QMM: Bulk Carriers Data Case

Library setup

```
library(dplyr)
library(lubridate)
```

Step 1: Exploratory analysis

Remember first to make sure that you are working in `R` in the same directory where you data files are located.

1a)

First, load the data and check the column types of each.

```
ship1 <- read.csv("ship1.csv", head=TRUE, row.names = NULL)
str(ship1)
```

```
## 'data.frame':    881 obs. of  5 variables:
##  $ Date        : chr  "2004-01-01" "2004-01-01" "2004-01-01" "2004-01-01" ...
##  $ SellingPrice: num  18.5 5.3 13.8 9.8 11.8 35 34.5 30 8.8 13 ...
##  $ VesselAge   : int  10 21 22 22 22 0 1 5 20 23 ...
##  $ Dwt         : int  45518 54138 61615 62343 63770 55500 52500 48913 45090 64120 ...
##  $ Freight     : num  31689 31689 31689 31689 31689 ...
```

You see that the date is a Factor, not a Date. Coerce it to be using the `as.Date()` function. Then, check that everything is in order.

```
ship1$Date <- as.Date(ship1$Date)
str(ship1)
```

```
## 'data.frame':    881 obs. of  5 variables:
##  $ Date        : Date, format: "2004-01-01" "2004-01-01" ...
##  $ SellingPrice: num  18.5 5.3 13.8 9.8 11.8 35 34.5 30 8.8 13 ...
##  $ VesselAge   : int  10 21 22 22 22 0 1 5 20 23 ...
##  $ Dwt         : int  45518 54138 61615 62343 63770 55500 52500 48913 45090 64120 ...
##  $ Freight     : num  31689 31689 31689 31689 31689 ...
```

You can then proceed. Use the `apply()` function to compute the mean over the columns (with MARGIN=2 - or over the rows with MARGIN=1) of the variables, excluding the Date variable:

```
apply(ship1[,-1], MARGIN=2, mean)
```

```
## SellingPrice    VesselAge          Dwt      Freight
##     22.90647     12.63564  52634.71737  22189.27360
```

Do the same for the standard deviation, simply changing the function in apply:

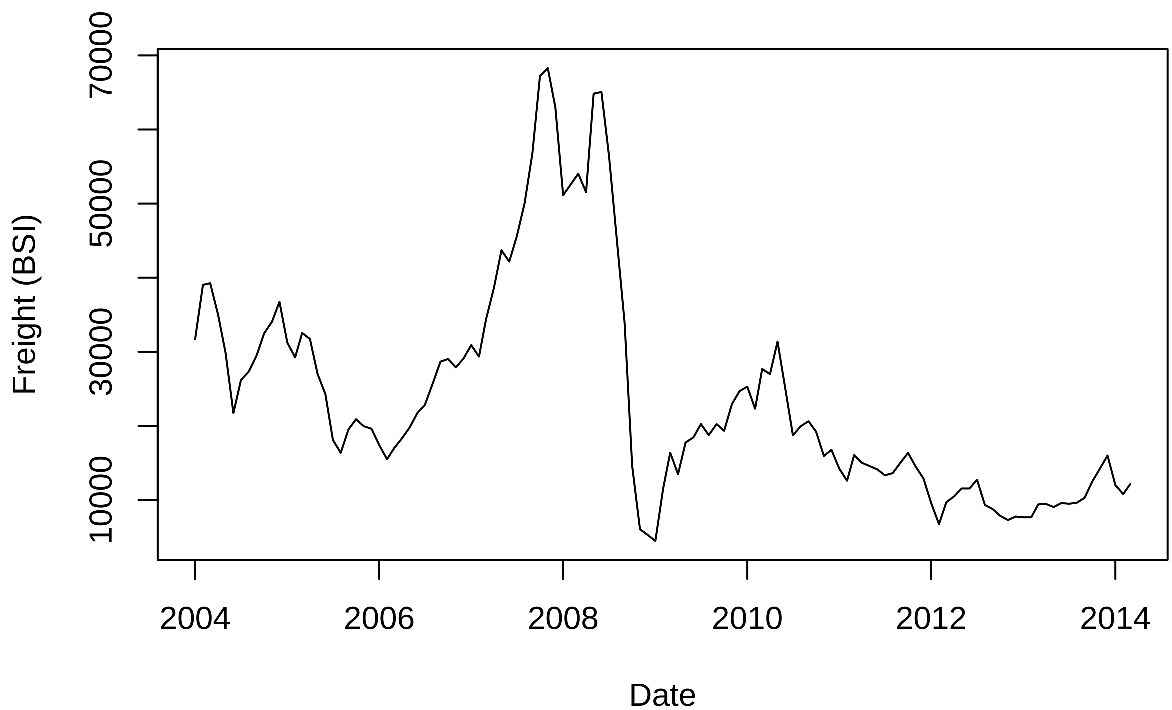
```
apply(ship1[,-1], MARGIN=2, sd)
```

```
## SellingPrice    VesselAge          Dwt      Freight
##    16.135586     9.463809  6659.531067 13321.019093
```

