University of Applied Sciences and Arts Northwestern Switzerland
School of Business

Data Science - Laboratory

Holger Wache

Predicting the Price

How can we predict the price?

brand	type	style	mpg	cyl	disp	wt	gear	carb	price
Mazda	coupe	basic	21	-6	160	2.62	4	4	33500
Mazda	station	medium	21	6	160	2.875	4.1	4	32500
Datsun	limosine	basic	0	4	108	2.32	3.9	1	16000
Hornet	station	basic	21.4	6	258	3.215	3.2	1	26700
Hornet	station	basic	18.7	8	360	3.44	3	2	25350
Vailant	limosine	basic	18.1	6	225	346	2.9	1	21050
Duster	limosine	basic	14.3	8	360	3.57	2.8	4	19650
Mercedes	limosine	basic	24.4	4	146.7	3.19	4	2	36200
Mercedes	station	basic		4	140.8	3.15	3.9	2	26000
Mer	limosine	med		6			4.1		
	limosine								

Predicting the Price

How can we predict the price?

brand	type	style	mpg	cyl	disp	wt	gear	carb	price
Mazda	coupe	basic	21	-6	160	2.62	4	4	33500
Mazda	station	medium	21	6	160	2.875	4.1	4	32500
Datsun	limosine	basic	0	4	108	2.32	3.9	1	16000
Hornet	station	basic	21.4	6	258	3.215	3.2	1	26700
Hornet	station	basic	18.7	8	360	3.44	3	2	25350
Vailant	limosine	basic	18.1	6	225	346	2.9	1	21050
Duster	limosine	basic	14.3	8	360	3.57	2.8	4	19650
Mercedes	limosine	basic	24.4	4	146.7	3.19	4	2	36200
Mercedes	station	basic		4	140.8	3.15	3.9	2	26000
Mer...	limosine	med...		6			4.1		
	...mosine								

Predicting it with linear regression:

$$Price = a_1 * \ brand + a_2 * type + a_3 * style + a_4 * mpg + \ a_5 * cyl + a_6 * disp + a_7 * wt + a_8 * gear + a_9 * carb$$

Prepare the Data

A data set is given: “cgh_cars”. But this data set is dirty and needs to be prepared for a linear regression:

- Data Cleaning
- Data Transformation
- Variable Transformation
- … ???

Your Task: Fokus the data wrangling, i.e. clean the data and prepare it! 😀

Prerequisites

First, install additional packages

1. an additional package containing nice functions

```
install.packages("tidyr")
```

After the installation these packages need to be "activated"

```
library(tidyr)
```

0: Read the file

Where is the file stored?

```
setwd("~/FHNW/O365_G_DataScience_Admin - General/_2021/Slides/02-
DataPreprocessing/_laboratory")
```

Read the file and store the content into “cgh_cars”

```
cgh_cars <- read.csv("cgh_cars.csv")
```

A.1: Show the data (1/2)

Get a first impression:

```
> cgh_cars
       brand        type  style  mpg cyl  disp       wt gear carb price
1      Mazda       coupe  basic 21.0  -6 160.0    2.620  4.0    4 33500
2      Mazda     station medium 21.0   6 160.0    2.875  4.1    4 32500
3     Datsun    limosine  basic  0.0   4 108.0    2.320  3.9    1 16000
4     Hornet     station  basic 21.4   6 258.0    3.215  3.2    1 26700
5     Hornet     station  basic 18.7   8 360.0    3.440  3.0    2 25350
6    Vailant    limosine  basic 18.1   6 225.0  346.000  2.9    1 21050
7     Duster    limosine  basic 14.3   8 360.0    3.570  2.8    4 19650
8   Mercedes    limosine  basic 24.4   4 146.7    3.190  4.0    2 36200
9   Mercedes     station  basic   NA   4 140.8    3.150  3.9    2 26000
…
```

There are variables that a qualitative. They needs to be translated into numbers
→ Task VT.1-VT.3

A.1: Show the data (2/2)

