Missing Data

Week 2

Outline of the course

1) Missing Data Mechanisms. How are missing data generated and why should we care? Complete Case Analysis. Getting comfortable with R.
2) Simple missing data fixes: listwise deletion, available case, LVCF, mean imputation, dummy variable methods, drawing from empirical distribution (also more on logistic regression, and more on R)
3) More complicated missing data fixes: weighting, hotdecking, regression imputation
4) Building blocks and overview of multiple imputation (including regression imputation with noise)
5) Multiple imputation in practice (software in R, simple analyses, and diagnostics)
6) Multiple imputation in practice (more complicated models and considerations, more advanced diagnostics)
7) More advanced imputation and other missing data methods

Simple missing data methods

Methods that throw away data

- Complete cases (listwise deletion)
- Complete variables

Methods that don’t throw away data

- Mean imputation
- Last value carried forward (LVCF)
- Dummy variable method
- Reports from others

Complete Cases

X_1	X_2	X_3	X_4
0	2	5	2
0	1	3	?
0	2	4	3
0	3	5	?
1	5	?	?
1	4	?	12
1	6	11	?
1	6	12	16

→

X_1	X_2	X_3	X_4
0	2	5	2
0	2	4	3
1	6	12	16

Complete Cases in R

Testing for missing values:

```
is.na(x)
```

Returns TRUE if *x* is missing.

Example:

```
> x=c(3, NA, 2)
> is.na(x)
[1] FALSE  TRUE FALSE
```

Complete Cases - R

The function `na.omit()` returns the object with listwise deletion of missing values.

Example:

```
> DF <- data.frame(x = c(1, 2, 3),
y = c(0, 10, NA))
> na.omit(DF)
  x  y
1 1  0
2 2 10
```

Complete Cases - R

The function `complete.cases()` returns a logical vector indicating which cases are complete.

Example:

```
> mat = matrix(c(1, NA, 3, 4), 2, 2)
> mat
     [,1] [,2]
[1,]    1    3
[2,]   NA    4
> mat[complete.cases(mat),]
[1] 1 3
```

Complete variables/Available variables

X_1	X_2	X_3	X_4
0	2	5	2
0	1	3	?
0	2	4	3
0	3	5	?
1	5	?	?
1	4	?	12
1	6	11	?
1	6	12	16

X_1	X_2
0	2
0	1
0	2
0	3
1	5
1	4
1	6
1	6

Complete variables/Available variables

- This throws out all *variables* that have any missing data
- *Advantage*: full sample size is retained
- *Disadvantage*: we may be left with only a few variables which aren’t of substantive interest
- How to do in R:

```
mat[, complete.cases(t(mat))]
[1] 3 4
```

Mean imputation

- Basic idea of imputation: substitute missing values with some ”reasonable” guess
- Mean imputation fills in all missing values for a given variable with the mean of the observed values for that variable
- Worst of all strategies: can cause *big* problems, particularly in skewed data or data with big pile-ups e.g. at zero (like income data)
- At best, reduces variance of that variable and biases measures of association with other variables (correlation, regression, …)
- Graham (2012): “I recommend that people should NEVER use this procedure.”

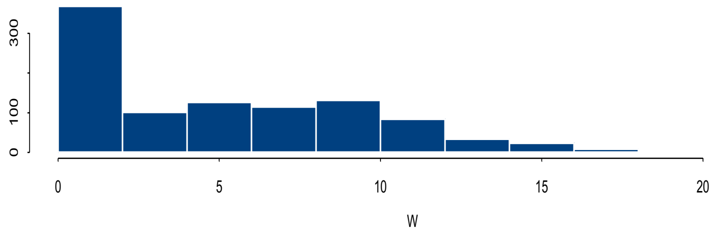
Mean imputation

$$\bar{X}_2 = 3.8$$
$$\bar{X}_3 = 7.8$$
$$\bar{X}_4 = 2.7$$

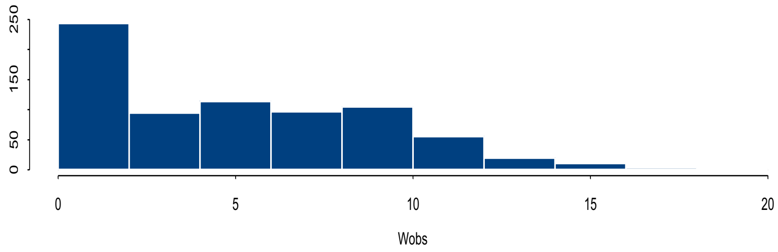
X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	?	7	3
0	?	6	?
1	5	?	?
1	3	?	?
1	4	9	?

