivArea' and 'TotalBsmtSF' seem to be linearly related with 'SalePrice'. Both relationships are positive, which means that as one variable increases, the other also increases. In the case of 'TotalBsmtSF', we can see that the slope of the linear relationship is particularly high.
- 'OverallQual' and 'YearBuilt' also seem to be related with 'SalePrice'. The relationship seems to be stronger in the case of 'OverallQual', where the box plot shows how sales prices increase with the overall quality.

We just analysed four variables, but there are many other that we should analyse. The trick here seems to be the choice of the right features (feature selection) and not the definition of complex relationships between them (feature engineering).

That said, let's separate the wheat from the chaff.

3. Keep calm and work smart

Until now we just followed our intuition and analysed the variables we thought were important. In spite of our efforts to give an objective character to our analysis, we must say that our starting point was subjective.

As an engineer, I don't feel comfortable with this approach. All my education was about developing a disciplined mind, able to withstand the winds of subjectivity. There's a reason for that. Try to be subjective in structural engineering and you will see physics making things fall down. It can hurt.

So, let's overcome inertia and do a more objective analysis.

The 'plasma soup'

'In the very beginning there was nothing except for a plasma soup. What is known of these brief moments in time, at the start of our study of cosmology, is largely conjectural. However, science has devised some sketch of what probably happened, based on what is known about the universe today.'

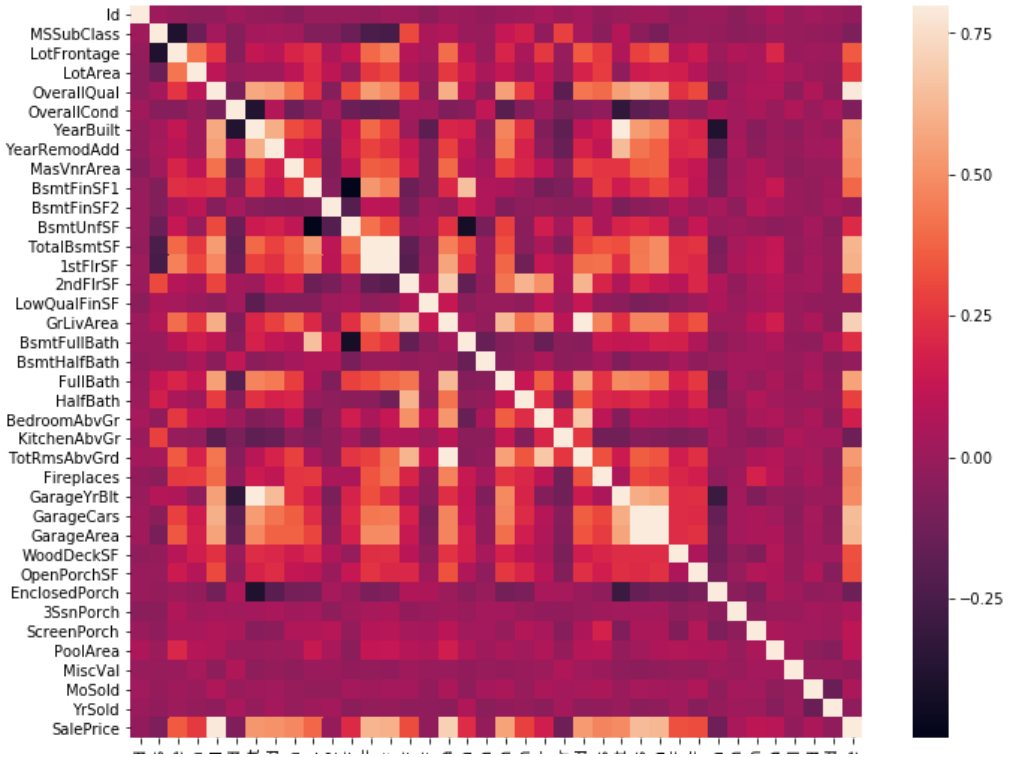
(source: http://umich.edu/~gs265/bigbang.htm (http://umich.edu/~gs265/bigbang.htm))

To explore the universe, we will start with some practical recipes to make sense of our 'plasma soup':

- Correlation matrix (heatmap style).
- 'SalePrice' correlation matrix (zoomed heatmap style).
- Scatter plots between the most correlated variables (move like Jagger style).

Correlation matrix (heatmap style)

In [11]:
```
#correlation matrix
corrmat = df_train.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True);
```



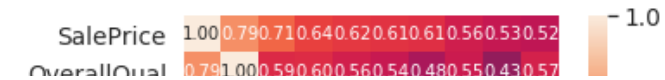
In my opinion, this heatmap is the best way to get a quick overview of our 'plasma soup' and its relationships. (Thank you @seaborn!)

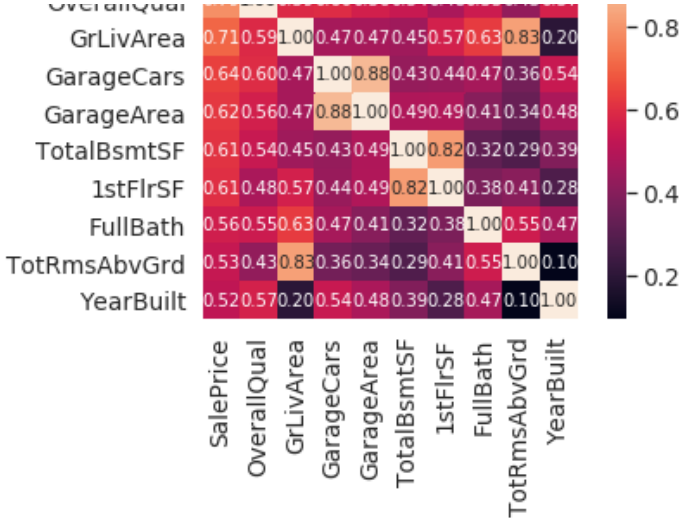
At first sight, there are two red colored squares that get my attention. The first one refers to the 'TotalBsmtSF' and '1stFlrSF' variables, and the second one refers to the 'Garage*X*' variables. Both cases show how significant the correlation is between these variables. Actually, this correlation is so strong that it can indicate a situation of multicollinearity. If we think about these variables, we can conclude that they give almost the same information so multicollinearity really occurs. Heatmaps are great to detect this kind of situations and in problems dominated by feature selection, like ours, they are an essential tool.

Another thing that got my attention was the 'SalePrice' correlations. We can see our well-known 'GrLivArea', 'TotalBsmtSF', and 'OverallQual' saying a big 'Hi!', but we can also see many other variables that should be taken into account. That's what we will do next.

'SalePrice' correlation matrix (zoomed heatmap style)

In [12]:
```
#saleprice correlation matrix
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', an
not_kws={'size': 10}, yticklabels=cols.values, xticklabels=cols.values
)
plt.show()
```





According to our crystal ball, these are the variables most correlated with 'SalePrice'. My thoughts on this:

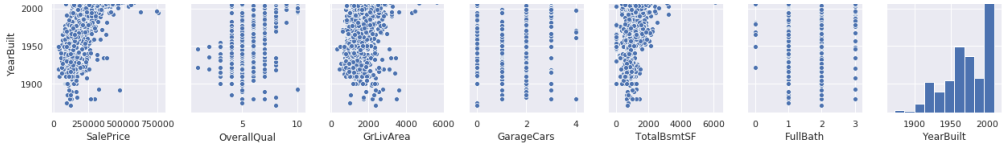
- 'OverallQual', 'GrLivArea' and 'TotalBsmtSF' are strongly correlated with 'SalePrice'. Check!
- 'GarageCars' and 'GarageArea' are also some of the most strongly correlated variables. However, as we discussed in the last sub-point, the number of cars that fit into the garage is a consequence of the garage area. 'GarageCars' and 'GarageArea' are like twin brothers. You'll never be able to distinguish them. Therefore, we just need one of these variables in our analysis (we can keep 'GarageCars' since its correlation with 'SalePrice' is higher).
- 'TotalBsmtSF' and '1stFloor' also seem to be twin brothers. We can keep 'TotalBsmtSF' just to say that our first guess was right (re-read 'So... What can we expect?').
- 'FullBath'?? Really?
- 'TotRmsAbvGrd' and 'GrLivArea', twin brothers again. Is this dataset from Chernobyl?
- Ah... 'YearBuilt'... It seems that 'YearBuilt' is slightly correlated with 'SalePrice'. Honestly, it scares me to think about 'YearBuilt' because I start feeling that we should do a little bit of time-series analysis to get this right. I'll leave this as a homework for you.

Let's proceed to the scatter plots.

Scatter plots between 'SalePrice' and correlated variables (move like Jagger style)

Get ready for what you're about to see. I must confess that the first time I saw these scatter plots I was totally blown away! So much information in so short space... It's just amazing. Once more, thank you @seaborn! You make me 'move like Jagger'!

In [13]:
```
#scatterplot
sns.set()
cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'TotalB
smtSF', 'FullBath', 'YearBuilt']
sns.pairplot(df_train[cols], size = 2.5)
plt.show();
```



Although we already know some of the main figures, this mega scatter plot gives us a reasonable idea about variables relationships.

One of the figures we may find interesting is the one between 'TotalBsmtSF' and 'GrLiveArea'. In this figure we can see the dots drawing a linear line, which almost acts like a border. It totally makes sense that the majority of the dots stay below that line. Basement areas can be equal to the above ground living area, but it is not expected a basement area bigger than the above ground living area (unless you're trying to buy a bunker).

The plot concerning 'SalePrice' and 'YearBuilt' can also make us think. In the bottom of the 'dots cloud', we see what almost appears to be a shy exponential function (be creative). We can also see this same tendency in the upper limit of the 'dots cloud' (be even more creative). Also, notice how the set of dots regarding the last years tend to stay above this limit (I just wanted to say that prices are increasing faster now).

Ok, enough of Rorschach test for now. Let's move forward to what's missing: missing data!

4. Missing data

Important questions when thinking about missing data:

- How prevalent is the missing data?
- Is missing data random or does it have a pattern?

The answer to these questions is important for practical reasons because missing data can imply a reduction of the sample size. This can prevent us from proceeding with the analysis. Moreover, from a substantive perspective, we need to ensure that the missing data process is not biased and hidding an inconvenient truth.

In [14]:

```
#missing data
total = df_train.isnull().sum().sort_values(ascending=False)
percent = (df_train.isnull().sum()/df_train.isnull().count()).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
missing_data.head(20)
```

Out[14]:

	Total	Percent
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageCond	81	0.055479
GarageType	81	0.055479
GarageYrBlt	81	0.055479
GarageFinish	81	0.055479
GarageQual	81	0.055479
BsmtExposure	38	0.026027
BsmtFinType2	38	0.026027
BsmtFinType1	37	0.025342
BsmtCond	37	0.025342
BsmtQual	37	0.025342
MasVnrArea	8	0.005479
MasVnrType	8	0.005479
Electrical	1	0.000685
Utilities	0	0.000000

Let's analyse this to understand how to handle the missing data.

We'll consider that when more than 15% of the data is missing, we should delete the corresponding variable and pretend it never existed. This means that we will not try any trick to fill the missing data in these cases. According to this, there is a set of variables (e.g. 'PoolQC', 'MiscFeature', 'Alley', etc.) that we should delete. The point is: will we miss this data? I don't think so. None of these variables seem to be very important, since most of them are not aspects in which we think about when buying a house (maybe that's the reason why data is missing?). Moreover, looking closer at the variables, we could say that variables like 'PoolQC', 'MiscFeature' and 'FireplaceQu' are strong candidates for outliers, so we'll be happy to delete them.

In what concerns the remaining cases, we can see that 'Garage*X*' variables have the same number of missing data. I bet missing data refers to the same set of observations (although I will not check it; it's just 5% and we should not spend $20 in 5$ problems). Since the most important information regarding garages is expressed by 'GarageCars' and considering that we are just talking about 5% of missing data, I'll delete the mentioned 'Garage*X*' variables. The same logic applies to 'Bsmt*X*' variables.

Regarding 'MasVnrArea' and 'MasVnrType', we can consider that these variables are not essential. Furthermore, they have a strong correlation with 'YearBuilt' and 'OverallQual' which are already considered. Thus, we will not lose information if we delete 'MasVnrArea' and 'MasVnrType'.

Finally, we have one missing observation in 'Electrical'. Since it is just one observation, we'll delete this observation and keep the variable.

In summary, to handle missing data, we'll delete all the variables with missing data, except the variable 'Electrical'. In 'Electrical' we'll just delete the observation with missing data.

In [15]:
```
#dealing with missing data
df_train = df_train.drop((missing_data[missing_data['Total'] > 1]).ind
ex,1)
df_train = df_train.drop(df_train.loc[df_train['Electrical'].isnull()]
.index)
df_train.isnull().sum().max() #just checking that there's no missing d
ata missing...
```

Out[15]:

Out[15]:
0

Out liars!

Outliers is also something that we should be aware of. Why? Because outliers can markedly affect our models and can be a valuable source of information, providing us insights about specific behaviours.

Outliers is a complex subject and it deserves more attention. Here, we'll just do a quick analysis through the standard deviation of 'SalePrice' and a set of scatter plots.

Univariate analysis

The primary concern here is to establish a threshold that defines an observation as an outlier. To do so, we'll standardize the data. In this context, data standardization means converting data values to have mean of 0 and a standard deviation of 1.

In [16]:
```
#standardizing data
saleprice_scaled = StandardScaler().fit_transform(df_train['SalePrice'
][:,np.newaxis]);
low_range = saleprice_scaled[saleprice_scaled[:,0].argsort()][:10]
high_range= saleprice_scaled[saleprice_scaled[:,0].argsort()][-10:]
print('outer range (low) of the distribution:')
print(low_range)
print('\nouter range (high) of the distribution:')
print(high_range)
```

```
outer range (low) of the distribution:
[[-1.83820775]
 [-1.83303414]
 [-1.80044422]
 [-1.78282123]
```

```
 [-1.77400974]
 [-1.62295562]
 [-1.6166617 ]
 [-1.58519209]
 [-1.58519209]
 [-1.57269236]]

outer range (high) of the distribution:
[[3.82758058]
 [4.0395221 ]
 [4.49473628]
 [4.70872962]
 [4.728631  ]
 [5.06034585]
 [5.42191907]
 [5.58987866]
 [7.10041987]
 [7.22629831]]
```

How 'SalePrice' looks with her new clothes:

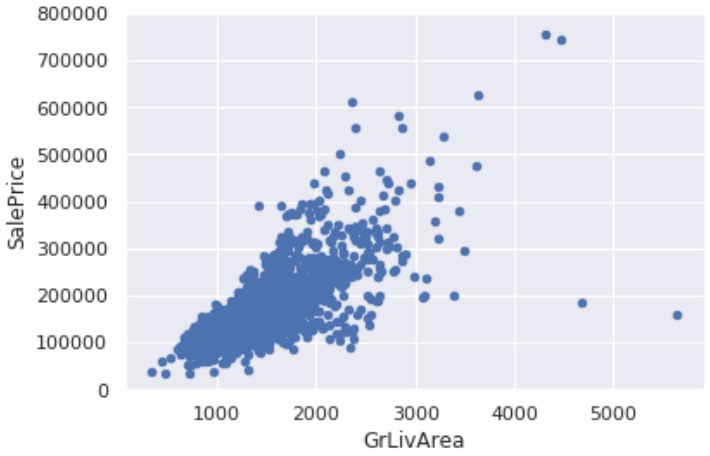
- Low range values are similar and not too far from 0.
- High range values are far from 0 and the 7.something values are really out of range.

For now, we'll not consider any of these values as an outlier but we should be careful with those two 7.something values.

Bivariate analysis

We already know the following scatter plots by heart. However, when we look to things from a new perspective, there's always something to discover. As Alan Kay said, 'a change in perspective is worth 80 IQ points'.

In [17]:
```
#bivariate analysis saleprice/grlivarea
var = 'GrLivArea'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```

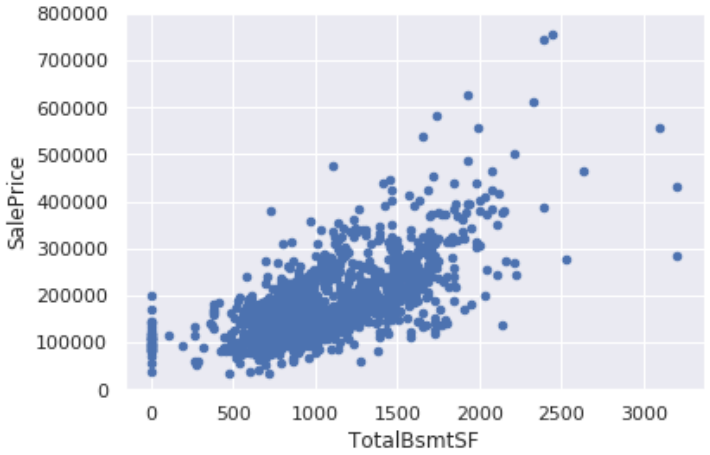


What has been revealed:

- The two values with bigger 'GrLivArea' seem strange and they are not following the crowd. We can speculate why this is happening. Maybe they refer to agricultural area and that could explain the low price. I'm not sure about this but I'm quite confident that these two points are not representative of the typical case. Therefore, we'll define them as outliers and delete them.
- The two observations in the top of the plot are those 7.something observations that we said we should be careful about. They look like two special cases, however they seem to be following the trend. For that reason, we will keep them.

In [18]:
```
#deleting points
df_train.sort_values(by = 'GrLivArea', ascending = False)[:2]
df_train = df_train.drop(df_train[df_train['Id'] == 1299].index)
df_train = df_train.drop(df_train[df_train['Id'] == 524].index)
```

In [19]:
```
#bivariate analysis saleprice/grlivarea
var = 'TotalBsmtSF'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```



We can feel tempted to eliminate some observations (e.g. TotalBsmtSF > 3000) but I suppose it's not worth it. We can live with that, so we'll not do anything.

5. Getting hard core

In Ayn Rand's novel, 'Atlas Shrugged', there is an often-repeated question: who is John Galt? A big part of the book is about the quest to discover the answer to this question.

I feel Randian now. Who is 'SalePrice'?

The answer to this question lies in testing for the assumptions underlying the statistical bases for multivariate analysis. We already did some data cleaning and discovered a lot about 'SalePrice'. Now it's time to go deep and understand how 'SalePrice' complies with the statistical assumptions that

It's time to go deep and understand how 'SalePrice' complies with the statistical assumptions that enables us to apply multivariate techniques.

