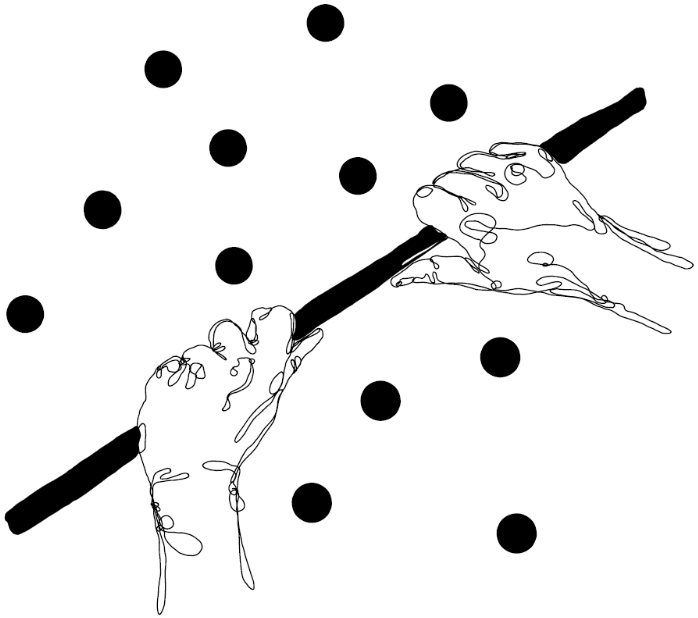
Linear regression

Mabel Carabali

Maybe a bit of Bayesian?

EBOH, McGill University

2024/30/09 (updated: 2024-10-02)



"Regression is the most common way in which we fit a line to explain variation"

The Effect by Nick Huntington-Klein

Expected competencies

- Knows how/when to use linear regression (LR) models.
- Can describe the LR model, assumptions, and implications.
- Can explain why its called OLS and the estimates least squares estimates.
- Can define regression line, fitted value, residual, and influence.
- Can state the relationships between:
  - Correlation and regression coefficients.
  - The two-sample t-test and a regression model with one binary predictor.
  - ANOVA and a regression model with categorical predictors.
- Knows how statistical packages estimate the parameters & make diagnostic plots.
- Can interpret regression model outputs (even transformed).

Objectives

1. Revise basic OLS a.k.a. Liner regression concepts

2. Learn how to formulate, code and interpret LR models

3. Identify opportunities to use advanced LR models

Recap! (1)

Continuous Outcomes, Variables and Line Fitting

- Conditional distributions
- Conditional means
- Line Fitting → ***Regression***
  - "the normal linear model, assuming that the *mean* of the response depends on the explanatory variables via a linear function"
  - Ordinary Least Square (OLS)
- Intercepts, Slopes
- Conditional conditional means a.k.a. "Control" or "Adjustment"

Recap! (2)

What's the Normal linear regression model?

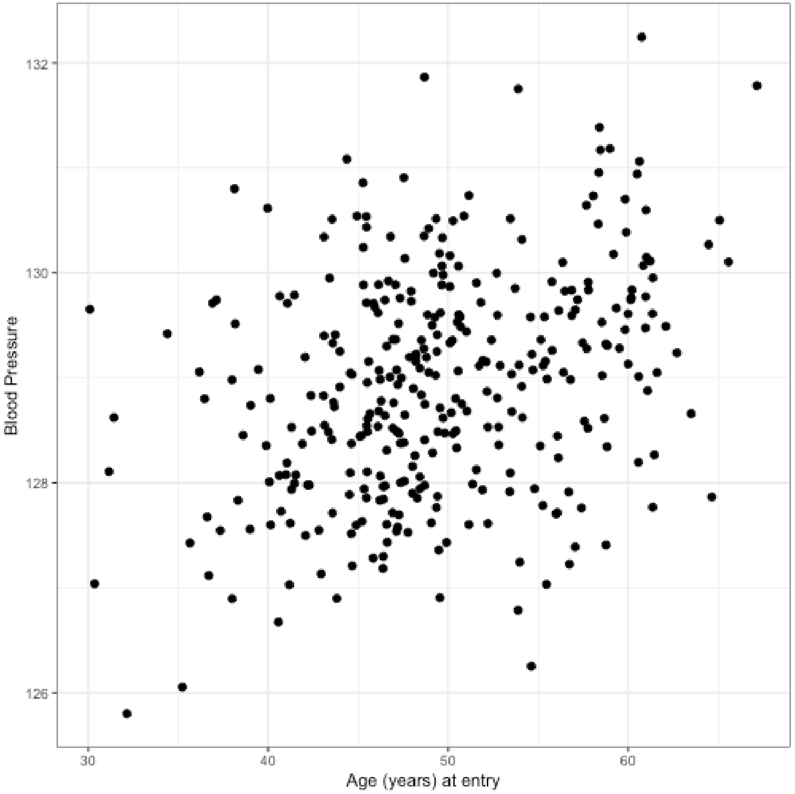
- Normal probability distribution (i.e., Gaussian distribution)
- Relationship between an outcome variable `(Y)` , assumed to be normally distributed, and one or more explanatory variables `(X)` about which *no distributional assumptions are made*. Referred to as 'the general linear model' (GLM).
- **Simple linear regression:** assumes a linear relationship between the response (outcome) and explanatory variables.

- The linear model states that the response Y is generated as a linear combination of the Xs plus a random error, ϵ_i:

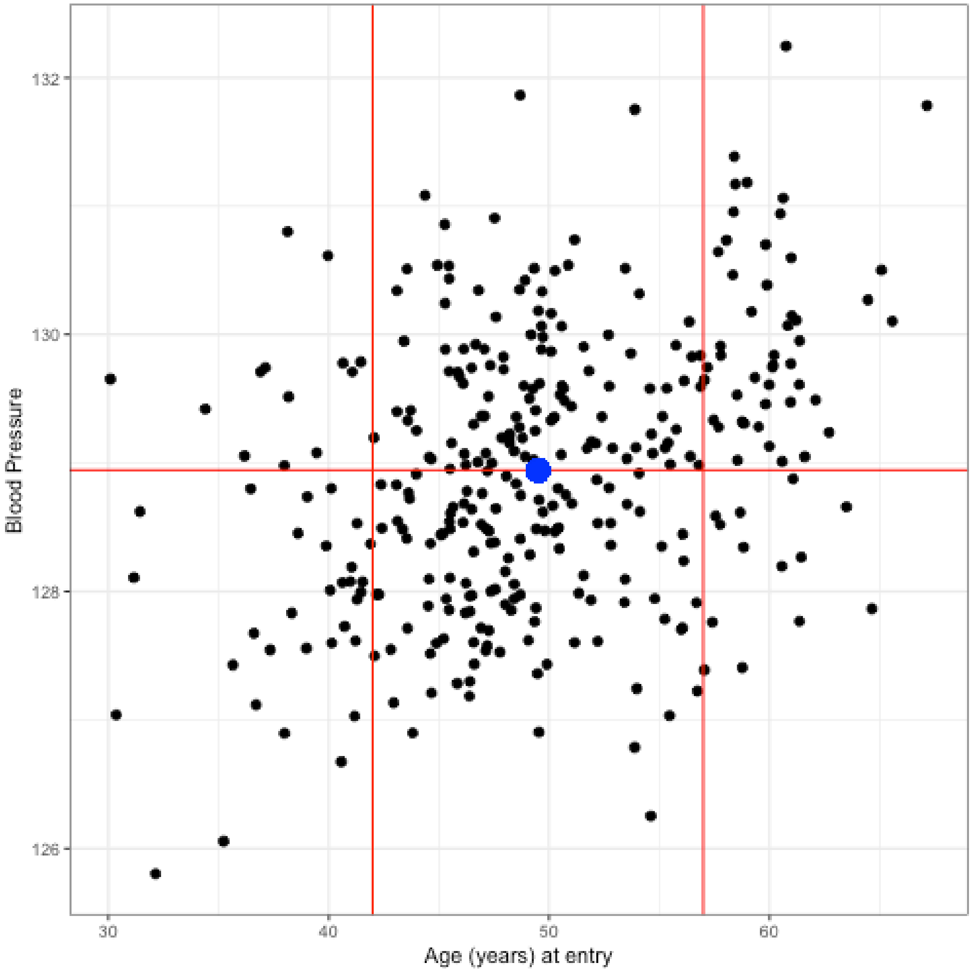
$$y_i = \beta_0 + \beta_1 x_{1i} + \epsilon_i$$

- ϵ_i (s) are assumed to be normally distributed and independent with mean 0 and a common variance σ^2.
- The model is a model for the **conditional** distribution of Y given X.

What would be the correlation between Age and Blood Pressure?



Does this help?



