Homework 3

Vitaly Druker

9/30/2020

Example of HW

```
# How to create a data frame
x <- data.frame(x1 = c(1, NA, 2, 4, 5, 6),
            x2 = c(1, 2, 3, NA, NA, NA),
            y = c(1, 0, 1, 0, 1, 0)
           )
complete.cases(x)
```

```
## [1]  TRUE FALSE  TRUE FALSE FALSE FALSE
```

```
# This is a way to start the AIC processes
x_complete <- x[complete.cases(x), ]; x_complete
```

```
##   x1 x2 y
## 1  1  1 1
## 3  2  3 1
```

```
# For this we would look at both true false and get a p-value to test significance
x$is_missing <- !complete.cases(x)

x
```

```
##   x1 x2 y is_missing
## 1  1  1 1      FALSE
## 2 NA  2 0       TRUE
## 3  2  3 1      FALSE
## 4  4 NA 0       TRUE
## 5  5 NA 1       TRUE
## 6  6 NA 0       TRUE
```

```
# Here you use the is_missing to use both true and false to get significance of true
lm(y ~ is_missing, data = x)
```

```
##
## Call:
## lm(formula = y ~ is_missing, data = x)
##
## Coefficients:
##    (Intercept)  is_missingTRUE
##           1.00           -0.75
```

```
# How to replace values with NA's
x1 <- data.frame(x = c(0, 0, 0, 0, 100, 100, 100))
x1
```

```
##     x
## 1   0
## 2   0
## 3   0
## 4   0
## 5 100
## 6 100
## 7 100
```

```
x1$x <- ifelse(x1$x == 0, NA, x1$x) # If x1 is 0, replace with NA's, otherwise leave it as is
x1 # now shows NA's
```

```
##     x
## 1  NA
## 2  NA
## 3  NA
## 4  NA
## 5 100
## 6 100
## 7 100
```

The problem that's happening for example if fit a model with just x1 as a variable against y, you're going to have these 5 points in there because R always drops out NA's automatically. And then if you fit a model with x2 you only get these 3 points. The problem is that you can't compare those models when you're fitting it on the full data. The reason for that is because it's data and so it's not a fair way to compare things.

You can fit a model with just the cases where all the columns are filled that you're interested in. That function is complete cases, returns a vector for each row of the data frame and it tells you which one of them have complete cases. The third code gives you only the rows that have complete data. The fourth code creates a variable in the data.

The fourth code creates a variable in the data for complete data. This is a way you could take this and put a regression with just this value against y. This is not an unreasonable way to see if there is a pattern to missing data. If there's a pattern to the missing data, then you can't just drop out everything with missing data because there's a reason it's missing. But if there is a relationship there it's not completely safe to drop stuff out but its not unreasonable to do so.

The step function is comparing models. It needs to have the same data per step because the AIC is not valid if you aren't using the exact data every time.

Sometimes 99's, or 999, or 9999, will indicate missingness. So the histograms we had to make are good to check for that. If you see a big spike at 999, it's probably a missing value

Missingness can be a significant value. Let's say you are running a clinical study and you want to take somebody's blood pressure 4 weeks into a study. And you're giving them a drug for 2 months. Sometimes what could happen is if the patient is unhappy with drug that their taking because its making them feel bad or not making them feel any better, they might not show up to the blood pressure reading. In which case you have a missing value but it's indicative of outcome. It's not that they randomly didn't show up, they didn't show up for a reason and it actually has to do with the affect that you're looking for. Miss is such a problem that you always have to think about and deal with and its difficult. You really have think about why something is missing and prove to yourself that it is missing at random and not for some reason.

What values would make it significant? The old alpha 0.05. You could also use 0.10. You just have to justify you're use of whatever alpha you're looking at, aka how bad is it if you find a false positive?

