Multiple linear regression - Case study

Case Study - Predicting medical expenses

Marcus Birkenkrahe

April 17, 2023

README

Figure 1: Bed-ridden wounded, knitting (1918-19), US Nat'l Archives

- This lecture and practice follows the case developed by Lantz' book Machine Learning with R , 3rd edition (2019). In the updated 4th edition (2023), this case has been exchanged by an automobile industry case.

- To code along with the lecture, download `6_regression_practice.org` from GitHub, complete the file and upload it to Canvas by the deadline.

Rationale

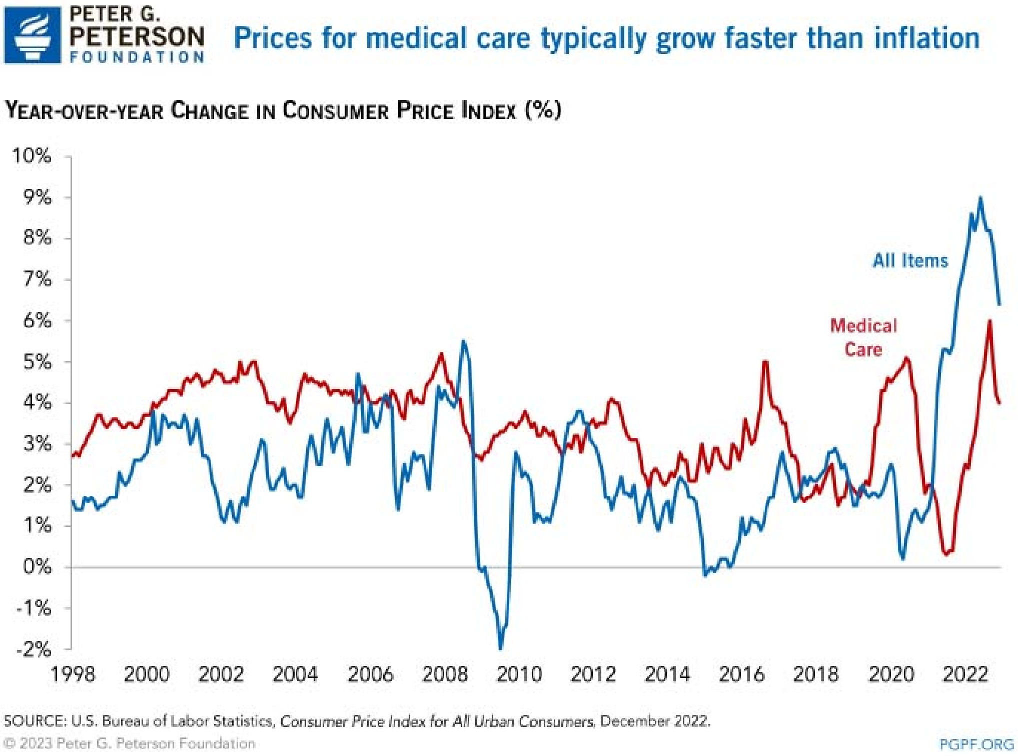
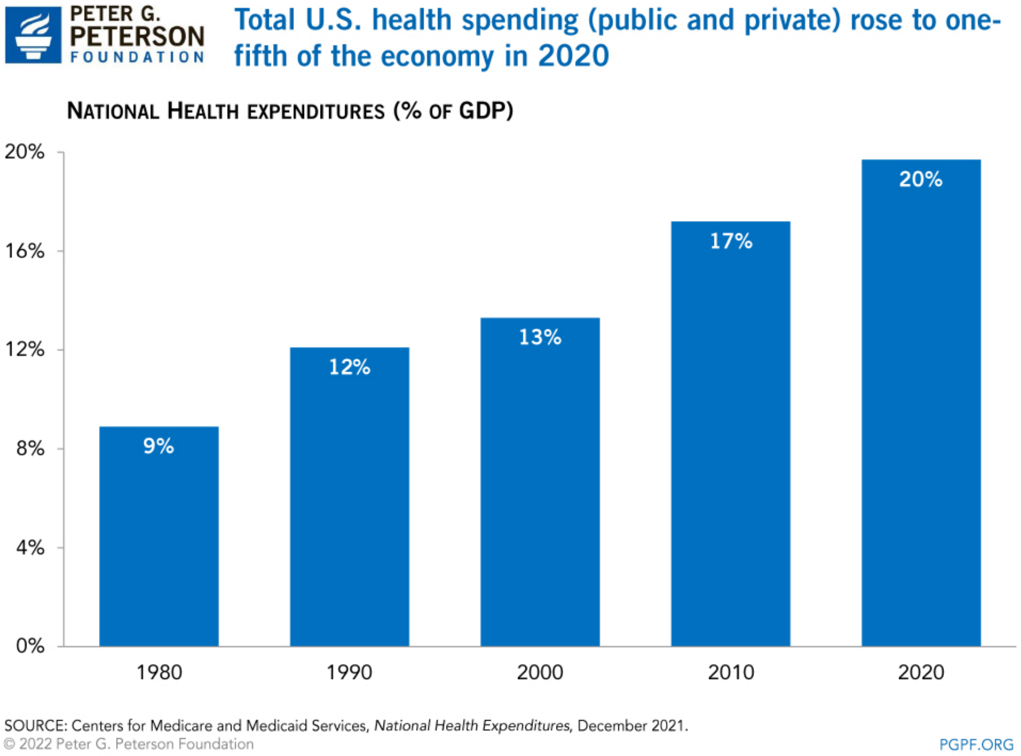


Figure 2: Source: Peter G Peterson foundation (01/30/2023)

- Health insurance companies only make money if they collect more in fees than they spend on medical care to its beneficiaries.
- What do you think are profit margins in other industries?

