Week 3: Logistic Regression

MATH-516 Applied Statistics

Tomas Masak

March 6th 2023

Section 1

Linear Model for Binary Response?

Linear Model

- data for linear model: $(Y_1, X_1^\top)^\top, \ldots, (Y_N, X_N^\top)^\top$ where
  - $X_n \in \mathbb{R}^q$ are explanatory variables
  - Y_n are responses
- what if the **responses** $Y_n \in \{0, 1\}$ are **binary**?

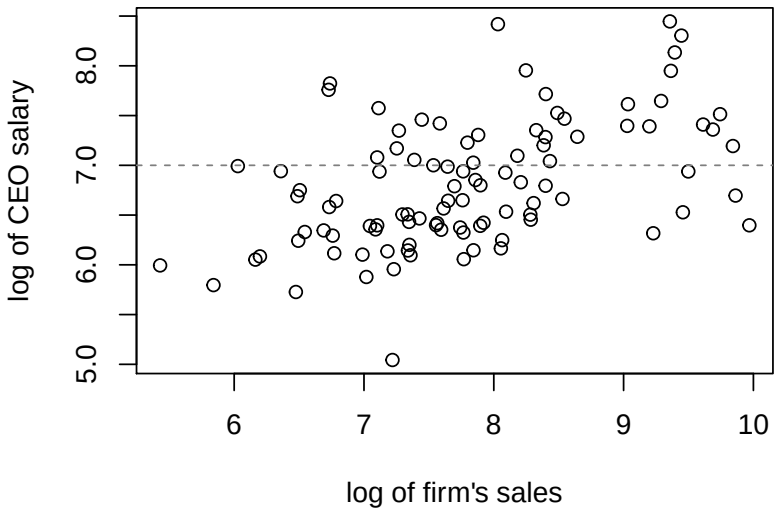
Well, least squares still work...

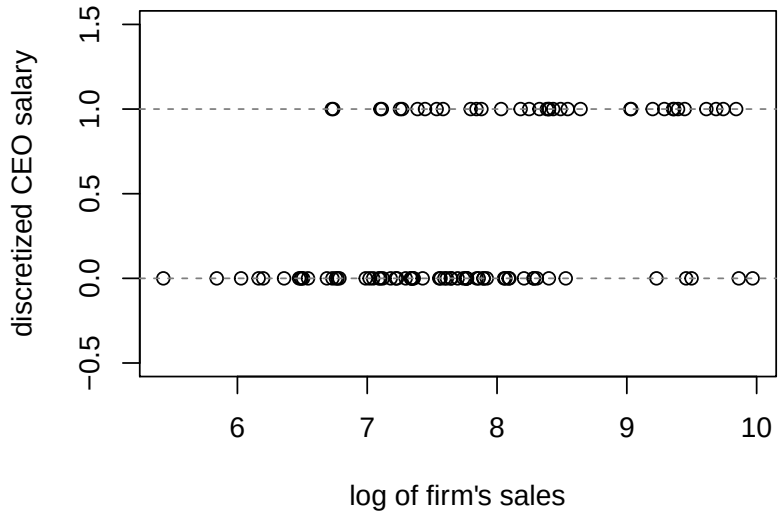
- *Note*: I am sure many people already know
  - that a linear model is for regression but a 0-1 response is a classification problem
  - about logistic regression
- but bear with me...

