Increasing Profits from Real Estate Leasing : Flexible Strategies based on Market Conditions

By
Cassie Ann Raazi

B.S. Construction Management
Pratt Institute, 2007

Submitted to the System Design and Management Program and the Program in Real Estate Development in conjunction with the Center for Real Estate in partial fulfillment of the requirements for the degrees of

Master of Science in Engineering and Management and
Master of Science in Real Estate Development
at the
Massachusetts Institute of Technology

February 2022

Signature of Author...
Cassie Ann Raazi
System Design and Management
Center for Real Estate
December 1, 2022

Certified by...
Professor Richard de Neufville
Professor of Engineering Systems, Institute for Data, Systems, and Society
Thesis Supervisor

Certified by...
Professor David Geltner
Professor of Real Estate Finance, Department of Urban Studies and Planning
Associate Director of Research, Center for Real Estate
Thesis Supervisor

Accepted by..
Joan S. Rubin
Executive Director
System Design and Management Program

Accepted by..
Professor Siqi Zheng
Samuel Tak Lee Professor of Urban and RE Sustainability
Department of Urban Studies and Planning

Increasing Profits from Real Estate Leasing : Flexible Strategies based on Market Conditions

By

Cassie Ann Raazi

Submitted to the System Design and Management Program and the Program in Real Estate Development in conjunction with the Center for Real Estate in partial fulfillment of the requirements for the degrees of Master of Science in Engineering and Management and Master of Science in Real Estate Development

ABSTRACT

The increasing use of modern data analytics is changing decision making processes in the commercial real estate industry. Advances in data analytics present opportunities for commercial real estate owners and managers to increase profits by integrating market cycles into leasing strategy. This research presents a model that exploits readily available data to simulate market volatility and uncertainty, inform leasing strategy, and make better decisions about lease durations offered. We compare the results of applying three different leasing strategies: consistent 5-year, consistent 10-year, and variable based on understanding of relative positioning within the market cycle. For comparative analysis of these strategies, Monte Carlo simulation via Julia is used to run 10,000 trials for each strategy, calculating the range of outcomes that could occur with each leasing strategy over the life of an asset. It is found that leasing with market knowledge is most optimal of the three strategies examined as it increases profits. The results suggest that incorporating knowledge of relative position within the market cycle to determine optimal lease length creates opportunity for increased profits from leasing. Given the increasing availability of real estate data, future research is directed at exploring different lease duration strategies and the use of real data feeding the simulation to make better models.

Thesis Supervisor: Richard de Neufville
Title: Professor of Engineering Systems

Thesis Supervisor: David Geltner
Title: Professor of Real Estate Finance

ACKNOWLEDGEMENTS

Thank you to Professors Richard de Neufville and David Geltner for being more than my advisors; each going beyond guiding and providing deep insight into the subject matter, but also being great sources of support and encouragement. Richard, your zest for life is admirable, and I hope to engender it in each of my days.

To my husband, Cactus, MIT couldn't have become a reality without you. Thank you for being my greatest source of support and inspiration. To my son, Winter, you were every excuse not to write, and every reason to finish writing.

To all my friends, colleagues, and the greater MIT community, thank you for the experience.

TABLE OF CONTENTS

INTRODUCTION....7

RESEARCH, MOTIVATION, & HYPOTHESIS....9

LITERATURE....16

COMMERCIAL LEASING....20

UNCERTAINTY & THE VALUE OF FLEXIBLITY....25

REAL ESTATE MARKET CYCLE....29

MONTE CARLO SIMULATION APPROACH....33

MODEL IMPLEMENTATION....37

RESULTS & INTERPRETATION....41

CONCLUSION....47

REFERENCES....50

APPENDIX: CODE....52

LIST OF FIGURES

Figure 1: Stereotypical Prevailing Commercial Property Characteristic Lease Terms in the United States (Geltner, Miller, Clayton, & Eiccholtz, 2014) 22

Figure 2: Manhattan Commercial Office Leases (Derived from proprietary data) 23

Figure 3: Long-run Market Cycle 31

Figure 4: Deterministic Model with Addition of Uncertainty via Monte Carlo Simulation 36

Figure 5: Real Estate Cycle Periods & Phases (Jalori, 2017)....................38

Figure 6: Sample Trial Scenario of Lease Executions....................40

Figure 7: Simulation Summary Output Distributions 41

Figure 8: Simulation Cumulative Target Curve....................44

Figure 9: Market Knowledge Strategy Distribution....................45

Figure 10: Configuration File ....**Error! Bookmark not defined.**

Figure 11: Lease Strategies....................**Error! Bookmark not defined.**

LIST OF TABLES

Table 1: Market Knowledge Strategy 39

Table 2: Simulation Summary Statistics...41

Table 3: Simulation Quartile Information 42

INTRODUCTION

Commercial property valuation techniques have yet to incorporate the uncertainty inherent in life today. Traditional valuation techniques, generally limited to the Discounted Cash Flow approach, offer deterministic valuations based on the assumption of a static future.

Given the increasing volatility of the business and economic cycle, with significant boom/bust episodes in this century roughly once a decade, it may make sense to enhance current valuation approaches. At the same time, lessors may benefit from a more dynamic approach to lease term and structure, particularly one which incorporates uncertainty to maximize cash flows (and therefore property valuation) over the economic cycle.

Real estate investors and property managers could benefit from the ability to incorporate uncertainty into their valuation approach. Their goal is to maximize the value of the properties under their supervision. The goal of this research is to demonstrate a path to greater valuation through a more flexible, dynamic approach.

A more dynamic valuation methodology incorporating uncertainty is proposed. In this thesis, thousands of future scenarios are simulated using stochastic analysis. Having established future uncertainty, different leasing options are evaluated to determine whether a flexible approach to lease term results in higher valuations over the expected life of the asset.

This work shows that we can improve decision-making and fundamental understanding in valuation by explicitly incorporating and modeling uncertainty to understand its implications, and potentially managing it to advantage. It is demonstrated that modifications to current leasing approaches, which tend to be rigid in any period and relatively static over the cycle, yield improvements in operating cash flows and, therefore, valuations.

In this thesis, Monte Carlo simulation is used to explore the range of possibilities that could occur by employing different leasing strategies over the life of the asset. Through a sequence of periodic lease agreements modeled over a fifty-year time period, multiple scenarios are played out. The model generates a total of 10,000 trials. Summary statistics suggest that there is opportunity to improve cash flow utilizing a more informed leasing approach.

As property owners, investors, and managers introduce uncertainty into their projections a gradual shift to a more flexible lease terms and structures should be implemented. It is believed that, for the majority of real estate owners, a more flexible leasing approach will lead to improved cash flows and valuations over the cycle.

RESEARCH, MOTIVATION, & HYPOTHESIS

The value of a property today depends on how much cash it can deliver in the future. The income approach is a widely used method to estimate the value of income-producing property (Kelliher & Mahoney, 2000). The income approach is based on the theory that since income-producing property is usually purchased or leased for investment purposes, the present value of all anticipated future cash flows, are the critical factors affecting the property's value. The market value of any long-term asset can be determined by simply calculating the present value of all future income streams associated with the related asset (Williams, 1938). Three critical components must be estimated to create the discounted cash flow (DCF) of the income approach: the amount of future cash flows, the timing of future cash flows, and an appropriate discount rate.

The main method of valuation for income producing real estate is the income/DCF approach. Real estate valuation based on DCF analysis has been well documented and researched (Weber, 1990). DCF is one of the best valuation techniques for estimating real estate value when reliable market comparables are difficult to find (Martin, 1990). The DCF approach involves projecting future years of cash flow and discounting them using a risk-adjusted discount rate to arrive at the present value of the project. DCF valuation estimates the fundamental value of an asset by the cumulative present values of its future cash flows. The difficulties involved with the valuation of any long-term income-producing property include estimating the amount and timing of the future cash flows, the discount rate, the rate of change (increase or decrease) in revenue or expense items, the holding period, and tax rates (Kelliher & Mahoney, 2000).

The standard DCF reflects and utilizes a single stream of future cash flows. The DCF can mislead decision makers, allowing them to ignore or overlook important challenging possibilities and opportunities alike. Each cash flow line item, has a single amount in each future period; in this context, the DCF is treating the world as if the future were deterministic, or has a single possible outcome. In reality, the only way that a DCF can show the outcome deterministically is if we apply the DCF analysis retrospectively, in which case, there is only one realized history and we can know it because it has already occurred.

Although widely acknowledged as an approximation of averages, the traditional DCF model makes it easy to think, analyze, and make decisions as if the world were deterministic. In a deterministic model based on average assumptions, there is one outcome; there is no uncertainty in the model because the set of assumptions always leads to the same result. De Neufville and Scholtes describe the Flaw of Averages as the widespread, but mistaken assumption that evaluating a project around average conditions produces a correct result (de Neufville & Scholtes, 2011). The Flaw of Averages is a major error that occurs when using averages in deterministic models instead of proper stochastic variables (Savage, Danziger, & Markowitz, 2012). The math behind the Flaw of Averages is based on Jensen's Inequality. In 1906, Danish mathematician Johan Jensen proved that the average of all the possible outcomes associated with uncertain parameters is generally not equal to the value obtained from using the average value of the parameters except when the function is entirely linear.

DCF analyses are typically used ex ante, or forward-looking; the objective is to look into the future, to try to estimate the future cash flows and inform decision making. This perspective has significant implications; before the future happens, there are many possible futures, many scenarios that could happen. Traditionally, DCF uses single-point estimates in the computations and does not explicitly consider the inherent uncertainty in those numbers. While ignoring uncertainty or using naïve methods to assess uncertainty may make the mathematical computations simple, the final estimate of value that results may be very unrealistic and subjective. If the input assumptions are not balanced and realistic, the output may point to either an overly optimistic or conservative solution, and there could be missed opportunity. The reality is that the future is unknown, and that the exact cash flow amounts in the DCF will almost certainly not happen.

Real estate valuation is a subjective way of measuring the worth of real estate. No valuation model is absolutely accurate (Li, 2000). The future is uncertain; this is a fundamental fact about valuation and management and, life itself (Geltner & de Neufville, 2018). Uncertainty is due to the lack of knowledge and poor or imperfect information about all the inputs that can be used in the valuation. Unless the input variables are certain then the resulting outcome (value) is also uncertain. Uncertainty is anything that is not known about the outcome of a valuation at the date of the valuation.

The discount rate used in DCF is intended to reflect the level of risk in a project, but this oversimplifies risk by relying on a single discount rate when there are multiple sources of uncertainty. It ignores the effect of options or possible changes which may occur during the life of the investment as owners and managers have flexibility to respond to changes in the environment and economy by making decisions that could affect future cash flows. The Uniform Standards of Professional Appraisal Practice (USPAP) state that in the income approach, the market value DCF analyses should be supported by market-derived data, and the assumptions should be both market and property specific. Market value DCF analyses are intended to reflect the expectations and perceptions of market participants along with available factual data (The Appraisal Foundation, 1999).

Incorporating uncertainty into real estate DCFs changes the approach to real estate valuation. When uncertainty is factored into the analysis, the focus shifts to modeling and managing uncertainty to make better decisions. The single best present value is no longer the objective, the range and distribution of outcomes becomes the focal point. Major variables used in the DCF analysis vary depending on different market conditions, which impact the validity of the model. Uncertainties that are inevitably encountered in real estate investment and development include demand, prices, and rents rising and falling, and governments changing taxes, zoning, and other regulations. Emphasizing that the future is uncertain, the forecast is always wrong in that what actually occurs almost always differs in some way from the DCF projection (Geltner & de Neufville, 2015). This uncertainty underlies the value of the manager's ability to adapt plans to actual circumstances.

To adequately consider uncertainty, we must consider the entire distribution of future possibilities. Monte Carlo simulation allows us to do this by considering many scenarios in one simulation. A major feature of Monte Carlo simulation is that it draws on probability distributions to generate independent, random scenarios of the future. Simulation is a practical, efficient way to explore uncertainty and to choose between alternative strategies for managing it. A simulation is a way to consider what might happen in the real world under different circumstances. It uses some model of reality, to run a 'what-if' analysis. It provides an answer to what would the outcome of a certain model be if certain input conditions apply?

From the perspective of the present, the future can contain many possible scenarios, any one, but only one, of which could actually occur. The objective of the Monte Carlo simulation is to obtain representative results, that collectively mimic what could happen in reality. In Monte Carlo simulation we refer to each individual, independently generated random future scenario as a trial. Each trial has an equal chance of actually happening in the real-world future, as we model it in the simulation.

Monte Carlo simulation provides the means to analyze and evaluate alternative flexible management strategies. It is able to inform management decisions quickly and efficiently in the context of a huge number of possible combinations by deploying three main features: big-picture focus, speed, and sampling. Simulation is used to build intuition and gain insight into the general nature, or the big picture, of tactical, strategic, and design and planning decisions. Simulation does not tell us exactly what to do in any given circumstance at any specific time; it provides general insight into what could occur and can potentially be interpreted into how to manage uncertainties to one's benefit. Real estate investors, developers, and managers look for ways to add value to their projects; one fundamental way to do this is by taking advantage of uncertainty (Cardin, Nuttall, de Neufville, & Dahlgren, 2007).

The success of any decision model depends on the reliability of the underlying inputs. Monte Carlo simulation may lead to more optimal decisions by uncovering complex relationships often associated with uncertain inputs. Realistically, in a world of uncertainty cash flow projections should represent expectations of probability distributions. Through Monte Carlo simulation, the likelihood of a certain value for relevant variables can be explored, making the DCF more versatile and the derived valuations more reliable.

Currently real estate practitioners generally do not exploit the full potential of uncertainty, often regarding it as negative because of possible downside events. However, uncertainty can also increase performance if managed properly. Managing uncertainty to advantage capitalizes on upside opportunities, and reduces losses in case of downside events. The ever evolving macro- and micro-economic environment makes real estate professionals evaluate their strategies. They do so in order to generate a maximum present value of the property. Decision makers usually have various types of flexibilities regarding what actions they can take and when to take them. They can and will respond to the actual scenarios as they happen. Recognizing this explicitly in valuation can change our understanding of the value of the investment. With today's technology, real estate professionals can easily use Monte Carlo simulation as a tool to quantify the inherent uncertainties surrounding many of the estimates used to model real estate valuation.

The most fundamental element of value to office properties is the cash flow generated by office rents (Ciochetti & Fisher, 2007). The leases that govern these cash flows should be highly important, yet relatively little empirical research has been done on commercial leases. Commercial property leases and leasing strategy are among the most fundamental, important, and complex topics in real estate investment and property management (Geltner, Miller, Clayton, & Eiccholtz, 2014). The nature of leases, and the major considerations in leasing strategy, are key elements in the operational management of commercial properties and important determinants of the investment performance and value of such assets.

Tse suggests that the optimal lease term is the one that minimizes the expected costs of contract negotiation from the perspective of landlords (Tse, 1999). Geltner et al further that the optimal lease term is largely a trade-off between releasing costs and the value of flexibility

(Geltner, Miller, Clayton, & Eiccholtz, 2014). As more favorable market conditions come, landlords tend to focus not on just filling space, but on maximizing and sustaining the value of their properties by negotiating leases on the basis of a long-term business plan of value enhancement (Hayman & Ulrich, 1995). When uncertainty is considered, the focus shifts from simply maximizing financial returns, to modeling and managing uncertainty to make better ex ante decisions.

For variables in the office space market such as rent and vacancy, a level of predictability exists due to patterns in momentum and cyclicality. For fixed rent leases, landlords should require different terms depending on their expectations of future market conditions. While a long year term helps to reduce vacancy risk, short year terms allow the contract rent to be reviewed closely to the effective open market rent. Rationally, the landlord should choose the optimal lease term to maximize total expected income from the rental property, but what is the optimal lease term? How rational are the market's future rent expectations? How often are terms agreed to in leases unbiased predictors of the actual corresponding future market? There is little solid empirical evidence that helps to answer this question, and what there is appears to be mixed (Geltner, Miller, Clayton, & Eiccholtz, 2014).

Expectations regarding the future trend in rents in the relevant space market make the opportunity cost of the lease a function of the lease term. Longer-term leases are generally thought to reduce risk, and other things being equal, are typically perceived as preferred by landlords. While it is true that longer-term leases reduce the uncertainty in the landlord' future cash flow expectations by contractually fixing more future years' worth of cash flow, with rising rent expectations, the expected opportunity cost (to the landlord) of implementing a strategy of shorter-term leases is greater for longer-term leases. If, however, rents are expected to fall, then the opposite is true, and longer-term leases should have a lower opportunity cost or greater value. Flexibility is valuable because it gives options to decision makers. Shorter-term leases increase a certain type of flexibility for landlords and tenants, specifically, the flexibility to take advantage of favorable movements in market rents. In the period between rent review, rents are fixed in nominal terms. The landlord is unable to adjust the agreement according to market conditions if a lease with a relatively long duration is chosen. In this case, the nominal rent will remain at the current level for some time and cannot increase to compensate the landlord for higher levels of inflation (Tse, 1999). Geltner et al. have suggested that shorter-term lease lengths provide both the landlord and the tenant more flexibility to take advantage of favorable developments in the rental market (Geltner, Miller, Clayton, & Eiccholtz, 2014).