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3.8	7	3
0	3.8	6	2.7
1	5	7.8	2.7
1	3	7.8	2.7
1	4	9	2.7

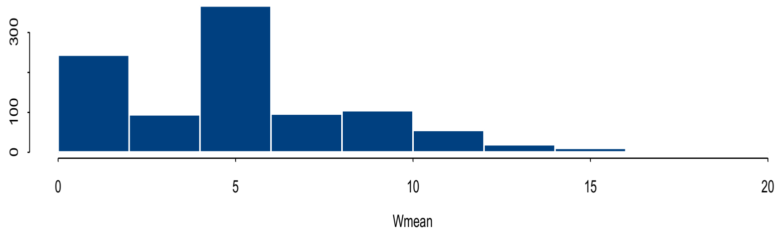
Illustration of mean imputation



First histogram is the complete data

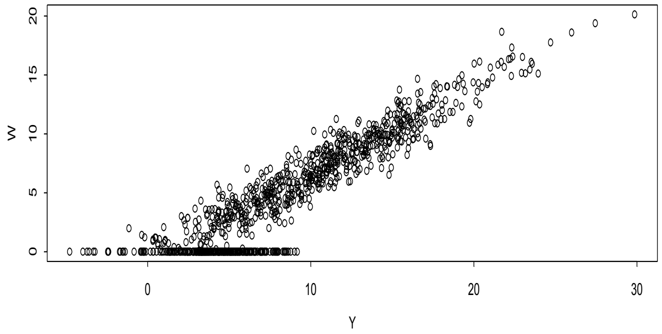


Second histogram is the observed data

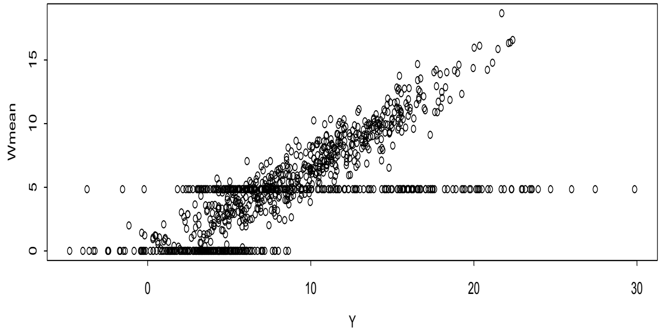


Third histogram is the data with mean imputation

Illustration of mean imputation



Scatterplot of the variable, *W*, before missing data, and another variable, *Y*



Scatterplot of the variable, *W*, with mean-imputed data, and another variable, Y

Correlation between variables *W* and *Y* is .92
Correlation between mean-imputed *W* and *Y* is .74

More advanced R code:

- Read data files in with `read.table` function
- User defined functions:

```
        fun1 = function(arguments)
                 {commands
              return(result)}
```

- More advanced vector referencing: `x[y < 0]` selects those `x` values whose corresponding `y` values are negative
- Use `cbind` to combine column vectors into matrix
- Use `sample` to select a sample from a vector

Mean imputation – Big dataset example

- Social Indicators Study (SIS) from Gelman and Hill book
- http://www.stat.columbia.edu/~gelman/arm/
- We will use the variable "earnings" and impute missing data using mean imputation
- First some cleaning needs to be done
- R code

Mean imputation – mice package

- Install the package

- The basic command is “mice”
- `mice(data, method = “mean”, m = 1, maxit = 1)`
- m = 1 indicates we want only one imputation
- maxit is the maximum number of iterations (needed for the multiple imputation methods)
- Data must have at least two columns!
- To see the complete data use complete() function

Drawing from the empirical distribution

- We know that one of the problems with mean imputation is that the imputed values don’t reflect the natural variability in the data
- One way to accomplish this is to draw from the empirical distribution of the observed data
- In other words we sample from the collection of observed values for a given value to fill in the missing values for that variable
- This is better than mean imputation but still ignores relationships between variables
- Suppose “var1” has missing data. Then code might look like:

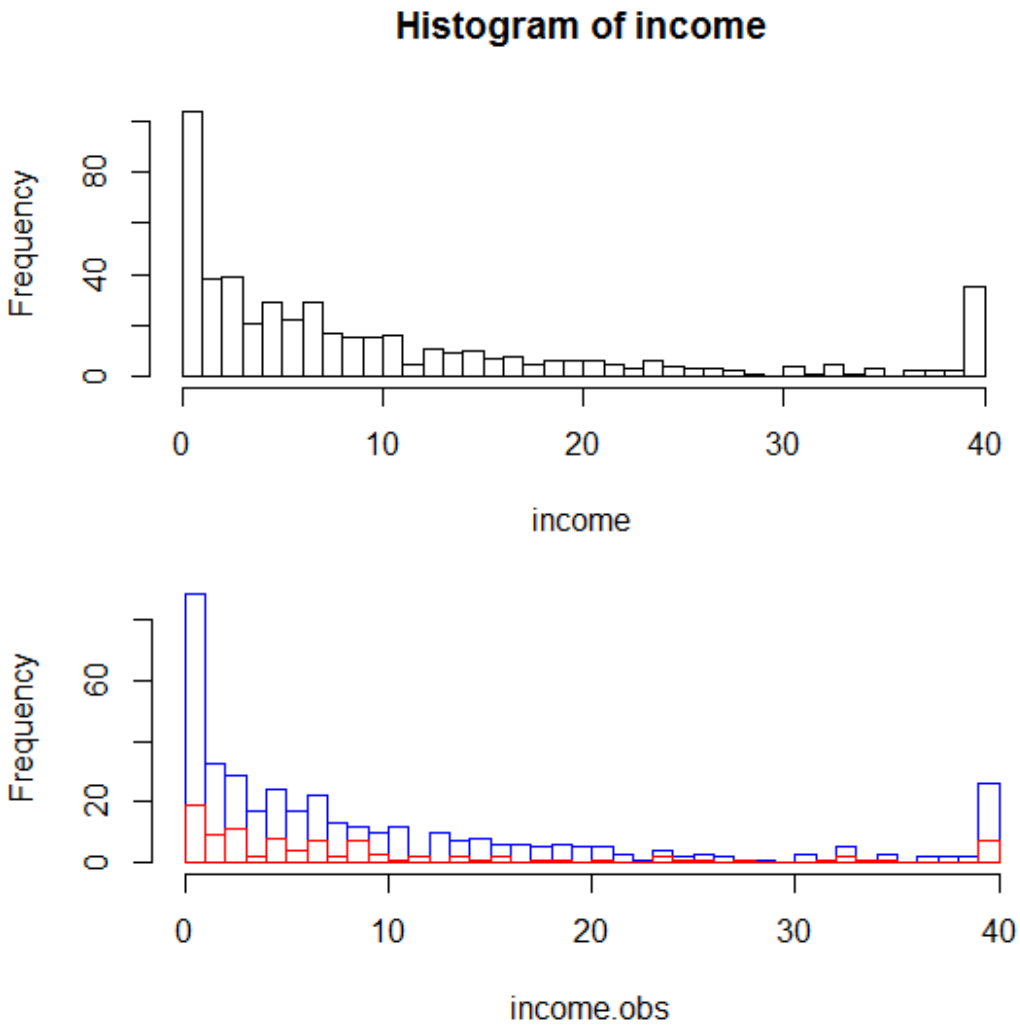
```
obs.sample = var1[!is.na(var1)]
n.missing = sum(is.na(var1))
var1[is.na(var1)] = sample(obs.sample, n.missing, replace=TRUE)
```

Drawing from the empirical distribution

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	?	7	3
0	?	6	?
1	5	?	?
1	3	?	?
1	4	9	?

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	5	7	3
0	3	6	3
1	5	8	2
1	3	6	2
1	4	9	3

Graphical illustration



Last value carried forward (LVCF)

X_{pre}	X_{post}
3	8
4	9
2	7
7	6
5	?
3	?
4	?

