A2

```
library(MASS)
library(vcd)
```

```
## Loading required package: grid
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4
```

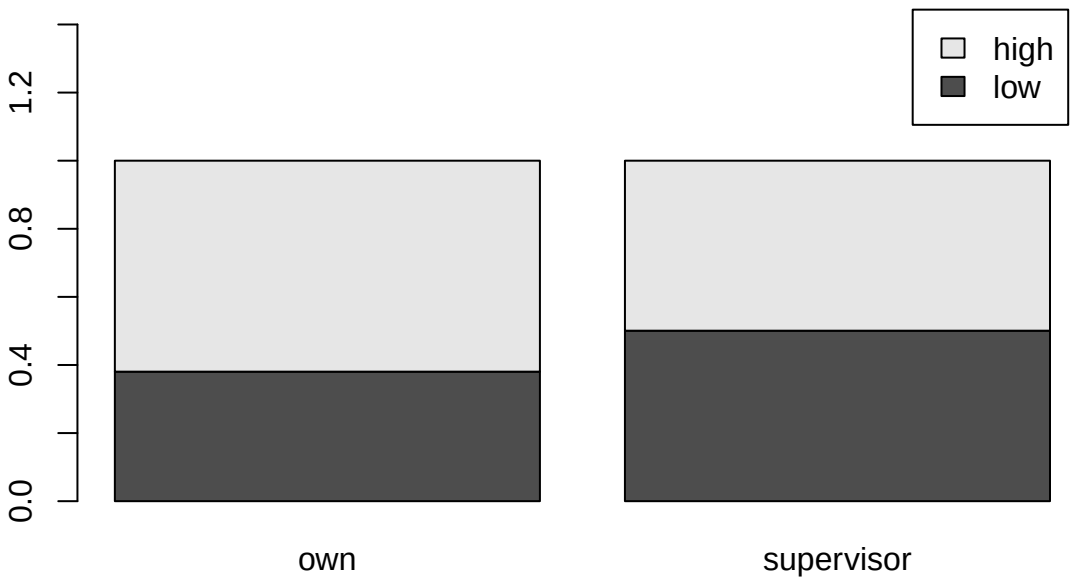
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
library(reshape2)
```

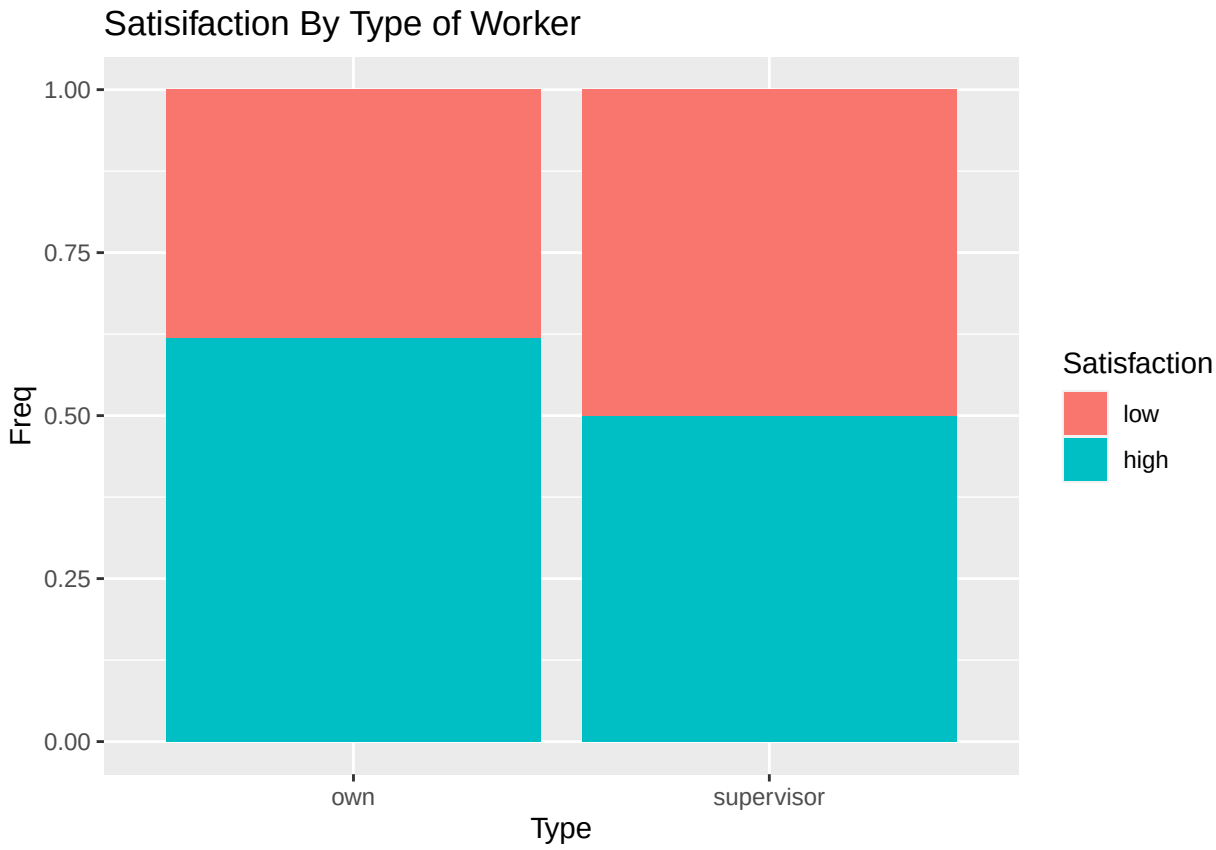
```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

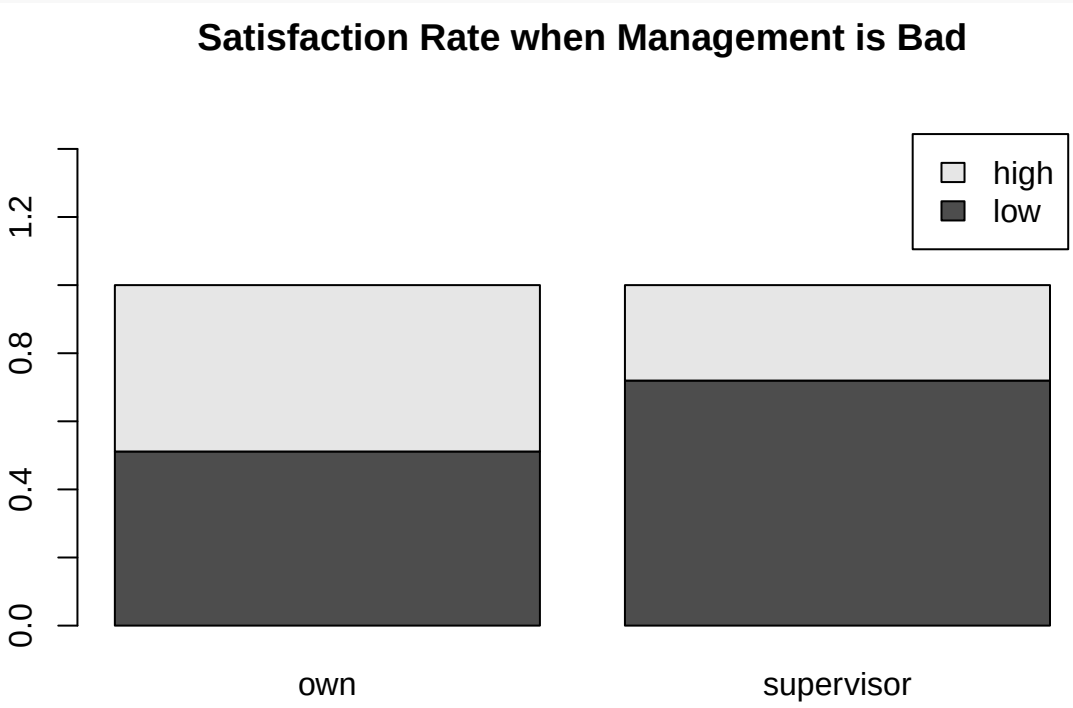
```
# use barplot to compare the overall satisfaction levels of the supervisors and workers.
supervisor<- aggregate(data=JobSatisfaction, Freq~supervisor, sum)
supervisor <- cbind(supervisor,'supervisor')
own <- aggregate(data=JobSatisfaction, Freq~own, sum)
own <- cbind(own,'own')
names(own) <- c('Satisfaction','Frequency','Type')
names(supervisor) <- c('Satisfaction','Frequency','Type')

plot.df <- rbind(supervisor,own)
tab <- xtabs(data=plot.df, Frequency ~ Satisfaction+Type)

# base graphics implementation
barplot(prop.table(tab,2), legend=TRUE, main='Satisfaction Rate by Type of Worker',
        ylim = c(0,1.5))
```



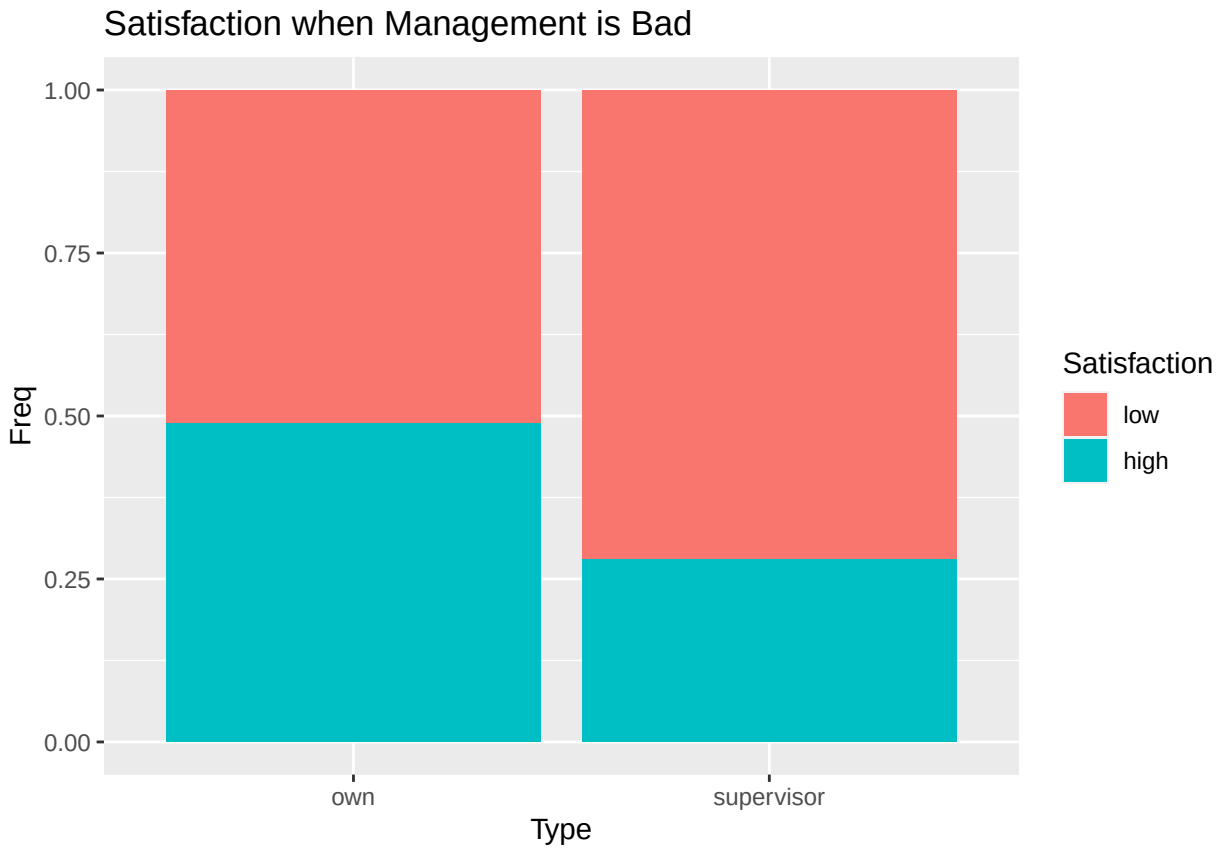

```
# ggplot implementation
ggplot(data=as.data.frame(tab)) + geom_bar(aes(x=Type,y=Freq, fill=Satisfaction),
                                           stat='identity', position='fill') +
  ggtitle("Satisifaction By Type of Worker")
```



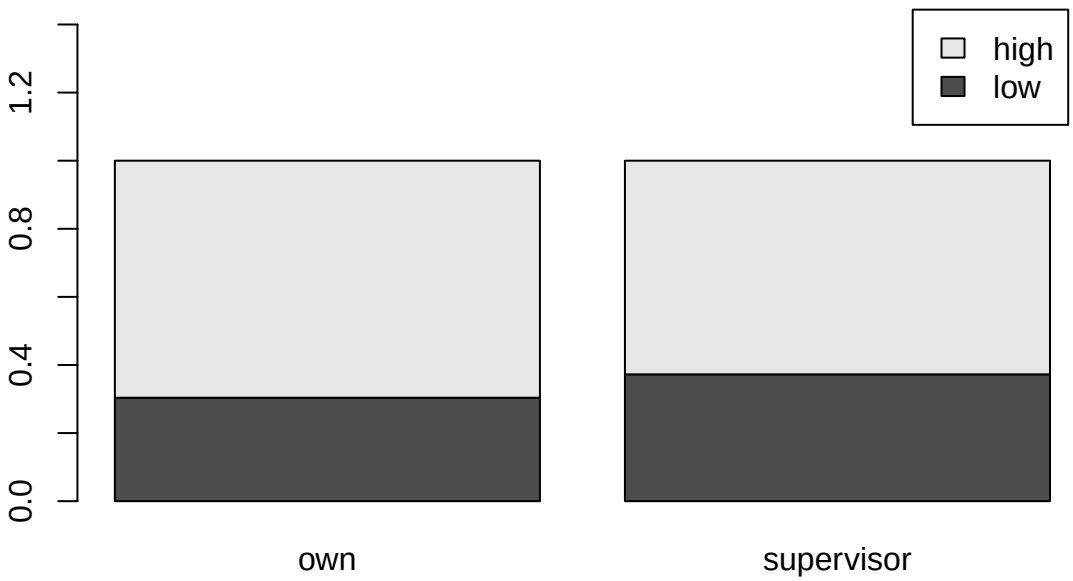
```r
# use barplot to compare the satisfaction levels of the supervisors and workers when management is bad.
subset.job <- filter(JobSatisfaction, management=='bad')
supervisor<- aggregate(data=subset.job, Freq~supervisor, sum)
supervisor <- cbind(supervisor,'supervisor')
own <- aggregate(data=subset.job, Freq~own, sum)
own <- cbind(own,'own')
names(own) <- c('Satisfaction','Frequency','Type')
names(supervisor) <- c('Satisfaction','Frequency','Type')

plot.df <- rbind(supervisor,own)
tab <- xtabs(data=plot.df, Frequency ~ Satisfaction+Type)

barplot(prop.table(tab,2), legend=TRUE, main='Satisfaction Rate when Management is Bad',
        ylim=c(0,1.5))
```



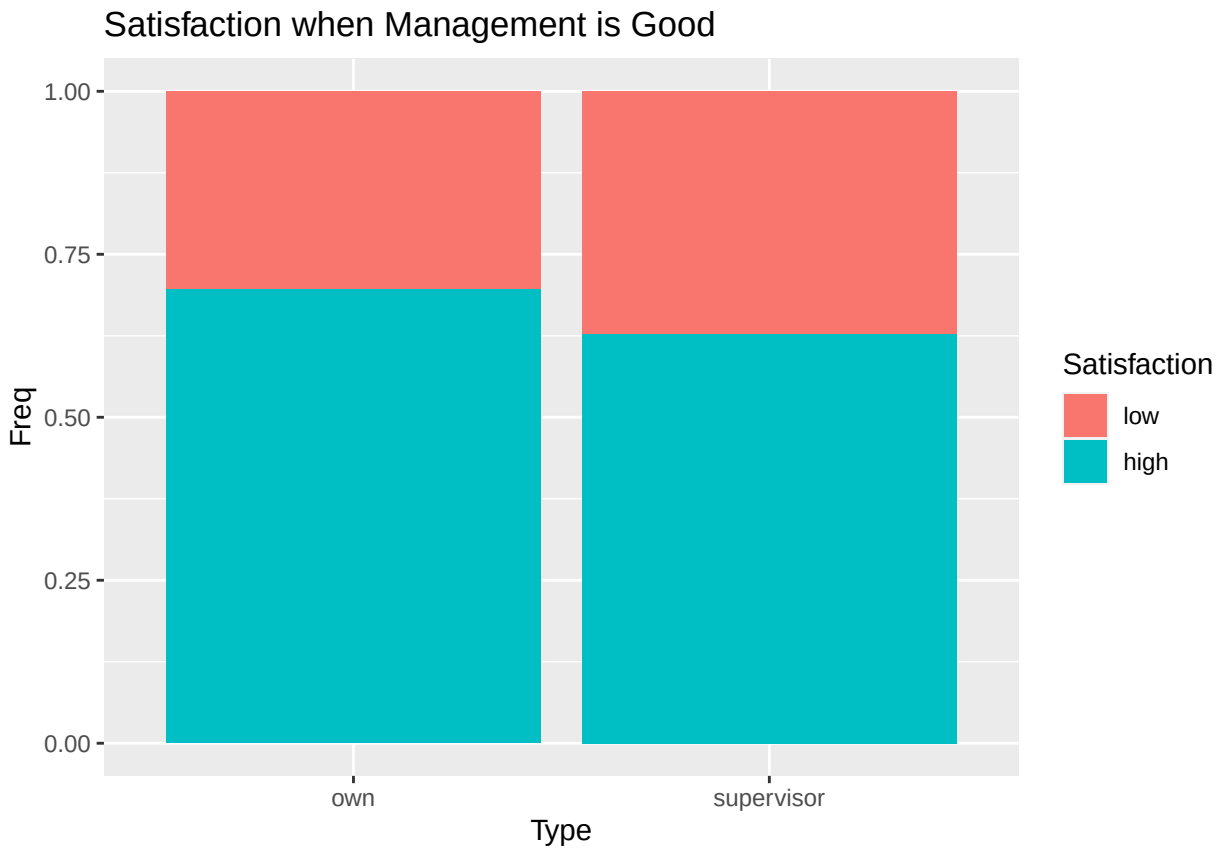

```r
ggplot(data=as.data.frame(tab)) + geom_bar(aes(x=Type,y=Freq, fill=Satisfaction),
                                           stat='identity', position='fill') +
  ggtitle('Satisfaction when Management is Bad')
```




```
# use barplot to compare the satisfaction levels of the supervisors and workers when management is good
subset.job <- filter(JobSatisfaction, management=='good')
supervisor<- aggregate(data=subset.job, Freq~supervisor, sum)
supervisor <- cbind(supervisor,'supervisor')
own <- aggregate(data=subset.job, Freq~own, sum)
own <- cbind(own,'own')
names(own) <- c('Satisfaction','Frequency','Type')
names(supervisor) <- c('Satisfaction','Frequency','Type')

plot.df <- rbind(supervisor,own)
tab <- xtabs(data=plot.df, Frequency ~ Satisfaction+Type)

barplot(prop.table(tab,2), legend=TRUE, main='Satisfaction Rate when Management is Good',
        ylim = c(0,1.5))
```