The DCF model can help to provide the basis for important management decisions. The key point is that decision-making and fundamental understanding in valuation can be improved by explicitly considering and modeling uncertainty and flexibility to adapt. Intelligent management should make use of its flexibility to react to circumstances as they develop. Good managers can maximize upside opportunities and minimize downside losses. This thesis proposes an iteration of modeling the DCF using the framework presented by Geltner and de Neufville in their 2018 book, Flexibility and Real Estate Valuation Under Uncertainty: A Practical Guide for Developers (Geltner & de Neufville, 2018). Departing from their focus on the implementation of large-scale or multi-phase real estate development projects, I use the tools presented to consider the impact of leasing strategies on valuation in the face of uncertainty. Through Monte Carlo simulation, the likelihood of a certain value for relevant variables is explored, making the DCF more versatile and the derived valuations more reliable. Through a sequence of periodic lease agreements modeled over a fifty-year time period, multiple scenarios are played out. The model then evaluates the scenarios as traditional DCFs, producing a present value metric for each. The simulation records any relevant summary results of the DCF analysis for review and comparison. The model then generates a new series of pricing factors, resulting in another scenario of leases, for a total of 10,000 trials.

LITERATURE

Commercial real estate, which includes office, retail, industrial, apartment, and hotel properties, represents a significant fraction of the investment universe. A real estate lease is simply the sale of the use of space for a specified period of time. Leases define the duration, rent structure, rights and responsibilities for both, tenants and landlords; essentially creating a governance structure affecting the value of leased assets. The ultimate value of commercial real estate emanates from its rental flow, which reflects the price the market is willing to pay for the use of space (Grenadier S. R., An Equilibrium Analysis of Real Estate Leases, 2005). The lack of a fully informed market causes decision-making difficulties with respect to leased premises, particularly when establishing the commercial components of the lease. The most commercial components of a lease are those components relating to the payment of rent and increases in rent during the lease term. (Robinson, 1999) Lease length, in particular, is regarded as an important determinant of the risk in the cash flows delivered by leased real estate assets. (Bond, Loizou, & McAllister, 2008) Despite the obvious importance of leases to asset value, little is known empirically about the determinants and tradeoffs among different lease provisions and the impact of these provisions on lease performance (Ciochetti & Fisher, 2007).

Some researchers have explored the strategic motivation for and the implication of lease structures. One prominent area of literature examines the choice and pricing of alternative contract provisions in the presence of asymmetric information (John D. Benjamin, 1992) (Mooradian & Yang, 2000; 2002). Wheaton suggests that agency problems in retail leasing may explain percentage lease arrangements (Wheaton W. C., 2000). Wheaton’s analysis shows how the lease arrangement improves the value of the relationship by anticipating opportunistic behavior of one party or another.

Others emphasize that real estate, as an asset class, plays a dominant role in both the U.S. and world economies and as such, similar to traditional financial assets such as stocks and bonds, commercial real estate involves the valuation of risky, state contingent cash flows over time. Grenadier states, while there has been an unceasing array of literature focusing on the application of modern financial theory towards financial assets, the degree of neglect toward applying similar modeling techniques to real estate has been puzzling. (Grenadier S. R., The Persistence of Real Estate Cycles, 1995)

Other researchers have used option pricing to value lease terms such as cancellation options (McConnell, 1985) (Grenadier S. , 1995), they suggest, options to renew “at market,” have no financial option value. It is suggested that these types of embedded options may be valuable to both parties (landlords and tenants) because they reduce costs of negotiation by establishing a starting place for discussions and the terms at which renewal may occur, while maintaining flexibility by marking the lease to market conditions upon renewal or by allowing the relationship to dissolve in the case of non-renewal. In this vein, many economic studies consider the use of options in addition to lease length to manage leasing relationships (K.J. Crocker, Mitigating Contractual Hazards: Unilateral Options and Contract Length, 1988) (K.J. Crocker, 1985). These authors add that unilateral options are often preferred to more bilateral or contingency clauses due to lower costs of exercising. They explore unilateral options designed to promote flexibility and those which operate like liquidated damages clauses by allowing the one of the parties to “breach” the original contract for a price; in office leases, renewal, termination, and expansion options in this fashion.

Specific to real estate, Grenadier provides a unified equilibrium approach to valuing commercial real estate leases using a game-theory approach to real options analysis. (Grenadier S. R., An Equilibrium Analysis of Real Estate Leases, 2005). He values leases as contingent claims on building values, where building values themselves are determined in an industry equilibrium, suggesting that lease rates reflect critical real estate market variables, such as the degree of concentration of developers, uncertainty over the future demand for space, and the current level of construction activity.

One feature of interest in this paper that has received little attention in real estate related academic literature is the variation in lease length. In economics related literature, empirical studies of coal and natural gas leases have examined lease length (Joskow, 2004) (Mulherin, 1986). These studies relate contract length to the potential for a landlord to seize the tenant's sunk investments, suggesting that when such opportunities exist, longer leases which, avoid frequent renegotiation, are expected to be optimal and the empirical data used confirms this hypothesis.

With respect to real estate, Geltner and Miller suggest that lease length largely reflects tradeoffs between releasing costs and needs for flexibility (Geltner, Miller, Clayton, & Eiccholtz, 2014). Fisher observes cross-sectional variation in average lease lengths across different uses of space in sale and leaseback transactions (Fisher L. , 2004). She posits that lease length is related to the anticipated difficulty of lease renegotiation and renewal. McCann and Ward model different transaction costs in determining the optimal length of leases when tenants expect to make repeated transactions within a specific timespan (Ward & McCann, 2004). Similar to aforementioned research, they view lease length as an endogenous part of the leasing decision.

Short-term leases provide flexibility so that the contractual relationship can be adapted to prevailing business and market conditions, whereas long-term leases may impede the renegotiation of the agreement as time continues and the original terms of the lease begin to deviate from those that would be optimal under current market conditions. Conversely, frequent negotiating, as occurs in short-term leases, can be costly, and especially so in the context of real estate and sunk investments, such as tenant improvements and moving, which may allow one party to extract value from the other. "The sum of all turnover costs can add up to a year's rent or more" (Grenadier S. R., The Persistence of Real Estate Cycles, 1995). As a result, the choice of lease duration is suggested to reflect the tradeoffs between the expected costs of repeated negotiation resulting from short term leases and the expected costs of poor adaptation to market conditions that results from longer leases (K.J. Crocker, Mitigating Contractual Hazards: Unilateral Options and Contract Length, 1988)

In another approach to lease term consideration, Tse analyzes the choice of the optimal lease term of office property in relation to expected lag vacancy, periods of rent-free and expected rental income growth (Tse, 1999). He finds that while a long year term helps to reduce vacancy risk, a short year term allows the contract rent to be reviewed closely to the effective market rent (similar to findings from other methods). His study shows that optimal lease terms tend to increase under conditions when the expected rate of rental growth decreases, the discount rate increases, or the expected lag vacancy increases.

There has been little empirical investigation of whether variation in lease terms produces variation in rents. Bond, Loizou, and McAllister use a sample of London office leases to investigate the relationship between lease length and initial lease rates, suggesting that leases which produce increased risks for investors, such as short-term leases, should provide an increased return or higher rent. Contrary to their hypothesis, they find that there was an upward sloping relationship between lease length and the initial rent, meaning that longer leases pay higher initial rates. (Bond, Loizou, & McAllister, 2008).

COMMERCIAL LEASING

Leasing may be the most important legal institution that has received virtually no systematic scholarly attention (Merrill, 2020). A lease is a transfer of an asset for a limited time in return for periodic payments called rent. The lessor is typically the owner of the asset and gets it back after the lease expires; the lessee is entitled to use the asset free of interference from the lessor during the lease provided the lessee pays the rent and performs other obligations of the lease. Leases are always for a limited duration, as distinguished from ownership, which lasts for an indefinite time. The limited time duration of a lease, as a matter of practice, is always less than the expected life of the asset (Geltner, Miller, Clayton, & Eiccholtz, 2014).

For a variety of reasons leasing has become the dominant way in which most commercial space is occupied and paid for by space users in the United States. The operation and management of built space has become a specialized industry in the US and other mature economies. It is believed that landlords specializing in this business can do a better job of it than one-off owner-occupiers who are not primarily focused on the operation and management of built space.

The operating cash flow on which the value of commercial properties is based derives fundamentally from the space market; this operating cash flow is mediated by leases, at least on the revenue side (Geltner, Miller, Clayton, & Eiccholtz, 2014). Therefore, lease terms are important determinants of the investment performance and value of real estate assets. In a survey conducted by Gallup for Richard Ellis in 1994, length of lease was cited by two out of three of the respondents as one of the most important factors behind property lending (Tse, 1999). The lease term refers to the initial contract lease period. The length of time covered by the lease, as well as the time the lease is signed and when it expires, can have value implications for the landlord.

While optimal lease-term length is largely accepted as a trade-off between releasing costs and the value of flexibility (Geltner, Miller, Clayton, & Eiccholtz, 2014), there is very little empirical research on lease duration and its implications in valuation. Booth shows that the present value of property with a longer lease duration will be more sensitive to a rise in inflation (Booth, 1993). Geltner et al. suggest that other things being equal, landlords would typically prefer longer-term leases (Geltner, Miller, Clayton, & Eiccholtz, 2014). Many suggest that shorter leases afford less protection to the landlord, are tenant-oriented, and that entering a long-term, non-cancellable lease may reduce the landlord's uncertainty. DiPasquale and Wheaton argue that with a very long lease term, the landlord has implicitly sold the rights to an uncertain market income stream in exchange for the present discounted value of all lease payments, pointing out that a shorter-term lease reduces the length of time for which the right to the space is relinquished, thereby preserving more flexibility for the landlord; in this sense, the value lies not in the lease, but in the lack of lease, temporally speaking (Di Pasquale & Wheaton, 1996).

Releasing costs are a consideration in leasing strategy that impacts duration. Typically, both the landlord and the tenant face costs associated with releasing. Landlords face expected vacancy and search costs to find a new tenant whenever a lease expires without being renewed. Tenants face moving costs, including disruption of operations. Because releasing presents potential deadweight costs to both sides in the lease agreement, releasing considerations generally affect both sides of the lease negotiation in the same direction as far as preferred lease term is concerned. In particular, releasing costs make it advantageous for both sides to prefer longer-term leases (Geltner, Miller, Clayton, & Eiccholtz, 2014). The consideration of releasing costs suggests a general bias toward longer-term leases at whatever term structure of rents prevails in the market, so as to minimize releasing costs over the long run.

In general, the need for and value placed on flexibility of lease duration is greatest in hotels and apartments and lowest in anchor retail and industrial space. Typical office space falls midway between, in both the releasing costs and the value of flexibility dimensions (Geltner, Miller, Clayton, & Eiccholtz, 2014). Figure 1 presents stereotypical characteristic lease terms that prevailed in various types of commercial property space markets during the latter part of the twentieth century.

Hotel:	1 day–1 week
Apartment:	1 year
Small retail:	2–5 years
Office:	3–10 years
Anchor retail:	5–15 years
Industrial:	5–20 years
Unique corporate space:	20 + years or user-occupant ownership

Figure 1: Stereotypical Prevailing Commercial Property Characteristic Lease Terms in the United States (Geltner, Miller, Clayton, & Eiccholtz, 2014)

In a survey of commercial real estate professionals, the terms for Manhattan commercial leases generally range from two to fifteen years. The majority citing two-year lease terms being the minimum because of the costs required to remarket a property (remarketing costs include legal fees, build-out costs, and rent lost during vacancy on the landlord's side, and the cost of moving and lost productivity on the tenant's side), and fifteen years being the upper bound due to uncertainty about future rents (as discussed throughout this paper, uncertainty pervades the industry more than just far out in the future). The survey also found that the most common lease terms were 5- and 10-year leases. For tenants seeking commercial office space in Manhattan, there are advantages and disadvantages associated with both short- and long-term commercial leases (Rosinsky, 2018). From a tenant perspective, the greatest advantage of a short-term lease is that it offers the tenant maximum flexibility, while the drawbacks include both unwillingness of landlords to provide much in the way of tenant improvements to the space and the possibility of rent increases upon expiration. Again, from the tenant perspective, long-term leases offer several distinct benefits including predictability of long-term real estate costs and protection from the risk of rent increases throughout the terms of their leases (although sometimes those benefits are tempered by agreed-upon rent escalations). Longer lease terms also provide tenants with greater leverage in negotiating terms with a landlord (for example, a landlord will generally offer more in the way of tenant improvement if they are locking in a reputable tenant and have a sufficient lease term to amortize the cost of construction). The main disadvantage for a tenant of a long-term lease is reduced flexibility.

In a sample of 7,000 US office leases that were brokered in 1989 by CB Commercial, it is found that the mean lease length is five years, but no leases are longer than twelve years. (Di Pasquale & Wheaton, 1996). A recent report found that New York City Manhattan's 12-month moving average lease term length was 123 months (10 years, 3 months) as of late 2018 with 10-year lease durations prevailing (Avison Young, 2021).

As empirical data is scarce, I obtained a leading commercial real estate company's highly proprietary data set of Manhattan, New York commercial real estate transactions to review lease terms and gain some insight. The data was anonymized and limited in features included for business reasons. The original data set was comprised of 10,000+ transactions with lease commencement dates ranging from 2004 to 2027 and lease term durations ranging between three and twenty years. After scrubbing the data of observations that were incomplete or duplicates, 8,100 observations remained. Several filters were applied to ensure a robust data set for analysis. Leases that were less than twelve months (one year) in length were excluded. Of the remaining observations, one was dropped because it was an outlier in terms of length (99 years). Others were eliminated because they were miscoded in a way that could not be interpreted. A summary of lease term duration is presented in Figure 2.

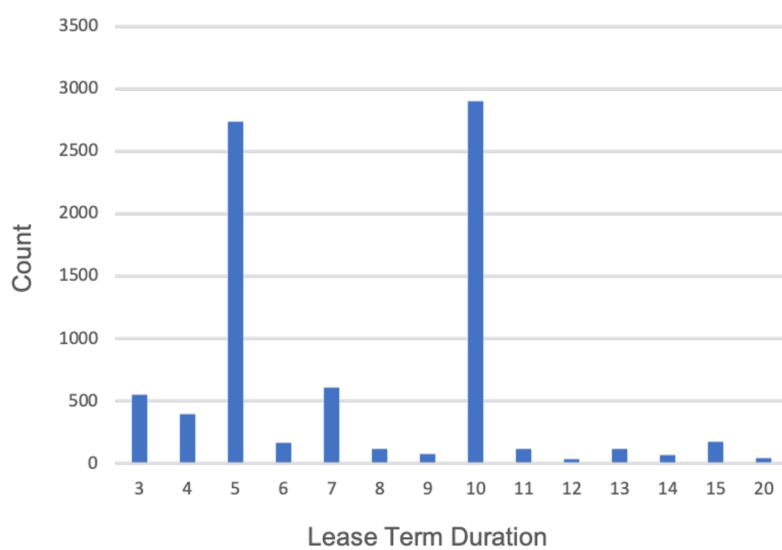

Figure 2: Manhattan Commercial Office Leases (Derived from proprietary data)

The mean of the 8,100 observations was 7.33 years. The most prevalent lease duration in the data was a 10-year lease, accounting for approximately 36% of the 8,100 observations. 5-year lease terms closely followed, accounting for approximately 34% of the observations. Approximately 12% of leases were 3-4-year leases; another 12% of leases were 6-9-year leases; and approximately 6% of the leases were greater than 10-years. This data aligns with and confirms both the survey and findings discussed above.

Given that commercial property leases and leasing strategy are among the most fundamental, important, and complex topics in real estate investment and property management (Geltner, Miller, Clayton, & Eiccholtz, 2014), and that the nature of leases are key determinants of the investment performance and value of such assets, I question whether different strategies applied to setting lease duration can result in added value in the face of uncertainty. Building on DiPasquale and Wheaton's argument that with longer lease terms, landlords implicitly sell the rights to an uncertain market income stream in exchange for the present discounted value of all lease payments, I propose an approach that incorporates that uncertain market income stream and market relevant data into the duration decision to quantitatively explore the impact of varying lease durations on valuation. Rather than assuming long-term leases are optimal (as suggested in previous work), I propose considering a more-informed approach to setting lease duration, where historical data and relative knowledge of position within market cycle (Jalori, 2017) are considered.

UNCERTAINTY & THE VALUE OF FLEXIBLITY

The forecast is always wrong. We cannot know what will happen in the future. No matter how hard we try to predict long-term requirements, the forecast is "always wrong." Trends change, surprises occur. (de Neufville & Scholtes, 2011) The term "flaw of averages" is a clever pun; it combines the notion of a flaw, or a mistake, with the concept of the law of averages, which is the notion that future events will balance out toward an average. Therefore, emphasizing that using the average input values to estimate future expected outcome values is a mistake. The flaw of averages refers to the idea that we should not base decisions only on the average values of the input parameters; doing so may lead to missed opportunities and/or risks that average values mask. The flaw of averages in general consists of failing to look beyond average conditions, which results in failing to consider all possible scenarios. In the context of real estate and the practice of discounted cash flow (DCF) valuation, the flaw of averages resides in excessive dependence on the traditional, single-stream cash flow pro forma.