Get further impressions:

```
> str(cgh_cars)
'data.frame':    32 obs. of  10 variables:
 $ brand: chr  "Mazda" "Mazda" "Datsun" "Hornet" ...
 $ type : chr  "coupe" "station" "limosine" "station" ...
 $ style: chr  "basic" "medium" "basic" "basic" ...
 $ mpg  : num  21 21 0 21.4 18.7 18.1 14.3 24.4 NA 19.2 ...
 $ cyl  : int  -6 6 4 6 8 6 8 4 4 6 ...
 $ disp : num  160 160 108 258 360 ...
 $ wt   : num  2.62 2.88 2.32 3.21 3.44 ...
 $ gear : num  4 4.1 3.9 3.2 3 2.9 2.8 4 3.9 4.1 ...
 $ carb : int  4 4 1 1 2 1 4 2 2 4 ...
 $ price: int  33500 32500 16000 26700 25350 21050 19650 36200 26000 34600 ...
```

A.2: Are there NAs? (1/2)

Analyse, if there are NAs:

```
> is.na(cgh_cars)
      brand  type style   mpg   cyl  disp    wt  gear  carb price
 [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
…
```

Good start but hard to analyse. Let's count NAs

```
> sum(is.na(cgh_cars))
[1] 5
```

Okay, there are NA… (5)

A.2: Are there NAs? (2/2)

But in which columns?

```
> colSums(is.na(cgh_cars))
brand  type style   mpg   cyl  disp    wt  gear  carb price
    0     0     0     4     0     0     0     0     0     1
```

Okay, we have to handle

- MPG (→ Task C.2)
- PRICE (→ Task C.3)

A.3: Also analyse the other values… (1/2)

"summary" gives you a compact overview

```
> summary(cgh_cars)
    brand               type              style                mpg              cyl
 Length:32          Length:32          Length:32          Min.   : 0.00   Min.   :-8.000
 Class :character   Class :character   Class :character   1st Qu.:14.93   1st Qu.: 4.000
 Mode  :character   Mode  :character   Mode  :character   Median :17.95   Median : 6.000
                                                          Mean   :18.33   Mean   : 4.812
                                                          3rd Qu.:21.10   3rd Qu.: 8.000
                                                          Max.   :33.90   Max.   : 8.000
                                                          NA's   :4
      disp             wt              gear            carb           price
 Min.   : 71.1   Min.   :   1.513   Min.   :2.800   Min.   :1.000   Min.   :16000
 1st Qu.:120.8   1st Qu.:   2.777   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:25250
 Median :196.3   Median :   3.440   Median :3.900   Median :2.000   Median :30000
 Mean   :230.7   Mean   :  81.349   Mean   :3.719   Mean   :2.812   Mean   :30635
 3rd Qu.:326.0   3rd Qu.:   3.841   3rd Qu.:4.100   3rd Qu.:4.000   3rd Qu.:34050
 Max.   :472.0   Max.   :2140.000   Max.   :5.200   Max.   :8.000   Max.   :59850
                                                                    NA's   :1
```

A.3: Also analyse the other values… (2/2)

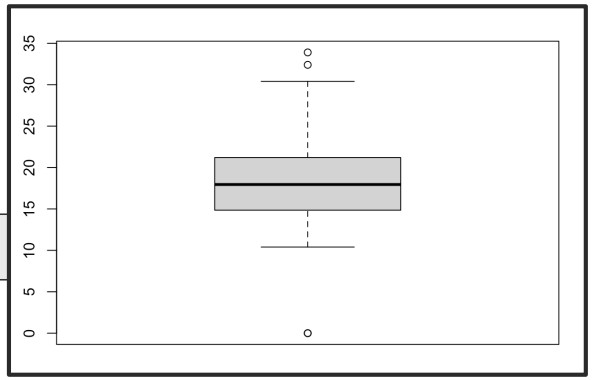
Apart from the former issues we also identify the following additional issues:

- MPG has value “0”, that can’t be. (→ Task C.1)
- CYL has negative values, that can’t be (→ Task C.4)
- WT has extreme values, close to either to < 10 or > 80, that’s strange (→ Task C.5)
- GEAR has rational numbers (numbers with a dot), that’s impossible(→ Task C.6)
- (DISP may be needed to scaled (→ Task T.1))

A.4: Also analyse the correlations…

The correlation analysis can only be performed on quantitative values. But the first three rows are qualitative. Therefore the correlation analysis needs to be postponed after the variable transformation (i.e. Task VT.3)

C.1: Zeros in “mpg”



Further analyse ”mpg”

```
> boxplot(cgh_cars$mpg)
```

It seems that 0 are outliners.