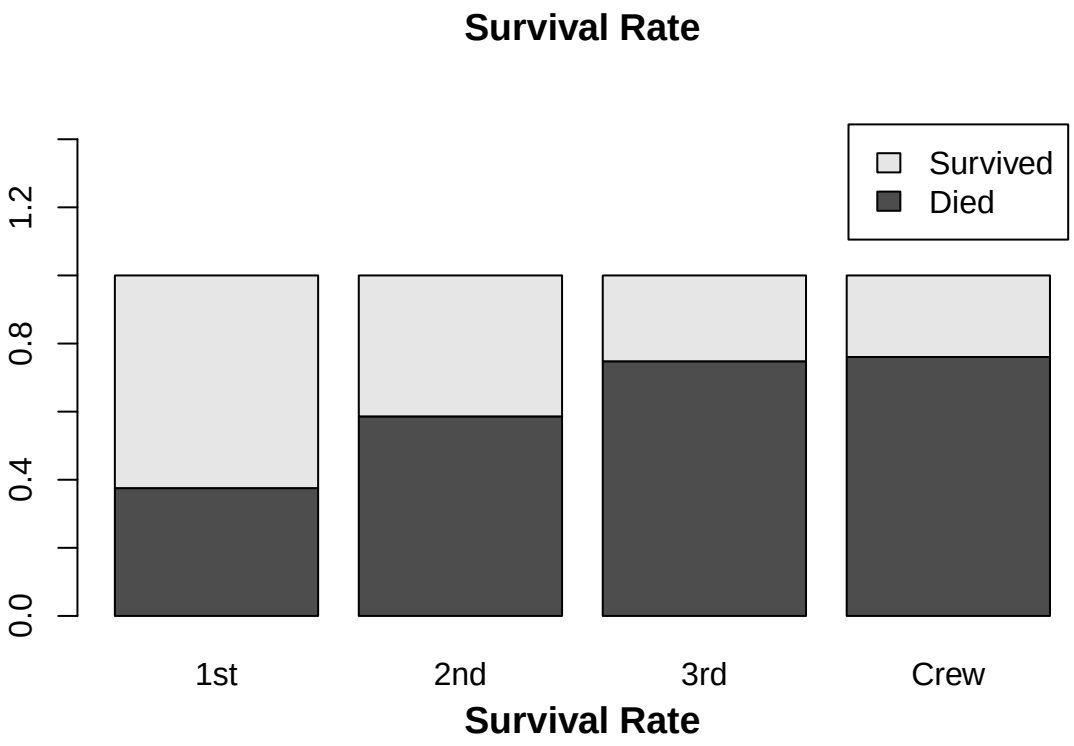

```
ggplot(data=as.data.frame(tab)) + geom_bar(aes(x=Type,y=Freq, fill=Satisfaction),
                                           stat='identity', position='fill') +
  ggtitle('Satisfaction when Management is Good')
```

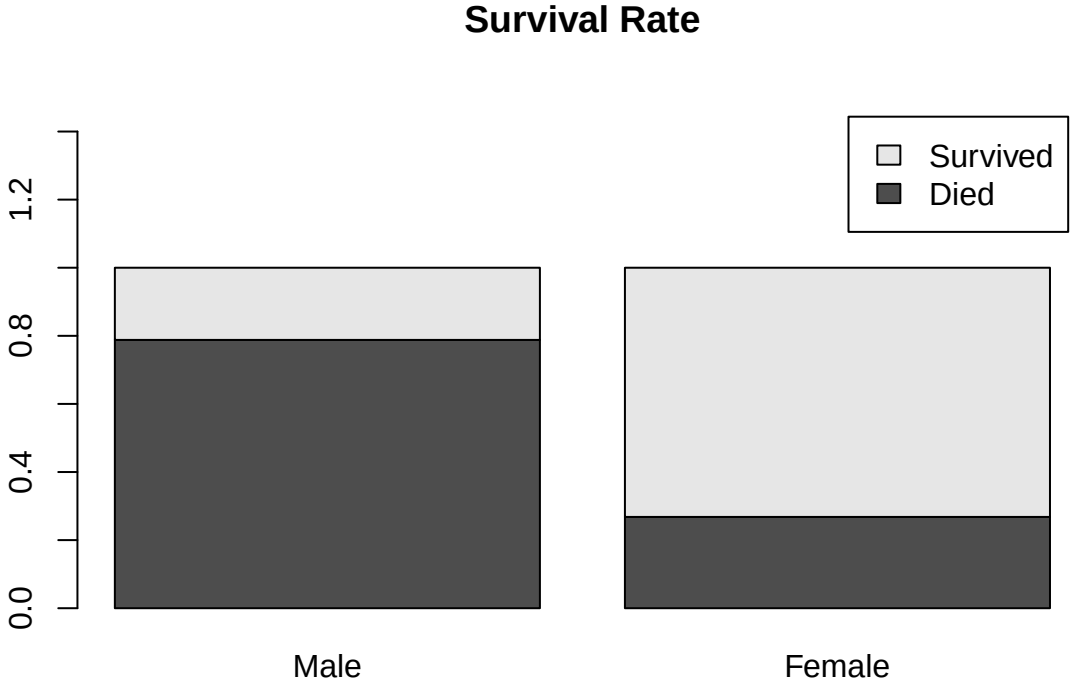


Management has a lower satisfaction rate than workers. Having bad management reduces the rate of

satisfaction in both supervisors and workers.

```
# relationship between "class" and "survived"

barplot.t <- function(num) {
  barplot(prop.table(apply(Titanic,c(4,num),sum),2),
        main='Survival Rate', legend.text =  c('Died', 'Survived'),
        ylim=c(0,1.5))
}

barplot.t(1)
```





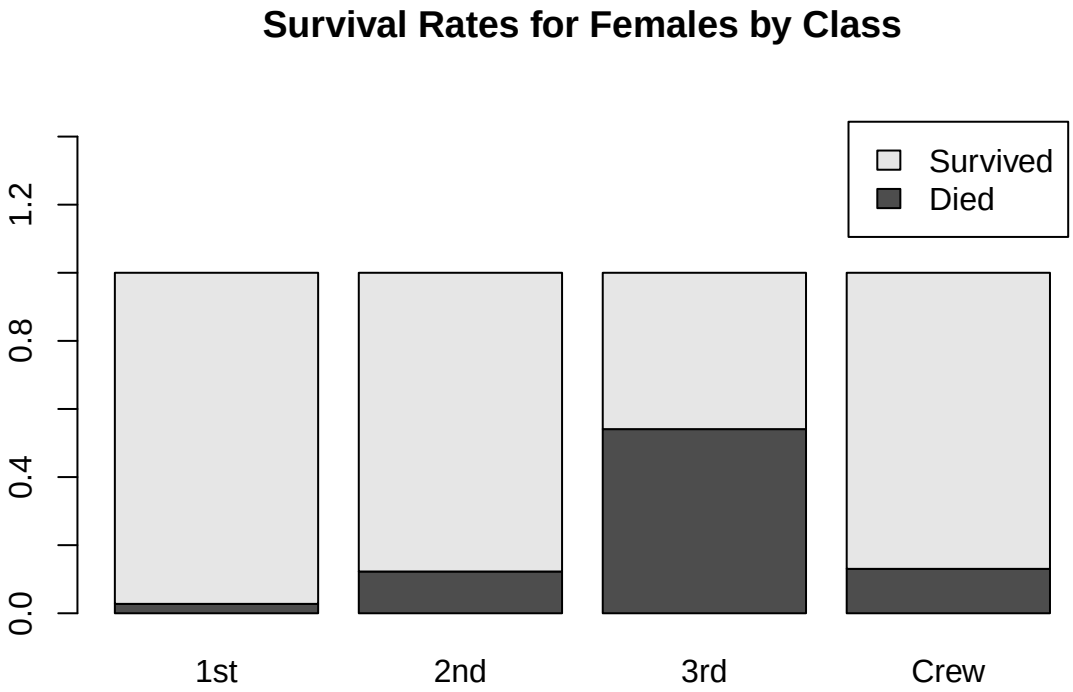
```r
#relationship between "class" and "survived" for Females
s.data <- melt(Titanic) %>% filter(Sex=='Female')

barplot(prop.table(xtabs(data=s.data, value~Survived + Class),2),
        main='Survival Rates for Females by Class', legend=TRUE,
        legend.text = c('Died','Survived'), ylim= c(0,1.5))
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "legend" is not a graphical
## parameter
```

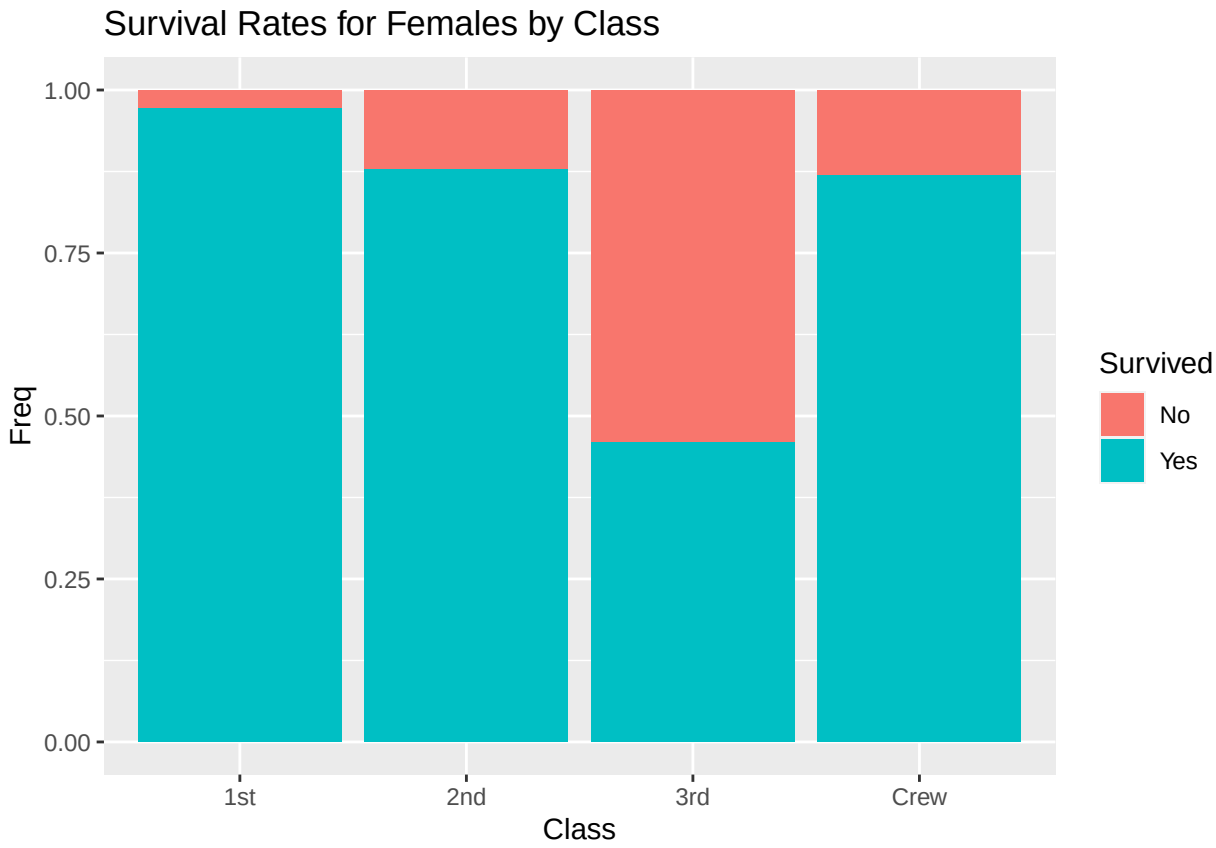
```
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : "legend" is not a graphical parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "legend" is not a graphical parameter
```

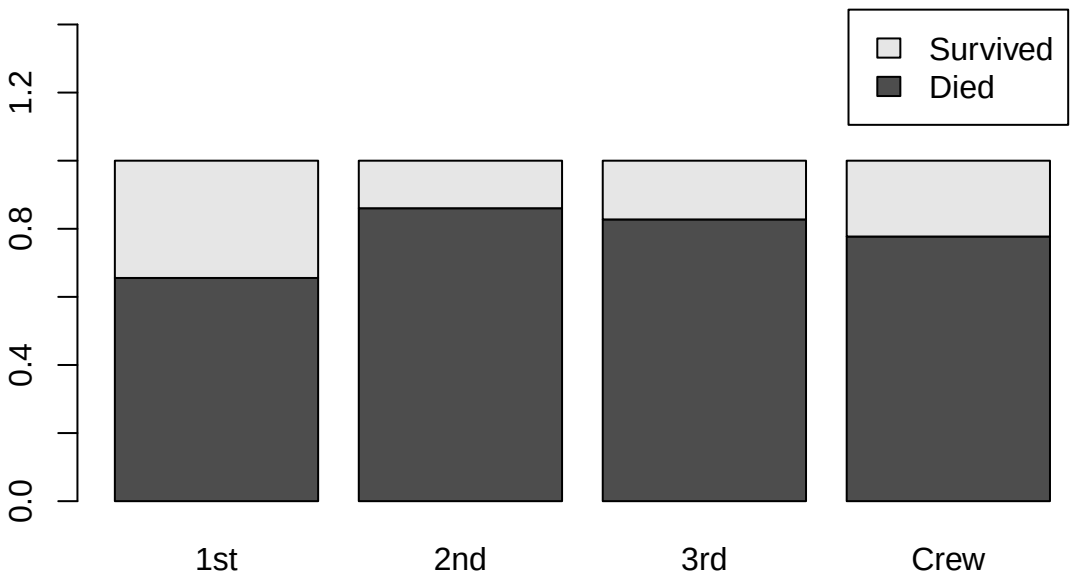
```
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): "legend" is not
## a graphical parameter
```




```r
dat <- as.data.frame(prop.table(xtabs(data=s.data, value~Survived + Class),2))

ggplot(data=dat) + geom_bar(aes(x=Class,y=Freq, fill=Survived),stat='identity') +
  ggtitle('Survival Rates for Females by Class')
```




```
#relationship between "class" and "survived" for Males
s.data <- melt(Titanic) %>% filter(Sex=='Male')

# base graphics
barplot(prop.table(xtabs(data=s.data, value~Survived + Class),2),
        main='Survival Rates for Males by Class',
        legend.text = c('Died','Survived'),
        ylim = c(0,1.5))
```




```
# ggplot
dat <- as.data.frame(prop.table(xtabs(data=s.data, value~Survived + Class),2))
ggplot(data=dat) + geom_bar(aes(x=Class,y=Freq, fill=Survived),stat='identity') +
  ggtitle('Survival Rates for Males by Class')
```

```r
# Identify all the categorical variables and change them into factors.
summary(birthwt)
```

```
##       low              age              lwt             race      
##  Min.   :0.0000   Min.   :14.00   Min.   : 80.0   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:19.00   1st Qu.:110.0   1st Qu.:1.000  
##  Median :0.0000   Median :23.00   Median :121.0   Median :1.000  
##  Mean   :0.3122   Mean   :23.24   Mean   :129.8   Mean   :1.847  
##  3rd Qu.:1.0000   3rd Qu.:26.00   3rd Qu.:140.0   3rd Qu.:3.000  
##  Max.   :1.0000   Max.   :45.00   Max.   :250.0   Max.   :3.000  
##      smoke            ptl               ht                ui        
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000  
##  Mean   :0.3915   Mean   :0.1958   Mean   :0.06349   Mean   :0.1481  
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000  
##  Max.   :1.0000   Max.   :3.0000   Max.   :1.00000   Max.   :1.0000  
##       ftv              bwt        
##  Min.   :0.0000   Min.   : 709  
##  1st Qu.:0.0000   1st Qu.:2414  
##  Median :0.0000   Median :2977  
##  Mean   :0.7937   Mean   :2945  
##  3rd Qu.:1.0000   3rd Qu.:3487  
##  Max.   :6.0000   Max.   :4990
```

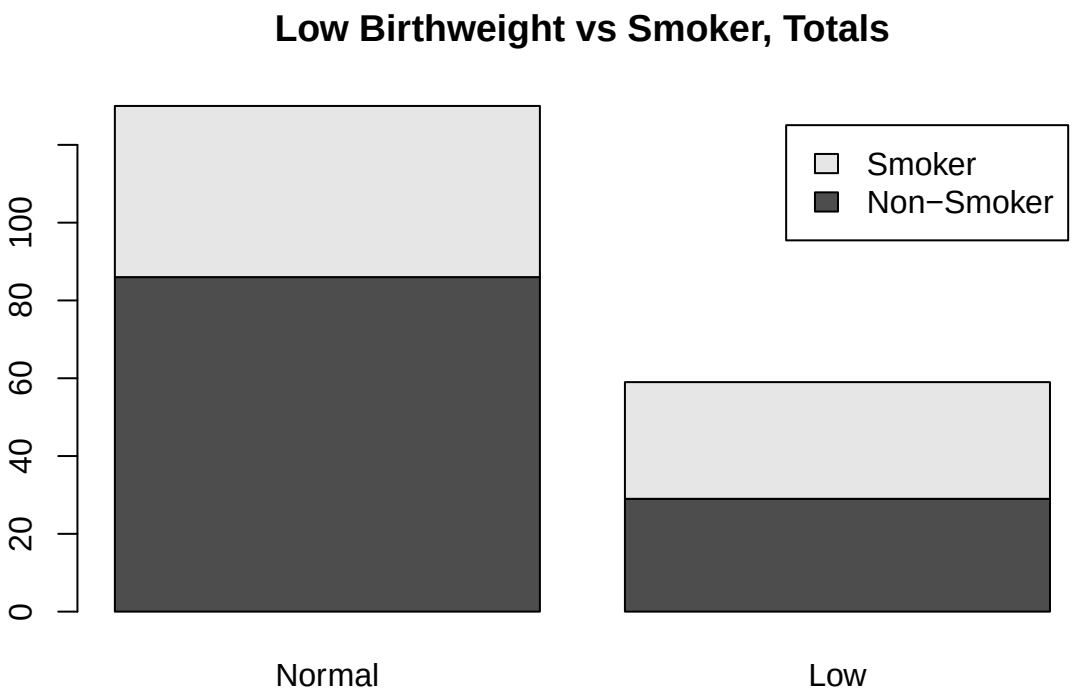
```r
head(birthwt)
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
## 88   0  21 108    1     1   0  0  1   2 2594
## 89   0  18 107    1     1   0  0  1   0 2600
## 91   0  21 124    3     0   0  0  0   0 2622
```

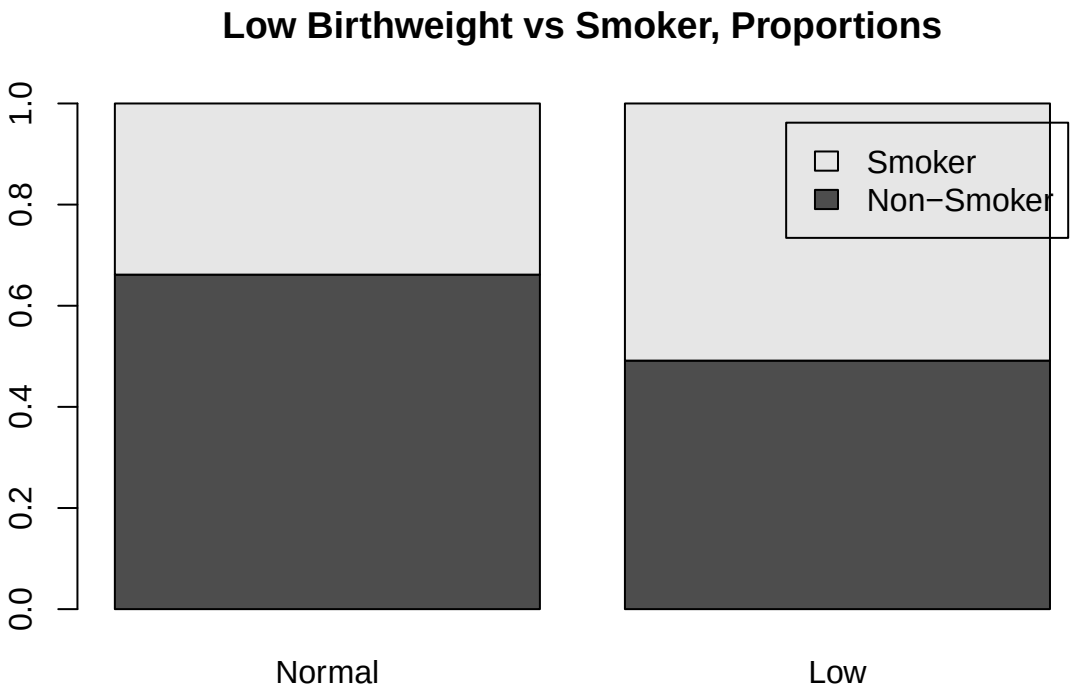
```r
data <- birthwt

# categorical according to viginette
cols <- c("low", "race", "smoke", "ht",'ui')
# convert to factor
data[cols] <- lapply(data[cols], factor)
```

```r
#study the relationship between variables "smoke" and "low" by constructing suitable tables and proport
df.smoke.low <- data[,c('low', 'smoke')]
df.smoke.low <- mutate(df.smoke.low, birthweight=recode(low,
                                '0'='Normal',
                                '1'='Low'))

df.smoke.low <- mutate(df.smoke.low, smoke=recode(smoke,
                                '0'='Non-Smoker',
                                '1'='Smoker'))

tab <- table(df.smoke.low[,c('smoke','birthweight')])

# base graphics implementation
```