Methods for correlation analyses

- Pearson correlation (r) - measures a **linear dependence** between two variables (x and y) when both are from **normal distribution**, to determine normality:

  i) `shapiro.test()`

  ii) normality plot (`ggpubr::ggqqplot()`)

- Kendall *tau* and Spearman *rho* are rank-based correlation coefficients (non-parametric test)

Pearson correlation formula

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}}$$

where m_x and m_y are the means of x and y variables

p-value of the correlation determined from the ***t*** value

$$t = r\sqrt{\frac{n-2}{1-r^2}} with \ \ df = n - 2$$

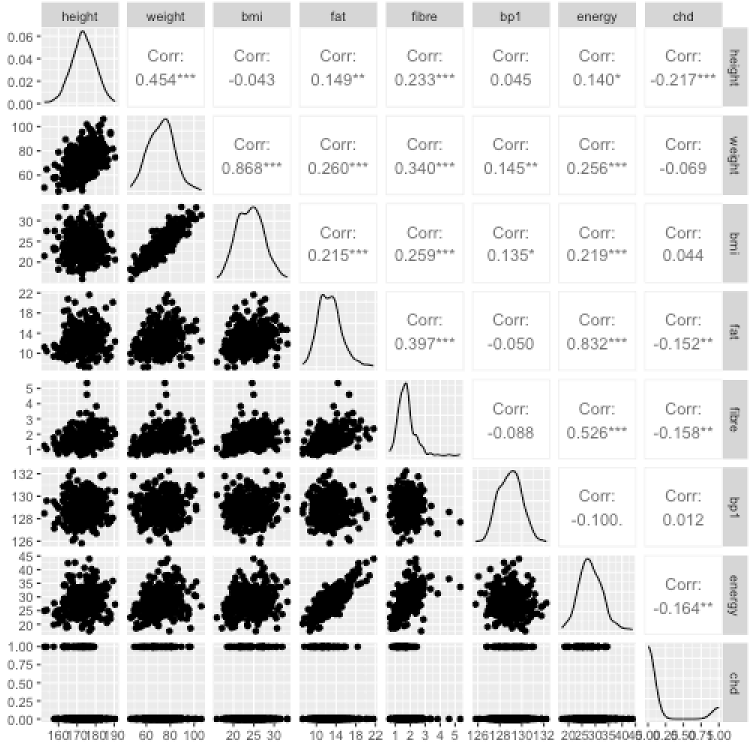
where n = number of observation in x and y variables

The correlation $r \rightarrow$ -1 < r < 1, no correlation r = 0

Analytical solution using R

	Overall (N=337)
height	
Mean (SD)	173 (6.36)
Median [Min, Max]	173 [152, 191]
weight	
Mean (SD)	72.5 (10.7)
Median [Min, Max]	72.8 [46.7, 106]
bmi	
Mean (SD)	24.1 (3.19)
Median [Min, Max]	24.1 [15.9, 33.3]
fat	
Mean (SD)	12.7 (2.37)
Median [Min, Max]	12.6 [7.26, 21.6]
fibre	
Mean (SD)	1.72 (0.562)
Median [Min, Max]	1.67 [0.605, 5.35]
Missing	4 (1.2%)
bp1	
Mean (SD)	129 (1.08)
Median [Min, Max]	129 [126, 132]
energy	
Mean (SD)	28.3 (4.42)
Median [Min, Max]	28.0 [17.5, 44.0]
factor(chd)	
0	291 (86.4%)
1	46 (13.6%)

```
df1<-df %>% select(height, weight,
bmi, fat, fibre , bp1, energy, chd )
GGally::ggpairs(df1)
```



Analytical solution using R

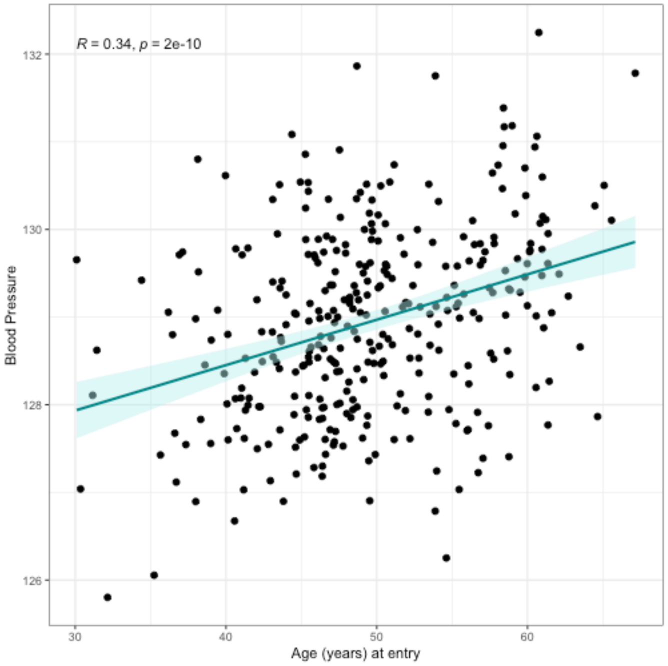
```r
pcor <- cor.test(df$ageye, df$bp1,
        method = "pearson")
pcor
```

```
##
##      Pearson's product-moment correlation
##
## data:  df$ageye and df$bp1
## t = 6.5624, df = 335, p-value = 2.013e-10
## alternative hypothesis: true correlation is not equ
## 95 percent confidence interval:
##  0.2392982 0.4288759
## sample estimates:
##       cor
## 0.3375049
```

```r
# confirm t value with hand calculation
tval <- pcor$estimate*sqrt(335/(1-pcor$esti
names(tval) <- c("");
tval
```

```
##
## 6.562412
```

```r
ggpubr::ggscatter(df, x = "ageye", y = "bp1
add = "reg.line", conf.int = TRUE,
add.params = list(color = "#008B8B", fill =
cor.coef = TRUE, cor.method = "pearson",
xlab = "Age (years) at entry", ylab = "Bloo
theme_bw()
```



Analytical solution using R

```
temp1 <- cor.test(df$ageye, df$bp1,
         method = "kendall");
temp1
```

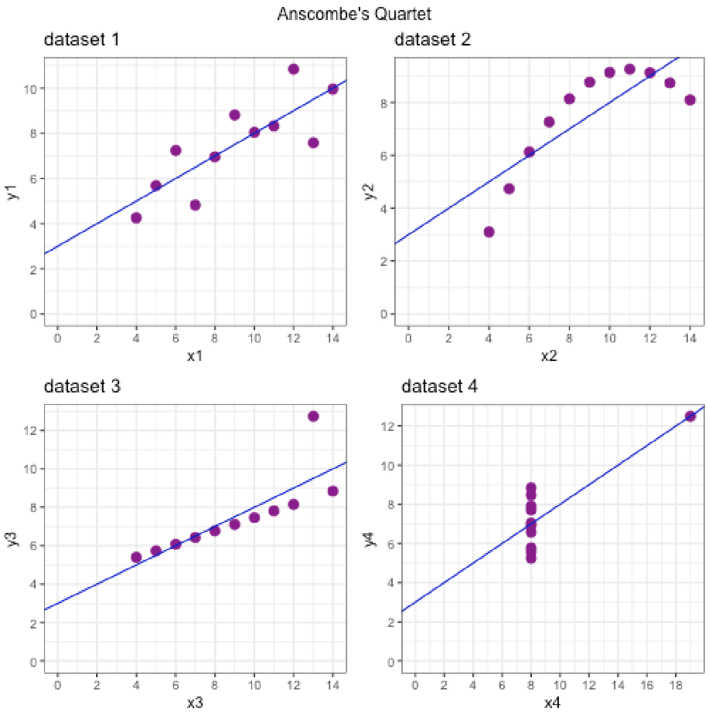
```
##
##     Kendall's rank correlation tau
##
## data:  df$ageye and df$bp1
## z = 5.9674, p-value = 2.41e-09
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.2178494
```

```
temp2 <- cor.test(df$ageye, df$bp1,
         method = "spearman")
temp2
```

```
##
##     Spearman's rank correlation rho
##
## data:  df$ageye and df$bp1
## S = 4369150, p-value = 3.355e-09
## alternative hypothesis: true rho is not equal to
## sample estimates:
##       rho
## 0.3150446
```

