Bar Charts as a Means of Presenting Multiple Samples of Quantitative Data

In chapter 5, we championed boxplots (or violin plots) as normally being the best option for presenting multiple samples of quantitative data, with histograms also occasionally being a reasonable choice. A final data presentation option for when you have multiple samples from which continuous data (or many possible values of discrete data) have been collected is a bar chart. As a reminder, the bar charts we will cover next concern quantitative data (measurements on a numerical scale), rather than qualitative (or categorical) data as covered in chapter 3. With qualitative data, we plotted the count or frequency in each level of a variable; with quantitative data, we can plot the mean values of each sample as a bar chart, as well as giving an indication of the variability using error bars. There is a range of possibilities in terms of what you can get your error bars to represent. You could show means with standard deviation, standard error, or 95% confidence intervals. Alternatively, you could even present medians with quartiles for each sample. However, we think boxplots (and violin plots) are just as compact as bar charts, and can provide all the same information and more. For this reason, we don’t recommend using bar charts for this sort of data. We also recommend the paper by Newman and Scholl (2012) listed in the ‘Further Reading’ at the end of this document for further reasons why bar charts can be a poor choice for quantitative data. However, we cannot deny that bar charts are commonly used in this context. This may be because they are more visually appealing than boxplots to many. If you feel this is an important issue leading you towards using bar charts this way, then we provide some advice on how to do that below. However, do bear the following general issues in mind:

* Think carefully about whether bars should show the mean or median.
* Make sure you use error bars to capture the within-sample variation in addition to your measure of central tendency (mean or median).
* Think carefully about which measure of variation to use.

The bar charts we create below use the downloadable Excel .csv file associated with chapter 5 (‘pig\_litters’), which gives the birthweights (in lbs) of Poland China pigs from seven different litters (adapted from Snedecor 1956). In case you are working through this material separately from the chapter, here is some of the code used in sections 5.2 and 5.3.2 to get the data loaded in and tidied-up:

**piglets <- read.table(file.choose(), header = T, sep = ",")**

**cleanpiglets <- lapply(piglets, function(col)col[!is.na(col)])**

**lit1<-cleanpiglets$litter1**

**lit2<-cleanpiglets$litter2**

**lit3<-cleanpiglets$litter3**

**lit4<-cleanpiglets$litter4**

**lit5<-cleanpiglets$litter5**

**lit6<-cleanpiglets$litter6**

**lit7<-cleanpiglets$litter7**

**n <- c(length(lit1),length(lit2),length(lit3),length(lit4),**

**length(lit5),length(lit6),length(lit7))**

# Creating a simple bar chart

***Step 1*:** Plotting means and standard deviations for samples of data is a common convention, so that is what we will do for our piglet birth weight data. First, we need to create a list of the mean birth weights of each litter (**y.means**), using the function **mean**, in a similar fashion to how we created a list of sample sizes using length above and in section 5.3.2.For ease, we’ll use the shortened names for the litters that were defined above and in section 5.3.2:

**y.means <- c(mean(lit1),mean(lit2),mean(lit3),mean(lit4), mean(lit5),mean(lit6),mean(lit7))**

Note that it’s best not to call a list of values in R by the name of a function that already exists in R, or Rcan get confused. That’s why our list here is called **y.means** (means that will be plotted against the y-axis)rather than just **mean**.

***Step 2*:** With this list of means it is now very easy to produce a bar chart using code introduced in chapter 3, though this time we will also give the plot a name (‘**mybar**’) using the **<-** assignment operator, which will make adding error bars easier later on:

**mybar<- barplot(y.means, names.arg=c(1:7), ylab="Mean birth weight (lb)", xlab="Litter", col="cadetblue1", ylim=c(0,4),yaxs = "i")**

**abline(h=0)**

For our x-axis tick labels (**names.arg**) we have set a list of numbers running from 1-7 (**c(1:7)**) to label each litter, and given the whole x-axis the label (**xlab**) ‘Litter’. We have also drawn on an x-axis running from 0 on the y-axis using **abline**.

