W271-2 – Spring 2016 – **Lab 2**

Juanjo Carin, Kevin Davis, Ashley Levato, Minghu Song

March 7, 2016

Contents

Question 6: CLM 3

Your analytics team has been tasked with analyzing aggregate revenue, cost and sales data, which have been provided to you in the R workspace/data frame `retailSales.Rdata`.

Your task is two fold. First, your team is to develop a model for predicting (forecasting) revenues. Part of the model development documentation is a backtesting exercise where you train your model using data from the first two years and evaluate the model's forecasts using the last two years of data.

Second, management is equally interested in understanding variables that might affect revenues in support of management adjustments to operations and revenue forecasts. You are also to identify factors that affect revenues, and discuss how useful management's planned revenue is for forecasting revenues.

Your analysis should address the following:

- **Exploratory Data Analysis: focus on bivariate and multivariate relationships.**
- **Be sure to assess conditions and identify unusual observations.**

First we explore the whole dataset.

```
load("retailSales.Rdata")
data <- retailSales; rm(retailSales)
summary(data)
```

```
##       Year                        Product.line
##  Min.   :2004   Camping Equipment       :24108
##  1st Qu.:2005   Golf Equipment          : 8820
##  Median :2006   Mountaineering Equipment:12348
##  Mean   :2006   Outdoor Protection      : 8820
##  3rd Qu.:2006   Personal Accessories    :30576
##  Max.   :2007
##
##                Product.type                   Product
##  Eyewear             : 9408   Aloe Relief       :  588
##  Watches             : 7644   Astro Pilot       :  588
##  Lanterns            : 7056   Auto Pilot        :  588
##  Cooking Gear        : 5880   Bear Edge         :  588
##  Navigation          : 5880   Bear Survival Edge:  588
##  Climbing Accessories: 4116   Bella             :  588
##  (Other)             :44688   (Other)           :81144
##    Order.method.type  Retailer.country    Revenue
##  E-mail      :12096   Australia: 4032   Min.   :       0
##  Fax         :12096   Austria  : 4032   1st Qu.:   18579
##  Mail        :12096   Belgium  : 4032   Median :   59867
##  Sales visit:12096    Brazil   : 4032   Mean   :  189418
##  Special     :12096   Canada   : 4032   3rd Qu.:  190193
##  Telephone   :12096   China    : 4032   Max.   :10054289
##  Web         :12096   (Other)  :60480   NA's   :59929
##  Planned.revenue     Product.cost        Quantity         Unit.cost
##  Min.   :      16   Min.   :      6   Min.   :     1   Min.   :  0.85
##  1st Qu.:   19557   1st Qu.:   9432   1st Qu.:   328   1st Qu.: 11.43
```

```
##  Median :   63907   Median :  32784   Median :  1043   Median : 36.83
##  Mean   :  198818   Mean   : 111625   Mean   :  3607   Mean   : 84.89
##  3rd Qu.:  203996   3rd Qu.: 111371   3rd Qu.:  3288   3rd Qu.: 80.00
##  Max.   :10054289   Max.   :6756853   Max.   :313628   Max.   :690.00
##  NA's   :59929      NA's   :59929     NA's   :59929    NA's   :59929
##    Unit.price       Gross.profit     Unit.sale.price
##  Min.   :   2.06   Min.   : -18160   Min.   :   0.00
##  1st Qu.:  23.00   1st Qu.:   8333   1st Qu.:  20.15
##  Median :  66.77   Median :  25794   Median :  62.65
##  Mean   : 155.99   Mean   :  77793   Mean   : 147.23
##  3rd Qu.: 148.30   3rd Qu.:  78254   3rd Qu.: 140.96
##  Max.   :1359.72   Max.   :3521098   Max.   :1307.80
##  NA's   :59929     NA's   :59929     NA's   :59929
```

The dataset contains 84,672 observations of 14 variables. 5 of them are categorical (`Product.line`, `Product.type`, `Product`, `Order.method.type`, `Retailer.country`), and `Year` should also be considered as categorical, since there are data from only 4 years (from 2004 to 2007).

```
data <- data %>%
  mutate(Year = as.factor(Year))
```

