پروژه امتحان معرفی به استاد درس محاسبات آماری

بهمن ۱۴۰۴

نام درس: محاسبات آماری
نام مدرس: دکتر کورش دادخواه
موضوع: پروژه پایان‌ترم: تحلیل داده‌های iris با R پایه و اسکریپت‌نویسی
مدت زمان پیشنهادی: ۳ روز کاری (جهت تکمیل و ارائه)

هدف پروژه:
هدف این پروژه، ارزیابی توانایی دانشجو در استفاده از مفاهیم و ابزارهای **پایه زبان برنامه‌نویسی R** برای انجام یک تحلیل داده ساده، مدیریت فضای کاری و پکیج‌ها، اعمال ساختارهای کنترل جریان، ایجاد توابع، کار با توزیع‌های احتمال، انجام آمار توصیفی، مصورسازی داده‌ها، و سازماندهی کد در قالب اسکریپت‌های R است.

داده‌کاوی:
در این پروژه، شما عمدتاً با مجموعه داده داخلی iris کار خواهید کرد. این دیتاست شامل اندازه‌گیری‌های طول و عرض کاسبرگ (sepal) و گلبرگ (petal) برای سه گونه مختلف گل زنبق است.

خروجی مورد انتظار:

۱. **یک فایل اسکریپت R** (.R) که شامل تمامی کدهای شما برای انجام وظایف پروژه است.
۲. **یک فایل متنی** (.txt) که شامل تمامی خروجی‌های متنی کدها (مانند print() یا cat()) و توضیحات و تفاسیر شما برای هر بخش باشد.

دستورالعمل‌های عمومی:

- کدها باید کاملاً مستندسازی شده (با استفاده از کامنت #) و خوانا باشند.
- تمامی خروجی‌های کدها را (چه متنی، چه نمودارها) در فایل .txt خود قرار دهید و هر نمودار را ذخیره کنید تا قابل ارائه باشد.
- توضیحات و تفاسیر هر بخش را به وضوح در فایل .txt خود ارائه دهید.

بخش‌ها و وظایف پروژه:

بخش ۰ : تنظیمات اولیه و آماده‌سازی محیط [فصول ۱، ۶]

۱. **نصب و بارگذاری پکیج‌ها:**
   - پکیج readxl را نصب و بارگذاری کنید (زیرا در ادامه به آن نیاز خواهید داشت، حتی اگر در این پروژه iris داخلی است). [۶,۲]

- هر پکیج دیگری که برای انجام وظایف نیاز دارید و در فصول ۱ تا ۱۰ معرفی شده است (به جز dplyr و ggplot۲) را نصب و بارگذاری کنید.

۲. **تنظیم تکرارپذیری**: از آنجایی که در بخش‌های بعدی نیاز به تولید اعداد تصادفی دارید، set.seed() را با یک مقدار ثابت (مثلاً ۱۲۳) تنظیم کنید. [۸,۲]

۳. **مدیریت فضای کاری**:

- مسیر دایرکتوری کاری فعلی خود را با getwd() نمایش دهید. [۴,۳]
- تمامی اشیاء موجود در فضای کاری خود را با ls() لیست کنید. [۶,۱]

بخش ۱: مدیریت فایل و ورود/خروج داده‌ها [فصل ۴]

۱. **ایجاد و ذخیره یک فایل CSV**:

- یک دیتافریم کوچک (مثلاً my_data) شامل حداقل ۳ ستون (یک عددی، یک کاراکتری، یک منطقی) و ۴ سطر ایجاد کنید.
- این دیتافریم را در یک فایل CSV با نام sample_data.csv در دایرکتوری کاری خود ذخیره کنید. اطمینان حاصل کنید که نام سطرها (row names) ذخیره نشود. [۴,۲]

۲. **خواندن یک فایل CSV**:

- فایل sample_data.csv را که ایجاد کرده‌اید، به یک دیتافریم جدید با نام loaded_data بخوانید. [۴,۱]
- با استفاده از head() و str()، ساختار و چند سطر اول loaded_data را بررسی کنید.

۳. **ذخیره و بارگذاری شیء R**:

- دیتافریم loaded_data را به عنوان یک شیء R جداگانه در یک فایل با نام loaded_data.rds ذخیره کنید (با saveRDS()). [۴,۲]
- همین شیء را دوباره از فایل loaded_data.rds بخوانید و در متغیر reloaded_data ذخیره کنید (با readRDS()).

بخش ۲: بارگذاری، ساختار و دستکاری داده‌های iris [فصل ۳، ۱۰]

۱. **بارگذاری داده‌ها**: دیتاست داخلی iris را بارگذاری کنید ((data(iris)).

۲. **بررسی ساختار**:

- از str()، head() و summary() برای بررسی iris استفاده کنید و نتایج را در فایل txt. خود گزارش دهید. [۳,۱, ۷,۱]

۳. **ایجاد ستون‌های جدید**:

- یک ستون جدید به نام Sepal.Area به دیتافریم iris اضافه کنید که حاصل ضرب Sepal.Length و Sepal.Width باشد.