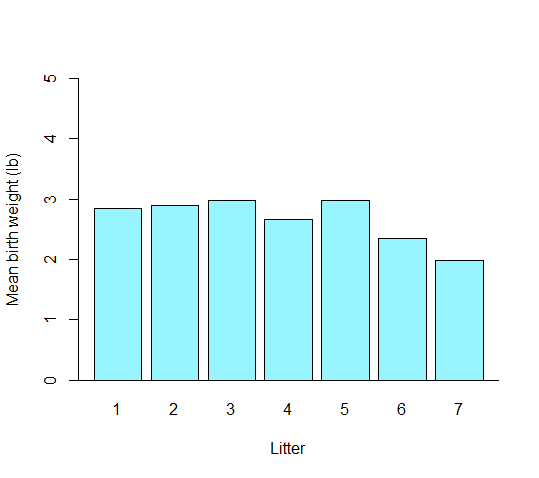


Figure A.5.1: A simple bar chart showing the mean birthweights (lbs) of Poland China pigs from seven different litters (adapted from Snedecor 1956).

Now the means are plotted (as in Figure A.5.1), we can turn to our measure of variability.

***Step 3*:** To prepare our variability measure for plotting, we need to create a list of the standard deviations of each sample. The function for this is simply **sd**:

**y.sd <- c(sd(lit1),sd(lit2),sd(lit3),sd(lit4),sd(lit5),sd(lit6),sd(lit7))**

Again, it’s worth noting that our list is called **y.sd**, not simply **sd,** so as to not confuse R.

***Step 4*:** Now, we can add error bars to the bar chart we named ‘**mybar**’ using a special kind of arrowhead in the function **arrows**:

**arrows(mybar, y.means-y.sd, mybar, y.means+y.sd, length=0.1, angle=90, code=3)**

Although **arrows** can, as its name suggests, be used to draw arrows between points on a figure (see section 8.6 for details), we can specify a particular kind of ‘arrowhead’ that will allow us to draw error bars using a base R function. The first four components listed in the **arrows** function are the coordinates given in the order x0, y0, x1, y1. By using **mybar** for our x-coordinates, we are telling R that the ‘arrows’, our error bars, should run vertically at the centre of each bar in the existing figure. We give our y-coordinates as the lowest reach of the error bar (that is the mean each bar represents minus its corresponding standard deviation) and the highest reach of the error bar (the mean each bar represents plus its corresponding standard deviation). **length** specifies the size of the ‘arrowhead’ (i.e. the caps of our error bars) in inches; you can edit this depending on the width of your bars, but 0.1 looks good to us here. **angle** specifies the angle of the ‘arrowhead’ relative to the shaft of the arrow, and a value of 90 allows our caps to sit perpendicular to the vertical error bar. Finally, the **code** parameter specifies that we want to draw a particular kind of ‘arrowhead’ on both ends of the ‘arrow’ (error bar).

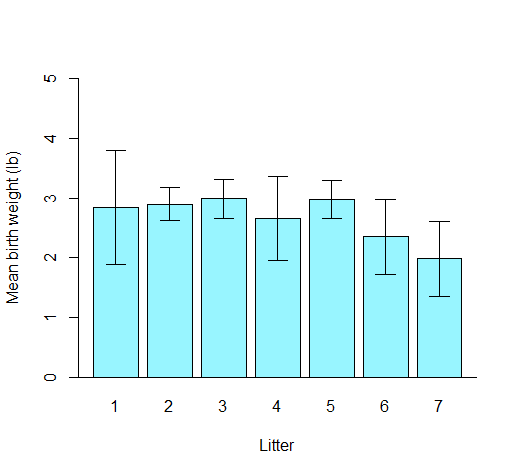


Figure A.5.2: As Figure A.5.1 but with the standard deviation of each sample included as error bars.