1b)

In the following, we plot the revenues over time.

```
plot(ship1$Date,ship1$Freight, type="l", xlab="Date",  ylab="Freight (BSI)")
```



We see that, in 2007 and 2008, the revenues were very high on average. This corresponds to a price bubble induced by the chinese economic boom (imports and exports were high at that time, alongside high speculation on the transportation prices).

1c)

In order to get the means of all variables for each year, one can use the `dplyr` verbs to do it efficiently. These verbs will be of crucial importance for those of you in the Business Analytics track that consider to take the Data Science lecture.

First, one has to "mutate" a new column to the existing data frame `ship1`. Then, one can group the observations by this new column and summarise all variables means in a new object and display it.

```
ship1 <- mutate(ship1, Year=year(Date))
means <- summarise_all(group_by(ship1, Year), "mean")
print.data.frame(means)
```

```
##    Year       Date SellingPrice VesselAge      Dwt   Freight
## 1  2004 2004-06-12     21.49420  12.24638 54617.87 33411.188
## 2  2005 2005-05-17     26.02025  10.24051 52609.42 26197.468
## 3  2006 2006-06-17     25.96889  11.02222 54054.11 22533.822
## 4  2007 2007-05-23     36.33086  13.62963 54791.68 44691.587
## 5  2008 2008-06-12     49.73333  10.05882 52919.80 39262.107
```

```
## 6  2009 2009-06-14     17.63950  13.83193 53294.06 17571.446
## 7  2010 2010-05-25     22.34938  12.59259 53392.56 22871.631
## 8  2011 2011-06-06     18.53372  14.04651 51835.19 14675.098
## 9  2012 2012-06-13     13.52326  15.44186 49096.17  9603.093
## 10 2013 2013-06-15     14.99820  12.36036 51230.11 10514.167
## 11 2014 2014-01-24     17.65714  10.53571 50395.50 11598.107
```

Using the so-called pipe operator, %>%, one rewrite the above code as:

```
ship1 <- ship1 %>% mutate(Year=year(Date))
(means <- ship1 %>% group_by(Year) %>% summarise(across(.fns=mean)))
```

This is another way to write the above code that helps to separate the operations. For example, when creating the means table, one uses the `ship1` data frame, group the observations by year and then summarise all columns by using their means.

1d)

The correlation matrix for the variables of interest can be found by coercing the variables of interest into a matrix and by then using the `cor()` function.

```
cor(as.matrix(ship1[,-c(1,6)]))
```

