Data Science Essentials

Lab 2 – Working with Summary Statistics

Overview

In this lab, you will learn how to use either R or Python to compute and understand the basics of descriptive statistics. Descriptive statistics aid in understanding a data set. Often your first step after receiving a new data set is to compute, examine and understand simple summary statistics.

What You'll Need

To complete this lab, you will need the following:

- A Web browser
- An Azure Machine Learning workspace.
- The files for this lab

Note: To set up the required environment for the lab, follow the instructions in the Setup Guide for this course.

Computing and Visualizing Summary Statistics with R

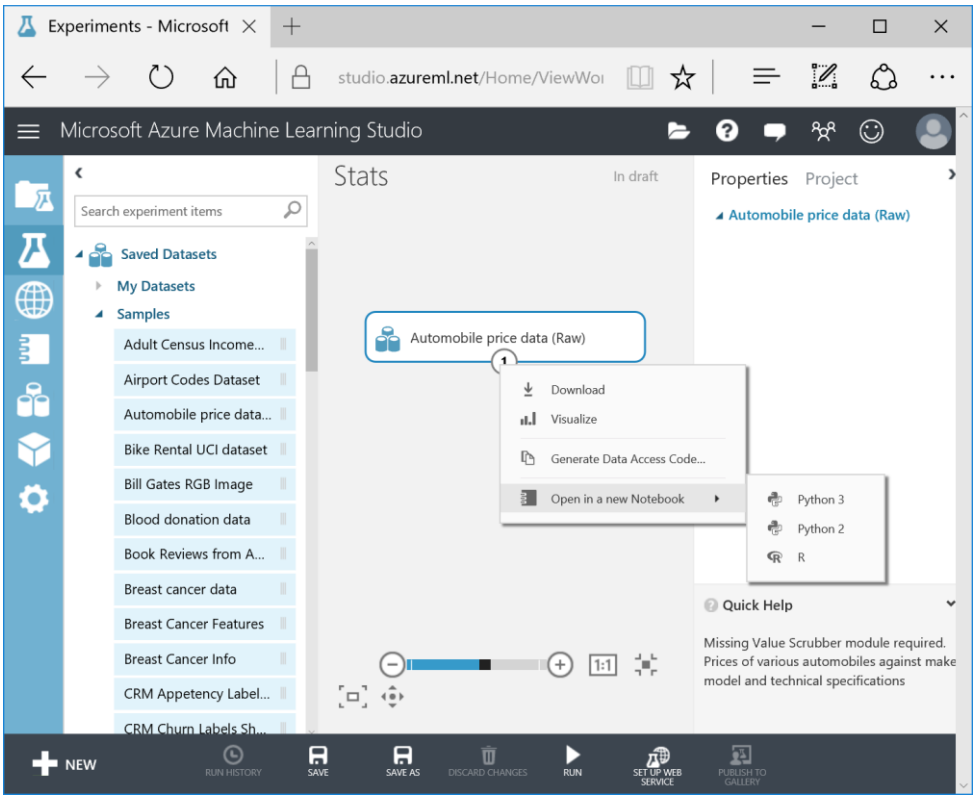
Summary statistics generally include the mean, the median and quartiles of the data. This gives you a first quick look at the distribution of data values. In this exercise, you will compute and interpret the computed summary statistics for the prices of automobiles from the Automotive Price Dataset.

You can copy and paste the R code in this exercise from **Stats.R** in the lab files folder for this module.

Note: If you prefer to work with Python, skip this exercise and complete the next exercise, Computing and Visualizing Summary Statistics with Python.

Create a Jupyter Notebook

1. Browse to https://studio.azureml.net and sign in using the Microsoft account associated with your free Azure ML account.
2. Create a new blank experiment, and name it **Stats**.
3. in the experiment items pane on the left, expand **Saved datasets**, expand **Samples**, and drag the **Automobile price data (Raw)** dataset onto the experiment canvas.
4. Right-click the output of the dataset, and in the menu, point to **Open in a new Notebook**, and then click **R** as shown here:



5. In the new browser tab that opens, note that a Jupyter notebook named **Automobile price data (Raw) notebook** has been created, and that it contains two cells. The first cell contains the following code, which loads the sample dataset into an R data frame named **dat**:

```
library("AzureML")
ws <- workspace()
dat <- download.datasets(ws, "Automobile price data (Raw)")
```

The second cell contains the following code, which displays the first few rows of the data frame:

```
head(dat)
```

Note: Using data frames in R is discussed in a later module of this course.

6. On the **Cell** menu, click **Run All** to run all of the cells in the workbook, and observe the output from the second cell, which shows the first few rows of data from the sample dataset.

Prepare the Data

1. On the **Insert** menu, click **Insert Cell Below** to add a new empty cell to the notebook.
2. In the new cell, enter the following code to clean up the data by converting some string columns to numeric and removing rows with missing values.

```
cols <- c('price', 'bore', 'stroke', 'horsepower', 'peak.rpm')
dat[, cols] <- lapply(dat[, cols], as.numeric)
dat <- dat[complete.cases(dat), ]
str(dat)
```

3. With the cursor still in the third cell, on the **Cell** menu click **Run** to run the selected cell.

4. Review the output, which shows the columns in the source data along with the data type and first few values for each column. Examine the feature names (column names) in this dataset, noting the various properties of the automobiles that might be useful in predicting the price.

View Summary Statistics for Price

1. On the **Insert** menu, click **Insert Cell Below** to add a new empty cell to the notebook.
2. In the new cell, enter the following code to define a function named **describe** that returns summary statistics for a specified column in a data frame.

```
describe <- function(df, col){
  tmp <- df[, col]
  sumry <- summary(tmp)
  nms <- names(sumry)
  nms <- c(nms, 'std')
  out <- c(sumry, sd(tmp))
  names(out) <- nms
  out
}
```

3. With the cursor still in the fourth cell, on the **Cell** menu click **Run** to run the selected cell.
4. Insert another new cell and add the following code to call the **describe** function for the **price** column in the **dat** data frame:

```
describe(dat, 'price')
```

5. Review the output, which shows the *minimum*, *1st quartile*, *median*, *mean*, *3rd quartile*, *maximum*, and *standard deviation* statistics for the **price** column as shown here:

Min	5118
1st Qu.	7756
Median	10240
Mean	13250
3rd Qu.	16510
Max	45400
std	8056.33009328236

6. Note the following properties of these data from the computed summary statistics:
   - The mean is noticeably larger than the median (midpoint or 50% quartile), indicating the distribution is skewed toward large values.
   - The first quartile is closer to the median than the third quartile, adding support to the hypothesis that these distributions are skewed toward large values.
   - The standard deviation is small relative to the range between the minimum and the maximum indicating the distribution has 'long tails'. This is likely to be especially the case toward the larger values.

Visualize Summary Statistics for Price

1. Insert a new cell into the notebook, and add the following code to install the **gridExtra** library and define as function named **plotstats** that plots the statistical data for a specified column:

```
install.packages('gridExtra')
plotstats <- function(df, col, bins = 30){
   require(ggplot2)
   require(gridExtra)
   dat <- as.factor('')
```

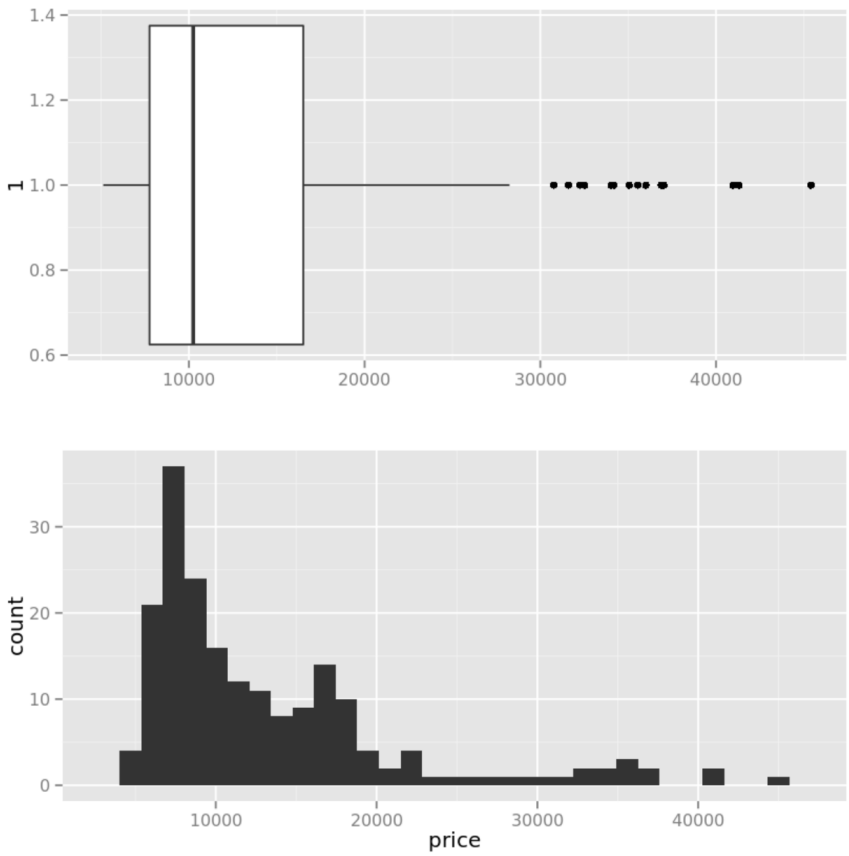
```
    ## Compute bin width
    bin.width <- (max(df[, col]) - min(df[, col]))/ bins

    ## Plot a histogram
    p1 <- ggplot(df, aes_string(col)) +
      geom_histogram(binwidth = bin.width)

    ## A simple boxplot
    p2 <- ggplot(df, aes_string(dat, col)) +
      geom_boxplot() + coord_flip() + ylab('')

    ## Now stack the plots
    grid.arrange(p2, p1, nrow = 2)
}
```

