Checking the assumption of normality

Sophie Marion de Proce

2022-12-07

Introduction

In this self-guided lab, you will practice checking the assumption of normality in R using graphs and hypothesis tests. We will be using a significance level of 5% throughout.

We'll be using functions from the tidyverse collection of packages, so let's load it in.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.3      v forcats 0.5.1

## Warning: package 'tibble' was built under R version 4.2.1

## Warning: package 'readr' was built under R version 4.2.1

## Warning: package 'dplyr' was built under R version 4.2.1

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

The messages above are telling us which functions from the base R are overwritten by function from the Tidyverse, and that some packages were set up using a previous version of R.

Exercise 1

Using the NHANES dataset, imagine you want to investigate the difference in average diastolic blood pressure (BPDiaAve) between adult males and females. The exercise here is to check whether the normality assumptionholds for each subgroup, using both graphs and hypothesis tests.

We will be looking at the NHANES dataset, a survey dataset collected by the US National Center for Health Statistics (NCHS) which has conducted a series of health and nutrition surveys since the early 1960's. Since1999, approximately 5,000 individuals of all ages are interviewed intheir homes every year

and complete the health examination component of the survey. Find out more here: https://cran.rstudio.com/web/packages/NHANES/index.html, and check the reference manual for the R package: https://cran.rstudio.com/web/packages/NHANES/NHANES.pdf

Let's start by installing the NHANES package and loading it into R.

```
# install.packages("NHANES") # remove the hashtag at the start to run if needed
library(NHANES)
# Now let's tell R that we'll be using the NHANES data
data("NHANES")
```