We also notice (from the output of `summary`) that some of the variables (all of them numerical) has a high number of `NA`s, the same in all cases (59929, i.e., 70.78% of the total number of observations). Do the `NA`s appear in the same observations for all those variables? Yes, they do.

```
# data_isNA <- as.data.frame(sapply(data, is.na))
data_isNA <- data %>% mutate_each(funs(is.na(.)))
head(data_isNA)
```

```
##    Year Product.line Product.type Product Order.method.type
## 1 FALSE        FALSE        FALSE   FALSE             FALSE
## 2 FALSE        FALSE        FALSE   FALSE             FALSE
## 3 FALSE        FALSE        FALSE   FALSE             FALSE
## 4 FALSE        FALSE        FALSE   FALSE             FALSE
## 5 FALSE        FALSE        FALSE   FALSE             FALSE
## 6 FALSE        FALSE        FALSE   FALSE             FALSE
##   Retailer.country Revenue Planned.revenue Product.cost Quantity Unit.cost
## 1            FALSE   FALSE           FALSE        FALSE    FALSE     FALSE
## 2            FALSE   FALSE           FALSE        FALSE    FALSE     FALSE
## 3            FALSE    TRUE            TRUE         TRUE     TRUE      TRUE
## 4            FALSE    TRUE            TRUE         TRUE     TRUE      TRUE
## 5            FALSE   FALSE           FALSE        FALSE    FALSE     FALSE
## 6            FALSE    TRUE            TRUE         TRUE     TRUE      TRUE
##   Unit.price Gross.profit Unit.sale.price
## 1      FALSE        FALSE           FALSE
## 2      FALSE        FALSE           FALSE
## 3       TRUE         TRUE            TRUE
## 4       TRUE         TRUE            TRUE
## 5      FALSE        FALSE           FALSE
## 6       TRUE         TRUE            TRUE
```

```
# vars_with_NAs <- apply(data_isNA, 2, sum)
vars_with_NAs <- data_isNA %>% summarise_each(funs(sum))
(vars_with_NAs <- names(vars_with_NAs)[vars_with_NAs>0])
```

```
## [1] "Revenue"         "Planned.revenue" "Product.cost"    "Quantity"
## [5] "Unit.cost"       "Unit.price"      "Gross.profit"    "Unit.sale.price"
```

```
sapply(data_isNA[, vars_with_NAs[-1]], identical,
       as.vector(data_isNA[, vars_with_NAs[1]]))
```

```
## Planned.revenue    Product.cost        Quantity       Unit.cost
##            TRUE            TRUE            TRUE            TRUE
##      Unit.price    Gross.profit Unit.sale.price
##            TRUE            TRUE            TRUE
```

And the amount of NAs per category is roughly the same for all categorical values (or at least there are non-missing data for all categories; below we just show the percentage per category for three of the numerical variables).

```
data_categorical <- data %>%
  select(which(names(data) %in% names(data)[sapply(data, is.factor)])) %>%
  mutate_each(funs(as.character(.))) %>% mutate(Revenue = data$Revenue)
data_categorical %>%
  select(Revenue, Year) %>%
  group_by(Year) %>%
  summarise_each(funs(100*mean(is.na(.)))) %>%
  rename("% of NAs in numerical variables" = Revenue)
```

```
## Source: local data frame [4 x 2]
##
##    Year % of NAs in numerical variables
##   (chr)                           (dbl)
## 1  2004                        67.95163
## 2  2005                        65.49981
## 3  2006                        71.70257
## 4  2007                        77.95729
```

```
data_categorical %>%
  select(Revenue, Product.line) %>%
  group_by(Product.line) %>%
  summarise_each(funs(100*mean(is.na(.)))) %>%
  rename("% of NAs in numerical variables" = Revenue)
```

```
## Source: local data frame [5 x 2]
##
##               Product.line % of NAs in numerical variables
##                      (chr)                           (dbl)
## 1        Camping Equipment                        65.26049
## 2           Golf Equipment                        68.67347
## 3 Mountaineering Equipment                        76.13379
## 4       Outdoor Protection                        66.62132
## 5     Personal Accessories                        74.77106
```

```
## Source: local data frame [21 x 2]
##
##    Retailer.country % of NAs in numerical variables
##               (chr)                            (dbl)
## 1         Australia                         77.15774
## 2           Austria                         72.44544
## 3           Belgium                         75.99206
## 4            Brazil                         81.49802
## 5            Canada                         57.66369
## 6             China                         77.33135
## 7           Denmark                         80.28274
## 8           Finland                         79.46429
## 9            France                         60.49107
## 10          Germany                         59.37500
## 11            Italy                         69.07242
## 12            Japan                         58.60615
## 13            Korea                         74.47917
## 14           Mexico                         73.36310
## 15      Netherlands                         70.03968
## 16        Singapore                         70.70933
## 17            Spain                         71.55258
## 18           Sweden                         74.25595
## 19      Switzerland                         80.03472
## 20   United Kingdom                         70.23810
## 21    United States                         52.28175
```