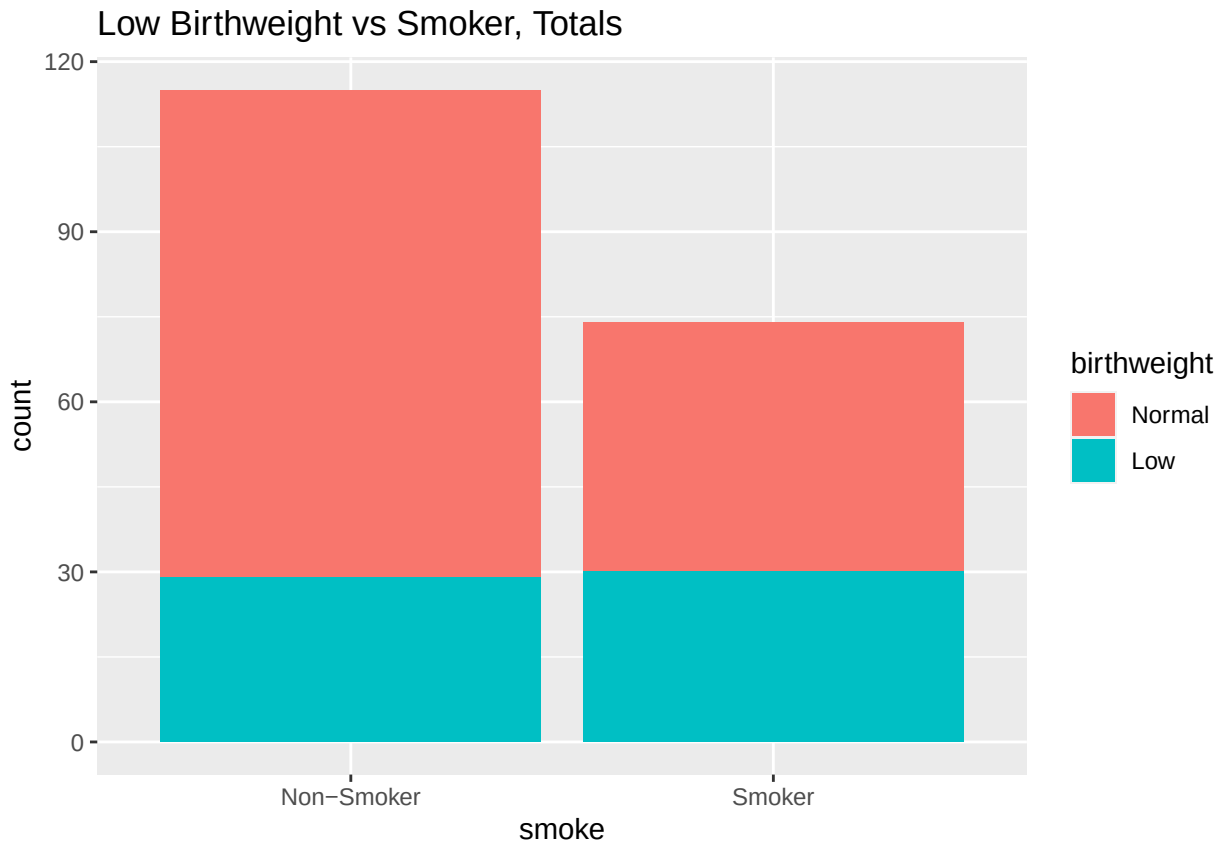
```r
barplot(tab, main='Low Birthweight vs Smoker, Totals', legend=TRUE)
```



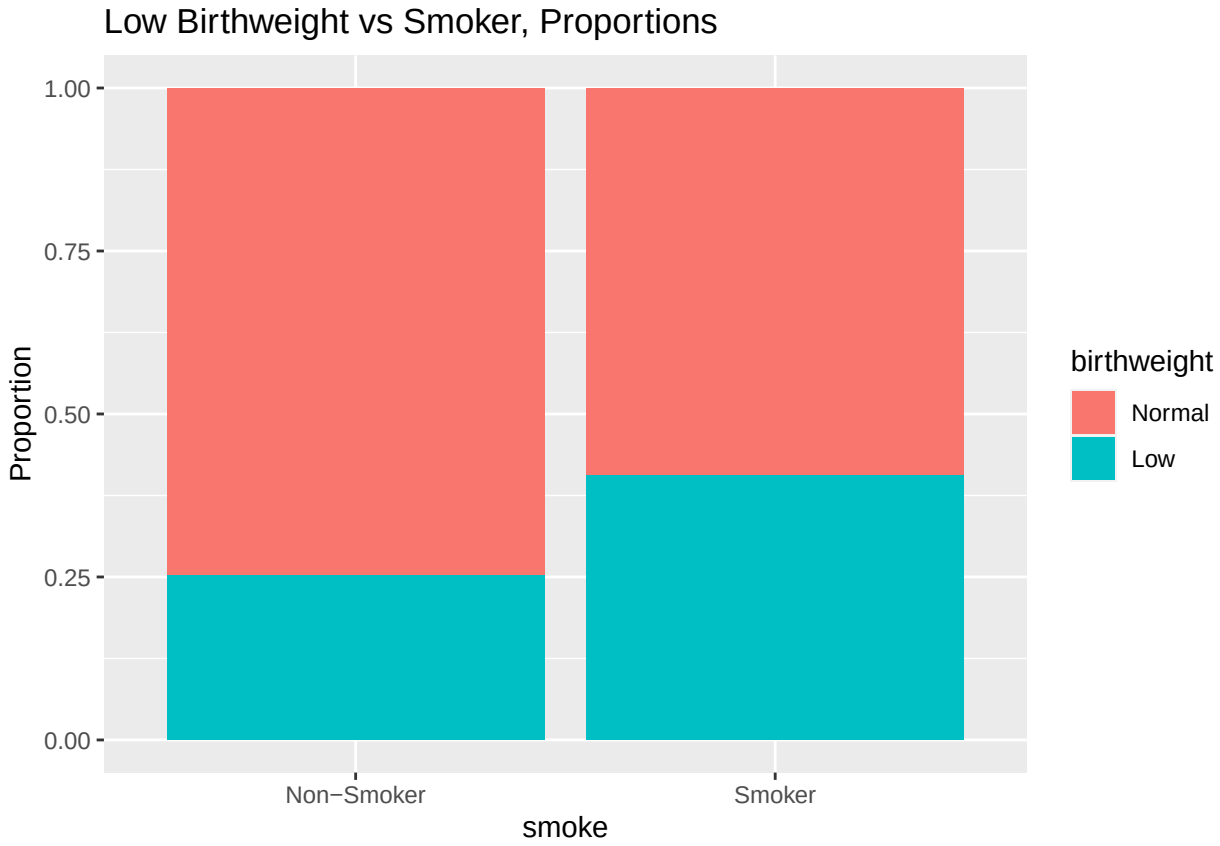

```r
barplot(prop.table(tab,2), legend=TRUE, main='Low Birthweight vs Smoker, Proportions')
```




```r
# ggplot implementation
ggplot(data=df.smoke.low) + geom_bar(aes(x=smoke, fill=birthweight)) +
  ggtitle('Low Birthweight vs Smoker, Totals')
```




```
ggplot(data=df.smoke.low) + geom_bar(aes(x=smoke, fill=birthweight),
                                     position='fill') + ylab('Proportion') +
  ggtitle('Low Birthweight vs Smoker, Proportions')
```



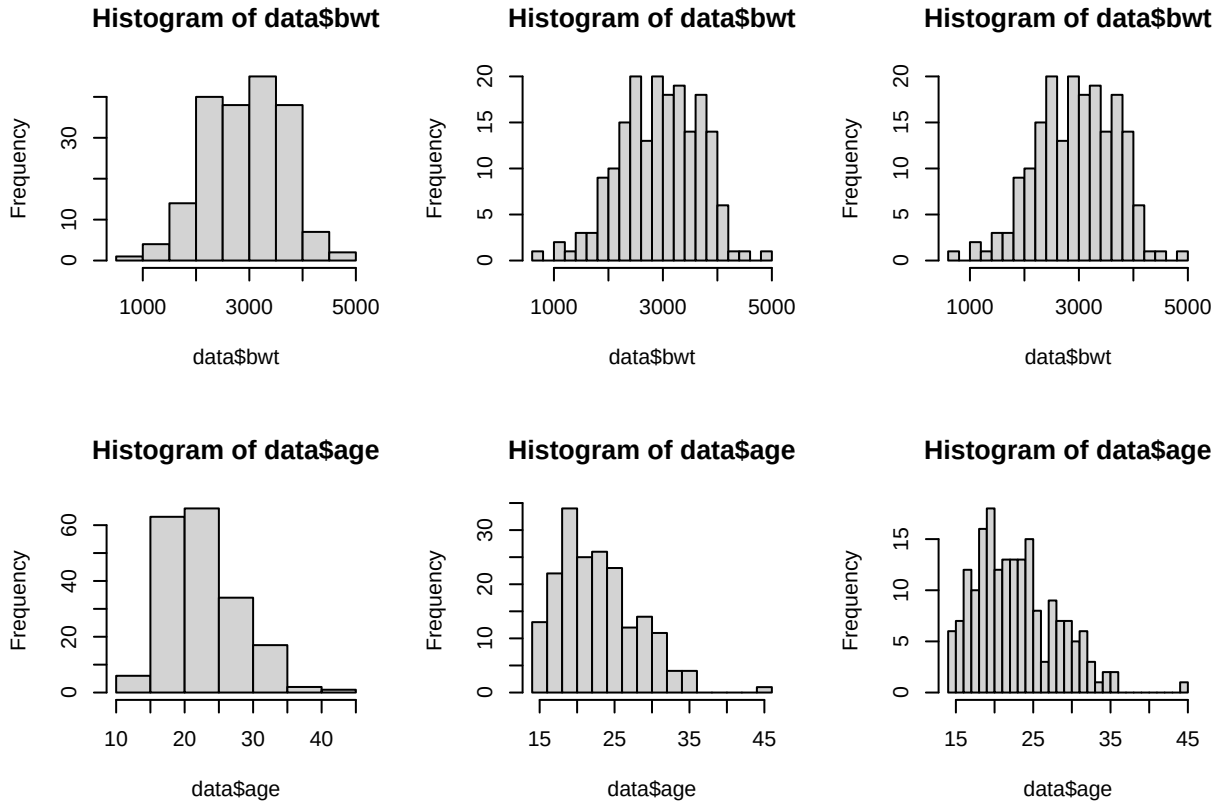

```
#plot the histogram for "bwt" and "age". use three different bandwidth for each histogram.

par(mfrow=c(2,3))
# base graphics implementation
c(10,20,30) %>% lapply(function(x) hist(data$bwt, breaks=x))
```

```
## [[1]]
## $breaks
##  [1]  500 1000 1500 2000 2500 3000 3500 4000 4500 5000
##
## $counts
## [1]  1  4 14 40 38 45 38  7  2
##
## $density
## [1] 1.058201e-05 4.232804e-05 1.481481e-04 4.232804e-04 4.021164e-04
## [6] 4.761905e-04 4.021164e-04 7.407407e-05 2.116402e-05
##
## $mids
## [1]  750 1250 1750 2250 2750 3250 3750 4250 4750
##
## $xname
## [1] "data$bwt"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
```

```
## [1] "histogram"
##
## [[2]]
## $breaks
##  [1]  600  800 1000 1200 1400 1600 1800 2000 2200 2400 2600 2800 3000 3200 3400
## [16] 3600 3800 4000 4200 4400 4600 4800 5000
##
## $counts
##  [1]  1  0  2  1  3  3  9 10 15 20 13 20 18 19 14 18 14  6  1  1  0  1
##
## $density
##  [1] 2.645503e-05 0.000000e+00 5.291005e-05 2.645503e-05 7.936508e-05
##  [6] 7.936508e-05 2.380952e-04 2.645503e-04 3.968254e-04 5.291005e-04
## [11] 3.439153e-04 5.291005e-04 4.761905e-04 5.026455e-04 3.703704e-04
## [16] 4.761905e-04 3.703704e-04 1.587302e-04 2.645503e-05 2.645503e-05
## [21] 0.000000e+00 2.645503e-05
##
## $mids
##  [1]  700  900 1100 1300 1500 1700 1900 2100 2300 2500 2700 2900 3100 3300 3500
## [16] 3700 3900 4100 4300 4500 4700 4900
##
## $xname
## [1] "data$bwt"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## [[3]]
## $breaks
##  [1]  600  800 1000 1200 1400 1600 1800 2000 2200 2400 2600 2800 3000 3200 3400
## [16] 3600 3800 4000 4200 4400 4600 4800 5000
##
## $counts
##  [1]  1  0  2  1  3  3  9 10 15 20 13 20 18 19 14 18 14  6  1  1  0  1
##
## $density
##  [1] 2.645503e-05 0.000000e+00 5.291005e-05 2.645503e-05 7.936508e-05
##  [6] 7.936508e-05 2.380952e-04 2.645503e-04 3.968254e-04 5.291005e-04
## [11] 3.439153e-04 5.291005e-04 4.761905e-04 5.026455e-04 3.703704e-04
## [16] 4.761905e-04 3.703704e-04 1.587302e-04 2.645503e-05 2.645503e-05
## [21] 0.000000e+00 2.645503e-05
##
## $mids
##  [1]  700  900 1100 1300 1500 1700 1900 2100 2300 2500 2700 2900 3100 3300 3500
## [16] 3700 3900 4100 4300 4500 4700 4900
##
## $xname
## [1] "data$bwt"
##
## $equidist
## [1] TRUE
```

```
##
## attr(,"class")
## [1] "histogram"
```

```
c(10,20,30) %>% lapply(function(x) hist(data$age, breaks=x))
```

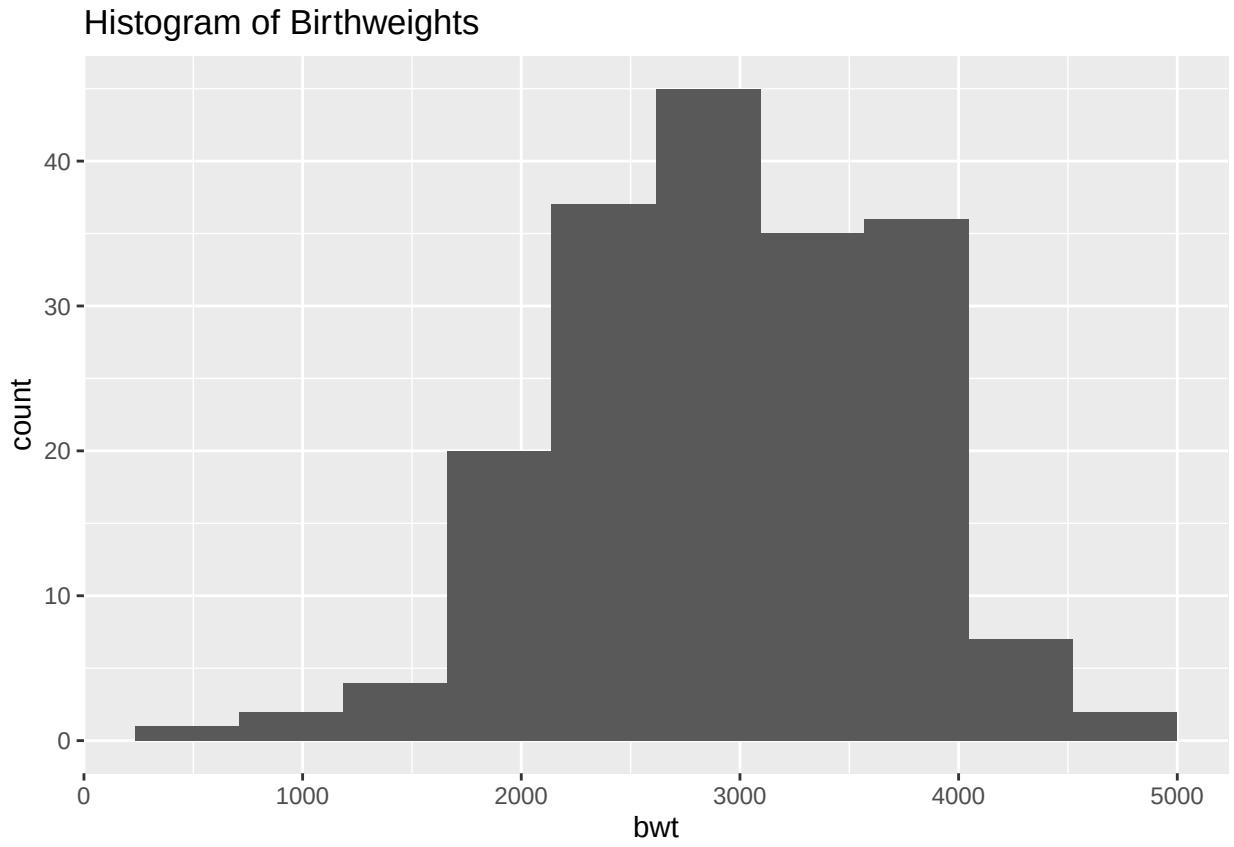



```
## [[1]]
## $breaks
## [1] 10 15 20 25 30 35 40 45
##
## $counts
## [1]  6 63 66 34 17  2  1
##
## $density
## [1] 0.006349206 0.066666667 0.069841270 0.035978836 0.017989418 0.002116402
## [7] 0.001058201
##
## $mids
## [1] 12.5 17.5 22.5 27.5 32.5 37.5 42.5
##
## $xname
## [1] "data$age"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
```