$\mathrm{X}_{\mathrm{pre}}$	$\mathrm{X}_{\mathrm{post}}$
3	8
4	9
2	7
7	6
5 →	5
3 →	3
4 →	4

Last value carried forward (LVCF)

- Requires longitudinal data, where attrition is the primary reason for missing data.
- In studies with pre-test/post-test type measures (or the same questions asked over repeated time periods), missing post-test values are filled in with the pre-test value (or observed value from most recent time point in past)
- This is often assumed to be conservative, but it can still lead to bias even under MCAR, and increased Type I errors (Lavori, 1992).
- Also will lead to underestimated standard errors

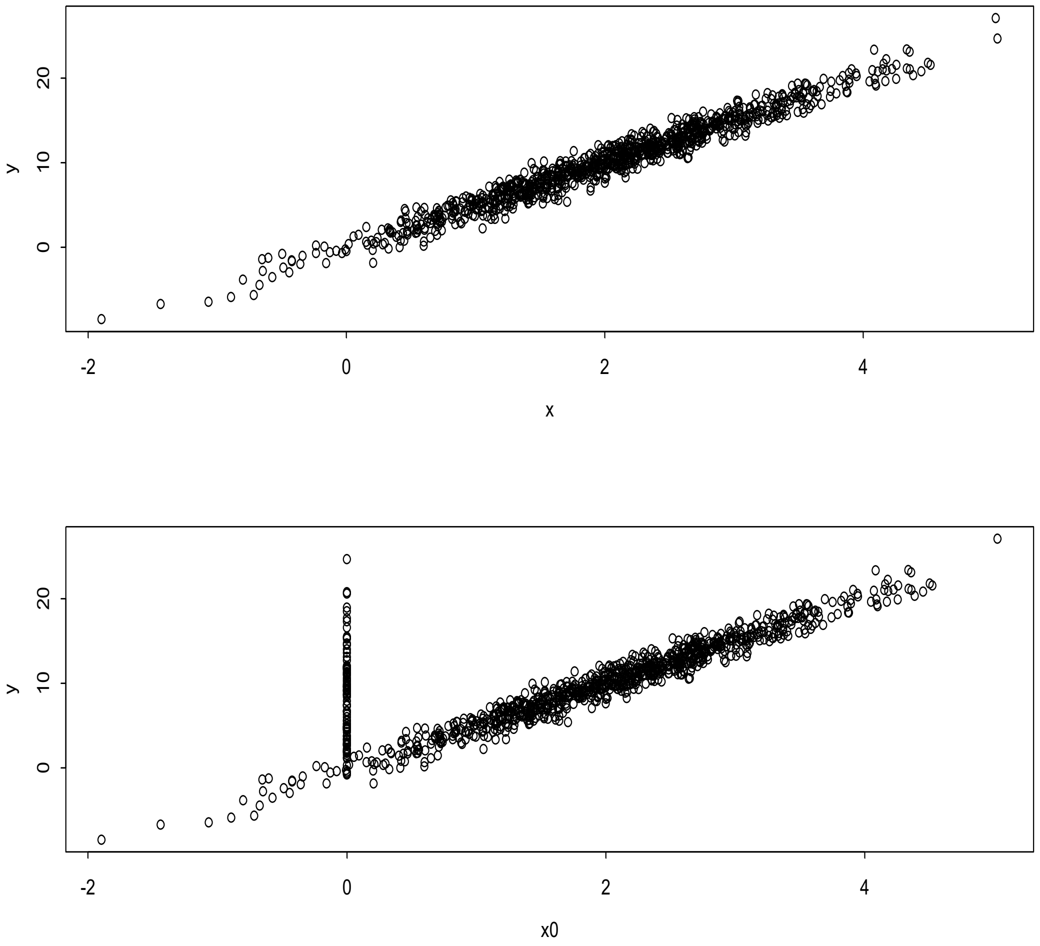
Next value carried backward (NVCB)

- If early observations are missing, this method carries subsequent values backwards.
- Helps when subject fails to complete baseline information, or a long buildup to the study leads to incomplete data.
- Both LVCF and NVCB are within-subject imputations and use only observed data – they are not model based procedures.
- Data augmentation and multiple imputation are model based procedures.

