Categorical Endpoints

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Import wages here  
library(readxl)  
wages <- read\_excel("wages-2.xlsx")  
  
# Fall back in case you cannot load wages  
#wages <- heights %>%  
# filter(income > 0) %>%  
# mutate(marital = as.character(marital),  
# sex = as.character(sex))  
  
# Import orings here  
library(readr)  
orings <- read\_csv("orings.csv")

## New names:  
## Rows: 23 Columns: 3  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," dbl  
## (3): ...1, temp, damage  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

# Fall back in case you cannot load orings  
require(faraway)

## Loading required package: faraway

## Warning: package 'faraway' was built under R version 4.4.2

##   
## Attaching package: 'faraway'  
##   
## The following object is masked \_by\_ '.GlobalEnv':  
##   
## orings

# data(orings)  
  
# Add the code to import hsb2 here  
library(readr)  
hsb <- read\_csv("hsb2.csv")

## Rows: 200 Columns: 11  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): gender, race, ses, schtyp, prog  
## dbl (6): id, read, write, math, science, socst  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

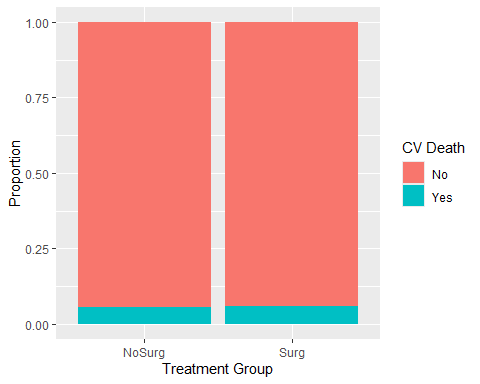
### Backup Import  
#hsb2 <- within(read.csv("https://stats.idre.ucla.edu/#stat/data/hsb2.csv"), {  
# race <- as.factor(race)  
# schtyp <- as.factor(schtyp)  
# prog <- as.factor(prog)  
#})

## Your Turn

* Save the wages dataset to your computer.
* Change the working directory to the same location.
* Import wages as wages and *copy the code to your setup chunk*.
* Be sure to set NA: to NA.
* Save the hsb2 dataset to your computer.
* Change the working directory to the same location.
* Import hsb as wages and *copy the code to your setup chunk*.
* Be sure to set NA: to NA.
* Save the orings dataset to your computer.
* Change the working directory to the same location.
* Import orings as wages and *copy the code to your setup chunk*.
* Be sure to set NA: to NA.

## Example: Chi-square Test of Independence Replication

### Chi-square example  
  
x <- c(rep("Surg",33),rep("NoSurg",18))  
y <- c(rep("Yes",2),rep("No",31),rep("Yes",1),rep("No",17))  
  
df <- data.frame(cbind(x,y))  
  
df.g <- df %>% group\_by(x) %>% count(y) %>% mutate(prop = n/sum(n))   
   
ggplot(data=df.g, aes(x = x, y = prop, fill = y)) +   
 geom\_bar(stat = "identity") +   
 labs(x="Treatment Group",fill="CV Death",y="Proportion")

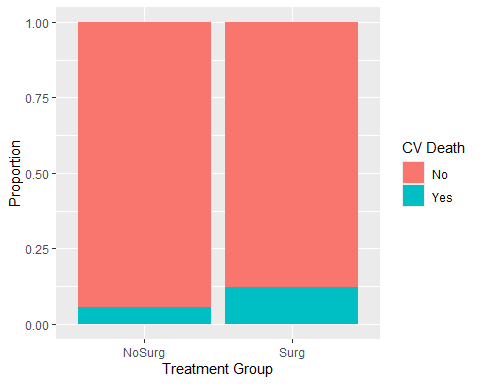


x.mat <- matrix(df.g$n,nrow=2,ncol=2)  
chisq.test(x.mat)

## Warning in chisq.test(x.mat): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: x.mat  
## X-squared = 7.1964e-32, df = 1, p-value = 1

### Chi-square example  
  
x <- c(rep("Surg",33),rep("NoSurg",18))  
y <- c(rep("Yes",4),rep("No",29),rep("Yes",1),rep("No",17))  
  
df <- data.frame(cbind(x,y))  
  
df.g <- df %>% group\_by(x) %>% count(y) %>% mutate(prop = n/sum(n))   
  
ggplot(data=df.g, aes(x = x, y = prop, fill = y)) +   
 geom\_bar(stat = "identity") +   
 labs(x="Treatment Group",fill="CV Death",y="Proportion")

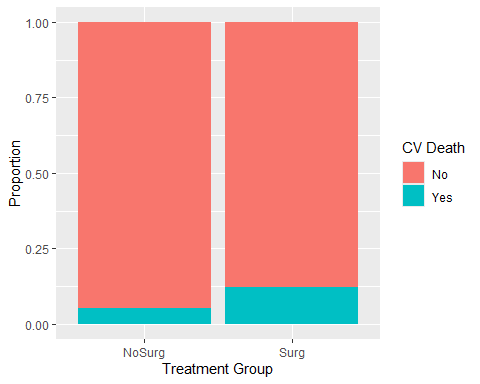


x.mat <- matrix(df.g$n,nrow=2,ncol=2)  
chisq.test(x.mat)

## Warning in chisq.test(x.mat): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: x.mat  
## X-squared = 0.068034, df = 1, p-value = 0.7942

