Descriptive Statistics

This chapter reviews essential univariate and bivariate analysis concepts that underpin the more complex statistical methods in subsequent chapters of this book. Univariate and bivariate analyses can be either descriptive or inferential; this chapter will cover descriptive techniques while chapter "Statistical Inference" will cover inferential methods.

Descriptive statistics are rudimentary analysis techniques that help describe and summarize a variable's data in a meaningful way. Descriptive statistics do not allow us to draw any conclusions beyond the available data but are helpful in interpreting the data at hand.

Univariate Analysis

Univariate analysis is the simplest form of statistical analysis, which explores each variable independently.

There are two categories of univariate analyses: (a) **measures of central tendency** describe the central position in a set of data, and (b) **measures of spread** describe how dispersed the data are.

Measures of Central Tendency

Mean

Perhaps the most intuitive measure of central tendency is the **mean**, which is often referred to as the average. The mean of a sample is denoted by $\bar{x}$ and is defined by:

C. Starbuck, *The Fundamentals of People Analytics*,
https://doi.org/10.1007/978-3-031-28674-2_7

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

The population mean is denoted by μ and is defined by:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{N}$$

The mean of a set of numeric values can be calculated using the `mean()` function in R:

```
# Fill vector x with integers
x <- c(1, 2, 3, 3, 100, 200, 300)

# Calculate average of vector x
mean(x)
```

```
## [1] 87
```

Median

The **median** represents the midpoint in a sorted vector of numbers. For vectors with an even number of values, the median is the average of the middle two numbers; it is simply the middle number for vectors with an odd number of values. When the distribution of data is skewed or there is an extreme value, the median *may* be a better measure of central tendency.

The `median()` function in R can be used to handle the sorting and midpoint selection:

```
# Calculate median of vector x
median(x)
```

```
## [1] 3
```

In this example, the median is only 3 while the mean is $\bar{x} = 87$. Large deltas between mean and median values provide important information about the distribution of data.

Here, a single value has significant leverage on these measures of central tendency. To demonstrate, let us eliminate one instance of 3 from the vector and recalculate the mean and median:

```
# Fill vector x1 with integers
x1 <- c(1, 2, 3, 100, 200, 300)
```

```
# Calculate mean of vector x1
mean(x1)
```

```
## [1] 101
```

```
# Calculate median of vector x1
median(x1)
```

```
## [1] 51.5
```

By removing a single value from this vector, the mean increased from $\bar{x} = 87$ to $\bar{x} = 101$ and the median from 3 to 51.5!

Note that differences in mean and median values for x and x1 are *not* due to an extreme value (outlier), as 3 is similar to half of the values in the vector. However, in some cases extreme values may be the cause of large discrepancies between mean and median values since the mean can be sensitive to extreme values. Consider the following set of values:

```
# Fill vector x2 with integers
x2 <- c(1, 2, 3, 4, 5, 1000)
```

```
# Calculate mean of vector x2
mean(x2)
```

```
## [1] 169.1667
```

```
# Calculate median of vector x2
median(x2)
```

```
## [1] 3.5
```

In this case, the value of 1000 has a significant influence on the mean ($\bar{x} = 169.2$) but the median of 3.5 is representative of the middle of values in this vector.

The reality is that both the mean and median can be misleading—and even inappropriate. It is important to understand how the data are distributed around these centers. It would not be too useful to calculate median organization tenure, for example, for a hyper-growth company that has hired the majority of its workforce in the past few months; long-tenured employees would be lost in this metric.

The larger the n-count, the less influential an extreme value will be on $\bar{x}$. As we will learn in chapter "Statistical Inference", sample size is fundamental to our ability to achieve precise estimates of population parameters based on sample statistics.

While the focus of this section is central tendency, it is important to recognize that outlying values are often the more actionable data points in an analysis since these cases may represent those with significantly different experiences relative to the average employee. Understanding the *distribution* of data is critical, and the spread of data around measures of central tendency will receive considerable attention throughout this book.

Mode

The **mode** is the most frequent number in a set of values.

While `mean()` and `median()` are standard functions in R, `mode()` returns the internal storage mode of the object rather than the statistical mode of the data. We can easily create a function to return the statistical mode(s):

```
# Fill vector x2 with integers
x3 <- c(1, 2, 3, 3, 100, 200, 300, 300)

# Create function to calculate statistical mode(s)
stat.mode <- function(x) {
  ux <- unique(x)
  tab <- tabulate(match(x, ux))
  ux[tab == max(tab)]
}

# Return mode(s) of vector x3
stat.mode(x3)
```

```
## [1]   3 300
```

In this case, we have a bimodal distribution since both `3` and `300` occur most frequently.

Range

The **range** is the difference between the maximum and minimum values in a set of numbers.

