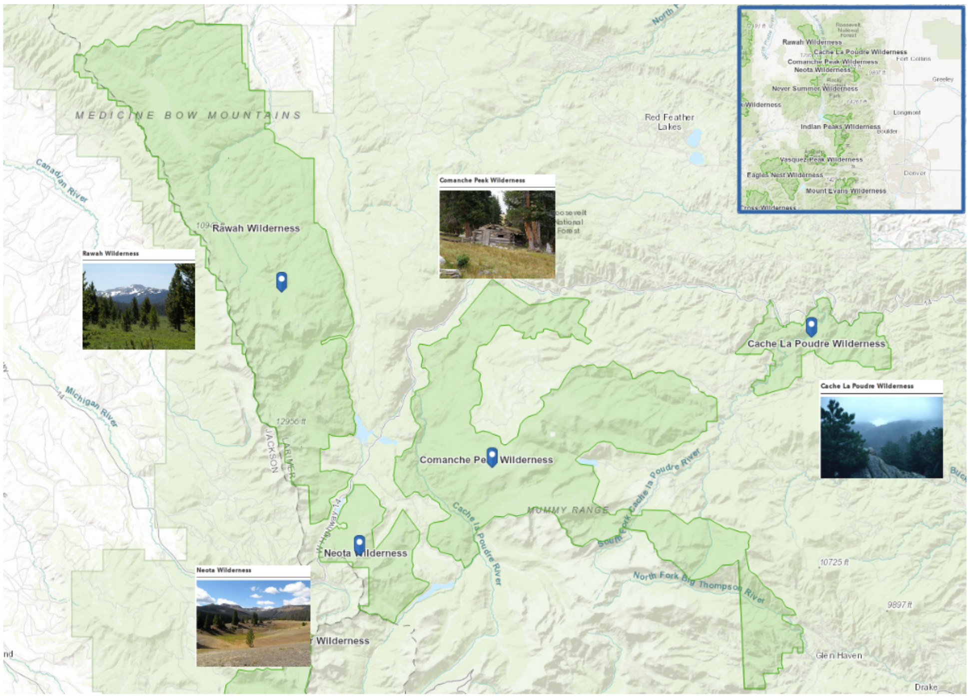
Zarandioon,-Reddy-Villanueva-DS7331-Lab1

January 28, 2019

DS7331 - Lab 1 Submitted by: Shravan Reddy, Samira Zarandioon, Jaime Villanueva
Forest Cover Type Analysis
Table of Contents

Introduction
The Roosevelt Natonal Forest is located about 100 miles northwest of Denver, Colorado, and is an area of more than 800,000 acres of land. The areas of interest in this forest for this analysis are the four wilderness areas, Rawah, Comanche Peak, Neota, and Cache la Poudre. A wilderness area is an official legal designation created by the Wilderness Preservation Act in 1964. This act created the Wilderness Preservation System and sets aside land areas in the United States to be managed and maintained in its natural wild state. This management is administered by four different government agencies: the National Park Service, the U.S. Forest Service (USFS), U.S. Fish and Wildlife Service, and the Bureau of Land Management.



images taken from : https://www.wilderness.net/NWPS/maps

Section **??** # Business Understanding

Being able to accurately catalog the natural resources of an area is important to land management agencies. In order to maintain the natural state of the forest, the natural resource managers are responsible for developing ecosystem management strategies. This process requires the collection of information from large areas of land in order to properly inventory an ecosystem, however the actual collection of such information can be time and cost prohibitive. Good predictive modeling can serve as an alternate method for creating these necessary inventories.

One of the most basic pieces of information that is collected from wildlife areas is the type of trees that are present. If a predictive model could take other attributes of the land that are either known or are easier to collect, and then use the information from these attributes to accurately predict the type of trees that would be found under those conditions, this has the potential to have a big cost and time saving for the federal agencies managing the area. The data in this analysis was derived from data originally obtained from US Geological Survey (USGS) and USFS data. The data comes from the aforementioned wilderness areas so they should have minimal human interactions, so we can be more confident that the current forest cover type is more a result of natural ecological processes rather than forest management practices. The data is a combination of information about the terrain with mapping information gathered by agencies using modeling software. Also the type of trees has been collected for these areas, so it is possible to develop a model based on the attributes to predict what type of land cover will be there. Then this can be compared against the actual cover types to get a sense of the accuracy of the model. The model may also be able to weed out any data that is not helpful in the determination of the cover type which could potentially have a cost saving as well. Since the there are several tree types, and the data is a collection of both numerical and categorical, potential models to use would be Linear Discriminant Analysis (LDA), Multi-nomial Regression, or some other classification algorithm such as Artificial Neural Network (ANN).

Section **??** # Data Understanding

Data Description

This data was taken from the UCI Machine Learning Archive: https://archive.ics.uci.edu/ml/datasets/covertype

Data can be downloaded from here: https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/

References for data information: https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/covtype.info http://web.cs.ucdavis.edu/~matloff/matloff/public_html/132/Data/ForestC

The four wilderness areas that the data was taken from vary greatly in size. The Rawah wilderness area is 73,213 acres. Comanche Peak is 67,680 acres. Neota is 9647 acres, and Cache la Poudre is 9433 acres. Each record is defined by a 30 x 30 meter cell from a computer model used by the USGS. This cell is directly defined by a digital elevation model (DEM) and is the source for the elevation attribute. All other attributes are based on this 30 x 30 meter cell. There are a total of 581,012 records each representing a cell.

There are ten numeric attributes which measure position relative to various features for each cell, as well as the amount of light at three different times of day during the summer solstice. The light measure is an index estimated by computer models. Also recorded is the amount of slope in each cell.

There are three different categorical attributes listed as well as two that are hidden. One of the categorical attributes is the cover type which is the variable that is being predicted. There are seven types of trees listed each being represented by an integer. The other categorical features are wilderness area and soil type both of which come dummy encoded in the data. There are four wilderness areas and forty soil types. The hidden categorical data are the climate zone and geologic zone which can be deduced from the USFS Ecological Landtype Unit (ELU) code listed for each soil type. The details for the attributes are listed below.

Attribute information:

Name	Data Type	Measurement	Description
Elevation	quantitative	meters	Elevation in meters
Aspect	quantitative	azimuth	Aspect in degrees azimuth
Slope	quantitative	degrees	Slope in degrees
Horizontal_Distance_To_Hydrology	quantitative	meters	Horz Dist to nearest surface water features
Vertical_Distance_To_Hydrology	quantitative	meters	Vert Dist to nearest surface water features
Horizontal_Distance_To_Roadways	quantitative	meters	Horz Dist to nearest roadway
Hillshade_9am	quantitative	0 to 255 index	Hillshade index at 9am, summer solstice
Hillshade_Noon	quantitative	0 to 255 index	Hillshade index at noon, summer soltice

Name	Data Type	Measurement	Description
Hillshade_3pm	quantitative	0 to 255 index	Hillshade index at 3pm, summer solstice
Horizontal_Distance_To_Fire_Points	quantitative	meters	Horz Dist to nearest wildfire ignition points
Rawah Wilderness Area	qualitative	0 (absence) or 1 (presence)	Wilderness area designation
Neota Wilderness Area	qualitative	0 (absence) or 1 (presence)	Wilderness area designation
Comanche Peak Wilderness Area	qualitative	0 (absence) or 1 (presence)	Wilderness area designation
Cache la Poudre Wilderness Area	qualitative	0 (absence) or 1 (presence)	Wilderness area designation
Soil_Type (40 binary columns)	qualitative	0 (absence) or 1 (presence)	Soil Type designation
Cover_Type (7 types)	integer	1 to 7	Forest Cover Type designation

Code Designations:
Soil Types: 1 to 40 : based on the USFS Ecological Landtype Units (ELUs) for this study area:

Study Code	USFS ELU Code	Description
1	2702	Cathedral family - Rock outcrop complex, extremely stony.
2	2703	Vanet - Ratake families complex, very stony.
3	2704	Haploborolis - Rock outcrop complex, rubbly.
4	2705	Ratake family - Rock outcrop complex, rubbly.
5	2706	Vanet family - Rock outcrop complex complex, rubbly.
6	2717	Vanet - Wetmore families - Rock outcrop complex, stony.
7	3501	Gothic family.
8	3502	Supervisor - Limber families complex.
9	4201	Troutville family, very stony.
10	4703	Bullwark - Catamount families - Rock outcrop complex, rubbly.

Study Code	USFS ELU Code	Description
11	4704	Bullwark - Catamount families - Rock land complex, rubbly.
12	4744	Legault family - Rock land complex, stony.
13	4758	Catamount family - Rock land - Bullwark family complex, rubbly.
14	5101	Pachic Argiborolis - Aquolis complex.
15	5151	unspecified in the USFS Soil and ELU Survey.
16	6101	Cryaquolis - Cryoborolis complex.
17	6102	Gateview family - Cryaquolis complex.
18	6731	Rogert family, very stony.
19	7101	Typic Cryaquolis - Borohemists complex.
20	7102	Typic Cryaquepts - Typic Cryaquolls complex.
21	7103	Typic Cryaquolls - Leighcan family, till substratum complex.
22	7201	Leighcan family, till substratum, extremely bouldery.
23	7202	Leighcan family, till substratum - Typic Cryaquolls complex.
24	7700	Leighcan family, extremely stony.
25	7701	Leighcan family, warm, extremely stony.
26	7702	Granile - Catamount families complex, very stony.
27	7709	Leighcan family, warm - Rock outcrop complex, extremely stony.
28	7710	Leighcan family - Rock outcrop complex, extremely stony.
29	7745	Como - Legault families complex, extremely stony.

Study Code	USFS ELU Code	Description
30	7746	Como family - Rock land - Legault family complex, extremely stony.
31	7755	Leighcan - Catamount families complex, extremely stony.
32	7756	Catamount family - Rock outcrop - Leighcan family complex, extremely stony.
33	7757	Leighcan - Catamount families - Rock outcrop complex, extremely stony.
34	7790	Cryorthents - Rock land complex, extremely stony.
35	8703	Cryumbrepts - Rock outcrop - Cryaquepts complex.
36	8707	Bross family - Rock land - Cryumbrepts complex, extremely stony.
37	8708	Rock outcrop - Cryumbrepts - Cryorthents complex, extremely stony.
38	8771	Leighcan - Moran families - Cryaquolls complex, extremely stony.
39	8772	Moran family - Cryorthents - Leighcan family complex, extremely stony.
40	8776	Moran family - Cryorthents - Rock land complex, extremely stony

Forest Cover Type Classes:

Measurement	Description
1	Spruce/Fir
2	Lodgepole Pine
3	Ponderosa Pine
4	Cottonwood/Willow
5	Aspen
6	Douglas-fir
7	Krummholz

Note:

First digit: climatic zone	Second digit: geologic zones

|
First digit
climatic zone
1
lower montane dry
2
lower montane
3
montane dry
4
montane
5
montane dry and montane
6
montane and subalpine
7
subalpine
8
alpine
|
Second digit
geologic zones
1
alluvium
2
glacial
3
shale
4
sandstone
5
mixed sedimentary
6
unspecified in the USFS ELU Survey
7
igneous and metamorphic
8
volcanic
Section **??** ## Data Quality

Because the data came from the a machine learning repository, it is already clean, but there are not any headings for the columns. It is advertised as having no missing data, and this is verified with code. Also the categorical predictors are already dummy coded. For certain analysis techniques, this is nice, but many of the visualizations planned seemed easier without the dummy coding. Therefore after the data is read and labeled, the categorical attributes were collapsed. Also the two hidden categorical attributes were added.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import warnings
        warnings.simplefilter('ignore', DeprecationWarning)
        warnings.simplefilter('ignore', FutureWarning)
```

```
In [2]: #Read in the .data file into a pandas dataframe
        df = pd.read_csv('data/covtype.data', header = None)
```

Check for missing data and duplicates

```
In [3]: # to check if there is any missing value inf df
        df.isnull().values.any()
        # there is no missing value
```

```
Out[3]: False
```

```
In [4]: # to get number of duplicated rows in df
        len(df[df.duplicated()])
        # there is no duplicated row
```

```
Out[4]: 0
```

Label the columns

```
In [5]: #Create Names for the columns based on covtypeinfo.txt
        quantitative = ['Elevation', 'Aspect', 'Slope', 'hDistance_to_Hydrology', 'vDistance_to_
                        'hDistance_to_Roads', 'Hillshade_9am', 'Hillshade_Noon', 'Hillshade_
        wilderness_area = ['Rawah', 'Neota', 'Comanche_Peak', 'Cache_la_Poudre']

        soil_type = ['Soil Type ' + str(i) for i in range(1,41)]

        cover_type = ['Cover_Type']

        #Assign names to columns
        df.columns = quantitative + wilderness_area + soil_type  + cover_type
```

Undoing the dummy coding

```
In [6]: #Preparing the dataframe with no dummy coding

        #Create separate df preparing for reversing dummy coding for categorical variables
        df_wilderness_area = df.iloc[:,10:14]
        df_soil_type = df.iloc[:,14:54]


        #Reverse dummy coding for wilderness area and soil type
        df['Wilderness_Area'] = pd.Series(df_wilderness_area.columns[np.where(df_wilderness_area
        df['Soil_Type'] = pd.Series(df_soil_type.columns[np.where(df_soil_type !=0)[1]])
```

Replacing cover type integer values with names

```
In [7]: #Map Cover Type Names into new column
        cover_type_map = {1:"Spruce/Fir", 2:"Lodgepole Pine", 3:"Ponderosa Pine", 4:"Cottonwood/
                          6:"Douglas-fir", 7:"Krummholz"}
        df['Cover Type Names'] = df['Cover_Type'].map(cover_type_map)
```

Creating climatic and geologic attributes from hidden categorical data

```
In [8]: #Map ELU codes into new column which will be used to generate columns for climatic and g
        elu_map = {"Soil Type 1": "2702", "Soil Type 2":"2703", "Soil Type 3": "2704", "Soil Typ
                   "Soil Type 6":"2717", "Soil Type 7":"3501", "Soil Type 8":"3502", "Soil Type
                   "Soil Type 11":"4704", "Soil Type 12":"4744", "Soil Type 13":"4758", "Soil Ty
                   "Soil Type 16":"6101", "Soil Type 17":"6102", "Soil Type 18":"6731", "Soil Ty
                   "Soil Type 21":"7103", "Soil Type 22":"7201", "Soil Type 23":"7202", "Soil Ty
                   "Soil Type 26":"7702", "Soil Type 27":"7709", "Soil Type 28":"7710", "Soil Ty
                   "Soil Type 31":"7755", "Soil Type 32":"7756", "Soil Type 33":"7757", "Soil Ty
                   "Soil Type 36":"8707", "Soil Type 37":"8708", "Soil Type 38":"8771", "Soil Ty
        df['ELU Codes'] = df['Soil_Type'].map(elu_map)

        #Create Climatic Zone column and map values into it
        climatic_map = {"1":"Lower Montane Dry", "2":"Lower Montane", "3":"Montane Dry", "4":"Mo
                        "5":"Montane Dry and Montane", "6":"Montane and Subalpine", "7":"Subalpine",

        climatic_zone = []
        for record in df['ELU Codes']:      #creates list from first digits of the ELU code
            climatic_zone.append(record[0])
        df['Climatic_Zone'] = climatic_zone #column is filled with first digits of ELU code
        df['Climatic_Zone'] = df['Climatic_Zone'].map(climatic_map) #map first digits to descrip

        #Create Geologic Zone column and map values into it
        geologic_map = {"1":"Alluvium", "2":"Glacial", "3":"Shale", "4":"Sandstone", "5":"Mixed
                        "7":"Igneous and Metamorphic", "8":"Volcanic"}

        geologic_zone = []
        for record in df['ELU Codes']:      #creates list from second digits of the ELU code
            geologic_zone.append(record[1])
        df['Geologic_Zone'] = geologic_zone #column is filled with first digits of ELU code
        df['Geologic_Zone'] = df['Geologic_Zone'].map(geologic_map) #map first digits to descrip
```

Dropping columns not needed, defining categorical types, and re-ordering columns

```
In [9]: #Drop dummy coded columns for wilderness area and soil type and the cover type column wi
        df = df.drop(wilderness_area, axis=1)
        df = df.drop(soil_type, axis=1)
        df = df.drop('Cover_Type', axis=1)
        df = df.drop('ELU Codes', axis=1)
```

