Logistic regression

Linear regression models are built for a continuous outcome variable with normally distributed residuals

So what if your outcome is *discrete*?

Examples:

Outcome of an individual trial:
- success (1) or a failure (0) (e.g. remembered correctly or not)
- chose option A or option B

Occurrence across subjects:
- received a diagnosis / developed a disease: yes (1) no (0)

Classification
- category A or category B - wt vs knockout

Binary outcomes = yes/no or success/failure
binomial measures = number of yesses or successes in n trials
A sequence of independent trials like this with the same probability of success is called a ***Bernoulli process***.

Linear regression models are built for a continuous outcome variable with normally distributed residuals

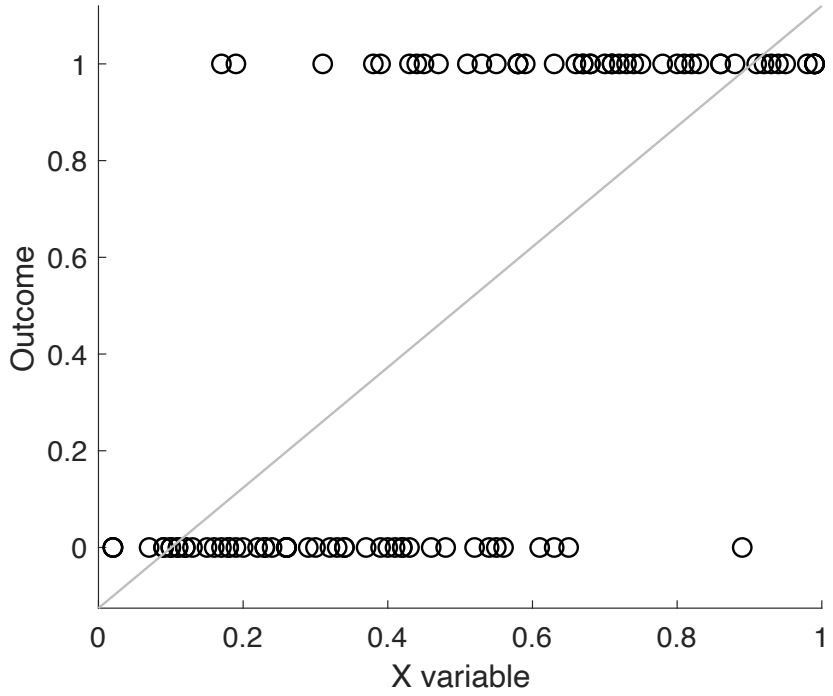
So what if your outcome is *discrete*?

You can run lm() but it will not fit your data well!

- normality of residuals may be violated
- violation of homoscedasticity (variance gets compressed at ends of range)
- model predictions could fall outside [0,1] range

→ biased parameter estimates and incorrect p-values!

~Expected values are really predicted probabilities~



Linear regression models are built for a continuous outcome variable with normally distributed residuals

So what if your outcome is *discrete*?

Expand our concept of regression model

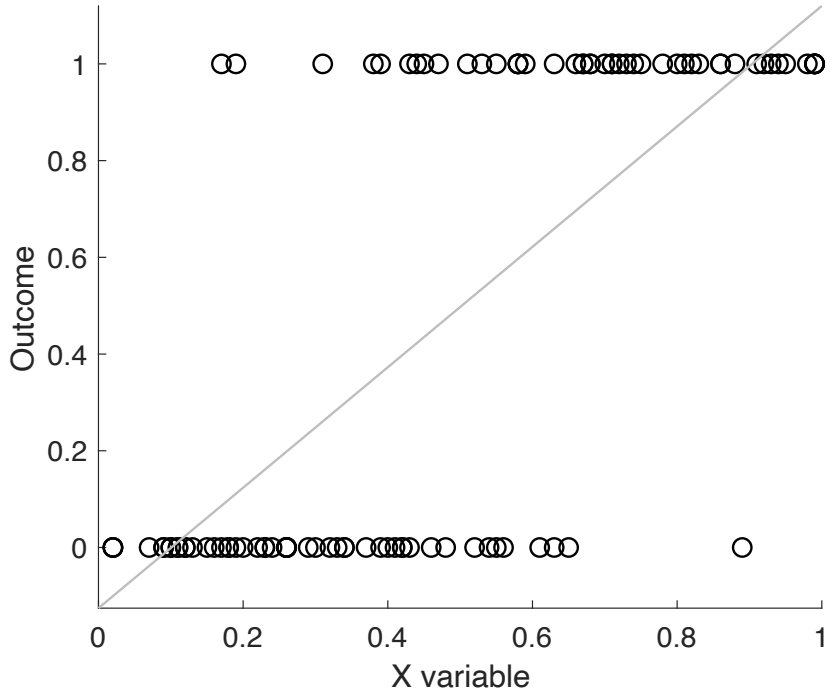
We have:

- linear predictor: optimal linear combination of predictor variables x
- response distribution: probability distribution for outcome variable

We need:

- link function: relate the linear predictor to the response distribution

* For standard linear regression, the link function is just an identity: the optimal linear combination predicts the outcome variable (y) as a normal distribution with constant variance – just add up predictors and multiply by appropriate coefficients *



Binary outcomes = yes/no or success/failure) and
binomial measures = number of yeses or successes in n trials
A sequence of independent trials like this with the same probability of success is called a ***Bernoulli process***.

Linear regression models are built for a continuous outcome variable with normally distributed residuals

So what if your outcome is *discrete*?

Expand our concept of regression model

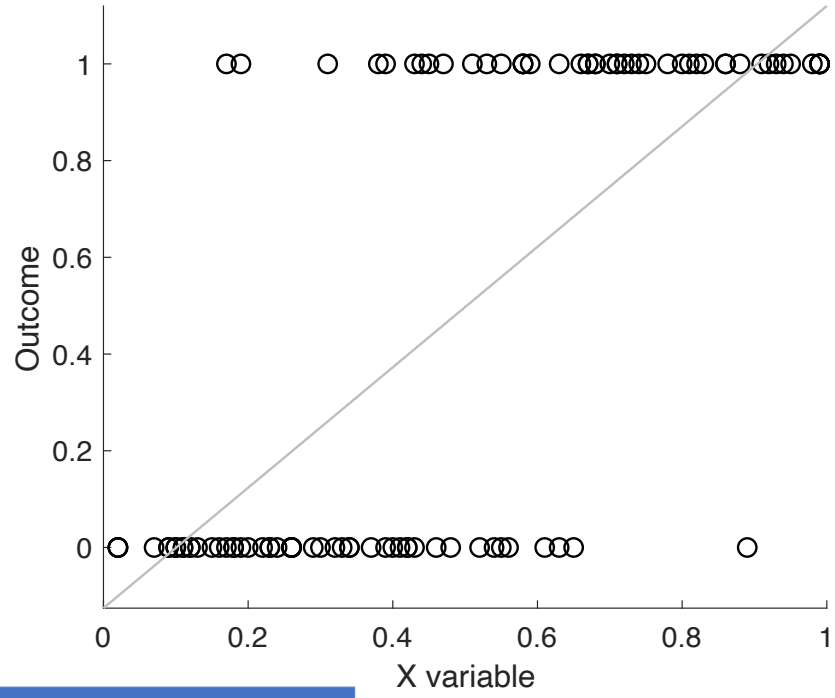
We have:

