Used Car Model

Model Result:

$$SalesPrice = 60060 - 11010 * CarAge - 0.000576 * Mileage + 0.5062 * MSRP$$

Model, Car Model, 车型
VIN, VIN Code, VIN码
Level, Level Code, Level_ID
Brand, Car Brand, 品牌
Series, Car series, 车系
LaunchYr, Launch year, 上市年
LaunchMo, Launch month, 上市月
MSRP, Manufacture Suggested Retail Price(New Car), 新车指导价
Mileage, Mileage(Km), 行驶里程
Accident, Whether it is a serious accident(Yes or No), 是否大事故
PurchasedDate, Purchase Date(yyyy/mm/dd), 采购日期
Insurance, Amount of insurance, 保险金额
Renovation, Amount of Renovation Cost(some not necessarily renovate), 整备金额
Cost, Cost of purchase, 采购价格
SalesPrice, Sales Price After Maintenance 整备后销售价格

To import the `r20150708.csv` file into R, you can use the `read.csv()` function. Here's an example:

```
# Import the CSV file
data <- read.csv("r20150708.csv")
data <- read.csv("/Users/chang/Documents/Data/Used Car Model/r20150708.csv")

# View the data
head(data)
```

In this example, the `read.csv()` function is used to import the `r20150708.csv` file into R and store it in the `data` object. The `head()` function is then used to view the first few rows of the data.

To run a used car price model using R, you could use a variety of techniques, such as linear regression, decision trees, or support vector machines (SVMs). Here's an example of how to train an SVM model using the `e1071` package in R:

```
# Load the e1071 package
library(e1071)

# Split the data into training and testing sets
train <- data[1:2236, ]
test <- data[2237:2796, ]

# Train an SVM model using the training set
svm_model <- svm(SalesPrice ~ Mileage + Renovation + Cost, data = train)

# Make predictions on the testing set
predictions <- predict(svm_model, test)

# Calculate the mean squared error of the predictions
mse <- mean((test$SalesPrice - predictions)^2)
print(paste0("Mean squared error: ", mse))
```

Gravel Factor Model by R

I want to know how to realise the gravel factor model by R, that mean, I want to know which factor is the most important factor, and the weight to calculate the used car sales price. I just want to get a simply formula and apply it by excel, when I know a used car's Lacuch year / month, mileage and MSRP, Accident, and the purchaseDate, I can calculate the Cost of purchase.

`To build a gravel factor model in R, we can use a technique called linear regression. Linear regression is a statistical method that allows you to model the relationship between a dependent variable (in this case, the sales price of a used car) and one or more independent variables (in this case, the car's launch year, mileage, MSRP, whether it was in an accident, and the purchase date).`

Here's an example of how to build a linear regression model in R using your data:

```
# Load the data from the CSV file
data <- read.csv("/Users/chang/Documents/Data/Used Car Model/r20150708.csv")

# Convert the "Launch year" and "Launch month" columns to a single date column
data$LaunchDate <- as.Date(paste(data$LaunchYr, data$LaunchMo, "1", sep = "-"))

# Convert the "Purchase Date" column to a date object
data$PurchaseDate <- as.Date(data$PurchasedDate, format = "%Y/%m/%d")
```

```
# Calculate the age of the car at the time of purchase
data$CarAge <- as.numeric(difftime(data$PurchasedDate, data$LaunchDate, units =

# Create a binary variable indicating whether the car was in an accident or not
data$AccidentBinary <- ifelse(data$Accident == "Yes", 1, 0)

# Create a linear regression model with SalesPrice as the dependent variable an
model <- lm(SalesPrice ~ CarAge + Mileage + MSRP + AccidentBinary, data = data)

# Print the model summary
summary(model)
```

In this example, we first load the data from the CSV file and prepare it by converting the "Launch year" and "Launch month" columns to a single date column, converting the "Purchase Date" column to a date object, and calculating the age of the car at the time of purchase. We also create a binary variable indicating whether the car was in an accident or not.

Next, we create a linear regression model using the `lm()` function, with `SalesPrice` as the dependent variable and `CarAge`, `Mileage`, `MSRP`, and `AccidentBinary` as independent variables. We then print the model summary using the `summary()` function, which will show us the coefficients and p-values for each variable in the model.

The coefficients in the model represent the weights that each variable contributes to the calculation of the used car sales price. For example, if the coefficient for `Mileage` is -0.03, it means that for every additional kilometer on the odometer, the predicted sales price will decrease by 0.03 units (assuming all other variables are held constant).