### Replicate Chi-square example result  
  
x <- c(rep("Surg",32990),rep("NoSurg",173965))  
y <- c(rep("Yes",4068),rep("No",28922),rep("Yes",9101),rep("No",164864))  
  
df <- data.frame(cbind(x,y))  
  
df.g <- df %>% group\_by(x) %>% count(y) %>% mutate(prop = n/sum(n))   
  
ggplot(data=df.g, aes(x = x, y = prop, fill = y)) +   
 geom\_bar(stat = "identity") +   
 labs(x="Treatment Group",fill="CV Death",y="Proportion")



x.mat <- matrix(df.g$n,nrow=2,ncol=2)  
  
x.mat <- matrix(c(28922,4068,164864,9101),nrow=2,ncol=2)  
chisq.test(x.mat)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: x.mat  
## X-squared = 2344.7, df = 1, p-value < 2.2e-16

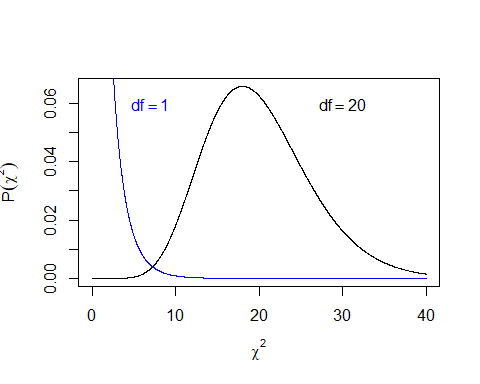
chisq.test(x.mat)$observed

## [,1] [,2]  
## [1,] 28922 164864  
## [2,] 4068 9101

chisq.test(x.mat)$expected

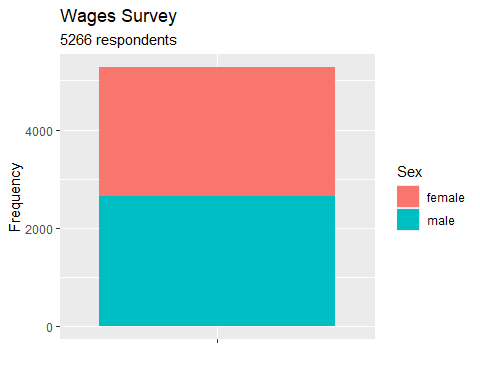
## [,1] [,2]  
## [1,] 30890.774 162895.23  
## [2,] 2099.226 11069.77

### Chi-square distribution  
  
x <- seq(0,40,0.01)  
y <- dchisq(x,20)  
plot(y~x,type="l",xlab=expression(chi^2),ylab=expression(P(chi^2)))  
text(30,0.06,expression(df==20))  
y2 <- dchisq(x,1)  
lines(y2~x,col="blue")  
text(7,0.06,expression(df==1),col="blue")



## Example: Chi-square Goodness of Fit test

### Barchart of sex  
  
wages <- wages %>% mutate(dummy="")  
  
wages %>% ggplot(aes(x=dummy,fill=sex)) +  
 geom\_bar(position="stack") +  
 xlab("") +  
 labs(title="Wages Survey",  
 subtitle="5266 respondents",  
 x="",  
 y="Frequency",  
 fill="Sex")



### Chi-square goodness of fit test on wages  
  
df.g <- wages %>% group\_by(sex) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))   
x.mat <- matrix(df.g$n,nrow=1,ncol=2)  
chisq.test(x.mat,p=c(0.5,0.5))

##   
## Chi-squared test for given probabilities  
##   
## data: x.mat  
## X-squared = 0.43752, df = 1, p-value = 0.5083

### Exact Binomial Confidence intervals for proportions  
binom.test(x.mat[1],x.mat[2]+x.mat[1]) # Females = 2609 out of 2609+2657 total

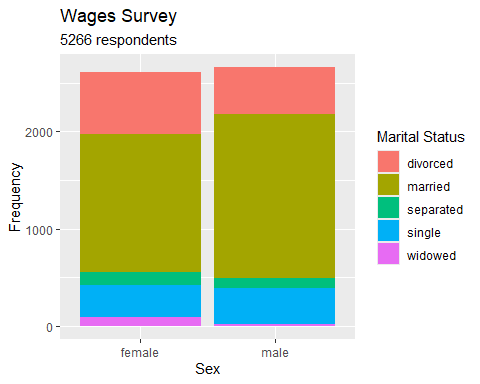
##   
## Exact binomial test  
##   
## data: x.mat[1] and x.mat[2] + x.mat[1]  
## number of successes = 2609, number of trials = 5266, p-value = 0.5172  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.4818493 0.5090407  
## sample estimates:  
## probability of success   
## 0.4954425

binom.test(x.mat[2],x.mat[2]+x.mat[1]) # Males = 2657 out of 2609+2657 total

##   
## Exact binomial test  
##   
## data: x.mat[2] and x.mat[2] + x.mat[1]  
## number of successes = 2657, number of trials = 5266, p-value = 0.5172  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.4909593 0.5181507  
## sample estimates:  
## probability of success   
## 0.5045575

## Example: Chi-square Test of Independence

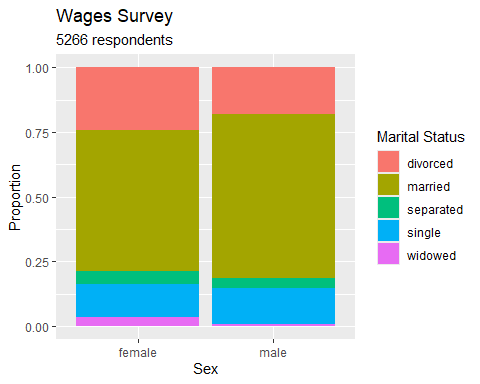
### Barchart of sex and marital status  
  
wages %>% ggplot(aes(x=sex,fill=marital)) +  
 geom\_bar() +  
 labs(title="Wages Survey",  
 subtitle="5266 respondents",  
 x="Sex",  
 y="Frequency",  
 fill="Marital Status")



df.g <- wages %>% group\_by(sex,marital) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))

