UNIT 1

Big Data and the Business Case

Big Data is quickly becoming more than just a buzzword. A plethora of organizations have made significant investments in the technology that surrounds Big Data and are currently starting to leverage the content within to find real value. Even so, there is still a great deal of confusion about Big Data, similar to what many information technology (IT) managers have experienced in the past with disruptive technologies. Big Data is disruptive in the way that it changes how business intelligence (BI) is used in a business—and that is a scary proposition for many senior executives. That situation puts chief technology officers, chief information officers, and IT managers in the unenviable position of trying to prove that a disruptive technology will actually improve business operations. Further complicating this situation is the high cost associated with in house. Big Data processing, as well as the security concerns that surround the processing of Big Data analytics off-site.

Perhaps some of the strife comes from the term Big Data itself. Nontechnical people may think of Big Data literally, as something associated with big problems and big costs. Presenting Big Data as "Big Analytics" instead may be the way to win over apprehensive decision makers while building a business case for the staff, technology, and results that Big Data relies upon.

The trick is to move beyond the accepted definition of Big Data— which implies that it is nothing more than data sets that have become too large to manage with traditional tools—and explain that Big Data is a combination of technologies that mines the value of large databases.

And large is the key word here, simply because massive amounts of data are being collected every second—more than ever imaginable— and the size of these data is greater than can be practically managed by today's current strategies and technologies.

That has created a revolution in which Big Data has become centered on the tsunami of data and how it will change the execution of businesses processes. These changes include introducing greater efficiencies, building new processes for revenue discovery, and fueling innovation. Big Data has quickly grown from a new buzzword being tossed around technology circles into a practical definition for what it is really all about, Big Analytics.

REALIZING VALUE

A number of industries—including health care, the public sector, retail, and manufacturing—can obviously benefit from analyzing their rapidly growing mounds of data. Collecting and analyzing transactional data gives organizations more insight into their customers' preferences, so the data can then be used as a basis for the creation of products and services. This allows the organizations to remedy emerging problems in a timely and more competitive manner.

The use of Big Data analytics is thus becoming a key foundation for competition and growth for individual firms, and it will most likely underpin new waves of productivity, growth, and consumer surplus.

BIG DATA OPTION

Options. There are several paths to take to the destination of Big Data, ranging from in-house big iron solutions (data centers running large mainframe systems) to hosted offerings in the cloud to a hybrid of the two. It is important to research these options and identify how each may work for

achieving Big Data analytics, as well as the pros and cons of each. Preferences and benefits should also be highlighted, allowing a financial decision to be tied to a technological decision.

THE RISE OF BIG DATA OPTIONS

Teradata, IBM, HP, Oracle, and many other companies have been offering terabyte-scale data warehouses for more than a decade, but those offerings were tuned for processes in which data warehousing was the primary goal. Today, data tend to be collected and stored in a wider variety of formats and can include structured, semistructured, and unstructured elements, which each tend to have different storage and management requirements. For Big Data analytics, data must be able to be processed in parallel across multiple servers. This is a necessity, given the amounts of information being analyzed. In addition to having exhaustively maintained transactional data from databases and carefully culled data residing in data warehouses, organizations are reaping untold amounts of log data from servers and forms of machine-generated data, customer comments from internal and external social networks, and other sources of loose, unstructured data.

Such data sets are growing at an exponential rate, thanks to Moore's Law. Moore's Law states that the number of transistors that can be placed on a processor wafer doubles approximately every 18 months. Each new generation of processors is twice as powerful as its most recent predecessor. Similarly, the power of new servers also doubles every 18 months, which means their activities will generate correspondingly larger data sets.

The Big Data approach represents a major shift in how data are handled. In the past, carefully culled data were piped through the network to a data warehouse, where they could be further examined.

However, as the volume of data increases, the network becomes a bottleneck. That is the kind of situation in which a distributed platform, such as Hadoop, comes into play. Distributed systems allow the analysis to occur where the data reside.

Traditional data systems are not able to handle Big Data effectively, either because those systems are not designed to handle the variety of today's data, which tend to have much less structure, or because the data systems cannot scale quickly and affordably. Big Data analytics works very differently from traditional BI, which normally relies on a clean subset of user data placed in a data warehouse to be queried in a limited number of predetermined ways.

Big Data takes a very different approach, in which all of the data an organization generates are gathered and interacted with. That allows administrators and analysts to worry about how to use the data later. In that sense, Big Data solutions prove to be more scalable than traditional databases and data warehouses.

To understand how the options around Big Data have evolved, one must go back to the birth of Hadoop and the dawn of the Big Data movement. Hadoop's roots can be traced back to a 2004 Google white paper that described the infrastructure Google built to analyze data on many different servers, using an indexing system called Bigtable.

Google kept Bigtable for internal use, but Doug Cutting, a developer who had already created the Lucene and Solr open source search engine, created an open source version of Bigtable, naming the technology Hadoop after his son's stuffed elephant.

One of Hadoop's first adopters was Yahoo, which dedicated large amounts of engineering work to refine the technology around 2006.

Yahoo's primary challenge was to make sense of the vast amount of interesting data stored across separated systems. Unifying those data and analyzing them as a whole became a critical goal for Yahoo, and Hadoop turned out to be an ideal platform to make that happen. Today Yahoo is one of the biggest users of Hadoop and has deployed it on more than 40,000 servers. The company uses the technology for multiple business cases and analytics chores. Yahoo's Hadoop clusters hold massive log files of what stories and sections users click on; advertisement activity is also stored, as are lists of all of the content and articles Yahoo publishes. For Yahoo, Hadoop has proven to be well suited for searching for patterns in large sets of text.

BEYOND HADOOP

Another name to become familiar with in the Big Data realm is the Cassandra database, a technology that can store 2 million columns in a single row. That makes Cassandra ideal for appending more data onto existing user accounts without knowing ahead of time how the data should be formatted.

Cassandra's roots can also be traced to an online service provider, in this case Facebook, which needed a massive distributed database to power the service's inbox search. Like Yahoo, Facebook wanted to use the Google Bigtable architecture, which could provide a columnand row-oriented database structure that could be spread on a large number of nodes.

However, Bigtable had a serious limitation: It used a master node– oriented design. Bigtable depended on a single node to coordinate all read-and-write activities on all of the nodes. This meant that if the head node went down, the whole system would be useless.

Cassandra was built on a distributed architecture called Dynamo, which the Amazon engineers who developed it described in a 2007 white paper. Amazon uses Dynamo to keep track of what its millions of online customers are putting in their shopping carts.

Dynamo gave Cassandra an advantage over Bigtable, since Dynamo is not dependent on any one master node. Any node can accept data for the whole system, as well as answer queries. Data are replicated on multiple hosts, creating resiliency and eliminating the single point of failure.

