جزوه شماره ۱۴

DATA SCIENCE IN PRACTICE

Data Visualization with Python

محمدطاهر طاهرپور

دانشجوی ارشد مدیریت شهری

دانشگاه تهران

mttaherpoor
@mtaherpoor
09336144947

ناهید نعمتی کوتنائی (تیسا)

دکتری جغرافیا و برنامه‌ریزی شهری

مدرس دانشگاه

Dr.nemati.K
@Nemati_k
09112230798

پاییز ۱۴۰۲

فهرست مطالب:

Data Science in Practice

تو این جزوه کاربرد data Science یا علم داده رو روی چند تا پروژه شـــهری مثل تحلیل مســـکن، تحلیل حمل و نقل و فضـــای ســـبز شـــهری با هم کار میکنیم و از کتابخونه‌هایی که تا اینجا یاد گرفتیم یعنی numpy, pandas, matplotlib, seaborn, squarify و missingno استفاده میکنیم.

کل این جزوه رو با کمک هوش مصنوعی Poe (ترکیبی از ChatGPT, GPT_4, Sage و Claude) نوشتیم. کاری که انجام دادیم پرسیدن یه سری سوالات متوالی و هدفمند بود و در نهایت کدهایی که بهمون داد رو بردیم تو نوت بوک ArcGIS pro و نمودارهاش رو استخراج و وارد جزوه کردیم.

از ســایت Kaggle بخش datasets هم یه فایل برای کار کردن با دســتورها دانلود کردیم و یکی از تحلیلهای جزوه رو بر اساس اون فایل نوشتیم. ۱۶ تا هوش مصنوعی جذاب هم تو همین جزوه معرفی شده.

منابع: ســایت https://poe.com/ برای نوشــتن محتوای جزوه و ســایت https://www.kaggle.com/datasets برای دانلود داده‌های تمرینی.

اول یه توضیح کلی در مورد هوش مصنوعی می‌دیم و بعدش وارد پروژه‌مون می‌شیم.

Artificial Intelligence هوش مصنوعی چیه؟

هوش مصنوعی (Artificial Intelligence) به مجموعه‌ای از تکنیک‌ها و روش‌های کامپیوتری گفته می‌شه که به کامپیوترها و ســیســتم‌های کامپیوتری این امکان رو میده که شــبیه انســان یاد بگیرن، برداشــت، نتیجه‌گیری، تفکر و تصمیم‌گیری کنن. هدف اصلی هوش مصنوعی اینه که سیستم‌هایی ایجاد کنه که بتونن الگوها رو تشخیص بدن، وظایف هوشمندانه انجام بدن و مسائل پیچیده رو حل کنن. در طول سال‌ها، هوش مصنوعی توسعه‌های چشمگیری پیدا کرده و ازش تو خیلی از برنامه‌های کاربردی اســتفاده شــده مثل ماشــینهای خودران، ســیســتم‌های پشــتیبانی از تصمیم‌گیری، تجزیه و تحلیل داده‌ها، ترجمه ماشینی و سیستم‌های پاسخگویی صوتی مثل سیری و آلکسا.

رویکردهای مختلفی در هوش مصــنوعی وجود داره که شــامل هوش مصــنوعی ضــعیف (یا محدود) و هوش مصنوعی قوی (عمومی) میشه:

۱. **هوش مصــنوعی ضــعیف (Weak AI):** در این رویکرد، ســیســتم‌های هوش مصــنوعی وظایفی مثل بازی‌های کامپیوتری، تشــخیص صــدا، ترجمه زبان و غیره رو انجام میدن. این ســیســتم‌ها براســاس الگوریتم‌ها و قوانین قبلی طراحی می‌شن و قدرت هوشی محدودی دارن.

۲. **هوش مصنوعی قوی (Strong AI):** هدف این رویکرد ایجاد سیستم‌هایی با هوش مشابه انسان هست که قادر به درک و تفسیر عمیق مسائل، خودآگاهی، احساسات و تفکر خلاقانه باشه. هنوز هوش مصنوعی قوی به اندازه کافی تحقق پیدا نکرده و خیلی از محققان در این زمینه به دنبال راه‌حل‌های نوآورانه هســتن. این رویکردهای خلاقانه شــامل موارد زیر میشه:

- **شــبکه‌های عصــبی عمیق (Deep Neural Networks):** یکی از مهمترین روش‌های اســتفاده شــده در هوش مصــنوعی قوی هســتن که از شــبکه‌های عصــبی در مغز انســان الهام گرفته شــدن و از طریق آموزش با داده‌های بزرگ یا Big data، قادر به تشــخیص الگوها و انجام وظایف پیچیده مثل تشــخیص تصــاویر و ترجمه متون هستن.
- **یادگیری تقویتی (Reinforcement Learning):** منظور اســتفاده از روشــی هســت که عامل هوشــمند با تعامل با محیط، به صــورت آزمون و خطا، یاد می‌گیره که چطوری تصــمیم‌های بهتری بگیره تا به هدف

خاصی برسه. این رویکرد در حال حاضر در حوزه‌هایی مثل بازی‌های کامپیوتری و رباتیک پیشرفت‌های قابل توجهی داشته.

- **یادگیری عمیق تقویتی (Deep Reinforcement Learning):** ترکیبی از شبکه‌های عصبی عمیق و یادگیری تقویتی هست. در این رویکرد، شبکه‌های عصبی عمیق به عنوان تقریب‌گر عملکرد (تحلیل داده‌ها، پیش‌بینی نتایج ممکن و تصمیم‌گیری بر اساس اطلاعات موجود) و یادگیری از تجربه عمل می‌کنن. این روش برای حل مسائل پیچیده و چالش‌برانگیزی مثل بازی‌های ویدئویی پیشرفت‌های قابل توجهی داشته.
- **یادگیری تقویتی عمیق معناشناسی:** این رویکرد روی ایجاد مدل‌های زبانی عمیق تمرکز داره که قادر به درک و تفسیر معنا و مفهوم جملات و متون هست. این رویکرد در تحلیل زبانی پیشرفته، پرسش و پاسخ مبتنی بر متن و تولید متون طبیعی مورد استفاده قرار می‌گیره.
- **معماری‌های ذهنی مصنوعی:** این معماری‌ها هنوز در مراحل اولیه تحقیقات و توسعه قرار دارن و هنوز به حد نهایی نرسیدن. با این حال قرار هست که معماری‌های ذهنی مصنوعی با هدف شبیه‌سازی ساختار و عملکرد مغز انسان به منظور شکل دادن به هوش مصنوعی قوی استفاده بشه مثل شبکه‌های عصبی مصنوعی موازی، شبکه‌های عصبی بازتابی، معماری‌ها و الگوریتم‌های الهام‌گرفته از حافظه‌های طولانی کوتاه‌مدت (LSTM)، شبکه‌های عصبی بازگشتی (RNN)، معماری‌های توجه، مدل‌های برپایه حافظه (Memory-based models) و مدل‌های برپایه دانش (Knowledge-based models). هدف این معماری بهبود قدرت تفکر، تعامل و خلاقیت هوش مصنوعی موجود در حال حاضر و افزایش قابلیت‌هاش هست.

