```
//Question 1:

val coinBaseURL = "/Users/domo/Desktop/spark-data/bitcoin-historical-
data/coinbaseUSD_1-min_data_2014-12-01_to_2018-06-27.csv"
val bitStampURL = "/Users/domo/Desktop/spark-data/bitcoin-historical-
data/bitstampUSD_1-min_data_2012-01-01_to_2018-06-27.csv"
val bitFlyerURL = "/Users/domo/Desktop/spark-data/bitcoin-historical-
data/bitflyerJPY_1-min_data_2017-07-04_to_2018-06-27.csv"
val coinCheckURL = "/Users/domo/Desktop/spark-data/bitcoin-historical-
data/coincheckJPY_1-min_data_2014-10-31_to_2018-06-27.csv"

var coinBaseDF = spark.read.format("csv").option("header",
"true").option("inferSchema", "true").load(coinBaseURL)
var bitStampDF = spark.read.format("csv").option("header",
"true").option("inferSchema", "true").load(bitStampURL)
var bitFlyerDF = spark.read.format("csv").option("header",
"true").option("inferSchema", "true").load(bitFlyerURL)
var coinCheckDF = spark.read.format("csv").option("header",
"true").option("inferSchema", "true").load(coinCheckURL)



//Question 2:

coinBaseDF = coinBaseDF.withColumnRenamed("Close", "CoinBase")
bitStampDF = bitStampDF.withColumnRenamed("Close", "BitStamp")
bitFlyerDF = bitFlyerDF.withColumnRenamed("Close", "BitFlyer")
coinCheckDF = coinCheckDF.withColumnRenamed("Close", "CoinCheck")


//Question 3:

//Dropping unnecessary tables from coinBaseDF and bitStampDF join
var newDF = coinBaseDF.join(bitStampDF,
"Timestamp").drop(bitStampDF("Open")).drop(bitStampDF("High")).drop(bi
tStampDF("Low")).drop(bitStampDF("Volume_(BTC)")).drop(bitStampDF("Vol
ume_(Currency)")).drop(bitStampDF("Weighted_Price"))

//Dropping unnecessary tables from newDF and bitFlyerDF join
newDF = newDF.join(bitFlyerDF,
"Timestamp").drop(bitFlyerDF("Open")).drop(bitFlyerDF("High")).drop(bi
tFlyerDF("Low")).drop(bitFlyerDF("Volume_(BTC)")).drop(bitFlyerDF("Vol
ume_(Currency)")).drop(bitFlyerDF("Weighted_Price"))

//Dropping unnecessary tables from newDF and coinCheckDF join
newDF = newDF.join(coinCheckDF,
"Timestamp").drop(coinCheckDF("Open")).drop(coinCheckDF("High")).drop(
coinCheckDF("Low")).drop(coinCheckDF("Volume_(BTC)")).drop(coinCheckDF
("Volume_(Currency)")).drop(coinCheckDF("Weighted_Price"))
```

```
//Question 4:

var q4DF = newDF.select("Timestamp", "CoinBase", "BitStamp",
"BitFlyer", "CoinCheck")
q4DF.show(10)


//Question 5:

import org.apache.spark.sql.functions._

q4DF.select(avg("CoinBase"), avg("BitStamp"), avg("BitFlyer"),
avg("CoinCheck")).show()



//Question 6:

q4DF.select(stddev_samp("CoinBase"), stddev_samp("BitStamp"),
stddev_samp("BitFlyer"), stddev_samp("CoinCheck")).show()



//Question 7

q4DF.select(stddev_samp("CoinBase")/avg("CoinBase") as
"CoefficientOfVariation(CoinBase)", stddev_samp("BitStamp")/
avg("BitStamp") as "CoefficientOfVariation(BitStamp)",
stddev_samp("BitFlyer")/avg("BitFlyer") as
"CoefficientOfVariation(BitFlyer)", stddev_samp("CoinCheck")/
avg("CoinCheck") as "CoefficientOfVariation(CoinCheck)").show()



//Question 8

print("BitStamp exchange has the smallest Coefficient Of Variation
(0.47435811585), which is " +  0.4743*100 +"%")



//Question 9

q4DF.select(countDistinct(floor(col("Timestamp")/86400))as "Distinct
Values").show()
```

```
//Question 10

q4DF.select(approx_count_distinct(floor(col("Timestamp")/86400),
0.1)as "Approx Distinct Values").show()
```