```python
#Make categorical as category type
df['Wilderness_Area'] = df['Wilderness_Area'].astype('category')
df['Soil_Type'] = df['Soil_Type'].astype('category')
df['Cover Type Names'] = df['Cover Type Names'].astype('category')
df['Climatic_Zone'] = df['Climatic_Zone'].astype('category')
df['Geologic_Zone'] = df['Geologic_Zone'].astype('category')

#Make the response variable last and rename to Cover Type
df['Cover_Type'] = df['Cover Type Names']
df = df.drop('Cover Type Names', axis=1)
```

The final prepared dataset for analysis

In [10]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 581012 entries, 0 to 581011
Data columns (total 15 columns):
Elevation                    581012 non-null int64
Aspect                       581012 non-null int64
Slope                        581012 non-null int64
hDistance_to_Hydrology       581012 non-null int64
vDistance_to_Hydrology       581012 non-null int64
hDistance_to_Roads           581012 non-null int64
Hillshade_9am                581012 non-null int64
Hillshade_Noon               581012 non-null int64
Hillshade_3pm                581012 non-null int64
hDistance_to_Fire Points     581012 non-null int64
Wilderness_Area              581012 non-null category
Soil_Type                    581012 non-null category
Climatic_Zone                581012 non-null category
Geologic_Zone                581012 non-null category
Cover_Type                   581012 non-null category
dtypes: category(5), int64(10)
memory usage: 47.1 MB
```

In [11]:
```python
df.head().transpose()
```

Out[11]:

	0	1 \
Elevation	2596	2590
Aspect	51	56
Slope	3	2
hDistance_to_Hydrology	258	212
vDistance_to_Hydrology	0	-6
hDistance_to_Roads	510	390
Hillshade_9am	221	220
Hillshade_Noon	232	235

```
Hillshade_3pm                                       148                       151
hDistance_to_Fire Points                           6279                      6225
Wilderness_Area                                   Rawah                     Rawah
Soil_Type                                  Soil Type 29              Soil Type 29
Climatic_Zone                                 Subalpine                 Subalpine
Geologic_Zone                  Igneous and Metamorphic  Igneous and Metamorphic
Cover_Type                                        Aspen                     Aspen

                                                      2                         3  \
Elevation                                          2804                      2785
Aspect                                              139                       155
Slope                                                 9                        18
hDistance_to_Hydrology                              268                       242
vDistance_to_Hydrology                               65                       118
hDistance_to_Roads                                 3180                      3090
Hillshade_9am                                       234                       238
Hillshade_Noon                                      238                       238
Hillshade_3pm                                       135                       122
hDistance_to_Fire Points                           6121                      6211
Wilderness_Area                                   Rawah                     Rawah
Soil_Type                                  Soil Type 12              Soil Type 30
Climatic_Zone                                   Montane                 Subalpine
Geologic_Zone                  Igneous and Metamorphic  Igneous and Metamorphic
Cover_Type                               Lodgepole Pine            Lodgepole Pine

                                                      4
Elevation                                          2595
Aspect                                               45
Slope                                                 2
hDistance_to_Hydrology                              153
vDistance_to_Hydrology                               -1
hDistance_to_Roads                                  391
Hillshade_9am                                       220
Hillshade_Noon                                      234
Hillshade_3pm                                       150
hDistance_to_Fire Points                           6172
Wilderness_Area                                   Rawah
Soil_Type                                  Soil Type 29
Climatic_Zone                                 Subalpine
Geologic_Zone                  Igneous and Metamorphic
Cover_Type                                        Aspen
```

Before doing any analysis we check again to make sure there is no missing or duplicate data after all our data manipulation. If there were either of those, that record would have to be investigated to see if it should be kept or deleted.

```
In [12]: # to check if there is any missing value inf df
         df.isnull().values.any()
         # there is no missing value
```

Out[12]: False

In [13]:
```
# to get number of duplicated rows in df
len(df[df.duplicated()])
# there is no duplicated row
```

Out[13]: 0

Outliers The outliers are probably better determined by looking at quick box plots and comparing to basic statistics, but we run some arbitrary numbers first to just get a feel for the data. We used nine times the standard deviation as the benchmark for outlier. Of course, whether this is a big number and whether it is an outlier or not depends on the spread. Do it this way returned forty-five outliers which considering the size of the data set would seem pretty good. But we will a more visual approach as well.

In [15]:
```
numeric_df = df[quantitative]
outliers = numeric_df[(np.abs( numeric_df-numeric_df.mean())> (9*numeric_df.std())).any
outliers
```

Out[15]:

	Elevation	Aspect	Slope	hDistance_to_Hydrology \
220084	2954	290	31	845
220445	2960	286	27	854
220812	2963	279	23	864
221187	2949	288	28	847
221188	2963	283	19	874
221567	2955	293	25	859
221956	2949	305	23	845
221957	2960	295	21	872
222355	2948	311	23	832
222356	2956	308	19	859
222357	2964	296	17	886
222774	2946	302	26	819
222775	2956	310	20	845
222776	2962	311	15	872
222777	2968	297	12	899
223207	2953	288	23	834
223208	2962	295	15	860
223209	2967	299	10	886
223210	2970	291	7	912
223652	2954	269	21	849
223653	2963	269	13	875
223654	2967	272	8	900
223655	2971	275	6	927
223885	2506	13	64	201
223886	2501	3	63	216
223887	2500	0	62	234
224109	2952	259	22	865
224110	2962	261	15	890

224111	2967	270	9	916
224112	2971	285	7	942
224573	2949	259	24	882
224574	2960	262	17	907
224575	2968	274	12	932
224576	2972	294	9	957
225045	2959	269	19	924
225046	2968	277	15	949
225047	2975	289	11	973
225517	2959	275	21	942
225518	2970	281	17	960
479525	3159	60	37	150
479789	3281	38	59	150
479790	3158	73	62	170
480340	3147	96	59	216
482917	3094	82	65	42
483577	3083	105	57	0

	vDistance_to_Hydrology	hDistance_to_Roads	Hillshade_9am \
220084	581	939	121
220445	588	953	134
220812	589	960	152
221187	573	940	132
221188	588	968	164
221567	583	949	143
221956	574	930	149
221957	587	959	155
222355	576	914	151
222356	585	942	165
222357	589	969	170
222774	574	899	138
222775	586	926	163
222776	590	953	179
222777	597	981	186
223207	578	912	150
223208	590	939	178
223209	597	966	193
223210	598	993	199
223652	577	927	160
223653	588	953	186
223654	595	979	199
223655	601	1006	205
223885	88	655	73
223886	81	626	55
223887	83	598	54
224109	573	942	161
224110	585	967	183
224111	592	994	197

224112	599	1020	202
224573	574	957	156
224574	581	983	175
224575	591	1008	189
224576	597	1034	194
225045	584	999	167
225046	589	1024	179
225047	598	1050	190
225517	582	1015	159
225518	595	1040	172
479525	0	3045	220
479789	123	3012	137
479790	-4	3042	191
480340	-6	3037	220
482917	3	3001	193
483577	0	3002	228

	Hillshade_Noon	Hillshade_3pm	hDistance_to_Fire Points
220084	219	230	2438
220445	226	227	2408
220812	236	221	2379
221187	225	227	2343
221188	237	212	2350
221567	225	219	2314
221956	220	208	2278
221957	229	212	2285
222355	217	202	2242
222356	225	199	2249
222357	233	202	2256
222774	217	215	2206
222775	223	198	2213
222776	228	189	2219
222777	236	190	2226
223207	231	218	2177
223208	235	197	2183
223209	237	184	2190
223210	239	179	2197
223652	242	219	2148
223653	244	197	2154
223654	243	182	2161
223655	241	176	2168
223885	30	0	1470
223886	40	0	1470
223887	45	67	1471
224109	246	220	2118
224110	247	202	2125
224111	243	186	2132
224112	240	178	2139

224573	245	223	2089
224574	246	209	2096
224575	243	193	2103
224576	238	184	2110
225045	243	214	2067
225046	242	202	2074
225047	239	189	2081
225517	240	218	2037
225518	239	206	2045
479525	0	17	1177
479789	42	0	1159
479790	0	0	1187
480340	0	0	1209
482917	0	0	1315
483577	0	0	1350

In [16]:
```
# fields that their value is >= 9
outliers_fields = np.abs(outliers-numeric_df.mean())/numeric_df.std()
outliers_fields
```

Out[16]:

	Elevation	Aspect	Slope	hDistance_to_Hydrology \
220084	0.019163	1.200418	2.256377	2.707944
220445	0.002267	1.164676	1.722206	2.750287
220812	0.012982	1.102128	1.188035	2.797335
221187	0.037021	1.182547	1.855749	2.717354
221188	0.012982	1.137869	0.653865	2.844383
221567	0.015591	1.227224	1.455121	2.773811
221956	0.037021	1.334449	1.188035	2.707944
221957	0.002267	1.245095	0.920950	2.834973
222355	0.040593	1.388062	1.188035	2.646782
222356	0.012020	1.361256	0.653865	2.773811
222357	0.016553	1.254030	0.386779	2.900841
222774	0.047736	1.307643	1.588664	2.585620
222775	0.012020	1.379127	0.787407	2.707944
222776	0.009410	1.388062	0.119694	2.834973
222777	0.030840	1.262966	0.280934	2.962003
223207	0.022734	1.182547	1.188035	2.656191
223208	0.009410	1.245095	0.119694	2.778516
223209	0.027268	1.280837	0.548020	2.900841
223210	0.037983	1.209353	0.948648	3.023165
223652	0.019163	1.012773	0.920950	2.726763
223653	0.012982	1.012773	0.147392	2.849088
223654	0.027268	1.039579	0.815105	2.966708
223655	0.041555	1.066386	1.082190	3.093737
223885	1.619250	1.274703	6.663286	0.321940
223886	1.637108	1.364058	6.529743	0.251369
223887	1.640680	1.390864	6.396201	0.166682
224109	0.026306	0.923418	1.054493	2.802040

224110	0.009410	0.941289	0.119694	2.919660
224111	0.027268	1.021708	0.681562	3.041984
224112	0.041555	1.155740	0.948648	3.164309
224573	0.037021	0.923418	1.321578	2.882021
224574	0.002267	0.950225	0.386779	2.999641
224575	0.030840	1.057450	0.280934	3.117261
224576	0.045126	1.236159	0.681562	3.234881
225045	0.001305	1.012773	0.653865	3.079623
225046	0.030840	1.084257	0.119694	3.197242
225047	0.055841	1.191482	0.414477	3.310157
225517	0.001305	1.066386	0.920950	3.164309
225518	0.037983	1.119998	0.386779	3.248995
479525	0.713020	0.854737	3.057633	0.561885
479789	1.148758	1.051317	5.995572	0.561885
479790	0.709448	0.738576	6.396201	0.467789
480340	0.670160	0.533061	5.995572	0.251369
482917	0.480864	0.658157	6.796829	1.070002
483577	0.441577	0.452642	5.728487	1.267603

	vDistance_to_Hydrology	hDistance_to_Roads	Hillshade_9am	\
220084	9.170238	0.905013	3.404797	
220445	9.290316	0.896035	2.919177	
220812	9.307470	0.891545	2.246780	
221187	9.033005	0.904372	2.993888	
221188	9.290316	0.886415	1.798515	
221567	9.204546	0.898600	2.582979	
221956	9.050160	0.910785	2.358846	
221957	9.273162	0.892187	2.134714	
222355	9.084468	0.921047	2.284135	
222356	9.238854	0.903089	1.761160	
222357	9.307470	0.885773	1.574383	
222774	9.050160	0.930667	2.769756	
222775	9.256008	0.913351	1.835870	
222776	9.324624	0.896035	1.238184	
222777	9.444703	0.878077	0.976696	
223207	9.118776	0.922329	2.321491	
223208	9.324624	0.905013	1.275539	
223209	9.444703	0.887697	0.715208	
223210	9.461857	0.870382	0.491076	
223652	9.101622	0.912709	1.947937	
223653	9.290316	0.896035	0.976696	
223654	9.410395	0.879360	0.491076	
223655	9.513319	0.862044	0.266944	
223885	0.713286	1.087152	5.197857	
223886	0.593207	1.105750	5.870254	
223887	0.627515	1.123708	5.907609	
224109	9.033005	0.903089	1.910581	
224110	9.238854	0.887056	1.088762	

224111	9.358933	0.869740	0.565787
224112	9.479011	0.853066	0.379010
224573	9.050160	0.893469	2.097358
224574	9.170238	0.876795	1.387606
224575	9.341779	0.860762	0.864630
224576	9.444703	0.844087	0.677853
225045	9.221700	0.866534	1.686449
225046	9.307470	0.850500	1.238184
225047	9.461857	0.833826	0.827275
225517	9.187392	0.856272	1.985292
225518	9.410395	0.840239	1.499672
479525	0.796272	0.445632	0.293388
479789	1.313678	0.424468	2.807111
479790	0.864888	0.443708	0.789919
480340	0.899196	0.440501	0.293388
482917	0.744810	0.417413	0.715208
483577	0.796272	0.418054	0.592231

	Hillshade_Noon	Hillshade_3pm	hDistance_to_Fire Points
220084	0.218462	2.285377	0.345651
220445	0.135633	2.206996	0.322995
220812	0.641483	2.050234	0.301095
221187	0.085048	2.206996	0.273909
221188	0.692068	1.815091	0.279195
221567	0.085048	1.997980	0.252009
221956	0.167877	1.710582	0.224822
221957	0.287388	1.815091	0.230109
222355	0.319632	1.553820	0.197636
222356	0.085048	1.475439	0.202922
222357	0.489728	1.553820	0.208209
222774	0.319632	1.893472	0.170450
222775	0.016122	1.449312	0.175736
222776	0.236803	1.214169	0.180267
222777	0.641483	1.240296	0.185553
223207	0.388558	1.971853	0.148550
223208	0.590898	1.423185	0.153081
223209	0.692068	1.083534	0.158367
223210	0.793238	0.952898	0.163653
223652	0.944993	1.997980	0.126650
223653	1.046163	1.423185	0.131181
223654	0.995578	1.031279	0.136467
223655	0.894408	0.874517	0.141753
223885	9.779032	3.723841	0.385360
223886	9.273181	3.723841	0.385360
223887	9.020256	1.973330	0.384604
224109	1.147333	2.024107	0.103994
224110	1.197918	1.553820	0.109281
224111	0.995578	1.135788	0.114567

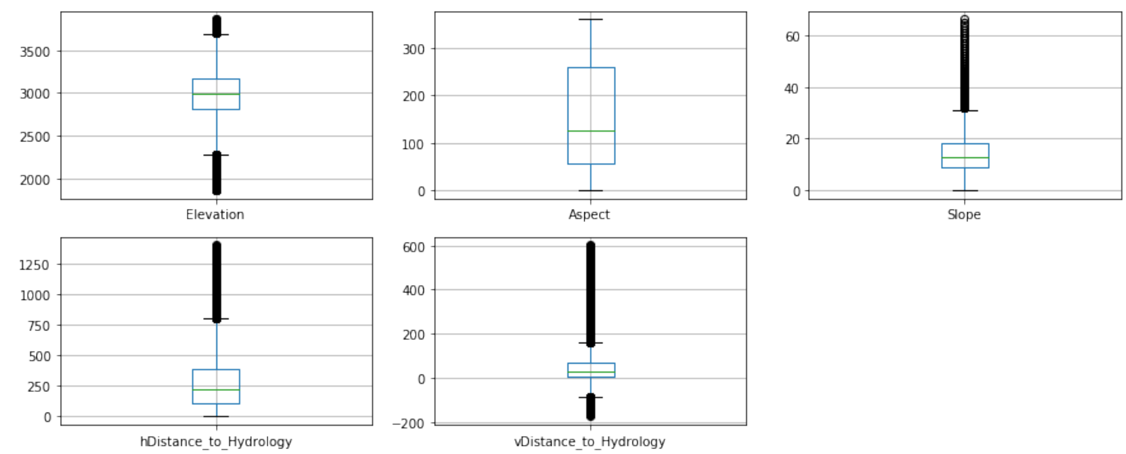
```
224112    0.843823    0.926771    0.119853
224573    1.096748    2.102488    0.082094
224574    1.147333    1.736709    0.087380
224575    0.995578    1.318677    0.092667
224576    0.742653    1.083534    0.097953
225045    0.995578    1.867345    0.065480
225046    0.944993    1.553820    0.070767
225047    0.793238    1.214169    0.076053
225517    0.843823    1.971853    0.042825
225518    0.793238    1.658328    0.048866
479525   11.296582    3.279681    0.606626
479789    9.172011    3.723841    0.620219
479790   11.296582    3.723841    0.599074
480340   11.296582    3.723841    0.582460
482917   11.296582    3.723841    0.502412
483577   11.296582    3.723841    0.475981
```