The `range()` function in R returns the minimum and maximum numbers:

```
# Return lowest and highest values of vector x
range(x)
```

```
## [1]   1 300
```

We can leverage the `max()` and `min()` functions to calculate the difference between these values:

```
# Calculate range of vector x
max(x, na.rm = TRUE) - min(x, na.rm = TRUE)
```

```
## [1] 299
```

In people analytics, there are many conventional descriptive metrics—largely counts, percentages, and averages cut by various time (e.g., day, month, quarter, year) and categorical (e.g., department, job, location, tenure band) dimensions. Here is a sample of common measures:

- Time to Fill: average days between job requisition posting and offer acceptance
- Offer Acceptance Rate: percent of offers extended to candidates that are accepted
- Pass-Through Rate: percent of candidates in a particular stage of the recruiting process who passed through to the next stage
- Progress to Goal: percent of approved positions that have been filled
- cNPS/eNPS: candidate and employee NPS (−100 to 100)
- Headcount: counts and percent of workforce across worker types (employee, intern, contingent)
- Diversity: counts and percent of workforce across gender, ethnicity, and generational cohorts
- Positions: count and percent of open, committed, and filled seats
- Hires: counts and rates
- Career Moves: counts and rates
- Turnover: counts and rates (usually terms/average headcount over the period)
- Workforce Growth: net changes over time, accounting for hires, internal transfers, and exits
- Span of Control: ratio of people leaders to individual contributors
- Layers/Tiers: average and median number of layers removed from CEO
- Engagement: average score or top-box favorability score

Measures of Spread

Variance

Variance is a measure of variability in the data. Variance is calculated using the average of squared differences—or deviations—from the mean.

Variance of a population is defined by:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}$$

Variance of a sample is defined by:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

It is important to note that since differences are squared, the variance is always non-negative. In addition, we cannot compare these squared differences to the arithmetic mean since the units are different. For example, if we calculate the variance of annual compensation measured in USD, variance should be expressed as USD^2 while the mean exists in the original USD unit of measurement.

In R, the sample variance can be calculated using the `var()` function:

```
# Load library
library(peopleanalytics)

# Load data
data("employees")

# Calculate sample variance for annual compensation
var(employees$annual_comp)
```

```
## [1] 1788038934
```

Sample statistics are the default in R. Since the population variance differs from the sample variance by a factor of $s^2(\frac{n-1}{n})$, it is simple to convert output from `var()` to the population variance:

```
# Store number of observations
n = length(employees$annual_comp)

# Calculate population variance for annual compensation
var(employees$annual_comp) * (n - 1) / n
```

```
## [1] 1786822581
```

Standard Deviation

The **standard deviation** is simply the square root of the variance.

The standard deviation of a population is defined by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}}$$

The standard deviation of a sample is defined by:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

Since a squared value can be converted back to its original units by taking its square root, the standard deviation expresses variability around the mean in the variable's original units.

In R, the sample standard deviation can be calculated using the `sd()` function:

```
# Calculate sample standard deviation for annual compensation
sd(employees$annual_comp)
```

```
## [1] 42285.21
```

Since the population standard deviation differs from the sample standard deviation by a factor of $s\sqrt{\frac{n-1}{n}}$, it is simple to convert output from `sd()` to the population standard deviation:

```
# Calculate population standard deviation for annual
↪  compensation
sd(employees$annual_comp) * sqrt((n - 1) / n)
```

```
## [1] 42270.82
```

Quartiles

A **quartile** is a type of quantile that partitions data into four equally sized parts after ordering the data. Each quartile is equally sized with respect to the number of data points—not the range of values in each. Quartiles are also related to **percentiles**. For example, Q1 is the 25th percentile—the value at or below which 25% of values

lie. Percentiles are likely more familiar than quartiles, as percentiles show up in the height and weight measurements of babies, performance on standardized tests like the SAT and GRE, among other things.

The **Interquartile Range (IQR)** represents the difference between Q3 and Q1 cut point values (the middle two quartiles). The IQR is sometimes used to detect extreme values in a distribution; values less than $Q1 - 1.5 * IQR$ or greater than $Q3 + 1.5 * IQR$ are generally considered outliers.

In R, the `quantile()` function returns the values that bookend each quartile:

```
# Return quartiles for annual compensation
quantile(employees$annual_comp)
```

```
##     0%    25%    50%    75%   100%
##  62400  99840 137280 174200 208000
```

Based on this output, we know that 25% of people in our data earn annual compensation of 99,840 USD or less, 137,280 USD is the median annual compensation, and 75% of people earn annual compensation of 174,200 USD or less.

We can also return a specific percentile value using the `probs` argument in the `quantile()` function. For example, if we want to know the 80th percentile annual compensation value, we can execute the following:

```
# Return 80th percentile annual compensation value
quantile(employees$annual_comp, probs = .8)
```

```
##    80%
## 180960
```

In addition, the `summary()` function returns several common descriptive statistics for an object:

```
# Return common descriptives
summary(employees$annual_comp)
```

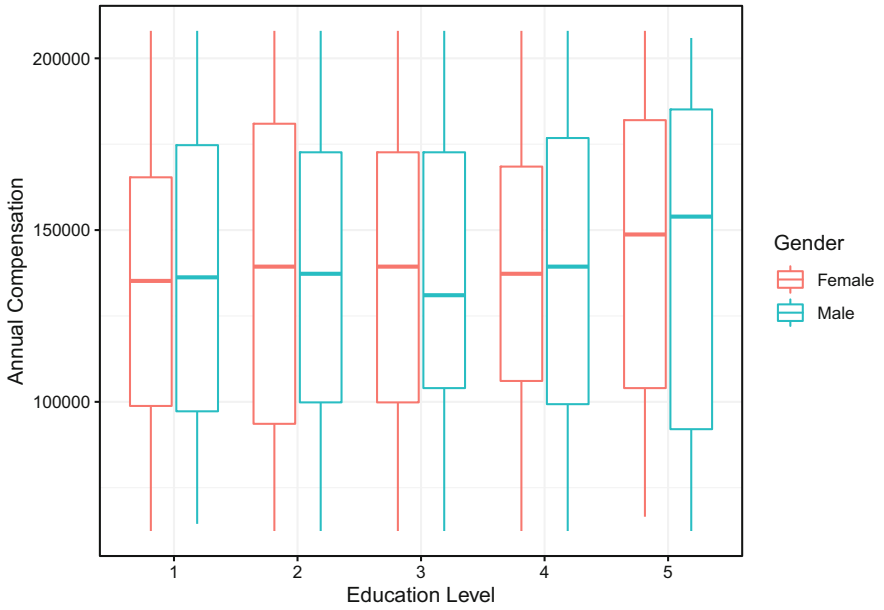
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   62400   99840  137280  137054  174200  208000
```

Box plots are a common way to visualize the distribution of data. Box plots are not usually found in presentations to stakeholders, since they are a bit more technical and often require explanation, but these are very useful to analysts for understanding data distributions during the EDA phase.

Let us visualize the spread of annual compensation by education level and gender using the `geom_boxplot()` function from the `ggplot2` library:

