Chapter 1
Variables, Data and Graphs

Section 1.3
Distribution of Data Values for One Variable

David Gerbing

The School of Business
Portland State University

- Distribution of Data Values for One Variable
  - Bar Chart and Pie Chart
  - Histogram
  - Histogram Artifacts and Issues
  - Cumulative Histogram
  - Pareto Chart

1.3a
Bar Chart and Pie Chart

How Often Does Each Value Occur?

First statistical analysis: Counting

- One basic understanding of the values of a categorical variable is how often each value occurs
- There are many examples of ongoing interest for the manager
  - Number of cars sold by each salesperson last week
  - Number of each size of blue jeans in inventory at the current time
  - Number of patients in the emergency room at the beginning of each hour throughout the day
  - Number of patients classified according to the urgency of their needed care, urgent or non-urgent
  - Number of applicants by gender for a job opening

A Basic Statistical Analysis

Count the number of times each value occurs

- **Count** or frequency of occurrence: Number of times a specific value occurs, which directly depends on the size of the sample
- **Proportion** (p) or relative frequency: A value's frequency of occurrence divided by the total number of values
- Proportion of occurrence for the j^{th} value, category: $p_j = \frac{n_j}{n}$
- Ex: Proportion of employees who call in sick on Friday is the count or number of such employees divided by the total number of employees
- The proportion expresses the concept of frequency independent of the sample size, n, by literally "dividing by n"
- **Distribution**: Display a distribution with a table or a graph of each value of a variable and its frequency and/or proportion

Illustration: Frequencies of a Categorical Variable

Sales by SalesPerson at a Car Dealership

- Sales Report: How many cars are each of the four salespeople selling each week?
- For each sale, record the salesperson
- The variable is salesperson, or just Person
- Here are the sales for a specific week, organized as a data table with only a single variable named Person
- How many cars does each salesperson sale for this week?
- Read the data, a `csv file`, into R with the `lessR` function `Read()`, or just `rd()`

```
> d <- Read("http://lessRstats.com/data/
                CarSales.csv")
```

Person
Bill
Cindy
Don
Andy
Don
Bill
Cindy
Cindy
Andy
Bill
Cindy
Cindy
Cindy
Bill
Bill
Andy
Bill
Bill
Bill
Andy
Andy
Don
Cindy
Bill
Cindy
Bill

Frequency Table

Obtain the table and the graph with one function call

- **Bar chart**: Display the frequencies of the values of a categorical variable with the height of each bar proportionate to its frequency, with spaces between the bars
- The lessR `Chart()` function counts the values, and then displays the table and graph

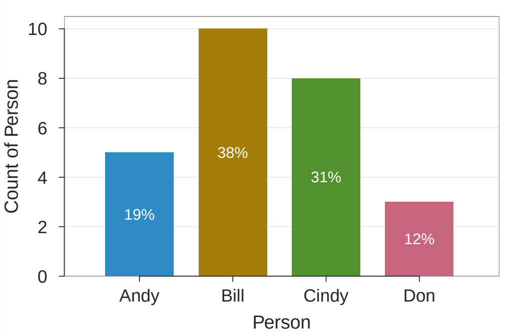
```
> Chart(Person)
```

- The table of the counts and proportions, the frequency table

```
                Andy  Bill Cindy   Don   Total
Frequencies:       5    10     8     3      26
Proportions:   0.192 0.385 0.308 0.115   1.000
```

Bar Chart: Example

```
> Chart(Person)
```

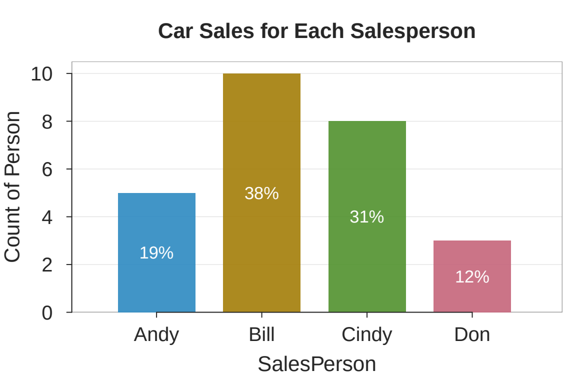


- **Key Concept**: The spaces between the bars of a bar chart indicate the lack of continuity of the categorical data values