Profit margins in other industries (sources below):



PETER G. PETERSON FOUNDATION

Total U.S. health spending (public and private) rose to one-fifth of the economy in 2020

NATIONAL HEALTH EXPENDITURES (% OF GDP)

SOURCE: Centers for Medicare and Medicaid Services, *National Health Expenditures*, December 2021.

© 2022 Peter G. Peterson Foundation

PGPF.ORG

Figure 3: Source: Peter G Peterson foundation (01/30/2023)

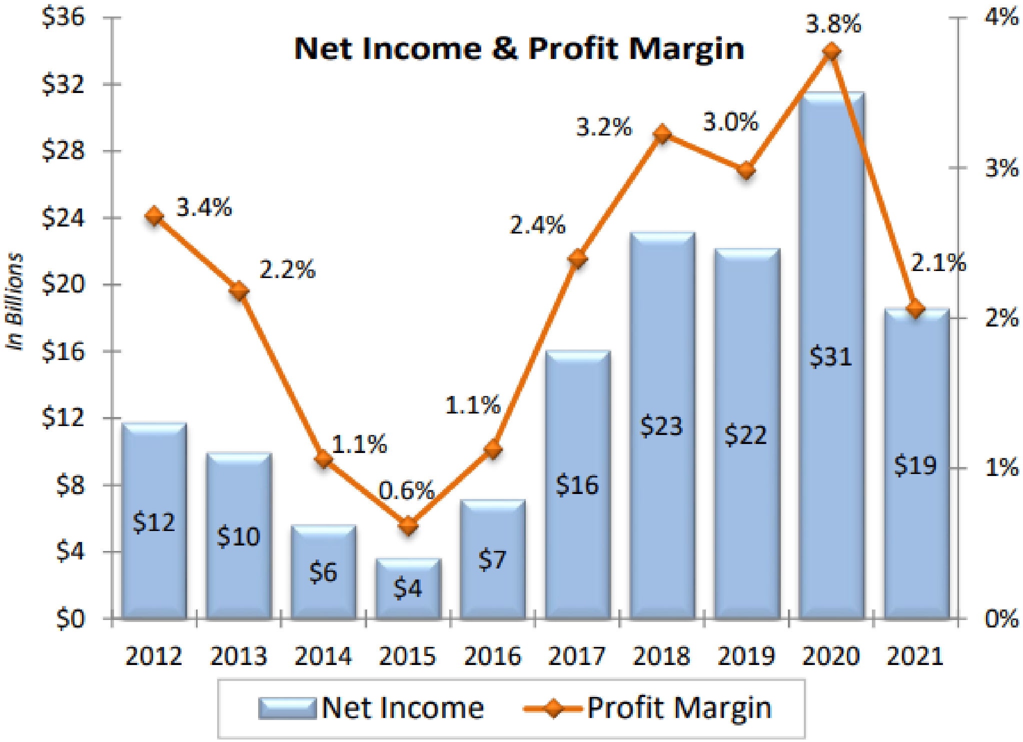


Figure 4: Source: US Health Insurance Industry Analysis report (NAIC)

- aerospace (2022: 8.28%)
    - retail (Amazon 2022: 43%)
    - cars (2020: 7.5%)
    - pharma (2023: 71%)

- Medical expenses are difficult to estimate because the conditions that are the most costly to treat are rare and seem random.
- Analysis goal: use patient data to forecast average medical expense for at-risk segments of the population (like smokers or obese).
- Image source: US Health Insurance Industry Analysis Report 2021

ML workflow

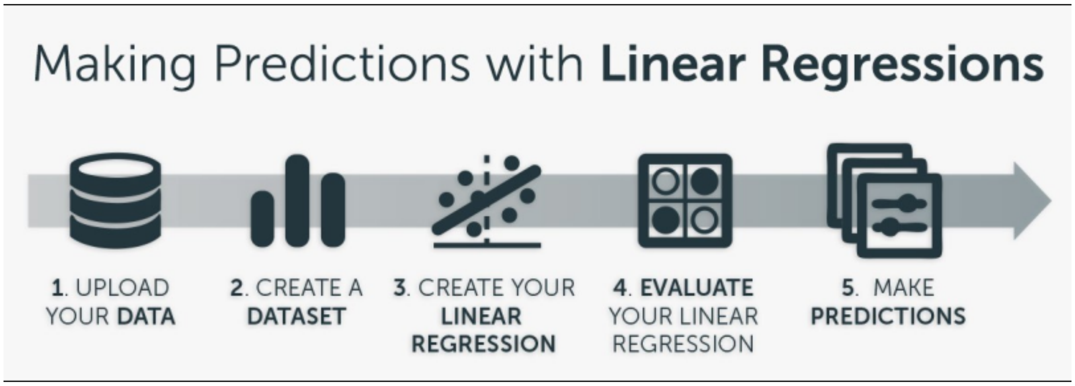

Making Predictions with Linear Regressions
1. UPLOAD YOUR DATA
2. CREATE A DATASET
3. CREATE YOUR LINEAR REGRESSION
4. EVALUATE YOUR LINEAR REGRESSION
5. MAKE PREDICTIONS

Figure 5: Source: blog.bigml.com (2019)

1. Collecting data: US Census bureau (modified)
2. Exploring the data: correlation matrix and scatterplot matrix
3. Training a linear model on the data with `lm`
4. Evaluating model performance with `predict`
5. Improving model performance: nonlinear effects/transformation