According to Hair et al. (2013) (https://amzn.to/2uC3j9p), four assumptions should be tested:

- **Normality** - When we talk about normality what we mean is that the data should look like a normal distribution. This is important because several statistic tests rely on this (e.g. t-statistics). In this exercise we'll just check univariate normality for 'SalePrice' (which is a limited approach). Remember that univariate normality doesn't ensure multivariate normality (which is what we would like to have), but it helps. Another detail to take into account is that in big samples (>200 observations) normality is not such an issue. However, if we solve normality, we avoid a lot of other problems (e.g. heteroscedacity) so that's the main reason why we are doing this analysis.
- **Homoscedasticity** - I just hope I wrote it right. Homoscedasticity refers to the 'assumption that dependent variable(s) exhibit equal levels of variance across the range of predictor variable(s)' (Hair et al., 2013) (https://amzn.to/2uC3j9p). Homoscedasticity is desirable because we want the error term to be the same across all values of the independent variables.
- **Linearity**- The most common way to assess linearity is to examine scatter plots and search for linear patterns. If patterns are not linear, it would be worthwhile to explore data transformations. However, we'll not get into this because most of the scatter plots we've seen appear to have linear relationships.
- **Absence of correlated errors** - Correlated errors, like the definition suggests, happen when one error is correlated to another. For instance, if one positive error makes a negative error systematically, it means that there's a relationship between these variables. This occurs often in time series, where some patterns are time related. We'll also not get into this. However, if you detect something, try to add a variable that can explain the effect you're getting. That's the most common solution for correlated errors.

What do you think Elvis would say about this long explanation? 'A little less conversation, a little more action please'? Probably... By the way, do you know what was Elvis's last great hit?

(...)

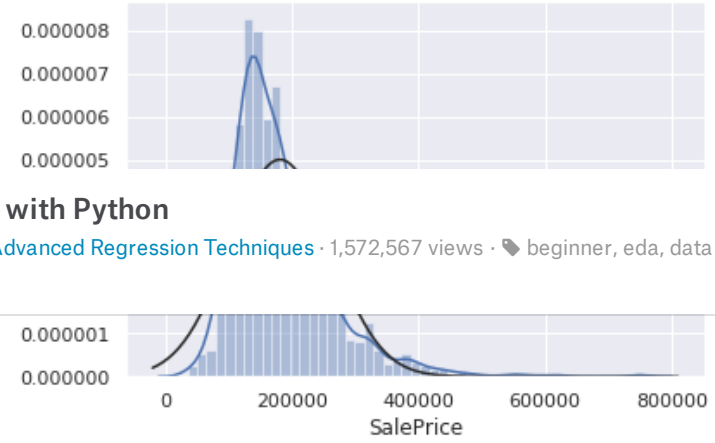
The bathroom floor.

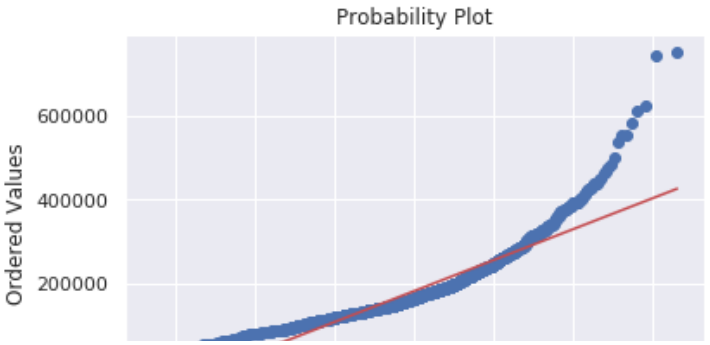
In the search for normality

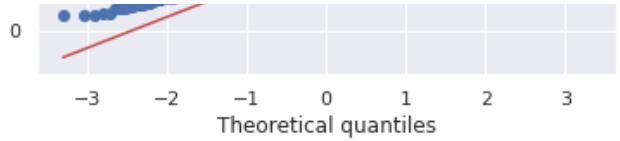
The point here is to test 'SalePrice' in a very lean way. We'll do this paying attention to:

- **Histogram** - Kurtosis and skewness.
- **Normal probability plot** - Data distribution should closely follow the diagonal that represents the normal distribution.

In [20]:
```
#histogram and normal probability plot
sns.distplot(df_train['SalePrice'], fit=norm);
fig = plt.figure()
res = stats.probplot(df_train['SalePrice'], plot=plt)
```







Ok, 'SalePrice' is not normal. It shows 'peakedness', positive skewness and does not follow the diagonal line.

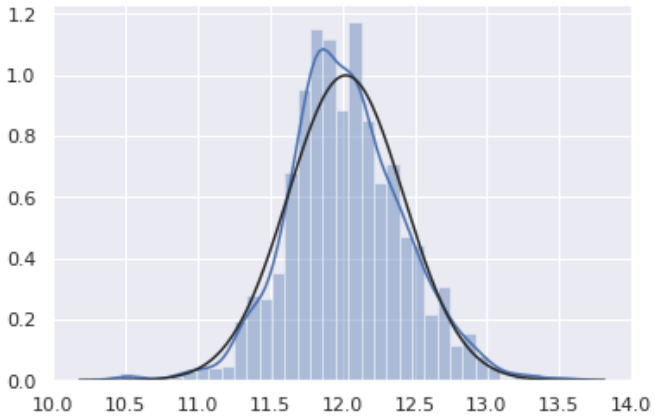
But everything's not lost. A simple data transformation can solve the problem. This is one of the awesome things you can learn in statistical books: in case of positive skewness, log transformations usually works well. When I discovered this, I felt like an Hogwarts' student discovering a new cool spell.

Avada kedavra!

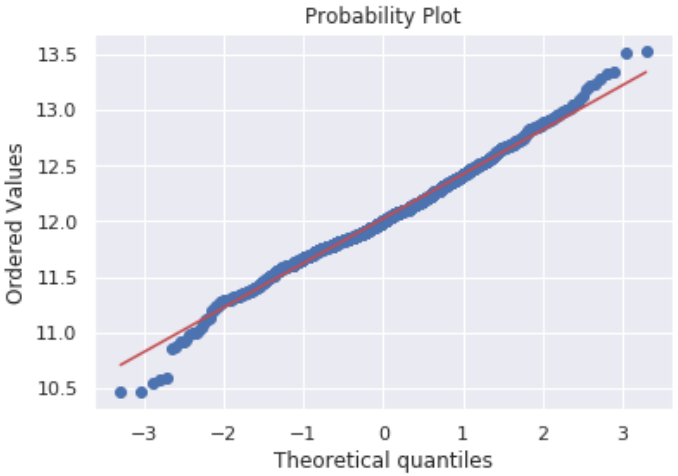
In [21]:
```
#applying log transformation
df_train['SalePrice'] = np.log(df_train['SalePrice'])
```



```
fig = plt.figure()
res = stats.probplot(df_train['SalePrice'], plot=plt)
```

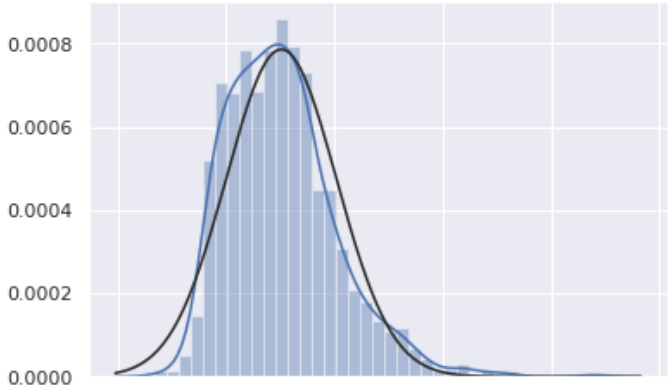


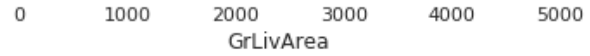
SalePrice

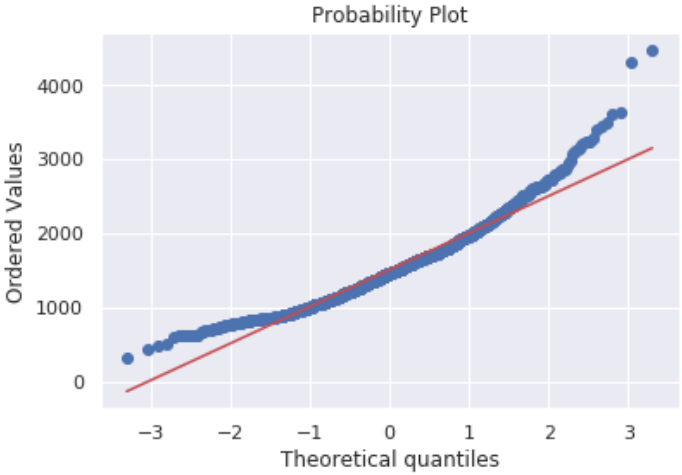


Done! Let's check what's going on with 'GrLivArea'.

In [23]:
```
#histogram and normal probability plot
sns.distplot(df_train['GrLivArea'], fit=norm);
fig = plt.figure()
res = stats.probplot(df_train['GrLivArea'], plot=plt)
```



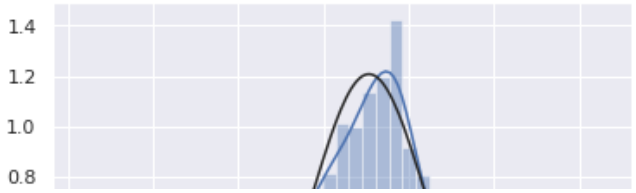


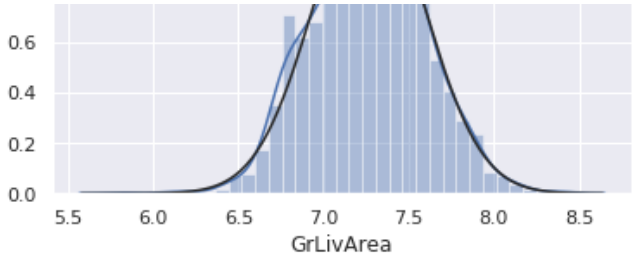


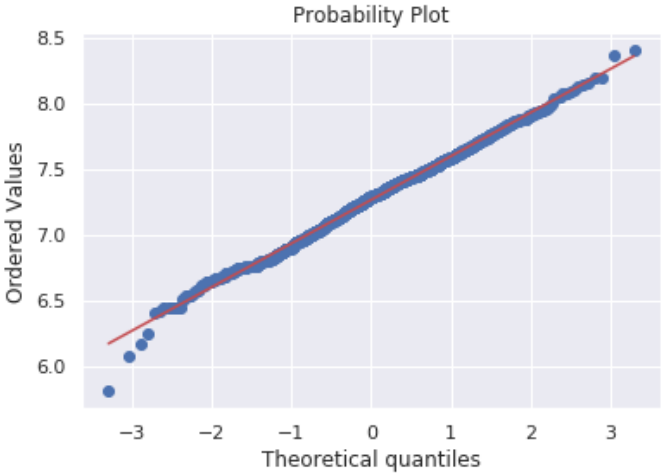
Tastes like skewness... *Avada kedavra!*

In [24]:
```
#data transformation
df_train['GrLivArea'] = np.log(df_train['GrLivArea'])
```

In [25]:
```
#transformed histogram and normal probability plot
sns.distplot(df_train['GrLivArea'], fit=norm);
fig = plt.figure()
res = stats.probplot(df_train['GrLivArea'], plot=plt)
```

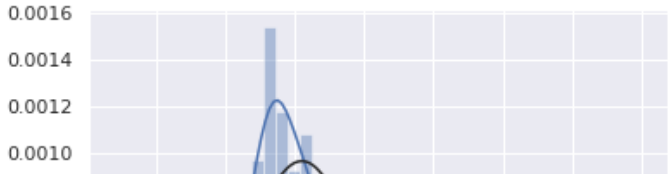


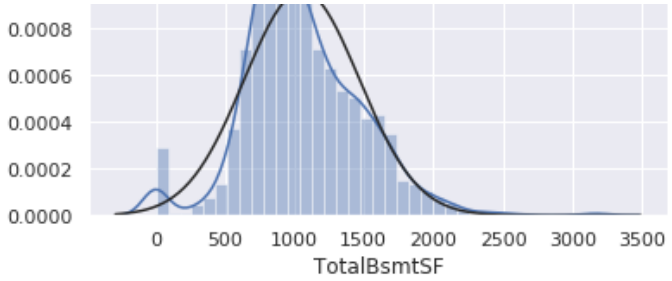


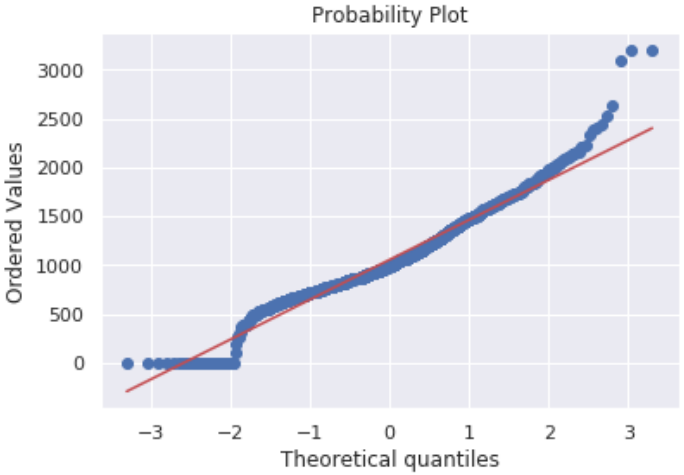


Next, please...

In [26]:
```
#histogram and normal probability plot
sns.distplot(df_train['TotalBsmtSF'], fit=norm);
fig = plt.figure()
res = stats.probplot(df_train['TotalBsmtSF'], plot=plt)
```







Ok, now we are dealing with the big boss. What do we have here?

- Something that, in general, presents skewness.
- A significant number of observations with value zero (houses without basement).
- A big problem because the value zero doesn't allow us to do log transformations.

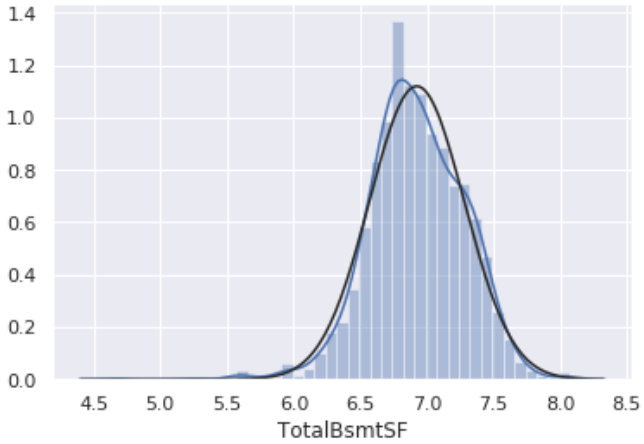
To apply a log transformation here, we'll create a variable that can get the effect of having or not having basement (binary variable). Then, we'll do a log transformation to all the non-zero observations, ignoring those with value zero. This way we can transform data, without losing the effect of having or not basement.

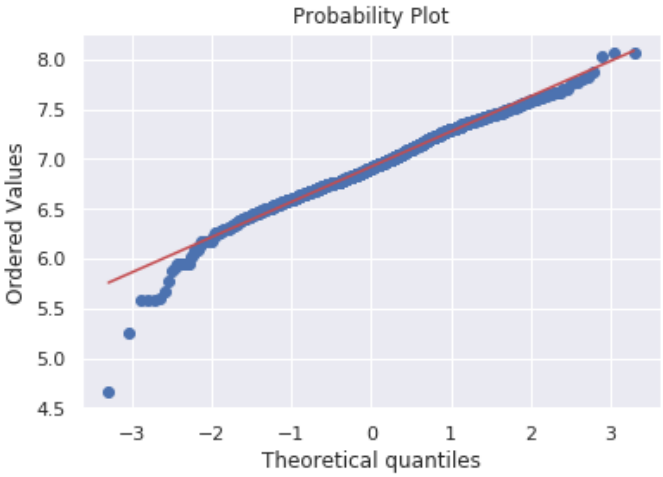
I'm not sure if this approach is correct. It just seemed right to me. That's what I call 'high risk engineering'.

In [27]:
```
#create column for new variable (one is enough because it's a binary categorical feature)
#if area>0 it gets 1, for area==0 it gets 0
df_train['HasBsmt'] = pd.Series(len(df_train['TotalBsmtSF']), index=df_train.index)
df_train['HasBsmt'] = 0
df_train.loc[df_train['TotalBsmtSF']>0,'HasBsmt'] = 1
```

In [28]:
```
#transform data
df_train.loc[df_train['HasBsmt']==1,'TotalBsmtSF'] = np.log(df_train['TotalBsmtSF'])
```

In [29]:
```
#histogram and normal probability plot
sns.distplot(df_train[df_train['TotalBsmtSF']>0]['TotalBsmtSF'], fit=norm);
fig = plt.figure()
res = stats.probplot(df_train[df_train['TotalBsmtSF']>0]['TotalBsmtSF'], plot=plt)
```



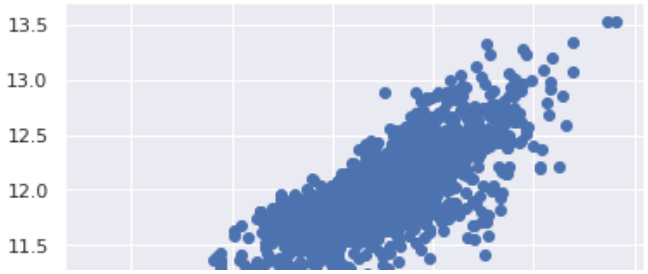


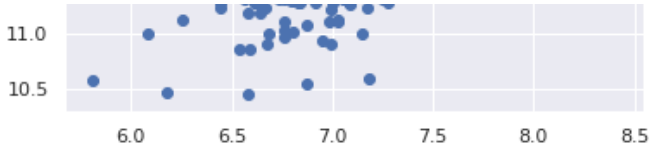
In the search for writing 'homoscedasticity' right at the first attempt

The best approach to test homoscedasticity for two metric variables is graphically. Departures from an equal dispersion are shown by such shapes as cones (small dispersion at one side of the graph, large dispersion at the opposite side) or diamonds (a large number of points at the center of the distribution).

Starting by 'SalePrice' and 'GrLivArea'...

In [30]:
```
#scatter plot
plt.scatter(df_train['GrLivArea'], df_train['SalePrice']);
```

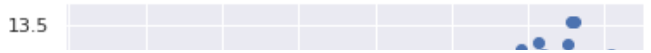




Older versions of this scatter plot (previous to log transformations), had a conic shape (go back and check 'Scatter plots between 'SalePrice' and correlated variables (move like Jagger style)'). As you can see, the current scatter plot doesn't have a conic shape anymore. That's the power of normality! Just by ensuring normality in some variables, we solved the homoscedasticity problem.

Now let's check 'SalePrice' with 'TotalBsmtSF'.

In [31]:
```
#scatter plot
plt.scatter(df_train[df_train['TotalBsmtSF']>0]['TotalBsmtSF'], df_tra
in[df_train['TotalBsmtSF']>0]['SalePrice']);
```



Did you find this Kernel useful?
Show your appreciation with an upvote

3952

Data

Data Sources

House Prices: Advanc...