Big Data Analytics Applications

This chapter discusses a number of important use cases for Big Data Analytics. In each case, Big Data Analytics is becoming integrated with business processes and traditional analytics to provide major outcomes. In many cases, these use cases represent game changers essential to the survival and growth of an organization in an increasingly competitive marketplace. Some of these use cases are still in their infancy, while others are becoming increasingly commonplace.

Social Media Command Center

Last year, Blackberry faced a serious outage when its email servers were down for more than a day. I tried powering my Blackberry off and on because I wasn't sure whether it was my device or the CSP. It never occurred to me that the outage could be at the Blackberry server itself. When I called the CSP, they were not aware of the problem. For a while, I was okay without receiving any emails, but then I started to become curious. So I turned to one obvious source: Twitter. Sure enough, I found information about the Blackberry outage on Twitter. One of my clients told me that his VP of Customer Service is glued to Twitter looking for customer service problems. Often,

Figure 3.1: Gatorade Social Media Command Center

someone discovers the problem on Twitter before the internal monitoring organization. We found that a large number of junior staffers employed by marketing, customer service, and public relations search through social media for relevant information. Does this sound like an automation opportunity?

A *Social Media Command Center* combines automated search and display of consumer feedback expressed publicly on the social media. Often, the feedback is summarized in the form of "positive" or "negative" sentiment. Once the feedback is obtained, the marketer can respond to specific comments by entering into a conversation with the affected consumers, whether to respond to questions about an outage or obtain feedback about a new product offering. The marketing organization for Gatorade, a sports drink product, decided to create a Social Media Command Center to increase consumer dialog with Gatorade.12 Figure 3.1 shows the monitoring station with the dashboard. Big Data Analytics can be used to monitor social media for feedback on product, price, and promotions as well as to automate the actions taken in response to the feedback. This may require communication with a number of internal organizations, tracking a product or service problem, and dialog with customers as the feedback results in product or service changes. When consumers provide feedback, the dialog can only be created if the responses are provided in low latency. The automated solutions are far better at systematically finding the information, categorizing it based on available attributes, organizing it into a dashboard, and orchestrating a response at conversation speed.

Product Knowledge Hub

As consumers turn into sophisticated users of technology and the marketplace becomes specialized, the product knowledge seldom belongs to one organization. Take the Apple iPhone as an example. The iPhone is marketed by Apple, but its parts came from a large supply chain pool, the apps running on the iPhone come from a large community of app developers, and the communications service is provided by a CSP. Google's Android is even more diverse, as Google provides the operating system while a cell phone manufacturer makes the device. The smartphones do not work in isolation. They act as WiFi hubs for other devices.

So, what happens if I want to know how to tether an iPhone to an Apple iPad? Do I call my CSP, or do I call Apple? Would either of their websites give me a simple step-by-step process I can follow?

Every time I get into these technical questions about products I am trying to use, I end up calling my son, who happens to know the answers to any such question. Recently, he decided to educate me on how he finds the answer, and so I was introduced to a myriad of third-party sites where a variety of solutions can be found. In most cases, we can find them by searching using any popular

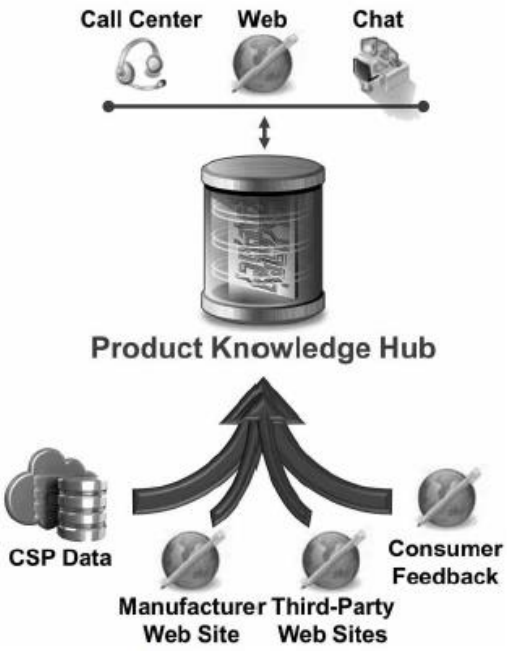


Figure 3.2: Product Knowledge Hub for a CSP

search engine. However, the solutions do not always favor the CSPs, and they are often dated, failing to take into account the latest offerings. Between the device operating system, the offerings from CSPs, and the apps, one must tread carefully through the versions to make sure the solution we discover is for the same version of software that is on the device. So now, we are facing data that is characterized by both variety and veracity. Can we use Big Data Analytics to solve this problem?

The solution involves three sets of technologies. Fortunately, Vivisimo has packaged these technologies into its Velocity product, making it easier to obtain an integrated solution. The first part of the solution is the capability to tap any sources of data. A CSP may already have pieces of the solution on its intranet, put together by product managers or customer service subject matter experts. Or, the information may reside on a device manufacturer site or a third-party site. All this data must be pulled and stripped of its control information so that the raw text is available to be reused. The second part of the solution is to create a set of indices so that the raw information can be categorized and found when needed. Because many combinations of products exist, we would like to collect and combine information for the devices searched. The federated indexing system lets us organize the information for easy access.

The third part of the solution involves creating an XML document against a query that can either be rendered by a mashup engine or made available to a third-party application. What we have created is a *knowledge hub*, which can now be used directly from a website or made available to the call centers. It significantly reduces call-handling time in the call centers and also increases first call resolution. By placing the information on the web, we are now promoting the CSP's website as the source of knowledge, which increases web traffic and reduces the number of people who resort to contacting the call center. Figure 3.2 depicts the Product Knowledge Hub.

Once we have created a single source of knowledge, this source can be used to upsell other products, connecting usage knowledge to product features and using the knowledge pool to discover new product or business partnership ideas. A lot of stray, fragmented knowledge about the products may be rapidly organized and find a variety of other uses.

Infrastructure and Operations Studies

A number of industries are exploring the use of Big Data to improve their infrastructure. In many situations, the best way to improve the infrastructure is to understand its use and how bottlenecks or configurations impact performance. In the past, this data required extensive manual data collection costs. Big Data provides a natural source of data with minimal data collection costs. I will lay out examples from public services to illustrate this point.

The city of Boston decided to use Big Data to identify potholes in the streets by sponsoring a competition in the analyst community. A winner came from

Sprout & Co., a nonprofit group in Somerville, Massachusetts. The solution included the use of magnitude-of-acceleration spikes along a cell phone's z-axis to spot impacts, plus additional filters to distinguish potholes from other irregularities on the road. The new algorithm made Street Bump, a free download in
Apple's App Store, a winner.13 This analysis can save significant road survey cost.
Navigation systems can also use the cell phone data to avoid traffic congestion and offer alternate routes. This type of use of Big Data is one of the best ways to gain acceptance without getting into privacy or security issues. In another example, city bus and train agencies are making their real-time transit information available to riders. This information significantly improves the user experience and reduces the uncertainty associated with both planned and unexpected delays. Transloc (*www.transloc.com*) provides this information for riders using a variety of technologies, including smartphones, web, and SMS messages. It also provides prediction capabilities on expected arrival time. Once the app is loaded on a smartphone, the rider can use it to accurately estimate travel time and also review the travel route.
IBM's Smarter Cities® initiative is using Big Data in a number of applications directed at city infrastructure and operations. Location data from cell phones can be used to provide raw material for detecting traffic patterns.
These patterns can then be used to decide on new transportation projects, to change controls, or to redirect traffic in case of an emergency.
Another important application for Big Data Analytics is public safety. The New
York Police Department is using Big Data for crime prevention.