Traditional investment theory defines real estate as a triangle of space, money, and time. In this sense a particular usage is attributed to a defined space which generates an estimated cash flow over a specified time. This perspective implies a relatively deterministic understanding of real estate where inflexibility appears to be a characteristic feature, creating the impression of certainty concerning the use, and therefore cash flow of real estate. The more uncertainty is included in the investors' considerations the less adequate the traditional valuation methods appear (Lucius, 2001). Uncertainty can be driven by changes in the macro-economy and the local economy. The market of focus in this thesis is the market for the use of built space. On the demand side of this market are potential tenants, it is by serving such demand that the real estate generates value. Underlying the demand side of the space market is the local and national (or even global) economy, which determines the amount, type, and location of real estate assets that will be useful and of value. On the supply side of the space market are landlords or other types of owners controlling and managing the operation and usage of real estate. The equilibrium between supply and demand in the space market determines the magnitude of the benefit flows attributable to the real estate. This annual benefit (a net income stream) is the output from the space market to the asset market in the form of a valuation. Uncertainty manifests in what the prices and usage levels and therefore the magnitude of income will be.

Valuation is the process of estimating price. The methods used to determine value attempt to model the thought processes of the market and estimate price by reference to observed historic data. This information is utilized in the discounted cash flow (DCF) valuation model to determine the single point valuation figure. However, the valuation will be affected by uncertainties: uncertainty in the comparable data available; uncertainty in the current and future market conditions and uncertainty in the specific inputs for the subject property. These input uncertainties will translate into an uncertainty with the output figure, the estimate of price.

DCF projections are based on a variety of sources, including knowledge of fixed contractual obligations (such as mortgage payments and lease terms); informed best estimates of specific income and expenses; and assumptions about the relevant real estate market and overall economic conditions, such as future prices. When estimating the “market value” using DCF analysis, the future cash flows should represent unbiased expectations, and the discount rate should equal the “opportunity cost of capital” (OCC) faced by investors or what they could expect to earn by investing in a similar investment of similar risk.

The more accurate the future expectations the more robust the valuation. This highlights the importance of dealing with future expectations in the valuation process and suggests that the adoption of multiple scenarios will greatly facilitate the valuer in providing sound competent professional advice. Uncertainty is a universal fact of property valuation. All valuations, by their nature, are uncertain (French, 2005). A 2002 report on valuation from the Royal Institution of Chartered Surveyors (RICS), stressed that ways should be sought to establish an acceptable method by which uncertainty could be expressed in the valuation (Royal Institution of Chartered Surveyors, 2002).

Valuing projects correctly requires the recognition of uncertainty. Evaluations based on average or "most likely" forecasts of future situations will systematically lead to incorrect answers. Proper analyses require an understanding of the effects of uncertainty. This thesis looks at the way in which uncertainty can be incorporated into the DCF model. Building upon Geltner & de Neufville's framework for valuation under uncertainty (Geltner & de Neufville, 2018), this is done by recognizing that the input variables are uncertain and will have a probability distribution pertaining to each of them. In using a probability-based valuation model it is possible to incorporate uncertainty into the analysis and address the shortcomings of the traditional DCF model. This is explicitly explored in this thesis by considering the uncertainty about market conditions and leasing strategy.

Flexibility enables landlords and managers to adapt their leasing strategies over time to the market conditions to achieve optimal performance of their assets. Unexpected changes can create both gains and losses. Analysts often gloss over uncertainty as it presents risk in their models, however uncertainties can also create new opportunities. This thesis suggests that owners and landlords can achieve better results by considering the ability to adapt to circumstances as they arise. Leasing strategies that can be modified to take advantage of new opportunities or to mitigate challenging conditions can lead to more optimal outcomes. The future is uncertain. Financial projections that do not account for a range of possibilities that may occur over a long lifetime run the risk of leaving significant value untapped, as shown in the following chapters. An uncertain future provides a range of opportunities and risks. We can deal best with these eventualities and maximize the expected value if we approach leasing with flexibility.

A flexible leasing strategy allows, but does not require a specific lease duration. A flexible leasing strategy positions the landlord to take stock of the current conditions, understand where the market is and select a lease duration that makes sense based on these conditions. A flexible leasing strategy recognizes that we will learn more about the market as time progresses, and that the future is, in fact, uncertain. Rigid leasing strategies lock in the possibility of less optimal terms simply because it's the way they operate. Acknowledging uncertainty, flexible leasing strategies allow owners and landlords to take advantage of favorable opportunities and mitigate downside risk when possible.

REAL ESTATE MARKET CYCLE

Property valuations are usually undertaken in an environment of uncertainty and incomplete information. For example, the exact measure of rental growth, a key component of cash flow projections, based on expectations is difficult because expected rental values are likely to be biased upwards during economic expansion, and downwards during contraction (Born & Phyrr, 1994). Meanwhile, empirically, it has been shown that office properties don't beat the rate of inflation over the long-run (Eichholtz, 1997) (Fisher, Geltner, & Webb, 1994) (Wheaton, Baranski, & Templeton, 2009). Growth in commercial real estate over the long run has been found to be slightly less than inflation because of depreciation (Fisher, Geltner, & Webb, 1994) (Wheaton, Baranski, & Templeton, 2009). Most real estate markets behave with both predictability and randomness and this should be reflected in the way financial models are created (Leung, 2014). There are infinite possibilities when it comes to events or shocks that may influence the real estate market and therefore, valuation. Understanding the general nature of real estate markets is important when contemplating valuation.

Most real estate assets trade in private search markets for whole (unique) assets rather than public auction markets for homogenous shares or units (Geltner & de Neufville, 2015). The mechanics of how real estate prices move over time has been studied extensively and can be described using a Stock Flow Model, which describes the process of how a durable stock of goods, such as real estate, increases and decreases and interacts over time with the flow of usage (i.e. leasing) of that stock of goods (Leung, 2014). Di Pasquale and Wheaton illustrate the relationship between construction, asset markets and space markets in the Four Quadrant model (Di Pasquale & Wheaton, 1996). As the economy goes through its ups and downs, real estate prices and rents go up and down because demand changes without a quick response from the supply side due to the durability of real estate and lag to deliver new space. Eventually, increases in rents and prices promote new construction which gradually alleviates pressure on rents as the new space is delivered to the market. Since there is a lag in construction, it is rare that the exact amount of completions comes online and perfectly meets demand; there will be overbuilding and underbuilding which leads to real estate incurring its own cycle.

Wheaton suggests that there are two ways the real estate market could manifest itself; one where real estate developers are completely rational and forward-looking, and another where they are backward-looking or myopic when forecasting future supply and demand (Wheaton W. , 1999). When agents are rational and forward-looking, they have a good understanding of how the market behaves with uncertainty, so prices reflect the present value of future cash flows and the uncertainty surrounding those cash flows. Whereas extrapolating average historic rates forward in financial models would be myopic, backward-looking behavior. Giaccotto and Clapp suggest that only current, rather than historical data should be used in valuation of real estate (Giaccotto & Clapp, 1992). Wheaton finds that both cases generate endogenous long-run cycles within real estate as developers struggle to forecast the exact amount of space to build. It is usually best to assume that future market prices will tend to reflect dynamic characteristics similar to those that occurred in the past, it is important to note, that is not the same as extrapolating average historic rates forward.

Recent databases and econometric discoveries can tell us about the nature and magnitude of uncertainty and volatility in real estate values. As depicted below in Figure 3, the long-run market cycle is the oscillating nature of real estate prices that is easily observable in a time-series graph. The peak-to-peak or trough-to-trough timing has been between 15-20 years in commercial real estate cycles (Geltner, Miller, Clayton, & Eiccholtz, 2014). There is also volatility that exists daily/monthly/yearly along the cycle preventing a smooth oscillating curve. Market volatility is a type of uncertainty that results from new information that provides a shock that the market takes time to respond to. This information allows us to create a model of the cyclical natures of real estate markets.

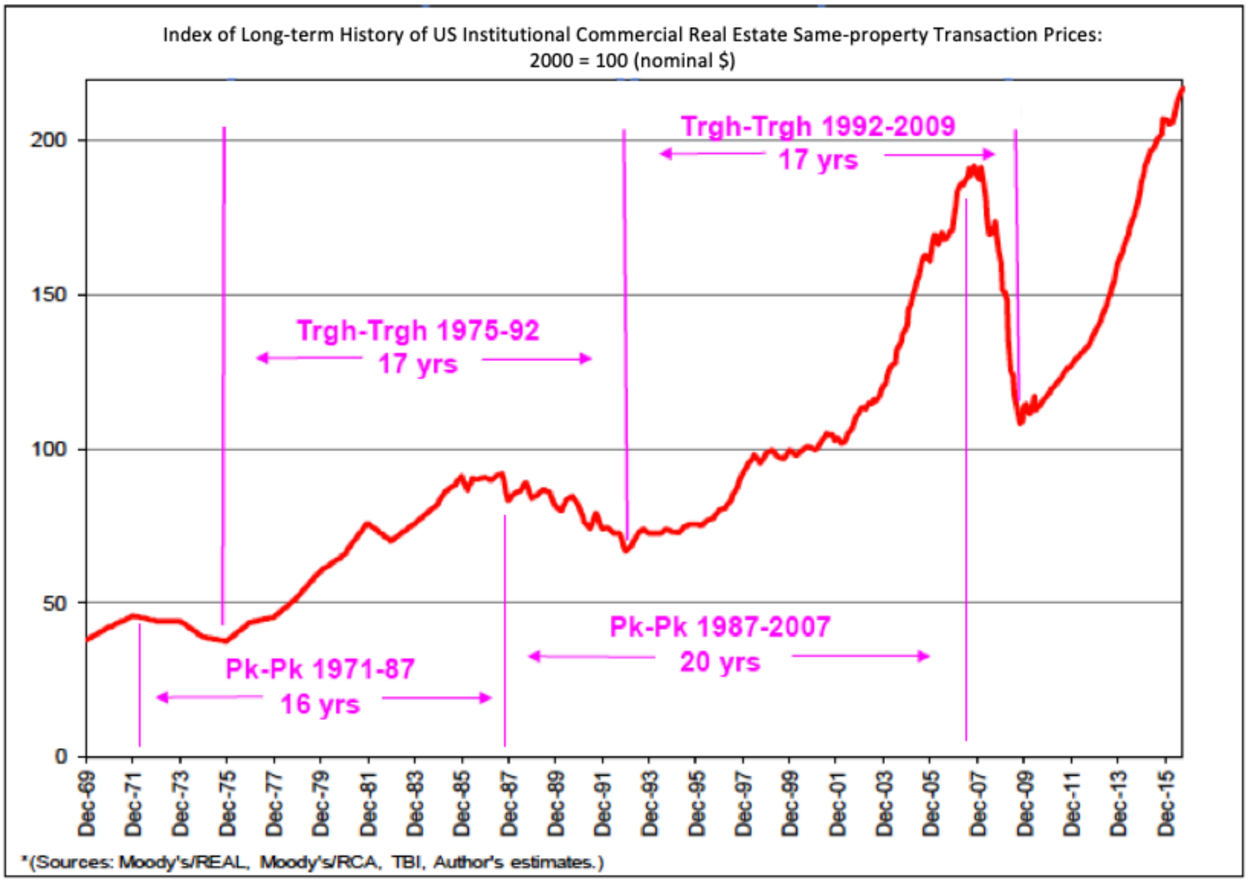

Figure 3: Long-run Market Cycle in U.S. Commercial Property Major Assets (Geltner 2020)

To model the market's cyclical effect, a sine curve is used to create a pricing factor for cash flows each year. The coefficient in front of the sine curve affects the amplitude, or the height of the waves. The numbers inside the sine function affect the duration and position of the curve. The sine curve is modeled:

$$\frac{a}{2} * \sin\left((\mathrm{y} + \mathrm{p}) * \frac{2\pi}{d}\right) + 1$$

Where: a = maximum amplitude in %

y = number of years since start year, with start year = 0

p = cycle starting position (years after upward mid-point)

d = duration of one full cycle in years

(+ 1 at the end of the function shifts the entire sine curve up to have a mid-point of 1 instead of 0)

The simulated real estate market cycles include all the components of actual cycles: market volatility, cyclicality, mean reversion, autoregressive behavior and idiosyncratic noise, but are kept as simple as possible. We use available data on the historical variations in market prices for real estate to model real estate market cycles to generate pricing factors as a way to develop representative scenarios. These enable us to account realistically for the price dynamics of real estate investments in our simulation analyses.

MONTE CARLO SIMULATION APPROACH

The traditional DCF model does not allow us to explore possibilities that might exists, or how we might intelligently react to what can happen as the investment develops over time. In traditional DCF analysis, each variable has a certain distribution of possible values, only one of which is actually used. Recognizing that the input variables are uncertain and will have a probability distribution pertaining to each of them, it is possible to incorporate uncertainty into the analysis and address the shortcoming of the current DCF model (French, 2005). Monte Carlo simulation attempts to imitate the ways that variables could combine as the future unfolds (Phyrr, 1973). Monte Carlo simulation creates a probabilistic model that uses probability density functions (a range of possible values) to describe each input. Each variable's uncertainty can be quantified on its own, then Monte Carlo simulation outputs the effect of uncertainty as a whole on the DCF valuation. Monte Carlo simulation can be used as a tool to quantify the real and inherent uncertainties surrounding many of the estimates used to create the DCF. Monte Carlo simulation is a widely available practical tool that allows calculation of distributions of the future market conditions within which the asset will generate cash flow. It enables us to calculate realistic expected values and to produce target curves and other information that fully describe the consequences of leasing choices.

There has not been wide-spread adoption of stochastic valuation techniques in real estate finance despite the positive track record of Monte Carlo simulation in corporate finance (Marshall & Kennedy, 1992). Recent decades have seen tremendous development in our ability to quantify the underlying volatility necessary for modelling the value of real estate, specifically, the advent of rich datasets of property returns and price dynamics, initially based on appraisals, and more recently directly on transaction prices in the private property market (Geltner & de Neufville, 2015). With today's technology, real estate professionals can easily use Monte Carlo simulation as a tool to quantify the inherent uncertainties surrounding many of the estimates used to model real estate valuation.

Simulation can be used to build intuition and gain insight into the big picture of tactical and strategic decisions. Simulation does not tell us exactly what to do in any given circumstance at any specific time, it provides general insight into possibilities. As such, simulation models should have a degree of abstraction and simplification. There is genuine practical value in elegance and parsimony (Geltner & de Neufville, 2018).

The success of any decision model depends on the reliability of the underlying inputs (Kelliher & Mahoney, 2000). The probability distributions that are specified in the inputs to the model govern the outcomes. To understand the entire range of possible outcomes, the probability distributions of future possibilities need to be quantified in order to have the information needed to carry out quantitative analysis. It is important to have more than one approach to obtaining information for quantifying probability distributions to avoid biases. Experts in many fields often believe they “know” what will happen regardless of persistent evidence to the contrary. Being confident in their knowledge, experts often minimize the range of possibilities and estimate probability distributions much too narrowly. Following trends is also a common, instinctive way to forecast the future, the problem is, trend forecasts neglect the fact that trends regularly break. It can be dangerous to assume that current trends will continue throughout the life of an asset. The dynamics (random volatility as well as tendencies toward, or susceptibility to, cyclicality, mean-reversion, inertia, etc.) are more informational, than the trend. Statistical analysis of historical data on market dynamics is generally the preferable way to obtain the probability distributions for investment analysis, when sufficient data exists. Fortunately, detailed historical data on real estate prices over time are increasingly available in major markets. These enable us to estimate the probability distributions of future real estate prices and rents using statistical analysis. A combination of expert judgment, market dynamics, and empirical data was used to quantify the probability distributions of model parameters.

Scenarios describe uncertainties in the DCF valuation. A scenario is one specific sequence of events that may occur. In each scenario available data on the historical variations in market prices for real estate is used to model real estate market cycles to generate pricing factors as a way to develop representative scenarios. The pricing factors attempt to account for the price dynamics of real estate in the simulation analyses. A pricing factor is a ratio that multiplies the original, single-stream DCF cash flow expectation to arrive at a future cash flow outcome for a given scenario. A base case DCF is used as a starting point and cash flows are modified by multiplying them by the pricing factors corresponding to the leasing strategy. With pricing factors projected into the future, different lease strategy DCFs are generated and compared under uncertainty.

Monte Carlo simulation is then a repetitive process. The simulation:

- Generates a trial scenario consisting of a future sequence of what might happen based on input probability distributions and a model of the functioning of the project in each period
- Calculates the project performance metrics of interest for the project outcome resulting from that scenario (for example, the DCF present value reflecting the future cash flows in that scenario for each strategy)
- Repeats this process many times (in this model, 10,000 trials of 3 strategies each are run in less than 5 seconds), thereby generating a sample (or simulation run) of 30,000 outcomes (one for each strategy of each trial)
- Displays results as graphical and statistical summaries of the entire distribution of the outcomes for the sample.