```
# Load library
library(ggplot2)

# Produce box plots to visualize compensation distribution by
↪  education level and gender
ggplot2::ggplot(employees, aes(x = as.factor(ed_lvl), y =
↪  annual_comp, color = gender)) +
ggplot2::geom_boxplot() +
ggplot2::labs(x = "Education Level", y = "Annual
↪  Compensation") +
ggplot2::guides(col = guide_legend("Gender")) +
ggplot2::theme_bw()
```



Box plots can be interpreted as follows:

- Horizontal lines represent median compensation values.
- The box in the middle of each distribution represents the IQR.
- The end of the line above the IQR represents the threshold for outliers in the upper range: $Q3 + 1.5 * IQR$.
- The end of the line below the IQR represents the threshold for outliers in the lower range: $Q1 - 1.5 * IQR$.
- Data points represent outliers: $x > Q3 + 1.5 * IQR$ or $x < Q1 - 1.5 * IQR$.

While box plots are pervasive in statistically oriented disciplines, they can be misleading. Figure 1 illustrates how information about the shape of a distribution

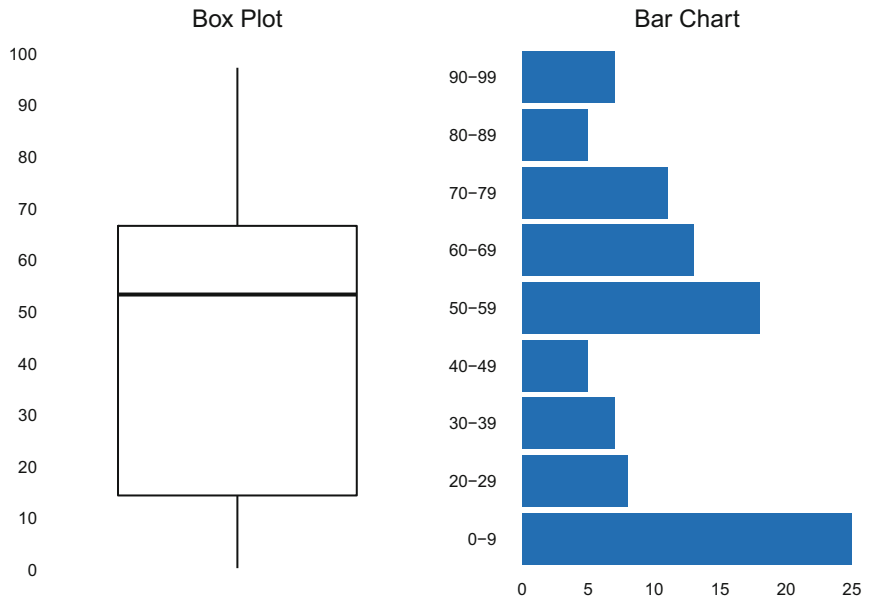


Fig. 1 The number range with the highest frequency (0–9) is not as apparent with a box plot (left) relative to the bar chart (right)

can be lost on a box plot. The range with the highest frequency (0–9) is not as obvious in the box plot relative to the bar chart.

Box plot alternatives such as **violin plots**, **jittered strip plots**, and **raincloud plots** are often more helpful in understanding data distributions. Figure 2 shows the juxtaposition of a raincloud plot against a box plot. While it may seem like an oxymoron, in this case the spread of data is clearer in the rain.

Skewness