Example: CEO Salaries

Consider a single predictor and create a 0-1 response

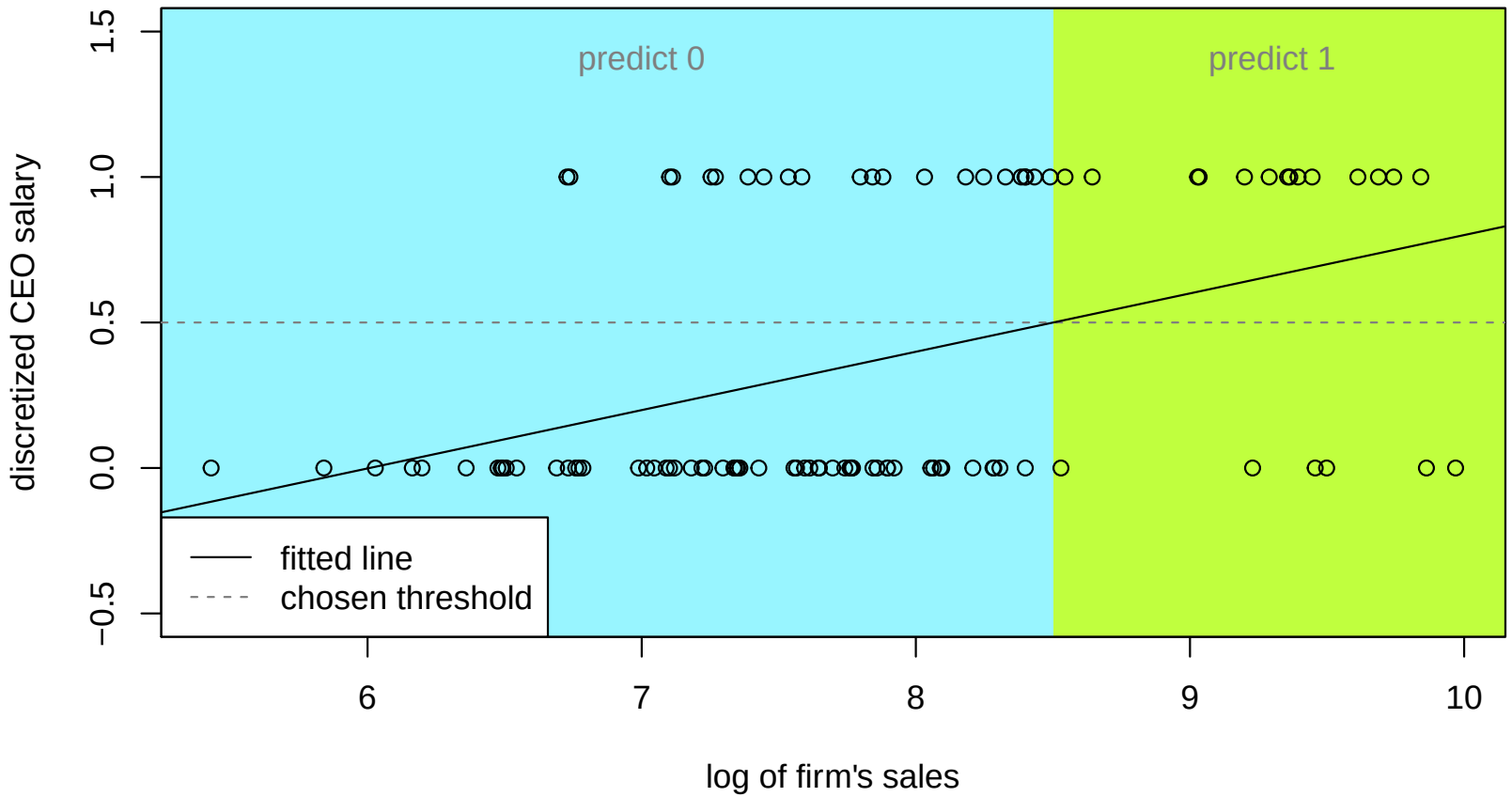
```
library(tidyverse)
Data <- read.csv("../Project-0/CEO_compensations.csv")
names(Data) <- tolower(names(Data))
Data <- Data %>% select(comp,sales) %>%
  mutate(comp=as.numeric(log(comp)>7),sales=log(sales))
```





Linear Model and Prediction

```
m1 <- lm(comp~sales, data=Data)
ythreshold <- 0.5
xthreshold <- (ythreshold-coef(m1)[1])/coef(m1)[2]
```



Confusion Matrix and ROC Curve

For a fixed threshold, we obtain a **confusion matrix**:

		Prediction 0	Prediction 1	Total
Truth	1	#FN	#TP	N_1
	0	#TN	#FP	N_0
	Total	#N	#P	N