So we can ommit all those missing observations (reducing our sample size to 24743), and continue with a further analysis of the numerical variables:

```
data <- data %>% na.omit()
data_categorical <- data %>%
  select(which(names(data) %in% names(data)[sapply(data, is.factor)]))
data_non_categorical <- data %>%
  select(which(names(data) %in% names(data)[!sapply(data, is.factor)]))
round(stat.desc(data_non_categorical, desc = TRUE, basic = TRUE), 2)
```

```
##                   Revenue Planned.revenue Product.cost     Quantity
## nbr.val      2.474300e+04    2.474300e+04 2.474300e+04     24743.00
## nbr.null     7.600000e+01    0.000000e+00 0.000000e+00         0.00
## nbr.na       0.000000e+00    0.000000e+00 0.000000e+00         0.00
## min          0.000000e+00    1.569000e+01 5.760000e+00         1.00
## max          1.005429e+07    1.005429e+07 6.756853e+06    313628.00
## range        1.005429e+07    1.005427e+07 6.756847e+06    313627.00
## sum          4.686776e+09    4.919342e+09 2.761941e+09 89237091.00
## median       5.986727e+04    6.390684e+04 3.278372e+04      1043.00
## mean         1.894182e+05    1.988175e+05 1.116251e+05      3606.56
## SE.mean      2.484130e+03    2.559050e+03 1.515680e+03        55.80
## CI.mean.0.95 4.869040e+03    5.015880e+03 2.970830e+03       109.38
## var          1.526863e+11    1.620349e+11 5.684198e+10 77048387.56
## std.dev      3.907509e+05    4.025355e+05 2.384156e+05      8777.72
## coef.var     2.060000e+00    2.020000e+00 2.140000e+00         2.43
##                 Unit.cost Unit.price  Gross.profit Unit.sale.price
## nbr.val          24743.00   24743.00  2.474300e+04        24743.00
```

```
## nbr.null           0.00       0.00  0.000000e+00          76.00
## nbr.na             0.00       0.00  0.000000e+00           0.00
## min                0.85       2.06 -1.815960e+04           0.00
## max              690.00    1359.72  3.521098e+06        1307.80
## range            689.15    1357.66  3.539257e+06        1307.80
## sum          2100344.99 3859701.42  1.924835e+09     3642909.71
## median            36.83      66.77  2.579376e+04          62.65
## mean              84.89     155.99  7.779311e+04         147.23
## SE.mean            0.83       1.57  1.005230e+03           1.48
## CI.mean.0.95       1.63       3.08  1.970320e+03           2.89
## var            17190.71   60912.60  2.500267e+10       53846.65
## std.dev          131.11     246.80  1.581223e+05         232.05
## coef.var           1.54       1.58  2.030000e+00           1.58
```

All numerical variables are right-skewed, with long right tails (i.e., with several observations more than 2 standard deviations far from the mean), especially the ones corresponding to aggregate—non-unit—results.