House Prices: Advanced Regression Techniques

sampl... 1459 x 2
test.c... 1459 x 80
train.c... 1460 x 81
data_description.txt

Predict sales prices and practice feature engineering, RFs, and gradient boosting
Last Updated: 3 years ago

About this Competition

File descriptions

- **train.csv** - the training set
- **test.csv** - the test set
- **data_description.txt** - full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used here
- **sample_submission.csv** - a benchmark submission from a linear regression on year and month of sale, lot square footage, and number of bedrooms

Data fields

Here's a brief version of what you'll find in the data description file.

- **SalePrice** - the property's sale price in dollars. This is the target variable that you're trying to predict.
- **MSSubClass**: The building class
- **MSZoning**: The general zoning classification
- **LotFrontage**: Linear feet of street connected to property
- **LotArea**: Lot size in square feet
- **Street**: Type of road access
- **Alley**: Type of alley access
- **LotShape**: General shape of property
- **LandContour**: Flatness of the property
- **Utilities**: Type of utilities available

Run Info

Succeeded	True	Run Time	24.6 seconds
Exit Code	0	Queue Time	0 seconds
Docker Image Name	kaggle/python(Dockerfile)	Output Size	0
Timeout Exceeded	False	Used All Space	False
Failure Message			

Log

Download Log

```
Time  Line #  Log Message
2.7s       1  [NbConvertApp] Converting notebook __notebook__.ipynb to notebook
2.8s       2  [NbConvertApp] Executing notebook with kernel: python3
21.0s      3  [NbConvertApp] Writing 1064987 bytes to __notebook__.ipynb
23.3s      4  [NbConvertApp] Converting notebook __notebook__.ipynb to html
24.0s      5  [NbConvertApp] Support files will be in __results___files/
              [NbConvertApp] Making directory __results___files
24.0s      6  [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
24.0s      7  [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Making directory __results___files
              [NbConvertApp] Writing 347607 bytes to __results__.html
24.0s      8
24.0s     10  Complete. Exited with code 0.
```

Comments (695)

Sort by

All Comments | Hotness

Click here to enter a comment...

Nilan • Posted on Version 72 • 4 months ago • Options • Reply

3

Really interesting work although not much variation in the kind of visualization techniques used

Bo Gu • Posted on Version 70 • 6 months ago • Options • Reply

3

Thank you! Very helpful

ThomasLoeb... • Posted on Version 62 • a year ago • Options • Reply

76

Thanks for your interesting notebook! However, it seems to me that we need to be careful not to put too much weight on the insights that can be gained from an EDA like this. The main problem is that generally it only tells us about at most 2-dimensional relationships, which can be misleading.

For example, I would be very careful to decide whether a relationship is linear or quadratic/exponential based on scatterplots, because once we control for other variables, the effect may be very different (think of Simpson's Paradox). I'm not saying we shouldn't even look at these scatterplots; they might be useful to get ideas about which things to try out, but my point is that I wouldn't make a decision about how to model something based on these scatterplots. Instead, it may be better to inspect the residuals plots to detect any non-linear relationships. In fact, we know that the actual data-generating process is almost certainly non-linear, so it comes down to how well we can model this without increasing the variance of the estimate by too much. Except for small data sets, the best-performing predictive models will probably model this nonlinearity with things like regression splines. In practice, this problem can only be decided upon by trying different

models (with, in particular, different degree of regularization) to see which model has the lowest mean squared error on withheld data.

Likewise, I would use a more formal procedure for feature selection and be very hesitant to throw away any data. Even if multiple predictors are strongly correlated, it is hard to say intuitively what threshold to use. So I would include them all, and if the regression coefficients of the respective variables are unstable, I would see how much worse the model performs (again using withheld data) if I just use one. An even better solution would be to extract the principal components and use those as predictors instead.

Furthermore, while I agree that it was important to inspect the histogram of the dependent variable, and then to take the log because its distribution is asymmetric, I want to stress that multivariate normality of the data is NOT an assumption of regression analysis (which is probably the most natural approach for this problem) . Instead, it assumes that the error terms are normally distributed (i.e., y given X). Again, we need to inspect the residual plot to check this. Similarly, homoscedasticity refers to the conditional distribution of y given X, so again we need to look at the error terms if we want to plot this in two dimensions.

I know EDA often checks these things, but I am not convinced what insights we should be looking for that modeling couldn't tell us in a more objective way. I'm not saying we shouldn't take a look at our data at all before diving into modeling, but what I usually look for (in addition to whether the dependent variable is asymmetrically distributed) is unexpected patterns in the data – e.g., cutoff thresholds – that would be hard to detect otherwise, and maybe even indicate problems with the data. (I hope to add more points to this list over time...)

Koji • Posted on Version 63 • 10 months ago • Options • Reply 0

Hi Thomas - I was wondering about the dependent variable linearity assumption made in this post so thank you for confirming. Quick question - when do we care if the distribution of the dependent variable is not normal? Besides outliers, etc.

Samarth A... • Posted on Version 71 • 5 months ago • Options • Reply 0

That's a very valid point, Thomas. I will refer to your points in my kernel and mention you. Hope that's fine.

Gang Fang • Posted on Version 62 • a year ago • Options • Reply 51

I found this a very poorly written post either for data analysts or software engineers. I am totally confused why it earns so many upvotes.

Here to point out some problems:

1. Unintelligible section headings:

   One can hardly identify the high level structure of the post by skimming it but getting something like "3. Keep calm and work smart", "Getting hard core", etc.

2. Bad code:

   `k = 10 #number of variables for heatmap`, using comments to explain obscure code (a meaningless variable "k") is considered bad practice according to Robert Martin's Clean Code;

   `cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index`, redundant `['SalePrice'];`

   `cm = np.corrcoef(df_train[cols].values.T)`, this variable is named `cm` seems to me just because `corrmat` is used. Bad programming practice again.

3. Tendency towards wishful thinking:

   "Maybe they refer to agricultural area and that could explain the low price. I'm not sure about this but I'm quite confident that these two points are not representative of the typical case.". The fact is neither of them refers to agricultural area. The matter of fact is not important, what matters is the author demonstrates a dangerous tendency to thinking subjectively instead of paying attention to fact / data when his job is to, fundamentally speaking, get to the truth. This is especially dangerous as this notebook has been read by many;

   "According to this, there is a set of variables (e.g. 'PoolQC', 'MiscFeature', 'Alley', etc.) that we should delete. The point is: will we miss this data?". Same. The author didn't even bother to check if NA stands for something else (NA represent no pool / misc feature /

alley) before declaring that they are missing value and those features are of less importance to the prediction problem.

Overall, I think everybody should be careful paying attention to what is presented out there and try real hard to identify what is good and true and what not.

Luis Herrer... • Posted on Version 72 • 5 months ago • Options • Reply 6

Bad Code? This is not designed to be shipped into production but to be used in a competition. In a competition, it's more important to get a good score than write high quality code.

nichen • Posted on Version 72 • 4 months ago • Options • Reply 1

I love this post mostly because of its vividity, and it's really helpful to beginners like me. But thank you for pointing those problems :) I also find the redundant ['SalePrice'] intolerable...

Peng Liu • Posted on Version 72 • 4 months ago • Options • Reply 1

Code quality matters even though it is not for production.
I think it is a good thing to point out that the code could be improved, especially for newbies like me to learn from others' work.

Pedro Marceli... Kernel Author • Posted on Version 62 • a year ago • Options • Reply 19

Hey people! Thank you all for the support and feedback you've been giving to this kernel. I posted a new one, this time to solve the Titanic problem. I hope it helps you on your data science journey!

Manuel Spier... • Posted on Version 37 • a year ago • Options • Reply 37

Thank you for this kernel. I learned a lot how to visualize and read the data. But I think the missing data part is little bit misleading. I would not say that 15% is missing as many times the null indicates that this features is not present in the house. For example no pool or no fence. By just removing these features, you lose quite a lot of information. Better would be to create a new boolean feature 'has_pool' for example.

Pratik Sing... • Posted on Version 43 • a year ago • Options • Reply 0

Yeah , I too think the same.

almansur92 • Posted on Version 48 • a year ago • Options • Reply 0

Me too!

a year ago

This Comment was deleted.

czw123 • Posted on Version 62 • a year ago • Options • Reply 0

Me too

Erik Bruin • Posted on Version 63 • a year ago • Options • Reply 0

That's right Manuel. Also, saying 'FullBath'?? Really?' may seem funny, but in my XGBoost model Total_bathrooms turns out to be a very important predictor.....

Jackson H... • Posted on Version 70 • 6 months ago • Options • Reply 3

(Take this comment with a grain of salt, as I'm quite new to ML and EDA): It is more than likely that these nulls correspond to a lack of some feature (pool, fence) as you said, but at the end of the day, given the information from the data description, we cannot be certain if it's a lack of pool, or missing data.

If we choose to engineer a feature called no_pool, this feature would aggregate all data from both cases, and "create data" for the cases where there was actually missing data. I feel like techniques like that are a little sketchy, but I suppose that's just a judgement call you have to make when doing EDA. The other part that bothers me is that in you would have no way of propagating errors on the data that you created and have no way of describing the certainty of your final model. In my head it is safer to scrap that data. That said, maybe having a little subjectivity is more practical with these kinds of problems.

Yinghui Jiang • Posted on Version 66 • 9 months ago • Options • Reply

2

Thanks!

sahilsharmag... • Posted on Version 64 • 10 months ago • Options • Reply

1

Nice work !

Wei Chun Ch... • Posted on Version 66 • 9 months ago • Options • Reply

2

Love it! Though the story is funny, the content of this kernel is really decent! It mentioned many points that I usually forget. Thnkas for sharing!

Wall-E • Posted on Version 36 • a year ago • Options • Reply

20

Regarding the GarageX variables (like GarageCond, GarageType, etc.), *NA* doesn't mean a missing value but refers to *No Garage* , which is explained in the data_description.txt :

> GarageType: Garage location

```
   2Types    More than one type of garage
   Attchd    Attached to home
   Basment    Basement Garage
   BuiltIn    Built-In (Garage part of house - typically has room above
garage)
   CarPort    Car Port
   Detchd    Detached from home
   NA    No Garage
```

However, by default pandas.read_csv will interpret *NA* as missing data. In personal opinion, a better way to handle these GarageX variables is to take them as ordinal variables and use label encoding for them. Specially, *NA* is a meaningful level, which just means there is no garage in this house.

AjinkyaPatil • Posted on Version 66 • 9 months ago • Options • Reply -1

Best kernel ever.

Rokas • Posted on Version 47 • a year ago • Options • Reply 8

This is really not informative enough. In fact pretty useless for somebody without considerable background in statistics. Mostly you just show what python tools can do as apposed to explaining the process and chain of thoughts while doing data exploration. You print skewness and kurtosis, but instead of explaining what does that indicate you continue with that dating joke.

Lee S • Posted on Version 58 • a year ago • Options • Reply 39

I disagree. Some people, like myself find this very useful. This tutorial is great for those whom have read enough books and gained enough of the theory, thereby understanding the concepts of skewness etc, but just want to see a way to actually make a program to implement it.

X1 • Posted on Version 63 • 10 months ago • Options • Reply

1

There's serveral questions in comments.

1. NA not means missing, maybe means absent.
2. correlation should take absolute value.
3. Use residuals plots to check none linear.
4. feature selection should be carefully.

Charles • Posted on Version 26 • a year ago • Options • Reply

25

Thanks a lot for the share and explicit explaination, it really inspired a lot for a starter like me. But I got a question and can not get any satisfied answer from Google, the quetion is: What's the purpose of applying log to SalePrice and other features? As shown it can transfer disturibution to nearly normal disturibution but I am still confused with the benefit that can bring. Or what disadvantages does it have if don't convert it to normal disturibution? Thanks again.

cvonsteg • Posted on Version 31 • a year ago • Options • Reply

4

I think one simple answer is that many statistical analysis techniques are based on an assumption of normality. By normalizing the distribution, (in this case, using logs), we can conduct more intuitive analysis of the data. Heavily skewed or kurtosed distributions, don't allow for this.

If you're interested in finding out more, I'd suggest starting here: http://onlinestatbook.com/2/transformations/contents.html

No doubt somebody can give a more rigorous answer, but I hope this helps for a start!

ShaneKeller • Posted on Version 35 • a year ago • Options • Reply 7

I read a 2014 paper that claims that log transformations should be used sparingly, if at all, and that newer analytic methods that are not dependent on the distribution the data, such as generalized estimating equations (GEE), should be used instead. Anyone have thoughts on this?

Here's a quote from the paper (yikes!):

> Despite the common belief that the log transformation can decrease the
> variability of data and make data conform more closely to the normal distribution, this is usually not the case. Moreover, the results of standard statistical tests performed on log-transformed data are often not relevant for the original, non-transformed data.

Ruoying W... • Posted on Version 44 • a year ago • Options • Reply 6

From my understanding, another convenient thing about taking log is that it could reduce the impact of extremely large numbers. So the result is less likely to be driven by outliers, which is suitable in the case of housing prices. Talking about economic models, many times people like to estimate elasticities. For example, if log(y) = beta*log(x), then beta gives d log(y) / (d log(x)) = (dy/y)/(dx/x), which is the definition of elasticity.

10 months ...

This Comment was deleted.

Alex Radae... • Posted on Version 67 • 8 months ago • Options • Reply 0

Another thing I can think of is that "Submissions are evaluated on Root-Mean-Squared-Error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price." So it is better to predict the logarithm of the sale price. My xgboost model results improved a little when I just used log y instead of y.

zhuosir • Posted on Version 53 • a year ago • Options • Reply 1

thanks, I learned a lot from this kernel

cucalves • Posted on Version 53 • a year ago • Options • Reply 1

It's the best kernel I ever seen! Thanks a lot!!!

wujingcheng • Posted on Version 14 • 2 years ago • Options • Reply 30

Great job Pedro! But I have one question, the heat map of correlations gives us an overview of the linear relationships between two variables, it is good visualization. However, what if the variables actually have strong non-linear relationships? If we choose the features only according to the linear relationships, it may be misleading from time to time, what do you think?

Ilnur Shug... • Posted on Version 23 • 2 years ago • Options • Reply 8

I am not Pedro but I'll try to give an answer :)
By looking at scatter-plots you probably can detect some non-linear relationships. And then you can make non-linear transformation on given

variable to achieve the linear relationship between two variables. Therefore, you can choose this transformed feature.

Gaurav Sh... • Posted on Version 58 • a year ago • Options • Reply 0

Its an interesting Q&A. In what scenarios you will make a linear to non linear transformation and vice versa?

yang • Posted on Version 61 • a year ago • Options • Reply 0

there are multiple ways to test for linear relationship, for example you can try residual plot or standardized residual plot, or conduct a hypo testing.

Utkarsh Deep • Posted on Version 47 • a year ago • Options • Reply 1

Thank you for the kernel. As a beginner I learned a lot from this.

cast42 • Posted on Version 12 • 2 years ago • Options • Reply 22

To deal with the log of series that contain zero, you could transform with log1p = log(x+1)

Abhinanda... • Posted on Version 29 • a year ago • Options • Reply 1

I wonder if we can use modifications of Box-Cox transformations - inverse hyperbolic sine transformations, on those lines

Pratik Sing... • Posted on Version 43 • a year ago • Options • Reply 0

Yeah , that I hope is a better approach log(x+1)

Zoro Lee • Posted on Version 61 • a year ago • Options • Reply 2

No, I did it, but it did not work well.The figure is ugly. You can try it yourself.

Kazim Sanlav • Posted on Version 44 • a year ago • Options • Reply 1

Thanks a lot :)

vachar • Posted on Version 11 • 2 years ago • Options • Reply 16

This is definitely a **valuable resource**, thank you so much for writing and sharing this. It has helped me think in a systematic format like you outlined above.

I wanted to point that there's a line that you don't need in 27, and it can be removed

```
dftrain['HasBsmt'] = pd.Series(len(dftrain['TotalBsmtSF']), index=df_train.index)
```

The line after this does the entire job of creating a column HasBsmt with 0's to populate for each row in the data.

```
df_train['HasBsmt'] = 0
```

Alex Radae... • Posted on Version 67 • 8 months ago • Options • Reply 1

I agree, we can substitutes all the code in the block with only 1 line:

```
df_train['HasBsmt'] = (df_train['TotalBsmtSF'] =
0).astype(int)
```

Arateris • Posted on Version 31 • a year ago • Options • Reply

4

Hi, Thanks for this script, that's a very useful learning material.
For the correlation matrix analysis, would it be interesting to check the absolute value instead of the raw corrmat? This way, we can use the negative correlation (that may be hidden by the selection of the max values) and the heatmap get to be the darkest when the correlation is actually 0, which also help the visualization. Details here, of course.
Thanks again.

Lucie Jilko... • Posted on Version 44 • a year ago • Options • Reply

0

I agree with both -- a great notebook and the heatmap absolute values / limits. Another possibility could be to keep the negative correlations and use a diverging colorscale which can be done, for example, by setting the parameter *center* for seaborn.heatmap.

Andrew Yip • Posted on Version 36 • a year ago • Options • Reply

1

Relationships amongst variables a great analogy; cuteness aside, a neat EDA. Quoting Rand in an EDA, this has to be a First.

Chhavi Saluja • Posted on Version 18 • 2 years ago • Options • Reply

5

Hi Pedro,

Could you also explain the pre-processing and transformation steps that would be carried out on the test set.
For example get_dummies on test set might give more columns if there is a value which exist in training set and not in test set. I want to understand how information from training set is used to carry out transformation on test set.

Pratik Sing... • Posted on Version 43 • a year ago • Options • Reply 7

Hi Chhavi ,
Please combine both train and test. save the length for training set (to extract the training data at later stage.)Do the transformation on the combined data and then split them into train test using the training dataset length as index. This way both will have same number of columns.Please see my implementation in the Titanic Dataset.

https://www.kaggle.com/impratiksingh/titanic-survival-prediction

[2]

train*df = pd.read*csv('../input/train.csv')

test*df = pd.read*csv('../input/test.csv')

train*len=len(train*df)

train_len

[5]

dataset=pd.concat(objs=[train*df, test*df], axis=0).reset_index(drop=True)

[47]

ENCODING

dataset=pd.get_dummies(dataset)

[51]

Separate train dataset and test dataset

train = dataset[:train_len]

```
test = dataset[train_len:]

test.drop(labels=["Survived"],axis = 1,inplace=True)
```

Pooja Babu • Posted on Version 11 • 2 years ago • Options • Reply 5

Thanks for the great tutorial. Really inspired me to create a kernel. I have a question totally out of the subject matter here. When you said "Is this dataset from Chernobyl?", referring to the two variables , I lost you. I have looked up all the online content to interpret what you meant. I feel that I missed out a significant point as I failed to interpret. Any explanations would be great.