- linear predictor: optimal linear combination of predictor variables x
- response distribution: probability distribution for outcome variable

We need:

- link function: relate the linear predictor to the response distribution

* For standard linear regression, the link function is just an identity: the optimal linear combination of x's predicts the outcome variable (y) as a normal distribution with constant variance – just add up predictors and multiply by appropriate coefficients *



	Linear Regression	Binomial (Logistic) Regression
Linear predictor	$\eta_i = \beta_0 + \beta_1 x_{1i} + \cdots$	$\eta_i = \beta_0 + \beta_1 x_{1i} + \cdots$
Link function	$\mu_i = \eta_i$	$\text{logit}(\mu_i) = \eta_i$
Response distribution	$y_i \mid \mu_i \sim N(\mu_i, \sigma^2)$	$y_i \mid \mu_i \sim \text{Bernouilli}(\mu_i)$

Logistic (binomial) regression

If outcomes follow a probability distribution, the logit function will transform outcome bounded by 0 and 1 to a continuous range:

$\text{logit}\, p = \ln(\frac{p}{1-p})$ - Now, your linear predictors (x's) predict logit(p), where p is the probability of y = 1

Where does this come from?
$(\frac{p}{1-p})$ is the *Odds Ratio*

	Linear Regression	Binomial (Logistic) Regression
Linear predictor	$\eta_i = \beta_0 + \beta_1 x_{1i} + \cdots$	$\eta_i = \beta_0 + \beta_1 x_{1i} + \cdots$
Link function	$\mu_i = \eta_i$	$\text{logit}(\mu_i) = \eta_i$
Response distribution	$y_i \mid \mu_i \sim N(\mu_i, \sigma^2)$	$y_i \mid \mu_i \sim \text{Bernouilli}(\mu_i)$

Logistic (binomial) regression

If outcomes follow a probability distribution, the logit function will transform outcome bounded by 0 and 1 to a continuous range:

$\text{logit}\, p = \ln(\frac{p}{1-p})$ - Now, your linear predictors (x's) predict logit(p), where p is the probability of y = 1

Where does this come from?

$(\frac{p}{1-p})$ is the *Odds* of an outcome

Odds and odds ratio example:

Table 6.1: Soccer goalkeepers' penalty kick saves when their team is and is not behind.

	Saves	Scores	Total
Behind	2	22	24
Not Behind	39	141	180
Total	41	163	204

(Source: Roskes et al. 2011.)

- The chance, or "odds" that a goalkeeper makes a save when their team is behind (2/24) vs that they don't (22/24) is $\frac{2/24}{22/24} = \frac{2}{22} = 0.09$
- We can also say the odds that a goal is scored when the goalkeeper's team is behind is $\frac{22}{2}$ or 11 to 1.
- On the other hand, the odds that they make a save when their team is *not behind* (39/180) vs that they don't (141/180) is 0.28
- We can also say that the odds that a goal is scored when the goalkeeper's team is *not* behind is $\frac{141}{39}$ or 3.6 to 1

So, the odds of the other team scoring a goal are much higher when the goalkeeper's team is behind.

- How much higher is the *Odds Ratio* = 11/3.6 → or the odds of scoring a goal are roughly 3x higher

Odds of success in either situation = *probability of success / probability of failure* = (#success/n) / (#failure/n)

- If success and failure are the only possible outcomes, then the probability of failure = 1- probability of success

let p = probability of success

Odds = $(\frac{p}{1-p})$

ln $(\frac{p}{1-p})$ = *log odds*

Note that this varies from 0 to ∞ !

Roskes et al., (2011). *Psychological Science*, *22*(11), 1403-1407. https://doi.org/10.1177/0956797611418677

Logistic (binomial) regression

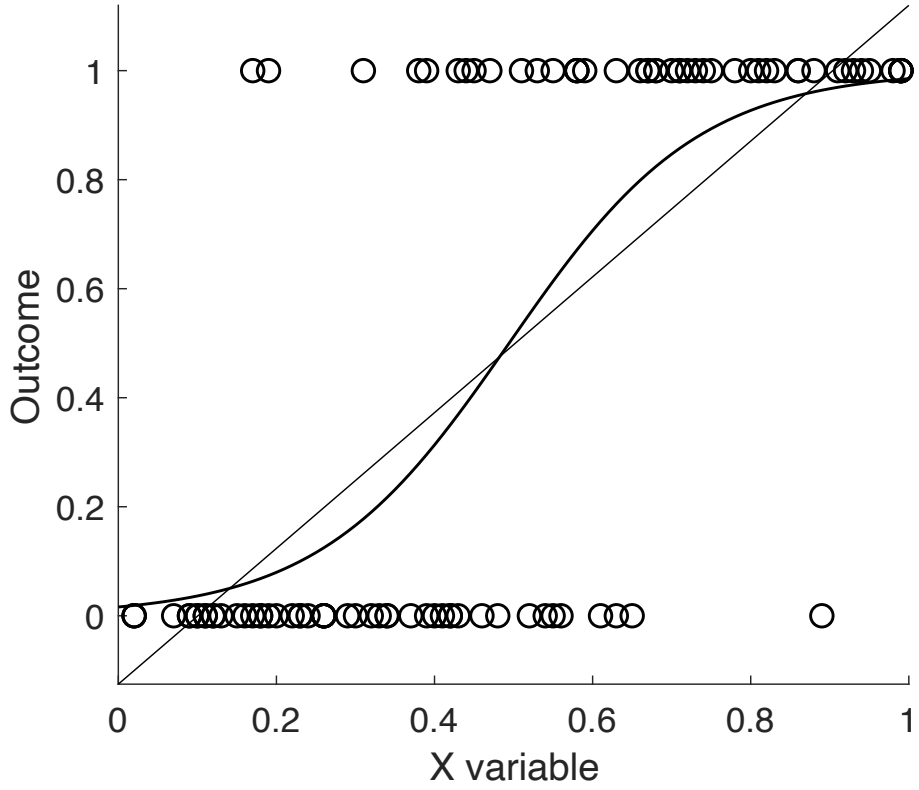
If outcomes follow a probability distribution, the logit function will transform outcome bounded by 0 and 1 to a continuous range:

$\text{logit}\, p = \ln(\frac{p}{1-p})$ - Now, your linear predictors (x's) predict logit(p), where p is the probability of y = 1

Where does this come from?