Data visualization as a tool

Consider 4 separate datasets:



which given the same LR results... **Would you assume the data sets are the same?**

which given the same LR results **Would you assume the data sets are the same?**

```
md0<-lm(y1 ~ x1, data=anscombe)
round(summ(md0, confint = T)$"coeftable", 2
```

```
##               Est. 2.5% 97.5% t val.    p
## (Intercept)    3.0 0.46  5.54   2.67 0.03
## x1             0.5 0.23  0.77   4.24 0.00
```

```
md1<-lm(y2 ~ x2, data=anscombe)
round(summ(md0, confint = T)$"coeftable", 2
```

```
##               Est. 2.5% 97.5% t val.    p
## (Intercept)    3.0 0.46  5.54   2.67 0.03
## x1             0.5 0.23  0.77   4.24 0.00
```

```
md2<-lm(y3 ~ x3, data=anscombe)
round(summ(md0, confint = T)$"coeftable", 2
```

```
##               Est. 2.5% 97.5% t val.    p
## (Intercept)    3.0 0.46  5.54   2.67 0.03
## x1             0.5 0.23  0.77   4.24 0.00
```

```
md3<-lm(y4 ~ x4, data=anscombe)
round(summ(md0, confint = T)$"coeftable", 2
```

```
##               Est. 2.5% 97.5% t val.    p
## (Intercept)    3.0 0.46  5.54   2.67 0.03
## x1             0.5 0.23  0.77   4.24 0.00
```

Use `??anscombe` to know about the datases

Line Fitting

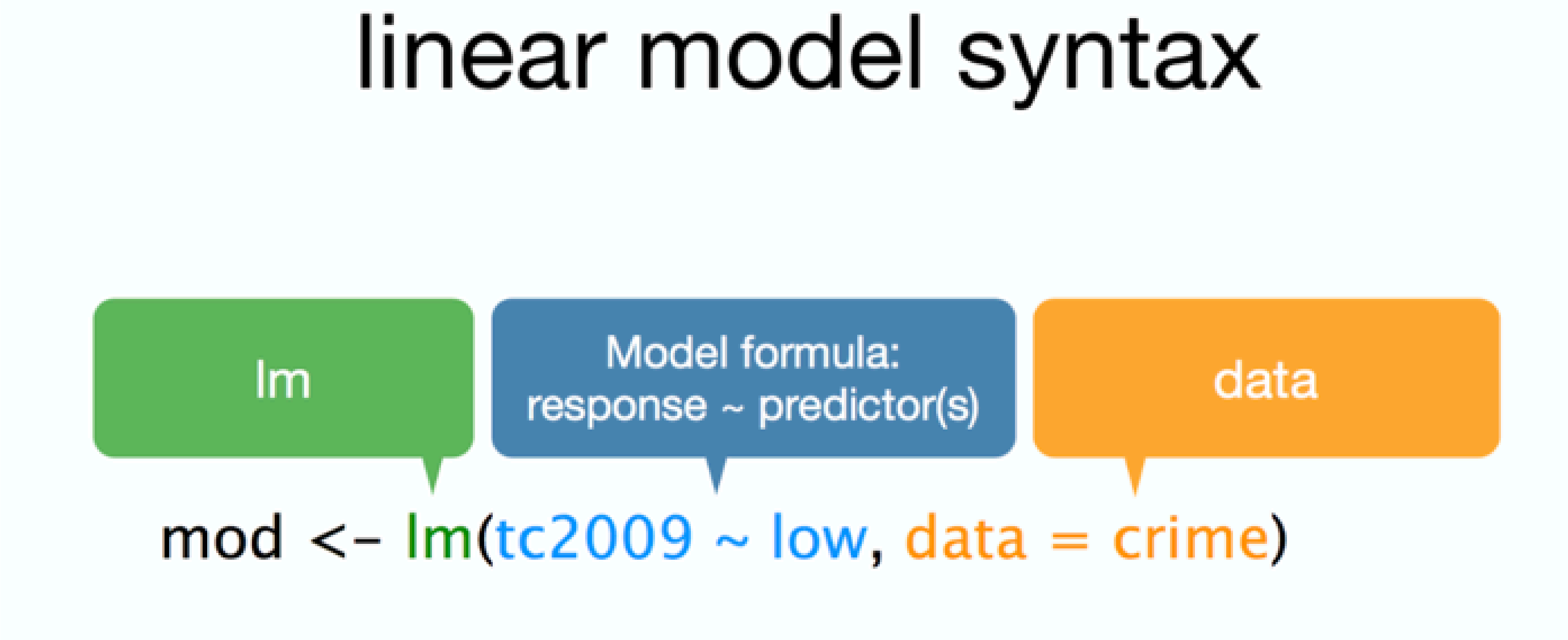
The *deterministic* relationship between X and Y, but IRL phenomena are *stochastic* or *probabilistic*

- "*Showing the mean of Y among local values of X is valuable, and can produce a highly detailed picture of the relationship between X and Y.* "

- Explore estimates of the mean Y conditional on values of X, with an assumed **shape**, often a **straight line**.

The most common way of address this is witht the REGRESSION

Regression with R



linear model syntax

lm

Model formula:
response ~ predictor(s)

data

```
mod <- lm(tc2009 ~ low, data = crime)
```

The same syntax is used for all R models (Poisson, logistic, Cox, etc).

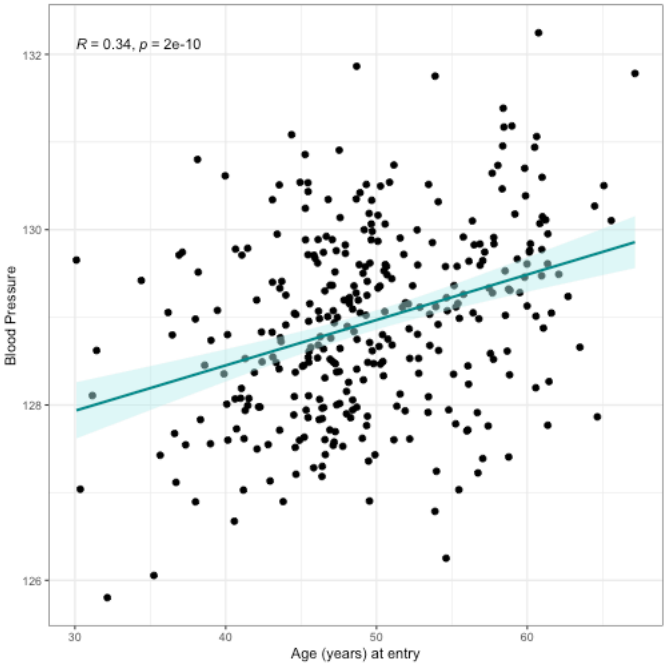
Line Fitting (Example)

$$Y = \beta_0 + \beta_1 X$$

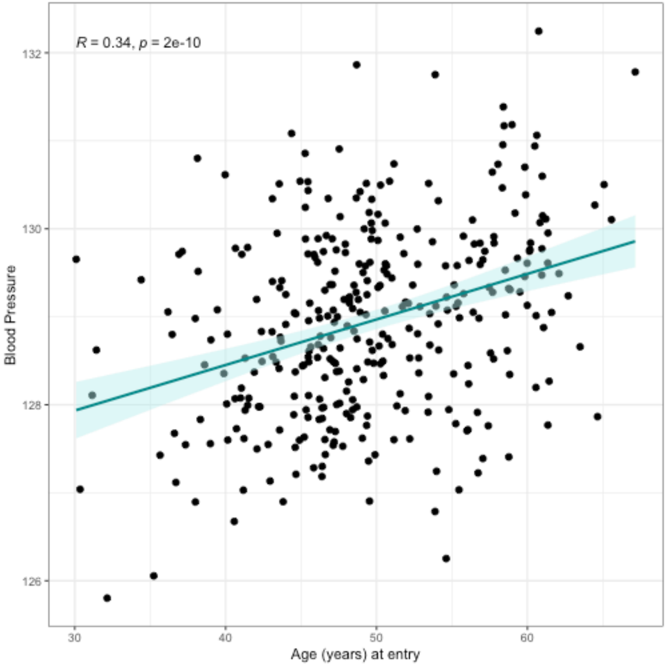
```
mod0<- lm(bp1~ ageye, data = df)
round(summ(mod0, confint = T)$"coeftable",
```

```
##                  Est.   2.5%  97.5% t val. p
## (Intercept) 126.38 125.60 127.16 320.35 0
## ageye         0.05   0.04   0.07   6.56 0
```

$$Y = 126.4 + 0.05X$$



Line Fitting (Example)



- $Y = \beta_0 + \beta_1 X$

- The mean of Y conditional on, $X = 30$ is:

$$Y = 126.4 + 0.05(30)$$

= 127.9

- The mean of Y conditional on a given value of X would be **0.05** higher if you instead made it conditional on a value of one unit higher.

The statistical properties of OLS

Ordinary Least Squares (OLS) is the most well-known application of line-fitting.

- OLS picks the line that gives the lowest sum of squared residuals.
    - Residual: difference between an observation's actual value and the conditional mean assigned by the line.

We determined that the conditional mean of Y when $X = 30$ is $126.4 + 0.05(30)$ = 127.9 , but what if we observe X =30 and Y = 130.5?