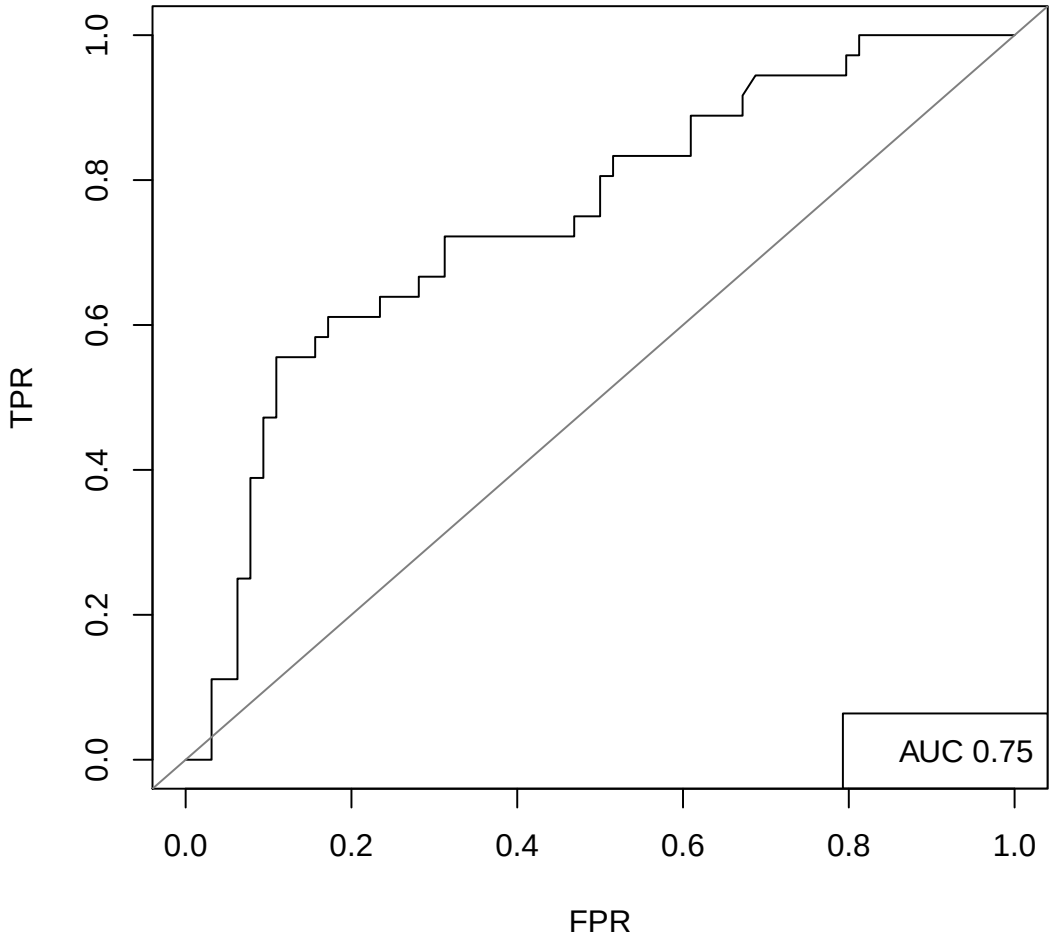
- N_0 and N_1 are fixed, the threshold just pours in the left-right direction
  - for a fixed threshold performance summarized e.g. by #TP and #FP

ROC curve: plot TPR=#TP/N_1 against FPR=#FP/N_0

- allows for comparison of classifiers across all possible thresholds
  - a curve on a 0-1 square, starts at (0,0), ends at (1,1)
  - higher ROC $\equiv$ better classifier
  - a coin-toss classifier $\equiv$ ROC curve is identity
  - should be estimated out-of-sample or cross-validated
  - area under curve (**AUC**) commonly reported for numerical purposes
  - reads right-to-left with increasing threshold

Example: CEO Salaries

In-sample ROC curve for our linear model classifier:



Problems with Least Squares for Discrete Responses

Everything naturally generalizes to more regressors.