- یک ستون جدید به نام Petal.Area به دیتافریم iris اضافه کنید که حاصل ضرب Petal.Length و Petal.Width باشد. [۳,۱]

۴. **زیرمجموعه‌سازی (Subsetting):**

- یک دیتافریم جدید به نام setosa_long_sepals ایجاد کنید که فقط شامل مشاهدات گونه "setosa" باشد که Sepal.Length آن‌ها بزرگتر از ۵/۰ است. این کار را با استفاده از تابع subset() انجام دهید. [۱۰,۱]
- یک دیتافریم جدید به نام virginica_wide_petals ایجاد کنید که فقط شامل مشاهدات گونه "virginica" باشد که Petal.Width آن‌ها بزرگتر از ۲/۰ است. این کار را با استفاده از **اندیس‌گذاری منطقی** (logical indexing) و براکت [] انجام دهید. [۱۰,۱ ,۳,۲]
- یک دیتافریم جدید به نام sepal_measurements ایجاد کنید که فقط شامل ستون‌های Sepal.Length, Sepal.Width از دیتافریم اصلی iris باشد. این کار را با اندیس‌گذاری با نام ستون‌ها انجام دهید. [۳,۲]

بخش ۳: برنامه‌نویسی و کنترل جریان [فصل ۵]

۱. **تعریف و استفاده از تابع:**

- یک تابع R به نام classify_petal() بنویسید که دو آرگومان length و width را می‌گیرد.
- اگر length > ٤ و width > ۱/۵ باشد، "Large Petal" را برگرداند.
- اگر length <= ٤ و width <= ۱/۵ باشد، "Small Petal" را برگرداند.
- در غیر این صورت، "Medium Petal" را برگرداند.
- از ساختار if...else if...else استفاده کنید. [۵,۲]
- تابع را با چند مقدار نمونه (مثلاً (۵, ۱/٦), (۳, ۱), (٤/۵, ۱/۲)) تست کنید و خروجی را در فایل txt. قرار دهید.

۲. **استفاده برداری از تابع با ifelse():**

- یک ستون جدید به نام Petal_Category به دیتافریم iris اضافه کنید.
- با استفاده از تابع ifelse() (به صورت تو در تو)، منطق تابع classify_petal() را به طور برداری روی ستون‌های Petal.Length و Petal.Width از دیتافریم iris اعمال کنید. [۵,۳]
- توضیح دهید که چرا ifelse() در اینجا نسبت به حلقه for با تابع شرطی، کارآمدتر است.

۳. **حلقه‌های for و while:**

- یک حلقه for بنویسید که برای هر گونه Species در دیتافریم iris، میانگین Sepal.Length را چاپ کند. (می‌توانید از unique() برای یافتن گونه‌ها و subset() برای فیلتر کردن استفاده کنید). [۵,۲]
- یک حلقه while بنویسید که اعداد زوج از ۱ تا ۱۰ را چاپ کند. (از next برای رد کردن اعداد فرد و break برای توقف در عدد ۸ استفاده کنید). [۵,۲]

بخش ۴: آمار توصیفی و مصورسازی [فصل ۷]

۱. **آماره‌های توصیفی:**

- میانگین، میانه، انحراف معیار، کمینه، بیشینه و تعداد مقادیر غیر NA را برای ستون Petal.Area در دیتافریم iris محاسبه کنید. [۷,۱]
- همین آماره‌ها را برای Petal.Area برای هر Species به صورت جداگانه (با استفاده از tapply() یا aggregate()) محاسبه کنید. [۷,۱]

۲. **مصورسازی داده‌ها (Base R Graphics):**

- **هیستوگرام**: یک هیستوگرام از Petal.Area رسم کنید. نمودار باید دارای عنوان، برچسب محور x و رنگ مناسب باشد. [۷,۲]
- **نمودار جعبه‌ای**: یک نمودار جعبه‌ای از Sepal.Length بر اساس Species رسم کنید (boxplot(Sepal.Length ~ Species, data = iris)). نمودار باید دارای عنوان و برچسب‌های مناسب باشد. [۷,۲]
- **نمودار پراکندگی**: یک نمودار پراکندگی از Sepal.Length (محور x) در برابر Petal.Length (محور y) رسم کنید. [۷,۲]
  - نقاط را بر اساس Species رنگ‌آمیزی کنید (col = iris$Species).
  - یک عنوان و برچسب‌های مناسب برای محورها اضافه کنید.
  - یک خط افقی برای میانگین کلی Petal.Length (با abline()) به نمودار اضافه کنید. [۷,۲]
- **ذخیره نمودارها**: تمامی نمودارهای تولید شده را به صورت فایل‌های تصویری (مثلاً .png) در دایرکتوری کاری خود ذخیره کنید.