Bar Chart with Title

- Use main and xlab options for title and new axis label

```
> Chart(Person, main="Car Sales for
     Each Salesperson", xlab="SalesPerson")
```



Pie Chart

Calculate the pie (ring) chart from the frequencies

- **Pie chart**: Display the frequencies of a categorical variable in which each frequency corresponds to a proportionate slice of a circle (i.e., pie)

The lessR function is `Chart()`, setting `type="pie"`

```
> Chart(Person, type="pie")
```

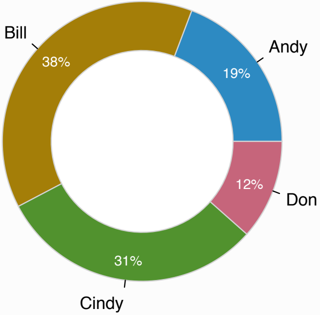


Illustration: Interpretation and Conclusion

Identify the sales performance for each sales person

- Bill and Cindy are the two top sales people for this week in which the data were analyzed, with Bill the overall leader with 10 sales
- Don was the least effective with only 3 sales

Qualify the results with the limitations of the data

- The data are only for a single week, so generalizing to long term performance on this basis is not appropriate
- The data provide one aspect of sales performance, but the data do not consider net profit per sale

1.3b
The Histogram

Measurements of a Continuous Variable

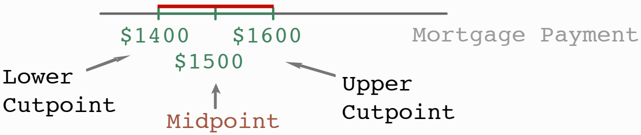
The issue is that there are typically many individual values

- Measurement Problem #1: Too many resulting data values to effectively plot on a single graph
  - Consider mortgage payment, where each single value to the nearest penny must be considered from $300 to $4000 or so
- Measurement Problem #2: Too many data values with a frequency of zero
  - Few specific mortgage payments such as $924.79 would occur at all unless the sample size was extremely large
- **Key Concept**: Group similar data values from a continuous variable together and then assign a single count to each group

Bins (or Classes)

Partition the range of values

- **Bins (classes)**: A sequence of adjacent, non-overlapping intervals, each generally of the same size
- Each bin contains approximately equal data values



- **Cutpoints**: Lower and upper boundaries of each bin
- **Bin width**: Distance between cutpoints
    - In this example, bin width = $200
- **Midpoint**: Single summary of all values within the bin
    - In this example, midpoint = $1500

Bin Assignment

Place data values into the bins

- Assign each data value to its corresponding bin
  - Assign mortgage payment of $1658 to bin: $1600 to $1800
  - Assign mortgage payment of $2336 to bin: $2200 to $2400
- Consistently assign values exactly equal to a cutpoint to either the adjacent lower bin or the adjacent higher bin
  - By default, R assigns a value equal to a cutpoint to the lower bin
  - With R, all values in the bin are larger than the lower cutpoint and smaller than or equal to the upper cutpoint

The Histogram

Graphical display of the variation of a continuous variable

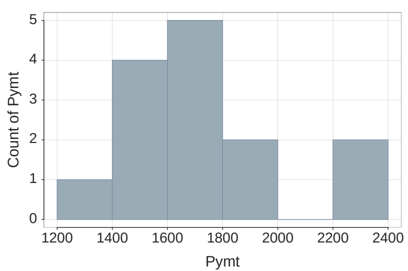
- Usually present the frequency distribution as a graph
- **Histogram**: Place each data value for a continuous variable into its corresponding bin represented by a bar with its height proportional to the frequency of its values
- **Key Concept**: Adjacent bars of a histogram share a common side, no gaps between bars to indicate the underlying continuity
- Ex: The data consists of the mortgage payments of 14 different home owners randomly sampled from one zip code:
- Read the data from the file `mortgage.csv`

```
> d <- Read("http://lessRstats.com/data/mortgage.csv")
```

Example Histogram

- The `lessR` function `X()`, for plotting a single variable on the x-axis, here generates the histogram for a variable named Pymt, the Monthly Mortgage Payment

```
> X(Pymt)
```