So least squares give us something, but:

- we cannot trust model-submodel tests!
  - the simple linear model $Y_n = X_n^\top \beta + \epsilon_n$ with $\epsilon_n \sim (0, \sigma^2)$ and $\epsilon_n \perp\!\!\!\perp X_n$ cannot hold for binary data
- fitting an unbounded line to the binary data is awkward
  - poor interpretation of the coefficients
  - residuals can be huge (artificially)
- uncertainty quantification does not work
- model fit measures such as R-squared or AIC are not informative
- overall we have little guidance in model building and model comparison

Section 2

Logistic Regression

Latent Model

- data for linear model: $(Y_1, X_1^\top)^\top, \ldots, (Y_N, X_N^\top)^\top$ where
  - $X_n \in \mathbb{R}^q$ are explanatory variables
  - Y_n are responses
- what if the responses $Y_n \in \{0, 1\}$ are binary?
- maybe they arose by discretization of latent variables Z_n that follow a linear model...
  - $Z_n = X_n^\top \gamma + \sigma \epsilon_n$, where
    - $\gamma \in \mathbb{R}^p$ and $\sigma \in (0, \infty)$
    - $\epsilon_n \overset{\perp\!\!\!\perp}{\sim} F$ for F standardized and symmetric
  - $Y_n = \mathbb{I}_{[Z_n > 0]}$ (or some other constant)
- it must be $Y_n \overset{\perp\!\!\!\perp}{\sim} Bern(\pi_n)$ and the model for the expectation is:

$$\pi_n = P(Z_n > 0) = P(\epsilon > -X_n^\top \gamma/\sigma) = 1 - F(-X_n^\top \gamma/\sigma) = F(X_n^\top \underbrace{\gamma/\sigma}_{=:\beta})$$

Latent Model (cont.)

$$\pi_n = F(X_n^\top \beta)$$

- $\pi_n \in [0, 1]$ and F takes the linear predictor $X_n^\top \beta$ to its range

How to estimate β from the binary responses we have? Maximum likelihood:

$$L(\beta) = \Pi_{n=1}^{N} \pi_n^{Y_n} (1 - \pi_n)^{1 - Y_n} \qquad \Big| \log(\cdot)$$

$$\ell(\beta) = \sum_{n=1}^{N} \left[Y_n \log F(X_n^\top \beta) + (1 - Y_n) \log F(-X_n^\top \beta) \right]$$

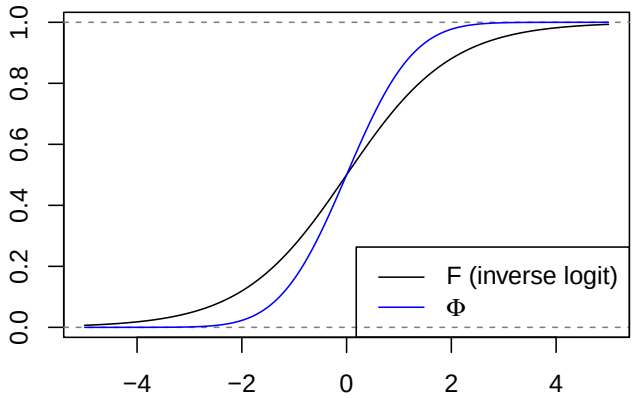
- maximize $\ell(\beta)$ numerically to obtain $\widehat{\beta}$
- we need to choose F...

Link Function

- we can take $F = \Phi$, i.e. the distribution function of a Gaussian, but another choice is customary:

$$F(x) = \frac{e^x}{1 + e^x} \quad \Rightarrow \quad X_n^\top \beta = F^{-1}(\pi_n) = \log\left(\frac{\pi_n}{1 - \pi_n}\right)$$

- . . . log-odds (F^{-1} is called "logit")
  - odds $\pi_n/(1 - \pi_n) \in (0, \infty)$, hence
  - regressors have a multiplicative effect on the odds of success
- the choice of link function F matters more for prediction than inference



Next James Bond Odds (just for fun)

Let's say that one of three following actors is known to be cast as the next James Bond, and the bookies have put out the following odds:



