FIT3152 Data analytics. Tutorial 04

Solutions thus far. Please email me any solutions/scripts and I'll paste in.

1. Slide 20 lists 3 models for data analytics: KDD, SEMMA and CRISP-DM. Describe each of them and outline the origin, main similarities and differences between models. You can use these Wikipedia pages as a starting point.

   http://en.wikipedia.org/wiki/Data_mining
   http://en.wikipedia.org/wiki/SEMMA
   http://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining

   xxxxxxx

2. Briefly read: A Taxonomy of Dirty Data. (Springer link will work from University: http://link.springer.com/article/10.1023%2FA%3A1021564703268)

   (a) Simplify the taxonomy by making groups of errors you think are closely related.

   (b) Choose 10 specific error types and give an example of each.

   xxxxxxx

3. Briefly read: Tidy Data http://www.jstatsoft.org/v59/i10/paper and summarize the main principles of tidy data.

   xxxxxxx

4. Using lecture notes and/or Chapter 12 of R for Data Science http://r4ds.had.co.nz/ as a guide, manually transform the table below to put it into tidy form. Write out the first 10 or so lines of the transformed table.

Student	English S1	English S2	Maths S1	Maths S2
Anna	50	-	77	69
Bobby	-	52	-	47
Carl	5	30	-	55
Duy	37	80	18	10
Enid	82	-	96	58
Fey	73	36	63	-
Geoff	95	72	13	90

```
#By Tooba
Students <- c('Anna','Bobby','Carl')
English.S1 <- c(50, '-', 5)
English.S2 <- c('-', 52, 30)
Math.S1 <- c(77,'-','-')
Math.S2 <- c(69,47,55)
Grades <-
as.data.frame(cbind(Students,English.S1,English.S2,Math.S1,Math.S2))
Grades
Grades$English.S2 <- as.character(Grades$English.S2)
Grades$English.S1 <- as.character(Grades$English.S1)
Grades$Math.S1 <- as.character(Grades$Math.S1)
Grades[Grades=='-']<- 0
Grades$English.S1 <- as.numeric(Grades$English.S1)
```

```
Grades$English.S2 <- as.numeric(Grades$English.S2)
Grades$Math.S1 <-  as.numeric(Grades$Math.S1)
Grades$Math.S2 <-as.numeric(as.character(Grades$Math.S2))
Grades
library(reshape2)
library(stringr)
tidyDf <- melt(Grades, id.vars = "Students", measure.vars =
c("English.S1","English.S2","Math.S1","Math.S2"))
colnames(tidyDf ) <- c("Students" ,"UnitName", "Grade")
tidyDf$Semester <- str_sub(tidyDf$UnitName, -2)
tidyDf$UnitName <- str_sub(tidyDf$UnitName, 1,-4)
tidyDf
#Sort dataframe
tidyDf[order(tidyDf$Students),]
tidyDf[order(tidyDf$Grade,decreasing = TRUE),]
tidyDf[order(tidyDf$Students,-tidyDf$Grade),]
```

5 The data file “Dunhumby1-20.csv” is a cut down and modified set of test data from the Kaggle competition to predict when consumers would next visit a Dunnhumby supermarket and how much they would spend. See: http://www.kaggle.com/c/dunnhumbychallenge for more information. The current modified data set contains the customer ID, Date of visit, Date since last visit (visit_delta), and Spend for 20 customers from the test set.