Homework 3

```
library(faraway)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

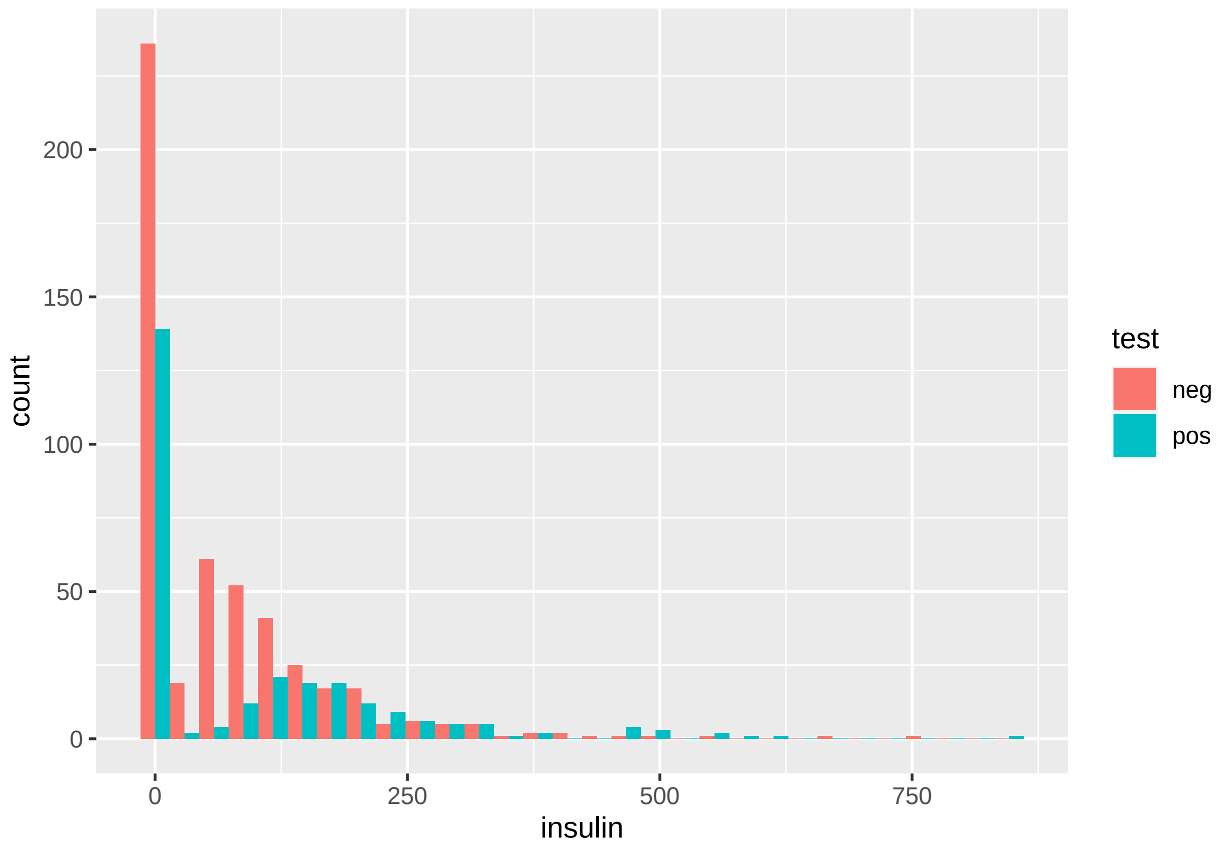
```
library(ggplot2)
```