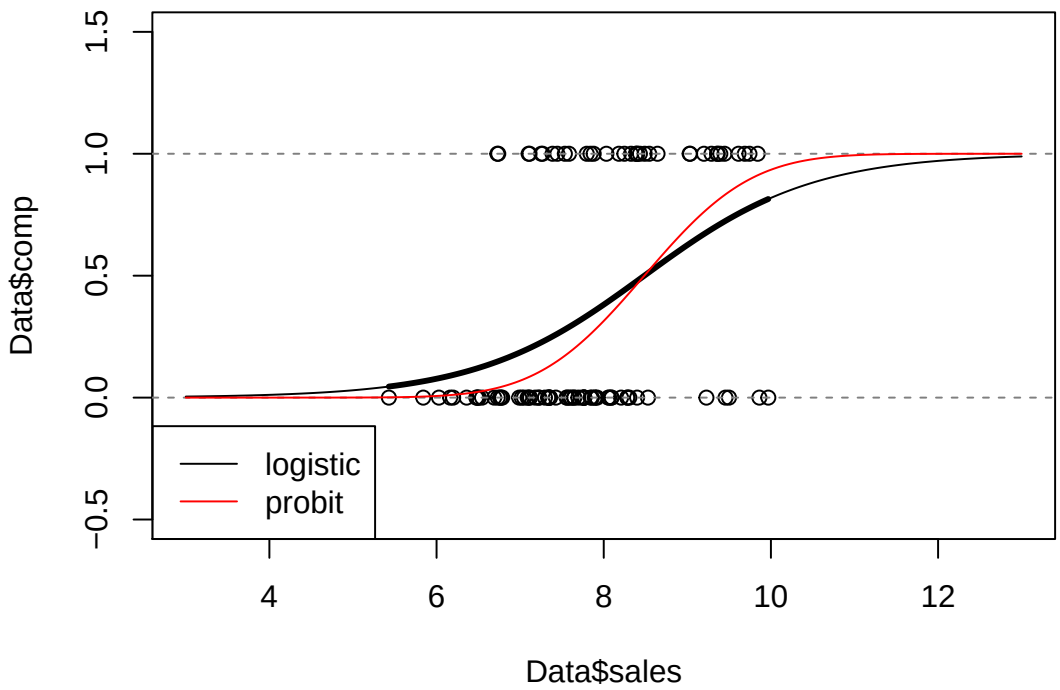
Actor	Odds of Winning	Probability of Winning
Henry Cavill	1	50 %
Aaron-Taylor Johnson	3/7	30 %
Idris Elba	1/4	20 %

- odds = prob/(1-prob), i.e. prob = odds/(1+odds)
- in bookkeeping:
  - odds are usually given as (a multiple of, strangely written) odds to loose
    - e.g. AT's odds would be written as 7-3
  - corresponding probabilities typically do not sum up to 1

Example: CEO Salaries

Start again with only a single regressor:

```
gm1 <- glm(comp~sales, data=Data, family="binomial")
plot(Data$sales,Data$comp,ylim=c(-0.5,1.5), xlim=c(3,13))
abline(h=c(0,1), lty=2, col="gray50")
x <- seq(2, 14, length=100)
points(x, plogis(coef(gm1)[1]+x*coef(gm1)[2]), type="l")
gm2 <- glm(comp~sales, data=Data, family=binomial(link="probit"))
```



Logistic Regression vs. Least Squares

What we get is not that different, but with logistic regression we have a valid likelihood and hence

- we can use model-submodel test
  - only asymptotically valid $\Rightarrow$ likelihood ratio instead of an F-test
- we have a measure of how well models fit
  - R-squared replaced by residual deviance
  - we can use penalized likelihood criteria such as AIC, AIC_C, BIC, etc.

$\Rightarrow$ model building is similar in flavor to linear regression

However, residual plots are confusing with discrete response, because

- residuals themselves are discrete (equal to $\widehat{Y}_n$ or $1 - \widehat{Y}_n$)
- FWL theorem does not hold
  - still, we kind of act like if it did, and it mostly works