ابزارهای هوش‌های مصنوعی پرکاربرد:

۱- **هوش مصنوعی ChatGPT:** مسئله هوش مصنوعی بیشتر با این ابزار سر و صدا کرد. توسط شرکت OpenAI توسعه پیدا کرد و میشه هر سوالی رو ازش پرسید و باهاش بحث و گفتگو کرد. یعنی یه سری سوال به صورت پیوسته ازش میپرسید که نکات مبهم رو براتون روشن کنه. ولی هنوز محدودیتهایی هم داره و ممکنه به مسائل پیچیده و موضوعاتی که به آخرین تحقیقات علمی نیاز دارن پاسخهای دقیقی نده. برای ایران تحریم هست. نسخه ChatGPT 4 رایگان نیست.

۲- **دستیار هوش مصنوعی چندپلتفرمی Sage:** مثل ChatGPT توسط OpenAI توسعه داده شده. این دستیار هوش مصنوعی بر اساس معماری GPT-3.5 ایجاد شده و توانایی پرسش و پاسخ در موضوعات مختلف رو داره. ازش سوال بپرسید بهتون جواب میده، براتون ترجمه هم میکنه. **چندپلترفمی** بودنش این مزیت رو بهش میده که در محیطهای مختلفی مثل وبسایت‌ها، اپلیکیشن‌ها، مسنجرها و بسیاری دیگر از پلتفرم‌ها میشه ازش استفاده کرد.

۳- **هوش مصنوعی claude2:** رقیب چت جی پی تی هست و توسط شرکت انتروپیک توسعه داده شده. داده‌هاش بروز هست و از فایلهای Pdf پشتیبانی میکنه. باهاش میشه مکالمات طولانی‌تر داشت و هزینه‌اش هم نسبت به ChatGPT پایین‌تر هست.

۴- **هوش مصنوعی Poe:** این هوش مصنوعی معروفترین هوشهای مصنوعی دنیا رو با هم ادغام کرده و ترکیبی از ChatGPT, GPT_4, Sage. Claude هست. در مورد هر موضوعی میتونید ازش سوال بپرسید و با جزئیات دقیق و کامل بهتون پاسخ میده (رایگان هست).

۵- **هوش مصنوعی paraphrasetool و neurodub:** باهاش میشه هر فیلم و عکس یا تصاویری رو به ۱۰۰ زبان دنیا ترجمه کرد. در واقع کاملترین ابزار ترجمه هست.

۶- **هوش مصنوعی Galactica:** نقش یه استاد و برات بازی میکنه و اصطلاحات و مسائل علمی رو مثل ریاضی و فیزیک یا هر علم دیگه‌ای رو راحت و کامل برات توضیح میده. در واقع از داشتن معلم خصوصی بی نیازت میکنه.

۷- **هوش مصنوعی Scribblediffusion:** برای بچه‌ها عالیه. هر نقاشی و خط خطهای ساده‌ای که بکشن رو براشون تبدیل به تصاویر واقعی میکنه. خیلی جذاب و سرگرم کننده هست.

۸- **هوش مصنوعی cancerfactfinder.org:** روشی هست که دانشگاه هاروارد راه انداخته. میتونید هر چیزی که شک دارید رو سرچ کنید و بر اساس منابع معتبر و رسمی بهتون میگه که سرطان‌زا هست یا نه.

۹- **هوش مصنوعی Perplexity.ai:** این هوش مصنوعی علاوه بر اینکه جواب سوالاتتون رو میده منابع معتبر رو هم بهتون معرفی میکنه. یعنی بهتون میگه که برای مطالعه تحقیقی به چه چیزهایی نیاز دارید و به کجا باید مراجعه کنید. اطلاعاتش هم آپدیت هست و از فارسی هم پشتیبانی میکنه.

۱۰- **هوش مصنوعی Stocking:** واسه کسب و کار مفید هست. به راحتی میتونید لوگو، پوستر، بنر، جلد کتاب یا هر کار گرافیکی که نیاز داشتید رو به راحتی براتون انجام بده.

۱۱- **هوش مصنوعی vidig.com:** یه سایت عالی واسه کسب درآمد از یوتیوب هست (واسه اینستاگرام هم میشه ازش استفاده کرد) یا یه فرمان از شما، عنوان محتوا، اسکریپت، توضیحات، کلمات کلیدی، تصاویر کوچک و صدا رو به راحتی تولید میکنه. فقط کافیه ایده رو تایپ کنید.

۱۲- **هوش مصنوعی hsabtra.vercel.app:** فارسی باهاش حرف بزنید براتون به انگلیسی تایپ میکنه.

۱۳- **افزونه MaxAI.me:** یک افزونه آنلاین قدرتمند مبتنی بر هوش مصنوعی هست. ویژگی‌های اصلیش شامل امکان مدیریت و بهبود بهره‌وری در انجام وظایف روزمره آنلاین، بهبود تصمیم‌گیری‌ها و انجام وظایف بهتر، سازگار بودن با مرورگرهای معروف مثل Chrome و Firefox، بهبود عملکرد و کارایی کاربران میشه.

۱۴- **هوش مصنوعی Pdf.ai:** ابزاری نوآورانه است که برای خلاصه کردن فایلهای pdf استفاده می‌شه. به این شکل که میشه یک کتاب الکترونیک یا یه متن مقاله چند صفحه‌ای رو به این هوش مصنوعی داد که به سرعت خلاصه خوبی از کل محتوا ارائه میده. بعدش میشه هر سوالی رو از این هوش مصنوعی پرسید و برای پاسخ به سوالات در داخل همون فایل pdf جستجو انجام میده. خود سیستم هم سوالاتی رو برای پرسش به کاربر پیشنهاد می‌ده. به عبارتی از اسناد حقوقی گرفته تا گزارش‌های مالی، این ابزار به کاربران این امکان رو میده تا سوال بپرسن، اطلاعات حیاتی رو استخراج کنن و حتی خلاصه‌های مختصر دریافت کنن.

۱۵- **هوش مصنوعی Yomu.ai:** برای نوشتن پایان نامه و مقاله ازش استفاده میشه. بهش عنوان میدی و برات مقدمه مینویسه. میتونی بهش بگی که متن رو به صورت آکادمیک برام بنویس. یا متن رو برام خلاصه یا کوتاه کن. یا اینکه متن رو بازنویسی کن. حتی مقاله‌ات رو برات داوری هم میکنه. نتیجه‌گیری مقاله رو هم برات می‌نویسه.

۱۶- **هوش مصنوعی researchrabbit.** برای استخراج منابع و مقالات در نوشتن پایان نامه و رساله ازش استفاده میشه. در واقع یه سایت هست به اسم www.researchrabbitapp.com که توش ثبتنام میکنی و یه فایل pdf بهش میدی و بر اســـاس اون فایل برات ســـرچ میکنه و کلی منابع با کیفیت و معتبر تو همین حیطه رو بهت معرفی میکنه. علاوه بر اون توی چند تا گراف جذاب نحوه ارتباط این مقالات رو هم بهت نشون میده.

برای نوشتن این جزوه که کاربرد علم داده در پروژه‌های شهری هست، از هوش مصنوعی **Poe** استفاده شده.