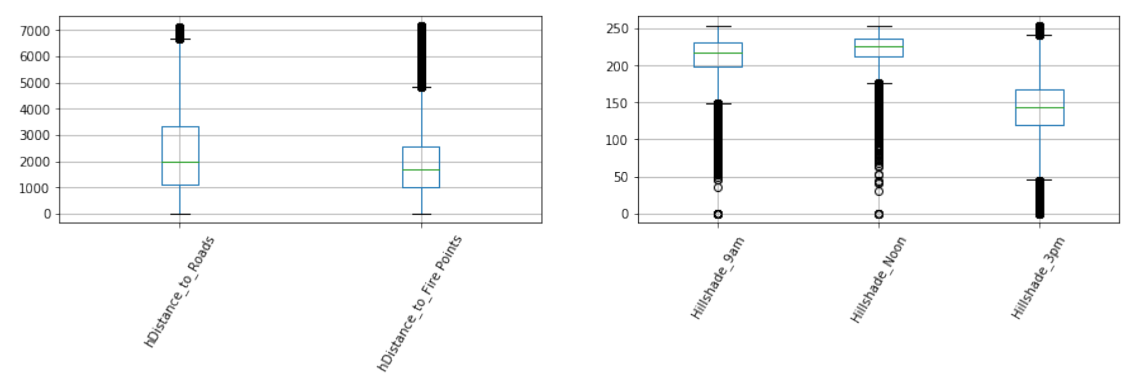
In [17]:
```
outliers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 45 entries, 220084 to 483577
Data columns (total 10 columns):
Elevation                   45 non-null int64
Aspect                      45 non-null int64
Slope                       45 non-null int64
hDistance_to_Hydrology      45 non-null int64
vDistance_to_Hydrology      45 non-null int64
hDistance_to_Roads          45 non-null int64
Hillshade_9am               45 non-null int64
Hillshade_Noon              45 non-null int64
Hillshade_3pm               45 non-null int64
hDistance_to_Fire Points    45 non-null int64
dtypes: int64(10)
memory usage: 3.9 KB
```

In [18]:
```
# Boxplots of quantitative attributes
%matplotlib inline
vars_to_plot_separate1 = [['Elevation'],
                          ['Aspect'],
                          ['Slope'],
                          ['hDistance_to_Hydrology'],
                          ['vDistance_to_Hydrology']]

vars_to_plot_separate2 = [['hDistance_to_Roads','hDistance_to_Fire Points'],
                          ['Hillshade_9am', 'Hillshade_Noon', 'Hillshade_3pm']]
plt.figure(figsize=(15, 6))
for index, plot_vars in enumerate(vars_to_plot_separate1):
```

```
    plt.subplot(len(vars_to_plot_separate1)/2,
                3,
                index+1)
    ax = df.boxplot(column=plot_vars)
plt.show()

plt.figure(figsize=(15, 3))
for index, plot_vars in enumerate(vars_to_plot_separate2):
    plt.subplot(len(vars_to_plot_separate2)/2,
                2,
                index+1)
    ax = df.boxplot(column=plot_vars)
    plt.xticks(rotation=60)
plt.show()
```





Above we can see that many of the attributes display some heavy tails creating skew. Aspect is the only attribute not showing any outliers. This will be verified in the next section with the distribution plots.

Section **??** ## Basic Statistics and Visualizations

First we will consider individually the numeric data, and then the categorical data. There are some basic statistics in the numeric data which are interesting.

First aspect which was the only attribute in the box plots to not show heavy taling has a standard deviation almost as big as its mean. Since this in degrees and is an angular measure from a reference, this would make the range from 0 t0 360. This means that all the range of values of aspect are about three standard deviations apart. Because this is a positioning measure from a reference point, this may not be significant, but the variance is noteworthy. Other attributes have standard deviations close to their mean and reach out of the interquartile range within a couple of standard deviations, but none are as extreme aspect.

The indexes for hillshade are interesting because, like aspect, the values are bounded; the index for hillshade is between 0 and 254. But unlike aspect, these indexes have a standard deviation that is much smaller than the mean and the interquartile range is pretty tight. The standard deviation is smallest for the measurement at noon when the sun is near directly overhead. This would produce the least amount of variation in the measurement and that is reflected in the values.

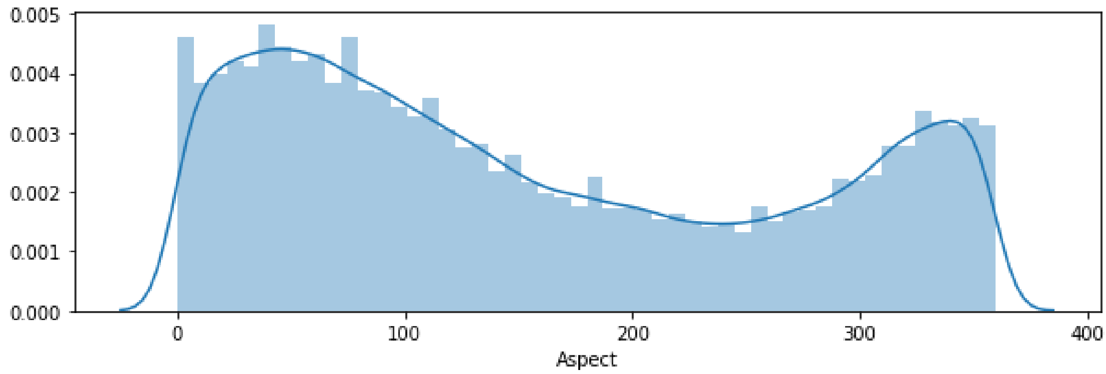
The last attribute that jumps out is the vertical distance to hydrology because it ranges into negative numbers. This means that the sometimes the mean distance is sometimes higher or lower than the water area.

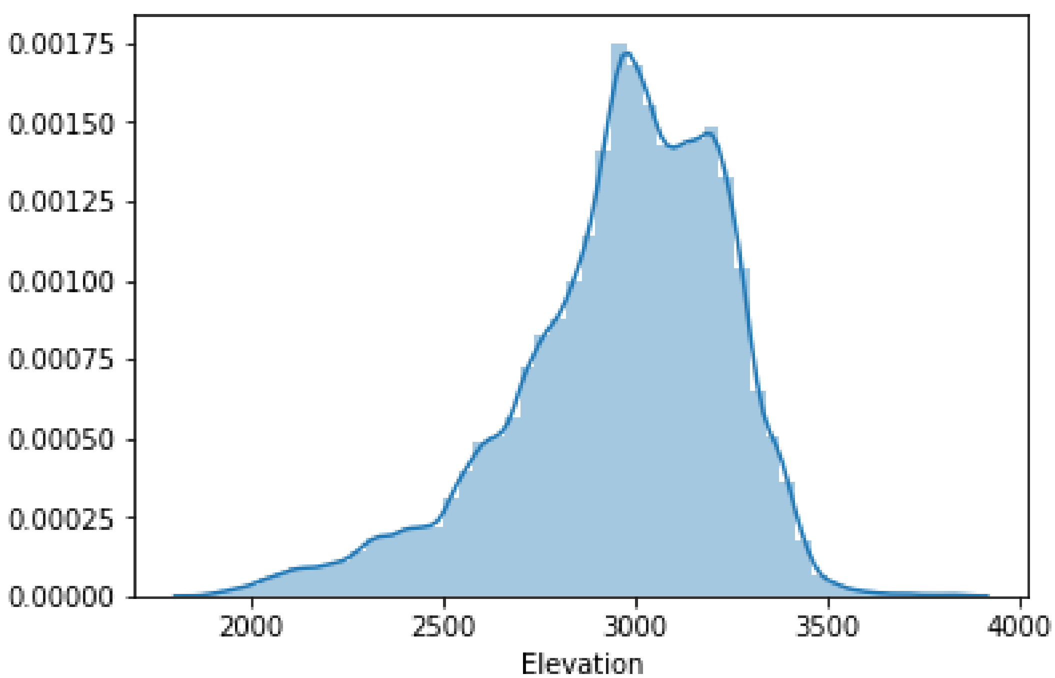
```
In [22]: df[quantitative].describe().transpose()
```

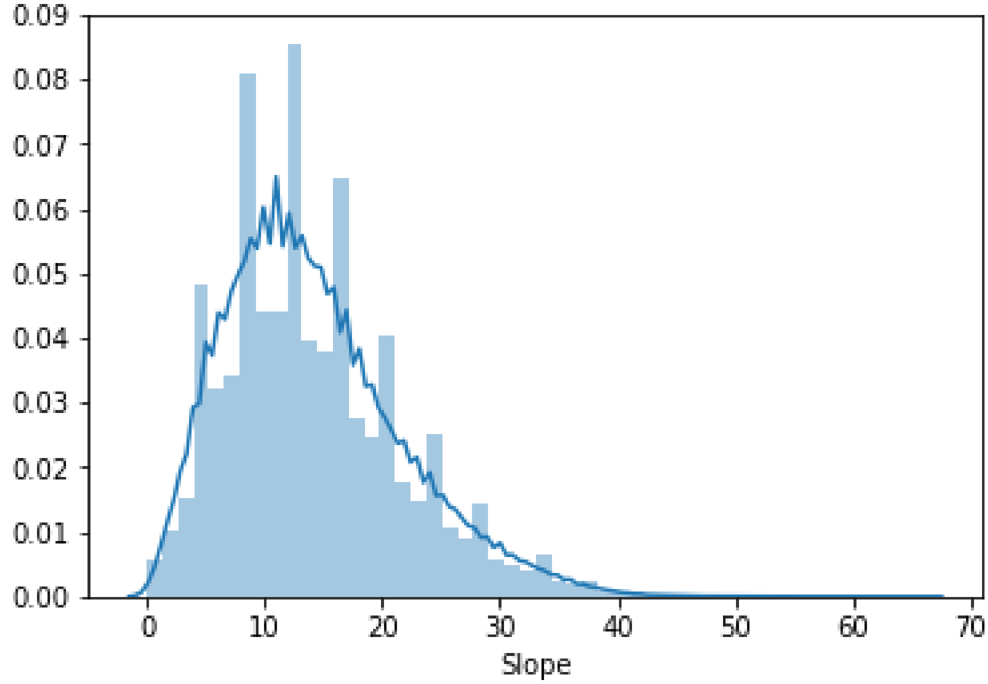
```
Out[22]:                             count         mean          std     min     25%  \
         Elevation                581012.0  2959.365301   279.984734  1859.0  2809.0
         Aspect                   581012.0   155.656807   111.913721     0.0    58.0
         Slope                    581012.0    14.103704     7.488242     0.0     9.0
         hDistance_to_Hydrology   581012.0   269.428217   212.549356     0.0   108.0
         vDistance_to_Hydrology   581012.0    46.418855    58.295232  -173.0     7.0
         hDistance_to_Roads       581012.0  2350.146611  1559.254870     0.0  1106.0
         Hillshade_9am            581012.0   212.146049    26.769889     0.0   198.0
         Hillshade_Noon           581012.0   223.318716    19.768697     0.0   213.0
         Hillshade_3pm            581012.0   142.528263    38.274529     0.0   119.0
         hDistance_to_Fire Points 581012.0  1980.291226  1324.195210     0.0  1024.0

                                     50%     75%     max
         Elevation                2996.0  3163.0  3858.0
         Aspect                    127.0   260.0   360.0
         Slope                      13.0    18.0    66.0
         hDistance_to_Hydrology    218.0   384.0  1397.0
         vDistance_to_Hydrology     30.0    69.0   601.0
         hDistance_to_Roads       1997.0  3328.0  7117.0
         Hillshade_9am             218.0   231.0   254.0
         Hillshade_Noon            226.0   237.0   254.0
         Hillshade_3pm             143.0   168.0   254.0
         hDistance_to_Fire Points 1710.0  2550.0  7173.0
```

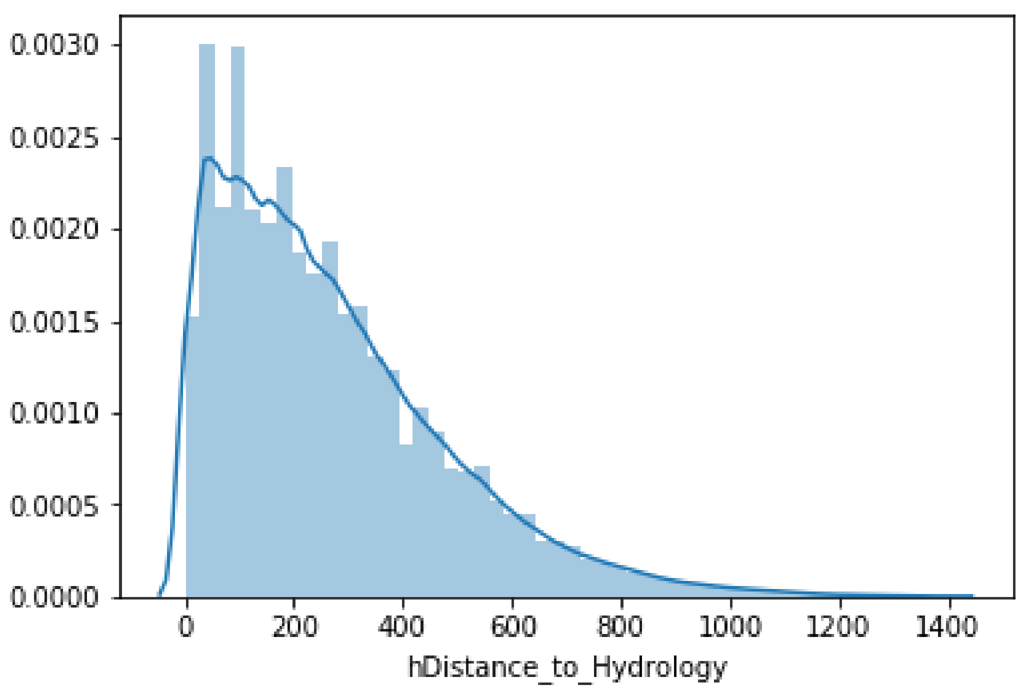
Distribution plots of all the numeric data were performed and it verifies what we saw in the box plots and in the statistical summaries. The variation for aspect is big and there is heavy tailing present for most of the other distributions. Only the amount of light at 3:00 PM looks approximately normal. This real data emphasizes the importance of the central limit theorem if needing to do an analysis where one of the assumptions is normality.