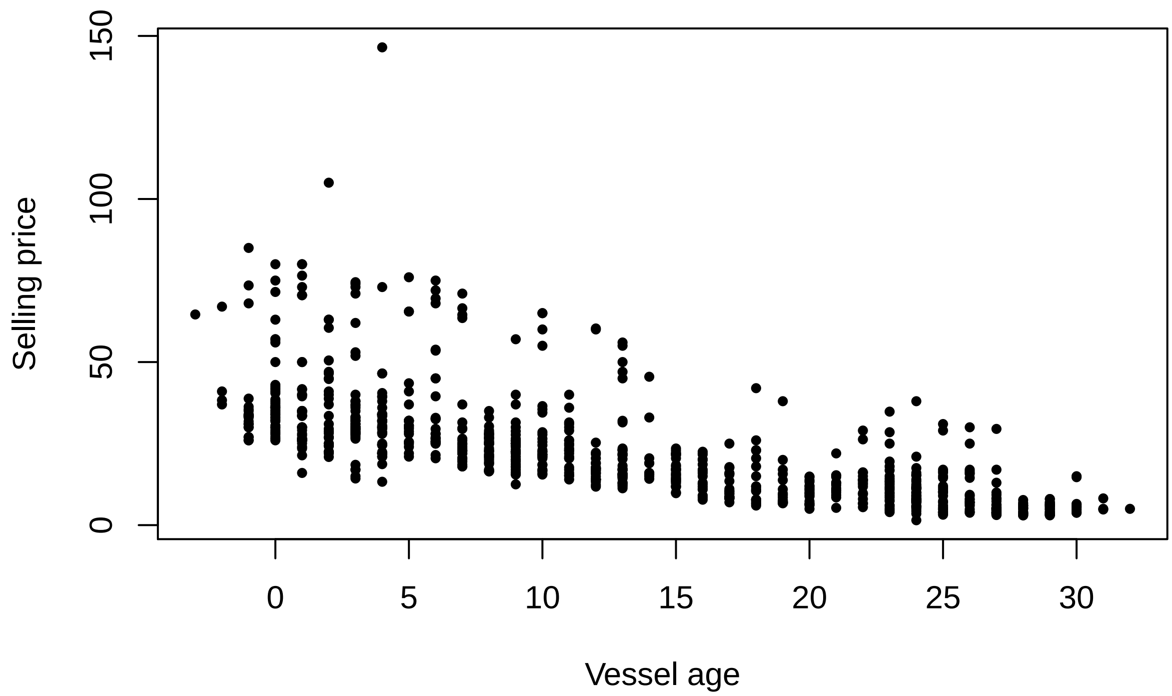
```
##              SellingPrice    VesselAge         Dwt     Freight
## SellingPrice   1.00000000 -0.694733541 0.030221612  0.56056080
## VesselAge     -0.69473354  1.000000000 0.008153955 -0.05515148
## Dwt            0.03022161  0.008153955 1.000000000  0.16325684
## Freight        0.56056080 -0.055151478 0.163256839  1.00000000
```

Step 2: Simple regression: selling price as a function of age

2a)

To easily access the columns, remember to use the `attach()` function:

```
attach(ship1)
plot(VesselAge, SellingPrice, xlab="Vessel age", ylab="Selling price", pch=16, cex=0.7)
```



2b)

To fit a linear model in the third order, one uses the `I()` function with the correct polynomial expression, as, above order 2, an expression of the form VesselAge^3 will not be recognised. One uses the usual `lm()` and one can then obtain the usual summary for a regression of this type:

```
mod1 <- lm(SellingPrice~VesselAge+I(VesselAge^2)+I(VesselAge^3))
summary(mod1)
```

```
##
## Call:
## lm(formula = SellingPrice ~ VesselAge + I(VesselAge^2) + I(VesselAge^3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.990  -5.923  -2.703   1.753 112.854
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    39.4026976  0.9211174  42.777   <2e-16 ***
## VesselAge      -1.4013284  0.3076993  -4.554    6e-06 ***
## I(VesselAge^2) -0.0125137  0.0268108  -0.467    0.641
## I(VesselAge^3)  0.0007679  0.0006284   1.222    0.222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.52 on 877 degrees of freedom
```

```
## Multiple R-squared:  0.4921, Adjusted R-squared:  0.4903
## F-statistic: 283.2 on 3 and 877 DF,  p-value: < 2.2e-16
```