We can handle them by replace them with NA

```
cgh_cars$mpg <- ifelse(cgh_cars$mpg == 0.0, NA, cgh_cars$mpg)
```

Check, if we now have more NA but only in “mpg”

```
colSums(is.na(cgh_cars))
```

Looks good!

C.2: NAs in “mpg” (1/2)

Proposal: Replace NAs with the mean (OR the median)

BTW, NA values impact the calculation of mean

```
mean(cgh_cars$mpg)
```

Ignore NA and calculate mean

```
mean(cgh_cars$mpg, na.rm = TRUE)
```

C.2: NAs in “mpg” (2/2)

Values missing in mpg column, fill up by calculating mean

```
cgh_cars$mpg <- ifelse(is.na(cgh_cars$mpg),
                       mean(cgh_cars$mpg, na.rm=TRUE),
                       cgh_cars$mpg)
```

OR with the median

```
cgh_cars$mpg <- ifelse(is.na(cgh_cars$mpg),
                       median(cgh_cars$mpg, na.rm=TRUE),
                       cgh_cars$mpg)
```

Check, if the NAs in “mpg” disappeared:

```
> colSums(is.na(cgh_cars))
brand  type style   mpg   cyl  disp    wt  gear  carb price
    0     0     0     0     0     0     0     0     0     1
```

C.3: NAs in “price”

“price” is the target variable. Replacing NA may result in wrong learnings. It’s better to drop these lines

There is a nice function in the library “tidyr”: drop_na

```
cgh_cars <- drop_na(cgh_cars,price)
```

Check, if the NAs in “mpg” disappeared:

```
> colSums(is.na(cgh_cars))
brand  type style   mpg   cyl  disp    wt  gear  carb price
    0     0     0     0     0     0     0     0     0     0
```

C.4: Outliers in “cyl” (1/2)

Check if there are negative values

```
cgh_cars["cyl"] < 0
```

Find the number of negative values

```
colSums(cgh_cars["cyl"] < 0)
```

Please note the difference between the two commands:

```
typeof(cgh_cars$cyl)
[1] "integer"
typeof(cgh_cars["cyl"])
[1] "list"
```

C.4: Outliers in “cyl” (2/2)

Replace negative numbers by positive

```
cgh_cars$cyl = ifelse((cgh_cars$cyl < 0),
                      cgh_cars$cyl * -1,
                      cgh_cars$cyl)
```

Verify if successfully replaced

```
colSums(cgh_cars["cyl"] < 0)
```

C.4: Outliers in “cyl” (2/2) (VARIANT)

Replace negative numbers by positive by using the function “abs” (absolute value)

```
cgh_cars$cyl = abs(cgh_cars$cyl)
```

Verify if successfully replaced

```
colSums(cgh_cars["cyl"] < 0)
```

C.5: Missing decimals and scale problems in “wt” (1/3)

Find numbers that have a quotient more than 1 digit (outliers with very high weight)

```
cgh_cars["wt"] %/% 10 > 0
```

Count them

```
> colSums(cgh_cars["wt"] %/% 10 > 0 )
wt
 3
```

List row details

```
> cgh_cars[cgh_cars$wt %/% 10 > 0, ]
     brand     type style      mpg cyl  disp   wt gear carb price
6  Vailant limosine basic 18.10000   6 225.0  346  2.9    1 21050
17    Fiat limosine basic 32.40000   4  78.7   22  4.0    1 31200
26 Porsche    coupe basic 19.73846   4 120.3 2140  5.1    2 33000
```

C.5: Missing decimals and scale problems in “wt” (2/3)

We need to convert each one by one:

First convert the largest value

```
cgh_cars[cgh_cars$wt %/% 1000 > 0, ]
cgh_cars$wt <- ifelse((cgh_cars$wt %/% 1000 > 0), cgh_cars$wt/1000, cgh_cars$wt)
```

Now convert the second largest value

```
cgh_cars[cgh_cars$wt %/% 100 > 0, ]
cgh_cars$wt <- ifelse((cgh_cars$wt %/% 100 > 0), cgh_cars$wt/100, cgh_cars$wt)
```

Finally the smallest conversion

```
cgh_cars[cgh_cars$wt %/% 10 > 0, ]
cgh_cars$wt <- ifelse((cgh_cars$wt %/% 10 > 0), cgh_cars$wt/10, cgh_cars$wt)
```

C.5: Missing decimals and scale problems in “wt” (3/3)

Verify if all conversions are complete

```
> colSums(cgh_cars["wt"] %/% 10 > 0 )
wt
 0
```

C.6: Noisy Data in “gear” (1/2)

We need to convert them. One possibility: One bin by one bin:

Everything below 1.5 will be considered at 1 gear

```
cgh_cars$gear <- ifelse((cgh_cars$gear < 1.5), 1, cgh_cars$gear)
```

Everything between 1.5 and 2.499 will be considered as 2 gears

```
cgh_cars$gear <- ifelse((cgh_cars$gear >= 1.5) & (cgh_cars$gear < 2.5), 2, cgh_cars$gear)
```

Everything between 2.5 and 3.499 will be considered as 3 gears

```
cgh_cars$gear <- ifelse((cgh_cars$gear >= 2.5) & (cgh_cars$gear < 3.5), 3, cgh_cars$gear)
```

C.6: Noisy Data in “gear” (2/2)

Everything between 3.5 and 4.499 will be considered as 4 gears

```
cgh_cars$gear <- ifelse((cgh_cars$gear >= 3.5) & (cgh_cars$gear < 4.5), 4, cgh_cars$gear)
```

Everything between 4.5 and 5.499 will be considered as 5 gears

```
cgh_cars$gear <- ifelse((cgh_cars$gear >= 4.5) & (cgh_cars$gear < 5.5), 5, cgh_cars$gear)
```

Everything above 5.5 will be considered as 5 gears

```
cgh_cars$gear <- ifelse((cgh_cars$gear >= 5.5), 6, cgh_cars$gear)
```

C.6: Noisy Data in “gear” (VARIANT)

… or we use this nice variant:

Round the values with the function “round”

```
cgh_cars$gear <- round(cgh_cars$gear, digits = 0)
```

VT.1: Convert “style” (1/2)

We also need to convert the qualitative variables. “style” is an ordered qualitative variable.

First try to get the values.

```
> unique(cgh_cars$style)
[1] "basic"  "medium" "luxus"
```

Order them and convert them to integers

```
as.integer(ordered(cgh_cars$style, levels= c("basic","medium","luxus")))
```

Eventually assign the values to the “style” column

```
cgh_cars$style <- as.integer(ordered(cgh_cars$style, levels= c("basic","medium","luxus")))
```

VT.1: Convert “style” (2/2)

Check the result:

```
> cgh_cars$style
 [1] 1 2 1 1 1 1 1 1 1 2 2 3 3 3 1 1 1 1 2 1 2 2 2 2 1 1 3 3 3 3 1
```

```
> cgh_cars
       brand        type style      mpg cyl  disp    wt gear carb price
1      Mazda       coupe     1 21.00000   6 160.0 2.620    4    4 33500
2      Mazda     station     2 21.00000   6 160.0 2.875    4    4 32500
3     Datsun    limosine     1 19.73846   4 108.0 2.320    4    1 16000
4     Hornet     station     1 21.40000   6 258.0 3.215    3    1 26700
5     Hornet     station     1 18.70000   8 360.0 3.440    3    2 25350
6    Vailant    limosine     1 18.10000   6 225.0 3.460    3    1 21050
7     Duster    limosine     1 14.30000   8 360.0 3.570    3    4 19650
…
```

VT.2: Convert “type” (1/3)

“type” is is also ordered qualitative variable.

First try to get the values.

```
> unique(cgh_cars$type)
[1] "coupe"       "station"     "limosine"    "convertible" "ship"        "hatchback"
```

Wait a moment: “ship”?

```
> cgh_cars[cgh_cars$type == "ship",]
      brand type style  mpg cyl disp   wt gear carb price
14 Cadillac ship     3 10.4   8  472 5.25    3    4 34200
```

We need to delete these lines with this value first.

```
cgh_cars <- cgh_cars[cgh_cars$type != "ship",]
```

```
> unique(cgh_cars$type)
[1] "coupe"       "station"     "limosine"    "convertible" "hatchback"
```

VT.2: Convert “type” (2/3)

The order of the values needs to reflect how car manufarture structure their prices. For other target variables other orderings may be appropriate.

