Accruals

Name: Abdul Moiz

I downloaded monthly data from CRSP for all the stocks with PERMNO, date, Share Code, Exchange Code, Return, Price and Shares Outstanding:

Preparing CRSP Data:

1. **Universe of Stocks:** I extracted the full data and did not pre filter by any share or exchange code. The data taken was from January 1961 to December 2018 in order to match with the Compustat data.
2. **-66,-77,-88,-99 and A,B,C,P,S,T Returns:** Places where returns were huge negative implied that they were missing or not available and putting 0s in the place would have manipulated the overall returns for calculating of deciles and therefore I dropped them from overall conclusion by turning them into NAs. For C Returns, since they are returns on first day of company, I dropped them because returns didn't exist when company was formed and returns from the first day only matter for our portfolios. Using same logic, I dropped all returns with character returns.
3. **Excess Returns:** Excess returns were calculated by subtracting holding period return of the company from that of S&P 500 value weighted index.
4. **Share and Exchange Code:** I subset the data using CRSP share codes 10 and 11 (i.e., common stocks; no ADRs), and Exchange codes 1, 2 and 3 (NYSE, AMEX or NASDAQ).
5. **Missing Data:** After accounting for everything, I dropped the data points that were missing because of shifting (lagging/leading) variables.

Preparing Compustat Quarterly Data:

I imported the following accounting variables for our accruals strategy: *Cash and Short Term Investments, Total Current Assets, Total Assets, Debt in Current Liabilities, Depreciation and Amortization Expense, Total Current Liabilities and Tax Payable.*

1. **Period:** I took period from January 1961 to December 2018.
2. **Taxes Payable:** Wherever taxes payable were NA, I changed them to zero by assuming that company had no outstanding taxes to save data points and generate meaningful results.

Accrual Factor:

I defined the accrual Factor, according to notes, as follows:

$$Accrual\ Factor = \frac{\Delta Non\ Cash\ Current\ Assets - \Delta Current\ Liabilities^{*} - Depreciation\ Expense}{Average\ Total\ Assets}$$

Where,

$$Non\ Cash\ Current\ Assets = Current\ Assets - Cash\ and\ Short\ Term\ Investments$$

$$Current\ Liabilities^{*} = Current\ Liabilities - Debt\ in\ Current\ Liabilities - \ Tax\ Payable$$

Factor Construction:

1. **Merging:** Since the fundamental data is available to us on quarterly basis, it made sense to rebalance the portfolios quarterly. I calculated quarterly returns from the crsp data earlier and merged with the Compustat data using 'Calendar Quarter' column in Compustat to get returns and fundamental for each company for each quarter.
2. **Lookahead Bias:** In order to avoid the lookahead bias, I shifted the fundamental data available to us forward by one quarter because there is normally a lag of 60-90 days between quarter-end and the actual release of financial statements. For example, quarterly data for March End was actually used in June end for portfolio construction.
3. **Standardization and Winsorization:** In order to reduce the effect of outliers and get approximately equal division of companies in each bin, I standardized my accrual factor and winsorized it at ± 3 standard deviation.
4. **Portfolios:** I divided all the companies into 5 bins in each quarter where Bin 1 represents companies with lowest accrual score and Bin 5 with highest. The Long Short Portfolio is Bin 1 minus Bin 5 where hypothesis is that companies with lowest accruals will outperform ones with highest accruals in cross-section.
5. **Returns:** Each quarter, after getting portfolio bins, I calculated the performance of each bin by getting returns for next year i.e. for next 4 Quarters. To give an example, for bins created in June end, performance was calculated from August of current year to June of next year.

Results:

Accruals strategy was found to be very powerful in the cross-section from 1961 to 2018 after accounting for all biases. Bin 1 resulted in an annualized excess return of 9.56% **(Exhibit 1)** whereas Bin 5 gave 5.04% with monotonic decrease across the bins **(Exhibit 2)** proving that the factor is indeed robust in out of sample. The volatility was around 9% for both extreme bins and therefore the sharpe ratio was 0.89 for the Long-Short Strategy. All the returns are in excess of S&P 500. If we plot cumulative returns of all bins **(Exhibit 3)**, we can see that, even when during market recession when Bin 1 through Bin 5 decline, the long short strategy stays positive implying that Bin 5 fell more than Bin 1 in market downturns. The good performance may also be attributed to the lower turnover of the strategy as it is rebalanced only quarterly. However, the strategy has been relatively flat/negative in recent quarters possibly due to crowding. **(Exhibit 4)**

Exhibit 1 (Statistics)

	1	2	3	4	5	LS
Mean (%)	9.568	8.174	7.421	5.822	5.046	4.522
Median (%)	7.790	7.494	5.831	4.857	3.818	4.188
Std. Dev (%)	9.449	7.033	6.583	6.721	9.049	5.046
Sharpe Ratio	**1.013**	**1.162**	**1.127**	**0.866**	**0.558**	**0.896**

Exhibit 2 (Portfolio Bins Returns)

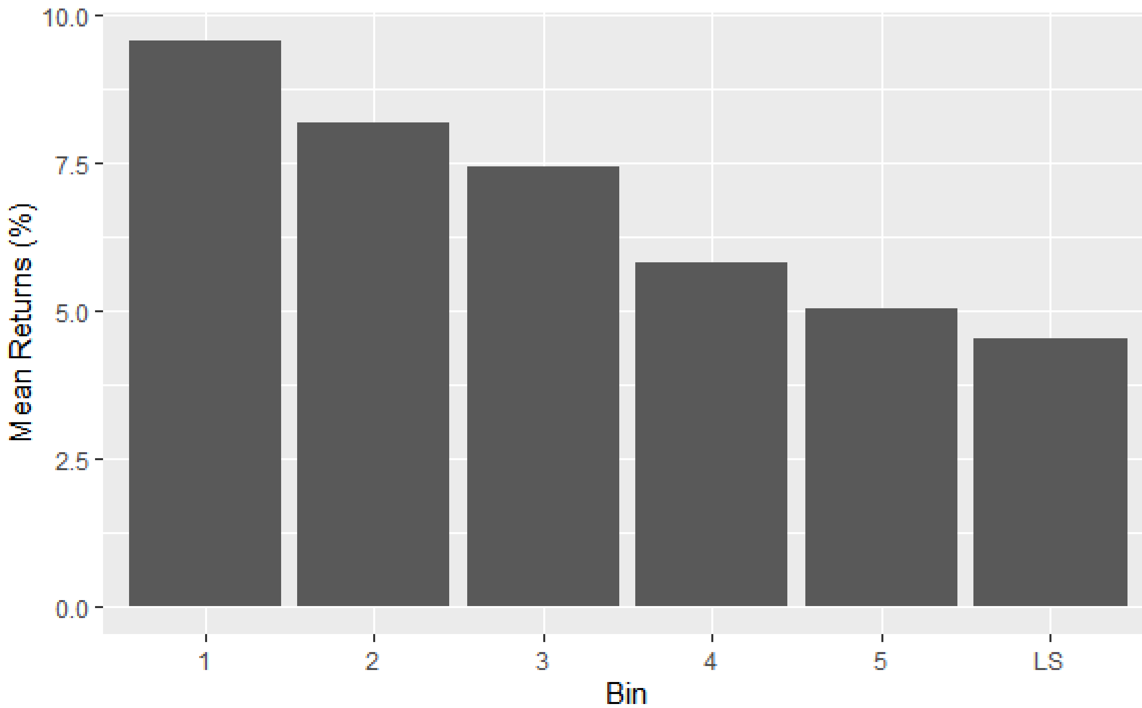

Exhibit 3 (Cumulative Returns)

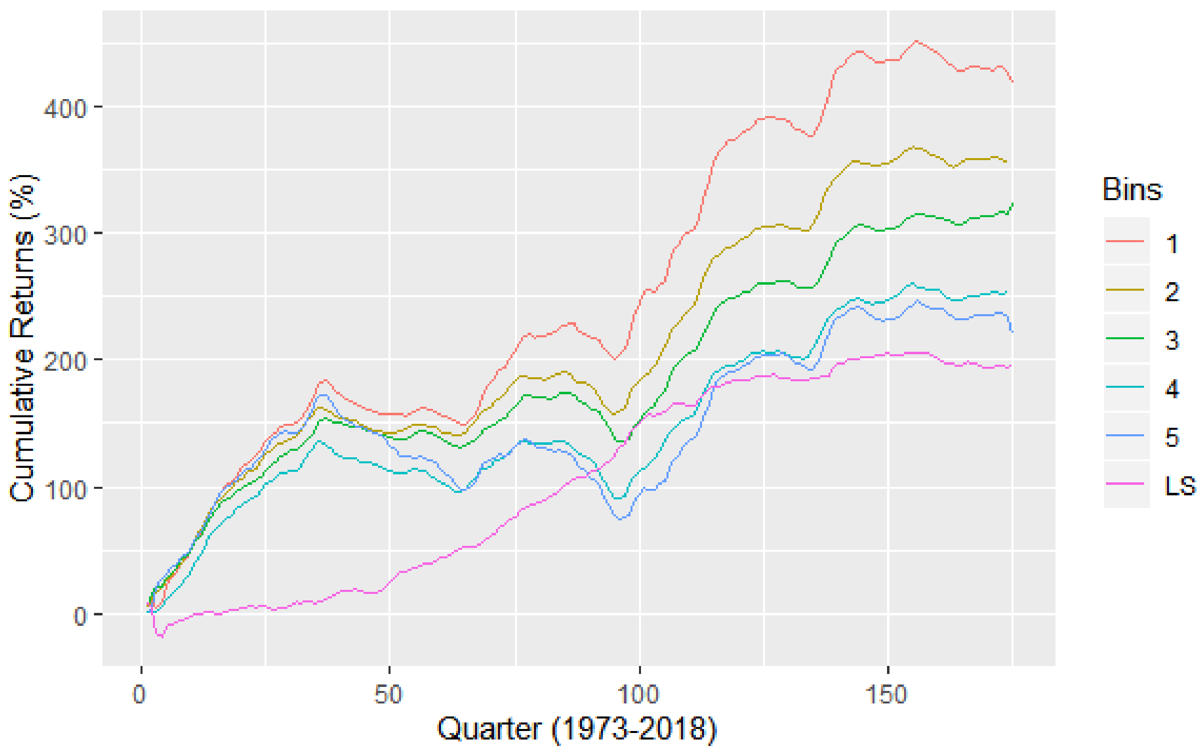

Exhibit 4 (Returns – Long Short Strategy)

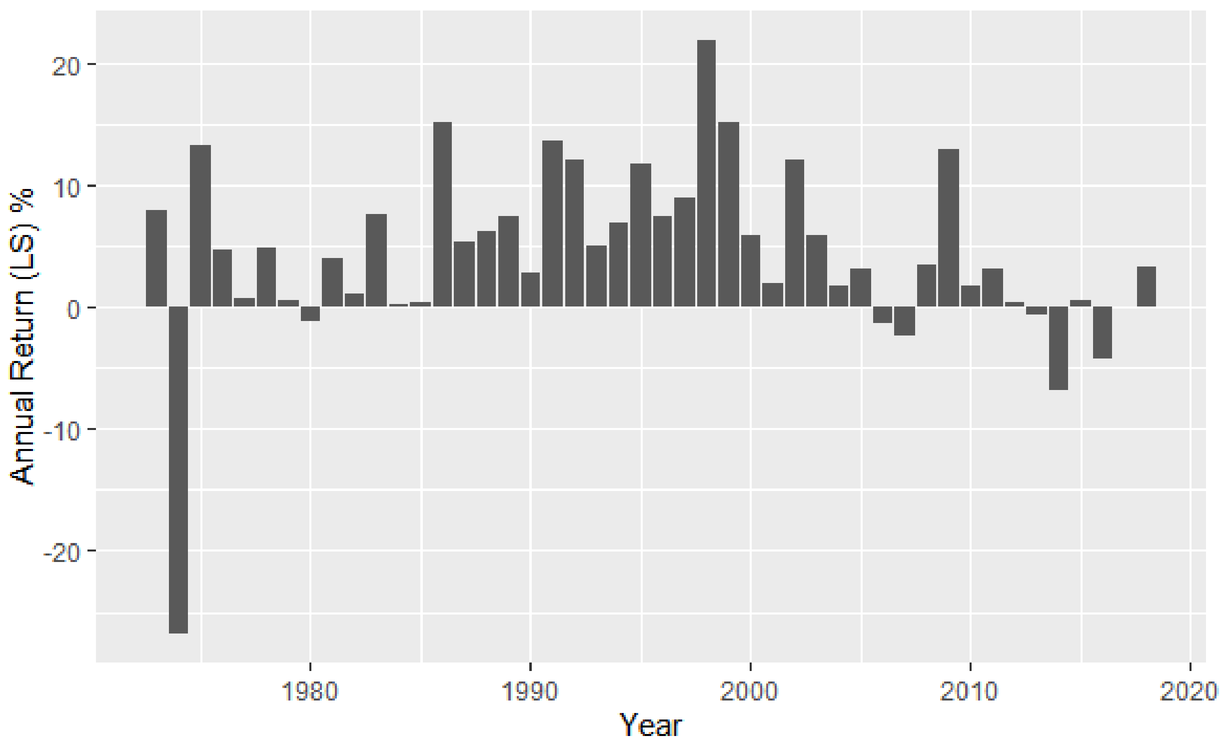

Exhibit 5 (Code)

I used R programming language for the project. The code is detailed as follows:

Importing and Cleaning Data

```
crsp=fread('crsp_all.txt')
compustat=fread('compustat_all.txt')

crsp[,date:=as.Date(parse_date_time(date,'ymd'))]
crsp=crsp[SHRCD==10 | SHRCD==11, ]
crsp=crsp[EXCHCD==1 | EXCHCD==2 | EXCHCD==3,]

for (i in c('RET')){
  crsp[get(i) %in% c('','A','B','C','P','S','T'),paste0(i):=NA]
  crsp[,paste0(i):=as.numeric(get(i))]
  crsp[get(i) %in% c(-66,-77,-88,-99),paste0(i):=NA]
}

crsp[,e_ret:=RET-sprtrn,]
crsp[,month:=month(date)]
crsp[,year:=year(date)]



crsp_clean=crsp[,.(PERMNO,month,year,COMNAM,e_ret)]

crsp_clean[,qtr:=ifelse(month %in% c(1,2,3),'Q1',ifelse(month %in% c(4,5,6),'
Q2',ifelse(month %in% c(7,8,9),'Q3','Q4')))]
crsp_clean[,datacqtr:=paste0(year,qtr)]
crsp_clean=crsp_clean[,qtr:=NULL,]
colnames(crsp_clean)[1]='LPERMNO'
```

Creating Factor in Compustat

```
setorder(compustat,LPERMNO,datacqtr)
compustat[is.na(txpq),txpq:=0]

compustat[,non_cash_ca:=actq-cheq]
compustat[,change_non_cash_ca:=non_cash_ca-shift(non_cash_ca),by=LPERMNO]

compustat[,cl_exclusive:=lctq-dlcq-txpq]
compustat[,change_cl_exclusive:=cl_exclusive-shift(cl_exclusive),by=LPERMNO]

compustat[,avg_ta:=(atq+shift(atq))/2,by=LPERMNO]

compustat[,accrual:=(change_non_cash_ca-change_cl_exclusive-dpq)/avg_ta]

compustat_clean=na.omit(compustat[,.(LPERMNO,datacqtr,accrual)])
```

Merging CRSP/Compustat, Accounting for Biases

```
normalize=function(x){
  x=(x-mean(x,na.rm = T))/sd(x,na.rm = T)
  return(Winsorize(x,probs=c(0.001,0.999),na.rm = T))
}

accruals=merge(crsp_clean,compustat_clean,by=c('LPERMNO','datacqtr'),all.x=T)

#Getting Quarterly Returns. Because of Fundamental Data, I rebalance quarterl
y
accruals[,q_ret:=sum(e_ret),by=.(LPERMNO,datacqtr)]

accruals_q=unique(accruals[,.(LPERMNO,datacqtr,COMNAM,accrual,q_ret)])

#Leading accruals by a quarter to avoid Lookahead Bias
setorder(accruals_q,LPERMNO,datacqtr)
accruals_q[,accrual_lead:=shift(accrual,n=1),by=LPERMNO]

#Getting Returns for next Year
accruals_q[,f_qret:=Reduce(`+`,shift(q_ret,n=1:4,type='lead'))/4,by=LPERMNO]

#Winsorizing our Factor to adjust for outliers
accruals_q[,accrual_lead_win:=normalize(accrual_lead),by=datacqtr]

#Making Bins for accruals
accruals_q[,bins:=.bincode(accrual_lead_win,breaks = quantile(accrual_lead_wi
n,probs=c(0:5)/5,na.rm = T),include.lowest = T),by=datacqtr]

#Getting Returns for each decile
accruals_q[,c_ret:=mean(f_qret,na.rm = T),by=.(datacqtr,bins)]
```

Portfolio Returns (All returns are in excess of S&P 500)

```
portfolio=na.omit(unique(accruals_q[,.(datacqtr,bins,c_ret)]))
setorder(portfolio,datacqtr,bins)

portfolio_bins=dcast.data.table(portfolio[c_ret!=0,],datacqtr~bins,value.var
= 'c_ret')

#Accrual Long Short (Lowest Decile - Highest Decile)

portfolio_bins[,LS:=`1`-`5`]

portfolio_bins_melt=melt.data.table(portfolio_bins,id.vars='datacqtr')
colnames(portfolio_bins_melt)[2]='Bins'
```

Summary Statistics

```
stats=data.frame(matrix(0,nrow=4,ncol=6))
rownames(stats)=c('Mean','Median','Std. Dev ','Sharpe Ratio')
colnames(stats)=colnames(portfolio_bins[,2:7])
```

```r
stats[1,]=apply(portfolio_bins[,2:7],2,function(x) mean(x,na.rm=T))*4*100
stats[2,]=apply(portfolio_bins[,2:7],2,function(x) median(x,na.rm=T))*4*100
stats[3,]=apply(portfolio_bins[,2:7],2,function(x) sd(x,na.rm=T))*sqrt(4)*100
stats[4,]=stats[1,]/stats[3,]


stats

##                     1        2        3        4        5        LS
## Mean         9.568255 8.174039 7.420832 5.821756 5.046394 4.5218617
## Median       7.789695 7.494297 5.831325 4.857402 3.818145 4.1883030
## Std. Dev     9.448796 7.033404 6.582919 6.720735 9.048856 5.0457192
## Sharpe Ratio 1.012643 1.162174 1.127286 0.866238 0.557683 0.8961778
```

Graphs and Cumulative Returns

```r
stats_melt=melt(stats[1,])

## No id variables; using all as measure variables

ggplot(stats_melt,aes(x=variable,y=value))+geom_bar(stat='identity')+ylab('Me
an Returns (%)')+xlab('Bin')

portfolio_LS=portfolio_bins_melt[Bins=='LS',.(datacqtr,Bins,value)]
portfolio_LS[,date:=year(as.Date(parse_date_time(substr(datacqtr,1,4),'y')))]
portfolio_LS[,cumvalue:=sum(value,na.rm = T),by=date]
portfolio_LS_years=unique(portfolio_LS[,.(date,cumvalue)])

ggplot(portfolio_LS_years,aes(x=date,y=cumvalue*100))+geom_bar(stat='identity
')+xlab('Year')+ylab('Annual Return (LS) %')

portfolio_bins_melt=na.omit(portfolio_bins_melt)
portfolio_bins_melt[,cumret:=cumsum(value),by=Bins]

#Cumulative Returns of all Bins

portfolio_bins_melt[,qtr:=1:.N,by=Bins]

ggplot(portfolio_bins_melt,aes(x=qtr,y=cumret*100,group=Bins))+geom_line(aes(
col=Bins))+ylab('Cumulative Returns (%)')+xlab('Quarter (1973-2018)')
```