کاربرد علم داده در مطالعات شهری

با استفاده از کتابخونه‌های Numpy, pandas, matplotlib.pyplot, seaborn, squarify و missingno

- **NumPy**: یک کتابخونه بنیادی برای محاسبات ریاضی و کار کردن با آرایه‌ها در پایتون هست. در پروژه برنامه‌ریزی شهری می‌تونی از NumPy برای **انجام محاسبات عددی، تغییرات آرایه‌ها و مدیریت داده‌های ناقص** استفاده کنی.
- **Pandas**: یک کتابخانه محبوب برای تبدیل و تحلیل داده هست. ساختارهای داده‌ای مثل DataFrame داره که بهت امکان ذخیره و تغییر یا دستکاری داده‌های جدولی رو میده. از Pandas می‌توانی **برای بارگیری، تمیز کردن، تبدیل و تحلیل داده‌های برنامه‌ریزی شهری** استفاده کنی. این کتابخونه خیلی خوب با کتابخونه‌های دیگه مثل NumPy و Matplotlib هماهنگی داره.
- **Matplotlib**: یک کتابخونه شماتیک برای تولید نمودارهای استاتیک، متحرک و تعاملی در پایتون هست. می‌تونی از Matplotlib برای **تجسم جوانب مختلف داده‌های برنامه‌ریزی شهری مثل توزیع مکانی، روندها و همبستگی‌ها** استفاده کنی. این کتابخانه امکانات زیادی برای ایجاد نمودارهای با کیفیت بالا و انعطاف‌پذیر و تنظیمات سفارشی داره.
- **Seaborn**: یک کتابخونه تجسم داده‌های آماری هست که روی Matplotlib ساخته شده. این کتابخانه یک رابط سطح بالاتر برای ایجاد نمودارهای آماری جذاب و اطلاعاتی فراهم میکنه. Seaborn **انواع نمودارهای داخلی و پالت‌های رنگی رو که واسه مصورسازی داده‌های شهری بهت میده.** واسه اینکه روابط و الگوها رو تو داده‌هات پیدا کنی مفید هست.
- **Squarify** یک کتابخونه هست که به طور مشخص برای ایجاد نمودارهای درختی طراحی شده. این کتابخونه بهت این امکان رو میده که با استفاده از مستطیل‌های تو در تو، **توزیع داده‌های مختلفی مثل استفاده از زمین، تراکم جمعیت و یا هر داده سلسله مراتبی دیگه رو تو برنامه‌ریزی شهری مصورسازی کنی.** میتونی با این کتابخونه نمودارهای درختی با رنگ‌ها و برچسب‌های سفارشی ایجاد کنی.
- **Missingno**: یک کتابخونه برای تجسم الگوهای داده‌های گم‌شده در مجموعه داده هست. بهت کمک می‌کنه تا مقادیر گم‌شده رو شناسایی کنی و توزیع اونها رو در متغیرهای مختلف درک کنی. در برنامه‌ریزی شهری، **داده‌های گم‌شده ممکنه در زمینه‌های مختلفی مثل جمعیت، زیرساخت یا داده‌های محیطی اتفاق بیفته**. Missingno نمودارهای مفیدی مثل نمودارهای میله‌ای و نمودارهای حرارتی رو برات ایجاد می‌کنه تا بهت کمک کنه با داده‌های گم‌شده به خوبی روبرو بشی.

برای شروع کار با این کتابخانه‌ها، باید اونها رو با استفاده از دستورات زیر نصب کنی:

```
pip install numpy pandas matplotlib seaborn squarify missingno
```

یا اینکه برای نصب کتابخونه‌ها جزوه شماره ۹ رو ببینی. حالا نوت بوک ArcGIS Pro رو باز کن و پروژه‌های زیر رو گام به گام توش انجام بده.

پروژه ۱: تحلیل مسکن

تحلیل داده‌ها درباره مسکن در یه شهر میتونه بهمون کمک کنه که الگوها و ویژگی‌های مختلف در بازار مسکن رو درک کنیم. واسه تحلیلش از روشهای زیر استفاده میشه:

۱. **تجزیه و تحلیل آماری:** از روش‌های آماری مختلف واسه شناخت ویژگی‌های مسکن مثل قیمت، متراژ، تعداد اتاق‌ها و ... استفاده هیمشه. میانگین، میانه، واریانس و هیستوگرام‌ها ابزارهای مفیدی هستن که میشه برای درک بهتر *توزیع و توالی داده‌ها* ازشون استفاده کرد.

۲. **نقشه‌برداری:** با استفاده از روش‌های نقشه‌برداری، میشه الگوها و توزیع مکانی مسکن در شهر رو بررسی کرد. مثلا موقعیت مسکن‌ها رو روی نقشه شهر نشون داد و الگوهای مکانی مرتبط با قیمت، مراکز خدماتی، مسیرهای حمل و نقل و سایر عوامل رو براش بررسی کرد.

۳. **تجزیه و تحلیل عاملی:** با استفاده از روش تجزیه و تحلیل عاملی، میشه ویژگی‌های مختلف مسکن رو تحلیل کرد و الگوهای مرتبط با اون رو شناخت. مثلا، میتونی عوامل مهمی مثل موقعیت جغرافیایی، امکانات نزدیکی، ویژگی‌های مسکن و ... رو مشخص کنی و تأثیر اونها رو روی قیمت و بقیه معیارها بررسی کنی.

۴. **تحلیل خوشه‌بندی:** با استفاده از روش‌های خوشه‌بندی میشه مسکن رو به گروه‌های مشابه تقسیم کرد. این روش کمک می‌کنه که الگوهای مشابه در مسکن رو شناسایی کنیم و مسکن رو بر اساس ویژگی‌هایی مثل اندازه، موقعیت و ... دسته‌بندی کنیم.

۵. **پیش‌بینی قیمت:** با استفاده از مدل‌های پیش‌بینی، میشه قیمت مسکن رو بر اساس ویژگی‌های مختلف پیش‌بینی کرد.

این روش‌ها تنها چند نمونه از روش‌های مختلف تحلیل داده هستن که میشه برای تحلیل مسکن در یک شهر ازش استفاده کرد. مهمترین و اولین قدم، جمع‌آوری داده‌های مناسب و کامل هست. بعدش می‌تونی از ابزارها و تکنیک‌های مختلف تحلیل داده استفاده کنی تا مسکن در شهر رو به طور جامع بررسی کنی و الگوها و ارتباطات مختلف رو شناسایی کنی.

در ادامه می‌خوایم از کتابخونه‌های پایتون برای تحلیل مسکن استفاده کنیم. تو مرحله اول نیاز به یه سری داده داری که تو یه جدول با فرمت xls یا csv ذخیره شده باشن. بعدش باید داده‌هات رو تو یه DataFrame pandas بارگیری و تمیزشون کنی و روشون *پیش‌پردازش داده* انجام بدی. با Missingno با *داده‌های گم‌شده* روبرو بشی. بعدش میتونی داده‌ها رو با استفاده از Matplotlib و Seaborn تبدیل به نمودار کنی. با Squarify هم میشه *نمودارهای درختی* ایجاد کرد. مراحل کار به شکل زیر هست.

مرحله اول: وارد کردن کتابخونه‌ها به نوت بوک ArcGIS Pro

تو مرحله اول باید کتابخونه‌ها رو با دستور import وارد کنی. یادت باشه که عبارت جادیی رو هم بنویسی که نمودارهایی که میکشی تو همون صفحه نوت بوکت نمایش داده بشن.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
import missingno as msno
%matplotlib inline
```


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
import missingno as msno
%matplotlib inline
```

مرحله دوم: مرتب کردن داده‌های پروژه در جدول اکسل و ذخیره داده‌ها به فرمت xls یا csv.

