Sofia Guo HW 1 C142

Sofia Guo

1/24/2019

```
##
## 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
```

```
## Warning: package 'Hmisc' was built under R version 3.5.2
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:Hmisc':
##
##     subplot
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

1. Conduct a simulation of drawing sample size *n* from Bernoulli distribution with mean *p*.

```
#set seed for consistent results
set.seed(1234)

#conduct each simulation
case1 <- rbinom(1000, 30, 0.05)
case2 <- rbinom(1000, 30, 0.25)
case3 <- rbinom(1000, 60, 0.05)
case4 <- rbinom(1000, 60, 0.25)

#function to find means, standard deviations, and ci length

find_stats <- function(casenum, n, p) {
  means <- casenum/n
  s_d <- (means*(1-means))^(1/2)
  ci_low <- means-((1.96*s_d)/n^(1/2))
  ci_hi <- means+((1.96*s_d)/n^(1/2))
  ci_length <- ci_hi-ci_low
  in_ci <- ci_low <= p & ci_hi >= p
  return(data.frame("Ybar" = means, "StDev" = s_d, "CI low" = ci_low, "CI High" = ci_hi,
 "CI length" = ci_length, "In CI" = in_ci))
}

#apply to all cases
case1_out <- find_stats(case1, 30, 0.05)
case2_out <- find_stats(case2, 30, 0.25)
case3_out <- find_stats(case3, 60, 0.05)
case4_out <- find_stats(case4, 60, 0.25)

#display results
head(case1_out)
```

```
##          Ybar     StDev       CI.low   CI.High CI.length In.CI
## 1 0.00000000 0.0000000  0.000000000 0.0000000 0.0000000 FALSE
## 2 0.06666667 0.2494438 -0.022595660 0.1559290 0.1785247  TRUE
## 3 0.06666667 0.2494438 -0.022595660 0.1559290 0.1785247  TRUE
## 4 0.06666667 0.2494438 -0.022595660 0.1559290 0.1785247  TRUE
## 5 0.10000000 0.3000000 -0.007353621 0.2073536 0.2147072  TRUE
## 6 0.06666667 0.2494438 -0.022595660 0.1559290 0.1785247  TRUE
```

```
head(case2_out)
```

```
##         Ybar     StDev     CI.low   CI.High CI.length In.CI
## 1 0.3333333 0.4714045 0.16464339 0.5020233 0.3373799  TRUE
## 2 0.2333333 0.4229526 0.08198169 0.3846850 0.3027033  TRUE
## 3 0.1666667 0.3726780 0.03330556 0.3000278 0.2667222  TRUE
## 4 0.2333333 0.4229526 0.08198169 0.3846850 0.3027033  TRUE
## 5 0.3000000 0.4582576 0.13601463 0.4639854 0.3279707  TRUE
## 6 0.2333333 0.4229526 0.08198169 0.3846850 0.3027033  TRUE
```

```
head(case3_out)
```

```
##          Ybar     StDev       CI.low    CI.High  CI.length In.CI
## 1 0.01666667 0.1280191 -0.015726634 0.04905997 0.06478660 FALSE
## 2 0.03333333 0.1795055 -0.012087823 0.07875449 0.09084231  TRUE
## 3 0.05000000 0.2179449 -0.005147681 0.10514768 0.11029536  TRUE
## 4 0.05000000 0.2179449 -0.005147681 0.10514768 0.11029536  TRUE
## 5 0.01666667 0.1280191 -0.015726634 0.04905997 0.06478660 FALSE
## 6 0.06666667 0.2494438  0.003548670 0.12978466 0.12623599  TRUE
```

```
head(case4_out)
```

```
##         Ybar     StDev     CI.low   CI.High CI.length In.CI
## 1 0.3500000 0.4769696 0.22931004 0.4706900 0.2413799  TRUE
## 2 0.2000000 0.4000000 0.09878604 0.3012140 0.2024279  TRUE
## 3 0.1833333 0.3869396 0.08542412 0.2812426 0.1958184  TRUE
## 4 0.2000000 0.4000000 0.09878604 0.3012140 0.2024279  TRUE
## 5 0.2166667 0.4119736 0.11242297 0.3209104 0.2084874  TRUE
## 6 0.2000000 0.4000000 0.09878604 0.3012140 0.2024279  TRUE
```

a) Report the mean estimate of p (the true mean) for each case

For case 1 where (n,p) is (30, 0.05):