Product Selection, Design, and Engineering

Product automation provides an enormous opportunity to measure customer experience. We take photos digitally and then post them on Facebook, providing an opportunity for face recognition without requiring laborious cycles in digitization. We listen to songs on Pandora, creating an opportunity to measure what we like or dislike or how often we skip a song after listening to the part of it that we like the most. We read books electronically online or on our favorite handheld devices, giving publishers an opportunity to understand what we read, how many times we read it, and which parts we look at. We watch television using a two-way set-top box that can record each channel click and correlate it to analyze whether the channel was switched right before, during, or after a commercial break. Even mechanical products such as automobiles are increasing electronic interactions. We make all of our ordering transactions electronically, giving third parties opportunities to analyze our spending habits by month, by season, by ZIP+4, and by tens of thousands of micro-segments. Usage data can be synthesized to study the quality of customer experience and can be mined for component defects, successes, or extensions. Marketing analysts can identify micro-segmentations using this data. For example, in a wireless company, we isolated problems in the use of cell phones to defective device antenna by analyzing call quality and comparing it across devices.
Products can be test marketed and changed based on feedback. They can also be customized and personalized for every consumer or micro-segment based on their needs. Analytics plays a major role in customizing, personalizing, and changing products based on customer feedback. Product engineering combines a set of independent components into a product in response to a customer need.

Component quality impacts overall product performance. Can we use analytics to isolate poorly performing components and replace them with good ones? In addition, can we simplify the overall product by removing components that are rarely used and offer no real value to the customer? A lot of product engineering analytics using customer experience data can lead to building simplified products that best meet customer requirements.

To conduct this analysis and predictive modeling, we need a good understanding of the components used and how they participate in the customer experience. Once a good amount of data is collected, the model can be used to isolate badly performing components by isolating the observations from customer experience and tracing them to the poorly performing component. Complex products, such as automobiles, telecommunications networks, and engineering goods, benefit from this type of analytics around product engineering.

The first level of analysis is in identifying a product portfolio mix and its success with the customers. For example, if a marketer has a large number of products, these products can be aligned to customer segments and their usage.

We may find a number of products that were purchased and hardly used, leading to their discontinuation in six months, while other products were heavily used and sparingly discontinued. Once we have identified less-used products, the next analysis question is whether we can isolate the cause of customer disinterest. By analyzing usage patterns, we can differentiate between successful products and unsuccessful ones.

Were the unsuccessful ones never launched? Did many users get stuck with the initial security screen? Maybe the identification process was too cumbersome.

How many users could use the product to perform basic functions offered by the product? What were the highest frequency functions?

The next level of analysis is to understand component failures. How many times did the product fail to perform? Where were the failures most likely? What led to the failure? What did the user do after the failure? Can we isolate the component, replace it, and repair the product online?

These analysis capabilities can now be combined with product changes to create a sophisticated test-marketing framework. We can make changes to the product, try the modified product on a test market, observe the impact, and, after repeated adjustments, offer the altered product to the marketplace.

Let us illustrate how Big Data is shaping improved product engineering and operations at the communications service providers. Major CSPs collect enormous amounts of data about the network, including network transport information coming from the routers and the switches, as well as usage information, popularly known as call detail records (CDRs), which are recorded each time we use telephones to connect with one another. As the CSP networks grew in sophistication, the CDRs were extended to data and video signals using IPDRs. Most CSPs refer to this usage information as xDRs (where *x* is now a variable that can be substituted for "any" usage information). For larger CSPs, the usage statistics not only are high volume (in billions of transactions a day) but also require low-latency analytics for a number of applications. For example, detecting a fraudulent transaction or abusive network user in the middle of a video download or call may be more valuable than finding out this information the next day. In addition, it is always a strategic driver for CSPs to lay out all the network and usage information on their network topology and geography and use a variety of automated analytics and manual visualization techniques to connect the dots between network trouble or inefficiencies and usage.

The analytics provides CSP with a valuable capability to improve the quality of the communication.
If every user call is dropping in a particular area that is a popular location for premier customers, it could lead to churn of those customers to competitors.
The information about xDRs, network events, customer trouble tickets, blogs, and tweets in the social media can be correlated for a variety of business purposes. CSPs have used this analytics to detect spots with poor network performance to reorganize towers and boosters. The differences in usage can be analyzed to detect device problems such as faulty antennas on specific models.
The variations can also be analyzed to find and fix network policies or routing problems. As CSPs race to implement high-volume, low-latency xDR hubs, they are finding plenty of business incentives to fund these programs and reap benefits in the form of improved product offerings to their customers.

Location-Based Services

A variety of industries have location information about their customers. Cell phone operators know customer location through the location of the phones.
Credit-card companies know the location of transactions, and auto manufacturers the location of cars, while social media is trying its best to get customers to disclose their location to their friends and family. On a recent short trip to India,
I decided to use Endomondo, an app on my cell phone to record my jogging activity in Mumbai, India, which was instantly posted on my Facebook page, thereby letting my friends know of my visit to Mumbai. Let us take a wireless CSP example to study how we collect and summarize location information. A cell phone is served by a collection of cell phone towers, and its specific location can be inferred by triangulating its distance from the nearest cell towers. In addition, most smartphones can provide GPS location information that is more accurate (up to about 1 meter). The location data includes longitude and latitude and, if properly stored, could take about 26 bytes of information. If we are dealing with 50 million subscribers and would like to store 24 hours of location information at the frequency of once a minute, the data stored is about 2 terabytes of information per day. This is the amount of information stored in the location servers at a typical CSP.
Customer locations can be summarized into “hang outs” at different levels of granularity. The location information can be aggregated into geohashes that draw geo boundaries and transform latitude-longitude data into geohash so that it can be counted and statistically analyzed. The presence of a person in a specific location for a certain duration is considered a space-time box and can be used to encode the hang out of an individual in a specific business or residential location for a specific time period.
Many of our smartphone apps collect location data, provided a subscriber “opts-in.”15 If a marketer is interested in increasing the traffic to a grocery store that is located in a specific geohash, they can run an effective market ing campaign by analyzing and understanding which neighborhood people are more likely to hang out or shop in that specific grocery store. Instead of blasting a promotion to all neighborhoods, the communication can now be directed to specific neighborhoods, thereby increasing the efficiency of the marketing campaign. This analysis can possibly be conducted using 6-byte location geohash over a span of one hour and finding all the cell phones that have visited the grocery store regularly. A predictive model can compute the

probability of a customer visiting the grocery store based on their past hang out history, and customer residence information can be clustered to identify neighborhoods most likely to visit the shopping center.