a Using the customer spend data from the beginning of September 2010 to the end of March 2011 investigate whether there is any difference between the amount spent by customers having the most predictable relationship for time between visits *vs* spend, and the least. To do this first calculate the coefficient of determination (r^2) of visit_delta and visit_spend for each customer. Using r^2 as your criterion create two groups of 10 customers: those with the most predictable visit_spend *vs* visit_delta (highest r^2) and those with the lowest. By comparing the average spend of customers in each group determine whether it is possible to see a difference between groups. Is this difference significant? At what level?

```
# From Sachith
rm(list = ls())
library(readr)
library(ggplot2)

DH = read.csv("Dunhumby1-20.csv", header = T)

DH$visit_date = as.Date(DH$visit_date, format = "%d-%m-%y")

# September 2010 to the end of March 2011

DH_sub = DH[as.Date(DH$visit_date ,"%Y-%m-%d") < as.Date("2011-04-01","%Y-
%m-%d"),]
DH_sub = DH_sub[as.Date(DH$visit_date ,"%Y-%m-%d") > as.Date("2010-08-
31","%Y-%m-%d"),]

attach(DH_sub)

DH_sub_corr = as.data.frame(as.table(by(DH_sub, customer_id,  function(df)
cor(df[3],df[4])^2)))
DH_sub_mean = as.data.frame(as.table(by(DH_sub, customer_id,  function(df)
mean(df$visit_spend))))

DH_data = cbind( DH_sub_corr, DH_sub_mean[2])

corr_median = median(DH_sub_corr$Freq)

names(DH_data) = c("customer_id","corr","spend")
```

```
DH_top10 = DH_data[DH_data$corr > corr_median,]
DH_bot10 = DH_data[DH_data$corr < corr_median,]

t.test(DH_top10$spend,DH_bot10$spend, "greater",  conf.level = 0.99)

##########################################################################

# From Sachith with modification by Tooba
rm(list = ls())
library(readr)
library(ggplot2)

DH = read.csv("Dunnhumby1-20.csv", header = T)

# DH$visit_date = as.Date(DH$visit_date, format = "%d-%m-%y")
# # September 2010 to the end of March 2011
# DH_sub = DH[as.Date(DH$visit_date ,"%Y-%m-%d") < as.Date("2011-04-
01","%Y-%m-%d"),]
# DH_sub = DH_sub[as.Date(DH$visit_date ,"%Y-%m-%d") > as.Date("2010-08-
31","%Y-%m-%d"),]

# by Tooba using plyr and lubridate
#install.packages("lubridate")
library(lubridate)
DH$visit_date <- dmy(DH$visit_date)
library(plyr)
DH_sub <- subset( DH, DH$visit_date > as.Date("2010-08-31","%Y-%m-%d") &
DH$visit_date < as.Date("01-04-2011","%d-%m-%Y"))

attach(DH_sub)

DH_sub_corr = as.data.frame(as.table(by(DH_sub, customer_id,  function(df)
cor(df[3],df[4])^2)))
DH_sub_mean = as.data.frame(as.table(by(DH_sub, customer_id,  function(df)
mean(df$visit_spend))))

DH_data = cbind( DH_sub_corr, DH_sub_mean[2])

corr_median = median(DH_sub_corr$Freq)

names(DH_data) = c("customer_id","corr","spend")

DH_top10 = DH_data[DH_data$corr > corr_median,]
DH_bot10 = DH_data[DH_data$corr < corr_median,]

t.test(DH_top10$spend,DH_bot10$spend, "greater",  conf.level = 0.99)

##########################################################################

# From Heshan
# fitting a linear model to obtain R-squared

rm(list = ls())
library(readr)
library(plyr)

#read data
D = read.csv("Dunhumby1-20.csv", header = TRUE)

#create a new data frame from September 2010 to March 2011
DHX = D[as.Date(D$visit_date,"%d-%m-%y") > as.Date("31-08-10","%d-%m-
%y"),]
```