## `summarise()` has grouped output by 'sex'. You can override using the `.groups`  
## argument.

### Barchart of sex and marital status proportions  
df.g %>% ggplot(aes(x=sex,y=prop,fill=marital)) +  
 geom\_col() +  
 labs(title="Wages Survey",  
 subtitle="5266 respondents",  
 x="Sex",  
 y="Proportion",  
 fill="Marital Status")



### Chi-square test of independence on wages  
  
x.mat <- matrix(df.g$n,nrow=5,ncol=2)  
(c <- chisq.test(x.mat))

##   
## Pearson's Chi-squared test  
##   
## data: x.mat  
## X-squared = 98.252, df = 4, p-value < 2.2e-16

c$observed

## [,1] [,2]  
## [1,] 632 483  
## [2,] 1418 1683  
## [3,] 136 97  
## [4,] 334 376  
## [5,] 89 18

c$expected

## [,1] [,2]  
## [1,] 552.41834 562.58166  
## [2,] 1536.36707 1564.63293  
## [3,] 115.43809 117.56191  
## [4,] 351.76415 358.23585  
## [5,] 53.01234 53.98766

c$residuals

## [,1] [,2]  
## [1,] 3.3859374 -3.3552137  
## [2,] -3.0198364 2.9924346  
## [3,] 1.9137659 -1.8964005  
## [4,] -0.9471496 0.9385552  
## [5,] 4.9427093 -4.8978596

x.mat

## [,1] [,2]  
## [1,] 632 483  
## [2,] 1418 1683  
## [3,] 136 97  
## [4,] 334 376  
## [5,] 89 18

(cont <- 100\*c$residuals^2/c$statistic) # which cells contribute the most?

## [,1] [,2]  
## [1,] 11.6684996 11.4577025  
## [2,] 9.2816243 9.1139472  
## [3,] 3.7276470 3.6603053  
## [4,] 0.9130495 0.8965548  
## [5,] 24.8649334 24.4157363

### Exact Binomial Confidence intervals for proportions  
binom.test(x.mat[5,1],x.mat[5,1]+x.mat[5,2]) # Proportion of divorced that are females

##   
## Exact binomial test  
##   
## data: x.mat[5, 1] and x.mat[5, 1] + x.mat[5, 2]  
## number of successes = 89, number of trials = 107, p-value = 1.781e-12  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.7472253 0.8971434  
## sample estimates:  
## probability of success   
## 0.8317757

binom.test(x.mat[5,2],x.mat[5,1]+x.mat[5,2]) # Proportion of divorced that are males

##   
## Exact binomial test  
##   
## data: x.mat[5, 2] and x.mat[5, 1] + x.mat[5, 2]  
## number of successes = 18, number of trials = 107, p-value = 1.781e-12  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.1028566 0.2527747  
## sample estimates:  
## probability of success   
## 0.1682243

binom.test(x.mat[5,1],x.mat[1,1]+x.mat[2,1]+x.mat[3,1]+x.mat[4,1]+x.mat[5,1]) # Proportion of females that are widowed

##   
## Exact binomial test  
##   
## data: x.mat[5, 1] and x.mat[1, 1] + x.mat[2, 1] + x.mat[3, 1] + x.mat[4, 1] + x.mat[5, 1]  
## number of successes = 89, number of trials = 2609, p-value < 2.2e-16  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.02748301 0.04181255  
## sample estimates:  
## probability of success   
## 0.03411269

binom.test(x.mat[5,2],x.mat[1,2]+x.mat[2,2]+x.mat[3,2]+x.mat[4,2]+x.mat[5,2]) # Proportion of males that are widowed

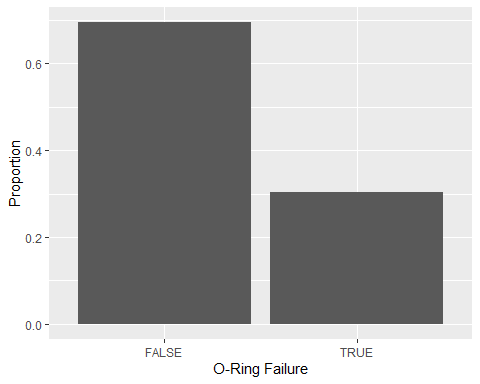
##   
## Exact binomial test  
##   
## data: x.mat[5, 2] and x.mat[1, 2] + x.mat[2, 2] + x.mat[3, 2] + x.mat[4, 2] + x.mat[5, 2]  
## number of successes = 18, number of trials = 2657, p-value < 2.2e-16  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.004019829 0.010685666  
## sample estimates:  
## probability of success   
## 0.006774558

## Example: Logistic Regression on Binary Y

### Logistic regression Challenger example  
  
data(orings)  
  
## Recode damage variable to a binary "Failure" versus "No Failure" variable  
orings <- orings %>% mutate(fail = damage > 0)  
orings %>% count(fail) # Checking recode