Analysis of machine-to-machine transaction data using Big Data technologies is revolutionizing how location-based services can be personalized and offered at low latency. Consider the example of Shopkick, a retail campaign tool that can be downloaded on a smartphone. Shopkick seeks and uses location data to offer campaigns. Once the app is downloaded, Shopkick seeks permission to use current location as recorded by the smartphone. In addition, Shopkick has a database of retailers and their geo-locations. It runs campaigns on behalf of the merchants and collects its revenues from merchants. Shopkick will let me know, for example, that the department store in my neighborhood would like me to visit the store. As a further incentive, Shopkick will deposit shopping points in my account for just visiting the store. As I walk through the store, Shopkick can use my current location in the smartphone to record my presence at the store and award points.

Jeff Jonas provided me tremendous motivation for playing with location data. I used *openpaths.cc*, a site that tracks cell phone location, to track my whereabouts for approximately three months. Watching my movements over these months was like having a video unfold my activities event by event. I could also see how I could improve the accuracy of the location data collected by openpaths with other known information such as street maps. With the help of a business directory, it is easy to find out the number and duration of my trips to Starbucks, Tokyo Joe's, and Sweet Tomato, my three most common eating hang outs. Why would a customer "opt-in"? Device makers, CSPs, and retailers are beginning to offer a number of location-based services, in exchange for location "opt-in." For example, smartphones offer "find my phone" services, which can locate a phone. If the phone is lost, the last known location can be ascertained via a website. In exchange, the CSP or the device manufacturer may seek location data for product or service improvement. These location-based services could also be revenue generating. A CSP may decide to charge for a configuration service that switches a smartphone to silent mode every time the subscriber enters the movie theater and switches back to normal ring tone once the subscriber leaves the movie theater. Prepaid wireless providers are engaging in location-based campaigns targeted at customers who are about to run out of prepaid minutes. These customers are the most likely to churn to a competitor and could easily continue with their current wireless provider if they were to be directed to a store that sells prepaid wireless cards.

These scenarios raise the obvious data privacy concern, which is a hotly debated topic worldwide. We will spend some time in the technical sections talking about data privacy, governance, and how consumer data can be protected and used only as permitted by the customer. As expected, there are many avenues for abuse of customer data, and data privacy must be engrained in the architecture for an effective protection of customer data.

Online Advertising

Television and radio have used advertising as their funding model for decades. As online content distribution becomes popular, advertising has followed the content distribution with increasing volumes and acceptance in the marketplace. The recently concluded Olympics in London provided a testament to the popularity of mobile and other online media distribution channels as compared with television.

Almost half of the Internet video delivered during the Olympics went to mobile phones and tablets. That's a watershed for portable TV. Nearly 28 million people visited *NBCOlympics.com*, eight percent higher as compared with the Beijing Olympics four years ago. Sixty four million video streams were served across all platforms, a 182 percent increase over Beijing. Nearly 6.4 million people used mobile devices.18

Online advertising is also becoming increasingly sophisticated. I discussed the supply chain for digital advertising with a number of specialized players in Section 2.3. The biggest focus is the advertisement bidding managed for a publisher, such as Google, by either a Supply Side Platform (SSP) or Advertising Exchange. Online advertising provides tremendous opportunity for advertising to a micro-segment and also for context-based advertising. How do we deliver these products, and how do they differ from traditional advertising?

The advertiser's main goal is to reach the most receptive online audience in the right context, who will then engage with the displayed ad and eventually take the desired action identified by the type of campaign.19 Big Data provides us with an opportunity to collect myriads of behavioral information. This information can be collated and analyzed to build two sets of insights about the customers, both of which are very relevant to online advertising. First, the micro-segmentation information and associated purchase history described in Section 3.6 allows us to establish buyer patterns for each micro-segment. Second, we can use the context of an online interaction to drive context-specific advertising. For example, for someone searching and shopping for a product, a number of related products can be offered in the advertisements placed on the web page.

Over the past year, I found an opportunity to study these capabilities with the help of Turn Advertising. Turn's Demand Side Platform (DSP) delivers over 500,000 advertisements per second using ad bidding platforms at most major platforms, including Google, Yahoo, and Facebook. A DSP manages online advertising campaigns for a number of advertisers through real-time auctions or bidding. Unlike a direct buy market (e.g., print or television), where the price is decided in advance based on reach and opportunities to see, the real-time Ad

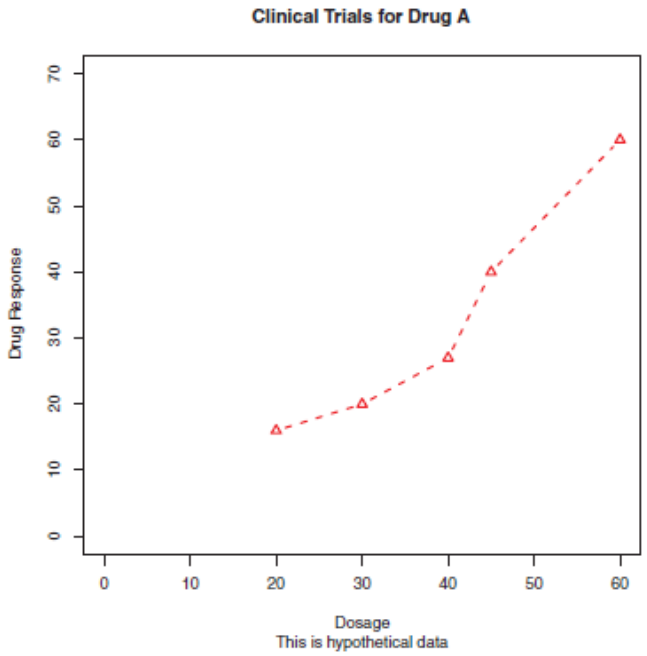
Exchange accept bids for each impression opportunity, and the impression is sold to the highest bidder in a public auction. DSPs are the platforms where all the information about users, pages, ads, and campaign constraints come together to make the best decision for advertisers. Let us consider an example to understand the flow of information and collaboration between publisher, Ad Exchange, DSP, and advertiser to deliver online advertisements. If a user initiates a web search for food in a particular zip code on a search engine, the search engine will take the request, parse it, and start to deliver the search result. While the search results are being delivered, the search engine decides to place a couple of advertisements on the screen. The search engine seeks bids for those spots, which are accumulated via Ad Exchange and offered to a number of DSPs competing for the opportunities to place advertisements for their advertisers. In seeking the bid, the publisher may supply some contextual information that can be matched with any additional information known to the DSP about the user. The DSP decides whether to participate in this specific bid and makes an offer to place an ad. The highest bidder is chosen, and their advertisement is delivered to the user in response to the search. Typically, this entire process may take 80 milliseconds.