```
pima %>% summary()
```

```
##     pregnant         glucose        diastolic         triceps
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     insulin           bmi           diabetes           age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##       test
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

```
# Create a factor variable

# This is by specifying the levels you want to use and what you want then to be labeled as
d <- pima %>%
  mutate(test = factor(test, levels = c(0, 1), labels = c("neg", "pos")))

d %>%
  ggplot(aes(x = insulin, fill = test)) +
  geom_histogram(position = "dodge")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

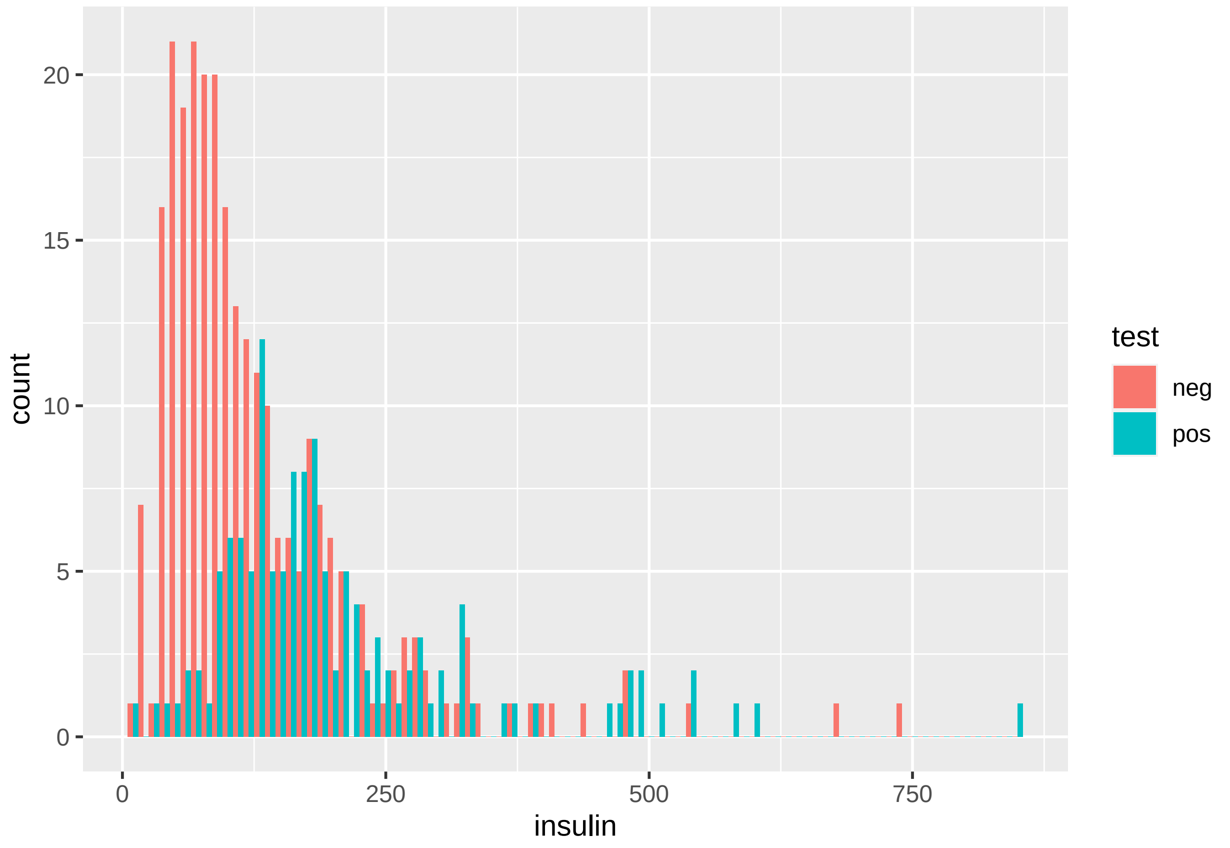


- If we kept this as numeric, it wouldn't work because ggplot wouldn't know how to work with it.
- Not physically possible for a human to have 0 insulin

```
# Replace the NA's
# The brackets mean subset,
# The first part of the code calls are the 0's in that variable and the second part turns them into NA'
d$insulin[d$insulin == 0] <- NA

# Another way to do it
# Ifelse -  if 0, make NA, otherwise leave it
d$insulin <- ifelse(d$insulin == 0, NA, d$insulin)

# Cleaner way to do it
# This is using all the columns that's in between glucose and bmi and applying the ifelse function to i
# The period in the ifelse code represents the columns itself, doing it over each column. saying if the
d <- d %>%
  mutate_at(vars(glucose:bmi), ~ifelse(. == 0, NA, .))

d %>% summary()
```

```
##     pregnant         glucose        diastolic        triceps
##  Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:22.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :29.00
##  Mean   : 3.845   Mean   :121.7   Mean   : 72.41   Mean   :29.15
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.:36.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##                   NA's   :5       NA's   :35       NA's   :227
##     insulin           bmi           diabetes           age          test
##  Min.   : 14.00   Min.   :18.20   Min.   :0.0780   Min.   :21.00   neg:500
##  1st Qu.: 76.25   1st Qu.:27.50   1st Qu.:0.2437   1st Qu.:24.00   pos:268
##  Median :125.00   Median :32.30   Median :0.3725   Median :29.00
##  Mean   :155.55   Mean   :32.46   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:190.00   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.00   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##  NA's   :374      NA's   :11
```

```
# GGplot
d %>%
  filter(!is.na(insulin)) %>% # This code takes out the warning
  ggplot(aes(x = insulin, fill = test)) +
  geom_histogram(position = "dodge", binwidth = 10) # Refers to the number of groups it creates
```



See that insulin and triceps have a bunch of missing data in it.