Travis Barr... • Posted on Version 13 • 2 years ago • Options • Reply 16

@poojababu I believe the intended analogy is that the radiation surrounding Chernobyl might cause an abnormal number of twins to be born in the area. As this data set seems to have an abnormal number of very similar features, the implication was that perhaps the data set is from Chernobyl (and hence the data twins).

Pooja Babu • Posted on Version 14 • 2 years ago • Options • Reply 1

Ahh! Now I understand.

Sergii Lutsan... • Posted on Version 5 • 2 years ago • Options • Reply 14

Great job, Pedro! By the way, a log transform of data set containing zero values can be easily handled by numpy.log1p()

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you Sergii! That's great advice. I'll add it to my 'toolbox' :)

Vamshi Kri... • Posted on Version 43 • a year ago • Options • Reply 0

Hey Sergii! Applying numpy.log1p() gives 'ValueError: cannot convert float NaN to integer'. Did you try using it?

Sergii Luts... • Posted on Version 45 • a year ago • Options • Reply 0

Hi Vamshi,
Following numpy's log1p() description, for real-valued input data types, log1p always returns real output. For each value that cannot be expressed as a real number or infinity, it yields **nan** and sets the invalid floating point error flag.
In short:
a=pd.np.nan
b=np.log1p(a) -> will return nan (with no ValueError).
Though,
int(a) -> will throw you exactly the ValueError.
I believe your issue is about converting of np.nan to int data type (most probably while doing data manipulation, feature generation, etc.)... There is no issue with np.log1p().
Hope that helps.

Johannes Vog... • Posted on Version 22 • 2 years ago • Options • Reply 1

Great job and thanks a lot for the very didactic outline to solving the problem! It helped me a lot :)

SNARC • Posted on Version 5 • 2 years ago • Options • Reply 8

Great job! It's so helpful! Just a little question, when you drew 'zoomed heatmap style', why not consider both positive and negative largest variables?

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 0

Because sometimes I'm stupid :) We really should consider both positive and negative largest variables. Thank you!

RamboYay • Posted on Version 5 • 2 years ago • Options • Reply 6

why all the features you have selected are numerical . How to do a feature selection for non-numerical or categorical

wikke • Posted on Version 6 • 2 years ago • Options • Reply 1

It is possible that this quick data examination process was a bit harsh for categorical variables.

In the "Relationship with categorical features" Section, boxplot is used for categorical features.

"factorplot" may helps, it has data,x,y,col,row,hue parameters, which could be used to draw a "whole map" of all relationship between non-numerical categorical features and Sales Price

Joeseph • Posted on Version 5 • 2 years ago • Options • Reply 6

谢谢你！ 棒棒哒！ Thank you so much ! you did a great job! so surprised to know that you can write Chinese characters!

Sarath Chan... • Posted on Version 5 • 2 years ago • Options • Reply 3

Hi Pedro,

Let me congratulate & thank you for working & sharing such a great piece of code before I pose you problems with my queries :P.

1. In your analogy ,how did you arrive at the success probability of 97.833% ?
2. In the case of TotalBsmnt, I am unable to get the log condition applied for the values less than zero.
3. Based on my understanding from the comment section below, we no need to check for normality but need to check do multi-variate analysis. But I think we need to check for normality even when we are applying OLS. If not can you provide me the conditions in which we can ignore these sanctity checks.

And btw, Aveda Kedevra is not a cool spell. Its a death spell :D

Any blogs that you follow for data science knowledge &
Last but not least , How many days it took you to compile this code ???

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 4

Hi Sarath! Thank you for your feedback.

Going straightforward:

1) Through a random process :)

2) I wouldn't follow my own approach. It's too empirical (and probably wrong). I'd recommend you to see the comment from Sergii Lutsanych (above). He gave an excellent hint!

3) Maybe the best option is to read Chapter 2.7.1 from Regression Modeling Strategies (Harrell, 2015).
4) Allow me to disagree, anything from Harry Potter is cool!

5) I like to read stuff from Sebastian Raschka. However, I spend most of my time reading books and the scikit-learn support material.

6) It took my 4 days. I adapted the Bill Gates' approach and went to an isolated beach in Portugal. That's something I'll definitely repeat in my next Kaggle competition.

Account fo... • Posted on Version 10 • 2 years ago • Options • Reply 1

Hi Pedro, that's some awesome work, taught me more than my professor did, and I study at Northwestern University, so that's no mean feat. Can you suggest some books to read. I have started Multivariate Data Analysis and Regression Modeling Strategies, reading your comments. Can you suggest me more?

nam choi • Posted on Version 5 • 2 years ago • Options • Reply 3

So nice!

AndreNurroh... • Posted on Version 5 • 2 years ago • Options • Reply 4

So thanks, I have learned from you so much. Great work.

SatishTyagi • Posted on Version 9 • 2 years ago • Options • Reply 1

Good analysis but I am curious to know about following.

'Amazing! If my love calculator is correct, our success probability is 97.834657%. I think we should meet again! Please, keep my number and give me a call if you're free next Friday.

See you in a while, crocodile!'

How did you get this success probability? Is it a random number to put?

James Smi... • Posted on Version 71 • 5 months ago • Options • Reply 0

Same question. Would love to know how this probability was calculated!

Jeroen Vuure... • Posted on Version 12 • 2 years ago • Options • Reply 0

Thank you for sharing this, very useful. I suggest using a Box-Cox transformation to correct for heteroscedasticity. Since the Box-Cox test gives a lambda very close to 0, it is justified to use a log-transformation.

ShaneKeller • Posted on Version 35 • a year ago • Options • Reply 0

http://scipy.github.io/devdocs/generated/scipy.stats.boxcox.html finds the optimal transformation for you, so you don't have to manually choose which type of transformation to use.

天线宝宝 • Posted on Version 5 • 2 years ago • Options • Reply 4

谢谢

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 5

别客气!

天线宝宝 • Posted on Version 5 • 2 years ago • Options • Reply 2

Your Chinese is so well.

summer_luo • Posted on Version 5 • 2 years ago • Options • Reply 0

中国的文化越来越国际化！！！

SonghuaHu • Posted on Version 12 • 2 years ago • Options • Reply 1

厉害了我的哥

huhu • Posted on Version 34 • a year ago • Options • Reply 0

Amazing to see Chinese here !

Ring Li • Posted on Version 35 • a year ago • Options • Reply 0

wonderful，kaggle里面有很多中国面孔呀！

Wall-E • Posted on Version 36 • a year ago • Options • Reply 1

大兄弟你好。

Mayank Jha • Posted on Version 11 • 2 years ago • Options • Reply 2

This has been of immense help. Thanks for sharing this :)

Panos • Posted on Version 6 • 2 years ago • Options • Reply

1

Great stuff! Learn a lot of tricks for Pandas. Thanks for this :)

mm2390 • Posted on Version 5 • 2 years ago • Options • Reply

1

Simple and easy to understand . Thank you.

Mirko76 • Posted on Version 5 • 2 years ago • Options • Reply

1

This Data Exploration really brings in a lot of useful ideas. Thanks!

rocha • Posted on Version 5 • 2 years ago • Options • Reply

1

Very excellent tutorial. I learn a lot from your visualization and explanation.

Kostas Tsanti... • Posted on Version 5 • 2 years ago • Options • Reply

1

Excellent work!! Thanks for sharing.

Jared Vasquez • Posted on Version 5 • 2 years ago • Options • Reply

1

This trick with the logarithmic transform is great, thanks for the tutorial!

Puneet Jindal • Posted on Version 5 • 2 years ago • Options • Reply

1

Great explanation

Aditya Mangal • Posted on Version 5 • 2 years ago • Options • Reply 1

Really helpful notebook. Thanks!!

Amit Basuri • Posted on Version 5 • 2 years ago • Options • Reply 1

Great

Steven Glem... • Posted on Version 5 • 2 years ago • Options • Reply 1

Great work, thanks for sharing

Fantastic Ho... • Posted on Version 5 • 2 years ago • Options • Reply 1

good job ! the procedure of processing data is awesome!

Luke Tonin • Posted on Version 5 • 2 years ago • Options • Reply 1

Great work. Thanks!

jsolis • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you! Definitely insightful.

TwentyTwo • Posted on Version 5 • 2 years ago • Options • Reply 1

Thanks for the effort.
A very nice work!

Pandey • Posted on Version 5 • 2 years ago • Options • Reply

1

Such Kernel, Much wow!!

Gosuddin Sid... • Posted on Version 5 • 2 years ago • Options • Reply

1

Story spun EDA!

Евгений Бори... • Posted on Version 5 • 2 years ago • Options • Reply

1

Good work!

puneeth019 • Posted on Version 5 • 2 years ago • Options • Reply

1

Awesome stuff man! Learnt a lot.

AndrewK. • Posted on Version 5 • 2 years ago • Options • Reply

1

awesome !

Sivaji • Posted on Version 6 • 2 years ago • Options • Reply

2

Very good and well documented analysis.

Anuradha • Posted on Version 5 • 2 years ago • Options • Reply

2

Pedro, such a wonderful work and well explained. Do you have the prediction code also done by you for this work

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you for your feedback, Anuradha. I just focused on data analysis, so I don't have the prediction code.

Prateek Gupta • Posted on Version 5 • 2 years ago • Options • Reply 1

Nice analysis Pedro. Many thanks.

chen shao • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you ! It's helpful for me.

AmjadAli • Posted on Version 5 • 2 years ago • Options • Reply 1

Great work!

nitien • Posted on Version 5 • 2 years ago • Options • Reply 1

Great work , Learned a lot . Thanks !

Dylan • Posted on Version 5 • 2 years ago • Options • Reply 1

Thanks!

rydn • Posted on Version 5 • 2 years ago • Options • Reply 1

Thanks :)

Edward • Posted on Version 5 • 2 years ago • Options • Reply 1

Great work!

jiang peng • Posted on Version 5 • 2 years ago • Options • Reply 1

Great! Thank you!

zell.fell • Posted on Version 5 • 2 years ago • Options • Reply 1

It's so helpful !Thank you!

EliThomas • Posted on Version 5 • 2 years ago • Options • Reply 1

Great work!- I was wondering if you can share the excel spreadsheet you created initially to determine the importance of variables. Wanted to see your thinking process behind choosing the final 4 to analyze.
Because it was spot on!

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you! I still don't know the best way to share documents in Kaggle, but in the meanwhile you can access my spreadsheet through this Google Sheets link. Maybe in the column 'Comments' you can find a little bit of my thinking process. However, I'd say that the secret sauce is to have

experience in real estate. As Einstein said, 'the only source of knowledge is experience'!

Prabhat D... • Posted on Version 5 • 2 years ago • Options • Reply 1

Why is feature 'functional' of type numerical, shouldn't it be categorical?

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 0

Yes, it should! I suppose I just read the variable description ('Home functionality rating') and assumed a numerical rating. Thanks!

Zidong Yang • Posted on Version 5 • 2 years ago • Options • Reply 1

Great and Helpful~

Jae-Seung Lee • Posted on Version 5 • 2 years ago • Options • Reply 1

Great! I've learned a lot from this. Thank you very much!

Md Sadik Hu... • Posted on Version 5 • 2 years ago • Options • Reply 2

good work..... I am beginner , It help me to understand feature exploration and transformation. Thanks

江南消夏 • Posted on Version 5 • 2 years ago • Options • Reply 1

It's a great tutorial, Thank you very much for sharing!

RichardLo • Posted on Version 5 • 2 years ago • Options • Reply 1

Wonderful teaching! Great thanks!

KendallFortn... • Posted on Version 5 • 2 years ago • Options • Reply 1

KendallFortn... • Posted on Version 5 • 2 years ago • Options • Reply 1

Great work!

Eva • Posted on Version 5 • 2 years ago • Options • Reply 1

Your tutorial is so great!
One question, will you consider to use robust Scaler instead of standard scaler for Univariate analysis?
From some other resources, we've learned Robust Scaler is better to against outliers.
Just curious.
Thank you!

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 2

Thank you :) I totally agree with you. Robust methods are a smart (and probably better) alternative to throwing out outliers. To be honest, I explored RANSAC models but didn't have time to read about how to choose an appropriate value for the inlier threshold. So, I just went for an 'old school' method. Probably, using robust methods is a good challenge for the next competition :)

chelo • Posted on Version 5 • 2 years ago • Options • Reply 1

great analysis, thanks!

chelo • Posted on Version 5 • 2 years ago • Options • Reply 1

Great analysis, thanks!

chelo • Posted on Version 5 • 2 years ago • Options • Reply 1

Great analysis, thanks!

chelo • Posted on Version 5 • 2 years ago • Options • Reply 1

Great analysis, thanks!

Phongdk • Posted on Version 5 • 2 years ago • Options • Reply 1

Great tutorial :D

amrutshintre • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank You @Pedro Marcelino for such detailed analysis. Cheers!

nemo • Posted on Version 5 • 2 years ago • Options • Reply 1

Thanks a lot

youngfire • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you very much! Just some little question,...should the text data be normalized as well? And, after prediction, is there any transformation for SalePrice of test data (because the SalePrice of train data is not actual after normalized)?

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 0

Thank you! I think this post from Sebastian answers to your questions (with a better explanation than I could ever give). Stay awesome!

youngfire • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you, it really helps!

Helena • Posted on Version 5 • 2 years ago • Options • Reply 1

Hey, this is great - really helped me out with data exploration! The jokes are not too bad either.

villefranche • Posted on Version 5 • 2 years ago • Options • Reply 1

Great job! Very comprehensive and helpful, thanks.

HaiNu Super... • Posted on Version 5 • 2 years ago • Options • Reply 2

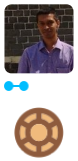
Very Nice ! ! !

dsds • Posted on Version 5 • 2 years ago • Options • Reply 1

Thank you! you are so wise! i need to do more work in this project and to improve my ability in research.

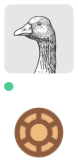
Jokus • Posted on Version 5 • 2 years ago • Options • Reply 1

thanks man



MaheshKulka... • Posted on Version 5 • 2 years ago • Options • Reply 1

very nice work Pedro!



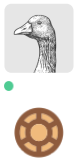
dsds • Posted on Version 5 • 2 years ago • Options • Reply 1

Great! i have studied more in it. but it gives me a question, why not use heatmap() early, so we can get related information about 'SalePrice' and to build our features.

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 0

Yes, we could do that and it would accelerate our analysis!

I just preferred to look at data in a more 'handcrafted' way first because it helps me develop some intuition. This gives me a critical opinion about what's going on, improving my ability to identify and correct errors in my model.



Jelly • Posted on Version 5 • 2 years ago • Options • Reply 2

Perfact!

Jordan Vonde... • Posted on Version 5 • 2 years ago • Options • Reply

2

Thanks so much! Please write more, I'll read your grocery lists.

ZeZhong • Posted on Version 5 • 2 years ago • Options • Reply

2

Very good tutorial! Thank you!

dxfang • Posted on Version 5 • 2 years ago • Options • Reply

1

so cool

chenhui • Posted on Version 5 • 2 years ago • Options • Reply

2

very good! I have learned so much.

蜡笔小Ke • Posted on Version 5 • 2 years ago • Options • Reply

2

ty for share,great work!

Shih-Wei Cha... • Posted on Version 5 • 2 years ago • Options • Reply

2

Nice work!

xiia • Posted on Version 5 • 2 years ago • Options • Reply

2

Salute !

vivic • Posted on Version 5 • 2 years ago • Options • Reply 2

Great help! So grateful for sharing your knowledge. Thank you.

fish • Posted on Version 5 • 2 years ago • Options • Reply 2

Great work!

MartianCoder • Posted on Version 5 • 2 years ago • Options • Reply 2

Thanks!

Jacob Humber • Posted on Version 5 • 2 years ago • Options • Reply 2

This is a great Kernel, however, I do want to point out one thing. The kernel spent a some time discussing homoskedasticity and to a lesser extent serial correlation. Given the goals of the competition, one need not speed any time considering these issues.

To understand why, note that when considering a simple linear regression (i.e. OLS), parameter estimates are unaffected by heteroskedasticity or serial correlation. All parameters are therefore unbiased, however, in the case of serial correlation estimators are less efficient (i.e. need more obs until it approaches population) . Therefore if we have enough data points, you'll be ok and better off spending your time considering different problems. It is important to know that while predictions will be unaffected, inference will be affected if either issue is present.

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 6

Thanks for your feedback Jacob! I agree with you, this kernel is not the most effective for the goals of the competition. Please, take this kernel as my intention to develop a general framework for data examination that could work for future reference in related projects :)

Vivek Srinivа... • Posted on Version 5 • 2 years ago • Options • Reply 2

Great Notebook !!!!!
Should we test the normality of independent variables? Usually we do it for dependent variable. Just curious to know how checking normality of independent variables helps in models building?

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 2

That's a good question! I would say that all depends on the kind of analysis that you're going to do after.

According to Hair et al. (2013), normality is one of the assumptions that we should test when we are applying multivariate analysis (the others assumptions are homoscedasticity, linearity and absence of correlated errors). Multivariate methods assume multivariate normality, which means that the individual variables (dependent and independent) are normal in a univariate sense and that their combinations are also normal. In most cases, assessing and achieving univariate normality for all variables is sufficient to guarantee multivariate normality. However, it is not *totally* guaranteed. A multivariate normal variable is also univariate normal, but the reverse is not necessarily true. In a strict sense, if we really want to guarantee multivariate normality, we have to apply specialized tests (which can make us sweat).

One important remark: as far as I know, normality is only a requirement for certain statistical tests and hypothesis tests (e.g. F and t statistics). If you're just thinking about training a linear regression model, you just need to verify linearity and additivity (see Chapter 2.7.1 in Harrell, F., 2015, *Regression Modeling Strategies*, Springer).

Nirai P • Posted on Version 5 • 2 years ago • Options • Reply 0

Great tutorial, you should write a series of tutorials for beginners. Have you done that already? I'll go looking for your other tutorials now.

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 0

That's a great compliment! I don't have any tutorials, but thanks for the motivation :)

Ades • Posted on Version 5 • 2 years ago • Options • Reply 2

@Pedro,

You really have a very good presentation of the work and I also love your analogy of a lady to explore the data. It is the very best analogy.
I will like to know why you choose to eliminate missing data above 15%? Is there a rule for that? Why not any other percentage?
Thanks

Pedro Mar... Kernel Author • Posted on Version 5 • 2 years ago • Options • Reply 0

Thanks Ades! It's a rule of thumb that I took from Hair et al. (2013). I wouldn't say that we should apply it blindly, but in this case that's what I did.

NataliaG • Posted on Version 5 • 2 years ago • Options • Reply 2

This is excellent, I felt both entertained and enlightened :P thank you!