A Data Management Platform (DMP) may collect valuable statistics about the advertisement and the advertising process. The key performance indicators (KPIs) include the number of times a user clicked the advertisement, which provides a measure of success. If a user has received a

single advertisement many times, it may cause saturation and reduce the probability that the user will click the advertisement.
As online advertising is integrated with online purchasing, the value of placing an advertisement in the right context may go up. If the placement of the ad results in the immediate purchase of the product, the advertiser is very likely to offer a higher price to the publisher. DSP and DMP success depends directly on their ability to track and match consumers based on their perceived information need and their ability to find advertising opportunities related closely to an online sale of associated goods or services

Adding text, customized axes, and legends

Many high-level plotting functions (for example, plot, hist, boxplot) allow you to include axis and text options, as well as graphical parameters. For example, the following adds a title (main), subtitle (sub), axis labels (xlab, ylab), and axis ranges (xlim, ylim). The results are presented in figure 3.8:




```
plot(dose, drugA, type="b",
col="red", lty=2, pch=2, lwd=2,
main="Clinical Trials for Drug A",
sub="This is hypothetical data",
xlab="Dosage", ylab="Drug Response",
xlim=c(0, 60), ylim=c(0, 70))
```

Again, not all functions allow you to add these options. See the help for the function of interest to see what options are accepted. For finer control and for modularization, you can use the functions described in the remainder of this section to control titles, axes, legends, and text annotations.
NoTe Some high-level plotting functions include default titles and labels. You can remove them by adding ann=FALSE in the plot() statement or in a separate par() statement.

Titles

Use the title() function to add title and axis labels to a plot. The format is

```
title(main="main title", sub="sub-title", xlab="x-axis label", ylab="y-axis label")
```

Graphical parameters (such as text size, font, rotation, and color) can also be specified in the title() function. For example, the following produces a red title and a blue subtitle, and creates green x and y labels that are 25 percent smaller than the default text size:

```
title(main="My Title", col.main="red", sub="My Sub-title", col.sub="blue", xlab="My X label",
ylab="My Y label", col.lab="green", cex.lab=0.75)
```

Axes

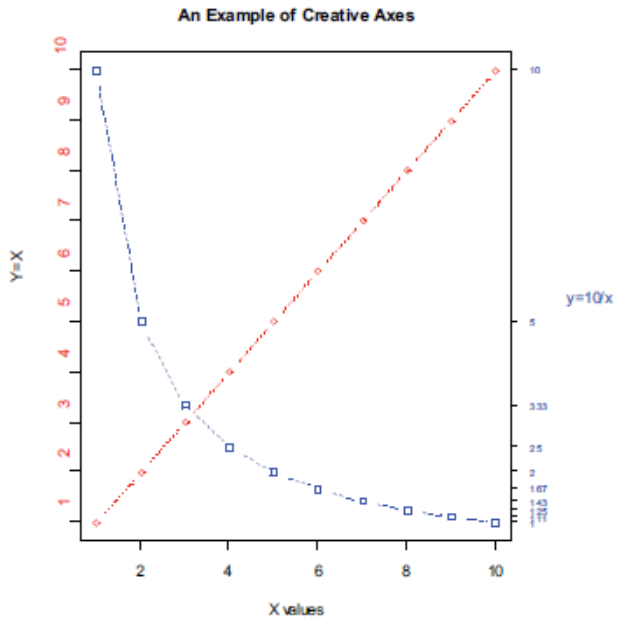
Rather than using R's default axes, you can create custom axes with the axis() function.
The format is

axis(*side*, at=, labels=, pos=, lty=, col=, las=, tck=, ...) where each parameter is described in table 3.7.

Table 3.7 Axis options

Option	Description
side	An integer indicating the side of the graph to draw the axis (1=bottom, 2=left, 3=top, 4=right).
at	A numeric vector indicating where tick marks should be drawn.
labels	A character vector of labels to be placed at the tick marks (if NULL, the at values will be used).
pos	The coordinate at which the axis line is to be drawn (that is, the value on the other axis where it crosses).
lty	Line type.
col	The line and tick mark color.
las	Labels are parallel (=0) or perpendicular (=2) to the axis.
tck	Length of tick mark as a fraction of the plotting region (a negative number is outside the graph, a positive number is inside, 0 suppresses ticks, 1 creates gridlines); the default is –0.01.
(...)	Other graphical parameters.

When creating a custom axis, you should suppress the axis automatically generated by the high-level plotting function. The option axes=FALSE suppresses all axes (including all axis frame lines, unless you add the option frame.plot=TRUE). The options xaxt="n" and yaxt="n" suppress the x- and y-axis, respectively (leaving the frame lines, without ticks). The following listing is a somewhat silly and overblown example that demonstrates each of the features we've discussed so far. The resulting graph is presented in figure 3.9.



At this point, we've covered everything in listing 3.2 except for the line()and the mtext() statements. A plot() statement starts a new graph. By using the line() statement instead, you can add new graph elements to an *existing* graph. You'll use it again when you plot the response of drug A and drug B on the same graph in section 3.4.4. The mtext() function is used to add text to the margins of the plot. The mtext()function is covered in section 3.4.5, and the line() function is covered more fully in chapter 11.

MiNoR TiCk MARks

Notice that each of the graphs you've created so far have major tick marks but not minor tick marks. To create minor tick marks, you'll need the minor.tick() function in the Hmisc package. If you don't already have Hmisc installed, be sure to install it first (see chapter 1, section 1.4.2). You can add minor tick marks with the code

library(Hmisc)

minor.tick(nx=*n*, ny=*n*, tick.ratio=*n*) where nx and ny specify the number of intervals in which to divide the area between major tick marks on the x-axis and y-axis, respectively. tick.ratio is the size of the minor tick mark relative to the major tick mark. The current length of the major tick mark can be retrieved using par("tck"). For example, the following statement will add one tick mark between each major tick mark on the x-axis and two tick marks between

Listing 3.2 An example of custom axes

```
x <- c(1:10)                          <-- Specify data
y <- x
z <- 10/x

opar <- par(no.readonly=TRUE)

par(mar=c(5, 4, 4, 8) + 0.1)          <-- Increase margins

plot(x, y, type="b",                  <-- Plot x versus y
     pch=21, col="red",
     yaxt="n", lty=3, ann=FALSE)

lines(x, z, type="b", pch=22, col="blue", lty=2)   <-- Add x versus 1/x line

axis(2, at=x, labels=x, col.axis="red", las=2)     <-- Draw your axes

axis(4, at=z, labels=round(z, digits=2),
     col.axis="blue", las=2, cex.axis=0.7, tck=-.01)

mtext("y=1/x", side=4, line=3, cex.lab=1, las=2, col="blue")   <-- Add titles and text

title("An Example of Creative Axes",
      xlab="X values",
      ylab="Y=X")

par(opar)
```

each major tick mark on the y-axis:
minor.tick(nx=2, ny=3, tick.ratio=0.5)
The length of the tick marks will be 50 percent as long as the major tick marks. An example of minor tick marks is given in the next section (listing 3.3 and figure 3.10).