## fail n  
## 1 FALSE 16  
## 2 TRUE 7

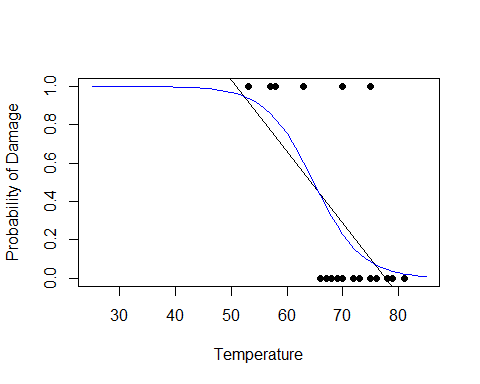
# Bar chart of failure  
  
df.g <- orings %>% group\_by(fail) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))   
  
ggplot(data=df.g, aes(x = fail, y = prop)) +   
 geom\_bar(stat = "identity") +   
 labs(x="O-Ring Failure",fill="",y="Proportion")



plot(fail~temp, orings, xlim=c(25,85), ylim=c(0,1),  
 xlab="Temperature",ylab="Probability of Damage",pch=19)  
abline(lm(fail~temp, orings))  
  
m <- glm(fail~temp, family = binomial, data = orings)  
summary(m)

##   
## Call:  
## glm(formula = fail ~ temp, family = binomial, data = orings)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 15.0429 7.3786 2.039 0.0415 \*  
## temp -0.2322 0.1082 -2.145 0.0320 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 28.267 on 22 degrees of freedom  
## Residual deviance: 20.315 on 21 degrees of freedom  
## AIC: 24.315  
##   
## Number of Fisher Scoring iterations: 5

x <- seq(25,85,1)  
lines(x, ilogit(coef(m)[1]+coef(m)[2]\*x),col="blue")



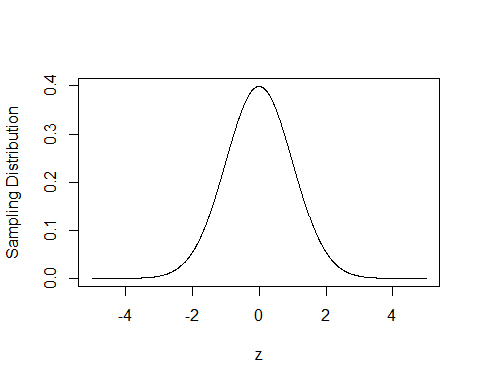
# Confidence Intervals for Regression Parameters  
# Using Normal Approximations  
# beta-hat +/- Z\_alpha/2 SE(beta-hat)  
alpha <- 0.1  
(lb <- coef(m)[2] - qnorm(1-alpha/2,0,1)\*summary(m)$coef[2,2])

## temp   
## -0.4101958

(ub <- coef(m)[2] + qnorm(1-alpha/2,0,1)\*summary(m)$coef[2,2])

## temp   
## -0.05412966

## Z distribution  
x <- seq(-5,5,0.01)  
z <- dnorm(x,0,1)  
plot(z~x,type="l",xlab="z",ylab="Sampling Distribution")



# Another approach  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

confint(m)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 3.3305848 34.34215133  
## temp -0.5154718 -0.06082076

# Interpretation of Regression Parameter  
# Transform it to an odds ratio  
exp(coef(m)[2])

## temp   
## 0.7928171

## Predict log odds and transform back to scale of probability  
  
# Predict the Response  
ilogit(coef(m)[1]+coef(m)[2]\*31)

## (Intercept)   
## 0.9996088

# Another way  
x.new <- c(1,31)  
link <- sum(x.new\*coef(m))  
ilogit(link)

## [1] 0.9996088

p <- predict(m, newdata = data.frame(temp=31), se=T)  
ilogit(p$fit - qnorm(1-alpha/2,0,1)\*p$se.fit)

## 1   
## 0.7684619

ilogit(p$fit + qnorm(1-alpha/2,0,1)\*p$se.fit)

## 1   
## 0.9999995

# Inverse prediction  
# What value of X will lead to a certain probability threshold of the event?  
library(MASS)  
dose.p(m,p=0.5)

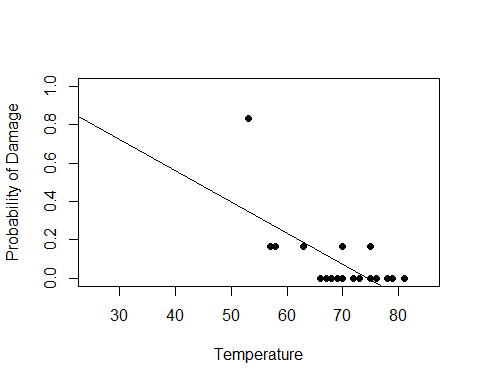
## Dose SE  
## p = 0.5: 64.79464 2.80903

dose.p(m,p=0.1)

## Dose SE  
## p = 0.1: 74.2588 3.775851

## Example: Logistic Regression on Proportion instead of Binary Y

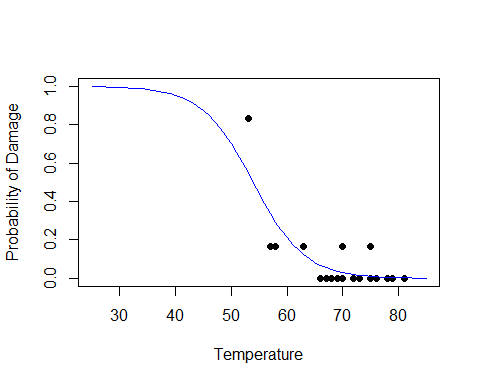
### Second Logistic regression Challenger example modeling the probability of   
### damage using the number of o-rings damaged out of 6.  
  
plot(damage/6~temp, orings, xlim = c(25,85),  
 ylim=c(0,1), xlab = "Temperature", ylab = "Probability of Damage",  
 pch = 19)  
abline(lm(damage/6~temp, orings))



m <- glm(cbind(damage,6-damage)~temp, family = binomial, data = orings)  
summary(m)