Pandey • Posted on Version 5 • 2 years ago • Options • Reply -1

dsds • Posted on Version 5 • 2 years ago • Options • Reply 0

Pandey • Posted on Version 5 • 2 years ago • Options • Reply 0

Pandey • Posted on Version 5 • 2 years ago • Options • Reply 0

Pandey • Posted on Version 5 • 2 years ago • Options • Reply 0

Jared Vasquez • Posted on Version 5 • 2 years ago • Options • Reply 0

Panos • Posted on Version 6 • 2 years ago • Options • Reply 0

akhil pannala • Posted on Version 6 • 2 years ago • Options • Reply 0

akhil pannala • Posted on Version 6 • 2 years ago • Options • Reply 0

Really great stuff. I have few questions.

What are the important variables do we need to use for regression analysis?
Are these variables OverallQual, YearBuilt, TotalBsmtSF, GrLivArea enough for regression analysis?
What about the categorical variables?
Thanks.

wikke • Posted on Version 6 • 2 years ago • Options • Reply 0

> It is possible that this quick data examination process was a bit harsh for categorical variables.

In the "Relationship with categorical features" Section, boxplot is used for categorical features.

"factorplot" may helps, it has data,x,y,col,row,hue parameters, which could be used to draw a "whole map" of all relationship between non-numerical categorical features and Sales Price

Will Cannif... • Posted on Version 47 • a year ago • Options • Reply 0

As wikke said above, he had a look at some of the categorical variables in the boxplot, but didn't really come back to them. You can use the categorical variables in regression, and may add some predictive power through the creation of parallel planes. The alternative is what he did at the end which is to translate those categories that you want to use into dummy variables so that each group is represented through binary outcomes across multiple rows. Either way the information can be used in the regression model. More info on dummy variables: https://www.moresteam.com/WhitePapers/download/dummy-variables.pdf

Jacob Rafati • Posted on Version 6 • 2 years ago • Options • Reply 0

I am really enjoying this tutorial and I should say that I learned a lot in last hour :) Thanks for your effort. My background in Statistics is not great. Would you also possibly suggest some good resources for data analysis; quite similar to this tutorial material? I appreciate. Jacob

[Deleted User] • Posted on Version 6 • 2 years ago • Options • Reply 0

have to say it is such a wonderful job

[Deleted User] • Posted on Version 6 • 2 years ago • Options • Reply 0

Bill Arbuckle • Posted on Version 7 • 2 years ago • Options • Reply 0

Nice analysis and well documented. Will need to go through this again. Thanks!

thesqlspot • Posted on Version 7 • 2 years ago • Options • Reply 0

One of the most comprehensive nb I've ever seen. Nicely done.

aongao • Posted on Version 7 • 2 years ago • Options • Reply 0

Good work. It's useful to me. Thanks a lot.

Reynold • Posted on Version 7 • 2 years ago • Options • Reply 0

Very helpful and easy to follow!

arvind raj • Posted on Version 7 • 2 years ago • Options • Reply 0

Really nice tutorial. can we use sklearn.feature_selection for the above question ?

Mark • Posted on Version 9 • 2 years ago • Options • Reply 0

Great notebook.....I have learned a lot. Thank you.

Chandrakant ... • Posted on Version 9 • 2 years ago • Options • Reply 0

Great Work!!

Gregory Mott... • Posted on Version 9 • 2 years ago • Options • Reply 0

Very good!

Amir Abu Jan... • Posted on Version 9 • 2 years ago • Options • Reply 0

This is not only amazing but very entertaining as well. Many thanks for spending allot of time on this valuable kernel. It is very useful. :)

Naeem Khos... • Posted on Version 10 • 2 years ago • Options • Reply 0

Thanks Pedro. This is a very informative analysis. I enjoyed it.
I would like to add a sentence about removing features with missing data. In the data_description.txt file that comes with the dataset, missing values for PoolQC and Alley are equivalent to "no pool" and "no alley access", respectively. I think another approach could be transforming each one of them into a one-column binary variable.

For example:
HasAlleyAccess (True/False)

HasPool (True/False)

TomasBielskis • Posted on Version 10 • 2 years ago • Options • Reply 0

Nice and clear! Thank you very much (:

shaurya • Posted on Version 10 • 2 years ago • Options • Reply 0

Pedro, Nice work but I thought why didn't you see the correlation between categorical feature? And through heat map, some more features look like twin brothers, So, is it better to drop them also?

shaurya • Posted on Version 10 • 2 years ago • Options • Reply 0

shaurya • Posted on Version 10 • 2 years ago • Options • Reply 0

Trueman • Posted on Version 11 • 2 years ago • Options • Reply 0

Thank u for your great instruction!

Pondel • Posted on Version 11 • 2 years ago • Options • Reply 0

Thanks for sharing!

walkerw62 • Posted on Version 11 • 2 years ago • Options • Reply 0

thanks

Lucky Pan • Posted on Version 11 • 2 years ago • Options • Reply 0

You really did a good job.Some part of your analysis has been translated into Chinese.

Lucky Pan • Posted on Version 11 • 2 years ago • Options • Reply 0

You really did a good job.Some part of your analysis has been translated into Chinese.

Lucky Pan • Posted on Version 11 • 2 years ago • Options • Reply 0

You really did a good job.Some part of your analysis has been translated into Chinese.

ZhangBorui • Posted on Version 11 • 2 years ago • Options • Reply 0

Wow, very educational, I learnt a lot from this. Thanks!

Emma Ren • Posted on Version 11 • 2 years ago • Options • Reply 0

This is really helpful for starters!

hushenglang • Posted on Version 11 • 2 years ago • Options • Reply 0

awesome analysis, give me a lot of intuition and techniques, thanks very much for sharing.

Birger • Posted on Version 11 • 2 years ago • Options • Reply 0

Great work, thanks for sharing

Victor Evang... • Posted on Version 11 • 2 years ago • Options • Reply 0

Great notebook! Thanks a lot!

Victor Evang... • Posted on Version 11 • 2 years ago • Options • Reply 0

Great notebook! Thanks a lot!

Rahul Dubey • Posted on Version 12 • 2 years ago • Options • Reply 0

Nice work. This was helpful
Congrats!!

Stephen Wist • Posted on Version 12 • 2 years ago • Options • Reply 0

Simply captivating-a great example of making a kernel fun to read!

Valery Piashc... • Posted on Version 12 • 2 years ago • Options • Reply 0

Thanks!

Bill Chang • Posted on Version 12 • 2 years ago • Options • Reply 0

This is really helpful. Thanks for sharing!

Bill Chang • Posted on Version 12 • 2 years ago • Options • Reply 0

This is really helpful. Thanks for sharing!

avinashkaitha • Posted on Version 12 • 2 years ago • Options • Reply 0

Very Comprehensive... Thanks... :)

JuanMendez • Posted on Version 12 • 2 years ago • Options • Reply 0

Amazing overview. Thank you so much!

cast42 • Posted on Version 12 • 2 years ago • Options • Reply 0

Fabio Grassi • Posted on Version 12 • 2 years ago • Options • Reply 0

Thanks for this very user friendly notebook. I have one question about categorical conversion: is there any reason why you are not using the drop_first=True option?

Fabio Grassi • Posted on Version 12 • 2 years ago • Options • Reply 0

Jess Sze • Posted on Version 13 • 2 years ago • Options • Reply 0

Thanks for sharing!

AdamChang • Posted on Version 13 • 2 years ago • Options • Reply 0

Thank you for sharing many great idea!
I learn a lot from your tutorial!
Thanks a lot!

PaulPerrone • Posted on Version 13 • 2 years ago • Options • Reply 0

Great work: thanks for the script!

Jake Yang • Posted on Version 13 • 2 years ago • Options • Reply 0

Great work! Thanks for sharing.

Tommy Jiang • Posted on Version 13 • 2 years ago • Options • Reply 0

Thanks for the excellent kernel!

MKDATA • Posted on Version 13 • 2 years ago • Options • Reply 0

What a story teller!

Pooja Sharma • Posted on Version 14 • 2 years ago • Options • Reply 0

Awesome work. This is so well documented. Thank you for sharing!

Jayashree • Posted on Version 14 • 2 years ago • Options • Reply 0

Can you please explain this portion of code

standardizing data

salepricescaled = StandardScaler().fittransform(dftrain['SalePrice'][:,np.newaxis]); lowrange = salepricescaled[salepricescaled[:,0].argsort()][:10]
highrange= salepricescaled[salepricescaled[:,0].argsort()][-10:] print('outer range (low) of the distribution:') print(lowrange)
print('\nouter range (high) of the distribution:')
print(high_range)

why do we calculate this?

Luiz Gustavo ... • Posted on Version 14 • 2 years ago • Options • Reply 0

Thanks for sharing! Great stuff!

liuchu • Posted on Version 14 • 2 years ago • Options • Reply 0

Thank you for your resource,it really helps , and I learned a lot!

nicsli • Posted on Version 14 • 2 years ago • Options • Reply 0

So great! Thank you for sharing!

Elton Paes • Posted on Version 16 • 2 years ago • Options • Reply 0

Thanks for that lecture in data analysis!

Tokuhana • Posted on Version 16 • 2 years ago • Options • Reply 0

Thanks man for sharing this great kernel!

Pamela Augu... • Posted on Version 16 • 2 years ago • Options • Reply 0

Thanks for sharing this! I love the analogies, data visualization, analysis and the great way you told the interesting story!

Nick Buchele... • Posted on Version 16 • 2 years ago • Options • Reply 0

Why must the labels be normally distributed? It's the error residuals distribution which matters in choosing the estimator...what does the normalcy of the training labels tell us?

Chhavi Saluja • Posted on Version 16 • 2 years ago • Options • Reply 0

Congratulations on good work Pedro. I have hit the ground running to learn Data Science. This was of immense help. Thanks again.

Kushal Chau... • Posted on Version 18 • 2 years ago • Options • Reply 0

A very comprehensive data analysis. Thank You!

Allwyn • Posted on Version 18 • 2 years ago • Options • Reply 0

Incredible work!!

Aditya Soni • Posted on Version 18 • 2 years ago • Options • Reply 0

Incredible work!!

Aditya Soni • Posted on Version 18 • 2 years ago • Options • Reply 0

ShaunakCha... • Posted on Version 18 • 2 years ago • Options • Reply 0

Pedro Sir!
Such a great Tutorial , a very valuable resource for newbies like me but I have a small doubt when I am making the correlation matrix for a value greater than 11(the second corr matrix) the matrix is quite distorted with entirely gray region for some entries.
Can you please help me with the same.
Thanks in Advance.!!! :)

ShaunakCha... • Posted on Version 18 • 2 years ago • Options • Reply 0

salepriceʃcaled = StandardScaler().fittransform(df_train['SalePrice'][:,np.newaxis]);

Can anyone please explain what's happening with this line?

raghuveer ... • Posted on Version 30 • a year ago • Options • Reply 0

np.newaxis raise the dimensionality of the array by 1. Since fit_transform needs 2d array to do standardization you pass np.newaxis argument to the indexing.

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler.fit_transform

look at the arguments of the method [X] in this case.. it is a 2 d array.

Run following to see the dimensions and shape of the indexing:

```
df_train['SalePrice'].shape

df_train['SalePrice'][:, np.newaxis].shape

df_train['SalePrice'].ndim

df_train['SalePrice'][:, np.newaxis].ndim
```

np.newaxis reference:
https://stackoverflow.com/questions/29241056/the-use-of-numpy-newaxis

Dan Xu • Posted on Version 18 • 2 years ago • Options • Reply 0

Why you guys so awesome

Matheus Barr... • Posted on Version 19 • 2 years ago • Options • Reply 0

Very good explanation. Thank you.

Philip Corr • Posted on Version 19 • 2 years ago • Options • Reply 0

Great process for exploring and cleaning the data. Quick question, you mention that in big samples (>200 observations) normality is not such an issue. Why is this? Thanks!

DitlevKiers... • Posted on Version 23 • 2 years ago • Options • Reply 0

In large samples (n -> infinity), the parameter estimates from e.g. a multivariate linear regression are asymptotically normally distributed whenever y is homoskedastic. Therefore, you don't need to assume normality of y to e.g. test whether a variable affects y (i.e. whether the parameter estimate is significantly different from zero). Thus, in large samples, normality of y isn't really an issue.

Rohith Raj R • Posted on Version 19 • 2 years ago • Options • Reply 0

This is what i call Art !!!

awtotty • Posted on Version 19 • 2 years ago • Options • Reply 0

This is a really great write-up! Very helpful. You have a great style.

awtotty • Posted on Version 19 • 2 years ago • Options • Reply 0

awtotty • Posted on Version 19 • 2 years ago • Options • Reply 0

Gaurav Agar... • Posted on Version 19 • 2 years ago • Options • Reply 0

Thank you for sharing this. I want to ask why can we not start directly from plotting the correlation matrix because that is also giving us an idea of what we found out before so what's the need of doing it when the matrix can easily tell us that?

Ranjeet Sing... • Posted on Version 19 • 2 years ago • Options • Reply 0

Nice analysis specially missing values part.

Matthew Mid... • Posted on Version 20 • 2 years ago • Options • Reply 0

Good notebook for learning, thanks!

[Deleted User] • Posted on Version 21 • 2 years ago • Options • Reply 0

Thanks for sharing! :)

Shigeaki Ima... • Posted on Version 21 • 2 years ago • Options • Reply 0

Great job!

Angry Bird • Posted on Version 21 • 2 years ago • Options • Reply 0

Amazing! This is the logic flow I'm looking for! Thanks for sharing your notebook.

DonglingLi • Posted on Version 21 • 2 years ago • Options • Reply 0

Such a funny guy.

Abhijith Man... • Posted on Version 21 • 2 years ago • Options • Reply 0

Really great data exploration. I can clearly see what you're thinking when you go about your variables

Abhijith Man... • Posted on Version 21 • 2 years ago • Options • Reply 0

Ragul Ram J • Posted on Version 22 • 2 years ago • Options • Reply 0

Really comprehensive. Im Trying to do something very similar to this with R for an In-house tool. Got a lot of pointers from here! Thanks!

Issam Mahm... • Posted on Version 22 • 2 years ago • Options • Reply 0

thank you, great analysis.

loan calculato... • Posted on Version 22 • 2 years ago • Options • Reply 0

Great job for publishing such a beneficial comprehensive and helpful article.
Here is my new guide for learners loan calculators excel .

Thanks for putting this together.
Keep up the awesome work!
Thanks

suprita • Posted on Version 23 • 2 years ago • Options • Reply 0

I am unable to see the codes/explanations. Where can I find them ?

Jeremy Seibe... • Posted on Version 23 • 2 years ago • Options • Reply 0

Awesome Notebook! I really needed the template that you laid out to get started. Again man awesome job!

McLearner • Posted on Version 23 • 2 years ago • Options • Reply 0

Thank you for this nice introduction to data exploration. I suppose the age of house at sell time could be feature engineered (yearsold - yearbuilt).

SaadAljadhai • Posted on Version 24 • 2 years ago • Options • Reply 0

The writing & analysis is top notch, thanks!

AugustinPott... • Posted on Version 24 • 2 years ago • Options • Reply 0

Thank you for this very useful notebook 😊

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

Hi, I have a question.
If I trained my model with log variables (for example, I used log1p('GrLivArea')), should I also log my test variables when I want to predict?

Charles • Posted on Version 26 • a year ago • Options • Reply 0

Of course you should, so as to keep your train data and test data in the same disturibution, as one of the hypothesis is train data and test data is at the same disturibution and what a model learnt is the disturibution, so to get precise prediction, your test data should get the same transformation.

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

HaoChien • Posted on Version 24 • a year ago • Options • Reply 0

Nitesh Tiwari • Posted on Version 26 • a year ago • Options • Reply 0

thanks for sharing

Utkan Cando... • Posted on Version 26 • a year ago • Options • Reply 0

Utkan Cando... • Posted on Version 26 • a year ago • Options • Reply 0

Utkan Cando... • Posted on Version 26 • a year ago • Options • Reply 0

Very impressive, thanks for sharing!

A quick comment: I believe in general it would better to sort the absolute values of correlations while constructing the 'zoomed correlation matrix' though. So that we do not miss the strongly negatively correlated features.

Larissa Ferna... • Posted on Version 26 • a year ago • Options • Reply 0

I have no background in coding, statistics or computer science, but had to write a paper and do some statistical analyses on a few datasets and it had to be in Python. I was able to follow this entire method to the "t" and now I have all the analyses to write my paper. Thank you so much. Not only extremely helpful, but also fun to read.

wangyan2 • Posted on Version 26 • a year ago • Options • Reply 0

Great job! Thanks for sharing.

William Droz • Posted on Version 26 • a year ago • Options • Reply 0

Nice in-dept explanations.

Mike Munkyu... • Posted on Version 26 • a year ago • Options • Reply 0

Amazing tutorial, learned so much. Thank you Pedro!

richarde • Posted on Version 26 • a year ago • Options • Reply 0

Good work Pedro, like you sense of humour too

Richard726 • Posted on Version 26 • a year ago • Options • Reply 0

wonderful job !

Jack • Posted on Version 27 • a year ago • Options • Reply 0

Thanks for putting this together. It's been very helpful.

Jack • Posted on Version 27 • a year ago • Options • Reply 0

For the correlation matrix is it not the white squares that show high positive correlation? In your notes, you say "At first sight, there are two **red** colored squares that get my attention. The first one refers to the 'TotalBsmtSF' and '1stFlrSF' variables, and the second one refers to the 'GarageX' variables.". But I cannot see any red squares where they intersect.

Olivier • Posted on Version 29 • a year ago • Options • Reply 0

The recent version of seaborn change the default colors of the heatmap. It was like that: https://elitedatascience.com/wp-content/uploads/2017/04/seaborn-heatmap-example.png

Natalia Motyli... • Posted on Version 29 • a year ago • Options • Reply 0

Mister, You officialy made ma day with your sense of humor :)

SIlvestreMari... • Posted on Version 29 • a year ago • Options • Reply 0

Excellent! Many thanks for sharing this. It is useful and fun :)

Alexander Te... • Posted on Version 29 • a year ago • Options • Reply 0

Thanks for this tutorial :)

Mustafa Yurt... • Posted on Version 29 • a year ago • Options • Reply 0

Great job, thanks.

Robel Denu • Posted on Version 29 • a year ago • Options • Reply 0

Hats off to you

Pedro Q • Posted on Version 29 • a year ago • Options • Reply 0

Really good!! And you're a Pedro as well! Great Job!

johnnyjana7... • Posted on Version 29 • a year ago • Options • Reply 0

Great job! Thanks:) I learned a lot from that. From the data plot section to feature scaling, every detail is so clean. What a great work!

林湧森 (Dyson... • Posted on Version 29 • a year ago • Options • Reply 0

I learned a lot from this notebook. Thanks!

António Leite • Posted on Version 29 • a year ago • Options • Reply 0

Ok, enough of Rorschach test for now. És o maior :D Bom trabalho pmarcelino.