Dummy variable adjustment

- For each *predictor* variable with missing data this method:
  - Plug in missing values with some arbitrary number (0’s or the mean) and then
  - include a new variable that is an indicator for missingness for the original variable with missing data
- This approach keeps cases in that would otherwise be dropped.
- “the method generally produces biased estimates of the coefficients.” (see Allison for details)
- Adding an interaction between the missingness dummy and other variables can help with the bias - leads to similar estimates as complete cases (see Jones, JASA, 1996) which is also problematic but generally less biased

Dummy variables example



Dummy variables: example

- Suppose we have data: *y* (outcome), treat (treatment variable), *x* (covariate) and we're interested in the effect of the treatment on the outcome (*n* =1000)
- The true model for the data is:

$$y = 5x + 0*\text{treat} + e$$

(there is no treatment effect)

- Higher missing data rate for x among controls than treated
- Generating data according to this model with x and treat correlated, estimate using the dummy variable method and got coefficient estimates of:

  4.6 for the x-coefficient

  2.0 for the coefficient on treatment (highly statistically significant!) thus leading one to believe there is a significant treatment effect when in fact none exists

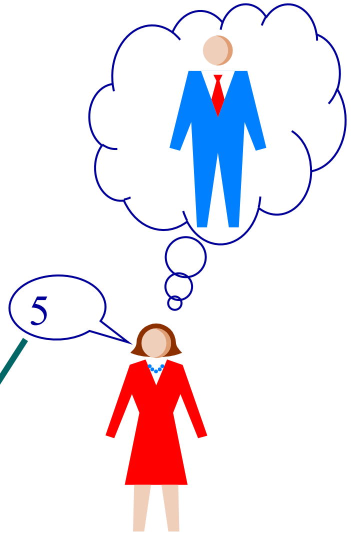
Reports from other people

- Suppose we are missing data regarding the fathers of children in a dataset
- Why not fill these values in with *mother's report* of the values?
- At best will probably have measurement error (added noise) and misestimated standard errors due to single imputation
- At worst could be systematically biased

Reports from other people

X_1	X_2	X_3
0	3	2
0	4	3
0	2	3
1	3	?
1	5	?
1	3	?
1	4	?

X_1	X_2	X_3
0	3	2
0	4	3
0	2	3
1	3	5
1	5	?
1	3	?
1	4	?



Reports from other people

- Example of a discrepancy you might not expect: comparing mothers and fathers reports of their marital status

```
               |   couple's marital
               |  status at 12 month
 mother’s      |     father’s report
 report        |      0 no       1 yes |     Total
---------------+----------------------+----------
    1 married |         27         229 |       256
      2 cohab |        981          48 |     1,029
3 nonresident |      1,022           9 |     1,031
---------------+----------------------+----------
        Total |      2,030         286 |     2,316
```

- Similar things in reports of number of sex acts and proportion of protected sex acts across members of the same couple.

Generating missing data

Why would we ever generate missing data in practice???

Reason 1: You never understand something well until you have to make it yourself from scratch

- When you generate the missing data yourself you have to think concretely about the mechanics of the process
  - This means you have to figure out the "rules" in effect by which that mechanism "plays"
  - When generating MAR missing data:
    - started with complete data
    - generated missing data in var1 given all the other variables
    - generated missing data in var2 given all the other variables
    - etc
- The better you understand how the assumption works, the better you'll be able to figure out whether it is plausible for your data or the particular aspects of it that might be *least* plausible for your data

Reason 2: Someday you may want to test out how well a missing data method works

- A great way to do this is to generate missing data in a complete (fully observed) dataset and then see if the missing data method allows you to recover the "true" values of your desired analysis

Reason 3: Someday you may want to be able to critically evaluate a paper that claims they have a great new missing data method and performs simulations to back it up

- You'll now have the experience to tell whether they are using reasonable missing data mechanisms in their simulations (a surprising number of papers just generate MCAR missing data!)