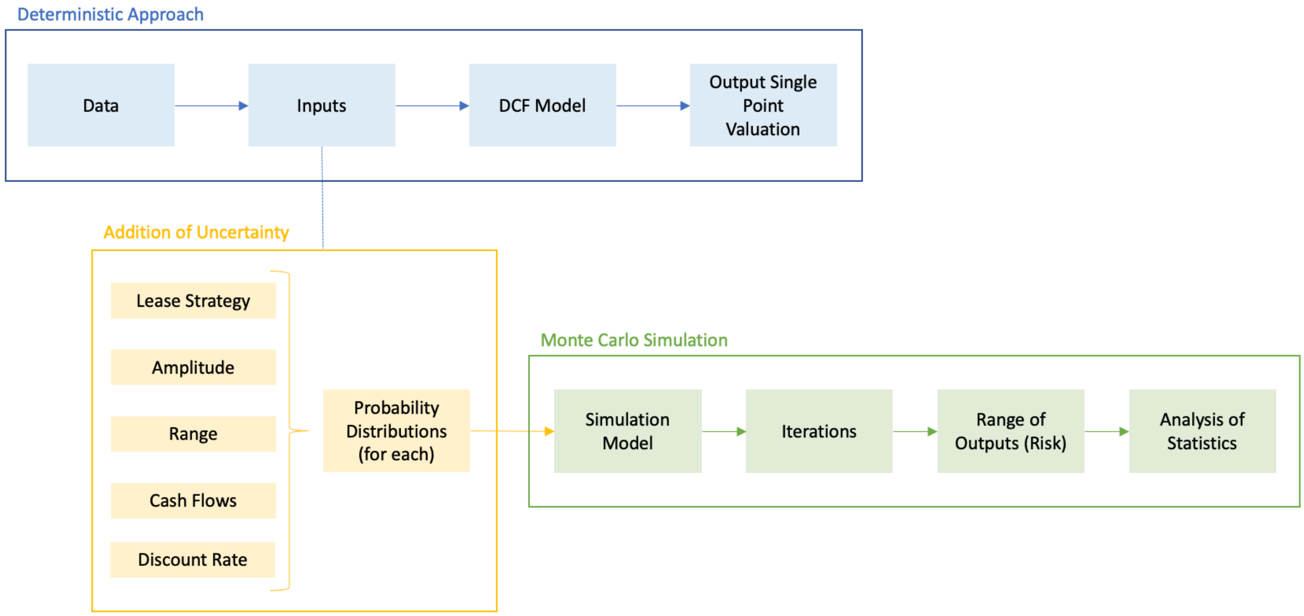

Figure 4: Deterministic Model with Addition of Uncertainty via Monte Carlo Simulation

The advantage of Monte Carlo simulation is that it provides statistical information about the certainty of the result. The standard deviation is a representation of the uncertainty of the outcome. The skewness represents the degree of asymmetry of the distribution around its mean (if the skew is to the right side it indicates the upside potential, the likelihood of an outcome higher than the mean is greater than the downside risk and vice versa). The importance of the statistics is that it is placing the single point valuation in the context of uncertainty of inputs and the corresponding risk pertaining to the output. This increases the utility of the valuation.

MODEL IMPLEMENTATION

While the origins of this model come from the basic spreadsheet software, Microsoft Excel (and are perfectly functional within), for this thesis, it has been implemented in the Julia programming language as a contribution to future potential uses in the context of big data, data science, and machine learning. Julia is a high-level, high-performance, dynamic programming language. While it is a general-purpose language and can be used to write any application, many of its features are well suited for numerical analysis and computational science. Julia is dynamically typed, feels like a scripting language, and has good support for interactive use.

The impact of lease duration is the focus of this thesis. As such, I model three different leasing strategies within one simulation. Strategies can be as simple or complex as the user decides, but it is important to note the value in simplicity. Within the model, a strategy is a function which is allowed to use any visible information in the market cycle in progress to generate its discrete probability distribution, which should capture the durations of leases it would consider signing, plus the probability of signing a lease of that particular duration. The first will be a shorter-term, 5-year strategy, which assumes the lease is renewed or replaced every five years; the second will be a longer-term, 10-year strategy, which assumes the lease is renewed or replaced every 10 years; and lastly there is variable strategy, titled Market Knowledge, which implements probability distributions while being informed by the context of the relative position within the market cycle. Two of the strategies are intended to be used as a reference to the conventional lease term durations (5- and 10-year leases).

The Market Knowledge strategy is a strategy that is intended to question whether added value can be gained by engaging data to establish lease durations, rather than following the status quo (typical 5- and 10- year lease durations). As shown in Figure 5 below, there are phases within a market cycle that regularly occur; mid-cycle heading up, peak heading down, mid-cycle heading down, and bottom headed up (Jalori, 2017). The Market Knowledge strategy uses these relative positions to establish distributions of lease duration probabilities.

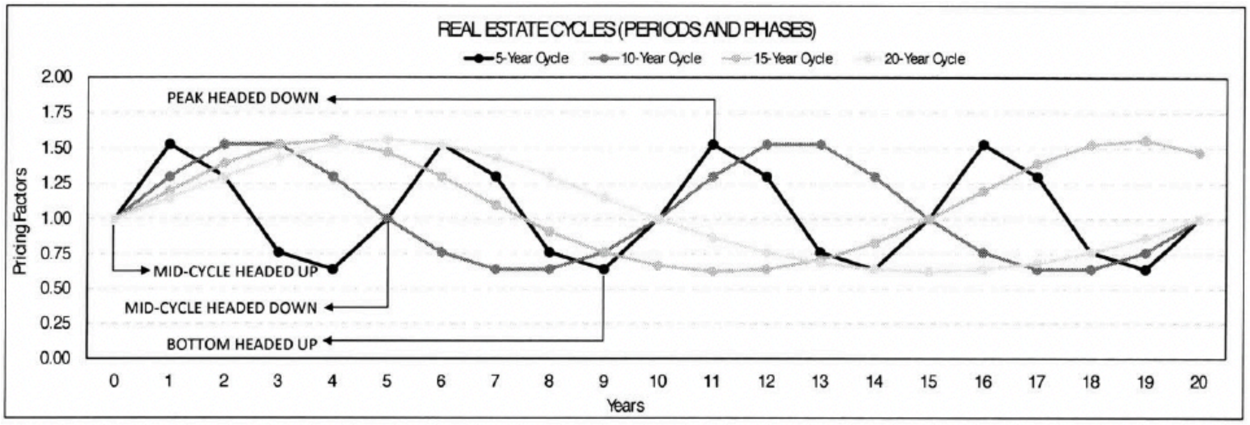

Figure 5: Real Estate Cycle Periods & Phases (Jalori, 2017)

The Market Knowledge strategy consists of probability distributions for different lease durations based on two factors stemming from those phases (from the landlord's perspective):

- Is the market currently increasing or decreasing?
- Are pricing factors currently above or below average?

Lumpy surges in supply, fairly predictable trends in absorption and vacancy rates, and the tendency to use similar lease terms within a market make it possible to determine whether it is a relatively soft or strong rental market from the landlord's perspective. In a soft market tenants are more able and more likely to bargain for longer lease terms. If the market is soft but expectations are for strong increases in rents in a few years, landlords are more likely to want to use short-term leases (Geltner, Miller, Clayton, & Eiccholtz, 2014). Extrapolating from this information, the Market Knowledge strategy incorporates relative direction and establishes the following:

	Landlord Perspective	Tenant Perspective	Probability Distribution
Above Mean Price Factor & Market Increasing	Prefers a medium lease to lock in elevated cash flow, & potentially catch the peak at expiration	Prefers a longer lease to avoid future increases	5-year 30% / 6-year 40% / 7-year 30%
Above Mean Price Factor & Market Decreasing	Prefers a long lease to lock in elevated cash flow, & hopefully avoid the bottom of a downturn	Prefers a shorter lease to capture decreases sooner	7-year 20% / 8-year 30% / 9-year 30% / 10-year 20%
Below Mean Price Factor & Market Increasing	Prefers a shorter lease to potentially catch the upswing at expiration while minimizing long periods of low cash flows	Prefers a longer lease to lock in lower rates, and avoid future increases	3-year 20% / 4-year 40% / 5-year 40%
Below Mean Price Factor & Market Decreasing	Prefers a medium lease to lock in elevated cash flow, & potentially catch the peak at expiration	Prefers a shorter lease to capture further decreases	3-year 30% / 4-year 40% / 5-year 30%

Table 1: Market Knowledge Strategy (Source: Author)

Lease rates will be locked in by their lease terms upon execution while the market can move in either direction at any rate of change during the lease; that future unfolds upon lease execution. Execution of a lease unlocks the future as it plays out in that particular market cycle scenario as the lease duration moves the cycle forward in time to the subsequent lease execution. In this model, we account for risk between leases by increasing the discount rate employed in the DCF calculations between leases.

In each scenario the three strategies are applied to the same 50-year series of market cycles. The model then evaluates the scenario as three different DCFs, producing a present value metric for each strategy. The simulation records any relevant summary results of the DCF analysis. Pricing factors provide a simple and straightforward way to reflect uncertainty over time in the DCF. They provide the means to incorporate our estimates of the probability distributions for relevant parameters (such as revenues) in the spreadsheet. The model then generates a new series of pricing factors (for a 50-year market cycle), resulting in another three-strategy scenario, for a total of 10,000 trials or 30,000 DCFs. It is computationally economical to apply the same pricing factors across the board to all the cash flow components, if warranted in real-world applications, it is not difficult to develop and apply pricing factors separately for each of the cash flow elements. Part of the art of modeling is to abstract reality, to avoid making the model too complex.

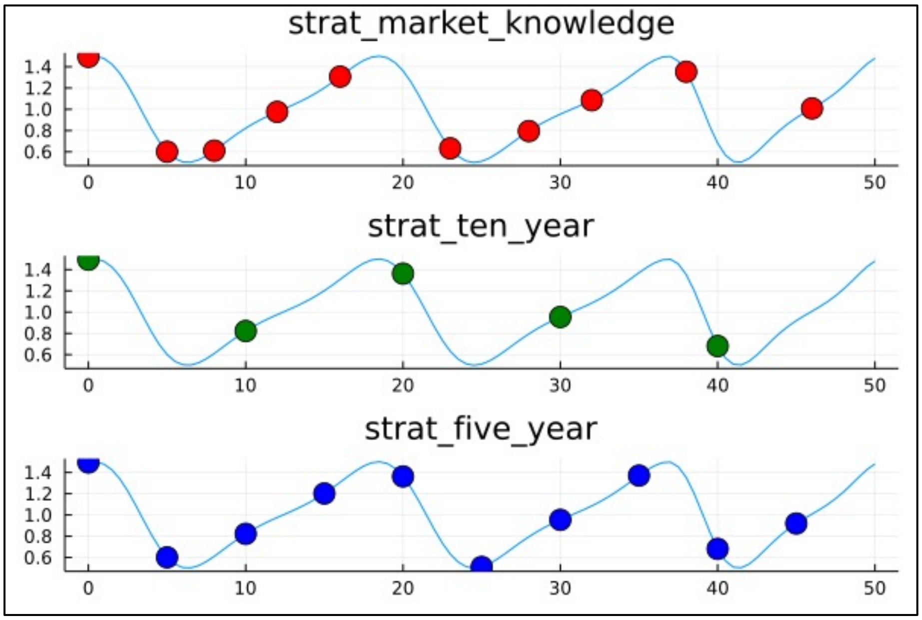

Figure 6: Sample Trial Scenario of Lease Executions

RESULTS & INTERPRETATION

One simulation consisting of 10,000 market scenarios, comparing three different lease duration strategies, totaling 30,000 DCFs is completed in 4.8 seconds. Summary statistics are an intuitive way in which simulation results are displayed. Below is a summary of the output distributions of the three strategies:

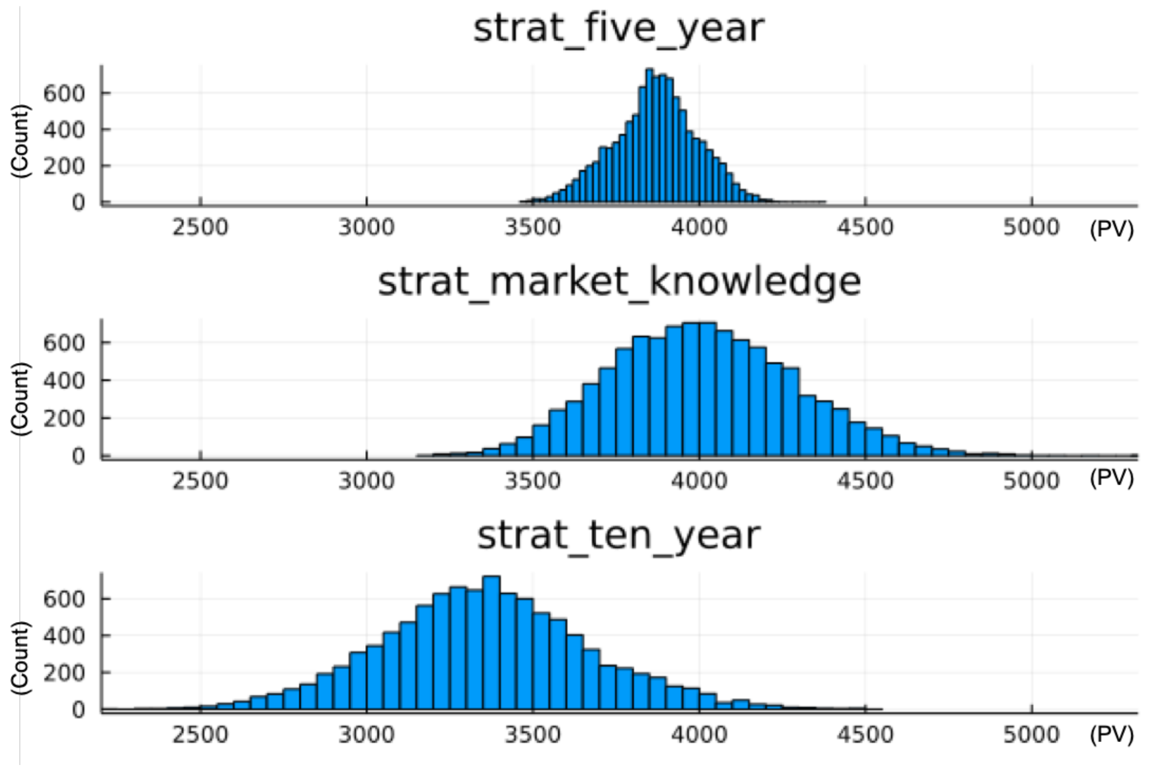

Figure 7: Simulation Summary Output Distributions

Strategy	Mean	STD	Range	VAR (5%)
Five-Year	3869	125	903	3654
Market Knowledge	**4010**	**278**	**2169**	**3573**
Ten-Year	3356	316	2334	2859

Table 2: Simulation Summary Statistics

Overall, the strategy using market knowledge outperforms both the 5-year and 10-year strategies with respect to the average(mean) outcome and the upside potential(higher high). The strategy employing market knowledge also outperforms the 10-year strategy in downside risk (having a higher low), while it falls short of the 5-year strategy in downside risk (having a lower low).

The quartile measures the spread of values above and below the mean by dividing the distribution into four groups. Each quartile contains 25% of the total observations. Quartiles are used to calculate the interquartile range, which is a measure of variability around the median, and an indication of the dispersion. Generally, the data is arranged from smallest to largest:

- First Quartile: 0-25%
- Second Quartile: 25.1-50.0% (up to the median)
- Third Quartile: 50.1-75% (above the median)
- Fourth Quartile: 75.1-100.0%

Below is a summary of quartile information from the simulation:

Strategy	min	25%	median	75%	max
Five-Year	3464	3790	3873	3950	4368
Market Knowledge	**3150**	**3811**	**4001**	**4195**	**5320**
Ten-Year	2203	3152	3351	3554	4536

Table 3: Simulation Quartile Information

For all three strategies, we find that there is greater dispersion among the smaller outcomes of the dataset than among the larger outcomes because the 25th percentile is farther away from the median than the 75th percentile in all cases.

The interquartile range is the range of the middle half of the data that shows how dispersed or spread out the data is. The interquartile ranges of the strategies support Figure 7: Simulation Summary Output Distributions above. It is observed that the five-year strategy has less dispersion (interquartile ranges: 5-year = 485; Market Knowledge = 1,045; 10-year = 1,351), and many observations are tightly gathered around the median and mean. While not the same, the market knowledge and ten-year strategies have similar shapes, flatter and wider than the tight peak of the five-year strategy, as there is more dispersion in their ranges.

Probability distributions are often represented by graphs. A good way to understand the opportunities of improving the leasing strategy is by looking at the target curve, known technically as the cumulative distribution function. It represents the cumulative chance of obtaining a result below any specific target value, going from the possibility of a result below the lowest value (which has no chance) to a result at or below the highest value (which has 100 percent chance). The target graph simply depicts the percentage of performance outcomes below the specified target level. The horizontal axis defines the possible events that can happen, and the vertical axis gives the probability of each such outcome. Cumulative target curves provide easy visual means to identify preferable alternatives. As a general rule, if one curve is always to the right, then it is the preferred case. Shifting the curve to the right provides a higher chance of achieving higher targets. The cumulative target curve easily identifies the value at risk of alternatives. A cumulative target curve uses a simple line graph to depict the distribution of the simulated results. It plots the probability of occurrence (on the vertical axis) against some performance metric of interest, such as the present value on the horizontal axis. The curve creates an estimate of the ex-ante probability distribution of the overall valuation. The curve represents a sample of the results of a simulation of possible future scenarios or outcomes for the target metric of interest, present value.