António Leite • Posted on Version 29 • a year ago • Options • Reply 0

António Leite • Posted on Version 29 • a year ago • Options • Reply 0

Francisco de ... • Posted on Version 30 • a year ago • Options • Reply 0

Very nice tutorial buddy! Carrega benfica

UrvangPatel • Posted on Version 30 • a year ago • Options • Reply 0

Thank you for sharing this notebook! I am new to data science and I have a question. Can we plot multiple regplots in grid just like you plotted multiple scatter plots while checking the correlation?

holly lai • Posted on Version 30 • a year ago • Options • Reply 0

This is so intuitive and helpful. Thank you so much!

lllx • Posted on Version 30 • a year ago • Options • Reply 0

Thanks, much help for begginer like me.

lllx • Posted on Version 30 • a year ago • Options • Reply 0

Thanks, much help for begginer like me.

TatsianaMiha... • Posted on Version 30 • a year ago • Options • Reply 0

Thank you ! Great job (and very nice:))

LeonPaul • Posted on Version 31 • a year ago • Options • Reply 0

Great exploration.

Marvi • Posted on Version 34 • a year ago • Options • Reply 0

Nice piece of work ! mainly I want to mention that you have really good sense of humor !

ChrisHeff • Posted on Version 34 • a year ago • Options • Reply 0

"+ 1" for referencing Alan Kay's "a change in perspective is worth 80 IQ points" I like his "Our job is to remind us that there are more contexts than the one that we're in — the one that we think is reality." Thanks for the humorous context of your notebook.

hemant11000... • Posted on Version 35 • a year ago • Options • Reply 0

good kernal

Pranali Jalgam • Posted on Version 35 • a year ago • Options • Reply 0

Why not use np.log1p(df_train['TotalBsmtSF']), gives the same result in a cpuple of lines.

Kumaresan ... • Posted on Version 35 • a year ago • Options • Reply 0

Thank you. Great work.

Varadarajan ... • Posted on Version 35 • a year ago • Options • Reply 0

Great explanation! Small comment though. In your last section, where you convert categorical variables into dummy variables, the command df*train = pd.get*dummies(df_train) would result in the dataset being perfectly multi-collinear. You would have to remove a column for each one of the categorical variable converted to dummy data! Cheers.

Abhishek • Posted on Version 63 • 10 months ago • Options • Reply 0

Do you have any shorter way to handle dummy variable trap?

Antonis Stellas • Posted on Version 35 • a year ago • Options • Reply 0

Thank you a lot Pedro that was really helpful (also funny :P).

I was wondering, why our data analysis in the correlation study, was limited in linear correlation between the parameters.
I mean , are there some other easy methods to find non-linear but important parameters??? (any suggestion?)

Thank you again!

LAONBAI • Posted on Version 35 • a year ago • Options • Reply 0

Thanks for share the method of data analysis.

M Siebert • Posted on Version 35 • a year ago • Options • Reply 0

Thank you so much for this thorough introduction! It was super helpful for getting started!

loveSnowBest • Posted on Version 35 • a year ago • Options • Reply 0

Great job! This kernel helped me a lot! It make me know how to handle data with so many(for me) features!

loveSnowBest • Posted on Version 35 • a year ago • Options • Reply 0

loveSnowBest • Posted on Version 35 • a year ago • Options • Reply 0

Janio Martine... • Posted on Version 35 • a year ago • Options • Reply 0

Awsome Kernel! Thanks for sharing! For me the hardest part of this kernel is the amount of features it had "81" columns! Nevertheless, I am the type of person who likes to perform an exhaustive exploratory analysis in order to get behind the story of our data! Then and only then I start working with the model itself. But I understand your point if we do an exhaustive analysis of this data we might spend months or even years exploring this data. :) Again, thanks for sharing the kernel its concise and descriptive.

Felipe • Posted on Version 35 • a year ago • Options • Reply 0

Nice job!

BenHall • Posted on Version 35 • a year ago • Options • Reply 0

This is great!

jayron soares • Posted on Version 36 • a year ago • Options • Reply 0

Excelente. Aprendi bastante, além disso sua abordagem pedagógica é fácil e simples. OBrigado

SanD • Posted on Version 36 • a year ago • Options • Reply 0

Really Great ! Enjoyed !

olli • Posted on Version 36 • a year ago • Options • Reply 0

Thank you for this notebook. To pick up a point that several other comments addressed: I do not understand why especially the independent variables have to be normally distributed. To quote from [http://www.statisticssolutions.com/normality/][1]

> The normality assumption is one of the most misunderstood in all of statistics. In multiple regression, the assumption requiring a normal distribution applies only to the disturbance term, not to the independent variables as is often believed.

I fitted the data to simple regression models and get almost same scores with or without the log-transformations. I'm not an expert, but I cannot think of or find any explanation, why everything (dependent AND independent variables) has to be normally distributed in regression analysis.

olli • Posted on Version 36 • a year ago • Options • Reply 0

sorry double post

Gabriel Falco... • Posted on Version 36 • a year ago • Options • Reply 0

Thanks for the effort on this kernel. Awesome explanation!

PranitSrivats... • Posted on Version 36 • a year ago • Options • Reply 0

amazing kernal

tnlin • Posted on Version 36 • a year ago • Options • Reply 0

impressive, mark

Lastnight • Posted on Version 36 • a year ago • Options • Reply 0

Nice explanations. thanks :)

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply
0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Wall-E • Posted on Version 36 • a year ago • Options • Reply 0

Sergio Marin • Posted on Version 37 • a year ago • Options • Reply 0

Hello Pedro,

I am new in Data science and this work for me is just amazing, it is my starter point, I am so grateful with you.

I have only one question, Why do you treat the missing data in the middle of the process instead to do that in the beginning?

Kumar Subh... • Posted on Version 37 • a year ago • Options • Reply 0

Great work Sir. Truly helped me a lot as a beginner

Aditya Soni • Posted on Version 37 • a year ago • Options • Reply 0

Awesome Plots..
Thank You..

Miguel • Posted on Version 37 • a year ago • Options • Reply 0

Muito bom Pedro! Obrigado

chenwucun • Posted on Version 38 • a year ago • Options • Reply 0

Thanks for your marvellous work, and I learned a lot from it . Thank you , please write more.

himanshu • Posted on Version 38 • a year ago • Options • Reply 0

very well explained

racekiller • Posted on Version 38 • a year ago • Options • Reply 0

Any one knows where to get this Multivariate book for lower prices? compared to amazon?

USB • Posted on Version 38 • a year ago • Options • Reply 0

This is very helpful.

Markus Lang • Posted on Version 39 • a year ago • Options • Reply 0

Well, i understood it pretty well, but when i use pd.get_dummies() the length of the test set and training set differs. How to deal with that? :(

Tommao • Posted on Version 39 • a year ago • Options • Reply 0

Thanks a lot for your share, and I have one question, what does "df*train = pd.get*dummies(df_train)" used for?

Aschwin • Posted on Version 39 • a year ago • Options • Reply 0

Ha Pedro,

Thanks for this awesome kernel! Learned a ton and love your funny style of writing. The checking out our girl method - really helps to remember the steps in a data science project. Gives you some good (love) handles for tackling big datasets.

Cheers, A.

Hakeem Frank • Posted on Version 39 • a year ago • Options • Reply 0

Aside from the kernel being extremely informative, the analogies used were hilarious as well. Great teaching!

JustinChow • Posted on Version 39 • a year ago • Options • Reply 0

Information was good. Jokes were terrible

Chang Gao • Posted on Version 39 • a year ago • Options • Reply 0

If I didn't miss anything, seems like no N/A value transformations. Anyway, good job!

gectrics • Posted on Version 39 • a year ago • Options • Reply 0

Hello,
Thank you a lot.
I have one question: Why are you normalizing GrLivArea and TotalBsmtSF ?
Thank you

Yajie Dong • Posted on Version 40 • a year ago • Options • Reply 0

Thank you so much! It helps me a lot!

Gabriel Preda • Posted on Version 40 • a year ago • Options • Reply 0

Great Kernel. And you got it right with Homoscedasticity although I never understood why in English someone will write Heteroskedasticity and Homoscedasticity for its opposite :-).

Jacinto Domí... • Posted on Version 40 • a year ago • Options • Reply 0

Thanks for your comprehensivility! Nice tuto.

MaximeG • Posted on Version 40 • a year ago • Options • Reply 0

Thanks for your nice analysis!
A question about outliers: would it be reasonable to use the probability plots (after log transformation) to identify them? In particular, the probability plot for log(SalePrice) clearly shows 5 isolated data point in the bottom left and 2 in the top right.

ManjeetKum... • Posted on Version 40 • a year ago • Options • Reply 0

Thanks a lot

Blueview • Posted on Version 40 • a year ago • Options • Reply 0

Fantastic kernel! Thanks for sharing!

repun • Posted on Version 40 • a year ago • Options • Reply 0

Thank you for useful information!
It helped me a lot.

李俊 • Posted on Version 40 • a year ago • Options • Reply 0

祝大家新年快乐！

HELEN TIAN • Posted on Version 41 • a year ago • Options • Reply 0

Thank you ! I learned a lot~~

CedSoft • Posted on Version 41 • a year ago • Options • Reply 0

Thanks a lot for your post! It's very great. I've seen in my research there is some built-in function to dectect outliers.

Would you have a such of post but for Time series? or something like that?

Thanks ahead

Square • Posted on Version 42 • a year ago • Options • Reply 0

Thanks a lot for taking the time to write this. Practical advice on real data is absolutely crucial for beginners.

Makhthum S... • Posted on Version 42 • a year ago • Options • Reply 0

That's a great and simple step by step explanation for data exploration. Thanks a lot for sharing.

Sharath Koth... • Posted on Version 43 • a year ago • Options • Reply 0

Great tutorial Pedro,Thanks for sharing.
I have One question, In case of numerical features we use correlation matrix to know the relation between the features,so

how do we find the correlation between continous/categorical target variable and categorical features.

Pedro Mar... Kernel Author • Posted on Version 43 • a year ago • Options • Reply 1

Thanks Sharath! I'd say it depends on what you want to do. In a strict sense, I think we can't talk about correlation because it usually expresses a relationship of this kind: 'when variable *x* increases, variable *y* increases/decreases/or stay the same'. Thus, with continuous/categorical we can't establish this type of relationship. However, we can use other strategies to understand how the variables work together. I'd say that box plots or significance tests can be helpful. At least, that's what I'd try in first place. I hope it helps!

Sharath K... • Posted on Version 43 • a year ago • Options • Reply 1

Thank you Pedro!!!

Tony Nguyen • Posted on Version 43 • a year ago • Options • Reply 0

Thank you for sharing, amazing kernel

Tan Moy • Posted on Version 43 • a year ago • Options • Reply 0

How do I learn this level of Data Mining? Can some one(Pedro?) recommend me some books, or point to some resources?

Ashan Priyad... • Posted on Version 43 • a year ago • Options • Reply 0

Wow thanks for sharing this kernal. Really helped me with many practical basics. Wonderful read!

Vamshi Krish... • Posted on Version 43 • a year ago • Options • Reply 0

Very useful for a beginner. Thanks Pedro!

Michael Ma • Posted on Version 44 • a year ago • Options • Reply 0

Appreciate. Quite good for learning of Entry level like me.

Dibya Ranjan ... • Posted on Version 44 • a year ago • Options • Reply 0

Great work!!!

Alif Ahmed • Posted on Version 44 • a year ago • Options • Reply 0

Great post

Akshay • Posted on Version 45 • a year ago • Options • Reply 0

Great work! Very informative!

Anil kumar G... • Posted on Version 45 • a year ago • Options • Reply 0

Thanks a lot!

Andrew Yip • Posted on Version 45 • a year ago • Options • Reply 0

Love this kernel. In addition to the very helpful demonstration of EDA techniques, it has a wonderful style too! This really helps making it memorable. I'll try something like this in my next kernel :)

Tony Long • Posted on Version 45 • a year ago • Options • Reply 0

Thanks very much for the sharing! This is the first kernel I read in Kaggle and I immediately found how helpful it is.

Brice Walker • Posted on Version 46 • a year ago • Options • Reply 0

This was very helpful!

J Kelley • Posted on Version 46 • a year ago • Options • Reply 0

This is really good. I really liked the analogies you provided.

Vishal Garg • Posted on Version 46 • a year ago • Options • Reply 0

Thanks a lot for this kernel. Learnt a lot of new stuff.

jruots • Posted on Version 46 • a year ago • Options • Reply 0

Excellent resource Pedro, well done! No matter how practiced people feel about their approach to data exploration, I feel that there's always something new you can pick-up by reading other people's approaches :) Cheers!

vivek • Posted on Version 47 • a year ago • Options • Reply 0

Really helpful Tuts for beginner

Sebastian Kr... • Posted on Version 47 • a year ago • Options • Reply 0

This is really great, thx a lot for this kernel!

Nitish Singh • Posted on Version 47 • a year ago • Options • Reply 0

This one is a keeper (swipe right) !! Awesome commentary too.

Matheus Baldi • Posted on Version 48 • a year ago • Options • Reply 0

Really awesome! Thank you!

henok • Posted on Version 48 • a year ago • Options • Reply 0

This was really helpful. Thanks a lot !!

J Theys • Posted on Version 48 • a year ago • Options • Reply 0

Thanks for sharing this notebook.

kobi23 • Posted on Version 48 • a year ago • Options • Reply 0

Great work - thanks!

David Joy • Posted on Version 53 • a year ago • Options • Reply 0

Brilliant Note book

Ardi Tan • Posted on Version 53 • a year ago • Options • Reply 0

Thanks for the kernel :), really helpful

guochuan • Posted on Version 53 • a year ago • Options • Reply 0

Hey, Pedro! Thanks for your sharing. I've learned much about how to analyze and visualize the features. Really inspire me! Thanks again! (Forgive my poor English)

Miles • Posted on Version 53 • a year ago • Options • Reply 0

Thanks for the effort!

Miles • Posted on Version 53 • a year ago • Options • Reply 0

Akhilesh • Posted on Version 53 • a year ago • Options • Reply 0

Nice write up. I liked the section where you made use of log transformation to make the variables normally distributed. Learning for me.

Pedro Mar... Kernel Author • Posted on Version 53 • a year ago • Options • Reply 0

Thank you Akhilesh! If you liked that, I suppose you will like to read this post where Box-Cox transformations are applied to make data normally distributed.

Zigzag14 • Posted on Version 53 • a year ago • Options • Reply 0

nice and clear. thanks for sharing

forever0612 • Posted on Version 53 • a year ago • Options • Reply 0

I would like to ask a question:By using codes " train.corr()", how dose python calculate the correlation of features which are object type(like MSZoning, LotShape, LotConfig and so on)?

Pedro Mar... Kernel Author • Posted on Version 58 • a year ago • Options • Reply 0

Good question forever0612! Object variables can have different types of data (e.g. strings or timestamps). This will influence the way you'll analyze the data.

Nevertheless, I'd say that, in general, you will end up using an encoding solution. There are several encoding solutions, all of them with different purposes and characteristics. You can get an overview of encoding solutions for the specific case of categorical variables in this site.

But, let's look at our data set and run an example. Imagine that you want to include MSZoning in your correlation analysis. This variable has 8 different types of zoning (A, C, FV, I, RH, RL, RP, RM). One way to encode this is through a 'one hot encoding' approach. This approach will convert each of the 8 category values into a new column, assigning a 1 or 0 (True/False) value to the column. Accordingly, if your observation belongs to the 'A' zone, it will have value 1 in the recently created column that refers to 'A', and 0 in the remaining columns created (referring to C, FV, I, etc.). I believe it looks complicated, but relax. Pandas already did the hardest part :)

Say something if I wasn't sufficiently clear in my explanation (or if I misunderstood your question)!

forever0612 • Posted on Version 58 • a year ago • Options • Reply 0

I'm so glad to see your reply! I know what you are talking about. You are talking about how to deal with categorical variables, but my question is how does python deal with these data by using the specific function 'train.corr()' to calculate the correlation(details of how to calculate the correlation)?
I look forward to seeing your reply!

Pedro Mar... Kernel Author • Posted on Version 58 • a year ago • Options • Reply 0

Ah, got it! So, if you take a look at the correlation heatmap, you'll notice that Python excludes the object variables. *corr()* can't deal with MSZoning, LotShape, LotConfig and so on. For more details, you can always look at the source code of corr(). It should be enlightening :) Thanks for your question!

forever0612 • Posted on Version 58 • a year ago • Options • Reply 0

Oh, yes...I should say I didn't read the correlation heatmap very carefully... Thank you very much for your reply!

arjitsakhuja • Posted on Version 58 • a year ago • Options • Reply 0

Very informative!!

Anand Jena • Posted on Version 58 • a year ago • Options • Reply 0

Can anyone explain why normality is important and why transformation is done using the log function..

Priyanka Soni • Posted on Version 58 • a year ago • Options • Reply 0

This was really helpful to me!

SDD_123 • Posted on Version 58 • a year ago • Options • Reply 0

thank you,I've learned a lot

Ave 30 • Posted on Version 58 • a year ago • Options • Reply 0

Quite useful to me ! Thanks

Prakash Jhun... • Posted on Version 59 • a year ago • Options • Reply 0

Thanks a lot for sharing!

Mheboobkhan • Posted on Version 59 • a year ago • Options • Reply 0

GooD ONE :-*

EdisonLiu • Posted on Version 59 • a year ago • Options • Reply 0

Interesting storytelling for the data ! Thanks a lot.

Ryuhei F. • Posted on Version 59 • a year ago • Options • Reply 0

Thank you so much, Pedro!!
I didn't know what to do at first, but you showed me the way how to do preprocessing.

However, maybe some beginner like me don't know what to do after this analysis.
For those beginners including me, I created a Kernel of this competition.
https://www.kaggle.com/ryuheeeei/what-we-do-after-preprocessing
Any comments and questions are very welcome!

Bishdata • Posted on Version 61 • a year ago • Options • Reply

0

Good try!

HJKIM • Posted on Version 61 • a year ago • Options • Reply

0

Thank your for the kernel. The variable transform part was really useful for me.
I hope that there was more information after the dummy part and applied models for transformed data as well.

Zerecas • Posted on Version 61 • a year ago • Options • Reply

0

Excuse me, I'm trying to understand the code as follow:

cols = corrmat.nlargest(10, '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)

So we started with 10 variables by SalePrice on top, but what about ['SalePrice'].index? Wouldn't be .index just fine?
And then what does it mean with df_train[cols].values.T which subsequently affect yticklabels and xticklabels.

I tried one version without .index and without .values.T, the label turns out to be a myriad of number, I just gonna assume the original code somewhat convert the number to text at the end?

Prashant Bra... • Posted on Version 61 • a year ago • Options • Reply

0

what is the difference between isnull().sum() and isnull().count() ? As a newbie I couldn't figure out.

Umar Murt... • Posted on Version 61 • a year ago • Options • Reply 0

sum() adds the values in the column/variable and count() only counts the entries in the column/variable
in sum, True is taken as 1 and False as 0, where as in count both are taken as one entry

also count excludes NaNs, thats why isnull is applied to make all the entries either True or False and no NaN

Eduard Pock... • Posted on Version 61 • a year ago • Options • Reply 0

very interesting kernel

cxue34 • Posted on Version 61 • a year ago • Options • Reply 0

Good work, thanks for sharing!