One way to generate missing data: Brief review of logistic regression

Missing data patterns and logistic regression

- Since we were generating our missing data patterns one variable at a time we used models for binary data (because values are either observed or missing)
- With MCAR this is just a simple coin flip (Binomial model)
- With MAR and NMAR, by definition, the probability that a value is missing depends on other information (that is either observed by the researcher or not) therefore we generate using conditional models
- One of the simplest such models uses the "logit link" to create a mapping between these probabilities and predictor values – this is link used in logistic regression

Recall

Let R be the **matrix** of variables R_1, …, R_p, corresponding to variables in our dataset, X_1, …, X_p, that indicate whether a given value of the corresponding X variable is observed (=1) or missing (=0)

X_1	X_2	X_3	X_4
0	2	5	2
0	1	3	?
0	2	4	3
0	3	5	?
1	5	?	?
1	4	?	12
1	6	11	?
1	6	12	16

R_1	R_2	R_3	R_4
1	1	1	1
1	1	1	0
1	1	1	1
1	1	1	0
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Brief review of logistic regression

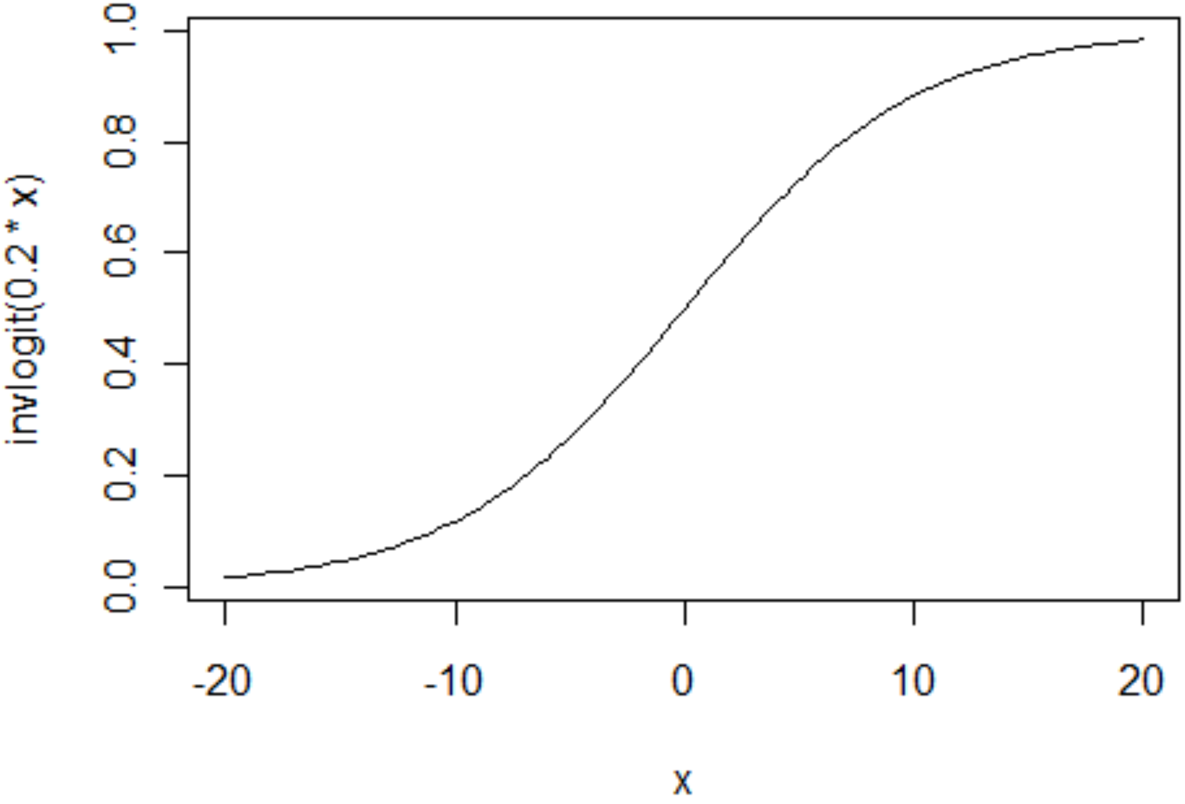
- Let’s review logistic regression a bit…
- For a binary outcome that we’ll denote R we say that

  $\log(\pi/1\text{-}\pi) = \text{logit}(\pi) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots. + \beta_k X_k$,

  where $\pi = \text{E}[R \mid X] = \Pr[R{=}1 \mid X]$
- $\log(\pi/1\text{-}\pi)$ is referred to as the “logit” of π
- To express π as a function of *X*, we say that it equals the *inverse logit* of the linear combination of the covariates,