Figure A.5.2 now looks pretty smart. As with the boxplots, though, our plotting obscures how many values were in each sample of data (i.e. how many piglets were in each litter). This can be amended by adding on our sample sizes as text.

# Refining our bar chart: adding sample sizes as text at various heights

***Step 1*:** Now that our bar chart has been given a name, ‘**mybar**’, and given that we already have the list ‘**n**’ of sample sizes from earlier in this additional guide, it is easy to add the sample sizes on as text. Instead of typing out all the x-coordinates, we can just say ‘**mybar**’ in the text code and R will know that we want the list of text positioned separately at each bar. Try running this code:

**text(mybar, 4.5, labels=paste("n=",n,sep=""),col="royalblue3")**

**4.5** just refers to the y-axis coordinates. This time when adding our sample sizes, instead of just printing the values of **n** (using **labels= c(n)** as we did in section 5.3.2), here we use **paste** to put **"n="** before each value of **n**, and **sep=""** just separates out the values. You can see how this looks in Figure A.5.3.

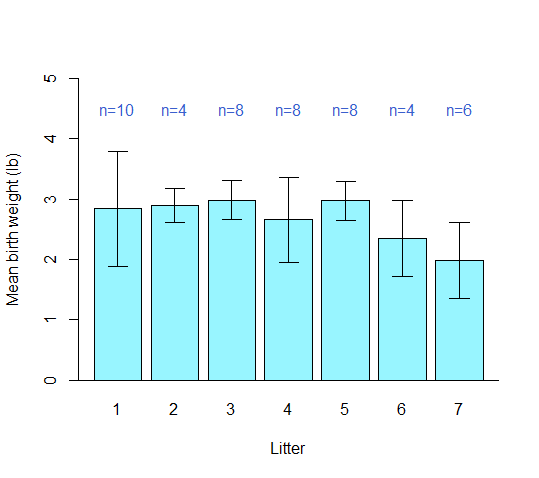


Figure A.5.3: A high-quality bar chart showing the mean (± standard deviation) birthweights (lbs) of Poland China pigs from seven different litters (adapted from Snedecor 1956), with litter sizes shown in dark blue above bars.

***Step 2*:** Alternatively, we could place the sample sizes at varying heights up the y-axis depending on the heights of the bars. Instead of them all being at **4.5** lbs on the y-axis, we can try making them all 1.0 lb higher on the y-axis than their corresponding error bar to demonstrate this. We do this by first drawing the base figure with error bars again (this time naming it ‘**altbar**’):

**altbar<-barplot(y.means, names.arg=c(1:7), ylab="Mean birth weight (lb)", xlab="Litter",**

**col="cadetblue1", ylim=c(0,5),yaxs = "i")**

**abline(h=0)**

**arrows(altbar, y.means-y.sd, altbar, y.means+y.sd, length=0.1, angle=90, code=3)**

***Step 3:*** We then change the y-axis coordinate value in the **text** code from a standard **4.5** for all sample sizes to **1.0** greater than the mean value (**y.means**) plus the standard deviation (**y.sd**) of each corresponding bar:

**text(altbar, y.means+y.sd+1.0, labels=paste("n=",n,sep=""),col="royalblue3")**

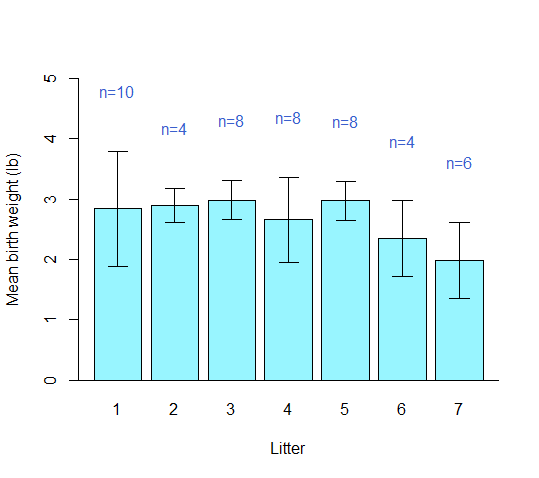


Figure A.5.4: As Figure A.5.4, but with the sample sizes all positioned 1.0 lb higher on the y-axis than the top of their corresponding error bar.