Reference lines

The abline() function is used to add reference lines to our graph. The format is

abline(h=*yvalues*, v=*xvalues*)

Other graphical parameters (such as line type, color, and width) can also be specified in the abline() function. For example:
abline(h=c(1,5,7)) adds solid horizontal lines at y = 1, 5, and 7, whereas the code
abline(v=seq(1, 10, 2), lty=2, col="blue") adds dashed blue vertical lines at x = 1, 3, 5, 7, and 9.
Listing 3.3 creates a reference line for our drug example at y = 30. The resulting graph is displayed in figure 3.10.

Legend

When more than one set of data or group is incorporated into a graph, a legend can help you to identify what's being represented by each bar, pie slice, or line. A legend can be added (not surprisingly) with the legend() function. The format is legend(*location*, *title*, *legend*, ...)
The common options are described in table 3.8.
Other common legend options include bty for box type, bg for background color, cex for size, and text.col for text color.

Table 3.8 Legend options

Option	Description
location	There are several ways to indicate the location of the legend. You can give an x,y coordinate for the upper-left corner of the legend. You can use `locator(1)`, in which case you use the mouse to indicate the location of the legend. You can also use the keywords `bottom`, `bottomleft`, `left`, `topleft`, `top`, `topright`, `right`, `bottomright`, or `center` to place the legend in the graph. If you use one of these keywords, you can also use `inset=` to specify an amount to move the legend into the graph (as fraction of plot region).
title	A character string for the legend title (optional).
legend	A character vector with the labels.
...	Other options. If the legend labels colored lines, specify `col=` and a vector of colors. If the legend labels point symbols, specify `pch=` and a vector of point symbols. If the legend labels line width or line style, use `lwd=` or `lty=` and a vector of widths or styles. To create colored boxes for the legend (common in bar, box, or pie charts), use `fill=` and a vector of colors.

Specifying horiz=TRUE sets the legend horizontally rather than vertically. For more on legends, see help(legend). The examples in the help file are particularly informative.
Let's take a look at an example using our drug data (listing 3.3). Again, you'll use a number of the features that we've covered up to this point. The resulting graph is presented in figure 3.10.

Listing 3.3 Comparing Drug A and Drug B response by dose

```
dose  <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)

opar <- par(no.readonly=TRUE)

par(lwd=2, cex=1.5, font.lab=2)

plot(dose, drugA, type="b",
     pch=15, lty=1, col="red", ylim=c(0, 60),
     main="Drug A vs. Drug B",
     xlab="Drug Dosage", ylab="Drug Response")

lines(dose, drugB, type="b",
      pch=17, lty=2, col="blue")

abline(h=c(30), lwd=1.5, lty=2, col="gray")

library(Hmisc)
minor.tick(nx=3, ny=3, tick.ratio=0.5)

legend("topleft", inset=.05, title="Drug Type", c("A","B"),
       lty=c(1, 2), pch=c(15, 17), col=c("red", "blue"))

par(opar)
```

Increase line, text, symbol, label size

Generate graph

Add minor tick marks

Add legend

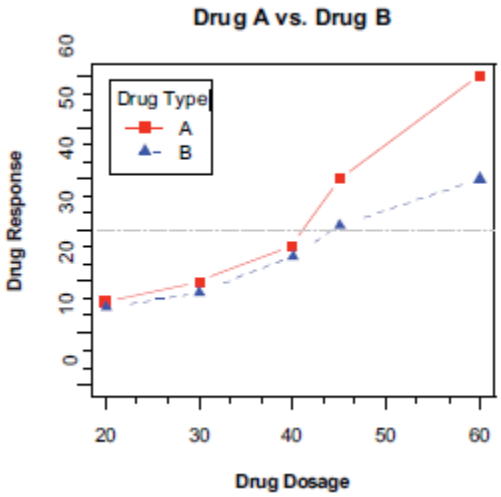


FIGURE 3.10

Almost all aspects of the graph in figure 3.10 can be modified using the options discussed in this chapter. Additionally, there are many ways to specify the options desired.
The final annotation to consider is the addition of text to the plot itself. This topic is covered in the next section.

Text annotations

Text can be added to graphs using the text() and mtext() functions. text() places text within the graph whereas mtext() places text in one of the four margins. The formats are
text(*location*, "*text to place*", *pos*, ...)
mtext("*text to place*", *side*, line=*n*, ...) and the common options are described in table 3.9.

Table 3.9 Options for the `text()` and `mtext()` functions

Option	Description
`location`	Location can be an x,y coordinate. Alternatively, the text can be placed interactively via mouse by specifying location as `locator(1)`.
`pos`	Position relative to location. 1 = below, 2 = left, 3 = above, 4 = right. If you specify `pos`, you can specify `offset=` in percent of character width.
`side`	Which margin to place text in, where 1 = bottom, 2 = left, 3 = top, 4 = right. You can specify `line=` to indicate the line in the margin starting with 0 (closest to the plot area) and moving out. You can also specify `adj=0` for left/bottom alignment or `adj=1` for top/right alignment.

Other common options are cex, col, and font (for size, color, and font style, respectively). The text() function is typically used for labeling points as well as for adding other text annotations. Specify location as a set of x, y coordinates and specify the text to place as a

vector of labels. The x, y, and label vectors should all be the same length. An example is given next and the resulting graph is shown in figure 3.11.