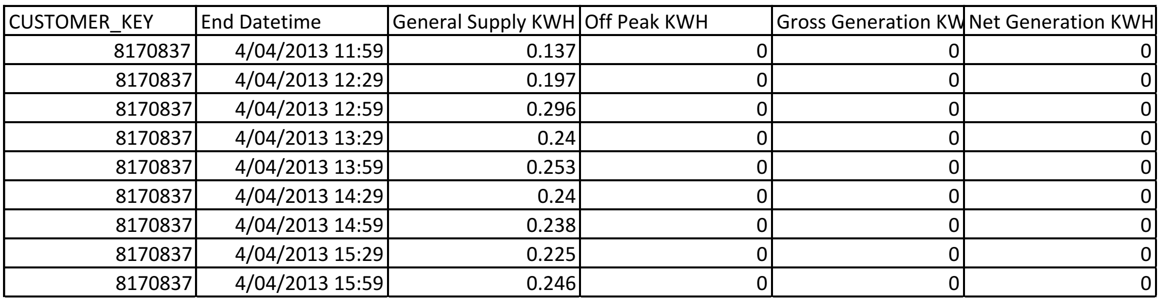
```
DHX = DHX[as.Date(DHX$visit_date,"%d-%m-%y") < as.Date("01-04-11","%d-%m-
%y"),]

#Fit linear models between visit_spend and Visit_delta for each customer
id
Models = by(DHX, DHX$customer_id, function(df)
lm(df$visit_spend~df$visit_delta))

Summary = llply(Models, summary,.print = TRUE) # get list of model
summaries

# get ordered r.squared values from summary list
R = as.data.frame(as.table(sapply(Summary, '[[', 8)))
R_ordered = R[order(R$Freq),]
colnames(R_ordered) = c("customer_id", "r squared")

#group 10 most predictable and least predictable customers
lowest10 =  R_ordered[1:10,]
highest10 = R_ordered[11:20,]

LOW = D[(D$customer_id %in% lowest10$customer_id),]
HIGH = D[(D$customer_id %in% highest10$customer_id),]

#get average spend for low and high groups
LOW_AVG = aggregate(LOW[4], LOW[1], mean)
HIGH_AVG =  aggregate(HIGH[4], HIGH[1], mean)

#check significance of difference in avg spend
print(t.test(HIGH_AVG[2], LOW_AVG[2], "greater", conf.level = 0.99))

############################################################################

# From Heshan with Modification by Abishek
# fitting a linear model to obtain R-squared

rm(list = ls())
library(readr)
library(plyr)

#read data
D = read.csv("Dunnhumby1-20.csv", header = TRUE)

#create a new data frame from September 2010 to March 2011
DHX = D[as.Date(D$visit_date,"%d-%m-%y") > as.Date("31-08-10","%d-%m-
%y"),]
DHX = DHX[as.Date(DHX$visit_date,"%d-%m-%y") < as.Date("01-04-11","%d-%m-
%y"),]

# #Fit linear models between visit_spend and Visit_delta for each customer
id
# Models = by(DHX, DHX$customer_id, function(df)
lm(df$visit_spend~df$visit_delta))
# Summary = llply(Models, summary,.print = TRUE) # get list of model
summaries
# # get ordered r.squared values from summary list
# R = as.data.frame(as.table(sapply(Summary, '[[', 8)))

# Abishek replaced by
R = ddply(DHX, 'customer_id', function(df) summary(lm(visit_spend ~
visit_delta, data = df))$r.squared)

R_ordered = R[order(R$V1),]
colnames(R_ordered) = c("customer_id", "r squared")
```

```
#group 10 most predictable and least predictable customers
lowest10 =  R_ordered[1:10,]
highest10 = R_ordered[11:20,]

LOW = D[(D$customer_id %in% lowest10$customer_id),]
HIGH = D[(D$customer_id %in% highest10$customer_id),]

#get average spend for low and high groups
LOW_AVG = aggregate(LOW[4], LOW[1], mean)
HIGH_AVG =  aggregate(HIGH[4], HIGH[1], mean)

#check significance of difference in avg spend
print(t.test(HIGH_AVG[2], LOW_AVG[2], "greater", conf.level = 0.99))
```

b Over the same time period investigate whether customers who spend the most in total have a greater number of visits to the store than those who spend the least. To do this create two groups of 10: those having the highest spend in total and those with the lowest. You can now compare the number of visits made by each customer in each of the two groups. Hint: you might want to use the "length" function to count the number of visits made by each customer.

```
# From Heshan

#get ordered total spend within period of interest for each customer
TS = aggregate(DHX[4], DHX[1], sum)
TS_ordered = TS[order(TS$visit_spend),]

#group 10 highest spending and least spending customers
lowest10_S =  TS_ordered[1:10,]
highest10_S = TS_ordered[11:20,]

LOW_S = D[(D$customer_id %in% lowest10_S$customer_id),]
HIGH_S = D[(D$customer_id %in% highest10_S$customer_id),]

attach(LOW_S)
visit_count_low = as.data.frame(as.table(by(visit_spend,list(customer_id),
length)))

attach(HIGH_S)
visit_count_high =
as.data.frame(as.table(by(visit_spend,list(customer_id), length)))
```

6 The data file "govhackelectricitytimeofusedataset.csv" has been created from the .txt file originally available as part of the Australian Government's data resources. See link at: https://data.gov.au/dataset/sample-household-electricity-time-of-use-data. The file contains the smart meter records for a number of households recorded at 30 minute intervals over varying periods of time. The first few rows of the csv file are below.