And let's have a quick look at the dataset

```
str(NHANES)
```

```
## tibble [10,000 x 76] (S3: tbl_df/tbl/data.frame)
##  $ ID              : int [1:10000] 51624 51624 51624 51625 51630 51638 51646 51647 51647 51647 ...
##  $ SurveyYr        : Factor w/ 2 levels "2009_10","2011_12": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender          : Factor w/ 2 levels "female","male": 2 2 2 2 1 2 2 1 1 1 ...
##  $ Age             : int [1:10000] 34 34 34 4 49 9 8 45 45 45 ...
##  $ AgeDecade       : Factor w/ 8 levels " 0-9"," 10-19",..: 4 4 4 1 5 1 1 5 5 5 ...
##  $ AgeMonths       : int [1:10000] 409 409 409 49 596 115 101 541 541 541 ...
##  $ Race1           : Factor w/ 5 levels "Black","Hispanic",..: 4 4 4 5 4 4 4 4 4 4 ...
##  $ Race3           : Factor w/ 6 levels "Asian","Black",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ Education       : Factor w/ 5 levels "8th Grade","9 - 11th Grade",..: 3 3 3 NA 4 NA NA 5 5 5 ...
##  $ MaritalStatus   : Factor w/ 6 levels "Divorced","LivePartner",..: 3 3 3 NA 2 NA NA 3 3 3 ...
##  $ HHIncome        : Factor w/ 12 levels " 0-4999"," 5000-9999",..: 6 6 6 5 7 11 9 11 11 11 ...
##  $ HHIncomeMid     : int [1:10000] 30000 30000 30000 22500 40000 87500 60000 87500 87500 87500 ...
##  $ Poverty         : num [1:10000] 1.36 1.36 1.36 1.07 1.91 1.84 2.33 5 5 5 ...
##  $ HomeRooms       : int [1:10000] 6 6 6 9 5 6 7 6 6 6 ...
##  $ HomeOwn         : Factor w/ 3 levels "Own","Rent","Other": 1 1 1 1 2 2 1 1 1 1 ...
##  $ Work            : Factor w/ 3 levels "Looking","NotWorking",..: 2 2 2 NA 2 NA NA 3 3 3 ...
##  $ Weight          : num [1:10000] 87.4 87.4 87.4 17 86.7 29.8 35.2 75.7 75.7 75.7 ...
##  $ Length          : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ HeadCirc        : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ Height          : num [1:10000] 165 165 165 105 168 ...
##  $ BMI             : num [1:10000] 32.2 32.2 32.2 15.3 30.6 ...
##  $ BMICatUnder20yrs: Factor w/ 4 levels "UnderWeight",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ BMI_WHO         : Factor w/ 4 levels "12.0_18.5","18.5_to_24.9",..: 4 4 4 1 4 1 2 3 3 3 ...
##  $ Pulse           : int [1:10000] 70 70 70 NA 86 82 72 62 62 62 ...
##  $ BPSysAve        : int [1:10000] 113 113 113 NA 112 86 107 118 118 118 ...
##  $ BPDiaAve        : int [1:10000] 85 85 85 NA 75 47 37 64 64 64 ...
##  $ BPSys1          : int [1:10000] 114 114 114 NA 118 84 114 106 106 106 ...
##  $ BPDia1          : int [1:10000] 88 88 88 NA 82 50 46 62 62 62 ...
##  $ BPSys2          : int [1:10000] 114 114 114 NA 108 84 108 118 118 118 ...
##  $ BPDia2          : int [1:10000] 88 88 88 NA 74 50 36 68 68 68 ...
##  $ BPSys3          : int [1:10000] 112 112 112 NA 116 88 106 118 118 118 ...
##  $ BPDia3          : int [1:10000] 82 82 82 NA 76 44 38 60 60 60 ...
##  $ Testosterone    : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ DirectChol      : num [1:10000] 1.29 1.29 1.29 NA 1.16 1.34 1.55 2.12 2.12 2.12 ...
##  $ TotChol         : num [1:10000] 3.49 3.49 3.49 NA 6.7 4.86 4.09 5.82 5.82 5.82 ...
##  $ UrineVol1       : int [1:10000] 352 352 352 NA 77 123 238 106 106 106 ...
##  $ UrineFlow1      : num [1:10000] NA NA NA NA 0.094 ...
##  $ UrineVol2       : int [1:10000] NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ UrineFlow2      : num [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ Diabetes        : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DiabetesAge     : int [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ HealthGen       : Factor w/ 5 levels "Excellent","Vgood",..: 3 3 3 NA 3 NA NA 2 2 2 ...
##  $ DaysPhysHlthBad : int [1:10000] 0 0 0 NA 0 NA NA 0 0 0 ...
##  $ DaysMentHlthBad : int [1:10000] 15 15 15 NA 10 NA NA 3 3 3 ...
##  $ LittleInterest  : Factor w/ 3 levels "None","Several",..: 3 3 3 NA 2 NA NA 1 1 1 ...