##   
## Call:  
## glm(formula = cbind(damage, 6 - damage) ~ temp, family = binomial,   
## data = orings)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 11.66299 3.29626 3.538 0.000403 \*\*\*  
## temp -0.21623 0.05318 -4.066 4.78e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 38.898 on 22 degrees of freedom  
## Residual deviance: 16.912 on 21 degrees of freedom  
## AIC: 33.675  
##   
## Number of Fisher Scoring iterations: 6

plot(damage/6~temp, orings, xlim = c(25,85),  
 ylim=c(0,1), xlab = "Temperature", ylab = "Probability of Damage",  
 pch = 19)  
x <- seq(25,85,1)  
lines(x, ilogit(coef(m)[1]+coef(m)[2]\*x),col="blue")



# Confidence Intervals for Regression Parameters  
# Using Normal Approximations  
# beta-hat +/- Z\_alpha/2 SE(beta-hat)  
alpha <- 0.1  
(lb <- coef(m)[2] - qnorm(1-alpha/2,0,1)\*summary(m)$coef[2,2])

## temp   
## -0.3037021

(ub <- coef(m)[2] + qnorm(1-alpha/2,0,1)\*summary(m)$coef[2,2])

## temp   
## -0.1287652

# Another approach  
library(MASS)  
confint(m)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) 5.575195 18.737598  
## temp -0.332657 -0.120179

# Interpretation of Regression Parameter  
# Transform it to an odds ratio  
exp(coef(m)[2])

## temp   
## 0.8055471

## Predict log odds and transform back to scale of probability  
  
# Predict the Response  
ilogit(coef(m)[1]+coef(m)[2]\*31)

## (Intercept)   
## 0.9930342

# Another way  
x.new <- c(1,31)  
link <- sum(x.new\*coef(m))  
ilogit(link)

## [1] 0.9930342

p <- predict(m, newdata = data.frame(temp=31), se=T)  
ilogit(p$fit - qnorm(1-alpha/2,0,1)\*p$se.fit)

## 1   
## 0.9017892

ilogit(p$fit + qnorm(1-alpha/2,0,1)\*p$se.fit)

## 1   
## 0.9995484

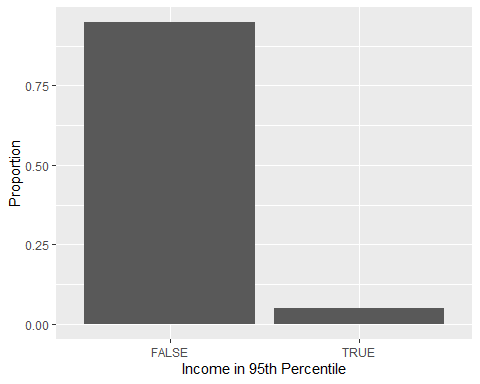
# Inverse prediction  
# What value of X will lead to a certain probability threshold of the event?  
dose.p(m,p=0.5)

## Dose SE  
## p = 0.5: 53.93697 2.515673

### Your Turn 6  
  
## Recode variable to quantile  
wages <- wages %>% mutate(new.income = income > quantile(income,0.95))  
wages %>% count(new.income) # Checking recode

## # A tibble: 2 × 2  
## new.income n  
## <lgl> <int>  
## 1 FALSE 5002  
## 2 TRUE 264

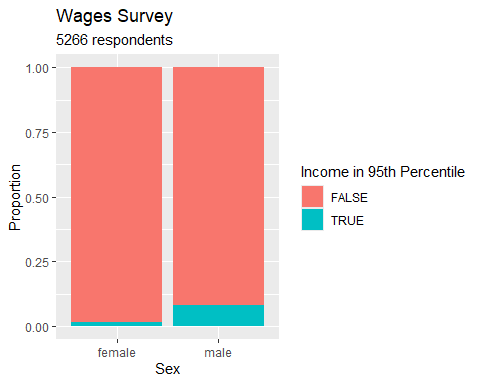
# Bar chart  
  
df.g <- wages %>% group\_by(new.income) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))   
  
ggplot(data=df.g, aes(x = new.income, y = prop)) +   
 geom\_bar(stat = "identity") +   
 labs(x="Income in 95th Percentile",fill="",y="Proportion")



df.g <- wages %>% group\_by(sex,new.income) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))

## `summarise()` has grouped output by 'sex'. You can override using the `.groups`  
## argument.

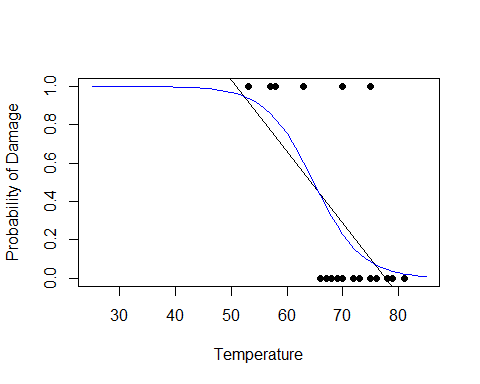
### Barchart of sex and income status proportions  
df.g %>% ggplot(aes(x=sex,y=prop,fill=new.income)) +  
 geom\_col() +  
 labs(title="Wages Survey",  
 subtitle="5266 respondents",  
 x="Sex",  
 y="Proportion",  
 fill="Income in 95th Percentile")



plot(fail~temp, orings, xlim=c(25,85), ylim=c(0,1),  
 xlab="Temperature",ylab="Probability of Damage",pch=19)  
abline(lm(fail~temp, orings))  
  
m <- glm(fail~temp, family = binomial, data = orings)  
summary(m)