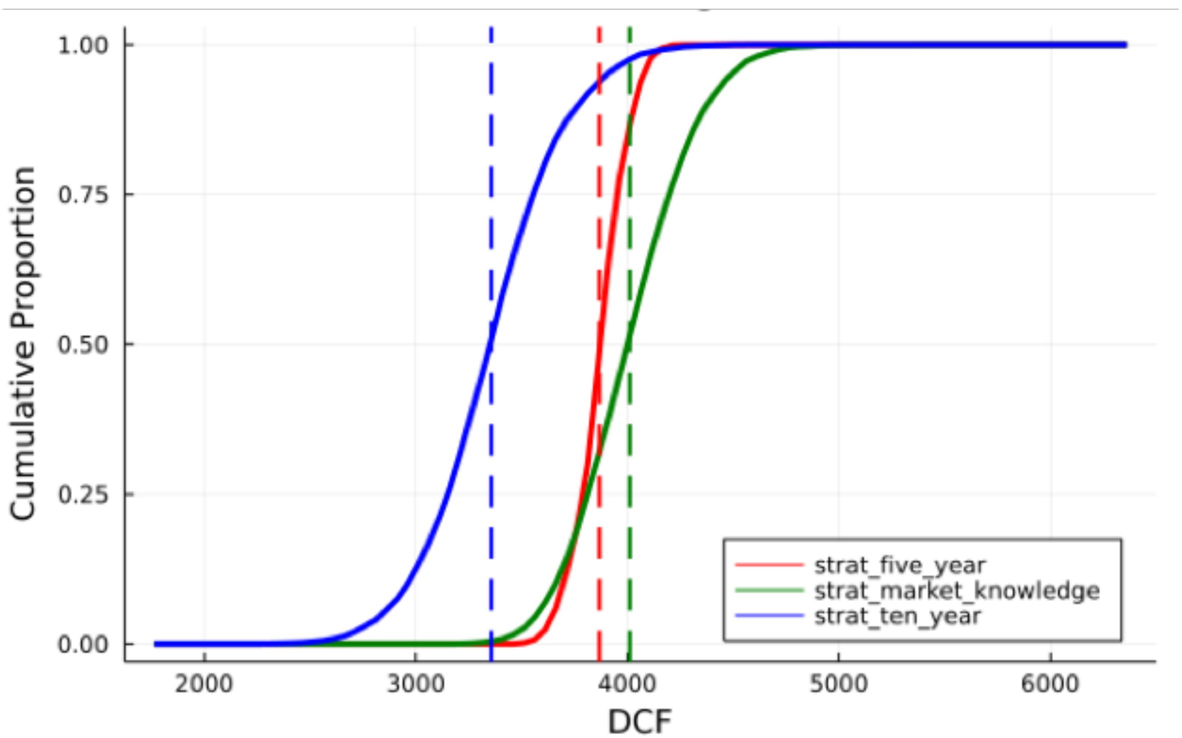

Figure 8: Simulation Cumulative Target Curve

Cumulative target curves sometimes cross each other (as observed in this simulation); these cases imply that one alternative has a broader range of outcomes than the other, and is therefore riskier in the sense that it is possible that the outcomes have wider dispersion. Such situations typically arise when a risky alternative that has a higher average outcome is compared with a safer choice that has lower probability of losses but also less opportunity for the highest upside outcomes. The optimal choice then depends on the decision-maker's risk appetite. In this context the model suggests that the market knowledge strategy is more-risky than the 5-year strategy, despite offering greater upside. The concept of Value at risk (VAR) refers to the amount that might be lost or that the target value might not be attained with a specified probability (de Neufville & Scholtes, 2011). Setting the probability of occurrence at the 5% level, the Five-year strategy outperforms the Market-Knowledge strategy. Observing the target curve in Figure 8 for the output of the simulation, it can be seen that there is approximately a 20% chance that the Market Knowledge strategy underperforms the five-year strategy, or stated another way, roughly an 80% chance that it outperforms the five-year strategy.

Reviewing Figure 9 below, we analyze the simulation results of the Market Knowledge strategy. The data indicates that the distribution of lease term durations exhibits positive skew. The bulk of the observations are shorter-term durations, with less than 10% of the outcomes resulting in longer-term (> 7-year) durations. As can be seen in Figure 9 below, roughly 50% of the outcomes were accounted for by 4-5-year lease durations. 5-year leases dominated the distribution of leases, accounting for approximately 29%, followed closely by 4-year leases, accounting for approximately 20% of the leases.

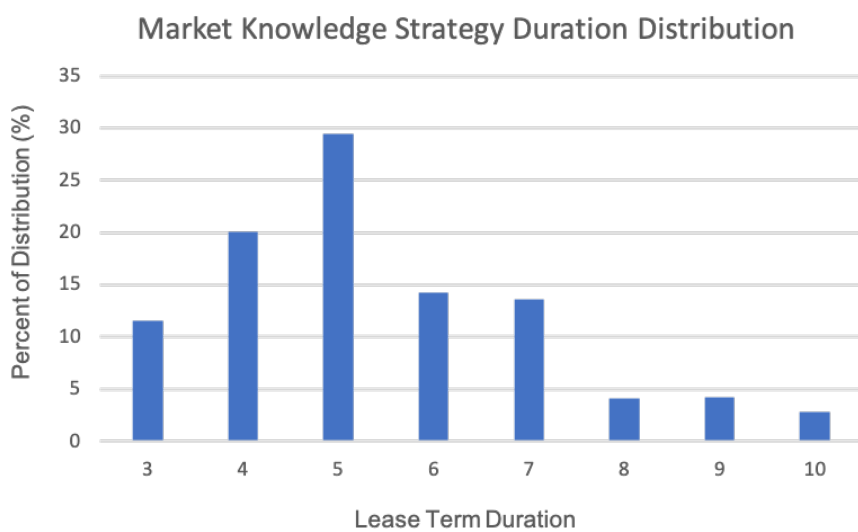

Figure 9: Market Knowledge Strategy Distribution

Drawing from the output of this simulation, the data suggests that landlord preference for longer duration (10-year) leases is not optimal. Landlords may actually be better off with shorter duration (5-year or variable) leases. In both cases, 5-year and Market Knowledge, the present value of each strategy outperforms the 10-year strategy, both having higher highs and higher lows. The 5-year strategy, which based on aforementioned data seems to be widely used in industry, is more conservative than the Market Knowledge strategy in that the upside potential is less than that of the Market Knowledge strategy, and has a higher low than the Market Knowledge strategy. The Market Knowledge strategy is slightly riskier, with a slightly lower low, while offering greater upside potential.

This exercise explored a heuristic strategy created using interpreted intuition about landlord and tenant preferences. In light of this, it is important to point out that the results are not empirical in that the model was not built by extrapolating historic data or using current data, therefore the results do not necessarily suggest that 4- or 5-year lease durations are optimal. The results do indicate that shorter-term lease durations maximize present value.

CONCLUSION

As property investors gradually embrace modern financial concepts it is clear that real estate valuation theory will have to change (Lucius, 2001). The model developed in this thesis provides a framework for users to explore alternative leasing strategies within the context of an uncertain future.

Valuation is the process of estimating price. Traditional methods used to determine value attempt to model the thought processes of the market and thus estimate price by reference to observed historic data. This information is utilized in the DCF valuation model to determine the single point valuation figure. However, the valuation will be affected by uncertainties: uncertainty in the comparable data available; uncertainty in the current and future market conditions and uncertainty in the specific inputs for the subject property. There are both exogenous and endogenous sources of uncertainty that cause volatility in real estate valuations. Managers can choose to act differently in response to circumstances over the course of the asset's useful lifetime, and this can affect the value of the investment.
These sources are many and difficult to predict. These input uncertainties will translate into an uncertainty with the output figure, the estimate of price. Uncertainty means that it is possible to have a range of different future scenarios in most real estate projects. The single-stream, DCF valuation model tends to hide this important fact.

Commercial property leases and leasing strategy are among the most fundamental, important, and complex topics in real estate investment and property management (Geltner, Miller, Clayton, & Eiccholtz, 2014). Understanding leases and leasing strategy is also central to many types of professional real estate careers. This thesis recognizes cash flow uncertainties explicitly, and employs leasing strategies to explore the impact of harnessing this uncertainty to advantage.

The DCF model is pervasive in the real estate industry. Business analysts and decision-makers worldwide use common spreadsheet programs, making the DCF a common language in the business and financial world, which greatly facilitates communication, transparency, understanding, and use. DCF models based on computer spreadsheets have tremendous range and flexibility in what they can do analytically, especially when considering uncertainty, and even more so in the world of big data and data analytics. Spreadsheets take in numerical data and calculate outputs, allowing users to easily change one or more entries and recalculate to see the results instantaneously.

Previously published research advocating for the use of probabilistic valuation techniques were missing data from a sufficient number of market cycles to describe the behavior of market factors and uncertainty. Relatively little empirical research has been done on commercial leases, largely because of a lack of large-scale, detailed databases on such leases. Real estate has been perceived as less sophisticated compared to other asset classes such as stocks and bonds. This perception is largely due to the private nature of real estate transactions and the lack of data available for economic analysis. Without reliable data to guide financial decisions, real estate professionals have depended on their instincts and intuition. As the 21st century unfolds, it is the age of big data. Big data and the increasing usage of data science is changing the way the real estate industry is functioning. From pricing estimates and valuation to marketing and leasing, the power of predictive analytics is improving the business processes and presenting new ways of operating (Park, 2020). There are vast areas of opportunity in applying data science to real estate.

As an interdisciplinary field, data science combines scientific learnings from statistics, advanced math, algorithms, and modeling. Incorporated with business knowledge, it can find patterns and consequently meaningful information from large sets of data. In this process, data scientists utilize econometrics to explain the data set's causality by testing hypotheses. Machine learning is used to predict the future by learning the patterns observed in the past. Data scientists support decision-making processes based on past data and predict future outcomes (Park, 2020). In this light, the model presented in this thesis could be implemented utilizing real data to improve the modeling of the real estate cycle and identify better patterns to inform leasing strategies.

Real estate indices provide data relevant to different types of uncertainty. Specifically, transaction-based indices (TBI) use actual sales data of commercial real estate to track the market. TBIs are relatively young (early 2000s), but have great potential because the underlying transaction price data not only quantifies market volatility reflected in the indices themselves, but also the individual asset idiosyncratic uncertainty using the residuals of the price regressions (Geltner, Miller, Clayton, & Eiccholtz, 2014).

In light of the increasing availability of real estate data, future research is directed at exploring the use of real data feeding the simulation to make better models.

References

Avison Young. (2021, June 14). *The Data Bite: Longer Manhattan office term lengths are signaling a rebound*. Retrieved from Sightlines Thought Capital: https://www.avisonyoung.us/sightline/-/article/2021/06/14/longer-manhattan-office-term-lengths-are-signaling-a-rebound

Baumeister, R. F., Bratslavsky, E., Finkenauer, C., & Vohs, K. D. (2021). Bas is stronger than good. *Review of General Psychology*, p. 323.

Bond, S. A., Loizou, P., & McAllister, P. (2008). Lease Maturity and Initial Rent: Is There a Term Structure for UK Commercial Property Leases? *The Journal of Real Estate Finance and Economics, 36*, 451-468.

Booth, P. (1993). The move to floating exchange rates: prospect for inflation and the implications for property values. *Journal of Property Finance, 4*, 9-23.

Born, W., & Phyrr, S. (1994). Real estate valuation: the effect of market and property cycle. *The Journal of Real Estate Research, 9*, 455-85.

Byrne, P. (1984). *Risk, Uncertainty and Decision-Making in Property Development.* Spon Press.

Cardin, M.-A., Nuttall, W., de Neufville, R., & Dahlgren, J. (2007). Extracting Value from Uncertainty: A Methodology for Engineering Systems Design. *Systems Engineering: Key to Intelligence Enterprises INCOSE 2007 - 17th Annual International Symposium Proceedings.* INCOSE.

Ciochetti, L. M., & Fisher, B. A. (2007). *Determinants of the Terms and Performance of Office Leases.* Massachusetts Institute of Technology, Department of Urban Studies and Planning, and Center for Real Estate. Cambridge: MIT.

de Neufville, R., & Scholtes, S. (2011). *Flexibility in Engineering Design.* Cambridge: MIT Press.

Di Pasquale, D., & Wheaton, W. (1996). *Urban Economics and Real Estate Markets.* Englewood Cliffs: Prentice-Hall.

Eichholtz, P. M. (1997). A Long Run House Price Index: The Herengracht Index, 1628-1973. *Real Estate Economics, 25*(2), 175-192.

Fisher, J., Geltner, D., & Webb, R. B. (1994). Value indices of commercial real estate: A comparison of index construction methods. *The Journal of Real Estate Finance and Economics, 9*(2), 137-164.

Fisher, L. (2004). The Wealth Effects of Sales and Leasebacks: New Evidence. *Real Estate Economics, 32(4)*, 619-643.

French, N. (2005). Discounted cash flow: accounting for uncertainty. *Journal of Property Investment & Finance, 23*(1), 76-89.

Geltner, D. (2020, April). Quantifying & Simulating Future Value Uncertainty & Price Dynamics in Real Estate Assets.

Geltner, D., & de Neufville, R. (2015). Uncertainty, Flexibility, Valuation and Design: How 21st Century Information and Knowledge Can Improve 21st Century Urban Development (Part I of II). *Pacific Rim Property Research Journal*, 251-276.

Geltner, D., & de Neufville, R. (2018). *Flexibility and Real Estate Valuation Under Uncertainty: A Practical Guide for Developers.* Hoboken: John Wiley & Sons.

Geltner, D., Miller, N., Clayton, J., & Eiccholtz, P. (2014). Chapter 30: Leases and Leasing Strategy. In Geltner, Miller, Clayton, & Eiccholtz, *Commercial Real Estate Analysis and Investments (3rd Edition).* Mbition LLC.

Giaccotto, C., & Clapp, J. (1992). Appraisal-Based Real Estate Returns Under Alternative Market Regimes. *The Journal of American Real Estate and Urban Economics Association, 20*(1), 1-24.

Grenadier, S. (1995). Valuing Lease Contracts: A Real-Options Approach. *Journal of Financial Economics, 38(3)*, 297-331.

Grenadier, S. R. (1995). The Persistence of Real Estate Cycles. *Journal of Real Estate Finance and Economics, 10*, 95-119.

Grenadier, S. R. (2005). An Equilibrium Analysis of Real Estate Leases. *The Journal of Business, 78*(4), 1173-1214.

Hayman, A., & Ulrich, P. (1995). Strategies for negotiating leases in transitional real estate market. *Real Estate Finance Journal, 11*, 59-63.

Jalori, S. (2017). *Optimal phasing: a deterministic and probabalistic analysis of different real estate development profiles.* Cambridge: Massachusetts Institute of Technology.

John D. Benjamin, J. D. (1992). Security Deposits, Adverse Selection and Office Leases. *Journal of the American Real Estate and Urban Economics Association, 20(2)*, 259-272.

Joskow, P. (2004). Contact Duration and Relationship-Specific Investments: Empirical Evidence from Coal Markets. *American Economic Review, 77(1)*, 168-185.

K.J. Crocker, S. M. (1985). Efficient Adaptation in Long-Term Contracts: Take-or-Pay Provisions for NAtural Gas. *American Economic Review, 75(5)*, 1083-1093.

K.J. Crocker, S. M. (1988). Mitigating Contractual Hazards: Unilateral Options and Contract Length. *RAND Journal of Economics, 19(3)*, 327-343.

Kelliher, C. F., & Mahoney, L. S. (2000). Using Monte Carlo simulation to improve long-term investment decisions. *The Appraisal Journal, 68*, 44.

Leung, K. C.-K. (2014). *Beyond DCF Analysis in Real Estate Financial Modeling: Probabalistic Evaluation of Real Estate Ventures.* Cambridge: Massachusetts Institute of Technology.

Li, L. H. (2000). Simple Computer Applications Improve the Versatility of Discounted Cash Flow Analysis. *The Appraisal Journal, 68*(1), 86-92.

Lucius, D. I. (2001). Real options in real estate development. *Journal of Property Investment and Finance, 19*, 73-78.

Marshall, K., & Kennedy, C. (1992). Development Valuation Techniques. *Journal of Property Valuation and Investment, 4*(11), 57-66.

Martin, V. (1990). Reviewing Discounted Cash Flow Analysis. *The Appraisal Journal*, 83-87.

McConnell, J. S. (1985). A Model for the Determination of "Fair" Premiums on Lease Cancellation Insurance Policies. *Journal of Finance, 40(5)*, 1439-1457.

Merrill, T. W. (2020). The Economics of Leasing. *Journal of Legal Analysis, 12*, 221-272.

Mooradian, R., & Yang, S. (2000). Cancellation Strategies in Commerical Real Estate Leasing. *Journal of Real Estate Economics, 28(1)*, 65-88.

Mooradian, R., & Yang, S. (2002). Commercial Real Estate Leasing, Asymmetric Information, and Monopolistic Competition. *Journal of Real Estate Economics, 30(2)*, 293-315.