##  $ Depressed       : Factor w/ 3 levels "None","Several",..: 2 2 2 NA 2 NA NA 1 1 1 ...
##  $ nPregnancies    : int [1:10000] NA NA NA NA 2 NA NA 1 1 1 ...
##  $ nBabies         : int [1:10000] NA NA NA NA 2 NA NA NA NA NA ...
##  $ Age1stBaby      : int [1:10000] NA NA NA NA 27 NA NA NA NA NA ...
##  $ SleepHrsNight   : int [1:10000] 4 4 4 NA 8 NA NA 8 8 8 ...
##  $ SleepTrouble    : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 1 1 1 ...
##  $ PhysActive      : Factor w/ 2 levels "No","Yes": 1 1 1 NA 1 NA NA 2 2 2 ...
##  $ PhysActiveDays  : int [1:10000] NA NA NA NA NA NA NA 5 5 5 ...
##  $ TVHrsDay        : Factor w/ 7 levels "0_hrs","0_to_1_hr",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ CompHrsDay      : Factor w/ 7 levels "0_hrs","0_to_1_hr",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ TVHrsDayChild   : int [1:10000] NA NA NA 4 NA 5 1 NA NA NA ...
##  $ CompHrsDayChild : int [1:10000] NA NA NA 1 NA 0 6 NA NA NA ...
##  $ Alcohol12PlusYr : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 2 2 2 ...
##  $ AlcoholDay      : int [1:10000] NA NA NA NA 2 NA NA 3 3 3 ...
##  $ AlcoholYear     : int [1:10000] 0 0 0 NA 20 NA NA 52 52 52 ...
##  $ SmokeNow        : Factor w/ 2 levels "No","Yes": 1 1 1 NA 2 NA NA NA NA NA ...
##  $ Smoke100        : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 1 1 1 ...
##  $ Smoke100n       : Factor w/ 2 levels "Non-Smoker","Smoker": 2 2 2 NA 2 NA NA 1 1 1 ...
##  $ SmokeAge        : int [1:10000] 18 18 18 NA 38 NA NA NA NA NA ...
##  $ Marijuana       : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 2 2 2 ...
##  $ AgeFirstMarij   : int [1:10000] 17 17 17 NA 18 NA NA 13 13 13 ...
##  $ RegularMarij    : Factor w/ 2 levels "No","Yes": 1 1 1 NA 1 NA NA 1 1 1 ...
##  $ AgeRegMarij     : int [1:10000] NA NA NA NA NA NA NA NA NA NA ...
##  $ HardDrugs       : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 1 1 1 ...
##  $ SexEver         : Factor w/ 2 levels "No","Yes": 2 2 2 NA 2 NA NA 2 2 2 ...
##  $ SexAge          : int [1:10000] 16 16 16 NA 12 NA NA 13 13 13 ...
##  $ SexNumPartnLife : int [1:10000] 8 8 8 NA 10 NA NA 20 20 20 ...
##  $ SexNumPartYear  : int [1:10000] 1 1 1 NA 1 NA NA 0 0 0 ...
##  $ SameSex         : Factor w/ 2 levels "No","Yes": 1 1 1 NA 2 NA NA 2 2 2 ...
##  $ SexOrientation  : Factor w/ 3 levels "Bisexual","Heterosexual",..: 2 2 2 NA 2 NA NA 1 1 1 ...
##  $ PregnantNow     : Factor w/ 3 levels "Yes","No","Unknown": NA NA NA NA NA NA NA NA NA NA ...
```

```
head(NHANES)
```

```
## # A tibble: 6 x 76
##      ID Surve~1 Gender   Age AgeDe~2 AgeMo~3 Race1 Race3 Educa~4 Marit~5 HHInc~6
##   <int> <fct>   <fct>  <int> <fct>     <int> <fct> <fct> <fct>   <fct>   <fct>
## 1 51624 2009_10 male      34 " 30-3~     409 White <NA>  High S~ Married 25000-~
## 2 51624 2009_10 male      34 " 30-3~     409 White <NA>  High S~ Married 25000-~
## 3 51624 2009_10 male      34 " 30-3~     409 White <NA>  High S~ Married 25000-~
## 4 51625 2009_10 male       4 " 0-9"       49 Other <NA>  <NA>    <NA>    20000-~
## 5 51630 2009_10 female    49 " 40-4~     596 White <NA>  Some C~ LivePa~ 35000-~
## 6 51638 2009_10 male       9 " 0-9"      115 White <NA>  <NA>    <NA>    75000-~
## # ... with 65 more variables: HHIncomeMid <int>, Poverty <dbl>,
## #   HomeRooms <int>, HomeOwn <fct>, Work <fct>, Weight <dbl>, Length <dbl>,
## #   HeadCirc <dbl>, Height <dbl>, BMI <dbl>, BMICatUnder20yrs <fct>,
## #   BMI_WHO <fct>, Pulse <int>, BPSysAve <int>, BPDiaAve <int>, BPSys1 <int>,
```

```
## #   BPDia1 <int>, BPSys2 <int>, BPDia2 <int>, BPSys3 <int>, BPDia3 <int>,
## #   Testosterone <dbl>, DirectChol <dbl>, TotChol <dbl>, UrineVol1 <int>,
## #   UrineFlow1 <dbl>, UrineVol2 <int>, UrineFlow2 <dbl>, Diabetes <fct>, ...
```