یه جدول از سایت Kaggle بخش dataset برداشتیم.

https://www.kaggle.com/datasets/fedesoriano/the-boston-houseprice-data

این جدول اطلاعاتی در مورد قیمت خونه در بوستن داره. میتونی از هر جدول اطلاعاتی دیگه‌ای هم استفاده کنی. اطلاعات جدولیش رو به اسم boston.csv تو فایلهای تمرینی براتون گذاشتیم. این جدول ۱۴ تا ستون با محتوای زیر داره:

۱) CRIM: نرخ سرانه جرم بر اساس شهر

۲) ZN: نسبت زمین‌های مسکونی پهنه‌بندی شده برای قطعات بیش از ۲۵۰۰۰ فوت مربع.

۳) INDUS: نسبت هکتارهای تجاری غیرخرده‌فروشی در هر شهر

۴) CHAS: متغیر ساختگی رودخانه چارلز (۱ اگر مسیر به رودخانه محدود میشه؛ ۰ در غیر این صورت)

۵) NOX: غلظت اکسید نیتریک (قسمت در هر ۱۰ میلیون) [parts/10M]

۶) RM: میانگین تعداد اتاق در هر خانه

۷) AGE: نسبت واحدهای تحت اشغال ساخته شده قبل از ۱۹۴۰

۸) DIS: فاصله‌های وزنی تا پنج مرکز استخدامی بوستون

۹) RAD: شاخص دسترسی به بزرگراه‌های شعاعی

۱۰) TAX: نرخ مالیات بر دارایی تمام ارزش به ازای هر ۱۰۰۰۰ دلار [$/10K]

۱۱) PTRATIO: نسبت دانش‌آموز به معلم بر اساس شهر

۱۲) B: نتیجه معادله B=1000(Bk - 0.63)^2 که در آن Bk نسبت سیاهپوستان بر اساس شهر است.

۱۳) LSTAT: وضعیت پایین‌تر از جمعیت به درصد

متغیرهای خروجی:

14) MEDV: ارزش متوسط خانه‌های تحت اشغال در ۱۰۰۰ دلار [k$]

مرحله سوم: بارگیری و تجزیه و تحلیل اکتشافی داده‌ها با pandas

برای وارد کردن جدول اطلاعاتی باید با استفاده از کتابخونه pandas با کد زیر بیاریش تو نوت بوک ArcGIS Pro:

```
data = pd.read_csv("boston.csv")
```

برای دیدن جدول هم میتونی اسم فایل که اینجا data شده رو بنویسی و شیفت اینتر بزنی که اطلاعاتش رو نشون بده.

```
data = pd.read_csv("boston.csv")
```

```
data
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.88	11.9

506 rows × 14 columns

تجزیه و تحلیل اکتشافی داده (EDA)

قبل از ورود به تجزیه و تحلیل داده، مهمه که ساختار و ویژگی‌های مجموعه داده رو درک کنیم. بیا با استفاده از کتابخانه pandas، چندتا تجزیه و تحلیل اکتشافی داده رو با هم انجام بدیم.

نمایش چند سطر اول از مجموعه داده (معمولا ۵ سطر اول رو نشون میده)

```
print(data.head())
```

```
print(data.head())
      CRIM    ZN  INDUS  CHAS    NOX  ...    TAX  PTRATIO       B  LSTAT  MEDV
0  0.00632  18.0   2.31     0  0.538  ...  296.0     15.3  396.90   4.98  24.0
1  0.02731   0.0   7.07     0  0.469  ...  242.0     17.8  396.90   9.14  21.6
2  0.02729   0.0   7.07     0  0.469  ...  242.0     17.8  392.83   4.03  34.7
3  0.03237   0.0   2.18     0  0.458  ...  222.0     18.7  394.63   2.94  33.4
4  0.06905   0.0   2.18     0  0.458  ...  222.0     18.7  396.90   5.33  36.2

[5 rows x 14 columns]
```

دریافت خلاصه آماری از مجموعه داده (مثل مجموع، میانگین، میانه و ...)

```
print(data.describe())
```

```
print(data.describe())
             CRIM          ZN       INDUS  ...           B       LSTAT        MEDV
count  506.000000  506.000000  506.000000  ...  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779  ...  356.674032   12.653063   22.532806
std      8.601545   23.322453    6.860353  ...   91.294864    7.141062    9.197104
min      0.006320    0.000000    0.460000  ...    0.320000    1.730000    5.000000
25%      0.082045    0.000000    5.190000  ...  375.377500    6.950000   17.025000
50%      0.256510    0.000000    9.690000  ...  391.440000   11.360000   21.200000
75%      3.677083   12.500000   18.100000  ...  396.225000   16.955000   25.000000
max     88.976200  100.000000   27.740000  ...  396.900000   37.970000   50.000000

[8 rows x 14 columns]
```

اگه فقط چند مورد مثل میانگین و انحراف معیار مدنظرت هست میتونی جدا بنویسی شون:

محاسبه میانگین و انحراف معیار

```
mean = data.mean()
std = data.std()
```

نمایش میانگین و انحراف معیار

```
print(mean)
print(std)
```

```
print(mean)
CRIM        3.613524
ZN         11.363636
INDUS      11.136779
CHAS        0.069170
NOX         0.554695
RM          6.284634
AGE        68.574901
DIS         3.795043
RAD         9.549407
TAX       408.237154
PTRATIO    18.455534
B         356.674032
LSTAT      12.653063
MEDV       22.532806
dtype: float64
```

```
print(std)
CRIM        8.601545
ZN         23.322453
INDUS       6.860353
CHAS        0.253994
NOX         0.115878
RM          0.702617
AGE        28.148861
DIS         2.105710
RAD         8.707259
TAX       168.537116
PTRATIO     2.164946
B          91.294864
LSTAT       7.141062
MEDV        9.197104
dtype: float64
```

- **بررسی نوع داده هر ستون** (دو نوع داده داریم عدد و متن str. عدد هم دو بخش میشه صحیح int و اعشاری float)

```
print(data.dtypes)
```

```
CRIM        float64
ZN          float64
INDUS       float64
CHAS          int64
NOX         float64
RM          float64
AGE         float64
DIS         float64
RAD           int64
TAX         float64
PTRATIO     float64
B           float64
LSTAT       float64
MEDV        float64
dtype: object
```

- **بررسی تعداد مقادیر ناقص در هر ستون** (اینکه داده null یا خالی تو ستونها هست یا نه)

```
print(data.isnull().sum())
```

```
CRIM        0
ZN          0
INDUS       0
CHAS        0
NOX         0
RM          0
AGE         0
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT       0
MEDV        0
dtype: int64
```

تو این جدول داده null یا خالی نداریم و همه ستونها پر هستن.

مرحله چهارم: تعیین میانگین قیمت مسکن، واریانس مسکن و همبستگی بین تعداد اتاقها و قیمت مسکن با numpy

- **استخراج ستون قیمت مسکن که اطلاعاتش تو ستون MEDV اومده**