$(\frac{p}{1-p})$ is the *Odds* of an outcome

Logit function = log odds



Logistic (binomial) regression

If outcomes follow a probability distribution, the logit function will transform outcome bounded by 0 and 1 to a continuous range:

$\text{logit}\, p = \ln(\frac{p}{1-p})$ - Now, your linear predictors (x's) predict logit(p), where p is the probability of y = 1

Where does this come from?

$(\frac{p}{1-p})$ is the *Odds* of an outcome

Logit function = log odds

X variables → linear transformation → nonlinear transformation → + noise → Y variable

Linear regression

X variables → linear transformation → + noise → Y variable

- weighted combination of X's

- Gaussian

- Continuous

Logistic regression

X variables → linear transformation → nonlinear transformation → + noise → Y variable

- weighted combination of X's

- logit function

- binomial outcome

- follows a probability distribution

General linear model *assumes linear relationships between x's and y variable*

X variables → linear transformation → + noise → Y variable

- *weighted combination of X's*

- *Gaussian*

- *Continuous*

GLM ≠ GLM

Generalized linear model *allows the relationship between x's and y to vary via a link function*

X variables → linear transformation → potentially nonlinear transformation → + noise → Y variable

- *weighted combination of X's*

- *link function*

- *many different distributions*

Generalized Linear Models *in R*

Generalized linear models are fit using the **glm( )** function. The form of the **glm** function is

glm(*formula* **, family=** *familytype***(link=***linkfunction***), data=)**

Family	Default Link Function
binomial	(link = "logit")
gaussian	(link = "identity")
Gamma	(link = "inverse")
inverse.gaussian	(link = "1/mu^2")
poisson	(link = "log")
quasi	(link = "identity", variance = "constant")
quasibinomial	(link = "logit")
quasipoisson	(link = "log")

Logit is for logistic regressions - when you are predicting a binary outcome from a set of continuous predictors.

```
# Logistic Regression
# where F is a binary factor and
# x1-x3 are continuous predictors
fit <- glm(F~x1+x2+x3,data=mydata,family=binomial())
summary(fit) # display results
confint(fit) # 95% CI for the coefficients
exp(coef(fit)) # exponentiated coefficients
exp(confint(fit)) # 95% CI for exponentiated coefficients
predict(fit, type="response") # predicted values
residuals(fit, type="deviance") # residuals
```

Generalized Linear Models *in R*

Generalized linear models are fit using the **glm( )** function. The form of the **glm** function is

glm(*formula* **, family=** *familytype***(link=***linkfunction***), data=)**

Family	**Default Link Function**
binomial	(link = "logit")
gaussian	(link = "identity")
Gamma	(link = "inverse")
inverse.gaussian	(link = "1/mu^2")
poisson	(link = "log")
quasi	(link = "identity", variance = "constant")
quasibinomial	(link = "logit")
quasipoisson	(link = "log")

The identity function turns this into a linear regression

Generalized Linear Models *in R*

Generalized linear models are fit using the **glm( )** function. The form of the **glm** function is

glm(*formula* **, family=** *familytype***(link=***linkfunction***), data=)**

Family	Default Link Function
binomial	(link = "logit")
gaussian	(link = "identity")
Gamma	(link = "inverse")
inverse.gaussian	(link = "1/mu^2")
poisson	(link = "log")
quasi	(link = "identity", variance = "constant")
quasibinomial	(link = "logit")
quasipoisson	(link = "log")

```
# Poisson Regression
# where count is a count and
# x1-x3 are continuous predictors
fit <- glm(count ~ x1+x2+x3, data=mydata, family=poisson())
summary(fit) display results
```

Poisson regression is useful when predicting counts as an outcome variable, from a set of continuous predictors.

Use GLMs to analyze your data rather than comparing means!!!

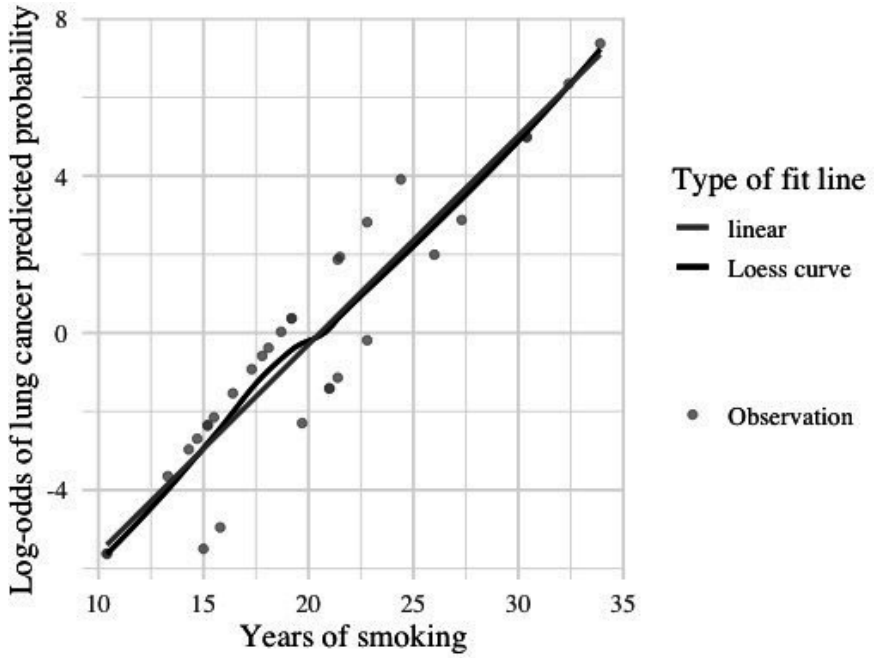
```
> t.test(mean_trial ~ condition, data = mouse_means, var.equal = TRUE)

        Two Sample t-test

data:  mean_trial by condition
t = -3.2308, df = 18, p-value = 0.004638
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.23103996 -0.04896004
sample estimates:
mean in group control    mean in group drug
                 0.61                  0.75
```