بخش ۵: توزیع‌های احتمال و شبیه‌سازی [فصل ۸]

۱. **توزیع نرمال:**

- با استفاده از set.seed(۱۲۳)، ۲۰۰ عدد تصادفی از توزیع نرمال با میانگین ۱۰ و انحراف معیار ۲ تولید و در متغیری به نام sim_normal_data ذخیره کنید. [۸,۲]
- احتمال اینکه یک مشاهده تصادفی از این توزیع (mean=۱۰, sd=۲) کمتر از ۸ باشد را محاسبه کنید (pnorm()). [۸,۱]
- مقدار x را پیدا کنید که برای این توزیع، ۹۰٪ از مشاهدات کمتر از آن باشند (qnorm()). [۸,۱]

۲. **توزیع دوجمله‌ای:**

- ۱۰ بار پرتاب یک سکه را شبیه‌سازی کنید (تعداد آزمایش ۱, احتمال موفقیت ۰/۵) و این کار را ۵۰ بار تکرار کنید (با rbinom()). نتایج را در یک بردار ذخیره کرده و میانگین تعداد شیرها را محاسبه کنید. [۸,۱]

۳. **توزیع پواسون:**

- فرض کنید تعداد تماس‌های دریافتی در یک ساعت از توزیع پواسون با نرخ متوسط lambda = ۵ پیروی می‌کند. احتمال دریافت دقیقاً ۳ تماس در یک ساعت را محاسبه کنید [۸,۱] .((dpois()
- ۱۰ مقدار تصادفی از این توزیع (lambda=۵) را تولید کنید [۸,۱] .((rpois()

بخش ۶: اندازه‌گیری عملکرد [فصل ۹]

۱. **مقایسه عملکرد حلقه و برداری‌سازی:**
   - یک بردار عددی بسیار بزرگ (مثلاً شامل ۱,۰۰۰,۰۰۰ عنصر) ایجاد کنید.
   - یک تابع بنویسید که یک حلقه for را برای جمع مربعات تمامی عناصر بردار اجرا می‌کند.
   - یک عملیات برداری برای جمع مربعات تمامی عناصر بردار اجرا کنید ((sum(vector^۲).
   - زمان اجرای هر دو روش را با system.time() اندازه‌گیری کنید. [۹,۳]
   - در فایل .txt، نتایج زمان‌بندی را مقایسه و توضیح دهید که چرا رویکرد برداری شده بسیار سریع‌تر است. [۹,۲]

بخش ۷: ادغام داده‌ها (Merge) [فصل ۱۰]

۱. **ایجاد دیتافریم‌های ساختگی:**
   - دو دیتافریم کوچک ایجاد کنید:
     - df_students: شامل StudentID (مثلاً ۱۰۱, ۱۰۲, ۱۰۳) و Name (مثلاً "Ali", "Sara", "Reza").
     - df_grades: شامل StudentID (مثلاً ۱۰۱, ۱۰۲, ۱۰٤ ـ برای ایجاد یک ناهماهنگی)، Course (مثلاً "Math", "Physics", "Chemistry") و Score (مثلاً ۸۵, ۹۰, ۷۵).
۲. **ادغام داخلی (Inner Join)**: با استفاده از تابع merge(), df_students و df_grades را بر اساس StudentID ادغام کنید (Inner Join). نتایج را نمایش دهید. [۱۰,۲]
۳. **ادغام چپ (Left Join)**: با استفاده از تابع merge(), df_students و df_grades را ادغام کنید به طوری که تمامی دانشجویان از df_students حفظ شوند (Left Join). نتایج را نمایش دهید. [۱۰,۲]
٤. **ادغام کامل (Full Join)**: با استفاده از تابع merge(), df_students و df_grades را ادغام کنید به طوری که تمامی سطرها از هر دو دیتافریم حفظ شوند (Full Join). نتایج را نمایش دهید. [۱۰,۲]

بخش ۸: نتیجه‌گیری و تفسیر [فصول ۱، ۷]

۱. **خلاصه یافته‌ها**: در فایل .txt، مهمترین یافته‌های خود را درباره دیتاست iris، تفاوت گونه‌ها در ابعاد گلبرگ/کاسبرگ، و نیز نکات کلیدی که از انجام این پروژه آموخته‌اید (مانند اهمیت برداری‌سازی و مدیریت فایل‌ها) در یک پاراگراف توضیح دهید.
۲. **خود ارزیابی**: در فایل .txt، تجربه‌ی خود را از کار با R پایه و چالش‌های این پروژه (با توجه به مباحثی مانند اسکوپ، برداری‌سازی یا مدیریت پکیج‌ها) در یک پاراگراف مختصر شرح دهید.

راهنمای ارائه:

- **فایل اسکریپت** (Project_Name.R) **R** و **فایل متنی توضیحات** (Project_Name_Report.txt) را ارسال کنید.
- تمامی نمودارهای تولید شده (تصاویر .png) را نیز به همراه فایل‌ها ارسال کنید.
- اطمینان حاصل کنید که تمامی خروجی‌های کدها (متنی) در فایل .txt شما موجود باشد.

موفق باشید!