Mulherin, J. (1986). Complexity in Long-Term Contracts: An Analysis of NAtural Gas Contractual Provisions. *Journal of Law, Economics & Organization, 2(1)*, 105-117.

Park, S. J. (2020). *Data Science Strategies for Real Estate Development.* Cambridge: Massachusetts Institute of Technology.
Phyrr, S. (1973). A Computer Simulation Model to Measure the Risk in Real Estate Investment. *American Real Estate and Urban Economics Journal*, 48-78.
Robinson, J. (1999). Commercial lease terms and the property cycle. *Facilities, 17*(5/6), 177-182.
Rosinsky, A. (2018, January 31). *Short-Term vs. Long-Term Manhattan Commercial Leases*. Retrieved from Metro Manhattan Office Space: https://www.metro-manhattan.com/blog/short-term-vs-long-term-manhattan-commercial-leases/
Royal Institution of Chartered Surveyors. (2002). *The Carsberg Report.* Royal Institution of Chartered Surveyors.
Savage, S. L., Danziger, J., & Markowitz, H. (2012). *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty.* Wiley.
The Appraisal Foundation. (1999). *Uniform Standards of Professional Appraisal Practice.* Washington, D.C.: The Appraisal Foundation.
Tse, R. (1999). Lag vacancy, effective rents and optimal lease term. *Journal of Property Investment & Finance, 17*(1), 75-88.
Ward, P., & McCann, C. (2004). Real Estate Rental Payments: Application of Stock-Inventory Modeling. *The Journal of Real Estate Finance and Economics, 28(2/3)*, 273-292.
Weber, B. C. (1990). Market Value Without a Market. *The Appraisal Journal*, 523-532.
Wheaton, W. (1999). Real estate "cycles": some fundamentals. *Real Estate Economics, 27*(2), 209-230.
Wheaton, W. C. (2000). Percentage Rent in Retail Leasing: The Alignment of Landlord-Tenant Interests. *Journal of Real Estate Economics, 28(2)*, 185-204.
Wheaton, W. C., Baranski, M. S., & Templeton, C. A. (2009). 100 yeras of commercial real estate prices in Manhattan. *Real Estate Economics, 37*(1), 69-83.
Wheaton, W., Baranski, M., & Templeton, C. (2009). 100 yeras of commercial real estate prices in Manhattan. *Real Estate Economics, 37*(1), 69-83.
Williams, J. B. (1938). *The Theory of Investment Value.* Amsterdam, Netherlands: North Holland Publishing Company.

APPENDIX: CODE

```
module RESim
using Parameters
using ArgCheck
```

```julia
using TOML
using OrderedCollections
using Distributions
using Statistics
using Plots
using DataFrames
using DataFramesMeta
using CSV
using EmpiricalCDFs
using DataStructures


#----- parameters/cycle_parameters.jl -----#



# This approach to structuring and using model parameters inspired by
# https://discourse.julialang.org/t/model-configuration-parameterization-file/8982/6
#
# Also potentially relevant (for example, for asserts in constructor):
# https://mauro3.github.io/Parameters.jl/v0.9/manual.html

"""
Constraints for a single NestedSineCurve within a NestedSineCurveSeq, reflecting how
Real Estate markets behave. These constraints can be set in the config file.
"""
@with_kw struct CycleConstraints{C}
    # Rent Cycle Settings
    period_min::C = 10.0
    period_max::C = 20.0

    @assert period_min <= period_max
    @assert period_min > 0
    @assert period_max > 0
end


"Uncertainty and Dynamic Inputs to a *single* model (cells R1:V23)."
@with_kw struct CycleParameters{C}
    RentCycPer::C
    RentCycPhse::C

    ### Uncomment anything you want to add to the model ###
    # InitPriceFactor = 100.00
```

```
    # LRtrendMean = 0.0
    # InitialRentHalfRange = 0.0
    # TrendYearHalfRange = 0.0
    # VolatilityInput = 0.0
    # AutoRegTerm = 0.20
    # MeanRevertRate = 0.3
    RentCycAmpli::C = 0.5  # 50%
    # CapRcycAmpli = 0.0
    # CapRcycPhse = 0.0
    # CapRcycPer = 0.0
    # NoiseAmt = 0.0
    # BlkSwnProb = 0.0
    # BlkSwnEffect = -0.25
    # TerminalCapRate = 0.05  # value from proforma
    # Selling_Expense = 0.02
    # Riskfree_Rate = 0.03
    # ProFormaCFgroRateInput = 0.02
    # DiscRateInput = 0.07
    # GivenPropPrice = 1000  # value from proforma
end


"""
Initialize all CycleParameters, including random rent cycle period + phase, based on the
values in CycleConstraints.

If you want some kind of constant or distribution to be available to your
NestedSineCurveSeq, this is where it needs to be initialized and/or configured.
"""
function CycleParameters(ccs::CycleConstraints)
    @unpack_CycleConstraints ccs  # extract model constraints into local scope

    # Cycle periods
    period = period_min + rand() * (period_max - period_min)
    phase = rand() * period

    CycleParameters(
        RentCycPer = period,
        RentCycPhse = phase,
    )
end

#----- parameters/market_parameters.jl -----#
```

```
"""
Uncertainty and Dynamic Input settings for an entire RealEstateMarket model.

This is in contrast to CycleContraints and CycleParameters, which just affect individual
NestedSineCurves within a RealEstateMarket.
"""
@with_kw struct RealEstateMarketConstraints{D}
   # General market settings
   num_years::D = 50    # How many years total does the market run for?
end

@with_kw struct RealEstateMarketParameters
   num_years
end


"""
Initialize all RealEstateMarketParameters based on the settings in
RealEstateMarketConstraints.

If you want some kind of constant or distribution to be available to your RealEstateMarket,
this is where it needs to be initalized and configured.
"""
function RealEstateMarketParameters(remcs::RealEstateMarketConstraints)
   RealEstateMarketParameters(
      num_years = remcs.num_years,
   )
end

#----- parameters/simulation_parameters.jl -----#


@with_kw struct SimulationParameters{D}
   num_markets::D = 10000
   @assert num_markets > 0

   compute_num_wins_each_strategy::Bool = true
   plot_single_histogram_all_strategy_dcfs::Bool = true
   plot_cumulative_target_curve_dcfs::Bool = true
   record_leaselength_distbns::Bool = true
   record_dcf_summary_statistics::Bool = true
end
```

```
#----- sine_curve.jl -----#


"""A SineCurve has an amplitude, period > 0, phase, and (vertical) shift."""
struct SineCurve
   amp::Float64
   period::Float64
   phase::Float64
   shift::Float64

   function SineCurve(amp, period, phase, shift)
      @argcheck period > 0
      new(amp, period, phase, shift)
   end
end


"""
   SineCurve(amp=2, period=20, phase_frac=0.5, shift=2)

Create a SineCurve for which phase = period * phase_frac.
"""
function SineCurve(; amp, period, phase_frac, shift)
   @argcheck 0 <= phase_frac && phase_frac <= 1
   phase = period * phase_frac
   SineCurve(amp, period, phase, shift)
end


"""
   fval(sc, t)    # sc(t)

Compute value of captured sine curve at time t.
"""
function fval(sc::SineCurve, t::Real)
   sc.shift + sc.amp * sin((t - sc.phase) * (2*œÄ / sc.period))
end


"""
   fvals(sc, ts)    # [sc(t) for t in ts]

Compute values of captured sine curve at times ts.
```

```
"""
function fvals(sc::SineCurve, ts)
    [fval(sc, t) for t in ts]
end

"Nice way to print out a SineCurve."
function Base.string(sc::SineCurve)
    "SineCurve(amp=$(sc.amp), period=$(sc.period), phase=$(sc.phase), shift=$(sc.shift))"
end

# Define more properties, accessible via dot notation
function Base.getproperty(sc::SineCurve, v::Symbol)
    if v == :phase_frac
        sc.phase / sc.period
    else
        getfield(sc, v)
    end
end


"""
    Plots.plot(sc, ts)

Plot points [(t, sc[t]) for t in ts] using the Plots library with sensible defaults.
"""
@recipe function f(sc::SineCurve, ts)
    vals = fvals(sc, ts)

    # Set plot defaults: https://docs.juliaplots.org/latest/recipes/#Recipe-Syntax/Rules
    legend --> false
    title  --> string(sc)

    # Return the xs and ys to be plotted
    ts, vals
end

#----- nested_sine_curve.jl -----#


################
# CONSTRUCTORS #
################
"A composition of two SineCurves with the same period and phase."
struct NestedSineCurve
```

```
    outer::SineCurve
    inner::SineCurve

    NestedSineCurve(outer, inner) = begin
        @argcheck outer.period == inner.period && outer.phase == inner.phase
        new(outer, inner)
    end
end

function NestedSineCurve(ampouter, ampinner, period, phase, shiftouter, shiftinner)
    outer = SineCurve(ampouter, period, phase, shiftouter)
    inner = SineCurve(ampinner, period, phase, shiftinner)
    NestedSineCurve(outer, inner)
end

"""
    NestedSineCurve(ccs::CycleConstraints[, phase_frac])

Given a set of CycleConstraints, generate a randomized NestedSineCurve.
Cells AP1:AT4 in the spreadsheet.

If a phase_frac is provided, then phase = phase_frac * period.
Otherwise, the phase will be a random fraction of the period, like in the spreadsheet.

The ability to specify the phase_frac is crucial to the generation of leastCommonMultiple
NestedSineCurves that smoothly transition from one to the next, which is how we
generate a NestedSineCurveSeq.
"""
function NestedSineCurve(ccs::CycleConstraints, phase_frac=nothing)
    @argcheck 0 < phase_frac && phase_frac <= 1
    cps = CycleParameters(ccs)
    @unpack_CycleParameters cps  # dump CycleParameters fields into local scope

    # Convert dumped model inputs into args for outer and inner
    period = RentCycPer
    SpPer1 = period
    SpPer2 = period

    SpAmp1 = RentCycAmpli
    SpAmp2 = SpPer2 / 10

    phase = isnothing(phase_frac) ? RentCycPhse : period * phase_frac
    SpPhs1 = phase
    SpPhs2 = phase
```

```
    # Shift isn't defined in spreadsheet - let's center around 1 == 100%
    SpShift1 = 1
    SpShift2 = 0

    outer = SineCurve(SpAmp1, SpPer1, SpPhs1, SpShift1)
    inner = SineCurve(SpAmp2, SpPer2, SpPhs2, SpShift2)
    NestedSineCurve(outer, inner)
end


#################
# FUNCTIONALITY #
#################
"""
Compute value of captured sine curves at time t.
"""
function fval(nsc::NestedSineCurve, t::Real)
    o, i = nsc.outer, nsc.inner
    o.shift + o.amp * sin((t - o.phase - fval(i, t)) * (2*œÄ/o.period))
end

"""
Compute values of captured sine curves at times ts.
"""
function fvals(nsc::NestedSineCurve, ts)
    [fval(nsc, t) for t in ts]
end


# Define more properties, accessible via dot notation
function Base.getproperty(nsc::NestedSineCurve, v::Symbol)
    if v == :period
        """
        Mathematically, period(nsc) = leastCommonMultiple(period(outer), period(inner)).

        Since both the inner and outer NestedSineCurves constructors guarantee that
        period(outer) == period(inner) for all NestedSineCurves,
        period(nsc) is just equal to the period of either outer or inner.
        """
        @argcheck nsc.outer.period == nsc.inner.period
        return nsc.outer.period
    elseif v == :phase_frac
        """
```

```
    What fraction of the period is the phase?

    This would ordinarily be poorly defined, but since the NestedSineCurve constructors
    guarantee that `outer` and `inner` have the same period and phase, we can compute it
    using the period and phase of either `outer` or `inner`.
    """
    @argcheck nsc.outer.period == nsc.inner.period
    @argcheck nsc.outer.phase == nsc.inner.phase
    return nsc.outer.phase / nsc.outer.period

  elseif v == :amp
    """
    At the end of the day, a NestedSineCurve takes the form A * sin(...) + D,
    where A = nsc.outer.amp and D = nsc.outer.shift
    """
    return nsc.outer.amp
  elseif v == :shift
    """
    At the end of the day, a NestedSineCurve takes the form A * sin(...) + D,
    where A = nsc.outer.amp and D = nsc.outer.shift
    """
    return nsc.outer.shift
  elseif v == :phase
    """
    Get the phase of a NestedSineCurve.
    Since the phase of the outer and inner curves is guaranteed to be equal, the phase
    of a NestedSineCurve will be equal to the phase value of both outer and inner.
    """
    return nsc.outer.phase
  else
    getfield(nsc, v)
  end
end


"""
  Plots.plot(nsc, ts)

Plot points [(t, nsc[t]) for t in ts] using the Plots library with sensible defaults.
"""
@recipe function f(nsc::NestedSineCurve, ts)
  vals = fvals(nsc, ts)

  # Set plot defaults: https://docs.juliaplots.org/latest/recipes/#Recipe-Syntax/Rules
```

```
    @unpack outer, inner = nsc
    legend --> false
    title  --> "Nested Sine Curve, period=$(round(nsc.outer.period, digits=3))"

    # Return the xs and ys to be plotted
    ts, vals
end

#----- nested_sine_curve_seq.jl -----#


"""
A sequence of NestedSineCurves that smoothly transition from one to the next, such that
the sum of the periods of the NestedSineCurves in the sequence is >= `duration`.

To achieve the smooth transition mathematically, we must:
1. Ensure that each NestedSineCurve has a phase equal to the same fraction of its period.
2. Offset the times for all NestedSineCurves after the first one by the sum of all the
   periods of the previous NestedSineCurves when computing nscseq[t].

#1 must be handled by `nsc_maker`, which takes `nsc_maker_pos_args...` and
`nsc_maker_kw_args...` as input and should be able to consistently generate
NestedSineCurves with the same phase_frac as output.

#2 is explained and handled by Base.getindex(nscseq::NestedSineCurveSeq) below.
"""
struct NestedSineCurveSeq
    transition_times::Vector{Float64}
    duration::UInt16
    nscs::Vector{NestedSineCurve}

    NestedSineCurveSeq(nsc_maker, duration,
                    nsc_maker_pos_args... ; nsc_maker_kw_args...) = begin
        @argcheck duration > 0

        # Generate enough NestedSineCurves so that sum(periods) >= duration
        nscs = []
        period_so_far = 0.0
        while period_so_far < duration
            nsc = nsc_maker(nsc_maker_pos_args...; nsc_maker_kw_args...)
            push!(nscs, nsc)
            period_so_far += nsc.period
        end
        @assert length(nscs) >= 1
```

```
    # Generate the transition_times
    transition_times = [0.0]
    append!(transition_times, cumsum([nsc.period for nsc in nscs]))

    # Make sure everything looks good
    if (length(nscs) > 1)
      @assert all([nsc.phase_frac ,âà nscs[1].phase_frac for nsc in nscs[2:end]])
    end
    @assert length(transition_times) == length(nscs) + 1

    new(transition_times, duration, nscs)
  end
end


"""
Compute value of captured piecewise function at time t.

The keys to correctly computing fval(nscseq::NestedSineCurveSeq, t) are:
1. Selecting the correct NestedSineCurve from nscseq.nscs
  (calculated in variable `nsc_index` below)
2. Adjusting t so that it's relative to the start of that correct NestedSineCurve
  (calculated in variable `t_adj` below)

To see this in practice, consider nscseq::NestedSineCurveSeq with nscseq.nscs = [
  nsc1 == NestedSineCurve(period=18),
  nsc2 == NestedSineCurve(period=12)
], with both NestedSineCurves shifted so that they start at their max values.

For any time t ,àà [0, 18], nsc_index = 1 and we don't need to adjust t at all - just
return fval(nsc1, t).

To calculate fval(nscseq, 19), though, we want nsc_index = 2.
And we can't just call fval(nsc2, 19) because nsc2 will be more than halfway through its
second cycle at t=19 due to its period of 12, meaning that there won't be a smooth
transition from nsc1 (returning to its max value at t=18) to nsc2
(more than halfway through its second cycle at t=19 due to its shorter period).

Instead, fval(nscseq, 19) == fval(nsc2, 19-18) == fval(nsc2, 1), where 18 is nsc1's period.
And in general, for t ,àà [18, 30] fval(nscseq, t) == fval(nsc2, t-18).

Thankfully, transition_times contains all the information we need to quickly compute
(a) the relevant NestedSineCurve, and (b) the adjusted time to use in that NestedSineCurve.
```

```
"""
function fval(nscseq::NestedSineCurveSeq, t::Real)
    lookback_cutoff = -1
    @argcheck lookback_cutoff <= t && t <= nscseq.duration
    @unpack transition_times, duration, nscs = nscseq

    transition_time_index = max(findfirst(ttime -> t <= ttime, transition_times) - 1, 1)
    nsc_index = transition_time_index
    t_adj = t - transition_times[transition_time_index]

    @assert 1 <= transition_time_index && transition_time_index <= length(transition_times)
    @assert 1 <= nsc_index && nsc_index <= length(nscs)
    @assert t_adj >= lookback_cutoff

    nsc = nscs[nsc_index]
    fval(nsc, t_adj)
end


"""
Compute values of captured sine curves at times ts.

Could make this more efficient in the case where many of the ts values are all relevant
to the same NestedSineCurve.
"""
fvals(nscseq::NestedSineCurveSeq, ts) = [fval(nscseq, t) for t in ts]


# Define more properties, accessible via dot notation
function Base.getproperty(nscseq::NestedSineCurveSeq, v::Symbol)
    if v == :period
        "Not defined - the period changes from NSC to NSC."
        error("a NestedSineCurveSeq doesn't have a period! rethink your use case")
    elseif v == :phase_frac
        """
        What fraction of the period is the phase?
        Since smooth transitions are guaranteed by all NestedSineCurves having the same
        phase_frac, this is well defined.
        """
        nscs[1].phase_frac
    elseif v == :shift
        """
        If every NestedSineCurve in the NestedSineCurveSeq has the same vertical shift k,
        then the NestedSineCurveSeq will also have vertical shift k.
```

```
    Otherwise, error!
    """
    shifts = [nsc.shift for nsc in nscseq.nscs]
    if all(k -> k == shifts[1], shifts)
      shifts[1]
    else
      error("""
        This NestedSineCurveSeq's NestedSineCurves have different vertical shifts,
        so its vertical shift is not well-defined. Rethink your use case.
        """)
    end
  else
    getfield(nscseq, v)
  end
end


"""
  Plots.plot(nsc, ts)

Plot points [(t, nsc[t]) for t in ts] using the Plots library with sensible defaults.
"""
@recipe function f(nscseq::NestedSineCurveSeq, ts)
  vals = fvals(nscseq, ts)

  # Set plot defaults: https://docs.juliaplots.org/latest/recipes/#Recipe-Syntax/Rules
  legend --> false
  title  --> "Nested Sine Curve Sequence, $(nscseq.duration) Years Total"

  # Return the xs and ys to be plotted
  ts, vals
end

#----- re_space_market.jl -----#



"""
A RESpaceMarket is a NestedSineCurveSeq and a duration.

Note that the duration of the underlying NestedSineCurveSeq is actually twice as long as the
duration of the RESpaceMarket. This is so that, if you sign a 10-year lease in year 48 and
you don't lock in the market's price_factor for the entire lease duration, you actually have
a NestedSineCurve to query for values in years 48, ..., 57.
```

"Twice as long" might be overkill, but all of these operations are pretty cheap.