You'll notice that there are 76 variables and 10,000 observations in this dataset. The variables are a mix of numerical and categorical variables. For this exercise, we will focus on the numerical average diastolic blood pressure variable (BPDiaAve) and the categorical Gender variable.

Task

Imagine you want to investigate the difference in BPDiaAve between adult males and females, check whether the normality assumption holds for each subgroup. Remember to filter the dataset based on the numerical Age variable to keep only adult individuals (aged 18 or over), and based on the Gender to look at each subgroup separately.

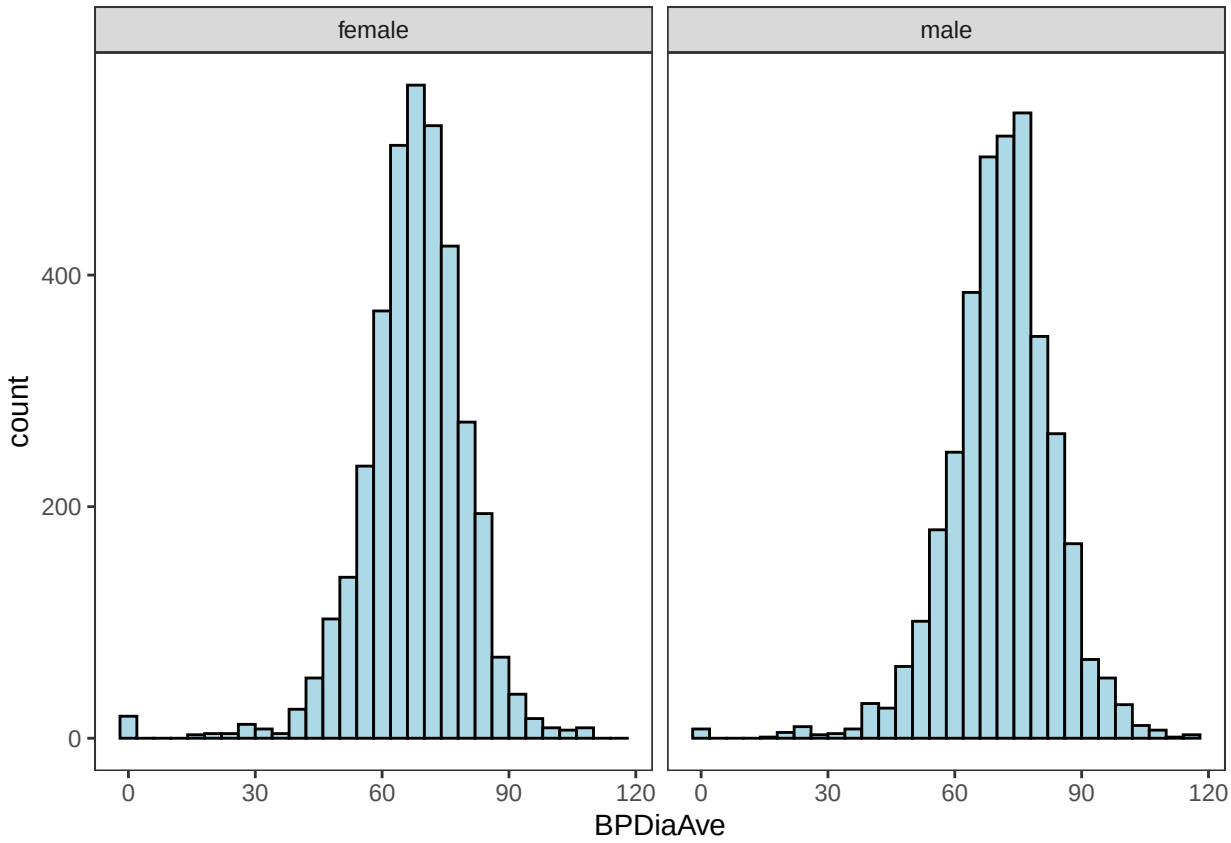
Solution

We first create a subset of the NHANES dataset with only male individuals aged 18 and over, who have both BPDiaAve and Gender information.

```
subsetNHANES <-NHANES %>%
  filter(Age >= 18) %>%
  drop_na(BPDiaAve,Gender) # remove rows where BPDiaAve is NA
```

Then we can check the assumption of normal distribution of the BPDiaAve variable. We can plot a histogram for BPDiaAve for each Gender.