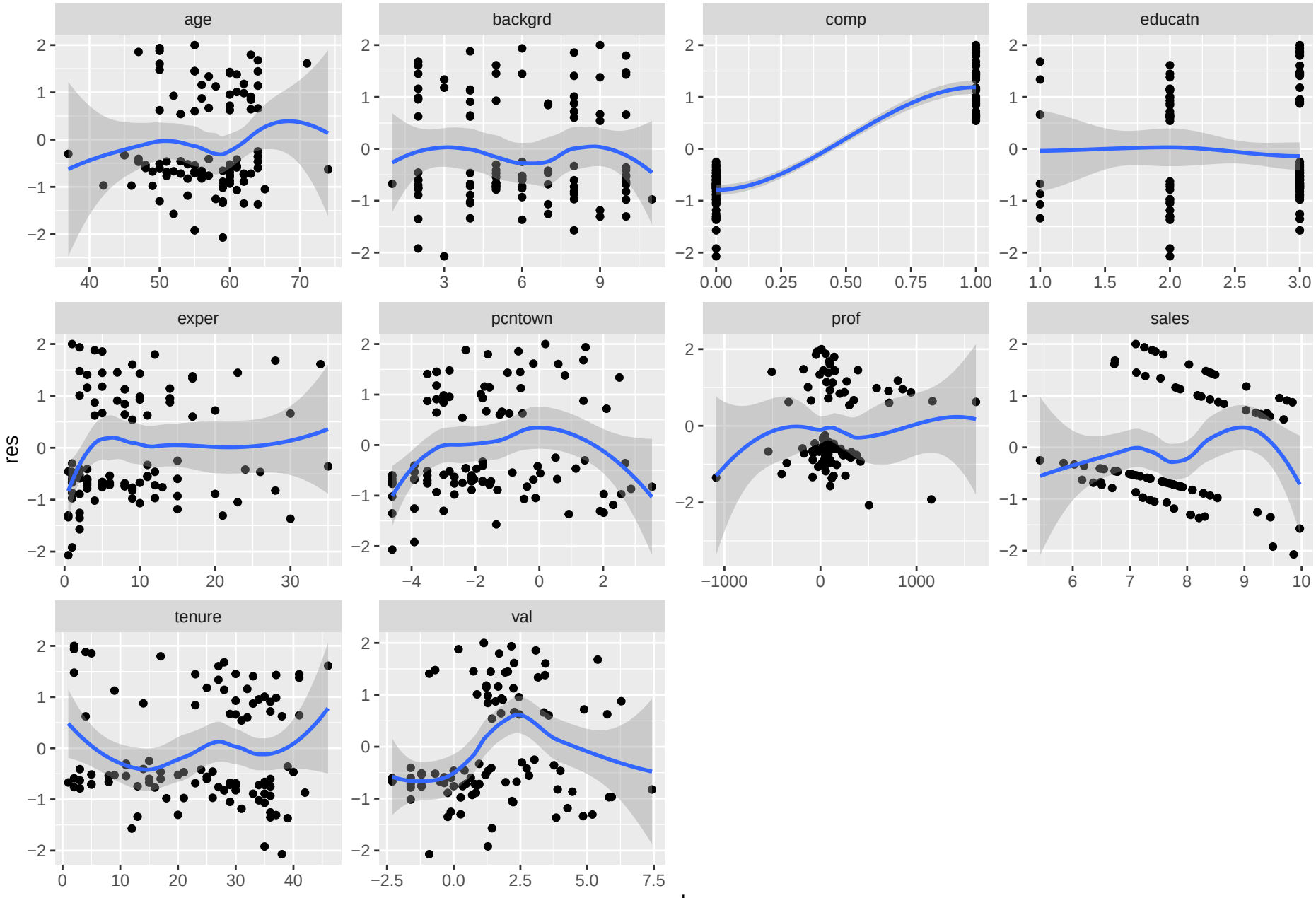
Example: CEO Salaries (all regressors)

```
### read data again
Data <- read.csv("../Project-0/CEO_compensations.csv")
names(Data) <- tolower(names(Data))
Data <- Data %>% mutate(backgrd=as.factor(backgrd), educatn=as.factor(educatn),
                        comp=log(comp), sales=log(sales), pcntown=log(pcntown),
                        val=log(val)) %>%
  select(-birth,-company) %>% mutate(comp=as.numeric(I(comp > 7)))

### full and a smaller model
gm1 <- glm(comp~., data=Data, family="binomial")
# library(car)
# Anova(gm1,type=2,test="LR")
gm0 <- glm(comp~.-backgrd-exper-val-pcntown-prof-age-tenure, data=Data,
           family="binomial")
# anova(gm0,gm1,test="LR") ### p-value 0.84

### residual plots
Data %>%
  mutate(res=resid(gm0),backgrd=as.numeric(backgrd),
         educatn=as.numeric(educatn)) %>%
  pivot_longer(-res) %>% ggplot(aes(y=res,x=value)) +
  facet_wrap(~ name, scales = "free") + geom_point() + geom_smooth()
```