##   
## Call:  
## glm(formula = fail ~ temp, family = binomial, data = orings)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 15.0429 7.3786 2.039 0.0415 \*  
## temp -0.2322 0.1082 -2.145 0.0320 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 28.267 on 22 degrees of freedom  
## Residual deviance: 20.315 on 21 degrees of freedom  
## AIC: 24.315  
##   
## Number of Fisher Scoring iterations: 5

x <- seq(25,85,1)  
lines(x, ilogit(coef(m)[1]+coef(m)[2]\*x),col="blue")



## Your Turn 1

Use the pchisq() function to calculate the p-value for with 1 and 10 degrees of freedom and make a decision.

p\_value\_1 <- 1 - pchisq(3, df = 1)  
  
p\_value\_10 <- 1 - pchisq(3, df = 10)  
  
p\_value\_1

## [1] 0.08326452

p\_value\_10

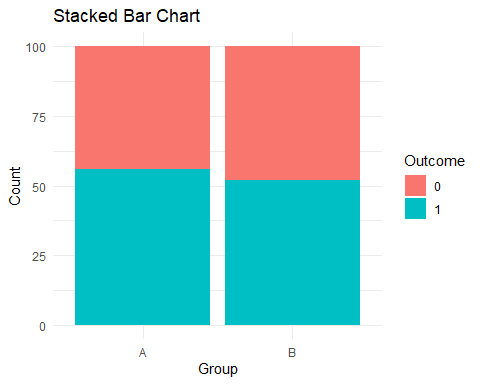
## [1] 0.9814241

#As the degrees of freedom increase, so does the spread of values. As a result, higher DFs (like 10) will  
#lead to higher probabilities that the values are to the right of our chi value of 3.

## Your Turn 2

Uncomment the code to generate binary data for two groups with equal proportions, then create a stacked barchart using ggplot().

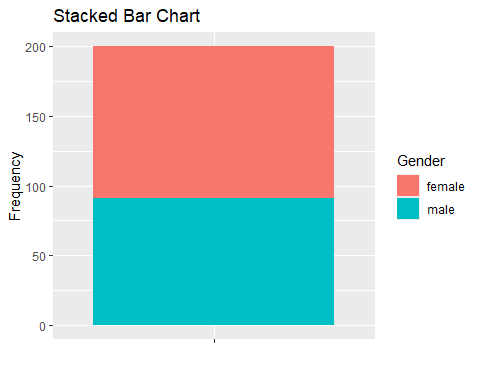
x <- c(rep("A",100),rep("B",100))  
y <- rbinom(200,1,0.5)  
  
# Create a stacked bar chart using ggplot2  
ggplot(mapping = aes(x = x, fill = factor(y, labels = c("0", "1")))) +  
 geom\_bar(position = "stack") +  
 labs(title = "Stacked Bar Chart",  
 x = "Group",  
 y = "Count",  
 fill = "Outcome") +  
 theme\_minimal()



## Your Turn 3

Use the hsb2 dataset to test whether there are an equal number of males and females surveyed. Create the stacked barchart and calculate the exact 95% confidence intervals for each gender. Interpret.

hsb <- hsb %>% mutate(dummy="")  
  
hsb %>% ggplot(aes(x=dummy,fill=gender)) +  
 geom\_bar(position="stack") +  
 xlab("") +  
 labs(title="Stacked Bar Chart",  
 x="",  
 y="Frequency",  
 fill="Gender")



df.g <- hsb %>% group\_by(gender) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))  
x.mat <- matrix(df.g$n, nrow=1, ncol=2)  
chisq.test(x.mat,p=c(0.5,0.5))

##   
## Chi-squared test for given probabilities  
##   
## data: x.mat  
## X-squared = 1.62, df = 1, p-value = 0.2031

binom.test(x.mat[1],x.mat[2]+x.mat[1]) #109 / 200 females

##   
## Exact binomial test  
##   
## data: x.mat[1] and x.mat[2] + x.mat[1]  
## number of successes = 109, number of trials = 200, p-value = 0.2292  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.4732592 0.6153870  
## sample estimates:  
## probability of success   
## 0.545

binom.test(x.mat[2],x.mat[2]+x.mat[1]) #91 / 200 males

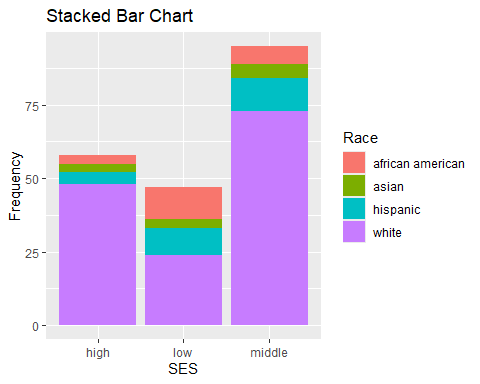
##   
## Exact binomial test  
##   
## data: x.mat[2] and x.mat[2] + x.mat[1]  
## number of successes = 91, number of trials = 200, p-value = 0.2292  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.3846130 0.5267408  
## sample estimates:  
## probability of success   
## 0.455

#We cannot reject H0 (Oi = Ei for all i) as the p value (0.2031) is greater than 0.05.

## Your Turn 4

Use the hsb2 dataset to test whether race and socio-economic status are independent. Don’t forget to visualize the relationship and generate estimates and confidence intervals. Interpret.