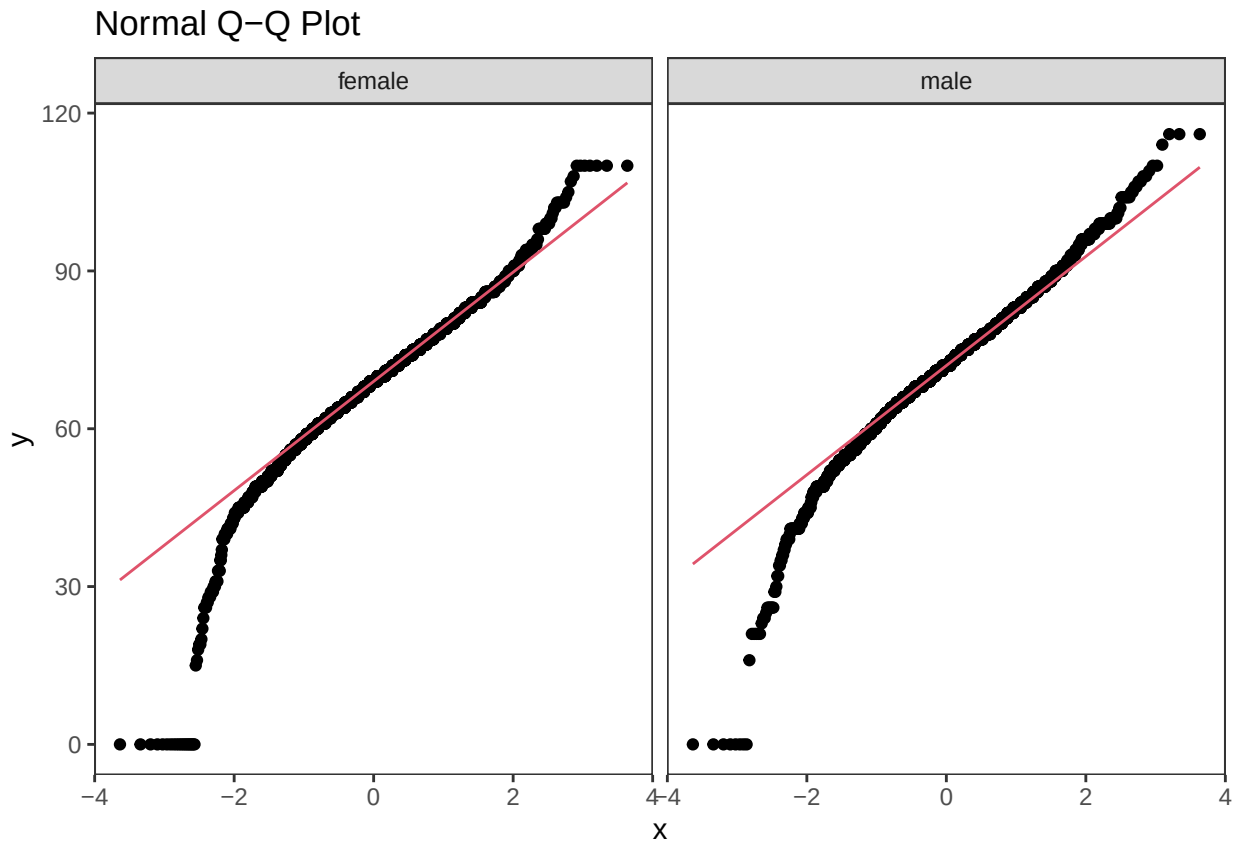
```
subsetNHANES %>%
  ggplot(aes(x = BPDiaAve)) +
  geom_histogram(col="black",fill="lightblue",bins = 30 ) +
  facet_wrap(~Gender) +              ## plot histograms for each gender side-by-side
  theme_bw() +                       ## remove gray background
  theme(panel.grid=element_blank())  ## remove grid
```



The histograms look close to normal, but there is a slightly longer left tail, suggesting that the data has a negative (or left-) skew.