- OLS $\rightarrow$ squared the difference of the observed and assigned/expected Y and adds all the prediction in the data.
- Selects values on β_0 and β_1 in the line $Y = \beta_0 + \beta_1 X$ that makes that sum of squared residuals as small as possible.

What we know about the OLS/LR

- Uses X to explain or predict Y
- OLS/LR gives the "best linear approximation" of the relationship between X and Y

- Pro: Efficient use of variation
- Pro: Straightforward explanation
- Con: We may lose some important variation
- Con: If we choose the wrong shape for the relationship, results aren't valid

- In an univariate/bivariate regression, the β_1 a.k.a. slope is the covariance of X, Y divided by the variance of X.

```
round((cov(df$ageye, df$bp1))/var(df$ageye),2)
```

```
## [1] 0.05
```

Assumptions ordinary linear regression

1. A linear relationship between the independent and dependent variable
2. Independent errors
3. Normal distribution of errors
4. Homoscedasticity

The only thing that changes with Bayesian linear regression, is that instead of using MLE to find point estimates for the parameters, we treat them as random variables, assign priors for them, and use Bayes theorem to derive the posterior distribution.
So Bayesian model inherits these same assumptions, since it"s all about the likelihood

Basically, we are assuming that the likelihood function we've chosen is a reasonable representation of the data

Basic regression model assumptions (Mathematical)

- Developing a probabilistic model for linear regression with additive Gaussian errors $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$
- Note, $E[Y_i \mid X_i = x_i] = \mu_i = \beta_0 + \beta_1 x_i$ (linear relationship)
- Here the ϵ_i are assumed iid $N(0, \sigma^2)$ (independent errors, normally distributed)
- Note, $Var(Y_i \mid X_i = x_i) = \sigma^2$ (variance assumed constant - homoscedasticity)
- Likelihood equivalent model specification is that the Y_i are independent $N(\mu_i, \sigma^2)$
- Least squares is an estimation tool

The error term

- There's going to be a difference between the line that we fit and the observation we get.

Hence, $Y = \beta_0 + \beta_1 X \rightarrow Y = \beta_0 + \beta_1 X + \epsilon_i$

- **Residual:** difference between the prediction we make with our fitted line and the actual value.

- **Error:** difference between the true best fit-line and the actual value.
    - The error effectively contains everything that causes Y that is not included in the model.

Sampling variation

If we want to say that our OLS estimates of β_1 will, on average, give us the population β_1, then it must be the case that X is uncorrelated with ϵ

- *Regression coefficients are estimates, and even though there's a true population model out there, the estimate we get varies from sample to sample due to sampling variation.*

What is that normal distribution that the OLS coefficients follow?

- In $Y = \beta_0 + \beta_1 X + \epsilon_i$, the coefficient $\beta_1 \sim N(\beta_1, \sqrt{\sigma/(var(X)n)})$

- n= nb of observations;

- σ= is the SD of ϵ;
- and the variance of X is $var(X)$

How to reduce an OLS estimate's sampling variation?

(1) Shrink the SD of the error term σ, i.e., make the model predict Y more accurately.

(2) Pick an X with large variation

- an X that changes a lot makes it easier to check for whether Y is changing in the same way.

(3) Use a big sample so n gets big.

How do we call this *standard deviation of the error*?

Standard Error

Likelihood

$$\mathcal{L}(\beta, \sigma) = \prod_{i=1}^{n} \left\{ (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mu_i)^2 \right) \right\}$$

so that the twice the negative log (base e) likelihood is

$$-2\log\{\mathcal{L}(\beta, \sigma)\} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \mu_i)^2 + n\log(\sigma^2)$$

Discussion

- Maximizing the likelihood is the same as minimizing -2 log likelihood
- The least squares estimate for $\mu_i = \beta_0 + \beta_1 x_i$ is exactly the maximum likelihood estimate (regardless of σ)

Interpreting the intercept

- β_0 is the expected value of the response when the predictor is 0

$$E[Y|X=0] = \beta_0 + \beta_1 \times 0 = \beta_0$$

- Note, this isn't always of interest, for example when $X = 0$ is impossible or far outside of the range of data. (X is blood pressure, or height etc.)
- Consider that

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i = \beta_0 + a\beta_1 + \beta_1(X_i - a) + \epsilon_i = \tilde{\beta}_0 + \beta_1(X_i - a) + \epsilon_i$$

  So, shifting you X values by value a changes the intercept, but not the slope.
- Often a is set to $\bar{X}$ so that the intercept is interpreted as the expected response at the average X value.

Interpreting the slope

- β_1 is the expected change in response for a 1 unit change in the predictor

$$E[Y \mid X = x+1] - E[Y \mid X = x] = \beta_0 + \beta_1(x+1) - (\beta_0 + \beta_1 x) = \beta_1$$

- Consider the impact of changing the units of X.

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i = \beta_0 + \frac{\beta_1}{a}(X_i a) + \epsilon_i = \beta_0 + \tilde{\beta}_1(X_i a) + \epsilon_i$$

- Therefore, multiplication of X by a factor a results in dividing the coefficient by a factor of a.
- **Example:** X is height in m and Y is weight in kg. Then β_1 is kg/m.

- Converting X to cm implies multiplying X by $100cm/m$. To get β_1 in the right units, we have to divide by $100cm/m$ to get it to have the right units.
$Xm \times \frac{100cm}{m} = (100X)cm \text{ and } \beta_1 \frac{kg}{m} \times \frac{1m}{100cm} = \left(\frac{\beta_1}{100}\right)\frac{kg}{cm}$

Interpretation

```
## MODEL INFO:
## Observations: 337
## Dependent Variable: bp1
## Type: OLS linear regression
##
## MODEL FIT:
## F(1,335) = 43.07, p = 0.00
## R² = 0.11
## Adj. R² = 0.11
##
## Standard errors:OLS
## ------------------------------------------------------------
##                       Est.     2.5%    97.5%   t val.      p
## ----------------- -------- -------- -------- -------- ------
## (Intercept)         126.38   125.60   127.16   320.35   0.00
## ageye                 0.05     0.04     0.07     6.56   0.00
## ------------------------------------------------------------
```

What's the relationship between slope in LR and Pearson's *r*?

- $E[Y \mid X = x] = \beta_0 + \beta_1 x$

- $Var(Y \mid X = x) = \sigma^2$

- ML estimates of β_0 and β_1 are the least squares estimates
  $\hat{\beta}_1 = Cor(Y, X)\frac{Sd(Y)}{Sd(X)} \quad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$

```
# standardize x & y, can do manually or easier with scale function
# perform LR with standardized data
df_std <- df %>% mutate(across(where(is.numeric), scale))
mod1 <- lm( bp1~ageye, df_std); #tidy(mod1)
round(summ(mod1)$"coeftable", 2)
```

```
##             Est. S.E. t val. p
## (Intercept) 0.00 0.05   0.00 1
## ageye       0.34 0.05   6.56 0
```

Once data is standardized then r = slope

Simple regression models and Difference in Means

Estimating the mean is the same as regressing on a constant term

Generate some fake data & calculate the mean and std error

```
set.seed(12345)
n_0 <- 40
y_0 <- rnorm(n_0, 2.0, 5.0)
fake_0 <- data.frame(y_0)
mean(fake_0$y_0)
```

```
## [1] 3.200926
```

```
sd(fake_0$y_0)/sqrt(n_0)
```

```
## [1] 0.8209468
```

Regression on a constant term

```
fit_0 <- lm(y_0 ~ 1, data=fake_0);
print(fit_0)
```

```
##
## Call:
## lm(formula = y_0 ~ 1, data = fake_0)
##
## Coefficients:
## (Intercept)
##       3.201
```

Estimating a difference Equivalent to regressing on an indicator variable

Add new group: 50 observations from N(8.0, 5.0) Calculate the mean difference & std error

```
set.seed(12345)
n_1 <- 50;  y_1 <- rnorm(n_1, 8.0, 5.0)
diff <- base::mean(y_1) - base::mean(y_0)
se_0 <- sd(y_0)/sqrt(n_0)
se_1 <- sd(y_1)/sqrt(n_1);
se <- sqrt(se_0^2 + se_1^2)
print(c(diff, se))
```

```
## [1] 5.696905 1.129249
```

5.7 for the difference and 1.13 for its std error, consistent with the simulation with expected true population difference = 6.0