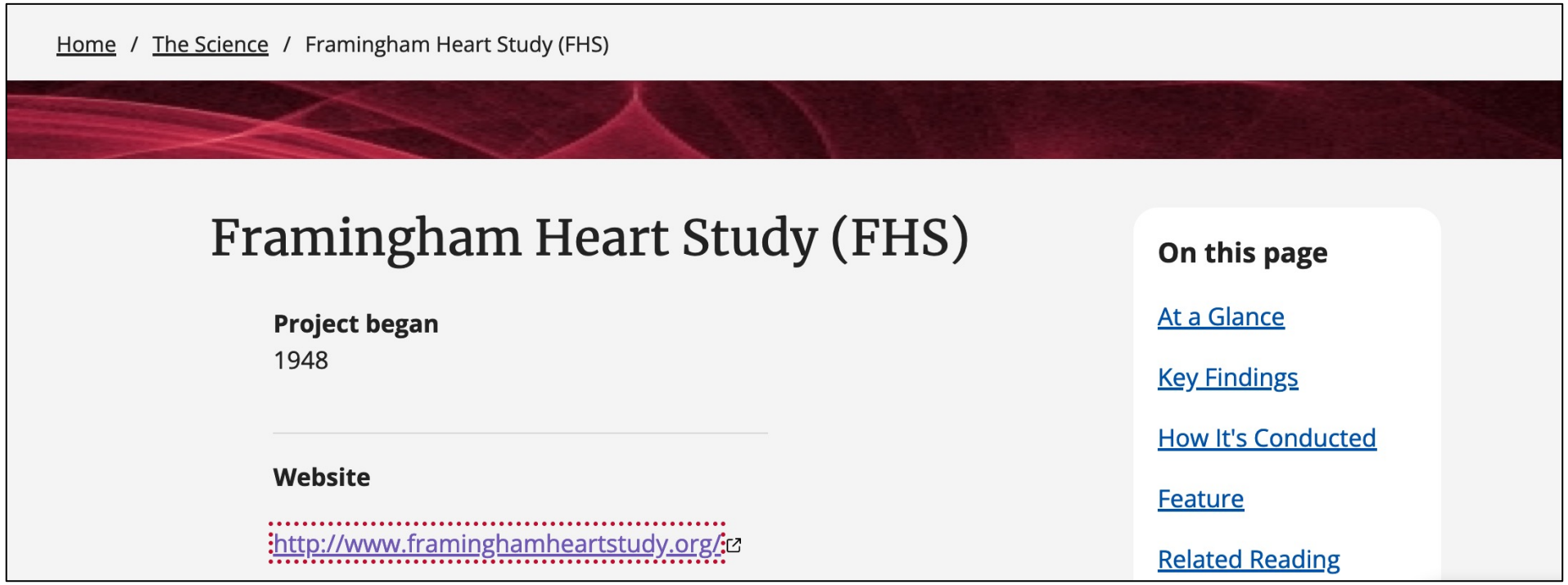
```
> glm(trial_outcome ~ condition, data = recog_data, family = binomial()) %>% summary()

Call:
glm(formula = trial_outcome ~ condition, family = binomial(),
    data = recog_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6651  -1.3723   0.7585   0.9943   0.9943

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.4473     0.1450   3.085  0.00203 **
conditiondrug   0.6513     0.2184   2.983  0.00286 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Logistic Regression Assumptions

1.Binary Outcomes The response variable is dichotomous (two possible responses) or the sum of dichotomous responses.
2.Independence The observations must be independent of one another.
3.Lack of multicollinearity among predictor variables
4.Linearity x variables are linearly related to the log odds of the outcome

Note: homoscedasticity is not an assumption. By definition, the variance of a binomial random variable is np(1−p), so that variability is highest when p=.5.





Outcome:

Occurrence of coronary heard disease (CHD) within 10 years (1=yes, 2=no)

CHD = Myocardial infarction (MI), angina pectoris, heart failure (HF), and coronary death

Predictors:

	VIFs:
Age	1.2186
Sex	1.1684
Heart rate	1.0663
cigarettes per day	1.1945
cholesterol	1.1032
diabetes	1.0209
BMI	1.1360
high blood pressure	1.2256