```
price = data['MEDV']
```

- **محاسبه میانگین قیمت مسکن با استفاده از NumPy**

```
mean_price = np.mean(price)
```

- **خروجی گرفتن از میانگین قیمت مسکن**

```
print(mean_price)
```

- **محاسبه واریانس مسکن با استفاده از NumPy**

```
variance = np.var(price)
```

- **خروجی گرفتن از واریانس مسکن**

```
print(variance)
```

- **استخراج ستون تعداد اتاق‌ها که اطلاعاتش تو ستون RM اومده**

```
rooms = data['RM']
```

- **محاسبه همبستگی بین تعداد اتاق‌ها و قیمت مسکن با استفاده از NumPy**

```
correlation = np.corrcoef(rooms, price)[0,1]
```

- **خروجی گرفتن از همبستگی بین تعداد اتاق‌ها و قیمت مسکن**

```
print(correlation)
```

```
price = data['MEDV']
mean_price = np.mean(price)
print('mean_price: ', mean_price)
variance = np.var(price)
print('variance: ', variance)
rooms = data['RM']
correlation = np.corrcoef(rooms, price)[0,1]
print('correlation: ', correlation)
```

```
mean_price:  22.532806324110677
variance:  84.4195561561656
correlation:  0.6953599470715395
```

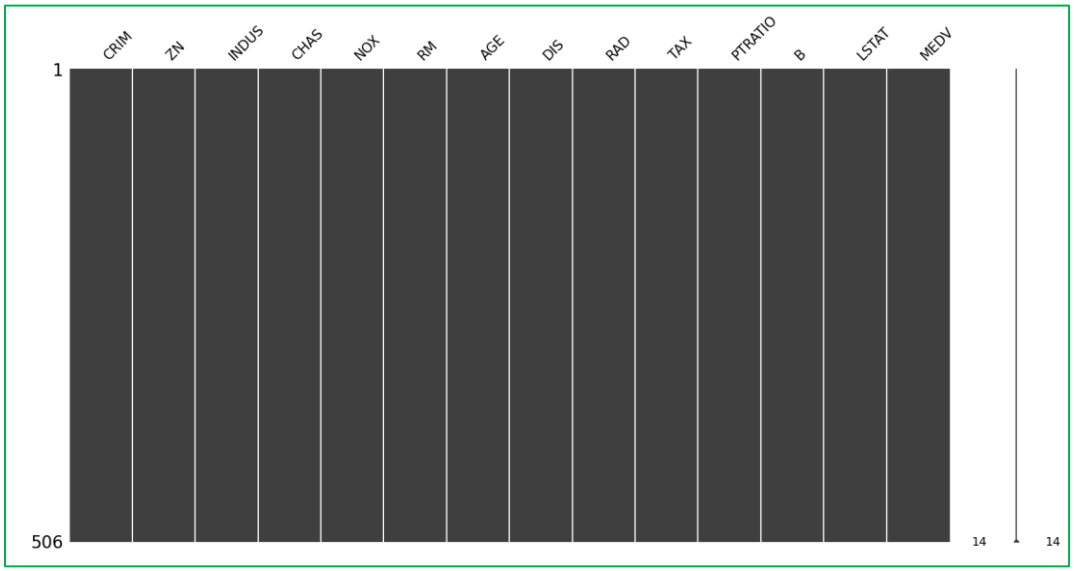
مرحله پنجم: مدیریت داده‌های ناقص با missingno

داده‌های ناقص می‌تونن روی دقت تجزیه و تحلیل داده‌ها تأثیر بذارن. کتابخانه missingno روش مناسبی برای بصری‌سازی الگوهای داده‌های ناقص بهمون میده. بیا ازش واسه شناسایی مقادیر ناقص در مجموعه داده استفاده کنیم.

بصری‌سازی داده‌های ناقص با استفاده از نمودار ماتریس

msno.matrix(data)
plt.show()

```
msno.matrix(data)
plt.show()
```



نمودار ماتریس نشون میده که داده خالی تو جدولمون وجود نداره و همه اطلاعات ستونها کامل هستن. اگه کامل نبودن باید رگه‌های سفید تو نمودار دیده می‌شد.

مرحله ششم: تحلیل داده‌ها با seaborn و matplotlib

می‌تونی از کتابخانه seaborn و matplotlib برای تجسم‌سازی داده‌ها استفاده کنی. میخوایم یه نمودار Scatter با دو تا ستون CRIM برای محور x و MEDV برای محور y بسازیم.

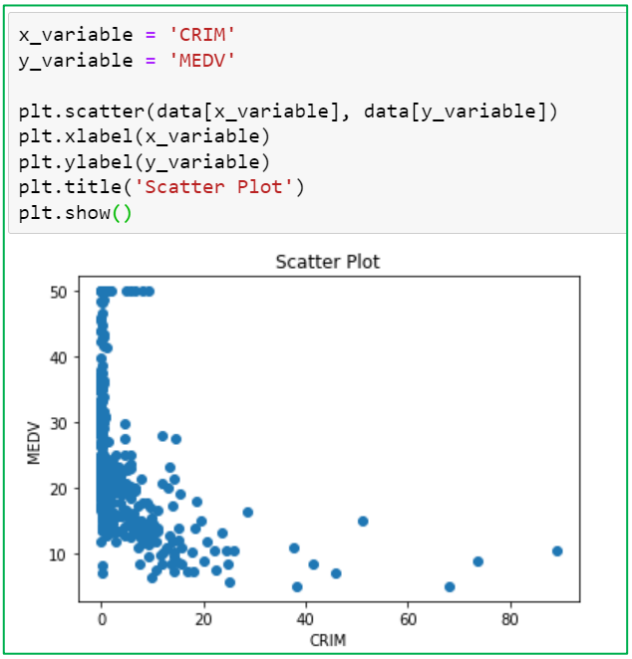
ساخت نمودار scatter با matplotlib:

✓ اول باید دو تا متغیر برای محور x و y رو تعریف کنیم:

```
x_variable = 'CRIM'
y_variable = 'MEDV'
```

✓ بعد باهاشون با plt.scatter() نمودار Scatter بسازیم. میتونیم به محورهاش لیبل بدیم و برای نمودار سرتیتر هم مشخص کنیم.

```
plt.scatter(data[x_variable], data[y_variable])
plt.xlabel(x_variable)
plt.ylabel(y_variable)
plt.title('Scatter Plot')
plt.show()
```



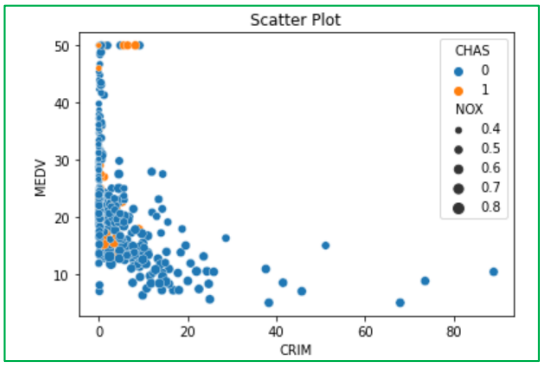
ساخت نمودار scatter با seaborn:

✓ اول باید دو تا متغیر x و y رو تعریف کنیم:

```
x_variable = 'CRIM'
y_variable = 'MEDV'
```

✓ بعد باهاشون با sns.scatterplot() نمودار Scatter بسازیم. میتونیم به محورهاش لیبل بدیم و برای نمودار سرتیتر هم مشخص کنیم.

