```
# ********************  Part B ******************************
# ----------------------  Question 1 --------------------------
# --- a ---
marriage.csv <- read.table("marriage.csv", head=TRUE, sep=",")
print("Part B. Question 1. a:")
print(str(marriage.csv))
# --- b ,c ,d ---
marriageWithDateBreakDown <- within(marriage.csv, {
  Year <- as.numeric(substr(TIME_PERIOD , 1, 4))
  Month <- month.abb[as.numeric(substr(TIME_PERIOD , 6, 7))]
})
print("Part B. Question 1. b,c,d:")
print(head(marriage.csv))
# --- e ---
aggregateOverCivicCenter <- with(
    marriageWithDateBreakDown,
    setNames(
        aggregate(MARRIAGE_LICENSES, list(TIME_PERIOD,Month,Year), sum),
        c("TIME_PERIOD", "Month", "Year", "MARRIAGE_LICENSES")))
save(aggregateOverCivicCenter ,file="data.Rda")
print("Part B. Question 1. e:")
print(head(aggregateOverCivicCenter))
# ----------------------  Question 2 --------------------------
# --- a ---
marriageByCivicCentre <- with(marriageWithDateBreakDown, tapply(MARRIAGE_LICENSES ,
CIVIC_CENTRE, sum))
print("Part B. Question 2. a:")
print(head(marriageByCivicCentre))
# --- b ---
print("Part B. Question 2. b:")
barplot(marriageByCivicCentre[attr(marriageByCivicCentre,
"dimnames")[[1]][order(-marriageByCivicCentre)]])
# --- c ---
marriageByYearAndMonth <- with(marriageWithDateBreakDown, tapply(MARRIAGE_LICENSES
, list(Year,Month) , sum))[,month.abb]
print("Part B. Question 2. c:")
print(marriageByYearAndMonth)
# --- d ---
marriageByCivicCenterAndMonth <- with(marriageWithDateBreakDown,
tapply(MARRIAGE_LICENSES , list(CIVIC_CENTRE,Month) , sum))
print("Part B. Question 2. d:")
barplot(marriageByCivicCenterAndMonth[,month.abb])
# ----------------------  Question 3 --------------------------
# --- a ---
businessLicences.csv <- read.csv("businessLicences.csv", head=T, stringsAsFactors =
F)
businessLicencesCleaned <- within(businessLicences.csv, {
  Issued <- as.Date(Issued, format = "%d/%m/%y")
})
print("Part B. Question 3. a:")
print(str(businessLicences.csv))
# --- b ---
businessLicencesWithDateBreakDown <- within(businessLicencesCleaned, {
  Year <- as.numeric(substr(as.character(Issued), 1, 4))
```

```
  Month <- month.abb[as.numeric(substr(as.character(Issued) , 6, 7))]
})
print("Part B. Question 3. b:")
print(str(businessLicencesWithDateBreakDown))
# --- c ---
businessLicencesWithDateOnly <-
businessLicencesWithDateBreakDown[,c('Month','Year')]
marriageWithDateOnly <- marriageWithDateBreakDown[,c('Month','Year')]

businessLicencesWithDataDate <- within(businessLicencesWithDateOnly,{
  YYYYMM <- Year* 100 + match(Month, month.abb)
})

marriageDataDate <- unique(within(marriageWithDateOnly, {
  YYYYMM <- Year* 100 + match(Month, month.abb)
})["YYYYMM"])

businessLicencesWithDataDateLeaned <-
  businessLicencesWithDataDate[businessLicencesWithDataDate$YYYYMM %in%
marriageDataDate$YYYYMM,]
print("Part B. Question 3. c:")
print(head(businessLicencesWithDataDateLeaned))
# --- d ---
businessLicencesWithDataDateLeaned[,"Count"] <-
rep(1,nrow(businessLicencesWithDataDateLeaned))
businessLicencesByYearAndMonth <-
  with(businessLicencesWithDataDateLeaned, tapply(Count , list(Year, Month) ,
sum))[,month.abb]
print("Part B. Question 3. d:")
print(businessLicencesByYearAndMonth)
# **********************  Part C ******************************
# ----------------------  Question 1 --------------------------
combineData <- cbind(c(businessLicencesByYearAndMonth), c(marriageByYearAndMonth))
print("Part C. Question 1")
colnames(combineData) <- c("BusinessLicenses","MarriageLicenses")
print(head(combineData))
# ----------------------  Question 2 --------------------------
print("Part C. Question 2")
plot(combineData)
# ----------------------  Question 3 --------------------------
# --- a ---
lmModel = lm(MarriageLicenses ~ BusinessLicenses , as.data.frame(combineData))
print("Part C. Question 3. a")
print(lmModel)
# --- b ---
# ---------
# --- c ---
print("Part C. Question 3. c")
print(confint(lmModel, level = 0.88))
# --- d ---
print("Part C. Question 3. d")
print(predict(lmModel,  data.frame(BusinessLicenses =c(500)),interval="predict",
level=0.95))
```