```
# Use the full dataset w/NA
# Use test as the outcome, the dot is the short hand for using everything but test, aka all the variabl
mod_c <- glm(test ~ . , data = d, family = "binomial")

# Same as writing all the predictors out
mod_c2 <- glm(test ~ pregnant + glucose + diastolic + bmi + diabetes + age + insulin + triceps, family =
nobs(mod_c)
```

```
## [1] 392
```

```
# Remove triceps and insulin
# Can't compare these two models because it's not having the same number of observations
```

```
mod_d <- glm(test ~ pregnant + glucose + diastolic + bmi + diabetes + age, data = d, family= "binomial"
nobs(mod_d)
```

```
## [1] 724
```

```
# One way to get all the complete data
d_complete <- d %>%
  filter(complete.cases(.))

# Another way to get complete data
# More simpler
d_complete <-d[complete.cases(d), ]
```

```
# Refit the models using the complete dataset
mod_c <- glm(test ~ . , data = d_complete, family = "binomial")
mod_d <- glm(test ~ pregnant + glucose + diastolic + bmi + diabetes + age,
             data = d_complete,
             family= "binomial")
nobs(mod_c)
```

```
## [1] 392
```

```
nobs(mod_d)
```

```
## [1] 392
```

```
# Chi-squared test by hand
pchisq(deviance(mod_d) - deviance(mod_c), df.residual(mod_d) - df.residual(mod_c), lower.tail= FALSE)
```

```
## [1] 0.6507347
```

```
# Anova test, gives the same exact thing
anova(mod_c, mod_d, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age
## Model 2: test ~ pregnant + glucose + diastolic + bmi + diabetes + age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       383     344.02
## 2       385     344.88 -2 -0.85931   0.6507
```

- The .65 in chi squared test means the two models are not different statistically, that the deviance seems to come from the distribution. which means that we aren't improving the model

```
# Fake example of a categorical variable with 3 categories: a,b,c
# The sample function, you sample these 3 values, how any samples you want: as many in the data set, an
d_complete2 <- d_complete %>%
  mutate(CAT1 = sample(c("A", "B", "C"), size = nrow(d_complete), replace = T))

# See what happens to degrees of freedom in step function when you have categorical data
mod_full <- glm(test ~ . + I(triceps^2) , data = d_complete2, family = "binomial")
mod_full <- step(mod_full) # Shows that removing CAT1, removing 2 df, because we have 3 predictors here
```

```
## Start:  AIC=367.13
## test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age + CAT1 + I(triceps^2)
##
```