Skewness is a measure of the horizontal distance between the mode and mean—a representation of symmetric distortion. In most practical settings, data are not normally distributed. That is, the data are skewed either positively (right-tailed distribution) or negatively (left-tailed distribution). The coefficient of skewness is one of many ways in which we can ascertain the degree of skew in the data. The skewness of sample data is defined as:

$$Sk = \frac{1}{n} \frac{\sum_{i=1}^{n} (x_i - \bar{x})^3}{s^3}$$

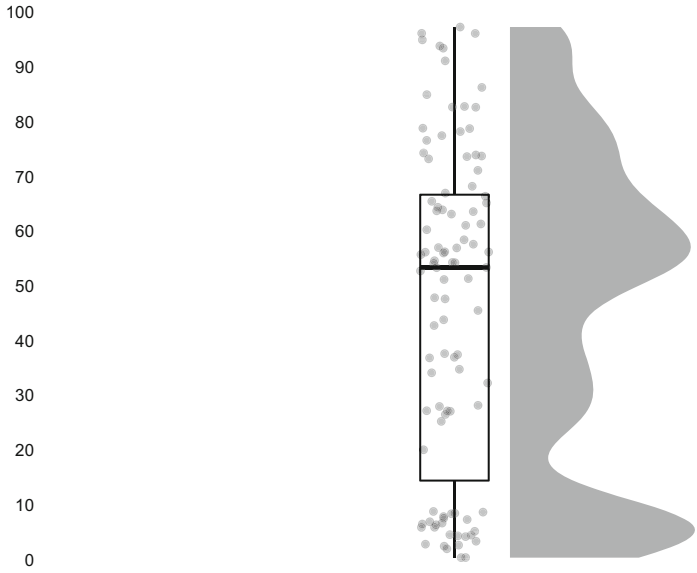


Fig. 2 Raincloud plot superimposed on a box plot to illustrate the data distribution

A positive skewness coefficient indicates positive skew, while a negative coefficient indicates negative skew. The order of descriptive statistics can also be leveraged to ascertain the direction of skew in the data:

- Positive skewness: mode < median < mean
- Negative skewness: mode > median > mean
- Symmetrical distribution: mode = median = mean

Figure 3 illustrates the placement of these descriptive statistics in each of the three types of distributions. The magnitude of skewness can be determined by measuring the distance between the mode and mean relative to the variable's scale. Alternatively, we can simply evaluate this using the coefficient of skewness:

- If skewness is between -0.5 and 0.5, the data are considered symmetrical.
- If skewness is between -0.5 and -1 or 0.5 and 1, the data are moderately skewed.
- If skewness is < -1 or > 1, the data are highly skewed.

Since there is not a base R function for skewness, we can leverage the moments library to calculate skewness:

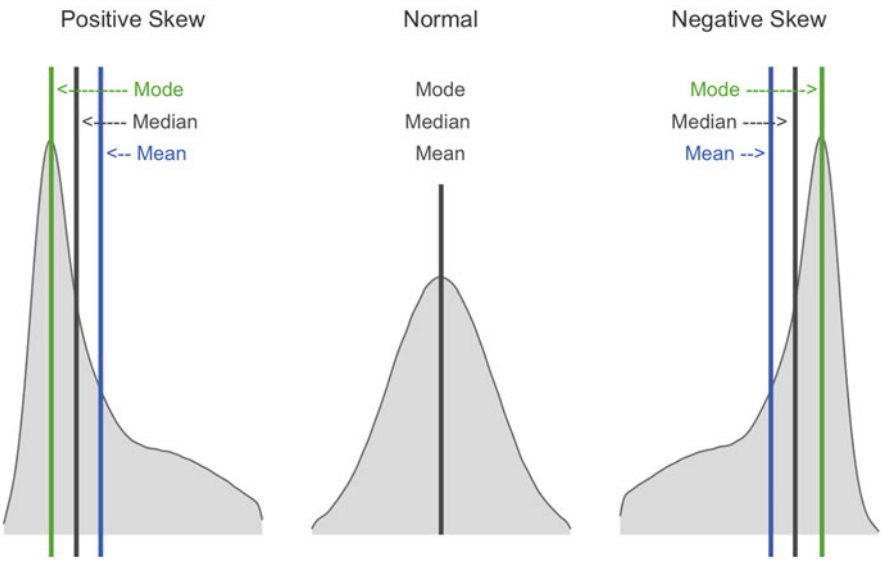


Fig. 3 Skewness