We can also plot a Normal Q-Q plot for BPDiaAve for each Gender.

```
subsetNHANES %>%
  ggplot(aes(sample=BPDiaAve)) +
  stat_qq() +
  stat_qq_line(color=2) +
  facet_wrap(~Gender) +              ## plot Q-Q plots for each gender side-by-side
  labs(title="Normal Q-Q Plot") +    ## add title
  theme_bw() +                       ## remove gray background
  theme(panel.grid=element_blank())  ## remove grid
```



The QQ-plots confirm the observations from the histogram and also show a deviation from normality on the right-hand side of the plot.

Moving on to hypothesis tests, we perform the Shapiro-Wilk test on the BPDiaAve variable for each Gender

```
# Males
subsetNHANES %>%
  filter(Gender=="male") %>%     ## keep only male individuals
  pull(BPDiaAve) %>%
  shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.96636, p-value < 2.2e-16
```

```
# Females
subsetNHANES %>%
  filter(Gender=="female") %>%     ## keep only female individuals
  pull(BPDiaAve) %>%
  shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  .
## W = 0.9393, p-value < 2.2e-16
```

We can also use the Kolmogorov-Smirnov test BPDiaAve to test for normality.

```
# Males
subsetNHANES %>%
  filter(Gender=="male") %>%     ## keep only male individuals
  pull(BPDiaAve) %>%
  ks.test(.,"pnorm",mean=mean(.),sd=sd(.))
```

```
## Warning in ks.test.default(., "pnorm", mean = mean(.), sd = sd(.)): ties should
## not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  .
## D = 0.060667, p-value = 7.132e-12
## alternative hypothesis: two-sided
```

```
# Females
subsetNHANES %>%
  filter(Gender=="female") %>%     ## keep only female individuals
  pull(BPDiaAve) %>%
  ks.test(.,"pnorm",mean=mean(.),sd=sd(.))
```

```
## Warning in ks.test.default(., "pnorm", mean = mean(.), sd = sd(.)): ties should
## not be present for the Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  .
## D = 0.068278, p-value = 4.219e-15
## alternative hypothesis: two-sided
```

Both hypothesis tests show a significant departure from normality, although we should keep in mind that they are sensitive to small departures, especially when looking at a large dataset such as NHANES. The warning for the Kolmogorov-Smirnov test means that there are some ties in the ranks of the data.

Overall, it seems like the distribution of the average diastolic blood pressure is not quite normally distributed in either of the two Gender categories.

Exercise 2

This time, we will use a simulated dataset of birthweights. First we create a dataset (tibble) of 1000 birth weights with a mean of 3510 grams and a standard deviation of 385 grams.

```r
birthweight <- tibble(
  birthwt = rnorm(1000,3510,385)
)
```

This creates a tibble named birthweight with one numerical variable named birthwt and 1000 observations.