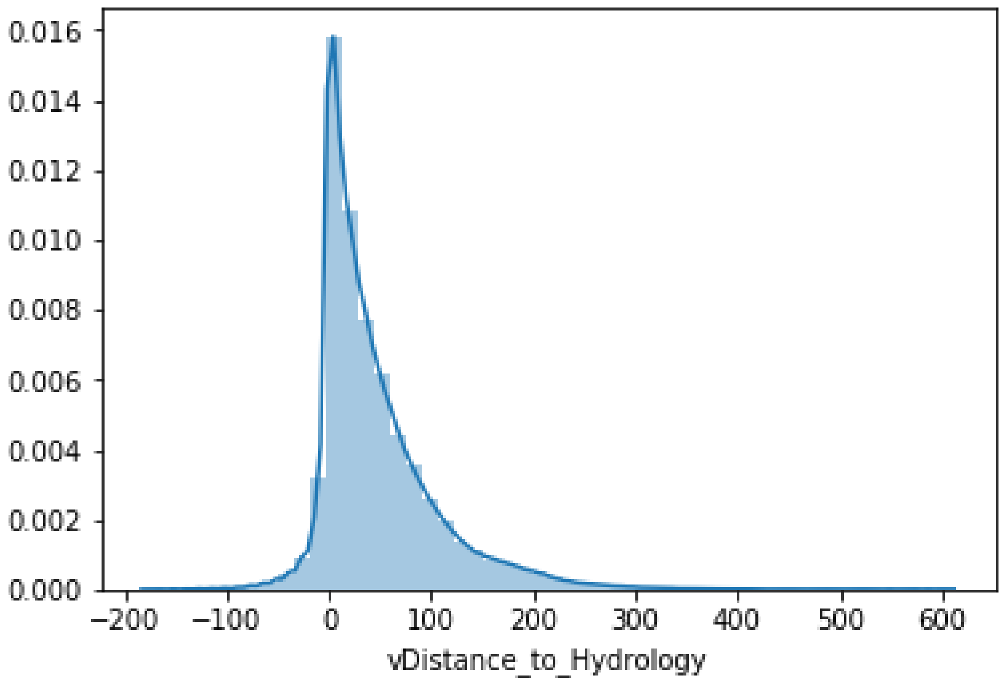
```
In [23]: %matplotlib inline
         plt.figure(figsize=(10,3))
         for i, col in enumerate(df[quantitative]):
             plt.figure(i)
             sns.distplot(df[quantitative][col])
```

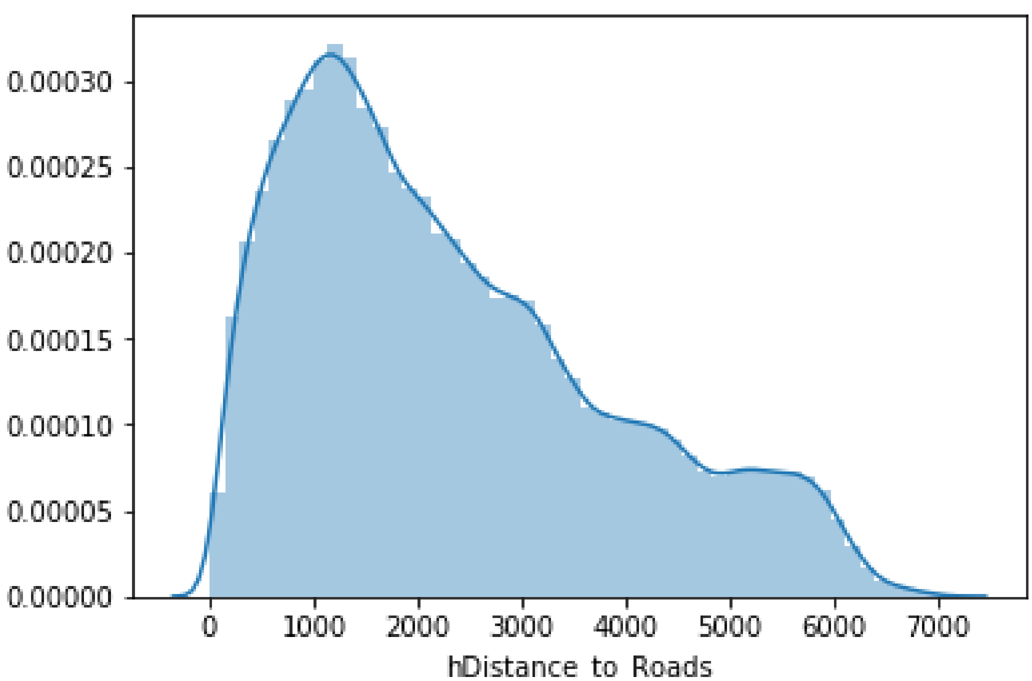


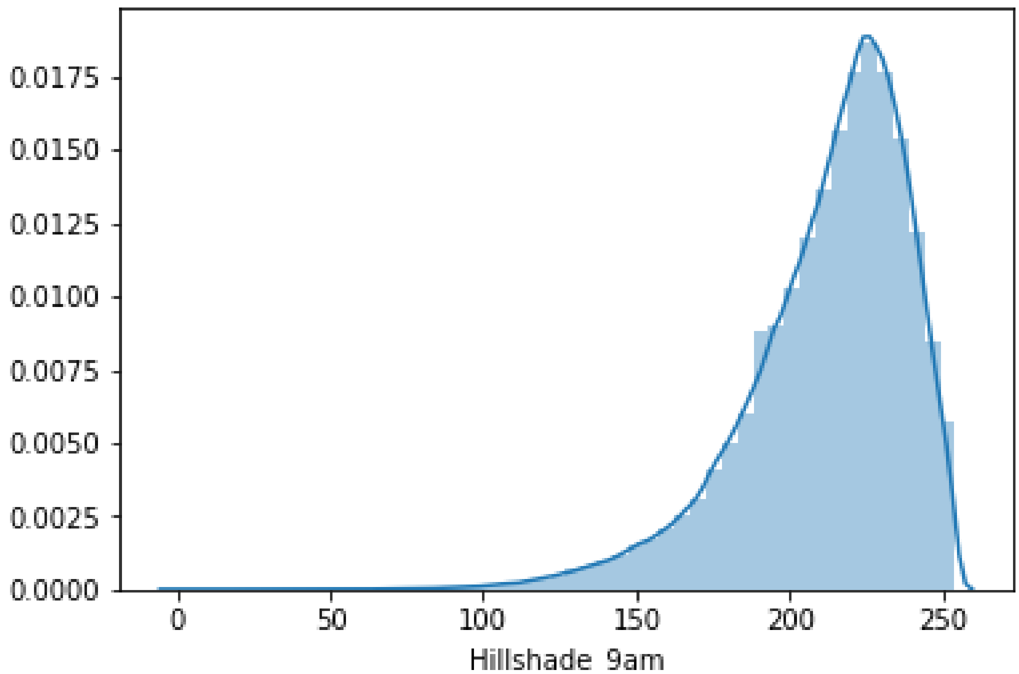


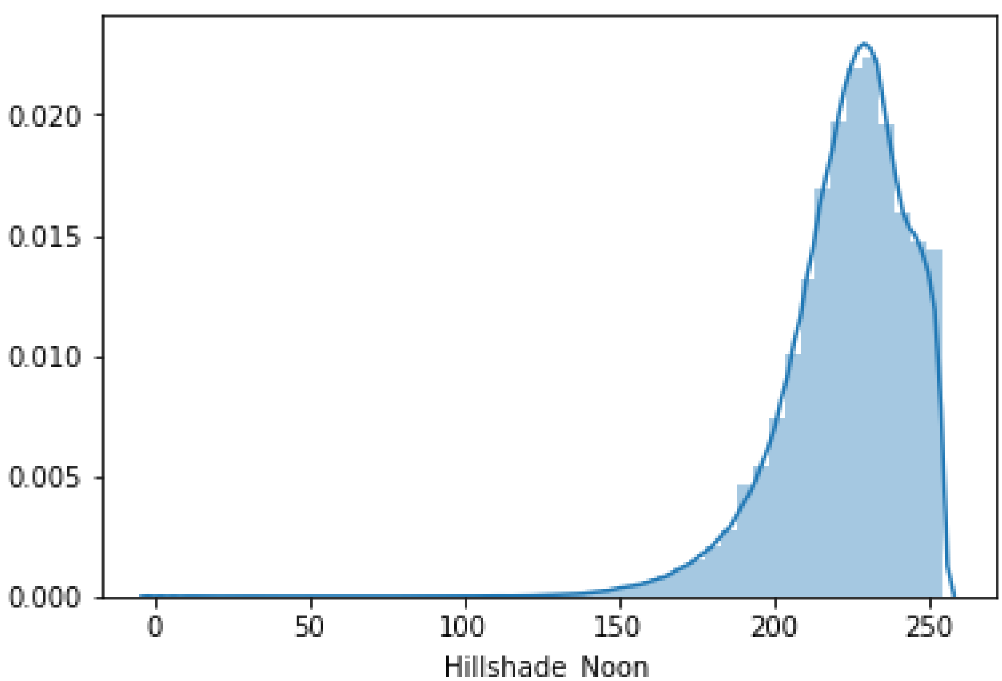

0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
50
60
70
Slope

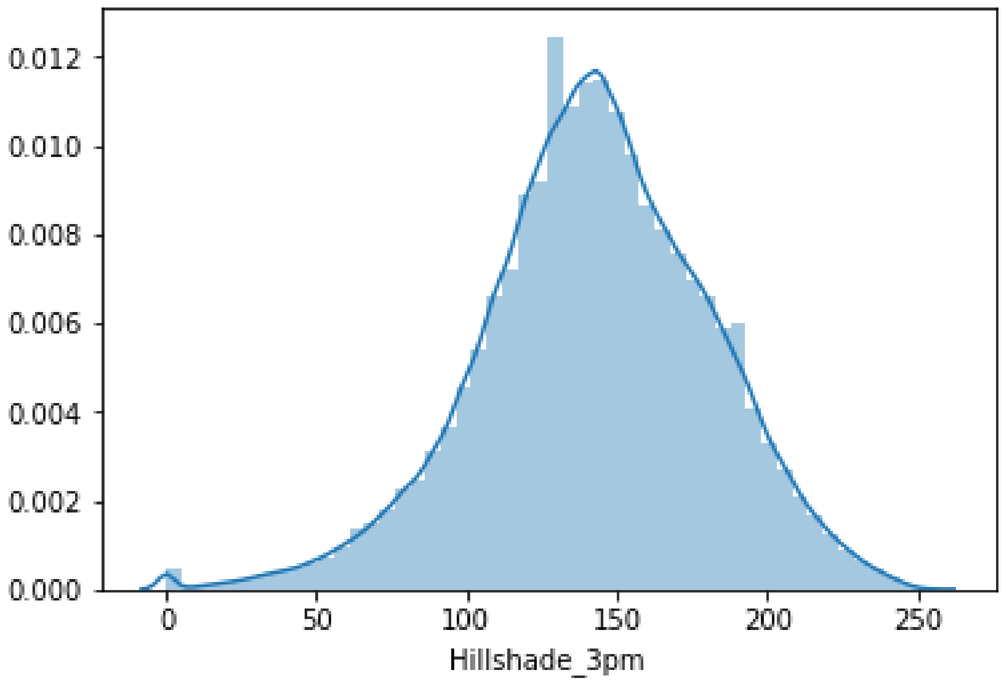

0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
200
400
600
800
1000
1200
1400
hDistance_to_Hydrology

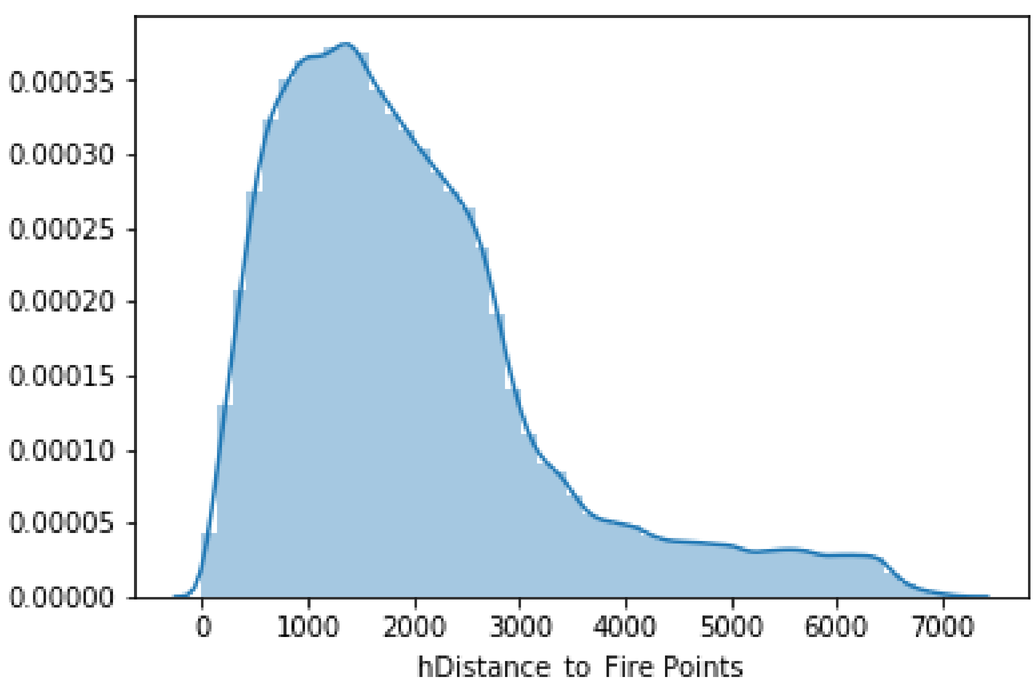

0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
−200
−100
0
100
200
300
400
500
600
vDistance_to_Hydrology


0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0
1000
2000
3000
4000
5000
6000
7000
hDistance_to_Roads


0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0
50
100
150
200
250
Hillshade_9am


0.000
0.005
0.010
0.015
0.020
0
50
100
150
200
250
Hillshade_Noon



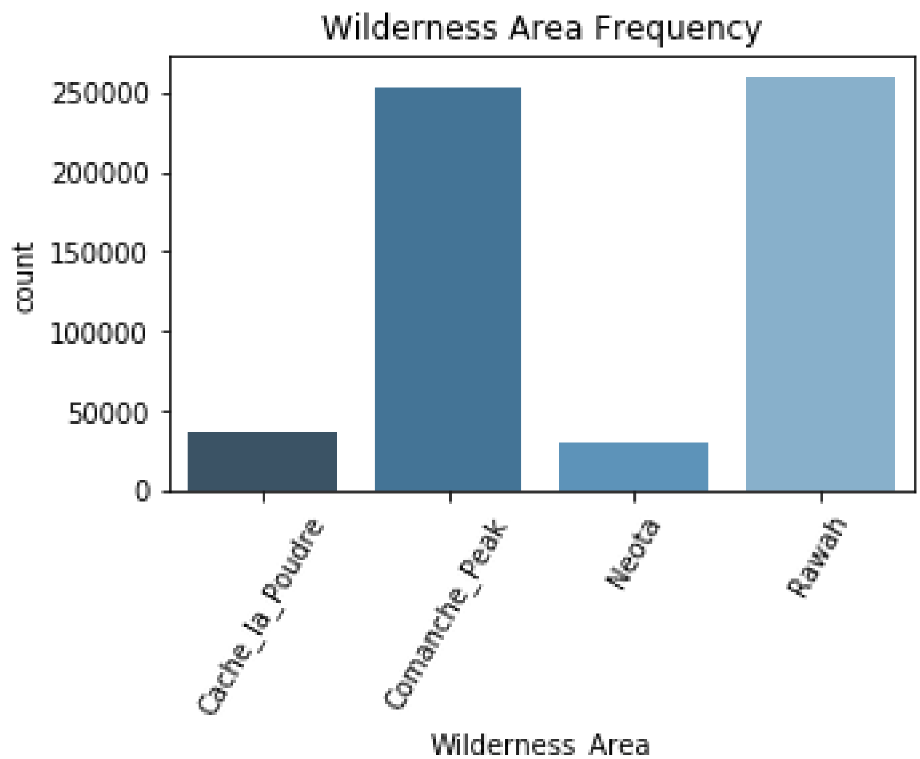


For the categorical data individual analysis is done with basic frequency counts displayed by bar graphs. The first graph clearly shows that most of the data comes from the Rawah and

Comanch Peak wilderness areas. This is expected because the acreage from those two areas is larger and thus there can be more 30 x 30 m cells.

```
In [17]: %matplotlib inline
         plt.figure(figsize=(5,3))
         ax = sns.countplot(x="Wilderness_Area",data=df, palette="Blues_d")
         ax.set_title('Wilderness Area Frequency')
         plt.xticks(rotation=60)
```

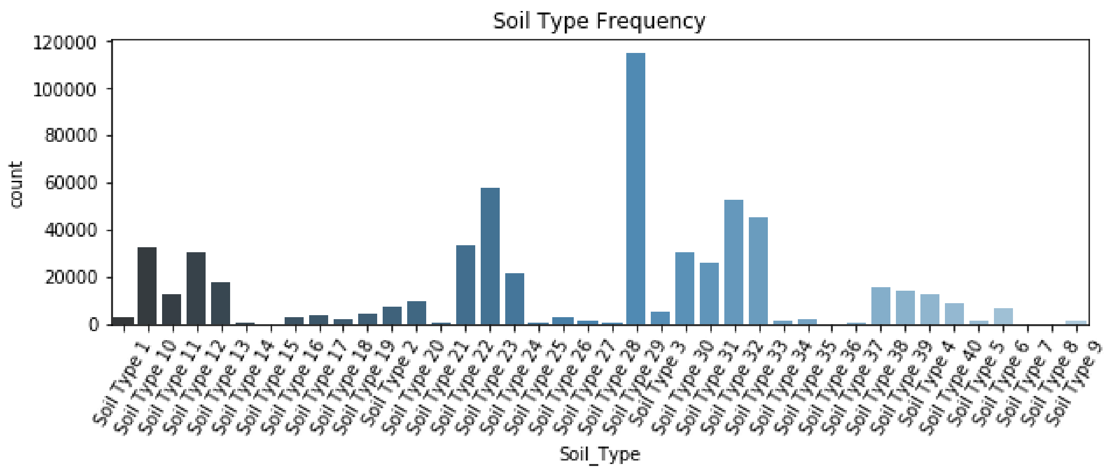
```
Out[17]: (array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)
```