```
# Load library
library(moments)

# Calculate skewness for org tenure, rounded to two
↪  significant figures via the round() function
round(moments::skewness(employees$org_tenure), 2)
```

```
## [1] 2.27
```

Statistical Moments, after which this library was named, play an important role in specifying the appropriate probability distribution for a set of data. Moments are a set of statistical parameters used to describe the characteristics of a distribution. Skewness is the third statistical moment in the set; hence the sum of cubed differences and cubic polynomial in the denominator of the formula above. The complete set of moments comprises: (1) expected value or mean, (2) variance and standard deviation, (3) skewness, and (4) kurtosis.

We can verify that the `skewness()` function from the moments library returns the expected value (per the aforementioned formula) by validating against a manual calculation:

```
# Store components of skewness calculation
n = length(employees$org_tenure)
x = employees$org_tenure
```

```
x_bar = mean(employees$org_tenure)
s = sd(employees$org_tenure)

# Calculate skewness manually, rounded to two significant
↪  figures via the round() function
round(1/n * (sum((x - x_bar)^3) / s^3), 2)
```

```
## [1] 2.27
```

A skewness coefficient of 2.27 indicates that organization tenure is positively skewed. We can visualize the data to confirm the expected right-tailed distribution (Fig. 4):

```
# Produce histogram to visualize sample distribution
ggplot2::ggplot() +
ggplot2::aes(employees$org_tenure) +
ggplot2::labs(x = "Organization Tenure", y = "Density") +
ggplot2::geom_histogram(aes(y = ..density..), fill =
↪  "#414141") +
ggplot2::geom_density(fill = "#ADD8E6", alpha = 0.6) +
ggplot2::theme_bw()
```

Kurtosis

While skewness provides information on the symmetry of a distribution, **kurtosis** provides information on the heaviness of a distribution's tails ("tailedness"). Kurtosis is the fourth statistical moment, defined by:

$$K = \frac{1}{n} \frac{\sum_{i=1}^{n} (x_i - \bar{x})^4}{s^4}$$

Note that the quartic functions characteristic of the fourth statistical moment are the only differences from the skewness formula we reviewed in the prior section (which featured cubic functions).

The terms **leptokurtic** and **platykurtic** are often used to describe distributions with light and heavy tails, respectively. "Platy-" in platykurtic is the same root as "platypus," and I have found it helpful to recall the characteristics of the flat platypus when characterizing frequency distributions as platykurtic (wide and flat) vs. its antithesis, leptokurtic (tall and skinny). The normal (or Gaussian) distribution is referred to as a **mesokurtic** distribution in the context of kurtosis.

Figure 5 illustrates the three kurtosis categorizations.

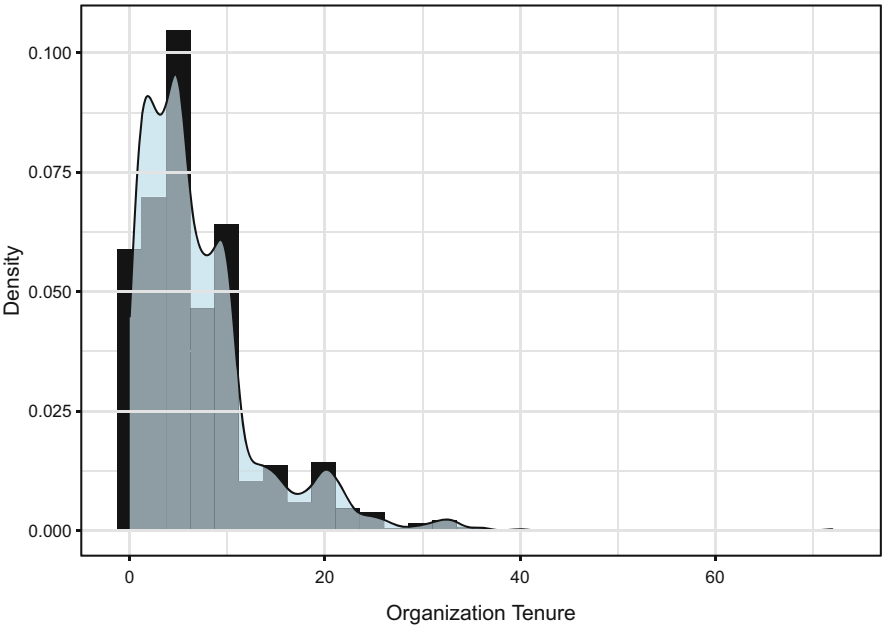
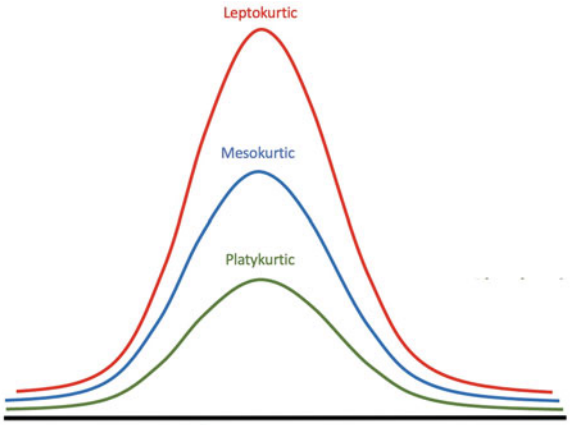


Fig. 4 Organization tenure distribution

Fig. 5 Kurtosis



Kurtosis is measured relative to a normal distribution. Normal distributions have a kurtosis coefficient of 3. Therefore, the kurtosis coefficient is greater than 3 for leptokurtic distributions and less than 3 for platykurtic distributions.

The moments library can also be used to calculate kurtosis in R:

```
# Calculate kurtosis for org tenure, rounded to one
↪  significant figure
round(moments::kurtosis(employees$org_tenure), 1)
```

```
## [1] 13.4
```

We can verify that the `kurtosis()` function returns the expected value (per the aforementioned formula) by validating against a manual calculation:

```
# Calculate kurtosis manually, rounded to one significant
↪  figure
round(1/n * (sum((x - x_bar)^4) / s^4), 1)
```

```
## [1] 13.4
```

Our kurtosis coefficient of 13.4 indicates a leptokurtic distribution which is supported by the visual in Fig. 4.

It is important not to characterize a distribution based on a single isolated metric; we need the complete set of statistical moments to fully understand the distribution of data.

Bivariate Analysis

As we covered, univariate analysis explores a *single* variable. This section will cover **bivariate analysis**, which explores statistical relationships between *two* variables.

Covariance

While variance provides an understanding of how values for a single variable vary, **covariance** is an unstandardized measure of how two variables vary together. Values can range from $-\infty$ to $+\infty$, and these values can be used to understand the direction of the linear relationship between variables. Positive covariance values indicate that the variables vary in the same direction (e.g., tend to increase or decrease together), while negative covariance values indicate that the variables vary in opposite directions (e.g., when one increases, the other decreases, or vice versa).

Covariance of a sample is defined by:

$$cov_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

It is important to note that while covariance aids our understanding of the direction of the relationship between two variables, we cannot use it to understand the strength of the association since it is unstandardized. Due to differences in variables' units of measurement, the strength of the relationship between two variables with large covariance could be weak, while the strength of the relationship between another pair of variables with relatively small covariance could be strong.

In R, we can compute the covariance between a pair of numeric variables by passing the two vectors into the `cov()` function:

```
# Calculate sample covariance between annual compensation and
↪  age using complete observations (missing values will cause
↪  issues if not addressed)
cov(employees$annual_comp, employees$age, use =
↪  "complete.obs")
```