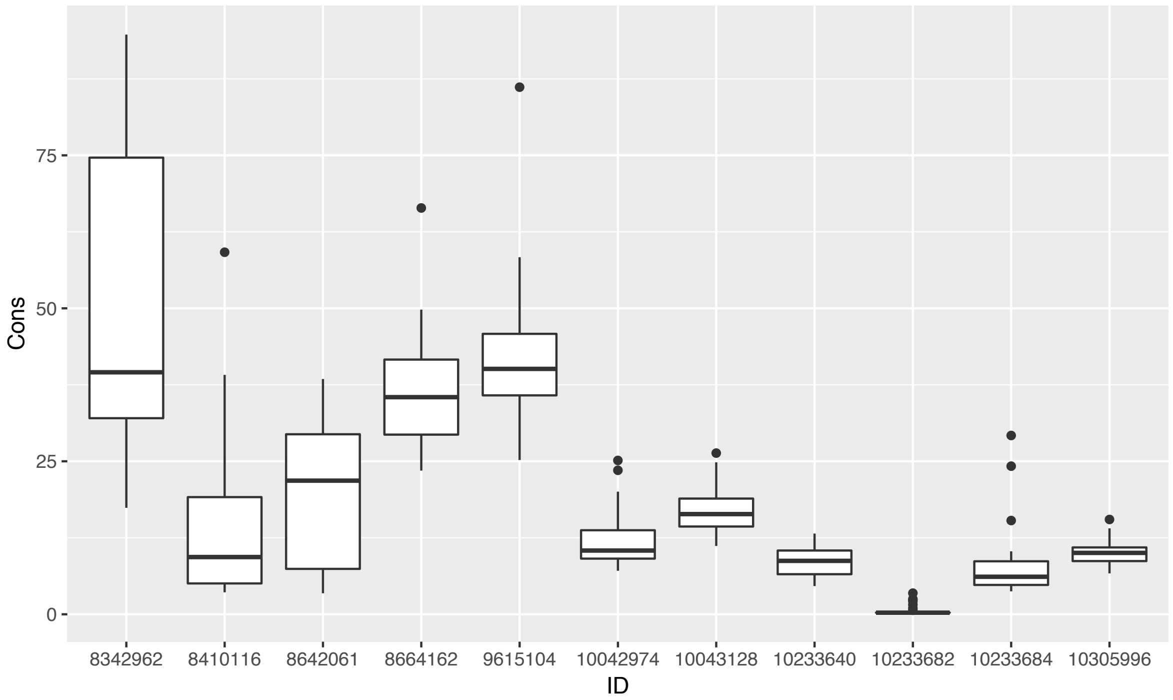
CUSTOMER_KEY	End Datetime	General Supply KWH	Off Peak KWH	Gross Generation KW	Net Generation KWH
8170837	4/04/2013 11:59	0.137	0	0	0
8170837	4/04/2013 12:29	0.197	0	0	0
8170837	4/04/2013 12:59	0.296	0	0	0
8170837	4/04/2013 13:29	0.24	0	0	0
8170837	4/04/2013 13:59	0.253	0	0	0
8170837	4/04/2013 14:29	0.24	0	0	0
8170837	4/04/2013 14:59	0.238	0	0	0
8170837	4/04/2013 15:29	0.225	0	0	0
8170837	4/04/2013 15:59	0.246	0	0	0

The columns of interest are “Customer_Key” (meter), “End Datetime”, and “General SupplyKWH” (power used each 30 mins).

Using the 30 minute general supply, calculate the daily supply for each meter for every day there is data available. Because the number of records is unreliable you will also need to count the number of daily observations for each (day, meter). You should then discard any (day, meter) readings that do not have the complete number of observations.