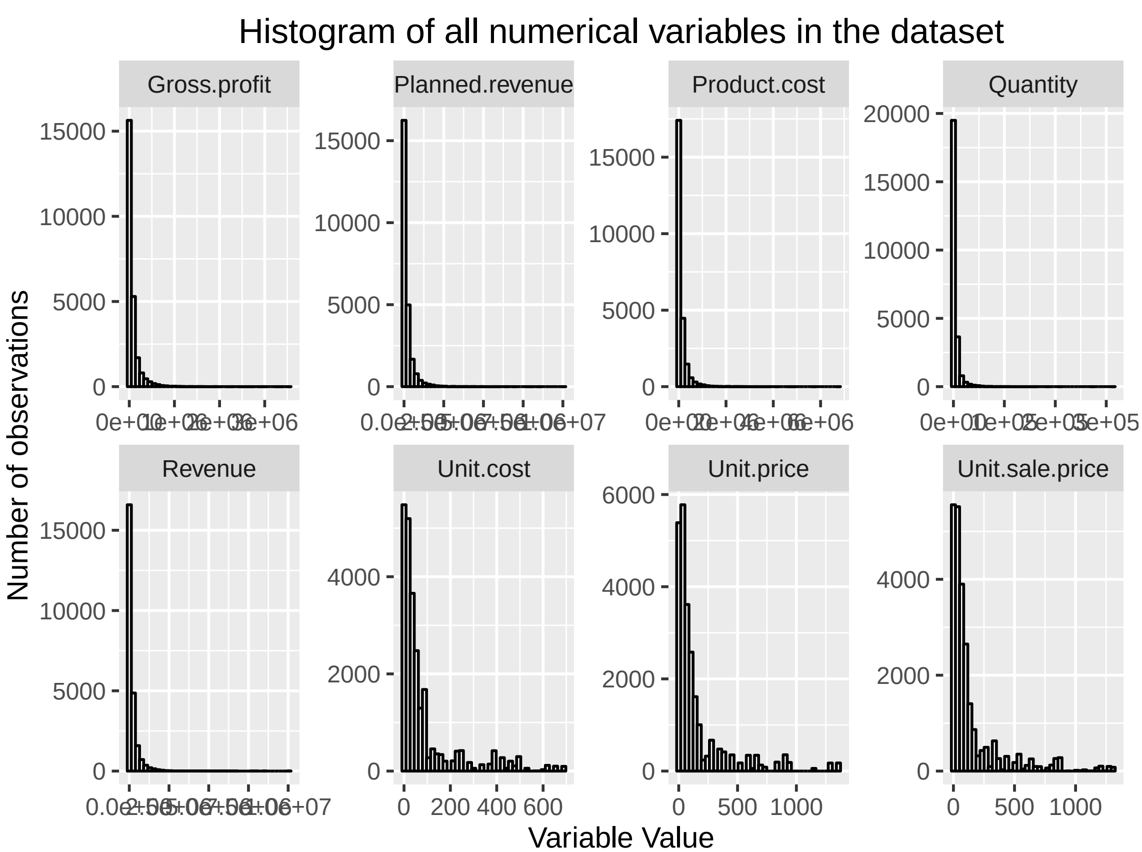

Figure 1: Histogram of all non-categorical variables in the dataset

Below we show the correlation matrix of the numerical variables, as well as two different representations of the scatterplot matrix (where we've used a sample of the data of size 500 because the plotting functions consume a lot of resources; that's why the correlations shown in the second Figure, only approximate, differ from the ones shown right below). As we might have expected, the correlations between `Revenue`, `Planned.revenue`, `Product.cost`, and `Gross.profit` (i.e., the aggregate values), as well as those between `Unit.cost`, `Unit.price`, and `Unit.sale.price` (i.e., the values per unit), are positive and very high. `Quantity` is negatively correlated with the unitary variables (but that correlation is negligible in absolute value), and is moderately correlated ($\rho \simeq 0.5$) with the aggregate values.

```
cor(data_non_categorical)
```

```
##                   Revenue Planned.revenue Product.cost   Quantity
## Revenue         1.0000000       0.9990586    0.9903575  0.5055979
## Planned.revenue 0.9990586       1.0000000    0.9895792  0.4994770
## Product.cost    0.9903575       0.9895792    1.0000000  0.5061298
## Quantity        0.5055979       0.4994770    0.5061298  1.0000000
## Unit.cost       0.2463441       0.2550054    0.2415089 -0.1687497
## Unit.price      0.2332806       0.2421026    0.2194407 -0.1677662
## Gross.profit    0.9779407       0.9767878    0.9395732  0.4862920
## Unit.sale.price 0.2360448       0.2444078    0.2220105 -0.1674531
##                  Unit.cost Unit.price Gross.profit Unit.sale.price
## Revenue          0.2463441  0.2332806    0.9779407       0.2360448
## Planned.revenue  0.2550054  0.2421026    0.9767878       0.2444078
## Product.cost     0.2415089  0.2194407    0.9395732       0.2220105
## Quantity        -0.1687497 -0.1677662    0.4862920      -0.1674531
## Unit.cost        1.0000000  0.9886870    0.2446187       0.9889263
## Unit.price       0.9886870  1.0000000    0.2456107       0.9992750
## Gross.profit     0.2446187  0.2456107    1.0000000       0.2485667
## Unit.sale.price  0.9889263  0.9992750    0.2485667       1.0000000
```

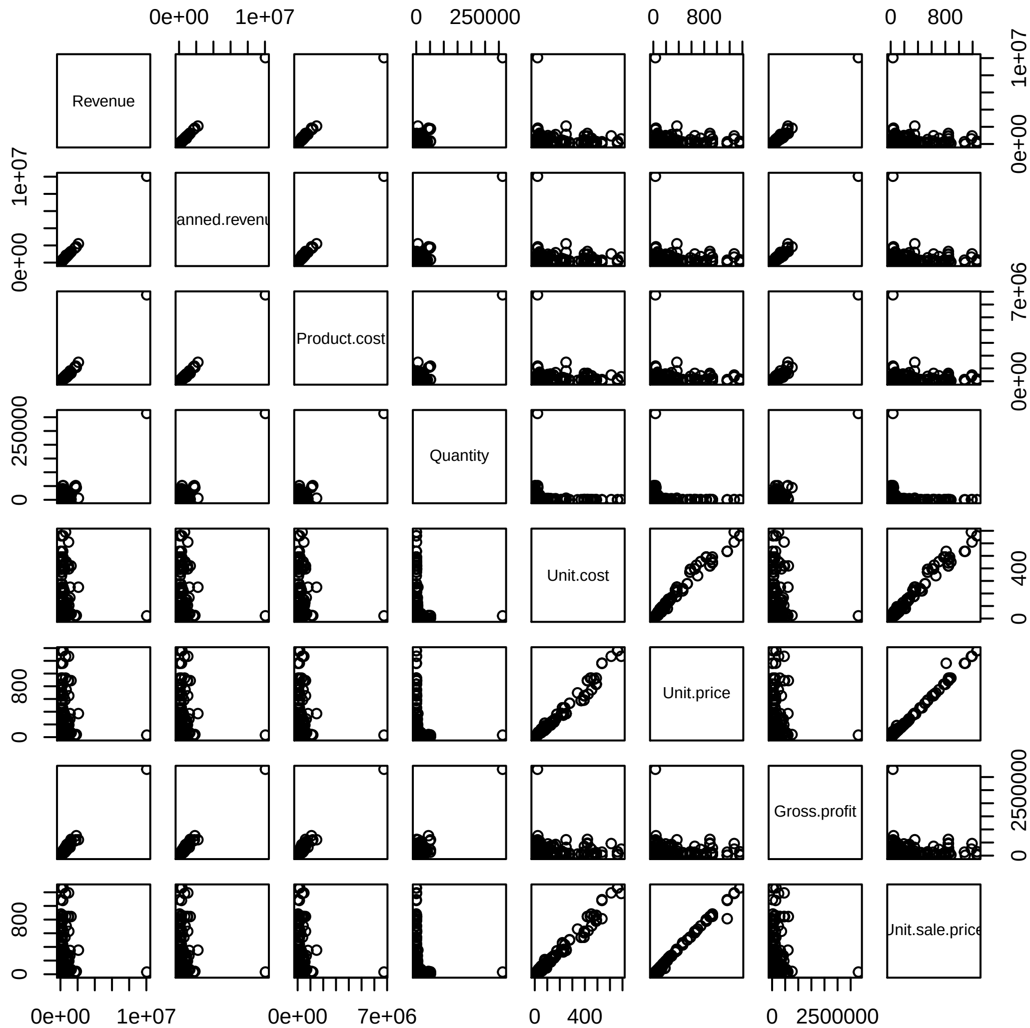

Figure 2: Scatterplot matrix of a sample of the dataset

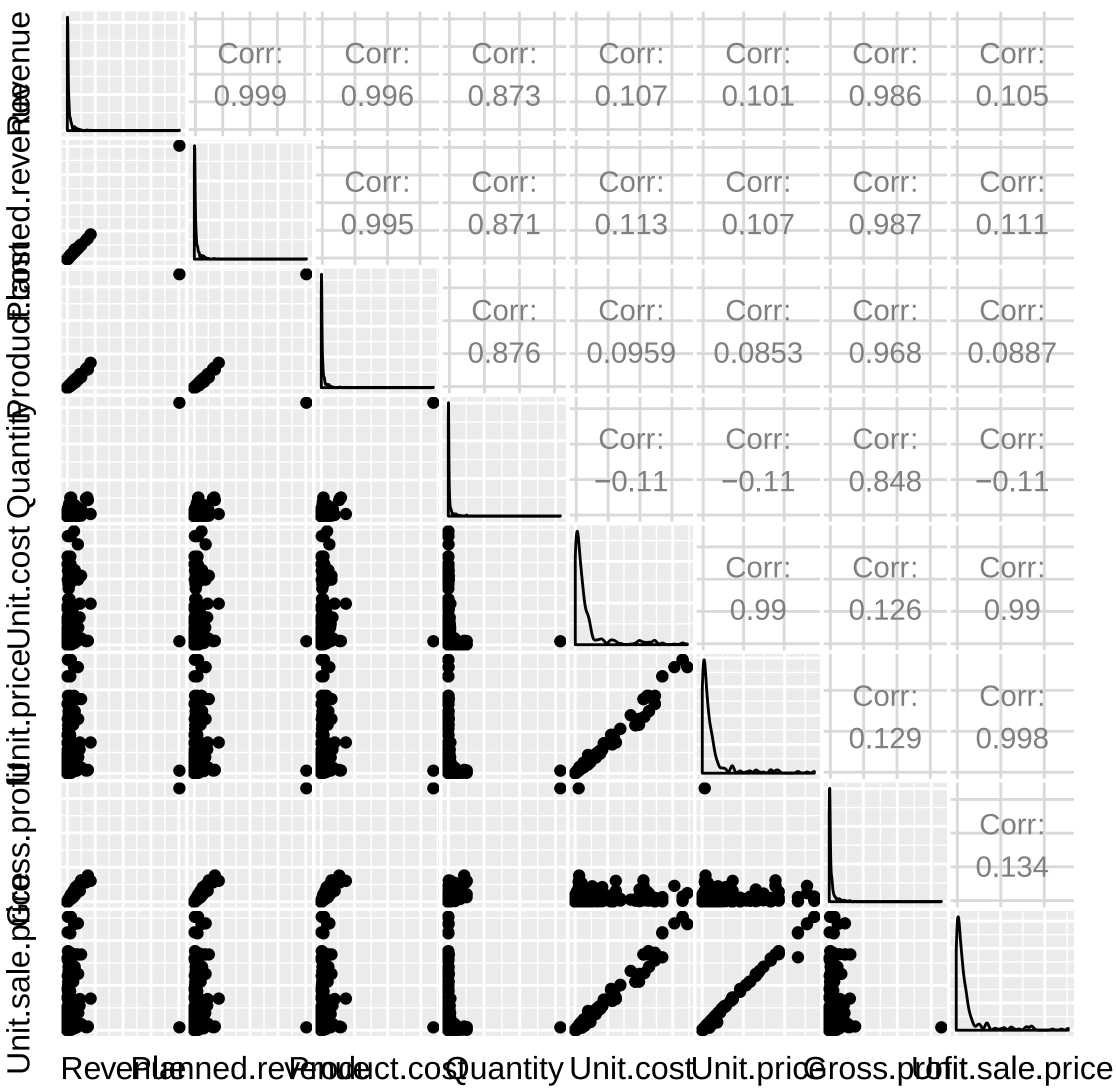

Figure 3: Scatterplot matrix of a sample of the dataset (with correlations)