https://www.kaggle.com/code/nishanthbejgam/logistic-regression-beginner-learning/input

Outcome:

Occurrence of coronary heard disease (CHD) within 10 years (1=yes, 2=no)

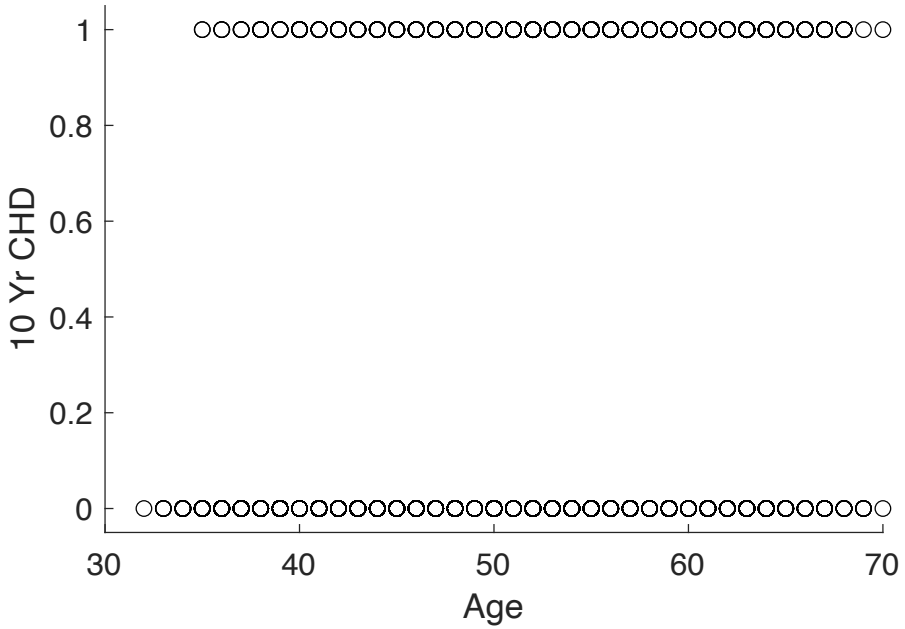
CHD = Myocardial infarction (MI), angina pectoris, heart failure (HF), and coronary death

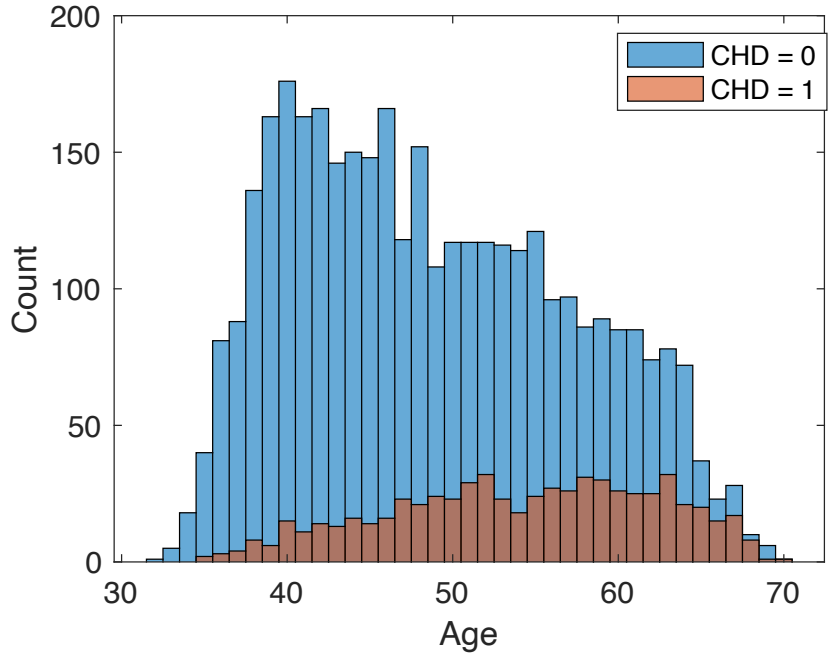
Predictors: Age
Sex
Heart rate
cigarettes per day
cholesterol
diabetes
BMI
high blood pressure

	1 Estimate	2 SE	3 tStat	4 pValue
1 (Intercept)	-7.1861	0.5634	-12.7545	2.9434e-37
2 x1	0.0729	0.0060	12.2396	1.9098e-34
3 x2	0.4723	0.1001	4.7209	2.3477e-06
4 x3	0.0015	0.0039	0.3789	0.7047
5 x4	0.0220	0.0039	5.5916	2.2492e-08
6 x5	0.0023	0.0010	2.2476	0.0246
7 x6	0.7577	0.2202	3.4415	5.7857e-04
8 x7	0.0124	0.0114	1.0852	0.2778
9 x8	0.6137	0.0994	6.1751	6.6107e-10

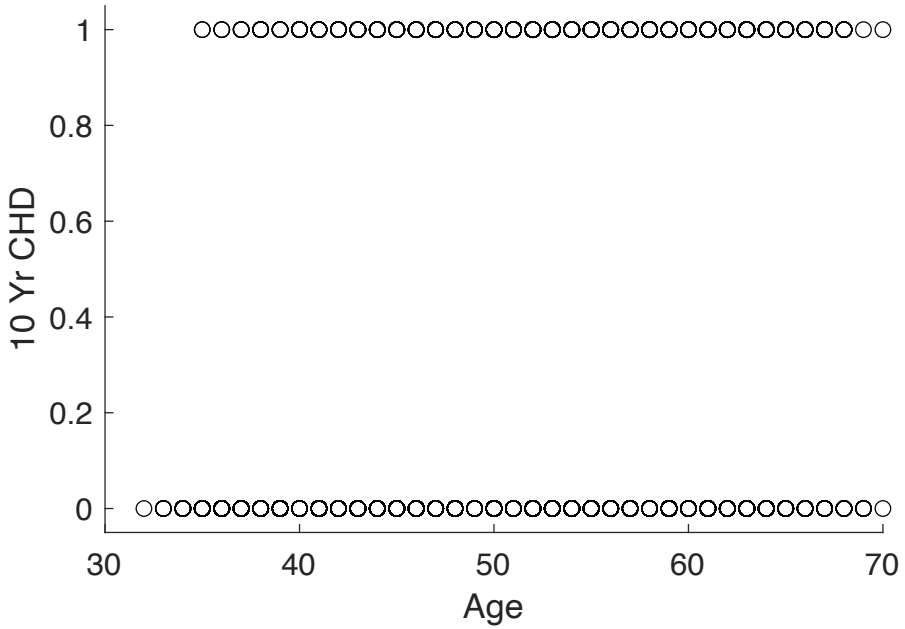
https://www.kaggle.com/code/nishanthbejgam/logistic-regression-beginner-learning/input

Does Age predict 10 year risk of CHD?

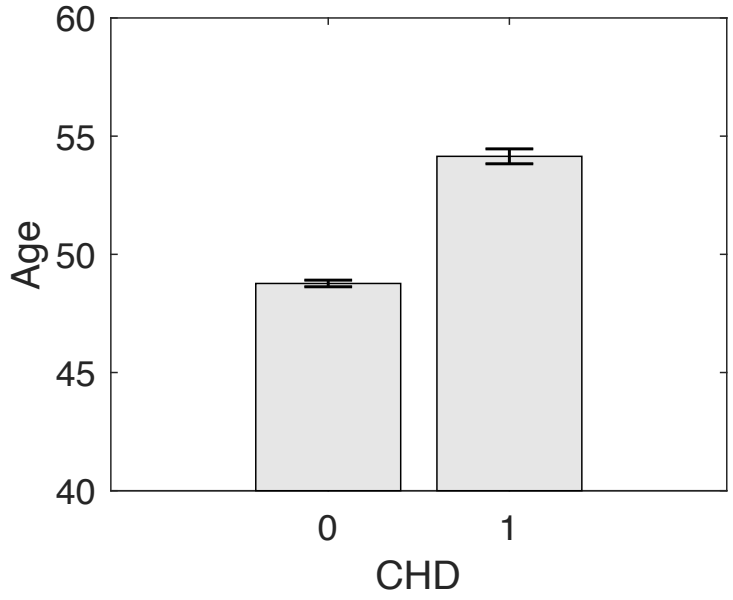




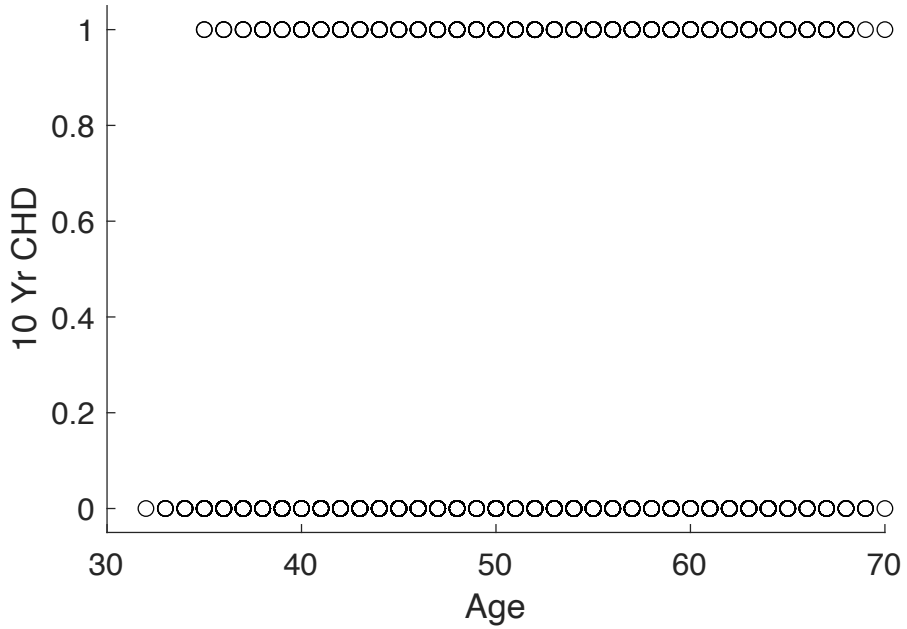
Does Age predict 10 year risk of CHD?



t-test $p = 6.8 \times 10^{-50}$

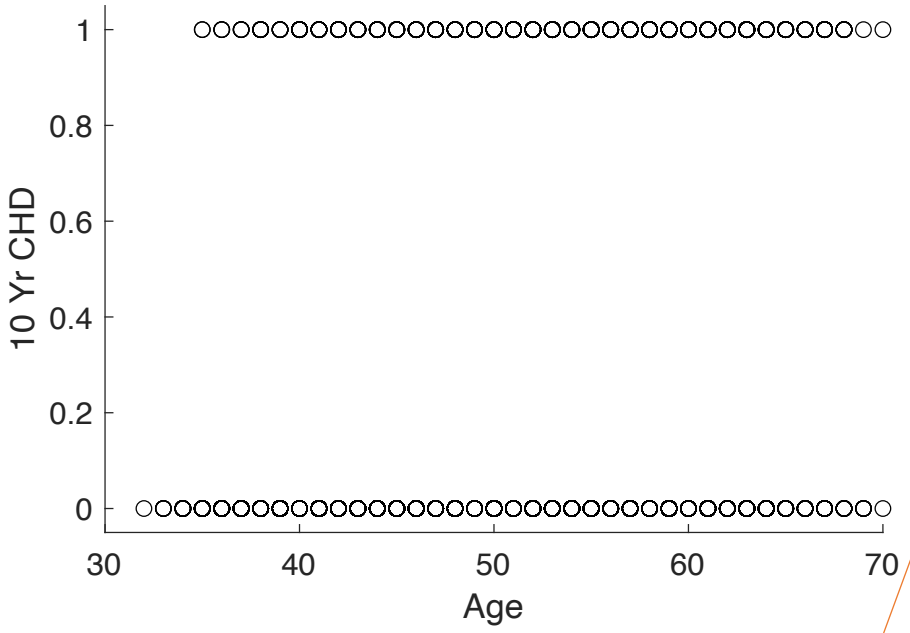


Does Age predict 10 year risk of CHD?



Predictors: Age
Sex
Heart rate
cigarettes per day
cholesterol
diabetes
BMI
high blood pressure

Does Age predict 10 year risk of CHD?

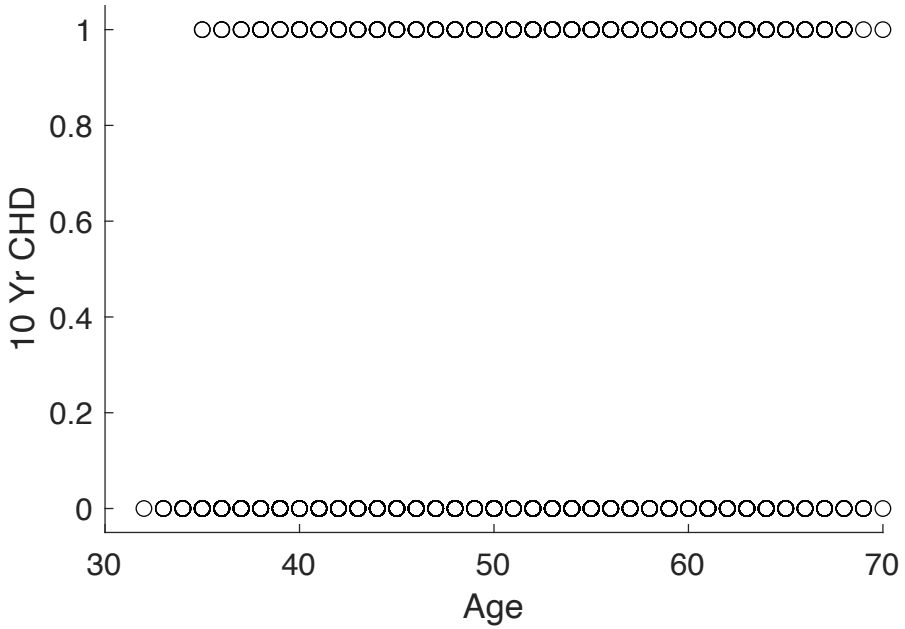