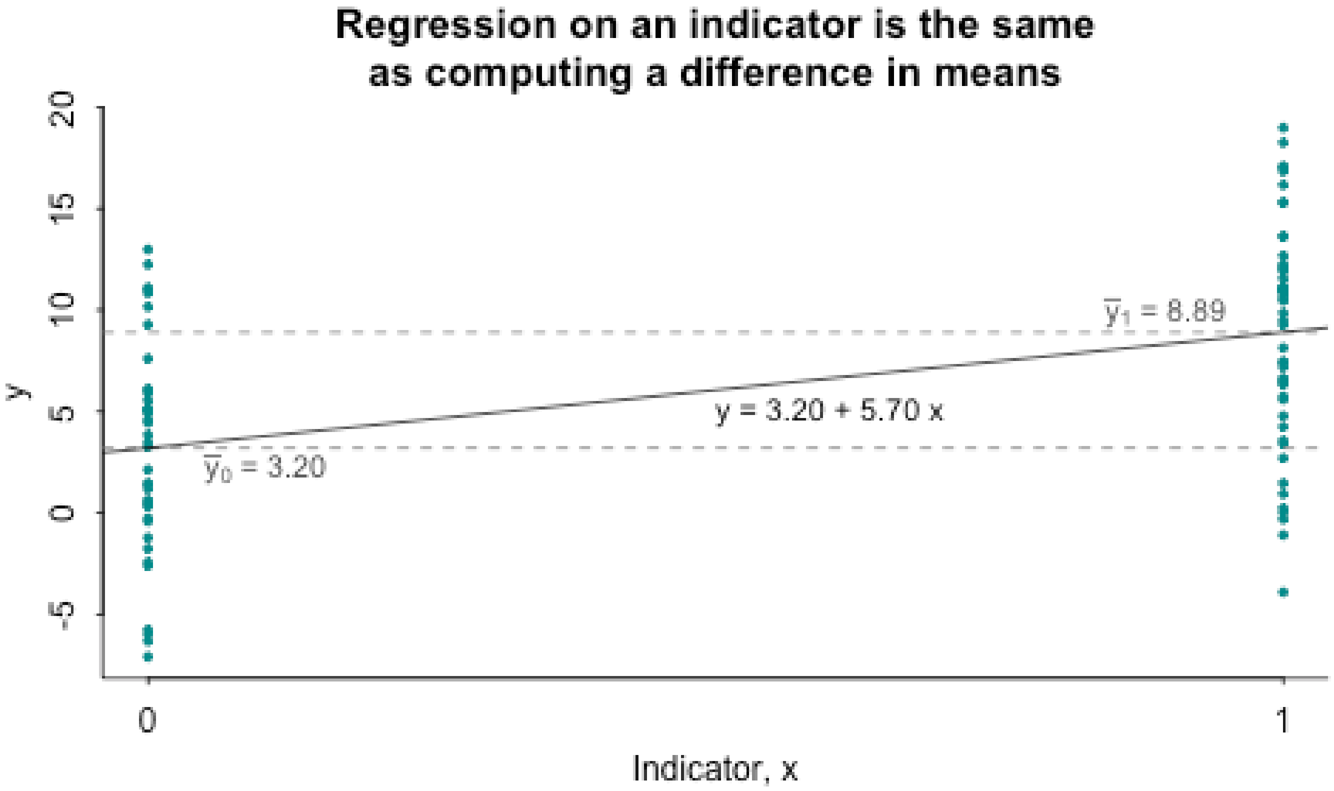
Regression with indicator variable

```
n <- n_0 + n_1; y <- c(y_0, y_1);
x <- c(rep(0, n_0), rep(1, n_1))
fake <- data.frame(x, y)
fit <- lm(y ~ x, data=fake); print(fit)
```

```
##
## Call:
## lm(formula = y ~ x, data = fake)
##
## Coefficients:
## (Intercept)            x
##       3.201        5.697
```

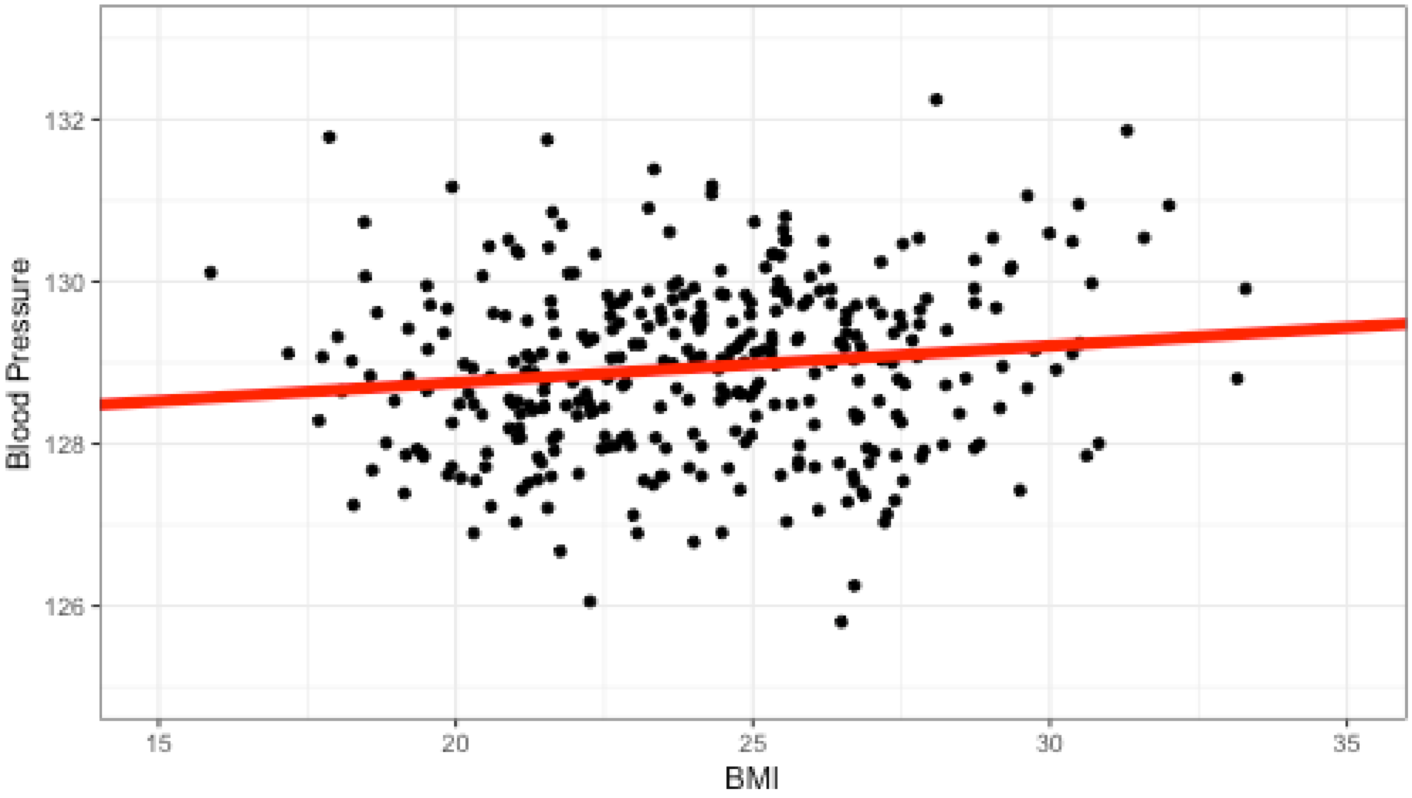
The estimate of the slope, 5.7, is identical to the difference in means, $\bar{y_1} - \bar{y_0}$ and the intercept (3.2) = $\bar{y_0}$

Visual equivalence



- For binary indicator, slope is the average difference in the outcome between the two group
- For continuous variable estimated slope is a weighted average of slopes for every possible pair of 2 points

SBP & BMI, what is your interpretation?



Increase age in BMI, increase Systolic Blood Pressure?

Linear Regression / OLS outputs

```
fit1 <- lm(bp1 ~ bmi, data= df)
summary(fit1)
```

```
##
## Call:
## lm(formula = bp1 ~ bmi, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2488 -0.7917  0.0489  0.7104  3.1221
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 127.84876    0.44371 288.133   <2e-16 ***
## bmi           0.04541    0.01824   2.489   0.0133 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.067 on 335 degrees of freedom
## Multiple R-squared:  0.01816,    Adjusted R-squared:  0.01523
## F-statistic: 6.198 on 1 and 335 DF,  p-value: 0.01328
```

Linear Regression / OLS: Getting a more interpretable intercept

```r
fit2<- lm(bp1 ~I(bmi -mean(bmi)), data=df); summary(fit2)
```

```
##
## Call:
## lm(formula = bp1 ~ I(bmi - mean(bmi)), data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2488 -0.7917  0.0489  0.7104  3.1221
##
## Coefficients:
##                     Estimate Std. Error  t value Pr(>|t|)
## (Intercept)        128.94386    0.05812 2218.636   <2e-16 ***
## I(bmi - mean(bmi))   0.04541    0.01824    2.489   0.0133 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.067 on 335 degrees of freedom
## Multiple R-squared:  0.01816,    Adjusted R-squared:  0.01523
## F-statistic: 6.198 on 1 and 335 DF,  p-value: 0.01328
```

The intercept is now the average SBP at the average BMI. The slope is unchanged and represents how much the SBP changes for a 1 unit increase in BMI.