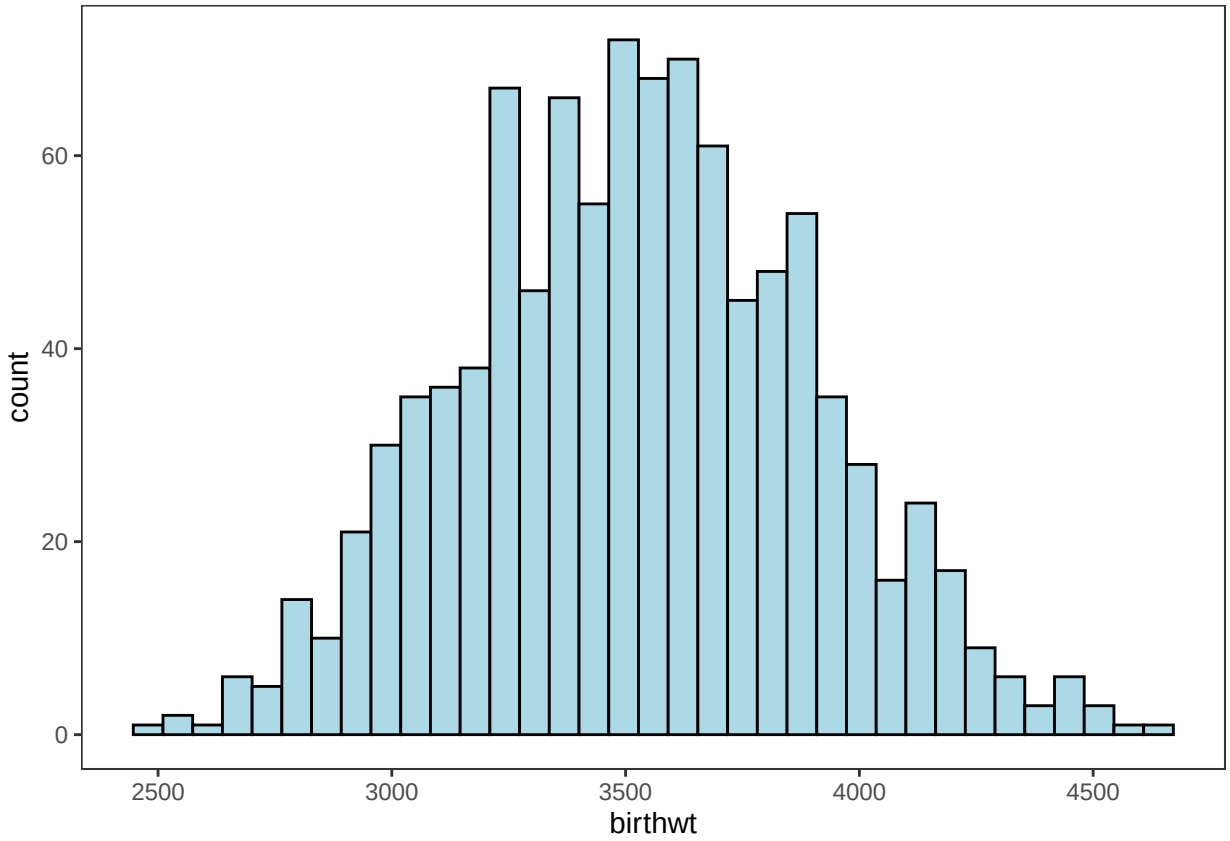
Task

Does the birthwt variable follow a normal distribution?

Solution

We first plot a histogram for the birthwt variable.