```
  mean(case1_out$Ybar)
```

```
## [1] 0.05176667
```

For case 2 where (n,p) is (30, 0.25):

```
mean(case2_out$Ybar)
```

```
## [1] 0.2460333
```

For case 3 where (n,p) is (60, 0.05):

```
mean(case3_out$Ybar)
```

```
## [1] 0.05123333
```

For case 4 where (n,p) is (60, 0.25):

```
mean(case4_out$Ybar)
```

```
## [1] 0.2493667
```

b) Report the mean estimate of s (the true standard deviation) for each case

For case 1 where (n,p) is (30, 0.05):

```
mean(case1_out$StD)
```

```
## [1] 0.1886503
```

For case 2 where (n,p) is (30, 0.25):

```
mean(case2_out$StD)
```

```
## [1] 0.4214603
```

For case 3 where (n,p) is (60, 0.05):

```
mean(case3_out$StD)
```

```
## [1] 0.2080865
```

For case 4 where (n,p) is (60, 0.25):

```
mean(case4_out$StD)
```

```
## [1] 0.4276934
```

c) Report the fraction of the time that the CI contains the true p (coverage rate) for each case

For case 1 where (n,p) is (30, 0.05):

```
sum(case1_out$In.CI==T) /1000
```

```
## [1] 0.791
```

For case 2 where (n,p) is (30, 0.25):

```
sum(case2_out$In.CI==T) /1000
```

```
## [1] 0.947
```

For case 3 where (n,p) is (60, 0.05):

```
sum(case3_out$In.CI==T) / 1000
```

```
## [1] 0.819
```