Sane • Posted on Version 62 • a year ago • Options • Reply 0

Thank you for your kind instruction!!

Katherine Zh... • Posted on Version 62 • a year ago • Options • Reply 0

Very helpful! Thanks a lot

yuuki_k • Posted on Version 62 • a year ago • Options • Reply 0

Great work! Thanks

Juan • Posted on Version 62 • a year ago • Options • Reply 0

Thank you, it's helpful.

Joe K • Posted on Version 62 • a year ago • Options • Reply 0

Nice

xzhendong • Posted on Version 62 • a year ago • Options • Reply 0

Thank you for sharing. It's very useful for me!

Kunal • Posted on Version 62 • a year ago • Options • Reply 0

Thanks.... it gave my data science journey a good start :)

WendongQu • Posted on Version 62 • a year ago • Options • Reply 0

thank you so much!

Js20 • Posted on Version 62 • a year ago • Options • Reply 0

worst thing ever

xuyiren • Posted on Version 62 • a year ago • Options • Reply 0

Thank you for youruseful kernel!

Alexandre Ga... • Posted on Version 62 • a year ago • Options • Reply 0

Hey... thanks a lot for this wonderful tutorial!!! I am a noon, I learned more in few lines than in hundred of mooc 's hours

Muhammed ... • Posted on Version 62 • a year ago • Options • Reply 0

Great article. Thanks a lot.
So what I gather going through various helpful comments is that, one of the goals of doing **log transformation** is to avoid skewing of our data. My questions are:

1. Are we going to train our model on this *log transformed* data ?
2. If yes, then do we need to *log transform* our test data as well ?
3. How do we get back the normal/actual values, say for the variable 'SalePrice', after *log transforming* it ?

Thanks.

X1 • Posted on Version 63 • 10 months ago • Options • Reply 0

1. Yes,
2. we need log transform to our test data too but not predict value like 'SalePrice'
3. SalePrice is not transformed, it's still keep original style.

Fauzan Taufik • Posted on Version 62 • a year ago • Options • Reply 0

same as you, this is like magic, avada kadavara, thank for magical tutorial

Balachandar ... • Posted on Version 62 • a year ago • Options • Reply 0

Great work. Learned a lot of new stuff related to data analysis & feature engineering.

Siddharth Jain • Posted on Version 63 • a year ago • Options • Reply 0

Really helpful for budding data scientists like me.

michal-lis • Posted on Version 63 • a year ago • Options • Reply 0

Fantastic! Very informative and entertaining.

Yuma Uchiumi • Posted on Version 63 • a year ago • Options • Reply 0

It is amazing. Thank you.

Yuri Istomin • Posted on Version 63 • a year ago • Options • Reply 0

Thanks for the notebook

DLuna • Posted on Version 63 • a year ago • Options • Reply 0

Very helpful notebook, thanks!

BIKUTA TEN • Posted on Version 63 • a year ago • Options • Reply 0

Negative correlation is not a bad thing. Sort the correlation with an absolute.

Ludovico Rist... • Posted on Version 63 • a year ago • Options • Reply 0

Very interesting and useful notebook, many thanks.
Only a little question about the use of order or unordered factors. I tried some simplified examples of linear models based on 2-3 variables where at least one was a factor. Well, probably I have been unlucky, but I found that the nature (ordered or unorderd) of the factor didn't affect the overall R2 and p-value, whereas it could affect singles p-values.
To sum up: why assigning the correct nature to a factor is important? Can anyone provide me with a simple example where defining a factor as ordered dramatically impact the performance of the model?
Thanks again, bye.

Song • Posted on Version 63 • a year ago • Options • Reply 0

Thanks for sharing, really helpful

RoniFinTech • Posted on Version 63 • 10 months ago • Options • Reply 0

Thanks...

AnmolArora • Posted on Version 63 • 10 months ago • Options • Reply 0

Thank you for putting it together. This is quality stuff.

Ayush Singhal • Posted on Version 63 • 10 months ago • Options • Reply 0

Nice work

Sai Darahas • Posted on Version 63 • 10 months ago • Options • Reply 0

Great

Pulkit Sharma • Posted on Version 63 • 10 months ago • Options • Reply 0

Nice !!

Gunjan Acha... • Posted on Version 63 • 10 months ago • Options • Reply 0

Good Work

Alex Coleman • Posted on Version 63 • 10 months ago • Options • Reply 0

Interesting and useful notebook thanks.

myexpfactory • Posted on Version 63 • 10 months ago • Options • Reply 0

Thank you very much for sharing your work!
As a newcomer, I appreciated the effort on humor!

Irving • Posted on Version 63 • 10 months ago • Options • Reply 0

now，I'm too busy at party to forgot what data analysis is.

Roopam Shar... • Posted on Version 63 • 10 months ago • Options • Reply 0

Great explanation, learned a lot from this kernel

Shilpi Goel • Posted on Version 63 • 10 months ago • Options • Reply

0

Very well explained. Great learning material for beginners!

Petr Mitev • Posted on Version 63 • 10 months ago • Options • Reply

0

Appreciate the help. Great starting point!

wakit • Posted on Version 65 • 10 months ago • Options • Reply

0

Thanks for sharing this!

SM Niaz Arifin • Posted on Version 65 • 10 months ago • Options • Reply

0

Awesome write-up! Useful, nice plots!

Jiayu • Posted on Version 65 • 10 months ago • Options • Reply

0

This is really nice, thank you!

Anshul Mohil • Posted on Version 65 • 10 months ago • Options • Reply

0

Thank for sharing. It helped a lot!!

liujing • Posted on Version 65 • 9 months ago • Options • Reply

0

Thank you for the kernel. As a beginner I learned a lot from this.

Manuel Dons... • Posted on Version 65 • 9 months ago • Options • Reply 0

Thanks for this amazing kernel! Very useful for a beginner like me :)

Stefano Zakh... • Posted on Version 66 • 9 months ago • Options • Reply 0

Great Kernel!

J Sherfey • Posted on Version 66 • 9 months ago • Options • Reply 0

Super helpful! Good work

MC • Posted on Version 66 • 9 months ago • Options • Reply 0

wonderfull work, Thanks for sharing

Madhuri Sival... • Posted on Version 66 • 9 months ago • Options • Reply 0

@Pedro - This is an excellent notebook! I learnt about Homoscedasticity, thanks to your notebook! I have a better understanding regarding it now :)

Benjamin Tan • Posted on Version 66 • 9 months ago • Options • Reply 0

Excellent work!

Davis Peixoto • Posted on Version 66 • 9 months ago • Options • Reply 0

Thx for this, buddy.

David Ly • Posted on Version 66 • 9 months ago • Options • Reply 0

Pretty cool.

VenkataRam... • Posted on Version 66 • 9 months ago • Options • Reply 0

simply supersuper.. thank you

abcde • Posted on Version 66 • 9 months ago • Options • Reply 0

Very good!

mkiuchi • Posted on Version 66 • 9 months ago • Options • Reply 0

Thanks !

Denis Bilalov • Posted on Version 66 • 9 months ago • Options • Reply 0

Wonderful kernel! Thanks!

akshatp • Posted on Version 66 • 9 months ago • Options • Reply 0

Very nice kernel.Thank you!!

akshatp • Posted on Version 66 • 9 months ago • Options • Reply 0

Zhifu Zhu • Posted on Version 66 • 9 months ago • Options • Reply 0

Good work, thank you!

Rogério Ignác... • Posted on Version 66 • 9 months ago • Options • Reply 0

:D

MeiNinghang • Posted on Version 66 • 9 months ago • Options • Reply 0

very good!

Eric Qian • Posted on Version 66 • 9 months ago • Options • Reply 0

It's cool! Thanks a lot!

ShumpeiWat... • Posted on Version 66 • 9 months ago • Options • Reply 0

nice!

Michael Maill... • Posted on Version 66 • 9 months ago • Options • Reply 0

Thanks for the exploration. I'll definitely learn how to use seaborn now, because I can see how useful it was here.

Shiva Moorthy • Posted on Version 66 • 9 months ago • Options • Reply 0

Thanks for the nice explanation!

Denis Teslenko • Posted on Version 66 • 9 months ago • Options • Reply 0

Good work, thank you.

Shen Yue • Posted on Version 66 • 9 months ago • Options • Reply 0

Thanks a lot. Helping me understand the data.

Türkay AVCI • Posted on Version 66 • 9 months ago • Options • Reply 0

Thanks for this awesome kernel, it teaches a lot.

Akshay Bhar... • Posted on Version 66 • 9 months ago • Options • Reply 0

Great kernel! Love your humor as well.

Yerassyl Diyas • Posted on Version 66 • 9 months ago • Options • Reply 0

Thank u for kernel, I learned a lot from it. But personally for me it seemed strange to drop columns like this:

```
df_train = df_train.drop((missing_data[missing_data['Total'] > 1]).index,1)
```

Isnt't it better to drop columns explicitly?
Because we can not apply this code to unseen data, since unseen data might have different columns that has no data, so we might end up deleting different columns. I think it is better to decide explicitly which columns to drop, so that when we deal with unseen data (i mean predict on unseen data) we drop exactly same columns. This in my intuition about that.

P.S:I am a novice to ml, sorry If I wrote some bullsh*t :).

Shen Yue • Posted on Version 66 • 9 months ago • Options • Reply 0

high quality !

MJLEE • Posted on Version 66 • 9 months ago • Options • Reply 0

Thank you !

Marouane Di... • Posted on Version 66 • 9 months ago • Options • Reply 0

Very good kernel and references
Thanks

Harsh Sharma • Posted on Version 66 • 9 months ago • Options • Reply 0

Quite informative!

Ayush Saxena • Posted on Version 67 • 9 months ago • Options • Reply 0

Nicely written. The NA in garage variables is not missed data but it means garage not available. So shouldn't those variables should not be deleted?

Brendan Fitz... • Posted on Version 67 • 9 months ago • Options • Reply 0

Awesome and clear explanations. Super helpful!

PankajMahan • Posted on Version 67 • 9 months ago • Options • Reply 0

Thanks for this great demonastration.

Andy • Posted on Version 67 • 9 months ago • Options • Reply 0

Thanks for your effort with this, learned a lot! :)

じーたく • Posted on Version 67 • 9 months ago • Options • Reply 0

Thanks!

Kaustubh Ma... • Posted on Version 67 • 9 months ago • Options • Reply 0

Very helpful kernel indeed !!

Austin Arain • Posted on Version 67 • 9 months ago • Options • Reply 0

This was really interesting, thank you for sharing your work!

Niharika yadav • Posted on Version 67 • 9 months ago • Options • Reply 0

Nice work. Thanks for sharing.

Haithem Mzo... • Posted on Version 67 • 9 months ago • Options • Reply 0

Good Job !

LennyMaz • Posted on Version 67 • 9 months ago • Options • Reply 0

Nice job and enjoyed the humorous spin

Kim YoonSoo • Posted on Version 67 • 9 months ago • Options • Reply 0

Thank you for sharing this kernel

AasifMultani • Posted on Version 67 • 8 months ago • Options • Reply 0

Great Kernel for EDA

Pierre Aumja... • Posted on Version 67 • 8 months ago • Options • Reply 0

Great tutorial, thanks!

Chitipolu Sri ... • Posted on Version 67 • 8 months ago • Options • Reply 0

Hi Marcelino,I have general question on NAN values.you removed the features with more NAN values. But in electrical there is only one observation with NAN value rather than removing it we can also replace with mean /median/mode .so by that it will give high scores.

Pedro Mar... Kernel Author • Posted on Version 67 • 8 months ago • Options • Reply 0

Sure. You can always go for some different way to deal with missing data. Not always mean/median/mode will improve your score. If you really want to get that answer, you should test it. Scikit-learn has a nice example on the effect of missing data imputation. I'd recommend you to read it. You'll enjoy it :)

Chitipolu S... • Posted on Version 67 • 8 months ago • Options • Reply 0

thank you

Peter Leonard • Posted on Version 67 • 8 months ago • Options • Reply 0

Thanks! really good comprehensive guide to start with!

DavidLeal • Posted on Version 67 • 8 months ago • Options • Reply 0

Thanks very helpful

upendra sing... • Posted on Version 67 • 8 months ago • Options • Reply 0

Thanks, this Data Exploration really brings in a lot of useful ideas.

Mani • Posted on Version 67 • 8 months ago • Options • Reply 0

Great kernel !

eb1c0n • Posted on Version 67 • 8 months ago • Options • Reply 0

AWSM!

savvas • Posted on Version 67 • 8 months ago • Options • Reply 0

Great job, very helpfull thanks.

KaiMonmoy • Posted on Version 67 • 8 months ago • Options • Reply 0

Awesome tutorial! Great story! Loved it!!!

Akash Tyagi • Posted on Version 67 • 8 months ago • Options • Reply 0

cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index
cm = np.corrcoef(df_train[cols].values.T)
Can someone help me with what's going on here ?
I get that we are trying to find the correlation metrics between the k no. of values , but even after trying hard couldn't get how to above code is working. A walkthrough of above snippet would be very helpful.
Thanks

Alex Radae... • Posted on Version 67 • 8 months ago • Options • Reply 1

Coorrmat is the square dataframe that contains correlation of all variables. Columns names and indexes are the names of variables from df*train. nlargest(k, 'Sale price') gives you new dataframe that contains k largest values (correlations) in the column "Sale price". The column "Sale price" contains correlation of the sale price with all other variables. When you take index of it, you get the names of k variables with the highest correlation with Sale price. Next line (cm=...) calculates coefficients of correlation for df*train columns with these names.

Akash Tyagi • Posted on Version 68 • 8 months ago • Options • Reply 0

Thanks, this explanation made things clear.

Akash Tyagi • Posted on Version 68 • 8 months ago • Options • Reply 0

What does T in values.T do ?

Alex Radae... • Posted on Version 68 • 8 months ago • Options • Reply 0

.T transposes the values vector.
corrcoef is a numpy function, it expects a matrix as an input with rows to represent variables and columns - observations, so we need to transpose.

Min Kwon • Posted on Version 67 • 8 months ago • Options • Reply 0

Really helpful guide for beginner like me, who is not familiar with many variables. thanks..!

Joseph H. • Posted on Version 67 • 8 months ago • Options • Reply 0

This is awesome, I can't wait to sink my teeth into it.

André Thiago • Posted on Version 67 • 8 months ago • Options • Reply 0

I have a question: do I have to apply the log transformation in the test dat set?

Pedro Mar... Kernel Author • Posted on Version 68 • 8 months ago • Options • Reply 0

Yes. The test data set should be transformed like the train data set.

Deepak • Posted on Version 67 • 8 months ago • Options • Reply 0

This is terrific! Thanks a lot for sharing

Nocturne_Jay • Posted on Version 68 • 8 months ago • Options • Reply 0

Funny story and nice work!

Starshunter • Posted on Version 68 • 8 months ago • Options • Reply 0

Thanks for sharing

Shreyas • Posted on Version 68 • 8 months ago • Options • Reply 0

In some columns like the Alley column the NA(No alley access) is represented by nan . So that means that not all values are null , some may be NA . Am i right in thinking like this , if so what options do we have in regards to feature engineering that particular column.

Kevin Degila • Posted on Version 68 • 8 months ago • Options • Reply 0

This was so well written and i love the visualization. Learned a lot

Hariprasad M... • Posted on Version 68 • 8 months ago • Options • Reply 0

Thanks for your interesting notebook.

Mehul jain • Posted on Version 68 • 8 months ago • Options • Reply 0

Thanks, Well explained

Reine Asakaw... • Posted on Version 68 • 8 months ago • Options • Reply 0

Nice work! Thank you!

NajibBakahoui • Posted on Version 68 • 7 months ago • Options • Reply 0

Very good work. Thanks

Xudong • Posted on Version 68 • 7 months ago • Options • Reply 0

Thank you so much !

Konstantin S... • Posted on Version 68 • 7 months ago • Options • Reply 0

Great job! Thank you for very much for sharing!

sonalpingle • Posted on Version 68 • 7 months ago • Options • Reply 0

This is excellent, thank you!

Navya Mokm... • Posted on Version 68 • 7 months ago • Options • Reply 0

Thanks for sharing!

Ruturaj Gujar • Posted on Version 68 • 7 months ago • Options • Reply 0

Thank you for the detailed explaination! Got to learn many new things.

jolyuD2 • Posted on Version 68 • 7 months ago • Options • Reply 0

thanks HTML

jolyuD2 • Posted on Version 68 • 7 months ago • Options • Reply 0

thanks HTML

Doris Ma • Posted on Version 68 • 7 months ago • Options • Reply 0

It's really helpful, thanks!

Jake Taylor • Posted on Version 68 • 7 months ago • Options • Reply 0

Thanks!

Shashank • Posted on Version 68 • 7 months ago • Options • Reply 0

Lots of Important stuff. Thanks, Best kernel

poompoowit • Posted on Version 68 • 7 months ago • Options • Reply 0

Thanks a lot!

Alakar Ramani • Posted on Version 68 • 7 months ago • Options • Reply 0

Thanks! Your code provides great insight on the data and makes it easy to model.

vashisht arora • Posted on Version 68 • 7 months ago • Options • Reply 0

Thanks

vashisht arora • Posted on Version 68 • 7 months ago • Options • Reply 0

Arnab Rajkho... • Posted on Version 68 • 7 months ago • Options • Reply 0

Hi,

I am a newbie to ML and I have a question here. Since the test set and the training set are given separately, do we need to concatenate them before searching for missing values and filling out the missing values accordingly.

As I have seen here and a few other related kernels, all the analysis is being carried on the test set only. I hope you understand my question here and help me out. Thank you.

Pedro Mar... Kernel Author • Posted on Version 68 • 7 months ago • Options • Reply 2

Hey Arnab! How are you doing?

The answer to your question is that you shoulnd't concatenate the training and the test set. The test set is given to you so that you can have an estimate of the performance of your prediction model. Accordingly, you will **not** use it to train your model. You'll just use the test set to evaluate your model, simulating its performance on unseen data. If you concatenate the data, you'll fall into a leakage situation and **you don't want that** :)

Please, confirm me if I understood your question correctly and if this answer makes sense to you.

Best!

Chris Garcia • Posted on Version 68 • 7 months ago • Options • Reply 0

Great Kernel!

Evgeny Pogr... • Posted on Version 68 • 7 months ago • Options • Reply 0

Great balance between prose, code and graphics.

José David Ca... • Posted on Version 68 • 7 months ago • Options • Reply 0

Great job!

Sam Yao • Posted on Version 69 • 7 months ago • Options • Reply 0

Why are your's figures so beautiful, I almost try every parameter. I mean the distplot command. I cannot find how to control vertical lines and the colors of the background.