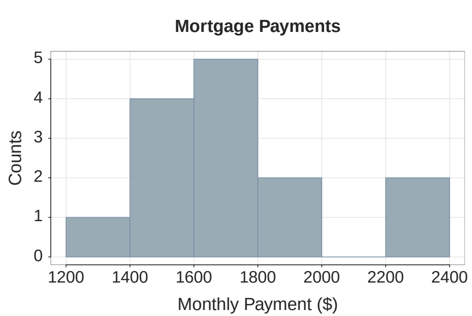


- Interpretation: The range of monthly mortgage payments varies from about $1200 to $2400, with the most common values between $1400 and $2000. Only a few values are above $2000.

Histogram, title and axes labels

- Use the xlab, ylab and main options in virtually any R graphics routine to label the x and y axes and provide a title

```
> X(Pymt, xlab="Monthly Payment ($)",
      ylab="Counts", main="Mortgage Payments")
```



Frequency Table of Bins

Distribution can be presented as a graph or as a table

- A frequency distribution for a variable can also be presented as a table, which includes each bin and corresponding Count, Proportion, Cumulative Count and Cumulative Proportion
- The lessR X() function also provides the frequency distribution as a table

Bin	Midpoint	Count	Prop	Cumul.c	Cumul.p
1200 > 1400	1300	1	0.07	1	0.07
1400 > 1600	1500	4	0.29	5	0.36
1600 > 1800	1700	5	0.36	10	0.72
1800 > 2000	1900	2	0.14	12	0.86
2000 > 2200	2100	0	0.00	12	0.86
2200 > 2400	2300	2	0.14	14	1.00

1.3c
Histogram Artifacts and Issues

The Arbitrariness of a Histogram: Bin Width

Choice of optimal bin width is partially subjective

- **Bin Width artifact**: Change the bin width of a histogram, and the shape of the histogram likely changes
- The final choice of bin width is subjective, so different bin widths should generally be explored beyond whatever default bin width is provided by the computer
- The most efficient way to set bin width manually is to first obtain a histogram with the default bin width, then manually modify the bin width
- **Key Concept**: Select a bin width to display as much detail as possible for the sample size without excessive random noise

The Problem of Oversmoothing for Bin Width

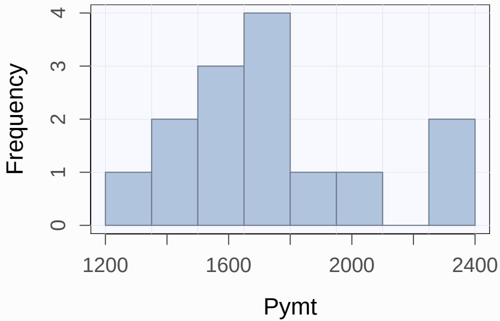
Bin width too large

- **Oversmoothing**: Not enough bins results in the bin width too large, obscuring properties of the underlying distribution
- An oversmoothed histogram provides insufficient detail relative to the available data
- Exploring different bin widths with the previous histogram of Pymt reveals that the default bin width of 200 is somewhat too large, resulting in an oversmoothed histogram
- To demonstrate, re-generate the histogram for Pymt by explicitly specifying bins with a smaller width
- Many possibilities to explicitly specify the bins, *optionally* enter `?Histogram` to view the options
- The easiest method that applies here is to invoke the `bin_width` option for the `lessR` `X()` function

Histogram, with Specified Bins

Bin width set at 150

```
> X(Pymt, bin_width=150)
```



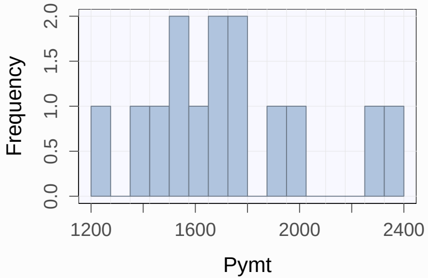
- This histogram with a bin width of 150 provides more meaningful detail than the default bin width of 200

The Problem of Undersmoothing for Bin Width

Bin width too small

- **Undersmoothing**: The bin width is too small relative to the available data so that too many bins result in too much detail

```
> X(Pymt, bin_width=75)
```