Access to these years will still be validated by MarketInProgress's get_spacemarket_val.

```
"""
struct RESpaceMarket
    nscseq
    num_years

    RESpaceMarket(remcs::RealEstateMarketConstraints, ccs::CycleConstraints) = begin
        # Convert Market constraints into actual parameters
        remps = RealEstateMarketParameters(remcs)
        num_years = remps.num_years

        # Create a NestedSineCurveSeq that spans remps.num_years
        nscseq_duration = num_years * 2    # underlying NSCSeq is twice as long!
        nscseq_phase_frac = rand()
        nscseq = NestedSineCurveSeq(NestedSineCurve, nscseq_duration,
                            # pass positional args to NestedSineCurve constructor
                            ccs, nscseq_phase_frac)

        new(nscseq, num_years)
    end

    RESpaceMarket() = new(RealEstateMarketConstraints(), CycleConstraints())
end

"""The length, in years, of the RESpaceMarket."""
duration(spacemarket) = spacemarket.num_years

"""Get the value of `spacemarket` at time t."""
fval(spacemarket::RESpaceMarket, t) = fval(spacemarket.nscseq, t)

"""Get the value of `spacemarket` at times ts."""
fvals(spacemarket::RESpaceMarket, ts) = fvals(spacemarket.nscseq, ts)

"""
    Plots.plot(spacemarket::RESpaceMarket, ts)

Plot the real estate market over its entire run.
"""
@recipe function f(spacemarket::RESpaceMarket, ts)
    vals = fvals(spacemarket, ts)
```

```
    # Set plot defaults: https://docs.juliaplots.org/latest/recipes/#Recipe-Syntax/Rules
    legend --> false
    title  --> "RESpaceMarket"

    # Return the xs and ys to be plotted
    ts, vals
end

#----- market_in_progress.jl -----#

      sign_lease!,
      ismipdone,
      isincreasing,
      currenttime,
      duration

"""
A MarketInProgress bundles up one or more full Real Estate Market components, along with a
"current time" current_t. Users of a MarketInProgress (mainly Strategies) are able to access
the information contained in the Real Estate Market components for any time 0 ,â§ t ,â§
current_t, but cannot see past current_t.

To move current_t forward and gain access to more information, use `sign_lease!`.

To add an additional piece of bundled information,
1. Add it to the MarketInProgress struct
2. Create a getter that verifies its input and then forwards the call along
   (ala get_spacemarket_val)
3. If you want the new bundled info to appear in the graph, change the @recipe.
"""
mutable struct MarketInProgress
    # Unchanging components
    spacemarket::RESpaceMarket

    # Changing components
    current_t

    # Constructors
    MarketInProgress(spacemarket::RESpaceMarket) = begin
        new(spacemarket, 0)
    end

    MarketInProgress(spacemarket::RESpaceMarket, t) = begin
        @assert 0 <= t && t < duration(spacemarket)
```

```
    new(spacemarket, t)
  end

  MarketInProgress(remcs::RealEstateMarketConstraints, ccs::CycleConstraints) = begin
    spacemarket = RESpaceMarket(remcs, ccs)
    new(spacemarket, 0)
  end
end

duration(mip::MarketInProgress) = duration(mip.spacemarket)
currenttime(mip::MarketInProgress) = mip.current_t
Base.copy(mip::MarketInProgress) = MarketInProgress(mip.spacemarket, currenttime(mip))

function get_spacemarket_val(mip::MarketInProgress)
  fval(mip.spacemarket, currenttime(mip))
end

"""
Shouldn't actually check to make sure that t <= duration, since if you sign a
10-year lease in year 48/50 and you're not locking in the market price_factor for
the entire duration of the lease, you'll need market values at years 51, ..., 57.
"""
function get_spacemarket_val(mip::MarketInProgress, t)
  @argcheck -1 <= t && t <= currenttime(mip)  # -1: at t=0, can peek back a bit
  fval(mip.spacemarket, t)
end


function sign_lease!(mip::MarketInProgress, lease_length::Integer)
  @argcheck !ismipdone(mip)
  @argcheck lease_length > 0
  mip.current_t += lease_length
  return nothing
end

function ismipdone(mip::MarketInProgress)
  currenttime(mip) >= duration(mip)
end

function isincreasing(mip::MarketInProgress)::Bool
  œµ = 0.001
  currentval = get_spacemarket_val(mip)
  pastval = get_spacemarket_val(mip, currenttime(mip) - œµ)
  currentval > pastval
```

```
end

function isabovemeanrent(mip::MarketInProgress)::Bool
   get_spacemarket_val(mip) > mip.spacemarket.nscseq.shift
end

"""
   Plots.plot(mip)

Plot all information up to current_t.
"""
@recipe function f(mip::MarketInProgress; t_int=0.5)
   ts = 0 : t_int : duration(mip)
   spacemarket_vals = [get_spacemarket_val(mip, t) for t in ts]

   # Set plot defaults: https://docs.juliaplots.org/latest/recipes/#Recipe-Syntax/Rules
   legend --> false
   title  --> "MarketInProgress, current_t=$(current_t)"

   # Return the xs and ys to be plotted
   ts, spacemarket_vals
end

#----- strategy.jl -----#

      get_strats,
      fnname,
      lengths2times

"""
Generally speaking, a strategy is a function: strat :: MarketInProgress ->
   Distributions.DiscreteProbDistribution

A strategy function is allowed to use any visible information in the MarketInProgress to
generate its DiscreteProbDistribution, which should capture the durations of leases it would
consider signing, plus the probability of signing a lease of that particular duration.

To define a new strategy called <fn>, you should:
1. Create a boolean in StrategyParameters called use_<fn>. This will generate an entry in
  blank config files and get parsed when reading config files.
2. Modify get_strats below to include an if...end for stratparams.use_<fn>. This will mean
  that the strategy is included in the simulation based on the value of use_<fn> from the
  config file.
3. Define a new function <fn> :: MarketInProgress -> DiscreteProbDistribution below. In
```

```
  particular, the distribution it returns should respond to Base.rand() by generating an
  integer leaselength from its distribution.
"""


"""
StrategyParameters indicates which strategies should be used in a simulation. It
should be updated whenever a new strategy is defined, so that the option to use
the new strategy appears in config files.
"""
@with_kw struct StrategyParameters
    use_strat_five_year = true
    use_strat_ten_year = true
    use_strat_market_knowledge = true
    use_strat_triangular = false
end


"""
Return a Vector of the strategies specified in StrategyParameters, so that the
Simulator can iterate through them.
"""
function get_strats(stratparams::StrategyParameters)
    strats = []
    if stratparams.use_strat_five_year
        push!(strats, strat_five_year)
    end
    if stratparams.use_strat_ten_year
        push!(strats, strat_ten_year)
    end
    if stratparams.use_strat_market_knowledge
        push!(strats, strat_market_knowledge)
    end
    if stratparams.use_strat_triangular
        push!(strats, strat_triangular)
    end
    strats
end

####################
# HELPER FUNCTIONS #
####################

"""Get the name of a function dynamically."""
```

```
function fnname(fn)
  String(Symbol(fn))
end

"""
Convert a vector of leaselengths a strategy signed into the times at which those leases
were signed. Essentially the cumulative sum, with a 0 at the beginning and the last one
dropped
"""
function lengths2times(leaselengths)
  append!([0], cumsum(leaselengths[1:end-1]))
end


##############
# STRATEGIES #
##############
"""strat_five_year always signs a five-year lease."""
function strat_five_year(mip::MarketInProgress)
  Dirac(5)
end


"""strat_ten_year always signs a ten-year lease."""
function strat_ten_year(mip::MarketInProgress)
  Dirac(10)
end

"""
strat_market_knowledge uses two factors to determine lease distribution:
1. Is the market currently increasing or decreasing?
2. Are market rents currently above or below mean rent?

Note that these lease durations are from the landlord's perspective:
- if the market's low, sign a shorter lease to wait for higher rents
- if the market's high, sign a longer lease to lock in higher rents
"""
function strat_market_knowledge(mip::MarketInProgress)
  if isabovemeanrent(mip) && isincreasing(mip)
    # landlord: medium lease, lock in higher prices but maybe catch the top
    return DiscreteNonParametric([  5,    6,    7  ],
                                 [3//10, 4//10, 3//10])
  elseif isabovemeanrent(mip) && !isincreasing(mip)
    # market crashing -> sign long lease!
```

```
        return DiscreteNonParametric([  7,     8,     9  ,  10  ],
                              [2//10, 3//10, 3//10, 2//10])
    elseif !isabovemeanrent(mip) && isincreasing(mip)
        # recovering from bottom -> shorter lease, higher rents ahead
        return DiscreteNonParametric([  3,     4,     5],
                              [2//10, 4//10, 4//10])
    else    # !isabovemeanrent(mip) && !isincreasing(mip)
        # on the way towards bottom -> medium lease, skip the bottom
        return DiscreteNonParametric([  3,     4,     5  ],
                              [3//10, 4//10, 3//10])
    end
end


"""
Fake triangular distribution!
"""
function strat_triangular(mip::MarketInProgress)
    return DiscreteNonParametric([  4,     5,      6,      7,      8  ],
                           [1//10, 2//10,  4//10,  2//10,  1//10])
end

#----- dcf_variable.jl -----#

      total_dcf

"""
Parameters that control DCF calculations.

The parameters `discount_rate_increases` and `lease_length_cutoffs_lt` work as follows:

For example, if  discount_rate_increases = [0.015, 0.01, 0.005]
           and lease_length_cutoffs_lt = [  4  ,   7],

then after a signing a lease for n years,
the amount to increase the discount rate by for the next lease will be:
if n < 4
    r += 0.015
elseif n < 7
    r += 0.01
else
    r += 0.005
"""
@with_kw struct DCFParameters{C,D,E}
```

```
    lock_in_price_factor_for_lease_duration::Bool = true
    base_rent::E = 100

    initial_discount_rate::C = 0.07

    discount_rate_increases::Vector{C} = [0.01, 0.005]
    lease_length_cutoffs_lt::Vector{D} = [5]
    @assert length(discount_rate_increases) == 1 + length(lease_length_cutoffs_lt)
end


"""
How much should the discount rate increase for the next lease?
Depends on DCFParameters.discount_rate_increases and
DCFParameters.lease_length_cutoffs_lt.
"""
function r_inc(dcfparams::DCFParameters, leaselength)
    @unpack_DCFParameters dcfparams
    index = findfirst(cutoff -> leaselength < cutoff, lease_length_cutoffs_lt)
    if isnothing(index)
        index = length(discount_rate_increases)
    end
    discount_rate_increases[index]
end


"""
Compute the discounted cash flow (DCF) for a single lease, where
  r = discount rate
  rent = how much will be paid each year
  leaselength = how many years payments will be generated (dur in formula below)
  pricefactors = market values at each year of the lease (pfs in formula below)

Then
  DCF = rent*pfs[1] / (1+r)^1 +  ... + rent*pfs[dur] / (1+r)^dur
      = rent * [pfs[1] / (1+r)^1 + ... + pfs[dur] / (1+r)^dur
"""
function lease_dcf(r, rent, leaselength, pricefactors)
    @argcheck length(pricefactors) == leaselength
    sum(rent .* [pricefactors[i] / (1+r)^i for i in 1:leaselength])
end


"""
```

```
Get the pricefactors for a `lease_length`-year lease signed at time `year_signed` during
MarketInProgress `mip`, either (locking in the starting market value for the entire lease
duration) or (using each year's market value) as controlled by
`dcfparams.lock_in_price_factor_for_lease_duration`.
"""
function get_pricefactors(dcfparams, mip::MarketInProgress, year_signed, leaselength)
  if dcfparams.lock_in_price_factor_for_lease_duration
    fill(get_spacemarket_val(mip, year_signed), leaselength)
  else
    years = year_signed : (year_signed + leaselength - 1)
    [get_spacemarket_val(mip, year) for year in years]
  end
end


"""
Compute the total discounted cash flow (DCF) for a series of leases `leaselengths` according
to the parameters in `dcfparams`, assuming the market had annual values `pricefactors`.
"""
function total_dcf(dcfparams::DCFParameters, leaselengths, mip::MarketInProgress)
  @argcheck ismipdone(mip)
  @unpack_DCFParameters dcfparams

  leasetimes = lengths2times(leaselengths)
  total = 0.0
  r = initial_discount_rate
  for (leaselength, year_signed) in zip(leaselengths, leasetimes)
    pricefactors = get_pricefactors(dcfparams, mip, year_signed, leaselength)
    total += lease_dcf(r, base_rent, leaselength, pricefactors)
    r += r_inc(dcfparams, leaselength)
  end
  total
end

#----- config_files.jl -----#

    parseconfigfile,
    SimulatorInputs


"""
The master list of configuration structs.

If you want a struct to be configurable via configuration files, include it here,
```

in the order you'd like it to appear in the configuration file (and make sure it's defined @with_kw and with default values for its fields).

This is used by createblankconfiginfo to generate a blank configuration file with the appropriate sections and default values.

It's also used by parseconfigfile to ensure that the parsed configuration file contains all the sections required to start a simulation.
"""
```
const configstructs = [
    CycleConstraints,
    RealEstateMarketConstraints,
    DCFParameters,
    StrategyParameters,
    SimulationParameters,
]
```

"""
A SimulatorInputs contains all information needed to run a Simulation.
The members of a SimulationSettings should mirror the contents of `configstructs`.
A SimulationInputs is generated upon successful parsing of a configuration file.

Assumes: config is a Dict with keys that include the names of the configstructs
constructors: "CycleConstraints", "ProFormaParameters", etc. By far the easiest way to obtain such a Dict is to parse a valid configuration file using `parseconfigfile`. Or use the external SimulatorInputs constructor.