After our EDA, we can divide the dataset into two separate ones, to train and evaluate the model:

```
# Year back to integer (factor only useful for vizs)
data <- data %>% mutate(Year = as.numeric(levels(Year))[Year])
# One dataset per couple of years
data_200405 <- data %>% filter(Year <= 2005)
data_200607 <- data %>% filter(Year > 2005)
```

Not all products appear in both periods so some re-factoring is needed:

```
# Re-factor Product (since the levels differ by period)
products_200405 <- data.frame(Product = levels(droplevels(data_200405$Product)))
products_200607 <- data.frame(Product = levels(droplevels(data_200607$Product)))
continuing_products <- intersect(products_200405, products_200607)
(new_or_discontinued_products <- union(products_200405, products_200607) %>%
  setdiff(continuing_products))
```

```
##        Product
## 1 Trail Master
## 2   Trail Star
## 3   Auto Pilot
```

```
# Products present in one period and not the other are labelled as "Other"
data_200405 <- data_200405 %>%
  mutate(Product = ifelse(Product %in% new_or_discontinued_products$Product,
                          "Other", as.character(Product))) %>%
  mutate(Product = factor(Product))
data_200607 <- data_200607 %>%
  mutate(Product = ifelse(Product %in% new_or_discontinued_products$Product,
                          "Other", as.character(Product))) %>%
  mutate(Product = factor(Product))
```

There are some variables that are calculated from `Revenue` (or vice versa) so including them in any regression model would lead to a perfect fit. In particular, `Gross.profit` = `Revenue - Product.cost`. And `Revenue` should be equal to `Unit.sale.price` times `Quantity`, though this is not always the case, and there are differences in many cases (53.4% of the total number of observations).

```
head(data %>% select(Revenue, Product.cost, Gross.profit) %>%
       mutate(Revenue2 = Product.cost + Gross.profit))
```

```
##     Revenue Product.cost Gross.profit  Revenue2
## 1 315044.33    158371.76    156672.57 315044.33
## 2  13444.68      6298.80      7145.88  13444.68
## 3 181120.24     89413.06     91707.18 181120.24
## 4  69608.15     35326.25     34281.90  69608.15
## 5  30940.35     16370.97     14569.38  30940.35
## 6  74321.18     36531.63     37789.55  74321.18
```

```
all(round(data$Revenue - data$Product.cost, 2) == round(data$Gross.profit, 2))
```

```
## [1] TRUE
```

```
head(data %>% select(Revenue, Unit.sale.price, Quantity) %>%
       mutate(Revenue2 = Unit.sale.price * Quantity))
```

```
##     Revenue Unit.sale.price Quantity  Revenue2
## 1 315044.33        5.195714    66385 344917.49
## 2  13444.68        6.190000     2172  13444.68
## 3 181120.24        5.488000    35696 195899.65
## 4  69608.15        5.040000    15205  76633.20
## 5  30940.35        3.950000     7833  30940.35
## 6  74321.18        5.585000    14328  80021.88
```

So Revenue and Product.cost should definitely not be included in the regression model, but Unit.sale.price and Quantity might.

Let's start with the simplest model:

```
# Simplest model
params = c("Planned.revenue")
model1 <- lm(as.formula(paste("Revenue", paste(params, sep = "",
                                                collapse = " + "),
                               sep = " ~ ")), data_200405)
coeftest(model1, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                   Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)     -3.2766e+03  3.5760e+02  -9.163 < 2.2e-16 ***
## Planned.revenue  9.6938e-01  2.7204e-03 356.333 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
new_data <- data.frame(data_200607[, params])
```

We'll use the RMSE to compare different models:

```
(RMSE <- sqrt(sum((model1_predictions[, 1] - data_200607$Revenue)^2) /
                dim(data_200607)[1]))
```

```
## [1] 19593.67
```

- **Is the change in the average revenue different from 95 cents when the planned revenue increases by $1?**

As shown below, the change in the average revenue is significantly different from \$0.95 when the revenue increases by \$1 (while the F statistic of the exact value, which is quite close to \$0.95, has a p value equal to 1):