In Figure A.5.4, all of the sample sizes are positioned 1.0 lb higher on the y-axis than the top of the corresponding error bar for their litter.

# Grouped multiple samples bar charts for quantitative data

Grouped bar charts for qualitative data were already covered in section 3.4, but the data from the seed germination experiment described in section 5.6.1 is quantitative. We will use similar methods here to display this data as a grouped multiple-samples bar chart, but with added code that will be familiar from our earlier discussion in this supplement and from chapter 5. First, though, we will need to load and subset the data as we did in section 5.6.2, using the following code:

**seeds <- read.table(file.choose(), header = T, sep = ",")**

**uncovered <- subset(seeds, treatment=="uncovered")**

**covered <- subset(seeds, treatment=="covered")**

Now we should organize the data that we’ll be plotting by preparing lists of the means and standard deviations for each treatment under all five watering levels; we will introduce a new, time-saving function to do this.

***Step 1*:** We will first need to calculate the means we are going to plot as bars, as we did above. However, the **seeds** germination data does not have the values for each treatment arranged in easy-to-access lists—there are three columns to the data, with the **germinated** values corresponding to both a **treatment** and the level of **water** received. In order to easily extract the mean number of seeds germinated under each watering level for both treatments, we can use the **tapply** function on subsets of the data for each treatment. First, for the subsetted data **uncovered**:

**tapply(uncovered$germinated, uncovered$water, mean)**

The **tapply** function can be used to break data into groups defined by one of its variables, compute a function on each of the subsets of the data, and return the results in an accessible form. The structure we have used here asks R to look at our germinated seed values (**uncovered$germinated**), to split them into groups depending on the watering level (**uncovered$water**), and to compute the **mean** number of seeds that germinated at each watering level.

The output looks like this:

1 2 3 4 5

24.25 46.00 66.75 78.00 72.75

The first row is our different groups (watering levels), running 1-5, and the second row is the corresponding means.

***Step 2*:** We can now simply create a list of these values as the means to be plotted for our uncovered treatment data:

**uncovmeans <- c(24.25, 46.00, 66.75, 78.00, 72.75)**

***Step 3*:** Fortunately, **tapply** is versatile, and so we can also use it to calculate the standard deviations for the uncovered treatment data before turning these into a list as well:

**tapply(uncovered$germinated, uncovered$water, sd)**

**uncovsd <- c(2.217356, 9.273618, 11.586630, 5.830952, 4.856267)**

***Step 4*:** We can calculate and list both the means and standard deviations for the covered treatment data in exactly the same way:

**tapply(covered$germinated, covered$water, mean)**

**covmeans <- c(42.75000, 75.25000, 76.25000, 52.00000, 37.33333)**

**tapply(covered$germinated, covered$water, sd)**

**covsd <- c(1.707825, 6.946222, 3.304038, 9.201449, 7.094599)**

***Step 5*:** Now, as needed for grouped bars in chapter 3, we will combine our lists of means from the two treatments into a single dataset (**seedmeans**) with two rows:

**seedmeans <- rbind(uncovmeans,covmeans)**

View the new dataset **seedmeans** if you need a reminder of how **rbind** works.

***Step 6*:** Using **seedmeans** and the all-important **beside=TRUE** (used previously in chapter 3), we can now easily plot a grouped bar chart of our data, remembering to give it a name (**seedbar**) for ease of adding error bars shortly):

**seedbar<-barplot(seedmeans, beside=TRUE,**

**names.arg=c(1:5), ylab="Seeds germinating per box",**

**xlab="Amount of water",**

**col=c("gray","yellow"), ylim=c(0,100),yaxs = "i")**

**abline(h=0)**

***Step 7*:** This time, before we add our error bars, we will need to arrange our lists of standard deviations into two rows, as we did with the means. This is needed so that they will be added correctly to our grouped bars, alternating uncovered and covered for each level of watering.