For case 4 where (n,p) is (60, 0.25):

```
sum(case4_out$In.CI==T) / 1000
```

```
## [1] 0.931
```

Do you agree that n or 30 or larger is enough to ensure that asymptotic confidence intervals work well?

According to the results to part b), it seems like the fraction of time that the CI for all cases contains the true p is pretty high (low at 79%, high at 95% of the time). Depending on how accurate one desires their estimates to be, if ~80% accuracy is "well" and accurate, then one may say that n of 30 or more is enough. However, if one is shooting for near 100% accuracy, the answer would be that n should be at least more than 60 (which yields only a low 90's percent accuracy) to ensure that asymptotic confidence intervals are working "well".

2. Working with mother's weight dataset

a) Develop the following graphs:

i) Find the deciles of the mother’s weight, then put each mother in a decile, and get the mean birthweight of babies. Plot the mean baby weight against mean mother birthweight in decile (binscatter plot):

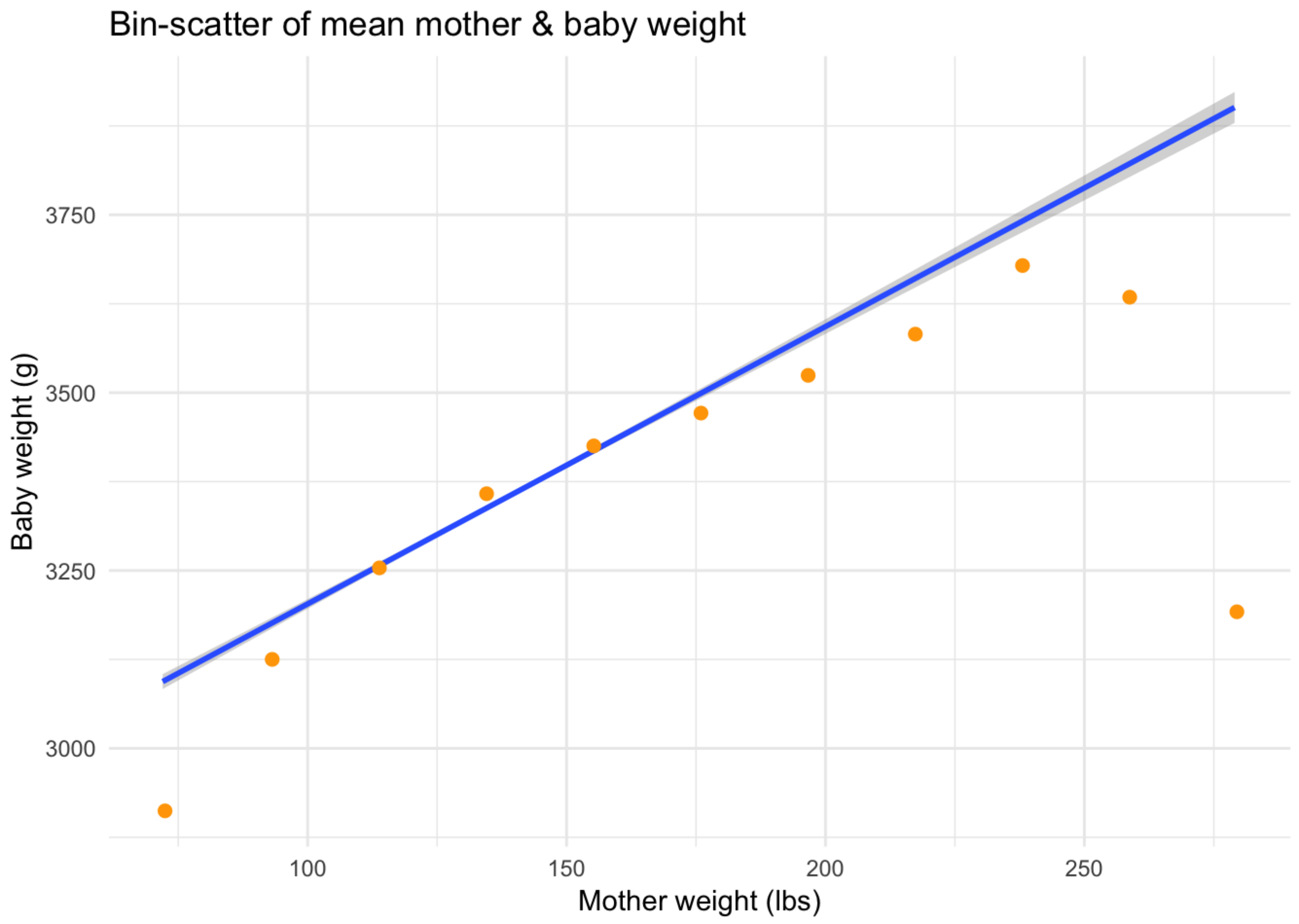
```
#load dataset in
mothers <- read.csv("/Users/sofia/Box/Cal (sofiaguo@berkeley.edu)/2018-19/Spring 2019/Ec
on 142/PSETS/ps1.csv")