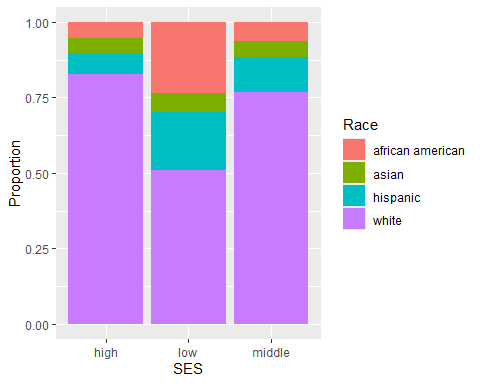
hsb %>% ggplot(aes(x=ses,fill=race)) +  
 geom\_bar() +  
 labs(title="Stacked Bar Chart",  
 x="SES",  
 y="Frequency",  
 fill="Race")



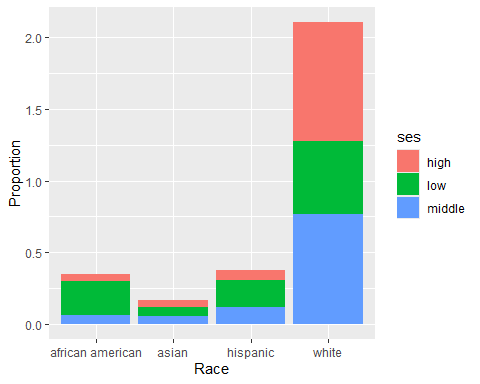
df.g <- hsb %>% group\_by(ses,race) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))

## `summarise()` has grouped output by 'ses'. You can override using the `.groups`  
## argument.

### Barchart of ses and race status proportions  
df.g %>% ggplot(aes(x=ses,y=prop,fill=race)) +  
 geom\_col() +  
 labs(x="SES",  
 y="Proportion",  
 fill="Race")



### Barchart of race and ses status proportions  
df.g %>% ggplot(aes(x=race,y=prop,fill=ses)) +  
 geom\_col() +  
 labs(x="Race",  
 y="Proportion",  
 fill="ses")



### Chi-square test of independence on ses with race  
  
x.mat <- matrix(df.g$n,nrow=4,ncol=3)  
(c <- chisq.test(x.mat))

## Warning in chisq.test(x.mat): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: x.mat  
## X-squared = 18.516, df = 6, p-value = 0.005064

#The p value is 0.005, so reject H0  
  
x.mat

## [,1] [,2] [,3]  
## [1,] 3 11 6  
## [2,] 3 3 5  
## [3,] 4 9 11  
## [4,] 48 24 73

(cont <- 100\*c$residuals^2/c$statistic)

## [,1] [,2] [,3]  
## [1,] 7.30032270 45.6076011 6.96412362  
## [2,] 0.06111819 0.3598236 0.05232783  
## [3,] 6.79873590 10.8106906 0.07579999  
## [4,] 4.54696823 16.0882288 1.33425943

#It looks like the biggest contributors to rejection are African Americans and Whites.  
#So we'll get confidence intervals for those.  
  
### Exact Binomial Confidence intervals for proportions  
binom.test(x.mat[1,1],x.mat[1,1]+x.mat[1,2] + x.mat[1,3]) # Proportion of african american that are low SES

##   
## Exact binomial test  
##   
## data: x.mat[1, 1] and x.mat[1, 1] + x.mat[1, 2] + x.mat[1, 3]  
## number of successes = 3, number of trials = 20, p-value = 0.002577  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.03207094 0.37892683  
## sample estimates:  
## probability of success   
## 0.15

binom.test(x.mat[1,2],x.mat[1,1]+x.mat[1,2] + x.mat[1,3]) # Proportion of african american that are middle SES

##   
## Exact binomial test  
##   
## data: x.mat[1, 2] and x.mat[1, 1] + x.mat[1, 2] + x.mat[1, 3]  
## number of successes = 11, number of trials = 20, p-value = 0.8238  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.3152781 0.7694221  
## sample estimates:  
## probability of success   
## 0.55

binom.test(x.mat[1,3],x.mat[1,1]+x.mat[1,2] + x.mat[1,3]) # Proportion of african american that are high SES

##   
## Exact binomial test  
##   
## data: x.mat[1, 3] and x.mat[1, 1] + x.mat[1, 2] + x.mat[1, 3]  
## number of successes = 6, number of trials = 20, p-value = 0.1153  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.1189316 0.5427892  
## sample estimates:  
## probability of success   
## 0.3

binom.test(x.mat[4,1],x.mat[4,1]+x.mat[4,2] + x.mat[4,3]) # Proportion of whites that are low SES

##   
## Exact binomial test  
##   
## data: x.mat[4, 1] and x.mat[4, 1] + x.mat[4, 2] + x.mat[4, 3]  
## number of successes = 48, number of trials = 145, p-value = 5.777e-05  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.2551841 0.4139503  
## sample estimates:  
## probability of success   
## 0.3310345

binom.test(x.mat[4,2],x.mat[4,1]+x.mat[4,2] + x.mat[4,3]) # Proportion of whites that are middle SES

##   
## Exact binomial test  
##   
## data: x.mat[4, 2] and x.mat[4, 1] + x.mat[4, 2] + x.mat[4, 3]  
## number of successes = 24, number of trials = 145, p-value < 2.2e-16  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.1090229 0.2361923  
## sample estimates:  
## probability of success   
## 0.1655172

binom.test(x.mat[4,3],x.mat[4,1]+x.mat[4,2] + x.mat[4,3]) # Proportion of whites that are high SES

##   
## Exact binomial test  
##   
## data: x.mat[4, 3] and x.mat[4, 1] + x.mat[4, 2] + x.mat[4, 3]  
## number of successes = 73, number of trials = 145, p-value = 1  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.4192907 0.5874621  
## sample estimates:  
## probability of success   
## 0.5034483

## Your Turn 5

Use the hsb2 dataset. Fit a logistic regression model to test whether reading scores are associated with mathematics scores in at least the 75th percentile. Visualize the data, generate estimates and confidence intervals, and interpret the results.