```
## [1] 9381.677
```

In this example, using the default method the covariance between annual compensation and age is 9381.7. The positive value indicates that annual compensation is generally higher for older employees and lower for younger employees.

Just as we multiplied the sample variance by $(n-1)/n$ to obtain the population variance, we can apply the same approach to convert the sample covariance returned by `cov()` to the population covariance:

```
# Calculate population covariance between annual compensation
↪  and age
cov(employees$annual_comp, employees$age, use =
↪  "complete.obs") * (n - 1) / n
```

```
## [1] 9375.295
```

Rather than looking at isolated pairwise relationships, we can produce a covariance matrix to surface pairwise associations among many variables by passing a data frame or matrix object into the `cov()` function:

```
# Generate a covariance matrix among select continuous
↪  variables
cov(subset(employees, select = c("annual_comp", "age",
↪  "org_tenure", "job_tenure", "prior_emplr_cnt",
↪  "commute_dist")), use = "complete.obs")
```

```
##                  annual_comp          age    org_tenure    job_tenure
## annual_comp      1.788039e+09 9381.6772019 -3921.9601469 -3693.1960749
## age              9.381677e+03   83.4550488    17.9255146     7.0467503
## org_tenure      -3.921960e+03   17.9255146    39.7967987    16.9797312
## job_tenure      -3.693196e+03    7.0467503    16.9797312    13.1271220
## prior_emplr_cnt  2.340406e+03    6.8377387    -1.8547177    -0.8213802
## commute_dist     1.067158e+04   -0.1248728     0.7746438     0.5535206
##                 prior_emplr_cnt  commute_dist
## annual_comp        2340.4057552 10671.5790741
## age                   6.8377387    -0.1248728
## org_tenure           -1.8547177     0.7746438
## job_tenure           -0.8213802     0.5535206
## prior_emplr_cnt       6.2400490    -0.5923586
## commute_dist         -0.5923586    65.7212510
```

Using the default Pearson method, the `cov()` function will return sample variances for each variable down the diagonal, since covariance is not applicable in the context of a variable with itself. We can confirm by calculating the variance for age and comparing it to the value at the intersection of the row and column corresponding to age in the matrix:

```
# Return sample variance for age
var(employees$age)
```

```
## [1] 83.45505
```

As expected, the variance for age ($s^2 = 83.5$) matches the value found in the age x age cell of the covariance matrix.

Correlation

Correlation is a scaled form of covariance. While covariance provides an unstandardized measure of the direction of a relationship between variables, correlation provides a standardized measure that can be used to quantify both the direction and strength of bivariate relationships. Correlation coefficients range from -1 to 1, where -1 indicates a perfectly negative association, 1 indicates a perfectly positive association, and 0 indicates the absence of an association. **Pearson's product-moment correlation coefficient** r is defined by:

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

In R, Pearson's r can be calculated using the `cor()` function:

Measurement Scale		Correlation Coefficient
x	*y*	
Continuous	Continuous	Pearson's Product Moment
Continuous	Dichotomous	Point-Biserial
Continuous	Ordinal	Spearman or Kendall Rank
Dichotomous	Dichotomous	Phi, Contingency
Ordinal	Dichotomous	Rank-Biserial
Ordinal	Ordinal	Spearman or Kendall Rank

Fig. 6 Proper applications of correlation coefficients

```
# Calculate the correlation between annual compensation and
↪  age
cor(employees$annual_comp, employees$age, use =
↪  "complete.obs")
```

```
## [1] 0.02428654
```

While we already know that the relationship between annual compensation and age is positive based on the positive covariance coefficient, Pearson's r of 0.02 indicates that the strength of the positive association is weak ($r = 0$ represents the absence of a relationship). Though there are no absolute rules for categorizing the strength of relationships, as thresholds often vary by domain, the following is a general rule of thumb for interpreting the strength of bivariate associations:

- Weak = Absolute value of correlation coefficients between 0 and 0.3
- Moderate = Absolute value of correlation coefficients between 0.4 and 0.6
- Strong = Absolute value of correlation coefficients between 0.7 and 1

There are several correlation coefficients, and the measurement scale of x and y determine the appropriate type (Fig. 6). Pearson's r can be used when both variables are measured on continuous scales or when one is continuous and the other is dichotomous (point-biserial correlation).

When one or both variables are ordinal, we can leverage **Spearman's** ρ or **Kendall's** τ, which are both standardized nonparametric measures of the association between one or two rank-ordered variables. Let us look at Spearman's ρ, which is defined as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Fig. 7 2×2 table for random variables x and y

	$y = 0$	$y = 1$
$x = 0$	A	B
$x = 1$	C	D

We can override the default Pearson method in the `cor()` function to implement a specific form of rank correlation using the `method` argument:

```
# Calculate the correlation between job level and education
↪  level using Spearman's method
cor(employees$job_lvl, employees$ed_lvl, method = "spearman",
↪  use = "complete.obs")
```

```
## [1] 0.1074192
```

The ρ coefficient of 0.11 indicates that the positive association between job level and education level is weak. We could also pass `method = "kendall"` to this `cor()` function to implement Kendall's τ.

The **Phi Coefficient** (ϕ), sometimes referred to as the **mean square contingency coefficient** or **Matthews correlation** in ML, can be used to understand the association between two dichotomous variables.

For the 2×2 table for two random variables x and y depicted in Fig. 7, the ϕ coefficient is defined as:

$$\phi = \frac{(AD - BC)}{\sqrt{(A + B)(C + D)(A + C)(B + D)}}$$

To illustrate, let us examine whether there is a relationship between gender and performance after transforming performance from its ordinal form to a dichotomous variable (high vs. low performance). We can leverage the `psych` library to calculate ϕ in R:

```
# Set females to 1 and everything else to 0
employees$gender_code <- ifelse(employees$gender == 'Female',
↪  1, 0)

# Set stock options to 1 if level > 0
employees$stock_option_code <- ifelse(employees$stock_opt_lvl
↪  > 0, 1, 0)

# Create a 2x2 contingency table
```

```
contingency_tbl <- table(employees$gender_code,
↪  employees$stock_option_code)

# Calculate the Phi Coefficient between dichotomous variables
psych::phi(contingency_tbl)
```

```
## [1] -0.01
```

ϕ is essentially 0, which means stock options are distributed equitably across gender categories (good news!). While there are not differences in the proportion of males and females who receive at least some stock options, examining whether there is equity in the amount of stock grants and refreshes may be a good next step.

A correlation matrix can be produced to surface associations among many variables by passing a data frame or matrix object into the `cor()` function:

```
# Generate a correlation matrix among select continuous
↪  variables
cor(subset(employees, select = c("annual_comp", "age",
↪  "org_tenure", "job_tenure", "prior_emplr_cnt",
↪  "commute_dist")), use = "complete.obs")
```

```
##                annual_comp         age  org_tenure  job_tenure prior_emplr_cnt
## annual_comp      1.00000000  0.02428654 -0.01470248 -0.02410622      0.02215688
## age              0.02428654  1.00000000  0.31104359  0.21290106      0.29963476
## org_tenure      -0.01470248  0.31104359  1.00000000  0.74288567     -0.11769547
## job_tenure      -0.02410622  0.21290106  0.74288567  1.00000000     -0.09075393
## prior_emplr_cnt  0.02215688  0.29963476 -0.11769547 -0.09075393      1.00000000
## commute_dist     0.03113059 -0.00168612  0.01514695  0.01884500     -0.02925080
##                commute_dist
## annual_comp       0.03113059
## age              -0.00168612
## org_tenure        0.01514695
## job_tenure        0.01884500
## prior_emplr_cnt  -0.02925080
## commute_dist      1.00000000
```

Based on this correlation matrix, most pairwise associations are weak with the exception of the relationship between `org_tenure` and `job_tenure` ($r = 0.7$). The values down the diagonal are 1 because these represent the correlation between each variable with itself. You may also notice that the information above and below the diagonal is identical and, therefore, redundant.

A great R library for visualizing correlation matrices is corrplot. Several arguments can be specified for various visual representations of the relationships among variables, as illustrated in Fig. 8.

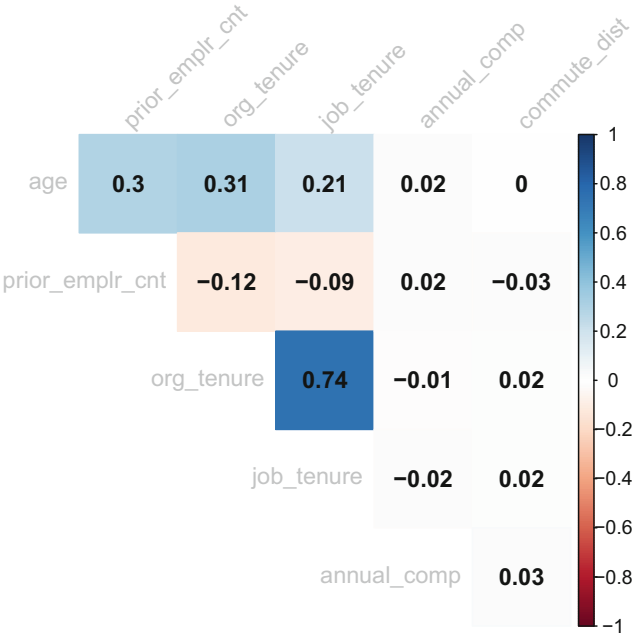


Fig. 8 Corrplot correlation matrix

```
# Store correlation matrix to object M
M <- cor(subset(employees, select = c("annual_comp", "age",
↪  "org_tenure", "job_tenure", "prior_emplr_cnt",
↪  "commute_dist")), use = "complete.obs")

# Visualize correlation matrix
corrplot::corrplot(M, method = "color",
                   type = "upper", order = "hclust", # Apply
                   ↪  hierarchical clustering for ordering
                   ↪  coefficients above the diagonal
                   addCoef.col = "black", # Add correlation
                   ↪  coefficient
                   tl.col = "grey", tl.srt = 45, # Label color
                   ↪  and rotation
                   diag = FALSE # Hide correlation coefficient
                   ↪  on the principal diagonal
                   )
```

The `GGally` library produces a variety of useful information, including correlation coefficients, bivariate scatterplots, and univariate distributions, as illustrate in Fig. 9:

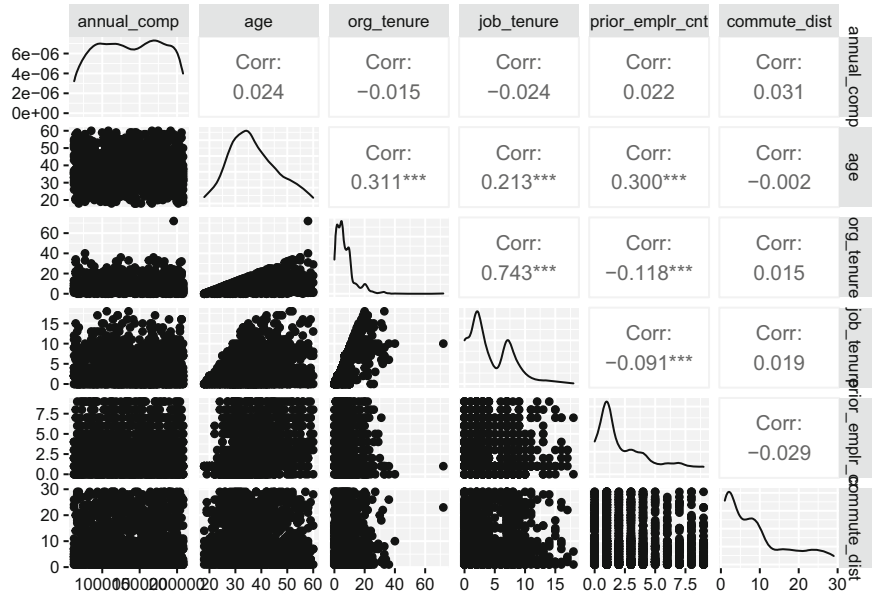


Fig. 9 GGpairs bivariate correlations and data distributions