```
sns.scatterplot(data=data, x=x_variable, y=y_variable, hue='CHAS', size='NOX')
plt.title('Scatter Plot')
plt.show()
```



به طور کلی، seaborn امکانات بیشتری رو برای سفارشی‌سازی نمودارها و اعمال استایل‌های زیبا به اونها فراهم می‌کنه. واسه همین، میتونی از توابع و تنظیمات بیشتری که در seaborn موجود هست برای بهبود ظاهر و قابلیت خوانایی نمودارها استفاده کنی. کتابخونه seaborn امکانات زیادی برای تنظیم رنگ و اندازه نقاط در نمودارها داره. برای تنظیم رنگ و اندازه نقاط در تجسم‌سازی نمودار توزیع مکانی با seaborn، می‌تونی از پارامترهای hue و size استفاده کنی.

پارامتر hue بهت این امکان رو میده تا بر اساس یک متغیر دیگه، نقاط رو با رنگ‌های متفاوت مشخص کنی. مثلاً اگر میخوای نقاط رو بر اساس یک متغیر دسته‌ای مثل "نوع" (type) رنگ‌بندی کنی، کافیه پارامتر hue رو برابر با نام اون متغیر تنظیم کنی.

- **پارامتر size** هم برای تنظیم اندازه نقاط استفاده میشه. برای مثال، اگه می‌خوای اندازه نقاط بر اساس یک متغیر عددی مثل "تعداد" (count) تغییر کنه یا اینجا که بر اساس ستون NOX اندازه‌ها رو مشخص کردیم، کافیه پارامتر size رو برابر با نام اون متغیر تنظیم کنی.

از پارامترهای مهم دیگه‌ای هم میتونی تو Seaborn استفاده کنی مثل:

- **style** این پارامتر بهت این امکان رو میده که نقاط رو بر اساس یک متغیر دیگه به صورت متفاوتی سبک‌بندی کنی، مثل خط یا نقطه.
- **alpha** با استفاده از این پارامتر، می‌تونی شفافیت (opacity) نقاط رو تنظیم کنی. مقادیر بین ۰ و ۱ برای این پارامتر قابل قبول هستن.
- **markers** با استفاده از این پارامتر، می‌توانی نوع نشانگر (marker) نقاط رو تعیین کنی.
- **palette** این پارامتر برای تعیین یک پالت رنگ برای رنگ‌بندی داده‌ها کاربرد داره. می‌تونی یکی از پالت‌های پیش‌فرض seaborn رو انتخاب کنی یا یک پالت رنگ سفارشی تعریف کنی.
- **legend** با استفاده از این پارامتر، می‌تونی نمایش/مخفی کردن نمایه (legend) رو کنترل کنی.

مرحله هفتم: ترسیم نمودار درختی با Squarify

نمودار درختی با استفاده از کتابخونه Squarify ساخته میشه، اندازه و نسبت بین مقادیر مختلف رو با استفاده از مستطیل‌ها نشون میده. هر مستطیل در نمودار نشون دهنده یک دسته‌بندی هست و اندازه اون نشون‌دهنده تعداد رکوردهای متناظر با اون دسته‌بندی هست.

محاسبه تعداد رکوردها بر اساس ناحیه (ستون RAD شاخص دسترسی به بزرگراه‌ها رو بهمون نشون میده)

```
region_counts = data['RAD'].value_counts()
```

تنظیم اندازه نمودار درختی

```
plt.figure(figsize=(۱۰, ۶))
```

ساخت نمودار درختی با استفاده از Squarify

```
squarify.plot(sizes=region_counts.values, label=region_counts.index, alpha=۰٫۷)
```

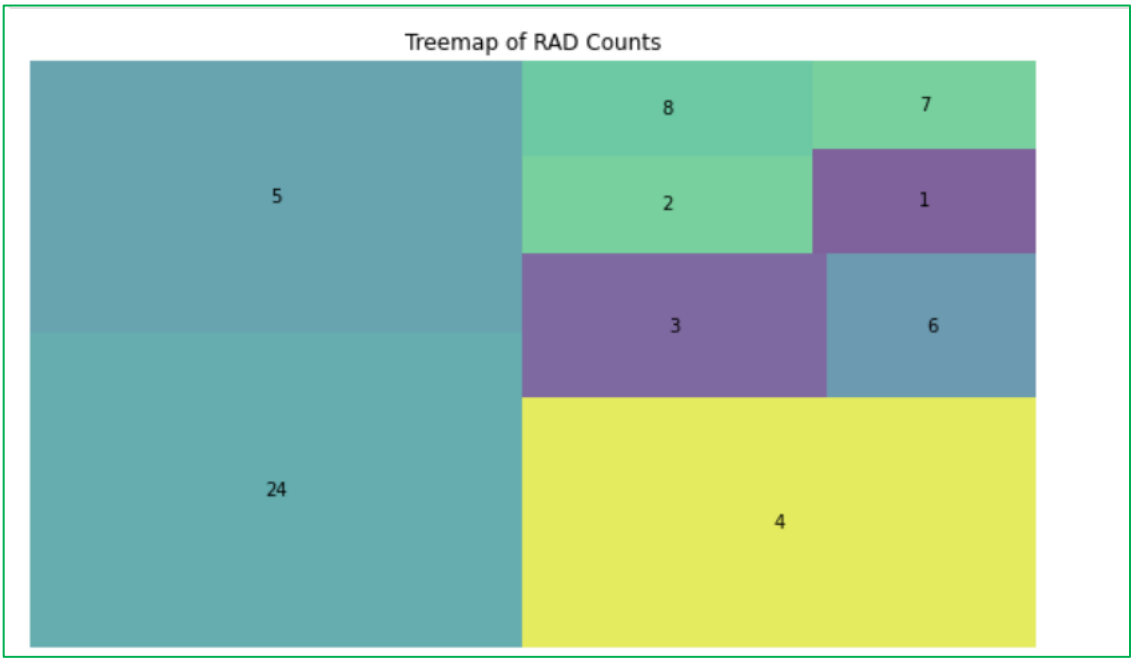
تنظیم عنوان و محورها

```
plt.title('Treemap of RAD Counts')
plt.axis('off')
```

نمایش نمودار درختی

```
plt.show()
```

```
region_counts = data['RAD'].value_counts()
plt.figure(figsize=(10, 6))
squarify.plot(sizes=region_counts.values, label=region_counts.index, alpha=0.7)
plt.title('Treemap of RAD Counts')
plt.axis('off')
plt.show()
```



تو کد بالا، از ستون "RAD" (راه دسترسی به بزرگراه‌ها) جدول "boston.csv" استفاده شده. تعداد رکوردها بر اساس مقادیر مختلف در این ستون بررسی می‌شه و بعد نمودار درختی با استفاده از Squarify ساخته می‌شه. اندازه هر مستطیل در نمودار بر اساس تعداد رکوردها هست و برچسب‌ها هم نمایش داده می‌شن. اندازه و شکل مستطیل‌ها طوری تعیین میشن که نسبت‌های تعداد رکوردها رو به طور دقیق نشون بدن. این نمودار درختی به صورت حجمی و قابل تفسیر هست و بهت این امکان رو میده تا به سرعت و با یک نگاه، اندازه و توزیع مقادیر در دسته‌بندی‌های مختلف را در جدول "boston.csv" ببینی.

خلاصه دستورهای تحلیل مسکن در بوستن در یک نگاه

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
import missingno as msno
%matplotlib inline
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
import missingno as msno
%matplotlib inline
```