Francisco Litv... • Posted on Version 69 • 7 months ago • Options • Reply 0

Very helpful for a beginner like me! Thanks a lot :)

Francisco Litv... • Posted on Version 69 • 7 months ago • Options • Reply 0

Very helpful for a beginner like me! Thanks a lot :)

Arek Noster • Posted on Version 69 • 7 months ago • Options • Reply 0

You are the one, who helped me come through the entry-level barrier. Thank you!

Vibhanshu V... • Posted on Version 69 • 7 months ago • Options • Reply 0

Great work!!

Christian Gre... • Posted on Version 69 • 7 months ago • Options • Reply 0

Cool piece of work!

Ajinkya Gaik... • Posted on Version 69 • 7 months ago • Options • Reply 0

Really helpful. Thanks a lot!!

Flavian Manea • Posted on Version 69 • 7 months ago • Options • Reply 0

Thank you! Really helpful!

sunghyo • Posted on Version 69 • 7 months ago • Options • Reply 0

Thanks a lot for this great kernel!

Anthony Chan • Posted on Version 69 • 7 months ago • Options • Reply 0

Good kernel for beginners. Thanks!

BhargavTang... • Posted on Version 69 • 7 months ago • Options • Reply 0

Nice Work. Thanks!

flyingfeng • Posted on Version 69 • 7 months ago • Options • Reply 0

Great kernel, this helps me a lot.

flyingfeng • Posted on Version 69 • 7 months ago • Options • Reply 0

ikimavi • Posted on Version 69 • 6 months ago • Options • Reply 0

Awesome, the best so far, will follow

Small Timid B... • Posted on Version 69 • 6 months ago • Options • Reply 0

I have a XGBoost model , but I found that the performance is worse when I drop one of the 'twin brothers' pairs, maybe it's better we keep all related data if the data size is large enough?

Gabriel Preda • Posted on Version 70 • 6 months ago • Options • Reply 0

This is a very long Kernel :-). For testing the homoscedasticity, you can also try either Levene or Bartlett test, implemented also in Python, for example: https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.levene.html.

Jeung H Byun • Posted on Version 70 • 6 months ago • Options • Reply 0

Thanks for sharing! But I couldn't really understand the scatter plot in line 13.
Why is it not diagonally symmetric? And why is the diagonal scatter plots not y=x plot?

Das Kabital • Posted on Version 70 • 6 months ago • Options • Reply 0

Thank you!

Raj • Posted on Version 70 • 6 months ago • Options • Reply 0

Nice work Pedro....

Piotr Pawłow... • Posted on Version 70 • 6 months ago • Options • Reply 0

Very useful. Great job!! ;D

James A • Posted on Version 70 • 6 months ago • Options • Reply 0

Thank you! Very thorough and clear!

sriharshacha... • Posted on Version 70 • 6 months ago • Options • Reply 0

Good kernel to start with

SourabhP • Posted on Version 70 • 6 months ago • Options • Reply 0

Thank you.Great Kernel!

huynth • Posted on Version 70 • 6 months ago • Options • Reply 0

Thanks for sharing your notebook!

Scott Cohn • Posted on Version 70 • 6 months ago • Options • Reply 0

Thanks for sharing!

Nikita Sharma • Posted on Version 70 • 6 months ago • Options • Reply 0

Thanks for the awesome post, please check my kernel here and provide your thoughts - https://www.kaggle.com/nikkisharma536/data-exploration-and-visualisation-for-

beginner?scriptVersionId=6327684

Ilya • Posted on Version 70 • 6 months ago • Options • Reply

0

Great kernel, learned a lot!

HimanshuCh... • Posted on Version 70 • 5 months ago • Options • Reply

0

Thanks for explaining this!

Kuppo • Posted on Version 70 • 5 months ago • Options • Reply

0

It's very help for me,thank you.

易长安 • Posted on Version 70 • 5 months ago • Options • Reply

0

good jobs

Pei • Posted on Version 71 • 5 months ago • Options • Reply

0

Very helpful

Konstantin C... • Posted on Version 71 • 5 months ago • Options • Reply

0

Tortoise • Posted on Version 71 • 5 months ago • Options • Reply

0

Thanks for this great tutorial!
I got some confusion here that how did you come to know Id's of points which are outliers in following code

deleting points

df*train.sort*values(by = 'GrLivArea', ascending = False)[:2]
df*train* = *df*train.drop(df*train[df*train['Id'] == 1299].index)
df*train* = *df*train.drop(df*train[df*train['Id'] == 524].index)

Tommy • Posted on Version 71 • 5 months ago • Options • Reply

0

Amazing! So well explained:)

Nobuaki Oka... • Posted on Version 71 • 5 months ago • Options • Reply

0

Thank you very much for the nice kernel!
This kernel is really helpful for learning a data engineering sequence.
Also, this made me realize the importance of data engineering before predictions.

ShilpaDwivedi • Posted on Version 71 • 5 months ago • Options • Reply

0

Hi, i am not able to see your n notebook.

prabu rocking • Posted on Version 71 • 5 months ago • Options • Reply

0

very much helpful

Kaeldric • Posted on Version 71 • 5 months ago • Options • Reply

0

Thank you very much. I really liked your work and it taught me a lot of things.
I have three questions for you.

1 - When you deal with **TotalBsmtSF** and decide to not include zero values you wrote:

```
df_train['HasBsmt'] = pd.Series(len(df_train['TotalBsmtSF']),
index=df_train.index)
df_train['HasBsmt'] = 0
```

isn't the first row useless?

2 - You used log to make the variables normal because they have a positive skewness. What we should use when we have a negative skewness?

3 - It seems to me that you read a lot of books on the subject. Can you suggest a couple of books that you consider very useful to understand how to handle the very different situations a data scientist must face?

Thank you again.

Vivek • Posted on Version 71 • 5 months ago • Options • Reply 0

Nice work!

K Hariprakas... • Posted on Version 71 • 5 months ago • Options • Reply 0

good job dude..

Acdemical LJ • Posted on Version 71 • 5 months ago • Options • Reply 0

nice,666

Acdemical LJ • Posted on Version 71 • 5 months ago • Options • Reply 0

Emrul Hasan • Posted on Version 72 • 5 months ago • Options • Reply 0

Excellent Kernel! Very helpful. Thank you! how to replace the the outliers? Is there any kernel on this? Thanks!

wxqsjtu • Posted on Version 72 • 4 months ago • Options • Reply 0

Very helpful!

Cristian Vera • Posted on Version 72 • 4 months ago • Options • Reply 0

Hi! Thanks Pedro it was very helpful.
I have a question, when you applied log transformation on GrLivArea and TotalBsmtSF features, it is necessary applied also to test data on the same features

Trofimenko D... • Posted on Version 72 • 4 months ago • Options • Reply 0

Thank you.

Nils Schlüter • Posted on Version 72 • 4 months ago • Options • Reply 0

Good Summary, thank you!

Johan Eliasson • Posted on Version 72 • 4 months ago • Options • Reply 0

Thanks for a good walk-thru!

Albert • Posted on Version 72 • 4 months ago • Options • Reply 0

Thk u

Rostislav Liho... • Posted on Version 72 • 4 months ago • Options • Reply 0

Nice job! That's really comprehencive, thank you =)

Liyin Shao • Posted on Version 72 • 4 months ago • Options • Reply 0

Loud & Clear

ARAJafri • Posted on Version 72 • 4 months ago • Options • Reply 0

Hello , great post! I found it very helpful and enjoyable! I do have a uestion though ... are you only applying log transformations on features that have high correlation?

Kaja Mulumba • Posted on Version 72 • 4 months ago • Options • Reply 0

Thank you very much for this :).

Marcus Costa • Posted on Version 72 • 4 months ago • Options • Reply 0

Good job! Thanks for this kernel, it was very helpful!

soojung • Posted on Version 72 • 4 months ago • Options • Reply 0

Thank you! Very helpful

Dani Gordo • Posted on Version 72 • 4 months ago • Options • Reply 0

Nice Ayn Rand reference ;) Do you recomend EDA before o after missing value imputation?

Panks • Posted on Version 72 • 4 months ago • Options • Reply 0

Well, The Big Bang theory link was quite interesting!!!

nichen • Posted on Version 72 • 4 months ago • Options • Reply 0

Thank you for such great work! Just a little suggestion: in In[14], I think the more direct way to calculate percent should be “percent = (total/len(df*train)).sort*values(ascending=False) ” :)

Akshath Varu... • Posted on Version 72 • 4 months ago • Options • Reply 0

Thanks for the kernel! Very useful.

Dzmitry Shak... • Posted on Version 72 • 4 months ago • Options • Reply 0

Thanks for your notebook!

Oniel Gracious • Posted on Version 72 • 4 months ago • Options • Reply 0

Nice Work

Oniel Gracious • Posted on Version 72 • 4 months ago • Options • Reply 0

Bigkizd • Posted on Version 72 • 4 months ago • Options • Reply 0

so googoodgd

Siftnoor Singh • Posted on Version 72 • 4 months ago • Options • Reply 0

This kernel was very helpful. Thanks!

Sulav Jha • Posted on Version 72 • 3 months ago • Options • Reply 0

Interesting

Aman Nagari... • Posted on Version 72 • 3 months ago • Options • Reply 0

Great post! Thank you!
Any suggestion for other Notebook which is helpful to understand more about feature selection, OUTLIERS in-details also give the more clear picture to handle missing data.
Thanks in advance :)

Max Metz • Posted on Version 72 • 3 months ago • Options • Reply 0

First of all, thank you for this kernel. Despite the controvercies, I enjoyed reading it and learned a lot from this as a beginner. I have two questions:

1.) The code In [27 - 29]:
I did not understand what happened here. While I get the part of df_train.loc, I do not understand the plots you have created in [29]. Why did you write requirements here? (TotalBsmtSf > 0..) Isn't the data adjusted in the dataframe?

2.) I could not find a solution just from research, so I'd like to know: The most common solution for correlated errors is "trying to add a variable that explains this effect." according to Pedro Marcelino. What does this mean? An example might clarify this.

To whomever, thank you so much in advance!

paul • Posted on Version 73 • 3 months ago • Options • Reply 0

''SalePrice' correlation matrix (zoomed heatmap style)

both with correlation value 0.61, 'TotalBsmtSF' and '1stFloor' are twin brothers, no problem;
'GarageCars'(0.64) and 'GarageArea'(0.62) are like twin brothers,ok;
WHY 'TotRmsAbvGrd'(0.53) and 'GrLivArea'(0.71)?
Thanks!
moreover,'FullBath'?? Really?,only 2 words, with correlation value 0.56, tell us more please!

yippoday • Posted on Version 73 • 3 months ago • Options • Reply 0

Very interesting work

JP Beaudry • Posted on Version 73 • 3 months ago • Options • Reply 0

Thank you for the inspiration!

shruti kalra • Posted on Version 73 • 3 months ago • Options • Reply 0

nice work!!

chenyu3yu3 • Posted on Version 73 • 3 months ago • Options • Reply 0

thanks it helps me a lot!

Anand Praka... • Posted on Version 73 • 3 months ago • Options • Reply 0

Thanks!

Himanshu Sa... • Posted on Version 73 • 3 months ago • Options • Reply 0

Good Job

Yuma Uchiumi • Posted on Version 73 • 3 months ago • Options • Reply 0

Thank you!

Raj Choudhar... • Posted on Version 73 • 3 months ago • Options • Reply 0

Amazing explanation. Thank you!

zhuo • Posted on Version 73 • 3 months ago • Options • Reply 0

Thanks! But I have one question. why we need normality, by using log function.I am new for data mining,thanks for your help.

Piyush Gauta... • Posted on Version 73 • 3 months ago • Options • Reply 0

Nice job! Very well explained.

Olliviaaa • Posted on Version 73 • 3 months ago • Options • Reply 0

Why all the non-normal data should be transformed to normal data?

Olliviaaa • Posted on Version 73 • 3 months ago • Options • Reply 0

Thomas W • Posted on Version 73 • 3 months ago • Options • Reply 0

Thank you!

jmsbyl • Posted on Version 74 • 3 months ago • Options • Reply 0

interesting notebook - thanks

Oliver Wales • Posted on Version 74 • 2 months ago • Options • Reply 0

Thanks!

hourglass_ho... • Posted on Version 74 • 2 months ago • Options • Reply 0

Nice!

roottrx • Posted on Version 74 • 2 months ago • Options • Reply 0

Thank you! Very helpful

juanluna • Posted on Version 74 • 2 months ago • Options • Reply 0

Thank you for taking your time writing this. Even though there might be some "weak" points, the overall content is good enough to give you an upvote for the effort

KraLMachine • Posted on Version 74 • 2 months ago • Options • Reply 0

That's good work. I have a question in this work. How do you give the data points different colors when you show data on charts?

Estelle • Posted on Version 74 • 2 months ago • Options • Reply 0

Thank you lots, very appreciated!

shahad • Posted on Version 74 • 2 months ago • Options • Reply 0

Thank You!

Xiaochen Hu... • Posted on Version 74 • 2 months ago • Options • Reply 0

Awesome notebook with a lot of basic data exploration techniques covered. Great starting point for any one who wants to learn EDA!

Gerardo • Posted on Version 74 • 2 months ago • Options • Reply 0

Hola, buenas, creo que ahi utilizas panda

Gavin He • Posted on Version 74 • 2 months ago • Options • Reply 0

Hi,I'm so interested in this kernel.I think it's very useful for green hand. I want to translate this kernel into Chinese
and present as a new kernel. If you decide to give me the permission to translate, please reply to me! Thanks.

hidorder • Posted on Version 74 • 2 months ago • Options • Reply 0

thanks!

Kenny Hunt • Posted on Version 74 • 2 months ago • Options • Reply 0

Thanks very much, this was a very useful kernel and well explained

Aditya Zope • Posted on Version 74 • 2 months ago • Options • Reply 0

If I use k>10 for correlation matrix's nlargest , some rows go missing.
Why it is so?

ganesh • Posted on Version 74 • 2 months ago • Options • Reply 0

great ! highlight of your code is

convert categorical variable into dummy

RaviShankar • Posted on Version 74 • 2 months ago • Options • Reply 0

Its just wow, so many concepts explained in a pleasing fashion. Bookmarking this kernel for long time to come.

Byron Lopez • Posted on Version 75 • 2 months ago • Options • Reply 0

Great job! Thanks for sharing Pedro! By the way, boxcox1p from the scipy library is other alternative to handle data set containing zero values. Like Jeroen Vuurens says.

Umit Dogan • Posted on Version 75 • a month ago • Options • Reply 0

I learned a lot from this kernel. Thank you ! Very helpful

MauricioPerei... • Posted on Version 75 • a month ago • Options • Reply 0

Awesome man! It helped a lot :)

Dark • Posted on Version 75 • a month ago • Options • Reply 0

First thing first the explanation was wonderful and I enjoyed reading through this kernel (my personal favourite part was the proposal) . But I still have some doubts:-
What If we used df_train.corr() to get the relationship among all the features and select only those features that have a significant impact on the SalePrice (like correlation between OverallQual and SalePrice is 0.7909) ?
And the correalation between YearBuild and SalePrice is 0.52 whereas that between YearRemodAdd and SalePrice is 0.50 but we didn't selected YearRemodAdd as a feature of that importance . Can anyone help me plS :)

Eshaan Mang... • Posted on Version 75 • a month ago • Options • Reply 0

How can we remove Skeewness from data???? Does Taking Log always resolve this issue?

Rajeev A.K • Posted on Version 75 • a month ago • Options • Reply 0

Thanks .. this document helped me

GoceGor • Posted on Version 75 • a month ago • Options • Reply 0

I like the fact that your Kernel is not just helpful, but also very creative and keeps the attention very well! Keep up the good work!

Andrew Nigh... • Posted on Version 75 • a month ago • Options • Reply

0

Thankyou!

Vijaykumar H... • Posted on Version 75 • a month ago • Options • Reply

0

Thanks for sharing! very much helpful!

Masaya Yum... • Posted on Version 75 • a month ago • Options • Reply

0

Thanks.

HAFAR • Posted on Version 75 • a month ago • Options • Reply

0

Interesting!

Sam • Posted on Version 75 • a month ago • Options • Reply

0

thank you

CD2020 • Posted on Version 75 • 23 days ago • Options • Reply

0

Thanks

aditi • Posted on Version 75 • 20 days ago • Options • Reply

0

Nice work!

Asura • Posted on Latest Version • 17 days ago • Options • Reply 0

Great ! Thanks

Ali Nasri Nazif • Posted on Latest Version • 15 days ago • Options • Reply 0

Thanks for this great Kernel. I learned a lot.

Josh Janjua • Posted on Latest Version • 15 days ago • Options • Reply 0

Thank you for sharing and I really appreciate how entertaining the write up is. You have worked very hard and it is clearly very helpful to many of us.

Andrew Zolot... • Posted on Latest Version • 14 days ago • Options • Reply 0

Awesome job

Li Qinyue • Posted on Latest Version • 14 days ago • Options • Reply 0

Excellent EDA tutorial for beginners!
It helps me a lot!
Thank you!

karthik • Posted on Latest Version • 14 days ago • Options • Reply 0

Thank you!! great explanation ....

Jorge • Posted on Latest Version • 10 days ago • Options • Reply 0

Thanks for this kernell learned a lot.

Shubham Jai... • Posted on Latest Version • 9 days ago • Options • Reply 0

Thank you for sharing

Shubham Jai... • Posted on Latest Version • 9 days ago • Options • Reply 0

Dominick • Posted on Latest Version • 9 days ago • Options • Reply 0

Helpful and mildly entertaining. Thanks!

Victor Valente • Posted on Latest Version • 9 days ago • Options • Reply 0

This is the first data analysis ever that actually made me laugh. Completely worth of an upvote.

RipoJay • Posted on Latest Version • 8 days ago • Options • Reply 0

This is a really helpful kernel, thank you!

AlvaroCalleC... • Posted on Latest Version • 8 days ago • Options • Reply 0

Very informative work. Thanks for sharing

Kai Hirota • Posted on Latest Version • 7 days ago • Options • Reply 0

I tell all my friends who are starting out in data science to study this notebook. Thanks!

Shubhankar ... • Posted on Latest Version • 6 days ago • Options • Reply

0

Great work, thanks for this kernel, lot of information to be considered.
Also thanks for sharing the books and references.

wenwen • Posted on Latest Version • 5 days ago • Options • Reply

0

Thanks a lot! It really helpful

Satyam • Posted on Latest Version • 4 days ago • Options • Reply

0

Awesome tutorial .... and the story line as well..

a year ago

This Comment was deleted.

a year ago

This Comment was deleted.

a year ago

This Comment was deleted.

10 months ago

This Comment was deleted.

9 months ago

This Comment was deleted.

9 months ago

This Comment was deleted.

9 months ago

This Comment was deleted.

8 months ago

This Comment was deleted.

5 months ago

This Comment was deleted.

5 months ago

This Comment was deleted.

5 months ago

This Comment was deleted.

5 months ago

This Comment was deleted.

4 months ago

This Comment was deleted.

Similar Kernels




[Private Kernel]

[Private Kernel]