دانشکده مهندسی برق

آمار و احتمال مهندسی

فاز دوم پروژه

محمد مهدی رزمجو – ۴۰۰۱۰۱۲۷۲

بهار ۱۴۰۲

۱ ضریب β

یکی از مهم ترین مفاهیم در تصمیم گیری سرمایه گذاری ریسک و بازده ، ضریب بتا می باشد. هر سهم یا پرتفوی از سهام اگر در فاصله خاص از زمان خریداری، نگهداری و فروخته شود بازده خاصی نیز نصیب مالک خود می نماید. این بازدهی شامل تغییر قیمت زمان خرید و فروش است. همیشه در تصمیم گیری های مالی وجود معیار اندازه گیری ریسک مفید است. امروزه اکثر پژوهشگران ریسک سرمایه گذاری را با انحراف معیار نرخ بازده مرتبط می دانند، یعنی هر چه قدر بازده سرمایه گذاری بیشتر تغییر کند سرمایه گذاری مزبور ریسک بیشتری خواهد داشت. پیش از توضیح ضریب بتا ، ابتدا انواع ریسک ها را بررسی میکنیم.
منابع ریسک که باعث تغییر و پراکندگی در بازده می شود به دودسته کلی تقسیم می کنند:
ریسک غیرسیستماتیک (قابل اجتناب):
این ریسک آن بخش از کل ریسک مجموعه سهام را که مختص به یک شرکت یا صنعت خاص است، نشان می دهد. برخی از عوامل ایجادکننده این ریسک شامل کالاها و خدمات تولیدی شرکت یا صنعت، نوع مدیریت، اقدامات رقبا و ساختار هزینه های شرکت است. بنابراین این نوع ریسک قابل کنترل است.
ریسک سیستماتیک (غیرقابل اجتناب):
بخشی از ریسک است که نمی توان آن را به راحتی کاهش داد. ازجمله این عوامل ریسک زا می توان تحولات سیاسی و اقتصادی، چرخه های تجاری، تورم و بیکاری را نام برد. بنابراین، این نوع ریسک غیر قابل کنترل است.

پرسش تئوری ۱

به نظر شما چرا ریسک غیرسیستماتیک قابل کنترل و ریسک سیستماتیک غیر قابل کنترل است؟

پاسخ ۱

به طور کلی، ریسک غیرسیستماتیک ناشی از عوامل داخلی بوده و در نتیجه، مدیریت آن برای شرکت قابل انجام می باشد. اما ریسک سیستماتیک به عوامل خارجی وابسته بوده که کنترل آن خارج از اراده شرکت می باشد.

پرسش تئوری ۲

نظر شما چگونه ریسک غیرسیستماتیک را میتوان کنترل کرد؟

پاسخ ۲

به عنوان یک شخص سرمایه گذار، اگر سرمایه خود را در مجموعه سهام های مختلف سرمایه گذاری کنیم، به نوعی ریسک غیرسیستماتیک را کاهش داده ایم زیرا، در صورت مواجه یک سهام با افت شدید، سرمایه خود را بطور کلی از دست نمی دهیم.

درواقع بتا معیار اندازه گیری ریسک سیستماتیک یک اوراق بهادار است که به عنوان قسمتی از ریسک کلی نمی توان آن را کاهش داد یا از بین برد. بتا معیار نسبی ریسک یک سهم با توجه به پرتفوی بازار تمامی سهام ها است. برای محاسبه β از شاخص قیمت بورس سهام استفاده می کنند. ضریب بتا برای یک سهم بخصوص، به این صورت تعیین خواهد شد که ریسک سیستماتیک آن سهم را با ریسک سیستماتیک متعلق به شاخص بورس سهام مقایسه می کنند. به طور کلی میزان همسویی بازده یک سهم نسبت به بازار کل (شاخص کل) را با ضریب بتا می سنجند. بتای مثبت نشانه همسویی با بازار، بتای منفی نشانه واگرایی با بازار و بتای صفر نشانه عملکرد خنثی (عدم تاثیرپذیری) سهم نسبت به بازار (شاخص) است. در ضریب بتا، عدد یک را به عنوان مبنا قرار می دهیم. به عنوان مثال اگر بگوییم ضریب بتای سهمی نسبت به شاخص عدد ۲ می باشد، یعنی به میزان دو برابری شاخص بازدهی دارد یا به عبارتی اگر شاخص ۲۰ درصد رشد نماید آن سهم ۴۰ درصد رشد می کند و برعکس.

پرسش تئوری ۳

به نظر شما زمانی که انتظار رشد شاخص سهام بازار را داریم، بین دو سهم با بتا های متفاوت، کدام را خریداری کنیم؟ اگر انتظار کاهش رشد شاخص داشته باشیم چطور؟

پاسخ ۳

طبیعتا زمانی که انتظار رشد شاخص سهام بازار را داریم، سهمی را بر می گزینیم که β بیشتری داشته باشد که سود بیشتری ببریم. با همین استدلال، در زمانی که انتظار کاهش شاخص سهام بازار را داریم، بایستی سهم با β کمتر را انتخاب کنیم تا ضرر کمتری را متحمل شویم/

پرسش تئوری ٤

با توجه به اینکه دامنه تغییر بتا $-\infty$ تا $+\infty$ می باشد، هرکدام از شرایط زیر را مانند بالا تحلیل کنید:
$\beta = 0$ ، $\beta = -2$ ، $\beta = 0.5$

پاسخ ۴

به ازای $\beta = 0$ ، در صورتی بازار شاخص بالا برود یا پایین برود، هیچ تغییری در سهامی مورد نظر از لحاظ سود یا زیان حاصل نمی شود.
به ازای $\beta = -2$ اگر شاخص کل به میزان x درصد رشد کند، سهام مورد نظر به اندازه $2x$ سقوط خواهد کرد و باالعکس.
به ازای $\beta = 0.5$ ، اگر سهامی به اندازه x درصد رشد کند، سهام مورد نظر به اندازه $0.5x$ درصد رشد خواهد داشت و باالعکس.

پرسش تئوری ۵

اگر فرمول بتا برای سهم در بازار به صورت زیر رو باشد:

$$\beta = \frac{Cov(r_1, r_2)}{Var(r_1)^m}$$

که r_1 بازدهی بازار و r_2 بازدهی سهم است. با توجه به مبنا در بتا و بتای کل بازار که برابر با یک است، m را محاسبه کنید.

پاسخ ۵

از آنجایی که ضریب β برای کل بازای برابر یک است و در این صورت r_1 و r_2 با یکدیگر برابر خواهند بود خواهیم داشت:

$$1 = \frac{Var(r_1)}{Var(r_1)^m}$$

در نتیجه میتوان گفت که $m = 1$ می باشد.

پرسش تئوری ٦

با توجه به اطلاعات خودتان از واریانس و کواریانس و توضیحاتی که درمورد ضریب بتا دادیم، به نظرتون چرا فرمول پیشنهادی برای بتا به صورتی است که در پرسش تئوری پنج گفته شده است؟

پاسخ ۶

با توجه به اینکه $covariance$ میزان تغییرات دو متغیر تصادفی نسبت به یکدیگر را نشان می دهد و از طرفی ، $variance$ بیان کننده پخش بودن داده های یک متغیر تصادفی نسبت به یکدیگر می باشد، میتوان گفت که ضریب بتا از فرمول داده شده، محاسبه می شود. در این فرمول،تغییرات بازدهی سهم مورد نظر و بازهی بازار نسبت به یکدیگر (کواریانس این دو متغیر) نسبت به تغییرات بازدهی بازار (واریانس) ، نشان دهنده ریسک سهام مورد نظر می باشد.

پرسش شبیه سازی ۱

دو دنباله تصادفی ۱۰عضوی ایجاد کنید که هرکدام از دنباله ها قیمت پایانی یک سهم در پایان روز i ام هستند. (قیمت هارا به صورت تصادفی بین ۱تا ۱۰۰۰ تعریف کنید.) برای هر سهم R_i را به صورت بازده خرید سهم در روز i ام و فروش آن در روز $i + 1$ ام تعریف میکنیم. وبازده کل سهم را نیز میانگین R_i ها در نظر میگیریم که i از ۱ تا ۹ است. بازده کل دو دنباله ایجادی را محاسبه کنید.

قطعه کد این بخش به صورت زیر می باشد:

```
first_list = []
second_list = []

for i in range(10):
    first_list.append(random.randint(1, 1000))
    second_list.append(random.randint(1, 1000))

first_Ris = []
second_Ris = []

for i in range(9):
    first_Ris.append((first_list[i+1]-first_list[i])/first_list[i]*100)
    second_Ris.append((second_list[i+1]-second_list[i])/second_list[i]*100)

total_return1 = sum(first_Ris)/9
total_return2 = sum(second_Ris)/9

print("Total return of the first sequence : "+str(round(total_return1, 2)))
print("Total return of the second sequence : "+str(round(total_return2, 2)))
```

خروجی نمونه بالا به صورت زیر می باشد:

$$Total\ return\ of\ the\ first\ sequence\ :\ 184.83$$
$$Total\ return\ of\ the\ second\ sequence\ :\ 11.03$$

۲ سرمایه گذار کاردرست

یک سرمایه گذار خوب برای اینکه اسیر احساسات خود نشود، باید قبل از اقدام به خرید سهام حد سود و حد ضرر برای خود تعیین کند. برای این منظور، یک تابع «خروج از معامله» به صورت زیر تعریف می کنیم:

$$f(x) = \begin{cases} f(t-1)+1 & If\ the\ price\ change\ on\ day\ t\ is\ positive \\ f(t-1)-1 & If\ the\ price\ change\ on\ day\ t\ is\ negative \end{cases}$$

منظور از تغییرات قیمت، اختلاف قیمت پایانی است و همچنین $f(0)=0$. اگر هر یک از دو شرط زیر برقرار شد، سرمایه گذار تصمیم به خروج از معامله می گیرد:

- $f(t) \geq H$: یعنی از حد سود عبور کرده باشیم.
- $f(t) \leq L$: یعنی از حد ضرر عبور کرده باشیم.

فرض کنید احتمال اینکه تغییرات قیمت در یک روز مثبت باشد، مستقل از روزهای دیگر برابر p است.