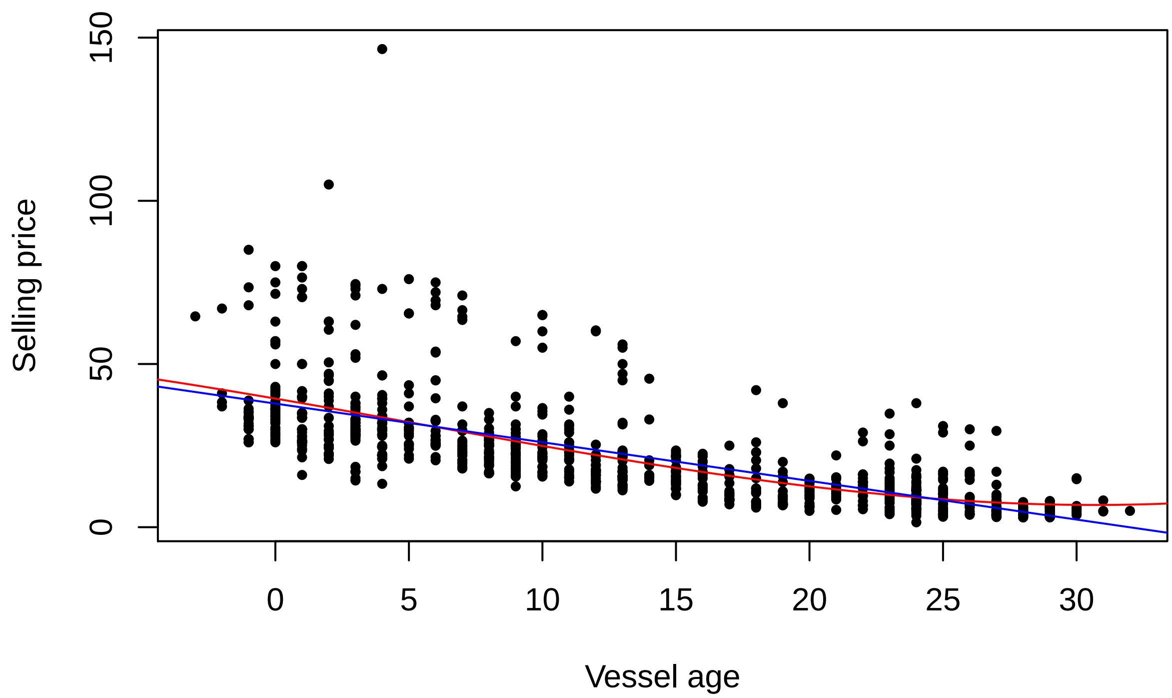
The regression equation is therefore:

$$sp_i = 39.4027 - 1.4013va_i - 0.0125va_i^2 + 0.0007va_i^3.$$

The coefficient of determination is 0.4921. This means that the cubic model explains approximately half of the variation in the selling price.

If you want now to plot the regression line on the scatterplot from point 2a), you have to use a trick: you first create a sequence of points covering the interval of your x-axis, below it is the `seq` sequence, with a fine enough resolution (I chose steps of 0.5 for the line to be smooth enough); then, for each of the position in the sequence, you use the regression coefficients from the model fitted above and you use as the value of VesselAge this position in the sequence. The result is the red curve on the graph below.

```
plot(VesselAge, SellingPrice, xlab="Vessel age", ylab="Selling price", pch=16, cex=0.7)
seq <- seq(from=-5, to=35, by=0.5)
lines(seq, as.numeric(mod1$coefficients[1])+as.numeric(mod1$coefficients[2])*seq
  +as.numeric(mod1$coefficients[3])*seq^2+as.numeric(mod1$coefficients[4])*seq^3, col="red")
abline(lm(SellingPrice~VesselAge), col="blue")
```



For comparison, I added a blue curve with the automatic procedure when one has a simple linear regression, simply in VesselAge. One can see that the model in the third order better fits the behaviour of the points, at a cost of maybe overfitting.