The next one is soil type, and though there are forty soil types, from the graph, we can estimate the most of the records come from about 25% of the soil types. The most prevalent soil type is number 29 which correlates to: Como - Legault families complex, extremely stony Since this is a stony area, it would be interesting in the relations section to see what wilderness area and tree type grow from this soil. The graph is slightly offset making it look like soil type 28, but running the numbers with a cross-tab show that it should be soil type 29.

```
In [18]: %matplotlib inline
         plt.figure(figsize=(10,3))
         ax = sns.countplot(x="Soil_Type",data=df, palette="Blues_d")
         ax.set_title('Soil Type Frequency')
         plt.xticks(rotation=60)
```

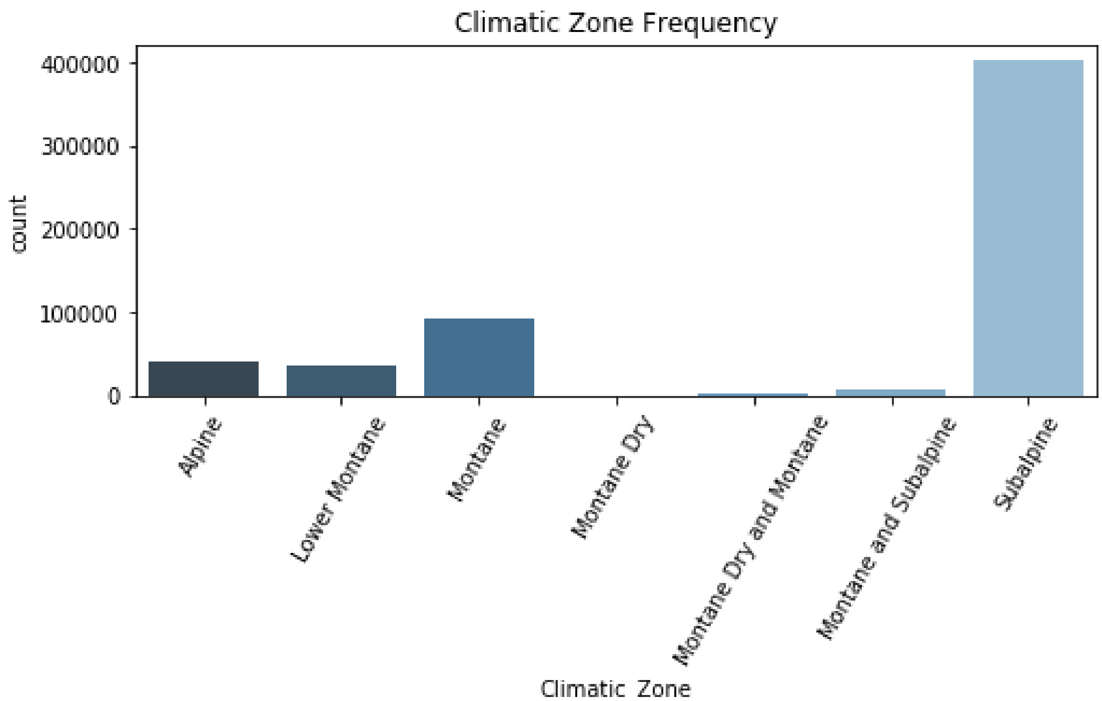
Out[18]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39]), <a list of 40 Text xticklabel objects>)



The bar graph shows that a majority of the records come from the subalpine climate zone. The subalpine zone is just below the tree line at high elevations (9000-12000 ft.) and is cool year round.

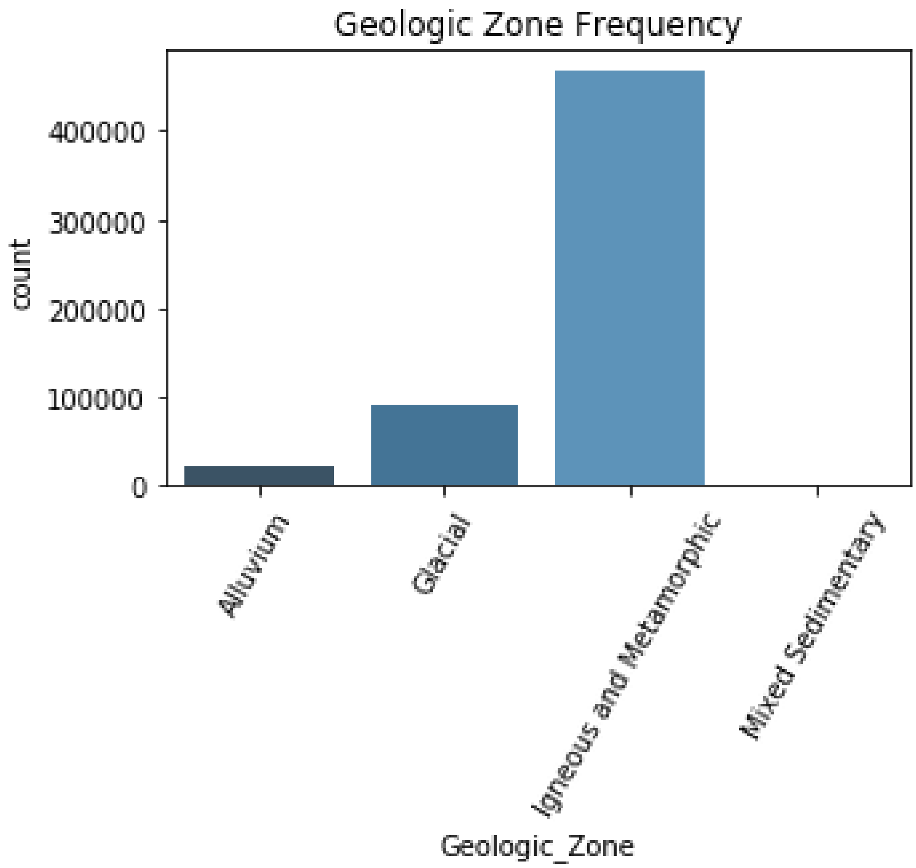
```
In [19]: %matplotlib inline
         plt.figure(figsize=(8,3))
         ax = sns.countplot(x="Climatic_Zone",data=df, palette="Blues_d")
         ax.set_title('Climatic Zone Frequency')
         plt.xticks(rotation=60)
```

Out[19]: (array([0, 1, 2, 3, 4, 5, 6]), <a list of 7 Text xticklabel objects>)



There is mostly igneous and metamorphic rock in from these wilderness areas. This corresponds with the prevalence of soil type 29 seen in the soil type chart.

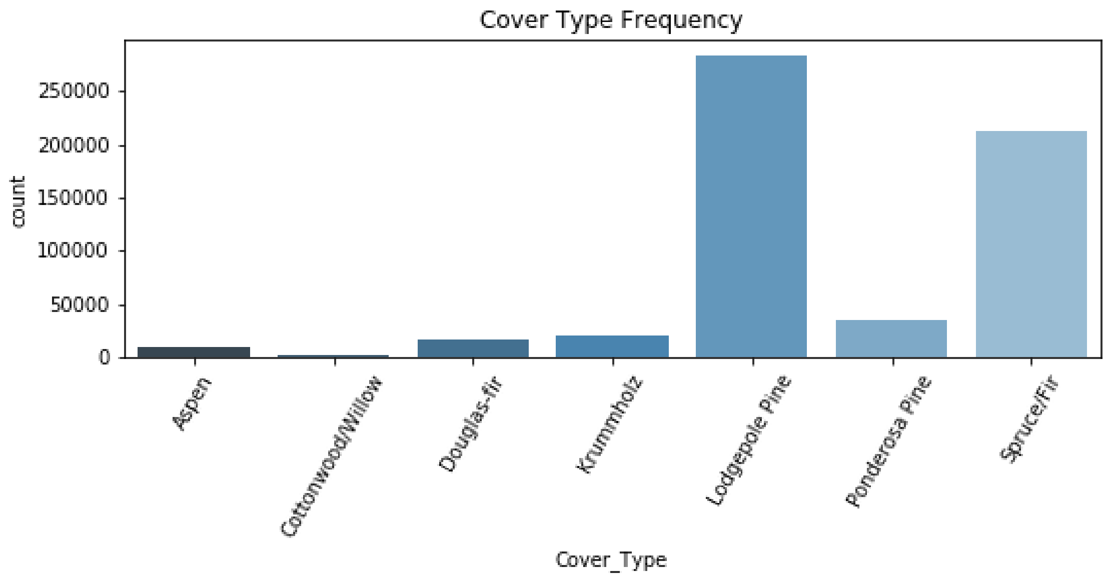
```
In [20]: %matplotlib inline
         plt.figure(figsize=(5,3))
         ax = sns.countplot(x="Geologic_Zone",data=df, palette="Blues_d")
         ax.set_title('Geologic Zone Frequency')
         plt.xticks(rotation=60)

Out[20]: (array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)
```



The graph shows the most of the trees in these wilderness areas are Lodgepole Pine, and Spruce/Fir. This probably means that the Rawah and Comanche Peak areas predominantly have these types. It would be interesting to see whether the other tree types could be predicted with better or worse accuracy than these two.

```
In [21]: %matplotlib inline
         plt.figure(figsize=(9,3))
         ax = sns.countplot(x="Cover_Type",data=df, palette="Blues_d")
         ax.set_title('Cover Type Frequency')
         plt.xticks(rotation=60)
```

```
Out[21]: (array([0, 1, 2, 3, 4, 5, 6]), <a list of 7 Text xticklabel objects>)
```



Section **??** ## Attributes of Interest

Cover Type The cover type is what the other attributes are trying to predict and would be the response for the analysis, but it would be interesting to look at an overall numerical breakdown of tree types. We already saw that Lodgepine and Spruce/Fir occured more frequently, but a percentage breakdown would complete the picture. It would also be interesting to see the most frequent tree types in each wilderness area, climatic zone, and at what elevation.

```
In [10]: pd.crosstab(df.Cover_Type, df.Wilderness_Area, margins=True, margins_name="Total")
```

Out[10]:

Wilderness_Area Cover_Type	Cache_la_Poudre	Comanche_Peak	Neota	Rawah	Total
Aspen	0	5712	0	3781	9493
Cottonwood/Willow	2747	0	0	0	2747
Douglas-fir	9741	7626	0	0	17367
Krummholz	0	13105	2304	5101	20510
Lodgepole Pine	3026	125093	8985	146197	283301
Ponderosa Pine	21454	14300	0	0	35754
Spruce/Fir	0	87528	18595	105717	211840
Total	36968	253364	29884	260796	581012

This a breakdown of the trees in the various wilderness areas by count. Noticeable immediately are the zeros. The cottonwood/willow cover type only occurs in Cache la Poudre. The douglas fir does not occur in Neota and Rawah. This is curious because the biggest areas are Rawah and Comanche Peak, so there must be some other condition difference between Rawah and Comanch Peak that lets the tree only grow in one and not the other. The same circumstance is seen with the Ponderosa Pine. The spruce fir which is the second most populous tree on the list does not occur in Cache la Poudre. Cache la Poudre is looking a little exclusive regarding cover types because neither Krummholz nor Aspen grow there. The numbers here also confirm what the graph showed which is that there are much more Lodgepole Pines and Spruce/Fir cover types.

```
In [15]: pd.crosstab(df.Cover_Type, df.Wilderness_Area, margins=True, margins_name="Total", norm
```

```
Out[15]: Wilderness_Area    Cache_la_Poudre  Comanche_Peak     Neota     Rawah
         Cover_Type
         Aspen                     0.000000       0.601707  0.000000  0.398293
         Cottonwood/Willow         1.000000       0.000000  0.000000  0.000000
         Douglas-fir               0.560891       0.439109  0.000000  0.000000
         Krummholz                 0.000000       0.638957  0.112335  0.248708
         Lodgepole Pine            0.010681       0.441555  0.031715  0.516048
         Ponderosa Pine            0.600045       0.399955  0.000000  0.000000
         Spruce/Fir                0.000000       0.413180  0.087779  0.499042
         Total                     0.063627       0.436074  0.051434  0.448865
```

Normalizing the tree type numbers gives the percentage breakdown of tree type per wilderness area by the total of each individual tree type. All the rows should add up to 100%. The majority of the total number of trees are in the Rawah and Comanche Peak area, but these are the biggest areas. The Douglas Fir and Ponderosa Pine are about evenly split between Cache la Poudre and Comanche Peak. The biggest occurrence is with Krummholz where 64% of its trees are in Comanche Peak. The smallest occurrence is with the Lodgepole Pine. Only 1% of it's trees are found in Cache la Poudre.

```
In [16]: pd.crosstab(df.Cover_Type, df.Wilderness_Area, margins=True, margins_name="Total", norm
```

```
Out[16]: Wilderness_Area    Cache_la_Poudre  Comanche_Peak     Neota     Rawah  \
         Cover_Type
         Aspen                     0.000000       0.022545  0.000000  0.014498
         Cottonwood/Willow         0.074308       0.000000  0.000000  0.000000
         Douglas-fir               0.263498       0.030099  0.000000  0.000000
         Krummholz                 0.000000       0.051724  0.077098  0.019559
         Lodgepole Pine            0.081855       0.493728  0.300663  0.560580
         Ponderosa Pine            0.580340       0.056441  0.000000  0.000000
         Spruce/Fir                0.000000       0.345463  0.622239  0.405363

         Wilderness_Area       Total
         Cover_Type
         Aspen              0.016339
         Cottonwood/Willow  0.004728
         Douglas-fir        0.029891
         Krummholz          0.035300
         Lodgepole Pine     0.487599
         Ponderosa Pine     0.061537
         Spruce/Fir         0.364605
```

The next table is similar except it normalizes the tree types according to the wilderness area numbers so that we can see the breakdown of trees within each area. Each column should add to 100%. Between this table and the table normalized according to row, a better distribution of the trees throughout the wilderness area can be seen. The other important item from this table is that the total column gives the fraction from all the tree types. Therefore, the Lodgepol Pine makes 49% of all the tree types recorded. The Cottnwood/Willow is less than one percent.

Soil Type This categorical variable has forty values, and since the type of soil can affect the plant life, it would be good to get a handle on some of the numbers.

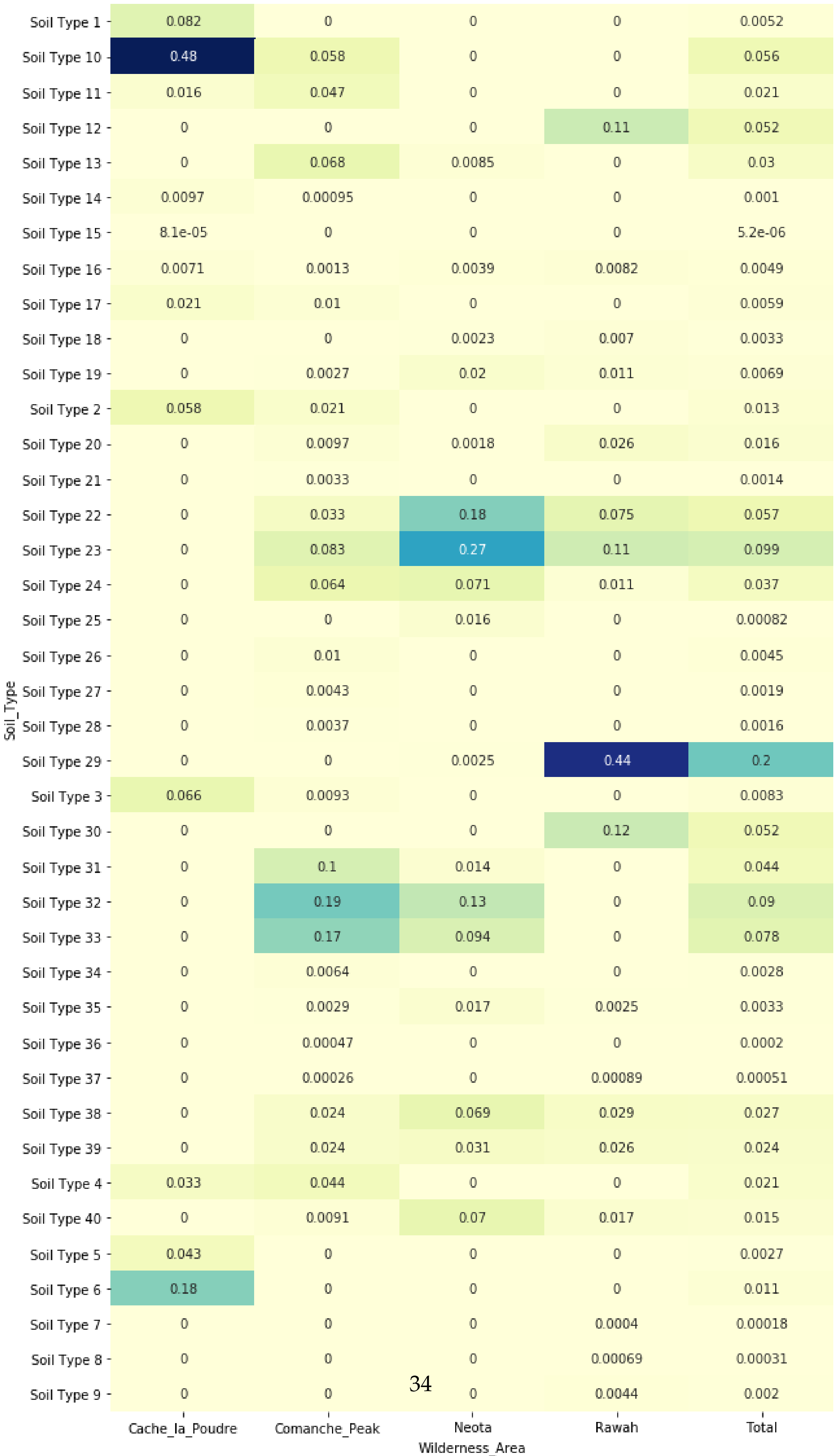
```
In [18]: pd.crosstab(df.Soil_Type, df.Wilderness_Area, margins=True, margins_name="Total")
```

Out[18]:

Wilderness_Area	Cache_la_Poudre	Comanche_Peak	Neota	Rawah	Total
Soil_Type					
Soil Type 1	3031	0	0	0	3031
Soil Type 10	17914	14720	0	0	32634
Soil Type 11	596	11814	0	0	12410
Soil Type 12	0	0	0	29971	29971
Soil Type 13	0	17176	255	0	17431
Soil Type 14	359	240	0	0	599
Soil Type 15	3	0	0	0	3
Soil Type 16	263	325	117	2140	2845
Soil Type 17	793	2629	0	0	3422
Soil Type 18	0	0	70	1829	1899
Soil Type 19	0	675	597	2749	4021
Soil Type 2	2144	5381	0	0	7525
Soil Type 20	0	2452	55	6752	9259
Soil Type 21	0	838	0	0	838
Soil Type 22	0	8362	5363	19648	33373
Soil Type 23	0	21071	8153	28528	57752
Soil Type 24	0	16252	2123	2903	21278
Soil Type 25	0	0	474	0	474
Soil Type 26	0	2589	0	0	2589
Soil Type 27	0	1086	0	0	1086
Soil Type 28	0	946	0	0	946
Soil Type 29	0	0	74	115173	115247
Soil Type 3	2455	2368	0	0	4823
Soil Type 30	0	0	0	30170	30170
Soil Type 31	0	25240	426	0	25666
Soil Type 32	0	48758	3761	0	52519
Soil Type 33	0	42337	2817	0	45154
Soil Type 34	0	1611	0	0	1611
Soil Type 35	0	732	503	656	1891
Soil Type 36	0	119	0	0	119
Soil Type 37	0	66	0	232	298
Soil Type 38	0	5993	2073	7507	15573
Soil Type 39	0	6117	931	6758	13806
Soil Type 4	1238	11158	0	0	12396
Soil Type 40	0	2309	2092	4349	8750
Soil Type 5	1597	0	0	0	1597
Soil Type 6	6575	0	0	0	6575
Soil Type 7	0	0	0	105	105
Soil Type 8	0	0	0	179	179
Soil Type 9	0	0	0	1147	1147
Total	36968	253364	29884	260796	581012

So the big numbers and little numbers are easily determined. Soil type 29 dominates the numbers and soil type 15 only has three occurrences. Soil type 15 is unspecified. The next lowest number is soil type 7 which is the gothic family. Because there are so many values, it makes it a little harder to read so we will make a partial visual plot to help.

```
In [24]: plt.figure(figsize=(10,20))
         sns.heatmap(pd.crosstab(df.Soil_Type, df.Wilderness_Area, normalize="columns", margins=
                     cmap="YlGnBu", annot=True, cbar=False)

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb5dc12c7b8>
```

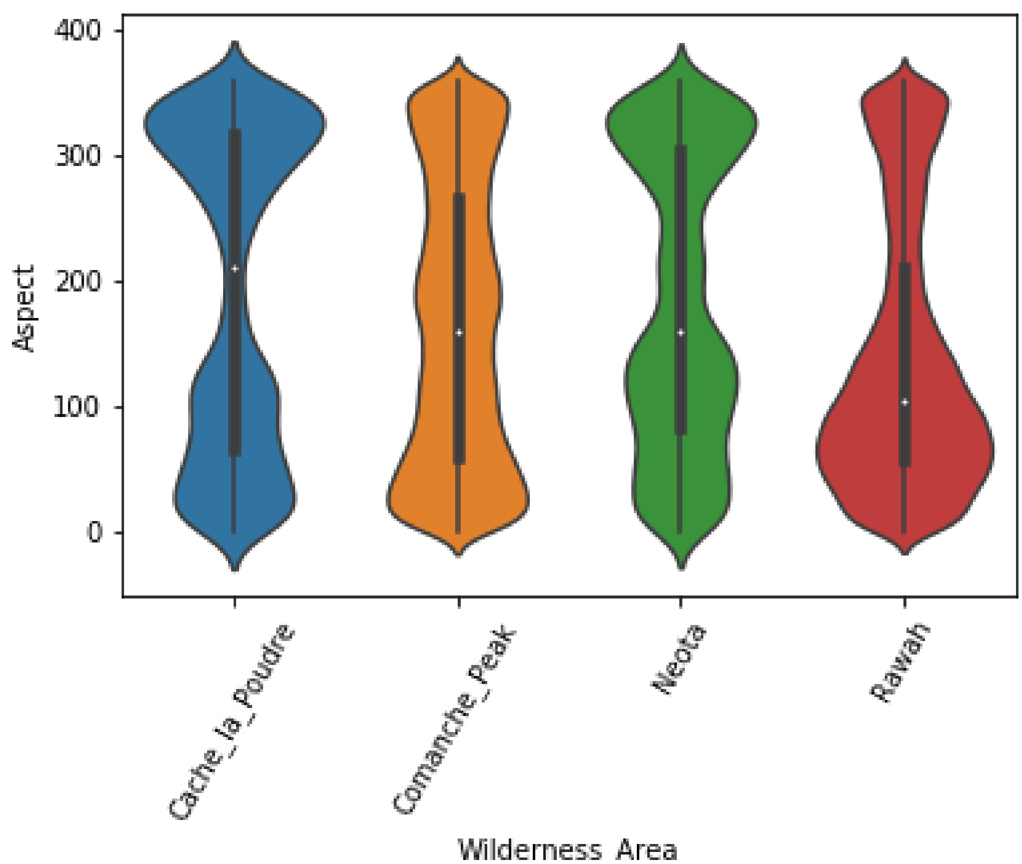

Soil_Type
Wilderness_Area
Cache_la_Poudre
Comanche_Peak
Neota
Rawah
Total
Soil Type 1 0.082 0 0 0 0.0052
Soil Type 10 0.48 0.058 0 0 0.056
Soil Type 11 0.016 0.047 0 0 0.021
Soil Type 12 0 0 0 0.11 0.052
Soil Type 13 0 0.068 0.0085 0 0.03
Soil Type 14 0.0097 0.00095 0 0 0.001
Soil Type 15 8.1e-05 0 0 0 5.2e-06
Soil Type 16 0.0071 0.0013 0.0039 0.0082 0.0049
Soil Type 17 0.021 0.01 0 0 0.0059
Soil Type 18 0 0 0.0023 0.007 0.0033
Soil Type 19 0 0.0027 0.02 0.011 0.0069
Soil Type 2 0.058 0.021 0 0 0.013
Soil Type 20 0 0.0097 0.0018 0.026 0.016
Soil Type 21 0 0.0033 0 0 0.0014
Soil Type 22 0 0.033 0.18 0.075 0.057
Soil Type 23 0 0.083 0.27 0.11 0.099
Soil Type 24 0 0.064 0.071 0.011 0.037
Soil Type 25 0 0 0.016 0 0.00082
Soil Type 26 0 0.01 0 0 0.0045
Soil Type 27 0 0.0043 0 0 0.0019
Soil Type 28 0 0.0037 0 0 0.0016
Soil Type 29 0 0 0.0025 0.44 0.2
Soil Type 3 0.066 0.0093 0 0 0.0083
Soil Type 30 0 0 0 0.12 0.052
Soil Type 31 0 0.1 0.014 0 0.044
Soil Type 32 0 0.19 0.13 0 0.09
Soil Type 33 0 0.17 0.094 0 0.078
Soil Type 34 0 0.0064 0 0 0.0028
Soil Type 35 0 0.0029 0.017 0.0025 0.0033
Soil Type 36 0 0.00047 0 0 0.0002
Soil Type 37 0 0.00026 0 0.00089 0.00051
Soil Type 38 0 0.024 0.069 0.029 0.027
Soil Type 39 0 0.024 0.031 0.026 0.024
Soil Type 4 0.033 0.044 0 0 0.021
Soil Type 40 0 0.0091 0.07 0.017 0.015
Soil Type 5 0.043 0 0 0 0.0027
Soil Type 6 0.18 0 0 0 0.011
Soil Type 7 0 0 0 0.0004 0.00018
Soil Type 8 0 0 0 0.00069 0.00031
Soil Type 9 0 0 0 0.0044 0.002

The darker colors highlight higher numbers with the level of darkness proportional to the magnitude of percentage. This map makes it a little easier to see that soil type 29 is 20% of all the soil types throughout the wilderness areas, but it is 44% of the soil types in the Rawah area. Soil type 10 is 48% of all the soil in Cache la Poudre. We know from the previous cover type analysis is that this area is more exclusive and is largely made up of Ponderosa Pine. Soil type 10 however is only 6% of the total soil types in all the areas. This makes sense that exclusivity of Cache la Poudre in soil type agrees with its exclusivity in tree type.

Aspect Aspect is the azimuth measured in degrees from a reference point, so it is horizontal angular distance to some reference. This is a positional attribute, but what makes it interesting is its variance. The total range of degrees from 0 to 360 is covered in 3 standard deviations. We'll look at violin plots to visualize this.

```
In [31]: sns.violinplot(data = df, x="Wilderness_Area", y="Aspect")
         plt.xticks(rotation=60)
```

```
Out[31]: (array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)
```

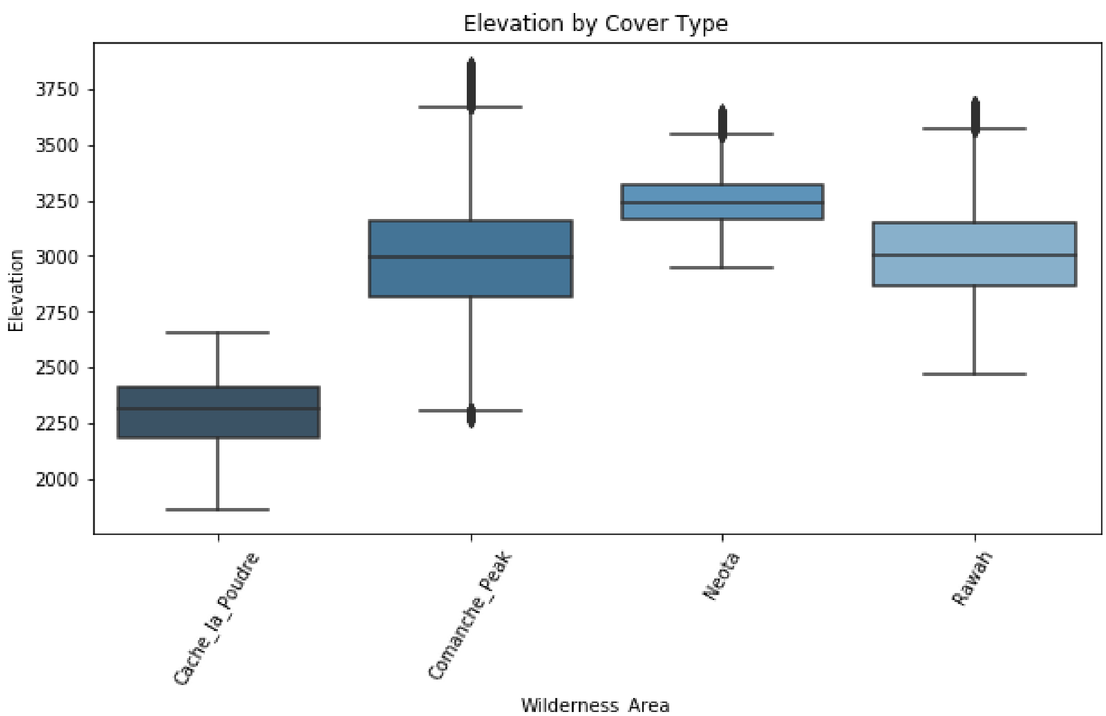


It looks like the distribution is largely bidmodal for all the areas, but if this is an angular distance from a reference line, then interestingly enough, this should represent a positional clustering of areas. This suggests not as many cells across the wilderness areas at 200 degrees from reference, and most around 0 and 360 degrees which are right at the reference line. This probably means that the reference line for this measure is drawn right through the area with the most cells.

Elevation The previous analysis have shown some exclusivity with cover types, soil types, and wilderness area, so it would be interesting to look at the elevation of these areas.

```
In [32]: plt.figure(figsize=(10,5))
         ax = sns.boxplot(x='Wilderness_Area',y='Elevation',data=df, palette = "Blues_d")
         ax.set_title('Elevation by Cover Type')
         plt.xticks(rotation=60)
```

```
Out[32]: (array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)
```



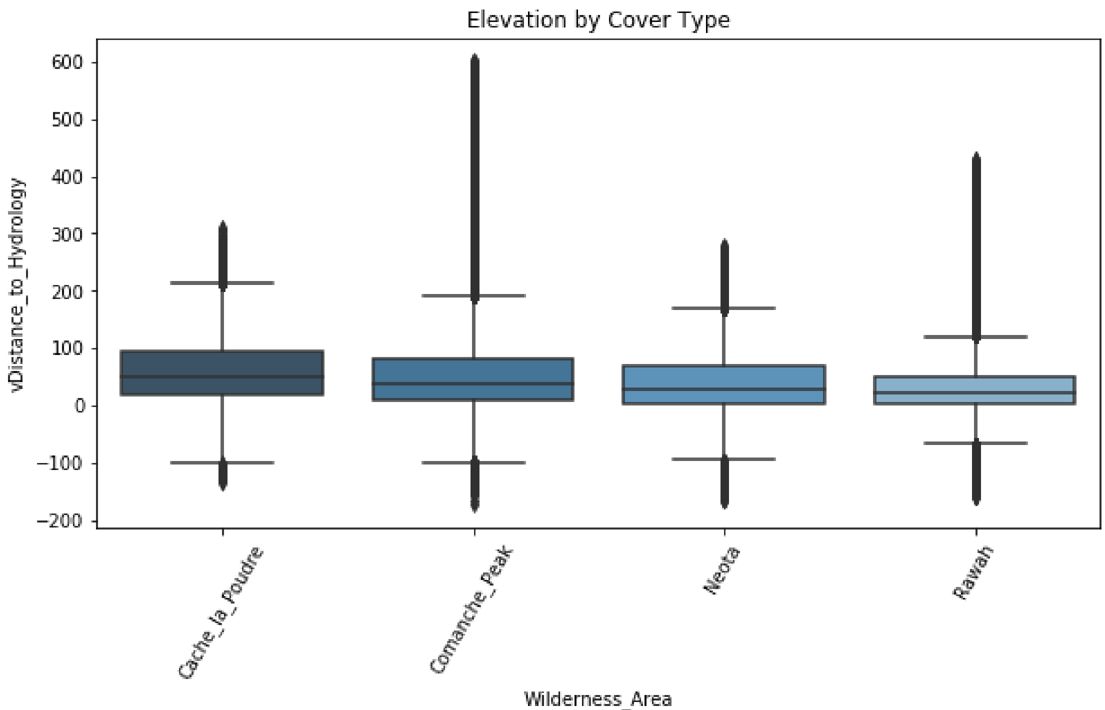
The boxplot shows that the area showing the most exclusivity is also the one with the lowest mean elevation. The two largest areas which have the most number of cover types are at about the same elevation. The neota area has the highest elevation

Vertical distance to Hydrology This attribute had negative values, so it would be interesting to look at boxplots to see what they look like.

```
In [33]: plt.figure(figsize=(10,5))
         ax = sns.boxplot(x='Wilderness_Area',y='vDistance_to_Hydrology',data=df, palette = "Blu
```

```
ax.set_title('Elevation by Cover Type')
plt.xticks(rotation=60)
```

Out[33]: (array([0, 1, 2, 3]), <a list of 4 Text xticklabel objects>)



The area with the lowest elevation, Cache la Poudre, and the least number of trees, also has the highest mean distance to water. The Comanche Peak area has the most outliers probably marking a very diverse terrain. Comanche Peak, Neota, and Rawah have about the same distance to hydrology. This may have something to do with the big body of water that sits between all of them.

Section **??** ## Attribute - Attribute Relationships

In [26]:
```
pd.crosstab(df.Soil_Type, df.Wilderness_Area, margins=True, margins_name="Total")
```

Out[26]:

Wilderness_Area	Cache_la_Poudre	Comanche_Peak	Neota	Rawah	Total
Soil_Type					
Soil Type 1	3031	0	0	0	3031
Soil Type 10	17914	14720	0	0	32634
Soil Type 11	596	11814	0	0	12410
Soil Type 12	0	0	0	29971	29971
Soil Type 13	0	17176	255	0	17431
Soil Type 14	359	240	0	0	599
Soil Type 15	3	0	0	0	3
Soil Type 16	263	325	117	2140	2845
Soil Type 17	793	2629	0	0	3422

```
Soil Type 18                   0           0     70    1829    1899
Soil Type 19                   0         675    597    2749    4021
Soil Type 2                 2144        5381      0       0    7525
Soil Type 20                   0        2452     55    6752    9259
Soil Type 21                   0         838      0       0     838
Soil Type 22                   0        8362   5363   19648   33373
Soil Type 23                   0       21071   8153   28528   57752
Soil Type 24                   0       16252   2123    2903   21278
Soil Type 25                   0           0    474       0     474
Soil Type 26                   0        2589      0       0    2589
Soil Type 27                   0        1086      0       0    1086
Soil Type 28                   0         946      0       0     946
Soil Type 29                   0           0     74  115173  115247
Soil Type 3                 2455        2368      0       0    4823
Soil Type 30                   0           0      0   30170   30170
Soil Type 31                   0       25240    426       0   25666
Soil Type 32                   0       48758   3761       0   52519
Soil Type 33                   0       42337   2817       0   45154
Soil Type 34                   0        1611      0       0    1611
Soil Type 35                   0         732    503     656    1891
Soil Type 36                   0         119      0       0     119
Soil Type 37                   0          66      0     232     298
Soil Type 38                   0        5993   2073    7507   15573
Soil Type 39                   0        6117    931    6758   13806
Soil Type 4                 1238       11158      0       0   12396
Soil Type 40                   0        2309   2092    4349    8750
Soil Type 5                 1597           0      0       0    1597
Soil Type 6                 6575           0      0       0    6575
Soil Type 7                    0           0      0     105     105
Soil Type 8                    0           0      0     179     179
Soil Type 9                    0           0      0    1147    1147
Total                      36968      253364  29884  260796  581012
```

Soil Type 1 and **15** are only present in the **Cache_la_Poudre** wilderness area. **Soil Type 7**, **8**, **9**, **12**, and **30**, are only present in the **Rawah** wilderness area. **Soil Type 14**, **17**, **2**, **3**, and **4**, are only present in the **Cache_la_Poudre** and **Comanche_Peak** wilderness area. Only **Soil Type 16** is present in all wilderness areas.

```
In [27]: pd.crosstab(df.Soil_Type, df.Wilderness_Area, margins=True, margins_name="Total", norma
```

```
Out[27]: Wilderness_Area  Cache_la_Poudre  Comanche_Peak     Neota     Rawah
         Soil_Type
         Soil Type 1             1.000000       0.000000  0.000000  0.000000
         Soil Type 10            0.548937       0.451063  0.000000  0.000000
         Soil Type 11            0.048026       0.951974  0.000000  0.000000
         Soil Type 12            0.000000       0.000000  0.000000  1.000000
         Soil Type 13            0.000000       0.985371  0.014629  0.000000
         Soil Type 14            0.599332       0.400668  0.000000  0.000000
```

Soil Type 15	1.000000	0.000000	0.000000	0.000000
Soil Type 16	0.092443	0.114236	0.041125	0.752197
Soil Type 17	0.231736	0.768264	0.000000	0.000000
Soil Type 18	0.000000	0.000000	0.036862	0.963138
Soil Type 19	0.000000	0.167869	0.148471	0.683661
Soil Type 2	0.284917	0.715083	0.000000	0.000000
Soil Type 20	0.000000	0.264823	0.005940	0.729236
Soil Type 21	0.000000	1.000000	0.000000	0.000000
Soil Type 22	0.000000	0.250562	0.160699	0.588739
Soil Type 23	0.000000	0.364853	0.141173	0.493974
Soil Type 24	0.000000	0.763794	0.099774	0.136432
Soil Type 25	0.000000	0.000000	1.000000	0.000000
Soil Type 26	0.000000	1.000000	0.000000	0.000000
Soil Type 27	0.000000	1.000000	0.000000	0.000000
Soil Type 28	0.000000	1.000000	0.000000	0.000000
Soil Type 29	0.000000	0.000000	0.000642	0.999358
Soil Type 3	0.509019	0.490981	0.000000	0.000000
Soil Type 30	0.000000	0.000000	0.000000	1.000000
Soil Type 31	0.000000	0.983402	0.016598	0.000000
Soil Type 32	0.000000	0.928388	0.071612	0.000000
Soil Type 33	0.000000	0.937614	0.062386	0.000000
Soil Type 34	0.000000	1.000000	0.000000	0.000000
Soil Type 35	0.000000	0.387097	0.265997	0.346906
Soil Type 36	0.000000	1.000000	0.000000	0.000000
Soil Type 37	0.000000	0.221477	0.000000	0.778523
Soil Type 38	0.000000	0.384833	0.133115	0.482052
Soil Type 39	0.000000	0.443068	0.067434	0.489497
Soil Type 4	0.099871	0.900129	0.000000	0.000000
Soil Type 40	0.000000	0.263886	0.239086	0.497029
Soil Type 5	1.000000	0.000000	0.000000	0.000000
Soil Type 6	1.000000	0.000000	0.000000	0.000000
Soil Type 7	0.000000	0.000000	0.000000	1.000000
Soil Type 8	0.000000	0.000000	0.000000	1.000000
Soil Type 9	0.000000	0.000000	0.000000	1.000000
Total	0.063627	0.436074	0.051434	0.448865

In [8]:
```
df_sample = df.sample(frac=0.02, replace=True, random_state=1) #sample with 10% of the d
df_sample.shape
```

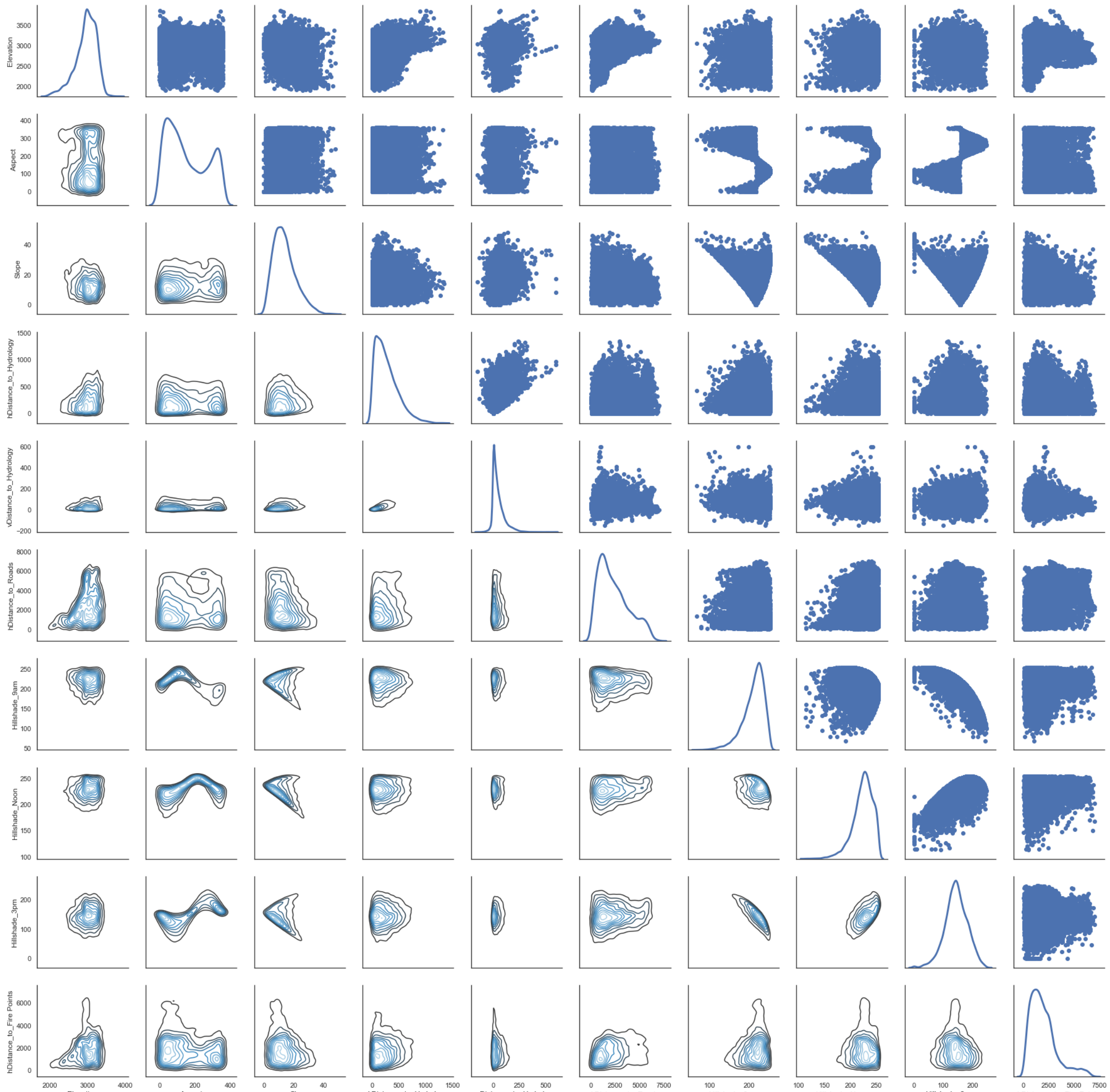
Out[8]: (11620, 15)

In [18]:
```
sns.set(style="white")

g = sns.PairGrid(df_sample[quantitative], diag_sharey=False)
g.map_lower(sns.kdeplot, cmap="Blues_d") # use joint kde on the lower triangle
g.map_upper(plt.scatter) # scatter on the upper
g.map_diag(sns.kdeplot, lw=3) # kde histogram on the diagonal
```

CPU times: user 5 ţs, sys: 0 ns, total: 5 ţs

```
Wall time: 9.06 ţs
```

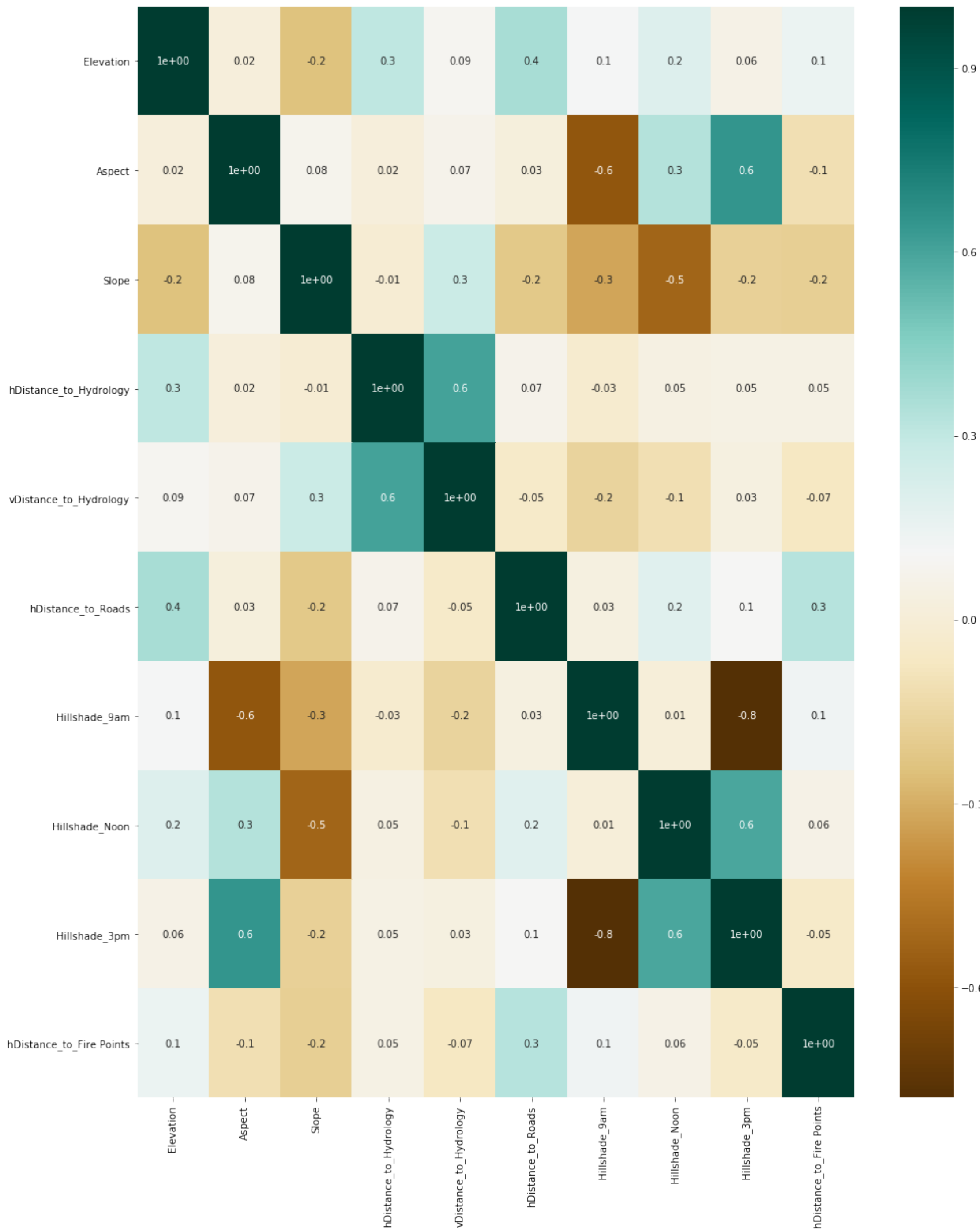
```
Out[18]: <seaborn.axisgrid.PairGrid at 0x7f2f74f89ba8>
```



We can see the bimodal nature of aspect and how it relates to the other continuous variables throug this plot. Also we can quickly see some positive and negative correlation for the hillshade index.

```
In [32]: %matplotlib inline
         cvt = df[quantitative]
         corr = cvt.corr()
         plt.figure(figsize=[16,20])
         sns.heatmap(corr,annot=True, fmt=".1", cmap="BrBG")
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7f880a019c18>



Very quickly we can see how the Hillshade variables are correlated with eachother. **Hillshade_9am** and **Hillshade_3pm** have the highest correlation among all of the continuous variables (-.8). We also see that **Hillshade_3pm** is also highly correlated with **Hillshade_Noon** (.6). Outside of the HIllshade variables we can see that the **Aspect** variable is correlated with **Hillshade_3pm** (.6) and **Hillshade_Noon** (-.6). The variables **Slope** and **Hillshade_Noon** also have a relatively high correlation (-.5). The last set of variables worth mentioning are the distance to hydrology variables. The vertical and horizontal **Distance_to_Hydrology** have a correlation coefficient of .6.

When considering feature selection we try to avoid including variables with high levels of correlation. Based on the correlation analysis we would need to investigate strategies to combine the variables mentioned above to build a better predictive model.

Section **??** ## Attribute - Response Relationship

```
In [33]: df[quantitative + ['Cover_Type']].groupby(by='Cover_Type').mean()
```

```
Out[33]:                      Elevation      Aspect      Slope  hDistance_to_Hydrology  \
         Cover_Type
         Aspen               2787.417571  139.283051  16.641315              212.354893
         Cottonwood/Willow   2223.939934  137.139425  18.528941              106.934838
         Douglas-fir         2419.181897  180.539068  19.048886              159.853458
         Krummholz           3361.928669  153.236226  14.255924              356.994686
         Lodgepole Pine      2920.936061  152.060515  13.550499              279.916442
         Ponderosa Pine      2394.509845  176.372490  20.770208              210.276473
         Spruce/Fir          3128.644888  156.138227  13.127110              270.555245

                             vDistance_to_Hydrology  hDistance_to_Roads  Hillshade_9am  \
         Cover_Type
         Aspen                            50.610344         1349.765722     223.474876
         Cottonwood/Willow                41.186749          914.199490     228.345832
         Douglas-fir                      45.437439         1037.169805     192.844302
         Krummholz                        69.474305         2738.250463     216.967723
         Lodgepole Pine                   45.884219         2429.530799     213.844423
         Ponderosa Pine                   62.446915          943.940734     201.918415
         Spruce/Fir                       42.156939         2614.834517     211.998782

                             Hillshade_Noon  Hillshade_3pm  hDistance_to_Fire Points
         Cover_Type
         Aspen                   219.035816     121.920889               1577.719794
         Cottonwood/Willow       216.997088     111.392792                859.124135
         Douglas-fir             209.827662     148.284044               1055.351471
         Krummholz               221.746026     134.932033               2070.031594
         Lodgepole Pine          225.326596     142.983466               2168.154849
         Ponderosa Pine          215.826537     140.367176                910.955949
         Spruce/Fir              223.430211     143.875038               2009.253517
```

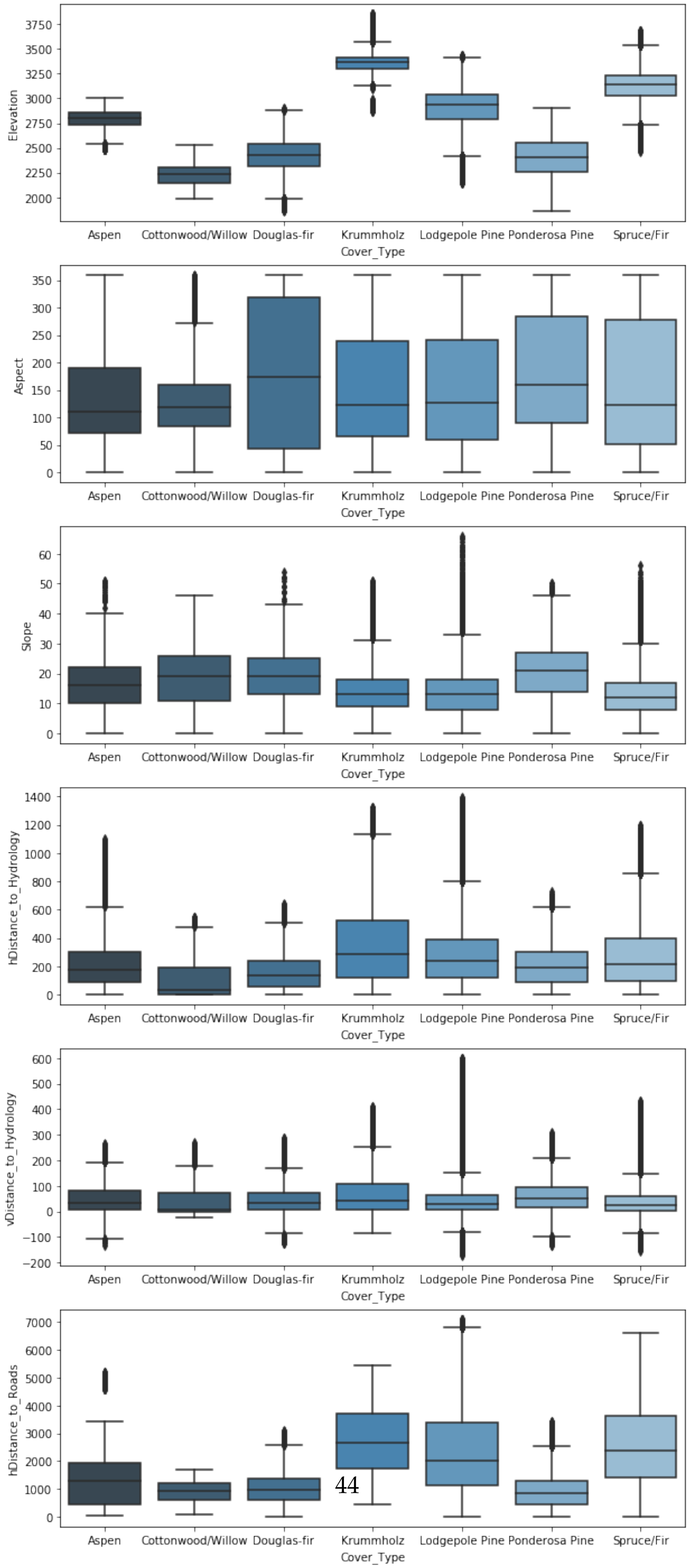
```
In [34]: df[quantitative + ['Cover_Type']].groupby(by='Cover_Type').median()
```

```
Out[34]:                    Elevation  Aspect  Slope  hDistance_to_Hydrology  \
         Cover_Type
         Aspen                   2796     111     16                     175
         Cottonwood/Willow       2231     119     19                      30
         Douglas-fir             2428     173     19                     134
         Krummholz               3363     123     13                     283
         Lodgepole Pine          2935     127     13                     240
```

```
Ponderosa Pine          2404     160     21                      190
Spruce/Fir              3146     122     12                      218

                   vDistance_to_Hydrology  hDistance_to_Roads  Hillshade_9am  \
Cover_Type
Aspen                                  35                1282            228
Cottonwood/Willow                       6                 949            235
Douglas-fir                            34                 966            196
Krummholz                              43                2654            221
Lodgepole Pine                         30                2039            219
Ponderosa Pine                         50                 853            213
Spruce/Fir                             24                2389            216

                   Hillshade_Noon  Hillshade_3pm  hDistance_to_Fire Points
Cover_Type
Aspen                         224            128                      1471
Cottonwood/Willow             220            113                       806
Douglas-fir                   213            150                       942
Krummholz                     224            140                      1969
Lodgepole Pine                227            142                      1846
Ponderosa Pine                221            142                       824
Spruce/Fir                    226            144                      1825
```

In [34]:
```python
fig,axs=plt.subplots(6, figsize=(10,25))
sns.boxplot(x='Cover_Type',y='Elevation',data=df,ax=axs[0], palette = "Blues_d")#highes
sns.boxplot(x='Cover_Type',y='Aspect',data=df,ax=axs[1], palette = "Blues_d")
sns.boxplot(x='Cover_Type',y='Slope',data=df,ax=axs[2], palette = "Blues_d")
sns.boxplot(x='Cover_Type',y='hDistance_to_Hydrology',data=df,ax=axs[3], palette = "Blu
sns.boxplot(x='Cover_Type',y='vDistance_to_Hydrology',data=df,ax=axs[4], palette = "Blu
sns.boxplot(x='Cover_Type',y='hDistance_to_Roads',data=df,ax=axs[5], palette = "Blues_d
```

Out[34]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb5d86dccc0>
```

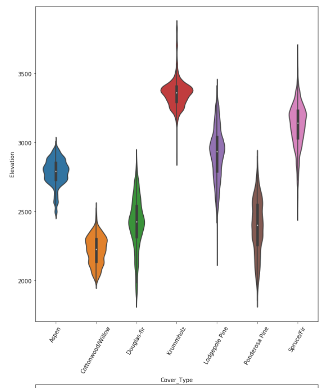

Elevation
3750
3500
3250
3000
2750
2500
2250
2000
Aspect
350
300
250
200
150
100
50
0
Slope
60
50
40
30
20
10
0
hDistance_to_Hydrology
1400
1200
1000
800
600
400
200
0
vDistance_to_Hydrology
600
500
400
300
200
100
0
−100
−200
hDistance_to_Roads
7000
6000
5000
4000
3000
2000
1000
0
Aspen
Cottonwood/Willow
Douglas-fir
Krummholz
Lodgepole Pine
Ponderosa Pine
Spruce/Fir
Cover_Type

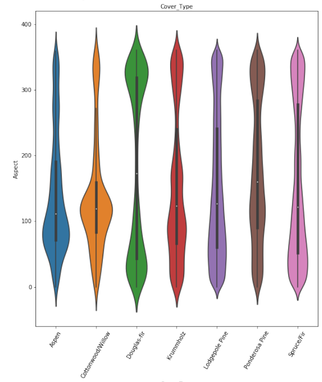
0.1 Analysis of Continuous Variables within each Cover Type

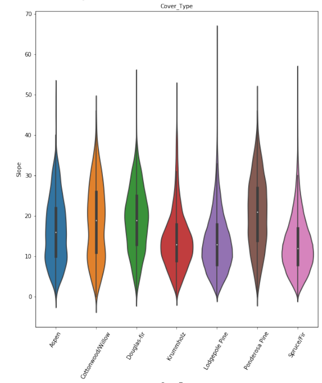
These boxplots help visualize the distribution of our continuous variables within each cover type. Here You'll see that the majority of the variables dont have much of a distinct distribution when comparing between the cover types. The exceptions seem to be the variables **Elevation**, **Slope**, **Horizontal Distance to Hydrology**, **Horizontal Distance to Roadways**, and **Horizontal Distance to Fire Points**. This analysis can give us an idea of which variables to include when building a predictive model.

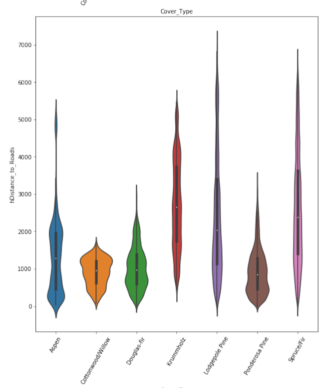
If there was one variable to highlight, it would be elevation. The degree of distinction among between the cover types is quite easy to see from the box plots.

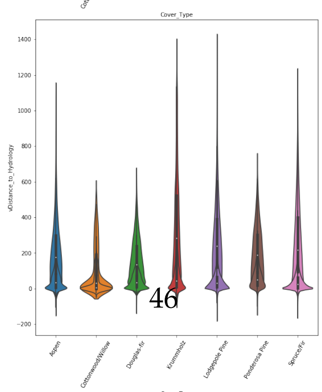
```
In [35]: fig,axs=plt.subplots(5, figsize=(9,60))
         sns.violinplot(x='Cover_Type',y='Elevation',data=df,ax=axs[0])
         sns.violinplot(x='Cover_Type',y='Aspect',data=df,ax=axs[1])
         sns.violinplot(x='Cover_Type',y='Slope',data=df,ax=axs[2])
         sns.violinplot(x='Cover_Type',y='hDistance_to_Hydrology',data=df)
         sns.violinplot(x='Cover_Type',y='vDistance_to_Hydrology',data=df)
         sns.violinplot(x='Cover_Type',y='hDistance_to_Roads',data=df,ax=axs[3])
         for ax in axs:
             plt.sca(ax)
             plt.xticks(rotation=60)
```


Elevation
3500
3000
2500
2000
Aspen
Cottonwood/Willow
Douglas-fir
Krummholz
Lodgepole Pine
Ponderosa Pine
Spruce/Fir
Cover_Type


Aspect
400
300
200
100
0
Aspen
Cottonwood/Willow
Douglas-fir
Krummholz
Lodgepole Pine
Ponderosa Pine
Spruce/Fir
Cover_Type


Slope
70
60
50
40
30
20
10
0
Aspen
Cottonwood/Willow
Douglas-fir
Krummholz
Lodgepole Pine
Ponderosa Pine
Spruce/Fir
Cover_Type

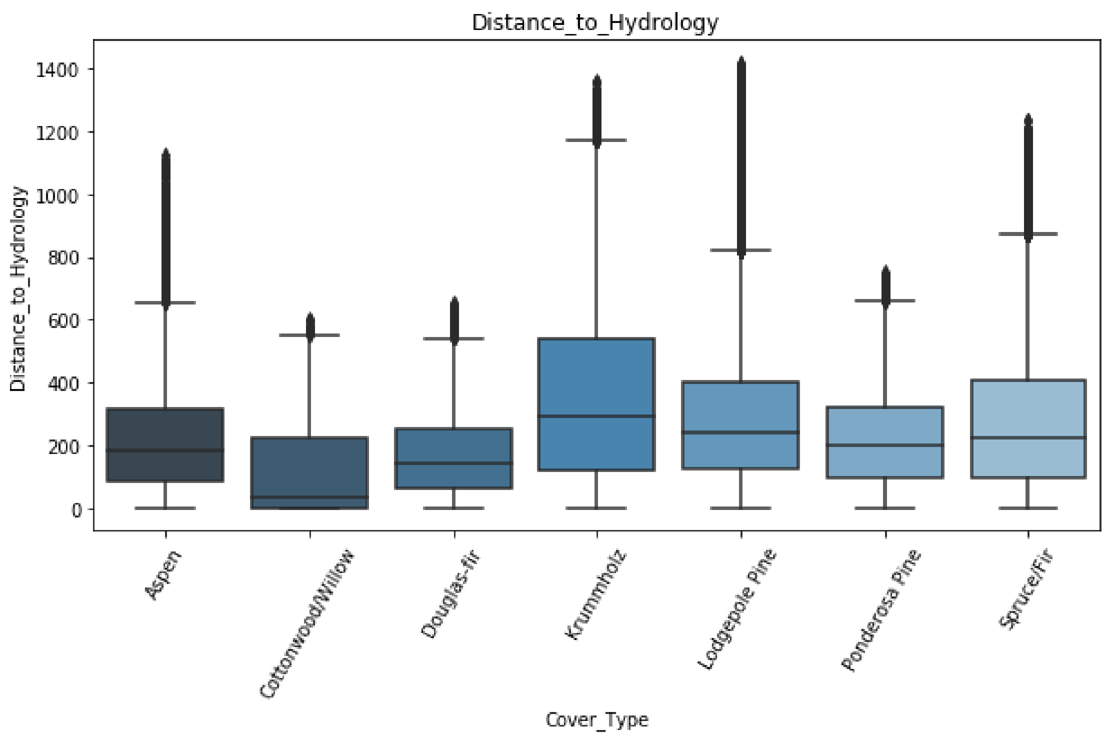

HDistance_to_Roads
7000
6000
5000
4000
3000
2000
1000
0
Aspen
Cottonwood/Willow
Douglas-fir
Krummholz
Lodgepole Pine
Ponderosa Pine
Spruce/Fir
Cover_Type


vDistance_to_Hydrology
1400
1200
1000
800
600
400
200
0
-200
Aspen
Cottonwood/Willow
Douglas-fir
Krummholz
Lodgepole Pine
Ponderosa Pine
Spruce/Fir
Cover_Type

Section **??** ## Additional Features

```
In [39]: df["Distance_to_Hydrology"] = ( (df["hDistance_to_Hydrology"] ** 2) + \
                                          (df["vDistance_to_Hydrology"] ** 2) ) ** (0.5)

         plt.figure(figsize=(10,5))
         ax = sns.boxplot(x='Cover_Type',y='Distance_to_Hydrology',data=df, palette = "Blues_d")
         ax.set_title('Distance_to_Hydrology')
         plt.xticks(rotation=60)
         #sns.swarmplot(x='Cover_Type', y="Distance_To_Hydrology", data=df, color=".25")
```

```
Out[39]: (array([0, 1, 2, 3, 4, 5, 6]), <a list of 7 Text xticklabel objects>)
```



When reviewing our correlation analysis we observed that the **Distance_to_Hydrology** variables were highly correlated. To help reduce the number of correlated variables we can find ways to combine certain data features. Using the Pythagorean Theorem we can determine the straight distance to hydrology and create 1 variable that incorporates two highly correlated variables.

Section **??** #### end