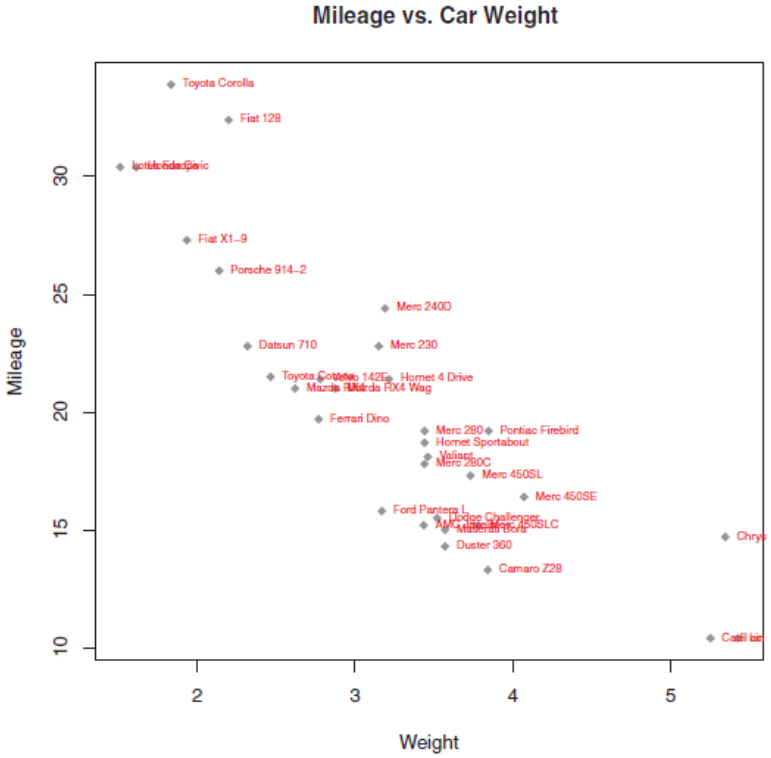


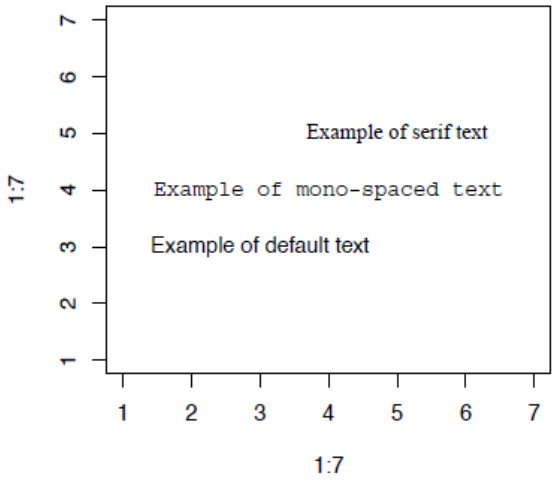
Figure 3.11 Example of a scatter plot (car weight vs. mileage) with labeled points (car make)

```
attach(mtcars)
plot(wt, mpg,
main="Mileage vs. Car Weight",
xlab="Weight", ylab="Mileage",
pch=18, col="blue")
text(wt, mpg,
row.names(mtcars),
cex=0.6, pos=4, col="red")
detach(mtcars)
```

Here we've plotted car mileage versus car weight for the 32 automobile makes provided in the mtcars data frame. The text() function is used to add the car makes to the right of each data point. The point labels are shrunk by 40 percent and presented in red. As a second example, the following code can be used to display font families:

```
opar <- par(no.readonly=TRUE)
par(cex=1.5)
plot(1:7,1:7,type="n")
text(3,3,"Example of default text")
text(4,4,family="mono","Example of mono-spaced text")
text(5,5,family="serif","Example of serif text")
par(opar)
```

The results, produced on a Windows platform, are shown in figure 3.12. Here the par() function was used to increase the font size to produce a better display. The resulting plot will differ from platform to platform, because plain, mono, and serif text are mapped to different font families on different systems. What does it look like on yours?



MATh ANNoTATioNs

Finally, you can add mathematical symbols and formulas to a graph using TEX-like rules.

See help(plotmath) for details and examples. You can also try demo(plotmath) to see this in action. A portion of the results is presented in figure 3.13. The plotmath() function can be used to add mathematical symbols to titles, axis labels, or text annotation in the body or margins of the graph.

You can often gain greater insight into your data by comparing several graphs at one time. So, we'll end this chapter by looking at ways to combine more than one graph into a single image.

Arithmetic Operators		Radicals	
x + y	x + y	sqrt(x)	$\sqrt{x}$
x – y	x – y	sqrt(x, y)	$\sqrt[y]{x}$
x * y	xy	Relations	
x/y	x/y	x == y	x = y
x %+–% y	x ± y	x != y	x ≠ y
x%/%y	x √ y	x < y	x < y
x %*% y	x × y	x <= y	x ″ y
x %.% y	x · y	x > y	x > y
–x	–x	x >= y	x ≥ y
+x	+x	x %~~% y	x ⊕ y
Sub/Superscripts		x %=~% y	x ≡ y
x[i]	x_i	x %==% y	x ≡ y
x^2	x^2	x %prop% y	x ∝ y
Juxtaposition		Typeface	
x * y	xy	plain(x)	x
paste(x, y, z)	xyz	italic(x)	*x*
Lists		bold(x)	**x**
list(x, y, z)	x, y, z	bolditalic(x)	***x***
		underline(x)	x

Figure 3.13 **Partial results from `demo(plotmath)`**

Combining graphs

R makes it easy to combine several graphs into one overall graph, using either the par() or layout() function. At this point, don't worry about the specific types of graphs being combined; our focus here is on the general methods used to combine them. The creation and interpretation of each graph type is covered in later chapters. With the par() function, you can include the graphical parameter mfrow=c(*nrows, ncols*) to create a matrix of *nrows* x *ncols* plots that are filled in by row. Alternatively, you can use mfcol=c(*nrows, ncols*) to fill the matrix by columns.

For example, the following code creates four plots and arranges them into two rows and two columns:

```
attach(mtcars)
opar <- par(no.readonly=TRUE)
par(mfrow=c(2,2))
plot(wt,mpg, main="Scatterplot of wt vs. mpg")
plot(wt,disp, main="Scatterplot of wt vs disp")
hist(wt, main="Histogram of wt") boxplot(wt, main="Boxplot of wt")
par(opar)
detach(mtcars)
```

The results are presented in figure 3.14.

As a second example, let's arrange 3 plots in 3 rows and 1 column. Here's the code:

```
attach(mtcars)
opar <- par(no.readonly=TRUE)
par(mfrow=c(3,1))

hist(wt)
hist(mpg)
hist(disp)
par(opar)
detach(mtcars)
```

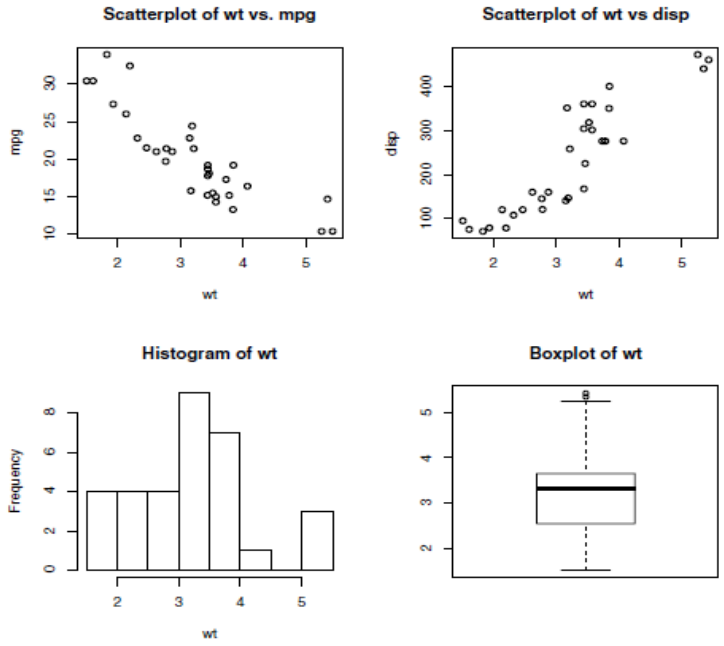


Figure 3.14 **Graph combining four figures through `par(mfrow=c(2,2))`**

The graph is displayed in figure 3.15. Note that the high-level function hist()includesa default title (use main="" to suppress it, or ann=FALSE to suppress all titles and labels).