Interpreting regression results

```r
summ(fit2, confint = T)
```

```
## MODEL INFO:
## Observations: 337
## Dependent Variable: bp1
## Type: OLS linear regression
##
## MODEL FIT:
## F(1,335) = 6.20, p = 0.01
## R² = 0.02
## Adj. R² = 0.02
##
## Standard errors:OLS
## --------------------------------------------------------------------
##                              Est.     2.5%    97.5%    t val.      p
## ------------------------ -------- -------- -------- --------- ------
## (Intercept)                128.94   128.83   129.06   2218.64   0.00
## I(bmi - mean(bmi))           0.05     0.01     0.08      2.49   0.01
## --------------------------------------------------------------------
```

R-squared

```r
(var(df$bp1) - summary(fit2)$sigma^2)/var(df$bp1)
```

```
## [1] 0.01523312
```

```r
cor(df$bp1, df$bmi)^2
```

```
## [1] 0.01816397
```

> *Bluntly speaking, the R-squared is comparing a meaningful measure (residual variation) to a meaningless one (the total variation), and therefore it becomes meaningless, and so should be avoided.*

t-statistic (coefficient divided by the standard error); t-distribution with n - 2 degrees of freedom. NHT, $H_0 : \beta_1$ = 0

F-statistic: A statistic for NHT, H_0 = all the coefficents in the model (except the intercept/constant) = 0, at once, and tests how unlikely results are given that null.

Standardized variables

```r
fit3 <- lm(bp1 ~ bmi + ageye + fat + fibre, data= df)
#summary(fit3)
round(summ(fit3)$"coeftable", 3)
```

```
##                Est.  S.E.  t val.     p
## (Intercept) 125.376 0.640 195.841 0.000
## bmi           0.058 0.018   3.224 0.001
## ageye         0.052 0.008   6.546 0.000
## fat          -0.012 0.025  -0.472 0.637
## fibre        -0.155 0.109  -1.429 0.154
```

How to make regression coefficients comparable?

- "Scale" coefficients by the (study) population standard deviation ($X/sd(X)$)
- Effects interpretable as effects per population standard deviation, i.e., the change in Y per 1 population standard deviation.

Standardized variables

```
fit3a <- lm(bp1 ~ bmi + I(ageye/ sd(ageye) ) + I(fat/sd(fat, na.rm=T)) +
              I(fibre/sd(fibre, na.rm=T)), data= df)
#summary(fit3a)
round(summ(fit3a)$"coeftable", 3)
```

```
##                                  Est.  S.E.  t val.     p
## (Intercept)                   125.376 0.640 195.841 0.000
## bmi                             0.058 0.018   3.224 0.001
## I(ageye/sd(ageye))              0.365 0.056   6.546 0.000
## I(fat/sd(fat, na.rm = T))      -0.028 0.060  -0.472 0.637
## I(fibre/sd(fibre, na.rm = T))  -0.087 0.061  -1.429 0.154
```

- Note that T-statistics and p-values remain the same, only (re)scaled the coefficients

There is no guarantee that the relative sizes of the population sd's of variables are constant across populations. Hence, comparisons of standardized effects do not apply to other study populations.

Predictions from the normal regression model

- If we would like to guess the outcome at a particular value of the predictor, say X, the regression model guesses

$$\hat\beta_0 + \hat\beta_1 X$$

- Note that at the observed value of Xs, we obtain the predictions

$$\hat \mu_i = \hat Y_i = \hat\beta_0 + \hat\beta_1 X_i$$

- Remember that least squares minimizes

$$\sum_{i=1}^n (Y_i - \mu_i)$$

  for μ_i expressed as points on a line

Predicting some SBP as a function of BMI

Numerical predictions

1. Select the values of X to predict values of Y
2. Use the OLS regression results to estimate the prediction
    - Substituting into the equation
    - Using the `predict` function in `R`

```r
newx <- c(20, 30, 35)
#predict by substituting into the equation
coef(fit1)[1] + coef(fit1)[2] * newx
```

```
## [1] 128.7569 129.2110 129.4381
```

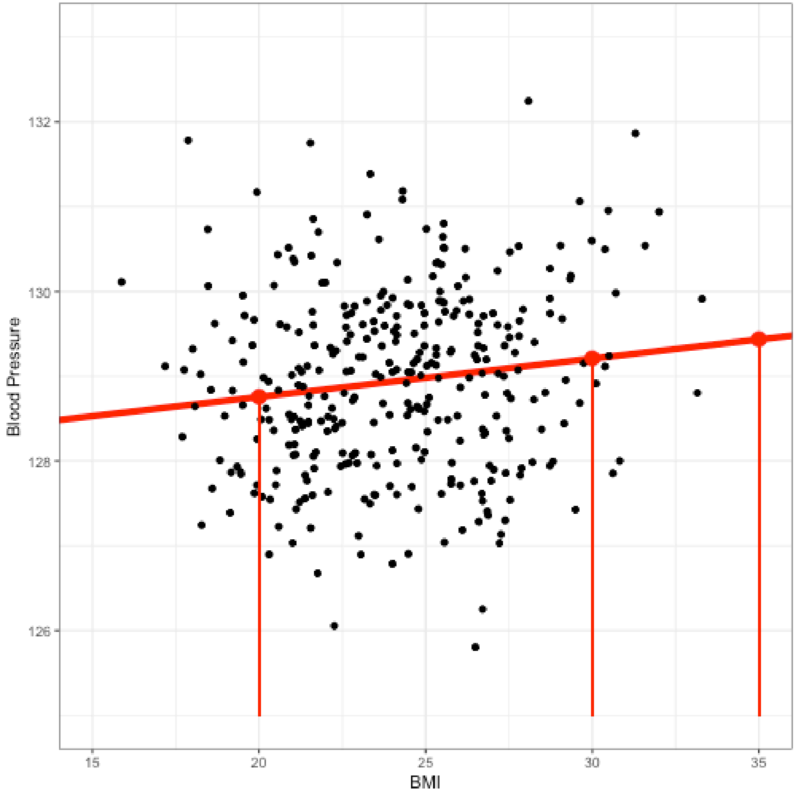
```r
# predict using the predict function
predict(fit1,
        newdata = data.frame(bmi = newx))
```

```
##        1        2        3 
## 128.7569 129.2110 129.4381
```

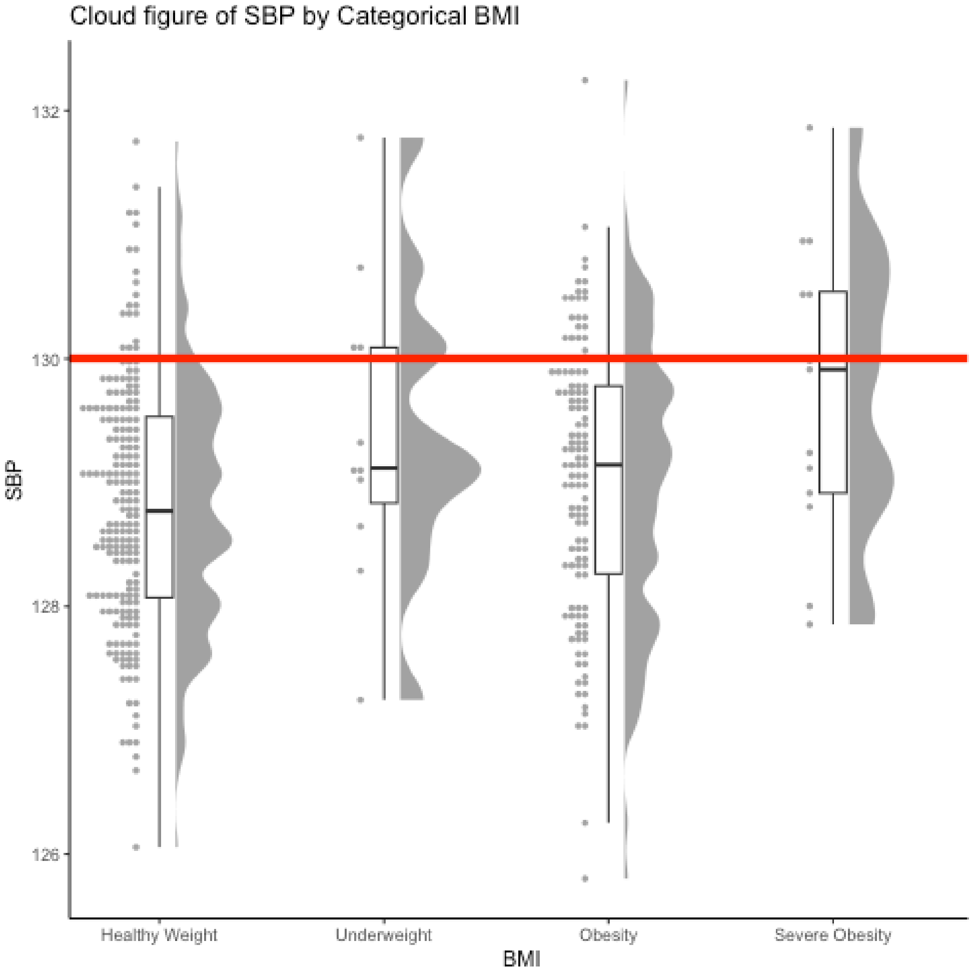
Predicting some SBP as a function of BMI: Graphical display Some R code

```
new_df <- data.frame(bmi= newx, bp1 = predict(fit1, newdata = data.frame(bmi = newx)))

gg_reg_mean + geom_point() +
  geom_point(data=new_df, aes(x=bmi, y=bp1), color="red", size=4) +
  geom_segment(aes(x = 20, y = 125, xend = 20, yend = 128.76), colour = "red") +
  geom_segment(aes(x = 30, y = 125, xend = 30, yend = 129.2), colour = "red") +
  geom_segment(aes(x = 35, y = 125, xend = 35, yend = 129.44), colour = "red")
```