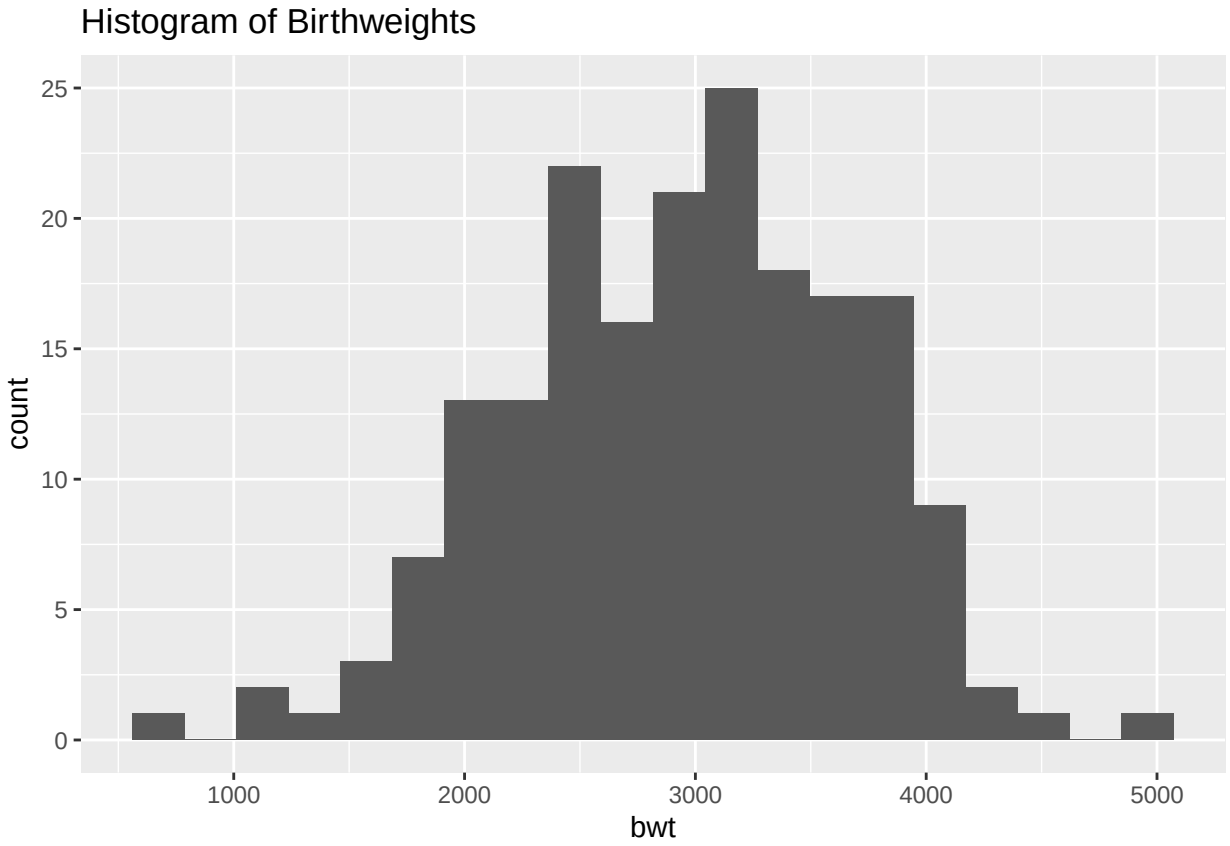
```
## [[2]]
## $breaks
##  [1] 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
##
## $counts
##  [1] 13 22 34 25 26 23 12 14 11  4  4  0  0  0  0  1
##
## $density
##  [1] 0.034391534 0.058201058 0.089947090 0.066137566 0.068783069 0.060846561
##  [7] 0.031746032 0.037037037 0.029100529 0.010582011 0.010582011 0.000000000
## [13] 0.000000000 0.000000000 0.000000000 0.002645503
##
## $mids
##  [1] 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45
##
## $xname
## [1] "data$age"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## [[3]]
## $breaks
##  [1] 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
## [26] 39 40 41 42 43 44 45
##
## $counts
##  [1]  6  7 12 10 16 18 12 13 13 13 15  8  3  9  7  7  5  6  3  1  2  2  0  0  0
## [26]  0  0  0  0  0  1
##
## $density
##  [1] 0.031746032 0.037037037 0.063492063 0.052910053 0.084656085 0.095238095
##  [7] 0.063492063 0.068783069 0.068783069 0.068783069 0.079365079 0.042328042
## [13] 0.015873016 0.047619048 0.037037037 0.037037037 0.026455026 0.031746032
## [19] 0.015873016 0.005291005 0.010582011 0.010582011 0.000000000 0.000000000
## [25] 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
## [31] 0.005291005
##
## $mids
##  [1] 14.5 15.5 16.5 17.5 18.5 19.5 20.5 21.5 22.5 23.5 24.5 25.5 26.5 27.5 28.5
## [16] 29.5 30.5 31.5 32.5 33.5 34.5 35.5 36.5 37.5 38.5 39.5 40.5 41.5 42.5 43.5
## [31] 44.5
##
## $xname
## [1] "data$age"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```r
# ggplot2 implementation
c(10,20,30) %>% lapply(function(x) ggplot(data=data) +
                         geom_histogram(aes(x=bwt), bins=x) +
                         ggtitle('Histogram of Birthweights'))
```

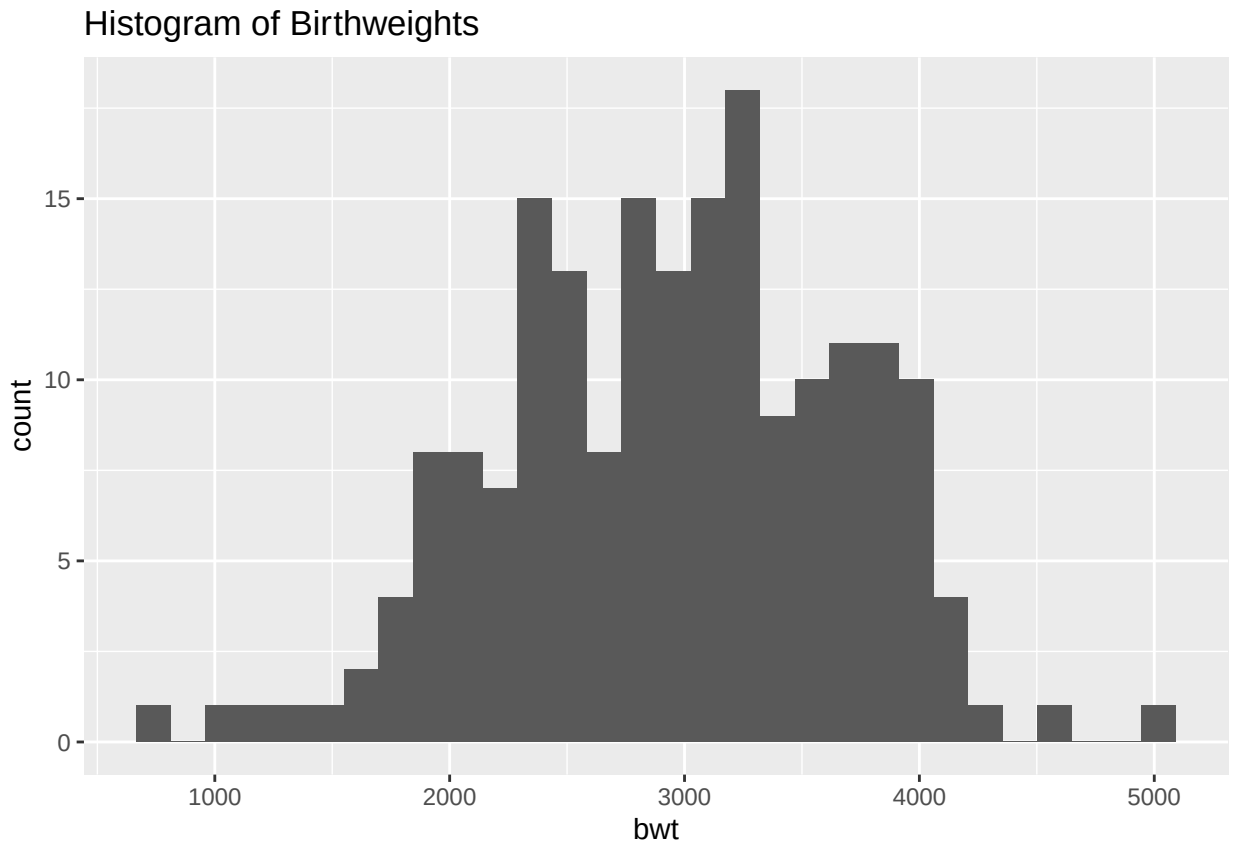
```
## [[1]]
```




```
##
## [[2]]
```

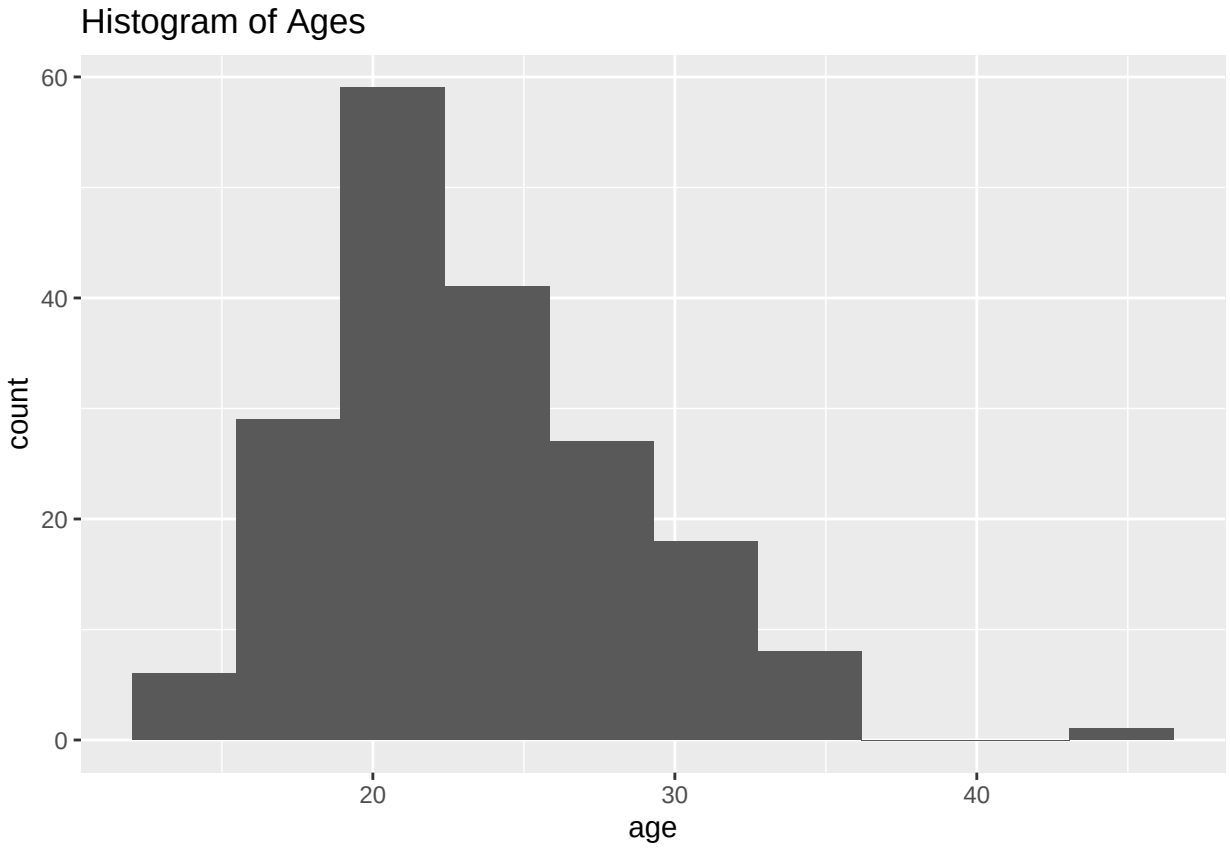

Histogram of Birthweights
count
bwt
0
5
10
15
20
25
1000
2000
3000
4000
5000

```
##
## [[3]]
```

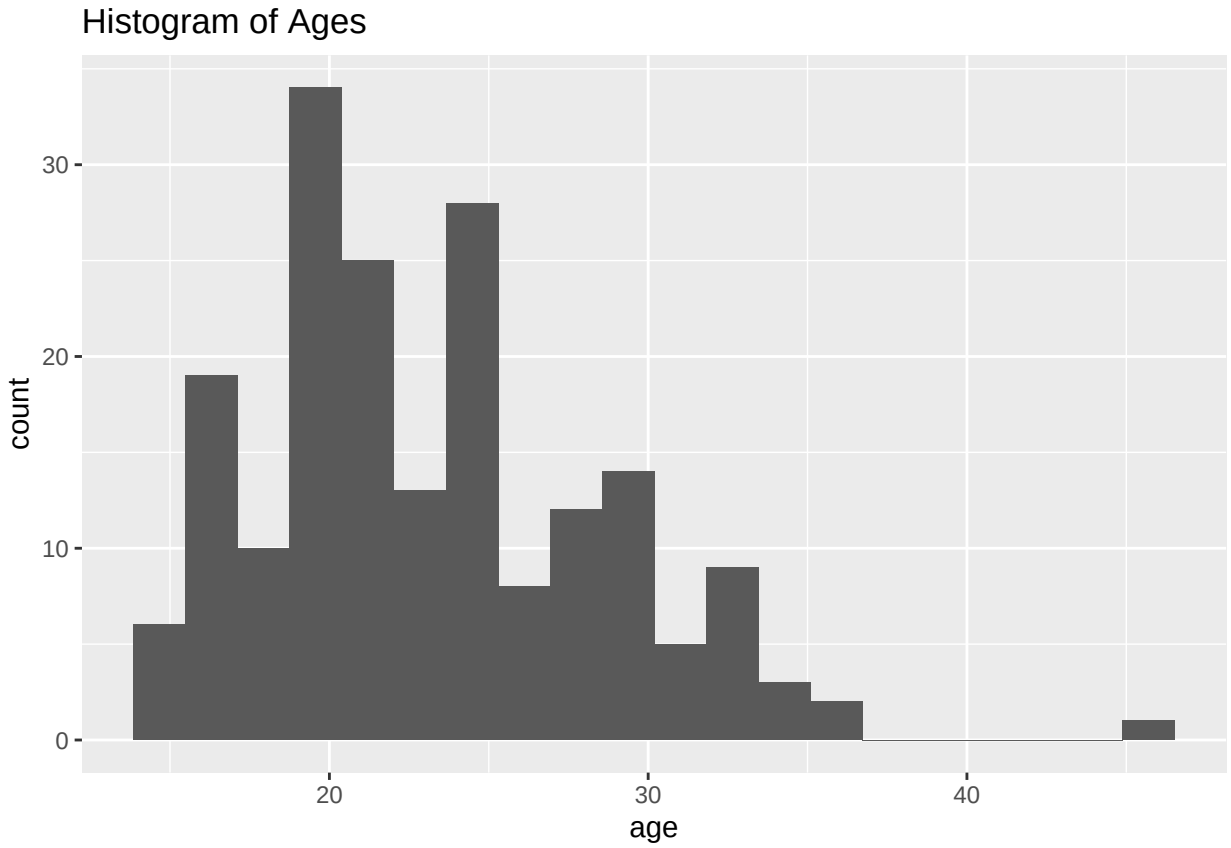



```
c(10,20,30) %>% lapply(function(x) ggplot(data=data) +
                         geom_histogram(aes(x=age), bins=x)+
                         ggtitle('Histogram of Ages'))
```

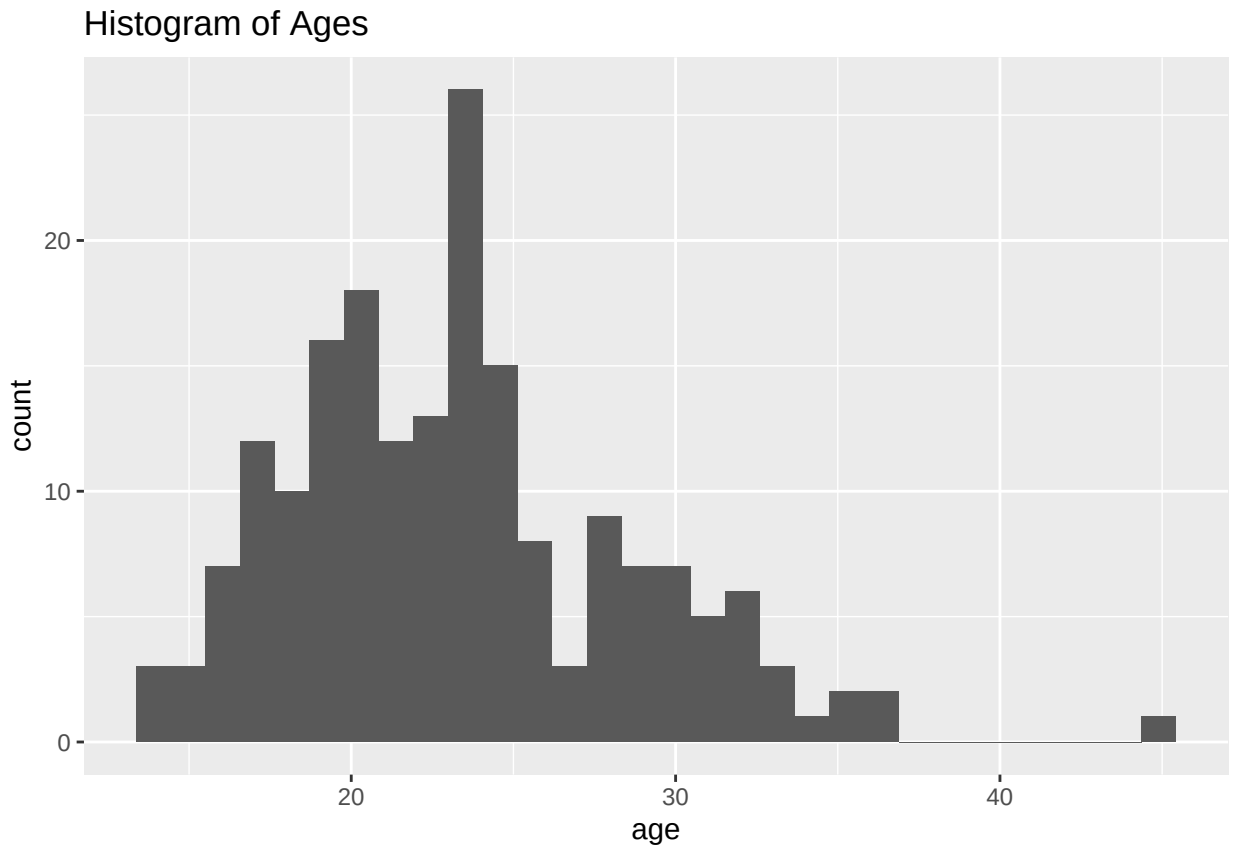
```
## [[1]]
```




```
##
## [[2]]
```




```
##
## [[3]]
```