The layout() function has the form layout(*mat*) where *mat* is a matrix object specifying the location of the multiple plots to combine. In the following code, one figure is placed in row 1 and two figures are placed in row 2:

```
attach(mtcars)
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))
hist(wt)
hist(mpg)
hist(disp)
detach(mtcars)
```

The resulting graph is presented in figure 3.16.

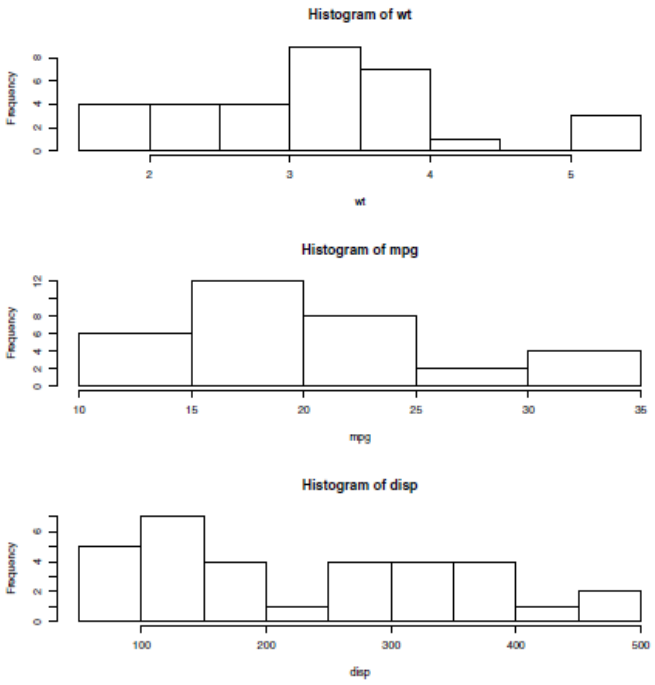


Figure 3.15 Graph combining with three figures through `par(mfrow=c(3,1))`

Optionally, you can include widths= and heights= options in the layout() function to control the size of each figure more precisely. These options have the form widths = a vector of values for the widths of columns heights = a vector of values for the heights of rows. Relative widths are specified with numeric values. Absolute widths (in centimeters) are specified with the lcm() function.

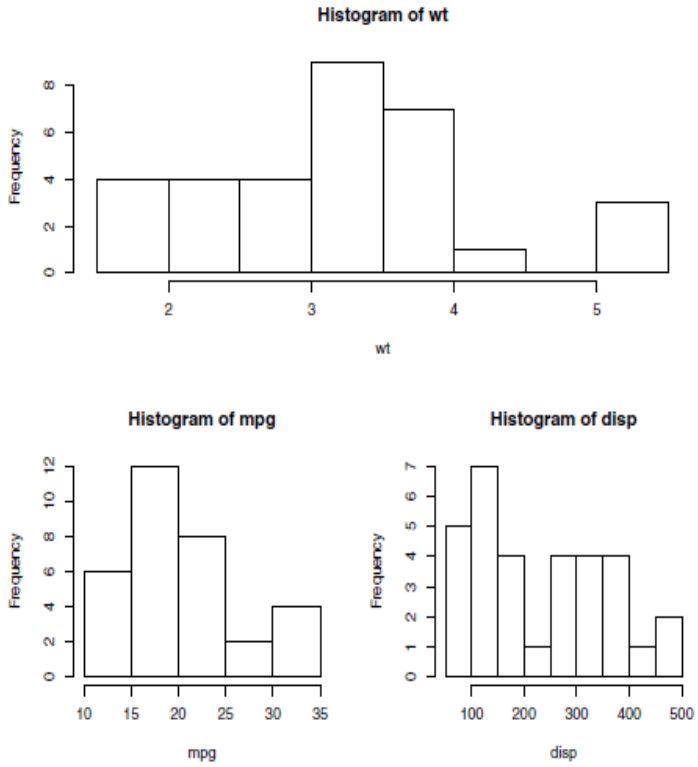


Figure 3.16 Graph combining three figures using the `layout()` function with default widths

In the following code, one figure is again placed in row 1 and two figures are placed in row 2. But the figure in row 1 is one-third the height of the figures in row 2. Additionally, the figure in the bottom-right cell is one-fourth the width of the figure in the bottom-left cell:

```
attach(mtcars)
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE),
widths=c(3, 1), heights=c(1, 2))
hist(wt)
hist(mpg)
hist(disp)
detach(mtcars)
```

The graph is presented in figure 3.17.

As you can see, the layout() function gives you easy control over both the number and placement of graphs in a final image and the relative sizes of these graphs. See help(layout) for more details.

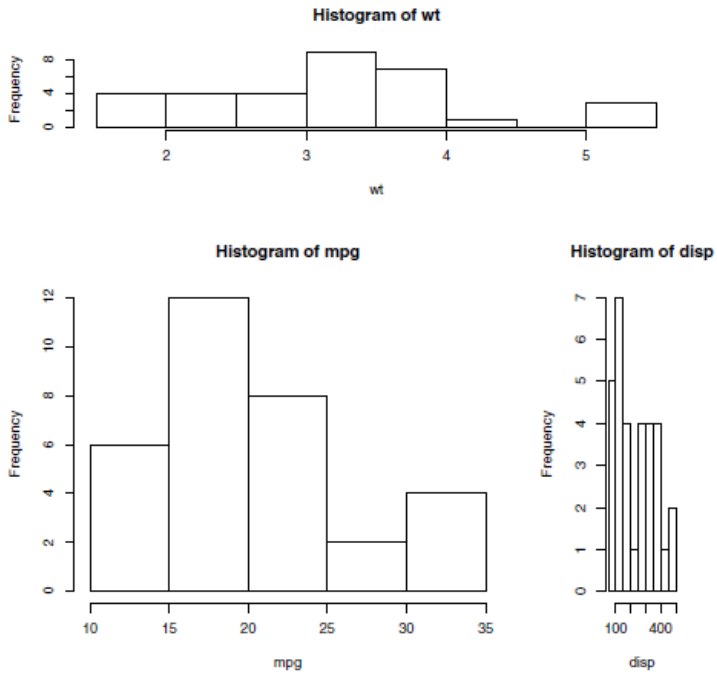


Figure 3.17 **Graph combining three figures using the `layout()` function with specified widths**

fgdgdf