To add another field to an existing input, just modify the relevant struct:
- CycleConstraints if you want to control something about individual real estate cycles,
- DCFParameters if you want to add another parameter to the DCF calculation, etc.
Generally speaking, you can use:
- field::C for a continuous-valued number (decimals)
- field::D for a discrete-valued number (nice round number: years, length, etc.)
- field::E for a value that you'd like to work for either decimals or discrete values

To add an entirely new type of input to the Simulator,
1. Define a @with_kw struct in a file somewhere in /src
2. Make sure the file containing your new struct is `include`d by RESim.jl
3. Add your new struct to configstructs above
4. Add a new field to SimulationInputs that will hold your new input
5. Add a new line to the constructor that initialized that new field
"""
```
struct SimulatorInputs
```

```
    ccs::CycleConstraints
    remcs::RealEstateMarketConstraints
    dcfparams::DCFParameters
    stratparams::StrategyParameters
    simparams::SimulationParameters

    SimulatorInputs(config) = begin
        ccs = CycleConstraints(; config["CycleConstraints"]...)
        remcs = RealEstateMarketConstraints(; config["RealEstateMarketConstraints"]...)
        dcfparams = DCFParameters(; config["DCFParameters"]...)
        sps = StrategyParameters(; config["StrategyParameters"]...)
        simparams = SimulationParameters(; config["SimulationParameters"]...)
        new(ccs, remcs, dcfparams, sps, simparams)
    end
end

"""Create a SimulatorInputs directly from the path of a configuration file."""
function SimulatorInputs(configpath::AbstractString)
    config = parseconfigfile(configpath)
    SimulatorInputs(config)
end


"""
Convert `configstructs` into an OrderedDict mapping structnames to their default values.
This OrderedDict (called 'maindict' below) ends up looking like:
{
    :CycleConstraints -> {
        :period_min -> 10.0,
        :period_max -> 20.0,
    },
    :RealEstateMarketConstraints -> {
        :num_years -> 50
    },
    ...
}
"""
function createblankconfiginfo()
    maindict = OrderedDict()
    for ctor in configstructs
        defaultinstance = ctor()  # used just for field names
        d = Dict(key=>getfield(defaultinstance, key) for key ,àà fieldnames(ctor))
        maindict[nameof(ctor)] = d
    end
```

```
    maindict
end


"""
Generate a blank configuration file for a simulator run by writing all configuration
structs in configstructs to a .toml file, including their default values.

Assumes: `path` is a valid path that ends in .toml
         a file doesn't already exist at `path`
"""
function genblankconfigfile!(path)
    @assert !isfile(path)
    @assert path |> endswith(".toml")
    configdata = createblankconfiginfo()
    open(path, "w") do io
        TOML.print(io, configdata)
    end
end


"""
Convert a configuration file into a SimulationSettings instance containing
all specified Simulation settings.

The easist way to create a valid configuration file is to run main.jl with
the "gen" argument, then tweak the resulting generated config file.
"""
function parseconfigfile(path)
    @assert isfile(path)
    @info "Parsing $path..."

    configunordered = try
        TOML.parsefile(path)
    catch e
        if isa(e, TOML.ParserError)
            @error "Error parsing $path - please check file structure and try again."
        end
        rethrow()
    end

    # Make sure the parsed data has at least the `configstructs` sections
    for ctor in configstructs
        key = String(nameof(ctor))
```

```
        if !(key in keys(configunordered))
            error("Section for $key not found in $path. Please fix.")
        end
    end

    # Convert String keys into Symbols for constructor splatting purposes
    config = Dict(key => Dict(Symbol(innerkey) => innerval
                    for (innerkey, innerval) in val)
              for (key, val) in configunordered)

    @debug "Read the following config file contents from $path: " config
    config
end

#----- simulation.jl -----#

       runsimulation,
       runtrial,
       analyze_simulation_results,
       getdcfs

"""
Time to run some simulations!
"""

#################################
# SINGLE TRIAL: MIP x STRATEGY #
#################################
"""Run strategy `strat` through a MarketInProgress, signing leases along the way."""
function runtrial(mip::MarketInProgress, strat)::Vector{UInt8}
    @assert currenttime(mip) == 0
    leaselengths::Vector{UInt8} = []
    while !ismipdone(mip)
        # Generate a lease duration
        leasedistbn = strat(mip)
        leaselength = rand(leasedistbn)
        push!(leaselengths, leaselength)

        # Sign the lease, moving currenttime(mip) forward `leaselength` years
        sign_lease!(mip, leaselength)
    end
    leaselengths
end
```

```
###############################################################
# FULL SIMULATION: MANY RANDOM MARKETS x ALL STRATEGIES #
###############################################################
"The results of a full simulation, for later analysis."
struct SimulationResults
   stratdcfs
   leaselength_counts
   trial1info
end

getdcfs(simresults::SimulationResults) = simresults.stratdcfs
getleaselengthdistbn(simresults::SimulationResults) = simresults.leaselength_counts
gettrial1info(simresults::SimulationResults) = simresults.trial1info

"""
The main simulation function: run all strategies specified in `simsettings.stratparams`
across the same `simparams.num_markets` random markets, where the randomness of each
market is controlled by `simparams.ccs` and `simparams.remcs`.
"""
function runsimulation(simsettings::SimulatorInputs)
   @info "Running Simulation with SimulationInputs:" simsettings

   # Unpack useful things from simsettings
   numtrials = simsettings.simparams.num_markets
   strategies = get_strats(simsettings.stratparams)

   # Make data structures to keep track of the results
   totaldcfs = Dict(fnname(s) =>
                 Vector{Float64}(undef, numtrials) for s in strategies)
   leaselength_counts = Dict(fnname(s) => counter(Integer) for s in strategies)

   trial1info = Dict()

   for trial in 1:numtrials
      # Generate a random market based on the constraints in simsettings
      randommarket = MarketInProgress(simsettings.remcs, simsettings.ccs)

      for strat in strategies
         market = copy(randommarket)
         leaselengths = runtrial(market, strat)

         # Total DCF
         dcf = total_dcf(simsettings.dcfparams, leaselengths, market)
```

```
            totaldcfs[fnname(strat)][trial] = dcf

            # Add leaselengths from this trial to the master counter
            trial_leaselength_counts = counter(leaselengths)
            merge!(leaselength_counts[fnname(strat)], trial_leaselength_counts)

            if trial == 1
                trial1info[fnname(strat)] = leaselengths
                if !haskey(trial1info, "market")
                    trial1info["market"] = market
                end
            end
        end
    end

    @assert length(trial1info) == length(strategies) + 1
    SimulationResults(DataFrame(totaldcfs),
                      leaselength_counts,
                      trial1info)
end

################################
# ANALYZE SIMULATION RESULTS #
################################

"""
The main analysis function - calls the sub-analysis functions specified in `siminputs`
to analyze the simulation results contained in `results`, writing any relevant output
to `results_folderpath`.
"""
function analyze_simulation_results(results::SimulationResults,
                                    results_folderpath,
                                    siminputs::SimulatorInputs)

    if (siminputs.simparams.compute_num_wins_each_strategy)
        write_num_wins_each_strategy(results, results_folderpath)
    end

    if (siminputs.simparams.plot_single_histogram_all_strategy_dcfs)
        plot_single_histogram_all_strategy_dcfs(results, results_folderpath)
    end

    if (siminputs.simparams.plot_cumulative_target_curve_dcfs)
        plot_cumulative_target_curve_dcfs(results, results_folderpath)
```

```
    end

    if (siminputs.simparams.record_leaselength_distbns)
        write_leaselength_freq(results, results_folderpath)
    end

    if (siminputs.simparams.record_dcf_summary_statistics)
        write_dcf_summary_statistics(results, results_folderpath)
    end

    plot_trial1_leases(gettrial1info(results), results_folderpath)
end

"""
After Trial #1 of each simulation, the MarketInProgress and each strat's
lease lengths are saved so that the various strategies can be visualized.
Make a plot of each strat's lease lengths.
"""
function plot_trial1_leases(trial1info, results_folderpath)
    market::MarketInProgress = trial1info["market"]
    @argcheck ismipdone(market)
    delete!(trial1info, "market")

    numstrats = length(trial1info)  # one entry per strat

    markersize = 8
    markershapes = fill(:circle, numstrats)
    colors = [:red :green :blue :orange :black :yellow :pink]

    plots = []
    for (i, (stratname, leaselengths)) in enumerate(trial1info)
        plt = Plots.plot(market; title=stratname)
        shape = markershapes[i]
        color = colors[i]

        xs = lengths2times(leaselengths)
        ys = [get_spacemarket_val(market, t) for t in xs]
        Plots.scatter!(plt, xs, ys,
                        shape=shape,
                        color=color,
                        markersize=markersize)
        push!(plots, plt)
    end
```

```
    stacked = plot(plots..., layout=(length(trial1info), 1))
    pngpath = joinpath(results_folderpath, "trial1leases.png")
    png(stacked, pngpath)
end


"""
Create a Dict : strategy name -> the number of markets it had the greatest DCF for
"""
function count_num_wins_each_strategy(dcfs::DataFrame)
    # Makes a new dataframe containing a single column: the winning strategy
    winning_strats = select(dcfs, AsTable(:) => ByRow(argmax) => :strat)

    # DataFramesMeta approach - wasn't sure how to do argmax across all columns?
    # @chain dcfs begin
    #     @select(dcfs, :winner = argmax(:))
    # end

    # Group by :strat value
    groups = groupby(winning_strats, :strat)

    # Get the length of each group
    counts = combine(groups, :strat => length)

    # Convert to a Dict: strat -> count
    symdict = Dict(Pair.(counts.strat, counts.strat_length))

    # We want Strings, not Symbols
    Dict(String(sym) => val for (sym, val) in symdict)
end

function write_num_wins_each_strategy(results::SimulationResults, outdir)
    counts = count_num_wins_each_strategy(getdcfs(results))
    counts_df = DataFrame(strat= [key for key in keys(counts)],
                          numwins = [val for val in values(counts)])
    @transform!(counts_df, percent = :numwins / sum(values(counts)) * 100)
    sort!(counts_df, [:numwins], rev=true)
    outname = joinpath(outdir, "num_wins.csv")
    CSV.write(outname, counts_df)
end

function write_dcf_summary_statistics(results::SimulationResults, outdir)
    dcfs = getdcfs(results)
    summary_statistics = describe(dcfs, :mean, :std, :min, :q25, :median, :q75, :max)
```

```
    outname = joinpath(outdir, "dcf_summary_statistics.csv")
    CSV.write(outname, summary_statistics)
end

function plot_single_histogram_all_strategy_dcfs(results::SimulationResults, outdir)
    dcfs = getdcfs(results)
    numstrats = length(names(dcfs))

    minvalue = describe(dcfs)[!, "min"] |> minimum  # smallest dcf ever seen
    maxvalue = describe(dcfs)[!, "max"] |> maximum  # largest dcf ever seen
    hists = [histogram(dcfs[:, i],
                    title=name,
                    xlim=(minvalue, maxvalue),
                    label=false) for (i, name) in enumerate(names(dcfs))]

    stacked = plot(hists..., layout=(numstrats, 1))

    pngpath = joinpath(outdir, "stacked_strat_dcf_histogram.png")
    png(stacked, pngpath)
end


function plot_cumulative_target_curve_dcfs(results::SimulationResults, outdir)
    dcfs = getdcfs(results)
    numstrats = length(names(dcfs))

    # Create an empirical CDF for each strategy's DCF values
    cdfs = [EmpiricalCDF() for _ in 1:numstrats]
    for (i, col) in enumerate(eachcol(dcfs))
        append!(cdfs[i], col)
        sort!(cdfs[i])
    end

    # Create the xs to evaluate each CFD at
    minvalue = describe(dcfs)[!, "min"] |> minimum  # smallest dcf seen, all strats
    maxvalue = describe(dcfs)[!, "max"] |> maximum  # largest dcf seen, all strats
    increment = 50
    padding = .20
    xs = minvalue*(1-padding) : increment : maxvalue*(1+padding)

    # Get CDF values for each strategy's results
    V = hcat([cdf(xs) for cdf in cdfs]...)

    # Make a plot!
```

```
  colors = [:red :green :blue :orange :black :yellow :pink]
  plt = Plots.plot(xs, V,
            title="Cumulative Target Curve",
            xlab="DCF", ylab="Cumulative Proportion",
            # Line info
            lw=3,
            # marker=:auto,
            # Legend Information
            label=reshape(names(dcfs), :, numstrats),
            labelspacing=1,
            legend=:bottomright,
            color=colors
            )

  # Add dotted vertical lines at each strategy mean
  stratmeans = describe(dcfs)[!, "mean"]
  stratmeans = reshape(stratmeans, 1, numstrats)
  vline!(plt, stratmeans,
      lw=2, style=:dash,
      label="",   # don't add entries to the legend
      color=colors
      )
  Plots.display(plt)

  pngpath = joinpath(outdir, "cumulative_target_curve_dcfs.png")
  png(plt, pngpath)
end

function write_leaselength_freq(results::SimulationResults, outdir)
  leaselength_counts = results.leaselength_counts
  for (fnname, counts) in leaselength_counts
    leaselengths = Vector{UInt8}()
    freqs = Vector{Integer}()
    for (ll, freq) in counts
      append!(leaselengths, ll)
      append!(freqs, freq)
    end
    counts_df = DataFrame(leaselength = leaselengths, count = freqs)
    @transform!(counts_df, percent = :count / sum(freqs) * 100)
    sort!(counts_df, [:leaselength])
    outname = joinpath(outdir, fnname * "_leaselength_freqs.csv")
    CSV.write(outname, counts_df)
  end
end
```

```
"""
Helper functions for main.jl, mostly related to filesystem.
"""

##########################
# main.jl GEN helpers #
##########################

"""
Extract optional argument [<argkey>] from parsed arguments `parsed`, or return
`default` if the user didn't provide the optional argument.
"""
function getoptionalarg(parsed, argkey, default)
  argval = parsed[argkey]
  isnothing(argval) ? default : argval
end


"""
Get the optional <runname> that the user passed after 'gen', or return 'blank'
if no <runname> was provided.
"""
function extract_newconfigfilename(parsedargs, argkey="<runname>", default="blank")
    runname = getoptionalarg(parsedargs, argkey, default)

    # Do any cleaning of the provided argument here
    while endswith(runname |> lowercase, ".toml")
      runname = chop(runname, tail=5)
    end

    configfilename = runname * ".toml"
    configfilename
end


"""
Throw an error if there's a file at `path` to ensure that we don't overwrite an existing
config file.
"""
function error_if_file_exists(path)
  if isfile(path)
    error("File $path already exists; please rename/remove and rerun.")
  end
```

```
end


#########################
# main.jl RUN helpers #
#########################

"""
Make sure that a top-level 'runs' folder exists to store the results of runs.
"""
function ensurerunsdir!(runspath)
  if !endswith(runspath, "/")
    runspath = runspath * "/"
  end

  if !isdir(runspath)
    @info "No runs folder found at $runspath. Creating folder to store run results..."
    mkdir(runspath)
    @assert isdir(runspath)
  end
  return runspath
end


"""
Extract the <configpath> provided after 'run'; error if file doesn't exist!
Note that *something* will have been provided: required argument.
"""
function extract_configfilepath(args, argkey="<configpath>")
  configpath = args[argkey]
  if !isfile(configpath)
    error("File $configpath not found; pls check existence/spelling and try again.")
  end
  if !(configpath |> lowercase |> endswith(".toml"))
    error("Configuration file must end with '.toml'; $configpath does not.")
  end
  configpath
end

"""
Extract the name of the folder that should be created to hold the results of the run.
Matches the name of the configuration file that the user provided.
"""
function extract_runname(configfilepath)
```

```
  @assert configfilepath |> lowercase |> endswith(".toml")
  _, file = splitdir(configfilepath)
  chop(file, tail=length(".toml"))
end

"""
Throw an error if there's a folder at `path` to ensure that we don't overwrite an existing
runs result filder.
"""
function error_if_folder_exists(path)
  if isdir(path)
    error("Folder $path already exists; please rename/remove and rerun.")
  end
end
using Pkg
Pkg.activate(".")

# Pkg.add("DocOpt")
using DocOpt
using Logging

include("./src/RESim.jl")
using .RESim

include("main_helpers.jl")
include("export_code.jl")

doc = """Running the Real Estate Simulation.

Usage:
  main.jl gen [<runname>] [--overwrite]
  main.jl run <configpath> [--overwrite]
  main.jl export
  main.jl -h | --help

Options:
  -h --help          Show this help message.
  -o --overwrite     Overwrite config file or run output folder if it exists. IGNORED
"""

function main()
  args = docopt(doc)
  @debug "Command line options: " args
```

```
runspath = "runs/"
runspath = ensurerunsdir!(runspath)

# Generate a blank configuration file
if args["gen"]
  newconfigfilename = extract_newconfigfilename(args)
  newconfigfilepath = joinpath(runspath, newconfigfilename)
  error_if_file_exists(newconfigfilepath)
  @info "Generating new configuration file at $newconfigfilepath..."
  genblankconfigfile!(newconfigfilepath)

# Run a simulation using specified configuration file
elseif args["run"]
  # Extract and verify full configuration file path from parsed args
  configfilepath = extract_configfilepath(args)

  # Extract the runname == config file name without ".toml" at the end
  runname = extract_runname(configfilepath)
  results_folderpath = joinpath(runspath, runname)

  # Make sure a results folder doesn't already exist; create it
  error_if_folder_exists(results_folderpath)
  mkdir(results_folderpath)

  # Copy configuration file to results folder
  config_copy_path = joinpath(results_folderpath, runname * ".toml")
  cp(configfilepath, config_copy_path)

  # Run the simulation!
  siminputs = SimulatorInputs(configfilepath)
  @time results = runsimulation(siminputs)
  println("Completed $(siminputs.simparams.num_markets) market simulations")

  # Analyze the results
  println("Analyzing results...")
  analyze_simulation_results(results::SimulationResults,
                             results_folderpath,
                             siminputs::SimulatorInputs)

  return results

# Export all used src code from the project into a single file, for appendix purposes
elseif args["export"]
  println("Exporting all relevant source code!")
```

```
    export_code()
  end
end

# useful if you run the simulation in interactive mode via -i flag:
# $ julia -i main.jl run runs/MYRUN.toml
results = main()

if !isnothing(results)  # results is nothing when generating new config file
  dcfs = getdcfs(results)
end

end # module RESim
```