- This histogram reflects too much random sampling variability – too many random ups and downs – relative to the likely much smoother true shape of the underlying distribution

The Arbitrariness of a Histogram: Bin Shift

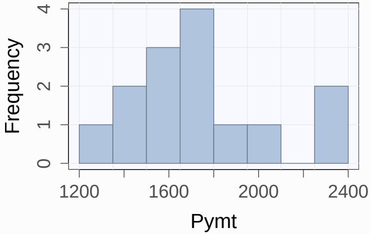
Bin Shift

- **Bin Shift artifact**: Change the starting point of a histogram, and the shape of the histogram likely changes
- There are several possibilities, but the easiest method that applies here is to invoke the `bin_start` option for the `lessR` `X()` function
- If `bin_start` is specified without `bin_width`, then the default bin width is used
- Specifying `bin_start` and `bin_width` together is one way to achieve complete control over the specification of the bins
- There is also a bin.end option to provide an ending point for the bins, useful if to have several histograms of different variables share common starting and ending points

The Arbitrariness of a Histogram: Bin Shift

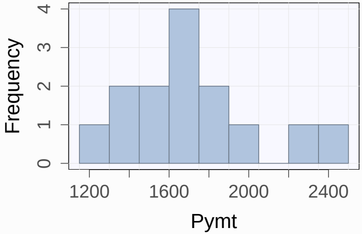
Same data, two different starting points

```
> X(Pymt,
    bin_start=1200,
    bin_width=150)
```



Histogram with bins starting at 1200, with a width of 150

```
> X(Pymt,
    bin_start=1150,
    bin_width=150)
```



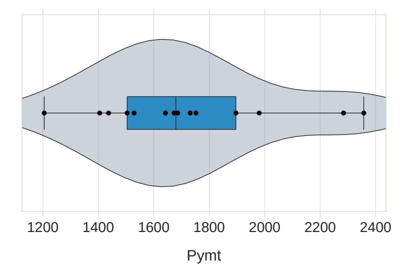
Histogram with bins starting at 1150, with a width of 150

VBS Plot: Integrated Violin, Box, Scatter Plot

More informative alternative to the histogram

- Consider again the 14 Monthly Mortgage Payments
- Use the lessR function X() with type="vbs"

```
> X(Pymt, type="vbs")
```



- This plot is three plots in one: a violin (density) plot, a box plot, and a 1-dimensional scatter plot, or a dot plot

1.3d
The Cumulative Histogram

Cumulative Distribution

"At least" or "at most" questions

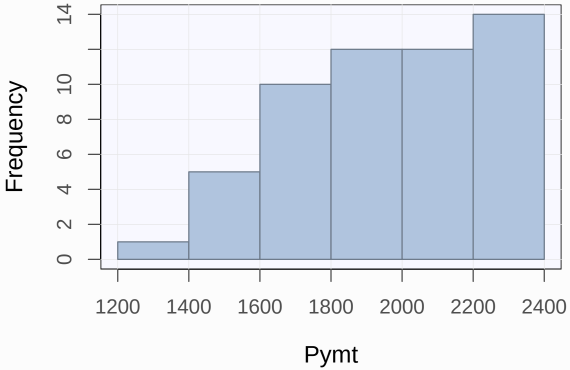
- The values of a continuous distribution are ordered, so it is meaningful to ask questions regarding order
  - How many students got at least 90% on the midterm?
  - How many monthly mortgage payments are below $1500?
- **Cumulative frequency** of a value: Sum of frequencies for all values up to and including the specified value
- **Cumulative proportion** of a value: Sum of proportions for all values up to and including the specified value
- Frequencies are never negative, so as the values of the variable increase, a cumulative distribution always increases in value or stays the same
- **Cumulative histogram**: A histogram of the cumulative distribution of the values of a continuous variable

Cumulative Histogram

Example of a cumulative histogram

- lessR function `X()` with `cumulate` option set to `"on"`

```
> X(Pymt, cumulate="on")
```