پرسش تئوری ۷

احتمال اینکه سرمایه گذار با موفقیت از معامله خارج شود را محاسبه کنید.

پاسخ ۷

$$P(Successfully\ Exit) = p^H + p^H(p(1-p)) + p^H(p^2(1-p)^2) + \dots$$
$$\Rightarrow P(Successfully\ Exit) = \sum_{n=0}^{\infty} p^H(p^n(1-p)^n)$$

پرسش تئوری ۸

احتمال اینکه سرمایه گذار با شکست از معامله خارج شود را محاسبه کنید.

پاسخ ۸

$$P(Exit\ With\ Failure) = (1-p)^H + (1-p)^H(p(1-p)) + (1-p)^H(p^2(1-p)^2) + \dots$$
$$\Rightarrow P(Exit\ With\ Failure) = \sum_{n=0}^{\infty}(1-p)^H(p^n(1-p)^n)$$

یک روش ساده برای مدل کردن تغییرات قیمت در بازار سرمایه، به این صورت است که تغییرات در هر روز را یک متغیر تصادفی با توزیع گوسی و میانگین صفر و مستقل از روزهای دیگر در نظر می گیریم. یعنی:

$$X_i = X_{i-1} + D_i \qquad D_i \sim \mathcal{N}(0, \sigma^2)$$

که X_i قیمت در روز i ام است.

پرسش شبیه سازی ۲

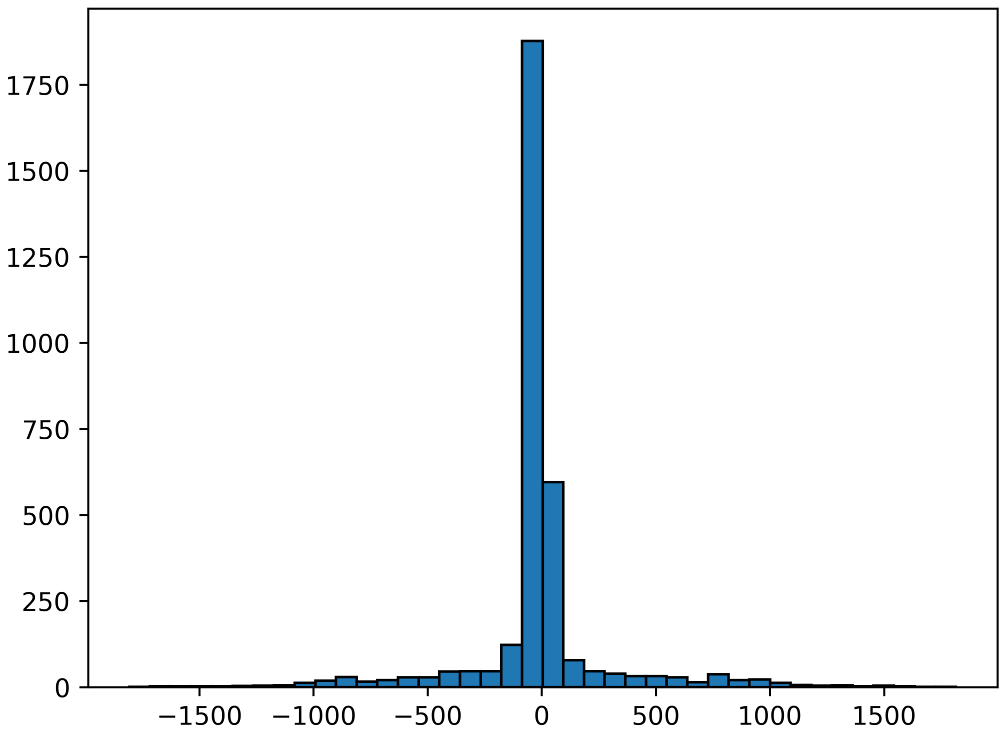
پارامتر D را با استفاده از دیتای سهام استخراج کنید و هیستوگرام آن را رسم نمایید. همچنین میانگین و انحراف معیار آن را بدست آورید.

قطعه کد مربوط به این بخش به صورت زیر می باشد:

```
df = pd.read_excel("data.xls")

last_price = df["      "]
first_price = df["      "]

Ds = [lp - fp for lp, fp in zip(last_price, first_price)]

plt.hist(Ds,density=False, bins=40,edgecolor='black')
plt.show()

mean, standard_deviation = statistics.mean(Ds), statistics.stdev(Ds)
print(mean, standard_deviation)
```

خروجی این بخش به صورت زیر می باشد:

$$\begin{aligned} Mean &= -1.285 \\ Standard\ Deviation &= 300.627 \end{aligned}$$



پرسش تئوری ۹

توزیع احتمال شرطی قیمت فردا را به شرط مشاهدهٔ قیمت امروز بدست آورید.

پاسخ ۹

به طور شهودی میتوان گفت که

$$X_{i+1}|X_i \sim \mathcal{N}(X_i, \sigma^2)$$

زیراکه قیمت روز $i+1$ ام برابر است با قیمت روز i ام به علاوه مقداری که از توزیع $\mathcal{N}(0, \sigma^2)$ به دست می آید که دارای امید ریاضی صفر است.

پرسش تئوری ۱۰

توزیع احتمال تغییرات ماهانهٔ قیمت را بدست آورید.

پاسخ ۱۰

اگر ماه را n روز در نظر بگیریم و برای تغییرات قیمت در طی n روز خواهیم داشت:

$$X_{m+n} - Xm + n - 1 + X_{m+n-1} - X_{m+n-2} + ... + X_{m+1} - X_m$$
$$= D_{m+n-1} + D_{m+n-2} + ... + D_m$$

از آنجایی که تغییرات قیمت روزانه نسبت به یکدیگر مستقل هستند میتوان گفت که توزیع احتمال تغییرات ماهانهٔ قیمت با فرض n روزه بودن ماه به صورت زیر است:

$$\sim \mathcal{N}(0, n\sigma^2)$$

پرسش تئوری ۱۱

فرض کنید توزیع احتمال قیمت امروز، یکنواخت باشد. یعنی $X_i \sim \mathcal{U}(a, b)$. در این صورت تابع چگالی احتمال قیمت فردا،یعنی $f_{X_{i+1}}(x_{i+1})$ را محاسبه کنید.

پاسخ ۱۱

از آنجایی که X_i و D_{i+1} نسب به یکدیگر مستقل هستند می توان گفت که:

$$f_{X_{i+1}}(x_{i+1}) = f_X(x_{i+1}) * f_D(x_{i+1})$$

یکی از پارامترهای مهم در تحلیل سهام، «نوسان روزانه» است. این پارامتر به صورت زیر تعریف می شود:

$$Z = \frac{P_{open} - P_{close}}{P_{open}} \times 100$$

که P_{open} قیمت باز شدن و P_{close} قیمت بسته شدن سهام می باشد.

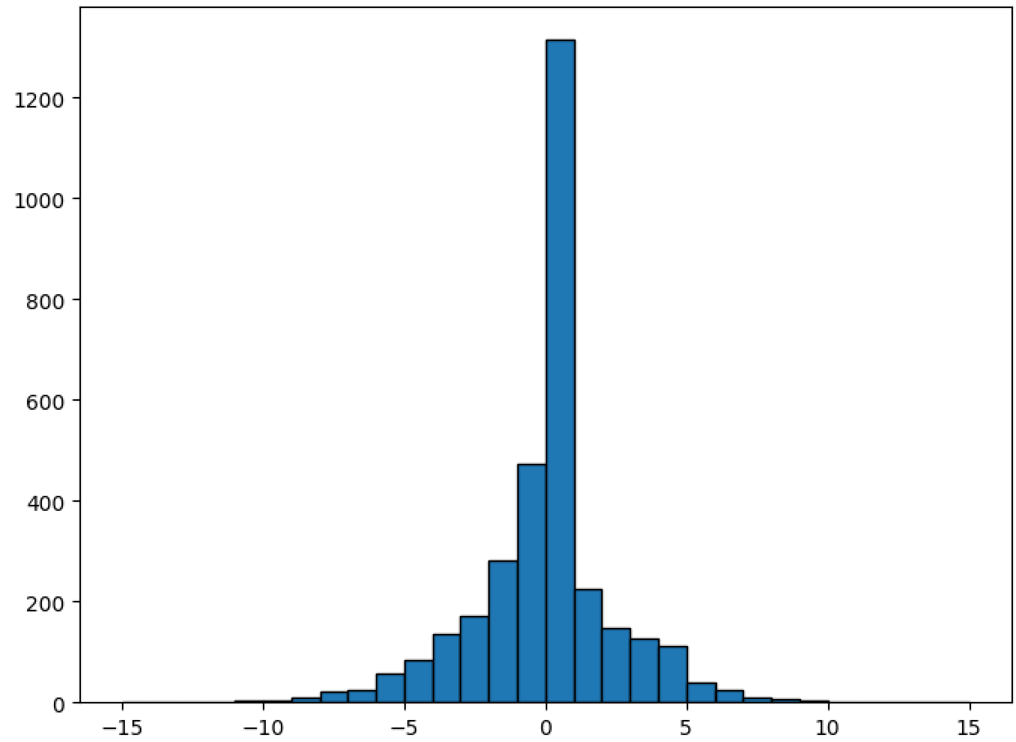
پرسش شبیه سازی ۳

پارامتر نوسان روزانه را با استفاده از دیتای سهام استخراج کنید و هیستوگرام این پارامتر را در بازهٔ [-۱۵، ۱۵] و در ۳۰ پنجره رسم کنید.

قطعه کد این بخش به صورت زیر می باشد:

```
df = pd.read_excel("data.xls")

p_close = df["      "].to_list()
p_open = df["      "].to_list()

z = [(a-b)/a*100 for a,b in zip(p_close, p_open)]

plt.hist(z,density=False,range=[-15,15],bins=30,edgecolor='black')
plt.show()
```

خروجی به صورت زیر خواهد بود:



۳ قانون اعداد بزرگ

در این بخش میخواهیم قانون اعداد بزرگ را برای متغیرهای تصادفی iid ساخته شده از توزیع های مختلف بررسی کنیم. اگر متغیرهای تصادفی تولید شده از یک توزیع را با X_i ها نشان دهیم مقدار متوسط برای n تا متغیر به صورت زیر میباشد:

$$\overline{X_n} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

پرسش شبیه سازی ٤

برای توزیع های زیر به ازای ۱۰۰،...، ۲، ۱ = n متغیر تصادفی iid تولید شده، همگرا شدن $\overline{X_n} = \mathbb{E}[X]$ بررسی و رسم کنید.

$Beta(2, 2)$
$Lognormal(0, 0.5)$
$Gamma(5, 0.5)$
$Poisson(4)$
$Exponential(1)$
$Exponential(1)$

قطعه کد این بخش به صورت زیر می باشد:

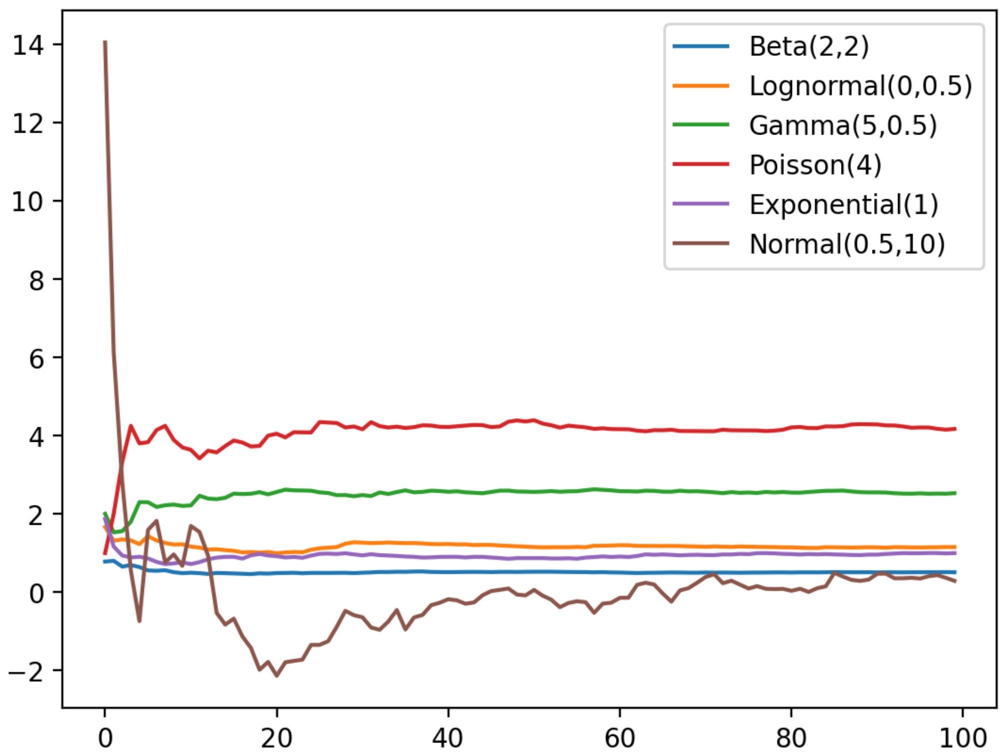
```
#Beta(2,2)
samples1 = np.random.beta(2, 2, size=100)
Xis1 =[]

for i in range(100):
    Xis1.append(sum(samples1[0:i+1])/(i+1))

plt.plot(Xis1)

#Lognormal(0,0.5)
samples2 = np.random.lognormal(0, 0.5, size=100)
Xis2 =[]

for i in range(100):
    Xis2.append(sum(samples2[0:i+1])/(i+1))

plt.plot(Xis2)

#Gamma(5,0.5)
samples3 = np.random.gamma(5, 0.5, size=100)
Xis3 =[]

for i in range(100):
    Xis3.append(sum(samples3[0:i+1])/(i+1))

plt.plot(Xis3)

#Poisson(4)
samples4 = np.random.poisson(4, size=100)
Xis4 =[]

for i in range(100):
    Xis4.append(sum(samples4[0:i+1])/(i+1))

plt.plot(Xis4)

#Exponential(1)
samples5 = np.random.exponential(1, size=100)
Xis5 =[]

for i in range(100):
    Xis5.append(sum(samples5[0:i+1])/(i+1))

plt.plot(Xis5)

#Normal(0.5,10)
samples6 = np.random.normal(0.5, 10, size=100)
Xis6 =[]

for i in range(100):
    Xis6.append(sum(samples6[0:i+1])/(i+1))

plt.plot(Xis6)
```

```python
plt.gcf().set_dpi(200)
plt.legend(['Beta(2,2)','Lognormal(0,0.5)','Gamma(5,0.5)','Poisson(4)','Exponential(1)','Normal
    (0.5,10)'])
```

برای توزیع های داده شده داریم:

$Distribution$	$\mathbb{E}[X]$
$Beta(a,b)$	$\frac{a}{a+b}$
$Lognormal(a,b)$	$e^{a+\frac{b^2}{2}}$
$Gamma(a,b)$	$a \times b$
$Poisson(a)$	a
$Exponential(a)$	$\frac{1}{a}$
$Normal(a,b)$	a

خروجی قطعه کد بالا به صورت زیر خواهد بود:



همانطور که مشخص است، به ازای مقدار های بزرگ n، مقدار $\overline{X_n}$ برای $Beta(2,2)$ به ۰.۵، برای $Lognormal(0,0.5)$ به ۱.۱۳، برای $Gamma(5,0.5)$ به ۲.۵، برای $Poisson(4)$ به ۴، برای $Exponential(1)$ به ۱ و برای $Normal(0.5,10)$ به ۰.۵ میل میکند.

پرسش شبیه سازی ۵

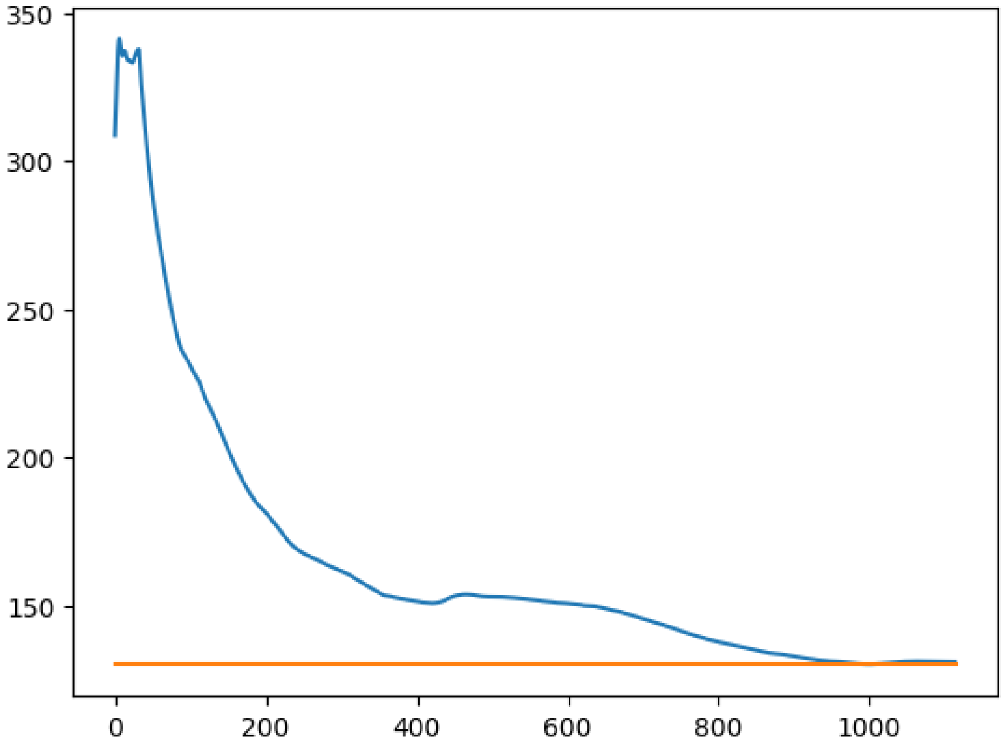
با استفاده از داده های مربوط به معاملات سهام شرکتی که در اختیار دارید ابتدا میانگین قیمت پایانی روز ها تا قبل از سال ۲۰۱۱ را بدست آورید و سپس مانند قسمت قبل برای داده های تا قبل از سال ۲۰۱۱ همگرا شدن میانگین داده ها به عدد بدست آمده را با رسم نمودار نشان دهید.

قطعه کد مربوط به این به صورت زیر می باشد:

```python
import matplotlib.pyplot as plt

df = pd.read_excel("data.xls")

last_price = df["      "].tolist()

datas=[]

Xn_bar=sum(last_price[0:1115])/1115
print("Average Final Price : "+str(round(Xn_bar,2)))

for i in range(1114):
    datas.append(sum(last_price[0:i+1])/(i+1))

plt.plot(datas)
```

```

x=[0,1114]
y=[130,130]
plt.plot(x,y)
plt.show()
```

میانگین قیمت پایانی روز ها تا قبل از سال ۲۰۱۱ تقریبا برابر ۱۳۰.۹۴ می باشد و خروجی به صورت زیر خواهد بود: (خط نارنجی رنگ نشان دهنده ی خط $y = 130$ می باشد.)



۴ حد مرکزی

برای هرکدام از توزیع های قسمت قبل مراحل زیر را اجرا کنید:

۱.توزیع انتخاب شده را F و میانگین و واریانس آن را μ و σ^2 بنامید.

۲ .دنباله مستقل از متغیر های تصادفی $\{\{X_i^{(j)}\}_{i=1}^{n}\}_{j=1}^{k}$ با توزیع F تولید کنید.

۳.برای هر کدام از این دانباله ها $Y_n := \sqrt{n}(\overline{X_n} - \mu)$ را محاسبه کنید و در یک آرایه به نام Y ذخیره کنید. بدیهتا این آرایه k عضو خواهد داشت.

۴.هیستوگرام آرایه Y را رسم کنید. ۵.در همان نمودار تابع توزیع نرمال $\mathcal{N}(0, \sigma^2)$ را نیز رسم کنید.آن را با هیستوگرام مقابسه کنید.

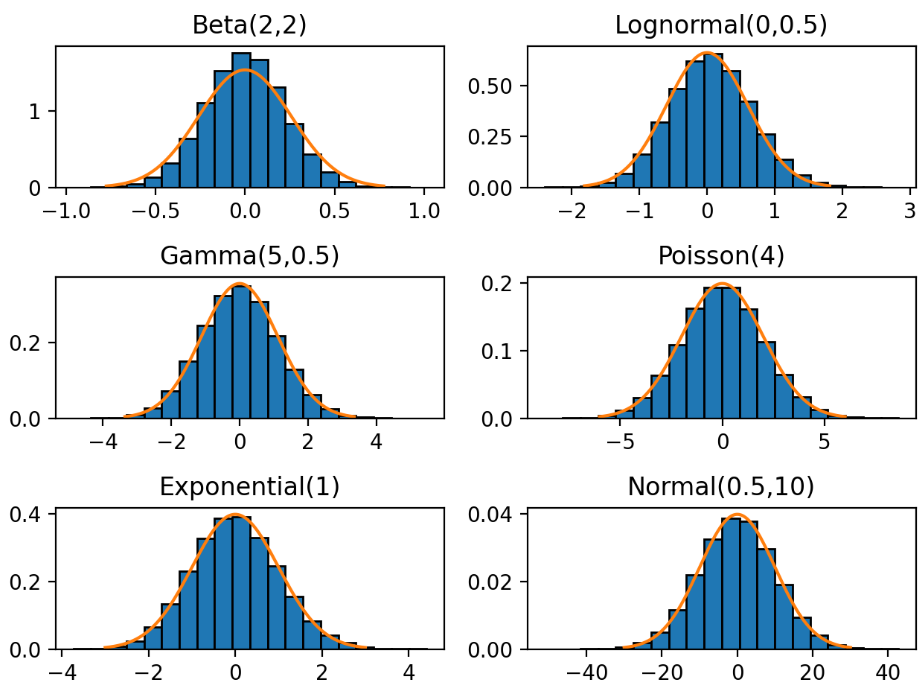
در این سوال n را برابر با ۳۰۰ و k برا برابر با ۱۰۰۰۰۰ بگیرید.

قطعه کد این بخش به صورت زیر می باشد:

```python
def Y_calculator(n,mu,X):
    return np.sqrt(n)*((1/n)*sum(X)-mu)

fig,axs = plt.subplots(3, 2)

#Beta(2,2)
sequences1 = np.zeros((100000, 300))
for i in range(100000):
    sequences1[i] = beta.rvs(a=2, b=2, size=300)
Y1=[]
for i in range(100000):
    Y1.append(Y_calculator(300,0.5,sequences1[i]))

axs[0,0].hist(Y1,density=True, bins=20,edgecolor='black')
mu1 = 0
variance1 = 1/15
sigma1 = np.sqrt(variance1)
x1 = np.linspace(mu1 - 3*sigma1, mu1 + 3*sigma1, 200)
y1 = norm.pdf(x1, mu1, sigma1)
axs[0,0].plot(x1, y1)
axs[0,0].set_title('Beta(2,2)')

#Lognormal(0,0.5)
sequences2 = np.zeros((100000, 300))
for i in range(100000):
    sequences2[i] = lognorm.rvs(s=0.5, scale=np.exp(0), size=300)
Y2=[]
for i in range(100000):
    Y2.append(Y_calculator(300,1.13,sequences2[i]))

axs[0,1].hist(Y2,density=True, bins=20,edgecolor='black')
mu2 = 0
variance2 = 0.364
sigma2 = np.sqrt(variance2)
x2 = np.linspace(mu2 - 3*sigma2, mu2 + 3*sigma2, 200)
y2 = norm.pdf(x2, mu2, sigma2)
axs[0,1].plot(x2, y2)
axs[0,1].set_title('Lognormal(0,0.5)')

#Gamma(5,0.5)
sequences3 = np.zeros((100000, 300))
for i in range(100000):
    sequences3[i] = gamma.rvs(a=5, scale=0.5, size=300)
Y3=[]
for i in range(100000):
    Y3.append(Y_calculator(300,2.5,sequences3[i]))

axs[1,0].hist(Y3,density=True, bins=20,edgecolor='black')
mu3 = 0
variance3 = 1.25
sigma3 = np.sqrt(variance3)
x3 = np.linspace(mu3 - 3*sigma3, mu3 + 3*sigma3, 200)
y3 = norm.pdf(x3, mu3, sigma3)
axs[1,0].plot(x3, y3)
axs[1,0].set_title('Gamma(5,0.5)')

#Poisson(4)
sequences4 = np.zeros((100000, 300))
for i in range(100000):
    sequences4[i] = poisson.rvs(mu=4, size=300)
Y4=[]
for i in range(100000):
    Y4.append(Y_calculator(300,4,sequences4[i]))
```

```python

axs[1,1].hist(Y4,density=True, bins=20,edgecolor='black')
mu4 = 0
variance4 = 4
sigma4 = np.sqrt(variance4)
x4 = np.linspace(mu4 - 3*sigma4, mu4 + 3*sigma4, 200)
y4 = norm.pdf(x4, mu4, sigma4)
axs[1,1].plot(x4, y4)
axs[1,1].set_title('Poisson(4)')

#Exponential(1)
sequences5 = np.zeros((100000, 300))
for i in range(100000):
    sequences5[i] = expon.rvs(scale=1, size=300)
Y5=[]
for i in range(100000):
    Y5.append(Y_calculator(300,1,sequences5[i]))

axs[2,0].hist(Y5,density=True, bins=20,edgecolor='black')
mu5 = 0
variance5 = 1
sigma5 = np.sqrt(variance5)
x5 = np.linspace(mu5 - 3*sigma5, mu5 + 3*sigma5, 200)
y5 = norm.pdf(x5, mu5, sigma5)
axs[2,0].plot(x5, y5)
axs[2,0].set_title('Exponential(1)')

#Normal(0.5,10)
sequences6 = np.zeros((100000, 300))
for i in range(100000):
    sequences6[i] = norm.rvs(loc=0.5, scale=10, size=300)
Y6=[]
for i in range(100000):
    Y6.append(Y_calculator(300,0.5,sequences6[i]))

axs[2,1].hist(Y6,density=True, bins=20,edgecolor='black')
mu6 = 0
variance6 = 100
sigma6 = np.sqrt(variance6)
x6 = np.linspace(mu6 - 3*sigma6, mu6 + 3*sigma6, 200)
y6 = norm.pdf(x6, mu6, sigma6)
axs[2,1].plot(x6, y6)
axs[2,1].set_title('Normal(0.5,10)')

fig.set_dpi(200)
fig.tight_layout()
```

خروجی به صورت زیر خواهد بود:



آنچه در بالا به دست آمد در واقع تصدیق کننده فرم $Lindeberg - Lvy$ قضیه حد مرکزی است که بیان می دارد با فرض اینکه $\{X_1, ..., X_N, ...\}$ مجموعه ای از متغیر های تصادفی iid با امید μ و واریانس $\sigma^2 < \infty$ باشد؛ زمانی که n به سمت بینهایت می رود، متغیر تصادفی $\sqrt{n}(\bar{X}_n - \mu)$ به $\mathcal{N}(0, \sigma^2)$ همگرا میشود.

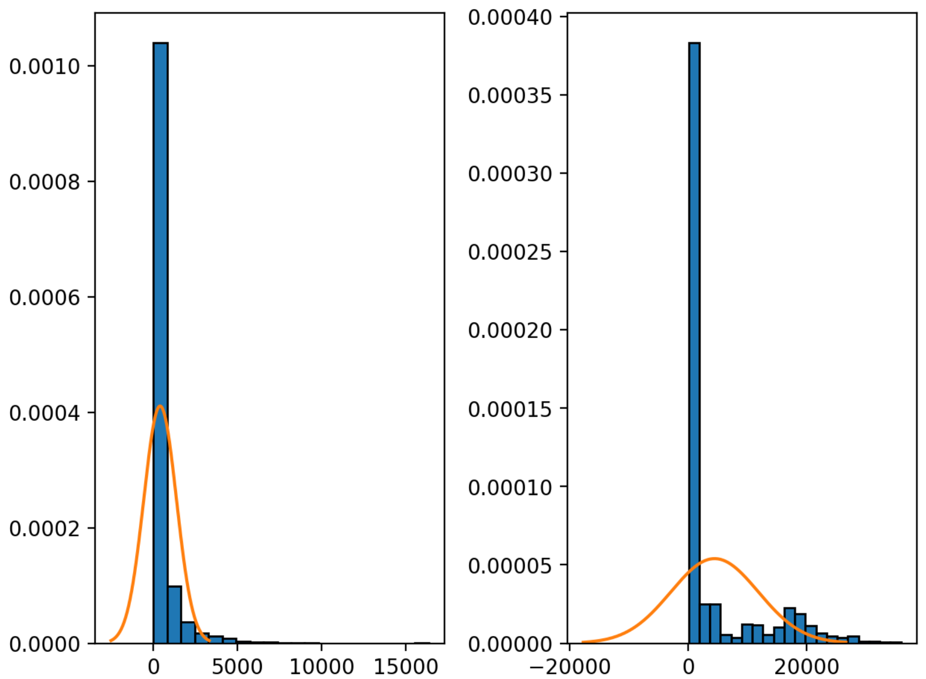
پرسش شبیه سازی ٦

میانگین و واریانس تعداد معاملات را بدست بیاورید. سپس هیستوگرام تعداد معاملات در روز را رسم کرده و در کنار توزیع نرمال با همان میانگین و واریانس را رسم کنید. اینکار را برای قیمت پایانی تا قبل از سال ۲۰۱۱ نیز تکرار کنید.

قطعه کد این بخش به صورت زیر می باشد:

```
df = pd.read_excel("data.xls")
fig,axs = plt.subplots(1, 2)

transactions = df["       "].tolist()

mean = statistics.mean(transactions)
variance = statistics.variance(transactions)

axs[0].hist(transactions,density=True, bins=20,edgecolor='black')
sigma = np.sqrt(variance)
x = np.linspace(mean - 3*sigma, mean + 3*sigma, 200)
y = norm.pdf(x, mean, sigma)
axs[0].plot(x, y)

print("Mean : "+str(round(mean,2)))
print("Variance : "+str(round(variance,2)))

last_price = df["      "].tolist()

mean = statistics.mean(last_price)
variance = statistics.variance(last_price)

axs[1].hist(last_price,density=True, bins=20,edgecolor='black')
sigma = np.sqrt(variance)
x = np.linspace(mean - 3*sigma, mean + 3*sigma, 200)
y = norm.pdf(x, mean, sigma)
axs[1].plot(x, y)

fig.set_dpi(200)
fig.tight_layout()
```

مقدار میانگین برابر 404.47 و واریانس برابر 943899.99 می باشد و خروجی به صورت زیر است:



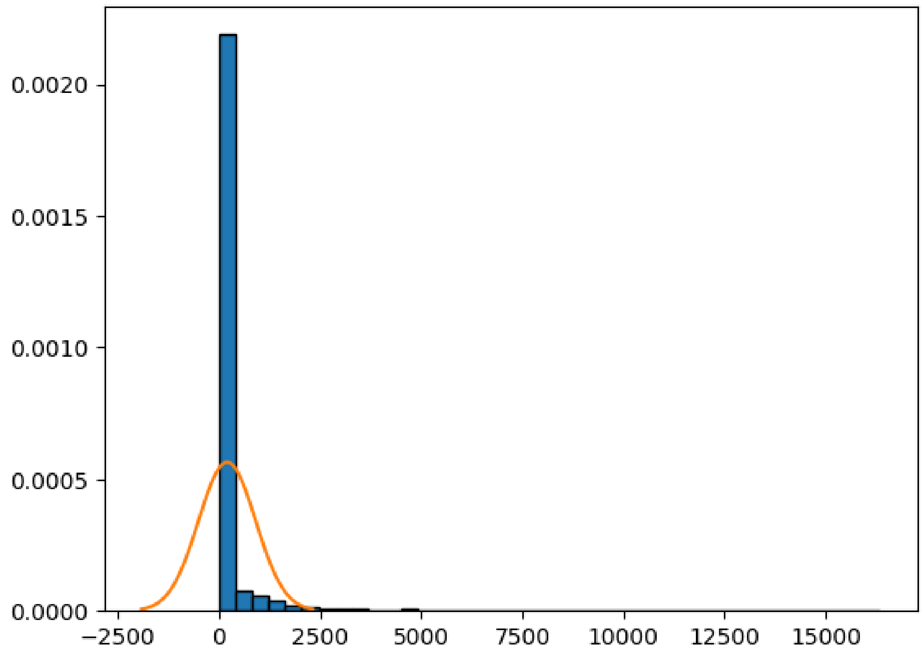
پرسش شبیه سازی ۷

اینبار عملیات قبلی را برای تعداد معاملات تکرار کنید اما روزهایی که معامله ای انجام نشده را نیز لحاظ کنید. و تغییرات واریانس و میانگین را توجیه کنید.

قطعه کد این بخش به صورت زیر است:

```
df = pd.read_excel("data.xls")

transactions = df["        "].tolist()
transactions.extend([0] * (6762-len(transactions)))

mean = statistics.mean(transactions)
variance = statistics.variance(transactions)

plt.hist(transactions,density=True, bins=40,edgecolor='black')
sigma = np.sqrt(variance)
x = np.linspace(mean - 3*sigma, mean + 3*sigma, 100)
y = norm.pdf(x, mean, sigma)
plt.plot(x, y)
plt.show()

print("Mean : "+str(round(mean,2)))
print("Variance : "+str(round(variance,2)))
```

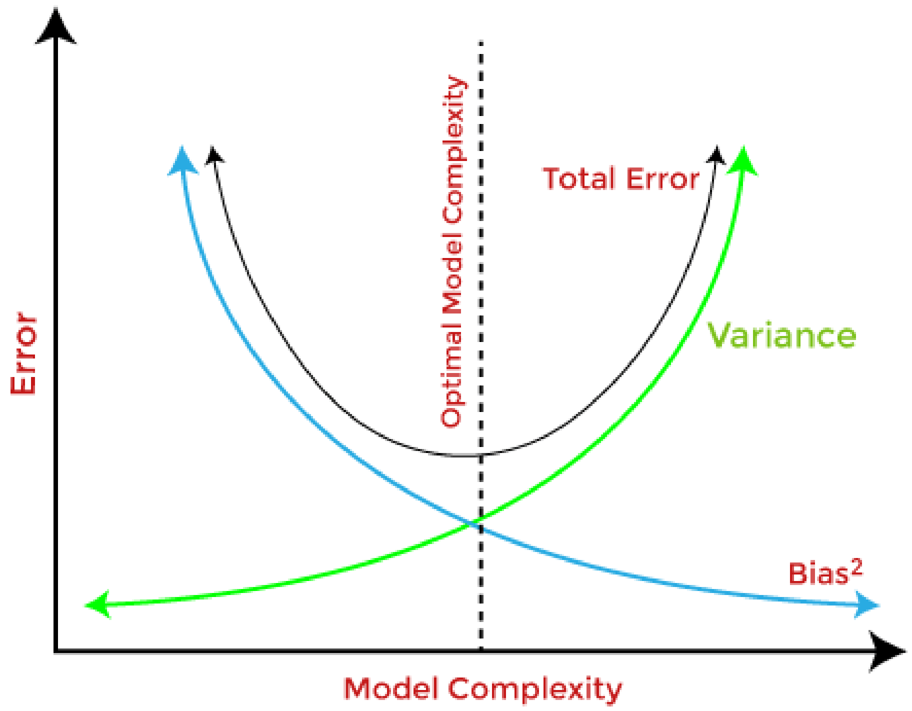
خروجی به صورت زیر خواهد بود:



مقدار میانگین به 196.67 تغییر پیدا کرده است.دلیل این امر هم این است که قبل از اضافه شدن روز هایی که معامله ای در آن انجام نشده (طبیعتا) امید ریاضی بزرگ تر از صفر است و در نتیجه با اضافه شدن مقداری صفر ، امید ریاضی کمتر شده. از طرفی مقدار واریانس به 499771.25 تغییر پیدا کرده است. در واقع با اضافه کردن این صفر ها، داده هایی که در نزدیکی میانگین قرار داشته اند افزایش یافته و واریانس کم شده است.

۵ بایاس و واریانس

در مدل های آماری و بخصوص الگوریتم های یادگیری ماشین، مسئله موازنه واریانس و بایاس (اریبی) مورد بحث قرار می گیرد.در اغلب مدل های پیش بینی کننده وجود بایاس کوچک برای پارامترها موجب واریانس بزرگ برای مدل خواهد شد. البته برعکس این حالت نیز وجود دارد، به این معنی که با کوچک کردن واریانس مدل، با مشکل بزرگ شدن بایاس یا اریبی پارامترها مواجه خواهیم شد.
مسئله اصلی آن است که در یک مدل مناسب، هم بایاس و هم واریانس باید حداقل ممکن باشند.ولی متاسفانه کمینه سازی هردو این شاخص ها به شکل توام امکان پذیر نیست.چنین وضعیتی را «تناقض واریانس –اریبی» می نامند.
پس به این نتیجه رسیدیم نمی توان هم بایاس هم واریانس را به صورت توام کاهش داد.

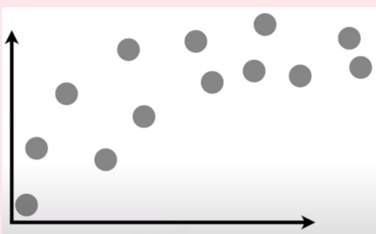


پرسش تئوری ۱۲

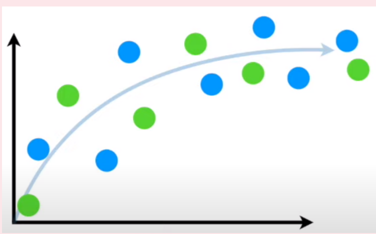
خطای بایاس چیست؟

پاسخ ۱۲

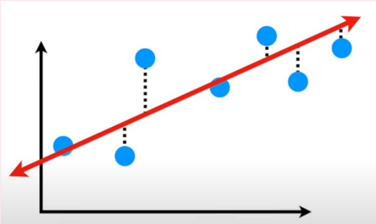
اگر فرض کنیم دیتایی که داریم به صورت زیر باشد:



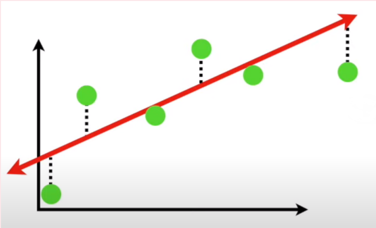
برای اینکه بتوانیم داده مورد نظر را به مدل کنیم و نتیجه حاصل را ارزیابی کنیم، نیاز است که داده ها را به دو بخش تقسیم کرده و از یک بخش برای مدل سازی ($Training\ Set$) و از بخش دیگر برای صحت سنجی مدل ($Testing\ Set$) ارائه شده استفاده کنیم. برای مدل نمونه ارائه شده در بالا این کار با میتوانیم به صورت زیر انجام دهیم:



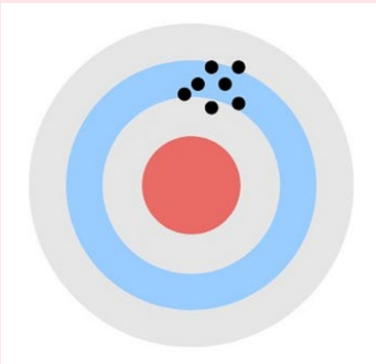
در اینجا نقطعه های آبی $Training\ Set$ و نقاط سبز $Testing\ Set$ می باشند (نسبت $Training\ Set$ و $Testing\ Set$ به صورت حدودا ۸۰ درصد به ۲۰ درصد می باشد؛ اما اینجا به دلیل اینکه یک مدل انتزاعی می باشد، نسبت ۵۰ به ۵۰ انتخاب شده است.) حال یکی از مدل سازی هایی که میتوانیم انجام دهیم، استفاده از خط راست برای مدل سازی می باشد:



همانطور که مشخص است، مدل ارائه شده، به خوبی به روی داده ها فیت نشده است. حال اگر مدل را برای $Testing\ Set$ تست کنیم خواهیم داشت:



با توجه به نتیجه حاصل می بینیم که مدل ارائه شده روی $Testing\ Set$ نیز به خوبی فیت نشده است اما اگر فاصله نقاط در $Training\ Set$ تا مدل را حساب کرده و این مقدار را برای نقاط $Testing\ Set$ نیز حساب کنیم، می بینیم که تغییر آنچنانی در این مقدار حاصل نشده است. به عبارت دیگر خطای بایاس در اینجا زیاد می باشد. در واقع نا توانی یک مدل در به درستی پیش بینی کردن و فیت شدن روی دیتا را خطای بایاس میگویند.(در اینجا خطای واریانس کم است) تصویر زیر در راستای مطالب ارائه شده در بالا میتواند کمک کننده باشد: (اگر فیت شدن مدل را نزدیک بودن به هدف در سیبل زیر در نظر بگیریم، میبینیم که مدل در پیش بینی به خوبی عمل نکرده است ولی تست کردن دیتا های مختلف نتاجی نزدیک به هم را به جای گذاشته است(خطای واریانس کم))

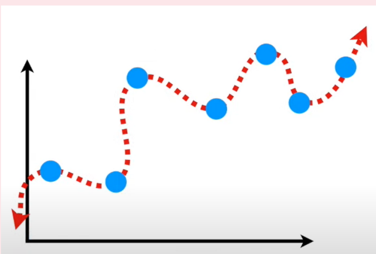


پرسش تئوری ۱۳

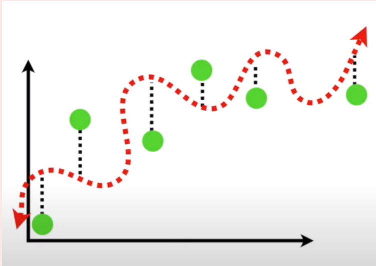
خطای واریانس چیست؟

پاسخ ۱۳

اگر همان مجموعه دیتا در مثال بخش قبل را در نظر بگیریم، نوعی دیگر از مدل سازی میتواند به صورت زیر باشد:

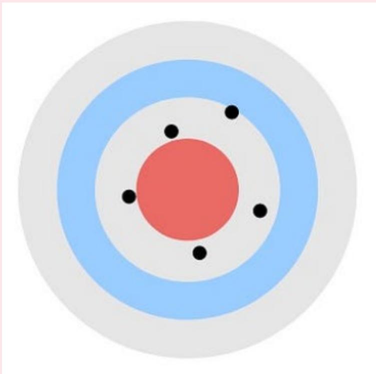


همانطور که مشخص است، مدل ارائه شده به خوبی به روی داده ها فیت شده است. حال اگر مدل را برای $Testing\ Set$ تست کنیم خواهیم داشت:

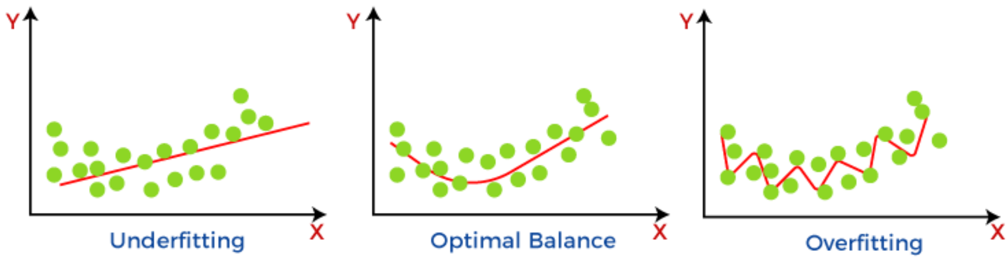


نتیجه بالا نشان میدهد که مدل رائه شده دارای خطای زیادی در پیش بینی $Testing\ Set$ می باشد. در اینجا خطای واریانس زیاد می باشد. در واقع ، به تفاوت زیاد در فیت شدن به ازای مجموعه داده های متفاوت خطای واریانس گفته می شود.(در اینجا خطای بایاس کم می باشد)

تصویر زیر در راستای مطالب ارائه شده در بالا میتواند کمک کننده باشد: (اگر فیت شدن مدل را نزدیک بودن به هدف در سیبل زیر در نظر بگیریم، می بینیم که به ازای مجموعه های مختلفی از داده ها، نتایج پراکنده ای حاصل شده است)



احتمالا هنگامی که درباره ی خطای اریبی و واریانس تحقیق میکرده اید با عبارت های $under\ fitting$ و $over\ fitting$ آشنا شده اید.اغلب بزرگ بودن خطای اریبی موجب به کم برازش بودن یا $under\ fitting$ میشود و همچنین افزایش واریانس موجب بیش برازش یا $over\ fitting$ میشود.



پرسش تئوری ١٤

مشکل اینکه فقط خطای بایاس را حداقل کرد و به واریانس توجه نکرد چیست؟

پاسخ ١۴

در این صورت ممکن است که مدل ارائه شده به ازای $Training\ Set$ به خوبی عمل کند، اما برای $Testing\ Set$ با خطای بسیار زیادی همراه شود و داده ها را به خوبی مدل نکند.

پرسش تئوری ١٥

مشکل اینکه فقط خطای واریانس را حداقل کرد و به بایاس توجه نکرد چیست؟

پاسخ ١۵

به ازای $Testing\ Set$ ممکن است که خطای یکسانی در پیش بینی داده ها داشته باشیم اما این مقدار خطا لزوما کم نیست و ممکن است مدل ارائه شده به خوبی بر روی داده ها فیت نشود.

پرسش تئوری ١٦

چند نمونه از کاربرد های موازنه بایاس و واریانس را بیان کنید.

پاسخ ۱۶

یکی از کاربرد های آن در تحلیل رگرسیون می باشد که در آن به دنبال پیدا کردن n (درجه چند جمله ای) بیهنه برای تقریب داده ها هستیم یا در تحلیل تصویر ، از موازنه بایاس و واریانس برای پیدا کردن الگو در تصویر استفاده می کنیم(با کاهش خطای بایاس ، به نتایج بهتری در الگو یابی می رسیم) یا در جایی دیگر ، برای تحلیل متن، میتوانیم از آن برای تشخیص احساس، شباهت دو متن و پرسش و پاسخ استفاده کنیم.

پرسش شبیه سازی ۸

درباره ی داده های جمع آوری شده و نمونه ای که انتخاب کرده اید توضیح دهید.

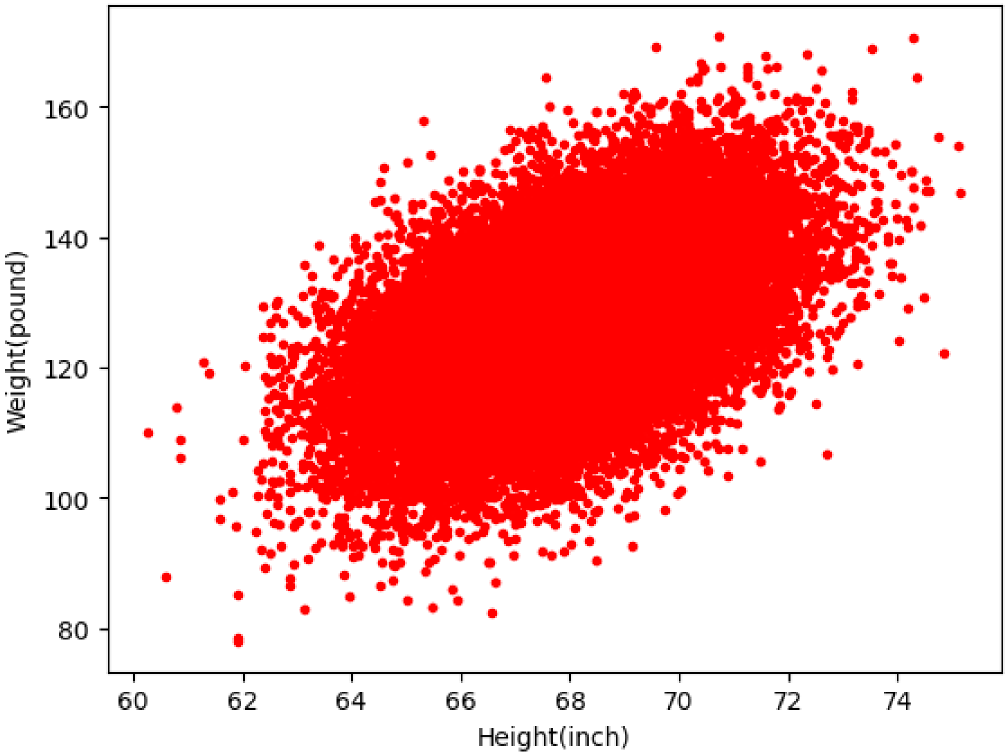
دیتا استفاده شده در این بخش مربوط به قد و وزن حدودا ۲۵۰۰۰ نمونه می باشد. (فایل داده ها در کنار فایل گزارش با نام $HandW.csv$ موجود می باشد.)

پرسش شبیه سازی ۹

نمودار داده های خود را رسم کنید.

قطعه کد این بخش به صوت زیر می باشد:

```
df = pd.read_csv("HandW.csv")

Height = df['Height'].to_list()
Weight = df['Weight'].to_list()

plt.plot(Height,Weight,'r.')
plt.xlabel("Height(inch)")
plt.ylabel("Weight(pound)")
plt.show()
```



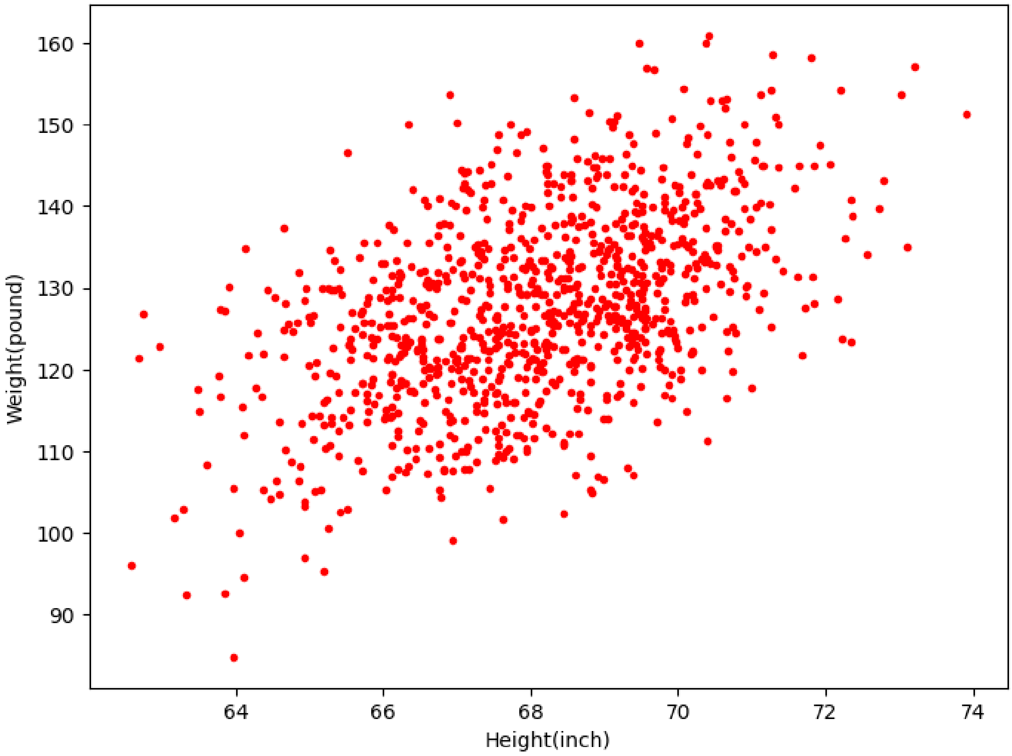
پرسش شبیه سازی ۱۰

نمودار نمونه انتخاب شده را رسم کنید.

۱۰۰۰ نمونه از داده ها را به صورت تصادفی انتخاب میکنیم.
قطعه کد این بخش به صورت زیر می باشد:

```
df = pd.read_csv("HandW.csv")

np.random.seed(400101272)
sample_datas = df.sample(n=1000)
```

```python

Height = sample_datas['Height'].to_list()
Weight = sample_datas['Weight'].to_list()

plt.plot(Height,Weight,'r.')
plt.xlabel("Height(inch)")
plt.ylabel("Weight(pound)")
plt.show()
```



اکنون از شما میخواهیم به کمک نمونه ای که انتخاب کرده اید کل داده ها را به چند حالتی که در زیر بیان شده است تقریب بزنید و نمودار آنها را رسم کنید:(یکی از شیوه هایی که می توانید با آن تقریب بزنید استفاده از درجه های مختلف تقریب رگرسیون چند جمله ای است)

پرسش شبیه سازی ۱۱

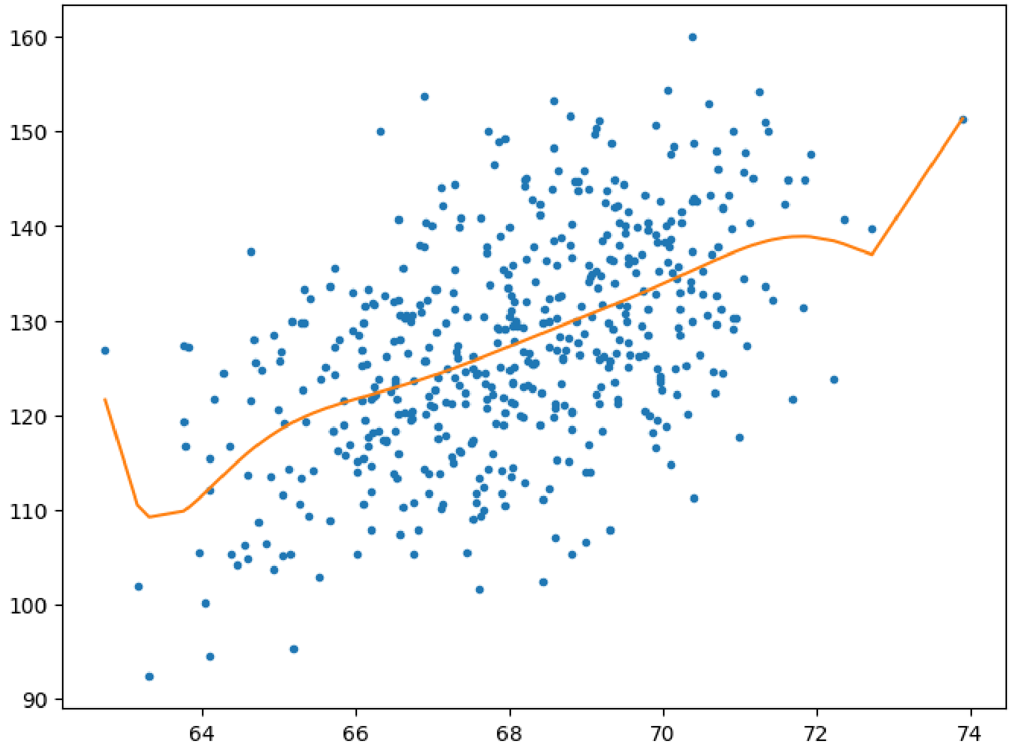
بایاس یا اریبی بسیار کم و در نتیجه واریانس نسبتا زیاد.

با استفاده از تقریب رگرسیون چند جمله ای با درجه ۱۰، ۵۰۰ داده های آزمایشی را مدل میکنیم. قطعه کد این بخش به صورت زیر می باشد:

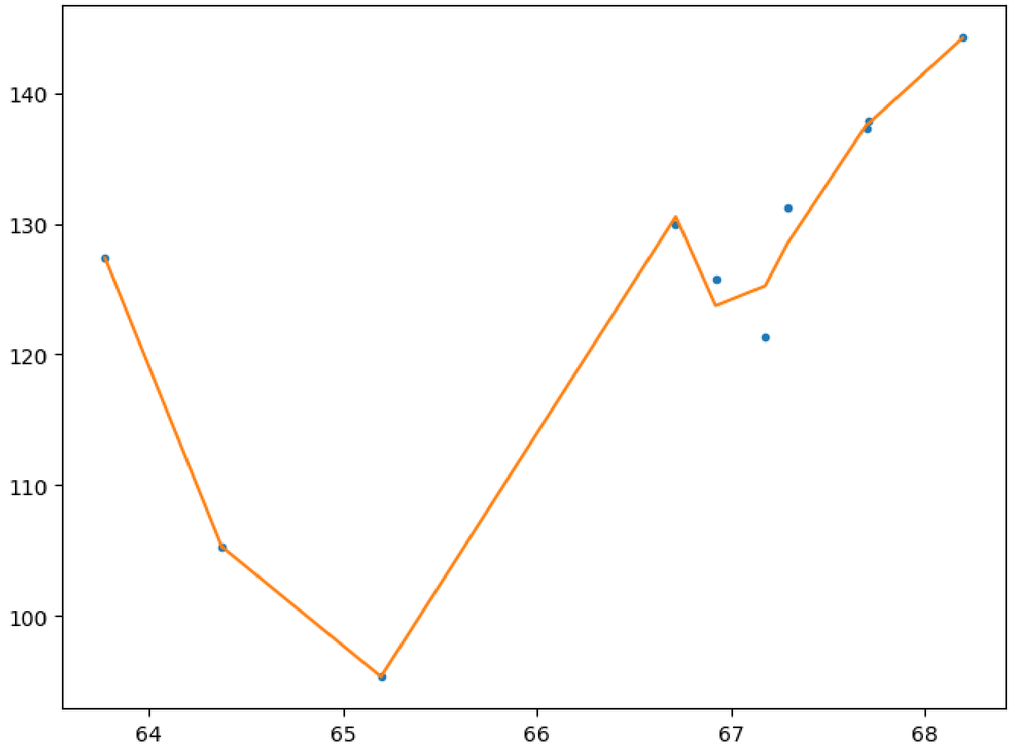
```python
def fit_poly( degree ):
    p = np.polyfit( curve.x, curve.y, deg = degree )
    curve['fit'] = np.polyval( p, curve.x )
    return plt.plot( curve.x, curve.fit, label='fit' )


n = 10
np.random.seed(400101272)
df = pd.read_csv("HandW.csv")

sample_datas = df.sample(n=500)
sample_datas_sorted = sample_datas.sort_values('Height', ascending=True)


Height = sample_datas_sorted['Height']
Weight = sample_datas_sorted['Weight']

x = np.array(sample_datas_sorted['Height'])
y = np.array(sample_datas_sorted['Weight'])
curve = pd.DataFrame(np.column_stack([x,y]),columns=['x','y'])
plt.plot(curve['x'],curve['y'],'.')

fit_poly(n)
```

در ابتدا سید را مقداری مشخصی قرار می دهیم تا به نتایج یکسانی دست یابیم.در ادامه با انتخاب ۵۰۰ نمونه تصادفی از داده های خود، با استفاده از تابع $fit-poly$ چند جمله ای با درجه n را روی داده ها فیت میکنیم. تابع $plotfit$ برای تطبیق داده ها با استفاده از روش کمترین مربعات استفاده می شود.خروجی این تابع ضرایب چند جمله ای ها خواهد بود. دستور $polyval$ نیز برای محاسبه چند جمله ای به ازای x های دلخواه میباشد(ورودی اول این دستور همان ضرایب چند جمله ای می باشند)

خروجی به صورت زیر خواهد بود:



در نمونه بالا خطای واریانس بالا می باشد، برای شهود بهتر ، مقدار $Training\ Set$ کاهش می دهیم. به ازای $n = 10$ به طور نمونه داریم:



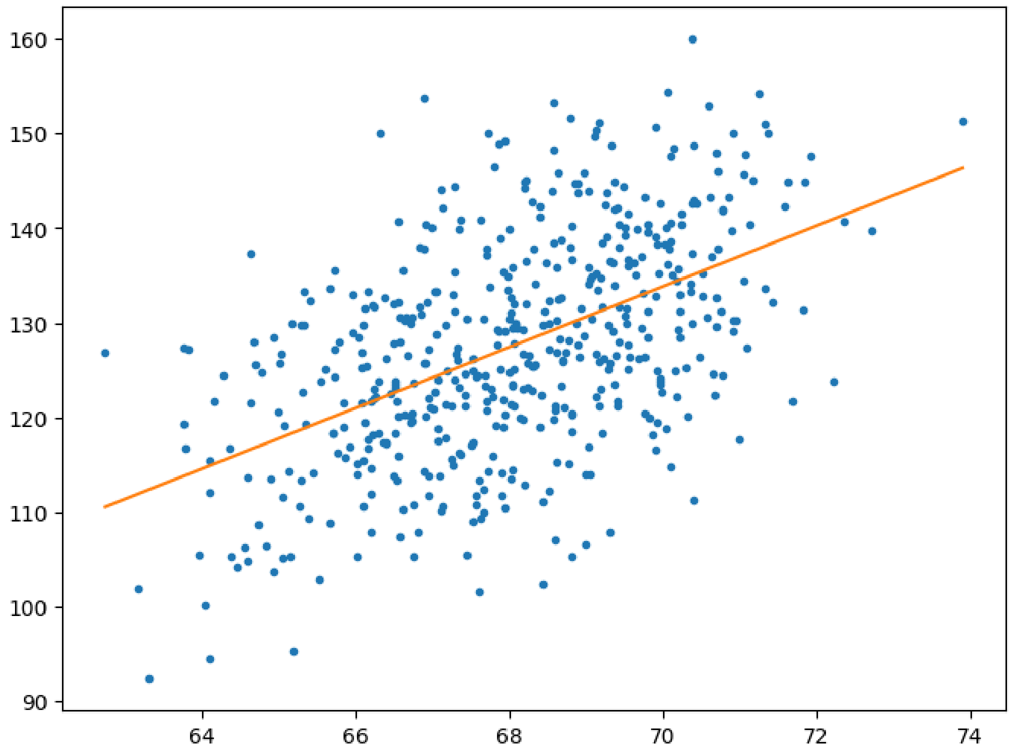
پرسش شبیه سازی ۱۲

بایاس یا اریبی بسیار کم و در نتیجه واریانس نسبتا زیاد.

برای کمترین میزان اریبی، درجه تقریب رگرسیون چند جمله ای را برابر ۱ قرار می دهیم.

```
def fit_poly( degree ):
    p = np.polyfit( curve.x, curve.y, deg = degree )
    curve['fit'] = np.polyval( p, curve.x )
    return plt.plot( curve.x, curve.fit, label='fit' )


n = 1
np.random.seed(400101272)
df = pd.read_csv("HandW.csv")

sample_datas = df.sample(n=500)
sample_datas_sorted = sample_datas.sort_values('Height', ascending=True)


Height = sample_datas_sorted['Height']
Weight = sample_datas_sorted['Weight']

x = np.array(sample_datas_sorted['Height'])
y = np.array(sample_datas_sorted['Weight'])
curve = pd.DataFrame(np.column_stack([x,y]),columns=['x','y'])
plt.plot(curve['x'],curve['y'],'.')

fit_poly(n)
```

خروجی به صورت زیر خواهد بود:



پرسش شبیه سازی ۱۳

به صورت متعادل به گونه ای که تعادل بین بایاس و واریانس حفظ شود.

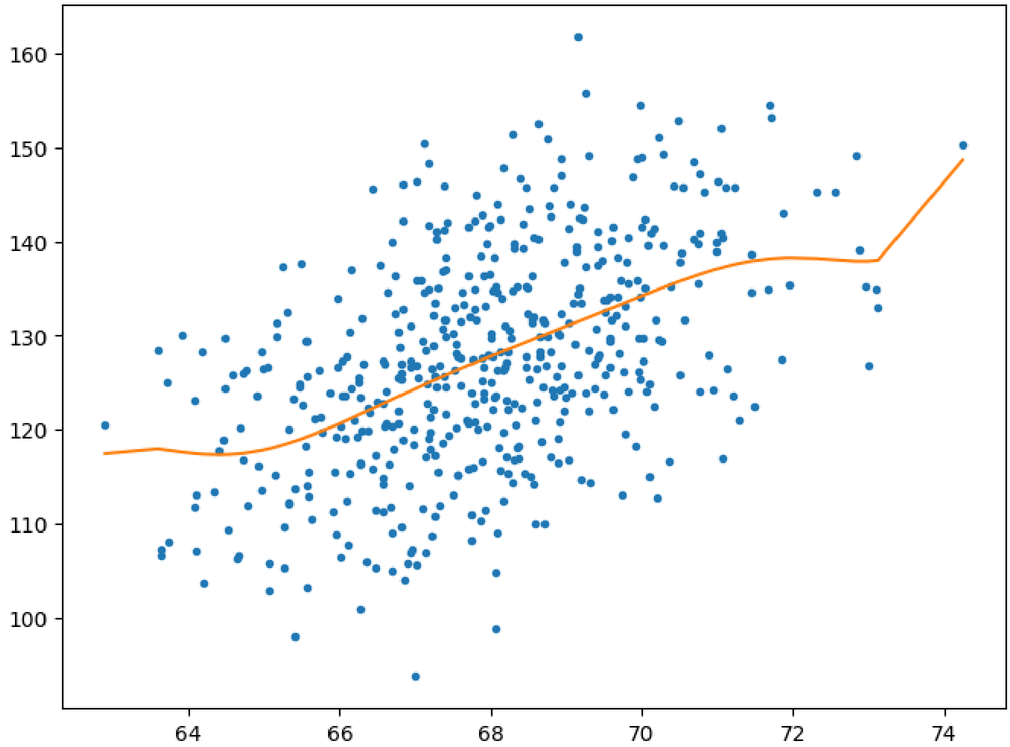
قطعه کد زیر برای پیدا کردن درجه بهینه تقریب رگرسیون چند جمله ای نوشته شده است:

```
def fit_poly( degree ):
    p = np.polyfit( curve.x, curve.y, deg = degree )
    curve['fit'] = np.polyval( p, curve.x )
    return plt.plot( curve.x, curve.fit, label='fit' )

df = pd.read_csv('HandW.csv')

dff = df.sort_values('Height', ascending=True)

np.random.seed(400101272)

```

```
X_train, X_test, y_train, y_test = train_test_split(dff['Weight'], dff['Height'], test_size=0.2)

degrees = [i for i in range(1, 101)]
mse_min=100
degree_min=0
for degree in degrees:
    poly_features = PolynomialFeatures(degree=degree)
    X_train_poly = poly_features.fit_transform(X_train.values.reshape(-1, 1))

    model = LinearRegression()
    model.fit(X_train_poly, y_train)

    X_test_poly = poly_features.transform(X_test.values.reshape(-1, 1))
    y_pred = model.predict(X_test_poly)
    mse = mean_squared_error(y_test, y_pred)
    if mse<mse_min:
        mse_min=mse
        degree_min = degree

print(f"Degree {degree_min}: MSE = {round(mse_min,2)}")

n = degree_min
sample_datas = df.sample(n=500)
sample_datas_sorted = sample_datas.sort_values('Height', ascending=True)


Height = sample_datas_sorted['Height']
Weight = sample_datas_sorted['Weight']

x = np.array(sample_datas_sorted['Height'])
y = np.array(sample_datas_sorted['Weight'])
curve = pd.DataFrame(np.column_stack([x,y]),columns=['x','y'])
plt.plot(curve['x'],curve['y'],'.')

fit_poly(n)
```

> در قطعه کد بالا، ابتدا با استفاده از $train_{t}est_{s}plit$ داده ها را به دو گروه $Training\ Set$ ، $Testing\ Set$ تقسیم میکنیم تا با استفاده از روش $Cross\ Validation$ بتوانیم عملکرد مدل خود را بررسی کنیم. سپس برای درجه های ۱ تا ۱۰۰، سعی در تقریب داده ها با استفاده از رگرسیون چند جمله ای داریم. در هر مرحله نیز میزان خطای هر کدام ($Mean\ Squared\ Error$) را محاسبه میکنیم تا در نهایت درجه از تقریب رگرسیون چند جمله ای که کمترین خطا را دارد را بیابیم.
> درون حلقه for ابتدا یک چند جمله ای با درجه مشخص ایجاد کرده به طوری که بر روی داده های $Training\ Set$ فیت بشود(طبیعتا اگر درجه چند جمله ای از تعداد نقاط بیشتر از یک واحد اختلاف داشته باشد، تقریب از تمام داد ها عبور نخواهد کرد) سپس با محاسبه y به ازای x های $Testing\ Set$ و تقریب چند جمله ای، مقدار خطای را محاسبه میکنیم. در نهایت با یافتن درجه متعلق به کمترین خطا، آن را مانند قبل رسم میکنم.

نتیجه قطعه کد بالا به صورت $Degree: 7: MSE = 2.68$ می باشد. در نتیجه به ازای $n = 3$ داریم:



پرسش تئوری ۱۷

درباره ی شیوه های تقریبی که در سه پرسش عملی قبلی استفاده کرده اید توضیح دهید.

پاسخ ۱۷

در قطعه کد های بالا از تقریب رگرسیون چند جمله ای برای مدل سازی استفاده کردیم.در اینجا به شرح مختصری از این روش می پردازیم:
برای تعیین معادله یک خط راست تنها دو نقطه کافی است. برای یک منحنی درجه دو ، تنها ۳ نقطه نیاز است. به طور کلی هرچه درجه چند جمله ای بیشتر باشد، به نقاط بیشتری نیاز داریم تا بتوانیم معادله این چند جمله ای را بنویسیم. (تعداد نقاط از درجه چند جمله ای یکی بیشتر است) در رگرسیون به دنبال این هستیم تا با چند جمله ای هایی با درجه بسیار کمتر از تعداد نقطه ها، تقریب خوبی از داده ها داشته باشیم. (با توجه به انچه در قبل دیدیم اگر درجه چند جمله ای یکی کمتر از تعداد نقاط باشد ، چند جمله ای رو تمامی نقاط فیت می شود و به عبارتی دچار $Overfitting$ می شویم.) معادله کلی تابع چند جمله ای مرتبه k به صورت زیر است:

$$f(x) = a_0 + a_1x + a_2x^2 + ... + a_kx^k$$