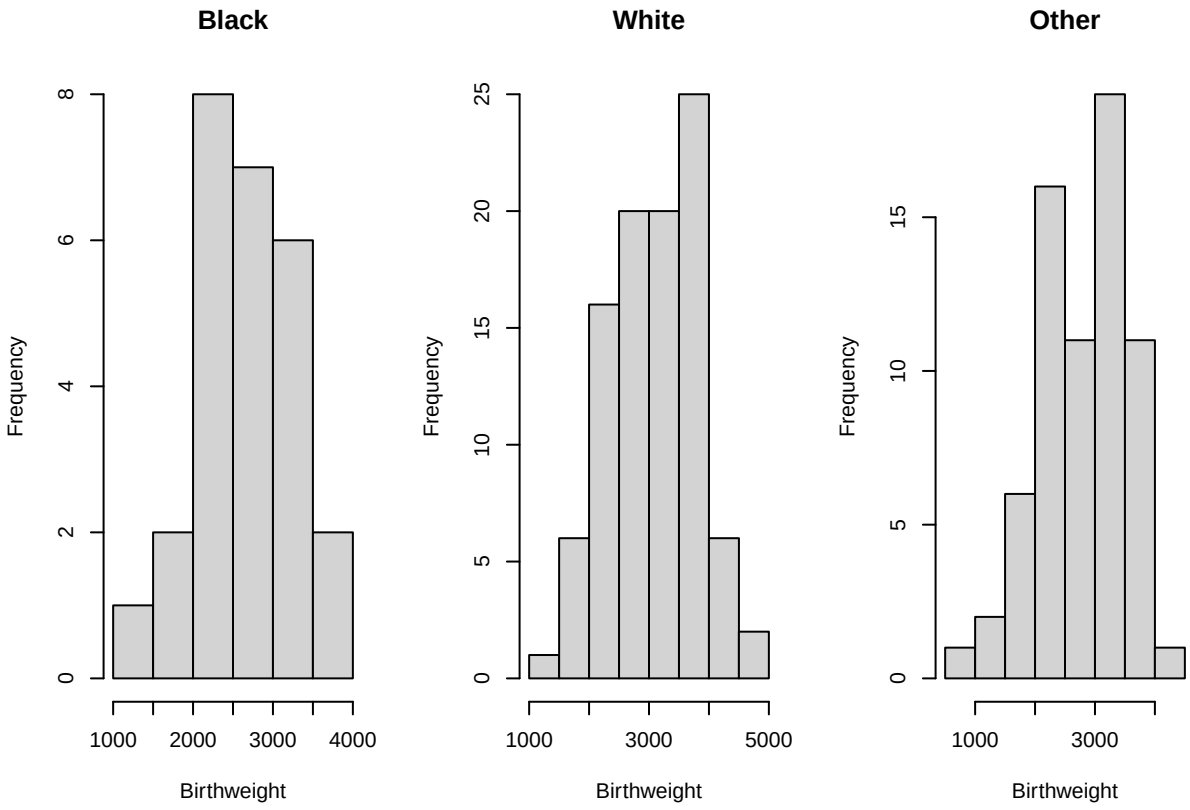

```
#study the relationship between variables "race" and "bwt" using suitable numberical summaries (any kin

df.race.bwt <- data[,c('race', 'bwt')]
df.race.bwt <- mutate(df.race.bwt, race=recode(race,
                                '1'='White',
                                '2'='Black',
                                '3'='Other'))

# numerical summaries
# mean
aggregate(data=df.race.bwt,bwt~race, FUN=mean)
```

```
##    race      bwt
## 1 White 3102.719
## 2 Black 2719.692
## 3 Other 2805.284
```

```
# standard deviation
aggregate(data=df.race.bwt,bwt~race, FUN=sd)
```

```
##    race      bwt
## 1 White 727.8861
## 2 Black 638.6839
## 3 Other 722.1944
```

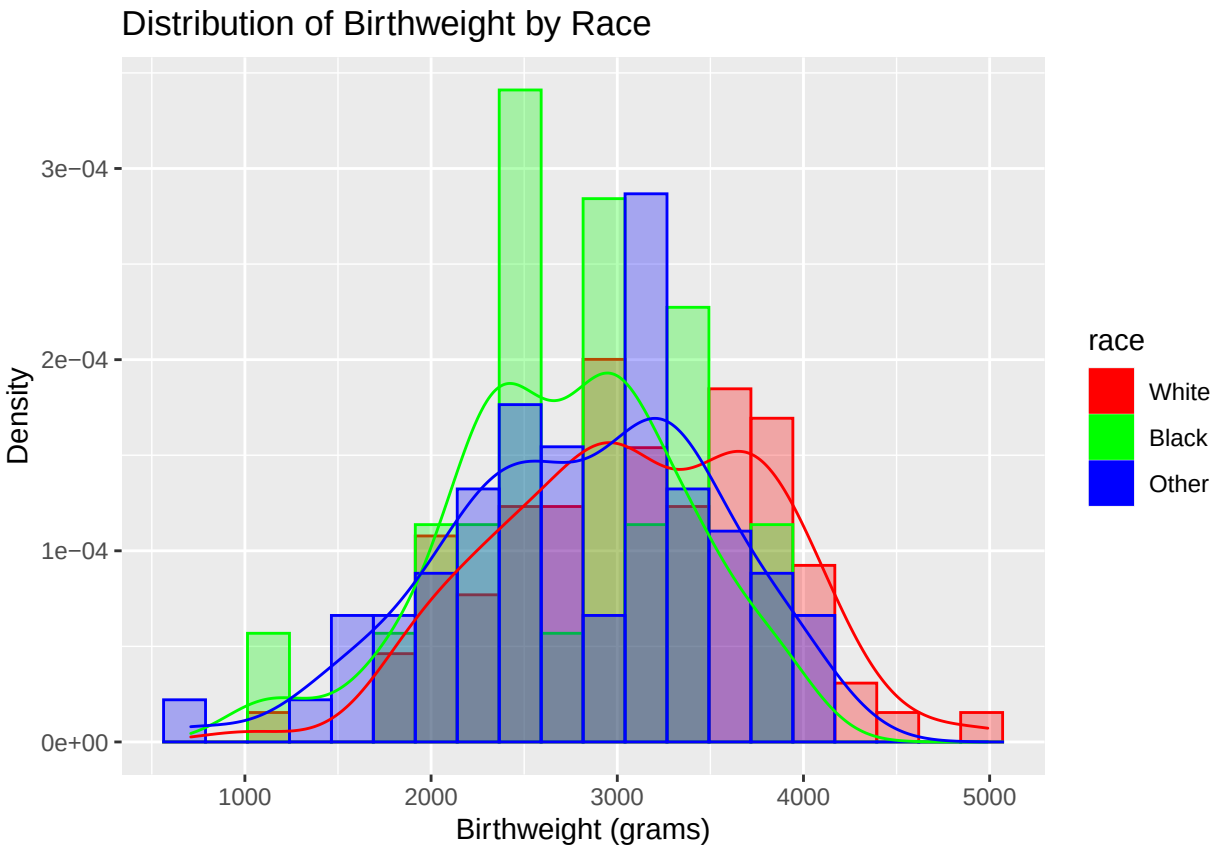
```
# base graphic implementation
par(mfrow=c(1,3))
c('Black','White','Other') %>% lapply(function(x) hist(filter(df.race.bwt, race==x)$bwt,main=x, xlab =
```




```
## [[1]]
## $breaks
## [1] 1000 1500 2000 2500 3000 3500 4000
##
## $counts
## [1] 1 2 8 7 6 2
##
## $density
## [1] 7.692308e-05 1.538462e-04 6.153846e-04 5.384615e-04 4.615385e-04
## [6] 1.538462e-04
##
## $mids
## [1] 1250 1750 2250 2750 3250 3750
##
## $xname
## [1] "filter(df.race.bwt, race == x)$bwt"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## [[2]]
## $breaks
## [1] 1000 1500 2000 2500 3000 3500 4000 4500 5000
##
## $counts
```

```
## [1]  1  6 16 20 20 25  6  2
##
## $density
## [1] 2.083333e-05 1.250000e-04 3.333333e-04 4.166667e-04 4.166667e-04
## [6] 5.208333e-04 1.250000e-04 4.166667e-05
##
## $mids
## [1] 1250 1750 2250 2750 3250 3750 4250 4750
##
## $xname
## [1] "filter(df.race.bwt, race == x)$bwt"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## [[3]]
## $breaks
## [1]  500 1000 1500 2000 2500 3000 3500 4000 4500
##
## $counts
## [1]  1  2  6 16 11 19 11  1
##
## $density
## [1] 2.985075e-05 5.970149e-05 1.791045e-04 4.776119e-04 3.283582e-04
## [6] 5.671642e-04 3.283582e-04 2.985075e-05
##
## $mids
## [1]  750 1250 1750 2250 2750 3250 3750 4250
##
## $xname
## [1] "filter(df.race.bwt, race == x)$bwt"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
# ggplot implementation
ggplot(data=df.race.bwt) + geom_histogram(aes(x=bwt,color=race, fill=race, y=0.333*..density..), alpha=(
  scale_color_manual(values = rainbow(3)) +
  scale_fill_manual(values = rainbow(3)) + geom_density(aes(x=bwt,color=race, y=0.333*..density..)) +
  ggtitle('Distribution of Birthweight by Race') + ylab('Density') + xlab('Birthweight (grams)')
```



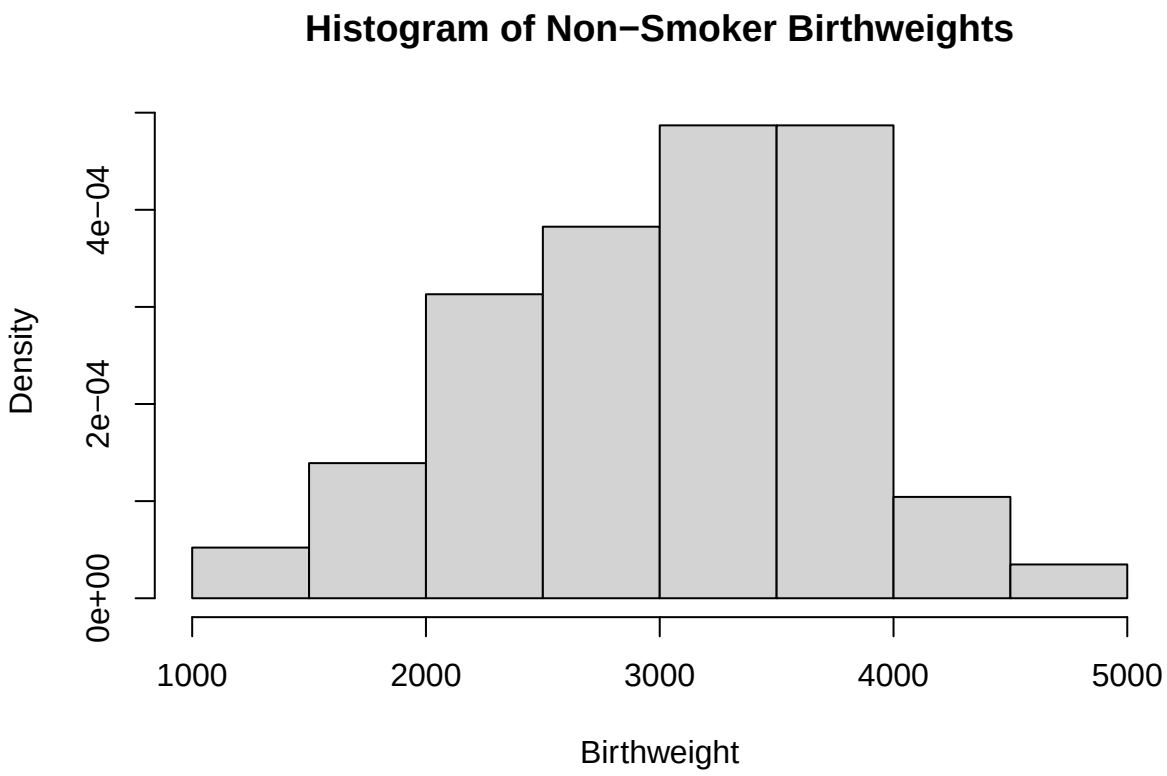

```
#study the relationship between variables "smoke" and "bwt" using suitable numberical summaries (any ki

df.smoke.bwt <- data[,c('smoke', 'bwt')]
df.smoke.bwt <- mutate(df.smoke.bwt, smoke=recode(smoke,
                                  '0'='Non-Smoker',
                                  '1'='Smoker'))

# numerical summaries
# mean
aggregate(data=df.smoke.bwt,bwt~smoke, FUN=mean)
```

```
##        smoke      bwt
## 1 Non-Smoker 3055.696
## 2     Smoker 2771.919
```

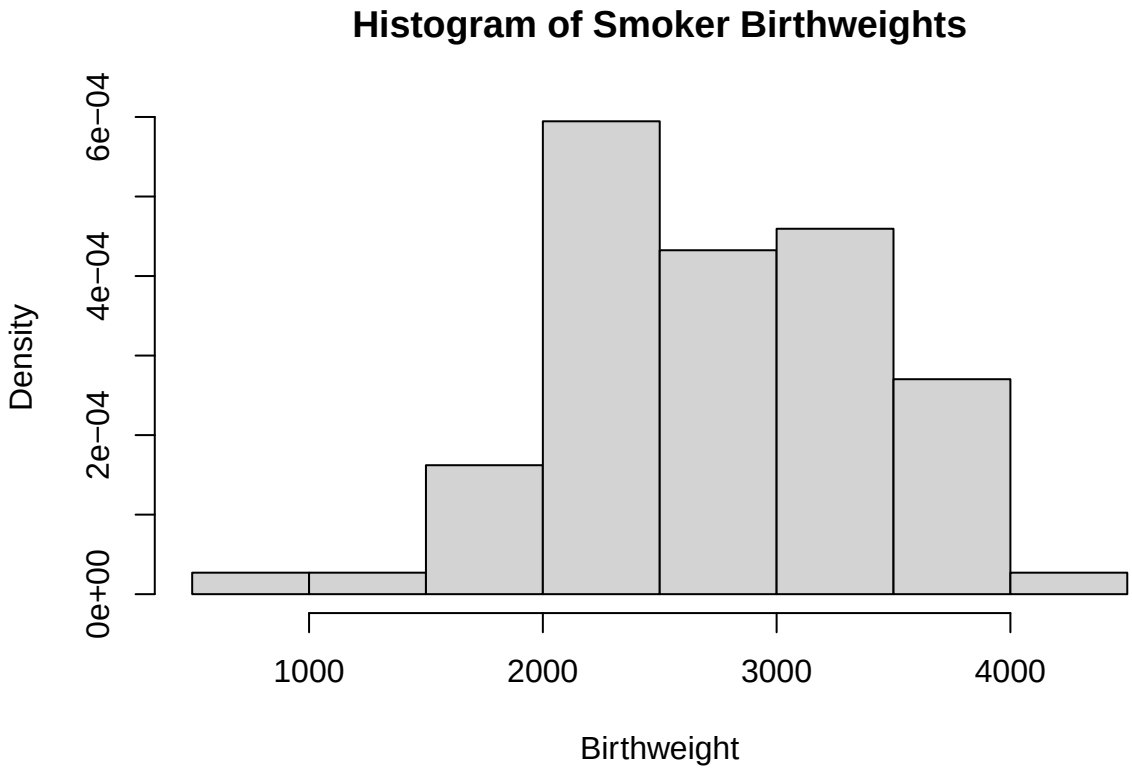
```
# standard deviation
aggregate(data=df.smoke.bwt,bwt~smoke, FUN=sd)
```

```
##        smoke      bwt
## 1 Non-Smoker 752.6566
## 2     Smoker 659.6349
```

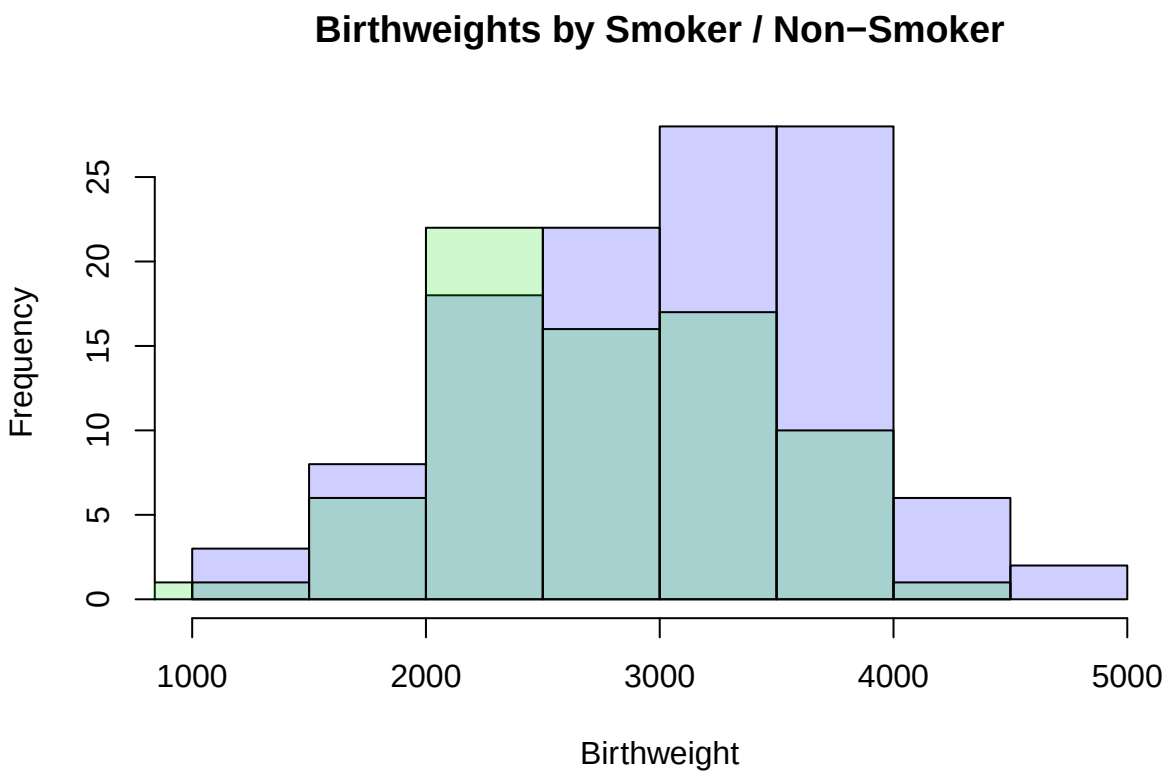
```
# base graphics implementation
hg1 <- hist(filter(df.smoke.bwt,smoke=='Non-Smoker')$bwt,
            main = 'Histogram of Non-Smoker Birthweights', xlab='Birthweight',
            freq = FALSE)
```



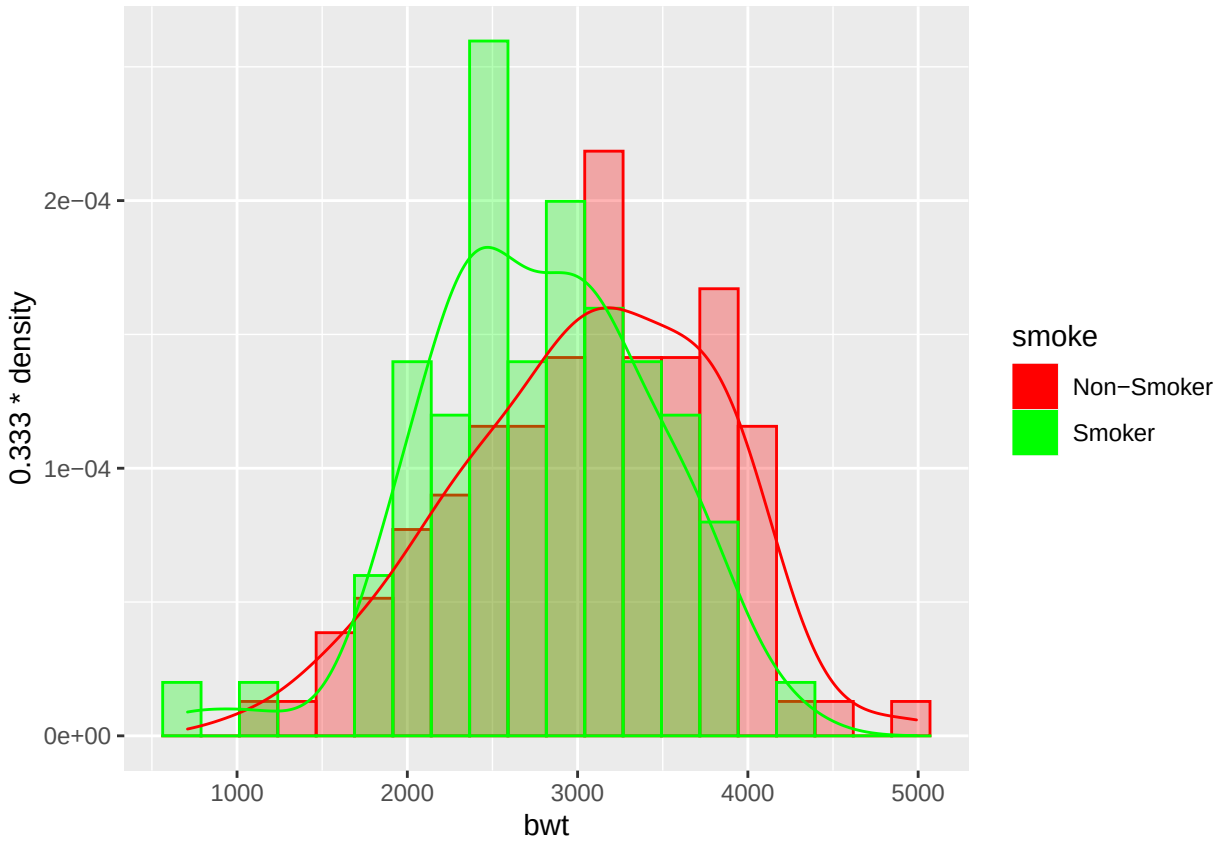

```
hg2 <- hist(filter(df.smoke.bwt,smoke=='Smoker')$bwt,
            main = 'Histogram of Smoker Birthweights', xlab='Birthweight',
            freq = FALSE)
```



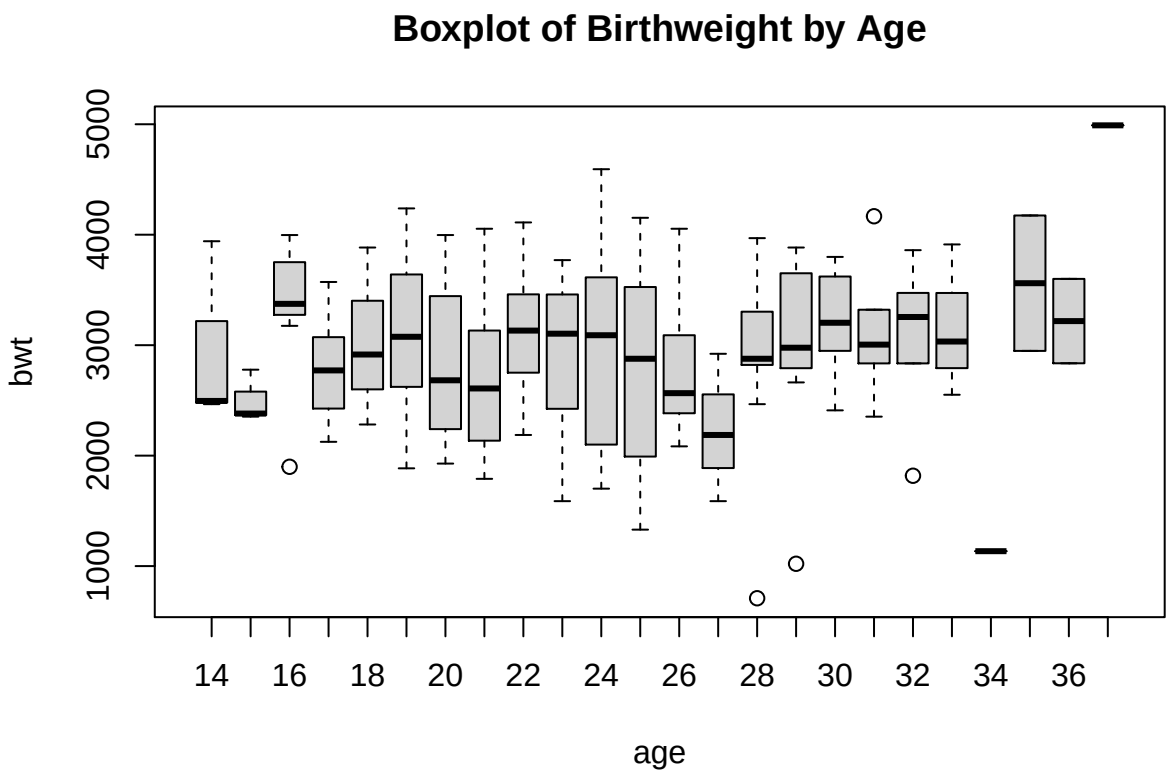
```r
c1 <- rgb(0, 0, 255, max = 255, alpha = 50, names = "blue")
c2 <- rgb(0, 225, 0, max = 255, alpha = 50, names = "green")

plot(hg1,col=c1, main='Birthweights by Smoker / Non-Smoker', xlab='Birthweight')
plot(hg2,col=c2,add = TRUE)
```



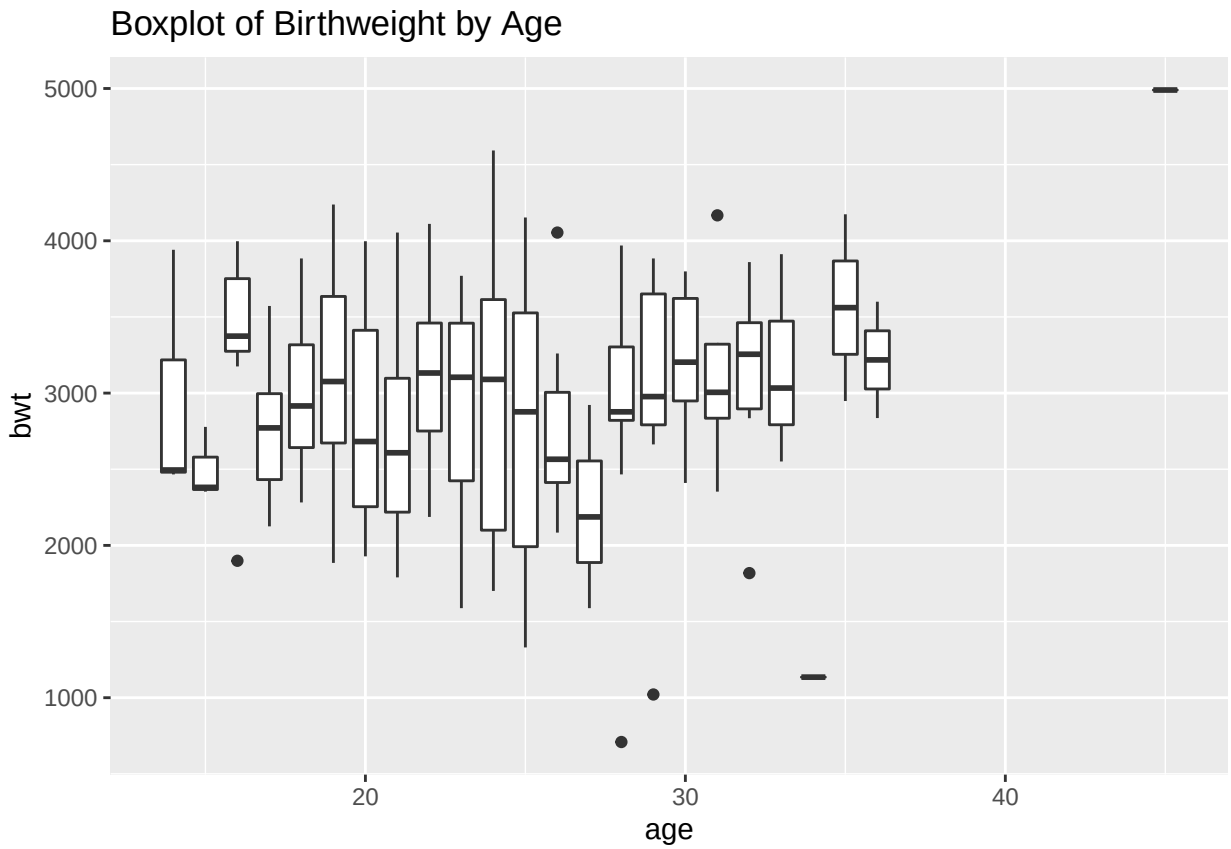

```r
# ggplot implementation
ggplot(data=df.smoke.bwt) +
  geom_histogram(aes(x=bwt,color=smoke, fill=smoke, y=0.333*..density..), alpha=0.3, position='identity
  scale_color_manual(values = rainbow(3)) +
  scale_fill_manual(values = rainbow(3)) +
  geom_density(aes(x=bwt,color=smoke, y=0.333*..density..))
```



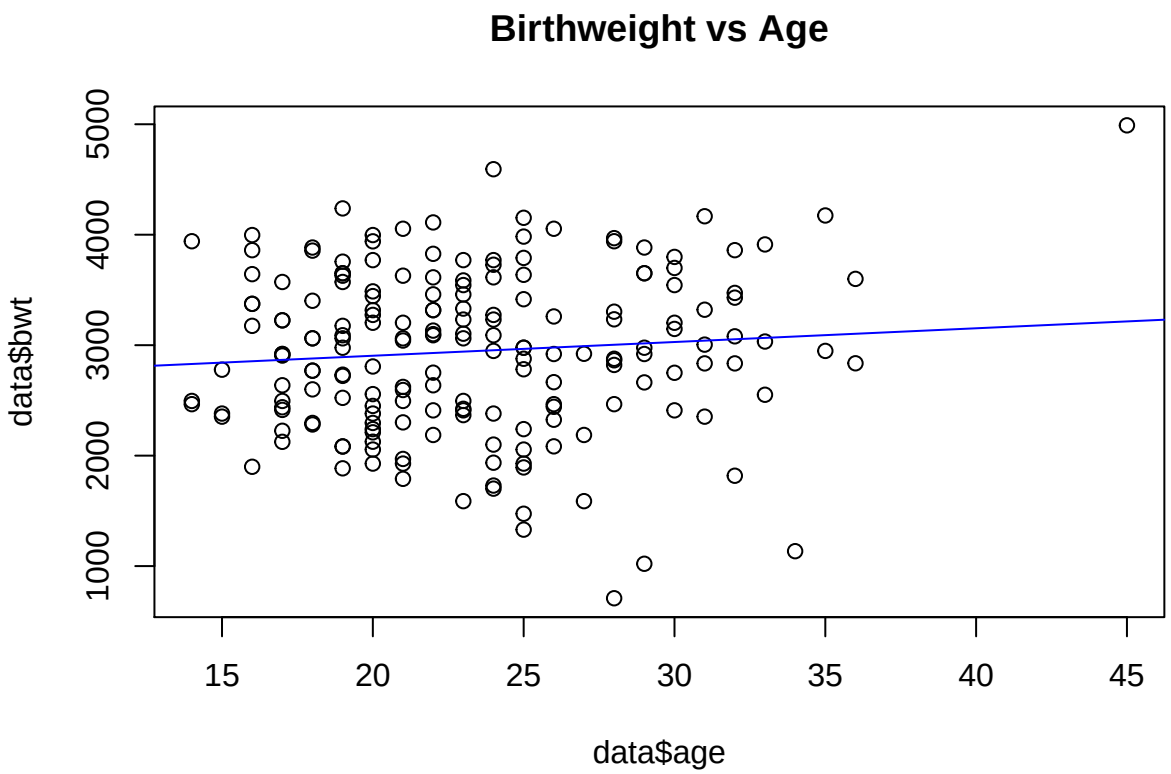

```
#plot the boxplot for "bwt" and "age".
df.age.bwt <- data[,c('age', 'bwt')]

# base graphics implementation
boxplot(data=df.age.bwt, bwt~age, main='Boxplot of Birthweight by Age')
```




```
#ggplot implementation
ggplot(data=data) + geom_boxplot(aes(y=bwt, group=age, x=age)) +
  ggtitle('Boxplot of Birthweight by Age')
```

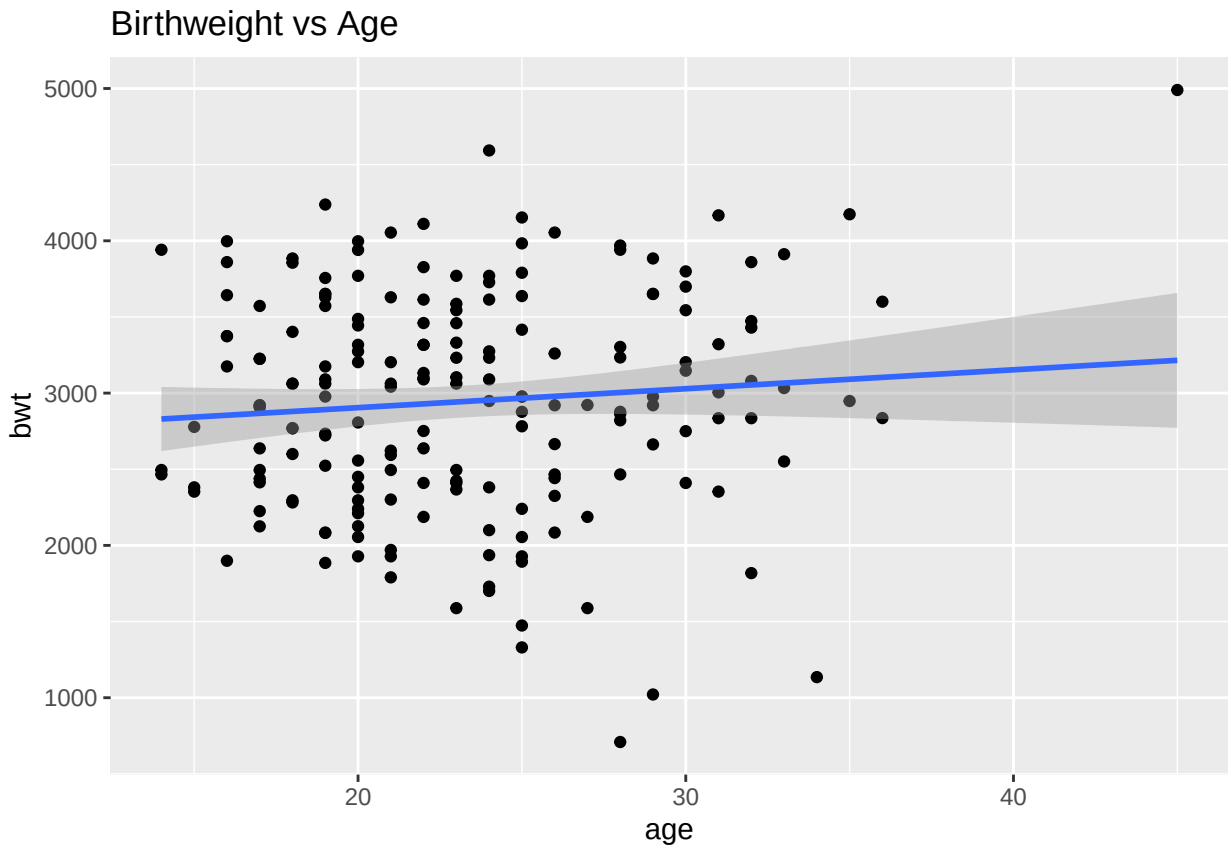



```
# plot the scatter plot for "bwt" and "age". What kind of relationship do you observe?

# base graphics
plot(data$age,data$bwt, main='Birthweight vs Age')
abline(lm(data$bwt ~data$age), col='Blue')
```



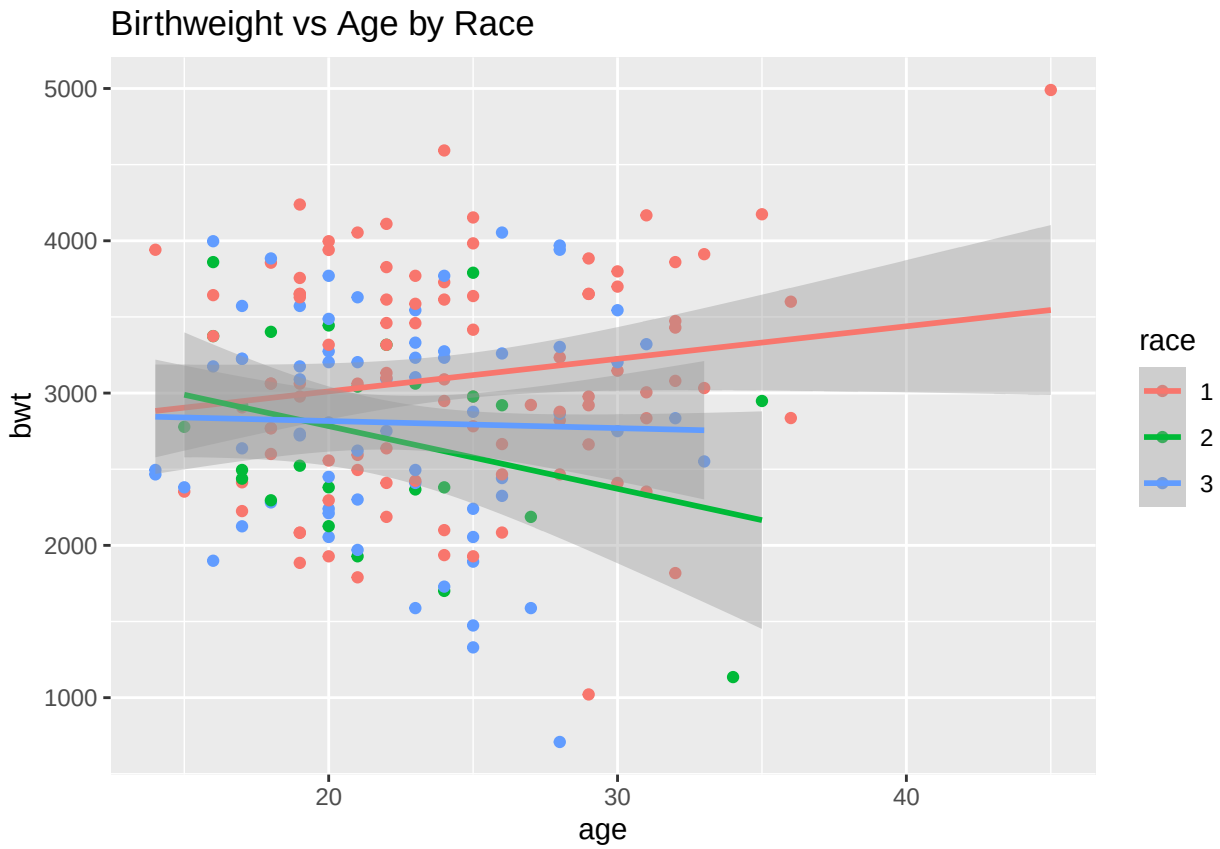

```
# ggplot implementation
ggplot(data=data,aes(y=bwt, x=age)) + geom_point( aes(y=bwt, x=age)) + stat_smooth(method=lm) +
  ggtitle('Birthweight vs Age')
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# compare the scatter plot for "bwt" and "age" for people of different race.
# ggplot
ggplot(data=data, aes(y=bwt, x=age, color=race)) + geom_point() + stat_smooth(method=lm) +
  ggtitle('Birthweight vs Age by Race')
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# base graphics
par(mfrow=c(1,3),oma = c(0, 0, 2, 0))
data1 <- filter(data, race=='1')
plot(data1$age, data1$bwt, col=data1$race,
     main='White') + abline(lm(data1$bwt~data1$age))
```

```
## integer(0)
```

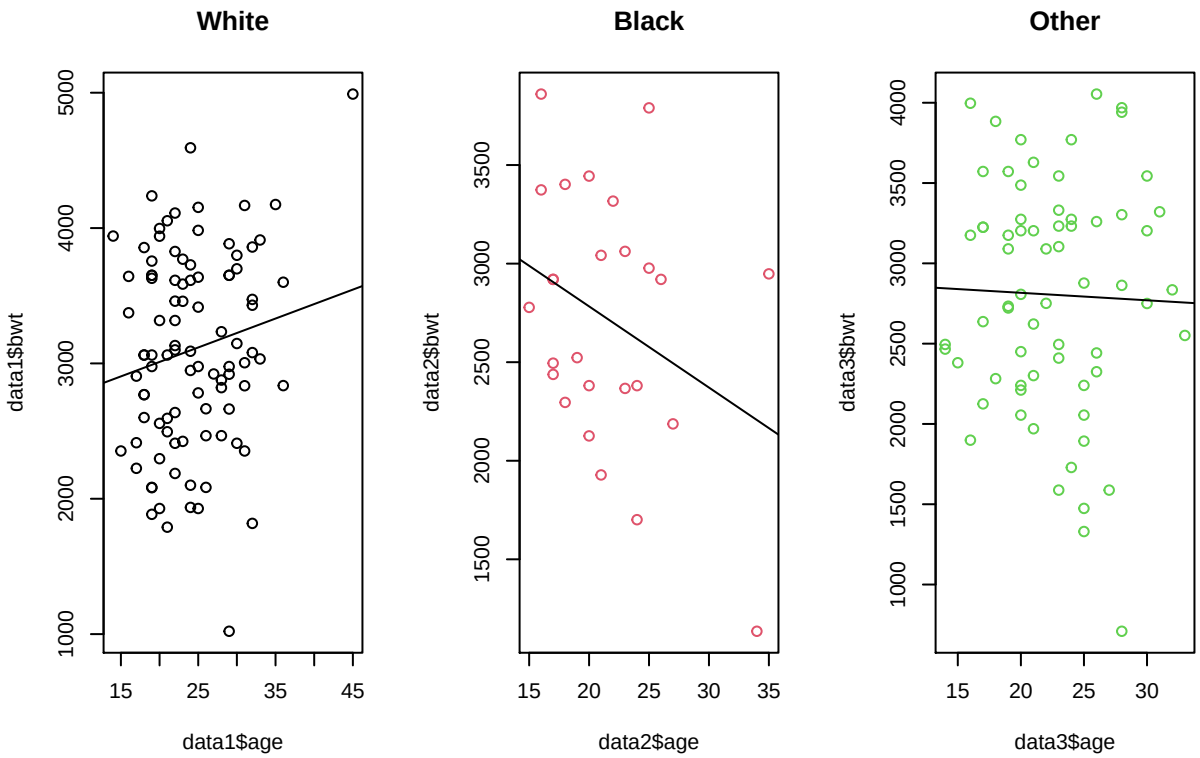
```
data2 <- filter(data, race=='2')
plot(data2$age, data2$bwt, col=data2$race,
     main='Black') + abline(lm(data2$bwt~data2$age))
```

```
## integer(0)
```

```
data3 <- filter(data, race=='3')
plot(data3$age, data3$bwt, col=data3$race,
     main='Other') + abline(lm(data3$bwt~data3$age))
```

```
## integer(0)
```

```
mtext('Birthweight vs Age by Race', outer = TRUE, cex = 1.5)
```




```
stu <- read.delim('StudentsPerformance.csv', header=TRUE, sep = ',')
dim(stu)
```

```
## [1] 1000    8
```

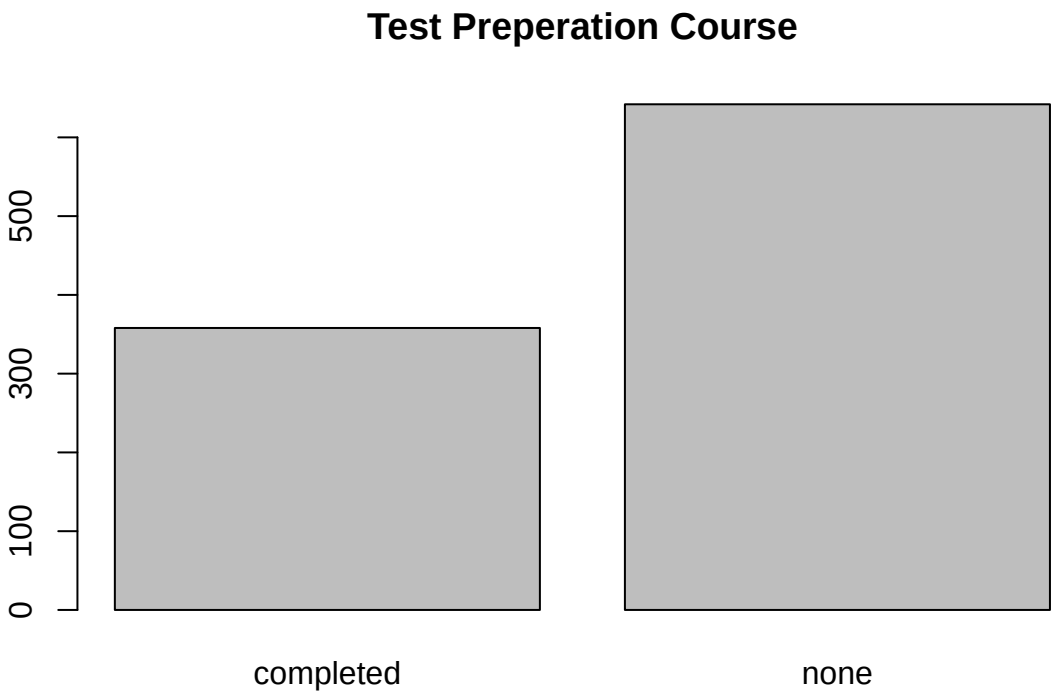
```
summary(stu)
```

```
##     gender           race.ethnicity     parental.level.of.education
##  Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##     lunch           test.preparation.course   math.score     reading.score
##  Length:1000        Length:1000             Min.   :  0.00   Min.   : 17.00
##  Class :character   Class :character        1st Qu.: 57.00   1st Qu.: 59.00
##  Mode  :character   Mode  :character        Median : 66.00   Median : 70.00
##                                             Mean   : 66.09   Mean   : 69.17
##                                             3rd Qu.: 77.00   3rd Qu.: 79.00
##                                             Max.   :100.00   Max.   :100.00
##  writing.score
##  Min.   : 10.00
##  1st Qu.: 57.75
##  Median : 69.00
##  Mean   : 68.05
##  3rd Qu.: 79.00
##  Max.   :100.00
```

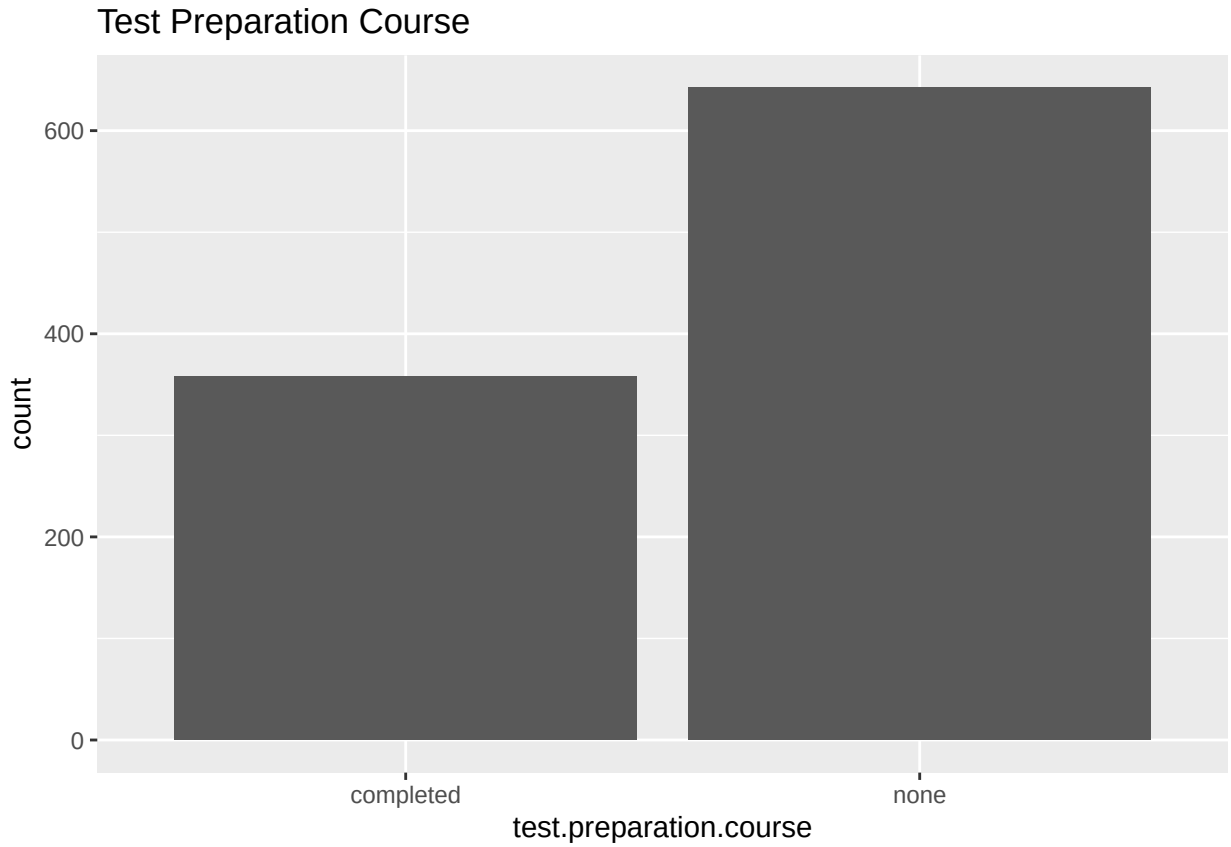
```
head(stu)
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group B           bachelor's degree     standard
## 2 female        group C                some college     standard
## 3 female        group B             master's degree     standard
## 4   male        group A          associate's degree free/reduced
## 5   male        group C                some college     standard
## 6 female        group B          associate's degree     standard
##   test.preparation.course math.score reading.score writing.score
## 1                    none         72            72            74
## 2               completed         69            90            88
## 3                    none         90            95            93
## 4                    none         47            57            44
## 5                    none         76            78            75
## 6                    none         71            83            78
```

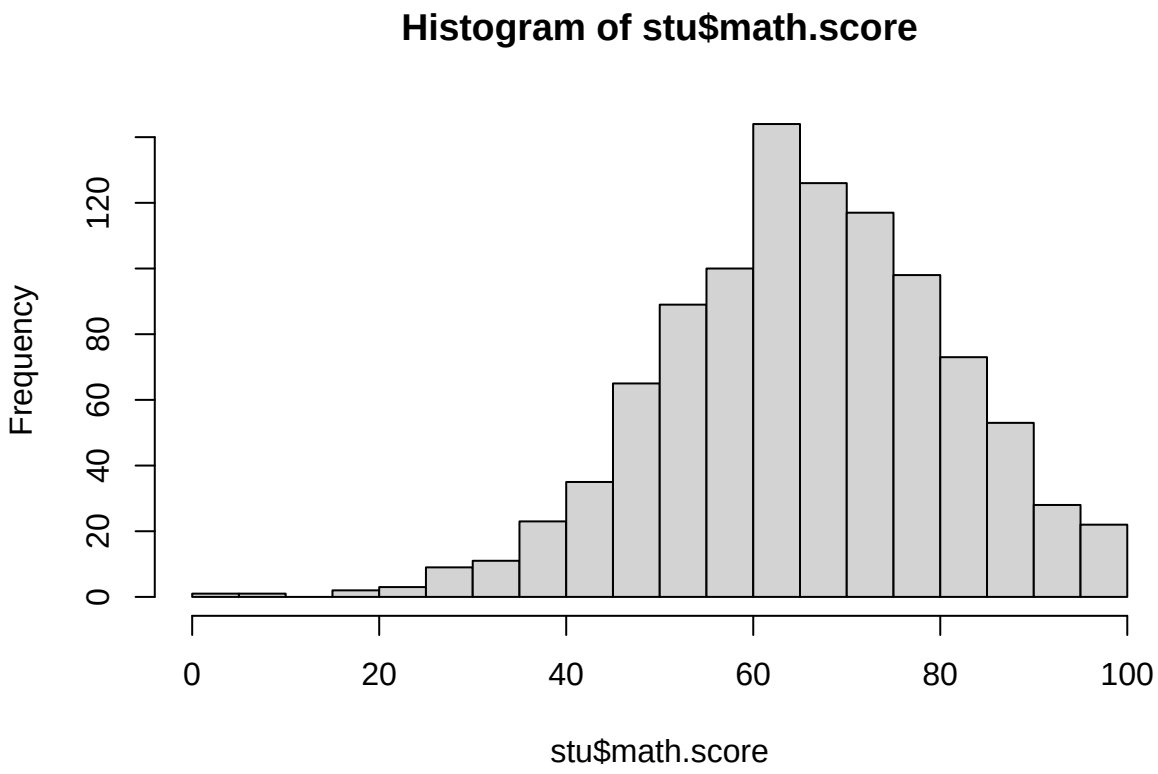
```
p <- ggplot(data=stu)

#Find one categorical variable and then use barplot to show the distribution of its values. If your dat

barplot(table(stu$test.preparation.course), main = "Test Preperation Course")
```




```
p + geom_bar(aes(x=test.preparation.course)) + ggtitle('Test Preparation Course')
```

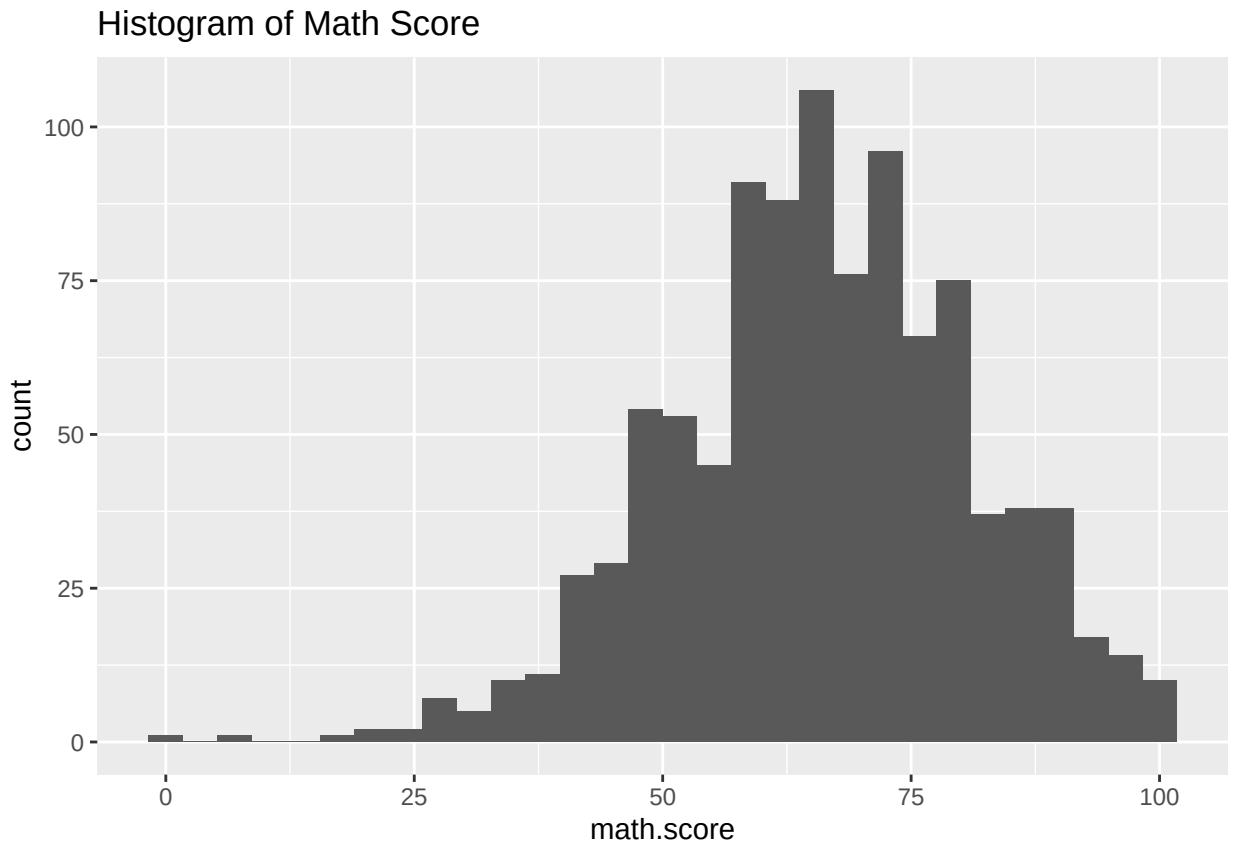



```
# Find one continuous variable and then use histogram and boxplot to show the distribution of its value

#base grahics
hist(stu$math.score, breaks=20)
```

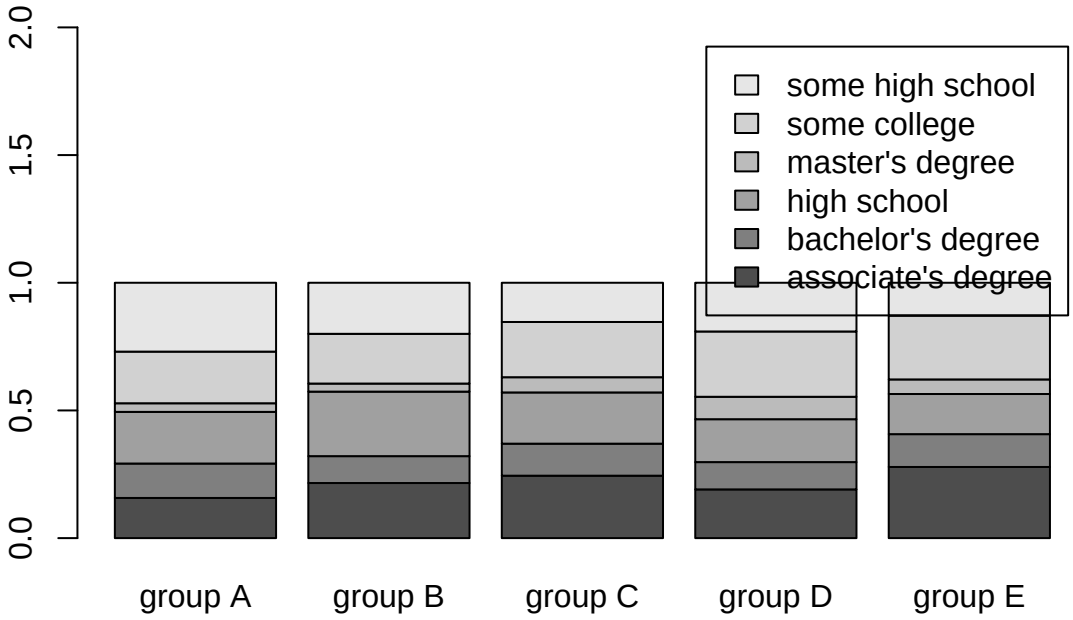



```
# ggplot
p + geom_histogram(aes(math.score)) + ggtitle('Histogram of Math Score')
```

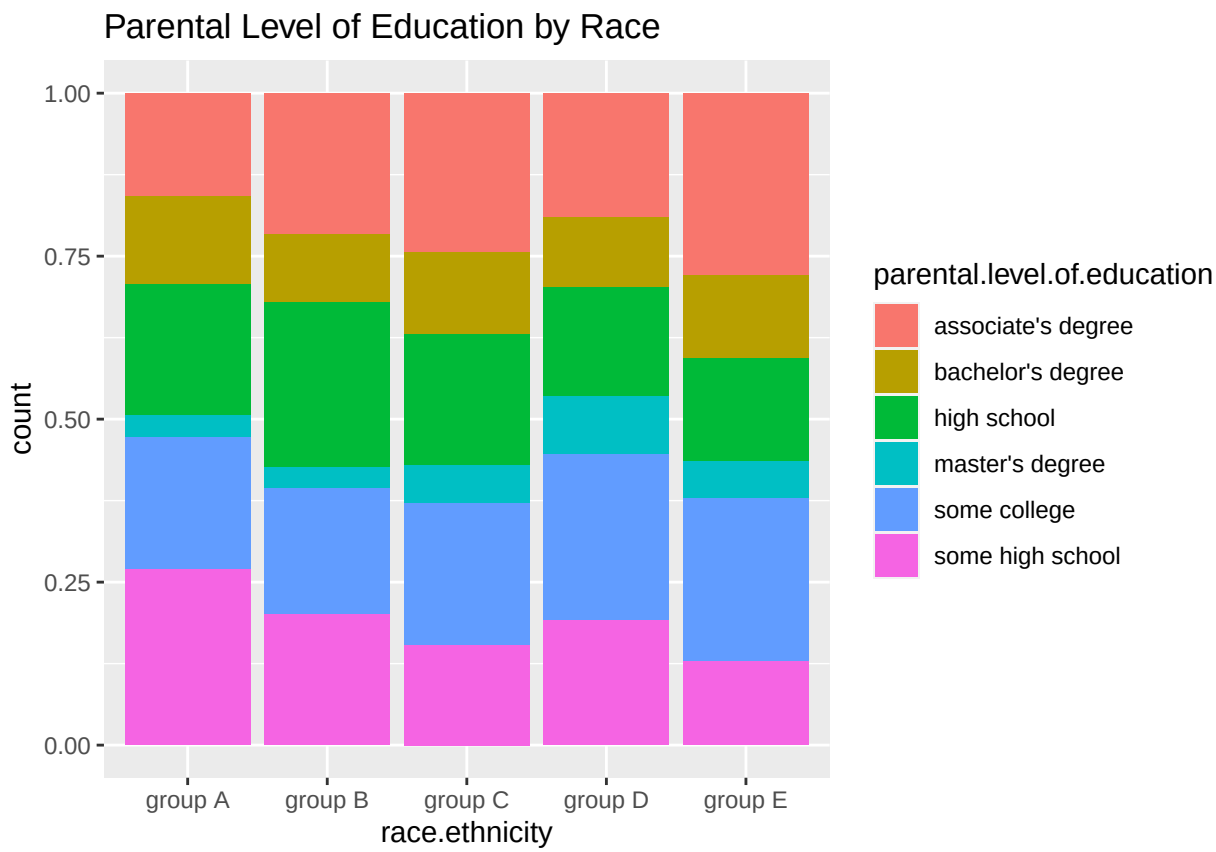
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



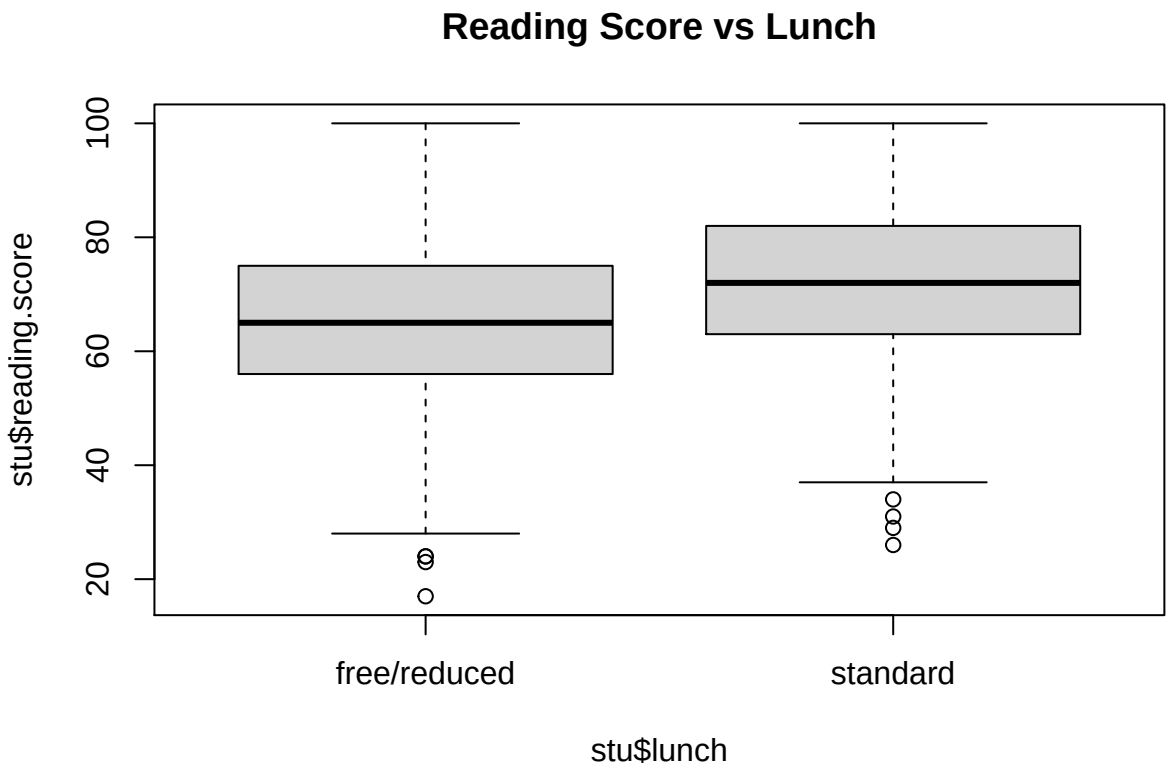

```
#Show the relationship between two categorical variables using tables and plots.

# base graphics
stu[,c('parental.level.of.education','race.ethnicity')] %>% table() %>% prop.table(2) %>%  barplot(leger
```




```
# ggplot
p + geom_bar(aes(x=race.ethnicity,fill=parental.level.of.education), position='fill') +
  ggtitle('Parental Level of Education by Race')
```




```
#Show the relationship between a categorical and a numerical variable using numerical comparisons (any
#interaction.plot(stu$test.preparation.course,stu$race.ethnicity, stu$math.score)

boxplot(stu$reading.score ~ stu$lunch, main='Reading Score vs Lunch')
```

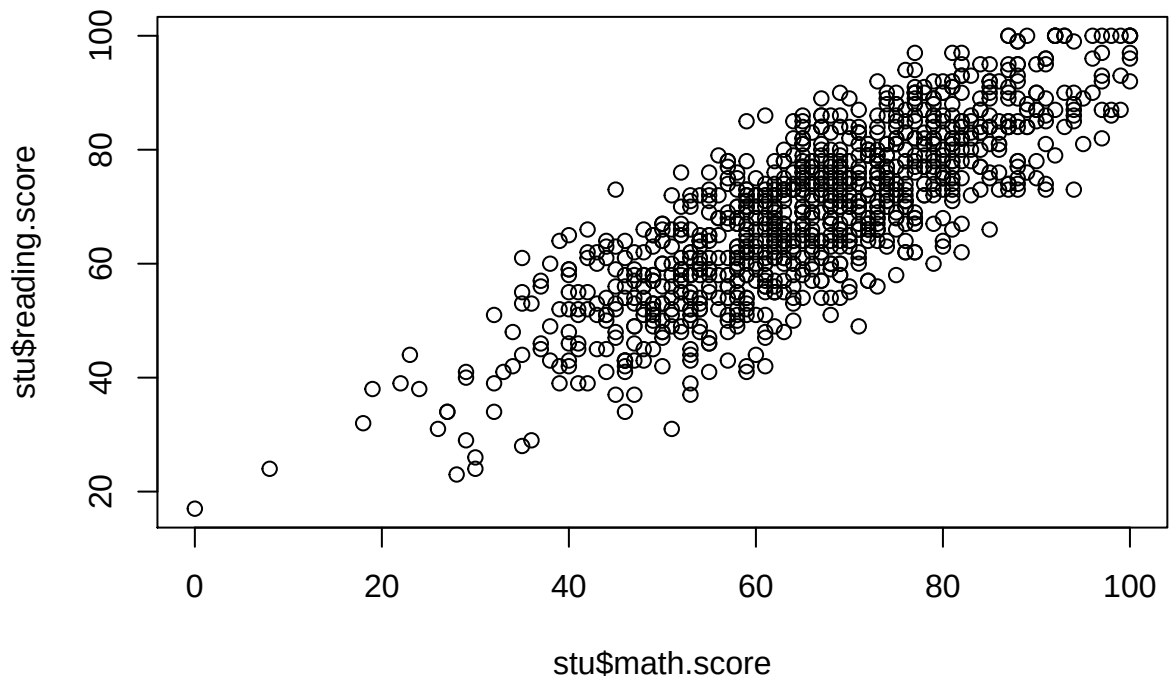



```
p + geom_boxplot(aes(x=lunch,y=reading.score)) + ggtitle('Reading Score vs Lunch')
```

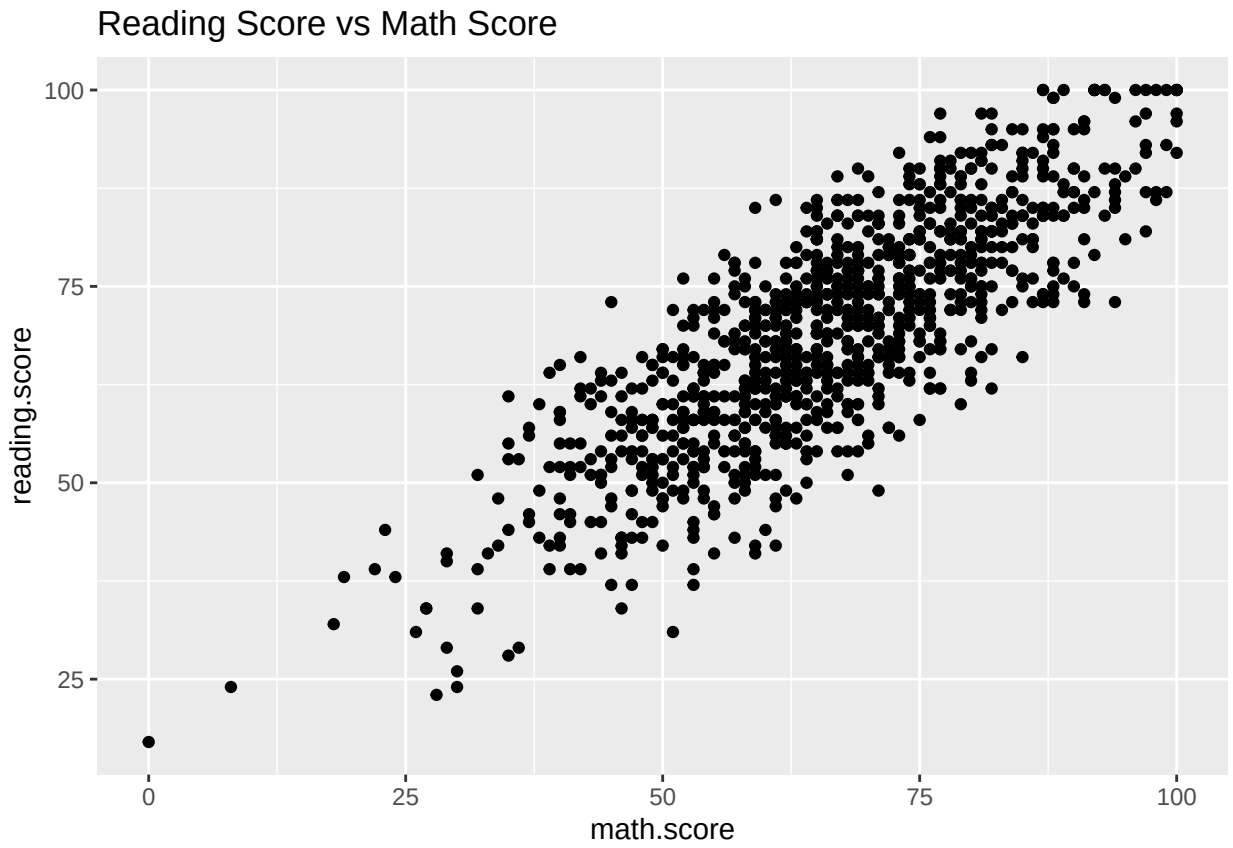
```r
#Show the relationship between two continuous variables useing using numerical comparisons (any compari
summary(lm(stu$math.score~ stu$reading.score))
```

```
##
## Call:
## lm(formula = stu$math.score ~ stu$reading.score)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.3419  -6.3419  -0.0221   6.2713  24.6581
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.35759    1.33818   5.498 4.87e-08 ***
## stu$reading.score  0.84910    0.01893  44.855  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.736 on 998 degrees of freedom
## Multiple R-squared:  0.6684, Adjusted R-squared:  0.6681
## F-statistic:  2012 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
# base graphics
plot(stu$math.score, stu$reading.score)
```




```r
# ggplot
p + geom_point(aes(x=math.score,y=reading.score)) + ggtitle('Reading Score vs Math Score')
```



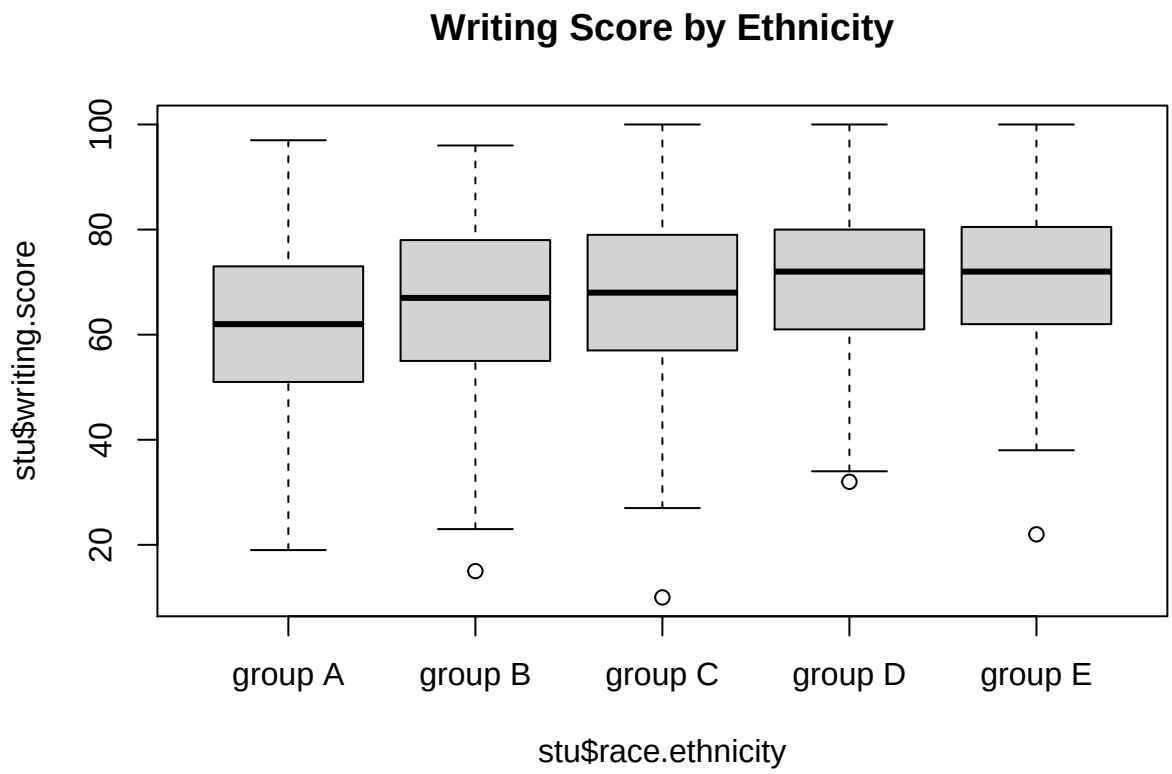

```r
#Identify a research question of your own about the dataset and try to answer it using simple statistic

# Is there differences in the writing score between races?

# statistical summary
aggregate(stu, list(stu$race.ethnicity), mean)[,c('Group.1','writing.score')]
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

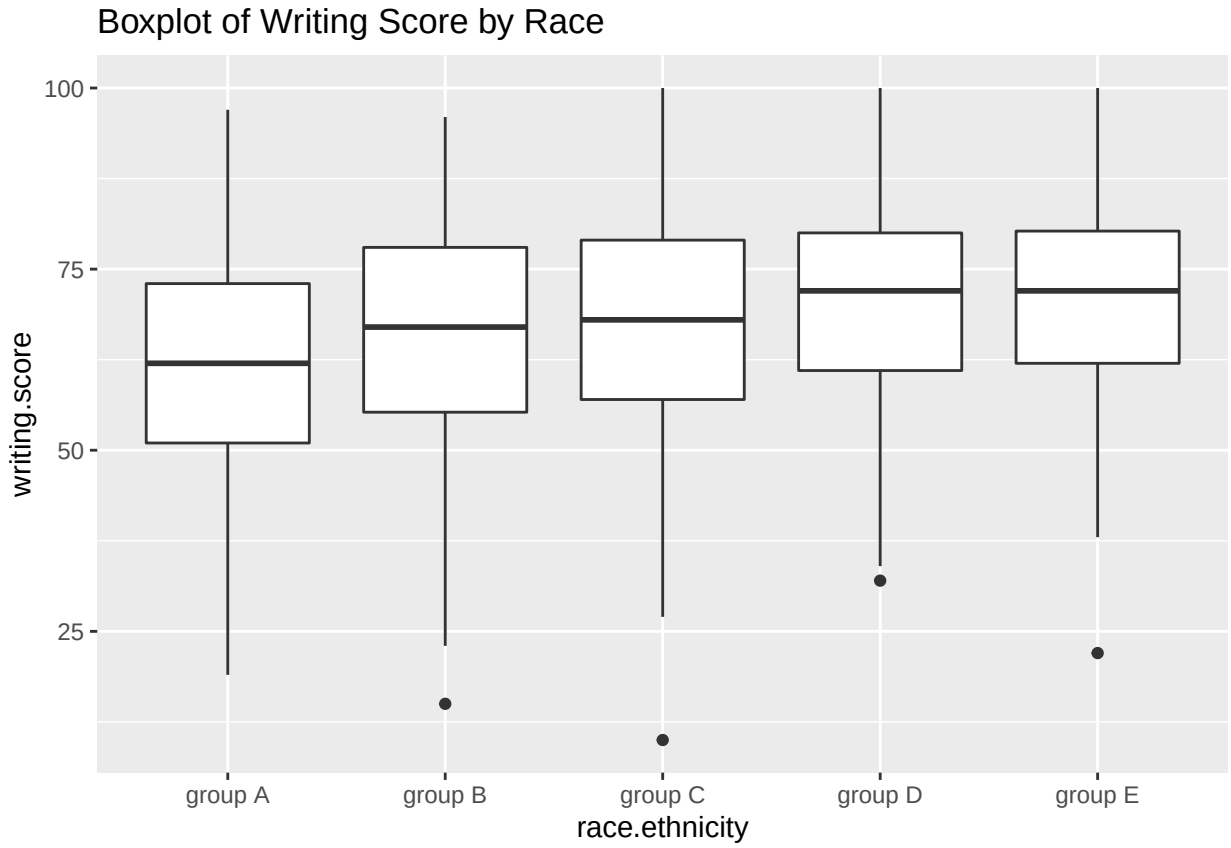
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
##   Group.1 writing.score
## 1 group A      62.67416
## 2 group B      65.60000
## 3 group C      67.82759
## 4 group D      70.14504
## 5 group E      71.40714
```

```r
# base graphics
boxplot(stu$writing.score ~ stu$race.ethnicity, main = 'Writing Score by Ethnicity')
```




```r
# ggplot2
p + geom_boxplot(aes(y=writing.score, x=race.ethnicity)) + ggtitle('Boxplot of Writing Score by Race')
```


Boxplot of Writing Score by Race
writing.score
100
75
50
25
group A
group B
group C
group D
group E
race.ethnicity