```
# Visualize correlation matrix
GGally::ggpairs(subset(employees, select = c("annual_comp",
↪  "age", "org_tenure", "job_tenure", "prior_emplr_cnt",
↪  "commute_dist")))
```

We may find that these bivariate associations look quite different for certain business areas or jobs, assuming departments and jobs were created at different points in the company's history. There is often a lot of noise in data at the broader company level, so understanding the nature and nuance of associations is important.

A classic example of this is a statistical phenomenon known as **Simpson's Paradox**, which is particularly common in the social sciences. Simpson's Paradox occurs when a correlation is present in subsets of data but disappears or reverses when the subsets are combined. The prototypical case is a study of gender discrimination at the University of California, Berkeley (Bickel et al., 1975). The overall data indicated that men were more likely than women to gain admission to the university's graduate programs, though there was no evidence of bias in any individual department. Upon closer evaluation, researchers found that women were more likely to apply to departments with lower acceptance rates while men tended to apply to less selective departments. The more nuanced relationships, such as

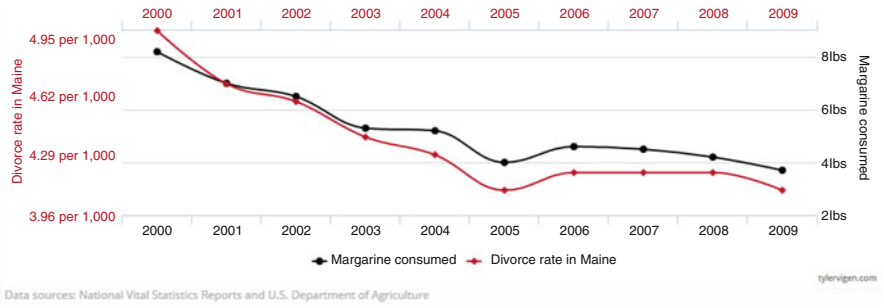


Fig. 10 Correlation between Maine divorce rate and margarine consumption (r = 0.99)

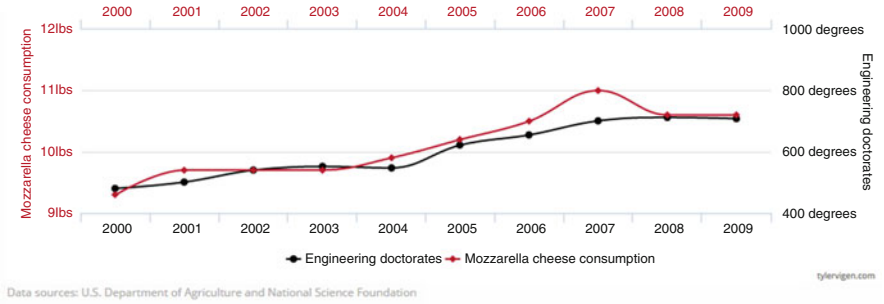


Fig. 11 Correlation between mozzarella cheese consumption and civil engineering doctorate conferrals (r = 0.96)

the association between gender and the partitioning variable (department) in this example, can lead to incorrect conclusions when examining relationships only at the broader population level. We will explore how to control for this in the context of linear regression beginning in chapter "Linear Regression".

Finally, it is important to remember that correlation is not causation. Correlations can be spurious (variables related by chance), and drawing conclusions based on bivariate associations alone—especially in the absence of sound theoretical underpinnings—can be dangerous. Figures 10 and 11 are two examples of nearly perfect correlations between variables for which there is likely no true direct association.

Neither covariance nor correlation alone is sufficient for determining whether an observed association in sample data is also present in the population. For this, we need to graduate from descriptive to inferential statistics.

Review Questions

1. How does the mean and median compare with respect to sensitivity to extreme values (outliers)?
2. What does the standard deviation tell us about the spread of data, and how does it compare to the variance?
3. How does the order of the mean, median, and mode differ between positively and negatively skewed distributions?
4. Do large covariance coefficients always indicate strong bivariate associations? Why or why not?
5. What information is represented in box plots?
6. Do quartiles relate to percentiles?
7. What type of correlation coefficient should be used when evaluating the relationship between a pair of rank-ordered variables?
8. What type of correlation coefficient should be used when evaluating the relationship between a pair of dichotomous variables?
9. How would you characterize the shape of platykurtic, leptokurtic, and mesokurtic distributions?
10. When using the Pearson method, what do the values down the diagonal of a *covariance* matrix represent?