#Get the value for the 75th percentile in math  
cutoff <- quantile(hsb$math, 0.75)  
  
#Create a new column called Y that is either 1 or 0 for Math scores  
#Create the model  
hsb <- hsb %>% mutate(Y = ifelse(math >= cutoff, 1, 0))  
model\_1 <- glm(Y ~ read, data = hsb, family = binomial)  
summary(model\_1)

##   
## Call:  
## glm(formula = Y ~ read, family = binomial, data = hsb)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.27693 1.35712 -6.836 8.16e-12 \*\*\*  
## read 0.14841 0.02339 6.344 2.24e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 227.10 on 199 degrees of freedom  
## Residual deviance: 167.25 on 198 degrees of freedom  
## AIC: 171.25  
##   
## Number of Fisher Scoring iterations: 5

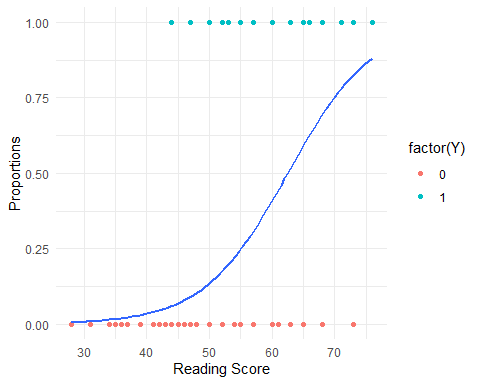
#B0 = 9.27693  
#B1 = 0.14841  
  
#p value (2.24e-10) and z score (6.344) lead us to reject H0. As a result, B1 != 0  
  
confint(model\_1)

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) -12.1440607 -6.7956693  
## read 0.1053248 0.1975125

ggplot(hsb, aes(x = read, y = Y)) +  
 geom\_point(aes(color = factor(Y))) +  
 geom\_smooth(method = "glm", method.args = list(family = binomial), se = FALSE) +   
 labs(x = "Reading Score",  
 y = "Proportions") +  
 theme\_minimal()

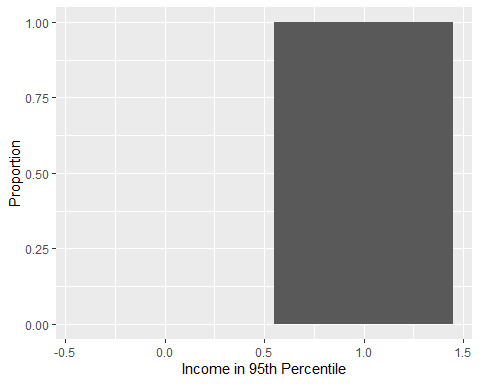
## `geom\_smooth()` using formula = 'y ~ x'



## Your Turn 6

Use the wages dataset. Fit a logistic regression model to test whether sex is associated with income in at least the 95th percentile. Visualize the data, generate estimates and confidence intervals, and interpret the results.

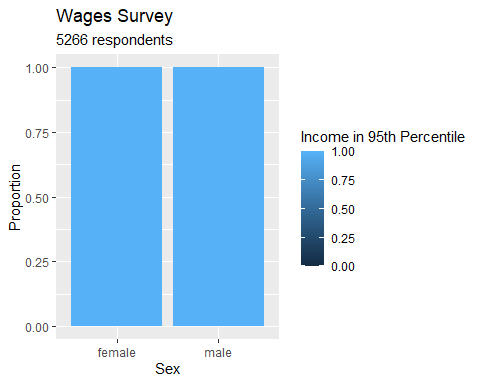
#Get the value for the 95th percentile in income  
income\_95th\_percentile <- quantile(wages$income, 0.95)  
  
#Create a new column called Y that is either 1 or 0 for Income scores  
#Create the model  
wages <- wages %>% mutate(Y = ifelse(income >= cutoff, 1, 0))  
wages <- wages %>% mutate(factor\_sex = ifelse(sex == "male", 1, 0))  
  
df.g <- wages %>% group\_by(Y) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))   
  
ggplot(data=df.g, aes(x = Y, y = prop)) +   
 geom\_bar(stat = "identity") +   
 labs(x="Income in 95th Percentile",fill="",y="Proportion")



df.g <- wages %>% group\_by(sex,Y) %>% summarise(n=n()) %>% mutate(prop = n/sum(n))

## `summarise()` has grouped output by 'sex'. You can override using the `.groups`  
## argument.

df.g %>% ggplot(aes(x=sex,y=prop,fill=Y)) +  
 geom\_col() +  
 labs(title="Wages Survey",  
 subtitle="5266 respondents",  
 x="Sex",  
 y="Proportion",  
 fill="Income in 95th Percentile")



model\_2 <- glm(Y ~ factor\_sex, data = wages, family = binomial)  
summary(model\_2)

##   
## Call:  
## glm(formula = Y ~ factor\_sex, family = binomial, data = wages)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 7.866 1.000 7.865 3.7e-15 \*\*\*  
## factor\_sex 17.700 4190.428 0.004 0.997   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 19.138 on 5265 degrees of freedom  
## Residual deviance: 17.733 on 5264 degrees of freedom  
## AIC: 21.733  
##   
## Number of Fisher Scoring iterations: 24

confint(model\_2)

## Waiting for profiling to be done...

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## 2.5 % 97.5 %  
## (Intercept) 6.384363 10.73068  
## factor\_sex -908.793163 NA

ggplot(wages, aes(x = sex, y = Y)) +  
 geom\_point(aes(color = factor(Y))) +  
 geom\_smooth(method = "glm", method.args = list(family = binomial), se = FALSE) +   
 labs(x = "Income level",  
 y = "Probability of Income") +  
 theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'