```
##                 Df Deviance    AIC
## - CAT1           2   343.31 363.31
## - diastolic      1   343.14 365.14
## - insulin        1   343.49 365.49
## - I(triceps^2)   1   343.88 365.88
## - triceps        1   344.12 366.12
## - pregnant       1   344.97 366.97
## <none>               343.13 367.13
## - age            1   346.70 368.70
## - bmi            1   349.77 371.77
## - diabetes       1   350.98 372.98
## - glucose        1   394.12 416.12
##
## Step:  AIC=363.31
## test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age + I(triceps^2)
##
##                 Df Deviance    AIC
## - diastolic      1   343.31 361.31
## - insulin        1   343.70 361.70
## - I(triceps^2)   1   344.02 362.02
## - triceps        1   344.26 362.26
## - pregnant       1   345.14 363.14
## <none>               343.31 363.31
## - age            1   346.93 364.93
## - bmi            1   349.92 367.92
## - diabetes       1   351.27 369.27
## - glucose        1   395.19 413.19
##
## Step:  AIC=361.31
## test ~ pregnant + glucose + triceps + insulin + bmi + diabetes +
##     age + I(triceps^2)
##
##                 Df Deviance    AIC
## - insulin        1   343.70 359.70
## - I(triceps^2)   1   344.04 360.04
## - triceps        1   344.28 360.28
## - pregnant       1   345.14 361.14
## <none>               343.31 361.31
## - age            1   347.04 363.04
## - bmi            1   350.45 366.45
## - diabetes       1   351.32 367.32
## - glucose        1   395.73 411.73
##
## Step:  AIC=359.7
## test ~ pregnant + glucose + triceps + bmi + diabetes + age +
##     I(triceps^2)
##
##                 Df Deviance    AIC
## - I(triceps^2)   1   344.42 358.42
## - triceps        1   344.68 358.68
## - pregnant       1   345.63 359.63
## <none>               343.70 359.70
## - age            1   347.30 361.30
## - bmi           1   350.49 364.49
## - diabetes      1   351.56 365.56
## - glucose       1   409.07 423.07
##
## Step:  AIC=358.42
## test ~ pregnant + glucose + triceps + bmi + diabetes + age
##
##            Df Deviance    AIC
## - triceps   1   344.89 356.89
## <none>          344.42 358.42
## - pregnant  1   346.74 358.74
## - age       1   347.87 359.87
## - bmi       1   351.32 363.32
## - diabetes  1   351.90 363.90
## - glucose   1   411.11 423.11
##
## Step:  AIC=356.89
## test ~ pregnant + glucose + bmi + diabetes + age
##
##            Df Deviance    AIC
## <none>          344.89 356.89
## - pregnant  1   347.23 357.23
## - age       1   348.72 358.72
## - diabetes  1   352.72 362.72
## - bmi       1   360.44 370.44
## - glucose   1   411.85 421.85
```

```
# You can't compare models with different number of rows so we have to use the complete dataset, with n
mod_full <- glm(test ~ ., data = d_complete, family = "binomial")

# AIC step , the trace = 0 means it doesn't show in the output
mod_full <- step(mod_full, trace = 0)
nobs(mod_full)
```

```
## [1] 392
```

- The step function trace: the dash is actually a minus sign, if we remove diastolic here is the AIC, insulin here is the AIC, triceps here is the AIC, and the none if we don't do anything it stays the same, the number is the same as the start AIC.
- It steps through all of it until the lowest AIC is where none is listed at the top
- All of the degrees or freedom are 1, it has to do with how many columns you're removing.
- When can one variable actually become a bunch of different columns? If you have a categorical variable. when you create a categorical variable, that actually will create a bunch of different columns.
- In the fake data, when you do step function, shows that removing CAT1, removing 2 df, because we have 3 predictors here that removes 2 columns
- AIC is nice because it can quantify the addition of one column that actually becomes 2 df
- It's not surprising that this is the 1st thing that's removed because by definition it is a random variable
- You can refit the model after if you want

```
# Make true and false to see if any missing
d_any_missing <- d %>%
  mutate(any_missing = complete.cases(.))
```

```r
# Model with missing
mod_missing_test <- glm(test ~ any_missing , data = d_any_missing, family = "binomial")
# See the significant
mod_missing_test %>% summary()
```

```
##
## Call:
## glm(formula = test ~ any_missing, family = "binomial", data = d_any_missing)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9564  -0.9564  -0.8977   1.4159   1.4857
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -0.5450     0.1070  -5.094 3.51e-07 ***
## any_missingTRUE  -0.1558     0.1515  -1.028    0.304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.48  on 767  degrees of freedom
## Residual deviance: 992.43  on 766  degrees of freedom
## AIC: 996.43
##
## Number of Fisher Scoring iterations: 4
```

```r
# See if it's useful,
anova(mod_missing_test, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: test
##
## Terms added sequentially (first to last)
##
##
##             Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                          767     993.48
## any_missing  1   1.0579       766     992.43   0.3037
```

- Tells us that having a missing value, it's not actually indictive of whether you test positive or not.
- If we do have a missing value for a test, what could make it indictive of having diabetes? A selection bias if you deliberately leave something out because its embarrassing and that value could actually make it positive or even say whatever medications they are on. Even false info can be really bad, if data you have is incorrect and you don't know that its incorrect.
- Another way is that what if they have kidney disease and are maybe too sick to come in to do whatever test