```
Call:
glm(formula = CHD ~ age, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0384  -0.6262  -0.4582  -0.3697   2.4488

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.558053   0.283790  -19.59   <2e-16 ***
age          0.074598   0.005266   14.17   <2e-16 ***
---
```

What is this??

Deviance residuals are analogous to regular residuals. If the median is close to 0, the model is not biased

Does Age predict 10 year risk of CHD?



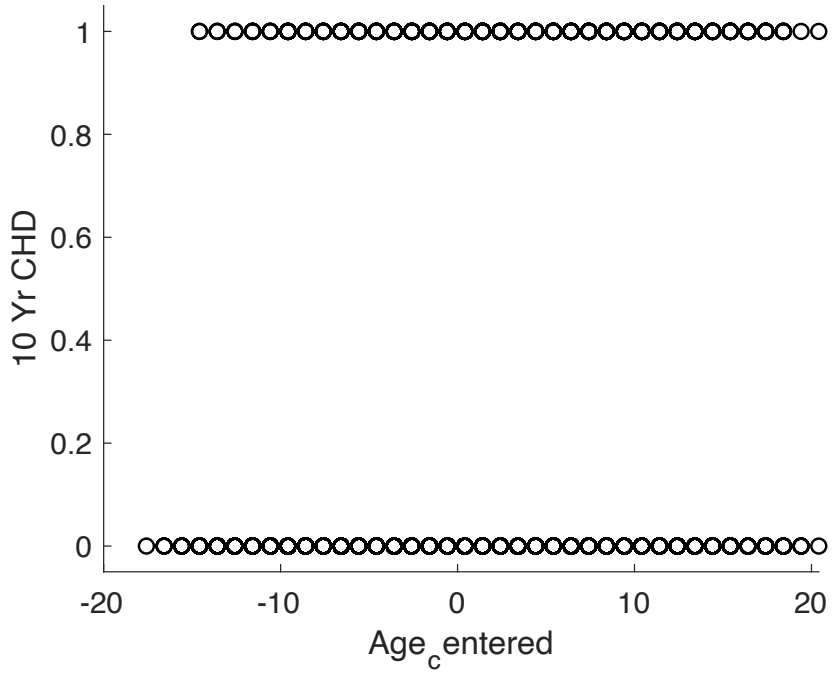