Categorical explanatory variables



Categorical explanatory variables

Using the categorical status

```r
fit4 <- lm(bp1 ~ as.factor(bmicat), data= df)
#summary(fit4)
summ(fit4)
```

```
## MODEL INFO:
## Observations: 337
## Dependent Variable: bp1
## Type: OLS linear regression
##
## MODEL FIT:
## F(3,333) = 4.16, p = 0.01
## R² = 0.04
## Adj. R² = 0.03
##
## Standard errors:OLS
## ---------------------------------------------------------
##                              Est.   S.E.    t val.      p
## ------------------------ -------- ------ --------- ------
## (Intercept)                128.82   0.08   1687.86   0.00
## as.factor(bmicat)1           0.58   0.33      1.76   0.08
## as.factor(bmicat)2           0.19   0.12      1.56   0.12
## as.factor(bmicat)3           0.92   0.30      3.02   0.00
## ---------------------------------------------------------
```

Categorical explanatory variables

Using the categorical status and removing the intercept

```
fit4a <- lm(bp1 ~0 + as.factor(bmicat), data= df)
#summary(fit4a)
summ(fit4a)
```

```
## MODEL INFO:
## Observations: 337
## Dependent Variable: bp1
## Type: OLS linear regression
##
## MODEL FIT:
## F(4,333) = 1245985.42, p = 0.00
## R² = 1.00
## Adj. R² = 1.00
##
## Standard errors:OLS
## ---------------------------------------------------------
##                               Est.   S.E.    t val.      p
## ------------------------ -------- ------ --------- ------
## as.factor(bmicat)0         128.82   0.08   1687.86   0.00
## as.factor(bmicat)1         129.40   0.32    404.76   0.00
## as.factor(bmicat)2         129.01   0.10   1332.90   0.00
## as.factor(bmicat)3         129.74   0.29    441.18   0.00
## ---------------------------------------------------------
```

Collinearity

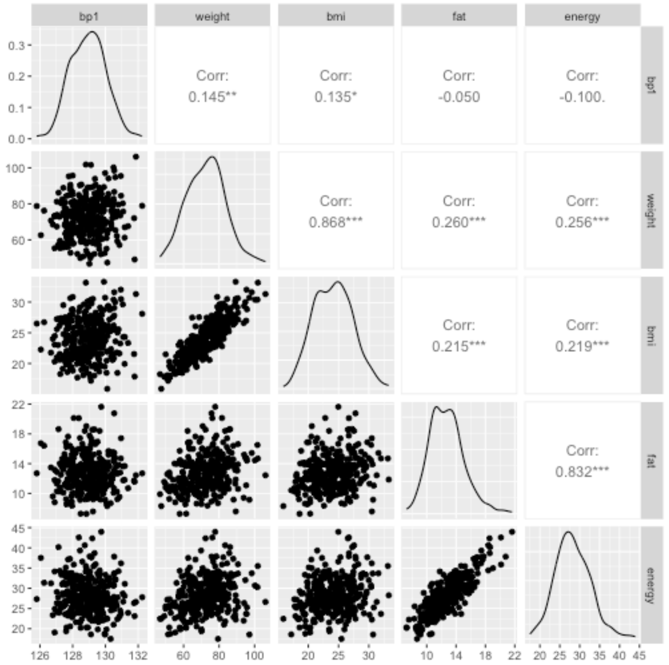
```
fit5 <- lm(bp1 ~ bmi + ageye + fat +
             energy, data= df)
round(summ(fit5)$"coeftable", 3)
```

```
##                Est.  S.E.  t val.     p
## (Intercept) 125.666 0.668 188.107 0.000
## bmi           0.056 0.018   3.202 0.001
## ageye         0.051 0.008   6.424 0.000
## fat           0.028 0.042   0.675 0.500
## energy       -0.033 0.023  -1.487 0.138
```

```
round(summ(fit5, confint = T)$"coeftable",
```

```
##                Est.    2.5%   97.5%  t val.     p
## (Intercept) 125.666 124.352 126.981 188.107 0.000
## bmi           0.056   0.022   0.091   3.202 0.001
## ageye         0.051   0.035   0.066   6.424 0.000
## fat           0.028  -0.054   0.110   0.675 0.500
## energy       -0.033  -0.078   0.011  -1.487 0.138
```

```
df2<-df %>% select(bp1, weight, bmi,
                   fat, energy)
GGally::ggpairs(df2)
```



Collinearity

```
fit5a <- lm(bp1 ~ bmi + ageye + energy, data= df)
round(summ(fit5a, confint = T)$"coeftable", 3)
```

```
##                Est.    2.5%   97.5%  t val.     p
## (Intercept) 125.641 124.330 126.952 188.516 0.000
## bmi           0.057   0.022   0.091   3.251 0.001
## ageye         0.051   0.035   0.066   6.471 0.000
## energy       -0.021  -0.046   0.004  -1.640 0.102
```

```
fit5b <- lm(bp1 ~ bmi + ageye + fat, data= df)
round(summ(fit5b, confint = T)$"coeftable", 3)
```

```
##                Est.    2.5%   97.5%  t val.     p
## (Intercept) 125.343 124.098 126.588 198.064 0.000
## bmi           0.054   0.020   0.089   3.097 0.002
## ageye         0.052   0.037   0.068   6.649 0.000
## fat          -0.023  -0.070   0.024  -0.962 0.337
```

Interpreting coefficients on transformed variables

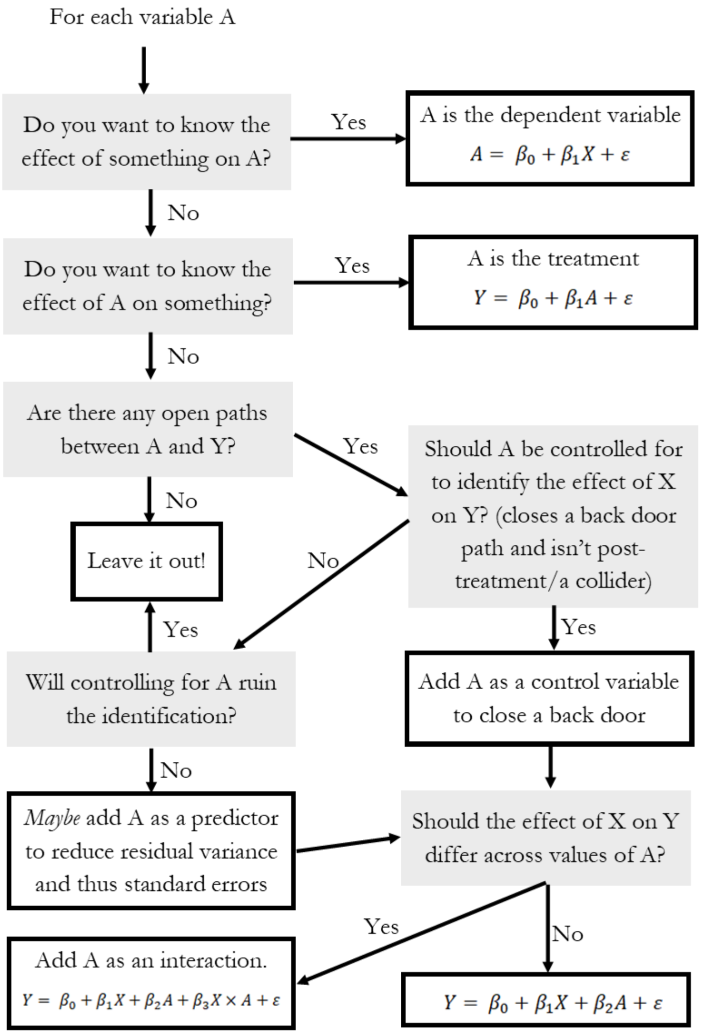
```
fit6 <- lm(bp1 ~ bmi + ageye +energy + log(weight), data= df)
round(summ(fit6, confint = T)$"coeftable", 3)
```

```
##                Est.    2.5%   97.5% t val.     p
## (Intercept) 121.374 116.436 126.312 48.351 0.000
## bmi           0.005  -0.062   0.072  0.158 0.874
## ageye         0.052   0.037   0.068  6.632 0.000
## energy       -0.024  -0.050   0.001 -1.898 0.059
## log(weight)   1.297  -0.150   2.745  1.763 0.079
```