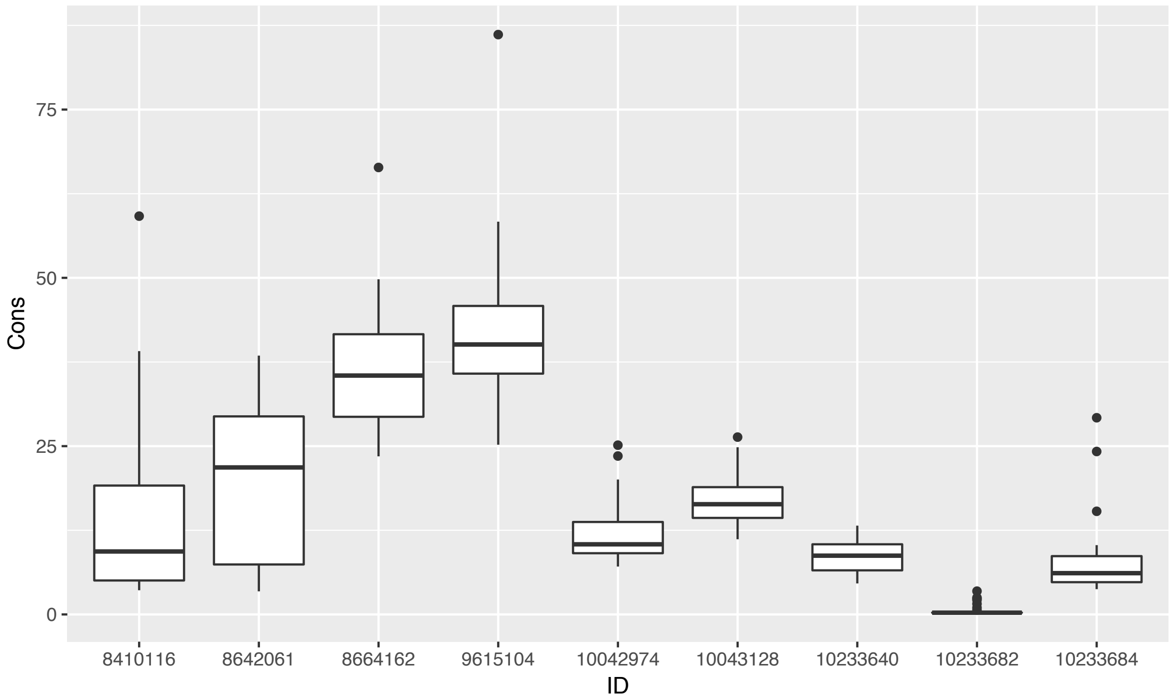
Draw a boxplot of the daily consumption for each meter in January 2013 by meter.

```
rm(list = ls())
library(readr)
library(ggplot2)
GH <- read_csv("govhackelectricitytimeofusedataset.csv")

GH = GH[,1:3]
colnames(GH) = c("ID", "DateTime", "Consumption")
# Extract date, now in YYYY-MM-DD format
GH$Date = as.Date(GH$DateTime,format = "%d/%m/%Y")
GH$DateTime = NULL
GHX = GH[as.Date(GH$Date,"%Y-%m-%d") > as.Date("2012-12-31","%Y-%m-%d"),]
GHX = GHX[as.Date(GHX$Date,"%Y-%m-%d") < as.Date("2013-02-01","%Y-%m-
%d"),]

attach(GHX)

GHDayCons = as.data.frame(as.table(by(Consumption, list(Date, ID), sum)))
GHDayCount = as.data.frame(as.table(by(Consumption, list(Date, ID),
length)))
GHDay = cbind(GHDayCons,GHDayCount)
GHDay = GHDay[,c(1,2,3,6)]
colnames(GHDay) = c("Date", "ID", "Cons", "N")
rm(GHDayCons)
rm(GHDayCount)
rm(GH)
detach(GHX)
rm(GHX)

# remove NA days
GHDay = GHDay[complete.cases(GHDay), ]

# keep only days with 48 observations
GHDay = GHDay[GHDay$N == 48,]

# Plot all data, ignoring incomplete months
g = ggplot(GHDay, aes(ID, Cons)) + geom_boxplot()
g
ggsave("Jan 2013 Cons All.pdf", g, width = 20, height = 12, unit = "cm")
```



Extension, now exclude those meters that do not have a complete set of readings over the month of January (that is 31 days).

```
# to exclude incomplete months..
attach(GHDay)
# count number of days
DayCount = as.data.frame(as.table(by(Cons, ID, length)))

# remove days less than 31
DayCount = DayCount[DayCount$Freq == 31,]

# now keep only those IDs with 31 days
GHDay = GHDay[(GHDay$ID %in% DayCount$ID),]

g = ggplot(GHDay, aes(ID, Cons)) + geom_boxplot()
g
ggsave("Jan 2013 Cons Complete.pdf", g, width = 20, height = 12, unit =
"cm")
```



7. Analyse the Anscombe data set (anscombe). This data set is part of the base R installation and consists of 4 pairs of x,y observations.

   (a) Using summary statistics and correlation describe the main similarities and differences between the pairs.

   (b) Now, using some visual analysis describe the similarities and differences between the pairs.

```
Type ?anscombe into R to get the following code:

require(stats); require(graphics)
summary(anscombe)

##-- now some "magic" to do the 4 regressions in a loop:
ff <- y ~ x
mods <- setNames(as.list(1:4), paste0("lm", 1:4))
for(i in 1:4) {
  ff[2:3] <- lapply(paste0(c("y","x"), i), as.name)
  ## or    ff[[2]] <- as.name(paste0("y", i))
  ##       ff[[3]] <- as.name(paste0("x", i))
  mods[[i]] <- lmi <- lm(ff, data = anscombe)
  print(anova(lmi))
}

## See how close they are (numerically!)
sapply(mods, coef)
lapply(mods, function(fm) coef(summary(fm)))

## Now, do what you should have done in the first place: PLOTS
op <- par(mfrow = c(2, 2), mar = 0.1+c(4,4,1,1), oma =  c(0, 0, 2, 0))
for(i in 1:4) {
  ff[2:3] <- lapply(paste0(c("y","x"), i), as.name)
  plot(ff, data = anscombe, col = "red", pch = 21, bg = "orange", cex =
1.2,
       xlim = c(3, 19), ylim = c(3, 13))
  abline(mods[[i]], col = "blue")
}
mtext("Anscombe's 4 Regression data sets", outer = TRUE, cex = 1.5)
par(op)
```