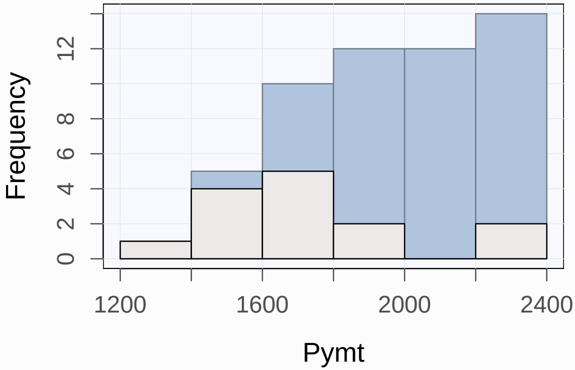


Cumulative and Regular Histograms Together

Regular histogram superimposed on cumulative histogram

- `lessR` function `X()` with `cumulate` option set to `"both"`

```
> X(Pymt, cumulate="both")
```



1.3e
The Pareto Chart

Introducing the Pareto Chart

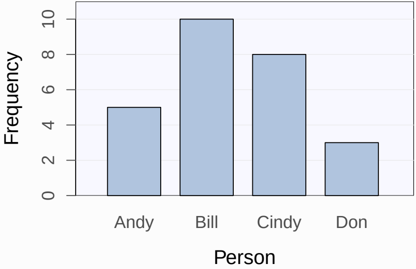
A combination of the bar chart and cumulative distribution

- **Pareto Chart**: Bar chart of categories listed in order of their underlying frequencies, with the plot of the cumulative frequencies superimposed over the corresponding bars
- The Pareto chart is often used in quality control in which the categories . . .
  - represent different types of defects
  - are listed in order from the most frequently occurring defect to the least frequently occurring
- Use the Pareto chart in place of the traditional bar chart in any application in which the ordered frequencies of the values of a categorical variable are of interest

Previous Example of a Bar Chart

Pareto chart provides more information than the bar chart

- Consider, again, the example of sales for a week by the four salespeople at a car dealership
- Data File: `http://lessRstats.com/data/CarSales.csv`
- Bar chart:



- Now obtain the more informative Pareto chart

qcc Pareto Chart, from Data

Count the values of a categorical variable

- Obtain the Pareto chart from the `pareto.chart()` function in the `qcc` package, not initially provided in R
- One time only, first download the package,

```
> install.packages("qcc")
```

- Then, for any one R session, load the functions contained in the package into memory

```
> library(qcc)
```

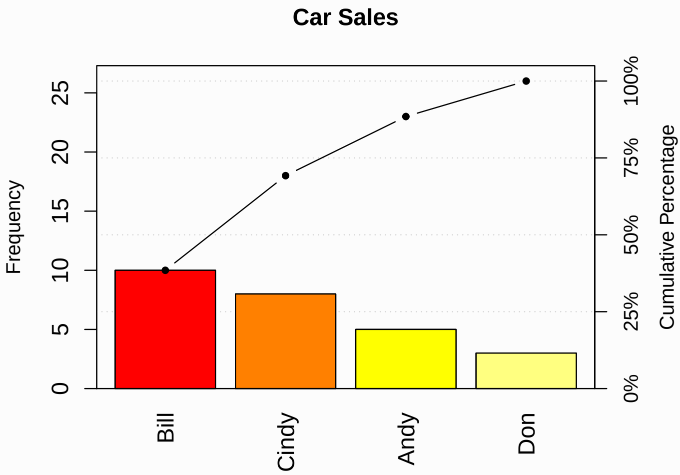
- First invoke lessR function `Chart()` to calculate and then store the counts, here in the object called `myCount`

```
> myCount <- Chart(Person)
```

  Refer to the stored counts in `mycount` as `myCount$freq`

```
> pareto.chart(myCount$freq, main="Car Sales")
```

qcc Pareto Chart



qcc Pareto Chart, from Counts Entered Directly

Car sales by salesperson, once again

- The Pareto chart is computed from the table of counts, which either can be
  - computed from the data, as in the previous example, or
  - entered directly
- Enter the counts directly using the `c()` or "combine" function, illustrated here for Sales by salesperson

```
> myCount <- c(5,10,8,3)
```

- Next specify the category labels

```
> names(myCount) <- c("Andy","Bill","Cindy","Don")
```

- Then call the `pareto.chart()` function

```
> pareto.chart(myCount, main="Car Sales")
```

- The End