**seedsd <- rbind(uncovsd,covsd)**

***Step 8*:** We can now add the error bars as easily as we did earlier in the chapter, using **arrows**:

**arrows(seedbar, seedmeans-seedsd, seedbar, seedmeans+seedsd, length=0.06, angle=90, code=3)**

You can see that we have arranged our coordinates in the same way as earlier in the chapter, and kept the **angle** and **code** parameters the same to ensure we draw error bars with caps at either end. The only argument we have changed notably is **length**, which has been reduced because our grouped bars are narrower than they were in the earlier figures and so we will need narrower caps on the ends of our error bars.

***Step 9*:** Again, because we have plotted the data from the two different treatments alongside each other, we need to finish off our figure with a legend that explains our use of different colours:

**legend("topleft", bty="n", legend=c("Uncovered","Covered"), fill=c("gray","yellow"))**

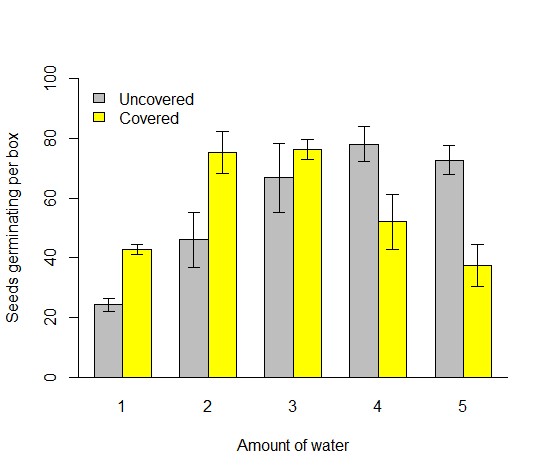


Figure A.5.5: The mean number of seeds (± SD) that germinated in uncovered (grey) and covered (yellow) boxes experiencing different levels of watering (n=4 for all except covered boxes with a watering level of 5, for which n=3). Adapted from data from Chatfield (1982).

In Figure A.5.5, we now have the means and standard deviations of our data presented clearly, with uncovered and covered treatments grouped neatly together at each watering level along the x-axis. As to which figure type—boxplot or bar char—is the better choice for presenting grouped multiple samples data, the principles outlined in section 5.5 still stand and the answer will depend on what you want to show from your data.

We have a final small disclaimer, mentioned in section 5.6.2. Ordinarily we would recommend treating data with only four discrete values for each treatment as qualitative data rather than quantitative data (therefore tables, pie charts, or bar charts without error bars, as covered in chapters 2 and 3, would be better presentation tools). However, here we wanted to demonstrate grouped multiple-samples plots using data from a clear and uncomplicated experiment, without an overwhelming volume of data.

# Further reading:

* See the below paper for further reason why we discourage the use of bar charts for continuous data:

Newman, G.E. and Scholl, B.J., 2012. [Bar graphs depicting averages are perceptually misinterpreted: The within-the-bar bias](https://link.springer.com/article/10.3758/s13423-012-0247-5). Psychonomic bulletin & review, 19(4), pp. 601–607.

* [‘R Function of the Day: tapply’](https://www.r-bloggers.com/r-function-of-the-day-tapply-2/)

# References:

CHATFIELD, C. 1982. Teaching a Course in Applied Statistics. *Journal of the Royal Statistical Society. Series C (Applied Statistics),* 31**,** 272-289.

NEWMAN, G. E. & SCHOLL, B. J. 2012. Bar graphs depicting averages are perceptually misinterpreted: The within-the-bar bias. *Psychonomic Bulletin & Review,* 19**,** 601-607.

SNEDECOR, G. W. 1956. *Statistical Methods,* Ames, Iowa State College Press.