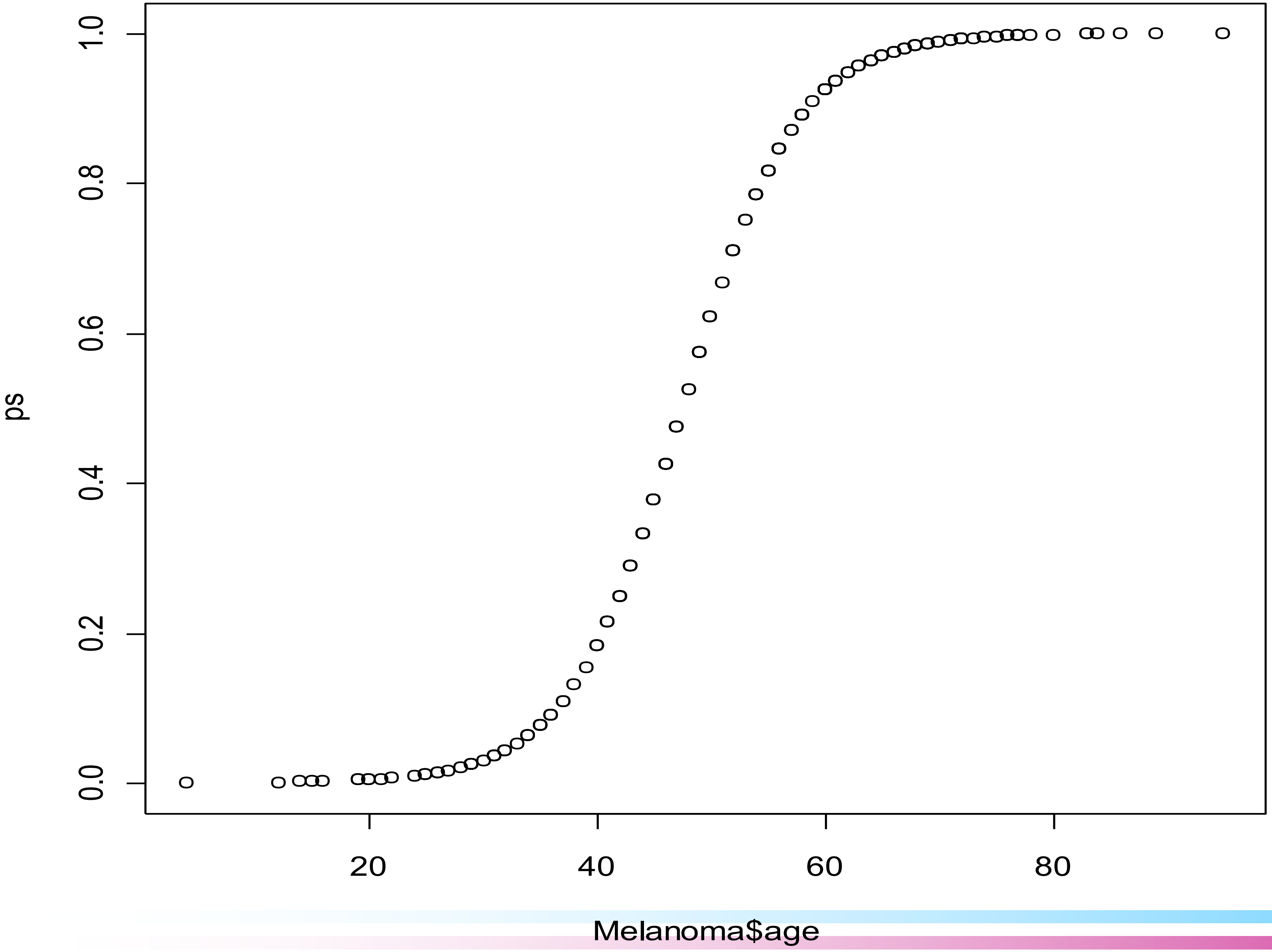
  $\pi = \text{logit}^{-1}(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots. + \beta_k X_k)$
- The invlogit() command takes as input the value of this linear combination and yields the probability [or if you feed it a vector of values, it will output the corresponding vector of probabilities]

$\pi = \text{logit}^{-1}(.2x)$



R code (from arm package): `curve(invlogit(.2*x),-20,20)`

Here's an example of $\pi = \text{logit}^{-1}(.9 + .2\textit{age})$


ps
Melanoma$age