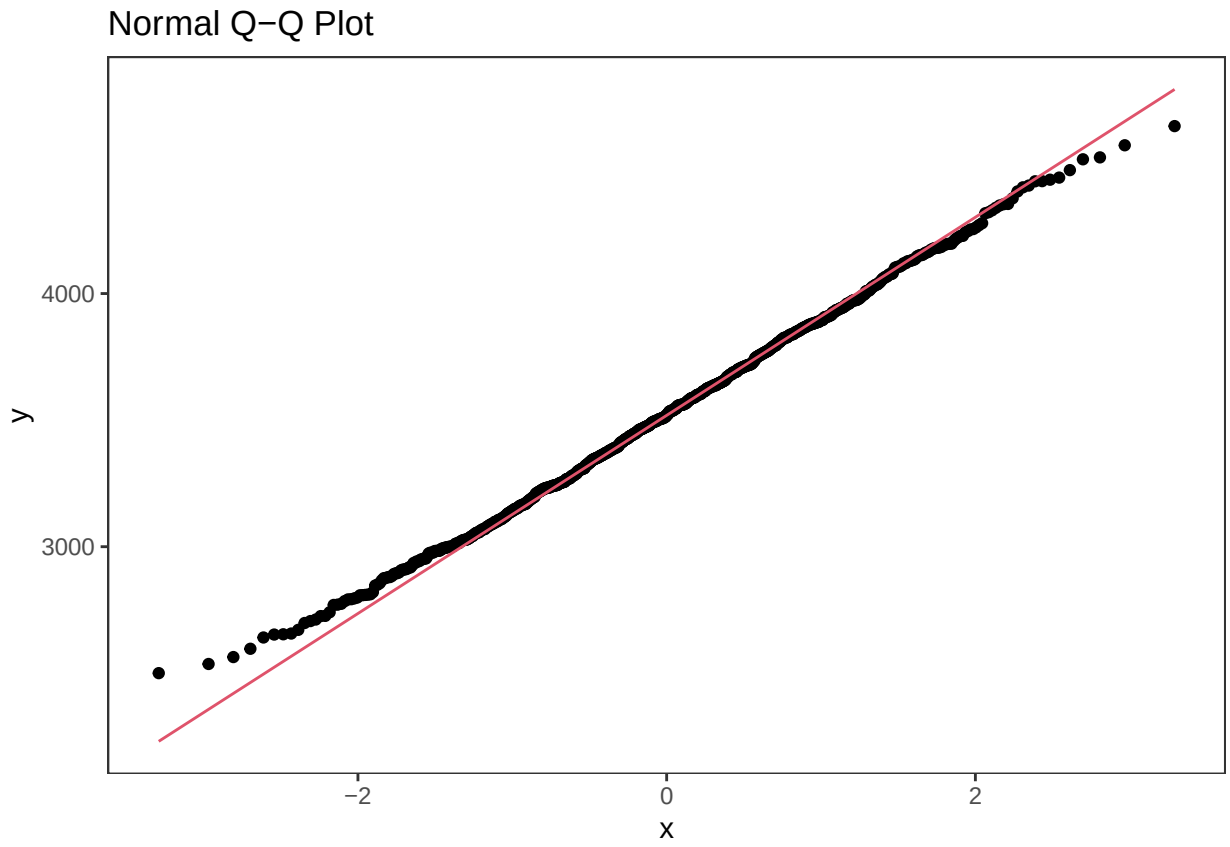
```r
birthweight %>%
  ggplot(aes(x = birthwt)) +
  geom_histogram(col="black",fill="lightblue",bins = 35 ) +
  theme_bw() +                          ## remove gray background
  theme(panel.grid=element_blank())   ## remove grid
```



The histogram seems close to the bell shape expected for a normal distribution.

We can also create a Normal Q-Q plot for birthwt.

```
birthweight %>%
  ggplot(aes(sample=birthwt)) +
  stat_qq() +
  stat_qq_line(color=2) +
  labs(title="Normal Q-Q Plot") +     ## add title
  theme_bw() +                       ## remove gray background
  theme(panel.grid=element_blank())  ## remove grid
```



The Q-Q plot also suggests that the data follows the normal quantiles for most of the data points, with a few data points at either end departing slightly from it.

Moving on to hypothesis tests, we can perform the Shapiro-Wilk test on the birthwt variable.

```
birthweight %>%
  pull(birthwt) %>%
  shapiro.test()
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.99822, p-value = 0.3869
```

Finally, we can test for normality of birthwt using the Kolmogorov-Smirnov test.

```r
birthweight %>%
  pull(birthwt) %>%
  ks.test(.,"pnorm",mean=mean(.),sd=sd(.))
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  .
## D = 0.019469, p-value = 0.8429
## alternative hypothesis: two-sided
```

Both hypothesis tests give a p-value higher than 0.05, therefore we cannot reject the null hypothesis that the data is normally distributed.

Overall, it seems that the birthweight variable is normally distributed. You might have noticed that the function we used to create that data was actually sampling from a normal distribution, so we expected this to be normally distributed.