Figure 6: US Census Bureau HQ in Maryland

Getting the data

- Fun fact: the firm that designed the USCB HQ also designed the Burj Khalifa (Dubai), the Sears Tower (Chicago) and One World Trade Center (NYC)

- The dataset contains 1,338 examples of beneficiaries enrolled in an insurance plan with patient features and total medical expenses charged to the insurance plan for the calendar year:

  1. `age`: An integer indicating the age of the primary beneficiary (excluding those above 64 years, as they are generally covered by the government).
  2. `sex`: The policy holder's gender: either `male` or `female`.
  3. `bmi`: The body mass index (BMI), which provides a sense of how over or underweight a person is relative to their height. BMI is equal to weight (in kilograms) divided by height (in meters) squared. An ideal BMI is within the range of 18.5 to 24.9.
  4. `children`: An integer indicating the number of children/dependents covered by the insurance plan.

5. `smoker`: A "yes" or "no" categorical variable that indicates whether the insured regularly smokes tobacco.
6. `region`: The beneficiary's place of residence in the US, divided into four geographic regions: `northeast`, `southeast`, `southwest`, or `northwest`.

- Import the data from `insurance.csv` after checking the file online in GitHub: bit.ly/ml_insurance.

- You can check the dataset in Emacs with `M-x eww` followed by the URL. eww is the Emacs World-wide Web browser (good on text/images).

- You can write the text file right away with `C-x C-w` to `insurance.csv`!

- Try `google.com` in eww.

- Import the data with `read.csv` and save them to `insurance`:

```
insurance <- read.csv("../data/insurance.csv")
```

Exploring the data: variables and distribution

- Exploring the data follows the old adage: data structure, statistical summary, overview visualization (numeric data), frequency check (categorical data).

- But this exploration is not an activity for its own sake: especially in the case of linear regression we need to check if the data conform to the minimum criteria (or else we can stop):

  1. **missing** data? (We may have to get a different sample)
  2. **categorical** features? (We may have to transform the data)
  3. **linearity** a reasonable assumption? (May have to resample/rescale)

- Display the dataframe structure:

```
str(insurance)
```

```
'data.frame': 1338 obs. of  7 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : chr  "female" "male" "male" "male" ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...
 $ children: int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : chr  "yes" "no" "no" "no" ...
 $ region  : chr  "southwest" "southeast" "southeast" "northwest" ...
 $ expenses: num  16885 1726 4449 21984 3867 ...
```

- What is the model's dependent variable?

> Answer: `insurance$expenses`, which measure the medical costs each person charged to the insurance plan for the year, and which the insurance company wants to minimize.

- Linear regression does not require a normally distributed dependent variable but the model often fits better when this is true (why?[1])
- To check distribution qualities quickly, we can summarize the stats:

```
summary(insurance$expenses)
summary(insurance)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1122    4740    9382   13270   16640   63770
     age             sex                 bmi            children
Min.   :18.00   Length:1338        Min.   :16.00   Min.   :0.000
1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
Median :39.00   Mode  :character   Median :30.40   Median :1.000
Mean   :39.21                      Mean   :30.67   Mean   :1.095
3rd Qu.:51.00                      3rd Qu.:34.70   3rd Qu.:2.000
Max.   :64.00                      Max.   :53.10   Max.   :5.000
   smoker             region             expenses
Length:1338        Length:1338        Min.   : 1122
Class :character   Class :character   1st Qu.: 4740
Mode  :character   Mode  :character   Median : 9382
      Mean   :13270
      3rd Qu.:16640
      Max.   :63770
```

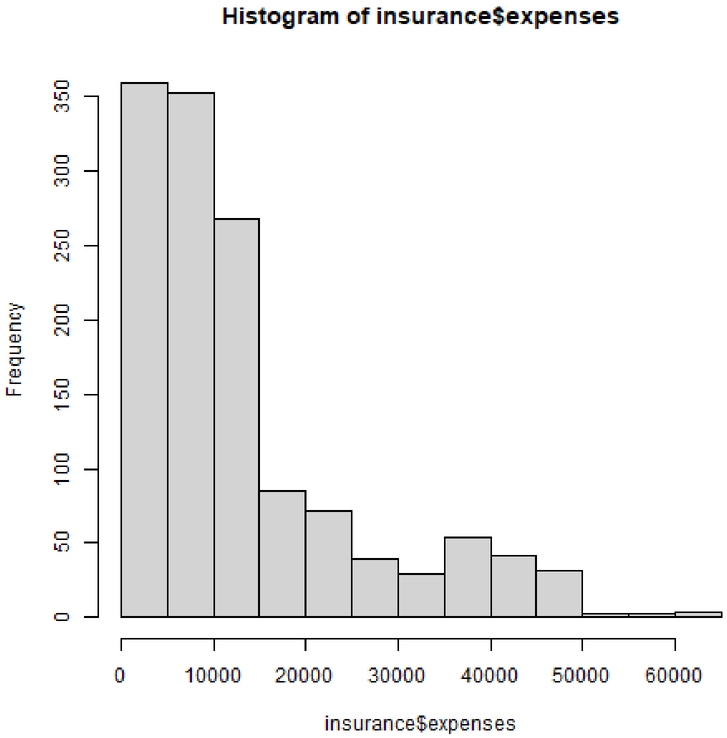
[1] Normal distribution means that standard stats (mean=expected value=0, standard deviation=1 etc.) are known, in other words the distribution is of known spread and centrality. This means we can compare it better with other distributions (in fact, mapping on a normal distribution is a way of ensuring comparability), and deviations stand out more clearly, too.

- What do you observe?

    1. The mean is greater than the median (the middle magnitude is left of the average), which means the distribution is **right** skewed[2].
    2. The spread is significant (minimum vs. maximum values).

- We visualize the distribution (what's the best graph for that?):

```
## Visualize numerical distributions = frequencies with a histogram
hist(insurance$expenses)
```



[2] The **skewedness** highlights the opposite of the maximum of the points - a left/right leaning distribution is skewed to the right/left, because the outlying points cause the problem in terms of analysis: they are harder to distinguish and kind of "fall off the end". Transformations will affect them more strongly.

- The graph shows that the majority of people have annual medical expenses below US$15,000. Knowing the graphs structural weakness ahead of time will help us improve the linear model later on.

Exploring the data: correlation matrix

- The **correlation matrix** gives an overview of how the variables relate to one another: given a set of variables, it provides a correlation for each pairwise relationship.

- To create a correlation matrix, use the `cor` command - take a look at its arguments first:

  ```
  args(cor)
  ```

  ```
  function (x, y = NULL, use = "everything", method = c("pearson",
      "kendall", "spearman"))
  NULL
  ```

- Let's build this up slowly: the default for `y` is only relevant if `x` is a matrix: how is the dependent variable correlated **with itself**?

  ```
  ## Just the dependent variable - formatted as matrix
  x <- as.matrix(insurance$expenses)
  head(x)
  cor(x)
  ```

  ```
           [,1]
  [1,] 16884.92
  [2,]  1725.55
  [3,]  4449.46
  [4,] 21984.47
  [5,]  3866.86
  [6,]  3756.62
       [,1]
  [1,]    1
  ```

- This makes sense because:

  ```
  var(x,x)/(sd(x)*sd(x)) ## sd^2 = var
  ```

```
     [,1]
[1,]    1
```

- Now for all `numeric` variables:

```
str(insurance)
ins_num <- c("age","bmi","children","expenses")
cor(insurance[ins_num]) # only numerical features
```

```
'data.frame': 1338 obs. of  7 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : chr  "female" "male" "male" "male" ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...
 $ children: int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : chr  "yes" "no" "no" "no" ...
 $ region  : chr  "southwest" "southeast" "southeast" "northwest" ...
 $ expenses: num  16885 1726 4449 21984 3867 ...
       age        bmi   children   expenses
age      1.0000000 0.10934101 0.04246900 0.29900819
bmi      0.1093410 1.00000000 0.01264471 0.19857626
children 0.0424690 0.01264471 1.00000000 0.06799823
expenses 0.2990082 0.19857626 0.06799823 1.00000000
```

- What do we learn?

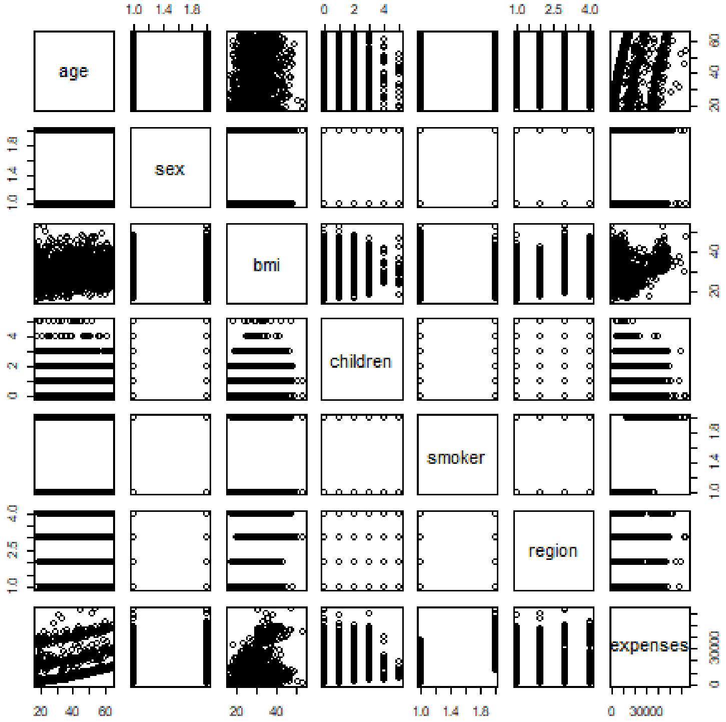
    1. the diagonal of the correlation matrix is always 1 (a variable is always perfectly correlated with itself: `cor(x,x) = 1`).
    2. the matrix transpose is identical to itself (correlation is symmetrical: `cor(x,y) = cor(y,x)`).
    3. None of the correlations is strong (i.e. we need them all).
    4. `age` and `bmi` are weakly positively correlated: as you age, your BMI slightly increases.
    5. Expenses go up with age, body mass, and number of children.

Exploring the data: scatterplot matrix

- A *scatterplot matrix* or *pair plot* shows the relationship of each variable with every other as a graph.

- You can feed the whole dataframe into the generic `plot` function:

  ```
  plot(insurance)
  ```

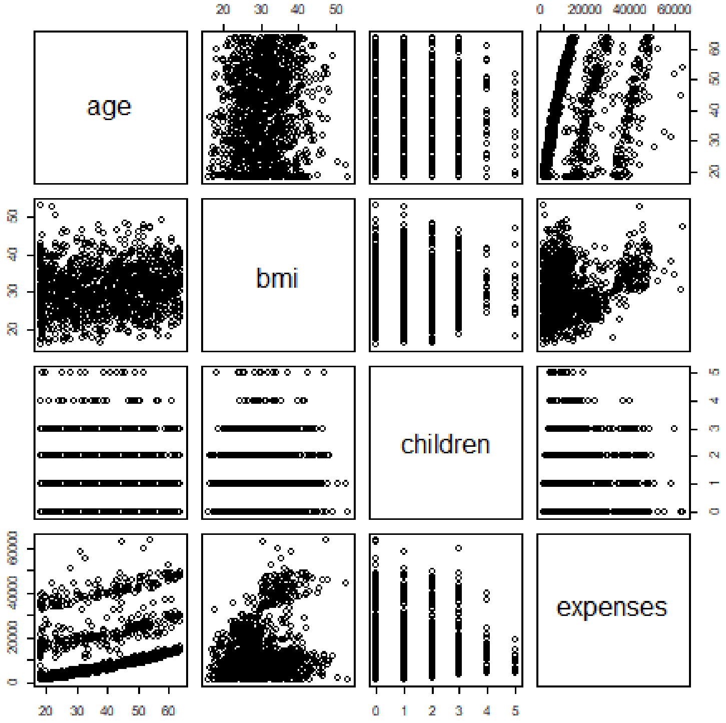


- However, `plot` does not distinguish between numeric and categorical variables, and a scatterplot is meaningless for the latter.

- An alternative is `graphics::pairs`[3]:

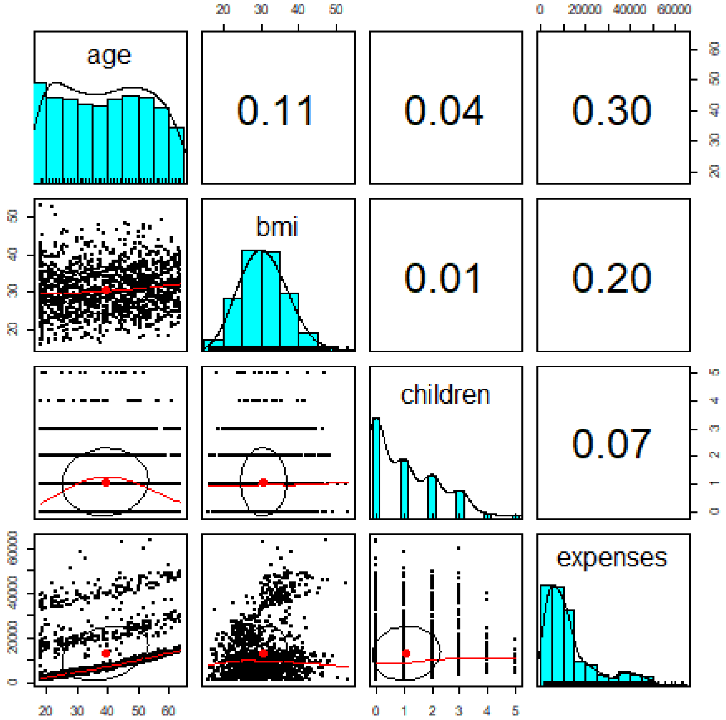
  ```
  pairs(insurance[ins_num]) ## ins_num <- c("age","children","bmi","expenses")
  ```

[3] The result is the same as `plot(insurance[ins_num])` but `pairs` offers different customization options than the generic `plot` - see `help(pairs)`.



- The intersection of each row and column holds the scatterplot of the variables indicated by the row and column pair: e.g. the plot in the 2nd row and 2nd column shows `age ~ bmi` or "age" as a function of "bmi" - its transpose value shows `bmi ~ age`.
- Do you notice any patterns in these plots?
  1. Visible nearly straight lines in `age ~ expenses`
  2. Two point clusters in `bmi ~ expenses`
  3. Invisible structure in the `age ~ bmi` plot
- The `pairs.panels` function in the `psych` package contains more information:

```
library(psych)
pairs.panels(insurance[ins_num])
```



- What do you see?

  1. The scatterplots above the diagonal are now a correlation matrix
  2. The diagonal shows histograms for the feature distributions with a density estimate (smoothing) to more clearly show profile.
  3. Each scatterplot shows a *correlation ellipse* indicating spread: the more it is stretched, the stronger the correlation - e.g. `children ~ bmi` is almost round indicating that the number of children is largely independent of the BMI (and vice versa) = 0.01.
  4. The correlation ellipse for `expenses ~ age` is much more stretched: these features are more correlated = 0.30.
  5. The red dot at the center of the ellipsis is the mean value.

6. The red curve drawn on the scatterplot is a `loess curve`: the curves for `children ~ age` peaks around middle age: the oldest and youngest people in the sample have fewer children.

- The `age ~ children` trend is non-linear and cannot be seen in the correlations! (Unlike e.g. the `age ~ bmi` loess curve.)

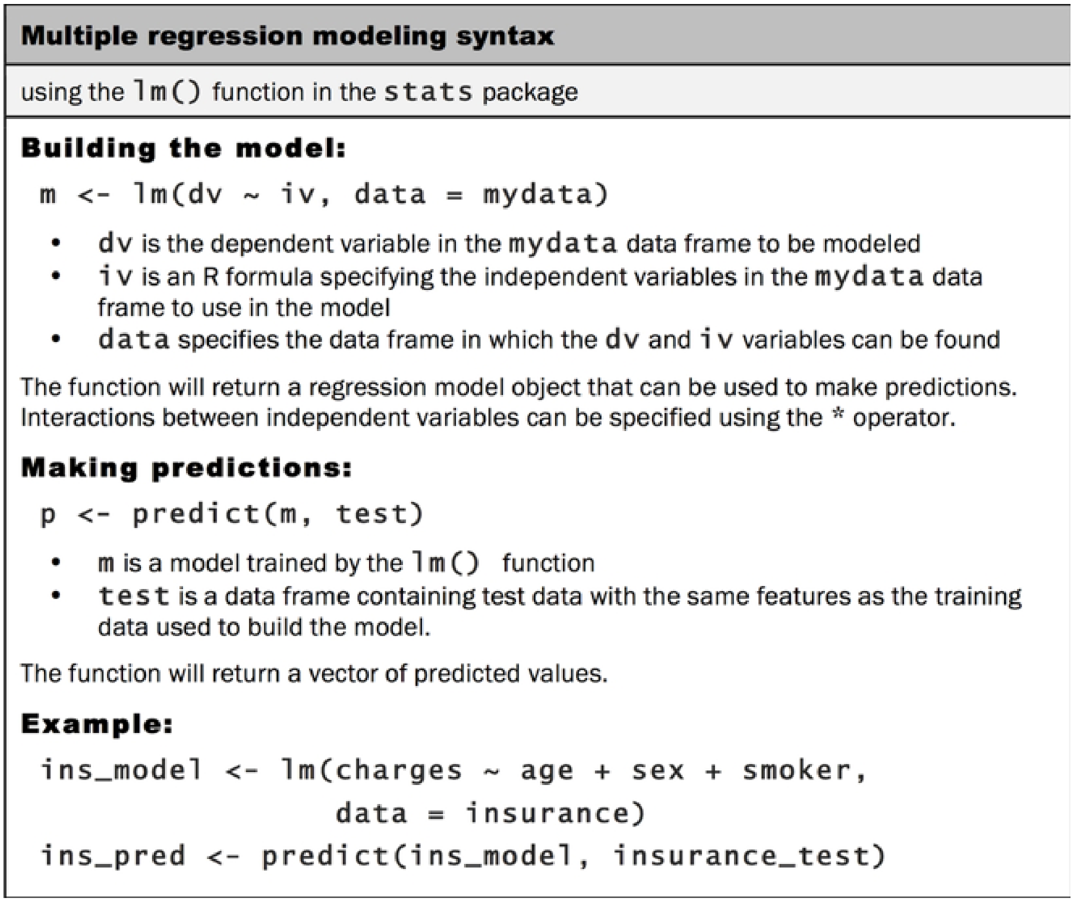
IN PROGRESS Training a model on the data

- We use the generic `lm` function from `stats`- check arguments:

```
args(lm)
environment(lm)
```

```
function (formula, data, subset, weights, na.action, method = "qr",
    model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE,
    contrasts = NULL, offset, ...)
NULL
<environment: namespace:stats>
```

- Here's a syntax overview (Lantz, 2019):



Multiple regression modeling syntax

using the `lm()` function in the `stats` package

Building the model:

```
m <- lm(dv ~ iv, data = mydata)
```

- `dv` is the dependent variable in the `mydata` data frame to be modeled
- `iv` is an R formula specifying the independent variables in the `mydata` data frame to use in the model
- `data` specifies the data frame in which the `dv` and `iv` variables can be found

The function will return a regression model object that can be used to make predictions. Interactions between independent variables can be specified using the * operator.

Making predictions:

```
p <- predict(m, test)
```

- `m` is a model trained by the `lm()` function
- `test` is a data frame containing test data with the same features as the training data used to build the model.

The function will return a vector of predicted values.

Example:

```
ins_model <- lm(charges ~ age + sex + smoker,
                data = insurance)
ins_pred <- predict(ins_model, insurance_test)
```

- Uses the "formula" syntax - the independent variables can **all** be included with the `.` operator: `lm(dep ˜ ., data)` or individually with the `+` operator.

- Just like seen in the `glm` example (logistic regression), you can include *interactions* between independent variables with the `*` operator to model the combined effect of two or more features.

- The following model relates the six independent variables to the total medical `expenses`:

  ```
  ins_model <- lm(expenses ˜ . ,data = insurance)
  ```

- To see the estimated β coefficients, print the model:

```
ins_model
```

```
Call:
lm(formula = expenses ~ ., data = insurance)

Coefficients:
    (Intercept)              age          sexmale              bmi         childre
       -11941.6            256.8           -131.4            339.3            475
      smokeryes  regionnorthwest  regionsoutheast  regionsouthwest
        23847.5           -352.8          -1035.6           -959.3
```

- The `Intercept` is the predicted value when the independent variables are zero (not realistic since living persons have BMI > 0, age > 0).
- The β coefficients indicate the estimated increase (slope) in expenses for an increase of one unit in each of the features, assuming all other values are held *constant*.
- For example: for each additional year of `age`, we expect an average of `256.8` expense increase per year.
- The `lm` function automatically dummy-codes each `factor` type variable included, like `sex`, `smoker` and `region` (split in four dummy variables).
- When adding dummy variables, one category is always left out as a reference category (e.g. `sex=female`, `region=northeast`): e.g. males have `$131.4` less medical expenses than females per year relatives to females[4].
- Which `region` has the highest medical expenses?

  > The reference group - `northeast`, because all other values are negative.

- In summary: old age, smoking and obesity can be linked to additional health issues, and additional family members may result in an increase. But how well is this model fitting the data?

[4] In R, the first `level` is taken as reference. You can use `relevel` to change this.

Evaluating model performance

- Why don’t we use a confusion matrix?

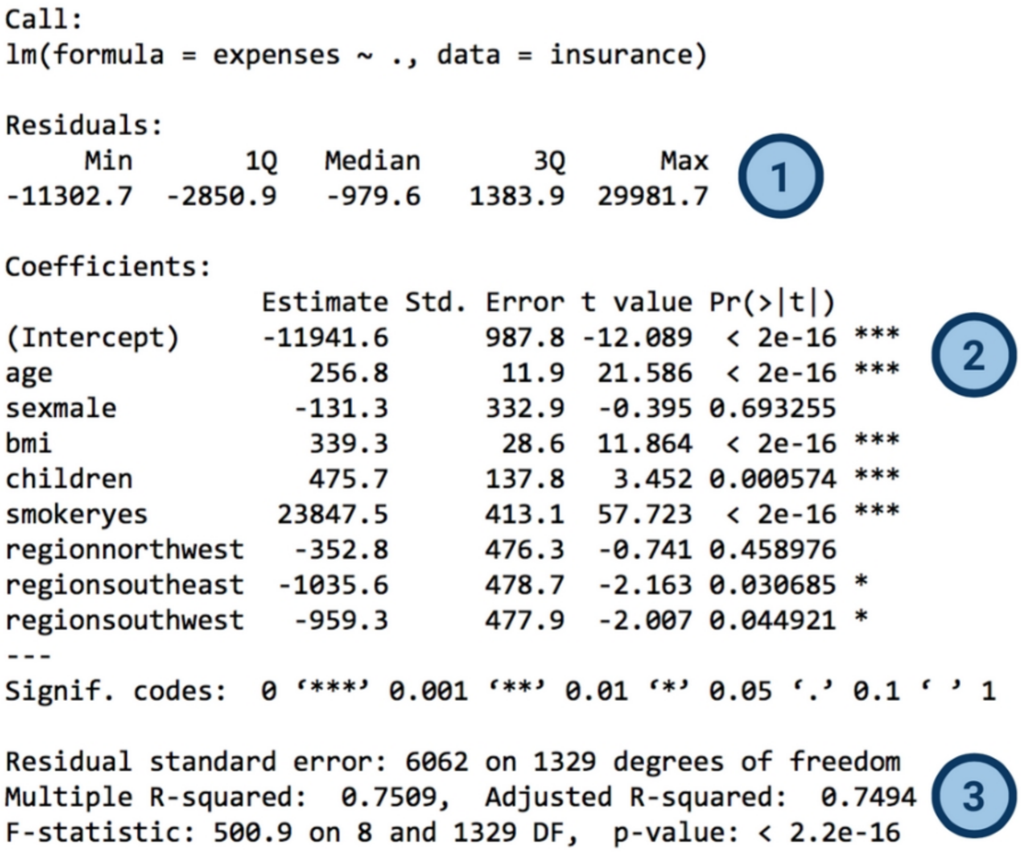
  > Answer: the confusion matrix is for classification of categorical variables, not continuous numeric variables.

- To evaluate model performance, we can use `summary`:

  ```
  summary(ins_model)



  Call:
  lm(formula = expenses ~ ., data = insurance)

  Residuals:
       Min       1Q   Median       3Q      Max
  -11302.7  -2850.9   -979.6   1383.9  29981.7

  Coefficients:
  Estimate Std. Error t value Pr(>|t|)
  (Intercept)     -11941.6      987.8 -12.089  < 2e-16 ***
  age                256.8       11.9  21.586  < 2e-16 ***
  sexmale           -131.3      332.9  -0.395 0.693255
  bmi                339.3       28.6  11.864  < 2e-16 ***
  children           475.7      137.8   3.452 0.000574 ***
  smokeryes        23847.5      413.1  57.723  < 2e-16 ***
  regionnorthwest   -352.8      476.3  -0.741 0.458976
  regionsoutheast  -1035.6      478.7  -2.163 0.030685 *
  regionsouthwest   -959.3      477.9  -2.007 0.044921 *
  ---
  Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

  Residual standard error: 6062 on 1329 degrees of freedom
  Multiple R-squared:  0.7509,Adjusted R-squared:  0.7494
  F-statistic: 500.9 on 8 and 1329 DF,  p-value: < 2.2e-16
  ```

- The *summary* explained:

  1. The **Residuals** give summary statistics: a residual is the true value minus the predicted value, the maximum error `29981.7`



```
Call:
lm(formula = expenses ~ ., data = insurance)

Residuals:
     Min       1Q   Median       3Q      Max
-11302.7  -2850.9   -979.6   1383.9  29981.7

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -11941.6      987.8 -12.089  < 2e-16 ***
age                256.8       11.9  21.586  < 2e-16 ***
sexmale           -131.3      332.9  -0.395 0.693255
bmi                339.3       28.6  11.864  < 2e-16 ***
children           475.7      137.8   3.452 0.000574 ***
smokeryes        23847.5      413.1  57.723  < 2e-16 ***
regionnorthwest   -352.8      476.3  -0.741 0.458976
regionsoutheast  -1035.6      478.7  -2.163 0.030685 *
regionsouthwest   -959.3      477.9  -2.007 0.044921 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6062 on 1329 degrees of freedom
Multiple R-squared:  0.7509,  Adjusted R-squared:  0.7494
F-statistic: 500.9 on 8 and 1329 DF,  p-value: < 2.2e-16
```

Figure 7: Evaluation of the regression model with summary()

suggests that the model underperformed and under-predicted expenses by \$30,000 for at least one observation.

50% of all errors fall between the 3rd and the 1st quartile, i.e. the majority of the predictions were between \$2,850 over and \$1,380 under the true value.

2. For each coefficient, the `p-value` in the last column estimates statistical significance: small values suggest that the coeffcient is very unlikely to be zero (feature is related to the dependent variable). The stars `***` represent the significance level set beforehand. Few such terms would be cause for concern: the features wouldn't be very predictive of the outcome.
3. The *multiple R-squared* value (also called 'coefficient of determination') is a measure of how well the model as a whole explains the values of the dependent variable: the closer to 1 the better. A value of 0.75 means that the model explains 75% of the observed variation in the dependent variable.[5]

- Given these three performance indicators - residual error, p-value and multiple R-squared value - the model performs fairly well. The large error maximum is worrying but consistent with what we know of medical expense data.

Excursion: z value and Pr(>|z|)

- The z value is the number of standard deviations a value is away from the mean.
- The `Pr(>|z|)` column represents the *p-value* associated with the value in the z column.
- If the p-value is less than a certain significance level (for example $\alpha = 0.05$), then this indicates that the predictor has a statistically significant relationship with the response variable in the model.
- *Statistical significance* means that a prediction is not the result of chance but can instead be attributed to a specific cause.
- Another way of saying it: α is the probability of the prediction rejecting the null hypothesis (, and the p-value of a result is the probability of

[5] The Adjusted R-Squared value corrects for models with many features.

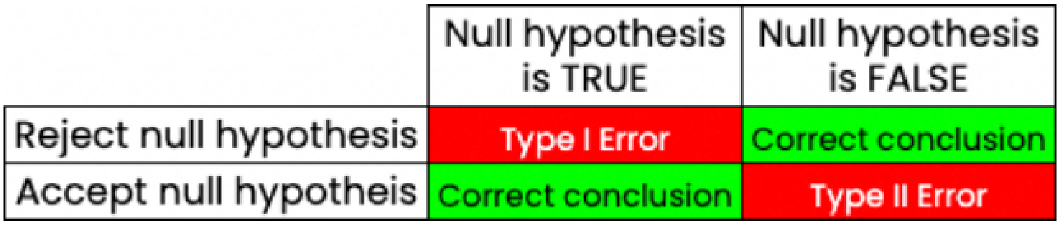
obtaining a result at least as extreme, given that the null hypothesis is true.

- The *null hypothesis* is true if there is no relationship between the predictor and the target variable, i.e. changes in the predictors only lead to random changes in the target variable but not because the two are meaningfully correlated.

- So the first thing to do when discovering a correlation is to check statistical significance to make sure that the discovery is not the result of random fluctuations in the sample.

- The α must be set before evaluation - if it is tampered with when the result does not satisfy one's prejudices, this is called "p-hacking", which is very widespread e.g. in clinical trials (Adda et al, 2020): insights are presented as statistically significant even though they're not.

- There is a kind of confusion matrix here, too:



	Null hypothesis is TRUE	Null hypothesis is FALSE
Reject null hypothesis	Type I Error	Correct conclusion
Accept null hypotheis	Correct conclusion	Type II Error

Figure 8: Type I and Type II statistical errors

1. Type I errors are false positives
2. Type II errors are false negatives

- What would be the null hypothesis for our prediction of insurance expenses?

  - This question only makes sense with regard to a particular feature - e.g. our null hypothesis could be "smoking does not lead to increased medical expenses."
  - Type I error: we find "Smoking increases expenses" (while it actually does not).
  - Type II error: we find "smoking does NOT increase expenses" (while it actually does).

Improving model performance

Regression typically leaves feature selection to the user - subject matter knowledge (on how a feature is related to the outcome) is important! We explore three alterations of the model:

- Adding non-linear relationships among independent variables
- Transform numeric independent variables to binary indicators
- Adding interaction effects between independent variables

Adding non-linear relationships

- To account for a non-linear relationship, we can add a higher order term treating the model as a polynomial:

$$y = \alpha + \beta_1 x + \beta_2 x^2$$

Figure 9: Adding a higher order term to the regression equation

- The additional β coefficient will capture the effect of the xˆ{2} term.
- Looking at the correlation ellipsis and the loess curve which revealed a slight non-linearity, `age` might be a good candidate.
- In R, we simply create a new variable `age2` - this will add a feature vector and another column to our β coefficients matrix:

```
insurance$age2 <- insurance$age^2
str(insurance)

'data.frame': 1338 obs. of  8 variables:
 $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
 $ sex     : chr  "female" "male" "male" "male" ...
 $ bmi     : num  27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...
 $ children: int  0 1 3 0 0 0 1 3 2 0 ...
 $ smoker  : chr  "yes" "no" "no" "no" ...
 $ region  : chr  "southwest" "southeast" "southeast" "northwest" ...
 $ expenses: num  16885 1726 4449 21984 3867 ...
 $ age2    : num  361 324 784 1089 1024 ...
```

- When we build the model, we add both age variables to the formula, as in `expenses ~ age + age2`, allowing `lm` to separate the terms.

TODO Converting numeric variable to binary indicator

TODO Adding interaction effects

Predictions with the improved regression model

TODO Glossary of code

COMMAND	MEANING

TODO Summary

TODO Solutions

References

- Adda et al (2020). P-hacking in clinical trials and how incentives shape the distribution of results across phases. In: Proc Nat Acad Sci 117(24):13386-13392. URL: doi.org/10.1073/pnas.1919906117

- Data: PacktPublishing (2019). Machine learning with R (3e). URL: github.com.

- Lantz (2019). Machine learning with R (3e). Packt. URL: packtpub.com.

- R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.