```
Call:
glm(formula = CHD ~ age, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0384  -0.6262  -0.4582  -0.3697   2.4488

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.558053   0.283790  -19.59   <2e-16 ***
age          0.074598   0.005266   14.17   <2e-16 ***
---
```

Intercept: log odds of CHD when age = 0

$\ln(\frac{p}{1-p})$ = -5.56 → p ~= 0.0038 1-p ~= 0.996

This would be much more useful if we mean center the predictor!

Does Age predict 10 year risk of CHD?



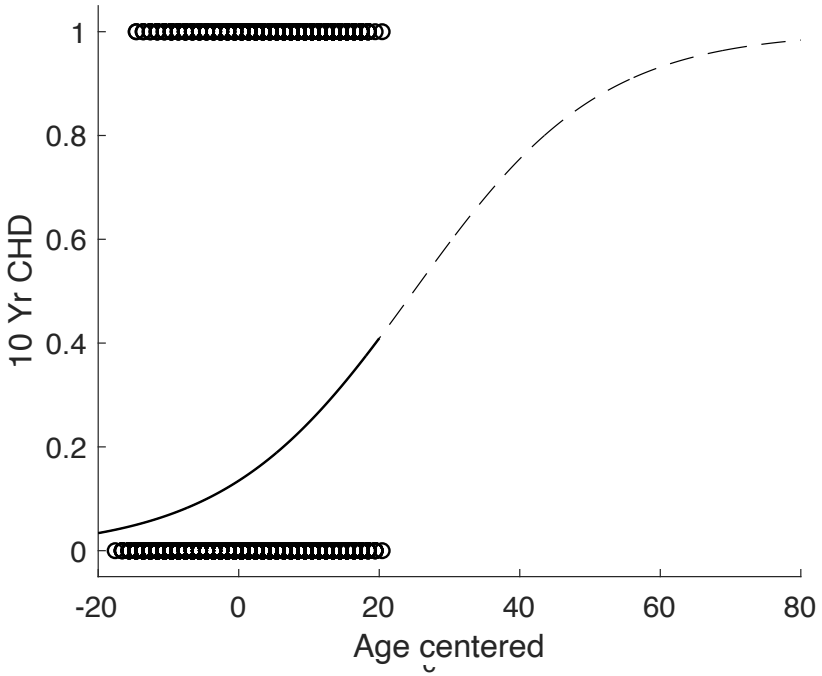

```
Call:
glm(formula = CHD ~ age_centered, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0384  -0.6262  -0.4582  -0.3697   2.4488

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.859115   0.048000  -38.73   <2e-16 ***
age_centered  0.074598   0.005266   14.17   <2e-16 ***
---
```

Intercept: log odds of CHD when age_centered = 0, or age = mean age = 45.6

$\ln(\frac{p}{1-p})$ = -1.86 → p ~= 0.135 1-p ~= 0.865

So the probability of having CHD within 10 years at the age of 45.6 is 13.5%, the probability of not having it is 86.5%

Does Age predict 10 year risk of CHD?

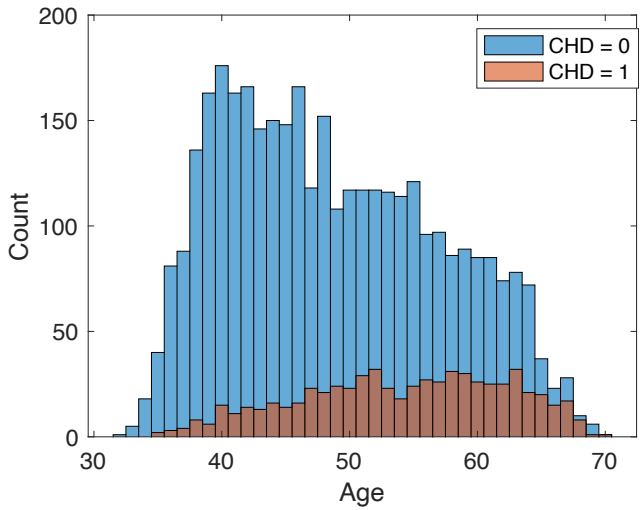


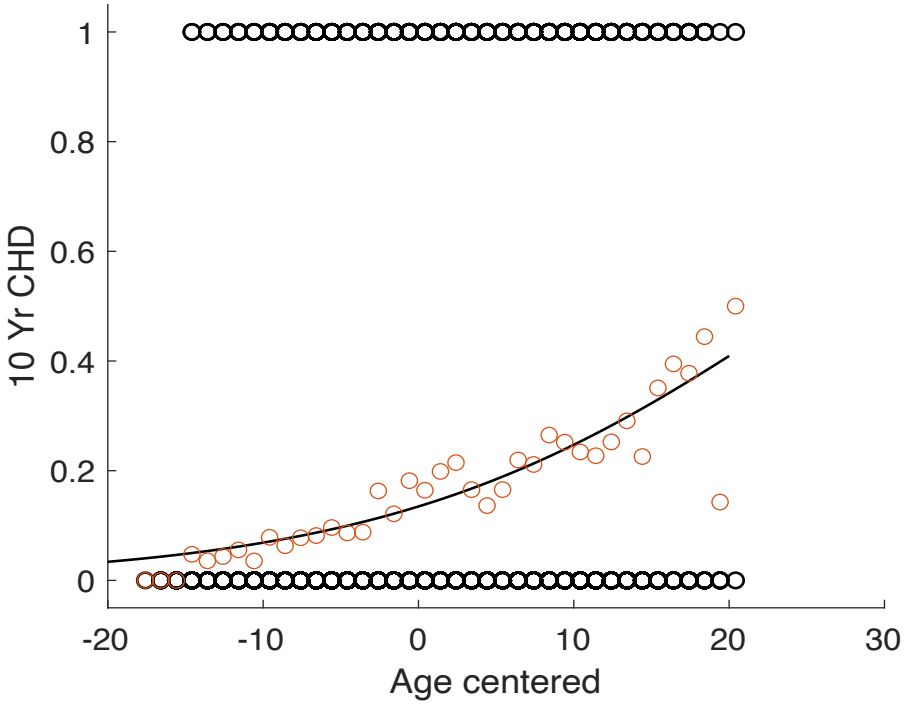