Example: CEO Salaries (all regressors)



Example: CEO Salaries (all regressors)

```
gm2 <- glm(comp~educatn+sales+pcntown+I(pcntown^2), data=Data, family="binomial")
anova(gm0,gm2,test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: comp ~ (age + educatn + backgrd + tenure + exper + sales + val +
##     pcntown + prof) - backgrd - exper - val - pcntown - prof -
##     age - tenure
## Model 2: comp ~ educatn + sales + pcntown + I(pcntown^2)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        96    105.872
## 2        94     91.183  2   14.689 0.000646 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m1 <- lm(comp~educatn+sales+pcntown+I(pcntown^2), data=Data)
AIC(gm0,gm2,m1)
```

```
##     df      AIC
## gm0  4 113.8720
## gm2  6 103.1827
## m1   7 111.3620
```

Example: Comments

- residuals are organized in 2 clouds (since 2 values of the response)
  - one wouldn't see much, so normally close resids would be binned, but that's kind of what a smoother does, so...
- a smoother still suggests patterns
- quadratic effect in `pcntown` is the most obvious one
  - we also see e.g. the same problems with high `sales` values that we saw previously with linear models (both with continuous and discretized responses)

Interpretation

- let $X_n^\top \beta = \beta_1 + X_{n,2}\beta_2 + \ldots + X_{n,p}\beta_p$
- the intercept captures odds (of success) with zero regressors:
  - $P(Y_n = 1 \mid X_{n,2} = \ldots = X_{n,p} = 0) =: \pi_0$
  - $\Rightarrow \log\left(\frac{\pi_0}{1-\pi_0}\right) = \beta_1$
  - $\Rightarrow e^{\beta_1} = \frac{\pi_0}{1-\pi_0}$
- other coefficients capture odds ratio (between two observations that differ by 1 in the corresponding regressor):
  - let $x = (x_1, \ldots, x_p)^\top$ and $\tilde{x}^{(j)} = (x_1, \ldots, x_{j-1}, x_j + 1, x_{j+1}, \ldots, x_p)^\top$
  - $P(Y_n = 1 \mid X_n = x) =: \pi_n \Rightarrow \log\left(\frac{\pi_n}{1-\pi_n}\right) = \beta^\top x$
  - $P(Y_n = 1 \mid X_n = \tilde{x}^{(j)}) =: \tilde{\pi}_n^{(j)} \Rightarrow \log\left(\frac{\tilde{\pi}_n^{(j)}}{1-\tilde{\pi}_n^{(j)}}\right) = \beta^\top \tilde{x}^{(j)}$
  - subtracting: $\beta_j = \log\left(\frac{\pi_n}{1-\pi_n} \Big/ \frac{\tilde{\pi}_n^{(j)}}{1-\tilde{\pi}_n^{(j)}}\right)$
  - $\Rightarrow e^{\beta_j} = \frac{\pi_n}{1-\pi_n} \Big/ \frac{\tilde{\pi}_n^{(j)}}{1-\tilde{\pi}_n^{(j)}} = \frac{\pi_n(1-\tilde{\pi}_n^{(j)})}{\tilde{\pi}_n^{(j)}(1-\pi_n)}$

Example: Interpretation

```
summary(gm2)
```

```
##
## Call:
## glm(formula = comp ~ educatn + sales + pcntown + I(pcntown^2),
##     family = "binomial", data = Data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3939  -0.7928  -0.2812   0.7569   1.8091
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -9.72256    2.74777  -3.538 0.000403 ***
## educatn1      0.01703    1.09642   0.016 0.987610
## educatn2     -0.86728    1.13810  -0.762 0.446037
## sales         1.36324    0.34991   3.896 9.78e-05 ***
## pcntown      -0.16411    0.21027  -0.780 0.435111
## I(pcntown^2) -0.22832    0.07655  -2.982 0.002860 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 130.684  on 99  degrees of freedom
## Residual deviance:  91.183  on 94  degrees of freedom
## AIC: 103.18
##
## Number of Fisher Scoring iterations: 5
```

Section 3

Project 2

Data and Goals

- $N = 11630$ observations
  - sessions, i.e. instances of user visiting an e-commerce website, they take some time and may or may not result in a purchase
- 17 features
  - 10 numerical/ordinal
  - 7 categorical
- plus the response variable (whether there was a purchase during the session)

Goals:

1. explore the data to understand customer behavior on the e-commerce website
2. build a logistic regression model
   - explain how are the odds of purchase affected by date-related features
3. evaluate prediction performance of your model (code provided below)

Features

- coming from the e-commerce website itself
  - `Revenue` - the response variable, 0/1 indicating whether a purchase was made or not
  - `Administrative` - no. of administrative-type pages that the user visited
  - `Administrative_Duration` - time spent on administrative pages
  - `Informational` - no. of informational-type pages visited
  - `Informational_Duration` - time spent on informational-type pages
  - `ProductRelated` - no. of product-related-type pages visited
  - `ProductRelated_Duration` - time spent on product-related-type pages
- or are derived from date
  - `Month` - which month the session took place
  - `Weekend` - binary indicator of whether the session took place during a weekend
  - `SpecialDay` - value in $[0, 1]$ indicating closeness of the session to a special day (e.g. Mother's day, etc.) taking into account delivery, etc. (e.g. `special_day > 0` from Feb 2 reaching value 1 on Feb 12)

Features (cntd.)

- or coming from Google Analytics
  - `BounceRates` - avg. bounce rate of pages visited (for a specific webpage, the bounce rate is the percentage of users who enter the webpage and then leave without triggering any other requests during their sessions)
  - `ExitRates` - avg. exit rate of pages visited (for a specific webpage, the exit rate is the proportion of page views to the page that were the last in the session)
  - `PageValues` - avg. page value of pages visited (for a specific webpage, the page value gives an idea of how much each page contributes to the site' revenue)
  - `OperatingSystems` - operating systems of the users coded as integers
  - `Browser` - web browsers of the users coded as integers
  - `Region` - geographic region in which the user is located coded as integers
  - `TrafficType` - where from the user arrived at the site (e.g. ad banner, SMS link, direct URL, etc.) coded as integers
  - `VisitorType` - self-explanator (but no clue what `Other` mean)

Detailed Instructions (for your convenience only)

1. Improve the variable names.
   - change `Revenue` to `Purchase`
   - change the strange mix of `CamelCase` and `snake_case` to a proper `snake_case`
   - think of better variable names, but keep consistency, e.g. I would go for `n_admin_page` and `time_admin_page` instead of the first two feature names above
2. Wrangle the data further.
   - some variables should be factors, re-type them
   - some levels of some factors should be merged together
   - what about some transformation?
   - any strange values of regressors that can warrant their own parameter(s)?
3. Build a **logistic regression** model.
   - **use deviance tests** to help you decide which features to discard
   - deviance test will be done next time, but you can simply do `anova(model,submodel,test="LRT")` or `Anova(model,type="II",test="LR")` like in linear models
   - you can also use linear instead of logistic models to help you at the

Detailed Instructions (for your convenience only)

4. Calculate AUC of your final model using the provided code.
   - clearly report the AIC, residual deviance, and AUC of your final model as a table at the end of your report (winner gets candy!)

```
library(pROC)
AUC_eval <- function(gmodel,Data){
  set.seed(517)
  Folds <- matrix(sample(1:dim(Data)[1]), ncol=5)
  AUC <- rep(0,5)
  for(k in 1:5){
    train <- Data[-Folds[,k],]
    test <- Data[Folds[,k],]
    my_gm <- glm(gmodel$formula, family="binomial", data=train)
    test_pred <- predict(my_gm, newdata = test, type="response")
    AUC[k] <- auc(test$purchase,test_pred)
  }
  return(mean(AUC))
}
gm1 <- glm(purchase~., family="binomial", data=Data)
AUC_eval(gm1,Data)
```

Detailed Instructions (for your convenience only)

Benchmark AUC: 0.895

Note: This project is heavy on data wrangling, the data set is large, and models with discrete response and (many) continuous regressors are tricky w.r.t to residual diagnostics, so I do not expect detailed justifications. This is really a more prediction-oriented, ML-type project, but... In a ML competition, there would be a hold-out set to evaluate performance, and the winner would be whoever has the best test set performance. Here, there is no hold-out set, instead winner is the one who can explain **why** is his cross-validated performance the best.