2c)

In order to address this question, we first create a vector with 36 rows corresponding to the vessel ages from 0 to 35 years old. We fit our model at each of these ages and we create a new data frame with the two. We then use a for loop to obtain the relative changes over time. We augment our data frame with this new information.

```
ageseq <- seq(0, 35, by=1)
fitval <- as.numeric(mod1$coefficients[1])+as.numeric(mod1$coefficients[2])*ageseq
  +as.numeric(mod1$coefficients[3])*ageseq^2+as.numeric(mod1$coefficients[4])*ageseq^3
dfmod1 <- as.data.frame(cbind(ageseq, fitval))

relvar <- c()
for(i in 1:nrow(dfmod1)){
  if(i==1){
    relvar[i] <- 0
  }else{
    relvar[i] <- (dfmod1$fitval[i]-dfmod1$fitval[i-1])/dfmod1$fitval[i-1]
  }
}
```

The resulting dataframe is:

```
dfmod1 <- as.data.frame(cbind(dfmod1, relvar))
print(dfmod1)
```

```
##    ageseq     fitval      relvar
## 1       0 39.4026976  0.00000000
## 2       1 38.0013692 -0.03556428
## 3       2 36.6000408 -0.03687573
## 4       3 35.1987124 -0.03828762
## 5       4 33.7973840 -0.03981192
## 6       5 32.3960556 -0.04146263
## 7       6 30.9947272 -0.04325614
## 8       7 29.5933989 -0.04521183
## 9       8 28.1920705 -0.04735274
## 10      9 26.7907421 -0.04970647
## 11     10 25.3894137 -0.05230644
## 12     11 23.9880853 -0.05519341
## 13     12 22.5867569 -0.05841768
## 14     13 21.1854285 -0.06204204
## 15     14 19.7841001 -0.06614586
## 16     15 18.3827717 -0.07083104
## 17     16 16.9814433 -0.07623053
## 18     17 15.5801149 -0.08252116
## 19     18 14.1787865 -0.08994339
## 20     19 12.7774581 -0.09883275
## 21     20 11.3761297 -0.10967192
## 22     21  9.9748013 -0.12318147
## 23     22  8.5734729 -0.14048685
## 24     23  7.1721446 -0.16344933
## 25     24  5.7708162 -0.19538485
## 26     25  4.3694878 -0.24283019
## 27     26  2.9681594 -0.32070771
## 28     27  1.5668310 -0.47212033
## 29     28  0.1655026 -0.89437113
```

```
## 30      29 -1.2358258 -8.46710878
## 31      30 -2.6371542  1.13392064
## 32      31 -4.0384826  0.53137901
## 33      32 -5.4398110  0.34699379
## 34      33 -6.8411394  0.25760608
## 35      34 -8.2424678  0.20483845
## 36      35 -9.6437962  0.17001321
```

What we observe is that the selling prices are decreasing up to 31 years old and that they increase after this, by substantial amount. However, note that they increase in regions over which we have no data points (we have no vessels aged more than 32 years old). This is just the behaviour that you see in the graph above, with the red curve increasing again for high ages, a particularity of this third order model. You would observe a constant decrease with a simple linear model.

2d)

Thanks to the description of the exponential model in terms of a linear regression, one simply has to convert the selling prices in log and to then undertake a simple linear regression.

```
lnSellingPrice = log(SellingPrice) #note that the log is by default the natural log
mod2 <- lm(lnSellingPrice~VesselAge)
summary(mod2)
```