```
Call:
glm(formula = CHD ~ age_centered, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0384  -0.6262  -0.4582  -0.3697   2.4488

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.859115   0.048000  -38.73   <2e-16 ***
age_centered  0.074598   0.005266   14.17   <2e-16 ***
---
```

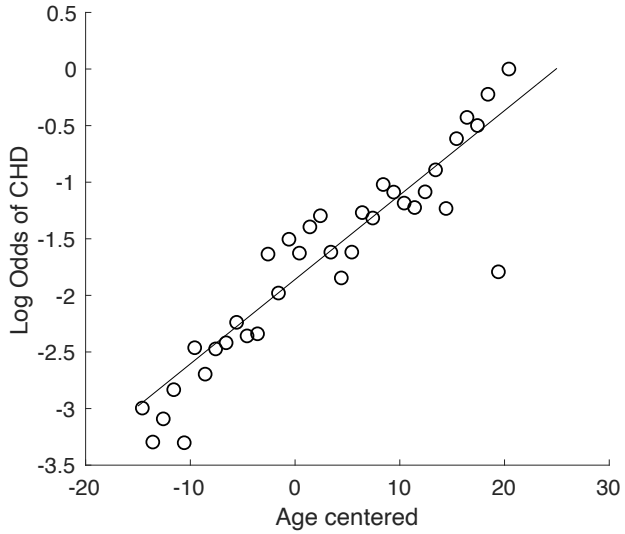
Intercept: log odds of CHD when age_centered = 0, or age = mean age = 45.6

Age_centered coefficient: the change in log-odds of having CHD in 10yrs with each unit (year) increase in age

~Expected values are really predicted probabilities~







The probability of having CHD within 10 years at the age of 45.6 is 13.5%, the probability of not having it is 86.5%

Outcome:

Occurrence of coronary heard disease (CHD) within 10 years (1=yes, 2=no)

CHD = Myocardial infarction (MI), angina pectoris, heart failure (HF), and coronary death

Predictors: Age
Sex
Heart rate
cigarettes per day
cholesterol
diabetes
BMI
high blood pressure

	1 Estimate	2 SE	3 tStat	4 pValue
1 (Intercept)	-7.1861	0.5634	-12.7545	2.9434e-37
2 x1	0.0729	0.0060	12.2396	1.9098e-34
3 x2	0.4723	0.1001	4.7209	2.3477e-06
4 x3	0.0015	0.0039	0.3789	0.7047
5 x4	0.0220	0.0039	5.5916	2.2492e-08
6 x5	0.0023	0.0010	2.2476	0.0246
7 x6	0.7577	0.2202	3.4415	5.7857e-04
8 x7	0.0124	0.0114	1.0852	0.2778
9 x8	0.6137	0.0994	6.1751	6.6107e-10

https://www.kaggle.com/code/nishanthbejgam/logistic-regression-beginner-learning/input

Things to know…

Interactions in logistic regression models are complicated. The model is already in log scale, so the relationship between variables is *multiplicative* not additive. Adding an interaction term test for synergy (more than multiplicative) between two variables, or whether two variables interact to be less than multiplicative

Confidence intervals are weird. (if R gives CIs for a logistic regression model, go with them)
The 95% CI around b_1 is usually b_1+/-1.96 x SE(b_1)
But since logistics are multiplicative, the 95% CI for the Odds Ratio is:

exp[b_1 - 1.96 x SE(b_1)] to exp[b_1 + 1.96 x SE(b_1)]

This is an asymmetric CI for the odds ratio!

Calculating model significance is also different. For the linear regression, we evaluate the overall model fit with the variance explained by all the predictors. For the logistic regression, we cannot calculate a variance, so we define and evaluate the *deviance* instead.

Overall model significance

- The *residual deviance* output from the glm call is the difference between the deviance of a null model (ie, no predictors, just the data) and the fitted model.
- The decease in deviance after including the predictors follows a chi-square ($\chi 2$) distribution, and can be used for significance testing

Call: summary(mdl)

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.859115   0.048000  -38.73   <2e-16 ***
age_centered  0.074598   0.005266   14.17   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3611.5  on 4237  degrees of freedom
Residual deviance: 3396.3  on 4236  degrees of freedom
AIC: 3400.3

Number of Fisher Scoring iterations: 5
```

→

```
mdl <- glm(CHD~age_centered,family = binomial)
summary(mdl)
d_chi <- 3611.5 - 3396.3
d_df <- 4237 - 4236
1 - pchisq(d_chi, d_df)
```

```
[1] 0
```

Summary

- If outcomes follow a probability distribution, the logit function will transform outcome bounded by 0 and 1 to a continuous range
- Logit is one of many link functions that can be implemented in generalized linear models

Generalized linear model *allows the relationship between x's and y to vary via a link function*

X variables → linear transformation → potentially nonlinear transformation → + noise → Y variable

- many different distributions

- weighted combination of X's

- link function

Linear regression:

Logistic regression: