Prueba de evaluación 1

Entornos de data science con Python
Curso 2015/16.

Parte 1

En esta primera parte trabajaremos con datos de cotización de Bitcoin. En esta página:

http://www.coindesk.com/price/

Se pueden descargar los valores de “cierre” de esa cotización en dólares en diferentes mercados “exchanges” mediante el botón “Export” del gráfico, como CSV.

1.1. Lectura de datos

- Obtén de coindex los datos de al menos tres exchanges como ficheros CSV separados para al menos un período de un año.
- Carga los datos como un DataFrame fusionando los ficheros en uno solo.
- Utiliza como índice la fecha de cotización.

1.3. Procesamiento

- Obtén los día de máximo y mínimo valor de cotización y un gráfico con la media móvil (“*rolling mean*”) simple (https://en.wikipedia.org/wiki/Moving_average) de los valores de una de las series (puedes probar con varios valores de n). Nota: La media móvil puede encontrarse implementada en alguna biblioteca, no hace falta calcularla programando.
- Obtén el máximo “spread” (diferencia para el mismo día en los diferentes exchanges) del precio de bitcoin, y el “spread medio” para todo el periodo.
- Crea gráficos que los comparen dos a dos, y que muestren con alguna indicación gráfica (por ejemplo, un punto de otro color) los días de mayor “spread”. Pista: puedes obtener un subconjunto del dataframe solo con estos días y dibujarlo como si fuese otra serie diferente.

Parte 2

En esta segunda parte trabajaremos con un dataset de R que describe las características físicas de un conjunto de diamantes. Puede cargarse mediante get_rdataset(), y está en el paquete "ggplot2" con nombre "diamonds".

- Calcula los valores máximo, mínimo y medio de la variable carat.
- Dibuja su histograma y sobre el mismo gráfico, el histograma de una muestra aleatoria del mismo tamaño con puntos obtenidos de una distribución gamma con la misma media y desviación típica. Haz que la visualización sea más adecuada haciendo transparente uno de los histogramas utilizando el parámetro *alpha*.
- Utilizando matplotlib, dibuja la nube de puntos de carat contra el precio, con ambos ejes en escala logarítmica.
- Ahora repite el gráfico pero utiliza solo los datos cuyo color sea "E" y su claridad sea "SI1"
- Investiga con un gráfico la relación entre el volumen (que puedes aproximar a partir de x, y, z) y el precio, de nuevo en escala logarítmica.
- Utilizando scipy.stats, comprueba si se la relación anterior ajusta bien a un modelo de regresión lineal, tomando solo los 1000 primeros valores del dataset.

- Utilizando seaborn, representa la nube de puntos del precio contra el carat y asociando al diagrama las distribuciones de las dos variables.
- Identifica con un comando los diferentes tipos de corte (cut).
- Comprueba si hay diferencias significativas en la media del precio para los de corte "Ideal" y "Premium" y el resto, tomando 1000 valores de cada (nótese que no se puede asumir que las muestras son de una distribución normal). Puedes investigar funciones lógicas vectorizadas en NumPy como logical_or().
- Obtén la mediana del precio por cada valor de la claridad (clarity) y ordena los resultados de manera descendente.

Parte 3

Podemos descargar datos de empresas del dataset Forbes2000 (paquete HSAUR) mediante get_rdataset(). Encontraremos un ranking de empresas categorizadas por sector, y con datos de su país, ventas, beneficios, recursos y valor de mercado.

- Seleccionar del dataframe los nombres de empresa que incluyen "elect" (sin tener en cuenta mayúsculas). Intentamos con esto obtener empresas “del sector de la electricidad” de forma aproximada.
- Utilizando como índice el país y la categoría y ordenándolo (sort_index), encontrar el valor de mercado de las empresas japonesas y estadounidenses.
- Sobre el mismo subconjunto de empresas eléctricas, obtener el país donde el sector de utilities tiene mayor valor de mercado medio en sus empresas.
- Volviendo al dataset original, obtener las empresas españolas.
- Indexar el dataframe anterior por beneficios (profits) y obtener los nombres y rangos de las que tengan un beneficio nulo o negativo. Nota: intenta indexar de manera mixta con números (filas) y etiquetas columnas), hay una operación específica para esto en pandas.