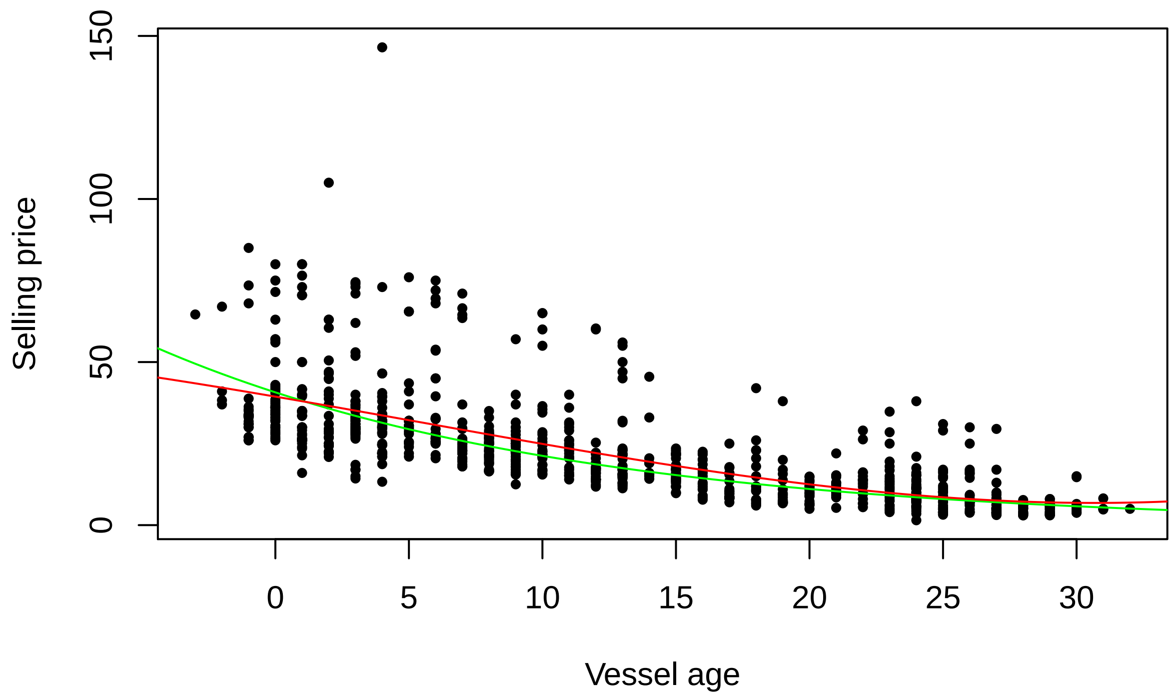
```
##
## Call:
## lm(formula = lnSellingPrice ~ VesselAge)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73967 -0.26003 -0.05492  0.20728  1.54074
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.706520   0.023385  158.50   <2e-16 ***
## VesselAge   -0.065058   0.001482  -43.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4159 on 879 degrees of freedom
## Multiple R-squared:  0.6869, Adjusted R-squared:  0.6865
## F-statistic:  1928 on 1 and 879 DF,  p-value: < 2.2e-16
```

Now, to find β_0, one has to take the exponential (the inverse of the log base e), which is $e^{3.706520} = 40.7119$ to report the model equation in the following form:

$$sp_i = 40.7119e^{-0.0651va_i}.$$

The coefficient of determination is 0.6869, a bit higher than the previous one. This model seems to better explain the variation in the response, i.e. the selling price. One can again plot the regression line of the exponential model on the scatterplot of 2a), by simply changing the equation in the above code, again using the sequence along the x-axis to create the curve, in green in the graph below:

```
plot(VesselAge, SellingPrice, xlab="Vessel age", ylab="Selling price", pch=16, cex=0.7)
seq <- seq(from=-5, to=35, by=0.5)
lines(seq, exp(as.numeric(mod2$coefficients[1]))*exp(mod2$coefficients[2]*seq), col="green")
lines(seq, as.numeric(mod1$coefficients[1])+as.numeric(mod1$coefficients[2])*seq
  +as.numeric(mod1$coefficients[3])*seq^2+as.numeric(mod1$coefficients[4])*seq^3, col="red")
```



For comparison, I added in red the curve corresponding to the cubic model. Note that, with the exponential model, we do not have this increasing effect in the right tail of the selling price, as we do in the cubic model. However, note that this exponential model does not take into account the residual value of the ships. The green curve decreases smoothly, yet when a ship is sold to be demolished, the scrapped materials (steel and others) are still worth 4 to 5 million USD, behaviour inconsistent with such exponential model.