Once you have the coefficients for each variable, you can use them to calculate the sales price of a used car using a formula like this:

```
SalesPrice = b0 + b1*CarAge + b2*Mileage + b3*MSRP + b4*AccidentBinary
```

where `b0`, `b1`, `b2`, `b3`, and `b4` are the coefficients from the linear regression model, and `CarAge`, `Mileage`, `MSRP`, and `AccidentBinary` are the values for each variable for the used car you're interested in.

You can then implement this formula in Excel to calculate the sales price of a used car given its launch year/month, mileage, MSRP, whether it was in an accident, and the purchase date.

```
> summary(model)
```

```
Call:
lm(formula = SalesPrice ~ CarAge + Mileage + MSRP + AccidentBinary,
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-356656  -11363   -1154    9068 1146491

Coefficients: (1 not defined because of singularities)
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     6.006e+04  1.779e+03  33.760   <2e-16 ***
CarAge         -1.101e+04  2.542e+02 -43.295   <2e-16 ***
Mileage        -5.760e-04  7.214e-04  -0.798    0.425
MSRP            5.062e-01  6.139e-03  82.459   <2e-16 ***
AccidentBinary         NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33740 on 2792 degrees of freedom
Multiple R-squared:  0.7664,    Adjusted R-squared:  0.7662
F-statistic:  3054 on 3 and 2792 DF,  p-value: < 2.2e-16
```

The summary output of the model provides information on the coefficients, standard errors, t-values, and p-values for each of the predictor variables in the model. Here's a breakdown of the key information:

- The intercept coefficient represents the expected value of the response variable (SalesPrice) when all predictor variables are equal to zero. In this case, the intercept is estimated to be 60,060.
- The CarAge coefficient represents the expected change in SalesPrice for each one unit increase in CarAge. A one unit increase in CarAge is associated with a decrease of 11,010 in SalesPrice.
- The Mileage coefficient represents the expected change in SalesPrice for each one unit increase in Mileage. However, the coefficient estimate is not statistically significant (p-value = 0.425), which suggests that there is not strong evidence of a linear relationship between Mileage and SalesPrice.
- The MSRP coefficient represents the expected change in SalesPrice for each one unit increase in MSRP. A one unit increase in MSRP is associated with an increase of 0.5062 in SalesPrice.
- The AccidentBinary coefficient is not estimated because it is a binary variable and the model is using dummy coding. The model is estimating the difference in SalesPrice between cars with and without accidents.

- The Residual standard error is an estimate of the standard deviation of the error term, which represents the variation in SalesPrice that is not explained by the predictor variables. The value is 33,740, which suggests that the model is able to explain a large proportion of the variability in SalesPrice.
- The Multiple R-squared is a measure of the proportion of variance in the response variable that is explained by the predictor variables. In this case, the model explains 76.64% of the variance in SalesPrice.
- The Adjusted R-squared adjusts the Multiple R-squared for the number of predictor variables in the model. In this case, the Adjusted R-squared is very similar to the Multiple R-squared, which suggests that there is no penalty for the inclusion of the predictor variables in the model.
- The F-statistic is a measure of the overall significance of the model. The value is very high (3054) and the p-value is very low (< 2.2e-16), which suggests that the model is highly significant and that at least one of the predictor variables is significantly associated with SalesPrice.

Based on the output of the summary of the linear regression model, the final formula for SalesPrice would be:

$$SalesPrice = 6.006e+04 - 1.101e+04 * CarAge - 5.760e-04 * Mileage + 5.062e$$
$$SalesPrice = 60060 - 11010 * CarAge - 0.000576 * Mileage + 0.5062 * MSRP$$

Note that the coefficient for the AccidentBinary variable is not defined due to singularities.

Draw a plot for these factors

Yes, it is possible to create a plot to show the weight of each factor. One way to do this is to use the ggplot2 package in R.

Here's an example code that you can modify to create your own plot:

```
library(ggplot2)

# Extract coefficients from the model
coef_df <- data.frame(names = names(model$coefficients)[-4],
                      coef = model$coefficients[-4])

# Create a bar plot of the coefficients
ggplot(coef_df, aes(x = names, y = coef, fill = names)) +
  geom_bar(stat = "identity") +
  labs(x = "Factor", y = "Weight") +
```

```
    ggtitle("Weight of Factors in Used Car Price Model")
```

This code will create a bar plot that shows the weight of each factor in the used car price model. The x-axis shows the factor names, the y-axis shows the weight of each factor, and the fill color of each bar corresponds to the factor name. You can customize the plot by changing the axis labels, title, and color palette.