#find deciles of mother's weight
momw_dec_array <- quantile(mothers$momweight, prob = seq(0, 1, length = 11), type = 5)

momw_dec_array
```

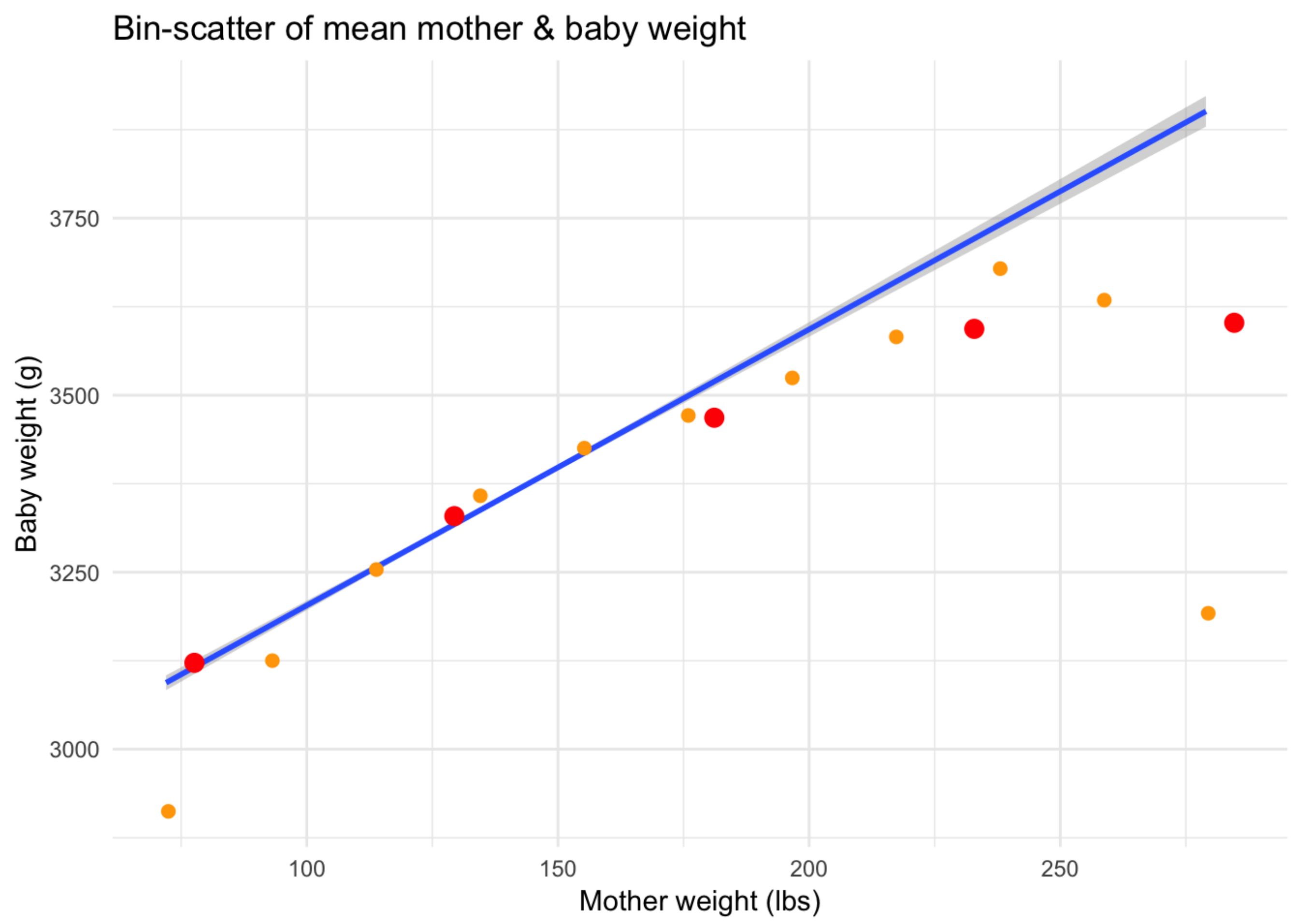
```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   72  108  115  121  128  133  140  147  158  172  279
```

```
#construct the binscatter
ggplot(mothers, aes(momweight, babyweight)) +
    geom_smooth(method='lm')+
  labs(y = 'Baby weight (g)', x = 'Mother weight (lbs)',
  title = 'Bin-scatter of mean mother & baby weight') +
  stat_summary_bin(fun.y='mean', bins=10,
                   color='orange', size=2, geom='point')+
  theme_minimal()
```



ii) In addition to the mean baby's weight, construct the 10, 25, 50, 75 and 90 percentiles and plot against mean mother's weight in each decile. Any interesting patterns?

```
#plot the new data
ggplot(mothers, aes(momweight, babyweight)) +
  geom_smooth(method='lm')+
  labs(y = 'Baby weight (g)', x = 'Mother weight (lbs)',
  title = 'Bin-scatter of mean mother & baby weight') +
    stat_summary_bin(fun.y='mean', bins=10,
                   color='orange', size=2, geom='point') +
    stat_summary_bin(fun.y='mean', bins=4,
                   color='red', size=3, geom='point')+
  theme_minimal()
```

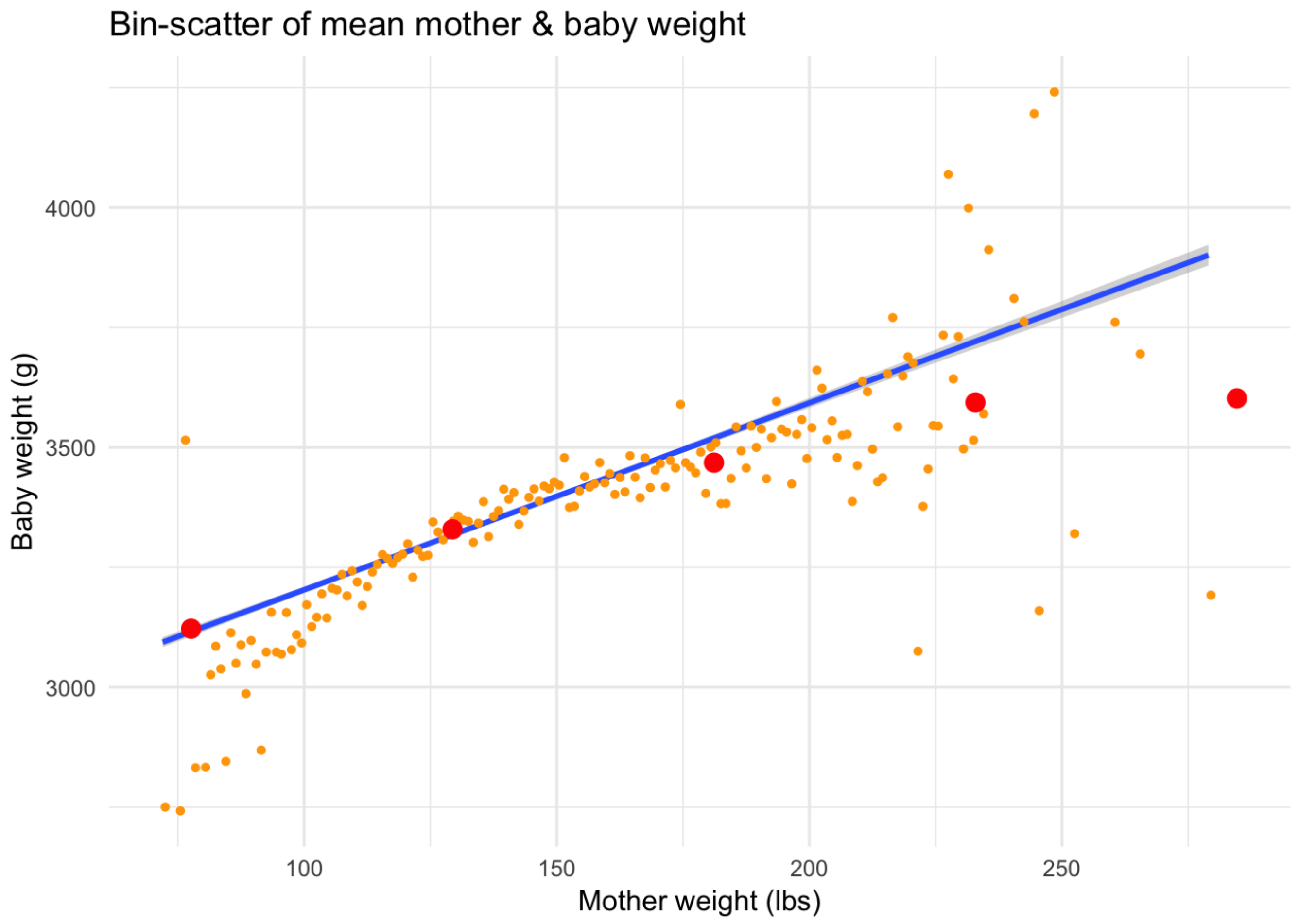


There is a pattern appearing where the larger bins show a smoother, flatter relationship (the blue line indicating the best fit for the original data). The red dots indicating the quintiles show a logistic-type relationship, where baby weight increases at a slower rate as mothers are heavier. You can see a similar pattern but less clear in the decile scatter with orange dots. This might suggest that the relationship between infant birth weight and mother weight before pregnancy is logistic with decreasing returns to scale.

iii) Redo i) using individual 1-lb values of mother's weight and compare to binscatter plot.

```
#construct just a scatter using 1-lb increments for mother's weight

ggplot(mothers, aes(momweight, babyweight)) +
    geom_smooth(method='lm')+
  labs(y = 'Baby weight (g)', x = 'Mother weight (lbs)',
  title = 'Bin-scatter of mean mother & baby weight') +
  stat_summary_bin(fun.y='mean', bins=max(mothers$momweight)-min(mothers$momweight),
                   color='orange', size=1, geom='point')+
  stat_summary_bin(fun.y='mean', bins=4,
                   color='red', size=3, geom='point')+
  theme_minimal()
```



Compared to the decile plot, this is much more dispersed especially towards the higher end of mother weight. Compared to the quintile representation in red, although the 1-lb bins begin as a logistic-looking relationship, it disperses when mother weight reaches above 200 lbs, suggesting that the relationship may not be as simple as a logistic one.

b) Divide mothers into 5-pound buckets, starting at 95 pounds. Calculate baby weights separately for 4 groups of mother's within each bucket.

```
#filter mothers weight starting at 95
momw_95 <- mothers %>%
  dplyr::filter(momweight >= 95) %>%
  arrange(momweight)

#find number of bins needed
(max(momw_95$momweight) - 95)/5
```

```
## [1] 36.8
```

```
#rounding up 36.8 to 37 total bins of 5 pounds each, add a categorization to the data fr
ame
momw_95_bin <- momw_95 %>%
  mutate(mombuckets = as.numeric(cut(momw_95$momweight, 37)))

#calculate baby weights for the 4 groups specified

momh_60 <- momw_95_bin %>%
  dplyr::filter(momheight<= 60) %>%
  group_by(mombuckets) %>%
  dplyr::summarize(meanbw = mean(babyweight))

head(momh_60)
```

```
## # A tibble: 6 x 2
##   mombuckets meanbw
##        <dbl>  <dbl>
## 1          1  3074.
## 2          2  3123.
## 3          3  3174.
## 4          4  3184.
## 5          5  3190.
## 6          6  3226.
```

```
momh_61_63 <- momw_95_bin %>%
  dplyr::filter(momheight > 61 & momheight < 63) %>%
  group_by(mombuckets) %>%
  dplyr::summarize(meanbw = mean(babyweight))

head(momh_61_63)
```

```
## # A tibble: 6 x 2
##   mombuckets meanbw
##        <dbl>  <dbl>
## 1          1  3109.
## 2          2  3192.
## 3          3  3187.
## 4          4  3214.
## 5          5  3255.
## 6          6  3276.
```

```
momh_64_66 <- momw_95_bin %>%
  dplyr::filter(momheight > 64 & momheight < 66) %>%
  group_by(mombuckets) %>%
  dplyr::summarize(meanbw = mean(babyweight))

head(momh_64_66)
```

```
## # A tibble: 6 x 2
##   mombuckets meanbw
##        <dbl>  <dbl>
## 1          1  3224.
## 2          2  3228.
## 3          3  3256.
## 4          4  3267.
## 5          5  3303.
## 6          6  3344.
```

```
momh_67 <- momw_95_bin %>%
  dplyr::filter(momheight >=67) %>%
  group_by(mombuckets) %>%
  dplyr::summarize(meanbw = mean(babyweight))

head(momh_67)
```

```
## # A tibble: 6 x 2
##   mombuckets meanbw
##        <dbl>  <dbl>
## 1          1  3028.
## 2          2  3300.
## 3          3  3285.
## 4          4  3314.
## 5          5  3386.
## 6          6  3336.
```

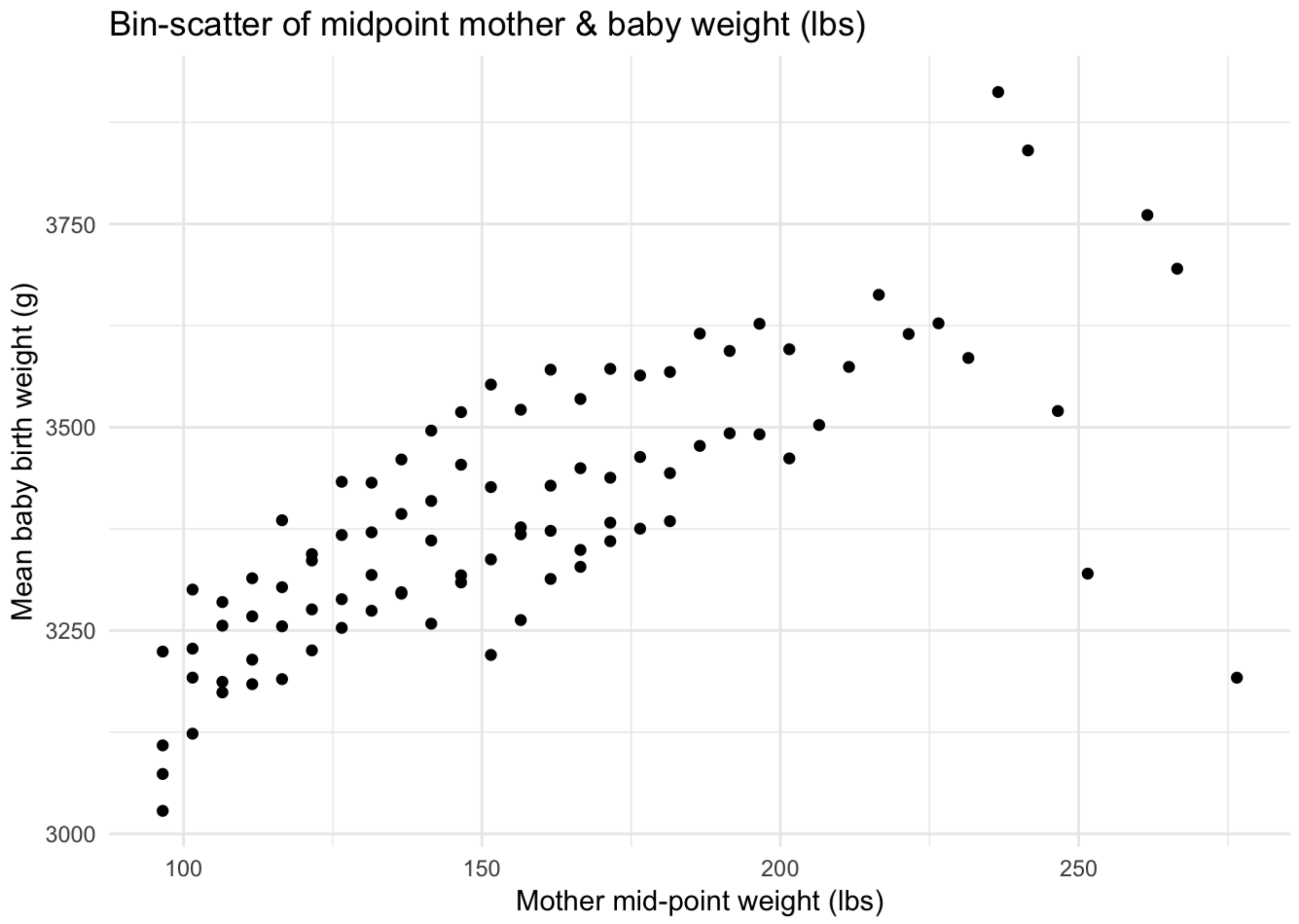
```
#plot mean baby weights against the median weight in the bucket

midpoints <- seq(96.5, 276.5, 5)
midpoints_df <- data.frame(mombuckets = 1:37,
           midheights = midpoints)

momh_master <- rbind(momh_60, momh_61_63, momh_64_66, momh_67)

momh_midpoints <- merge(momh_master, midpoints_df, by = 'mombuckets')

#plot the new data
ggplot(momh_midpoints, aes(midheights, meanbw)) +
  geom_point() +
  labs(x = 'Mother mid-point weight (lbs)', y = 'Mean baby birth weight (g)',
  title = 'Bin-scatter of midpoint mother & baby weight (lbs)') +
  theme_minimal()
```



Looking at each mid-point weight for mothers, the four corresponding data points represent the mean baby weight for the four height cateogories. Given that there does not seem to be a consistent or clear separation between the four points as mother weight changes (i.e. you do not see the patttern of 4 distinct ‘lines’ being formed for each height category), this graph suggests that mother height does not have a consistent or clear effect on birthweight. However, the log-like relationship between mother weight and birthweight persists in this graph, further supporting the idea that while heavier mothers produce heavier babies, each additional pound a mother weighs has less effect on increasing birthweight than when mothers are below 200 lbs.

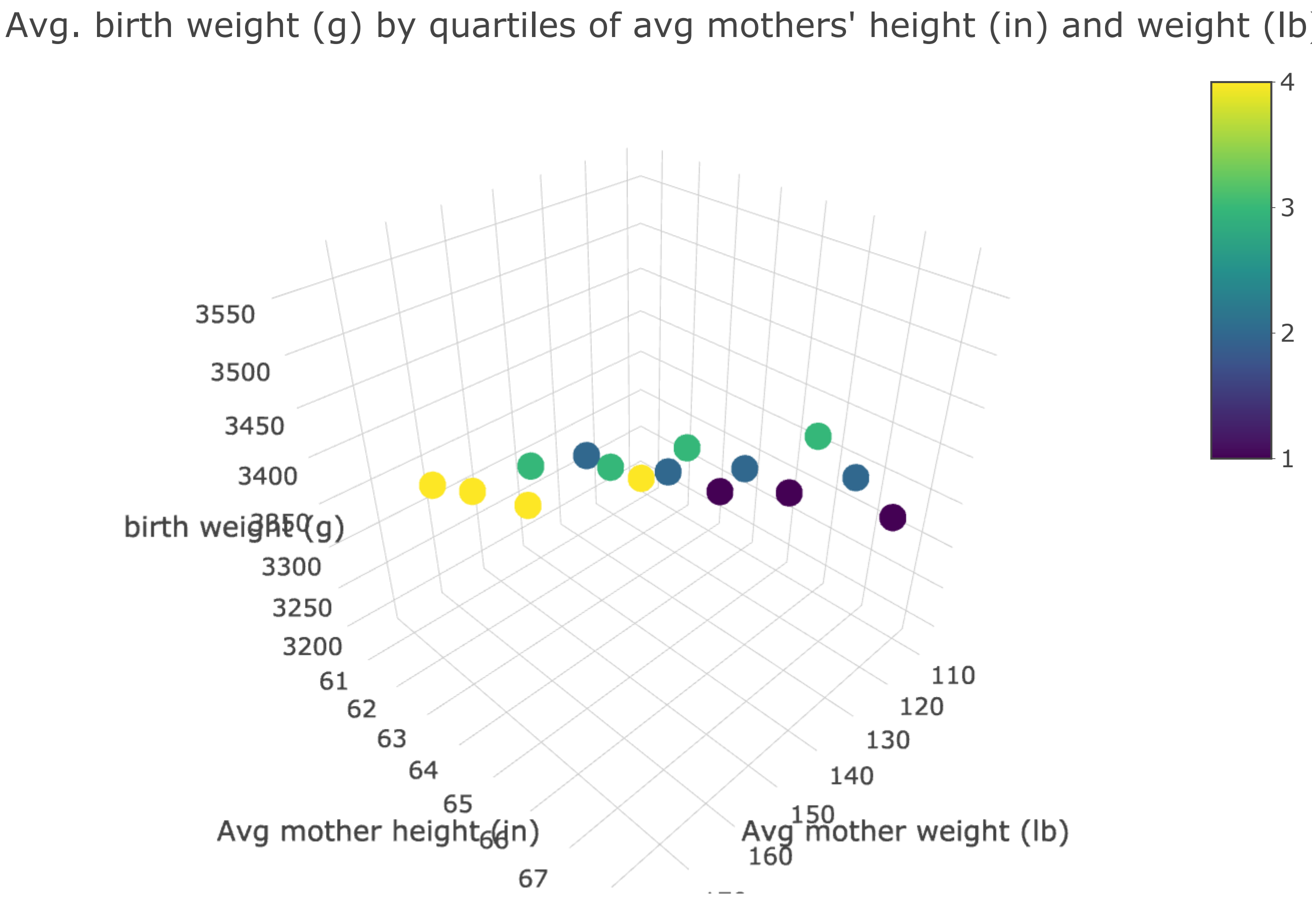
c) Construct a 3D binscatter

```
#construct 5 binsc

momw_df <- mutate(mothers, momw_quartile = as.integer(cut(momweight, quantile(momweight,
 probs=0:4/4), include.lowest=TRUE)),
                  momh_quartile = as.integer(cut(momheight, quantile(momheight, probs=0:
4/4), include.lowest=TRUE)))

momw_df_final <- momw_df %>%
  group_by(momw_quartile, momh_quartile) %>%
  dplyr::summarize(meanbw = mean(babyweight),
                   momh = mean(momheight),
                   momw = mean(momweight))

plot_ly(x=momw_df_final$momw, y=momw_df_final$momh, z=momw_df_final$meanbw, type="scatte
r3d", mode="markers", color=momw_df_final$momw_quartile) %>%
  layout(
    title = "Avg. birth weight (g) by quartiles of avg mothers' height (in) and weight
 (lb)",
    scene = list(
      xaxis = list(title = "Avg mother weight (lb)"),
      yaxis = list(title = "Avg mother height (in)"),
      zaxis = list(title = "Avg birth weight (g)")
    ))
```


Avg. birth weight (g) by quartiles of avg mothers' height (in) and weight (lb)
birth weight (g)
Avg mother height (in)
Avg mother weight (lb)