کتابخونه Pandas

برای وارد کردن جدول اطلاعاتی به نوت بوک

```
data = pd.read_csv("boston.csv")
```

تجزیه و تحلیل اکتشافی داده:

```
# نمایش چند سطر اول از مجموعه داده (معمولا ۵ سطر اول رو نشون میده)
print(data.head())

#دریافت خلاصه آماری از مجموعه داده (مثل مجموع، میانگین، میانه و ...)
print(data.describe())
#اگه فقط چند مورد مدنظرت هست میتونی جدا بنویسی شون:
محاسبه و مشاهده میانگین و انحراف معیار
print(data.mean())
print(std = data.std())
```

```
      CRIM    ZN  INDUS  CHAS    NOX  ...    TAX  PTRATIO       B  LSTAT  MEDV
0  0.00632  18.0   2.31     0  0.538  ...  296.0     15.3  396.90   4.98  24.0
1  0.02731   0.0   7.07     0  0.469  ...  242.0     17.8  396.90   9.14  21.6
2  0.02729   0.0   7.07     0  0.469  ...  242.0     17.8  392.83   4.03  34.7
3  0.03237   0.0   2.18     0  0.458  ...  222.0     18.7  394.63   2.94  33.4
4  0.06905   0.0   2.18     0  0.458  ...  222.0     18.7  396.90   5.33  36.2

[5 rows x 14 columns]

             CRIM          ZN       INDUS  ...           B       LSTAT        MEDV
count  506.000000  506.000000  506.000000  ...  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779  ...  356.674032   12.653063   22.532806
std      8.601545   23.322453    6.860353  ...   91.294864    7.141062    9.197104
min      0.006320    0.000000    0.460000  ...    0.320000    1.730000    5.000000
25%      0.082045    0.000000    5.190000  ...  375.377500    6.950000   17.025000
50%      0.256510    0.000000    9.690000  ...  391.440000   11.360000   21.200000
75%      3.677083   12.500000   18.100000  ...  396.225000   16.955000   25.000000
max     88.976200  100.000000   27.740000  ...  396.900000   37.970000   50.000000

[8 rows x 14 columns]
```

```
# بررسی نوع داده هر ستون (دو نوع داده داریم عدد و متن str. عدد هم دو بخش میشه صحیح int و اعشاری float)
print(data.dtypes)

#بررسی تعداد مقادیر ناقص در هر ستون (اینکه داده null یا خالی تو ستونها هست یا نه)
print(data.isnull().sum())
```

```
CRIM       float64
ZN         float64
INDUS      float64
CHAS         int64
NOX        float64
RM         float64
AGE        float64
DIS        float64
RAD          int64
TAX        float64
PTRATIO    float64
B          float64
LSTAT      float64
MEDV       float64
dtype: object
```

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

کتابخونه Numpy

تعیین میانگین قیمت مسکن، واریانس مسکن و همبستگی بین تعداد اتاقها و قیمت مسکن با numpy

```
# استخراج ستون قیمت مسکن که اطلاعاتش تو ستون MEDV اومده
price = data['MEDV']
# محاسبه میانگین قیمت خان ها با استفاده از NumPy
mean_price = np.mean(price)
#خروجی گرفتن از میانگین قیمت مسکن
print(mean_price)
#محاسبه واریانس مسکن با استفاده از NumPy
variance = np.var(price)
#خروجی گرفتن از واریانس مسکن
print(variance)
#استخراج ستون تعداد اتاق‌ها که اطلاعاتش تو ستون RM اومده
rooms = data['RM']
#محاسبه همبستگی بین تعداد اتاق‌ها و قیمت مسکن با استفاده از NumPy
correlation = np.corrcoef(rooms, price)[۰,۱]
#خروجی گرفتن از همبستگی بین تعداد اتاق‌ها و قیمت مسکن
print(correlation)
```

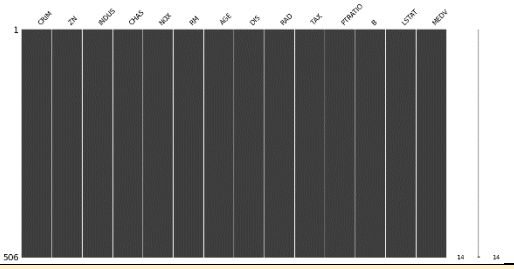
```
price = data['MEDV']
mean_price = np.mean(price)
print('mean_price: ', mean_price)
variance = np.var(price)
print('variance: ', variance)
rooms = data['RM']
correlation = np.corrcoef(rooms, price)[0,1]
print('correlation: ', correlation)
```

```
mean_price:  22.532806324110677
variance:  84.4195561561656
correlation:  0.6953599470715395
```

کتابخونه missingno

بصری‌سازی داده‌های ناقص

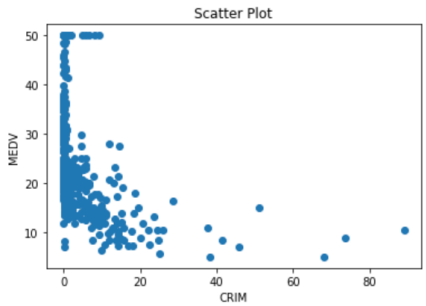
```
#بصری‌سازی داده‌های ناقص با استفاده از نمودار ماتریس
msno.matrix(data)
plt.show()
```



کتابخونه matplotlib

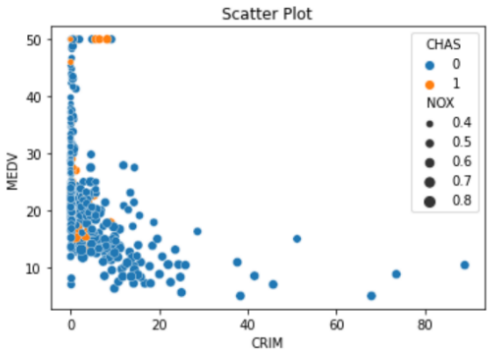
ترسیم نمودار Scatter با دو تا ستون CRIM برای محور x و MEDV برای محور y

```
#اول باید دو تا متغیر رو تعریف کنیم:
x_variable = 'CRIM'
y_variable = 'MEDV'
#بعد باهاشون نمودار Scatter بسازیم.
plt.scatter(data[x_variable], data[y_variable])
#میتونیم به محورهاش لیبل بدیم
plt.xlabel(x_variable)
plt.ylabel(y_variable)
#برای نمودار سرتیتر مشخص می‌کنیم
plt.title('Scatter Plot')
plt.show()
```