```r
bmi_1st <- quantile(d_complete$bmi, .25)
bmi_3rd <- quantile(d_complete$bmi, .75)
```

```r
bmi_diff <- bmi_3rd - bmi_1st

# That 75% , to take it out unnamed
bmi_diff <- unname(bmi_diff)

# Take a look at the estimate, std error, p value, CI high and low, filtering for the bmi term
tidy_mod <- broom::tidy(mod_full, conf.int = T) %>%
    filter(term == "bmi")


# Multiply the bmi_diff to each of these values and then exponentiate it
# This is because the CI is semantic on the linear scale so you can't really just multiply it when its
# The .*: the dot is the column, mutating the estimate and the CI
# This code is short hand for the one below
tidy_mod %>%
  select(estimate, conf.low, conf.high) %>%
  mutate_all(~exp(.*bmi_diff))
```

```
## # A tibble: 1 x 3
##   estimate conf.low conf.high
##      <dbl>    <dbl>     <dbl>
## 1     1.97     1.40      2.84
```

```r
# This way also lets you keep everything in a nice tibble which is a basically a data frame, so its nic

# More explicate way of doing that^
exp(tidy_mod$conf.low * bmi_diff)
```

```
## [1] 1.400901
```

```r
exp(tidy_mod$estimate * bmi_diff)
```

```
## [1] 1.973495
```

```r
exp(tidy_mod$conf.high * bmi_diff)
```

```
## [1] 2.836714
```

```r
# Specific example for understanding odds, creating a data frame with 2 rows where the only diff with bi
subjects <- data.frame(
  pregnant = 1,
  age = 10,
  glucose = 119,
  diastolic = 70,
  triceps = 29,
  insulin = 125,
  diabetes = 0.52,
  bmi = c(bmi_1st + 10, bmi_3rd + 10)
)

# Predict shows probability, 26% chance of having bmi with a 3rd quantile value
# Odds and probability are two sides of the same coin
x <- predict(mod_full, newdata = subjects, type = "response")


(p1 <- x[1]) #28% with a 30yr, #16% with a 10yr
```

```
##       25%
## 0.1642348
```

```
(o1 <- p1/(1-p1)) #to make a prob an odds
```

```
##       25%
## 0.1965084
```

```
# Often what happens when you have low prob, it's very similar to the odds

(p2 <- x[2]) #43% with a 30yr, 27% with 10yr
```

```
##       75%
## 0.2794394
```

```
(o2 <- p2/(1-p2))
```

```
##       75%
## 0.3878083
```

```
# Odds ratio
o2/o1
```

```
##      75%
## 1.973495
```

```
exp(tidy_mod$estimate * bmi_diff)
```

```
## [1] 1.973495
```

```
broom::tidy(mod_full, conf.int = T)
```

```
## # A tibble: 6 x 7
##   term        estimate std.error statistic  p.value    conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>       <dbl>     <dbl>
## 1 (Intercept)  -9.99     1.09        -9.19 3.80e-20 -12.2         -7.97
## 2 pregnant      0.0840   0.0550       1.53 1.27e- 1  -0.0233       0.193
## 3 glucose       0.0365   0.00498      7.32 2.41e-13   0.0270       0.0466
## 4 bmi           0.0781   0.0206       3.79 1.49e- 4   0.0387       0.120
## 5 diabetes      1.15     0.424        2.71 6.67e- 3   0.338        2.00
## 6 age           0.0344   0.0178       1.93 5.37e- 2  -0.0000239    0.0701
```

- If the coefficient (beta) was negative, you're lower CI will be more negative
- Odds ratio will not change if you change the other predators, but the odds will change, because it's looking at a specific case.
- If you're bmi changes by 8.7, if you raise your bmi by 8.7, your odds of having the disease go up by 1.97.
- Say if you look at their bmi, and their bmi goes up by 8.7, their odds of having disease is 1.97. Maybe you should lower the bmi these people because then it lowers the risk of disease.
- Is it more helpful to take about odds or probability? Odds is easier, odds change by 1.97 no matter what odds you start at. Not the same when you talk about probability
- If you just look at the prob, for the difference of a 30yr and a 10yr, we can't say the probability changes for any change in bmi, it depend on where you start. and that's the problem with probability. It's harder to talk about changes in probability that apply to any person.
- We can say for a person of age 10, here's the change in probability, but we can't say independent of all other predators, how does bmi affect the probability. Where we can say is affect probability on the odds scale because of the way the odds ratio's multiply the odds.

- Say if you're talking about the effect of one piece of your model, you have to talk about it in terms of an odds ratio. However if you are talking about a particular patient, or patient population, you can talk about probability in that case. When it comes down to it people do like prob better.