Order them and convert them to integers

```
as.integer(ordered(cgh_cars$type, levels=
                                c("hatchback","limosine","station","coupe","convertible")))
```

Eventually assign the values to the “type” column

```
cgh_cars$style <- as.integer(ordered(cgh_cars$type, levels=
                                c("hatchback","limosine","station","coupe","convertible")))
```

VT.3: Convert “brand” (1/2)

“brand” is is an unordered qualitative variable (You may have your own order, but there is no shared agreement ☺). Therefore use “dummy_cols”:

```
cgh_cars <- dummy_cols(cgh_cars, select_columns="brand", remove_selected_columns = TRUE)
```

VT.3: Convert “brand” (2/2)

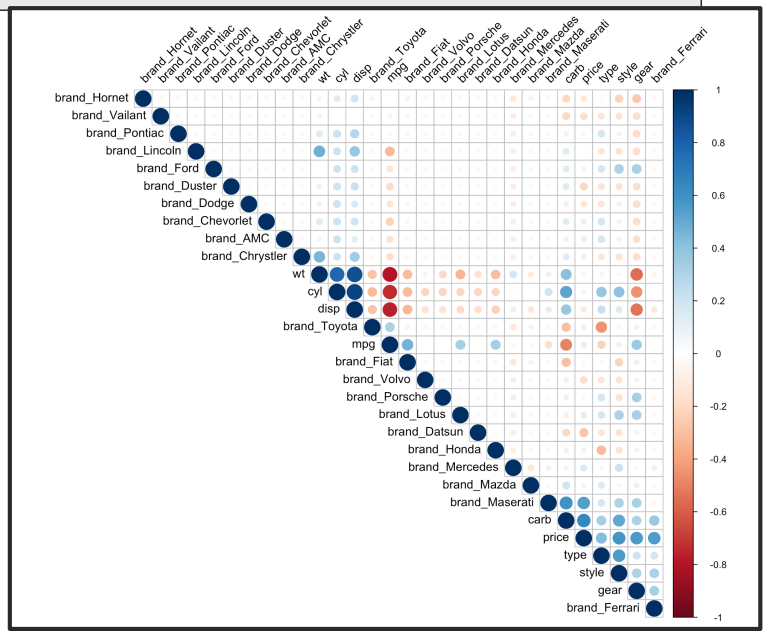
Check the result:

```
> str(cgh_cars)
'data.frame':  30 obs. of  30 variables:
 $ type           : int  4 3 2 3 3 2 2 2 3 2 ...
 $ style          : int  1 2 1 1 1 1 1 1 1 2 ...
 $ mpg            : num  21 21 19.7 21.4 18.7 ...
 $ cyl            : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp           : num  160 160 108 258 360 ...
 $ wt             : num  2.62 2.88 2.32 3.21 3.44 ...
 $ gear           : num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb           : int  4 4 1 1 2 1 4 2 2 4 ...
 $ price          : int  33500 32500 16000 26700 25350 21050 19650 36200 26000 34600 ...
 $ brand_AMC      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Chevorlet: int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Chrystler: int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Datsun   : int  0 0 1 0 0 0 0 0 0 0 ...
 $ brand_Dodge    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Duster   : int  0 0 0 0 0 0 1 0 0 0 ...
 $ brand_Ferrari  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Fiat     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Ford     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Honda    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Hornet   : int  0 0 0 1 1 0 0 0 0 0 ...
 $ brand_Lincoln  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Lotus    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Maserati : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Mazda    : int  1 1 0 0 0 0 0 0 0 0 ...
 $ brand_Mercedes : int  0 0 0 0 0 0 0 1 1 1 ...
 $ brand_Pontiac  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Porsche  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Toyota   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ brand_Vailant  : int  0 0 0 0 0 0 1 0 0 0 ...
 $ brand_Volvo    : int  0 0 0 0 0 0 0 0 0 0 ...
```

A.4 after VT.3: Also analyse the correlations…

"summary" gives you a compact overview

```
> library(corrplot)
> corrplot(cor(cgh_cars), type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)
```



Probably we must be careful with mpg, wt and cyl. But first try to find the linear regression without removing one of them.

Encoded Problems

- 2 types of missing data in **mpg** - NA and 0.0, remove NA and replace by mean/median
- some values in **cyl** are negative, find outliers and change negative values to positive
- some values in **wt** have a missing decimals while some are in pounds instead of kilo pounds, identify rows with discrepancies and fix them on a case by case basis
- Noisy data in **gear**. Apply (simple form of) binning
- (the **disp** column can be scale/normalize the **disp** column)