کتابخونه seaborn

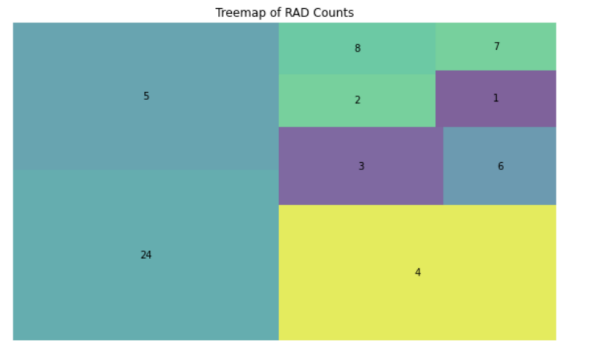
ترسیم نمودار Scatter با دو تا ستون CRIM برای محور x و MEDV برای محور y




```
#اول باید دو تا متغیر رو تعریف کنیم:
x_variable = 'CRIM'
y_variable = 'MEDV'
#بعد باهاشون نمودار Scatter بسازیم. بهش hue برای دادن به متغیر دیگه برای رنگ دادن به نمودار که اینجا از ستون CHAS استفاده کرده و size واسه دادن یه متغیر دیگه برای اندازه نقاط به نمودار که اینجا از ستون NOX استفاده کرده.
sns.scatterplot(data=data, x=x_variable, y=y_variable, hue='CHAS', size='NOX')
#میتونیم به محورهاش لیبل بدیم
plt.xlabel(x_variable)
plt.ylabel(y_variable)
#برای نمودار سرتیتر مشخص می‌کنیم
plt.title('Scatter Plot')
plt.show()
```

کتابخونه Squarify

برای ترسیم Treemap (نمودار مستطیلی) از اطلاعات رکوردهای یه ستون که اینجا RAD هست و دسترسی به بزرگراه‌ها رو نشون میده.




```
#محاسبه تعداد رکوردها بر اساس ناحیه (ستون RAD شاخص دسترسی به بزرگراه‌ها رو بهمون نشون میده)
region_counts = data['RAD'].value_counts()
#تنظیم اندازه نمودار درختی
plt.figure(figsize=(۱۰, ۶))
#ساخت نمودار درختی با استفاده از Squarify
squarify.plot(sizes=region_counts.values, label=region_counts.index, alpha=۰٫۷)
#تنظیم عنوان و محورها
plt.title('Treemap of RAD Counts')
plt.axis('off')
#نمایش نمودار درختی
plt.show()
```

اگه بخوایم یه تحلیل کلی روی کارهایی که کردیم داشته باشیم میتونیم بگیم که در نمودار Scatter دو تا ستون CRIM (نرخ سرانه جرم بر اساس شهر) و MEDV (ارزش متوسط خانه‌های تحت اشغال در ۱۰۰۰ دلار [k$]) در تقابل با هم هستن و با بالا رفتن یکی، دومی می‌آد پایین.

در مورد نمودار درختی که بر اساس ستون RAD (شاخص دسترسی به بزرگراه‌های شعاعی) ترسیم شده میشه به این شکل تحلیل کرد بیشتر مردم دسترسی کمی به بزرگراه‌ها دارن یا ازش کمتر برخوردار هستن.

پروژه ۲: تحلیل حمل و نقل و ترافیک

مشابه روش بالا رو میتونی برای تحلیلهای دیگه شهری هم انجام بدی مثلا در مورد تحلیل حمل و نقل و ترافیک میتونی میانگین زمان سفر رو محاسبه کنی و براش نمودار ترسیم کنی. اینجا فقط مراحل رو توضیح میدم به جای ستونهای موجود تو دستورها از ستونهای فایل شخصی که داری استفاده کن. کدها رو ببر تو نوت بوک یا هر جایی که بشه باهاش کد پایتون نوشت و نتایجش رو ببین.

❖ مرحله اول: وارد کردن کتابخونه‌ها و عبارت جادویی

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import squarify
import missingno as msno
%matplotlib inline
```

❖ مرحله دوم: آوردن فایل مورد نظر به نوت بوک که فرض میکنیم اســمش 'transportation.csv' هســت، با کمک pandas

```
df = pd.read_csv(' transportation.csv')
```

❖ مرحله سوم: نمایش ابتدایی جدول با پرینت گرفتن از ۵ سطر اولش

```
print(df.head())
```

❖ مرحله چهارم: بررسی داده‌های گمشده با missingno

```
msno.matrix(df)
plt.title('Missing Values')
plt.show()
```

❖ مرحله پنجم: محاسبه آمار توصیفی با استفاده از numpy و pandas

```
statistics = df.describe()
print(statistics)
```

❖ مرحله ششم: ساخت نمودار درختی یا treemap با استفاده از squarify

اینجا Category اسم فرضی یکی از ستونها هست.

```
category_counts = df['Category'].value_counts()

plt.figure(figsize=(10, 6))

squarify.plot(sizes=category_counts.values, label=category_counts.index, alpha=0.7)

plt.title('Transportation Categories')

plt.axis('off')

plt.show()
```

❖ مرحله هفتم: محاسبه میانگین زمان سفر ترسیم نمودار میانگین زمان سفر بر اساس حمل و نقل با استفاده از pandas و seaborn

اینجا transportation_mode و commute_Time ستونهای فرضی جدول هستن.

```
avg_commute_time = df.groupby('transportation_mode')['commute_time'].mean()
print(avg_commute_time)

sns.barplot(x=avg_commute_time.index, y=avg_commute_time.values)
plt.xlabel('Transportation Mode')
plt.ylabel('Average Commute Time')
```

```
plt.title('Average Commute Time by Transportation Mode')
plt.xticks(rotation=45)
plt.show()
```

پروژه ۳: تحلیل فضای سبز

مشابه مراحل بالا، میتونی درصد فضای سبز هر محله رو مشخص کنی و براش میزان فضای سبز در هر محله یه نمودار دلخواه مثلا نمودار جعبه‌ای ترسیم کنی.

اینجا 'neighborhood'، 'area' و 'land_use' ستونهای فرضی جدول اطلاعاتیت هستن. به جاش میتونی سرستونهای خودت رو بذاری

```
#محاسبه درصد فضای سبز در هر محله
total_area = data.groupby('neighborhood')['area'].sum()
green_space_area = data[data['land_use'] == 'Green Space'].groupby('neighborhood')['area'].sum()
green_space_percentage = (green_space_area / total_area) * 100
print(green_space_percentage)
#نمودار میزان فضای سبز در هر محله
plt.figure(figsize=(8, 6))
sns.barplot(x=green_space_percentage.index, y=green_space_percentage.values)
plt.xlabel('Neighborhood')plt.ylabel('Percentage of Green Spaces')
plt.title('Percentage of Green Spaces by Neighborhood')
plt.xticks(rotation=45)
plt.show()
```

با کمک هوش مصنوعی و درک کتابخونه‌های علم داده میشه روی داده‌های پروژه‌های شهری تحلیلهای مختلف انجام داد. میتونید خودتون هم چند تا پروژه جدید به مطالب این جزوه اضافه کنید. ولی پروژه‌های شهری نیاز به تحلیل **داده‌های مکانی** دارن یعنی باید داده‌هامون مکان محور بشن. برای انجام اینکار باید کتابخونه‌های مربوط به داده‌های جغرافیایی که شامل geoplotlib، geopandas و rasterio میشن رو هم یاد بگیریم.

تو جزوه‌های بعدی این ۳ تا کتابخونه مهم رو با هم کار میکنیم و بعدش یه پروژه شهری انجام میدیم که داده‌های مکانی رو بشه باهاشون تحلیل کرد.