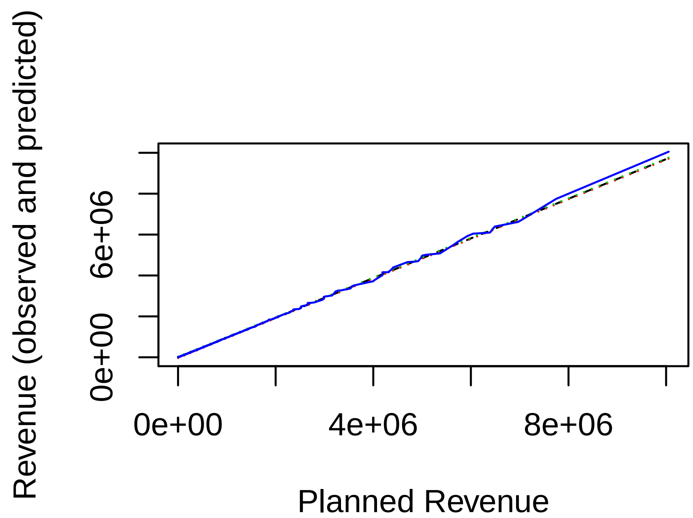

Figure 4: Planned Revenue vs. Revenue (observed and predicted) in 2006 and 2007

```
model1_full <- lm(as.formula(paste("Revenue", paste(params, sep = "",
                                                    collapse = " + "),
                              sep = " ~ ")), data)
coeftest(model1_full, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                     Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)     -3.3970e+03  3.0377e+02 -11.183 < 2.2e-16 ***
## Planned.revenue  9.6981e-01  1.8151e-03 534.297 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1_full, "Planned.revenue = 0.95", vcov = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## Planned.revenue = 0.95
##
## Model 1: restricted model
## Model 2: Revenue ~ Planned.revenue
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1  24742
## 2  24741  1 119.12 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
linearHypothesis(model1_full, paste("Planned.revenue =",
                                    coeftest(model1, vcov = vcovHC)[2, 1]),
                 vcov = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## Planned.revenue = 0.969384229459145
##
## Model 1: restricted model
## Model 2: Revenue ~ Planned.revenue
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1  24742
## 2  24741  1 0.0551 0.8145
```

```
params = c("Year", "Planned.revenue")
model2 <- lm(as.formula(paste("Revenue", paste(params, sep = "",
                                                collapse = " + "),
                              sep = " ~ ")), data_200405)
coeftest(model2, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                    Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)     -1.2835e+06  4.9020e+05  -2.6183  0.008846 **
## Year             6.3868e+02  2.4457e+02   2.6115  0.009025 **
## Planned.revenue  9.6935e-01  2.7230e-03 355.9794 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
new_data <- data.frame(data_200607[, params])
names(new_data) <- params
```

```
## [1] 19649.12
```

- **Explain what interaction terms in your model mean in context supported by data visualizations.**
- **Give two reasons why the OLS model coefficients may be biased and/or not consistent, be specific.**
- **Propose (but do not actually implement) a plan for an IV approach to improve your forecasting model.**

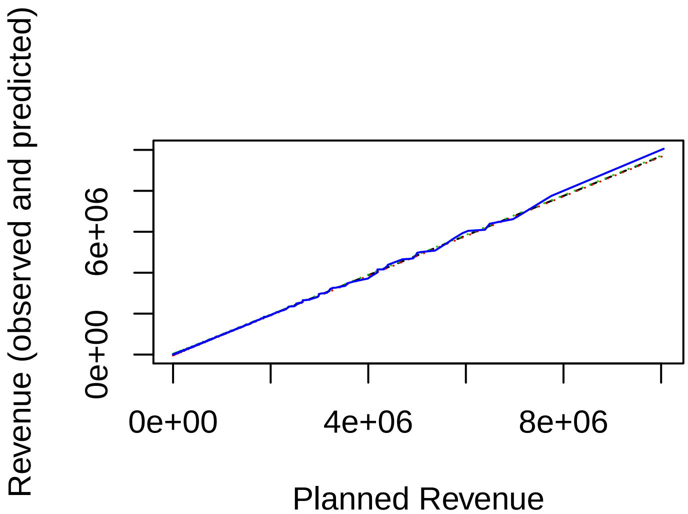

Figure 5: Planned Revenue vs. Revenue (observed and predicted) in 2006 and 2007