```
# One way that you can do the same thing,
# We can rescale the bmi, it will go up by 8.7 in the original scale
d_new <- d_complete %>%
  mutate(bmi = bmi/bmi_diff)

# Refit the model
d_new_mod <- glm(test ~ . , data = d_new, family = "binomial") %>%
  step(trace = 0)

# New model with transformed bmi, the estimate and CI are actually the same
broom::tidy(d_new_mod, exponentiate = T, conf.int = T) %>%
  filter(term == "bmi")
```

```
## # A tibble: 1 x 7
##   term  estimate std.error statistic  p.value conf.low conf.high
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 bmi       1.97     0.179      3.79 0.000149     1.40      2.84
```

```
# Note that by scaling the bmi, the p-value doesn't change. This is good because otherwise you could ha

# Old model where we exponentiate by hand
bind_rows(
broom::tidy(d_new_mod, exponentiate = T, conf.int = T) %>%
  filter(term == "bmi"),
broom::tidy(mod_full, conf.int = T) %>%
  filter(term == "bmi") %>%
  mutate_at(vars(estimate, conf.low, conf.high), ~exp(.*bmi_diff))
)
```

```
## # A tibble: 2 x 7
##   term  estimate std.error statistic  p.value conf.low conf.high
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 bmi       1.97    0.179       3.79 0.000149     1.40      2.84
## 2 bmi       1.97    0.0206      3.79 0.000149     1.40      2.84
```

```
# Add diastolic back in to see if its significant and look at the p-value
update(mod_full, . ~ . + diastolic) %>% broom::tidy(exponentiate = T)
```

```
## # A tibble: 7 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 0.0000472   1.18      -8.43   3.53e-17
## 2 pregnant    1.09        0.0551     1.53   1.27e- 1
## 3 glucose     1.04        0.00500    7.30   2.85e-13
## 4 bmi         1.08        0.0216     3.64   2.69e- 4
## 5 diabetes    3.16        0.425      2.70   6.85e- 3
## 6 age         1.04        0.0182     1.90   5.71e- 2
## 7 diastolic   0.999       0.0118    -0.0678 9.46e- 1
```

```
# p-value is 9.4 which is highly nonsignificant

# Take a look at the univariate data
# We find that in the positive row the diastolic mean is higher than the negative one,
```

```r
d_complete %>%
  group_by(test) %>%
  summarise(diastolic_mean = mean(diastolic),
            diastoic_se = sd(diastolic))
```

```
## # A tibble: 2 x 3
##   test  diastolic_mean diastoic_se
##   <fct>          <dbl>       <dbl>
## 1 neg             69.0        11.9
## 2 pos             74.1        13.0
```

```r
# Saying a high diastolic is associated with a positive test value

# Run the model with just diastolic
glm(test ~ diastolic, data = d_complete, family = "binomial") %>%
  broom::tidy(exponentiate = T)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic    p.value
##   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)   0.0421   0.677       -4.68 0.00000284
## 2 diastolic     1.04     0.00923      3.74 0.000187
```

```r
# See that there is a significant p-value and the test is positive
```

- Why is this happening? Why is diastolic significant but not involved in the model?
- When you put them in a model together, you have to take in consideration all the other coefficients in the model. There could be something that is very similar to it, highly correlated, co-linear (if they move they move together),
- Really what they're doing is given that all of the other data were modeling on, how helpful is diastolic? When we fit the large model, it's not helpful.