SBP increases 1.3 mmHg for a 10% increase in weight

- Transforming explanatory variables will produce absolute effects of the response variable for a relative change of the explanatory variable.
- The size of the relative change reflected in the parameter estimate is determined by the base of the logarithm used.

Constructing a Regression Equation From The Effect



For each variable A

Do you want to know the effect of something on A?

A is the dependent variable $A = \beta_0 + \beta_1 X + \varepsilon$

Do you want to know the effect of A on something?

A is the treatment $Y = \beta_0 + \beta_1 A + \varepsilon$

Are there any open paths between A and Y?

Should A be controlled for to identify the effect of X on Y? (closes a back door path and isn't post-treatment/a collider)

Leave it out!

Will controlling for A ruin the identification?

Add A as a control variable to close a back door

Maybe add A as a predictor to reduce residual variance and thus standard errors

Should the effect of X on Y differ across values of A?

Add A as an interaction. $Y = \beta_0 + \beta_1 X + \beta_2 A + \beta_3 X \times A + \varepsilon$

$Y = \beta_0 + \beta_1 X + \beta_2 A + \varepsilon$

Diagnostic Plots

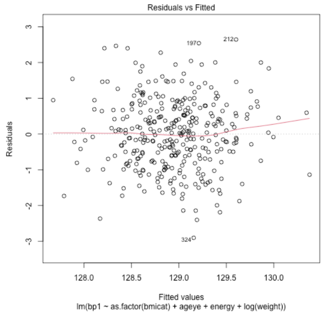
Consider this:

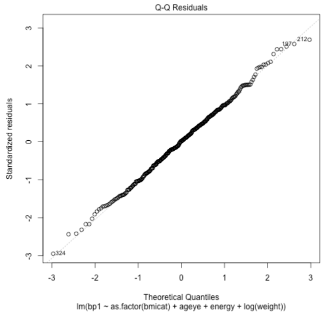
```r
fit7 <- lm(bp1 ~ as.factor(bmicat) + ageye
round(summ(fit7)$"coeftable", 3)
```

```
##                        Est.  S.E. t val.     p
## (Intercept)         119.856 2.473 48.470 0.000
## as.factor(bmicat)1    0.646 0.343  1.883 0.061
## as.factor(bmicat)2   -0.076 0.155 -0.488 0.626
## as.factor(bmicat)3    0.268 0.346  0.774 0.439
## ageye                 0.050 0.008  6.387 0.000
## energy               -0.022 0.013 -1.690 0.092
## log(weight)           1.687 0.588  2.871 0.004
```

Plot Residuals vs fitted and Q-Q plots

```r
plot(fit7, which=1:2)
```





Summary

Regression is a mathematical tool for making predictions and comparisons

Regression coefficients can always be interpreted as average comparisons

Regression coefficients can always be used for predictions

But, regression coefficients can only sometimes be interpreted as effects, depending on your causal model

QUESTIONS?

COMMENTS?

RECOMMENDATIONS?

Extra slide:

Code for the cloud figure (slide 45):

```r
gg1 <- ggplot(df, aes(x = bmicat1, y = bp1)) + ## add half-violin from {ggdist} package
  ggdist::stat_halfeye(## custom bandwidth
    adjust = .5, ## adjust height
    width = .6, ## move geom to the right
    justification = -.2, ## remove slab interval
    .width = 0,
    point_colour = NA
  ) +
  geom_boxplot(
    width = .12, ## remove outliers
    outlier.color = NA ## `outlier.shape = NA` works as well
  ) + ## add dot plots from {ggdist} package
  ggdist::stat_dots(## orientation to the left
    side = "left", ## move geom to the left
    justification = 1.1, ## adjust grouping (binning) of observations
    binwidth = .05) + ## remove white space on the left
  coord_cartesian(xlim = c(1.2, NA)) +
  geom_hline(yintercept=130, color = "red", size=2)+ theme_classic()
```

Bayesian Linear Regression

Four key steps for Bayesian modeling

Guide for the fundamentals of both single-level and hierarchical linear regression modeling

Can use **Stan** and front end `rstanarm` package (`brms` is good alternative)

Detailed vignettes can be found here

- **Step 1** Specify the data model and prior - Prior *likelihood* $\propto$ *posterior*
  -*Step 2* Estimate the model parameters - Bayes theorem typically involves using a numerical algorithm to draw a representative sample from the posterior distribution
- **Step 3** Check sampling quality and model fit - Graphical and numerical checks are necessary, if fails go back to Step 1
- **Step 4** Summarize, interpret results - Make posterior predictions

For some simple models, analytical (closed-form) solutions are possible
Almost all non-trivial models the full posterior has to be approximated numerically by sampling (simulating draws) based on Markov Chain Monte Carlo algorithms
If you have 1 hour check out this video to really understand MCMC

Linear regression (Bayesian)

Can also use `brms` package as the font end and get the same results

```r
library(brms)
fit_1b <- stan_glm(bp1 ~ bmi, data=df, seed=123, refresh = 0)
print(fit_1b, digits=2)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      bp1 ~ bmi
##  observations: 337
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 127.84   0.45
## bmi           0.05   0.02
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 1.07   0.04
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Default priors

`stan_glm` uses default weakly informative priors seen with `prior_summary(model)`

```
prior_summary(fit_1b)
```

```
## Priors for model 'fit_1b'
## ------
## Intercept (after predictors centered)
##   Specified prior:
##     ~ normal(location = 129, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 129, scale = 2.7)
##
## Coefficients
##   Specified prior:
##     ~ normal(location = 0, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 0, scale = 0.84)
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 0.93)
## ------
## See help('prior_summary.stanreg') for more details
```

Priors

Priors are often viewed as the Achilles' heel of Bayesian analyses.

Personally, they can be a **strength** as they allow the incorporation of prior knowledge, are entirely transparent and are updated by the current data following the uncontested laws of probability.

Bayesian analyses are sometimes done using **flat** or **non-informative** priors to allow final results to be completely dominated by the data.

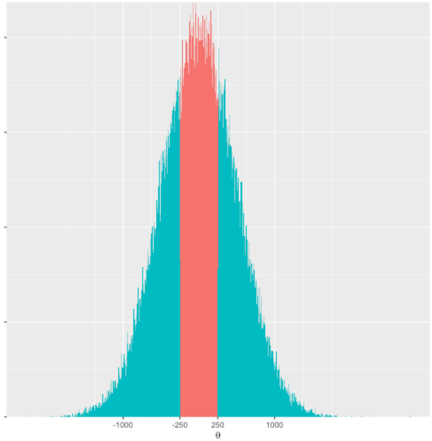
This is rarely a good idea

For example, using the prior θ = N(0, σ = 500) produces some strange beliefs

Non-informative priors are rarely a good idea

Consider one such non-informative prior N(0,500)

```
## [1] "Pr(-250 < theta < 250) = 0.38"
```



How could this represent anyone's serious prior beliefs
Some prior information usually available. Even if nothing to suggest a priori that a coefficient will be + or -, almost always can suggest that different orders of magnitude are not equally likely.

vague rather than **non-informative** priors are the default priors in most packages and should be used unless specific informative priors are available

Same results with OLS

Since there are 1,000 data points the priors probably contribute very little.
Therefore may expect to get the same numerical results with standard linear regression using `lm` function

```
fit_2 <- lm(bp1 ~ bmi, data=df)
print(fit_2, digits=2)
```

```
##
## Call:
## lm(formula = bp1 ~ bmi, data = df)
##
## Coefficients:
## (Intercept)          bmi
##     127.849        0.045
```

Same results as with the Bayesian approach