Note: Plotting with R is discussed in a later module of this course.

2. Run the cell, and note that the **gridExtra** library is installed into the current session.
3. Insert a new cell in the notebook, and add the following code; which uses the **plotstats** function to create visualizations of the **price** column in the **dat** data frame.

```
plotstats(dat, 'price')
```

4. Review the output, which should look similar to this:



Note: The box plot is plotted horizontally so it can be compared directly with the histogram plotted below.

5. Examine these plots and note the following observations, which are consistent with the analysis of the summary statistics in the preceding exercise:
    - The dark line in the box of the box plot is at the median value and is to the right of the mode (the highest peak on the histogram).
    - The box of the box plot contains the middle two quartiles of the data. A considerable number of data values can be seen to the right (high value side) of the box and in the right 'tail' of the histogram.
    - The box plot exhibits a considerably longer whisker and dots showing several outliers on the right side.

Compute and Visualize Summary Statistics for horsepower

The autos dataset also includes a column named **horsepower**. In this procedure you will view statistics for this column.

1. Use the **describe** function to compute summary statistics for the **horsepower** column.
2. Use the **plotstats** function to create a box chart and histogram for the **horsepower** column.

Computing and Visualizing Summary Statistics with Python

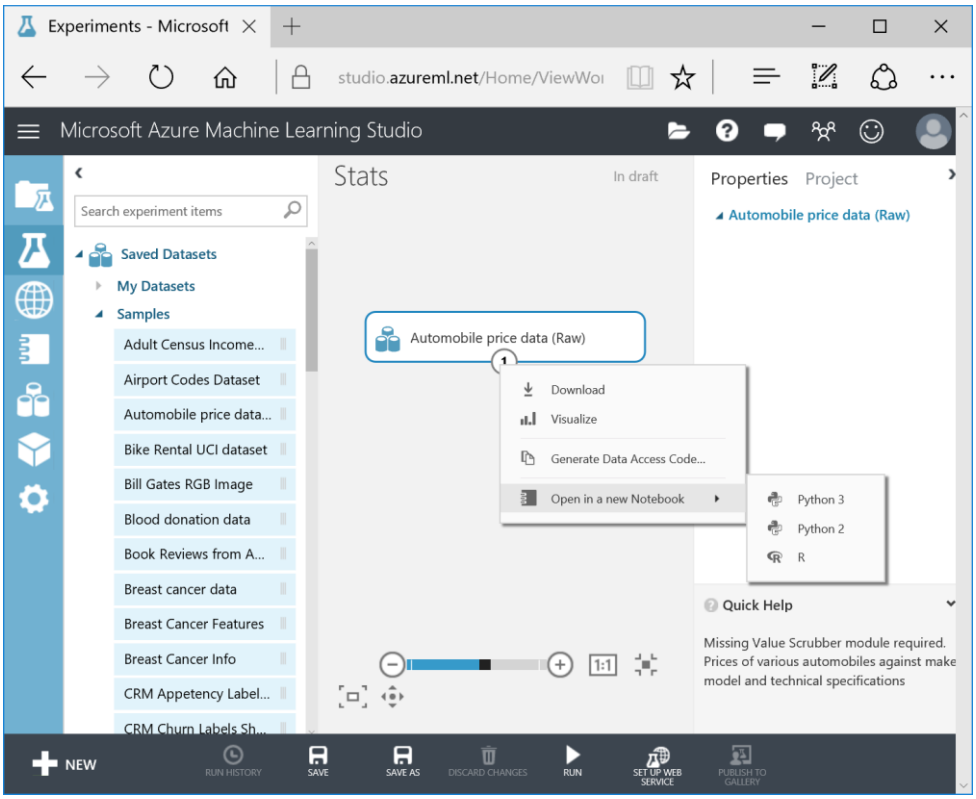
Summary statistics generally include the mean, the median and quartiles of the data. This gives you a first quick look at the distribution of data values. In this exercise, you will compute and interpret the computed summary statistics for the prices of automobiles from the Automotive Price Dataset.

You can copy and paste the Python code in this exercise from **Stats.py** in the lab files folder for this module.

Note: If you prefer to work with R, skip this exercise and complete the previous exercise, *Computing and Visualizing Summary Statistics with R*.

Create a Jupyter Notebook

1. Browse to https://studio.azureml.net and sign in using the Microsoft account associated with your free Azure ML account.
2. Create a new blank experiment, and name it **Stats**.
3. in the experiment items pane on the left, expand **Saved datasets**, expand **Samples**, and drag the **Automobile price data (Raw)** dataset onto the experiment canvas.
4. Right-click the output of the dataset, and in the menu, point to **Open in a new Notebook**, and then click **Python 2** as shown here:



5. In the new browser tab that opens, note that a Jupyter notebook named **Automobile price data (Raw) notebook** has been created, and that it contains two cells. The first cell contains the following code, which loads the sample dataset into a Pandas data frame named **frame**:

```
from azureml import Workspace
ws = Workspace()
ds = ws.datasets['Automobile price data (Raw)']
frame = ds.to_dataframe()
```

The second cell contains the following code, which displays the data frame:

```
frame
```

Note: Using data frames in Python is discussed in a later module of this course.

6. On the **Cell** menu, click **Run All** to run all of the cells in the workbook, and observe the output from the second cell, which shows the first few rows of data from the sample dataset.

Prepare the Data

1. On the **Insert** menu, click **Insert Cell Below** to add a new empty cell to the notebook.
2. In the new cell, enter the following code to clean up the data by converting some string columns to numeric and removing rows with missing values.

```
cols = ['price', 'bore', 'stroke', 'horsepower', 'peak-rpm']
frame[cols] = frame[cols].convert_objects(convert_numeric = True)
frame.dropna(axis = 0, inplace = True)
frame.dtypes
```

3. With the cursor still in the third cell, on the **Cell** menu click **Run** to run the selected cell.

Note: Ignore the warning that is displayed.

4. Review the output, which shows the columns in the source data along with the data type for each column. Examine the feature names (column names) in this dataset, noting the various properties of the automobiles that might be useful in predicting the price.

View Summary Statistics for Price

1. On the **Insert** menu, click **Insert Cell Below** to add a new empty cell to the notebook.
2. In the new cell, enter the following code to define a function named **describe** that returns summary statistics for a specified column in a data frame.

```
def describe(df, col):
    ## Compute the summary stats
    desc = df[col].describe()

    ## Change the name of the 50% index to median
    idx = desc.index.tolist()
    idx[5] = 'median'
    desc.index = idx
    return desc
```

3. With the cursor still in the fourth cell, on the **Cell** menu click **Run** to run the selected cell.
4. Insert another new cell and add the following code to call the **describe** function for the **price** column in the **frame** data frame:

```
describe(frame, 'price')
```

5. Review the output, which shows the statistics for the **price** column as shown here:

```
count       195.000000
mean      13248.015385
std        8056.330093
min        5118.000000
25%        7756.500000
median    10245.000000
75%       16509.000000
max       45400.000000
```

6. Note the following properties of these data from the computed summary statistics:
    - The mean is noticeably larger than the median (midpoint or 50% quartile), indicating the distribution is skewed toward large values.
    - The first quartile is closer to the median than the third quartile, adding support to the hypothesis that these distributions are skewed toward large values.
    - The standard deviation is small relative to the range between the minimum and the maximum indicating the distribution has 'long tails'. This is likely to be especially the case toward the larger values.

Visualize Summary Statistics for Price

1. Insert a new cell into the notebook, and add the following code to define a function named **plotstats** that plots the statistical data for a specified column:

```
def plotstats(df, col):
    import matplotlib.pyplot as plt
    ## Setup for ploting two charts one over the other
    fig, ax = plt.subplots(2, 1, figsize = (12,8))
```

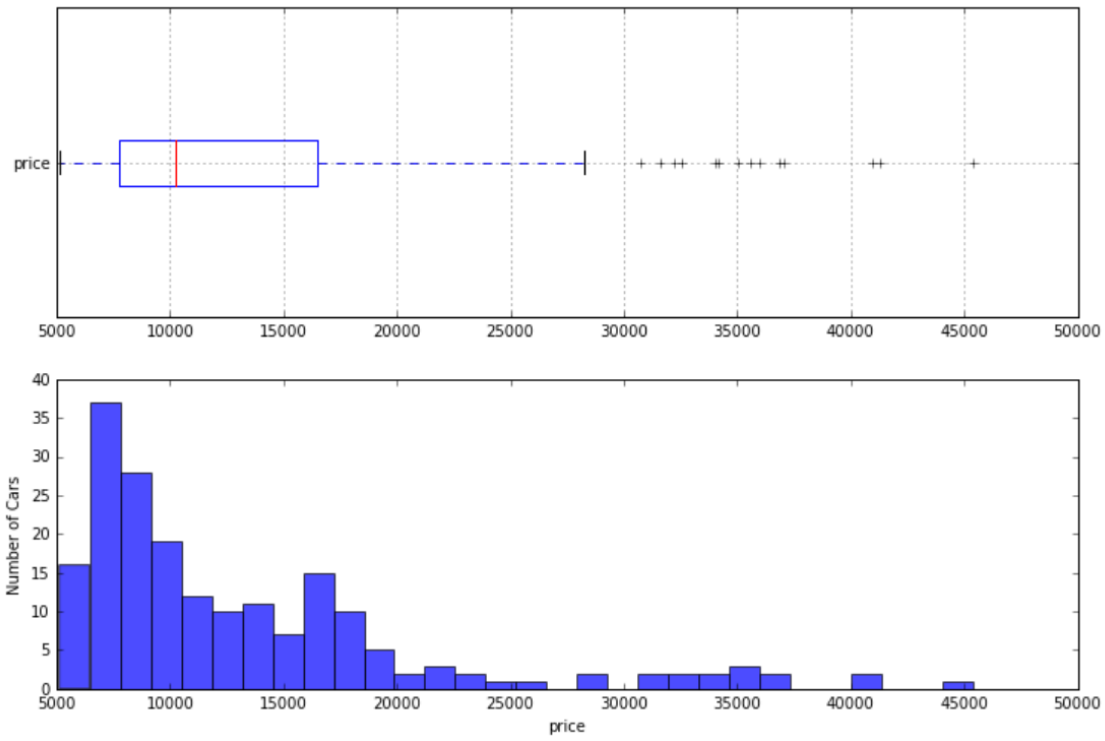
```
## First a box plot
df.dropna().boxplot(col, ax = ax[0], vert=False,
                    return_type='dict')

## Plot the histogram
    temp = df[col].as_matrix()
    ax[1].hist(temp, bins = 30, alpha = 0.7)
    plt.ylabel('Number of Cars')
    plt.xlabel(col)
    return [col]
```

Note: Plotting with Python is discussed in a later module of this course.

2. Run the cell – no output is returned.
3. Insert a new cell in the notebook, and add the following code; which uses the **plotstats** function to create visualizations of the **price** column in the **frame** data frame.

```
plotstats(frame, 'price')
```

4. Review the output, which should look similar to this:



Note: The box plot is plotted horizontally so it can be compared directly with the histogram plotted below.

5. Examine these plots and note the following observations, which are consistent with the analysis of the summary statistics in the preceding exercise:
    - The dark line in the box of the box plot is at the median value and is to the right of the mode (the highest peak on the histogram).
    - The box of the box plot contains the middle two quartiles of the data. A considerable number of data values can be seen to the right (high value side) of the box and in the right 'tail' of the histogram.

- The box plot exhibits a considerably longer whisker and dots showing several outliers on the right side.

Compute and Visualize Summary Statistics for horsepower

The autos dataset also includes a column named **horsepower**. In this procedure you will view statistics for this column.

3. Use the **describe** function to compute summary statistics for the **horsepower** column.
4. Use the **plotstats** function to create a box chart and histogram for the **horsepower** column.

Summary

This lab you have used either R or Python in a Jupyter notebook to compute and examine some descriptive statistics for the automobile price data set. You also created some simple exploratory charts to aid in understanding this dataset. You will apply this simple, yet useful, techniques many times in your career as a data scientist.